Dimensionality Reduction

2022-06-09

R Markdown

1. Defining the question

a) Specifying the question

form the marketing department on the most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

b) Defining the metrics of success

explore a recent marketing dataset by performing various unsupervised learning techniques and later providing recommendations based on your insights. reduce your dataset to a low dimensional dataset using PCA

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.

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.

d) Recording the experimental design

1. Problem Definition

2. Data Sourcing
3. Check the Data
4. Perform Data Cleaning
5. Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate)
6. Implement the Solution
7. Challenge the Solution
8. Follow up Questions

e) Data relevance

The data is relevant since it was provided by the company itself and can be used to answer the question.

```
# Loading the dataset
#url <- http://bit.ly/CarreFourDataset
```

```r
df <- read.csv('C:\\Users\\USER\\Documents\\Moringa School\\R Programming\\Unsupervised learning 2\\IP\\
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
```

```r
# Preview the last 6 items in the 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
```

```r
# Checking the shape/ dimension of the dataframe
dim(df)
```

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

The dataset has 1000 rows and 16 columns

```r
# Checking the summury 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
# Checking the structure of the dataset
str(df)
```

```
## '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" "I
##  $ 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 ...
```

Data Cleaning

```
# Checking for uniformity in the column names of the dataset
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"
```

```
# Checking for 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
```

There are no missing values in the data set

```
# Checking for duplicates in the dataset
sum(duplicated(df))
```

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

There are no duplicates in the data set

```
# Selecting the numerical values in the dataset
library(dplyr)
```

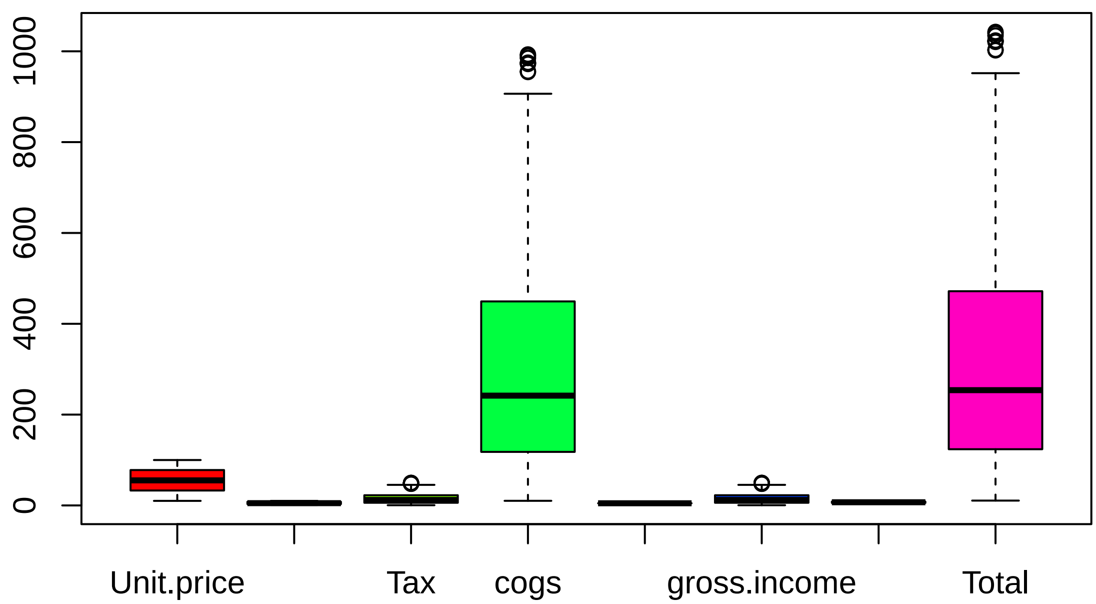
```
##
## Attaching package: 'dplyr'

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

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

```
numeric <- select_if(df, 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
```

```
# Looking for outliers
boxplot(numeric, col = rainbow(ncol(numeric)))
```



The tax, cogs, gross income and total columns have outliers but they shall be retained for further analysis

```
# Dropping unecessary columns
library(dplyr)
df1 = select(df, -c(Invoice.ID, Date, Time))
head(df1)
```

```
##   Branch Customer.type Gender           Product.line Unit.price Quantity
## 1      A        Member Female      Health and beauty      74.69        7
## 2      C        Normal Female Electronic accessories      15.28        5
## 3      A        Normal   Male     Home and lifestyle      46.33        7
## 4      A        Member   Male      Health and beauty      58.22        8
## 5      A        Normal   Male      Sports and travel      86.31        7
```

```
## 6      C        Normal   Male Electronic accessories      85.39        7
##       Tax     Payment   cogs gross.margin.percentage gross.income Rating
## 1 26.1415     Ewallet 522.83                4.761905      26.1415    9.1
## 2  3.8200        Cash  76.40                4.761905       3.8200    9.6
## 3 16.2155 Credit card 324.31                4.761905      16.2155    7.4
## 4 23.2880     Ewallet 465.76                4.761905      23.2880    8.4
## 5 30.2085     Ewallet 604.17                4.761905      30.2085    5.3
## 6 29.8865     Ewallet 597.73                4.761905      29.8865    4.1
##      Total
## 1 548.9715
## 2  80.2200
## 3 340.5255
## 4 489.0480
## 5 634.3785
## 6 627.6165
```

Exploratory Data Analysis

```r
# Import necessary libraries
library(dplyr)
library(tidyr)
```

```r
# Finding the mean of the numerical columns

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
# Finding the median of the numerical columns

df1 %>% 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
# Finding the range of the numerical columns

df1 %>% 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
```

```r
# Finding the standard deviation of the numerical columns

df1 %>% 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
```

```
# Finding the variance of the numerical columns

df1 %>% 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
```

```
# Finding the quantiles of the numerical columns

df1 %>% summarise_if(is.numeric, quantile)
```

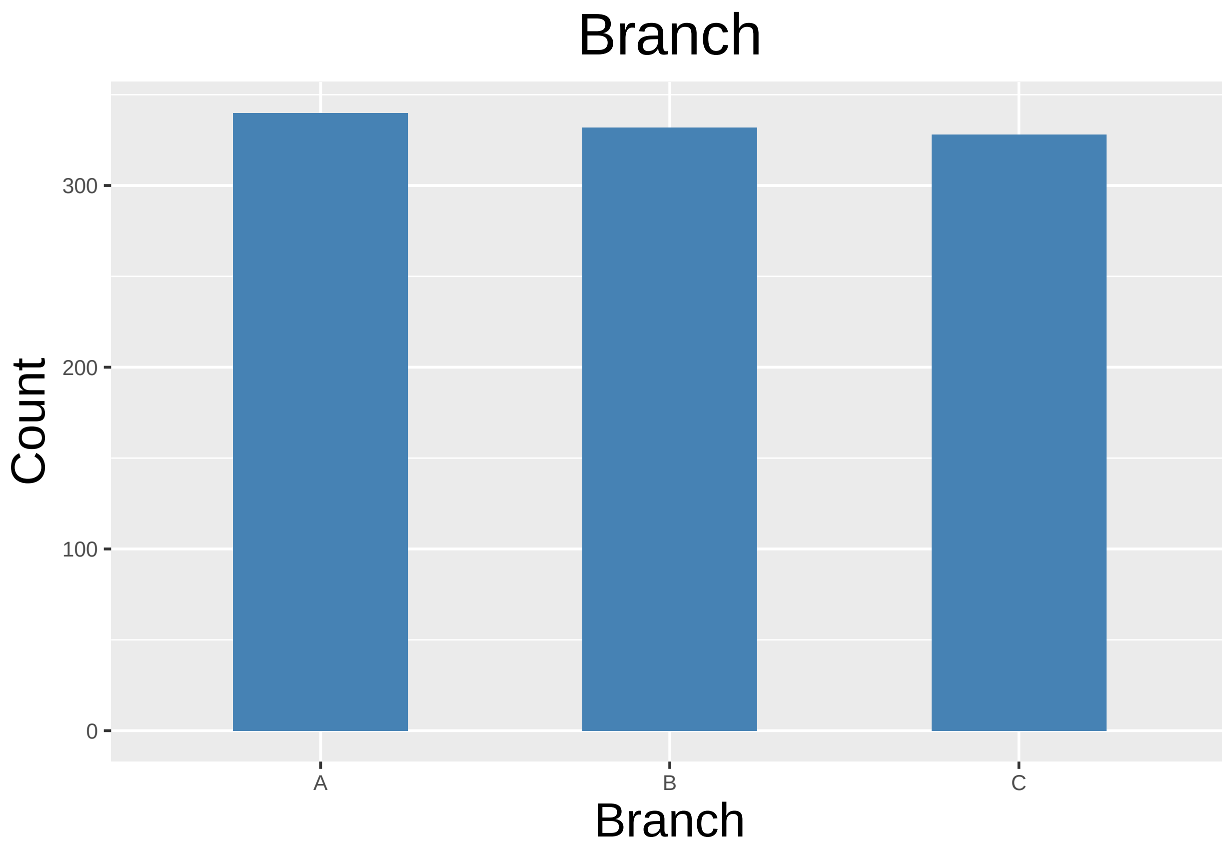
```
##   Unit.price Quantity       Tax     cogs gross.margin.percentage gross.income
## 1     10.080        1  0.508500  10.1700                4.761905     0.508500
## 2     32.875        3  5.924875 118.4975                4.761905     5.924875
## 3     55.230        5 12.088000 241.7600                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
```

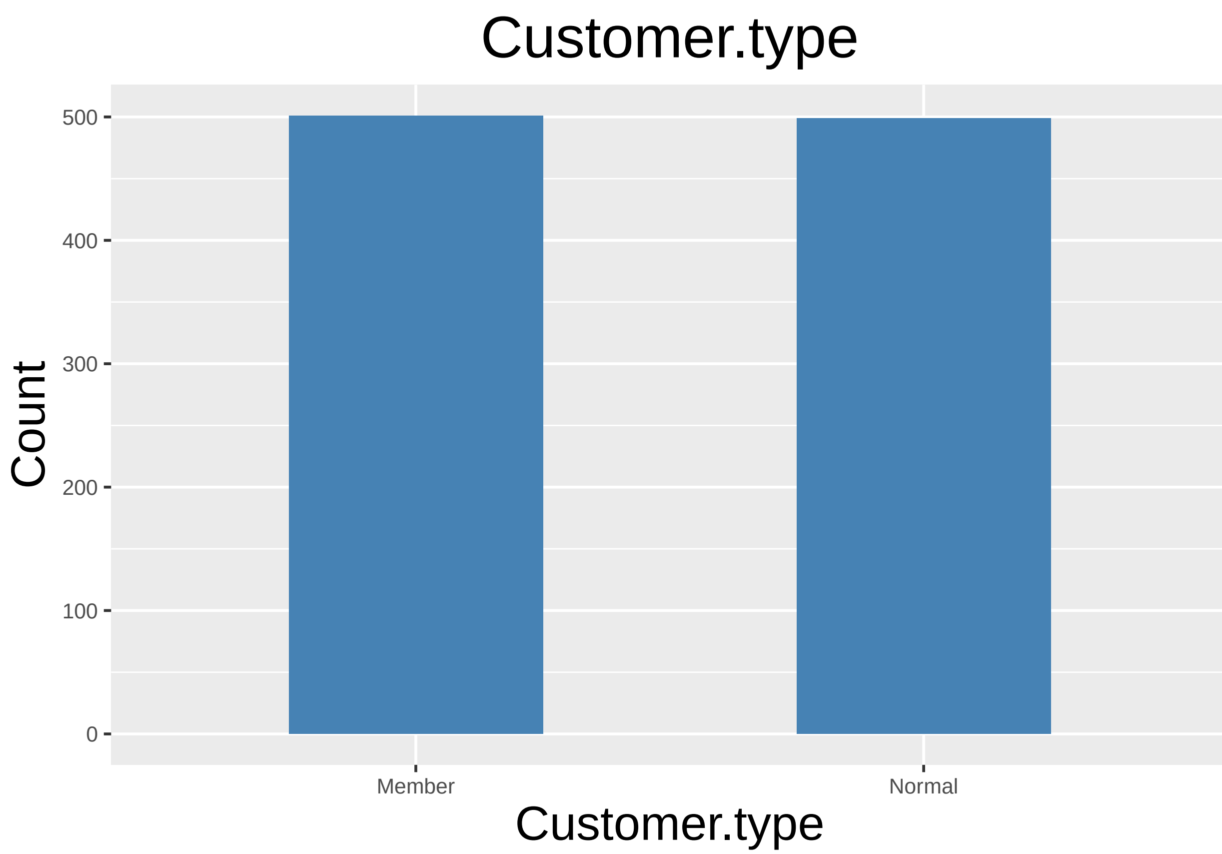
```
head(df1)
```

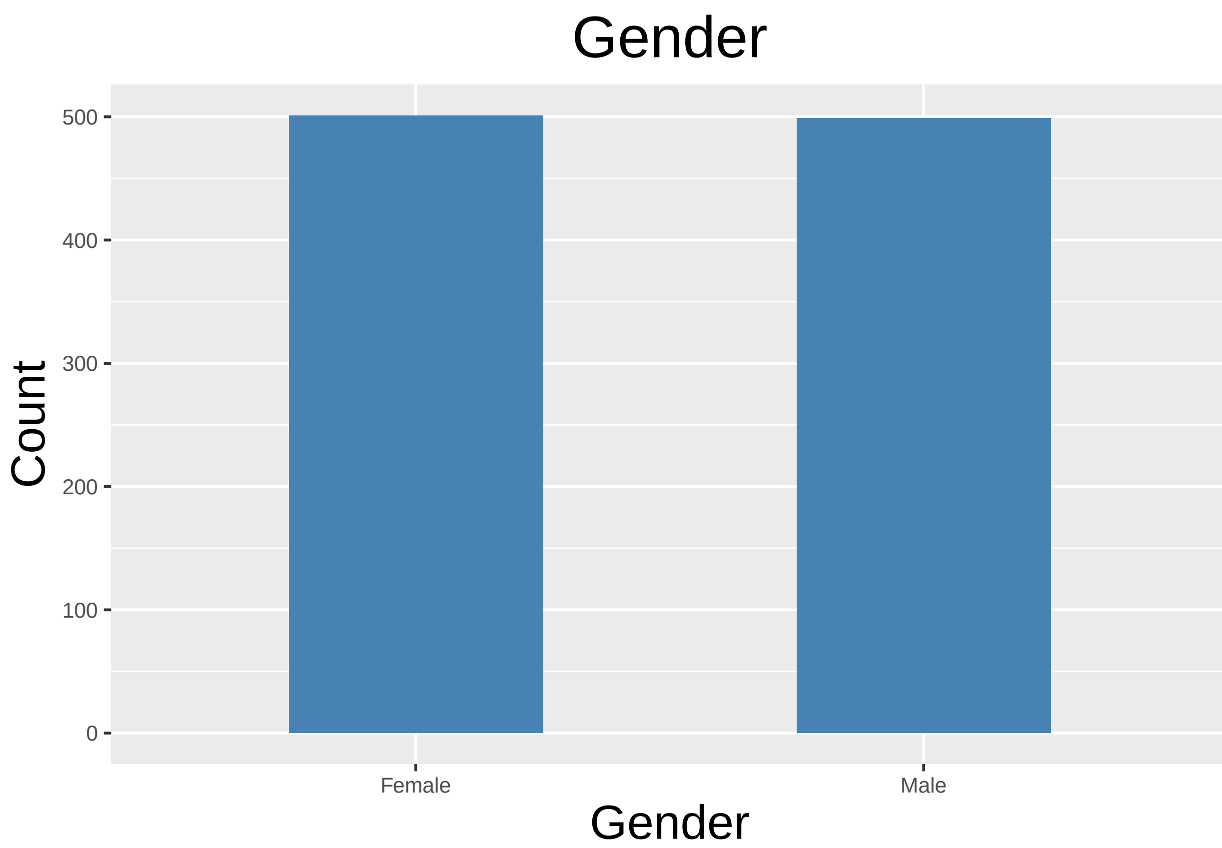
```
##   Branch Customer.type Gender           Product.line Unit.price Quantity
## 1      A        Member Female      Health and beauty      74.69        7
## 2      C        Normal Female Electronic accessories      15.28        5
## 3      A        Normal   Male     Home and lifestyle      46.33        7
## 4      A        Member   Male      Health and beauty      58.22        8
## 5      A        Normal   Male      Sports and travel      86.31        7
## 6      C        Normal   Male Electronic accessories      85.39        7
##       Tax     Payment   cogs gross.margin.percentage gross.income Rating
## 1 26.1415     Ewallet 522.83                4.761905      26.1415    9.1
## 2  3.8200        Cash  76.40                4.761905       3.8200    9.6
## 3 16.2155 Credit card 324.31                4.761905      16.2155    7.4
## 4 23.2880     Ewallet 465.76                4.761905      23.2880    8.4
## 5 30.2085     Ewallet 604.17                4.761905      30.2085    5.3
## 6 29.8865     Ewallet 597.73                4.761905      29.8865    4.1
##      Total
## 1 548.9715
## 2  80.2200
## 3 340.5255
## 4 489.0480
## 5 634.3785
## 6 627.6165
```

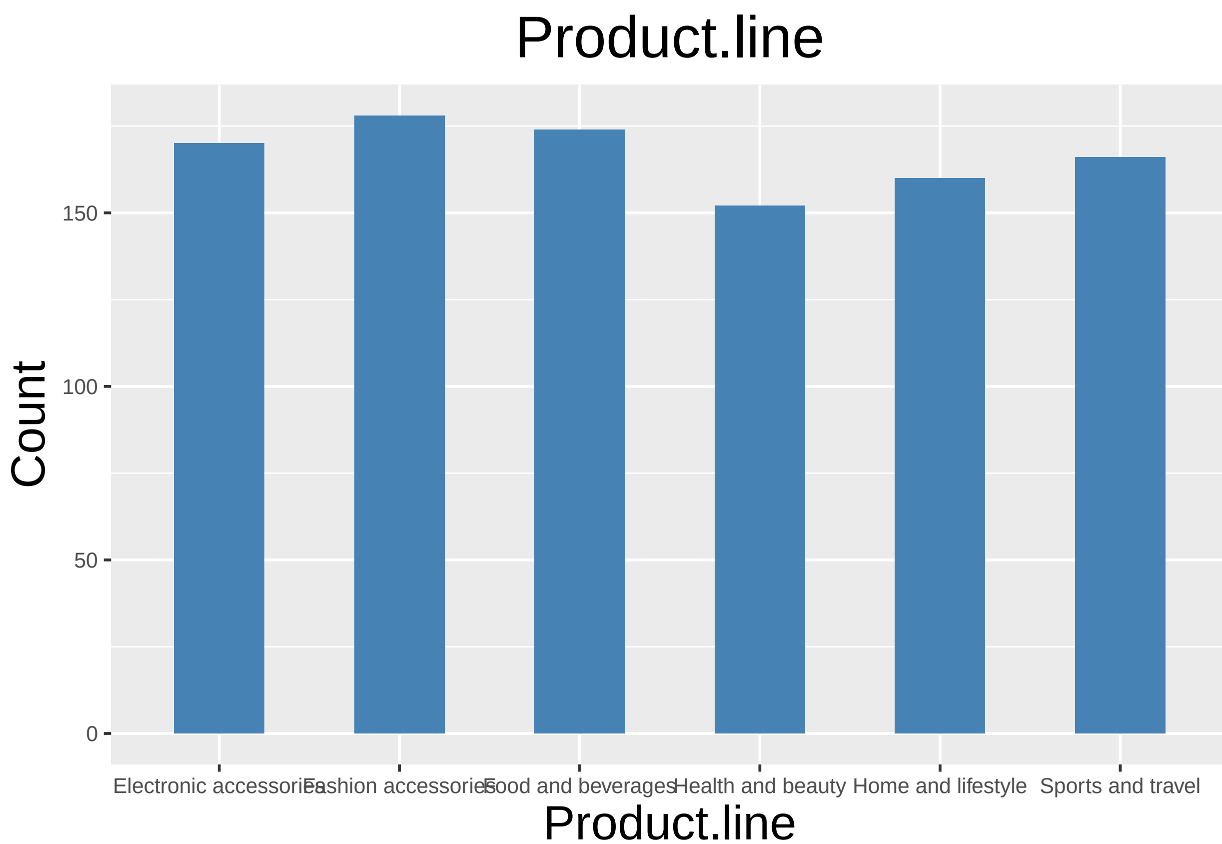
```
# Countplots for the categorical variables
library(ggplot2)
cat = df1[, c(1:4,8)]
```

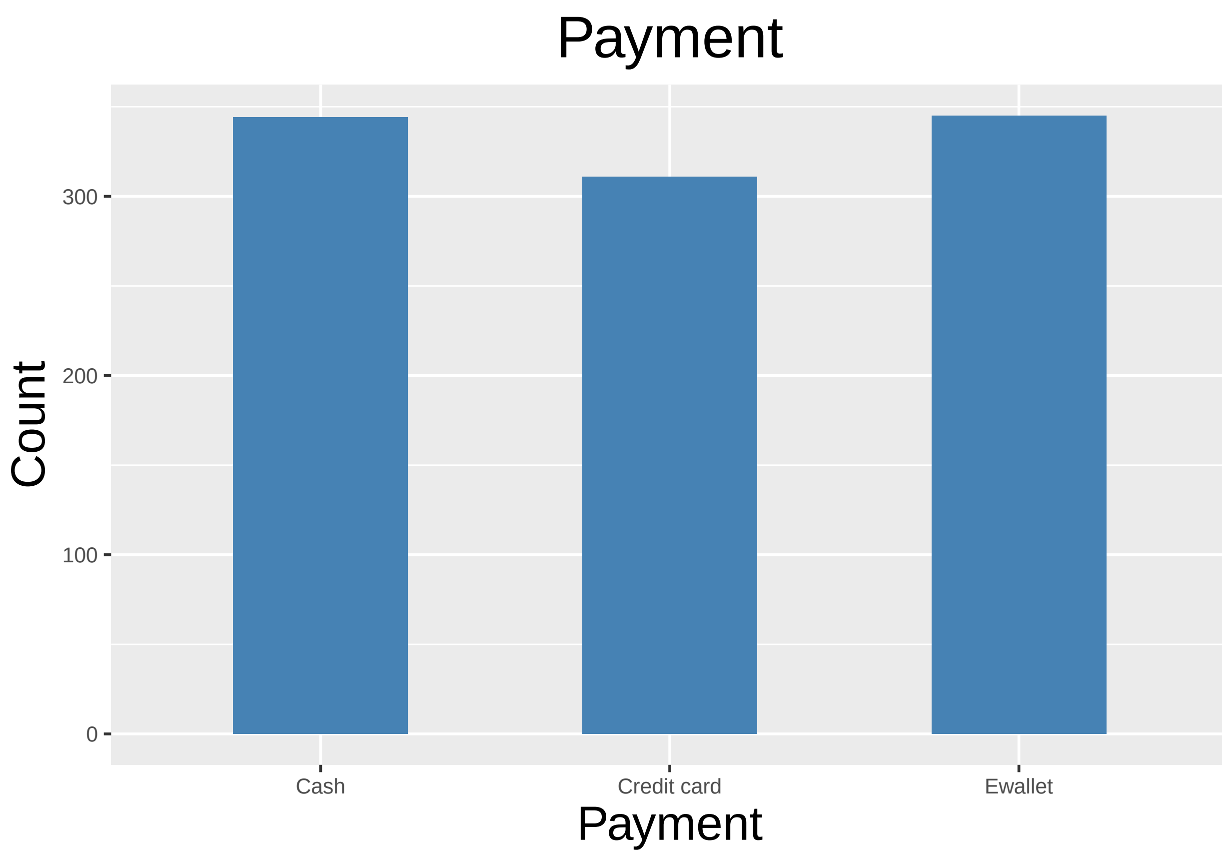
```
for (i in 1:length(cat)) {
  options(repr.plot.width = 30, repr.plot.height = 8)
  print(ggplot(cat, aes(x = factor(cat[,i]))) +
  geom_bar(fill = "steelblue", width = 0.5) +
  labs(title = names(cat[i]), x = names(cat[i]), y = "Count") +
  theme(axis.text = element_text(size=8),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 22)))
  cat("\n", "\n")
}
```




Customer.type
Count
500
400
300
200
100
0
Member
Normal
Customer.type


Gender
Count
500
400
300
200
100
0
Female
Male
Gender


Product.line
Count
0
50
100
150
Electronic accessories
Fashion accessories
Food and beverages
Health and beauty
Home and lifestyle
Sports and travel
Product.line




```
# Checking the actual count in gender
table(df1$Gender)
```

```
##
## Female   Male
##    501    499
```

```
# Checking the actual count in gender
table(df1$Customer.type)
```

```
##
## Member Normal
##    501    499
```

Most of the payment was done via e-wallet, followed by cash then credit card Most clients used branch A There was a difference in 2 people between the members and normal people There were 2 more females than males Fashion accessories accounted for majority of the sales while health and beauty for the least

Bivariate Analysis

```
# Installing GGally package to plot the pairplot
library("GGally")
```

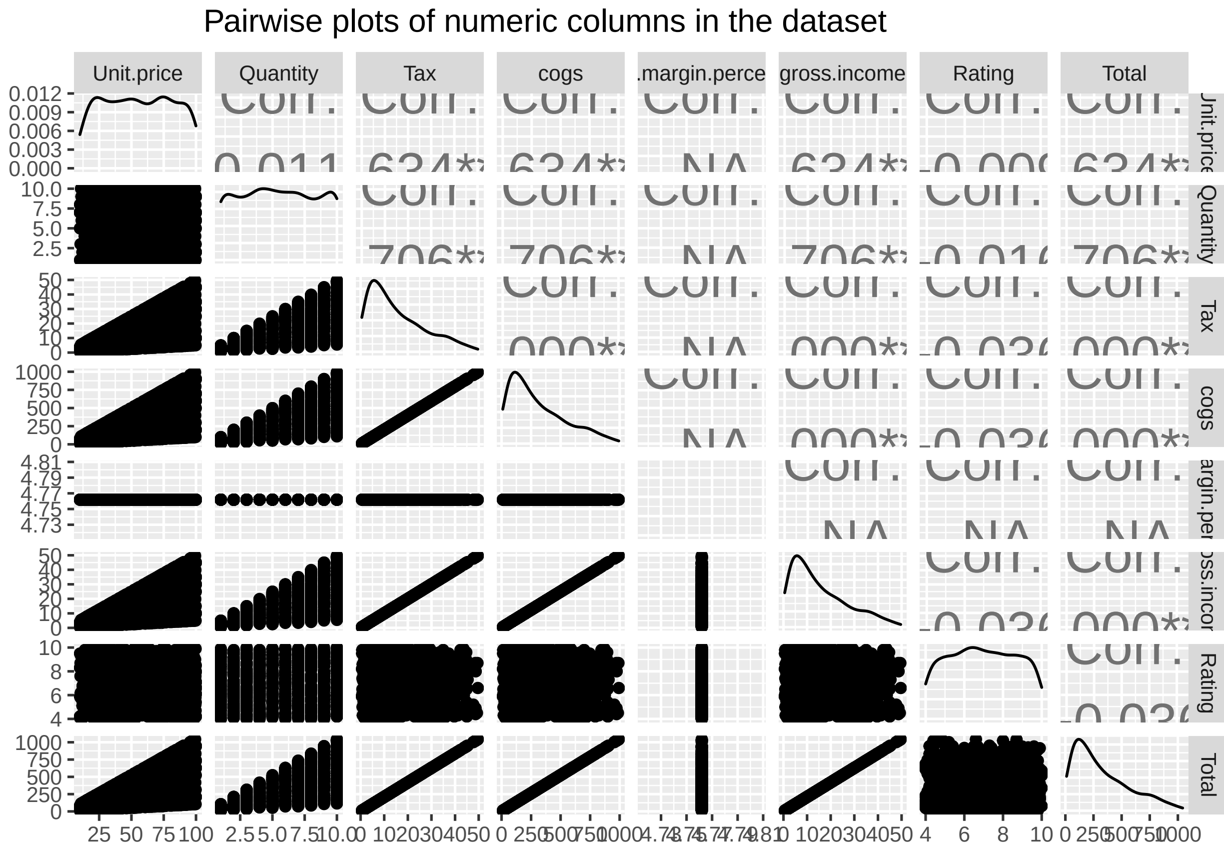
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
head(df1)
```

```
##   Branch Customer.type Gender           Product.line Unit.price Quantity
## 1      A        Member Female      Health and beauty      74.69        7
## 2      C        Normal Female Electronic accessories      15.28        5
## 3      A        Normal   Male     Home and lifestyle      46.33        7
## 4      A        Member   Male      Health and beauty      58.22        8
## 5      A        Normal   Male      Sports and travel      86.31        7
## 6      C        Normal   Male Electronic accessories      85.39        7
##       Tax     Payment   cogs gross.margin.percentage gross.income Rating
## 1 26.1415     Ewallet 522.83                4.761905      26.1415    9.1
## 2  3.8200        Cash  76.40                4.761905       3.8200    9.6
## 3 16.2155 Credit card 324.31                4.761905      16.2155    7.4
## 4 23.2880     Ewallet 465.76                4.761905      23.2880    8.4
## 5 30.2085     Ewallet 604.17                4.761905      30.2085    5.3
## 6 29.8865     Ewallet 597.73                4.761905      29.8865    4.1
##      Total
## 1 548.9715
## 2  80.2200
## 3 340.5255
## 4 489.0480
## 5 634.3785
## 6 627.6165
```

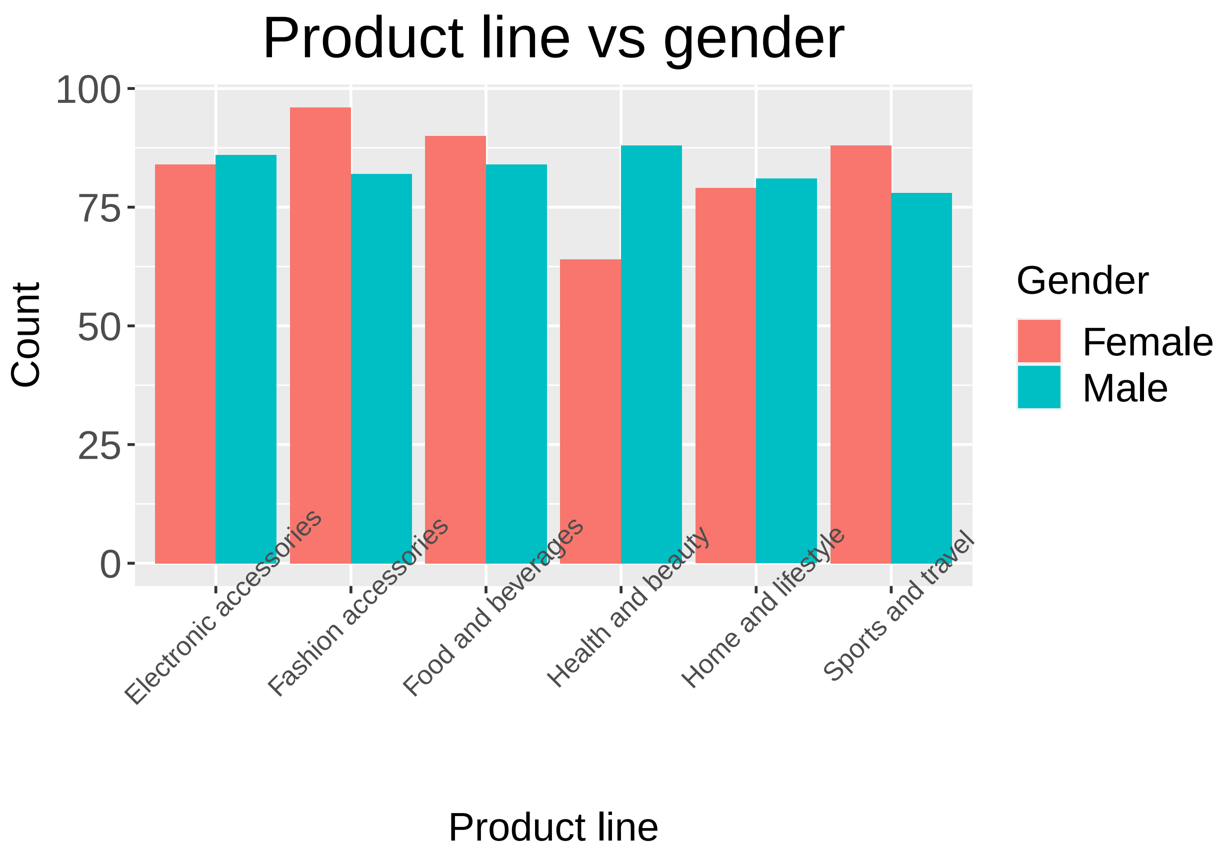
```r
# Plotting pair plots for numeric columns

options(repr.plot.width = 40, repr.plot.height = 18)
ggpairs(df1[, c(5:7, 9:13)], upper = list(continuous = wrap("cor", size = 7))) +
labs(title = "Pairwise plots of numeric columns in the dataset") +
    theme_grey(base_size = 10) +
    theme(plot.title = element_text(hjust = 0.3))
```

```
## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero

## Warning in cor(x, y): the standard deviation is zero
```



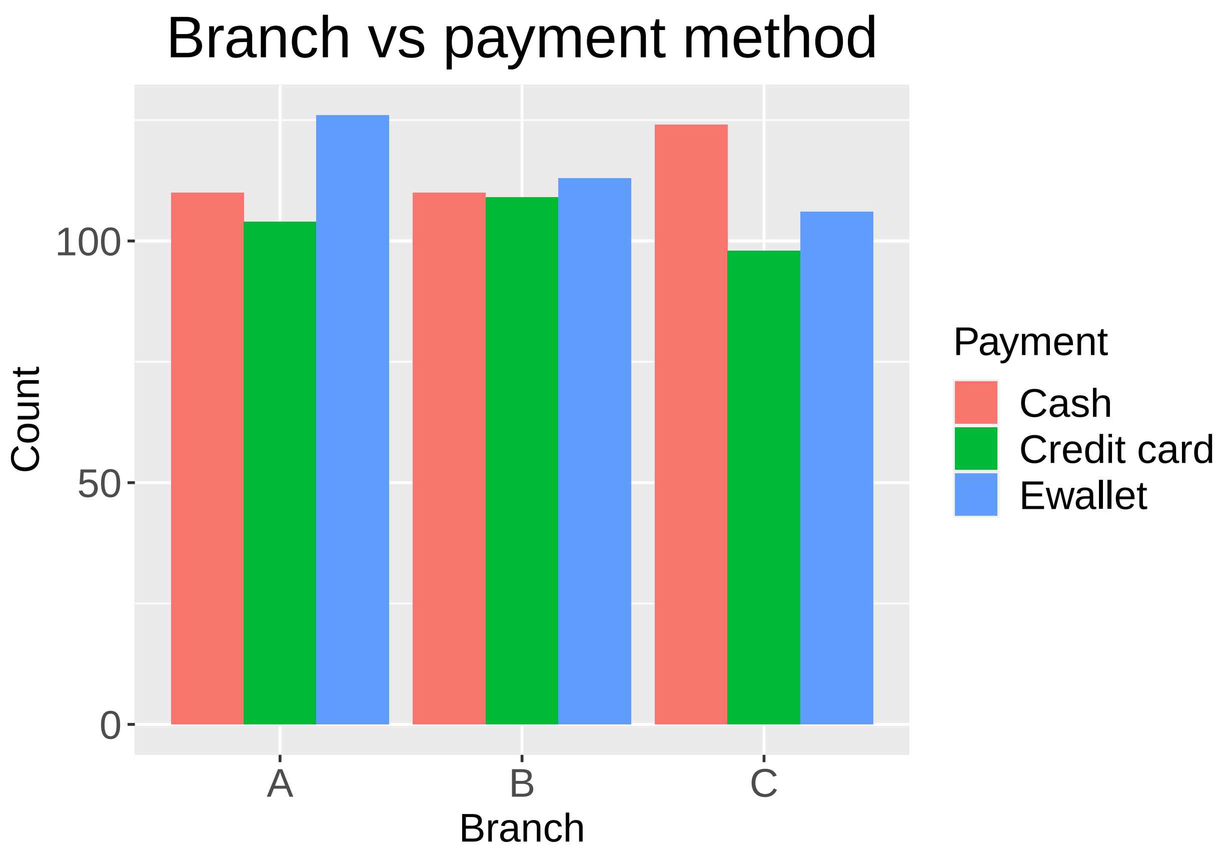
There is a positive correlation between tax and cogs, tax and gross income, tax and total, cogs and total

```
# Plot of product line vs gender
options(repr.plot.width = 20, repr.plot.height = 10)
ggplot(df1, aes(x = Product.line, fill = Gender)) +
  geom_bar(position = "dodge") +
  labs(title = "Product line vs gender", x = "Product line", y = "Count") +
    theme(axis.text.x = element_text(size=10, angle = 45),
          axis.text.y = element_text(size=15),
          axis.title = element_text(size = 15),
          plot.title = element_text(hjust = 0.5, size = 22),
          legend.title = element_text(size=15),
          legend.text = element_text(size=15))
```



males were known to purchase electronic accessories, health and beauty products and home and lifestyle products females were known to purchase more of fashion and accessories, food and beverages, sports and travel

```
# Plot between payment method and branch

options(repr.plot.width = 15, repr.plot.height = 8)
ggplot(df, aes(x = Branch, fill = Payment)) +
  geom_bar(position = "dodge") +
  labs(title = "Branch vs payment method", x = "Branch", y = "Count") +
    theme(axis.text = element_text(size=15),
          axis.title = element_text(size = 15),
          plot.title = element_text(hjust = 0.5, size = 22),
          legend.title = element_text(size=15),

                legend.text = element_text(size=15))
```



Cash was most used in branch C, credit card in branch B and e-wallets in branch A

```
# Loading package ggcorrplot

library(ggcorrplot)
```

```
# Finding the covariance
cov(numeric)
```

```
##                          Unit.price     Quantity          Tax        cogs
## Unit.price              701.9653313   0.83477848  196.6683401  3933.36680
## Quantity                  0.8347785   8.54644645   24.1495704   482.99141
## Tax                     196.6683401  24.14957038  137.0965941  2741.93188
## cogs                   3933.3668019 482.99140761 2741.9318829 54838.63766
## gross.margin.percentage   0.0000000   0.00000000    0.0000000     0.00000
## gross.income            196.6683401  24.14957038  137.0965941  2741.93188
## Rating                   -0.3996675  -0.07945646   -0.7333003   -14.66601
## Total                  4130.0351420 507.14097799 2879.0284770 57580.56954
##                         gross.margin.percentage gross.income       Rating
## Unit.price                                    0  196.6683401  -0.39966752
## Quantity                                      0   24.1495704  -0.07945646
## Tax                                           0  137.0965941  -0.73330028
## cogs                                          0 2741.9318829 -14.66600553
## gross.margin.percentage                       0    0.0000000   0.00000000
## gross.income                                  0  137.0965941  -0.73330028
## Rating                                        0   -0.7333003   2.95351823
## Total                                         0 2879.0284770 -15.39930581
##                               Total
```

```
## Unit.price                4130.03514
## Quantity                   507.14098
## Tax                       2879.02848
## cogs                     57580.56954
## gross.margin.percentage      0.00000
## gross.income              2879.02848
## Rating                     -15.39931
## Total                    60459.59802
```

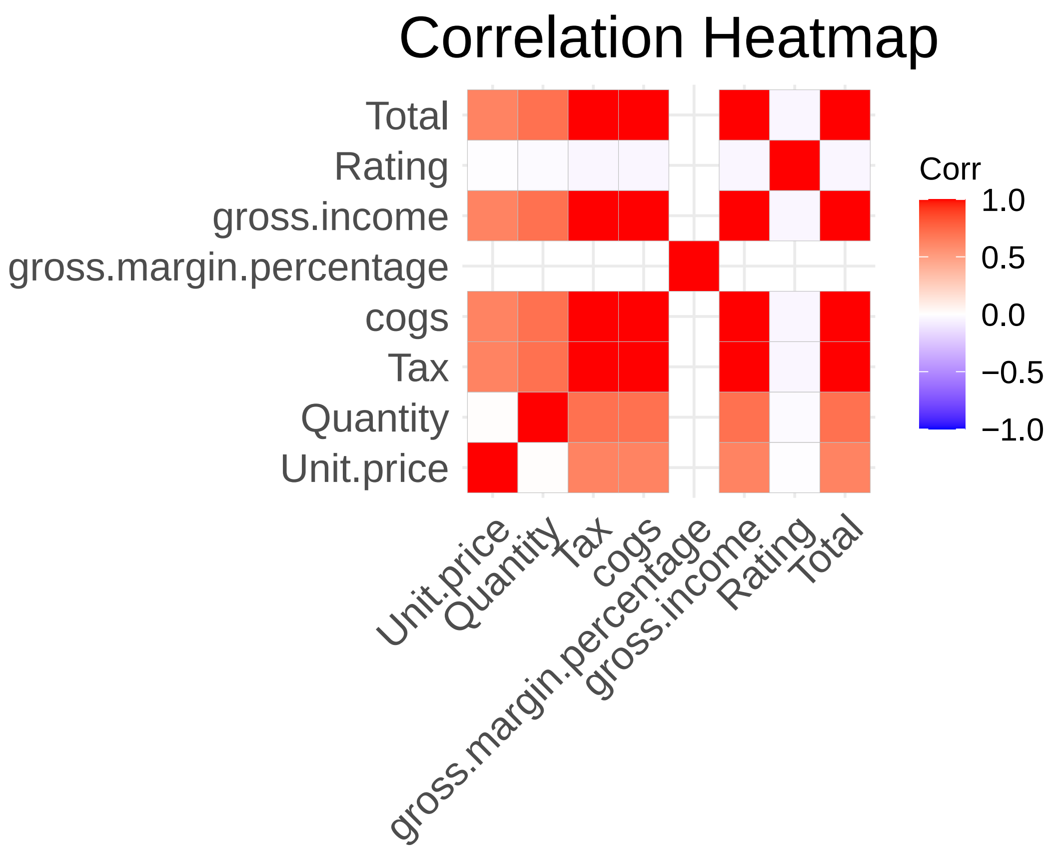
```r
# Finding the correlation matrix
cor(numeric)
```

```
## Warning in cor(numeric): the standard deviation is zero
```

```
##                           Unit.price    Quantity        Tax       cogs
## Unit.price               1.000000000  0.01077756  0.6339621  0.6339621
## Quantity                 0.010777564  1.00000000  0.7055102  0.7055102
## Tax                      0.633962089  0.70551019  1.0000000  1.0000000
## cogs                     0.633962089  0.70551019  1.0000000  1.0000000
## gross.margin.percentage           NA          NA         NA         NA
## gross.income             0.633962089  0.70551019  1.0000000  1.0000000
## Rating                  -0.008777507 -0.01581490 -0.0364417 -0.0364417
## Total                    0.633962089  0.70551019  1.0000000  1.0000000
##                         gross.margin.percentage gross.income       Rating
## Unit.price                                   NA    0.6339621 -0.008777507
## Quantity                                     NA    0.7055102 -0.015814905
## Tax                                          NA    1.0000000 -0.036441705
## cogs                                         NA    1.0000000 -0.036441705
## gross.margin.percentage                       1           NA           NA
## gross.income                                 NA    1.0000000 -0.036441705
## Rating                                       NA   -0.0364417  1.000000000
## Total                                        NA    1.0000000 -0.036441705
##                              Total
## Unit.price               0.6339621
## Quantity                 0.7055102
## Tax                      1.0000000
## cogs                     1.0000000
## gross.margin.percentage         NA
## gross.income             1.0000000
## Rating                  -0.0364417
## Total                    1.0000000
```

```r
# Plotting the correlation matrix
options(repr.plot.width = 15, repr.plot.height = 10)
ggcorrplot(cor(numeric), tl.cex = 15) +
  labs(title = "Correlation Heatmap") +
    theme(axis.title = element_text(size = 12),
          plot.title = element_text(hjust = 0.5, size = 22),
          legend.title = element_text(size=12),
          legend.text = element_text(size=12))
```

```
## Warning in cor(numeric): the standard deviation is zero
```



Feature Selection

Filter method

```r
# Getting the structure of the columns
str(df1)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ 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 ...
##  $ 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 ...
```

```r
# Encoding the branch, customer type, Gender, Payment and quantity columns

df1$Branch = ifelse(df1$Branch == "A", 0,
                  ifelse(df$Branch == "B", 1,2))
```

```r
df1$Customer.type = ifelse(df1$Customer.type == "Normal", 0, 1)
df1$Gender = ifelse(df1$Gender == "Male", 0, 1)
df1$Payment = ifelse(df1$Payment == "Cash", 0,
                  ifelse(df1$Payment == "Credit card", 1,2))

df1$Product.line = ifelse(df1$Product.line == "Electronic accessories", 0,
            ifelse(df1$Product.line == "Fashion accessories", 1,
            ifelse(df1$Product.line == "Food and beverages", 2,
            ifelse(df1$Product.line == "Health and beauty", 3,
            ifelse(df1$Product.line == "Home and lifestyle", 4, 5)))))

head(df1)
```

```
##   Branch Customer.type Gender Product.line Unit.price Quantity     Tax Payment
## 1      0             1      1            3      74.69        7 26.1415       2
## 2      2             0      1            0      15.28        5  3.8200       0
## 3      0             0      0            4      46.33        7 16.2155       1
## 4      0             1      0            3      58.22        8 23.2880       2
## 5      0             0      0            5      86.31        7 30.2085       2
## 6      2             0      0            0      85.39        7 29.8865       2
##     cogs gross.margin.percentage gross.income Rating    Total
## 1 522.83                4.761905      26.1415    9.1 548.9715
## 2  76.40                4.761905       3.8200    9.6  80.2200
## 3 324.31                4.761905      16.2155    7.4 340.5255
## 4 465.76                4.761905      23.2880    8.4 489.0480
## 5 604.17                4.761905      30.2085    5.3 634.3785
## 6 597.73                4.761905      29.8865    4.1 627.6165
```

```r
# Loading the caret and mlbench packages

library(mlbench)
library(caret)
```

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

```r
# Filtering the numerical columns

num = df1[, c(5,7,9,11,12)]
head(num)
```

```
##   Unit.price     Tax   cogs gross.income Rating
## 1      74.69 26.1415 522.83      26.1415    9.1
## 2      15.28  3.8200  76.40       3.8200    9.6
## 3      46.33 16.2155 324.31      16.2155    7.4
## 4      58.22 23.2880 465.76      23.2880    8.4
## 5      86.31 30.2085 604.17      30.2085    5.3
## 6      85.39 29.8865 597.73      29.8865    4.1
```

```r
# Determining the correlated features

set.seed(5)

# calculate correlation matrix
correlationMatrix <- cor(num)

# find attributes that are highly corrected (ideally >0.75)
```

```r
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.75)

# print indexes of highly correlated attributes
names(num[, highlyCorrelated])
```

```
## [1] "Tax"  "cogs"
```

The Tax, cogs are considered the redundant columns since they are highly correlated

```r
# Removing the redundant columns and gross percentage column

df2 <- num[-highlyCorrelated]
head(df2)
```

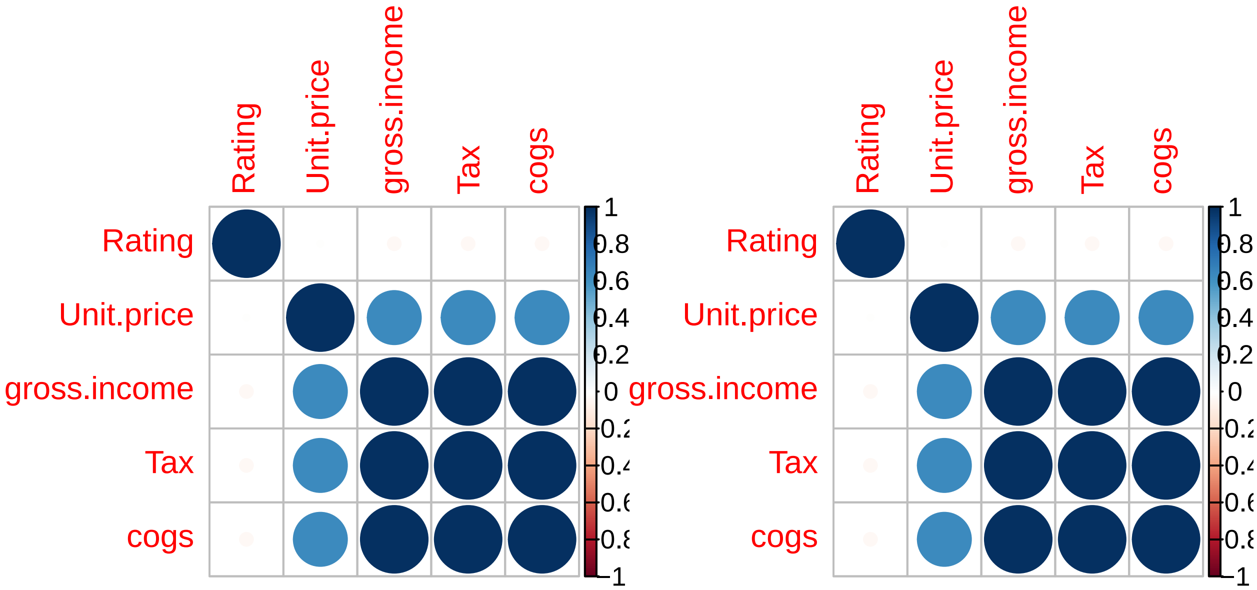
```
##   Unit.price gross.income Rating
## 1      74.69      26.1415    9.1
## 2      15.28       3.8200    9.6
## 3      46.33      16.2155    7.4
## 4      58.22      23.2880    8.4
## 5      86.31      30.2085    5.3
## 6      85.39      29.8865    4.1
```

Feature Selection

```r
# Performing our graphical comparison

library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
par(mfrow = c(1, 2))
corrplot(correlationMatrix, order = "hclust")
corrplot(cor(num), order = "hclust")
```




```
# Checking if there is a zero column to unit variance
# Dropping gross.margin.percentage column since it has a standard deviation of 0

df3 <- within(num, rm(gross.margin.percentage))
```

```
## Warning in rm(gross.margin.percentage): object 'gross.margin.percentage' not
## found
```

```
head(df2)
```

```
##   Unit.price gross.income Rating
## 1      74.69      26.1415    9.1
## 2      15.28       3.8200    9.6
## 3      46.33      16.2155    7.4
## 4      58.22      23.2880    8.4
## 5      86.31      30.2085    5.3
## 6      85.39      29.8865    4.1
```

```
# Pass df to the prcomp(). We also set two arguments, center and scale

df.pca <- prcomp(df3, center = TRUE, scale. = TRUE)
summary(df.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3       PC4       PC5
## Standard deviation     1.8673 0.9996 0.7171 2.415e-16 1.533e-16
## Proportion of Variance 0.6973 0.1998 0.1028 0.000e+00 0.000e+00
## Cumulative Proportion  0.6973 0.8972 1.0000 1.000e+00 1.000e+00
```

As a result we obtain 5 principal components, each which explain a percentage of the total variation of the dataset. PC1 explains 69.7% of the total variance, which means that nearly two-thirds of the information in the dataset (5 variables) can be encapsulated by just that one Principal Component. PC2 and 3 explains 29% of the variance.

```
# Calling str() to have a look at your PCA object
# ---
#
str(df.pca)
```

```
## List of 5
##  $ sdev    : num [1:5] 1.87 1.00 7.17e-01 2.41e-16 1.53e-16
##  $ rotation: num [1:5, 1:5] -0.4039 -0.528 -0.528 -0.528 0.0246 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "Unit.price" "Tax" "cogs" "gross.income" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 55.67 15.38 307.59 15.38 6.97
##   ..- attr(*, "names")= chr [1:5] "Unit.price" "Tax" "cogs" "gross.income" ...
##  $ scale   : Named num [1:5] 26.49 11.71 234.18 11.71 1.72
##   ..- attr(*, "names")= chr [1:5] "Unit.price" "Tax" "cogs" "gross.income" ...
##  $ x       : num [1:1000, 1:5] -1.7153 2.2171 0.0354 -1.0882 -2.4971 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```
# Plotting our pca. This will provide us with some very useful insights i.e.
# Installing our ggbiplot visualisation package


library(devtools)
```

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

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

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools 4.2 from https://cran.r-project.org/bin/windows/Rtools/ or https:/

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

```
# Loading our ggbiplot library

library(ggbiplot)
```

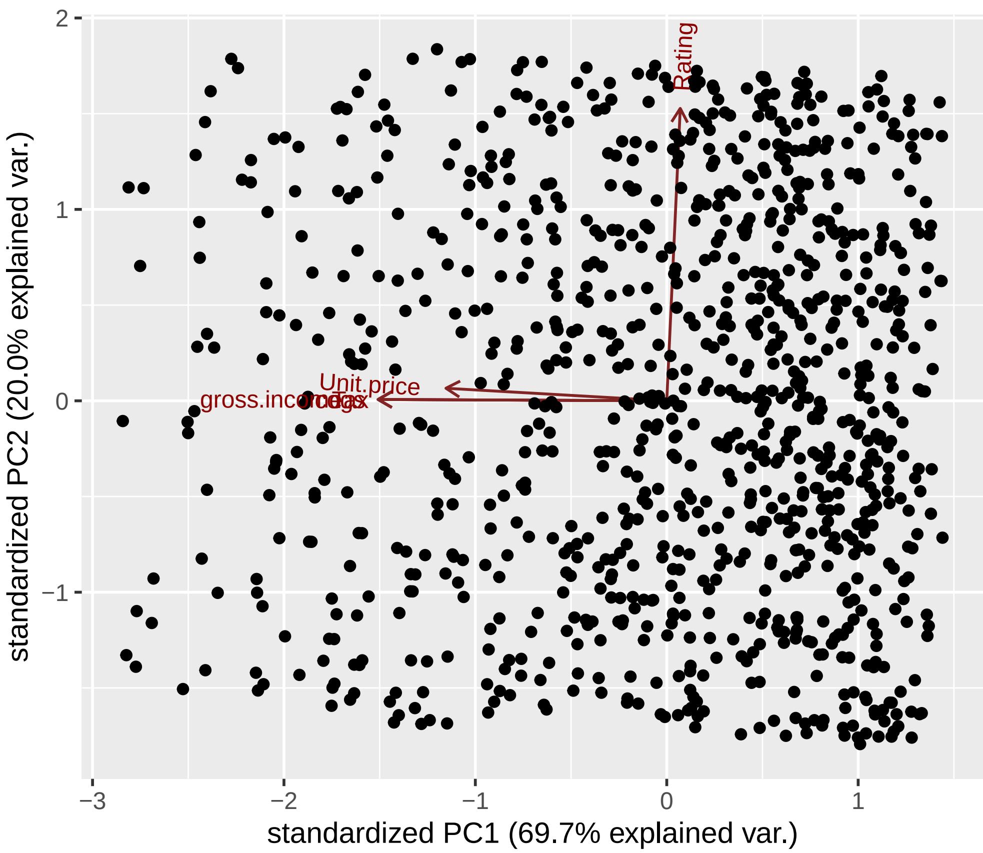
```
## Loading required package: plyr

## ------------------------------------------------------------------------------

## 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:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

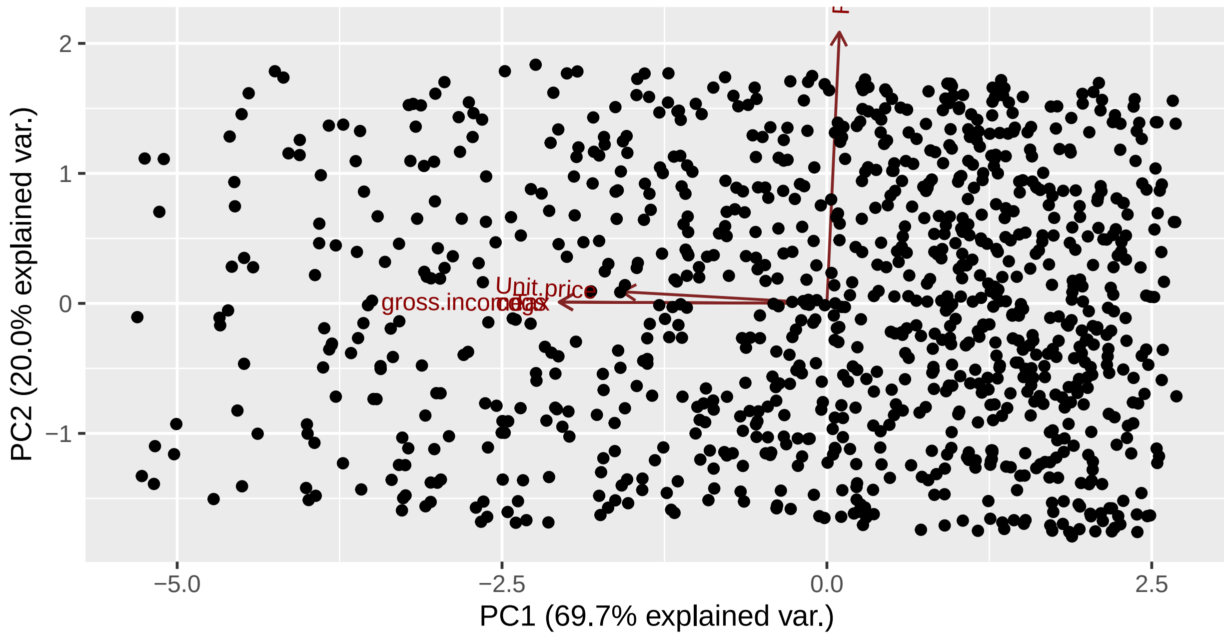
```
## Loading required package: scales
```

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

```
# Visualizing our plot

ggbiplot(df.pca)
```




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

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


PC2 (20.0% explained var.)
PC1 (69.7% explained var.)
2
1
0
−1
−5.0
−2.5
0.0
2.5
Unit.price