Homework 5 Regression Methods

Bheeni Garg

May 27, 2016

Question 1:

Step 1: Collecting Data —-

For this analysis, the insurance dataset is used containing medical expenses for patients in the United States. The insurance.csv file includes 1,338 examples of beneficiaries currently enrolled in the insurance plan, with features indicating characteristics of the patient as well as the total medical expenses charged to the plan for the calendar year. The features are: • age: This is an integer indicating the age of the primary beneficiary (excluding those above 64 years, since they are generally covered by the government). • sex: This is the policy holder's gender, either male or female. • bmi: This is the body mass index (BMI), which provides a sense of how over or under-weight a person is relative to their height. BMI is equal to weight (in kilograms) divided by height (in meters) squared. An ideal BMI is within the range of 18.5 to 24.9. • children: This is an integer indicating the number of children / dependents covered by the insurance plan. • smoker: This is yes or no depending on whether the insured regularly smokes tobacco. • region: This is the beneficiary's place of residence in the U.S., divided into four geographic regions: northeast, southeast, southwest, or northwest.

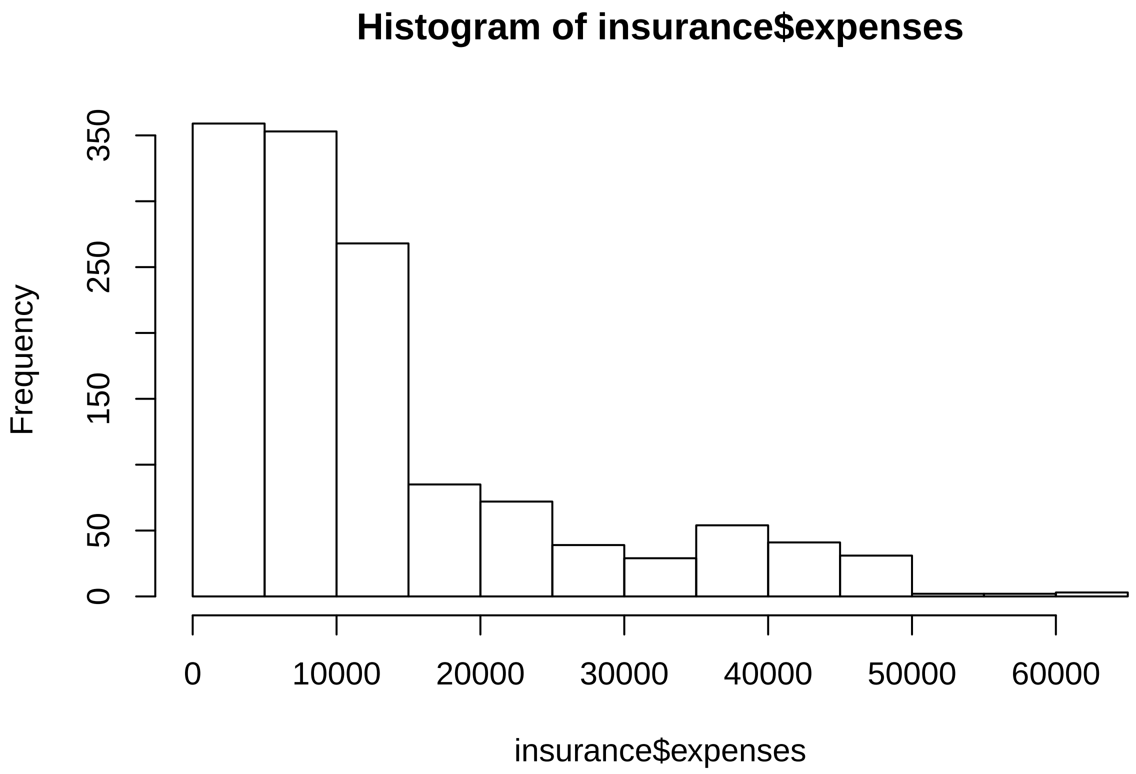
```
## Step 2: Exploring and preparing the data ----
insurance <- read.csv("insurance.csv", stringsAsFactors = TRUE)
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ expenses: num  16885 1726 4449 21984 3867 ...
```

```
# summarize the charges variable
summary(insurance$expenses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1122    4740    9382   13270   16640   63770
```

```
# histogram of insurance charges
hist(insurance$expenses)
```



It can be noticed from the histogram that the majority of individuals have yearly medical expenses in the range of $0-15000.

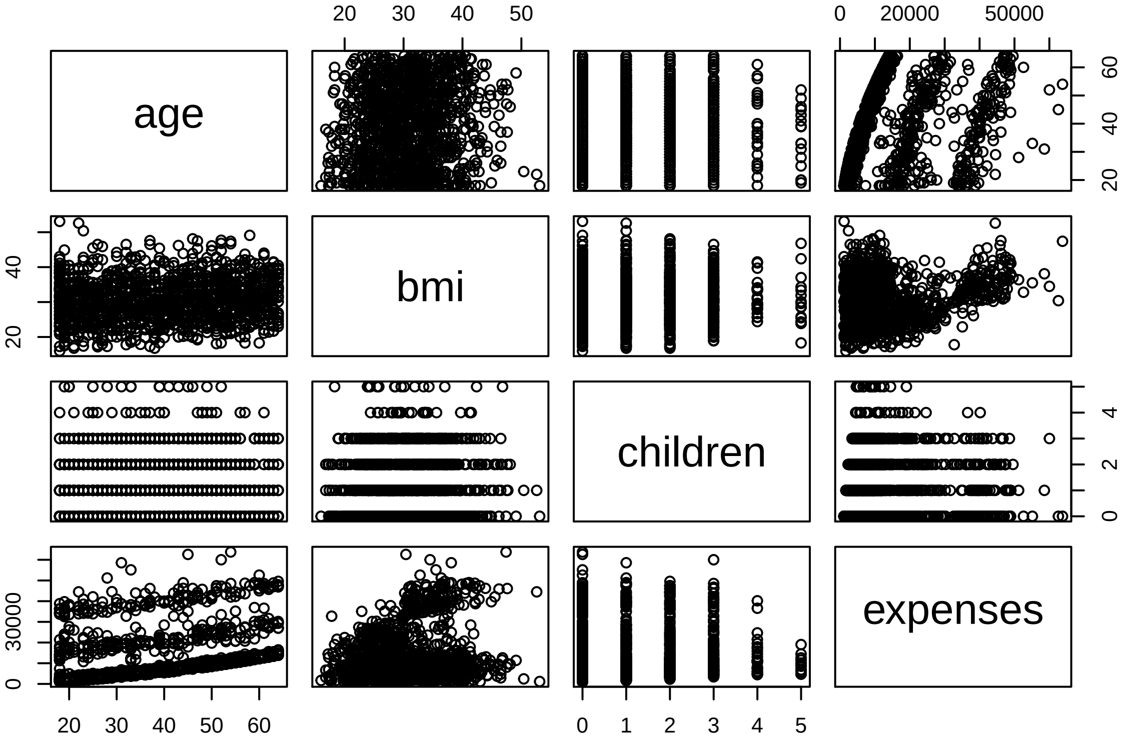
```
# table of region
table(insurance$region)
```

```
##
## northeast northwest southeast southwest
##       324       325       364       325
```

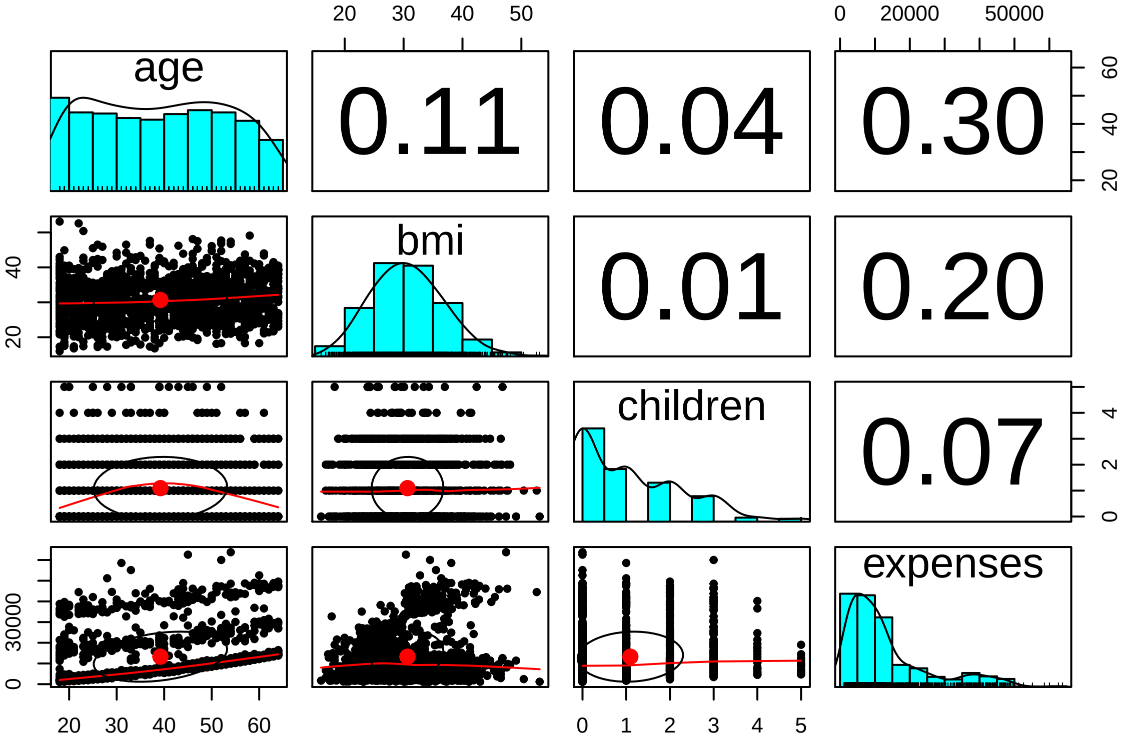
```
# exploring relationships among features: correlation matrix
cor(insurance[c("age", "bmi", "children", "expenses")])
```

```
##                age        bmi   children   expenses
## age      1.0000000 0.10934101 0.04246900 0.29900819
## bmi      0.1093410 1.00000000 0.01264471 0.19857626
## children 0.0424690 0.01264471 1.00000000 0.06799823
## expenses 0.2990082 0.19857626 0.06799823 1.00000000
```

```
# visualing relationships among features: scatterplot matrix
pairs(insurance[c("age", "bmi", "children", "expenses")])
```



```
# more informative scatterplot matrix
library(psych)
pairs.panels(insurance[c("age", "bmi", "children", "expenses")])
```



```r
# splitting the dataset
library(caTools)
set.seed(123)
split <- sample.split(insurance$expenses, SplitRatio = 0.75)
train_insurance <- subset(insurance, split == TRUE)
test_insurance <- subset(insurance, split == FALSE)

summary(train_insurance$expenses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1136    4720    9433   13520   17100   63770 
```

```r
summary(test_insurance$expenses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1122    4839    9288   12520   15090   48680 
```

```r
## Step 3: Training a model on the data ----
ins_model <- lm(expenses ~ age + children + bmi + sex + smoker + region, data = train_insurance)

# see the estimated beta coefficients
ins_model
```

```
## 
## Call:
## lm(formula = expenses ~ age + children + bmi + sex + smoker + 
##     region, data = train_insurance)
## 
## Coefficients:
##     (Intercept)              age         children              bmi  
##        -12480.7            259.9            491.7            362.8  
##         sexmale        smokeryes  regionnorthwest  regionsoutheast  
##          -441.0          24331.3           -393.1          -1162.4  
## regionsouthwest  
##         -1231.4  
```

```r
# see more detail about the estimated beta coefficients
summary(ins_model)
```

```
## 
## Call:
## lm(formula = expenses ~ age + children + bmi + sex + smoker + 
##     region, data = train_insurance)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -11467  -3044  -1020   1712  29621 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     -12480.73    1165.39 -10.709  < 2e-16 ***
## age                259.88      14.11  18.415  < 2e-16 ***
## children           491.69     162.75   3.021  0.00258 **
## bmi                362.83      33.33  10.887  < 2e-16 ***
## sexmale           -440.97     396.73  -1.112  0.26662
## smokeryes        24331.26     488.45  49.813  < 2e-16 ***
## regionnorthwest   -393.12     567.28  -0.693  0.48848
## regionsoutheast  -1162.36     562.89  -2.065  0.03918 *
## regionsouthwest  -1231.36     559.72  -2.200  0.02804 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6223 on 994 degrees of freedom
## Multiple R-squared:  0.7547, Adjusted R-squared:  0.7528
## F-statistic: 382.3 on 8 and 994 DF,  p-value: < 2.2e-16
```

The summary shows that the significant predictors in the model are age, children, bmi, smokeryes, regionsoutheast and regionsouthwest.

The Rsquared value for the training set is 0.7494.

```r
## Step 4: Evaluating model performance ----

insurance_predict <- predict(ins_model, newdata = test_insurance)
head(insurance_predict)
```

```
##         2         4         5         8        11        16
## 3349.1081 3497.4819 5487.1379 8267.2081 3081.4494  201.9686
```

```r
head(test_insurance$expenses)
```

```
## [1]  1725.55 21984.47  3866.86  7281.51  2721.32  1837.24
```

```r
sse <- sum((test_insurance$expenses - insurance_predict)^2)
sse
```

```
## [1] 10485871027
```

```r
sst <- sum((test_insurance$expenses - mean(test_insurance$expenses))^2)
sst
```

```
## [1] 38864498053
```

```r
rsq <- 1 - (sse/sst)
rsq
```

```
## [1] 0.7301941
```

Thus, the Rsquared value for the test data set is 0.7528 which is very close to that of our training model.

Step 5: Improving model performance —-

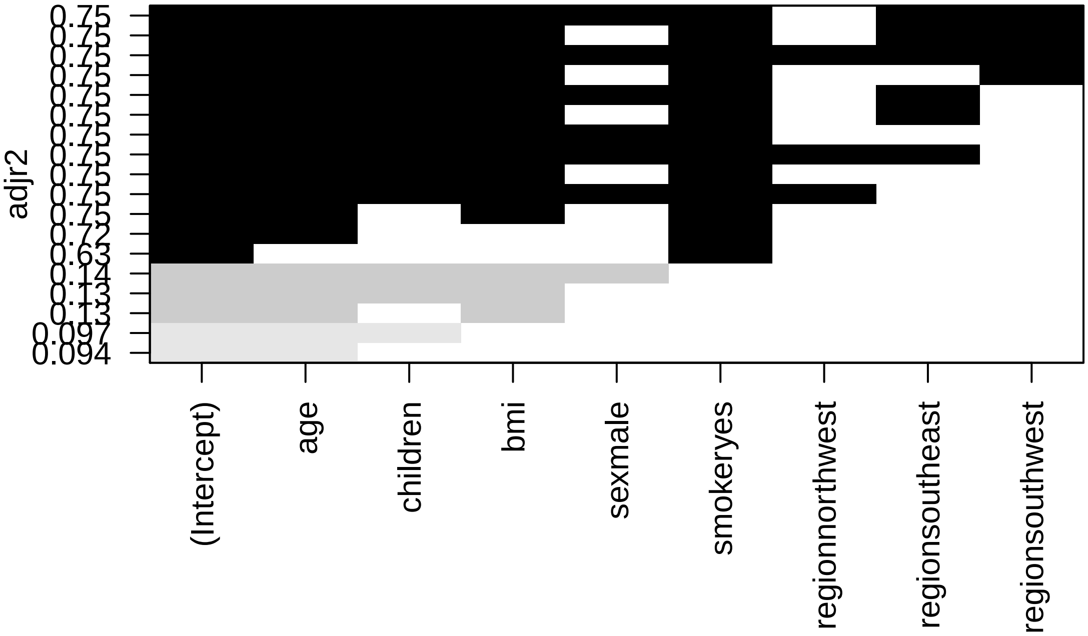
Variable Selection

With a model with large number of predictors, it is imperative to select the ones which contribute the most. One of the methods of variable selection is backward selection demonstrated below:

```r
library(leaps)
back <- regsubsets(expenses ~ age + children + bmi + sex + smoker + region,
    data = train_insurance, method = "backward", nbest = 3)
summary(back)
```

```
## Subset selection object
## Call: regsubsets.formula(expenses ~ age + children + bmi + sex + smoker +
##     region, data = train_insurance, method = "backward", nbest = 3)
## 8 Variables  (and intercept)
##                  Forced in Forced out
## age                  FALSE      FALSE
## children             FALSE      FALSE
## bmi                  FALSE      FALSE
## sexmale              FALSE      FALSE
## smokeryes            FALSE      FALSE
## regionnorthwest      FALSE      FALSE
## regionsoutheast      FALSE      FALSE
## regionsouthwest      FALSE      FALSE
## 3 subsets of each size up to 8
## Selection Algorithm: backward
##          age children bmi sexmale smokeryes regionnorthwest
## 1  ( 1 ) " " " "      " " " "     "*"       " "
## 1  ( 2 ) "*" " "      " " " "     " "       " "
## 2  ( 1 ) "*" " "      " " " "     "*"       " "
## 2  ( 2 ) "*" " "      "*" " "     " "       " "
## 2  ( 3 ) "*" "*"      " " " "     " "       " "
## 3  ( 1 ) "*" " "      "*" " "     "*"       " "
## 3  ( 2 ) "*" "*"      "*" " "     " "       " "
## 4  ( 1 ) "*" "*"      "*" " "     "*"       " "
## 4  ( 2 ) "*" "*"      "*" "*"     " "       " "
## 5  ( 1 ) "*" "*"      "*" " "     "*"       " "
## 5  ( 2 ) "*" "*"      "*" " "     "*"       " "
## 5  ( 3 ) "*" "*"      "*" "*"     "*"       " "
## 6  ( 1 ) "*" "*"      "*" " "     "*"       " "
## 6  ( 2 ) "*" "*"      "*" "*"     "*"       " "
## 6  ( 3 ) "*" "*"      "*" "*"     "*"       "*"
## 7  ( 1 ) "*" "*"      "*" "*"     "*"       " "
## 7  ( 2 ) "*" "*"      "*" "*"     "*"       "*"
## 8  ( 1 ) "*" "*"      "*" "*"     "*"       "*"
##          regionsoutheast regionsouthwest
## 1  ( 1 ) " "             " "
## 1  ( 2 ) " "             " "
## 2  ( 1 ) " "             " "
## 2  ( 2 ) " "             " "
## 2  ( 3 ) " "             " "
## 3  ( 1 ) " "             " "
## 3  ( 2 ) " "              " "
## 4  ( 1 ) " "              " "
## 4  ( 2 ) " "              " "
## 5  ( 1 ) " "              "*"
## 5  ( 2 ) "*"              " "
## 5  ( 3 ) " "              " "
## 6  ( 1 ) "*"              "*"
## 6  ( 2 ) "*"              " "
## 6  ( 3 ) " "              " "
## 7  ( 1 ) "*"              "*"
## 7  ( 2 ) "*"              " "
## 8  ( 1 ) "*"              "*"
```

```
plot(back, scale = "adjr2")
```



The plot shows variables that are significant in black while the ones which are not significant are in white. This method uses adjusted Rsquared as the selection criteria i.e. the subset of variables that lead to the largest value of adjusted Rsquared is the best subset. Backward selection starts with all the variables in the model and throws out the ones which contribute not as much or don't contribute at all. The remaining significant variables are finally listed in the output.

It can be noticed that the best subset of variables selected by the backward selection method include age, children, bmi, smokeyes, regionnorthwest, regionsoutheast and regionsouthwest.

```
# add a higher-order 'age' term
insurance$age2 <- insurance$age^2

# add an indicator for BMI >= 30
insurance$bmi30 <- ifelse(insurance$bmi >= 30, 1, 0)

# create final model
ins_model2 <- lm(expenses ~ age + age2 + children + bmi + sex + bmi30 * smoker +
    region, data = insurance)
```

```r
summary(ins_model2)
```

```
##
## Call:
## lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 *
##     smoker + region, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17297.1  -1656.0  -1262.7   -727.8  24161.6
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       139.0053  1363.1359   0.102 0.918792
## age               -32.6181    59.8250  -0.545 0.585690
## age2                3.7307     0.7463   4.999 6.54e-07 ***
## children          678.6017   105.8855   6.409 2.03e-10 ***
## bmi               119.7715    34.2796   3.494 0.000492 ***
## sexmale          -496.7690   244.3713  -2.033 0.042267 *
## bmi30            -997.9355   422.9607  -2.359 0.018449 *
## smokeryes       13404.5952   439.9591  30.468  < 2e-16 ***
## regionnorthwest  -279.1661   349.2826  -0.799 0.424285
## regionsoutheast  -828.0345   351.6484  -2.355 0.018682 *
## regionsouthwest -1222.1619   350.5314  -3.487 0.000505 ***
## bmi30:smokeryes 19810.1534   604.6769  32.762  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4445 on 1326 degrees of freedom
## Multiple R-squared:  0.8664, Adjusted R-squared:  0.8653
## F-statistic: 781.7 on 11 and 1326 DF,  p-value: < 2.2e-16
```

A model could also be improved by adding higher order terms, transforming variables, adding interaction terms.

Question 2: Code from Chapter 3- Introduction to Statistical Learning

```r
library(MASS)
library(ISLR)

# Simple Linear Regression

attach(Boston)
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
lm.fit = lm(medv ~ lstat)
lm.fit
```

```
## 
## Call:
## lm(formula = medv ~ lstat)
## 
## Coefficients:
## (Intercept)        lstat  
##       34.55        -0.95
```

```
summary(lm.fit)
```

```
## 
## Call:
## lm(formula = medv ~ lstat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -15.168  -3.990  -1.318   2.034  24.500 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432 
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"         
##  [5] "fitted.values" "assign"        "qr"            "df.residual"  
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(lm.fit)
```

```
## (Intercept)       lstat 
##  34.5538409  -0.9500494
```

```
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

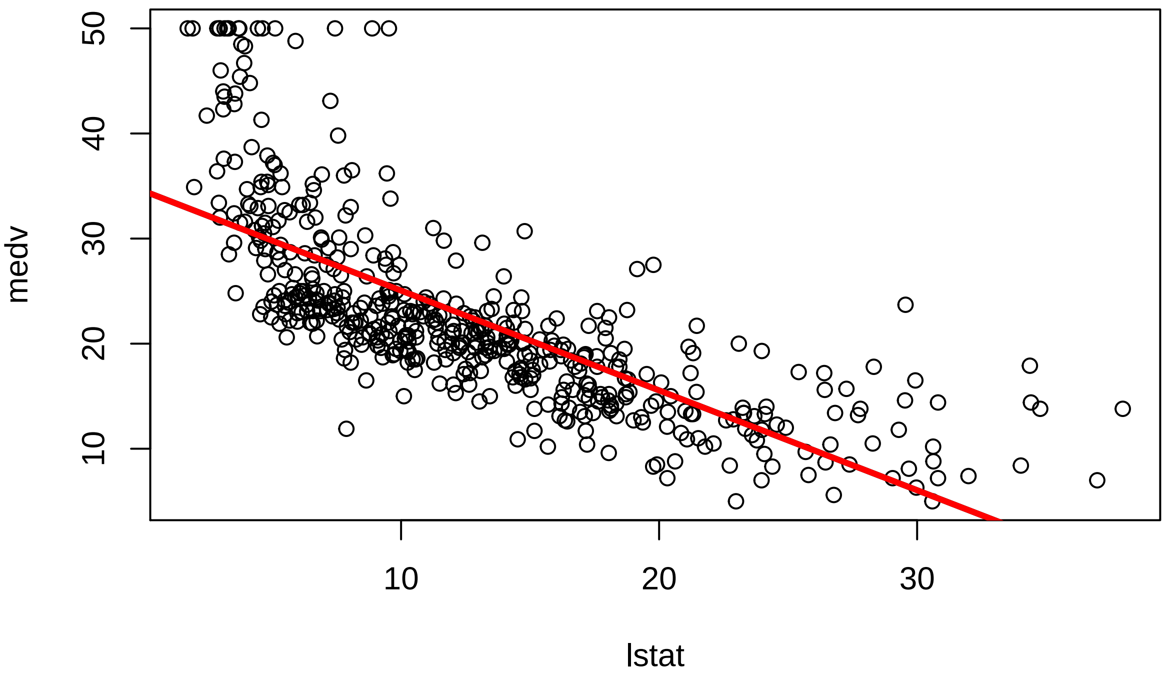
```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))), interval = "confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

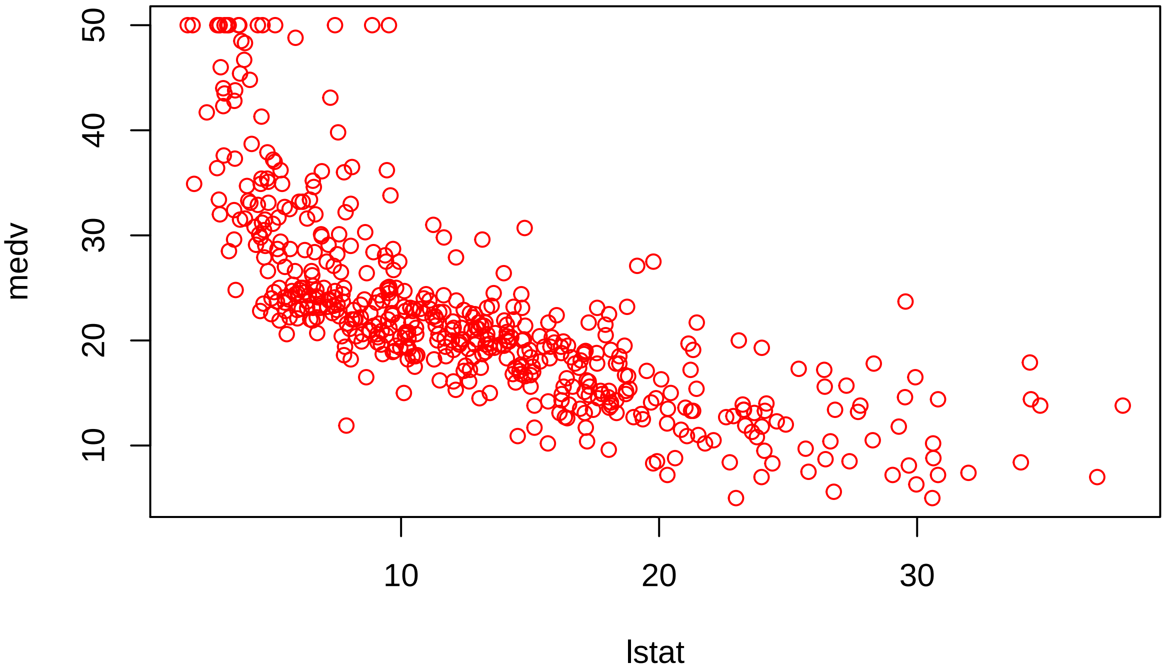
```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))), interval = "prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

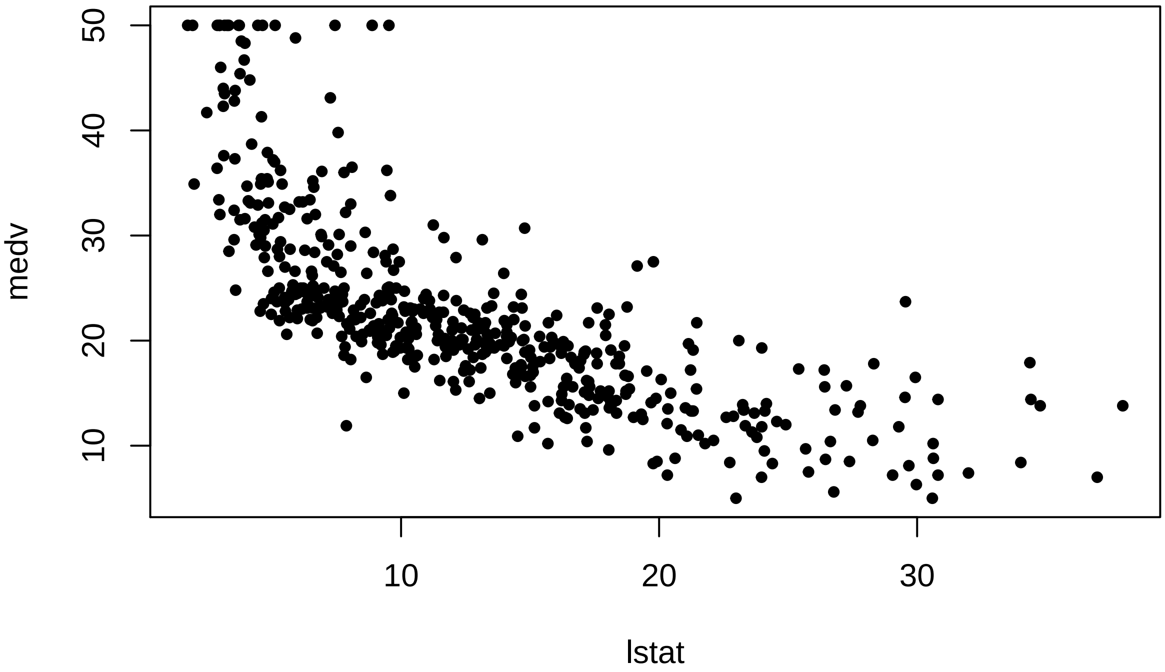
```
plot(lstat, medv)
abline(lm.fit)
abline(lm.fit, lwd = 3)
abline(lm.fit, lwd = 3, col = "red")
```



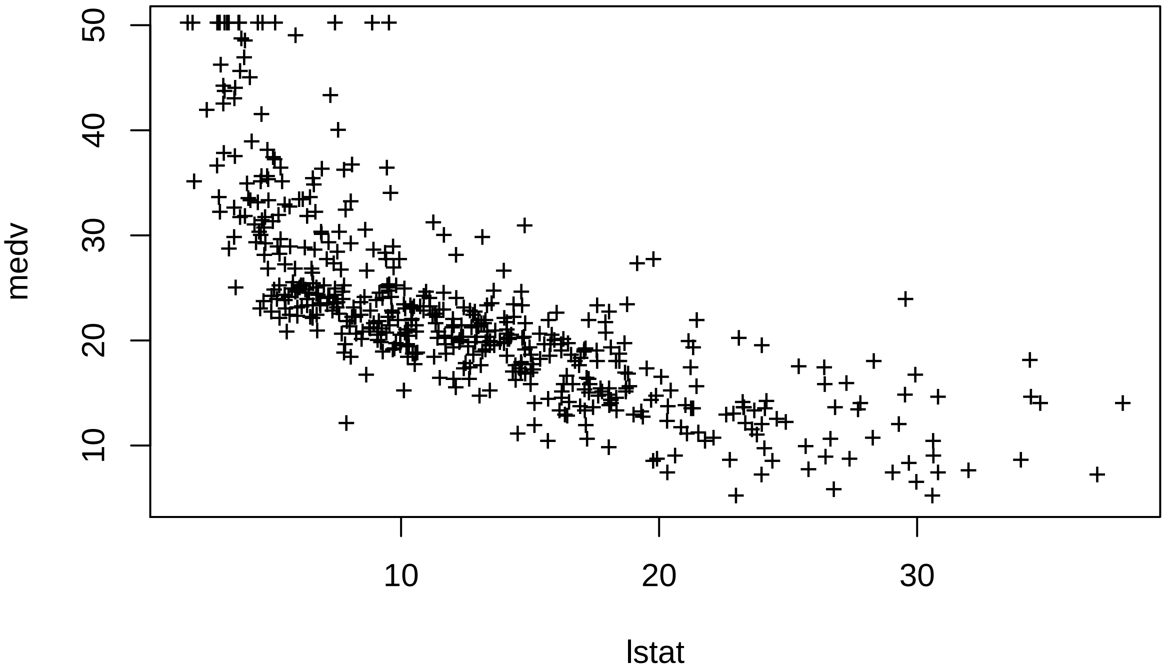
```
plot(lstat, medv, col = "red")
```



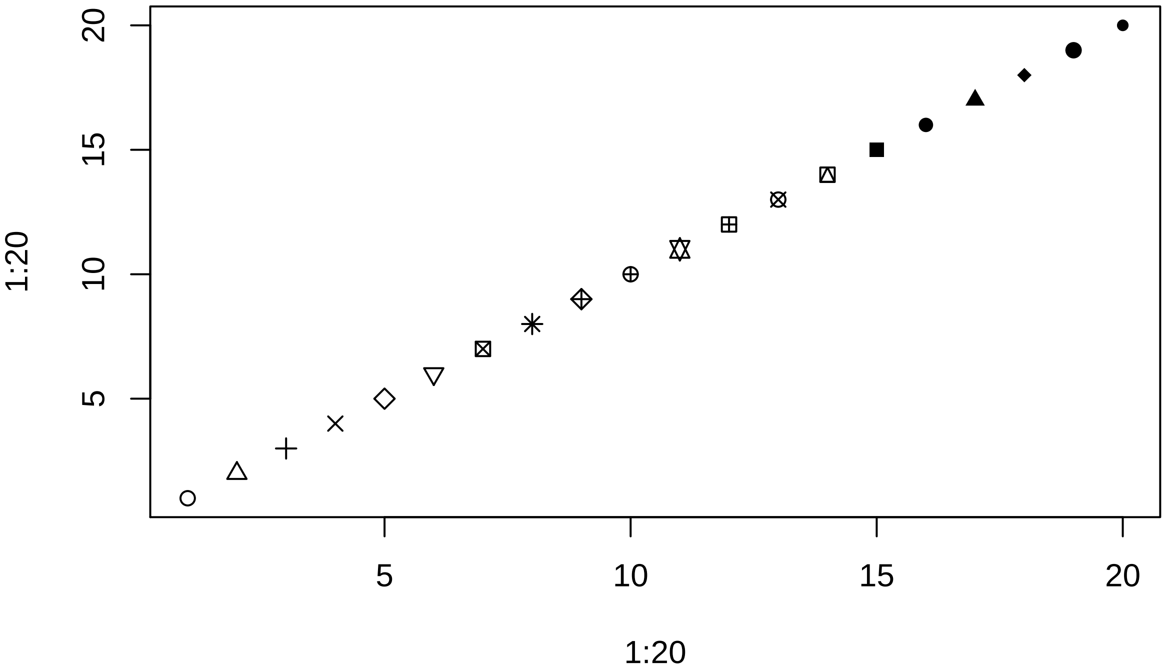
```r
plot(lstat, medv, pch = 20)
```



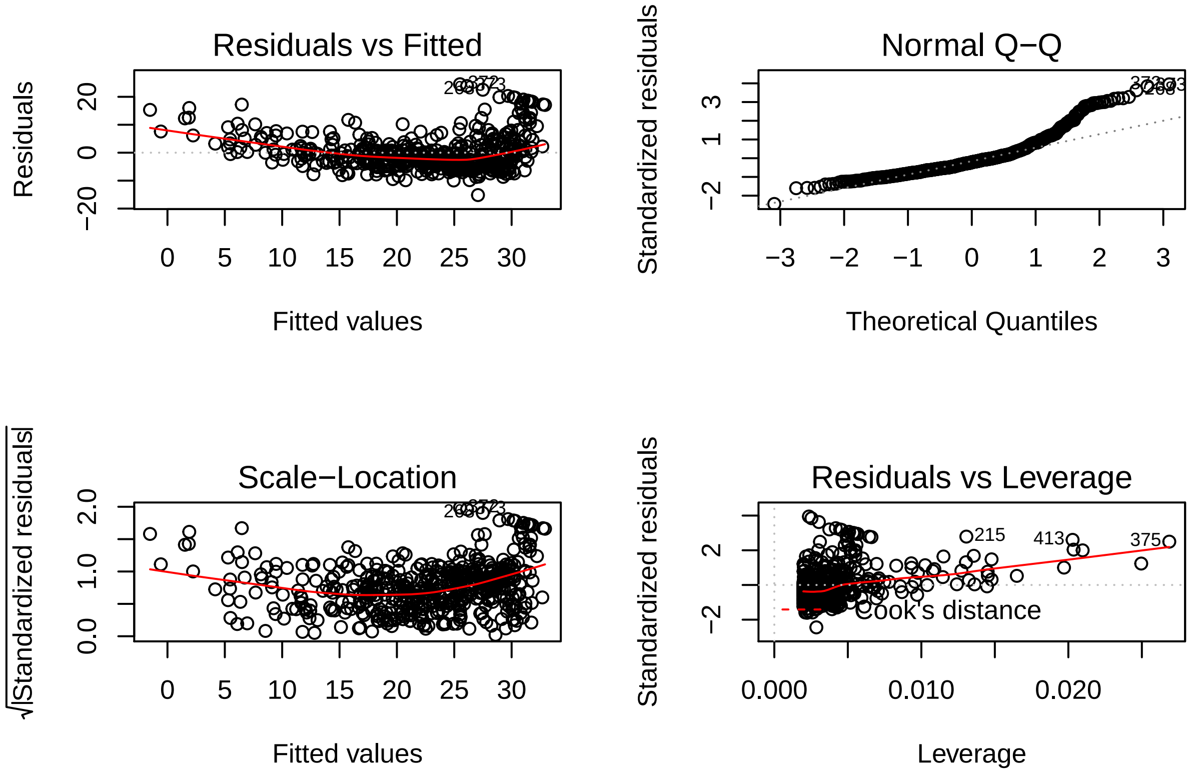
```r
plot(lstat, medv, pch = "+")
```



```
plot(1:20, 1:20, pch = 1:20)
```



```
par(mfrow = c(2, 2))
plot(lm.fit)
```



```
plot(predict(lm.fit), residuals(lm.fit))
plot(predict(lm.fit), rstudent(lm.fit))
plot(hatvalues(lm.fit))
which.max(hatvalues(lm.fit))
```

```
## 375
## 375
```

```
# Multiple Linear Regression

lm.fit = lm(medv ~ lstat + age, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
lm.fit = lm(medv ~ ., data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
library(car)
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

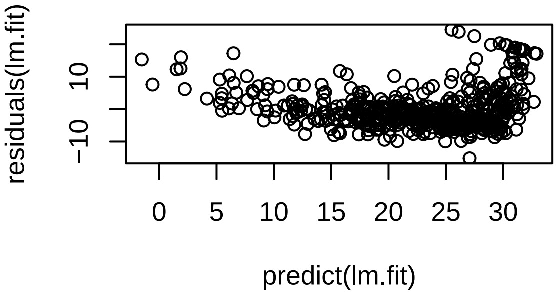
```
vif(lm.fit)
```

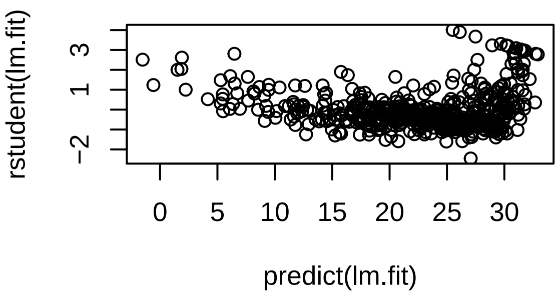
```
##     crim        zn     indus      chas       nox        rm       age       dis
## 1.792192 2.298758 3.991596 1.073995 4.393720 1.933744 3.100826 3.955945
##      rad      tax  ptratio    black    lstat
## 7.484496 9.008554 1.799084 1.348521 2.941491
```

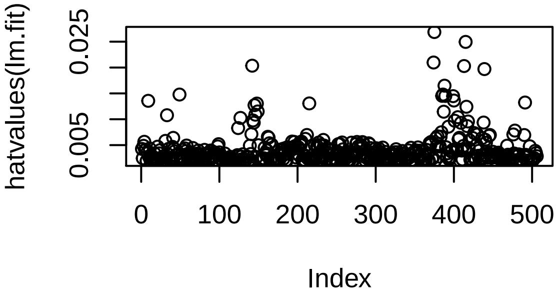
```
lm.fit1 = lm(medv ~ . - age, data = Boston)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
lm.fit1 = update(lm.fit, ~. - age)
```







Question 3 Estimating Wine Quality using Regression Trees

Step 1: Collecting data

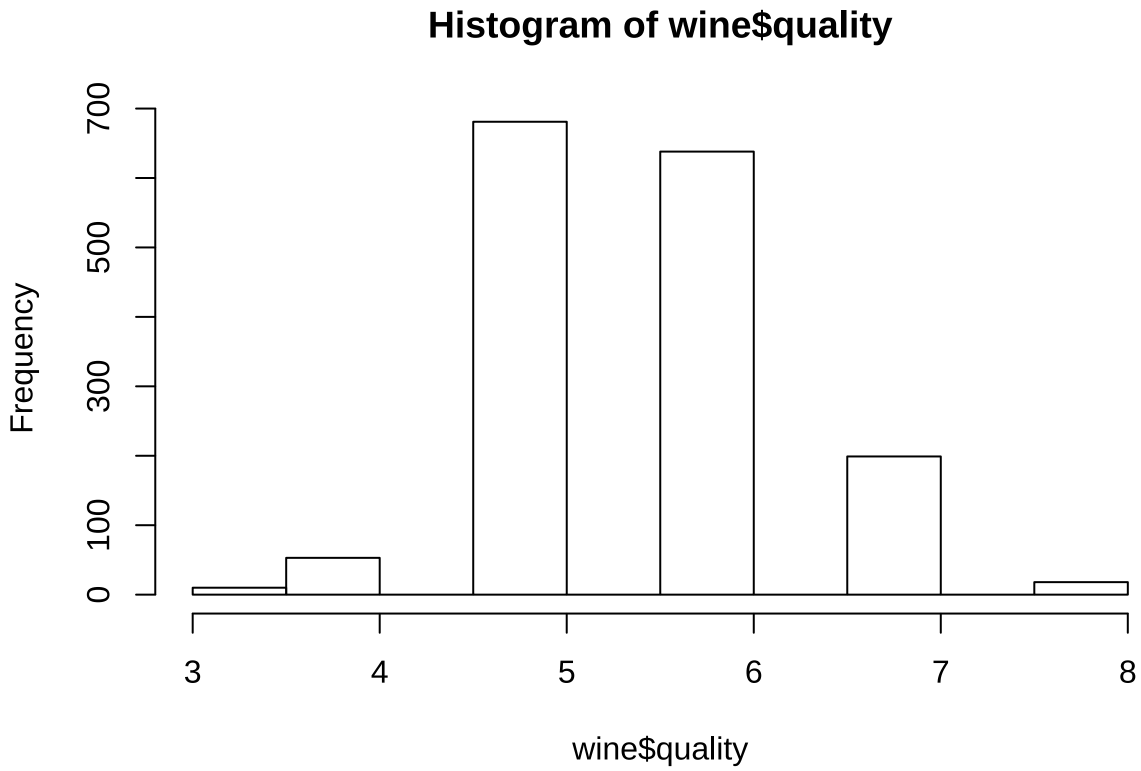
The red wine data includes information on 12 chemical properties of 1599 wine samples. For each wine, a laboratory analysis measured characteristics such as the acidity, sugar content, chlorides, sulfur, alcohol, pH, and density. The samples were then rated in a blind tasting by panels of no less than three judges on a quality scale ranging from zero (very bad) to 10 (excellent). In the case that the judges disagreed on the rating, the median value was used.

```
## Step 2: Exploring and preparing the data ----
wine <- read.csv("redwines.csv")

# examine the wine data
str(wine)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  6.5 9.1 6.9 7.3 12.5 5.4 10.4 7.9 7.3 9.5 ...
##  $ volatile.acidity    : num  0.9 0.22 0.52 0.59 0.28 0.74 0.28 0.4 0.39 0.37 ...
##  $ citric.acid         : num  0 0.24 0.25 0.26 0.54 0.09 0.54 0.3 0.31 0.52 ...
##  $ residual.sugar      : num  1.6 2.1 2.6 2 2.3 1.7 2.7 1.8 2.4 2 ...
##  $ chlorides           : num  0.052 0.078 0.081 0.08 0.082 0.089 0.105 0.157 0.074 0.088 ...
##  $ free.sulfur.dioxide : num  9 1 10 17 12 16 5 2 9 12 ...
##  $ total.sulfur.dioxide: num  17 28 37 104 29 26 19 45 46 51 ...
##  $ density             : num  0.995 0.999 0.997 0.996 1 ...
##  $ pH                  : num  3.5 3.41 3.46 3.28 3.11 3.67 3.25 3.31 3.41 3.29 ...
##  $ sulphates           : num  0.63 0.87 0.5 0.52 1.36 0.56 0.63 0.91 0.54 0.58 ...
##  $ alcohol             : num  10.9 10.3 11 9.9 9.8 11.6 9.5 9.5 9.4 11.1 ...
##  $ quality             : int  6 6 5 5 7 6 5 6 6 6 ...
```

```r
# the distribution of quality ratings
hist(wine$quality)
```



```r
# summary statistics of the wine data
summary(wine)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar  
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900  
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900  
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200  
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539  
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600  
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500  
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00      
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00      
##  Median :0.07900   Median :14.00       Median : 38.00      
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47      
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00      
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00      
##     density             pH          sulphates         alcohol     
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40  
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50  
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20  
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42  
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10  
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90  
##     quality     
##  Min.   :3.000  
##  1st Qu.:5.000  
##  Median :6.000  
##  Mean   :5.636  
##  3rd Qu.:6.000  
##  Max.   :8.000  
```

```r
wine_train <- wine[1:1200, ]
wine_test <- wine[1201:1599, ]

## Step 3: Training a model on the data ---- regression tree using rpart
library(rpart)
m.rpart <- rpart(quality ~ ., data = wine_train)

# get basic information about the tree
m.rpart
```

```
## n= 1200 
## 
## node), split, n, deviance, yval
##       * denotes terminal node
## 
##  1) root 1200 771.63250 5.642500  
##    2) alcohol< 11.45 982 519.39820 5.489817  
##      4) sulphates< 0.585 393 147.11960 5.201018  
##        8) volatile.acidity>=1.0125 16  11.75000 4.375000 *
##        9) volatile.acidity< 1.0125 377 123.98940 5.236074 *
##      5) sulphates>=0.585 589 317.62990 5.682513  
##       10) alcohol< 9.975 248 114.22180 5.415323  
##         20) volatile.acidity>=0.555 109  35.02752 5.165138 *
##         21) volatile.acidity< 0.555 139  67.02158 5.611511 *
##       11) alcohol>=9.975 341 172.82700 5.876833  
##         22) volatile.acidity>=0.405 217  91.85253 5.718894 *
##         23) volatile.acidity< 0.405 124  66.08871 6.153226 *
##    3) alcohol>=11.45 218 126.22020 6.330275  
##      6) sulphates< 0.635 95  52.00000 6.000000  
##       12) pH>=3.265 65  29.53846 5.769231 *
##       13) pH< 3.265 30  11.50000 6.500000 *
##      7) sulphates>=0.635 123  55.85366 6.585366 *
```

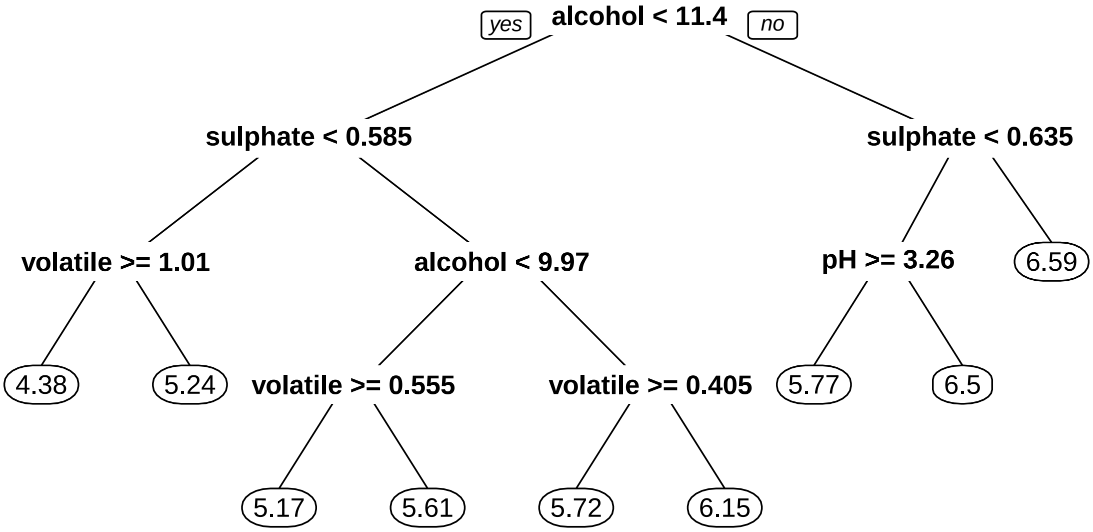
```r
# get more detailed information about the tree
summary(m.rpart)
```

```
## Call:
## rpart(formula = quality ~ ., data = wine_train)
##   n= 1200 
## 
##           CP nsplit rel error    xerror       xstd
## 1 0.16330850      0 1.0000000 1.0007319 0.04420910
## 2 0.07082218      1 0.8366915 0.8591946 0.04169890
## 3 0.03963173      2 0.7658693 0.8072778 0.03973961
## 4 0.02380217      3 0.7262376 0.7901676 0.04002801
## 5 0.01929122      4 0.7024354 0.7549478 0.03702665
## 6 0.01577521      5 0.6831442 0.7539890 0.03671571
## 7 0.01474822      6 0.6673690 0.7411164 0.03567896
## 8 0.01420565      7 0.6526208 0.7417990 0.03553972
## 9 0.01000000      8 0.6384151 0.7331810 0.03481983
##
## Variable importance
##              alcohol            sulphates     volatile.acidity
##                   36                   18                   12
##              density          citric.acid        fixed.acidity
##                   11                    6                    6
##                   pH            chlorides total.sulfur.dioxide
##                    5                    3                    2
##  free.sulfur.dioxide
##                    1
##
## Node number 1: 1200 observations,    complexity param=0.1633085
##   mean=5.6425, MSE=0.6430271
##   left son=2 (982 obs) right son=3 (218 obs)
##   Primary splits:
##       alcohol          < 11.45    to the left,  improve=0.16330850, (0 missing)
##       sulphates        < 0.645    to the left,  improve=0.11979140, (0 missing)
##       volatile.acidity < 0.555    to the right, improve=0.09842250, (0 missing)
##       citric.acid      < 0.295    to the left,  improve=0.06785835, (0 missing)
##       density          < 0.995565 to the right, improve=0.06322246, (0 missing)
##   Surrogate splits:
##       density          < 0.994185 to the right, agree=0.874, adj=0.307, (0 split)
##       fixed.acidity    < 5.5      to the right, agree=0.834, adj=0.087, (0 split)
##       chlorides        < 0.0525   to the right, agree=0.829, adj=0.060, (0 split)
##       pH               < 3.695    to the left,  agree=0.825, adj=0.037, (0 split)
##       volatile.acidity < 0.14     to the right, agree=0.821, adj=0.014, (0 split)
##
## Node number 2: 982 observations,    complexity param=0.07082218
##   mean=5.489817, MSE=0.5289187
##   left son=4 (393 obs) right son=5 (589 obs)
##   Primary splits:
##       sulphates            < 0.585    to the left,  improve=0.10521540, (0 missing)
##       volatile.acidity     < 0.5875   to the right, improve=0.08454043, (0 missing)
##       alcohol              < 9.975    to the left,  improve=0.08070901, (0 missing)
##       citric.acid          < 0.295    to the left,  improve=0.04293423, (0 missing)
##       total.sulfur.dioxide < 83.5     to the right, improve=0.03466039, (0 missing)
##   Surrogate splits:
##       volatile.acidity     < 0.6525   to the right, agree=0.647, adj=0.117, (0 split)
##       total.sulfur.dioxide < 80.5     to the right, agree=0.629, adj=0.074, (0 split)
##       citric.acid          < 0.085    to the left,  agree=0.625, adj=0.064, (0 split)
##       density              < 0.99477  to the left,  agree=0.612, adj=0.031, (0 split)
##       residual.sugar       < 6.25     to the right, agree=0.607, adj=0.018, (0 split)
##
## Node number 3: 218 observations,    complexity param=0.02380217
##   mean=6.330275, MSE=0.5789917
##   left son=6 (95 obs) right son=7 (123 obs)
##   Primary splits:
##       sulphates        < 0.635    to the left,  improve=0.14551180, (0 missing)
##       citric.acid      < 0.325    to the left,  improve=0.07869813, (0 missing)
##       fixed.acidity    < 7.75     to the left,  improve=0.07347418, (0 missing)
##       pH               < 3.375    to the right, improve=0.05090070, (0 missing)
##       volatile.acidity < 0.425    to the right, improve=0.05063722, (0 missing)
##   Surrogate splits:
##       fixed.acidity    < 7.15     to the left,  agree=0.688, adj=0.284, (0 split)
##       citric.acid      < 0.285    to the left,  agree=0.679, adj=0.263, (0 split)
##       density          < 0.99405  to the left,  agree=0.661, adj=0.221, (0 split)
##       pH               < 3.415    to the right, agree=0.628, adj=0.147, (0 split)
##       volatile.acidity < 0.5925   to the right, agree=0.624, adj=0.137, (0 split)
##
## Node number 4: 393 observations,    complexity param=0.01474822
##   mean=5.201018, MSE=0.3743501
##   left son=8 (16 obs) right son=9 (377 obs)
##   Primary splits:
##       volatile.acidity < 1.0125   to the right, improve=0.07735342, (0 missing)
##       chlorides        < 0.13     to the right, improve=0.03583731, (0 missing)
##       pH               < 3.425    to the right, improve=0.03093882, (0 missing)
##       sulphates        < 0.525    to the left,  improve=0.02835730, (0 missing)
##       density          < 0.99689  to the right, improve=0.01729366, (0 missing)
##
## Node number 5: 589 observations,    complexity param=0.03963173
##   mean=5.682513, MSE=0.5392697
##   left son=10 (248 obs) right son=11 (341 obs)
##   Primary splits:
##       alcohol              < 9.975    to the left,  improve=0.09627913, (0 missing)
##       volatile.acidity     < 0.405    to the right, improve=0.09248853, (0 missing)
##       total.sulfur.dioxide < 82.5     to the right, improve=0.06299294, (0 missing)
##       density              < 0.995745 to the right, improve=0.04054997, (0 missing)
##       chlorides            < 0.0975   to the right, improve=0.03815234, (0 missing)
##   Surrogate splits:
##       chlorides            < 0.144    to the right, agree=0.632, adj=0.125, (0 split)
##       total.sulfur.dioxide < 61.5     to the right, agree=0.625, adj=0.109, (0 split)
##       density              < 0.99675  to the right, agree=0.620, adj=0.097, (0 split)
##       sulphates            < 1.045    to the right, agree=0.613, adj=0.081, (0 split)
##       pH                   < 3.045    to the left,  agree=0.596, adj=0.040, (0 split)
##
## Node number 6: 95 observations,    complexity param=0.01420565
##   mean=6, MSE=0.5473684
##   left son=12 (65 obs) right son=13 (30 obs)
##   Primary splits:
##       pH                  < 3.265    to the right, improve=0.2107988, (0 missing)
##       volatile.acidity    < 0.495    to the right, improve=0.1387017, (0 missing)
##       citric.acid         < 0.445    to the left,  improve=0.1376449, (0 missing)
##       free.sulfur.dioxide < 31.5     to the left,  improve=0.1286196, (0 missing)
##       fixed.acidity       < 8.7      to the left,  improve=0.1103214, (0 missing)
##   Surrogate splits:
##       citric.acid         < 0.335    to the left,  agree=0.874, adj=0.600, (0 split)
##       fixed.acidity       < 7.8      to the left,  agree=0.863, adj=0.567, (0 split)
##       volatile.acidity    < 0.385    to the right, agree=0.800, adj=0.367, (0 split)
##       chlorides           < 0.0995   to the left,  agree=0.758, adj=0.233, (0 split)
##       free.sulfur.dioxide < 34       to the left,  agree=0.747, adj=0.200, (0 split)
##
## Node number 7: 123 observations
##   mean=6.585366, MSE=0.4540948
##
```

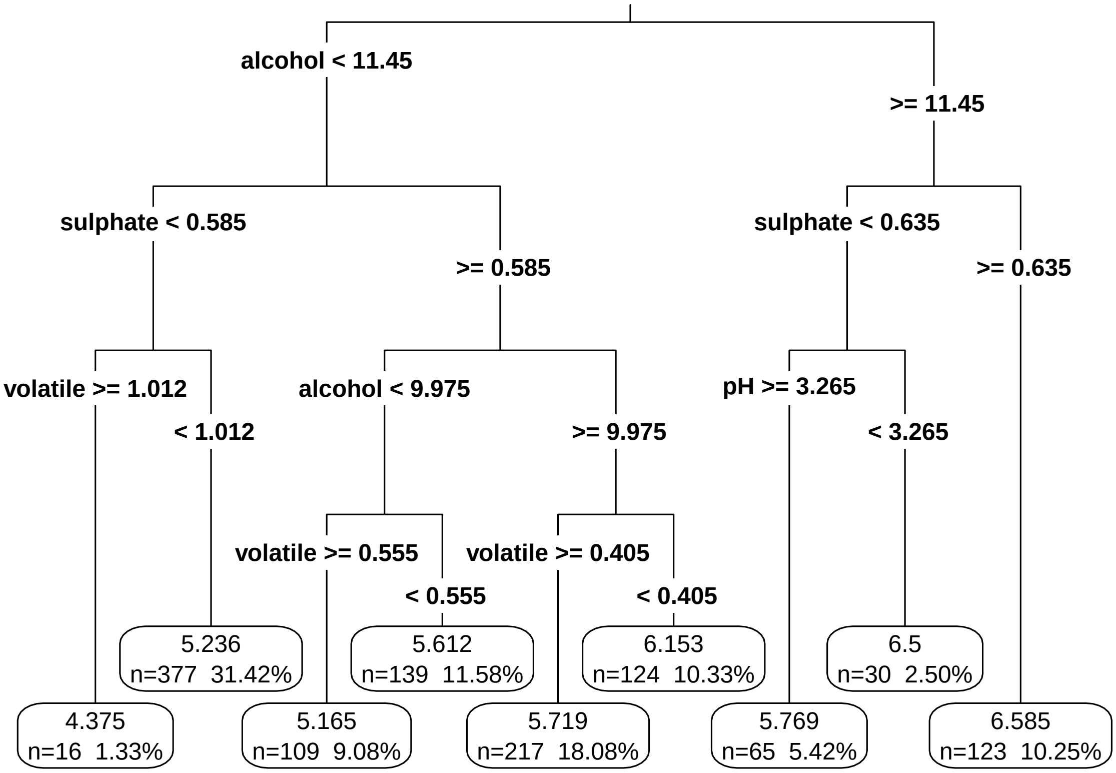
```
## Node number 8: 16 observations
##   mean=4.375, MSE=0.734375
##
## Node number 9: 377 observations
##   mean=5.236074, MSE=0.3288843
##
## Node number 10: 248 observations,    complexity param=0.01577521
##   mean=5.415323, MSE=0.4605717
##   left son=20 (109 obs) right son=21 (139 obs)
##   Primary splits:
##       volatile.acidity     < 0.555    to the right, improve=0.10657050, (0 missing)
##       fixed.acidity        < 11.8     to the left,  improve=0.08845325, (0 missing)
##       total.sulfur.dioxide < 46.5     to the right, improve=0.07759974, (0 missing)
##       free.sulfur.dioxide  < 22.5     to the right, improve=0.07143705, (0 missing)
##       pH                   < 2.99     to the right, improve=0.02794989, (0 missing)
##   Surrogate splits:
##       citric.acid          < 0.245    to the left,  agree=0.766, adj=0.468, (0 split)
##       fixed.acidity        < 7.15     to the left,  agree=0.641, adj=0.183, (0 split)
##       density              < 0.99701  to the left,  agree=0.637, adj=0.174, (0 split)
##       total.sulfur.dioxide < 51.5     to the right, agree=0.617, adj=0.128, (0 split)
##       pH                   < 3.455    to the right, agree=0.601, adj=0.092, (0 split)
##
## Node number 11: 341 observations,    complexity param=0.01929122
##   mean=5.876833, MSE=0.506824
##   left son=22 (217 obs) right son=23 (124 obs)
##   Primary splits:
##       volatile.acidity     < 0.405    to the right, improve=0.08613085, (0 missing)
##       sulphates            < 0.725    to the left,  improve=0.06072160, (0 missing)
##       total.sulfur.dioxide < 83       to the right, improve=0.05282714, (0 missing)
##       pH                   < 3.48     to the right, improve=0.04183314, (0 missing)
##       citric.acid          < 0.295    to the left,  improve=0.03721782, (0 missing)
##   Surrogate splits:
##       citric.acid          < 0.315    to the left,  agree=0.762, adj=0.347, (0 split)
##       sulphates            < 0.765    to the left,  agree=0.683, adj=0.129, (0 split)
##       chlorides            < 0.0675   to the right, agree=0.663, adj=0.073, (0 split)
##       total.sulfur.dioxide < 10.5     to the right, agree=0.651, adj=0.040, (0 split)
##       free.sulfur.dioxide  < 33.5     to the left,  agree=0.648, adj=0.032, (0 split)
##
## Node number 12: 65 observations
##   mean=5.769231, MSE=0.4544379
##
## Node number 13: 30 observations
##   mean=6.5, MSE=0.3833333
##
## Node number 20: 109 observations
##   mean=5.165138, MSE=0.3213534
##
## Node number 21: 139 observations
##   mean=5.611511, MSE=0.4821697
##
## Node number 22: 217 observations
##   mean=5.718894, MSE=0.4232836
##
## Node number 23: 124 observations
##    mean=6.153226, MSE=0.5329735
```

```
# use the rpart.plot package to create a visualization
library(rpart.plot)

# a basic decision tree diagram
rpart.plot(m.rpart, digits = 3)
```



```
# a few adjustments to the diagram
rpart.plot(m.rpart, digits = 4, fallen.leaves = TRUE, type = 3, extra = 101)
```



```
## Step 4: Evaluate model performance ----

# generate predictions for the testing dataset
p.rpart <- predict(m.rpart, wine_test)

# compare the distribution of predicted values vs. actual values
summary(p.rpart)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.375   5.236   5.612   5.610   5.769   6.585
```

```
summary(wine_test$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.617   6.000   8.000
```

```
# compare the correlation
cor(p.rpart, wine_test$quality)
```

```
## [1] 0.5986281
```

```
# function to calculate the mean absolute error
MAE <- function(actual, predicted) {
    mean(abs(actual - predicted))
}

# mean absolute error between predicted and actual values
MAE(p.rpart, wine_test$quality)
```

```
## [1] 0.5384269
```

```
# mean absolute error between actual values and mean value
mean(wine_train$quality)
```

```
## [1] 5.6425
```

```
MAE(5.87, wine_test$quality)
```

```
## [1] 0.692807
```

```
## Step 5: Improving model performance ---- train a M5' Model Tree
library(RWeka)
```

```
##
## Attaching package: 'RWeka'
```

```
## The following object is masked from 'package:caTools':
##
##     LogitBoost
```

```
m.m5p <- M5P(quality ~ ., data = wine_train)

# display the tree
m.m5p
```

```
## M5 pruned model tree:
## (using smoothed linear models)
##
## alcohol <= 10.45 : LM1 (678/73.826%)
## alcohol >  10.45 :
## |   sulphates <= 0.645 : LM2 (233/87.031%)
## |   sulphates >  0.645 : LM3 (289/83.863%)
##
## LM num: 1
## quality =
##  -0.8395 * volatile.acidity
##  + 0.0282 * residual.sugar
##  - 1.4283 * chlorides
##  + 0.0001 * free.sulfur.dioxide
##  - 0.0025 * total.sulfur.dioxide
##  - 0.0108 * pH
##  + 0.5998 * sulphates
##  + 0.2429 * alcohol
##  + 3.3295
##
## LM num: 2
## quality =
##  -1.0129 * volatile.acidity
##  - 0.0611 * chlorides
##  + 0.0248 * free.sulfur.dioxide
##  - 0.004 * total.sulfur.dioxide
##  - 2.9878 * density
##  - 1.0146 * pH
##  + 1.7173 * sulphates
##  + 0.3149 * alcohol
##  + 7.8841
##
## LM num: 3
## quality =
##  -0.0668 * volatile.acidity
##  - 3.0162 * chlorides
##  + 0.0007 * free.sulfur.dioxide
##  - 0.0083 * total.sulfur.dioxide
##  - 2.4374 * density
##  - 0.0599 * pH
##  + 0.9254 * sulphates
##  + 0.2697 * alcohol
##  + 5.669
##
## Number of Rules : 3
```

```r
# get a summary of the model's performance
summary(m.m5p)
```

```
##
## === Summary ===
##
## Correlation coefficient                  0.6136
## Mean absolute error                      0.5051
## Root mean squared error                  0.6332
## Relative absolute error                 74.5819 %
## Root relative squared error             78.967  %
## Total Number of Instances             1200
```

```r
# generate predictions for the model
p.m5p <- predict(m.m5p, wine_test)

# summary statistics about the predictions
summary(p.m5p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.768   5.259   5.461   5.603   5.916   7.049
```

```r
# correlation between the predicted and true values
cor(p.m5p, wine_test$quality)
```

```
## [1] 0.6639448
```

```r
# mean absolute error of predicted and true values (uses a custom function
# defined above)
MAE(wine_test$quality, p.m5p)
```

```
## [1] 0.4908107
```