Learning R for Data Analysis: Project One

Tshepo Ralehoko

24 June 2018

Contents

List of Figures

List of Tables

Acknowledgements

I want to acknowledge the useful of the book Elements of Statistical Learning for the theory that is related to the results herein, analysis and interpretation thereof. The book has been a great resource for further developing my statistical computing skills in *R* and data analysis in general. Furthermore, many thanks to various platforms which are easily accessible, and with great provision and support in *R* related content for data analysis.

List of Notations

σ^2 : Variance

σ : Standard Deviation

μ : Mean

List of Keywords

Bias-Variance Trade-off

Correlation Matrix

Goodness of fit

Kurtosis

Probability Density Curve

Random Variable

Skewness

Training set

Test set

Configurations

Working directory

Below is the directory that was created for the project as it pertains to the laptop that was used. This can be changed accordingly depending on where the user wants to save their work. We also go ahead and load the dataset.

```
#clearing the work space
rm(list = ls())
#the current working directory
getwd()
[1] "C:/Users/Tshepo Ralehoko/Downloads/Data Science/Data Science - R/Projects/Credit data Project 1"

#setting up the working directory
setwd(file.path("C:", "Users", "Tshepo Ralehoko", "Downloads",
                "Data Science","Data Science - R", "Projects",
                "Credit data Project 1"))

#loading the dataset into R
credit <- read.table(file = file.path("C:", "Users","Tshepo Ralehoko", "Downloads",
                                      "Data Science", "Data Science - R",
                                      "Projects", "Credit data Project 1",
                                      "data.txt"),
                                      header = TRUE, sep = '')
```

Introduction

This is a personal project. The project deals with the well-known *Credit* dataset. A brief description of the dataset shall follow. The aim of the project is to build the best model for predicting the output variable using, all or a subset of the features. On that note, we wish to indicate that the dataset we shall be dealing with falls into the *supervised learning* paradigm. For assessing the accuracy of the models in predicting the corresponding *target* variable, we will generate and utilize the necessary *goodness of fit* statistics. We will also keep an eye of the *bias-variance trade-off* during the model building process. This concept is explained in detail under the **Data Modelling** section. Furthermore, we want to underscore that the project is for learning purposes, and as a result, any constructive input is appreciate.

For achieving the aim of the project, our dataset will be randomly split into a *training* and *test* set. We will sometimes refer to the latter set as the *validation* set. This method is widely used for validating the accuracy and performance of the model in predicting observations that were not used in building or training the model (out-of-sample observations).

The next section will take a look at **Data Description and Data Summary**. It will use various functions to study the structure of the dataset; the variables that make up the dataset and summary statistics. The **Data Preparation** section is dedicated to addressing any issues that we might have discovered in the preceding section, and taking the corrective steps to prepare the dataset for model building and analysis.

Data Description and Data Summary

The dataset has *11* predictor variables, and each of the variables contains *400* observations. The names of the features are: Income (in thousands of dollars), Limit (credit limit), Rating (credit rating), Cards (number of credit cards), Age, Education (years of education), Gender, Student (student status), Married (marital status) and Ethnicity (Caucasian, African American or Asian). The class of the variables is split among *integer*, *factor* and *numeric* variables. The dataset has complete cases. For convenience, we will sometimes use the variable names (which provide less description about the variables) instead of the relevant description of the variables. For instance, we might use Education to refer to "the number of years of education".

The results from this section are very insightful, and allows the user to pursue other data mining techniques on the dataset that are beyond the scope covered by the project. This is done to accommodate any further analysis that may be of interest on the dataset in the future. I want to reiterate that, with this project, I desire to take a pragmatic approach and learn new skills beyond what I have gathered from the classroom environment during the course of my studies in Data Science. We now focus our attention to the plots, figures and tables from the *R* output.

In search of discovering interesting insights into the dataset, we decided to plot the column means by both gender and marital status. Below in figure 1 and figure 2 we have the plots of the *column means* by gender and marital status respectively. We have only used numerical variables for the below stacked barplots.

From the plots we can also see that the average of most of the variables is very small across gender and marital status. Further insights from table 2 shows that this is in fact the case for the *column means* for a few variables when ignoring the groupings by gender and marital status. The Limit variable dominates both stacked barcharts with its large mean. Taking a closer look, the average credit limit of females is greater than that of males.

Figure 1: A baplot - Column Means by Gender

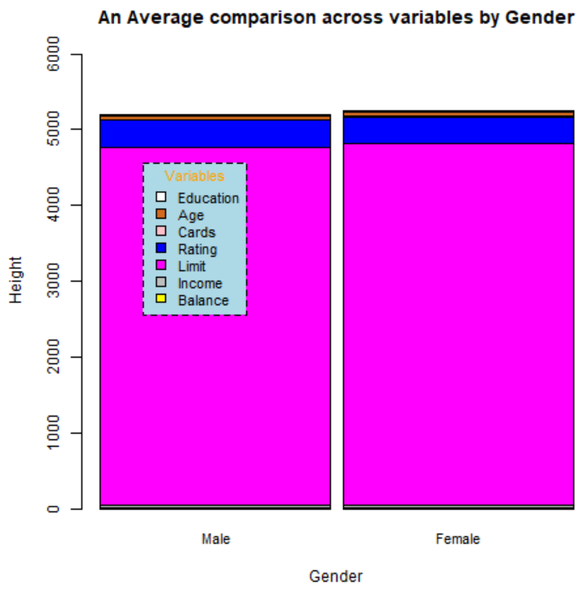
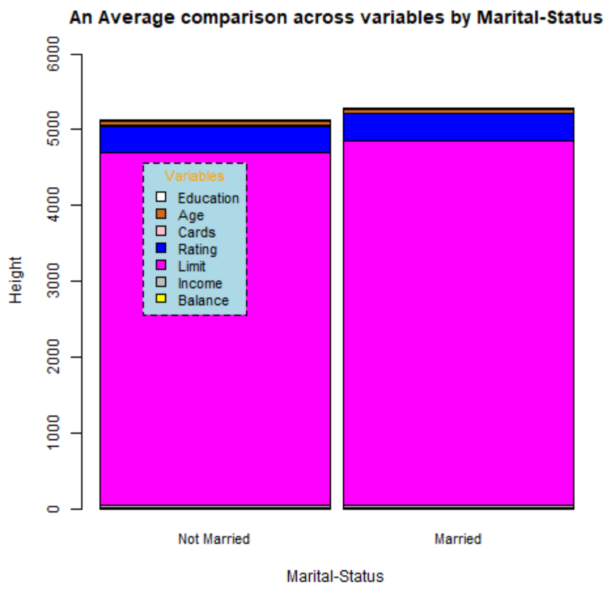


Figure 2: A barplot - Column Means by Marital Status



Below is a box-and-whisker diagram of the numeric variables. We have also marked the outliers (in asterisk-like characters) using a magenta colour. Certainly, these variables can be thought of as *random variables*. In this light, the plot also plays an important role in aiding us to get a rough idea of the distribution of our random variables. It is clear from the figure that the *Limit* predictor variable has a distribution whose underlying statistics can be uniquely identified in this case. It is characterized by a large variance and mean and several outliers on the *upper fence*.

The lower fence and upper fence are situated below the whisker at the bottom of the box and above the whisker at the top side of the box respectively. The respective values for these fences are computed as follows:

$$\text{Lower fence} = Q_1 - 1.5 \times IQR$$

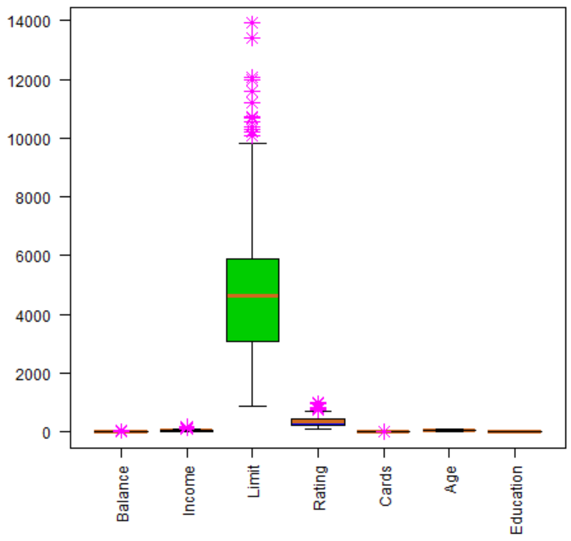
$$\text{Upper fence} = Q_3 + 1.5 \times IQR$$

where,

- Q_1 is the first or lower quartile
- Q_3 is the third or upper quartile
- IQR is the interquartile range which is obtained by subtracting Q_1 from Q_3

It is important to keep the outliers in mind when building models. Outliers are generally undesirable and need to be scrutinized in the model building process. Bearing in mind that these are observations that do not fit the general pattern observed in the dataset, they can cause misleading interpretation. For instance, a case could arise where a model is rejected due to a violation of model assumptions caused by outliers, when in actual fact, the correct model is chosen for the dominant pattern of observations in the dataset. This discussion pertains to figure 3 below.

Figure 3: Box and whisker plot - Numeric Variables



The next figure looks at the column variances of our numeric variables. The results below in figure 4 are not surprising. Similar information can be seen from the box-and-whisker plot in figure 3. Therefore, for some analysis, it might be a good idea to standardize the variables so that no one variable is dominant over the others. In any case, we will not consider scaling or standardizing the dataset.

Figure 4: A barplot Depicting Column Variances

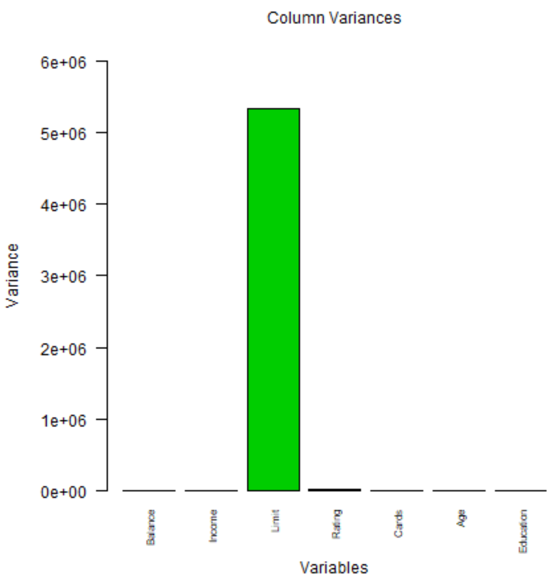
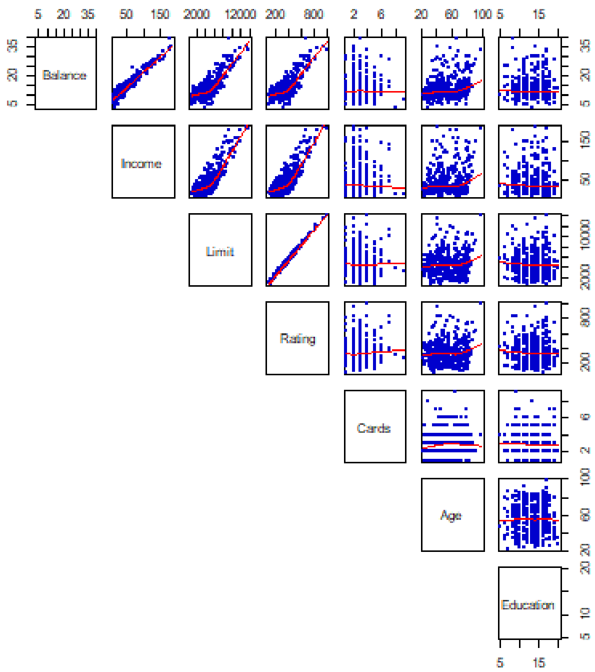


Figure 5 is a plot that represents pairwise scatter plots of the variables. Included in the scatter plots is a "smooth" curve that is fitted to the data points. There is clearly a *linear* association between *Limit* and *Rating*.

This is because the data points from the corresponding scatter plot of the two variables can be determined using a liner model that is approximately deterministic . Additionally, similar associated is observed between *Income* and *Limit* and between *Income* and *Rating*, but the strength of the relationships is not as strong.

The strength of the associations between various pairs of variables are found in the correlation matrix in table 1. In reality, we can expect credit card limit to be proportional to income. However, a domain expert would not more about these intricacies. This phenomenon of association between features is known as *collinearity*. From the plot, we also see that quite a number of features seem to be linearly related to the target variable.

Figure 5: Pairwise Scatter Plots of Numeric Variables



When the data points from the pairwise plots are plotted by gender or marital status, it would seem that it is a challenge to pick up any apparent pattern between the variables across both the levels of the gender and marital status factor variable. For this information we refer to figure 6 and figure 7

Figure 6: Pairwise Plots - Points Plotted by Marital-Status

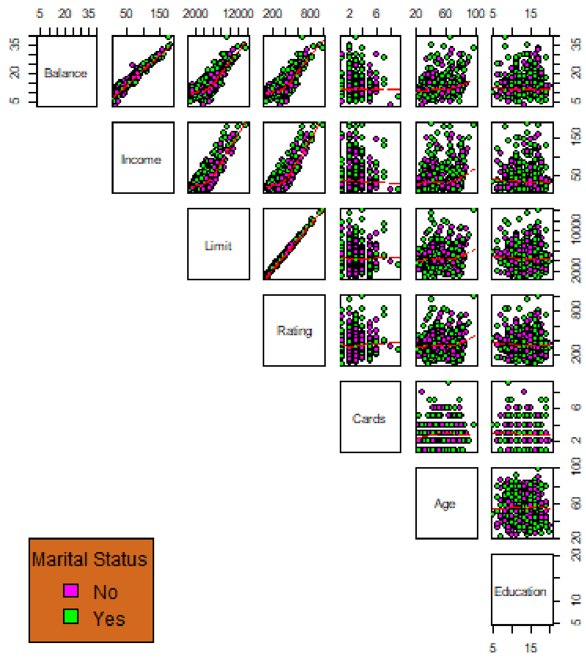


Figure 7: Pairwise Plots - Points Plotted by Gender

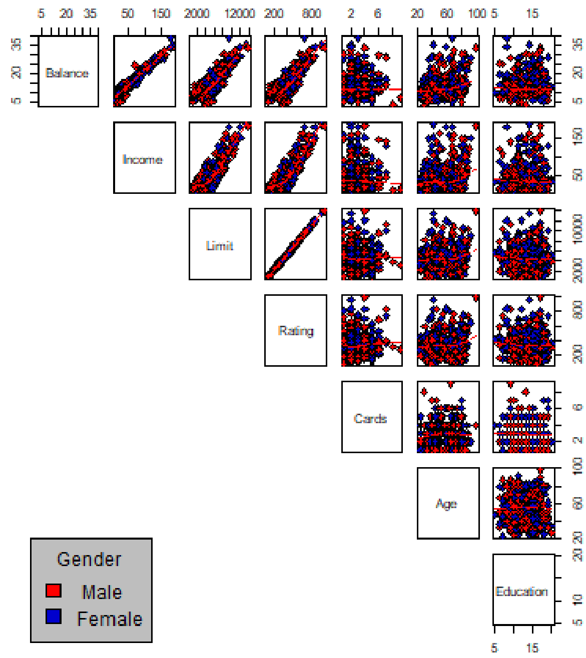


Table 1 below shows the output from the correlation matrix of the numeric variables. It is a measure of the strength of the linear association between the combination of pairwise scatter plots of the numeric variables in the dataset.

Table 1: Correlation Matrix - Tabular representation

Variable	Balance	Income	Limit	Rating	Cards	Age	Education
Balance	1.00	0.97	0.76	0.76	-0.01	0.23	0.01
Income	0.97	1.00	0.79	0.79	-0.02	0.18	-0.03
Limit	0.76	0.79	1.00	1.00	0.01	0.10	-0.02
Rating	0.76	0.79	1.00	1.00	0.05	0.10	-0.03
Cards	-0.01	-0.02	0.01	0.05	1.00	0.04	-0.05
Age	0.23	0.18	0.10	0.10	0.04	1.00	0.00
Education	0.01	-0.03	-0.02	-0.03	-0.05	0.00	1.00

Table 2: Distribution Statistics

	σ^2	σ	μ	minimum	maximum	range	Q_1	Q_2	Q_3	IQR	kurtosis	skewness
Balance	32.14	5.67	13.43	3.75	38.79	35.04	9.89	11.78	15.24	5.35	2.58	1.54
Income	1242.16	35.24	45.22	10.35	186.63	176.28	21.01	33.12	57.47	36.46	2.87	1.73
Limit	5327781.92	2308.20	4735.60	855.00	13913.00	13058.00	3088.00	4622.50	5872.75	2784.75	0.96	0.83
Rating	23939.56	154.72	354.94	93.00	982.00	889.00	247.25	344.00	437.25	190.00	1.01	0.86
Cards	1.88	1.37	2.96	1.00	9.00	8.00	2.00	3.00	4.00	2.00	0.90	0.79
Age	297.56	17.25	55.67	23.00	98.00	75.00	41.75	56.00	70.00	28.25	-1.08	0.01
Education	9.77	3.13	13.45	5.00	20.00	15.00	11.00	14.00	16.00	5.00	-0.60	-0.33

Figure 8: Probability Density Curve of Balance

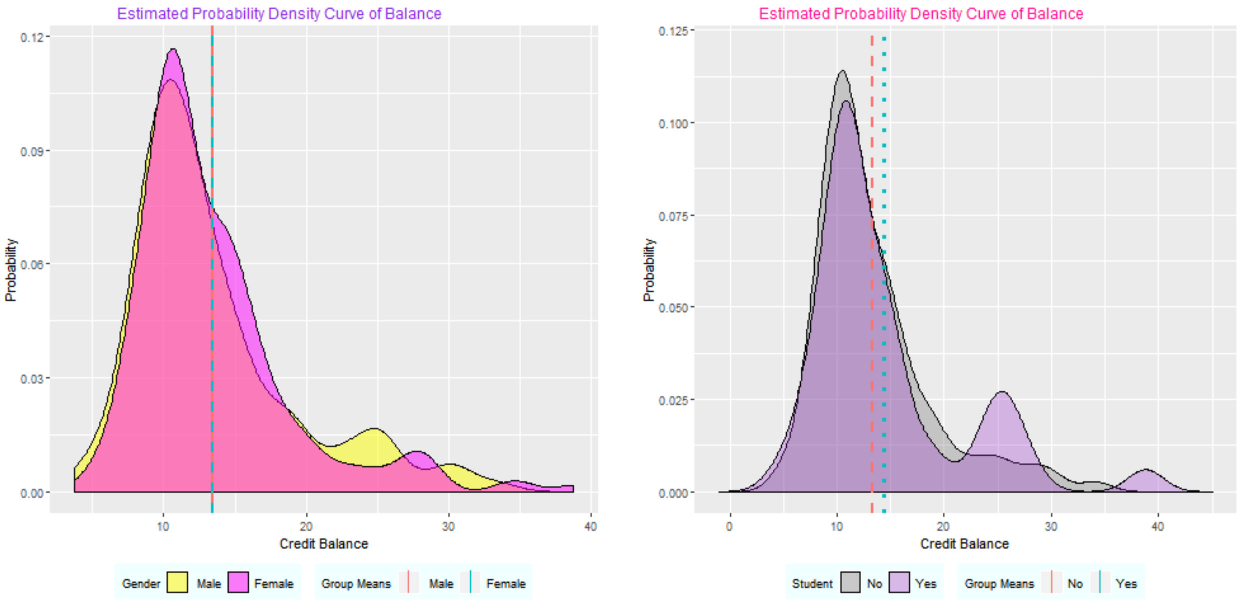


Figure 9: Probability Density Curve of Income

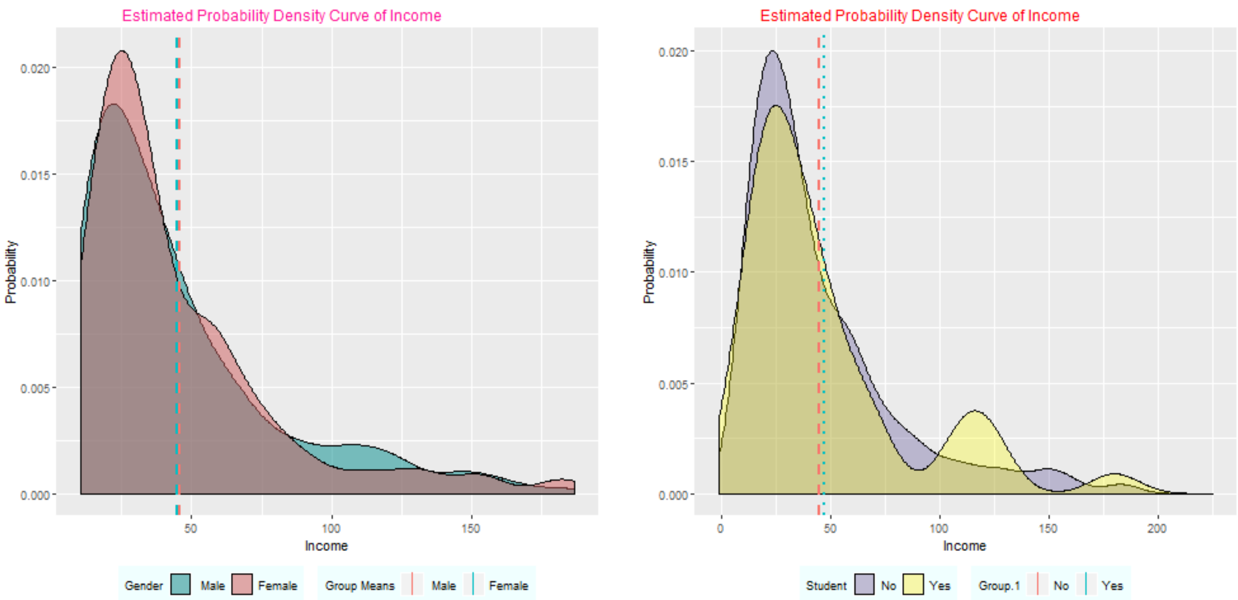


Figure 10: Probability Density Curve of Limit

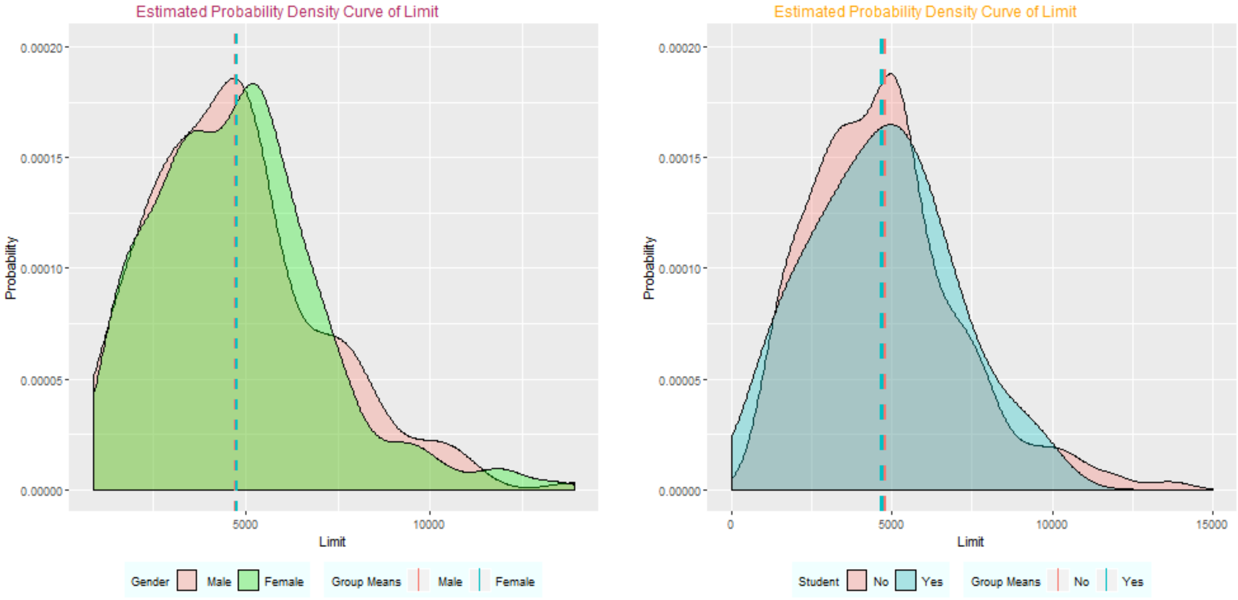


Figure 11: Probability Density Curve of Rating

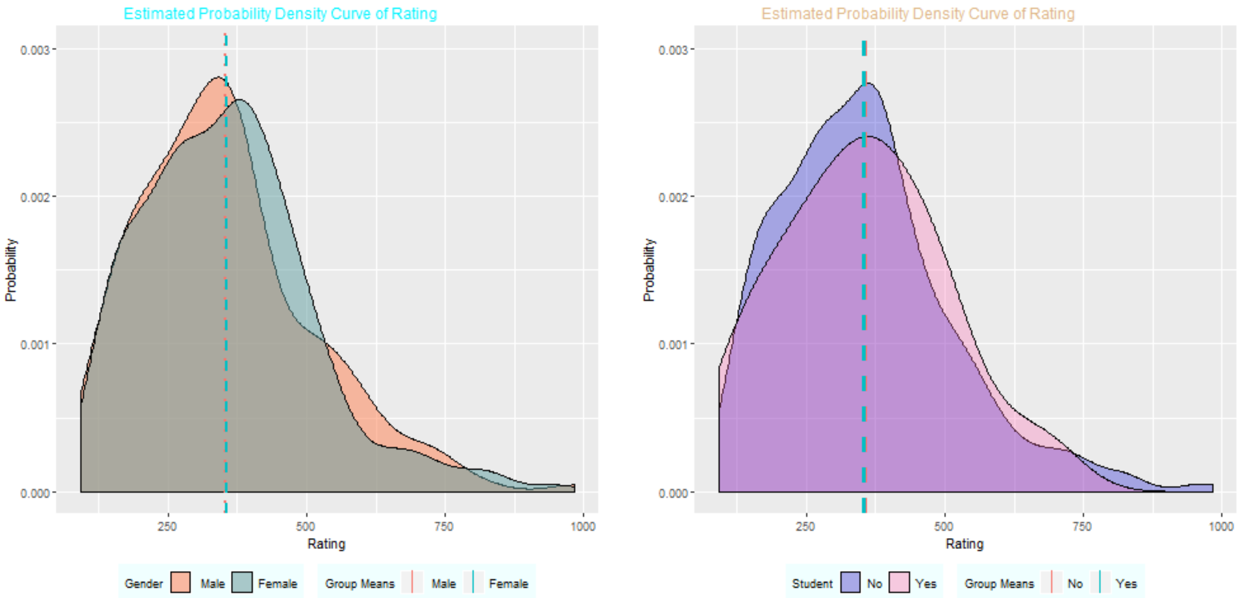


Figure 12: Probability Density Curve of Cards

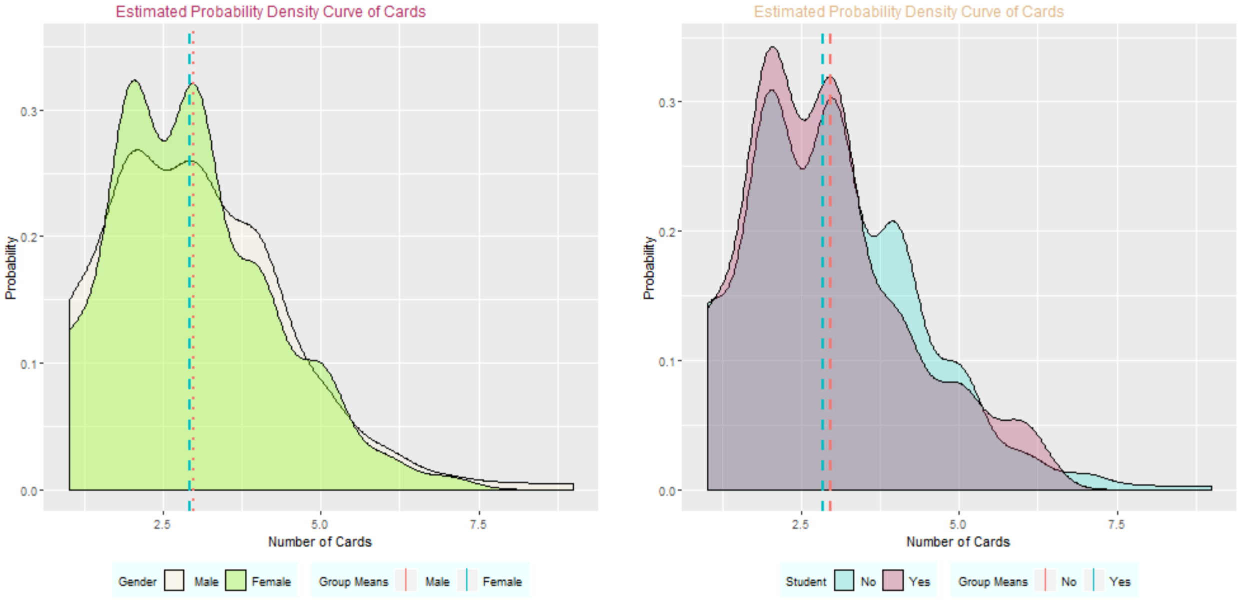


Figure 13: Probability Density Curve of Age

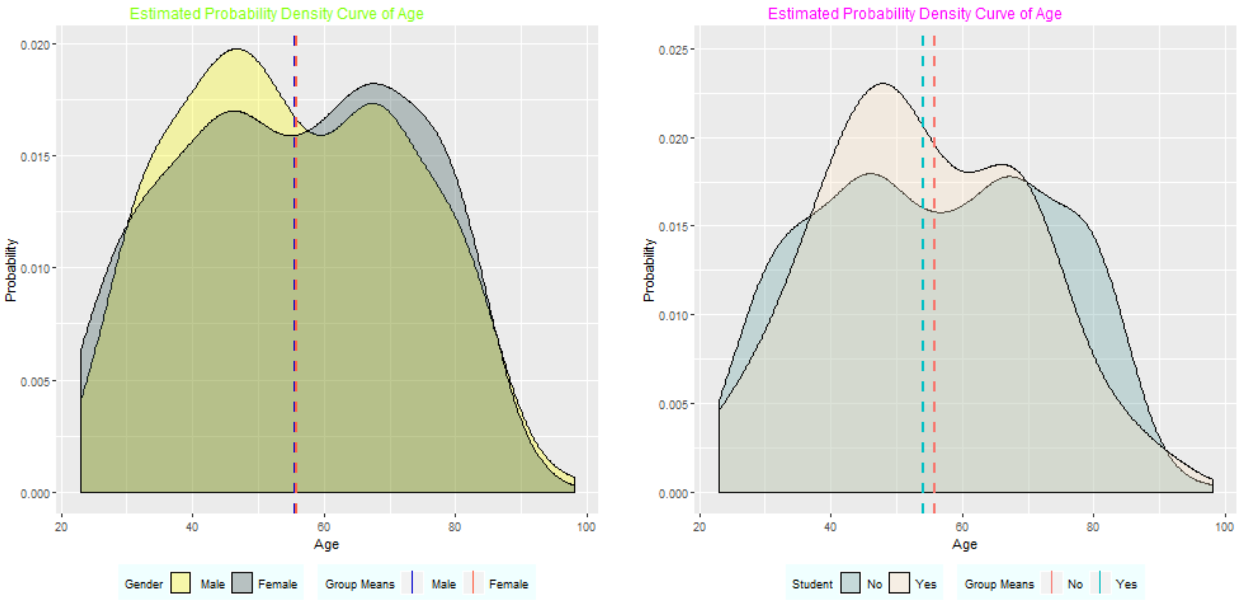


Figure 14: Probability Density Curve of Education

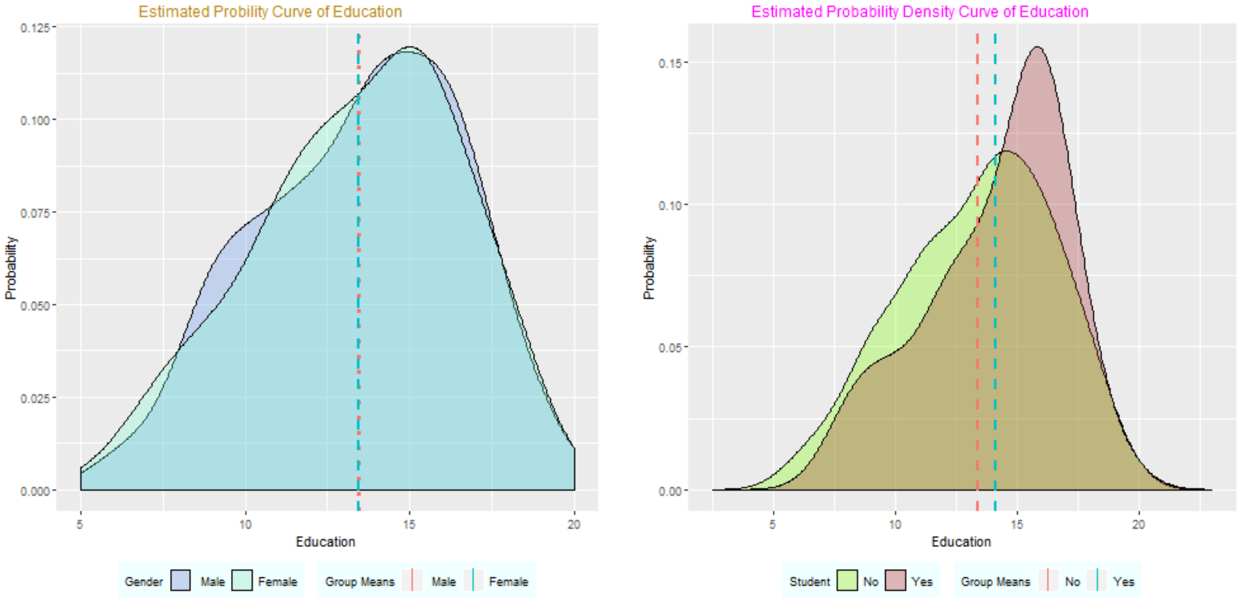


Figure 15: Boxplot of Balance Random Variable

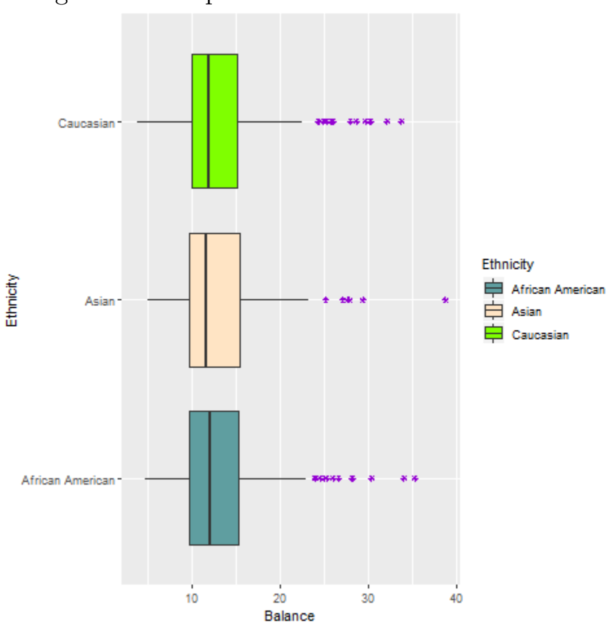


Figure 16: Boxplot of Income Random Variable

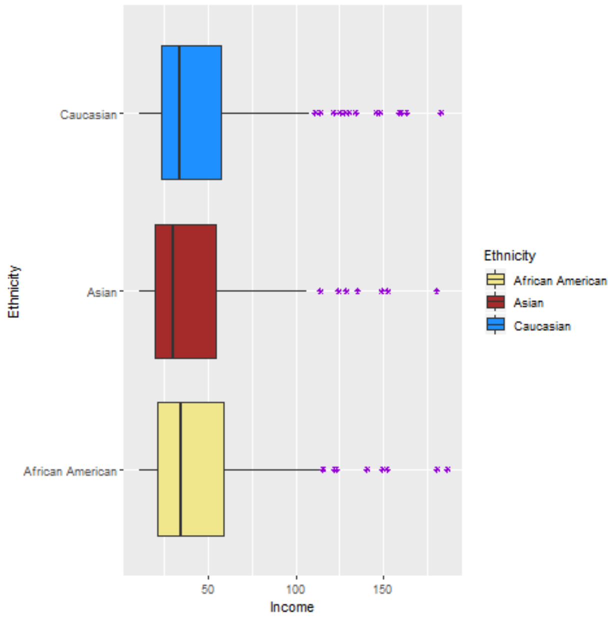


Figure 17: Boxplot of Limit Random Variable

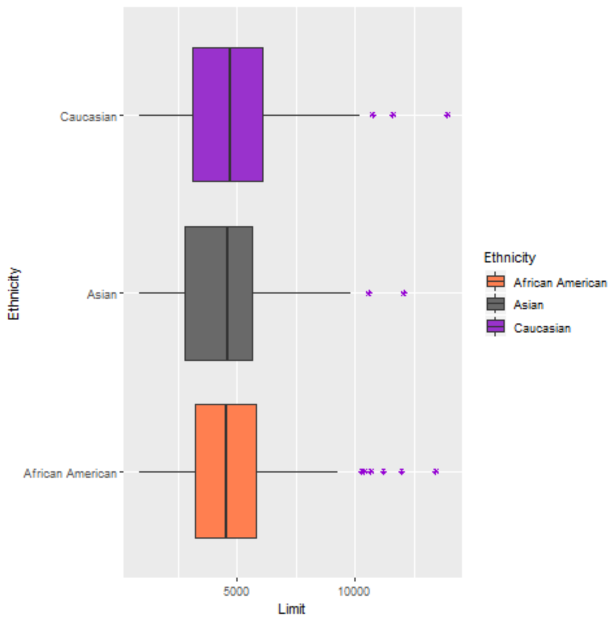


Figure 18: Boxplot of Rating Random Variable

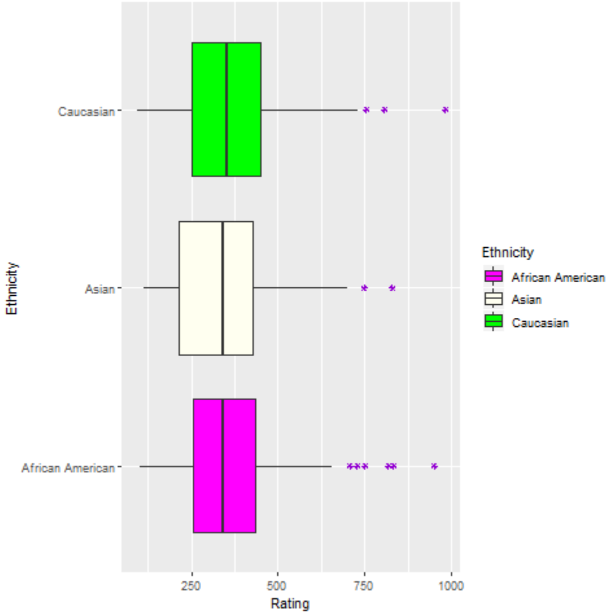


Figure 19: Boxplot of Cards Random Variable

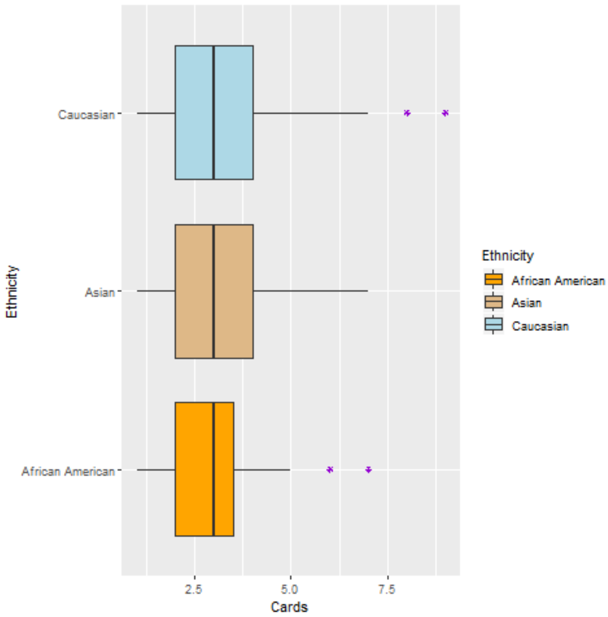


Figure 20: Boxplot of Age Random Variable

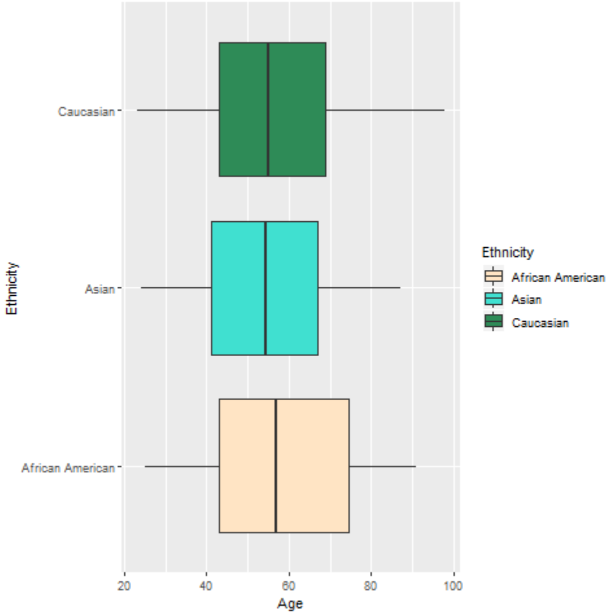
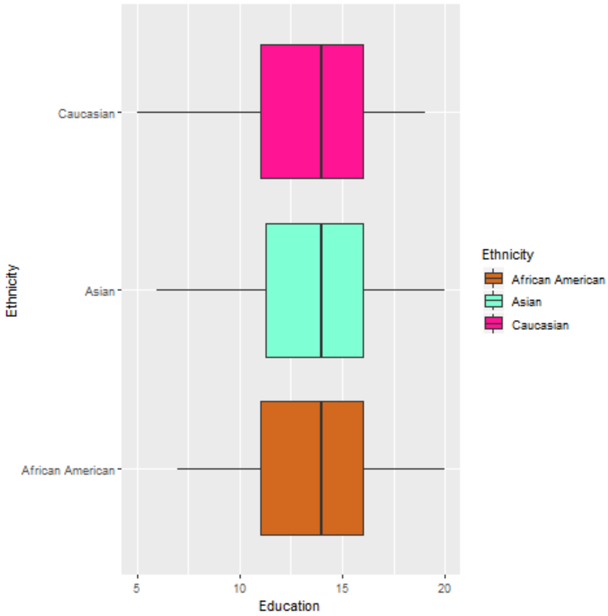


Figure 21: Boxplot of Education Random Variable




```
#taking a peek at the dataset
head(credit, 10)
    Balance   Income Limit Rating Cards Age Education Gender Student
1  12.24080  14.891  3606    283     2  34        11   Male      No
2  23.28333 106.025  6645    483     3  82        15 Female     Yes
3  22.53041 104.593  7075    514     4  71        11   Male      No
```

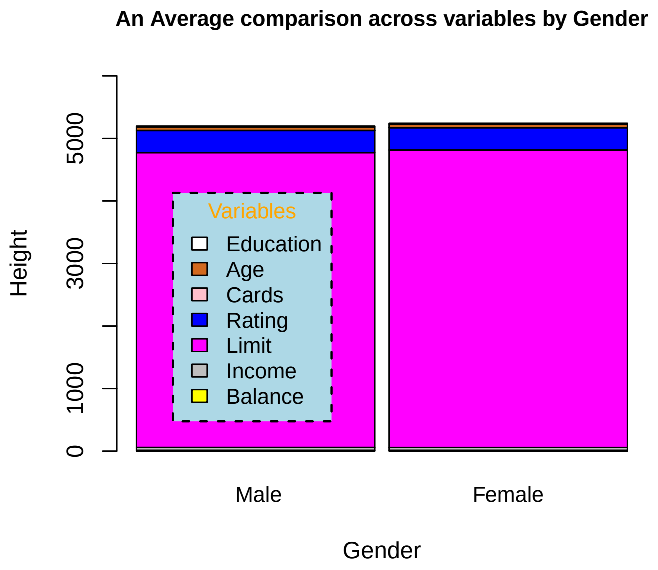
```
4  27.65281 148.924  9504    681     3  36         11 Female      No
5  16.89398  55.882  4897    357     2  68         16   Male      No
6  22.48618  80.180  8047    569     4  77         10   Male      No
7  10.57452  20.996  3388    259     2  37         12 Female      No
8  14.57620  71.408  7114    512     2  87          9   Male      No
9   7.93809  15.125  3300    266     5  66         13 Female      No
10 17.75696  71.061  6819    491     3  41         19 Female     Yes
   Married        Ethnicity
1      Yes        Caucasian
2      Yes            Asian
3       No            Asian
4       No            Asian
5      Yes        Caucasian
6       No        Caucasian
7       No African American
8       No            Asian
9       No        Caucasian
10     Yes African American


#the number of rows and columns
dim(credit)
[1] 400  11


#the structure of the dataset
str(credit)
'data.frame':   400 obs. of  11 variables:
 $ Balance  : num  12.2 23.3 22.5 27.7 16.9 ...
 $ Income   : num  14.9 106 104.6 148.9 55.9 ...
```

```
 $ Limit    : int  3606 6645 7075 9504 4897 8047 3388 7114 3300 6819 ...
 $ Rating   : int  283 483 514 681 357 569 259 512 266 491 ...
 $ Cards    : int  2 3 4 3 2 4 2 2 5 3 ...
 $ Age      : int  34 82 71 36 68 77 37 87 66 41 ...
 $ Education: int  11 15 11 11 16 10 12 9 13 19 ...
 $ Gender   : Factor w/ 2 levels " Male","Female": 1 2 1 2 1 1 2 1 2 2 ...
 $ Student  : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...
 $ Married  : Factor w/ 2 levels "No","Yes": 2 2 1 1 2 1 1 1 1 2 ...
 $ Ethnicity: Factor w/ 3 levels "African American",..: 3 2 2 2 3 3 1 2 3 1 ...

#we get a count of the number of missing cases or observations
sum(complete.cases(credit) == FALSE)
[1] 0

#calling the names in the data frame into the working space
attach(credit)



#doing a comparison of each of the numeric variables among the two genders
aggr_gender <- aggregate.data.frame(x = credit[, -c(8:11)],
                     by = list(Gender), data = credit,
                     FUN = mean, simplify = TRUE)
aggr_gender
  Group.1  Balance   Income    Limit   Rating    Cards      Age Education
1    Male 13.44544 45.61032 4713.166 353.5181 2.989637 55.59585  13.46632
2  Female 13.41401 44.85393 4756.517 356.2657 2.927536 55.73430  13.43478
```

```
#the library with color pallettes that we want to use

library(RColorBrewer)



#the color pallette for our bar graph

colors <- brewer.pal(n = 7,name = "Set1")



#a list of color pallettes to choose from

brewer.pal.info

         maxcolors category colorblind

BrBG            11      div       TRUE

PiYG            11      div       TRUE

PRGn            11      div       TRUE

PuOr            11      div       TRUE

RdBu            11      div       TRUE

RdGy            11      div      FALSE

RdYlBu          11      div       TRUE

RdYlGn          11      div      FALSE

Spectral        11      div      FALSE

Accent           8     qual      FALSE

Dark2            8     qual       TRUE

Paired          12     qual       TRUE

Pastel1          9     qual      FALSE

Pastel2          8     qual      FALSE

Set1             9     qual      FALSE

Set2             8     qual       TRUE

Set3            12     qual      FALSE

Blues            9      seq       TRUE

BuGn             9      seq       TRUE
```

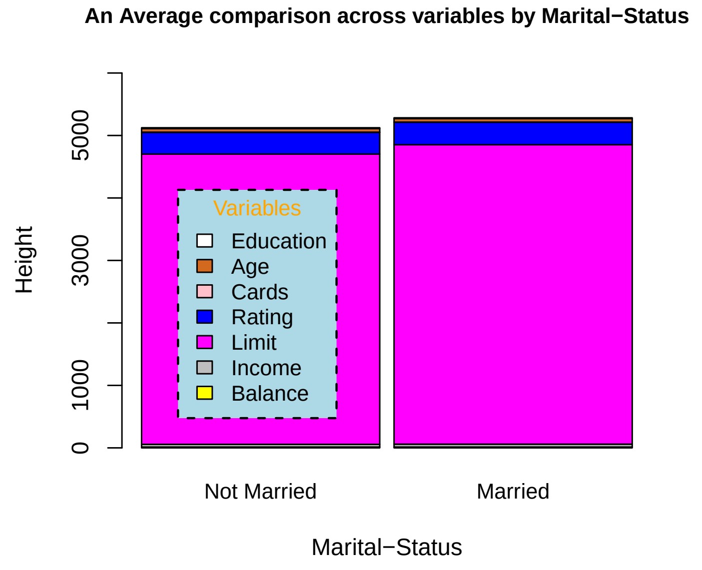
```
BuPu            9       seq       TRUE

GnBu            9       seq       TRUE

Greens          9       seq       TRUE

Greys           9       seq       TRUE

Oranges         9       seq       TRUE

OrRd            9       seq       TRUE

PuBu            9       seq       TRUE

PuBuGn          9       seq       TRUE

PuRd            9       seq       TRUE

Purples         9       seq       TRUE

RdPu            9       seq       TRUE

Reds            9       seq       TRUE

YlGn            9       seq       TRUE

YlGnBu          9       seq       TRUE

YlOrBr          9       seq       TRUE

YlOrRd          9       seq       TRUE


#we created the below code so that we can identify the graphs of the two genders involved

rownames(aggr_gender) <- aggr_gender[,1]

aggr1_gender <- aggr_gender[,-1]




#the stacked bar graphs drawn side-by-side

barplot(height = cbind(t(as.vector(aggr_gender[1, 2:8])),

                       t(as.vector(aggr_gender[2, 2:8]))),

        beside = FALSE, cex.main = 0.9,

        col = c('yellow', 'grey', 'magenta',

                'blue', 'pink', 'chocolate', 'white'),
```

```
        main = 'An Average comparison across variables by Gender',
        horiz = FALSE, xlab = 'Gender', ylab = 'Height', cex.names = 0.9,
        space = 0.06, names.arg = rownames(aggr1_gender), ylim = c(0,6000),
        legend.text = rownames(t(as.vector(aggr_gender[1, 2:8]))),
        args.legend = list(xjust = 2.9, yjust = 1.45,cex = 0.9,
                           bg = 'lightblue', box.lty = 2,
                           box.lwd = 1.5, horiz = FALSE,
                           title = 'Variables', title.col = 'orange'))
```



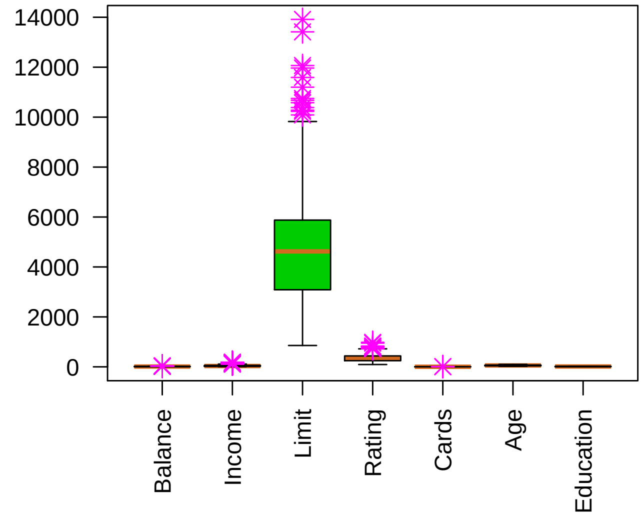

```
#doing a comparision across variables by marital status
aggr_status <- aggregate.data.frame(x = credit[, -c(8:11)],
                    by = list(Married), data = credit, FUN = mean,
                    simplify = TRUE)
aggr_status

  Group.1  Balance   Income    Limit   Rating    Cards      Age Education
1      No 13.49351 43.64109 4645.303 347.8000 2.974194 57.25161  13.25806
2     Yes 13.38847 46.21708 4792.727 359.4571 2.946939 54.66531  13.57143
```

```
#we created the below code so that we can identify the barplots of marital status involved
rownames(aggr_status) <- c('Not Married', 'Married')
aggr_status
            Group.1  Balance   Income    Limit   Rating    Cards      Age
Not Married      No 13.49351 43.64109 4645.303 347.8000 2.974194 57.25161
Married         Yes 13.38847 46.21708 4792.727 359.4571 2.946939 54.66531
            Education
Not Married  13.25806
Married      13.57143
aggr1_status <- aggr_status[,-1]


#the stacked bar graphs drawn side-by-side
barplot(height = cbind(t(as.vector(aggr_status[1, 2:8])),
                       t(as.vector(aggr_status[2, 2:8]))),
        beside = FALSE, cex.main = 0.9,
        col = c('yellow', 'grey', 'magenta', 'blue','pink','chocolate', 'white'),
        main = 'An Average comparison across variables by Marital-Status',
        horiz = FALSE, xlab = 'Marital-Status', ylab = 'Height',
        cex.names = 0.9, space = 0.06, names.arg = rownames(aggr1_status),
        ylim = c(0,6000),
        legend.text = rownames(t(as.vector(aggr_status[1, 2:8]))),
        args.legend = list(xjust = 2.9, yjust = 1.45,cex = 0.90,
                           bg = 'lightblue', box.lty = 2,
                           box.lwd = 1.5, horiz = FALSE,
                           title = 'Variables', title.col = 'orange'))
```

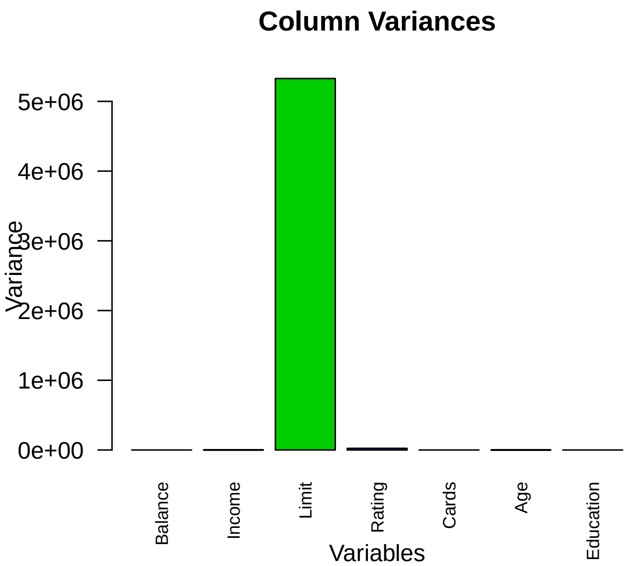



```r
#creating a data frame of the numeric variables

credit_num <- credit[, -c(8:11)]



#creating a data frame of the categorical variables

credit_fac <- credit[, -c(1:7)]



#computing the column variances of the numeric variables



#creating an empty matrix for storing the variances

var <- rep(0, 7)



#corresponding for loop

for (i in 1:7){

  v = var(credit_num[, i]) #temporal storage for the variances

  var[i] = v #printing them into the desired matrix

}
```

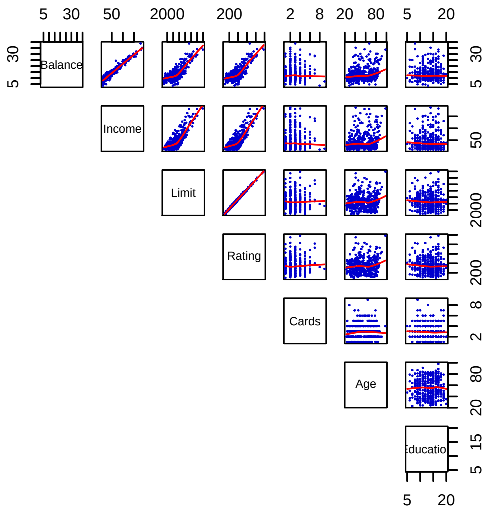
```r
#a box-and-whisker plot of the numeric variables
boxplot.default(credit_num, notch = FALSE, col = 1:7,
                cex = 1.2, boxlty = 1, whisklty = 7, outpch = 8, outcex = 1.5,
                outcol = 'magenta', medcol = 'chocolate', las = 2)
```




```r
#a barplot of the column variances
barplot(var, col = 1:7, names.arg = names(credit_num),
        main = 'Column Variances', cex.names = 0.75, xlab = 'Variables',
        ylab = 'Variance', las = 2)
```

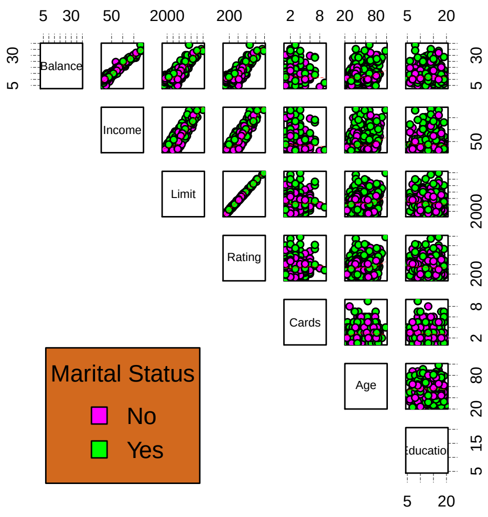


```r
#pairwise scatterplots of the numeric variables
#a smooth curve fitting the scatter plots
pairs(credit_num, pch = 20, lower.panel = NULL,
      upper.panel = panel.smooth, gap = 1,
      col = 'blue3', lty = 1, lwd = 1.2, cex = 0.2,
      oma = c(5, 5, 5, 10))
```




```r
#pairwise plot between numeric variables shown by marital status
#a smooth fitting curve for the scatter plots
pairs(credit_num, pch = 21, bg = c("magenta", "green")[Married],
      upper.panel = panel.smooth, lower.panel = NULL, lty = 5,
      row1attop = TRUE, oma = c(5, 5, 5, 10), lwd = 0.3)
#allowing plotting of the legend outside the figure region
par(xpd  = TRUE)
#legend
legend("bottomleft", fill = c("magenta", "green"),
       legend = c(levels(Married)), bg = 'chocolate',
```

```
       title = 'Marital Status')
```



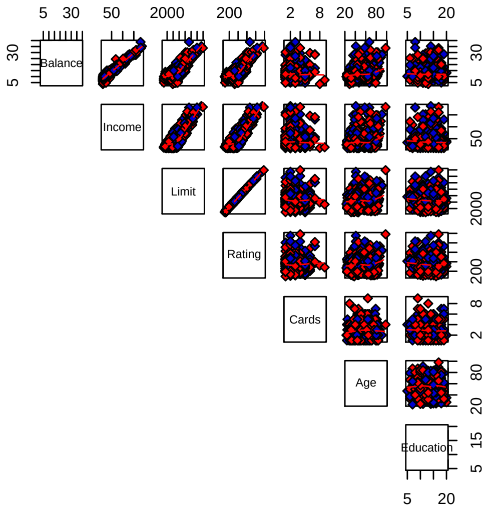

```
#pairwise plot between numeric variables shown by gender
#a smooth curve fitting the scatter plots
pairs(credit_num, pch = 23, bg = c('red', 'blue3')[Gender],
      upper.panel = panel.smooth, lower.panel = NULL, lty = 5,
      oma = c(5, 5, 5, 10))
#allow plotting of the legend outside the figure region
par(xpd = TRUE)
#legend
legend('bottomleft', fill = c("red", "blue3"),
       legend = c(levels(Gender)), bg = 'grey',
       title = "Gender")


#correlation matrix


#a library for creating a table
library(xtable)
```

```r
#tabular representation of a correlation matrix
print(xtable(cor(credit_num)), type = 'latex', comment = FALSE)
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrrrr}
  \hline
 & Balance & Income & Limit & Rating & Cards & Age & Education \\
  \hline
Balance & 1.00 & 0.97 & 0.76 & 0.76 & -0.01 & 0.23 & 0.01 \\
  Income & 0.97 & 1.00 & 0.79 & 0.79 & -0.02 & 0.18 & -0.03 \\
  Limit & 0.76 & 0.79 & 1.00 & 1.00 & 0.01 & 0.10 & -0.02 \\
  Rating & 0.76 & 0.79 & 1.00 & 1.00 & 0.05 & 0.10 & -0.03 \\
  Cards & -0.01 & -0.02 & 0.01 & 0.05 & 1.00 & 0.04 & -0.05 \\
  Age & 0.23 & 0.18 & 0.10 & 0.10 & 0.04 & 1.00 & 0.00 \\
  Education & 0.01 & -0.03 & -0.02 & -0.03 & -0.05 & 0.00 & 1.00 \\
   \hline
\end{tabular}
\end{table}


#summary statistics and more statistics


#declaring the matrices for storing the summary statistics
var_num <- matrix(0, nrow = 7, byrow = TRUE)
mean_num <- matrix(0, nrow = 7, byrow = TRUE)
min_num <- matrix(0, nrow = 7, byrow = TRUE)
max_num <- matrix(0, nrow = 7, byrow = TRUE)
range_num <- matrix(0, nrow = 7, byrow = TRUE)
```

```r
median_num <- matrix(0, nrow = 7, byrow = TRUE)

sd_num <- matrix(0, nrow = 7, byrow = TRUE)

IQR_num <- matrix(0, nrow = 7, byrow = TRUE)

Q1_num <- matrix(0, nrow = 7, byrow = TRUE)

Q3_num <- matrix(0, nrow = 7, byrow = TRUE)

skew_num <- matrix(0, nrow = 7, byrow = TRUE)

kurt_num <- matrix(0, nrow = 7, byrow = TRUE)



#the package is helpful for computing kurtosis and skewness

library(e1071)


#computing the aforementioned statistics

for (i in 1:length(credit_num)){

  var_num[i] = var(credit_num[, i]) #matrix of variances

  mean_num[i] = mean(credit_num[, i]) #matrix of means

  min_num[i] = min(credit_num[, i]) #matrix of minima

  max_num[i] = max(credit_num[, i]) #matrix of maxima

  range_num = max_num - min_num #matrix of range values

  median_num[i] = median(credit_num[, i]) #matrix of medians

  sd_num[i] = sd(credit_num[, i]) #matrix standard deviations

  IQR_num[i] = IQR(credit_num[, i]) #matrix of Interquantile range values

  Q1_num[i] = quantile(credit_num[, i], probs = 0.25) #matrix of first quantile range values

  Q3_num[i] = quantile(credit_num[, i], probs = 0.75) #matrix of third quantile range values

  kurt_num[i] = kurtosis(credit_num[, i]) #matrix of kurtosis values

  skew_num[i] = skewness(credit_num[, i]) #matrix of skewness values

}
```

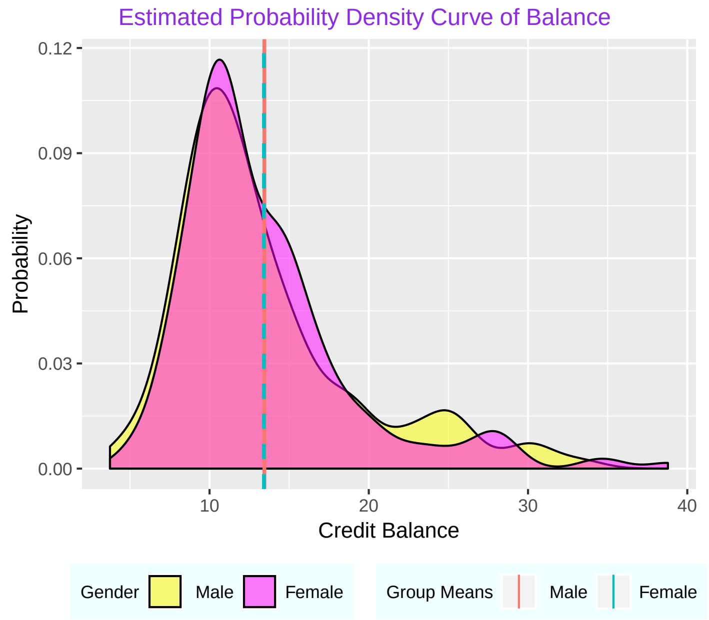
```
#the distribution of the variables


#aggregate statistics any other statistics of the data
summary_stats <- data.frame(var = var_num, std = sd_num,
                            mean = mean_num, minimum = min_num,
                            maximum = max_num, range = range_num,
                            Q1 = Q1_num, Q2 = median_num,
                            Q3 = Q3_num, IQR = IQR_num,
                            kurtosis = kurt_num, skewness = skew_num)


#including rownames to the data frame
rownames(summary_stats) <- names(credit_num)


#library for creating a table for the results above
library(xtable)


#table for the results
print(xtable(summary_stats), type = 'latex',
      table.placement = "H", include.colnames = TRUE,
      include.rownames = TRUE, comment = FALSE)
\begin{table}[H]
\centering
\begin{tabular}{rrrrrrrrrrrrr}
  \hline
 & var & std & mean & minimum & maximum & range & Q1 & Q2 & Q3 & IQR & kurtosis & skewness \\
  \hline
Balance & 32.14 & 5.67 & 13.43 & 3.75 & 38.79 & 35.04 & 9.89 & 11.78 & 15.24 & 5.35 & 2.58 & 1.54 \\
  Income & 1242.16 & 35.24 & 45.22 & 10.35 & 186.63 & 176.28 & 21.01 & 33.12 & 57.47 & 36.46 & 2.87 & 1.73
```

```
  Limit & 5327781.92 & 2308.20 & 4735.60 & 855.00 & 13913.00 & 13058.00 & 3088.00 & 4622.50 & 5872.75 & 27
  Rating & 23939.56 & 154.72 & 354.94 & 93.00 & 982.00 & 889.00 & 247.25 & 344.00 & 437.25 & 190.00 & 1.01
  Cards & 1.88 & 1.37 & 2.96 & 1.00 & 9.00 & 8.00 & 2.00 & 3.00 & 4.00 & 2.00 & 0.90 & 0.79 \\
  Age & 297.56 & 17.25 & 55.67 & 23.00 & 98.00 & 75.00 & 41.75 & 56.00 & 70.00 & 28.25 & -1.08 & 0.01 \\
  Education & 9.77 & 3.13 & 13.45 & 5.00 & 20.00 & 15.00 & 11.00 & 14.00 & 16.00 & 5.00 & -0.60 & -0.33 \\
   \hline
\end{tabular}
\end{table}

#computing the group means of our numeric variables by student status
aggr_student_status <- aggregate(credit_num, by = list(Student), FUN = mean,
                                simplify = TRUE)

#by ethnicity
aggr_ethnicity <- aggregate(credit_num, by = list(Ethnicity), FUN = mean,
                            simplify = TRUE)

#the library below is going to be used for graphics

#we are going to plot the estimated probability density function
#of our numerical variables by gender and student status
library(ggplot2)
```



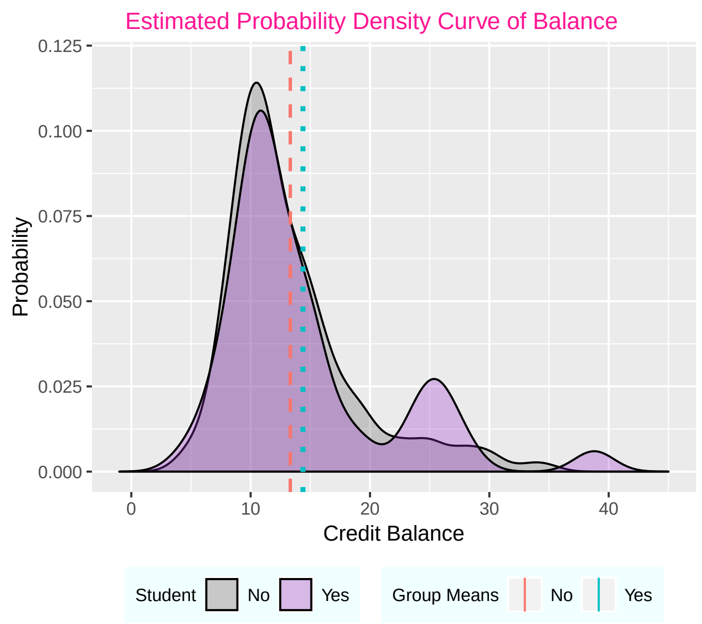

```r
#side-by-side (gender + student status)


#Balance

ggplot(data = credit,

       mapping = aes(x = Balance, fill = Gender)) +

geom_density(alpha = 0.5) +

geom_vline(data = aggr_gender,

           mapping = aes(xintercept = Balance ,colour = Group.1),

           linetype = c(1, 2), lwd = c(0.9, 0.9)) +

labs(x = "Credit Balance",

       title = "Estimated Probability Density Curve of Balance",

       y = "Probability") +

theme(plot.title = element_text(size = 12, face = "plain",

          color = "blueviolet", hjust = 0.3, vjust = 0.7),

        legend.position = "bottom", legend.title =

          element_text(size = 09, face = "plain"),

        legend.direction = "horizontal",
```

```r
      legend.background = element_rect(fill = "azure", linetype = 1)) +
scale_color_discrete(aes(colour = "Group Means")) +
scale_fill_manual(values = c("yellow", "magenta")) +
guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2))
```



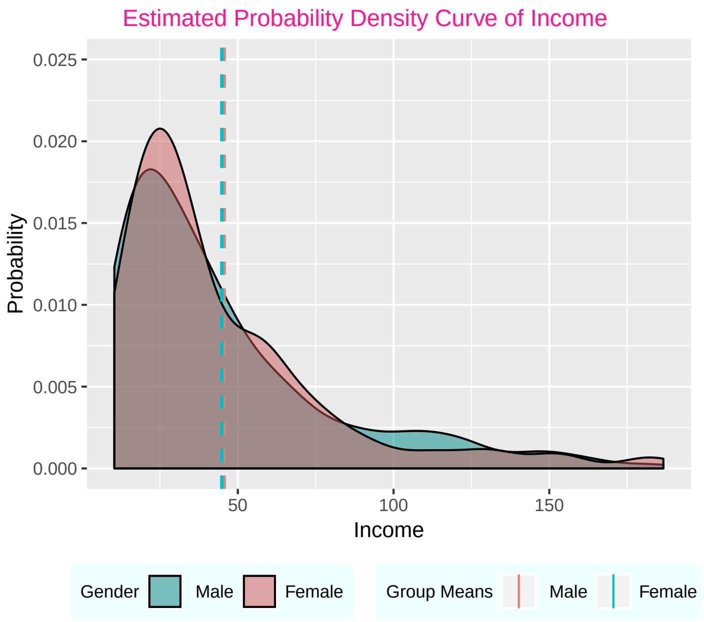

```r
ggplot(data = credit, mapping  = aes(x = Balance, fill = Student)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_student_status,
           aes(xintercept = aggr_student_status[, 2], colour = Group.1),
           linetype = c(2, 3), lwd = c(0.8, 1.2)) +
labs(title = "Estimated Probability Density Curve of Balance",
     x = "Credit Balance", y = "Probability") +
theme(legend.position = "bottom",
      legend.background = element_rect(fill = "azure", linetype =  1),
      plot.title = element_text(vjust = 0.5, hjust = 0.3, face = "plain",
          size = 12, colour = "deeppink"), legend.direction = "horizontal",
      legend.title = element_text(size = 09, face = "plain")) +
```

```
scale_colour_discrete(aes(colour = "Group Means")) +
scale_fill_manual(values = c("dimgrey", "darkorchid")) +
guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +
ylim(0, 0.12) + xlim(-1, 45)
```



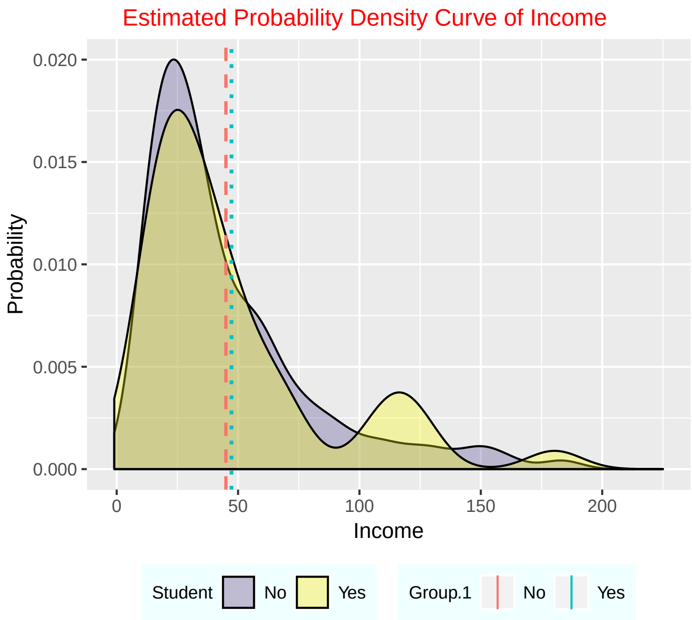

```
#income
ggplot(data = credit,
       mapping = aes(x = Income, fill = Gender)) +
geom_density(alpha = 0.5) +
geom_vline(data = aggr_gender,
             mapping = aes(xintercept = Income, colour = Group.1),
      linetype = c(2,2), lwd = c(0.8, 0.8)) +
scale_colour_discrete(aes(colour = "Group Means")) +
labs(title = "Estimated Probability Density Curve of Income",
     x = "Income", y = "Probability") +
theme(plot.title = element_text(size = 12, face = "plain",
          hjust = 0.3, vjust = 0.5, colour = "deeppink"),
      legend.position = "bottom",legend.direction = "horizontal",
```

```
      legend.title = element_text(size = 09, face = "plain"),

      legend.background = element_rect(fill = "azure",

                                       linetype = 1)) +

scale_fill_manual(values = c("darkcyan", "indianred")) +

guides(colour = guide_legend(order = 2), fill = guide_legend(order = 1)) +

ylim(0, 0.025)
```



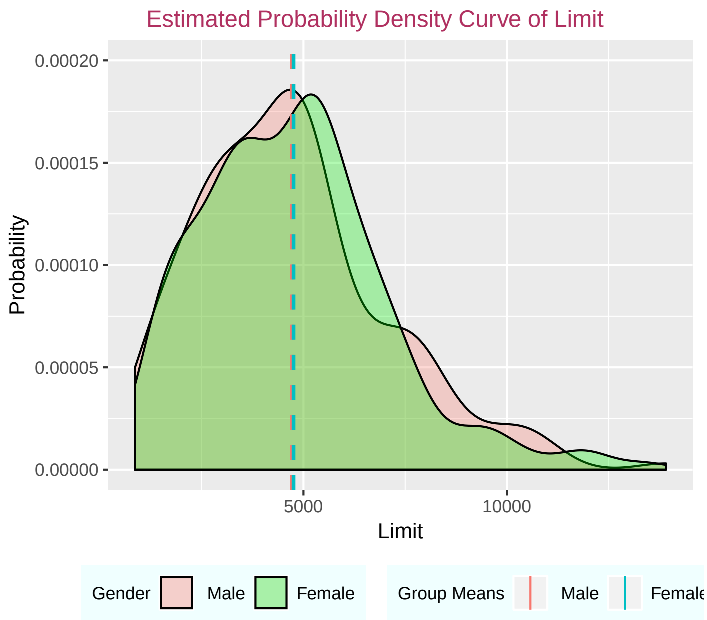

```
ggplot(data = credit,

       mapping = aes(x = Income, fill = Student)) +

geom_density(alpha = 0.3) +

geom_vline(data = aggr_student_status,

           aes(xintercept = Income, colour = Group.1),

      linetype = c(2, 3), lwd = c(0.8, 0.9)) +

labs(title = "Estimated Probability Density Curve of Income",

     y = "Probability", x = "Income") +

theme(plot.title = element_text(size = 12, face = "plain",

              hjust = 0.3, vjust = 0.5, color = "red"),
```

```
    legend.background = element_rect(fill = "azure", linetype = 1),
    legend.position = "bottom", legend.title =
      element_text(size = 09, face = "plain")) +
scale_fill_manual(values = c("darkslateblue", "yellow")) +
guides(colour = guide_legend(order = 2), fill = guide_legend(order = 1)) +
xlim(-1, 225)
```



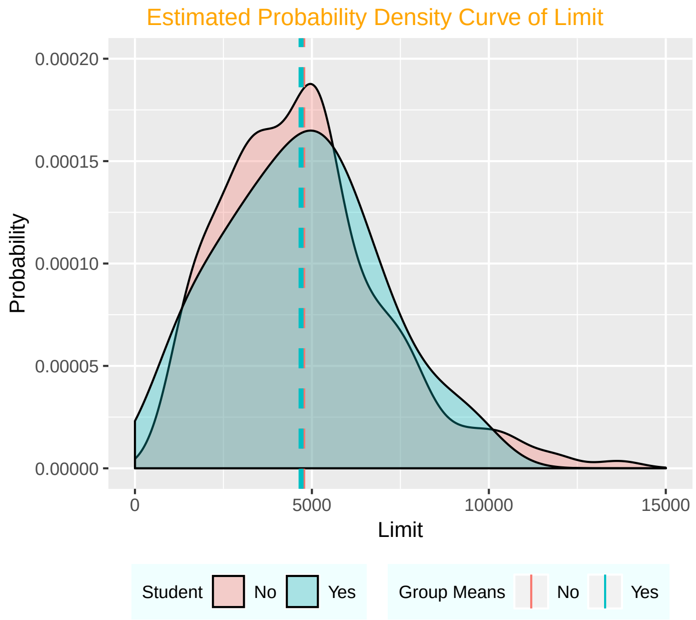

```
#limit
ggplot(data = credit,
    mapping = aes(x = Limit, fill = Gender)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_gender,
      aes(xintercept = Limit, colour = Group.1),
    linetype = c(2, 2), lwd = c(0.9, 0.9)) +
labs(x = "Limit", y = "Probability",
    title = "Estimated Probability Density Curve of Limit") +
theme(legend.background = element_rect(fill = "azure",
      linetype = 1), legend.title = element_text(size = 09, face = "plain"),
```

```
  legend.position = "bottom", legend.direction = "horizontal",

  plot.title = element_text(size = 12, face = "plain", vjust = 0.5, hjust = 0.3,

      colour = "maroon")) +

scale_fill_manual(values = c("salmon", "green2")) +

scale_colour_discrete(aes(colour = "Group Means")) +

guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +

ylim(0, .00020)
```



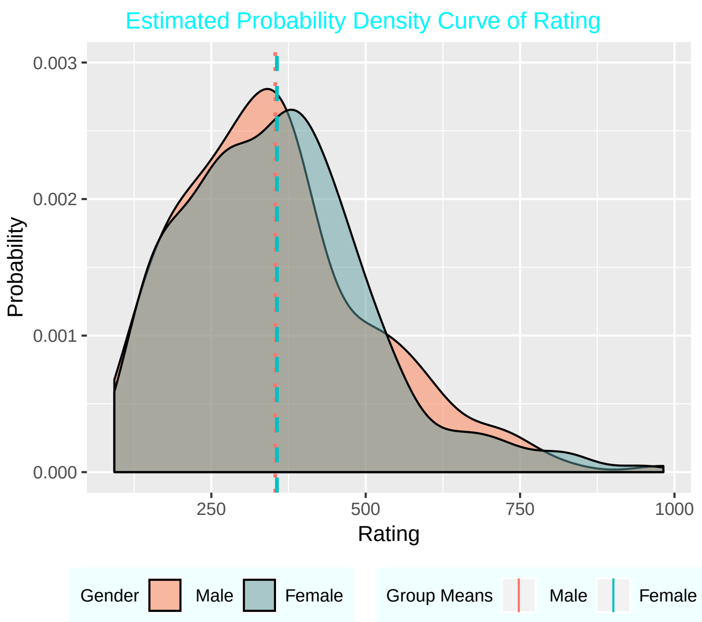

```
ggplot(data = credit,

       mapping = aes(x = Limit, fill = Student)) +

geom_density(alpha = 0.3) +

geom_vline(data = aggr_student_status,

           aes(xintercept = Limit, colour = Group.1),

      linetype = c(2, 2), lwd = c(1.2, 1.2)) +

theme(legend.background = element_rect(fill = "azure", linetype = 1),

    legend.position = "bottom", legend.title =

    element_text(size = 09, face = "plain" ),
```

```r
  plot.title = element_text(size = 12, face = "plain",
                            colour = "orange", hjust = 0.3, vjust = 0.5)) +
labs(x = "Limit", y = "Probability",
     title = "Estimated Probability Density Curve of Limit") +
scale_colour_discrete(aes(colour = "Group Means")) +
guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +
ylim(0,0.00020) + xlim(0, 15000)
```



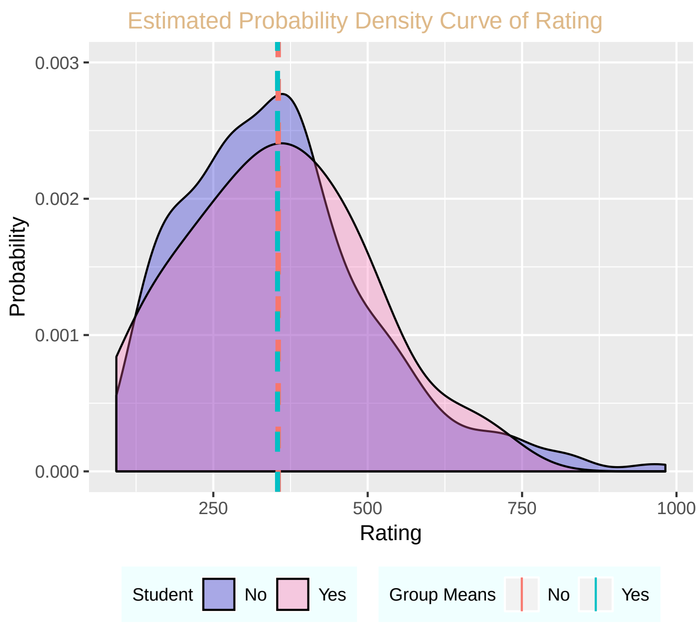

```r
#rating
ggplot(data = credit,
       mapping = aes(x = Rating, fill = Gender)) +
geom_density(alpha = 0.5) +
geom_vline(data = aggr_gender,
           mapping = aes(xintercept = Rating, colour = Group.1),
           linetype = c(3, 2), lwd = c(0.9, 0.9)) +
labs(x = "Rating", y = "Probability",
     title = "Estimated Probability Density Curve of Rating") +
```

```
theme(plot.title = element_text(size = 12, face = "plain", hjust = 0.3,
                         vjust = 0.5,color = "turquoise1"),
      legend.title = element_text(size = 9, face = "plain"),
      legend.position = "bottom", legend.direction = "horizontal",
    legend.background = element_rect(fill = "azure", linetype = 1)) +
scale_fill_manual(values = c("coral", "cadetblue")) +
scale_colour_discrete(aes(colour = "Group Means")) +
guides(color = guide_legend(order = 2), fill = guide_legend(order = 1)) +
ylim(0, 0.003)
```



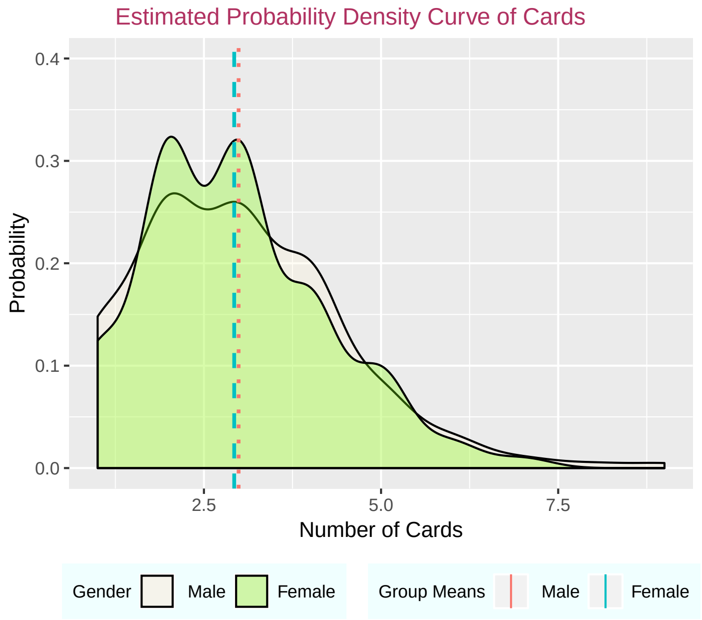

```
ggplot(data = credit,
       mapping = aes(x = Rating, fill = Student)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_student_status, aes(xintercept = Rating,
          colour = Group.1), linetype = c(2, 2), lwd = c(1.3, 1.2)) +
labs(x = "Rating", y = "Probability",
     title = "Estimated Probability Density Curve of Rating") +
```

```
theme(plot.title = element_text(colour = "burlywood", size = 12,
      face = "plain", hjust = 0.3, vjust = 0.5),
    legend.background = element_rect(fill = "azure", linetype = 1),
    legend.position = "bottom",
    legend.title = element_text(size = 9)) +
scale_colour_discrete(aes(colour = "Group Means")) +
scale_fill_manual(values = c("blue3", "hotpink")) +
guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +
ylim(0, 0.003)
```



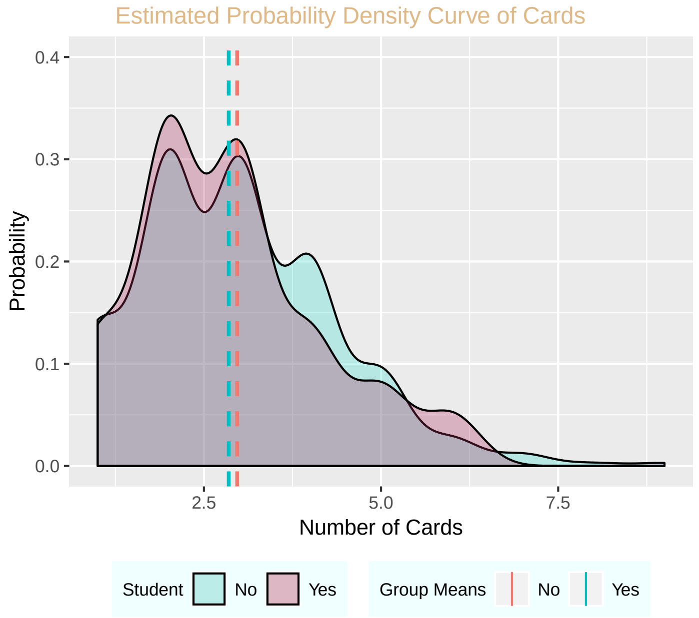

```
#cards
ggplot(data = credit,
       mapping = aes(x = Cards, fill = Gender)) +
geom_density(alpha = .3) +
geom_vline(data = aggr_gender,
             mapping = aes(xintercept = Cards, colour = Group.1),
      linetype = c(3, 2), lwd = c(0.9, 0.9)) +
```

```
labs(x = "Number of Cards", y = "Probability",
     title = "Estimated Probability Density Curve of Cards") +
theme(plot.title = element_text(size = 12, face = "plain", vjust = 0.5,
          hjust = 0.3, colour = "maroon"), legend.direction = "horizontal",
      legend.position = "bottom", legend.title =
        element_text(size = 09, face = "plain"),
    legend.background = element_rect(fill = "azure", linetype = 1)) +
scale_fill_manual(values = c("cornsilk", "chartreuse")) +
scale_colour_discrete(aes(colour = "Group Means")) +
guides(colour = guide_legend(order = 2),
       fill = guide_legend(order = 1)) +
ylim(0, 0.4)
```



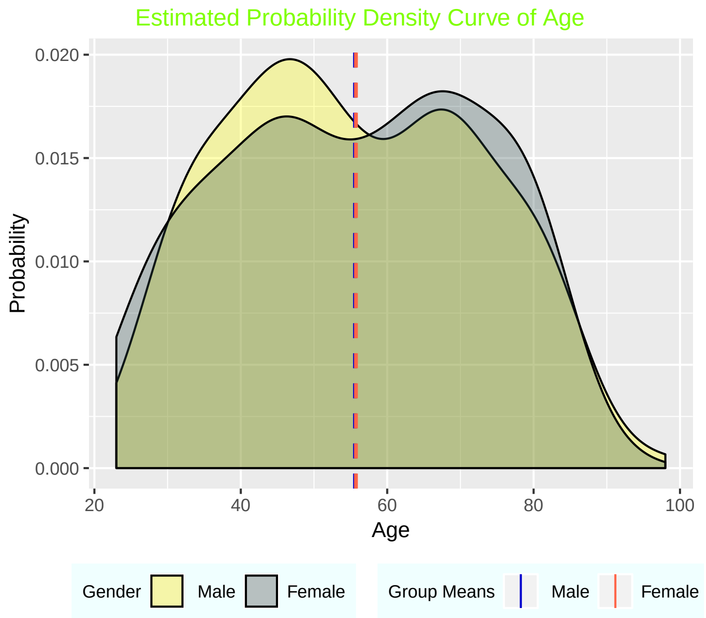

```
ggplot(data = credit,
       mapping = aes(x = Cards, fill = Student)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_student_status,
```

```r
      aes(xintercept = Cards, colour = Group.1),

  linetype = c(2, 2), lwd = c(0.9, 0.9)) +

labs(x = "Number of Cards", y = "Probability",

     title = "Estimated Probability Density Curve of Cards") +

theme(legend.title = element_text(size = 9, face = "plain"), plot.title =

  element_text(size = 12, face = "plain", hjust = 0.3, vjust = 0.5,

    colour = "burlywood"), legend.background = element_rect(fill = "azure",

      linetype = 1), legend.position = "bottom", legend.direction =

    "horizontal") +

scale_colour_discrete(aes(colour = "Group Means")) +

scale_fill_manual(values = c("turquoise", "maroon")) +

guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +

ylim(0,0.4)
```



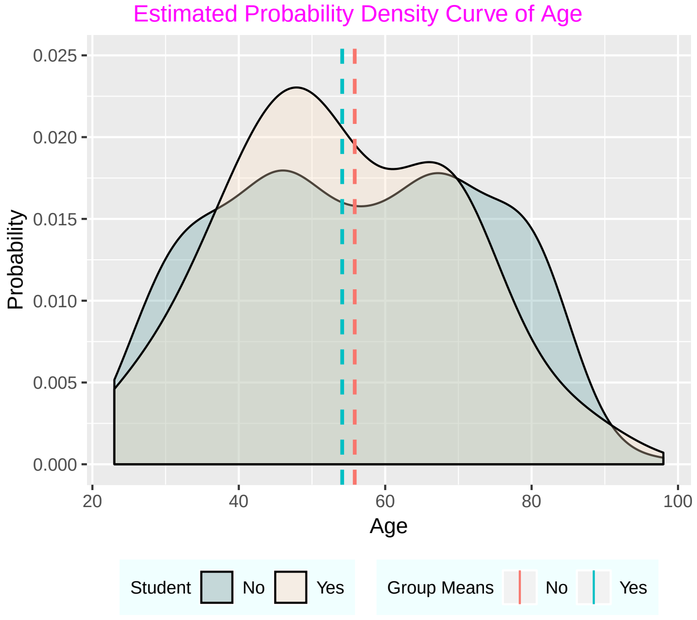

```r
#age

ggplot(data = credit,

      mapping = aes(x = Age, fill = Gender)) +
```

```r
geom_density(alpha = 0.3) +

geom_vline(data = aggr_gender, aes(xintercept = Age, colour = Group.1),

    linetype = c(2, 2), lwd = c(0.9, 0.9)) +

labs(x = "Age", y = "Probability",

    title = "Estimated Probability Density Curve of Age") +

theme(legend.background = element_rect(fill = "azure", linetype = 1),

    legend.title = element_text(size = 9, face = "plain"),

    plot.title = element_text(size = 12, face = "plain", vjust = 0.5,

          hjust = 0.3, colour = "chartreuse"), legend.position = "bottom",

    legend.direction = "horizontal") +

scale_fill_manual(values = c("yellow", "darkslategrey")) +

scale_colour_manual(values = c("blue3", "tomato"),

                    aes(colour = "Group Means")) +

guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2))
```



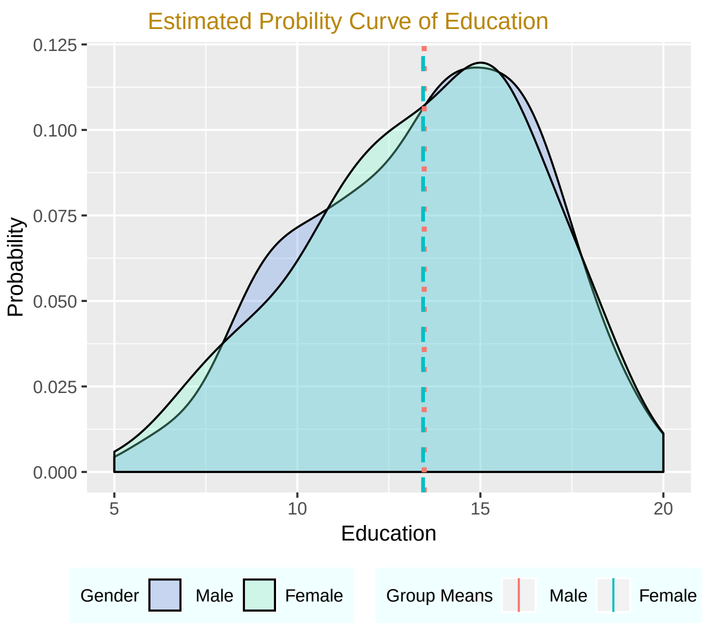

```r
ggplot(data = credit,

       mapping = aes(x = Age, fill = Student)) +
```

```
geom_density(alpha = 0.3) +

geom_vline(data = aggr_student_status,

      aes(xintercept = Age, colour = Group.1),

  linetype = c(2, 2), lwd = c(0.9, 0.9)) +

labs(x = "Age", y = "Probability",

  title = "Estimated Probability Density Curve of Age") +

theme(legend.title = element_text(face = "plain", size = 9),

  legend.background = element_rect(fill = "azure",

      linetype = 1), legend.position = "bottom",

  legend.direction = "horizontal", plot.title = element_text(size = 12,

      colour = "magenta", hjust = 0.3, vjust = 0.5, face = "plain")) +

scale_fill_manual(values = c("cadetblue", "bisque")) +

scale_colour_discrete(aes(colour = "Group Means")) +

guides(fill = guide_legend(order = 1), colour = guide_legend(order = 2)) +

ylim(0, 0.025)
```

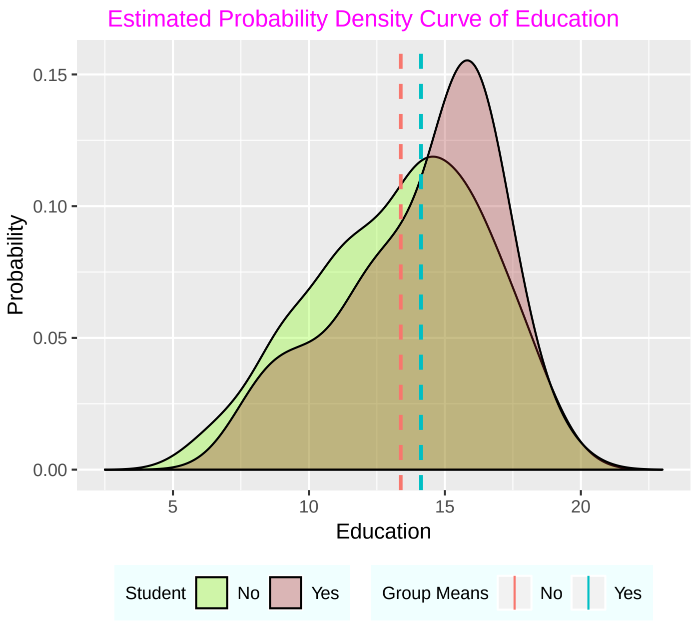



```
#education
```

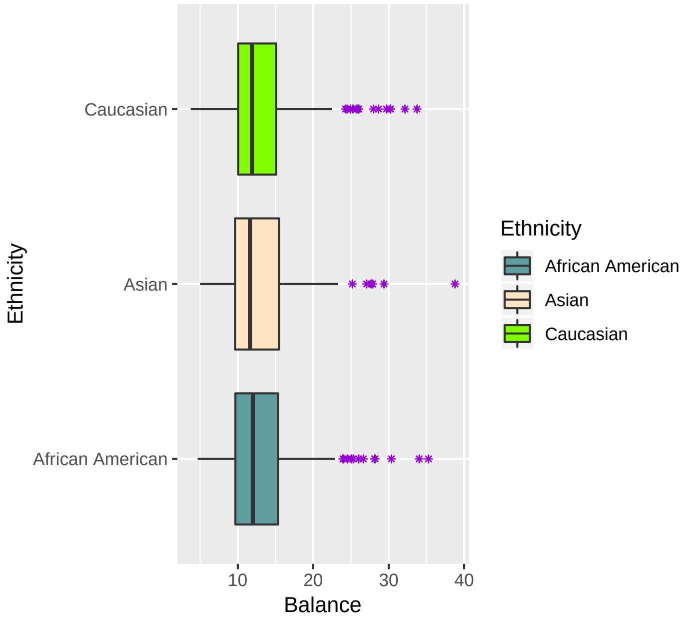
```r
ggplot(data = credit,
       mapping = aes(x = Education, fill = Gender)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_gender,
             mapping = aes(xintercept = Education, colour = Group.1),
    linetype = c(3, 2), lwd = c(1.2, 0.9)) +
labs(x = "Education", y = "Probability",
     title = "Estimated Probility Curve of Education") +
theme(plot.title = element_text(size = 12, face = 'plain', colour = "darkgoldenrod", hjust = 0.3, vjust =
      legend.direction = "horizontal", legend.title =
        element_text(face = "plain", size = 9), legend.box = "horizontal",
      legend.background = element_rect(fill = "azure",
                                       linetype = 1)) +
scale_fill_manual(values = c("cornflowerblue", "aquamarine")) +
scale_colour_discrete(aes(colour = "Group Means")) +
guides(colour = guide_legend(order = 2),
       fill = guide_legend(order = 1))
```



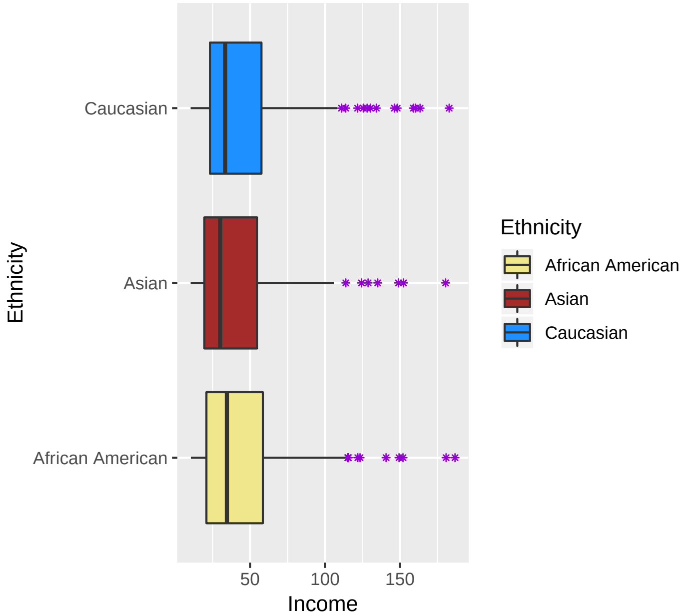
```
ggplot(data = credit,
      mapping = aes(x = Education, fill = Student)) +
geom_density(alpha = 0.3) +
geom_vline(data = aggr_student_status,
  aes(xintercept = Education, colour = Group.1), linetype = c(2, 2),
  lwd = c(0.9, 0.9)) +
labs(x = "Education", y = "Probability",
    title = "Estimated Probability Density Curve of Education") +
theme(legend.title = element_text(size = 9, face = "plain"),
    legend.background = element_rect(fill = "azure", linetype = 1),
    legend.position = "bottom", legend.direction = "horizontal",
  plot.title = element_text(size = 12, face = "plain", colour = "magenta",
          hjust = 0.3, vjust = 0.5)) +
scale_colour_discrete(aes(colour = "Group Means")) +
scale_fill_manual(values = c("chartreuse", "brown")) +
guides(fill = guide_legend(order = 1), guide_legend(order = 2)) +
xlim(2.5, 23)
```



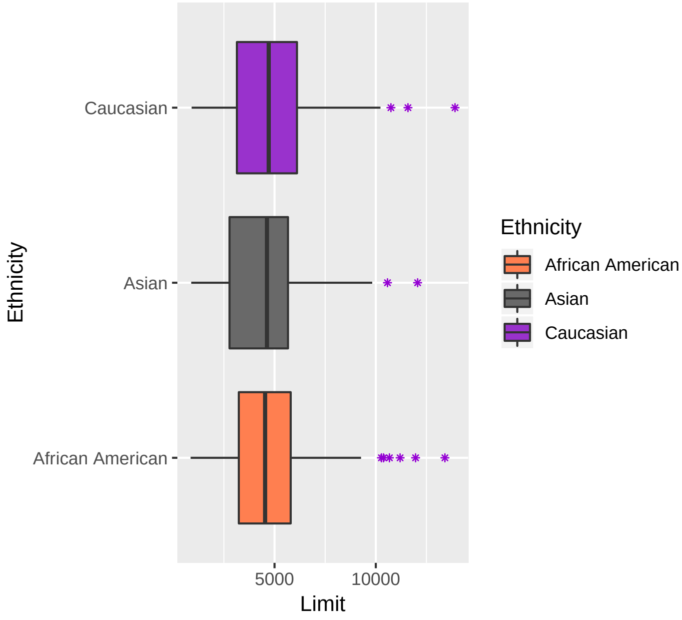

```r
#balance
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Balance, fill = Ethnicity)) +
geom_boxplot(outlier.colour = "darkviolet", outlier.shape = 8,
             outlier.size = 1) +
coord_flip() +
scale_fill_manual(values = c("Asian" = "bisque", "African American" = "cadetblue","Caucasian" = "chartreus
```



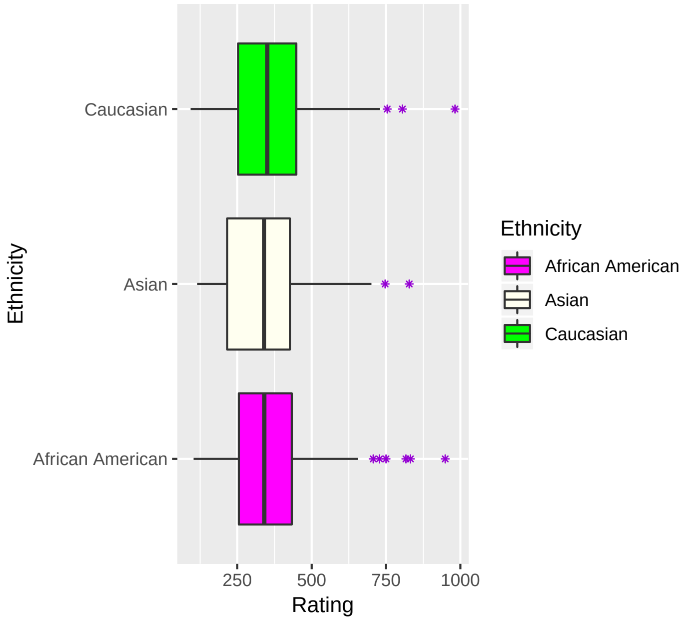
```r
#income
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Income, fill = Ethnicity)) +
geom_boxplot(outlier.colour = "darkviolet", outlier.shape = 8,
             outlier.size = 1) +
coord_flip() +
scale_fill_manual(values = c("khaki", "brown", "dodgerblue"))
```



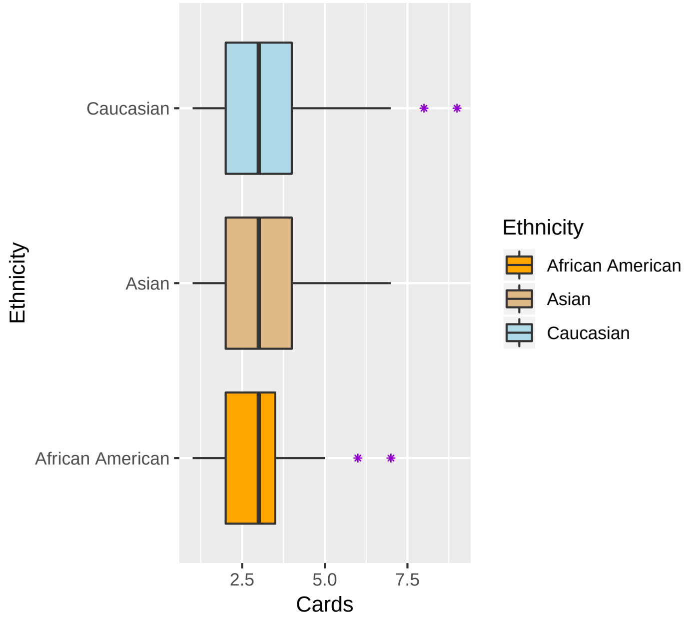

```r
#limit
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Limit, fill = Ethnicity)) +
geom_boxplot(outlier.colour = "darkviolet", outlier.shape = 8,
             outlier.size = 1) +
coord_flip() +
scale_fill_manual(values = c("coral", "dimgrey", "darkorchid"))
```



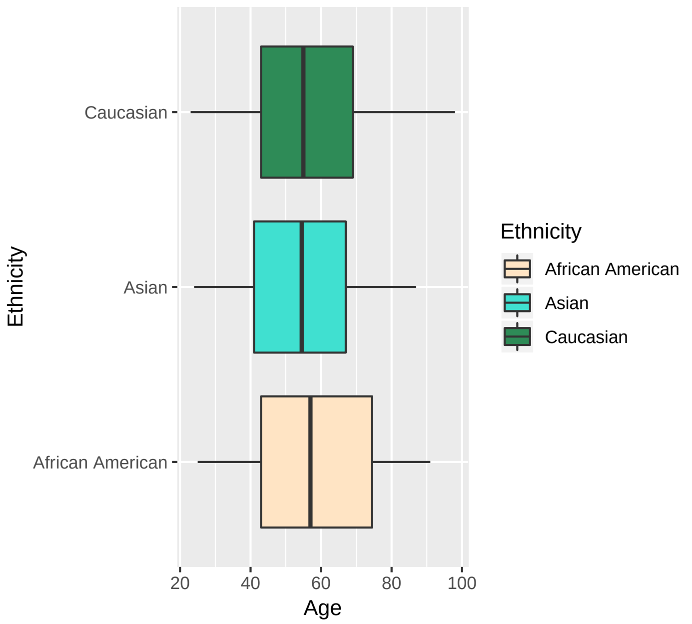

```r
#rating
ggplot(data = credit,
      mapping = aes(x = Ethnicity, y = Rating, fill = Ethnicity)) +
geom_boxplot(outlier.size = 1, outlier.colour = "darkviolet",
             outlier.shape = 8) +
coord_flip() +
scale_fill_manual(values = c("magenta", "ivory", "green"))
```



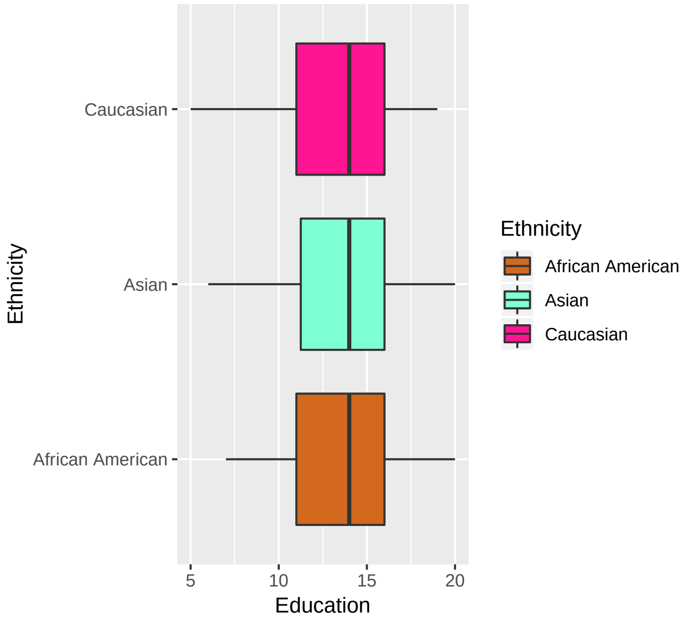
```
#cards
ggplot(data = credit,
    mapping = aes(x = Ethnicity, y = Cards, fill = Ethnicity)) +
geom_boxplot(outlier.size = 1, outlier.colour = "darkviolet",
             outlier.shape = 8) +
coord_flip() +
scale_fill_manual(values = c("orange", "burlywood", "lightblue"))
```



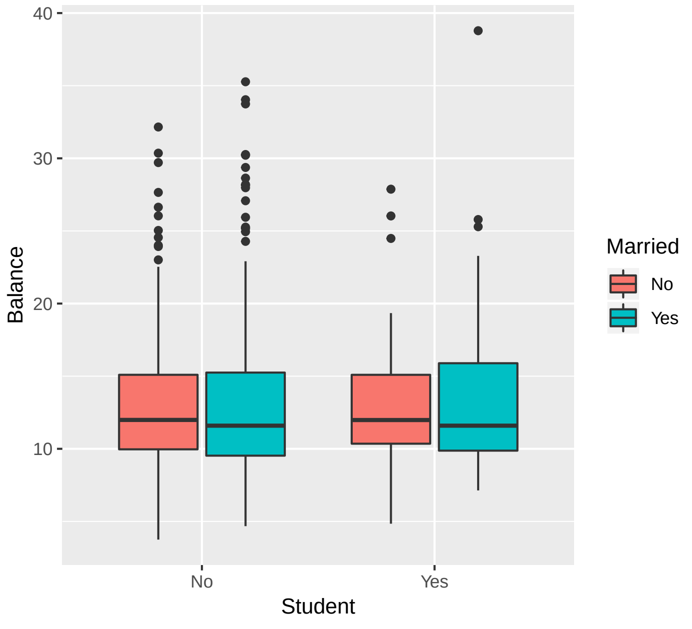

```
#age
ggplot(data = credit,
  mapping = aes(x = Ethnicity, y = Age, fill = Ethnicity)) +
geom_boxplot(outlier.size = 1, outlier.shape = 8,
             outlier.colour = "darkviolet") +
coord_flip() +
scale_fill_manual(values = c("bisque", "turquoise", "seagreen"))
```



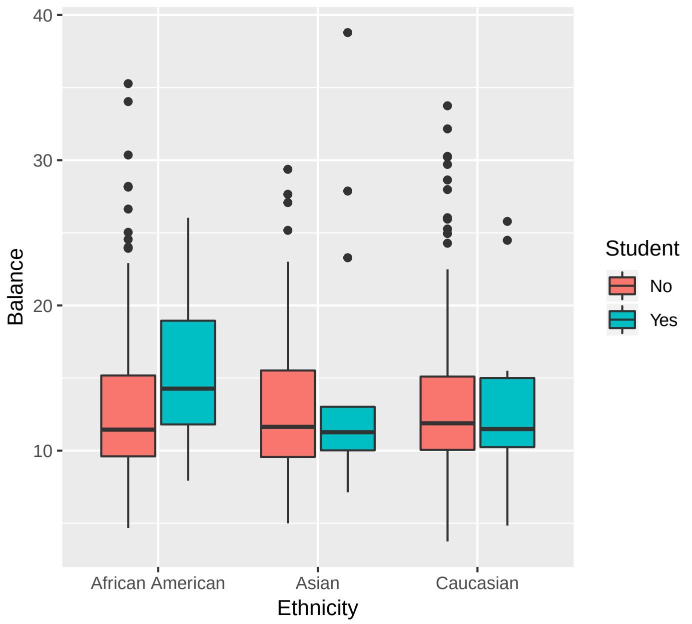

```
#education
ggplot(data = credit,
    mapping = aes(x = Ethnicity, y = Education, fill = Ethnicity)) +
geom_boxplot(outlier.size = 1, outlier.shape = 8,
             outlier.colour = "darkviolet") +
coord_flip() +
scale_fill_manual(values = c("chocolate", "aquamarine", "deeppink"))
```



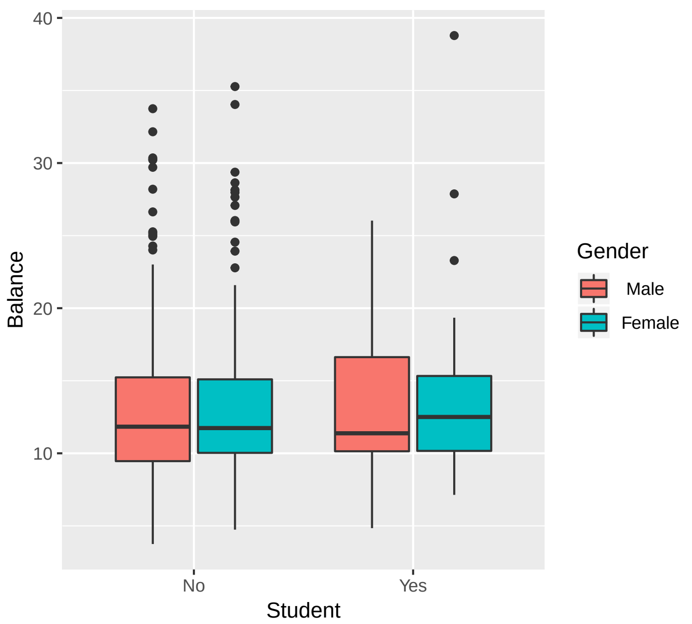
```r
#exploring interaction effects between the categorical variables

#student + married
ggplot(data = credit,
      mapping = aes(x = Student, y = Balance, fill = Married)) +
geom_boxplot()
```




```r
#student + ethnicity
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Balance, fill = Student)) +
geom_boxplot()
```

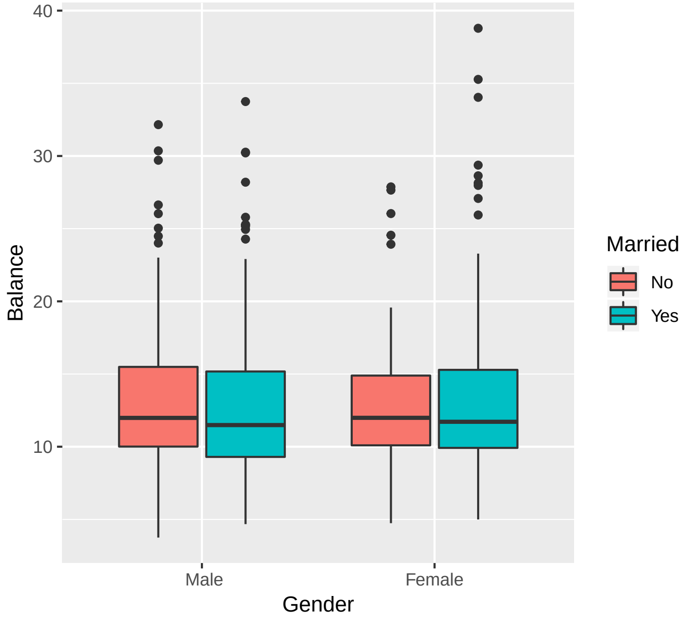



```r
#student + gender
ggplot(data = credit,
       mapping = aes(x = Student, y = Balance, fill = Gender)) +
geom_boxplot()
```

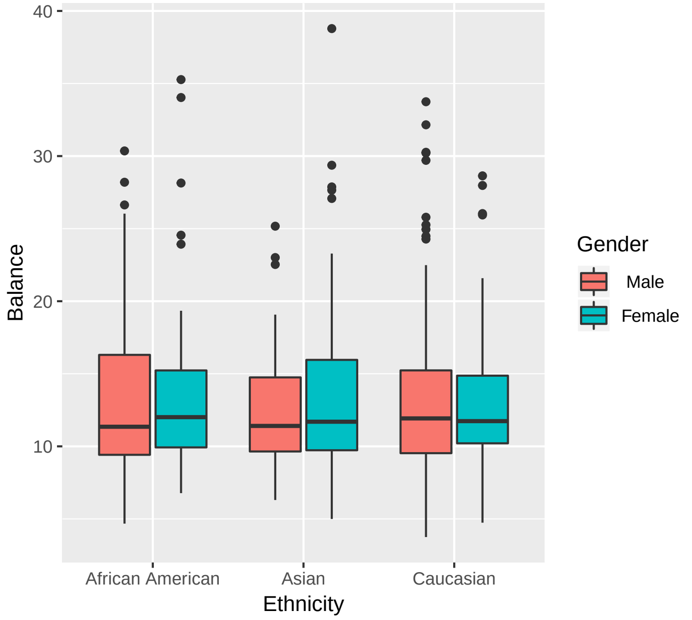



```r
#gender + married
```

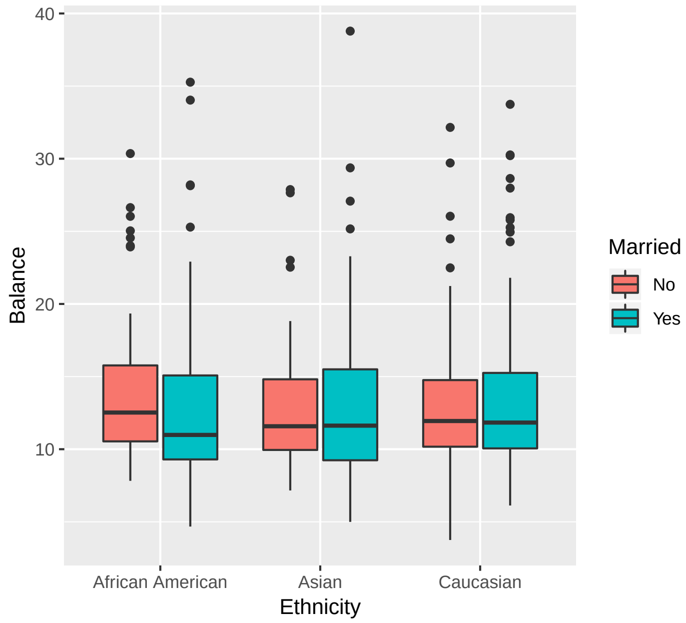
```r
ggplot(data = credit,
       mapping = aes(x = Gender, y = Balance, fill = Married)) +
geom_boxplot()
```




```r
#gender + ethnicity
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Balance, fill = Gender)) +
geom_boxplot()
```



```r
#married + ethnicity
ggplot(data = credit,
       mapping = aes(x = Ethnicity, y = Balance, fill = Married)) +
geom_boxplot()
```



Data Preparation

Data Modelling

Models

Linear Regression

```r
#linear regression model


#this library is for the variance inflation factor function
library(car)
```

```
#the full model
full_model = lm(Balance~., data = credit)


#the dummy variable assignment (categorical variable)


#student
contrasts(Student)


#Ethnicity
contrasts(Ethnicity)


#Gender
contrasts(Gender)


#Married
contrasts(Married)


#full model
full_model


#vif()


#regsubsets
library(leaps)




null_model = lm(Balance~1)
full_model = lm(Balance~., data = credit)
```

```r
step_backward =step(object = null_model, scope = list(lower = null_model, upper = full_model),
     scale = 0, direction = c("forward"), k = 2)


names(step_backward)
names(summary(step_backward))


step_backward$anova
```