ITEC 621 - Homework 1 - R and Stats Refresher

Kogod School of Business

O'Malley, Conie

19 Jan 2025

Table of Contents

Q1 - Functions and Loops (12.5 pts.)

Write a simple R function named `area()` that takes 2 values as input parameters (representing the two sides of a rectangle) and returns the product of the two values (representing the rectangle's area).

Then use the functions `print()` and `paste()` to output this result: **"The area of a rectangle of sides 6x4 is 24"**, where 24 is calculated with the `area()` function you just created (not a fixed number). That is, the `area()` function you created, **must appear** inside the `print(paste())` function.

```
area <- function(x, y){ # create a function with 2 input parameters
  z = x*y # calculate area
  print(paste("The area of a rectangle of sides", x, "x", y, "is", z)) #
print phrase with calculations
}

area(6,4) # function test

## [1] "The area of a rectangle of sides 6 x 4 is 24"
```

Write a simple **for loop** for i from **1 to 10** to compute **squared** values for the numbers 1 through 10. In each loop pass, compute the square of i (i.e., `i^2`) and in each case, display **exactly** "The square of 1 is 1" for i=1, "The square of 2 is 4" for i=2, and so on. You must use the functions `pring(paste())` and the formula `i^2` to display your results.

```
for (i in 1:10){ # for loop range
  a = i**2 # calculate squares
  print(paste("The square of", i, "is", a)) # print phrase with calculations
}
```

```
## [1] "The square of 1 is 1"
## [1] "The square of 2 is 4"
## [1] "The square of 3 is 9"
## [1] "The square of 4 is 16"
## [1] "The square of 5 is 25"
## [1] "The square of 6 is 36"
## [1] "The square of 7 is 49"
## [1] "The square of 8 is 64"
## [1] "The square of 9 is 81"
## [1] "The square of 10 is 100"
```

Q2 - Data Frames and Vectors (12.5 pts.)

We suspect that income and loan balance may predict the credit rating of a bank's customer. Let's explore. Copy the Credit.csv data file to your working directory, then:

Read the Credit.csv data table into a data frame named "Credit" (tip: use the `read.table()` function with the appropriate header= and sep= attributes)

Display (only) the **first 10 rows** of (only) the Rating, Income and Balance columns. Use the `cbind()` function to bind the 3 columns and ensure that your columns have labels (e.g. "Rating"=Credit$Rating[1:10], etc.).

```
read.table("Credit.csv", header = TRUE, sep = ",") -> Credit # read data into
table

cbind("Rating" = Credit$Rating[1:10], # display required columns
      "Income" = Credit$Income[1:10],
      "Balance" = Credit$Balance[1:10])
```

```
##       Rating  Income Balance
##  [1,]    283  14.891     333
##  [2,]    483 106.025     903
##  [3,]    514 104.593     580
##  [4,]    681 148.924     964
##  [5,]    357  55.882     331
##  [6,]    569  80.180    1151
##  [7,]    259  20.996     203
##  [8,]    512  71.408     872
```

```
##  [9,]    266  15.125     279
## [10,]    491  71.061    1350
```

Then, display the object class for the Credit data frame and for the vectors Gender (i.e., `Credit$Gender`), Income and Cards

```
class(Credit) # obtain class of Credit

## [1] "data.frame"

class(Credit$Gender) # obtain class of Gender

## [1] "character"

class(Credit$Income) # obtain class of Income

## [1] "numeric"

class(Credit$Cards) # obtain class of Cards

## [1] "integer"
```

Finally, create a vector named **Rating.vect** with data from the Rating column and display the first 6 values of this vector.

```
Rating.vect <- Credit$Rating # assign vector
Rating.vect[1:6] # display first 6 values

## [1] 283 483 514 681 357 569
```

Q3. Descriptive Analytics (12.5 pts.)

Let's analyze the data quantitatively. Compute the **mean, minimum, maximum, standard deviation** and **variance** for all the values in this Rating.vect vector. Store the results in variables named **rtg.mean, rtg.min, rtg.min, rtg.stdev** and **rtg.var** respectively. Then use the `c()` function to concatenate the 5 results into a vector, but name each of the values accordingly (e.g., "Mean Rtg"=rtg.mean). Also, enclose the `c()` function inside the `print()` function and use `digits=5` to limit the significant digits to display (to 2 decimals) (e.g., `print(c(...), digits=5)`)

```
rtg.mean <- mean(Rating.vect)
rtg.min <- min(Rating.vect)
rtg.max <- max(Rating.vect)
rtg.stdev <- sd(Rating.vect)
rtg.var <- var(Rating.vect)
print(c("Mean Rtg" = rtg.mean, "Min Rtg" = rtg.min, "Max Rtg" = rtg.max,
"StDev Rtg" = rtg.stdev, "Var Rtg" = rtg.var), digits = 5)

##  Mean Rtg   Min Rtg   Max Rtg StDev Rtg   Var Rtg
##    354.94     93.00    982.00    154.72  23939.56
```

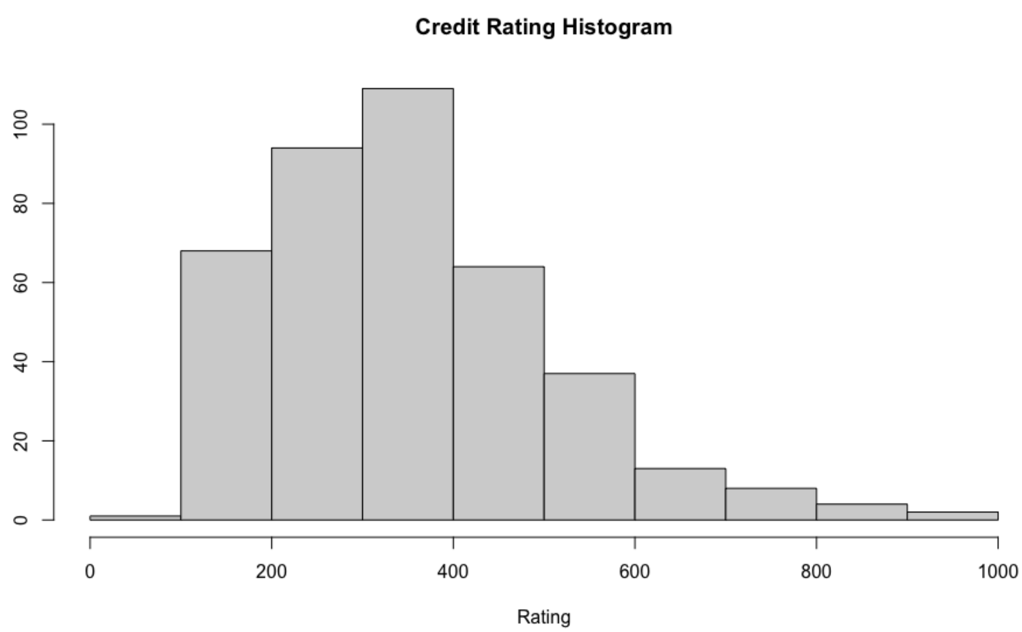
Divide the plot output into 1 row and 2 columns to display 2 graphs side by side using `par(mfrow=c(1,2))`. Display a histogram for **Credit Rating**, with the main title **Credit Rating Histogram** and X label **Rating**. Since the histogram is a bit skewed to the right, display a histogram for the **Log of Credit Rating**, with the main title **Log of Credit Rating Histogram** and X label **Log Rating**. Tip: use the `log()` function (lower case) to log variables. Then, reset the graph layout to 1 by 1 using par(mfrow=c(1,1)).

Technical Note: Notice that we divide the plot window to display 2 graphs side by side and then reset to the normal window to a single plot, and we then do this repeatedly for every exercise below. This is only necessary so that when you press the play icon for the chunk, you can see the 2 graphs side by side. If you don't want to do this repeatedly, you could set the window for the first pair of plots and reset it after the last pair of plots. The play icon will not give you side by side graphs, but the knitted document should show the graphs side by side. Can you figure out why? (no need to answer, just think)

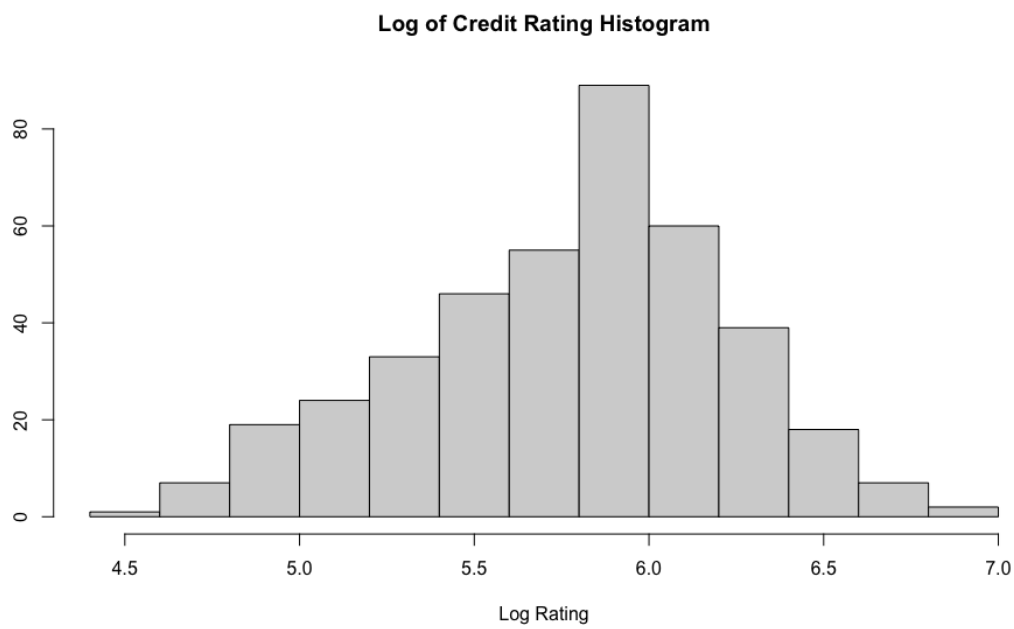
Technical note: also note in the HW template that I have sized the graphs to `fig.width=10`, `fig.height=6`. You can change this if you prefer a different graph size.

```
#par(mfrow=c(1,2)) # set parameters
par(mfrow=c(1,1)) # reset parameters

hist(Credit$Rating, main = "Credit Rating Histogram", xlab = "Rating", ylab =
NULL) # initial histogram
```




```
hist(log(Credit$Rating), main = "Log of Credit Rating Histogram", xlab = "Log
Rating", ylab = NULL) # log histogram to address skewness
```

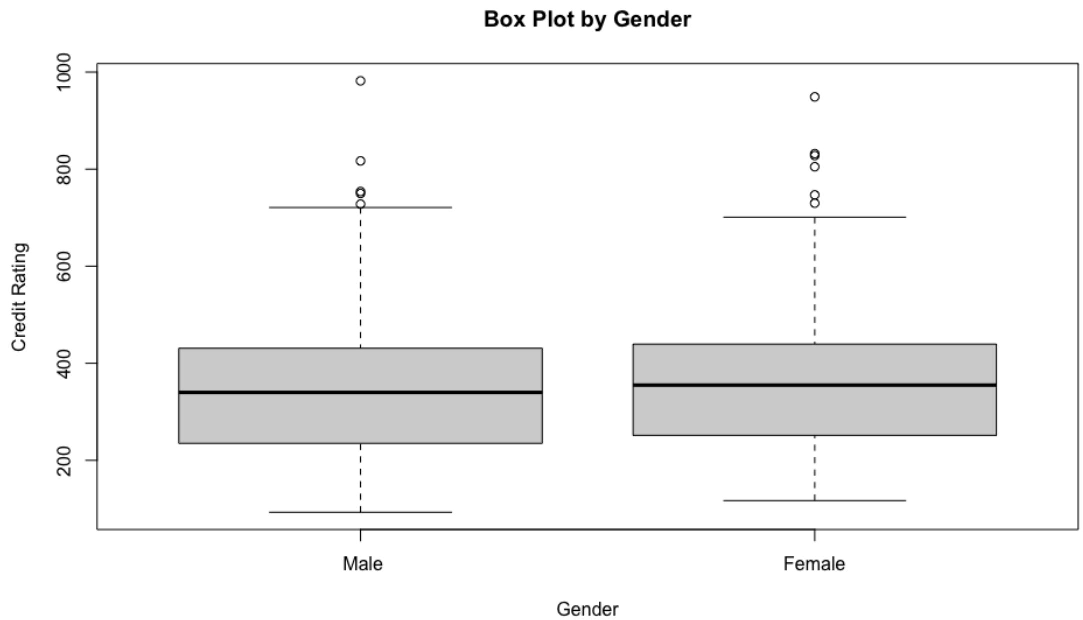


Q4. Visual Analytics (12.5 pts.)

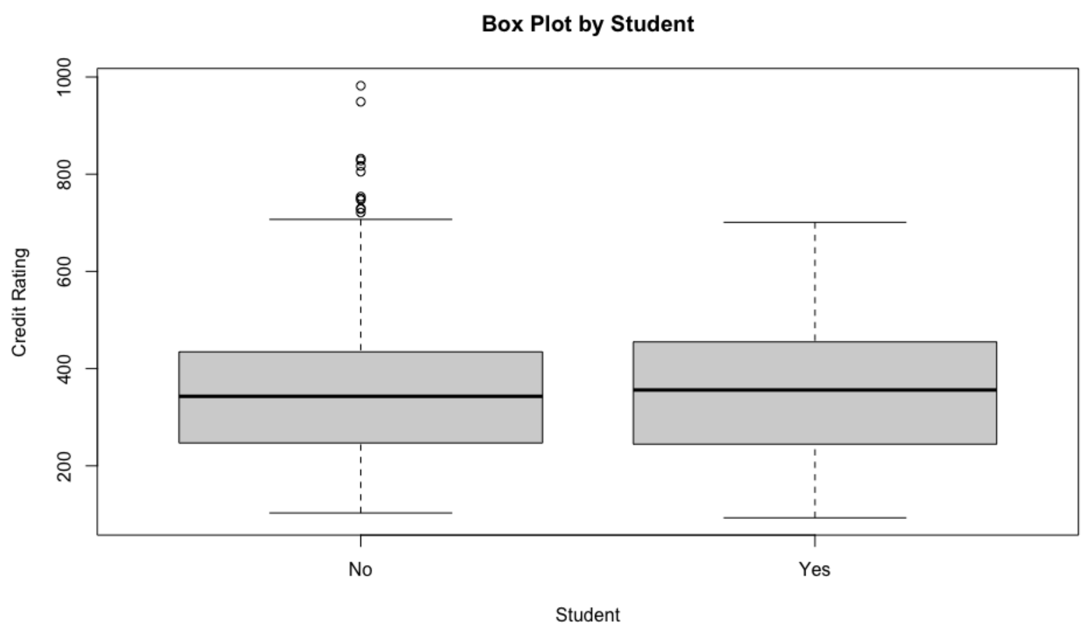
Let's do some visual inspection of the data. Divide the plot output into 1 row and 2 columns to display 2 graphs side by side using `par(mfrow=c(1,2))`. Then display a box plot for **Rating** by **Gender**, and then another box plot for **Balance** by **Student**. Then, reset the graph layout to 1 by 1 using par(mfrow=c(1,1)).

```
#par(mfrow=c(1,2)) # set parameters
par(mfrow=c(1,1)) # reset parameters

boxplot(Credit$Rating~Credit$Gender, main = "Box Plot by Gender", xlab =
"Gender", ylab = "Credit Rating") # gender boxplot
```

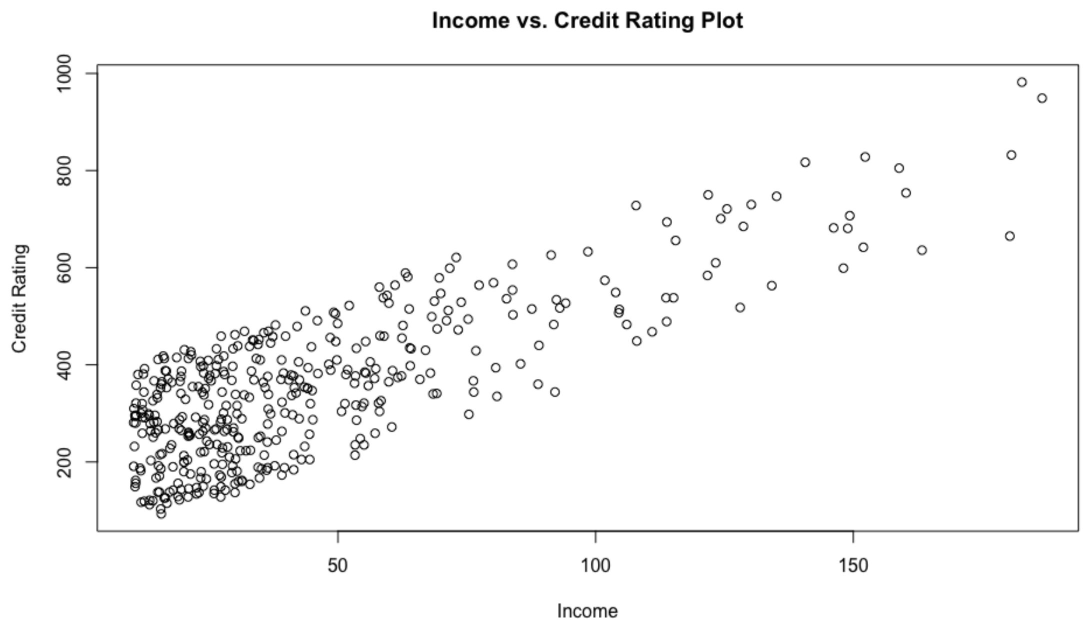



```r
boxplot(Credit$Rating~Credit$Student, main = "Box Plot by Student", xlab =
"Student", ylab = "Credit Rating") # student boxplot
```

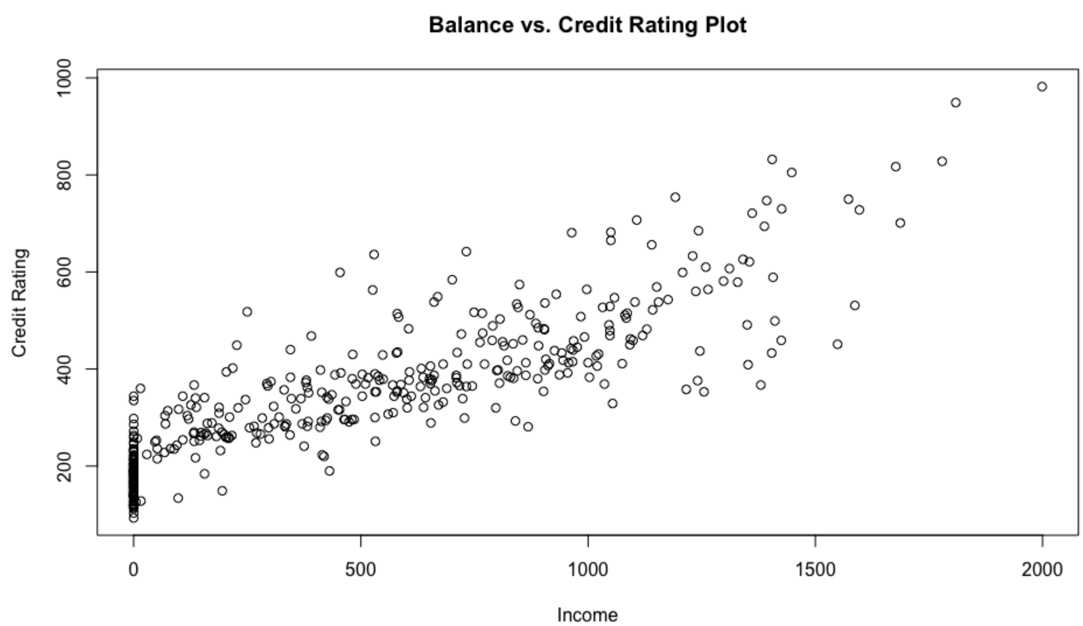


Divide again the plot output into 1 row and 2 columns to display 2 graphs side by side using `par(mfrow=c(1,2))`. Then plot Credit **Rating** (Y axis) against **Income** (X axis), with respective labels **Income** and **Credit Rating**. Then another plot for Credit **Rating** (Y axis) against **Balance** (X axis), with respective labels **Credit Rating** and **Balance**. Then, reset the graph layout to 1 by 1 using par(mfrow=c(1,1)).

```
#par(mfrow=c(1,2)) # set parameters
par(mfrow=c(1,1)) # reset parameters
plot(Credit$Income, Credit$Rating, main = "Income vs. Credit Rating Plot",
xlab = "Income", ylab = "Credit Rating") # income vs. credit rating plot
```



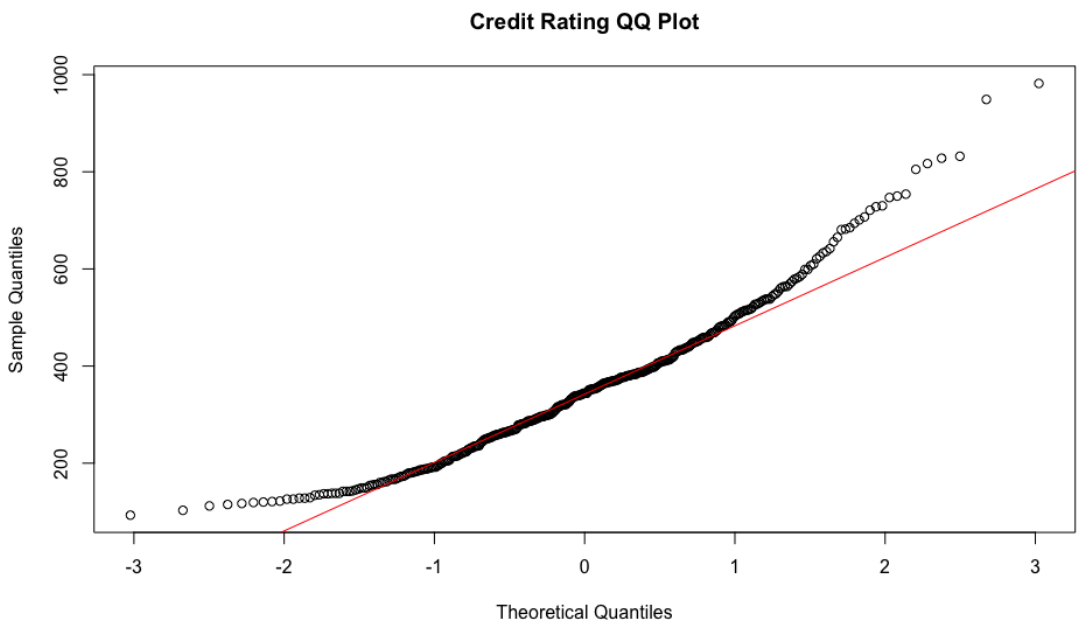

```
plot(Credit$Balance, Credit$Rating, main = "Balance vs. Credit Rating Plot",
xlab = "Income", ylab = "Credit Rating") # balance vs. credit rating plot
```



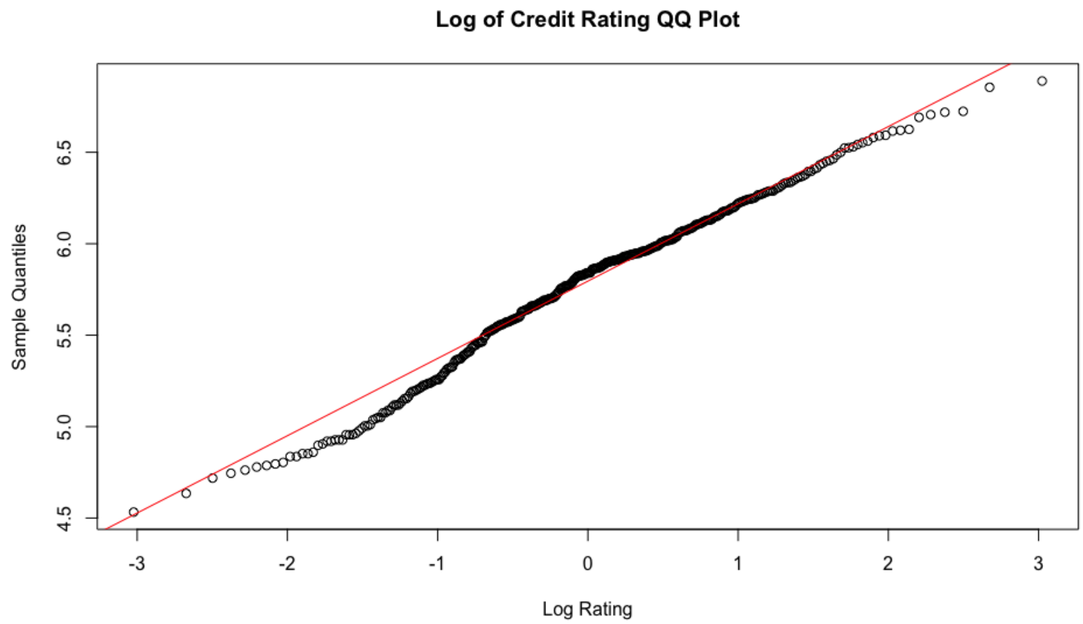
Again, divide the plot output into 1 row and 2 columns to display 2 graphs side by side using `par(mfrow=c(1,2))`. Display a qqplot (using the `qqnorm()` function and `qqline()`) for the credit rating variable, with the main title **Credit Rating QQ PLot** (use the `main=` attribute). Since the QQ Plot shows some deviation from the normal distribution line, display a QQ Plot for the **Log of Credit Rating**, with the main title **Log of Credit Rating QQ PLot** and X label **Log Rating**. Then, reset the graph layout to 1 by 1 using `par(mfrow=c(1,1))`. It looks more normally distributed, right?

```
#par(mfrow=c(1,2)) # set parameters
par(mfrow=c(1,1)) # reset parameters

qqnorm(Credit$Rating, main = "Credit Rating QQ Plot", xlab = "Theoretical
Quantiles", ylab = "Sample Quantiles")
qqline(Credit$Rating, col = "red")
```




```
qqnorm(log(Credit$Rating), main = "Log of Credit Rating QQ Plot", xlab = "Log
Rating", ylab = "Sample Quantiles")
qqline(log(Credit$Rating), col = "red")
```



Q5. {ggplot} Graphs (12.5 pts.)

{ggplot2} is one of the best visual analytic packages out there. Let's explore it briefly. Load the **{ggplot2}** library.

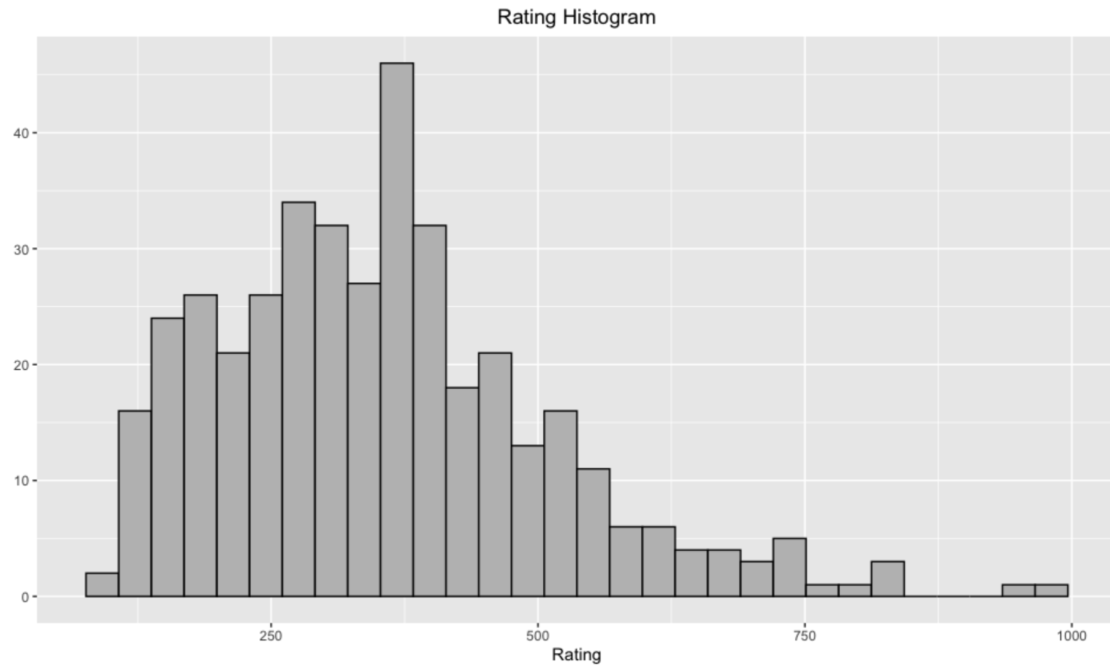
Note: `ggplot()` renders one plot at a time, so there is no need to split the plot window into more than one row or column.

Then, use the `ggplot()` function to draw a histogram of the **Rating** variable and then draw another histogram for **log(Rating)**. Tip: you need to use `ggplot(Credit)` first to point to the data set you want to use. You then need to add the necessary graph attributes with the **+** operator. In this case you need to use the `geom_histogram(aes(etc.))` graph attribute and aesthetic.

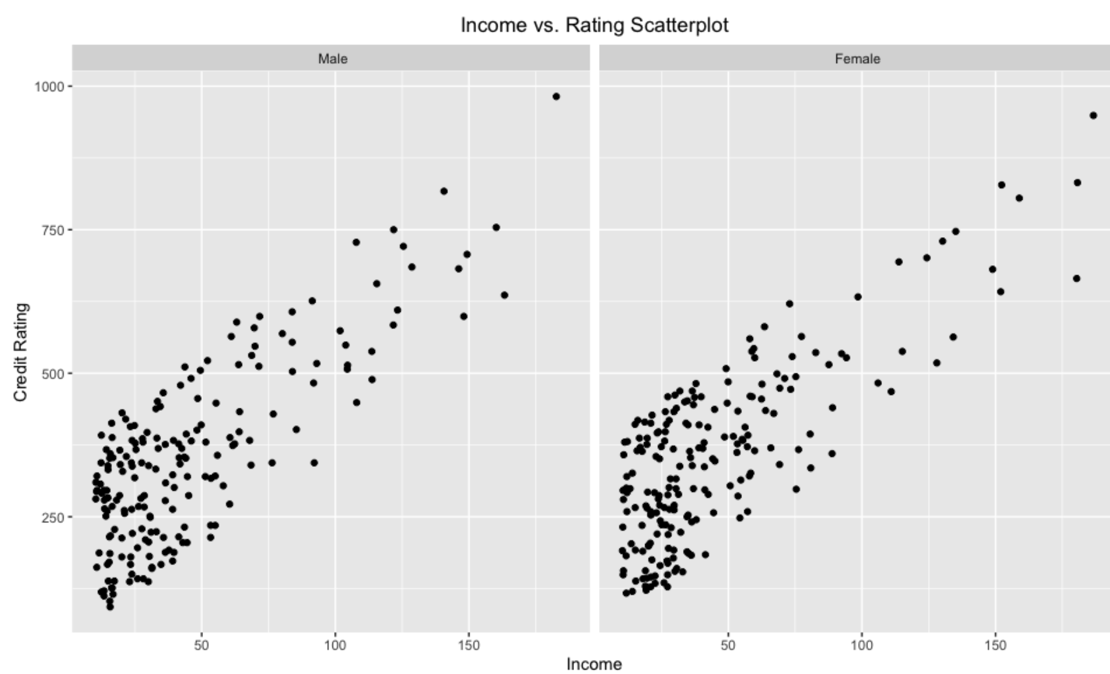
Then, use the `ggplot()` function again to draw a couple of scatter plots. In the first one, plot **Income** (x axis) against **Rating** (y axis), but separated by **Gender**. Tip, use `ggplot(Credit, aes(x=Income, y=Rating))` to use the 2 variable plot aesthetic with the **Credit** data set. Then add `geom_point()` to draw a scatter plot, and then add `facet_wrap(~Gender)` to draw two **facets by Gender**.

Then draw a similar scatter plot, but this time use **Balance** (x axis) by **Rating** (y axis), faceted by **Student**.

```
ggplot(Credit, aes(Rating)) +
  geom_histogram(color = "black", fill = "grey") + # adjust histogram
aesthetics
  labs(title = "Rating Histogram", y = NULL) + # label updates
  theme(plot.title = element_text(hjust = 0.5)) # label adjustment
```

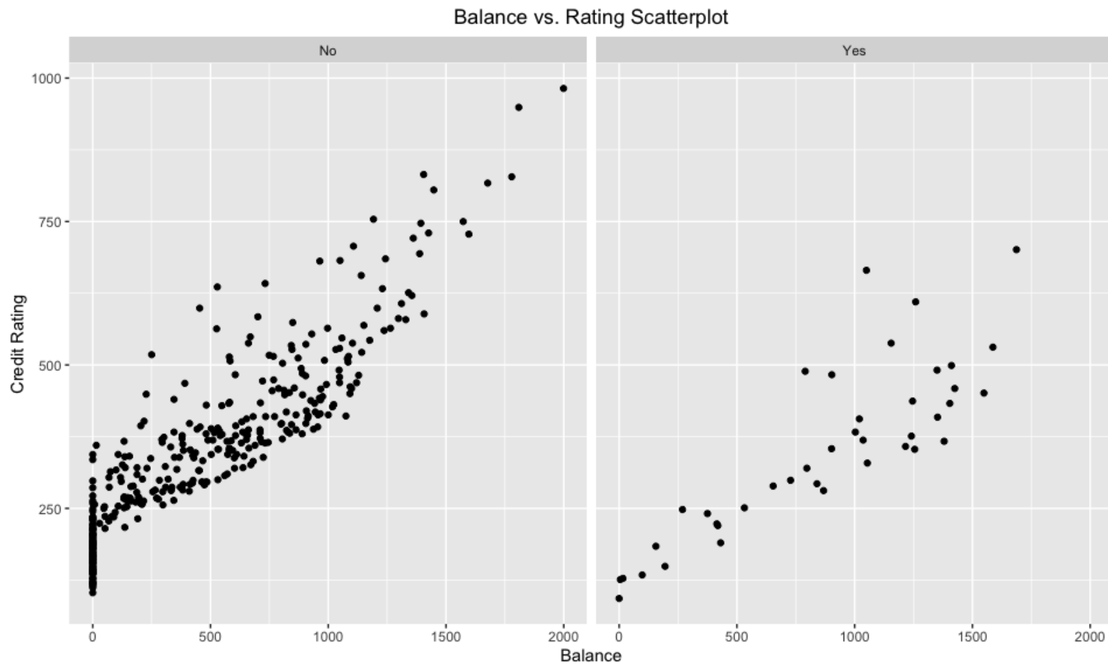



```
ggplot(Credit, aes(Income, Rating)) +
  geom_point() + # scatterplot
  labs(title = "Income vs. Rating Scatterplot", x = "Income", y = "Credit
Rating") + # label updates
  facet_wrap(~Gender) + # gender facet wrap
  theme(plot.title = element_text(hjust = 0.5)) # label adjustment
```




```
ggplot(Credit, aes(Balance, Rating)) +
  geom_point() + # scatterplot
  labs(title = "Balance vs. Rating Scatterplot", x = "Balance", y = "Credit
```

```
Rating") + # label updates
  facet_wrap(~Student) + # student facet wrap
  theme(plot.title = element_text(hjust = 0.5)) # label adjustment
```



Q6. Linear Regression Models (12.5 pts.)

Fit a **small** linear regression model object with the `lm()` function to predict credit **Rating** using **Income** and **Balance** as predictors. Name the resulting linear model **fit.small**. Then display the `summary()` results.

```
fit.small <- lm(Rating ~ Income + Balance, Credit) # build a lm model
summary(fit.small) # compute summary statistics

##
## Call:
## lm(formula = Rating ~ Income + Balance, data = Credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.485   -9.665   14.543   24.576   53.931
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.454e+02  3.285e+00   44.25   <2e-16 ***
## Income      2.186e+00  6.063e-02   36.06   <2e-16 ***
## Balance     2.129e-01  4.648e-03   45.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.82 on 397 degrees of freedom
```

```
## Multiple R-squared:  0.9405, Adjusted R-squared:  0.9402
## F-statistic:  3140 on 2 and 397 DF,  p-value: < 2.2e-16
```

Now fit a larger linear regression model, same as **fit.small**, but add 2 more predictors this time, **Age** and **Gender**. Name the resulting linear model **fit.large**. Then display the `summary()` results.

```r
fit.large <- lm(Rating ~ Income + Balance + Age + Gender, Credit) # build a lm model
summary(fit.large) # compute summary statistics
```

```
##
## Call:
## lm(formula = Rating ~ Income + Balance + Age + Gender, data = Credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104.760   -9.683   14.686   24.959   50.408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.380e+02  7.022e+00  19.660   <2e-16 ***
## Income       2.171e+00  6.190e-02  35.078   <2e-16 ***
## Balance      2.134e-01  4.673e-03  45.680   <2e-16 ***
## Age          1.370e-01  1.120e-01   1.223    0.222
## GenderFemale 1.591e-01  3.789e+00   0.042    0.967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.84 on 395 degrees of freedom
## Multiple R-squared:  0.9408, Adjusted R-squared:  0.9402
## F-statistic:  1569 on 4 and 395 DF,  p-value: < 2.2e-16
```

Q7. ANOVA (12.5 pts.)

Do an `anova()` test to evaluate if **fit.large** has significantly more predictive power than **fit.small**.

Then provide a brief **interpretation** of your results for Q5, Q6 and Q7. Which predictors are significant? which are not? and why? which of the two models is preferred? why?

```r
anova(fit.small) # anova of small model
```

```
## Analysis of Variance Table
##
## Response: Rating
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Income      1 5982140 5982140  4182.1 < 2.2e-16 ***
## Balance     1 3001866 3001866  2098.6 < 2.2e-16 ***
## Residuals 397  567878    1430
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit.large) # anova of large model
```

```
## Analysis of Variance Table
##
## Response: Rating
##            Df  Sum Sq Mean Sq   F value Pr(>F)
## Income      1 5982140 5982140 4176.7861 <2e-16 ***
## Balance     1 3001866 3001866 2095.9310 <2e-16 ***
## Age         1    2142    2142    1.4958 0.2220
## Gender      1       3       3    0.0018 0.9665
## Residuals 395  565733    1432
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# IMPORTANT: I'm writing my commentary inside the R code chunk with a # text
entry, only because I want to hide it for the homework. Once I turn the echo
on with echo=T to show the code you will see my answer. However, you should
NEVER write your interpretation reports inside the R code chunk. It looks
unprofessional and some times {knitr} does not knit it well. Write all your
answers outside of the code chunk.
```

ANOVA Analysis

While the SSE is slightly less in the **fit.large** model than in the **fit.small** model (565733 < 567878, respectively), neither of the additional variables that we added to the **fit.large** model are statistically significant (p (0.222) > 0.05 and p (0.966) > 0.05) to the model. To adhere to the principle of model simplicity, the minimal change in SSE does not offset the two additional variables and complication of the model, therefore the **fit.small** model should be used.

Q5 Analysis

There **Rating** variable is right skewed looking at the histogram and is confirmed by the clustering of data points in the lower ends of the scatterplots. The scatterplots also indicate a possible linear relationship between **Gender** and **Student** with **Rating**, but the skewness could result in a nonlinear relationship with further investigation. I recommend plotting the residuals to assess skewness and homoscedasticity.

Q6 Analysis

Comparing **fit.small** and **fit.large** models through summary statistics, we see that all variables are significant to the **fit.small** model (p (2.2e-16) < 0.05 for both **Income** and **Balance**), while adding **Gender** and **Student** to the **fit.large** model does not increase the Adjusted R-squared value (r^2 = 0.9402 in both models). The variables **Gender** and **Student** are not statistically significant the **fit.large** model either (p (0.222) > 0.05 and p (0.966) > 0.05, respectively).

Q7 Analysis

As stated above in *ANOVA Analysis*, the SSE of the **fit.large** model does not significantly decrease compared to the **fit.small** model and the variables **Gender** and **Student** are confirmed to be statistically insignificant to the model.

Conclusion

fit.small is the better model because the **fit.large** model does not present significant improvement by any measure and the complication of the model by adding new variables is not justified.

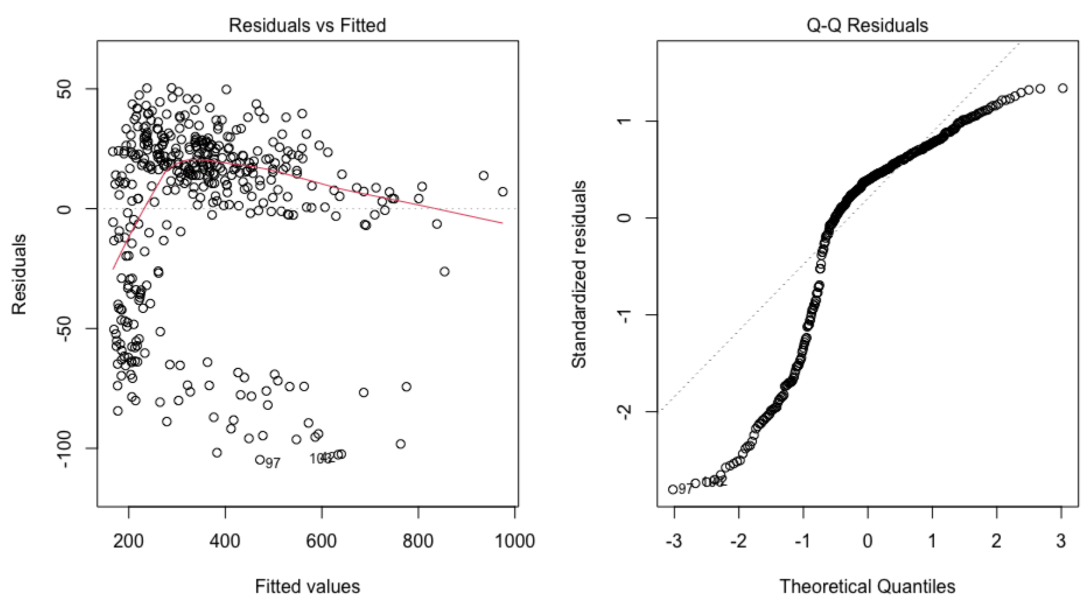
Q8. Regression Plots (12.5 pts.)

As you should know, the linear model object contains 4 plots. But in this exercise we will only plot the first 2. So, divide the plot window into 1 rows and 2 columns to display the 2 plots side by side. Then plot the **fit.large** linear model object, but use the attribute `which=1`. Then plot it again, but using `which=2` instead. Then reset the plot window to a single graph as you did earlier.

Then provide a brief interpretation of what you see in the two graphs and what issues the may entail.

```
par(mfrow=c(1,2)) # set parameters

plot(fit.large, which = 1) # plot 1
plot(fit.large, which = 2) # plot 2
```




```
#par(mfrow=c(1,1)) # reset parameters
```

Analysis

Plot 1 (Residual vs. Fitted Plot) shows a clear nonlinear relationship in the distribution of residuals. **Plot 2** (Residual QQ Plot) also suggests the residuals are not normally distributed. The model should be first computed using a log transformation function to see if there is a deeper nonlinear relationship that requires a multivariate quadratic model to assess. The model should also be computed using a Weighted Least Squares model to address heteroscedasticity or the fact that certain variables may not meet the independence assumption of linear models.