Bienvenido Hiyas Jr JC672032

13824819

Tutorial Project – Assessment 2 Total Marks: 34

Objective:

The purpose of this tutorial project is to implement and explore techniques mainly covered in weeks two and three of the Foundations of Data Science subject. This activity will focus on a dataset containing a collection of credit card applications and the subsequent credit approval decisions (positive/successful or negative/unsuccessful). The project is segmented into the following parts:

1. Importing data and handling variable types, variable names and missing values
2. Calculating and visualising proximity measurements
3. Visually exploring data relationships using ggplot2()

1. (a) Importing

The data is publicly available at:

http://archive.ics.uci.edu/ml/machine-learning-databases/credit-screening/crx.data

This dataset is interesting because there is a good mix of attributes: continuous, nominal with small numbers of values, and nominal with larger numbers of values. It contains data regarding corporate MasterCard (credit card) applications from the Commonwealth Bank during 1984. This was a time when credit approvals was done manually, and research into automation was active to improve equity and accuracy into the credit approval process. In the public source where the data is currently available, the variable names have been removed for confidentiality reasons; however, we provide the variable names below, based on the original publication:

Variable Name	**Comments**
Gender	Gender of applicant. Nominal variable with two factor levels
Age	Age of applicant. Numeric variable
MonthlyExpenses	Monthly house hold expenses. Numeric variable. Units of $100
MaritalStatus	Marital status of applicant (Married, Single, Other). Nominal variable, three factor levels
HomeStatus	Home living arrangements of applicant (Renting, Own/Buying, Living with Relatives). Nominal variable, three factor levels
Occupation	Occupation category of applicant. Nominal variable with multiple factor levels
BankingInstitution	Primary banking/credit union institution used by applicant. Nominal variable
YearsEmployed	Number of years the applicant has worked in current or previous employment. Numeric variable
NoPriorDefault	Finical judgements of defaulting on a repayment. Nominal variable with two levels
Employed	Current employment status of applicant. Nominal variable, two factor levels
CreditScore	Offset normalised credit rating score: summary attribute score of tabulated values corresponding to application. Numeric variable

DriversLicense	If the applicant has a current drivers licence. Nominal variable, two factor levels
AccountType	Type of account in primary banking institution, e.g. savings account, etc. Nominal variable
MonthlyIncome	Monthly disposable income. Units of $1. Summary attribute from application. Numeric variable
AccountBalance	Amount in primary account in primary banking institution. Numeric variable
Approved	Approval status of application. Nominal variable, two factor levels

Import the data using the R function **read.table()**. Note: in the data file, missing values are recorded using the "?" character. So, in order to correctly import the missing values in R, you will need to use the input argument **na.strings = "?"** in the call to **read.table()**. Also, note that the data file does not contain the variable names, so you can use the argument **header = FALSE** to instruct R that the first line of the file contains data, rather than variable names.

Table 1 ***Enter*** *your R code you used to import the data here.* Marks (2)

```
getwd()
setwd("~/Documents/JCU/SP1_2020/FoundationsOfDataScience")
read.table("crx.data")
Data = read.table("crx.data", header = FALSE,sep = ",",na.strings = "?")
```

(b)Variable names and types

Because the data file does not contain the variable names, we will need to explicitly set them using the R function **names()**.

Table 2 ***Use*** *this R code to add names to the dataset*

```
names(Data) <- c("Gender", "Age", "MonthlyExpenses", "MaritalStatus", "HomeStatus", "Occupation",
"BankingInstitution", "YearsEmployed", "NoPriorDefault", "Employed", "CreditScore", "DriversLicense",
"AccountType", "MonthlyIncome", "AccountBalance", "Approved")
```

When the data is imported using the function **read.table()**, the variable type is automatically assigned. Data types can also be manually assigned and may need to be re-assigned to perform some types of numerical analysis – such as re-coding a two-level factor into a numeric binary variable, forexample.

Table 3 ***Use*** *the following R code to manually define the variables*

```
Data$Gender <- as.factor(Data$Gender)
Data$Age <- as.numeric(Data$Age)
Data$MonthlyExpenses <- as.integer(Data$MonthlyExpenses)
Data$MaritalStatus <- as.factor(Data$MaritalStatus)
Data$HomeStatus <- as.factor(Data$HomeStatus)
Data$Occupation <- as.factor(Data$Occupation)
Data$BankingInstitution <- as.factor(Data$BankingInstitution)
Data$YearsEmployed <- as.numeric(Data$YearsEmployed)
Data$NoPriorDefault <- as.factor(Data$NoPriorDefault)
Data$Employed <- as.factor(Data$Employed)
Data$CreditScore <- as.numeric(Data$CreditScore)
Data$DriversLicense <- as.factor(Data$DriversLicense)
```

```
Data$AccountType <- as.factor(Data$AccountType)
Data$MonthlyIncome <- as.integer(Data$MonthlyIncome)
Data$AccountBalance <- as.numeric(Data$AccountBalance)
Data$Approved <- as.factor(Data$Approved)
```

Table 4 Variables Gender and DriversLicense are both nominal binary variables, i.e., unordered factors with two levels (values). They don't need to, but they could, be represented as numeric binary variables, taking values 0 and 1. The two values for Gender stand for Male and Female, and the two values for DriversLicense indicate whether or not the individual has a current drivers license. ***Discuss*** *if these variables are better interpreted as symmetric or asymmetric for the sake of credit approval analysis, justifying and/or contextualising your answer.* Marks(4)

Gender is better represented as symmetric
DriversLicense should be Asymmetric

A binary variable is symmetric if both of its states are equally valuable, that is, there is no preference on which outcome should be coded as 1. A binary variable is asymmetric if the outcome of the states is not equally important. Therefore, in our Data, Gender is better represented as symmetric since male and female on credit card application are equally valuable. On the other hand, DriversLicense should be Asymmetric since having a current driver's license has different value or importance than having no current driver's license.

(c) Records with missing values

There are also a few missing values. For this project, observations with missing values need to be removed. To remove observations with missing values, use the R function **na.omit()**.

Table 5 ***Use*** *this R code to remove the records with missing values from the dataset*

```
Data <- na.omit(Data)
```

Table 6 In the original data, ***how many missing values*** *in total are there? Hint: use the function* ***summary()****.* ***How many records*** *are removed by using the function* ***na.omit()****?* Marks(1)

Original Data = 690 obs. Of 16 variables
Gender has 12 NA's,
Age has 12 NA's,
MaritalStatus has 6 NA's,
HomeStatus has 6 NA's,
Ocuupation has 9 NA's,
BankingInstitution has 9 NA's,
MonthlyIncome has 13 NA's,
A total of 67 NA's

Na.omit(Data) = 653 obs. of 16 variables
From Original Data - Na.omit(Data) = 690-653 = **37 rows were removed**

2. Calculating and visualising proximity measurements

*Table 7 The dataset contains variables with mixed types. Use R function **daisy()** from package **cluster** to compute a Gower dissimilarity (distance) matrix between the data records, and refer to the result as "Dist". **Enter** the R code you used, including any libraries needed.* Marks (2)

```
install.packages("cluster")
library("cluster")
Dist = daisy(Data,metric ='gower')
Dist
```

The R object produced from the function **daisy()** is called a dissimilarity object and is efficient in storing information, but is not readily visualised or easy to extract information from. To make the dissimilarity object easier to work with, we can convert it to a matrix.

*Table 8 **Use** the R code to convert the Gower dissimilarity object into a distance matrix*

```
Dist <- as.matrix(Dist)
```

*Table 9 Using the new distance matrix, **what** is the Gower similarity measure between the 10th and the 60th observation (row)? Answer using R command(s).* Marks(2)

```
> Dist[10,60]
[1] 0.3962307
```

Because there are a large number of observations/records (rows) in the dataset, it is typical to visualise the distance matrix to gain insight into data structures.

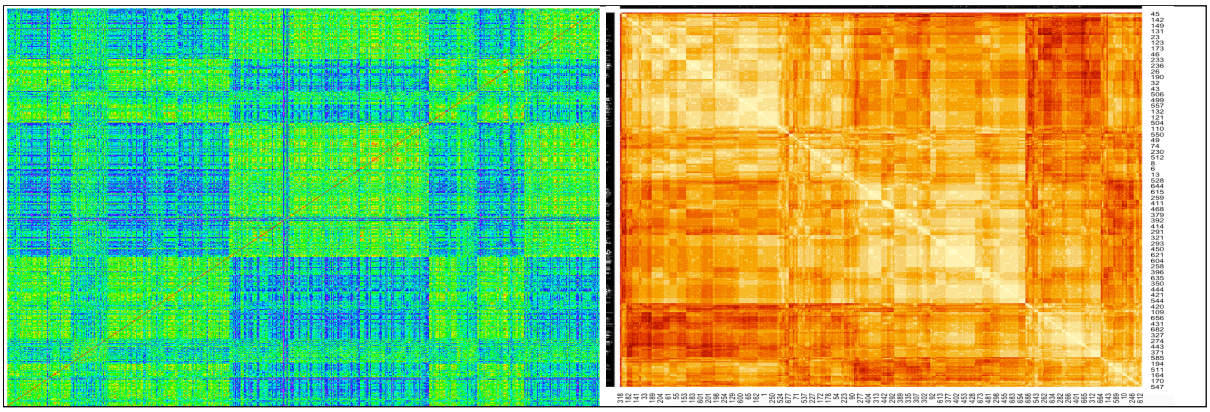
*Table 10 **Use** the following R code to visualise the distance matrix*

```
dim <- ncol(Dist)  # used to define axis in image
image(1:dim, 1:dim, Dist, axes = FALSE, xlab="", ylab="", col = rainbow(100))
```

NOTE (Optional): Additionally, you could also reorder the rows and columns of the matrix according to their similarities before visualising, using a technique called **clustering** (which will be studied as part of other, more advanced subjects, namely, data mining and machine learning): Marks(1)

```
heatmap(Dist, Rowv=TRUE, Colv="Rowv", symm = TRUE)
```

Table 11 ***Insert*** *the image(s) of the distance matrix below, then* ***Describe*** *the pattern you see when visualising it(them).* Marks (1)



On the image on left above, I can see a pattern that the observations on the middle has a color blue which corresponds to values near to 1 meaning they are similar.
Likewise, on the heatmap on the right picture above, we can see a high concentration color Red/Orange in the lower left of the heatmap. This represents the values close to 1 or are Similar. These values are approximately from (480-600) paired with (1-100) observations.

Visualising a distance matrix is one form of initially exploring the dataset. Correlation matrices between numerical data types can also be useful when exploring the data.

Table 12 ***Enter*** *your R code used to calculate the Pearson and then the Spearman correlation matrices using all* ***numerical*** *variables.* Marks(2)

```
DataRequired = c("Age","MonthlyExpenses",
"YearsEmployed","CreditScore","MonthlyIncome","AccountBalance")
Data2 = Data[ ,DataRequired]
cor(Data2, method ="pearson")
cor(Data2, method ="spearman")
```

3. Visually exploring data patterns and relationships

We may have preconceived notions of what to expect in some datasets. In credit card applications, we may hypothesise that approval would be aligned, for example, with account balance, monthly expenses, credit score and/or age.

Table 13 ***Use*** *the ggplot2 library to produce box plots for AccountBalance, MonthlyExpenses, CreditScore and Age segmented by approval (variable "Approved").* ***Insert*** *the R codes and resulting images into the table below.* Marks(4)

Enter your answer here	Enter your answer here

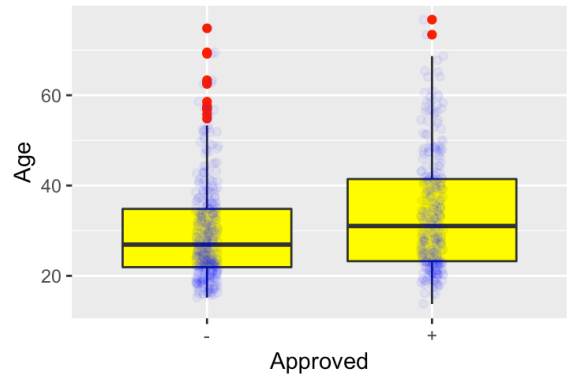
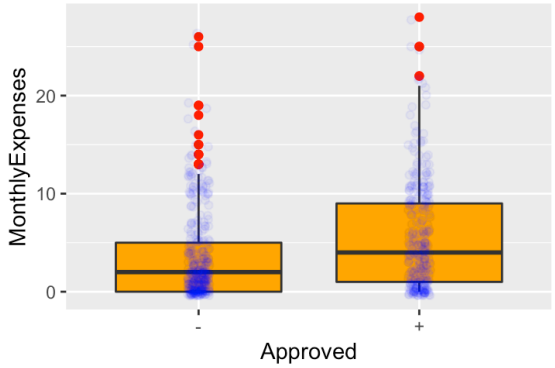
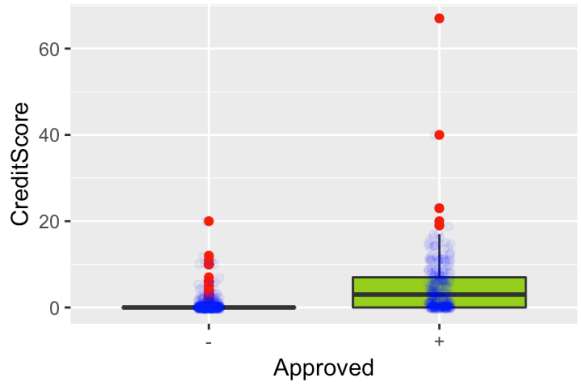
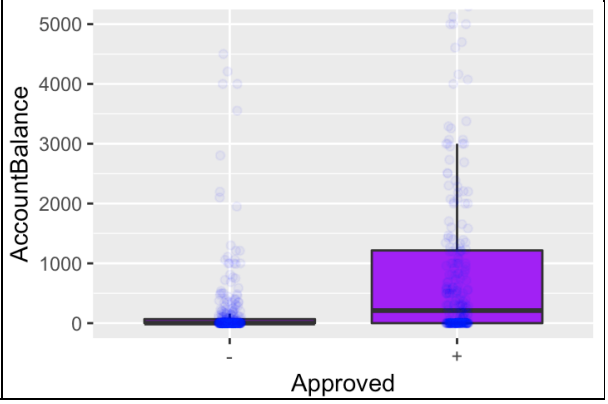
ggplot(Data,map=aes(x=Approved,y=Age))+ geom_boxplot(outlier.color = "red",fill ="Yellow") + geom_jitter(width=0.05, alpha=0.05,color = "Blue") 	ggplot(Data,map=aes(x=Approved,y=MonthlyExpenses))+ geom_boxplot(outlier.color = "red",fill ="Orange") + geom_jitter(width=0.05, alpha=0.05,color = "Blue") 
ggplot(Data,map=aes(x=Approved,y=CreditScore))+ geom_boxplot(outlier.color = "red",fill ="YellowGreen") + geom_jitter(width=0.05, alpha=0.05,color = "Blue") 	ggplot(Data,map=aes(x=Approved,y=AccountBalance))+geom_boxplot(outlier.shape = NA,fill ="Purple") + geom_jitter(width=0.05, alpha=0.05,color="Blue")+ coord_cartesian(ylim=quantile(y=AccountBalance,c(0,5000))) 

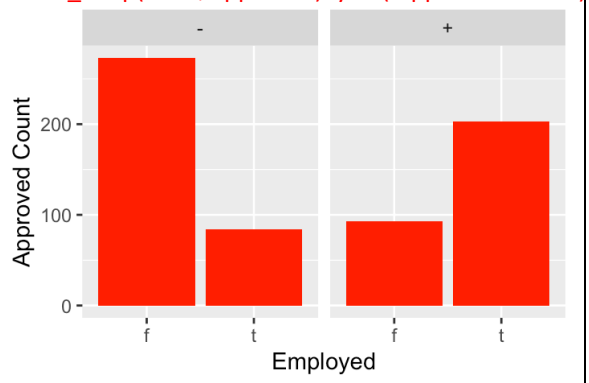
Table 14 ***Describe*** *the apparent patterns shown in the visualisations in Table 13.* Marks (4)

The boxplots above show a strong influence of CreditScore and AccountBalance to the credit card applications. Most of the approved applications has higher count of CreditScore and AccountBalance compared to applications that were not approved. Additonally, Age and MonthlyExpenses also have an effect on the credit card application. Most of the approved applications were around 30 yrs old and higher MonthlyExpenses. (Note: AccountBalance Boxplots outliers has been disregarded to show actual boxplots, else both + and – plots cannot be visualized.

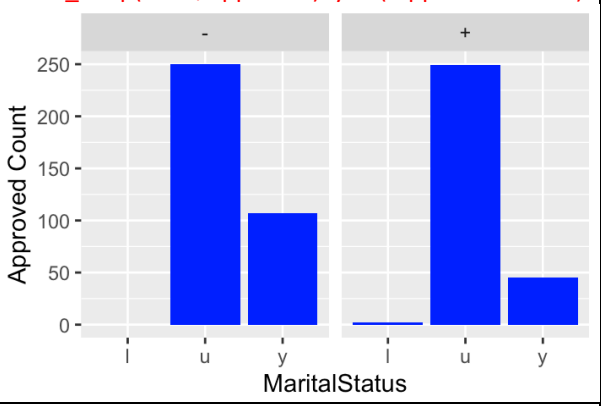
Table 15 ***Use*** *the ggplot2 library to produce bar plots for Employed, MaritalStatus, BankingInstitution, and NoPriorDefault, all segmented by approval (variable "Approved"). **Insert** the R codes and resulting images into the table below.* Marks (4)

Enter your answer here	Enter your answer here

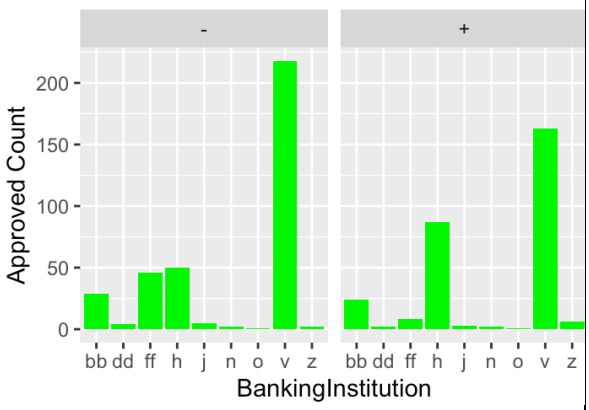
```
ggplot(Data,map=aes(Employed))+geom_bar(fill="Red")+
 facet_wrap(Data$Approved)+ylab('Approved Count')
```




```
ggplot(Data,map=aes(MaritalStatus))+geom_bar(fill="Blue")+
facet_wrap(Data$Approved)+ylab('Approved Count')
```




```
ggplot(Data,map=aes(BankingInstitution))+geom_bar(fill="Green")+
facet_wrap(Data$Approved)+ylab('Approved Count')
```




```
ggplot(Data,map=aes(NoPriorDefault))+geom_bar(fill="Brown")+
facet_wrap(Data$Approved)+ylab('Approved Count')
```

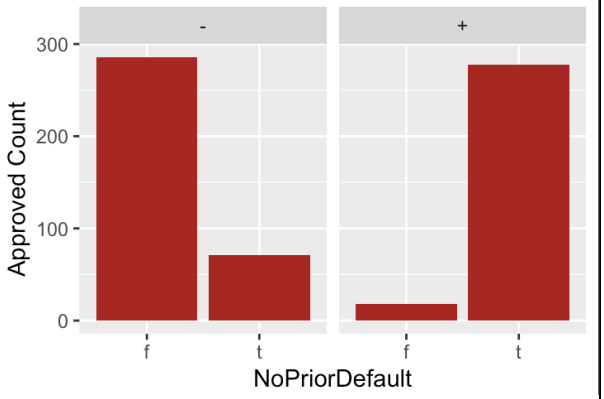


Table 16 Considering that the values "f" (False) and "t" (true) of variables Employed and NoPriorDefault mean unemployed/employed and defaulted-before/never-defaulted-before, respectively, and regardless of the meaning of the values for MaritalStatus and BankingInstitution, ***describe*** *interesting relationships (if any) between these nominal variables and the approval of the application by visually* ***inspecting*** *the bar plots in table 15.* Marks(4)

As shown in the barplots above, NoPriorDefault and Approved Count have a strong influence on the credit card application. Most of the approved application were never-defaulted-before "t" and were Employed(t)). In addition, most of the approved applications as well as those that were not approved have the same MaritalStatus "u" and BangkingInstitution "v". Therefore, the strongest influencing factors in the approval of the applications is if the applicants has a prior default and their employment status.

Apparently, the strongest influencing factor in the approval of the applications is if there has been a prior credit default. By using a contingency table, we can examine the strength of this relationship.

Table 17 ***Use*** *the function* ***table()*** *and* ***calculate*** *the Simple Matching Coefficient (SMC) between NoPriorDefault and Approved, assuming that values "f" (False) and "t" (True) for*

NoPriorDefault are associated with "-" and "+" for Approved, respectively. ***Discuss*** *the interpretation of the SMC in this scenario. Is Jaccard meaningful in this case?* Marks(3)

```
> table(Data$NoPriorDefault,Data$Approved)
    -  +
 f 286  18
 t  71 278
> #using SMC formula
SMC = (286+278)/(286+18+71+278) = 0.863706
#using Jaccard formula
Jaccard = (278)/(18+71+278) = 0.7574932
```

SMC counts both when the values of NoPriorDefault (t) and Approved(+) are both present in the set and when their values are absent in both sets(f and –) making them highly similar. In addition SMC is used for variables with symmetric attributes and Jaccard is use for variables with assymetric attributes.
Since NoPriorDefault and Approved are both asymmetrical Binary variables, Jaccard should be meaningful to use.

Table 18 Further explore this dataset (freely), possibly using other types of plots, such as histograms and scatter plots, and try to obtain further insights and hypotheses. For instance, is there any interesting relationship between monthly disposable income ("MonthlyIncome") and approval ("Approved"), in particular when compared to what you would in principle expect (if anything)? Share your main insights / hypotheses (no more than 1 page). NOT ASSESSED

Basing on the graphs that was plotted below, It seems there is no relationship between MonthyIncome and Approved variables except for an outlier that can be seen on the scatterplot below.