Carrefour Market Analysis using Dimensionality Reduction and Feature Selection

Christopher Toromo

2022-06-10

1. Defining the Question

a) Specifying the Question

The objective is reduce the features of the dataset and later select the important features in order to perform and provide insights on the features that contribute the most information to the dataset.

b) Defining the Metric for Success

Our study will be considered successful if we are able to draw insights from the dataset

c) Understanding the context

Carrefour is a French multinational retail corporation headquartered in Massy, France. The eighth-largest retailer in the world by revenue, it operates a chain of hypermarkets, groceries stores and convenience stores, which as of January 2021, comprises its 12,225 stores in over 30 countries. The Carrefour Kenya 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).

d). Recording the Experimental Design

1. Data sourcing and loading
2. Data Understanding
3. Data Relevance
4. Data Preparation
5. Univariate Analysis
6. Bivariate Analysis
7. Performing PCA and Feature Selection
8. Challenging the solution
9. Conclusion and Recommendation
10. Follow up questions

e) Data Relevance

The datasets we used for this project can be found on https://bit.ly/CarreFourDataset

2. Reading the Data

```
# Importing the data

data <- read.csv("https://bit.ly/CarreFourDataset")
```

3. Checking the Data

a) Checking the Top data

```
head(data)
```

```
##    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
```

b) Checking the Bottom data

```
tail(data)
```

```
##       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
```

c) Checking the Structure of the Dataset

```
str(data)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ Invoice.ID            : chr  "750-67-8428" "226-31-3081" "631-41-3108" "123-19-1176" ...
##  $ Branch                : chr  "A" "C" "A" "A" ...
##  $ Customer.type         : chr  "Member" "Normal" "Normal" "Member" ...
##  $ Gender                : chr  "Female" "Female" "Male" "Male" ...
##  $ Product.line          : chr  "Health and beauty" "Electronic accessories" "Home and lifestyle" "
##  $ 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 ...
##  $ Date                  : chr  "1/5/2019" "3/8/2019" "3/3/2019" "1/27/2019" ...
##  $ Time                  : chr  "13:08" "10:29" "13:23" "20:33" ...
##  $ Payment               : chr  "Ewallet" "Cash" "Credit card" "Ewallet" ...
##  $ cogs                  : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ gross.margin.percentage: num  4.76 4.76 4.76 4.76 4.76 ...
##  $ 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 ...
```

d). Checking the shape of our data

```
dim(data)
```

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

From the above we can see our dataset contains 1000 records and 16 features.

4. Tidying the Dataset

a) Validation

Checking for unnecessary columns that do not contribute to the study.

```
colnames(data)
```

```
##  [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"
```

We do not have an irrelevant column in the dataframe.

Checking for invalid values

```
summary(data)
```

```
##   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
```

We do not have invalid characters.

b). Constistency

Checking for the missing values

```
colSums(is.na(data))
```

```
##               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
```

We do not have missing values in our data

c). Completeness

Checking for Duplicate Values in our data

```
sum(duplicated(data))
```

```
## [1] 0
```

There are no duplicate records in our data.

d). Uniformity

Checking Uniformity in the columns

```
str(data)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ Invoice.ID            : chr  "750-67-8428" "226-31-3081" "631-41-3108" "123-19-1176" ...
##  $ Branch                : chr  "A" "C" "A" "A" ...
##  $ Customer.type         : chr  "Member" "Normal" "Normal" "Member" ...
##  $ Gender                : chr  "Female" "Female" "Male" "Male" ...
##  $ Product.line          : chr  "Health and beauty" "Electronic accessories" "Home and lifestyle" "
##  $ 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 ...
##  $ Date                  : chr  "1/5/2019" "3/8/2019" "3/3/2019" "1/27/2019" ...
```

```
##  $ Time                   : chr  "13:08" "10:29" "13:23" "20:33" ...
##  $ Payment                : chr  "Ewallet" "Cash" "Credit card" "Ewallet" ...
##  $ cogs                   : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ gross.margin.percentage: num  4.76 4.76 4.76 4.76 4.76 ...
##  $ 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 shall convert all the 'chr' datatypes to factor.

```
data <- as.data.frame(unclass(data), stringsAsFactors = TRUE)

# Checking the Data again

str(data)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ Invoice.ID             : Factor w/ 1000 levels "101-17-6199",..: 815 143 654 19 340 734 316 265 7
##  $ Branch                 : Factor w/ 3 levels "A","B","C": 1 3 1 1 1 3 1 3 1 2 ...
##  $ Customer.type          : Factor w/ 2 levels "Member","Normal": 1 2 2 1 2 2 1 2 1 1 ...
##  $ Gender                 : Factor w/ 2 levels "Female","Male": 1 1 2 2 2 2 1 1 1 1 ...
##  $ Product.line           : Factor w/ 6 levels "Electronic accessories",..: 4 1 5 4 6 1 1 5 4 3 ...
##  $ 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 ...
##  $ Date                   : Factor w/ 89 levels "1/1/2019","1/10/2019",..: 27 88 82 20 58 77 49 48 2
##  $ Time                   : Factor w/ 506 levels "10:00","10:01",..: 147 24 156 486 30 394 215 78 34
##  $ Payment                : Factor w/ 3 levels "Cash","Credit card",..: 3 1 2 3 3 3 3 3 2 2 ...
##  $ cogs                   : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ gross.margin.percentage: num  4.76 4.76 4.76 4.76 4.76 ...
##  $ 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 ...
```

e).Checking for outliers

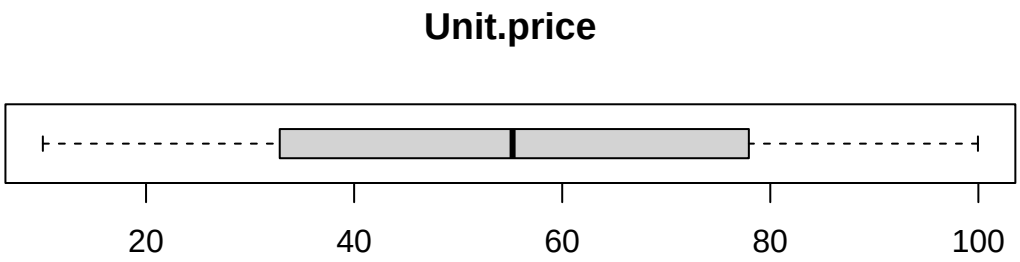
```
# Selecting Numerical columns
library("dplyr")
```

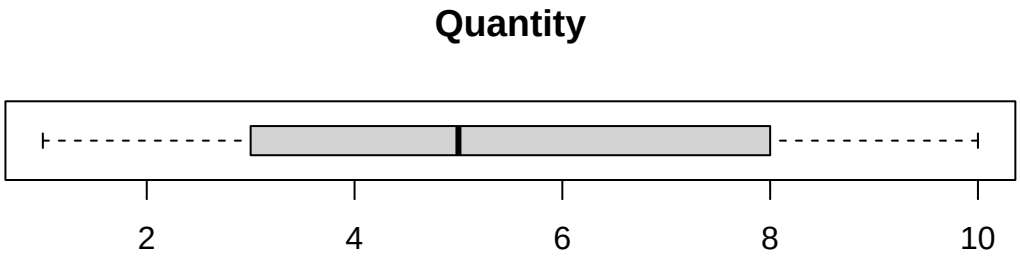
```
##
## Attaching package: 'dplyr'

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

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

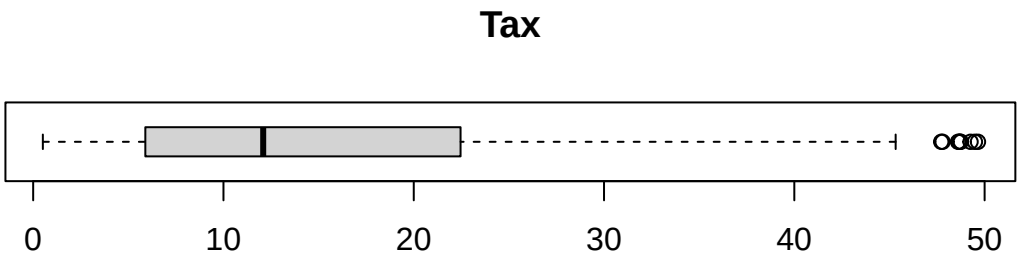
```
num_col <- select_if(data, is.numeric)

dim(num_col)
```

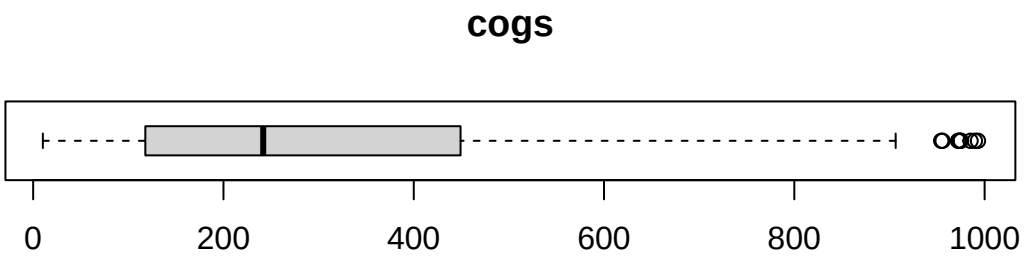
```
## [1] 1000    8
```

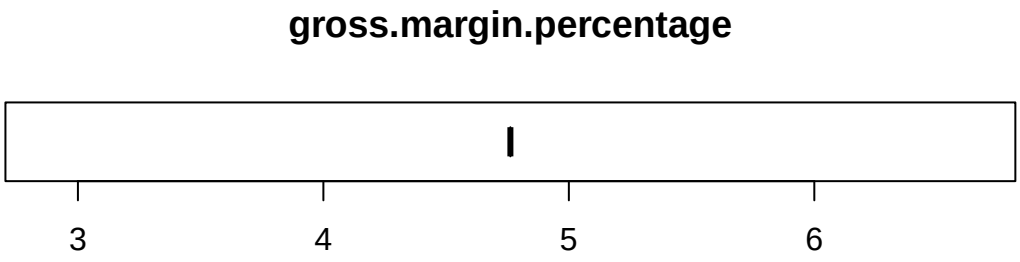
```
library("dplyr")

par(mfrow = c(2,1))

for (i in 1:8){
 boxplot(num_col[,i], main = names(num_col)[i], horizontal = TRUE)
}
```

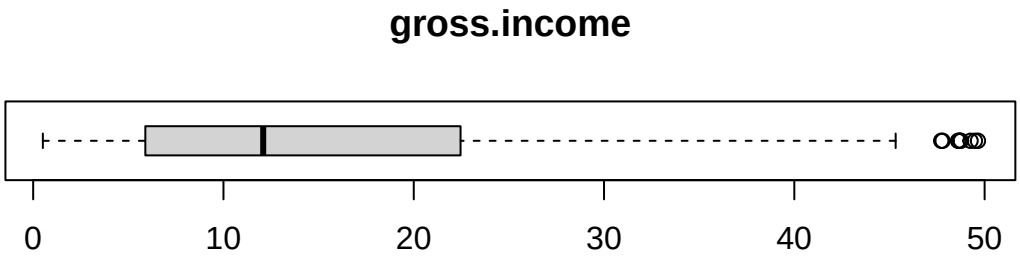


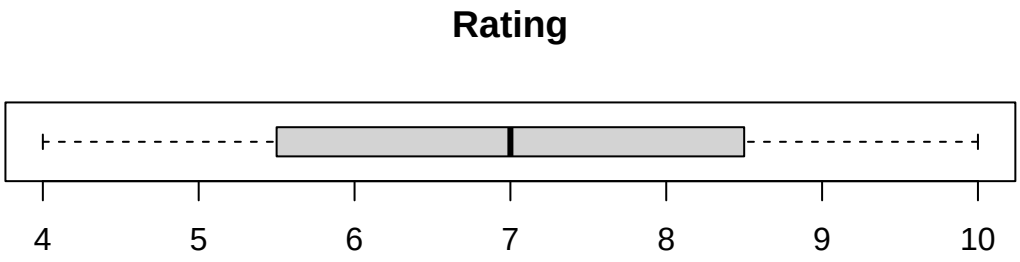


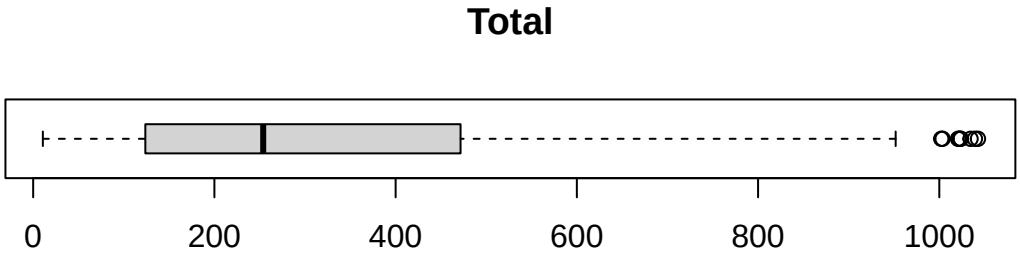

Tax
0
10
20
30
40
50


cogs
0
200
400
600
800
1000


gross.margin.percentage
3
4
5
6


gross.income
0
10
20
30
40
50





5. Exploratory Data Analysis

a) Univariate Analysis

Descriptive Statistics

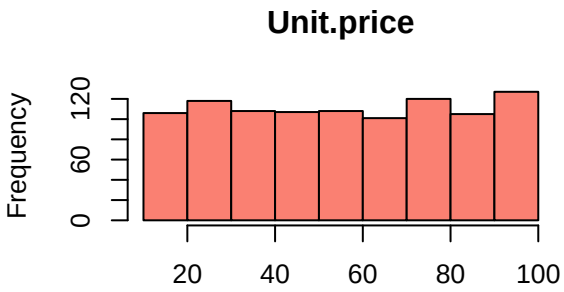
```
library(psych)

describe(data)
```

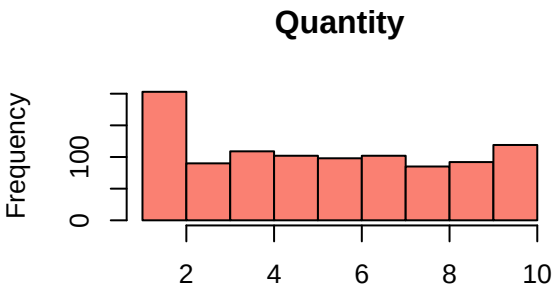
```
##                        vars    n   mean     sd median trimmed    mad   min
## Invoice.ID*               1 1000 500.50 288.82 500.50  500.50 370.65  1.00
## Branch*                   2 1000   1.99   0.82   2.00    1.99   1.48  1.00
## Customer.type*            3 1000   1.50   0.50   1.00    1.50   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
```

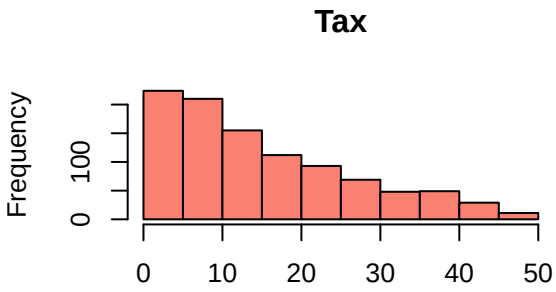
```
## gross.margin.percentage   13 1000   4.76   0.00   4.76    4.76   0.00  4.76
## gross.income              14 1000  15.38  11.71  12.09   14.00  11.13  0.51
## Rating                    15 1000   6.97   1.72   7.00    6.97   2.22  4.00
## Total                     16 1000 322.97 245.89 253.85  293.91 233.78 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
```

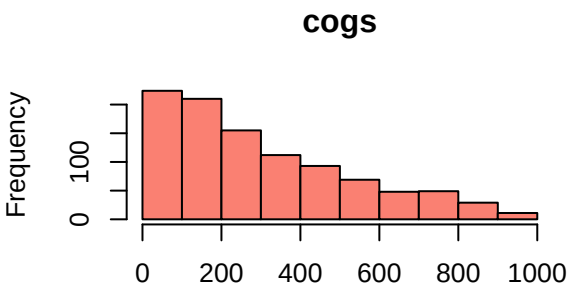
Distribution of the Features using Histogram

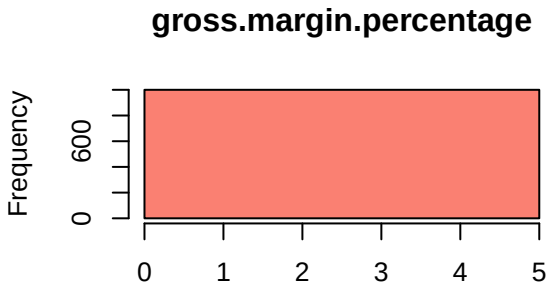
```
par(mfrow = c(2,2))
for (i in 1:8){ hist(num_col[,i],main = names(num_col)[i], xlab = NULL,col = "salmon")
  }
```

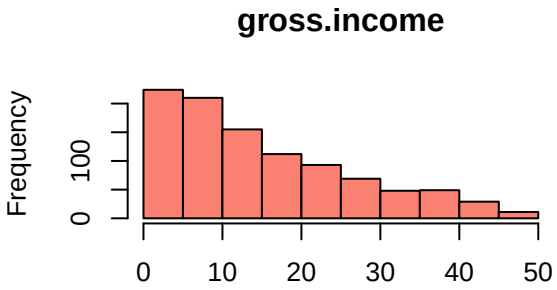

Unit.price
Frequency
0
60
120
20
40
60
80
100

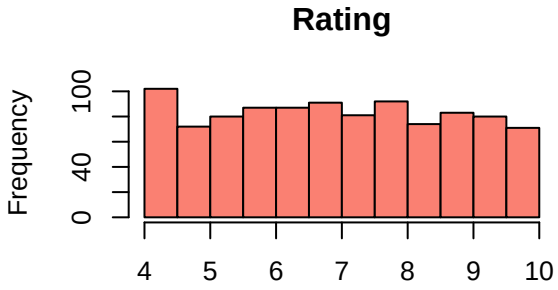

Quantity
Frequency
0
100
2
4
6
8
10

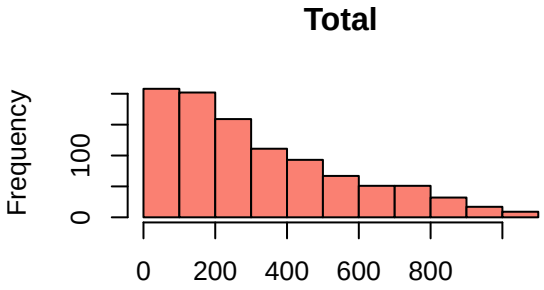

Tax
Frequency
0
100
0
10
20
30
40
50


cogs
Frequency
0
100
0
200
400
600
800
1000







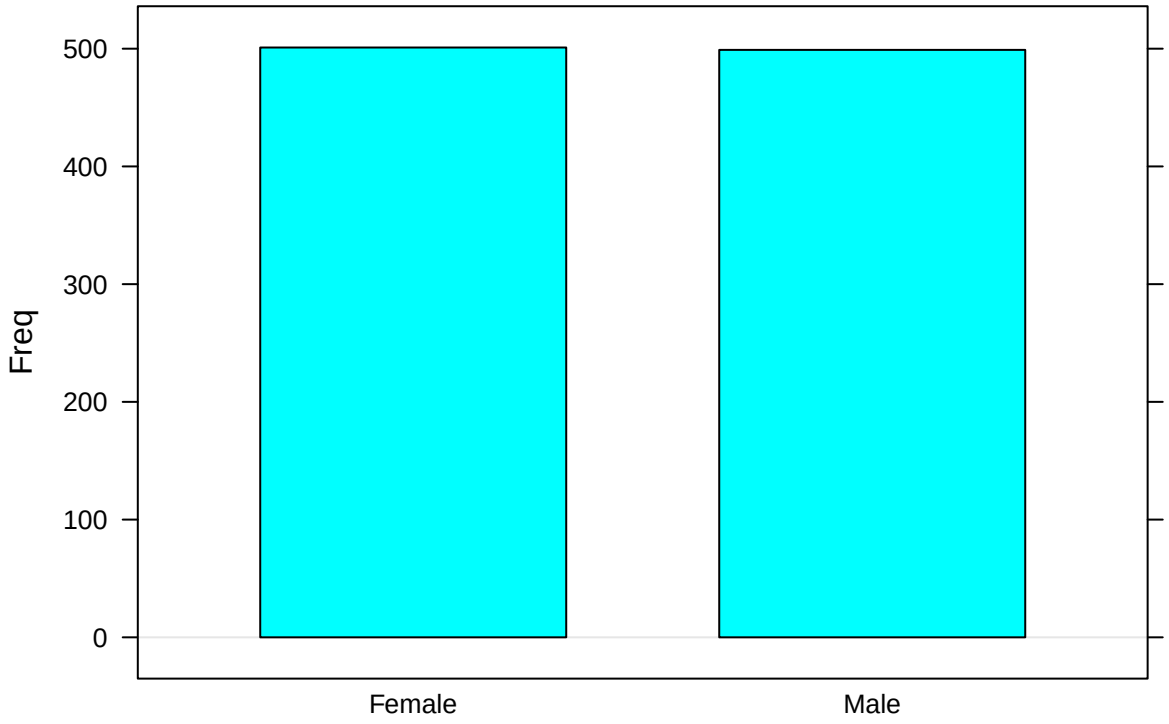


Bargraphs to visualize categorical features

```
# Plotting the Bar chart for Gender
if (!require('lattice')){
  install.packages('lattice')
  library('lattice')
}
```

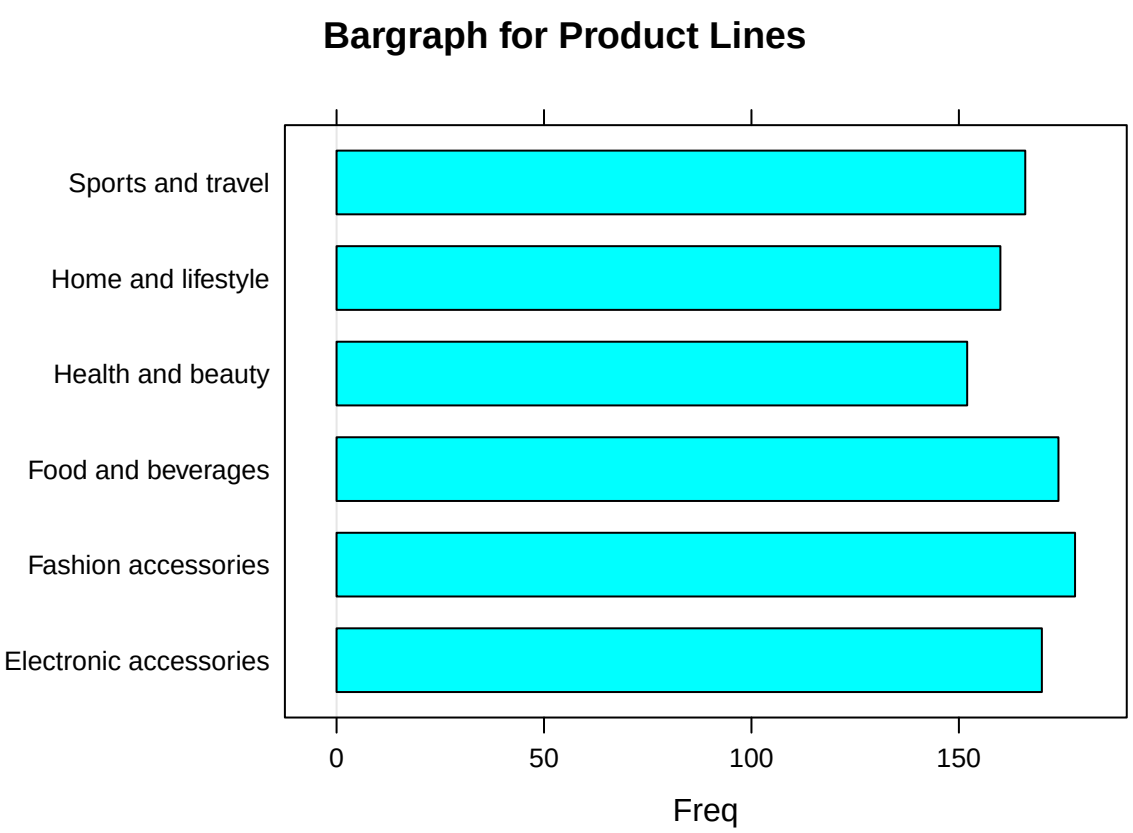
```
## Loading required package: lattice
```

```
barchart(data$Gender, Main = "Bar graph for Gender", horizontal = FALSE)
```



From the above we can see we have almost equal number of females and males

```
# Bar graph for Products

barchart(data$Product.line, color = "salmon", main = "Bargraph for Product Lines")
```



b) Bivariate Analysis

```
#Correlation Matrix
library("corrplot")
```

```
## corrplot 0.92 loaded
```

```
head(num_col)
```

```
##   Unit.price Quantity     Tax   cogs gross.margin.percentage gross.income
## 1      74.69        7 26.1415 522.83                4.761905      26.1415
## 2      15.28        5  3.8200  76.40                4.761905       3.8200
## 3      46.33        7 16.2155 324.31                4.761905      16.2155
## 4      58.22        8 23.2880 465.76                4.761905      23.2880
## 5      86.31        7 30.2085 604.17                4.761905      30.2085
## 6      85.39        7 29.8865 597.73                4.761905      29.8865
##   Rating    Total
## 1    9.1 548.9715
## 2    9.6  80.2200
## 3    7.4 340.5255
## 4    8.4 489.0480
## 5    5.3 634.3785
## 6    4.1 627.6165
```

```
library("GGally")
```

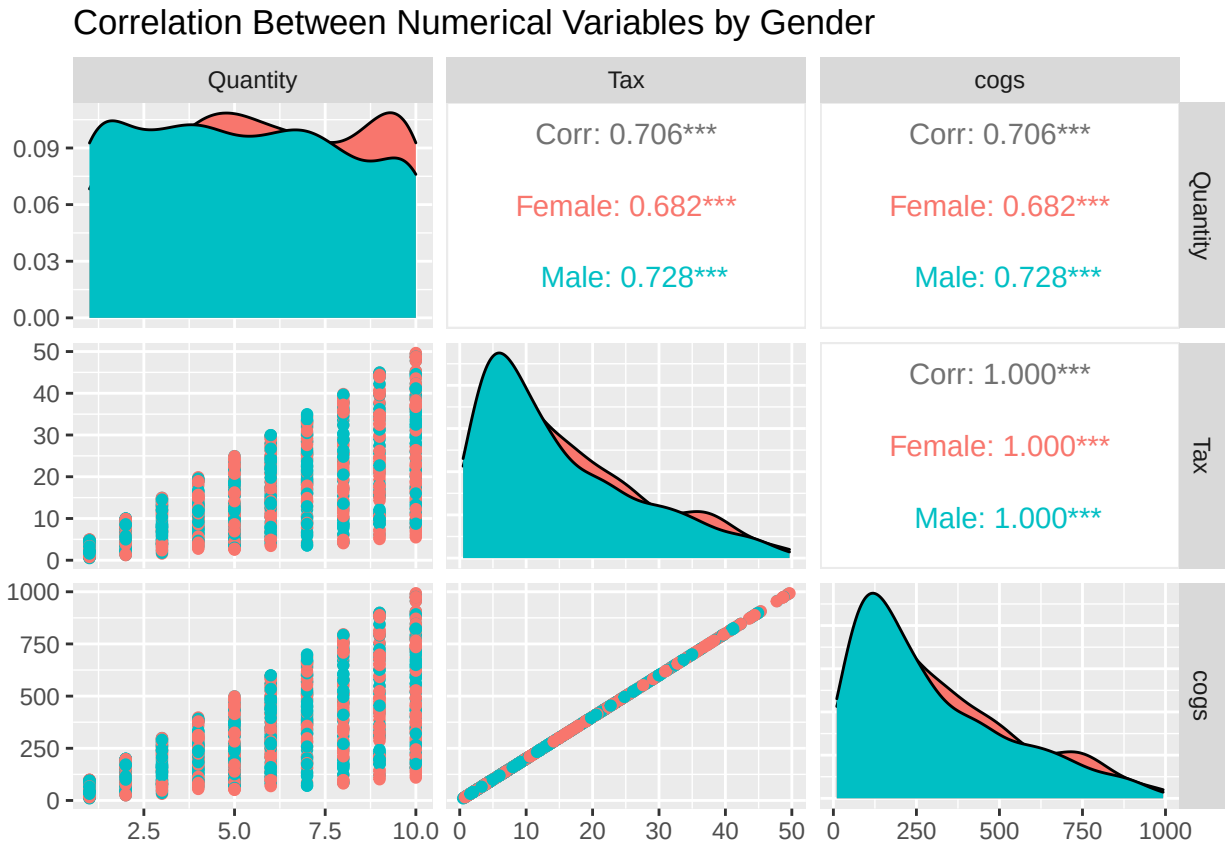
```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

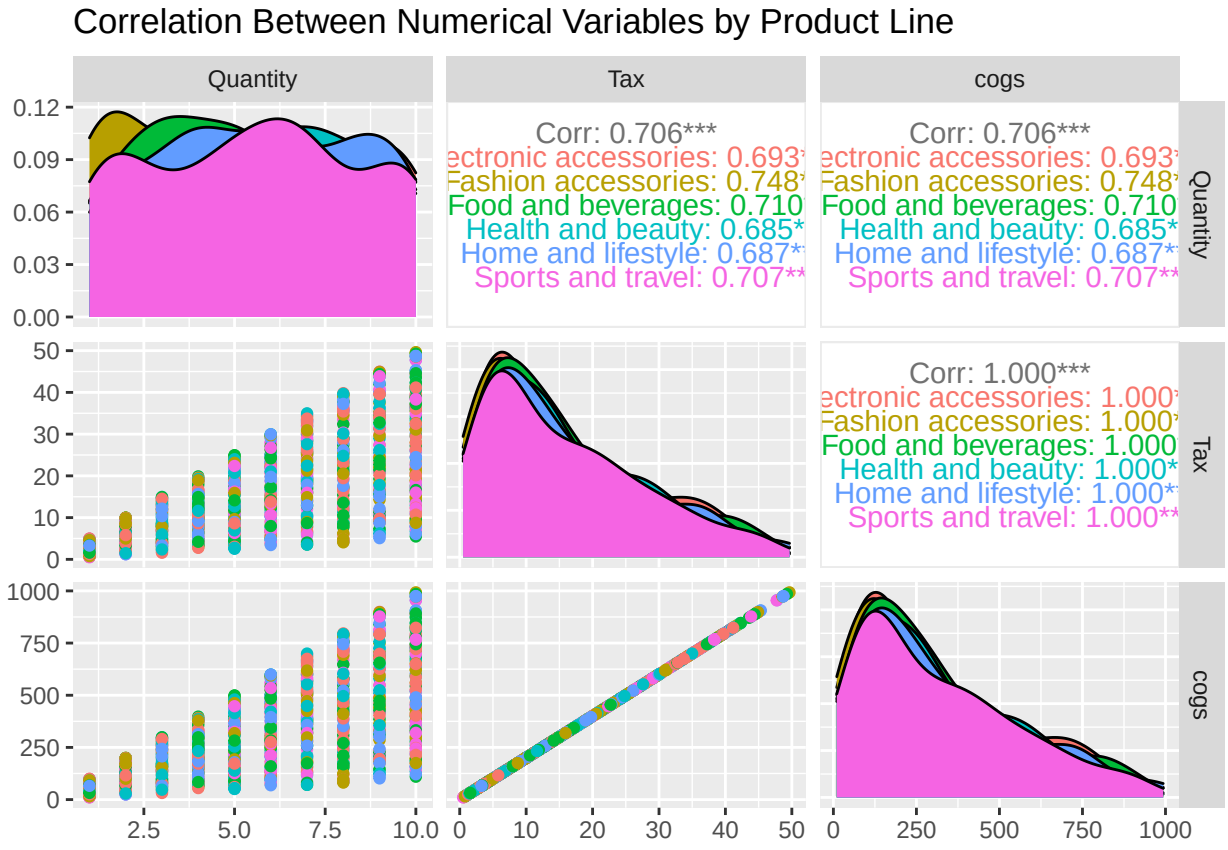
```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

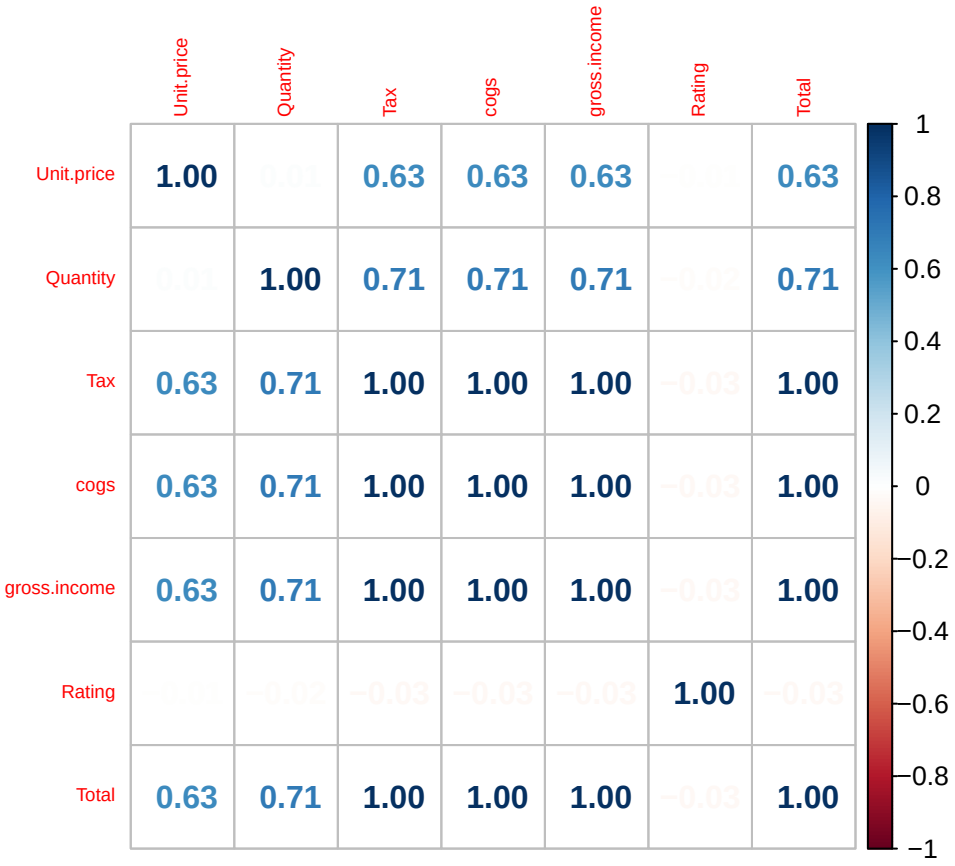
```
# plotting the correlation
ggpairs(num_col, columns = 2:4, ggplot2::aes(colour=data$Gender), title="Correlation Between Numerical V
```




```
# plotting the correlation
ggpairs(num_col, columns = 2:4, ggplot2::aes(colour=data$Product.line), title="Correlation Between Numer
```




```
corr_matrix <- cor(num_col[-2,-5,-7])

corrplot(corr_matrix, method='number',tl.cex = 0.6)
```



6. Modelling

a) Dimensionality Reduction

We shall be using the Principal Component Analysis(PCA) to apply the reduction technique

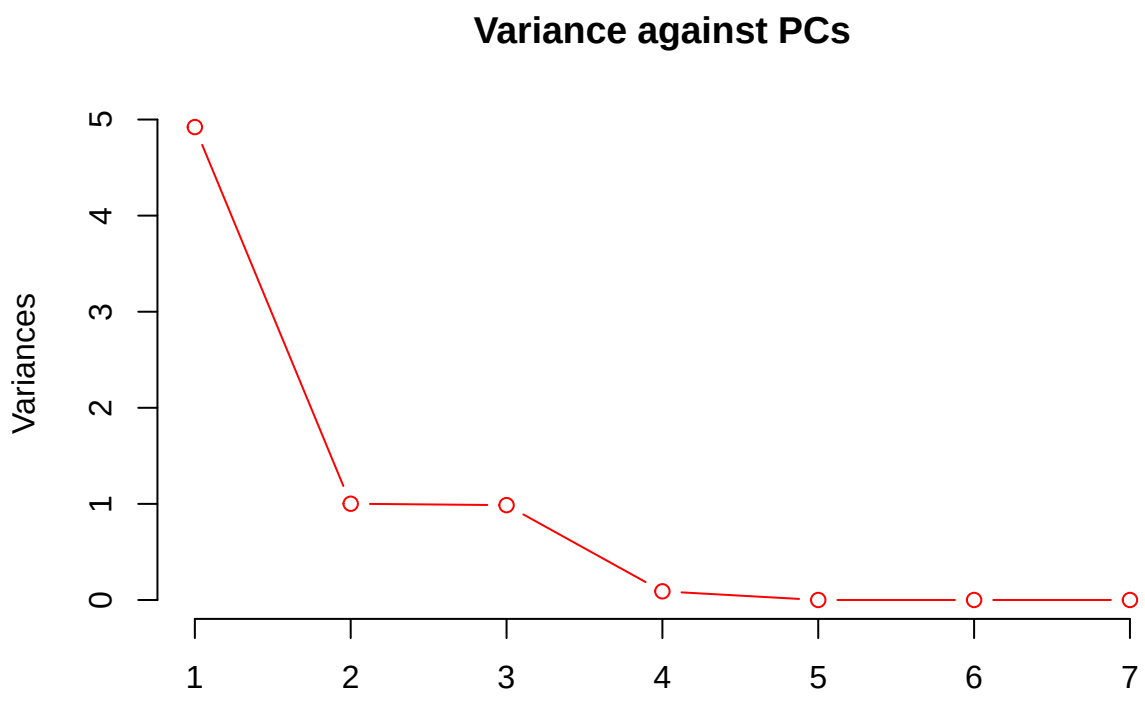
```
# We shall pass the prcomp() to our data

data.pca <- prcomp(num_col[-2,-5,-7],center = TRUE, scale. = TRUE)
summary(data.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4       PC5       PC6
## Standard deviation     2.218 1.0008 0.9934 0.30010 1.964e-16 1.277e-16
## Proportion of Variance 0.703 0.1431 0.1410 0.01287 0.000e+00 0.000e+00
## Cumulative Proportion  0.703 0.8461 0.9871 1.00000 1.000e+00 1.000e+00
##                            PC7
## Standard deviation     2.503e-17
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

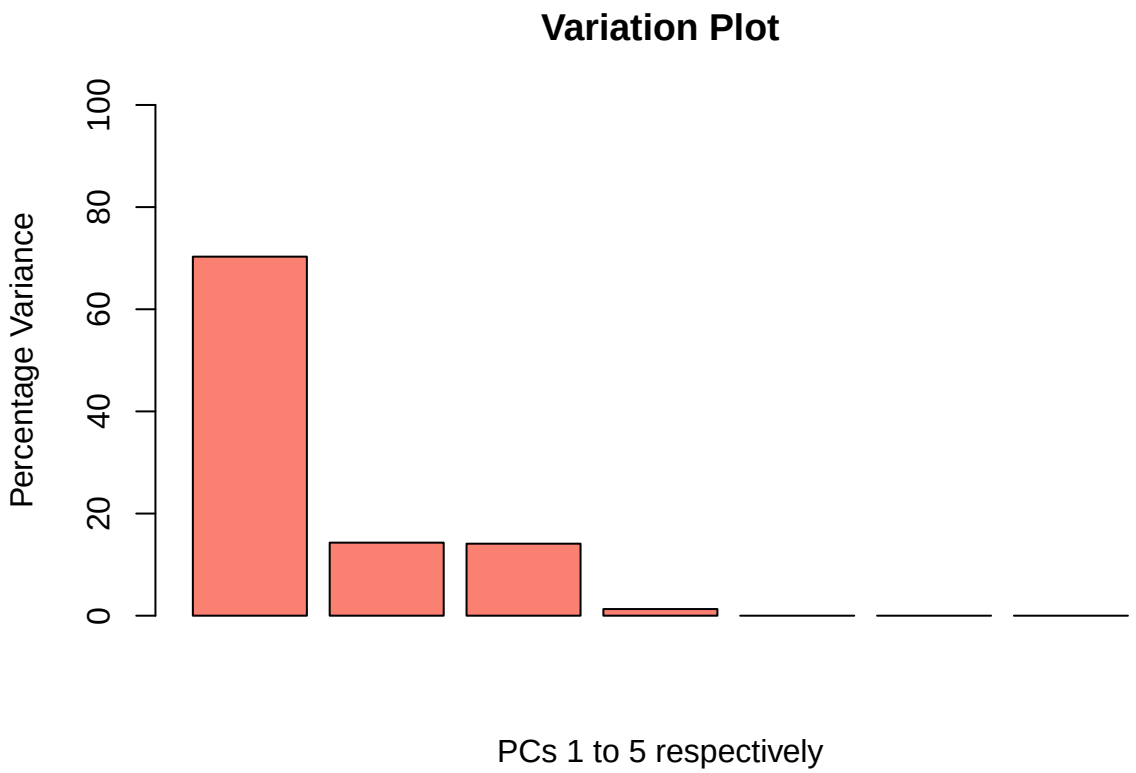
```r
# Checking the structure of PCAs

str(data.pca)
```

```
## List of 5
##  $ sdev    : num [1:7] 2.22 1.00 9.93e-01 3.00e-01 1.96e-16 ...
##  $ rotation: num [1:7, 1:7] -0.292 -0.325 -0.45 -0.45 -0.45 ...
##   ..- 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] 55.71 5.51 15.39 307.82 15.39 ...
##   ..- attr(*, "names")= chr [1:7] "Unit.price" "Quantity" "Tax" "cogs" ...
##  $ scale   : Named num [1:7] 26.48 2.92 11.71 234.18 11.71 ...
##   ..- attr(*, "names")= chr [1:7] "Unit.price" "Quantity" "Tax" "cogs" ...
##  $ x       : num [1:999, 1:7] -2.004 -0.184 -1.503 -2.796 -2.749 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:999] "1" "3" "4" "5" ...
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```r
# Plotting Variance vs PCs

plot(data.pca, type = "l" ,col = "red",main = "Variance against PCs")
```



```
variation_pca <- data.pca$sdev^2
var_percentage_pca <- round(variation_pca/sum(variation_pca) * 100, 1)
barplot(var_percentage_pca, main = "Variation Plot", xlab = "PCs 1 to 5 respectively", ylab = "Percentag
```




```
# Installing our ggbiplot visualisation package
#
library(devtools)
```

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

```
remotes::install_github('vqv/ggbiplot',force = TRUE)
```

```
## Downloading GitHub repo vqv/ggbiplot@HEAD
```

```
##
## * checking for file 'C:\Users\HP\AppData\Local\Temp\RtmpiMG6Vf\remotesc3c3382d80\vqv-ggbiplot-7325e8
## * preparing 'ggbiplot':
## * checking DESCRIPTION meta-information ... OK
## * checking for LF line-endings in source and make files and shell scripts
## * checking for empty or unneeded directories
## * looking to see if a 'data/datalist' file should be added
## * building 'ggbiplot_0.55.tar.gz'
##
```

```
## Installing package into 'C:/Users/HP/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
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
```

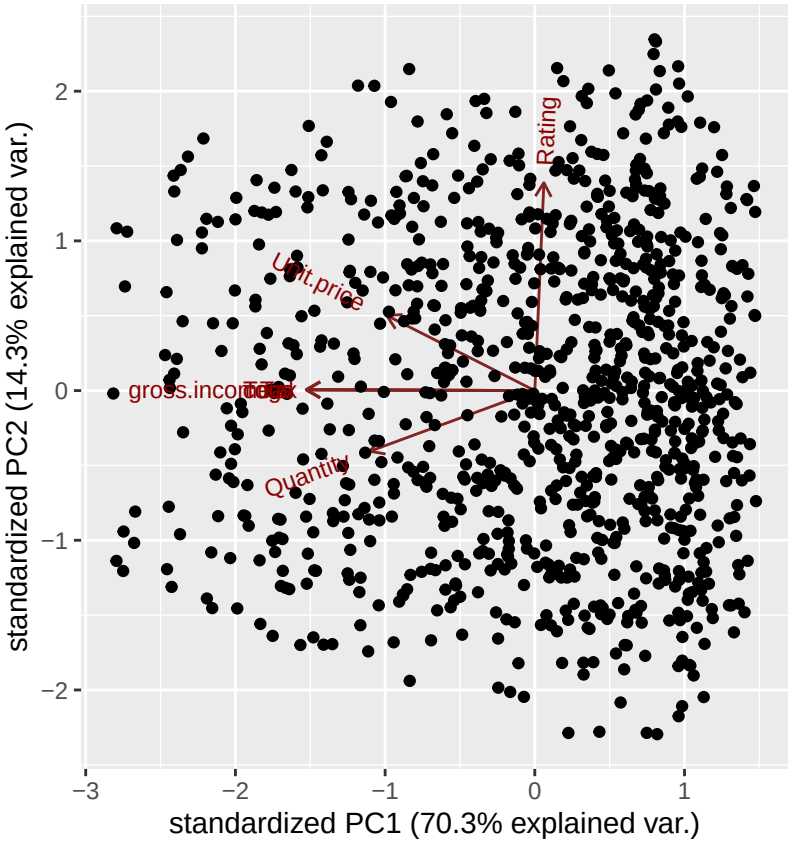
```
## Loading required package: scales
```

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

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

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

```
ggbiplot(data.pca)
```



b) Feature Selection

```
# Installing the FSelector package

suppressWarnings(
        suppressMessages(if
                         (!require(FSelector, quietly=TRUE))
                 install.packages("FSelector")))
library(FSelector)
```

```
library('caret')

correlationMatrix <- cor(num_col[-2,-5,-7])

# Checking the highly correlated attributes
highly_correlated <- findCorrelation(correlationMatrix, cutoff = .75)
highly_correlated
```

```
## [1] 3 5 4
```

Checking the Names of highly correlated

```
names(num_col[-2,-5,-7][, highly_correlated])
```

```
## [1] "Tax"          "gross.income" "cogs"
```

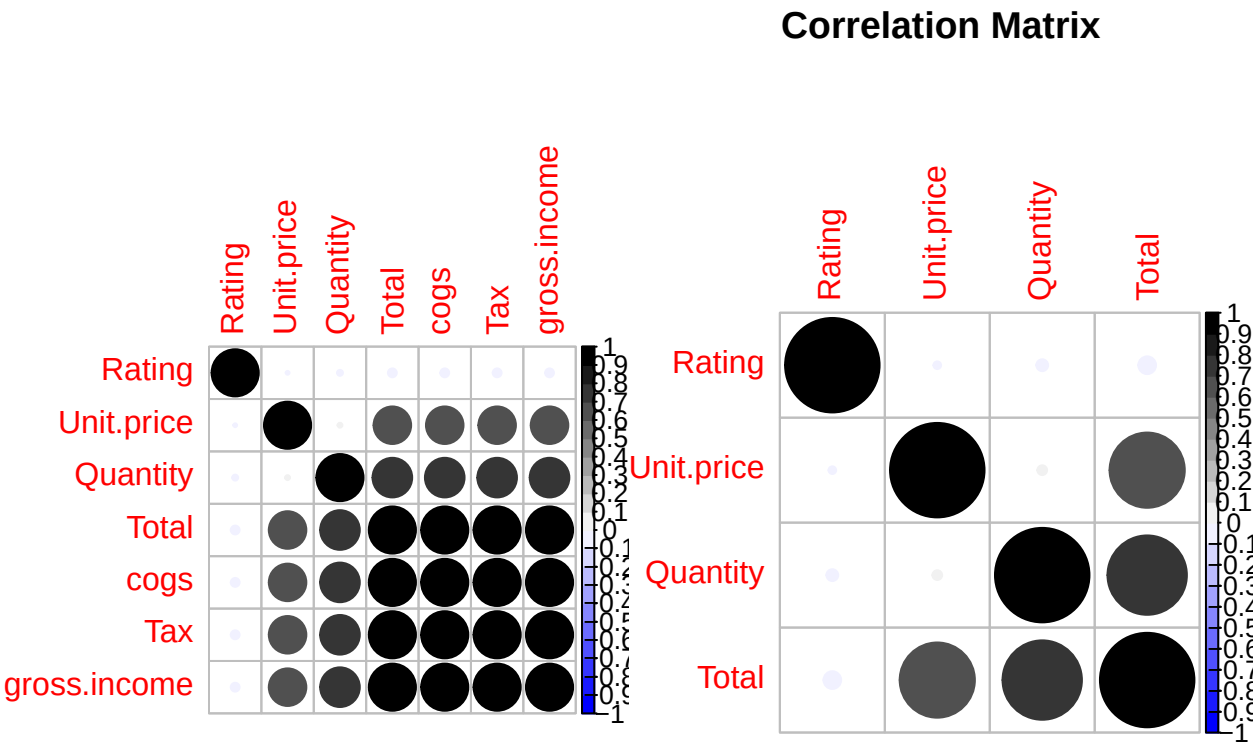
We can now remove features with high correlation

```
features_df <- num_col[-2,-5,-7][, -c(3,5,4)]
head(features_df)
```

```
##   Unit.price Quantity Rating    Total
## 1      74.69        7    9.1 548.9715
## 3      46.33        7    7.4 340.5255
## 4      58.22        8    8.4 489.0480
## 5      86.31        7    5.3 634.3785
## 6      85.39        7    4.1 627.6165
## 7      68.84        6    5.8 433.6920
```

Plotting correlation matrix to show comparison of relevant attributes

```
features_cor <- cor(features_df)

p.mat <- cor(features_cor)

par(mfrow = c(1, 2))

col<- colorRampPalette(c("blue", "white", "black"))(20)

corrplot(correlationMatrix, order = "hclust", col = col)

corrplot(cor(features_df), order = "hclust", title="Correlation Matrix", mar=c(0,0,1,0), col = col)
```



7. Conclusion

We have managed to obtain seven principal components, each which explain a percentage of the total variation of the dataset PC1 explains 70.3% of the total variance, which means that more than two-thirds of the information in the dataset (7 variables) can be encapsulated by just that one Principal Component. PC2 explains 14.3% of the variance.

Through our feature selection we have managed to select on four features i.e Unit.price, Quantity, Rating and Total.

8. Follow up questions

a) Did we have the right data?

Yes, the dataset available for this analysis was relevant to the research problem.

b) Do we need other data to answer the research question?

No, the dataset provided had relevant information for the research question.

c) Did we have the right question?

Yes, the research question was simple and specific enough.