KSB-999 HW0 - R Overview for Business Analytics

J. Alberto Espinosa

January 12, 2020

Table of Contents

Overview

The goal of this homework **HW0** is to get you prepared to succeed in **ITEC 620 Business Insights from Analytics** and **ITEC 621 Predictive Analytics**. R will be used in those courses as the main tools for analytics. Parts of This homework are somewhat similar to HW1 in ITEC 621, so once you complete this HW0, you will be in a much better position to complete ITEC 621's HW1 and succeed in that course. The hardest thing in ITEC 621 is to learn the basics of predictive modeling and R at the same time. If you are somewhat fluent in the R syntax, you will be able to focus on nuances of predictive modeling.

R Markdown

R Markdown is a package that allows you to write R code and prepare an analytics report in a single file. To use R Markdown, you first need to install it in your computer using the command `install.packages("rmarkdown")`. If you have not done this yet, go to the **R Console** and install R Markdown. Once you have done this, you can create R Markdown files from the File -> New File menu.

When you create an R Markdown file, it will look like text comingled with R code. You will see a button option named **Knit** in your tool bar. Once you are done with all the coding, click on the **Knit** button and R Markdown will knit a Word, HTML, PDF or PowerPoint document for you, depending on the output type you specified, with all your typed text and R results.

For this practice homework, I have provided a blank R Markdown file named **HW0_YourLastName.Rmd**. Download this file and rename it with your last name, and then do all your coding there. If you fell more comfortable with a plain R Scrip, just complete this homework in your **HW0_YourLastName.R** script. But I encourage you to try to copy your R code to the appropriate sections of the R Markdown file so that you get used to it, because you will be using this all the time in ITEC 621.

R Markdown contains three main types of content:

1. The **YAML** (YAML Ain't Markup Language) header, which is where you place the title, author, date, type of output, etc. It is at the top of the R Markdown file and starts and ends with `---`. I suggest using an output type `word_document`. HTML works well, but blackboard will not read HTML files submitted by students (for security reasons).

2. **Markup** sections, which is where you type any text you wish, which will show up as typed text. You will learn these later.

3. **Code chunks**: which is where you write your R code. An R code chunk starts with a ```` ```**{r}** ```` and ends with a ```` ``` ````.

I recommend that you first create an R Script called **HW0_YourLastName.R** to try your R code. Once you are satisfied that the R code is working fine, then copy/paste the respective code segments to an R Markdown file named **HW0_YourLastName.Rmd**. I recommend using the template I provided on Blackboard.

Your knitted file must:

- Display all your R commands (leave `echo=T` in the global options; `echo=F` suppresses the R code)
- Display the resulting **R output results**
- Contain anay necessary text and explanations, as needed; and
- Be formatted for good readability and in a business like manner
- Be in the same order as the questions and with the corresponding question numbers

Submission:

Knit a Word document with your R Markdown file and knitted to your **.Rmd** file. You don't need to submit this on Blackboard for KSB-999. But keep your results to be discussed in ITEC 621, where you will complete a related homework and get the solution.

Questions

1. Write a simple R function named **Area()** that takes 2 values as 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

```
Area <- function(x,y) {return(x*y)}
print(paste("The area of a 4x6 rectanlge is", Area(4,6)))

## [1] "The area of a 4x6 rectanlge is 24"
```

2. Write a simple **for loop** for i from 1 to 10. In each loop pass, compute the area of a rectangle of sides i and i*2 (i.e., all rectangles have one side double the lenght than the other) and for each of the 10 rectangles display "The area of an 1 x 2 rectangle is 2" for i=1, "The area of an 2 x 4 rectangle is 8", and so on.

```
for (i in 1:10) {
  print(paste("The area of a", i, "x", i*2,
              "rectangle is", Area(i, 2*i)))
  }
```

```
## [1] "The area of a 1 x 2 rectangle is 2"
## [1] "The area of a 2 x 4 rectangle is 8"
## [1] "The area of a 3 x 6 rectangle is 18"
## [1] "The area of a 4 x 8 rectangle is 32"
## [1] "The area of a 5 x 10 rectangle is 50"
## [1] "The area of a 6 x 12 rectangle is 72"
## [1] "The area of a 7 x 14 rectangle is 98"
## [1] "The area of a 8 x 16 rectangle is 128"
## [1] "The area of a 9 x 18 rectangle is 162"
## [1] "The area of a 10 x 20 rectangle is 200"
```

3. Copy the Credit.csv data file to your working directory:

- Then read the Credit.csv data table into an object named "Credit"
- List the top 6 rows of the table
- List the first 5 columns of the top 5 rows

```
Credit <- read.table("Credit.csv", header=TRUE, sep=",")
head(Credit)
```

```
##   X  Income Limit Rating Cards Age Education Gender Student Married
## 1 1  14.891  3606    283     2  34        11   Male      No     Yes
## 2 2 106.025  6645    483     3  82        15 Female     Yes     Yes
## 3 3 104.593  7075    514     4  71        11   Male      No      No
## 4 4 148.924  9504    681     3  36        11 Female      No      No
## 5 5  55.882  4897    357     2  68        16   Male      No     Yes
## 6 6  80.180  8047    569     4  77        10   Male      No      No
##   Ethnicity Balance
## 1 Caucasian     333
## 2     Asian     903
## 3     Asian     580
## 4     Asian     964
## 5 Caucasian     331
## 6 Caucasian    1151
```

```
head(Credit)[1:5,1:5]
```

```
##   X  Income Limit Rating Cards
## 1 1  14.891  3606    283     2
## 2 2 106.025  6645    483     3
## 3 3 104.593  7075    514     4
## 4 4 148.924  9504    681     3
## 5 5  55.882  4897    357     2
```

4. Create a simple linear model object with the lm() function to fit credit rating as a function of all remaining variables as predictors and:

- Save the results in an object called "fit.rating";

- Display the model summary results (summary() function)

```
fit.rating <- lm(Rating~., data=Credit)
summary(fit.rating)
```

```
##
## Call:
## lm(formula = Rating ~ ., data = Credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.5242  -7.4159  -0.7162   6.3920  27.7848
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        32.350766   4.247242   7.617 2.01e-13 ***
## X                  -0.001616   0.004469  -0.362  0.71784
## Income              0.124708   0.046958   2.656  0.00824 **
## Limit               0.063169   0.001429  44.198  < 2e-16 ***
## Cards               4.589177   0.392002  11.707  < 2e-16 ***
## Age                 0.013257   0.030448   0.435  0.66351
## Education          -0.234753   0.163978  -1.432  0.15306
## GenderFemale        0.177666   1.027679   0.173  0.86284
## StudentYes         -2.061302   2.812202  -0.733  0.46401
## MarriedYes          2.392026   1.059948   2.257  0.02458 *
## EthnicityAsian     -2.009116   1.450902  -1.385  0.16693
## EthnicityCaucasian -0.278405   1.259476  -0.221  0.82517
## Balance             0.012076   0.005190   2.327  0.02049 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.16 on 387 degrees of freedom
## Multiple R-squared:  0.9958, Adjusted R-squared:  0.9957
## F-statistic:  7682 on 12 and 387 DF,  p-value: < 2.2e-16
```

5. Write a simple linear model to predict credit ratins using the most significant predictors: Income, Limit, Cards, MarriedYes, Balance; and

- Display the regression summary results

```
fit.rating.5 <- lm(Rating~Income+Limit+Cards+Married+Balance, data=Credit)
summary(fit.rating.5)
```

```
##
## Call:
## lm(formula = Rating ~ Income + Limit + Cards + Married + Balance,
##     data = Credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.0051  -7.0024  -0.9291   6.3789  26.2751
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.711e+01  2.187e+00  12.396  < 2e-16 ***
## Income      9.750e-02  3.352e-02   2.909  0.00383 **
## Limit       6.415e-02  9.004e-04  71.247  < 2e-16 ***
## Cards       4.711e+00  3.762e-01  12.521  < 2e-16 ***
## MarriedYes  2.122e+00  1.044e+00   2.032  0.04281 *
## Balance     8.436e-03  3.131e-03   2.694  0.00735 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.14 on 394 degrees of freedom
## Multiple R-squared:  0.9958, Adjusted R-squared:  0.9957
## F-statistic: 1.85e+04 on 5 and 394 DF,  p-value: < 2.2e-16
```

6. Display the object class for Gender (i.e., Credit$Gender), Income and Cards

```
class(Credit$Gender)

## [1] "factor"

class(Credit$Income)

## [1] "numeric"

class(Credit$Cards)

## [1] "integer"
```

7. Briefly answer: what do these classes mean?

```
# Gender is a Factor object and its values can be one of various categories;
Income is a numeric object and can contain decimals; Cards is an integer and
cannot contain decimals.
```

8. Create a vector named “Income.vect” with data from the Income column

- Display the first 6 values of this vector

```
Income.vect <- Credit$Income
head(Income.vect)

## [1]  14.891 106.025 104.593 148.924  55.882  80.180
```

9. Compute and display (separtely) the mean, minimum, maximum, standard deviation and variance for all the values in this income vector

```
mean(Income.vect)

## [1] 45.21889

min(Income.vect)

## [1] 10.354

max(Income.vect)

## [1] 186.634
```

```
sd(Income.vect)

## [1] 35.24427

var(Income.vect)

## [1] 1242.159
```

10. Create a vector called Income.stats with 5 values you computed above and
- Display these Income.stats values

```
Income.stats <- c(mean(Income.vect), min(Income.vect), max(Income.vect),
sd(Income.vect), var(Income.vect))
Income.stats

## [1]   45.21889   10.35400  186.63400   35.24427 1242.15879
```

11. Now give these elements these names: “Mean”, “Min”, “Max”, “StDev”, and “Var” and
- Display these values with their names

```
names(Income.stats) <- c("Mean","Min","Max","StDev", "Var")
Income.stats

##       Mean        Min        Max      StDev        Var
##   45.21889   10.35400  186.63400   35.24427 1242.15879
```

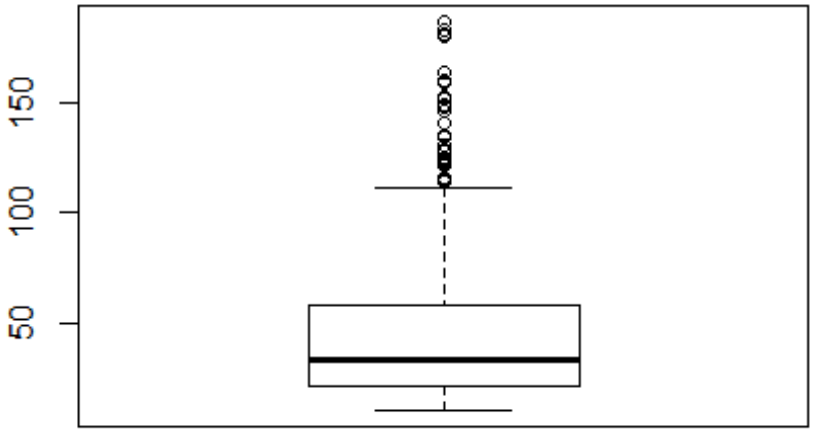
12. Suppose that you want to adjust income for inflation by 5%. Multiply the income values in this vector by 1.05 and
- Display the first 6 records of this computation

```
head(Income.vect*1.05)

## [1]  15.63555 111.32625 109.82265 156.37020  58.67610  84.18900
```

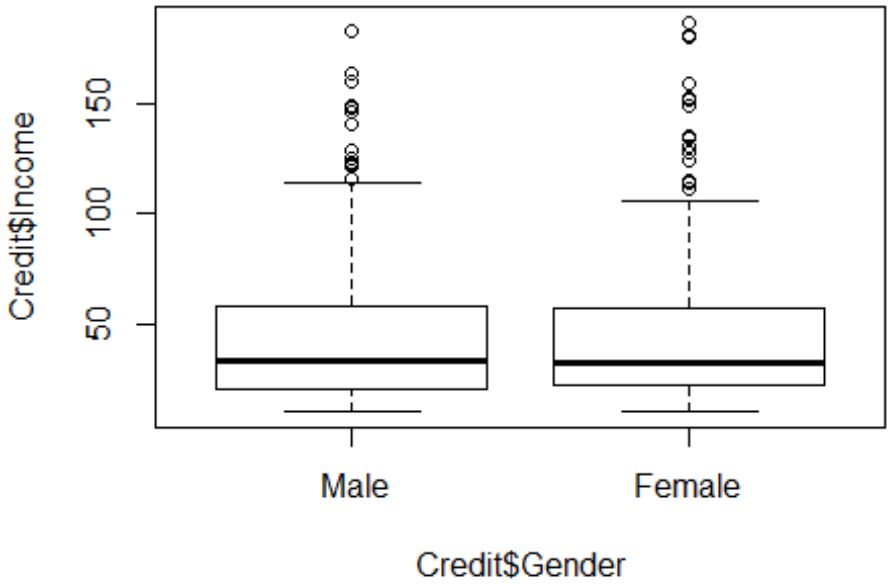
13. Display a boxplot for the predictor “Income”

```
boxplot(Credit$Income)
```



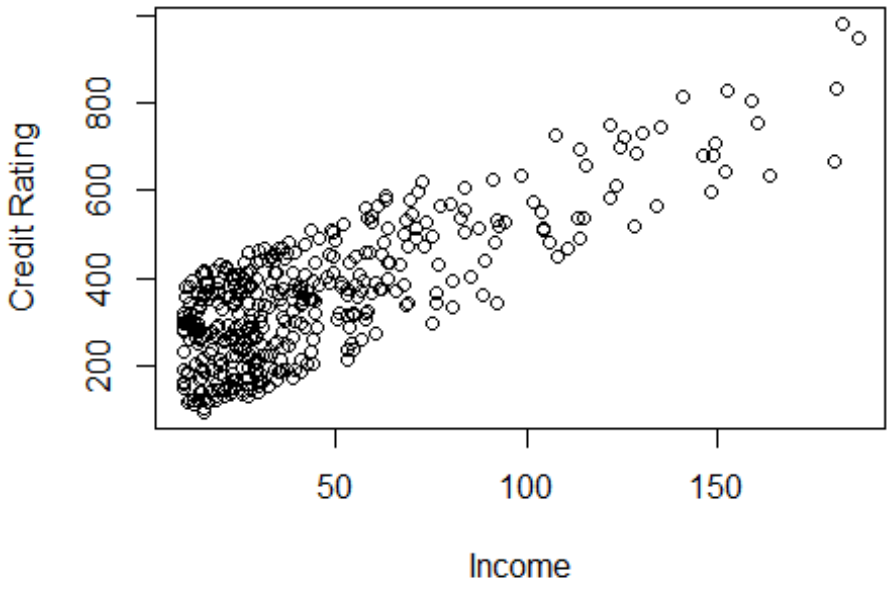
14. Display a boxplot of "Income" by "Gender". No need to respond, but can you tell if there is gender income inequality in this data set?

```
boxplot(Credit$Income~Credit$Gender)
```



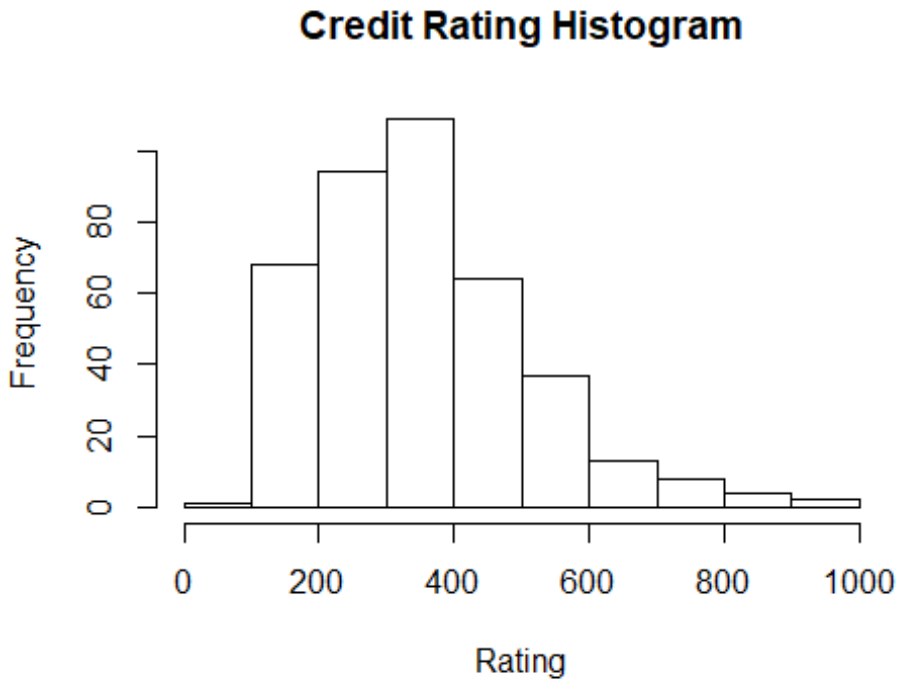
15. Plot Credit Rating (Y axis) against Income (X axis), with respective labels “Income” and "Credit Rating

```
plot(Credit$Income, Credit$Rating, ylab="Credit Rating", xlab="Income")
```



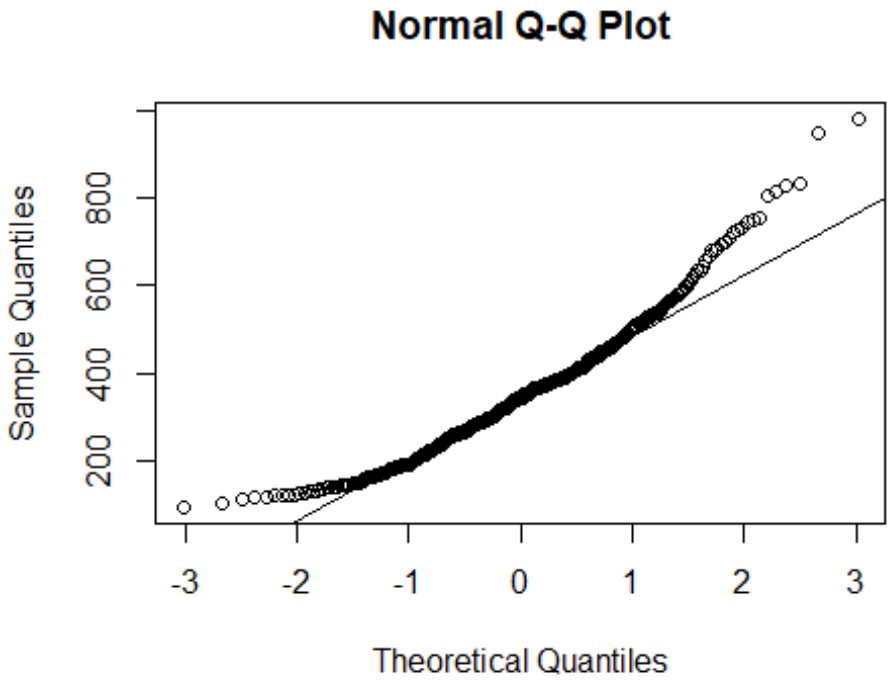
16. Display a histogram for Credit Rating, with the main title “Credit Rating Histogram” and X label “Rating”

```
hist(Credit$Rating, main="Credit Rating Histogram", xlab="Rating")
```



17. Check the qqplot (using the qqnorm() function and qqline() to evaluate the normality of credit rating).

```
qqnorm(Credit$Rating)
qqline(Credit$Rating)
```

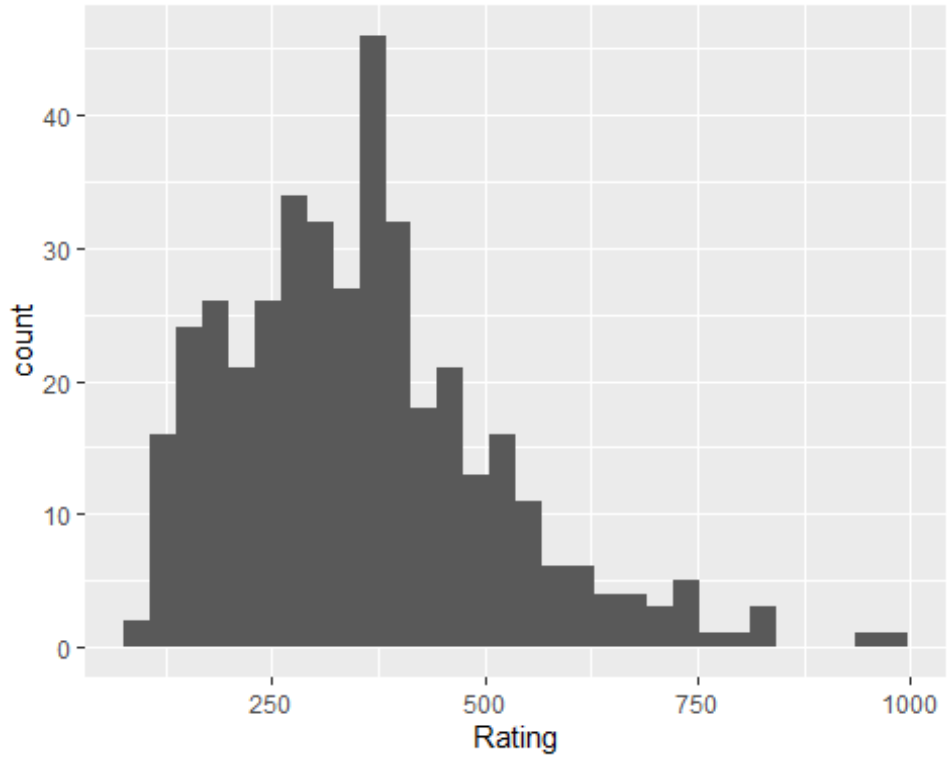


18. Briefly answer: Do you think that this data is somewhat normally distributed? Why or why not?

```
# The data is somewhat normal, but the qqplot deviates from the qqline thus
providing some indication of non-normality. The histogram shows some skewness
to the left
```

19. Install (only if not installed already) and load the "ggplot2" package and draw a histogram with the ggplot() function

```
require(ggplot2)
ggplot(data=Credit) + geom_histogram(aes(x = Rating))
```



20. Then draw a dual line plot with Credit Rating on the Y axis and Income on the X axis, separated by gender (i.e., facet_wrap())

```
ggplot(Credit, aes(y=Rating, x=Income)) + geom_line() + facet_wrap(~Gender)
```