Dimensionality Reduction & Feature Selection

Jenipher Mawia

11/13/2020

1. Problem Definition

1.1 Defining the question

- Perform dimensionality reduction on the data provided
- Also perform feature selection on the same data

1.2 Specifying the question

Implement the solutions using unsupervised learning techniques for :

- dimensionality reduction: reduce your dataset to a low dimensional dataset using the t-SNE algorithm or PCA

- and feature selection

2. Defining the metrics for success

This project will be considered a success if the following are achieved: - Unsupervised learning techniques are used for dimensionality reduction and feature selection without any errors.

3. 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 two parts where you'll explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights.

4. Experimental Design taken

The project consists of two parts. The following is the order in which I went about to achieve the objectives of this project:

- Data Sourcing and Understanding

- Checking the data (head and tail, shape(number of records), datatypes)
- Data cleaning procedures (handling null values,outliers, anomalies)
- Exploratory data analysis (Univariate, Bivariate and Multivariate analyses)
- Implementing the solution
    - dimension reduction
    - Feature selection
- Conclusion and recommendation

5. Data Sourcing

The data used for this project was provided by Moringa School and can be downloaded [here]

Reading the data

```
superdata <- read.csv("http://bit.ly/CarreFourDataset")
```

6. Checking the Data

checking the top of the data

```
# checking the first 6 rows in the data
head(superdata)
```

```
##    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
```

checking the bottom of the data

```
# checking the last 6 rows in the data
tail(superdata)
```

```
##       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
```

checking the shape of the data

```
# checking the dimensions of the data (number of entries and fields)
dim(superdata)
```

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

The data has 1000 entries and 16 columns.

checking the datatypes of the column

```
# getting the datatypes of each column
str(superdata)
```

```
## '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" "Health and beauty" ...
##  $ 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 ...
```

The data consists of columns that contain numeric, integer and character datatypes.

Checking the number of unique values in each column

```
lengths(lapply(superdata, unique))
```

```
##              Invoice.ID                  Branch           Customer.type
##                    1000                       3                       2
##                  Gender            Product.line              Unit.price
##                       2                       6                     943
##                Quantity                     Tax                    Date
##                      10                     990                      89
##                    Time                 Payment                    cogs
##                     506                       3                     990
```

```
## gross.margin.percentage             gross.income                   Rating
##                       1                      990                      61
##                   Total
##                     990
```

There exists categorical datatypes in the data as shown by the number of unique values per column.

7. Appropriateness of the available data to answer the given question

The data contains columns such as:

- Invoice id which has the invoice number of a given transaction made by a customer. It should be unique for each customer.
- Branch: This suggests that there are more than one branches of the same store from which customers in different regions can shop from.
- Customer type: the type of customer either a member or normal customer.
- Gender: sex of the customer
- Product line: defining the category from which a customer purchased a product from
- Unit price: definng the price of the product per unit
- Quantity: number of products bought
- Tax: amount of tax charged
- Date of transaction
- Time of the day the purchase was made
- Payment: type of payment method used: by cash, credit card, ewallet etc.
- Cogs: Cost of goods sold
- Gross Margin Percentage: percentage change in the gross margin when the purchase is made
- Gross Income: gross income generated from the purchase
- Rating: rating of the transaction in form of numeric values, low values could indicate a poor rating while high values suggest good rating
- Total: total amount generated from the purchase

All these fields are useful in informing the marketing department on the most relevant marketing strategies that will result in the highest number of sales

Therefore, it can be concluded that the data available is appropriate and relevant to answer the given question.

8. Data Cleaning

8.1 Standardizing column names

Column names should be in the same format to ensure consistency

```
# change column names to all lowercase
colnames(superdata) = tolower(colnames(superdata))

# check changes made
colnames(superdata)
```

```
##  [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"
```

```
# change two and three letter columns to a standard format
names(superdata)[names(superdata) == "invoice.id"] <- "invoice_id"
names(superdata)[names(superdata) == "customer.type"] <- "customer_type"
names(superdata)[names(superdata) == "product.line"] <- "product_line"
names(superdata)[names(superdata) == "unit.price"] <- "unit_price"
names(superdata)[names(superdata) == "gross.margin.percentage"] <-
"gross_margin_percentage"
names(superdata)[names(superdata) == "gross.income"] <- "gross_income"

#check changes made
colnames(superdata)
```

```
##  [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"
```

8.2 Datatype conversion

Some columns such as rating, customer type, branch, gender, payment and product line are categorical fields but are in numeric or character data types. These need to be converted to factors.

```
# convert the datatypes to factors
superdata$rating <-as.factor(superdata$rating)
superdata$customer_type <-as.factor(superdata$customer_type)
superdata$branch <-as.factor(superdata$branch)
superdata$product_line <-as.factor(superdata$product_line)
superdata$gender <-as.factor(superdata$gender)
superdata$payment <-as.factor(superdata$payment)

# check the datatypes once more to see changes made
str(superdata)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ invoice_id             : chr  "750-67-8428" "226-31-3081" "631-41-3108"
"123-19-1176" ...
##  $ branch                 : Factor w/ 3 levels "A","B","C": 1 3 1 1 1 3 1
3 1 2 ...
##  $ customer_type          : Factor w/ 2 levels "Member","Normal": 1 2 2 1
2 2 1 2 1 1 ...
##  $ gender                 : Factor w/ 2 levels "Female","Male": 1 1 2 2 2
2 1 1 1 1 ...
##  $ product_line           : Factor w/ 6 levels "Electronic
accessories",..: 4 1 5 4 6 1 1 5 4 3 ...
##  $ 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                : Factor w/ 3 levels "Cash","Credit card",..: 3
1 2 3 3 3 3 3 2 2 ...
##  $ 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                 : Factor w/ 61 levels "4","4.1","4.2",..: 52 57
35 45 14 2 19 41 33 20 ...
##  $ total                  : num  549 80.2 340.5 489 634.4 ...
```

8.3 Duplicated Entries

```
#Checking for duplicated rows
duplicates <- superdata[duplicated(superdata),]
dim(duplicates)
```

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

There are no duplicated entries in the data.

8.4 Missing Values

```
# check for missing values in each column in the data
colSums(is.na(superdata))
```

```
##              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 in the data

8.5 Outliers

```
# get numerical columns from the data
nums <- unlist(lapply(superdata, is.numeric))

# output the numeric columns in form of a dataframe and check the top of the
resulting dataframe
numeric_cols <- superdata[ , nums]
head(numeric_cols)
```

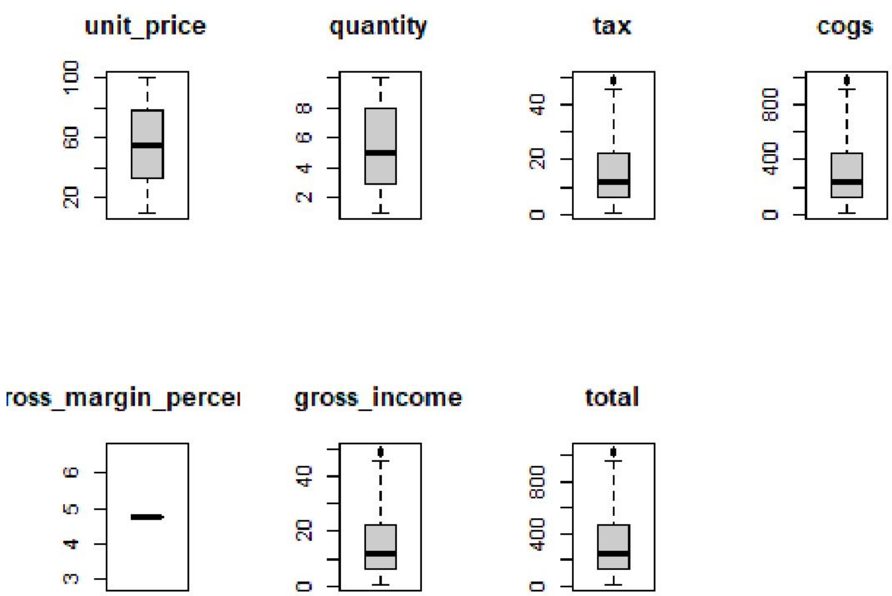
```
##   unit_price quantity     tax   cogs gross_margin_percentage gross_income
## 1      74.69        7 26.1415 522.83                4.761905      26.1415
## 2      15.28        5  3.8200  76.40                4.761905       3.8200
## 3      46.33        7 16.2155 324.31                4.761905      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
##      total
## 1 548.9715
## 2  80.2200
## 3 340.5255
## 4 489.0480
## 5 634.3785
## 6 627.6165
```

There are 7 numeric columns from the total 16.

```
#boxplot(numeric_cols)

# make a plot of multiple boxplots to check for outliers
```

```r
par ( mfrow= c (  2, 4 ))
for (i in 1 : length (numeric_cols)) {
boxplot (numeric_cols[,i], main= names (numeric_cols[i]), type= "l" )
}
```



There are a few outliers present in the data. We will not remove them because of the dynamics that usually occur in purchases where some customers can be extremely extravagant and others extremely conservative hence causing outliers in the data. Removing them will make the resulting data not be a picture of the actual data.

9. Exploratory Data Analysis

9.1 Univariate Data Analysis

Measures of central tendency

Mean

```r
# get the mean of all numerical columns
library(ggplot2)
library(psych)
```

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

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

```r
colMeans(numeric_cols)
```

```
##               unit_price                 quantity                      tax
##                55.672130                 5.510000                15.379369
##                     cogs  gross_margin_percentage             gross_income
##               307.587380                 4.761905                15.379369
##                    total
##               322.966749
```

Median

```r
# Get the median of all numerical columns
apply (numeric_cols, 2 ,median)
```

```
##               unit_price                 quantity                      tax
##                55.230000                 5.000000                12.088000
##                     cogs  gross_margin_percentage             gross_income
##               241.760000                 4.761905                12.088000
##                    total
##               253.848000
```

Mode

```r
# Create the function.
getmode <- function(v) {
uniqv <- unique(v)
uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```r
lapply(superdata,FUN=getmode)
```

```
## $invoice_id
## [1] "750-67-8428"
##
## $branch
## [1] A
## Levels: A B C
##
## $customer_type
## [1] Member
## Levels: Member Normal
##
## $gender
## [1] Female
## Levels: Female Male
##
## $product_line
## [1] Fashion accessories
## 6 Levels: Electronic accessories Fashion accessories ... Sports and travel
```

```
##
## $unit_price
## [1] 83.77
##
## $quantity
## [1] 10
##
## $tax
## [1] 39.48
##
## $date
## [1] "2/7/2019"
##
## $time
## [1] "19:48"
##
## $payment
## [1] Ewallet
## Levels: Cash Credit card Ewallet
##
## $cogs
## [1] 789.6
##
## $gross_margin_percentage
## [1] 4.761905
##
## $gross_income
## [1] 39.48
##
## $rating
## [1] 6
## 61 Levels: 4 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.1 5.2 5.3 5.4 5.5 ...
10
##
## $total
## [1] 829.08
```

- Branch A is the most popular branch of the three branches.
- Most customers visiting the store are members and most of them are female.
- The most popular product line is the Fashion accessories.
- Most customers made payment through Ewallet
- Most customers gave a rating or 6
- The most popular time is 1948 hours

Measures of dispersion

Find the minimum, maximum and quantiles of the columns in the data.

```
summary(numeric_cols)
```

```
##    unit_price       quantity          tax                cogs       
##  Min.   :10.08   Min.   : 1.00   Min.   : 0.5085   Min.   : 10.17  
##  1st Qu.:32.88   1st Qu.: 3.00   1st Qu.: 5.9249   1st Qu.:118.50  
##  Median :55.23   Median : 5.00   Median :12.0880   Median :241.76  
##  Mean   :55.67   Mean   : 5.51   Mean   :15.3794   Mean   :307.59  
##  3rd Qu.:77.94   3rd Qu.: 8.00   3rd Qu.:22.4453   3rd Qu.:448.90  
##  Max.   :99.96   Max.   :10.00   Max.   :49.6500   Max.   :993.00  
##  gross_margin_percentage  gross_income         total        
##  Min.   :4.762           Min.   : 0.5085   Min.   :  10.68  
##  1st Qu.:4.762           1st Qu.: 5.9249   1st Qu.: 124.42  
##  Median :4.762           Median :12.0880   Median : 253.85  
##  Mean   :4.762           Mean   :15.3794   Mean   : 322.97  
##  3rd Qu.:4.762           3rd Qu.:22.4453   3rd Qu.: 471.35  
##  Max.   :4.762           Max.   :49.6500   Max.   :1042.65
```

Range

Range is the difference between the maximum point and the minimum point in a set of data.

```
# get the range for all numeric columns
lapply(numeric_cols,FUN=range)
```

```
## $unit_price
## [1] 10.08 99.96
## 
## $quantity
## [1]  1 10
## 
## $tax
## [1]  0.5085 49.6500
## 
## $cogs
## [1]  10.17 993.00
## 
## $gross_margin_percentage
## [1] 4.761905 4.761905
## 
## $gross_income
## [1]  0.5085 49.6500
## 
## $total
## [1]   10.6785 1042.6500
```

Interquartile Range

The interquartile range also commonly known as IQR is the range between the 1st and 3rd quantiles. It is the difference between the two quantiles.

```
# get the IQR for the numeric columns
lapply(numeric_cols,FUN=IQR)
```

```
## $unit_price
## [1] 45.06
##
## $quantity
## [1] 5
##
## $tax
## [1] 16.52037
##
## $cogs
## [1] 330.4075
##
## $gross_margin_percentage
## [1] 0
##
## $gross_income
## [1] 16.52037
##
## $total
## [1] 346.9279
```

Standard Deviation

Find the standard deviation of the numerical columns in the data

```
apply (numeric_cols, 2 ,sd)
```

```
##              unit_price                quantity                     tax
##               26.494628                2.923431               11.708825
##                    cogs gross_margin_percentage            gross_income
##              234.176510                0.000000               11.708825
##                   total
##              245.885335
```

Variance

Find the variance of the numerical columns

```
sapply (numeric_cols, var)
```

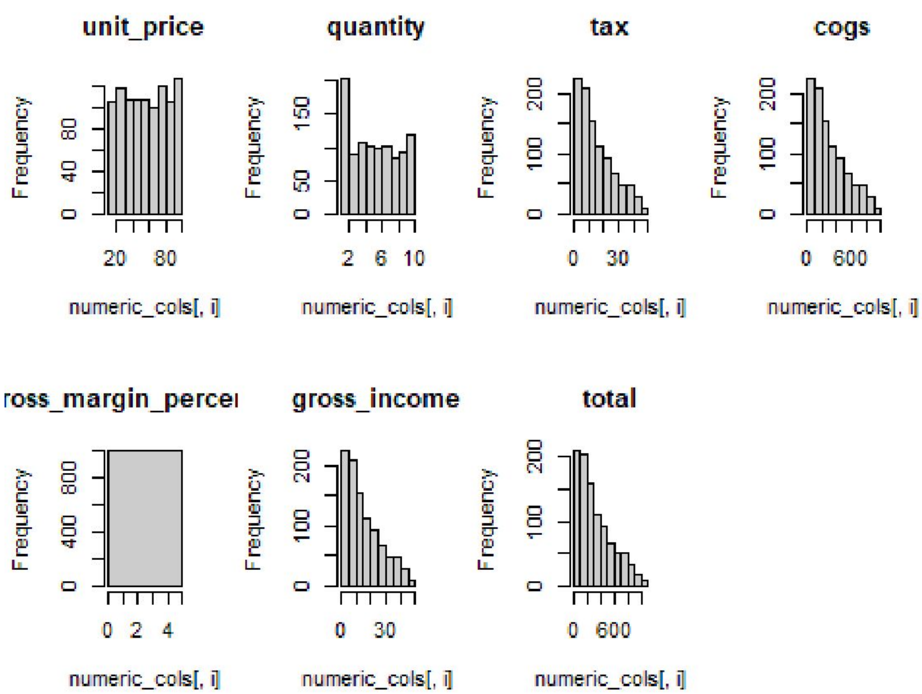
```
##              unit_price                quantity                     tax
##              701.965331                8.546446              137.096594
##                    cogs gross_margin_percentage            gross_income
##            54838.637658                0.000000              137.096594
```

```
##                        total
##              60459.598018
```

The variables appear to be measured in different units hence contributing to the fact that some variables have larger variances than others.

Histograms

```
par( mfrow= c ( 2 , 4 ))
for(i in 1 : length(numeric_cols)) {
hist(numeric_cols[,i], main= names(numeric_cols[i]))
}
```



The columns: tax, cogs, gross income, total are skewed to the right.

```
str(superdata)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ invoice_id              : chr  "750-67-8428" "226-31-3081" "631-41-3108" 
"123-19-1176" ...
##  $ branch                  : Factor w/ 3 levels "A","B","C": 1 3 1 1 1 3 1 
3 1 2 ...
##  $ customer_type           : Factor w/ 2 levels "Member","Normal": 1 2 2 1 
2 2 1 2 1 1 ...
##  $ gender                  : Factor w/ 2 levels "Female","Male": 1 1 2 2 2 
2 1 1 1 1 ...
##  $ product_line            : Factor w/ 6 levels "Electronic 
accessories",..: 4 1 5 4 6 1 1 5 4 3 ...
##  $ 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                  : Factor w/ 3 levels "Cash","Credit card",..: 3
1 2 3 3 3 3 3 2 2 ...
##  $ 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                   : Factor w/ 61 levels "4","4.1","4.2",..: 52 57
35 45 14 2 19 41 33 20 ...
##  $ total                    : num  549 80.2 340.5 489 634.4 ...
```

9.2 Bivariate and Multivariate analysis

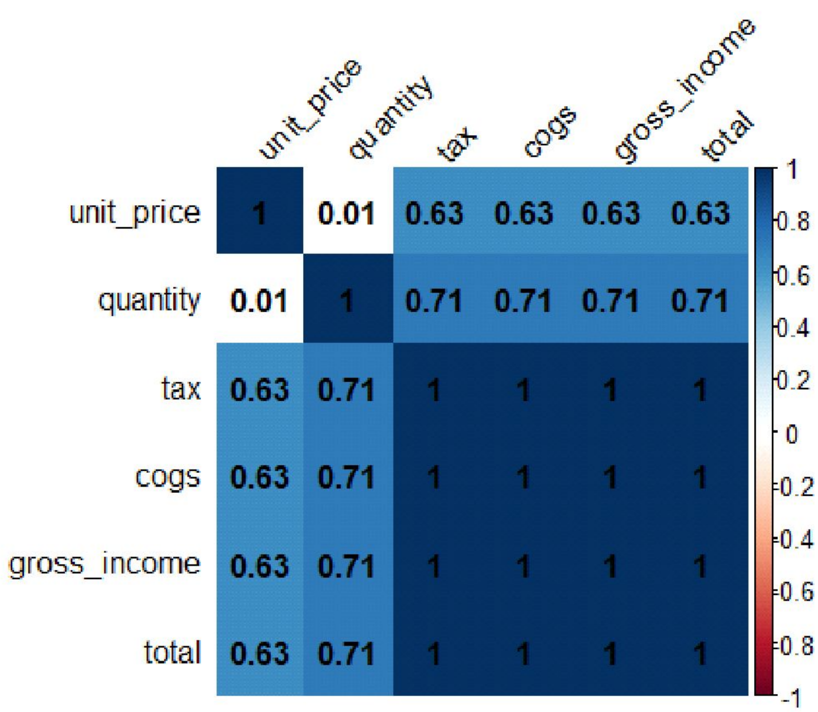
Correlation between the different variables

```
# Checking the correlation coefficients for numeric variables
library(ggcorrplot)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlations <- round(cor(numeric_cols[-5]), 2 )
```

```
corrplot(correlations, method = "color", type = "full", tl.col = "black",
tl.srt = 45, addCoef.col = "black")
```

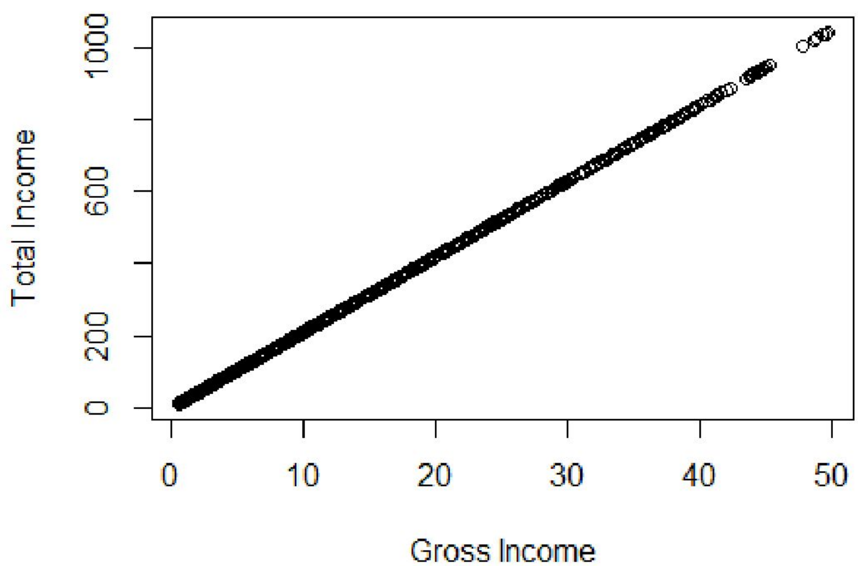


```
#ggcorrplot(corr, ggtheme = ggplot2::theme_gray, colors = c("#6D9EC1",
"white", "#E46726"), lab = T)
```

There is a high correlation between the numeric variables except the quantity column. This is quite expected as they relate to a specific purchase made by a customer i.e. the cost of goods sold will be dependent on tax, gross income is a function of cost of goods sold and net tax, total income is a function of net pay given by gross income minus tax charged. These relationships and co-dependence result in high correlations between the variables

Scatter Plot

```
plot(superdata$gross_income, superdata$total, xlab="Gross Income",
ylab="Total Income")
```



As expected, the scatter plot distribution between these two values follows a straight line. This shows a linear relationship as shown above from the correlation plot.

Since all numerical columns are highly related, it is expected that they will take the same shape of distribution when plotted.

10. Implementing the Solution

10.1 Data Pre-processing

Before we begin modelling, we must ensure that the datatypes in the data we will use are in the appropriate mode i.e. numeric.

```r
# save data to a new dataframe to avoid messing up with original data
data <- superdata

# change datatypes
data$branch <- as.numeric(data$branch)
data$customer_type <- as.numeric(data$customer_type)
data$gender <- as.numeric(data$gender)
data$product_line <- as.numeric(data$product_line)
data$payment <- as.numeric(data$payment)
data$rating <- as.numeric(data$rating)

#check the datatypes
str(data)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ invoice_id             : chr  "750-67-8428" "226-31-3081" "631-41-3108"
"123-19-1176" ...
##  $ branch                 : num  1 3 1 1 1 3 1 3 1 2 ...
##  $ customer_type          : num  1 2 2 1 2 2 1 2 1 1 ...
##  $ gender                 : num  1 1 2 2 2 2 1 1 1 1 ...
##  $ product_line           : num  4 1 5 4 6 1 1 5 4 3 ...
##  $ 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                : num  3 1 2 3 3 3 3 3 2 2 ...
##  $ 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  52 57 35 45 14 2 19 41 33 20 ...
##  $ total                  : num  549 80.2 340.5 489 634.4 ...
```

Since we will be implementing unsupervised learning algorithms, it is also important to remove the target variable "Total". We will also exclude columns that are not numerical.

```r
# remove the target and character variables
df <- data[c(-16, -1,-9, -10, -13)]
str(df)
```

```
## 'data.frame':    1000 obs. of  11 variables:
##  $ branch       : num  1 3 1 1 1 3 1 3 1 2 ...
##  $ customer_type: num  1 2 2 1 2 2 1 2 1 1 ...
##  $ gender       : num  1 1 2 2 2 2 1 1 1 1 ...
##  $ product_line : num  4 1 5 4 6 1 1 5 4 3 ...
##  $ 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 ...
##  $ payment      : num  3 1 2 3 3 3 3 3 2 2 ...
##  $ 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  52 57 35 45 14 2 19 41 33 20 ...
```

10.2 Dimensionality Reduction

Dimensionality reduction is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data.

There are various ways of dimensionality reduction. We will apply PCA for this project.

PCA uses linear combinations of the variables, known as principal components. The new projected variables (principal components) are uncorrelated with each other and are ordered so that the first few components retain most of the variation present in the original variables. Thus, PCA is useful in situations where the independent variables are correlated with each other. We have observed this aspect of multicolinearity from the correlation matrix above.

```
# apply pca on the data
df_pca <- prcomp(df, center = TRUE, scale. = TRUE)

# preview the summary of the pca object
summary(df_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6
PC7
## Standard deviation     1.9836 1.0631 1.03159 1.00991 0.99289 0.9771
0.96270
## Proportion of Variance 0.3577 0.1027 0.09674 0.09272 0.08962 0.0868
0.08425
## Cumulative Proportion  0.3577 0.4604 0.55719 0.64991 0.73953 0.8263
0.91058
##                            PC8     PC9      PC10      PC11
## Standard deviation     0.94823 0.29062 2.758e-16 1.113e-16
## Proportion of Variance 0.08174 0.00768 0.000e+00 0.000e+00
## Cumulative Proportion  0.99232 1.00000 1.000e+00 1.000e+00
```

As a result, we obtain 11 principal components.

The first principal component explains a huge percentage of the variance at 35.77%. PC2 explains 10%, PC3 and PC4 explain 9% of the variance and PC5 and PC6 explain 8% of the variance.

We can conclude that the first 5 components explain 73.95% of the variance, hence we can reduce the dimensions of the original data to 5 components

```
# Calling str() to have a look at the PCA object
str(df_pca)
```

```
## List of 5
##  $ sdev    : num [1:11] 1.984 1.063 1.032 1.01 0.993 ...
##  $ rotation: num [1:11, 1:11] 0.0267 -0.0155 -0.0338 0.0206 0.3273 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11] "branch" "customer_type" "gender" "product_line" ...
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:11] 1.99 1.5 1.5 3.45 55.67 ...
##   ..- attr(*, "names")= chr [1:11] "branch" "customer_type" "gender"
"product_line" ...
##  $ scale   : Named num [1:11] 0.818 0.5 0.5 1.715 26.495 ...
##   ..- attr(*, "names")= chr [1:11] "branch" "customer_type" "gender"
"product_line" ...
##  $ x       : num [1:1000, 1:11] 1.79 -2.05 0.11 1.29 2.43 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

Here we note on the pca object:

- 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), and

-the values of each sample in terms of the principal components ($x)

Plotting the PCA object

```
# plotting the first 2 principal components
library(devtools)
```

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

```
library(ggbiplot)
```

```
## Loading required package: plyr
```

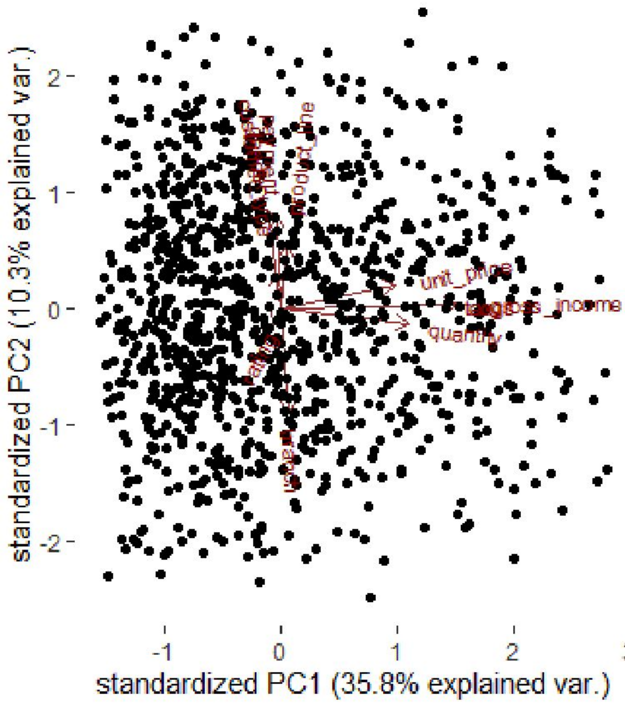
```
## Loading required package: scales
```

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

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

```
## Loading required package: grid
```

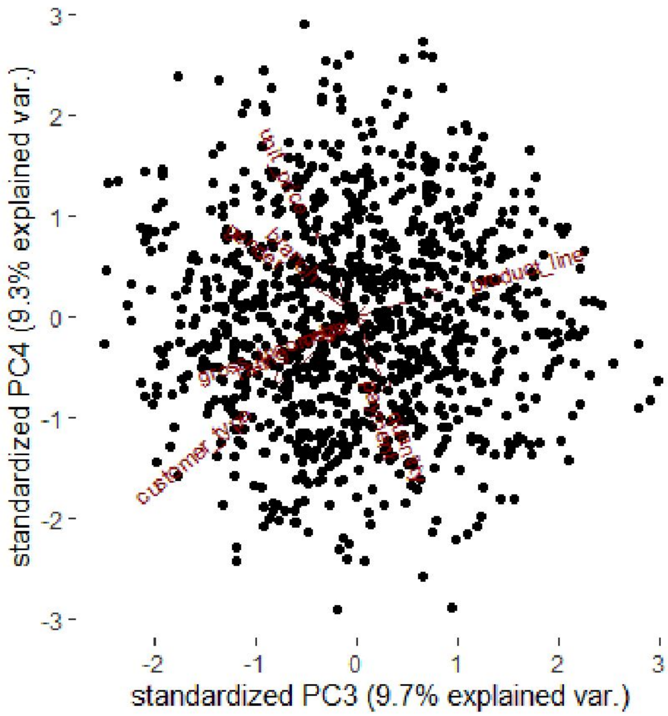
```r
ggbiplot(df_pca)
```



From the graph we can see that the variables quantity, unit price, gross income and tax contibute to PC1 with higher values in those variables moving the samples to the right on the plot, while variables such as product line and payment contribute to PC2.

The first 2 principal components explain 46% of the variance, which is almost half of the total variance.

```r
#plotting the third and fourth components
ggbiplot(df_pca,ellipse=TRUE,choices=c(3,4))
```



PC3 and PC4 explain very small percentages of the total variation hence it is difficult to derive insights from the plot.

```
str(superdata)
```

```
## 'data.frame':    1000 obs. of  16 variables:
##  $ invoice_id             : chr  "750-67-8428" "226-31-3081" "631-41-3108"
"123-19-1176" ...
##  $ branch                 : Factor w/ 3 levels "A","B","C": 1 3 1 1 1 3 1
3 1 2 ...
##  $ customer_type          : Factor w/ 2 levels "Member","Normal": 1 2 2 1
2 2 1 2 1 1 ...
##  $ gender                 : Factor w/ 2 levels "Female","Male": 1 1 2 2 2
2 1 1 1 1 ...
##  $ product_line           : Factor w/ 6 levels "Electronic
accessories",..: 4 1 5 4 6 1 1 5 4 3 ...
##  $ 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                : Factor w/ 3 levels "Cash","Credit card",..: 3
1 2 3 3 3 3 3 2 2 ...
##  $ 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                 : Factor w/ 61 levels "4","4.1","4.2",..: 52 57
```

```
35 45 14 2 19 41 33 20 ...
##  $ total                     : num  549 80.2 340.5 489 634.4 ...
```

Adding more detail to the plot of PC1 and PC2

```
library(ggfortify)
```

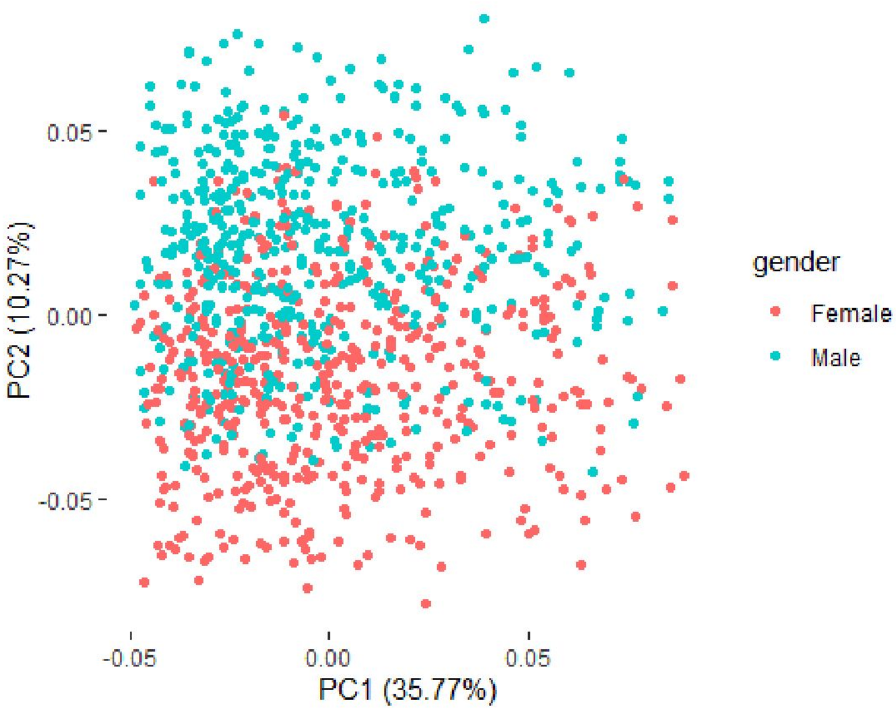
```
##
## Attaching package: 'ggfortify'
```

```
## The following object is masked from 'package:ggbiplot':
##
##     ggbiplot
```

```
pca.plot <- autoplot(df_pca, data = superdata, colour="gender")
```

```
## Warning: `select_()` is deprecated as of dplyr 0.7.0.
## Please use `select()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
pca.plot
```



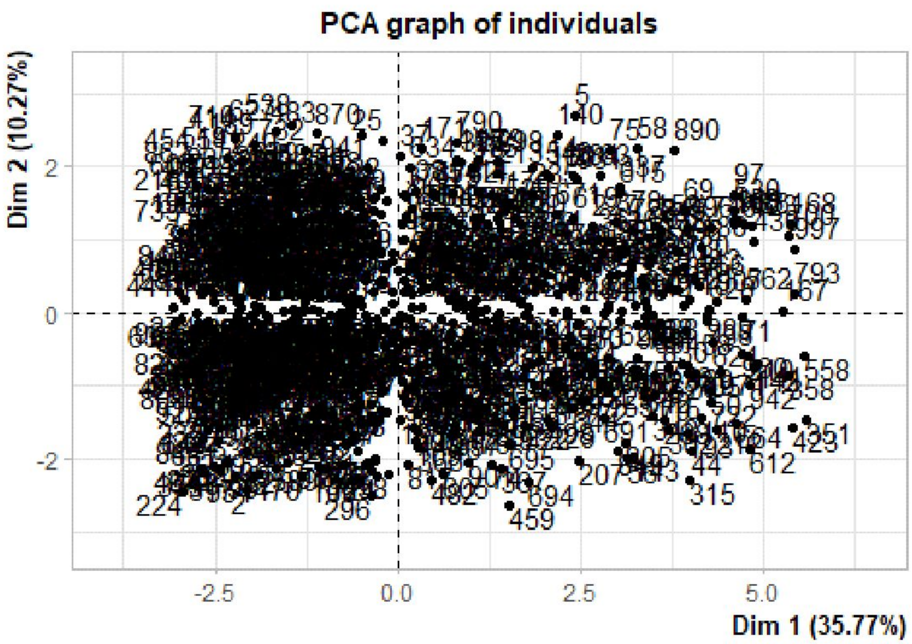
More females appear to be concentrated below in the graph than males who appear to be above in the graph implying that gender contibutes to PC2.

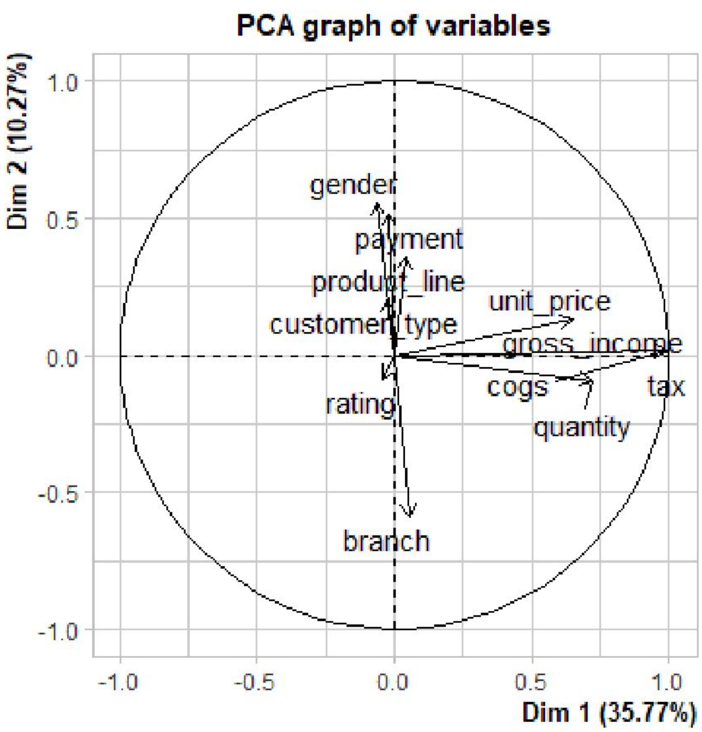
PCA dimension reduction with 5 components

Since we concluded earlier that 5 components explain a higher percentage of variance, we will go ahead and create a pca component of only 5 components

```r
library(FactoMineR)

# apply PCA
df_PCA = PCA(df, scale.unit = TRUE, ncp = 5, graph = TRUE)
```





From the graph of components, we can still observe the same variables we listed above to be contributing to PC1 and PC2

Interpreting Principal Components

```
#check the correlation of the features to the principal components
df_PCA$var$coord
```

```
##                    Dim.1       Dim.2       Dim.3        Dim.4
Dim.5
## branch        0.05288613 -0.59309158  0.17956247 -0.183291180
-0.0668974458
## customer_type -0.03081536  0.20195385  0.55277495  0.456780202
-0.2613730397
## gender        -0.06697299  0.55604352  0.38247162 -0.281717091
-0.0678688607
## product_line   0.04087902  0.35959700 -0.55708960 -0.163021569
0.3161573498
## unit_price     0.64916432  0.13018812  0.26187869 -0.552976395
0.0325917496
## quantity       0.72358694 -0.09209482 -0.20626211  0.504756839
-0.0038281177
## tax            0.99626090  0.01612665  0.01696622  0.006054973
-0.0003647549
## payment       -0.01876966  0.51525392 -0.07339405  0.285770908
0.0191124954
## cogs           0.99626090  0.01612665  0.01696622  0.006054973
-0.0003647549
## gross_income   0.99626090  0.01612665  0.01696622  0.006054973
-0.0003647549
## rating        -0.04332869 -0.08601070  0.39035353  0.171478850
0.8983529736
```

From the outputs, we can observe that:

- unit price, quantity, tax, cogs and gross income are highly correlated with PC1. This correlation suggests the five variables vary together and when one goes down, the others decrease as well
- branch, gender, payment are highly correlated with PC2
- customer type, product line are highly correlated with PC3
- rating is highly correlated with PC5

10.3 Feature Selection

This is a process that reduces the number of features in a dataset by excluding or including them without any change as opposed to dimensionality reduction methods which do so by creating new combinations of features.

For this project we will implement the **feature ranking** method of feature selection.

```
# load libraries
suppressWarnings(
```

```r
        suppressMessages(if
                         (!require(FSelector, quietly=TRUE))
                install.packages("FSelector")))

library(rJava)
library(FSelector)

# remove character datatypes from the data
feature_data <- data[c(-1, -9,-10)]
str(feature_data)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ branch                 : num  1 3 1 1 1 3 1 3 1 2 ...
##  $ customer_type          : num  1 2 2 1 2 2 1 2 1 1 ...
##  $ gender                 : num  1 1 2 2 2 2 1 1 1 1 ...
##  $ product_line           : num  4 1 5 4 6 1 1 5 4 3 ...
##  $ 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 ...
##  $ payment                : num  3 1 2 3 3 3 3 3 2 2 ...
##  $ 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  52 57 35 45 14 2 19 41 33 20 ...
##  $ total                  : num  549 80.2 340.5 489 634.4 ...
```

Feature ranking using correlations

From the FSelector package, we use the correlation coefficient as a unit of valuation to rank the variables by attribute importance.

```r
Scores <- linear.correlation(total~., feature_data)
Scores
```

```
##                           attr_importance
## branch                         0.04104666
## customer_type                  0.01967028
## gender                         0.04945099
## product_line                   0.03162072
## unit_price                     0.63396209
## quantity                       0.70551019
## tax                            1.00000000
## payment                        0.01243364
## cogs                           1.00000000
## gross_margin_percentage                NA
## gross_income                   1.00000000
## rating                         0.03644170
```

From the output above, we observe a list containing rows of variables and their corresponding scores on the right. We can observe that gross margin percentage score has not been included as its importance is very minimal. We saw earlier its variance was very low and hence it contributes a minimum percentage of information to the data.

We need to define a cutoff to select the top most representative variables.

```
# select the top 5 most representative features and output as a dataframe
Subset <- cutoff.k(Scores, 5)
as.data.frame(Subset)
```

```
##          Subset
## 1           tax
## 2          cogs
## 3 gross_income
## 4      quantity
## 5    unit_price
```

The columns: tax, cogs, gross income, quantity, unit price have been selected with a cutoff of 5. We can observe that these variables have higher correlations and hence higher importance.

Setting cutoff as a percentage would indicate that we would want to work with percentage of the best variables:

```
Subset2 <-cutoff.k.percent(Scores, 0.4)
as.data.frame(Subset2)
```

```
##         Subset2
## 1           tax
## 2          cogs
## 3 gross_income
## 4      quantity
## 5    unit_price
```

The same variables selected above have also been selected in the second subset.

Feature ranking using information gain

Instead of using the scores for the correlation coefficient, we can use an entropy - based approach. Does this change the variables selected?

```
Scores2 <- information.gain(total~., feature_data)
Scores2
```

```
##                          attr_importance
## branch                         0.0000000
## customer_type                  0.0000000
## gender                         0.0000000
## product_line                   0.0000000
## unit_price                     0.3084863
## quantity                       0.4211154
## tax                            1.6094379
## payment                        0.0000000
```

```
## cogs                              1.6094379
## gross_margin_percentage           0.0000000
## gross_income                      1.6094379
## rating                            0.0000000
```

Looking at the attribute importances, we can observe that categorical feature importances have been shrinked to 0.

```
Subset3 <- cutoff.k(Scores2, 5)
as.data.frame(Subset3)
```

```
##        Subset3
## 1          tax
## 2         cogs
## 3 gross_income
## 4     quantity
## 5   unit_price
```

The selected features are still similar to the ones selected using correlations. This features can then be used to build an unsupervised learning model.

Comparing to the correlation method, the entropy based approach is more strict.