6

Linear Regression – More Topics

This chapter describes additional approaches for fitting a multiple linear regression model. We discuss numerical and graphical diagnostics to detect violation of regression assumptions, including outlying, high leverage, or influential cases, or the problem of multicollinearity in the predictors. The chapter also discusses remedies for detected problems. The last section describes variable selection using classical methods such as best subset regression and stepwise regression, as well as modern approaches such as regularized regression such as lasso, and elasticnet.

Regression Diagnostics

6.1 Introduction

Regression diagnostics refer to graphical, informal numerical procedures, or statistical inferential methods that enable us to assess the validity of a fitted regression model – to assess whether model assumptions are satisfied, to study the effect due to collinearity in the explanatory variables, or study the influence of one or more cases (observations) on different aspects of the fitted model.

Diagnostic measures are used for detecting outliers, high leverage cases, or influential cases. A case may be influential to the regression fit because it is an outlier, or a high-leverage point, or both. Belsley et al. (1980) give an extensive description of these topics, while a more theoretical discussion is found in Chatterjee and Hadi (1988).

Multicollinearity implies that two or more predictors are highly linearly dependent. This issue can crop up in several examples, especially as the predictor dimension increases. Diagnostics for multicollinearity only depend on the predictor matrix $\mathbf{X}$, and remedies range from judiciously omitting problematic predictors from the model to early computer-age methods such as ridge regression, or principal components regression.

Variable selection in MLR models enables us to identify predictors that best explain the response. We discuss classical approaches such as best subsets regression and stepwise regression, as well as more recent approaches such as regularized regression (lasso, or elastic net), random forest regression, gradient boosting, etc.

6.2 Essential Terminology and Notation

✓ The **eigenvalues** $\lambda_1, \ldots, \lambda_k$ of a symmetric $k \times k$ matrix $\mathbf{A}$ are solutions to the equation $|\mathbf{A} - \lambda\mathbf{I}| = 0$. Then, the determinant of $\mathbf{A}$ is the product of the eigenvalues.

✓ The **eigenvectors** $\mathbf{q}_1, \ldots, \mathbf{q}_k$ of a symmetric $k \times k$ matrix $\mathbf{A}$ with eigenvalues $\lambda_j,\ j = 1, \ldots, k$ satisfy the equations $\mathbf{A}\mathbf{q}_j = \lambda_j \mathbf{q}_j,\ j = 1, \ldots, k$.

✓ A square matrix $\mathbf{A}$ is an **orthogonal matrix** if its transpose is its inverse, i.e., $\mathbf{A}' = \mathbf{A}^{-1}$. It follows that $\mathbf{A}\mathbf{A}' = \mathbf{A}'\mathbf{A} = \mathbf{I}$.

✓ A square matrix $\mathbf{A}$ is a **singular matrix** if $|\mathbf{A}| = 0$, and is nearly singular if $|\mathbf{A}|$ is very close to 0.

✓ The **spectral decomposition** of a symmetric matrix $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Delta}\mathbf{Q}' \tag{6.1}$$

where $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_k)$ is a matrix whose columns are the orthogonal eigenvectors of $\mathbf{A}$ corresponding to its ordered eigenvalues denoted by $\lambda_{(1)} \leq \lambda_{(2)} \leq \ldots \leq \lambda_{(k)}$, and $\boldsymbol{\Delta} = \text{diag}(\lambda_{(1)}, \ldots, \lambda_{(k)})$.

✓ Let $\overline{X}_j = \frac{1}{n}\sum_{i=1}^n X_{ij}$, $S_{jh} = \sum_{i=1}^n (X_{ij} - \overline{X}_j)(X_{ih} - \overline{X}_h)$, for $j, h = 1, \ldots, p$, and $S_{yy} = \sum_{i=1}^n (Y_i - \overline{Y})^2$. The **centered and scaled form of the MLR model** is

$$Y_i^* = \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i, \tag{6.2}$$

where for $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

$$\begin{aligned} X_{ij}^* &= \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}}, \\ Y_i^* &= \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}}, \text{ and,} \\ \beta_j^* &= \beta_j (S_{jj}/S_{yy})^{1/2}. \end{aligned}$$

Let

$$\mathbf{X}^* = \begin{pmatrix} X_{11}^* & X_{12}^* & \ldots & X_{1p}^* \\ X_{21}^* & X_{22}^* & \ldots & X_{2p}^* \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1}^* & X_{n2}^* & \ldots & X_{np}^* \end{pmatrix};$$

then $\mathbf{X}^{*\prime}\mathbf{X}^*$ denotes the $p \times p$ sample correlation matrix of the explanatory variables (since the intercept column becomes zero by the centering). The MLR model in (6.29) can be written in vector-matrix notation as

$$\mathbf{y}^* = \mathbf{X}^* \boldsymbol{\beta}^* + \boldsymbol{\epsilon}. \tag{6.3}$$

✓ **Regression diagnostics** refer to graphical or informal numerical procedures or statistical inferential methods that enable us to assess the validity of a fitted regression model. The diagnostics help us assess whether model assumptions are satisfied through a careful look at the model residuals, or to study the influence of one or more cases (observations) on different aspects of the fitted model.

✓ An **outlier** is an anomalous observation that does not reasonably fit the assumed model so that the observed response deviates considerably from the fitted response. Simply stated, an outlying case is one with an unusually large absolute residual.

✓ The **predicted residual** for the ith case is defined as

$$e_{i(i)} = Y_i - \hat{Y}_{i(i)} = \frac{e_i}{1 - h_{ii}}, \tag{6.4}$$

where for $i = 1, \ldots, n$, $\hat{Y}_{i(i)}$ is the fitted value for the ith response when the ith case is omitted in fitting the model (5.4).

✓ A **branch-and-bound algorithm** is a computing paradigm used for solving combinatorial optimization problems, such as selecting a best subset of predictors from a set of p predictors in an MLR model. The set of candidate solutions forms a tree, whose root consists of all the predictors. This algorithm efficiently searches the branches of this tree by using bounds on the objective function to prune large parts of the tree.

6.3 Regression Diagnostics

Choosing a model in order to draw conclusions about the relationship between predictors and some variable, y, or in order to make accurate prediction is only the first step in the data analysis process. After we collect data and fit the model we must check whether the model is adequate. If not, we must check why and see if there are steps we can take which will remediate some problems in our model. We discussed some notions of goodness of fit in previous chapters, and now we provide a more detailed approach which is especially tailored for regression models. We call this process "regression diagnostics", and we use it to check validity of the assumptions of the MLR model, and to investigate whether the data contains any "unusual" points (e.g. outliers in the data set, or points which have a large influence on the results from the modeling.)

In this chapter we still focus on the linear regression framework, so it is good to recall the underlying assumptions, which are the linearity of the relationship between the mean response and predictors, the equal error variance for all cases, and independence and normality of the errors and responses. As usual, we explain the process with a motivating example. In this case, we introduce an interesting dataset which wil allow us to investigate factors contributing to deforestation in Western China.

Deforestation

Example 6.3.1 (Deforestation) Agricultural populations whose livelihood heavily depends on natural resources such as timber production may be affected by deforestation which results from the increasing demand for timber. The data in this example was collected from four mountainous provinces in Western China, (Deng et al. (2011)). The data

file deforest.csv has observations from 405 counties, and $p = 17$ predictors which may help explain changes in the forest cover. The variable `fcover` in our data is the observed cover divided by 1000. The predictors included in this dataset are: total population per square km (`tpop`), agricultural population per square km (`apop`), per capita primary industry added value 10,000 yuan (`gdp1`), per capita secondary industry added value 10,000 yuan (`gdp2`), per capita third industry added value 10,000 yuan (`gdp3`), distance to provincial capital (`distpvc`), distance to port or open city (`distport`), length of highway (`hwaylen`), length of natural express (`explen`), proportion of plain area (`plain`), average elevation (`elev`), average precipitation for many years (`prec`) average slope in degrees (`slope`), average temperature in degrees Celsius (`temp`), accumulated temperature greater than 0 degrees Celsius (`temp0`), total land area (`area`), and an indicator for important reforestation policies (`policy`).

Our research question is stated as follows:

Research question

Is deforestation in the four mountainous provinces in Western China related to any of the predictors in the data? explainability

This is the general question, but we will see that getting the answer is not as simple as fitting a linear model with all 17 predictors. In order to answer it properly, we must go through several steps in which we will identify, and try to remediate some problems.

We read the data from deforest.csv into `data1`. There are 16 continuous predictors, and one binary predictor. The response is the square-root of `fcover`.

```
data1 <- read.csv("/Data/deforest.csv", header = TRUE)
data1$sqrtfcover <- sqrt(data1$fcover)
```

We do an 80-20 split of `data1` into a training (calibration, or fitting) set, and a test (hold-out) set using code shown below. That is, starting with a random seed, we randomly sample without replacement 80% of the rows of `data1` into a training set denoted by `train.set`, while the remaining 20% go into a test set, denoted by `test.set`. We will build a regression model on the training data and validate the fit on the test data.

```
set.seed(123457)
train.prop <- 0.80
trnset <- sort(sample(1:nrow(data1), ceiling(nrow(data1)*train.prop)))
# create the training and test sets
train.set <- data1[trnset, ]
test.set  <- data1[-trnset, ]
```

Next, we standardize the train and test sets. To do this correctly, we must use the mean and standard deviation of the training set to standardize the continuous variables in *both* the training and test sets using the R package *caret*, as the code below shows. Note that we do not standardize the binary predictior `policy` is binary. The data sets `data.train` and `data.test` contain the standardized continuous predictors, the binary predictor and the response (square root of `fcover`).

```
library(caret)
```

```
contpredcols <- 5:20
# Find mean and std dev of train.set.1
normParam <- preProcess(train.set[,contpredcols],
                        method = c("center", "scale"))
# standardize the training set based on its mean and std
data.train <- cbind(train.set[,c("sqrtfcover", "policy")],
                    predict(normParam, train.set[,contpredcols]))
# standardize the test set based on the above mean and std dev of the
    training set
data.test <- cbind(test.set[,c("sqrtfcover", "policy")],
                   predict(normParam, test.set[,5:20]))
```

Note that it would be incorrect to (a) use the mean and standard deviation of the entire data to standardize both sets, or (b) use the mean and standard deviation of the test data to standardize the test data set.

Remark 1. If the binary or categorical variable is character-valued (and not factors), then the function preProcess() will find the mean and scale only for continuous variables.

We regress the response (square root of forest cover) on the predictors in `pred`, which has $n = 324$ rows.

```
mod.1 <- lm(sqrtfcover ~., data = data.train)
summary(mod.1)
plot(mod.1)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.33554    0.18571  61.037  < 2e-16 ***
policy       -0.34658    0.25149  -1.378  0.16918
tpop          1.24817    0.60013   2.080  0.03837 *
apop         -1.60689    0.58919  -2.727  0.00675 **
gdp1          0.02893    0.12092   0.239  0.81107
gdp2          0.03107    0.12316   0.252  0.80097
gdp3         -0.12546    0.10933  -1.148  0.25207
distpvc       0.64993    0.34667   1.875  0.06177 .
distport     -1.63072    0.19628  -8.308 3.19e-15 ***
hwaylen      -0.69049    0.34755  -1.987  0.04785 *
explen       -0.06788    0.11886  -0.571  0.56837
plain        -0.33745    0.13519  -2.496  0.01308 *
elev         -0.02341    0.32213  -0.073  0.94210
prec          0.38424    0.21509   1.786  0.07502 .
slope         2.09290    0.18867  11.093  < 2e-16 ***
temp         11.82725    1.50847   7.841 7.51e-14 ***
temp0       -10.35239    1.48779  -6.958 2.10e-11 ***
area          4.48566    0.18103  24.779  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.856 on 306 degrees of freedom
Multiple R-squared:  0.8428,        Adjusted R-squared:  0.8341
F-statistic:  96.5 on 17 and 306 DF,  p-value: < 2.2e-16
```

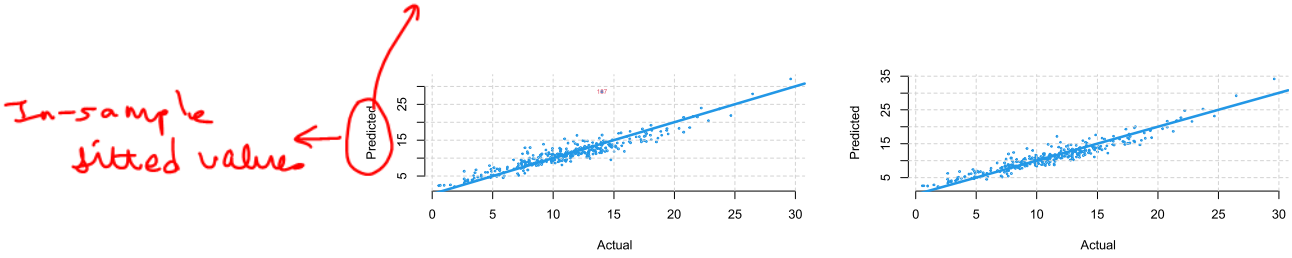


FIGURE 6.1: Fitted versus actual plot from a least squares fit of all standardized predictors on $\sqrt{\text{fcover}}$ for the deforestation data. Left plot uses all training data, right plot excludes the two outliers.

Based on the p-values, we can see that some variables are significant in explaining $\sqrt{\text{fcover}}$, while others seem to be ineffective. There are two anomalous cases, denoted by `extpts`, cases 187 and 152. The fitted versus actual plot for the training data in the left panel of Figure 6.1 shows reasonably good fit, and indicates an outlier, case 187. Note that in an MLR model, we classify the ith case (or observation) as an outlier if the magnitude of the raw residual, internally Studentized residual r_i, and/or the externally Studentized residual r_i^* (see Table 5.1) is large when compared with the rest of the cases in the data set. Note that the plot(mod.1) function can produce a set of five useful plots.

```
plot(data.train$sqrtfcover, predict(mod.1,newdata = data.train),
     col=4, cex=0.3, xlab="Actual", ylab="Predicted", axes=FALSE)
extpts <- which(abs(residuals(mod.1)) > 3*sd(residuals(mod.1)))
text(data.train$sqrtfcover[extpts],
     predict(mod.1,newdata = data.train)[extpts],
     rownames(data.train)[extpts], cex=0.5, col=2)
axis(1); axis(2); grid(); abline(0,1, col=4, lwd=3)
```

The presence of outliers may seriously bias parameter estimation and inference in the MLR model discussed in Chapter 5. We can look at results from fitting an MLR model to the training data without the two outliers. We can compare the fitted versus actual plot based on the smaller training data set with the one in Figure 6.1.

```
data.train.2 <- data.train[-extpts,]
mod.2 <- lm(sqrtfcover ~., data = data.train.2)
summary(mod.2)
## diagnostic plots:
plot(mod.2)
## fitted versus actual plot:
plot(data.train.2$sqrtfcover, predict(mod.2,newdata = data.train.2),
     col=4, cex=0.3, xlab="Actual", ylab="Predicted", axes=FALSE)
axis(1); axis(2); grid(); abline(0,1, col=4, lwd=3)
```

6.3.1 High leverage cases

Let $\mathbf{x}_i = (1, X_{i1}, \ldots, X_{ip})'$ be the vector of explanatory variables for the ith case and let $\overline{\mathbf{x}} = (1/n, \overline{X}_1, \ldots, \overline{X}_p)'$ be the average of the columns of the $\mathbf{X}$ matrix. The diagonal elements

h_{ii} of the hat (or projection) matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ have some useful properties:

$$0 \leq h_{ii} \leq 1 \text{ and } \sum_{i=1}^{n} h_{ii} = p + 1. \quad (6.5)$$

A large value of h_{ii} indicates that $\mathbf{x}_i$ is far removed from the average $\bar{\mathbf{x}}$, i.e., the ith case is an *outlier in the X space*. The reciprocal $1/h_{ii}$ is the effective or equivalent number of observations that determines $\widehat{Y}_i$ (Huber (1981)). We may also interpret the off-diagonal elements of $\mathbf{H}$, i.e., h_{ij} as the amount of *leverage* each Y_j has on determining the ith fit $\widehat{Y}_i$, irrespective of its actual value (Hoaglin and Welsch (1978)). Since

$$\widehat{Y}_i = h_{ii}Y_i + \sum_{j \neq i} h_{ij}Y_j, \quad (6.6)$$

a large value of h_{ii} implies that Y_i important in determining $\widehat{Y}_i$, and that the residual e_i is small (since $\text{Var}(e_i) = \sigma^2(1 - h_{ii})$). When $h_{ii} = 1$, we can show that $h_{ij} = 0$ for $j \neq i$, so that $\widehat{Y}_i = Y_i$, indicating that $e_i = 0$, and $\text{Var}(e_i) = 0$. Huber (1981) suggested the following rules of thumb to identify the ith case as a high leverage point: (i) $h_{ii} \geq \frac{2(p+1)}{n}$, or (ii) $h_{ii} > 0.5$.

```
n <- nrow(data.train)
p <- ncol(data.train)-1
(hilev <- which(influence(mod.1)$hat > max(2*(p+1)/n,0.5)))
```

An L-R plot is useful for distinguishing between outlier and high leverage points. This plot combines the leverage values and residuals in a single scatter plot of leverages h_{ii} versus squared normalized residuals $\widehat{a}_i^2$, where $\widehat{a}_i = e_i/\sqrt{\mathbf{e}'\mathbf{e}}$, $i = 1, \ldots, n$. The points in this plot must lie within the triangle satisfying the following conditions: (i) $0 \leq h_{ii} \leq 1$, (ii) $0 \leq \widehat{a}_i^2 \leq 1$, and (iii) $h_{ii} + \widehat{a}_i^2 \leq 1$. Points that fall in the lower right corner of the L-R plot are outliers, while points that fall in the upper left corner are high leverage points. The plot is constructed using the code below and is shown in Figure 6.2.

```
par(mfrow=c(1,1))
plot(rstandard(mod.1)^2, influence(mod.1)$hat, pch =19, cex=0.5, col="blue",
     xlab = "squared residual", ylab = "leverage")
inf0 <- which(influence(mod.1)$hat > 0.5)
text(rstandard(mod.1)[hilev]^2, influence(mod.1)$hat[hilev], labels = inf0,
    cex = 0.9, font =2, pos =1)
```

6.3.2 Influential cases

We say that an observation (or case) is influential, if its deletion from the data set would noticeably change a result in the fitted MLR model. Neither outliers nor high-leverage points are necessarily influential. We can choose to study the influence of the ith case on the properties of $\widehat{\boldsymbol{\beta}}$, or the fitted values $\widehat{\mathbf{y}}$, or individual estimated coefficients $\widehat{\beta}_j$ $j = 1, \ldots, p$. Here, we discuss three widely used measures of influence Huber (1981)), i.e., Cook's Distance, Welsch-Kuh Distance (DFFITS), and DFBETAS. These measures follow from an application

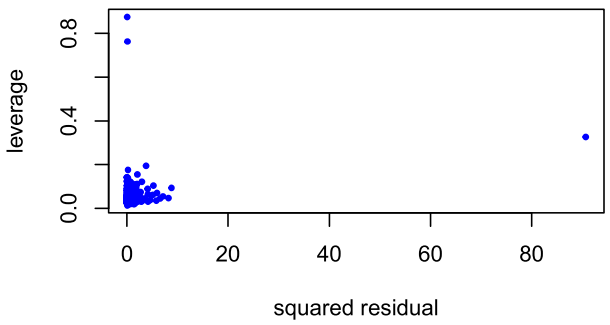


FIGURE 6.2: L-R plot for the deforestation data.

of the influence function to the MLR model (Cook and Weisberg (1983); Chatterjee and Hadi (1988)) and are summarized in Table 6.1.

We need the following notation. For $j = 1, \ldots, p$, if $\mathbf{x}_j$ denotes the column for the jth predictor, let $\mathbf{X}_{(j)}$ be the predictor matrix without the jth column; then, let $\mathbf{H}_{(j)} = \mathbf{X}_{(j)}(\mathbf{X}'_{(j)}\mathbf{X}_{(j)})^{-1}\mathbf{X}'_{(j)}$, and $\mathbf{W}_j = (\mathbf{I} - \mathbf{H}_{(j)})\mathbf{x}_j$.

TABLE 6.1: Influence diagnostics

Diagnostic	Formula	Influence of the ith case on what
Cook's distance	$C_i = \frac{1}{p}\frac{h_{ii}}{(1-h_{ii})}r_i^2$	on least squares estimates and fits
DFFITS	$WK_i = \lvert r_i^* \rvert \sqrt{\frac{h_{ii}}{1-h_{ii}}}$	on the ith fitted response $\hat{Y}_i$
DFBETAS	$r_i^* \frac{w_{ij}}{\sqrt{\mathbf{W}_j'\mathbf{W}_j}} \frac{1}{\sqrt{1-h_{ii}}}$	on the jth coefficient $\hat{\beta}_j$

Cook's Distance

Cook's Distance C_i is an influence diagnostic which can be calculated from a single MLR model fit using the formula shown in Table 6.1, as a function of the leverage h_{ii} and Studentized residual r_i^2 for the ith case. Note that C_i is large if h_{ii} is large, or r_i^2 is large, or both. Although it has been suggested that each C_i be compared to $F_{p,n-p,1-\alpha}$, we can use a boxplot of C_i, $i = 1, \ldots, n$ to answer the question 'how large is large?'. Two other useful interpretations of C_i are given in the appendix.

Welsch-Kuh Distance (DFFITS)

The Welsch-Kuh distance (DFFITS) measures the influence of the ith case on the ith fitted response $\hat{Y}_i$ by the change in the ith fitted value when the ith case is omitted, relative to

the se($\hat{Y}_i$):

$$WK_i = \frac{|\hat{Y}_i - \hat{Y}_{i(i)}|}{\hat{\sigma}_{(i)}\sqrt{h_{ii}}} \tag{6.7}$$

While cut-off points like $2\sqrt{p/(n-p)}$ have been recommended to decide whether WK_i is large, we can also look at a boxplot of these values and see whether some cases are outliers.

DFBETAS

It is also possible to measure the influence of the ith case on a single regression coefficient $\hat{\beta}_j$ using DFBETAS$_{ij}$ shown in Belsley et al. (1980) suggested that absolute values of $DFBETAS_{ij}$ exceeding $2/\sqrt{n}$ are influential on $\hat{\beta}_j$. For the deforestation data, we can obtain the DFBETAS for each β_j using the code below.

```
dfbetas.all  <- as.data.frame(dfbetas(mod.1))
dfbet.fun=function(x){
  which(abs(x) > 2/sqrt(n))
}
hidfbeta=apply(dfbetas.all, 2, dfbet.fun)
```

We can also look at boxplots of the residuals, leverages, Cook's D, DFFITS, and selected DFBETAS (plots not shown here).

```
# Boxplots for residuals, leverages, Cook's D and DFFITS
par(mfrow=c(2,2))
boxplot (rstudent (mod.1), sub =" Stud . resid ")
boxplot (influence(mod.1)$hat , sub=" leverages ")
boxplot(cooks.distance(mod.1), sub = "Cook's D")
boxplot(dffits(mod.1), sub = "DFFITS")
par(mfrow = c(1,2))
boxplot(dfbetas.all[,3], sub ="apop")
boxplot(dfbetas.all[,4], sub ="gdp1")
```

6.4 Multicollinearity

As we saw in Chapter 5, an MLR model fit is useful if the response variable Y is highly correlated with the set of explanatory variables, as indicated by a high R^2 value. However, it is necessary that the explanatory variables are not highly correlated *among themselves*. The problem of multicollinearity exists when there is high correlations or 'near-dependency' between the columns of $\mathbf{X}$, resulting in the matrix $\mathbf{X}'\mathbf{X}$ being nearly singular. Then, from (5.21) and (5.26), we see that the least squares estimates as well as their variances and covariances can become unstable. In such cases, the data/model pair is said to be ill-conditioned.

Although multicollinearity does not affect the fitted values much, a high degree of multicollinearity tends to cause the following problems in an MLR fit:

$Y = X\beta + \varepsilon$
MLR model
$p > 1$ predictors

(i) The standard errors of the regression coefficients will be very large, resulting in small associated t-statistics , leading to the conclusion that truly useful explanatory variables are insignificant in explaining the regression.

(ii) The sign of regression coefficients may be the opposite of what a mechanistic understanding of the problem would suggest.

(iii) Deleting a row or a column of the $\mathbf{X}$ matrix could cause large changes in the estimates corresponding to coefficients of other variables; in general, the regression fit can be unstable.

Several approaches have been suggested in the literature for the detection of multicollinearity as well as to mitigate or remedy its effect.

Deforestation

Example 6.4.1 (Deforestation) We continue to use the deforestation data from the deforestation.csv file in order to investigate factors driving deforestation in four mountainous provinces in Western China. We continue to use the standardized continuous predictors and square root of forest cover.

Our research question is now stated as follows:

Research question #2

Are any of the 17 predictors linearly dependent on any of the others? If this is the case, how can we detect which predictors are? How can we remedy this and fit a regression to explain `sqrtfcover`?

6.4.1 Detecting Multicollinearity

Assess correlations among predictors

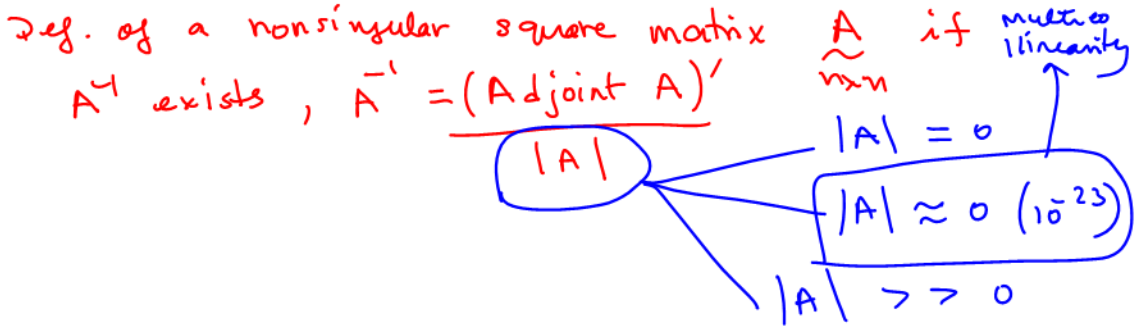
A simple approach consists of assessing the simple, multiple, and partial correlations among the predictors. A commonly used rule is that if a correlation coefficient exceeds 0.95 or 0.99, there is a possibility of collinearity between the corresponding pair of predictors.

```
pred.df <- data.train[,-1] #data frame of predictors only
cor.pred <- cor(pred.df)
off.diag <- function(x) x[col(x) > row(x)]
v <- off.diag(cor.pred)
table(v >=0.95)
table(v >=0.99)
```

While two correlations exceed 0.95, one correlation exceeds 0.99. However, pairwise correlations do not give much insight into interrelationships between three or more predictors.

Another rule suggests multicollinearity if a simple correlation exceeds the multiple correlation from the MLR fit as shown below.

The code below shows that three simple correlations exceed the R^2 value shown in the output `mod.1` shown in Section 6.3.



```
table(v > summary(mod.1)$r.squared)
```

We can also compute the pairwise *partial* correlation coefficients between the predictors. One suggested rule of thumb is a partial correlation coefficient greater than 0.5. However, the cut-off value of 0.5 is quite arbitrary, and further partial correlations only capture linear relationships between pairs of variables, and would not be useful in detecting multicollinearity that is a result of a more complex dependency. For example, X_1 may not be highly correlated individually with X_2 or X_3, but it may be highly correlated with some linear combination of X_2 and X_3.

```
library(ppcor)
pcor1 <- pcor(pred.df)
vp <- off.diag(pcor1$estimate)
table(vp >= 0.5)
```

Results from the code above shows that three partial correlations exceed 0.5. Another indication of multicollinearity is that the F-statistic from the MLR model fit is large implying that the predictors taken together explain the mean response well, but almost all the marginal t-statistics for the individual coefficients are small. In such cases, it is difficult to decide whether the coefficient a for predictor is insignificant due to multicollinearity or simply because the predictor does not explain the response well. Also, this procedure would not indicate which explanatory variable is highly correlated with other variables. Variance inflation factors (VIFs) are helpful diagnostics forthis purpose.

Variance Inflation Factor (VIF)

For $j = 1, \ldots, p$, let R_j^2 denote the coefficient of determination of a linear regression of the explanatory variable X_j on the remaining $p-1$ explanatory variables, i.e.,

$$X_{i,j} = \gamma_0 + \sum_{\ell=1}^{j-1} \gamma_\ell X_{i,\ell} + \sum_{\ell=j+1}^{p} \gamma_\ell X_{i,\ell} + \epsilon_i, \; i = 1, \ldots, n.$$

A large value of R_j^2 close to 1 indicates that X_j is collinear with at least one of the other $p-1$ explanatory variables. The quantity

$$VIF_j = \frac{1}{1 - R_j^2} \tag{6.8}$$

denotes the variance inflation factor for X_j. If X_j is uncorrelated with (or orthogonal to) the other $p-1$ explanatory variables, $R_j^2 = 0$, so that $VIF_j = 1$. As R_j^2 increases from zero, VIF_j increases as well. For example, if $R_j^2 = 0.8$, then $VIF_j = 5.0$, while if $R_j^2 = 0.99$, then $VIF_j = 100$.

General guidelines exist in the literature on how large VIF_j should be to indicate multicollinearity. One suggestion is that any VIF_j greater than 10 (or, greater than 30) indicates multicollinearity. Alternatively, we can compute the average of the variance inflation factors, i.e., $\overline{VIF} = \frac{1}{p}\sum_{j=1}^{p} VIF_j$, and if $\overline{VIF}$ considerably exceeds 1, multicollinearity is indicated.

The code and results below show that the VIF's corresponding to `temp` and `temp0` exceed

200. Although the values are much lower, `tpop`, `apop`, `distpvc` and `hwaylen` also have VIF's bigger than 10. The results suggest evidence of multicollinearity.

```
car::vif(mod.1)
```

```
>car::vif(mod.1)
    policy       tpop       apop       gdp1       gdp2       gdp3    distpvc   distport
  1.409765  33.773757  32.553432   1.371252   1.422323   1.120951  11.269900   3.612788
   hwaylen     explen      plain       elev       prec      slope       temp      temp0
 11.327585   1.324763   1.713787   9.730837   4.338333   3.338234 213.387048 207.574291
      area
  3.073224
```

Condition Indices and Condition Number

Let the (real) eigenvalues of $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ be $\lambda_j,\ j = 1, \ldots, p$; $0 \leq |\mathbf{X}^{*\prime}\mathbf{X}^{*}| \leq 1$. Since $|\mathbf{X}^{*\prime}\mathbf{X}^{*}| = \prod_{j=1}^{p} \lambda_j$, if one or more eigenvalues are close to zero, $|\mathbf{X}^{*\prime}\mathbf{X}^{*}|$ will be close to zero as well. Let $d_j = \lambda_j^2,\ j = 1, \ldots, p$. The k *condition indices* are defined as

$$\eta_j = \frac{d_{\max}}{d_j},\ j = 1, \ldots, p. \tag{6.9}$$

A value of d_j which is relatively close to zero will be associated with a large condition index. If d_j is exactly equal to zero, there is an exact linear relationship among the p explanatory variables.

The code below shows how to construct the condition indices. Note that unlike the VIF's, the 18 values shown below must not be associated drectly with the intercept and 17 predictors (see the appendox for some details).

```
library(olsrr)
(mod.condind <- ols_eigen_cindex(mod.1)[,2])
```

```
 [1]  1.000000  1.672370  1.764450  1.932458  2.373442  2.448060  2.632633  2.723273
 [9]  3.128310  3.651675  4.341692  4.968210  5.477556  6.430869  8.484049 11.364146
[17] 20.191643 50.895901
```

The *condition number* is defined as

$$C = \frac{d_{\max}}{d_{\min}}, \tag{6.10}$$

and always exceeds 1. A large condition number, say $C > 15$ (or $C > 30$), indicates evidence of strong multicollinearity and a need for corrective action. The code below enables us to compute the condition number of 50.8959 for this data.

```
(mod.condnum <- max(mod.condind)/min(mod.condind))
```

6.4.2 Remedies for Multicollinearity

We discuss a few usual remedies for handling multicollinearity in MLR models.

Dropping predictors from the model

Once detected, an obvious remedy could be to drop the variables that are highly correlated with the others. However, if the dropped variable is potentially valuable in an understanding of the response, we would get absolutely no information about it. Moreover, it may not always be clear as to how the omission of a variable will affect the estimates of the remaining model parameters. A second, but not a very promising, solution to the problem is to include additional data, if available, to the analysis; in some cases, this might resolve the multicollinearity. Formal and more rigorous statistical procedures for dealing with the multicollinearity problem include ridge regression, and principal components regression.

We refit a model by excluding one of the predictors with the largest VIF; here, we can drop the variable `temp`, and refit the MLR model and obtain the VIF's.

```
data.train.1 <- subset(data.train, select = -c(temp))
mod.droptemp <- lm(sqrtfcover ~., data = data.train.1)
summary(mod.droptemp)
anova(mod.droptemp)
 car::vif(mod.droptemp)
```

```
> car::vif(mod.droptemp)
   policy      tpop      apop      gdp1      gdp2      gdp3   distpvc  distport   hwaylen
 1.396241 33.509798 32.528053  1.356162  1.420866  1.120451 10.329263  3.375474 10.409090
   explen     plain      elev      prec     slope     temp0      area
 1.311268  1.701280  9.713470  4.224137  3.052512  6.356137  2.704171
```

The VIF's are now much smaller than 200, and the VIF for `temp0` is only 6.35. We now see two variables, `tpop` and `apop`, with VIF's exceeding 30, and two variables `distpvc` and `hwaylen` with VIF's just above 10. We can refit an MLR by dropping `tpop`, and get the VIF's shown below.

```
data.train.2 <- subset(data.train.1, select = -c(tpop))
mod.droptpop <- lm(sqrtfcover ~ ., data = data.train.2)
summary(mod.droptpop)
anova(mod.droptpop)
car::vif(mod.droptpop)
```

```
> car::vif(mod.droptpop)
   policy      apop      gdp1      gdp2      gdp3   distpvc  distport   hwaylen    explen
 1.394528  1.901877  1.351789  1.418626  1.118372 10.006197  3.286475 10.191837  1.308195
    plain      elev      prec     slope     temp0      area
 1.674786  9.713219  4.182711  3.022363  6.336799  2.695253
```

The LS estimates from this model after dropping `temp` and `tpop` are shown below.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.20193    0.20421  54.854  < 2e-16 ***
policy      -0.12904    0.27608  -0.467 0.640528
apop        -0.14494    0.15719  -0.922 0.357201
gdp1         0.14740    0.13252   1.112 0.266880
gdp2         0.07559    0.13576   0.557 0.578068
gdp3        -0.13046    0.12054  -1.082 0.279968
distpvc      1.27179    0.36054   3.527 0.000484 ***
distport    -1.93938    0.20663  -9.386  < 2e-16 ***
hwaylen     -1.33245    0.36387  -3.662 0.000295 ***
explen       0.01025    0.13036   0.079 0.937412
plain       -0.38120    0.14750  -2.584 0.010216 *
elev        -0.13467    0.35523  -0.379 0.704867
prec         0.16914    0.23311   0.726 0.468637
slope        2.47577    0.19815  12.494  < 2e-16 ***
temp0        1.17265    0.28692   4.087 5.58e-05 ***
area         4.02095    0.18712  21.488  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.048 on 308 degrees of freedom
Multiple R-squared:  0.8072,        Adjusted R-squared:  0.7978
F-statistic: 85.98 on 15 and 308 DF,  p-value: < 2.2e-16
```

We can look at the p-values from the marginal t-tests in the above output to see which variables are significant in explaining `sqrtfcover`. We can look at the residual diagnostic plots from this model by the command plot(mod.droptpop).

Remark 1. We can also use the *dplyr* package for dropping the variable `temp`; see the sample code below.

```
data.train.1 <- select(data.train, -c(temp))
```

Ridge Regression

Ridge regression is a procedure which addresses multicollinearity in an MLR model. Instead of minimizing the error sum of squares in the centered and scaled MLR model defined in (6.20), we minimize the error sum of squares subject to a penalty function denoted by $\lambda \parallel \boldsymbol{\beta} \parallel^2]$. We can denote the ridge estimates as $\widehat{\boldsymbol{\beta}}^*(\lambda) = (\widehat{\beta}_1^*(\lambda), \ldots, \widehat{\beta}_p^*(\lambda))'$. When the penalty parameter $\lambda = 0$, the ridge estimates coincide with the LS estimates of the parameters. More details are given in the appendix. A plot of $\widehat{\beta}_j^*(\lambda)$ versus λ is known as the *ridge trace*, and can be useful for selecting a suitable value of $\lambda \neq 0$ such that the ridge estimates of β_j^* will stabilize to reasonable values. We can also choose λ using a k-fold cross-validation approach, as we show below.

For the deforestation data, ridge regression estimates can be obtained using the *glmnet* package. The function maximizes a penalized likelihood function (see Section 6.6 for more details in the context of regularized regression). This function does not work with data frames, so we must create a numeric matrix for the predictors (training features) and a vector of responses (target values).

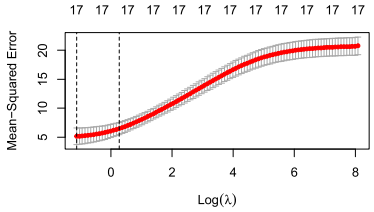


FIGURE 6.3: MSE versus log(λ) using the *glmnet* package

```
pred.mat <- as.matrix(pred.df)
resp <- data.train$sqrtfcover
```

We can use the glmnet() function and manually select the value of λ as follows.

```
mod.ridge.1=glmnet(pred, resp, alpha = 0, nlambda = 100, family = "gaussian",
           standardize=FALSE, intercept = TRUE)
```

We can get the same output by just using the simpler code below, allowing all other options to be at their default settings.

```
mod.ridge.1=glmnet(pred, resp, alpha = 0, standardize=FALSE)
```

If we choose the last of the 100 λ values, we can obtain the ridge estimates at this $\lambda = 0.3240187$ using the code below.

```
(lambda.m  <- mod.ridge.1$lambda[100])
coef(mod.ridge.1, s= lambda.m)
```

The *glmnet* package allows us to select the best λ value using k-fold cross-validation, by minimizing a user selected criterion, such as MSE. We use 10-fold validation below. Figure 6.3 plots the criterion, MSE versus log(λ).

```
cvfit.ridge <- cv.glmnet(pred.mat,resp,alpha=0, standardize=FALSE,
                         type.measure = "mse", nfolds = 10)
plot(cvfit.ridge)
```

We can select the λ value which minimizes the mean cross-validated error.

```
cvfit.ridge$lambda.min
```

```
[1] 0.3240187
[1] 1.308069
```

Or, we can select the λ value which gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

```
cvfit.ridge$lambda.1se
```

```
[1] 1.308069
```

We can obtain the ridge estimates for lambda.min and lambda.1se, as shown below.

```
(all.coef <- cbind(coef(cvfit.ridge, s = "lambda.min"),
                   coef(cvfit.ridge, s = "lambda.1se")))
```

```
(Intercept) 11.17102307 11.14515660
policy      -0.07871911 -0.03660486
tpop         0.17234495 -0.02564813
apop        -0.27277357 -0.12012392
gdp1         0.02156346 -0.09547346
gdp2         0.01684329 -0.06999666
gdp3        -0.16293999 -0.16744178
distpvc      0.58672995  0.48782723
distport    -1.53486017 -1.04680556
hwaylen     -0.33502190  0.18424987
explen       0.05094388  0.10235664
plain       -0.30424267 -0.21065072
elev         0.39327040  0.43910665
prec         0.27549262  0.28368329
slope        1.97680655  1.27846803
temp         0.99010972  0.42703827
temp0        0.24887686  0.25125813
area         3.32437651  2.29857012
```

Note that while the coefficients are shrunk towards zero, no sparsity is achieved since of none of the coefficients becomes exactly zero.

Ridge regression estimates of the standardized coefficient vector $\boldsymbol{\beta}$ can also be obtained using the lm.ridge() function in the *MASS* package.

Another remedy for multicollinearity, called principal components regression, is sometimes used. This method invokes the idea of reducing the dimension of the predictor space by regressing Y on a set of orthogonal linear combinations of the original standardized predictors.

6.5 Variable Selection

In MLR modeling, variable selection consists of using statistical procedures to choose a subset of all available explanatory variables (or predictors or features) for inclusion in the final model (possibly with reduced dimension) in order to best explain and predict the response Y. Starting with p available predictors, suppose we wish to fit an MLR model (5.4) which includes q of these variables, using suitable methods. In general, variable selection seeks the *principle of parsimony*, i.e., if two models fit the data equally well, the simpler (smaller) model is the better model. We describe a few procedures that differ in their approaches and computational complexity.

We continue discussing the deforestation data. Our research question is stated as follows:

Research question #2

Can we explain and predict `sqrtfcover` well using only a select few out of the $p = 17$ predictors? How can we select these variables?

6.5.1 Best Subsets Regression

Given an MLR model (5.4) with p predictors, our goal is to select the best subset $X_1, X_2, \ldots, X_q$, where $q \leq p$, according to some criterion. The *best subsets regression* approach compares all the $2^p + 1$ models based on suitable variable selection criteria. Criteria such as SSE, MSE, R^2 and adjusted R^2 which we defined in Chapter 5 for the MLR model (5.4) are useful criteria for variable selection. We show these, along with a few other useful criteria in the table below.

Criterion	Formula	Best value
SSE_p	$\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$	Smallest
MSE_p	$\frac{\text{SSE}_p}{n-p}$	Smallest
R^2_p	$1 - \frac{\text{SSE}_p}{\text{SST}}$	Largest
$R^2_{\text{adj},p}$	$1 - \left(\frac{n-1}{n-p}\right)\frac{\text{SSE}_p}{\text{SST}}$	Largest
Mallows C_p	$\frac{SSE_p}{MSE} - (n - 2p)$	Small, closest to p
PRESS_p	$\sum_{i=1}^{n} e^2_{i(i)} = \sum_{i=1}^{n}(Y_i - \widehat{Y}_{i(i)})^2$	Smallest
AIC_p	$n\log(\text{SSE}_p) - n\log n + 2p$	Smallest
BIC_p	$n\log(\text{SSE}_p) - n\log n + p\log(n)$	Smallest

Remark 1. When selecting the best model, we look for models with the largest values of R^2 or adjusted R^2, and smallest values of SSE, MSE, PRESS, AIC, or BIC.

Remark 2. Unlike SSE_p, MSE_p does not decrease monotonically as p increases, and serves as a reasonable variable selection criterion.

Remark 3. While R^2_p increases monotonically as p increases and does not penalize the model for increasing dimension, adjusted R^2 penalizes the model for increasing dimension.

Remark 4. Mallows (1973) showed that if a fitted model is adequate and does not suffer from lack of fit, we expect that $C_p \approx p$, while models with substantial bias will tend to fall considerably above the line $C_p = p$.

Remark 5. In general, the BIC_p criterion favors more parsimonious models compared to the AIC_p criterion.

The data with standardized response and predictors is in the data frame `data.train`. The regsubsets() function in the R package *leaps* performs an exhaustive search for the best subsets of the predictors to explain the response Y in an MLR model, using an efficient branch-and-bound algorithm and one of several criteria such as R^2, Mallows C_p, or BIC. When there are p possible predictors, and any subset of the predictors is allowed to specify a model, there will be $2^p + 1$ possible models; this includes a model with just the intercept and none of the predictors.

In the code below, the int = TRUE option includes an intercept in each model. The nvmax = ncol(pred.df) option says that the maximum size of subsets to examine is $p = 17$; the default is 8. The nbest =1 option asks for the best model of each size to report; while the default is also 1, if we set it as 2, then two subsets of each size will be reported. We can also obtain a criterion, such as the Bayesian Information Criterion (BIC).

```
library(leaps)
out <- regsubsets(x = pred.df, y = resp, int = TRUE, nvmax = ncol(pred.df),
    nbest =1)
(bic <- summary(out)$bic)
```

```
 [1] -218.8674 -270.1163 -426.4575 -495.6111 -532.4471 -540.6090 -537.6201 -535.4083
 [9] -532.6811 -529.7310 -525.4680 -522.5265 -517.9942 -512.5991 -506.9044 -501.1818
[17] -495.
```

The following code outputs a logical matrix (output not shown) indicating which elements are in each model, which includes p predictors, $p = 1, \ldots, 17$. We can now pull out more information about the best model based on the smallest BIC (i.e., -505.6398). We write out the estimated coefficients from the best model. The selected predictors are `apop`, `distport`, `slope`, `temp`, `temp0`, and `area`.

```
(elmts <- summary(out)$which)
new <- as.data.frame(cbind(vars, bic))
min.bic <- min(new$bic)
rw <- which(new$bic==min.bic)
coef(out, rw)
```

```
(Intercept)        apop    distport       slope        temp       temp0        area
 11.1226740  -0.4778413  -1.5891113   2.1007565  12.2341041 -10.4065223   4.4836042
```

In the above analysis, we used the default method = "exhaustive" option. Other available options are forward selection, backward selection or sequential replacement. Instead of the BIC criterion, we can use the rsq, adjr2 or cp options corresponding to the R^2, adjusted R^2, or Mallow's C_p. When $p > 50$, we must use the really.big = TRUE option.

6.5.2 Stepwise Regression

Stepwise regression procedures seek to reduce the computing complexity of best subsets regression by sequentially selecting variables based on *partial* F-tests. A popular approach is *stepwise selection* which includes forward selection and backward elimination as special cases. Stepwise selection uses partial F-statistics (equivalently, t-statistics) and associated $p-$values in order to decide whether to include or exclude each of the p predictors into the model.

Stepwise selection for the deforestation data uses the following code and selects a model whose coefficients are shown below. We see that this method selects a model with 12 out of the 17 predictors. Note that `direction = "both"` implies both forward and backward selection.

```
fit.step <- lm(sqrtfcover ~., data = data.train)
mod.step <- step(fit.step, direction = "both", trace = 0)
summary(mod.step)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.3614     0.1770  64.178  < 2e-16 ***
policy       -0.3887     0.2349  -1.654  0.09904 .
tpop          1.2477     0.5936   2.102  0.03635 *
apop         -1.5965     0.5783  -2.761  0.00611 **
distpvc       0.6154     0.3249   1.894  0.05912 .
distport     -1.6484     0.1748  -9.431  < 2e-16 ***
hwaylen      -0.6938     0.3394  -2.044  0.04177 *
plain        -0.3306     0.1331  -2.485  0.01349 *
prec          0.3825     0.1885   2.030  0.04325 *
slope         2.0688     0.1750  11.821  < 2e-16 ***
temp         11.8463     1.4824   7.992 2.63e-14 ***
temp0       -10.3715     1.4628  -7.090 9.05e-12 ***
area          4.4830     0.1681  26.676  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.846 on 311 degrees of freedom
Multiple R-squared:  0.8419,        Adjusted R-squared:  0.8358
F-statistic:   138 on 12 and 311 DF,  p-value: < 2.2e-16
```

How does the stepwise selection work? To illustrate the idea, suppose $K = 4$ predictors are available, denoted by X_1, X_2, X_3, and X_4. Our goal is to select the best MLR model of the form (5.4) involving $p \leq K$ predictors. The algorithm consists of the following steps.

Step 0. Specify two different levels of significance α_{enter} and α_{stay}. These are respectively the probability of a Type I error for including a predictor into the current MLR model and the probability of a Type I error for retaining in the model a predictor variable that was previously entered. Recommended values are $0.05, 0.10$ or 0.15. All the models are fit to cases $i = 1, \ldots, n$.

Step 1. Fit K single predictor models for $j = 1, \ldots, K$ denoted by

$$Y_i = \beta_0 + \beta_1 X_{ij} + \epsilon_i. \tag{6.11}$$

Compute the partial F-statistic (or t-statistic) and the corresponding p-value for testing $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ under each of the K models. Note that under H_0, the model becomes

$$Y_i = \beta_0 + \epsilon_i. \tag{6.12}$$

(i) If the p-values exceed α_{enter} for all K models, the procedure stops, i.e., no variables can enter the model (6.12), and we assume that Y can be best explained by just the intercept, estimated by $\overline{Y}$.

(ii) If at least one of the p-values is less than or equal to α_{enter}, then the predictor with the smallest such p-value (i.e., the largest partial F-statistic value) enters the model (6.12). Denote this variable by $X_{[1]}$, so the model at the end of Step 1 is

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \epsilon_i. \tag{6.13}$$

In the illustration, suppose X_3 enters the model in Step 1, while X_1, X_2 and X_4 do not; then $X_{[1]} = X_3$.

Step 2. Fit $K - 1$ models with two predictors, of which $X_{[1]}$ was selected in Step 1, while a second variable must be selected from the remaining $K - 1$ predictors:

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \beta_2 X_{ij} + \epsilon_i. \tag{6.14}$$

For each model, compute the partial F-statistic and p-value for testing $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$. Let $X_{[2]}$ denote the predictor corresponding to the smallest p-value (or largest t-statistic). The variable $X_{[2]}$ will be included in the model (6.13) if its p-value is smaller than α_{enter}, to give

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \beta_2 X_{i[2]} + \epsilon_i. \tag{6.15}$$

The stepwise procedure then checks to see whether or not $X_{[1]}$ should continue to remain in the model. If the p-value corresponding to $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ in the model (6.15) is smaller than α_{stay}, $X_{[1]}$ is retained in the model and the model at the end of Step 2 is (6.15); otherwise, it is dropped from the model and the current one-variable model after Step 2 is

$$Y_i = \beta_0 + \beta_2 X_{i[2]} + \epsilon_i. \tag{6.16}$$

We now are back to the position at the start of Step 2, and must search for another varible that is significant and can be included in the model. In the illustration, suppose X_1 enters the model in Step 2, so the model now includes $X_{[1]} = X_3$ and $X_{[2]} = X_1$.

This method continues with the steps of adding predictor variables into the model, one at a time. At each step, a variable is added to the model only if it has the smallest p-value (or largest t-statistic) among all variables that are not in the model, and also, it is significant at the α_{enter} level. Once a variable is added to the model, the procedure checks all the variables in the model and removes any variable that is not significant at the α_{stay} level, doing this deletion step before attempting to add another *new* variable. The procedure terminates when all the predictor variables not included in the model are insignificant at the α_{enter} level, or when the variable to be added into the model is the one that was just removed.

Forward selection is a simpler version of stepwise selection, where a variable that has entered the model stays in the model and is not removed. For *backward elimination*, we start with a rich model that includes all K predictors and uses partial F-statistics and corresponding p-values to eliminate variables that are not significant in the presence of other variables; once removed, a variable may not reenter the model.

Remark 1. We can obtain results from forward selection for the deforestation data by running the code below (output is not shown here). changing direction = "both" to direction = "forward". We can also use the regsubsets() function in the *leaps* package.

```
out.forward <- regsubsets(x = pred.df, y = resp, int = TRUE, nbest = 1,
    method = "forward")
```

Remark 2. For backward elimination, use direction = "backward"

6.6 Regularized Regression

(Sparsity inducing regularized (or penalized) regression)

↓

Lasso

Elasticnet

Recall that we obtain the least squares regression estimates coefficients of the MLR model in (5.21) to minimize the sum of squared errors in (5.20) (a loss function). In practice, we split the rows of the data frame randomly into training data and test data. We will build a model on the training dataset, holding out the test data, i.e., not using the test data for model fitting. We will then evaluate performance of the fitted model on the test dataset. This is called the *holdout-validation* approach for evaluating model performance.

If the coefficients of the MLR model are large, they can lead to over-fitting on the training dataset, and the fitted model will not generalize well on the unobserved test data. Regularization can help to overcome this shortcoming by penalizing large coefficients. That is, in regularized regression, we seek to estimate coefficients which minimize a *penalized* error sum of squares.

6.6.1 Lasso regression

The most popular variant of regularized regression is *Lasso regression* which minimizes

$$S(\beta_0, \beta_1, \ldots, \beta_p, \lambda) = \sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|, \qquad (6.17)$$

cf. this to (6.26)

where, $\lambda \sum_{j=1}^{p}|\beta_j|$ is the penalty term, and λ is called the *shrinkage penalty* parameter which we must select. Lasso is an acronym for least absolute shrinkage and selection operator, and the term $\sum_{j=1}^{p}|\beta_j| = \|\boldsymbol{\beta}\|_1$ is called the ℓ_1-norm. By using the ℓ_1-norm constraint, we force some of the regression coefficients to zero, and the corresponding less important predictors will be discarded from the fitted model. The predictors (features) corresponding to non-zero coefficients will remain in the fitted MLR model. By reducing the coefficients of the less important predictors (features) to zero, lasso regression is used for *feature selection*. This is a useful technique, especially when p is large.

We can use the R package *glmnet* to fit Lasso regression to the deforestation data. The regularization path is computed for the penalty (regularization) parameter λ at a grid of values (on the log scale). The code to get Lasso estimates is similar to what we used to carry out ridge regression, except for replacing `alpha=0` by `alpha =1`, which is also the default in the *glmnet* package. We can again use 10-fold cross-validation to select the λ value which minimizes the mean cross-validated error, or the λ value which gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

```
cvfit.lasso <- cv.glmnet(pred.mat, resp,alpha=1,
                         standardize=FALSE, type.measure = "mse", nfolds =
    10)
plot(cvfit.lasso)
```

```
>cvfit.lasso$lambda.min
[1] 0.0006820283
> (cvfit.lasso$lambda.1se)
[1] 0.2181991
```

We can then obtain the lasso estimates corresponding to these two λ values (shown below). When we use the ℓ_2-norm constraint $\|\boldsymbol{\beta}\|_2$ instead of the ℓ_1-norm in (6.17), the regularized regression becomes the ridge regression we saw earlier. Note that while Lasso regression shrinks some of the regression coefficients to zero, ridge regression retains all the model predictors.

6.6.2 Elastic net

Elastic net is a widely used regularized regression approach which is a combination of lasso and ridge regression. Similar to ridge regression, Lasso pertains to a convex optimization problem. However, unlike ridge regression, Lasso is not always strictly convex, and therefore, it need not always have a unique solution. Zou and Hastie (2005) defined "elastic net", which is always strictly convex, and combines the predictive properties of ridge regression with the sparsity properties of Lasso.

The elastic net penalty

$$P_\alpha(\boldsymbol{\beta}) = \alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2 \tag{6.18}$$

is a convex combination of the Lasso penalty and the ridge regression penalty, $\alpha \in (0,1)$. The elastic net estimates minimize the criterion

$$\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|_2^2 + \lambda\{\alpha \| \boldsymbol{\beta} \|_1 + (1-\alpha) \| \boldsymbol{\beta} \|_2^2\},$$

where $\lambda > 0$ is an unknown penalty parameter. The elastic net estimates are also obtained using the R package *glmnet*, and can be viewed as stabilized Lasso estimates of the coefficients. The code below is similar to what we used for ridge and Lasso regressions, except that we let $\alpha = 0.5$.

```
cvfit.enet <- cv.glmnet(x, y,alpha=0.5, standardize=FALSE, type.measure =
    "mse", nfolds = 10)
```

```
>cvfit.enet$lambda.min
[1] 0.00149705
> (cvfit.enet$lambda.1se) # (Default)
[1] 0.4363981
```

By writing the regression estimates side-by-side, we can compare the output from (left to right) OLS, ridge, lasso and elastic net estimates.

```
(all.coef.1 <- cbind(mod.1$coef, coef(cvfit.ridge, s = "lambda.1se"),
coef(cvfit.lasso, s = "lambda.1se"), coef(cvfit.enet, s = "lambda.1se")))
```

LS Ridge Lasso Elasticnet

```
(Intercept)  11.33554099 11.14515660 11.122673986 11.1226740
policy       -0.34657744 -0.03660486  .            .
tpop          1.24816817 -0.02564813  .            .
apop         -1.60688702 -0.12012392  .            .
gdp1          0.02893167 -0.09547346  .            .
gdp2          0.03107325 -0.06999666  .            .
gdp3         -0.12545952 -0.16744178  .            .
distpvc       0.64993428  0.48782723  0.211369389  0.3931055
distport     -1.63071675 -1.04680556 -1.444079017 -1.3527115
hwaylen      -0.69049129  0.18424987  .            .
explen       -0.06787578  0.10235664  .            .
plain        -0.33744629 -0.21065072 -0.001878076  .
elev         -0.02341371  0.43910665  .            .
prec          0.38423546  0.28368329  .            .
slope         2.09290274  1.27846803  1.969709185  1.7016063
temp         11.82725272  0.42703827  0.857053569  0.5672179
temp0       -10.35239292  0.25125813  .            .
area          4.48566486  2.29857012  3.311560175  2.9558426
```

- the coefficients are "shrunk" to zero

Appendix Chapter 6-Statistical Details

Centered and Scaled Regression Model

Remember to center-scale continuous predictors only.

The centered and scaled multiple linear regression model is

$$Y_i^* = \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i, \tag{6.19}$$

where $\beta_j^* = \beta_j (S_{jj}/S_{yy})^{1/2}$, $j = 1, \ldots, p$ and

$$X_{ij}^* = \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}}, \quad \text{and} \quad Y_i^* = \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}}, \quad \text{for } i = 1, \ldots, n;\ j = 1, \ldots, p. \tag{6.20}$$

Let

$$\mathbf{X}^* = \begin{pmatrix} X_{11}^* & X_{12}^* & \ldots & X_{1p}^* \\ X_{21}^* & X_{22}^* & \ldots & X_{2p}^* \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1}^* & X_{n2}^* & \ldots & X_{np}^* \end{pmatrix};$$

then $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ denotes the $p \times p$ sample correlation matrix of the explanatory variables (since the intercept column becomes zero by the centering).

Cook's Distance

Two other interpretations of Cook's distance C_i are useful.

(i) We can interpret C_i as the scaled distance between the centers of the joint confidence region for $\boldsymbol{\beta}$ under two setups (Cook and Weisberg (1983)). The least squares estimate $\hat{\boldsymbol{\beta}}$ is the center of the $100(1-\alpha)\%$ joint (ellipsoidal) confidence region for $\boldsymbol{\beta}$ when all n observations are used to fit the MLR model. When the ith observation is omitted, the center of the the joint confidence region changes to $\hat{\boldsymbol{\beta}}_{(i)}$. Cook's Distance is defined as

$$C_i = \frac{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)})'(\mathbf{X}'\mathbf{X})(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)})}{(p+1)\hat{\sigma}^2}, \ i = 1, \ldots, n. \tag{6.21}$$

(i) We can also interpret C_i as the scaled distance between $\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}_{(i)}$, i.e., the fitted response vector from the MLR model based on n observations, and the fitted model after deleting the ith case respectively:

$$C_i = \frac{(\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)})'(\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)})}{(p+1)\hat{\sigma}^2}, \ i = 1, \ldots, n. \tag{6.22}$$

Condition Indices and Condition Number

Suppose $\boldsymbol{X}^*$ be the $n \times p$ standardized predictor matrix. Let the (real) eigenvalues of $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ be $\lambda_j, \ j = 1, \ldots, p+1; 0 \leq |\mathbf{X}^{*\prime}\mathbf{X}^{*}| \leq 1$. Since $|\mathbf{X}^{*\prime}\mathbf{X}^{*}| = \prod_{j=1}^{p} \lambda_j$, if one or more eigenvalues are close to zero, $|\mathbf{X}^{*\prime}\mathbf{X}^{*}|$ will be close to zero as well.

While the columns of $\mathbf{X}$ are orthogonal if $|\mathbf{X}^{*\prime}\mathbf{X}^{*}| = 1$, it is clear that $|\mathbf{X}^{*\prime}\mathbf{X}^{*}| = 0$, there exists at least one exact linear dependency among the columns of $\mathbf{X}$. The closer to zero that $|\mathbf{X}^{*\prime}\mathbf{X}^{*}|$ is, greater the severity of multicollinearity. However, this measure does not give us an indication of the number or form of exact or near-exact linear dependencies.

The condition number C of $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ is defined by

$$C = \sqrt{\frac{\max(\lambda_1, \ldots, \lambda_p)}{\min(\lambda_1, \ldots, \lambda_p)}}, \tag{6.23}$$

We can use the following approach for detecting multicollinearity, based on column-equilibrating the matrix $\mathbf{X}$ by dividing each column of $\mathbf{X}$ by the sum of squares of its elements. Let $\mathbf{X}_E$ denote the column-equilibrated matrix, in which the sum of squares of each column will be unity. We find the singular value decomposition of $\mathbf{X}_E$, i.e., $\mathbf{X}_E = \mathbf{UDV}'$, where $\mathbf{U}$ is $n \times p$, $\mathbf{D}$ is $p \times p$, and $\mathbf{V}$ is $p \times p$, $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{V}\mathbf{V}' = \mathbf{I}_p$. Let $\mathbf{D} = \text{diag}(d_1, \ldots, d_p)$, where the non-negative d_j's are called the singular values of $\mathbf{X}_E$.

Now

$$\mathbf{X}_E'\mathbf{X}_E = \mathbf{VDU}'\mathbf{UDV}' = \mathbf{VD}^2\mathbf{V}' \tag{6.24}$$

gives the spectral decomposition of the symmetric matrix $\mathbf{X}_E'\mathbf{X}_E$, so that $\mathbf{D}^2 =$

$D^2 = DD = C = \text{diag}(d_1^2, d_2^2, \ldots d_p^2)$

$diag(c_1, \ldots, c_p)$, where $c_j = d_j^2$ are the eigenvalues of $\mathbf{X}_E'\mathbf{X}_E$ while $\mathbf{V} = \{v_{ij}\}$ is the corresponding orthogonal eigenvector matrix. We next obtain

The colldiag implement procedures Belsley et al. (1980)to examine the 'conditioning' of the matrix of independent variables by computing the condition indexes of the matrix. If the condition index larger than 30 it indicates collinearity problems. All large condition indexes may be worth investigating. If a large condition index is associated two or more variables with large variance decomposition proportions,(50%) these variables may be causing collinearity problems.

Ridge Regression

center + scale Y variable and all continuous X variables using mean & SD of training data,

Consider the centered and scaled form of the MLR model defined in (6.20):

$$\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta}^* + \boldsymbol{\epsilon}.$$

For $\lambda \geq 0$, ridge regression estimates of coefficients $\beta_j^*,\ j = 1, \ldots, p$ are obtained as

$$\widehat{\boldsymbol{\beta}}^*(\lambda) = (\mathbf{X}^{*\prime}\mathbf{X}^* + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^{*\prime}\mathbf{y}^*. \tag{6.25}$$

Although this was not explicitly done by Hoerl and Kennard (1970), we can describe the ridge regression estimate $\widehat{\boldsymbol{\beta}}^*(\lambda)$ as a penalized maximum likelihood estimator, i.e.,

L-2 distance, penalty term

$$\widehat{\boldsymbol{\beta}}^*(\lambda) = \arg\min_{\boldsymbol{\beta}}[\| \mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|^2]. \tag{6.26}$$

cf to (6.17)

Difference in objective function between LS and Ridge

Recall: LS, we $\sum_{i=1}^{n} \varepsilon_i^2$ minimized

$\|Y^* - X^*\beta^*\|^2 = \|\varepsilon\|^2$

L_2-distance

l_2 or Euclidean distance

find β value which minimizes the L_2-distance

then β: argmin $\|\beta\|^2 = \sum_{j=1}^{p} (\beta_j^*)^2$

$\beta^* = (\beta_1^*, \ldots, \beta_p^*)$

When the penalty parameter $\lambda = 0$, (6.26) leads to the OLS estimate of $\boldsymbol{\beta}^*$. Values of $\lambda > 0$ penalize choices of $\boldsymbol{\beta}^*$ that lead to large values of $\| \boldsymbol{\beta} \|^2$, biasing $\widehat{\boldsymbol{\beta}}^*(\lambda)$ towards the origin (shrinkage).

The predictors in an MLR model can be numerical or categorical. When there are two or more continuous predictors, it is possible that they have widely different scales. In this case, it is important to consider standardizing them before including in the MLR model. Standardizing X_j consists of subtracting the mean and dividing by the standard deviation, i.e.,

$$X_{ij}^* = \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}},\ i = 1, \ldots, n. \tag{6.27}$$

Failure to do this may result in Standardizing such variables can also help us accurately determine which variables are important in explaining the response. One way to decide whether or not to standardize the predictors is to see whether or not the spreads in the centered (mean-subtracted) variables are very different. If the spreads are more or less the same, standardizing may not be necessary. Also, if the predictors have nearly equal means, it may be enough to just scale the variables, without mean-subtraction in (6.27). Note that we may choose to standardize (or scale) the response Y, or not; the standardized response can be denoted by

$$Y_i^* = \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}},\ i = 1, \ldots, n. \tag{6.28}$$

We do not standardize categorical predictors; for ease of notation, we can just set $X_{ij}^* = X_{ij}$ for categorical predictors.

$\|\beta\| = \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$

L_1 - distance

$\|\beta\|^2 = \sum_{j=1}^{p} \beta_j^2$

L_2-distance

The MLR model with standardized response and continuous predictors can be represented as follows:

$$Y_i^* = \beta_0^* + \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i. \tag{6.29}$$

If we do not standardize Y, we set $Y_i^* = Y_i$ in (6.29). The standardized coefficients β_j^* are related to the original coefficients β_j by

$$\beta_j^* = \beta_j (S_{jj}/S_{yy})^{1/2},\ j = 1, \ldots, p. \tag{6.30}$$

β_0^* denotes the intercept in the model. We recommend to always include an intercept in a model, except when the science of the domain precludes an intercept (i.e., requires fitting a regression through the origin). Once we fit the MLR model in (6.29), we can obtain the original coefficients inverting the relation in (6.30), i.e.,

$$\beta_j = \beta_j^* (S_{yy}/S_{jj})^{1/2},\ j = 1, \ldots, p. \tag{6.31}$$

Bibliography

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, New Jersey: John Wiley & Sons, 2^{nd} ed.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, New York: John Wiley & Sons.

Chatterjee, S. and Hadi, A. S. (1988), *Sensitivity Analysis in Linear Regression*, New York: John Wiley & Sons.

Cleveland, W. S. (1985), *The Elements of Graphing Data*, Monterey, CA: Wadsworth.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, New York: John Wiley & Sons, 3^{rd} ed.

Conover, W. J. and Iman, R. L. (1981), "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, 35, 124–129.

Cook, R. D. and Weisberg, S. (1983), "Diagnostics for heteroscedasticity in regression," *Biometrika*, 70, 1–10.

D'Agostino, R. B. and Stephens, M. A. (1986), *Goodness-of-fit Techniques*, Marcel Dekker.

Deng, X., Huang, J., Uchida, E., Rozelle, S., and Gibson, J. (2011), "Pressure cookers or pressure valves: do roads lead to deforestation in China?" *Journal of Environmental Economics and Management*, 61, 79–94.

Galton, F. (1886), "Regression towards mediocrity in hereditary stature," *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15, 246–263.

Glynn, E. (2005), "Chart of R colors," *Stowers Institute of Medical Research*.

Hoaglin, D. C. and Welsch, R. E. (1978), "The hat matrix in regression and ANOVA," *The American Statistician*, 32, 17–22.

Hoerl, A. E. and Kennard, R. W. (1970), "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, 12, 55–67.

Hollander, M., Wolfe, D. A., and Chicken, E. (2013), *Nonparametric Statistical Methods*, New York: John Wiley & Sons, 3^{rd} ed.

Huber, P. J. (1981), *Robust Statistics*, New York: John Wiley & Sons.

Koch, G. and Edwards, S. (1988), *Clinical efficiency trials with categorical data. In K. E. Peace (ed.), Biopharmaceutical Statistics for Drug Development.*, Marcel Dekker, New York.

Kruskal, W. H. and Wallis, W. A. (1952), "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, 47, 583–621.

Kutner, M. H., Nachtsheim, C. J., Neter, J., and Li, W. (2005), *Applied Linear Statistical Models*, McGraw Hill/Irwin, 5^{th} ed.

Mallows, C. L. (1973), "Some comments on C_p," *Technometrics*, 15, 661–675.

Mann, H. and Whitney, D. (1947), "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, 18, 50–60.

Montgomery, D. C. (2017), *Design and Analysis of Experiments*, New York: John Wiley & Sons, 9^{th} ed.

Muralidharan, K. and Prakash, N. (2017), "Cycling to School: Increasing Secondary School Enrollment for Girls in India," *American Economic Journal: Applied Economics*, 9, 321–50.

Pruzek, R. M. and Helmreich, J. E. (2009), "Enhancing dependent sample analyses with graphics," *Journal of Statistics Education*, 17.

Rahlf, T. (2017), *Data Visualization with R: 100 Examples*, Springer-Verlag.

Ramsey, F. L. and Schafer, D. W. (2013), *The Statistical Sleuth: A Course in Methods of Data Analysis*, Boston: Brooks/Cole, 3^{rd} ed.

Shapiro, S. S. and Wilk, M. B. (1965), "An analysis of variance test for normality," *Biometrika*, 52, 591–611.

Snedecor, G. and Cochran, W. G. (1989), *Statistical Methods*, Blackwell Publishing, 8th ed.

Welch, B. L. (1947), "The generalization of Student's problem when several different population variances are involved," *Biometrika*, 34, 28–35.

Wilcoxon, F. (1949), *Some Rapid Approximate Statistical Procedures*, Stanford.

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net," 67, 301–320.