# R-Dimentionality_Reduction_UL

Vivian Bwana

2022-06-10

## IDENTIFYING RELEVANT MARKETING STRATEGIES USING DIMENSIONALITY REDUCTION TECHNIQUE

### Defining the Question

#### a) Specifying the question

To create a model that will identify the most relevant marketing strategies that will result in the highest no. of sales (total price including tax)

#### b) Metric for success

To be able to inform the marketing department on the most relevant marketing strategies that will result in the highest no. of sales.

#### c) Understanding the Context

Carrefour is one of the leading retail shops, (supermarkets) in the world. It was founded in France, in 1959. It has over the years expanded it's operations internationaly with the Kenyan branch opening in 1995.It has several branches in different parts of major cities countrywide.

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) Experimental Design

1. Problem Definition
2. Data Sourcing
3. Check the Data
4. Perform Data Cleaning
5. Perform Dimensionality Reduction
6. Conclusion
7. Recommendation

### e) Data Relevance /Sourcing

The dataset is relevant and reliable since it was provided by the client. We were able to draw relevant insights from it.

## Data Understanding

Loading Libraries

```
# loading the necessary libraries
library(data.table)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(purrr)
```

```
##
## Attaching package: ’purrr’

## The following object is masked from ’package:caret’:
##
##     lift

## The following object is masked from ’package:data.table’:
##
##     transpose
```

```
library(dbplyr)
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:dbplyr’:
##
##     ident, sql

## The following objects are masked from ’package:data.table’:
##
##     between, first, last

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

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

```
library(data.table)
```

# 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.

Dataset for Part 1: Dimensionality Reduction using PCA

```
# loading dataset
library(readr)
cafo12 <- fread("~/Downloads/Supermarket_Dataset_1 - Sales Data.csv")
#preview
head(cafo12)
```

```
##     Invoice ID Branch Customer type Gender           Product line Unit price
##         <char> <char>        <char> <char>                 <char>      <num>
## 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
##       <int>   <num>    <char> <char>      <char>  <num>                   <num>
## 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
##           <num>  <num>    <num>
## 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
```

### Exploring the Dataset

Dimensions

```
# checking the dimensions of the datasets
# to see how many rows and coulums there are
dim(cafo12)
```

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

There are 1000 and 16 columns. Since the features are not so many, we will use PCA.

Data Types

```
#Checking the datatypes of the dataset
str(cafo12)
```

```
## Classes 'data.table' and '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 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

The variables are mixed up, integers, characters, categorical, and are correctly assigned.

Descriptive Statistics Summary

```
# checking summary of the dataframe
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
#library(describe)
#describe(cafo12)
summary(cafo12)
```

```
##   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
```

The function summary gives the statistical summary of mean, median, minimum, maximum and quantile ranges as shown above

Column Names

```r
# checking the column names
#colnames(cafo12)
```

Missing Values

```r
#Checking for the sum of Missing values
colSums(is.na(cafo12))
```

```
##               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 missing values

Duplicates

```
# checking for duplicates
cafo12.duplicates <- cafo12[duplicated(cafo12),]

#printing duplicated rows
cafo12.duplicates
```

```
## Empty data.table (0 rows and 16 cols): Invoice ID,Branch,Customer type,Gender,Product line,Unit pric
```

There are no duplicates

# Performing PCA

## Selecting Numerical Features

We first extract numerical features to use on PCA

Preview column names

```
# checking the column names
# colnames(cafo12)
```

Extracting numerical cols

```
# extracting numerical columns
nump <-data.frame(cafo12[,c(6,7,8,12,14,15,16)])
# previewing
head(nump)
```

```
##   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
```

We now have a dataset with 7 columns only

Applying pca function

```r
# Apllying prcomp() fn and scaling the data
library(pcaPP)
nump.pca <- prcomp(scale(nump), center = TRUE, scale. = TRUE)
```

Checking summary to see the statistics of the PCAs

```r
# previewing with summary
summary(nump.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4       PC5       PC6
## Standard deviation     2.2185 1.0002 0.9939 0.30001 4.002e-16 1.446e-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     1.222e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

We have obtained 7 principal components. PC1 explains 70% of the total variance, meaning it can be used to capture all the information in the dataset. PC2 and PC3 explain 14% of the variance. The rest have very low percentages hence can just be ignored in the pca.

Checking the PCA object

```r
# using str() to look at the PCA object

str(nump.pca)
```

```
## List of 5
##  $ sdev    : num [1:7] 2.22 1.00 9.94e-01 3.00e-01 4.00e-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] -1.06e-16 7.53e-17 -4.33e-17 2.02e-17 -4.33e-17 ...
##   ..- attr(*, "names")= chr [1:7] "Unit.price" "Quantity" "Tax" "cogs" ...
##  $ scale   : Named num [1:7] 1 1 1 1 1 1 1
##   ..- 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"
```

This function gives us the \$standard deviation, \$center, \$scale, $rotationandthevaluests$(x) of each principal component.

Plotting our pca for more insights

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

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

```
install_github("vqv/ggbiplot")
```

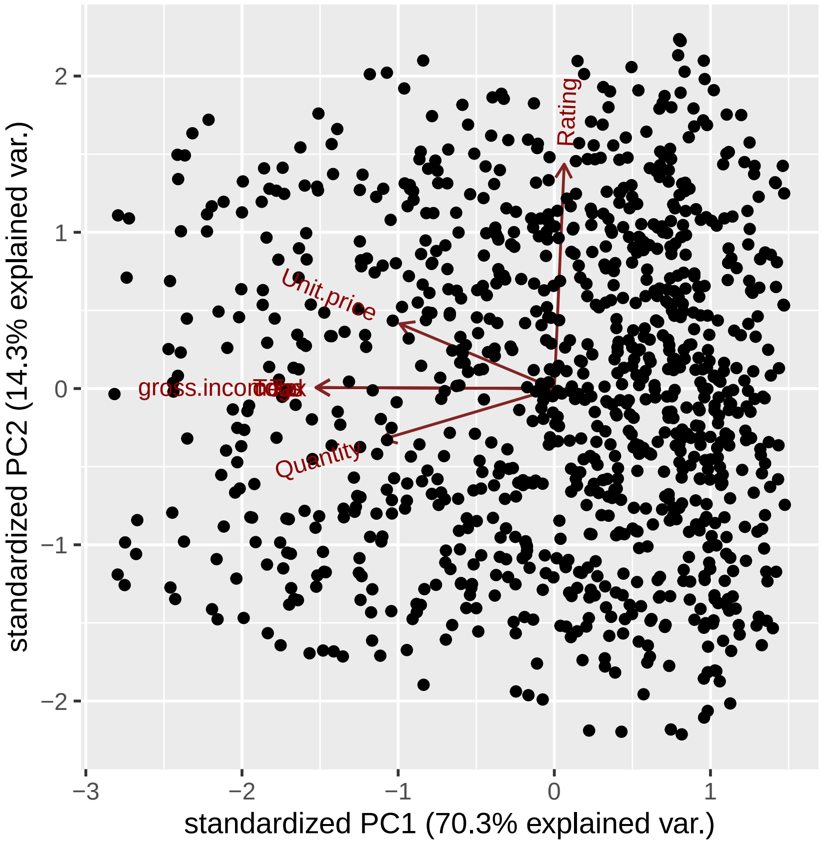
```
## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last i
##   Use `force = TRUE` to force installation
```

Displaying our Plot

```
# 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:Hmisc':
##
##     is.discrete, summarize

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

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

## Loading required package: scales

##
## Attaching package: 'scales'

## The following object is masked from 'package:readr':
##
##     col_factor

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

## Loading required package: grid
```

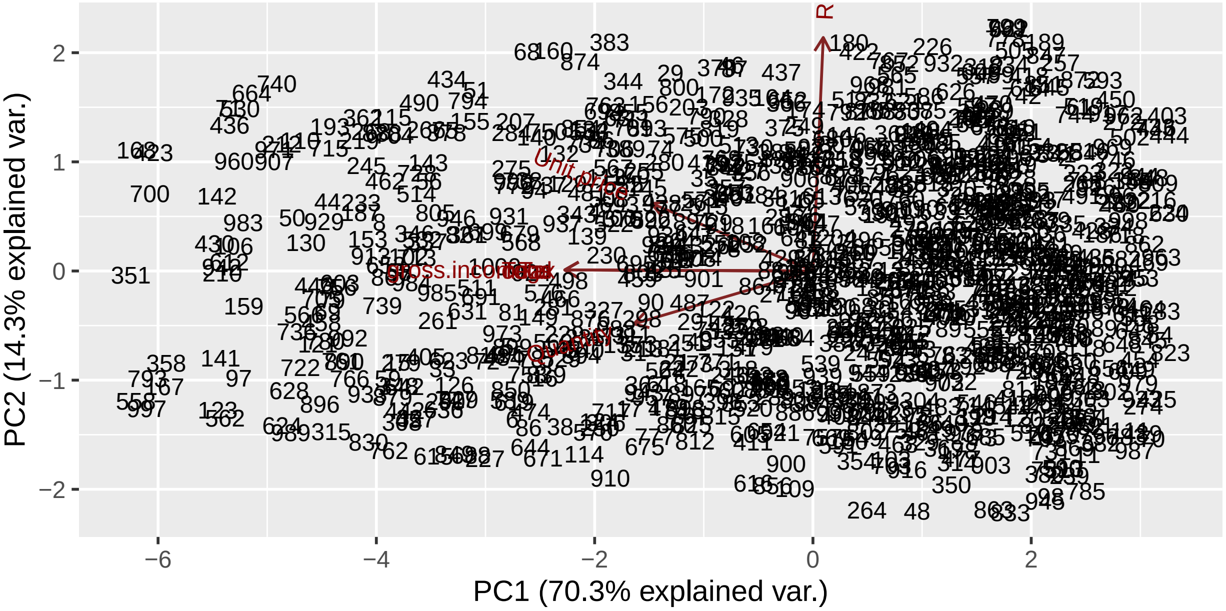
```
ggbiplot(nump.pca)
```



From the graph we will see that the variables quantity, unit price and gross income contribute to PC2.

Adding more detail to the plot

```
# providing details like labels
ggbiplot(nump.pca, labels=rownames(nump), obs.scale = 1, var.scale = 1)
```



## Conclusion

We have obtained 7 principal components. PC1 explains 70% of the total variance, meaning it can be used to capture all the information in the dataset. PC2 and PC3 explain 14% of the variance. The rest have very low percentages hence can just be ignored in the pca.

## Recommendation

From the graph we will see that the variables quantity, unit price and gross income contribute to PC2. These should be considered during modeling.