TITLE: CARREFOUR SALES HISTORY - DIMENSIONALITY REDUCTION AND FEATURE SELECTION

AUTHOR: JOSEPH NJUGUNA

DATE: 10/6/22

1.Defining the question

a) Specifying the question

Reduce dataset to a low dimensional dataset using the t-SNE algorithm or PCA.

Perform feature selection through the use of the unsupervised learning methods

b) Defining the metric for success

Reduction of variables via PCA.

Feature selection of important variables.

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). 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.

d) Recording the experimental design

- Exploratory data analysis

- Cleaning data

- Implementing the solution

- Conclusions

- Recommendations

- Follow up questions

2. Reading the data

```
salesdf <- read.csv("Sales Data part(1-2).csv", header = TRUE, sep = ",")
```

3. Exploring the data

```
### viewing first 5 rows of our dataset
head(salesdf)
```

```
##    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
```

```
### viewing last 5 rows of our dataset
tail(salesdf)
```

```
##       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
```

```r
### glimpse of unique values
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
```

```r
glimpse(salesdf)
```

```
## Rows: 1,000
## Columns: 16
## $ Invoice.ID              <chr> "750-67-8428", "226-31-3081", "631-41-3108", "~
## $ Branch                  <chr> "A", "C", "A", "A", "A", "C", "A", "C", "A", "~
## $ Customer.type           <chr> "Member", "Normal", "Normal", "Member", "Norma~
## $ Gender                  <chr> "Female", "Female", "Male", "Male", "Male", "M~
## $ Product.line            <chr> "Health and beauty", "Electronic accessories",~
## $ Unit.price              <dbl> 74.69, 15.28, 46.33, 58.22, 86.31, 85.39, 68.8~
## $ Quantity                <int> 7, 5, 7, 8, 7, 7, 6, 10, 2, 3, 4, 4, 5, 10, 10~
## $ Tax                     <dbl> 26.1415, 3.8200, 16.2155, 23.2880, 30.2085, 29~
## $ Date                    <chr> "1/5/2019", "3/8/2019", "3/3/2019", "1/27/2019~
## $ Time                    <chr> "13:08", "10:29", "13:23", "20:33", "10:37", "~
## $ Payment                 <chr> "Ewallet", "Cash", "Credit card", "Ewallet", "~
## $ cogs                    <dbl> 522.83, 76.40, 324.31, 465.76, 604.17, 597.73,~
## $ gross.margin.percentage <dbl> 4.761905, 4.761905, 4.761905, 4.761905, 4.7619~
## $ gross.income            <dbl> 26.1415, 3.8200, 16.2155, 23.2880, 30.2085, 29~
## $ Rating                  <dbl> 9.1, 9.6, 7.4, 8.4, 5.3, 4.1, 5.8, 8.0, 7.2, 5~
## $ Total                   <dbl> 548.9715, 80.2200, 340.5255, 489.0480, 634.378~
```

```r
### checking data types and their class
str(salesdf)
```

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

Our dataset has 16 columns: 8 categorical and 8 numerical.

```
### dimensions of our dataset
dim(salesdf)
```

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

The dataset has 1000 instances and 16 columns.

```
### brief statistical summary on our dataset
summary(salesdf)
```

```
##   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
```

```
### description of our dataset
library(psych)
describe(salesdf)
```

```
##                          vars    n   mean     sd median trimmed    mad   min
## Invoice.ID*                 1 1000 500.50 288.82 500.50  500.50 370.65  1.00
## Branch*                     2 1000   1.99   0.82   2.00    1.99   1.48  1.00
## Customer.type*              3 1000   1.50   0.50   1.00    1.50   0.00  1.00
## Gender*                     4 1000   1.50   0.50   1.00    1.50   0.00  1.00
## Product.line*               5 1000   3.45   1.72   3.00    3.44   1.48  1.00
## Unit.price                  6 1000  55.67  26.49  55.23   55.62  33.37 10.08
## Quantity                    7 1000   5.51   2.92   5.00    5.51   2.97  1.00
## Tax                         8 1000  15.38  11.71  12.09   14.00  11.13  0.51
## Date*                       9 1000  45.58  25.89  47.00   45.63  34.10  1.00
## Time*                      10 1000 252.18 147.07 249.00  252.49 190.51  1.00
## Payment*                   11 1000   2.00   0.83   2.00    2.00   1.48  1.00
## cogs                       12 1000 307.59 234.18 241.76  279.91 222.65 10.17
## gross.margin.percentage    13 1000   4.76   0.00   4.76    4.76   0.00  4.76
## gross.income               14 1000  15.38  11.71  12.09   14.00  11.13  0.51
## Rating                     15 1000   6.97   1.72   7.00    6.97   2.22  4.00
## Total                      16 1000 322.97 245.89 253.85  293.91 233.78 10.68
##                              max   range  skew kurtosis   se
## Invoice.ID*              1000.00  999.00  0.00    -1.20 9.13
## Branch*                     3.00    2.00  0.02    -1.51 0.03
## Customer.type*              2.00    1.00  0.00    -2.00 0.02
## Gender*                     2.00    1.00  0.00    -2.00 0.02
## Product.line*               6.00    5.00  0.06    -1.28 0.05
## Unit.price                 99.96   89.88  0.01    -1.22 0.84
## Quantity                   10.00    9.00  0.01    -1.22 0.09
## Tax                        49.65   49.14  0.89    -0.09 0.37
## Date*                      89.00   88.00 -0.03    -1.23 0.82
## Time*                     506.00  505.00  0.00    -1.25 4.65
## Payment*                    3.00    2.00  0.00    -1.55 0.03
## cogs                      993.00  982.83  0.89    -0.09 7.41
## gross.margin.percentage     4.76    0.00   NaN      NaN 0.00
## gross.income               49.65   49.14  0.89    -0.09 0.37
## Rating                     10.00    6.00  0.01    -1.16 0.05
## Total                    1042.65 1031.97  0.89    -0.09 7.78
```

From the statistical summary on the dataset, It is observable that the highest amount of tax charged on a product was 49.6/=

Highest earning income as per gross income is 49.6/=

Mean tax paid on products is 15.37/=

4. Cleaning the data

Uniformity

```
### aligning case of our columns to lower case for all
names(salesdf) <- tolower(names(salesdf))
```

```
### lets check for duplicate values
duplicates <- salesdf[duplicated(salesdf),]
duplicates
```

```
##  [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)
```

The dataset has no duplicate values.

```
### detecting missing values
colSums(is.na(salesdf))
```

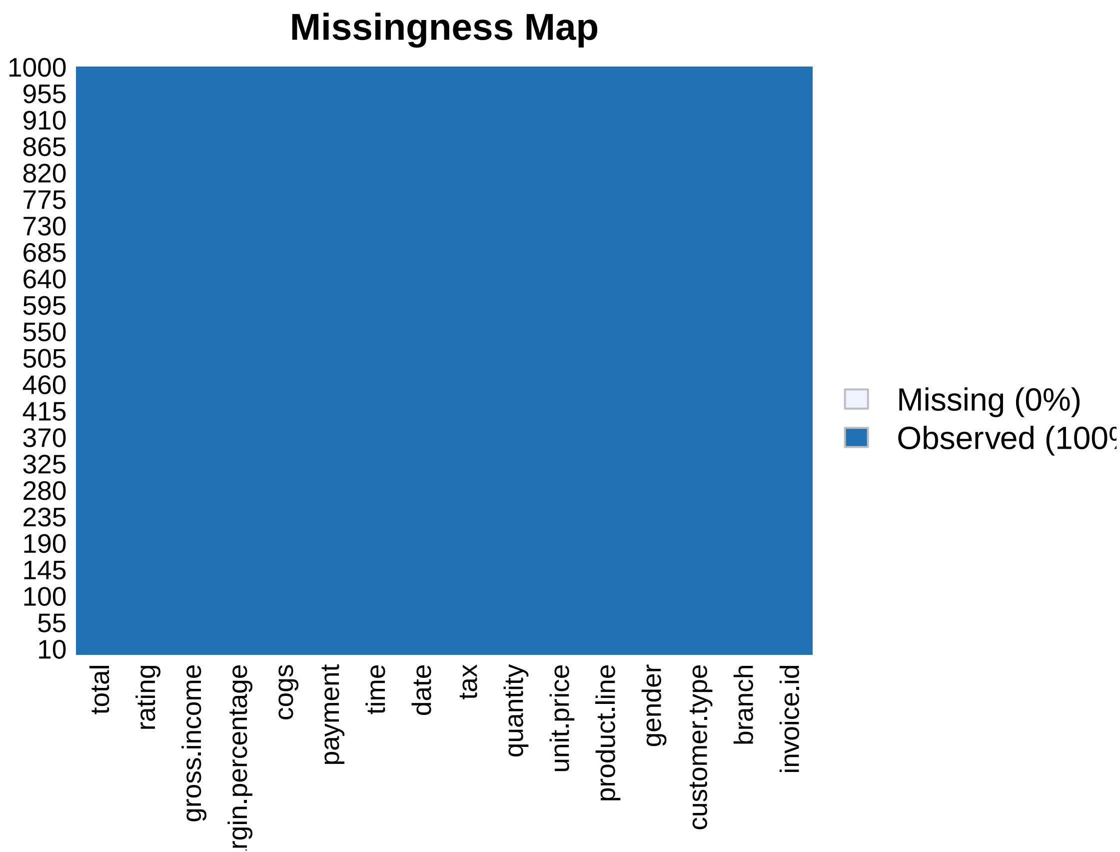
```
##              invoice.id                  branch           customer.type
##                       0                       0                       0
##                  gender            product.line              unit.price
##                       0                       0                       0
##                quantity                     tax                    date
##                       0                       0                       0
##                    time                 payment                    cogs
##                       0                       0                       0
## gross.margin.percentage            gross.income                  rating
##                       0                       0                       0
##                   total
##                       0
```

```
### miss map visual of whether any missing data exists
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(salesdf)
```



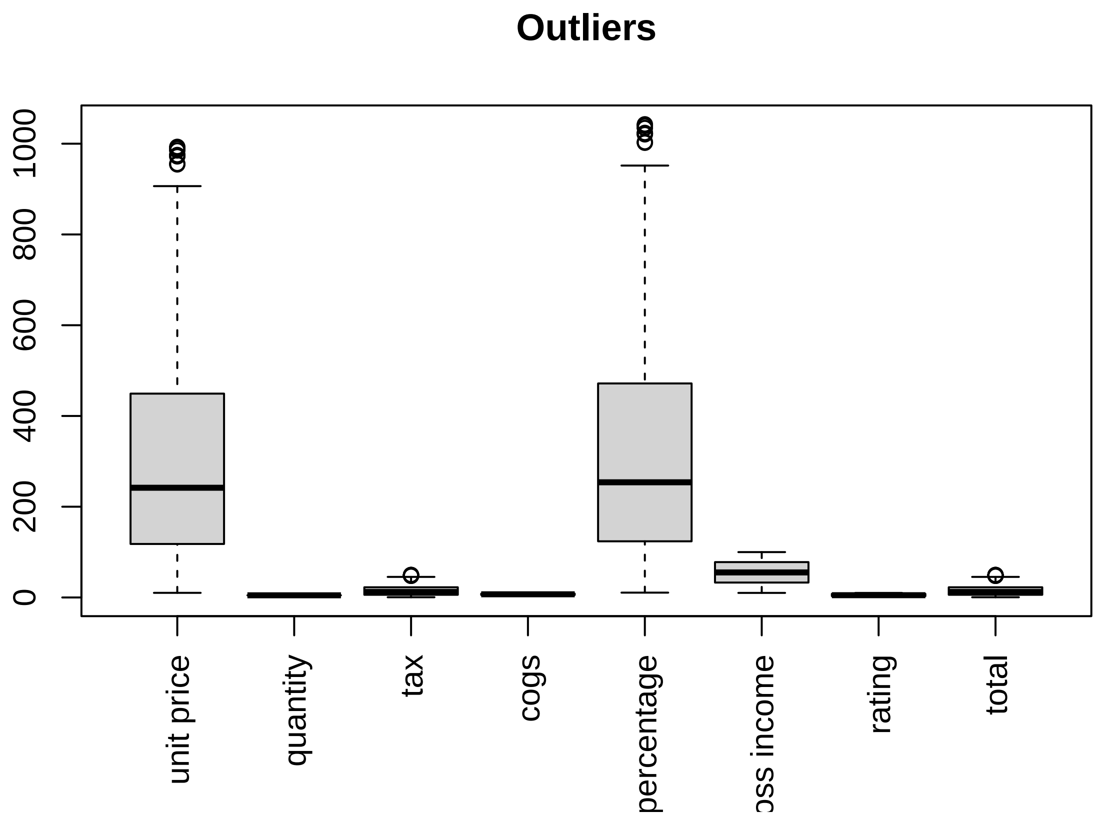
We have absolutely no missing data.

Outliers

```
### numerical columns
### creating list of numerical columns
salesdf.num <- salesdf[c(12:16, 6,7,8)]
str(salesdf.num)
```

```
## 'data.frame':    1000 obs. of  8 variables:
##  $ 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 ...
##  $ 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 ...
```

```
### creating boxplots
boxplot(salesdf.num, names =c('unit price', 'quantity', 'tax', 'cogs', 'gross margin percentage', 'gros
```



Outliers exist in the tax, cogs, gross income, total columns. This is understable as gross income is never meant to be equal; Total is affected by quantity and hence large quantities have a huge total whilst low have smaller total. ### Outliers shall not be removed as vital information/insight could be lost.

5. Data Analysis

Univariate Analysis

Measures of central tendecy

```
### mean
colMeans(salesdf[sapply(salesdf, is.numeric)])
```

```
##             unit.price               quantity                    tax
##              55.672130               5.510000              15.379369
##                   cogs gross.margin.percentage           gross.income
##             307.587380               4.761905              15.379369
##                 rating                  total
##               6.972700             322.966749
```

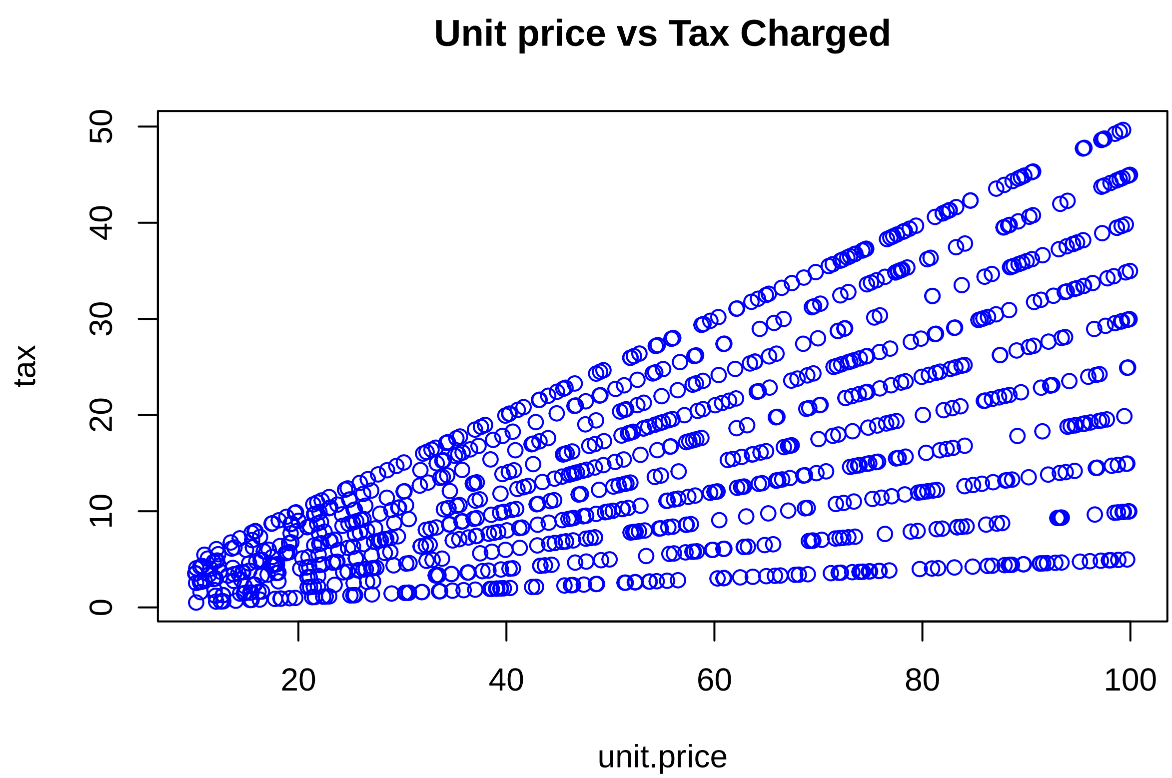
Mean gross income is 15.4/=

Mean rating is 6.9

Bivariate Analysis

Numerical-Numerical variables

```
# correlation between unit price and tax
plot(tax ~ unit.price, dat = salesdf,
      col = "Blue",
      main = "Unit price vs Tax Charged")
```



As price of a unit increases, tax also increases

5. Implementing the solution

A) Principal Component Analysis

PCA can only be applied to numerical data.

It is imperative that a feature set must be normalized.

Is an unsupervised learning algorithm

```
### prcomp doesnt function when handling columns with constant variance, we therefore remove columns tha
### calling out columns with constant variance
finalsalesdf.num <- salesdf.num[ , which(apply(salesdf.num, 2, var) != 0)]

### pass prcomp to salesdf numeric
### parameters, scale and center
salesdf.pca <- prcomp(finalsalesdf.num, center = TRUE, scale. = TRUE)
summary(salesdf.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4       PC5      PC6
## Standard deviation     2.2185 1.0002 0.9939 0.30001 3.953e-16 1.01e-16
## Proportion of Variance 0.7031 0.1429 0.1411 0.01286 0.000e+00 0.00e+00
## Cumulative Proportion  0.7031 0.8460 0.9871 1.00000 1.000e+00 1.00e+00
##                              PC7
## Standard deviation     2.906e-31
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

PC1 explains 70% of the total ariance.

PC2 explains 14% of the variance.

```
### Eigenvalues
library(factoextra)
```

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

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

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

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

```r
library(corrplot)
```

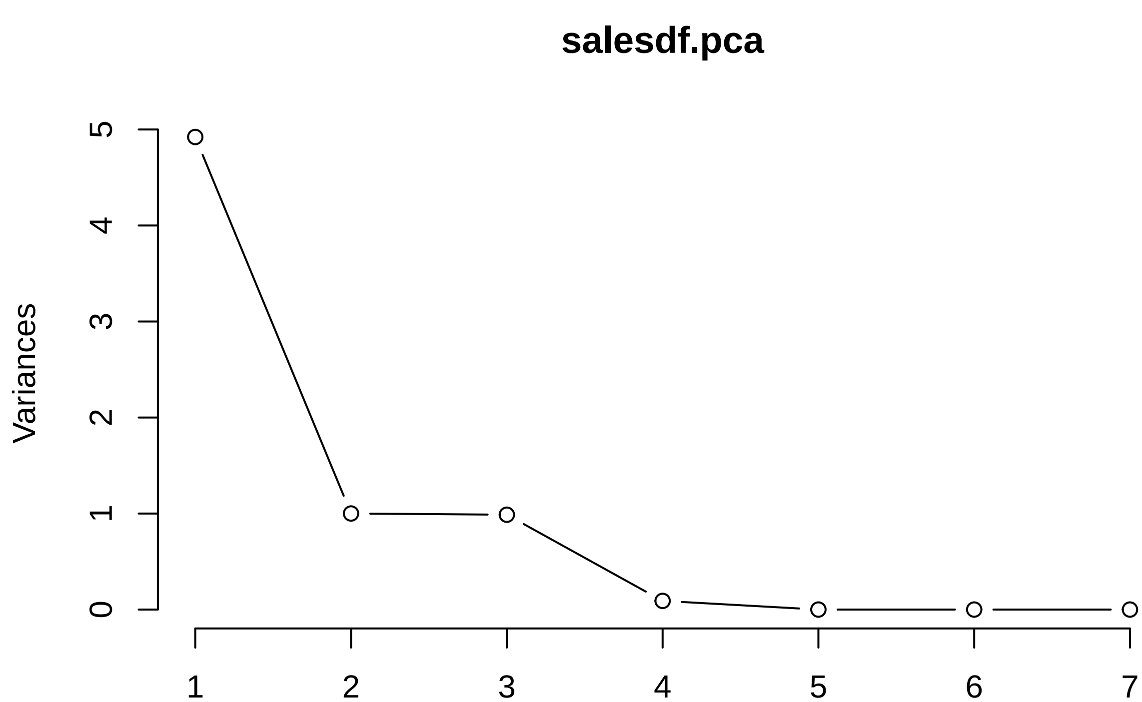
```
## corrplot 0.92 loaded
```

```r
eig.val=get_eigenvalue(salesdf.pca)
eig.val
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1 4.921797e+00     7.031139e+01                    70.31139
## Dim.2 1.000400e+00     1.429143e+01                    84.60282
## Dim.3 9.877961e-01     1.411137e+01                    98.71419
## Dim.4 9.000673e-02     1.285810e+00                   100.00000
## Dim.5 1.562713e-31     2.232448e-30                   100.00000
## Dim.6 1.019773e-32     1.456819e-31                   100.00000
## Dim.7 8.447741e-62     1.206820e-60                   100.00000
```

It is observed that eigenvalues decrease steadily from PC1; this indicates that the first principal component is strongest.

```r
### arm bend
plot.salesdf.pca <- plot(salesdf.pca, type="l")
```



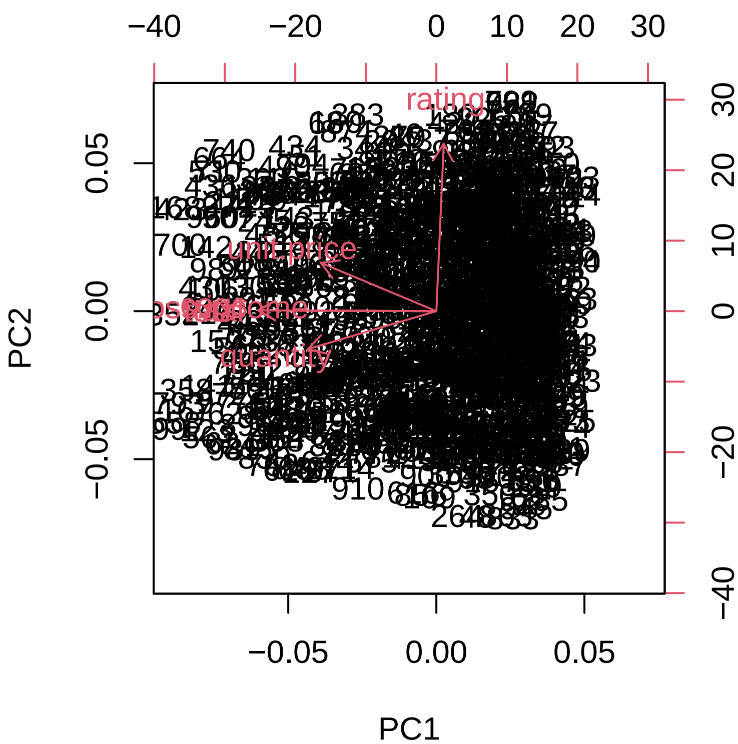
```r
plot.salesdf.pca
```

```
## NULL
```

The plot shows the bend at PC2 and PC3

An arm bend reps decrease in cumulative contribution.

```r
# better understanding of linear transformation we use a biplot
biplot.salesdf.pca <- biplot(salesdf.pca)
```




```r
biplot.salesdf.pca
```

```
## NULL
```

X-axis reps Pc1

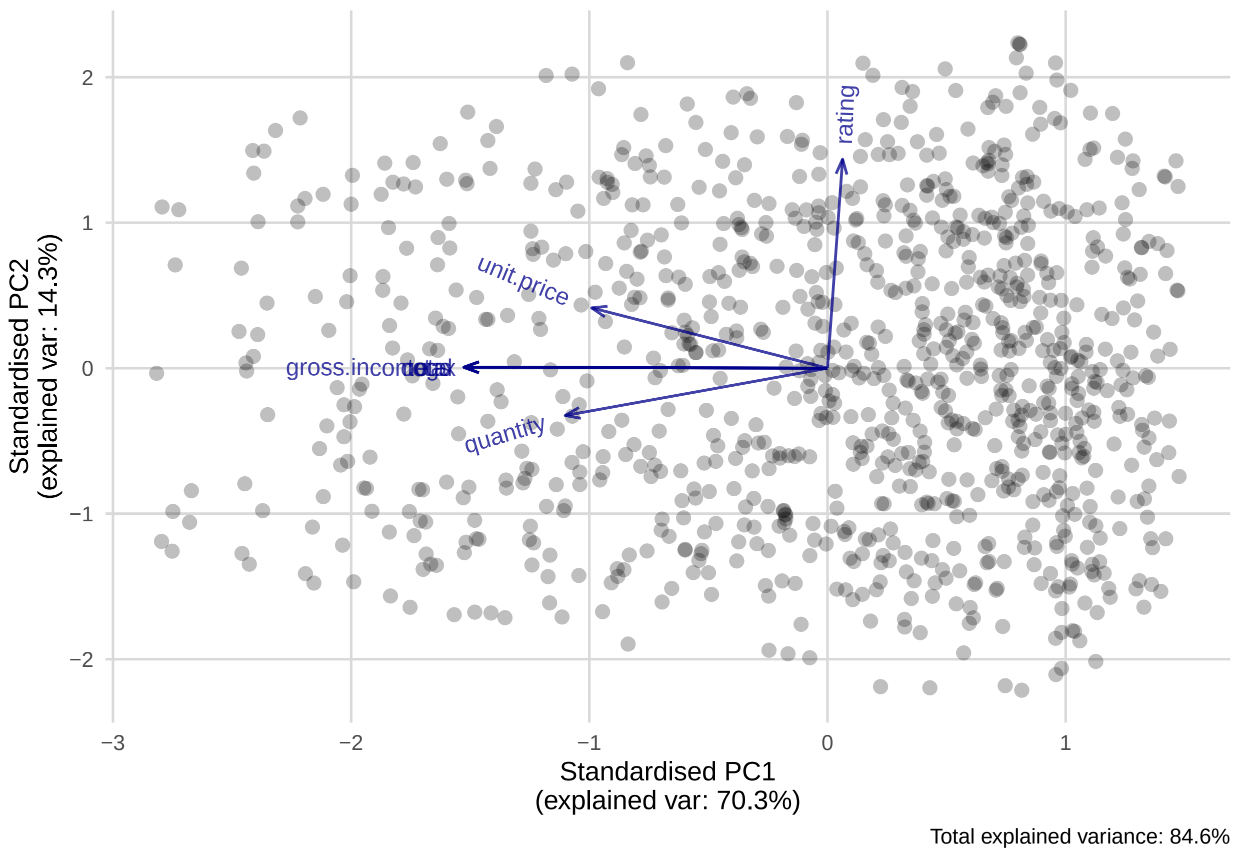
Y=y-axis reps PC2

```r
### ggplot of linear transformation
library(ggplot2)
library(AMR)
```

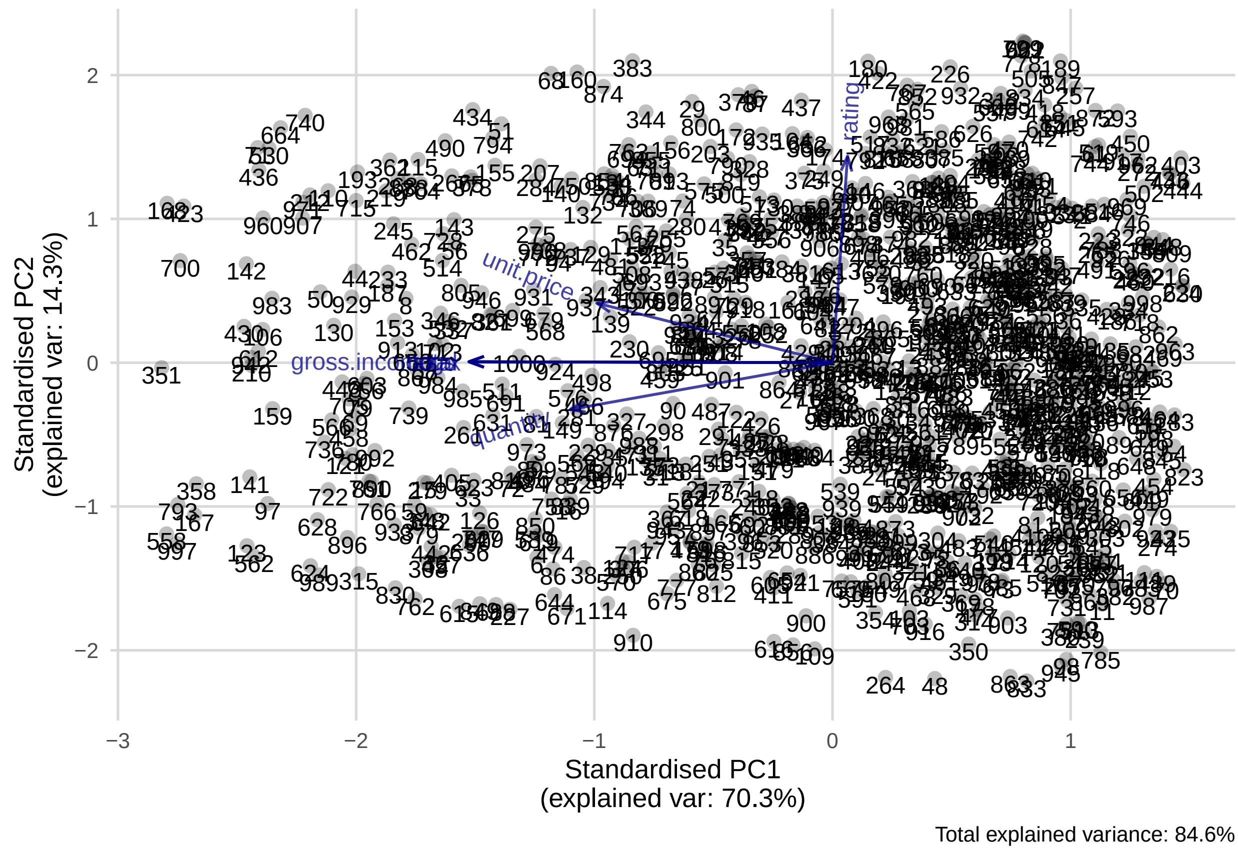
```
##
## Attaching package: 'AMR'

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

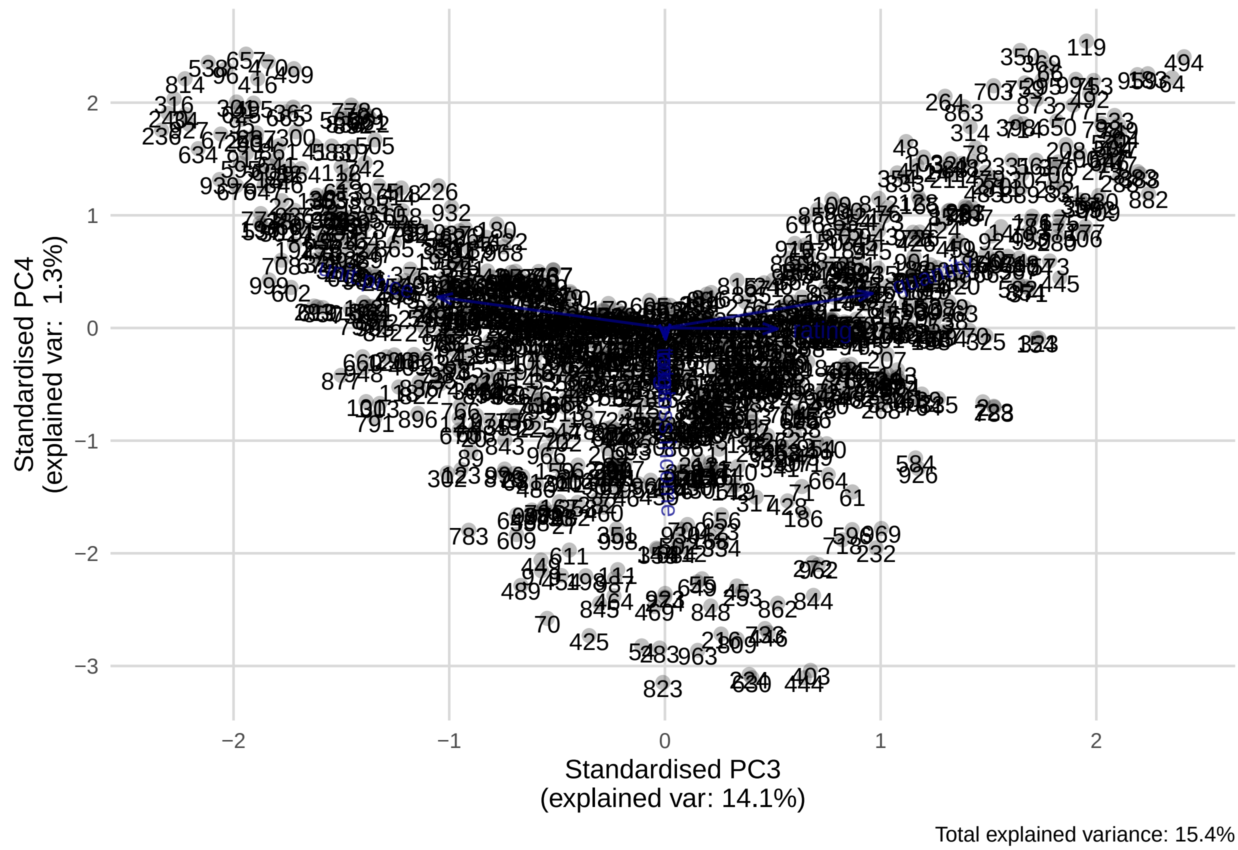
```r
ggplot_pca(salesdf.pca)
```




```r
### adding detail to plot
ggplot_pca(salesdf.pca, labels=rownames(salesdf), obs.scale = 1, var.scale = 1)
```




```
### plot of PC3 and PC4
ggplot_pca(salesdf.pca,ellipse=TRUE,choices=c(3,4),    labels=rownames(salesdf))
```



Due to the minute explained variance explained by PC2 and PC3 we cannot drow any meaningful insights.

B) Feature Selection

i) Filter Method.

I love this method due to the ease.

very concise!

```
### loading necessary librarires
library(caret)
```

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

```
library(corrplot)
### Recall that we had already assigned a variable to our numerical columns - salesdf.num
correlationMatrix <- cor(finalsalesdf.num)

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

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

```
## [1] 1 4 2
```

```r
names(finalsalesdf.num[,highlyCorrelated])
```

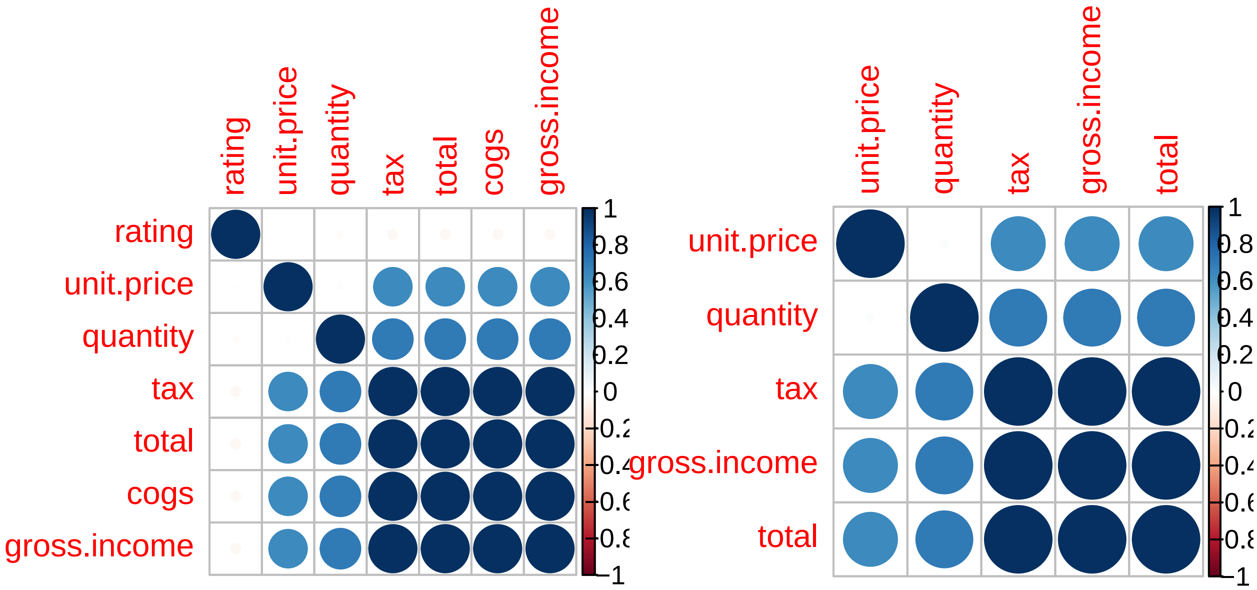
```
## [1] "cogs"         "total"        "gross.income"
```

Cogs, total and gross income columns are highly correlated.

We shall therefore remove them.

```r
# We can remove the variables with a higher correlation
# and comparing the results graphically as shown below
# ---
#
# Removing Redundant Features
# ---
#
filtered.salesdf<-salesdf.num[-highlyCorrelated]

# Performing our graphical comparison
# ---
#
par(mfrow = c(1, 2))
corrplot(correlationMatrix, order = "hclust")
corrplot(cor(filtered.salesdf), order = "hclust")
```


rating
unit.price
quantity
tax
total
cogs
gross.income
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
unit.price
quantity
tax
gross.income
total

6. Conclusion

PC1 explains 70% of the total ariance.

PC2 explains 14% of the variance.

Eigenvalues decrease steadily from PC1

Cogs, total and gross income columns are highly correlated.

Mean gross income is 15.4/=

Mean rating is 6.9

As price of a unit increases, tax also increases

7. Recommendation

Discounts on common items should be offered, to give incentive to low income earners to purchase these products.

8. Follow up questions

a) Did we have right data?

```
Yes.
```

b) Do we need other data to answer our question?

```
No.
```

c) Did we have the right question?

```
Yes.
```