Dimensionality Reduction and Feature Selection

Cynthia Mwadime

2022-10-06

Dimensionality Reduction and Feature Selection

```r
# Loading Libraries
library(e1071)
library(Rtsne)
library(ggplot2)
library(CatEncoders)
```

```
##
## Attaching package: 'CatEncoders'

## The following object is masked from 'package:base':
##
##     transform
```

```r
library(lattice)
library(caret)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(readr)
library(ROCR)
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following objects are masked from 'package:e1071':
##
##     kurtosis, skewness

## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(ggcorrplot)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(rpart)
library(caTools)
library(class)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-4
```

```r
library(Hmisc)
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

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

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

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

## The following object is masked from 'package:e1071':
##
##     impute

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

```r
library(funModeling)
```

```
## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(klaR)
library(scales)
```

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

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

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

```r
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(DataExplorer)
library(ClustOfVar)
library(GGally)
```

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

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:funModeling':
##
##     range01
```

Reading the data

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

```
##    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
```

Investigating the structure

```
#getting the datatypes and dimentions
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                 : 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 ...
```

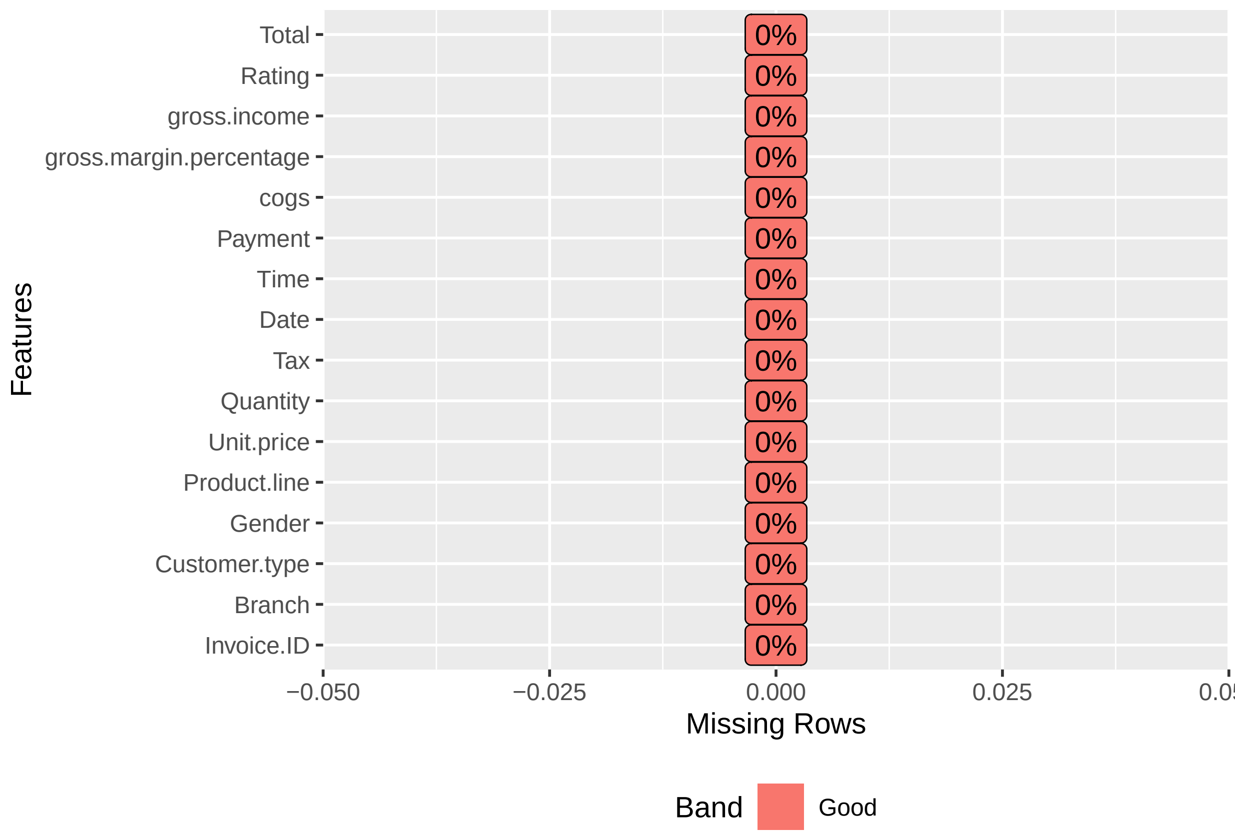
Data Cleaning

```
# checking for duplicates in the data
data[duplicated(data), ]
```

```
##  [1] Invoice.ID              Branch                  Customer.type
##  [4] Gender                  Product.line            Unit.price
##  [7] Quantity                Tax                     Date
## [10] Time                    Payment                 cogs
## [13] gross.margin.percentage gross.income            Rating
## [16] Total
## <0 rows> (or 0-length row.names)
```

- No duplicates found

```
# checking for missing values
plot_missing(data)
```



* No mising values

```r
head(data)
```

```
##    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
# removing the Invoice id column
data$Invoice.ID <- NULL
# fixing the data types
data$Branch <- as.factor(data$Branch)
data$Customer.type <- as.factor(data$Customer.type)
data$Gender <- as.factor(data$Gender)
data$Product.line <- as.factor(data$Product.line)
data$Payment <- as.factor(data$Payment)
data$Date <- as.Date(data$Date, format = "%m/%d/%y")
```

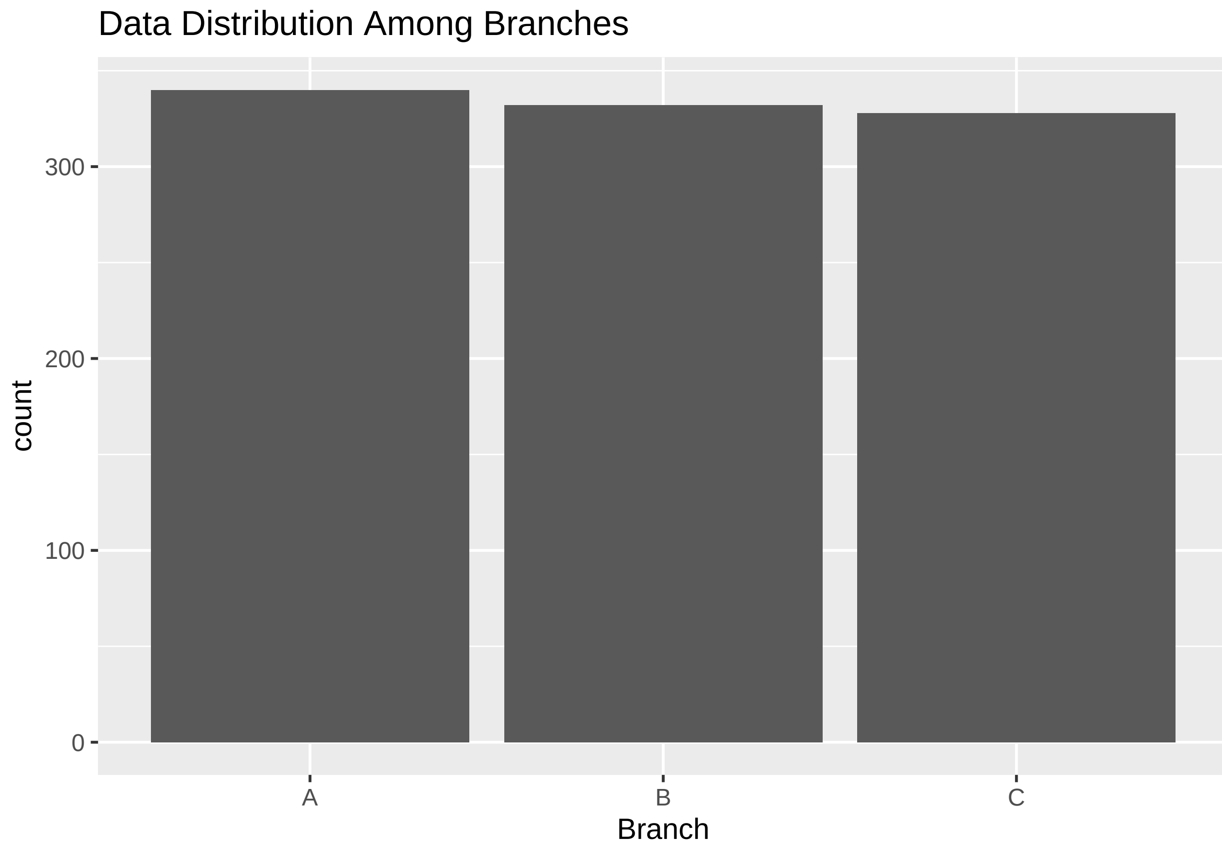
Exploaratory Data Analysis

Univariate Analysis

```r
# creating a mode function
mode <- function(x){
  uniqx <- unique(x)
  uniqx[which.max(tabulate(match(x, uniqx)))]
}
```

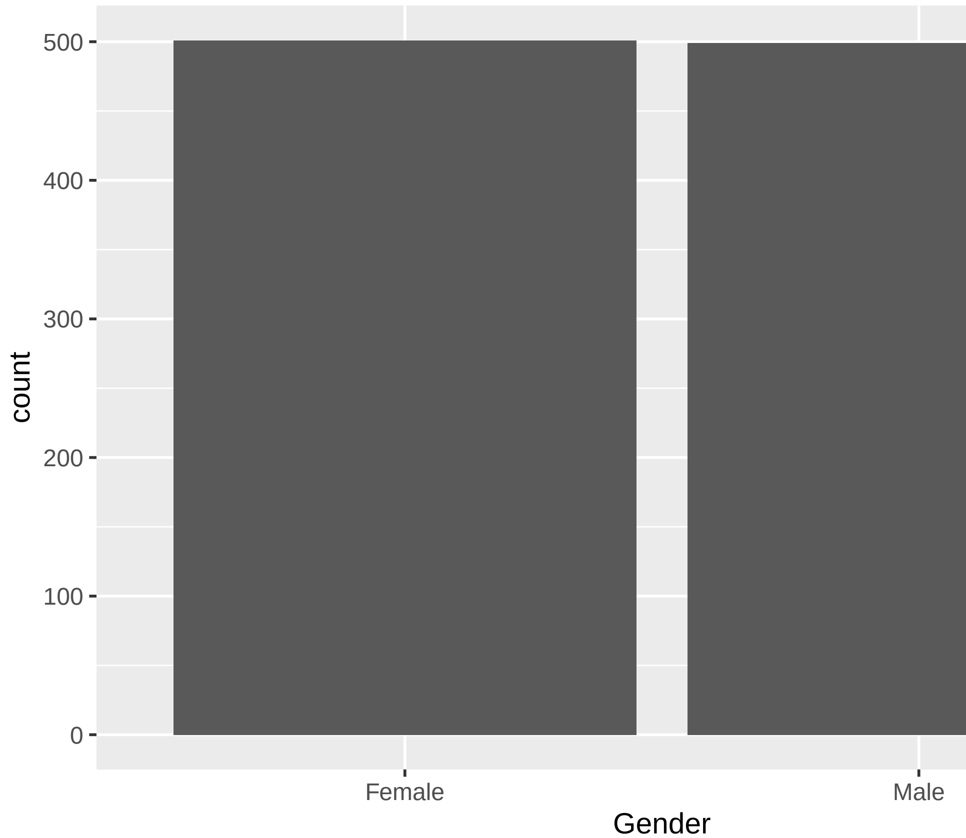
Branch Visualization Investigating how much data was contributed by each branch and coparing them

```r
ggplot(data, aes(Branch)) + geom_bar(stat="count") + labs(title="Data Distribution Among Branches ")
```



- Data was provided almost equaly by all branches

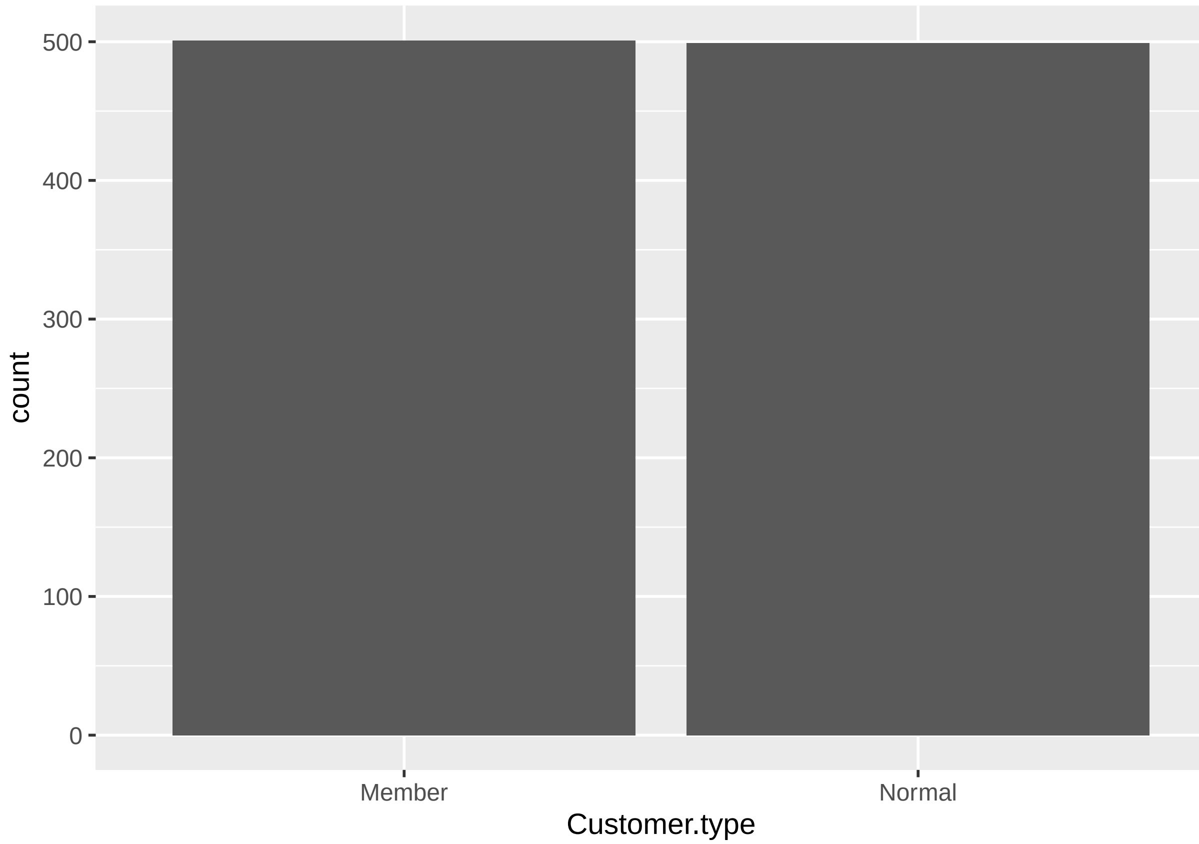
```
ggplot(data, aes(Gender)) + geom_bar(stat="count")
```



Investigating Gender distribution

The gender distribution in the dataset is balanced.

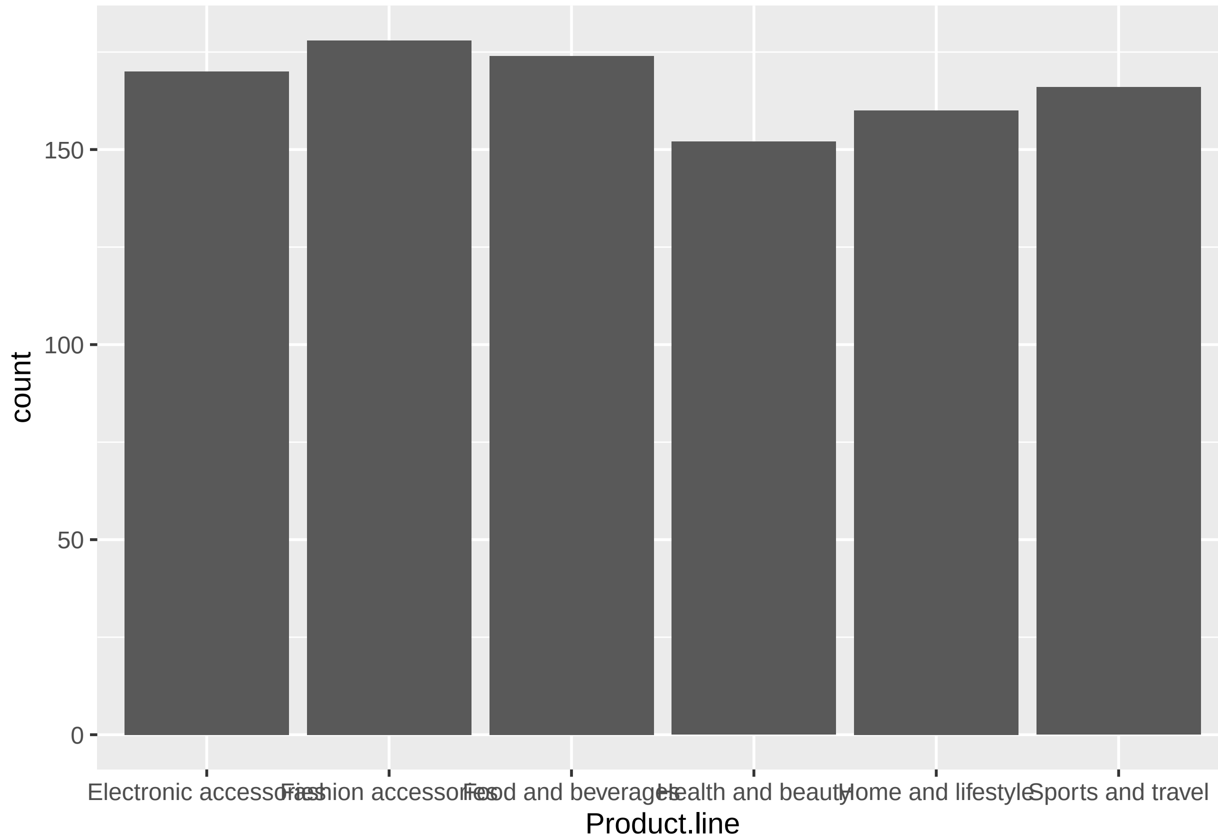
```
ggplot(data, aes(Customer.type)) + geom_bar()
```



Customer type

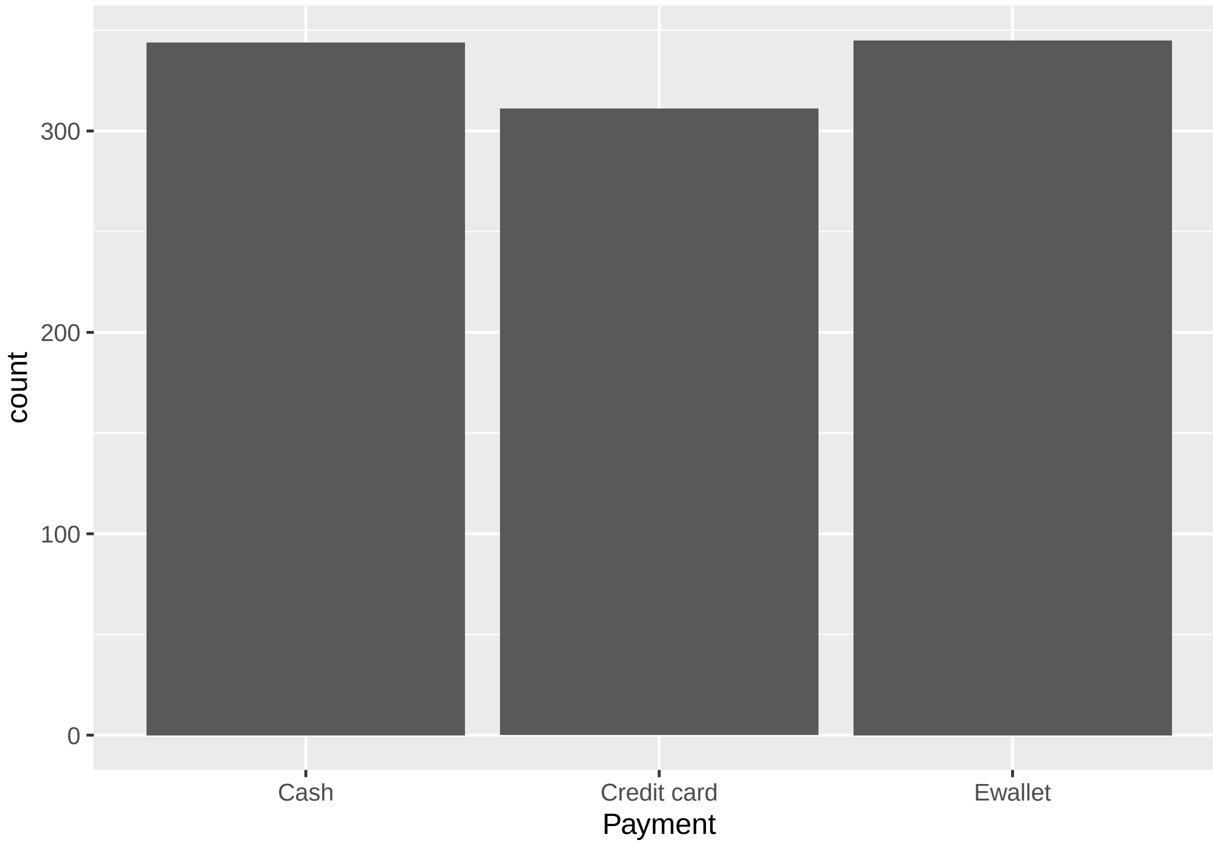
The balanced dustribution in customer type as well

```
ggplot(data, aes(Product.line)) + geom_bar()
```



Product Line
* Fashion Accessories and Food and Beverage are the most bought, fashion accesories being the most bought of the two categories. The distribution is quite okay.

```
# visualizing Payment mode
ggplot(data, aes(Payment)) + geom_bar(stat="count")
```



Payment

There is a fair distribution in the payment variable. However, fewer people tend to pay by Credit Card in these stores

Unit Price Investigating Measures of disperion among numerical variables

```
# Mean
uprice.mean <- mean(data$Unit.price)
uprice.mean
```

```
## [1] 55.67213
```

```
# Mode
uprice.mode <- mode(data$Unit.price)
uprice.mode
```

```
## [1] 83.77
```

```
# Median
uprice.median <- median(data$Unit.price)
uprice.median
```

```
## [1] 55.23
```

```r
# Standard Deviation
uprice.sd <- sd(data$Unit.price)
uprice.sd
```

```
## [1] 26.49463
```

```r
# Kurtosis
uprice.kurt <- kurtosis(data$Unit.price)
uprice.kurt
```

```
## [1] -1.218501
```

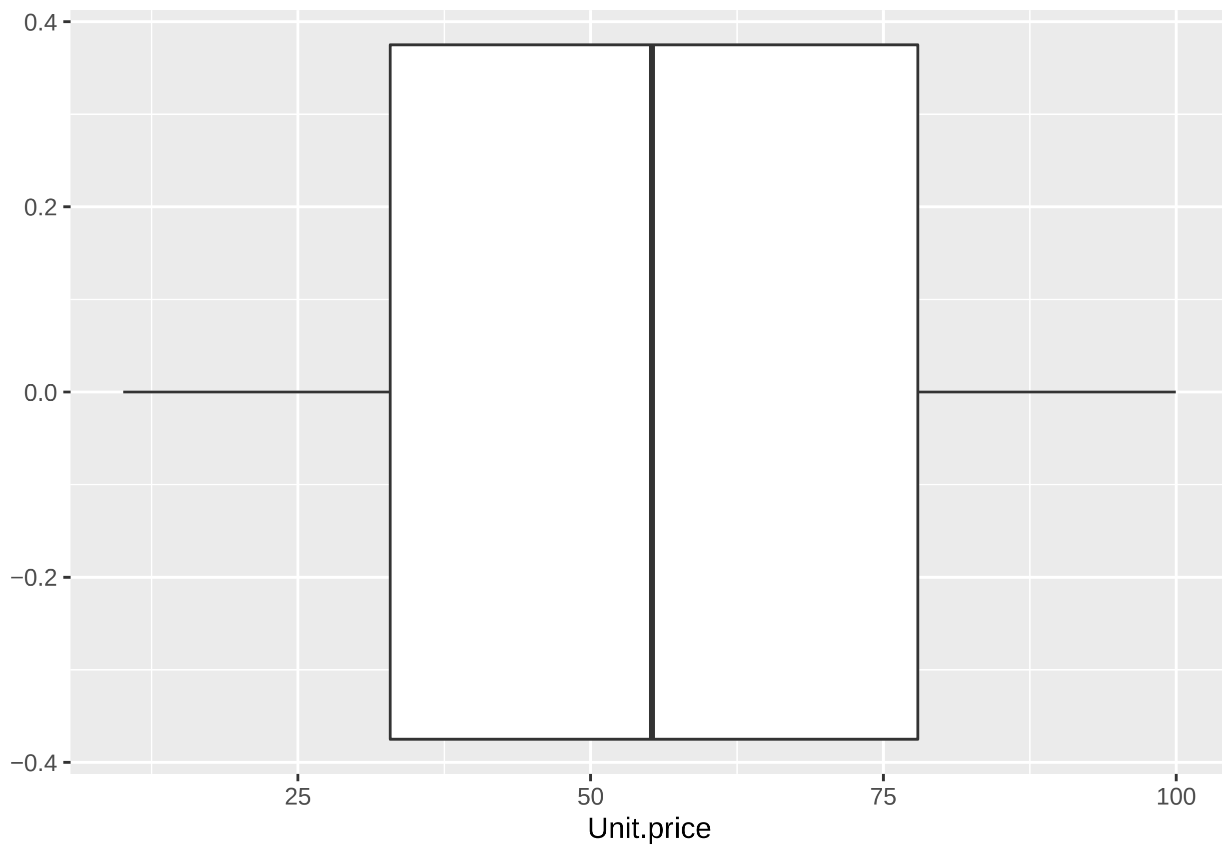
```r
# SKewness
uprice.skew <- skewness(data$Unit.price)
uprice.skew
```

```
## [1] 0.007066827
```

```r
# Range
uprice.range <- range(data$Unit.price)
uprice.range
```

```
## [1] 10.08 99.96
```

```r
# Visualizing distribution
ggplot(data, aes(Unit.price)) +
  geom_boxplot(outlier.colour = "red")
```




```r
# mean
quantity.mean <- mean(data$Quantity)
quantity.mean
```

Quantity

```
## [1] 5.51
```

```r
# Mode
quantity.mode <- mode(data$Quantity)
quantity.mode
```

```
## [1] 10
```

```r
# Median
quantity.median <- median(data$Quantity)
quantity.median
```

```
## [1] 5
```

```r
# Standard Deviation
quantity.sd <- sd(data$Quantity)
quantity.sd
```

```
## [1] 2.923431
```

```r
# Range
quantity.range <- range(data$Quantity)
quantity.range
```

```
## [1]  1 10
```

```r
# Kurtosis
quantity.kurt <- kurtosis(data$Quantity)
quantity.kurt
```

```
## [1] -1.215472
```

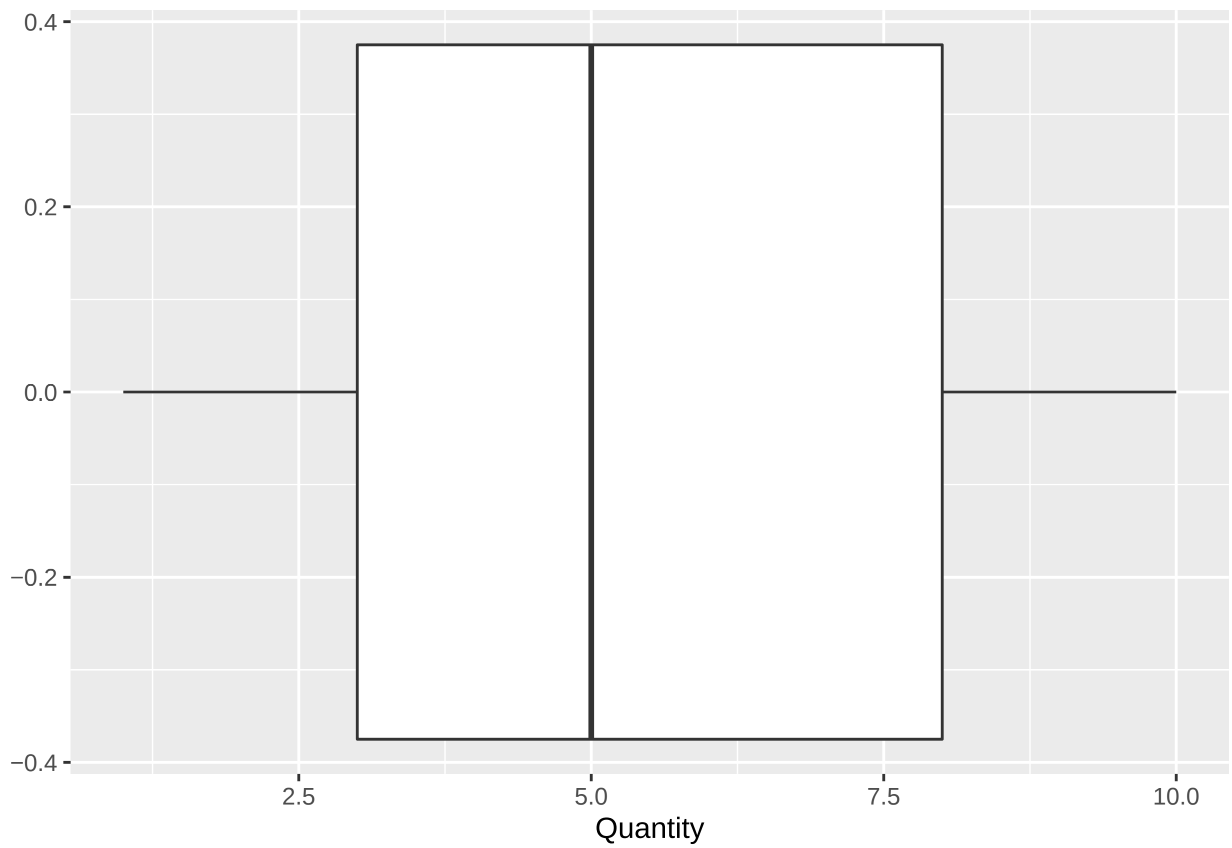
```r
# Skewness
quantity.skew <- skewness(data$Quantity)
quantity.skew
```

```
## [1] 0.01292163
```

```r
# Quantiles
quantity.quants <- quantile(data$Quantity)
quantity.quants
```

```
##   0%  25%  50%  75% 100% 
##    1    3    5    8   10
```

```r
# Visualizing distribution
ggplot(data, aes(Quantity)) +
  geom_boxplot(outlier.colour = "red")
```




```r
# mean
tax.mean <- mean(data$Tax)
tax.mean
```

Tax

```
## [1] 15.37937
```

```r
# mode
tax.mode <- mode(data$Tax)
tax.mode
```

```
## [1] 39.48
```

```r
# Median
tax.median <- median(data$Tax)
tax.median
```

```
## [1] 12.088
```

```r
# Standard Deviation
tax.sd <- sd(data$Tax)
tax.sd
```

```
## [1] 11.70883
```

```r
# Kurtosis
tax.kurt <- kurtosis(data$Tax)
tax.kurt
```

```
## [1] -0.08746991
```

```r
# Skewness
tax.skew <- skewness(data$Tax)
tax.skew
```

```
## [1] 0.8912304
```

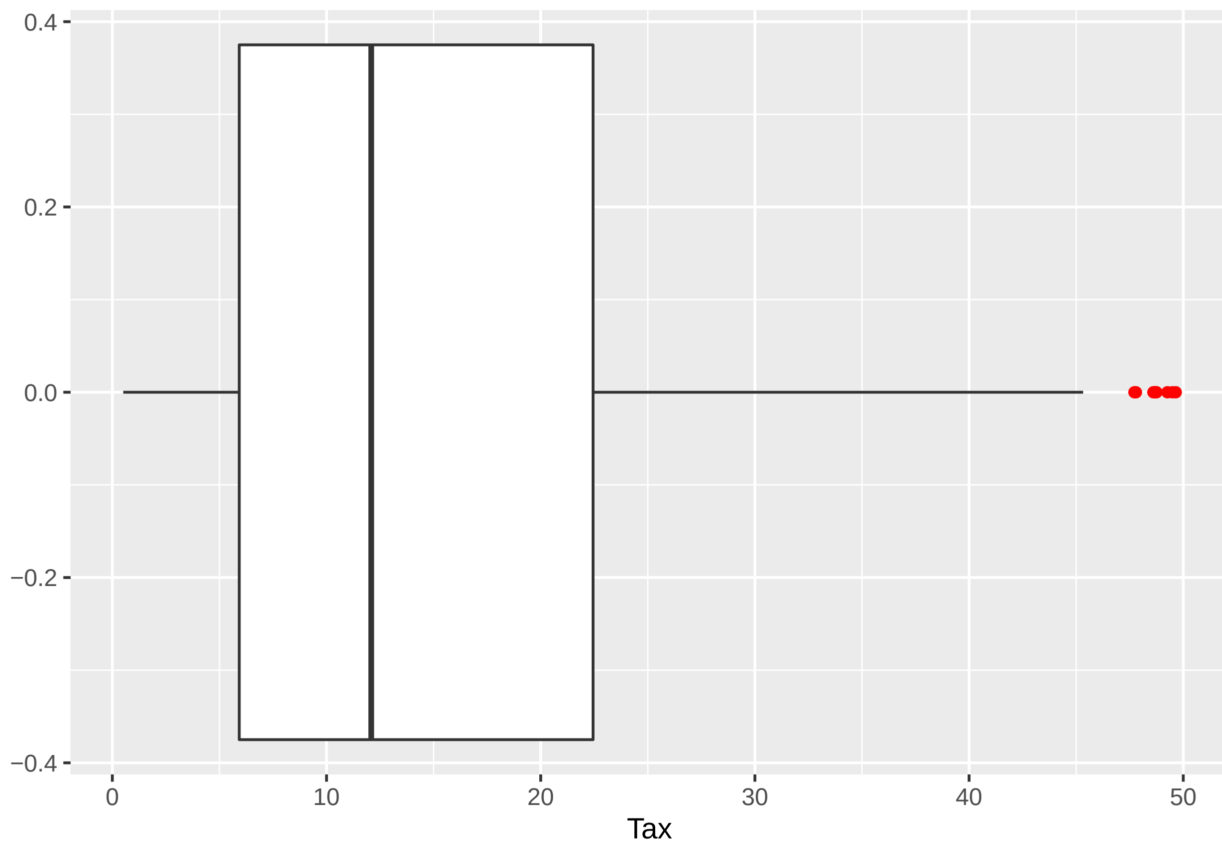
```r
# Range
tax.range <- range(data$Tax)
tax.range
```

```
## [1]  0.5085 49.6500
```

```r
# Quantiles
tax.quantiles <- quantile(data$Tax)
tax.quantiles
```

```
##        0%       25%       50%       75%      100%
##  0.508500  5.924875 12.088000 22.445250 49.650000
```

```r
# Visualizing dustribution
ggplot(data, aes(Tax)) +
  geom_boxplot(outlier.colour = "red")
```




```
# mode
date.mode <- mode(data$Date)
date.mode
```

Date

```
## [1] "2020-02-07"
```

```
# median
date.median <- median(data$Date)
date.median
```

```
## [1] "2020-02-13"
```

```
# standard deviation
date.sd <- sd(data$Date)
date.sd
```

```
## [1] 25.51686
```

```r
# Kurtosis
date.kurt <- kurtosis(data$Date)
date.kurt
```

```
## [1] -1.197667
```

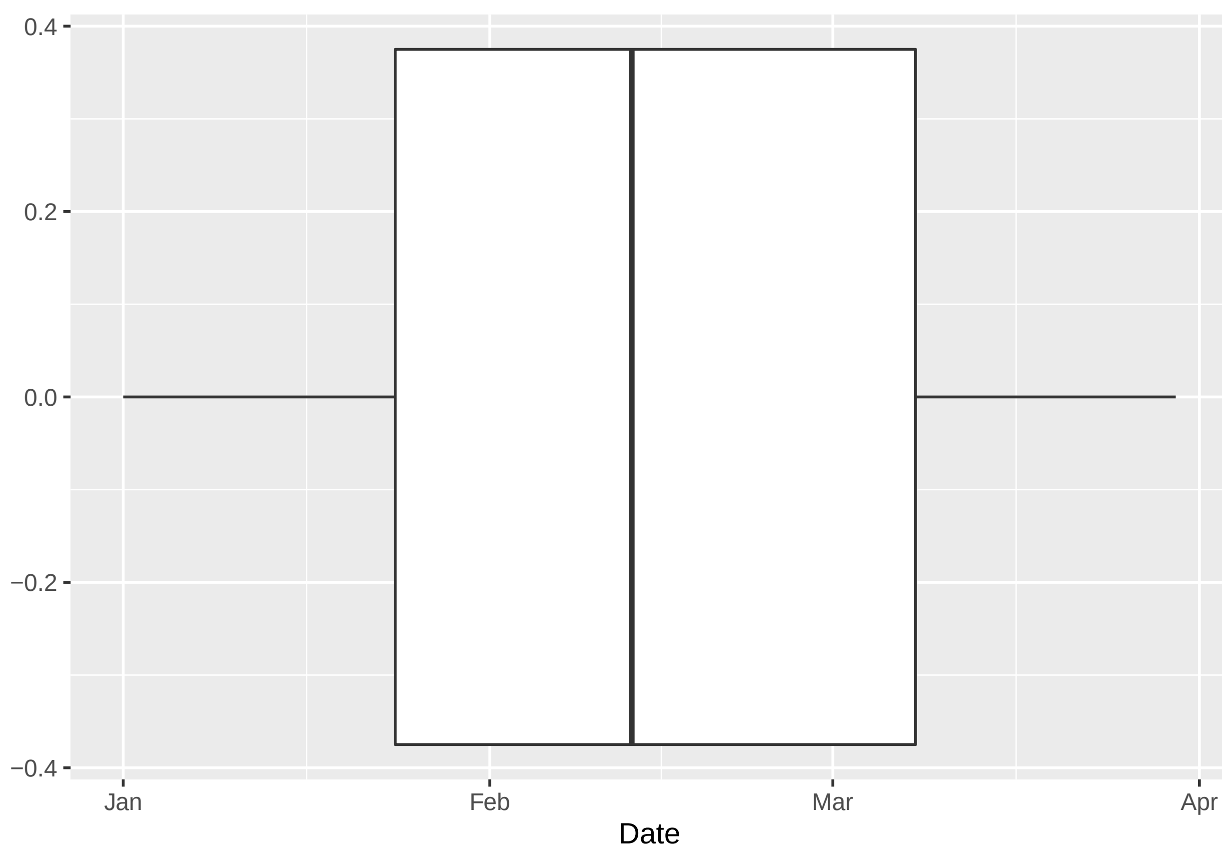
```r
# Skewness
date.skew <- skewness(data$Date)
date.skew
```

```
## [1] 0.03704369
```

```r
# Range
date.range <- range(data$Date)
date.range
```

```
## [1] "2020-01-01" "2020-03-30"
```

```r
# Visualizing dustribution
ggplot(data, aes(Date)) +
  geom_boxplot(outlier.colour = "red")
```



* Most data is from February and March

```r
head(data)
```

```
##   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       Date  Time     Payment   cogs gross.margin.percentage
## 1 26.1415 2020-01-05 13:08     Ewallet 522.83                4.761905
## 2  3.8200 2020-03-08 10:29        Cash  76.40                4.761905
## 3 16.2155 2020-03-03 13:23 Credit card 324.31                4.761905
## 4 23.2880 2020-01-27 20:33     Ewallet 465.76                4.761905
## 5 30.2085 2020-02-08 10:37     Ewallet 604.17                4.761905
## 6 29.8865 2020-03-25 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
# mean
cogs.mean <- mean(data$cogs)
cogs.mean
```

COGS

```
## [1] 307.5874
```

```r
# mode
cogs.mode <- mode(data$cogs)
cogs.mode
```

```
## [1] 789.6
```

```r
# median
cogs.median <- median(data$cogs)
cogs.median
```

```
## [1] 241.76
```

```r
# standard deviation
cogs.sd <- sd(data$cogs)
cogs.sd
```

```
## [1] 234.1765
```

```r
# range
cogs.range <- range(data$cogs)
cogs.range
```

```
## [1]  10.17 993.00
```

```r
# kurtosis
cogs.kurt <- kurtosis(data$cogs)
cogs.kurt
```

```
## [1] -0.08746991
```

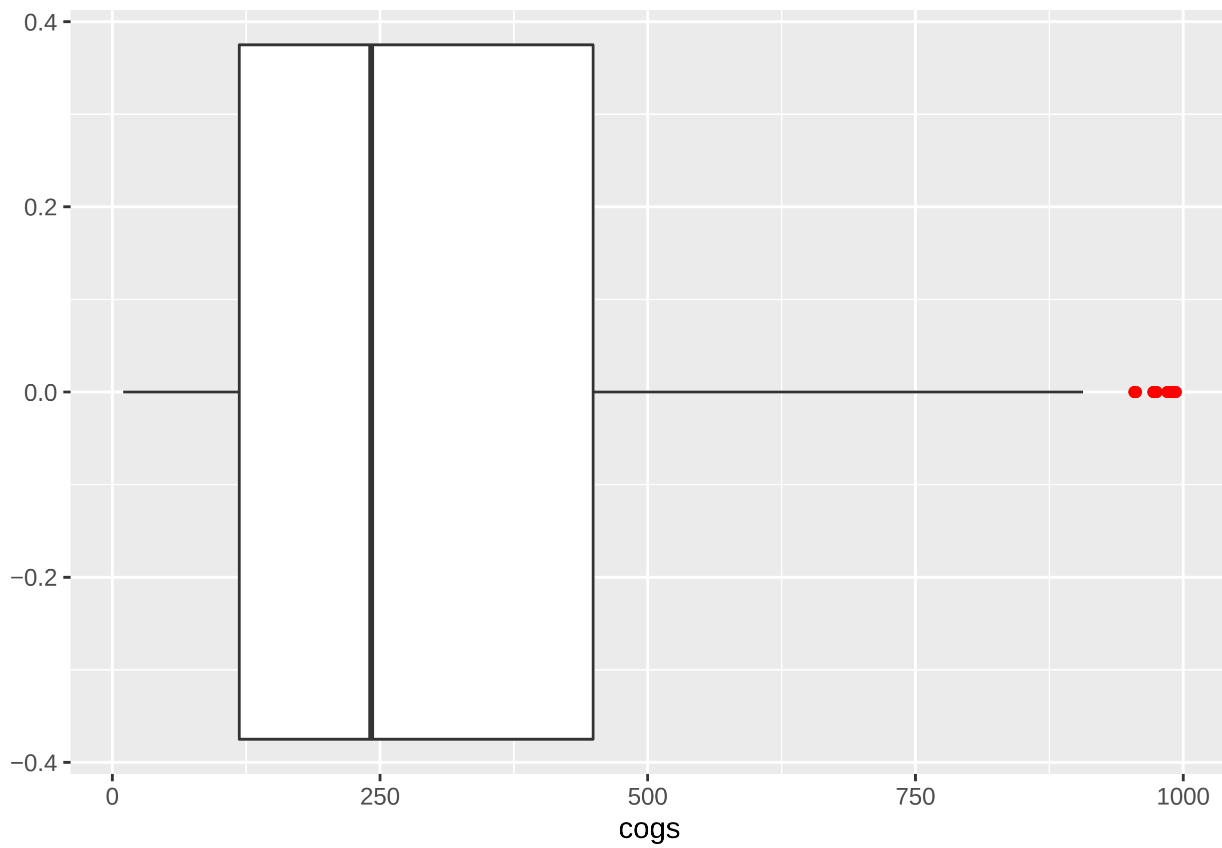
```r
# skewness
cogs.skew <- skewness(data$cogs)
cogs.skew
```

```
## [1] 0.8912304
```

```r
# quantiles
cogs.quantiles <- quantile(data$cogs)
cogs.quantiles
```

```
##       0%      25%      50%      75%     100%
##  10.1700 118.4975 241.7600 448.9050 993.0000
```

```r
# visualizing
ggplot(data, aes(cogs)) +
  geom_boxplot(outlier.colour = "red")
```




```r
# mean
gross.mean <- mean(data$gross)
```

Gross Income

```
## Warning in mean.default(data$gross): argument is not numeric or logical:
## returning NA
```

```r
gross.mean
```

```
## [1] NA
```

```r
# mode
gross.mode <- mode(data$gross)
gross.mode
```

```
## NULL
```

```r
# median
gross.median <- median(data$gross)
gross.median
```

```
## NULL
```

```r
# range
gross.range <- range(data$gross)
```

```
## Warning in min(x, na.rm = na.rm): no non-missing arguments to min; returning Inf
```

```
## Warning in max(x, na.rm = na.rm): no non-missing arguments to max; returning
## -Inf
```

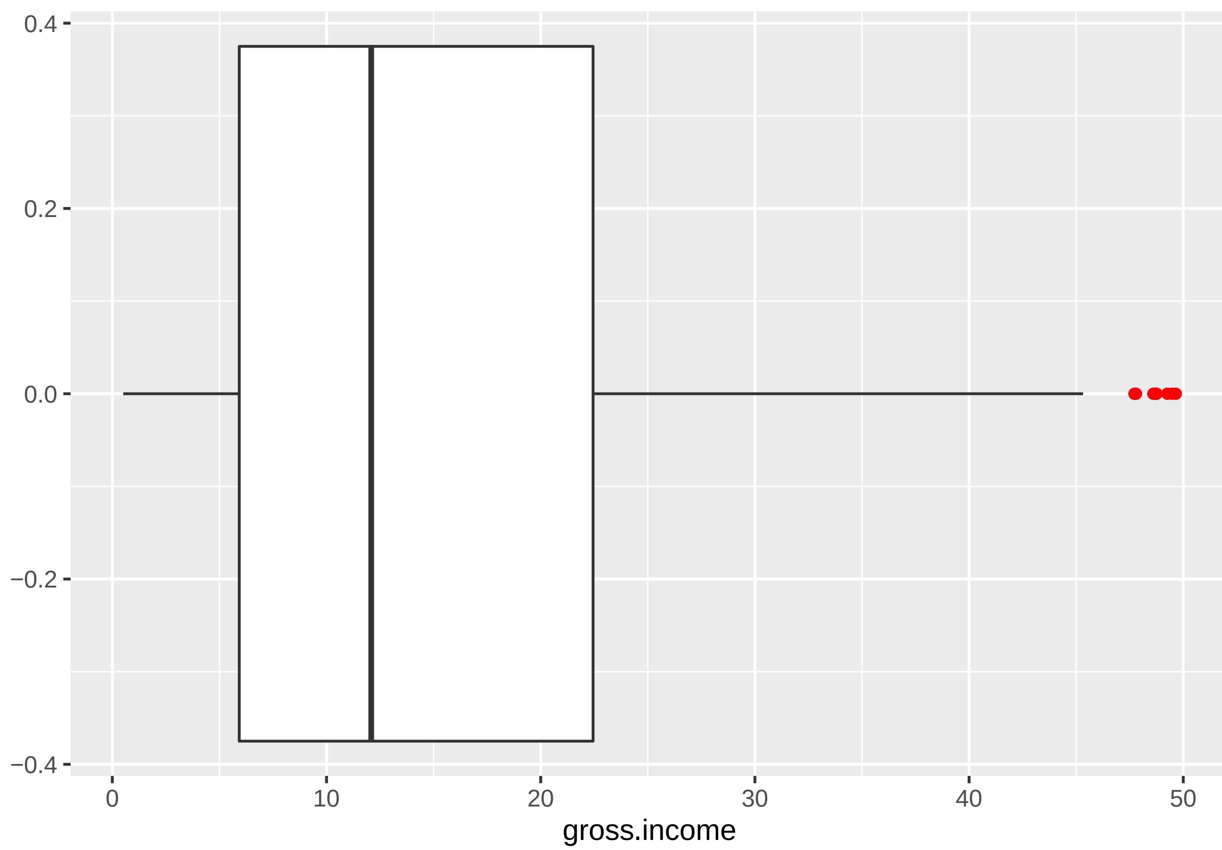
```r
gross.range
```

```
## [1]  Inf -Inf
```

```r
# standard deviation
gross.sd <- sd(data$gross)
gross.sd
```

```
## [1] NA
```

```r
# visualizing distribution
 ggplot(data, aes(gross.income)) +
   geom_boxplot(outlier.colour = "red")
```



```r
# mean
rate.mean <- mean(data$Rating)
rate.mean
```

Rating

```
## [1] 6.9727
```

```r
# mode
rate.mode <- mode(data$Rating)
rate.mode
```

```
## [1] 6
```

```r
# median
rate.median <- median(data$Rating)
rate.median
```

```
## [1] 7
```

```r
# standard deviation
rate.sd <- sd(data$Rating)
rate.sd
```

```
## [1] 1.71858
```

```r
# range
rate.range <- range(data$Rating)
rate.range
```

```
## [1]  4 10
```

```r
# quantiles
rate.quantiles <- quantile(data$Rating)
rate.quantiles
```

```
##   0%  25%  50%  75% 100%
##  4.0  5.5  7.0  8.5 10.0
```

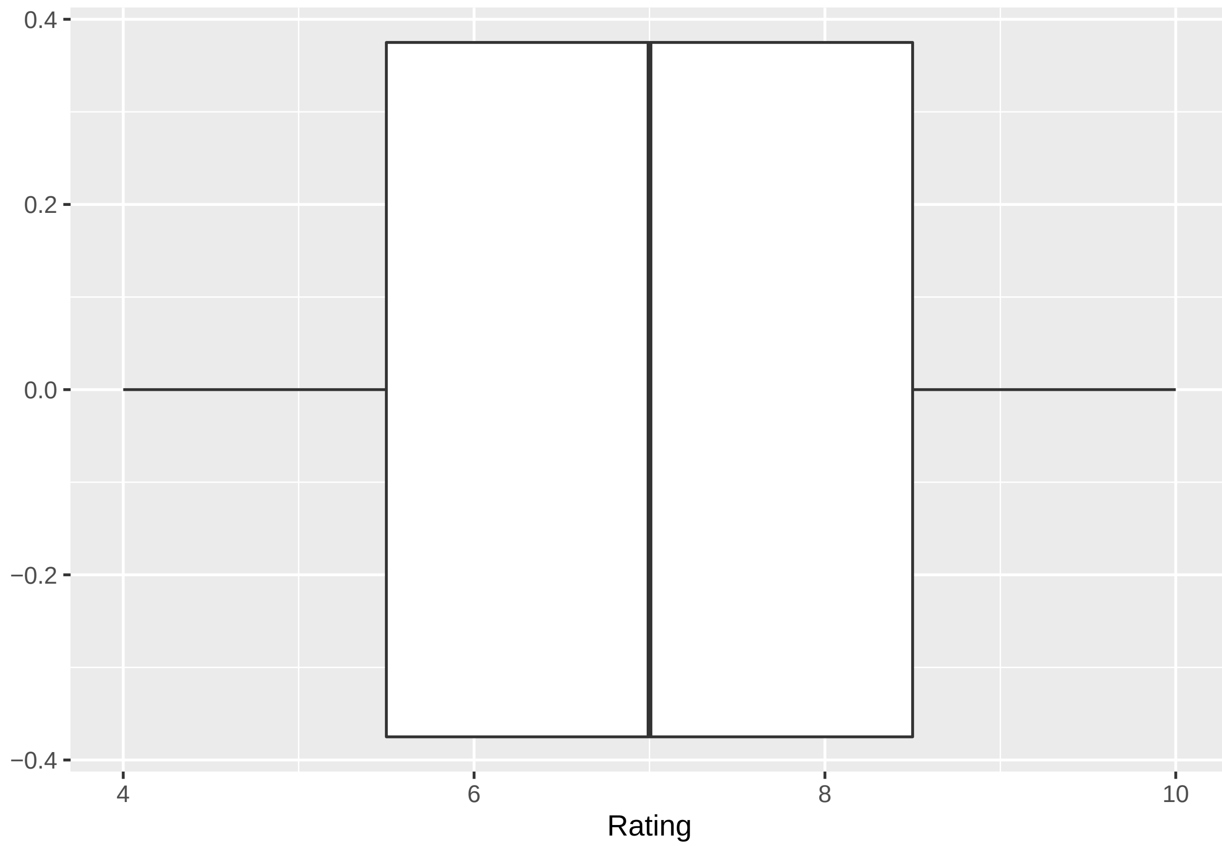
```r
# kurtosis
rate.kurt <- kurtosis(data$Rating)
rate.kurt
```

```
## [1] -1.151831
```

```r
# skewness
rate.skew <- skewness(data$Rating)
rate.skew
```

```
## [1] 0.008996129
```

```r
# visualizing distribution
ggplot(data, aes(Rating)) + geom_boxplot(outlier.colour = "red")
```



* 6-8 ratings are the most common in the dataset

```r
# mean
total.mean <- mean(data$Total)
total.mean
```

Total

```
## [1] 322.9667
```

```r
# median
total.median <- median(data$Total)
total.median
```

```
## [1] 253.848
```

```r
# mode
total.mode <- mode(data$Total)
total.mode
```

```
## [1] 829.08
```

```r
# standard deviation
total.sd <- sd(data$Total)
total.sd
```

```
## [1] 245.8853
```

```r
# range
total.range <- range(data$Total)
total.range
```

```
## [1]   10.6785 1042.6500
```

```r
# kurtosis
total.kurt <- kurtosis(data$Total)
total.kurt
```

```
## [1] -0.08746991
```

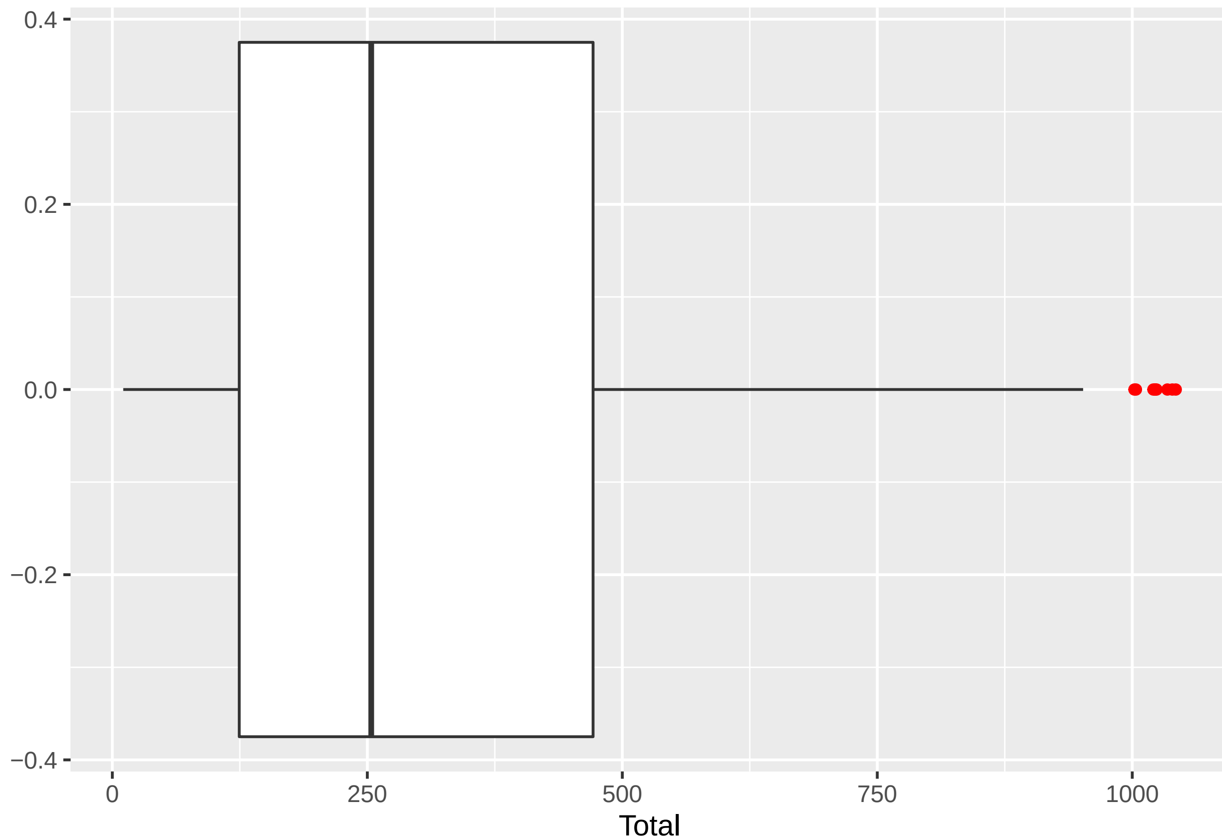
```r
# skewness
total.skew <- skewness(data$Total)
total.skew
```

```
## [1] 0.8912304
```

```r
# quantiles
total.quantiles <- quantile(data$Total)
total.quantiles
```

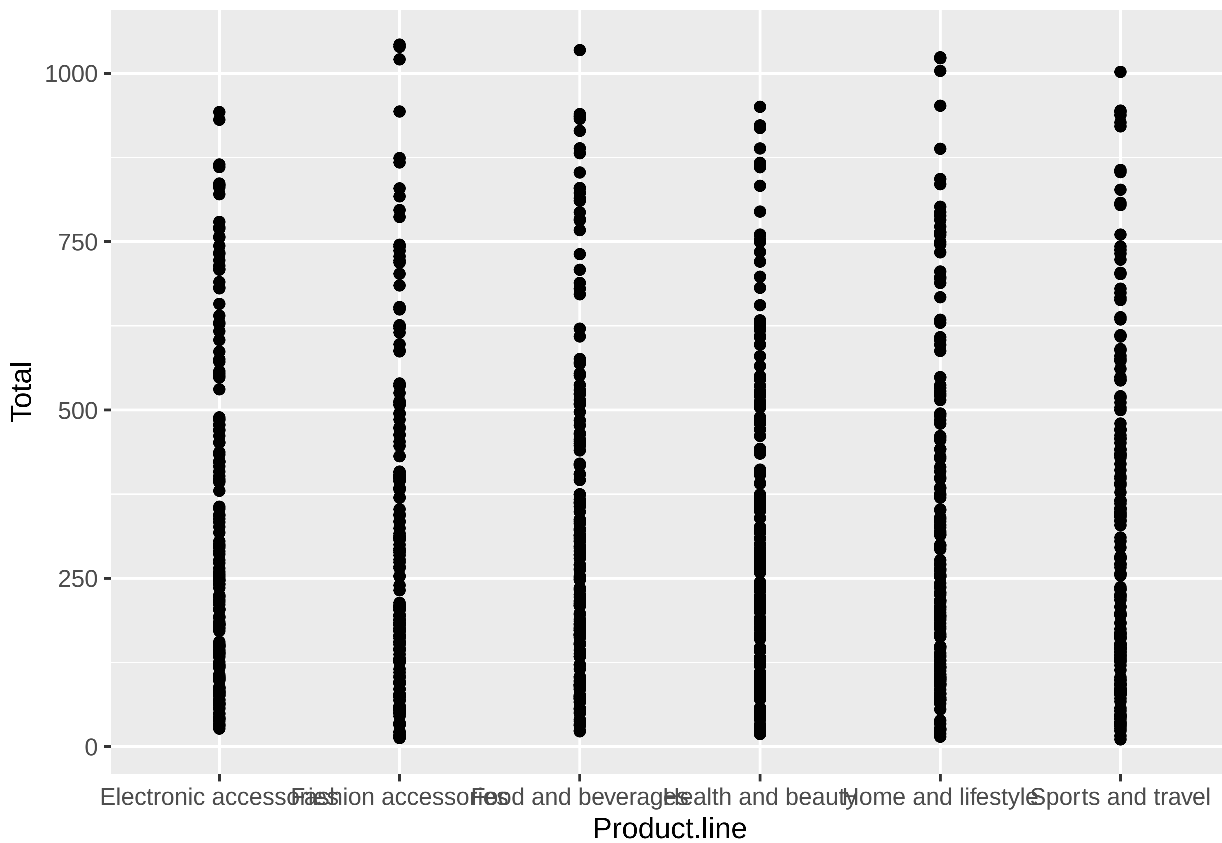
```
##        0%       25%       50%       75%      100%
##   10.6785  124.4224  253.8480  471.3502 1042.6500
```

```r
# visual
ggplot(data, aes(Total)) +
  geom_boxplot(outlier.colour = "red" )
```



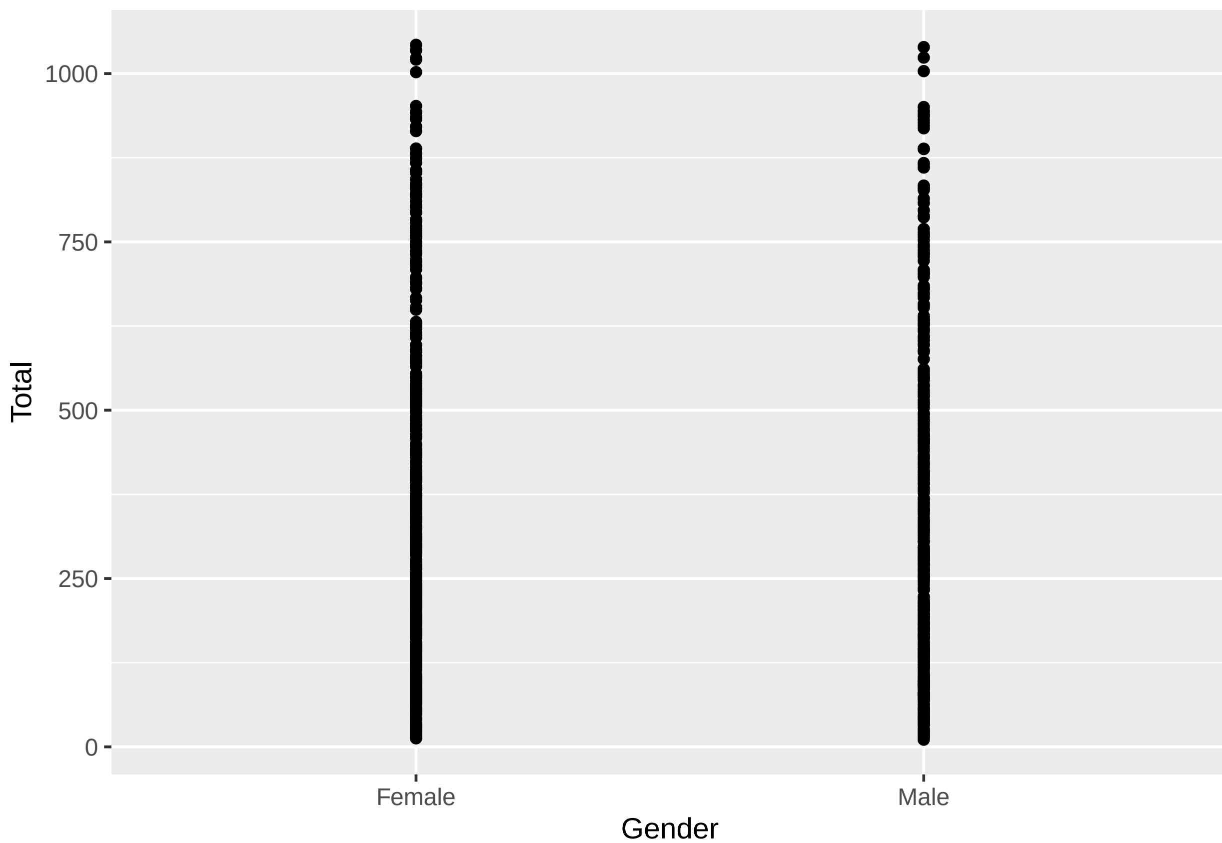
Bivariate Analysis

```
ggplot(data, aes(x=Product.line, y=Total)) + geom_point()
```



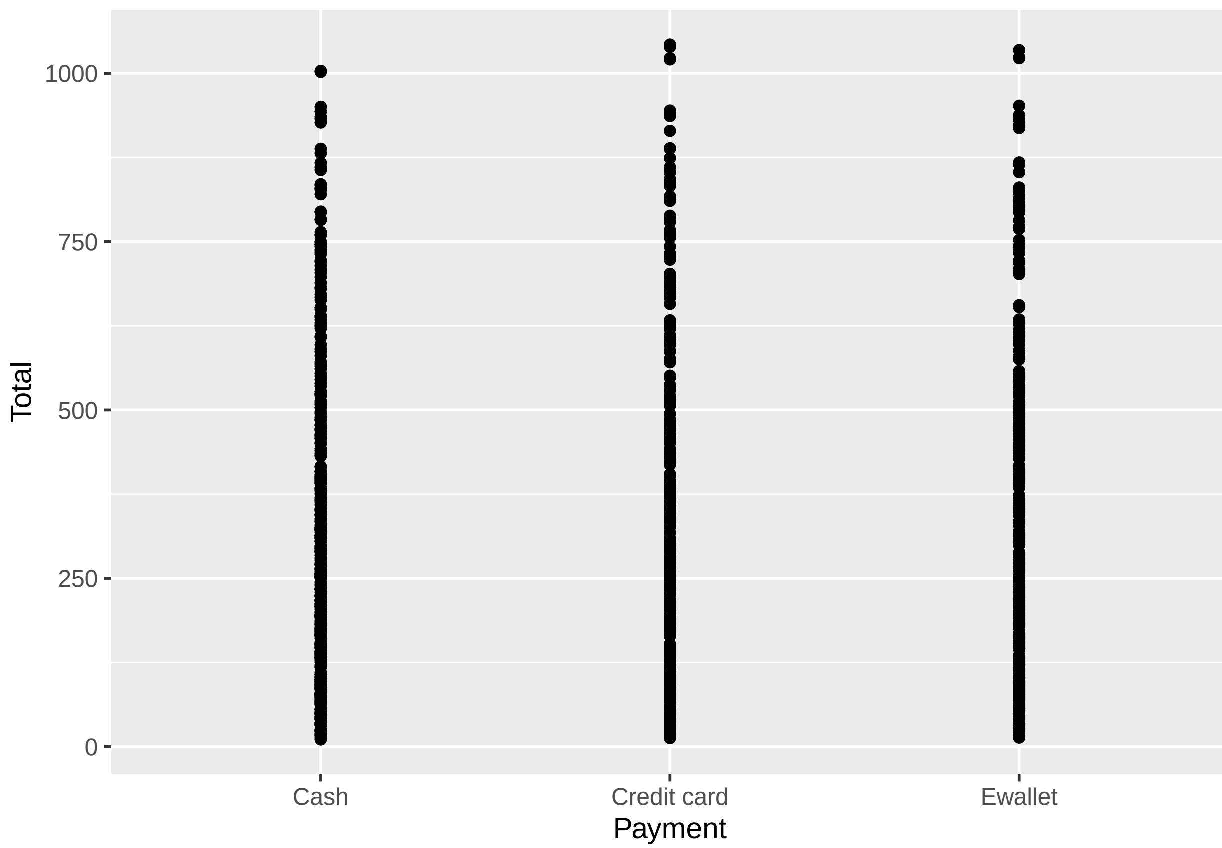
Fashion Accessories have the highest Total prices while health and beauty products have a relatively lower price.

```
ggplot(data ,aes(Gender, Total)) + geom_point()
```



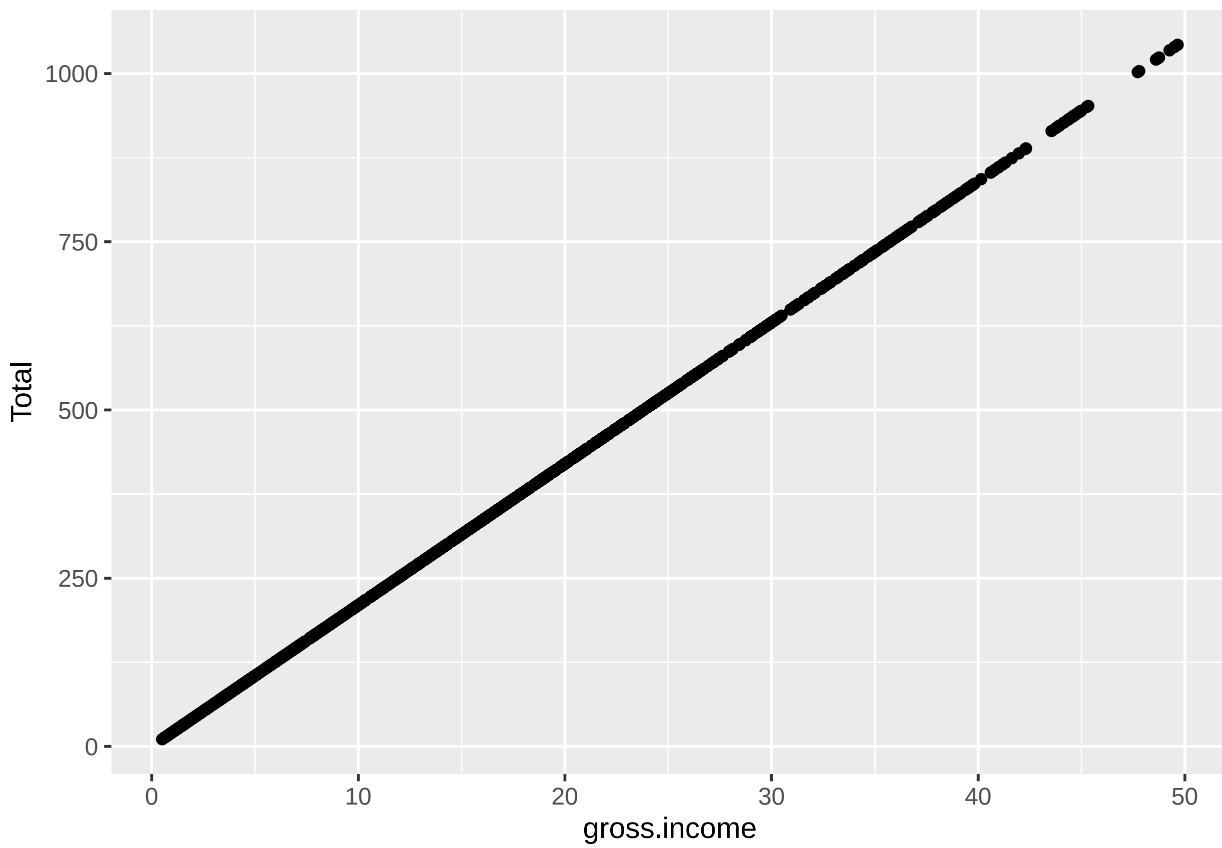
Apparently Females spend slightly more on the products

```
ggplot(data, aes(Payment, Total)) +
  geom_point()
```



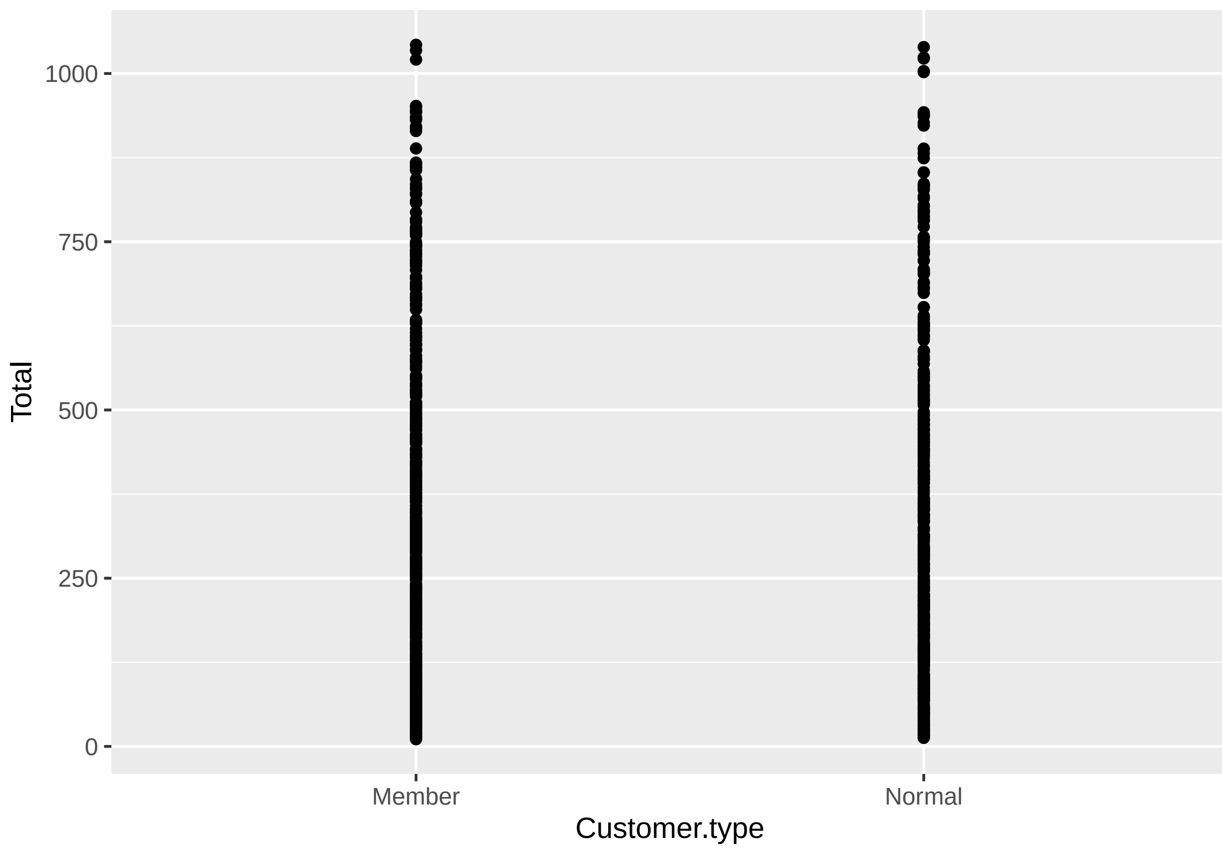
The payment methods are nearly identical for the total prices of items at checkouts with some more expensive ones being attributed with Credit card payments.

```
ggplot(data, aes(gross.income, Total)) + geom_point()
```



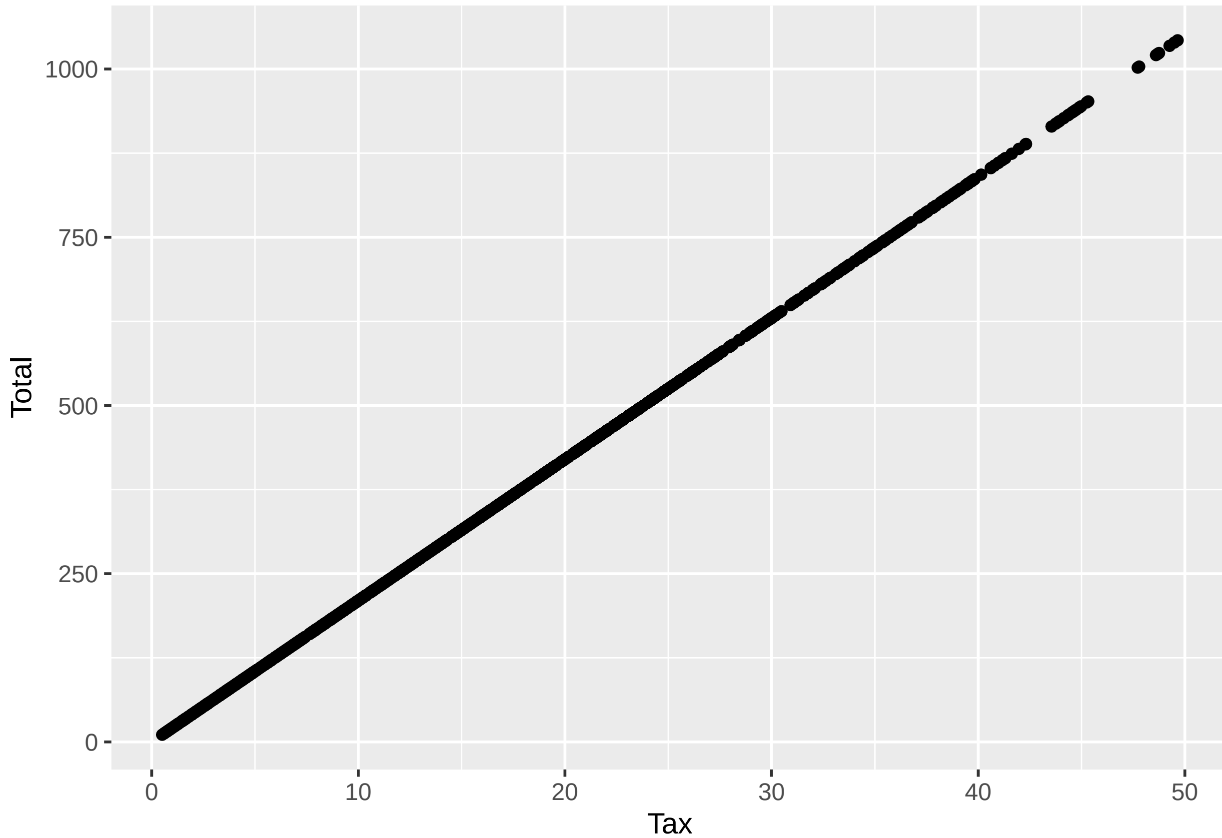
there's a linear relationship between the total at checkout and the consumers gross income where a gross income increases so does the total.

```
ggplot(data, aes(Customer.type , Total)) +
  geom_point()
```



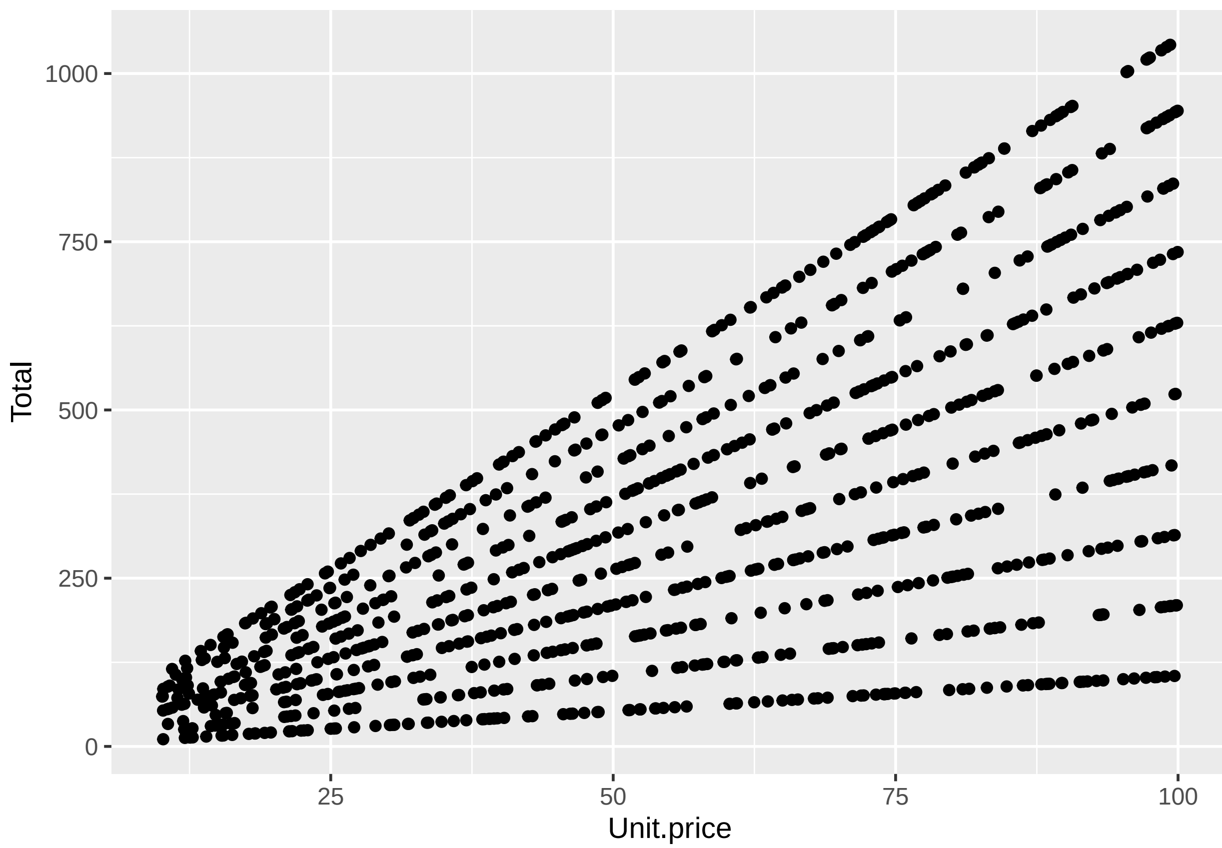
Members and non members have a nearly equal distribution in expenditure with Members having no visible breaks in prices.

```
ggplot(data, aes(Tax, Total)) +
  geom_point()
```



Tax has aa similar relationship with Total as that of gross income.

```
ggplot(data, aes(Unit.price, Total)) +
  geom_point()
```

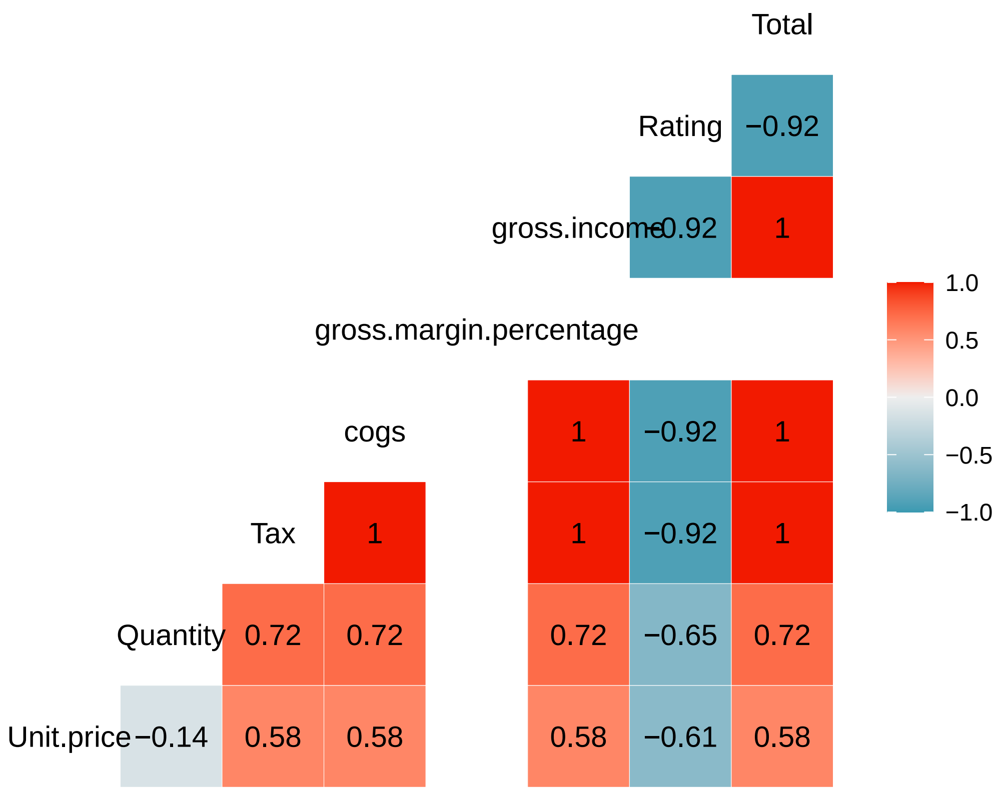


There are several linear relationships with the Unit Price, that is, the higher the unit price is, the higher the total price is. This is most likely brought about by the fact that products being of different types.

```
corr<- cor(data[,unlist(lapply(data, is.numeric))])
```

```
## Warning in cor(data[, unlist(lapply(data, is.numeric))]): the standard deviation
## is zero
```

```
ggcorr(corr, label = T, label_round = 2)
```




```
Total has a trong negative correlation to rating. Rating has a strong negative correlation
to gross income, cogs and tax. Cogs and tax are highly possitively correlated with a
correlation of 1. Gross income has a corelatio of 1 as well with cogs and tax as well
which isn't surprice given the bivariate analysis results
```

```r
# creating a copy of the dataset
copy <- data[, -c(8, 9, 12, 15)]
# defining the label
label <- data[, 15]
```

USING THE t-SNE ALGORITHM

Thi ection entails reducing the dataset to a low dimensional dataset using the t-SNE algorithm ### Label Encoding the categorical columns

```r
branch <- LabelEncoder.fit(copy$Branch)
copy$Branch <- transform(branch, factor(data$Branch))
gender <- LabelEncoder.fit(copy$Gender)
copy$Gender <- transform(gender, factor(data$Gender))
customer <- LabelEncoder.fit(copy$Customer.type)
copy$Customer.type <- transform(customer, factor(copy$Customer.type))
product <- LabelEncoder.fit(copy$Product.line)
copy$Product.line <- transform(product, factor(copy$Product.line))
pay <- LabelEncoder.fit(copy$Payment)
copy$Payment <- transform(pay, factor(copy$Payment))
```

Building the model

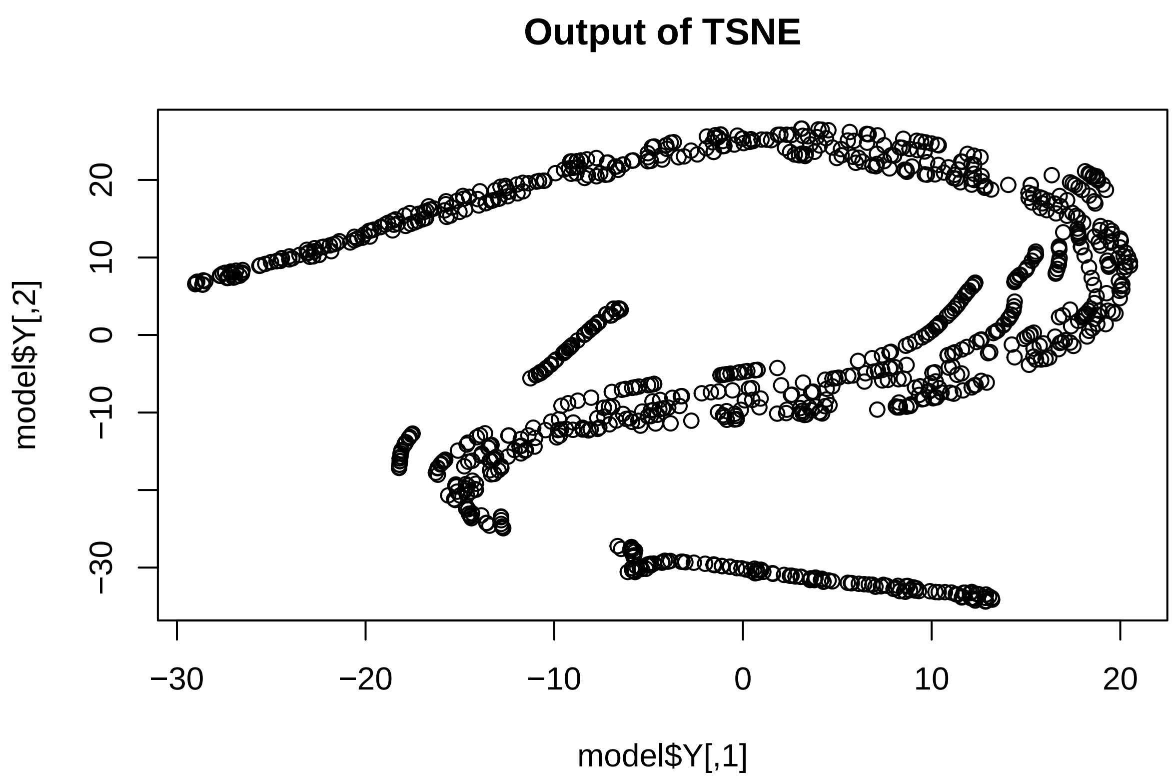
```
model <- Rtsne(copy, dims=2, perplexity=30, verbose= TRUE, max_iter=1000)
```

```
## Performing PCA
## Read the 1000 x 11 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.31 seconds (sparsity = 0.102676)!
## Learning embedding...
## Iteration 50: error is 59.676531 (50 iterations in 0.17 seconds)
## Iteration 100: error is 52.879959 (50 iterations in 0.17 seconds)
## Iteration 150: error is 51.798932 (50 iterations in 0.22 seconds)
## Iteration 200: error is 51.365005 (50 iterations in 0.20 seconds)
## Iteration 250: error is 51.149497 (50 iterations in 0.25 seconds)
## Iteration 300: error is 0.576812 (50 iterations in 0.26 seconds)
## Iteration 350: error is 0.409352 (50 iterations in 0.20 seconds)
## Iteration 400: error is 0.366684 (50 iterations in 0.16 seconds)
## Iteration 450: error is 0.351530 (50 iterations in 0.22 seconds)
## Iteration 500: error is 0.344341 (50 iterations in 0.15 seconds)
## Iteration 550: error is 0.336047 (50 iterations in 0.24 seconds)
## Iteration 600: error is 0.331091 (50 iterations in 0.17 seconds)
## Iteration 650: error is 0.329118 (50 iterations in 0.23 seconds)
## Iteration 700: error is 0.324561 (50 iterations in 0.14 seconds)
## Iteration 750: error is 0.324090 (50 iterations in 0.22 seconds)
## Iteration 800: error is 0.324095 (50 iterations in 0.26 seconds)
## Iteration 850: error is 0.322250 (50 iterations in 0.19 seconds)
## Iteration 900: error is 0.321211 (50 iterations in 0.21 seconds)
## Iteration 950: error is 0.321234 (50 iterations in 0.22 seconds)
## Iteration 1000: error is 0.320375 (50 iterations in 0.25 seconds)
## Fitting performed in 4.15 seconds.
```

```
summary(model)
```

```
##                     Length Class  Mode
## N                      1   -none- numeric
## Y                   2000   -none- numeric
## costs               1000   -none- numeric
## itercosts             20   -none- numeric
## origD                  1   -none- numeric
## perplexity             1   -none- numeric
## theta                  1   -none- numeric
## max_iter               1   -none- numeric
## stop_lying_iter        1   -none- numeric
## mom_switch_iter        1   -none- numeric
## momentum               1   -none- numeric
## final_momentum         1   -none- numeric
## eta                    1   -none- numeric
## exaggeration_factor    1   -none- numeric
```

```
plot(model$Y, t='p', main="Output of TSNE")
```

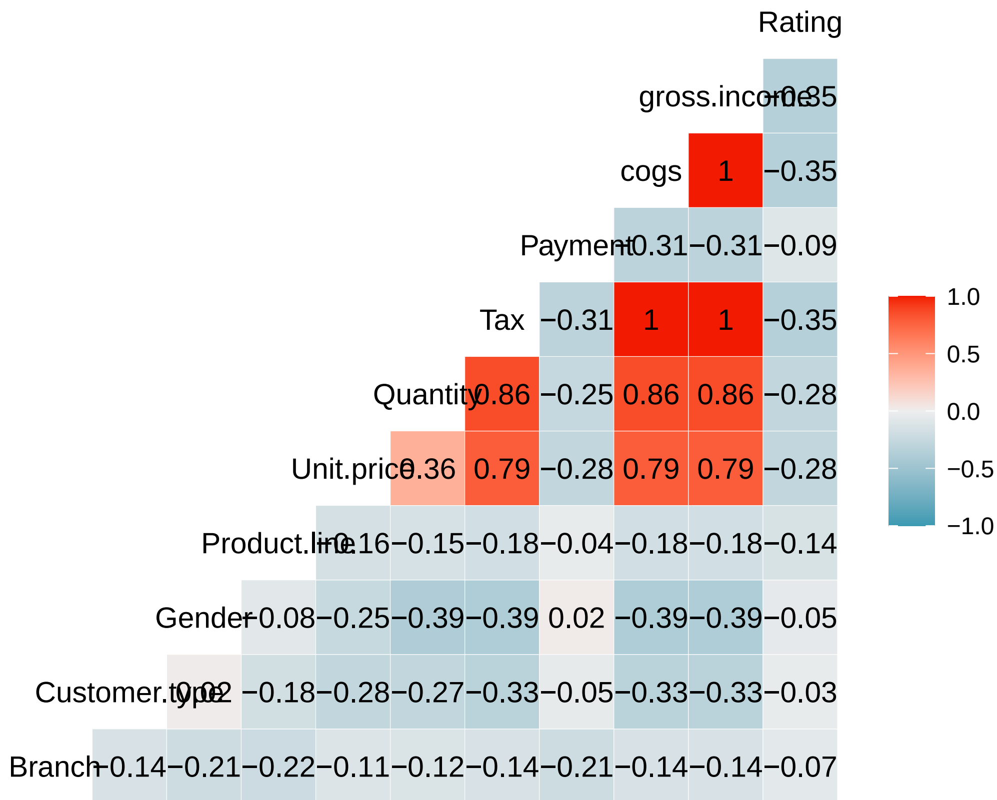


Feature Selection

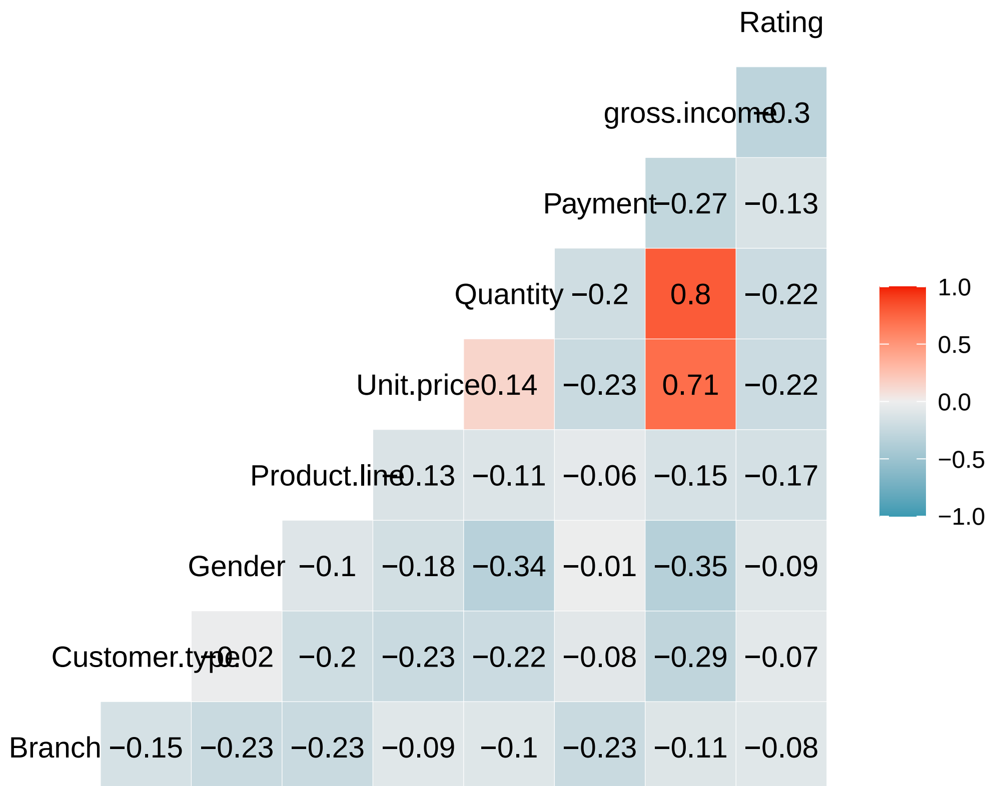
```
corrMat <- cor(copy)
# storing highly correlated features in "high"
high <- findCorrelation(corrMat, cutoff = .75)
#getting their names
names(copy[, high])
```

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

```
# removing the highly correlated variables
copy2 <- copy[-high]
par(mfrow = c(1, 2))
# plotting the comparison
ggcorr(corrMat, label = T, label_round = 2)
```




```
ggcorr(cor(copy2), label = T, label_round = 2)
```



Much better. Now it's suitable for modeling.