TP 2 : Analyse univariée et bivariée avec R

Zakaria ELhajoui

1 May 2022

Contents

1 Introduction à l'énoncé du problème :

Une banque veut s'occuper de la fidélisation des clients pour son produit, les comptes d'épargne. La banque souhaite que vous identifiiez les clients susceptibles d'abandonner les soldes inférieurs au solde minimum. Vous disposez des informations sur les clients telles que l'âge, le sexe, les données démographiques et leurs transactions avec la banque. Votre tâche en tant que scientifique des données serait de prédire la probabilité de résiliation pour chaque client.

2 Objectifs :

Les objectifs fixés sont :

- Faire de l'analyse univariée avec le langage R
- Faire de l'analyse bivariée avec le langage R

3 Génération des hypothèses du problème

La génération d'hypothèses consiste à préparer une liste exhaustive de questions ou de possibilités qui affectent directement ou indirectement l'énoncé du problème ou la variable cible. Il s'agit d'une étape très importante, car elle nous évite de se lancer dans une course folle pendant l'analyse exploratoire des données. Elle réduit ce processus aux aspects les plus essentiels.

Pour générer les hypothèses, on a besoin des éléments suivants :

- Du bon sens ou de la rationalité
- Connaissance du domaine si possible
- Communication avec des experts du domaine

Ci-dessous les hypothèses avec lesquelles on travaillera cette analyse exploratoire des données.

Sur la base des données démographiques :

1. Les femmes sont-elles moins susceptibles de résilier que les hommes ?
2. Les jeunes clients sont-ils plus susceptibles de se désabonner ?
3. Les clients à faible revenu sont-ils plus susceptibles de se désabonner ?
4. Les clients ayant une ou plusieurs personnes à charge sont-ils moins susceptibles de résilier ?
5. Les clients dont la taille moyenne de la famille est inférieure à 4 sont plus susceptibles de résilier ?

Sur la base du comportement des clients :

1. Les clients vintage sont-ils moins susceptibles de se désabonner ?
2. Les clients dont le solde moyen est plus élevé sont-ils moins susceptibles de se désabonner ?
3. Les clients dont le solde mensuel diminue sont-ils plus susceptibles de se désabonner ?
4. Les clients n'ayant effectué aucune transaction au cours des trois derniers mois sont-ils plus susceptibles de se désabonner ?
5. Les clients ayant effectué des retraits importants au cours du dernier mois ont-ils plus de chances de se désabonner ?
6. Les clients ayant effectué des retraits importants au cours du dernier trimestre sont-ils plus susceptibles de se désabonner ?
7. Les clients qui ne se sont pas engagés avec la banque au cours du dernier trimestre sont-ils plus susceptibles de se désabonner ?

4 Description du Dataset

Le fichier joint “Banking_churn_prediction.csv” contient notre dataset. Il est composé de multiples variables qui peuvent être divisées en trois catégories :

Informations démographiques sur le client:

Variable	Description
customer_id	Identifiant du client
vintage	Ancienneté du client auprès de la banque en nombre de jour
age	Age du client
gender	Sexe du client
dependents	Nombre de personne à charge
occupation	Profession du client
city	Ville du client (anonymisée)

Informations bancaires des clients :

Variable	Description
customer_nw_category	Valeur nette du client (3:faible 2:moyenne 1:élevée)
branch_code	Code de la branche pour le compte du client
days_since_last_transaction	Nombre de jours depuis le dernier crédit au cours de la dernière année

Informations bancaires des clients :

Variable	Description
current_balance	Solde à ce jour
previous_month_end_balance	Solde à la fin du mois précédent
average_monthly_balance_prevQ	Soldes mensuels moyens (AMB) au trimestre précédent
average_monthly_balance_prevQ2	Soldes mensuels moyens (AMB) à l’avant dernier trimestre
percent_change_credits	Variation en pourcentage des crédits entre les deux derniers trimestres
current_month_credit	Montant total du crédit du mois en cours
previous_month_credit	Montant total du crédit du mois précédent
current_month_debit	Montant total du débit du mois en cours
previous_month_debit	Montant total du débit du mois précédent
current_month_balance	Solde moyen du mois en cours
previous_month_balance	Solde moyen du mois précédent
churn	Le solde moyen du client devient inférieur au solde minimum au cours du trimestre suivant (1/0).

5 Lecture de fichiers en R :

```r
#import data

data <- read.csv("Banking_churn_prediction.csv", header=TRUE, stringsAsFactors=FALSE)
```

5.1 Structure d'un dataframe :

```
# Checking the dimensions of a data frame
dim(data)
```

```
## [1] 28382    21
```

```
# Returning the column names
colnames(data)
```

```
##  [1] "customer_id"                    "vintage"
##  [3] "age"                            "gender"
##  [5] "dependents"                     "occupation"
##  [7] "city"                           "customer_nw_category"
##  [9] "branch_code"                    "current_balance"
## [11] "previous_month_end_balance"     "average_monthly_balance_prevQ"
## [13] "average_monthly_balance_prevQ2" "current_month_credit"
## [15] "previous_month_credit"          "current_month_debit"
## [17] "previous_month_debit"           "current_month_balance"
## [19] "previous_month_balance"         "churn"
## [21] "last_transaction"
```

```
# Viewing a summary of the data
summary(data)
```

```
##   customer_id       vintage          age           gender
##  Min.   :    1   Min.   :  73   Min.   : 1.00   Length:28382
##  1st Qu.: 7557   1st Qu.:1958   1st Qu.:36.00   Class :character
##  Median :15150   Median :2154   Median :46.00   Mode  :character
##  Mean   :15144   Mean   :2091   Mean   :48.21
##  3rd Qu.:22707   3rd Qu.:2292   3rd Qu.:60.00
##  Max.   :30301   Max.   :2476   Max.   :90.00
##
##    dependents       occupation            city        customer_nw_category
##  Min.   : 0.0000   Length:28382       Min.   :   0.0   Min.   :1.000
##  1st Qu.: 0.0000   Class :character   1st Qu.: 409.0   1st Qu.:2.000
##  Median : 0.0000   Mode  :character   Median : 834.0   Median :2.000
##  Mean   : 0.3472                      Mean   : 796.1   Mean   :2.226
##  3rd Qu.: 0.0000                      3rd Qu.:1096.0   3rd Qu.:3.000
##  Max.   :52.0000                      Max.   :1649.0   Max.   :3.000
##  NA's   :2463                         NA's   :803
##   branch_code   current_balance   previous_month_end_balance
##  Min.   :   1   Min.   :  -5504   Min.   :  -3150
##  1st Qu.: 176   1st Qu.:   1784   1st Qu.:   1906
##  Median : 572   Median :   3281   Median :   3380
##  Mean   : 926   Mean   :   7381   Mean   :   7496
##  3rd Qu.:1440   3rd Qu.:   6636   3rd Qu.:   6657
##  Max.   :4782   Max.   :5905904   Max.   :5740439
##
##  average_monthly_balance_prevQ average_monthly_balance_prevQ2
##  Min.   :   1429               Min.   : -16506
```

```
##  1st Qu.:   2181              1st Qu.:   1833
##  Median :   3543              Median :   3360
##  Mean   :   7497              Mean   :   7124
##  3rd Qu.:   6667              3rd Qu.:   6518
##  Max.   :5700290              Max.   :5010170
##
##  current_month_credit previous_month_credit current_month_debit
##  Min.   :       0     Min.   :      0.0     Min.   :      0
##  1st Qu.:       0     1st Qu.:      0.3     1st Qu.:      0
##  Median :       1     Median :      0.6     Median :     92
##  Mean   :    3433     Mean   :   3261.7     Mean   :   3659
##  3rd Qu.:     707     3rd Qu.:    749.2     3rd Qu.:   1360
##  Max.   :12269845     Max.   :2361808.3     Max.   :7637857
##
##  previous_month_debit current_month_balance previous_month_balance
##  Min.   :      0.0    Min.   :  -3374       Min.   :  -5172
##  1st Qu.:      0.4    1st Qu.:   1997       1st Qu.:   2074
##  Median :    110.0    Median :   3448       Median :   3465
##  Mean   :   3339.8    Mean   :   7451       Mean   :   7495
##  3rd Qu.:   1357.6    3rd Qu.:   6668       3rd Qu.:   6655
##  Max.   :1414168.1    Max.   :5778185       Max.   :5720144
##
##      churn        last_transaction
##  Min.   :0.0000   Length:28382
##  1st Qu.:0.0000   Class :character
##  Median :0.0000   Mode  :character
##  Mean   :0.1853
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

```
# Viewing the structure of the data
str(data)
```

```
## 'data.frame':    28382 obs. of  21 variables:
##  $ customer_id                   : int  1 2 4 5 6 7 8 9 10 11 ...
##  $ vintage                       : int  2101 2348 2194 2329 1579 1923 2048 2009 2053 2295 ...
##  $ age                           : int  66 35 31 90 42 42 72 46 31 40 ...
##  $ gender                        : chr  "Male" "Male" "Male" "" ...
##  $ dependents                    : num  0 0 0 NA 2 0 0 0 0 3 ...
##  $ occupation                    : chr  "self_employed" "self_employed" "salaried" "self_employed" .
##  $ city                          : num  187 NA 146 1020 1494 ...
##  $ customer_nw_category          : int  2 2 2 2 3 2 1 2 2 2 ...
##  $ branch_code                   : int  755 3214 41 582 388 1666 1 317 4110 38 ...
##  $ current_balance               : num  1459 5390 3913 2292 928 ...
##  $ previous_month_end_balance    : num  1459 8705 5815 2292 1402 ...
##  $ average_monthly_balance_prevQ : num  1459 7799 4910 2085 1643 ...
##  $ average_monthly_balance_prevQ2: num  1449 12419 2816 1007 1871 ...
##  $ current_month_credit          : num  0.2 0.56 0.61 0.47 0.33 ...
##  $ previous_month_credit         : num  0.2 0.56 0.61 0.47 714.61 ...
##  $ current_month_debit           : num  0.2 5486.27 6046.73 0.47 588.62 ...
##  $ previous_month_debit          : num  0.2 100.6 259.2 2143.3 1538.1 ...
##  $ current_month_balance         : num  1459 6497 5006 2292 1157 ...
##  $ previous_month_balance        : num  1459 8788 5070 1670 1677 ...
```

```
##  $ churn                         : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ last_transaction              : chr  "2019-05-21" "2019-11-01" "NaT" "2019-08-06" ...
```

5.2 Accéder aux variables d'un dataframe

```
# Returning the values of a data frame component

# data$gender

# Returning only first or last values

head(x = data$age)
```

```
## [1] 66 35 31 90 42 42
```

```
# Returning a component of the data frame

# data['gender']
```

5.3 Subsetting

```
data2 <- read.csv("Banking_churn_prediction.csv")

fl = subset(data2, gender == "Male")

# With dplyr's filter function:

# install.packages('dplyr')

# With dplyr's filter function:

library(stats)
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
```

```
# f2 = filter(data2, gender == "Male")
```

5.4 Ajout et suppression d'une variable dans un dataframe

```
# Removing NA values
d = data.frame(data2)
## d[complete.cases(d), ]

# Adding a new column
data$next_age
```

```
## NULL
```

```
# data$next_age = age + 1
```

5.5 Grouping Operations

```
# Applying summarize to groups of observations

# by_gender = group_by()
```

6 Identification des variables et typage

```
# A closer look at the data types present in the data
str(data)
```

```
## 'data.frame':    28382 obs. of  21 variables:
##  $ customer_id                   : int  1 2 4 5 6 7 8 9 10 11 ...
##  $ vintage                       : int  2101 2348 2194 2329 1579 1923 2048 2009 2053 2295 ...
##  $ age                           : int  66 35 31 90 42 42 72 46 31 40 ...
##  $ gender                        : chr  "Male" "Male" "Male" "" ...
##  $ dependents                    : num  0 0 0 NA 2 0 0 0 0 3 ...
##  $ occupation                    : chr  "self_employed" "self_employed" "salaried" "self_employed" .
##  $ city                          : num  187 NA 146 1020 1494 ...
##  $ customer_nw_category          : int  2 2 2 2 3 2 1 2 2 2 ...
##  $ branch_code                   : int  755 3214 41 582 388 1666 1 317 4110 38 ...
##  $ current_balance               : num  1459 5390 3913 2292 928 ...
##  $ previous_month_end_balance    : num  1459 8705 5815 2292 1402 ...
##  $ average_monthly_balance_prevQ : num  1459 7799 4910 2085 1643 ...
##  $ average_monthly_balance_prevQ2: num  1449 12419 2816 1007 1871 ...
##  $ current_month_credit          : num  0.2 0.56 0.61 0.47 0.33 ...
##  $ previous_month_credit         : num  0.2 0.56 0.61 0.47 714.61 ...
##  $ current_month_debit           : num  0.2 5486.27 6046.73 0.47 588.62 ...
##  $ previous_month_debit          : num  0.2 100.6 259.2 2143.3 1538.1 ...
##  $ current_month_balance         : num  1459 6497 5006 2292 1157 ...
##  $ previous_month_balance        : num  1459 8788 5070 1670 1677 ...
##  $ churn                         : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ last_transaction              : chr  "2019-05-21" "2019-11-01" "NaT" "2019-08-06" ...
```

Il y a beaucoup de variables visibles en même temps, alors réduisons cela en regardant un type de données à la fois. Nous allons commencer par int

```
data_intger <- select_if(data, is.integer) # Subset integer columns with dplyr

fd = sapply(data_intger, class)# Print subset to RStudio console

fd
```

```
##          customer_id              vintage                  age
##            "integer"            "integer"            "integer"
## customer_nw_category          branch_code                churn
##            "integer"            "integer"            "integer"
```

```
data$dependents <- as.integer(data$dependents) #"dependents" devrait être un nombre entier. Devrait êtr


data$last_transaction <- as.Date(data$last_transaction, "%Y-%m-%d") #la colonne 'last_transaction' doit



data$jour_de_l_annee <- strftime(data$last_transaction, format = "%j")
data$mois_de_l_annee <- strftime(data$last_transaction, format = "%m")
data$jour_du_mois <- strftime(data$last_transaction, format = "%d")
data$jour_de_la_s0emaine <- strftime(data$last_transaction, format = "%w")
data$semaine_de_l_annee <- strftime(data$last_transaction, format = "%V")

data$jour_de_l_annee <- as.integer(data$jour_de_l_annee)
data$mois_de_l_annee <- as.integer(data$mois_de_l_annee)
data$jour_du_mois <- as.integer(data$jour_du_mois)
data$jour_de_la_semaine <- as.integer(data$jour_de_la_s0emaine)
data$semaine_de_l_annee <- as.integer(data$semaine_de_l_annee)

data <- within(data, rm(last_transaction))
str(data)
```

```
## 'data.frame':    28382 obs. of  26 variables:
##  $ customer_id                  : int  1 2 4 5 6 7 8 9 10 11 ...
##  $ vintage                      : int  2101 2348 2194 2329 1579 1923 2048 2009 2053 2295 ...
##  $ age                          : int  66 35 31 90 42 42 72 46 31 40 ...
##  $ gender                       : chr  "Male" "Male" "Male" "" ...
##  $ dependents                   : int  0 0 0 NA 2 0 0 0 0 3 ...
##  $ occupation                   : chr  "self_employed" "self_employed" "salaried" "self_employed" .
##  $ city                         : num  187 NA 146 1020 1494 ...
##  $ customer_nw_category         : int  2 2 2 2 3 2 1 2 2 2 ...
##  $ branch_code                  : int  755 3214 41 582 388 1666 1 317 4110 38 ...
##  $ current_balance              : num  1459 5390 3913 2292 928 ...
##  $ previous_month_end_balance   : num  1459 8705 5815 2292 1402 ...
##  $ average_monthly_balance_prevQ : num  1459 7799 4910 2085 1643 ...
##  $ average_monthly_balance_prevQ2: num  1449 12419 2816 1007 1871 ...
##  $ current_month_credit         : num  0.2 0.56 0.61 0.47 0.33 ...
##  $ previous_month_credit        : num  0.2 0.56 0.61 0.47 714.61 ...
##  $ current_month_debit          : num  0.2 5486.27 6046.73 0.47 588.62 ...
```

```
##  $ previous_month_debit          : num  0.2 100.6 259.2 2143.3 1538.1 ...
##  $ current_month_balance         : num  1459 6497 5006 2292 1157 ...
##  $ previous_month_balance        : num  1459 8788 5070 1670 1677 ...
##  $ churn                         : int  0 0 0 1 1 0 0 0 0 0 ...
##  $ jour_de_l_annee               : int  141 305 NA 218 307 305 267 193 346 365 ...
##  $ mois_de_l_annee               : int  5 11 NA 8 11 11 9 7 12 12 ...
##  $ jour_du_mois                  : int  21 1 NA 6 3 1 24 12 12 31 ...
##  $ jour_de_la_s0emaine           : chr  "2" "5" NA "2" ...
##  $ semaine_de_l_annee            : int  21 44 NA 32 44 44 39 28 50 1 ...
##  $ jour_de_la_semaine            : int  2 5 NA 2 0 5 2 5 4 2 ...
```

7 Analyse univariée

repose sur l'analyse des variables (les colonnes) dont, les méthodes et fonctions utilisées seront différentes selon qu'il s'agit d'une variable quantitative (variable numérique pouvant prendre un grand nombre de valeurs) ou d'une variable qualitative (variable pouvant prendre un nombre limité de valeurs appelées modalités : situation familiale par exemple) ### Analyse d'une variable quantitative Une variable quantitative est une variable de type numérique (un nombre) qui peut prendre un grand nombre de valeurs. En effet, On en a plusieurs dans notre jeu de données, notamment l'âge, vintage, current_balance ou le current_month_balance ... ####Représentation graphique ***Age***

7.1 Analyse d'une variable quantitative

Une variable quantitative est une variable de type numérique (un nombre) qui peut prendre un grand nombre de valeurs. La description d'une variable quantitative se base sur les statistiques suivantes : la moyenne, la médiane, la variance, l'écart-type, les quantiles. On peut aller plus loin en regardant l'asymétrie et l'aplatissement. On différencie deux types de variables : - les variables quantitatives - les variables qualitatives

7.2 Représentation graphique

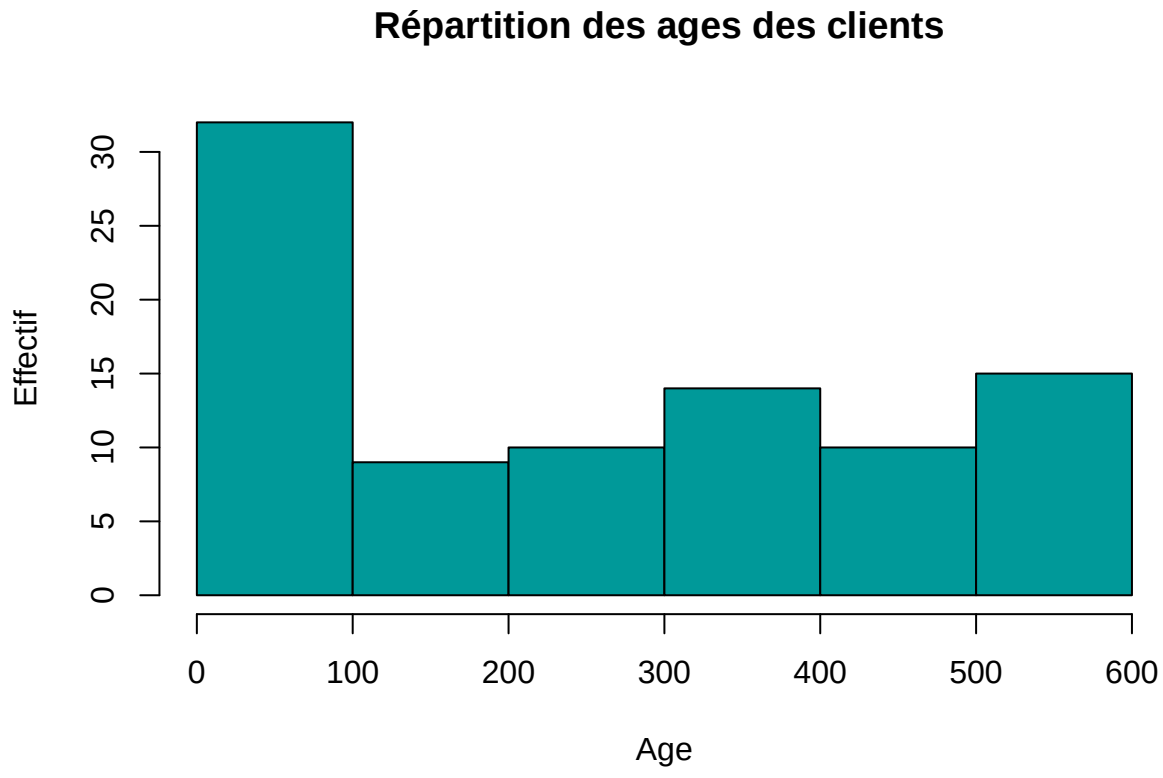
```
db_banque = tibble::as_tibble(data) # Transformation to a tibble
head(db_banque)
```

```
## # A tibble: 6 x 26
##   customer_id vintage   age gender  dependents occupation  city customer_nw_cat~
##         <int>   <int> <int> <chr>        <int> <chr>      <dbl>            <int>
## 1           1    2101    66 "Male"           0 self_empl~   187                2
## 2           2    2348    35 "Male"           0 self_empl~    NA                2
## 3           4    2194    31 "Male"           0 salaried     146                2
## 4           5    2329    90 ""              NA self_empl~  1020                2
## 5           6    1579    42 "Male"           2 self_empl~  1494                3
## 6           7    1923    42 "Femal~          0 self_empl~  1096                2
## # ... with 18 more variables: branch_code <int>, current_balance <dbl>,
## #   previous_month_end_balance <dbl>, average_monthly_balance_prevQ <dbl>,
## #   average_monthly_balance_prevQ2 <dbl>, current_month_credit <dbl>,
## #   previous_month_credit <dbl>, current_month_debit <dbl>,
## #   previous_month_debit <dbl>, current_month_balance <dbl>,
## #   previous_month_balance <dbl>, churn <int>, jour_de_l_annee <int>,
## #   mois_de_l_annee <int>, jour_du_mois <int>, jour_de_la_s0emaine <chr>, ...
```

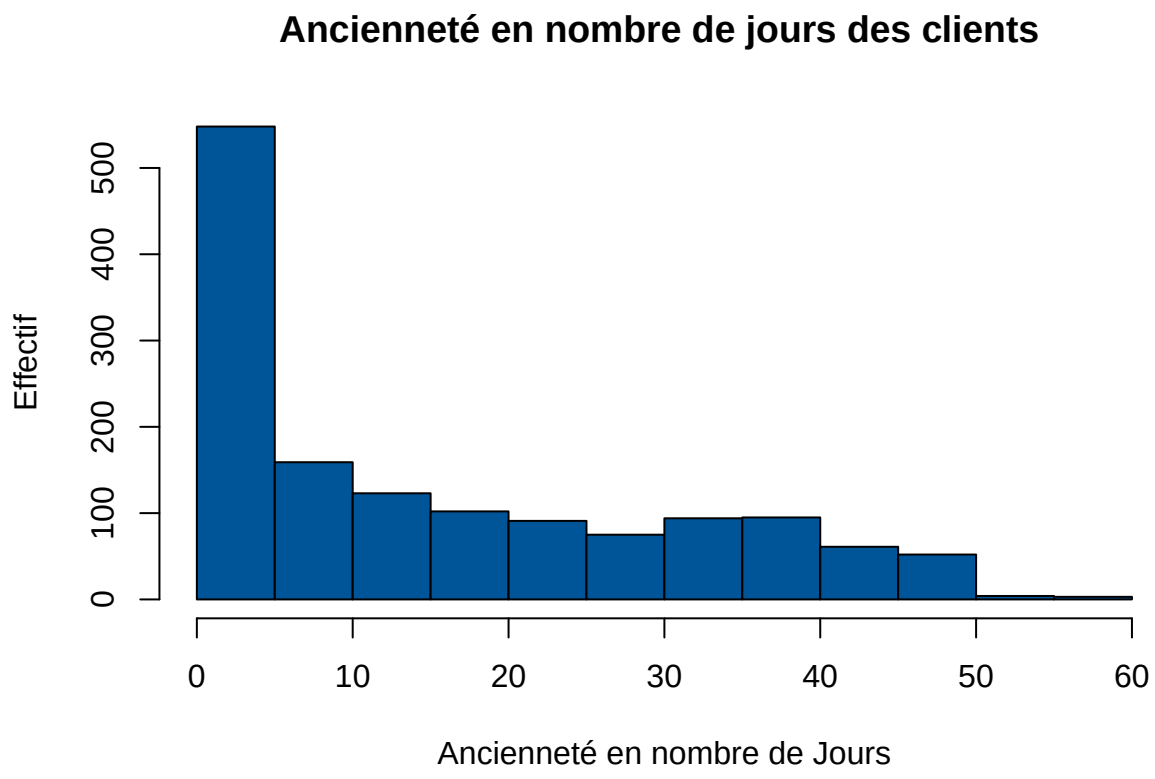
```r
library(tidyr)
db_banque <- drop_na(db_banque) # DELETE NA
head(db_banque)
```

```
## # A tibble: 6 x 26
##   customer_id vintage   age gender dependents occupation   city customer_nw_cat~
##         <int>   <int> <int> <chr>       <int> <chr>       <dbl>             <int>
## 1           1    2101    66 Male            0 self_emplo~   187                 2
## 2           6    1579    42 Male            2 self_emplo~  1494                 3
## 3           7    1923    42 Female          0 self_emplo~  1096                 2
## 4           8    2048    72 Male            0 retired      1020                 1
## 5           9    2009    46 Male            0 self_emplo~   623                 2
## 6          10    2053    31 Male            0 salaried     1096                 2
## # ... with 18 more variables: branch_code <int>, current_balance <dbl>,
## #   previous_month_end_balance <dbl>, average_monthly_balance_prevQ <dbl>,
## #   average_monthly_balance_prevQ2 <dbl>, current_month_credit <dbl>,
## #   previous_month_credit <dbl>, current_month_debit <dbl>,
## #   previous_month_debit <dbl>, current_month_balance <dbl>,
## #   previous_month_balance <dbl>, churn <int>, jour_de_l_annee <int>,
## #   mois_de_l_annee <int>, jour_du_mois <int>, jour_de_la_s0emaine <chr>, ...
```

```r
hist(table(db_banque$age),col = "#009999",
main = "Répartition des ages des clients ",
xlab = "Age",
ylab = "Effectif")
```



```r
# The seniority of customers in number of days (Vintage)

hist(table(db_banque$vintage),col = "#005599",
main = "Ancienneté en nombre de jours des clients ",
xlab = "Ancienneté en nombre de Jours",
ylab = "Effectif")
```




```r
## Customer Revenue (customer_nw_category)

barplot(height = table(db_banque$customer_nw_category), main = 'Client Revenue Classification', las= 2,
```

7.3 Indicateurs de tendance centrale

Characterize a quantitative variable

Age

```
min(db_banque$age)
```

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

```
max(db_banque$age)
```

```
## [1] 90
```

```
mean(db_banque$age)
```

```
## [1] 48.42042
```

```
range(db_banque$age)
```

```
## [1]  1 90
```

```r
median(db_banque$age)
```

```
## [1] 46
```

Vintage in days

```r
min(db_banque$vintage)
```

```
## [1] 73
```

```r
max(db_banque$vintage)
```

```
## [1] 2476
```

```r
mean(db_banque$vintage)
```

```
## [1] 2090.469
```

```r
range(db_banque$vintage)
```

```
## [1]   73 2476
```

```r
median(db_banque$vintage)
```

```
## [1] 2154
```

7.3.0.1 Indicateurs de dispersion Les indicateurs de dispersion permettent de mesurer si les valeurs sont plutôt regroupées ou au contraire plutôt dispersées.

```r
indicateur_age <- max(db_banque$age) - min(db_banque$age)
indicateur_age
```

```
## [1] 89
```

```r
var(x = db_banque$age) # Variance
```

```
## [1] 285.5581
```

```r
sd(x = db_banque$age) # Ecart-type
```

```
## [1] 16.89846
```

Les indicateurs de dispersion les plus utilisés sont la variance ou, de manière équivalente, l'écarttype (qui est égal àla racine carrée de la variance).

```
quantile(x = db_banque$age)
```

```
##   0%  25%  50%  75% 100%
##    1   36   46   60   90
```

```
quantile(x = db_banque$age, probs = 0.25) ## Premier quartile
```

```
## 25%
##  36
```

```
quantile(x = db_banque$age, probs = 0.75) ## Troisième quartile
```

```
## 75%
##  60
```

```
## Notons enfin que la fonction summary permet d'obtenir d'un coup plusieurs indicateurs classiques

summary(object = db_banque$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   36.00   46.00   48.42   60.00   90.00
```

7.3.0.2 Indicateur d'asymétrie et d'aplatissement

7.3.0.2.1 Le coefficient d'asymétrie (skewness) Le fait qu'une distribution soit asymétrique désigne le fait que les observations sont réparties de manière inégale de part et d'autre du milieu de la distribution. L'indice statistique qui permet de rendre compte du niveau d'asymétrie est le coefficient d'asymétrie, ou skewness en anglais

```
library(e1071)
skewness(x = db_banque$age)
```

```
## [1] 0.365632
```

```
## install.packages("moments")
library(moments)
```

7.3.0.2.2 Le coefficient d'applatissement (kurtosis)

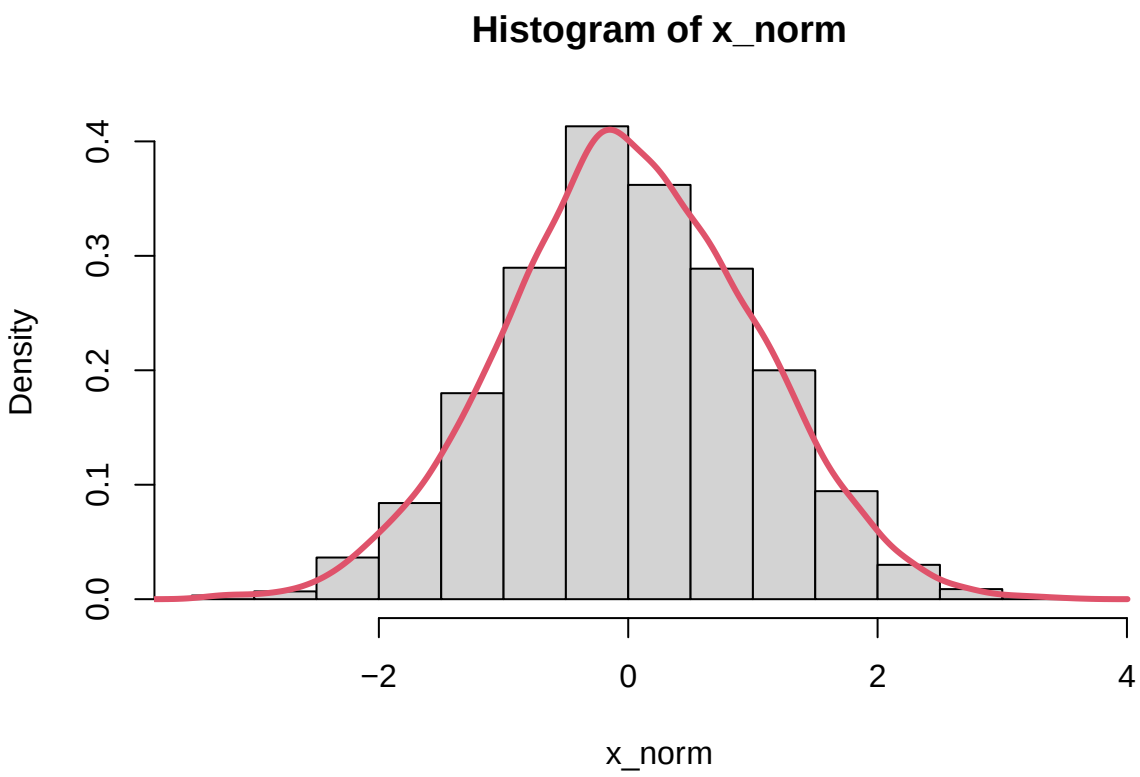
```
##
## Attaching package: 'moments'
```

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

```r
kurtosis(db_banque$age)
```

```
## [1] 2.821942
```

```r
# Calculons le skewness et le kurtosis pour une distribution normale :
set.seed(101)
x_norm <- rnorm(5000)
hist(x_norm, prob = TRUE)
lines(density(x_norm), col = 2, lwd = 3)
```




```r
moments::kurtosis(x_norm)
```

```
## [1] 2.945199
```

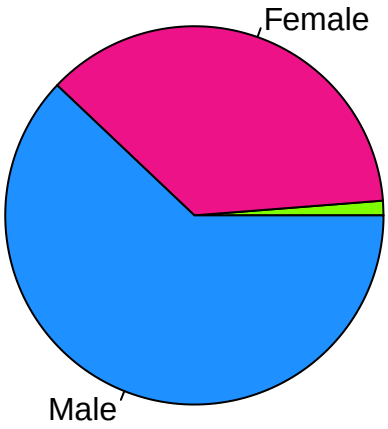
```r
moments::skewness(x_norm)
```

```
## [1] -0.0007481016
```

7.4 Analyse bivariée

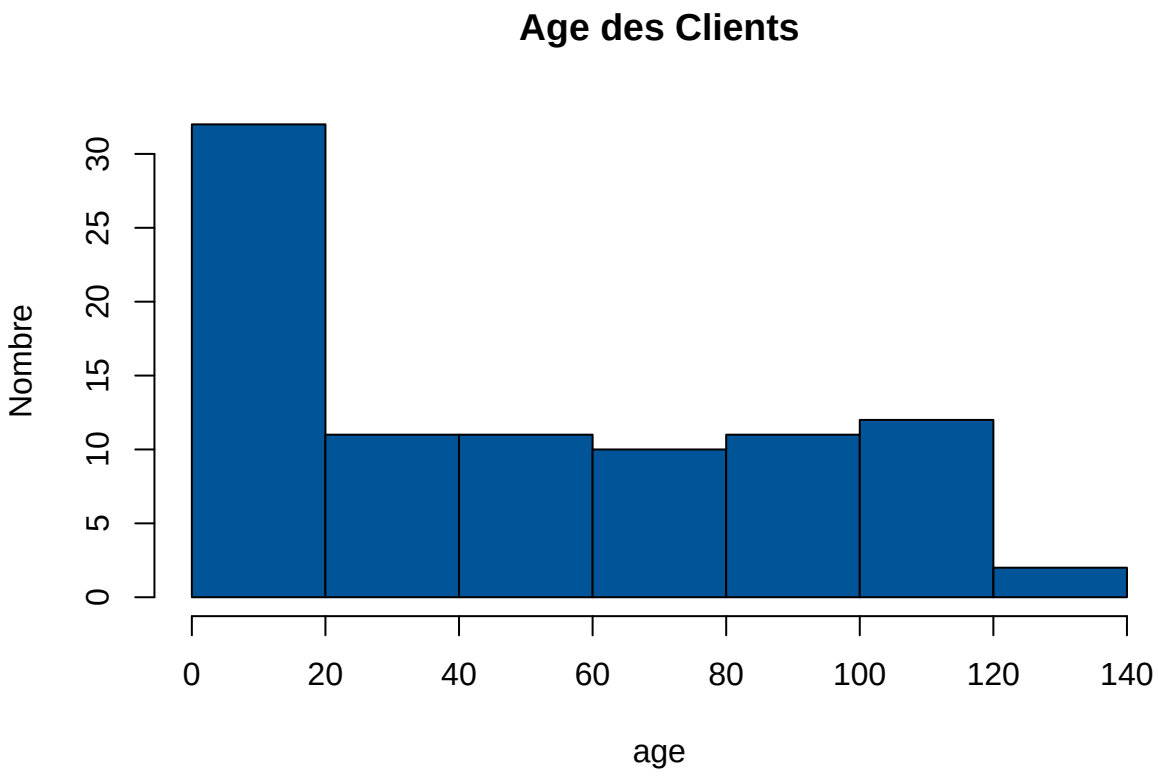
7.4.1 Les femmes sont-elles moins susceptibles de résilier que les hommes ?

```r
data1 <- db_banque |> filter(db_banque$churn == 1) # les clients qui ont abandonnés les soldes inférieu
pie(table(data1$gender), col = c("chartreuse1","deeppink2","dodgerblue"))
```



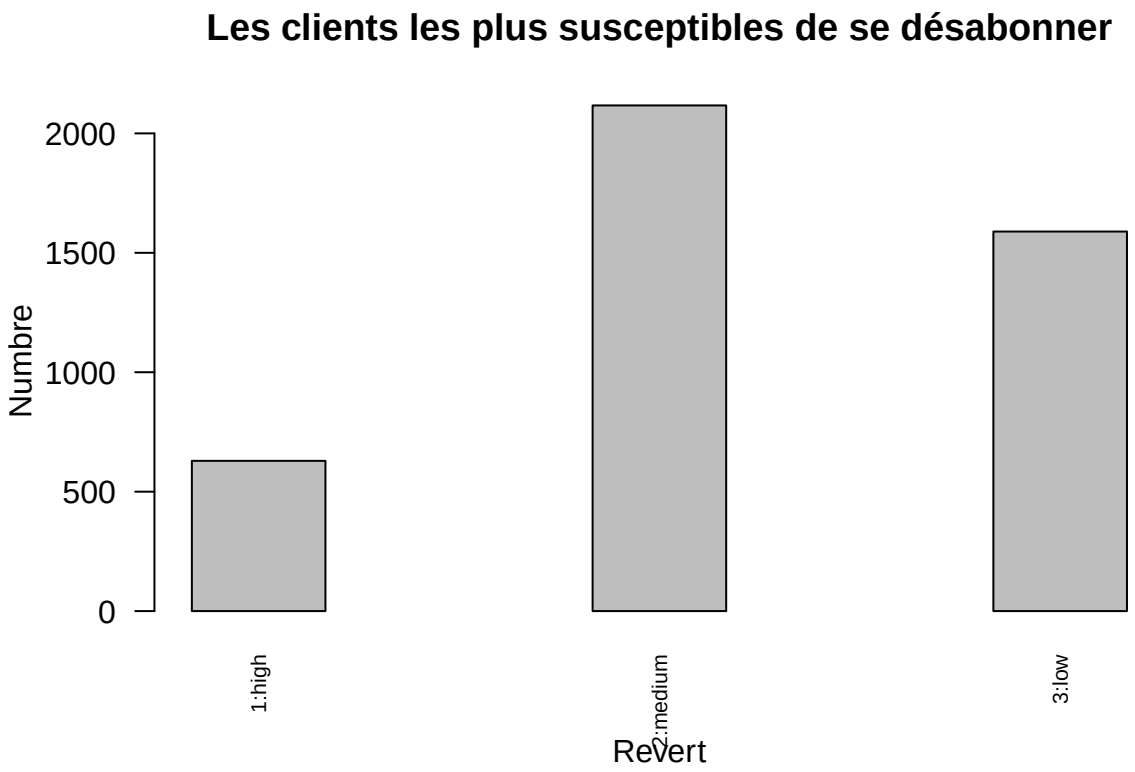
7.4.2 Les jeunes clients sont-ils plus susceptibles de se désabonner ?

```r
hist(table(data1$age), col = "#005599",
 main = "Age des Clients",
 xlab = "age",
 ylab = "Nombre")
```



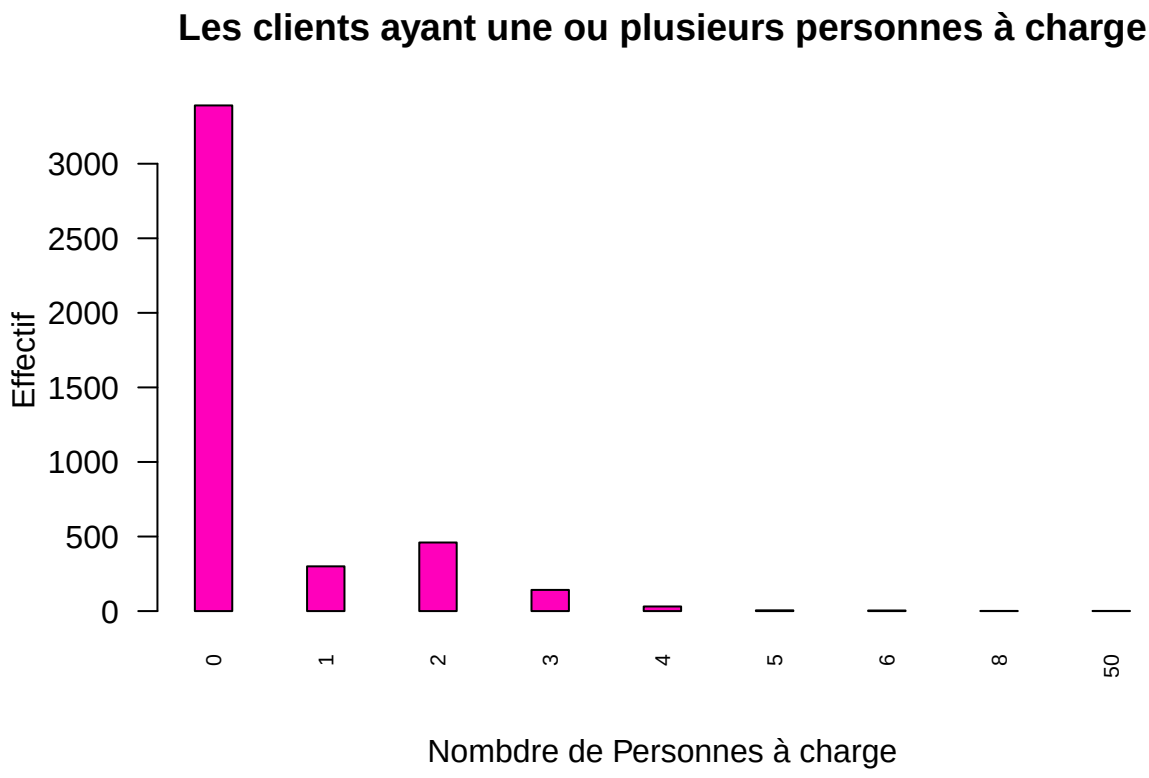
7.4.3 Les clients à faible revenu sont-ils plus susceptibles de se désabonner ?

```
barplot(height = table(data1$customer_nw_category), main = 'Les clients les plus susceptibles de se dés
```



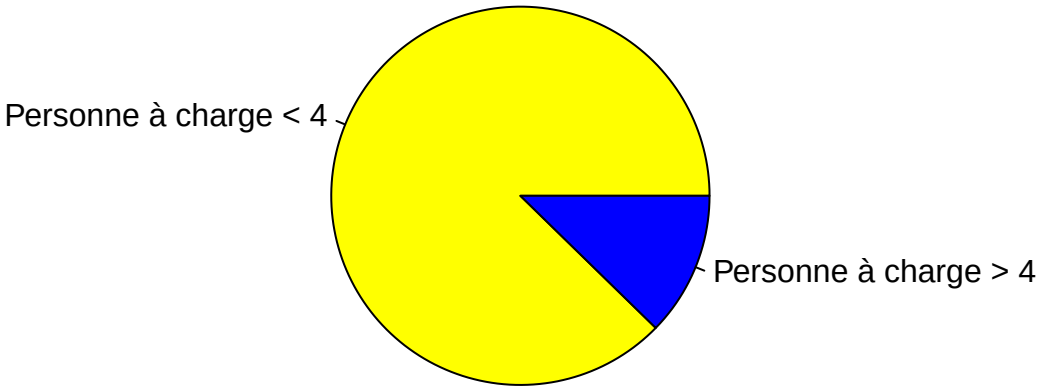
7.4.4 Les clients ayant une ou plusieurs personnes à charge sont-ils moins susceptibles de résilier ?

```
barplot(height = table(data1$dependents), main = 'Les clients ayant une ou plusieurs personnes à charge
```



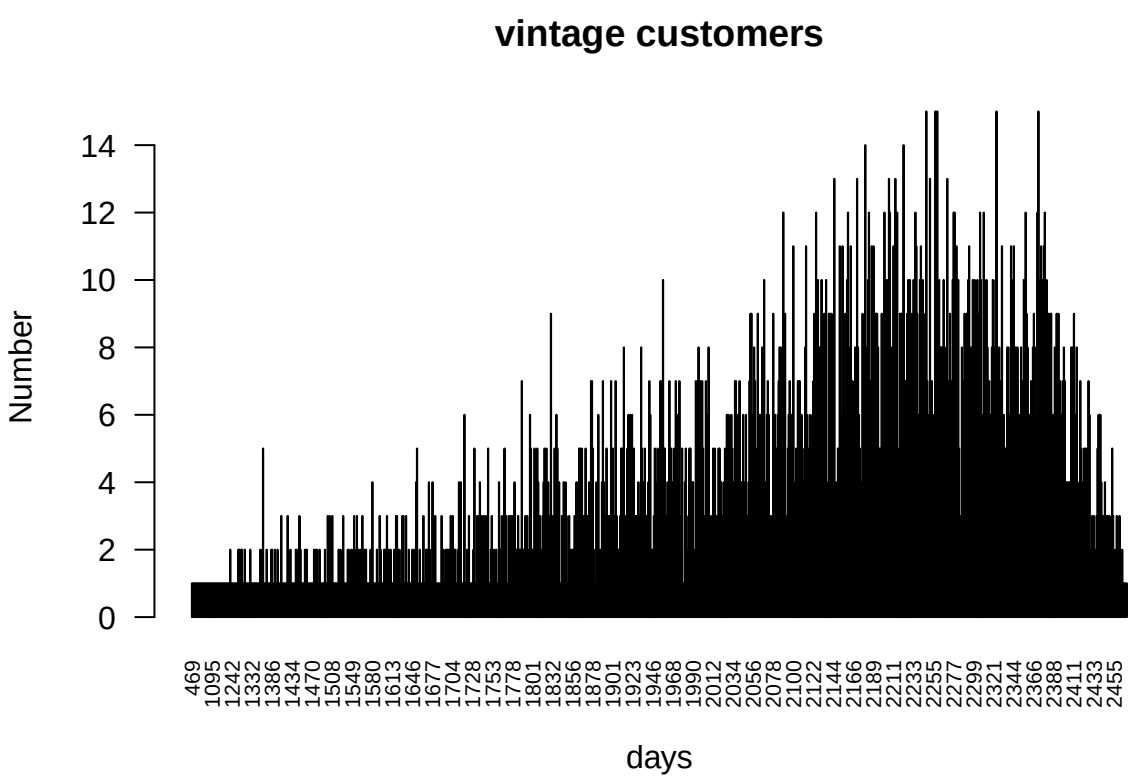
7.4.5 Les clients dont la taille moyenne de la famille est inférieure à 4 sont plus susceptibles de résilier ?

```
inferieur4<- data1 |> filter(dependents<4)
inf4=sum(inferieur4$dependents)
superieur4<- data1 |> filter(dependents>=4)
sup4=sum(superieur4$dependents)
y=c("Personne à charge < 4", "Personne à charge > 4")
x=c(inf4,sup4)
d=data.frame(x,y)
pie(d$x,labels = d$y, col=c("yellow", "blue"))
```



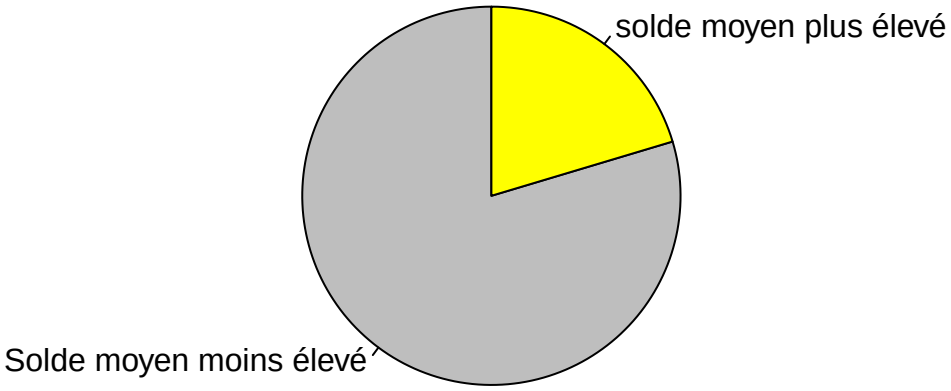
7.4.6 Les clients vintage sont-ils moins susceptibles de se désabonner ?

```
barplot(height = table(data1$vintage), main = 'vintage customers', las= 2, cex.names = 0.7, space=2, xla
```



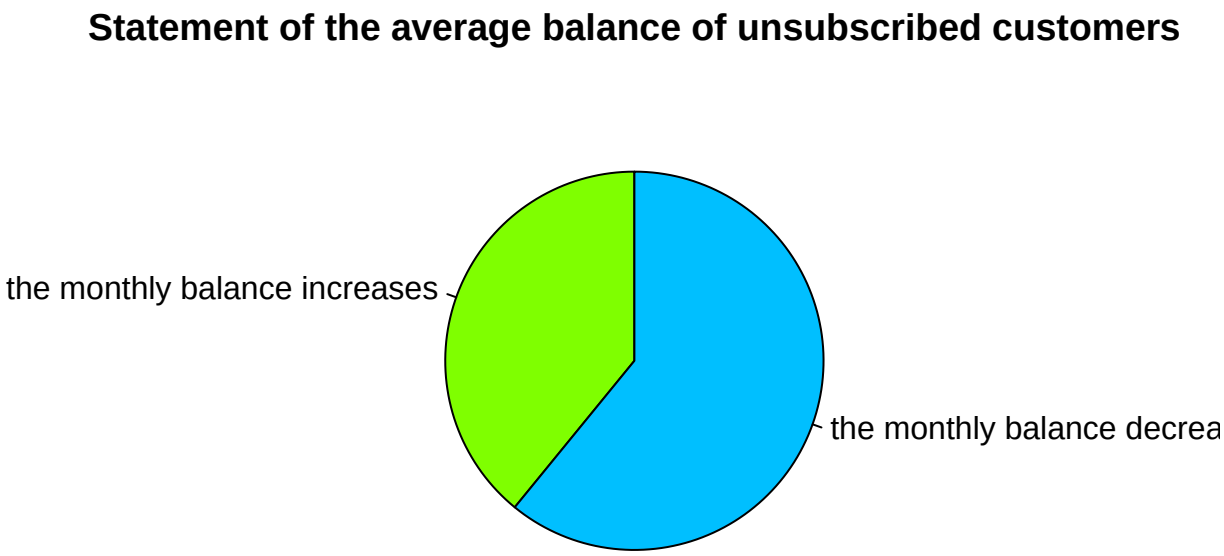
7.4.7 Les clients dont le solde moyen est plus élevé sont-ils moins susceptibles de se désabonner ?

```
moyen_solde <- mean(data1$current_month_balance)
d_sup_moy <- data1 |> filter(current_month_balance > moyen_solde) |> count()
d_inf_moy <- data1|> filter(current_month_balance <= moyen_solde) |> count()
y1=c("Solde moyen moins élevé", "solde moyen plus élevé ")
x1=c(d_inf_moy,d_sup_moy)
d2=tibble(x1,y1)
pie(as.double(d2$x1),labels = d2$y1, col=c("gray", "yellow"), init.angle = 90)
```



7.4.8 Les clients dont le solde mensuel diminue sont-ils plus susceptibles de se désabonner ?

```
# New Variable variable percent_change_credits
data_etat_sole_mensuel <- data1 |> select(average_monthly_balance_prevQ2,average_monthly_balance_prevQ)
  mutate(percent_change_credits=average_monthly_balance_prevQ2-average_monthly_balance_prevQ)


# classification
data_sup0 <- data_etat_sole_mensuel|> filter(percent_change_credits >0) |> count()
data_inf0 <- data_etat_sole_mensuel|> filter(percent_change_credits <0) |> count()

y2=c("the monthly balance increases", "the monthly balance decreases ")
x2=c(data_sup0,data_inf0)
d3=tibble(x2,y2)
pie(as.double(d3$x2),labels = d3$y2, col=c("chartreuse", "deepskyblue"), init.angle = 90, main="Stateme
```

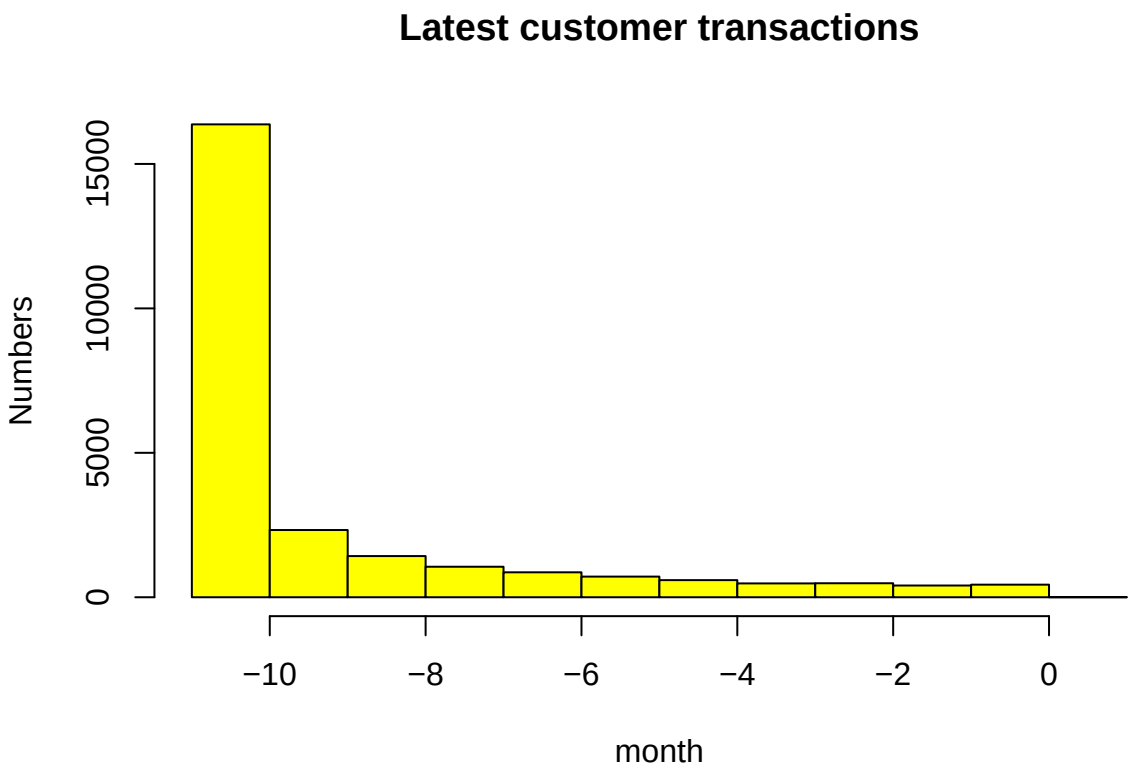


7.4.9 Les clients n'ayant effectué aucune transaction au cours des trois derniers mois sont-ils plus susceptibles de se désabonner ?

```
d1 <- read.csv("Banking_churn_prediction.csv", header=TRUE, stringsAsFactors=FALSE)

transaction_3mois <- d1 |> filter(last_transaction != 'NaT') # DELETE last_transaction 'NaT'

#  Adding new variable diff_last_transaction, containing the duration between le 01/01/2019 et last_tra

#install.packages("zoo")
library(zoo)
```

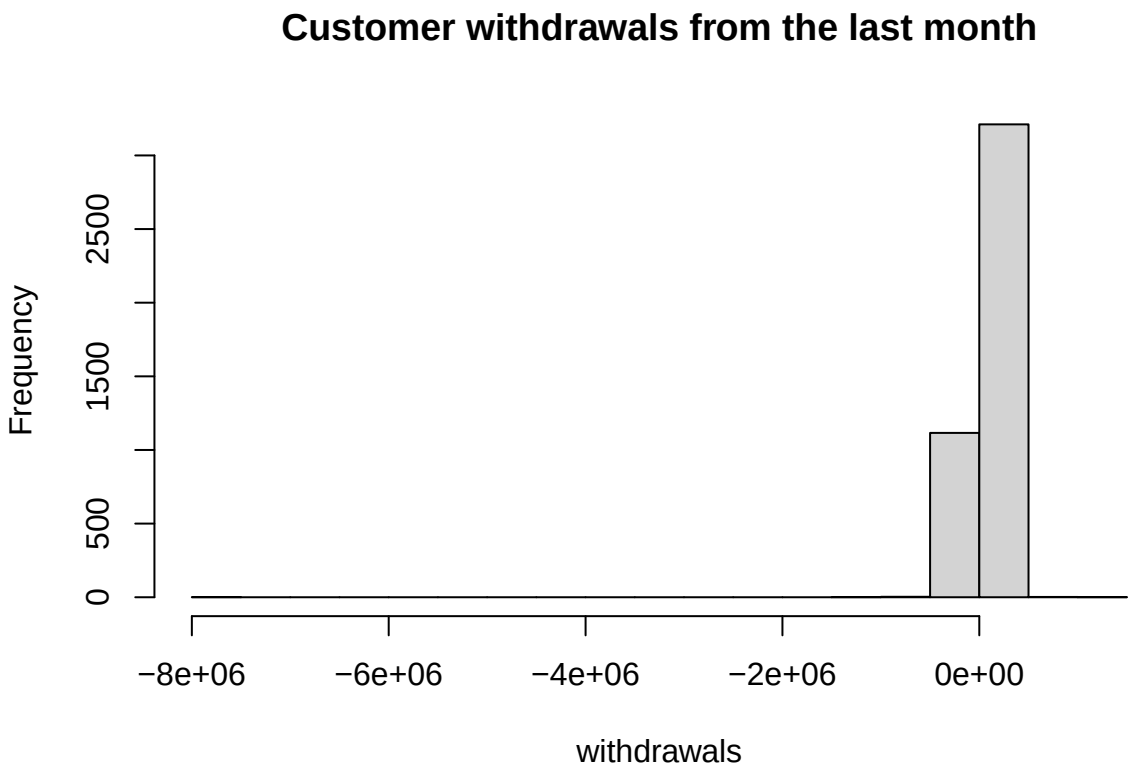
```
##
## Attaching package: 'zoo'

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

```
transaction_3mois <- transaction_3mois |> mutate(diff_last_transaction= 12* ((as.yearmon('2019-01-01'))-

hist(round(transaction_3mois$diff_last_transaction,2), main="Latest customer transactions", xlab="month
```



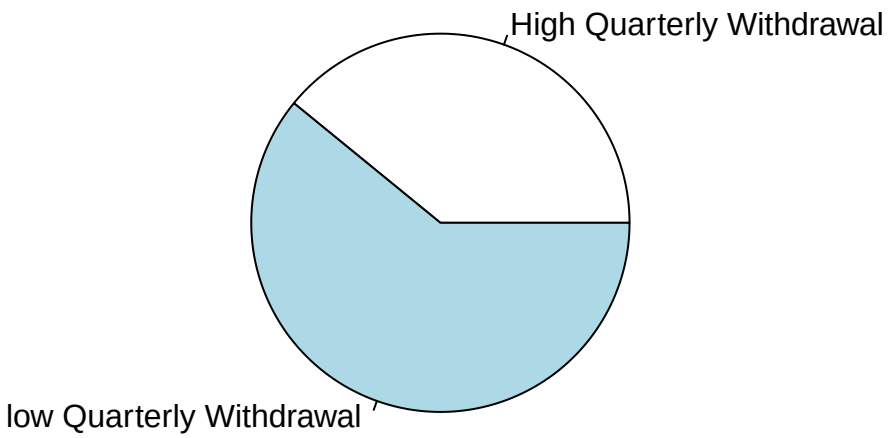
7.4.10 Les clients ayant effectué des retraits importants au cours du dernier mois ont-ils plus de chances de se désabonner ?

```
retrait_dernier_mois <- data1 |> mutate(retrait_mois_precedent=previous_month_end_balance-current_month
hist(retrait_dernier_mois$retrait_mois_precedent, main="Customer withdrawals from the last month", xlab=
```



7.4.11 Les clients ayant effectué des retraits importants au cours du dernier trimestre sont-ils plus susceptibles de se désabonner ?

```
retrait_dernier_trimestre <- data1 |> mutate(trimestre=average_monthly_balance_prevQ2-average_monthly_ba

retrait_sup_0 <- retrait_dernier_trimestre|> filter(trimestre >0) |> count()
retrait_inf_0 <- retrait_dernier_trimestre|> filter(trimestre <0) |> count()

y3=c("High Quarterly Withdrawal", "low Quarterly Withdrawal ")
x3=c(retrait_sup_0,retrait_inf_0)
d4=tibble(x3,y3)
pie(as.double(d4$x3), main="Last quarter customer withdrawals", labels = d4$y3)
```


Last quarter customer withdrawals
High Quarterly Withdrawal
low Quarterly Withdrawal