Final Project

Prabh Talwar (300327180)

17/04/2022

Introduction

The Loan Application data set has 21 variables and 1000 observations.The type of variables in this data set are nominal, ordinal and numeric. There is no missing data in this data set. The data set contains the credit information, checking balance, saving balance of the customers, amount of loan they requested, purpose of loan, age, personal status and there are other variables in it. The analysis is conducted to know the type of customers comes to the bank for asking the loan and what is the probability of making the default in the payment by the customers based on the personal information, loan amount provided and the factors that affects the default rate.

Body

To determine our typical type of customers from this data set, Clustering method is being used to group the customers based on their similarity. While examining the data set, it can be said that data need to be cleaned as it has mixed variables both numerical and categorical. To convert our ordinal categorical data into numerical variable, label encoding has been done. And, dummy encoded in the excel as to perform the clustering, our data should have numerical variables. However, it increased the dimensions of the data set. So the method that can be used to cluster the data is the PAM method (Partitioning Around Medoids) using the Gowers distance and silhouette width. Gowers distance is a metrics which finds the distance in the data set in which the variables are numerical and categorical. In order to use PAM method the original data has been used without any cleaning and to know our main types of customers from this data set, selecting the most important variables is necessary. So, the variables been removed are foreign worker as this variables has the near zero variance which is not useful for our analysis, land Line and saving balance.

`daisy` function is a part of the cluster package. This function is used when the data variables in a our data set are not in same format i.e, numeric, nominal and ordinal. `daisy` function returns a distance matrix and K-means cannot be applied on the output of daisy function because K-means cannot cluster the data based on the distance matrix. The two options left are Kmedoids (PAM) and Hierarchical clustering. When used the Hierarchical clustering the dendrogram came out to be cluttered and didn't provided any useful information. `daisy` function uses the `gower`measure to calculate the distance. When any data is presented to the `daisy` function, it looks for the type of data in the data set if it finds the mixed data as our data set we are working on, it automatically selects the `gowers` measure to find the distance and it applies a suitable distance measure considering our data types. For instance, to convert our numerical data manhattan distance is used to calculate the distance and for ordinal data, converts into ranks and then uses the manhattan distance.

Conclusion

So, the output generated tells about the two clusters in which the type of customers falls into cluster 1 has the following characteristics: they are single, their average age is 28, requested amount is `$2284`, purpose of loan is electronics/home entertainment, the duration is 24 months, their credit history says repaid, they are skilled workers and have been employed for 4 - 7 years, their installment rate is quaterly, they have no other debtors, they have 1 loan pending, they have 1 dependent and their checking balance is `< $0`, they live in their own housing, they have been living in the present residence since 2 months and they don't have any installment plan

The type of customers falls into cluster 2 has the following characteristics: they are in common law, their average age is 29, requested amount is `$3959`, purpose of loan is new vehicle, the duration is 15 months, their credit history says repaid, they are skilled workers and have been employed for 1 - 4 years, their installment rate is 3 months, they have no other debtors, they have 1 loan pending, they have 1 dependent and their checking balance is unknown, they live in their own housing, they have been living in the present residence since 2 months, they don't have any installment plan and they have building society savings as a property

Appendix

```r
# Loading the data set
LoanApplicationData <- read.csv("LoanApplicationData.csv", stringsAsFactors = TRUE)
```

Exploratory Data Analysis

Identifying the Variables

The Loan Application data set has 21 variables and 1000 observations.The type of variables in this data set are nominal, ordinal and numeric.

```r
#makeDataReport(LoanApplicationData)
```

checking_balance

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	"unknown"

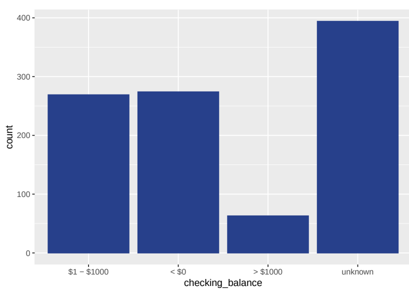
months_loan_duration

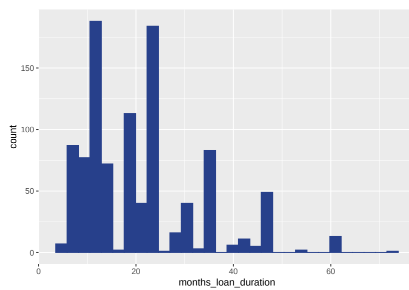
Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	33
Median	18
1st and 3rd quartiles	12; 24
Min. and max.	4; 72

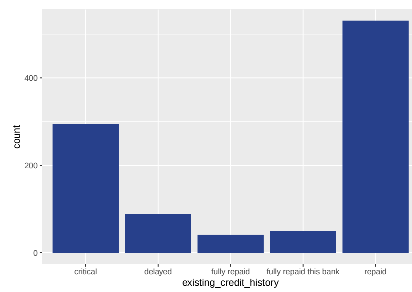
existing__credit__history

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	5
Mode	“repaid”

```
ggplot(LoanApplicationData, aes(x = checking_balance))+
  geom_bar(fill = "royalblue4",color = "royalblue4")

ggplot(LoanApplicationData, aes(x = months_loan_duration))+
  geom_histogram(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = existing_credit_history))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```







The above graph shows the the checking balance for around 400 customers is unknown and for some is less than `$0` and between `$1 - $1000`. The credit history for around 500 customers says to be repaid the loan and for 293 customers it says is critical. The loan duration ranges between 4 months to 72 months.

purpose__of__loan

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	10
Mode	“electronics/home entertainment”

requested__amount

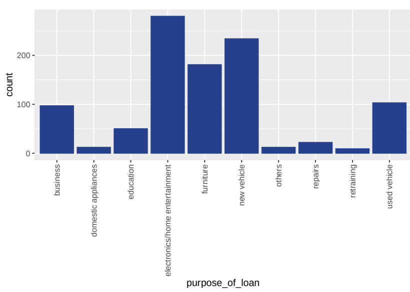
Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	921
Median	2319.5
1st and 3rd quartiles	1365.5; 3972.25

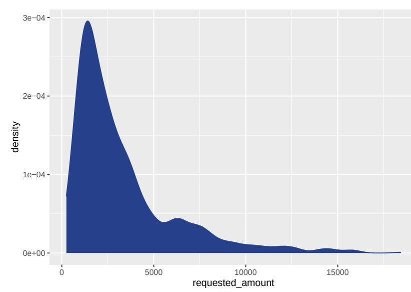
Feature	Result
Min. and max.	250; 18424

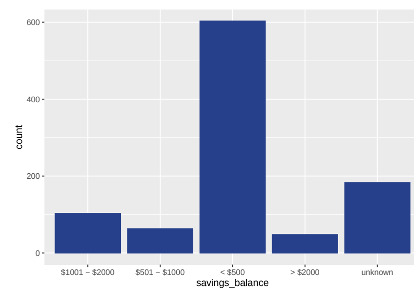
savings_balance

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	5
Mode	“< $500”

```
ggplot(LoanApplicationData, aes(x = purpose_of_loan))+
  geom_bar(fill = "royalblue4",color= "royalblue4")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

ggplot(LoanApplicationData, aes(x = requested_amount))+
  geom_density(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = savings_balance))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```







The above graphs tells that the purpose for loan for most customers is for electronics/home entertainment, new vehicle and furniture. The requested amount for loan ranges between `$250 to $18424` and around 7500 customers’ saving balance is less than $500.

employment_length

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	5
Mode	“1 - 4 yrs”

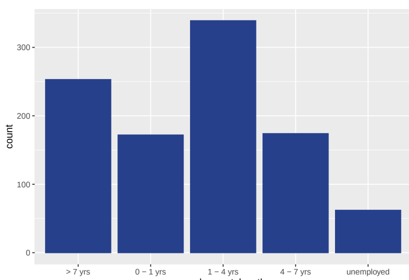
installment_rate

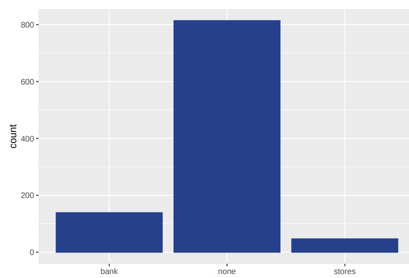
Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“4”
Reference category	1

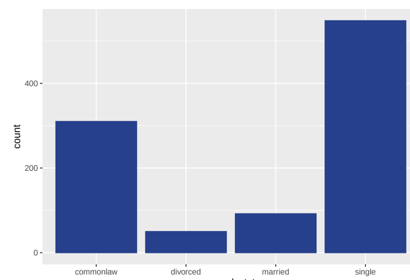
personal_status

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“single”

```
ggplot(LoanApplicationData, aes(x = employment_length))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData,aes(x = installment_plan))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = personal_status))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```







The above graphs shows that 548 number of the bank customers are single and the data says that the customers who are unemployed are 62 and rest of them are all employed.The installment plans are bank and stores, 814 customers don’t have any installment.

other_debtors

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	3
Mode	“none”

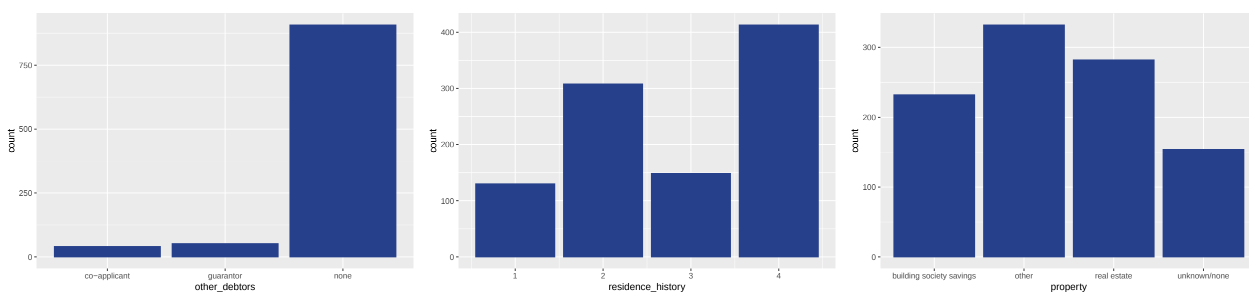
residence_history

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“4”
Reference category	1

property

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“other”

```
ggplot(LoanApplicationData, aes(x = other_debtors))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = residence_history))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = property))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```



More than 800 customers don’t have any other debtors, around 250 customers own a real estate as a property and around 150 customers don’t have any property. The data set says the residence history for the most number od customers is 4 years and 2 years.

age

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	53
Median	33
1st and 3rd quartiles	27; 42

Feature	Result
Min. and max.	19; 75

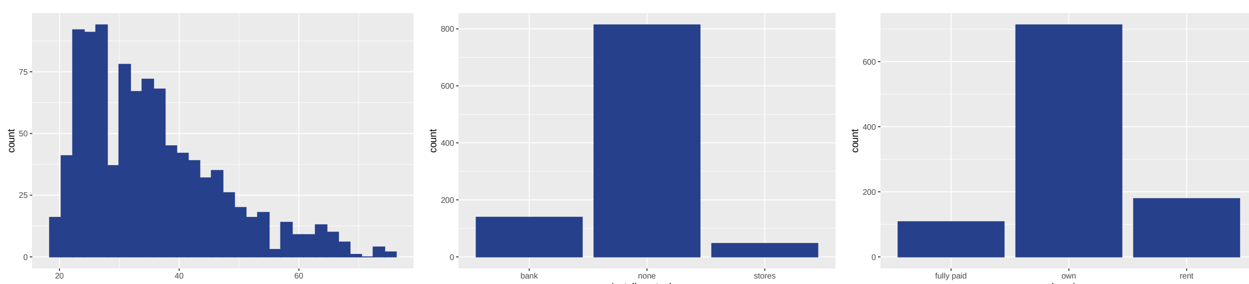
installment_plan

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	3
Mode	“none”

housing

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	3
Mode	“own”

```
ggplot(LoanApplicationData, aes(x = age))+
  geom_histogram(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = installment_plan))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = housing))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```



The age group customers fall into ranges between 19-75 year old. About 800 customers don’t have any installment plans the rest of the customers have bank and stores as their installment plans and around 700 borrowers owns housing which is the highest and around 180 customers have rental housing.

existing_loans

Feature	Result
Variable type	numeric

Feature	Result
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“1”
Reference category	1

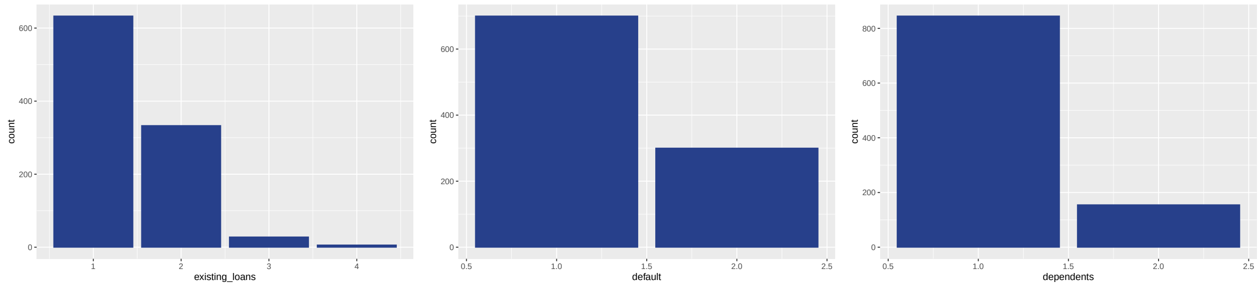
default

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	“1”
Reference category	1

dependents

Feature	Result
Variable type	numeric
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	“1”
Reference category	1

```
ggplot(LoanApplicationData, aes(x = existing_loans))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = default))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = dependents))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```



Most of the customers have only 1 dependent i.e. 845, and 700 customers have made the default and 300 have no default history. 633 customers have 1 existing loan and 333 customers have 2 existing loans.

landline

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	“none”

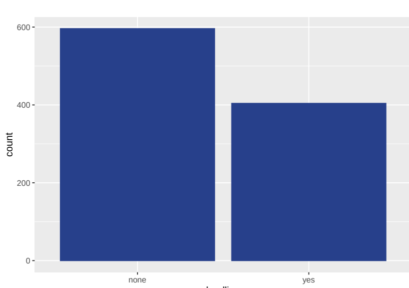
foreign_worker

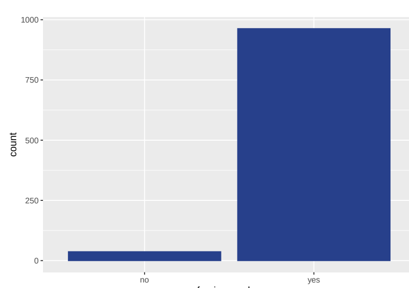
Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	2
Mode	“yes”

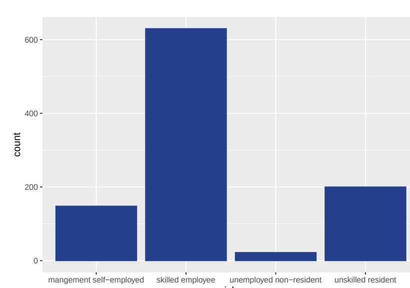
job

Feature	Result
Variable type	character
Number of missing obs.	0 (0 %)
Number of unique values	4
Mode	“skilled employee”

```r
ggplot(LoanApplicationData, aes(x = landline))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = foreign_worker))+
  geom_bar(fill = "royalblue4",color= "royalblue4")

ggplot(LoanApplicationData, aes(x = job))+
  geom_bar(fill = "royalblue4",color= "royalblue4")
```







963 customers are foreign workers in which 22 customers are unemployed, 630 are skilled employees whereas, 200 are unskilled employees.

Clustering

To know about our typical customer, clustering algorithm is used to group borrowers on the bases of their similar characteristics or features. And to cluster customers, the relevant variables have been selected from the data set. Our loan application data set have both numerical and categorical variables.

```
LoanData <- read_csv("LoanData.csv")
```

```
## Rows: 1000 Columns: 41
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): checking_balance, savings_balance, employment_length
## dbl (38): months_loan_duration, existing_credit_history_critical, existing_c...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(LoanData)
```

While examining the data set, it can be said that data need to be cleaned as it has mixed variables both numerical and categorical. To convert our ordinal categorical data into numerical variable, label encoding has been done. And, the above data has been dummy encoded in the excel as to perform the clustering as our data should have numerical variables.The dummy encoding has been done to convert our nominal data variables. The variables that have been treated ordinal are `checking_balance`, `savings_balance` and `employment_length`.

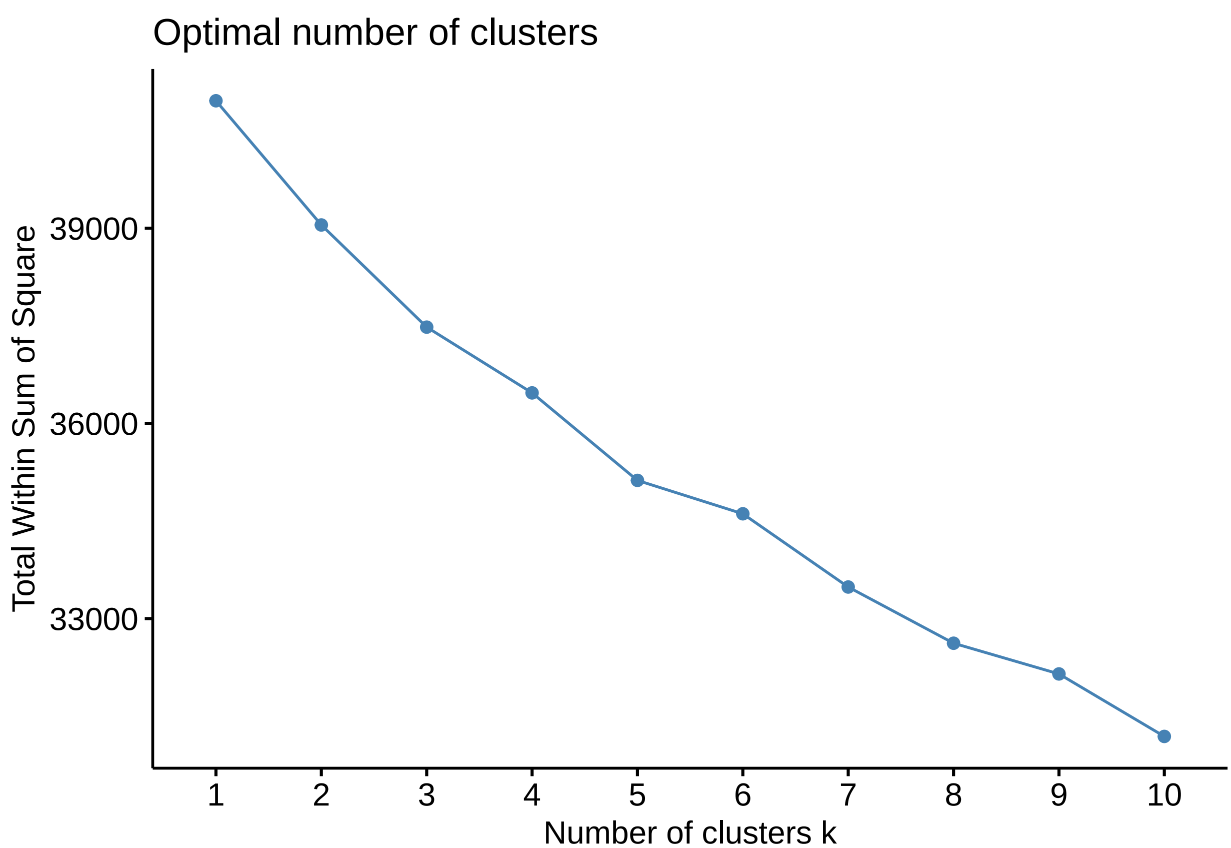
```
#Label encoding ordinal data variables

LoanData$checking_balance <- factor(LoanData$checking_balance,
                                   levels = c("< $0",
                                              "$1 - $1000",
                                              "> $1000",
                                              "unknown"),
                                   ordered = TRUE)

LoanData$checking_balance <- as.numeric(LoanData$checking_balance)

LoanData$savings_balance <- factor(LoanData$savings_balance,
                                   levels = c("< $500",
                                              "$501 - $1000",
                                              "$1001 - $2000",
                                              "> $2000",
                                              "unknown"),
                                   ordered = TRUE)

LoanData$savings_balance <- as.numeric(LoanData$savings_balance)

LoanData$employment_length <- factor(LoanData$employment_length,
                                    levels = c("unemployed",
                                               "0 - 1 yrs",
```

```
                                                  "1 - 4 yrs",
                                                  "4 - 7 yrs",
                                                  "> 7 yrs"),
                                        ordered = TRUE)

LoanData$employment_length <- as.numeric(LoanData$employment_length)

#View(LoanData)
```

```
# Scaling the data

Scaled_Loan_Data <- scale(LoanData)
```

As the data has been cleaned and the variables are now converted into numeric variables, clustering has been performed and graphed the elbow plot to determine the number of clusters. The method used for clustering is K-means, here clusters are represented by its center i.e, centroid.

```
# Estimating the optimal number of clusters

fviz_nbclust(Scaled_Loan_Data, kmeans, method = "wss")
```



As we can see, the elbow plot failed to show the number of clusters to use in the clustering, as in our data the number of dimensions are large , so the other method that can be used to cluster the data is the PAM method (Partitioning Around Medoids) using the Gowers distance and silhouette width. Gowers distance is a metrics which finds the distance in the data set in which the variables are numerical and categorical. In order to use PAM method the original data has been used without any cleaning.

In order to know our main types of customers from this data set, selecting the most important variables is necessary. So, the variables been removed are foreign worker as this variables has the near zero variance which is not useful for our analysis, land Line and saving balance.

```
LoanApplication <- LoanApplicationData %>%
  select(checking_balance, months_loan_duration, existing_credit_history,
         purpose_of_loan, requested_amount, employment_length,
         installment_rate, personal_status, other_debtors, residence_history,
         property, age, installment_plan, housing, existing_loans, default,
         dependents, job)
```

```
#Label encoding ordinal data variables

LoanApplication$checking_balance <- factor(LoanApplication$checking_balance,
                                   levels = c("< $0",
                                              "$1 - $1000",
                                              "> $1000",
                                              "unknown"),
                                   ordered = TRUE)


LoanApplication$employment_length <- factor(LoanApplication$employment_length,
                                    levels = c("unemployed",
                                               "0 - 1 yrs",
                                               "1 - 4 yrs",
                                               "4 - 7 yrs",
                                               "> 7 yrs"),
                                    ordered = TRUE)
```

```
LoanApplication$default <- as.factor(LoanApplication$default)
```

```
str(LoanApplication)
```

```
## 'data.frame':    1000 obs. of  18 variables:
##  $ checking_balance       : Ord.factor w/ 4 levels "< $0"<"$1 - $1000"<..: 1 2 4 1 1 4 4 2 4 2 ...
##  $ months_loan_duration   : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ existing_credit_history: Factor w/ 5 levels "critical","delayed",..: 1 5 1 5 2 5 5 5 5 1 ...
##  $ purpose_of_loan        : Factor w/ 10 levels "business","domestic appliances",..: 4 4 3 5 6 3 5 1
##  $ requested_amount       : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ employment_length      : Ord.factor w/ 5 levels "unemployed"<"0 - 1 yrs"<..: 5 3 4 4 3 3 5 3 4 1
##  $ installment_rate       : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status        : Factor w/ 4 levels "commonlaw","divorced",..: 4 1 4 4 4 4 4 4 2 3 ...
##  $ other_debtors          : Factor w/ 3 levels "co-applicant",..: 3 3 3 2 3 3 3 3 3 3 ...
##  $ residence_history      : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property               : Factor w/ 4 levels "building society savings",..: 3 3 3 1 4 4 1 2 3 2 ..
##  $ age                    : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ installment_plan       : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ housing                : Factor w/ 3 levels "fully paid","own",..: 2 2 2 1 1 1 2 3 2 2 ...
##  $ existing_loans         : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ default                : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1 2 ...
##  $ dependents             : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ job                    : Factor w/ 4 levels "mangement self-employed",..: 2 2 4 2 2 4 2 1 4 1 ...
```

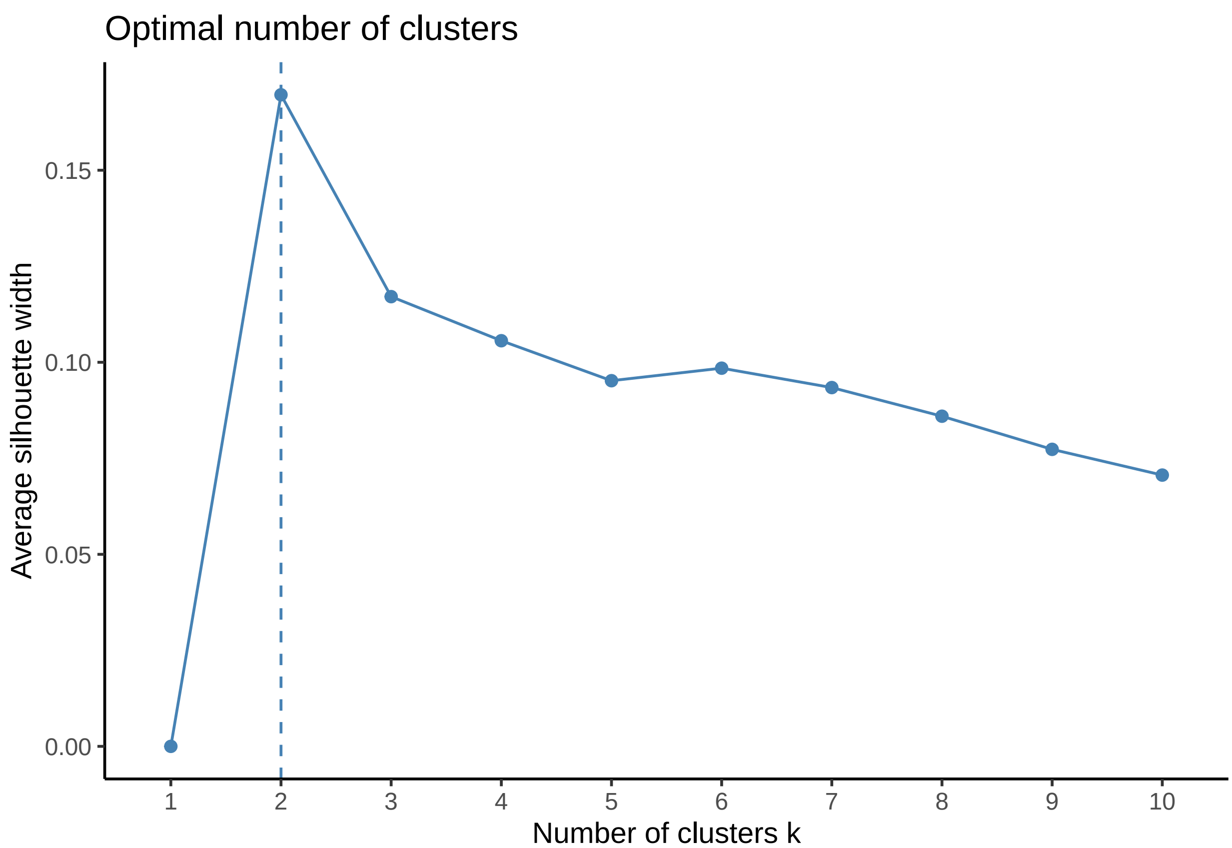
```r
gower_loan <- daisy(LoanApplication,
                    metric = "gower" ,
                    type = list(logratio = 2))

summary(gower_loan)
```

```
## 499500 dissimilarities, summarized :
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001055 0.335030 0.406130 0.405560 0.476610 0.821770
## Metric :  mixed ;  Types = O, I, N, N, I, O, I, N, N, I, N, I, N, N, I, N, I, N
## Number of objects : 1000
```

`daisy` function is a part of the cluster package. This function is used when the data variables in a our data set are not in same format i.e, numeric, nominal and ordinal. `daisy` function returns a distance matrix and K-means cannot be applied on the output of daisy function because K-means cannot cluster the data based on the distance matrix. The two options left are Kmedoids (PAM) and Hierarchical clustering. When used the Hierarchical clustering the dendrogram came out to be cluttered and didn't provided any useful information.

`daisy` function uses the `gower`measure to calculate the distance. When any data is presented to the `daisy` function, it looks for the type of data in the data set if it finds the mixed data as our data set we are working on, it automatically selects the `gowers` measure to find the distance and it applies a suitable distance measure considering our data types. For instance, to convert our numerical data manhattan distance is used to calculate the distance and for ordinal data, converts into ranks and then uses the manhattan distance.

```r
fviz_nbclust(as.matrix(gower_loan), pam, method = "silhouette") +
theme_classic()
```



From the above plot, it can be said that the optimal number of clusters are two and we can classify the observations into two clusters.

```
clusters = pam(as.matrix(gower_loan), # Converting Gower dissimilarity into a distance matrix
               diss = TRUE,
               stand = FALSE,
               k = 2)
```

```
LoanApplication[clusters$medoids, ]
```

```
##     checking_balance months_loan_duration existing_credit_history
## 695          unknown                   24                  repaid
## 859             < $0                   15                  repaid
##                   purpose_of_loan requested_amount employment_length
## 695 electronics/home entertainment             2284         4 - 7 yrs
## 859                   new vehicle             3959         1 - 4 yrs
##     installment_rate personal_status other_debtors residence_history
## 695                4          single          none                 2
## 859                3       commonlaw          none                 2
##                    property age installment_plan housing existing_loans
## 695                   other  28             none     own              1
## 859 building society savings  29             none     own              1
##     default dependents              job
## 695       1          1 skilled employee
## 859       2          1 skilled employee
```

So, the output generated tells about the two clusters in which the type of customers falls into cluster 1 has the following characteristics: they are single, their average age is 28, requested amount is $2284, purpose of loan is electronics/home entertainment, the duration is 24 months, their credit history says repaid, they are skilled workers and have been employed for 4 - 7 years, their installment rate is quaterly, they have no other debtors, they have 1 loan pending, they have 1 dependent and their checking balance is < $0, they live in their own housing, they have been living in the present residence since 2 months and they don't have any installment plan

The type of customers falls into cluster 2 has the following characteristics: they are in common law, their average age is 29, requested amount is $3959, purpose of loan is new vehicle, the duration is 15 months, their credit history says repaid, they are skilled workers and have been employed for 1 - 4 years, their installment rate is 3 months, they have no other debtors, they have 1 loan pending, they have 1 dependent and their checking balance is unknown, they live in their own housing, they have been living in the present residence since 2 months, they don't have any installment plan and they have building society savings as a property

Logistic Regression

To estimate the probability of default, we will be using logistic regression.

```
# Loading the data set
LoanApplicationData <- read.csv("LoanApplicationData.csv", stringsAsFactors = TRUE)
```

```
nearZeroVar(LoanApplicationData, saveMetrics= TRUE) %>%
rownames_to_column() %>%
filter(nzv)
```

```
##          rowname freqRatio percentUnique zeroVar  nzv
## 1 foreign_worker  26.02703           0.2   FALSE TRUE
```

As the foreign worker has near zero variance, it has been removed from the data set, as it does not provides any useful information to a model.

```
# Removing the foreign worker variable
LoanApplicationData <- LoanApplicationData[-20]
head(LoanApplicationData)
```

```
##   checking_balance months_loan_duration existing_credit_history
## 1             < $0                    6                critical
## 2       $1 - $1000                   48                  repaid
## 3          unknown                   12                critical
## 4             < $0                   42                  repaid
## 5             < $0                   24                 delayed
## 6          unknown                   36                  repaid
##                    purpose_of_loan requested_amount savings_balance
## 1 electronics/home entertainment             1169         unknown
## 2 electronics/home entertainment             5951          < $500
## 3                      education             2096          < $500
## 4                      furniture             7882          < $500
## 5                    new vehicle             4870          < $500
## 6                      education             9055         unknown
##   employment_length installment_rate personal_status other_debtors
## 1           > 7 yrs                4          single          none
```

```
## 2         1 - 4 yrs                2       commonlaw          none
## 3         4 - 7 yrs                2          single          none
## 4         4 - 7 yrs                2          single     guarantor
## 5         1 - 4 yrs                3          single          none
## 6         1 - 4 yrs                2          single          none
##   residence_history                 property age installment_plan    housing
## 1                 4              real estate  67             none        own
## 2                 2              real estate  22             none        own
## 3                 3              real estate  49             none        own
## 4                 4 building society savings  45             none fully paid
## 5                 4             unknown/none  53             none fully paid
## 6                 4             unknown/none  35             none fully paid
##   existing_loans default dependents landline                job
## 1              2       1          1      yes   skilled employee
## 2              1       2          1     none   skilled employee
## 3              1       1          2     none unskilled resident
## 4              1       1          2     none   skilled employee
## 5              2       2          2     none   skilled employee
## 6              1       1          2      yes unskilled resident
```

```
#Label encoding ordinal data variables

LoanApplicationData$checking_balance <- factor(
  LoanApplicationData$checking_balance,
  levels = c("< $0",
             "$1 - $1000",
             "> $1000",
             "unknown"),
  ordered = TRUE)

LoanApplicationData$checking_balance <- as.numeric(
  LoanApplicationData$checking_balance)

LoanApplicationData$savings_balance <- factor(
  LoanApplicationData$savings_balance,
  levels = c("< $500",
             "$501 - $1000",
             "$1001 - $2000",
             "> $2000",
             "unknown"),
  ordered = TRUE)

LoanApplicationData$savings_balance <- as.numeric(
  LoanApplicationData$savings_balance)




LoanApplicationData$employment_length <- factor(
  LoanApplicationData$employment_length,
  levels = c("unemployed",
             "0 - 1 yrs",
             "1 - 4 yrs",
             "4 - 7 yrs",
```

```r
             "> 7 yrs"),
  ordered = TRUE)

LoanApplicationData$employment_length <- as.numeric(
  LoanApplicationData$employment_length)
```

```r
# Dummy encoding nominal data variables

LoanApplicationData <- dummy_cols(LoanApplicationData,
                                  select_columns = c("existing_credit_history",
                                                     "purpose_of_loan",
                                                     "personal_status",
                                                     "other_debtors",
                                                     "property",
                                                    "installment_plan",
                                                    "housing",
                                                    "landline",
                                                    "job"),
                                  remove_first_dummy =TRUE ,
                                  remove_selected_columns = TRUE)
```

```r
# scaling the data
scaled_data <- scale(LoanApplicationData[-10]) # removing the response variable

LoanApplicationData2 <- cbind(scaled_data ,
                              default = LoanApplicationData$default)

LoanApplicationData2 <- as.data.frame(LoanApplicationData2)
```

```r
# Re code class to 1 = No, 2 = Yes
LoanApplicationData2$default[LoanApplicationData2$default == 1 ] <- "No"
LoanApplicationData2$default[LoanApplicationData$default == 2 ] <- "Yes"

LoanApplicationData2$default <- as.factor(LoanApplicationData2$default)
```

```r
set.seed(123) #for reproducibility

#splitting the data
loan_split <- initial_split(LoanApplicationData2, prop = 0.8)

loan_train <- training(loan_split)
loan_test <- testing(loan_split)
```

```r
# Logistic Regression Model

model <- glm(default ~ . , data = loan_train,  family = "binomial")

summary(model)
```

```
##
## Call:
## glm(formula = default ~ ., family = "binomial", data = loan_train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2823  -0.6926  -0.4011   0.6486   2.6909
##
## Coefficients:
##                                                  Estimate Std. Error z value
## (Intercept)                                     -1.271368   0.107073 -11.874
## checking_balance                                -0.704568   0.104098  -6.768
## months_loan_duration                             0.341569   0.121658   2.808
## requested_amount                                 0.398524   0.137223   2.904
## savings_balance                                 -0.421979   0.110562  -3.817
## employment_length                               -0.173387   0.108293  -1.601
## installment_rate                                 0.415354   0.109126   3.806
## residence_history                                0.005754   0.104741   0.055
## age                                             -0.109540   0.111179  -0.985
## existing_loans                                   0.119872   0.126393   0.948
## dependents                                       0.184911   0.101805   1.816
## existing_credit_history_delayed                  0.182763   0.107653   1.698
## `existing_credit_history_fully repaid`           0.292953   0.098601   2.971
## `existing_credit_history_fully repaid this bank` 0.339283   0.105482   3.217
## existing_credit_history_repaid                   0.341574   0.144789   2.359
## `purpose_of_loan_domestic appliances`            0.122464   0.097480   1.256
## purpose_of_loan_education                        0.175417   0.114092   1.538
## `purpose_of_loan_electronics/home entertainment` 0.059649   0.168976   0.353
## purpose_of_loan_furniture                        0.074375   0.152136   0.489
## `purpose_of_loan_new vehicle`                    0.377722   0.159202   2.373
## purpose_of_loan_others                          -0.055796   0.109403  -0.510
## purpose_of_loan_repairs                          0.137046   0.098164   1.396
## purpose_of_loan_retraining                      -0.082626   0.131005  -0.631
## `purpose_of_loan_used vehicle`                  -0.232948   0.150612  -1.547
## personal_status_divorced                         0.078637   0.091980   0.855
## personal_status_married                          0.047408   0.096351   0.492
## personal_status_single                          -0.357332   0.116620  -3.064
## other_debtors_guarantor                         -0.325426   0.145566  -2.236
## other_debtors_none                              -0.102373   0.137649  -0.744
## property_other                                   0.084546   0.122262   0.692
## `property_real estate`                          -0.074596   0.128243  -0.582
## `property_unknown/none`                          0.217747   0.185624   1.173
## installment_plan_none                           -0.154169   0.104466  -1.476
## installment_plan_stores                         -0.068751   0.102427  -0.671
## housing_own                                      0.004778   0.244148   0.020
## housing_rent                                     0.149338   0.215081   0.694
## landline_yes                                    -0.147136   0.110344  -1.333
## `job_skilled employee`                           0.132451   0.148770   0.890
## `job_unemployed non-resident`                   -0.023781   0.102985  -0.231
## `job_unskilled resident`                         0.095981   0.152440   0.630
##                                                 Pr(>|z|)
## (Intercept)                                      < 2e-16 ***
## checking_balance                                 1.3e-11 ***
## months_loan_duration                            0.004991 **
## requested_amount                                0.003682 **
## savings_balance                                 0.000135 ***
## employment_length                               0.109356
```

```
## installment_rate                                 0.000141 ***
## residence_history                                0.956193
## age                                              0.324498
## existing_loans                                   0.342922
## dependents                                       0.069320 .
## existing_credit_history_delayed                  0.089565 .
## `existing_credit_history_fully repaid`           0.002967 **
## `existing_credit_history_fully repaid this bank` 0.001298 **
## existing_credit_history_repaid                   0.018319 *
## `purpose_of_loan_domestic appliances`            0.209010
## purpose_of_loan_education                        0.124170
## `purpose_of_loan_electronics/home entertainment` 0.724085
## purpose_of_loan_furniture                        0.624933
## `purpose_of_loan_new vehicle`                    0.017663 *
## purpose_of_loan_others                           0.610051
## purpose_of_loan_repairs                          0.162685
## purpose_of_loan_retraining                       0.528231
## `purpose_of_loan_used vehicle`                   0.121941
## personal_status_divorced                         0.392587
## personal_status_married                          0.622700
## personal_status_single                           0.002183 **
## other_debtors_guarantor                          0.025379 *
## other_debtors_none                               0.457044
## property_other                                   0.489240
## `property_real estate`                           0.560784
## `property_unknown/none`                          0.240773
## installment_plan_none                            0.140002
## installment_plan_stores                          0.502083
## housing_own                                      0.984386
## housing_rent                                     0.487471
## landline_yes                                     0.182388
## `job_skilled employee`                           0.373300
## `job_unemployed non-resident`                    0.817382
## `job_unskilled resident`                         0.528935
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 959.84  on 799  degrees of freedom
## Residual deviance: 719.74  on 760  degrees of freedom
## AIC: 799.74
##
## Number of Fisher Scoring iterations: 5
```

From the above model, it seems not all variables are statistically significant, lets build another model with just the variables that are statistically significant.

```
model1 <- glm(default ~ checking_balance +
                months_loan_duration +
                requested_amount +
                savings_balance +
                installment_rate +
                existing_credit_history_delayed +
```

```r
                `existing_credit_history_fully repaid`+
                `existing_credit_history_fully repaid this bank` +
                existing_credit_history_repaid +
                `purpose_of_loan_domestic appliances` +
                purpose_of_loan_education +
                `purpose_of_loan_electronics/home entertainment`+
                purpose_of_loan_furniture +
                `purpose_of_loan_new vehicle`+
                purpose_of_loan_others +
                purpose_of_loan_repairs +
                purpose_of_loan_retraining +
                `purpose_of_loan_used vehicle`+
                personal_status_divorced +
                personal_status_married +
                personal_status_single +
                other_debtors_guarantor+
                other_debtors_none,
                data = loan_train, family = "binomial")

summary(model1)
```

```
##
## Call:
## glm(formula = default ~ checking_balance + months_loan_duration +
##     requested_amount + savings_balance + installment_rate + existing_credit_history_delayed +
##     `existing_credit_history_fully repaid` + `existing_credit_history_fully repaid this bank` +
##     existing_credit_history_repaid + `purpose_of_loan_domestic appliances` +
##     purpose_of_loan_education + `purpose_of_loan_electronics/home entertainment` +
##     purpose_of_loan_furniture + `purpose_of_loan_new vehicle` +
##     purpose_of_loan_others + purpose_of_loan_repairs + purpose_of_loan_retraining +
##     `purpose_of_loan_used vehicle` + personal_status_divorced +
##     personal_status_married + personal_status_single + other_debtors_guarantor +
##     other_debtors_none, family = "binomial", data = loan_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2916  -0.7058  -0.4224   0.7195   2.7040
##
## Coefficients:
##                                                   Estimate Std. Error z value
## (Intercept)                                       -1.23260    0.10311 -11.954
## checking_balance                                  -0.73074    0.10144  -7.204
## months_loan_duration                               0.37090    0.11628   3.190
## requested_amount                                   0.33998    0.12522   2.715
## savings_balance                                   -0.40577    0.10653  -3.809
## installment_rate                                   0.35145    0.10283   3.418
## existing_credit_history_delayed                    0.18994    0.10573   1.796
## `existing_credit_history_fully repaid`             0.31972    0.09584   3.336
## `existing_credit_history_fully repaid this bank`  0.37078    0.09113   4.069
## existing_credit_history_repaid                     0.31770    0.11483   2.767
## `purpose_of_loan_domestic appliances`              0.11304    0.09477   1.193
## purpose_of_loan_education                          0.23111    0.10899   2.120
## `purpose_of_loan_electronics/home entertainment`  0.05159    0.16417   0.314
```

```
## purpose_of_loan_furniture                          0.07874    0.14609   0.539
## `purpose_of_loan_new vehicle`                      0.38913    0.15438   2.521
## purpose_of_loan_others                            -0.06311    0.10735  -0.588
## purpose_of_loan_repairs                            0.14466    0.09605   1.506
## purpose_of_loan_retraining                        -0.09014    0.12530  -0.719
## `purpose_of_loan_used vehicle`                    -0.18680    0.14272  -1.309
## personal_status_divorced                           0.05580    0.08732   0.639
## personal_status_married                            0.03405    0.09368   0.364
## personal_status_single                            -0.33183    0.10590  -3.134
## other_debtors_guarantor                           -0.34240    0.14145  -2.421
## other_debtors_none                                -0.12697    0.13596  -0.934
##                                                   Pr(>|z|)
## (Intercept)                                        < 2e-16 ***
## checking_balance                                  5.86e-13 ***
## months_loan_duration                              0.001424 **
## requested_amount                                  0.006628 **
## savings_balance                                   0.000140 ***
## installment_rate                                  0.000631 ***
## existing_credit_history_delayed                   0.072431 .
## `existing_credit_history_fully repaid`            0.000850 ***
## `existing_credit_history_fully repaid this bank` 4.73e-05 ***
## existing_credit_history_repaid                    0.005661 **
## `purpose_of_loan_domestic appliances`             0.232947
## purpose_of_loan_education                         0.033967 *
## `purpose_of_loan_electronics/home entertainment` 0.753328
## purpose_of_loan_furniture                         0.589898
## `purpose_of_loan_new vehicle`                     0.011717 *
## purpose_of_loan_others                            0.556605
## purpose_of_loan_repairs                           0.132032
## purpose_of_loan_retraining                        0.471922
## `purpose_of_loan_used vehicle`                    0.190590
## personal_status_divorced                          0.522833
## personal_status_married                           0.716212
## personal_status_single                            0.001727 **
## other_debtors_guarantor                           0.015494 *
## other_debtors_none                                0.350370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 959.84  on 799  degrees of freedom
## Residual deviance: 739.90  on 776  degrees of freedom
## AIC: 787.9
##
## Number of Fisher Scoring iterations: 5
```

```r
#Prediction

prediction <- predict(model1, loan_train, type = "response")

head(prediction)
```

```
##       415       463       179       526       195       938
```

```
## 0.5604768 0.3734891 0.0955121 0.2158853 0.1740204 0.3835831
```

```r
head(loan_train$default)
```

```
## [1] Yes No  No  No  Yes No
## Levels: No Yes
```

Here, for the first prediction the model says there is probability of default is 0.5604768 and the prediction is correct. For the second prediction, the model says the probability of default is 0.3734891 and the prediction is correct as well.

```r
#Misclassification error-train data

pred1 <- ifelse(prediction > 0.5, "Yes", "No")

table <- table(predicted = pred1, Actutal = loan_train$default) # confusion matrix
table
```

```
##          Actutal
## predicted  No Yes
##       No  510 125
##       Yes  60 105
```

```r
sum(diag(table))/ sum(table)
```

```
## [1] 0.76875
```

The above confusion matrix explains that in actual the customers with probability of default are 230 and the model predicted 105 correctly, the patients with no probability of default are 570 and the model predicted 510 correctly.The accuracy of the model is 76.8%

```r
#mutating the probability of default in to the original data
model_prob <- predict(model1, LoanApplicationData2, type = "response")

LoanApplicationData2$Prob_of_Default <- model_prob
```

```r
#default probability buckets

LoanApplicationData2 <- LoanApplicationData2 %>%
  mutate(POD = case_when(Prob_of_Default < 0.25 ~ "Low",
                         Prob_of_Default > 0.65 ~ "High", TRUE ~ "Medium"))


LoanApplicationData2$POD <- factor(LoanApplicationData2$POD,
                                   levels = c("Low", "Medium", "High"),
                                   ordered = TRUE)


LoanApplicationData2$POD <- as.numeric(LoanApplicationData2$POD)
table(LoanApplicationData2$POD)
```

```
##
##   1   2   3
## 546 348 106
```

The probability of default buckets says the the number of customers having low, medium and high POD are 546, 348 and 106 respectively.

Codebook

Variable Name	Variable Label	Missing Data	Typical Range	Data Type	Value	Label
Checking Account balance	Checking__balance	-	-	Char	-	-
Duration of loans in months	months__loan__duration	-	4-72	Int	-	-
Existing credit history	existing__credit__history	-	-	Char	-	-
Purpose of loan	purpose__of__loan	-	-	Char	-	-
Requested amount of loan	requested__amount	-	250-18424	Int	-	-
Savings balance	savings__balance	-	-	Char	-	-
Length of employment	employment__length	-	-	Char	-	-
Installment rate	installment__rate	-		Char	1,2,3,4	weekly,bi-weekly,monthly,quarterly
Personal status	personal__status	-	-	Char	-	-
Other debtors	other__debtors	-	-	Char	-	-
Residence history	residence__history	-	1-4	Int	-	-
Property	Property	-	-	Char	-	-
Installment plan	installment__plan	-	-	Char	-	-
Housing	housing	-	-	Char	-	-
Existing loans	existing__loans	-	1-4	Int	-	-
Default in loan	default	-		Char	1,2	No, Yes
Number of dependents	dependents	-	1-2	Int	-	-
Landline	landline	-	-	Char	-	-
Foreign worker	foreign__worker	-	-	Char	-	-
Job	job	-	-	Char	-	-