User Manual for GUIDE ver. 46.1*

Wei-Yin Loh
Department of Statistics
University of Wisconsin–Madison

January 24, 2026

Contents

1 Warranty disclaimer **5**

2 Introduction **5**
2.1 Installation 7
2.2 LaTeX 10

3 Program operation **11**
3.1 Required files 11
3.2 Running GUIDE 14

4 Classification: fish data **15**
4.1 Univariate splits 15
4.1.1 Input file creation 15
4.1.2 Contents of classout.txt 17
4.1.3 Contents of classfit.txt 27
4.1.4 Contents of class.r 27
4.2 Linear splits and discriminant models 31
4.2.1 Input file creation 31

*Based on work partially supported by grants from the U.S. Army Research Office, National Science Foundation, National Institutes of Health, Bureau of Labor Statistics, USDA Economic Research Service, and Eli Lilly & Co. Work on precursors to GUIDE additionally supported by IBM Research and Pfizer.

4.2.2 Contents of qda.out 33

5 Missing-value flag variables **43**
5.1 Classification: univariate splits 46
5.1.1 Input file generation 46
5.1.2 Contents of output file 48
5.2 Classification: linear trees 57

6 Least squares regression **64**
6.1 Piecewise constant 64
6.1.1 Input file creation 64
6.1.2 Contents of `cons.out` 66
6.1.3 Population mean estimation 73
6.2 Piecewise simple polynomial 74
6.2.1 Input file creation 74
6.2.2 Partial output 76
6.2.3 Plots of data 80
6.3 Stepwise linear 80
6.3.1 Input file creation 84
6.3.2 Results 86

7 Quantile regression: CE data **87**
7.1 Piecewise constant: one quantile 87
7.1.1 Input file creation 87
7.2 Best simple linear 96
7.2.1 Input file creation 96
7.3 Two quantiles 107
7.3.1 Input file creation 107
7.3.2 Output file 109

8 Periodic variables: NHTSA data **115**
8.1 Regression 115
8.1.1 Input file creation 118
8.1.2 Results 120
8.2 Classification 125
8.2.1 Univariate splits 125
8.2.2 Linear splits 133
8.2.3 Results 135

9 Poisson regression **146**
9.1 Piecewise-constant: solder data . . . 146
9.1.1 Input file creation . . . 146
9.2 Multiple linear: solder data . . . 149
9.2.1 Input file creation . . . 149
9.2.2 Contents of `mul.out` . . . 150
9.3 Offset variable: lung cancer data . . . 155
9.3.1 Input file creation . . . 157
9.3.2 Results . . . 158

10 Censored response: RHC data **162**
10.1 Proportional hazards . . . 167
10.1.1 Input file generation . . . 167
10.1.2 Output file . . . 169
10.2 Restricted mean event time . . . 179
10.2.1 Input file creation . . . 179
10.2.2 Contents of `rest.out` . . . 181

11 Randomized treatments **186**
11.1 Multiple treatment arms: CAPE data . . . 186
11.1.1 Input file creation . . . 186
11.1.2 Contents of `gi.out` . . . 188
11.2 Censored response: prop. hazards . . . 193
11.2.1 Without linear prognostic control . . . 195
11.2.2 Simple linear prognostic control . . . 203
11.3 Censored response: restricted mean . . . 214
11.3.1 Without linear prognostic control . . . 214
11.3.2 With linear prognostic control . . . 223

12 Nonrandomized treatments: RHC data **223**
12.1 Proportional hazards . . . 225
12.1.1 Gi option . . . 225
12.2 Restricted mean . . . 236
12.2.1 Gi option . . . 236

13 Multiresponse: NMES data **241**
13.1 Input file creation . . . 244
13.2 Contents of `mult.out` . . . 246

14 Longitudinal response **251**
14.1 Input file creation . . . 253
14.2 Contents of `wage.out` . . . 255

15 Logistic regression **263**
15.1 Piecewise constant . . . 264
15.1.1 Input file creation . . . 264
15.1.2 Contents of `logitc.out` . . . 265
15.2 Simple linear . . . 273
15.2.1 Input file creation . . . 273
15.2.2 Contents of `logits.out` . . . 276

16 Importance scoring **280**
16.1 Classification: RHC data . . . 280
16.1.1 Input file creation . . . 280
16.1.2 Contents of `imp.out` . . . 281
16.2 Censored response with R variable . . . 285
16.2.1 Input file creation . . . 285
16.2.2 Partial contents of imp_surv.out . . . 288

17 Propensity scores **291**
17.1 Causal inference . . . 291
17.1.1 Input file creation . . . 292
17.1.2 Contents of `propen.out` . . . 293
17.2 Missing-value imputation . . . 304
17.2.1 Input file creation . . . 305
17.2.2 Output file . . . 306

18 Differential item functioning **313**

19 Bootstrap confidence intervals **317**

20 Tree ensembles **320**
20.1 GUIDE forest: CE data . . . 322
20.1.1 Input file creation . . . 322
20.1.2 Contents of `gf.out` . . . 324
20.2 Bagged GUIDE . . . 326

21 Other features **326**
21.1 Pruning with test samples 326
21.2 Prediction of test samples 326
21.3 GUIDE in R and in simulations 327
21.4 Generation of powers and products 328
21.5 Data formatting functions 329

A CE variables **332**

1 Warranty disclaimer

Redistribution and use in binary forms, with or without modification, are permitted provided that the following condition is met:

Redistributions in binary form must reproduce the above copyright notice, this condition and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY WEI-YIN LOH "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL WEI-YIN LOH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the author and should not be interpreted as representing official policies, either expressed or implied, of the University of Wisconsin.

2 Introduction

GUIDE is an acronym for *Generalized, Unbiased, Interaction Detection and Estimation*. It is an algorithm for construction of classification and regression trees and forests. It is a descendent of the FACT (Loh and Vanichsetakul, 1988), SUPPORT

(Chaudhuri et al., 1994, 1995), QUEST (Loh and Shih, 1997), CRUISE (Kim and Loh, 2001, 2003), and LOTUS (Chan and Loh, 2004; Loh, 2006a) algorithms. GUIDE is the only classification and regression tree algorithm with all these features:

1. Unbiased variable selection for data with and without missing values
2. Unbiased importance scoring and thresholding of predictor variables
3. Automatic handling of missing values without requiring prior imputation
4. Allowance for multiple missing-value codes and missing-value flag variables
5. Optional automatic creation of missing-value indicator variables for regression
6. Periodic or cyclic variables, such as angular direction, hour of day, day of week, month of year, and seasons
7. Subgroup identification for differential treatment effects
8. Propensity score estimation
9. Linear splits for classification and regression trees
10. Kernel and nearest-neighbor node models for classification trees
11. Weighted least squares, least median of squares, logistic, quantile, Poisson, relative risk (proportional hazards), and propensity score models
12. Univariate, multivariate, censored, and longitudinal response variables
13. Piecewise polynomial, multiple, and stepwise linear regression models
14. Pairwise interaction detection at each node
15. Categorical variables may be used for splitting only, fitting only (via 0-1 dummy variables), or both in regression trees
16. Tree ensembles (bagging and forests)
17. Tree diagrams in LaTeX code
18. Predicted functions in R code

Tables 1 and 2 compare the features of GUIDE with QUEST, CRUISE, C4.5 (Quinlan, 1993), CTREE (Hothorn et al., 2006), MOB (Hothorn and Zeileis, 2015), RPART (Therneau et al., 2017) [1], and M5' (Quinlan, 1992; Witten and Frank, 2000).

The GUIDE algorithm is documented in Loh (2002) for regression trees and Loh (2009) for classification trees. Reviews of the subject may be found in Loh (2008a, 2011, 2014). Advanced features of the algorithm are reported in Chaudhuri and Loh (2002), Loh (2006b, 2008b), Kim et al. (2007), Loh et al. (2007, 2019b, 2016, 2015, 2019c), and Loh and Zhou (2021). For third-party applications of GUIDE and predecessors, see `http://www.stat.wisc.edu/~loh/apps.html`. This manual demonstrates use of the GUIDE software and interpretation of the results.

2.1 Installation

GUIDE is available free as compiled 64-bit executables for Linux, macOS, and Windows. Data and DSC files used in this manual are in the zip file.

Linux: There are two executables to choose from, compiled with **gfortran** versions 11.4.0 and 13.3.0. Unzip the files with "`gunzip guide.gz`" and, if necessary, make it executable by typing "`chmod a+x guide`" in a `Terminal` window. To execute, type "`./guide`".

macOS: There are two versions to choose from, one for Apple Arm processors (macOS Sequoia 15.4.1) and the other for Intel processors (macOS Monterey 12.7.6). Download the desired `guide.zip` file and double-click its icon to see if it opens. You will likely see a waring message saying, "Apple could not verify that guide is free of malware." Instead of choosing to move it to Trash, choose "Done" and do this: go to System Settings > Privacy & Security and scroll down to "Security". Look for an entry for "guide" and click "Open Anyway" and then confirm with your password. More details are given here `https://support.apple.com/guide/mac-help/open-a-mac-app-from-an-unknown-develope` Move the `guide` file outside the `Downloads` folder (if that is where you have it), open a `Terminal` window, change directory to where `guide` is located, and type the command "`./guide`" to run it.

Windows: Download the file `guide.zip` and unzip it (right-click on file icon and select "Extract all"). The resulting file `guide.exe` may be placed in one of three places:

[1] RPART is an implementation of CART (Breiman et al., 1984) in R. CART is a registered trademark of California Statistical Software, Inc.

Table 1: Comparison of GUIDE, QUEST, CRUISE, CART, C4.5, and CTREE classification tree algorithms. Node models: S = simple, K = kernel, LDA = linear discriminant, QDA = quadratic discriminant, N = nearest-neighbor.

	GUIDE	QUEST	CRUISE	RPART	C4.5	CTREE
Unbiased splits	Yes	Yes if no missing values	Yes	No	No	Yes if no missing values
Splits per node	2	2	≥ 2	2	2	2
Linear splits	Yes	Yes	Yes	Yes	No	No
Categorical variable splits	Subsets	Subsets	Subsets	Subsets	Atoms	Subsets
Periodic variable splits	Yes	No	No	No	No	No
Interaction tests	Yes	No	Yes	No	No	No
Class priors	Yes	Yes	Yes	Yes	No	No
Misclassification costs	Yes	Yes	Yes	Yes	No	No[a]
Case weights	No[b]	No	No	Yes	Yes	Yes[c]
Node models	S, K, N, LDA, QDA	S	S, LDA	S	S	S
Missing values in splits	Missing as observed	Node mean or mode imputation	Surrogate splits	Surrogate splits	Weights	Random splits[d]
Missing-value flag variables	Yes	No	No	No	No	No
Pruning	Yes	Yes	Yes	Yes	No	No
Tree diagrams	Text and LaTeX			R	Text	R
Bagging	Yes	No	No	No	No	No
Forests	Yes	No	No	No	No	cforest
Importance scores	Yes	No	No	Yes	No	Yes

[a] user defined
[b] positive weights treated as 1
[c] non-negative integer counts
[d] surrogate splits is a non-default option

Table 2: Comparison of GUIDE, RPART, M5', and MOB regression tree algorithms

	GUIDE	RPART	M5'	MOB
Unbiased splits	Yes	No	No	Yes
Linear splits	Yes	No	No	No
Interaction tests	Yes	No	No	No
Loss functions	Weighted least squares, least median of squares, logistic, quantile, Poisson, proportional hazards	Least squares, least absolute deviations	Least squares	Generalized linear models
Censored response	Yes	Yes	No	Yes
Longitudinal and multi-response	Yes	No	No	Yes
Node models	Constant, multiple, stepwise linear, polynomial, ANCOVA	Constant	Constant, stepwise	Constant, multiple linear
Variable roles	Split only, fit only, both, neither, weight, offset	Split only	Split and fit	Similar to GUIDE
Categorical variable splits	Subsets	Subsets	Singletons	Subsets
Periodic variables	Yes	No	No	No
Tree diagrams	Text and LaTeX	R	PostScript	R
Sampling weights	Yes	Yes	No	No[a]
Transformations	Powers and products	No	No	Yes
Missing values in splits	Missing as observed or imputed with indicators	Surrogate splits	Mean/mode imputation	Random splits
Missing values in linear predictors	Node mean imputation & missing-value indicators	N/A	Global imputation	Omitted
Missing-value flag variables	Yes	No	No	No
Bagging & forests	Yes & yes	No & no	No & no	cforest
Importance scores	Yes	Yes	No	Yes[b]

[a] replicate weights only
[b] from cforest or ctree

1. Top level of your `C` drive. Type "`C:\guide`" in a **Command Prompt** window to execute—see Section 3.1.
2. A folder that contains your data files. Type "`guide`" in that folder to execute.
3. A folder on your search path. Type "`guide`" anywhere to execute.

2.2 LaTeX

GUIDE uses the public-domain software LaTeX to produce tree diagrams. The .tex files produced by GUIDE can be edited to change colors, node sizes, etc., in the trees—see Pstricks User Guide.

There are two ways to produce postscript and pdf versions of the diagrams:

1. Upload the .tex file produced by GUIDE to Overleaf, and compile it with the **XeLaTeX** flavor of LaTeX.
2. Download and install the LaTeX software from:

   Linux: TeX Live

   Mac: MacTeX or MiKTeX. Both include the **TeXShop** GUI app.

   Windows: MiKTeX. Choose Net installer under the "All downloads" tab.

   There are two ways to generate pdf files from the .tex files. The following example assumes that the LaTeX file is named `diagram.tex`.

   (a) **Terminal window (simplest).** Type this single command in the **Terminal** (Linux or Mac) or **Command Prompt** (Win) window that contains `diagram.tex`:

   - `xelatex diagram`

   This will produce pdf file named `diagram.pdf` in the same folder.

   (b) **TeXShop, TeXworks, or TeXStudio.** Double-click `diagram.tex` to load it into one of these apps. Select **XeLaTeX** to typeset it to pdf.

In macOS, the **Preview** app can open postscript and pdf files for conversion to jpg, png, and other formats. In Windows, the same can be done with ImageMagick. To insert pdf figures in MS PowerPoint or Word documents, convert them to jpg for macOS and png for Windows, or copy-and-paste them from the pdf viewer.

3 Program operation

GUIDE runs within a **terminal window** of the computer operating system.

Linux. Any terminal program will do.

macOS. The program is called **Terminal**; it is in the **Applications Folder**.

Windows. The terminal program is started from the **Start button** by choosing **All Programs** → **Accessories** → **Command Prompt**

After the terminal window is opened, change to the folder where the data and program files are stored. Mac and Windows users are unfamiliar with terminal commands may consult
https://wiredpen.com/resources/basic-unix-commands-for-osx/
and https://www.geeksforgeeks.org/techtips/most-useful-cmd-commands-in-windows/, respectively.

Do not double-click the GUIDE icon on the desktop!

3.1 Required files

GUIDE requires two text files.

Data file: This file contains the data from the training sample. Each data record consists of observations on the dependent variable, the predictor (i.e., X or independent) variables, and optional weight, missing value flag, time, offset, periodic, and event indicator (for censored responses) variables. Entries in each record are comma, space, or tab delimited (multiple spaces are treated as one space, but not for commas). A record can occupy more than one line in the file, but each record must begin on a new line.

Values of categorical variables can contain any ascii character except single and double quotation marks, which are used to enclose values that contain spaces and commas. Values can be up to 60 characters long. Class labels are truncated to 10 characters in tabular output.

A common problem among first-time users is getting the data file in proper shape. If the data are in a spreadsheet and there are **no empty cells**, export them to a **MS-DOS Comma Separated** (csv) file (the MS-DOS CSV format takes care of carriage return and line feed characters properly). If there are empty cells, a good solution is to read the spreadsheet into R (using `read.csv`

with proper specification of the `na.strings` argument), verify that the data are correctly read, and then export them to a text file using either `write.table` or `write.csv`.

Note to R users: GUIDE can optionally generate R code for the tree model and its prediction function. Because GUIDE treats `"NA"` (with quotes) the same as `NA` (without quotes), the two are treated as missing values in the R function.

DSC file: "DSC" is an abbreviation for **"data specification and control."** This text file provides information about the name and location of the data file, column locations and names of the variables, and their roles in the analysis. Different models may be fitted by changing the roles of the variables. An example is `fishdsc.txt` (in www.stat.wisc.edu/~loh/treeprogs/guide/datafiles.zip), whose contents are:

```
fishdata.txt
NA
2
1 species d
2 weight n
3 length1 n
4 length2 n
5 length3 n
6 heightpc n
7 widthpc n
8 sex c
```

The 1st line gives the name of the data file `fishdata.txt`, which is also in www.stat.wisc.edu/~loh/treeprogs/guide/datafiles.zip. If the file is not in the current folder, its full path must be given (e.g., `"c:\data\fishdata.txt"` for Windows users or `"~/data/fishdata.txt"` for Mac users) surrounded by matching quotes (because it contains non-alphanumeric characters). The 2nd line gives the missing value code, which can be up to 80 characters long. If it contains non-alphanumeric characters, it too must be surrounded by matching quotation marks. A missing value code **must appear** in the second line of the file even if there are no missing values in the data (in which case any character string not present among the data values can be used). The 3rd line gives the line number of the first data record in the data file. A "2" is shown here because

the variable names appear in the first line of `fishdata.txt`. If the 1st line of the data file contains the 1st record, this entry would be "`1`" . Blank lines in the data and DSC files are ignored. The column location, name and role of each variable comes next (in that order), with one line for each variable.

Variable names must begin with an alphabet and be not more than 60 characters long. If a name contains non-alphanumeric characters, it must be enclosed in matching single or double quotes. Spaces and the four special characters, `#`, `%`, `{`, and `}`, in a variable name are replaced by dots (periods) in the outputs. Variable names are truncated to 10 characters in tabular text output (but not in R output). Leading and trailing spaces in variable names are dropped.

The letters (lower or upper case) below are the permissible roles.

b Categorical variable used **b**oth for splitting and for node modeling in regression. Such variables are converted to 0-1 dummy variables when fitting models within nodes for regression. They are converted to **c** type for classification.

c **C**ategorical variable used for splitting only.

d **D**ependent variable or **d**eath indicator variable. Except for longitudinal and multiple response data (Sec. 13), there can only be one **d** variable. For censored responses in proportional hazards models, it is the 0-1 event (**d**eath) indicator. For all other models, it is the response variable. It can take character string values for classification.

e **E**stimated probability variable, for logistic regression without **r** variable; see Section 15 for an example.

f Numerical variable used only for **f**itting the linear models in the nodes of the tree. It is not used for splitting the nodes and is disallowed in classification.

i Categorical variable internally converted to 0-1 **i**ndicator variables for fitting regression models within nodes.

m **M**issing value flag variable. Each such variable should follow immediately after a **c**, **n** or **s** variable in the DSC file. Missing value flag variables associated with any other variable type (including **b** and **p**) should be specified as **c**.

n **N**umerical variable used both for splitting the nodes and for fitting the node regression models. It is converted to type **s** in classification.

p **P**eriodic (cyclic) variable, such as an angle, hour of day, day of week, or month of year. See Sec. 8 for an example.

Table 3: Predictor variable role descriptors

Type of variable	Role of variable		
	Split nodes	Fit node models	Both
Categorical	`c`	`i`	`b`
Numerical	`s`	`f`	`n`

r Categorical treatment (**R**x) variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes.

s Numerical-valued variable only used for **s**plitting the nodes. It is not used as a linear predictor in in regression models. It is suitable for ordinal categorical variables if they take numerical values that reflect the orderings.

t **T**ime variable, either time to event for proportional hazards models or observation time for longitudinal models.

w **W**eight variable for weighted least squares regression or for excluding observations in the training sample from tree construction. See Sec. 21.2 for the latter. Except for longitudinal models, a record with a missing value in a `d`, `t`, or `z`-variable is automatically assigned zero weight.

x E**x**cluded variable. Models may be fitted to different subsets of variables by indicating excluded variables in the DSC file without editing the data file.

z **O**ffset variable used only in Poisson regression.

Table 3 summarizes the possible roles for predictor variables.

3.2 Running GUIDE

GUIDE uses a "restaurant" model that requires two invocations of the program. The first invocation creates an input file (similar to a restaurant order) and the second invocation reads the input file and fits the data. The first invocation is started by typing the word "guide" (lowercase) name in a terminal window and then typing "`1`" to answer some questions and save the answers into a file. In the following, the sign (>) is the computer prompt (not to be typed!).

```
> guide
GUIDE Classification and Regression Trees and Forests
Version 46.1 (Build date: January 23, 2026)
Compiled with GFortran 15.1.0 on macOS Tahoe 26.0.1
Copyright (c) 1997-2025 Wei-Yin Loh. All rights reserved.
Software based upon work partially supported by the U.S. Army Research Office,
National Science Foundation, National Institutes of Health,
```

```
Bureau of Labor Statistics, USDA Economic Research Service, and Eli Lilly.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create a GUIDE input file
```

4 Classification: fish data

The file `fishdata.txt` give measurements on 7 predictor variables for 7 species of 159 fish. The predictor variables are weight (gm), length1 (nose to beginning of tail, cm), length2 (nose to notch of tail, cm), length3 (nose to end of tail, cm), heightpc (maximal height as percent of length3), widthpc (maximal width as percent of length3), and sex (female, male, and unknown). The species variable takes values Bream, Parkki, Perch, Pike, Roach, Smelt, and Whitefish. Further details are given in https://search.r-project.org/CRAN/refmans/rrcov/html/fish.html. The DSC file for this dataset is `fishdsc.txt` mentioned above. We first show how to create an input file to fit a classification tree model with univariate splits.

4.1 Univariate splits

4.1.1 Input file creation

Most of the questions prompted by GUIDE have default answers indicated by the symbol `<cr>` ("`cr`" stands for "carriage return"). For example, "`<cr>=1`" means that the answer "`1`" may be selected by simply hitting the carriage return once, as demonstrated several times below.

```
> guide
GUIDE Classification and Regression Trees and Forests
Version 46.1 (Build date: January 23, 2026)
Compiled with GFortran 15.2 on macOS Tahoe 26.2
Copyright (c) 1997-2025 Wei-Yin Loh. All rights reserved.
Software based upon work partially supported by the U.S. Army Research Office,
National Science Foundation, National Institutes of Health,
Bureau of Labor Statistics, USDA Economic Research Service, and Eli Lilly.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: classin.txt
```

```
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: classout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: fishdsc.txt
Reading DSC file ...
Training sample file: fishdata.txt
Missing value code: NA
Records in data file start on line 2
6 N variables changed to S
D variable is species
Reading data file ...
Number of records in data file: 159
Length of longest entry in data file: 9
Checking for missing values ...
Finished checking
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 7
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
    Class  #Cases     Proportion
bream          35     0.22012579
parkki         11     0.06918239
perch          56     0.35220126
pike           17     0.10691824
roach          20     0.12578616
smelt          14     0.08805031
whitefish       6     0.03773585
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       159          0          2        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 159
```

```
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Default max. number of split levels: 12
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): class.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: classfit.txt
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: class.r
Input rank of top variable to split root node ([1:7], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < classin.txt
```

4.1.2 Contents of classout.txt

Following are the contents of the file `classout.txt` after running GUIDE. The selected tree is marked with two asterisks (**) in the section titled "`Size and CV mean cost and SE of subtrees`" below.

```
Classification tree
Pruning by cross-validation
DSC file: fishdsc.txt
Training sample file: fishdata.txt
Missing value code: NA
Records in data file start on line 2
6 N variables changed to S
D variable is species
Number of records in data file: 159
Length of longest entry in data file: 9
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 7
Training sample class proportions of D variable species:
```

```
    Class  #Cases     Proportion
bream          35     0.22012579
parkki         11     0.06918239
perch          56     0.35220126
pike           17     0.10691824
roach          20     0.12578616
smelt          14     0.08805031
whitefish       6     0.03773585

Summary information for training sample of size 159
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  species    d                                 7
      2  weight     s    5.900        1650.                    2
      3  length1    s    7.500        59.00
      4  length2    s    8.400        63.40
      5  length3    s    8.800        68.00
      6  heightpc   s    14.50        44.50
      7  widthpc    s    8.700        20.90
      8  sex        c                                 2      87

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       159          0          2        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 159
Number of cases excluded due to 0 W or missing D variable: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Simple node models
Estimated priors
Unit misclassification costs
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
```

```
Maximum number of split levels: 15
Minimum node sample size: 2
Ranks of variables and their 1-df chi-squared values at root node
     1  0.1398E+03   heightpc
     2  0.8109E+02   widthpc
     3  0.6776E+02   length3
     4  0.6712E+02   length2
     5  0.6622E+02   length1
     6  0.5275E+02   weight
     7  0.1818E+02   sex

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       11   9.434E-02   2.318E-02   9.847E-03   9.375E-02   2.661E-02
   2       10   9.434E-02   2.318E-02   9.847E-03   9.375E-02   2.661E-02
   3**      8   7.547E-02   2.095E-02   8.597E-03   6.250E-02   7.698E-03
   4        5   1.635E-01   2.933E-02   2.011E-02   1.292E-01   3.768E-02
   5        4   2.390E-01   3.382E-02   3.169E-02   2.188E-01   5.079E-02
   6        3   3.208E-01   3.702E-02   2.194E-02   3.125E-01   3.212E-02
   7        2   4.277E-01   3.924E-02   2.459E-02   3.750E-01   4.935E-02
   8        1   6.478E-01   3.788E-02   2.952E-02   6.562E-01   3.998E-02

0-SE tree based on mean is marked with * and has 8 terminal nodes
0-SE tree based on median is marked with + and has 8 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
       Node    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1      159      159    perch          6.478E-01 heightpc
          2      113      113    perch          5.053E-01 heightpc
          4       31       31    pike           4.544E-01 weight
          8T      14       14    smelt          0.000E+00 -
          9T      17       17    pike           0.000E+00 -
          5       82       82    perch          3.191E-01 weight
         10       60       60    perch          3.850E-01 heightpc
         20T      22       22    perch          4.922E-02 -
         21       38       38    roach          5.023E-01 length2 +length3
         42T      22       22    roach          1.410E-01 length2 +heightpc
```

```
         43T      16       16    perch          0.000E+00 -
         11T      22       22    perch          1.396E-01 sex
          3       46       46    bream          2.425E-01 length2
          6T      12       12    parkki         8.863E-02 -
          7T      34       34    bream          0.000E+00 -

 Number of terminal nodes of final tree: 8
 Total number of nodes of final tree: 15
 Second best split variable (based on curvature test) at root node is widthpc

 Classification tree:

 Node 1: heightpc <= 33.900000
   Node 2: heightpc <= 20.100000
     Node 4: weight <= 109.95000
       Node 8: smelt
     Node 4: weight > 109.95000 or NA
       Node 9: pike
   Node 2: heightpc > 20.100000 or NA
     Node 5: weight <= 452.00000 or NA
       Node 10: heightpc <= 25.250000
         Node 20: perch
       Node 10: heightpc > 25.250000 or NA
         Node 21: -0.92577064 * length3 + length2 <= -0.16303289
           Node 42: roach
         Node 21: -0.92577064 * length3 + length2 > -0.16303289 or NA
           Node 43: perch
     Node 5: weight > 452.00000
       Node 11: perch
 Node 1: heightpc > 33.900000 or NA
   Node 3: length2 <= 26.150000
     Node 6: parkki
   Node 3: length2 > 26.150000 or NA
     Node 7: bream

 ****************************************************************
 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if heightpc <= 33.900000
 heightpc mean = 28.313208
     Class      Number   Posterior
 bream              35  0.2201E+00
 parkki             11  0.6918E-01
 perch              56  0.3522E+00
 pike               17  0.1069E+00
```

```
 roach              20  0.1258E+00
 smelt              14  0.8805E-01
 whitefish           6  0.3774E-01
 Number of training cases misclassified = 103
 Predicted class is perch
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if heightpc <= 20.100000
 heightpc mean = 23.769912
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              56  0.4947E+00
 pike               17  0.1502E+00
 roach              20  0.1767E+00
 smelt              14  0.1237E+00
 whitefish           6  0.5300E-01
 Number of training cases misclassified = 57
 Predicted class is perch
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if weight <= 109.95000
 weight mean = 399.17742
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch               0  0.2201E-02
 pike               17  0.5456E+00
 roach               0  0.7862E-03
 smelt              14  0.4493E+00
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 14
 Predicted class is pike
 ----------------------------
Node 8: Terminal node
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch               0  0.2201E-02
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt              14  0.9943E+00
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 0
 Predicted class is smelt
 ----------------------------
```

```
Node 9: Terminal node
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch               0  0.2201E-02
 pike               17  0.9944E+00
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 0
 Predicted class is pike
 ---------------------------
Node 5: Intermediate node
 A case goes into Node 10 if weight <= 452.00000 or NA
 weight mean = 341.14074
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              56  0.6809E+00
 pike                0  0.6682E-03
 roach              20  0.2432E+00
 smelt               0  0.5503E-03
 whitefish           6  0.7295E-01
 Number of training cases misclassified = 26
 Predicted class is perch
 ---------------------------
Node 10: Intermediate node
 A case goes into Node 20 if heightpc <= 25.250000
 heightpc mean = 26.005000
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              37  0.6150E+00
 pike                0  0.6682E-03
 roach              20  0.3320E+00
 smelt               0  0.5503E-03
 whitefish           3  0.4992E-01
 Number of training cases misclassified = 23
 Predicted class is perch
 ---------------------------
Node 20: Terminal node
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              21  0.9508E+00
 pike                0  0.6682E-03
```

```
 roach               1  0.4596E-01
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 1
 Predicted class is perch
 ---------------------------
Node 21: Intermediate node
 A case goes into Node 42 if -0.92577064 * length3 + length2 <= -0.16303289
 Linear combination mean = -0.31127156
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              16  0.4206E+00
 pike                0  0.6682E-03
 roach              19  0.4977E+00
 smelt               0  0.5503E-03
 whitefish           3  0.7869E-01
 Number of training cases misclassified = 19
 Predicted class is roach
 ---------------------------
Node 42: Terminal node
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch               0  0.2201E-02
 pike                0  0.6682E-03
 roach              19  0.8590E+00
 smelt               0  0.5503E-03
 whitefish           3  0.1357E+00
 Number of training cases misclassified = 3
 Predicted class is roach
 ---------------------------
Node 43: Terminal node
     Class      Number   Posterior
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              16  0.9960E+00
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 0
 Predicted class is perch
 ---------------------------
Node 11: Terminal node
     Class      Number   Posterior
```

```
 bream               0  0.1376E-02
 parkki              0  0.4324E-03
 perch              19  0.8604E+00
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           3  0.1357E+00
 Number of training cases misclassified = 3
 Predicted class is perch
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if length2 <= 26.150000
 length2 mean = 30.056522
     Class      Number   Posterior
 bream              35  0.7575E+00
 parkki             11  0.2381E+00
 perch               0  0.2201E-02
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 11
 Predicted class is bream
 ----------------------------
Node 6: Terminal node
     Class      Number   Posterior
 bream               1  0.8419E-01
 parkki             11  0.9114E+00
 perch               0  0.2201E-02
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 1
 Predicted class is parkki
 ----------------------------
Node 7: Terminal node
     Class      Number   Posterior
 bream              34  0.9951E+00
 parkki              0  0.4324E-03
 perch               0  0.2201E-02
 pike                0  0.6682E-03
 roach               0  0.7862E-03
 smelt               0  0.5503E-03
 whitefish           0  0.2358E-03
 Number of training cases misclassified = 0
```

```
Predicted class is bream
---------------------------

Classification matrix for training sample:
Predicted      True class
class          bream    parkki     perch      pike     roach     smelt
bream             34         0         0         0         0         0
parkki             1        11         0         0         0         0
perch              0         0        56         0         1         0
pike               0         0         0        17         0         0
roach              0         0         0         0        19         0
smelt              0         0         0         0         0        14
whitefish          0         0         0         0         0         0
Total             35        11        56        17        20        14

          whitefish
bream             0
parkki            0
perch             3
pike              0
roach             3
smelt             0
whitefish         0
Total             6

Number of cases used for tree construction: 159
Number misclassified: 8
Resubstitution estimate of mean misclassification cost: 0.50314465E-01

Observed and fitted values are stored in classfit.txt
LaTeX code for tree is in class.tex
```

The second last paragraph of the output file shows that 8 out of 159 observations are misclassified. The LaTeX tree diagram in Figure 1 shows that:

1. The predicted specias and sample size are printed beneath each terminal node.
2. The highest posterior class probaility is printed beside each terminal node. For example, nodes 7, 8, 9, and 43 have highest posterior probability 1.00, indicating that they are pure nodes (nodes containing only one species).
3. Node 5 is split on `weight` $\leq_*$ 452, with the symbol $\leq_*$ indicating that observations with missing values in `weight` go to the left branch (node 10).
4. Node 21 is a linear split on variables `length2` and `length3` and is indicated by a blue circle.

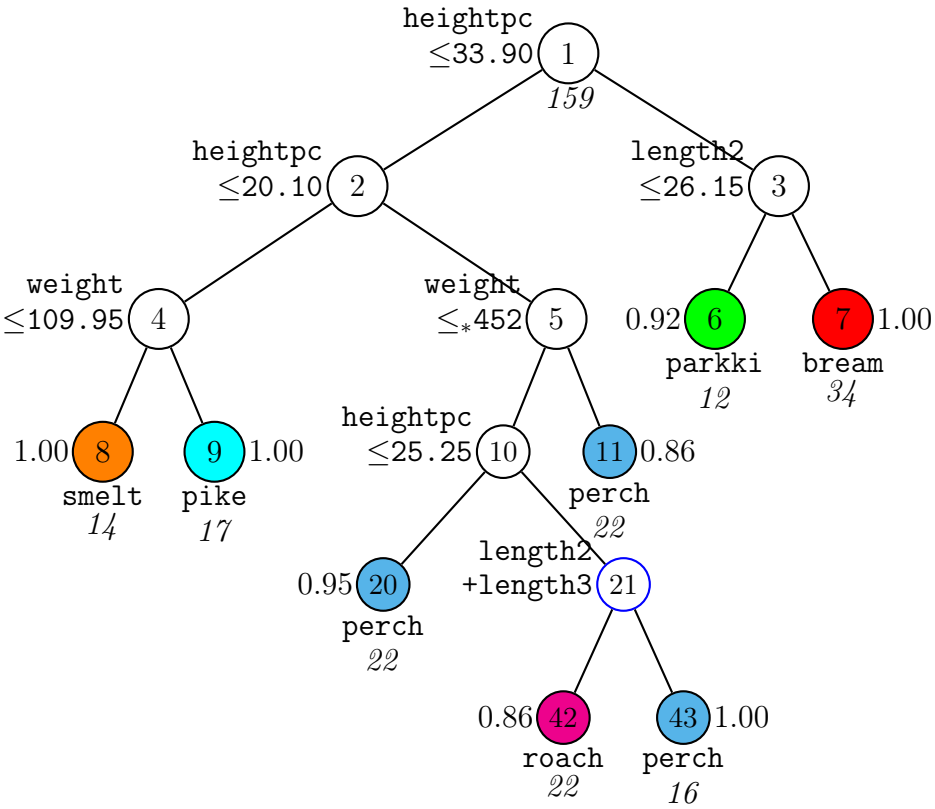

Figure 1: GUIDE v.46.1 0.250-SE classification tree for predicting `species` using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Splits at nodes drawn with **blue** circles are linear splits. Predicted class and total sample size (in *italics*) printed below each terminal node; highest class posterior probability beside node. Second best split variable at root node is `widthpc`.

4.1.3 Contents of classfit.txt

Following are the top five lines of the file `classfit.txt`. They show the classification results of the first 4 observations. The column "`train`" shows whether an observation is used to fit the model (`y`, for yes) or not (`n`, for no). In this example, the column contains only `y` because all 159 observations are used in the fitting. The second column labeled "`node`" gives the terminal node number. In this example the first observation landed in node 6 while the other three observations went to node 7. The columns labeled "`observed`" and "`predicted`" give the true and predicted class of each observation. The remaining columns give the posterior probabilities of each observation in the seven classes.

train	node	observed	predicted	"P(bream)"	"P(parkki)"	"P(perch)"	"P(pike)"	"P(roach)"	"P(smelt)"	"P(whitefish)"
y	6	"bream"	"parkki"	0.84188E-01	0.91137E+00	0.22013E-02	0.66824E-03	0.78616E-03	0.55031E-03	0.23585E-03
y	7	"bream"	"bream"	0.99513E+00	0.43239E-03	0.22013E-02	0.66824E-03	0.78616E-03	0.55031E-03	0.23585E-03
y	7	"bream"	"bream"	0.99513E+00	0.43239E-03	0.22013E-02	0.66824E-03	0.78616E-03	0.55031E-03	0.23585E-03
y	7	"bream"	"bream"	0.99513E+00	0.43239E-03	0.22013E-02	0.66824E-03	0.78616E-03	0.55031E-03	0.23585E-03

The posterior probabilities are calculated as follows. Let J be the number of classes, N_j be the number of class j observations in the whole sample and $N = \sum_j N_j$. Let π_j be the (estimated or specified) prior probability of class j. Let $n_j(t)$ be the number of class j training samples in node t. The posterior probability of class j in t is $p_j(t) = \pi_j n_j(t) N_j^{-1} / \sum_i \pi_i n_i(t) N_i^{-1}$. If $\min_j p_j(t) = 0$, the posterior probability is redefined to be $(N p_j(t) + \pi_j)/(N+1)$; this ensures that no probability is zero if all π_j are positive.

4.1.4 Contents of class.r

Following are the contents of `class.r` giving an R prediction function.

```
predicted <- function(){
 if(!is.na(heightpc) & heightpc <= 33.9000000000 ){
   if(!is.na(heightpc) & heightpc <= 20.1000000000 ){
     if(!is.na(weight) & weight <= 109.950000000 ){
       nodeid <- 8
       predclass <- "smelt"
       posterior <- c( 0.13758E-02, 0.43239E-03, 0.22013E-02, 0.66824E-03, 0.78616E-03, 0.99430E+00, 0
     } else {
       nodeid <- 9
       predclass <- "pike"
       posterior <- c( 0.13758E-02, 0.43239E-03, 0.22013E-02, 0.99442E+00, 0.78616E-03, 0.55031E-03, 0
     }
   } else {
     if(is.na(weight) | weight <= 452.000000000 ){
       if(!is.na(heightpc) & heightpc <= 25.2500000000 ){
         nodeid <- 20
         predclass <- "perch"
         posterior <- c( 0.13758E-02, 0.43239E-03, 0.95078E+00, 0.66824E-03, 0.45957E-01, 0.55031E-03,
       } else {
         if(!is.na(length2) & !is.na(length3) &  -0.925770636221  * length3 + length2 <= -0.16303288757
           nodeid <- 42
           predclass <- "roach"
```

```
23            posterior <- c( 0.13758E-02, 0.43239E-03, 0.22013E-02, 0.66824E-03, 0.85902E+00, 0.55031E-03
24          } else {
25            nodeid <- 43
26            predclass <- "perch"
27            posterior <- c( 0.13758E-02, 0.43239E-03, 0.99595E+00, 0.66824E-03, 0.78616E-03, 0.55031E-03
28          }
29        }
30      } else {
31        nodeid <- 11
32        predclass <- "perch"
33        posterior <- c( 0.13758E-02, 0.43239E-03, 0.86044E+00, 0.66824E-03, 0.78616E-03, 0.55031E-03, 0
34      }
35    }
36  } else {
37    if(!is.na(length2) & length2 <= 26.1500000000 ){
38      nodeid <- 6
39      predclass <- "parkki"
40      posterior <- c( 0.84188E-01, 0.91137E+00, 0.22013E-02, 0.66824E-03, 0.78616E-03, 0.55031E-03, 0.23
41    } else {
42      nodeid <- 7
43      predclass <- "bream"
44      posterior <- c( 0.99513E+00, 0.43239E-03, 0.22013E-02, 0.66824E-03, 0.78616E-03, 0.55031E-03, 0.23
45    }
46  }
47  return(c(nodeid,predclass,posterior))
48 }
49 ## end of function
50 ##
51 ##
52 ## If desired, replace "fishdata.txt" with name of file containing new data
53 ## New file must have at least the same variables with same names
54 ## (but not necessarily the same order) as in the training data file
55 ## Missing value code is converted to NA if not already NA
56 newdata <- read.table("fishdata.txt",header=TRUE,colClasses="character")
57 ## node contains terminal node ID of each case
58 ## pred.class contains predicted class
59 ## prob contains predicted posterior probabilities
60 node <- NULL
61 pred.class <- NULL
62 prob <- NULL
63 for(i in 1:nrow(newdata)){
64     weight <- as.numeric(newdata$weight[i])
65     length2 <- as.numeric(newdata$length2[i])
66     length3 <- as.numeric(newdata$length3[i])
67     heightpc <- as.numeric(newdata$heightpc[i])
68     tmp <- predicted()
69     node <- c(node,as.numeric(tmp[1]))
70     pred.class <- rbind(pred.class,tmp[2])
71     prob <- rbind(prob,as.numeric(tmp[-c(1,2)]))
72 }
```

Figure 2 shows the partitions in the nodes of the tree. The R code for producing the plots follows.

```
1 ancestor <- function(inter,leaf){
2   n <- leaf
3   while(n > inter){ n <- floor(n/2) }
```

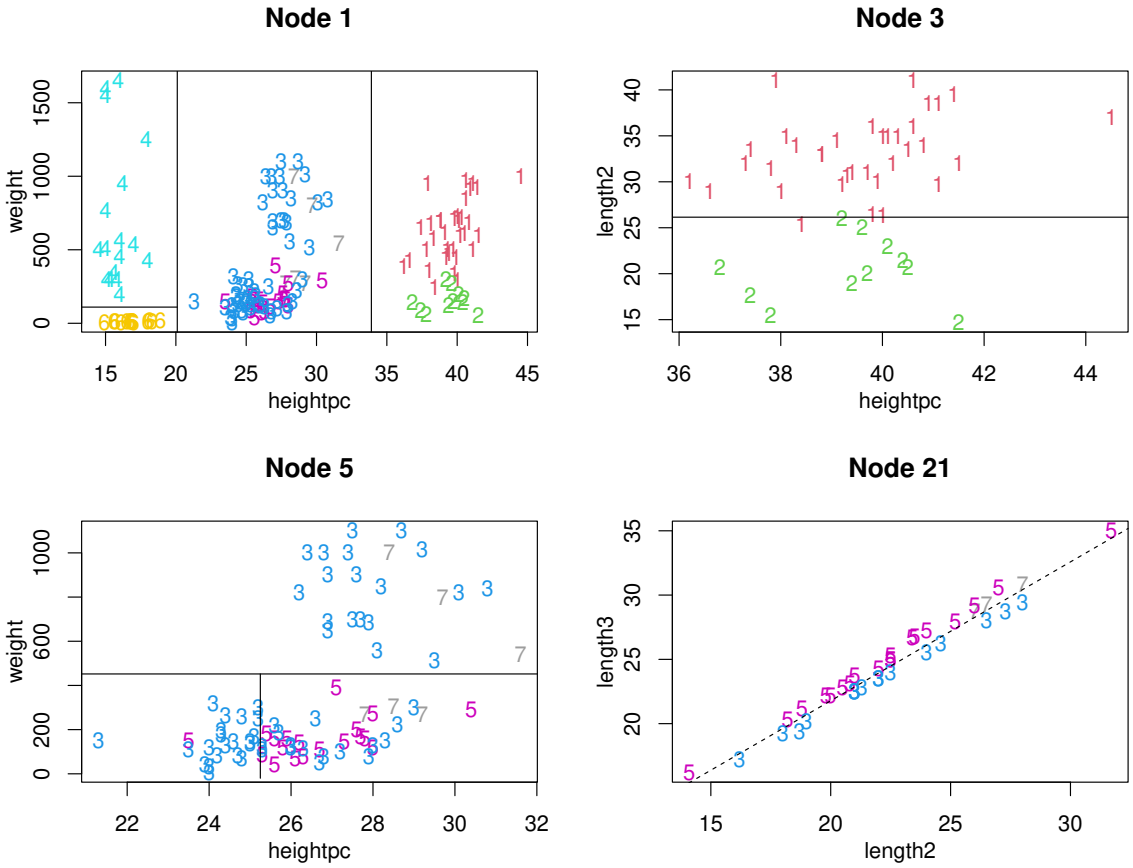

Figure 2: Data in nodes of Figure 1. 1 = Bream, 2 = Parkki, 3 = Perch, 4 = Pike, 5 = Roach, 6 = Smelt, 7 = Whitefish.

```
4   if(n == inter) TRUE else FALSE
5 }
6 i.node <- function(num,termnodes){
7   n <- length(termnodes)
8   gp <- rep(FALSE,n)
9   for(i in 1:n){ gp[i] <- ancestor(num,termnodes[i]) }
10  return(gp)
11 }
12 par(mfrow=c(2,2),mar=c(3,4,4,1),mgp=c(2,1,0),cex=1.2)
13 z <- read.table("fishdata.txt",header=TRUE)
14 var.names <- names(z)
15 z$species <- as.factor(z$species)
16 class.sym <- as.numeric(z$species)
17 class.col <- class.sym+1
18 class <- z$species
19 zf <- read.table("classfit.txt",header=TRUE)
20 node <- zf$node
21 v1.id <- 6
22 v2.id <- 2
23 x <- z[,v1.id]
24 y <- z[,v2.id]
25 gp <- rep(TRUE,length(class))
26 plot(x[gp],y[gp],xlab=var.names[v1.id],ylab=var.names[v2.id],type="n")
27 text(x[gp],y[gp],labels=class.sym[gp],col=class.col[gp])
28 abline(v=c(20.1,33.9))
29 lines(c(10,20.1),rep(109.95,2))
30 title("Node␣1")
31 nodeid <- 3
32 gp <- i.node(nodeid,node)
33 v1.id <- 6
34 v2.id <- 4
35 x <- z[,v1.id]
36 y <- z[,v2.id]
37 plot(x[gp],y[gp],xlab=var.names[v1.id],ylab=var.names[v2.id],type="n")
38 text(x[gp],y[gp],labels=class.sym[gp],col=class.col[gp])
39 abline(h=26.15)
40 title("Node␣3")
41 nodeid <- 5
42 gp <- i.node(nodeid,node)
43 v1.id <- 6
44 v2.id <- 2
45 x <- z[,v1.id]
46 y <- z[,v2.id]
47 plot(x[gp],y[gp],xlab=var.names[v1.id],ylab=var.names[v2.id],type="n")
48 text(x[gp],y[gp],labels=class.sym[gp],col=class.col[gp])
49 abline(h=452)
```

```
lines(rep(25.25,2),c(-20,452))
title("Node␣5")
nodeid <- 21
v1.id <- 4
v2.id <- 5
x <- z[,v1.id]
y <- z[,v2.id]
gp <- i.node(nodeid,node)
plot(x[gp],y[gp],xlab=var.names[v1.id],ylab=var.names[v2.id],type="n")
text(x[gp],y[gp],labels=class.sym[gp],col=class.col[gp])
a <- 0.16303289/0.92577064
b <- 1/0.92577064
abline(c(a,b),lty=2)
title("Node␣21")
```

4.2 Linear splits and discriminant models

This section shows how to construct a tree that uses linear splits and quadratic discriminant analysis (QDA) models in the nodes. As before, the default choice at each prompt (indicated by <cr>) can be accepted by simply hitting the "carriage return" key.

4.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: qda.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: qda.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
 Input 1 for simple, 2 for nearest-neighbor, 3 for kernel, 4 for lda models,
       5 for qda models ([1:5], <cr>=4): 4
 Input 0 to specify tree with fixed no. of nodes, 1 to prune by CV,
       2 by test sample, 3 for no pruning ([0:3], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: fishdsc.txt
 Reading DSC file ...
 Training sample file: fishdata.txt
 Missing value code: NA
```

```
Records in data file start on line 2
D variable is species
C variables changed to B
Reading data file ...
Number of records in data file: 159
Length of longest entry in data file: 9
Checking for missing values ...
Finished checking
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 7
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
    Class  #Cases     Proportion
bream          35     0.22012579
parkki         11     0.06918239
perch          56     0.35220126
pike           17     0.10691824
roach          20     0.12578616
smelt          14     0.08805031
whitefish       6     0.03773585
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       159          0          2        0        6        3        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        1        0        0
Number of dummy variables (excl. missing-value indicators) created: 2
Number of missing-value indicators created: 1
Number of cases used for training: 159
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Default number of cross-validations:           10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.25):
```

```
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Default maximum number of non-categorical splits at each node is 100
Choose 1 to accept the default, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Default max. number of split levels: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 5
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): qda.tex
Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram:
Input 0 for highest posterior, 1 for sample sizes, 2 for sample proportions,
      3 for posterior probs, 4 for misclassification cost, 5 for nothing
Input your choice ([0:5], <cr>=4):
You can store discriminant coefficients in a file
Choose 1 to skip this step, 2 to store the coefficients
Input your choice ([1:2], <cr>=2):
Input file name: qda.coefs
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: qda.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: qda.r
You can output R code for plotting data in the nodes
Input 1 to output R code
Input 2 to skip this step
Input your choice ([1:2], <cr>=1):
Input filename to storing plot data: qdaplots.r
Input file name for storing data for plotting: qdadata.txt
Input file is created!
Run GUIDE with the command: guide < qda.in
```

4.2.2 Contents of qda.out

```
Classification tree
Linear discriminant splits and QDA node models
Pruning by cross-validation
DSC file: fishdsc.txt
Training sample file: fishdata.txt
```

```
Missing value code: NA
Records in data file start on line 2
D variable is species
C variables changed to B
Number of records in data file: 159
Length of longest entry in data file: 9
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 7
Training sample class proportions of D variable species:
    Class  #Cases     Proportion
bream          35     0.22012579
parkki         11     0.06918239
perch          56     0.35220126
pike           17     0.10691824
roach          20     0.12578616
smelt          14     0.08805031
whitefish       6     0.03773585

Summary information for training sample of size 159
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                    #Codes/
                                                    Levels/
 Column  Name                Minimum      Maximum    Periods   #Missing
      1  species        d                                 7
      2  weight         n   5.9000E+00   1650.                     2
      3  length1        n    7.500       59.00
      4  length2        n    8.400       63.40
      5  length3        n    8.800       68.00
      6  heightpc       n    14.50       44.50
      7  widthpc        n    8.700       20.90
      8  sex            b                                2      87
  ================== Constructed variables ==================
      9  sex.female     f    0.000       1.000
     10  sex.male       f    0.000       1.000
     11  weight.NA      f    0.000       1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       159          0          2        0        6        3        0
    #P-var   #M-var   #B-var   #C-var   #I-var
```

```
         0        0        1        0        0
Number of dummy variables (excl. missing-value indicators) created: 2
Number of missing-value indicators created: 1
Number of cases used for training: 159
Number of cases excluded due to 0 W or missing D variable: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Estimated priors
Unit misclassification costs
Max number of splits: 100
Maximum number of split levels: 15
Minimum node sample size: 5
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   0        5   1.069E-01   2.451E-02   1.277E-02   1.250E-01   2.387E-02
   1        4   1.069E-01   2.451E-02   1.277E-02   1.250E-01   2.387E-02
   2        3   1.069E-01   2.451E-02   1.277E-02   1.250E-01   2.387E-02
   3**      2   8.805E-02   2.247E-02   1.352E-02   6.250E-02   1.735E-02
   4        1   1.824E-01   3.062E-02   2.438E-02   1.604E-01   3.663E-02

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
       Node    Total  Misclassification
      label    cases  cost
          1      159  1.384E-01
          2T      68  0.000E+00
          3T      91  2.198E-02

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3

Classification tree with linear discriminant splits and QDA node models
```

```
 Node 1: D1 <= 0.17486273
   Node 2: 0/68 misclassified
 Node 1: D1 > 0.17486273
   Node 3: 2/91 misclassified

 ********************************************************************
 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if D1 <= 0.17486273 where the coefficients of D1
 are the coefficients of the 1st crimcoord below:
 Variable        Crimcoord1 coef  Crimcoord2 coef   Variable mean
 weight          -0.10646693E-04  -0.20019881E-03    401.23503
 length1         -0.41494497E-02  -0.60131669E-01    26.247170
 length2          0.37732915E-02   0.68470458E-01    28.415723
 heightpc         0.10100866E-01   0.30955883E-03    28.313208
 widthpc         -0.54354328E-02  -0.10493626E-01    14.116352
 sex.female      -0.14854441E-02  -0.30499910E-01   0.34591195
 sex.male         0.64111909E-02  -0.12073413E-01   0.10691824

 Class          Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 bream       0.315985        0.203253E-03    0.179849        0.381899E-03     0.22013
 parkki      0.319137        0.176877E-03    0.859815E-01    0.594026E-03     0.06918
 perch       0.173082        0.198006E-03    0.113651        0.155527E-02     0.35220
 pike        0.909419E-01    0.128065E-03    0.301851        0.107943E-02     0.10692
 roach       0.186620        0.185920E-03    0.921213E-01    0.121835E-02     0.12579
 smelt       0.114490        0.608590E-04    0.111652E-01    0.304533E-03     0.08805
 whitefish   0.201284        0.552147E-04    0.143276        0.753174E-03     0.03774

 Class        Correlation
 bream      -0.182529
 parkki     -0.363430
 perch       0.641609E-01
 pike       -0.204668
 roach       0.206973
 smelt       0.157189
 whitefish   0.162032E-01
 ----------------------------
Node 2: Terminal node
 Cases classified by QDA on 1st and 2nd crimcoords with coefs below:
 Variable        Crimcoord1 coef  Crimcoord2 coef   Variable mean
 weight          -0.32521310E-03   0.78169131E-04    372.28529
 length1         -0.10887904      -0.21649848E-01    26.614706
 length2          0.79213134E-01   0.39149660E-01    28.636765
 length3          0.36084671E-01  -0.22898231E-01    30.607353
 heightpc         0.97600350E-02   0.19359325E-01    21.158824
```

```
 widthpc          0.27050333E-02   0.31862627E-02    13.211765

 Class          Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 perch       0.638530        0.284798E-02    0.426285        0.480424E-03     0.51369
 pike        0.685068        0.888589E-03    0.141566        0.825493E-03     0.24911
 roach       0.630274        0.364608E-04    0.388643        0.759983E-03     0.03001
 smelt       0.377875        0.171544E-02    0.284848        0.612044E-03     0.20515

 Class        Correlation
 perch      -0.294882
 pike        0.173587
 roach      -0.857166
 smelt       0.714333
 ----------------------------
Node 3: Terminal node
 Cases classified by QDA on 1st and 2nd crimcoords with coefs below:
 Variable        Crimcoord1 coef  Crimcoord2 coef   Variable mean
 weight          -0.32519883E-04   0.21423970E-03    423.35393
 length1          0.12447415E-01   0.22990857        25.972527
 length2          0.68660090E-01  -0.27853334        28.250549
 length3         -0.74942028E-01   0.41636881E-01    31.690110
 widthpc          0.41454782E-02  -0.93710673E-02    14.792308

 Class          Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 bream      -0.185470        0.345736E-03   -0.656897        0.613959E-02     0.38359
 parkki     -0.246289E-01    0.172909E-03   -0.511178        0.341214E-02     0.12056
 perch       0.739426E-01    0.472161E-03   -0.701695        0.675896E-02     0.23153
 roach      -0.313040E-01    0.567822E-03   -0.519797        0.460589E-02     0.19735
 whitefish  -0.144249E-01    0.752335E-04   -0.707769        0.350085E-03     0.06576

 Class        Correlation
 bream      -0.253159E-01
 parkki      0.951333
 perch      -0.243683
 roach       0.122187
 whitefish  -0.870033
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          bream    parkki     perch      pike     roach     smelt
 bream             35         0         0         0         0         0
 parkki             0        11         0         0         2         0
 perch              0         0        56         0         0         0
 pike               0         0         0        17         0         0
 roach              0         0         0         0        18         0
```

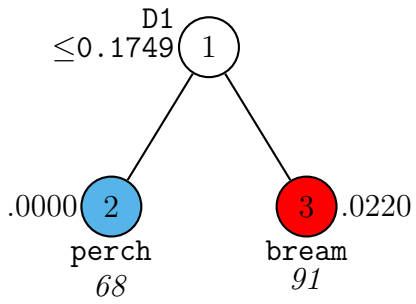

Figure 3: GUIDE v.46.1 0.250-SE classification tree for predicting `species` using linear discriminant splits and quadratic discriminant node models. At each split, an observation goes to the left branch if and only if the condition is satisfied. Class minimizing misclassification cost and sample size (in *italics*) printed below each terminal node; mean misclassification cost beside node.

```
smelt              0        0        0        0        0       14
whitefish          0        0        0        0        0        0
Total             35       11       56       17       20       14

          whitefish
bream             0
parkki            0
perch             0
pike              0
roach             0
smelt             0
whitefish         6
Total             6

Number of cases used for tree construction: 159
Number misclassified: 2
Resubstitution estimate of mean misclassification cost: 0.12578616E-01

Observed and fitted values are stored in qda.fit
LaTeX code for tree is in qda.tex
R code is stored in qda.r
Discriminant coefficients are stored in qda.coefs
R function for plotting data are stored in qdaplots.r
Plot data are stored in qdadata.txt
```

Figure 3 shows that the tree has 2 terminal nodes. The results in `qda.out` show that it misclassifies only 2 observations. Figure 4 plots the data and splits in the nodes of the tree. They are produced by the R code in `qdaplots.r` whose contents are below.

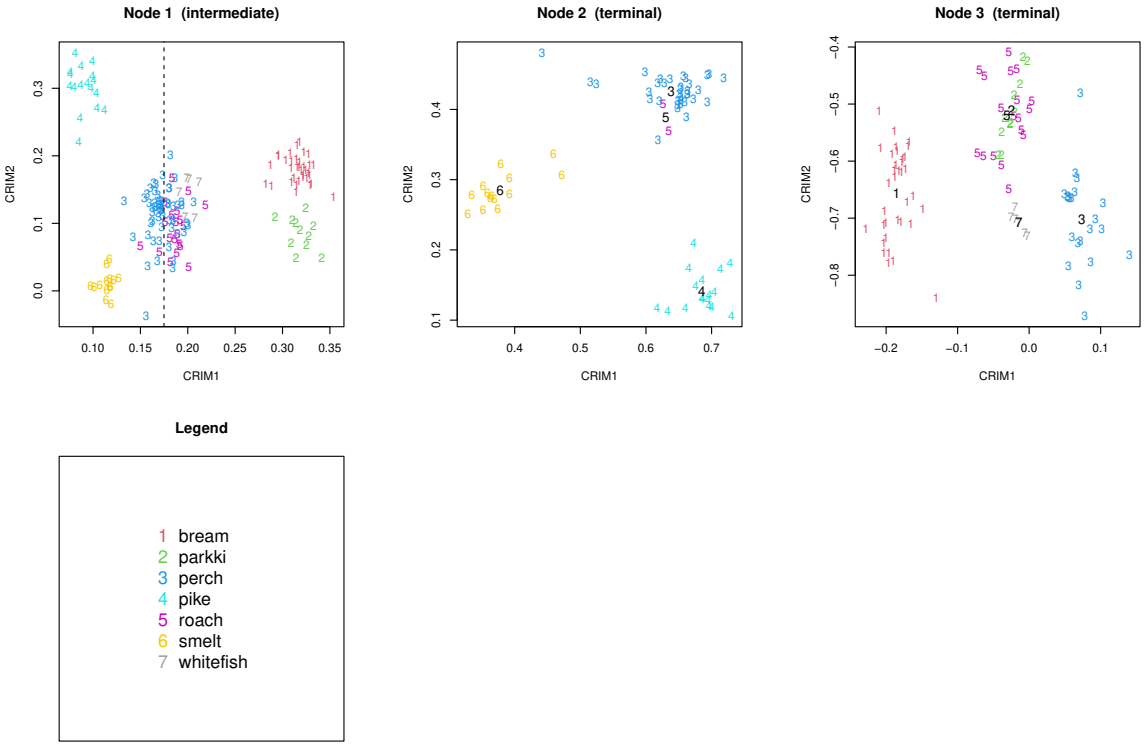

Figure 4: Data and splits in nodes of Figure 3. Class means are indicated by numbers in black.

```
library(MASS)
descendents <- function(node){
    desc <- NULL
    for(m in tnodes){
        j <- m
        while(j > 1){
            j <- floor(j/2)
            if(j == node) desc <- c(desc,m)
        }
    }
    return(desc)
}
pdf(file='qdaplots.pdf',width=8,height=6)
## postscript(file='qdaplots.ps',horizontal=TRUE)
par(pty='s',mfrow=c(3,4),cex=0.5,cex.main=0.8,cex.lab=0.6,cex.axis=0.6)
eqplot <- FALSE
splits <- c( 0.220278, 0.129039, 0.516425, 0.554004E-1)
varnames <- scan(file='qdadata.txt',what='character',nlines=1)
nvar <- length(varnames)
z <- as.data.frame(matrix(scan(file='qdadata.txt',skip=1),ncol=nvar,byrow=TRUE))
colnames(z) <- varnames
y.id <- 1 # class variable column
z[,y.id] <- as.factor(z[,y.id])
nonmissing <- !is.na(z[,y.id])
z <- z[nonmissing,]
class.codes <- unclass(z[,y.id])
uniq.codes <- unique(sort(class.codes))
pred <- read.table(file='qda.fit',header=TRUE)
pred <- pred[nonmissing,]
tnodes <- unique(sort(pred$node))  ## terminal nodes
coefs <- read.table(file='qda.coefs',header=TRUE)
inodes <- unique(sort(coefs$node))
inodes <- setdiff(inodes,tnodes)   ## intermediate nodes
allnodes <- sort(c(inodes,tnodes))
k <- 0
for(node in allnodes){
    goodplot <- TRUE
    if(node %in% inodes){ ## intermediate node
        terminal <- FALSE
        d <- descendents(node)
        if(length(d) > 0){
            gp <- pred$node %in% d
            k <- k+1
        } else {
            goodplot <- FALSE
        }
```

```
47      } else { ## terminal node
48          terminal <- TRUE
49          gp <- pred$node == node
50          tmp <- unique(sort(z[gp,y.id]))
51          if(length(tmp) <= 1) goodplot <- FALSE
52      }
53      if(goodplot){
54          vgp <- coefs$node == node
55          xvars <- coefs$variable[vgp]
56          xmeans <- as.numeric(coefs$variablemean[vgp])
57          subdatx <- z[gp,]
58          subdatx <- subdatx[,which(names(subdatx) %in% xvars)]
59          subdaty <- z[gp,y.id]
60          for(j in 1:ncol(subdatx)){ ## impute missing values
61              tmp <- subdatx[,j]
62              tmp[is.na(tmp)] <- xmeans[j]
63              subdatx[,j] <- tmp
64          }
65          crimc <- as.matrix(coefs[vgp,3:4])
66          xmat <- as.matrix(subdatx)
67          crimx <- xmat %*% crimc
68          if(eqplot){ ## find variances
69              v1 <- 0
70              v2 <- 0
71              n <- 0
72              for(i in uniq.codes){
73                  cgp <- subdaty == i
74                  nc <- sum(cgp)
75                  if(nc > 1){
76                      n <- n+nc
77                      v1 <- v1+(nc-1)*var(crimx[cgp,1])
78                      v2 <- v2+(nc-1)*var(crimx[cgp,2])
79                  }
80              }
81              v1 <- v1/n
82              v2 <- v2/n
83              crimx[,1] <- crimx[,1]/sqrt(v1)
84              crimx[,2] <- crimx[,2]/sqrt(v2)
85              eqscplot(crimx[,1],crimx[,2],type='n',
86                  xlab='scaled␣CRIM1',ylab='scaled␣CRIM2')
87          } else {
88              plot(crimx[,1],crimx[,2],type='n',xlab='CRIM1',ylab='CRIM2')
89          }
90          text(crimx[,1],crimx[,2],labels=subdaty,col=1+class.codes[gp])
91          if(terminal){
92              for(i in uniq.codes){ # add class means
```

```
93                  cgp <- subdaty == i
94                  m1 <- mean(crimx[cgp,1])
95                  m2 <- mean(crimx[cgp,2])
96                  text(m1,m2,labels=as.character(i),cex=1.1)
97              }
98              if(length(allnodes) == 1){ # only 1 node
99                  if(eqplot){
100                     abline(v=splits/sqrt(v1),lty=2)
101                 } else {
102                     abline(v=splits,lty=2)
103                 }
104             }
105             title(paste('Node',node,'␣(terminal)'))
106         } else {
107             if(eqplot){
108                 abline(v=splits[k]/sqrt(v1),lty=2)
109             } else {
110                 abline(v=splits[k],lty=2)
111             }
112             title(paste('Node',node,'␣(intermediate)'))
113         }
114     }
115 }
116 plot(c(-1,1),c(-1,1),type='n',xaxt='n',yaxt='n',xlab='',ylab='')
117 leg.txt <- unique(sort(pred$observed))
118 leg.pch <- as.character(uniq.codes)
119 leg.col <- unique(sort(class.codes))+1
120 legend('center',legend=leg.txt,col=leg.col,pch=leg.pch,
121 ncol=1,bty='n',cex=1.2)
122 title('Legend')
123 dev.off()
```

5 Missing-value flag variables

The Consumer Expenditure (CE) Survey is carried out by the Census Bureau for the Bureau of Labor Statistics (BLS). Conducted quarterly, the survey is a rotating panel survey that collects data on expenditures, income, and demographic characteristics of a sample of about 6000 consumer units (CUs) in the United States. After a CU is in the survey for four quarters, it is dropped and a new unit selected to replace it. The BLS defines *CU* and *reference person* of the CU as follows.

1. A CU consists of any of the following:
   (a) All members of a particular household who are related by blood, marriage, adoption, or other legal arrangements.
   (b) A person living alone or sharing a household with others or living as a roomer in a private home or lodging house or in permanent living quarters in a hotel or motel, but who is financially independent.
   (c) Two or more persons living together who use their incomes to make joint expenditure decisions. Financial independence is determined by spending behavior with regard to the three major expense categories: housing, food, and other living expenses. To be considered financially independent, the respondent must provide at least two of the three major expenditure categories, either entirely or in part.
2. A reference person of the CU is the first member mentioned by the respondent when asked "What are the names of all the persons living or staying here? Start with the name of the person or one of the persons who owns or rents the home." It is with respect to this person that the relationship of the other CU members is determined.

The data in the file `ce2021.txt` consist of 3965 observations on 550 variables. They are extracted from the second, third and fourth quarters of 2021 and the first quarter of 2022 of the Interview part of the CE survey. For the purpose of illustration and because it is not possible to link CUs between quarters, each CU in the sample is treated as unique. Table 11 in the Appendix gives the names and definitions of some of the 550 variables and their missing-value rates.

About 20% of the variables are *missing-value flags* that give the reasons for missing values. Table 4 lists the flag codes. A variable takes value `NA` (nonresponse) if its flag variable code is `A`, `B`, or `C`. The names of flag variables are typically the same as their parents, except for the addition or substitution of an underscore. For

Table 4: Codes for missing-value flag variables

A	Valid nonresponse; a response is not anticipated
B	Invalid nonresponse; nonresponse inconsistent with other data reported by CU
C	"Don't know", refusal, or other type of nonresponse
D	Valid unadjusted data value
T	Valid value topcoded or suppressed

example, INTRDVX_ is the flag variable for INTRDVX (amount of income received from interest and dividends). In this dataset, INTRDVX_ has no B codes and records with A codes are removed. Thirty-seven percent of the records (1478) have INTRDVX_ = C.

A T flag code indicates that the value of a variable is "top-coded." Top-coding is a method used by the BLS to protect the privacy of the respondents in the top 3 percent of the data. Usually, the reported values of the CUs in this group are replaced by their group mean. For example, below are the values of AGE2 (age of spouse) and AGE2_ in rows 112–117 of the data:

```
    AGE2 AGE2_
112   29    D
113   87    T
114   NA    A
115   57    D
116   87    T
117   NA    A
```

Respondents 113 and 116 are topcoded and have their values equal to 87, the mean of the top 3 percent of AGE2. See https://www.bls.gov/cex/pumd_doc.htm for names of all the variables and Loh et al. (2019b, 2020) for an analysis of an earlier dataset.

Variable FINLWT21 is a sampling weight. For classification, GUIDE treats all observations with positive sampling weight equally; observations with non-positive weights are ignored in tree construction.

Missing-value flag variables are indicated by the letters "m" or "M" in the DSC file. To indicate to GUIDE to which variable is associated with which M variable, the latter must follow immediately after a B, C, N, P, or S variable in the DSC file. For example, the following lines from the file ce2021class.dsc indicate that DIRACC_ is the flag variable for DIRACC, AGE_REF_ is the flag variable for AGE_REF, and INCN_NW1 is the flag variable for INCNONW1.

```
ce2021.txt
```

```
NA
2
1 DIRACC n
2 DIRACC_ m
3 AGE_REF n
4 AGE_REF_ m
5 AGE2 n
6 AGE2_ m
7 AS_COMP1 n
8 AS_COMP2 n
9 AS_COMP3 n
10 AS_COMP4 n
11 AS_COMP5 n
12 BATHRMQ n
13 BATHRMQ_ m
14 BEDROOMQ n
15 BEDR_OMQ m
16 BLS_URBN n
17 BUILDING c
18 CUTENURE c
19 EARNCOMP c
20 EDUC_REF n
21 EDUCA2 n
22 EDUCA2_ m
:
406 INTRDVX x
407 INTRDVX_ d
408 IRAB n
409 IRAB_ m
:
547 WHLFYR c
548 WHLFYR_ m
549 FFTAXOWE n
550 FSTAXOWE n
```

A split on an `N`, `P`, or `S` variable that has an associated missing-value flag variable can take several forms. For example, a split on `RETSURVX` (retirement, survivor, or disability pensions in past 12 months) with flag variable `RETS_RVX` (which takes values `A`, `C`, `D`, and `T`) can take 7 forms:

1. RETS_RVX = A (only A flag values go left)
2. RETS_RVX = C (only C flag values go left)
3. RETSURVX = NA (all missing values go left)
4. RETSURVX $\leq c$
5. RETSURVX $\leq_* c$ (the symbol "$\leq_*$" means "$\leq$ or is missing")
6. RETSURVX $\leq c$ or RETS_RVX = A
7. RETSURVX $\leq c$ or RETS_RVX = C

Similarly, a split on a C variable such as INCNONW2 that has missing-value flag variable INCN_NW2 can take these forms (see Figure 12):

1. INCNONW2 in S (where S is a subset of values of INCNONW2)
2. INCNONW2 = NA
3. INCNONW2 in S or INCN_NW2 in S^* (where S^* is a subset of flag codes)

5.1 Classification: univariate splits

This section shows how to construct a classification tree for predicting INTRDVX_ using the DSC file ce2021class.dsc.

5.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: classin.txt
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: classout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021class.dsc
Reading DSC file ...
```

```
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 3965
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Warning: No interaction and linear splits; too many predictor variables
```

```
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Warning: All positive weights treated as 1
Default max. number of split levels: 12
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): class.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: classfit.txt
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: class.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < classin.txt
```

5.1.2 Contents of output file

```
Classification tree
Pruning by cross-validation
DSC file: ce2021class.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 3
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1478     0.37276166
```

```
D        2431     0.61311475
T          56     0.01412358

Summary information for training sample of size 3965
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     s    1.000        1.000                 170  constant
      2  DIRACC_    m                                2
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                0
      :
     31  FINLWT21   w    1072.       0.9390E+05
      :
    407  INTRDVX_   d                                3
      :
    549  FFTAXOWE   s  -0.3368E+05   0.3997E+06
    550  FSTAXOWE   s   -3309.       0.7223E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      378
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of constant S variables: 5
Number of cases used for training: 3965
Number of cases excluded due to 0 W or missing D variable: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Warning: No interaction and linear splits; too many predictor variables
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
No interaction splits
No linear splits
```

```
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 3
Ranks of variables and their 1-df chi-squared values at root node
     1  0.2336E+03   INCLASS2
     2  0.1936E+03   STATE
     3  0.1519E+03   ERANKH
     4  0.1350E+03   PSU
     5  0.1307E+03   RETSURVX
     :
   420  0.2626E-02   OTHLOAN
   421  0.2542E-02   VFUELOIC
   422  0.6293E-03   TEXTILCQ

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       69   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   2       68   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   3       67   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   4       66   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   5       65   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   6       64   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   7       63   3.021E-01   7.292E-03   7.727E-03   3.030E-01   1.124E-02
   8*      60   3.021E-01   7.292E-03   7.704E-03   3.030E-01   1.139E-02
   9       53   3.037E-01   7.303E-03   7.308E-03   3.030E-01   9.744E-03
  10       52   3.047E-01   7.309E-03   7.311E-03   3.030E-01   1.051E-02
  11       50   3.042E-01   7.306E-03   7.383E-03   3.030E-01   1.117E-02
  12       48   3.052E-01   7.313E-03   7.977E-03   3.018E-01   1.223E-02
  13       34   3.052E-01   7.313E-03   8.202E-03   3.018E-01   1.287E-02
  14       25   3.054E-01   7.315E-03   8.156E-03   3.018E-01   1.253E-02
  15       24   3.054E-01   7.315E-03   8.156E-03   3.018E-01   1.253E-02
  16       23   3.044E-01   7.308E-03   8.572E-03   2.967E-01   1.316E-02
  17       22   3.047E-01   7.309E-03   8.544E-03   2.980E-01   1.299E-02
  18       21   3.049E-01   7.311E-03   8.659E-03   2.929E-01   1.475E-02
  19**     20   3.039E-01   7.304E-03   7.919E-03   2.929E-01   1.416E-02
  20       15   3.059E-01   7.318E-03   7.691E-03   3.030E-01   1.559E-02
  21       13   3.057E-01   7.316E-03   7.713E-03   3.043E-01   1.599E-02
  22       10   3.062E-01   7.320E-03   7.703E-03   3.131E-01   1.412E-02
  23        8   3.173E-01   7.391E-03   7.984E-03   3.253E-01   1.190E-02
  24        3   3.359E-01   7.501E-03   9.174E-03   3.295E-01   1.160E-02
  25        2   3.417E-01   7.532E-03   1.010E-02   3.317E-01   1.779E-02
  26        1   3.869E-01   7.735E-03   8.543E-03   3.851E-01   1.446E-02

0-SE tree based on mean is marked with * and has 60 terminal nodes
0-SE tree based on median is marked with + and has 20 terminal nodes
```

```
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as + tree
** tree same as -- tree
++ tree same as -- tree
+ tree same as ++ tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
```

Node label	Total cases	Train cases	Predicted class	Node cost	Split variables	Interacting variable
1	3965	3965	D	3.869E-01	INCLASS2	
2	248	248	C	2.299E-01	FINCBTAX	
4T	105	105	C	2.872E-02	FINCBTAX	
5	143	143	C	3.777E-01	FINDRETX	
10	127	127	C	3.072E-01	UNISTRQ	
20T	16	16	D	6.258E-02	-	
21T	111	111	C	2.163E-01	-	
11T	16	16	D	6.258E-02	-	
3	3717	3717	D	3.613E-01	PSU	
6T	126	126	C	1.668E-01	TFOODHOP	
7	3591	3591	D	3.448E-01	STATE	
14T	1360	1360	D	2.154E-01	INCNONW2	
15	2231	2231	D	4.236E-01	RETSURVX	
30	1609	1609	D	4.201E-01	FINDRETX	
60	968	968	D	4.928E-01	INCLASS2	
120	538	538	D	4.108E-01	STOCKX	
240	507	507	D	3.984E-01	RENTEQVX	
480T	378	378	D	3.360E-01	-	
481T	129	129	C	4.187E-01	-	
241T	31	31	C	3.872E-01	-	
121	430	430	C	4.395E-01	EARNCOMP	
242	32	32	T	5.938E-01	FINCBTAX	
484T	10	10	C	1.001E-01	-	
485T	22	22	T	4.091E-01	BUILT	
243	398	398	C	4.196E-01	REFGEN	
486T	89	89	D	4.494E-01	-	
487	309	309	C	3.819E-01	REFGEN	
974T	8	8	C	2.501E-01	-	
975	301	301	C	3.854E-01	CASHCOCQ	
1950T	220	220	C	3.092E-01	-	

```
      1951T      81       81    D             4.074E-01 -
        61      641      641    D             3.105E-01 HLFBATHQ
       122      350      350    D             3.514E-01 FSTAXOWE
       244T     299      299    D             2.977E-01 -
       245T      51       51    C             3.334E-01 -
       123T     291      291    D             2.612E-01 HLFBATHQ
        31      622      622    D             4.325E-01 RETSURVX
        62T      86       86    C             1.397E-01 NETRENTB
        63T     536      536    D             3.638E-01 STOCKYRX

Number of terminal nodes of final tree: 20
Total number of nodes of final tree: 39
Second best split variable (based on curvature test) at root node is STATE

Classification tree:
For categorical variable splits, values not in training data go to the right

Node 1: INCLASS2 = NA
  Node 2: FINCBTAX <= 0.50000000
    Node 4: C
  Node 2: FINCBTAX > 0.50000000 or NA
    Node 5: FINDRETX <= 300.00000
      Node 10: UNISTRQ = "1", "2", "6", "7", "8"
        Node 20: D
      Node 10: UNISTRQ /= "1", "2", "6", "7", "8"
        Node 21: C
    Node 5: FINDRETX > 300.00000 or NA
      Node 11: D
Node 1: INCLASS2 /= NA
  Node 3: PSU = "S49F"
    Node 6: C
  Node 3: PSU /= "S49F"
    Node 7: STATE = "2", "6", "10", "11", "21", "24", "25", "27", "31", "40",
             "41", "47", "48", "49"
      Node 14: D
    Node 7: STATE /= "2", "6", "10", "11", "21", "24", "25", "27", "31", "40",
             "41", "47", "48", "49"
      Node 15: RETSURVX = NA & RETS_RVX = "A"
        Node 30: FINDRETX <= 391.50000
          Node 60: INCLASS2 <= 4.5000000
            Node 120: STOCKX <= 190000.00  or STOCKX = NA & STOCKX_ = "A"
              Node 240: RENTEQVX <= 1990.5000 or NA
                Node 480: D
              Node 240: RENTEQVX > 1990.5000
                Node 481: C
            Node 120: not (STOCKX <= 190000.00 or STOCKX = NA & STOCKX_ = "A")
```

```
              Node 241: C
            Node 60: INCLASS2 > 4.5000000 or NA
              Node 121: EARNCOMP = "8"
                Node 242: FINCBTAX <= 105777.50
                  Node 484: C
                Node 242: FINCBTAX > 105777.50 or NA
                  Node 485: T
              Node 121: EARNCOMP /= "8"
                Node 243: REFGEN = "5"
                  Node 486: D
                Node 243: REFGEN /= "5"
                  Node 487: REFGEN = "1"
                    Node 974: C
                  Node 487: REFGEN /= "1"
                    Node 975: CASHCOCQ <= 112.50000
                      Node 1950: C
                    Node 975: CASHCOCQ > 112.50000 or NA
                      Node 1951: D
          Node 30: FINDRETX > 391.50000 or NA
            Node 61: HLFBATHQ <= 0.50000000
              Node 122: FSTAXOWE <= 10256.500
                Node 244: D
              Node 122: FSTAXOWE > 10256.500 or NA
                Node 245: C
            Node 61: HLFBATHQ > 0.50000000 or NA
              Node 123: D
        Node 15: not (RETSURVX = NA & RETS_RVX = "A")
          Node 31: RETSURVX = NA
            Node 62: C
          Node 31: RETSURVX /= NA
            Node 63: D

 ****************************************************************
 Predictor means below are weighted means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if INCLASS2 = NA
 INCLASS2 mean = 4.4617238
 Class      Number   Posterior
 C            1478  0.3728E+00
 D            2431  0.6131E+00
 T              56  0.1412E-01
 Number of training cases misclassified = 1534
 Predicted class is D
 ----------------------------
Node 2: Intermediate node
```

```
 A case goes into Node 4 if FINCBTAX <= 0.50000000
 FINCBTAX mean = 18232.969
 Class      Number   Posterior
 C             191  0.7701E+00
 D              57  0.2299E+00
 T               0  0.3561E-05
 Number of training cases misclassified = 57
 Predicted class is C
 ----------------------------
Node 4: Terminal node
 Class      Number   Posterior
 C             102  0.9713E+00
 D               3  0.2872E-01
 T               0  0.3561E-05
 Number of training cases misclassified = 3
 Predicted class is C
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if FINDRETX <= 300.00000
 FINDRETX mean = 1054.5743
 Class      Number   Posterior
 C              89  0.6223E+00
 D              54  0.3777E+00
 T               0  0.3561E-05
 Number of training cases misclassified = 54
 Predicted class is C
 ----------------------------
 :
 ----------------------------
Node 31: Intermediate node
 A case goes into Node 62 if RETSURVX = NA
 RETSURVX mean = 25637.530
 Class      Number   Posterior
 C             259  0.4164E+00
 D             353  0.5675E+00
 T              10  0.1608E-01
 Number of training cases misclassified = 269
 Predicted class is D
 ----------------------------
Node 62: Terminal node
 Class      Number   Posterior
 C              74  0.8603E+00
 D              12  0.1397E+00
 T               0  0.3561E-05
 Number of training cases misclassified = 12
 Predicted class is C
```

```
 ----------------------------
Node 63: Terminal node
 Class      Number   Posterior
 C             185  0.3451E+00
 D             341  0.6362E+00
 T              10  0.1866E-01
 Number of training cases misclassified = 195
 Predicted class is D
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              C         D         T
 C                663       212         2
 D                814      2211        41
 T                  1         8        13
 Total           1478      2431        56

 Number of cases used for tree construction: 3965
 Number misclassified: 1078
 Resubstitution estimate of mean misclassification cost: 0.27187894

 Observed and fitted values are stored in classfit.txt
 LaTeX code for tree is in class.tex
 R code is stored in class.r
```

Figure 5 shows the classification tree. It has four different kinds of splits involving missing values:

Node 1. `INCLASS2` = `NA` (go left if and only if `INCLASS2` is missing).

Node 2. `FINCBTAX` ≤ 0.5 (go left if and only `FINCBTAX` is nonmissing and ≤ 0.5).

Node 15. `RETS_RVX` = `A` (go left if and only if `RETSURVX` is missing with flag variable `RETS_RVX` = `A`).

Node 240. `RENTEQVX` $\leq_*$ 1990.5 (go left if and only if `RENTEQVX` = `NA` or $\leq$ 1990.5).

The top several lines of the file of fitted values `class.fit` are given below. The posterior probabilities of predicting class `T` are very low (see Section 4.1.3 for the calculation of the posterior probabilities).

```
 train   node  observed  predicted     "P(C)"       "P(D)"       "P(T)"
    y      6    "C"        "C"     0.83322E+00  0.16678E+00  0.35612E-05
```

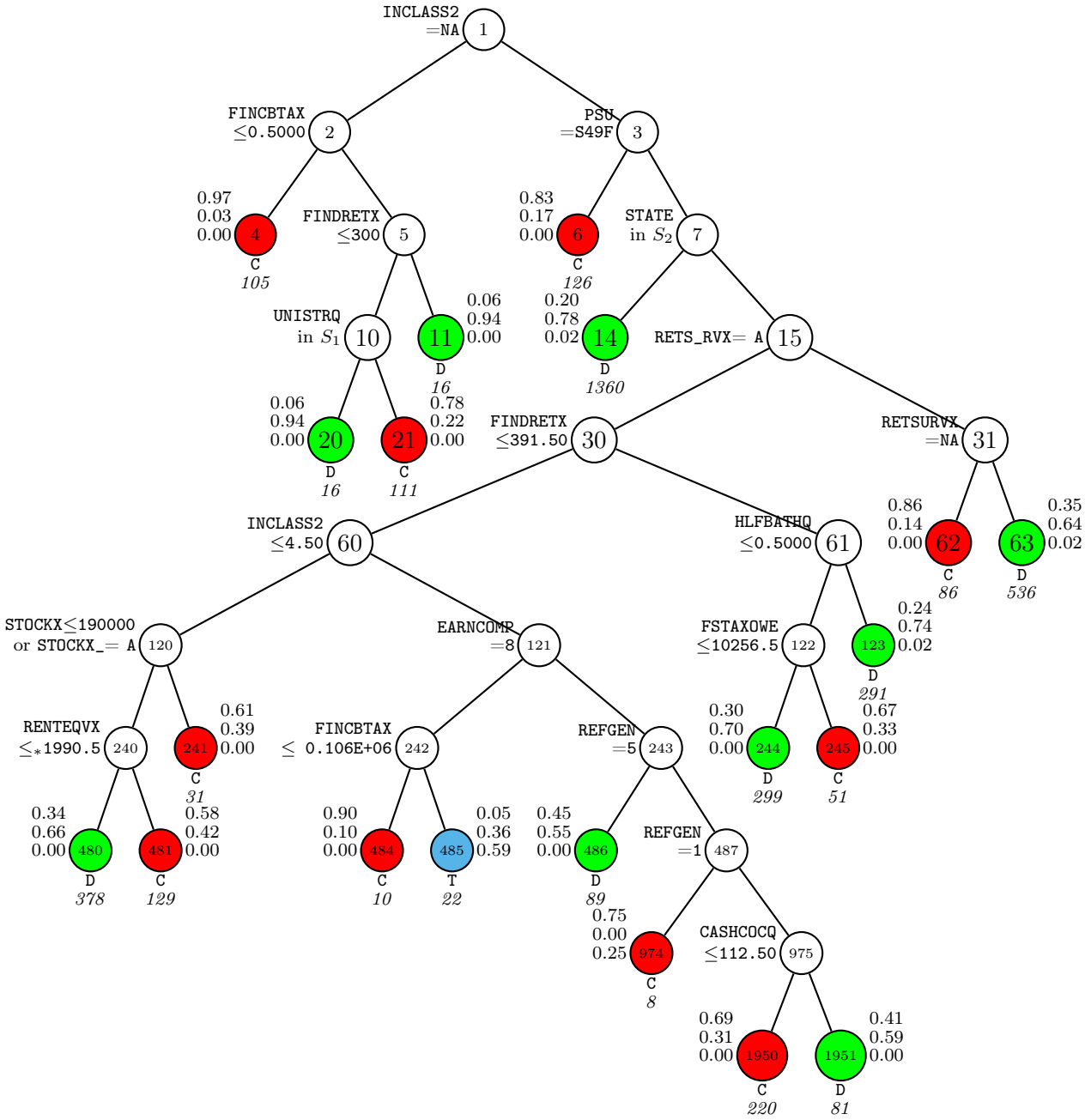

Figure 5: GUIDE v.46.1 0.250-SE classification tree for predicting `INTRDVX_` using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {1, 2, 6, 7, 8}. S_2 = {2, 6, 10, 11, 21, 24, 25, 27, 31, 40, 41, 47, 48, 49}. Predicted class and sample size (in *italics*) printed below each terminal node; class posterior probabilities for `INTRDVX_` = `C`, `D`, and `T`, respectively, beside node. Second best split variable at root node is `STATE`.

```
y     14    "D"        "D"     0.19779E+00  0.78456E+00  0.17647E-01
y     14    "D"        "D"     0.19779E+00  0.78456E+00  0.17647E-01
y   1950    "D"        "C"     0.69083E+00  0.30917E+00  0.35612E-05
y     14    "D"        "D"     0.19779E+00  0.78456E+00  0.17647E-01
y    123    "T"        "D"     0.23711E+00  0.73883E+00  0.24055E-01
```

5.2 Classification: linear trees

Following are results from fitting a classification tree with linear splits and LDA node models. C and M variables are automatically converted to B.

```
Classification tree
Linear discriminant splits and LDA node models
Pruning by cross-validation
DSC file: ce2021class.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
D variable is INTRDVX_
M variables changed to B
C variables changed to B
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 3
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358

Summary information for training sample of size 3965
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                    #Codes/
                                                    Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  DIRACC          n   1.0000E+00    1.000                   170  constant
      2  DIRACC_         b                                 3
```

```
     3  AGE_REF         n    18.00        87.00
     4  AGE_REF_        b                                  2
     5  AGE2            n   2.1000E+01    87.00                    1734
     6  AGE2_           b                                  3
     :
    31  FINLWT21        w    1072.       0.9390E+05
     :
   407  INTRDVX_        d                                  3
     :
   549  FFTAXOWE        n  -0.3368E+05   0.3997E+06
   550  FSTAXOWE        n   -3309.       0.7223E+05
 ================== Constructed variables ==================
   551  DIRACC_.C       f    0.000        1.000
   552  DIRACC_.D       f    0.000        1.000
   553  AGE_REF_.T      f    0.000        1.000
   554  AGE2_.D         f    0.000        1.000
   555  AGE2_.T         f    0.000        1.000
     :
  1100  WHOLIFX.NA      f    0.000        1.000
  1101  VEHICTAX.NA     f    0.000        1.000
  1102  CREDYR.NA       f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1      378      552        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0      164        0        0
Number of constant N variables: 5
Number of dummy variables (excl. missing-value indicators) created: 467
Number of missing-value indicators created: 85
Number of cases used for training: 3965
Number of cases excluded due to 0 W or missing D variable: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Max number of splits: 100
Maximum number of split levels: 15
Minimum node sample size: 30
crimcoords switched in node 2
crimcoords switched in node 5
```

```
crimcoords switched in node 10
crimcoords switched in node 6
crimcoords switched in node 7
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       25   3.972E-01   7.771E-03   1.915E-02   3.934E-01   1.785E-02
   2       24   3.985E-01   7.775E-03   1.863E-02   3.934E-01   1.752E-02
   3       23   3.945E-01   7.762E-03   1.922E-02   3.859E-01   2.171E-02
   4       22   3.876E-01   7.737E-03   2.062E-02   3.733E-01   2.616E-02
   5       21   3.854E-01   7.729E-03   2.106E-02   3.690E-01   2.800E-02
   6       18   3.834E-01   7.721E-03   2.107E-02   3.615E-01   2.667E-02
   7       15   4.005E-01   7.782E-03   3.370E-02   3.577E-01   3.041E-02
   8       14   4.005E-01   7.782E-03   3.370E-02   3.577E-01   3.041E-02
   9       13   3.950E-01   7.763E-03   3.540E-02   3.564E-01   3.171E-02
  10        4   3.602E-01   7.624E-03   3.371E-02   3.140E-01   3.704E-02
  11**      2   3.011E-01   7.285E-03   1.040E-02   2.992E-01   1.294E-02
  12        1   3.291E-01   7.462E-03   8.841E-03   3.317E-01   1.334E-02

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
       Node    Total  Misclassification
      label    cases  cost
          1     3965  3.120E-01
         2T     3664  2.604E-01
         3T      301  2.924E-01

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3

Classification tree with linear discriminant splits and LDA node models

Node 1: D1 <= 0.36376700E-1
  Node 2: 954/3664 misclassified
Node 1: D1 > 0.36376700E-1
  Node 3: 88/301 misclassified
```

```
 ************************************************************
 Predictor means below are weighted means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if D1 <= 0.36376700E-1 where the coefficients of D1
 are the coefficients of the 1st crimcoord below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 AGE_REF           0.10286356E-03  -0.18305945E-03    56.815873
 AGE2             -0.55263465E-05  -0.17796121E-03    56.569604
 BATHRMQ          -0.66238493E-03  -0.23729633E-03    1.9788369
 BEDROOMQ          0.26475570E-02  -0.14139684E-02    3.0598017
 EDUC_REF         -0.14127559E-02   0.37877657E-02    14.419353
 :
 RETSRVBX.NA      -0.24148305E-03   0.41288727E-03   0.98644714
 STOCKYRX.NA      -0.39704159E-03  -0.47029904E-03   0.93379120
 STOCKX.NA        -0.43097163E-03  -0.36438702E-03   0.92629161
 WHOLIFX.NA        0.31332909E-06  -0.17989280E-03   0.97664602

 Class      Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 C       0.180418E-01    0.201764E-03   -0.657768E-02    0.217229E-03     0.37276
 D       0.127579E-01    0.201764E-03    0.492010E-02    0.217229E-03     0.61311
 T       0.710481E-01    0.201764E-03    0.185411E-01    0.217229E-03     0.01412

 Class    Correlation
 C      -0.218710
 D       0.328716E-01
 T       0.870809
 ----------------------------
Node 2: Terminal node
 Cases classified by LDA on 1st and 2nd crimcoords with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 AGE_REF           0.96285117E-04  -0.63857323E-04    56.278127
 EDUC_REF         -0.19041907E-04   0.14995258E-02    14.406429
 FINDRETX         -0.47640375E-06   0.30569910E-06    3503.0889
 :
 STOCKYRX.NA       0.50317660E-02   0.30502394E-03   0.93767194
 STOCKX.NA        -0.56818615E-03  -0.53271423E-03   0.92981836

 Class      Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 C       0.258611E-01    0.239888E-03    0.470209E-02    0.218749E-03     0.36081
 D       0.242641E-01    0.239888E-03    0.199238E-01    0.218749E-03     0.63783
 T       0.179619        0.239888E-03    0.409829E-01    0.218749E-03     0.00136

 Class    Correlation
 C      -0.541558E-01
 D       0.518177E-02
```

```
 T       0.954405
 ----------------------------
Node 3: Terminal node
 Cases classified by LDA on 1st and 2nd crimcoords with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 BLS_URBN          0.33425912E-01   0.11807432        1.0367633
 FINCBTAX         -0.16920898E-06  -0.14394214E-06    206064.08
 INC_RANK          0.42300178E-01  -0.25649083E-01   0.69192227
 ERANKHM          -0.71334772E-02  -0.25523355E-01   0.74976596
 FS_MTHI           0.25232937E-01   0.38869046E-01    3.8260458
 STUDFINX         -0.88087983E-03   0.12671364E-02    12.736634
 FINCBT_X.T        0.46683415E-01   0.43507913E-01   0.52302645
 INC__ANK.T        0.46683415E-01   0.43507913E-01   0.52302645
 ERANKH_.D         0.14007054E-01  -0.60120585E-02   0.72058900
 LIQU_YRX.D       -0.15443567E-01  -0.63917957E-01   0.11509276
 LIQUIDX_.C       -0.25385359E-01   0.25380838E-01   0.10418600
 LIQUIDX_.D        0.36041174E-01  -0.92514583E-01   0.12576118
 RETS_RVX.C       -0.29421341E-01   0.49024608E-01   0.13415560
 RETS_VBX.B        0.15085335E-01  -0.15294843E-01   0.97117048E-01
 INCLASS2.NA      -0.14007054E-01   0.60120585E-02   0.27941100
 IRAX.NA           0.62440841E-03   0.34600390E-01   0.87740311
 IRAYRX.NA         0.20033036E-01   0.25966634E-01   0.88171064
 LIQUDYRX.NA       0.12205883E-01  -0.88538824E-01   0.85571789
 LIQUIDX.NA       -0.46188132E-01  -0.58461879E-01   0.83981802
 RETSURVX.NA       0.24330200E-01   0.70294665E-01   0.84697755

 Class      Crim1Mean   Crim1Variance       Crim2Mean   Crim2Variance      Prior
 C       0.138806        0.184603E-02    0.250541        0.273223E-02     0.51827
 D       0.192089        0.184603E-02    0.205875        0.273223E-02     0.31229
 T       0.240629        0.184603E-02    0.274041        0.273223E-02     0.16944

 Class    Correlation
 C      -0.253259
 D       0.189470
 T       0.457696
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              C         D         T
 C                703       234         0
 D                764      2184        20
 T                 11        13        36
 Total           1478      2431        56

 Number of cases used for tree construction: 3965
```

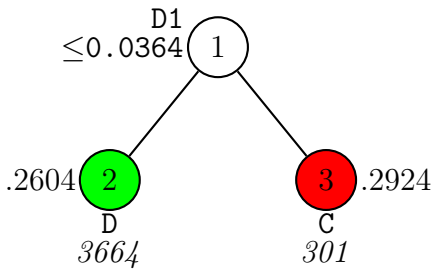

Figure 6: GUIDE v.46.1 0.250-SE classification tree for predicting INTRDVX_ using linear discriminant splits and linear discriminant node models. At each split, an observation goes to the left branch if and only if the condition is satisfied. Class minimizing misclassification cost and sample size (in *italics*) printed below each terminal node; mean misclassification cost beside node.

```
Number misclassified: 1042
Resubstitution estimate of mean misclassification cost: 0.26279950

Observed and fitted values are stored in lda.fit
LaTeX code for tree is in lda.tex
R code is stored in lda.r
Discriminant coefficients are stored in lda.coefs
R function for plotting data are stored in ldaplot.r
Plot data are stored in ldadata.txt
```

Figure 6 shows the tree and Figure 7 shows the data in the nodes.

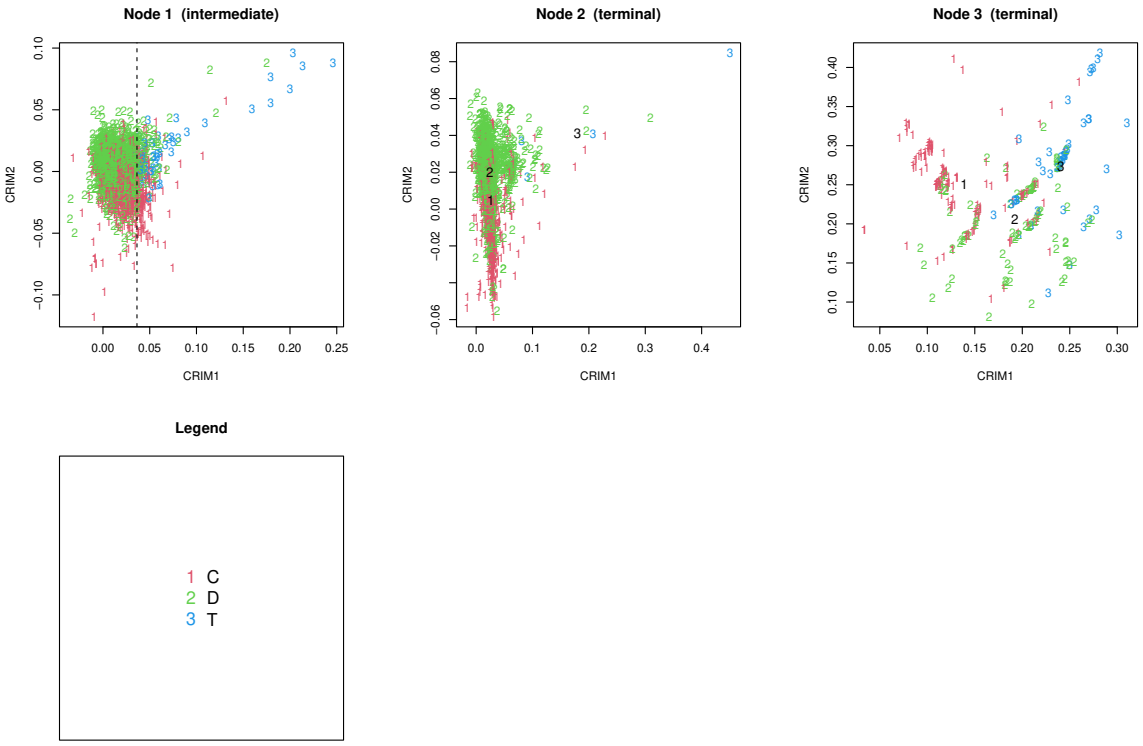

Figure 7: Data and splits in nodes of Figure 6. Class means are indicated by numbers in black.

6 Least squares regression

GUIDE can fit least-squares (LS), quantile, Poisson, proportional hazards, and least-median-of-squares (LMS) regression tree models. We illustrate least squares and quantile models with the CE data, using `INTRDVX` as the dependent (`d`) variable and excluding (`x`) its flag `INTRDVX_`. The DSC file is `ce2021reg.dsc`, which sets `FINLWT21` as a weight (`w`) variable.

6.1 Piecewise constant

6.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: cons.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: cons.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
```

```
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Default max. number of split levels: 12
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2): 1
Input file name to store LaTeX code (use .tex as suffix): cons.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: cons.fit
You can output an R function for predicting future observations
```

```
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: cons.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < cons.in
```

6.1.2 Contents of `cons.out`

```
Least squares regression tree
Pruning by cross-validation
DSC file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     s    1.000        1.000                 125  constant
      2  DIRACC_    m                                 2
      3  AGE_REF    s    19.00        87.00
      4  AGE_REF_   m                                 0
      :
     31  FINLWT21   w    1072.       0.9390E+05
```

```
      :
    404  FSMPFRMX   s  -0.1160E+06   0.7703E+06
    405  FSMP_RMX   m                                  0
    406  INTRDVX    d    1.000       0.1413E+06
      :
    547  WHLFYR     c                                  1     2487
    548  WHLFYR_    m                                  1
    549  FFTAXOWE   s  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   s   -3074.       0.5654E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      376
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of constant S variables: 7
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction and linear splits; too many predictor variables
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 12
Minimum node sample size: 2
Ranks of variables and their 1-df chi-squared values at root node
     1  0.8297E+02   REFGEN
     2  0.8111E+02   AGE_REF
     3  0.7066E+02   INCNONW1
     :
   416  0.3569E-02   TGASMOTC
   417  0.1029E-03   MAINRPPQ

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       62   6.106E+12   7.688E+11   4.771E+11   5.779E+12   7.257E+11
   2       61   6.106E+12   7.688E+11   4.771E+11   5.779E+12   7.257E+11
   :
  35       10   6.103E+12   7.677E+11   4.945E+11   5.796E+12   7.342E+11
```

```
  36**      9   5.915E+12   7.601E+11   4.635E+11   5.668E+12   6.327E+11
  37        8   6.433E+12   8.423E+11   5.350E+11   6.000E+12   7.476E+11
  38        5   6.573E+12   8.685E+11   6.476E+11   6.000E+12   9.316E+11
  39        4   7.614E+12   9.359E+11   7.113E+11   8.502E+12   1.107E+12
  40        1   8.287E+12   1.032E+12   6.955E+11   8.542E+12   7.418E+11

0-SE tree based on mean is marked with * and has 9 terminal nodes
0-SE tree based on median is marked with + and has 9 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of INTRDVX in the node
Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases Matrix    Node      Node          Split
       label   cases      fit  rank    D-mean     MSE           variable
          1    2487     2487     1  5.131E+03  8.287E+12   REFGEN
          2T    573      573     1  9.804E+02  1.105E+12   PSU
          3    1914     1914     1  6.476E+03  1.027E+13   INC_RANK
          6T   1345     1345     1  2.823E+03  1.657E+12   REF_RACE
          7     569      569     1  1.498E+04  2.828E+13   EARNCOMP
         14      75       75     1  5.278E+04  7.423E+13   RETSURV
         28T     46       46     1  2.764E+04  4.076E+13   -
         29T     29       29     1  8.586E+04  7.873E+13   -
         15     494      494     1  9.170E+03  1.560E+13   FFTAXOWE
         30T    247      247     1  2.647E+03  3.560E+12   UNISTRQ
         31     247      247     1  1.570E+04  2.573E+13   AGE2
         62T    156      156     1  8.036E+03  1.274E+13   LIQDYRBX
         63      91       91     1  3.030E+04  4.131E+13   BATHRMQ
        126      87       87     1  2.564E+04  3.295E+13   STATE
        252T     42       42     1  4.145E+04  5.339E+13   -
        253T     45       45     1  5.032E+03  8.537E+11   STATE
        127T      4        4     1  1.243E+05  8.997E+12   -

Number of terminal nodes of final tree: 9
Total number of nodes of final tree: 17
Second best split variable (based on curvature test) at root node is AGE_REF

Regression tree:
For categorical variable splits, values not in training data go to the right
```

```
 Node 1: REFGEN = "5"
   Node 2: INTRDVX-mean = 980.35292
 Node 1: REFGEN /= "5"
   Node 3: INC_RANK <= 0.84018625
     Node 6: INTRDVX-mean = 2822.6445
   Node 3: INC_RANK > 0.84018625 or NA
     Node 7: EARNCOMP = "8"
       Node 14: RETSURV = "1"
         Node 28: INTRDVX-mean = 27641.282
       Node 14: RETSURV /= "1"
         Node 29: INTRDVX-mean = 85859.276
     Node 7: EARNCOMP /= "8"
       Node 15: FFTAXOWE <= 27769.500
         Node 30: INTRDVX-mean = 2646.5367
       Node 15: FFTAXOWE > 27769.500 or NA
         Node 31: AGE2 <= 56.500000 or NA
           Node 62: INTRDVX-mean = 8036.3341
         Node 31: AGE2 > 56.500000
           Node 63: BATHRMQ <= 4.5000000
             Node 126: STATE = "6", "17", "21", "24", "27", "29", "32", "37", "42"
               Node 252: INTRDVX-mean = 41453.995
             Node 126: STATE /= "6", "17", "21", "24", "27", "29", "32", "37", "42"
               Node 253: INTRDVX-mean = 5032.1141
           Node 63: BATHRMQ > 4.5000000 or NA
             Node 127: INTRDVX-mean = 124323.81

 ****************************************************************
 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if REFGEN = "5"
 REFGEN mode = "3"
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
```

```
 Constant      5131.        12.43       0.3014-313
 ----------------------------
Node 2: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant      980.4        3.232       0.3014-313
 INTRDVX mean = 980.353
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if INC_RANK <= 0.84018625
 INC_RANK mean = 0.64479638
 ----------------------------
Node 6: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant      2823.        11.12       0.3014-313
 INTRDVX mean = 2822.64
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if EARNCOMP = "8"
 EARNCOMP mode = "2"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if RETSURV = "1"
 RETSURV mode = "1"
 ----------------------------
Node 28: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.2764E+05    4.116       0.3014-313
 INTRDVX mean = 27641.3
 ----------------------------
Node 29: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.8586E+05    7.878       0.3014-313
 INTRDVX mean = 85859.3
 ----------------------------
Node 15: Intermediate node
 A case goes into Node 30 if FFTAXOWE <= 27769.500
 FFTAXOWE mean = 36491.278
 ----------------------------
Node 30: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant      2647.        3.066       0.3014-313
```

```
 INTRDVX mean = 2646.54
 ----------------------------
Node 31: Intermediate node
 A case goes into Node 62 if AGE2 <= 56.500000 or NA
 AGE2 mean = 54.155118
 ----------------------------
Node 62: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant      8036.        3.982       0.3014-313
 INTRDVX mean = 8036.33
 ----------------------------
Node 63: Intermediate node
 A case goes into Node 126 if BATHRMQ <= 4.5000000
 BATHRMQ mean = 2.8097992
 ----------------------------
Node 126: Intermediate node
 A case goes into Node 252 if STATE = "6", "17", "21", "24", "27", "29", "32", "37", "42"
 STATE mode = "42"
 ----------------------------
Node 252: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.4145E+05    5.568       0.3014-313
 INTRDVX mean = 41454.0
 ----------------------------
Node 253: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant      5032.        4.335       0.3014-313
 INTRDVX mean = 5032.11
 ----------------------------
Node 127: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1243E+06    14.02       0.3014-313
 INTRDVX mean = 124324.
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.4174

 Observed and fitted values are stored in cons.fit
 LaTeX code for tree is in cons.tex
 R code is stored in cons.r
```

In the above results, the pruned tree is marked with two asterisks (tree #36).

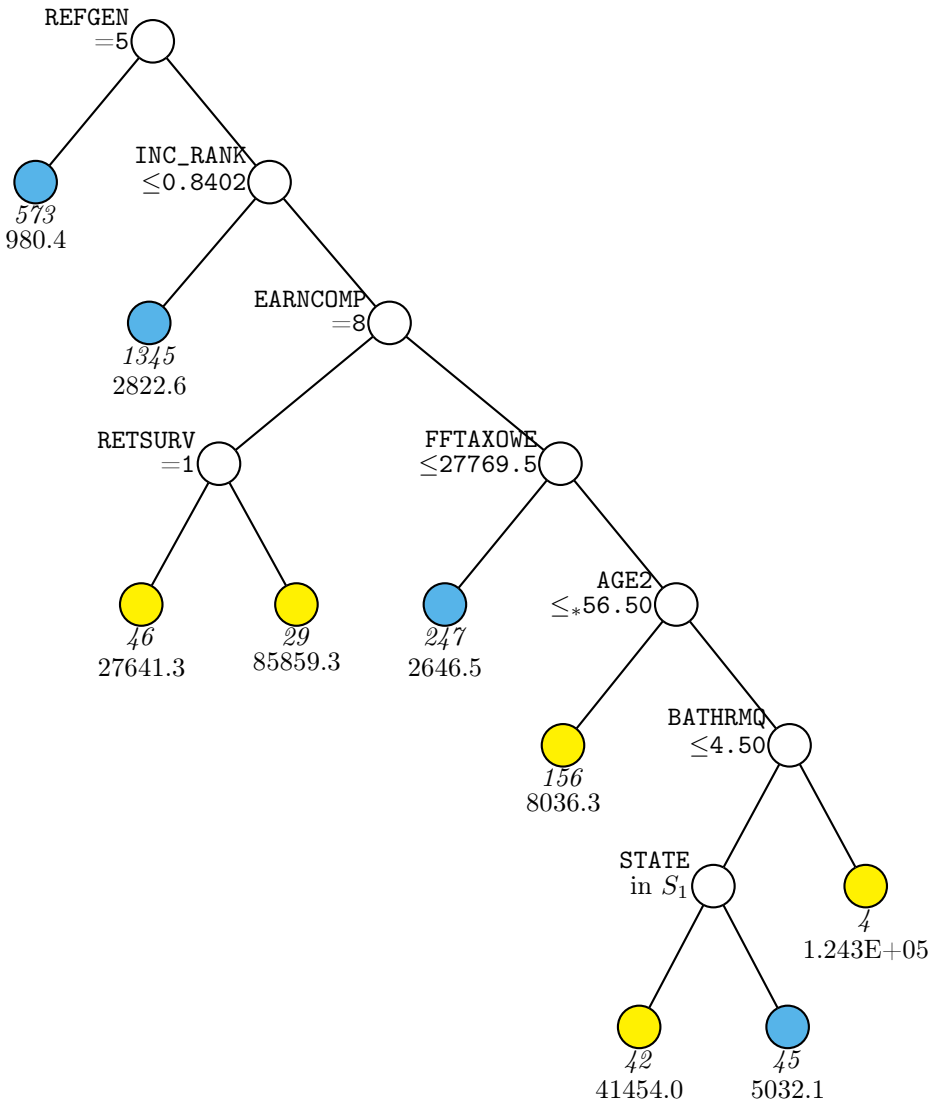

Figure 8: GUIDE v.46.1 0.250-SE piecewise-constant weighted least-squares regression tree for predicting INTRDVX. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. $S_1 = \{6, 17, 21, 24, 27, 29, 32, 37, 42\}$. Sample size (in *italics*) and weighted mean of INTRDVX printed below nodes. Terminal nodes with means above and below value of 5130.6 at root node are painted yellow and skyblue respectively. Second best split variable at root node is AGE_REF.

It has 9 terminal nodes and a cross-validation estimate of prediction mean squared error of 5.915E+12. Figure 8 shows the tree. The first split is on REFGEN=5, meaning that millenials go to node 2, which is terminal. The first 7 entries of cons.fit below show that the 1st observation, for which INTRDVX is missing (the letter "n" in the first column indicates that it is not used to train the model), belongs to node 6 and has a predicted value of $2822.64.

train	node	observed	predicted
n	6	NA	2822.64
y	6	1087.00	2822.64
y	6	1000.00	2822.64
y	6	300.000	2822.64
y	2	10.0000	980.353
y	252	141304.	41454.0
y	6	55.0000	2822.64

6.1.3 Population mean estimation

Predicted values from the regression tree may be used to estimate population means. Let w_i denote the sampling weight (FINLWT21) and S_1, S_2 denote the sets of observations nonmissing and missing y_i (INTRDVX). Then an estimate of the population mean of INTRDVX is

$$\Big(\sum_{k\in S_1\cup S_2} w_k\Big)^{-1}\Big(\sum_{i\in S_1} w_i y_i + \sum_{j\in S_2} w_j \hat{y}_j\Big) \tag{1}$$

where $\hat{y}_j$ denotes the predicted value of INTRDVX. The R code below gives an estimated population mean INTRDVX of 4750.376. See Loh et al. (2019b) for a similar analysis of an earlier data set.

```
data <- read.table("ce2021.txt",header=TRUE)
y <- data$INTRDVX
w <- data$FINLWT21
fitted <- read.table("cons.fit",header=TRUE)
pred <- fitted$predicted
S1 <- !is.na(fitted$observed)
S2 <- is.na(fitted$observed)
popmean <- (sum(w[S1]*y[S1])+sum(w[S2]*pred[S2]))/sum(w)
```

6.2 Piecewise simple polynomial

GUIDE can also fit a simple polynomial regression model in each node of the form

$$y = \beta_0 + \sum_{k=1}^{p} \beta_k x^k + \epsilon \tag{2}$$

where p is the degree of polynomial desired and x is selected from the set of `n` and `f` variables. The variable x is the one among all `n` and `f` variables that yields the smallest sum of squared residuals. Variable x can vary from node to node. If there are missing values in the x variable, GUIDE fits two separate models to the data in the node: model (2) to the observations with complete values in x and y and a constant ($y = \beta_0 + \epsilon$) to those with missing values in x. This is equivalent to imputing missing x values with a constant c and adding the missing-value indicator $I(x = \texttt{NA})$ as linear predictor:

$$y = \beta_0 + \sum_{k=1}^{p} \beta_k x_1^k + \beta_2 x_2 + \epsilon$$

where $x_1 = xI(x \neq \texttt{NA}) + cI(x = \texttt{NA})$ and $x_2 = I(x = \texttt{NA})$. The predicted values are independent of c but the least-squares estimates of the β coefficients are not.

Truncation note: Extrapolation can adversely affect the prediction accuracy of parametric models. To guard against extrapolation, GUIDE has several options to truncate the predicted values, with the default being to truncate the predicted values if they fall outside the range of the observed values. The option of no truncation is available as well. Default truncation is used in this manual.

6.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: lin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
```

```
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      383        0        0
```

```
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Default max. number of split levels: 12
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: lin.var
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: lin.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in
```

6.2.2 Partial output

```
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       36   6.941E+12  8.475E+11   9.177E+11   5.842E+12   1.280E+12
   2       35   6.941E+12  8.475E+11   9.177E+11   5.842E+12   1.280E+12
   :
  12       22   6.861E+12  8.491E+11   1.085E+12   5.115E+12   1.436E+12
  13+      21   6.861E+12  8.491E+11   1.085E+12   5.115E+12   1.436E+12
  14       18   6.874E+12  8.491E+11   1.084E+12   5.206E+12   1.434E+12
  15       17   6.833E+12  8.484E+11   1.086E+12   5.166E+12   1.452E+12
  16       16   6.714E+12  8.414E+11   1.061E+12   5.195E+12   1.303E+12
  17       10   6.597E+12  8.385E+11   1.018E+12   5.157E+12   1.347E+12
  18        9   6.432E+12  8.279E+11   1.060E+12   5.154E+12   1.439E+12
  19**      8   6.144E+12  7.700E+11   8.069E+11   5.239E+12   1.583E+12
```

```
  20        7   6.430E+12   8.188E+11   7.088E+11   5.868E+12   1.326E+12
  21        5   7.158E+12   8.931E+11   7.326E+11   7.966E+12   1.153E+12
  22        4   7.397E+12   9.042E+11   7.487E+11   8.079E+12   1.044E+12
  23        1   8.303E+12   1.011E+12   7.002E+11   8.514E+12   8.534E+11

0-SE tree based on mean is marked with * and has 8 terminal nodes
0-SE tree based on median is marked with + and has 21 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree
* tree same as ** tree
* tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of INTRDVX in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split         Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable      variables
          1    2487      209     2  5.131E+03  7.702E+12  0.0710 CUTENURE  +STOCKYRX
          2     855      855     2  8.856E+03  1.219E+13  0.1292 FJSSDEDX  +FINCBTAX
          4     578      578     2  9.919E+03  1.137E+13  0.2914 FINCBTAX  +FINCBTAX
          8T    500      500     2  3.863E+03  1.323E+12  0.3043 INC_RANK  +ALCBEVCQ
          9      78       47     2  5.014E+04  4.793E+13  0.2919 RETSURV  -RETSURVX
         18T     47       47     2  2.613E+04  2.689E+13  0.2996 -  +FULOILCQ
         19T     31        7     2  8.229E+04  3.579E+13  0.5077 -  -ROYESTX
          5T    277      277     2  6.780E+03  4.930E+12  0.4882 PERINSCQ  +ETOTALC
          3    1632     1087     2  3.221E+03  4.540E+12  0.1032 RENTEQVX  +RENTEQVX
          6T   1520     1520     2  1.846E+03  1.564E+12  0.0959 PSU  +VELECTRC
          7     112      112     2  2.163E+04  3.560E+13  0.1590 FOODCQ  +JFS_AMT
         14      58       58     2  2.970E+04  2.765E+13  0.5220 FINCBTAX  +PERINSCQ
         28T     26       26     2  1.943E+04  7.830E+12  0.8283 -  +APPARCQ
         29T     32       32     2  3.722E+04  1.910E+13  0.7186 -  +HEALTHCQ
         15T     54       54     2  1.262E+04  1.537E+13  0.3474 HLTHINPQ  +JFS_AMT

Number of terminal nodes of final tree: 8
Total number of nodes of final tree: 15
Second best split variable (based on curvature test) at root node is REFGEN
```

```
 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: CUTENURE = "2", "6"
   Node 2: FJSSDEDX <= 2720.0000
     Node 4: FINCBTAX <= 114750.50
       Node 8: INTRDVX-mean = 3863.4422
     Node 4: FINCBTAX > 114750.50 or NA
       Node 9: RETSURV = "1"
         Node 18: INTRDVX-mean = 26127.783
       Node 9: RETSURV /= "1"
         Node 19: INTRDVX-mean = 82288.430
   Node 2: FJSSDEDX > 2720.0000 or NA
     Node 5: INTRDVX-mean = 6780.2396
 Node 1: CUTENURE /= "2", "6"
   Node 3: RENTEQVX <= 3625.0000 or NA
     Node 6: INTRDVX-mean = 1845.8461
   Node 3: RENTEQVX > 3625.0000
     Node 7: FOODCQ <= 1029.1667
       Node 14: FINCBTAX <= 244530.00
         Node 28: INTRDVX-mean = 19427.633
       Node 14: FINCBTAX > 244530.00 or NA
         Node 29: INTRDVX-mean = 37216.426
     Node 7: FOODCQ > 1029.1667 or NA
       Node 15: INTRDVX-mean = 12623.057

 ****************************************************************
 Predictor means below are weighted means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if CUTENURE = "2", "6"
 CUTENURE mode = "1"
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      3167.        2.042       0.4246E-01
```

```
 STOCKYRX     0.1749E-01    11.42        0.000        0.000       0.3617E+06   0.5450E+07
 If regressor has missing values, predicted value = 4720.7960
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if FJSSDEDX <= 2720.0000
 FJSSDEDX mean = 3273.8110
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if FINCBTAX <= 114750.50
 FINCBTAX mean = 60208.956
 ----------------------------
Node 8: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      2892.        8.287       0.9637E-15
 ALCBEVCQ      19.81        14.76       0.1403E-15    0.000        49.04        4670.
 If regressor has missing values, predicted value = 3863.4422
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if RETSURV = "1"
 RETSURV mode = "1"
 ----------------------------
:
Node 29: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1557E+05    2.856       0.7707E-02
 HEALTHCQ      83.68        8.753       0.9246E-09    0.000        258.7        1470.
 If regressor has missing values, predicted value = 37216.426
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 15: Terminal node
 Coefficients of weighted least squares regression function and weighted means:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      9035.        2.534       0.1432E-01
 JFS_AMT       93.41        5.262       0.2738E-05    0.000        38.41        1050.
 If regressor has missing values, predicted value = 12623.057
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.6047

 Observed and fitted values are stored in lin.fit
 Regressor names and coefficients are stored in lin.var
 LaTeX code for tree is in lin.tex
```

Figure 9 shows the pruned tree, which has 8 terminal nodes and a cross-validation estimate of prediction mean squared error of 6.142E+12. Below each terminal node are printed the sample size (in *italics*), the sample mean of `INTRDVX` and the signed simple linear predictor, with the sign being that of the slope coefficient. Nodes with mean of the `d` variable above and below the mean at the root node are colored yellow and purple, respectively.

6.2.3 Plots of data

Figure 10 shows plots of the data and fitted regression lines in the terminal nodes of the tree. The plots are drawn using the R code in Figure 11, which reads the files `lin.fit` and `lin.var`. The contents of the latter are below. The first row is a header line. Each subsequent row gives the terminal node number, predictor variable name, intercept and slope of the regression line, and lower and upper truncation limits on the predicted values (the defaults are the global minimum and maximum observed values of the dependent variable).

```
node   variable   beta0         beta1     lower   upper
   8   ALCBEVCQ   2892.          19.81     1.000   0.1413E+6
  18   FULOILCQ   0.2096E+5    181.5       1.000   0.1413E+6
  19   ROYESTX    0.1215E+6     -0.5119    1.000   0.1413E+6
   5   ETOTALC   -953.9          0.7695    1.000   0.1413E+6
   6   VELECTRC   1875.        166.1       1.000   0.1413E+6
  14   DMSXCCPQ   0.1346E+5    112.6       1.000   0.1413E+6
  15   ECARTKUC   7371.        302.2       1.000   0.1413E+6
```

6.3 Stepwise linear

Besides piecewise constant and best simple polynomial, GUIDE can fit a multiple linear (where all `n` and `f` variables are used as regressors) or a stepwise linear (where forward and backward selection is used to select a subset of regressors) regression model at each node. Quite often, these models have higher prediction accuracy, as hinted by the cross-validation estimates of MSE in the output.

For stepwise regression, missing values in each x variable are imputed with the weighted mean of x in the node and a stepwise linear regression model is fitted to the y variable, using the imputed x variables and their missing-value indicators. The name of the indicator of x is denoted by "x.`NA`", where x.`NA` $= I(x =$ `NA`$)$.

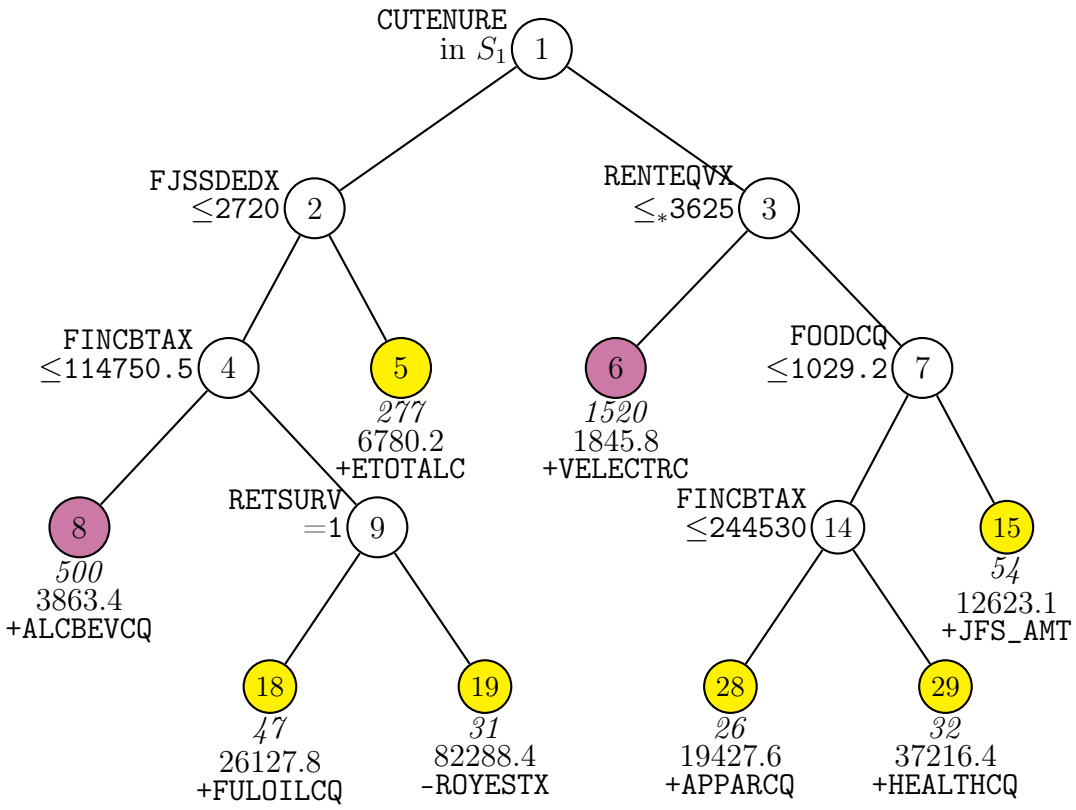

Figure 9: GUIDE v.46.1 0.250-SE piecewise simple linear weighted least-squares regression tree (constant fitted to incomplete cases in terminal nodes) for predicting INTRDVX. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. $S_1 = \{2, 6\}$. Sample size (in *italics*), weighted mean of INTRDVX, and name of regressor (with sign of slope) printed below nodes. Terminal nodes with means above and below value of 5130.6 at root node are painted yellow and purple respectively. Second best split variable at root node is REFGEN.

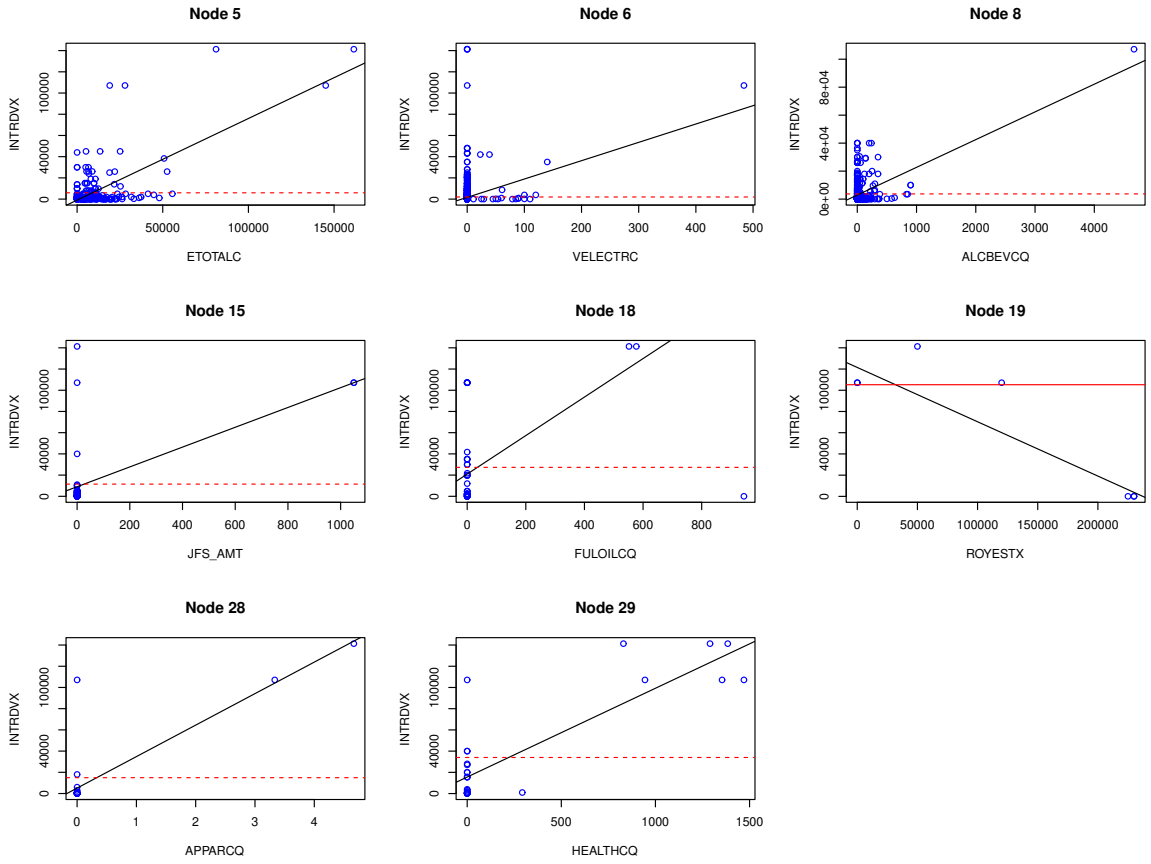

Figure 10: Data and regression lines in terminal nodes of tree in Figure 9. If there are missing values in the regressor, a solid red line marks their d mean. If there are no missing values, a dashed red line marks the d mean of all points in the node.

```
z <- read.table("ce2021.txt",header=TRUE)
par(mfrow=c(3,3))
z1 <- read.table("lin.fit",header=TRUE)
z2 <- read.table("lin.var",header=TRUE)
nodes <- unique(sort(z1$node))
y <- z$INTRDVX
for(n in nodes){
    gp <- z1$node == n & z1$train == "y"
    vrow <- z2$node == n
    b0 <- z2$beta0[vrow]
    b1 <- z2$beta1[vrow]
    reg <- z2$variable[vrow]
    k <- which(names(z) %in% reg)
    x <- z[,k]
    plot(y[gp] ~ x[gp],xlab=reg,ylab="INTRDVX",col="blue")
    abline(c(b0,b1))
    nomiss <- z1$node == n & z1$train == "y" & !is.na(x)
    if(sum(nomiss) < sum(gp)){
        miss <- z1$node == n & z1$train == "y" & is.na(x)
        abline(h=mean(y[miss]),col="red",lty=1)
    } else {
        abline(h=mean(y[gp]),col="red",lty=2)
    }
    title(paste("Node",n))
}
```

Figure 11: R code for Figure 10

6.3.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: step.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: step.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3): 0
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input 1 for univariate splits, 2 for univariate+linear splits ([1:2], <cr>=2):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
D variable is INTRDVX
M variables changed to B
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 164 categorical variables
Finished assigning codes to 10 categorical variables
```

```
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Finished assigning codes to 50 categorical variables
Finished assigning codes to 60 categorical variables
Finished assigning codes to 70 categorical variables
Finished assigning codes to 80 categorical variables
Finished assigning codes to 90 categorical variables
Finished assigning codes to 100 categorical variables
Finished assigning codes to 110 categorical variables
Finished assigning codes to 120 categorical variables
Finished assigning codes to 130 categorical variables
Finished assigning codes to 140 categorical variables
Finished assigning codes to 150 categorical variables
Finished assigning codes to 160 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      383      309        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0      116       48        0
Number of dummy variables (excl. missing-value indicators) created: 224
Number of missing-value indicators created: 85
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Default max. number of split levels: 8
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): step.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
```

```
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: step.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: step.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: step.r
Input rank of top variable to split root node ([1:856], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < step.in
```

6.3.2 Results

The tree has no splits because the fitted regression model in the root node has a R^2 value very close to 1.0. The regression coefficients from `step.out` are below.

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	0.2316E+05	876.2	0.3625E-16			
FINCBTAX	1.000	0.3081E+05	0.6113E-16	-0.1050E+06	0.1160E+06	0.1039E+07
FJSSDEDX	-0.1466E-03	-0.9102	0.3628	0.000	6354.	0.4366E+05
FRRETIRX	-1.000	-0.2400E+05	0.1970E-15	0.000	0.1071E+05	0.1116E+06
FSALARYX	-1.000	-0.2841E+05	0.8374E-16	0.000	0.8172E+05	0.7645E+06
FSSIX	-1.000	-3698.	0.1310E-15	0.000	83.02	0.3600E+05
OTHRINCX	-1.000	-6139.	0.6918E-16	150.0	0.2064E+05	0.1041E+06
PERSLT18	-0.3120	-0.7659	0.4438	0.000	0.3957	7.000
DMSXCCPQ	0.7173E-03	1.021	0.3076	-2408.	144.6	0.1527E+05
CASHCOPQ	-0.6518E-05	-0.7545E-01	0.9399	0.000	802.6	0.1200E+06
EENTRMTC	-0.7989E-04	-0.2645	0.7914	0.000	376.7	0.3022E+05
FSMPFRMX	-1.000	-0.2671E+05	0.5637E-16	-0.1160E+06	6641.	0.7703E+06
JFS_AMT	-0.9994	-1843.	0.3068E-15	0.000	64.43	9600.
NETRENTX	-1.000	-0.1885E+05	0.1083E-15	-0.1402E+05	0.1462E+05	0.1589E+06
NETRNTBX	-1.000	-3706.	0.1756E-15	-2400.	9325.	0.7130E+05
OTHREGBX	-1.000	-2468.	0.000	488.0	6380.	0.4200E+05
OTHREGX	-0.9998	-9576.	0.000	100.0	0.1281E+05	0.8288E+05
RETSURVX	-1.000	-0.2268E+05	0.4449E-16	134.0	0.2762E+05	0.1739E+06
RETSRVBX	-1.000	-5520.	0.000	3500.	0.2850E+05	0.6200E+05
ROYESTBX	-1.001	-244.1	0.2051E-15	200.0	7464.	0.6000E+05
ROYESTB	5.369	0.2390	0.8112	1.000	2.461	12.00
ROYESTX	-1.000	-0.1609E+05	0.000	5.000	0.4599E+05	0.2300E+06
FINCBT_X.T	0.4732	0.1146	0.9088	0.000	0.1044	1.000
FSAL_RYX.T	0.8994	0.1913	0.8483	0.000	0.4732E-01	1.000

```
OTHR_NCX.T      -0.5737      -0.3731E-01   0.9702        0.000       0.3928E-02    1.000
WELF_REX.C      -0.1096E+05   -323.0        0.000        0.000       0.9226E-04    1.000
WELF_REX.D       -915.6       -131.3       0.2932E-15   0.000       0.2086E-02    1.000
FSMP_RMX.T       -2.382      -0.4181       0.6759       0.000       0.1027E-01    1.000
NETR_NTB.D       -9325.       -1538.       0.2245E-15   0.000       0.2754E-02    1.000
NETR_NTX.T      -0.8048      -0.1449       0.8848       0.000       0.6320E-02    1.000
OTHR_GBX.D       -6380.       -1647.       0.9177E-16   0.000       0.6767E-02    1.000
OTHREGX_.T       -15.27       -1.556       0.1197       0.000       0.2762E-02    1.000
RETS_RVX.T       -4.974      -0.8233       0.4104       0.000       0.1413E-01    1.000
ROYE_TBX.D       -7463.       -1150.        0.000       0.000       0.2422E-02    1.000
ROYESTX_.T      -0.8850      -0.7034E-01   0.9439       0.000       0.4268E-02    1.000
OTHRINCX.NA      0.2064E+05    7103.       0.4358E-16   0.000       0.9715        1.000
NETRENTX.NA      0.1462E+05   0.1090E+05   0.9388E-16   0.000       0.9211        1.000
OTHREGX.NA       0.1281E+05   0.1070E+05   0.1820E-15   0.000       0.9028        1.000
RETSURVX.NA      0.2762E+05   0.2160E+05    0.000       0.000       0.7632        1.000
RETSRVBX.NA      0.2850E+05    5780.       0.1534E-15   0.000       0.9949        1.000
ROYESTX.NA       0.4599E+05   0.1798E+05    0.000       0.000       0.9594        1.000
INTRDVX mean = 5130.60
Predicted values truncated at 1.00000 & 141304.
----------------------------
Proportion of variance (R-squared) explained by tree model: 1.0000

Observed and fitted values are stored in step.fit
Regressor names and coefficients are stored in step.reg
LaTeX code for tree is in step.tex
```

The names of the selected predictor variables can also be read from the output file `step.reg`, which gives, on each line, the terminal node number, lower and upper truncation values, and the names of the variables selected by the stepwise regression model in the node.

7 Quantile regression: CE data

GUIDE can build piecewise-constant and piecewise-linear quantile regression models. First we show how to build a piecewise-constant 0.90-quantile regression model.

7.1 Piecewise constant: one quantile

7.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
```

```
Input your choice: 1
Name of batch input file: quantcon.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: quantcon.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.90
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
```

```
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantcon.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: quantcon.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: quantcon.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantcon.in
```

Contents of `quantcon.out`

```
Quantile regression tree with quantile probability 0.9000
Pruning by cross-validation
DSC file: ce2021reg.dsc
Training sample file: ce2021.txt
```

```
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     s    1.000        1.000                 125  constant
      2  DIRACC_    m                                  2
      3  AGE_REF    s    19.00        87.00
      4  AGE_REF_   m                                  0
      5  AGE2       s    21.00        87.00                1092
      6  AGE2_      m                                  1
      :
     31  FINLWT21   w    1072.       0.9390E+05
      :
    406  INTRDVX    d    1.000       0.1413E+06
      :
    547  WHLFYR     c                                  1     2487
    548  WHLFYR_    m                                  1
    549  FFTAXOWE   s  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   s   -3074.       0.5654E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      376
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
```

```
Number of constant S variables: 7
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 2
Ranks of variables and their 1-df chi-squared values at root node
     1  0.6943E+02   CUTENURE
     2  0.6324E+02   REFGEN
     3  0.5982E+02   RENTEQVX
     4  0.5957E+02   AGE_REF
     5  0.5754E+02   AGE2
     6  0.5689E+02   STOCKX
     7  0.5598E+02   INCNONW1
     8  0.5547E+02   NO_EARNR
     :
   413  0.1064E-02   ECARTKNP
   414  0.4983E-03   FULOILPQ
   415  0.3478E-03   TOTHFARP
   416  0.4489E-04   MAJAPPPQ
   417  0.3034E-04   OTHLNYR

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       48   5.948E+07   4.764E+06   4.889E+06   5.743E+07   7.105E+06
   2       47   5.948E+07   4.764E+06   4.889E+06   5.743E+07   7.105E+06
   3       46   5.946E+07   4.763E+06   4.878E+06   5.743E+07   7.105E+06
   4       45   5.946E+07   4.763E+06   4.879E+06   5.743E+07   7.112E+06
   5       42   5.946E+07   4.763E+06   4.879E+06   5.743E+07   7.112E+06
   6       41   5.941E+07   4.761E+06   4.867E+06   5.749E+07   7.120E+06
   7*      39   5.941E+07   4.761E+06   4.871E+06   5.749E+07   7.122E+06
   8       37   5.953E+07   4.761E+06   4.886E+06   5.772E+07   7.145E+06
   :
  18       26   5.964E+07   4.775E+06   4.803E+06   5.790E+07   6.953E+06
  19       25   5.984E+07   4.773E+06   4.802E+06   5.790E+07   6.960E+06
  20**     24   6.021E+07   4.807E+06   4.653E+06   5.738E+07   6.891E+06
```

```
  21        23   6.171E+07   4.997E+06   4.587E+06   5.720E+07   6.362E+06
  22        21   6.171E+07   4.997E+06   4.587E+06   5.720E+07   6.362E+06
  23        20   6.140E+07   4.984E+06   4.660E+06   5.720E+07   6.656E+06
  24        17   6.087E+07   4.957E+06   4.440E+06   5.720E+07   5.580E+06
  25        15   6.103E+07   4.964E+06   4.385E+06   5.717E+07   5.761E+06
  26        14   6.078E+07   4.980E+06   4.519E+06   5.717E+07   5.861E+06
  27        13   6.109E+07   5.026E+06   4.483E+06   5.769E+07   6.522E+06
  28        12   6.155E+07   5.042E+06   4.334E+06   5.769E+07   6.263E+06
  29        10   6.155E+07   5.042E+06   4.334E+06   5.769E+07   6.263E+06
  30++       9   6.147E+07   5.029E+06   4.341E+06   5.708E+07   6.351E+06
  31         7   6.580E+07   5.409E+06   5.027E+06   6.057E+07   8.897E+06
  32         5   7.073E+07   5.668E+06   6.656E+06   6.916E+07   1.257E+07
  33         4   7.094E+07   5.845E+06   6.872E+06   7.191E+07   1.356E+07
  34         3   7.561E+07   5.755E+06   5.616E+06   6.904E+07   8.622E+06
  35         1   9.107E+07   7.613E+06   5.017E+06   9.297E+07   5.472E+06

0-SE tree based on mean is marked with * and has 39 terminal nodes
0-SE tree based on median is marked with + and has 9 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
+ tree same as ++ tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-quant is quantile of INTRDVX in the node
Cases fit give the number of cases used to fit node
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank    D-quant  variable       variables
          1    2487     2487     1  9.800E+03  CUTENURE
          2     872      872     1  2.500E+04  INC_RANK
          4     664      664     1  1.200E+04  RNTAPYCQ
          8     660      660     1  1.000E+04  INC_RANK
         16T    333      333     1  5.361E+03  -
         17     327      327     1  1.800E+04  TTRANPRP
         34     249      249     1  1.124E+04  AGE2
         68     209      209     1  7.500E+03  TGASMOTC
        136T    207      207     1  7.139E+03  -
        137T      2        2     1  1.071E+05  -
         69T     40       40     1  2.700E+04  -
         35T     78       78     1  3.000E+04  -
          9T      4        4     1  4.800E+04  -
          5    208      208     1  1.071E+05  RETPENPQ
```

```
          10       64       64     1  1.413E+05  PROPTXCQ
          20T      47       47     1  1.413E+05  PROPTXCQ
          21T      17       17     1  1.071E+05  -
          11      144      144     1  3.000E+04  EENTMSCC
          22      121      121     1  1.500E+04  STATE
          44T      17       17     1  4.400E+04  -
          45T     104      104     1  5.600E+03  -
          23T      23       23     1  1.071E+05  -
           3     1615     1615     1  4.200E+03  FFTAXOWE
           6     1374     1374     1  3.000E+03  REF_RACE
          12T      28       28     1  1.500E+04  FLRCVRPQ
          13     1346     1346     1  2.400E+03  STOCKYRX
          26     1341     1341     1  2.400E+03  AGE_REF
          52      990      990     1  1.000E+03  STATE
         104T     388      388     1  3.000E+03  FSSIX
         105T     602      602     1  4.500E+02  OCCUCOD2
          53      351      351     1  8.161E+03  OTHHEXCQ
         106T     176      176     1  4.200E+03  EDUC_REF
         107      175      175     1  1.400E+04  OTHSTYRB
         214T     173      173     1  1.400E+04  -
         215T       2        2     1  1.071E+05  -
          27T       5        5     1  1.413E+05  -
           7      241      241     1  2.123E+04  REF_RACE
          14T       4        4     1  1.413E+05  -
          15      237      237     1  2.000E+04  LIQUDYRB
          30      231      231     1  1.550E+04  AGE2
          60T     187      187     1  6.000E+03  ROYESTX
          61       44       44     1  1.071E+05  INCNONW2
         122       38       38     1  4.200E+04  VNATLGAC
         244T      35       35     1  2.355E+04  -
         245T       3        3     1  1.071E+05  -
         123T       6        6     1  1.071E+05  -
          31T       6        6     1  1.071E+05  -
Warning: tree very large, omitting node numbers in LaTeX file

Number of terminal nodes of final tree: 24
Total number of nodes of final tree: 47
Second best split variable (based on curvature test) at root node is REFGEN

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: CUTENURE = "2", "5"
  Node 2: INC_RANK <= 0.81810030
    Node 4: RNTAPYCQ <= 1158.3333
      Node 8: INC_RANK <= 0.49406705
```

```
        Node 16: INTRDVX sample quantile = 5361.0000
      Node 8: INC_RANK > 0.49406705 or NA
        Node 17: TTRANPRP <= 20.500000
          Node 34: AGE2 <= 76.500000 or NA
            Node 68: TGASMOTC <= 660.50000
              Node 136: INTRDVX sample quantile = 7139.0000
            Node 68: TGASMOTC > 660.50000 or NA
              Node 137: INTRDVX sample quantile = 107121.00
          Node 34: AGE2 > 76.500000
            Node 69: INTRDVX sample quantile = 27000.000
        Node 17: TTRANPRP > 20.500000 or NA
          Node 35: INTRDVX sample quantile = 30000.000
    Node 4: RNTAPYCQ > 1158.3333 or NA
      Node 9: INTRDVX sample quantile = 48000.000
  Node 2: INC_RANK > 0.81810030 or NA
    Node 5: RETPENPQ <= 90.250000
      Node 10: PROPTXCQ <= 875.00000
        Node 20: INTRDVX sample quantile = 141304.00
      Node 10: PROPTXCQ > 875.00000 or NA
        Node 21: INTRDVX sample quantile = 107121.00
    Node 5: RETPENPQ > 90.250000 or NA
      Node 11: EENTMSCC <= 66.666500
        Node 22: STATE = "1", "17", "29", "37", "46"
          Node 44: INTRDVX sample quantile = 44000.000
        Node 22: STATE /= "1", "17", "29", "37", "46"
          Node 45: INTRDVX sample quantile = 5600.0000
      Node 11: EENTMSCC > 66.666500 or NA
        Node 23: INTRDVX sample quantile = 107121.00
Node 1: CUTENURE /= "2", "5"
  Node 3: FFTAXOWE <= 31377.000
    Node 6: REF_RACE = "3"
      Node 12: INTRDVX sample quantile = 15000.000
    Node 6: REF_RACE /= "3"
      Node 13: STOCKYRX <= 875000.00 or NA
        Node 26: AGE_REF <= 64.500000
          Node 52: STATE = "1", "2", "6", "8", "9", "10", "12", "15", "21",
                   "22", "36", "41", "45", "48", "53"
            Node 104: INTRDVX sample quantile = 3000.0000
          Node 52: STATE /= "1", "2", "6", "8", "9", "10", "12", "15", "21",
                   "22", "36", "41", "45", "48", "53"
            Node 105: INTRDVX sample quantile = 450.00000
        Node 26: AGE_REF > 64.500000 or NA
          Node 53: OTHHEXCQ <= 16.666650
            Node 106: INTRDVX sample quantile = 4200.0000
          Node 53: OTHHEXCQ > 16.666650 or NA
            Node 107: OTHSTYRB = NA & OTHS_YRB = "A"
```

```
              Node 214: INTRDVX sample quantile = 14000.000
            Node 107: not (OTHSTYRB = NA & OTHS_YRB = "A")
              Node 215: INTRDVX sample quantile = 107121.00
      Node 13: STOCKYRX > 875000.00
        Node 27: INTRDVX sample quantile = 141304.00
  Node 3: FFTAXOWE > 31377.000 or NA
    Node 7: REF_RACE = "6"
      Node 14: INTRDVX sample quantile = 141304.00
    Node 7: REF_RACE /= "6"
      Node 15: LIQUDYRB = NA & LIQU_YRB = "A"
        Node 30: AGE2 <= 57.500000 or NA
          Node 60: INTRDVX sample quantile = 6000.0000
        Node 30: AGE2 > 57.500000
          Node 61: INCNONW2 = "2"
                  or (INCNONW2 = NA & INCN_NW2 = "A")
            Node 122: VNATLGAC <= 11.000000
              Node 244: INTRDVX sample quantile = 23554.000
            Node 122: VNATLGAC > 11.000000 or NA
              Node 245: INTRDVX sample quantile = 107121.00
          Node 61: INCNONW2 /= "2"
                  & not (INCNONW2 = NA & INCN_NW2 = "A")
            Node 123: INTRDVX sample quantile = 107121.00
      Node 15: not (LIQUDYRB = NA & LIQU_YRB = "A")
        Node 31: INTRDVX sample quantile = 107121.00

 ***************************************************************
 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if CUTENURE = "2", "5"
 CUTENURE mode = "1"
 Predicted quantile = 9800.00
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if INC_RANK <= 0.81810030
```

```
 INC_RANK mean = 0.59625137
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if RNTAPYCQ <= 1158.3333
 RNTAPYCQ mean = 11.380252
 ----------------------------
 :
Node 60: Terminal node
 Predicted quantile = 6000.00
 ----------------------------
Node 61: Intermediate node
 A case goes into Node 122 if INCNONW2 = "2"
    or INCNONW2 = NA & INCN_NW2 =  "A"
 INCN_NW2 mode = "A"
 ----------------------------
Node 122: Intermediate node
 A case goes into Node 244 if VNATLGAC <= 11.000000
 VNATLGAC mean = 3.4919712
 ----------------------------
Node 244: Terminal node
 Predicted quantile = 23554.0
 ----------------------------
Node 245: Terminal node
 Predicted quantile = 107121.
 ----------------------------
Node 123: Terminal node
 Predicted quantile = 107121.
 ----------------------------
Node 31: Terminal node
 Predicted quantile = 107121.
 ----------------------------
 Observed and fitted values are stored in quantcon.fit
 LaTeX code for tree is in quantcon.tex
```

Figure 12 shows the quantile regression tree. The sample size (in *italics*) and 0.90-quantile are given beneath each terminal node.

7.2 Best simple linear

We demonstrate this with a linear 0.90-quantile regression tree.

7.2.1 Input file creation

```
 0. Read the warranty disclaimer
```

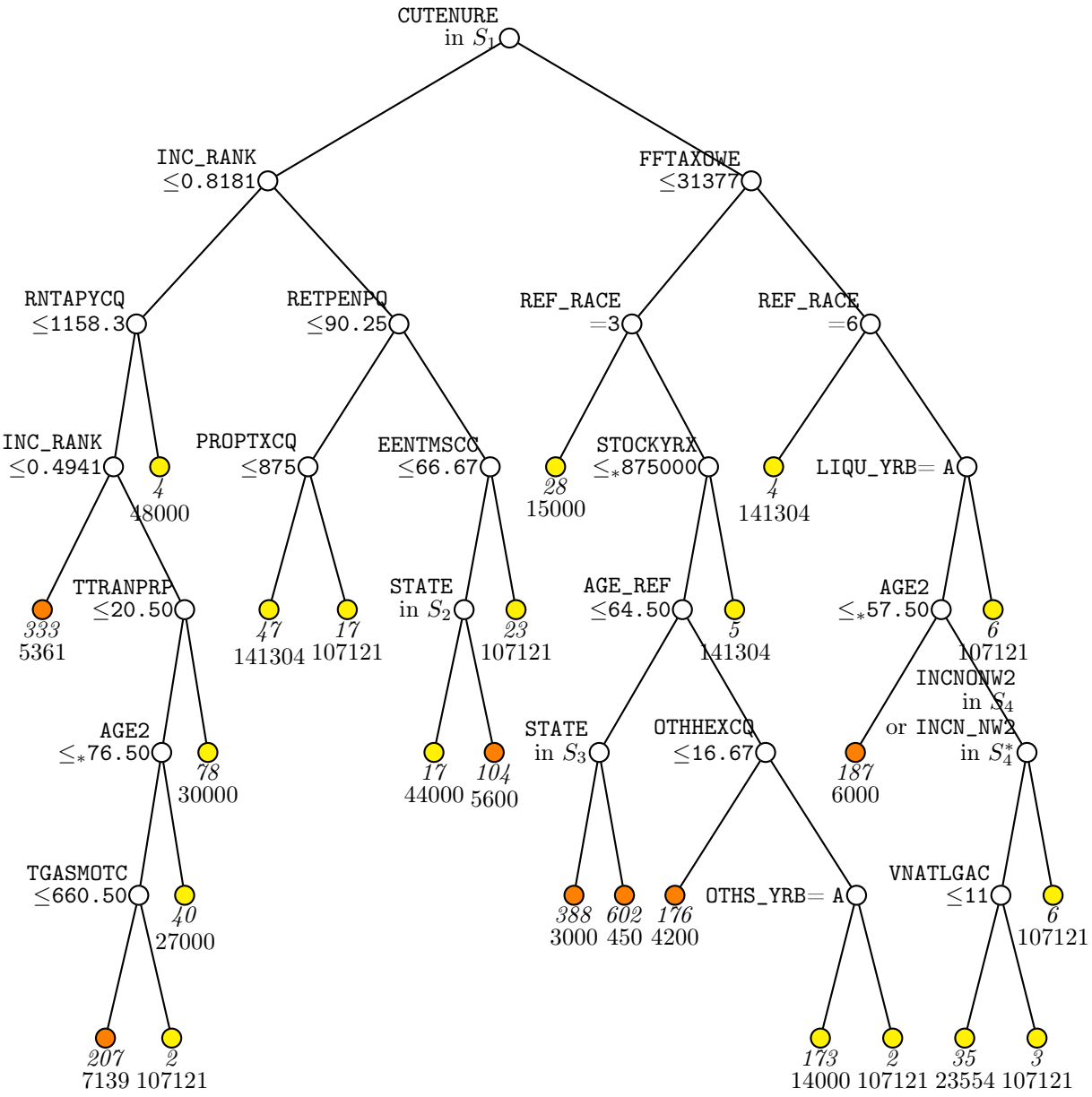

Figure 12: GUIDE v.46.1 0.250-SE piecewise-constant 0.900-quantile regression tree for predicting INTRDVX. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '≤*' stands for '≤ or missing'. S_1 = {2, 5}. S_2 = {1, 17, 29, 37, 46}. S_3 = {1, 2, 6, 8, 9, 10, 12, 15, 21, 22, 36, 41, 45, 48, 53}. S_4 = {2}; S_4^* = {A}. Sample size (in *italics*) and 0.900-quantile of INTRDVX printed below nodes. Terminal nodes with quantiles above and below value of 9800 at root node are painted yellow and orange respectively. Second best split variable at root node is REFGEN.

```
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: quantlin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: quantlin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.90
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
```

```
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      383        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantlin.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: quantlin.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: quantlin.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: quantlin.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantlin.in
```

Contents of `quantlin.out`

```
Quantile regression tree with quantile probability 0.9000
No truncation of predicted values
```

```
Pruning by cross-validation
DSC file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     n   1.0000E+00    1.000                  125  constant
      2  DIRACC_    m                                2
      3  AGE_REF    n    19.00        87.00
      4  AGE_REF_   m                                0
      5  AGE2       n   2.1000E+01    87.00                 1092
      6  AGE2_      m                                1
      :
     31  FINLWT21   w    1072.       0.9390E+05
      :
    406  INTRDVX    d    1.000       0.1413E+06
      :
    547  WHLFYR     c                                1     2487
    548  WHLFYR_    m                                1
    549  FFTAXOWE   n  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   n   -3074.       0.5654E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
```

```
      3965       1478       3965        1      376        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of constant N variables: 7
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 25
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       32   6.034E+07   5.099E+06   4.652E+06   6.263E+07   8.243E+06
   2       30   6.033E+07   5.099E+06   4.648E+06   6.263E+07   8.246E+06
   3       29   6.030E+07   5.099E+06   4.656E+06   6.259E+07   8.243E+06
   :
  19       11   5.915E+07   4.948E+06   3.769E+06   6.056E+07   4.819E+06
  20**     10   5.872E+07   4.905E+06   3.882E+06   5.921E+07   5.297E+06
  21        9   6.000E+07   4.936E+06   4.054E+06   6.137E+07   6.008E+06
  22++      8   6.037E+07   4.486E+06   3.640E+06   5.969E+07   6.153E+06
  23        7   6.087E+07   4.695E+06   4.010E+06   6.498E+07   6.932E+06
  24        6   6.420E+07   5.043E+06   5.316E+06   6.498E+07   7.856E+06
  25        4   7.040E+07   5.888E+06   5.816E+06   7.232E+07   9.671E+06
  26        3   7.352E+07   5.991E+06   6.546E+06   6.965E+07   1.071E+07
  27        1   8.361E+07   6.413E+06   3.673E+06   8.229E+07   5.259E+06

0-SE tree based on mean is marked with * and has 10 terminal nodes
0-SE tree based on median is marked with + and has 10 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as + tree
** tree same as -- tree
* tree same as ** tree
* tree same as -- tree
```

```
Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-quant is quantile of INTRDVX in the node
Cases fit give the number of cases used to fit node
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank    D-quant  variable       variables
          1     2487     1939     2  9.800E+03  MRTINTPQ
          2     1371     1371     2  1.400E+04  FINCBTAX
          4      916      916     2  7.000E+03  STOCKX
          8T     874      874     2  6.000E+03  STATE
          9T      42       42     2  3.000E+04  -
          5      455      320     2  4.500E+04  RETPENPQ
         10      117      117     2  1.071E+05  RETSURV
         20T      67       20     2  1.071E+05  -
         21T      50       50     2  1.413E+05  -
         11      338      338     2  1.500E+04  RENTEQVX
         22T     305      305     2  6.000E+03  RETSURVX
         23T      33       33     2  1.071E+05  -
          3     1116       19     2  5.000E+03  STOCKYRX
          6     1077       19     2  4.400E+03  FFTAXOWE
         12T     930        7     2  3.000E+03  REFGEN
         13      147      147     2  2.123E+04  FDXMAPCQ
         26T      72       72     2  1.071E+05  RENTEQVX
         27T      75       75     2  1.000E+04  -
          7T      39       39     2  1.071E+05  -

Number of terminal nodes of final tree: 10
Total number of nodes of final tree: 19
Second best split variable (based on curvature test) at root node is EMRTPNOP

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: MRTINTPQ <= 1.5000000
  Node 2: FINCBTAX <= 95076.000
    Node 4: STOCKX <= 125000.00  or STOCKX = NA & STOCKX_ = "A"
      Node 8: INTRDVX sample quantile = 6000.0000
    Node 4: not (STOCKX <= 125000.00 or STOCKX = NA & STOCKX_ = "A")
      Node 9: INTRDVX sample quantile = 30000.000
  Node 2: FINCBTAX > 95076.000 or NA
    Node 5: RETPENPQ <= 571.83330
      Node 10: RETSURV = "1"
        Node 20: INTRDVX sample quantile = 107121.00
```

```
      Node 10: RETSURV /= "1"
        Node 21: INTRDVX sample quantile = 141304.00
    Node 5: RETPENPQ > 571.83330 or NA
      Node 11: RENTEQVX <= 3073.0000 or NA
        Node 22: INTRDVX sample quantile = 6000.0000
      Node 11: RENTEQVX > 3073.0000
        Node 23: INTRDVX sample quantile = 107121.00
Node 1: MRTINTPQ > 1.5000000 or NA
  Node 3: STOCKYRX <= 135000.00  or STOCKYRX = NA & STOC_YRX = "A"
    Node 6: FFTAXOWE <= 40686.000
      Node 12: INTRDVX sample quantile = 3000.0000
    Node 6: FFTAXOWE > 40686.000 or NA
      Node 13: FDXMAPCQ <= 238.33335
        Node 26: INTRDVX sample quantile = 107121.00
      Node 13: FDXMAPCQ > 238.33335 or NA
        Node 27: INTRDVX sample quantile = 10000.000
  Node 3: not (STOCKYRX <= 135000.00 or STOCKYRX = NA & STOC_YRX = "A")
    Node 7: INTRDVX sample quantile = 107121.00

*****************************************************************
Predictor means below are weighted means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if MRTINTPQ <= 1.5000000
 MRTINTPQ mean = 619.33599
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     -5793.
 RENTEQVX      9.896        65.00        2252.        6302.
 If regressor has missing values, predicted quantile = 3000.00
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if FINCBTAX <= 95076.000
 FINCBTAX mean = 89898.898
```

```
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if STOCKX <= 125000.00 or STOCKX_ = "A"
 STOCKX mean = 324021.34
 ----------------------------
Node 8: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant      1969.
 FRRETIRM     0.2330        0.000       0.1471E+05   0.7378E+05
 If regressor has missing values, predicted quantile = 6000.00
 ----------------------------
Node 9: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     -6275.
 INC_RANK     0.6613E+05   0.9591E-01   0.4320       0.7292
 If regressor has missing values, predicted quantile = 30000.0
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if RETPENPQ <= 571.83330
 RETPENPQ mean = 2829.5709
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if RETSURV = "1"
 RETSURV mode = "1"
 ----------------------------
Node 20: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     0.1071E+06
 CREDITX      -14.01        0.000        833.7        7648.
 If regressor has missing values, predicted quantile = 40000.0
 ----------------------------
Node 21: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -0.6379E+05
 INCLASS2     0.3418E+05    5.000        5.601        6.000
 If regressor has missing values, predicted quantile = 141304.
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if RENTEQVX <= 3073.0000 or NA
 RENTEQVX mean = 2523.9099
 ----------------------------
Node 22: Terminal node
```

```
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant      4200.
 PROPTXCQ      32.49        0.000        227.1        3168.
 If regressor has missing values, predicted quantile = 6000.00
 ----------------------------
Node 23: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     0.3182E+05
 TALCBEVC      376.5        0.000        46.91        448.0
 If regressor has missing values, predicted quantile = 107121.
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if STOCKYRX <= 135000.00 or STOC_YRX = "A"
 STOCKYRX mean = 277744.06
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if FFTAXOWE <= 40686.000
 FFTAXOWE mean = 17090.349
 ----------------------------
Node 12: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     -4692.
 WHLFYRBX      3.195        1500.       0.1386E+05   0.3500E+05
 If regressor has missing values, predicted quantile = 3000.00
 ----------------------------
Node 13: Intermediate node
 A case goes into Node 26 if FDXMAPCQ <= 238.33335
 FDXMAPCQ mean = 510.41115
 ----------------------------
Node 26: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     0.1550E+05
 OWNDWECQ      28.30        0.000        450.6        7125.
 If regressor has missing values, predicted quantile = 107121.
 ----------------------------
Node 27: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant      5000.
 MISCPQ        78.55        0.000        121.7        1412.
 If regressor has missing values, predicted quantile = 10000.0
 ----------------------------
```

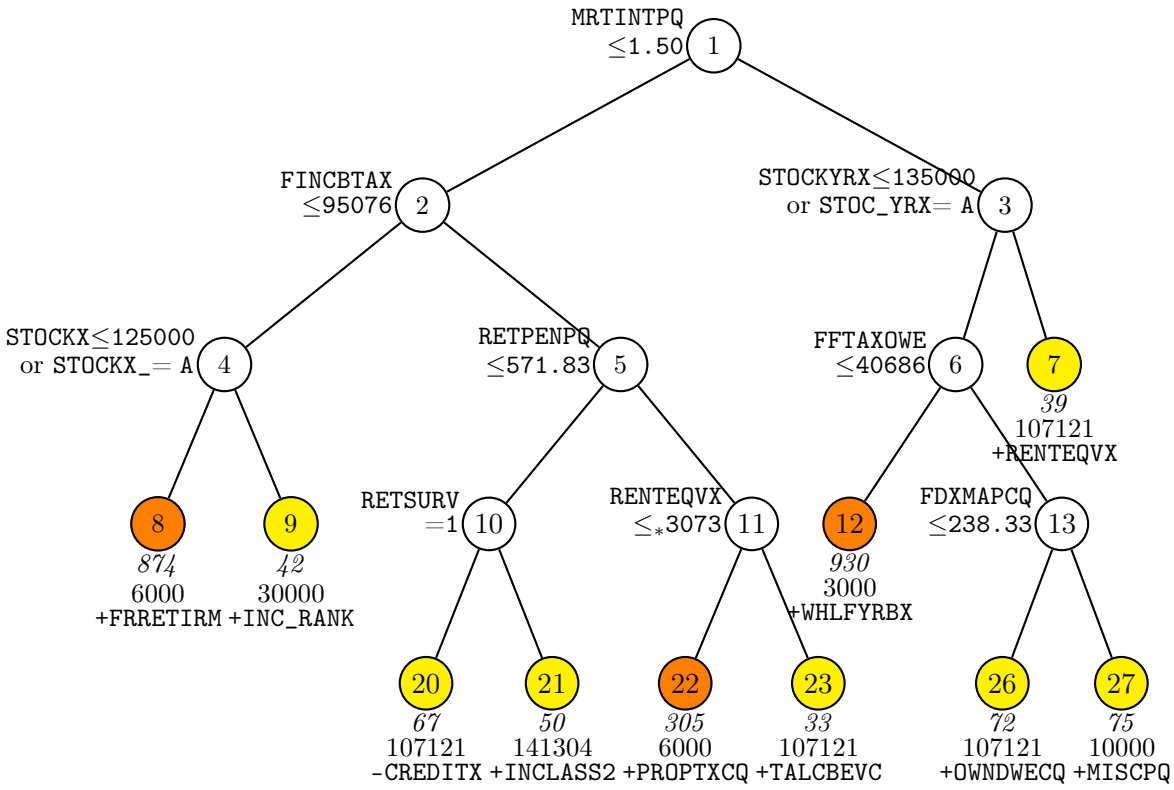

Figure 13: GUIDE v.46.1 0.250-SE piecewise simple linear 0.900-quantile regression tree (constant fitted to incomplete cases in terminal nodes) for predicting INTRDVX. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Sample size (in *italics*), 0.900-quantile of INTRDVX, and sign and name of best regressor printed below nodes. Terminal nodes with quantiles above and below value of 9800 at root node are painted yellow and orange respectively. Second best split variable at root node is EMRTPNOP.

```
Node 7: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -0.2434E+05
 RENTEQVX      26.32        925.0        3338.        6294.
 If regressor has missing values, predicted quantile = 107121.
 ----------------------------
 Observed and fitted values are stored in quantlin.fit
 Regressor names and coefficients are stored in quantlin.reg
 LaTeX code for tree is in quantlin.tex
```

Figure 13 shows the 0.90-quantile simple linear regression tree.

7.3 Two quantiles: checking variance heterogeneity

Checking variance homogeneity in the residuals is a standard practice in fitting regression models. Here we show how GUIDE can do this by constructing a quantile regression tree models for the 25th and 75th quantiles simultaneously.

7.3.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: twoquant.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: twoquant.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1): 2
Input 1st quantile probability ([0.00:1.00], <cr>=0.25):
Input 2nd quantile probability ([0.00:1.00], <cr>=0.75):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 0 to get tree with fixed no. of nodes, 1 to prune by CV, 2 for no pruning ([0:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
```

```
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Default number of cross-validations:           10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.25):
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
Default max. number of split levels: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): twoquant.tex
Choose a color for the terminal nodes:
```

```
(1) white
(2) lightgray
(3) aqua
(4) skyblue
(5) lime
(6) yellow
(7) red
(8) mauve
(9) green
(10) orange
(11) cyan
Input your choice ([1:11], <cr>=8):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: twoquant.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: twoquant.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < twoquant.in
```

7.3.2 Output file

```
Dual-quantile regression tree with 0.2500 and 0.7500 quantiles
Pruning by cross-validation
DSC file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
```

```
Missing values found among non-categorical variables
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     s    1.000        1.000                  125  constant
      2  DIRACC_    m                                 2
      3  AGE_REF    s    19.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       s    21.00        87.00                 1092
      6  AGE2_      m                                 1
      :
     31  FINLWT21   w    1072.       0.9390E+05
      :
    406  INTRDVX    d    1.000       0.1413E+06
      :
    549  FFTAXOWE   s  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   s   -3074.       0.5654E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      376
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of constant S variables: 7
Number of cases used for training: 2487
Number of cases excluded due to 0 W or missing D variable: 1478

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction and linear splits; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
```

```
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 2
Ranks of variables and their 1-df chi-squared values at root node
     1  0.1744E+03   STATE
     2  0.1192E+03   FINCBTAX
     3  0.1135E+03   INC_RANK
     :
   415  0.5618E-03   TFEESADC
   416  0.4974E-03   INC_HRS2
   417  0.4468E-03   OTHLNYRX

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       68   1.078E+08   7.583E+06   5.419E+06   1.098E+08   8.082E+06
   2       66   1.078E+08   7.583E+06   5.419E+06   1.098E+08   8.082E+06
   :
  11       51   1.078E+08   7.584E+06   5.417E+06   1.097E+08   8.074E+06
  12+      50   1.078E+08   7.584E+06   5.417E+06   1.097E+08   8.074E+06
  13       49   1.078E+08   7.584E+06   5.417E+06   1.097E+08   8.078E+06
   :
  31       13   1.077E+08   7.636E+06   5.331E+06   1.118E+08   7.930E+06
  32++     11   1.059E+08   7.642E+06   5.547E+06   1.107E+08   7.791E+06
  33       10   1.065E+08   7.705E+06   5.541E+06   1.128E+08   8.121E+06
  34--      7   1.058E+08   7.671E+06   5.251E+06   1.128E+08   7.954E+06
  35        5   1.084E+08   8.069E+06   5.362E+06   1.134E+08   7.313E+06
  36**      4   1.074E+08   7.820E+06   4.994E+06   1.128E+08   6.979E+06
  37        1   1.164E+08   9.292E+06   5.964E+06   1.190E+08   7.350E+06

0-SE tree based on mean is marked with * and has 7 terminal nodes
0-SE tree based on median is marked with + and has 50 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Column labeled 'Split variable' gives median if node is terminal
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank     median  variable       variables
          1     2487     2487     1  1.500E+01  STATE
          2T     420      420     1  5.000E+00  2.500E+02  FFTAXOWE
```

```
          3     2067     2067     1  2.000E+01  INC_RANK
          6T    1554     1554     1  1.500E+01  1.500E+03  CUTENURE
          7      513      513     1  1.200E+02  INCNONW2
         14T     417      417     1  1.000E+02  3.000E+03  STATE
         15T      96       96     1  4.000E+02  1.071E+05  OCCUCOD1

 Number of terminal nodes of final tree: 4
 Total number of nodes of final tree: 7
 Second best split variable (based on curvature test) at root node is FINCBTAX

 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: STATE = "19", "24", "31", "40", "45", "49", "51", "54"
   Node 2: INTRDVX sample quantiles = 5.0000000, 250.00000
 Node 1: STATE /= "19", "24", "31", "40", "45", "49", "51", "54"
   Node 3: INC_RANK <= 0.87960485
     Node 6: INTRDVX sample quantiles = 15.000000, 1500.0000
   Node 3: INC_RANK > 0.87960485 or NA
     Node 7: INCNONW2 = NA & INCN_NW2 = "A"
       Node 14: INTRDVX sample quantiles = 100.00000, 3000.0000
     Node 7: not (INCNONW2 = NA & INCN_NW2 = "A")
       Node 15: INTRDVX sample quantiles = 400.00000, 107121.00

 ************************************************************
 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if STATE = "19", "24", "31", "40", "45", "49", "51", "54"
 STATE mode = "6"
 Sample 0.250-quantile, 0.750-quantile, and median:
     1.5000E+01     1.9300E+03     1.7400E+02
 ----------------------------
Node 2: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
```

```
    5.0000E+00     2.5000E+02     2.0000E+01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if INC_RANK <= 0.87960485
 INC_RANK mean = 0.65419245
 ----------------------------
Node 6: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    1.5000E+01     1.5000E+03     1.5000E+02
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if INCNONW2 = NA & INCN_NW2 = "A"
 INCN_NW2 mode = "A"
 ----------------------------
Node 14: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    1.0000E+02     3.0000E+03     5.0000E+02
 ----------------------------
Node 15: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    4.0000E+02     1.0712E+05     9.5000E+03
 ----------------------------
 Observed and fitted values are stored in twoquant.fit
 LaTeX code for tree is in twoquant.tex
```

Figure 14 shows the tree. Beneath each terminal node are three numbers. The first (in *italics*) is the node sample size. The other two are the sample 0.75 and 0.25-quantiles in the node. The large differences in inter-quartile range in the nodes indicates substantial variance heterogeneity.

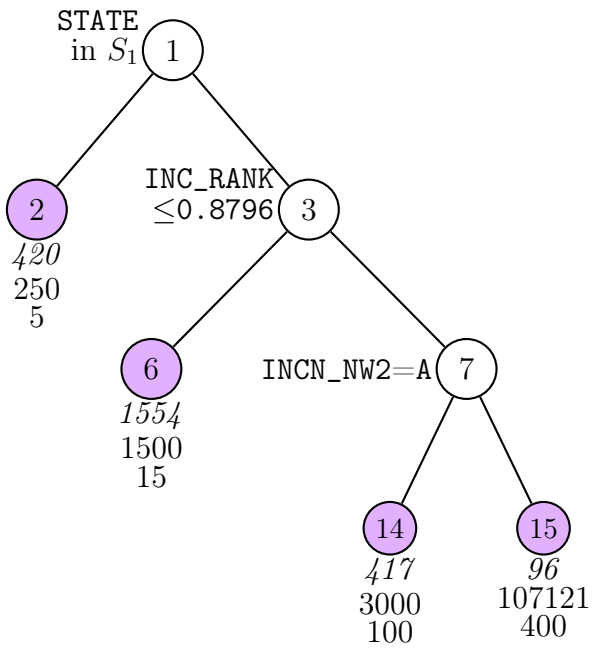

Figure 14: GUIDE v.46.1 0.250-SE piecewise-constant 0.250 and 0.750-quantile regression tree for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. $S_1 = \{$`19`, `24`, `31`, `40`, `45`, `49`, `51`, `54`$\}$. Sample size (in *italics*) and sample 0.750 and 0.250-quantiles of `INTRDVX` printed below nodes. Second best split variable at root node is `FINCBTAX`.

8 Periodic variables: NHTSA data

Periodic variables that have a cyclic property, such as angular measurements, hour of day, day of week, and month of year, can be designated as `P` variables in the DSC file. There can be multiple `P` variables in the same data set. Unlike the other types of variables, each line in the DSC file containing a `P` variable must have the value of its period (e.g., 360 for angular measurements, 24 for hour of day, 7 for day of week, and 12 for month of year) immediately after `P` on the same line. GUIDE does not allow `P` variables to have missing-value flag (`M`) variables.

The National Highway Traffic Safety Administration (NHTSA) has been conducting vehicle crash tests since 1972. Data from 3310 crash tests are in the file `nhtsadatam.txt` (see www-nrd.nhtsa.dot.gov/database/veh/ for more information). Variable `HIC` (head injury criterion) is a measure of severity of head injury. Experts believe that HIC $>$ 999 is life threatening. Table 5 gives the definitions of the variables appearing in the models below. Besides missing values, there are many variables with illogical values (such as negative values for diameter). To identify these values, we adopt the strategy in the CE data of creating a missing-value flag variable for each variable having illogical values, with the flags being `A`, `B`, and `D` for validly missing, illogical, and valid response, respectively. The data also contain some angular variables (with periods of 360 degrees and for which 0 degrees indicates straight-ahead or head-on) that are defined as `P`.

8.1 Regression

The DSC file for regression is `nhtsadsc.txt`; its contents follow.

```
nhtsadatam.txt
NA
2
1 BARRIG c
2 BARSHP b
3 BARANG p 360
4 BARDIA n
5 OCCWT n
6 OCCWT_ m
7 DUMSIZ c
8 HH n
9 HH_ m
10 HW n
11 HW_ m
12 HR n
```

```
13 HR_ m
14 HS n
15 HS_ m
16 CD n
17 CD_ m
18 CS n
19 CS_ m
20 AD n
21 AD_ m
22 HD n
23 HD_ m
24 KD n
25 KD_ m
26 HB n
27 HB_ m
28 NB n
29 NB_ m
30 CB n
31 CB_ m
32 KB n
33 SEPOSN c
34 HIC d
35 TKSURF c
36 TKCOND c
37 CLSSPD n
38 CLSSPD_ m
39 IMPANG p 360
40 OFFSET n
41 IMPPNT n
42 MAKED c
43 MODELD c
44 YEAR n
45 BODY c
46 ENGINE c
47 ENGDSP n
48 ENGDSP_ m
49 TRANSM c
50 VEHTWT n
51 VEHTWT_ m
52 CURBWT n
53 WHLBAS n
54 WHLBAS_ m
55 VEHLEN n
56 VEHLEN_ m
57 VEHWID n
58 VEHWID_ m
```

```
59 VEHCG n
60 VEHCG_ m
61 COLMEC c
62 BX1 n
63 BX1_ m
64 BX2 n
65 BX2_ m
66 BX3 n
67 BX3_ m
68 BX4 n
69 BX4_ m
70 BX5 n
71 BX5_ m
72 BX6 n
73 BX6_ m
74 BX7 n
75 BX7_ m
76 BX8 n
77 BX8_ m
78 BX9 n
79 BX9_ m
80 BX10 n
81 BX10_ m
82 BX11 n
83 BX11_ m
84 BX12 n
85 BX12_ m
86 BX13 n
87 BX13_ m
88 BX14 n
89 BX14_ m
90 BX15 n
91 BX15_ m
92 BX16 n
93 BX16_ m
94 BX17 n
95 BX17_ m
96 BX18 n
97 BX18_ m
98 BX19 n
99 BX19_ m
100 BX20 n
101 BX20_ m
102 BX21 n
103 BX21_ m
104 VEHSPD n
```

Table 5: Some variable definitions for NHTSA data

Variable	Meaning
BARSHP	barrier shape (21 values)
BX2	distance from rear surface of vehicle to front of engine (mm)
BX5	distance from rear surface of vehicle to upper leading edge of left door (mm)
BX8	distance from rear surface of vehicle to upper trailing edge of right door (mm)
BX12	distance from rear surface of vehicle to bottom of a post of right side (mm)
COLMEC	steering column collapse mechanism (9 values)
ENGDSP	engine displacement (liters)
IMPANG	impact angle (clockwise with 0 degrees being straight ahead)
OCCAGE	dummy occupant age
PDOF	principal direction of force (degrees)
TRANSM	transmission type (9 values)
VEHTWT	vehicle test weight (kg)
VEHSPD	vehicle speed (km/h)
VEHWID	vehicle width (mm)
WHLBAS	wheel base (mm)
YEAR	vehicle model year (1972–2017)

```
105 VEHSPD_ m
106 CRBANG p 360
107 PDOF p 360
108 CARANG p 360
109 VEHOR p 360
110 RSTFRT c
111 HIC2 x
112 estHIC2 x
```

We show the results of fitting a piecewise-linear regression tree here.

8.1.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: lin.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: lin.out
```

```
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: nhtsadsc.txt
Reading DSC file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 11 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
```

```
      3310         34       3310        5       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       11        0
No weight variable in data file
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: lin.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: lin.r
Input rank of top variable to split root node ([1:65], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in
```

8.1.2 Results

```
Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
DSC file: nhtsadsc.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 0.0500
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
```

```
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables

Summary information for training sample of size 3276 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      1  BARRIG     c                                  3
      2  BARSHP     c                                 21
      3  BARANG     p    0.000        330.0          360       14
      4  BARDIA     n   1.9100E+02    1000.                   2807
      :
     33  SEPOSN     c                                  5       81
     34  HIC        d    0.000       0.1225E+05
     35  TKSURF     c                                  5       80
      :
    101  BX20_      m                                  2
    102  BX21       n   5.4000E+01   0.1085E+05                964
    103  BX21_      m                                  2
    104  VEHSPD     n   3.0000E-01    99.10                     6
    105  VEHSPD_    m                                  2
    106  CRBANG     p    0.000        315.0          360       24
    107  PDOF       p    0.000        345.0          360       23
    108  CARANG     p    0.000        99.00          360      991
    109  VEHOR      p    0.000        90.00          360      995
    110  RSTFRT     c                                  3

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        5       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       11        0
No weight variable in data file
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
```

```
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 33
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       77   4.136E+05   7.785E+04   7.934E+04   3.550E+05   8.140E+04
   2       75   4.136E+05   7.785E+04   7.934E+04   3.550E+05   8.140E+04
   :
  46       14   4.327E+05   8.921E+04   1.012E+05   3.435E+05   7.641E+04
  47+      12   4.317E+05   8.921E+04   1.012E+05   3.435E+05   7.611E+04
  48       11   3.852E+05   7.886E+04   9.474E+04   3.439E+05   4.904E+04
  49       10   3.872E+05   7.917E+04   9.452E+04   3.439E+05   4.783E+04
  50**      6   3.558E+05   6.574E+04   5.578E+04   3.448E+05   3.927E+04
  51        1   3.750E+05   6.843E+04   5.388E+04   4.032E+05   4.736E+04

0-SE tree based on mean is marked with * and has 6 terminal nodes
0-SE tree based on median is marked with + and has 12 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree
* tree same as ** tree
* tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of HIC in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split         Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable      variables
          1     3276     3272     2  5.127E+02  3.743E+05  0.1007 BARSHP  -YEAR
          2T      40       32     2  1.606E+02  1.337E+04  0.3122 -  +HW
          3     3236     3232     2  5.170E+02  3.749E+05  0.1066 BARSHP  -YEAR
          6T      34       31     2  2.749E+02  5.098E+04  0.2716 -  -VEHSPD
          7     3202     3199     2  5.196E+02  3.771E+05  0.1080 BARSHP  -YEAR
```

```
         14      320      206     2  3.440E+02  4.708E+05  0.4316 BX4  -IMPPNT
         28T      65        7     2  6.285E+02  4.322E+05  0.7203 -  -BARDIA
         29T     255      171     2  2.715E+02  2.911E+05  0.5381 SEPOSN  -IMPPNT
         15     2882     2879     2  5.391E+02  3.315E+05  0.1147 CLSSPD  -YEAR
         30T    1292        9     2  4.440E+02  5.638E+05  0.0267 HS  -CB
         31T    1590     1590     2  6.164E+02  1.141E+05  0.4156 TRANSM  -YEAR

Number of terminal nodes of final tree: 6
Total number of nodes of final tree: 11
Second best split variable (based on curvature test) at root node is IMPANG

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: BARSHP = "488", "EOL", "GRL", "MBR", "OTH", "ROR"
  Node 2: HIC-mean = 160.57500
Node 1: BARSHP /= "488", "EOL", "GRL", "MBR", "OTH", "ROR"
  Node 3: BARSHP = "128", "SGN", "UNK", "US1"
    Node 6: HIC-mean = 274.94118
  Node 3: BARSHP /= "128", "SGN", "UNK", "US1"
    Node 7: BARSHP = "134", "EOB", "FAB", "IAT", "LUM"
      Node 14: BX4 <= 2536.5000  or BX4 = NA & BX4_ = "A"
        Node 28: HIC-mean = 628.50769
      Node 14: not (BX4 <= 2536.5000 or BX4 = NA & BX4_ = "A")
        Node 29: HIC-mean = 271.47843
    Node 7: BARSHP /= "134", "EOB", "FAB", "IAT", "LUM"
      Node 15: CLSSPD <= 55.450000
        Node 30: HIC-mean = 443.96749
      Node 15: CLSSPD > 55.450000 or NA
        Node 31: HIC-mean = 616.36541

*****************************************************************
Predictor means below are means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.
```

```
Node 1: Intermediate node
 A case goes into Node 2 if BARSHP = "488", "EOL", "GRL", "MBR", "OTH", "ROR"
 BARSHP mode = "LCB"
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3893E+05    19.40       0.1488E-14
 YEAR         -19.21       -19.14       0.9685E-15    1972.        2000.        2017.
 If regressor has missing values, predicted value = 471.00000
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -84.23      -0.7127       0.4815
 HW           0.5093        2.369       0.2447E-01    414.0        540.9        787.0
 If regressor has missing values, predicted value = 37.875000
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if BARSHP = "128", "SGN", "UNK", "US1"
 BARSHP mode = "LCB"
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      767.7        4.103       0.3026E-03
 VEHSPD       -7.404       -2.561       0.1590E-01    40.20        63.02        98.00
 If regressor has missing values, predicted value = 5.3333333
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if BARSHP = "134", "EOB", "FAB", "IAT", "LUM"
 BARSHP mode = "LCB"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if BX4 <= 2536.5000 or BX4_ = "A"
 BX4 mean = 3273.8683
 ----------------------------
Node 28: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.5577E+06    3.041       0.2873E-01
 BARDIA       -2557.       -3.027       0.2917E-01    216.0        217.1        218.0
 If regressor has missing values, predicted value = 411.89655
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 29: Terminal node
```

```
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      145.6        3.044       0.2708E-02
 IMPPNT       -20.71       -15.03       0.000       -203.0       -5.936       0.000
 If regressor has missing values, predicted value = 277.47619
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 15: Intermediate node
 A case goes into Node 30 if CLSSPD <= 55.450000
 CLSSPD mean = 49.475944
 ----------------------------
Node 30: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1004E+05    14.75       0.1575E-05
 CB           -9.554       -13.47       0.2915E-05    537.0        948.6        1000.
 If regressor has missing values, predicted value = 440.23461
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 31: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.5474E+05    33.99       0.1594E-15
 YEAR         -27.07       -33.60       0.000        1974.        1999.        2017.
 If regressor has missing values, predicted value = 616.36541
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.2578

 Observed and fitted values are stored in lin.fit
 Regressor names and coefficients are stored in lin.reg
 LaTeX code for tree is in lin.tex
```

The piecewise-linear regression tree is shown in Figure 15.

8.2 Classification

8.2.1 Univariate splits

Below are the results of fitting a classification tree with univariate splits to `HIC2` which takes value `1` if `HIC` > 999 and value `0` otherwise. The DSC file is `nhtsahic2.dsc`.

```
 Classification tree
 Pruning by cross-validation
```

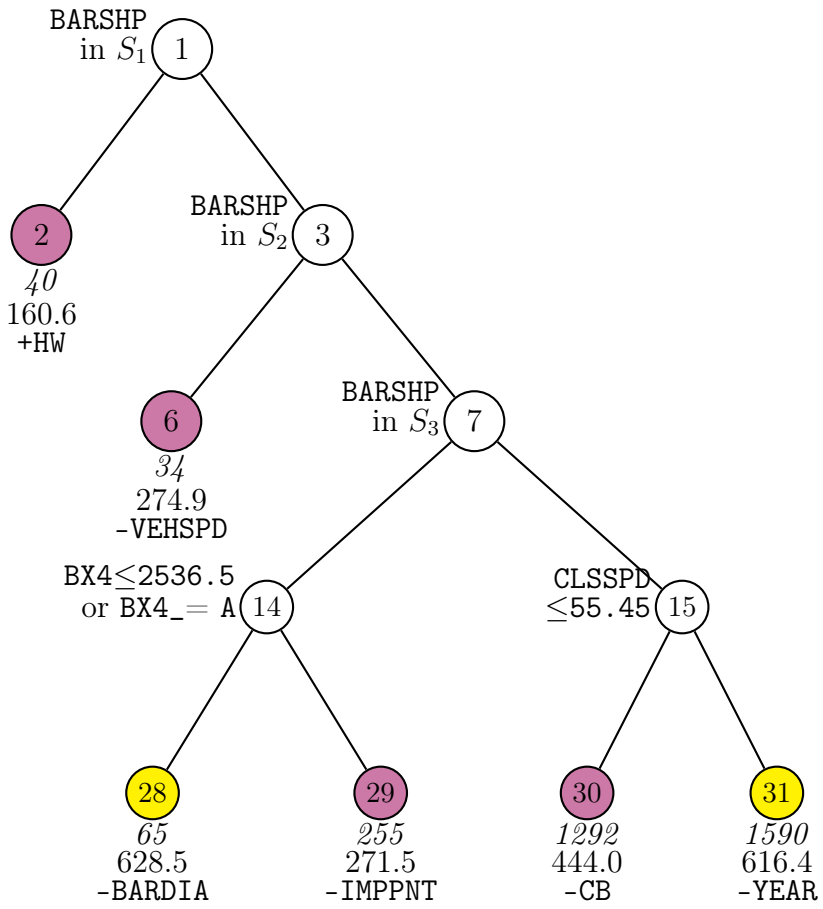

Figure 15: GUIDE v.46.1 0.250-SE piecewise simple linear least-squares regression tree (constant fitted to incomplete cases in terminal nodes) for predicting HIC. At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {488, EOL, GRL, MBR, OTH, ROR}. S_2 = {128, SGN, UNK, US1}. S_3 = {134, EOB, FAB, IAT, LUM}. Sample size (in *italics*), mean of HIC, and name of regressor (with sign of slope) printed below nodes. Terminal nodes with means above and below value of 512.7 at root node are painted yellow and purple respectively. Second best split variable at root node is IMPANG.

```
DSC file: nhtsahic2.dsc
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
Warning: B variables changed to C
D variable is HIC2
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable HIC2:
Class  #Cases     Proportion
0        2999     0.91544567
1         277     0.08455433

Summary information for training sample of size 3276 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                 3
      2  BARSHP     c                                21
      3  BARANG     p    0.000        330.0          360       14
      4  BARDIA     s    191.0        1000.                   2807
      5  OCCWT      s    72.00        83.00                   3265
      6  OCCWT_     m                                  2
      :
    106  CRBANG     p    0.000        315.0          360       24
    107  PDOF       p    0.000        345.0          360       23
    108  CARANG     p    0.000        99.00          360      991
    109  VEHOR      p    0.000        90.00          360      995
    110  RSTFRT     c                                  3
    111  HIC2       d                                  2

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
```

```
       3310          34        3310         5         0         0        48
     #P-var    #M-var    #B-var    #C-var    #I-var
          6        42         0        11         0
 Number of cases used for training: 3276
 Number of cases excluded due to 0 W or missing D variable: 34

 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Simple node models
 Estimated priors
 Misclassification cost matrix from file costmatrix.txt:
Predicted     True class
              0          1
0             0.000E+00  1.000E+01
1             1.000E+00  0.000E+00

Class         Altered prior
0             0.5198
1             0.4802
 Univariate split highest priority
 Interaction and linear splits 2nd and 3rd priorities
 Split values for N and S variables based on exhaustive search
 Max number of interaction splits: 100
 Maximum number of split levels: 12
 Minimum node sample size: 3
 Ranks of variables and their 1-df chi-squared values at root node
      1  0.4697E+03   COLMEC
      2  0.3441E+03   YEAR
      3  0.3169E+03   RSTFRT
      4  0.2902E+03   HS
      :
     64  0.8370E+00   BARDIA
     65  0.1921E-01   IMPANG

 Size and CV mean cost and SE of subtrees:
  Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
    1       51   3.480E-01   2.721E-02   1.915E-02   3.582E-01   1.993E-02
    2       50   3.480E-01   2.721E-02   1.915E-02   3.582E-01   1.993E-02
    :
   23       23   2.576E-01   2.047E-02   2.144E-02   2.470E-01   3.640E-02
   24*      22   2.561E-01   2.046E-02   2.365E-02   2.424E-01   3.951E-02
   25       20   2.582E-01   2.047E-02   2.371E-02   2.500E-01   3.756E-02
   26       11   2.616E-01   2.069E-02   2.143E-02   2.530E-01   3.319E-02
   27**     10   2.592E-01   2.027E-02   1.991E-02   2.546E-01   2.753E-02
```

```
  28        9   2.637E-01   2.069E-02   2.090E-02   2.473E-01   3.482E-02
  29++      8   2.625E-01   2.049E-02   2.150E-02   2.443E-01   3.918E-02
  30        7   2.643E-01   2.029E-02   2.509E-02   2.550E-01   4.066E-02
  31        6   2.842E-01   2.076E-02   2.261E-02   2.855E-01   3.314E-02
  32        5   2.824E-01   1.951E-02   2.107E-02   2.886E-01   3.030E-02
  33        4   3.092E-01   2.163E-02   2.774E-02   2.805E-01   2.103E-02
  34        2   3.700E-01   2.630E-02   2.480E-02   3.572E-01   2.321E-02
  35        1   8.455E-01   4.861E-02   4.656E-03   8.537E-01   7.280E-03

0-SE tree based on mean is marked with * and has 22 terminal nodes
0-SE tree based on median is marked with + and has 22 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
       Node    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1     3276     3276    0             8.369E-01 COLMEC
          2     2596     2596    0             2.707E-01 RSTFRT
          4      602      602    0             6.576E-01 BARSHP
          8T      64       64    1             5.877E-01 BODY
          9T     538      538    0             2.576E-01 CRBANG
          5     1994     1994    0             1.539E-01 IMPANG
         10       74       74    1             8.292E-01 YEAR
         20T      50       50    1             7.522E-01 HW
         21T      24       24    0             0.000E+00 -
         11T    1920     1920    0             9.794E-02 IMPPNT
          3      680      680    1             6.899E-01 RSTFRT
          6      598      598    1             6.554E-01 BARSHP
         12T     510      510    1             6.074E-01 TKSURF
         13       88       88    0             5.623E-01 CURBWT
         26T      81       81    0             2.444E-01 YEAR +VEHWID
         27T       7        7    1             5.656E-01 BX19
          7       82       82    0             4.828E-01 PDOF
         14T      79       79    0             2.506E-01 TKSURF
         15T       3        3    1             3.299E-01 -

Number of terminal nodes of final tree: 10
Total number of nodes of final tree: 19
```

```
 Second best split variable (based on curvature test) at root node is YEAR

 Classification tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: COLMEC = "BWU", "NA", "NAP", "UNK"
   Node 2: RSTFRT = "0"
     Node 4: BARSHP = "LCB"
       Node 8: 1
     Node 4: BARSHP /= "LCB"
       Node 9: 0
   Node 2: RSTFRT /= "0"
     Node 5: IMPANG in [269, 287]
       Node 10: YEAR <= 2010.5000
         Node 20: 1
       Node 10: YEAR > 2010.5000 or NA
         Node 21: 0
     Node 5: IMPANG not in [269, 287] or NA
       Node 11: 0
 Node 1: COLMEC /= "BWU", "NA", "NAP", "UNK"
   Node 3: RSTFRT = "0"
     Node 6: BARSHP = "FLB", "LCB", "LUM"
       Node 12: 1
     Node 6: BARSHP /= "FLB", "LCB", "LUM"
       Node 13: CURBWT <= 1529.0000 or NA
         Node 26: 0
       Node 13: CURBWT > 1529.0000
         Node 27: 1
   Node 3: RSTFRT /= "0"
     Node 7: PDOF in [-1, 1]
       Node 14: 0
     Node 7: PDOF not in [-1, 1] or NA
       Node 15: 1

 ****************************************************************
 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if COLMEC = "BWU", "NA", "NAP", "UNK"
 COLMEC mode = "UNK"
 Class      Number   Posterior
 0            2999  0.9154E+00
 1             277  0.8455E-01
 Misclassification cost =                2770
 Predicted class is 0
 ----------------------------
```

```
Node 2: Intermediate node
 A case goes into Node 4 if RSTFRT = "0"
 RSTFRT mode = "1"
 Class      Number   Posterior
 0            2525  0.9727E+00
 1              71  0.2735E-01
 Misclassification cost =                710
 Predicted class is 0
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if BARSHP = "LCB"
 BARSHP mode = "POL"
 Class      Number   Posterior
 0             562  0.9336E+00
 1              40  0.6645E-01
 Misclassification cost =                400
 Predicted class is 0
 ----------------------------
:
Node 7: Intermediate node
 A case goes into Node 14 if PDOF in [-1, 1]
 PDOF mean = 5.1219512
 Class      Number   Posterior
 0              78  0.9512E+00
 1               4  0.4878E-01
 Misclassification cost =                40
 Predicted class is 0
 ----------------------------
Node 14: Terminal node
 Class      Number   Posterior
 0              77  0.9747E+00
 1               2  0.2532E-01
 Misclassification cost =                20
 Predicted class is 0
 ----------------------------
Node 15: Terminal node
 Class      Number   Posterior
 0               1  0.3333E+00
 1               2  0.6667E+00
 Misclassification cost =                1
 Predicted class is 1
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              0         1
```

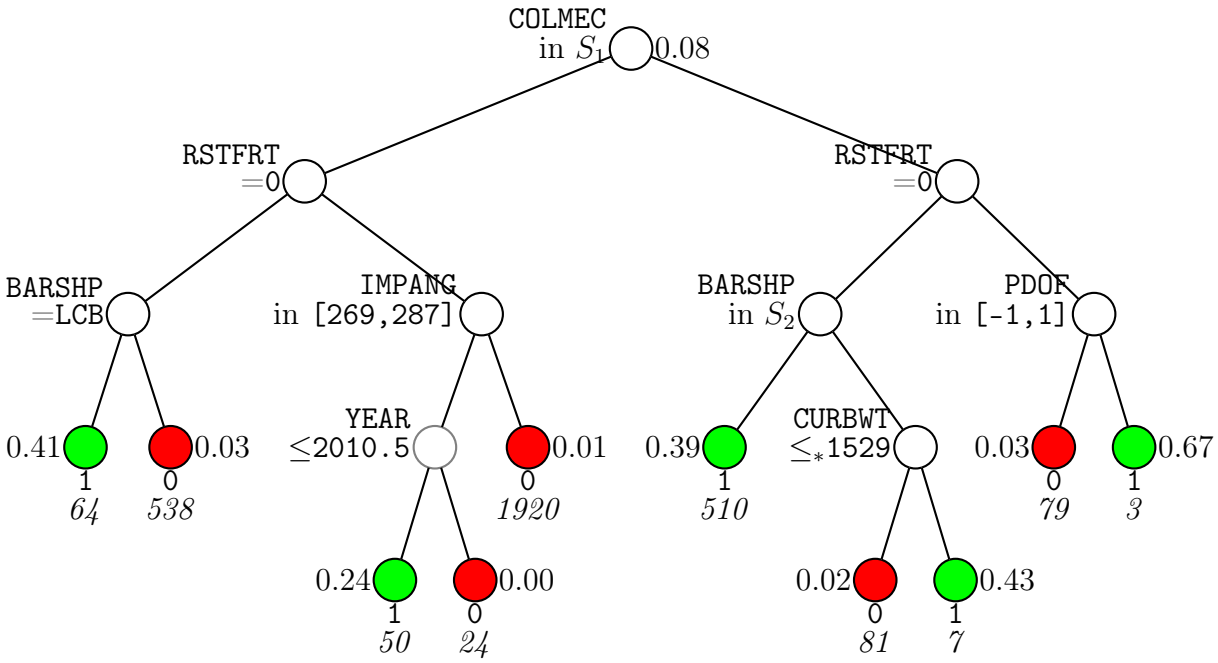

Figure 16: GUIDE v.45.2 0.250-SE classification tree for predicting HIC2 using estimated priors and specified misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. $S_1 = \{$BWU, NA, NAP, UNK$\}$. $S_2 = \{$FLB, LCB, LUM$\}$. Predicted class and sample size (in *italics*) printed below each terminal node; class sample proportion for HIC2 = 1 beside node. Second best split variable at root node is YEAR.

```
0               2605        37
1                394       240
Total           2999       277

Number of cases used for tree construction: 3276
Number misclassified: 431
Resubstitution estimate of mean misclassification cost: 0.23321123

Observed and fitted values are stored in class10.fit
LaTeX code for tree is in class10.tex
```

8.2.2 Linear splits

To use P variables for classification, we need to convert their values to sines and cosines. The data after conversion are in the file sincosdata.txt. The created variables have names beginning with "s." (for sine) and "c." (for cosine). For example, the values of the variable s.BARANG are $\sin(2\pi \times \texttt{BARANG}/360)$. The contents of the DSC file sincosdscHIC2.txt for this task are given below.

```
sincosdata.txt
NA
2
1 BARRIG b
2 BARSHP b
3 BARANG p 360
4 BARDIA n
5 OCCWT n
6 OCCWT_ m
7 DUMSIZ b
8 HH n
9 HH_ m
10 HW n
11 HW_ m
12 HR n
13 HR_ m
14 HS n
15 HS_ m
16 CD n
17 CD_ m
18 CS n
19 CS_ m
20 AD n
21 AD_ m
22 HD n
23 HD_ m
24 KD n
25 KD_ m
26 HB n
27 HB_ m
28 NB n
29 NB_ m
30 CB n
31 CB_ m
32 KB n
33 SEPOSN b
34 HIC x
35 TKSURF b
```

```
36 TKCOND b
37 CLSSPD n
38 CLSSPD_ m
39 IMPANG p 360
40 OFFSET n
41 IMPPNT n
42 YEAR n
43 BODY b
44 ENGINE b
45 ENGDSP n
46 ENGDSP_ m
47 TRANSM b
48 VEHTWT n
49 VEHTWT_ m
50 CURBWT n
51 WHLBAS n
52 WHLBAS_ m
53 VEHLEN n
54 VEHLEN_ m
55 VEHWID n
56 VEHWID_ m
57 VEHCG n
58 VEHCG_ m
59 COLMEC b
60 BX1 n
61 BX1_ m
62 BX2 n
63 BX2_ m
64 BX3 n
65 BX3_ m
66 BX4 n
67 BX4_ m
68 BX5 n
69 BX5_ m
70 BX6 n
71 BX6_ m
72 BX7 n
73 BX7_ m
74 BX8 n
75 BX8_ m
76 BX9 n
77 BX9_ m
78 BX10 n
79 BX10_ m
80 BX11 n
81 BX11_ m
```

```
82 BX12 n
83 BX12_ m
84 BX13 n
85 BX13_ m
86 BX14 n
87 BX14_ m
88 BX15 n
89 BX15_ m
90 BX16 n
91 BX16_ m
92 BX17 n
93 BX17_ m
94 BX18 n
95 BX18_ m
96 BX19 n
97 BX19_ m
98  BX20 n
99  BX20_ m
100 BX21 n
101 BX21_ m
102 VEHSPD n
103 VEHSPD_ m
104 CRBANG p 360
105 PDOF p 360
106 CARANG p 360
107 VEHOR p 360
108 RSTFRT b
109 HIC2 d
110 HIC3 x
111 s.BARANG n
112 c.BARANG n
113 s.IMPANG n
114 c.IMPANG n
115 s.CRBANG n
116 c.CRBANG n
117 s.PDOF n
118 c.PDOF n
119 s.CARANG n
120 c.CARANG n
121 s.VEHOR n
122 c.VEHOR n
```

8.2.3 Results

The output using LDA splits and LDA node models follow.

```
Classification tree
Linear discriminant splits and LDA node models
Pruning by cross-validation
DSC file: sincosdscHIC2.txt
Training sample file: sincosdata.txt
Missing value code: NA
Records in data file start on line 2
Warning: P variables excluded from LDA and QDA models
D variable is HIC2
M variables changed to B
Number of records in data file: 3310
Length of longest entry in data file: 20
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable HIC2:
Class  #Cases     Proportion
0        2999     0.91544567
1         277     0.08455433

Summary information for training sample of size 3276 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                    #Codes/
                                                    Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  BARRIG          b                                 3
      2  BARSHP          b                                21
      4  BARDIA          n   1.9100E+02    1000.                   2807
      :
    108  RSTFRT          b                                 3
    109  HIC2            d                                 2
    111  s.BARANG        n  -5.0000E-01    1.000                     14
    112  c.BARANG        n   6.1232E-17    1.000                     14
    113  s.IMPANG        n  -1.0000E+00    1.000                      4
    114  c.IMPANG        n  -1.0000E+00    1.000                      4
    115  s.CRBANG        n  -1.0000E+00    1.000                     24
    116  c.CRBANG        n  -9.3969E-01    1.000                     24
    117  s.PDOF          n  -1.0000E+00    1.000                     23
    118  c.PDOF          n  -1.0000E+00    1.000                     23
    119  s.CARANG        n   0.0000E+00   0.9877                    991
```

```
   120  c.CARANG        n  -1.5643E-01    1.000                   991
   121  s.VEHOR         n   0.0000E+00    1.000                   995
   122  c.VEHOR         n   6.1232E-17    1.000                   995
 ================== Constructed variables ==================
   123  BARRIG.R        f    0.000        1.000
   124  BARRIG.U        f    0.000        1.000
   125  BARSHP.134      f    0.000        1.000
   126  BARSHP.488      f    0.000        1.000
     :
   354  BX21.NA         f    0.000        1.000
   355  VEHSPD.NA       f    0.000        1.000
   356  s.BARANG.NA     f    0.000        1.000
   357  c.BARANG.NA     f    0.000        1.000
   358  s.IMPANG.NA     f    0.000        1.000
   359  c.IMPANG.NA     f    0.000        1.000
   360  s.CRBANG.NA     f    0.000        1.000
   361  c.CRBANG.NA     f    0.000        1.000
   362  s.PDOF.NA       f    0.000        1.000
   363  c.PDOF.NA       f    0.000        1.000
   364  s.CARANG.NA     f    0.000        1.000
   365  c.CARANG.NA     f    0.000        1.000
   366  s.VEHOR.NA      f    0.000        1.000
   367  c.VEHOR.NA      f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        8       60      244        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0       53        0        0
Number of constant F variables: 1
Number of dummy variables (excl. missing-value indicators) created: 185
Number of missing-value indicators created: 60
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Estimated priors
Unit misclassification costs
Max number of splits: 100
Maximum number of split levels: 15
Minimum node sample size: 29
Size and CV mean cost and SE of subtrees:
```

Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
1	23	9.646E-02	5.158E-03	3.529E-03	9.604E-02	2.221E-03
2	20	9.646E-02	5.158E-03	3.529E-03	9.604E-02	2.221E-03
3	19	9.188E-02	5.047E-03	2.306E-03	9.007E-02	3.784E-03
4	18	9.096E-02	5.024E-03	1.852E-03	9.146E-02	2.930E-03
5	15	9.005E-02	5.001E-03	2.758E-03	9.007E-02	3.124E-03
6	14	8.974E-02	4.994E-03	3.018E-03	9.007E-02	3.178E-03
7	9	8.791E-02	4.947E-03	2.959E-03	8.855E-02	3.878E-03
8**	6	8.333E-02	4.829E-03	2.240E-03	8.396E-02	4.838E-03
9++	2	8.730E-02	4.932E-03	3.248E-03	8.397E-02	3.987E-03
10	1	9.707E-02	5.172E-03	3.415E-03	9.756E-02	4.450E-03

```
0-SE tree based on mean is marked with * and has 6 terminal nodes
0-SE tree based on median is marked with + and has 6 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as + tree
** tree same as -- tree
* tree same as ** tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of cases in node
       Node    Total  Misclassification
      label    cases  cost
          1     3276  8.489E-02
          2     3183  8.271E-02
          4     2473  9.925E-02
          8      148  3.611E-01
        16T       82  2.293E-01
        17T       66  2.999E-01
          9     2325  8.173E-02
        18T      253  0.000E+00
        19T     2072  8.741E-02
         5T      710  2.509E-02
         3T       93  0.000E+00

Number of terminal nodes of final tree: 6
Total number of nodes of final tree: 11

Classification tree with linear discriminant splits and LDA node models
```

```
 Node 1: D1 <= 0.44816965E-01
   Node 2: D2 <= 1.8490250
     Node 4: D4 <= -0.36315036E-02
       Node 8: D8 <= -0.64715070
         Node 16: 19/82 misclassified
       Node 8: D8 > -0.64715070
         Node 17: 20/66 misclassified
     Node 4: D4 > -0.36315036E-02
       Node 9: D9 <= -2.9160169
         Node 18: 0/253 misclassified
       Node 9: D9 > -2.9160169
         Node 19: 183/2072 misclassified
   Node 2: D2 > 1.8490250
     Node 5: 18/710 misclassified
 Node 1: D1 > 0.44816965E-01
   Node 3: 0/93 misclassified

 ****************************************************************
 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if D1 <= 0.44816965E-01 where the coefficients of D1
 are the coefficients of the 1st crimcoord below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 HH               -0.26082179E-04  -0.27612200E-04    360.68138
 HW                0.20159169E-04   0.32088560E-04    601.14091
 HR               -0.16556541E-04  -0.15234824E-04    223.20486
 :
 HD.NA            -0.17015129E-01  -0.59529266E-02   0.60439560E-01
 IMPPNT.NA         0.59105812E-02   0.24570562E-02   0.51678877
 CURBWT.NA        -0.15752245E-02  -0.12535642E-02   0.87118437
 BX19.NA          -0.74896802E-02   0.24272996E-02   0.16483516
 BX20.NA           0.10360932E-01  -0.42393704E-02   0.16452991
 s.CARANG.NA      -0.22888994E-02   0.96496985E-03   0.30250305
 c.CARANG.NA      -0.22888994E-02   0.96496985E-03   0.30250305
 s.VEHOR.NA        0.16756736E-02  -0.11686222E-02   0.30372405
 c.VEHOR.NA        0.16756736E-02  -0.11686222E-02   0.30372405

 Class      Crim1Mean   Crim1Variance      Prior
 0       0.291610E-01    0.230109E-03     0.91545
 1      -0.202517E-02    0.230109E-03     0.08455
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if D2 <= 1.8490250 where the coefficients of D2
 are the coefficients of the 1st crimcoord below:
```

```
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 HH               -0.18283492E-04   0.37090012E-04    361.65711
 HW                0.97458529E-05  -0.28266264E-04    599.65925
 HR               -0.11262926E-04   0.56739314E-04    222.45743
 :
 IMPPNT.NA         0.70009329E-02   0.18940052E-02   0.52466227
 CURBWT.NA         0.14010492E-02  -0.28062937E-02   0.87998743
 BX1.NA            0.62704176E-02   0.43320642E-02   0.15519950
 BX19.NA          -0.74606039E-02   0.24567888E-01   0.15739868
 BX20.NA           0.55646220E-02  -0.31285166E-01   0.15708451
 s.CARANG.NA      -0.34155275E-02   0.18553556E-02   0.30788564
 c.CARANG.NA      -0.34155275E-02   0.18553556E-02   0.30788564
 s.VEHOR.NA        0.42417812E-02  -0.44439240E-02   0.30914232
 c.VEHOR.NA        0.42417812E-02  -0.44439240E-02   0.30914232

 Class      Crim1Mean   Crim1Variance      Prior
 0        1.83870        0.236742E-03     0.91298
 1        1.80745        0.236742E-03     0.08702
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if D4 <= -0.36315036E-02 where the coefficients of D4
 are the coefficients of the 1st crimcoord below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 HW                0.68509332E-05  -0.23218602E-04    584.71009
 HR               -0.19941354E-04  -0.52771962E-04    217.38686
 HS                0.50748455E-04  -0.10019163E-03    297.49170
 CS               -0.42544670E-04  -0.87227140E-05    304.67124
 :
 RSTFRT.1          0.15745298E-01   0.55253049E-02   0.54346947
 HS.NA            -0.23738869E-02   0.45846082E-02   0.73999191E-01
 HD.NA            -0.16121582E-02  -0.84382252E-02   0.73594824E-01
 IMPPNT.NA         0.30091921E-02  -0.45152172E-02   0.58592802
 CURBWT.NA        -0.68622555E-02  -0.29935730E-02   0.91872220
 BX19.NA          -0.13671335E-01   0.18293443E-01   0.16215123
 BX20.NA           0.18542804E-01  -0.21064303E-01   0.16174687

 Class      Crim1Mean   Crim1Variance      Prior
 0       0.348710E-01    0.307940E-03     0.89001
 1       0.344502E-02    0.307940E-03     0.10999
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if D8 <= -0.64715070 where the coefficients of D8
 are the coefficients of the 1st crimcoord below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 CD               -0.11617182E-02   0.11154176E-02    558.12687
 CS.NA             0.20856652       0.21722404       0.81081081E-01
```

```
 Class      Crim1Mean   Crim1Variance      Prior
 0      -0.603613        0.628197E-02     0.41892
 1      -0.651562        0.628197E-02     0.58108
 ----------------------------
Node 16: Terminal node
 Cases classified by LDA on 1st crimcoord with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 TKSURF.DOS        0.51266023      -0.50000000       0.24390244E-01
 ENGINE.S6IF       0.51266023       0.50000000       0.24390244E-01

 Class      Crim1Mean   Crim1Variance      Prior
 0       0.891583E-01    0.108556E-01     0.28049
 1        0.00000        0.108556E-01     0.71951
 ----------------------------
Node 17: Terminal node
 Cases classified by LDA on 1st crimcoord with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 BARDIA           -0.32814153       0.51703301        216.66667
 TRANSM.MF         0.25421953       0.16134364       0.21212121

 Class      Crim1Mean   Crim1Variance      Prior
 0       -71.0825        0.133512E-01     0.59091
 1       -70.9870        0.133512E-01     0.40909
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if D9 <= -2.9160169 where the coefficients of D9
 are the coefficients of the 1st crimcoord below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 HW                0.88247264E-05   0.91092030E-04    588.62545
 HR                0.17950110E-04  -0.68764149E-05    220.03499
 HS               -0.57206862E-04   0.72666912E-04    302.03050
 :
 IMPPNT.NA        -0.96667873E-02   0.19714749E-01   0.56387097
 BX1.NA           -0.16071609E-01   0.95530151E-02   0.15741935
 BX19.NA          -0.57664540E-02   0.11290297E-02   0.16043011
 BX20.NA           0.17985659E-01   0.22360619E-03   0.16000000

 Class      Crim1Mean   Crim1Variance      Prior
 0       -2.89873        0.380789E-03     0.92000
 1       -2.87276        0.380789E-03     0.08000
 ----------------------------
Node 18: Terminal node
 Cases classified by LDA on 1st crimcoord with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 DUMSIZ.UN         0.61621885E-01  -0.17497988       0.39525692E-02
```

```
 HH_.B             0.24110964      -0.22997355       0.11857708E-01
 HW_.D            -0.72232482E-02  -0.39995400E-01   0.98418972
 HR_.D            -0.49885955E-01   0.11998620       0.98814229
 :
 ENGDSP.NA        -0.25563576E-01   0.80589233E-15   0.15810277E-01
 WHLBAS.NA         0.18959178E-01  -0.14998275E-01   0.79051383E-02
 VEHLEN.NA         0.18959178E-01  -0.14998275E-01   0.79051383E-02
 VEHWID.NA        -0.29613193E-01   0.36394333E-15   0.15810277E-01
 VEHCG.NA         -0.14013121       0.70661542E-15   0.15810277E-01

 Class      Crim1Mean   Crim1Variance      Prior
 0       0.450076        0.398406E-02     0.99209
 1       0.450076        0.398406E-02     0.00791
 ----------------------------
Node 19: Terminal node
 Cases classified by LDA on 1st crimcoord with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 HW               -0.12968685E-04   0.28863139E-04    581.90505
 HR               -0.20887869E-04  -0.12800171E-03    217.46098
 HS                0.59987332E-04   0.31142132E-03    295.26527
 CS               -0.18317599E-04  -0.12019715E-03    307.01234
 :
 BX20_.D           0.25809426E-01  -0.91565862E-02   0.83928571
 IMPPNT.NA         0.89226448E-02   0.52268840E-02   0.62113900
 BX1.NA            0.26774011E-01  -0.43044841E-02   0.15781853
 BX19.NA           0.54807095E-02  -0.78635445E-02   0.16119691
 BX20.NA          -0.25809426E-01   0.91565862E-02   0.16071429

 Class      Crim1Mean   Crim1Variance      Prior
 0        3.37049         0.434208E-03     0.91120
 1        3.34592         0.434208E-03     0.08880
 ----------------------------
Node 5: Terminal node
 Cases classified by LDA on 1st crimcoord with coefs below:
 Variable         Crimcoord1 coef  Crimcoord2 coef   Variable mean
 ENGINE.4CIF       0.23089070      -0.24340008       0.12676056E-01
 ENGINE.OTHR       0.26092209       0.24158366       0.11267606E-01

 Class      Crim1Mean   Crim1Variance      Prior
 0       0.521075E-02    0.135158E-02     0.99296
 1       0.983626E-01    0.135158E-02     0.00704
 ----------------------------
Node 3: Terminal node
 Pure node
 All cases classified as "0"
 ----------------------------
```

```
Classification matrix for training sample:
Predicted      True class
class              0         1
0               2956       199
1                 43        78
Total           2999       277

Number of cases used for tree construction: 3276
Number misclassified: 242
Resubstitution estimate of mean misclassification cost: 0.73870574E-01

Observed and fitted values are stored in lda.fit
LaTeX code for tree is in lda.tex
R code is stored in lda.r
Discriminant coefficients are stored in lda.coefs
R function for plotting data are stored in ldaplots.r
Plot data are stored in ldadata.txt
```

The LDA classification tree is shown in Figure 17 and the data plots in Figure 18.

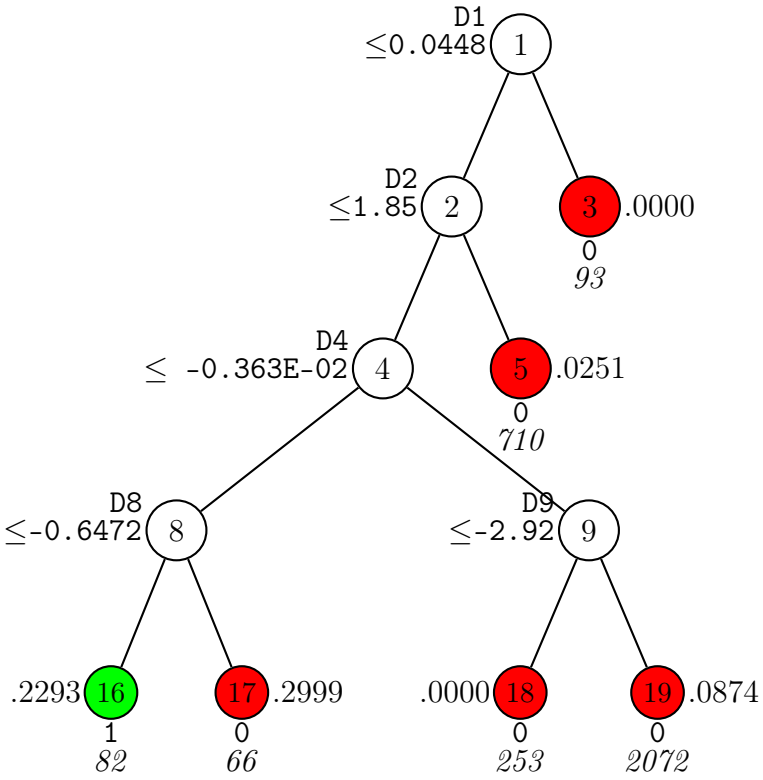

Figure 17: GUIDE v.46.1 0.250-SE classification tree for predicting `HIC2` using linear discriminant splits and linear discriminant node models. At each split, an observation goes to the left branch if and only if the condition is satisfied. Class minimizing misclassification cost and sample size (in *italics*) printed below each terminal node; mean misclassification cost beside node.

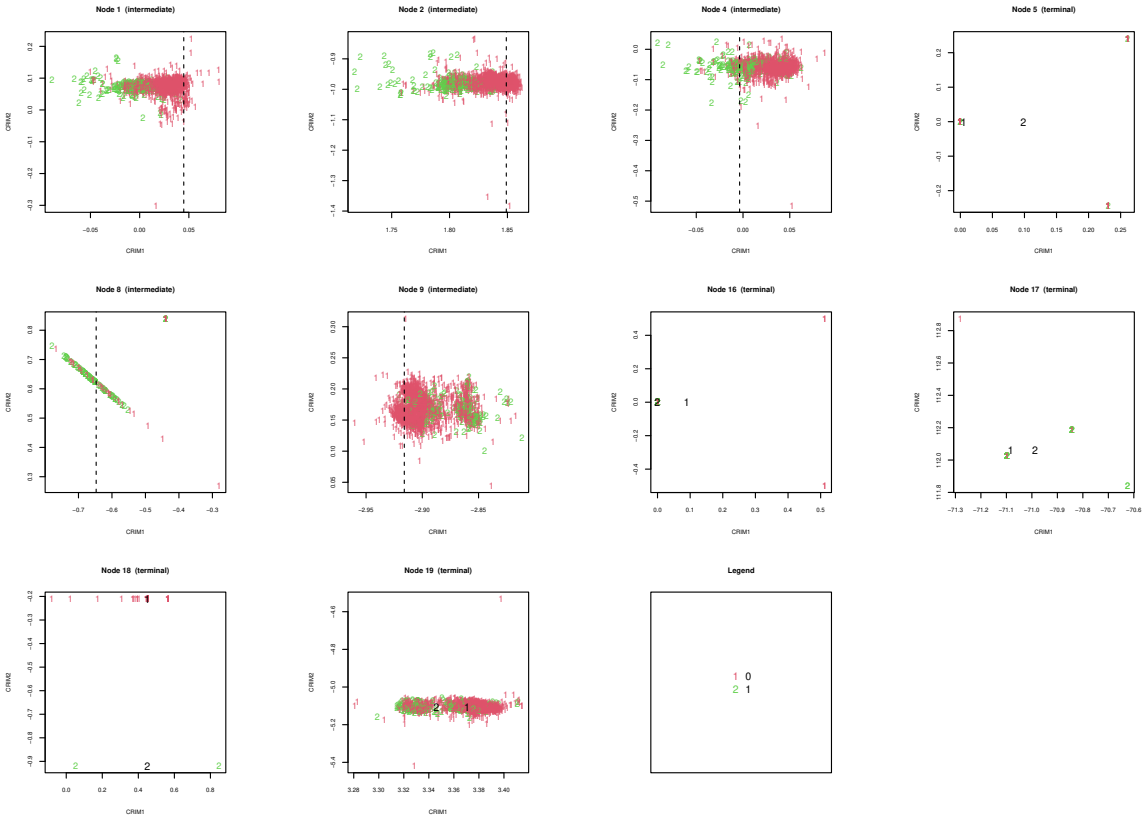

Figure 18: Data and splits in nodes of tree in Figure 17.

9 Poisson regression: solder data

We use a data set on printed circuit board soldering to show how GUIDE fits Poisson regression models. The data were analyzed in Chambers and Hastie (1992) and are given in `solder.dat`. The DSC file `solder.dsc` uses the `b` descriptor for the 5 categorical variables:

```
solder.dat
NA
1
1, skips, d
2, opening, b
3, solder, b
4, mask, b
5, padtype, b
6, panel, b
```

9.1 Piecewise-constant: solder data

9.1.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: cons.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: cons.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 3
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (recommended for prediction)
 Choose 2 for best simple polynomial in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
 1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: solder.dsc
 Reading DSC file ...
```

```
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
Warning: B variables changed to C
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 478
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        5        0
No offset variable in data file.
Number of cases used for training: 720
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): cons.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: cons.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: cons.r
Input rank of top variable to split root node ([1:5], <cr>=1):
Input file is created!
```

The tree is shown in Figure 19, which is quite large. One way to reduce the size of the tree is to fit a more complex Poisson regression model in each node.

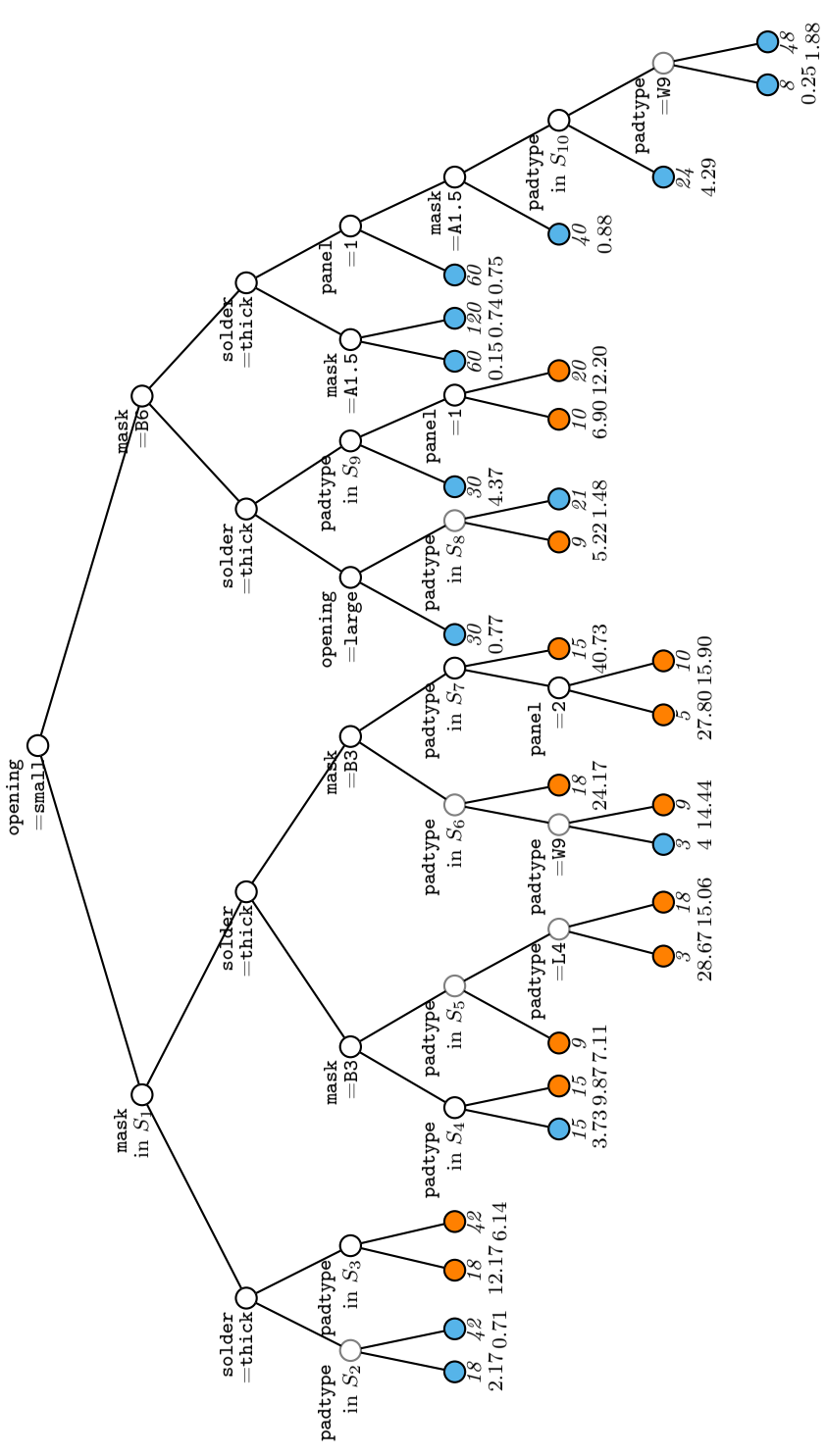

Figure 19: GUIDE v.46.1 0.250-SE piecewise-constant Poisson regression tree for predicting `skips`. At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {`A1.5`, `A3`}. S_2 = {`D7`, `L4`, `L8`}. S_3 = {`D4`, `D7`, `L4`}. S_4 = {`D6`, `L6`, `L9`, `W4`, `W9`}. S_5 = {`L6`, `L9`, `W9`}. S_6 = {`L6`, `L7`, `L9`, `W9`}. S_7 = {`L6`, `L7`, `L8`, `L9`, `W9`}. S_8 = {`D7`, `L4`, `L8`}. S_9 = {`D6`, `L6`, `L7`, `L9`, `W9`}. S_{10} = {`D4`, `D7`, `L4`}. Splits at nodes drawn with gray circles are not statistically significant. Sample size (in *italics*) and mean of `skips` printed below nodes. Terminal nodes with means above and below value of 4.97 at root node are painted orange and skyblue respectively. Second best split variable at root node is `mask`.

9.2 Multiple linear: solder data

Now we construct a tree where each node is fitted with a Poisson model containing only the main effects. This is where the "B" descriptor in `solder.dsc` is for.

9.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: mul.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: mul.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 3
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: solder.dsc
Reading DSC file ...
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 478
GUIDE will try to create the variables in the description file.
```

```
If it is unsuccessful, please create the columns yourself...
Creating dummy variables ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0       17        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        5        0        0
Number of dummy variables created: 17
No offset variable in data file.
Number of cases used for training: 720
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Default max. number of split levels: 8
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): mul.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: mul.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: mul.r
Input rank of top variable to split root node ([1:22], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mul.in
```

9.2.2 Contents of `mul.out`

```
Poisson regression tree
No truncation of predicted values
Pruning by cross-validation
DSC file: solder.dsc
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
D variable is skips
Piecewise linear model
Number of records in data file: 720
Length of longest entry in data file: 6
```

```
Number of cases with positive D values: 478

Summary information for training sample of size 720
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
z=offset variable
                                                    #Codes/
                                                    Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  skips           d    0.000        48.00
      2  opening         b                                3
      3  solder          b                                2
      4  mask            b                                4
      5  padtype         b                               10
      6  panel           b                                3
  ================== Constructed variables ==================
      7  opening.medium  f    0.000        1.000
      8  opening.small   f    0.000        1.000
      9  solder.thin     f    0.000        1.000
     10  mask.A3         f    0.000        1.000
     11  mask.B3         f    0.000        1.000
     12  mask.B6         f    0.000        1.000
     13  padtype.D6      f    0.000        1.000
     14  padtype.D7      f    0.000        1.000
     15  padtype.L4      f    0.000        1.000
     16  padtype.L6      f    0.000        1.000
     17  padtype.L7      f    0.000        1.000
     18  padtype.L8      f    0.000        1.000
     19  padtype.L9      f    0.000        1.000
     20  padtype.W4      f    0.000        1.000
     21  padtype.W9      f    0.000        1.000
     22  panel.2         f    0.000        1.000
     23  panel.3         f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0       17        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        5        0        0
Number of dummy variables created: 17
No offset variable in data file.
Number of cases used for training: 720
```

```
Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 7
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       55   2.849E+00   1.872E-01   2.033E-01   2.772E+00   2.508E-01
   2       53   2.849E+00   1.872E-01   2.033E-01   2.772E+00   2.508E-01
   3       52   2.849E+00   1.872E-01   2.033E-01   2.772E+00   2.508E-01
   :
  36        4   1.488E+00   8.070E-02   8.672E-02   1.449E+00   7.036E-02
  37**      3   1.457E+00   7.447E-02   9.380E-02   1.343E+00   7.680E-02
  38        2   1.527E+00   7.949E-02   9.597E-02   1.455E+00   6.790E-02
  39        1   1.660E+00   8.239E-02   7.060E-02   1.651E+00   7.689E-02

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of skips in the node
Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank    D-mean    deviance  variable       variables
          1      720      720    18  4.965E+00  1.610E+00  solder
          2T     360      360    17  2.481E+00  1.279E+00  mask
          3      360      360    17  7.450E+00  1.628E+00  opening :mask
          6T     120      120    15  1.636E+01  1.367E+00  padtype
          7T     240      240    16  2.996E+00  1.403E+00  mask

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5
Second best split variable (based on curvature test) at root node is opening
```

```
 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: solder = "thick"
   Node 2: skips sample mean = 2.4805556
 Node 1: solder /= "thick"
   Node 3: opening = "small"
     Node 6: skips sample mean = 16.358333
   Node 3: opening /= "small"
     Node 7: skips sample mean = 2.9958333

 ****************************************************************

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if solder = "thick"
 solder mode = "thick"
 Coefficients of regression function for log mean:
 Regressor         Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant          -1.220       -12.81       0.6614E-15
 mask.A3           0.4282        5.674       0.2043E-07    0.000       0.2500        1.000
 mask.B3            1.202        17.95       0.3926E-15    0.000       0.2500        1.000
 mask.B6            1.866        29.58        0.000        0.000       0.2500        1.000
 opening.medium    0.2585        3.884       0.1126E-03    0.000       0.3333        1.000
 opening.small      1.893        35.31       0.7867E-15    0.000       0.3333        1.000
 padtype.D6       -0.3687       -5.164       0.3144E-06    0.000       0.1000        1.000
 padtype.D7       -0.9844E-01   -1.487       0.1374        0.000       0.1000        1.000
 padtype.L4        0.2624        4.321       0.1774E-04    0.000       0.1000        1.000
 padtype.L6       -0.6685       -8.525        0.000        0.000       0.1000        1.000
 padtype.L7       -0.4902       -6.619       0.7177E-10    0.000       0.1000        1.000
 padtype.L8       -0.2712       -3.907       0.1023E-03    0.000       0.1000        1.000
 padtype.L9       -0.6365       -8.203       0.1231E-14    0.000       0.1000        1.000
 padtype.W4       -0.1100       -1.657       0.9804E-01    0.000       0.1000        1.000
 padtype.W9        -1.438       -13.80        0.000        0.000       0.1000        1.000
 panel.2           0.3335        7.929       0.8472E-14    0.000       0.3333        1.000
```

```
 panel.3           0.2544        5.947       0.4318E-08    0.000        0.3333        1.000
 solder.thin        1.100        28.46        0.000         0.000        0.5000        1.000
 If regressor has missing values, predicted value = 1.6024692
 ----------------------------
Node 2: Terminal node
 Coefficients of regression function for log mean:
 Regressor         Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant          -2.431       -10.68        0.000
 mask.A3           0.4670        2.373       0.1820E-01    0.000        0.2500        1.000
 mask.B3            1.831        11.01        0.000         0.000        0.2500        1.000
 mask.B6            2.520        15.71        0.000         0.000        0.2500        1.000
 opening.medium    0.8641        5.567       0.5228E-07    0.000        0.3333        1.000
 opening.small      2.465        18.18        0.000         0.000        0.3333        1.000
 padtype.D6       -0.3238       -2.034       0.4274E-01    0.000        0.1000        1.000
 padtype.D7        0.1201       0.8480       0.3970        0.000        0.1000        1.000
 padtype.L4        0.6985        5.534       0.6221E-07    0.000        0.1000        1.000
 padtype.L6       -0.4002       -2.458       0.1448E-01    0.000        0.1000        1.000
 padtype.L7        0.4167E-01   0.2887       0.7730        0.000        0.1000        1.000
 padtype.L8        0.1481        1.052       0.2936        0.000        0.1000        1.000
 padtype.L9       -0.5921       -3.426       0.6877E-03    0.000        0.1000        1.000
 padtype.W4       -0.5466E-01  -0.3696       0.7119        0.000        0.1000        1.000
 padtype.W9        -1.324       -5.886       0.9394E-08    0.000        0.1000        1.000
 panel.2           0.2224        2.718       0.6895E-02    0.000        0.3333        1.000
 panel.3           0.6825E-01   0.8049       0.4214        0.000        0.3333        1.000
 solder.thin        0.000        0.000        1.000        0.000         0.000        0.000
 If regressor has missing values, predicted value = 0.90848255
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if opening = "small"
 opening mode = "large"
 ----------------------------
Node 6: Terminal node
 Coefficients of regression function for log mean:
 Regressor         Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant           2.080        21.50        0.000
 mask.A3           0.3085        3.329       0.1202E-02    0.000        0.2500        1.000
 mask.B3            1.050        12.84        0.000         0.000        0.2500        1.000
 mask.B6            1.504        19.34        0.000         0.000        0.2500        1.000
 opening.medium     0.000        0.000        1.000         0.000         0.000        0.000
 opening.small      0.000        0.000        1.000         1.000         1.000        1.000
 padtype.D6       -0.2534       -2.788       0.6302E-02    0.000        0.1000        1.000
 padtype.D7       -0.1476       -1.671       0.9763E-01    0.000        0.1000        1.000
 padtype.L4        0.8309E-01   0.9980       0.3206        0.000        0.1000        1.000
 padtype.L6       -0.7187       -6.847       0.4730E-09    0.000        0.1000        1.000
 padtype.L7       -0.6473       -6.315       0.6560E-08    0.000        0.1000        1.000
 padtype.L8       -0.4255       -4.452       0.2127E-04    0.000        0.1000        1.000
```

```
 padtype.L9       -0.6404       -6.262       0.8418E-08    0.000        0.1000        1.000
 padtype.W4       -0.8668E-01  -0.9978       0.3207        0.000        0.1000        1.000
 padtype.W9        -1.376       -10.29        0.000        0.000        0.1000        1.000
 panel.2           0.3070        5.470       0.3070E-06    0.000        0.3333        1.000
 panel.3           0.1850        3.210       0.1762E-02    0.000        0.3333        1.000
 solder.thin        0.000        0.000        1.000        1.000         1.000        1.000
 If regressor has missing values, predicted value = 2.7947375
 ---------------------------
Node 7: Terminal node
 Coefficients of regression function for log mean:
 Regressor         Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant         -0.3711       -1.947       0.5284E-01
 mask.A3           0.8061        4.546       0.8965E-05    0.000        0.2500        1.000
 mask.B3            1.008        5.849       0.1735E-07    0.000        0.2500        1.000
 mask.B6            2.267        14.64       0.2731E-15    0.000        0.2500        1.000
 opening.medium    0.1030        1.379       0.1692        0.000        0.5000        1.000
 opening.small      0.000        0.000        1.000        0.000         0.000        0.000
 padtype.D6       -0.7995       -4.649       0.5709E-05    0.000        0.1000        1.000
 padtype.D7       -0.1915       -1.345       0.1800        0.000        0.1000        1.000
 padtype.L4        0.2065        1.601       0.1108        0.000        0.1000        1.000
 padtype.L6       -0.8201       -4.735       0.3894E-05    0.000        0.1000        1.000
 padtype.L7       -0.7595       -4.477       0.1206E-04    0.000        0.1000        1.000
 padtype.L8       -0.3606       -2.413       0.1662E-01    0.000        0.1000        1.000
 padtype.L9       -0.6660       -4.051       0.7039E-04    0.000        0.1000        1.000
 padtype.W4       -0.2254       -1.568       0.1183        0.000        0.1000        1.000
 padtype.W9        -1.747       -7.027       0.2514E-10    0.000        0.1000        1.000
 panel.2           0.5841        5.732       0.3190E-07    0.000        0.3333        1.000
 panel.3           0.6931        6.931       0.4388E-10    0.000        0.3333        1.000
 solder.thin        0.000        0.000        1.000        1.000         1.000        1.000
 If regressor has missing values, predicted value = 1.0972224
 ---------------------------
 Observed and fitted values are stored in mul.fit
 LaTeX code for tree is in mul.tex
```

Figure 20 shows the tree, which is much shorter than that in Figure 19. Node 3 is drawn in brown color to indicate that the split there is due to an interaction between two variables (`opening` and `mask`).

9.3 Offset variable: lung cancer data

We use a data set from an epidemiological study of the effect of public drinking water on cancer mortality in Missouri (Choi et al., 2005). The data file `lungcancer.txt` gives the number of deaths (`deaths`) from lung cancer among 115 counties (`county`) during the period 1972–1981 for both sexes (`sex`) and four age groups (`agegp`): 45–54, 55–64, 65–74, and over 75. The DSC file `lungcancer.dsc` below lists the variables

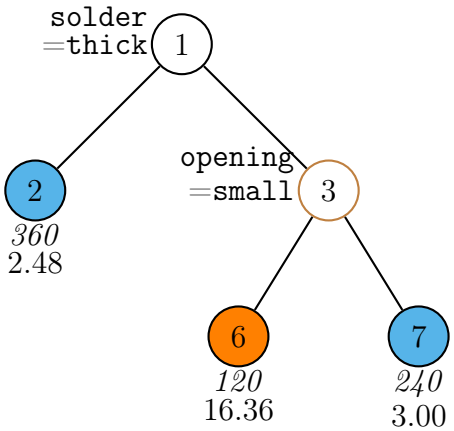

Figure 20: GUIDE v.46.1 0.250-SE multiple linear Poisson regression tree (constant fitted to incomplete cases in terminal nodes) for predicting `skips`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Splits at nodes drawn with **brown** circles are interaction splits. Sample size (in *italics*) and mean of `skips` printed below nodes. Terminal nodes with means above and below value of 4.97 at root node are painted orange and skyblue respectively. Second best split variable at root node is `opening`.

together with the county population (`pop`) and the natural log of `pop` (`logpop`). The latter is specified as `z` to serve an an offset variable and `pop` is excluded (`x`) from the analysis. The contents of `lungcancer.dsc` are:

```
lungcancer.txt
NA
1
1 county c
2 sex b
3 agegp c
4 deaths d
5 pop x
6 logpop z
```

Our goal is to construct a Poisson regression tree for the gender-specific rate of lung cancer deaths, where rate is the expected number of deaths in a county divided by its population size for each gender. That is, letting μ denote the expected number of gender-specific deaths in a county, we fit this model in each node of the tree:

$$\log(\mu/\texttt{pop}) = \beta_0 + \beta_1 I(\texttt{sex} = \texttt{M}).$$

This is achieved by fitting a linear Poisson regression model with `sex` as `b` so that its dummy indicator variable serves as a linear predictor in the Poisson node models.

9.3.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: poi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: poi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 3
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: lungcancer.dsc
Reading DSC file ...
Training sample file: lungcancer.txt
Missing value code: NA
Records in data file start on line 1
D variable is deaths
Reading data file ...
Number of records in data file: 920
Length of longest entry in data file: 8
Checking for missing values ...
Finished checking
Assigning integer codes to values of 3 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 869
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
```

```
Creating dummy variables ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       920          0          0        1        0        1        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        1        2        0
Number of dummy variables created: 1
Offset variable in column:           6
Number of cases used for training: 920
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Default max. number of split levels: 8
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): poi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: poi.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: poi.r
Input rank of top variable to split root node ([1:4], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < poi.in
```

9.3.2 Results

```
Poisson regression tree
No truncation of predicted values
Pruning by cross-validation
DSC file: lungcancer.dsc
Training sample file: lungcancer.txt
Missing value code: NA
Records in data file start on line 1
D variable is deaths
Piecewise linear model
Number of records in data file: 920
Length of longest entry in data file: 8
```

```
Number of cases with positive D values: 869

Summary information for training sample of size 920
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
z=offset variable
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  county     c                               115
      2  sex        b                                 2
      3  agegp      c                                 4
      4  deaths     d    0.000        1046.
      6  logpop     z    4.828        10.96
  ================== Constructed variables ==================
      7  sex.M      f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       920          0          0        1        0        1        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        1        2        0
Number of dummy variables created: 1
Offset variable in column 6
Number of cases used for training: 920

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 7
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       55   2.680E+00   2.855E-01   1.856E-01   2.609E+00   3.165E-01
   2       54   2.680E+00   2.855E-01   1.856E-01   2.609E+00   3.165E-01
   :
  44        4   2.329E+00   3.284E-01   2.704E-01   2.126E+00   3.242E-01
  45**      3   2.255E+00   3.279E-01   2.717E-01   1.954E+00   2.711E-01
```

```
  46        2   4.702E+00   8.054E-01   4.866E-01   4.153E+00   6.629E-01
  47        1   9.431E+00   1.420E+00   9.674E-01   9.043E+00   9.329E-01

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Rate is mean of Y/exp(offset)
Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank     rate     deviance  variable       variables
          1      920      920     2  1.382E-02  9.179E+00  agegp
          2T     230      230     2  5.493E-03  1.863E+00  county
          3      690      690     2  1.763E-02  4.357E+00  agegp
          6T     230      230     2  1.339E-02  3.003E+00  county
          7T     460      460     2  2.093E-02  1.802E+00  agegp

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5
Second best split variable (based on curvature test) at root node is sex

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: agegp = "45-54"
  Node 2: deaths sample rate = 0.54928582E-2
Node 1: agegp /= "45-54"
  Node 3: agegp = "55-64"
    Node 6: deaths sample rate = 0.13389777E-1
  Node 3: agegp /= "55-64"
    Node 7: deaths sample rate = 0.20932715E-1

*************************************************************

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.
```

```
 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if agegp = "45-54"
 agegp mode = "45-54"
 Coefficients of regression function for log expected rate:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -5.172       -366.9        0.000
 sex.M         1.437        89.64        0.000        0.000       0.5000        1.000
 Node mean for offset variable =    6.727
 If regressor has missing values, predicted rate = 0.13824405E-1
 ----------------------------
Node 2: Terminal node
 Coefficients of regression function for log expected rate:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -5.834       -161.5       0.3750E-15
 sex.M         1.038        24.44       0.1308E-15    0.000       0.5000        1.000
 Node mean for offset variable =    6.857
 If regressor has missing values, predicted rate = 0.54928582E-2
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if agegp = "55-64"
 agegp mode = "55-64"
 ----------------------------
Node 6: Terminal node
 Coefficients of regression function for log expected rate:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -5.117       -199.8        0.000
 sex.M         1.285        43.87        0.000        0.000       0.5000        1.000
 Node mean for offset variable =    6.920
 If regressor has missing values, predicted rate = 0.13389777E-1
 ----------------------------
Node 7: Terminal node
 Coefficients of regression function for log expected rate:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -4.907       -256.9        0.000
 sex.M         1.714        79.68        0.000        0.000       0.5000        1.000
 Node mean for offset variable =    6.567
 If regressor has missing values, predicted rate = 0.20932715E-1
 ----------------------------
 Observed and fitted values are stored in poi.fit
```

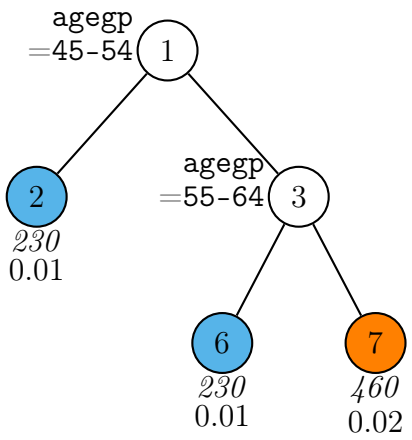

Figure 21: GUIDE v.46.1 0.250-SE multiple linear Poisson regression tree (constant fitted to incomplete cases in terminal nodes) for predicting rate of `deaths`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and sample rate printed below nodes. Terminal nodes with rates above and below value of 0.014 at root node are painted orange and skyblue respectively. Second best split variable at root node is `sex`.

```
LaTeX code for tree is in poi.tex
```

The results show that the death rate increases with age and that the rate for males is consistently higher than that for females. The tree diagram is given in Figure 21.

10 Censored response: RHC data

Doctors believe that direct measurement of cardiac function by right heart catheterization (RHC) for some critically ill patients yields better outcomes. Relative risk of death is higher in elderly and patients with acute myocardial infarction who received RHC. Benefit of RHC has not been demonstrated in a randomized clinical trial, because physicians refuse to allow their patients to be randomized. Treatment selection is confounded with patient factors that are also related to outcomes, e.g., patients with low blood pressure are more likely to get RHC, and such patients are also more likely to die. Data consist of observations on more than 60 variables for 5735 patients from 5 medical centers over 5 years (Connors et al., 1996). Response

Table 6: Demographics & outcomes for RHC data [#missing in brackets]

Variable	Definition
swang1	Right heart catheterization (RHC) [0]
age	Age in years [0]
sex	Sex (female/male) [0]
wtkilo1	Weight in kilograms [515]
edu	Years of Education [0]
race	Race [0]
income	Income bracket (<11k, 11–25k, 25–50k, >50k) [0]
ninsclas	Medical insurance (Medicaid, Medicare, Medicare & Medicaid, no insurance, private, private & Medicare) [0]
t3d30	Days from admission to death within 30 days [0]
dth30	Death indicator for t3d30 (0=no, 1=yes) [0]
survtime	Days from admission to death or last contact day [0]
death	Death indicator for survtime (0=no, 1=yes) [0]
transhx	Transfer (> 24 hours) from another hospital (no/yes) [0]

variables are `t3d30` (censored 30-day survival time) and `survtime` (days till death or last contact). Variable `dth30` is 1 if death occurs within 30 days of hospital admission and 0 otherwise; `death` is 1 if the subject eventually dies and 0 if death is unknown. Variable `swang1` takes values "RHC" and "NoRHC", indicating whether or not a patient received RHC. Other variables are given in Tables 6–8.

GUIDE can fit a proportional hazards model to the censored survival time if the event indicator `death` is specified as "`D`" and `survtime` as "`T`". The DSC file is `rhcdsc2.txt` whose contents follow.

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death d
10 cardiohx c
11 chfhx c
12 dementhx c
```

Table 7: Disease variables for RHC data [#missing in brackets]

Variable	Definition
cat1	Primary disease category (9 levels) [0]
cat2	Secondary disease category (6 levels) [2798]
ca	Cancer (3 levels) [0]
card	Cardiovascular diagnosis [0]
gastr	Gastrointestinal diagnosis [0]
hema	Hematologic diagnosis [0]
meta	Metabolic diagnosis [0]
neuro	Neurological diagnosis [0]
ortho	Orthopedic diagnosis [0]
renal	Renal diagnosis [0]
resp	Respiratory diagnosis [0]
seps	Sepsis diagnosis [0]
trauma	Trauma diagnosis [0]

Table 8: Medical history for RHC data [#missing in brackets]

Variable	Definition
amihx	Definite myocardial infarction (no/yes) [0]
cardiohx	Acute MI, peripheral vascular disease, severe cardiovascular symptoms [0]
chfhx	Congestive heart failure (no/yes) [0]
chrpulhx	Chronic or severe pulmonary disease (no/yes) [0]
dementhx	Dementia, stroke or cerebral infarction, Parkinson's disease (no/yes) [0]
gibledhx	Upper GI bleeding (no/yes) [0]
liverhx	Cirrhosis, hepatic failure (no/yes) [0]
malighx	Solid tumor, metastatic disease, chronic leukemia/myeloma, acute leukemia, lymphoma (no/yes) [0]
immunhx	Immunosuppression, organ transplant, HIV positivity, diabetes mellitus, connective tissue disease(no/yes) [0]
psychhx	Psychiatric history, active psychosis or severe depression (no/yes) [0]
renalhx	Chronic renal disease, chronic hemodialysis or peritoneal dialysis (no/yes) [0]

Table 9: Admission variables for RHC data [#missing in brackets]

Variable	Definition
alb1	Albumin [0]
bili1	Bilirubin [0]
crea1	Serum creatinine [0]
hema1	Hematocrit [0]
hrt1	Heart rate [159]
meanbp1	Mean blood pressure [80]
pot1	Serum potassium [0]
pao2	Partial pressure of arterial oxygen [0]
fio2	Fraction of inspired oxygen [0]
pafi1	PaO2/(0.01*FiO2) [0]
paco21	Partial pressure of arterial carbon dioxide [0]
ph1	Serum ph [0]
resp1	Respiration rate [136]
scoma1	Glasgow coma score [0]
sod1	Serum sodium [0]
temp1	Temperature (Celsius) [0]
urin1	Urine output [3028]
wblc1	White blood cell count [0]
aps1	APACHE III score ignoring coma [0]
adld3p	Katz Activities of Daily Living Scale [3016]
das2d3pc	DASI (Duke Activity Status Index) [0]
dnr1	DNR (do-not-resuscitate) status [0]
surv2md1	Estimated probability of 2-month survival [0]

```
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 c
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
```

```
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

10.1 Proportional hazards

GUIDE has two options for modeling censored response data. The first is a piecewise Cox proportional hazards model.

Let the survival time of a subject be U with probability density $f(u)$ and distribution function $F(u)$. The survival probability function is $S(u) = P(U > u) = 1-F(u)$ and the hazard rate (instantaneous rate of death) at time u is $\lambda(u) = f(u)/S(u)$. Let U_i and C_i be survival and censoring times of subject i. Let $Y_i = \min(U_i, C_i)$ be the observed censored survival time and let $\delta_i = I(U_i < C_i)$ denote the event indicator. The proportional hazards model assumes that $\lambda(u, \mathbf{x}) = \lambda_0(u) \exp(\beta'\mathbf{x})$, where $\lambda_0(u)$ is an unknown baseline hazard function. Unlike other regression tree methods for survival data, $\lambda_0(u)$ is the same for all terminal nodes of a GUIDE tree.

10.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: censored.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: censored.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3):
```

```
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc2.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of cases excluded due to 0 W or missing D or T variables: 0
Finished reading data file
Default max. number of split levels: 15
```

```
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): censored.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: censored.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: censored.r
Input rank of top variable to split root node ([1:54], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < censored.in
```

10.1.2 Output file

```
Proportional hazards regression tree for censored response
Pruning by cross-validation
DSC file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
```

Column	Name		Minimum	Maximum	#Codes/ Levels/ Periods	#Missing
2	cat1	c			9	
3	cat2	c			6	4535
4	ca	c			3	
9	death	d	0.000	1.000		
10	cardiohx	c			2	
11	chfhx	c			2	
12	dementhx	c			2	
13	psychhx	c			2	
14	chrpulhx	c			2	
15	renalhx	c			2	
16	liverhx	c			2	
17	gibledhx	c			2	
18	malighx	c			2	
19	immunhx	c			2	
20	transhx	c			2	
21	amihx	c			2	
22	age	s	18.04	101.8		
23	sex	c			2	
24	edu	s	0.000	30.00		
25	surv2md1	s	0.000	0.9620		
26	das2d3pc	s	11.00	33.00		
29	aps1	s	3.000	147.0		
30	scoma1	s	0.000	100.0		
31	meanbp1	s	10.00	259.0		80
32	wblc1	s	0.000	192.0		
33	hrt1	s	8.000	250.0		159
34	resp1	s	2.000	100.0		136
35	temp1	s	27.00	43.00		
36	pafi1	s	11.60	937.5		
37	alb1	s	0.3000	29.00		
38	hema1	s	2.000	66.19		
39	bili1	s	0.9999E-01	58.20		
40	crea1	s	0.9999E-01	25.10		
41	sod1	s	101.0	178.0		
42	pot1	s	1.100	11.90		
43	paco21	s	1.000	156.0		
44	ph1	s	6.579	7.770		
45	swang1	c			2	
46	wtkilo1	s	19.50	244.0		515
47	dnr1	c			2	
48	ninsclas	c			6	
49	resp	c			2	
50	card	c			2	

```
     51  neuro      c                                  2
     52  gastr      c                                  2
     53  renal      c                                  2
     54  meta       c                                  2
     55  hema       c                                  2
     56  seps       c                                  2
     57  trauma     c                                  2
     58  ortho      c                                  2
     59  adld3p     s    0.000        7.000                  4296
     60  urin1      s    0.000        9000.                  3028
     61  race       c                                  3
     62  income     c                                  4
     64  survtime   t    2.000        1943.
  =================== Constructed variables ===================
     65  lnbasehaz  z   -3.818        2.038

 Numbers of N, F, S, and P below exclude constant variables
      Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
 Survival time variable in column: 64
 Event indicator variable in column: 9
 Proportion uncensored among nonmissing T and D variables: 0.649
 Number of cases used for training: 5735
 Number of cases excluded due to 0 W or missing D or T variables: 0

 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Nodewise interaction tests on all variables
 Split values for N and S variables based on exhaustive search
 Max number of interaction splits: 100
 Maximum number of split levels: 15
 Minimum node sample size: 5
 Number of iterations for fitting: 20
 Ranks of variables and their 1-df chi-squared values at root node
      1  0.7537E+03   surv2md1
      2  0.3238E+03   adld3p
      3  0.2324E+03   cat1
      4  0.2262E+03   aps1
      :
     51  0.4567E-01   meta
     52  0.2905E-01   chfhx
```

```
    53  0.1966E-01   cardiohx
    54  0.1463E-01   chrpulhx

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1      693   1.897E+00   4.509E-02   3.758E-02   1.879E+00   5.321E-02
   2      692   1.897E+00   4.509E-02   3.758E-02   1.879E+00   5.321E-02
   :
 420       11   1.254E+00   1.908E-02   1.498E-02   1.238E+00   2.194E-02
 421**     10   1.242E+00   1.732E-02   1.342E-02   1.227E+00   1.880E-02
 422        8   1.250E+00   1.701E-02   1.140E-02   1.235E+00   1.777E-02
 423        7   1.257E+00   1.701E-02   1.238E-02   1.243E+00   2.317E-02
 424        6   1.267E+00   1.698E-02   1.215E-02   1.257E+00   1.838E-02
 425        5   1.283E+00   1.726E-02   1.261E-02   1.281E+00   1.891E-02
 426        3   1.291E+00   1.700E-02   1.253E-02   1.281E+00   2.092E-02
 427        2   1.332E+00   1.684E-02   1.207E-02   1.329E+00   1.682E-02
 428        1   1.455E+00   1.624E-02   6.159E-03   1.451E+00   9.894E-03

0-SE tree based on mean is marked with * and has 10 terminal nodes
0-SE tree based on median is marked with + and has 10 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split          Other
       label   cases      fit  rank   survtime   deviance  variable       variables
          1    5735     5735     1  1.920E+02  1.455E+00  surv2md1
          2    2157     2157     1  2.400E+01  1.494E+00  adld3p
          4    1924     1924     1  1.900E+01  1.523E+00  surv2md1
          8T    705      705     1  1.100E+01  1.431E+00  cat1
          9    1219     1219     1  2.800E+01  1.487E+00  dnr1
         18T   1025     1025     1  3.700E+01  1.423E+00  psychhx :surv2md1
         19T    194      194     1  8.500E+00  1.427E+00  liverhx
          5T    233      233     1  1.950E+02  9.305E-01  ca
          3    3578     3578     1  3.290E+02  1.223E+00  surv2md1
          6    1806     1806     1  2.270E+02  1.345E+00  adld3p
         12    1364     1364     1  1.270E+02  1.455E+00  dnr1
         24T   1213     1213     1  1.680E+02  1.410E+00  das2d3pc
         25T    151      151     1  2.500E+01  1.589E+00  dementhx
```

```
         13T     442      442     1  3.750E+02  8.581E-01  das2d3pc
          7     1772     1772     1  4.030E+02  1.022E+00  chfhx
         14     1279     1279     1  4.410E+02  1.039E+00  das2d3pc
         28T     861      861     1  3.640E+02  1.071E+00  cat1
         29T     418      418     1  5.470E+02  8.881E-01  surv2md1
         15T     493      493     1  3.730E+02  9.358E-01  surv2md1

Number of terminal nodes of final tree: 10
Total number of nodes of final tree: 19
Second best split variable (based on curvature test) at root node is adld3p

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: surv2md1 <= 0.56398225
  Node 2: adld3p = NA
    Node 4: surv2md1 <= 0.35699725
      Node 8: Median survival time = 11.000000
    Node 4: surv2md1 > 0.35699725 or NA
      Node 9: dnr1 = "No"
        Node 18: Median survival time = 37.000000
      Node 9: dnr1 /= "No"
        Node 19: Median survival time = 8.5000000
  Node 2: adld3p /= NA
    Node 5: Median survival time = 195.00000
Node 1: surv2md1 > 0.56398225 or NA
  Node 3: surv2md1 <= 0.71695924
    Node 6: adld3p = NA
      Node 12: dnr1 = "No"
        Node 24: Median survival time = 168.00000
      Node 12: dnr1 /= "No"
        Node 25: Median survival time = 25.000000
    Node 6: adld3p /= NA
      Node 13: Median survival time = 375.00000
  Node 3: surv2md1 > 0.71695924 or NA
    Node 7: chfhx = "0"
      Node 14: das2d3pc <= 24.798830
        Node 28: Median survival time = 364.00000
      Node 14: das2d3pc > 24.798830 or NA
        Node 29: Median survival time = 547.00000
    Node 7: chfhx /= "0"
      Node 15: Median survival time = 373.00000

************************************************************
Predictor means below are means of cases with no missing values.
```

```
 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if surv2md1 <= 0.56398225
 surv2md1 mean = 0.59245008
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant      0.000
 Predicted log-relative hazard = 192.00000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.3605150
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if surv2md1 <= 0.35699725
 surv2md1 mean = 0.38119025
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant      1.011
 Predicted log-relative hazard = 11.000000
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 18: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3648
 Predicted log-relative hazard = 37.000000
 ----------------------------
Node 19: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
```

```
 Constant      1.168
 Predicted log-relative hazard = 8.5000000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.2260E-02
 Predicted log-relative hazard = 195.00000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if surv2md1 <= 0.71695924
 surv2md1 mean = 0.71756975
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if adld3p = NA
 adld3p mean = 1.1606335
 ----------------------------
Node 12: Intermediate node
 A case goes into Node 24 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 24: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.1478E-01
 Predicted log-relative hazard = 168.00000
 ----------------------------
Node 25: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant     0.5762
 Predicted log-relative hazard = 25.000000
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.5115
 Predicted log-relative hazard = 375.00000
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if chfhx = "0"
 chfhx mode = "0"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if das2d3pc <= 24.798830
 das2d3pc mean = 21.932716
```

```
 ----------------------------
Node 28: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.5874
 Predicted log-relative hazard = 364.00000
 ----------------------------
Node 29: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant     -1.261
 Predicted log-relative hazard = 547.00000
 ----------------------------
Node 15: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.4073
 Predicted log-relative hazard = 373.00000
 ----------------------------
 Observed and fitted values are stored in censored.fit
 LaTeX code for tree is in censored.tex
```

Figure 22 shows the fitted model. Sample size (in *italics*), relative hazard, and median survival time are printed below each terminal node. The top lines of the file `censored.fit` are:

train	node	observed	event	logbasecumhaz	survivalprob	mediansurvtime
y	13	0.240000E+003	n	-0.256265E+000	0.628727E+000	0.375000E+003
y	15	0.450000E+002	y	-0.799746E+000	0.741494E+000	0.373000E+003
y	8	0.317000E+003	n	-0.495237E-001	0.731318E-001	0.110000E+002
y	18	0.370000E+002	y	-0.884460E+000	0.551714E+000	0.370000E+002
y	19	0.200000E+001	y	-0.400645E+001	0.943169E+000	0.850000E+001

The columns give the following information:

`train`: equals `y` if observation is used for model fitting; equals `n` if not used.

`node`: terminal node label of observation.

`observed`: observed survival time (`t` variable in DSC file).

`event`: equals `y` if `observed` is uncensored (`d`=1); equals `n` if censored (`d`=0).

`logbasecumhaz`: log of the estimated baseline cumulative hazard function $\log \Lambda_0(t) = \log \int_0^t \lambda_0(u)\, du$ at observed time t.

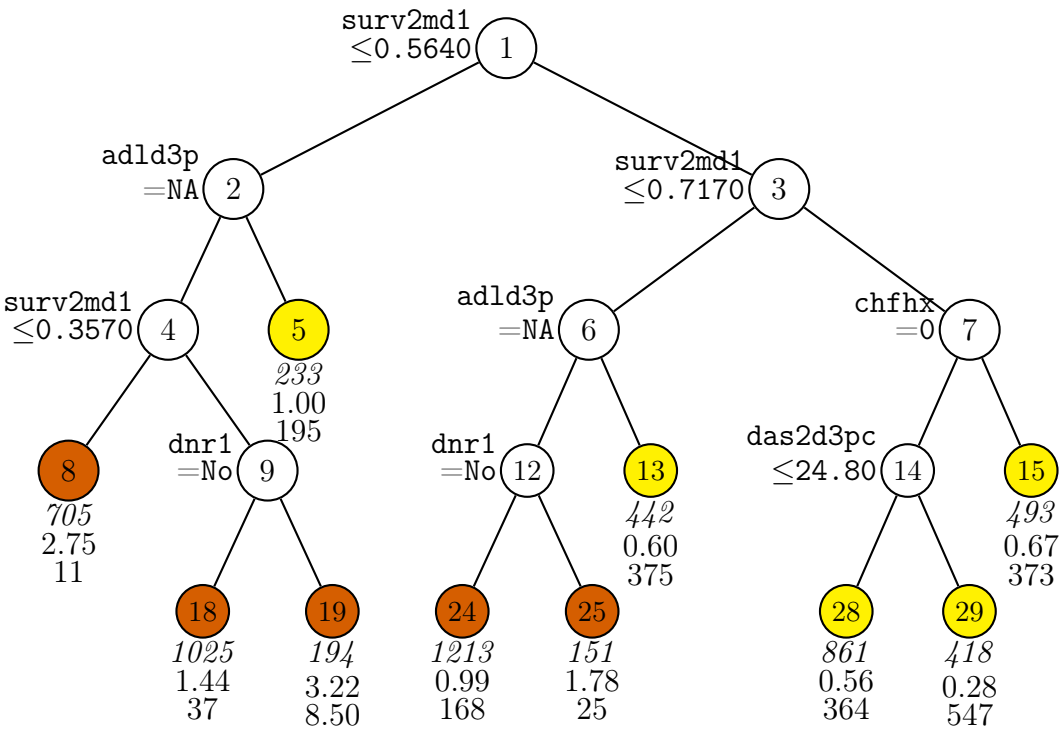

Figure 22: GUIDE v.46.1 0.250-SE piecewise-constant proportional hazards regression tree for `survtime`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*), relative hazard, and median survival time printed below nodes. Terminal nodes with median survival times above and below 192 (median at root node) are painted yellow and vermillion respectively. Second best split variable at root node is `adld3p`.

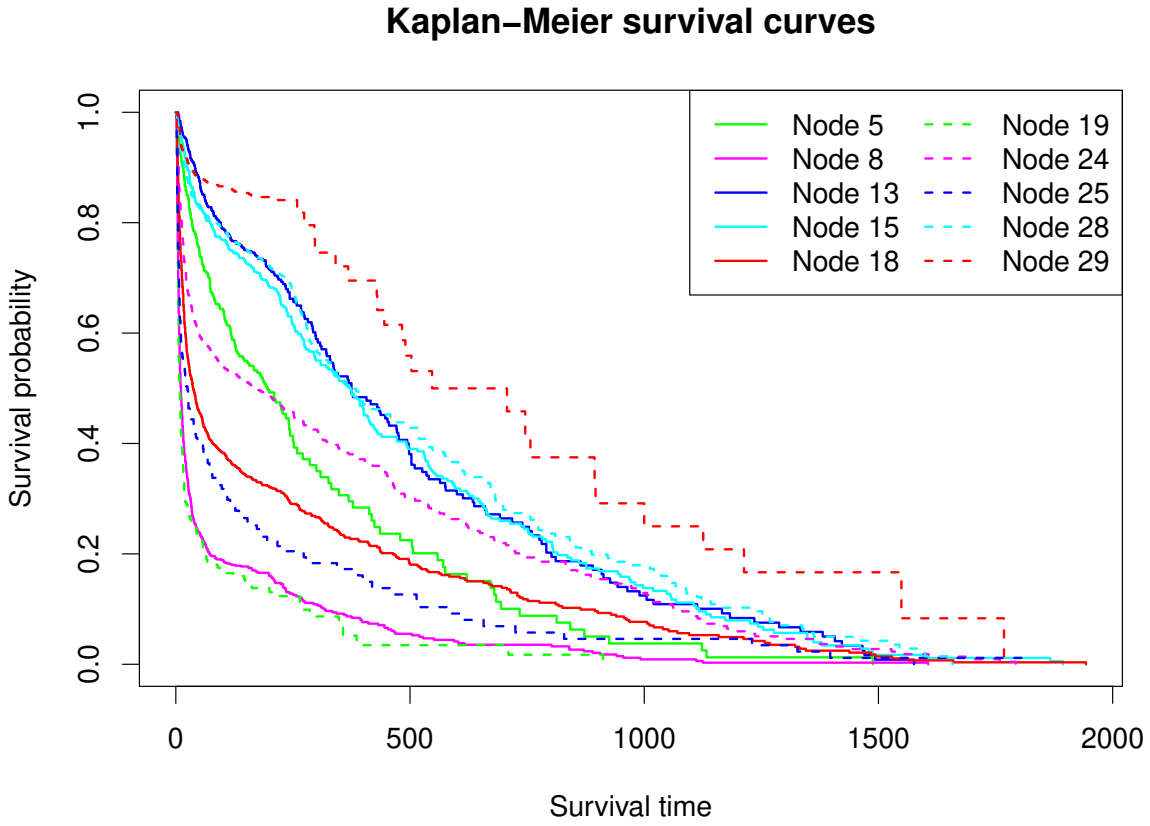

Figure 23: Kaplan-Meier survival curves for data in terminal nodes of Figure 22

`survivalprob`: probability that the subject survives up to observed time t. For the first subject, this is

$$\begin{aligned} \exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\beta_0 + \texttt{logbasecumhaz})\} \\ &= \exp(-\exp(-0.514911594896 - 0.261185)) \\ &= 0.6311581 \end{aligned}$$

where $t = 240$ and $\beta_0 = -0.514911594896$ is the constant term in the node (`censored.r` gives β_0 to higher precision than `censored.out`).

`mediansurvtime`: median survival time among observations in node estimated from Kaplan-Meier survival function. A trailing plus (+) sign indicates estimate is censored.

Figure 23 plots the estimated survival curves in the terminal nodes of the tree. The plot is produced by the following R code.

```
library(survival)
z0 <- read.table("rhcdata.txt",header=TRUE)
z1 <- read.table("censored.fit",header=TRUE)
nodenum <- unique(sort(z1$node))
leg.txt <- paste("Node",nodenum)
leg.col <- c("green","magenta","blue","cyan","red")
leg.lty <- rep(c(1,2),c(5,5))
fit <- survfit(Surv(z0$survtime,z0$death) ~ z1$node, conf.type="none")
plot(fit,mark.time=FALSE,xlab="Survival time",ylab="Survival probability",
     col=leg.col,lwd=2,lty=leg.lty)
title("Kaplan-Meier survival curves")
legend("topright",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2,ncol=2)
```

10.2 Restricted mean event time

The mean survival time is not estimable if there is censoring. But given a pre-specified time point τ, the restricted mean survival time $\mu(X) = E(Y|X)$ is estimable, where $Y = \min(U, C, \tau)$ and X is a covariate vector (Andersen et al., 2004; Chen and Tsiatis, 2001; Tian et al., 2014). GUIDE has an option to fit a *restricted event time model* to each node of the tree such that $\mu(X)$ is linear in the covariates.

10.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: rest.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: rest.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
 Choose complexity of model to use at each node:
 Choose 0 for stepwise linear regression (recommended for prediction)
 Choose 1 for multiple regression
 Choose 2 for best simple polynomial in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
```

```
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc2.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=972.00):

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of cases excluded due to 0 W or missing D variable: 2003
Finished reading data file
Default max. number of split levels: 15
```

```
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: rest.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: rest.r
Input rank of top variable to split root node ([1:54], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest.in
```

10.2.2 Contents of `rest.out`

```
Restricted mean event time regression tree
Pruning by cross-validation
DSC file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Interval for restricted mean event time is from 0 to 972.

Summary information for training sample of size 3732 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                #Codes/
```

```
                                                  Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1       c                                 9
      3  cat2       c                                 6     2807
      4  ca         c                                 3
      9  death      d    0.000        1.000
     10  cardiohx   c                                 2
      :
     60  urin1      s    0.000        9000.                  2093
     61  race       c                                 3
     62  income     c                                 4
     64  survtime   t    2.000        1943.

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of cases excluded due to 0 W or missing D variable: 2003

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 3
Ranks of variables and their 1-df chi-squared values at root node
      1  0.1868E+03   adld3p
      2  0.1629E+03   surv2md1
      3  0.1122E+03   cat1
      4  0.6234E+02   aps1
      :
     52  0.2165E+00   race
     53  0.1196E+00   amihx
     54  0.6209E-01   income

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      771   1.417E+05   5.391E+03   2.630E+03   1.392E+05   4.541E+03
   2      770   1.417E+05   5.391E+03   2.630E+03   1.392E+05   4.541E+03
```

```
   3      769   1.417E+05   5.391E+03   2.630E+03   1.392E+05   4.541E+03
   :
 504        8   1.134E+05   3.719E+03   1.992E+03   1.110E+05   2.521E+03
 505        7   1.112E+05   3.530E+03   1.623E+03   1.109E+05   2.809E+03
 506--      6   1.084E+05   3.219E+03   1.489E+03   1.092E+05   1.373E+03
 507**      4   1.091E+05   3.046E+03   1.501E+03   1.097E+05   2.588E+03
 508++      3   1.097E+05   3.048E+03   1.409E+03   1.093E+05   1.989E+03
 509        2   1.102E+05   3.057E+03   1.510E+03   1.101E+05   2.162E+03
 510        1   1.225E+05   3.100E+03   2.805E+02   1.225E+05   4.687E+02

0-SE tree based on mean is marked with * and has 6 terminal nodes
0-SE tree based on median is marked with + and has 6 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases Matrix    Node      Node      Split          Interacting
       label   cases      fit  rank    D-mean     MSE       variable       variable
          1     3732     3732     1  3.144E+02  1.800E+05   adld3p
          2T     664      664     1  4.685E+02  2.273E+05   surv2md1
          3     3068     3068     1  2.647E+02  1.556E+05   surv2md1
          6T    1262     1262     1  1.607E+02  8.878E+04   dnr1
          7     1806     1806     1  3.225E+02  1.880E+05   urin1
         14T     800      800     1  2.032E+02  8.072E+04   adld3p
         15T    1006     1006     1  4.004E+02  2.479E+05   surv2md1

Number of terminal nodes of final tree: 4
Total number of nodes of final tree: 7
Second best split variable (based on curvature test) at root node is surv2md1

Regression tree:

Node 1: adld3p <= 5.5000000
  Node 2: survtime-mean = 468.46294
Node 1: adld3p > 5.5000000 or NA
  Node 3: surv2md1 <= 0.49098337
    Node 6: survtime-mean = 160.70095
```

```
   Node 3: surv2md1 > 0.49098337 or NA
     Node 7: urin1 <= 7212.5000
       Node 14: survtime-mean = 203.19504
     Node 7: urin1 > 7212.5000 or NA
       Node 15: survtime-mean = 400.36912

 ****************************************************************
 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if adld3p <= 5.5000000
 adld3p mean = 1.2733830
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      314.4        45.27       0.3032-313
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      468.5        25.32       0.3032-313
 survtime mean = 468.463
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if surv2md1 <= 0.49098337
 surv2md1 mean = 0.54259828
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      160.7        19.16       0.3032-313
 survtime mean = 160.701
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if urin1 <= 7212.5000
 urin1 mean = 1998.7301
```

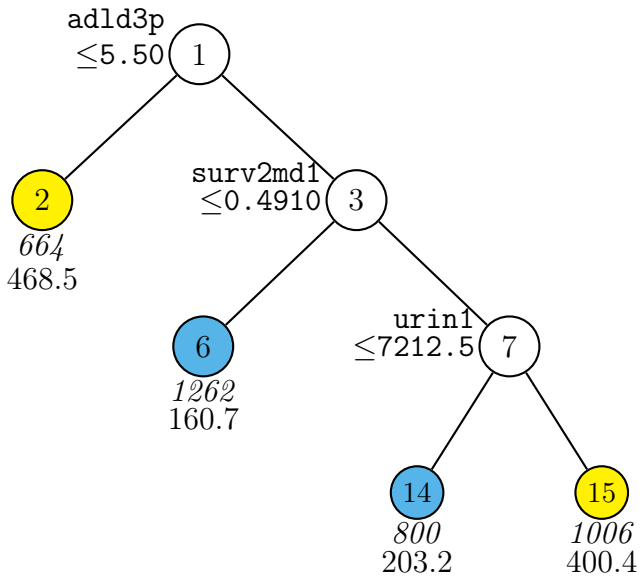

Figure 24: GUIDE v.46.1 0.250-SE piecewise-constant regression tree for mean `survtime` restricted to less than 972.000. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and restricted mean of `survtime` printed below nodes. Terminal nodes with means above and below value of 314.4 at root node are painted yellow and skyblue respectively. Second best split variable at root node is `surv2md1`.

```
 ----------------------------
Node 14: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      203.2        20.23       0.3032-313
 survtime mean = 203.195
 ----------------------------
Node 15: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      400.4        25.50       0.3032-313
 survtime mean = 400.369
 ----------------------------
 Observed and fitted values are stored in rest.fit
 LaTeX code for tree is in rest.tex
```

Figure 24 shows the restricted mean event time tree.

11 Randomized treatments

Causal effects of treatments are best studied in a randomized trial where the treatments are assigned randomly to subjects. The goal is to show that one treatment is more efficacious than another across all subjects. If this determination is not achieved, a secondary goal may be to search for subgroups of subjects with differential treatment effects.

There are two types of covariates for identification of subgroups with differential treatment effects. A *prognostic* variable is a clinical or biologic characteristic that provides information on the likely outcome of the disease in an untreated individual (e.g., patient age, family history, disease stage, and prior therapy). A *predictive* variable is one that provides information on the likely benefit from the treatment. Predictive variables can be used to identify subgroups of patients who are most likely to benefit from a given therapy. In general, prognostic variables define the effects of patient or tumor characteristics on the patient outcome, whereas predictive variables define the effect of treatment on the tumor (Italiano, 2011). Accordingly, GUIDE has two options, called `Gi` and `Gs`. `Gi` is more sensitive to predictive variables and `Gs` tends to be equally sensitive to prognostic and predictive variables (Loh et al., 2015).

11.1 Multiple treatment arms: CAPE data

We first demonstrate this on a data set from a three-armed randomized controlled experiment to find out whether two interventions (DVD or Phone) are more efficacious than a control at promoting mammography screening. The relevant data and DSC files are `cape.txt` and `cape.dsc`. Note that the three treatment levels (contained in the treatment (`R`) variable `group`) are assumed to be categorical (i.e., nominal valued). See Loh et al. (2016) for more information on the data.

Because the response variable (`resp6`) is 0-1 (0=no, 1=yes), we use least-squares regression with `resp6` designated as the dependent variable `D` or `d` in the DSC file. The treatment variable (`group`) is designated as `R` or `r` (for "Rx").

11.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: gi.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
conversion ([1:3], <cr>=1):
```

```
Name of batch output file: gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: cape.dsc
Reading DSC file ...
Training sample file: cape.txt
Missing value code: NA
Records in data file start on line 1
R variable present
21 N variables changed to S
Warning: model changed to linear in treatment
D variable is resp6
Reading data file ...
Number of records in data file: 1681
Length of longest entry in data file: 25
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Treatment (R) variable is group with values "Control", "DVD", and "Phone"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
```

```
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Proportion of training sample for each level of group
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1681         43         84        1        0        0       21
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       17        0        1
No weight variable in data file
Number of cases used for training: 1638
Number of split variables: 38
Number of dummy variables created: 2
Number of cases excluded due to 0 W or missing D or R variables: 43
Finished reading data file
Default max. number of split levels: 14
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: gi.r
Input rank of top variable to split root node ([1:41], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < gi.in
```

11.1.2 Contents of `gi.out`

```
Least squares regression tree
Pruning by cross-validation
DSC file: cape.dsc
Training sample file: cape.txt
Missing value code: NA
Records in data file start on line 1
R variable present
```

```
21 N variables changed to S
Warning: model changed to linear in treatment
D variable is resp6
Piecewise linear model
Number of records in data file: 1681
Length of longest entry in data file: 24
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is group with values "Control", "DVD", and "Phone"
Proportion of training sample for each level of group
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413

Summary information for training sample of size 1638 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                    #Codes/
                                                    Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  resp6          d    0.000        1.000
      3  group          r                                3
      4  age            s    41.00        75.00                    1
      5  educyrs        s    2.000        20.00
      6  collegeormore  c                                2
      :
     37  susc           s    5.000        25.00
     38  fear           s    8.000        40.00
     39  fatal          s    11.00        42.00
     40  know           s    1.000        7.000
     41  stage          c                                4
  ================== Constructed variables ==================
     42  group.DVD      f    0.000        1.000
     43  group.Phone    f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1681         43         84        1        0        2       21
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       17        0        1
```

```
Number of dummy variables created: 2
No weight variable in data file
Number of cases used for training: 1638
Number of cases excluded due to 0 W or missing D or R variables: 43

Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 8
Minimum fraction of cases per treatment at each node: 0.066
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       16   2.561E-01   4.759E-03   4.872E-03   2.580E-01   7.191E-03
   2       15   2.560E-01   4.748E-03   4.807E-03   2.580E-01   7.191E-03
   3       14   2.561E-01   4.732E-03   4.732E-03   2.580E-01   7.247E-03
   4       13   2.560E-01   4.725E-03   4.717E-03   2.580E-01   7.188E-03
   5       11   2.560E-01   4.692E-03   4.769E-03   2.580E-01   7.191E-03
   6        9   2.530E-01   4.434E-03   4.209E-03   2.537E-01   5.715E-03
   7        8   2.528E-01   4.310E-03   3.948E-03   2.537E-01   5.991E-03
   8        7   2.424E-01   3.624E-03   4.306E-03   2.415E-01   5.952E-03
   9*       5   2.408E-01   3.157E-03   2.616E-03   2.410E-01   3.743E-03
  10**      1   2.414E-01   2.372E-03   5.044E-04   2.410E-01   6.719E-04

0-SE tree based on mean is marked with * and has 5 terminal nodes
0-SE tree based on median is marked with + and has 1 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as + tree
** tree same as -- tree
++ tree same as -- tree
+ tree same as ++ tree

WARNING: tree based on mean CV estimate of error has no splits
Choosing smallest nontrivial tree with no larger CV error estimate

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of resp6 in the node
```

```
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split           Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable        variables
          1     1638     1638     3  4.035E-01  2.410E-01  0.0006 sf12gh
          2      903      903     3  3.732E-01  2.336E-01  0.0046 know
          4      703      703     3  3.898E-01  2.384E-01  0.0018 educyrs
          8      543      543     3  3.720E-01  2.324E-01  0.0105 yearmam
         16T     427      427     3  2.998E-01  2.091E-01  0.0107 -
         17T     116      116     3  6.379E-01  2.248E-01  0.0518 -
          9T     160      160     3  4.500E-01  2.387E-01  0.0535 know
          5T     200      200     3  3.150E-01  2.039E-01  0.0693 fear
          3T     735      735     3  4.408E-01  2.455E-01  0.0081 sf12sf

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9
Second best split variable (based on curvature test) at root node is know

Regression tree:

Node 1: sf12gh <= 72.500000
  Node 2: know <= 6.5000000
    Node 4: educyrs <= 15.500000
      Node 8: yearmam <= 3.5000000
        Node 16: resp6-mean = 0.29976581
      Node 8: yearmam > 3.5000000 or NA
        Node 17: resp6-mean = 0.63793103
    Node 4: educyrs > 15.500000 or NA
      Node 9: resp6-mean = 0.45000000
  Node 2: know > 6.5000000 or NA
    Node 5: resp6-mean = 0.31500000
Node 1: sf12gh > 72.500000 or NA
  Node 3: resp6-mean = 0.44081633

*****************************************************************
Predictor means below are means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
effects and post-selection inference", Statistics in Medicine, v.38, 545-557.
```

```
 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if sf12gh <= 72.500000
 sf12gh mean = 65.921856
 group.DVD       -0.7366E-02  -0.2465       0.8054        0.000       0.3309        1.000
 group.Phone      0.2188E-01   0.7378       0.4608        0.000       0.3413        1.000
 No truncation of predicted values
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if know <= 6.5000000
 know mean = 5.6087154
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if educyrs <= 15.500000
 educyrs mean = 13.800853
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if yearmam <= 3.5000000
 yearmam mean = 2.0055249
 ----------------------------
Node 16: Terminal node
 Coefficients of least squares regression function:
 group.DVD       -0.9843E-01   -1.790       0.7419E-01    0.000       0.3489        1.000
 group.Phone      0.2237E-02   0.4068E-01   0.9676        0.000       0.3489        1.000
 resp6 mean = 0.299766
 No truncation of predicted values
 ----------------------------
Node 17: Terminal node
 Coefficients of least squares regression function:
 group.DVD        0.1154        1.037       0.3019        0.000       0.3362        1.000
 group.Phone      0.2674        2.458       0.1550E-01    0.000       0.3707        1.000
 resp6 mean = 0.637931
 No truncation of predicted values
 ----------------------------
Node 9: Terminal node
 Coefficients of least squares regression function:
 group.DVD        0.2366        2.611       0.9889E-02    0.000       0.3250        1.000
 group.Phone     -0.2165E-01  -0.2244       0.8227        0.000       0.2625        1.000
 resp6 mean = 0.450000
 No truncation of predicted values
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression function:
 group.DVD        0.2883        3.791       0.1993E-03    0.000       0.3500        1.000
```

```
 group.Phone      0.1050        1.321       0.1882        0.000       0.2950        1.000
 resp6 mean = 0.315000
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression function:
 group.DVD       -0.1101       -2.407       0.1634E-01    0.000       0.3156        1.000
 group.Phone     -0.3832E-01   -0.8659      0.3868        0.000       0.3619        1.000
 resp6 mean = 0.440816
 No truncation of predicted values
 ----------------------------
 Number of times Li-Martin approximation used = 3
 Proportion of variance (R-squared) explained by tree model: 0.0579

 Observed and fitted values are stored in gi.fit
 LaTeX code for tree is in gi.tex
```

Figure 25 shows the tree diagram. The tree has 5 terminal nodes (subgroups) and the results for each terminal node give the treatment effects of `DVD` and `Phone` versus `Control`, which is the first treatment level in alphabetical order.

11.2 Censored response: prop. hazards

We now consider a randomized controlled breast cancer trial where the response variable is a censored survival time (Schmoor et al., 1996). The data are in the file `cancerdata.txt`; they are included in the `TH.data` R package (Hothorn, 2017) as well. In the DSC file `cancerdsc.txt` below, the treatment variable is hormone therapy, `horTh`. The variable `time` is (censored) time to recurrence of cancer and the event indicator `event` = 1 if the cancer recurred and = 0 if it did not. Ordinal predictor variables may be designated as "`n`" or "`s`" (with this option of no linear prognostic control, `n` variables are automatically changed to `s` when the program executes). See Loh et al. (2019a, 2016, 2015, 2019c) and Loh and Zhou (2020) for further analysis of the data.

```
cancerdata.txt
NA
1
1 horTh r
2 age n
3 menostat c
4 tsize n
5 tgrade c
6 pnodes n
7 progrec n
```

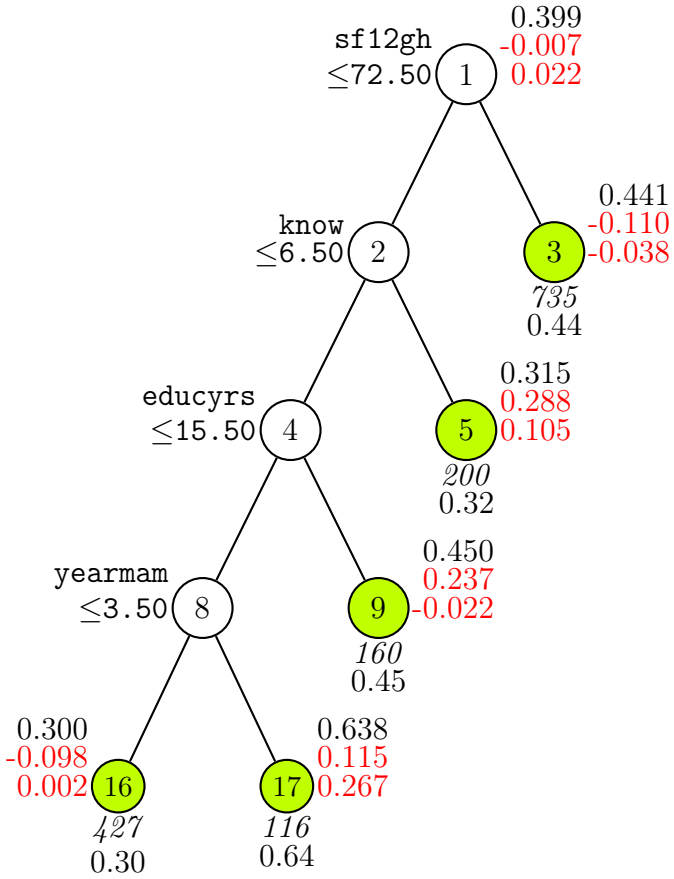

Figure 25: GUIDE v.46.1 0.000-SE (0.250-SE has no splits) least-squares regression tree using Gi option for dependent variable `resp6` and treatment variable `group` without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and mean of `resp6` printed below nodes. `resp6` mean for `group` reference level `Control` followed by `group` effects (in red) of levels `DVD`, `Phone` (relative to `Control`) beside nodes. Second best split variable at root node is `know`.

```
8 estrec n
9 time t
10 event d
```

11.2.1 Without linear prognostic control

The simplest model only uses the covariates to split the intermediate nodes; terminal nodes are fitted with treatment means.

Input file generation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: ph-gi.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1):
 Name of batch output file: ph-gi.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (recommended for prediction)
 Choose 2 for best simple linear in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
 1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: cancerdsc.txt
 Reading DSC file ...
 Training sample file: cancerdata.txt
 Missing value code: NA
 Records in data file start on line 2
 R variable present
 6 N variables changed to S
 Warning: model changed to linear in treatment
 D variable is death
 Reading data file ...
```

```
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: .445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): ph-gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
```

```
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: ph-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: ph-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ph-gi.in
```

Results The contents of `ph-gi.out` follow.

```
Proportional hazards regression tree for censored response
Pruning by cross-validation
DSC file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
  horTh     Uncensored     Censored
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
```

```
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  horTh      r                                 2
      2  age        s    21.00        80.00
      3  menostat   c                                 2
      4  tsize      s    3.000        120.0
      5  tgrade     s    1.000        3.000
      6  pnodes     s    1.000        51.00
      7  progrec    s    0.000        2380.
      8  estrec     s    0.000        1144.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  ================== Constructed variables ==================
     11  lnbasehaz  z   -6.510       0.5887E-01
     12  horTh.yes  f    0.000        1.000

 Numbers of N, F, S, and P below exclude constant variables
      Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        0        1        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
 Number of dummy variables created: 1
 Survival time variable in column: 9
 Event indicator variable in column: 10
 Proportion uncensored among nonmissing T and D variables: 0.445
 Number of cases used for training: 672

 Constant fitted to cases with missing values in regressor variables
 Predictive priority (Gi)
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 No nodewise interaction tests
 Split values for N and S variables based on exhaustive search
 Max number of interaction splits: 100
 Maximum number of split levels: 8
 Minimum node sample size: 6
 Minimum fraction of cases per treatment at each node: 0.072
 Number of iterations for fitting: 20
 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1       18   1.571E+00   7.736E-02   8.051E-02   1.474E+00   1.007E-01
    2       17   1.570E+00   7.758E-02   8.031E-02   1.482E+00   1.007E-01
```

```
   3       16   1.562E+00   7.713E-02   8.018E-02   1.480E+00   9.509E-02
   4       14   1.562E+00   7.713E-02   8.018E-02   1.480E+00   9.509E-02
   5       13   1.559E+00   7.702E-02   7.846E-02   1.480E+00   9.493E-02
   6       12   1.560E+00   7.699E-02   7.817E-02   1.480E+00   9.478E-02
   7       11   1.550E+00   7.681E-02   7.878E-02   1.457E+00   1.003E-01
   8       10   1.547E+00   7.680E-02   7.906E-02   1.457E+00   9.958E-02
   9        9   1.521E+00   7.628E-02   7.936E-02   1.410E+00   1.003E-01
  10        8   1.490E+00   7.465E-02   8.255E-02   1.424E+00   9.213E-02
  11        6   1.463E+00   6.426E-02   6.453E-02   1.425E+00   8.040E-02
  12        4   1.418E+00   5.585E-02   3.363E-02   1.443E+00   4.561E-02
  13**      2   1.398E+00   5.064E-02   1.949E-02   1.400E+00   2.803E-02
  14        1   1.435E+00   5.100E-02   1.066E-02   1.446E+00   1.482E-02

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1      672      672     1  1.807E+03  1.431E+00  progrec
          2T     274      274     1  1.140E+03  1.601E+00  estrec
          3T     398      398     1  2.286E+03  1.188E+00  menostat

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is estrec

Regression tree:

Node 1: progrec <= 21.500000
  Node 2: Median survival time = 1140.0000
Node 1: progrec > 21.500000 or NA
  Node 3: Median survival time = 2286.0000

*************************************************************

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:
```

```
 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if progrec <= 21.500000
 progrec mean = 110.91518
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 horTh.yes   -0.3654       -2.933       0.3471E-02    0.000       0.3601        1.000
 Predicted log-relative hazard = -0.30206062E-02
 ----------------------------
Node 2: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3729
 horTh.yes   -0.1140      -0.6871       0.4926        0.000       0.3613        1.000
 Predicted log-relative hazard = 0.45682185
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant    -0.2596
 horTh.yes   -0.6453       -3.375       0.8098E-03    0.000       0.3593        1.000
 Predicted log-relative hazard = -0.34487497
 ----------------------------
 Observed and fitted values are stored in ph-gi.fit
 LaTeX code for tree is in phi-gi.tex
```

Let $\lambda(u, \mathbf{x})$ denote the hazard function at time u and predictor values $\mathbf{x}$ and let $\lambda_0(u)$ denote the baseline hazard function. The results in `ph-gi.out` show that the fitted proportional hazards model is

$$\begin{aligned}\lambda(u, \mathbf{x}) &= \lambda_0(u)[\exp\{\hat{\beta}_1 + \hat{\gamma}_1 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} \le 21.5) \\ &\quad + \exp\{\hat{\beta}_2 + \hat{\gamma}_2 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} > 21.5)]\end{aligned}$$

with $\hat{\beta}_1 = 0.37292$, $\hat{\gamma}_1 = -0.11404$, $\hat{\beta}_2 = -0.25964$, and $\hat{\gamma}_2 = -0.64531$.

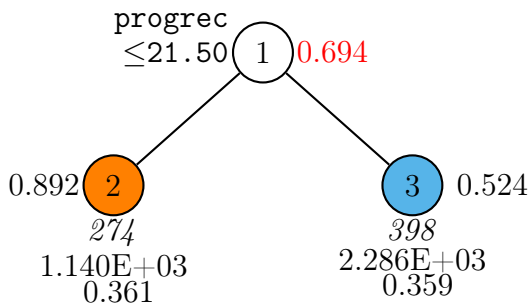

Figure 26: GUIDE v.45.0 0.250-SE proportional hazards regression tree using Gi option for `time` and event indicator `death` and treatment variable `horTh` without adjustment for linear prognostic effects. Constants fitted to incomplete cases in terminal nodes. At each split, an observation goes to the left branch if and only if the condition is satisfied. Treatment `horTh` hazard ratio of level `yes` to level `no` beside nodes. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Terminal nodes with treatment hazard ratio above and below 0.694 (ratio at root node) are painted orange and skyblue respectively. Second best split variable at root node is `estrec`.

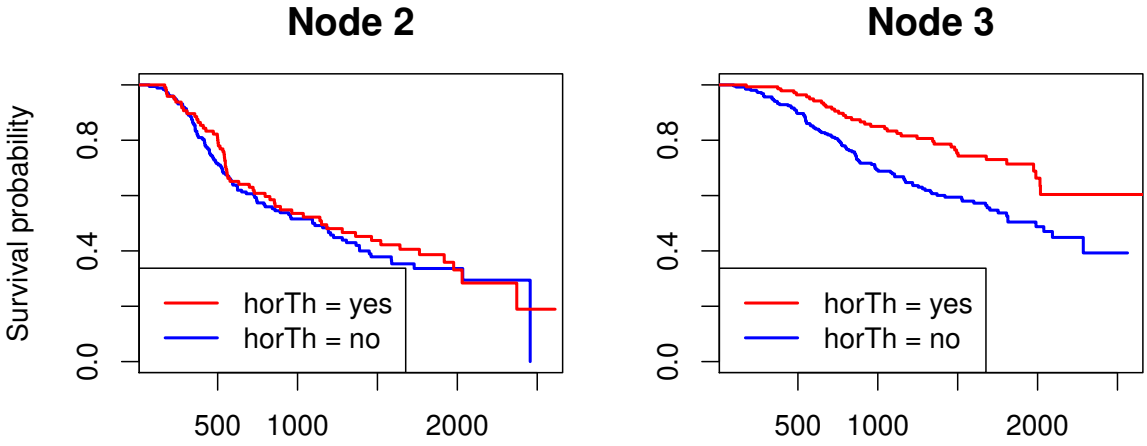

Figure 27: Estimated survival probability functions for breast cancer data

Figure 26 shows the tree diagram. The numbers beside each terminal node are relative hazards of `horTh` = `yes` versus `no`, namely, $\exp(\hat{\gamma}_1) = \exp(-0.11404) = 0.8922223$ for node 2 and $\exp(\hat{\gamma}_2) = \exp(-0.64531) = 0.5244999$ for node 3. Figure 27 shows Kaplan-Meier survival functions of the data in the terminal nodes. The plots are produced by the following R code.

```
library(survival)
z <- read.table("cancerdata.txt",header=TRUE)
leg.txt <- c("horTh = yes","horTh = no")
leg.col <- c("red","blue")
leg.lty <- 1:2
xr <- range(z$time)
zg <- read.table("ph-gi.fit",header=TRUE)
nodes <- zg$node
uniq.gp <- unique(sort(nodes))
plotted <- FALSE
for(g in uniq.gp){
    gp <- nodes == g
    y <- z$time[gp]
    stat <- z$death[gp]
    treat <- z$horTh[gp]
    fit <- survfit(Surv(y,stat) ~ treat, conf.type="none")
    if(plotted){
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="",col=c("blue","red"),lwd=2)
    } else {
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="Survival probability",
             col=c("blue","red"),lwd=2)
        plotted <- TRUE
    }
    title(paste("Node",g))
    legend("bottomleft",legend=leg.txt,lty=1,col=leg.col,lwd=2)
}
```

Estimated relative risks and survival probabilities The file `ph-gi.fit` gives the terminal node number, observed survival time, event indicator (`y`=uncensored, `n`=censored), log baseline cumulative hazard, survival probability, median survival time, and treatment effect (regression coefficient of treatment indicator) of each observation in the training sample (`cancerdata.txt`). The results for the first few observations are shown below.

train	node	observed	event	logbasecumhaz	survivalprob	mediansurvtime	horTh.yes
y	3	1814.00	y	-0.335623	0.576131	2286.00	-0.645311
y	3	2018.00	y	-0.210308	0.720485	2286.00	-0.645311
y	3	712.000	y	-1.28452	0.894065	2286.00	-0.645311

```
y     3  1807.00     y   -0.358191       0.753697          2286.00   -0.645311
y     3  772.000     y    -1.16232       0.785652          2286.00   -0.645311
y     2  448.000     y    -2.08322       0.834592          1140.00   -0.114042
y     3  2172.00     n   -0.121866       0.698971          2286.00   -0.645311
```

11.2.2 Simple linear prognostic control

To reduce or eliminate confounding between treatment and covariate variables, it may be desirable to adjust for the effects of the latter by fitting a regression model that allows for the linear effects of one or more prognostic variables in each node (Loh et al., 2019c). This is done by choosing the "simple linear" or the "multiple linear" option and specifying each potential linear predictor as "n" in the DSC file (no change is needed in `cancerdsc.txt`). First we show how to choose the simple linear model, where a single prognostic variable is used as regressor in each node. There are two options: the **Gi** (default) option is more sensitive to detecting *predictive* variables while the **Gs** option is equally sensitive to detecting *prognostic* variables—see Loh et al. (2015) for definitions.

Input file generation for Gi method

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: lin-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
conversion ([1:3], <cr>=1):
Name of batch output file: lin-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
```

```
Reading DSC file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        6        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: .445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
```

```
Finished reading data file
Warning: missing regressor values imputed with node means
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin-gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: lin-gi.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: lin-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin-gi.in
```

Results for Gi method The following output shows that the pruned tree is trivial with no splits and that the variable `pnodes` is the best simple linear predictor.

```
Proportional hazards regression tree for censored response
No truncation of predicted values
Pruning by cross-validation
DSC file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
  horTh     Uncensored     Censored
   "no"      2456.0000    2563.0000
```

```
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  horTh      r                                 2
      2  age        n    21.00        80.00
      3  menostat   c                                 2
      4  tsize      n    3.000        120.0
      5  tgrade     n    1.000        3.000
      6  pnodes     n    1.000        51.00
      7  progrec    n    0.000        2380.
      8  estrec     n    0.000        1144.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  ================== Constructed variables ==================
     11  lnbasehaz  z   -6.510       0.5887E-01
     12  horTh.yes  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        6        1        0
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Number of dummy variables created: 1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: 0.445
Number of cases used for training: 672

Warning: missing regressor values imputed with node means
Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500
```

```
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       18   1.247E+07   1.219E+07   1.214E+07   2.720E+00   3.919E+06
   2       17   1.247E+07   1.219E+07   1.214E+07   2.720E+00   3.919E+06
   3       15   1.247E+07   1.219E+07   1.214E+07   2.720E+00   3.919E+06
   4       14   1.247E+07   1.219E+07   1.214E+07   2.701E+00   3.919E+06
   5        9   1.247E+07   1.219E+07   1.214E+07   2.616E+00   3.919E+06
   6        8   2.741E+05   2.739E+05   2.591E+05   2.484E+00   6.230E-01
   7        7   2.741E+05   2.739E+05   2.591E+05   1.936E+00   6.723E-01
   8        6   2.741E+05   2.739E+05   2.591E+05   1.542E+00   1.931E-01
  9++       2   1.370E+00   7.295E-02   5.276E-02   1.320E+00   3.197E-02
 10**       1   1.355E+00   5.363E-02   2.719E-02   1.330E+00   2.698E-02

0-SE tree based on mean is marked with * and has 1 terminal node
0-SE tree based on median is marked with + and has 2 terminal node
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
+ tree same as ++ tree
* tree same as ** tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1T     672      672     3  1.807E+03  1.343E+00  estrec
Best split at root node is estrec <= 4.5000

Number of terminal nodes of final tree: 1
Total number of nodes of final tree: 1

Regression tree:
```

```
 Node 1: Median survival time = 1807.0000

 ****************************************************************

Node 1: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 pnodes       0.5630E-01    8.575        0.000        1.000        4.987        51.00
 horTh.yes   -0.3465       -2.778       0.5627E-02    0.000       0.3601        1.000
 ----------------------------
 Observed and fitted values are stored in lin-gi.fit
 Regressor names and coefficients are stored in lin-gi.reg
 LaTeX code for tree is in lin-gi.tex
```

The file `lin-gi.reg` reports the selected regressor in each terminal node of the tree (there is only one node here):

```
node bestvar
1 pnodes
```

Input file generation for Gs method

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: lin-gs.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1):
 Name of batch output file: lin-gs.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (recommended for prediction)
 Choose 2 for best simple linear in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
```

```
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading DSC file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2): 1
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        6        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Survival time variable in column: 9
Event indicator variable in column: 10
```

```
Proportion uncensored among nonmissing T and D variables: .445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Warning: missing regressor values imputed with node means
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin-gs.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: lin-gs.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin-gs.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: lin-gs.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin-gs.in
```

Results for Gs method The Gs method gives a tree with three terminal nodes.

```
Proportional hazards regression tree for censored response
No truncation of predicted values
Pruning by cross-validation
DSC file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
```

```
  horTh     Uncensored     Censored
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      1  horTh      r                                  2
      2  age        n    21.00        80.00
      3  menostat   c                                  2
      4  tsize      n    3.000        120.0
      5  tgrade     n    1.000        3.000
      6  pnodes     n    1.000        51.00
      7  progrec    n    0.000        2380.
      8  estrec     n    0.000        1144.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  ================== Constructed variables ==================
     11  lnbasehaz  z   -6.510       0.5887E-01
     12  horTh.yes  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        6        1        0
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Number of dummy variables created: 1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: 0.445
Number of cases used for training: 672

Warning: missing regressor values imputed with node means
Prognostic priority (Gs)
Pruning by v-fold cross-validation, with v = 10
```

```
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       14   1.940E+00   1.773E-01   1.115E-01   1.892E+00   1.131E-01
   2       13   1.962E+00   1.774E-01   1.183E-01   1.892E+00   1.364E-01
   3       12   1.960E+00   1.774E-01   1.180E-01   1.892E+00   1.342E-01
   4        9   1.809E+00   1.056E-01   1.040E-01   1.790E+00   1.197E-01
   5        8   1.722E+00   1.007E-01   1.006E-01   1.681E+00   1.295E-01
   6        7   1.666E+00   1.006E-01   1.034E-01   1.599E+00   1.750E-01
   7        6   1.565E+00   9.305E-02   1.036E-01   1.424E+00   9.816E-02
   8        4   1.441E+00   6.851E-02   5.627E-02   1.424E+00   7.813E-02
   9**      3   1.336E+00   5.196E-02   3.403E-02   1.289E+00   3.960E-02
  10        2   1.362E+00   5.631E-02   3.638E-02   1.314E+00   5.650E-02
  11        1   1.383E+00   5.502E-02   2.787E-02   1.359E+00   2.776E-02

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1      672      672     3  1.807E+03  1.371E+00  pnodes
          2      370      370     3  2.659E+03+ 1.092E+00  age
          4T     142      142     3  2.563E+03+ 9.548E-01  tsize
          5T     228      228     3  2.030E+03  1.044E+00  tgrade
          3T     302      302     3  9.830E+02  1.552E+00  progrec

Number of terminal nodes of final tree: 3
```

```
 Total number of nodes of final tree: 5
 Second best split variable (based on curvature test) at root node is progrec

 Regression tree:

 Node 1: pnodes <= 3.5000000
   Node 2: age <= 49.500000
     Node 4: Median survival time = 2563.0000+
   Node 2: age > 49.500000 or NA
     Node 5: Median survival time = 2030.0000
 Node 1: pnodes > 3.5000000 or NA
   Node 3: Median survival time = 983.00000

 ****************************************************************

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if pnodes <= 3.5000000
 pnodes mean = 4.9866071
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 pnodes       0.5725E-01    8.744        0.000        1.000        4.987        51.00
 horTh.yes   -0.3528       -2.828       0.4823E-02    0.000       0.3601        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if age <= 49.500000
 age mean = 53.235135
 ----------------------------
Node 4: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      5.162
 age         -0.1344       -5.463       0.2096E-06    21.00        43.00        49.00
 horTh.yes   -0.7981       -1.502       0.1353        0.000       0.1690        1.000
 ----------------------------
```

```
Node 5: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3737
 progrec     -0.3152E-02   -2.547       0.1152E-01    0.000        112.1        1490.
 horTh.yes   -0.6723       -2.877       0.4400E-02    0.000       0.4474        1.000
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1.039
 progrec     -0.2870E-02   -4.036       0.6925E-04    0.000        105.2        2380.
 horTh.yes   -0.3303       -2.112       0.3549E-01    0.000       0.3841        1.000
 ----------------------------
 Observed and fitted values are stored in lin-gs.fit
 Regressor names and coefficients are stored in lin-gs.reg
 LaTeX code for tree is in lin-gs.tex
```

The tree is shown in Figure 28. It does not display the linear predictor selected at each terminal node. This information is given in the file `lin-gs.out` or, more conveniently, in tabular form in `lin-gs.reg` as shown below.

```
node bestvar
4 age
5 progrec
3 progrec
```

11.3 Censored response: restricted mean

Besides a proportional hazards tree, GUIDE can also fit a tree to estimate the restricted mean survival time in each node (Chen and Tsiatis, 2001; Tian et al., 2014). This section shows how this is carried out. The time restriction may be changed by the user during when the input file is created.

11.3.1 Without linear prognostic control

The piecewise-constant Gi tree has no splits when the restricted mean option is chosen.

Input file generation for Gi method

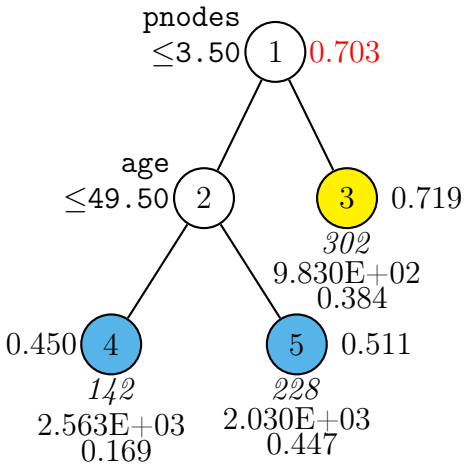

Figure 28: GUIDE v.46.1 0.250-SE proportional hazards regression tree using Gs option for `time` and event indicator `death` and treatment variable `horTh` with adjustment for simple linear prognostic effects. Missing regressor values imputed with node means. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Treatment `horTh` hazard ratio of level `yes` to `no` beside nodes. Terminal nodes with treatment hazard ratio above and below 0.703 (ratio at root node) are painted yellow and skyblue respectively. Second best split variable at root node is `progrec`.

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
conversion ([1:3], <cr>=1):
Name of batch output file: rest-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading DSC file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
```

```
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Smallest observed uncensored time is 72.0000
Largest observed censored or uncensored time is 2659.0000
Input restriction on event time ([72.00:2659.00], <cr>=1222.00):
Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: rest-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: rest-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest-gi.in
```

Results for Gi method

```
Restricted mean event time regression tree
Pruning by cross-validation
DSC file: cancerdsc.txt
```

```
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
  horTh     Uncensored     Censored
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Interval for restricted mean event time is from 0 to 1222.
Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640

Summary information for training sample of size 533 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      1  horTh      r                                  2
      2  age        s    21.00        80.00
      3  menostat   c                                  2
      4  tsize      s    3.000        120.0
      5  tgrade     s    1.000        3.000
      6  pnodes     s    1.000        36.00
      7  progrec    s    0.000        1490.
      8  estrec     s    0.000        1091.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  ================== Constructed variables ==================
     11  horTh.yes  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
```

```
       686          0          0        0        0        1        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Number of dummy variables created: 1
No weight variable in data file
Number of cases used for training: 533

Constant fitted to cases with missing values in regressor variables
Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.073
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       10   4.514E+05   2.208E+04   1.033E+04   4.444E+05   1.310E+04
   2        8   4.514E+05   2.208E+04   1.033E+04   4.444E+05   1.310E+04
   3        6   4.514E+05   2.208E+04   1.033E+04   4.444E+05   1.310E+04
   4        5   4.488E+05   2.182E+04   9.798E+03   4.433E+05   1.329E+04
   5        2   4.408E+05   2.079E+04   8.011E+03   4.414E+05   1.228E+04
   6**      1   4.338E+05   1.732E+04   6.012E+03   4.385E+05   7.335E+03

0-SE tree based on mean is marked with * and has 1 terminal node
0-SE tree based on median is marked with + and has 1 terminal node
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split          Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable       variables
          1T     533      533     2  9.873E+02  1.519E+05  0.0106 estrec
Best split at root node is estrec <= 8.5000
```

```
 Number of terminal nodes of final tree: 1
 Total number of nodes of final tree: 1

 Regression tree:

 Node 1: terminal

 ****************************************************************

Node 1: Terminal node
 Coefficients of least squares regression function:
 horTh.yes     73.85        2.385       0.1744E-01    0.000       0.3591        1.000
 time mean = 987.273
 No truncation of predicted values
 ----------------------------
 Observed and fitted values are stored in rest-gi.fit
 LaTeX code for tree is in rest-gi.tex
```

Results for Gs method The piecewise-constant Gs tree has one split, as shown below.

```
 Restricted mean event time regression tree
 Pruning by cross-validation
 DSC file: cancerdsc.txt
 Training sample file: cancerdata.txt
 Missing value code: NA
 Records in data file start on line 2
 R variable present
 6 N variables changed to S
 Warning: model changed to linear in treatment
 D variable is death
 Piecewise linear model
 Number of records in data file: 686
 Length of longest entry in data file: 4
 Treatment (R) variable is horTh with values "no" and "yes"
 Smallest uncensored time: 72.0000
 Largest uncensored and censored time by horTh
   horTh     Uncensored     Censored
    "no"      2456.0000    2563.0000
   "yes"      2372.0000    2659.0000
 Interval for restricted mean event time is from 0 to 1222.
 Proportion of training sample for each level of horTh
  "no"    0.6360
```

```
"yes"    0.3640

Summary information for training sample of size 533 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      1  horTh      r                                  2
      2  age        s    21.00        80.00
      3  menostat   c                                  2
      4  tsize      s    3.000        120.0
      5  tgrade     s    1.000        3.000
      6  pnodes     s    1.000        36.00
      7  progrec    s    0.000        1490.
      8  estrec     s    0.000        1091.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  ================== Constructed variables ==================
     11  horTh.yes  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        0        1        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Number of dummy variables created: 1
No weight variable in data file
Number of cases used for training: 533

Constant fitted to cases with missing values in regressor variables
Prognostic priority (Gs) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.073
```

```
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       15   4.129E+05   2.146E+04   1.382E+04   4.201E+05   2.357E+04
   2       14   4.129E+05   2.146E+04   1.382E+04   4.201E+05   2.357E+04
   3       13   4.129E+05   2.146E+04   1.382E+04   4.201E+05   2.357E+04
   4       11   4.129E+05   2.146E+04   1.382E+04   4.201E+05   2.357E+04
   5       10   4.127E+05   2.146E+04   1.387E+04   4.201E+05   2.374E+04
   6        8   4.123E+05   2.145E+04   1.405E+04   4.201E+05   2.375E+04
   7        6   4.118E+05   2.136E+04   1.404E+04   4.178E+05   2.348E+04
   8        5   4.090E+05   2.127E+04   1.396E+04   4.020E+05   2.373E+04
   9        4   4.094E+05   2.122E+04   1.393E+04   4.039E+05   2.327E+04
  10        3   4.028E+05   2.096E+04   1.376E+04   4.023E+05   1.971E+04
  11**      2   3.798E+05   1.852E+04   1.523E+04   3.804E+05   1.576E+04
  12        1   4.338E+05   1.732E+04   6.012E+03   4.385E+05   7.335E+03

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split          Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable       variables
          1      533      533     2  9.873E+02  1.519E+05  0.0106 pnodes
          2T     332      332     2  1.073E+03  1.048E+05  0.0129 estrec
          3T     201      201     2  8.312E+02  1.842E+05  0.0174 progrec

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is progrec

Regression tree:

Node 1: pnodes <= 4.5000000
  Node 2: terminal
Node 1: pnodes > 4.5000000 or NA
  Node 3: terminal
```

```
 ************************************************************

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if pnodes <= 4.5000000
 pnodes mean = 4.8475943
 horTh.yes     73.85        2.385       0.1744E-01    0.000       0.3591        1.000
 No truncation of predicted values
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression function:
 horTh.yes     66.83        2.074       0.3884E-01    0.000       0.3483        1.000
 time mean = 1072.91
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression function:
 horTh.yes     106.5        1.879       0.6164E-01    0.000       0.3786        1.000
 time mean = 831.171
 No truncation of predicted values
 ----------------------------
 Observed and fitted values are stored in rest-gs.fit
 LaTeX code for tree is in rest-gs.tex
```

11.3.2 With linear prognostic control

A trivial tree is obtained for both the Gi and Gs methods if a linear regressor is included in each node.

12 Nonrandomized treatments: RHC data

For the RHC data, GUIDE can fit a tree model to find subgroups where the treatment (represented by variable `swang1`) is beneficial or not for survival. This is done by

specifying the treatment variable as "r" and the event variable death (1=die, 0=not die) as "d" in the DSC file rhcdsc3.txt below.

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death d
10 cardiohx c
11 chfhx c
12 dementhx c
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
```

```
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 r
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

12.1 Proportional hazards

GUIDE can fit models with the Gi or Gs options. The Gi option is designed to be sensitive to detect *predictive* variables (variables that have interactions with the treatment variable) while Gs option is equally sensitive to such variables as well as *prognostic* variables (those that have an effect on the outcome irrespective of the treatment). See Loh et al. (2015) for details.

12.1.1 Gi option

Gi input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: surv-gi.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1):
 Name of batch output file: surv-gi.out
```

```
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc3.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
```

```
Number of cases with D=1 and T >= smallest uncensored: 3722
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 W or missing D, T or R variables: 0
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): surv-gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: surv-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: surv-gi.r
Input rank of top variable to split root node ([1:55], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < surv-gi.in
```

Contents of `surv-gi.out`

```
Proportional hazards regression tree for censored response
Pruning by cross-validation
DSC file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
  swang1     Uncensored     Censored
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                #Codes/
                                                Levels/
 Column  Name             Minimum      Maximum   Periods   #Missing
      2  cat1        c                                9
      3  cat2        c                                6     4535
      4  ca          c                                3
      9  death       d    0.000       1.000
     10  cardiohx    c                                2
     11  chfhx       c                                2
      :
     56  seps        c                                2
```

```
     57  trauma      c                                   2
     58  ortho       c                                   2
     59  adld3p      s    0.000        7.000                    4296
     60  urin1       s    0.000        9000.                    3028
     61  race        c                                   3
     62  income      c                                   4
     64  survtime    t    2.000        1943.
  ================== Constructed variables ==================
     65  lnbasehaz0  z   -3.818        2.038
     66  swang1.RHC  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        1       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Number of dummy variables created: 1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of cases excluded due to 0 W or missing D, T or R variables: 0

Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.076
Number of iterations for fitting: 20
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       21   1.401E+00   2.992E-02   3.401E-02   1.353E+00   3.843E-02
   2       20   1.396E+00   2.973E-02   3.530E-02   1.353E+00   4.072E-02
   3       19   1.387E+00   2.888E-02   3.537E-02   1.353E+00   3.181E-02
   4       17   1.359E+00   2.024E-02   1.939E-02   1.352E+00   1.744E-02
   5       15   1.356E+00   2.002E-02   1.931E-02   1.349E+00   1.559E-02
   6       11   1.354E+00   1.976E-02   1.946E-02   1.337E+00   1.324E-02
   7       10   1.353E+00   1.974E-02   1.937E-02   1.338E+00   1.235E-02
   8        9   1.340E+00   1.626E-02   9.638E-03   1.336E+00   9.990E-03
```

```
   9        8   1.338E+00   1.624E-02   1.003E-02   1.337E+00   9.677E-03
  10        7   1.338E+00   1.621E-02   1.009E-02   1.337E+00   9.949E-03
  11--      6   1.327E+00   1.561E-02   1.001E-02   1.329E+00   1.023E-02
  12**      5   1.330E+00   1.552E-02   8.802E-03   1.330E+00   1.108E-02
  13        1   1.367E+00   1.526E-02   6.317E-03   1.358E+00   9.980E-03

0-SE tree based on mean is marked with * and has 6 terminal nodes
0-SE tree based on median is marked with + and has 6 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1    5735     5735     1  1.920E+02  1.367E+00  ph1
          2    1411     1411     1  1.150E+02  1.454E+00  cat2
          4T   1307     1307     1  1.570E+02  1.416E+00  paco21
          5T    104      104     1  1.400E+01  1.636E+00  malighx
          3    4324     4324     1  2.070E+02  1.334E+00  resp1
          6    3341     3341     1  2.200E+02  1.333E+00  paco21
         12T    687      687     1  6.900E+01  1.531E+00  income
         13T   2654     2654     1  2.390E+02  1.265E+00  paco21
          7T    983      983     1  1.640E+02  1.319E+00  hrt1

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9
Second best split variable (based on curvature test) at root node is resp1

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: ph1 <= 7.3344730
  Node 2: cat2 = "MOSF w/Sepsis", "NA"
    Node 4: Median survival time = 157.00000
  Node 2: cat2 /= "MOSF w/Sepsis", "NA"
    Node 5: Median survival time = 14.000000
Node 1: ph1 > 7.3344730 or NA
```

```
   Node 3: resp1 <= 38.500000 or NA
     Node 6: paco21 <= 29.498050
       Node 12: Median survival time = 69.000000
     Node 6: paco21 > 29.498050 or NA
       Node 13: Median survival time = 239.00000
   Node 3: resp1 > 38.500000
     Node 7: Median survival time = 164.00000

 ************************************************************
 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if ph1 <= 7.3344730
 ph1 mean = 7.3884135
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.000
 swang1.RHC    0.1504        4.494       0.7131E-05    0.000       0.3808        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if cat2 = "MOSF w/Sepsis", "NA"
 cat2 mode = "NA"
 ----------------------------
Node 4: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.6181E-01
 swang1.RHC    0.4067        6.034       0.2086E-08    0.000       0.4499        1.000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.8005
 swang1.RHC   -0.3295       -1.558       0.1223        0.000       0.3558        1.000
```

```
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if resp1 <= 38.500000 or NA
 resp1 mean = 28.418652
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if paco21 <= 29.498050
 paco21 mean = 36.054906
 ----------------------------
Node 12: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.3006
 swang1.RHC   -0.3237E-01  -0.3424       0.7322        0.000       0.3916        1.000
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.7105E-01
 swang1.RHC    0.5937E-02   0.1159       0.9078        0.000       0.3632        1.000
 ----------------------------
Node 7: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.1150E-01
 swang1.RHC    0.3555        4.329       0.1651E-04    0.000       0.3316        1.000
 ----------------------------
 Observed and fitted values are stored in surv-gi.fit
 LaTeX code for tree is in surv-gi.tex
```

Figure 29 shows the tree and Figure 30 shows the estimated survival curves in its terminal nodes. The R code for making the plots is given below.

```
library(survival)
z0 <- read.table("rhcdata.txt",header=TRUE)
par(mar=c(3,4,3,1),mfrow=c(2,3),cex=1)
leg.txt <- c("NoRHC","RHC"); leg.col <- c("blue","red"); leg.lty <- 2:1
xr <- range(z0$survtime)
zg <- read.table("surv-gi.fit",header=TRUE)
nodes <- zg$node
uniq.gp <- unique(sort(nodes))
ii <- 0
for(g in uniq.gp){
    ii <- ii+1
    gp <- nodes == g
```

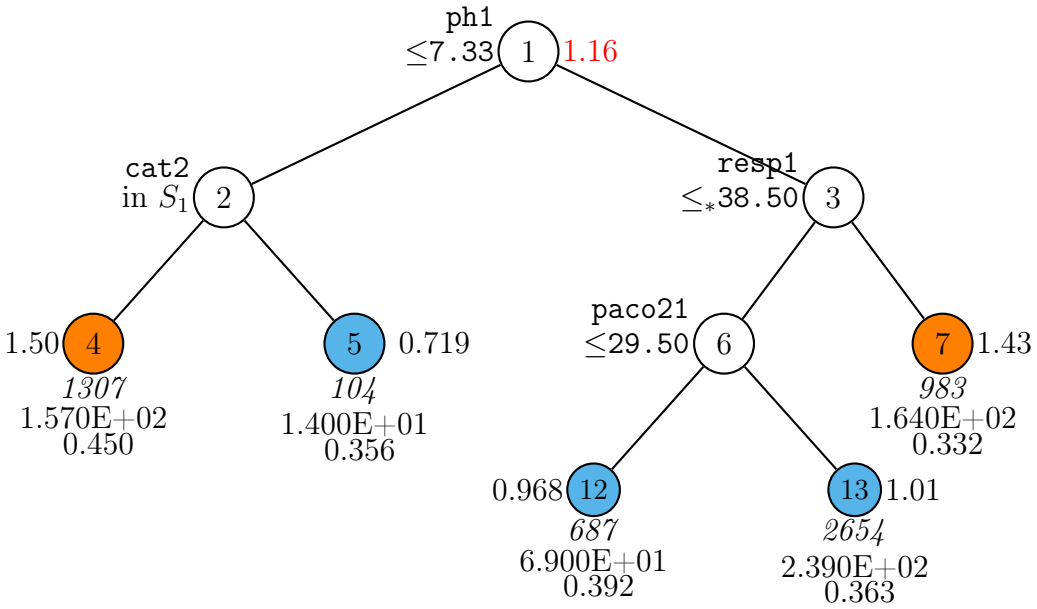

Figure 29: GUIDE v.46.1 0.250-SE proportional hazards regression tree using Gi option for `survtime` and event indicator `death` and treatment variable `swang1` without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`MOSF w/Sepsis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to level `NoRHC` beside nodes. Sample size (in *italics*), median survival time, and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with treatment hazard ratio above and below 1.162 (ratio at root node) are painted orange and skyblue respectively. Second best split variable at root node is `resp1`.

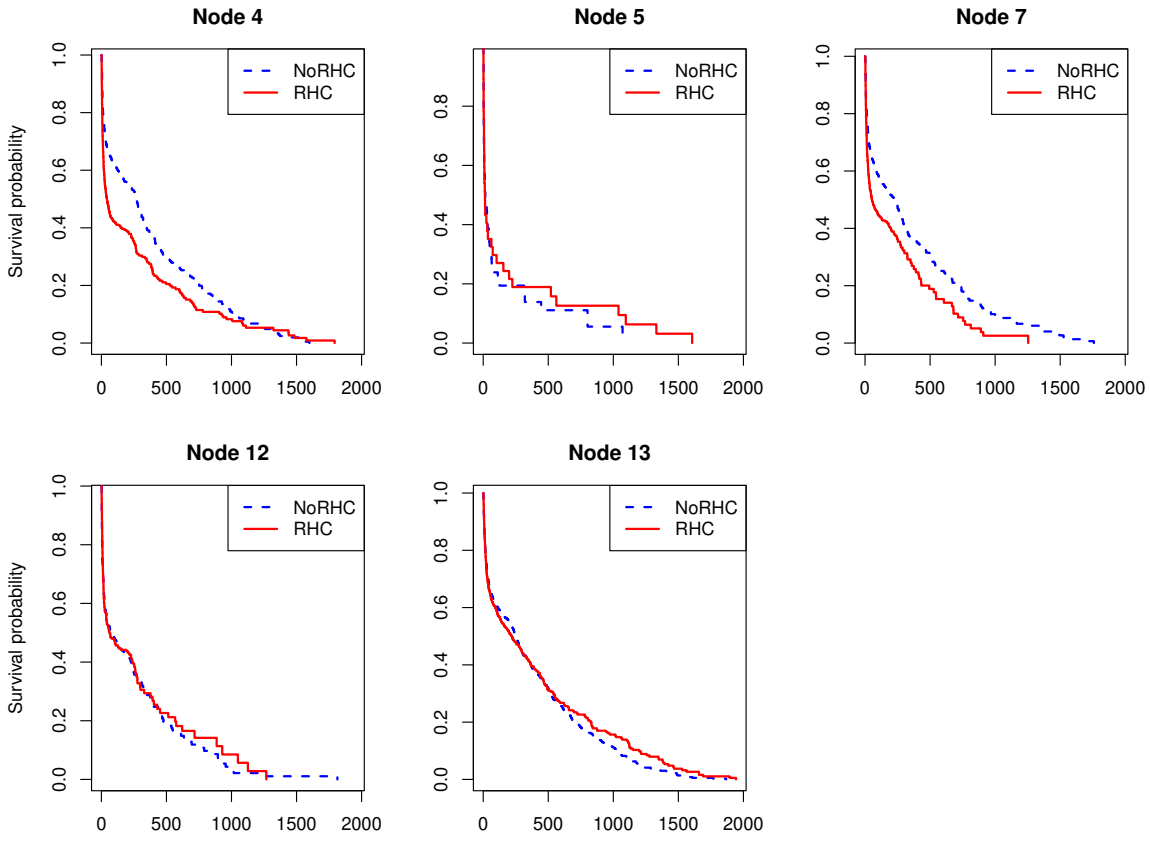

Figure 30: Survival curves for RHC data in nodes of Figure 29

```
    y <- z0$survtime[gp]
    stat <- z0$death[gp]
    treat <- z0$swang1[gp]
    fit <- survfit(Surv(y,stat) ~ treat, conf.type="none")
    if(g == 4 | g == 12){
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="Survival probability",
             col=leg.col,lwd=2,lty=leg.lty)
    } else {
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="",col=leg.col,lwd=2,lty=leg.lty)
    }
    title(paste("Node",g))
    legend("topright",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2)
}
```

Following are the top 3 lines of the file `surv-gi.fit`

```
train  node  observed  event logbasecumhaz   survivalprob  mediansurvtime  swang1.RHC
    y    13  240.000     n   -0.269165        0.490850          239.000    0.593672E-002
    y     4  45.0000     y   -0.757608        0.515901          157.000    0.406690
    y     7  317.000     n   -0.633003E-001   0.266047          164.000    0.355517
```

The column definitions are

`train`: `y` if the observation is used for model fitting, `n` if not.

`node`: terminal node label of observation.

`observed`: observed survival time t.

`event`: `y` if uncensored (death), `n` if censored.

`logbasecumhaz`: log of the estimated baseline cumulative hazard function $\log \Lambda_0(t) = \log \int_0^t \lambda_0(u)\, du$ at observed time t.

`survivalprob`: probability that the subject survives up to observed time t. For the first subject, this is

$$\begin{aligned} \exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\beta_0 + \texttt{logbasecumhaz})\} \\ &= \exp(-\exp(-0.242135921383 - 0.3029494)) \\ &= 0.5600147 \end{aligned}$$

where $t = 240$ and $\beta_0 = -0.242135921383$ is the constant term in the node (`surv-gs.r` gives β_0 to higher precision than `surv-gs.out`).

`mediansurvtime`: median survival time among observations in node estimated from Kaplan-Meier survival function. A trailing plus (+) sign indicates estimate is censored.

`swang1.RHC`: estimated treatment effect β_1 for level `RHC` of `swang1`.

12.2 Restricted mean

GUIDE can also construct a tree model such that a restricted mean event time (Chen and Tsiatis, 2001; Tian et al., 2014) is fitted in each node of the tree.

12.2.1 Gi option

Gi input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: rest-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc3.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
```

```
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
GUIDE will try to create the variables in the DSC file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=622.00):
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
```

```
No weight variable in data file
Number of cases used for training: 3763
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 W or missing D or R variables: 1972
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-gi.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: rest-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: rest-gi.r
Input rank of top variable to split root node ([1:55], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest-gi.in
```

Contents of `rest-gi.out`

```
Restricted mean event time regression tree
Pruning by cross-validation
DSC file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
  swang1     Uncensored     Censored
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Interval for restricted mean event time is from 0 to 622.
```

```
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007

Summary information for training sample of size 3763 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name             Minimum      Maximum    Periods   #Missing
      2  cat1        c                                 9
      3  cat2        c                                 6     2836
      4  ca          c                                 3
      9  death       d    0.000        1.000
     10  cardiohx    c                                 2
      :
     58  ortho       c                                 2
     59  ad1d3p      s    0.000        7.000                  3041
     60  urin1       s    0.000        9000.                  2115
     61  race        c                                 3
     62  income      c                                 4
     64  survtime    t    2.000        1943.
  ================== Constructed variables ==================
     65  swang1.RHC  f    0.000        1.000

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        1       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Number of dummy variables created: 1
No weight variable in data file
Number of cases used for training: 3763
Number of cases excluded due to 0 W or missing D or R variables: 1972

Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
```

```
Max number of interaction splits: 100
Maximum number of split levels: 8
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.080
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       17   1.194E+05   3.630E+03   2.243E+03   1.185E+05   1.903E+03
   2       16   1.194E+05   3.631E+03   2.232E+03   1.185E+05   1.902E+03
   :
  14        3   1.185E+05   3.572E+03   1.952E+03   1.178E+05   1.228E+03
  15**      2   1.154E+05   3.251E+03   1.536E+03   1.149E+05   1.343E+03
  16        1   1.198E+05   3.143E+03   9.972E+02   1.190E+05   1.421E+03

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split          Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable       variables
          1     3763     3763     2  2.583E+02  9.489E+04  0.0043 scoma1
          2T    3124     3124     2  2.781E+02  9.938E+04  0.0075 pafi1
          3T     639      639     2  1.333E+02  4.975E+04  0.0016 sod1

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is ph1

Regression tree:

Node 1: scoma1 <= 49.500000
  Node 2: terminal
Node 1: scoma1 > 49.500000 or NA
  Node 3: terminal

*************************************************************
Predictor means below are means of cases with no missing values.
```

```
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if scoma1 <= 49.500000
 scoma1 mean = 20.462797
 swang1.RHC    -33.80       -4.020       0.5926E-04    0.000       0.3808        1.000
 No truncation of predicted values
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression function:
 swang1.RHC    -44.75       -4.866       0.1195E-05    0.000       0.3949        1.000
 survtime mean = 278.051
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression function:
 swang1.RHC    -17.66       -1.003       0.3161        0.000       0.2916        1.000
 survtime mean = 133.272
 No truncation of predicted values
 ----------------------------
 Number of times Li-Martin approximation used = 51
 Observed and fitted values are stored in rest-gi.fit
 LaTeX code for tree is in rest-gi.tex
```

Figure 31 shows the Gi restricted mean event time tree.

13 Multiresponse: NMES data

GUIDE has two options for fitting a piecewise-constant regression model to predict two or more dependent variables simultaneously (Loh and Zheng, 2013). The first (named `multiresponse` or option `5` in the input file) requires the number of dependent variables to be the same for each observation. Observations with missing values

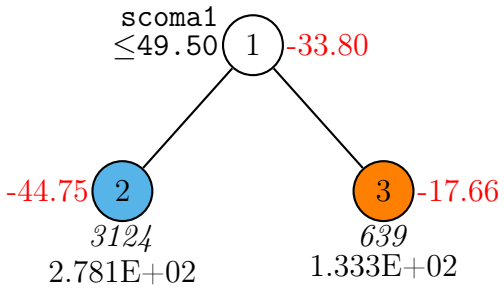

Figure 31: GUIDE v.46.1 0.250-SE regression tree using Gi option for mean `survtime` restricted to less than 622.00 and treatment variable `swang1` without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and mean of `death` printed below nodes. `swang1` effects (in red, relative to reference level `NoRHC`) of levels `RHC` (relative to `NoRHC`) beside nodes. Terminal nodes with treatment effect above and below -33.80 (effect at root node) are painted orange and skyblue respectively. Second best split variable at root node is `ph1`.

in one or more dependent variables are excluded. The second (named `longitudinal data (with T variables)` or option `6` in the input file) requires each dependent variable to be associated with an observation time variable. It fits a model to all observations, including those with missing values in some dependent variables. The observation times are not required to be the same for all subjects, i.e., they may vary from subject to subject, but observations with missing times are excluded from model fitting. We demonstrate the first option in this section. The second option is used in Section 14.

The data file `nmes.txt` contains observations on 4406 subjects from a National Medical Expenditure Survey (NMES) conducted in 1987 and 1988. Table 10 gives the names of the variables and their definitions. The data were previously analyzed in Deb and Trivedi (1997), Cameron and Trivedi (1998, chap. 6), and Zeileis (2006). Here we construct a regression tree to predict the outcomes for the first 6 variables (`ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`). The contents of the description file `nmes.dsc` follow.

```
nmes.txt
NA
1
1 ofp d
2 ofnp d
3 opp d
4 opnp d
```

Table 10: Definitions of variables in NMES data

ofp	number of physician office visits
ofnp	number of nonphysician office visits
opp	number of physician outpatient visits
opnp	number of nonphysician outpatient visits
emer	number of emergency room visits
hosp	number of hospitalizations
health	self-perceived health (poor, average, or excellent)
numchron	number of chronic conditions
adldiff	has condition that limits daily living (no, yes)
region	region of U.S. (midwest, northeast, west, other)
age	age in years
black	African American (no, yes)
gender	sex (female, male)
married	married (no, yes)
school	number of years of education
faminc	family income in $10,000
employed	employed (no, yes)
privins	covered by private insurance (no, yes)
medicaid	covered by Medicaid (no, yes)

```
5 emer d
6 hosp d
7 health c
8 numchron n
9 adldiff c
10 region c
11 age n
12 black c
13 gender c
14 married c
15 school n
16 faminc n
17 employed c
18 privins c
19 medicaid c
```

13.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: mult.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1):
 Name of batch output file: mult.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 5
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: nmes.dsc
 Reading DSC file ...
 Training sample file: nmes.txt
 Missing value code: NA
 Records in data file start on line 1
 4 N variables changed to S
 Number of D variables: 6
 D variables are:
 ofp
```

```
ofnp
opp
opnp
emer
hosp
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):

Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):

Reading data file ...
Number of records in data file: 4406
Length of longest entry in data file: 9
Checking for missing values ...
Finished checking
Assigning integer codes to values of 9 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Normalizing data
Rereading data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2): 1
#cases w/ miss. D = number of cases with some D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4406          0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        9        0        1
Number of cases used for training: 4406
Number of split variables: 13
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): mult.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save node IDs of observations, 1 otherwise ([1:2], <cr>=2):
```

```
Input name of file to store terminal node ID of each case: mult.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=2):
Input name of file to store node fitted values: mult.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: mult.r
Input rank of top variable to split root node ([1:13], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mult.in
```

13.2 Contents of mult.out

```
Multi-response or longitudinal data without T variables
Pruning by cross-validation
DSC file: nmes.dsc
Training sample file: nmes.txt
Missing value code: NA
Records in data file start on line 1
4 N variables changed to S
Number of D variables: 6
Univariate split variable selection method
Mean-squared errors (MSE) are calculated from normalized D variables
D variables equally weighted
Piecewise constant model
Number of records in data file: 4406
Length of longest entry in data file: 9
Model fitted to subset of observations with complete D values
PCA on dependent variables for split selection

Summary information for training sample of size 4406
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                              #Codes/
                                              Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  ofp        d    0.000        89.00
      2  ofnp       d    0.000        104.0
      3  opp        d    0.000        141.0
      4  opnp       d    0.000        155.0
      5  emer       d    0.000        12.00
      6  hosp       d    0.000        8.000
      7  health     c                                3
      8  numchron   s    0.000        8.000
```

```
      9  adldiff    c                                  2
     10  region     c                                  4
     11  age        s    6.600        10.90
     12  black      c                                  2
     13  gender     c                                  2
     14  married    c                                  2
     15  school     s    0.000        18.00
     16  faminc     s   -1.012        54.84
     17  employed   c                                  2
     18  privins    c                                  2
     19  medicaid   c                                  2

#cases w/ miss. D = number of cases with some D values missing
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4406          0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 4406

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 44
Ranks of variables and their 1-df chi-squared values at root node
     1  0.5227E+03   numchron
     2  0.3280E+03   health
     3  0.1718E+03   adldiff
     4  0.6060E+02   privins
     5  0.5697E+02   region
     6  0.5290E+02   age
     7  0.5172E+02   medicaid
     8  0.3676E+02   school
     9  0.2714E+02   gender
    10  0.2577E+02   black
    11  0.1036E+02   faminc
    12  0.9199E+01   married
    13  0.7753E+01   employed

Size and CV Loss and SE of subtrees:
```

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	19	1.639E+00	1.305E-01	1.434E-01	1.428E+00	1.081E-01
2	18	1.620E+00	1.304E-01	1.461E-01	1.428E+00	1.073E-01
3	17	1.617E+00	1.304E-01	1.464E-01	1.428E+00	1.087E-01
4	16	1.592E+00	1.304E-01	1.471E-01	1.428E+00	8.733E-02
5	15	1.572E+00	1.304E-01	1.477E-01	1.428E+00	6.736E-02
6	13	1.528E+00	1.304E-01	1.542E-01	1.339E+00	9.682E-02
7	11	1.497E+00	1.304E-01	1.567E-01	1.301E+00	8.848E-02
8	10	1.468E+00	1.304E-01	1.589E-01	1.295E+00	6.558E-02
9	8	1.417E+00	1.304E-01	1.443E-01	1.268E+00	7.670E-02
10	7	1.362E+00	1.304E-01	1.525E-01	1.225E+00	9.328E-02
11	6	1.122E+00	1.296E-01	1.491E-01	9.523E-01	9.815E-02
12++	4	1.050E+00	1.294E-01	1.475E-01	8.972E-01	1.046E-01
13**	3	1.063E+00	1.293E-01	1.422E-01	9.711E-01	7.747E-02
14	2	1.259E+00	1.296E-01	1.461E-01	1.068E+00	9.920E-02
15	1	1.635E+00	1.308E-01	1.448E-01	1.421E+00	1.078E-01

```
0-SE tree based on mean is marked with * and has 4 terminal nodes
0-SE tree based on median is marked with + and has 4 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as -- tree
+ tree same as ++ tree
* tree same as ++ tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases      Node    Split
      label    cases      fit      MSE     variable
          1     4406     4406  1.000E+00   numchron
          2T    2523     2523  5.688E-01   numchron
          3     1883     1883  1.528E+00   health
          6T     426      426  2.282E+00   medicaid
          7T    1457     1457  1.277E+00   region

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5
Second best split variable (based on curvature test) at root node is health

Regression tree for multi-response data:
```

```
 For categorical variable splits, values not in training data go to the right

 Node 1: numchron <= 1.5000000
   Node 2: Mean cost = 0.56857139
 Node 1: numchron > 1.5000000 or NA
   Node 3: health = "poor"
     Node 6: Mean cost = 2.2768607
   Node 3: health /= "poor"
     Node 7: Mean cost = 1.2765754

 ****************************************************************

Node 1: Intermediate node
 A case goes into Node 2 if numchron <= 1.5000000
 numchron mean = 1.5419882
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   5.7744E+00   1.6180E+00   7.5079E-01   5.3609E-01   2.6350E-01
   2.9596E-01
 ----------------------------
Node 2: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   4.4392E+00   1.4491E+00   4.6968E-01   3.9516E-01   1.6488E-01
   1.6647E-01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if health = "poor"
 health mode = "average"
 ----------------------------
Node 6: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   9.4319E+00   1.5000E+00   1.5282E+00   6.8310E-01   7.2066E-01
   7.9108E-01
 ----------------------------
Node 7: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   7.0172E+00   1.9451E+00   1.0103E+00   7.3713E-01   3.0062E-01
   3.7543E-01
 ----------------------------
 Case and node IDs are in file: mult.nid
 Node fitted values are in file: mult.fit
 LaTeX code for tree is in mult.tex
```

The tree is shown in Figure 32. The file `mult.fit` saves the mean values of the dependent variables in each terminal node:

```
  node    ofp          ofnp         opp          opnp         emer         hosp
```

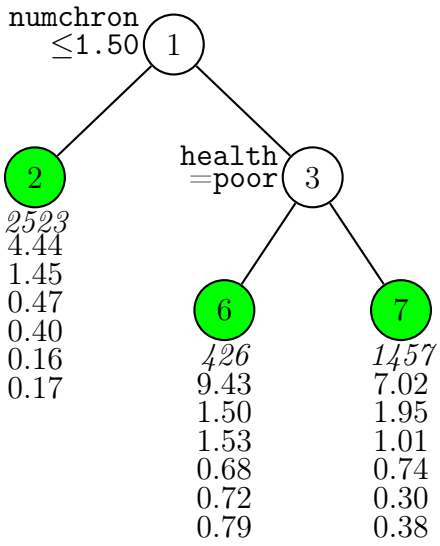

Figure 32: GUIDE v.46.1 0.250-SE regression tree for predicting response variables `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`, using PCA at each node. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and predicted values of `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp` printed below nodes. Second best split variable at root node is `health`.

```
2  0.44392E+01  0.14491E+01  0.46968E+00  0.39516E+00  0.16488E+00  0.16647E+00
6  0.94319E+01  0.15000E+01  0.15282E+01  0.68310E+00  0.72066E+00  0.79108E+00
7  0.70172E+01  0.19451E+01  0.10103E+01  0.73713E+00  0.30062E+00  0.37543E+00
```

The file `mult.nid` gives the terminal node number for each observation, including those that are not used to construct the tree (indicated by the letter "`n`" in the `train` column of the file).

14 Longitudinal response with varying times

The data come from a longitudinal study on the hourly wage of 888 male high-school dropouts (246 black, 204 Hispanic, 438 white), where the observation time points as well as their number (1–13) varied across individuals (Murnane et al., 1999; Singer and Willett, 2003). An earlier version of GUIDE was used to analyze the data in Loh and Zheng (2013).

The response variable is hourly wage (in 1990 dollars) and the predictor variables are `hgc` (highest grade completed; 6–12), `exper` (years in labor force; 0.001–12.7 yrs), and `race` (Black, Hispanic, and White). The data file `wagedat.txt` is in **wide format**, where each record refers to one individual. The DSC file `wagedsc.txt` is given below. Observation time points are indicated by `t`. The `d` and `t` variable columns may appear anywhere in the data, but the first `d` must be associated with the first `t`, second `d` with the second `t`, and so on. The number of `d` and `t` variables must be the same. Missing `d` values are permitted to allow for observations with unequal numbers of observation times. Observations with missing values in one or more `t` variable are excluded from model fitting.

```
wagedat.txt
NA
1
1 id x
2 hgc n
3 exper1 t
4 exper2 t
5 exper3 t
6 exper4 t
7 exper5 t
8 exper6 t
9 exper7 t
10 exper8 t
11 exper9 t
```

```
12 exper10 t
13 exper11 t
14 exper12 t
15 exper13 t
16 postexp1 x
17 postexp2 x
18 postexp3 x
19 postexp4 x
20 postexp5 x
21 postexp6 x
22 postexp7 x
23 postexp8 x
24 postexp9 x
25 postexp10 x
26 postexp11 x
27 postexp12 x
28 postexp13 x
29 wage1 d
30 wage2 d
31 wage3 d
32 wage4 d
33 wage5 d
34 wage6 d
35 wage7 d
36 wage8 d
37 wage9 d
38 wage10 d
39 wage11 d
40 wage12 d
41 wage13 d
42 ged1 x
43 ged2 x
44 ged3 x
45 ged4 x
46 ged5 x
47 ged6 x
48 ged7 x
49 ged8 x
50 ged9 x
51 ged10 x
52 ged11 x
53 ged12 x
54 ged13 x
55 uerate1 x
56 uerate2 x
57 uerate3 x
```

```
58 uerate4 x
59 uerate5 x
60 uerate6 x
61 uerate7 x
62 uerate8 x
63 uerate9 x
64 uerate10 x
65 uerate11 x
66 uerate12 x
67 uerate13 x
68 race c
```

14.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: wage.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1):
 Name of batch output file: wage.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 6
 Input 1 for lowess smoothing, 2 for spline smoothing ([1:2], <cr>=1):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: wagedsc.txt
 Reading DSC file ...
 Training sample file: wagedat.txt
 Missing value code: NA
 Records in data file start on line 1
 One N variable changed to S
 Number of D variables: 13
 D variables are:
 wage1
 wage2
 wage3
 wage4
```

```
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
D variables can be grouped into segments to look for patterns
Input 1 for equal-sized groups, 2 for custom groups ([1:2], <cr>=1):
Input number of roughly equal-sized groups ([2:9], <cr>=3):
Input number of interpolating points for prediction ([10:100], <cr>=31):
Reading data file ...
Number of records in data file: 888
Length of longest entry in data file: 16
Checking for missing values ...
Finished checking
Missing values found in D variables
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       888          0          0       40        0        0        1
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
```

```
Number of cases used for training: 888
Number of split variables: 2
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): wage.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=3):
Input file name: wage.var
Input 2 to save node IDs of observations, 1 otherwise ([1:2], <cr>=2):
Input name of file to store terminal node ID of each case: wage.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=2):
Input name of file to store node fitted values: wage.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: wage.r
Input rank of top variable to split root node ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < wage.in
```

14.2 Contents of wage.out

```
Longitudinal data with T variables
Lowess smoothing
Pruning by cross-validation
DSC file: wagedsc.txt
Training sample file: wagedat.txt
Missing value code: NA
Records in data file start on line 1
One N variable changed to S
Number of D variables: 13
Number of D variables: 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
```

```
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
Number of records in data file: 888
Length of longest entry in data file: 16
Missing values found in D variables
Model fitted to subset of observations with complete D values

Summary information for training sample of size 888
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      2  hgc        s     6.000        12.00
      3  exper1     t   0.1000E-02     5.637
      4  exper2     t     0.000        7.584                    38
      5  exper3     t     0.000        9.777                    77
      6  exper4     t     0.000        10.81                   124
      7  exper5     t     0.000        11.78                   159
      8  exper6     t     0.000        10.59                   233
      9  exper7     t     0.000        11.28                   325
     10  exper8     t     0.000        10.58                   428
     11  exper9     t     0.000        11.62                   551
     12  exper10    t     0.000        12.26                   678
     13  exper11    t     0.000        11.98                   791
     14  exper12    t     0.000        12.56                   856
     15  exper13    t     0.000        12.70                   882
     29  wage1      d     2.030        68.65
```

```
     30  wage2      d    2.069        50.40                   38
     31  wage3      d    2.046        34.50                   77
     32  wage4      d    2.117        33.15                  124
     33  wage5      d    2.104        49.30                  159
     34  wage6      d    2.208        74.00                  233
     35  wage7      d    2.104        47.28                  325
     36  wage8      d    2.316        37.71                  428
     37  wage9      d    2.529        46.11                  551
     38  wage10     d    2.998        56.54                  678
     39  wage11     d    4.084        22.20                  791
     40  wage12     d    3.432        46.20                  856
     41  wage13     d    4.563        7.776                  882
     68  race       c                                3

 Numbers of N, F, S, and P below exclude constant variables
      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
        888          0          0       40        0        0        1
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0        1        0
 Number of cases used for training: 888
 Number of cases excluded due to 0 W or missing D variable: 0

 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 No nodewise interaction tests
 Split values for N and S variables based on exhaustive search
 Max number of interaction splits: 100
 Maximum number of split levels: 15
 Minimum node sample size: 10
 Ranks of variables and their 1-df chi-squared values at root node
      1  0.1235E+02   hgc
      2  0.6915E+01   race

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1       17   1.283E+02   1.051E+01   9.475E+00   1.261E+02   1.045E+01
    2       16   1.283E+02   1.051E+01   9.475E+00   1.261E+02   1.045E+01
    3       15   1.253E+02   1.048E+01   9.841E+00   1.218E+02   1.118E+01
    4       13   1.253E+02   1.048E+01   9.841E+00   1.217E+02   1.118E+01
    5       10   1.254E+02   1.058E+01   9.880E+00   1.224E+02   1.126E+01
    6*       8   1.240E+02   1.055E+01   9.851E+00   1.205E+02   1.108E+01
    7+       7   1.243E+02   1.064E+01   1.002E+01   1.205E+02   1.125E+01
    8**      1   1.244E+02   1.065E+01   1.011E+01   1.210E+02   1.171E+01
```

```
0-SE tree based on mean is marked with * and has 8 terminal nodes
0-SE tree based on median is marked with + and has 7 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree

WARNING: tree based on mean CV estimate of error has no splits
Choosing smallest nontrivial tree with no larger CV error estimate

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases      Node    Split
      label    cases      fit      MSE     variable
          1      888      888  1.222E+02   hgc
          2      873      873  1.215E+02   hgc
          4T      32       32  9.598E+01   race
          5      841      841  1.226E+02   race
         10T     233      233  1.159E+02   hgc
         11      608      608  1.248E+02   race
         22      192      192  1.251E+02   hgc
         44T     118      118  1.017E+02   hgc
         45T      74       74  1.569E+02   hgc
         23      416      416  1.247E+02   hgc
         46T     284      284  1.004E+02   hgc
         47T     132      132  1.718E+02   hgc
          3T      15       15  1.565E+02   -

Number of terminal nodes of final tree: 7
Total number of nodes of final tree: 13
Second best split variable (based on curvature test) at root node is race

Regression tree for longitudinal data:
For categorical variable splits, values not in training data go to the right

Node 1: hgc <= 11.500000
  Node 2: hgc <= 6.5000000
    Node 4: Mean cost = 92.977093
  Node 2: hgc > 6.5000000 or NA
    Node 5: race = "black"
      Node 10: Mean cost = 115.39395
```

```
     Node 5: race /= "black"
       Node 11: race = "hispanic"
         Node 22: hgc <= 9.5000000
           Node 44: Mean cost = 100.87229
         Node 22: hgc > 9.5000000 or NA
           Node 45: Mean cost = 154.79614
       Node 11: race /= "hispanic"
         Node 23: hgc <= 9.5000000
           Node 46: Mean cost = 100.02686
         Node 23: hgc > 9.5000000 or NA
           Node 47: Mean cost = 170.51645
 Node 1: hgc > 11.500000 or NA
   Node 3: Mean cost = 146.10583

 ***************************************************************

Node 1: Intermediate node
 A case goes into Node 2 if hgc <= 11.500000
 hgc mean = 8.9166667
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if hgc <= 6.5000000
 hgc mean = 8.8636884
 ----------------------------
Node 4: Terminal node
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if race = "black"
 race mode = "white"
 ----------------------------
Node 10: Terminal node
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if race = "hispanic"
 race mode = "white"
 ----------------------------
Node 22: Intermediate node
 A case goes into Node 44 if hgc <= 9.5000000
 hgc mean = 9.0000000
 ----------------------------
Node 44: Terminal node
 ----------------------------
Node 45: Terminal node
 ----------------------------
Node 23: Intermediate node
 A case goes into Node 46 if hgc <= 9.5000000
```

```
 hgc mean = 8.9495192
 ----------------------------
Node 46: Terminal node
 ----------------------------
Node 47: Terminal node
 ----------------------------
Node 3: Terminal node
 ----------------------------
 Case and node IDs are in file: wage.nid
 Node fitted values are in file: wage.fit
 LaTeX code for tree is in wage.tex
 Split and fit variable names are stored in wage.var
```

Figure 33 shows the tree and Figure 34 plots lowess-smoothed curves of mean wage in the two terminal nodes. The figure is produced by the following R code.

```
z <- read.table("wagedat.txt",header=FALSE)
names(z) <- c("id","hgc","exper1","exper2","exper3","exper4","exper5","exper6",
              "exper7","exper8","exper9","exper10","exper11","exper12","exper13",
              "postexp1","postexp2","postexp3","postexp4","postexp5","postexp6",
              "postexp7","postexp8","postexp9","postexp10","postexp11","postexp12",
              "postexp13","wage1","wage2","wage3","wage4","wage5","wage6","wage7",
              "wage8","wage9","wage10","wage11","wage12","wage13","ged1","ged2",
              "ged3","ged4","ged5","ged6","ged7","ged8","ged9","ged10","ged11",
              "ged12","ged13","uerate1","uerate2","uerate3","uerate4","uerate5",
              "uerate6","uerate7","uerate8","uerate9","uerate10","uerate11",
              "uerate12","uerate13","race")
exper <- c(z$exper1,z$exper2,z$exper3,z$exper4,z$exper5,z$exper6,z$exper7,
           z$exper8,z$exper9,z$exper10,z$exper11,z$exper12,z$exper13)
wage <- c(z$wage1,z$wage2,z$wage3,z$wage4,z$wage5,z$wage6,z$wage7,z$wage8,
          z$wage9,z$wage10,z$wage11,z$wage12,z$wage13)
xr <- range(exper,na.rm=TRUE)
yr <- range(wage,na.rm=TRUE)

guide.fit <- read.table("wage.fit",header=TRUE)
g.node <- guide.fit$node
g.start <- guide.fit$t.start
g.end <- guide.fit$t.end
n <- length(g.node)
m <- dim(guide.fit)[2]
npts <- m-3  # number of time points for plotting

xvals <- guide.fit[,2:3]
xvals <- as.numeric(unlist(xvals))
yvals <- guide.fit[,4:m]
```

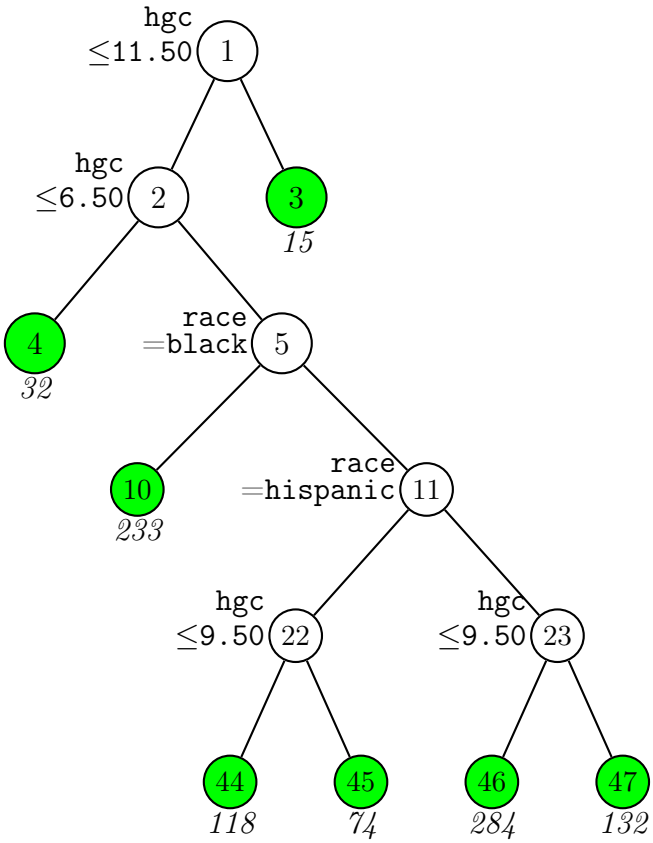

Figure 33: GUIDE v.46.1 0.038-SE (0.250-SE has no splits) regression tree for predicting longitudinal variables `wage1`, `wage2`, etc. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Second best split variable at root node is `race`.

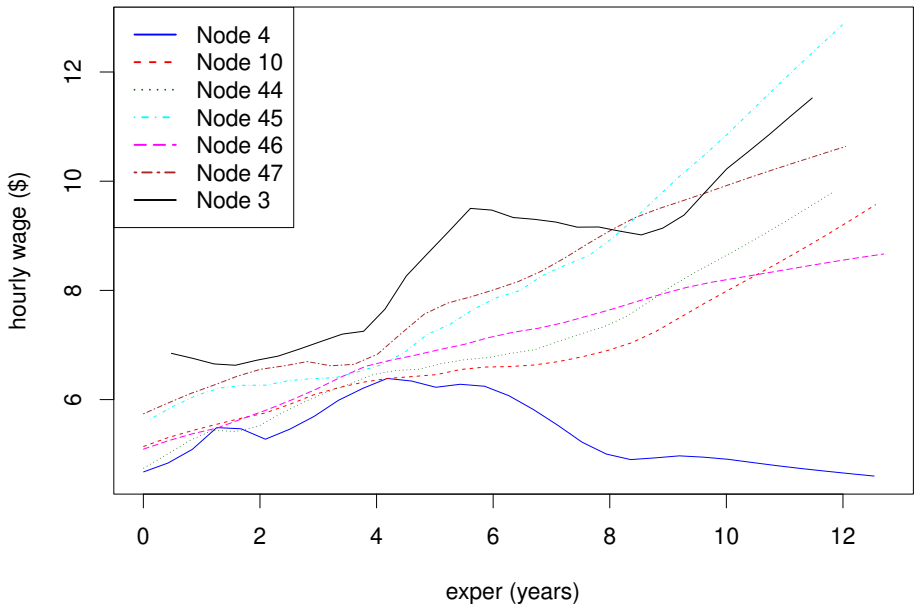

Figure 34: Lowess-smoothed mean wage curves in the terminal nodes of Figure 33.

```
yvals <- as.numeric(unlist(yvals))
plot(range(xvals),range(yvals),type="n",xlab="exper (years)",ylab="hourly wage ($)")
leg.col <- c("blue","red","darkgreen","cyan","magenta","brown","black")
leg.lty <- 1:n
for(i in 1:n){
    node <- g.node[i]
    start <- g.start[i]
    end <- g.end[i]
    gap <- (end-start)/(npts-1)
    x <- start+(0:(npts-1))*gap
    y <- as.numeric(guide.fit[i,4:m])
    lines(x,y,col=leg.col[i],lty=leg.lty[i])
}
leg.txt <- paste("Node",g.node)
legend("topleft",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2)
```

The plotting values are obtained from the result file `wage.fit` whose contents are given below. The first column gives the node number and the next two columns the start and end of the times at which fitted values are computed. The other columns give the fitted values equally spaced between the start and end times.

```
 node   t.start      t.end        fitted1      fitted2      fitted3      fitted4      fitted5      f
```

```
 4   0.50000E-02  0.12535E+02  0.46759E+01  0.48382E+01  0.50885E+01  0.54866E+01  0.54655E+01  0
10   0.40000E-02  0.12558E+02  0.51416E+01  0.53033E+01  0.54330E+01  0.55485E+01  0.56582E+01  0
44   0.60000E-02  0.11831E+02  0.47453E+01  0.49937E+01  0.52497E+01  0.54392E+01  0.54149E+01  0
45   0.12200E+00  0.11990E+02  0.56433E+01  0.58784E+01  0.60783E+01  0.62114E+01  0.62619E+01  0
46   0.10000E-02  0.12700E+02  0.50925E+01  0.52457E+01  0.53753E+01  0.54807E+01  0.56485E+01  0
47   0.20000E-02  0.12045E+02  0.57397E+01  0.59284E+01  0.61090E+01  0.62643E+01  0.64272E+01  0
 3   0.48000E+00  0.11473E+02  0.68471E+01  0.67557E+01  0.66532E+01  0.66289E+01  0.67211E+01  0
```

The contents of the file `wage.var` are given below. The 1st column gives the node number. The 2nd column is a letter, with `t` indicating that the node is terminal and `c`, `s`, or `n` indicating an intermediate node split on a `c`, `n` or `s` variable. The 3rd column gives the name of the variable used to split the node; the name `NONE` is used if a terminal node cannot be split by any variable. The 4th column gives the name of the interacting variable if there is one; otherwise the name of the split variable is repeated. If the node is terminal, the 5th column contains the letter "`t`"; otherwise if it is non-terminal, the 5th column is an integer indicating the number of split values to follow (a split on a `c` variable may have more than one value). In the example below, node 1 is split on `s` variable `hgc` at value 9.50. Nodes 2 and 3 are terminal nodes; each would be split on `race` if they were not terminal.

```
1 s hgc hgc        1   0.9500000000E+01
2 t race race   t
3 t race race   t
```

15 Logistic regression

If the dependent variable Y takes values 0 and 1, GUIDE can construct a tree model such that a simple or multiple linear logistic regression model is fitted in each node. The tree model may be more efficient (in terms of size and prediction accuracy) if a preliminary estimate of $p = P(Y = 1)$ is available. The preliminary estimate of p is not necessary, but it may be easily obtained by fitting a GUIDE forest or kernel discriminant model to the data. If a variable containing the estimated p values are included in the data, it should be specified as an "`e`" variable in the description file (see Section 3.1). Missing values in the predictor variables used in the logistic regression node models are imputed with node means; see Loh (2021) for more details.

We use the NHTSA data to demonstrate this, with $Y =$ `HIC2`, which takes value 1 if `HIC` > 999 and 0 otherwise. The DSC file is `nhtsadsc2.txt`. The "`e`" variable is `estHIC2` which is a column of estimated values of $p = P(Y = 1)$ obtained from GUIDE forest.

15.1 Piecewise constant

15.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: logitc.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: logitc.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 7
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: nhtsadsc2.txt
Reading DSC file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
Warning: B variables changed to C
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
```

```
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        2        0        0       48
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable:   8.4554334554334559E-002
Finished reading data file
Default max. number of split levels: 15
Minimum number of D=0 and D=1 in each node:           9
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): logitc.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: logitc.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: logitc.r
Input rank of top variable to split root node ([1:67], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < logitc.in
```

15.1.2 Contents of `logitc.out`

```
Binary logistic regression tree
Pruning by cross-validation
DSC file: nhtsadsc2.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
```

```
Warning: B variables changed to C
D variable is HIC2
Piecewise constant model
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables

Summary information for training sample of size 3276 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
e=estimated success probability
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                3
      2  BARSHP     c                               21
      3  BARANG     p    0.000        330.0          360       14
      4  BARDIA     s    191.0        1000.                   2807
      5  OCCWT      s    72.00        83.00                   3265
      6  OCCWT_     m                                  2
      :
    108  CARANG     p    0.000        99.00          360      991
    109  VEHOR      p    0.000        90.00          360      995
    110  RSTFRT     c                                  3
    111  HIC2       d    0.000        1.000
    112  estHIC2    e    0.000       0.8455

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        2        0        0       48
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable: 0.084554

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
```

```
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 32
Minimum number of D=0 and D=1 in each node: 9
Ranks of variables and their 1-df chi-squared values at root node
     1  0.1218E+04   COLMEC
     2  0.9001E+03   YEAR
     3  0.8714E+03   MODELD
     :
    65  0.2537E+00   CB
    66  0.1349E+00   KB
    67  0.4871E-01   HB

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1+       11   4.562E-01   2.044E-02   8.368E-03   4.472E-01   8.232E-03
   2        10   4.545E-01   2.020E-02   7.273E-03   4.484E-01   8.265E-03
   3**       9   4.529E-01   2.009E-02   6.781E-03   4.484E-01   5.719E-03
   4         8   4.590E-01   2.024E-02   6.010E-03   4.516E-01   5.748E-03
   5         6   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   6         5   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   7         4   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   8         3   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   9         2   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
  10         1   5.795E-01   2.316E-02   2.216E-03   5.834E-01   3.465E-03

0-SE tree based on mean is marked with * and has 9 terminal nodes
0-SE tree based on median is marked with + and has 11 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree
* tree same as ** tree
* tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.
```

```
D-mean is mean of HIC2 in the node
Cases fit give the number of cases used to fit node
Node deviance is residual deviance divided by residual degrees of freedom
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank    D-mean    deviance  variable       variables
          1    3276     3276     1  8.455E-02  5.797E-01  COLMEC
          2    2610     2610     1  2.797E-02  2.553E-01  MODELD
          4     436      436     1  1.468E-01  8.361E-01  IMPANG
          8T     56       56     1  3.214E-01  1.279E+00  -
          9     380      380     1  1.211E-01  7.400E-01  RSTFRT
         18T     54       54     1  5.185E-01  1.411E+00  -
         19     326      326     1  5.521E-02  4.285E-01  MODELD
         38T     46       46     1  1.957E-01  1.011E+00  -
         39T    280      280     1  3.214E-02  2.852E-01  -
          5T   2174     2174     1  4.140E-03  5.372E-02  -
          3     666      666     1  3.063E-01  1.234E+00  MODELD
          6     213      213     1  7.089E-01  1.212E+00  MODELD
         12T    113      113     1  5.310E-01  1.395E+00  CLSSPD
         13T    100      100     1  9.100E-01  6.112E-01  -
          7     453      453     1  1.170E-01  7.234E-01  MODELD
         14T    116      116     1  3.276E-01  1.276E+00  VEHSPD
         15T    337      337     1  4.451E-02  3.651E-01  -

Number of terminal nodes of final tree: 9
Total number of nodes of final tree: 17
Second best split variable (based on curvature test) at root node is YEAR

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: COLMEC = "BWU", "CYL", "NA", "NAP", "UNK"
  Node 2: MODELD = "4RUNNER", "9-5", "ACHIEVA", "AEROSTAR", "ASTRO",
            "BEAUVILLE", "BEETLE", "CLUBWAGON MPV", "DAKOTA", "DE VILLE",
            "DEFORMABLE IMPACTOR", "DURANGO", "ELANTRA", "EUROVAN",
            "FIVE HUNDRED", "FORESTER", "FRONTIER", "GALANT", "GRAND AM",
            "INSIGHT", "ION", "L200", "MAXIMA", "MAZDA3", "METRO", "MPV",
            "MUSTANG", "NEON", "PASEO", "PATHFINDER", "PT CRUISER", "QUEST",
            "RAM", "RAM WAGON VAN", "RODEO", "S-10", "S10 BLAZER", "S80",
            "SIDEKICK", "SILVERADO2500", "SPACECAB", "SPORTVAN", "TACOMA",
            "TEMPO", "TERCEL", "TRACKER", "TRAILBLAZER", "TROOPER II",
            "VERONA", "VUE"
    Node 4: IMPANG in [269, 287]
      Node 8: HIC2 proportion of 1s = 0.32142857
    Node 4: IMPANG not in [269, 287] or NA
      Node 9: RSTFRT = "0"
        Node 18: HIC2 proportion of 1s = 0.51851852
```

```
      Node 9: RSTFRT /= "0"
        Node 19: MODELD = "BEAUVILLE", "DE VILLE", "PATHFINDER", "PT CRUISER",
                 "RAM WAGON VAN", "S-10", "TEMPO", "TRAILBLAZER", "VERONA"
          Node 38: HIC2 proportion of 1s = 0.19565217
        Node 19: MODELD /= "BEAUVILLE", "DE VILLE", "PATHFINDER", "PT CRUISER",
                 "RAM WAGON VAN", "S-10", "TEMPO", "TRAILBLAZER", "VERONA"
          Node 39: HIC2 proportion of 1s = 0.32142857E-1
  Node 2: MODELD /= "4RUNNER", "9-5", "ACHIEVA", "AEROSTAR", "ASTRO",
           "BEAUVILLE", "BEETLE", "CLUBWAGON MPV", "DAKOTA", "DE VILLE",
           "DEFORMABLE IMPACTOR", "DURANGO", "ELANTRA", "EUROVAN",
           "FIVE HUNDRED", "FORESTER", "FRONTIER", "GALANT", "GRAND AM",
           "INSIGHT", "ION", "L200", "MAXIMA", "MAZDA3", "METRO", "MPV",
           "MUSTANG", "NEON", "PASEO", "PATHFINDER", "PT CRUISER", "QUEST",
           "RAM", "RAM WAGON VAN", "RODEO", "S-10", "S10 BLAZER", "S80",
           "SIDEKICK", "SILVERADO2500", "SPACECAB", "SPORTVAN", "TACOMA",
           "TEMPO", "TERCEL", "TRACKER", "TRAILBLAZER", "TROOPER II",
           "VERONA", "VUE"
    Node 5: HIC2 proportion of 1s = 0.41398344E-2
Node 1: COLMEC /= "BWU", "CYL", "NA", "NAP", "UNK"
  Node 3: MODELD = "18", "200", "200 SX", "2000", "210", "240", "244",
           "300 ZX", "318", "325 I", "4000", "4RUNNER", "5000", "504", "505",
           "604", "626", "ALLIANCE", "ASTRO", "AXXESS", "B2000 PICKUP",
           "BLAZER", "CAPRI", "CAPRICE", "CAR ELECTRIC", "CHAMP",
           "CLUBWAGON MPV", "COLT PICKUP", "COLT VISTA", "CONQUEST",
           "COURIER", "DAKOTA", "E100 VAN", "EL CAMINO", "ELECTRA",
           "ELECTREK", "ELECTRICA", "EXCEL GLS", "EXP", "F150 PICKUP",
           "FAIRMONT", "FESTIVA", "FUEGO", "FURY", "GF", "GL", "GLC", "GV",
           "I-MARK", "IMPULSE", "KING CAB PICKUP", "LANCER", "LE BARON",
           "LE CAR", "LEGACY", "LEOPARD", "LIBERTY", "LUMINA", "LUV",
           "MALIBU", "MARQUIS", "MEDALLION", "MERKUR", "METRO", "MONTERO",
           "MONZA", "MPV", "NEW YORKER", "NOVA", "PASSAT", "PICKUP",
           "PREVIA", "PRIZM", "PULSAR", "QUANTUM", "S-10", "SABLE",
           "SCIROCCO", "SHADOW", "SIDEKICK", "SOMERSET", "SONATA",
           "SPECTRUM", "SPIRIT", "SPORTSWAGON", "SPRINT", "ST. REGIS",
           "STANZA", "STARLET", "SUBURBAN", "T1000", "TEMPO", "TORINO",
           "TRANSIT CONNECT", "TREDIA", "TREKKER", "TROOPER II", "VANAGON",
           "WAGON", "WRANGLER"
    Node 6: MODELD = "18", "200", "244", "300 ZX", "626", "ALLIANCE", "BLAZER",
             "CAPRI", "CAPRICE", "COLT VISTA", "CONQUEST", "COURIER",
             "DAKOTA", "ELECTREK", "EXCEL GLS", "EXP", "FAIRMONT", "GLC",
             "IMPULSE", "LE BARON", "LEGACY", "LIBERTY", "LUV", "MEDALLION",
             "METRO", "MPV", "NOVA", "PASSAT", "PICKUP", "PREVIA", "S-10",
             "SABLE", "SCIROCCO", "SHADOW", "SONATA", "SPIRIT", "TEMPO",
             "WAGON", "WRANGLER"
      Node 12: HIC2 proportion of 1s = 0.53097345
    Node 6: MODELD /= "18", "200", "244", "300 ZX", "626", "ALLIANCE", "BLAZER",
```

```
            "CAPRI", "CAPRICE", "COLT VISTA", "CONQUEST", "COURIER",
            "DAKOTA", "ELECTREK", "EXCEL GLS", "EXP", "FAIRMONT", "GLC",
            "IMPULSE", "LE BARON", "LEGACY", "LIBERTY", "LUV", "MEDALLION",
            "METRO", "MPV", "NOVA", "PASSAT", "PICKUP", "PREVIA", "S-10",
            "SABLE", "SCIROCCO", "SHADOW", "SONATA", "SPIRIT", "TEMPO",
            "WAGON", "WRANGLER"
      Node 13: HIC2 proportion of 1s = 0.91000000
  Node 3: MODELD /= "18", "200", "200 SX", "2000", "210", "240", "244",
          "300 ZX", "318", "325 I", "4000", "4RUNNER", "5000", "504", "505",
          "604", "626", "ALLIANCE", "ASTRO", "AXXESS", "B2000 PICKUP",
          "BLAZER", "CAPRI", "CAPRICE", "CAR ELECTRIC", "CHAMP",
          "CLUBWAGON MPV", "COLT PICKUP", "COLT VISTA", "CONQUEST",
          "COURIER", "DAKOTA", "E100 VAN", "EL CAMINO", "ELECTRA",
          "ELECTREK", "ELECTRICA", "EXCEL GLS", "EXP", "F150 PICKUP",
          "FAIRMONT", "FESTIVA", "FUEGO", "FURY", "GF", "GL", "GLC", "GV",
          "I-MARK", "IMPULSE", "KING CAB PICKUP", "LANCER", "LE BARON",
          "LE CAR", "LEGACY", "LEOPARD", "LIBERTY", "LUMINA", "LUV",
          "MALIBU", "MARQUIS", "MEDALLION", "MERKUR", "METRO", "MONTERO",
          "MONZA", "MPV", "NEW YORKER", "NOVA", "PASSAT", "PICKUP",
          "PREVIA", "PRIZM", "PULSAR", "QUANTUM", "S-10", "SABLE",
          "SCIROCCO", "SHADOW", "SIDEKICK", "SOMERSET", "SONATA",
          "SPECTRUM", "SPIRIT", "SPORTSWAGON", "SPRINT", "ST. REGIS",
          "STANZA", "STARLET", "SUBURBAN", "T1000", "TEMPO", "TORINO",
          "TRANSIT CONNECT", "TREDIA", "TREKKER", "TROOPER II", "VANAGON",
          "WAGON", "WRANGLER"
    Node 7: MODELD = "CELICA", "CHEVETTE", "COLT", "CONCORD", "CORDIA",
            "CRESSIDA", "CUTLASS", "DL", "E150 VAN", "EXPLORER", "FOX",
            "GRANADA", "JETTA", "MAXIMA", "MOTOR HOME", "PONY EXCEL",
            "PRELUDE", "RANGER", "SENTRA", "SPORTSMAN", "SPORTVAN",
            "TAURUS", "TERCEL", "THUNDERBIRD", "VAN"
      Node 14: HIC2 proportion of 1s = 0.32758621
    Node 7: MODELD /= "CELICA", "CHEVETTE", "COLT", "CONCORD", "CORDIA",
            "CRESSIDA", "CUTLASS", "DL", "E150 VAN", "EXPLORER", "FOX",
            "GRANADA", "JETTA", "MAXIMA", "MOTOR HOME", "PONY EXCEL",
            "PRELUDE", "RANGER", "SENTRA", "SPORTSMAN", "SPORTVAN",
            "TAURUS", "TERCEL", "THUNDERBIRD", "VAN"
      Node 15: HIC2 proportion of 1s = 0.44510386E-1

*************************************************************
Predictor means below are means of cases with no missing values.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.
```

```
 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if COLMEC = "BWU", "CYL", "NA", "NAP", "UNK"
 COLMEC mode = "UNK"
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -2.382        17.39        0.000
 Proportion of ones in variable HIC2 = 0.845543E-1
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if MODELD = "4RUNNER", "9-5", "ACHIEVA", "AEROSTAR", "ASTRO",
 "BEAUVILLE", "BEETLE", "CLUBWAGON MPV", "DAKOTA", "DE VILLE", "DEFORMABLE IMPACTOR", "DURA
 "ELANTRA", "EUROVAN", "FIVE HUNDRED", "FORESTER", "FRONTIER", "GALANT", "GRAND AM", "INSIG
 "MUSTANG", "NEON", "PASEO", "PATHFINDER", "PT CRUISER", "QUEST", "RAM", "RAM WAGON VAN", "
 "S10 BLAZER", "S80", "SIDEKICK", "SILVERADO2500", "SPACECAB", "SPORTVAN", "TACOMA", "TEMPO
 "TRACKER", "TRAILBLAZER", "TROOPER II", "VERONA", "VUE"
 MODELD mode = "ACCORD"
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if IMPANG in [269, 287]
 IMPANG mean = 35.607798
 ----------------------------
Node 8: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.7472        5.150       0.3616E-05
 Proportion of ones in variable HIC2 = 0.321429
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if RSTFRT = "0"
 RSTFRT mode = "1"
 ----------------------------
Node 18: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.7411E-01    7.626       0.3935E-09
 Proportion of ones in variable HIC2 = 0.518519
 ----------------------------
Node 19: Intermediate node
 A case goes into Node 38 if MODELD = "BEAUVILLE", "DE VILLE", "PATHFINDER",
 "PT CRUISER", "RAM WAGON VAN", "S-10", "TEMPO", "TRAILBLAZER", "VERONA"
```

```
 MODELD mode = "MAXIMA"
 ----------------------------
Node 38: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -1.414        3.345       0.1667E-02
 Proportion of ones in variable HIC2 = 0.195652
 ----------------------------
Node 39: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -3.405        3.049       0.2513E-02
 Proportion of ones in variable HIC2 = 0.321429E-1
 ----------------------------
Node 5: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -5.483        3.006       0.2675E-02
 Proportion of ones in variable HIC2 = 0.413983E-2
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if MODELD = "18", "200", "200 SX", "2000", "210", "240", "244",
 "300 ZX", "318", "325 I", "4000", "4RUNNER", "5000", "504", "505", "604", "626", "ALLIANCE
 "B2000 PICKUP", "BLAZER", "CAPRI", "CAPRICE", "CAR ELECTRIC", "CHAMP", "CLUBWAGON MPV",
 "COLT PICKUP", "COLT VISTA", "CONQUEST", "COURIER", "DAKOTA", "E100 VAN", "EL CAMINO", "EL
 "ELECTREK", "ELECTRICA", "EXCEL GLS", "EXP", "F150 PICKUP", "FAIRMONT", "FESTIVA", "FUEGO"
 "IMPULSE", "KING CAB PICKUP", "LANCER", "LE BARON", "LE CAR", "LEGACY", "LEOPARD", "LIBERT
 "MARQUIS", "MEDALLION", "MERKUR", "METRO", "MONTERO", "MONZA", "MPV", "NEW YORKER", "NOVA"
 "QUANTUM", "S-10", "SABLE", "SCIROCCO", "SHADOW", "SIDEKICK", "SOMERSET", "SONATA", "SPECT
 "SPORTSWAGON", "SPRINT", "ST. REGIS", "STANZA", "STARLET", "SUBURBAN", "T1000", "TEMPO", "
 "TRANSIT CONNECT", "TREDIA", "TREKKER", "TROOPER II", "VANAGON", "WAGON", "WRANGLER"
 MODELD mode = "CIVIC"
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if MODELD = "18", "200", "244", "300 ZX", "626", "ALLIANCE", "BLAZER",
 "CAPRI", "CAPRICE", "COLT VISTA", "CONQUEST", "COURIER", "DAKOTA", "ELECTREK", "EXCEL GLS"
 "FAIRMONT", "GLC", "IMPULSE", "LE BARON", "LEGACY", "LIBERTY", "LUV", "MEDALLION", "METRO"
 "SCIROCCO", "SHADOW", "SONATA", "SPIRIT", "TEMPO", "WAGON", "WRANGLER"
 MODELD mode = "PICKUP"
 ----------------------------
Node 12: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1241        11.31       0.5471E-15
 Proportion of ones in variable HIC2 = 0.530973
 ----------------------------
```

```
Node 13: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant      2.314        31.80        0.000
 Proportion of ones in variable HIC2 = 0.910000
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if MODELD = "CELICA", "CHEVETTE", "COLT", "CONCORD", "CORDIA",
 "CRESSIDA", "CUTLASS", "DL", "E150 VAN", "EXPLORER", "FOX", "GRANADA", "JETTA", "MAXIMA",
 "MOTOR HOME", "PONY EXCEL", "PRELUDE", "RANGER", "SENTRA", "SPORTSMAN", "SPORTVAN", "TAURU
 "THUNDERBIRD", "VAN"
 MODELD mode = "CIVIC"
 ----------------------------
Node 14: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.7191        7.518        0.000
 Proportion of ones in variable HIC2 = 0.327586
 ----------------------------
Node 15: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -3.067        3.962       0.9074E-04
 Proportion of ones in variable HIC2 = 0.445104E-1
 ----------------------------
 Observed and fitted values are stored in logitc.fit
 LaTeX code for tree is in logitc.tex
```

The logistic regression tree is shown in Figure 35.

15.2 Simple linear

We can also construct a logistic regression tree with a simple linear logistic regression model fitted to each node.

15.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: logits.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: logits.out
```

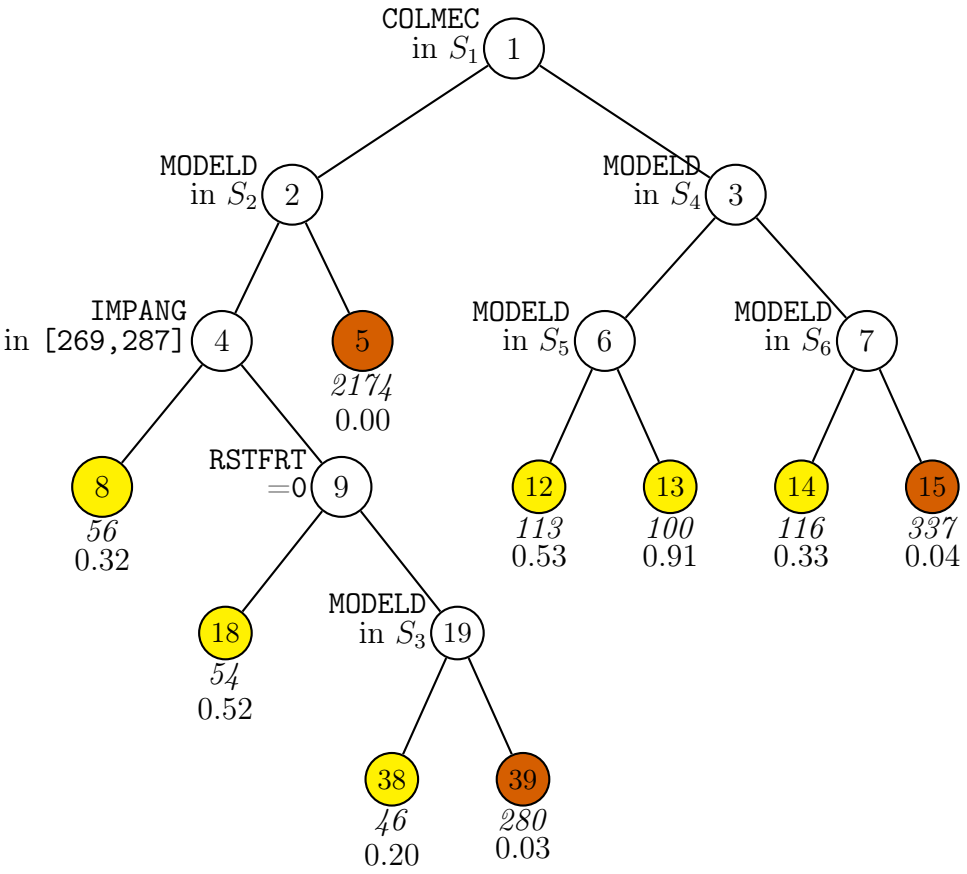

Figure 35: GUIDE v.46.1 0.250-SE piecewise-constant logistic regression tree for predicting P(HIC2=1). At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {BWU, CYL, NA, NAP, UNK}. S_2 = {4RUNNER, 9-5, ACHIEVA, AEROSTAR, ASTRO, BEAUVILLE, BEETLE, CLUBWAGON MPV, DAKOTA, DE VILLE, DEFORMABLE IMPACTOR, DURANGO, ELANTRA, EUROVAN, FIVE HUNDRED, FORESTER, FRONTIER, GALANT, GRAND AM, INSIGHT, ION, L200, MAXIMA, MAZDA3, METRO, MPV, MUSTANG, NEON, PASEO, PATHFINDER, PT CRUISER, QUEST, RAM, RAM WAGON VAN, RODEO, S-10, S10 BLAZER, S80, SIDEKICK, SILVERADO2500, SPACECAB, SPORTVAN, TACOMA, TEMPO, TERCEL, TRACKER, TRAILBLAZER, TROOPER II, VERONA, VUE}. S_3 = {BEAUVILLE, DE VILLE, PATHFINDER, PT CRUISER, RAM WAGON VAN, S-10, TEMPO, TRAILBLAZER, VERONA}. S_4 = {18, 200, 200 SX, 2000, 210, 240, 244, 300 ZX, 318, 325 I, 4000, 4RUNNER, 5000, 504, 505, 604, 626, ALLIANCE, ASTRO, AXXESS, B2000 PICKUP, BLAZER, CAPRI, CAPRICE, CAR ELECTRIC, CHAMP, CLUBWAGON MPV, COLT PICKUP, COLT VISTA, CONQUEST, COURIER, DAKOTA, E100 VAN, EL CAMINO, ELECTRA, ELECTREK, ELECTRICA, EXCEL GLS, EXP, F150 PICKUP, FAIRMONT, FESTIVA, FUEGO, FURY, GF, GL, GLC, GV, I-MARK, IMPULSE, KING CAB PICKUP, LANCER, LE BARON, LE CAR, LEGACY, LEOPARD, LIBERTY, LUMINA, LUV, MALIBU, MARQUIS, MEDALLION, MERKUR, METRO, MONTERO, MONZA, MPV, NEW YORKER, NOVA, PASSAT, PICKUP, PREVIA, PRIZM, PULSAR, QUANTUM, S-10, SABLE, SCIROCCO, SHADOW, SIDEKICK, SOMERSET, SONATA, SPECTRUM, SPIRIT, SPORTSWAGON, SPRINT, ST. REGIS, STANZA, STARLET, SUBURBAN, T1000, TEMPO, TORINO, TRANSIT CONNECT, TREDIA, TREKKER, TROOPER II, VANAGON, WAGON, WRANGLER}. S_5 = {18, 200, 244, 300 ZX, 626, ALLIANCE, BLAZER, CAPRI, CAPRICE, COLT VISTA, CONQUEST, COURIER, DAKOTA, ELECTREK, EXCEL GLS, EXP, FAIRMONT, GLC, IMPULSE, LE BARON, LEGACY, LIBERTY, LUV, MEDALLION, METRO, MPV, NOVA, PASSAT, PICKUP, PREVIA, S-10, SABLE, SCIROCCO, SHADOW, SONATA, SPIRIT, TEMPO, WAGON, WRANGLER}. S_6 = {CELICA, CHEVETTE, COLT, CONCORD, CORDIA, CRESSIDA, CUTLASS, DL, E150 VAN, EXPLORER, FOX, GRANADA, JETTA, MAXIMA, MOTOR HOME, PONY EXCEL, PRELUDE, RANGER, SENTRA, SPORTSMAN, SPORTVAN, TAURUS, TERCEL, THUNDERBIRD, VAN}. Sample size (in *italics*) and proportion of 1s in HIC2 printed below nodes. Terminal nodes with proportions of 1s above and below value of 0.08 at root node are painted yellow and vermillion respectively. Second best split variable at root node is YEAR.

```
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 7
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: nhtsadsc2.txt
Reading DSC file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        2       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
```

```
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable:   8.4554334554334559E-002
Finished reading data file
Constant fitted to cases with missing values in regressor variables
Default max. number of split levels: 15
Minimum number of D=0 and D=1 in each node:           9
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): logits.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=2):
Input file name: logits.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: logits.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: logits.r
Input rank of top variable to split root node ([1:67], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < logits.in
```

15.2.2 Contents of `logits.out`

```
Binary logistic regression tree
Pruning by cross-validation
DSC file: nhtsadsc2.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC2
Piecewise simple linear logistic model
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
```

```
Summary information for training sample of size 3276 (excluding observations
 with non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
e=estimated success probability
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                 3
      2  BARSHP     c                                21
      3  BARANG     p    0.000        330.0          360       14
      4  BARDIA     n   1.9100E+02    1000.                   2807
      :
    109  VEHOR      p    0.000        90.00          360      995
    110  RSTFRT     c                                 3
    111  HIC2       d    0.000        1.000
    112  estHIC2    e    0.000       0.8455

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        2       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable: 0.084554

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 33
Minimum number of D=0 and D=1 in each node: 9
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       15   7.144E-01   4.578E-02   7.517E-02   6.485E-01   1.081E-01
   2       14   7.121E-01   4.579E-02   7.570E-02   6.485E-01   1.116E-01
```

```
   3       13   7.121E-01   4.574E-02   7.593E-02   6.437E-01   1.127E-01
   4       10   6.540E-01   2.923E-02   6.352E-02   6.137E-01   7.424E-02
   5        8   6.712E-01   2.906E-02   6.403E-02   6.092E-01   9.406E-02
   6        7   6.597E-01   2.885E-02   6.506E-02   5.613E-01   9.848E-02
   7        6   6.721E-01   2.097E-02   6.801E-02   5.613E-01   1.205E-01
   8        5   6.735E-01   2.102E-02   6.844E-02   5.613E-01   1.225E-01
   9        3   7.398E-01   1.931E-02   8.496E-02   6.962E-01   1.514E-01
  10        2   4.547E-01   1.932E-02   9.157E-03   4.653E-01   1.100E-02
  11**      1   4.547E-01   1.932E-02   9.157E-03   4.653E-01   1.100E-02

 0-SE tree based on mean is marked with * and has 1 terminal node
 0-SE tree based on median is marked with + and has 1 terminal node
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree, ** tree, + tree, and ++ tree all the same

 Following tree is based on mean CV with naive SE estimate (**)

 Structure of final tree. Each terminal node is marked with a T.

 D-mean is mean of HIC2 in the node
 Cases fit give the number of cases used to fit node
 Node deviance is residual deviance divided by residual degrees of freedom
        Node    Total    Cases Matrix    Node       Node    Split          Other
        label   cases      fit  rank    D-mean    deviance  variable       variables
           1T    3276     3276     2  8.455E-02  4.546E-01 RSTFRT   -YEAR
 Best split at root node is on RSTFRT

 Number of terminal nodes of final tree: 1
 Total number of nodes of final tree: 1

 Regression tree:

 Node 1: HIC2 proportion of 1s = 0.84554335E-1

 ************************************************************
 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

Node 1: Terminal node
 Coefficients of logit function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      258.0        17.26       0.3833E-15
 YEAR        -0.1306       -17.38       0.9021E-15    1972.        2000.        2017.
 If regressor has missing values, predicted value = 0.84554335E-1
```

```
----------------------------
Observed and fitted values are stored in logits.fit
Regressor names and coefficients are stored in logits.reg
LaTeX code for tree is in logits.tex
```

The results show that the tree has no splits. It fits a simple linear logistic regression model to the whole data set with `YEAR` as linear predictor. If the value of `YEAR` is missing, the predicted value of p is the mean of `HIC2`.

16 Importance scoring

When there are numerous predictor variables, it may be useful to rank them in order of their "importance". GUIDE has a facility to do this. In addition, it provides thresholds for grouping the variables by their importance—see Loh and Zhou (2021).

16.1 Classification: RHC data

We show here how to obtain the importance scores for predicting `swang1`, the variable that takes values `RHC` and `NoRHC`.

16.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: imp.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
conversion ([1:3], <cr>=1): 2
Name of batch output file: imp.out
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc1.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
```

```
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):

Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):

Default max. number of split levels: 4
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=0):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
You can create a DSC file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: imp.scr
Input file is created!
Run GUIDE with the command: guide < imp.in
```

16.1.2 Contents of `imp.out`

The most interesting part of the output file is at the end, as shown below.

```
Scaled importance scores of predictor variables
     Score     Rank  Variable
 2.007E+01     1.00  cat1
 1.927E+01     2.00  aps1
 1.575E+01     3.00  crea1
 1.542E+01     4.00  pafi1
 1.409E+01     5.00  meanbp1
 9.920E+00     6.00  neuro
 9.256E+00     7.00  alb1
 9.002E+00     8.00  cat2
 8.441E+00     9.00  card
 8.033E+00    10.00  hema1
 7.422E+00    11.00  wtkilo1
 6.709E+00    12.00  seps
 6.503E+00    13.00  adld3p
 5.492E+00    14.00  dnr1
 5.408E+00    15.00  resp
 5.120E+00    16.00  bili1
 4.429E+00    17.00  paco21
 4.166E+00    18.00  surv2md1
 3.520E+00    19.00  transhx
 3.435E+00    20.00  resp1
 3.359E+00    21.00  chrpulhx
 3.094E+00    22.00  hrt1
 2.981E+00    23.00  ninsclas
 2.837E+00    24.00  dementhx
 2.711E+00    25.00  ph1
 1.977E+00    26.00  psychhx
 1.933E+00    27.00  gastr
 1.913E+00    28.00  das2d3pc
 1.825E+00    29.00  renal
 1.661E+00    30.00  income
 1.597E+00    31.00  cardiohx
 1.287E+00    32.00  urin1
 1.230E+00    33.00  trauma
 1.108E+00    34.00  age
 1.103E+00    35.00  sex
 1.092E+00    36.00  edu
 1.048E+00    37.00  sod1
 9.518E-01    38.00  immunhx
 9.517E-01    39.00  wblc1
 9.030E-01    40.00  ca
 8.900E-01    41.00  malighx
 7.916E-01    42.00  scoma1
 7.817E-01    43.00  amihx
 7.000E-01    44.00  chfhx
```

```
 6.124E-01    45.00  gibledhx
 4.960E-01    46.00  ortho
 4.872E-01    47.00  renalhx
 4.450E-01    48.00  pot1
 4.402E-01    49.00  liverhx
 4.146E-01    50.00  hema
 3.704E-01    51.00  temp1
 3.674E-01    52.00  meta
 2.300E-01    53.00  race
99% threshold is 1.2588
95% threshold is 1.0000
90% threshold is 0.8965
80% threshold is 0.7408
50% threshold is 0.5542
Number of variables above 99% threshold is 32
Number of variables between 95% and 99% thresholds is 5
Number of variables between 90% and 95% thresholds is 3
Number of variables between 80% and 90% thresholds is 3
Number of variables between 50% and 80% thresholds is 2
Number of variables below 50% threshold is 8
Importance scores are stored in imp.scr
```

The variables, sorted according to their importance scores, are divided into 5 groups:

A. Scores above 99% threshold

B. Scores above 95% threshold and below 99% threshold

C. Scores above 90% threshold and below 95% threshold

D. Scores above 80% threshold and below 90% threshold

E. Scores above 50% threshold and below 80% threshold

F. Scores below 50% threshold

The groups and thresholds have the following interpretation. Let H_0 denote the null hypothesis H_0 that the dependent variable is independent of the predictor variables (it is not assumed that the predictor variables are independent of each other). If H_0 is true, there is a 0.01, 0.05, 0.10, 0.20, and 0.50 probability that one or more predictor variables falls into groups A, $A \cup B$, $A \cup B \cup C$, $A \cup B \cup C \cup D$, and $A \cup B \cup C \cup D \cup E$, respectively. The importance scores are normalized so that the 95% threshold is 1.0.

The file `imp.scr` lists the rank, group membership, importance score, number of missing values, and variable name.

Rank	Group	Score	Missing	Variable
1	A	2.007E+01	0	cat1
2	A	1.927E+01	0	aps1
3	A	1.575E+01	0	crea1
4	A	1.542E+01	0	pafi1
5	A	1.409E+01	80	meanbp1
6	A	9.920E+00	0	neuro
7	A	9.256E+00	0	alb1
8	A	9.002E+00	4535	cat2
9	A	8.441E+00	0	card
10	A	8.033E+00	0	hema1
11	A	7.422E+00	515	wtkilo1
12	A	6.709E+00	0	seps
13	A	6.503E+00	4296	adld3p
14	A	5.492E+00	0	dnr1
15	A	5.408E+00	0	resp
16	A	5.120E+00	0	bili1
17	A	4.429E+00	0	paco21
18	A	4.166E+00	0	surv2md1
19	A	3.520E+00	0	transhx
20	A	3.435E+00	136	resp1
21	A	3.359E+00	0	chrpulhx
22	A	3.094E+00	159	hrt1
23	A	2.981E+00	0	ninsclas
24	A	2.837E+00	0	dementhx
25	A	2.711E+00	0	ph1
26	A	1.977E+00	0	psychhx
27	A	1.933E+00	0	gastr
28	A	1.913E+00	0	das2d3pc
29	A	1.825E+00	0	renal
30	A	1.661E+00	0	income
31	A	1.597E+00	0	cardiohx
32	A	1.287E+00	3028	urin1
33	B	1.230E+00	0	trauma
34	B	1.108E+00	0	age
35	B	1.103E+00	0	sex
36	B	1.092E+00	0	edu
37	B	1.048E+00	0	sod1
38	C	9.518E-01	0	immunhx
39	C	9.517E-01	0	wblc1
40	C	9.030E-01	0	ca
41	D	8.900E-01	0	malighx
42	D	7.916E-01	0	scoma1
43	D	7.817E-01	0	amihx

```
44  E     7.000E-01          0     chfhx
45  E     6.124E-01          0     gibledhx
46  F     4.960E-01          0     ortho
47  F     4.872E-01          0     renalhx
48  F     4.450E-01          0     pot1
49  F     4.402E-01          0     liverhx
50  F     4.146E-01          0     hema
51  F     3.704E-01          0     temp1
52  F     3.674E-01          0     meta
53  F     2.300E-01          0     race
```

Figure 36 shows a barplot of the scores, produced by the following R code.

```
par(las=1,mar=c(5,12,4,2),cex.axis=0.8)
leg.col <- c("red","orange","yellow","green","blue","white")
leg.txt <- c("A (99%)","B (95%)","C (90%)","D (80%)","E (50%)","F (< 50%)")
x <- read.table("imp.scr",header=TRUE)
n <- nrow(x)
score <- x$Score
vars <- x$Variable
group <- x$Group
barcol <- rep("white",n)
letrs <- c("A","B","C","D","E","F")
for(i in 1:5){
    barcol[group == letrs[i]] <- leg.col[i]
}
barplot(rev(score[1:n]),names.arg=rev(vars[1:n]),col=rev(barcol[1:n]),horiz=TRUE,
        xlab="GUIDE importance scores")
legend("bottomright",legend=leg.txt,fill=leg.col)
```

16.2 Censored response with R variable

Following is the corresponding scoring procedure for a censored response with a treatment (`R`) variable (`swang1`). The `R` variable is not given a score because it acts as a linear predictor in the nodes of the tree.

16.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: imp_surv.in
```

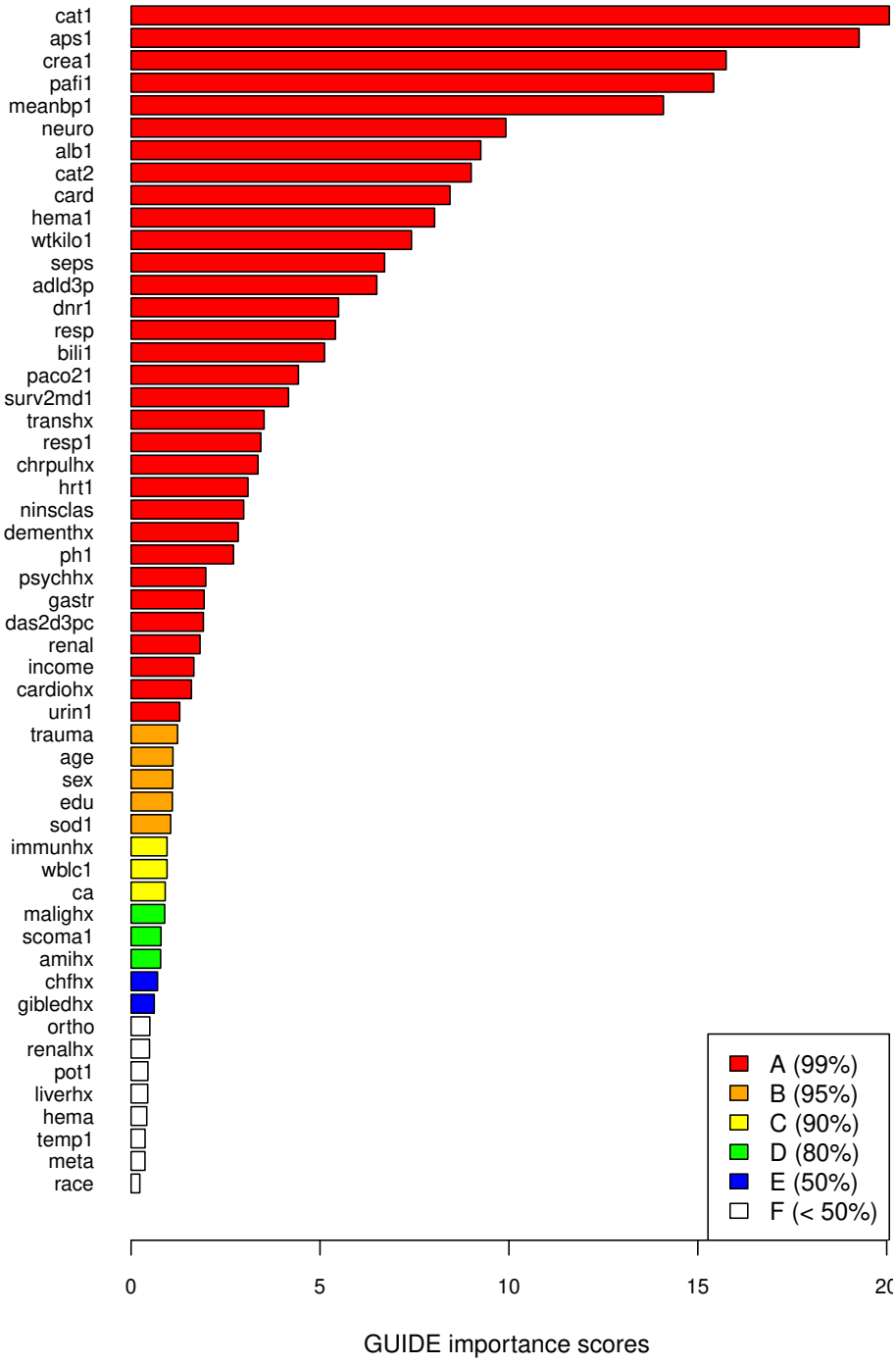

Figure 36: Scores of important variables for predicting `swang1`

```
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1): 2
Name of batch output file: imp_surv.out
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc3.txt
Reading DSC file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Input 1 if randomized trial, 2 if observational study: ([1:2], <cr>=1): 2
```

```
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        1       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Number of dummy variables created: 1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of cases excluded due to 0 W or missing D, T or R variables: 0
Finished reading data file
Default max. number of split levels: 8
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=0):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
You can create a DSC file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: imp_surv.scr
Input file is created!
Run GUIDE with the command: guide < imp_surv.in
```

16.2.2 Partial contents of imp_surv.out

```
Scaled importance scores of predictor variables
 (F, I and R variables are excluded)
     Score     Rank  Variable
```

```
9.968E-01     1.00  dnr1
8.981E-01     2.00  ph1
7.948E-01     3.00  chrpulhx
7.360E-01     4.00  paco21
7.110E-01     5.00  resp1
4.373E-01     6.00  liverhx
4.306E-01     7.00  pot1
4.148E-01     8.00  gastr
3.964E-01     9.00  cat2
3.826E-01    10.00  age
3.521E-01    11.00  gibledhx
3.428E-01    12.00  pafi1
3.195E-01    13.00  aps1
3.129E-01    14.00  malighx
2.958E-01    15.00  hrt1
2.856E-01    16.00  amihx
2.838E-01    17.00  surv2md1
2.803E-01    18.00  ninsclas
2.576E-01    19.00  income
2.491E-01    20.00  meanbp1
2.469E-01    21.00  edu
2.287E-01    22.00  das2d3pc
2.044E-01    23.00  ortho
2.019E-01    24.00  scoma1
1.790E-01    25.00  sex
1.755E-01    26.00  alb1
1.664E-01    27.00  meta
1.646E-01    28.00  hema1
1.584E-01    29.00  temp1
1.555E-01    30.00  crea1
1.551E-01    31.00  ca
1.528E-01    32.00  renalhx
1.522E-01    33.00  trauma
1.449E-01    34.00  hema
1.401E-01    35.00  psychhx
1.392E-01    36.00  wtkilo1
1.390E-01    37.00  urin1
1.366E-01    38.00  neuro
1.337E-01    39.00  dementhx
1.231E-01    40.00  chfhx
1.204E-01    41.00  wblc1
1.066E-01    42.00  sod1
9.480E-02    43.00  race
9.229E-02    44.00  adld3p
8.731E-02    45.00  seps
8.553E-02    46.00  cardiohx
```

```
 8.401E-02    47.00  cat1
 6.692E-02    48.00  resp
 5.832E-02    49.00  card
 5.806E-02    50.00  immunhx
 5.768E-02    51.00  renal
 5.684E-02    52.00  bili1
 5.596E-02    53.00  transhx
99% threshold is 1.3762
95% threshold is 1.0000
90% threshold is 0.9475
80% threshold is 0.7654
50% threshold is 0.7235
Number of variables above 99% threshold is 0
Number of variables between 95% and 99% thresholds is 0
Number of variables between 90% and 95% thresholds is 1
Number of variables between 80% and 90% thresholds is 2
Number of variables between 50% and 80% thresholds is 1
Number of variables below 50% threshold (excluding F, I and R variables) is 51
```

17 Propensity scores

17.1 Causal inference

Propensity scores are often used in causal inference to estimate average treatment effects. Given a treatment variable Z taking values 0 (no treatment) and 1 (treatment), the propensity score for a subject with covariate $X = x$ is $\pi(x) = P(Z = 1 \,|\, X = x)$. If n denotes the sample size and Y_i the response of the ith subject, the average treatment effect may be estimated by the *Horvitz-Thompson estimate (HT)*

$$n^{-1} \sum_{i=1}^{n} \left\{ \frac{Z_i Y_i}{\hat{\pi}(X_i)} - \frac{(1 - Z_i) Y_i}{1 - \hat{\pi}(X_i)} \right\}$$

or the *Hájek inverse probability estimate (IPW)*

$$\frac{\sum_i Z_i Y_i / \hat{\pi}(X_i)}{\sum_i Z_i / \hat{\pi}(X_i)} - \frac{\sum_i (1 - Z_i) Y_i / (1 - \hat{\pi}(X_i))}{\sum_i (1 - Z_i) / (1 - \hat{\pi}(X_i))}$$

where $\hat{\pi}(x)$ is an estimate of $\pi(x)$. Clearly, $\hat{\pi}(x)$ cannot be 0 or 1.

Propensity scores are traditionally estimated by logistic regression, but this approach has difficulties if there are missing values in the covariates or if the number of covariates is large. Random forest has been used, but the version implemented in R is not applicable to data with missing values in predictor variables. Even when there are no missing values, the propensity score estimates from logistic regression and random forest are not easy to interpret.

A logistic regression tree or a piecewise-constant regression tree for estimating $\pi(x)$ is more interpretable than a forest. To prevent any $\hat{\pi}$ from being 0 or 1, the "propensity score" option in GUIDE fits a piecewise-constant regression tree to the Z_i such that no terminal node has all $Z_i = 0$ or all $Z_i = 1$. If this option is used to estimate the propensity scores, the HT and IPW estimates are identical and reduce to the *sample size weighted estimate* $n^{-1} \sum_t n_t \hat{\beta}_t$, where the sum is over the terminal nodes and n_t and $\hat{\beta}_t$ are the sample size and estimated treatment effect in node t.

We demonstrate the propensity score feature with the RHC data. Doctors believe that direct measurement of cardiac function by right heart catheterization for some critically ill patients yields better outcomes. The benefit of RHC has not been demonstrated in a randomized clinical trial due to ethical concerns. In observational studies, the relative risk of death was found to be higher in the elderly and in patients with acute myocardial infarction who received RHC. In such studies, the decision to use RHC is at the discretion of the physician. Therefore treatment assignment is

confounded with patient factors that are also related to outcomes, e.g., patients with low blood pressure are more likely to get RHC, and such patients are also more likely to die. The data consist of observations on more than 60 variables for 5735 patients from 5 medical centers over 5 years (Connors et al., 1996). The treatment variable is `swang1` (`RHC` or `NoRHC`) which we define as 1 if `RHC` and 0 if `NoRHC`. The resulting data are in the file `propendata.csv` and the DSC file is `propen.dsc` where `swang` is specified as `d` and `death` and `dth30` are specified as `x`.

The next section shows how the input file for propensity score trees is created. After the propensity score option is chosen, the program will ask for a value for the parameter $\delta = \min(p, 1-p)$, where p denotes a propensity score in any node of the tree, i.e., $\delta < p < 1-\delta$. The default value $\delta = 0.05$ ensures that no propensity score is less than 0.05 or greater than 0.95.

17.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: propen.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: propen.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 3
Input min(p,1-p), where p is propensity score ([0.01:0.49], <cr>=0.05):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: propen.dsc
Reading DSC file ...
Training sample file: propendata.csv
Missing value code: NA
Records in data file start on line 2
35 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
```

```
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
No weight variable in data file
Number of cases used for training: 5735
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Default max. number of split levels: 12
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): propen.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: propen.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: propen.r
Input rank of top variable to split root node ([1:53], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < propen.in
```

17.1.2 Contents of `propen.out`

```
Propensity score tree with min(p,1-p) = 0.050 where p is an estimated propensity score
Least squares regression tree
Pruning by cross-validation
DSC file: propen.dsc
Training sample file: propendata.csv
```

```
Missing value code: NA
Records in data file start on line 2
35 N variables changed to S
D variable is swang1
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1       c                                 9
      3  cat2       c                                 6       4535
      4  ca         c                                 3
     10  cardiohx   s    0.000        1.000
     11  chfhx      s    0.000        1.000
      :
     44  ph1        s    6.579        7.770
     45  swang1     d    0.000        1.000
     46  wtkilo1    s    19.50        244.0                     515
     47  dnr1       c                                 2
     48  ninsclas   c                                 6
     49  resp       c                                 2
     50  card       c                                 2
     51  neuro      c                                 2
     52  gastr      c                                 2
     53  renal      c                                 2
     54  meta       c                                 2
     55  hema       c                                 2
     56  seps       c                                 2
     57  trauma     c                                 2
     58  ortho      c                                 2
     59  adld3p     s    0.000        7.000                  4296
     60  urin1      s    0.000        9000.                  3028
     61  race       c                                 3
     62  income     c                                 4

Numbers of N, F, S, and P below exclude constant variables
```

```
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
No weight variable in data file
Number of cases used for training: 5735
Number of cases excluded due to 0 W or missing D variable: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 12
Minimum node sample size: 5
Ranks of variables and their 1-df chi-squared values at root node
     1  0.3346E+03   cat1
     2  0.2728E+03   aps1
     3  0.2430E+03   crea1
     4  0.2402E+03   meanbp1
     :
    43  0.1052E+01   meta
    44  0.6357E+00   race

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       39   2.044E-01   2.778E-03   1.535E-03   2.059E-01   1.812E-03
   2       37   2.044E-01   2.778E-03   1.535E-03   2.059E-01   1.812E-03
   3       36   2.044E-01   2.778E-03   1.535E-03   2.059E-01   1.821E-03
   4*      33   2.041E-01   2.773E-03   1.554E-03   2.048E-01   1.699E-03
   5       32   2.042E-01   2.772E-03   1.557E-03   2.050E-01   1.726E-03
   6--     31   2.042E-01   2.765E-03   1.589E-03   2.047E-01   2.013E-03
   7       30   2.046E-01   2.740E-03   1.493E-03   2.049E-01   1.762E-03
   :
  17       18   2.047E-01   2.679E-03   1.405E-03   2.066E-01   2.138E-03
  18**     16   2.047E-01   2.647E-03   1.224E-03   2.049E-01   1.498E-03
  19++     15   2.048E-01   2.603E-03   9.774E-04   2.050E-01   1.448E-03
  20       14   2.050E-01   2.593E-03   1.010E-03   2.055E-01   1.741E-03
  21       12   2.052E-01   2.586E-03   9.845E-04   2.056E-01   1.461E-03
  22       11   2.053E-01   2.564E-03   8.040E-04   2.056E-01   1.172E-03
  23       10   2.057E-01   2.518E-03   9.355E-04   2.059E-01   1.089E-03
  24        9   2.062E-01   2.497E-03   1.152E-03   2.060E-01   8.934E-04
  25        8   2.062E-01   2.497E-03   1.152E-03   2.060E-01   8.934E-04
```

```
  26        7   2.070E-01   2.498E-03   1.448E-03   2.060E-01   9.126E-04
  27        6   2.093E-01   2.468E-03   1.610E-03   2.092E-01   1.371E-03
  28        5   2.100E-01   2.403E-03   1.650E-03   2.106E-01   1.783E-03
  29        4   2.121E-01   2.384E-03   1.268E-03   2.123E-01   1.435E-03
  30        3   2.192E-01   2.130E-03   1.388E-03   2.194E-01   1.228E-03
  31        2   2.284E-01   1.885E-03   1.252E-03   2.281E-01   1.729E-03
  32        1   2.358E-01   1.528E-03   6.632E-05   2.357E-01   1.152E-04

0-SE tree based on mean is marked with * and has 33 terminal nodes
0-SE tree based on median is marked with + and has 31 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of swang1 in the node
Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases Matrix    Node      Node          Split
       label   cases      fit  rank    D-mean     MSE           variable
          1    5735     5735     1  3.808E-01  2.358E-01   cat1
          2    1683     1683     1  5.401E-01  2.485E-01   meanbp1
          4    1117     1117     1  6.204E-01  2.357E-01   pafi1
          8     655      655     1  6.962E-01  2.118E-01   resp1
        16T     197      197     1  8.731E-01  1.114E-01   -
        17T     458      458     1  6.201E-01  2.361E-01   adld3p
          9     462      462     1  5.130E-01  2.504E-01   ninsclas
        18T     218      218     1  3.853E-01  2.379E-01   chfhx
        19T     244      244     1  6.270E-01  2.348E-01   -
         5T     566      566     1  3.816E-01  2.364E-01   alb1
          3    4052     4052     1  3.147E-01  2.157E-01   pafi1
          6    1292     1292     1  4.837E-01  2.499E-01   paco21
         12    1042     1042     1  5.278E-01  2.495E-01   aps1
         24     463      463     1  4.255E-01  2.450E-01   resp1
        48T     152      152     1  5.987E-01  2.419E-01   hema1
        49T     311      311     1  3.408E-01  2.254E-01   aps1
         25     579      579     1  6.097E-01  2.384E-01   resp1
        50T     111      111     1  8.198E-01  1.491E-01   -
        51T     468      468     1  5.598E-01  2.469E-01   -
        13T     250      250     1  3.000E-01  2.108E-01   pafi1
          7    2760     2760     1  2.355E-01  1.801E-01   aps1
         14    2100     2100     1  1.838E-01  1.501E-01   cat1
         28    1326     1326     1  2.353E-01  1.801E-01   wtkilo1
```

```
         56T     390      390     1  1.333E-01  1.159E-01   -
         57T     936      936     1  2.778E-01  2.008E-01   alb1
         29T     774      774     1  9.561E-02  8.658E-02   cat1
         15      660      660     1  4.000E-01  2.404E-01   crea1
         30T     129      129     1  2.171E-01  1.713E-01   -
         31      531      531     1  4.444E-01  2.474E-01   wtkilo1
         62T      85       85     1  1.765E-01  1.471E-01   -
         63T     446      446     1  4.955E-01  2.505E-01   adld3p

Number of terminal nodes of final tree: 16
Total number of nodes of final tree: 31
Second best split variable (based on curvature test) at root node is aps1

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: meanbp1 <= 68.500000 or NA
    Node 4: pafi1 <= 266.15625
      Node 8: resp1 <= 17.000000 or NA
        Node 16: swang1-mean = 0.87309645
      Node 8: resp1 > 17.000000
        Node 17: swang1-mean = 0.62008734
    Node 4: pafi1 > 266.15625 or NA
      Node 9: ninsclas = "Medicaid", "Medicare", "Medicare & Medicaid"
        Node 18: swang1-mean = 0.38532110
      Node 9: ninsclas /= "Medicaid", "Medicare", "Medicare & Medicaid"
        Node 19: swang1-mean = 0.62704918
  Node 2: meanbp1 > 68.500000
    Node 5: swang1-mean = 0.38162544
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: pafi1 <= 142.35938
    Node 6: paco21 <= 47.046875
      Node 12: aps1 <= 56.500000
        Node 24: resp1 <= 17.000000
          Node 48: swang1-mean = 0.59868421
        Node 24: resp1 > 17.000000 or NA
          Node 49: swang1-mean = 0.34083601
      Node 12: aps1 > 56.500000 or NA
        Node 25: resp1 <= 25.500000
          Node 50: swang1-mean = 0.81981982
        Node 25: resp1 > 25.500000 or NA
          Node 51: swang1-mean = 0.55982906
    Node 6: paco21 > 47.046875 or NA
      Node 13: swang1-mean = 0.30000000
  Node 3: pafi1 > 142.35938 or NA
```

```
    Node 7: aps1 <= 62.500000
      Node 14: cat1 = "ARF", "Colon Cancer", "MOSF w/Malignancy"
        Node 28: wtkilo1 <= 56.949995 or NA
          Node 56: swang1-mean = 0.13333333
        Node 28: wtkilo1 > 56.949995
          Node 57: swang1-mean = 0.27777778
      Node 14: cat1 /= "ARF", "Colon Cancer", "MOSF w/Malignancy"
        Node 29: swang1-mean = 0.95607235E-1
    Node 7: aps1 > 62.500000 or NA
      Node 15: crea1 <= 1.3498535
        Node 30: swang1-mean = 0.21705426
      Node 15: crea1 > 1.3498535 or NA
        Node 31: wtkilo1 <= 57.399980
          Node 62: swang1-mean = 0.17647059
        Node 31: wtkilo1 > 57.399980 or NA
          Node 63: swang1-mean = 0.49551570

 ************************************************************
 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3808        59.39       0.3027-313
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if meanbp1 <= 68.500000 or NA
 meanbp1 mean = 72.674985
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if pafi1 <= 266.15625
 pafi1 mean = 241.37331
 ----------------------------
```

```
Node 8: Intermediate node
 A case goes into Node 16 if resp1 <= 17.000000 or NA
 resp1 mean = 28.430124
 ----------------------------
Node 16: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.8731        36.72       0.3027-313
 swang1 mean = 0.873096
 ----------------------------
Node 17: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.6201        27.31       0.3027-313
 swang1 mean = 0.620087
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if ninsclas = "Medicaid", "Medicare", "Medicare & Medicaid"
 ninsclas mode = "Private"
 ----------------------------
Node 18: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3853        11.66       0.3027-313
 swang1 mean = 0.385321
 ----------------------------
Node 19: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.6270        20.21       0.3027-313
 swang1 mean = 0.627049
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3816        18.67       0.3027-313
 swang1 mean = 0.381625
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if pafi1 <= 142.35938
 pafi1 mean = 211.08630
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if paco21 <= 47.046875
 paco21 mean = 40.529679
 ----------------------------
```

```
Node 12: Intermediate node
 A case goes into Node 24 if aps1 <= 56.500000
 aps1 mean = 60.471209
 ----------------------------
Node 24: Intermediate node
 A case goes into Node 48 if resp1 <= 17.000000
 resp1 mean = 26.771058
 ----------------------------
Node 48: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.5987        15.01       0.3027-313
 swang1 mean = 0.598684
 ----------------------------
Node 49: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3408        12.66       0.3027-313
 swang1 mean = 0.340836
 ----------------------------
Node 25: Intermediate node
 A case goes into Node 50 if resp1 <= 25.500000
 resp1 mean = 33.916667
 ----------------------------
Node 50: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.8198        22.37       0.3027-313
 swang1 mean = 0.819820
 ----------------------------
Node 51: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.5598        24.37       0.3027-313
 swang1 mean = 0.559829
 ----------------------------
Node 13: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3000        10.33       0.3027-313
 swang1 mean = 0.300000
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if aps1 <= 62.500000
 aps1 mean = 50.276087
 ----------------------------
```

```
Node 14: Intermediate node
 A case goes into Node 28 if cat1 = "ARF", "Colon Cancer", "MOSF w/Malignancy"
 cat1 mode = "ARF"
 ----------------------------
Node 28: Intermediate node
 A case goes into Node 56 if wtkilo1 <= 56.949995 or NA
 wtkilo1 mean = 72.028899
 ----------------------------
Node 56: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1333        7.736       0.3027-313
 swang1 mean = 0.133333
 ----------------------------
Node 57: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.2778        18.96       0.3027-313
 swang1 mean = 0.277778
 ----------------------------
Node 29: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.9561E-01    9.040       0.3027-313
 swang1 mean = 0.956072E-1
 ----------------------------
Node 15: Intermediate node
 A case goes into Node 30 if crea1 <= 1.3498535
 crea1 mean = 3.0464749
 ----------------------------
Node 30: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.2171        5.957       0.3027-313
 swang1 mean = 0.217054
 ----------------------------
Node 31: Intermediate node
 A case goes into Node 62 if wtkilo1 <= 57.399980
 wtkilo1 mean = 75.515042
 ----------------------------
Node 62: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1765        4.243       0.3027-313
 swang1 mean = 0.176471
 ----------------------------
```

```
Node 63: Terminal node
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.4955        20.91       0.3027-313
 swang1 mean = 0.495516
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.1805

 Observed and fitted values are stored in propen.fit
 LaTeX code for tree is in propen.tex
 R code is stored in propen.r
```

The propensity score tree is shown in Figure 37. The two numbers below each terminal node are the sample size (in italics) and the estimate of $P(\texttt{swang1} = \texttt{RHC})$.

The file `propen.fit` gives the estimated propensity scores in the last column. Below are the top 7 rows of the file:

train	node	observed	predicted
y	48	0.00000	0.598684
y	17	1.00000	0.620087
y	63	1.00000	0.495516
y	56	0.00000	0.133333
y	18	1.00000	0.385321
y	29	0.00000	0.956072E-001
y	56	0.00000	0.133333

The following R code may be used to compute the Horvitz-Thompson and inverse probability weighted (IPW) estimates of average treatment effect for probability of `death`.

```
data <- read.csv("propendata.csv",header=TRUE)
y <- 1-data$dth30
fit <- read.table("propen.fit",header=TRUE)
train <- fit$train == "y"
z <- fit$observed[train]
p <- fit$predicted[train]
node <- fit$node[train]
nodenum <- unique(sort(node))
n <- sum(train)
horvitz <- (sum(z*y/p)-sum((1-z)*y/(1-p)))/n
ipw <- sum(z*y/p)/sum(z/p) - sum((1-z)*y/(1-p))/sum((1-z)/(1-p))
```

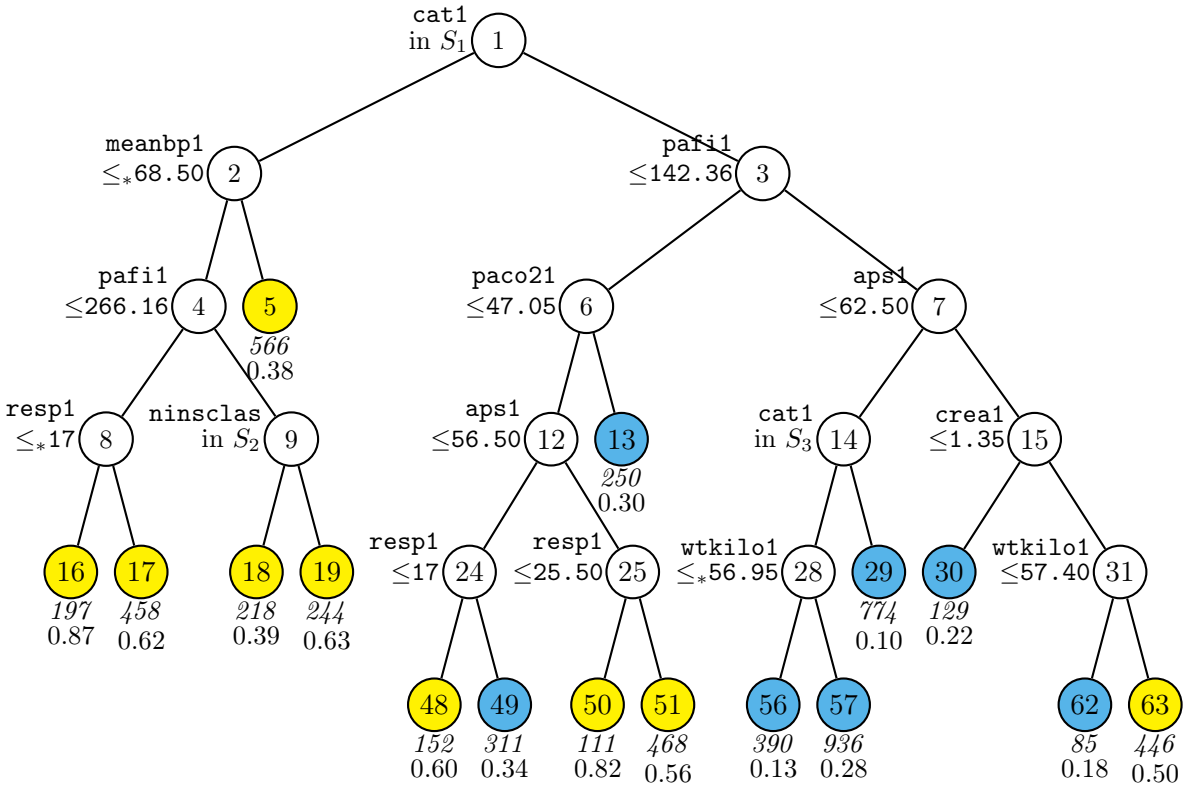

Figure 37: GUIDE v.46.1 0.250-SE tree for estimating propensity scores of `swang1`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`CHF`, `MOSF w/Sepsis`}. S_2 = {`Medicaid`, `Medicare`, `Medicare & Medicaid`}. S_3 = {`ARF`, `Colon Cancer`, `MOSF w/Malignancy`}. Sample size (in *italics*) and estimated propensity score printed below nodes. Terminal nodes with propensity scores above and below value of 0.381 at root node are painted yellow and skyblue respectively. Second best split variable at root node is `aps1`.

17.2 Missing-value imputation

Section 6.1.3 showed how to use a regression tree to impute missing values in a response variable to estimate the population mean of INTRDVX in the BLS data. Another common imputation method is hot-deck via propensity scores. In this method, propensity scores are used to partition the sample space into "imputation cells" and missing values in a cell are imputed with random draws from the observed responses in the cell.

To construct a propensity score tree and the imputation cells, we first replace the values of INTRDVX in ce2021.txt by a nonmissing indicator variable (INTRDVXnonmiss, say) that takes value 1 if INTRDVX is nonmissing and 0 otherwise, and save the new file as ce2021miss.txt, as shown by the following R code.

```
z <- read.table("ce2021.txt",header=TRUE)
y <- z$INTRDVX
INTRDVX <- rep(1,nrow(z))
INTRDVX[is.na(y)] <- 0
z$INTRDVX <- INTRDVX
names(z)[which(names(z) == "INTRDVX")] <- "INTRDVXnonmiss"
write.table(z,"ce2021miss.txt",row.names=FALSE,col.names=TRUE)
```

Key parts of the corresponding DSC file, which we call ce2021miss.dsc, are shown below. The file is the same as ce2021reg.dsc except for (i) a change in the first line from "ce2021.txt" to "ce2021miss.txt", and (ii) a change in the line for the dependent variable from "INTRDVX" to "INTRDVXnonmiss".

```
ce2021miss.txt
NA
2
1 DIRACC n
2 DIRACC_ m
3 AGE_REF n
4 AGE_REF_ m
5 AGE2 n
6 AGE2_ m
:
404 FSMPFRMX n
405 FSMP_RMX m
406 INTRDVXnonmiss d
407 INTRDVX_ x
408 IRAB n
409 IRAB_ m
:
```

```
547 WHLFYR c
548 WHLFYR_ m
549 FFTAXOWE n
550 FSTAXOWE n
```

17.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: propen.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: propen.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score tree
 Input your choice ([1:3], <cr>=1): 3
 Input min(p,1-p), where p is propensity score ([0.01:0.49], <cr>=0.05):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: ce2021miss.dsc
 Reading DSC file ...
 Training sample file: ce2021miss.txt
 Missing value code: NA
 Records in data file start on line 2
 Number of M variables associated with C variables: 19
 384 N variables changed to S
 D variable is INTRDVXnonmiss
 Reading data file ...
 Number of records in data file: 3965
 Length of longest entry in data file: 11
 Checking for missing values ...
 Finished checking
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
 Finding number of levels of M variables associated with C variables ...
 Assigning integer codes to values of 47 categorical variables
 Finished assigning codes to 10 categorical variables
 Finished assigning codes to 20 categorical variables
 Finished assigning codes to 30 categorical variables
 Finished assigning codes to 40 categorical variables
 Associating missing values of N and S variables with M variable codes ...
 Re-checking data ...
 Allocating missing value information ...
```

```
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 3965
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Default max. number of split levels: 15
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): propen.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: propen.fit
You can output an R function for predicting future observations
Input 1 to skip this step
Input 2 to output an R function that applies to the training data
Input 3 to output an R function that applies to a new data set
Input your choice ([1:3], <cr>=2):
Input filename to write R function: propen.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < propen.in
```

17.2.2 Output file

```
Propensity score tree with min(p,1-p) = 0.050 where p is an estimated propensity score
Least squares regression tree
Pruning by cross-validation
DSC file: ce2021miss.dsc
Training sample file: ce2021miss.txt
Missing value code: NA
Records in data file start on line 2
```

```
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVXnonmiss
Piecewise constant model
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 3965
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                    #Codes/
                                                    Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  DIRACC          s    1.000        1.000                  170  constant
      2  DIRACC_         m                                2
      3  AGE_REF         s    18.00        87.00
      4  AGE_REF_        m                                0
      5  AGE2            s    21.00        87.00                 1734
      :
     31  FINLWT21        w    1072.       0.9390E+05
      :
    406  INTRDVXnonmiss  d    0.000        1.000
      :
    547  WHLFYR          c                                1     3964
    548  WHLFYR_         m                                1
    549  FFTAXOWE        s  -0.3368E+05   0.3997E+06
    550  FSTAXOWE        s   -3309.       0.7223E+05

Numbers of N, F, S, and P below exclude constant variables
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      379
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of constant S variables: 5
Weight variable FINLWT21 in column: 31
Number of cases used for training: 3965
Number of cases excluded due to 0 W or missing D variable: 0
```

```
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Max number of interaction splits: 100
Maximum number of split levels: 15
Minimum node sample size: 3
Ranks of variables and their 1-df chi-squared values at root node
     1  0.2268E+03   STATE
     2  0.1587E+03   INCLASS2
     3  0.1569E+03   ERANKH
     :
   416  0.2087E-04   TOTEXPCQ
   417  0.2087E-04   TOTEX4CQ
   418  0.8284E-07   TAIRFARP

Warning: all zeroes in node 207; shrinking propensity score away from 0
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       63   4.881E+03   1.008E+02   5.767E+01   4.799E+03   1.154E+02
   2       62   4.880E+03   1.008E+02   5.734E+01   4.799E+03   1.146E+02
   :
  28       10   4.770E+03   9.675E+01   7.654E+01   4.675E+03   1.296E+02
  29--      9   4.740E+03   9.532E+01   6.872E+01   4.658E+03   1.190E+02
  30**      8   4.762E+03   9.353E+01   7.339E+01   4.687E+03   9.692E+01
  31        7   4.954E+03   8.175E+01   6.892E+01   4.903E+03   1.144E+02
  32        6   4.954E+03   8.175E+01   6.892E+01   4.903E+03   1.144E+02
  33        5   4.951E+03   8.137E+01   6.997E+01   4.903E+03   1.144E+02
  34        2   5.183E+03   7.727E+01   7.528E+01   5.182E+03   1.099E+02
  35        1   5.519E+03   6.287E+01   3.904E+01   5.515E+03   4.372E+01

0-SE tree based on mean is marked with * and has 9 terminal nodes
0-SE tree based on median is marked with + and has 9 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as ++ tree
* tree same as -- tree

Following tree is based on mean CV with naive SE estimate (**)

Structure of final tree. Each terminal node is marked with a T.
```

```
D-mean is weighted mean of INTRDVXnonmiss in the node
Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases Matrix    Node      Node          Split
       label   cases      fit  rank    D-mean     MSE           variable
          1     3965     3965     1  6.217E-01  5.518E+03   STATE
          2     1425     1425     1  7.722E-01  3.850E+03   INCLASS2
          4T      65       65     1  2.990E-01  4.739E+03   BEDROOMQ
          5T    1360     1360     1  7.952E-01  3.561E+03   DIVISION
          3     2540     2540     1  5.458E-01  6.038E+03   RETSURVX
          6     1838     1838     1  5.478E-01  6.186E+03   FINDRETX
         12     1164     1164     1  4.692E-01  6.128E+03   ERANKH
         24      635      635     1  5.696E-01  5.969E+03   STATE
         48T      85       85     1  1.702E-01  2.519E+03   MISCCQ
         49T     550      550     1  6.125E-01  6.025E+03   -
         25T     529      529     1  3.515E-01  5.690E+03   LIQUIDX
         13T     674      674     1  6.781E-01  5.597E+03   POPSIZE
          7      702      702     1  5.403E-01  5.659E+03   RETSURVX
         14T     129      129     1  1.048E-01  2.294E+03   STATE
         15T     573      573     1  6.465E-01  5.131E+03   PSU

Number of terminal nodes of final tree: 8
Total number of nodes of final tree: 15
Second best split variable (based on curvature test) at root node is INCLASS2

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: STATE = "2", "6", "10", "11", "21", "24", "25", "27", "31", "40", "41",
         "47", "48", "49"
  Node 2: INCLASS2 = NA
    Node 4: INTRDVXnonmiss-mean = 0.29897738
  Node 2: INCLASS2 /= NA
    Node 5: INTRDVXnonmiss-mean = 0.79522322
Node 1: STATE /= "2", "6", "10", "11", "21", "24", "25", "27", "31", "40", "41",
         "47", "48", "49"
  Node 3: RETSURVX = NA & RETS_RVX = "A"
    Node 6: FINDRETX <= 391.50000
      Node 12: ERANKH <= 0.65269515
        Node 24: STATE = "13", "15", "19", "22", "28", "32", "45"
          Node 48: INTRDVXnonmiss-mean = 0.17022443
        Node 24: STATE /= "13", "15", "19", "22", "28", "32", "45"
          Node 49: INTRDVXnonmiss-mean = 0.61251648
      Node 12: ERANKH > 0.65269515 or NA
        Node 25: INTRDVXnonmiss-mean = 0.35154294
```

```
    Node 6: FINDRETX > 391.50000 or NA
      Node 13: INTRDVXnonmiss-mean = 0.67809123
  Node 3: not (RETSURVX = NA & RETS_RVX = "A")
    Node 7: RETSURVX = NA
      Node 14: INTRDVXnonmiss-mean = 0.10477178
    Node 7: RETSURVX /= NA
      Node 15: INTRDVXnonmiss-mean = 0.64648526
```

Figure 38 shows the propensity score tree for INTRDVX being nonmissing. The terminal nodes of the tree can serve as imputation cells for hot-deck imputation of the missing values in INTRDVX to obtain the values of $\hat{y}_i$ in equation (1). The estimated mean is highly random, because hot-deck itself is an intrinsically random process. As a result, it may be necessary to repeat the hot-deck sampling several times, as shown in the following R code.

```
z <- read.table("ce2021.txt",header=TRUE)
w <- z$FINLWT21
y <- z$INTRDVX
prop <- read.table("propen.fit",header=TRUE)
node <- prop$node
ntimes <- 100
est.mean <- 0
totwt <- 0
for(k in 1:ntimes){
    for(i in 1:nrow(z)){
        if(is.na(y[i])){
            responses <- node == node[i] & !is.na(y)
            if(sum(responses) > 0){
                y[i] <- sample(y[responses],1)
            }
        }
    }
    gp <- !is.na(y)
    est.mean <- est.mean+sum(w[gp]*y[gp])/sum(w[gp])
}
est.mean <- est.mean/ntimes
print(est.mean)
```

An alternative method to estimate a population mean is by inverse propensity weighting

$$\Big(\sum_{i\in S_1} w_i/\hat{\pi}_i\Big)^{-1}\sum_{i\in S_1} w_i y_i/\hat{\pi}_i$$

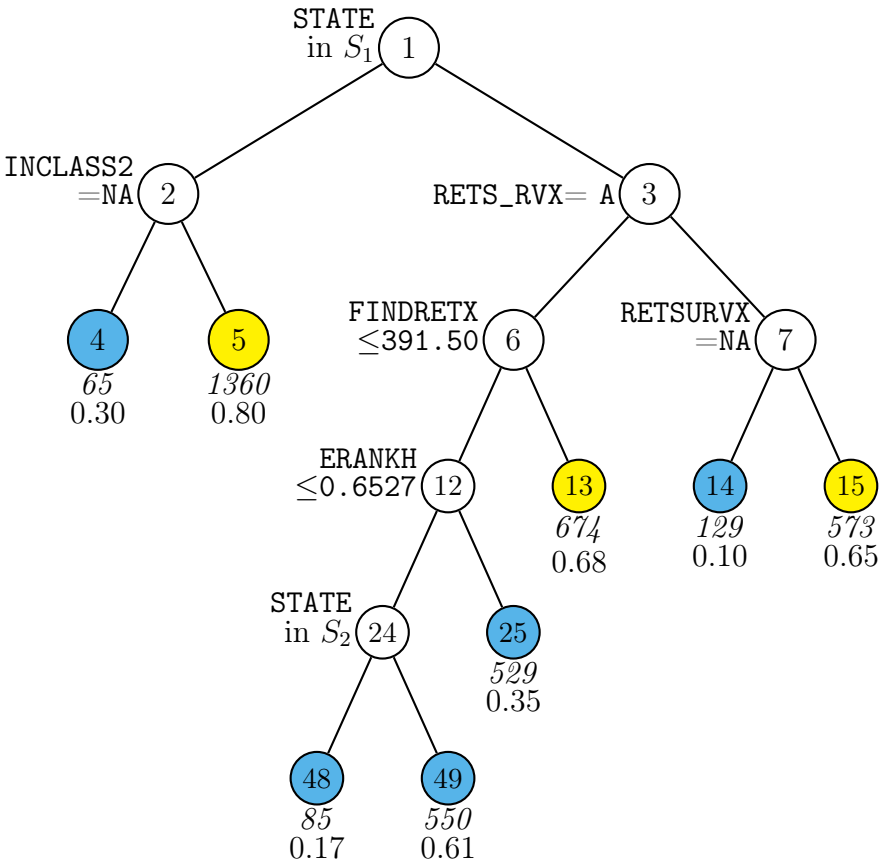

Figure 38: GUIDE v.46.1 0.250-SE tree for estimating propensity scores of `INTRDVXnonmiss`. At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {2, 6, 10, 11, 21, 24, 25, 27, 31, 40, 41, 47, 48, 49}. S_2 = {13, 15, 19, 22, 28, 32, 45}. Sample size (in *italics*) and estimated propensity score printed below nodes. Terminal nodes with propensity scores above and below value of 0.622 at root node are painted yellow and skyblue respectively. Second best split variable at root node is `INCLASS2`.

where S_1 is the set of observations where `INTRDVX` is nonmissing and $\hat{\pi}_i$ is its estimated propensity score. This method yields a nonrandomized estimate of 5246.455 as calculated by the following R code.

```
z <- read.table("ce2021.txt",header=TRUE)
w <- z$FINLWT21
y <- z$INTRDVX
prop <- read.table("propen.fit",header=TRUE)
score <- prop$predicted
gp <- !is.na(y)
ipw <- sum(w[gp]*y[gp]/score[gp])/sum(w[gp]/score[gp])
```

18 Differential item functioning: GDS data

GUIDE has an experimental option to identify important predictor variables and items with differential item functioning (DIF) in a data set with two or more item (dependent variable) scores. We illustrate it with a data set from Broekman et al. (2011, 2008) and Marc et al. (2008). It consists of responses from 1978 subjects on 15 items. There are 3 predictor variables (age, education, and gender). The data and DSC files are `GDS.dat` and `GDS.dsc`. Although the item responses in this example are 0-1, GUIDE allows them to be in any ordinal (e.g., Likert) scale. The contents of `GDS.dsc` are:

```
GDS.dat
NA
1
1 rid x
2 satis d
3 drop d
4 empty d
5 bored d
6 spirit d
7 afraid d
8 happy d
9 help d
10 home d
11 memory d
12 alive d
13 worth d
14 energy d
15 hope d
16 better d
17 total x
18 gender c
19 education n
20 age n
21 dxcurren x
22 sumscore x
```

Here is the session log to create an input file for identifying DIF items and the important predictor variables:

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: dif.in
 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
 conversion ([1:3], <cr>=1): 2
```

```
Name of batch output file: dif.out
Input 1 for classification, 2 for regression, 3 for propensity score tree
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: GDS.dsc
Reading DSC file ...
Training sample file: GDS.dat
Missing value code: NA
Records in data file start on line 1
2 N variables changed to S
Number of D variables: 15
D variables are:
satis
drop
empty
bored
spirit
afraid
happy
help
home
memory
alive
worth
energy
hope
better
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1): 2
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Reading data file ...
Number of records in data file: 1978
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Missing values found in D variables
```

```
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Some D variables have missing values
Rereading data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1978          0          0        4        0        0        2
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 1977
Number of split variables: 3
Number of cases excluded due to 0 W or missing D variable: 1
Finished reading data file
Input 1 to save p-value matrix for differential item functioning (DIF), 2
otherwise ([1:2], <cr>=1):
Input file name to store DIF p-values: dif.pv
Default max. number of split levels: 4
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=0):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
You can create a DSC file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: dif.scr
Input file is created!
Run GUIDE with the command: guide < dif.in
```

The importance scores are in the file `dif.scr`. They show that `age` is most important, followed by `gender` and `education`.

```
Rank         Score    Variable
1.00     8.89389E+00  age
2.00     5.05776E+00  gender
3.00     3.40934E+00  education
```

The word 'yes' in the last column of dif.pv below shows which item has DIF. In this example, only item #10 (memory) has DIF.

Item	Itemname	education	age	gender	DIF
1	satis	0.465E-01	0.377E-01	0.102E+00	no
2	drop	0.135E-01	0.218E+00	0.923E+00	no
3	empty	0.180E-02	0.122E+00	0.185E+00	no
4	bored	0.230E-05	0.218E+00	0.300E+00	no
5	spirit	0.973E+00	0.734E+00	0.362E-01	no
6	afraid	0.297E-01	0.411E-03	0.264E-02	no
7	happy	0.734E+00	0.353E+00	0.248E-01	no
8	help	0.397E-01	0.624E+00	0.435E-02	no
9	home	0.359E+00	0.114E+00	0.771E-03	no
10	memory	0.357E+00	0.000E+00	0.205E-01	yes
11	alive	0.172E+00	0.151E+00	0.444E+00	no
12	worth	0.405E+00	0.739E+00	0.698E+00	no
13	energy	0.661E+00	0.657E+00	0.117E-03	no
14	hope	0.607E+00	0.413E+00	0.216E+00	no
15	better	0.518E+00	0.644E+00	0.446E+00	no

19 Bootstrap confidence intervals

Owing to the numerous procedures that are performed during tree construction (such as selection of the variable and the split set to partition each intermediate node), proper statistical inference must account for the multiple testing and estimation issues. Otherwise, the error variance will be underestimated. Suppose, for example, we wish to obtain confidence intervals for the proportion of "RHC" in each terminal node of the tree in Figure **??**. Let n denote the sample size in a node and $\hat{p}$ the proportion of observations in it with the response value RHC. The usual $(1-\alpha)$ binomial interval is then $\hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}$, where z_α is the α -quantile of the standard normal distribution. This formula yields intervals that are too short because it does not account for the extra variance due to model construction. Bonferroni corrections, which are traditionally used for multiple testing, are inapplicable here because the number of tests are not specified in advance. For example, the number of chi-squared tests at each node depends on the number of variables eligible to split the node and the number of levels of splits depends on the total sample size, extent of pruning, and other parameters such as the minimum sample size in each node.

As with the Bonferroni correction, a natural solution is to change the multiplier $z_{1-\alpha/2}$ to a larger value. The bootstrap method provides one simple solution. Called "bootstrap calibration", the procedure is described and analyzed in Loh (1987, 1991) in the context of estimating a nonparametric mean; it is extended to subgroup analysis from regression tree models in Loh et al. (2016, 2019c) and Loh and Zhou (2020). The R code below implements the procedure. It can be used by following these steps:

1. Change the name of the data file (`rhcdata.txt` here) to `realdata.txt`.
2. Change the name of the DSC file (`rhcdsc1.txt` here) to `real.dsc`.
3. Change the name of the GUIDE input file (`classin.txt` here) to `real.in`.
4. Change the word "RHC" in line 1 of the R code to the name of the desired class in the data file.
5. In Windows, change the word "`system`" in lines 32, 32, 74 and 75 to "`shell`" if necessary.
6. Source the program in R.

```
class.name <- "RHC"  ## name of desired class in realdata.txt
nboot <- 1000
probs <- c(0.80,0.90,0.95,0.98)
zstat <- rep(0,nboot)
### write bootstrap DSC file boot.dsc
file <- readLines("real.dsc")  ## read real DSC file
write("bootdata.txt",file="boot.dsc")
len <- length(file)
write(file[2:length(file)],"boot.dsc",append=TRUE)
write(paste(len-2,"w w"),"boot.dsc",append=TRUE)
### write bootstrap input file boot.in
file <- readLines("real.in")  ## read real input file
file2 <- gsub("real.","boot.",file) ## replace "real." with "boot."
write(file2,"boot.in")
### read real data
z0 <- read.table("realdata.txt",header=TRUE)
nobs <- nrow(z0)
zt <- cbind(z0,rep(0,nobs)) ### add column of weight 0
write("Bootstrap simultaneous intervals by linear interpolation of z",
      "results.txt")
write("trials   z80     z90     z95     z98     bias.err    sd.err",
      "results.txt", append=TRUE)
err.test <- rep(0,nboot) ## misclassification rates
bias <- 0
for(i in 1:nboot){
    zb <- z0[sample(nobs,nobs,replace=TRUE),]
    zb <- cbind(zb,rep(1,nobs)) ### add column of weight 1
    write.table(zb,"bootdata.txt",col.names=TRUE,row.names=FALSE)
    write.table(zt,"bootdata.txt",col.names=FALSE,row.names=FALSE,
                append=TRUE)
    system("rm -f log.txt boot.out boot.fit")
    system("guide < boot.in > log.txt")
    bfit <- read.table("boot.fit",header=TRUE)  ## read boot results
    test <- bfit$train == "n"
    err.test[i] <- sum(bfit$observed[test] != bfit$predicted[test])/nobs
    err.resub <- sum(bfit$observed[!test] != bfit$predicted[!test])/nobs
    bias <- bias+(err.resub-err.test[i])
    unodes <- unique(sort(bfit$node))
    for(j in 1:length(unodes)){
        gp <- bfit$node == unodes[j] & bfit$train == "y" ## training data
        n0 <- sum(bfit$observed[gp] != class.name)
        n1 <- sum(bfit$observed[gp] == class.name)
        ntot <- n0+n1
        estp <- n1/ntot
        if(n1 == 0 | n0 == 0){
            p <- (n1+0.5)/(ntot+1)
```

```
47              sd <- sqrt(p*(1-p)/(ntot+1))
48          } else {
49              sd <- sqrt(estp*(1-estp)/ntot)
50          }
51          gp <- bfit$node == unodes[j] & bfit$train == "n"  ## real data
52          n0 <- sum(bfit$observed[gp] != class.name)
53          n1 <- sum(bfit$observed[gp] == class.name)
54          realp <- n1/(n0+n1)
55          zstat[i] <- max(zstat[i],abs(realp-estp)/sd)
56      }
57      if(i %% 100 == 0){
58          sd.err <- sqrt(var(err.test[1:i])) ## linear interpolation
59          q <- quantile(zstat[1:i],probs=probs,type=4)
60          write(c(i,q,bias/i,sd.err),"results.txt",append=TRUE,ncol=7)
61      }
62  }
63  ### find calibrated z.alpha
64  write(paste("No. bootstraps =",nboot),"results.txt",append=TRUE)
65  write(c("Calibrated z at levels",probs),file="results.txt",ncol=5,
66        append=TRUE)
67  q <- quantile(zstat,probs=probs,type=4) ## linear interpolation
68  write(q,"results.txt",append=TRUE,ncol=4)
69  write(paste("Bootstrap estimate of bias of error rate =",bias/nboot),
70        "results.txt",append=TRUE)
71  write(paste("Bootstrap estimate of SD of error rate =",
72        sqrt(var(err.test))),"results.txt",append=TRUE)
73  ### fit real data
74  system("rm -f log.txt real.out real.fit")
75  system("guide < real.in > log.txt")
76  realfit <- read.table("real.fit",header=TRUE)
77  train <- realfit$train == "y"
78  err.obs <- sum(realfit$observed[train] != realfit$predicted[train])/nobs
79  write(paste("Real data observed error rate =",err.obs),"results.txt",
80        append=TRUE)
81  k <- 3      ## 95% level
82  z0 <- q[k] ## 95% z value
83  write(c("Simultaneous intervals at level",probs[k]),
84        file="results.txt",ncol=2,append=TRUE)
85  write(paste0("Node  N   P(",class.name,") halfwid   left   right"),
86        "results.txt", append=TRUE)
87  unodes <- unique(sort(realfit$node))
88  for(j in 1:length(unodes)){
89      gp <- realfit$node == unodes[j] & realfit$train == "y"
90      n0 <- sum(realfit$observed[gp] != class.name)
91      n1 <- sum(realfit$observed[gp] == class.name)
92      ntot <- n0+n1
```

```
93      if(n1 == 0 | n0 == 0){
94          p <- (n1+0.5)/(ntot+1)
95          sd <- sqrt(p*(1-p)/(ntot+1))
96      } else {
97          p <- n1/ntot
98          sd <- sqrt(p*(1-p)/(ntot))
99      }
100     p <- n1/ntot
101     halfwid <- z0*sd
102     left <- p-halfwid
103     rght <- p+halfwid
104     write(c(unodes[j],ntot,p,halfwid,left,rght),"results.txt",
105           append=TRUE,ncol=6)
106 }
107 ## write(sort(zstat),"zstat.txt",ncol=1) ## output sorted zstat values
```

Figure 39 gives the contents of the file `results.txt`. It shows that the calibrated z-multiplier is 3.961722, 4.325215, 4.690964, or 5.337637 for 80%, 90%, 95%, or 98% simultaneous confidence intervals. For 95% intervals, the left and right end points of the intervals in each terminal node are given in the bottom half of the file.

20 Tree ensembles

A tree ensemble is a collection of trees. GUIDE has two methods of constructing an ensemble.

GUIDE forest. This the preferred method. Similar to Random Forest (Breiman, 2001), it fits *unpruned* trees to bootstrap samples and randomly selects a small subset of variables to search for splits at each node. There are, however, two important differences:

1. GUIDE forest uses the unbiased GUIDE method for split selection; Random Forest uses the biased CART method. One consequence is that GUIDE forest can be very much faster than Random Forest if the dependent variable is a class variable having more than two distinct values and some categorical predictor variables have many categories.
2. GUIDE forest is applicable to data with missing values. The R implementation of Random Forest (Liaw and Wiener, 2002) requires apriori imputation of missing values in the predictor variables.

The default number of trees for GUIDE forest is 1000 if there are fewer than 500 training samples and 100 predictor variables; otherwise, the default is 500.

```
Bootstrap simultaneous intervals by linear interpolation of z
trials  z80    z90    z95     z98      bias.err     sd.err
100 4.036962 4.458809 4.545827 4.922293 -0.03357803 0.005906056
200 4.123996 4.508203 4.777955 5.035208 -0.03335222 0.005670584
300 4.093978 4.513735 4.918732 5.117146 -0.0335048 0.00598086
400 4.108083 4.519645 4.835633 5.28808 -0.03360811 0.005930667
500 4.108083 4.508203 4.826329 5.117146 -0.03377507 0.005887693
600 4.144132 4.548011 4.895352 5.408027 -0.03397879 0.005812075
700 4.123996 4.529434 4.889087 5.408027 -0.03377357 0.005839512
800 4.117319 4.51814 4.845685 5.365021 -0.03369159 0.00588305
900 4.108552 4.50332 4.835633 5.408027 -0.03358888 0.005924705
1000 4.108083 4.495735 4.845685 5.397256 -0.03353304 0.005951228
No. bootstraps =  1000
Calibrated z at levels 0.8 0.9 0.95 0.98
4.108083 4.495735 4.845685 5.397256
Bootstrap estimate of bias of error rate =  -0.0335330427201395
Bootstrap estimate of SD of error rate =  0.00595122775778847
Real data observed error rate =  0.296251089799477
Simultaneous intervals at level 0.95
Node  N   P(RHC) halfwid   left   right
5 566 0.3816254 0.09894446 0.282681 0.4805699
7 2760 0.2355072 0.03913718 0.1963701 0.2746444
8 655 0.6961832 0.08707675 0.6091065 0.78326
18 244 0.6270492 0.1500158 0.4770334 0.7770649
19 218 0.3853211 0.1597212 0.2255999 0.5450423
25 66 0.3484848 0.2842088 0.06427609 0.6326936
26 110 0.6363636 0.2222518 0.4141119 0.8586154
27 601 0.3627288 0.09503228 0.2676965 0.4577611
48 438 0.6552511 0.1100458 0.5452053 0.7652969
49 77 0.3506494 0.2635033 0.08714608 0.6141526
```

Figure 39: Contents of `results.txt`

Bagged GUIDE. This fits *pruned* GUIDE trees to bootstrap samples of the training data (Breiman, 1996). Each tree is pruned by 5-fold cross-validation. The default number of trees is 200 if there are fewer than 500 training samples and 100 predictor variables; otherwise, the default is 100.

With the default settings, GUIDE forest is typically much faster than bagged GUIDE.

20.1 GUIDE forest: CE data

20.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: gf.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data
conversion ([1:3], <cr>=1):
Name of batch output file: gf.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1): 2
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2):
Input 1 for classification, 2 for least-squares regression
Input your choice ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of DSC file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021class.dsc
Reading DSC file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 48 categorical variables
```

```
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Warning: No linear splits; number of S variables must be < 225
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Warning: All positive weights treated as 1
Default max. number of split levels: 18
Input name of file to store predicted class and probability: gf.pro
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < gf.in
```

20.1.2 Contents of `gf.out`

Note: Owing to the intrinsic randomness in forests, your results may differ from those shown below. "OOB" stands for "out-of-bag".

```
Random forest of classification trees
No pruning
DSC file: ce2021class.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
383 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 3965
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 3
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358

Summary information for training sample of size 3965
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     s    1.000        1.000                   170
      2  DIRACC_    m                                 2
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       s    21.00        87.00                  1734
      6  AGE2_      m                                 1
```

```
      :
    547  WHLFYR     c                                    1     3964
    548  WHLFYR_    m                                    1
    549  FFTAXOWE   s  -0.3368E+05   0.3997E+06
    550  FSTAXOWE   s   -3309.       0.7223E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      383
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       48        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0

Number of trees in ensemble: 500
Number of variables used for splitting: 21
Warning: No linear splits; number of S variables must be < 225
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
No interaction splits
No linear splits
Max number of splits on N and S variables: 1000
Maximum number of split levels: 18
Minimum node sample size: 2
Mean number of terminal nodes:    59.11

Classification matrix for training sample:
Predicted       True class
class              C         D         T
C                639        31         4
D                839      2400        48
T                  0         0         4
Total           1478      2431        56

Number of cases used for tree construction: 3965
Number misclassified: 922
Resubstitution estimate of mean misclassification cost: .2325

Number of OOB cases: 3965
Number OOB misclassified: 1130
OOB estimate of mean misclassification cost: .2850
Mean number of trees per OOB observation: 183.90
```

```
Predicted class probabilities are stored in gf.pro
```

Following are the top few rows of the file `gf.pro`, which give the estimated class posterior probabilities and the predicted and observed values of each case in the data.

```
train   "P(C)"       "P(D)"       "P(T)"    predicted  observed
  y   0.59569E+00  0.39907E+00  0.52325E-02    "C"       "C"
  y   0.23720E+00  0.75292E+00  0.98837E-02    "D"       "D"
  y   0.26304E+00  0.73013E+00  0.68219E-02    "D"       "D"
  y   0.31232E+00  0.67988E+00  0.78014E-02    "D"       "D"
  y   0.41418E+00  0.57643E+00  0.93887E-02    "D"       "D"
  y   0.37058E+00  0.46101E+00  0.16841E+00    "D"       "T"
  y   0.35868E+00  0.62997E+00  0.11344E-01    "D"       "D"
```

20.2 Bagged GUIDE

This option uses an ensemble of **pruned** GUIDE trees. It often takes longer to execute and does not appear to produce more accurate results. It is made available for research purposes.

21 Other features

21.1 Pruning with test samples

GUIDE typically has three pruning options for deciding the size of the final tree: (i) cross-validation, (ii) test sample, and (iii) no pruning. Test-sample pruning is available only when there are no derived variables, such as creation of dummy indicator variables when 'b' variables are present. If test-sample pruning is chosen, the program will ask for the name of the file containing the test samples. This file must have the same column format as the training sample file. Pruning with test-samples or no pruning are non-default options.

21.2 Prediction of test samples

GUIDE can produce R code to predict future observations from all except kernel and nearest neighbor classification and ensemble models. This is also a non-default option.

Predictions of the training data for all models can be obtained, however, at the time of tree construction. This feature can be used to obtain predictions on "test samples" (i.e., observations that are not used in tree construction) by adding them to the training sample file. There are two ways to distinguish the test observations from the training observations:

1. Use a *weight* variable (designated as `W` in the DSC file) that takes value 1 for each training observation and 0 or each test observation.
2. Replace the `D` values of the test observations with the missing value code.

For tree construction, GUIDE does not use observations in the training sample file that have zero weight.

21.3 GUIDE in R and in simulations

GUIDE can be used in simulations or used repeatedly on bootstrap samples to produce an ensemble of tree models. For the latter,

1. Create a file (with name `data.txt`, say) containing one set of bootstrapped data.
2. Create a DSC file (with name `desc.txt`, say) that refers to `data.txt`.
3. Create an input file (with name `input.txt`, say) that refers to `desc.txt`.
4. Write a batch program (Windows) or a shell script (Linux or Macintosh) that repeatedly:
   (a) replaces the file `data.txt` with new bootstrapped samples;
   (b) calls GUIDE with the command: `guide < input.txt`; and
   (c) reads and processes the results from each GUIDE run.

   In R, the command in step 4b depends on the operating system. If the GUIDE program and the files `data.txt` and `input.txt` are in the same folder as the working R directory, the command is:

   Linux/Macintosh: `system("guide < input.txt > log.txt")`

   Windows: `shell("guide < input.txt > log.txt")`

   If the files are not all in the same folder, full path names must be given. Here `log.txt` is a text file that stores messages during execution. If GUIDE does not run successfully, errors are also written to `log.txt`.

21.4 Generation of powers and products

GUIDE allows the creation of certain powers and products of regressor variables on the fly. Specifically, variables of the form $X_1^p X_2^q$, where X_1 and X_2 are numerical predictor variables and p and q are integers, can be created by adding one or more lines of the form

```
0 i p j q a
```

at the end of the DSC file. Here `i` and `j` are integers giving the column numbers of variables X_1 and X_2, respectively, in the data file and `a` is one of the letters `n`, `s`, or `f` (corresponding to a numerical variable used for both splitting and fitting, splitting only, or fitting only).

To demonstrate, suppose we wish to fit a piecewise quadratic model in the variable `wtgain` in the birthweight data. This is easily done by adding one line to the file `birthwt.dsc`. First we assign the `s` (for splitting only) designator to every numerical predictor except `wtgain`. This will prevent all variables other than `wtgain` from acting as regressors in the piecewise quadratic models. To create the variable $\texttt{wtgain}^2$, add the line

```
0 8 2 8 0 f
```

to the end of `birthwt.dsc`. The `8`'s in the above line refer to the column number of the variables `wtgain` in the data file, and the `f` tells the program to use the variable $\texttt{wtgain}^2$ for fitting terminal node models only. Note: The line defines $\texttt{wtgain}^2$ as $\texttt{wtgain}^2 \times \texttt{wtgain}^0$. Since we can equivalently define the variable by $\texttt{wtgain}^2 = \texttt{wtgain}^1 \times \texttt{wtgain}^1$, we could also have used the line: "`0 8 1 8 1 f`".

The resulting DSC file now looks like this:

```
birthwt.dat
NA
1
1 weight d
2 black c
3 married c
4 boy c
5 age s
6 smoke c
7 cigsper s
8 wtgain n
9 visit c
10 ed c
11 lowbwt x
0 8 2 8 0 f
```

When the program is given this DSC file, the output will show the regression coefficients of `wtgain` and `wtgain`^{2} in each terminal node of the tree.

21.5 Data formatting functions

GUIDE has a utility function for reformatting data files into forms required by some old statistical software packages:

1. R/Splus: Fields are space delimited. Missing values are coded as `NA`. Each record is written on one line. Variable names are given on the first line.

2. SAS: Fields are space delimited. Missing values are coded with periods. Character strings are truncated to eight characters. Spaces within character strings are replaced with underscores (_).

3. TEXT: Fields are comma delimited. Empty fields denote missing values. Character strings longer than eight characters are truncated. Each record is written on one line. Variable names are given on the first line.

4. STATISTICA: Fields are comma delimited. Commas in character strings are stripped. Empty fields denote missing values. Each record occupies one line.

5. SYSTAT: Fields are comma delimited. Strings are truncated to eight characters. Missing character values are replaced with spaces, missing numerical values with periods. Each record occupies one line.

6. BMDP: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are indicated by asterisks. Variable names longer than eight characters are truncated.

7. DataDesk: Fields are space delimited. Missing categorical values are coded with question marks. Missing numerical values are coded with asterisks. Each record is written on one line. Spaces within categorical values are replaced with underscores. Variable names are given on the first line of the file.

8. MINITAB: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are coded with asterisks. Variable names longer than eight characters are truncated.

9. NUMBERS: Same as **TEXT** option except that categorical values are converted to integer codes.

10. C4.5: This is the format required by the C4.5 (Quinlan, 1993) program.

11. ARFF: This is the format required by the WEKA (Witten and Frank, 2000) programs.

Following is a sample session where the NHTSA comma-separated data are reformatted to tab-delimited for R or Splus.

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: format.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1): 3
 Name of batch output file: format.out
 Input 1 if D variable is categorical, 2 if real ([1:2], <cr>=1):
 Input name of DSC file (max 100 characters);
 enclose with matching quotes if it has spaces: nhtsaclass.dsc
nhtsaclass.dsc
 Reading DSC file ...
 Training sample file: nhtsadata.csv
 Missing value code: NA
 Records in data file start on line 2
 Warning: 48 N variables changed to S
 Dependent variable is HIC2
 Reading data file ...
 Number of records in data file: 3310
 Length of longest entry in data file: 19
 Checking for missing values ...
 Total number of cases: 3310
 Number of classes: 2

 Warning: "x" variables will be excluded
 Choose one of the following data formats:
               Field  Miss.val.codes
No. Name       Separ  char.   numer. Remarks
-------------------------------------------------------------------
 1  R/Splus    space  NA      NA     1 line/case, var names on 1st line
 2  SAS        space  .       .      strings trunc., spaces -> '_'
 3  TEXT       comma  empty   empty  1 line/case, var names on 1st line
 4  STATISTICA comma  empty   empty  1 line/case, commas stripped
                                     var names on 1st line
 5  SYSTAT     comma  space   .      1 line/case, var names on 1st line
                                     strings trunc. to 8 chars
 6  BMDP       space          *      strings trunc. to 8 chars
                                     cat values -> integers (alph. order)
```

```
 7  DATADESK   space  ?       *      1 line/case, var names on 1st line
                                     spaces -> '_'
 8  MINITAB    space          *      cat values -> integers (alph. order)
                                     var names trunc. to 8 chars
 9  NUMBERS    comma  NA      NA     1 line/case, var names on 1st line
                                     cat values -> integers (alph. order)
10  C4.5       comma  ?       ?      1 line/case, dependent variable last
11  ARFF       comma  ?       ?      1 line/case
-------------------------------------------------------------------
 0                                   abort this job
 Input your choice ([0:11], <cr>=1):
 Input name of new data file: newdata.txt
 Input file is created!
 Run GUIDE with the command: guide < format.in
```

A CE variables

Table 11: Some CE variables and their missing rates (if any)

Name	Definition	Missing
AGE_REF	Age of reference person	
AGE2	Age of spouse	0.44
ALCBEVCQ	Alcoholic beverages this quarter	
ALCBEVPQ	Alcoholic beverages last quarter	
ALLFULCQ	Fuel oil and other fuels this quarter	
ALLFULPQ	Fuel oil and other fuels last quarter	
APPARCQ	Apparel and services this quarter (MENBOYCQ + WOMGRLCQ + CHLDRNCQ + FOOTWRCQ + OTHAPLCQ)	
APPARPQ	Apparel and services last quarter (same composition as APPARCQ)	
AS_COMP1	Number of males age 16 and over in CU	
AS_COMP2	Number of females age 16 and over in CU	
AS_COMP3	Number of males age 2 through 15 in CU	
AS_COMP4	Number of females age 2 through 15 in CU	
AS_COMP5	Number of members under age 2 in CU	
BATHRMQ	Number of complete baths in this unit	
BBYDAYCQ	Babysitting and child day care this quarter	
BBYDAYPQ	Babysitting and child day care last quarter	
BEDROOMQ	Number of bedrooms in CU	0.01
BLS_URBN	Is this CU located in an urban or rural area? (1=urban, 2=rural)	
BUILDING	Which of these descriptions from the list best describes this building? (1–11)	
BUILT	Year property was built	0.23
BUSCREEN	Has household had business expenses that could be reimbursed? (1=yes, 2=no)	<0.01
CARTKNCQ	Cars and trucks, new (net outlay) this quarter	
CARTKNPQ	Cars and trucks, new (net outlay) last quarter	
CARTKUCQ	Cars and trucks, used (net outlay) this quarter	
CARTKUPQ	Cars and trucks, used (net outlay) last quarter	
CASHCOCQ	Cash contributions this quarter	
CASHCOPQ	Cash contributions last quarter	

CHILDAGE	Age of children of reference person (0=no children, 1=all children less than 6, 2=oldest child 6–11 and at least one child less than 6, 3=all children 6–11, 4=oldest child 12–17 and at least one child less than 12, 5=all children 12–17, 6=oldest child greater than 17 and at least one child less than 17, 7=all children greater than 17)	
CREDFINX	What was the total amount paid in finance, late charges, and interest for all cards in last month?	0.82
CREDITB	Could you tell me which range that best reflects the total amount owed on all major credit cards including store cards and gas cards? (1=0–499, 2=500–999, 3=1000-2499, 4=2500–9999, 5=10000–34999, 6=35K and over)	0.99
CREDITBX	Median bracket range of CREDITB	0.99
CREDITX	Total amount owed on all cards	0.81
CREDTYRX	Total amount owed on all cards one year ago today	0.90
CREDYR	Did you have any credit cards including store cards and gas cards one year ago today? (1=yes, 2=no)	0.99
CREDYRB	Range that best reflects the total amount owed on all major credit cards including store cards and gas cards one year ago today (1=0–499, 2=500–999, 3=1000–2499, 4=2500–9999, 5=10000–34999, 6=35K and over)	0.99
CREDYRBX	Median bracket range of CREDYRB	0.99
CUTENURE	Housing tenure (1=homeowner with mortgage, 2=homeowner without mortgage, 3=homeowner, mortgage not reported, 4=rented, 5=occupied without payment of rent, 6=student housing)	
DEFBENRP	Do you have a defined retirement plan, such as a pension, from an employer? (1=yes, 2=no)	0.77
DIRACC	Is access to the quarters direct or through another unit? (1=direct, 2=another)	0.04
DIVISION	Census division (1=New England, 2=Middle Atlantic, 3=East North Central, 4=West North Central, 5=South Atlantic, 6=East South Central, 7=West South Central, 8=Mountain, 9=Pacific)	0.07
DOMSRVCQ	Domestic services this quarter	
DMSXCCCQ	Domestic services excluding child care this quarter	
DMSXCCPQ	Domestic services excluding child care last quarter	
DOMSRVPQ	Domestic services last quarter	

EARNCOMP	Composition of earners (1=reference person only, 2=reference person and spouse, 3=reference person, spouse and others, 4=reference person and others, 5=spouse only, 6=spouse and others, 7=others, 8=no earners)	
ECARTKNC	Outlays for new vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKNP	Outlays for new vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKUC	Outlays for used vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKUP	Outlays for used vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EDUC_REF	Education of reference person (10=grades 1–8; 11=grades 9–12, no degree; 12=high school graduate; 13=some college, no degree; 14=Assocaite's degree in college; Bachelors degree; 16=Masters degree or professional/doctorate degree)	
EDUCA2	Education level of spouse (same levels as EDUC_REF)	0.44
EDUCACQ	Education this quarter	
EDUCAPQ	Education last quarter	
EENTMSCC	Miscellaneous entertainment outlays this quarter including photographic and sports equipment and boat and RV rentals	
EENTMSCP	Miscellaneous entertainment outlays last quarter including photographic and sports equipment and boat and RV rentals	
EENTRMTC	Total entertainment outlays this quarter including sound systems, sports equipment, toys, cameras, and down payments on boats and campers (FEEADMCQ + TVRDIOCQ + PETTOYCQ + EOTHENTC)	
EENTRMTP	Total entertainment outlays last quarter including sound systems, sports equipment, toys, cameras, and down payments on boats and campers (same composition as EENTRMTC)	

EHOUSNGC	Total housing outlays this quarter including maintenance, fuels, public services, household operations, house furnishings, and mortgage (lump sum home equity loan or line of credit home equity loan) principle and interest (ESHELTRC + UTILCQ + HOUSOPCQ + HOUSEQCQ)
EHOUSNGP	Total housing outlays last quarter including maintenance, fuels, public services, household operations, house furnishings, and mortgage (lump sum home equity loan or line of credit home equity loan) principle and interest (same composition as EHOUSNGC)
ELCTRCCQ	Electricity this quarter
ELCTRCPQ	Electricity last quarter
EMISCELC	Miscellaneous outlays this quarter including reduction of mortgage principal (lump sum home equity loan) on other property (MISCPQ + EMISCMTP)
EMISCELP	Miscellaneous outlays last quarter including reduction of mortgage principal (lump sum home equity loan) on other property (same composition as EMISCELC)
EMISCMTC	Mortgage principal outlays this quarter for other property
EMISCMTP	Mortgage principal outlays last quarter for other property
EMRTPNOC	Mortgage principal outlays this quarter for owned home
EMRTPNOP	Mortgage principal outlays last quarter for owned home
EMRTPNVC	Mortgage principal outlays this quarter for owned vacation home
EMRTPNVP	Mortgage principal outlays last quarter for owned vacation home
EMOTRVHC	Outlays for motored recreational vehicles this quarter
EMOTRVHP	Outlays for motored recreational vehicles last quarter
ENOMOTRC	Outlays for non-motored recreational vehicles this quarter
ENOMOTRP	Outlays for non-motored recreational vehicles last quarter
ENTERTCQ	Entertainment this quarter (FEEADMCQ + TVRDIOCQ + OTHEQPCQ)

ENTERTPQ	Entertainment last quarter (same composition as ENTERTCQ)	
EOTHENTC	Outlays for other entertainment supplies this quarter, equipment, and services including down payments on boats and campers (ENOMOTRC + EMOTRVHC + EENTMSCC)	
EOTHENTP	Outlays for other entertainment supplies last quarter, equipment, and services including down payments on boats and campers (same composition as EOTHENTC)	
EOTHLODC	Outlays for other lodging this quarter such as owned vacation home, including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (OTHLODCQ + EMRTPNVC)	
EOTHLODP	Outlays for other lodging last quarter such as owned vacation home, including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (same composition as EOTHLODC)	
EOTHVEHP	Outlays for other vehicle purchases last quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOTHVEHC	Outlays for other vehicle purchases this quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOTHVEHP	Outlays for other vehicle purchases last quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOWNDWLC	Owned home outlays this quarter including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (OWNDWECQ + EMRTPNOC)	
EOWNDWLP	Owned home outlays last quarter including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (same composition as EOWNDWLC)	
ERANKH	Percent expenditure outlay rank	0.08

ERANKHM	Weighted cumulative percent expenditure outlay ranking of CU to total population
ESHELTRC	Shelter outlays this quarter including mortgage principle and interest for owned home and/or vacation home, rents, insurance, taxes, and maintenance (EOWNDWLC + RENDWECQ + EOTHLODC)
ESHELTRP	Shelter outlays last quarter including mortgage principle and interest for owned home and/or vacation home, rents, insurance, taxes, and maintenance (same composition as ESHELTRC)
ETOTALC	Total outlays this quarter, sum of outlays from all major expenditure categories (FOODCQ + ALCBEVCQ + EHOUSNGC + APPARCQ + ETRANPTC + HEALTHCQ + EENTRMTC + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + EMISCELC + CASHCOCQ + PERINSCQ)
ETOTALP	Total outlays last quarter, sum of outlays from all major expenditure categories (same composition as ETOTALC)
ETOTACX4	Adjusted total outlays this quarter, sum of outlays from all major expenditure categories (FOODCQ + ALCBEVCQ + EHOUSNGC + APPARCQ + ETRANPTC + HEALTHCQ + EENTRMTC + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + MISC1CQ + 4×MISC2CQ + EMISCMTC + PERINSCQ)
ETOTAPX4	Adjusted total outlays last quarter, sum of outlays from all major expenditure categories (same composition as ETOTACX4)
ETRANPTC	Total outlays for transportation this quarter including down payment, principal and finance charges paid on loans, gasoline and motor oil, maintenance and repairs, insurance, public and other transportation, and vehicle rental licenses and other charges (EVEHPURC + GASMOCQ + MAINRPCQ + VEHINSCQ + VRNTLOCQ + PUBTRACQ)

ETRANPTP	Total outlays for transportation last quarter including down payment, principal and finance charges paid on loans, gasoline and motor oil, maintenance and repairs, insurance, public and other transportation, and vehicle rental licenses and other charges (same composition as ETRANPTC)
EVEHPURC	Outlays for vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount (ECARTKNC + ECARTKUC + EOTHVEHC)
EVEHPURP	Outlays for vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount (same composition as EVEHPURC)
FAM_SIZE	Number of Members in CU
FAM_TYPE	Family type (1–9)
FDAWAYCQ	Food away from home this quarter
FDAWAYPQ	Food away from home last quarter
FDHOMECQ	Food at home this quarter
FDHOMEPQ	Food at home last quarter
FDMAPCQ	Meals as pay this quarter
FDMAPPQ	Meals as pay last quarter
FDXMAPCQ	Food away excluding meals as pay this quarter
FDXMAPPQ	Food away excluding meals as pay last quarter
FEEADMCQ	Fees and admissions this quarter
FEEADMPQ	Fees and admissions last quarter
FFTAXOWE	Weighted estimate for Federal tax liabilities for entire CU
FGOVRETM	Amount of government retirement deducted from last pay, annualized for all CU members
FGOVRETX	Amount of government retirement deducted from last pay annualized
FINCBTAX	Total family income before taxes in last 12 months (INTRDVX + INTRDVBX + ROYESTX + ROYESTBX + OTHREGX + OTHREGBX + WELFAREX + WELFREBX + RETSURVX + RETSRVBX + NETRENTX + NETRNTBX + OTHRINCX)
FINDRETX	Money placed in self-employed retirement plan in past year for all CU members

FINLWT21	Sampling weight	
FJSSDEDX	Estimated amount contributed to Social Security by all CU members past 12 mos.	
FLRCVRCQ	Floor coverings this quarter	
FLRCVRPQ	Floor coverings last quarter	
FMLPYYRX	Annual value of free meals received as part of pay	0.99
FOODCQ	Total food this quarter	
FOODPQ	Total food last quarter	
FPRIPENM	Amount of private pensions deducted from last pay, annualized, for all CU members	
FPRIPENX	Amount of private pensions	
FRRDEDM	Amount of Railroad Retirement deducted from last pay, annualized for all CU members	
FRRDEDX	Amount of railroad retirement deducted from last pay annualized	
FRRETIRM	Amount of social security and railroad retirement income, prior to deductions for medical insurance and Medicare, received by all CU members in the past 12 months	
FRRETIRX	Social security and railroad retirement income	
FS_MTHI	In how many of the last 12 months were food stamps or EBTs received?	0.98
FSALARYX	Wage and salary income of all members past 12 mos.	
FSMPFRMX	Family level summation for new variable SEMPFRMX and SMPFRMBX	
FSSIX	Amount supplemental security income from all sources received by all CU members in past 12 months	
FSTAXOWE	Weighted estimate for State tax liabilities for entire CU	
FURNTRCQ	Furniture this quarter	
FURNTRPQ	Furniture last quarter	
GASMOCQ	Gasoline and motor oil this quarter	
GASMOPQ	Gasoline and motor oil last quarter	
FULOILCQ	Fuel oil this quarter	
FULOILPQ	Fuel oil last quarter	
FURNTRCQ	Furniture this quarter	
FURNTRPQ	Furniture last quarter	
HEALTHCQ	Health care this quarter (HLTHINCQ + MEDSRVCQ + PREDRGCQ + MEDSUPCQ)	

HEALTHPQ	Health care last quarter (same composition as HEALTHCQ)	
HIGH_EDU	Highest level of education within the CU (0=never attended, 10=1–8 grade, 11=9–12 grade, 12=HS grad, 13=some college, 14=AA degree, 15=Bachelors, 16=Masters/professional/doctorate)	
HISP_REF	Hispanic origin of reference person (1=Hispanic, 2=non-Hispanic)	
HISP2	Hispanic origin of spouse (1=Hispanic, 2=non-Hispanic)	
HH_CU_Q	Count of CUs in household	
HLFBATHQ	How many half bathrooms are there in this unit?	0.01
HLTHINCQ	Health insurance this quarter	
HLTHINPQ	Health insurance last quarter	
HORREF1	Hispanic origin of reference person (1=Mexican, 2=Mexican-American, 3=Chicano, 4=Puerto Rican, 5=Cuban, 6=Other)	0.96
HORREF2	Hispanic origin of spouse (same codes as HORREF1)	0.98
HOUSCQ	Housing this quarter	
HOUSEQCQ	House furnishings and equipment this quarter (TEXTILCQ + FURNTRCQ + FLRCVRCQ + MAJAPPCQ + SMLAPPCQ + MISCEQCQ)	
HOUSEQPQ	House furnishings and equipment last quarter (same composition as HOUSEQCQ)	
HOUSOPCQ	Household operations this quarter	
HOUSPQ	Housing last quarter	
HOUSOPPQ	Household operations last quarter	
INC_HRS1	Number hours worked per week by reference person	0.38
INC_HRS2	Number hours worked per week by spouse	0.66
INC_RANK	Income rank of CU to total population	
INCLASS2	Income class based on INC_RANK (1=0–0.1667, 2=0.1667–0.3333, 3=0.3334–0.4999, 4=0.5000–0.6666 5=0.6667–0.8333, 6=0.8334–1, 7=incomplete reporting)	
INCNONW1	Reason for not working during past 12 months (1=retired, 2=take care of home, 3=going to school, 4=ill, disabled, unable to work, 5=unable to find work, 6=doing something else)	0.62
INCNONW2	Reason spouse did not work during past 12 months (same codes as INCNONW1)	0.78

INCOMEY1	Employer paying most earnings in past 12 months (1=private company, business or individual, 2=Federal govt, 3=State govt, 4=local govt, 5=self-employed, 6=family business or farm, working without pay)	0.38
INCOMEY2	Employer from which spouse received most earnings during the past 12 months	0.66
INCWEEK1	Weeks worked full or part time in last 12 months	
INCWEEK2	Weeks worked by spouse full or part time last 12 months	0.44
INTRDVX	Amount received in interest or dividend during past 12 mos.	0.37
IRA	Do you have any retirement accounts such as 401(k)s, IRAs, thrift saving plans? (1=yes, 2=no)	0.76
IRAB	Range that best reflects the total value of all retirement accounts such as 401(k)s, IRAs, and thrift savings plans (1=0–1999, 2=2000-9999, 3=10K–49999, 4=50K–199999, 5=200K–449999, 6=450K or more)	0.97
IRAX	Total amount put into retirement accounts past 12 mos.	0.87
IRAYRB	Range which best reflects the total value of all retirement accounts one year ago today (same codes as IRAB)	0.96
IRAYRBX	Median value of bracket range for IRAYRB	0.96
IRAYRX	Total value of retirement accounts one year ago	0.88
JFS_AMT	Annual value of food stamps	
LIFINSCQ	Life and other personal insurance this quarter	
LIFINSPQ	Life and other personal insurance last quarter	
LIQDYRBX	Median value of bracket range for LIQUDYRB	0.96
LIQUDYR	Did you have any checking savings money market accounts, or CDs one year ago? (1=yes, 2=no)	>0.99
LIQUID	Do you have any checking, saving, money market accounts, or CDs? (1=yes, 2=no)	0.76
LIQUIDB	Range that best reflects total value of checking, savings, money market accounts, CDs (1=0–499, 2=500-999, 3=1000–2499, 4=2.5K–9999, 5=10K–34999, 6=35K and over)	0.97
LIQUIDBX	Median value of bracket range LIQUIDB	0.97
LIQUIDX	Total value of all checking, savings, money market, and CD accounts	0.83

LIQUDYRB	Range that best reflects the total value of all checking, savings, money market accounts, and CDs one year ago today (same codes as LIQUIDB)	0.97
LIQUDYRX	Total value of all checking, savings, money market accounts, and CDs one year ago today	0.84
MAINRPCQ	Maintenance and repairs this quarter	
MAINRPPQ	Maintenance and repairs last quarter	
MAJAPPCQ	Major appliances this quarter	
MAJAPPPQ	Major appliances last quarter	
MARITAL1	Marital status of reference person (1=married, 2=widowed, 3=divorced, 4=separated, 5=never married)	
MEALSPAY	Have you received any free meals at work as part of your pay? (1=yes, 2=no)	<0.01
MEDSRVCQ	Medical services this quarter	
MEDSRVPQ	Medical services last quarter	
MEDSUPCQ	Medical supplies this quarter	
MEDSUPPQ	Medical supplies last quarter	
MENBOYCQ	Clothing for men and boys this quarter	
MENSIXCQ	Clothing for men, 16 and over this quarter	
MENSIXPQ	Clothing for men, 16 and over last quarter	
MENBOYPQ	Clothing for men and boys last quarter	
MISC1CQ	Miscellaneous expenditures this quarter	
MISC1PQ	Miscellaneous expenditures last quarter	
MISCEQCQ	Miscellaneous household equipment this quarter	
MISCEQPQ	Miscellaneous household equipment last quarter	
MISCCQ	Miscellaneous expenditures this quarter (MISC1CQ + MISC2CQ)	
MISCPQ	Miscellaneous expenditures last quarter (same composition as MISCCQ)	
MISCTAXX	During past 12 months, what was total amount paid for personal property taxes and other taxes not reported elsewhere by all CU members?	0.99
MISCX4CQ	Adjusted miscellaneous expenditures this quarter (MISC1CQ + 4×MISC2CQ)	
MISCX4PQ	Adjusted miscellaneous expenditures last quarter (same composition as MISCX4CQ)	
MLPAYWKX	About what was the weekly dollar value of these meals?	0.99

MLPYQWKS	For how many weeks did members of your household receive these meals during the past 12 months?	0.99
MRPINSCQ	Maintenance, repairs, insurance, and other expenses this quarter	
MRPINSPQ	Maintenance, repairs, insurance, and other expenses last quarter	
MRTINTCQ	Mortgage interest this quarter	
MRTINTPQ	Mortgage interest last quarter	
MRTPRNOC	Outlays on owned vacation home mortgage principle this quarter	
MRTPRNOP	Outlays on owned vacation home mortgage principle last quarter	
NETRENTB	Range that best reflects the total net rental income or loss during the past 12 months (1=0–999, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	0.99
NETRENTX	What was the amount of net rental income or loss?	0.92
NETRNTBX	Median value of bracket range of NETRENTB	0.99
NTLGASCQ	Natural gas this quarter	
NTLGASPQ	Natural gas last quarter	
NO_EARNR	Number of earners	
NONINCMX	Amount of other money receipts excluded from CU income before taxes received by CU in past 12 months	
NUM_AUTO	Total number of owned cars	
NUM_TVAN	Total number of owned trucks and vans	
OCCUCOD1	Highest paid occupation last 12 months (15 coded values)	0.38
OCCUCOD1	Job in which reference person received most earnings during past 12 months (15 coded values)	0.66
OCCUCOD2	Job in which spouse received most earnings during past 12 months (15 coded values)	0.66
OTHAPLCQ	Other apparel products and services this quarter	
OTHAPLPQ	Other apparel products and services last quarter	
OTHASTB	Range which best reflects the total value of these other financial assets (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	>0.99
OTHASTBX	Median value of bracket range for OTHASTB	>0.99
OTHASTX	Total value of these other financial assets as of today	0.99
OTHENTCQ	Other entertainment this quarter	

OTHENTPQ	Other entertainment last quarter	
OTHEQPCQ	Other equipment and services this quarter (PETTOYCQ + OTHENTCQ)	
OTHEQPPQ	Other equipment and services last quarter (same composition as OTHEQPCQ)	
OTHFINX	Total amount paid in finance, late charges, and interest for all other loans in the last month	0.99
OTHFLSCQ	Other fuels this quarter	
OTHFLSPQ	Other fuels last quarter	
OTHHEXCQ	Other household expenses this quarter	
OTHHEXPQ	Other household expenses last quarter	
OTHLNYR	Did you have any other debt such as medical loans or personal loans one year ago today? (1=yes, 2=no)	>0.99
OTHLNYRB	Range which best reflects the total amount owed on all other loans one year ago today (1=0–499, 2=500-999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99
OTHLODCQ	Other lodging this quarter	
OTHLODPQ	Other lodging last quarter	
OTHLONX	Total amount owed on all other loans	0.99
OTHLYRBX	Median value of bracket range for OTHLONBX	>0.99
OTHREGB	Range best reflects total amount received in Veteran's Administration (VA) payments, unemployment compensation, child support, or alimony during the past 12 months (1=0–1K, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	0.99
OTHREGBX	Median value of bracket range for OTHREGB	0.99
OTHRINCX	Amount received in other income including money from care of foster children, cash scholarships and fellowships, or stipends not based on working	0.97
OTHREGX	Income on a regular basis from any other source such as Veteran's Administration (VA) payments, unemployment compensation, child support, or alimony	0.92
OTHSTYRB	Range which best reflects total value of these other financial assets one year ago today (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	>0.99
OTHSTYRX	Value of these other financial assets one year ago today	0.99
OTHSYRBX	Median value of bracket range for OTHSTYRB	>0.99

OTHVEHCQ	Other vehicles this quarter	
OTHVEHPQ	Other vehicles last quarter	
OWNDWECQ	Owned dwellings this quarter (MRTINTCQ + PROPTXCQ + MRPINSCQ)	
OWNDWEPQ	Owned dwellings last quarter (same composition as OWNDWECQ)	
OWNVACC	Expenditures on owned vacation homes this quarter including mortgage interest, insurance, taxes, maintenance, and miscellaneous household equipment (VOTHRLOC + VMISCHEC)	
OWNVACP	Expenditures on owned vacation homes last quarter including mortgage interest, insurance, taxes, maintenance, and miscellaneous household equipment (same composition as OWNVACC)	
PERINSCQ	Personal insurance and pensions this quarter (LIFINSCQ + RETPENCQ)	
PERINSPQ	Personal insurance and pensions last quarter (same composition as PERINSCQ)	
PERSCACQ	Personal care this quarter	
PERSCAPQ	Personal care last quarter	
PERSLT18	Number of CU members less than 18	
PERSOT64	Number of CU members over 64	
PETTOYCQ	Pets, toys, and playground equipment this quarter	
PETTOYPQ	Pets, toys, and playground equipment last quarter	
POPSIZE	Population size of the PSU (1=more than 5M, 2=1–5M, 3=0.5–1M, 4=100–500K, 5=less than 100K)	
PREDRGCQ	Prescription drugs this quarter	
PREDRGPQ	Prescription drugs last quarter	
PRINEARN	Member number of principal earner (5 coded values)	
PROPTXCQ	Property taxes this quarter	
PROPTXPQ	Property taxes last quarter	
PSU	Primary sampling unit	0.52
PUBTRACQ	Public and other transportation this quarter (TRNTRPCQ + TRNOTHCQ)	
PUBTRAPQ	Public and other transportation last quarter (same composition as PUBTRACQ)	
RACE2	Race of spouse (same codes as REF_RACE)	0.44
READCQ	Reading this quarter	

READPQ	Reading last quarter	
REF_RACE	Race of reference person (1=white, 2=black, 3=native American, 4=Asian, 5=Pacific islander, 6=multi-race)	
REFGEN	Generation of reference person (1=Greatest/Silent: born 1945 or earlier, 3=Baby boomers: 1946–64, 4=Gen X: 1965–80, 5=Millennials: 1981 or later)	
REGION	Region (1=Northeast, 2=Midwest, 3=South, 4=West)	0.01
RELECTRC	Expenditures on electricity for rented vacation homes this quarter	
RELECTRP	Expenditures on electricity for rented vacation homes last quarter	
RENDWECQ	Rented dwelling this quarter (RNTXRPCQ + RNTAPYCQ)	
RENDWEPQ	Rented dwelling last quarter (same composition as RENDWECQ)	
RENTEQVX	Monthly rent if home rented today	0.20
RETPENCQ	Retirement, pensions, social security this quarter	
RETPENPQ	Retirement, pensions, social security last quarter	
RETSRVBX	Median value of bracket range for RETSURVB	0.99
RETSURV	Did you receive income from retirement, survivor, or disability pensions during past 12 months? (1=yes, 2=no)	
RETSURVX	Retirement, survivor, disability pensions received past 12 mos.	0.78
RNATLGAC	Expenditures on natural gas for rented vacation homes this quarter	
RNATLGAP	Expenditures on natural gas for rented vacation homes last quarter	
RNTAPYCQ	Rent as pay this quarter	
RNTAPYPQ	Rent as pay last quarter	
RNTXRPCQ	Rent excluding rent as pay this quarter	
RNTXRPPQ	Rent excluding rent as pay last quarter	
ROOMSQ	Number of rooms in CU living quarters, including finished living areas, excluding all baths	0.01
ROTHRFLC	Expenditures on other fuels for rented vacation homes this quarter	

ROYESTB	Range that best reflects total amount received in royalty income or income from estates and trusts during past 12 months (1=0–1K, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	>0.99
ROYESTBX	Median value of bracket range for ROYESTB	>0.99
ROYESTX	Amount received in royalty income or income from estates and trusts	0.96
RWATERPC	Expenditures on water and public services for rented vacation homes this quarter	
RWATERPP	Expenditures on water and public services for rented vacation homes last quarter	
SEX_REF	Sex of reference person (1=male, 2=female)	
SEX2	Sex of spouse (1=male, 2=female)	0.44
SHELTCQ	Shelter this quarter (OWNDWECQ + RENDWECQ + OTHLODCQ)	
SHELTPQ	Shelter last quarter (same composition as SHELTCQ)	
SMLAPPCQ	Small appliances, miscellaneous housewares this quarter	
SMLAPPPQ	Small appliances, miscellaneous housewares last quarter	
SMSASTAT	Does CU reside inside a Metropolitan Statistical Area (MSA)? (1=yes, 2=no)	
ST_HOUS	Are these living quarters presently used as student housing by a college or university? (1=yes, 2=no)	
STATE	1=AL, 2=AK, 4=AZ, 5=AR, 6=CA, 8=CO, 9=CT, 10=DE, 11=DC, 12=FL, 13=GA, 15=HI, 16=ID, 17=IL, 18=IN, 19=IA, 20=KS, 21=KY, 22=LA, 23=ME, 24=MD, 25=MA, 26=MI, 27=MN, 28=MS, 29=MO, 30=MT, 31=NE, 32=NV, 33=NH, 34=NJ, 36=NY, 37=NC, 39=OH, 40=OK, 41=OR, 42=PA, 44=RI, 45=SC, 46=SD, 47=TN, 48=TX, 49=UT, 51=VA, 53=WA, 54=WV, 55=WI	0.08
STCKYRBX	Median value of bracket range for STOCKYRB	0.98
STDNTYR	Did you have student loans one years ago today? (1=yes, 2=no)	>0.99
STDNTYRB	Range which best reflects the total amount owed on all student loans one year ago today (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99

STDNTYRX	Total amount owed on all student loans one year ago today	0.97
STDTYRBX	Median value of bracket range for STDNTYRB	>0.99
STOCKB	Range which best reflects total value of all directly-held stocks, bonds, and mutual funds (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	0.99
STOCKBX	Median value of bracket range for STOCKB	0.99
STOCKX	Value of directly-held stocks, bonds, mutual funds (median=59,950, mean=411,867)	0.93
STOCKYR	Did you have any directly-held stocks, bonds, or mutual funds one year ago? (1=yes, 2=no)	>0.99
STOCKYRB	Range which best reflects total value of all directly-held stocks, bonds, and mutual funds one year ago today (same codes as STOCKB)	0.98
STOCKYRX	Median value of bracket range of STOCKX	0.93
STUDFINX	Total amount paid in finance, late charges, and interest for all student loans in the last month	0.97
STUDNTB	Range which best reflects the total amount owed on all student loans (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99
STUDNTBX	Median value of bracket range for STUDNTB	>0.99
STUDNTX	Total amount owed on all student loans	0.97
TAIRFARC	Trip expenditures on airfare this quarter	
TAIRFARP	Trip expenditures on airfare last quarter	
TALCBEVC	Total trip expenditures this quarter on alcoholic beverages at restaurants, cafes, and bars	
TALCBEVP	Total trip expenditures last quarter on alcoholic beverages at restaurants, cafes, and bars	
TELEPHCQ	Telephone services this quarter	
TELEPHPQ	Telephone services last quarter	
TENTRMNC	Total trip expenditures on entertainment this quarter including sporting events, movies, and recreational vehicle rentals (TFEESADC + TOTHENTC)	
TENTRMNP	Total trip expenditures on entertainment last quarter including sporting events, movies, and recreational vehicle rentals (same composition as TENTRMNC)	
TEXTILCQ	Household textiles this quarter	
TEXTILPQ	Household textiles last quarter	

TFAREC	Trip expenditures this quarter on transportation fares including airfare, intercity bus, train, and ship fare (TAIRFARC + TOTHFARC)
TFAREP	Trip expenditures last quarter on transportation fares including airfare, intercity bus, train, and ship fare (same composition as TFAREC)
TFEESADC	Trip expenditures on miscellaneous entertainment this quarter including recreation expenses, participation sport fees, and admission fees to sporting events and movies
TFEESADP	Trip expenditures on miscellaneous entertainment last quarter including recreation expenses, participation sport fees, and admission fees to sporting events and movies
TFOODAWC	Food and non-alcoholic beverages this quarter at restaurants, cafes, and fast food places during out-of-town trips
TFOODAWP	Food and non-alcoholic beverages last quarter at restaurants, cafes, and fast food places during out-of-town trips
TFOODHOC	Food and beverages purchased and prepared by CU this quarter during out-of-town trips
TFOODHOP	Food and beverages purchased and prepared by CU last quarter during out-of-town trips
TFOODTOC	Total trip expenditures on food this quarter including both restaurant food and food prepared by CU (TFOODAWC + TFOODHOC)
TFOODTOP	Total trip expenditures on food last quarter including both restaurant food and food prepared by CU (same composition as TFOODTOC)
TGASMOTC	Trip expenditures on gas and oil this quarter
TGASMOTP	Trip expenditures on gas and oil last quarter
TLOCALTC	Trip expenditures this quarter on local transportation including taxis, buses etc.
TLOCALTP	Trip expenditures last quarter on local transportation including taxis, buses etc.
TOBACCCQ	Tobacco and smoking supplies this quarter
TOBACCPQ	Tobacco and smoking supplies last quarter
TOTEX4CQ	Adjusted total expenditures this quarter (TOTEXPCQ - MISCCQ + MISC1CQ + 4 × MISC2CQ)
TOTEX4PQ	Adjusted total expenditures last quarter (same composition as TOTEX4CQ)

TOTEXPCQ	Total expenditures this quarter (FOODCQ + ALCBEVCQ + HOUSCQ + APPARCQ + TRANSCQ + HEALTHCQ + ENTERTCQ + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + MISCCQ + CASHCOCQ + PERINSCQ)
TOTEXPPQ	Total expenditures last quarter (same composition as TOTEXPCQ)
TOTHENTC	Trip expenditures on recreational vehicle rentals this quarter including campers, boats, and other vehicles
TOTHENTP	Trip expenditures on recreational vehicle rentals last quarter including campers, boats, and other vehicles
TOTHFARC	Tip expenditures this quarter on other transportation fares including intercity bus and train fare, and ship fare
TOTHFARP	Tip expenditures last quarter on other transportation fares including intercity bus and train fare, and ship fare
TOTHRLOC	Total trip expenditures on lodging this quarter including rent for vacation home, and motels
TOTHRLOP	Total trip expenditures on lodging last quarter including rent for vacation home, and motels
TOTHTREC	Trip expenditures this quarter for other transportation expenses including parking fees, and tolls
TOTHTREP	Trip expenditures last quarter for other transportation expenses including parking fees, and tolls
TOTHVHRC	Trip expenditures on other vehicle rentals this quarter
TOTHVHRP	Trip expenditures on other vehicle rentals last quarter
TOTXEST	Estimated total taxes paid (FFTAXOWE + FSTAXOWE + MISCTAXX)
TRANSCQ	Transportation this quarter (CARTKNCQ + CARTKUCQ + OTHVEHCQ + GASMOCQ + VEHFINCQ + MAINRPCQ + VEHINSCQ + VRNTLOCQ + PUBTRACQ)
TRANSPQ	Transportation last quarter (same composition as TRANSCQ)
TRNOTHCQ	Local public transportation, excluding on trips this quarter
TRNOTHPQ	Local public transportation, excluding on trips last quarter
TRNTRPCQ	Public and other transportation on trips this quarter

TRNTRPPQ	Public and other transportation on trips last quarter
TTOTALC	Total of all trip expenditures this quarter (TFOODTOC + TALCBEVC + TOTHRLOC + TTRANPRC + TENTRMNPC)
TTOTALP	Total of all trip expenditures last quarter (same composition as TTOTALC)
TTRANPRC	Total trip expenditures on transportation this quarter including airfare, local transportation, tolls and parking fees, and car rentals (TGASMOTC + TVRENTLC + TTRNTRIC)
TTRANPRP	Total trip expenditures on transportation last quarter including airfare, local transportation, tolls and parking fees, and car rentals (same composition as TTRANPRC)
TTRNTRIC	Trip expenditures this quarter for public transportation, including airfares (TFAREC + TLOCALTC)
TTRNTRIP	Trip expenditures last quarter for public transportation, including airfares (same composition as TTRNTRIC)
TVRDIOCQ	Televisions, radios, and sound equipment this quarter
TVRDIOPQ	Televisions, radios, and sound equipment last quarter
TVRENTLC	Trip expenditures on vehicle rentals and other fees this quarter (TCARTRKC + TOTHVHRC + TOTHTREC)
TVRENTLP	Trip expenditures on vehicle rentals and other fees last quarter (same composition as TVRENTLC)
UNISTRQ	How many housing units, both occupied and vacant, are in this structure? (1=only other units, 2=mobile home or trailer, 3=one, detached, 4=one, attached, 5=2, 6=3–4, 7=5–9, 8=10–19, 9=20–49, 10=50 or more)
UTILCQ	Utilities, fuels and public services this quarter (NTLGASCQ + ELCTRCCQ + ALLFULCQ + TELEPHCQ + WATRPSCQ)
UTILOWNC	Expenditures on owned vacation home utilities this quarter including water, trash, electricity, and fuels (VFUELOIC + VOTHRFLC + VELECTRC + VNATLGAC + VWATERPC)
UTILOWNP	Expenditures on owned vacation home utilities last quarter including water, trash, electricity, and fuels (same composition as UTILOWNC)
UTILPQ	Utilities, fuels and public services last quarter

UTILRNTC	Expenditures on rented vacation home utilities this quarter including water, trash, electricity, and fuels (RFUELOIC + ROTHRFLC + RELECTRC + RNATLGAC + RWATERPC)	
UTILRNTP	Expenditures on rented vacation home utilities last quarter including water, trash, electricity, and fuels (same composition as UTILRNTC)	
VEHFINCQ	Vehicle finance charges this quarter	
VEHFINPQ	Vehicle finance charges last quarter	
VEHICTAX	Personal property taxes for vehicles	0.92
VEHINSCQ	Vehicle insurance this quarter	
VEHINSPQ	Vehicle insurance last quarter	
VEHQ	Total number of owned vehicles	
VEHQL	Total number of leased autos, trucks and vans	
VELECTRC	Expenditures on electricity for owned vacation homes this quarter	
VELECTRP	Expenditures on electricity for owned vacation homes last quarter	
VFUELOIC	Expenditures on fuel oil for owned vacation homes this quarter	
VFUELOIP	Expenditures on fuel oil for owned vacation homes last quarter	
VNATLGAC	Expenditures on natural gas for owned vacation homes this quarter	
VNATLGAP	Expenditures on natural gas for owned vacation homes last quarter	
VOTHRFLC	Expenditures on other fuels for owned vacation homes this quarter	
VOTHRFLP	Expenditures on other fuels for owned vacation homes last quarter	
VOTHRLOP	Expenditures on owned vacation homes last quarter including mortgage interest, insurance, taxes, and maintenance	
VOTHRLOC	Expenditures on owned vacation homes this quarter including mortgage interest, insurance, taxes, and maintenance	
VRNTLOCQ	Vehicle rental, leases, licenses, and other charges this quarter	

VRNTLOPQ	Vehicle rental, leases, licenses, and other charges last quarter	
VWATERPC	Expenditures on water and public services for owned vacation homes this quarter	
VWATERPP	Expenditures on water and public services for owned vacation homes last quarter	
WATRPSCQ	Water and other public services this quarter	
WATRPSPQ	Water and other public services last quarter	
WELFAREX	Amount received from public assistance or welfare including money received from job training grants	0.99
WELFREBX	Median of bracket range of WELFAREB	0.99
WHLFYR	Did you own any whole life insurance or other life insurance policies that can be surrendered for cash or borrowed against prior to the death of the person insured one year ago today? (1=yes, 2=no)	>0.99
WHLFYRB	Range which best reflects total surrender value of these policies one year ago today (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	0.99
WHLFYRBX	Median value of bracket range for WHLFYRB	0.99
WHLFYRX	Total surrender value of these policies one year ago today	0.98
WHOLIFB	Range which best reflects the total surrender value of these policies (same codes as WHLFYRB) >0.99	
WHOLIFBX	Median value of bracket range for WHOLIFB	>0.99
WHOLIFX	Total surrender value of these policies as of today	0.98

References

Andersen, P. K., Hansen, M. G., and Klein, J. P. (2004). Regression analysis of restricted mean survival time based on pseudo-observations. *Lifetime Data Analysis*, 10:335–350.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Table 12: PSU codes

S11A	Boston-Cambridge-Newton, MA-NH
S12A	New York-Newark-Jersey City, NY-NJ-PA
S12B	Philadelphia-Camden-Wilmington, PA-NJ-DE-MD
S23A	Chicago-Naperville-Elgin, IL-IN-WI
S23B	Detroit-Warren-Dearborn, MI
S24A	Minneapolis-St. Paul-Bloomington, MN-WI
S24B	St. Louis, MO-IL
S35A	Washington-Arlington-Alexandria, DC-VA-MD-WV
S35B	Miami-Fort Lauderdale-West Palm Beach, FL
S35C	Atlanta-Sandy Springs-Roswell, GA
S35D	Tampa-St. Petersburg-Clearwater, FL
S35E	Baltimore-Columbia-Towson, MD
S37A	Dallas-Fort Worth-Arlington, TX
S37B	Houston-The Woodlands-Sugar Land, TX
S48A	Phoenix-Mesa-Scottsdale, AZ
S48B	Denver-Aurora-Lakewood, CO
S49A	Los Angeles-Long Beach-Anaheim, CA
S49B	San Francisco-Oakland-Hayward, CA
S49C	Riverside-San Bernardino-Ontario, CA
S49D	Seattle-Tacoma-Bellevue, WA
S49E	San Diego-Carlsbad, CA
S49F	Honolulu, HI
S49G	Anchorage, AK

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont.

Broekman, B. F. P., Niti, M., Nyunt, M. S. Z., Ko, S. M., Kumar, R., and Ng, T. P. (2011). Validation of a brief seven-item response bias-free geriatric depression scale. *American Journal of Geriatric Psychiatry*, 19:589–596.

Broekman, B. F. P., Nyunt, S. Z., Niti, M., Jin, A. Z., Ko, S. M., Kumar, R., Fones, C. S. L., and Ng, T. P. (2008). Differential item functioning of the geriatic depression scale in an Asian population. *Journal of Affective Disorders*, 108:285–290.

Cameron, A. A. and Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge University Press.

Chambers, J. M. and Hastie, T. J. (1992). An appetizer. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, pages 1–12. Wadsworth & Brooks/Cole, Pacific Grove.

Chan, K.-Y. and Loh, W.-Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics*, 13:826–852.
http://www.stat.wisc.edu/~loh/treeprogs/lotus/lotus.pdf.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y., and Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica*, 4:143–167.
http://www3.stat.sinica.edu.tw/statistica/j4n1/j4n18/j4n18.htm.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica*, 5:641–666.
http://www3.stat.sinica.edu.tw/statistica/j5n2/j5n217/j5n217.htm.

Chaudhuri, P. and Loh, W.-Y. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, 8:561–576.
http://www.stat.wisc.edu/~loh/treeprogs/guide/quantile.pdf.

Chen, P. Y. and Tsiatis, A. A. (2001). Causal inference on the difference of the restricted mean lifetime between two groups. *Biometrics*, 57:1030–1038.

Choi, Y., Ahn, H., and Chen, J. J. (2005). Regression trees for analysis of count data with extra Poisson variation. *Computational Statistics & Data Analysis*, 49(3):893–915.

Connors, Jr., A. F., , Speroff, T., Dawson, N. V., et al. (1996). The effectiveness of right heart catheterization in the initial care of critically ill patients. *JAMA*, 276(11):889–897.

Deb, P. and Trivedi, P. K. (1997). Demand for medical care by the elderly: a finite mixture approach. *Journal of Applied Econometrics*, 12:313–336.

Hothorn, T. (2017). *TH.data: TH's Data Archive*. R package version 1.0-8.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15:651–674.

Hothorn, T. and Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in r. *Journal of Machine Learning Research*, 16:3905–3909.

Italiano, A. (2011). Prognostic or predictive? It's time to get back to definitions! *Journal of Clinical Oncology*, 29:4718.

Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604. http://www.stat.wisc.edu/~loh/treeprogs/cruise/cruise.pdf.

Kim, H. and Loh, W.-Y. (2003). Classification trees with bivariate linear discriminant node models. *Journal of Computational and Graphical Statistics*, 12:512–530. http://www.stat.wisc.edu/~loh/treeprogs/cruise/jcgs.pdf.

Kim, H., Loh, W.-Y., Shih, Y.-S., and Chaudhuri, P. (2007). Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, 39:565–579. http://www.stat.wisc.edu/~loh/treeprogs/guide/iie.pdf.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Loh, W.-Y. (1987). Calibrating confidence coefficients. *Journal of the American Statistical Association*, 82:155–162.

Loh, W.-Y. (1991). Bootstrap calibration for confidence interval construction and selection. *Statistica Sinica*, 1:477–491.

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386. http://www3.stat.sinica.edu.tw/statistica/j12n2/j12n21/j12n21.htm.

Loh, W.-Y. (2006a). Logistic regression tree analysis. In Pham, H., editor, *Handbook of Engineering Statistics*, pages 537–549. Springer. http://www.stat.wisc.edu/~loh/treeprogs/lotus/springer.pdf.

Loh, W.-Y. (2006b). Regression tree models for designed experiments. In Rojo, J., editor, *The Second Erich L. Lehmann Symposium–Optimality*, volume 49, pages 210–228. Institute of Mathematical Statistics Lecture Notes-Monograph Series. arxiv.org/abs/math.ST/0611192.

Loh, W.-Y. (2008a). Classification and regression tree methods. In Ruggeri, F., Kenett, R., and Faltin, F. W., editors, *Encyclopedia of Statistics in Quality and Reliability*, pages 315–323. Wiley, Chichester, UK. http://www.stat.wisc.edu/~loh/treeprogs/guide/eqr.pdf.

Loh, W.-Y. (2008b). Regression by parts: Fitting visually interpretable models with GUIDE. In Chen, C., Härdle, W., and Unwin, A., editors, *Handbook of Computational Statistics*, pages 447–469. Springer. http://www.stat.wisc.edu/~loh/treeprogs/guide/handbk.pdf.

Loh, W.-Y. (2009). Improving the precision of classification trees. *Annals of Applied Statistics*, 3:1710–1737. http://www.stat.wisc.edu/~loh/treeprogs/guide/aoas260.pdf.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14–23. http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf.

Loh, W.-Y. (2014). Fifty years of classification and regression trees (with discussion). *International Statistical Review*, 34:329–370. http://www.stat.wisc.edu/~loh/treeprogs/guide/LohISI14.pdf.

Loh, W.-Y. (2021). Logistic regression tree analysis. In Pham, H., editor, *Handbook of Engineering Statistics*. Springer, 2nd edition. To appear. http://www.stat.wisc.edu/~loh/treeprogs/guide/logistic2.pdf.

Loh, W.-Y., Cao, L., and Zhou, P. (2019a). Subgroup identification for precision medicine: a comparative review of thirteen methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(5):e1326. http://www.stat.wisc.edu/~loh/treeprogs/guide/wires19.pdf.

Loh, W.-Y., Chen, C.-W., and Zheng, W. (2007). Extrapolation errors in linear model trees. *ACM Trans. Knowl. Discov. Data*, 1(2):6.
http://www.stat.wisc.edu/~loh/treeprogs/guide/acm.pdf.

Loh, W.-Y., Eltinge, J., Cho, M. J., and Li, Y. (2019b). Classification and regression trees and forests for incomplete data from sample surveys. *Statistica Sinica*, 29:431–453.
http://www.stat.wisc.edu/~loh/treeprogs/guide/LECL19.pdf.

Loh, W.-Y., Fu, H., Man, M., Champion, V., and Yu, M. (2016). Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables. *Statistics in Medicine*, 35:4837–4855.
http://www.stat.wisc.edu/~loh/treeprogs/guide/LFMCY16.pdf.

Loh, W.-Y., He, X., and Man, M. (2015). A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine*, 34:1818–1833.
http://www.stat.wisc.edu/~loh/treeprogs/guide/LohHeMan15.pdf.

Loh, W.-Y., Man, M., and Wang, S. (2019c). Subgroups from regression trees with adjustment for prognostic effects and post-selection inference. *Statistics in Medicine*, 38:545–557.
http://www.stat.wisc.edu/~loh/treeprogs/guide/sm19.pdf.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7:815–840.
http://www3.stat.sinica.edu.tw/statistica/j7n4/j7n41/j7n41.htm.

Loh, W.-Y. and Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). *Journal of the American Statistical Association*, 83:715–728.
http://www.stat.wisc.edu/~loh/treeprogs/fact/LV88.pdf.

Loh, W.-Y., Zhang, Q., Zhang, W., and Zhou, P. (2020). Missing data, imputation and regression trees. *Statistica Sinica*, 30:1697–1722.
http://www.stat.wisc.edu/~loh/treeprogs/guide/LZZZ20.pdf.

Loh, W.-Y. and Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *Annals of Applied Statistics*, 7:495–522.
http://www.stat.wisc.edu/~loh//treeprogs/guide/AOAS596.pdf.

Loh, W.-Y. and Zhou, P. (2020). The GUIDE approach to subgroup identification. In Ting, N., Cappelleri, J. C., Ho, S., and Chen, D.-G., editors, *Design and analysis of Subgroups with Biopharmaceutical Applications*, pages 147–165. Springer. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ20.pdf.

Loh, W.-Y. and Zhou, P. (2021). Variable importance scores. *Journal of Data Science*, 19(4):569–592. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ21.pdf.

Marc, L. G., Raue, P. J., and Bruce, M. L. (2008). Screening performance of the 15-item geriatric depression scale in a diverse elderly home care population. *American Journal of Geriatric Psychiatry*, 16:914–921.

Murnane, R. J., Boudett, K. P., and Willett, J. B. (1999). Do male dropouts benefit from obtaining a GED, postsecondary education, and training? *Evaluation Reviews*, 23:475–502.

Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Schmoor, C., Olschewski, M., and Schumacher, M. (1996). Randomized and non-randomized patients in clinical trials: experiences with comprehensive cohort studies. *Statistics in Medicine*, 15:263–271.

Singer, J. D. and Willett, J. B. (2003). *Applied Longitudinal Data Analysis*. Oxford University Press, New York, NY.

Therneau, T., Atkinson, B., and Ripley, B. (2017). *rpart: Recursive Partitioning and Regression Trees*. CRAN.R-project.org/package=rpart.

Tian, L., Zhao, L., and Wei, L. J. (2014). Predicting the restricted mean event time with the subject's baseline covariates in survival analysis. *Biostatistics*, 15:222–233.

Witten, I. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, San Fransico, CA. http://www.cs.waikato.ac.nz/ml/weka.

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. *Journal of Statistical Software*, 16(9):1–16.