R Notebook

Problem 1 (60 Points)

Download the data set on student achievement in secondary education math education of two Portuguese schools (use the data set Students Math). Using any packages you wish, complete the following tasks: 1. (10 pts) Create scatter plots and pairwise correlations between age, absences, G1, and G2 and final grade (G3) using the pairs.panels() function in R.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

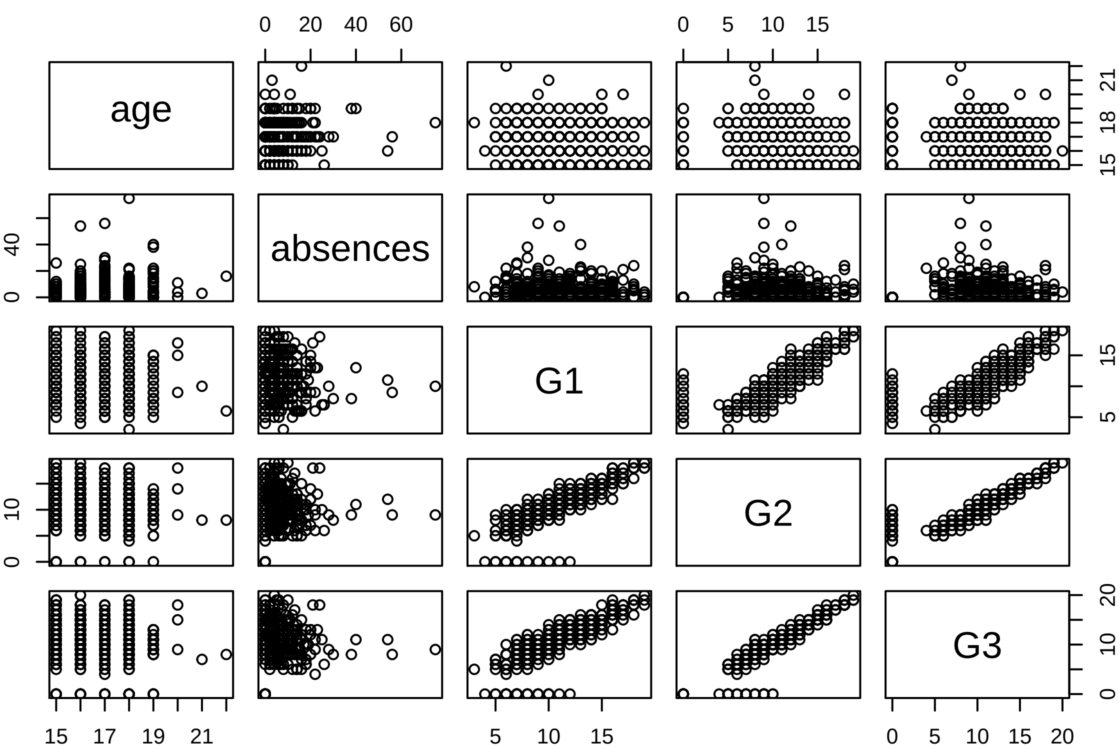
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#setwd('~/Documents/ML/')
df <- read.csv("student-mat.csv", sep = ";")
plt <- df %>% select(3,30,31,32,33)
pairs(plt)
```



2. (10 pts) Build a multiple regression model predicting final math grade (G3) using as many features as you like but you must use at least four. Include at least one categorical variables and be sure to properly convert it to dummy codes. Select the features that you believe are useful – you do not have to include all features.

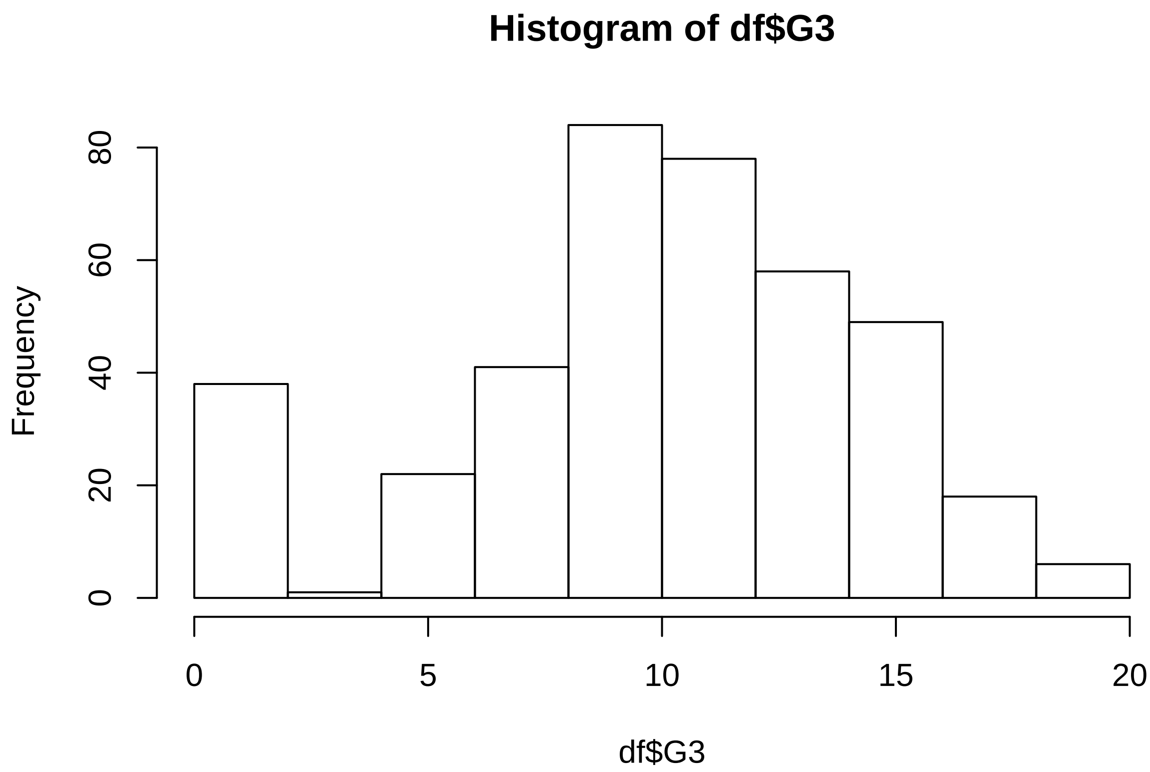
```r
str(df)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```r
summary(df$G3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.00   11.00   10.42   14.00   20.00
```

```r
hist(df$G3)
```




```r
reg_df <- df %>% select(14,31,32,33)
# Convert paid column to binary
dfpaid = data.frame(model.matrix(~ paid, data=df))
pyes <- dfpaid[-1]
reg_data <- cbind(reg_df, pyes)
mm <- lm(G3 ~ studytime + G1 + G2 + paidyes, data = reg_data)
mm
```

```
##
## Call:
## lm(formula = G3 ~ studytime + G1 + G2 + paidyes, data = reg_data)
##
## Coefficients:
## (Intercept)    studytime           G1           G2      paidyes
##     -1.6001      -0.1852       0.1656       0.9809       0.1677
```

3. (20 pts) Use stepwise backward elimination to remove all non-significant variables and then state the final model as an equation. State the backward elimination measure you applied (p-value, AIC, Adjusted R2). This tutorial shows how to use various feature elimination techniques.

The backward elimination measure applied was AIC.

Formula: G3 = 0.72153034 + schoolMS(0.51) + Fjobservices(-0.44) + reasonhome(-0.32) + activitiesyes(-0.3) + romanticyes(-0.31) + age(-0.26) + famrel(0.4) + Walc(0.13) + absences(0.05) + G1(0.17) + G2(0.97)

```r
dum_ft <- data.frame(model.matrix(~ school+sex+address+famsize+Pstatus+Mjob+Fjob+reason+guardian+school
num_ft <- df %>% select(3,7,8,13,14,15,c(24:33))
final_df <- cbind(dum_ft,num_ft)
# Apply AIC backward elimination measure
# Commenting it out so as not to print it all in the pdf file
# step(lm(G3 ~ schoolMS+sexM+addressU+famsizeLE3+PstatusT+     Mjobhealth+Mjobother+Mjobservices+Mjobte
# guardianother+schoolsupyes+famsupyes+paidyes+activitiesyes+nurseryyes+
```

```r
# higheryes+internetyes+romanticyes+age+Medu+Fedu+
# traveltime+studytime+failures+famrel+freetime+goout+
# Dalc+Walc+health+absences+G1+G2, data = final_df),direction = "backward")
```

4. (10 pts) Calculate the 95% confidence interval for a prediction – you may choose any data you wish for some new student.

The 95% confidence interval for the prediction is 8.04 to 11.72

```r
#Prediction = 0.72153034 + schoolMS(+0.51) + Fjobservices(-0.44) + reasonhome(-0.32) + activitiesyes(-0
#    romanticyes(-0.31) + age(-0.26) + famrel(0.4) + Walc(0.13) + absences(0.05) + G1(0.17) + G2(0.97)
Prediction = 0.72153034 + 1*(+0.51) + 1*(-0.44) + 0*(-0.32) + 1*(-0.3) +
    0*(-0.31) + 16*(-0.26) + 4*(0.4) + 3*(0.13) + 10*(0.05) + 8*(0.17) + 10*(0.97)
Prediction
```

```
## [1] 9.88153
```

```r
# Initializing columns of prediction and absolute error
final_df$P<-0
final_df$absErr<-0
# Making predictions and calculating absolute error
for (i in 1:nrow(final_df)){
  final_df$P[i] <- 0.51*final_df$schoolMS[i]+(-0.44)*final_df$Fjobservices[i]+(-0.32)*final_df$reasonhom
  final_df$P[i] <- round(final_df$P[i])
  final_df$absErr[i] <-abs(final_df[i,43]-final_df$P[i])
}
# Calculating MAD
MAD <- mean(final_df$absErr)
MAD
```

```
## [1] 1.16962
```

```r
# Prediction with a 95% prediction interval
D <- 0.8*MAD
D
```

```
## [1] 0.9356962
```

```r
CI1 <- Prediction - (1.96*D)
CI1
```

```
## [1] 8.047566
```

```r
CI2 <- Prediction + (1.96*D)
CI2
```

```
## [1] 11.71549
```

5. (10 pts) What is the RMSE for this model – use the entire data set for both training and validation. You may find the residuals() function useful. Alternatively, you can inspect the model object, e.g., if your model is in the variable m, then the residuals (errors) are in m$residuals and your predicted values (fitted values) are in m$fitted.values.

The RMSE for this model is 1.85

```r
# Making a column for squared error
final_df$SqErr <- 0
# Putting values in the column
for (i in 1:nrow(final_df)){
  final_df$SqErr[i] <- (final_df$absErr[i])^2
```

```r
}
head(final_df)
```

```
##   X.Intercept. schoolMS sexM addressU famsizeLE3 PstatusT Mjobhealth
## 1            1        0    0        1          0        0          0
## 2            1        0    0        1          0        1          0
## 3            1        0    0        1          1        1          0
## 4            1        0    0        1          0        1          1
## 5            1        0    0        1          0        1          0
## 6            1        0    1        1          1        1          0
##   Mjobother Mjobservices Mjobteacher Fjobhealth Fjobother Fjobservices
## 1         0            0           0          0         0            0
## 2         0            0           0          0         1            0
## 3         0            0           0          0         1            0
## 4         0            0           0          0         0            1
## 5         1            0           0          0         1            0
## 6         0            1           0          0         1            0
##   Fjobteacher reasonhome reasonother reasonreputation guardianmother
## 1           1          0           0                0              1
## 2           0          0           0                0              0
## 3           0          0           1                0              1
## 4           0          1           0                0              1
## 5           0          1           0                0              0
## 6           0          0           0                1              1
##   guardianother schoolsupyes famsupyes paidyes activitiesyes nurseryyes
## 1             0            1         0       0             0          1
## 2             0            0         1       0             0          0
## 3             0            1         0       1             0          1
## 4             0            0         1       1             1          1
## 5             0            0         1       1             0          1
## 6             0            0         1       1             1          1
##   higheryes internetyes romanticyes age Medu Fedu traveltime studytime
## 1         1           0           0  18    4    4          2         2
## 2         1           1           0  17    1    1          1         2
## 3         1           1           0  15    1    1          1         2
## 4         1           1           1  15    4    2          1         3
## 5         1           0           0  16    3    3          1         2
## 6         1           1           0  16    4    3          1         2
##   failures famrel freetime goout Dalc Walc health absences G1 G2 G3  P
## 1        0      4        3     4    1    1      3        6  5  6  6  5
## 2        0      5        3     3    1    1      3        4  5  5  6  4
## 3        3      4        3     2    2    3      3       10  7  8 10  8
## 4        0      3        2     2    1    1      5        2 15 14 15 13
## 5        0      4        3     2    1    2      5        4  6 10 10  9
## 6        0      5        4     2    1    2      5       10 15 15 15 16
##   absErr SqErr
## 1      1     1
## 2      2     4
## 3      2     4
## 4      2     4
## 5      1     1
## 6      1     1
```

```r
# Calculating RMSE
RMSE <- sqrt(mean(final_df$SqErr))
RMSE
```

```
## [1] 1.850077
```

Problem 2 (40 Points)

For this problem, the following short tutorial might be helpful in interpreting the logistic regression output.
1. (5 pts) Using the same data set as in Problem (1), add another column, PF – pass-fail. Mark any student whose final grade is less than 10 as F, otherwise as P and then build a dummy code variable for that new column. Use the new dummy variable column as the response variable.

```r
df$PF <- 'F'
for (i in 1:nrow(df)){
  if (df$G3[i] <= 9) {
    df$PF[i] <- "F"
  }
  else {
    df$PF[i] <- "P"
  }
}
head(df$PF)
```

```
## [1] "F" "F" "P" "P" "P" "P"
```

```r
dPF <- data.frame(model.matrix(~ PF, data=df))
dPF <- dPF[-1]

df <- cbind(df,dPF)
```

2. (10 pts) Build a binomial logistic regression model classifying a student as passing or failing. Eliminate any non-significant variable using an elimination approach of your choice. Use as many features as you like but you must use at least four – choose the ones you believe are most useful.

```r
f_df <- final_df %>% select(c(2:43))
f_df <- cbind(f_df,dPF)
# Select the significant variables using info gained from the previous backward elimination measure
# Apply AIC backward elimination measure
f_g3 <- f_df[-42]
#step(glm(PFP~.,data = f_g3),direction = "backward") Commented it out so as not to print 20 pages in th

#Logistic Regression Model
model <- glm(PFP ~Fjobother+ nurseryyes+ age+ failures+ famrel+ goout+ Walc+ absences+ G1 + G2, data = 
```

3. (5 pts) State the regression equation.

Formula: Prediction = (1.95)Fjobother + (-1.16)nurseryyes + (-0.59)age + (0.22)failures + (1.22)famrel +(-0.78)goout + (0.8)Walc + (-0.06)absences + (0.41)G1 + (2.26)G2 -18.52

```r
model
```

```
## 
## Call:  glm(formula = PFP ~ Fjobother + nurseryyes + age + failures + 
##     famrel + goout + Walc + absences + G1 + G2, family = "binomial", 
##     data = f_g3)
## 
## Coefficients:
## (Intercept)    Fjobother   nurseryyes          age     failures
```

```
##   -18.52015      1.95300     -1.16744     -0.59806      0.22010
##      famrel        goout         Walc     absences           G1
##     1.22384     -0.77770      0.80019     -0.06791      0.41290
##          G2
##     2.25897
##
## Degrees of Freedom: 394 Total (i.e. Null);  384 Residual
## Null Deviance:       500.5
## Residual Deviance: 101.4     AIC: 123.4
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = PFP ~ Fjobother + nurseryyes + age + failures +
##     famrel + goout + Walc + absences + G1 + G2, family = "binomial",
##     data = f_g3)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.54916  -0.01353   0.00132   0.06842   2.19722
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -18.52015    5.11306  -3.622 0.000292 ***
## Fjobother     1.95300    0.56861   3.435 0.000593 ***
## nurseryyes   -1.16744    0.73033  -1.599 0.109931
## age          -0.59806    0.23295  -2.567 0.010250 *
## failures      0.22010    0.33460   0.658 0.510669
## famrel        1.22384    0.39820   3.073 0.002116 **
## goout        -0.77770    0.28726  -2.707 0.006783 **
## Walc          0.80019    0.23469   3.410 0.000651 ***
## absences     -0.06791    0.03410  -1.992 0.046400 *
## G1            0.41290    0.21536   1.917 0.055209 .
## G2            2.25897    0.39240   5.757 8.57e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.50  on 394  degrees of freedom
## Residual deviance: 101.36  on 384  degrees of freedom
## AIC: 123.36
##
## Number of Fisher Scoring iterations: 9
```

4. (20 pts) What is the accuracy of your model? Use the entire data set for both training and validation.

Accuracy of the model is 94.17%

```r
f_g3$Pred<-0
f_g3$Pred <- predict(model, data = f_g3, type = 'response')
f_g3$Pred_f <- 0
for (i in 1:nrow(f_g3)){
  if (f_g3$Pred[i] <= 0.4) {
    f_g3$Pred_f[i] <- "0"
```

```r
  }
  else {
    f_g3$Pred_f[i] <- "1"
  }
}

f_g3$Pred_f <- as.numeric(f_g3$Pred_f)
f_g3$Pred_f <- as.factor(f_g3$Pred_f)
f_g3$PFP <- as.factor(f_g3$PFP)
confusionMatrix(f_g3$Pred_f, f_g3$PFP)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 114   7
##          1  16 258
##
##                Accuracy : 0.9418
##                  95% CI : (0.9139, 0.9627)
##     No Information Rate : 0.6709
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8658
##  Mcnemar's Test P-Value : 0.09529
##
##             Sensitivity : 0.8769
##             Specificity : 0.9736
##          Pos Pred Value : 0.9421
##          Neg Pred Value : 0.9416
##              Prevalence : 0.3291
##          Detection Rate : 0.2886
##    Detection Prevalence : 0.3063
##       Balanced Accuracy : 0.9253
##
##        'Positive' Class : 0
##
```

Problem 3 (10 Points)

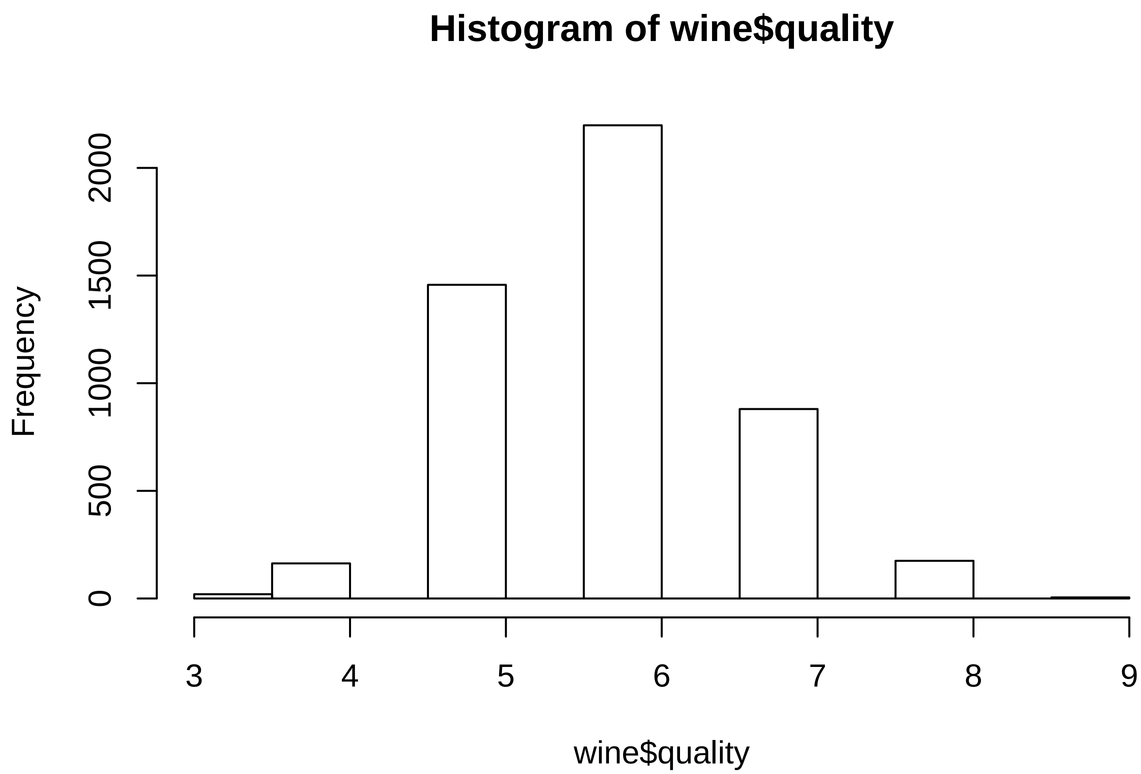
1. (8 pts) Implement the example from the textbook on pages 205 to 217 for the data set on white wines.

```r
library(rpart)
#install.packages("rpart.plot")
library(rpart.plot)
library(RWeka)
wine <- read.csv("whitewines.csv")
str(wine)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
```

```
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ 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 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
```

```r
hist(wine$quality)
```

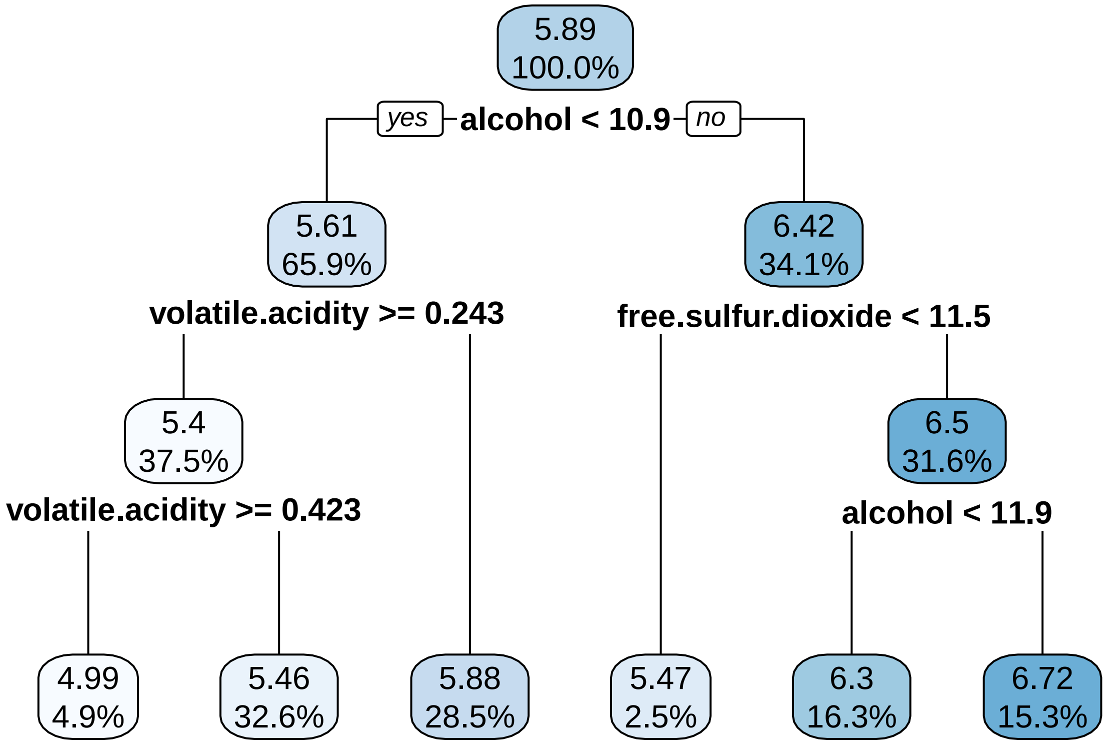



```r
# Divide into training and testing datasets
wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]
# Train the model
m.rpart <- rpart(quality ~ ., data = wine_train)
m.rpart
```

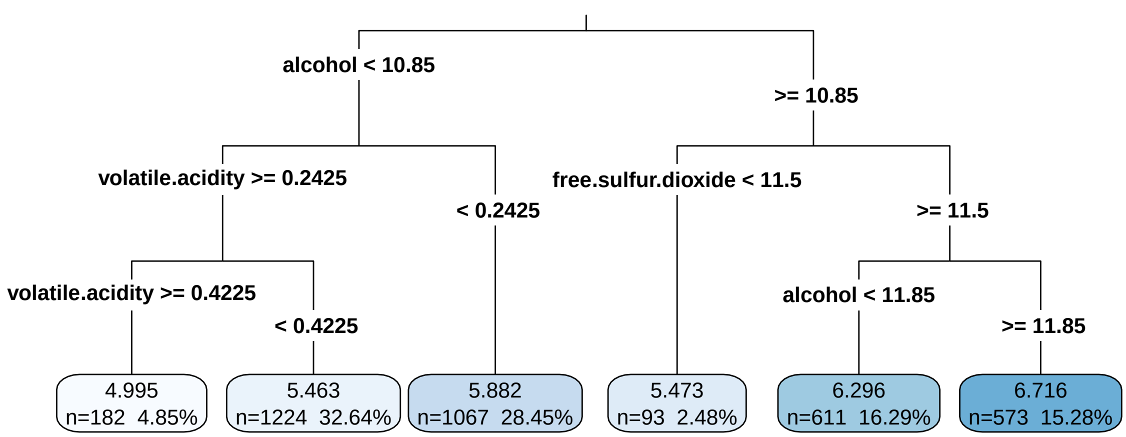
```
## n= 3750
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 3750 3140.06000 5.886933
##    2) alcohol< 10.85 2473 1510.66200 5.609381
##      4) volatile.acidity>=0.2425 1406  740.15080 5.402560
##        8) volatile.acidity>=0.4225 182   92.99451 4.994505 *
##        9) volatile.acidity< 0.4225 1224  612.34560 5.463235 *
##      5) volatile.acidity< 0.2425 1067  631.12090 5.881912 *
##    3) alcohol>=10.85 1277 1069.95800 6.424432
##      6) free.sulfur.dioxide< 11.5 93   99.18280 5.473118 *
```

```
##      7) free.sulfur.dioxide>=11.5 1184  879.99920 6.499155
##       14) alcohol< 11.85 611  447.38130 6.296236 *
##       15) alcohol>=11.85 573  380.63180 6.715532 *
```

```r
rpart.plot(m.rpart, digits = 3)
```




```r
rpart.plot(m.rpart, digits = 4, fallen.leaves = TRUE,
type = 3, extra = 101)
```




```r
# Evaluate model performance
p.rpart <- predict(m.rpart, wine_test)
summary(p.rpart)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.995   5.463   5.882   5.999   6.296   6.716
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.848   6.000   8.000
```

```r
cor(p.rpart, wine_test$quality)
```

```
## [1] 0.4931608
```

```r
# Measuring mean absolute error
MAE <- function(actual, predicted) {
mean(abs(actual - predicted))
}
MAE(p.rpart, wine_test$quality)
```

```
## [1] 0.5732104
```

```r
mean(wine_train$quality)
```

```
## [1] 5.886933
```

```r
# Predicting value of 5.78 for every wine sample, MAE is :
MAE(5.87, wine_test$quality)
```

```
## [1] 0.5815679
```

```r
# Improve model performance
m.m5p <- M5P(quality ~ ., data = wine_train)
summary(m.m5p) # Terribly poor results which don't match with the text book
```

```
##
## === Summary ===
##
## Correlation coefficient                 -0.2414
## Mean absolute error                    102.3629
## Root mean squared error                129.5719
## Relative absolute error              14704.2234 %
## Root relative squared error          14159.8116 %
## Total Number of Instances             3750
```

```r
p.m5p <- predict(m.m5p, wine_test)
summary(p.m5p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -539.90 -165.65 -107.07 -112.27  -33.70   32.49
```

```r
cor(p.m5p, wine_test$quality)
```

```
## [1] -0.2036594
```

```r
MAE(wine_test$quality, p.m5p)
```

```
## [1] 118.6835
```

2. (2 pts) Calculate the RMSE for the model.

The RMSE for the model is 0.71

```r
# Measuring root mean squared error
RMSE <- function(actual, predicted) {
sqrt(mean((actual - predicted)^2))
}
RMSE(wine_test$quality, p.rpart)
```

```
## [1] 0.7057153
```