Sales Analysis for Carrefour

Wangechi Mungai

01/04/2022

1.Defining the Question

a) Specifying the Question Reducing your dataset to a low dimensional dataset using the t-SNE algorithm or PCA.

b) Defining the Metric for Success The model will be appraised successful if it will be able to reduce the dataset to a low dimensional dataset. **c) Understanding the context**

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).

d) Recording the Experimental Design

The following are the experimental design i took in order to complete this project:

1.Importing all the necessary libraries

2.Loading the dataset

3.Reading, cleaning the dataset

4.Performing data reduction techniques using PCA.

e)Reading the Data

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

```
##    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
```

f) Checking the Data

```
#previewing tail of dataset
tail(sell)
```

```
##       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
```

```
#previewing the shape of the dataset
dim(sell)
```

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

The dataset contains 1000 rows and 16 columns

```
#previewing the descriptive statistics of dataset
summary(sell)
```

```
##   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
```

```
#checking the datatypes of the columns
sapply(sell, 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"
```

```
# information about the dataset
str(sell)
```

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

```
#checking for percentage of any null values
colMeans(is.na(sell)) *100
```

```
##              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 null values in the dataset

```
#checking for duplicate values
anyDuplicated(sell)
```

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

There are no duplicates values in the dataset

g)Data Cleaning

```
#checking numerical cols
num_cols <- unlist(lapply(sell, is.numeric))
num_cols
```

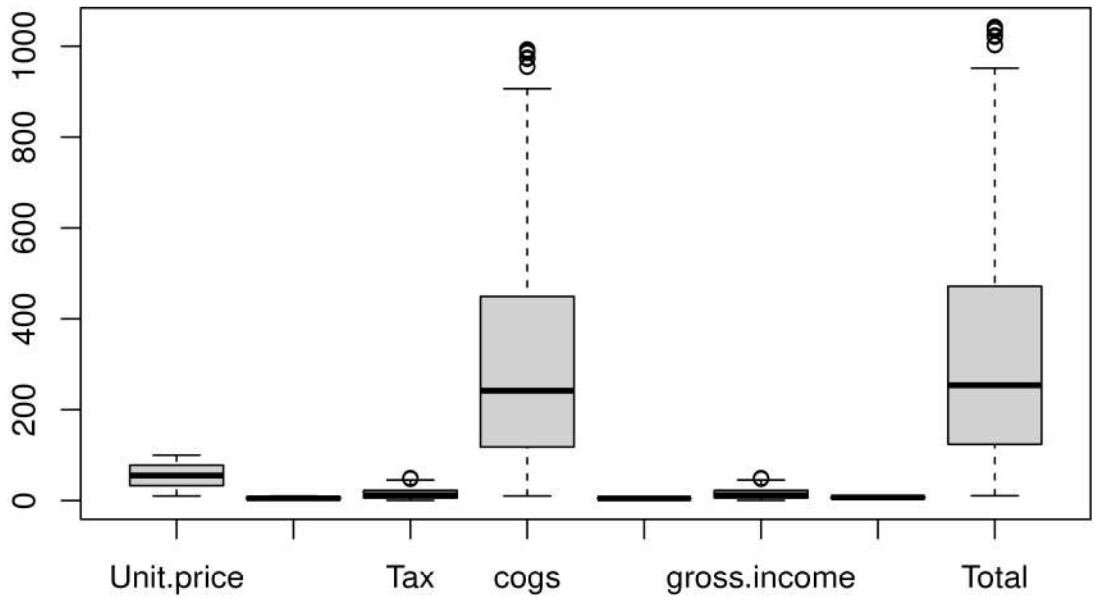
```
##              Invoice.ID                  Branch           Customer.type
##                   FALSE                   FALSE                   FALSE
##                  Gender           Product.line             Unit.price
##                   FALSE                   FALSE                    TRUE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

```
#displayig the numerical columns
df_num <- sell[ , num_cols]
head(df_num, 5)
```

```
##   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
##   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
```

```r
#checking outliers
outlier <- function(x){
  out <- boxplot.stats(x)$out
  return((length(out)/  1000)*100)
}
```

```r
 #visualizing the outliers
   boxplot(df_num)
```



there are a number of outliers in the i Tax,cogs,gross income, and Total columns but will not be removed since they are neccesary for our analysis

```r
# removing columns with zero variance
#
num <- df_num[ , which(apply(df_num, 2, var) != 0)]
head(num)
```

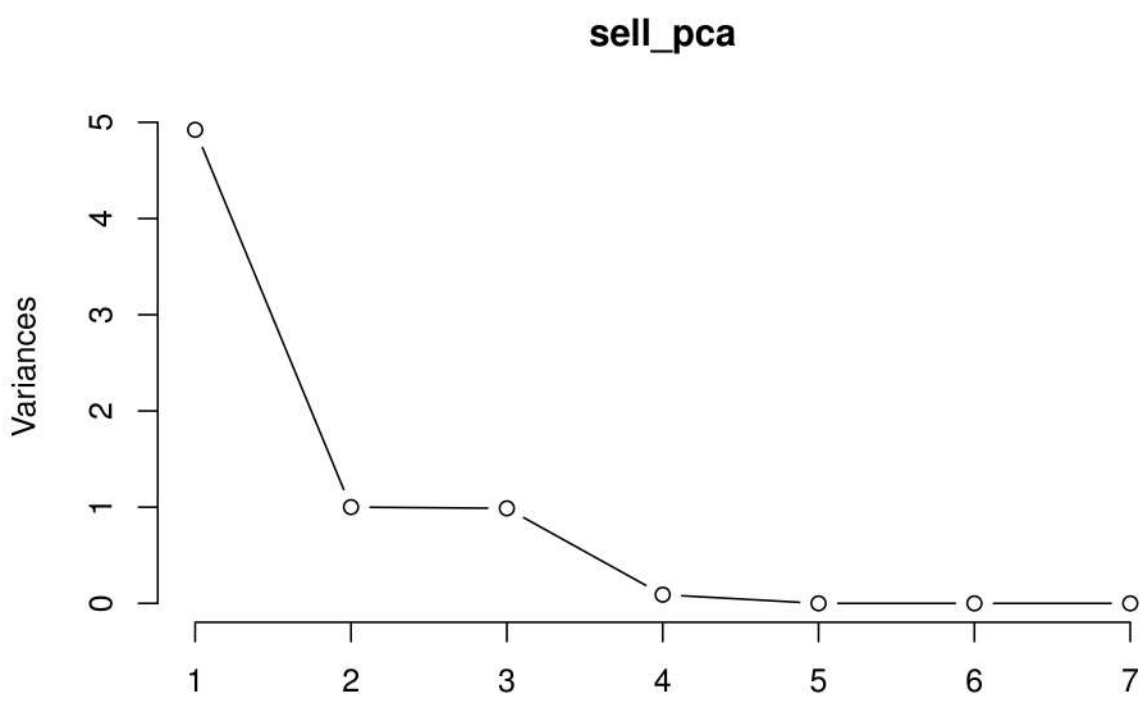
```
##   Unit.price Quantity     Tax   cogs gross.income Rating    Total
## 1      74.69        7 26.1415 522.83      26.1415    9.1 548.9715
## 2      15.28        5  3.8200  76.40       3.8200    9.6  80.2200
## 3      46.33        7 16.2155 324.31      16.2155    7.4 340.5255
## 4      58.22        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
```

h) Implementing solution

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

```r
#visualizing the principle components
plot(sell_pca,type = 'l')
```



the propotion of variance decreases as principle components increases

```
# Calling str() to have a look at your PCA object
# ---
#
str(sell_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"
```

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

```
## Loading required package: ggplot2

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid
```

```
library(devtools)
```

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

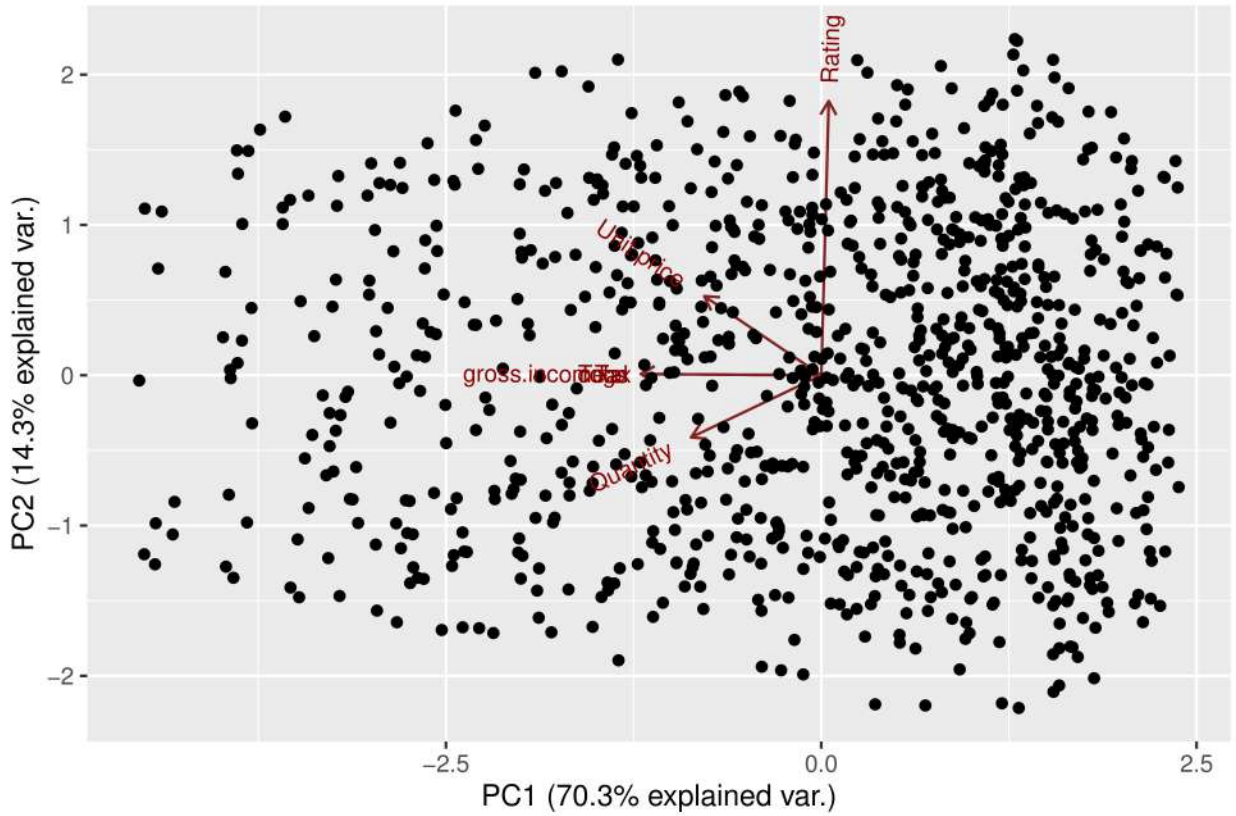
```
library(ggplot2)
library(plyr)
library(scales)
library(grid)
library(ggbiplot)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()    masks plyr::arrange()
## x readr::col_factor() masks scales::col_factor()
## x purrr::compact()    masks plyr::compact()
## x dplyr::count()      masks plyr::count()
## x purrr::discard()    masks scales::discard()
## x dplyr::failwith()   masks plyr::failwith()
## x dplyr::filter()     masks stats::filter()
## x dplyr::id()         masks plyr::id()
## x dplyr::lag()        masks stats::lag()
## x dplyr::mutate()     masks plyr::mutate()
## x dplyr::rename()     masks plyr::rename()
## x dplyr::summarise()  masks plyr::summarise()
## x dplyr::summarize()  masks plyr::summarize()
```

```
#visualizing pca
ggbiplot(sell_pca, scale =0.4)
```



the data points are distributed everywhere

Part 2: Feature Selection

```
wrap <- df_num
head(wrap)
```

```
##   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
```

```
#loading libraries
library(caret)
```

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

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

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
elim <- c("gross.margin.percentage")
#
#dropping the column
#
wrap <- wrap[, !names(wrap) %in% elim]
head(wrap)
```

```
##   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
```

```
# Checking correlation using corr matrix
cm <- cor(wrap)
# Find attributes that are highly correlated
highcorr<- findCorrelation(cm, cutoff=0.75)
highcorr
```

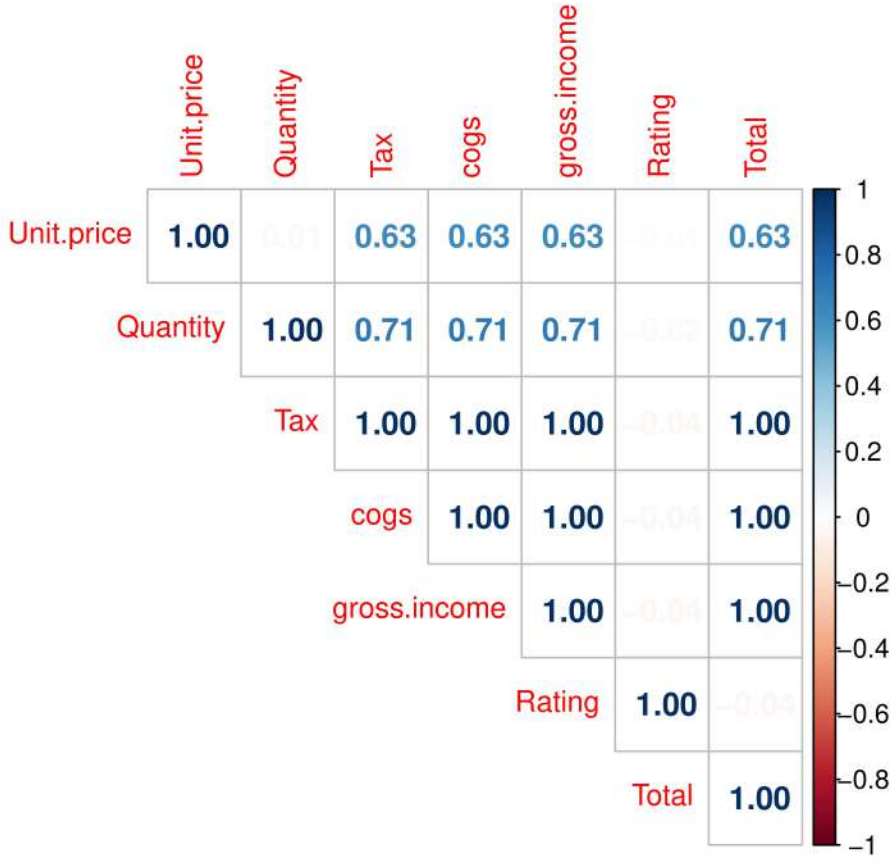
```
## [1] 4 7 3
```

```
# Removing the high correlated variables
final<-cm[-highcorr]
```

```
# scaling the dataset
#
library(dplyr)
scaling <- as.data.frame(scale(wrap))
head(wrap)
```

```
##   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
```

```
corrplot(cor(scaling), type = 'upper', method = 'number', tl.cex = 0.9)
```



Feature Ranking

```
library(FSelector)
colnames(scaling)
```

```
## [1] "Unit.price"   "Quantity"     "Tax"          "cogs"         "gross.income"
## [6] "Rating"       "Total"
```

```
result <- linear.correlation(Unit.price~., scaling)
result
```

```
##              attr_importance
## Quantity         0.010777564
## Tax              0.633962089
## cogs             0.633962089
## gross.income     0.633962089
## Rating           0.008777507
## Total            0.633962089
```

```
Subset <- cutoff.k(result, 5)
as.data.frame(Subset)
```

```
##         Subset
## 1          Tax
## 2         cogs
## 3 gross.income
## 4        Total
## 5     Quantity
```

```
Subset1 <-cutoff.k.percent(result, 0.4)
as.data.frame(Subset1)
```

```
##   Subset1
## 1     Tax
## 2    cogs
```

```
# we will use an entropy - based approach as shown below;
# ---
#
res <- information.gain(Unit.price~., scaling)

# Choosing Variables by cutoffSubset <- cutoff.k(Scores2, 5)
Subset2 <- cutoff.k(res, 5)
as.data.frame(Subset2)
```

```
##        Subset2
## 1          Tax
## 2         cogs
## 3 gross.income
## 4        Total
## 5     Quantity
```

using Wrapper Methods

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

library(clustvarsel)
```

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

```
head(wrap)
```

```
##   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
```

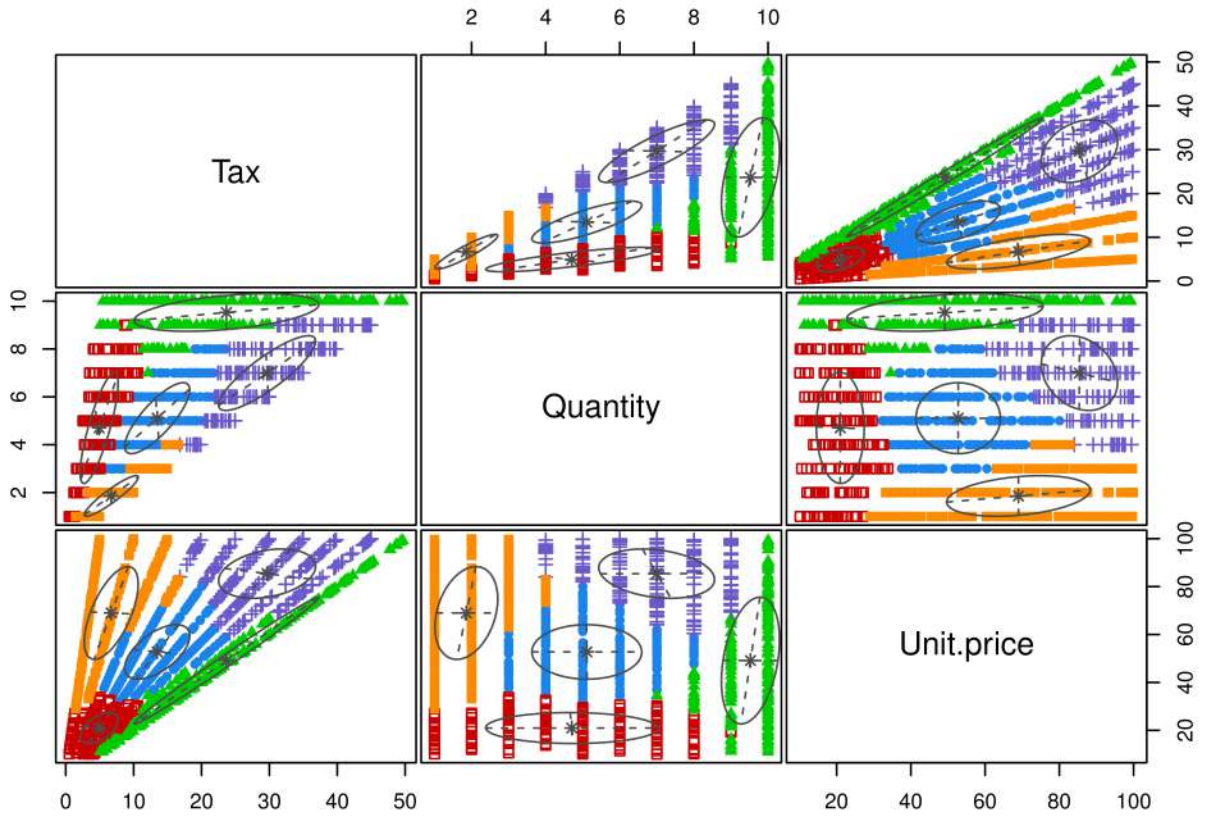
```
# Sequential forward greedy search (default)
gr = clustvarsel(scaling, G = 1:5)
gr
```

```
## ----------------------------------------------------
## 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 -3799.377   VEV 5  830.4539 Accepted
##         Unit.price          Add -2804.291   EVV 5 2339.8019 Accepted
##         Unit.price       Remove -3799.377   VEV 5 2339.8019 Rejected
##             Rating          Add -5778.207   EVV 5 -123.2240 Rejected
##         Unit.price       Remove -3799.377   VEV 5 2339.8019 Rejected
##
## Selected subset: Tax, Quantity, Unit.price
```

```
# building the clustering model:
sub = wrap[gr$subset]
mod = Mclust(sub, G = 1:5)
summary(mod)
```

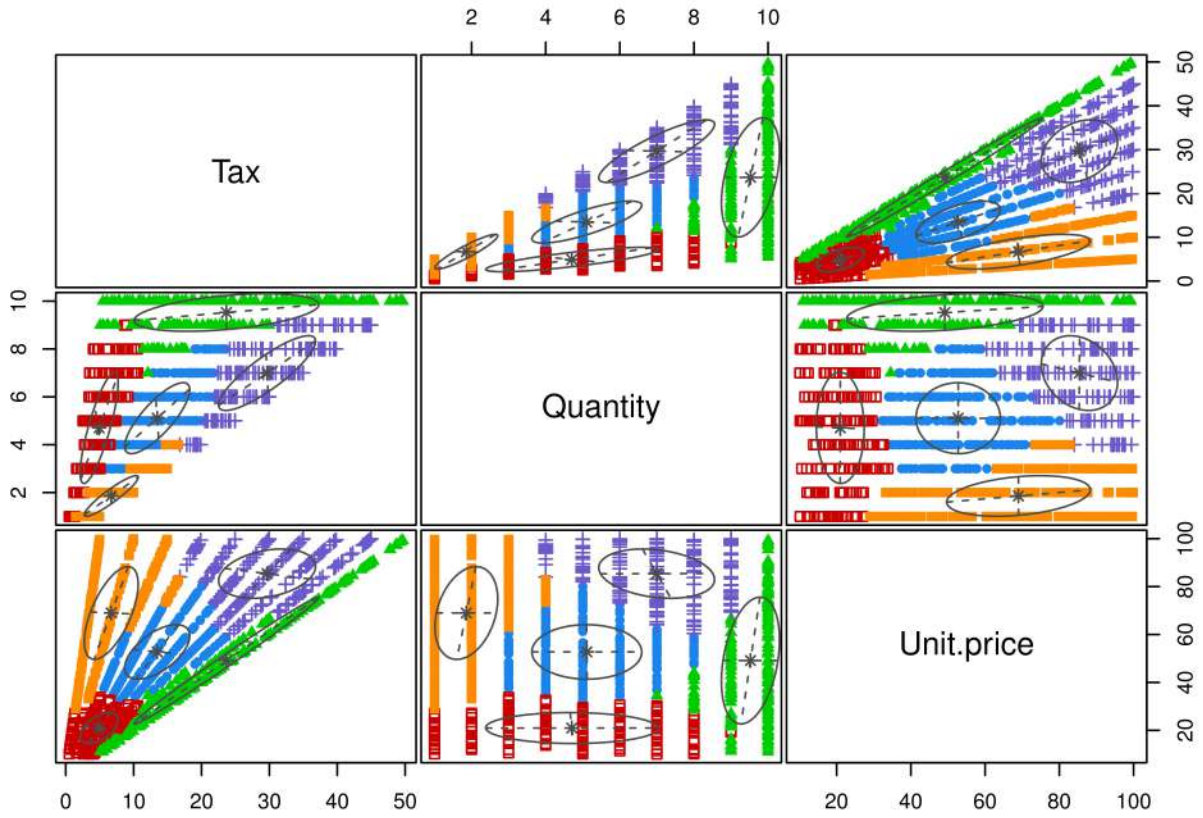
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EVV (ellipsoidal, equal volume) model with 5 components:
##
##  log-likelihood    n df       BIC       ICL
##       -7991.262 1000 45 -16293.37 -16407.49
##
## Clustering table:
##   1   2   3   4   5
## 220 190 186 190 214
```

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



there is presence of 4 cluster

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



Using Embedded Method

```
library(wskm)
```

```
## Loading required package: latticeExtra
```

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

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

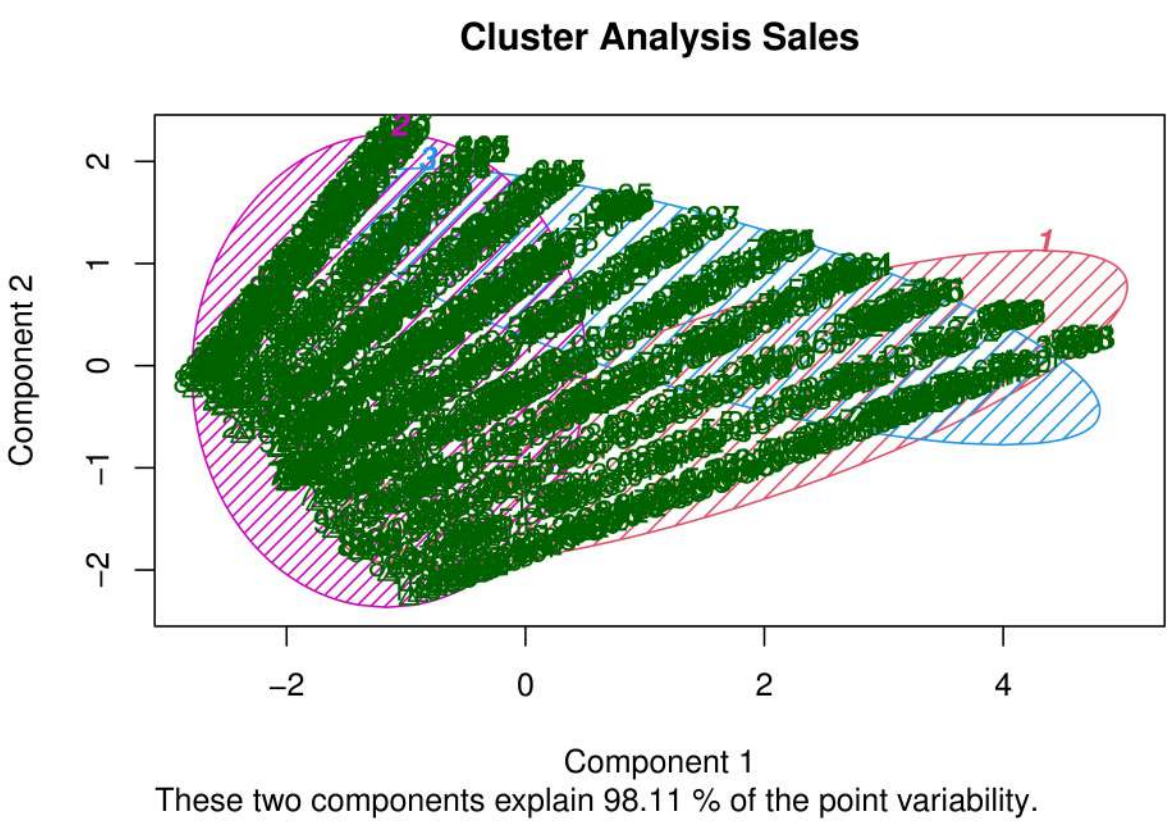
```
## Loading required package: fpc
```

```
model <- ewkm(scaling[1:4], 3, lambda=2, maxiter=1000)
model
```

```
## K-means clustering with 3 clusters of sizes 263, 488, 249
##
## Cluster means:
##    Unit.price    Quantity        Tax       cogs
## 1  0.05406789  1.04553189  0.7655679  0.7655679
## 2 -0.54976884 -0.60273337 -0.7691850 -0.7691850
```

```
## 3  1.02035075  0.07694377  0.6988672  0.6988672
##
## Clustering vector:
##    [1] 3 2 1 1 3 3 3 3 2 2 2 2 2 1 3 3 3 3 2 2 3 3 2 2 3 1 2 3 3 1 3 1 3 2 3 2 3
##   [38] 1 1 1 3 2 3 1 2 3 1 1 2 3 3 2 2 2 2 3 2 1 3 2 2 1 1 2 3 2 2 3 3 2 3 1 2 3
##   [75] 1 1 1 1 3 2 3 3 2 3 2 3 3 1 2 3 2 1 2 1 2 2 3 2 2 3 2 1 1 1 2 3 2 3 1 3 2
##  [112] 3 3 1 3 2 2 2 2 2 2 1 1 1 1 1 3 2 1 1 1 1 3 2 3 3 3 2 2 1 1 3 3 1 2 3 2 1 2
##  [149] 1 1 3 2 3 2 3 3 2 2 1 3 1 2 1 2 1 1 3 1 2 3 3 3 2 2 2 1 1 2 3 3 1 1 2 1 2
##  [186] 2 1 2 2 3 3 2 1 2 2 2 2 2 2 3 2 2 1 1 2 1 1 1 2 1 1 1 3 2 2 2 2 3 1 2 2 1
##  [223] 2 2 2 3 1 2 3 1 3 2 3 3 1 2 2 2 2 2 1 2 2 2 1 1 3 2 3 3 1 1 2 2 1 2 2 2 2
##  [260] 2 1 2 2 2 2 1 2 3 3 1 3 2 1 2 3 1 2 3 3 1 1 2 2 3 2 3 2 1 1 3 1 2 2 2 2 2
##  [297] 2 3 2 2 2 2 2 2 2 3 1 1 2 3 3 2 2 2 3 2 2 3 2 2 1 2 2 2 2 3 3 3 2 2 1 2 3
##  [334] 2 2 2 1 1 2 1 2 1 3 3 2 1 2 1 2 2 1 1 1 1 2 3 1 3 2 2 3 1 2 3 2 3 1 2 2 1
##  [371] 2 2 3 2 3 3 1 3 3 2 3 2 3 3 2 2 2 1 1 2 3 1 2 1 3 3 2 2 2 1 2 2 2 2 1 3 2
##  [408] 3 2 2 1 2 2 2 1 2 2 3 2 2 2 3 1 1 2 1 1 2 3 3 2 2 2 3 2 1 3 2 2 1 2 1 1 2
##  [445] 2 2 1 2 2 2 3 2 3 2 2 2 1 3 1 2 2 3 3 2 2 1 1 2 2 2 3 2 1 1 3 2 2 3 1 2 1
##  [482] 1 2 1 1 2 1 2 2 1 2 2 1 2 2 1 2 3 2 1 2 2 2 2 2 1 2 3 2 2 3 2 1 1 1 2 2 2
##  [519] 2 3 2 3 2 2 3 3 2 2 1 1 2 3 2 2 2 2 2 2 3 3 2 2 2 2 2 2 1 1 2 3 2 1 1 2 2
##  [556] 2 2 1 2 2 1 3 1 3 2 3 1 1 3 3 3 1 2 2 3 3 2 2 2 2 2 3 2 2 2 2 2 2 1 2 2 2
##  [593] 2 3 3 2 2 1 2 2 3 2 3 2 1 2 1 2 2 2 2 1 3 3 1 1 3 3 3 2 2 2 3 3 2 2 2 3 2
##  [630] 2 1 3 3 3 3 1 2 2 2 2 3 1 2 1 2 1 3 2 2 1 3 1 3 1 2 2 2 3 2 2 2 2 2 1 3 2 2
##  [667] 2 2 2 2 1 2 3 1 1 3 1 3 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 1 3 1 3 2 2 1 1 1 2 2
##  [704] 3 1 3 2 2 2 2 3 3 1 2 1 2 3 2 2 2 2 1 2 1 2 1 2 3 3 1 2 2 2 3 2 3 3 1 1 1
##  [741] 1 3 2 2 1 2 1 2 2 3 2 2 2 3 3 1 1 3 2 2 3 3 1 2 2 1 3 2 3 2 3 3 1 2 2 2 1
##  [778] 2 2 1 1 3 2 1 2 3 1 2 2 1 2 3 1 3 2 2 2 1 2 3 2 1 1 1 1 3 2 2 2 1 2 1 3 2
##  [815] 1 1 1 1 3 2 2 2 2 2 1 2 1 2 1 3 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 3 2 2 1 3 2
##  [852] 2 1 1 3 1 2 2 1 1 3 2 2 1 3 2 1 2 2 2 2 2 2 3 2 1 2 2 1 2 1 1 1 2 3 1 2 3
##  [889] 2 3 3 1 3 2 2 1 1 2 3 1 1 2 2 2 2 3 3 3 3 1 2 2 1 1 3 2 2 3 2 2 1 2 2 3 2
##  [926] 2 2 1 1 2 3 3 3 1 3 3 3 3 3 1 2 1 1 2 2 3 3 2 2 1 2 2 2 2 3 1 1 2 2 1 2 2
##  [963] 2 3 2 2 1 3 2 2 3 1 3 3 3 2 3 2 2 2 2 2 1 1 3 3 2 1 3 1 2 3 2 2 2 2 1 2 2
## [1000] 3
##
## Within cluster sum of squares by cluster:
## [1] 636.4580 774.8489 581.9746
##  (between_SS / total_SS =  50.1 %)
##
## Available components:
##
##  [1] "cluster"          "centers"          "totss"            "withinss"
##  [5] "tot.withinss"     "betweenss"        "size"             "iterations"
##  [9] "total.iterations" "restarts"         "weights"
```

```
# Cluster Plot against first 2 principal components
library("cluster")
clusplot(wrap[1:4], model$cluster, color=TRUE, shade=TRUE,
         labels=2, lines=3,main='Cluster Analysis Sales')
```



There are 4 clusters present where 2 components have 98.11 variability

```
#checking weights
round(model$weights*100,4)
```

```
##   Unit.price Quantity     Tax    cogs
## 1     0.0025  99.9925  0.0025  0.0025
## 2     0.0025   0.0025 49.9975 49.9975
## 3    99.9925   0.0025  0.0025  0.0025
```