# Part 1: Dimensionality Reduction

2022-03-29

## Research Question

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.

## Part 1: Dimensionality Reduction

This section of the project entails reducing your dataset to a low dimensional dataset using the t-SNE algorithm or PCA. You will be required to perform your analysis and provide insights gained from your analysis.

## Defining the question

### i)Specifying the Data Analytic Question

Reduce your dataset to a low dimensional dataset using PCA.

### ii)Defining the Metric for Success

Reduce your dataset to a low dimensional dataset.

### iii) Understanding the Context

This section of the project entails reducing your dataset to a low dimensional dataset using the t-SNE algorithm or PCA. You will be required to perform your analysis and provide insights gained from your analysis.

Dataset link http://bit.ly/CarreFourDataset

```r
#Necessary libraries
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(ggbiplot)
```

```
## Loading required package: ggplot2

## 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

## Loading required package: grid
```

```
library(tidyr)
```

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

```
#Lets preview the head
head(df)
```

```
##    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 6 records in our dataset
tail(df)
```

```
##       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
```

```
#Check the dimensions
dim(df)
```

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

1000 observations of 16 variables

```
#The class of the dataset
class(df)
```

```
## [1] "data.frame"
```

our dataset is a dataframe

```r
#Check the Summary of the dataframe
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
```

```r
#Check for null/missing values
colSums(is.na(df))
```

```
##              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
```

No null values in our dataset

```r
#Check for duplicate values
duplicated_rows <- df[duplicated(df),]
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)
```

No duplicates in our dataset

# EXPLORATORY DATA ANALYSIS

Univariate Data Analysis

```r
# Mean
df %>% summarise_if(is.numeric, mean)
```

```
##   Unit.price Quantity      Tax     cogs gross.margin.percentage gross.income
## 1   55.67213     5.51 15.37937 307.5874                4.761905     15.37937
##   Rating    Total
## 1 6.9727 322.9667
```

```r
# Median
df %>% summarise_if(is.numeric, median)
```

```
##   Unit.price Quantity    Tax   cogs gross.margin.percentage gross.income Rating
## 1      55.23        5 12.088 241.76                4.761905       12.088      7
##     Total
## 1 253.848
```

```r
# Mode
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
df %>% summarise_if(is.numeric, getmode)
```

```
##   Unit.price Quantity   Tax  cogs gross.margin.percentage gross.income Rating
## 1      83.77       10 39.48 789.6                4.761905        39.48      6
##    Total
## 1 829.08
```

```r
# Range
df %>% summarise_if(is.numeric, range)
```

```
##   Unit.price Quantity     Tax   cogs gross.margin.percentage gross.income
## 1      10.08        1  0.5085  10.17                4.761905       0.5085
## 2      99.96       10 49.6500 993.00                4.761905      49.6500
##   Rating     Total
## 1      4   10.6785
## 2     10 1042.6500
```

```
# Quantiles
df %>% summarise_if(is.numeric, quantile)
```

```
##   Unit.price Quantity       Tax     cogs gross.margin.percentage gross.income
## 1     10.080        1  0.508500  10.1700                4.761905     0.508500
## 2     32.875        3  5.924875 118.4975                4.761905     5.924875
## 3     55.230        5 12.088000 241.7600                4.761905    12.088000
## 4     77.935        8 22.445250 448.9050                4.761905    22.445250
## 5     99.960       10 49.650000 993.0000                4.761905    49.650000
##   Rating     Total
## 1    4.0   10.6785
## 2    5.5  124.4224
## 3    7.0  253.8480
## 4    8.5  471.3502
## 5   10.0 1042.6500
```

```
# Standard Deviation
df %>% summarise_if(is.numeric, sd)
```

```
##   Unit.price Quantity      Tax     cogs gross.margin.percentage gross.income
## 1   26.49463 2.923431 11.70883 234.1765                       0     11.70883
##    Rating    Total
## 1 1.71858 245.8853
```

```
# Variance
df %>% summarise_if(is.numeric, var)
```

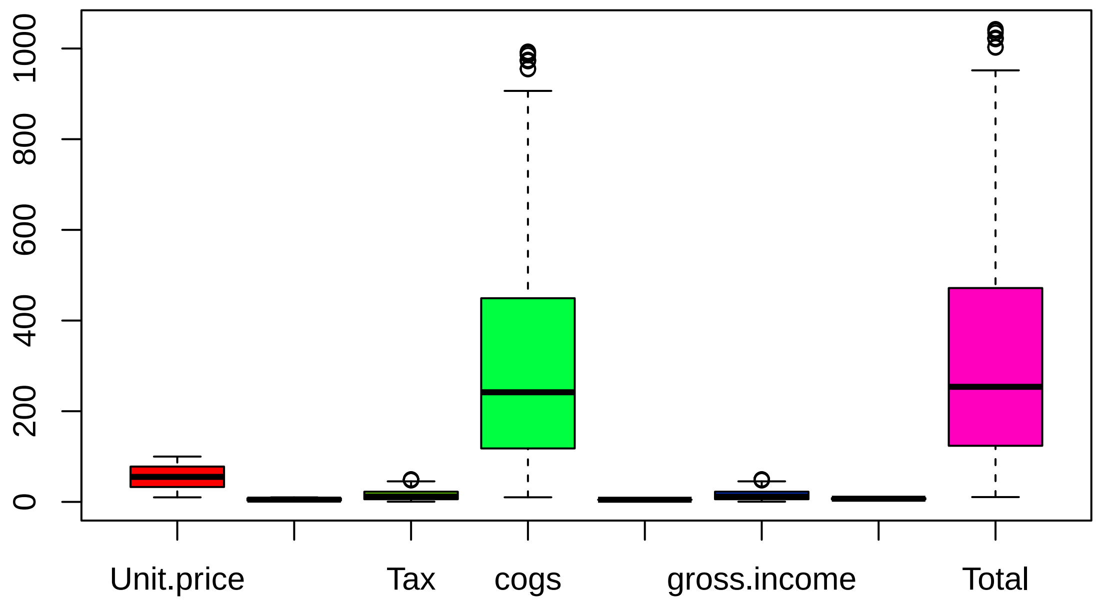
```
##   Unit.price Quantity      Tax     cogs gross.margin.percentage gross.income
## 1   701.9653 8.546446 137.0966 54838.64                       0     137.0966
##     Rating   Total
## 1 2.953518 60459.6
```

```
#selecting the numerical variables
numeric <- df %>% select_if(is.numeric)
head(numeric)
```

```
##   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
```

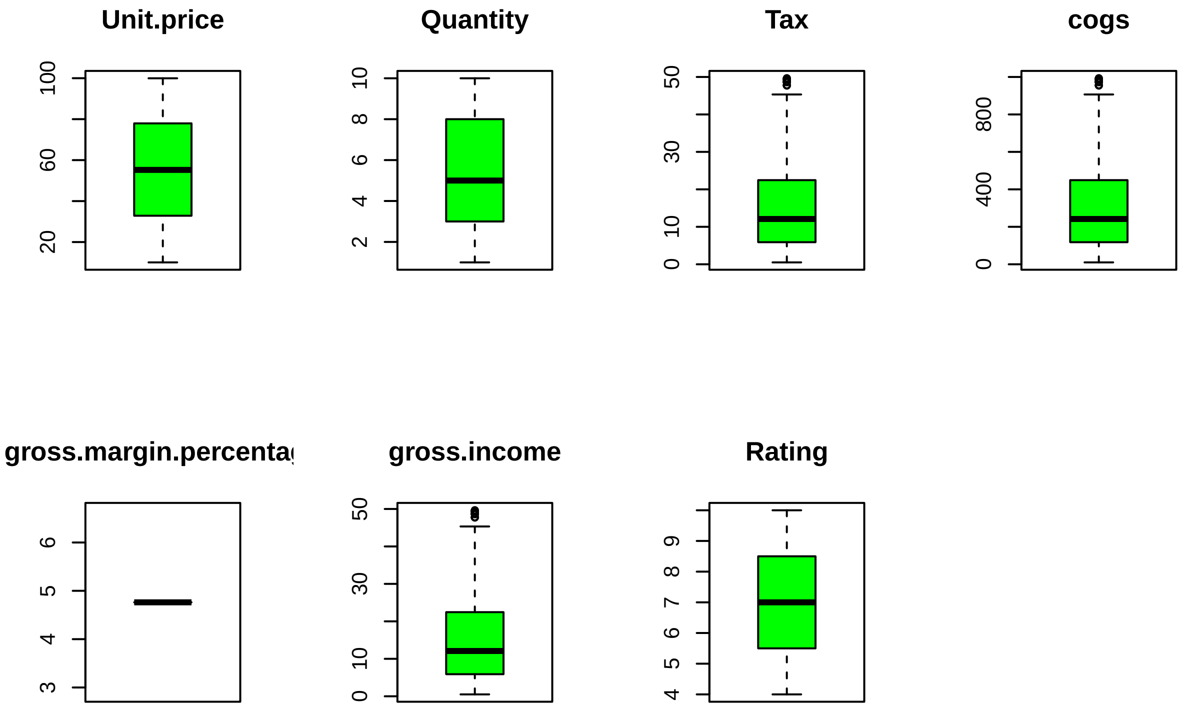
```
## 1     9.1 548.9715
## 2     9.6  80.2200
## 3     7.4 340.5255
## 4     8.4 489.0480
## 5     5.3 634.3785
## 6     4.1 627.6165
```

```
#Checking Outliers
# Boxplot from the  dataset with refernce to the numeric variables
boxplot(numeric, col = rainbow(ncol(numeric)))
```




We have outliers in the cogs and Total variable but we won't drop the outliers as they represent real data

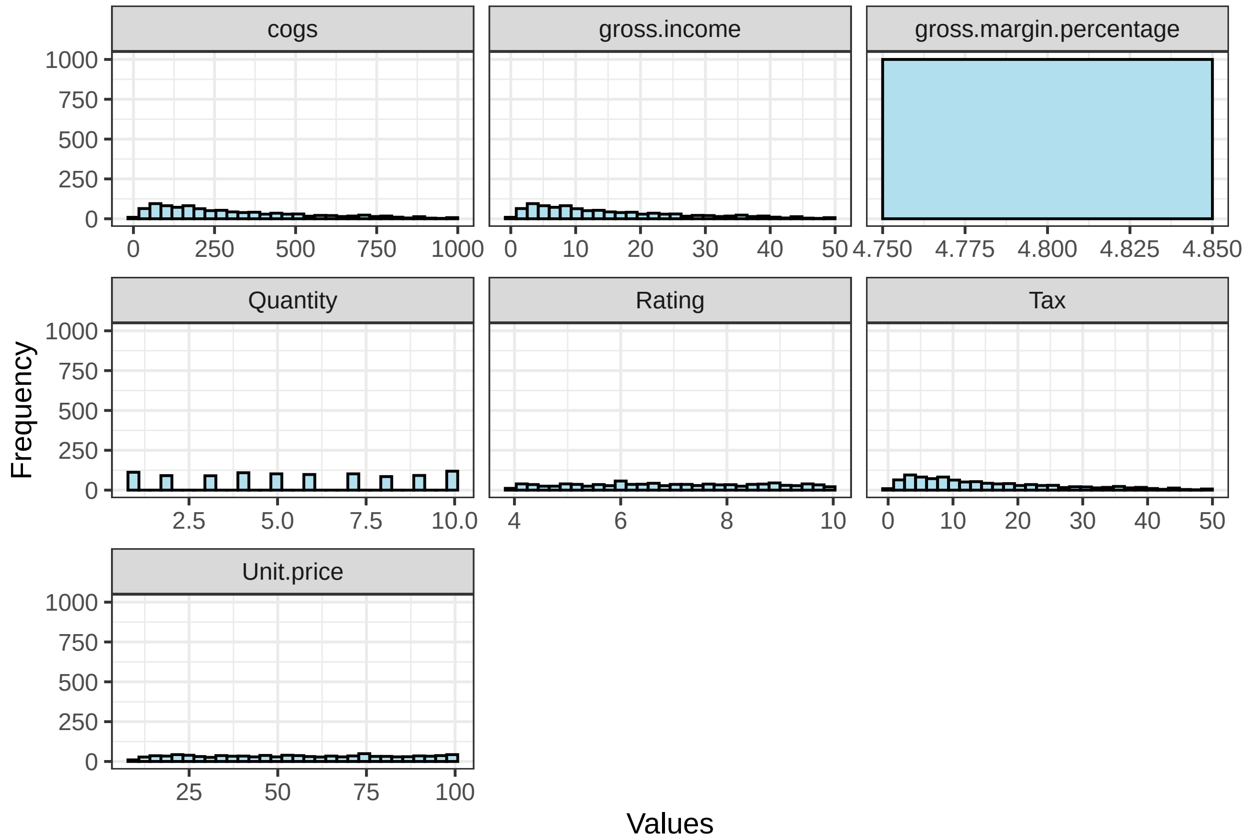
```
# Creating separate boxplots for each attribute
par(mfrow=c(2,4))
for(i in 1:7) {
    boxplot(numeric[,i], main=names(numeric)[i], col = "green")}
```




We have outliers in the tax variable also

```
#histogram representation of the numerical variables
numeric %>%
  gather(attributes, value, 1:7) %>%
  ggplot(aes(x = value)) +
  geom_histogram(fill = 'lightblue2', color = 'black') +
  facet_wrap(~attributes, scales = 'free_x') +
  labs(x="Values", y="Frequency") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




Most of the data is skewed

# IMPLEMENTING THE SOLUTION

```
#If there are integers, then you'll get variances of 0, causing the scaling to fail.
numeric$Quantity <- as.numeric(numeric$Quantity)
```

```
# If the standard deviation is zero, you can remove the variable and compute pca

df1 <- numeric %>% select(-gross.margin.percentage)
```

```
df1.pca <- prcomp(df1, center = TRUE, scale. = TRUE)
summary(df1.pca)
```

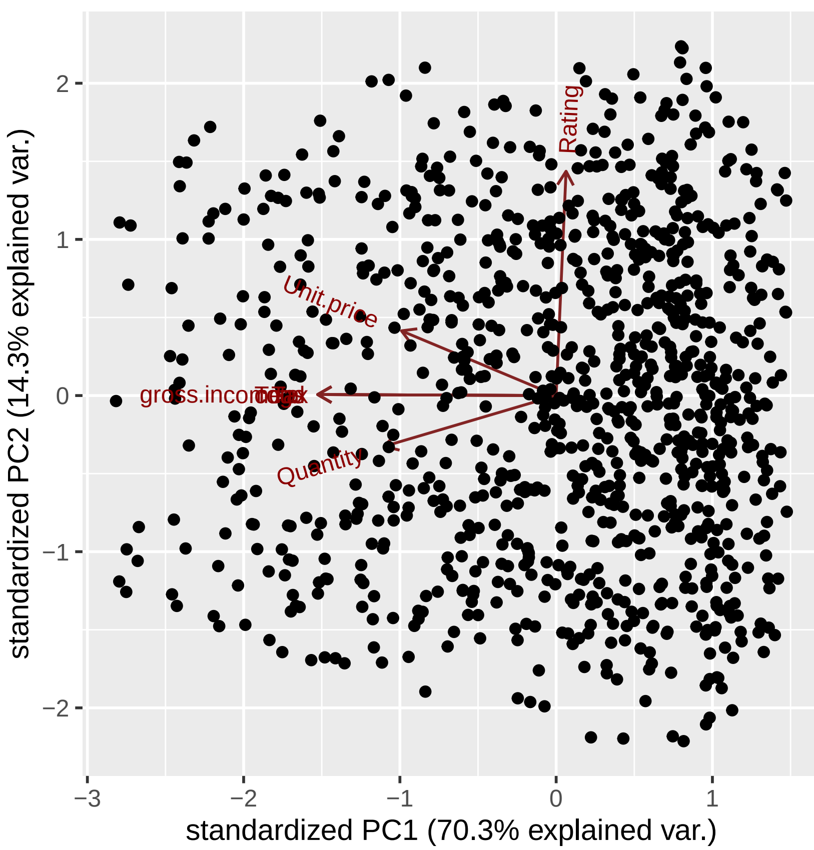
```
## Importance of components:
##                           PC1    PC2    PC3     PC4       PC5       PC6
## Standard deviation     2.2185 1.0002 0.9939 0.30001 2.981e-16 1.493e-16
## Proportion of Variance 0.7031 0.1429 0.1411 0.01286 0.000e+00 0.000e+00
## Cumulative Proportion  0.7031 0.8460 0.9871 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% of the total variance, which means that majority of the information in the dataset (16 variables) can be encapsulated by just that one Principal Component. PC2 and PC3 explains 14% of the variance.

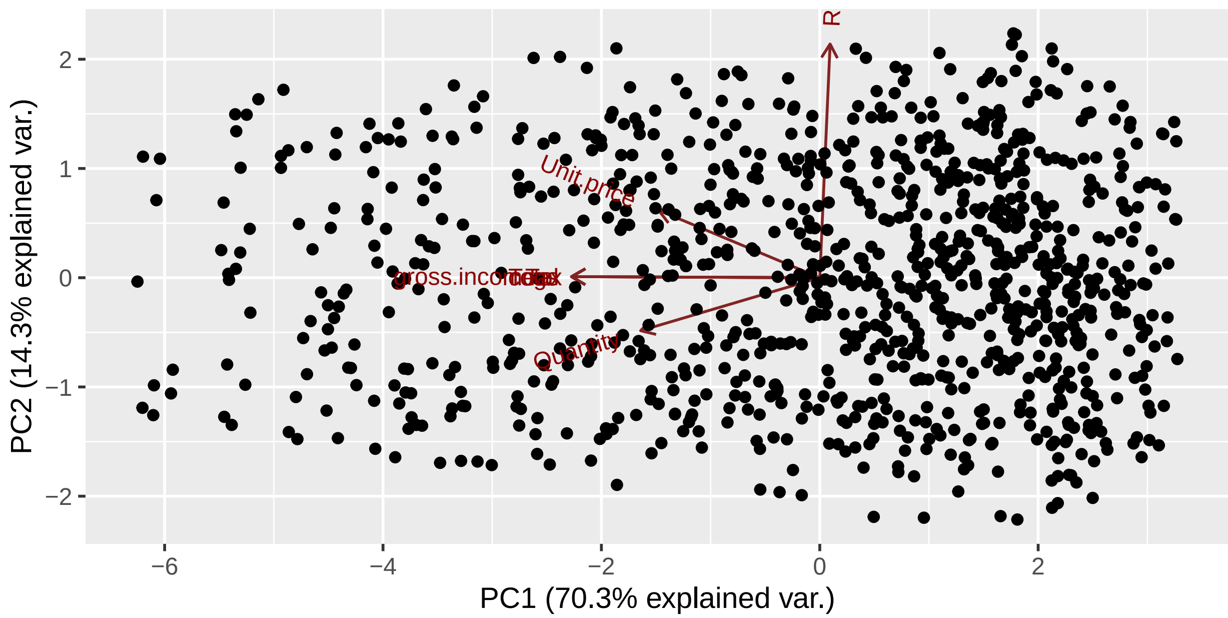
```
#First we can have a look at our df1.pca object with the aim of understanding its dimensions and compon
str(df1.pca)
```

```
## 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 the df.pca object
ggbiplot(df1.pca)
```



```r
# Adding more detail to the plot, we provide arguments rownames as labels
#
ggbiplot(df1.pca, labels=rownames(df1.pca), obs.scale = 1, var.scale = 1)
```




## CONCLUSION

PC1 accounts for 70.31% of the total variation in the dataset while PC2 accounts for 14.29% thus we can draw some valuable information and insights by just plotting the 2 principal components.