PeerReview_6

Minxin Cheng

```r
#install.packages("psych")
#install.packages("rpart")
#install.packages("rpart.plot")
#install.packages("Cubist")
#install.packages("randomForest")
library(psych)
library(gmodels)
library(rpart)
library(rpart.plot)
library(RWeka)
library(MuMIn)
library(Cubist)
```

```
## Loading required package: lattice
```

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:MuMIn':
##
##     importance
```

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

Problem 1

0. Read in data file

```r
math <- read.table("student/student-mat.csv",
                   sep = ";",
                   header = TRUE)
```

Question 1

Create scatter plots and pairwise correlations between age, absences, G1, and G2 and final grade (G3) using the pairs.panels() function in R.

```r
# check if there is any missing values in the dataset
any(is.na(math))
```

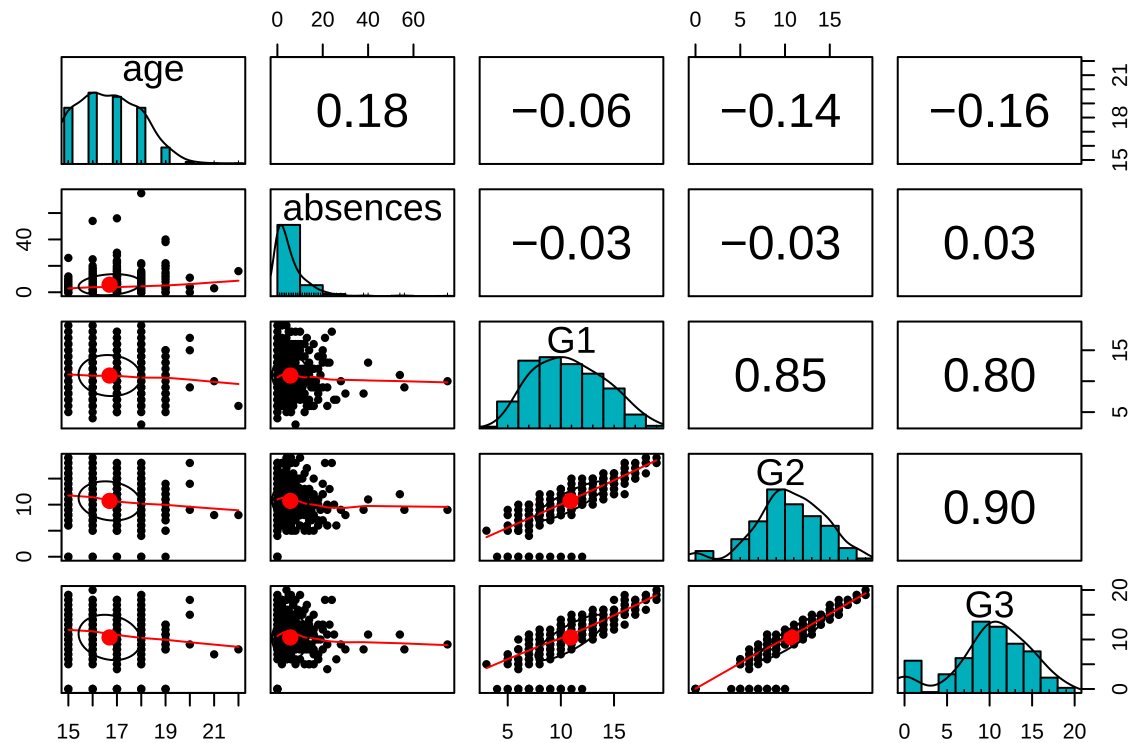
```
## [1] FALSE
```

```r
# create a summary table to get an overview of the dataset
summary(math)
```

```
##     school              sex                 age          address         
##  Length:395         Length:395         Min.   :15.0   Length:395        
##  Class :character   Class :character   1st Qu.:16.0   Class :character  
##  Mode  :character   Mode  :character   Median :17.0   Mode  :character  
##                                        Mean   :16.7                     
##                                        3rd Qu.:18.0                     
##                                        Max.   :22.0                     
##    famsize            Pstatus               Medu            Fedu      
##  Length:395         Length:395         Min.   :0.000   Min.   :0.000  
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000  
##  Mode  :character   Mode  :character   Median :3.000   Median :2.000  
##                                        Mean   :2.749   Mean   :2.522  
##                                        3rd Qu.:4.000   3rd Qu.:3.000  
##                                        Max.   :4.000   Max.   :4.000  
##      Mjob               Fjob              reason            guardian        
##  Length:395         Length:395         Length:395         Length:395        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##    traveltime      studytime        failures        schoolsup        
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:395        
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character  
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character  
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342                     
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000                     
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000                     
##     famsup              paid            activities          nursery         
##  Length:395         Length:395         Length:395         Length:395        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##     higher            internet           romantic             famrel     
##  Length:395         Length:395         Length:395         Min.   :1.000  
##  Class :character   Class :character   Class :character   1st Qu.:4.000  
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000  
##                                                           Mean   :3.944  
```

```
##                                                            3rd Qu.:5.000
##                                                            Max.   :5.000
##     freetime         goout            Dalc            Walc      
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000  
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000  
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291  
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000  
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  
##      health         absences            G1              G2       
##  Min.   :1.000   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00  
##  1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00  
##  Median :4.000   Median : 4.000   Median :11.00   Median :11.00  
##  Mean   :3.554   Mean   : 5.709   Mean   :10.91   Mean   :10.71  
##  3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00  
##  Max.   :5.000   Max.   :75.000   Max.   :19.00   Max.   :19.00  
##        G3       
##  Min.   : 0.00  
##  1st Qu.: 8.00  
##  Median :11.00  
##  Mean   :10.42  
##  3rd Qu.:14.00  
##  Max.   :20.00  
```

```r
# create the panel
pairs.panels(math[, c(3, 30:33)],
             method = "pearson",
             hist.col = "#00AFBB",
             density = TRUE,
             ellipses = TRUE)
```



From the panel, all five variables are not normally distributed. G1 and G2, G1 and G2, G2 and G3 showed strong and positive correlation, the ellipses over the fit line are very flat, also supported there are strong correlations between these variables.

Question 2

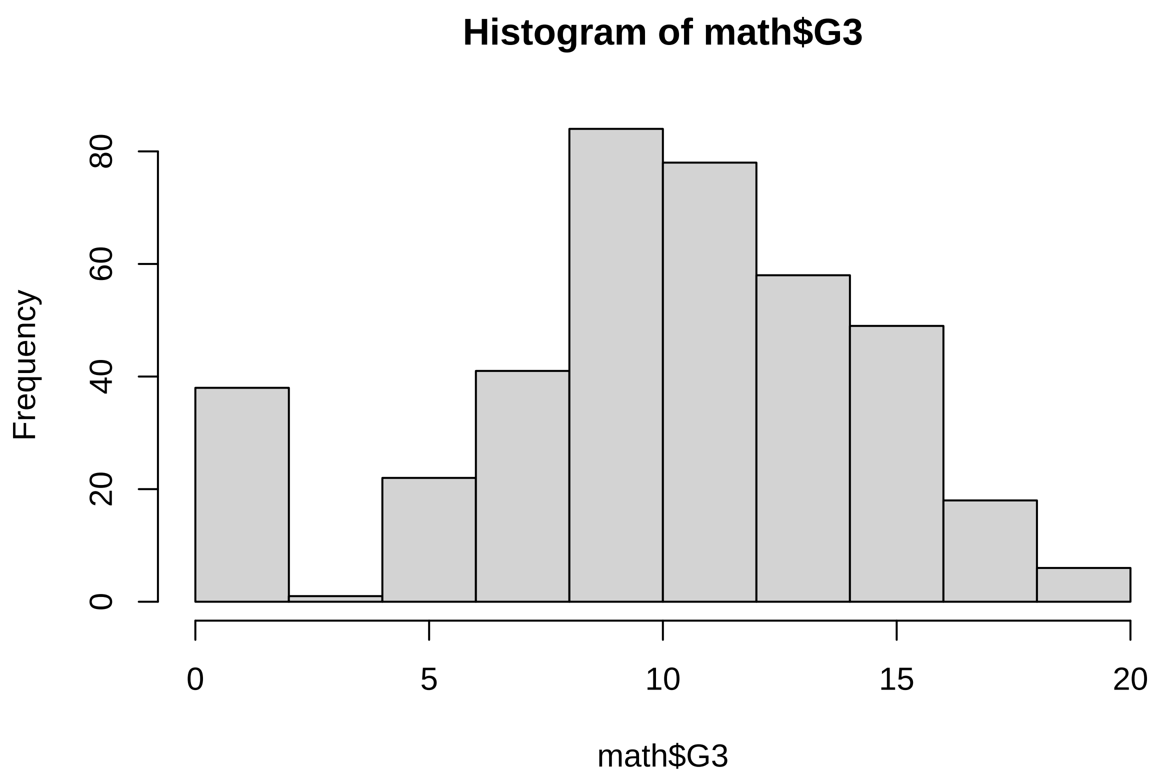
Build a multiple regression model predicting final math grade (G3) using as many features as you like but you mush use at least four. Include at least one categorical variables and be sure to properly convert it to dummy codes. Select the features that you believe are useful – you do not have to include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

1. Check the normality of data

```r
# perform Shapiro test to see the normality of G3
shapiro.test(math$G3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  math$G3
## W = 0.92873, p-value = 8.836e-13
```

```r
# plot a histogram to visually check
hist(math$G3)
```



From the test result, p < 0.05 meaning G3 is not normality distribute.

2. Try different transformation

```
# try min-max
# create a min-max function
normalize <- function(x){
  return((x - min(x)) / (max(x) - min(x)))
}
# min-max transfer then check normality
minMaxG3 <- normalize(math$G3)
shapiro.test(minMaxG3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  minMaxG3
## W = 0.92873, p-value = 8.836e-13
```

```
# z score transfer then check normality
zG3 <- scale(math$G3)
shapiro.test(zG3)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  zG3
## W = 0.92873, p-value = 8.836e-13
```

```
# squred root transfer then check normality
sqrtG3 <- sqrt(math$G3)
shapiro.test(sqrtG3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sqrtG3
## W = 0.73314, p-value < 2.2e-16
```

From the test result, normality didn't improve, will keep the original data.

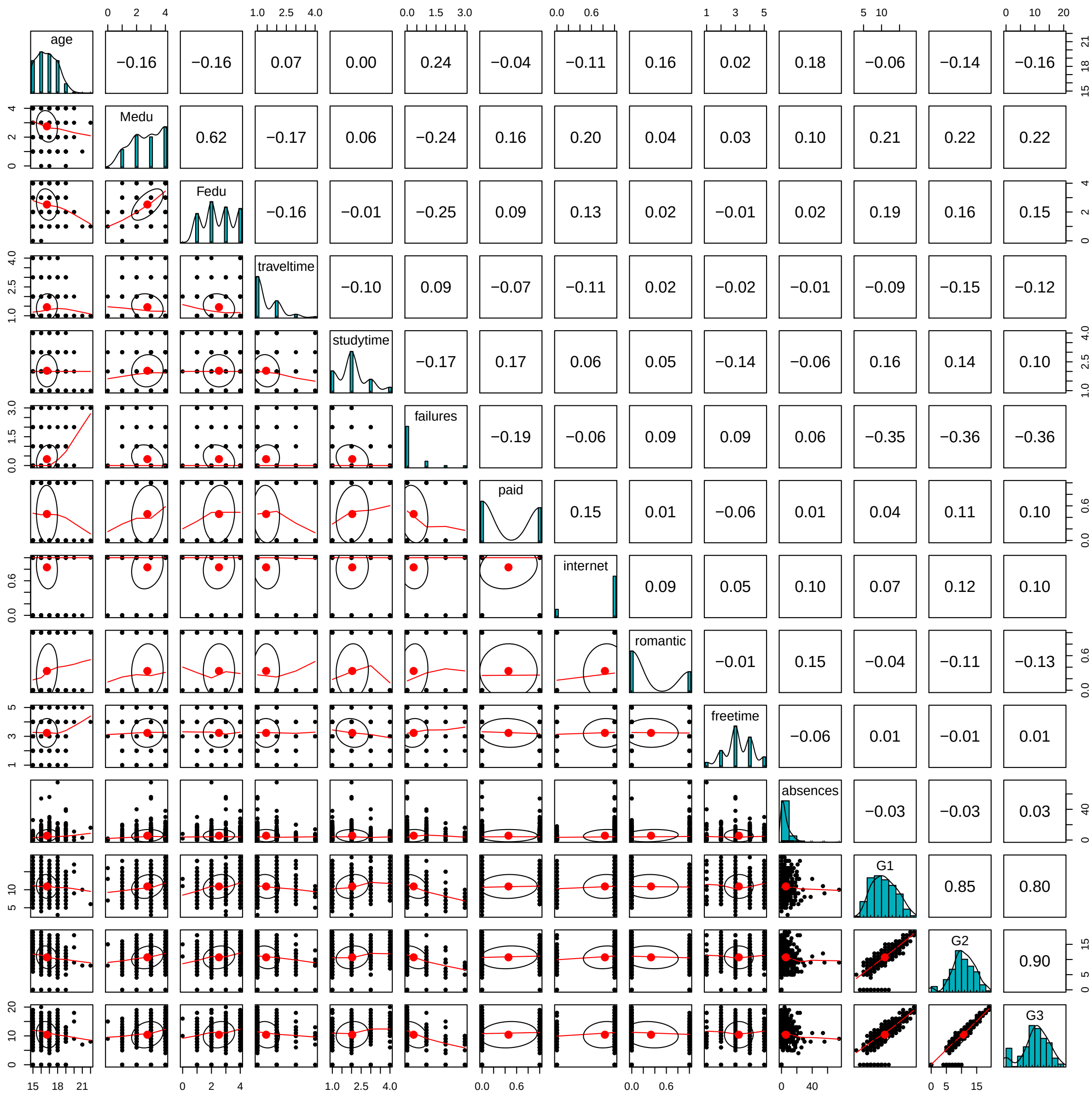
3. Create dummy codes

Form all columns, I am interested in the following variables: age, mom education, dad education, travel time, study time, failures, paid, internet, romantic, freetime, absences, G1, and G2. In these variables, paid, internet, and romantic are characters, will convert them to dummy codes.

```
# converth paid, internet, and romantic to dummy codes
math$paid <- ifelse(math$paid == 'yes', 1, 0)
math$internet <- ifelse(math$internet == 'yes', 1, 0)
math$romantic <- ifelse(math$romantic == 'yes', 1, 0)
```

4. Pair panel to all the interested variables

```
pairs.panels(math[, c(3, 7:8, 13:15, 18, 22:23, 25, 30:33)],
             method = "pearson",
             hist.col = "#00AFBB",
             density = TRUE,
             ellipses = TRUE)
```



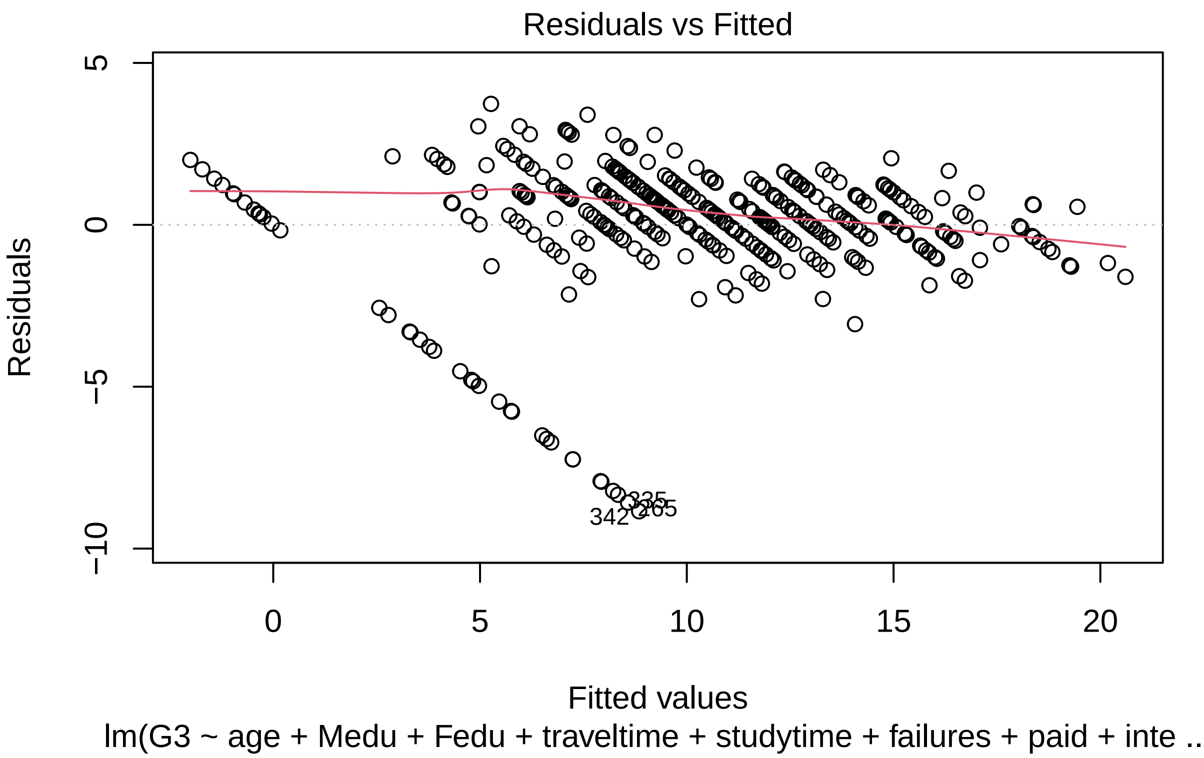
5. Created the first model with the entire dataset and all these interested variables

```
# create the model
fit <- lm(G3 ~ age + Medu + Fedu + traveltime +
            studytime + failures + paid + internet +
            romantic + freetime + absences +
            G1 + G2,
          data = math)
# summary the model
summary(fit)
```

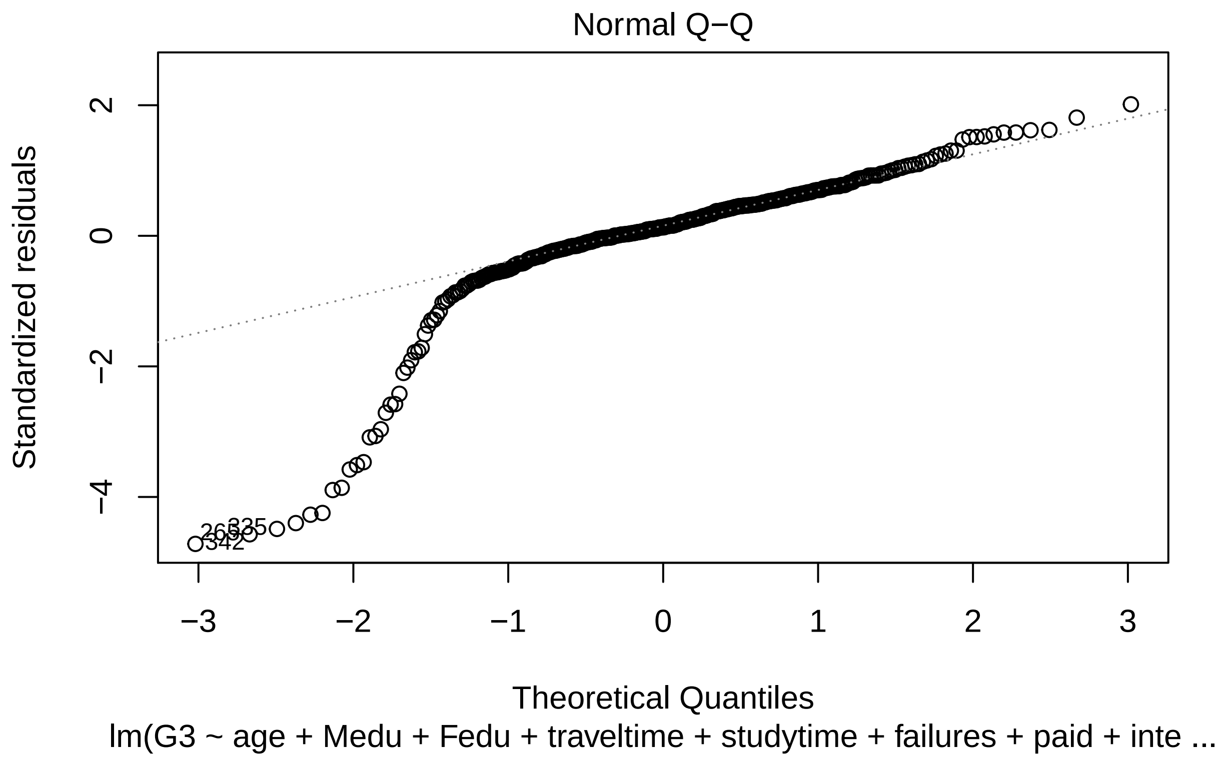
```
##
```

```
## Call:
## lm(formula = G3 ~ age + Medu + Fedu + traveltime + studytime +
##     failures + paid + internet + romantic + freetime + absences +
##     G1 + G2, data = math)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8482 -0.4007  0.2433  0.9722  3.7349
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.74391    1.51086   0.492 0.622737
## age         -0.15845    0.08124  -1.950 0.051854 .
## Medu         0.09831    0.11664   0.843 0.399870
## Fedu        -0.12212    0.11512  -1.061 0.289456
## traveltime   0.12655    0.14202   0.891 0.373462
## studytime   -0.14201    0.12057  -1.178 0.239571
## failures    -0.22421    0.14806  -1.514 0.130779
## paid         0.13332    0.20197   0.660 0.509581
## internet    -0.20980    0.26941  -0.779 0.436632
## romantic    -0.35845    0.21132  -1.696 0.090659 .
## freetime     0.11957    0.09821   1.218 0.224150
## absences     0.04516    0.01250   3.612 0.000345 ***
## G1           0.17398    0.05755   3.023 0.002670 **
## G2           0.95445    0.05109  18.681  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.899 on 381 degrees of freedom
## Multiple R-squared:  0.8338, Adjusted R-squared:  0.8282
## F-statistic: 147.1 on 13 and 381 DF,  p-value: < 2.2e-16
```

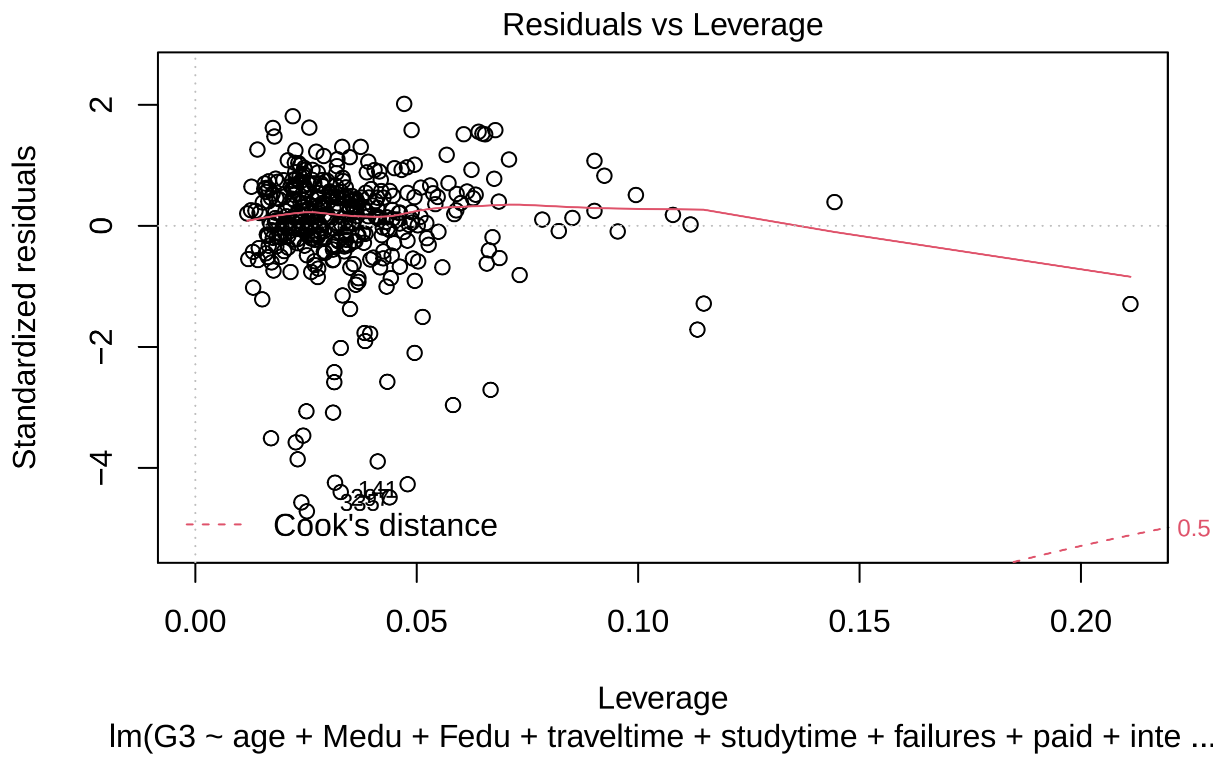
```r
# do plots to visualize
plot(fit)
```



Residuals vs Fitted
Residuals
Fitted values
lm(G3 ~ age + Medu + Fedu + traveltime + studytime + failures + paid + inte ...
5
0
−5
−10
0
5
10
15
20
342
335
265


Normal Q−Q
Standardized residuals
2
0
−2
−4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(G3 ~ age + Medu + Fedu + traveltime + studytime + failures + paid + inte ...
265
335
342

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
Fitted values
lm(G3 ~ age + Medu + Fedu + traveltime + studytime + failures + paid + inte ...
342



From the summary table, only absences, G1, and G2 individually has significant effect to G3. From figure 1, the fit line in general straight, meaning it is generally a linear relationship. There are there data points are standing out meaning they are to far away that the model didn't capture them. They are extreme cases. The model might be improved them by removing them. Figure 2 is the visualization of the real residuals compared against to the theoretical distances from the model. Figure 3 showed the distribution of residuals around the linear model in relation to G3 Figure 4 measured each data point's influence. From the figure, none of the extreme values have a huge impact on the model.

Question 3

Using the model from (2), 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).

1. Performe stepwise backward elimination

```r
step(lm(G3 ~ age + Medu + studytime + failures + internet +
          romantic + absences + G1 + G2, data = math),
     direction = "backward")
```

```
## Start:  AIC=516.62
## G3 ~ age + Medu + studytime + failures + internet + romantic +
##     absences + G1 + G2
##
```

```
##              Df Sum of Sq    RSS    AIC
## - Medu       1      0.32 1389.1 514.71
## - internet   1      1.73 1390.5 515.11
## - studytime  1      5.81 1394.5 516.27
## - failures   1      7.03 1395.8 516.61
## <none>                   1388.7 516.62
## - romantic   1     10.57 1399.3 517.61
## - age        1     12.08 1400.8 518.04
## - G1         1     33.59 1422.3 524.06
## - absences   1     45.46 1434.2 527.34
## - G2         1   1302.17 2690.9 775.90
##
## Step:  AIC=514.71
## G3 ~ age + studytime + failures + internet + romantic + absences +
##     G1 + G2
##
##              Df Sum of Sq    RSS    AIC
## - internet   1      1.55 1390.6 513.15
## - studytime  1      5.78 1394.8 514.35
## <none>                   1389.1 514.71
## - failures   1      7.65 1396.7 514.88
## - romantic   1     10.39 1399.5 515.65
## - age        1     12.76 1401.8 516.32
## - G1         1     33.94 1423.0 522.25
## - absences   1     47.01 1436.1 525.86
## - G2         1   1305.17 2694.2 774.39
##
## Step:  AIC=513.15
## G3 ~ age + studytime + failures + romantic + absences + G1 +
##     G2
##
##              Df Sum of Sq    RSS    AIC
## - studytime  1      6.09 1396.7 512.88
## <none>                   1390.6 513.15
## - failures   1      7.60 1398.2 513.30
## - romantic   1     11.39 1402.0 514.37
## - age        1     11.89 1402.5 514.51
## - G1         1     34.95 1425.5 520.95
## - absences   1     45.68 1436.3 523.92
## - G2         1   1310.52 2701.1 773.40
##
## Step:  AIC=512.88
## G3 ~ age + failures + romantic + absences + G1 + G2
##
##             Df Sum of Sq    RSS    AIC
## - failures  1      5.99 1402.7 512.57
## <none>                  1396.7 512.88
## - age       1     12.54 1409.2 514.41
## - romantic  1     12.73 1409.4 514.46
## - G1        1     33.38 1430.1 520.20
## - absences  1     48.21 1444.9 524.28
## - G2        1   1310.43 2707.1 772.28
##
## Step:  AIC=512.57
```

```
## G3 ~ age + romantic + absences + G1 + G2
##
##             Df Sum of Sq    RSS    AIC
## <none>                   1402.7 512.57
## - romantic  1     13.43 1416.1 514.33
## - age       1     17.24 1419.9 515.39
## - G1        1     37.99 1440.7 521.12
## - absences  1     47.80 1450.5 523.80
## - G2        1   1328.33 2731.0 773.75
```

```
##
## Call:
## lm(formula = G3 ~ age + romantic + absences + G1 + G2, data = math)
##
## Coefficients:
## (Intercept)          age     romantic     absences           G1           G2
##     0.93446     -0.17046     -0.40330      0.04461      0.18089      0.95515
```

The first model created above had an AIC 516.62, within these variables, mom education had the least effect (AIC 514.71) and G2 had the most effect (AIC 775.90). Therefore, Medu was dropped, the AIC then dropped a bit to 514.71. After few times of testing, the final model kept age, romantic, absences, G1, and G2. The equation of the model is: G3 = 0.93446 + (-0.17064) * age + (-0.40330) * absences + 0.18089 * G1 + 0.95515 * G2.

2. Summary the model

```r
fit <- lm(G3 ~ age + failures + romantic + absences +
            G1 + G2,
          data = math)
summary(fit)
```

```
##
## Call:
## lm(formula = G3 ~ age + failures + romantic + absences + G1 +
##     G2, data = math)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1405 -0.4081  0.2834  0.9461  3.7710
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.78162    1.35090   0.579 0.563202
## age         -0.14875    0.07969  -1.867 0.062715 .
## failures    -0.18312    0.14199  -1.290 0.197945
## romantic    -0.39301    0.20895  -1.881 0.060738 .
## absences     0.04480    0.01224   3.660 0.000288 ***
## G1           0.17111    0.05619   3.045 0.002486 **
## G2           0.95084    0.04984  19.080  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.897 on 388 degrees of freedom
## Multiple R-squared:  0.8311, Adjusted R-squared:  0.8285
## F-statistic: 318.2 on 6 and 388 DF,  p-value: < 2.2e-16
```

From the summary table, the r2 value dropped a bit (from 0.8338 to 0.8311) and the adjusted r2 improved a bit (from 0.8285 to 0.8285)

Question 4

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

1. Make prediction using the model

```r
# create a new column predG3 for the predicted G3 using the model
math$predG3 <- predict(fit,
                       newdata = subset(math,
                                        select = c(G3, age,
                                                   failures, romantic,
                                                   absences, G1, G2)))
```

2. Calculate 95% confident interval

```r
# use the first subject's data to calculate, from the summary table in question 3, the standard error i
math[1, 34] - 1.96 * 1.897
```

```
## [1] 1.215453
```

```r
math[1, 34] + 1.96 * 1.897
```

```
## [1] 8.651693
```

Question 5

What is the RMSE for this model – use the entire data set for both training and validation. You may find the residuals() function useful.

```r
# calculate the rooted mean squared error
mathRMSE <- sqrt(mean(residuals(fit) ^ 2))
mathRMSE
```

```
## [1] 1.880407
```

The result on shows, on average, each of the estimate was 1.88 points away from what it should be.

Problem 2

Question 1

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.

```
# create the new PF column
math$PF <- ifelse(math$G3 < 10, "F", "P")
# create dummy codes
math$PF <- as.factor(ifelse(math$PF == "F", 0, 1))
```

Question 2

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.

1. Create the first model using age, mom education, father education, study time, failures, paid, internet, romantic, freetime, absences, G1, and G2.

```
# create the first model
mathGlm_1 <- glm(PF ~ age + Medu + Fedu + traveltime +
                   studytime + failures + paid + internet +
                   romantic + freetime + absences + G1 + G2,
                 data = math,
                 family = binomial)
# summary the model
summary(mathGlm_1)
```

```
##
## Call:
## glm(formula = PF ~ age + Medu + Fedu + traveltime + studytime +
##     failures + paid + internet + romantic + freetime + absences +
##     G1 + G2, family = binomial, data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.08971  -0.02077   0.00291   0.07802   2.14774
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.48586    4.37462  -2.626  0.00865 **
## age          -0.49382    0.19790  -2.495  0.01258 *
## Medu          0.05423    0.31895   0.170  0.86498
## Fedu         -0.62077    0.33130  -1.874  0.06096 .
## traveltime    0.54160    0.33362   1.623  0.10451
## studytime    -0.67876    0.32738  -2.073  0.03814 *
## failures      0.04733    0.32128   0.147  0.88288
```

```
## paid          0.33950    0.48974   0.693  0.48817
## internet     -0.38099    0.61706  -0.617  0.53695
## romantic     -0.49696    0.51938  -0.957  0.33865
## freetime     -0.01800    0.25748  -0.070  0.94426
## absences     -0.02787    0.03027  -0.921  0.35724
## G1            0.43649    0.17909   2.437  0.01480 *
## G2            1.99869    0.33542   5.959 2.54e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.5  on 394  degrees of freedom
## Residual deviance: 122.5  on 381  degrees of freedom
## AIC: 150.5
##
## Number of Fisher Scoring iterations: 9
```

From the summary table, age, studytime, G1, and G2 can individually has signification effect to G3PF. Freetime, failures, and Medu have the highest p values (> 0.8), then is internet, paid, romantic, and absences (0.3 < p < 0.6). I will remove the highest three variables, then try the other 4.

2. Create more models

```r
# only remove the highest 3
mathGlm_2 <- glm(PF ~ age + Fedu + traveltime + studytime + paid +
                   internet + romantic + absences + G1 + G2,
                 data = math,
                 family = binomial)
summary(mathGlm_2)
```

```
##
## Call:
## glm(formula = PF ~ age + Fedu + traveltime + studytime + paid +
##     internet + romantic + absences + G1 + G2, family = binomial,
##     data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.09334  -0.02069   0.00293   0.07858   2.13498
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.44195    4.13799  -2.765  0.00569 **
## age          -0.48935    0.19576  -2.500  0.01243 *
## Fedu         -0.58625    0.25483  -2.301  0.02142 *
## traveltime    0.53846    0.32878   1.638  0.10147
## studytime    -0.68057    0.32355  -2.103  0.03543 *
## paid          0.34297    0.47486   0.722  0.47014
## internet     -0.35965    0.58829  -0.611  0.54097
## romantic     -0.48403    0.50823  -0.952  0.34090
## absences     -0.02664    0.02903  -0.918  0.35876
```

```
## G1             0.42648    0.16838   2.533  0.01131 *
## G2             1.99535    0.32327   6.172 6.73e-10 ***
## ---
## 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: 122.55  on 384  degrees of freedom
## AIC: 144.55
##
## Number of Fisher Scoring iterations: 8
```

```
# remove the highest 3 and internet and paid
mathGlm_3 <- glm(PF ~ age + Fedu + traveltime + studytime + romantic +
                   absences + G1 + G2,
                 data = math,
                 family = binomial)
summary(mathGlm_3)
```

```
##
## Call:
## glm(formula = PF ~ age + Fedu + traveltime + studytime + romantic +
##     absences + G1 + G2, family = binomial, data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.04484  -0.02282   0.00312   0.08233   2.13693
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.56129    4.08594  -2.830  0.00466 **
## age          -0.47400    0.19522  -2.428  0.01518 *
## Fedu         -0.56290    0.24941  -2.257  0.02401 *
## traveltime    0.51752    0.32651   1.585  0.11296
## studytime    -0.64394    0.31407  -2.050  0.04034 *
## romantic     -0.52212    0.50710  -1.030  0.30318
## absences     -0.02876    0.02868  -1.003  0.31603
## G1            0.42415    0.16796   2.525  0.01156 *
## G2            1.96238    0.31719   6.187 6.14e-10 ***
## ---
## 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: 123.43  on 386  degrees of freedom
## AIC: 141.43
##
## Number of Fisher Scoring iterations: 8
```

```
# remove the highest 3 and internet, paid, and romantic
mathGlm_4 <- glm(PF ~ age + Fedu + traveltime + studytime +
```

```r
                   absences + G1 + G2,
                 data = math,
                 family = binomial)
summary(mathGlm_4)
```

```
##
## Call:
## glm(formula = PF ~ age + Fedu + traveltime + studytime + absences +
##     G1 + G2, family = binomial, data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.99318  -0.02210   0.00312   0.08180   2.18064
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.65194    4.11306  -2.833  0.00461 **
## age          -0.46871    0.19565  -2.396  0.01659 *
## Fedu         -0.54390    0.24717  -2.200  0.02777 *
## traveltime    0.51525    0.32409   1.590  0.11187
## studytime    -0.68188    0.31280  -2.180  0.02926 *
## absences     -0.03631    0.02831  -1.282  0.19971
## G1            0.40555    0.16550   2.450  0.01427 *
## G2            1.97038    0.31519   6.251 4.07e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 500.5  on 394  degrees of freedom
## Residual deviance: 124.5  on 387  degrees of freedom
## AIC: 140.5
##
## Number of Fisher Scoring iterations: 8
```

```r
# remove the highest 3 and internet, paid, and absences
mathGlm_5 <- glm(PF ~ age + Fedu + traveltime + studytime +
                   romantic + G1 + G2,
                 data = math,
                 family = binomial)
summary(mathGlm_5)
```

```
##
## Call:
## glm(formula = PF ~ age + Fedu + traveltime + studytime + romantic +
##     G1 + G2, family = binomial, data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.03530  -0.02173   0.00327   0.07963   2.15932
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -11.2026     4.0677  -2.754  0.00589 **
## age          -0.5008     0.1946  -2.573  0.01009 *
## Fedu         -0.5767     0.2473  -2.332  0.01970 *
## traveltime    0.5003     0.3265   1.532  0.12540
## studytime    -0.5826     0.3067  -1.900  0.05745 .
## romantic     -0.6794     0.4863  -1.397  0.16235
## G1            0.4032     0.1682   2.398  0.01650 *
## G2            1.9703     0.3171   6.213 5.21e-10 ***
## ---
## 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: 124.65  on 387  degrees of freedom
## AIC: 140.65
##
## Number of Fisher Scoring iterations: 8
```

```r
# remove the highest 3 and internet, paid, absences, and romantic
mathGlm_6 <- glm(PF ~ age + Fedu + traveltime + studytime +
                   G1 + G2,
                 data = math,
                 family = binomial)
summary(mathGlm_6)
```

```
##
## Call:
## glm(formula = PF ~ age + Fedu + traveltime + studytime + G1 +
##     G2, family = binomial, data = math)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.95332  -0.02303   0.00350   0.07809   2.23118
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.1507     4.1002  -2.720  0.00654 **
## age          -0.5039     0.1954  -2.579  0.00992 **
## Fedu         -0.5574     0.2433  -2.291  0.02198 *
## traveltime    0.4883     0.3224   1.515  0.12986
## studytime    -0.6062     0.3054  -1.985  0.04712 *
## G1            0.3659     0.1635   2.239  0.02517 *
## G2            1.9827     0.3153   6.288 3.22e-10 ***
## ---
## 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: 126.64  on 388  degrees of freedom
## AIC: 140.64
##
## Number of Fisher Scoring iterations: 8
```

From the summaries above, PF ~ age + Fedu + traveltime + studytime + absences + G1 + G2 has the losest AIC value (140.5), the following model selection table (function from MuMIn package) will further compare the models.

```r
# create a model selection table
model.sel(mathGlm_1, mathGlm_2, mathGlm_3,
          mathGlm_4, mathGlm_5, mathGlm_6)
```

```
## Model selection table
##           (Intrc)    absnc     age   falrs    Fedu  fretm     G1    G2   intrn
## mathGlm_4  -11.65 -0.03631 -0.4687         -0.5439        0.4055 1.970
## mathGlm_6  -11.15          -0.5039         -0.5574        0.3659 1.983
## mathGlm_5  -11.20          -0.5008         -0.5767        0.4032 1.970
## mathGlm_3  -11.56 -0.02876 -0.4740         -0.5629        0.4241 1.962
## mathGlm_2  -11.44 -0.02664 -0.4893         -0.5862        0.4265 1.995 -0.3596
## mathGlm_1  -11.49 -0.02787 -0.4938 0.04733 -0.6208 -0.018 0.4365 1.999 -0.3810
##              Medu   paid   rmntc   stdyt  trvlt          family df  logLik
## mathGlm_4                        -0.6819 0.5153 binomial(logit)  8 -62.252
## mathGlm_6                        -0.6062 0.4883 binomial(logit)  7 -63.321
## mathGlm_5                -0.6794 -0.5826 0.5003 binomial(logit)  8 -62.326
## mathGlm_3                -0.5221 -0.6439 0.5175 binomial(logit)  9 -61.716
## mathGlm_2         0.3430 -0.4840 -0.6806 0.5385 binomial(logit) 11 -61.275
## mathGlm_1 0.05423 0.3395 -0.4970 -0.6788 0.5416 binomial(logit) 14 -61.251
##            AICc delta weight
## mathGlm_4 140.9  0.00  0.276
## mathGlm_6 140.9  0.05  0.269
## mathGlm_5 141.0  0.15  0.257
## mathGlm_3 141.9  1.02  0.166
## mathGlm_2 145.2  4.36  0.031
## mathGlm_1 151.6 10.73  0.001
## Models ranked by AICc(x)
```

The table listed in rank order based on decreasing quality of fit. Model 4 has the lowest AIC and highest weight. Therefore, I will use model 4 for prediction next.

Question 3

State the regression equation.

From question 2, the equation is PF = (-11.65194) + (-0.46871) * age + (-0.54390) * Fedu + 0.51525 * traveltime + (-0.68188) * study + 0.03631 * absences + 0.40555 * G1 + 1.97038 * G2.

Question 4

What is the accuracy of your model? Use the entire data set for both training and validation

1. Predict PF using the model selected above

```r
math$G3PF <- round(predict(mathGlm_4, math, type = "response"))
```

2. Evaluate the model performance

```r
# create a cross table
CrossTable(math$PF, math$G3PF,
           prop.chisq = FALSE,
           prop.c = FALSE,
           prop.r = FALSE,
           dnn = c("actualPF", "predictedPF"))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  395
##
##
##              | predictedPF
##     actualPF |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |       116 |        14 |       130 |
##              |     0.294 |     0.035 |           |
## -------------|-----------|-----------|-----------|
##            1 |        12 |       253 |       265 |
##              |     0.030 |     0.641 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       128 |       267 |       395 |
## -------------|-----------|-----------|-----------|
##
##
```

From the table, the prediction accuracy is (116 + 253) / 395 = 0.934

Problem 3

Question 1

Implement the example from the textbook on pages 205 to 217 for the data set on white wines.

1. Read in data file

```r
wine <- read.csv("whitewines.csv")
```

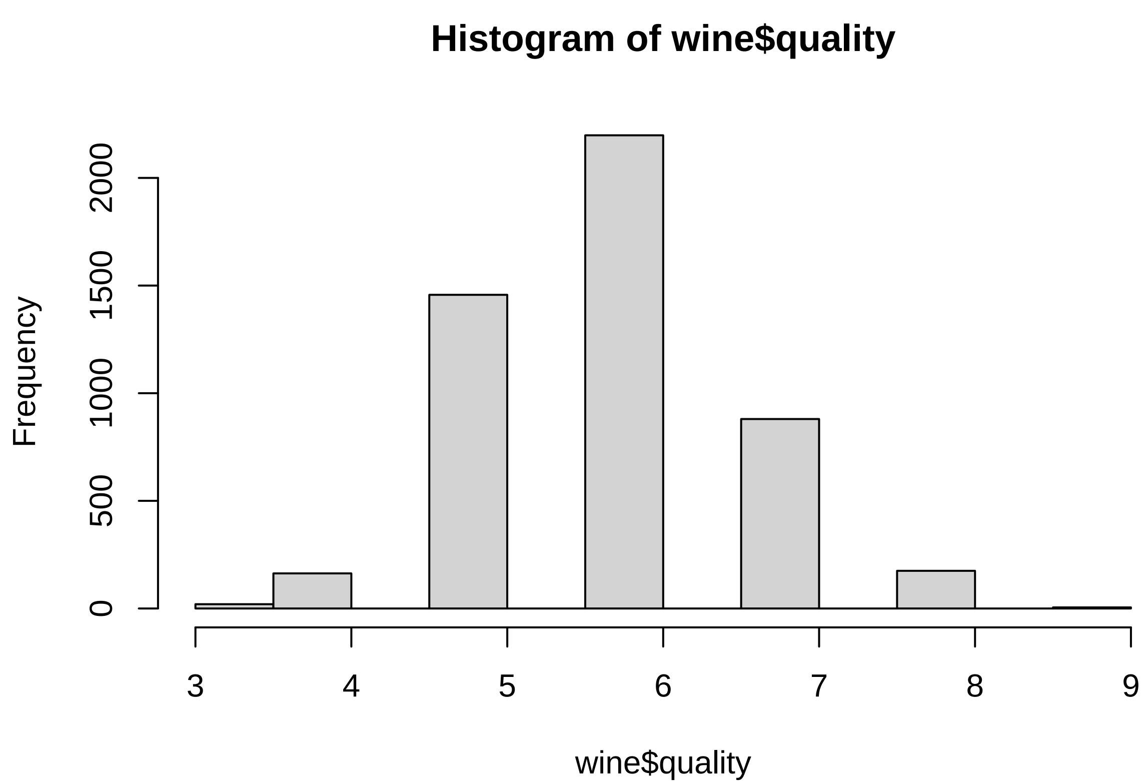
2. Check the structure of the dataset

```r
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 ...
```

3. Check the distribution of qulaity column

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



From the figure, it is a fairly normal distribution.

4. Devide dataset to training data and test data

```r
wine_train <- wine[1:3750, ]
wine_test <- wine[3751:4898, ]
```

5. Train a model

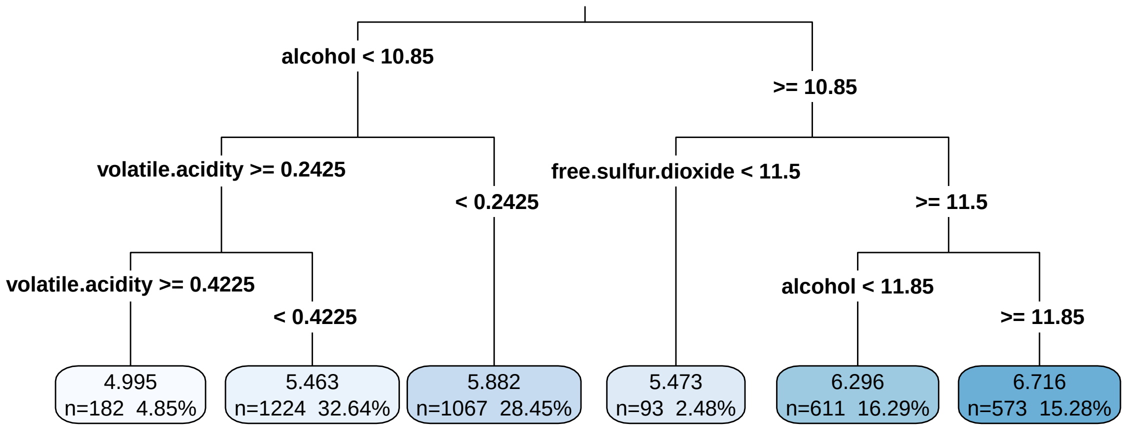
```r
# specify quality as the outcome variable and all the other columns as predictors
m.rpart <- rpart(quality ~ ., data = wine_train)
m.rpart
```

```
## 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
# uncommand summary to see the detail of the tree
#summary(m.rpart)
```

6. Visulaize the tree

```r
# plot the tree, digits control the number of numeric digits, fallen.leaves
# forces the leaf nodes to be aligned at the bottom of the plot, type and
# extra affect the way the tree being labeled
rpart.plot(m.rpart, digits = 4,
           fallen.leaves = TRUE,
           type = 3, extra = 101)
```



7. Evaluating the performance

```r
# make the prediction
p.rpart <- predict(m.rpart, wine_test)
# a quick overview of the validation data and predicted data
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
# check the correlation between the predicted and actual quality values
cor(p.rpart, wine_test$quality)
```

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

From the summary data, the model didn't correctly predict the extreme cases (min and max). It is fairly well between the first and third quartile. The correlation number indicated a well correlation between the predictions and true value. Following code will further measure the performance with the mean absolute error.

```
# create the function to calculate the mean absolute error
MAE <- function(actual, predicted){
  mean(abs(actual - predicted))
}
# calculate MAE between predicted value and true value
MAE(p.rpart, wine_test$quality)
```

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

The number indicates that on average, the difference between the model's predictions and the true quality score was about 0.57. Since there are not a lot of extreme values, use the mean value as the predict value might also be good.

```
# calculate the mean predicted quality value
mean(wine_train$quality)
```

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

```
# calculate MAE between mean value and true value
MAE(5.88, wine_test$quality)
```

```
## [1] 0.5778397
```

The mean absolute error is 0.58.

7. Improve the model performance

```
# build model tree using M5' algorithm
m.m5p <- M5P(quality ~ ., data = wine_train)
# summary the tree
summary(m.m5p)
```

```
##
## === 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
```

8. Make prediction using the new model

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

9. Evaluate the new model

```
# summary the prediction
summary(p.m5p)
```

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

```
# calculate the correlation
cor(p.m5p, wine_test$quality)
```

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

```
# calculate teh mean absolute error
MAE(p.m5p, wine_test$quality)
```

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

Question 2

Calculate the RMSE for the model.

```
# the original model's RMSE
sqrt(mean(wine_test$quality - p.rpart) ^ 2)
```

```
## [1] 0.1505778
```

```
# the improved model's RMSE
sqrt(mean(wine_test$quality - p.m5p) ^ 2)
```

```
## [1] 118.1177
```

Based on the original model, the predicted value is 0.15 points away from the actual value on average. However, the M5 model's result didn't make sense, this model might not be appropriate to this data set.

Try different algorism

Cubist

```
# train Cubist model
m.cubist <- cubist(x = wine_train[-12], y = wine_train$quality)
# summary the tree
summary(m.cubist)
```

```
##
## Call:
## cubist.default(x = wine_train[-12], y = wine_train$quality)
##
##
## Cubist [Release 2.07 GPL Edition]  Mon Nov  2 20:16:01 2020
## ---------------------------------
##
##     Target attribute ‘outcome’
##
## Read 3750 cases (12 attributes) from undefined.data
##
## Model:
##
##   Rule 1: [918 cases, mean 5.3, range 3 to 7, est err 0.5]
##
##     if
##  volatile.acidity > 0.26
##  alcohol <= 10.2
##     then
##  outcome = 66.6 + 0.187 alcohol + 0.041 residual.sugar - 65 density
##           - 1.38 volatile.acidity + 0.5 pH + 0.0028 free.sulfur.dioxide
##
##   Rule 2: [177 cases, mean 5.5, range 4 to 8, est err 0.5]
##
##     if
##  citric.acid > 0.42
##  residual.sugar <= 14.05
##  free.sulfur.dioxide > 49
##     then
##  outcome = 32.5 + 0.379 alcohol - 0.024 residual.sugar - 31 density
##           - 0.54 volatile.acidity + 0.15 sulphates
##           + 0.0003 total.sulfur.dioxide + 0.07 pH + 0.4 chlorides
##           + 0.01 fixed.acidity
##
##   Rule 3: [490 cases, mean 5.7, range 3 to 8, est err 0.5]
##
##     if
##  volatile.acidity <= 0.26
##  residual.sugar <= 12.75
##  free.sulfur.dioxide <= 49
##  alcohol <= 10.2
##     then
##  outcome = 253.6 - 252 density + 0.102 residual.sugar
##           - 2.63 volatile.acidity + 0.0149 free.sulfur.dioxide
##           + 1.27 sulphates + 0.52 pH + 0.012 alcohol
##
##   Rule 4: [71 cases, mean 5.8, range 5 to 7, est err 0.4]
##
##     if
##  fixed.acidity <= 7.5
##  volatile.acidity <= 0.26
##  residual.sugar > 14.05
##  alcohol > 9.1
```

```
##     then
##  outcome = 127.2 - 125 density + 0.055 residual.sugar
##            - 2.47 volatile.acidity + 0.24 fixed.acidity + 0.67 sulphates
##            + 0.0017 total.sulfur.dioxide + 1.8 chlorides + 0.23 pH
##            - 0.0015 free.sulfur.dioxide + 0.013 alcohol
##
##   Rule 5: [446 cases, mean 5.8, range 3 to 9, est err 0.5]
##
##     if
##  citric.acid <= 0.42
##  residual.sugar <= 14.05
##  free.sulfur.dioxide > 49
##     then
##  outcome = 29.6 + 0.372 alcohol + 2.81 citric.acid
##            - 2.94 volatile.acidity - 28 density + 0.013 residual.sugar
##            + 0.13 sulphates + 0.0003 total.sulfur.dioxide
##            + 0.01 fixed.acidity
##
##   Rule 6: [451 cases, mean 5.9, range 3 to 8, est err 0.7]
##
##     if
##  free.sulfur.dioxide <= 20
##  alcohol > 10.2
##     then
##  outcome = 16.2 + 0.0537 free.sulfur.dioxide + 0.311 alcohol
##            - 2.63 volatile.acidity + 0.037 residual.sugar
##            - 0.2 fixed.acidity - 13 density + 0.08 pH
##
##   Rule 7: [113 cases, mean 5.9, range 5 to 7, est err 0.5]
##
##     if
##  fixed.acidity <= 7.5
##  volatile.acidity <= 0.26
##  residual.sugar > 14.05
##  alcohol <= 9.1
##     then
##  outcome = -8.3 + 2.204 alcohol - 0.143 residual.sugar
##            + 0.0066 total.sulfur.dioxide - 1.65 sulphates
##            - 0.0092 free.sulfur.dioxide - 3 density
##
##   Rule 8: [35 cases, mean 6.2, range 3 to 8, est err 0.8]
##
##     if
##  fixed.acidity > 7.5
##  volatile.acidity <= 0.26
##  residual.sugar > 14.05
##  alcohol <= 10.2
##     then
##  outcome = 29.5 - 0.451 residual.sugar - 19.04 volatile.acidity
##            - 0.804 alcohol - 39.4 chlorides + 0.0127 total.sulfur.dioxide
##            - 0.64 fixed.acidity
##
##   Rule 9: [46 cases, mean 6.3, range 5 to 7, est err 0.4]
##
```

```
##     if
##  volatile.acidity <= 0.26
##  residual.sugar > 12.75
##  residual.sugar <= 14.05
##  free.sulfur.dioxide <= 49
##  alcohol <= 10.2
##     then
##  outcome = 11.9 - 13.32 volatile.acidity + 0.0216 total.sulfur.dioxide
##            - 8.01 sulphates - 0.0521 free.sulfur.dioxide - 16.2 chlorides
##
##   Rule 10: [1410 cases, mean 6.4, range 3 to 9, est err 0.6]
##
##     if
##  free.sulfur.dioxide > 20
##  alcohol > 10.2
##     then
##  outcome = 247.3 - 250 density + 0.11 residual.sugar + 1.26 pH
##            + 0.116 alcohol + 1.04 sulphates + 0.11 fixed.acidity
##            - 0.26 volatile.acidity + 0.0012 free.sulfur.dioxide
##
##
## Evaluation on training data (3750 cases):
##
##     Average  |error|               0.4
##     Relative |error|              0.63
##     Correlation coefficient       0.67
##
##
##  Attribute usage:
##    Conds  Model
##
##     85%    99%    alcohol
##     73%    84%    free.sulfur.dioxide
##     40%    97%    volatile.acidity
##     33%    99%    residual.sugar
##     15%    11%    citric.acid
##      5%    62%    fixed.acidity
##            98%    density
##            85%    pH
##            66%    sulphates
##            21%    total.sulfur.dioxide
##             8%    chlorides
##
##
## Time: 0.2 secs
```

1. Make prediction of using Cubist tree

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   3.315   5.574   6.093   6.028   6.437   7.647
```

```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
```

From the summary data, the model did pretty good on capturing extreme data, also for all the data in the middle

2. Evaluate the preformance of Cubist tree

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

```
## [1] 0.5683117
```

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

```
## [1] 0.5306253
```

```r
sqrt(mean(wine_test$quality - p.cubist) ^ 2)
```

```
## [1] 0.1798368
```

The correlation is 0.56, which is good. MAE = 0.53 indicating that on average, the difference between the model's predictions and the true quality score was about 0.53, it is also lower then the original model. The RMSE value is 0.18, meaning each of the estimate was 0.18 points away from what it should be.

Then I also tried another tree model as below ## Random Forest

```r
# train the model
m.forest <- randomForest(quality ~ ., data = wine_train)
summary(m.forest)
```

```
##                 Length Class  Mode     
## call               3   -none- call     
## type               1   -none- character
## predicted       3750   -none- numeric  
## mse              500   -none- numeric  
## rsq              500   -none- numeric  
## oob.times       3750   -none- numeric  
## importance        11   -none- numeric  
## importanceSD       0   -none- NULL     
## localImportance    0   -none- NULL     
## proximity          0   -none- NULL     
## ntree              1   -none- numeric  
## mtry               1   -none- numeric  
## forest            11   -none- list     
## coefs              0   -none- NULL
```

```
## y               3750   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```r
# make prediction
p.forest <- predict(m.forest, wine_test)
summary(p.forest)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.612   5.579   6.014   6.017   6.432   7.367
```

```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
```

From the summary, the model did fairly well, but not as good as Cubist model on estreme values.

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

```
## [1] 0.6079464
```

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

```
## [1] 0.5188401
```

```r
sqrt(mean(wine_test$quality - p.forest) ^ 2)
```

```
## [1] 0.1681036
```

The correlation was the strongest so far (0.61), and the MAE (0.52) and RMSE (0.16) are the lowest so far. Given this dataset doesn't have a lot of extreme values, this model might be the best for this dataset compared to rpart and Cubist.