Feature Selection

Contents

In this module, we will examine the following set of methods for feature selection.

- Theory: Use published literature or domain knowledge to pick variables.
- Filter Methods: Keep predictors that are relevant but not redundant.
- Subset Selection: Identify a subset of p predictors that are related to the outcome. Fit a model with this subset.
- Shrinkage Methods: Fit a model with all p predictors, but the estimated coefficients are shrunken towards zero relative to least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.
- Dimension reduction: Group predictors into a reduced number of components based on similarity. Use the components as predictors.

Dataset

In order to illustrate feature selection methods, we will use a `wine` dataset. In this dataset, a set of White Wines are characterized by chemical properties measured on eleven variables. The quality of each wine is evaluated by sensory tests by wine experts.

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

Source: UCI Archives

Read Data

Before running code below, be sure to set your working directory to point to the folder where the file is saved.

```
wine = read.table("winequality-white.csv",header=TRUE,sep=";")
```

Split Data

```
library(caret)
set.seed(1031)
split = createDataPartition(y=wine$quality,p = 0.7,list = F,groups = 100)
train = wine[split,]
test = wine[-split,]
```

```
str(train)
```

```
## 'data.frame':    3431 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 8.1 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.22 0.27 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.43 0.41 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.5 1.45 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.044 0.033 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 28 11 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 129 63 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.22 2.99 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.45 0.56 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 11 12 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 5 ...
```

Filter Methods

Screen predictors based on their relationship to the outcome variable (relevance) and to other predictors (redundancy) thereby retaining variables that are relevant and non-redundant.

- Relevant: Related to the outcome variable, and
- Non-redundant: Not related to other predictors

Bivariate Filter

A simple method for screening predictors is by examining their bivariate correlations.
* Relevant: High bivariate correlation with outcome * Non-redundant: Low bivariate correlation with other predictors

A correlation matrix tabulates all pairwise correlations among variables in a dataset.

```
cor(train[,-12])
```

```
##                      fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity           1.00000000      -0.01515433  0.27817084     0.09625899
## volatile.acidity       -0.01515433       1.00000000 -0.14698509     0.06864123
## citric.acid             0.27817084      -0.14698509  1.00000000     0.08620642
## residual.sugar          0.09625899       0.06864123  0.08620642     1.00000000
## chlorides               0.02638089       0.07566378  0.11487040     0.08865581
## free.sulfur.dioxide    -0.05566701      -0.10956176  0.10620600     0.28866975
## total.sulfur.dioxide    0.08797108       0.07346168  0.11606864     0.38796486
## density                 0.27149372       0.02819337  0.14976518     0.84145434
## pH                     -0.42442354      -0.04127985 -0.16429459    -0.20744023
## sulphates              -0.01116608      -0.03867351  0.06326400    -0.03076623
## alcohol                -0.11706395       0.08162245 -0.08837213    -0.45185254
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.02638089        -0.055667009          0.087971082
## volatile.acidity      0.07566378        -0.109561759          0.073461685
## citric.acid           0.11487040         0.106205999          0.116068643
## residual.sugar        0.08865581         0.288669752          0.387964865
## chlorides             1.00000000         0.084758073          0.191725704
## free.sulfur.dioxide   0.08475807         1.000000000          0.623359404
## total.sulfur.dioxide  0.19172570         0.623359404          1.000000000
## density               0.25538816         0.283239305          0.517248576
## pH                   -0.08831472         0.004228275         -0.001336239
## sulphates             0.02919814         0.062873142          0.137479481
## alcohol              -0.36003554        -0.248894101         -0.447177639
##                          density           pH   sulphates     alcohol
## fixed.acidity         0.27149372 -0.424423542 -0.01116608 -0.11706395
## volatile.acidity      0.02819337 -0.041279846 -0.03867351  0.08162245
## citric.acid           0.14976518 -0.164294592  0.06326400 -0.08837213
## residual.sugar        0.84145434 -0.207440226 -0.03076623 -0.45185254
## chlorides             0.25538816 -0.088314720  0.02919814 -0.36003554
## free.sulfur.dioxide   0.28323931  0.004228275  0.06287314 -0.24889410
## total.sulfur.dioxide  0.51724858 -0.001336239  0.13747948 -0.44717764
## density               1.00000000 -0.099901225  0.07648947 -0.77091097
## pH                   -0.09990122  1.000000000  0.15860599  0.12194939
## sulphates             0.07648947  0.158605995  1.00000000 -0.02050269
## alcohol              -0.77091097  0.121949392 -0.02050269  1.00000000
```

Let us simplify the matrix to see numbers better

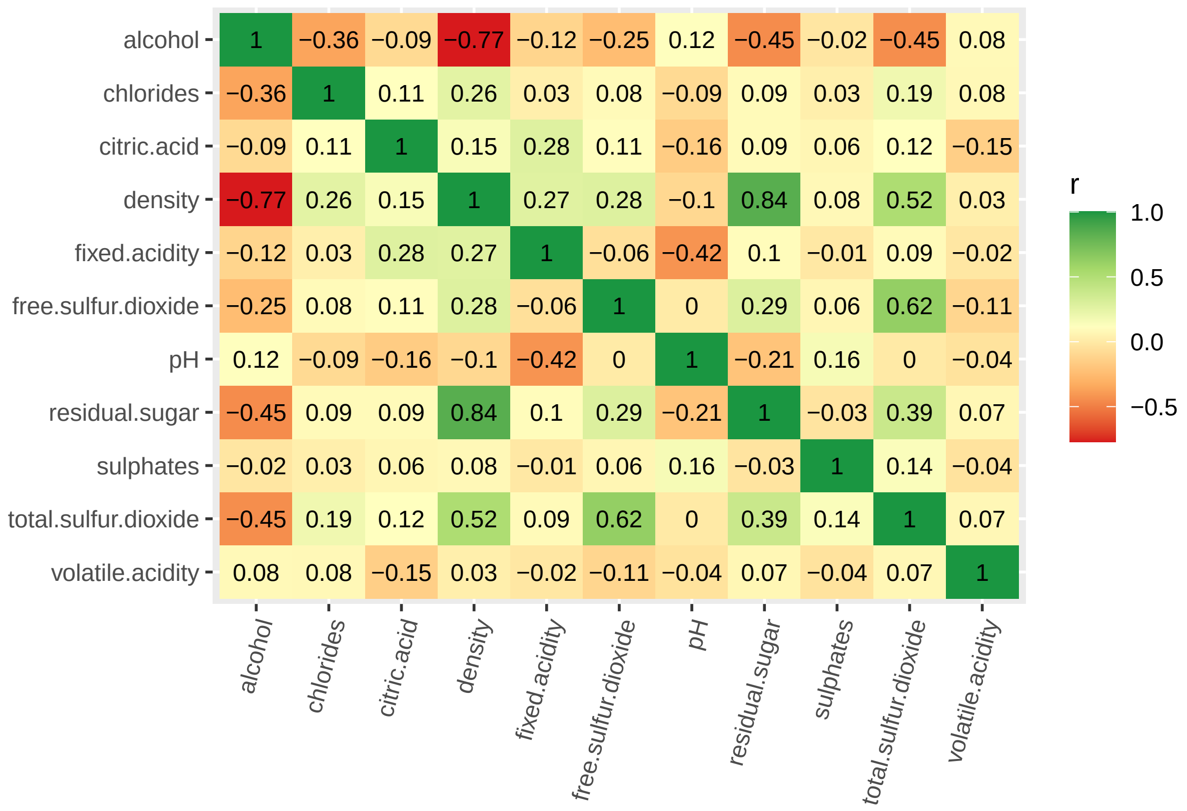
```r
round(cor(train[,-12]), 2)*100
```

```
##                      fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity                  100               -2          28             10
## volatile.acidity                -2              100         -15              7
## citric.acid                     28              -15         100              9
## residual.sugar                  10                7           9            100
## chlorides                        3                8          11              9
## free.sulfur.dioxide             -6              -11          11             29
## total.sulfur.dioxide             9                7          12             39
## density                         27                3          15             84
## pH                             -42               -4         -16            -21
## sulphates                       -1               -4           6             -3
## alcohol                        -12                8          -9            -45
##                      chlorides free.sulfur.dioxide total.sulfur.dioxide density
## fixed.acidity                3                  -6                    9      27
## volatile.acidity             8                 -11                    7       3
## citric.acid                 11                  11                   12      15
## residual.sugar               9                  29                   39      84
## chlorides                  100                   8                   19      26
## free.sulfur.dioxide          8                 100                   62      28
## total.sulfur.dioxide        19                  62                  100      52
## density                     26                  28                   52     100
## pH                          -9                   0                    0     -10
## sulphates                    3                   6                   14       8
## alcohol                    -36                 -25                  -45     -77
##                       pH sulphates alcohol
## fixed.acidity        -42        -1     -12
## volatile.acidity      -4        -4       8
## citric.acid          -16         6      -9
## residual.sugar       -21        -3     -45
## chlorides             -9         3     -36
## free.sulfur.dioxide    0         6     -25
## total.sulfur.dioxide   0        14     -45
## density              -10         8     -77
## pH                   100        16      12
## sulphates             16       100      -2
## alcohol               12        -2     100
```

Visualizing the matrix to spot high correlations may be easier. Here, we use ggplot2 to construct a correlation matrix.

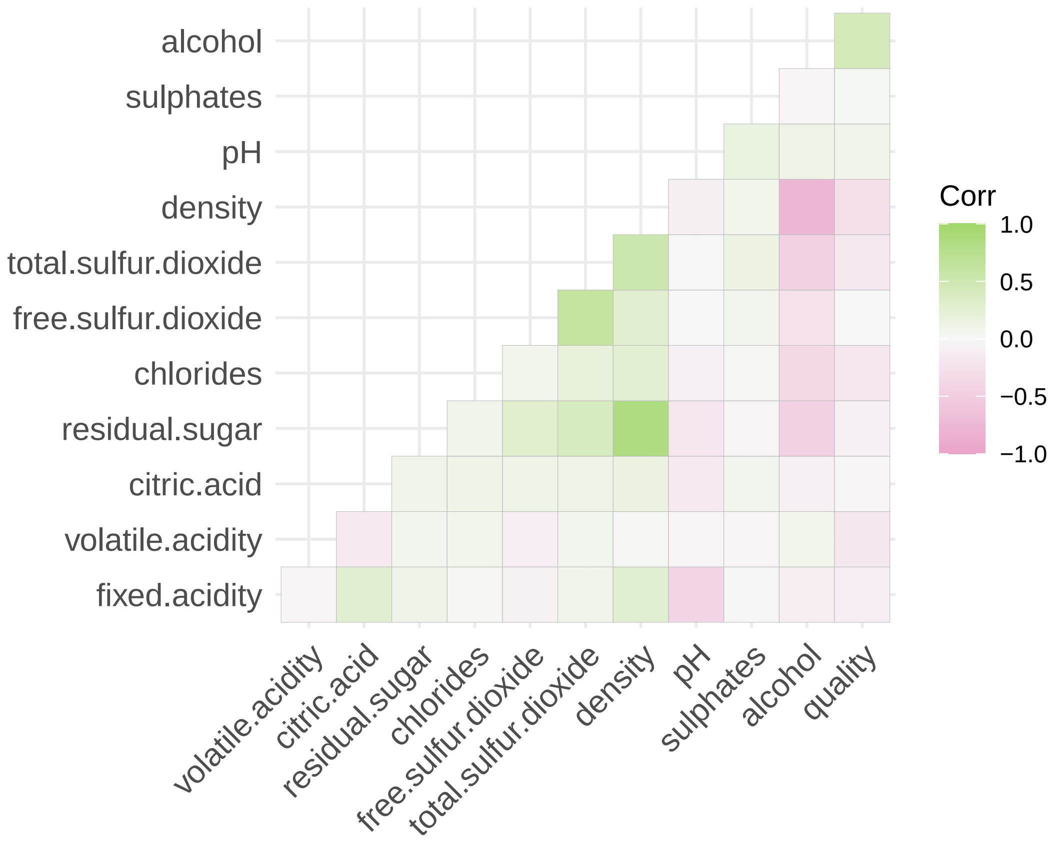
```r
library(tidyr); library(dplyr); library(ggplot2)
corMatrix = as.data.frame(cor(train[,-12]))
corMatrix$var1 = rownames(corMatrix)

corMatrix %>%
  gather(key=var2,value=r,1:11)%>%
  arrange(var1,desc(var2))%>%
  ggplot(aes(x=var1,y=reorder(var2, order(var2,decreasing=F)),fill=r))+
  geom_tile()+
  geom_text(aes(label=round(r,2)),size=3)+
  scale_fill_gradientn(colours = c('#d7191c','#fdae61','#ffffbf','#a6d96a','#1a9641'))+
  theme(axis.text.x=element_text(angle=75,hjust = 1))+xlab('')+ylab('')
```



Alternatively, one can use packages such as `library(ggcorrplot)` or `library(corrplot)` to construct correlation heat maps.

```
library(ggcorrplot)
ggcorrplot(cor(train),
           method = 'square',
           type = 'lower',
           show.diag = F,
           colors = c('#e9a3c9', '#f7f7f7', '#a1d76a'))
```



Multivariate Filter

We will examine relevance and redundancy of predictors of all predictors together. Reason for this is that often what is true in pairwise relationships may not be found when all predictors are considered together. Specifically, predictors found to be relevant and non-redundant based on bivariate correlations may be found to be redundant when effects of other predictors are accounted for. To see if this is the case, we will examine statistical significance of regression coefficients and variance inflating factor.

```
model = lm(quality~.,train)
library(broom)
summary(model) %>%
  tidy()
```

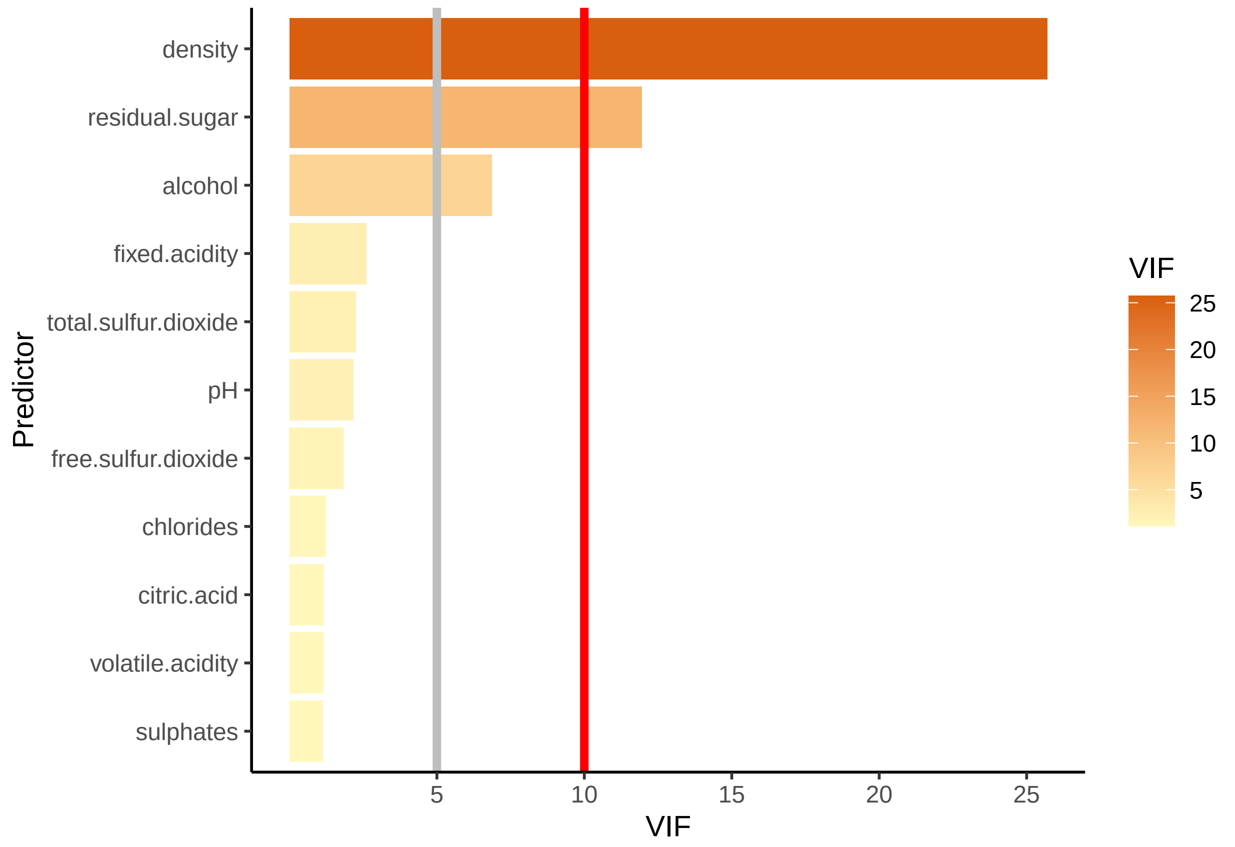
```
## # A tibble: 12 x 5
##    term                  estimate std.error statistic  p.value
##    <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)          141.       21.5         6.59   5.17e-11
##  2 fixed.acidity          0.0584    0.0248      2.35   1.86e- 2
##  3 volatile.acidity      -1.87      0.137     -13.6    3.15e-41
##  4 citric.acid           -0.00233   0.115      -0.0202 9.84e- 1
##  5 residual.sugar         0.0807    0.00880     9.17   7.84e-20
##  6 chlorides             -0.623     0.657      -0.949  3.42e- 1
##  7 free.sulfur.dioxide    0.00369   0.00102     3.60   3.20e- 4
##  8 total.sulfur.dioxide  -0.000318  0.000459   -0.692  4.89e- 1
##  9 density             -141.       21.8        -6.48   1.02e-10
## 10 pH                     0.640     0.126       5.07   4.10e- 7
## 11 sulphates              0.571     0.122       4.69   2.85e- 6
## 12 alcohol                    0.206     0.0278      7.41   1.60e-13
```

Threat of collinearity can also come from linear relationships between sets of variables. One way to assess the threat of multicollinearity in a linear regression is to compute the Variance Inflating Factor (VIF). 1<VIF<Inf. VIF>10 indicates seious multicollinearity while 5<VIF<10 may warrant examination.

```
library(car)
vif(model)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##             2.607386             1.149422             1.157102
##       residual.sugar            chlorides  free.sulfur.dioxide
##            11.948075             1.232863             1.831833
## total.sulfur.dioxide              density                   pH
##             2.248880            25.696470             2.168300
##            sulphates              alcohol
##             1.136248             6.864297
```

```
data.frame(Predictor = names(vif(model)), VIF = vif(model)) %>%
  ggplot(aes(x=VIF, y = reorder(Predictor, VIF), fill=VIF))+
  geom_col()+
  geom_vline(xintercept=5, color = 'gray', size = 1.5)+
  geom_vline(xintercept = 10, color = 'red', size = 1.5)+
  scale_fill_gradient(low = '#fff7bc', high = '#d95f0e')+
  scale_y_discrete(name = "Predictor")+
  scale_x_continuous(breaks = seq(5,30,5))+
  theme_classic()
```



Subset Selection

In this section, we will examine four subset selection methods:
1. Best subset selection
2. Forward
3. Backward
4. Stepwise

Models will be evaluated by estimating test error through a penalty to training error to account for bias due to overfitting. Some of the popular indices that do this are:
a. AIC: Akaike Information Criterion
b. BIC: Bayesian Information Criterion
c. Mallow's Cp
d. Adjusted R2

Best Subset Selection

This method can be summed up by the following steps:
1. Consider all possible subsets of p predictors
2. Fit a model with each subset
3. Pick the best performing model

`regsubsets` from `library(leaps)` can be used to examine all possible subsets. Models are evaluated with BIC, Mallows Cp and adjusted R2. This process can be slow and computationally intensive. Since we have 11 predictors, regsubsets() will generate 11 models to reflect the best combination for every set (e.g., one predictor, two predictors,..).

```
# install.packages('leaps')
library(leaps)
subsets = regsubsets(quality~.,data=train, nvmax=11)
summary(subsets)
```

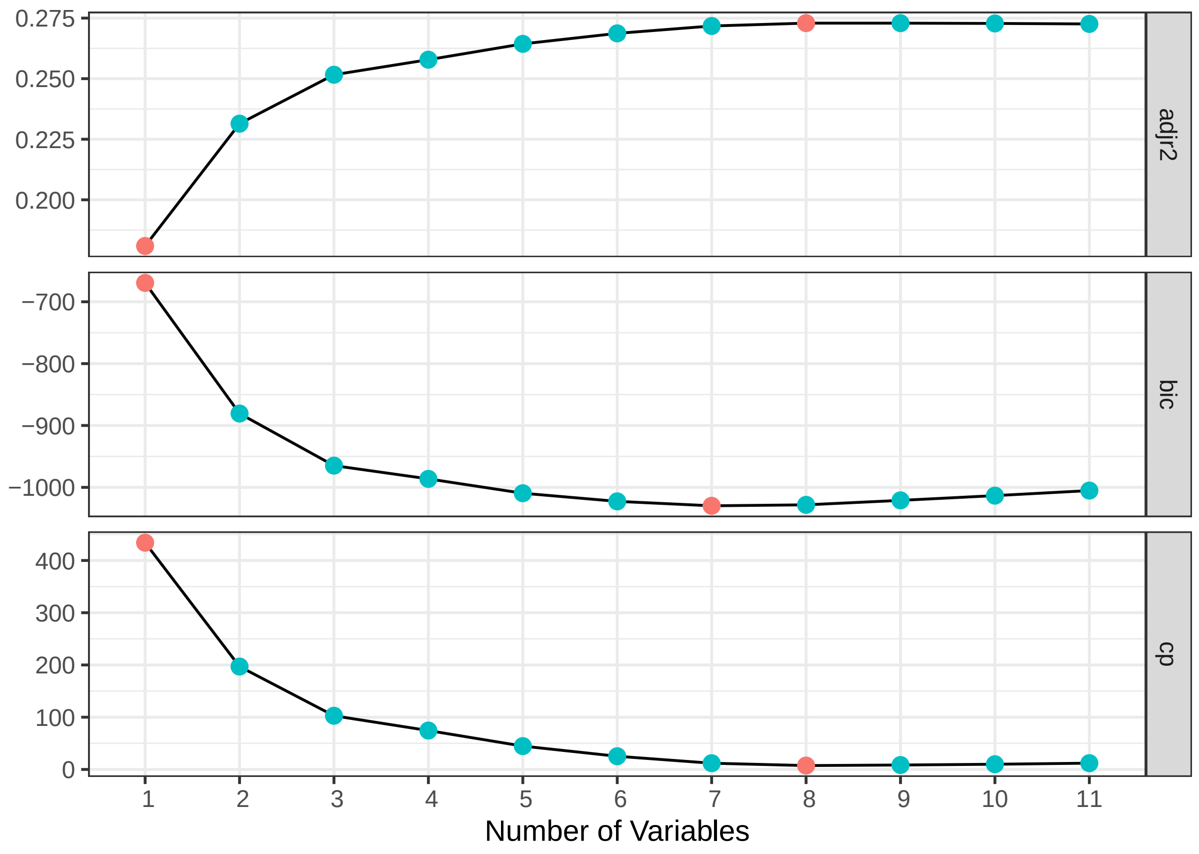
```
## Subset selection object
## Call: regsubsets.formula(quality ~ ., data = train, nvmax = 11)
## 11 Variables  (and intercept)
##                      Forced in Forced out
## fixed.acidity            FALSE      FALSE
## volatile.acidity         FALSE      FALSE
## citric.acid              FALSE      FALSE
## residual.sugar           FALSE      FALSE
## chlorides                FALSE      FALSE
## free.sulfur.dioxide      FALSE      FALSE
## total.sulfur.dioxide     FALSE      FALSE
## density                  FALSE      FALSE
## pH                       FALSE      FALSE
## sulphates                FALSE      FALSE
## alcohol                  FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
##           fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 )  " "           " "              " "         " "            " "
## 2  ( 1 )  " "           "*"              " "         " "            " "
## 3  ( 1 )  " "           "*"              " "         "*"            " "
## 4  ( 1 )  " "           "*"              " "         "*"            " "
## 5  ( 1 )  " "           "*"              " "         "*"            " "
## 6  ( 1 )  " "           "*"              " "         "*"            " "
## 7  ( 1 )  " "           "*"              " "         "*"            " "
## 8  ( 1 )  "*"           "*"              " "         "*"            " "
## 9  ( 1 )  "*"           "*"              " "         "*"            "*"
## 10  ( 1 ) "*"           "*"              " "         "*"            "*"
## 11  ( 1 ) "*"           "*"              "*"         "*"            "*"
##           free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates
## 1  ( 1 )  " "                 " "                  " "     " " " "
## 2  ( 1 )  " "                 " "                  " "     " " " "
## 3  ( 1 )  " "                 " "                  " "     " " " "
## 4  ( 1 )  " "                 " "                  "*"     " " " "
## 5  ( 1 )  " "                 " "                  "*"     "*" " "
## 6  ( 1 )  " "                 " "                  "*"     "*" "*"
## 7  ( 1 )  "*"                 " "                  "*"     "*" "*"
## 8  ( 1 )  "*"                 " "                  "*"     "*" "*"
## 9  ( 1 )  "*"                 " "                  "*"     "*" "*"
## 10  ( 1 ) "*"                 "*"                  "*"     "*" "*"
## 11  ( 1 ) "*"                 "*"                  "*"     "*" "*"
##           alcohol
## 1  ( 1 )  "*"
## 2  ( 1 )  "*"
## 3  ( 1 )  "*"
## 4  ( 1 )  "*"
## 5  ( 1 )  "*"
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
```

Based on Cp and Adj R2, 8-variables should be used. Based on BIC, 7-variables should be used.

```
#names(summary(subsets))
subsets_measures = data.frame(model=1:length(summary(subsets)$cp),
                              cp=summary(subsets)$cp,
                              bic=summary(subsets)$bic,
                              adjr2=summary(subsets)$adjr2)
subsets_measures
```

```
##    model        cp        bic     adjr2
## 1      1 434.01423  -669.5131 0.1809336
## 2      2 196.95206  -880.6710 0.2314236
## 3      3 102.77606  -964.8674 0.2516138
## 4      4  74.49801  -986.3182 0.2578241
## 5      5  44.71318 -1009.5186 0.2643580
## 6      6  25.27440 -1022.6803 0.2686978
## 7      7  11.97473 -1029.8232 0.2717355
## 8      8   7.41169 -1028.2597 0.2729177
## 9      9   8.48057 -1021.0532 0.2729032
## 10    10  10.00041 -1013.3944 0.2727927
## 11    11  12.00000 -1005.2542 0.2725801
```

```
library(ggplot2)
library(tidyr)
subsets_measures %>%
  gather(key = type, value=value, 2:4)%>%
  group_by(type)%>%
  mutate(best_value = factor(ifelse(value == min(value) | value== max(value),0,1)))%>%
  ungroup()%>%
  ggplot(aes(x=model,y=value))+
  geom_line(color='gray2')+
  geom_point(aes(color = best_value), size=2.5)+
  scale_x_discrete(limits = seq(1,11,1),name = 'Number of Variables')+
  scale_y_continuous(name = '')+
  guides(color=F)+
  theme_bw()+
  facet_grid(type~.,scales='free_y')
```



The 8 variables including in the model with the lowest Cp

```
#which.min(summary(subsets)$cp)
coef(subsets,which.min(summary(subsets)$cp))
```

```
##         (Intercept)       fixed.acidity    volatile.acidity      residual.sugar
##        1.476107e+02        6.241702e-02       -1.902335e+00        8.325004e-02
## free.sulfur.dioxide             density                  pH           sulphates
##        3.226972e-03       -1.477110e+02        6.625100e-01        5.665392e-01
##             alcohol
##        2.058946e-01
```

Forward Selection

1. Forward selection method begins with a model containing no predictors and then adds predictors to the model, one at a time

2. Variables are added in order of the marginal improvement to the model. At each stage, the variable that gives the largest marginal improvement is added

3. Addition of variables stops when marginal improvement is not significant

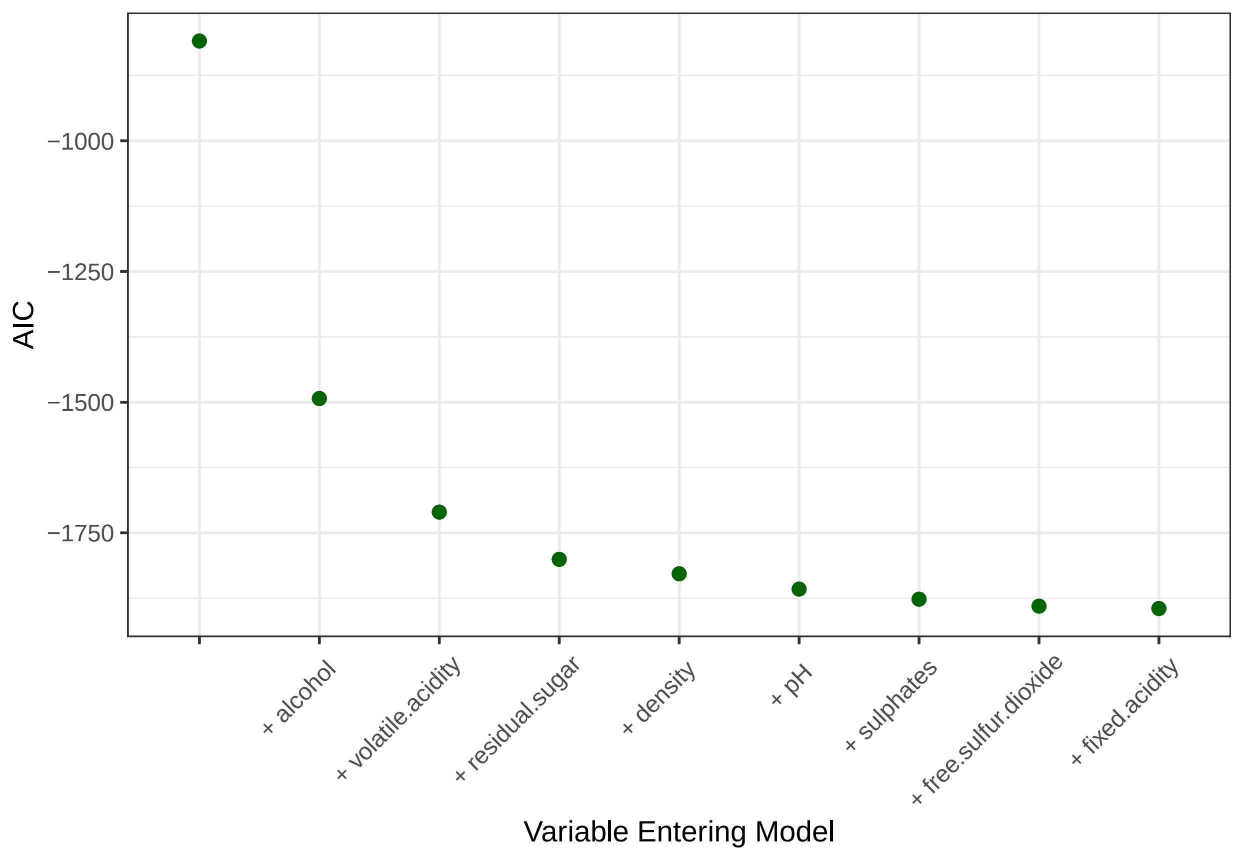
To implement this method we provide `step()` with a null model, a full model, and a model to start with (in this case, also null) and specify direction as forward. Results show variables being added successively until marginal improvement in AIC is no longer significant.

```
start_mod = lm(quality~1,data=train)
empty_mod = lm(quality~1,data=train)
full_mod = lm(quality~.,data=train)
forwardStepwise = step(start_mod,
                       scope=list(upper=full_mod,lower=empty_mod),
                       direction='forward')
```

```
## Start:  AIC=-809.17
## quality ~ 1
##
##                        Df Sum of Sq    RSS      AIC
## + alcohol               1    490.72 2217.9 -1492.97
## + density               1    237.39 2471.2 -1121.88
## + chlorides             1    128.11 2580.5  -973.42
## + volatile.acidity      1     97.44 2611.1  -932.88
## + total.sulfur.dioxide  1     77.93 2630.7  -907.34
## + fixed.acidity         1     34.46 2674.1  -851.11
## + pH                    1     23.88 2684.7  -837.56
## + residual.sugar        1     21.78 2686.8  -834.87
## + sulphates             1      5.00 2703.6  -813.51
## <none>                              2708.6  -809.17
## + citric.acid           1      1.06 2707.5  -808.51
## + free.sulfur.dioxide   1      0.45 2708.1  -807.74
##
## Step:  AIC=-1492.97
## quality ~ alcohol
##
##                        Df Sum of Sq    RSS     AIC
## + volatile.acidity      1   137.323 2080.5 -1710.3
## + free.sulfur.dioxide   1    40.745 2177.1 -1554.6
## + residual.sugar        1    35.873 2182.0 -1546.9
## + chlorides             1    12.840 2205.0 -1510.9
## + fixed.acidity         1    10.890 2207.0 -1507.9
## + sulphates             1     7.240 2210.6 -1502.2
## + density               1     6.874 2211.0 -1501.6
## + pH                    1     4.847 2213.0 -1498.5
## + total.sulfur.dioxide  1     1.453 2216.4 -1493.2
## <none>                              2217.9 -1493.0
## + citric.acid           1     0.870 2217.0 -1492.3
##
## Step:  AIC=-1710.27
## quality ~ alcohol + volatile.acidity
##
##                        Df Sum of Sq    RSS     AIC
## + residual.sugar        1    55.246 2025.3 -1800.6
## + free.sulfur.dioxide   1    28.329 2052.2 -1755.3
## + density               1    18.912 2061.6 -1739.6
## + fixed.acidity         1    11.332 2069.2 -1727.0
## + total.sulfur.dioxide  1     7.136 2073.4 -1720.0
## + chlorides             1     5.171 2075.4 -1716.8
## + sulphates             1     5.095 2075.4 -1716.7
## + pH                    1     2.550 2078.0 -1712.5
## <none>                             2080.5 -1710.3
## + citric.acid           1     0.524 2080.0 -1709.1
##
## Step:  AIC=-1800.6
## quality ~ alcohol + volatile.acidity + residual.sugar
##
##                        Df Sum of Sq    RSS     AIC
## + density               1   17.3925 2007.9 -1828.2
## + free.sulfur.dioxide   1   14.4222 2010.9 -1823.1
## + fixed.acidity         1   14.0056 2011.3 -1822.4
## + pH                    1    8.3002 2017.0 -1812.7
## + sulphates             1    6.5667 2018.7 -1809.8
## + chlorides             1    2.2830 2023.0 -1802.5
## + citric.acid           1    1.5557 2023.7 -1801.2
## <none>                              2025.3 -1800.6
## + total.sulfur.dioxide  1    1.1033 2024.2 -1800.5
##
## Step:  AIC=-1828.19
## quality ~ alcohol + volatile.acidity + residual.sugar + density
##
##                        Df Sum of Sq    RSS     AIC
## + pH                    1   18.2580 1989.7 -1857.5
## + sulphates             1   14.8503 1993.0 -1851.7
## + free.sulfur.dioxide   1   12.9128 1995.0 -1848.3
## + fixed.acidity         1    3.7373 2004.2 -1832.6
## + total.sulfur.dioxide  1    3.7310 2004.2 -1832.6
## + chlorides             1    1.4820 2006.4 -1828.7
## <none>                              2007.9 -1828.2
## + citric.acid           1    0.2422 2007.7 -1826.6
##
## Step:  AIC=-1857.54
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH
##
##                        Df Sum of Sq    RSS     AIC
## + sulphates             1   12.3150 1977.3 -1876.8
## + free.sulfur.dioxide   1   10.4119 1979.2 -1873.5
## + total.sulfur.dioxide  1    2.9155 1986.7 -1860.6
## + fixed.acidity         1    1.5872 1988.1 -1858.3
## <none>                              1989.7 -1857.5
## + chlorides             1    0.7316 1988.9 -1856.8
## + citric.acid           1    0.2167 1989.4 -1855.9
##
## Step:  AIC=-1876.84
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates
##
```

```
##                        Df Sum of Sq    RSS     AIC
## + free.sulfur.dioxide   1    8.7885 1968.5 -1890.1
## + fixed.acidity         1    3.4462 1973.9 -1880.8
## + total.sulfur.dioxide  1    1.7683 1975.6 -1877.9
## <none>                              1977.3 -1876.8
## + chlorides             1    0.8029 1976.5 -1876.2
## + citric.acid           1    0.1303 1977.2 -1875.1
##
## Step:  AIC=-1890.12
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates + free.sulfur.dioxide
##
##                        Df Sum of Sq    RSS     AIC
## + fixed.acidity         1    3.7700 1964.8 -1894.7
## <none>                              1968.5 -1890.1
## + chlorides             1    1.0270 1967.5 -1889.9
## + total.sulfur.dioxide  1    0.3598 1968.2 -1888.8
## + citric.acid           1    0.0075 1968.5 -1888.1
##
## Step:  AIC=-1894.7
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates + free.sulfur.dioxide + fixed.acidity
##
##                        Df Sum of Sq    RSS     AIC
## <none>                              1964.8 -1894.7
## + chlorides             1   0.53486 1964.2 -1893.6
## + total.sulfur.dioxide  1   0.28546 1964.5 -1893.2
## + citric.acid           1   0.00943 1964.8 -1892.7
```

```
forwardStepwise$anova %>%
  mutate(step_number = as.integer(rownames(forwardStepwise$anova))-1) %>%
  mutate(Step = as.character(Step))%>%
  ggplot(aes(x = reorder(Step,X = step_number), y = AIC))+
  geom_point(color = 'darkgreen', size = 2) +
  scale_x_discrete(name = 'Variable Entering Model')+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.9, hjust=0.9))
```



Variables included in the final model.

```
forwardStepwise
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + residual.sugar +
##     density + pH + sulphates + free.sulfur.dioxide + fixed.acidity,
##     data = train)
##
## Coefficients:
##         (Intercept)              alcohol     volatile.acidity
##           1.476e+02            2.059e-01           -1.902e+00
##      residual.sugar              density                   pH
##           8.325e-02           -1.477e+02            6.625e-01
##           sulphates  free.sulfur.dioxide        fixed.acidity
##           5.665e-01            3.227e-03            6.242e-02
```

Backward Selection

a. Backward selection method begins with a model containing all predictors and then removes predictors from the model, one at a time

b. Variables are removed iteratively with the least useful predictor being dropped first

c. Removal of variables stops when elimination of a predictor significantly worsens the model

Similar to Forward Selection, we provide `step()` with the range of models and a model to start with. The models go from full to null to suitably reflect the goal of backward selection. Results show variables being removed successively so long as the marginal AIC doesn't get significantly worse.

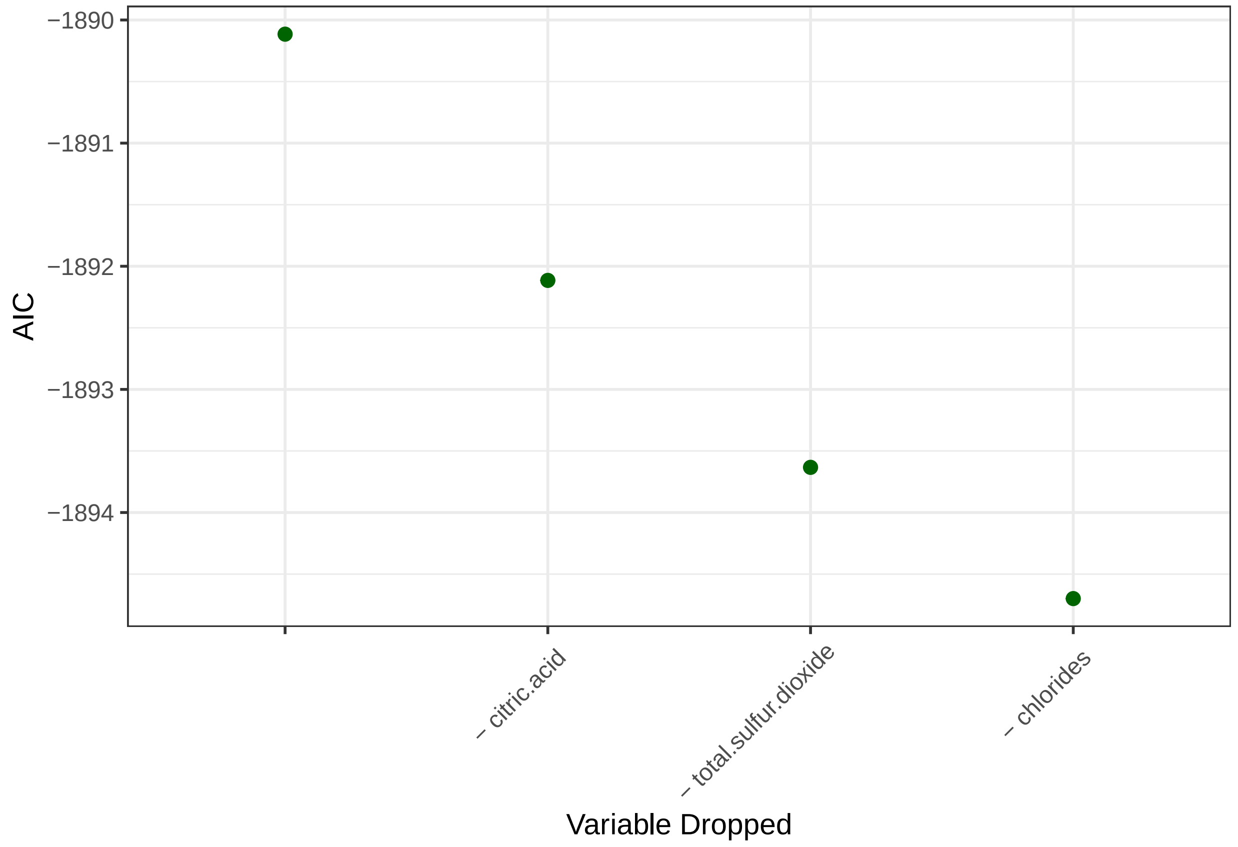
```
start_mod = lm(quality~.,data=train)
empty_mod = lm(quality~1,data=train)
full_mod = lm(quality~.,data=train)
backwardStepwise = step(start_mod,
                        scope=list(upper=full_mod,lower=empty_mod),
                        direction='backward')
```

```
## Start:  AIC=-1890.12
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - citric.acid           1     0.000 1964.0 -1892.1
## - total.sulfur.dioxide  1     0.275 1964.2 -1891.6
## - chlorides             1     0.518 1964.5 -1891.2
## <none>                              1964.0 -1890.1
## - fixed.acidity         1     3.182 1967.1 -1886.6
## - free.sulfur.dioxide   1     7.454 1971.4 -1879.1
## - sulphates             1    12.631 1976.6 -1870.1
## - pH                    1    14.789 1978.8 -1866.4
## - density               1    24.150 1988.1 -1850.2
## - alcohol               1    31.531 1995.5 -1837.5
## - residual.sugar        1    48.320 2012.3 -1808.7
## - volatile.acidity      1   106.700 2070.7 -1710.6
##
## Step:  AIC=-1892.11
## quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - total.sulfur.dioxide  1     0.276 1964.2 -1893.6
## - chlorides             1     0.525 1964.5 -1893.2
## <none>                              1964.0 -1892.1
## - fixed.acidity         1     3.215 1967.2 -1888.5
## - free.sulfur.dioxide   1     7.478 1971.4 -1881.1
## - sulphates             1    12.645 1976.6 -1872.1
## - pH                    1    14.936 1978.9 -1868.1
## - density               1    24.318 1988.3 -1851.9
## - alcohol               1    31.797 1995.8 -1839.0
## - residual.sugar        1    48.521 2012.5 -1810.4
## - volatile.acidity      1   109.649 2073.6 -1707.7
##
## Step:  AIC=-1893.63
## quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     chlorides + free.sulfur.dioxide + density + pH + sulphates +
##     alcohol
##
##                       Df Sum of Sq    RSS     AIC
## - chlorides            1     0.535 1964.8 -1894.7
## <none>                             1964.2 -1893.6
## - fixed.acidity        1     3.278 1967.5 -1889.9
## - free.sulfur.dioxide  1     9.258 1973.5 -1879.5
## - sulphates            1    12.439 1976.7 -1874.0
## - pH                   1    15.053 1979.3 -1869.4
## - density              1    26.374 1990.6 -1849.9
## - alcohol              1    31.651 1995.9 -1840.8
## - residual.sugar       1    50.486 2014.7 -1808.6
## - volatile.acidity     1   115.779 2080.0 -1699.1
##
## Step:  AIC=-1894.7
## quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     free.sulfur.dioxide + density + pH + sulphates + alcohol
##
##                       Df Sum of Sq    RSS     AIC
## <none>                             1964.8 -1894.7
## - fixed.acidity        1     3.770 1968.5 -1890.1
## - free.sulfur.dioxide  1     9.112 1973.9 -1880.8
## - sulphates            1    12.524 1977.3 -1874.9
## - pH                   1    16.462 1981.2 -1868.1
## - density              1    28.486 1993.3 -1847.3
## - alcohol              1    31.836 1996.6 -1841.5
## - residual.sugar       1    54.744 2019.5 -1802.4
## - volatile.acidity     1   118.428 2083.2 -1695.9
```

```
summary(backwardStepwise)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + residual.sugar +
##     free.sulfur.dioxide + density + pH + sulphates + alcohol,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7767 -0.4959 -0.0390  0.4662  3.1081
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.476e+02  2.068e+01   7.139 1.14e-12 ***
## fixed.acidity        6.242e-02  2.436e-02   2.562   0.0104 *
## volatile.acidity    -1.902e+00  1.325e-01 -14.362  < 2e-16 ***
## residual.sugar       8.325e-02  8.526e-03   9.765  < 2e-16 ***
## free.sulfur.dioxide  3.227e-03  8.100e-04   3.984 6.92e-05 ***
## density             -1.477e+02  2.097e+01  -7.044 2.25e-12 ***
## pH                   6.625e-01  1.237e-01   5.355 9.14e-08 ***
## sulphates            5.665e-01  1.213e-01   4.670 3.12e-06 ***
## alcohol              2.059e-01  2.765e-02   7.446 1.21e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7577 on 3422 degrees of freedom
## Multiple R-squared:  0.2746, Adjusted R-squared:  0.2729
## F-statistic: 161.9 on 8 and 3422 DF,  p-value: < 2.2e-16
```

```
backwardStepwise$anova %>%
  mutate(step_number = as.integer(rownames(backwardStepwise$anova))-1) %>%
  mutate(Step = as.character(Step))%>%
  ggplot(aes(x = reorder(Step,X = step_number), y = AIC))+
  geom_point(color = 'darkgreen', size = 2) +
  scale_x_discrete(name = 'Variable Dropped')+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.9, hjust=0.9))
```



Stepwise Variable Selection

a. Forward and Backward methods are run simultaneously

b. Predictors may be added or removed at each stage until the optimal model is reached

c. Stepwise method attempts to more closely mimic benefits of best subsets while retaining the computational advantages of forward and backward selection methods

`step()` function setup is similar to that for forward selection, but direction is specified as both. Results mirror those for Forward Selection as none of the variables added were later dropped.

```
start_mod = lm(quality~1,data=train)
empty_mod = lm(quality~1,data=train)
full_mod = lm(quality~.,data=train)
hybridStepwise = step(start_mod,
                      scope=list(upper=full_mod,lower=empty_mod),
                      direction='both')
```

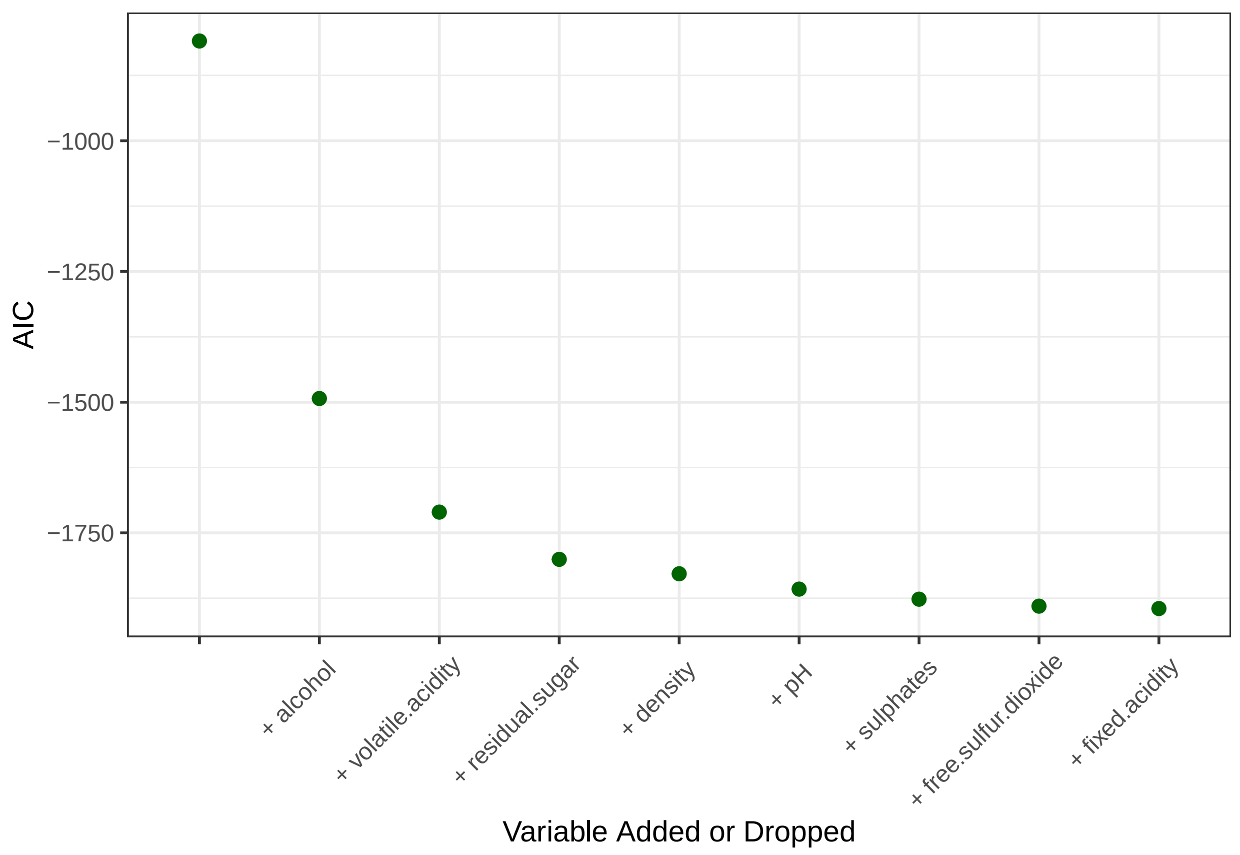
```
## Start:  AIC=-809.17
## quality ~ 1
##
##                        Df Sum of Sq    RSS      AIC
## + alcohol               1    490.72 2217.9 -1492.97
## + density               1    237.39 2471.2 -1121.88
## + chlorides             1    128.11 2580.5  -973.42
## + volatile.acidity      1     97.44 2611.1  -932.88
## + total.sulfur.dioxide  1     77.93 2630.7  -907.34
## + fixed.acidity         1     34.46 2674.1  -851.11
## + pH                    1     23.88 2684.7  -837.56
## + residual.sugar        1     21.78 2686.8  -834.87
## + sulphates             1      5.00 2703.6  -813.51
## <none>                              2708.6  -809.17
## + citric.acid           1      1.06 2707.5  -808.51
## + free.sulfur.dioxide   1      0.45 2708.1  -807.74
##
## Step:  AIC=-1492.97
## quality ~ alcohol
##
##                        Df Sum of Sq    RSS      AIC
## + volatile.acidity      1    137.32 2080.5 -1710.27
## + free.sulfur.dioxide   1     40.74 2177.1 -1554.59
## + residual.sugar        1     35.87 2182.0 -1546.92
## + chlorides             1     12.84 2205.0 -1510.89
## + fixed.acidity         1     10.89 2207.0 -1507.86
## + sulphates             1      7.24 2210.6 -1502.19
## + density               1      6.87 2211.0 -1501.62
## + pH                    1      4.85 2213.0 -1498.47
## + total.sulfur.dioxide  1      1.45 2216.4 -1493.22
## <none>                              2217.9 -1492.97
## + citric.acid           1      0.87 2217.0 -1492.31
## - alcohol               1    490.72 2708.6  -809.17
##
## Step:  AIC=-1710.27
## quality ~ alcohol + volatile.acidity
##
##                        Df Sum of Sq    RSS      AIC
## + residual.sugar        1     55.25 2025.3 -1800.60
## + free.sulfur.dioxide   1     28.33 2052.2 -1755.31
## + density               1     18.91 2061.6 -1739.60
## + fixed.acidity         1     11.33 2069.2 -1727.01
## + total.sulfur.dioxide  1      7.14 2073.4 -1720.05
## + chlorides             1      5.17 2075.4 -1716.80
## + sulphates             1      5.10 2075.4 -1716.68
## + pH                    1      2.55 2078.0 -1712.47
## <none>                              2080.5 -1710.27
## + citric.acid           1      0.52 2080.0 -1709.13
## - volatile.acidity      1    137.32 2217.9 -1492.97
## - alcohol               1    530.60 2611.1  -932.88
##
## Step:  AIC=-1800.6
## quality ~ alcohol + volatile.acidity + residual.sugar
##
##                        Df Sum of Sq    RSS      AIC
## + density               1     17.39 2007.9 -1828.19
## + free.sulfur.dioxide   1     14.42 2010.9 -1823.12
## + fixed.acidity         1     14.01 2011.3 -1822.41
## + pH                    1      8.30 2017.0 -1812.69
## + sulphates             1      6.57 2018.7 -1809.75
## + chlorides             1      2.28 2023.0 -1802.47
## + citric.acid           1      1.56 2023.7 -1801.24
## <none>                              2025.3 -1800.60
## + total.sulfur.dioxide  1      1.10 2024.2 -1800.47
## - residual.sugar        1     55.25 2080.5 -1710.27
## - volatile.acidity      1    156.70 2182.0 -1546.92
## - alcohol               1    569.86 2595.2  -951.95
##
## Step:  AIC=-1828.19
## quality ~ alcohol + volatile.acidity + residual.sugar + density
##
##                        Df Sum of Sq    RSS     AIC
## + pH                    1    18.258 1989.7 -1857.5
## + sulphates             1    14.850 1993.0 -1851.7
## + free.sulfur.dioxide   1    12.913 1995.0 -1848.3
## + fixed.acidity         1     3.737 2004.2 -1832.6
## + total.sulfur.dioxide  1     3.731 2004.2 -1832.6
## + chlorides             1     1.482 2006.4 -1828.7
## <none>                              2007.9 -1828.2
## + citric.acid           1     0.242 2007.7 -1826.6
## - density               1    17.392 2025.3 -1800.6
## - residual.sugar        1    53.726 2061.6 -1739.6
## - alcohol               1   111.757 2119.7 -1644.4
## - volatile.acidity      1   147.206 2155.1 -1587.5
##
## Step:  AIC=-1857.54
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH
##
##                        Df Sum of Sq    RSS     AIC
## + sulphates             1    12.315 1977.3 -1876.8
## + free.sulfur.dioxide   1    10.412 1979.2 -1873.5
## + total.sulfur.dioxide  1     2.915 1986.7 -1860.6
## + fixed.acidity         1     1.587 1988.1 -1858.3
## <none>                              1989.7 -1857.5
## + chlorides             1     0.732 1988.9 -1856.8
## + citric.acid           1     0.217 1989.4 -1855.9
## - pH                    1    18.258 2007.9 -1828.2
## - density               1    27.350 2017.0 -1812.7
## - residual.sugar        1    69.408 2059.1 -1741.9
## - alcohol               1    82.931 2072.6 -1719.4
## - volatile.acidity      1   140.745 2130.4 -1625.0
##
## Step:  AIC=-1876.84
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates
##
##                        Df Sum of Sq    RSS     AIC
## + free.sulfur.dioxide   1     8.789 1968.5 -1890.1
## + fixed.acidity         1     3.446 1973.9 -1880.8
## + total.sulfur.dioxide  1     1.768 1975.6 -1877.9
## <none>                             1977.3 -1876.8
## + chlorides             1     0.803 1976.5 -1876.2
## + citric.acid           1     0.130 1977.2 -1875.1
## - sulphates             1    12.315 1989.7 -1857.5
## - pH                    1    15.723 1993.0 -1851.7
## - density               1    35.091 2012.4 -1818.5
## - alcohol               1    69.377 2046.7 -1760.5
## - residual.sugar        1    79.209 2056.5 -1744.1
## - volatile.acidity      1   136.090 2113.4 -1650.5
##
## Step:  AIC=-1890.12
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates + free.sulfur.dioxide
##
##                        Df Sum of Sq    RSS     AIC
## + fixed.acidity         1     3.770 1964.8 -1894.7
## <none>                             1968.5 -1890.1
## + chlorides             1     1.027 1967.5 -1889.9
## + total.sulfur.dioxide  1     0.360 1968.2 -1888.8
## + citric.acid           1     0.008 1968.5 -1888.1
## - free.sulfur.dioxide   1     8.789 1977.3 -1876.8
## - sulphates             1    10.692 1979.2 -1873.5
## - pH                    1    13.746 1982.3 -1868.2
## - density               1    31.715 2000.3 -1837.3
## - residual.sugar        1    67.110 2035.7 -1777.1
## - alcohol               1    75.154 2043.7 -1763.6
## - volatile.acidity      1   127.262 2095.8 -1677.2
##
## Step:  AIC=-1894.7
## quality ~ alcohol + volatile.acidity + residual.sugar + density +
##     pH + sulphates + free.sulfur.dioxide + fixed.acidity
##
##                        Df Sum of Sq    RSS     AIC
## <none>                             1964.8 -1894.7
## + chlorides             1     0.535 1964.2 -1893.6
## + total.sulfur.dioxide  1     0.285 1964.5 -1893.2
## + citric.acid           1     0.009 1964.8 -1892.7
## - fixed.acidity         1     3.770 1968.5 -1890.1
## - free.sulfur.dioxide   1     9.112 1973.9 -1880.8
## - sulphates             1    12.524 1977.3 -1874.9
## - pH                    1    16.462 1981.2 -1868.1
## - density               1    28.486 1993.3 -1847.3
## - alcohol               1    31.836 1996.6 -1841.5
## - residual.sugar        1    54.744 2019.5 -1802.4
## - volatile.acidity      1   118.428 2083.2 -1695.9
```

```
summary(hybridStepwise)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + residual.sugar +
##     density + pH + sulphates + free.sulfur.dioxide + fixed.acidity,
##     data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7767 -0.4959 -0.0390  0.4662  3.1081
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.476e+02  2.068e+01   7.139 1.14e-12 ***
## alcohol              2.059e-01  2.765e-02   7.446 1.21e-13 ***
## volatile.acidity    -1.902e+00  1.325e-01 -14.362  < 2e-16 ***
## residual.sugar       8.325e-02  8.526e-03   9.765  < 2e-16 ***
## density             -1.477e+02  2.097e+01  -7.044 2.25e-12 ***
## pH                   6.625e-01  1.237e-01   5.355 9.14e-08 ***
## sulphates            5.665e-01  1.213e-01   4.670 3.12e-06 ***
## free.sulfur.dioxide  3.227e-03  8.100e-04   3.984 6.92e-05 ***
## fixed.acidity        6.242e-02  2.436e-02   2.562   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7577 on 3422 degrees of freedom
## Multiple R-squared:  0.2746, Adjusted R-squared:  0.2729
## F-statistic: 161.9 on 8 and 3422 DF,  p-value: < 2.2e-16
```

```
hybridStepwise$anova %>%
  mutate(step_number = as.integer(rownames(hybridStepwise$anova))-1) %>%
  mutate(Step = as.character(Step))%>%
  ggplot(aes(x = reorder(Step,X = step_number), y = AIC))+
  geom_point(color = 'darkgreen', size = 2) +
  scale_x_discrete(name = 'Variable Added or Dropped')+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.9, hjust=0.9))
```



Shrinkage

Shrinkage methods constrain or regularize coefficient estimates, or equivalently, shrink the coefficient estimates towards zero. Based on the shrinkage penalty used, there are two shrinkage methods, Ridge regression and Lasso. The shrinkage penalty in Ridge regression suppresses all coefficients, on the other hand, the penalty in Lasso has the effect of forcing some, not all, coefficient estimates to be exactly zero. By forcing only some coefficient estimate to zero, Lasso in essence performs feature selection.

Ridge Regression

The two shrinkage methods can be implemented using `library(glmnet)`. However, unlike most predictive modeling functions, `glmnet` takes as input a matrix of predictors and a vector of outcomes. We will first explore ridge regression (α=0). `glmnet` constructs a series of models with a set of 100 automatically generated λ.

```
library(glmnet)
x = model.matrix(quality~.-1,data=train)
y = train$quality
set.seed(617)
ridge = glmnet(x = x,
               y = y,
               alpha = 0)
```

Lasso Regression

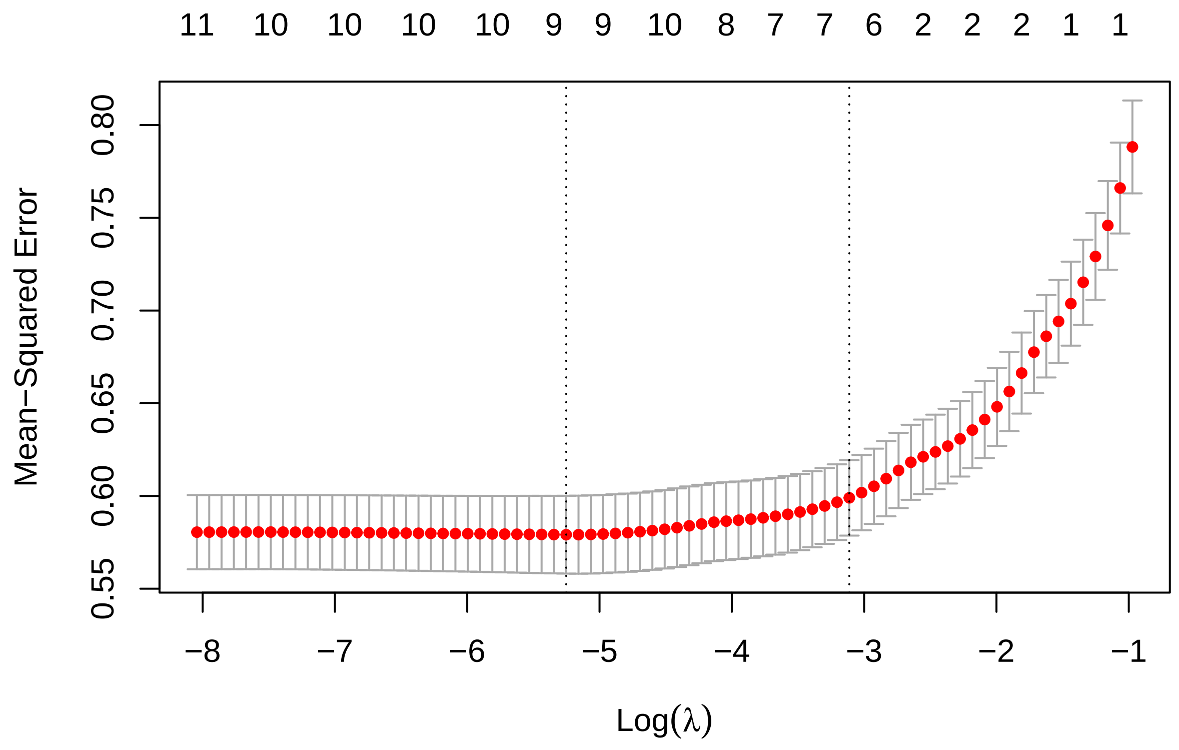
Since the shrinkage penalty of *lasso* forces some coefficient estimates to be exactly zero, it can be used as a feature selection method.

To illustrate, we will repeat what we did above but with α=1. When α=1, `glmnet` runs a *lasso* model. We will also use k-fold cross-validation to generate a set of cross-validation errors for each value of λ. This process is managed under the hood by the `cv.glmnet` function. (By default, this function uses 10-fold cross-validation to generate cross-validation errors for 100 automatically generated lambda values. It is possible to manually specify values of lambda and the folds for cross-validation.)

```
set.seed(617)
cv_lasso = cv.glmnet(x = x,
                     y = y,
                     alpha = 1,
                     type.measure = 'mse')
```

A plot of log-transformed lambda against Mean Squared Error (MSE) indicates that as λ goes up, shrinkage penalty goes up, and therefore number of variables retained goes down. The first dotted line indicates the λ for lowest MSE.

```
plot(cv_lasso)
```



```
cv_lasso
```

```
##
## Call:  cv.glmnet(x = x, y = y, type.measure = "mse", alpha = 1)
##
## Measure: Mean-Squared Error
```

```
##
##      Lambda Index Measure      SE Nonzero
## min 0.00524    47  0.5791 0.02099       9
## 1se 0.04451    24  0.5990 0.02035       6
```

Based on the above, the λ with the lowest mean squared error is

```
cv_lasso$lambda.min
```

```
## [1] 0.005237466
```

Here, we can see 2 coefficients have been forced to exactly zero.

```
coef(cv_lasso, s = cv_lasso$lambda.1se) %>%
  round(4)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                           s1
## (Intercept)           3.1700
## fixed.acidity        -0.0209
## volatile.acidity     -1.5234
## citric.acid           .
## residual.sugar        0.0094
## chlorides            -0.4766
## free.sulfur.dioxide   0.0016
## total.sulfur.dioxide  .
## density               .
## pH                    .
## sulphates             .
## alcohol               0.3025
```

Dimension Reduction

In this approach, p predictors are reduced to a smaller number of components based on a measure of similarity (e.g., correlation). Here, we will examine a popular dimension reduction technique called Principal Components Analysis.

Extract the predictors to be reduced.

```
trainPredictors = train[,-12]
testPredictors = test[,-12]
```

Conduct Principal Components Analysis on train sample. Principal components analysis will always generate as many components as variables. The first few components contain the most amount of variance. One heuristic for number of components to retain is a cumulative variance greater than 70%. In this case, we are extracting only six of eleven components.

```
pca = prcomp(trainPredictors,scale. = T)
train_components = data.frame(cbind(pca$x[,1:6], quality = train$quality))
```

Let's examine the components generated for these six components for the first 6 obs

```
head(train_components)
```

```
##          PC1        PC2         PC3        PC4        PC5        PC6 quality
## 1 -3.6734546  0.5871312 -0.94149256 -1.0689123 -0.4076974  0.8571145       6
## 2  0.6466845 -0.4089912 -0.38008932  1.0328963 -0.7381970  0.2992169       6
## 3 -0.1547839  1.1901776  0.05261354  0.3128794 -0.4129549  0.4344642       6
## 4 -1.4459397 -0.1428400  0.02786548 -0.4598627 -0.3860368 -0.7756376       6
## 5 -1.4459397 -0.1428400  0.02786548 -0.4598627 -0.3860368 -0.7756376       6
## 6 -0.1547839  1.1901776  0.05261354  0.3128794 -0.4129549  0.4344642       6
```

Construct a model using the components derived from the original predictors

```
train_model = lm(quality~.,train_components)
summary(train_model)
```

```
##
## Call:
## lm(formula = quality ~ ., data = train_components)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0214 -0.5601 -0.0145  0.5135  3.2110
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.877587   0.013840 424.684  < 2e-16 ***
## PC1          0.143733   0.007732  18.590  < 2e-16 ***
## PC2         -0.048724   0.011039  -4.414 1.05e-05 ***
## PC3          0.157180   0.012449  12.626  < 2e-16 ***
## PC4         -0.172381   0.013596 -12.678  < 2e-16 ***
## PC5          0.039105   0.014039   2.786  0.00537 **
## PC6          0.030457   0.014313   2.128  0.03341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8107 on 3424 degrees of freedom
## Multiple R-squared:  0.1692, Adjusted R-squared:  0.1678
## F-statistic: 116.3 on 6 and 3424 DF,  p-value: < 2.2e-16
```

Before we can apply the train model to the test set, we need to ensure that we apply the same variable transformations for the train sample on the test sample. Specifically, we need to apply the same component structure from the train sample predictors to the test predictors. Note, we are not running a fresh principal components analysis on the test sample, rather we are imposing the train component structure on the test sample.

```
test_pca = predict(pca,newdata=testPredictors)
test_components = data.frame(cbind(test_pca[,1:6], quality = test$quality))
```

Note, train_components and test_components have the same structure.

```
str(train_components)
```

```
## 'data.frame':    3431 obs. of  7 variables:
##  $ PC1    : num  -3.673 0.647 -0.155 -1.446 -1.446 ...
##  $ PC2    : num  0.587 -0.409 1.19 -0.143 -0.143 ...
##  $ PC3    : num  -0.9415 -0.3801 0.0526 0.0279 0.0279 ...
##  $ PC4    : num  -1.069 1.033 0.313 -0.46 -0.46 ...
##  $ PC5    : num  -0.408 -0.738 -0.413 -0.386 -0.386 ...
##  $ PC6    : num  0.857 0.299 0.434 -0.776 -0.776 ...
##  $ quality: num  6 6 6 6 6 6 6 6 6 5 ...
```

```
str(test_components)
```

```
## 'data.frame':    1467 obs. of  7 variables:
##  $ PC1    : num  0.647 0.972 3.072 1.731 0.547 ...
##  $ PC2    : num  -0.409 0.903 0.122 0.687 -0.912 ...
##  $ PC3    : num  -0.38 0.497 -1.537 -0.545 0.518 ...
##  $ PC4    : num  1.033 0.422 0.469 -0.218 0.453 ...
##  $ PC5    : num  -0.738 1.299 2.395 -0.472 -0.716 ...
##  $ PC6    : num  0.2992 -0.0371 -1.046 -0.5689 -0.4272 ...
##  $ quality: num  6 6 8 7 6 7 6 6 5 5 ...
```

Finally, we evaluate the estimated train model on the test data to assess model performance.

```
pred = predict(train_model,newdata=test_components)
sse = sum((pred-test_components$quality)^2)
sst = sum((mean(train_components$quality) - test_components$quality)^2)
r2_test = 1 - sse/sst
r2_test
```

```
## [1] 0.1885463
```

Summary

To sum up, we implemented five methods for carrying out feature selection

1. Filter methods
2. Subset selection
3. Shrinkage methods
4. Dimensionality Reduction