# Chapter 7

# Selection of terms in a model

Often we are faced with a large number of potential terms in the model based upon several covariates. (Recall that one or more covariates may generate a large number of terms through interactions, powers of numeric terms, etc.)

We wish to determine a simple, adequate model. If we include too few terms our model will be inadequate and we will introduce bias into the results. Including too many terms will introduce excess variability in the parameter estimates and the predictions from the model.

In the past, model selection procedures were considered as applying to individual columns of the model matrix for the "largest" model being considered. That often resulted in nonsensical results, either because individual columns associated with a categorical term were deleted while others were retained, or because the terms left in the model did not respect the hierarchy of terms. As stated in the documentation for the `drop1` function in the `stats` package,

> The hierarchy is respected when considering terms to be added or dropped: all main effects contained in a second-order interaction must remain, and so on.

**R Exercise:** Consider the `timetemp` data from the `EngrExpt` package, shown in Fig. 7.1 Potential models for these data, assuming that the within-type relationship between `time` and `temp` is more-or-less linear, which seems to be a reasonable assumption, are:

```
> lm1 <- lm(time ~ temp, timetemp)
> lm1a <- lm(time ~ type + temp, timetemp)
> lm1b <- lm(time ~ type + temp + type:temp, timetemp)
```

for which the summaries of the coefficients are

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -38.995      5.588   -6.98  5.3e-07
temp          -1.858      0.217   -8.57  1.8e-08
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.6872     2.4432  -13.79  5.4e-12
typeOEM      -3.7166     0.3721   -9.99  2.0e-09
temp         -1.7178     0.0936  -18.35  2.1e-14
```

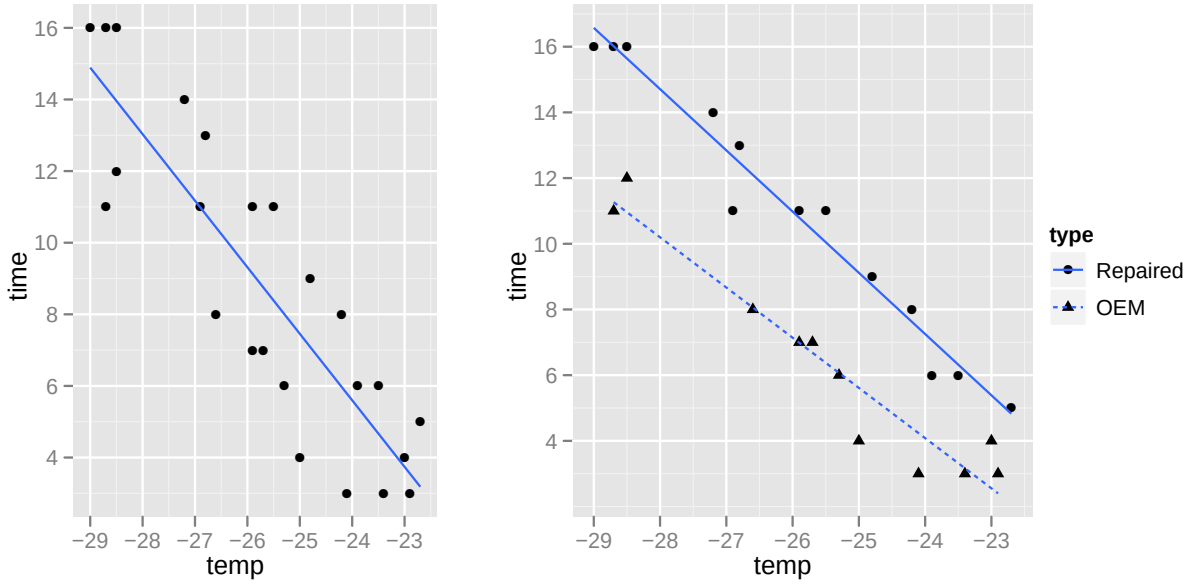


Figure 7.1: Time for panels to reach the temperature -10 C. according to the freezer temperature. There are two types of panels: Repaired and OEM (Original Equipment Manufacture). The Repaired panels have extra coats of pain.

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | -37.490 | 3.065 | -12.23 | 9.7e-11 |
| typeOEM | 4.891 | 4.582 | 1.07 | 0.298 |
| temp | -1.864 | 0.118 | -15.84 | 8.8e-13 |
| typeOEM:temp | 0.336 | 0.178 | 1.88 | 0.074 |

The first model, `lm1`, which does not incorporate the `type` variable, corresponds to the fit on the left panel. The third model, `lm1b`, corresponds to the two lines in the right-hand plot. Model `lm1a` is an intermediate model. It corresponds to parallel lines, one for the `Repaired` panels and one for the `OEM` panels.

If we know that the panels being tested are of two types, as shown in the right-hand panel, then the nature of the bias from having too few terms in the model is obvious; the time for Repaired panels is being underpredicted and the time for OEM panels is over-predicted. The "canned" residual plots, Fig. 7.2 show this to some extent because there are clearly two groups of residuals, one centered around +2 on the scale of the raw residuals and one centered around -2.

The two groups of residuals are even more obvious if we consider these residuals and how they are related to `type`. One possibility is to consider a comparative empirical density plot by type or a normal QQ plot by type, Fig. 7.3.

Inflation of the variability in the parameter estimates when terms are added is clear from the change in the standard error for the `typeOEM` coefficient from model `lm1a` to `lm1b`. The interpretation of this coefficient also changes: in model `lm1a` it is the vertical distance between two parallel lines

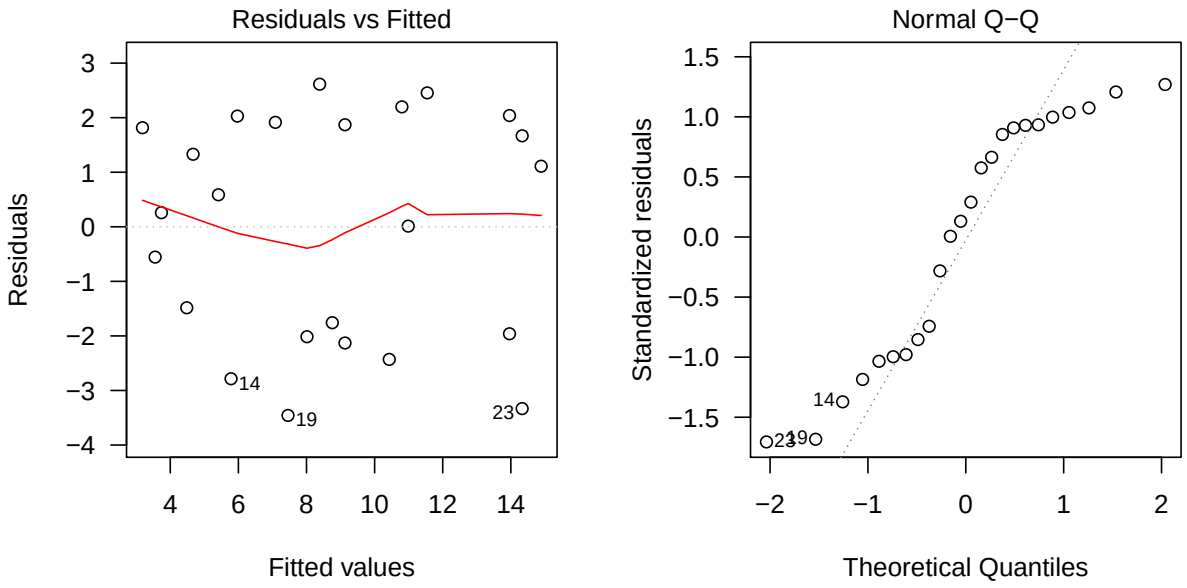


Figure 7.2: Residuals versus fitted values and a normal Q-Q plot for the residuals from model `lm1`.

whereas in model `lm1b` it is the change in the intercept of the two lines. That is, in model `lm1b` it is the vertical deviation at a temperature of $O^o$ C. only.

Using either the t-statistics in the coefficients tables or the comparative analysis of variance output

```
> anova(lm1, lm1a)

Analysis of Variance Table
Model 1: time ~ temp
Model 2: time ~ type + temp
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1     22 97.4
2     21 16.9  1      80.4 99.8 2.0e-09

> anova(lm1a, lm1b)

Analysis of Variance Table
Model 1: time ~ type + temp
Model 2: time ~ type + temp + type:temp
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     21 16.9
2     20 14.4  1      2.55 3.55  0.074
```

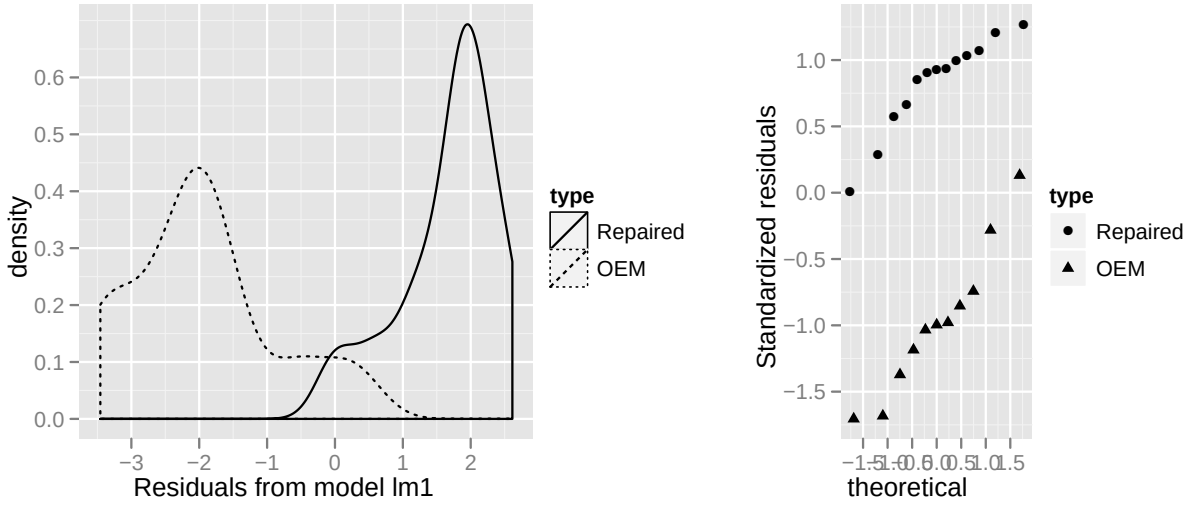


Figure 7.3: Comparative empirical density plot and normal Q-Q plot of the residuals from model `lm1b` by panel type.

we would conclude that model `lm1a` is a significantly better fit than `lm1`, which is obvious from the plots, and that the decision between `lm1a` and `lm1b` is in the gray area with a p-value for the comparison of 7.4%. If we adopt the standard of 5% or less for a significant term then we would accept the simpler model, `lm1a`.

When we check model `lm1b` with `drop1` to see if any of the terms could be dropped, it examines only the interaction term, because of the hierarchy principle. If the interaction term is retained then both the main effects terms should be retained.

```
> drop1(lm1b)

Single term deletions
Model:
time ~ type + temp + type:temp
          Df Sum of Sq  RSS   AIC
<none>                  14.4 -4.29
type:temp  1      2.55 16.9 -2.37
```

In this comparison the model with the interaction term produces a lower value of Akaike's Information Criterion (AIC), indicating that we should retain the interaction, corresponding to distinct non-parallel lines. Because of the hierarchy, the terms `type` and `temp` are retained when `type:temp` is retained.

We can obtain the value of the F test (relative to the original model) in addition to the AIC values as

```
> drop1(lm1b, test="F")
```

```
Single term deletions
Model:
time ~ type + temp + type:temp
          Df Sum of Sq  RSS   AIC F value Pr(F)
<none>                  14.4 -4.29
type:temp  1      2.55 16.9 -2.37    3.55 0.074
```

which shows apparently inconsistent results between the AIC comparison, which favors the model with the interaction, and the F test, which favors the model without.

This is not uncommon. Comparisons based on AIC or Schwartz's Bayesian Information criterion (BIC or, sometimes, SBC) often end up contradicting each other. This is why model selection criteria should be considered as guidance, not absolute.

To obtain a comparison using BIC with `drop1` we use the optional argument `k=log(n)` where `n` is the number of observations — 24 in this case. One way to count the number of observations used to fit the model (i.e. after possible eliminations of rows due to missing data) is as the number of rows in the model frame. A slightly safer way (which would take into account cases with a weight of zero) is show in

```
> drop1(lm1b, test="F", k=log(df.residual(lm1b) + lm1b$rank))

Single term deletions
Model:
time ~ type + temp + type:temp
          Df Sum of Sq  RSS   AIC F value Pr(F)
<none>                  14.4 0.419
type:temp  1      2.55 16.9 1.162    3.55 0.074
```

Even though the column is still labelled as AIC, it is BIC that is being calculated.

## 7.1 General model selection problem

The general model selection problem is often phrased in terms of selecting a subset of the columns $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ of a model matrix $\boldsymbol{X}$ to form a simple adequate model. However, this confuses the columns of $\boldsymbol{X}$ with terms in the model and those two are not always interchangeable — because a single term can correspond to more than one column. Also, this treats all terms as being equal, which they are not according to the hierarchy of terms.

Assuming that we had a multiple linear regression model in which there is a one-to-one correspondence between terms and columns we could examine all possible subsets but that would entail checking $2^p$ potential models (or $2^{p-1}$ if we assume that the intercept is always included). It would quickly become unmanageable to try to create F or t-tests to compare potential models. Instead we use a mechanism or algorithm to enumerate the potential models and usually derive a criterion to choose between them (always taking into account that this is only advisory, not prescriptive).

Some of the criteria commonly used are:

- Mallow's Cp

- AIC/BIC
- Cross Validation

**R Exercise** We have mentioned and used AIC and BIC in conjunction with the `drop1` function. You can also use Mallow's Cp statistic with the `drop1` or `step` functions. To show the more general usage of the `step` function we examine another data set, `swiss`, which provides certain measures on the cantons in Switzerland, including a fertility measure. (See `?swiss` for details.)

```
> str(swiss)

'data.frame':        47 obs. of  6 variables:
 $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
 $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
 $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
 $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
 $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
 $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...

> printCoefmat(coef(summary(lm2 <- lm(Fertility ~ ., swiss))))

                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       66.9152    10.7060    6.25  1.9e-07
Agriculture       -0.1721     0.0703   -2.45   0.0187
Examination       -0.2580     0.2539   -1.02   0.3155
Education         -0.8709     0.1830   -4.76  2.4e-05
Catholic           0.1041     0.0353    2.95   0.0052
Infant.Mortality   1.0770     0.3817    2.82   0.0073

> printCoefmat(coef(summary(mod <- step(lm2))))

Start:  AIC=191
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                  Df Sum of Sq  RSS AIC
- Examination       1        53 2158 190
<none>                         2105 191
- Agriculture       1       308 2413 195
- Infant.Mortality  1       409 2514 197
- Catholic          1       448 2553 198
- Education         1      1163 3268 209

Step:  AIC=190
Fertility ~ Agriculture + Education + Catholic + Infant.Mortality

                  Df Sum of Sq  RSS AIC
<none>                         2158 190
- Agriculture       1       264 2422 193
```

```
- Infant.Mortality  1       410 2568 196
- Catholic          1       957 3115 205
- Education         1      2250 4408 221
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       62.1013     9.6049    6.47  8.5e-08
Agriculture       -0.1546     0.0682   -2.27   0.0286
Education         -0.9803     0.1481   -6.62  5.1e-08
Catholic           0.1247     0.0289    4.31  9.5e-05
Infant.Mortality   1.0784     0.3819    2.82   0.0072

> drop1(lm2, scale=deviance(lm2)/df.residual(lm2))

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality

scale:  51.3

                Df Sum of Sq  RSS    Cp
<none>                       2105  6.00
Agriculture      1       308 2413  9.99
Examination      1        53 2158  5.03
Education        1      1163 3268 26.64
Catholic         1       448 2553 12.72
Infant.Mortality 1       409 2514 11.96
```

From the coefficient matrix for model `lm2` we see that the `Examination` term is not at all significant. The test for the reduced model without this term versus the full model has a p-value of about 32% and we prefer the simpler model. Applying `drop1` without other arguments

```
> drop1(lm2)

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                Df Sum of Sq  RSS AIC
<none>                       2105 191
Agriculture      1       308 2413 195
Examination      1        53 2158 190
Education        1      1163 3268 209
Catholic         1       448 2553 198
Infant.Mortality 1       409 2514 197
```

produces a table based on the AIC values. We need to look at the AIC column to see which model produces the lowest AIC value. In this case it is the current model minus the `Examination` term. It is easier to see this if we order the rows by increasing AIC

```
> d1 <- drop1(lm2)
> d1[order(d1[["AIC"]]), ]

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                Df Sum of Sq  RSS AIC
Examination      1        53 2158 190
<none>                     2105 191
Agriculture      1       308 2413 195
Infant.Mortality 1       409 2514 197
Catholic         1       448 2553 198
Education        1      1163 3268 209
```

As before, we can include the F tests

```
> d1 <- drop1(lm2, test="F")
> d1[order(d1[["AIC"]]),]

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                Df Sum of Sq  RSS AIC F value   Pr(F)
Examination      1        53 2158 190    1.03  0.3155
<none>                     2105 191
Agriculture      1       308 2413 195    5.99  0.0187
Infant.Mortality 1       409 2514 197    7.96  0.0073
Catholic         1       448 2553 198    8.72  0.0052
Education        1      1163 3268 209   22.64 2.4e-05
```

As all the sub-models correspond to single term deletions and all the potential terms for deletion involve one coefficient only, ordering by increasing AIC is the same as ordering by increasing F value. We usually order by increasing AIC because AIC is defined even for the model with no deletions.

The same approach is followed for BIC

```
> d1 <- drop1(lm2, test="F", k=log(df.residual(lm2)+lm2$rank))
> d1[order(d1[["AIC"]]),]

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                Df Sum of Sq  RSS AIC F value   Pr(F)
Examination      1        53 2158 199    1.03  0.3155
<none>                     2105 202
```

```
Agriculture       1       308 2413 204    5.99  0.0187
Infant.Mortality  1       409 2514 206    7.96  0.0073
Catholic          1       448 2553 207    8.72  0.0052
Education         1      1163 3268 219   22.64 2.4e-05
```

To use Mallow's Cp criterion (derived below) we specify a `scale` argument, which is an estimate of $\sigma^2$ to which all the models could be compared. Typically this will be the residual mean square for the most complex model

```
> drop1(lm2, scale=deviance(lm2)/df.residual(lm2))

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality

scale:  51.3

                 Df Sum of Sq  RSS    Cp
<none>                        2105  6.00
Agriculture       1       308 2413  9.99
Examination       1        53 2158  5.03
Education         1      1163 3268 26.64
Catholic          1       448 2553 12.72
Infant.Mortality  1       409 2514 11.96
```

and, again, it helps to order these by increasing value of Cp

```
> d1 <- drop1(lm2, scale=deviance(lm2)/df.residual(lm2))
> d1[order(d1[["Cp"]]), ]

Single term deletions
Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality

scale:  51.3

                 Df Sum of Sq  RSS    Cp
Examination       1        53 2158  5.03
<none>                        2105  6.00
Agriculture       1       308 2413  9.99
Infant.Mortality  1       409 2514 11.96
Catholic          1       448 2553 12.72
Education         1      1163 3268 26.64
```

However, we must be careful in interpreting these results. We evaluate AIC and BIC according to a "smaller is better" rule. This does not apply to Mallows' Cp. Desirable values of Cp are those

close to or less than the number of coefficients in the model. The definition is such that the Cp will be exactly the number of coefficients in the model from which the `scale` parameter is derived. In this case there are 6 coefficients in the original model and its value of Cp is exactly 6. The Cp values for the other models should be compared to 5, which indicates that the model with `Examination` removed should be considered but not the others.

So bear in mind that the model producing the lowest value of Cp is not chosen automatically.

## 7.2 Underfitting / Overfitting

The term *underfitting* means that we exclude covariates that should be included whereas *overfitting* refers to using additional covariates that should be excluded.

If the "true" model is $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$ and we derive an estimate $\tilde{\boldsymbol{\beta}}$ where $E[\tilde{\boldsymbol{\beta}}] = \boldsymbol{\beta}*$ and not $\boldsymbol{\beta}$, then

$$\begin{aligned} E[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta})^2] &= E[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^* + \boldsymbol{\beta}^* - \boldsymbol{\beta})^2] \\ &= E[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^2] + E[(\boldsymbol{\beta}^* - \boldsymbol{\beta})^2] + 2E[(\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)(\boldsymbol{\beta}^* - \boldsymbol{\beta})] \\ &= \mathrm{Var}(\tilde{\beta}) + \mathrm{Bias}^2 + 0 \end{aligned}$$

So the *mean squared error* (MSE) is composed of the variance and the bias and sometimes we trade them off against each other. If we write the "true" mean as

$$\boldsymbol{X}\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{bmatrix}$$

then fitting $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}_1\boldsymbol{\beta}_1^*, \sigma^2\boldsymbol{I})$ would be underfitting while $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}_1\boldsymbol{\beta}_1^* + \boldsymbol{X}_2\boldsymbol{\beta}_2^* + \boldsymbol{X}_3\boldsymbol{\beta}_3^*, \sigma^2\boldsymbol{I})$ results in overfitting.

The underfitting case is the same as setting $\boldsymbol{\beta}_2 = \mathbf{0}$ and $\widehat{\boldsymbol{\beta}^*}_1 \neq \widehat{\boldsymbol{\beta}}_1$ unless $\boldsymbol{X}_1 \perp \boldsymbol{X}_2$, in which case $\tilde{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\beta}^*}_1, \mathbf{0})$.

**Theorem 12.**

$$\widehat{\boldsymbol{\beta}} = (X'X)^{-1}X'Y \textit{ full model}$$

$\widehat{\boldsymbol{\beta}} = \begin{pmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{pmatrix}$ *from* $Y = X\beta + \epsilon$ $\qquad$ $\widehat{\boldsymbol{\beta}}_1^* = (X_1'X_1)^{-1}X_1'Y$ *estimated from reduced model fit* $Y = X_1\beta_1^* + \epsilon$

1. $E[\widehat{\boldsymbol{\beta}}_1^*] = \beta_1 + A\beta_2$ *where* $A = (X_1'X_1)^{-1}X_1'X_2 \Rightarrow$ *estimator is biased (unbiased if* $X_2$ *is orthogonal to* $X_1$*)*
2. $Cov(\widehat{\boldsymbol{\beta}}_1^*) = \sigma(X_1'X_1)^{-1}$
3. $Cov(\widehat{\boldsymbol{\beta}}_1) - Cov(\widehat{\boldsymbol{\beta}}_1^*) = AB^{-1}A'$ *where* $A = (X_1'X_1)^{-1}X_1'X_2$ *and* $B = X_2'X_2 - X_2'X_1A \Rightarrow Var(\widehat{\boldsymbol{\beta}}_j) > Var(\widehat{\boldsymbol{\beta}}_j^*)$
4. $Var(X_{01}'\widehat{\boldsymbol{\beta}}_1) - Var(X_{01}'\widehat{\boldsymbol{\beta}}_1^*) \geq 0$ *where* $X_{01}$ *is a p x 1 matrix*
5. $Var(X_0'\widehat{\boldsymbol{\beta}}) - Var(X_{01}'\widehat{\boldsymbol{\beta}}_1^*) \geq 0 \ \forall \ X_0$ *where* $Var(X_{01}'\widehat{\boldsymbol{\beta}}_1^*)$ *is the fitted value from the reduced model*

## 7.3 Risk

Want to fit $E[(X_1^*\widehat{\boldsymbol{\beta}}_1^* - X\beta)^2]$. $\mu = X\beta$ and $\hat{\mu} = X_1\beta_1^*$ ($X_1$ is an n x k matrix).

$$E[(X_1\beta_1^* - X\beta)'(X_1\widehat{\boldsymbol{\beta}}_1^* - X\beta)] = (*) = \sigma^2 k + \beta' X'(I - P_1)X\beta = R(k)$$

$$P_1 = X_1(X_1'X_1)^{-1}X_1$$

As k increases, $\beta' X'(I - P_1)X\beta$ decreases. We don't know $\sigma^2$ and true $\beta$. If we did, choose k with minimum risk. Assume $\begin{bmatrix} \uparrow & \uparrow \\ X_1 & X_p \\ \downarrow & \downarrow \end{bmatrix}$

Need an estimator of $(*)$

### 7.3.1 Estimating Risk

$$\begin{aligned}
RSS(k) &= ||Y - X_1\widehat{\boldsymbol{\beta}}_1^*||^2 \\
&= ||Y - P_1Y||^2 \\
&= Y'(I - P_1)Y \\
E[RSS(k)] &= E[Y'(I - P_1)Y] \\
&= E[tr(Y'(I - P_1)Y)] \\
&= E[tr(I - P_1)YY'] \\
&= tr[(I - P_1)E[YY']] \\
&= tr[(I - P_1)(\sigma^2 I + X\beta\beta' X')] \\
&= \sigma^2(n - k) + \beta' X'(I - P_1)X\beta
\end{aligned}$$

$$\hat{R}(k) = RSS(k) - (n - 2k)\sigma^2$$

$$\begin{aligned}
E(\hat{R}(k)) &= E(RSS(k)) - (n - 2k)\sigma^2 \\
&= (n - k)\sigma^2 + \beta' X'(I - P_1)X\beta - (n - 2k)\sigma^2 \\
&= \sigma^2 k + \beta' X'(I - P_1)X\beta
\end{aligned}$$

## 7.4 Mallow's Cp

$$Cp(k) = \frac{RSS(k)}{s^2} + 2k - n$$

$$E[Cp(k)] = E[(n - k) + \frac{\beta' X(I - P_1)X\beta}{\sigma^2} + 2k - n]$$

Should be roughly equal to $k$. In practice this is evaluated for the model with $k$ coefficients that provides the lowest residual sum of squares. Limitation: we need to decide what estimate, $s^2$ of $\sigma^2$ to use. Typically this is $s^2$ from the full model?

## 7.5 AIC - Akaike's Information Criterion

$f(\boldsymbol{y})$ is the true density of data $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$ and $g(\boldsymbol{y})$ is the density corresponding to the model of interest (candidate model) $\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}_1\boldsymbol{\beta}_1, \sigma^2\boldsymbol{I})$, evaluated at the estimate, $\widehat{\boldsymbol{\beta}^*}_1$. The Kullback-Leibler divergence is defined as

$$\mathrm{KL}(f, g) = \int_{\mathbb{R}^n} \left[\log \frac{f(y)}{g(y)}\right] f(\boldsymbol{y})\, d\boldsymbol{y} = E_f \left[\log \frac{f(y)}{g(y)}\right]$$

Two properties of the Kullback-Leibler divergence are

$$\mathrm{KL}(f, g) \leq \mathrm{KL}(f, f) = 0 \quad \text{and} \quad \mathrm{KL}(f, g) \neq \mathrm{KL}(g, f)$$

As described in Chap. 1, the log-likelihood, $\ell(\boldsymbol{\beta}, \sigma|\boldsymbol{y})$, of a linear model is

$$\ell(\boldsymbol{\beta}, \sigma|\boldsymbol{y}) = \log\left(L(\boldsymbol{\beta}, \sigma|\boldsymbol{y})\right) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2\sigma^2}$$

Once we have fit the model we evaluate this quantity at the parameter estimates. Typically we use the maximum likelihood estimate, $\widehat{\sigma}_{\mathrm{ML}} = S(\widehat{\beta})/n$, where $S(\widehat{\beta})$ is the sum of squared residuals at the parameter estimates providing

$$\ell(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}_{\mathrm{ML}}|\boldsymbol{y}) = \ell(\widehat{\boldsymbol{\beta}}|\boldsymbol{y}) = -\frac{n}{2}\left[\log\left(1 + \frac{2\pi S(\widehat{\boldsymbol{\beta}})}{n}\right)\right]$$

The AIC criterion is defined as

$$\mathrm{AIC} = -2\ell(\widehat{\boldsymbol{\beta}}|\boldsymbol{y}) + 2r$$

where $r$ is the dimension of the parameter vector. Most definitions include $\sigma^2$ as one of the parameters so, in a model with $p$ coefficients, $r = p + 1$. The BIC criterion is defined as

$$\mathrm{BIC} = -2\ell(\widehat{\boldsymbol{\beta}}|\boldsymbol{y}) + 2\log(n)\, r$$

Because these criteria are derived from the deviance, which is negative twice the log-likelihood, smaller is better. The deviance measures the fidelity of the model to the observed data and the term $2r$, for AIC, or $2\log(n)\, r$, for BIC, is a penalty on the number of parameters required to achieve this fidelity.

AIC is a simple criterion. It comes down to preferring a model with an additional parameter if the deviance can be reduced by 2 (or, equivalently, the log-likelihood can be increased by 1). BIC is a bit more subtle in that an additional parameter must increase the log-likelihood by $\log(n)$ or more to be judged effective.

**R Exercise** We have seen the use of AIC and BIC as model selection criteria in earlier sections. The values quoted for AIC and BIC in the `drop1` function are not exactly the same as the definitions given above but the differences are. In the `drop1` function AIC is calculated as

```
> AIC <- n * log(RSS/n) + 2 * p
```

where `p` is the number of coefficients. For the full model in

```
> drop1(lm1b)

Single term deletions
Model:
time ~ type + temp + type:temp
          Df Sum of Sq  RSS   AIC
<none>                 14.4 -4.29
type:temp  1      2.55 16.9 -2.37
```

we have $n = 24$ and $p = 4$ providing

```
> 24 * log(deviance(lm1b)/24) + 2 * 4

[1] -4.29
```

and for the reduced model it is

```
> 24 * log(deviance(lm1a)/24) + 2 * 3

[1] -2.37
```

(Recall that the `deviance` function applied to a linear model returns the residual sum of squares and not the quantity we are calling the deviance - a regretable confusion.)

If you wish to use the original definition instead (which is probably a good idea if you are reporting a model fit and want to compare the AIC to values from other software), you could use

```
> logLik(lm1a); logLik(lm1b)

'log Lik.' -29.9 (df=4)

'log Lik.' -27.9 (df=5)
```

providing

```
> AIC(lm1a, lm1b)

     df  AIC
lm1a  4 67.7
lm1b  5 65.8
```

These correspond to the formulas given above

```
> all.equal(unclass(logLik(lm1a)),
+           -12*(1 + log(2*pi*deviance(lm1a)/24)), check.attr=FALSE)

[1] TRUE

> all.equal(AIC(lm1a), -2*unclass(logLik(lm1a)) + 2*4, check.attr=FALSE)

[1] TRUE
```

as does the function `BIC` in the `stats4` package.

```
> all.equal(stats4::BIC(lm1a),  -2*unclass(logLik(lm1a))+log(24)*4, check.attr=FALSE)

[1] TRUE
```

This use of non-standard definitions would be troublesome except for the fact that the AIC and BIC values are not of interest by themselves. It is only the differences between models for these criteria that are important and those are consistent

```
> diff(drop1(lm1b)$AIC)

[1] 1.92

> diff(AIC(lm1b, lm1a)$AIC)

[1] 1.92

> diff(drop1(lm1b, k=log(24))$AIC)

[1] 0.744

> diff(stats4::BIC(lm1b, lm1a)$BIC)

[1] 0.744
```

## 7.6 Forward Selection and Backward Deletion

To this point we have only used `drop1` to perform what is called backward selection on a model. We begin with the largest model we want to entertain then drop terms as appropriate. An alternative is forward selection implemented in `add1`. A combination of forward and backward selection is called *stepwise* selection and is implemented in the `step` function. Very occasionally a term that is already in the model will no longer be significant when another term is added or a term that was deleted can be added back in when another term is deleted, which is why the `step` function will consider changes in both directions.

For `add1` we must specify a `scope` argument to indicate the largest model to be entertained and we can do the same for `step`

```
> add1(lm1a, ~ type + temp + type:temp)

Single term additions
Model:
time ~ type + temp
          Df Sum of Sq  RSS   AIC
<none>                 16.9 -2.37
type:temp  1      2.55 14.4 -4.29

> summary(step(lm(Fertility ~ 1, swiss),
+              ~ Agriculture + Examination + Education +
+              Catholic + Infant.Mortality))

Start:  AIC=238
Fertility ~ 1
                  Df Sum of Sq  RSS AIC
+ Education         1      3163 4015 213
+ Examination       1      2994 4184 215
+ Catholic          1      1543 5635 229
+ Infant.Mortality  1      1246 5932 231
+ Agriculture       1       895 6283 234
<none>                         7178 238

Step:  AIC=213
Fertility ~ Education

                  Df Sum of Sq  RSS AIC
+ Catholic          1       961 3054 202
+ Infant.Mortality  1       891 3124 203
+ Examination       1       466 3550 209
<none>                         4015 213
+ Agriculture       1        62 3953 214
- Education         1      3163 7178 238

Step:  AIC=202
Fertility ~ Education + Catholic

                  Df Sum of Sq  RSS AIC
+ Infant.Mortality  1       632 2422 193
+ Agriculture       1       486 2568 196
<none>                         3054 202
+ Examination       1         2 3052 204
- Catholic          1       961 4015 213
- Education         1      2581 5635 229

Step:  AIC=193
Fertility ~ Education + Catholic + Infant.Mortality
```

```
                   Df Sum of Sq  RSS AIC
+ Agriculture       1       264 2158 190
<none>                          2422 193
+ Examination       1         9 2413 195
- Infant.Mortality  1       632 3054 202
- Catholic          1       702 3124 203
- Education         1      2380 4803 224

Step:  AIC=190
Fertility ~ Education + Catholic + Infant.Mortality + Agriculture

                   Df Sum of Sq  RSS AIC
<none>                          2158 190
+ Examination       1        53 2105 191
- Agriculture       1       264 2422 193
- Infant.Mortality  1       410 2568 196
- Catholic          1       957 3115 205
- Education         1      2250 4408 221

Call:
lm(formula = Fertility ~ Education + Catholic + Infant.Mortality +
    Agriculture, data = swiss)

Residuals:
    Min      1Q  Median      3Q     Max
-14.676  -6.052   0.751   3.166  16.142

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       62.1013     9.6049    6.47  8.5e-08
Education         -0.9803     0.1481   -6.62  5.1e-08
Catholic           0.1247     0.0289    4.31  9.5e-05
Infant.Mortality   1.0784     0.3819    2.82   0.0072
Agriculture       -0.1546     0.0682   -2.27   0.0286

Residual standard error: 7.17 on 42 degrees of freedom
Multiple R-squared: 0.699,        Adjusted R-squared: 0.671
F-statistic: 24.4 on 4 and 42 DF,  p-value: 1.72e-10
```

Careful examination of the output from `step` shows that, at some steps, terms were considered for inclusion (prefaced by `+`) and for deletion (prefaced by `-`).

The value of the `step` function is the linear model fit that is chosen as the best, according to the criteria.

We have shown the use of the `step`, `drop1` and `add1` functions in the `stats` package. There are several other functions in other packages, notably the `mle.stepwise` function in the `wle` package and the `leaps` and `regsubsets` functions in the `leaps` package, that perform stepwise procedures but these are typically based on old Fortran code that considers the columns of the model matrix

as unrelated to each other and, thus, can produce nonsensical models.

## 7.7 Cross-Validation

Cross validation is a general method for variable selection. It is not constrained to linear models. Motivation for cross-validation:

1. Can find best possible scenario for model
2. Useful if there is other data of the same nature
3. Use model to predict values for other data

Let n = sample size

$$Y^{train} = \widehat{\boldsymbol{\beta}}_0^{train} + \widehat{\boldsymbol{\beta}}_1^{train} X_1 \qquad RSS = \frac{1}{n}\sum_{i=1}^{n}(Y_i^{train} - \widehat{\boldsymbol{\beta}}_0^{train} - \widehat{\boldsymbol{\beta}}_1^{train} X_1^{test})^2$$

Using residual sum of squares for test data $\Rightarrow$ prediction error, a.k.a. test set error

### 7.7.1 Theory Behind Cross-Validation

Data: $n$ iid observations $X_i = (Y_i, W_{i1}, \ldots, W_{ip})$. $X_1, \ldots, X_n$ learning set data used to learn population parameters. $P_0$ is the true data generating distribution. $X_i \sim P_0$

Model: $Y = W\beta + \epsilon \quad E(\epsilon|W) = 0$

Parameter of interest:

Denote parameters of interest $\mu_0 = \mu(W) = E_{P_0}(Y|W)$

Loss functions - quantify error in prediction. $L : (X, \mu) \to L(X, \mu) \in \mathbb{R}$. $L(y, \hat{y})$ elaborates loss incurred when predicting $y$ by $\hat{y}$

Squared error loss function $L(y, \hat{y}) = (y - \hat{y})^2$

Risk functions - for given loss function $L(X, \mu)$ with $\mu \in \psi$ ($\psi$ is parameter space $\mathbb{R}^p$ for $\beta$ with $p$ explanatory variables). $R(\mu, P_0) = E_{P_0}[L(X, \mu)] = \int L(X, \mu) dP_0(X) = \int L(X, \mu) f(x) dx$. When $P_0$ known, $\mu_{opt} = argmin_{\mu \in \psi} R(\mu, P_0)$

$P_n$ - empirical distribution of data $(X_1, \ldots, X_n)$ each data point gets mass $\frac{1}{n}$

**Definition 7.** *$\hat{\mu}$ estimator is mapping from empirical distributions to parameter space $\psi$*
*$\mu_n = W\widehat{\boldsymbol{\beta}}_{LS}$ full model*
*$\mu_n = W_1\widehat{\boldsymbol{\beta}}_{1,LS}$ sub model*

True unknown risk: $E_\beta[L(X, \mu_n)] = \int L(X, \mu_n) dP_0(x)$ where $L(X, \mu_n) = (Y - \mu_n(w))^2$

Loss reduces to RSS: $E_{P_n}[L(X, \mu_n)] = \frac{1}{n}\sum_{i=1}^{n}(Y_i - W_i\widehat{\boldsymbol{\beta}})^2$ (estimator of risk, but overfits)

$\mu_0(w)$ like true $X\beta$ and $X_n(w)$ like $X\widehat{\boldsymbol{\beta}}$

As $n_{TS} \to \infty$, empirical distribution converges to the true distribution.

Can reserve 1/3 of the dataset as test and 2/3 as training, but we don't usually have luxury to set aside part of the data set

### 7.7.2 Utilizing Cross Validation

$$E_{\beta_n} \int \overbrace{L(X, \underbrace{\hat{\mu}(P^0_{n,B_n})}_{\text{training}}) d \underbrace{P^1_{n,B_n}(x)}_{\text{validation}}}^{\text{Risk}}$$

where $L(X, \hat{\mu}(P^0_{n,B_n}))dP^1_{n,B_n}(x)$ is the risk, $\hat{\mu}(P^0_{n,B_n})$ is from training, $dP^1_{n,B_n}(x)$ is from validation.

Calculate $E_{B_n}$ by $\frac{1}{n_1} \sum_{i|B_n(i)=1} L(X_i, \hat{\mu}(P^0_{n,B_n}))$

#### Leave One Out Cross Validation(LOOCV)

n-1 training set 1 point for testing set

Compute statistic for each data point

#### V-fold Cross-Validation

5 or 10 fold used in practice

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Example: | $x_3$ | $x_2$ | $x_1$ | $x_4$ | $x_7$ |
| | $x_5$ | $x_8$ | $x_6$ | $x_9$ | $x_{10}$ |

n data points, v parts (Example - 5 fold cross validation with 10 data points)

1. Randomly submit 10 observations into 5 mutually exclusive and exhaustive sets of size $\frac{n}{v}$
2. Cycle 1: Box 1 validation set, boxes 2-5 learning set
3. Construct estimators based on training set $x_1, x_2, x_4, x_6, x_7, x_8, x_9, x_{10}$
   $\mu_{n,1} = \widehat{\boldsymbol{\beta}}_0 + \widehat{\boldsymbol{\beta}}_1 W_5$
   $\mu_{n,2} = \widehat{\boldsymbol{\beta}}_0 + \widehat{\boldsymbol{\beta}}_1 W_6 + \widehat{\boldsymbol{\beta}}_2 W_{10}$
   $\mu_{n,3} = \widehat{\boldsymbol{\beta}}_0 + \widehat{\boldsymbol{\beta}}_1 W_3 + \ldots$
   (Forward, backward, stepwise regression)
4. Compute validation set error
   $CV_1(\mu_{n,1}) = \frac{1}{2}[(Y_3 - \mu_{n_1}(X_3))^2] + [(Y_5 - \mu_{n_1}(X_5))^2]$
   $CV_1(\mu_{n,2}) = \vdots$
5. Cycle 2: Exclude box 2 and repeat using the same training models as previously selected $(\mu_{n,1}, \mu_{n,2}, etc.)$

6. *Final* CV error

$$CV(\mu_{n,1}) = \frac{1}{5}\sum_{i=1}^{5} CV_i(\mu_{n,1})$$

$$CV(\mu_{n,2}) = \frac{1}{5}\sum_{i=1}^{5} CV_i(\mu_{n,2})$$

$$\vdots$$

7. Choose model with minimum CV error

Note: run forward selection/backward deletion on *all* data yields candidate models
Run candidate models for each box. Choose model size based on cross-validation
V-fold cross-validation is implemented in the `CVlm` function from the `DAAG` package.

## 7.8 Notes about Variable Selection

### 7.8.1 Role of the Intercept

$\frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon}_i = 0 \Leftrightarrow \widehat{\boldsymbol{\beta}}_0 = \bar{y} - \widehat{\boldsymbol{\beta}}_1\bar{X}_1 - \cdots - \widehat{\boldsymbol{\beta}}_p\bar{X}_p$ (Only if there is an intercept)

### 7.8.2 Bias-Variance Tradeoff

$$\begin{aligned} MSE &= E(\widehat{\boldsymbol{\beta}} - \beta)^2 \\ &= E(\widehat{\boldsymbol{\beta}} - E(\widehat{\boldsymbol{\beta}}))^2 + (E(\widehat{\boldsymbol{\beta}}) - \beta)^2 \\ &= \text{Variance} + Bias^2 \end{aligned}$$

$$Var(\widehat{\boldsymbol{\beta}}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(x_{i1} - \bar{x}_1)^2}\frac{1}{1 - r_{12}^2}$$

Goal - achieve balance between bias and variance

### 7.8.3 Singular Design Matrices

In some cases, $(X^TX)^{-1}$ is almost singular

1. if $n < p$
2. if some variables are highly correlated (Solution: don't use all variables - choose using variable selection)

$(X^TX + \lambda I)^{-1}$ Removes singularity

### 7.8.4 Reasons for Variable Selection

1. Too many variables
2. Deal with collinearity (predictors are highly correlated)
3. Reasonably smaller size of predictors can simplify results

# Chapter 8

# Dealing with Multicollinearity

### 8.0.5 Dealing with Multicollinearity with large number of variables

1. Subset selection / methods for model construction
   Selection among models - Cp, AIC/BIC, CV

2. Shrinkage methods
   Ridge regression - uses all the covariates but imposes constraints on them
   Lasso

3. Derived input directions - Principle components regression (PCR), partial least squares (PLS)
   Use all the variables but form linear combinations (meta - predictors)

## 8.1 Ridge Regression

$$\widehat{\boldsymbol{\beta}}^{ridge} = \arg\min \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 \text{ such that } \sum_{j=1}^{p}\beta_j^2 \leq s$$

where $s$ is user specified

$$\Rightarrow \widehat{\boldsymbol{\beta}}^{ridge} = argmin \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p}\beta_j^2$$

with $\lambda \geq 0$. There is a 1-1 correspondance between $s, \lambda$

Ridge regression does not penalize the intercept. $X_{ij}$ is replaced by $X_{ij} - \bar{X}_j$
Use cross-validation to find the appropriate $\lambda$

$$RSS^{ridge}(\lambda) = (Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$$

$$\frac{\partial}{\partial\beta}RSS^{ridge}(\lambda) = -2X^TY + 2(X^TX + \lambda I)\beta$$

$$\widehat{\boldsymbol{\beta}}^{ridge} = (X^TX + \lambda I)^{-1}X^TY$$

$$X^TX + \lambda I = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} + \begin{bmatrix} \lambda & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \lambda \end{bmatrix}$$

Adding $\lambda$ makes $(X^TX)^{-1}$ stable

OLS $\Rightarrow s = \infty$. As s decreases, it forces $\beta_i$ to be smaller

### 8.1.1 Shrinkage Properties of Ridge Regression

$R = X^TX$ assuming $R^{-1}$ exists

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_\lambda^{ridge} &= (X^TX + \lambda I_p)^{-1}X^TY \\ &= (R + \lambda I_p)^{-1}R(R^{-1}X^TY) \\ &= [R(I_p + \lambda R)^{-1}]^{-1}R[(X^TX)^{-1}X^TY] \\ &= [I_p + \lambda R^{-1}]^{-1}R^{-1}R\widehat{\boldsymbol{\beta}}_{LSE} \\ &= [I_p + \lambda R^{-1}]^{-1}\widehat{\boldsymbol{\beta}}_{LSE} \\ E[\widehat{\boldsymbol{\beta}}_\lambda^{ridge}] &= E[(I_p + \lambda R^{-1})^{-1}\widehat{\boldsymbol{\beta}}_{LSE}] \\ &= (I_p + \lambda R^{-1})^{-1}\beta \end{aligned}$$

### 8.1.2 Special Case: If $X^TX = I_p$

$\widehat{\boldsymbol{\beta}}_{LSE} = (X^TX)^{-1}X^TY = X^TY$ $\widehat{\boldsymbol{\beta}}_\lambda^{ridge} = \frac{1}{1+\lambda}X^TY = \frac{1}{1+\lambda}\widehat{\boldsymbol{\beta}}_{LSE}$

Each element of $\widehat{\boldsymbol{\beta}}_{LSE}$ being shrunk by $\frac{1}{1+\lambda} \Rightarrow$ large $\lambda$, more shrinkage

$$\begin{aligned} \hat{Y}_{LSE} = X\widehat{\boldsymbol{\beta}}_{LSE} &= X(X'X)^{-1}X^TY \\ &= UDV^T[(UDV^T)'(UDV^T)]^{-1}(UDV^T)^TY \\ &= UU^TY \\ &= \sum_{j=1}^{p} U_{j_{nx1}}U^T_{j_{1xn}}Y_{nx1} \end{aligned}$$

$$\hat{Y}_{ridge} = X\widehat{\boldsymbol{\beta}}^{ridge} = \sum_{j=1}^{p} U_j \frac{d_j^2}{d_j^2 + \lambda}U_j^TY$$

If $d_i < d_j \Rightarrow \frac{d_i^2}{d_i^2+\lambda} < \frac{d_j^2}{d_j^2+\lambda}$

As $d_j$ decreases, sample variance decreases

Model selection form of shrinkage-setting certain covariates to zero

### 8.1.3 MSE of $\widehat{\beta}_{LSE}, \widehat{\beta}^{ridge}$

Let $X = (X_1, \ldots, X_p)$ and $Z = (XV_1, \ldots, XV_p) \leftarrow$ predictors
$\tilde{X} = XV = UDV^TV = UD$

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_{LSE} &= (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^TY = (D^TU^TUD)^{-1}D^TU^TY \\
\widehat{\boldsymbol{\beta}}_{LSE} &= diag(\frac{1}{d_j})U^TY \\
\widehat{\boldsymbol{\beta}}_{LSE,j} &= \frac{1}{d_j}U_j^TY \\
\widehat{\boldsymbol{\beta}}^{ridge} &= (\tilde{X}^T\tilde{X} + \lambda I_p)^{-1}\tilde{X}^TY = diag(\frac{d_j}{d_j^2+\lambda})U^TY \\
\widehat{\boldsymbol{\beta}}_j^{ridge} &= \frac{d_j}{d_j^2+\lambda}U_j^TY = \frac{d_j^2}{d_j^2+\lambda}\widehat{\boldsymbol{\beta}}_{LSE} \\
Var(\widehat{\boldsymbol{\beta}}_{LSE,j}) &= \frac{\sigma^2}{d_j^2} \\
Var(\widehat{\boldsymbol{\beta}}_j^{ridge}) &= \frac{\sigma^2 d_j^2}{(d_j^2+\lambda)^2} \\
E[(\beta_j - \widehat{\boldsymbol{\beta}}_{LSE,j})^2] &= MSE(\widehat{\boldsymbol{\beta}}_{LSE,j}) = \frac{\sigma^2}{d_j^2} \\
MSE(\widehat{\boldsymbol{\beta}}_j^{ridge}) &= (\beta_j - \beta_j\frac{d_j^2}{d_j^2+\lambda})^2 + \frac{\sigma^2}{d_j^2}(\frac{d_j^2}{d_j^2+\lambda})^2 \\
&= \frac{\sigma^2}{d_j^2}\frac{d_j^2(d_j^2+\lambda^2\beta_j^2/\sigma^2)}{(d_j^2+\lambda^2)} \\
&= MSE(\widehat{\boldsymbol{\beta}}_{LSE}) * \text{ something greater than 1 } \Rightarrow MSE(\widehat{\boldsymbol{\beta}}^{ridge}) < MSE(\widehat{\boldsymbol{\beta}}_{LSE})
\end{aligned}$$

## 8.2 Principle Components

Singular Value Decomposition: $X_{nxp} = U_{nxp}D_{pxp}V_{pxp}^T$ where X is centered $U^TU = V^TV = I_p$ $D$ diagonal $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ $U$ provides an orthonormal basis for the column space of $X$
$X^TX = (UDV^T)^T(UDV^T) = VDV^T$

**Definition 8.** *Columns of $V$ are called* principle component directions *of $X$*

$Z_j = XV_j$ principle components of $X$ $Z_j = UDV^TV_j = U_jd_j =$ jth column of U $*$jth element of D
$V_i^TV_j = 0 \quad i \neq j$

$$Z_1 = XV_1 = U_1d_1 = \begin{pmatrix} X_{11}V_{11} + \cdots + X_{1p}V_{1p} \\ \vdots \\ X_{n1}V_{n1} + \cdots + X_{np}V_{np} \end{pmatrix}$$

$$V = \begin{bmatrix} \uparrow & & \uparrow \\ V_1 & & V_p \\ \downarrow & & \downarrow \end{bmatrix} \qquad V_j = \begin{pmatrix} V_{ij} & \dots & V_{pj} \end{pmatrix}$$

Sample variance of $Z_1$ = sample variance of $d_1U_1 = \frac{1}{n}(U_1d_1)^T(U_1d_1) = \frac{1}{n}d_1^2U_1^TU_1 = \frac{1}{n}d_1^2$ Sample variance of $Z_2 = \frac{1}{n}d_2^2 \dots$ Sample variance of $Z_p = \frac{1}{n}d_p^2$

By construction, $\text{SV}(Z_1) \geq \text{SV}(Z_2) \cdots \geq \text{SV}(Z_p)$

Principle components - $Z_1$ trying to find greatest variance among $X$, the covariate space. Ex. - Covariates $X_1$ and $X_2$

$$Z_1 = \begin{bmatrix} \uparrow & \uparrow \\ X_1 & X_2 \\ \downarrow & \downarrow \end{bmatrix}_{nx2} \begin{bmatrix} \uparrow \\ V_1 \\ \downarrow \end{bmatrix}_{2x1}$$

Each data point multiplied by the corresponding weights. $Z_1$ lower dimensional summary of covariate vector.

$nxp, \quad p = 5000$ projection of data points on lower dimensional space.

Principle components are orthogonal so multicollinearity is mitigated.

### 8.2.1 Scree Plot

See the example for the function `screeplot`. It produces a plot of the variance of each of the principle components or, alternatively, the proportions $d_j^2 / \sum_{k=1}^{p} d_k^2$. It is used to decide how many components to include.

### 8.2.2 Principle Components Regression

$$Z_j = d_jU_j = XV_j$$

Idea: Regress on $Z_j$ $Y = Z\beta + \epsilon = XV\beta + \epsilon$ $Y \sim Z_1$ $Y \sim Z_1 + Z_2$ $Y \sim Z_1 + \dots Z_p$

Do variable selection on principle components using CV, AIC, BIC

Benefits - no multicollinearity, clear what order models should be looked at

However, no reason to believe why response is correlated more with $Z_i$ than $Z_{i+1}$

Principle component analysis - look at $X$, summarize as $XV_1, \dots, XV_p$ (lower dimensional)

Principle component regression - relate to response.

### 8.2.3 PCA Formulation

$$X = \begin{bmatrix} \uparrow & & \uparrow \\ X_1 & \dots & X_p \\ \downarrow & & \downarrow \end{bmatrix}$$

Find $Xw_1$ p x 1 vector such that it yields maximal sample variance - $maxvar(Xw_1)$ but $||w_1|| = 1$

$maxw_1^TX^TXw_1$ subject to $||w_1|| = 1$

$L = w_1^TX^TXw_1 - \lambda(w_1^Tw_1 - 1)$ $\frac{\partial L}{\partial w_1} = 2X^TXw_1 - 2\lambda w_1$

$XV_1$ - largest sample variance

$(X^TX)w_1 = \lambda w_1$ where $w_1$ is eigenvector, $\lambda$ is eigenvalue

Choose $w_1$ according to largest $\lambda$
$X^TX = VD^2V^T$ $D^2$ eigenvalues of $X^TX$

## 8.3 Partial Least Squares Regression

Algorithmic approach to deal with multicollinearity
$max_{w_1} \underbrace{Var(Xw_1)}_{PCA} \underbrace{Cov^2(Xw_1, Y)}_{OLS}$ $||w_1|| = 1$
$\Rightarrow max_{w_1} Cov^2(Xw_1, Y) \Rightarrow$ eigenvalue problem for $X^TYY^TX$, $X^TYY^Tw_1 = \lambda w_1$

### 8.3.1 Procedures in R

data in R - highly correlated predictors

```
library(MASS)
lm.ridge(Y~X, data, lambda(0,.1,.001))
#Argument for lambda (start, end, increment)
lm$lambda #list of lambda values
lm$coef #coefficients
matplot(   )  #Ridge trace plot - change in coefficient based on change in lambda

#As lambda increases - estimates stabilize and approach constant value
#Use cross-validation and intuition

select(lm.ridge(...) )

#Ridge regression automatically centers and standardizes

lm$coef[,1]/lm$scales #regains OLS estimates

#OLS and lm$coef[1,] will be different
#lm.ridge automatically scales
#Use P.C. - must standardize

apply(     )
eigen(   ) # find eigenvalues and eigenvectors

#eigenvectors of X'X are principle components
```

Scree plot - tall group - eigenvectors that capture most variability
Linear combinations have little intuitive meaning.
With eigenvectors - form metapredictors
If there was more data, compare in terms of prediction error

```
library(pls)
plsr(    )  # partial least squares regression can deal with multivariate response

lm2$scores # XW vector
```

PCR - $max w_1^T X^T X w_1$ subject to $w^T w = 1$ $V[,1] = w_1$ - eigenvector from $X^T X$ $V = eigen(X^T X)\$vector$ - new predictor $X w_1$

Partial least squares $max w_1^T X^T Y Y^T X w_1$ such that $w^T w = 0$ - new predictor $X w_1$

PC $\rightarrow X w_1$ $\quad Y \sim X w_1$ where $w^1$ is max(var(Xw))

$w_1 \rightarrow V[,1]$

PLS $\rightarrow X W_1^*$ $\quad Y \sim X w_1^*$ where $w_1^*$ is $maxCor^2(Y, Xw)$

$R^2$ fair comparison - have the same number of covariates

Usually partial least squares need fewer components than principle components - more informed way of forming direction vector

# Bibliography

C. A. Bache, J. W. Serum, W. D. Youngs, and D. J. Lisk. Polychlorinated Biphenyl residues: Accumulation in Cayuga Lake trout with age. *Science*, 117:1192–1193, 1972.

George E. P. Box and George C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading, MA, 1973.

George E. P. Box and Paul W. Tidwell. Transformations of the independent variables. *Technometrics*, 4:531–550, 1962.

J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *Linpack Users' Guide*. SIAM, Philadelphia, 1979.

G. A. F. Seber. *Linear Regression Analysis*. Wiley, New York, 1977.

G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.