Importing the libraries and previewing the data

```
library(stats)
library(plyr)
library(dplyr)
```

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

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

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

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

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## 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()
```

```
library(ggplot2)
library(caret)
```

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

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

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

```
library(psych)
```

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

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

```
library(rpart)
library(devtools)
```

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

```
#library(ggbiplot)
```

```
df = read.csv("Supermarket_Dataset_1 - Sales Data.csv",sep=",")
head(df)
```

Loading and previewing the data

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

```
head(df)
```

Checking the 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
```

```
summary(df)
```

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

```
##  1st Qu.:4.762           1st Qu.: 5.9249   1st Qu.: 5.500   1st Qu.: 124.42
##  Median :4.762           Median :12.0880   Median : 7.000   Median : 253.85
##  Mean   :4.762           Mean   :15.3794   Mean   : 6.973   Mean   : 322.97
##  3rd Qu.:4.762           3rd Qu.:22.4453   3rd Qu.: 8.500   3rd Qu.: 471.35
##  Max.   :4.762           Max.   :49.6500   Max.   :10.000   Max.   :1042.65
```

```
# Checking null values
colSums(is.na(df))
```

Cleaning the data

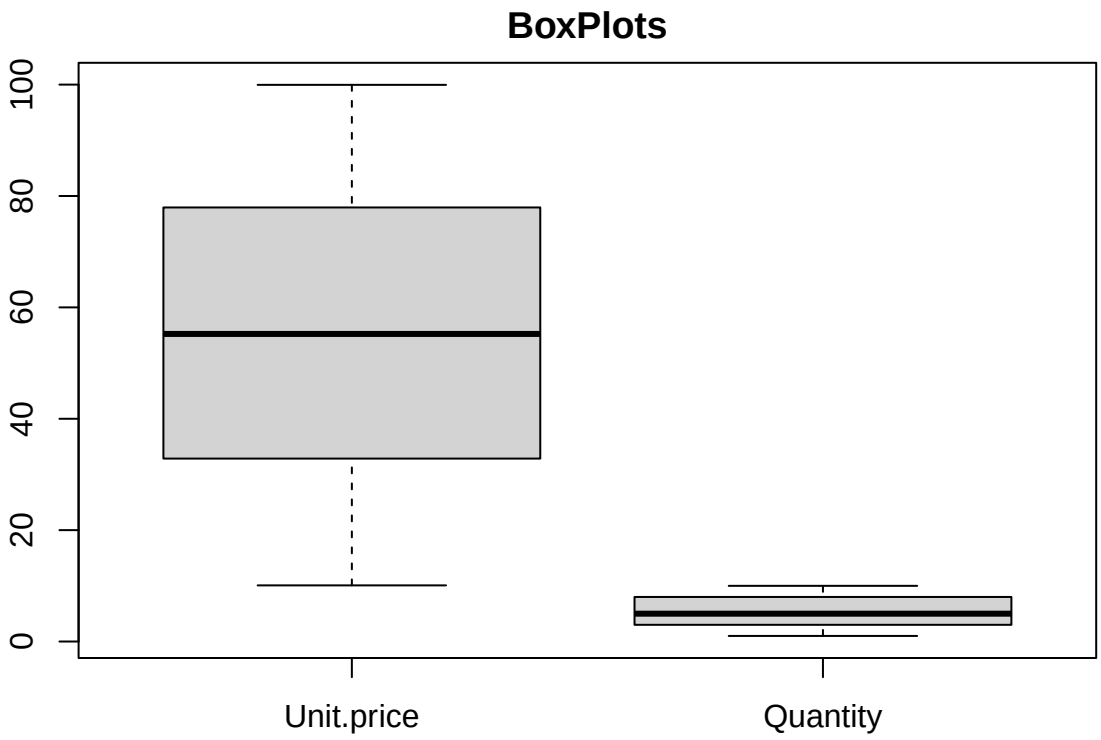
```
##               Invoice.ID                  Branch           Customer.type
##                        0                       0                       0
##                   Gender            Product.line              Unit.price
##                        0                       0                 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

```
# Checking to see whether we have duplicates in our data
dim(df[duplicated(df), ])
```

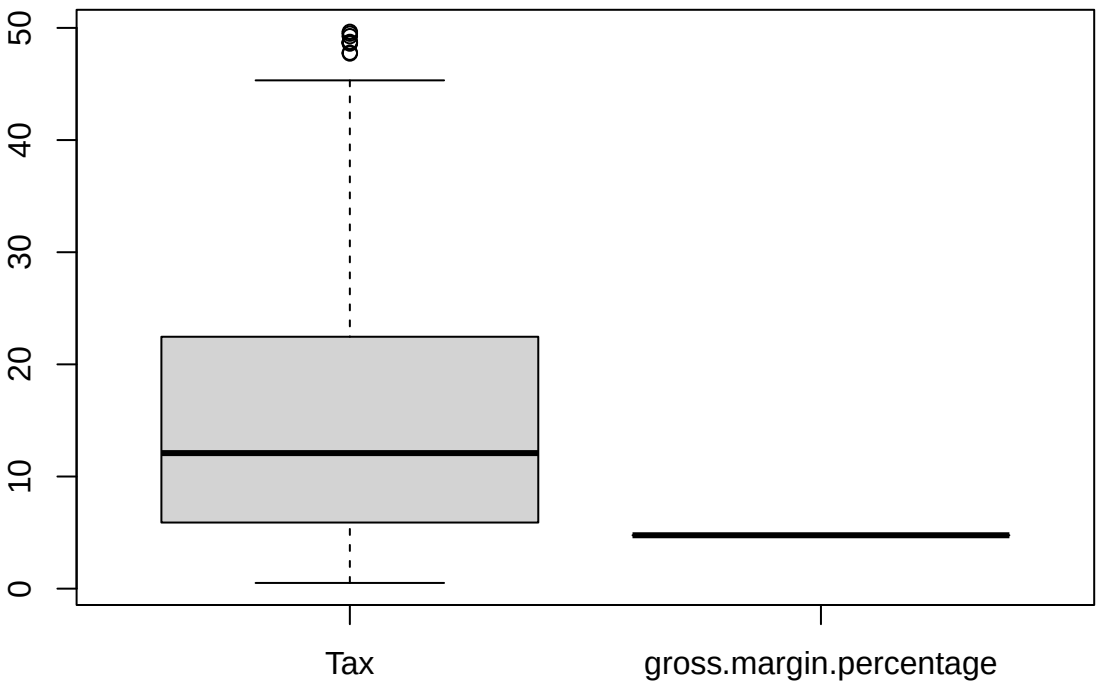
```
## [1]  0 16
```

```
# check for outliers/anomalies
numerical = df[, !sapply(df, is.character)]
```

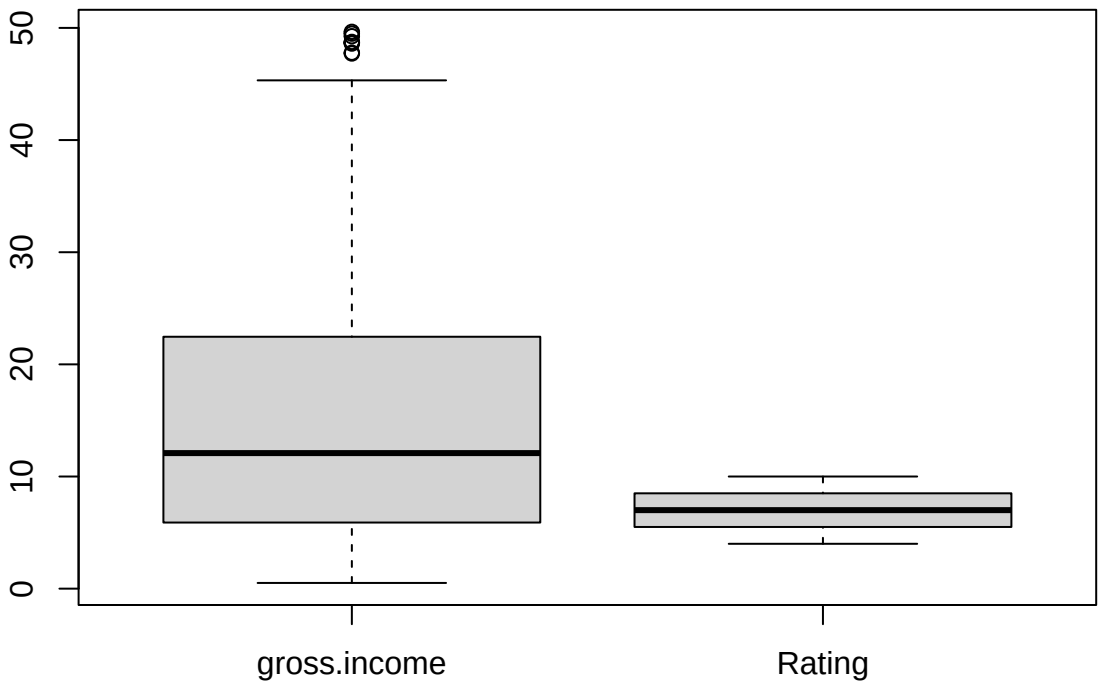
```
par(mfrow = c(1,1), mar = c(5,4,2,2))
boxplot(numerical[, c(1:2)], main='BoxPlots')
```



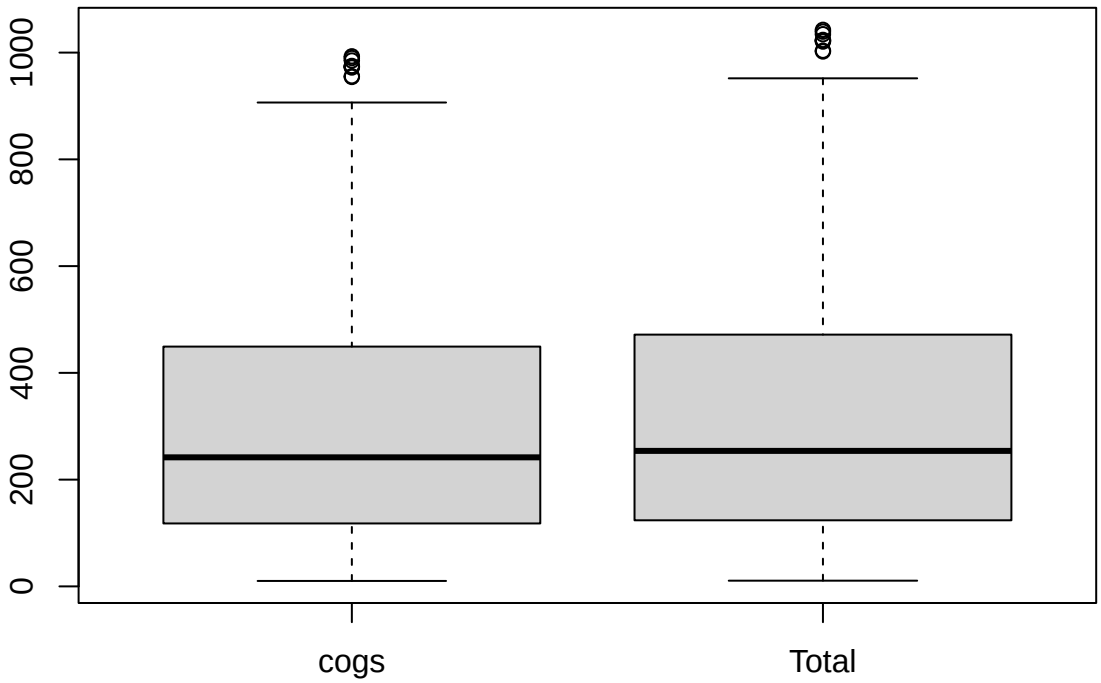

```
boxplot(numerical[, c(3,5)])
```




```
boxplot(numerical[, c(6,7)])
```




```
boxplot(numerical[ ,c(4,8)])
```



PCA

```
# creating a dataset for PCA
sales = df[,c(6,7,8,12,13,14,15,16)]
head(sales)
```

```
##   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
```

```
# Removing  gross margin percentage column
sales = subset(sales, select = -c(gross.margin.percentage, Total) )
```

```
#Cheking whether the column has been removed
head(sales)
```

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

```
#solution
pca <- prcomp(sales, scale=TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4      PC5       PC6
## Standard deviation     1.9817 1.0002 0.9939 0.2909 2.51e-16 1.477e-16
## Proportion of Variance 0.6545 0.1667 0.1646 0.0141 0.00e+00 0.000e+00
## Cumulative Proportion  0.6545 0.8213 0.9859 1.0000 1.00e+00 1.000e+00
```

The first three principal components accounted for 98% of the total variance

```
#library(Rcpp)
#ggbiplot(pca, groups = as.factor(df$Branch), ellipse = TRUE, circle = TRUE)
```

```
#ggbiplot(pca, groups = as.factor(df$Customer.type), ellipse = TRUE, circle = TRUE)
```

```
#ggbiplot(pca, groups = as.factor(df$Product.line), ellipse = TRUE, circle = TRUE)
```

```
#ggbiplot(pca, groups = as.factor(df$Payment), ellipse = TRUE, circle = TRUE)
```

FEATURE SELECTION USING R

```
library(caret)
library(corrplot)
```

Filter Method

```
## corrplot 0.92 loaded
```

```
head(sales, 5)
```

```
##   Unit.price Quantity     Tax   cogs gross.income Rating
## 1      74.69        7 26.1415 522.83      26.1415    9.1
## 2      15.28        5  3.8200  76.40       3.8200    9.6
## 3      46.33        7 16.2155 324.31      16.2155    7.4
## 4      58.22        8 23.2880 465.76      23.2880    8.4
## 5      86.31        7 30.2085 604.17      30.2085    5.3
```

```
correlationMatrix = cor(sales)
correlationMatrix
```

```
##                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
##                     Rating
## Unit.price   -0.008777507
## Quantity     -0.015814905
## Tax          -0.036441705
## cogs         -0.036441705
## gross.income -0.036441705
## Rating        1.000000000
```

```
highlyCorrelated = findCorrelation(correlationMatrix, cutoff=0.75)
highlyCorrelated
```

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

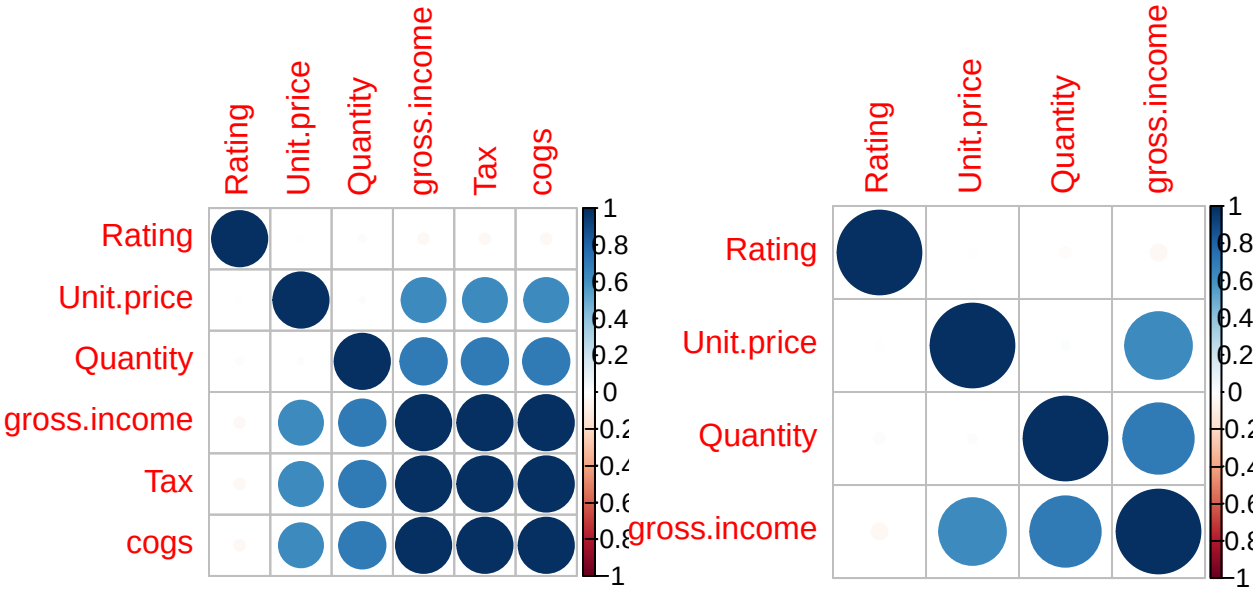
```
# Highly correlated attributes
highlyCorrelated
```

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

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

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

```
# Removing Redundant Features
# ---
#
hc = sales[-highlyCorrelated]
# Performing our graphical comparison
# ---
#
par(mfrow = c(1, 2))
corrplot(correlationMatrix, order = "hclust")
corrplot(cor(hc), order = "hclust")
```




```
# Sequential forward greedy search (default)
library(clustvarsel)
```

Wrapper Method

```
## Loading required package: mclust

## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.

##
## Attaching package: 'mclust'

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

## The following object is masked from 'package:purrr':
##
##     map

## Package 'clustvarsel' version 2.3.4
```

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

```
library(mclust)
out = clustvarsel(sales)
out
```

```
## ----------------------------------------------------------
## Variable selection for Gaussian model-based clustering
## Stepwise (forward/backward) greedy search
## ----------------------------------------------------------
##
##  Variable proposed Type of step   BICclust Model G     BICdiff Decision
##           Quantity          Add  -4308.761     E 9    687.4466 Accepted
##               cogs          Add -16650.156   VEV 9    739.7085 Accepted
##         Unit.price          Add -19381.035   VEV 9   5167.7215 Accepted
##           Quantity       Remove   5591.554   VEV 9 -21656.9934 Accepted
##           Quantity          Add -20596.091   EVV 9 -22872.0493 Rejected
##         Unit.price       Remove -13532.537     E 2  28021.5645 Rejected
##
## Selected subset: cogs, Unit.price
```

After employing wrapper method of feature selection, we get quantity, cogs and unit price as the most relevant features to use in building our machine learning model.