ISF_visualisation

2023-11-03

```
data<- read.csv("ifood_df.csv")
```

```
head(data)
```

```
##   Income Kidhome Teenhome Recency MntWines MntFruits MntMeatProducts
## 1  58138       0        0      58      635        88             546
## 2  46344       1        1      38       11         1               6
## 3  71613       0        0      26      426        49             127
## 4  26646       1        0      26       11         4              20
## 5  58293       1        0      94      173        43             118
## 6  62513       0        1      16      520        42              98
##   MntFishProducts MntSweetProducts MntGoldProds NumDealsPurchases
## 1             172               88          88                 3
## 2               2                1           6                 2
## 3             111               21          42                 1
## 4              10                3           5                 2
## 5              46               27          15                 5
## 6               0               42          14                 2
##   NumWebPurchases NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
## 1               8                  10                 4                 7
## 2               1                   1                 2                 5
## 3               8                   2                10                 4
## 4               2                   0                 4                 6
## 5               5                   3                 6                 5
## 6               6                   4                10                 6
##   AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain
## 1            0            0            0            0            0        0
## 2            0            0            0            0            0        0
## 3            0            0            0            0            0        0
## 4            0            0            0            0            0        0
## 5            0            0            0            0            0        0
## 6            0            0            0            0            0        0
##   Z_CostContact Z_Revenue Response Age Customer_Days marital_Divorced
## 1             3        11        1  63          2822                0
## 2             3        11        0  66          2272                0
## 3             3        11        0  55          2471                0
## 4             3        11        0  36          2298                0
## 5             3        11        0  39          2320                0
## 6             3        11        0  53          2452                0
##   marital_Married marital_Single marital_Together marital_Widow
## 1               0              1                0             0
## 2               0              1                0             0
## 3               0              0                1             0
## 4               0              0                1             0
## 5               1              0                0             0
```

```
## 6               0              0                1             0
##   education_2n.Cycle education_Basic education_Graduation education_Master
## 1                  0               0                    1                0
## 2                  0               0                    1                0
## 3                  0               0                    1                0
## 4                  0               0                    1                0
## 5                  0               0                    0                0
## 6                  0               0                    0                1
##   education_PhD MntTotal MntRegularProds AcceptedCmpOverall
## 1             0     1529            1441                  0
## 2             0       21              15                  0
## 3             0      734             692                  0
## 4             0       48              43                  0
## 5             1      407             392                  0
## 6             0      702             688                  0
```

Cette base de données contient des informations sur 2205 clients. L'objectif de ce projet R est de créer plusieurs groupes de clients qui différent sur plusieurs caractéristiques. Cette segmentation client va permettre par exemple d'adapter les offres et publicités en fonction des différents groupes de clients, et ainsi attirer plus de prospects.

```
dim(data)
```

```
## [1] 2205   39
```

```
cat("Le nombre de lignes du dataset est de:", dim(data)[1],'\n')
```

```
## Le nombre de lignes du dataset est de: 2205
```

```
cat("Le nombre de features du dataset est de:", dim(data)[2])
```

```
## Le nombre de features du dataset est de: 39
```

```
names(data)
```

```
##  [1] "Income"               "Kidhome"              "Teenhome"
##  [4] "Recency"              "MntWines"             "MntFruits"
##  [7] "MntMeatProducts"      "MntFishProducts"      "MntSweetProducts"
## [10] "MntGoldProds"         "NumDealsPurchases"    "NumWebPurchases"
## [13] "NumCatalogPurchases"  "NumStorePurchases"    "NumWebVisitsMonth"
## [16] "AcceptedCmp3"         "AcceptedCmp4"         "AcceptedCmp5"
## [19] "AcceptedCmp1"         "AcceptedCmp2"         "Complain"
## [22] "Z_CostContact"        "Z_Revenue"            "Response"
## [25] "Age"                  "Customer_Days"        "marital_Divorced"
## [28] "marital_Married"      "marital_Single"       "marital_Together"
## [31] "marital_Widow"        "education_2n.Cycle"   "education_Basic"
## [34] "education_Graduation" "education_Master"     "education_PhD"
## [37] "MntTotal"             "MntRegularProds"      "AcceptedCmpOverall"
```

Income: C'est le salaire que génére les clients annuellement

KidHome: 1 si le client a des enfants 0 sinon

TeenHome: 1 si le client a des enfants adolescents 0 sinon

Recency: Nombre de jours depuis le dernier achat

MntWines: Montant dépensé sur le vin sur 2 ans

MntFruits: Montant dépensé sur les fruits sur 2 ans

MntMeatProducts: Montant dépensé sur la viande sur 2 ans

MntMeatProducts: Montant dépensé sur la viande sur 2 ans

MntFishProducts: Montant dépensé sur les poissons sur 2 ans

MntSweetProducts: Montant dépensé sur les produits sucrés sur 2 ans

NumDealsPurchases: Nombre d'achats avec solde

NumWebPurchases: Nombre d'achats fait sur le site de l'entreprise

NumCatalogPurchases: Nombre d'achats en utilisant un catalogue

NumStorePurchases: Nombre d'achats fait directement en magasin

NumWebVisitsMonth: Nombre de visites dans le site de l'entreprise le dernier mois

AcceptedCmp1: 1 si le client accepte l'offre à la première publicité, 0 sinon

AcceptedCmp2: 1 si le client accepte l'offre à la deuxième publicité, 0 sinon

AcceptedCmp3: 1 si le client accepte l'offre à la troisième publicité, 0 sinon

AcceptedCmp4: 1 si le client accepte l'offre à la quatrième publicité, 0 sinon

AcceptedCmp5: 1 si le client accepte l'offre à la cinquième publicité, 0 sinon

AcceptedCmpOverall: Nombre total de publicités acceptées (accepter l'offre grâce à une publicité)

Response: 1 si le client accepte l'offre à la dernière publicité, 0 sinon

Complain: 1 si le client se plaint sur les deux dernières années

DtCustomer: date d'inscription du client auprès de l'entreprise

Customer_Days: nombre de jours depuis l'inscription en tant que client

education_2n Cycle: le client a fait des études secondaires

education_Basic: le client a une éducation de base

education_Graduation: le client a une license

education_Master: le client a un master

education_PhD: le client a un doctorat

marital_Divorced: 1 si le client est divorcé, 0 sinon

martial_Married: 1 si le client est marié, 0 sinon

marital_Single: 1 si le client est célibataire, 0 sinon

marital_Together: 1 si le client est dans une relation, 0 sinon

marital_Widow: 1 si le client est veuf, 0 sinon

Z_CostContact: ?

Z_Revenue: ?

Age: Age du client

MntTotal: Montant total dépensé par le client

MntRegularProds: ?

Nettoyage des données

On va tout d'abord voir s'il y a des valeurs manquantes sur la base de données

```
colSums(is.na(data))
```

```
##               Income              Kidhome             Teenhome
##                    0                    0                    0
##              Recency             MntWines            MntFruits
##                    0                    0                    0
##      MntMeatProducts      MntFishProducts     MntSweetProducts
##                    0                    0                    0
##         MntGoldProds    NumDealsPurchases      NumWebPurchases
##                    0                    0                    0
##  NumCatalogPurchases    NumStorePurchases    NumWebVisitsMonth
##                    0                    0                    0
##         AcceptedCmp3         AcceptedCmp4         AcceptedCmp5
##                    0                    0                    0
##         AcceptedCmp1         AcceptedCmp2             Complain
##                    0                    0                    0
##        Z_CostContact            Z_Revenue             Response
##                    0                    0                    0
##                  Age        Customer_Days     marital_Divorced
##                    0                    0                    0
##      marital_Married       marital_Single     marital_Together
##                    0                    0                    0
##        marital_Widow   education_2n.Cycle      education_Basic
##                    0                    0                    0
## education_Graduation     education_Master        education_PhD
##                    0                    0                    0
##             MntTotal      MntRegularProds   AcceptedCmpOverall
##                    0                    0                    0
```

Il n'y a donc pas de valeurs manquantes.

```
Unique <- sapply(data, function(x) nlevels(as.factor(x)))
```

La commande sapply nous donne le nombre de valeurs qu'il y a dans chaque colonne en enlevant les valeurs qui se répètent.

```
Unique
```

```
##               Income              Kidhome             Teenhome
##                 1963                    3                    3
##              Recency             MntWines            MntFruits
##                  100                  775                  158
##      MntMeatProducts      MntFishProducts     MntSweetProducts
##                  551                  182                  176
##         MntGoldProds    NumDealsPurchases      NumWebPurchases
##                  212                   15                   15
##  NumCatalogPurchases    NumStorePurchases    NumWebVisitsMonth
##                   13                   14                   16
##         AcceptedCmp3         AcceptedCmp4         AcceptedCmp5
##                    2                    2                    2
##         AcceptedCmp1         AcceptedCmp2             Complain
##                    2                    2                    2
##        Z_CostContact            Z_Revenue             Response
```

```
##                     1                     1                     2
##                   Age         Customer_Days      marital_Divorced
##                    56                   662                     2
##       marital_Married        marital_Single      marital_Together
##                     2                     2                     2
##         marital_Widow    education_2n.Cycle       education_Basic
##                     2                     2                     2
## education_Graduation      education_Master         education_PhD
##                     2                     2                     2
##              MntTotal       MntRegularProds    AcceptedCmpOverall
##                   897                   974                     5
```

```
Unique[Unique==1]
```

```
## Z_CostContact     Z_Revenue
##             1             1
```

On constate que les colonnes Z_CostContact et Z_revenue ne contiennent qu'une seule valeur, c'est à dire que tous les clients du dataset ont les mêmes valeurs dans ces deux colonnes. Ces deux colonnes ne servent donc à rien pour l'application d'un modèle de machine learning. On peut donc les enlever du dataset.

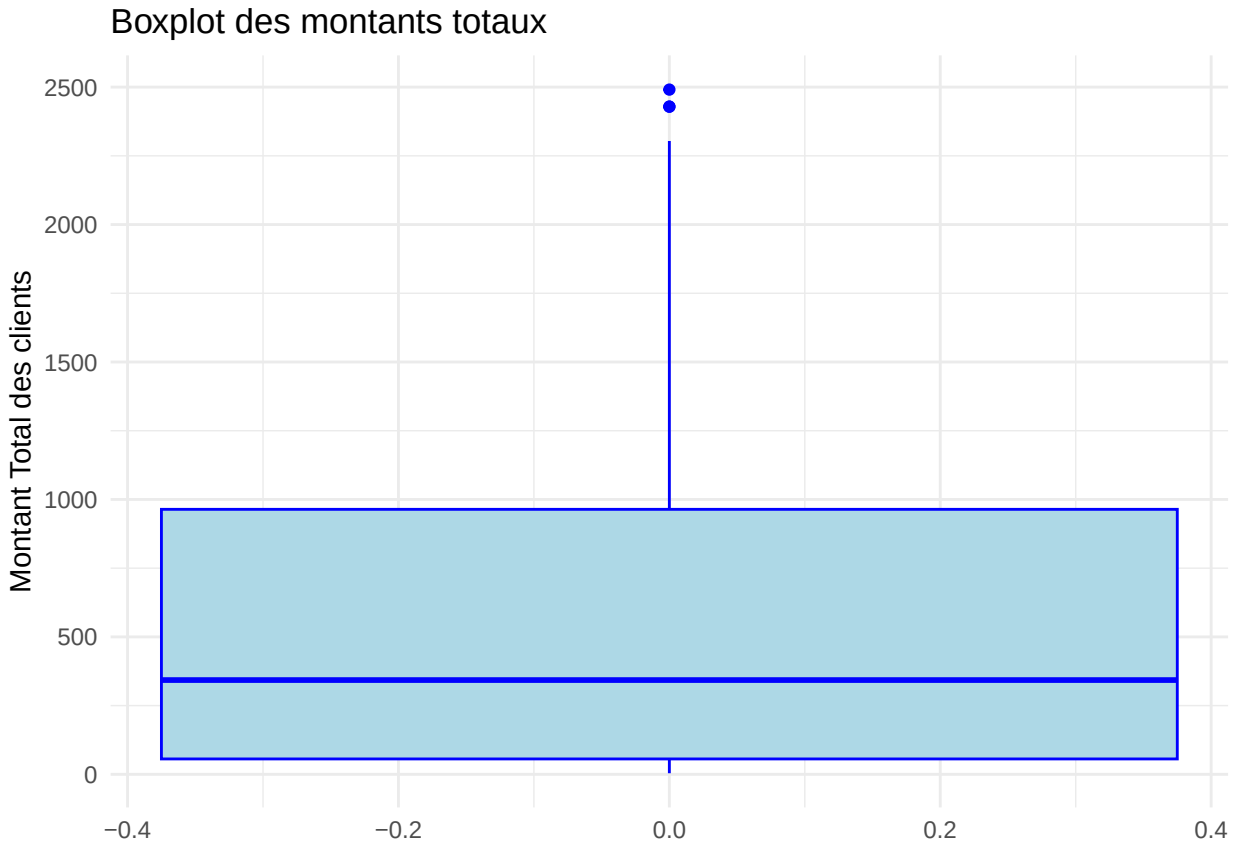
```
data<- subset(data, select= -c(Z_CostContact, Z_Revenue))
```

```
dim(data) #On a donc bien enlever ces deux colonnes
```

```
## [1] 2205   37
```

```
library(ggplot2) #On charge la librarie ggplot
```

```
ggplot(data, aes(y = MntTotal)) +
  geom_boxplot(fill = "lightblue", color = "blue") +
  labs(title = "Boxplot des montants totaux", y = "Montant Total des clients") +
  theme_minimal()
```



On remarque que moins de 25% des clients ont dépensé moins de 200

50% des clients ont dépensé moins de 400

75% des clients ont dépensé moins de 1000 ‘ Puis, il y a une très petite minorité de client qui ont dépensé plus de 2000

On observe également la présence d'outliers, on va les traiter grâce à la méthode des quantiles.

Si un individu de la base de donnée a une valeur plus petite que

$$quantile(0.25) - 1.5 * (quantile(0.75) - quantile(0.25))$$

alors c'est un outlier: il a une valeur extrêmement basse par rapport aux autres individus de la base de donnée. De même si un individu a une valleur plus grande que

$$quantile(0.75) + 1.5 * (quantile(0.75) - quantile(0.25))$$

, alors c'est un outlier: il a une valeur extrêmement grande par rapport aux autres individus de la base de données.

```
Q1<-quantile(data$MntTotal, 0.25)
Q3<-quantile(data$MntTotal, 0.75)
lower_bound<-Q1-1.5*(Q3-Q1)
upper_bound<-Q3+1.5*(Q3-Q1)
outliers<- data[data$MntTotal < lower_bound | data$MntTotal > upper_bound,]
outliers
```

```
##      Income Kidhome Teenhome Recency MntWines MntFruits MntMeatProducts
## 1160  90638       0        0      29     1156       120             915
## 1468  87679       0        0      62     1259       172             815
```

```
## 1548  90638       0        0      29     1156       120             915
##      MntFishProducts MntSweetProducts MntGoldProds NumDealsPurchases
## 1160              94              144          96                 1
## 1468              97              148          33                 1
## 1548              94              144          96                 1
##      NumWebPurchases NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
## 1160               3                   4                10                 1
## 1468               7                  11                10                 4
## 1548               3                   4                10                 1
##      AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain
## 1160            0            0            1            0            0        0
## 1468            1            0            1            1            0        0
## 1548            0            0            1            0            0        0
##      Response Age Customer_Days marital_Divorced marital_Married marital_Single
## 1160        0  29          2295                0              0              1
## 1468        1  32          2496                0              0              0
## 1548        1  29          2295                0              0              1
##      marital_Together marital_Widow education_2n.Cycle education_Basic
## 1160                0             0                 0               0
## 1468                1             0                 0               0
## 1548                0             0                 0               0
##      education_Graduation education_Master education_PhD MntTotal
## 1160                    0                1             0     2429
## 1468                    1                0             0     2491
## 1548                    0                1             0     2429
##      MntRegularProds AcceptedCmpOverall
## 1160            2333                  1
## 1468            2458                  3
## 1548            2333                  1
```

```
#ces outliers vont être enlevés de la base de données car ils ne sont
#pas représentatifs de la tendance des dépenses des clients
#Sur à peu près 2000 clients dans la base de données seulement 3 outliers apparaissent
#Le montant des dépenses de ces clients sont bien au dessus ou bien en dessous
#des dépenses des autres clients qui sont majoritaires, et en marketing on ne
#peut pas se focaliser sur chaque client un à un, il faut plutôt segmenter
#en groupe de population
```

On va maintenant enlever ces 3 outliers

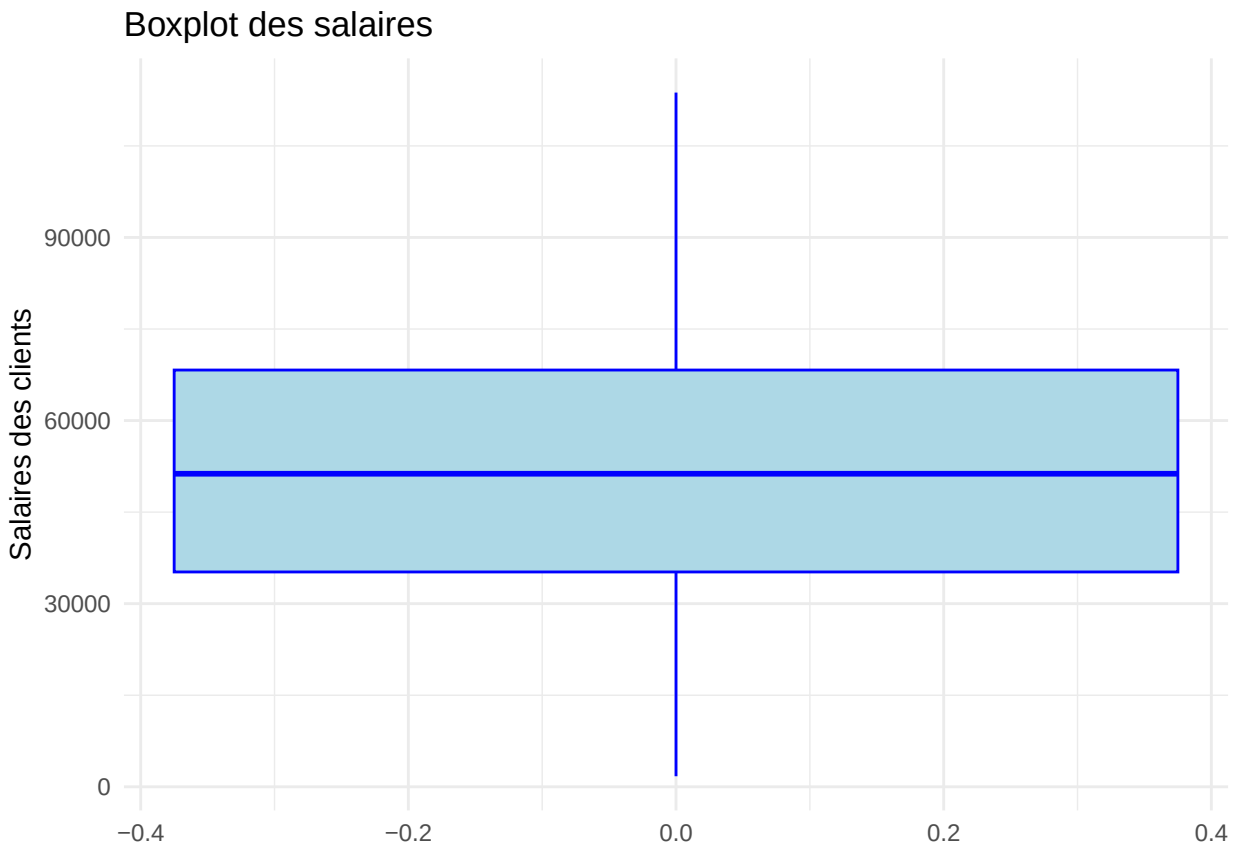
```
data<- data[data$MntTotal > lower_bound | data["MntTotal"] < upper_bound,]
summary(data)
```

```
##      Income          Kidhome          Teenhome         Recency
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00
##  1st Qu.: 35196   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:24.00
##  Median : 51287   Median :0.0000   Median :0.0000   Median :49.00
##  Mean   : 51622   Mean   :0.4422   Mean   :0.5066   Mean   :49.01
##  3rd Qu.: 68281   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:74.00
##  Max.   :113734   Max.   :2.0000   Max.   :2.0000   Max.   :99.00
##     MntWines        MntFruits     MntMeatProducts  MntFishProducts
##  Min.   :   0.0   Min.   :  0.0   Min.   :   0.0   Min.   :  0.00
##  1st Qu.:  24.0   1st Qu.:  2.0   1st Qu.:  16.0   1st Qu.:  3.00
##  Median : 178.0   Median :  8.0   Median :  68.0   Median : 12.00
##  Mean   : 306.2   Mean   : 26.4   Mean   : 165.3   Mean   : 37.76
```

```
##  3rd Qu.: 507.0   3rd Qu.: 33.0   3rd Qu.: 232.0   3rd Qu.: 50.00
##  Max.   :1493.0   Max.   :199.0   Max.   :1725.0   Max.   :259.00
##  MntSweetProducts  MntGoldProds    NumDealsPurchases NumWebPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   : 0.000    Min.   : 0.000
##  1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000    1st Qu.: 2.000
##  Median :  8.00   Median : 25.00   Median : 2.000    Median : 4.000
##  Mean   : 27.13   Mean   : 44.06   Mean   : 2.318    Mean   : 4.101
##  3rd Qu.: 34.00   3rd Qu.: 56.00   3rd Qu.: 3.000    3rd Qu.: 6.000
##  Max.   :262.00   Max.   :321.00   Max.   :15.000    Max.   :27.000
##  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth  AcceptedCmp3
##  Min.   : 0.000      Min.   : 0.000    Min.   : 0.000    Min.   :0.00000
##  1st Qu.: 0.000      1st Qu.: 3.000    1st Qu.: 3.000    1st Qu.:0.00000
##  Median : 2.000      Median : 5.000    Median : 6.000    Median :0.00000
##  Mean   : 2.645      Mean   : 5.824    Mean   : 5.337    Mean   :0.07392
##  3rd Qu.: 4.000      3rd Qu.: 8.000    3rd Qu.: 7.000    3rd Qu.:0.00000
##  Max.   :28.000      Max.   :13.000    Max.   :20.000    Max.   :1.00000
##   AcceptedCmp4      AcceptedCmp5      AcceptedCmp1      AcceptedCmp2
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.07438   Mean   :0.07302   Mean   :0.0644   Mean   :0.01361
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##     Complain          Response          Age        Customer_Days
##  Min.   :0.00000   Min.   :0.000   Min.   :24.0   Min.   :2159
##  1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:43.0   1st Qu.:2339
##  Median :0.00000   Median :0.000   Median :50.0   Median :2515
##  Mean   :0.00907   Mean   :0.151   Mean   :51.1   Mean   :2513
##  3rd Qu.:0.00000   3rd Qu.:0.000   3rd Qu.:61.0   3rd Qu.:2688
##  Max.   :1.00000   Max.   :1.000   Max.   :80.0   Max.   :2858
##  marital_Divorced marital_Married  marital_Single   marital_Together
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1043   Mean   :0.3873   Mean   :0.2163   Mean   :0.2576
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  marital_Widow     education_2n.Cycle education_Basic   education_Graduation
##  Min.   :0.00000   Min.   :0.0000     Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000     Median :0.00000   Median :1.0000
##  Mean   :0.03447   Mean   :0.0898     Mean   :0.02449   Mean   :0.5048
##  3rd Qu.:0.00000   3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000     Max.   :1.00000   Max.   :1.0000
##  education_Master education_PhD       MntTotal      MntRegularProds
##  Min.   :0.0000   Min.   :0.0000   Min.   :   4.0   Min.   :-283.0
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  56.0   1st Qu.:  42.0
##  Median :0.0000   Median :0.0000   Median : 343.0   Median : 288.0
##  Mean   :0.1651   Mean   :0.2159   Mean   : 562.8   Mean   : 518.7
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 964.0   3rd Qu.: 884.0
##  Max.   :1.0000   Max.   :1.0000   Max.   :2491.0   Max.   :2458.0
##  AcceptedCmpOverall
##  Min.   :0.0000
##  1st Qu.:0.0000
```

```
##  Median :0.0000
##  Mean   :0.2993
##  3rd Qu.:0.0000
##  Max.   :4.0000
```

```
ggplot(data, aes(y = Income)) +
  geom_boxplot(fill = "lightblue", color = "blue") +
  labs(title = "Boxplot des salaires", y = "Salaires des clients") +
  theme_minimal()
```



On remarque que moins de 25% des clients ont des revenus inférieures à 35000

50% des clients ont des revenus inférieures à 50000

75% des clients ont des revenus inférieures à 65000

On va voir avec la même méthode qu'avant si il y a des outliers pour les salaires

```
Q1<-quantile(data$Income, 0.25)
Q3<-quantile(data$Income, 0.75)
lower_bound<-Q1-1.5*(Q3-Q1)
upper_bound<-Q3+1.5*(Q3-Q1)
outliers<- data[data$Income < lower_bound | data$Income > upper_bound,]
outliers
```

```
##  [1] Income              Kidhome             Teenhome
##  [4] Recency             MntWines            MntFruits
##  [7] MntMeatProducts     MntFishProducts     MntSweetProducts
## [10] MntGoldProds        NumDealsPurchases   NumWebPurchases
## [13] NumCatalogPurchases NumStorePurchases   NumWebVisitsMonth
## [16] AcceptedCmp3        AcceptedCmp4        AcceptedCmp5
```

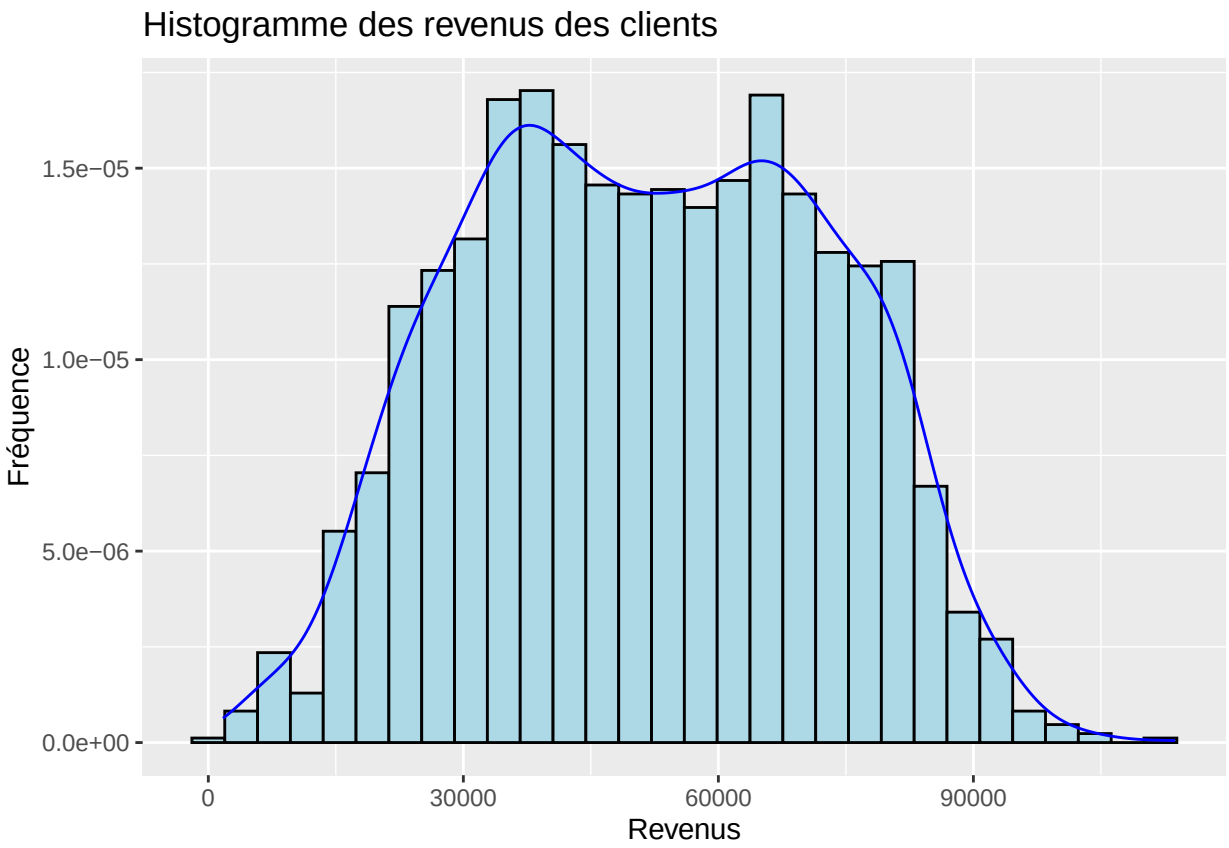
```
## [19] AcceptedCmp1          AcceptedCmp2          Complain
## [22] Response              Age                   Customer_Days
## [25] marital_Divorced      marital_Married       marital_Single
## [28] marital_Together      marital_Widow         education_2n.Cycle
## [31] education_Basic       education_Graduation  education_Master
## [34] education_PhD         MntTotal              MntRegularProds
## [37] AcceptedCmpOverall
## <0 rows> (or 0-length row.names)
```

```
#Pas d'outliers
```

```
ggplot(data, aes(x= Income )) +
  geom_histogram(fill = "lightblue", color = "black",aes(y = ..density..) ) +
  geom_density(color = "blue") +
  labs(title = "Histogramme des revenus des clients", x= "Revenus", y="Fréquence")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



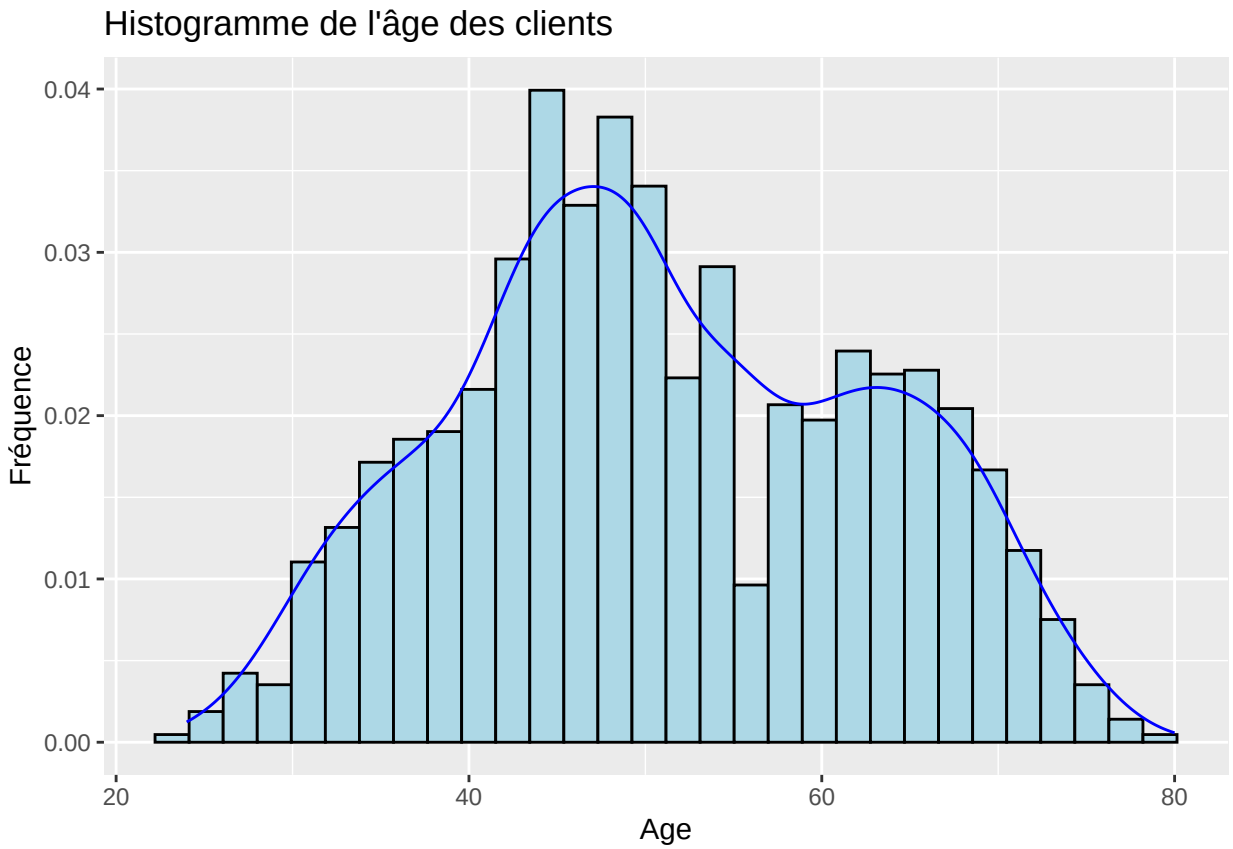
Les revenus des clients semblent suivre une loi normale

On remarque que la grande majorité des clients ont des revenus compris entre 35000 et 75000

```
ggplot(data, aes(x= Age )) +
  geom_histogram(fill = "lightblue", color = "black",aes(y = ..density..) ) +
  geom_density(color = "blue") +
```

```
  labs(title = "Histogramme de l'âge des clients", x= "Age", y="Fréquence")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



On remarque que les personnes âgées entre 40 et 50 ans sont les clients les plus majoritaires.

Même les clients qui ont la trentaine ou la soixantaine sont assez nombreux mais ne représentent cependant pas la majorité des clients.

```
library(e1071) #On charge cette librairie pour accès aux fonctions skewness et kurtosis
```

```
cat("Moyenne de l'age des clients:", mean(data$Age), "\n")
```

```
## Moyenne de l'age des clients: 51.09569
```

```
cat("Skewness de l'age des clients:", skewness(data$Age), "\n")
```

```
## Skewness de l'age des clients: 0.08981848
```

```
cat("Kurtosis de l'age des clients:", kurtosis(data$Age), "\n")
```

```
## Kurtosis de l'age des clients: -0.7999462
```

```
cat("Médiane de l'age des clients: ", quantile(data$Age, 0.5))
```

```
## Médiane de l'age des clients:  50
```

On a une skewness strictement positive mais proche de 0, donc la distribution de l'âge est légérement asymétrique vers la droite, cela peut s'expliquer certainement par le fait qu'il y a certains clients assez agés.

Et un kurtosis négatif légérement proche de 0 implique que la distribution de l'âge se rapproche d'une loi normale avec les extrémités de la distribution de l'âge qui converge vite vers 0 plus vite que la loi normale.

Ceci s'explique par le fait qu'il y a très peu de clients jeunes (la vingtaine) et très peu de clients très agés (80 ans et plus).

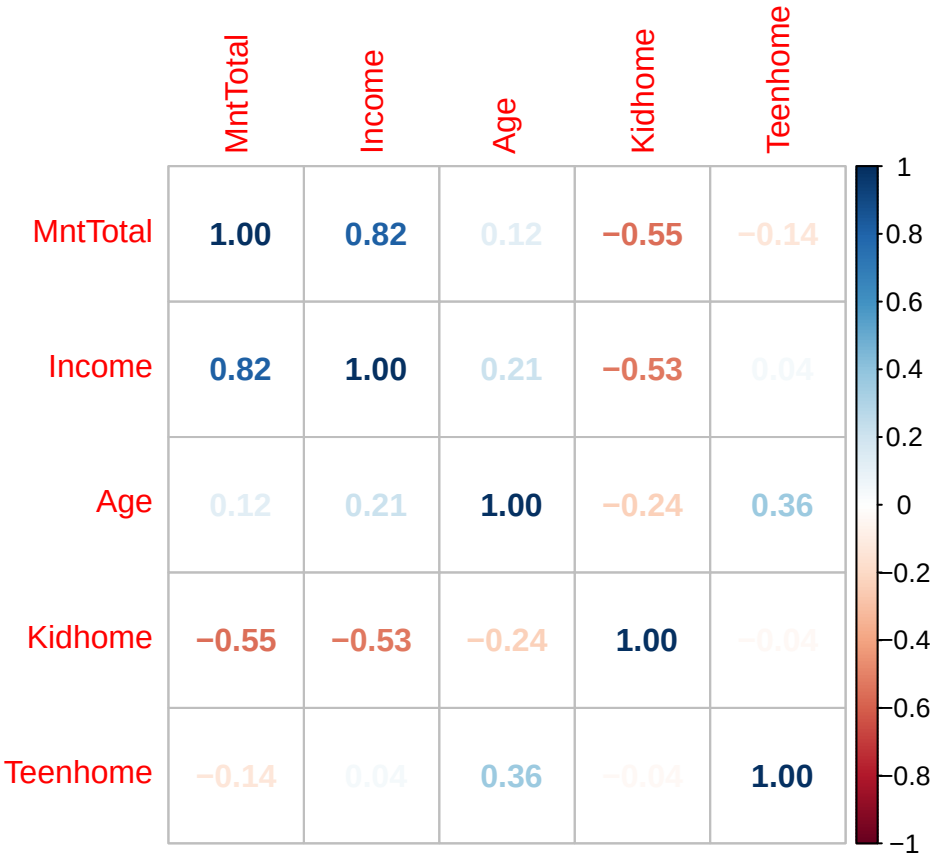
Corrélation des variables

```
#On trie les colonnes du dataset en fonction de leurs caractéristiques

cols_demographics <- c('Income','Age')
cols_children <- c('Kidhome', 'Teenhome')
cols_marital <- c('marital_Divorced', 'marital_Married','marital_Single',
                'marital_Together', 'marital_Widow')
cols_mnt <-c('MntTotal', 'MntRegularProds','MntWines', 'MntFruits',
            'MntMeatProducts', 'MntFishProducts',
            'MntSweetProducts', 'MntGoldProds')
cols_communication <- c('Complain', 'Response', 'Customer_Days')
cols_campaigns <- c('AcceptedCmpOverall', 'AcceptedCmp1', 'AcceptedCmp2',
                  'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5')
cols_source_of_purchase <- c('NumDealsPurchases', 'NumWebPurchases',
                           'NumCatalogPurchases', 'NumStorePurchases',
                           'NumWebVisitsMonth')
cols_education <- c('education_2n Cycle', 'education_Basic',
                  'education_Graduation', 'education_Master', 'education_PhD')
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
cor_matrix<- cor(data[c(c("MntTotal"), cols_demographics, cols_children)])
cor_matrix<-round(cor_matrix, 2)
corrplot(cor_matrix, method="number")
```



On remarque que le montant total du client est fortement corrélé positivement avec le revenu du client donc on peut en déduire que lorsque les revenus du client augmentent et plus le montant total du client augmente.

On remarque aussi que le montant total du client est moyennement corrélé négativement avec le nombre d'enfants à la maison.

Cela impliquerait que plus il y a d'enfants dans la famille du client et plus le montant total des dépenses du client diminue.

Cela parait assez étrange, mais on peut trouver plusieurs explications sociologiques.

1ère explication: Dans de nombreuses familles, les coûts liés aux enfants,tels que l'alimentation, l'habillement et les loisirs, peuvent être partagés entre les enfants.

Par conséquent, à mesure que le nombre d'enfants augmente, les dépenses individuelles par enfant peuvent diminuer, ce qui entraîne une baisse des dépenses totales.

2ème explication: Les familles avec plus d'enfants peuvent adopter des styles de vie différents, tels que des achats en vrac ou des choix budgétaires plus stricts, ce qui peut réduire les dépenses globales.

Les parents peuvent être plus enclins à rechercher des offres et à limiter les dépenses superflues.

3ème explication: Les familles avec plus d'enfants peuvent avoir des priorités financières différentes. Par exemple, elles peuvent consacrer une plus grande partie de leur budget aux besoins de base tels que l'éducation et la santé, ce qui réduit les dépenses dans d'autres domaines.

Maintenant, on va voir la corrélation qu'il y a entre la variable MntTotal et les variables qualitatifs du dataset (ici booléenne).

Pour calculer ce genre de corrélation, on utilise la corrélation de Point biséral.

```r
cor.test(data$marital_Divorced, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$marital_Divorced and data$MntTotal
## t = 0.1749, df = 2203, p-value = 0.8612
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03802287  0.04546261
## sample estimates:
##         cor
## 0.003726364
```

```r
cor.test(data$marital_Married, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$marital_Married and data$MntTotal
## t = -1.0446, df = 2203, p-value = 0.2963
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06393487  0.01951051
## sample estimates:
##         cor
## -0.02225093
```

```r
cor.test(data$marital_Single, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$marital_Single and data$MntTotal
## t = 0.27881, df = 2203, p-value = 0.7804
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03581212  0.04767158
## sample estimates:
##         cor
## 0.005940081
```

```r
cor.test(data$marital_Together, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$marital_Together and data$MntTotal
## t = 0.076139, df = 2203, p-value = 0.9393
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04012385  0.04336257
## sample estimates:
##        cor
## 0.00162219
```

```r
cor.test(data$marital_Widow, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$marital_Widow and data$MntTotal
## t = 1.685, df = 2203, p-value = 0.09212
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.005874454  0.077504909
## sample estimates:
##        cor
## 0.03587766
```

On remarque une corrélation bisérale très faible (proche de 0) entre les montants totaux dépensés par les clients et leur situation maritale (divorce, marriage, célibat, couple non marié, veuve...).

Par exemple, les montants totaux dépensés en moyenne par les clients veufs ne différent pas beaucoup des montants dépensés moyens des clients "non-veufs".

Le raisonnement est le même pour chaque statut marital.

```r
cor.test(data$education_2n.Cycle, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$education_2n.Cycle and data$MntTotal
## t = -2.8341, df = 2203, p-value = 0.004638
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.10175893 -0.01857504
## sample estimates:
##         cor
## -0.06027163
```

```r
cor.test(data$education_Basic, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$education_Basic and data$MntTotal
## t = -6.5702, df = 2203, p-value = 6.249e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.17933636 -0.09745146
## sample estimates:
##        cor
## -0.1386308
```

```r
cor.test(data$education_Graduation, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$education_Graduation and data$MntTotal
## t = 0.67276, df = 2203, p-value = 0.5012
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.02742765  0.05604187
## sample estimates:
##       cor
## 0.01433208
```

```
cor.test(data$education_Master, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$education_Master and data$MntTotal
## t = 0.2986, df = 2203, p-value = 0.7653
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03539098  0.04809228
## sample estimates:
##         cor
## 0.006361735
```

```
cor.test(data$education_PhD, data$MntTotal)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$education_PhD and data$MntTotal
## t = 3.3316, df = 2203, p-value = 0.0008776
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.02914726 0.11221608
## sample estimates:
##       cor
## 0.07080443
```

On remarque une corrélation bisérale très faible (proche de 0) entre les montants totaux dépensés par les clients et leur niveau d'éducation (Licence, Lycée, Master, doctorat).

Par exemple les montants dépensés en moyenne par les doctorants est très proche du montant moyen dépensé par les étudiants n'ayant pas effectués un doctorat.

Le raisonnement est le même pour niveau d'éducation.

Feature engineering

```
get_marital_status <- function(row) {
  ifelse(row["marital_Divorced"] == 1, "Divorced",
  ifelse(row["marital_Married"] == 1, "Married",
  ifelse(row["marital_Single"] == 1, "Single",
  ifelse(row["marital_Together"] == 1, "Together",
  ifelse(row["marital_Widow"] == 1, "Widow",
  "Unknown")))))
}
```

```
data$Marital <- apply(data, 1, get_marital_status) #On crée une nouvelle feature
```

```r
count_widow <- sum(data$Marital == "Widow")
count_together <- sum(data$Marital == "Together")
count_divorced <- sum(data$Marital == "Divorced")
count_married <- sum(data$Marital == "Married")
count_single <- sum(data$Marital == "Single")

cat("Il y a", count_widow, "veufs ou veuves parmi les clients\n")
```

```
## Il y a 76 veufs ou veuves parmi les clients
```

```r
cat("Il y a", count_together, "en couple parmi les clients\n")
```

```
## Il y a 568 en couple parmi les clients
```

```r
cat("Il y a", count_divorced, "divorcés parmi les clients\n")
```

```
## Il y a 230 divorcés parmi les clients
```

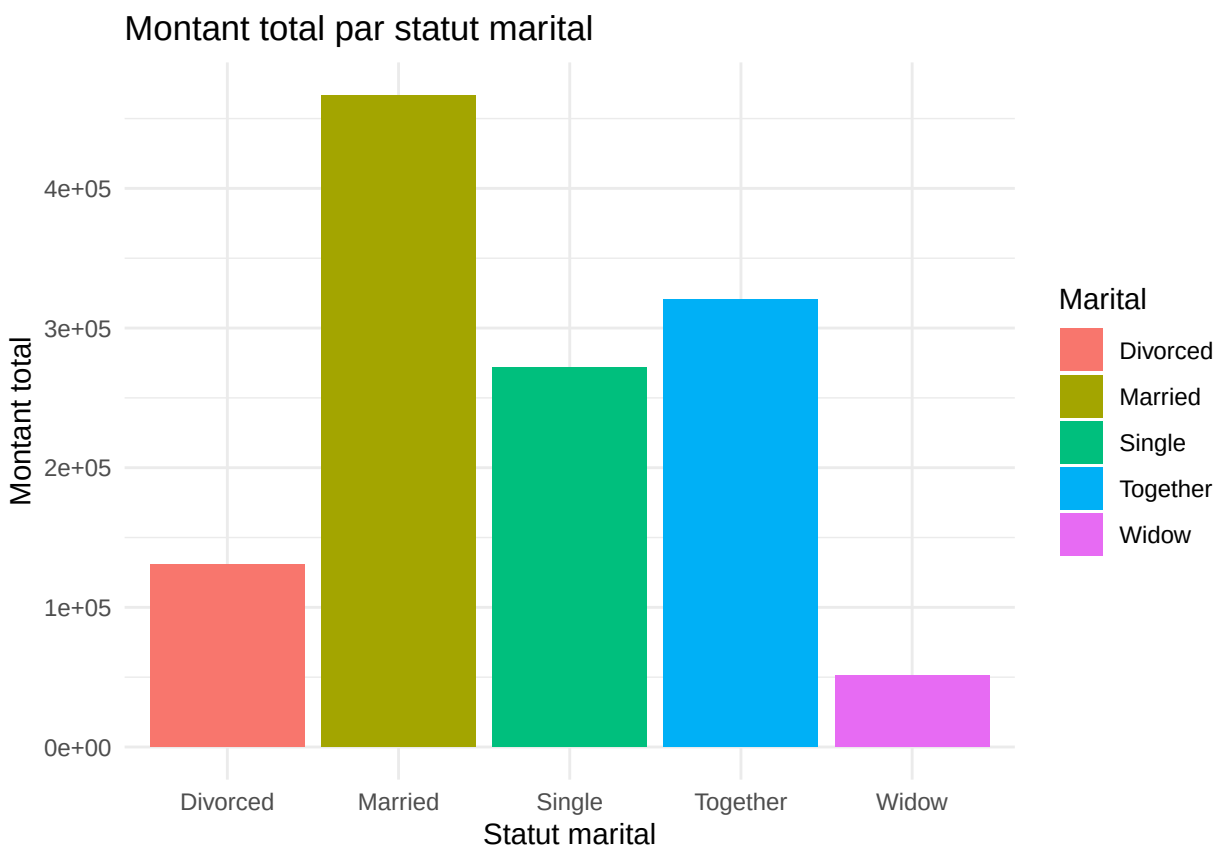
```r
cat("Il y a", count_married, "mariés parmi les clients\n")
```

```
## Il y a 854 mariés parmi les clients
```

```r
cat("Il y a", count_single, "célibataires parmi les clients\n")
```

```
## Il y a 477 célibataires parmi les clients
```

```r
ggplot(data, aes(x = Marital, y = MntTotal, fill = Marital)) +
  geom_bar(stat= 'identity') +
  labs(title = "Montant total par statut marital", x = "Statut marital", y = "Montant total") +
  theme_minimal()
```



```r
in_relationship <- function(row) {
  ifelse(row["marital_Married"] == 1, 1,
  ifelse(row["marital_Together"] == 1, 1,
  0))
}
```

```r
data$In_relationship <- apply(data, 1, in_relationship) #On crée une nouvelle feature, qui vaut 1 si le
```

Modèle de KMeans

On va maintenant classer les clients en plusieurs groupes, on va effectuer un clustering en utilisant seulement 3 variables de notre datasets: Le salaire, le montant total et la variable In_relationship. Dans le monde du marketing, ce sont les critères les plus primordiaux pour effectuer une segmentation, car ça nous apporte les meilleurs informations pour comprendre les comportement de certains clients.

```r
# Chargement du package dplyr
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
```

```r
#On va s'intéresser à ces trois variables
cols_for_clustering <- c("Income", "MntTotal", "In_relationship")
data_scaled <- data

# Normalisation des colonnes
data_scaled[cols_for_clustering] <- scale(data_scaled[cols_for_clustering])
#On normalise les données pour mettre à l'échelle nos variables;
#Normaliser les données permet une meilleure efficacité de l'algorithme KMeans, il améliore la compléxi
```

```r
summary(data_scaled[cols_for_clustering])
```

```
##      Income            MntTotal       In_relationship
##  Min.   :-2.40873   Min.   :-0.9702   Min.   :-1.3473
##  1st Qu.:-0.79303   1st Qu.:-0.8799   1st Qu.:-1.3473
##  Median :-0.01618   Median :-0.3816   Median : 0.7419
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.80427   3rd Qu.: 0.6967   3rd Qu.: 0.7419
##  Max.   : 2.99868   Max.   : 3.3480   Max.   : 0.7419
```

On va maintenent effectuer une ACP sur 2 composantes, cela va nous permettre de mieux visualiser le résultat du clustering.

Lorsqu'on a un grand nombre de caractéristiques (variables) dans nos données, l'ACP peut réduire la dimensionnalité en créant de nouvelles variables (composantes principales) qui capturent la majeure partie de la variance des données. Cette réduction de dimensionnalité peut rendre plus facile la visualisation des données, ce qui peut être utile pour observer les clusters formés par le K-Means ou d'autres algorithmes de clustering.

Après avoir effectué l'ACP, on peut choisir de visualiser les données dans un espace de dimension réduite en fonction des composantes principales. Cette visualisation peut aider à identifier des tendances, des structures ou des clusters potentiels dans les données. Cela peut être particulièrement utile lorsque vos données d'origine ont un grand nombre de dimensions et qu'il est difficile de les représenter graphiquement.

```
pca_res<- prcomp(data_scaled[cols_for_clustering], center= TRUE, scale.= TRUE, rank. =2 )

data_scaled$pc1<- pca_res$x[,1] #1ère composante principale
data_scaled$pc2<- pca_res$x[,2] #2ème composante principale
```

On va maintenant trouver le meilleur hyperparamètre K, c'est à dire le meilleur nombre de cluster à choisir pour le KMeans. On va étudier pour cela deux méthodes: Elbow Method et Silhouette Score

Elbow Method:

La méthode du coude (Elbow method) est une technique couramment utilisée pour déterminer le nombre optimal de clusters dans une analyse de clustering, comme le K-Means. Mathématiquement, cette méthode implique le calcul de la somme des carrés intra-cluster (WCSS, Within-Cluster Sum of Squares) pour différentes valeurs de K (nombre de clusters) et la recherche du "coude" dans le graphique de WCSS par rapport à k, ce qui indique un point d'inflexion où l'ajout de clusters supplémentaires n'apporte pas beaucoup d'amélioration dans la réduction des erreurs.

Voici la formule mathématique pour le WCSS :

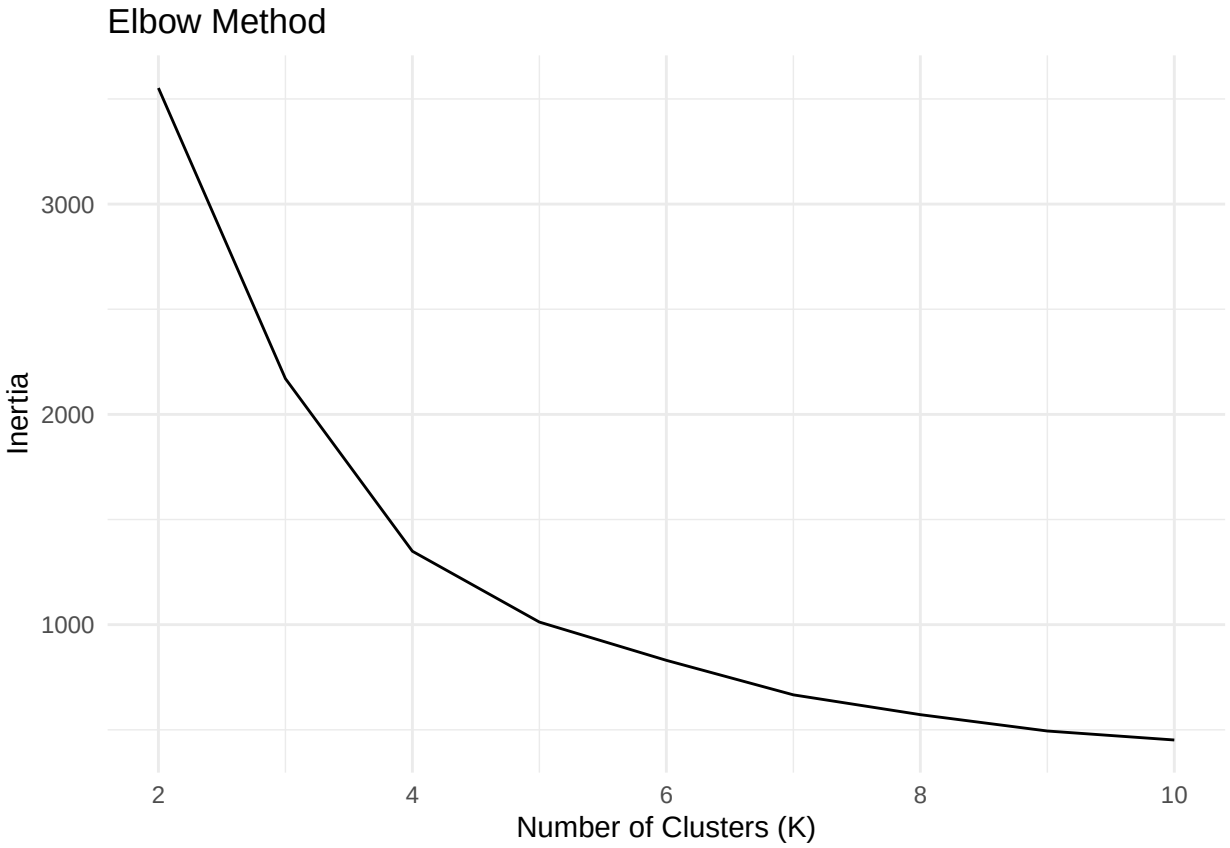
$$WCSS(K) = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

où K est le nombre clusters, C_i le i-ème cluster, x est un point de donnée et μ_i est le centroïde du i-ème cluster.

```
X <- data_scaled[cols_for_clustering]
inertia_list <- numeric()

# Boucle pour calculer l'inertie pour différentes valeurs de K
for (K in 2:10) {
  kmeans_result <- kmeans(X, centers = K, nstart = 10, iter.max = 100)
  inertia <- sum(kmeans_result$tot.withinss)
  inertia_list <- c(inertia_list, inertia)
}

print(inertia_list)
```

```
## [1] 3552.0691 2170.0704 1349.1919 1012.3704  830.3824  666.2435  571.9337
## [8]  494.2938  451.4488
```

```
inertia_data <- data.frame(K = 2:10, Inertia = inertia_list)

# Tracé du graphique
ggplot(inertia_data, aes(x = K, y = Inertia)) +
  geom_line() +
  labs(title = "Elbow Method",
       x = "Number of Clusters (K)",
       y = "Inertia") +
  theme_minimal()
```



On calcule l'inertie intra-cluster en fonction du nombre de clusters choisi pour le modèle.

On remarque que l'inertie baisse moins significativement à partir d'un nombre de cluster égale à 4, en général quand la courbe forme une sorte de coude comme on peut le voir dans la zone où le nombre de cluster égale à 4, cela veut dire en pratique que le nombre de cluster est optimal.

On remarque alors que 4 et 5 semble être les nombres de clusters optimaux.

On peut également choisir un nombre de cluster où l'inertie intra est plus petite comme par exemple un nombre de cluster égal à 7, mais il faut aussi prendre un nombre de cluster pas trop grand.

En effet, à la fin on a envie de segmenter plusieurs types de clients afin d'adapter les publicités, et les offres en fonction de ces différents groupes.

Si le nombre de cluster est trop grand on va devoir adapter des offres et des publicités pour de nombreux groupes de clients ce qui peut-être coûteux et fastidieux, en terme de publicité par exemple.

On se limitera donc à un nombre de cluster égale à 4 ou 5.

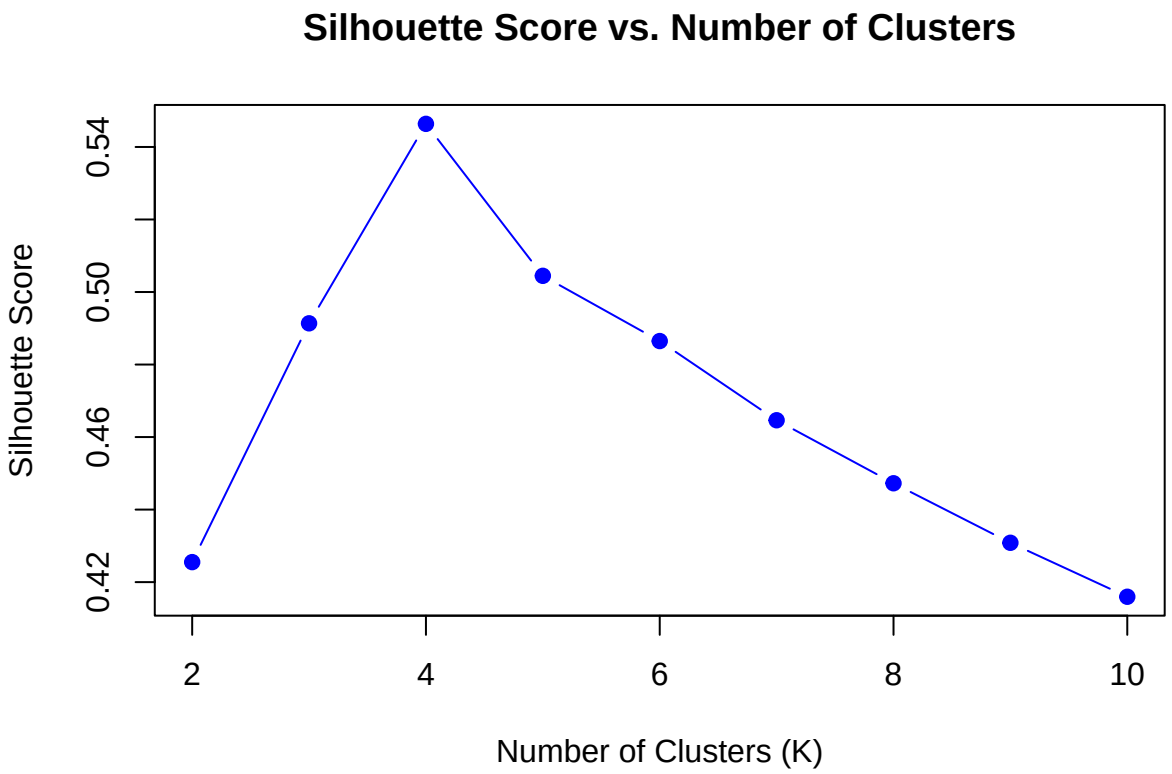
```
library(fpc)
```

silhouette score:

En partitionnement de données (clustering), le coefficient de silhouette est une mesure de qualité d'une partition d'un ensemble de données en classification automatique. Pour chaque point, son coefficient de silhouette est la différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation). Si cette différence est négative, le point est en moyenne plus proche du groupe voisin que du sien : il est donc mal classé. À l'inverse, si cette différence est positive, le point est en moyenne plus proche de son groupe que du groupe voisin : il est donc bien classé.

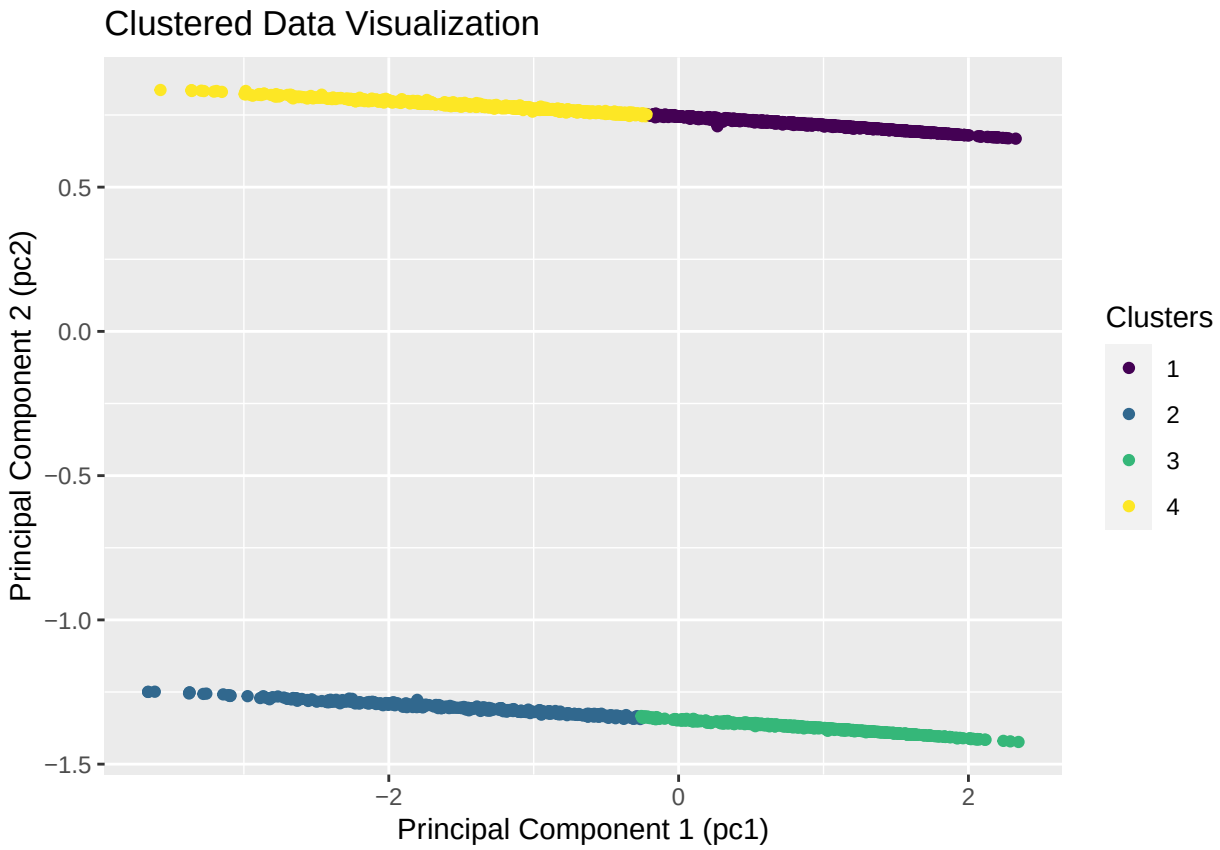
Le coefficient de silhouette proprement dit est la moyenne du coefficient de silhouette pour tous les points.

```r
silhouette_score_list <- numeric()

# Boucle pour calculer les scores de silhouette pour différentes valeurs de K
for (K in 2:10) {
  kmeans_result <- kmeans(X, centers = K, nstart = 10, algorithm = "Hartigan-Wong", trace = FALSE)
  silhouette <- cluster.stats(dist(X), kmeans_result$cluster)$avg.silwidth
  silhouette_score_list <- c(silhouette_score_list, silhouette)
}

# Tracé du graphique
plot(2:10, silhouette_score_list, type = "b", pch = 19, col = "blue", xlab = "Number of Clusters (K)", y
     main = "Silhouette Score vs. Number of Clusters")
```




```r
set.seed(123)
num_clusters <- 4
model_final <- kmeans(X, centers = num_clusters, nstart = 10, algorithm = "Hartigan-Wong")
cluster <- model_final$cluster
data_scaled$Cluster <- cluster
```

```r
ggplot(data_scaled, aes(x = pc1, y = pc2, color = factor(Cluster))) +
  geom_point() +
  labs(title = "Clustered Data Visualization",
       x = "Principal Component 1 (pc1)",    #On observe les clusters à l'aide de nos composantes princ
       y = "Principal Component 2 (pc2)") +
  scale_color_manual(values = viridis::viridis(4)) +
  guides(color = guide_legend(title = "Clusters"))
```




```
data$Cluster <- data_scaled$Cluster
data_means <- data %>%
  group_by(Cluster) %>%
  summarize(
    mean_Income = mean(Income),
    mean_MntTotal = mean(MntTotal),
    mean_In_relationship = mean(In_relationship)
  )
```

```
mnt_data <- data %>%
  group_by(Cluster) %>%
  summarise(across(all_of(cols_mnt), mean)) %>%
  ungroup()
```

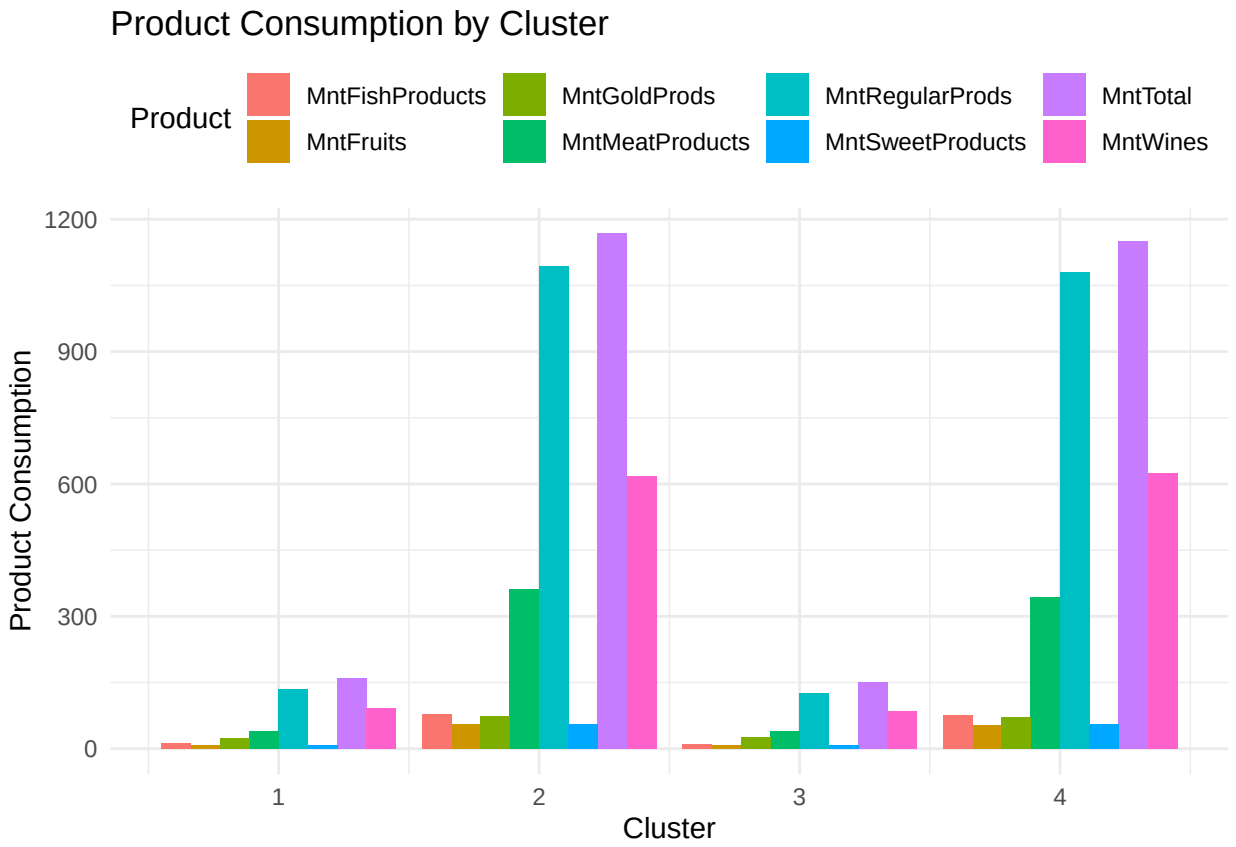
```
head(mnt_data)
```

```
## # A tibble: 4 x 9
##   Cluster MntTotal MntRegularProds MntWines MntFruits MntMeatProducts
##     <int>    <dbl>           <dbl>    <dbl>     <dbl>           <dbl>
## 1       1     159.            134.     92.4      7.66            39.4
## 2       2    1167.           1093.    617.      55.3            361.
## 3       3     151.            126.     85.5      7.83            38.8
## 4       4    1151.           1080.    625.      52.8            343.
## # i 3 more variables: MntFishProducts <dbl>, MntSweetProducts <dbl>,
## #   MntGoldProds <dbl>
```

```
library(tidyr)
```

```r
id_var <- "Cluster"
var_name <- "Product"
value_name <- "Consumption"
melted_data <- pivot_longer(mnt_data, cols = -all_of(id_var), names_to = var_name, values_to = value_nar
```

```r
ggplot(melted_data, aes(x = Cluster, y = Consumption, fill = Product)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Product Consumption by Cluster",
       x = "Cluster",
       y = "Product Consumption") +
  theme_minimal() +
  theme(legend.position = "top") +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5))
```



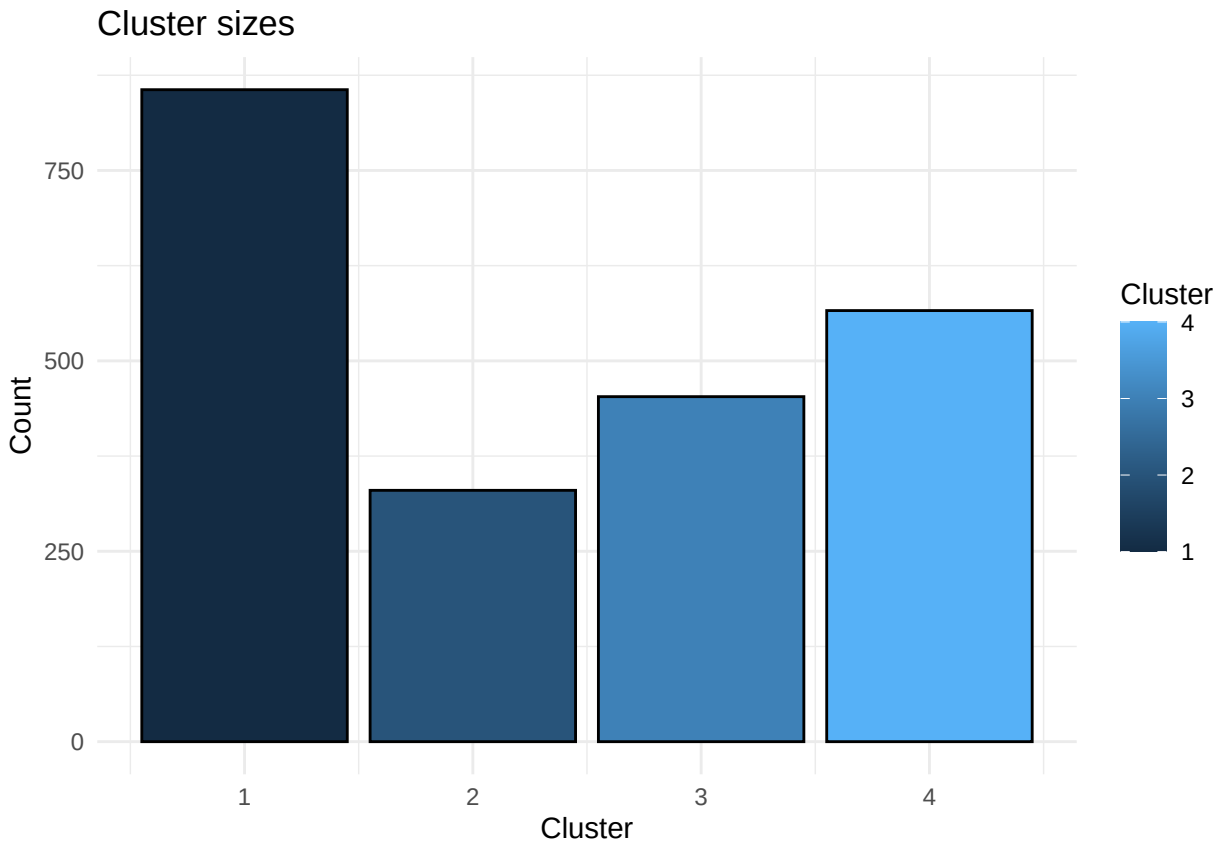
On observe ci-dessus les différentes consommations des clients pour chaque cluster pour déterminer les différences les plus flagrantes.

On voit tout d'abord qu'il y a deux groupes qui consomment beaucoup et ils consomment également des produits assez cher tel que le vin ou la viande en grande quantité.

Cela supposerait peut-être que ces deux groupes sont assez aisés ou qu'ils aient une famille nombreuse.. Ces deux groupes sont les clusters 1 et 2 tandis que pour les clusters 3 et 4, ils consomment beaucoup moins, cela serait peut-être dû à des salaires bas..

```r
cluster_sizes <- data %>%
  group_by(Cluster) %>%
  summarise(Count = n())
ggplot(cluster_sizes, aes(x = Cluster, y = Count, fill = Cluster)) +
  geom_bar(stat = "identity", color = "black") +
```

```
  labs(title = "Cluster sizes", x = "Cluster", y = "Count") +
  theme_minimal()
```




```
library(dplyr)
```

```
cluster_sizes
```

```
## # A tibble: 4 x 2
##   Cluster Count
##     <int> <int>
## 1       1   856
## 2       2   330
## 3       3   453
## 4       4   566
```

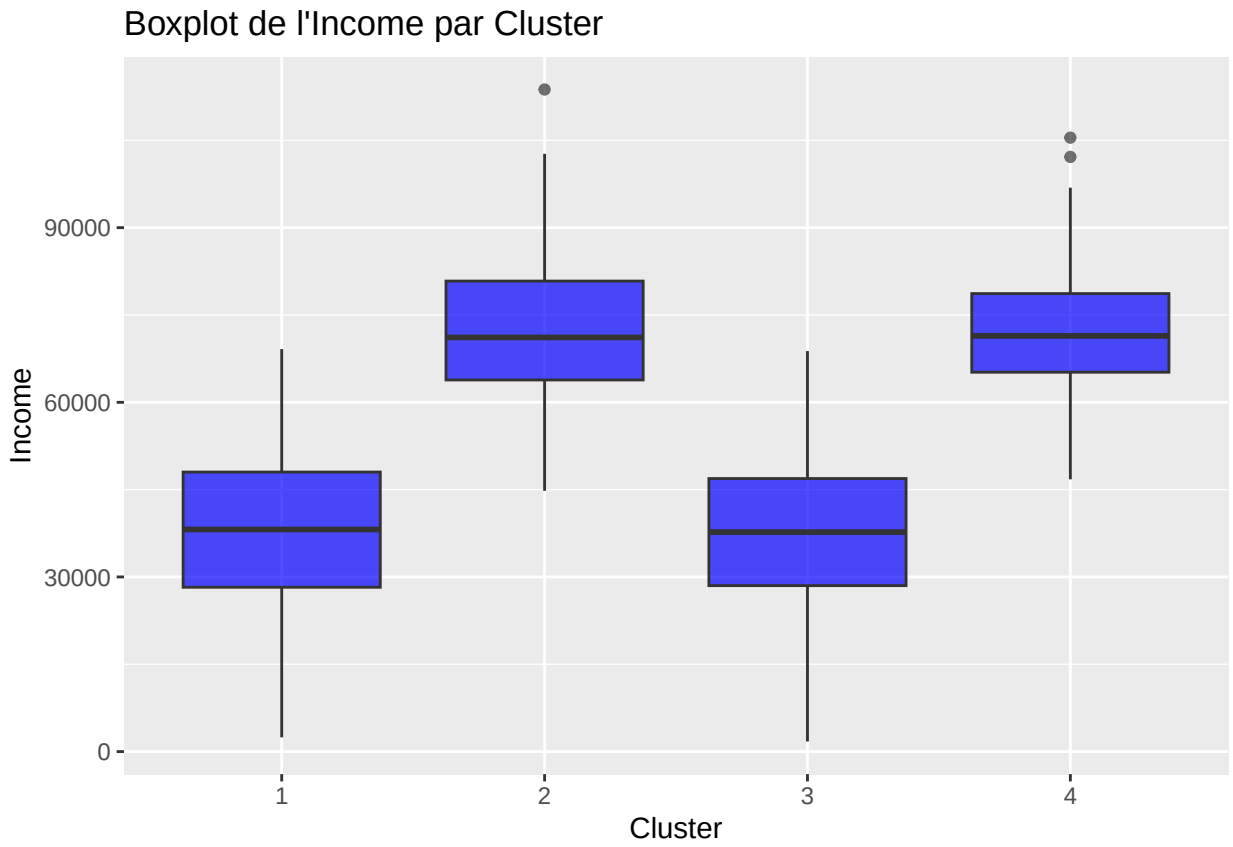
```
cluster_sizes$Pourcentage <- round((cluster_sizes$Count / nrow(data)) * 100, 0)
```

```
cluster_sizes
```

```
## # A tibble: 4 x 3
##   Cluster Count Pourcentage
##     <int> <int>       <dbl>
## 1       1   856          39
## 2       2   330          15
## 3       3   453          21
## 4       4   566          26
```

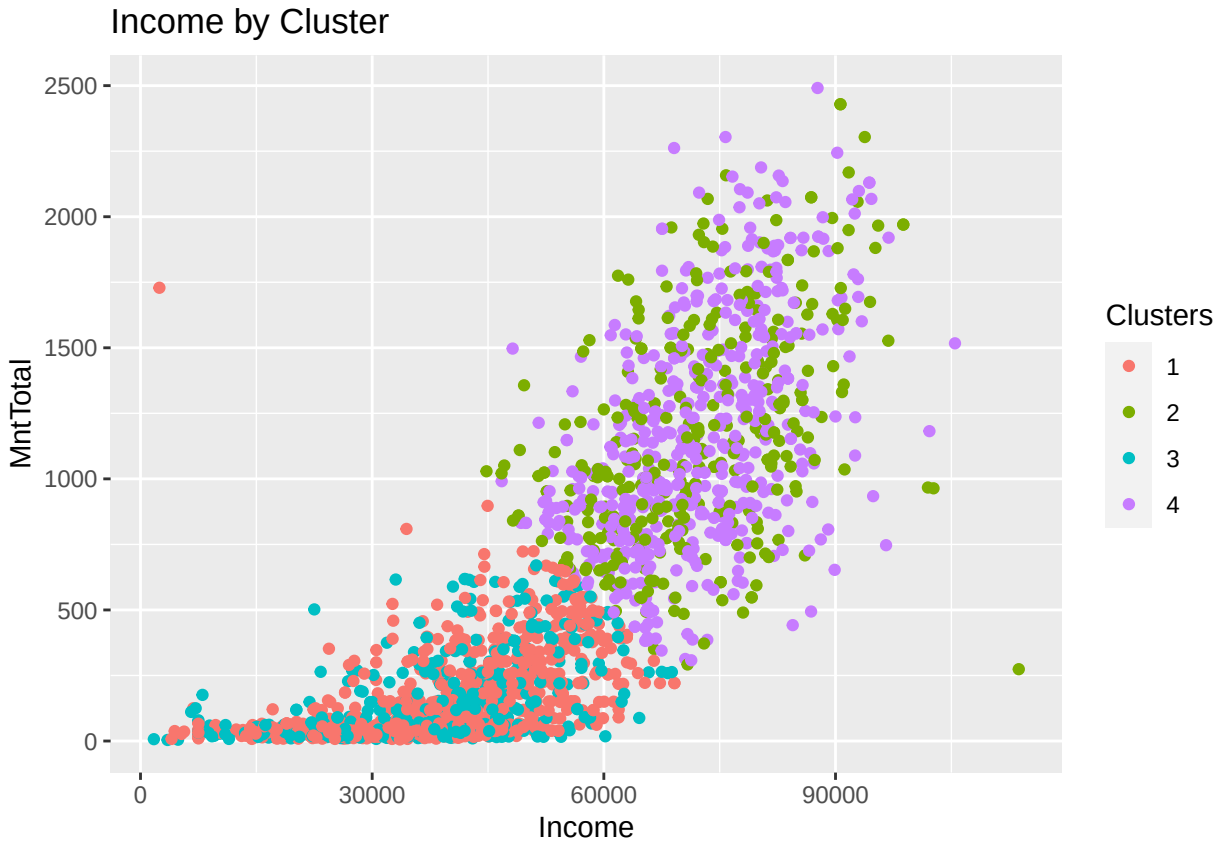
```
ggplot(data, aes(x = factor(Cluster), y = Income)) +
  geom_boxplot(fill = "blue", alpha = 0.7) +  # Personnalisez la couleur et la transparence
```

```
  labs(title = "Boxplot de l'Income par Cluster", x = "Cluster", y = "Income")
```

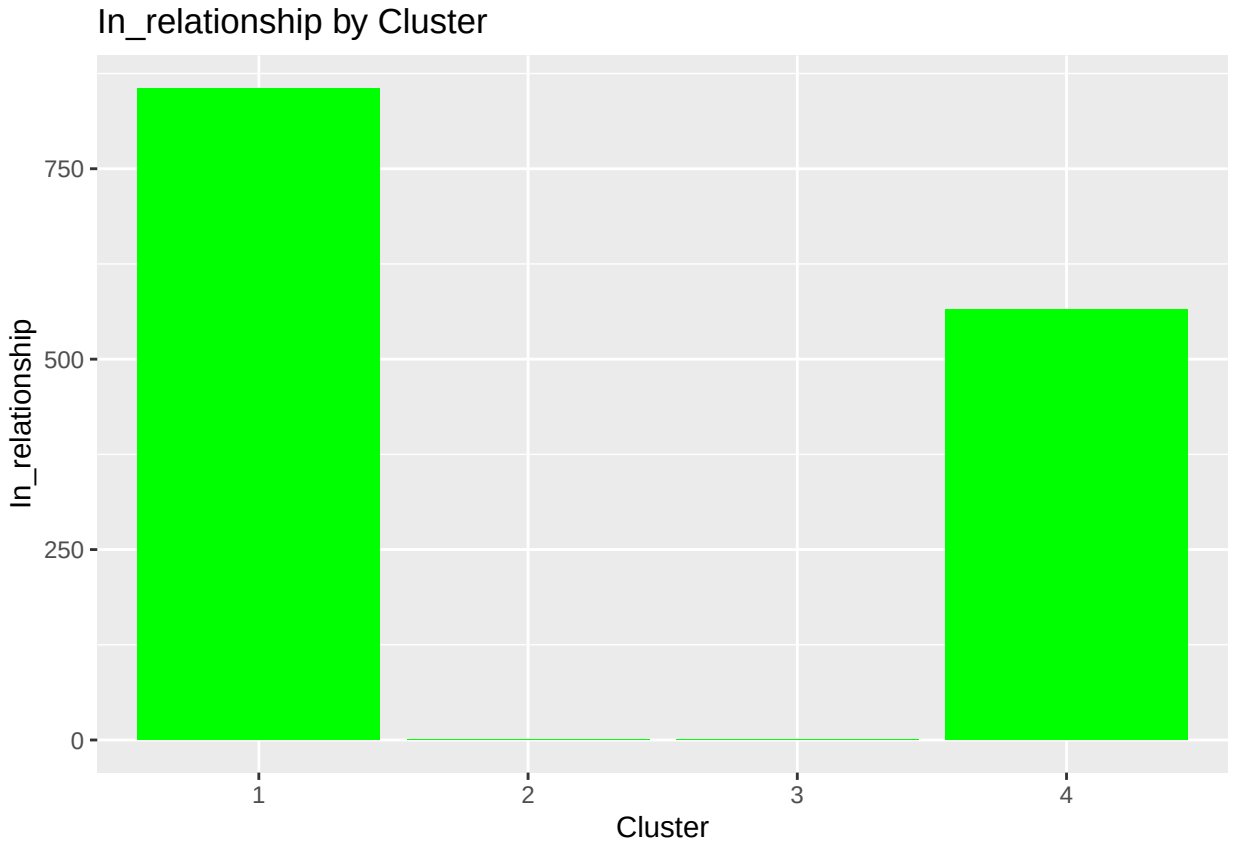


On remarque dans le graphique ci-dessus que les clusters 1 et 2 sont les groupes qui ont les salaires les plus élevés, ils ont d'ailleurs des salaires beaucoup plus élevés que ceux des clusters 3 et 4. Et d'ailleurs, on avait vu également que les clusters 1 et 2 consommaient également beaucoup plus.

```
ggplot(data, aes(x = Income, y = MntTotal, color = factor(Cluster))) +
  geom_point() +
  labs(title = "Income by Cluster", x = "Income", y = "MntTotal") +
  scale_color_discrete(name = "Clusters")
```




```r
ggplot(data, aes(x = factor(Cluster), y = In_relationship)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(title = "In_relationship by Cluster", x = "Cluster", y = "In_relationship")
```



On remarque par ce graphique que les clients des clusters 1 et 4 sont en dans une relation (soit marié, soit en couple), et les clients des clusters 2 et 3 ne le sont pas.

On distingue donc 4 types de clients:

Cluster 4:

-Client qui consomme beaucoup

-Client qui gagne un bon salaire

-Client qui est soit marié soit en couple

-Représente 26% des clients (d'après le tableau clusters_size: Voir plus haut)

Cluster 2: -Client qui consomme beaucoup

-Client qui gagne un bon salaire

-Client qui n'est marié, ni en couple

-Représente 15% des clients (d'après le tableau clusters_size: Voir plus haut)

Cluster 3:

-Client qui consomme peu

-Client qui gagne un salaire moyen ou faible

-Client qui n'est ni marié, ni en couple

-Représente 21% des clients (d'après le tableau clusters_size: Voir plus haut)

Cluster 1:

-Client qui consomme peu

-Client qui gagne un salaire moyen ou faible

-Client qui est marié ou en couple

-Représente 39% des clients (d'après le tableau clusters_size: Voir plus haut)

Solution pour générer plus de bénéfices pour les entreprises:

-Les clients du cluster 4 sont plutôt nombreux (26%), pour ces clients il faut mettre en avant des produits de qualité et assez cher (viande, vin), mettre en avant le côté famille dans les publicités. On remarque dans le graphique "Product Consumption by Clusters" que ces clients consomme pas mal de vin. Pour les entreprises vendant ce genre de produit pourrait revoir leur tarification pour réhausser un peu les prix, ou proposer de nouveaux types de vins adapter à cette clientèle.

-Les clients du cluster 2 sont en minorité. Comme pour le cluster 1, on pourrait leur proposer des produits de qualités, et revoir la tarification de certains produits. On s'adresse à une clientèle célibataire et donc surement à des jeunes ou à des clients divorcés, on pourrait donc proposer des pubs avec des amis, des évènements festifs, des voyages en solitaire..

-Pour les clients du cluster 3, on leur propose des bons de réduction, des offres intéressante rapport qualité-prix.

-Les clients du cluster 1 sont les plus nombreux, s'adapter à leur besoin serait plus rentable. On pourrait par exemple proposer des offres de réduction pour les couples, ou pour des familles avec enfants. Revoir la tarification de certains produits en sachant que les clients du cluster 4 ne gagne pas beaucoup. On pourrait également concevoir des pubs mettant en avant le côté famille pour plus l'attention de ces clients