Dimensionality Reduction

Brian Michira

9/10/2021

1.PROBLEM DEFINITION

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). Your project has been divided into four parts where you'll explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights.

2.Loading and previewing the dataset

```
# load our dataset
dataset <- read.csv("http://bit.ly/CarreFourDataset")
```

```
# preview the top of our dataset
head(dataset)
```

```
##    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
```

```
# preview the bottom of our dataset
tail(dataset)
```

```
##       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
```

```
# view the number of columns and rows
dim(dataset)
```

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

Our dataset has 1000 rows and 16 columns.

```
# checking the info
str(dataset)
```

```
## '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 ...
```

3.Cleaning the dataset

```
# checking for missing values
sum(is.na(dataset))
```

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

There is no missing values.

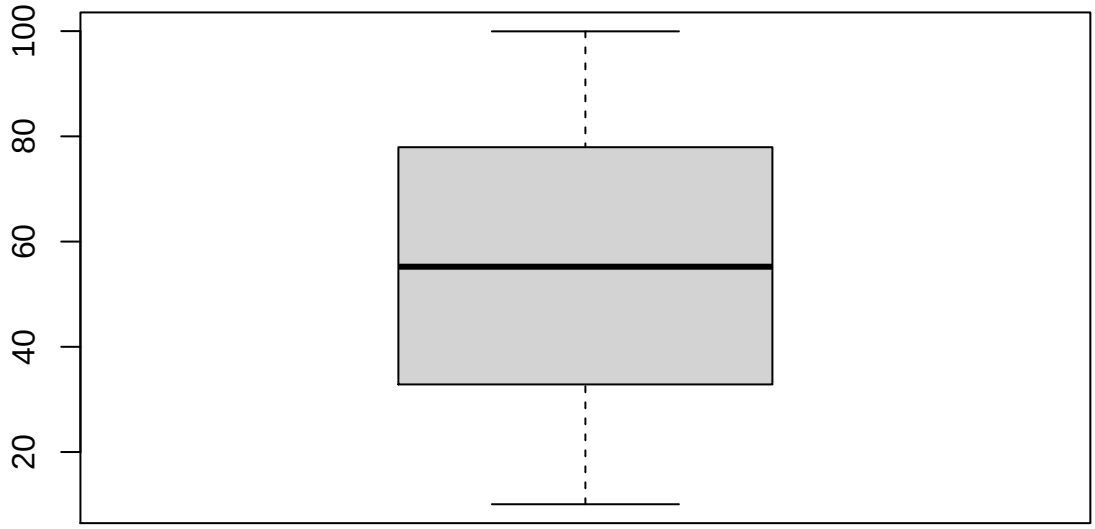
```
# find any duplicated rows in our dataset
duplicated_rows <- dataset[duplicated(dataset),]
duplicated_rows
```

```
##  [1] Invoice.ID              Branch                  Customer.type
##  [4] Gender                  Product.line            Unit.price
##  [7] Quantity                Tax                     Date
## [10] Time                    Payment                 cogs
## [13] gross.margin.percentage gross.income            Rating
## [16] Total
## <0 rows> (or 0-length row.names)
```

There appears to be no duplicated entries in our data.

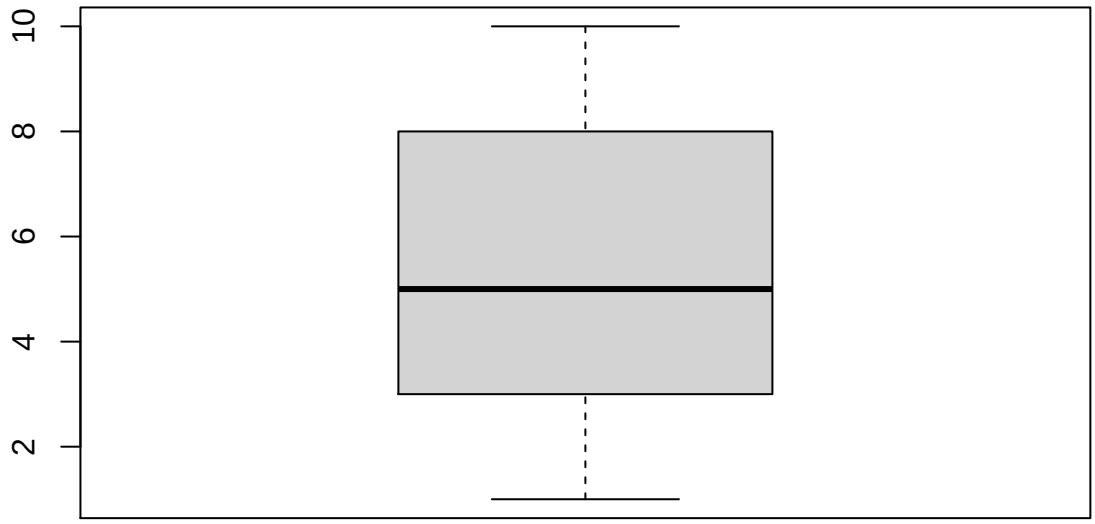
Handling Outliers

```
# checking for outliers on the Unit.price  column
boxplot(dataset$Unit.price)
```



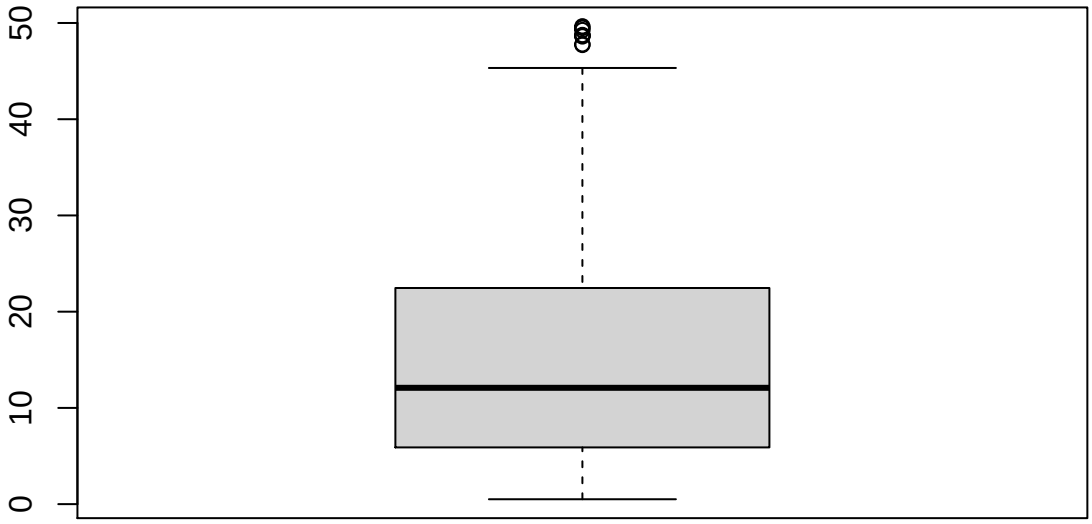
There is no presence of outliers in the Unit price column.

```
# checking for outliers on the Quantity column
boxplot(dataset$Quantity)
```



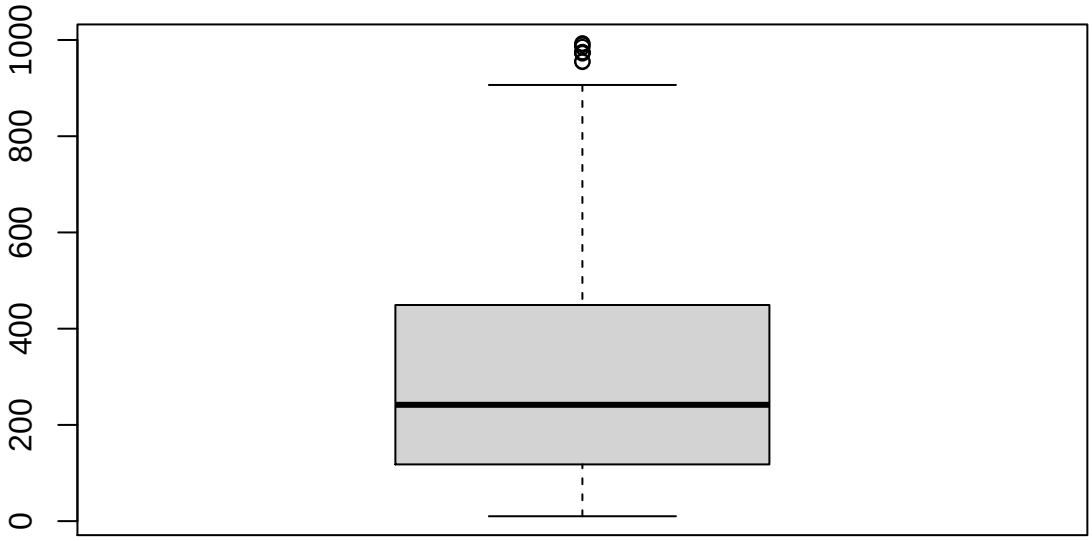
There appears to be no outliers in the quantity column.

```
# checking for outliers on the Tax column
boxplot(dataset$Tax)
```



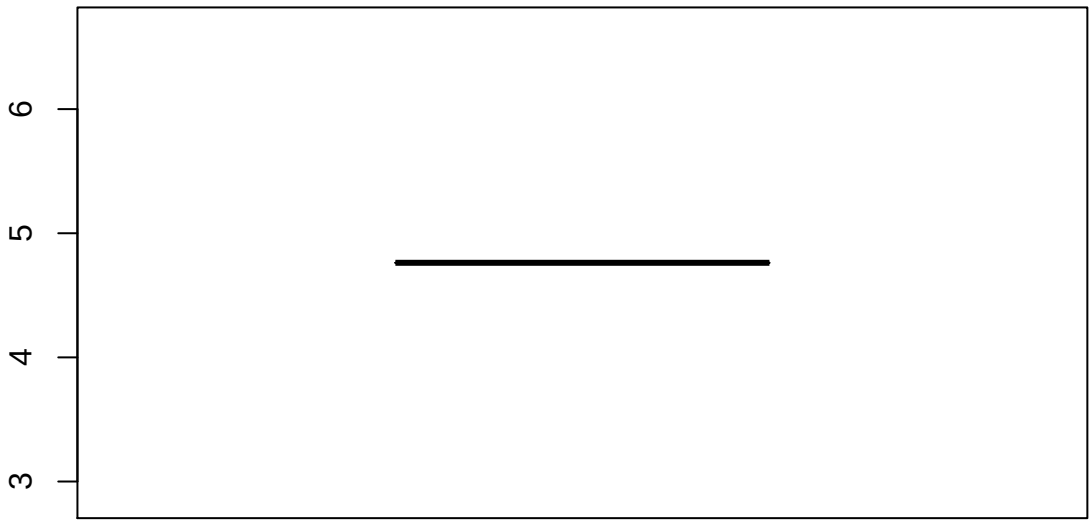
The outliers in the Tax column are as a result of the products that are taxed highly.

```
# checking for outliers on the cogs column
boxplot(dataset$cogs)
```



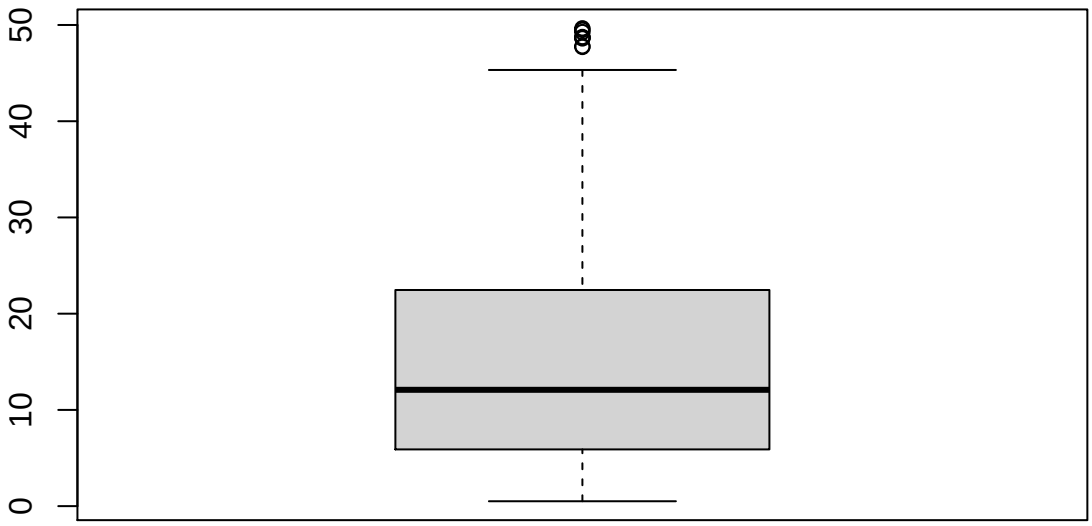
The outliers in the cogs column are as a result of the high cost of goods sold.

```
# checking for outliers on the gross.margin.percentage column
boxplot(dataset$gross.margin.percentage)
```



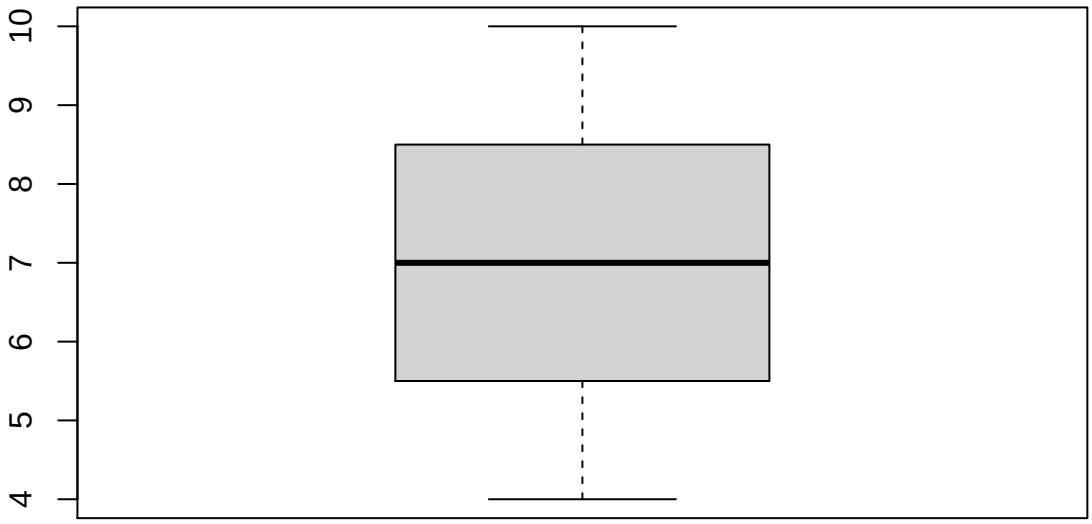
There appears to be no outliers in the gross.margin.percentage column.

```
# checking for outliers on the gross.income column
boxplot(dataset$gross.income)
```



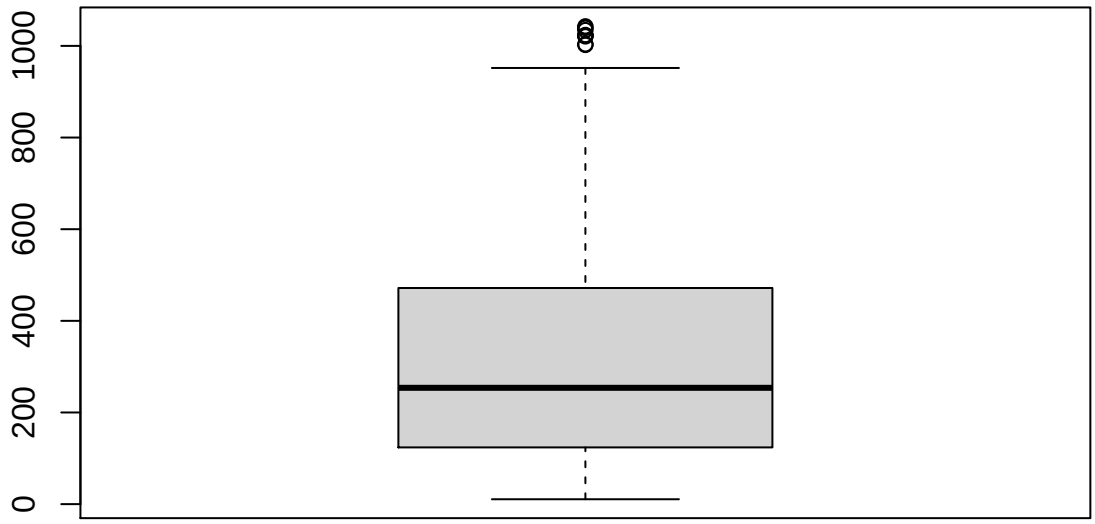
The outliers on the gross income column are as a result of the high gross income.

```
# checking for outliers on the Rating column
boxplot(dataset$Rating)
```



There is no presence of outliers in the Rating Column.

```
# checking for outliers on the Total column
boxplot(dataset$Total)
```



The outliers in the total Column are as a result of the high total amount.

4.Exploratory Data Analysis

Univariate Analysis

1.Measures of Central Tendency & Measures of Dispersion

```
summary(dataset)
```

```
##   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
```

```
### Identify numeric cols
num <- unlist(lapply(dataset, is.numeric))
col<- colnames(dataset[num])
col
```

```
## [1] "Unit.price"             "Quantity"
## [3] "Tax"                    "cogs"
## [5] "gross.margin.percentage" "gross.income"
## [7] "Rating"                 "Total"
```

```
num <-dataset[col]
head(num)
```

```
##   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
```

```
#measure of central tedency and measure of dispersion
library(moments)
univarite = function(x)list(
  Mean=mean(x, na.rm=TRUE),
  Median=median(x, na.rm=TRUE),
  Skewness=skewness(x, na.rm=TRUE),
  Kurtosis=kurtosis(x, na.rm=TRUE),
  Variance=var(x, na.rm=TRUE),
  Std.Dev=sd(x, na.rm=TRUE)
```

```
)
#calling the function
sapply(dataset[,c(col)], univarite)
```

```
##          Unit.price  Quantity   Tax       cogs      gross.margin.percentage
## Mean     55.67213    5.51       15.37937  307.5874  4.761905
## Median   55.23       5          12.088    241.76    4.761905
## Skewness 0.007066827 0.01292163 0.8912304 0.8912304 NaN
## Kurtosis 1.781499    1.784528   2.91253   2.91253   NaN
## Variance 701.9653    8.546446   137.0966  54838.64  0
## Std.Dev  26.49463    2.923431   11.70883  234.1765  0
##          gross.income Rating      Total
## Mean     15.37937     6.9727      322.9667
## Median   12.088       7           253.848
## Skewness 0.8912304    0.008996129 0.8912304
## Kurtosis 2.91253      1.848169    2.91253
## Variance 137.0966     2.953518    60459.6
## Std.Dev  11.70883     1.71858     245.8853
```

Univariate Graphical

```
head(dataset)
```

BarGraph

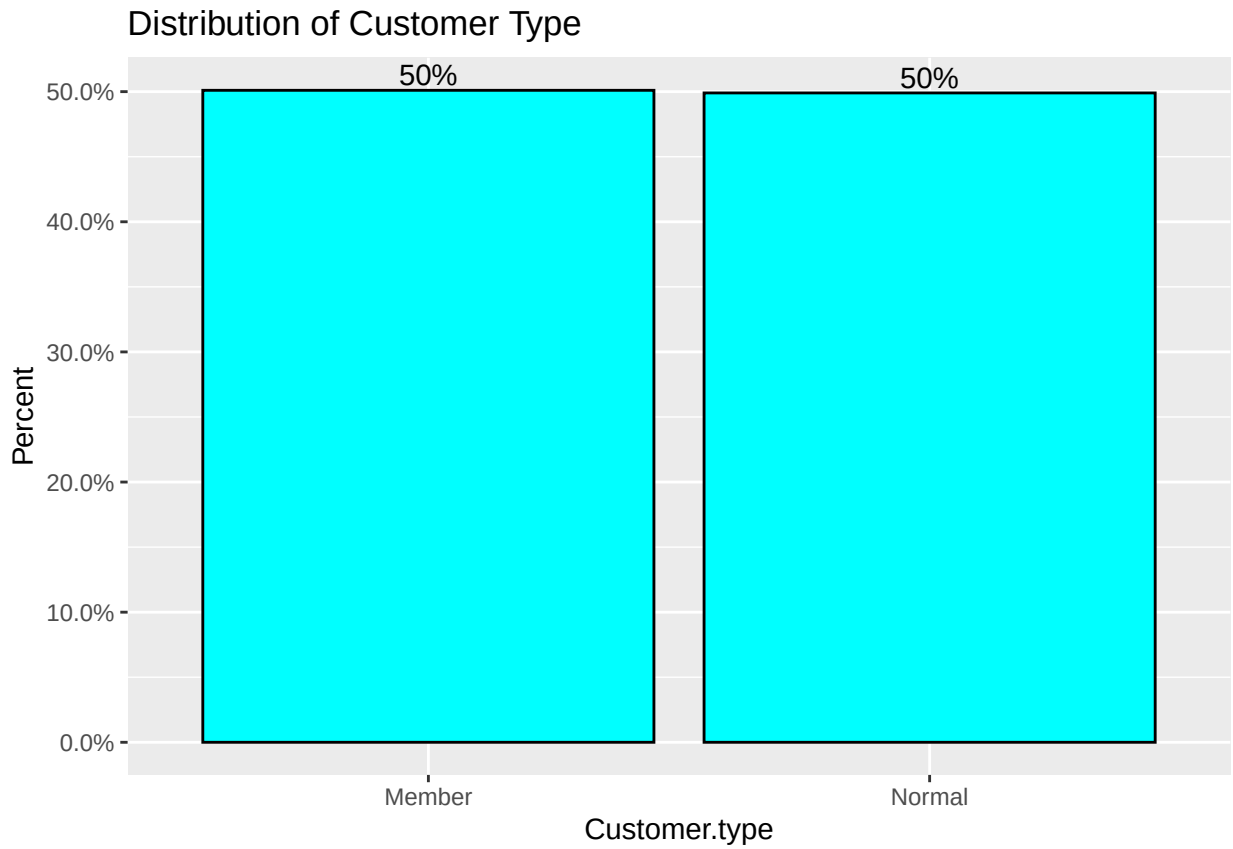
```
##    Invoice.ID Branch Customer.type Gender           Product.line Unit.price
## 1 750-67-8428      A        Member Female      Health and beauty      74.69
## 2 226-31-3081      C        Normal Female Electronic accessories      15.28
## 3 631-41-3108      A        Normal   Male     Home and lifestyle      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
```

```
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
```

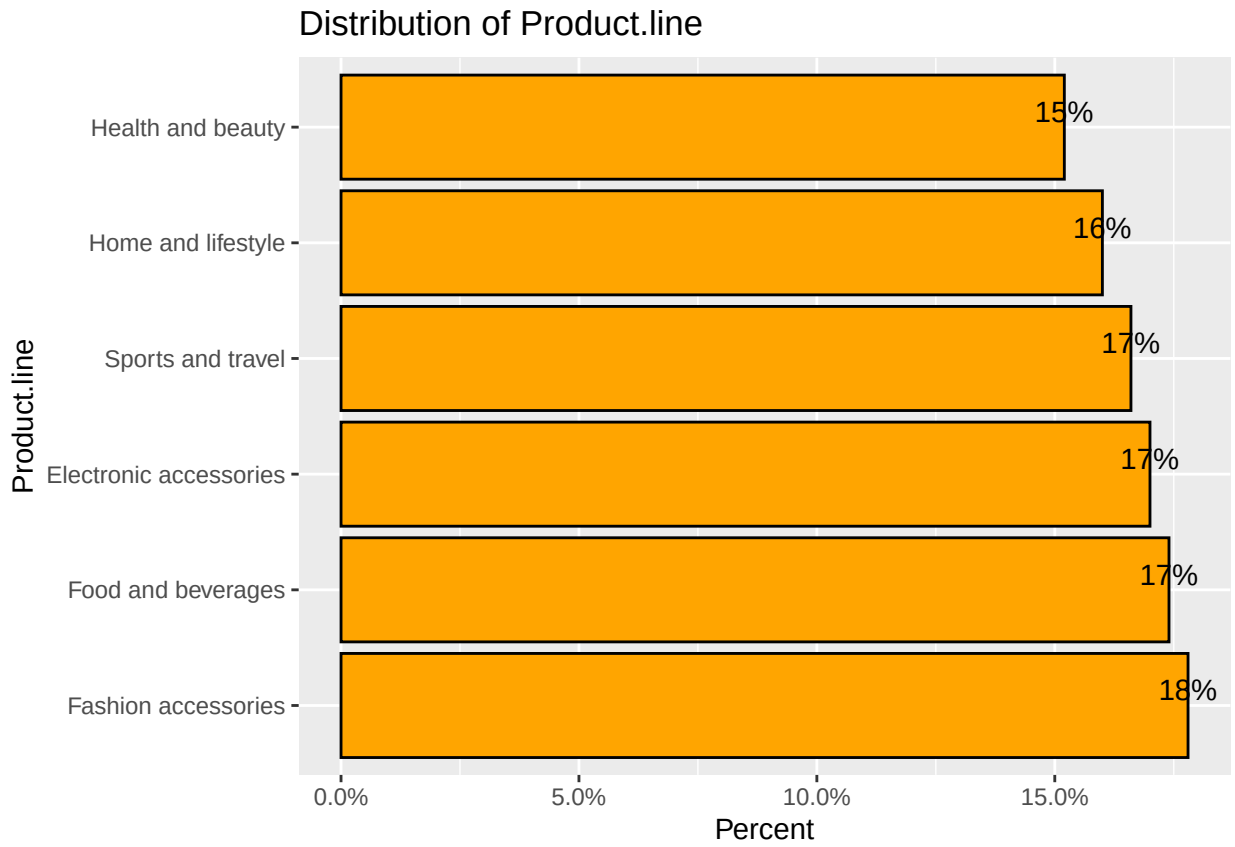
```
plotdata <- dataset %>%
  count(Payment) %>%
  mutate(pct = n / sum(n),
         pctlabel = paste0(round(pct*100), "%"))
library(scales)
library(ggplot2)
```

```
# Bar chart of Customer Type
plotdata <- dataset %>%
  count(Customer.type) %>%
  mutate(pct = n / sum(n),
         pctlabel = paste0(round(pct*100), "%"))

# plot the bars as percentages,
# in descending order with bar labels
ggplot(plotdata,
       aes(x = reorder(Customer.type, -pct),
           y = pct)) +
  geom_bar(stat = "identity",
           fill = "cyan",
           color = "black") +
  geom_text(aes(label = pctlabel),
            vjust = -0.25) +
  scale_y_continuous(labels = percent) +
  labs(x = "Customer.type",
       y = "Percent",
       title  = "Distribution of Customer Type")
```



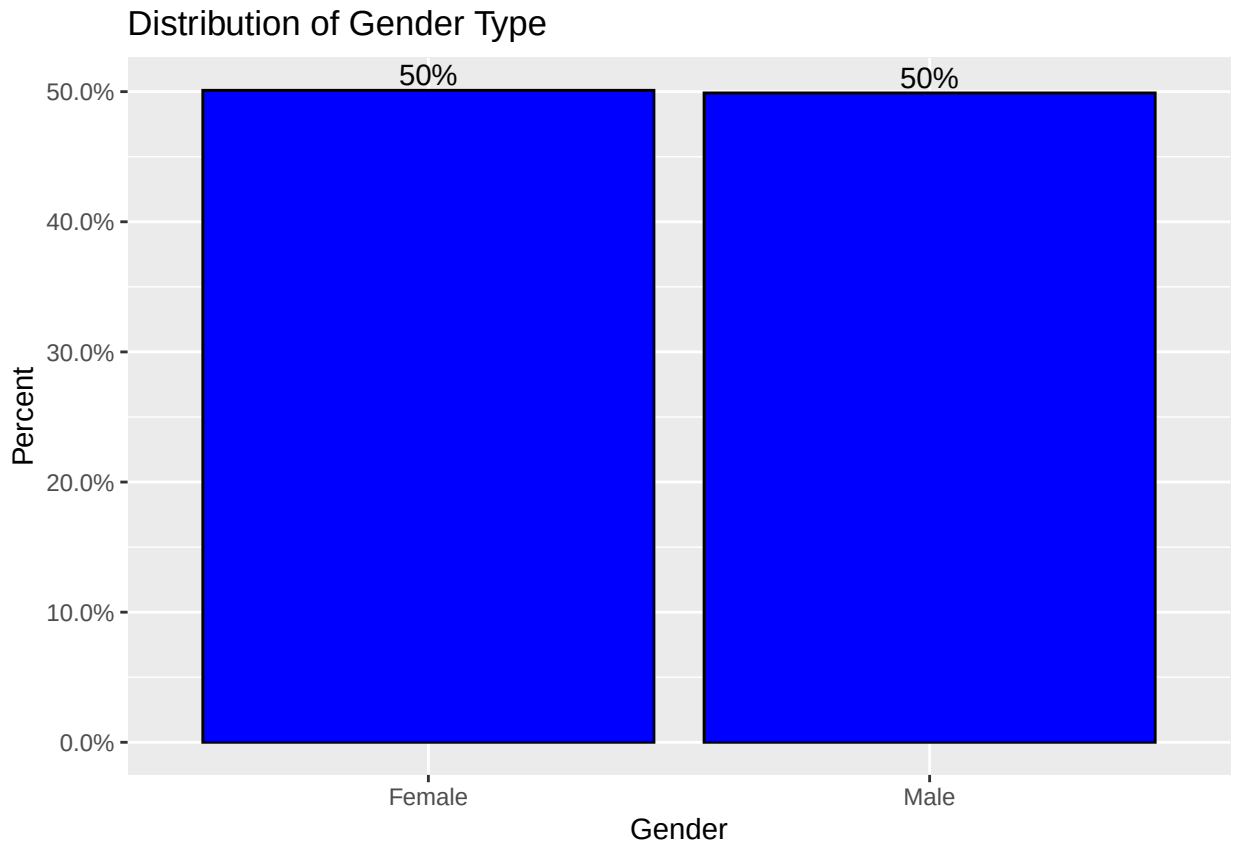
The customer type distribution was equal at 50% for the Member Customers and Normal Customers.

```
# Bar chart of Product Line
plotdata <- dataset %>%
  count(Product.line) %>%
  mutate(pct = n / sum(n),
         pctlabel = paste0(round(pct*100), "%"))

# plot the bars as percentages,
# in descending order with bar labels
ggplot(plotdata,
       aes(x = reorder(Product.line, -pct),
           y = pct)) +
  geom_bar(stat = "identity",
           fill = "orange",
           color = "black") +
  geom_text(aes(label = pctlabel),
            vjust = -0.25) +
  scale_y_continuous(labels = percent) +
  labs(x = "Product.line",
       y = "Percent",
       title  = "Distribution of Product.line") +
  coord_flip()
```



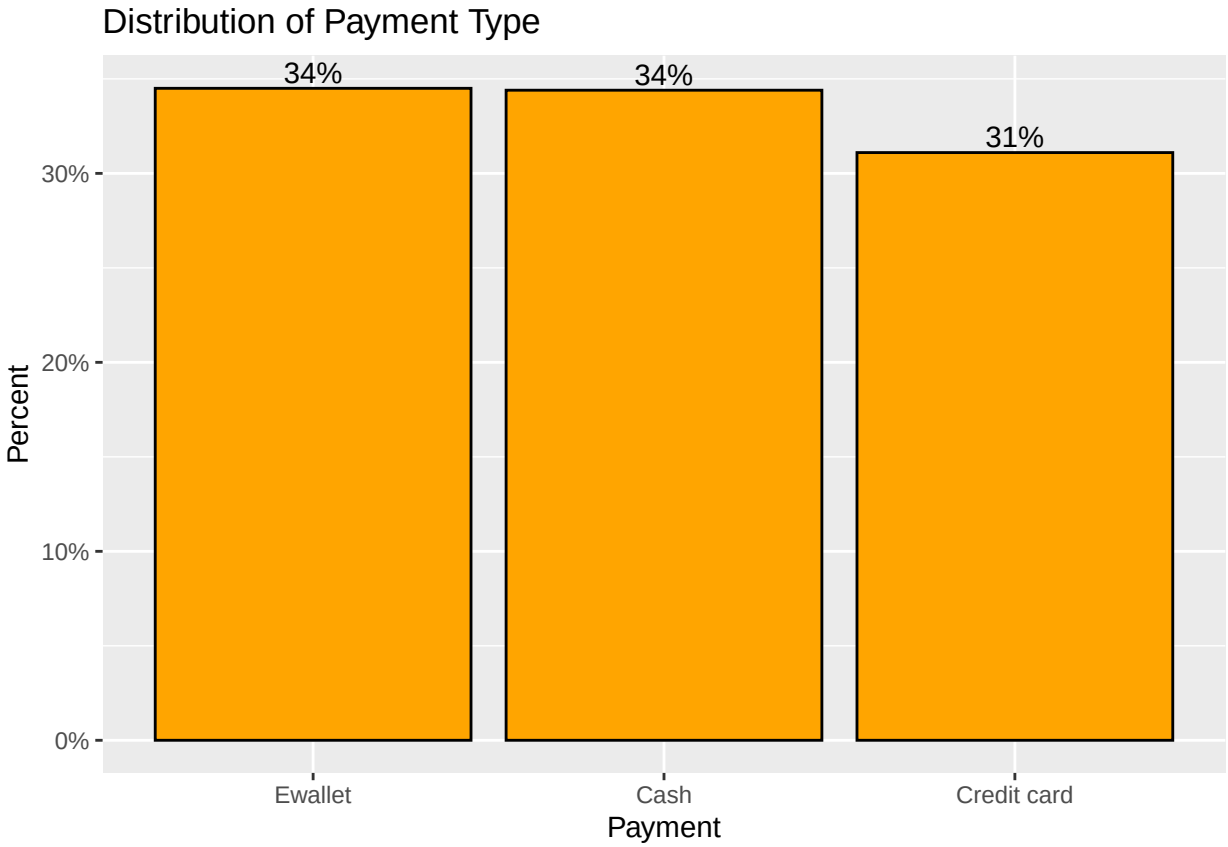
Fashion accessories had a higher distribution at 18% while Health and Beauty had a lower distribution at 15%

```
# Bar chart of Gender Distribution
plotdata <- dataset %>%
  count(Gender) %>%
  mutate(pct = n / sum(n),
         pctlabel = paste0(round(pct*100), "%"))

# plot the bars as percentages,
# in descending order with bar labels
ggplot(plotdata,
       aes(x = reorder(Gender, -pct),
           y = pct)) +
  geom_bar(stat = "identity",
           fill = "blue",
           color = "black") +
  geom_text(aes(label = pctlabel),
            vjust = -0.25) +
  scale_y_continuous(labels = percent) +
  labs(x = "Gender",
       y = "Percent",
       title  = "Distribution of Gender Type")
```



From our bar chart there appears to be an equal distribution of Males and Females.

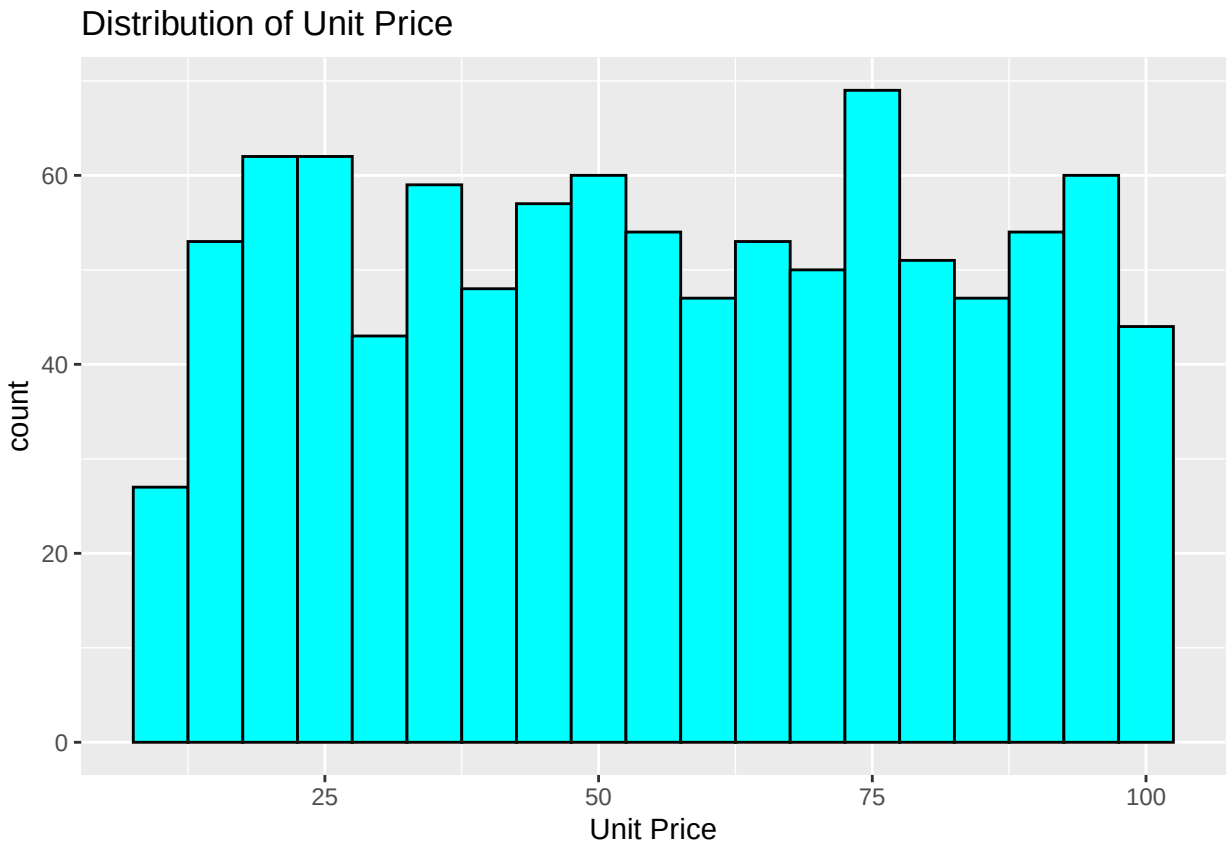
```
# Bar chart of Payment Distribution
plotdata <- dataset %>%
  count(Payment) %>%
  mutate(pct = n / sum(n),
         pctlabel = paste0(round(pct*100), "%"))

# plot the bars as percentages,
# in descending order with bar labels
ggplot(plotdata,
       aes(x = reorder(Payment, -pct),
           y = pct)) +
  geom_bar(stat = "identity",
           fill = "orange",
           color = "black") +
  geom_text(aes(label = pctlabel),
            vjust = -0.25) +
  scale_y_continuous(labels = percent) +
  labs(x = "Payment",
       y = "Percent",
       title  = "Distribution of Payment Type")
```



From our barchart the most common payment type is Ewallet and Cash Payment type represented by 34% and the least mode of payment is Credit card represented by 31%.

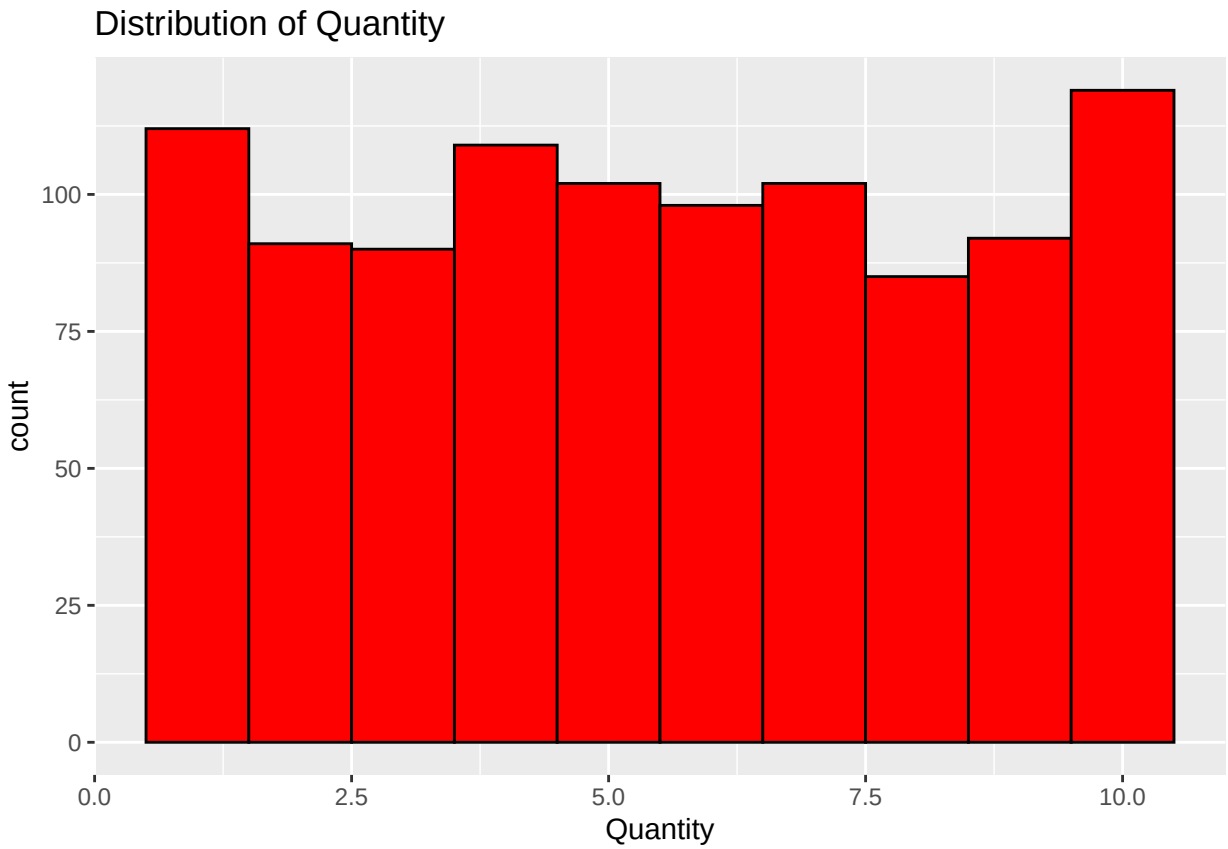
Histogram

```
# plot the histogram of Unit.price
ggplot(dataset, aes(x = Unit.price)) +
  geom_histogram(fill = "cyan",
                 color = "black",
                 binwidth = 5) +
  labs(title="Distribution of Unit Price",
       x = "Unit Price")
```



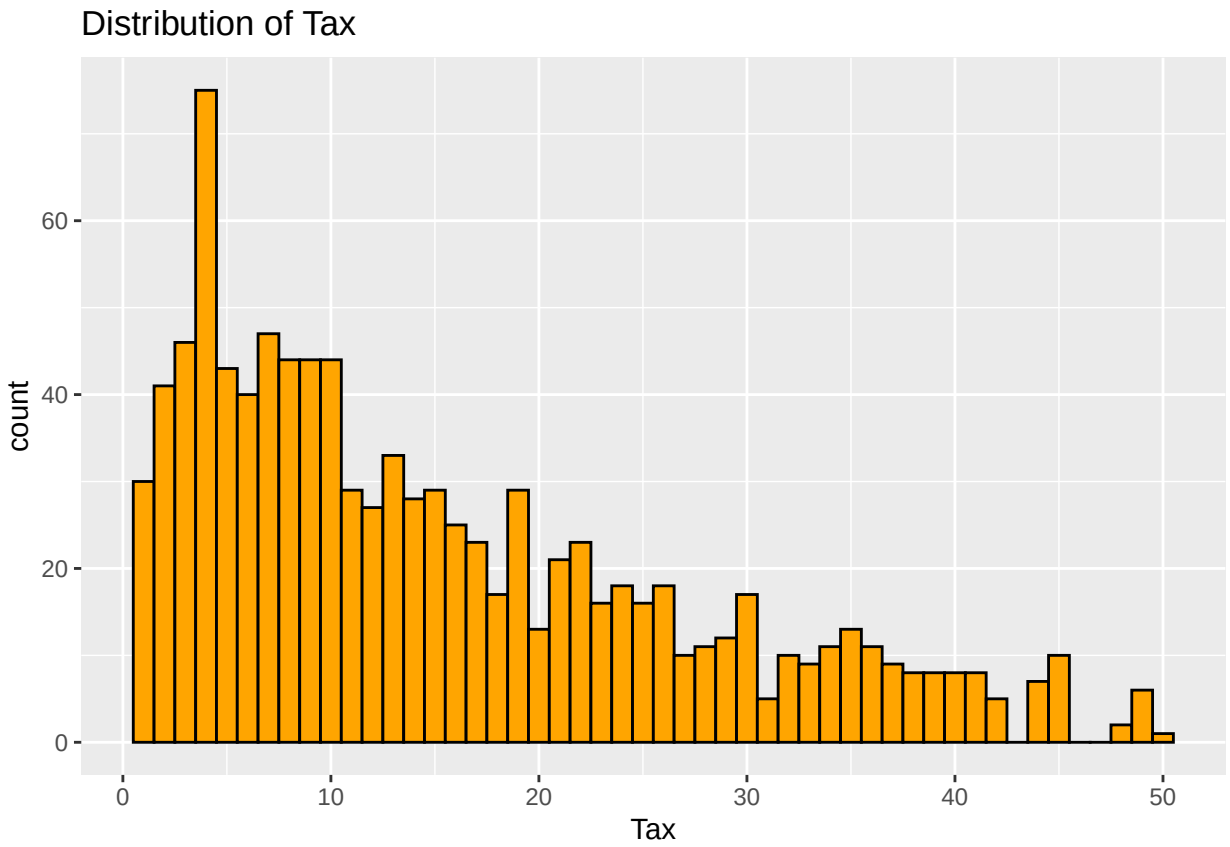
From our histogram we can see that the unit price of 75 had among the highest count.

```
# plot the histogram of Quantity
ggplot(dataset, aes(x = Quantity)) +
  geom_histogram(fill = "red",
                 color = "black",
                 binwidth = 1) +
  labs(title="Distribution of Quantity",
       x = "Quantity")
```



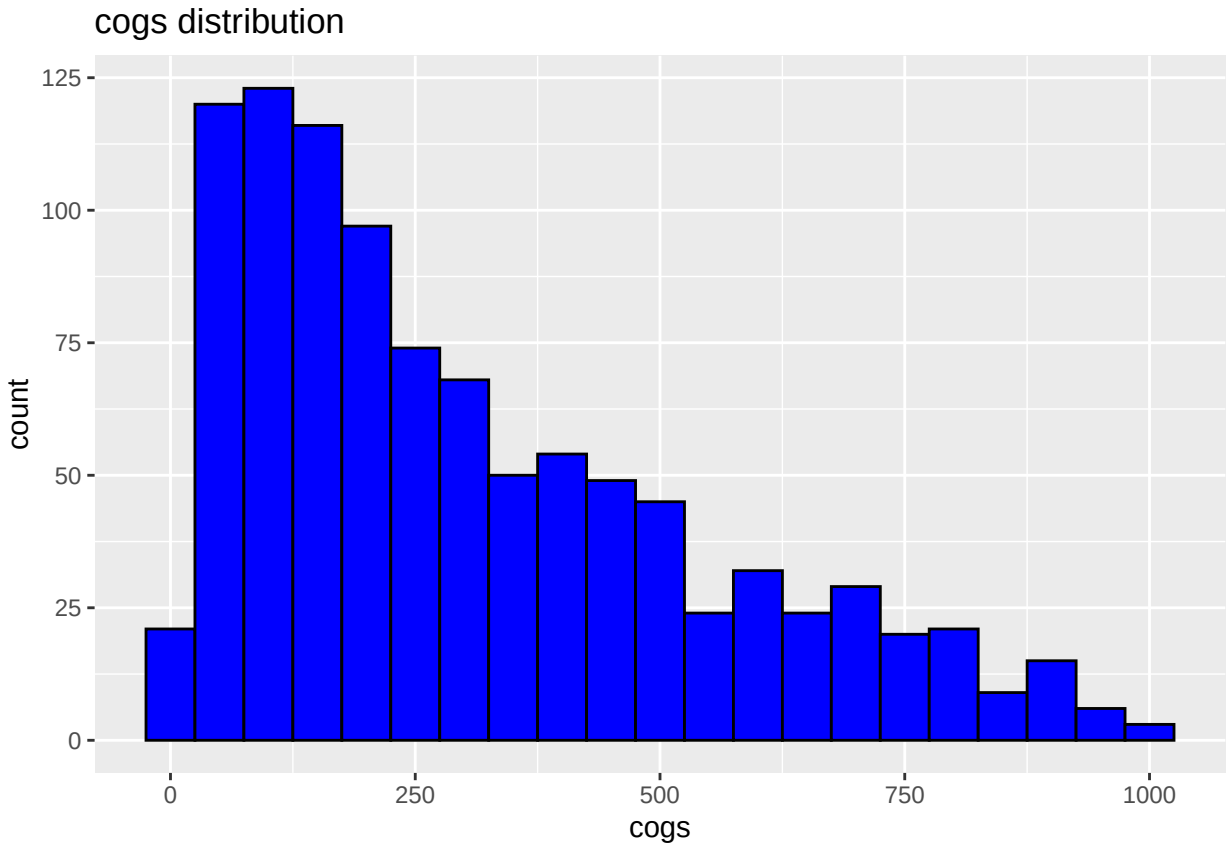
From our histogram of Quantity distribution the quantity of 10 had the highest count.

```
# plot the histogram of Tax
ggplot(dataset, aes(x = Tax)) +
  geom_histogram(fill = "orange",
                 color = "black",
                 binwidth = 1) +
  labs(title="Distribution of Tax",
       x = "Tax")
```



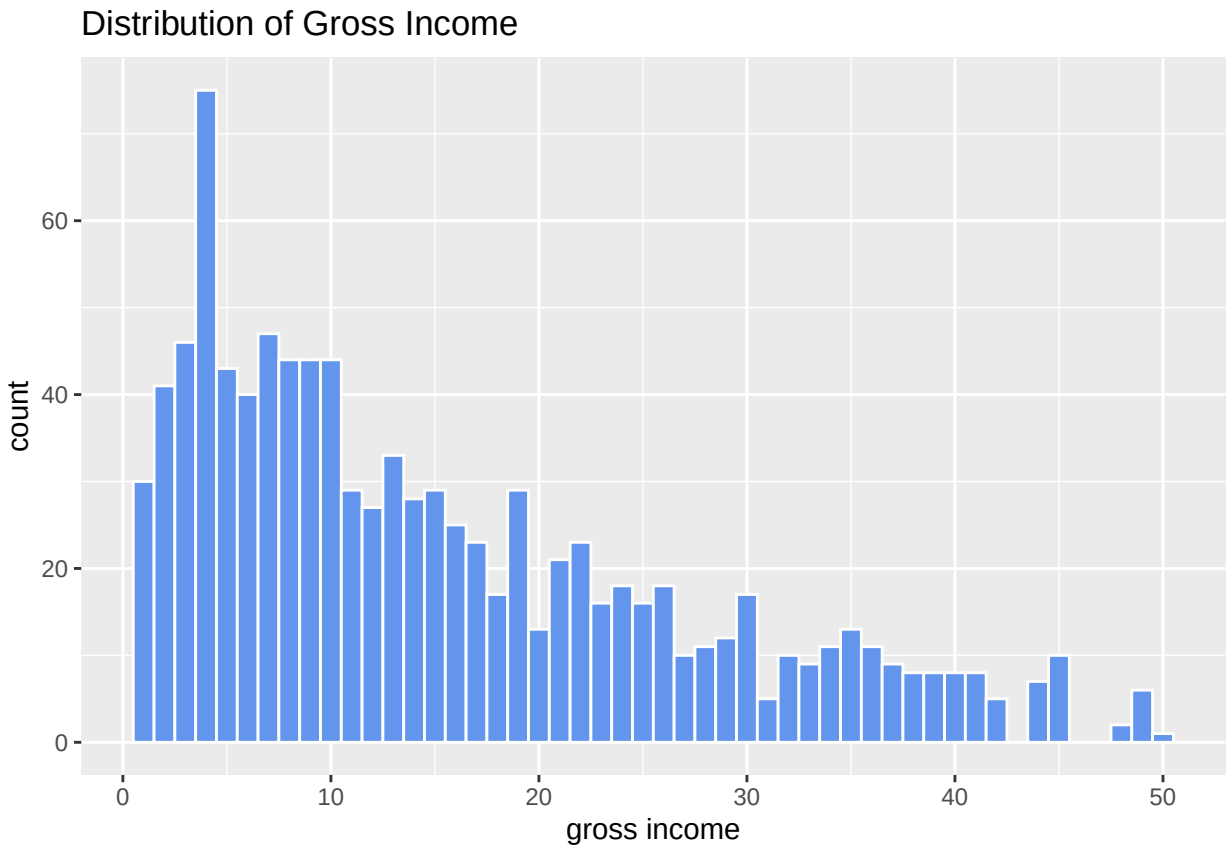
From our histogram tax rates was positively skewed.Showing that most of the values were greater than the mean.

```
# plot the histogram of cogs
ggplot(dataset, aes(x = cogs)) +
  geom_histogram(fill = "blue",
                 color = "black",
                 binwidth = 50) +
  labs(title="cogs distribution",
       x = "cogs")
```



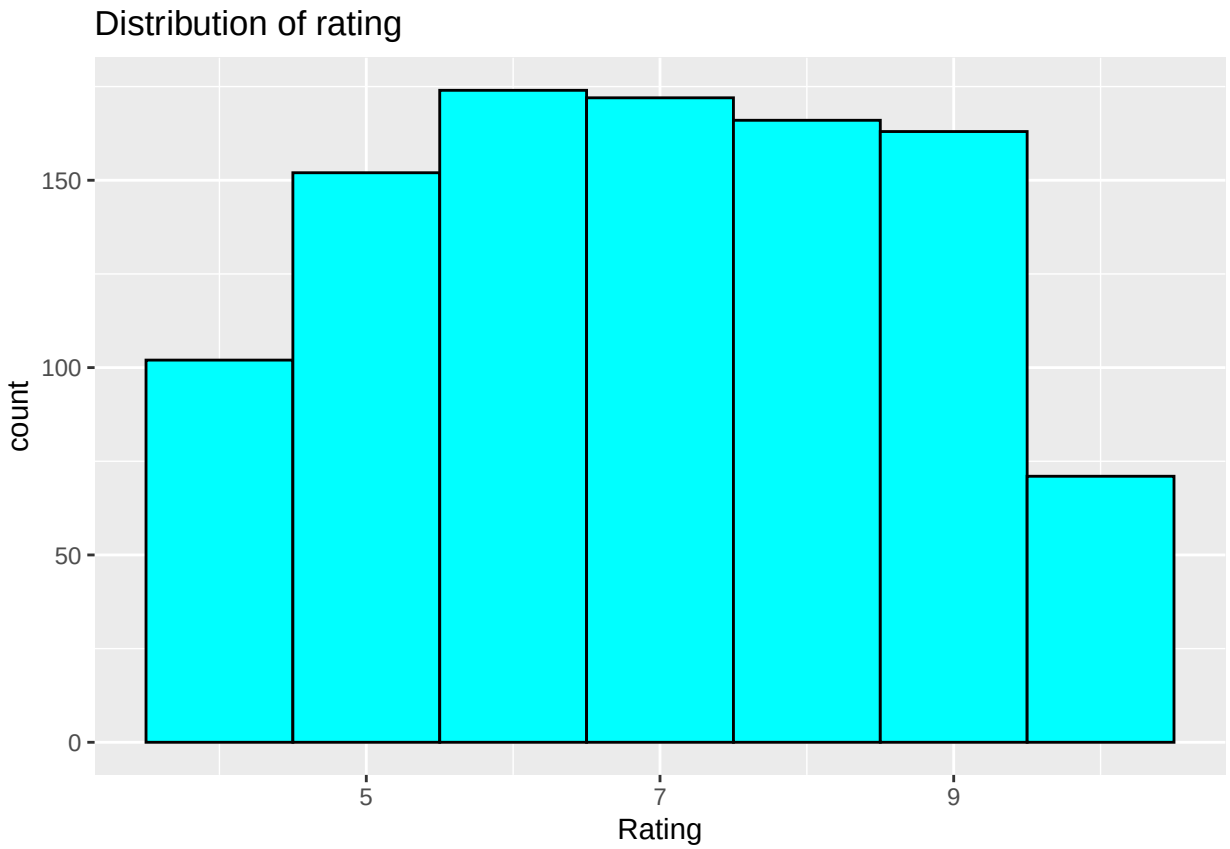
From our histogram cogs was positively skewed.Showing that most of the values were greater than the mean.

```
# plot the histogram of gross.income
ggplot(dataset, aes(x = gross.income)) +
  geom_histogram(fill = "cornflowerblue",
                 color = "white",
                 binwidth = 1) +
  labs(title="Distribution of Gross Income",
       x = "gross income")
```



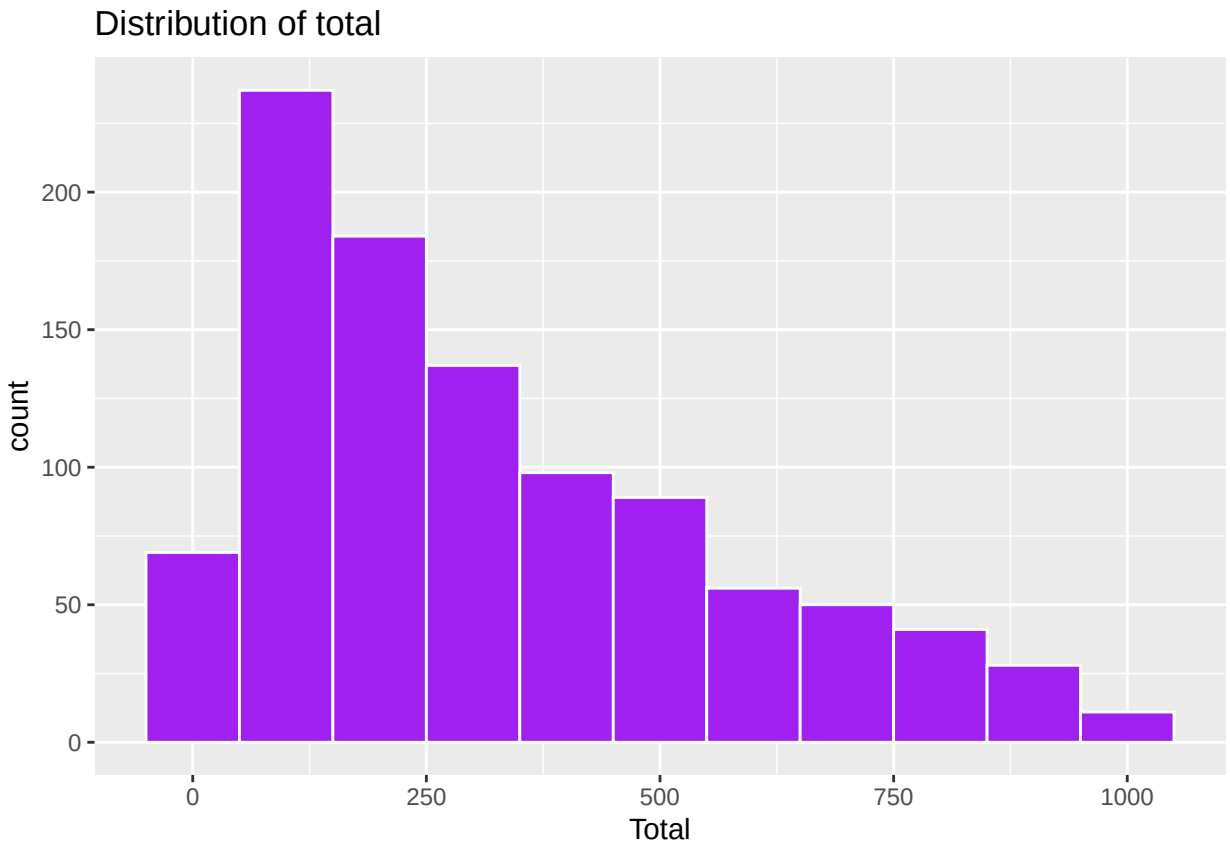
From our histogram gross income was positively skewed.Showing that most of the values were greater than the mean.

```
# plot the histogram of Rating
ggplot(dataset, aes(x = Rating)) +
  geom_histogram(fill = "cyan",
                 color = "black",
                 binwidth = 1) +
  labs(title="Distribution of rating",
       x = "Rating")
```



From our histogram of Rating it appeared to have a high count between 6 and 7.

```
# plot the histogram of Total
ggplot(dataset, aes(x = Total)) +
  geom_histogram(fill = "purple",
                 color = "white",
                 binwidth = 100) +
  labs(title="Distribution of total",
       x = "Total")
```



From our histogram total distribution is positively skewed.

Bivariate Analysis

###Correlation

```
#correlation matrix
correlation = cor(dataset[,c(6,7,8,12,14,15,16)])
correlation
```

```
##                Unit.price    Quantity        Tax       cogs gross.income
## Unit.price    1.000000000  0.01077756  0.6339621  0.6339621    0.6339621
## Quantity      0.010777564  1.00000000  0.7055102  0.7055102    0.7055102
## Tax           0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## cogs          0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## gross.income  0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## Rating       -0.008777507 -0.01581490 -0.0364417 -0.0364417   -0.0364417
## Total         0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
##                     Rating      Total
## Unit.price   -0.008777507  0.6339621
## Quantity     -0.015814905  0.7055102
## Tax          -0.036441705  1.0000000
## cogs         -0.036441705  1.0000000
## gross.income -0.036441705  1.0000000
## Rating        1.000000000 -0.0364417
## Total        -0.036441705  1.0000000
```

Part 1.Dimensionality Reduction

1.PCA

```
head(num)
```

```
##   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
```

```
tail(num)
```

```
##      Unit.price Quantity     Tax   cogs gross.margin.percentage gross.income
## 995       60.95        1  3.0475  60.95                4.761905       3.0475
## 996       40.35        1  2.0175  40.35                4.761905       2.0175
## 997       97.38       10 48.6900 973.80                4.761905      48.6900
## 998       31.84        1  1.5920  31.84                4.761905       1.5920
## 999       65.82        1  3.2910  65.82                4.761905       3.2910
## 1000      88.34        7 30.9190 618.38                4.761905      30.9190
##      Rating     Total
## 995     5.9   63.9975
## 996     6.2   42.3675
## 997     4.4 1022.4900
## 998     7.7   33.4320
## 999     4.1   69.1110
## 1000    6.6  649.2990
```

```
#Removing the gross.margin percentage column since it has the same value
df<-num[,-5]
head(df)
```

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

```
df.pca <- prcomp(df, center = TRUE, scale. = TRUE)
summary(df.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4       PC5       PC6
## Standard deviation     2.2185 1.0002 0.9939 0.30001 2.981e-16 1.493e-16
## Proportion of Variance 0.7031 0.1429 0.1411 0.01286 0.000e+00 0.000e+00
## Cumulative Proportion  0.7031 0.8460 0.9871 1.00000 1.000e+00 1.000e+00
##                            PC7
## Standard deviation     9.831e-17
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

As a result we obtain 7 principal components, each which explain a percentate of the total variation of the dataset.

PC1 explains 70.31% of the total variance, which means that more information from the dataset can be dervied from just that one Principal Component.

PC2 explains 14.29% of the variance.

```
#checking the structure
str(df.pca)
```

```
## List of 5
##  $ sdev    : num [1:7] 2.22 1.00 9.94e-01 3.00e-01 2.98e-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] 55.67 5.51 15.38 307.59 15.38 ...
##   ..- attr(*, "names")= chr [1:7] "Unit.price" "Quantity" "Tax" "cogs" ...
##  $ scale   : Named num [1:7] 26.49 2.92 11.71 234.18 11.71 ...
##   ..- attr(*, "names")= chr [1:7] "Unit.price" "Quantity" "Tax" "cogs" ...
##  $ x       : num [1:1000, 1:7] -2.005 2.306 -0.186 -1.504 -2.8 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

#Plot

```
#Plotting the pca
library(devtools)
```

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

```
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
```

```
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)
```

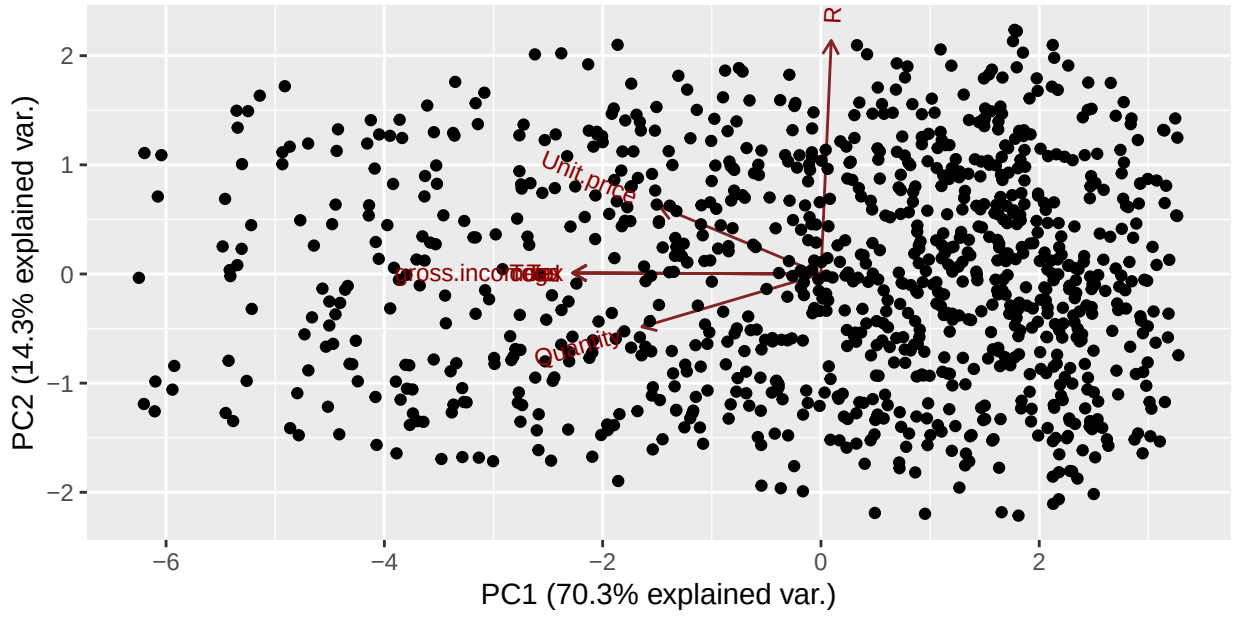
```
## ------------------------------------------------------------------------------
```

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

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

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

```
ggbiplot(df.pca, obs.scale = 1, var.scale = 1)
```



Conclusion

Quantity, Rating, Unit Price and Gross income are the most important features in this analysis. Marketing team when adversting their products should consider quality of the product, unit price, rating of the products and the gross income of their consumers.

Part 2.Feature Selection

i)Filter Methods

```
#selecting the numerical variables
num <- unlist(lapply(dataset, is.numeric))
col<- colnames(dataset[num])
col
```

```
## [1] "Unit.price"              "Quantity"
## [3] "Tax"                     "cogs"
## [5] "gross.margin.percentage" "gross.income"
## [7] "Rating"                  "Total"
```

```
num <-dataset[col]
head(num)
```

```
##   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
```

```
num<-num[-5]
head(num)
```

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

```
# Installing and loading our caret and corrplot package
# ---
#
suppressWarnings(
        suppressMessages(if
                         (!require(caret, quietly=TRUE))
                 install.packages("caret")))
library(caret)
```

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

```
# Calculating the correlation matrix
corr<- cor(num)
corr
```

```
##                Unit.price    Quantity        Tax       cogs gross.income
## Unit.price    1.000000000  0.01077756  0.6339621  0.6339621    0.6339621
## Quantity      0.010777564  1.00000000  0.7055102  0.7055102    0.7055102
## Tax           0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## cogs          0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## gross.income  0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## Rating       -0.008777507 -0.01581490 -0.0364417 -0.0364417   -0.0364417
## Total         0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
##                     Rating      Total
## Unit.price   -0.008777507  0.6339621
## Quantity     -0.015814905  0.7055102
## Tax          -0.036441705  1.0000000
## cogs         -0.036441705  1.0000000
## gross.income -0.036441705  1.0000000
## Rating        1.000000000 -0.0364417
## Total        -0.036441705  1.0000000
```

```
# Find attributes that are highly correlated
# ---
#
highlyCorr <- findCorrelation(corr, cutoff=0.75)
highlyCorr
```

```
## [1] 4 7 3
```

```
names(num[,highlyCorr])
```

```
## [1] "cogs"  "Total" "Tax"
```

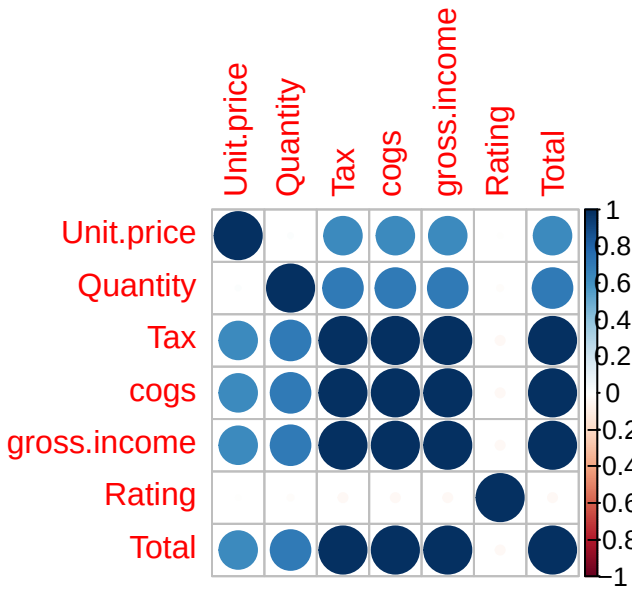
Cogs,Total and Tax are highly correlated

```
# We can remove the variables with a higher correlation
df<-num[-highlyCorr]
head(df)
```

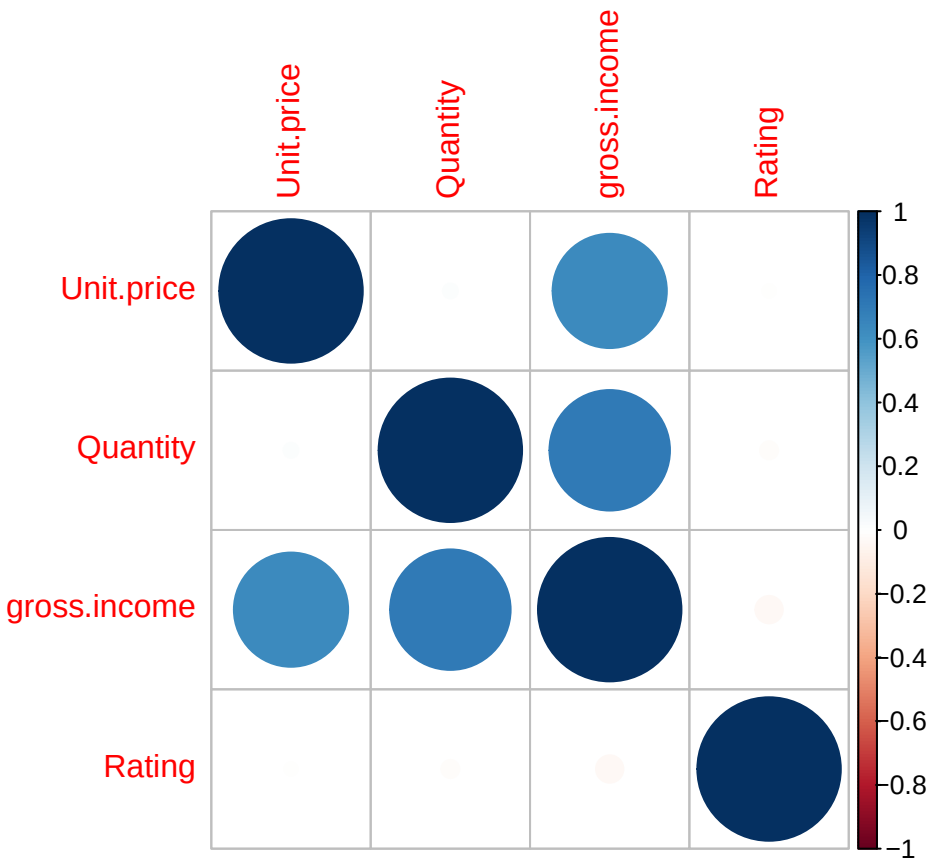
```
##   Unit.price Quantity gross.income Rating
## 1      74.69        7      26.1415    9.1
## 2      15.28        5       3.8200    9.6
## 3      46.33        7      16.2155    7.4
## 4      58.22        8      23.2880    8.4
## 5      86.31        7      30.2085    5.3
## 6      85.39        7      29.8865    4.1
```

After filtering the highly correlated variables we remain with unit price,Quantity,Gross income and Rating as the important features.

```
#Graphical comparison
par(mfrow = c(1, 2))
#Before removing the highly correlated features
corrplot(cor(num))
```




```
#Afer removing the highly correlated features
corrplot(cor(df))
```



ii)Wrapper Methods

```
# Installing and loading our clustvarsel and mclust package
suppressWarnings(
        suppressMessages(if
                         (!require(clustvarsel, quietly=TRUE))
                 install.packages("clustvarsel")))

library(clustvarsel)
```

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

```
#Normalize the data
library(dplyr)
df.norm<-as.data.frame(scale(num))
head(df.norm)
```

```
##    Unit.price   Quantity         Tax        cogs gross.income     Rating
## 1  0.71780097  0.5096752  0.91914693  0.91914693   0.91914693  1.2378240
## 2 -1.52454035 -0.1744526 -0.98723557 -0.98723557  -0.98723557  1.5287619
## 3 -0.35260468  0.5096752  0.07141032  0.07141032   0.07141032  0.2486355
## 4  0.09616553  0.8517391  0.67544187  0.67544187   0.67544187  0.8305111
```

```
## 5  1.15638044  0.5096752  1.26649176  1.26649176   1.26649176 -0.9733034
## 6  1.12165642  0.5096752  1.23899114  1.23899114   1.23899114 -1.6715541
##         Total
## 1  0.91914693
## 2 -0.98723557
## 3  0.07141032
## 4  0.67544187
## 5  1.26649176
## 6  1.23899114
```

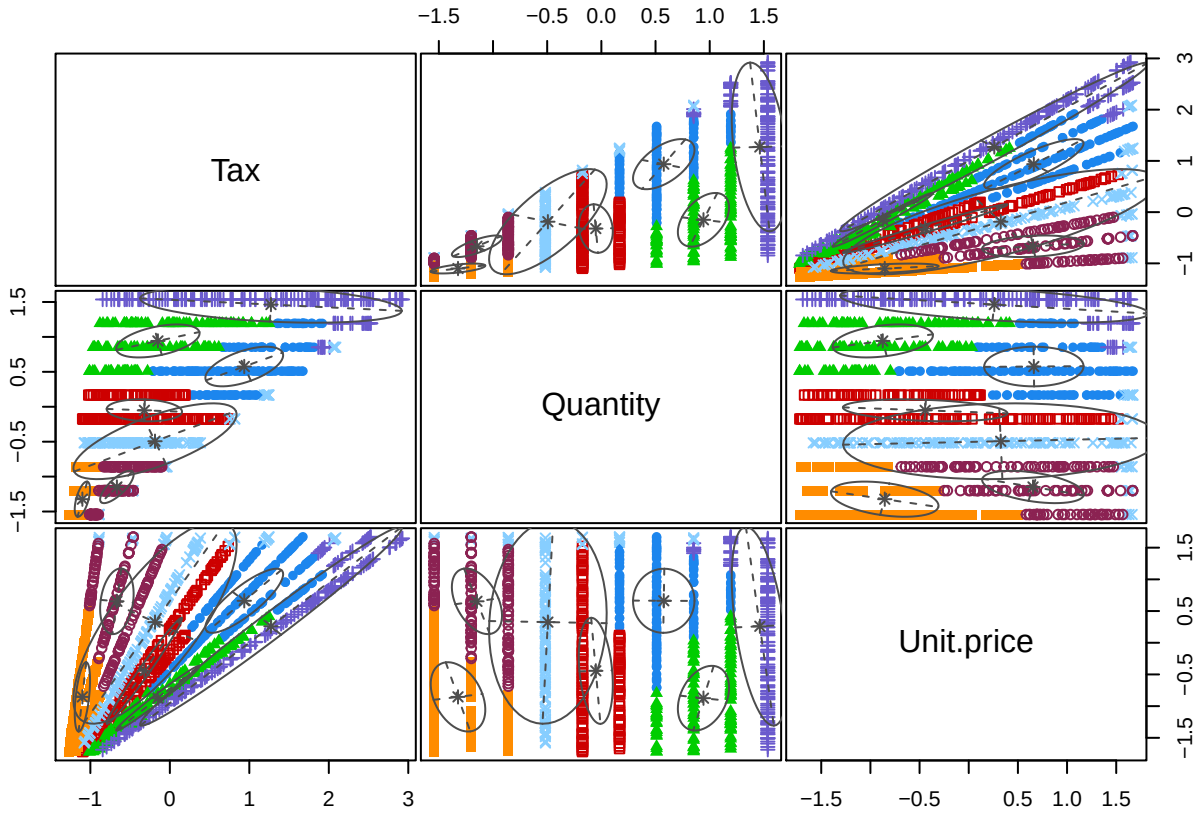
```
#Sequential forward greedy search:
out = clustvarsel(df.norm, G = 1:7)
out
```

```
## ----------------------------------------------------
## Variable selection for Gaussian model-based clustering
## Stepwise (forward/backward) greedy search
## ----------------------------------------------------
##
##  Variable proposed Type of step  BICclust Model G   BICdiff Decision
##                Tax          Add -2460.877     V 4  389.8147 Accepted
##           Quantity          Add -3599.179   VEV 7 1030.6521 Accepted
##         Unit.price          Add -1707.881   EVV 7 3236.0143 Accepted
##         Unit.price       Remove -3599.179   VEV 7 3236.0143 Rejected
##             Rating          Add -4815.681   EVV 7 -257.1079 Rejected
##         Unit.price       Remove -3599.179   VEV 7 3236.0143 Rejected
##
## Selected subset: Tax, Quantity, Unit.price
```

```
# Clustering model

Subset1 = df.norm[,out$subset]
mod = Mclust(Subset1, G = 1:7)
summary(mod)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVV (ellipsoidal, equal volume) model with 7 components:
##
##  log-likelihood    n df       BIC       ICL
##       -636.3461 1000 63 -1707.881 -1750.383
##
## Clustering table:
##   1   2   3   4   5   6   7
## 165 154 120 144 132 139 146
```

```
plot(mod,c("classification"))
```



iii)Embedded Methods

```
#Installing and loading our wskm and cluster package
suppressWarnings(
        suppressMessages(if
                         (!require(wskm, quietly=TRUE))
                 install.packages("wskm")))
library(wskm)
```

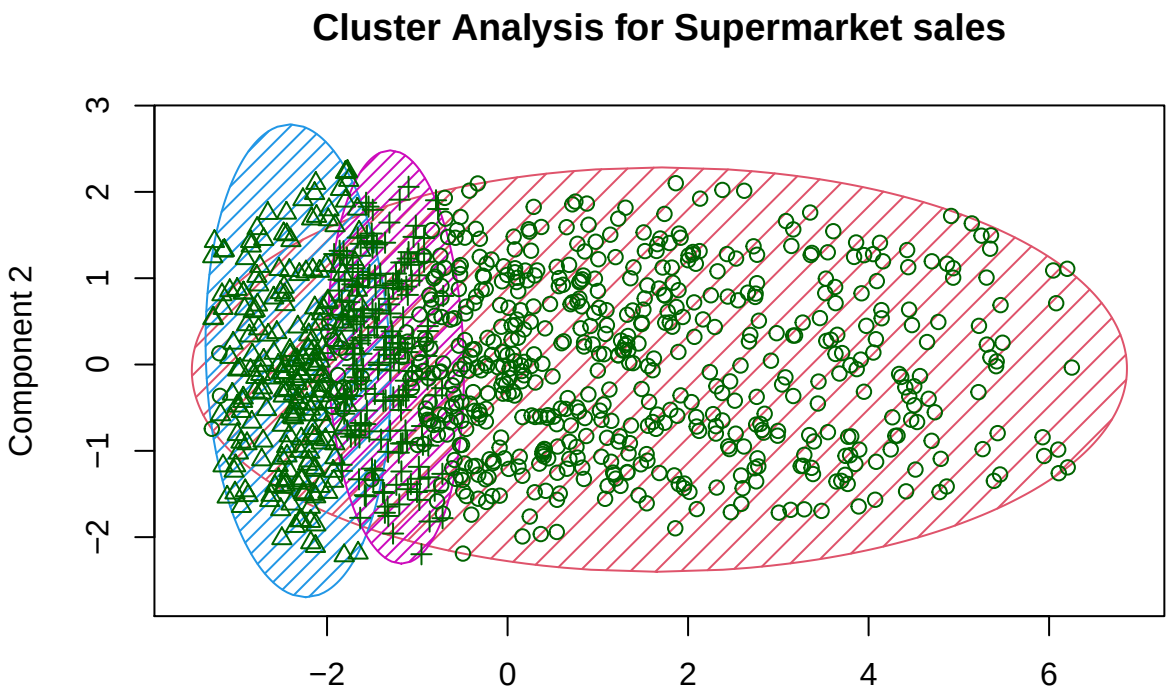
```
suppressWarnings(
        suppressMessages(if
                         (!require(cluster, quietly=TRUE))
                 install.packages("cluster")))
library("cluster")
```

```
#Deploying the function
set.seed(23)
model <- ewkm(num, 3, lambda=2, maxiter=1000)
model
```

```
## K-means clustering with 3 clusters of sizes 587, 230, 183
##
## Cluster means:
```

```
##   Unit.price Quantity       Tax      cogs gross.income   Rating     Total
## 1   66.22920 6.936968 22.451185 449.02370    22.451185 6.933220 471.47488
## 2   38.24948 2.469565  3.249196  64.98391     3.249196 6.796087  68.23311
## 3   43.70607 4.754098  7.941030 158.82060     7.941030 7.321311 166.76163
##
## Clustering vector:
##    [1] 1 2 1 1 1 1 1 1 2 3 2 2 1 1 1 1 1 1 3 2 1 1 2 3 1 1 2 3 1 1 1 1 1 3 1 2 1
##   [38] 1 1 1 2 1 1 1 2 1 1 3 1 1 1 1 1 2 2 2 1 2 1 1 3 2 1 1 3 1 3 2 1 1 2 1 1 3 1
##   [75] 1 1 1 3 1 2 1 1 3 1 1 1 1 1 2 1 1 1 3 1 2 2 1 2 3 1 2 1 1 1 1 1 3 1 1 1 2
##  [112] 1 1 1 1 2 2 2 1 2 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 3 3 1 1 1 1 1 2 1 1 1 1
##  [149] 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 3 1 1 1 1 1 1 1 1 3 1 3 1 3 3 1 1 1 1 3 1 2
##  [186] 2 1 3 2 1 1 3 1 2 3 2 2 2 3 1 1 1 1 1 2 1 1 1 2 1 1 1 1 3 1 2 3 1 1 3 1 1
##  [223] 2 2 2 3 1 3 1 1 3 2 1 1 1 3 2 1 2 2 1 3 3 1 1 1 1 2 1 1 1 1 2 2 1 3 2 3 3
##  [260] 1 1 2 3 1 2 1 3 1 1 1 1 2 3 2 1 1 1 1 1 1 1 2 1 1 3 1 2 3 1 1 1 1 2 2 3 1
##  [297] 1 1 1 2 1 2 2 3 3 1 1 1 2 3 1 3 2 3 1 3 2 1 1 3 1 2 2 1 3 1 1 1 3 3 1 2 1
##  [334] 2 2 3 1 1 3 1 3 1 1 1 3 1 2 1 3 3 1 1 1 1 2 1 1 1 2 2 1 1 1 1 3 1 1 1 3 1
##  [371] 3 3 1 1 1 1 1 1 1 2 1 2 1 1 3 2 1 1 1 3 1 1 2 1 2 1 1 3 1 3 1 3 2 3 1 1 2
##  [408] 1 1 3 1 2 2 2 1 2 3 2 3 3 1 1 1 1 2 1 1 2 1 1 3 2 3 1 2 1 1 1 2 1 2 1 1 2
##  [445] 3 2 1 3 2 2 1 3 1 2 2 3 1 1 1 2 3 1 1 2 1 1 1 1 2 2 1 3 1 1 1 1 3 1 1 2 1
##  [482] 1 3 1 1 1 1 3 2 1 2 1 3 3 1 1 3 1 2 1 2 2 3 3 2 3 3 1 1 2 1 1 1 1 1 2 1 3
##  [519] 2 1 3 1 1 3 1 1 3 1 1 1 3 1 1 3 3 1 2 2 1 1 2 2 2 3 2 3 1 1 1 1 3 1 1 3 3
##  [556] 1 3 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 2 3 1 1 1 3 1 3 2 1 3 2 1 3 3 1 1 2 1 3
##  [593] 2 1 1 2 3 1 3 3 2 2 1 1 1 3 1 1 2 3 2 1 1 2 1 1 1 1 1 1 1 2 1 1 2 3 3 1 3
##  [630] 2 1 1 3 1 1 1 1 2 2 3 1 1 1 1 2 3 1 2 2 1 1 1 1 1 1 2 2 1 3 2 3 3 1 1 1 1
##  [667] 3 2 1 2 1 3 3 1 1 3 1 1 1 1 2 2 1 2 3 2 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 3
##  [704] 1 1 1 3 2 3 2 1 1 1 3 1 3 1 2 3 2 3 1 3 1 2 1 3 1 1 1 2 3 2 1 2 1 1 1 1 1
##  [741] 1 2 3 2 1 2 1 2 1 1 2 3 3 1 1 1 1 1 3 1 1 1 1 2 1 1 1 2 1 2 1 1 1 2 3 3 1
##  [778] 2 3 1 3 1 2 1 2 1 1 2 3 1 2 1 1 1 2 2 2 1 2 1 3 1 1 1 1 1 1 3 3 2 1 2 1 1 2
##  [815] 1 1 1 1 1 1 1 1 2 1 1 1 2 1 3 1 1 2 1 2 2 1 2 3 1 1 3 3 2 2 2 2 1 2 2 1 1 2
##  [852] 3 1 1 1 1 3 1 1 1 2 2 2 1 1 2 1 3 2 1 3 2 1 1 2 1 2 2 1 2 1 2 1 1 3 1 2 1
##  [889] 2 1 1 1 1 3 1 1 1 1 1 1 1 3 3 2 1 1 1 1 3 1 2 1 1 1 1 3 2 1 1 2 1 2 2 1 1
##  [926] 2 2 1 1 2 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 2 3 1 3 2 1 2 1 1 1 1 3 1 2 2
##  [963] 1 1 2 2 1 1 2 3 1 1 1 1 3 2 1 3 2 2 1 1 1 1 1 1 2 1 1 1 1 1 3 3 2 2 1 2 2
## [1000] 1
##
## Within cluster sum of squares by cluster:
## [1] 53385220.1   464864.0   342620.4
##  (between_SS / total_SS =  53.4 %)
##
## Available components:
##
##  [1] "cluster"          "centers"          "totss"            "withinss"
##  [5] "tot.withinss"     "betweenss"        "size"             "iterations"
##  [9] "total.iterations" "restarts"         "weights"
```

```
#Clustering
clusplot(num, model$cluster, color=TRUE, shade=TRUE,
         lines=1,main='Cluster Analysis for Supermarket sales')
```



These two components explain 84.6 % of the point variability.

```
# Weights are calculated for each variable and cluster.
# They are a measure of the relative importance of each variable

round(model$weights*100,2)
```

```
##   Unit.price Quantity Tax cogs gross.income Rating Total
## 1          0        0   0    0            0  99.99     0
## 2          0        0  50    0           50   0.00     0
## 3          0        0  50    0           50   0.00     0
```

Conclusion

After conducting filtering the most important features left were price,Quantity,Gross income and Rating.