Dimensionality_Reduction in R

Dennis Kiarie

9/9/2021

```
library(tinytex)
```

1. Define the Question

1.1 Research Question

Our Research seeks to extract most relevant marketing strategies that will result in the highest no. of sales (total price including tax).

1.2 Metric of Success

To reduce the dataset to a lower dimension using PCA, perform the analysis and provide insights that will result in the highest no. of sales.

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

1.4 Experimental Design

1. Loading Data into RStudio.
2. Checking the Data.
3. Tidying the Data.
4. Conducting Exploratory Data Analysis i.e Univariate, Bivariate and Multivariate Analysis.
5. Reduce the dataset to a low dimensional dataset using the PCA
6. Implement the Solution
7. Challenge the Solution
8. Follow up Questions

1.5 Data Relevance

The data provided its appropriate for our analysis. The dataset for this analysis can be found in this link:[http://bit.ly/CarreFourDataset]

Description

The dataset consists of 1000 records and 6 features.

2.Data Preparation

```r
## Importing libraries
#---
#
library(pacman)
library(data.table)
pacman :: p_load(pacman,ggbiplot,plyr, dplyr,scales, readr, grid,factoextra, GGally,DataExplorer, ggplo
theme_set(theme_classic())
options(warn = -1)
```

```r
## Loading the data from a csv file
#---
#
df <- fread('http://bit.ly/CarreFourDataset')
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
##   ---
##  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
##    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
##    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
```

```
##preview the first five records
#---
#
head(df, n=5)
```

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

##preview the last 6 records of the dataset #—

```
tail(df)
```

```
##     Invoice ID Branch Customer type Gender           Product line Unit price
## 1: 652-49-6720      C        Member Female Electronic accessories      60.95
## 2: 233-67-5758      C        Normal   Male      Health and beauty      40.35
## 3: 303-96-2227      B        Normal Female     Home and lifestyle      97.38
## 4: 727-02-1313      A        Member   Male     Food and beverages      31.84
## 5: 347-56-2442      A        Normal   Male     Home and lifestyle      65.82
## 6: 849-09-3807      A        Member Female    Fashion accessories      88.34
##    Quantity     Tax      Date  Time Payment   cogs gross margin percentage
## 1:        1  3.0475 2/18/2019 11:40 Ewallet  60.95                4.761905
## 2:        1  2.0175 1/29/2019 13:46 Ewallet  40.35                4.761905
## 3:       10 48.6900  3/2/2019 17:16 Ewallet 973.80                4.761905
## 4:        1  1.5920  2/9/2019 13:22    Cash  31.84                4.761905
## 5:        1  3.2910 2/22/2019 15:33    Cash  65.82                4.761905
## 6:        7 30.9190 2/18/2019 13:28    Cash 618.38                4.761905
##    gross income Rating     Total
## 1:       3.0475    5.9   63.9975
## 2:       2.0175    6.2   42.3675
## 3:      48.6900    4.4 1022.4900
## 4:       1.5920    7.7   33.4320
## 5:       3.2910    4.1   69.1110
## 6:      30.9190    6.6  649.2990
```

3. Checking the data

```r
##preview the dataset
#---
#
View(df)
```

```r
##we check for the shape of the data
#---
#
dim(df)
```

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

```r
#our dataset for analysis has 1000 records and 16 columns
```

```r
## we check for the number of rows and columns
#---
#
cat("Rows:", nrow(df), "\nCols:", ncol(df))
```

```
## Rows: 1000
## Cols: 16
```

```r
##we check if datatypes are appropriate
#---
#
glimpse(df)
```

```
## Rows: 1,000
## Columns: 16
## $ `Invoice ID`              <chr> "750-67-8428", "226-31-3081", "631-41-3108",~
## $ Branch                    <chr> "A", "C", "A", "A", "A", "C", "A", "C", "A",~
## $ `Customer type`           <chr> "Member", "Normal", "Normal", "Member", "Nor~
## $ Gender                    <chr> "Female", "Female", "Male", "Male", "Male", ~
## $ `Product line`            <chr> "Health and beauty", "Electronic accessories~
## $ `Unit price`              <dbl> 74.69, 15.28, 46.33, 58.22, 86.31, 85.39, 68~
## $ Quantity                  <int> 7, 5, 7, 8, 7, 7, 6, 10, 2, 3, 4, 4, 5, 10, ~
## $ Tax                       <dbl> 26.1415, 3.8200, 16.2155, 23.2880, 30.2085, ~
## $ Date                      <chr> "1/5/2019", "3/8/2019", "3/3/2019", "1/27/20~
## $ Time                      <chr> "13:08", "10:29", "13:23", "20:33", "10:37",~
## $ Payment                   <chr> "Ewallet", "Cash", "Credit card", "Ewallet",~
## $ cogs                      <dbl> 522.83, 76.40, 324.31, 465.76, 604.17, 597.7~
## $ `gross margin percentage` <dbl> 4.761905, 4.761905, 4.761905, 4.761905, 4.76~
## $ `gross income`            <dbl> 26.1415, 3.8200, 16.2155, 23.2880, 30.2085, ~
## $ Rating                    <dbl> 9.1, 9.6, 7.4, 8.4, 5.3, 4.1, 5.8, 8.0, 7.2,~
## $ Total                     <dbl> 548.9715, 80.2200, 340.5255, 489.0480, 634.3~
```

```r
##we check for the number of columns
#---
#
length(df)
```

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

```
##we check the column names for easier reference
#---
#
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"
```

```
##we check for column data types
#---
#
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"
```

```
## we Check for unique characters
#---
#
sapply(df, function(x) length(unique(x)))
```

```
##              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
```

```
##we check 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>
```

4. Tidying the data

```
##we change the column names to lowercase for easier manipulation
#---
#
colnames(df) = tolower(colnames(df))
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"
```

```
##we replace spaces in column names for easier manipulation
#---

names(df) = str_replace_all(names(df), c(' ' = '_'))
names(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"
```

```
##we check for missing values
#---
#
sum(is.na(df))
```

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

```
#There are no missing values
```

```
##we Check the sum of missing values per column
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
```

```
## we check the column names containing missing observations
#---
#
list_na <- colnames(df)[ apply(df, 2, anyNA) ]
list_na
```

```
## character(0)
```

```
## we display rows which don't contain any missing values
#---
#
na.omit(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
##   ---
##  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
##    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
##    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
```

```
#we confirmed that our dataset has no missing values
```

```
## we check 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
```

```
#Our data for analysis has no duplicates
```

```
##we check for unique items
#---
#
unique_items <- df[!duplicated(df), ]
unique_items
```

```
##        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
```

```
##   ---
##  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
##    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
##    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
```

```
##we select numeric columns
#---
#
df1<- df %>% select_if(is.numeric)

#preview the column names
colnames(df1)
```

```
## [1] "unit_price"             "quantity"
## [3] "tax"                    "cogs"
## [5] "gross_margin_percentage" "gross_income"
## [7] "rating"                 "total"
```

```
plot_str(df)
```

```
## we select needed columns
#---
#
df2 <- subset(df1, select = c("unit_price", "quantity", "tax", "cogs", "gross_income", "rating", "total
#preview the column names
colnames(df2)
```

```
## [1] "unit_price"   "quantity"     "tax"          "cogs"         "gross_income"
## [6] "rating"       "total"
```

5. Explaratory Data Analysis

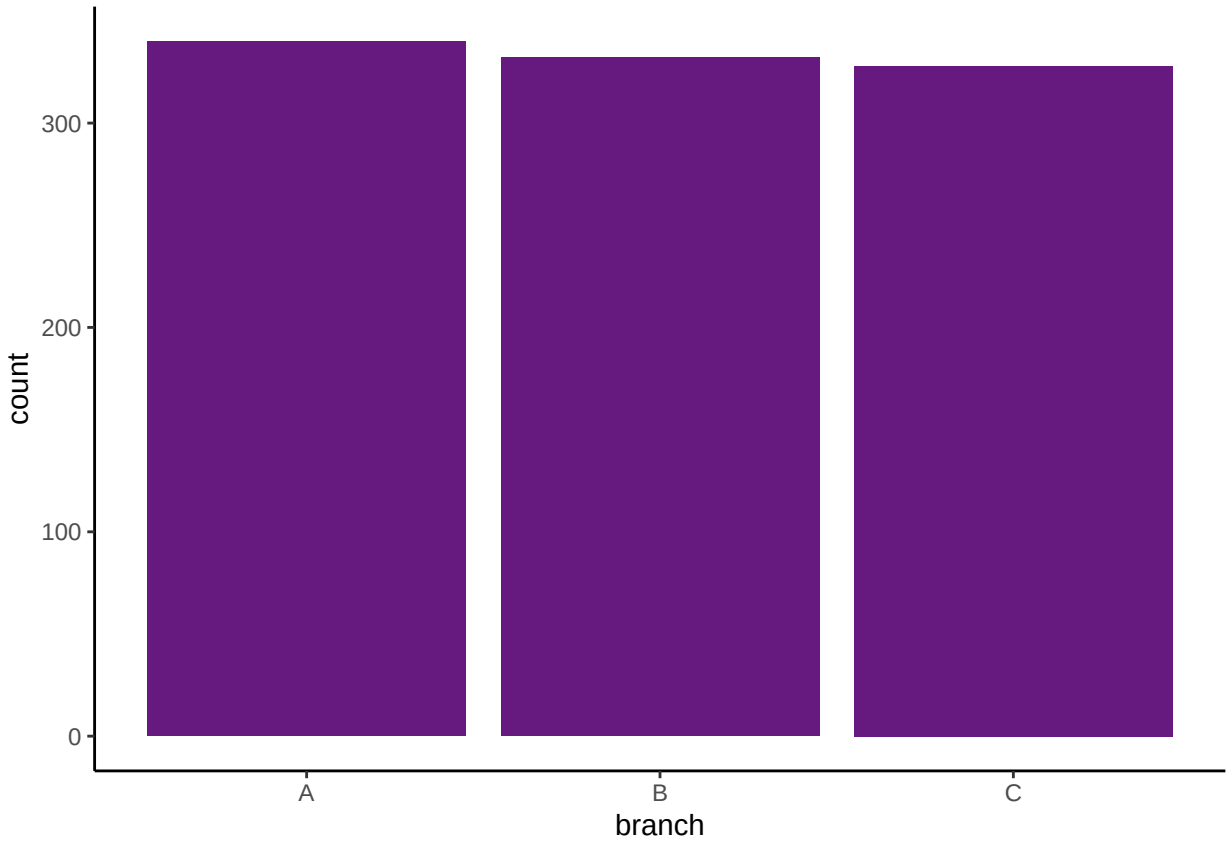
5.1 Univariate Analysis

```
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
```

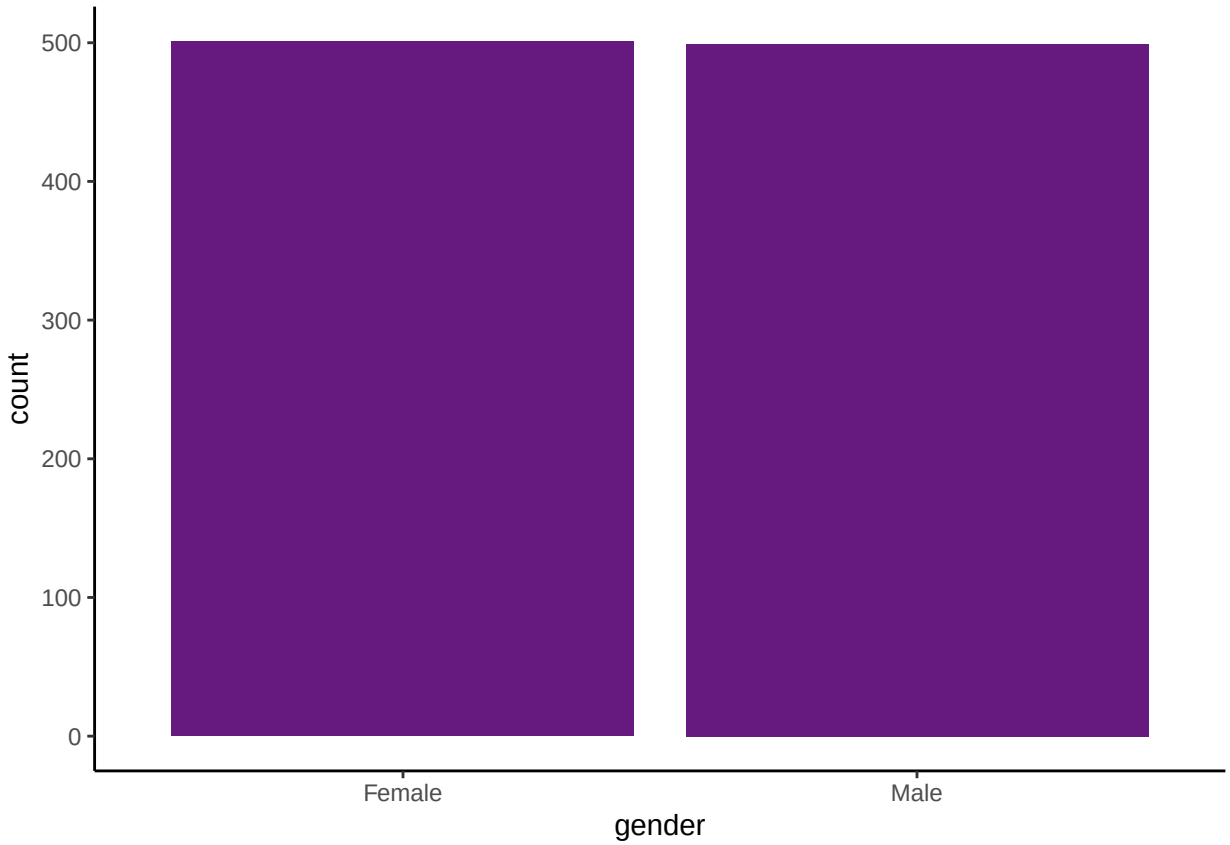
Countplots for the Categorical variables_

```
##we plot the countplot for the variable branch
#---
#
ggplot(df, aes(x=branch)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```



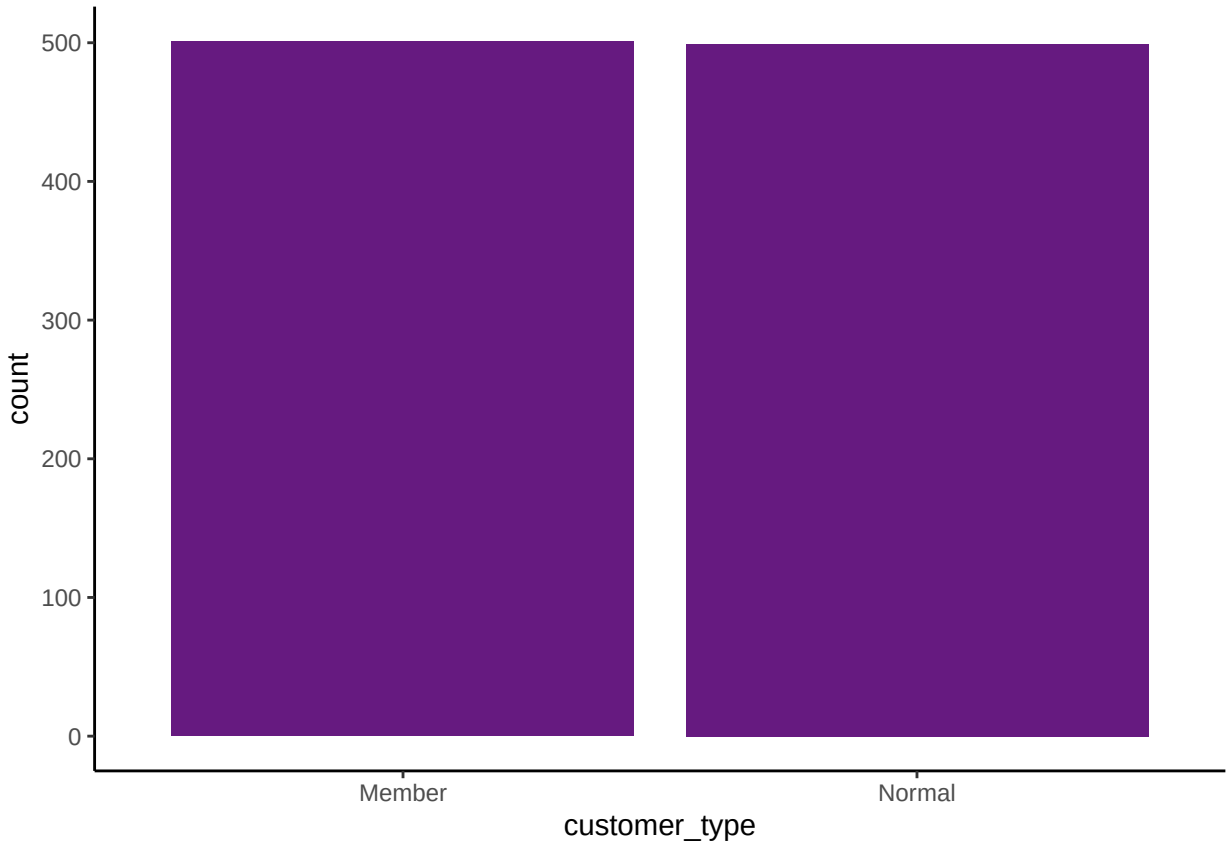

```
#From the plotted countplots Branch A has the highest number followed by B and then C.
```

```
##we plot the countplot for the variable gender
#---
#
ggplot(df, aes(x=gender)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```



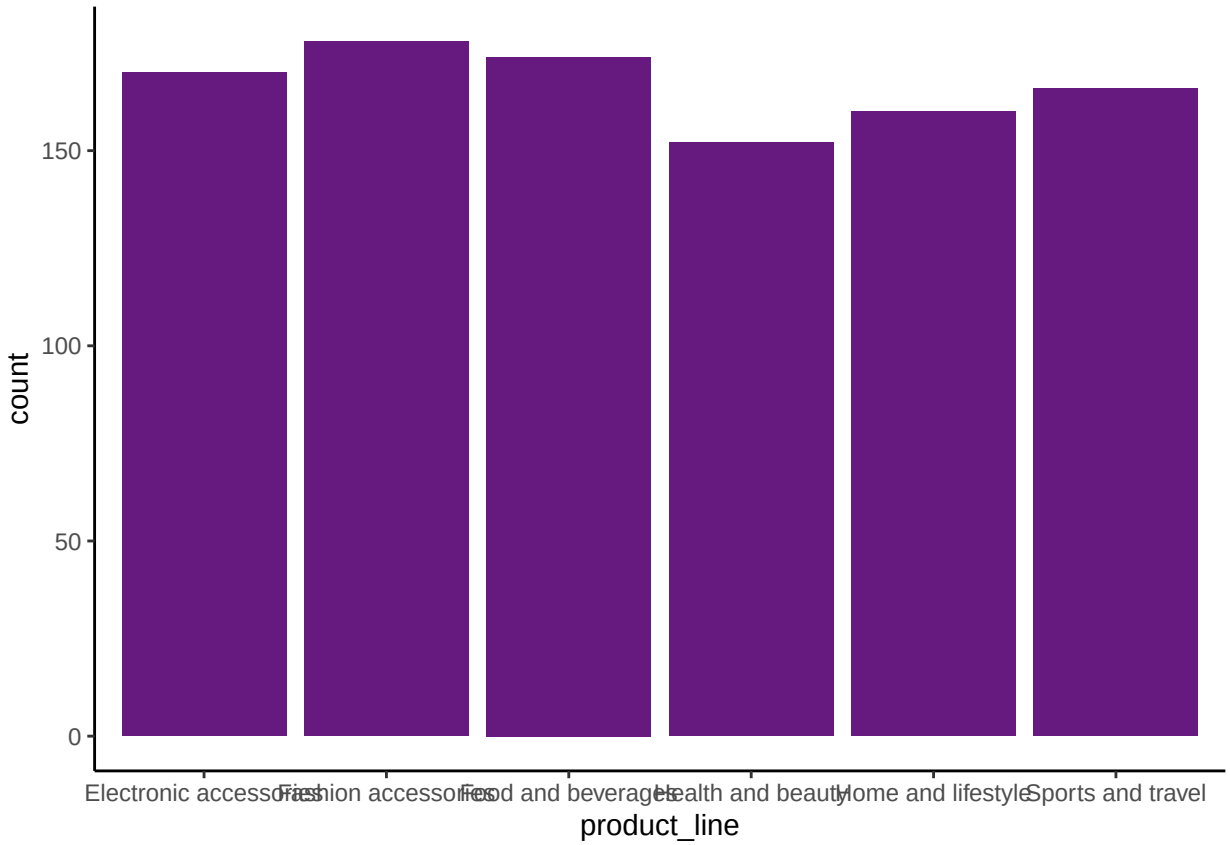

```
#From the plotted countplots the number of female is equal that of male.
```

```
##we plot the countplot for the variable customer type
#---
#
ggplot(df, aes(x=customer_type)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```



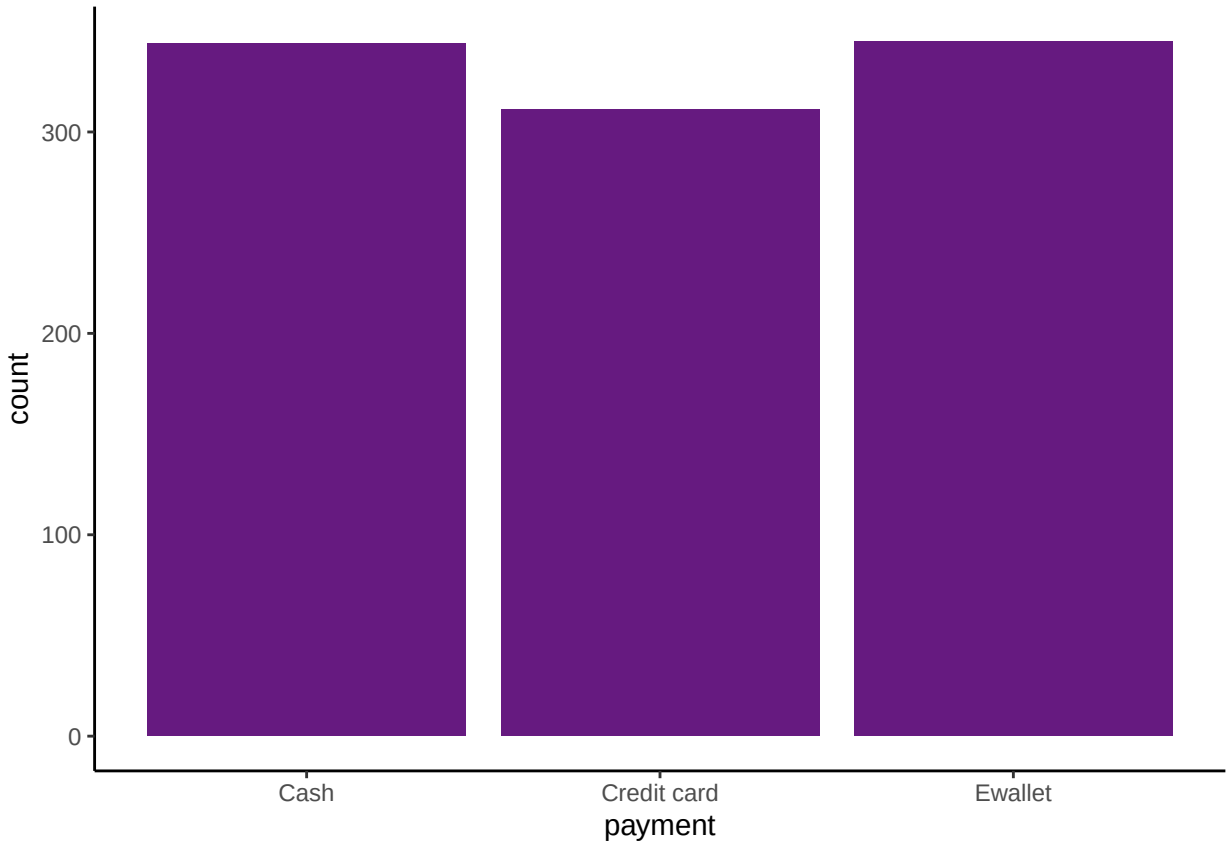

```
#From the plotted countplots the number of member customer is slightly higher than that of normal custo
```

```
##we plot the countplot for the product line
#---
#
ggplot(df, aes(x=product_line)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```




```
#From the plotted countplots the Fashion accessories has a higher number of sales as compared to the ot
```

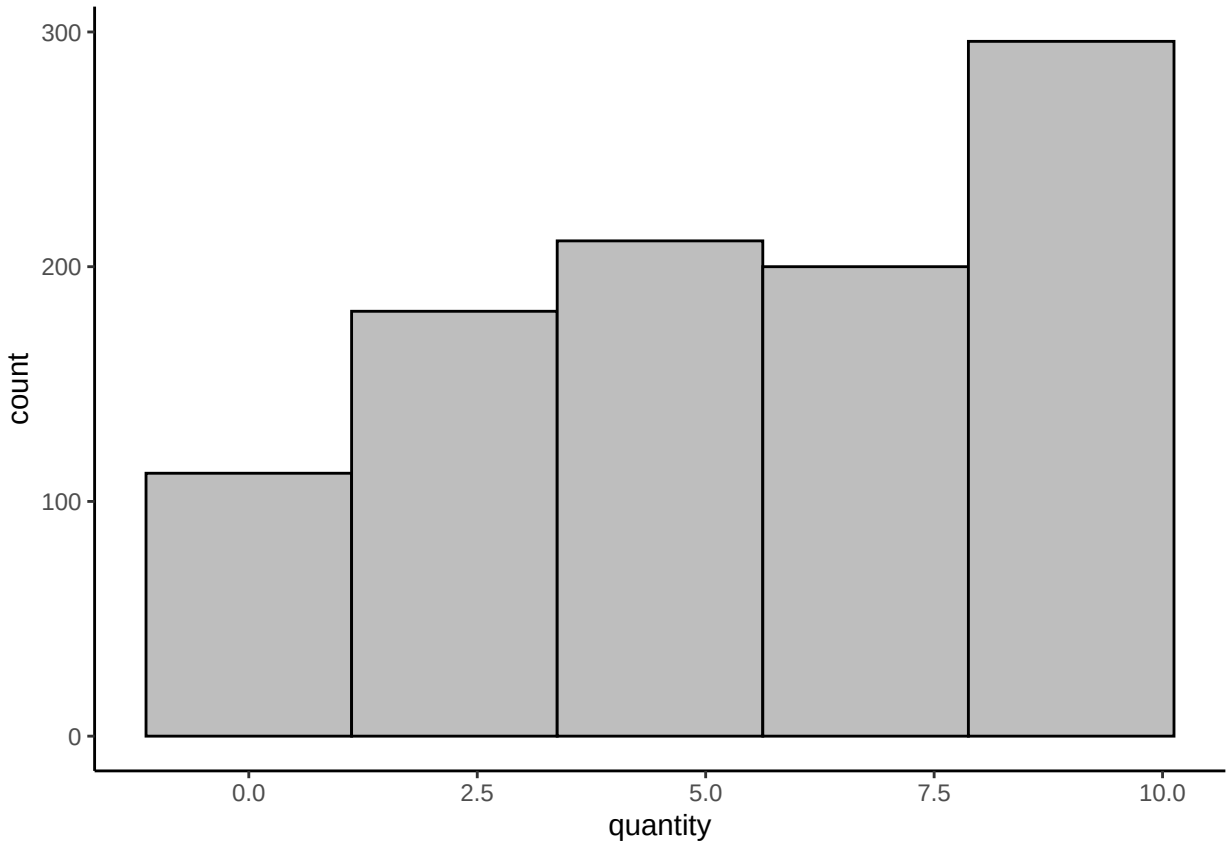
```
##we plot the countplot for the payment
#---
#
ggplot(df, aes(x=payment)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```



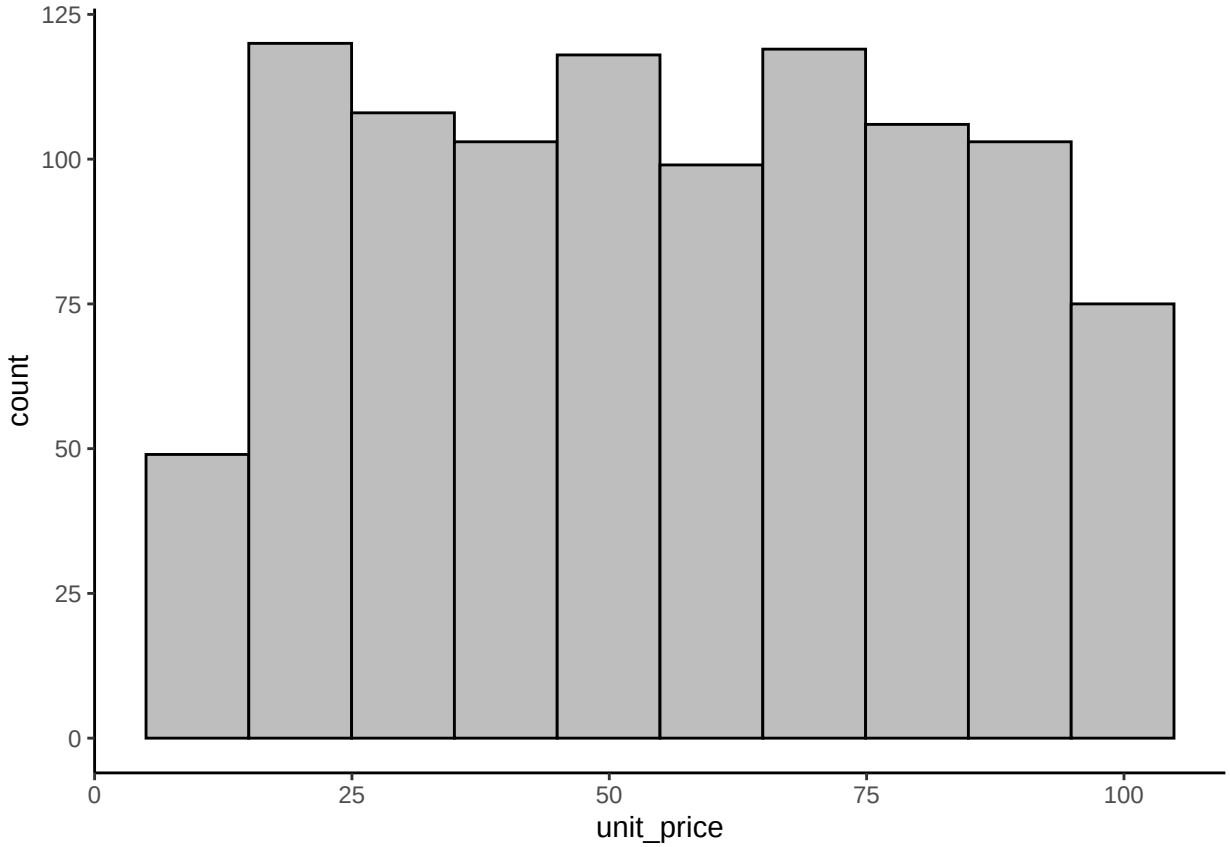

```
#From the plotted countplots the Ewallet payment mode had the highest number followed by cash and credi
```

Numerical Variables

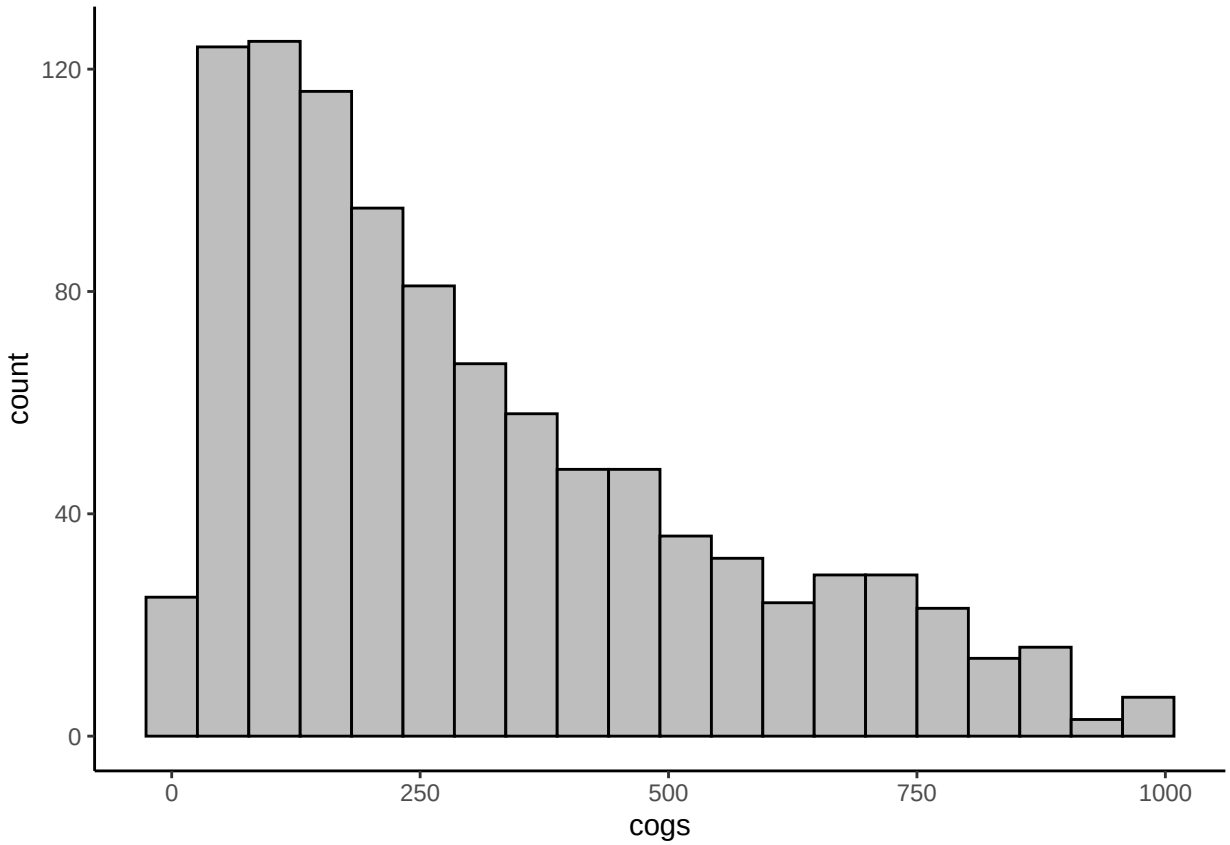
```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`quantity`)) +
 geom_histogram(colour="black", fill="grey",bins=5)
```



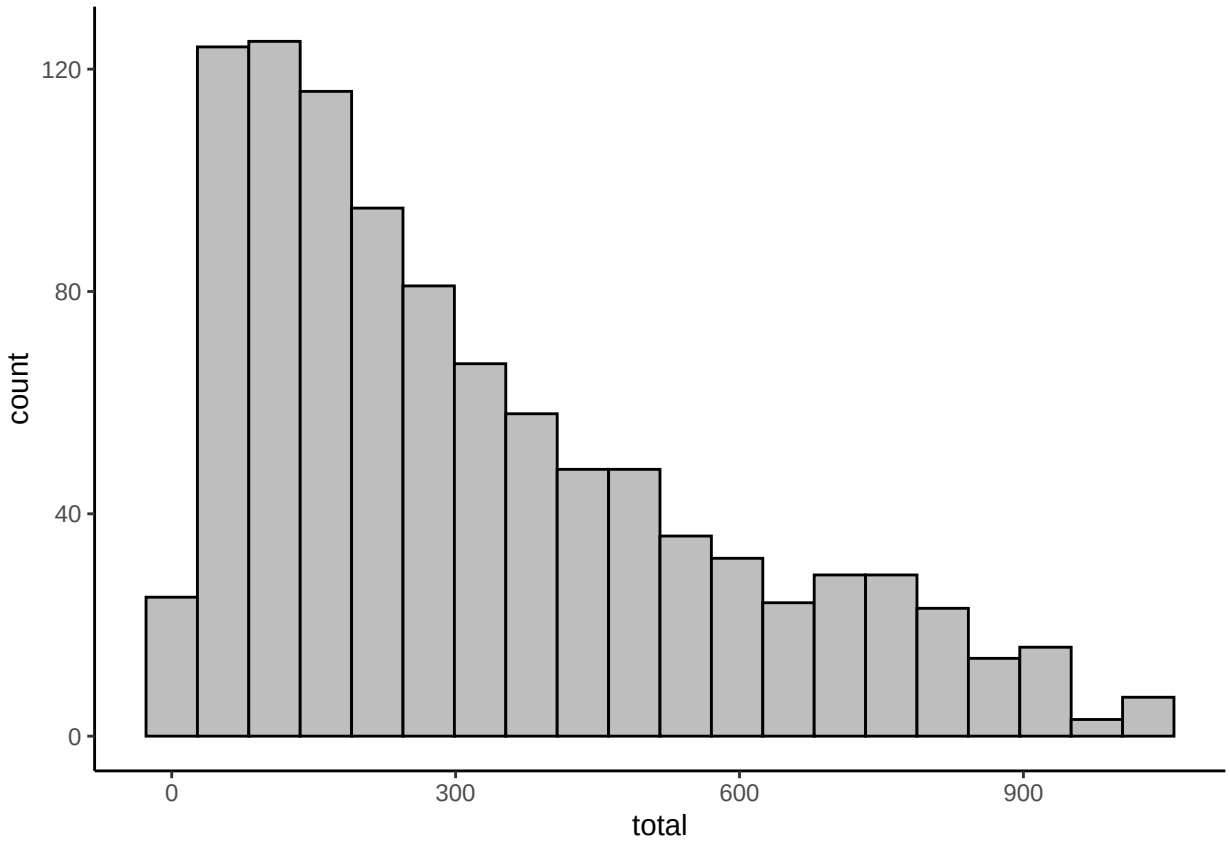

```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`unit_price`)) +
 geom_histogram(colour="black", fill="grey",bins=10)
```



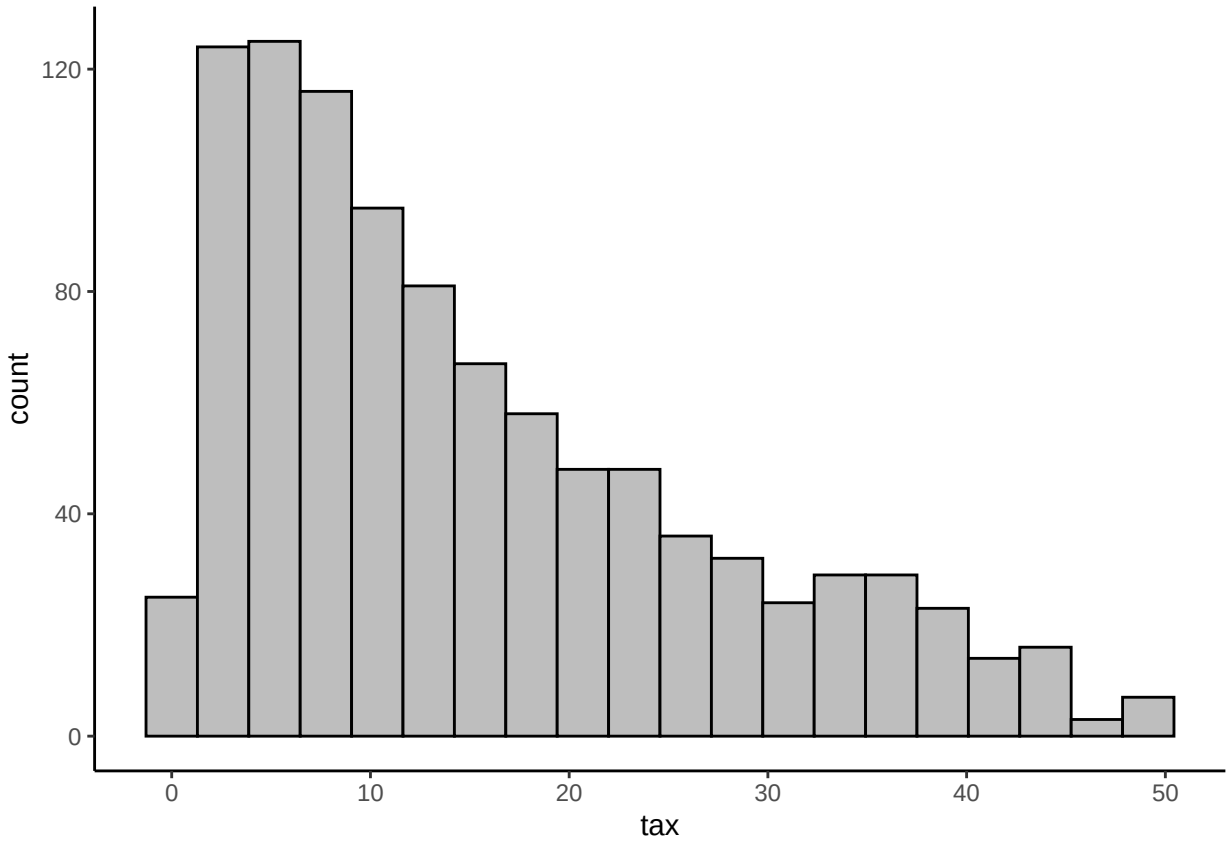

```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`cogs`)) +
 geom_histogram(colour="black", fill="grey",bins=20)
```



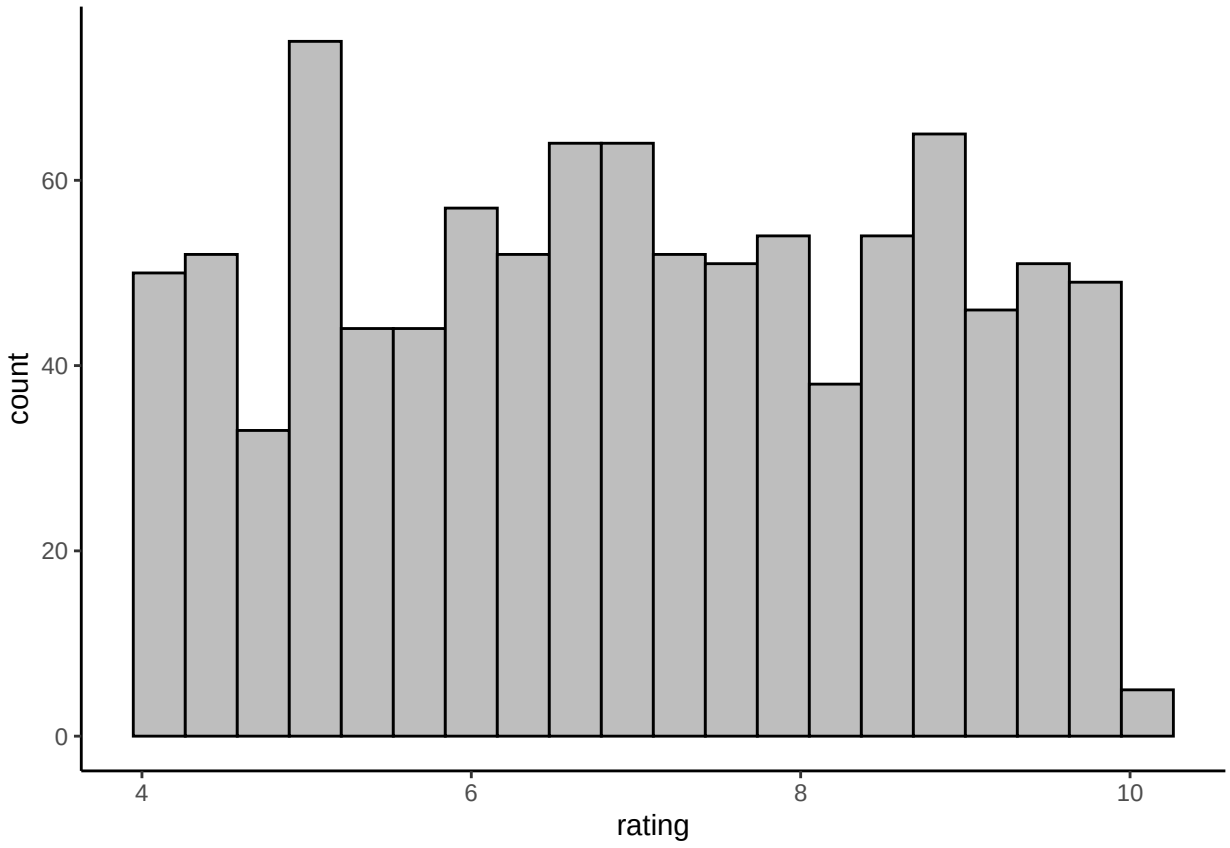

```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`total`)) +
 geom_histogram(colour="black", fill="grey",bins=20)
```




```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`tax`)) +
 geom_histogram(colour="black", fill="grey",bins=20)
```




```
##Histogram with density plot
#---
#
ggplot(df, aes(x=`rating`)) +
 geom_histogram(colour="black", fill="grey",bins=20)
```

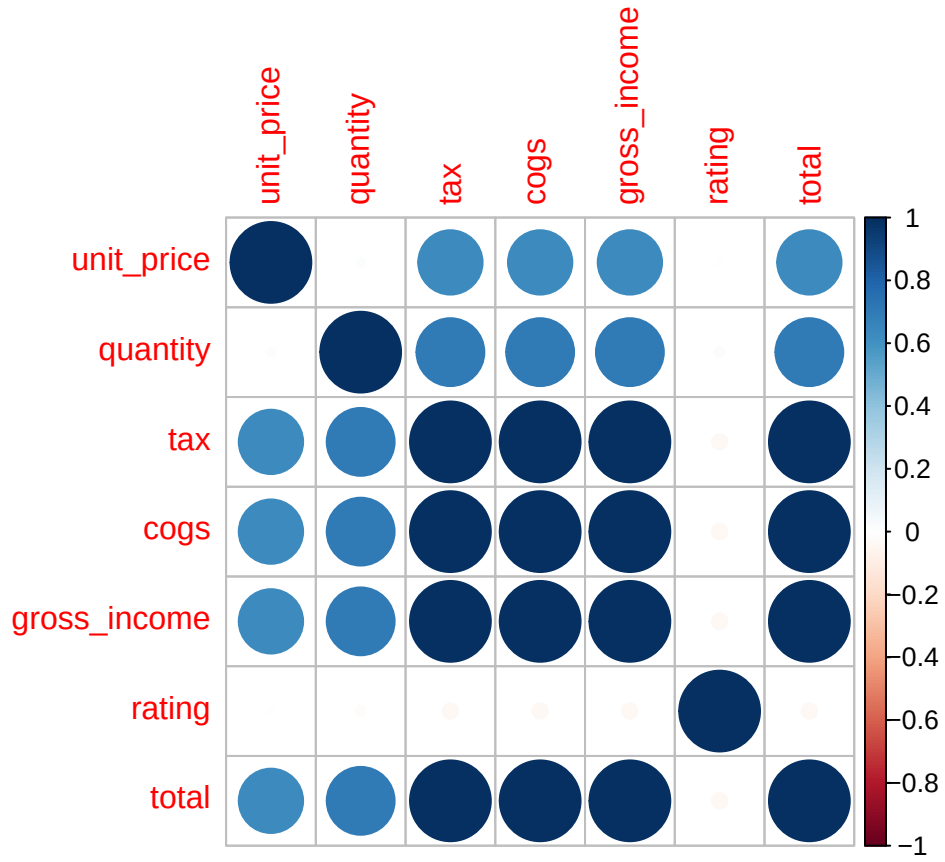


5.2 Bivariate Analysis

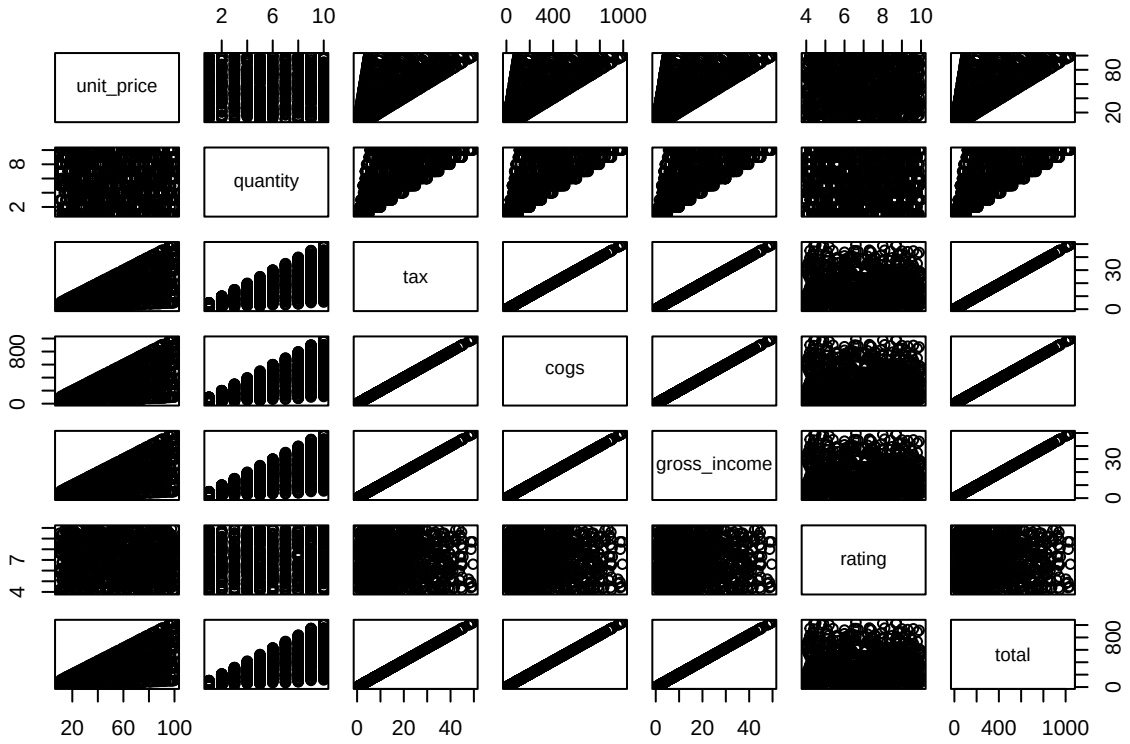
```
##we check the correlation
#---
#
# calculate correlations
correlations <- cor(df2[,1:7])
correlations
```

```
##                unit_price    quantity        tax       cogs gross_income
## unit_price    1.000000000  0.01077756  0.6339621  0.6339621    0.6339621
## quantity      0.010777564  1.00000000  0.7055102  0.7055102    0.7055102
## tax           0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## cogs          0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## gross_income  0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
## rating       -0.008777507 -0.01581490 -0.0364417 -0.0364417   -0.0364417
## total         0.633962089  0.70551019  1.0000000  1.0000000    1.0000000
##                     rating      total
## unit_price   -0.008777507  0.6339621
## quantity     -0.015814905  0.7055102
## tax          -0.036441705  1.0000000
## cogs         -0.036441705  1.0000000
## gross_income -0.036441705  1.0000000
## rating        1.000000000 -0.0364417
## total        -0.036441705  1.0000000
```

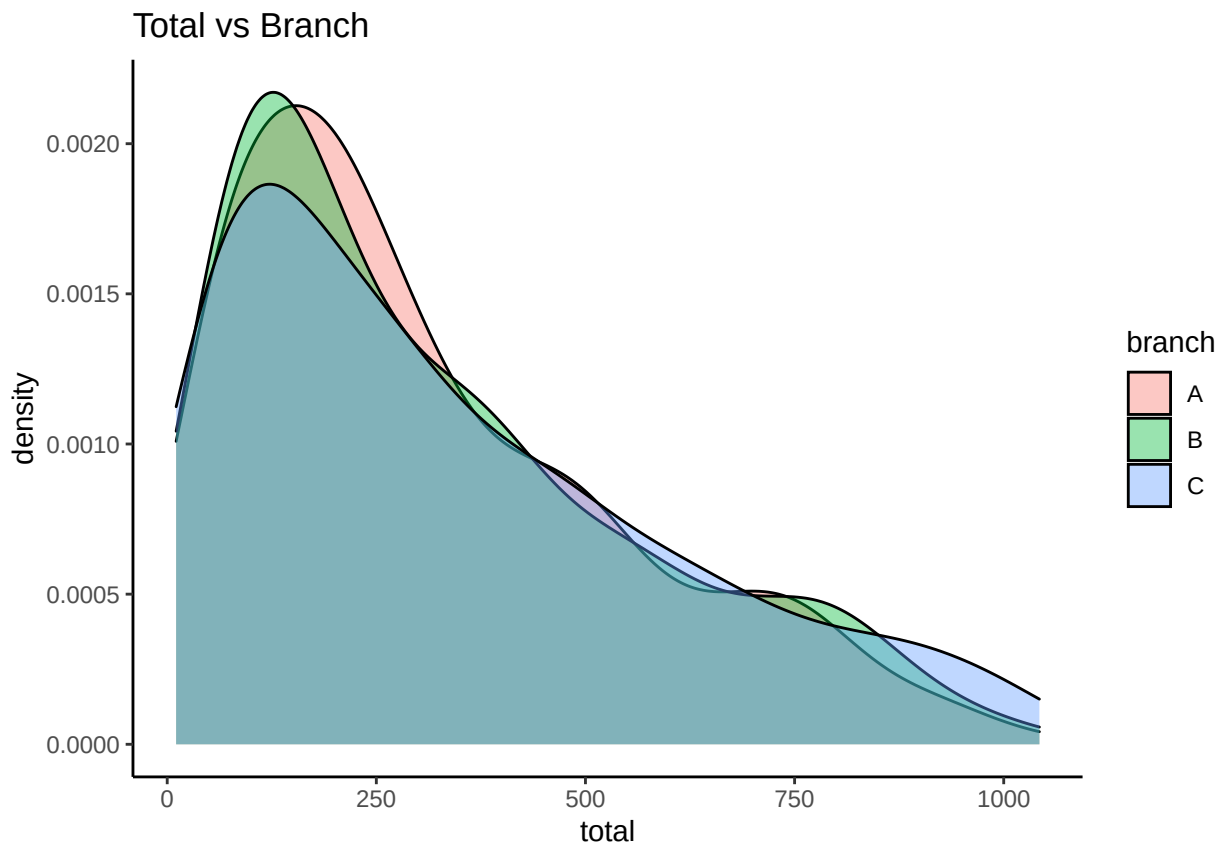
```
## create correlation plot
#---
corrplot(correlations, method="circle")
```



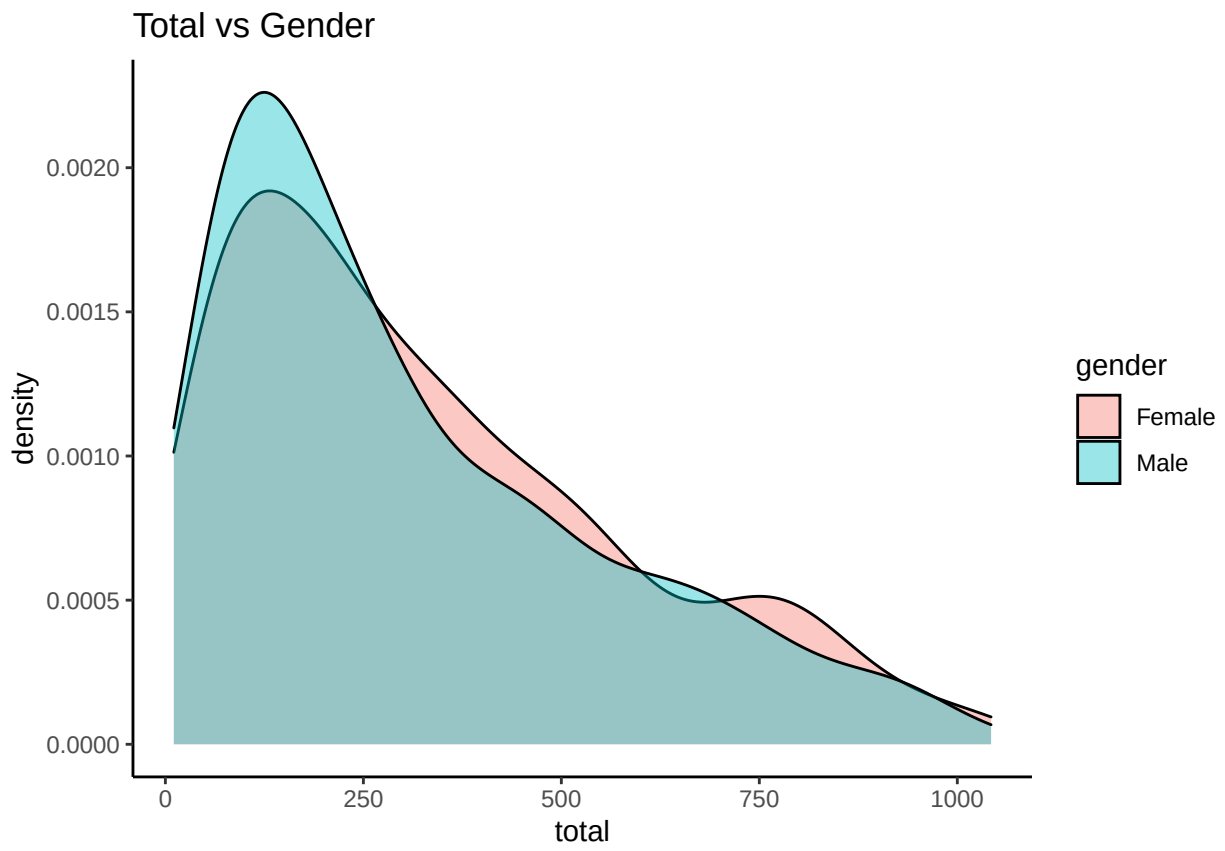

```
##we plot a pair plot
#---
#
pairs(df2[,1:7])
```



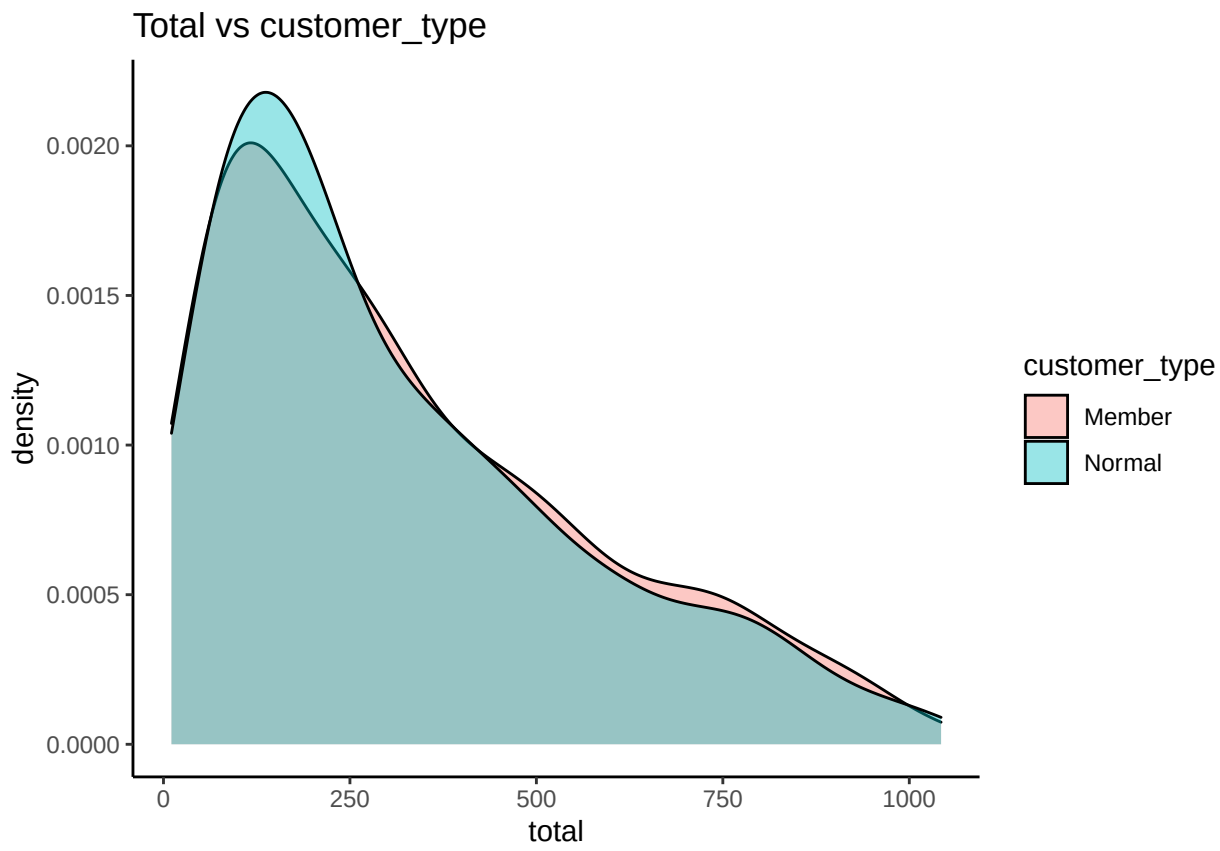

```
##we plot the stacked bar chart for total against branch
#---
#
ggplot(df, aes(x = total,  fill = branch)) +geom_density(alpha = 0.4) +
  labs(title = "Total vs Branch")
```



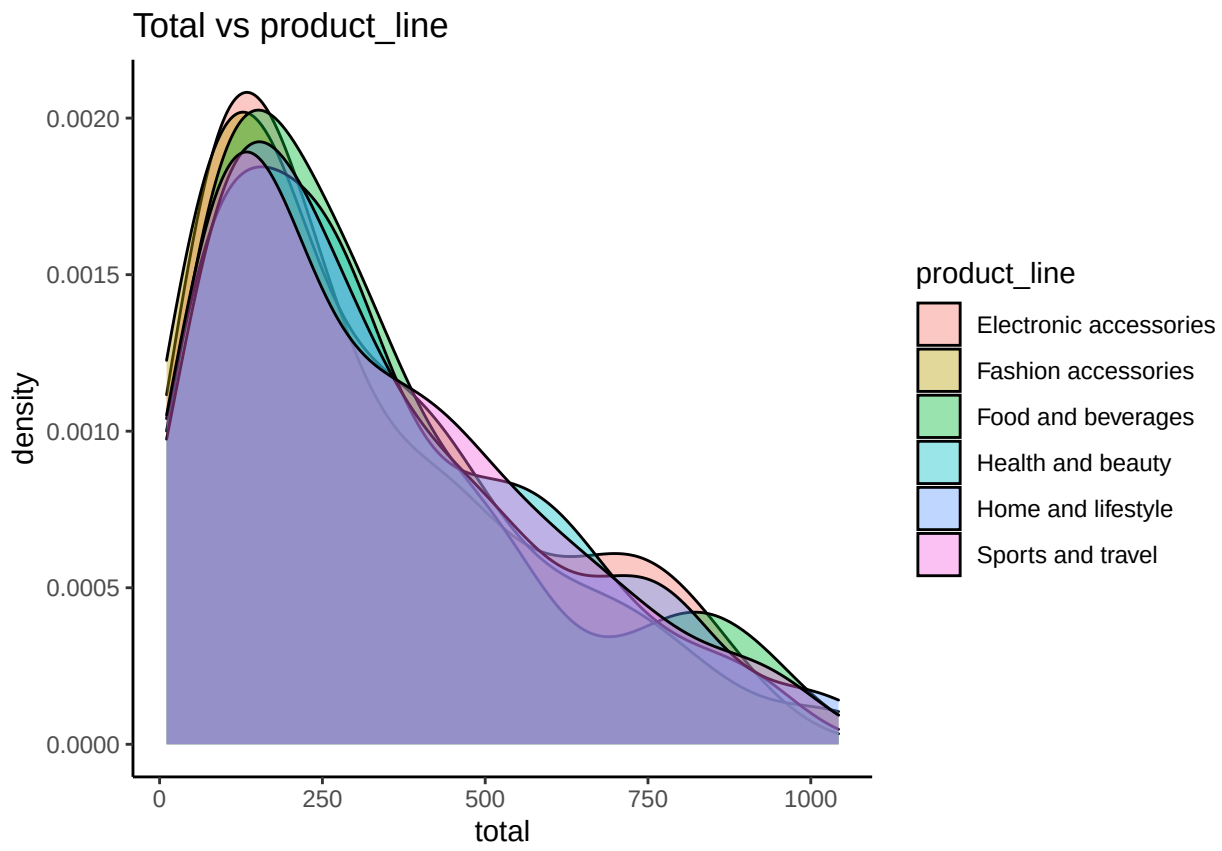

```
##we plot the stacked bar chart for total against gender
#---
#
ggplot(df, aes(x = total,  fill = gender)) +geom_density(alpha = 0.4) +
  labs(title = "Total vs Gender")
```



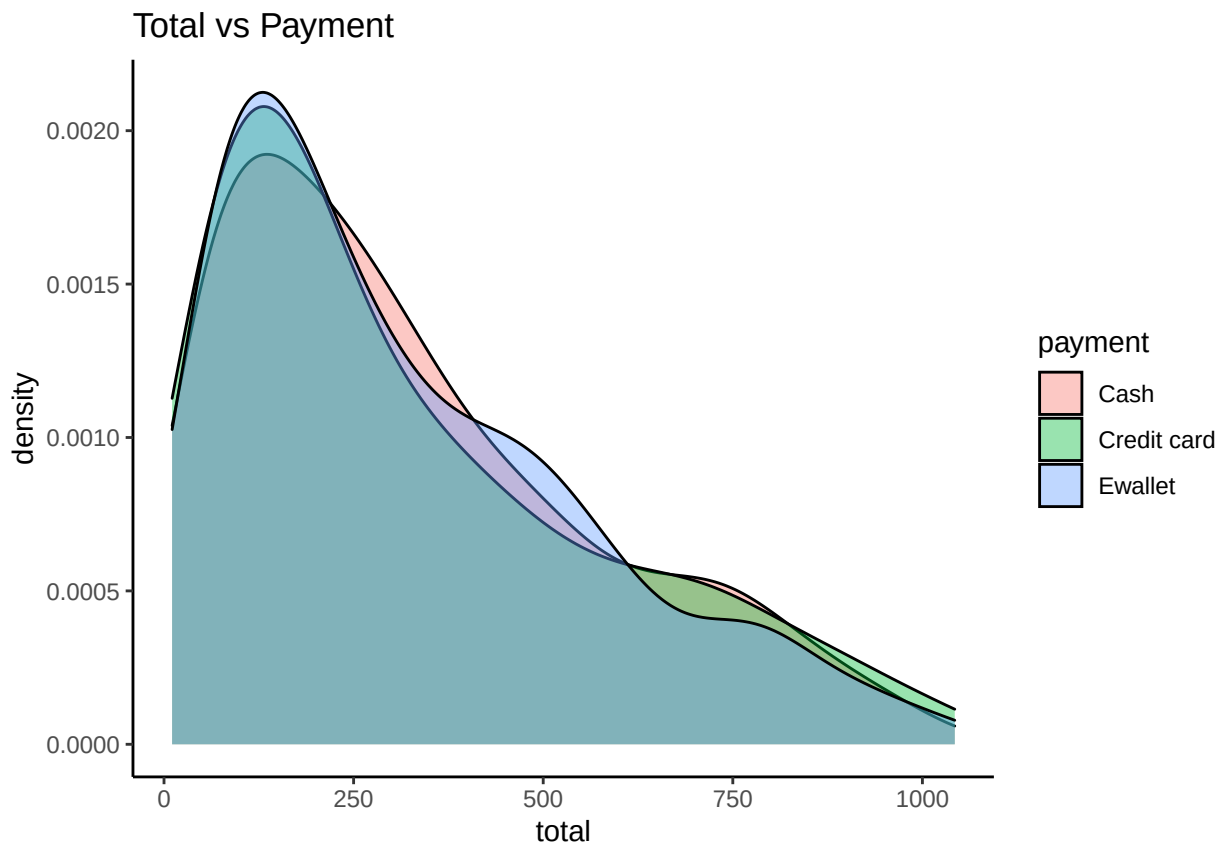

```
##we plot the stacked bar chart for total against customer type
#---
#
ggplot(df, aes(x = total,  fill = customer_type)) +geom_density(alpha = 0.4) +
  labs(title = "Total vs customer_type")
```




```
##we plot the stacked bar chart for total against product_line
#---
#
ggplot(df, aes(x = total,  fill = product_line)) +geom_density(alpha = 0.4) +
  labs(title = "Total vs product_line")
```




```
##we plot the stacked bar chart for total against payment
#---
#
ggplot(df, aes(x = total,  fill = payment)) +geom_density(alpha = 0.4) +
  labs(title = "Total vs Payment")
```



6. Dimensionality Reduction with PCA

```
##we check the structure of our data
#---
#
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>
```

```
##Standardize the data by using scale and apply "prcomp" function
#---
#
df3=prcomp(df2)#,center=T,scale.=T), center = TRUE, scale. = TRUE)
#preview
summary(df3)
```

```
## Importance of components:
```

```
##                             PC1      PC2     PC3     PC4       PC5       PC6
## Standard deviation     340.3819 20.53212 1.71932 1.24589 1.678e-13 7.548e-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     1.78e-15
## Proportion of Variance 0.00e+00
## Cumulative Proportion  1.00e+00
```

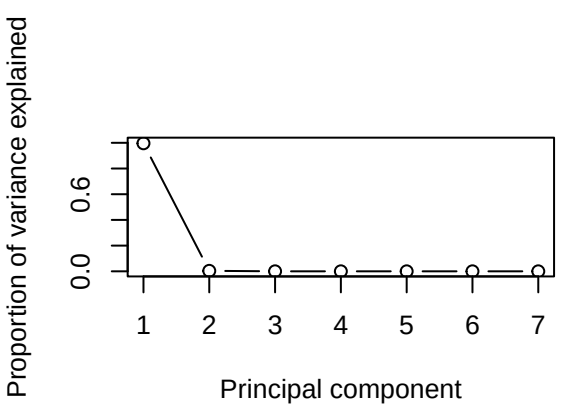
Conclusion

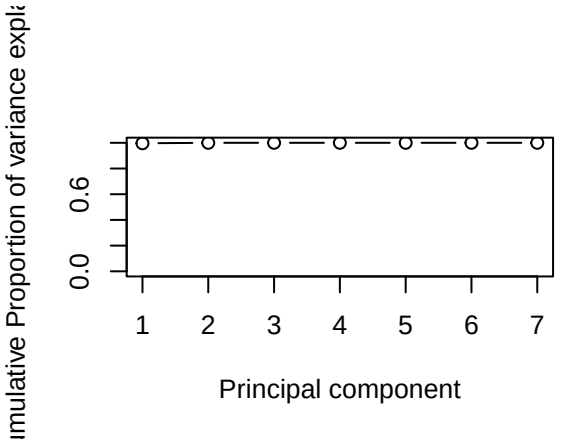
From the the summary, we can undersand PC1 explains 99.63% of variance and PC2 explains 0.363% and so on. We choose the principal component which explains the highest variance which usually explains about 95% variance can be considered for models.

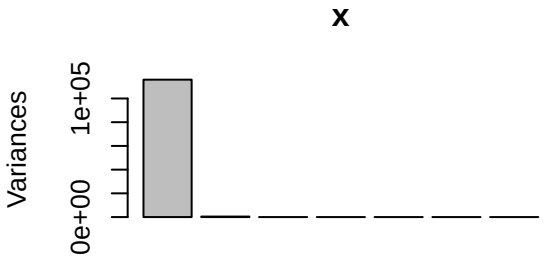
```
##we define a function which  is required to display all PCA related plots in 2X2 grid.
pcaCharts <- function(x) {
    x.var <- x$sdev ^ 2
    x.pvar <- x.var/sum(x.var)
    print("proportions of variance:")
    print(x.pvar)

    par(mfrow=c(2,2))
    plot(x.pvar,xlab="Principal component", ylab="Proportion of variance explained", ylim=c(0,1), type=
    plot(cumsum(x.pvar),xlab="Principal component", ylab="Cumulative Proportion of variance explained",
    screeplot(x)
    screeplot(x,type="l")
    par(mfrow=c(1,1))
}
```

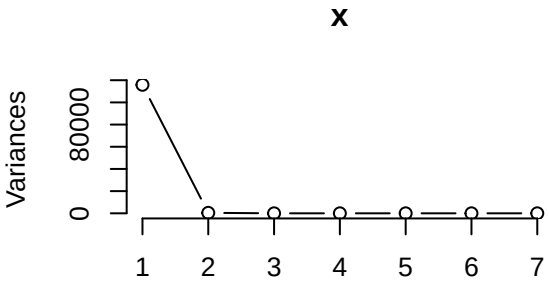
```
##visualization of the data in the new reduced dimension
#---
#
pcaCharts(df3)
```

```
## [1] "proportions of variance:"
## [1] 9.963360e-01 3.625270e-03 2.542068e-05 1.334852e-05 2.422713e-31
## [6] 4.899401e-34 2.725707e-35
```

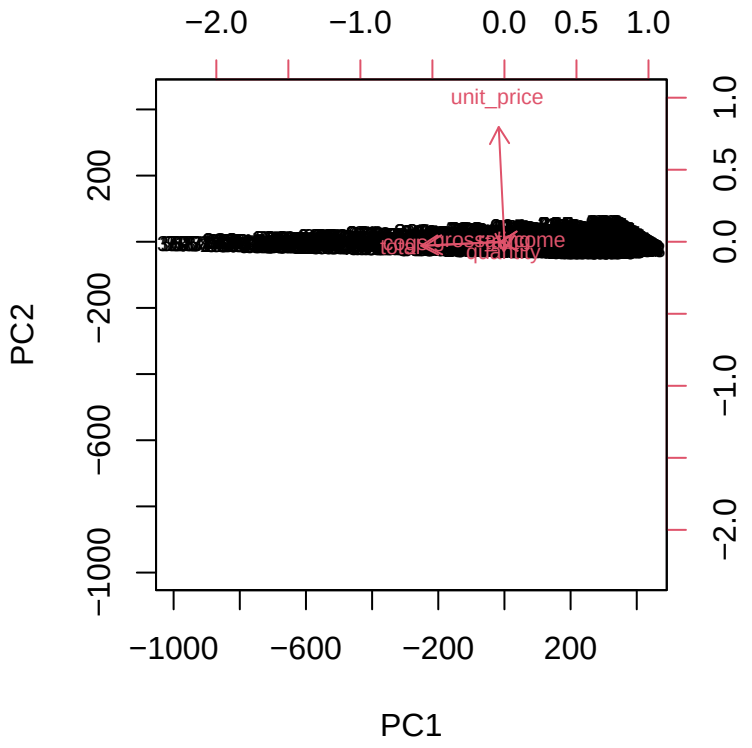




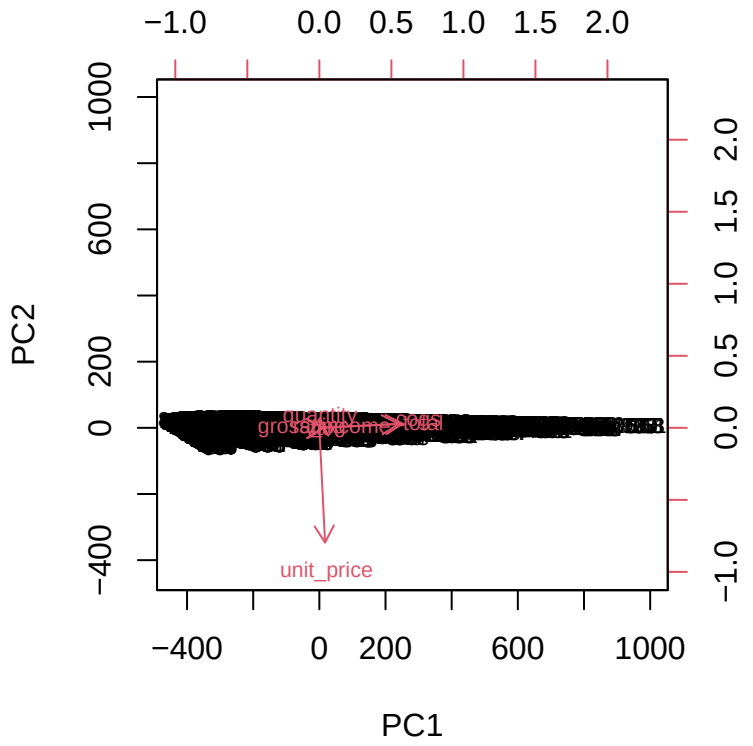





```
biplot(df3,scale=0, cex=.7)
```




```
pca.out <- df3
pca.out$rotation <- -pca.out$rotation
pca.out$x <- -pca.out$x
biplot(pca.out,scale=0, cex=.7)
```




```
pca.out$rotation[,1:2]
```

```
##                       PC1          PC2
## unit_price    0.0495237905 -0.995517376
## quantity      0.0060451467  0.080957901
## tax           0.0343989368  0.001683275
## cogs          0.6879787359  0.033665491
## gross_income  0.0343989368  0.001683275
## rating       -0.0001837158 -0.001550616
## total         0.7223776726  0.035348766
```

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

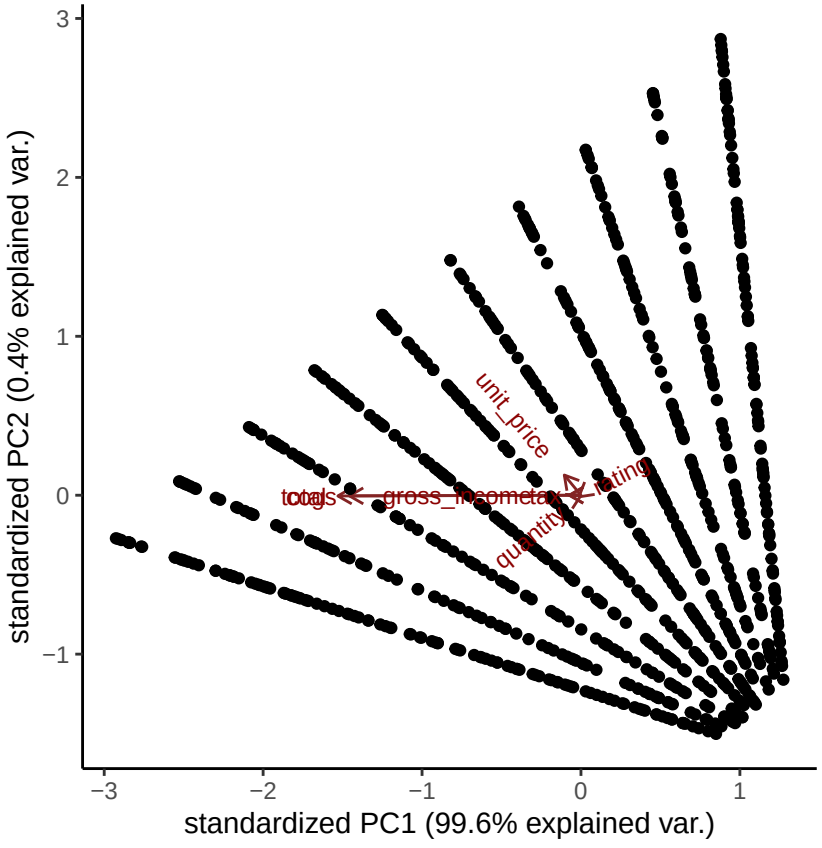
```
## List of 5
##  $ sdev    : num [1:7] 3.40e+02 2.05e+01 1.72 1.25 1.68e-13 ...
##  $ rotation: num [1:7, 1:7] -0.04952 -0.00605 -0.0344 -0.68798 -0.0344 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "unit_price" "quantity" "tax" "cogs" ...
##   .. ..$ : chr [1:7] "PC1" "PC2" "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"
```

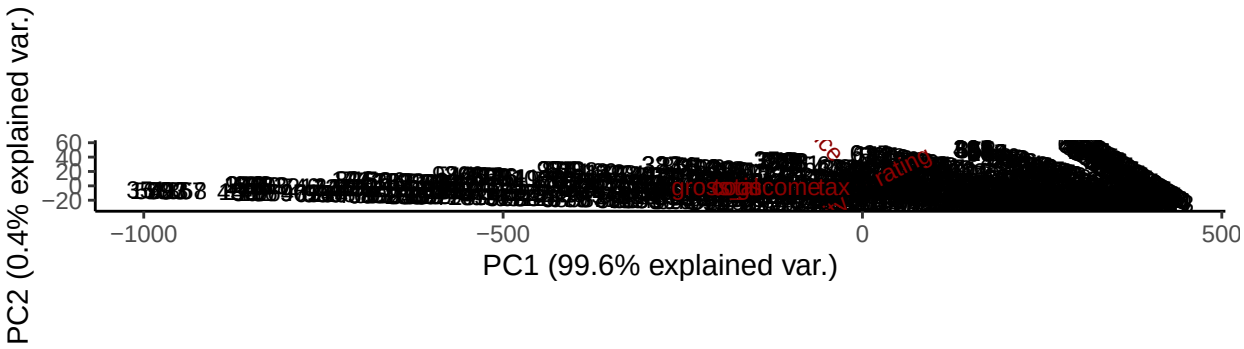
```
##Installing our ggbiplot visualisation package
#---
#
library(devtools)
```

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

```
Sys.setenv(R_REMOTES_NO_ERRORS_FROM_WARNINGS="true")
#install_github("vqv/ggbiplot",force=TRUE)
library(ggbiplot)
ggbiplot(df3)
```




```
## we add more detail to the plot, we provide arguments rownames as labels
#---
#
ggbiplot(df3, labels=rownames(df), obs.scale = 1, var.scale = 1)
```


PC2 (0.4% explained var.)
60
40
20
0
−20
−1000
−500
0
500
PC1 (99.6% explained var.)
rating
grossbudgetincometax