Dimensionality reduction

Overview of problem

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 three parts where you'll explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights. # Load and Preview Data

```
# Load Dataset
library(tinytex)
df1=read.csv('http://bit.ly/CarreFourDataset')

head(df1)
```

```
##    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
```

```
# CHeck the data
#shape
df1<-df1[-1]
dim(df1)
```

```
## [1] 1000   15
```

```
# datatypes
sapply(df1,class)
```

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

```
# SUmmary
summary(df1)
```

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

Observation

1. We have 1000 records, 16 fields
2. 8 columns are of character datatype
3. No null values
4. SInce the values are of different scale, we will need to scale them.

```
# CHecking for unique values
sapply(df1, function(x) length(unique(x)))
```

```
##                   Branch            Customer.type                   Gender
##                        3                        2                        2
##             Product.line             Unit.price                 Quantity
##                        6                      943                       10
##                      Tax                     Date                     Time
##                      990                       89                      506
##                  Payment                     cogs gross.margin.percentage
##                        3                      990                        1
##             gross.income                   Rating                    Total
##                      990                       61                      990
```

Data CLeaning

```
# Checking for missing values
colSums(is.na(df1))
```

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

```
# Checking for duplicates
anyDuplicated(df1)
```

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

There are no missing entries or duplicate fields

```
# COlumn names
colnames(df1)
```

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

Wewill drop gross mean percentage since it has a constant value of 4.76 and date and time columns

```
# Drop unneeded column

df1<-subset(df1,select=-c(gross.margin.percentage,Date  ,Time))
```

```
# Change Datatype
# to factor
abc<-c('Branch','Customer.type','Gender','Product.line','Payment')
df1[abc]<-lapply(df1[abc],factor)
sapply(df1,class)
```

```
##        Branch Customer.type        Gender  Product.line    Unit.price
##      "factor"      "factor"      "factor"      "factor"     "numeric"
##      Quantity           Tax       Payment          cogs  gross.income
##     "integer"     "numeric"      "factor"     "numeric"     "numeric"
##        Rating         Total
##     "numeric"     "numeric"
```

Exploratory Data Analysis

```
#Load libraries
library(tidyr)
library(ggplot2)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
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
```

```
library(psych)
```

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

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

```
colnames(df1)
```

```
##  [1] "Branch"        "Customer.type" "Gender"        "Product.line"
##  [5] "Unit.price"    "Quantity"      "Tax"           "Payment"
##  [9] "cogs"          "gross.income"  "Rating"        "Total"
```

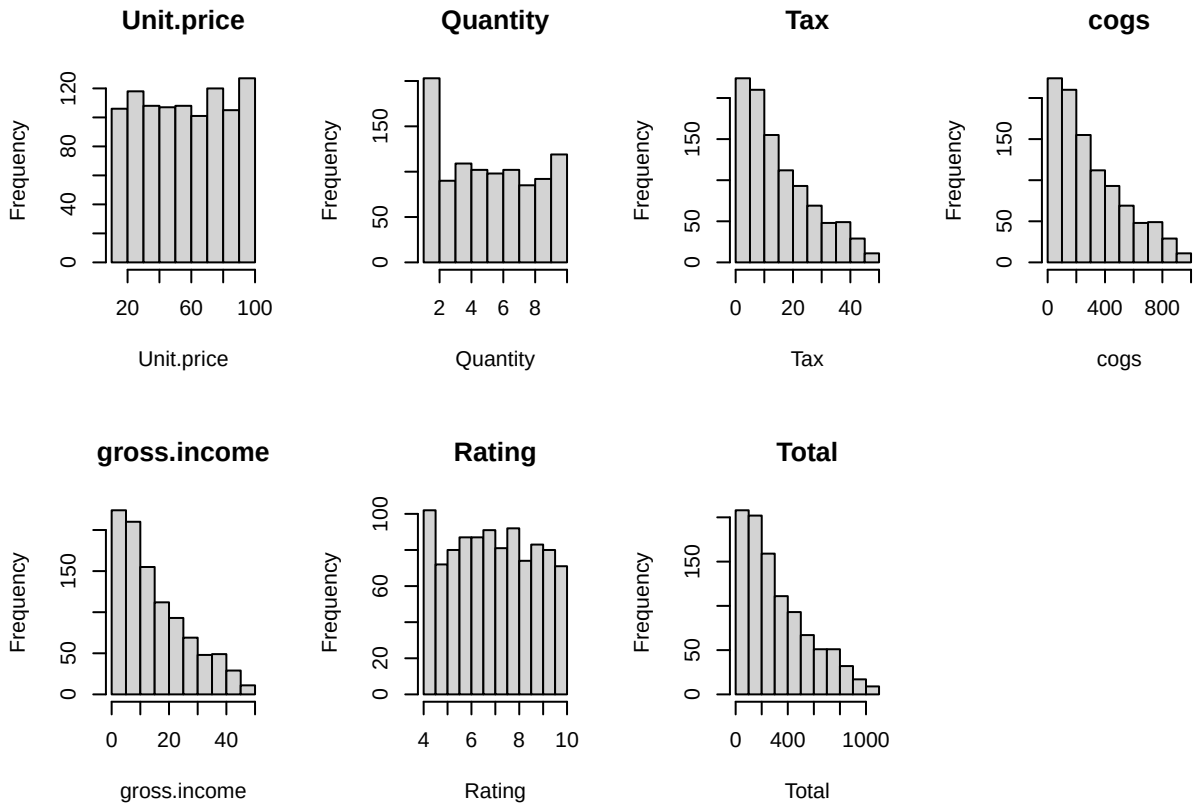
```
num<- subset(df1, select=c(Unit.price, Quantity,Tax,cogs,gross.income,Rating,Total))
head(num)
```

```
##   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
```

```
# Checkinh statistical measures of central tendency
describe(num)
```

```
##              vars    n   mean     sd median trimmed    mad   min     max
## Unit.price      1 1000  55.67  26.49  55.23   55.62  33.37 10.08   99.96
## Quantity        2 1000   5.51   2.92   5.00    5.51   2.97  1.00   10.00
## Tax             3 1000  15.38  11.71  12.09   14.00  11.13  0.51   49.65
## cogs            4 1000 307.59 234.18 241.76  279.91 222.65 10.17  993.00
## gross.income    5 1000  15.38  11.71  12.09   14.00  11.13  0.51   49.65
## Rating          6 1000   6.97   1.72   7.00    6.97   2.22  4.00   10.00
## Total           7 1000 322.97 245.89 253.85  293.91 233.78 10.68 1042.65
##                range skew kurtosis   se
## Unit.price     89.88 0.01    -1.22 0.84
## Quantity        9.00 0.01    -1.22 0.09
## Tax            49.14 0.89    -0.09 0.37
## cogs          982.83 0.89    -0.09 7.41
## gross.income   49.14 0.89    -0.09 0.37
## Rating          6.00 0.01    -1.16 0.05
## Total        1031.97 0.89    -0.09 7.78
```

```
# Distributions of feature variables
par(mfrow=c(2,4))
for(i in 1 :length(num)){
  hist(num[,i],main=names(num[i]),xlab=names(num[i]))
}
```



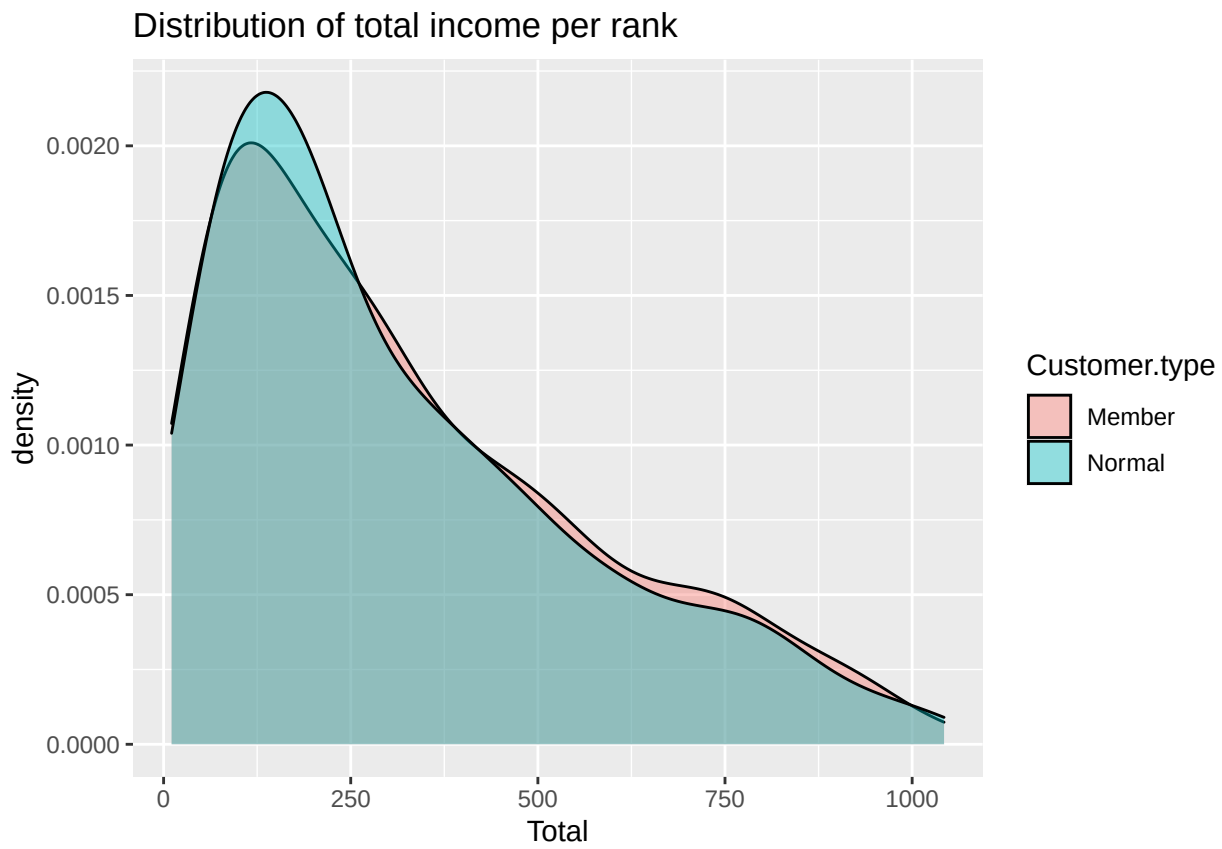
Observations - Amount purchased per unit price seems to vary at all prices through a unit price of 90-100 -Amount purchased seem to decrease with increase in Total and gross income, tax, and cogs with highest frequency levels being where variable values are least

checking relationship of target variable and feature variables

```
# GEnder
ggplot(df1,
       aes(x=Total,fill=Gender))+
  geom_density(alpha=0.4)+
  labs(title = 'Distribution of total income per gender')
```

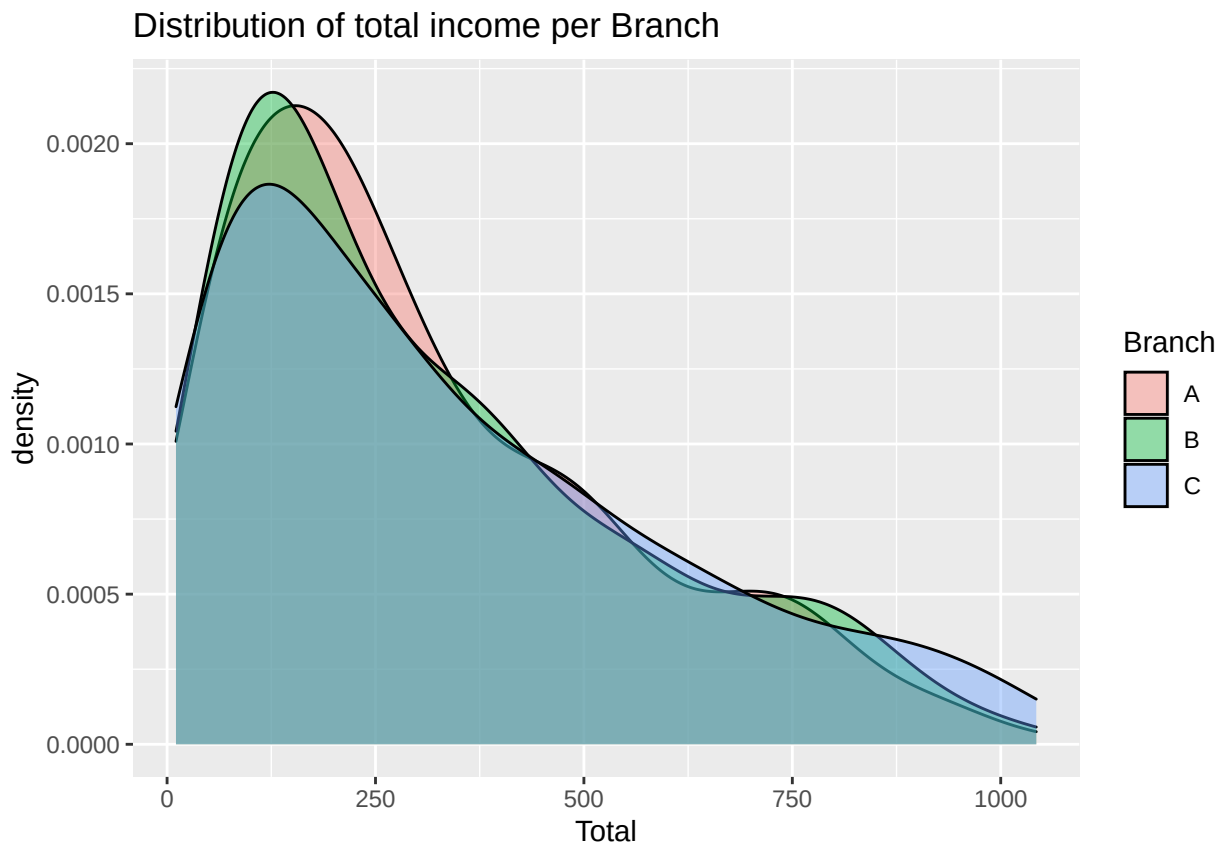
Both genders seem to affect Total income similarly with males having the max effect at around 150 and females exceedd males past 280

```
# Rank
ggplot(df1,
       aes(x=Total,fill=Customer.type))+
  geom_density(alpha=0.4)+
  labs(title = 'Distribution of total income per rank')
```



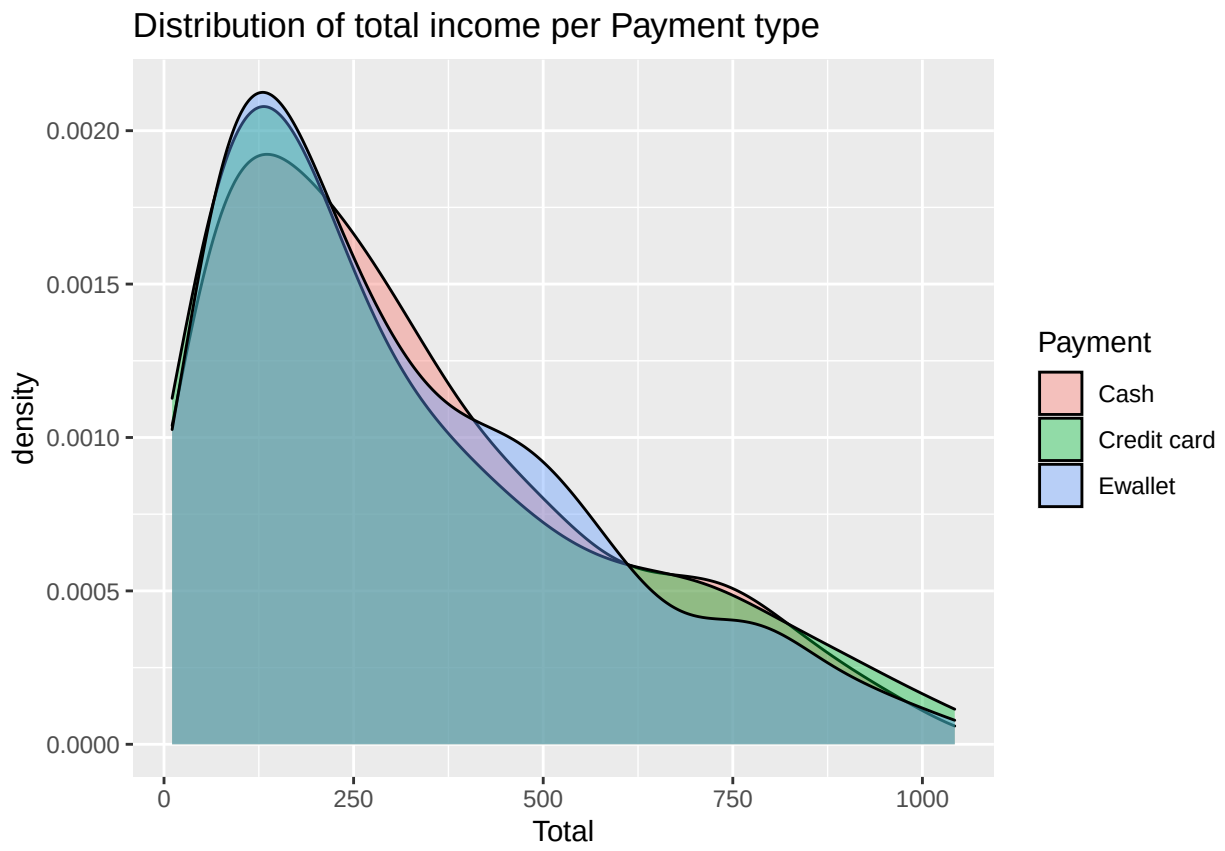
Normal customers have a slightly greater influence than members

```
# Branch
ggplot(df1,
       aes(x=Total,fill=Branch))+
  geom_density(alpha=0.4)+
  labs(title = 'Distribution of total income per Branch')
```

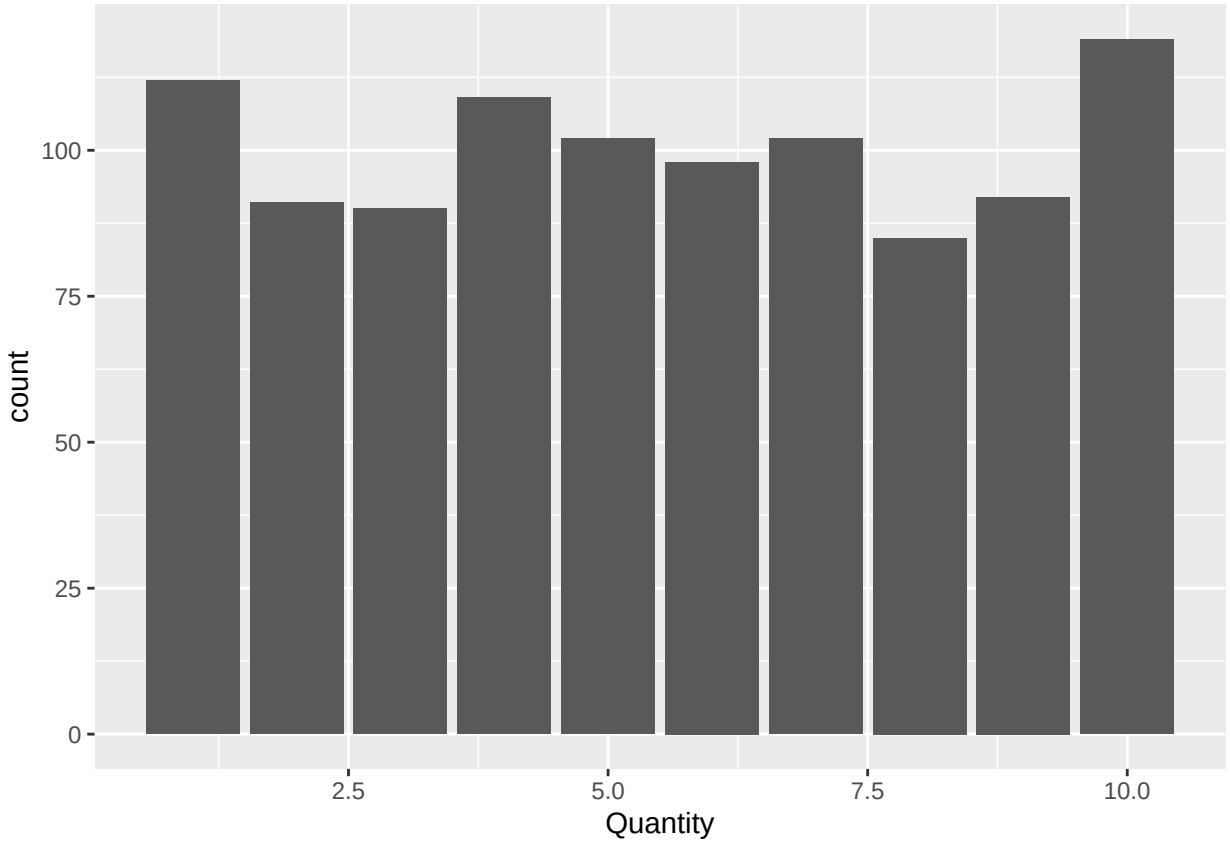


Branch A contributes more and branch C Contributes the least

```
# Payment method
ggplot(df1,
       aes(x=Total,fill=Payment))+
  geom_density(alpha=0.4)+
  labs(title = 'Distribution of total income per Payment type')
```




```
# The Quantity distribution of the store
ggplot(df1,aes(x=Quantity))+
  geom_bar()
```



Most people purchased 10 items followed by 1 Least bought number of items was 8

Feature engineering

```
# We drop target variab
df1<-df1[-12]
```

```
# Encoding the factor columns
df1$Branch <- as.numeric(df1$Branch)
df1$Gender <- as.numeric(df1$Gender)
df1$Customer.type<-as.numeric(df1$Customer.type)
df1$Product.line <- as.numeric(df1$Product.line)
df1$Payment <- as.numeric(df1$Payment)
```

##Dimensionality Reduction ## PCA

```
# We scale the data
scale_df1<-scale(df1)
```

```
# We pass dataframe to prcomp()
df1.pca <- prcomp(scale_df1, center = TRUE, scale. = TRUE)
summary(df1.pca)
```

```
## Importance of components:
```

```
##                            PC1    PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.9836 1.0631 1.03159 1.00991 0.99289 0.9771 0.96270
## Proportion of Variance 0.3577 0.1027 0.09674 0.09272 0.08962 0.0868 0.08425
## Cumulative Proportion  0.3577 0.4604 0.55719 0.64991 0.73953 0.8263 0.91058
##                            PC8     PC9      PC10      PC11
## Standard deviation     0.94823 0.29062 1.908e-16 4.398e-17
## Proportion of Variance 0.08174 0.00768 0.000e+00 0.000e+00
## Cumulative Proportion  0.99232 1.00000 1.000e+00 1.000e+00
```

We obtain 11 principal components each which explain a percentage of total variation PC1 (35.8%) and PC2 (10.3%) form 46% of the cummuliative frequency. The first 8 principal components add upto 99%

```
str(df1.pca)
```

```
## List of 5
##  $ sdev    : num [1:11] 1.984 1.063 1.032 1.01 0.993 ...
##  $ rotation: num [1:11, 1:11] 0.0267 -0.0155 -0.0338 0.0206 0.3273 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11] "Branch" "Customer.type" "Gender" "Product.line" ...
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:11] -1.84e-18 -2.19e-16 -2.19e-16 4.83e-17 -1.06e-16 ...
##   ..- attr(*, "names")= chr [1:11] "Branch" "Customer.type" "Gender" "Product.line" ...
##  $ scale   : Named num [1:11] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:11] "Branch" "Customer.type" "Gender" "Product.line" ...
##  $ x       : num [1:1000, 1:11] 1.79 -2.05 0.11 1.29 2.43 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:1000] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
# Then Loading our ggbiplot library
#
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'
```

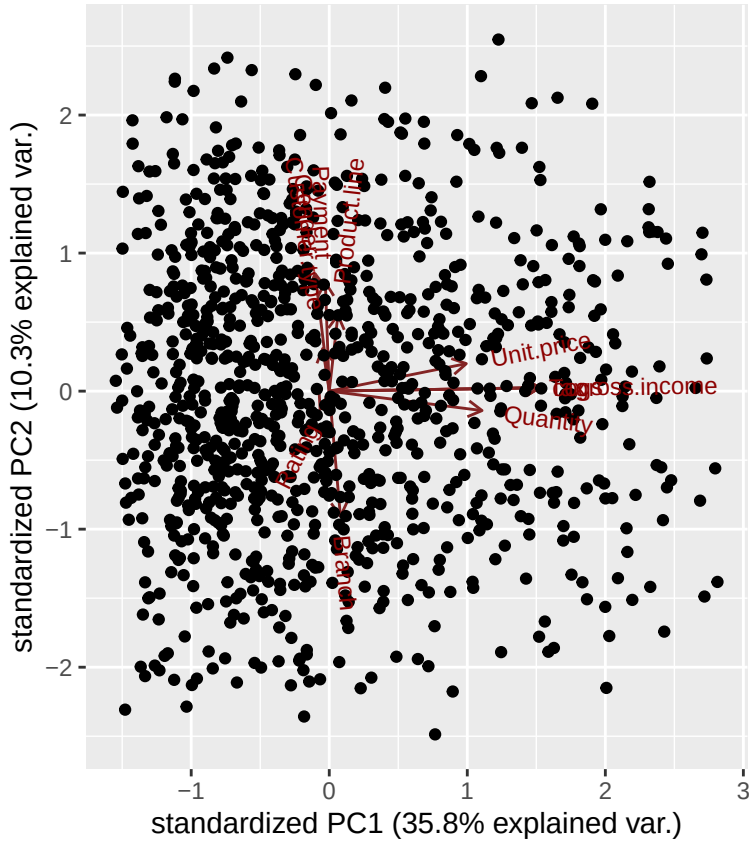
```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## Loading required package: scales

##
## Attaching package: 'scales'

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

## Loading required package: grid
```

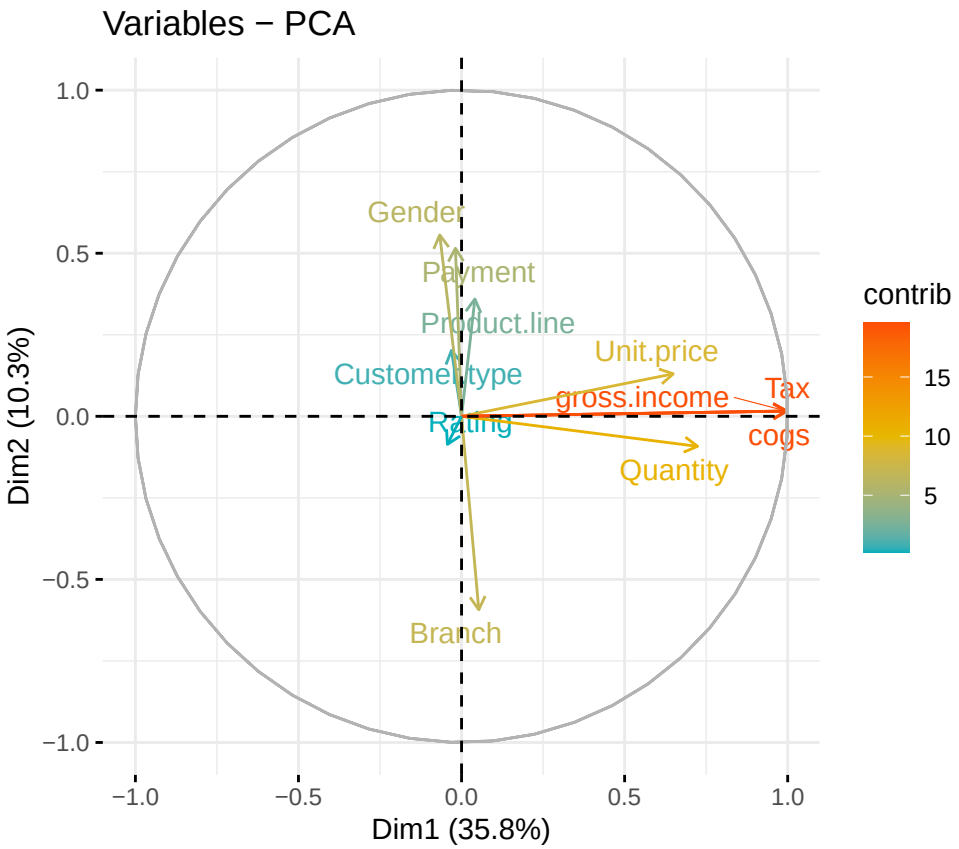
```
library(factoextra)
```

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

```
ggbiplot(df1.pca)
```




```
fviz_pca_var(df1.pca,
             col.var='contrib',gradient.cols=c('#00AFBB','#E7B800','#FC4E07'),
             repel=TRUE)
```



Gross income, Tax and cogs contribute highly to the PC1 whereas GEnder,branch , payment contribute to PC2

```
# Eigen Values
eig.val<-get_eigenvalue(df1.pca)
eig.val
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1  3.934732e+00     3.577029e+01                     35.77029
## Dim.2  1.130132e+00     1.027393e+01                     46.04423
## Dim.3  1.064187e+00     9.674426e+00                     55.71865
## Dim.4  1.019927e+00     9.272061e+00                     64.99071
## Dim.5  9.858334e-01     8.962122e+00                     73.95283
## Dim.6  9.548085e-01     8.680077e+00                     82.63291
## Dim.7  9.267825e-01     8.425296e+00                     91.05821
## Dim.8  8.991345e-01     8.173950e+00                     99.23216
## Dim.9  8.446266e-02     7.678424e-01                    100.00000
## Dim.10 3.641802e-32     3.310729e-31                    100.00000
## Dim.11 1.934653e-33     1.758775e-32                    100.00000
```

Feature Selection

Filter Methods

```
# load libraries
library(caret)
```

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

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
# Plot correlation matrix
cor_mat<-cor(df1)
cor_mat
```

```
##                   Branch Customer.type       Gender Product.line   Unit.price
## Branch         1.00000000   -0.01960787 -0.056317558 -0.053937557  0.028202440
## Customer.type -0.01960787    1.00000000  0.039996160 -0.036800311 -0.020237875
## Gender        -0.05631756    0.03999616  1.000000000  0.005193197  0.015444630
## Product.line  -0.05393756   -0.03680031  0.005193197  1.000000000  0.019321028
## Unit.price     0.02820244   -0.02023787  0.015444630  0.019321028  1.000000000
## Quantity       0.01596379   -0.01676271 -0.074258307  0.020256001  0.010777564
## Tax            0.04104666   -0.01967028 -0.049450989  0.031620725  0.633962089
## Payment       -0.05010429    0.01807344  0.044577609  0.029896383 -0.015941048
## cogs           0.04104666   -0.01967028 -0.049450989  0.031620725  0.633962089
## gross.income   0.04104666   -0.01967028 -0.049450989  0.031620725  0.633962089
## Rating         0.01023848    0.01888867  0.004800208 -0.020528973 -0.008777507
##                  Quantity         Tax      Payment        cogs gross.income
## Branch         0.01596379  0.04104666 -0.050104288  0.04104666   0.04104666
## Customer.type -0.01676271 -0.01967028  0.018073436 -0.01967028  -0.01967028
## Gender        -0.07425831 -0.04945099  0.044577609 -0.04945099  -0.04945099
## Product.line   0.02025600  0.03162072  0.029896383  0.03162072   0.03162072
## Unit.price     0.01077756  0.63396209 -0.015941048  0.63396209   0.63396209
## Quantity       1.00000000  0.70551019 -0.003920990  0.70551019   0.70551019
## Tax            0.70551019  1.00000000 -0.012433637  1.00000000   1.00000000
## Payment       -0.00392099 -0.01243364  1.000000000 -0.01243364  -0.01243364
## cogs           0.70551019  1.00000000 -0.012433637  1.00000000   1.00000000
## gross.income   0.70551019  1.00000000 -0.012433637  1.00000000   1.00000000
## Rating        -0.01581490 -0.03644170 -0.005381289 -0.03644170  -0.03644170
##                     Rating
## Branch         0.010238476
## Customer.type  0.018888672
## Gender         0.004800208
## Product.line  -0.020528973
## Unit.price    -0.008777507
## Quantity      -0.015814905
## Tax           -0.036441705
## Payment       -0.005381289
## cogs          -0.036441705
```

```
## gross.income  -0.036441705
## Rating         1.000000000
```

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

# Highly correlated attributes
# ---
#
highlyCorrelated
```

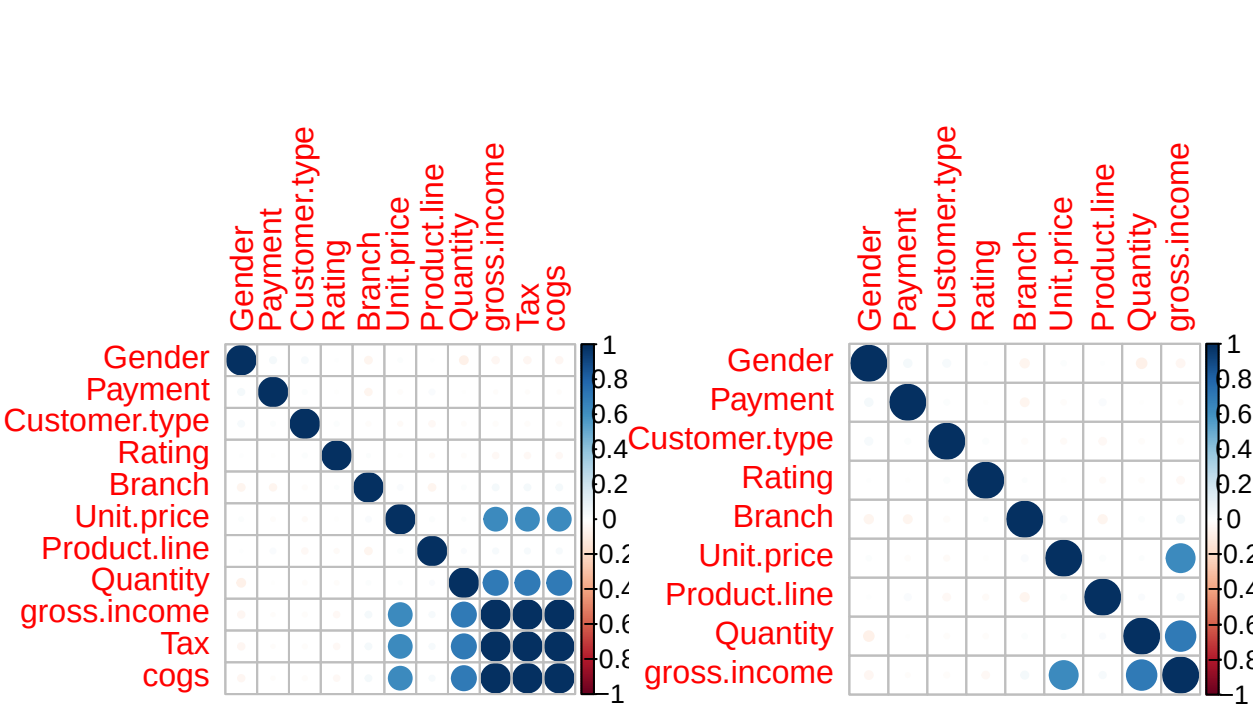
```
## [1] 7 9
```

```
names(df1[,highlyCorrelated])
```

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

```
# we remove highly correlated features
df1_cor<-df1[-highlyCorrelated]
```

```
# Performing our graphical comparison
# ---
#
par(mfrow = c(1, 2))
corrplot(cor_mat, order = "hclust",title = 'Unfiltered correlation matrix')
corrplot(cor(df1_cor), order = "hclust",title='FIltered Matrix')
```



We can conclude that the features that will be used for analysis : gender, payment, customer type, branch,unit price,Product line, quantity and gross income

##Wrapper Methods

```
#Installing libraries
library(clustvarsel)
```

```
## Loading required package: mclust
```

```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:psych':
##
##     sim
```

```
## Package 'clustvarsel' version 2.3.4
```

```
## Type 'citation("clustvarsel")' for citing this R package in publications.
```

```r
library(mclust)
```

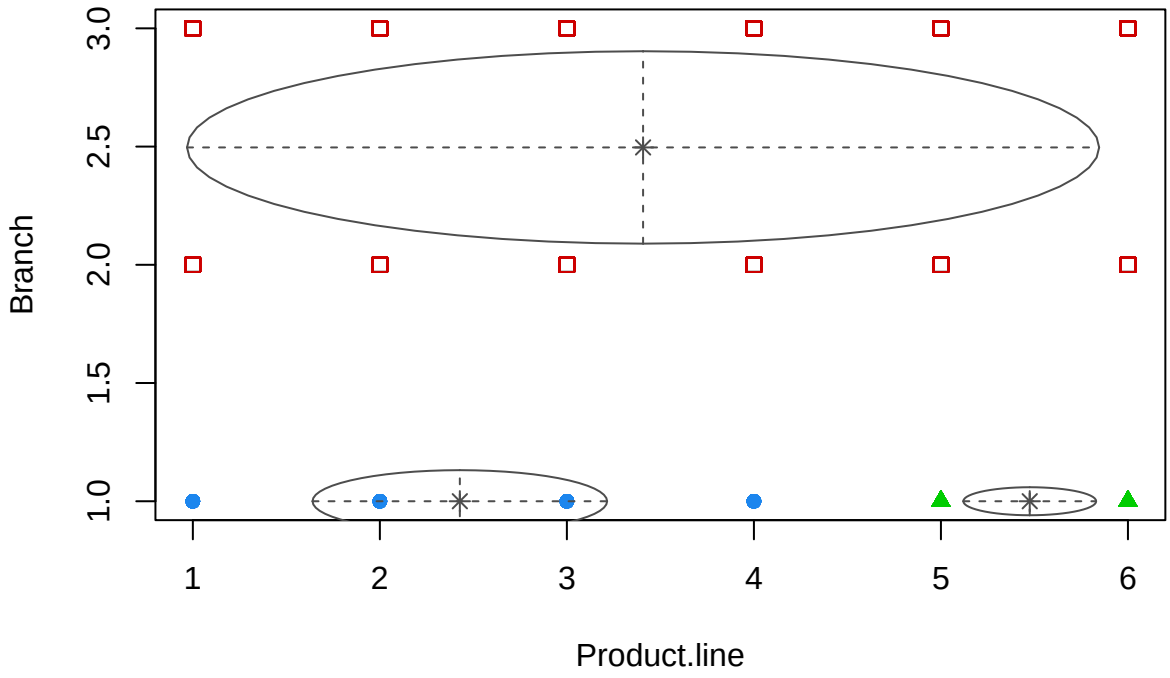
```r
# Sequential forward greedy search (default)
# ---
#
out = clustvarsel(df1 , G = 1:5)
out
```

```
## ----------------------------------------------------
## Variable selection for Gaussian model-based clustering
## Stepwise (forward/backward) greedy search
## ----------------------------------------------------
##
##  Variable proposed Type of step   BICclust Model G   BICdiff Decision
##       Product.line          Add  -3521.631     E 5  408.3674 Accepted
##             Branch          Add  -5530.063   VEI 3  439.5779 Accepted
##           Quantity          Add -10701.378   VEI 5 -175.1074 Rejected
##             Branch       Remove  -3498.098     E 5  416.0449 Rejected
##
## Selected subset: Product.line, Branch
```

```r
# Having identified the variables that we use, we proceed to build the clustering model:
# ---
#

Subset1 = df1[,out$subset]
mod = Mclust(Subset1, G = 1:5)
summary(mod)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEI (diagonal, equal shape) model with 3 components:
##
##  log-likelihood    n df       BIC       ICL
##       -2723.585 1000 12 -5530.063 -5531.496
##
## Clustering table:
##   1   2   3
## 216 660 124
```

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



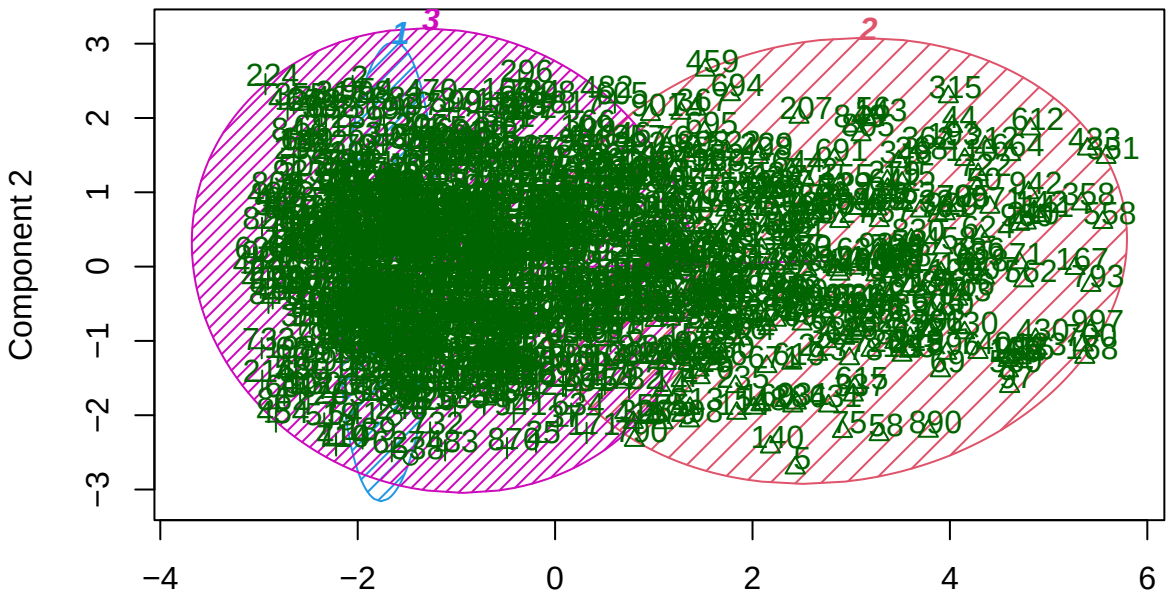
Embedded Methods

```
# install libraries
library(wskm)
```

```
## Loading required package: latticeExtra

##
## Attaching package: 'latticeExtra'

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

## Loading required package: fpc
```

```
set.seed(2)
model <- ewkm(df1, 3, lambda=2, maxiter=1000)
```

```
library(cluster)
clusplot(df1, model$cluster, color=TRUE, shade=TRUE,
         labels=2, lines=1,main='Cluster Analysis for Carrefour Supermarket')
```



These two components explain 46.04 % of the point variability.

```
# Weights remain stored in the model and we can check them as follows:
#
round(model$weights*100,2)
```

```
##   Branch Customer.type Gender Product.line Unit.price Quantity   Tax Payment
## 1      0          0.13   0.14            0          0        0 49.86       0
## 2      0         48.12  51.87            0          0        0  0.00       0
## 3      0         50.00  50.00            0          0        0  0.00       0
##   cogs gross.income Rating
## 1    0        49.86      0
## 2    0         0.00      0
## 3    0         0.00      0
```