

## Creating an R package

An R “package” facilitates the organization and sharing of R code, its user manual, data, and tests. It facilitates open and reproducible research. To create a package,

- Install software:
  - `install.packages("devtools"); require("devtools")` gives access to `create()`, `check()`, and `build()` (below).
  - `install.packages("roxygen2"); require("roxygen2")` gives access to `document()` (below), which is also called by `check()`.
  - (– Windows users: install Rtools from <http://cran.r-project.org/bin/windows/Rtools>. Find your R version by typing `version` in the console.)
- `create(path)` creates a skeleton package consisting of the new folders `path` and `path/R` (for “.R” code files); and the new files `path/DESCRIPTION` (for package metadata like title, author, version, dependencies) and others we’ll ignore. e.g. `create("jgUtilities")`
- Revise `path/DESCRIPTION`. e.g. See `jgUtilities/DESCRIPTION`.
- Write functions, one per “.R” file, in your `path/R` folder:
  - Put a function `f()` in the file `f.R`.
  - Write `f()`’s user manual (visible via `?f`) by pasting these comments above `f()`’s code and revising them:

```
## (title at top of help page)
##
## (Description paragraph)
## @param x (description of x; one line per parameter: Arguments section)
## @return (Value section)
## @details (Details section)
## @export
## @examples
## # (Examples section)
## f(x)
##
```

e.g. Copy `random.sort.R` and `baby.factorial.R` into `jgUtilities/R`.

- Include data, if any:
  - Create a `path/data` folder via the “New Folder” button on RStudio’s “Files” pane.
  - Save a data object `x` (a placeholder) via `save(x, file="path/data/x.RData")`. e.g.
 

```
x = data.frame(height=1:3, weight=4:6); save(x, file="jgUtilities/data/x.RData")
```
  - Document data object `x` by pasting these comments into `path/R/x.R` and revising:

```
#' This Is A One-line Title Describing x
#'
#' This is the Description paragraph. x is my favorite data set. Blah, blah,
#' blah.
#'
#' @format x is a data frame with 3 rows and 2 variables:
#' \describe{
#'   \item{h}{height (inches)}
#'   \item{w}{weight (pounds)}
#' }
#' @source
#' These data are from an experiment I did with my friend William
#' (\url{http://en.wikipedia.org/wiki/William_Sealy_Gosset})
#' at the Guinness Brewery in Dublin in 1908.
"x"
```

- (• `document(pkg=".")` creates a `path/man` folder (for user “manual” file *output*) and writes “.Rd” (for “R documentation” help) files in it, one for each “.R” file, from your function comments. e.g. `document(pkg="jgUtilities")`)
- `check(pkg=".")` calls `document()` and checks best practices. Fix errors and most warnings.
- `build(pkg=".")` creates a “.tar.gz” file from the directory `pkg` that can be shared with other R users. (Note: “.tar” indicates a “tape archive” file from which the directory of files can be extracted, and “.gz” indicates a file compressed with the `gzip` program.) e.g. `build(pkg="jgUtilities")`
- `install.packages(pkgs, repos=getOption("repos"), type=getOption("pkgType"))` installs the package on to your hard drive; using `repos=NULL` and `type="source"` allows the use of your “.tar.gz” file for `pkgs`. e.g.
 

```
install.packages(pkgs="jgUtilities_0.1.tar.gz", repos=NULL, type="source")
```
- `require(package)` loads `package` into the R session. Note: Restart R before `require()`. e.g. “Session > Restart R”, then `require("jgUtilities")`

For more, see

- the “R packages” online book by Hadley Wickham at <http://r-pkgs.had.co.nz>
- the “PACKAGE DEVELOPMENT” section of <https://support.rstudio.com/hc/en-us>
- <http://cran.r-project.org/web/packages/roxygen2/vignettes/formatting.html>