# Performing TSNE and Feature Selection exercise

victoria

2022-04-01

## Analysis on Carrefour Kenya to inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales.

### Defining the Question

You are a Data analyst at Carrefour Kenya and are currently undertaking a project that will inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

#### a) Specifying the Question

Reducing your dataset to a low dimensional dataset using the PCA

### b) Defining the Metric for Success

Involves reducing your dataset to a low dimensional dataset using the PCA. You will be required to perform your analysis and provide insights gained from your analysis..

### c) Understanding the context

### d) Recording the Experimental Design

Define the question, the metric for success, the context, experimental design taken.

Read and explore the given dataset.

Define the appropriateness of the available data to answer the given question.

Find and deal with outliers, anomalies, and missing data within the dataset.

Perform univariate, bivariate recording your observations.

Perform Principal Component Analysis.

Recommendation and conclusions

### e) Data Relevance

The link to the dataset [http://bit.ly/CarreFourDataset].

### Loading libraries

```
library(relaimpo)
```

```
## Loading required package: MASS
```

```
## Loading required package: boot
```

```
## Loading required package: survey
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##     aml
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
## Loading required package: mitools
```

```
## This is the global version of package relaimpo.
```

```
## If you are a non-US user, a version with the interesting additional metric pmvd is available
```

```
## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```
library(data.table)
library(ggplot2) # Data visualization
library(cluster)
library(ggthemes) # Plot themes
library(plotly) # Interactive data visualizations
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:MASS':
##
##     select

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(wskm)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

## Loading required package: latticeExtra
```

```
##
## Attaching package: 'latticeExtra'

## The following object is masked from 'package:ggplot2':
##
##     layer

## Loading required package: fpc
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggcorrplot)
library(moments)
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

## The following object is masked from 'package:boot':
##
##     logit
```

## Data Understanding

### Loading our dataset

```
df<- read.csv("http://bit.ly/CarreFourDataset")
```

**Previewing first 5 rows**

```
head(df)
```

```
##    Invoice.ID Branch Customer.type Gender           Product.line Unit.price
## 1 750-67-8428      A        Member Female      Health and beauty      74.69
## 2 226-31-3081      C        Normal Female Electronic accessories      15.28
## 3 631-41-3108      A        Normal   Male     Home and lifestyle      46.33
## 4 123-19-1176      A        Member   Male      Health and beauty      58.22
## 5 373-73-7910      A        Normal   Male      Sports and travel      86.31
## 6 699-14-3026      C        Normal   Male Electronic accessories      85.39
##   Quantity     Tax      Date  Time     Payment   cogs gross.margin.percentage
## 1        7 26.1415  1/5/2019 13:08     Ewallet 522.83                4.761905
## 2        5  3.8200  3/8/2019 10:29        Cash  76.40                4.761905
## 3        7 16.2155  3/3/2019 13:23 Credit card 324.31                4.761905
## 4        8 23.2880 1/27/2019 20:33     Ewallet 465.76                4.761905
## 5        7 30.2085  2/8/2019 10:37     Ewallet 604.17                4.761905
## 6        7 29.8865 3/25/2019 18:30     Ewallet 597.73                4.761905
##   gross.income Rating    Total
## 1      26.1415    9.1 548.9715
## 2       3.8200    9.6  80.2200
## 3      16.2155    7.4 340.5255
## 4      23.2880    8.4 489.0480
## 5      30.2085    5.3 634.3785
## 6      29.8865    4.1 627.6165
```

**Previewing last 5 rows**

```
tail(df)
```

```
##       Invoice.ID Branch Customer.type Gender           Product.line Unit.price
## 995  652-49-6720      C        Member Female Electronic accessories      60.95
## 996  233-67-5758      C        Normal   Male      Health and beauty      40.35
## 997  303-96-2227      B        Normal Female     Home and lifestyle      97.38
## 998  727-02-1313      A        Member   Male     Food and beverages      31.84
## 999  347-56-2442      A        Normal   Male     Home and lifestyle      65.82
## 1000 849-09-3807      A        Member Female    Fashion accessories      88.34
##      Quantity     Tax      Date  Time Payment   cogs gross.margin.percentage
## 995         1  3.0475 2/18/2019 11:40 Ewallet  60.95                4.761905
## 996         1  2.0175 1/29/2019 13:46 Ewallet  40.35                4.761905
## 997        10 48.6900  3/2/2019 17:16 Ewallet 973.80                4.761905
## 998         1  1.5920  2/9/2019 13:22    Cash  31.84                4.761905
## 999         1  3.2910 2/22/2019 15:33    Cash  65.82                4.761905
## 1000        7 30.9190 2/18/2019 13:28    Cash 618.38                4.761905
##      gross.income Rating     Total
## 995        3.0475    5.9   63.9975
```

```
## 996        2.0175    6.2   42.3675
## 997       48.6900    4.4 1022.4900
## 998        1.5920    7.7   33.4320
## 999        3.2910    4.1   69.1110
## 1000      30.9190    6.6  649.2990
```

**checking for data types**

```
sapply(df, class)
```

```
##              Invoice.ID                  Branch           Customer.type
##             "character"             "character"             "character"
##                  Gender           Product.line              Unit.price
##             "character"             "character"               "numeric"
##                Quantity                     Tax                    Date
##               "integer"               "numeric"             "character"
##                    Time                 Payment                    cogs
##             "character"             "character"               "numeric"
## gross.margin.percentage            gross.income                  Rating
##               "numeric"               "numeric"               "numeric"
##                   Total
##               "numeric"
```

We have character ,numeric and integer data types

**Checking for column names**

```
colnames(df)
```

```
##  [1] "Invoice.ID"              "Branch"
##  [3] "Customer.type"           "Gender"
##  [5] "Product.line"            "Unit.price"
##  [7] "Quantity"                "Tax"
##  [9] "Date"                    "Time"
## [11] "Payment"                 "cogs"
## [13] "gross.margin.percentage" "gross.income"
## [15] "Rating"                  "Total"
```

**checking for shape of dataset**

```
dim(df)
```

```
## [1] 1000   16
```

We have 1000 records and 16 variables

```
summary(df)
```

```
##   Invoice.ID           Branch           Customer.type        Gender
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  Product.line         Unit.price       Quantity          Tax
##  Length:1000        Min.   :10.08   Min.   : 1.00   Min.   : 0.5085
##  Class :character   1st Qu.:32.88   1st Qu.: 3.00   1st Qu.: 5.9249
##  Mode  :character   Median :55.23   Median : 5.00   Median :12.0880
##                     Mean   :55.67   Mean   : 5.51   Mean   :15.3794
##                     3rd Qu.:77.94   3rd Qu.: 8.00   3rd Qu.:22.4453
##                     Max.   :99.96   Max.   :10.00   Max.   :49.6500
##      Date               Time             Payment               cogs
##  Length:1000        Length:1000        Length:1000        Min.   : 10.17
##  Class :character   Class :character   Class :character   1st Qu.:118.50
##  Mode  :character   Mode  :character   Mode  :character   Median :241.76
##                                                           Mean   :307.59
##                                                           3rd Qu.:448.90
##                                                           Max.   :993.00
##  gross.margin.percentage  gross.income         Rating           Total
##  Min.   :4.762           Min.   : 0.5085   Min.   : 4.000   Min.   :  10.68
##  1st Qu.:4.762           1st Qu.: 5.9249   1st Qu.: 5.500   1st Qu.: 124.42
##  Median :4.762           Median :12.0880   Median : 7.000   Median : 253.85
##  Mean   :4.762           Mean   :15.3794   Mean   : 6.973   Mean   : 322.97
##  3rd Qu.:4.762           3rd Qu.:22.4453   3rd Qu.: 8.500   3rd Qu.: 471.35
##  Max.   :4.762           Max.   :49.6500   Max.   :10.000   Max.   :1042.65
```

## Data Cleaning

### 4.1 Completeness

```
colSums(is.na(df))
```

```
##              Invoice.ID                  Branch           Customer.type
##                       0                       0                       0
##                  Gender            Product.line              Unit.price
##                       0                       0                       0
##                Quantity                     Tax                    Date
##                       0                       0                       0
##                    Time                 Payment                    cogs
##                       0                       0                       0
## gross.margin.percentage            gross.income                  Rating
##                       0                       0                       0
##                   Total
##                       0
```

There no missing values

### 4.2 Consistency

```
# checking for duplicates
duplicated_rows <- colSums(df[duplicated(df),])
duplicated_rows
```

```
##              Invoice.ID                  Branch           Customer.type
##                       0                       0                       0
##                  Gender           Product.line             Unit.price
##                       0                       0                       0
##                Quantity                     Tax                    Date
##                       0                       0                       0
##                    Time                 Payment                    cogs
##                       0                       0                       0
## gross.margin.percentage            gross.income                  Rating
##                       0                       0                       0
##                   Total
##                       0
```

There are no duplicates

### 4.3 Uniformity

```
# Changing the column namesto lower case
names(df) <- tolower(names(df))
names(df)
```

```
##  [1] "invoice.id"              "branch"
##  [3] "customer.type"           "gender"
##  [5] "product.line"            "unit.price"
##  [7] "quantity"                "tax"
##  [9] "date"                    "time"
## [11] "payment"                 "cogs"
## [13] "gross.margin.percentage" "gross.income"
## [15] "rating"                  "total"
```

## Exploratory Data Analysis

### Univariate Analysis

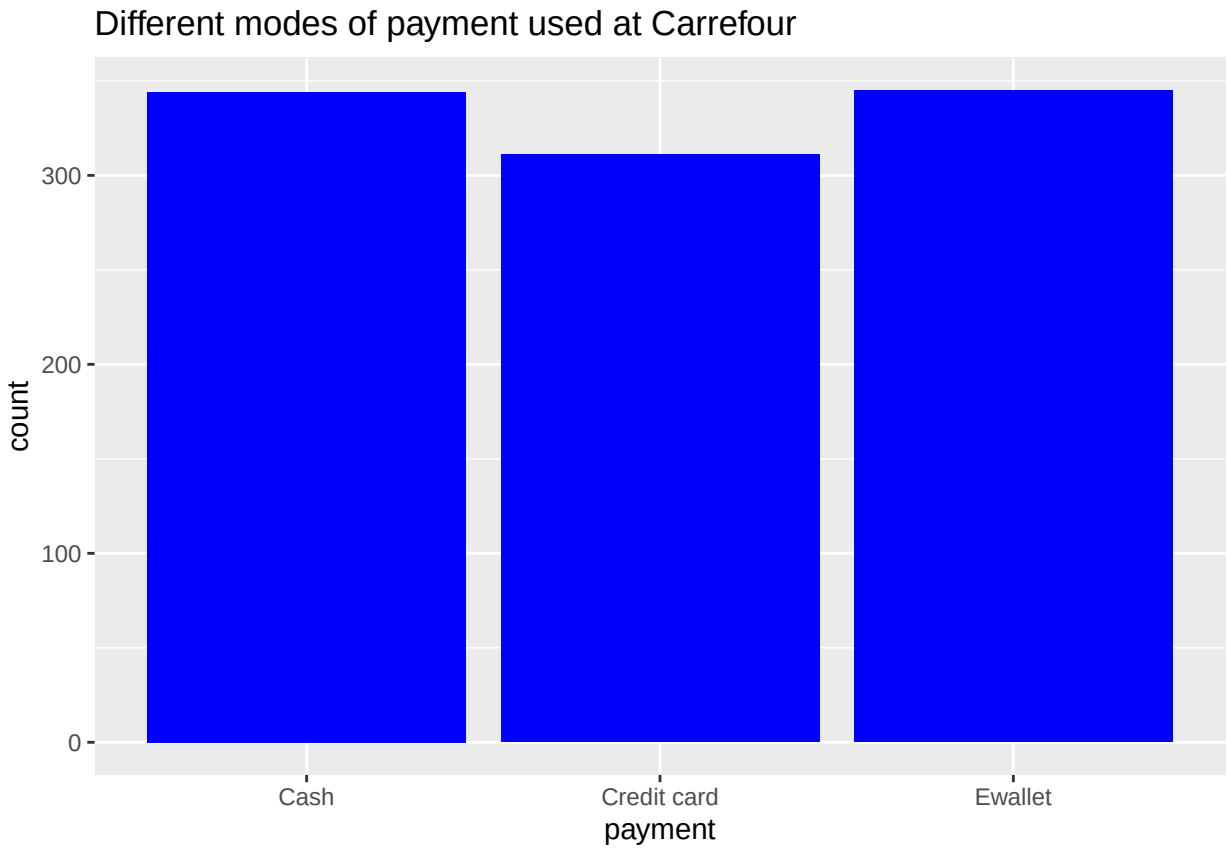
```
describe(df)
```

```
##                         vars    n   mean     sd median trimmed    mad   min
## invoice.id*                1 1000 500.50 288.82 500.50  500.50 370.65  1.00
## branch*                    2 1000   1.99   0.82   2.00    1.99   1.48  1.00
## customer.type*             3 1000   1.50   0.50   1.00    1.50   0.00  1.00
## gender*                    4 1000   1.50   0.50   1.00    1.50   0.00  1.00
## product.line*              5 1000   3.45   1.72   3.00    3.44   1.48  1.00
```

```
## unit.price                 6 1000  55.67  26.49  55.23   55.62  33.37 10.08
## quantity                   7 1000   5.51   2.92   5.00    5.51   2.97  1.00
## tax                        8 1000  15.38  11.71  12.09   14.00  11.13  0.51
## date*                      9 1000  45.58  25.89  47.00   45.63  34.10  1.00
## time*                     10 1000 252.18 147.07 249.00  252.49 190.51  1.00
## payment*                  11 1000   2.00   0.83   2.00    2.00   1.48  1.00
## cogs                      12 1000 307.59 234.18 241.76  279.91 222.65 10.17
## gross.margin.percentage   13 1000   4.76   0.00   4.76    4.76   0.00  4.76
## gross.income              14 1000  15.38  11.71  12.09   14.00  11.13  0.51
## rating                    15 1000   6.97   1.72   7.00    6.97   2.22  4.00
## total                     16 1000 322.97 245.89 253.85  293.91 233.78 10.68
##                             max   range  skew kurtosis   se
## invoice.id*             1000.00  999.00  0.00    -1.20 9.13
## branch*                    3.00    2.00  0.02    -1.51 0.03
## customer.type*             2.00    1.00  0.00    -2.00 0.02
## gender*                    2.00    1.00  0.00    -2.00 0.02
## product.line*              6.00    5.00  0.06    -1.28 0.05
## unit.price                99.96   89.88  0.01    -1.22 0.84
## quantity                  10.00    9.00  0.01    -1.22 0.09
## tax                       49.65   49.14  0.89    -0.09 0.37
## date*                     89.00   88.00 -0.03    -1.23 0.82
## time*                    506.00  505.00  0.00    -1.25 4.65
## payment*                   3.00    2.00  0.00    -1.55 0.03
## cogs                     993.00  982.83  0.89    -0.09 7.41
## gross.margin.percentage    4.76    0.00   NaN      NaN 0.00
## gross.income              49.65   49.14  0.89    -0.09 0.37
## rating                    10.00    6.00  0.01    -1.16 0.05
## total                   1042.65 1031.97  0.89    -0.09 7.78
```

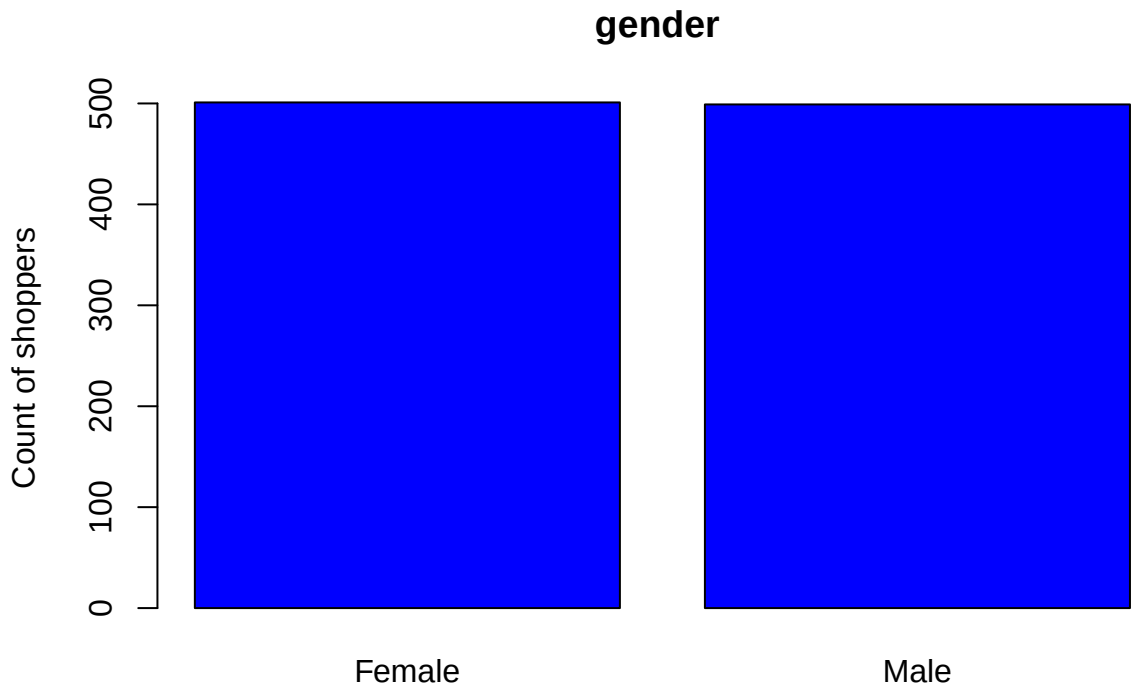
Above is a summary of mean ,mode,variance,standard deviation ,minimum ,maximum, range ,skewness and kurtosis of the dataset given.

```
ggplot(data = df) +
  geom_bar(mapping = aes(x = payment),fill="blue") +
  labs(title="Different modes of payment used at Carrefour")
```



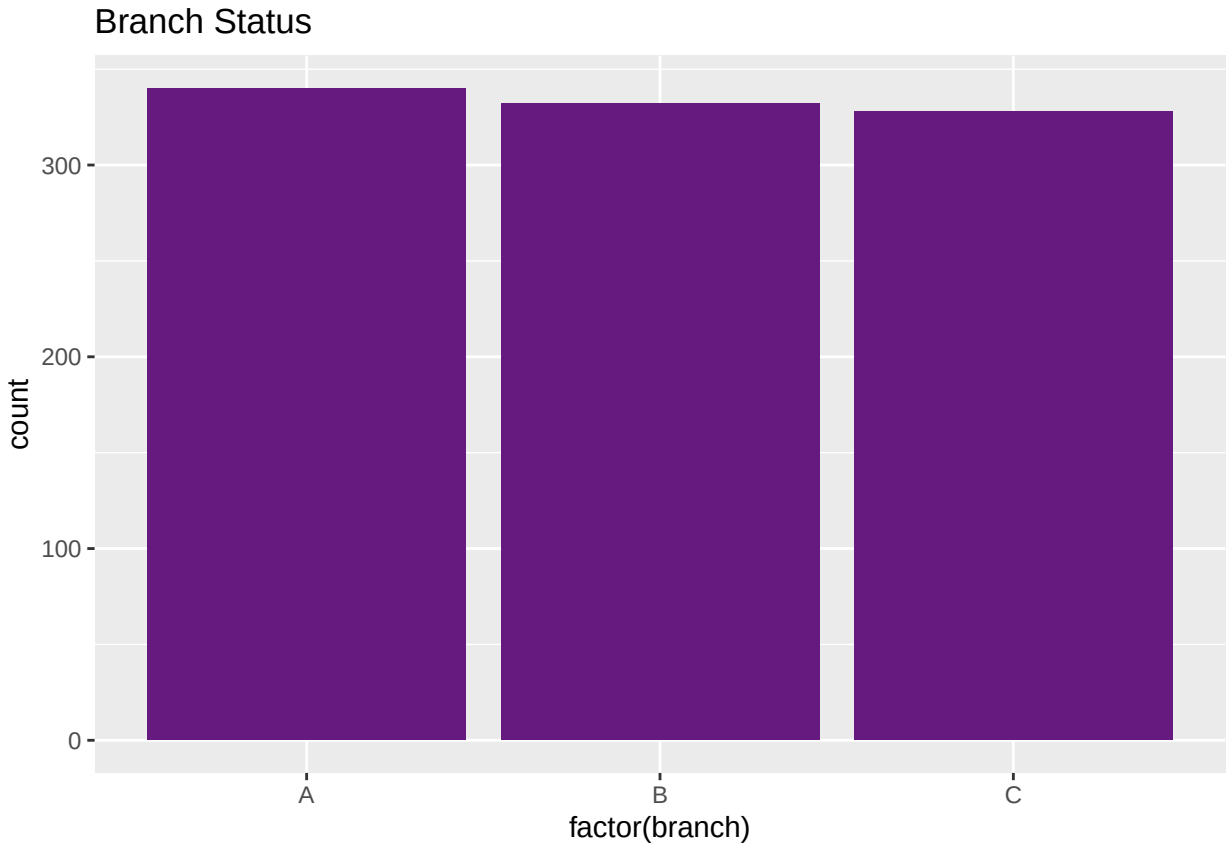

Most of the customers use Ewallet to make payments.

```
barplot(table(df$gender),main="gender ",col = "blue",ylab = "Count of shoppers")
```




There an equal number of shoppers in terms of gender.

```
ggplot(df, aes(x=factor(`branch`))) + geom_bar( fill=rgb(0.4,0.1,0.5)) +
  labs(title="Branch Status")
```




Branch A is the most busiest branch.

### Bivariate Analysis

## PCA

```
# selecting the numerical data columns
df1 <- df %>% select_if(is.numeric)
colnames(df1)
```

```
## [1] "unit.price"              "quantity"
## [3] "tax"                     "cogs"
## [5] "gross.margin.percentage" "gross.income"
## [7] "rating"                  "total"
```

```
# Encoding using Dummy variables and excluding unique ID and date time data
dums<-dummyVars("~.",data=df[,c(-1,-9,-10)])
dums
```

```
## Dummy Variable Object
##
## Formula: ~.
## <environment: 0x00000000314408c8>
```

```
## 13 variables, 0 factors
## Variables and levels will be separated by '.'
## A less than full rank encoding is used
```

```
dum.df<-data.frame(predict(dums,newdata =df[,c(-1,-9,-10)]))
```

```
df2 <- subset(df1, select = c("unit.price", "quantity", "tax", "cogs", "gross.income", "rating", "total"
colnames(df2)
```

```
## [1] "unit.price"   "quantity"     "tax"          "cogs"         "gross.income"
## [6] "rating"       "total"
```

```
#Finding the structure of the dataset
str(df2)
```

```
## 'data.frame':    1000 obs. of  7 variables:
##  $ unit.price  : num  74.7 15.3 46.3 58.2 86.3 ...
##  $ quantity    : int  7 5 7 8 7 7 6 10 2 3 ...
##  $ tax         : num  26.14 3.82 16.22 23.29 30.21 ...
##  $ cogs        : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ gross.income: num  26.14 3.82 16.22 23.29 30.21 ...
##  $ rating      : num  9.1 9.6 7.4 8.4 5.3 4.1 5.8 8 7.2 5.9 ...
##  $ total       : num  549 80.2 340.5 489 634.4 ...
```

```
# We then pass df to the prcomp(). We also set two arguments, center and scale,
# to be TRUE then preview our object with summary
df3 <- prcomp(df2)
summary(df3)
```

```
## Importance of components:
##                             PC1      PC2     PC3     PC4       PC5       PC6
## Standard deviation     340.3819 20.53212 1.71932 1.24589 1.678e-13 7.548e-15
## Proportion of Variance   0.9963  0.00363 0.00003 0.00001 0.000e+00 0.000e+00
## Cumulative Proportion    0.9963  0.99996 0.99999 1.00000 1.000e+00 1.000e+00
##                               PC7
## Standard deviation     1.78e-15
## Proportion of Variance 0.00e+00
## Cumulative Proportion  1.00e+00
```

```
# Calling str() to have a look at your PCA object
str(df3)
```

```
## List of 5
##  $ sdev    : num [1:7] 3.40e+02 2.05e+01 1.72 1.25 1.68e-13 ...
##  $ rotation: num [1:7, 1:7] -0.04952 -0.00605 -0.0344 -0.68798 -0.0344 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "unit.price" "quantity" "tax" "cogs" ...
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:7] 55.67 5.51 15.38 307.59 15.38 ...
##   ..- attr(*, "names")= chr [1:7] "unit.price" "quantity" "tax" "cogs" ...
##  $ scale   : logi FALSE
```

```
##  $ x       : num [1:1000, 1:7] -313 337.2 -23.8 -229.5 -431.5 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:1000] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
install_github("vqv/ggbiplot")
```

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools 4.0 from https://cran.r-project.org/bin/windows/Rtools/.
```

```
## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last i
##   Use 'force = TRUE' to force installation
```

```
Sys.setenv(R_REMOTES_NO_ERRORS_FROM_WARNINGS="true")
#install_github("vqv/ggbiplot",force=TRUE)
library(ggbiplot)
```

```
## Loading required package: plyr
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

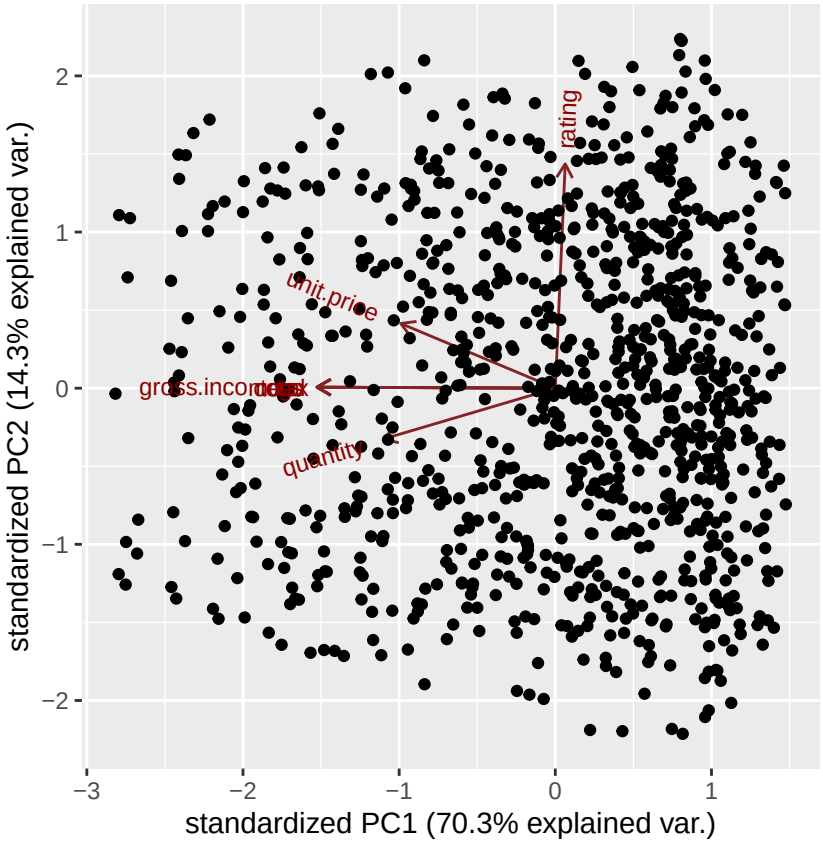
```
## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise
```

```
## Loading required package: scales
```

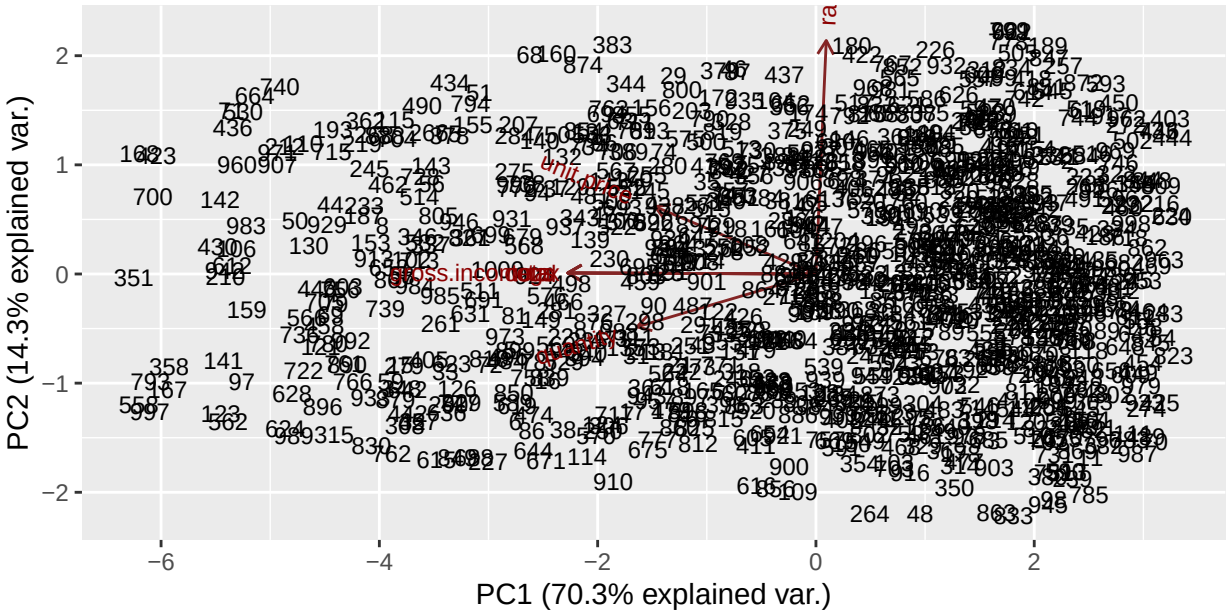
```
##
## Attaching package: 'scales'
```

```
## The following objects are masked from 'package:psych':
##
##     alpha, rescale
```

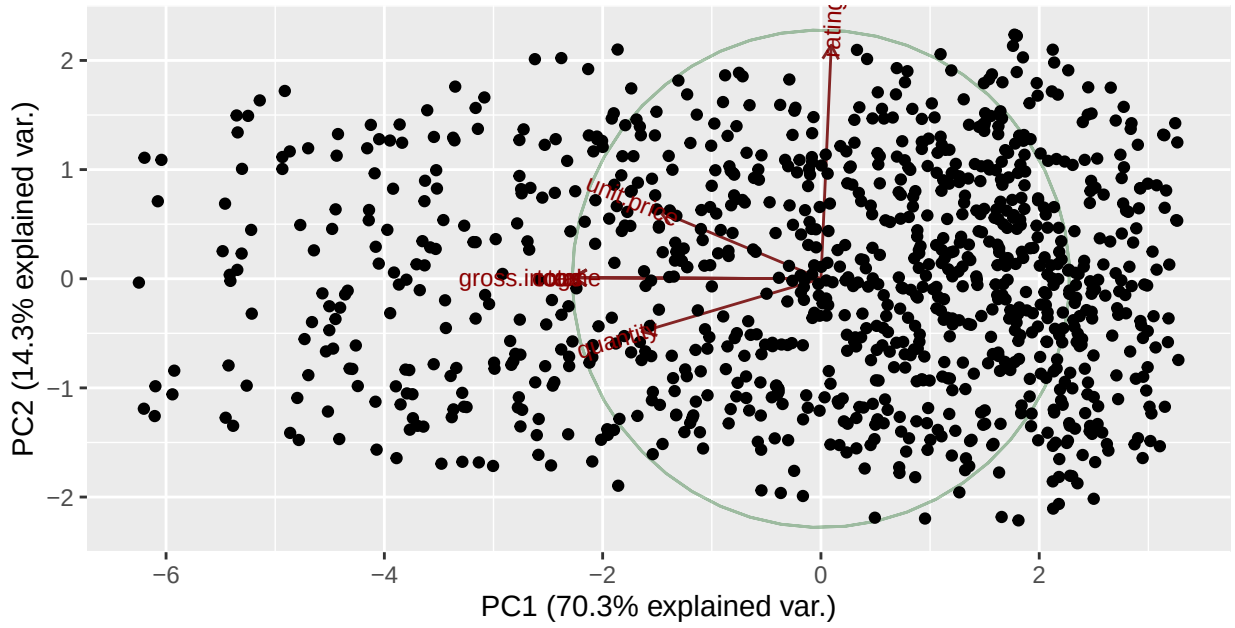
```
df3=prcomp(df2,center=T,scale.=T)
ggbiplot(df3)
```




```
# Adding more detail to the plot, we provide arguments rownames as labels
#
ggbiplot(df3, labels=rownames(df), obs.scale = 1, var.scale = 1)
```




```
# Getting the distribution of our categorical columns in the reduced dimension
ggbiplot(df3,obs.scale = 1,var.scale = 1,varname.adjust = 0.6,circle = TRUE,groups =df3$Payment)
```




# FEATURE SCALING

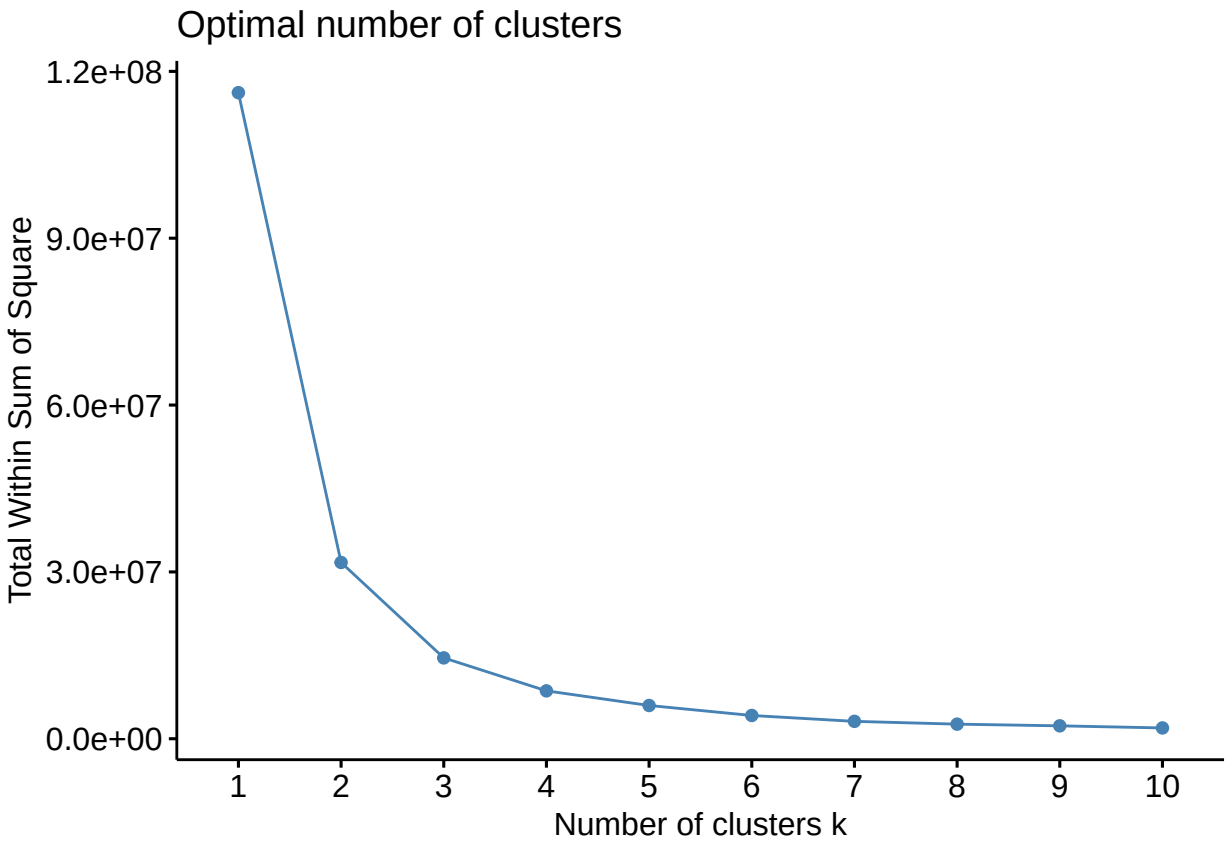
```
# Normalizing so as to perform cluster based feature selection using min max scaler
normalize<-function(x){
  return ((x-min(x))/(max(x)-min(x)))}
```

```
#Normalizing features
norm_df<-as.data.frame(lapply(dum.df, normalize))
summary(norm_df)
```

```
##     branchA         branchB         branchC      customer.typeMember
##  Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.00   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
##  Median :0.00   Median :0.000   Median :0.000   Median :1.000
##  Mean   :0.34   Mean   :0.332   Mean   :0.328   Mean   :0.501
##  3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.00   Max.   :1.000   Max.   :1.000   Max.   :1.000
##
##  customer.typeNormal  genderFemale     genderMale
##  Min.   :0.000       Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000       1st Qu.:0.000   1st Qu.:0.000
##  Median :0.000       Median :1.000   Median :0.000
##  Mean   :0.499       Mean   :0.501   Mean   :0.499
```

```
##  3rd Qu.:1.000       3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.000       Max.   :1.000   Max.   :1.000
##
##  product.lineElectronic.accessories product.lineFashion.accessories
##  Min.   :0.00                       Min.   :0.000
##  1st Qu.:0.00                       1st Qu.:0.000
##  Median :0.00                       Median :0.000
##  Mean   :0.17                       Mean   :0.178
##  3rd Qu.:0.00                       3rd Qu.:0.000
##  Max.   :1.00                       Max.   :1.000
##
##  product.lineFood.and.beverages product.lineHealth.and.beauty
##  Min.   :0.000                  Min.   :0.000
##  1st Qu.:0.000                  1st Qu.:0.000
##  Median :0.000                  Median :0.000
##  Mean   :0.174                  Mean   :0.152
##  3rd Qu.:0.000                  3rd Qu.:0.000
##  Max.   :1.000                  Max.   :1.000
##
##  product.lineHome.and.lifestyle product.lineSports.and.travel   unit.price
##  Min.   :0.00                   Min.   :0.000                 Min.   :0.0000
##  1st Qu.:0.00                   1st Qu.:0.000                 1st Qu.:0.2536
##  Median :0.00                   Median :0.000                 Median :0.5023
##  Mean   :0.16                   Mean   :0.166                 Mean   :0.5073
##  3rd Qu.:0.00                   3rd Qu.:0.000                 3rd Qu.:0.7550
##  Max.   :1.00                   Max.   :1.000                 Max.   :1.0000
##
##     quantity           tax          paymentCash    paymentCredit.card
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.2222   1st Qu.:0.1102   1st Qu.:0.000   1st Qu.:0.000
##  Median :0.4444   Median :0.2356   Median :0.000   Median :0.000
##  Mean   :0.5011   Mean   :0.3026   Mean   :0.344   Mean   :0.311
##  3rd Qu.:0.7778   3rd Qu.:0.4464   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.000
##
##  paymentEwallet       cogs        gross.margin.percentage  gross.income
##  Min.   :0.000   Min.   :0.0000   Min.   : NA             Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.1102   1st Qu.: NA             1st Qu.:0.1102
##  Median :0.000   Median :0.2356   Median : NA             Median :0.2356
##  Mean   :0.345   Mean   :0.3026   Mean   :NaN             Mean   :0.3026
##  3rd Qu.:1.000   3rd Qu.:0.4464   3rd Qu.: NA             3rd Qu.:0.4464
##  Max.   :1.000   Max.   :1.0000   Max.   : NA             Max.   :1.0000
##                                   NA's   :1000
##      rating           total
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2500   1st Qu.:0.1102
##  Median :0.5000   Median :0.2356
##  Mean   :0.4955   Mean   :0.3026
##  3rd Qu.:0.7500   3rd Qu.:0.4464
##  Max.   :1.0000   Max.   :1.0000
##
```

```
# Using the encoded set of data excluding the gross margin which is non variant. 4 are the optimum clus
fviz_nbclust(dum.df[,c(-21)],FUNcluster = kmeans,method = "wss")
```



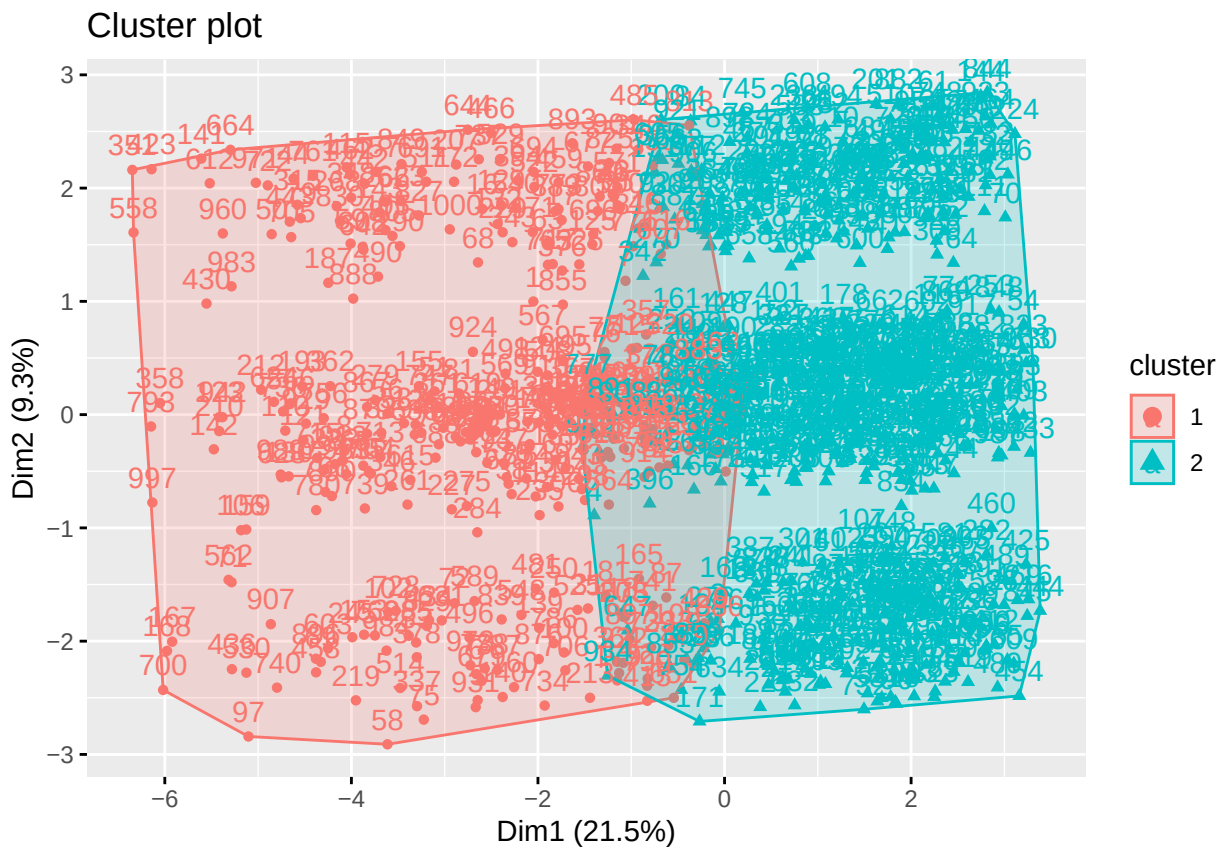

```
#Setting the initial clusters as 3 first and a variable for weight distribution
# We get to see the importance of every variable to the kmeans cluster
# We will exclude the gross margin percentage as its inclusion would give us errors in distance metrics

my_model<-ewkm(dum.df[,c(-21)],2,lambda = 2,maxiter=1000)
my_model
```

```
## K-means clustering with 2 clusters of sizes 340, 660
##
## Cluster means:
##     branchA   branchB   branchC customer.typeMember customer.typeNormal
## 1 0.3264706 0.3294118 0.3441176           0.5205882           0.4794118
## 2 0.3469697 0.3333333 0.3196970           0.4909091           0.5090909
##   genderFemale genderMale product.lineElectronic.accessories
## 1    0.5235294  0.4764706                          0.1735294
## 2    0.4893939  0.5106061                          0.1681818
##   product.lineFashion.accessories product.lineFood.and.beverages
## 1                       0.1705882                      0.1558824
## 2                       0.1818182                      0.1833333
##   product.lineHealth.and.beauty product.lineHome.and.lifestyle
## 1                     0.1117647                      0.1588235
## 2                     0.1727273                      0.1606061
##   product.lineSports.and.travel unit.price quantity      tax paymentCash
## 1                     0.2294118   75.58709 7.844118 29.08757   0.3470588
## 2                     0.1333333   45.41291 4.307576  8.31757   0.3424242
##   paymentCredit.card paymentEwallet     cogs gross.income rating    total
```

```
## 1          0.3029412      0.3500000 581.7514     29.08757  6.910 610.8389
## 2          0.3151515      0.3424242 166.3514      8.31757  7.005 174.6690
##
## Clustering vector:
##    [1] 1 2 2 2 1 1 1 1 2 2 2 2 2 1 1 1 1 1 2 2 1 2 2 2 2 1 2 2 1 2 1 1 1 2 1 2 1
##   [38] 1 2 2 2 2 1 1 2 1 1 2 2 1 1 2 2 2 2 1 2 1 1 2 2 2 1 2 2 2 2 1 1 2 1 1 2 1
##   [75] 1 1 1 2 1 2 1 2 2 2 2 1 1 2 2 2 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 1 2 1 2 1 2
##  [112] 2 1 1 1 2 2 2 2 2 1 1 1 1 1 1 2 2 1 1 2 1 2 1 1 2 2 2 1 1 1 1 1 2 1 2 2 2
##  [149] 1 2 2 2 1 2 1 1 2 2 1 1 2 2 2 2 1 2 1 1 2 1 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2
##  [186] 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 1 2 2 1 1 2 2 2 2 2 1 2 2 1
##  [223] 2 2 2 2 1 2 1 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 1 2 2 2 1 2 2 2 2
##  [260] 2 1 2 2 2 2 1 2 1 2 2 1 2 2 2 1 2 2 1 1 1 1 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 2
##  [297] 2 1 2 2 2 2 2 2 2 2 1 1 1 2 2 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1
##  [334] 2 2 2 1 1 2 1 2 2 1 1 2 1 2 1 2 2 1 2 1 2 2 2 1 1 2 2 1 1 2 1 2 2 1 2 2 2
##  [371] 2 2 1 2 2 1 2 1 1 2 1 2 1 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2
##  [408] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 1 2 1 1 2 2 2 2 1 1 2
##  [445] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 1 2 2 1 1 2 2 1 2 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1
##  [482] 2 2 1 1 2 1 2 2 1 2 2 2 2 2 2 1 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 1 1 1 2 2 2
##  [519] 2 1 2 1 2 2 1 1 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 1 2 1 1 2 2
##  [556] 2 2 1 2 2 2 1 2 1 2 1 1 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
##  [593] 2 2 1 2 2 1 2 2 2 2 1 1 2 2 1 2 2 2 2 1 2 2 1 1 1 1 1 2 2 2 1 1 2 2 2 1 2
##  [630] 2 1 2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2
##  [667] 2 2 2 2 1 2 2 2 1 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 1 1 1 1 2 2 1 1 1 2 2
##  [704] 1 1 1 2 2 2 2 1 2 1 2 1 2 1 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 1 2 1 1 1 1 1
##  [741] 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 1 2 1 2 2 1 1 1 2 2 1 2 2 1 2 1 1 1 2 2 2 2
##  [778] 2 2 1 2 1 2 2 2 1 1 2 2 1 2 2 1 1 2 2 2 2 2 2 1 2 2 1 1 1 2 2 2 2 2 1 2 1 1 2
##  [815] 1 1 2 2 2 2 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 2
##  [852] 2 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 1 2 2 2 1 2 1
##  [889] 2 1 2 1 1 2 2 2 1 1 2 1 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 1 2
##  [926] 2 2 2 1 2 1 2 1 2 1 2 1 2 1 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2
##  [963] 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 1 1 1 2 1 2 2 2 2 2 1 2 2
## [1000] 1
##
## Within cluster sum of squares by cluster:
## [1] 20387542 13971873
##  (between_SS / total_SS =  70.4 %)
##
## Available components:
##
##  [1] "cluster"          "centers"          "totss"            "withinss"
##  [5] "tot.withinss"     "betweenss"        "size"             "iterations"
##  [9] "total.iterations" "restarts"         "weights"
```

```
# Plotting the cluster with 2 as my maximum clusters
fviz_cluster(my_model,data=norm_df[,c(-21)])
```




```
# We get to the the importance of each parameter to the individual clusters
(my_model$weights)*10000
```

```
##      branchA    branchB    branchC customer.typeMember customer.typeNormal
## 1 0.04347494 0.04347494 0.04347494          0.04347494          0.04347494
## 2 0.04347505 0.04347505 0.04347505          0.04347505          0.04347505
##   genderFemale genderMale product.lineElectronic.accessories
## 1   0.04347494 0.04347494                            5.460446
## 2   0.04347505 0.04347505                            3.239693
##   product.lineFashion.accessories product.lineFood.and.beverages
## 1                       7.5796856                      40.8250251
## 2                       0.1738571                       0.1265718
##   product.lineHealth.and.beauty product.lineHome.and.lifestyle
## 1                   9916.371069                      28.98122
## 2                      1.205437                      17.34855
##   product.lineSports.and.travel unit.price   quantity        tax paymentCash
## 1                  4.347494e-02 0.04347494 0.04347494 0.04347494  0.04347494
## 2                  9.977167e+03 0.04347505 0.04347505 0.04347505  0.04347505
##   paymentCredit.card paymentEwallet       cogs gross.income     rating
## 1         0.04347494     0.04347494 0.04347494   0.04347494 0.04347494
## 2         0.04347505     0.04347505 0.04347505   0.04347505 0.04347505
##        total
## 1 0.04347494
## 2 0.04347505
```

## Conclusions

Male and female buyers have an equal number.

Ewallet payment is the most preferred mode of payment followed by cash then least preferred is credit card payment.

The busiest branch is A followed by B then C.

## Recommedations

The marketing department can come up with strategies to increase customer flow to the least busiest branch which is C in order to increase revenue.