R Notebook

DIMENSIONALITY REDUCTION

1. DEFINING THE QUESTION

a) Specifying the Question

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

b) Defining the Metrics of Success

Reducing the dataset to a low dimensional dataset using the PCA. Performing the analysis and providing insights gained from the analysis.

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

1. Defining the question, the metric for success, the context and the experimental design.
2. Reading and exploring the dataset.
3. Reducing the dataset to a low dimensional dataset using the PCA.

e) Relevance of the data

The data used will inform the marketing department on the most relevant marketing strategies that will result in the highest number of sales and total price including tax. The dataset link: http://bit.ly/CarreFourDataset

2. DATA ANALYSIS

a) Checking the Data

```r
# Loading libraries
library(relaimpo)
```

```
## Loading required package: MASS

## Loading required package: boot

## Loading required package: survey

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##
## Attaching package: ’survival’

## The following object is masked from ’package:boot’:
##
##     aml

##
## Attaching package: ’survey’

## The following object is masked from ’package:graphics’:
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```r
library(data.table)
library(ggplot2) # Data visualization
library(ggthemes) # Plot themes
library(plotly) # Interactive data visualizations
```

```
##
## Attaching package: ’plotly’

## The following object is masked from ’package:ggplot2’:
##
##     last_plot
## The following object is masked from ’package:MASS’:
##
##     select

## The following object is masked from ’package:stats’:
##
##     filter

## The following object is masked from ’package:graphics’:
##
##     layout
```

```
library(dplyr) # Data manipulation
```

```
##
## Attaching package: ’dplyr’

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

## The following object is masked from ’package:MASS’:
##
##     select

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

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

```
library(psych) # Will be used for correlation visualization
```

```
##
## Attaching package: ’psych’

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

## The following object is masked from ’package:boot’:
##
##     logit
```

```
# Importing the data
df <- fread('http://bit.ly/CarreFourDataset')
df
```

```
##        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
##   ---
##  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
##          <int>   <num>    <char> <char>      <char>  <num>
##    1:        7 26.1415  1/5/2019  13:08     Ewallet 522.83
##    2:        5  3.8200  3/8/2019  10:29        Cash  76.40
##    3:        7 16.2155  3/3/2019  13:23 Credit card 324.31
##    4:        8 23.2880 1/27/2019  20:33     Ewallet 465.76
##    5:        7 30.2085  2/8/2019  10:37     Ewallet 604.17
##   ---
##  996:        1  2.0175 1/29/2019  13:46     Ewallet  40.35
##  997:       10 48.6900  3/2/2019  17:16     Ewallet 973.80
##  998:        1  1.5920  2/9/2019  13:22        Cash  31.84
##  999:        1  3.2910 2/22/2019  15:33        Cash  65.82
## 1000:        7 30.9190 2/18/2019  13:28        Cash 618.38
##       gross margin percentage gross income Rating     Total
##                         <num>        <num>  <num>     <num>
##    1:                4.761905      26.1415    9.1  548.9715
##    2:                4.761905       3.8200    9.6   80.2200
##    3:                4.761905      16.2155    7.4  340.5255
##    4:                4.761905      23.2880    8.4  489.0480
##    5:                4.761905      30.2085    5.3  634.3785
##   ---
##  996:                4.761905       2.0175    6.2   42.3675
##  997:                4.761905      48.6900    4.4 1022.4900
##  998:                4.761905       1.5920    7.7   33.4320
##  999:                4.761905       3.2910    4.1   69.1110
## 1000:                4.761905      30.9190    6.6  649.2990
```

b) Data Checking

```
# Previewing the dataset
View(df)
```

```
# Previewing the column names
colnames(df)
```

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

```
# Previewing the datatypes of the dataset
sapply(df, class)
```

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

```
# Previewing the head of the dataset
head(df, n = 5)
```

```
##     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
##    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
##    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
```

```
# Previewing the bottom of the dataset
head(df, n = 5)
```

```
##     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
##    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
##    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
```

```
# Checking the structure of the data
str(df)
```

```
## 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>
```

```
# Checking the shape of the data
dim(df)
```

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

1000 rows and 16 columns

c) Data Cleaning

Missing Values

```
# Checking for missing values
sum(is.na(df))
```

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

There are no missing values in the data

Duplicates

```
# Checking for duplicates
duplicated_rows <- df[duplicated(df),]
duplicated_rows
```

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

There are no duplicates in the data

```
# Displaying unique items and assigning them to a variable unique_items below
unique_items <- df[!duplicated(df), ]
unique_items
```

```
##        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
##   ---
##  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
##          <int>   <num>    <char> <char>      <char>  <num>
##    1:        7 26.1415  1/5/2019  13:08     Ewallet 522.83
##    2:        5  3.8200  3/8/2019  10:29        Cash  76.40
##    3:        7 16.2155  3/3/2019  13:23 Credit card 324.31
##    4:        8 23.2880 1/27/2019  20:33     Ewallet 465.76
##    5:        7 30.2085  2/8/2019  10:37     Ewallet 604.17
##   ---
##  996:        1  2.0175 1/29/2019  13:46     Ewallet  40.35
##  997:       10 48.6900  3/2/2019  17:16     Ewallet 973.80
##  998:        1  1.5920  2/9/2019  13:22        Cash  31.84
##  999:        1  3.2910 2/22/2019  15:33        Cash  65.82
## 1000:        7 30.9190 2/18/2019  13:28        Cash 618.38
##       gross margin percentage gross income Rating     Total
##                         <num>        <num>  <num>     <num>
##    1:                4.761905      26.1415    9.1  548.9715
##    2:                4.761905       3.8200    9.6   80.2200
##    3:                4.761905      16.2155    7.4  340.5255
##    4:                4.761905      23.2880    8.4  489.0480
##    5:                4.761905      30.2085    5.3  634.3785
##   ---
##  996:                4.761905       2.0175    6.2   42.3675
##  997:                4.761905      48.6900    4.4 1022.4900
##  998:                4.761905       1.5920    7.7   33.4320
##  999:                4.761905       3.2910    4.1   69.1110
## 1000:                4.761905      30.9190    6.6  649.2990
```

```
# Displaying the numerical data columns
df1 <- df %>% select_if(is.numeric)
colnames(df1)
```

```
## [1] "Unit price"              "Quantity"
## [3] "Tax"                     "cogs"
## [5] "gross margin percentage" "gross income"
## [7] "Rating"                  "Total"
```

```
# Renaming columns for an easy analysis
df1 <- df1 %>% rename(Unit_price = "Unit price")
df1 <- df1 %>% rename(gross_income = "gross income")

# Selecting needed columns
df2 <- subset(df1, select = c("Unit_price", "Quantity", "Tax", "cogs", "gross_income", "Rating", "Total
colnames(df2)
```

```
## [1] "Unit_price"   "Quantity"     "Tax"          "cogs"         "gross_income"
## [6] "Rating"       "Total"
```

```
describe(df2)
```

```
##              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
```

3. DIMENSIONALITY REDUCTION WITH PCA

```
str(df2)
```

```
## Classes ’data.table’ and ’data.frame’:   1000 obs. of  7 variables:
##  $ 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 ...
##  $ cogs        : num  522.8 76.4 324.3 465.8 604.2 ...
##  $ 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>
```

```
# We then pass df to the prcomp(). We also set two arguments, center and scale,
# to be TRUE then preview our object with summary
df3 <- prcomp(df2)
summary(df3)
```

```
## Importance of components:
##                              PC1      PC2     PC3     PC4       PC5       PC6
## Standard deviation     340.3819 20.53212 1.71932 1.24589 4.021e-14 2.522e-15
## Proportion of Variance   0.9963  0.00363 0.00003 0.00001 0.000e+00 0.000e+00
## Cumulative Proportion    0.9963  0.99996 0.99999 1.00000 1.000e+00 1.000e+00
##                                PC7
## Standard deviation     5.734e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

As a result we obtain 9 principal components, each which explain a percentate of the total variation of the dataset

```
# Calling str() to have a look at your PCA object
str(df3)
```

```
## List of 5
##  $ sdev    : num [1:7] 3.40e+02 2.05e+01 1.72 1.25 4.02e-14 ...
##  $ rotation: num [1:7, 1:7] -0.04952 -0.00605 -0.0344 -0.68798 -0.0344 ...
##   ..- 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   : logi FALSE
##  $ x       : num [1:1000, 1:7] -313 337.2 -23.8 -229.5 -431.5 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

The center point (*center*), *scaling*(scale), standard deviation(sdev) of each principal component. The relationship (correlation or anticorrelation, etc) between the initial variables and the principal components (*rotation*).*Thevaluesofeachsampleintermsoftheprincipalcomponents*(x)

```
# Installing our visualisation package

library(devtools)
```

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

```
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
```

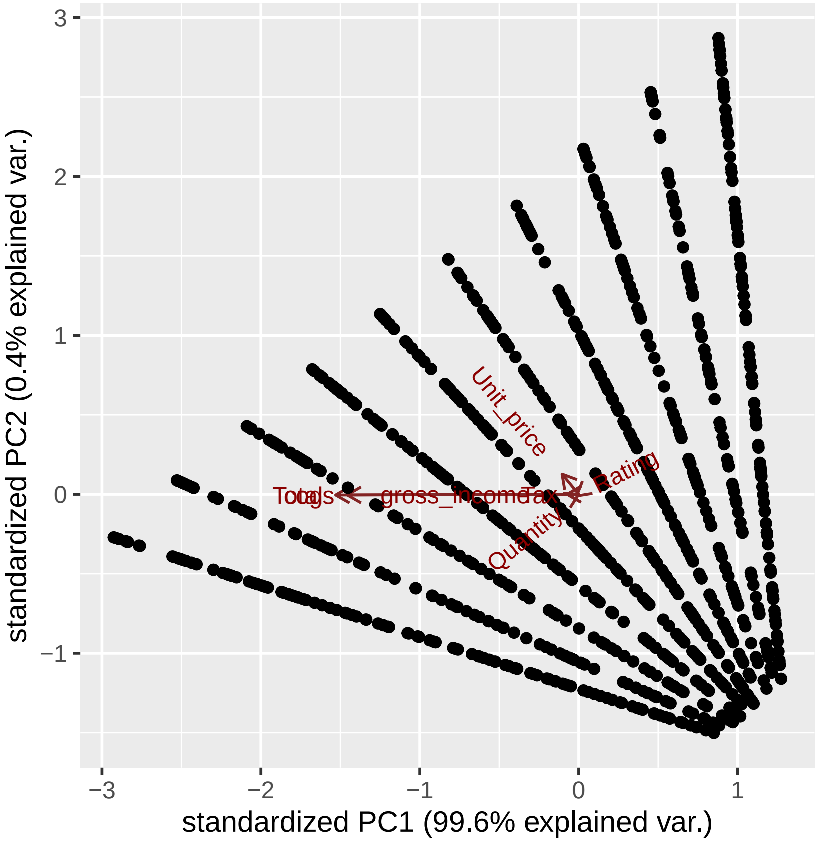
```
## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise
```

```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'
```

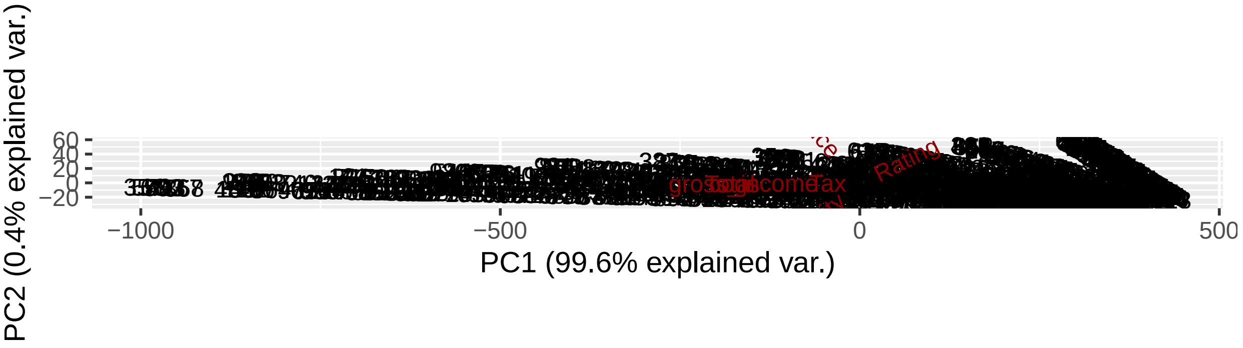
```
## The following objects are masked from 'package:psych':
##
##     alpha, rescale
```

```
ggbiplot(df3)
```



```
# Adding more detail to the plot, we provide arguments rownames as labels

ggbiplot(df3, labels=rownames(df), obs.scale = 1, var.scale = 1)
```



We find it difficult to derive insights from the given plot this is because explain very small percentages of the total variation, thus it would be surprising if we found that they were very informative and separated the groups or revealed apparent patterns.