Assignment 2

Sanket Praveen Patil

2023-10-18

Problem 1

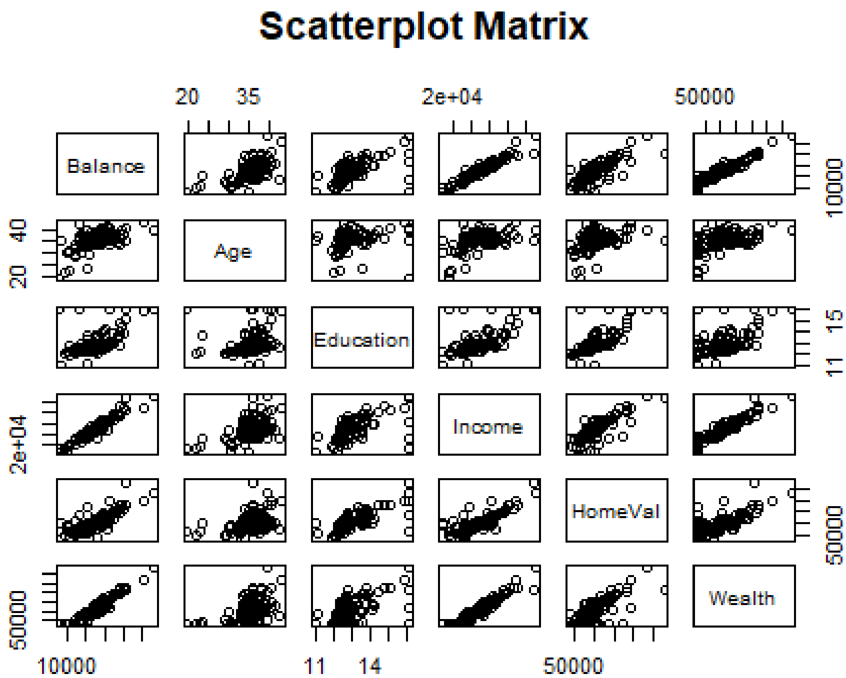
Importing data in R

```
df = read.table("Bankingfull.txt", header = T)
dim(df)

## [1] 102   6

head(df)

##    Age Education Income HomeVal Wealth Balance
## 1 35.9      14.8  91033  183104 220741   38517
## 2 37.7      13.8  86748  163843 223152   40618
## 3 36.8      13.8  72245  142732 176926   35206
## 4 35.3      13.2  70639  145024 166260   33434
## 5 35.3      13.2  64879  135951 148868   28162
## 6 34.8      13.7  75591  155334 188310   36708
```

Question 1 :

Creating Scatterpot matrix:

```
pairs(~Balance+Age+Education+Income+HomeVal+Wealth,data = df,main = "Scatterplot Matrix")
```



- By looking at the scatterplot matrix, we can see that Variables Income and Wealth both has strong positive linear relation with Balance. Also we are not able to see any outliers in those two variables.
- Variable HomeVal has also a strong linear relation with Balance.
- Variables Age and Education also have linear relation with Balance but not as strong as other variables. Also these two variables have outliers as well.
- All variables have strong or minimal linear relation with each other.

Question 2 :

```
cor(df)

##                Age Education    Income   HomeVal    Wealth   Balance
## Age       1.0000000 0.1734071 0.4771474 0.3864931 0.4680918 0.5654668
## Education 0.1734071 1.0000000 0.5753940 0.7535211 0.4694130 0.5548807
## Income    0.4771474 0.5753940 1.0000000 0.7953552 0.9466654 0.9516845
## HomeVal   0.3864931 0.7535211 0.7953552 1.0000000 0.6984778 0.7663871
## Wealth    0.4680918 0.4694130 0.9466654 0.6984778 1.0000000 0.9487117
## Balance   0.5654668 0.5548807 0.9516845 0.7663871 0.9487117 1.0000000
```

- By looking at corelation matrix, we can see that variable Wealth and Income have strong positive correlation with target variable Balance.
- Variable Age, Education and HomeVal have moderate positive correlation with dependent variable Balance.
- Variable Age and Education have the weakest correlation between them.

Question 3 :

```
model_M1 <- lm(Balance~Age+Education+Income+HomeVal+Wealth, data=df)

# VIF calculation
library(car)

## Loading required package: carData

vif(model_M1)

##       Age Education    Income   HomeVal    Wealth
##  1.342764  2.456706 14.901724  4.382999 10.714276
```

- We checked the VIF statistics for the above model and found that vaiables Income and Wealth have VIF factor > 10.
- Hence we can conclude that there is a problem of multicollinearity with variables Income and Wealth.

Question 4 :

#a)

```
summary(model_M1)

##
## Call:
## lm(formula = Balance ~ Age + Education + Income + HomeVal + Wealth,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5376.9 -1110.8   -77.2   872.3  7732.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.071e+04  4.261e+03  -2.514 0.013613 *
## Age          3.187e+02  6.099e+01   5.225 1.01e-06 ***
## Education    6.219e+02  3.190e+02   1.950 0.054135 .
## Income       1.463e-01  4.078e-02   3.588 0.000527 ***
## HomeVal      9.183e-03  1.104e-02   0.832 0.407505
## Wealth       7.433e-02  1.119e-02   6.643 1.85e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2056 on 96 degrees of freedom
## Multiple R-squared:  0.9469, Adjusted R-squared:  0.9441
## F-statistic: 342.4 on 5 and 96 DF,  p-value: < 2.2e-16
```

- In the above model, we can see the Income and Wealth variables have vif value >10. First we will try to refit the model by removing the variable Income as it has highest VIF value.
- Also variable HomeVal have greater P value hence we will remove that variable as well.

```
model_M2 <- lm(Balance~Age+Education+Wealth, data=df)
vif(model_M2)

##       Age Education    Wealth
##  1.285119  1.287161  1.598760

summary(model_M2)

##
## Call:
## lm(formula = Balance ~ Age + Education + Wealth, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7330.6 -1096.7    -5.5   872.9  7087.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.773e+04  3.802e+03  -4.664 9.80e-06 ***
## Age          3.678e+02  6.460e+01   5.694 1.30e-07 ***
## Education    1.300e+03  2.500e+02   5.202 1.08e-06 ***
## Wealth       1.165e-01  4.680e-03  24.887  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2226 on 98 degrees of freedom
## Multiple R-squared:  0.9365, Adjusted R-squared:  0.9345
## F-statistic: 481.5 on 3 and 98 DF,  p-value: < 2.2e-16
```

R-squared and adjusted R-squared

```
summary(model_M1)$adj.r.squared

## [1] 0.9441433

summary(model_M2)$adj.r.squared

## [1] 0.9345196
```

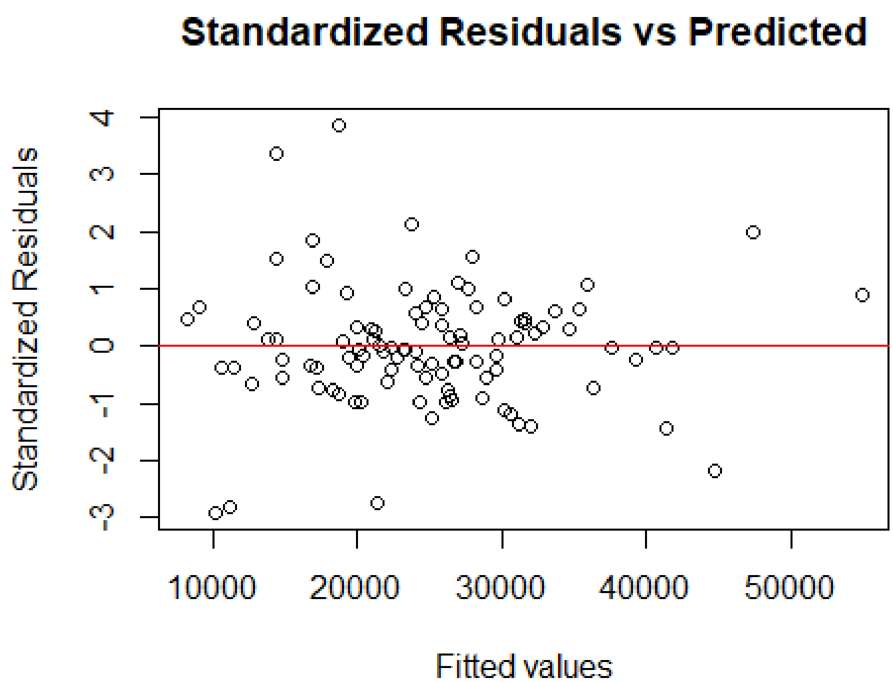
- By removing variables Income and HomeVal, we refit the model and checked the R2 and Adj R2 for both model_M1 and model_M2.
- We can see that model_M1 has better R2 and Adj R2 values than model_M2.

#b)

Residual Analysis

We will take model_M1 into account as it has better adj. R2.

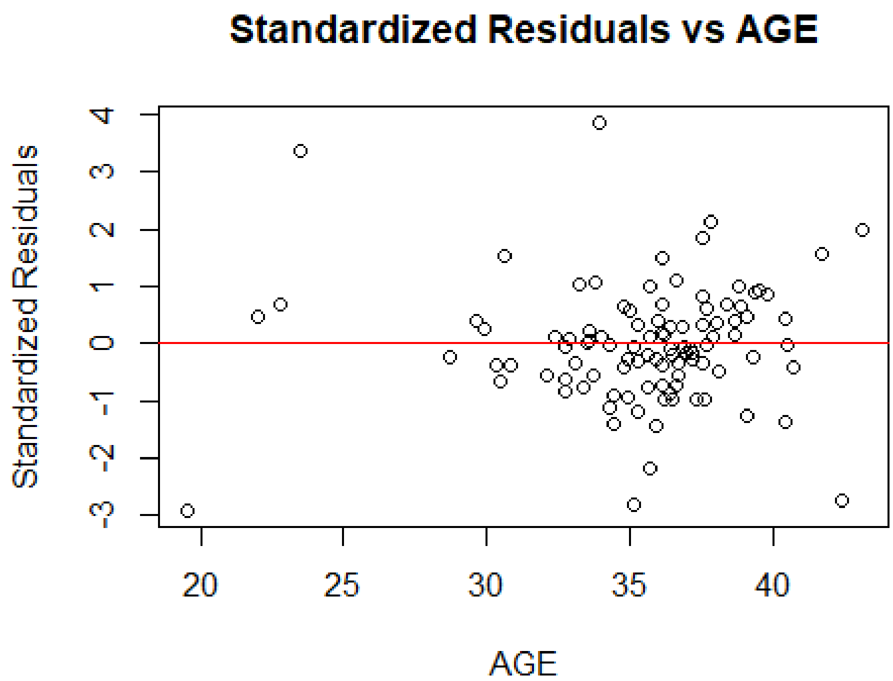
```
# Standardized Residuals vs Predicted
plot(fitted(model_M1), rstandard(model_M1), main="Standardized Residuals vs Predicted",
xlab="Fitted values", ylab="Standardized Residuals")
abline(h=0, col="red")  # Add a horizontal line at y=0 for reference
```



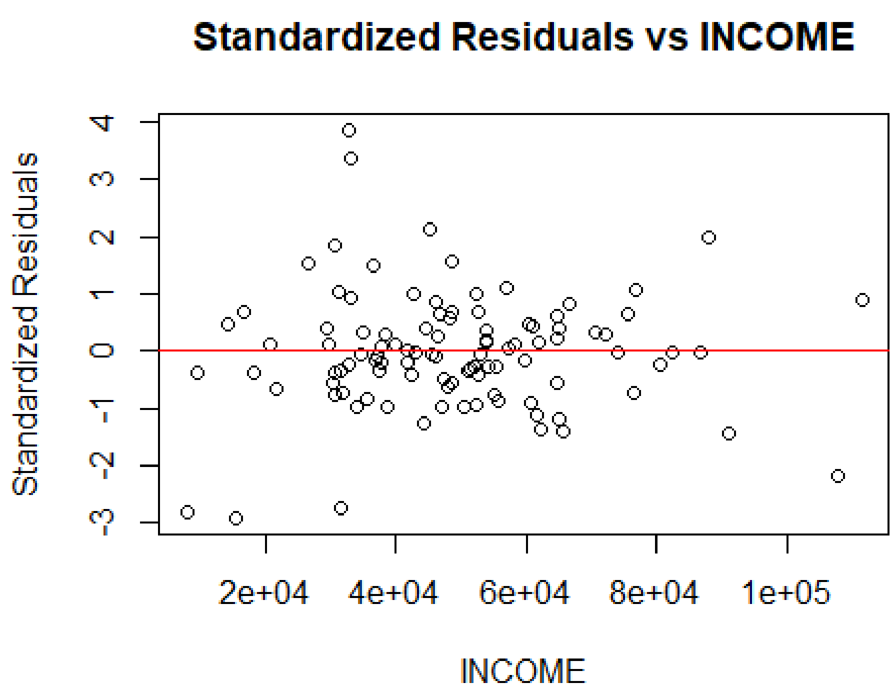
- By looking at plot, we can see there is less variation of residuals hence we can say model is good. There are also 2 to 3 outlier points.

Standardized Residuals vs X-variables

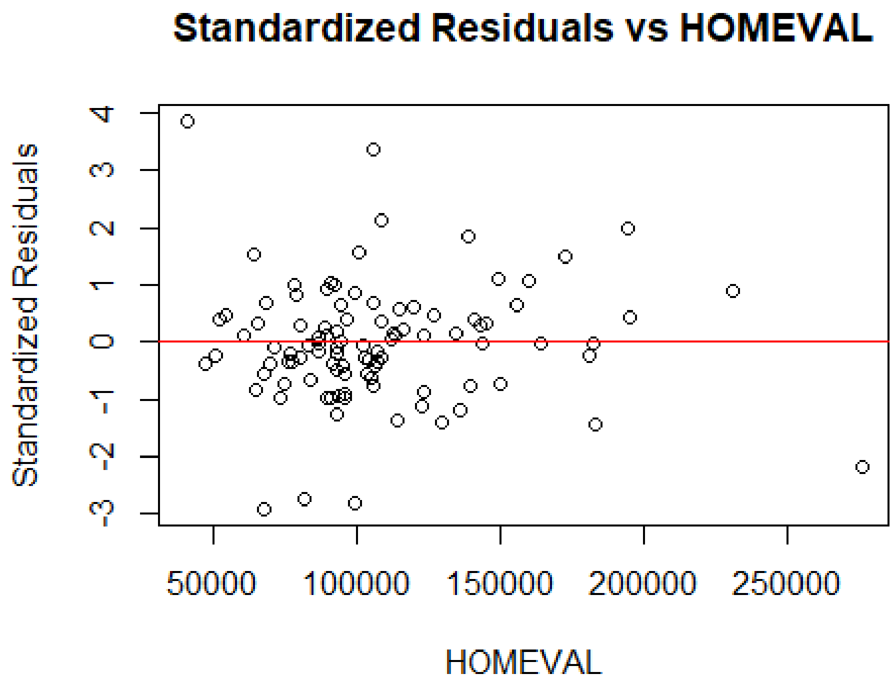
```
plot(df$Age, rstandard(model_M1), main="Standardized Residuals vs AGE", xlab="AGE",
ylab="Standardized Residuals")
abline(h=0, col="red")  # Add a horizontal line at y=0 for reference
```



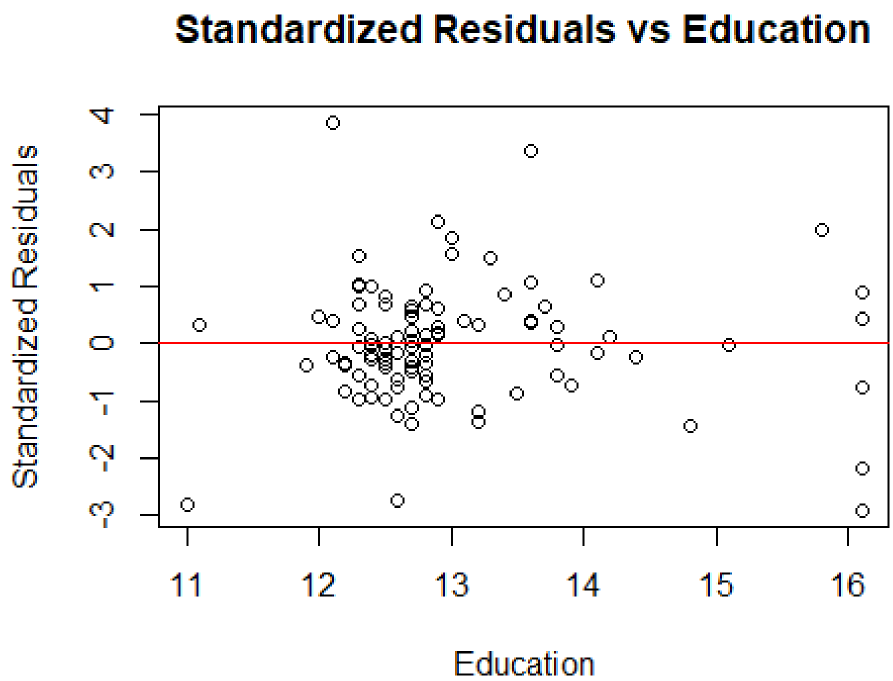

```
plot(df$Income, rstandard(model_M1), main="Standardized Residuals vs INCOME", xlab="INCOME",
ylab="Standardized Residuals")
abline(h=0, col="red")
```



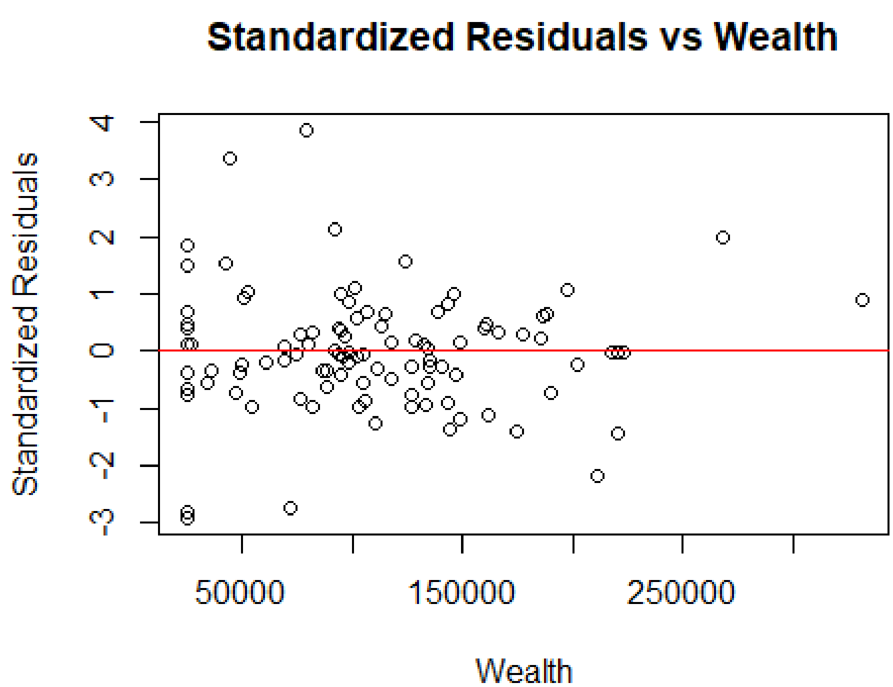
```r
plot(df$HomeVal, rstandard(model_M1), main="Standardized Residuals vs HOMEVAL",
xlab="HOMEVAL", ylab="Standardized Residuals")
abline(h=0, col="red")
```




```r
plot(df$Education, rstandard(model_M1), main="Standardized Residuals vs Education",
xlab="Education", ylab="Standardized Residuals")
abline(h=0, col="red")
```

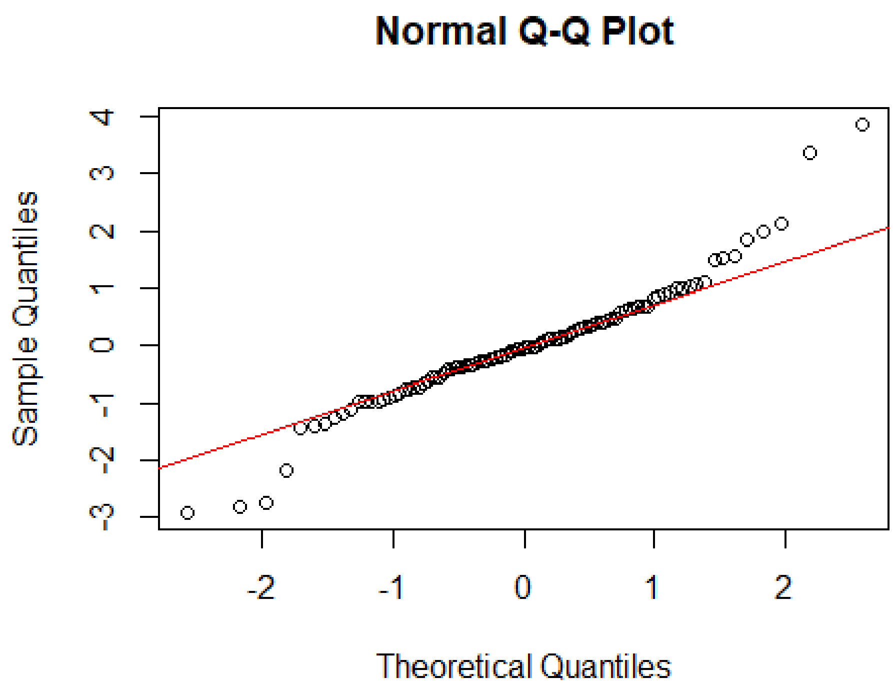



```
plot(df$Wealth, rstandard(model_M1), main="Standardized Residuals vs Wealth", xlab="Wealth",
ylab="Standardized Residuals")
abline(h=0, col="red")
```



Normal Plot of Residuals

```
qqnorm(rstandard(model_M1), main="Normal Q-Q Plot")
qqline(rstandard(model_M1), col="red")
```



- By looking at the QQ plot, we can see the points are following the line hence we can say it is a good model.

#c)

Finding outliers

```
residuals_standardized <- rstandard(model_M1)
outliers_indices <- which(residuals_standardized > 3)

# Extract values with standardized residuals greater than 3
outliers_values <- residuals_standardized[outliers_indices]
```

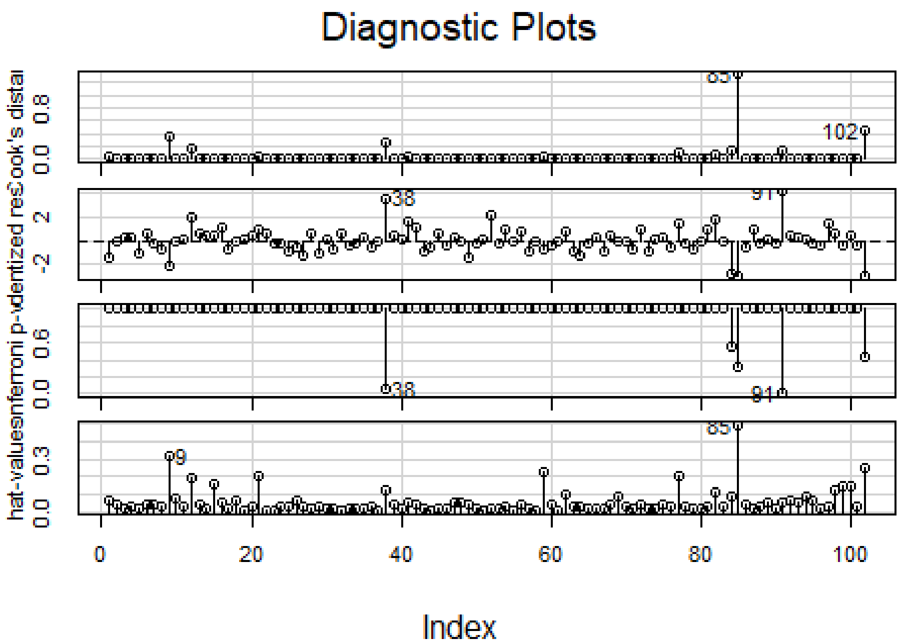
Show the indices and values of outliers

```
cat("Indices of outliers:", outliers_indices, "\n")

## Indices of outliers: 38 91
```

```r
cat("Values of outliers:", outliers_values, "\n")
```

```
## Values of outliers: 3.377844 3.867728
```

```r
cooksd <- cooks.distance(model_M1)
# Find indices of influential points with Cook's distance > 1
influential_indices <- which(cooksd > 1)

# influenceIndex Plot
influenceIndexPlot(model_M1)
```



- As there are outliers(Standardized Residuals>3) but the count of those outliers is less and also there is less variation in the residuals, we can conclude that it is a good model.

#d)

```r
library(dplyr)
```

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

## The following object is masked from 'package:car':
##
##     recode

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

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

```r
library(QuantPsyc)
```

```
## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

## Loading required package: purrr

##
## Attaching package: 'purrr'

## The following object is masked from 'package:car':
##
##     some

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'QuantPsyc'

## The following object is masked from 'package:base':
##
##     norm
```

```r
lm.beta(model_M1)
```

```
##        Age  Education     Income    HomeVal     Wealth
## 0.14239029 0.07186393 0.32572524 0.04095974 0.51136385
```

- By looking at the standardized coefficients, we can conclude that variable "Wealth" has the strongest effect on target variable variable "Balance".

Question 5 :

New data for prediction

```
new_df <- data.frame(Age = 34, Education = 13, Income = 64000, HomeVal = 140000, Wealth =
160000)
```

Prediction and Confidence Interval

```
Predicted_values <- data.frame(predict(model_M1, newdata=new_df, interval="confidence"))
```

Print in the desired format

```
cat("Predicted average bank balance =", Predicted_values[, 1], "\n")

## Predicted average bank balance = 30751.53

cat("Lower 95% Confidence Interval =", Predicted_values[, 2], "\n")

## Lower 95% Confidence Interval = 29952.27

cat("Upper 95% Confidence Interval =", Predicted_values[, 3], "\n")

## Upper 95% Confidence Interval = 31550.78
```

Problem 2 :

Importing data in R

```
data = read.csv("pgatour2006_small.csv", header = T)
dim(data)

## [1] 196   7

head(data)

##              Name PrizeMoney DrivingAccuracy   GIR PuttingAverage
## 1   Aaron Baddeley      60661           60.73 58.26          1.745
## 2       Adam Scott     262045           62.00 69.12          1.767
## 3      Alex Aragon       3635           51.12 59.11          1.787
## 4       Alex Cejka      17516           66.40 67.70          1.777
## 5      Arjun Atwal      16683           63.24 64.04          1.761
## 6 Arron Oberholser     107294           62.53 69.27          1.775
##   BirdieConversion PuttsPerRound
## 1            31.36         27.96
```

```
## 2          30.39       29.28
## 3          29.89       29.20
## 4          29.33       29.46
## 5          29.32       28.93
## 6          29.20       29.56
```

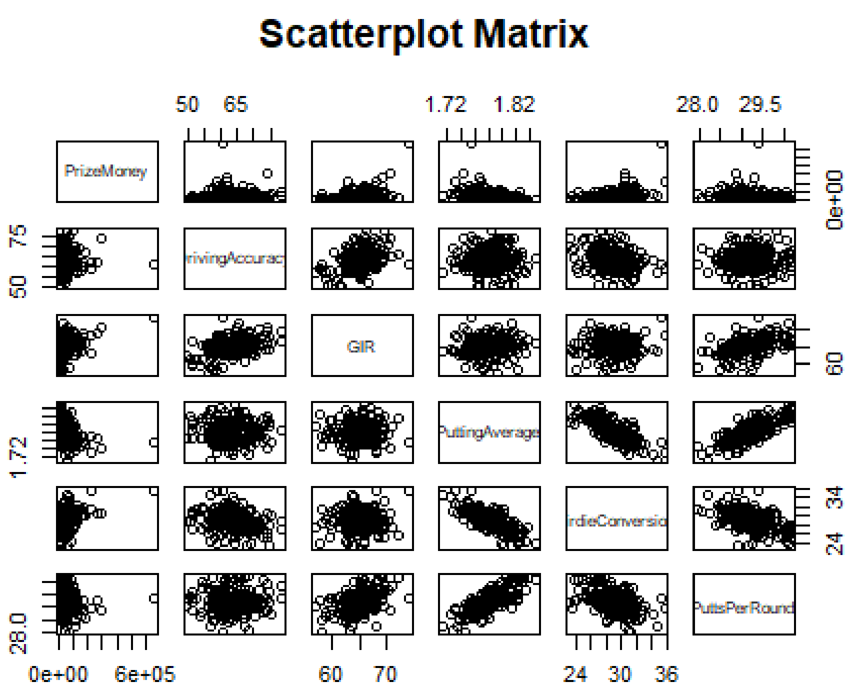
Remove the variable “Name” as it has unique values.

```
data <- data[, !names(data) %in% "Name"]
```

Question 1 :

Creating Scatterpot matrix:

```
pairs(~PrizeMoney+DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+PuttsPerRound,data = data,main = "Scatterplot Matrix")
```

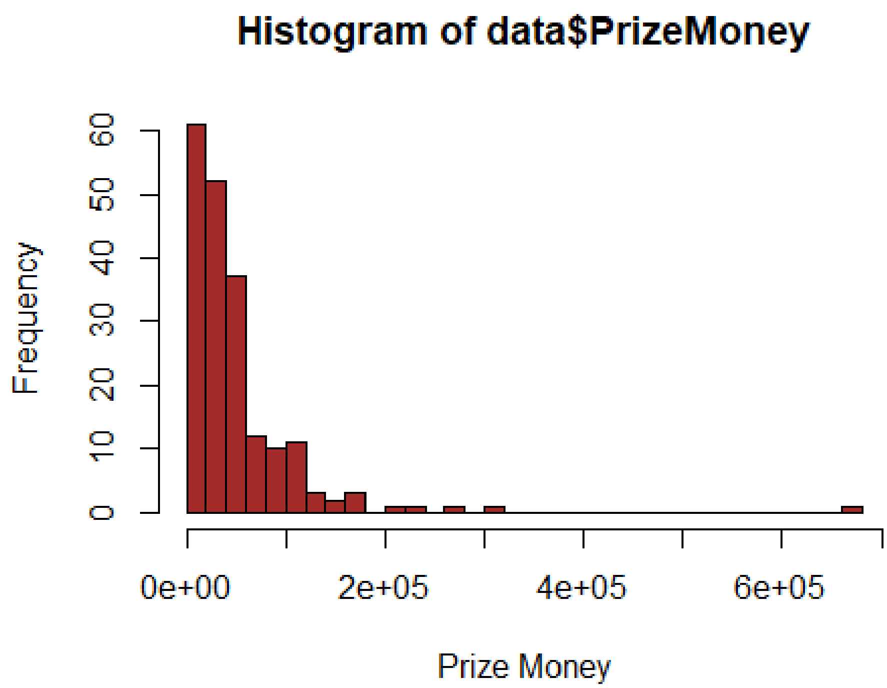


- By looking at the scatterplot matrix, we can conclude that there is no linear relationship between PrizeMoney and any of the independent variable.
- There is inverse linear relationsip between variables “PuttingAverage” and “BirdieConversion”.

Question 2 :

Histogram of PrizeMoney

```
hist(data$PrizeMoney,xlab = "Prize Money",col = "brown",breaks = 30, border = "black")
```



- By looking at the histogram, we can say that the data is highly right skewed. There are more records with less pricemoney than greater pricemoney.

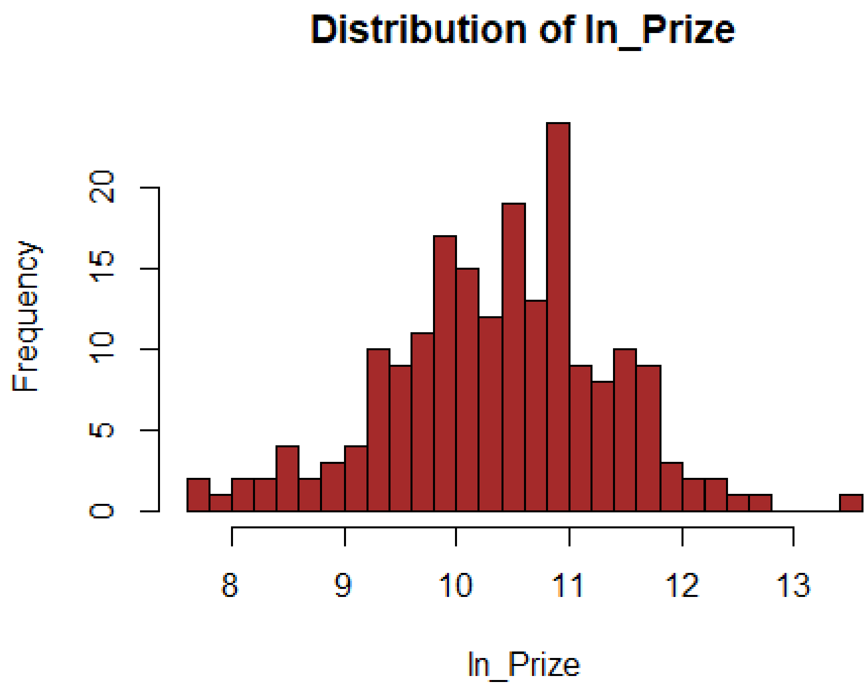
Question 3 :

Log transformation

```
data$ln_Prize <- log(data$PrizeMoney)
```

Histogram of ln_Prize

```
hist(data$ln_Prize, main="Distribution of ln_Prize", xlab = "ln_Prize",col = "brown",breaks = 30, border
= "black")
```



- After applying a log transformation to the variable 'PrizeMoney', the distribution of the data appears to approximate a normal distribution.

Question 4 :

Checking correlation

```
head(data)

##   PrizeMoney DrivingAccuracy   GIR PuttingAverage BirdieConversion
## 1      60661           60.73 58.26          1.745            31.36
## 2     262045           62.00 69.12          1.767            30.39
## 3       3635           51.12 59.11          1.787            29.89
## 4      17516           66.40 67.70          1.777            29.33
## 5      16683           63.24 64.04          1.761            29.32
## 6     107294           62.53 69.27          1.775            29.20
##   PuttsPerRound  ln_Prize
## 1         27.96 11.013056
## 2         29.28 12.476272
## 3         29.20  8.198364
## 4         29.46  9.770870
## 5         28.93  9.722146
## 6         29.56 11.583328

cor(data)
```

```
##                  PrizeMoney DrivingAccuracy        GIR PuttingAverage
## PrizeMoney        1.00000000      0.02467704 0.41021935    -0.31305150
## DrivingAccuracy   0.02467704      1.00000000 0.41635604    -0.02558269
## GIR               0.41021935      0.41635604 1.00000000     0.05880737
## PuttingAverage   -0.31305150     -0.02558269 0.05880737     1.00000000
## BirdieConversion  0.41342953     -0.25212523 0.02685014    -0.76795939
## PuttsPerRound    -0.11249143      0.06031385 0.48083985     0.79168281
## ln_Prize          0.74731908      0.18167291 0.50489317    -0.43011169
##                  BirdieConversion PuttsPerRound   ln_Prize
## PrizeMoney             0.41342953   -0.11249143  0.7473191
## DrivingAccuracy       -0.25212523    0.06031385  0.1816729
## GIR                    0.02685014    0.48083985  0.5048932
## PuttingAverage        -0.76795939    0.79168281 -0.4301117
## BirdieConversion       1.00000000   -0.50072564  0.4673991
## PuttsPerRound         -0.50072564    1.00000000 -0.1832980
## ln_Prize               0.46739910   -0.18329803  1.0000000
```

First we will train the model with all independent variables to predict Ln_Prize

```
model_m1 <- lm(ln_Prize ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+PuttsPerRound,
data=data)
```

a)

```
summary(model_m1)

##
## Call:
## lm(formula = ln_Prize ~ DrivingAccuracy + GIR + PuttingAverage +
##     BirdieConversion + PuttsPerRound, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55696 -0.51250 -0.08005  0.45090  2.11898
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.2410192  7.1611241   1.151 0.251261
## DrivingAccuracy  -0.0007584  0.0116109  -0.065 0.947992
## GIR               0.2687898  0.0287938   9.335  < 2e-16 ***
## PuttingAverage    8.7467774  5.3734220   1.628 0.105228
## BirdieConversion  0.1523018  0.0408329   3.730 0.000253 ***
## PuttsPerRound    -1.2094847  0.2672761  -4.525 1.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6725 on 190 degrees of freedom
## Multiple R-squared:  0.5414, Adjusted R-squared:  0.5293
## F-statistic: 44.86 on 5 and 190 DF,  p-value: < 2.2e-16
```

By looking at the summary of the model, we will remove variable "DrivingAccuracy" as it has the highest P-Value.

```
model_m2 <- lm(ln_Prize ~ GIR+PuttingAverage+BirdieConversion+PuttsPerRound, data=data)
summary(model_m2)
```

```
##
## Call:
## lm(formula = ln_Prize ~ GIR + PuttingAverage + BirdieConversion +
##     PuttsPerRound, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55608 -0.51122 -0.08109  0.45250  2.12227
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.02738    6.35383   1.263   0.2080
## GIR               0.26791    0.02536  10.563  < 2e-16 ***
## PuttingAverage    8.81065    5.26991   1.672   0.0962 .
## BirdieConversion  0.15360    0.03561   4.314 2.57e-05 ***
## PuttsPerRound    -1.20702    0.26391  -4.574 8.61e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6707 on 191 degrees of freedom
## Multiple R-squared:  0.5414, Adjusted R-squared:  0.5318
## F-statistic: 56.37 on 4 and 191 DF,  p-value: < 2.2e-16
```

- After refitting the model, we can see adjusted R2 increased from 0.5293 to 0.5318.
- We can still see variable "PuttingAverage" have high P-value. Hence we will again refit the model without variable "PuttingAverage".

```
model_m3 <- lm(ln_Prize ~ GIR+BirdieConversion+PuttsPerRound, data=data)
summary(model_m3)
```

```
##
## Call:
## lm(formula = ln_Prize ~ GIR + BirdieConversion + PuttsPerRound,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6140 -0.5152 -0.0761  0.4540  2.0583
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       15.8102     4.3446   3.639 0.000352 ***
## GIR                0.2454     0.0216  11.360  < 2e-16 ***
## BirdieConversion   0.1145     0.0270   4.243 3.43e-05 ***
```

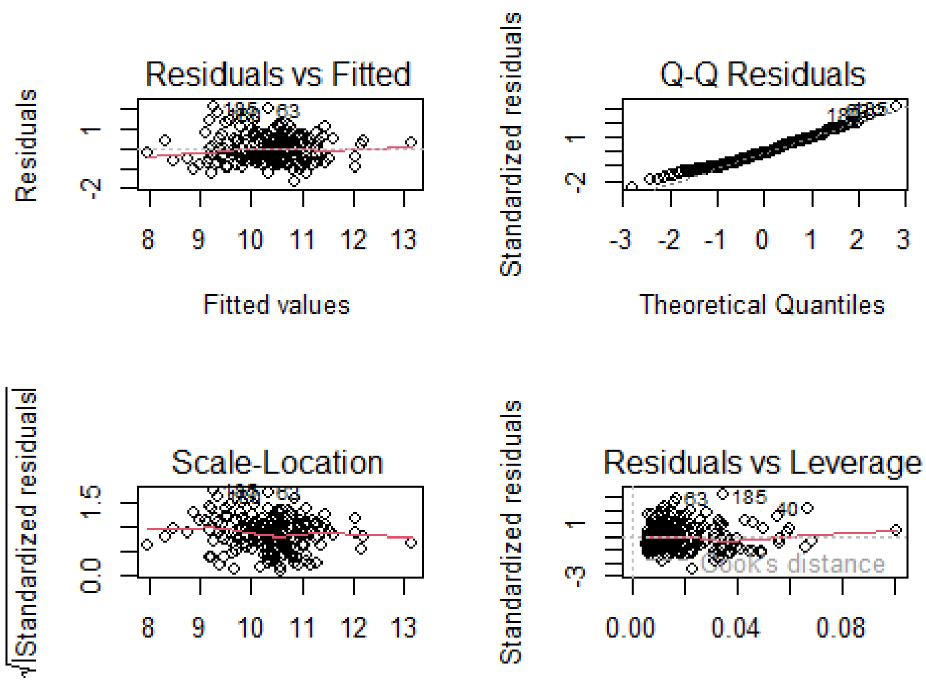
```
## PuttsPerRound     -0.8476     0.1538  -5.512 1.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6738 on 192 degrees of freedom
## Multiple R-squared:  0.5347, Adjusted R-squared:  0.5274
## F-statistic: 73.54 on 3 and 192 DF,  p-value: < 2.2e-16
```

- **After refitting the model, now we can see all variables looking significant.**
- **Adjusted R-squared: 0.5274**

b)

Residual plots

```
par(mfrow=c(2,2))
plot(model_m3)
```



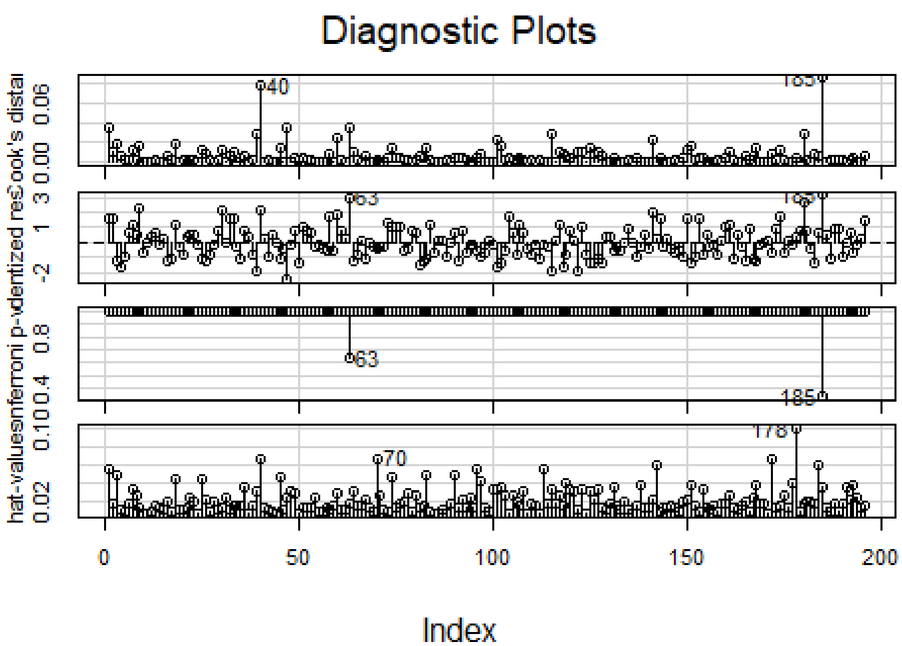
- By looking at the Residuals vs Fitted plot, we can say the model looks valid as variation is somewhat less and there are less number of outliers.

- With the help of Q-Q plot, we can see there are many points which follow the line.

c)

Influential points

```
influenceIndexPlot(model_m3)
```



Finding outliers

```
residuals_standardized_1 <- rstandard(model_m3)
outliers_indices_1 <- which(residuals_standardized_1 > 3)
```

Extract values with standardized residuals greater than 3

```
outliers_values_1 <- residuals_standardized_1[outliers_indices_1]
```

Showing the indices and values of outliers

```
cat("Indices of outliers:", outliers_indices_1, "\n")

## Indices of outliers: 185

cat("Values of outliers:", outliers_values_1, "\n")

## Values of outliers: 3.108311
```

- We have chosen these points as outliers as the value for standardized residuals is greater than 3. i.e these points are 3 standard deviations away from the mean. Hence we can call those points as outliers.

Question 5 :

```
coefficients <- coef(model_m3)
coefficients
```

```
##     (Intercept)            GIR BirdieConversion    PuttsPerRound
##      15.8101628      0.2454205        0.1145444       -0.8475661
```

- For each 1% increase in the GIR, we expect an average increase of exp(0.2454205) times in PrizeMoney, holding other factors constant.

Question 6 :

```
test_data <- data.frame(DrivingAccuracy = 64, GIR = 67, BirdieConversion = 28, PuttingAverage = 1.77, PuttsPerRound = 29.16)
```

Predictions and 95% prediction interval

```
prediction <- predict(model_m3, newdata = test_data, interval = "prediction", level = 0.95)
```

Display the results

```
print(prediction)
```

```
##        fit      lwr      upr
## 1 10.74555 9.407982 12.08312
```

Back-transform predictions and interval to original scale

```
prediction_original <- exp(prediction[, 1])
lower_bound_original <- exp(prediction[, 2])
upper_bound_original <- exp(prediction[, 3])
```

Display the back-transformed results

```
print(data.frame(Predicted_PrizeMoney = prediction_original, Lower_Bound = lower_bound_original, Upper_Bound = upper_bound_original))
```

```
##   Predicted_PrizeMoney Lower_Bound Upper_Bound
## 1             46422.93    12185.26    176860.2
```