

Analyse statistique sur R

Julien JACQUES

2025-02-20

Récupération et mise en forme des données

```
library(readr)
VisaPremier <- read_delim("VisaPremier.txt",
delim = "\t", escape_double = FALSE,
na = ".", trim_ws = TRUE)

## Rows: 1073 Columns: 48
## -- Column specification -----
## Delimiter: "\t"
## chr (5): sexe, sitfamil, csp, codeqlt, cartevp
## dbl (43): matricul, departem, ptvente, age, ancienne, nbimpaye, mtrejet, nbo...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
on affiche un résumé statistique de l'ensemble des variables
```

```
summary(VisaPremier)
```

##	matricul	departem	ptvente	sexe
##	Min. : 113333	Min. : 6.00	Min. : 1.000	Length:1073
##	1st Qu.: 860436	1st Qu.: 31.00	1st Qu.: 1.000	Class :character
##	Median : 1948586	Median : 31.00	Median : 1.000	Mode :character
##	Mean : 2489307	Mean : 41.18	Mean : 1.664	
##	3rd Qu.: 3901594	3rd Qu.: 61.25	3rd Qu.: 2.000	
##	Max. : 7589439	Max. : 97.00	Max. : 7.000	
##		NA's : 7		
##	age	sitfamil	ancienne	csp
##	Min. : 18.00	Length:1073	Min. : 1.0	Length:1073
##	1st Qu.: 33.00	Class :character	1st Qu.: 45.0	Class :character
##	Median : 43.00	Mode :character	Median : 136.0	Mode :character
##	Mean : 42.53		Mean : 157.1	
##	3rd Qu.: 52.00		3rd Qu.: 216.0	
##	Max. : 65.00		Max. : 870.0	
##				
##	codeqlt	nbimpaye	mtrejet	nbopguic
##	Length:1073	Min. : 0	Min. : -51.00000	Min. : 0.000
##	Class :character	1st Qu.: 0	1st Qu.: 0.00000	1st Qu.: 0.000
##	Mode :character	Median : 0	Median : 0.00000	Median : 1.000
##		Mean : 0	Mean : -0.07269	Mean : 1.505
##		3rd Qu.: 0	3rd Qu.: 0.00000	3rd Qu.: 2.000
##		Max. : 0	Max. : 0.00000	Max. : 28.000
##				

```

##      moycred3      aveparmo      endette      engagemt
## Min.   :    0.00  Min.   :    0  Min.   : 0.000  Min.   :    0
## 1st Qu.:    3.00  1st Qu.:    0  1st Qu.: 0.000  1st Qu.:    0
## Median :   12.00  Median : 6017  Median : 0.000  Median :    0
## Mean   :   47.63  Mean   : 57249  Mean   : 5.457  Mean   : 77316
## 3rd Qu.:   27.00  3rd Qu.: 57818  3rd Qu.: 6.000  3rd Qu.: 34927
## Max.   :19579.00  Max.   :970000  Max.   :99.000  Max.   :3472938
##
##      engagemc      engagemm      nbcptvue      moysold3
## Min.   :    0  Min.   :    0  Min.   :0.000  Min.   : -70050
## 1st Qu.:    0  1st Qu.:    0  1st Qu.:1.000  1st Qu.:   434
## Median :    0  Median :    0  Median :1.000  Median :  4371
## Mean   :  4199  Mean   : 20230  Mean   :1.028  Mean   : 10674
## 3rd Qu.:    0  3rd Qu.:    0  3rd Qu.:1.000  3rd Qu.: 11034
## Max.   :500780  Max.   :1618242  Max.   :4.000  Max.   :241827
##
##      moycredi      agemvt      nbop      mtfactur
## Min.   :    0.00  Min.   :    0.00  Min.   :    0  Min.   :    0
## 1st Qu.:    2.00  1st Qu.: 13.00  1st Qu.:    6  1st Qu.:    0
## Median :   11.00  Median : 13.00  Median :   25  Median :    0
## Mean   :   25.91  Mean   : 19.06  Mean   :   29  Mean   : 23379
## 3rd Qu.:   24.00  3rd Qu.: 14.00  3rd Qu.:   43  3rd Qu.:  3500
## Max.   :4079.00  Max.   :944.00  Max.   :262  Max.   :1331530
##
##      NA's :6
##      engageml      nbvie      mtvie      nbeparmo
## Min.   :    0  Min.   : 0.0000  Min.   :    0  Min.   : 0.0000
## 1st Qu.:    0  1st Qu.: 0.0000  1st Qu.:    0  1st Qu.: 0.0000
## Median :    0  Median : 0.0000  Median :    0  Median :1.0000
## Mean   :  52888  Mean   : 0.2395  Mean   : 35915  Mean   :1.473
## 3rd Qu.:    0  3rd Qu.: 0.0000  3rd Qu.:    0  3rd Qu.:2.0000
## Max.   :3472938  Max.   :13.0000  Max.   :5449561  Max.   :9.0000
##
##      mteparmo      nbeparlo      mteparlo      nblivret
## Min.   :    0  Min.   :0.0000  Min.   :    0  Min.   :0.0000
## 1st Qu.:    0  1st Qu.:0.0000  1st Qu.:    0  1st Qu.:0.0000
## Median :   6017  Median :0.0000  Median :    0  Median :1.0000
## Mean   :  75442  Mean   :0.6524  Mean   : 32184  Mean   :0.7586
## 3rd Qu.:  58390  3rd Qu.:1.0000  3rd Qu.: 23854  3rd Qu.:1.0000
## Max.   :19508920  Max.   :4.0000  Max.   :579603  Max.   :4.0000
##
##      mtlivret      nbeparlt      mteparlt      nbeparte
## Min.   :    0  Min.   :0.00000  Min.   :    0  Min.   :0.000000
## 1st Qu.:    0  1st Qu.:0.00000  1st Qu.:    0  1st Qu.:0.000000
## Median :   127  Median :0.00000  Median :    0  Median :0.000000
## Mean   : 20740  Mean   :0.05871  Mean   :  4325  Mean   :0.002796
## 3rd Qu.: 15544  3rd Qu.:0.00000  3rd Qu.:    0  3rd Qu.:0.000000
## Max.   :970000  Max.   :6.00000  Max.   :559559  Max.   :1.000000
##
##      mteparte      nbbon      mtbon      nbpaiecb
## Min.   :    0.00  Min.   :0.000000  Min.   :    0  Min.   : 0.0
## 1st Qu.:    0.00  1st Qu.:0.000000  1st Qu.:    0  1st Qu.: 0.0
## Median :    0.00  Median :0.000000  Median :    0  Median : 9.0
## Mean   :   19.71  Mean   :0.000932  Mean   :  18173  Mean   :11.5
## 3rd Qu.:    0.00  3rd Qu.:0.000000  3rd Qu.:    0  3rd Qu.:18.0

```

```
## Max. :21149.00 Max. :1.000000 Max. :19500000 Max. :69.0
## NA's :278
## nbc b nbc bptar avtscpte aveparfi
## Min. :0.00 Min. :0.0000 Min. : 0 Min. : 0
## 1st Qu.:0.00 1st Qu.:0.0000 1st Qu.: 3184 1st Qu.: 0
## Median :1.00 Median :0.0000 Median : 23993 Median : 0
## Mean :1.07 Mean :0.1361 Mean : 146819 Mean : 50727
## 3rd Qu.:2.00 3rd Qu.:0.0000 3rd Qu.: 114807 3rd Qu.: 500
## Max. :5.00 Max. :4.0000 Max. :19856243 Max. :7066619
##
## cartevp sexer cartevpr nbjdebit
## Length:1073 Min. :0.0000 Min. :0.0000 Min. : 0.00
## Class :character 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 0.00
## Mode :character Median :0.0000 Median :0.0000 Median : 0.00
## Mean :0.3774 Mean :0.3346 Mean : 12.08
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 10.00
## Max. :1.0000 Max. :1.0000 Max. :134.00
##
```

La variable matricule est codée à tort comme une variable numérique, on la transforme en une variable catégorielle

```
VisaPremier$matricul = as.factor(VisaPremier$matricul)
```

Cette variable ne nous sert à rien dans notre analyse, on la supprime

```
VisaPremier$matricul = NULL
```

La variable département doit être codée en factor également

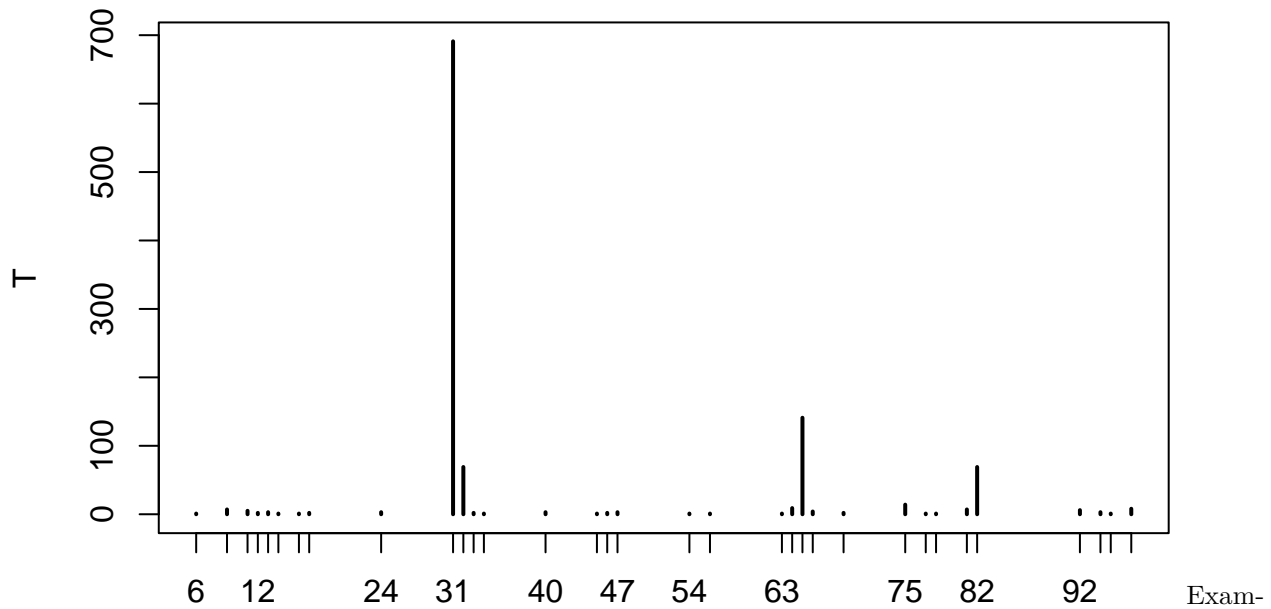
```
VisaPremier$departem = as.factor(VisaPremier$departem)
```

On va observer la distribution de cette variable à l'aide d'un tableau de contingence

```
T = table(VisaPremier$departem)
print(T)
```

```
##
## 6 9 11 12 13 14 16 17 24 31 32 33 34 40 45 46 47 54 56 63
## 1 7 5 2 3 1 1 2 3 691 69 2 1 3 1 2 3 1 1 1
## 64 65 66 69 75 77 78 81 82 92 94 95 97
## 9 141 4 2 14 1 1 7 69 6 3 1 8
```

```
plot(T)
```



Examinons un peu plus en détail le contenu de cette variable

```
summary(VisaPremier$departem)
```

```
##      6      9     11     12     13     14     16     17     24     31     32     33     34     40     45     46
##      1      7      5      2      3      1      1      2      3    691     69      2      1      3      1      2
##     47     54     56     63     64     65     66     69     75     77     78     81     82     92     94     95
##      3      1      1      1      9    141      4      2     14      1      1      7     69      6      3      1
##    97 NA's
##      8      7
```

Il y a 7 données manquantes. On a plusieurs choix : - supprimer ces individus (ok si ils sont peu nombreux) - on peut chercher à remplacer les valeurs manquantes par une valeur (à définir). On parle d'imputation de données manquantes. Un choix peut-être de remplacer cette données manquantes par la catégorie la plus fréquemment observée (mode de la distribution)

```
VisaPremier$departem[is.na(VisaPremier$departem)]="31"
```

Les variables ptvente et sexe, csp et sitfamil sont catégorielles, il faut les définir comme telle

```
VisaPremier$sexe=as.factor(VisaPremier$sexe)
VisaPremier$ptvente=as.factor(VisaPremier$ptvente)
VisaPremier$csp=as.factor(VisaPremier$csp)
VisaPremier$sitfamil=as.factor(VisaPremier$sitfamil)
VisaPremier$codeqlt=as.factor(VisaPremier$codeqlt)
```

La variable codeqlt a beaucoup de données manquantes. Les imputer par le mode de la distribution déséquilibrerait trop le jeu de données, on va faire l'imputation en respectant les proportions observées dans chaque catégorie

```
set.seed(1)
imputation=sample(levels(VisaPremier$codeqlt),133,replace=TRUE,prob=table(VisaPremier$codeqlt))
VisaPremier$codeqlt[is.na(VisaPremier$codeqlt)]=imputation
```

Le 'set.seed' permet d'initialiser la graine du générateur aléatoire, afin de créer un code "reproductible" (qui régénérera les mêmes valeurs aléatoires à chaque fois qu'on le relancera)

La variable nbimpaye est constante, elle n'a aucune utilité dans l'analyse statistique. On la supprime

```
VisaPremier$nbimpaye=NULL
```

La variable agemvt est quantitative, et elle a 6 données manquantes. On va également les imputer, par typiquement la médiane (moins sensible aux données extrêmes). Idem pour la variable nbpaiecb

```
VisaPremier$agemvt[is.na(VisaPremier$agemvt)]=median(VisaPremier$agemvt,na.rm = TRUE)
VisaPremier$nbpaiecb[is.na(VisaPremier$nbpaiecb)]=median(VisaPremier$nbpaiecb,na.rm = TRUE)
```

La variable nbbon a une distribution particulière : seul un individu sur les 1073 a la valeur 1, tous les autres sont à 0. Cherchons quel est cet individu :

```
which(VisaPremier$nbbon==1)
```

```
## [1] 2
```

C'est l'individu n°2, examinons ses données (ligne n° 2 du data frame VisaPremier, et toutes les colonnes):

```
VisaPremier[2,]
```

```
## # A tibble: 1 x 46
##   departem ptvte sexe   age sitfamil ancienne csp   codeqlt mtrejet nbopguic
##   <fct>    <fct>  <fct> <dbl> <fct>      <dbl> <fct> <fct>    <dbl>    <dbl>
## 1 82      6      Shom   52 Fmar      270 Pcad  A      0      4
## # i 36 more variables: moycred3 <dbl>, aveparmo <dbl>, endette <dbl>,
## #   engagemt <dbl>, engagemc <dbl>, engagemm <dbl>, nbcptvue <dbl>,
## #   moysold3 <dbl>, moycredi <dbl>, agemvt <dbl>, nbop <dbl>, mtfactur <dbl>,
## #   engageml <dbl>, nbvie <dbl>, mtvie <dbl>, nbeparmo <dbl>, mteparmo <dbl>,
## #   nbeparlo <dbl>, mteparlo <dbl>, nbilivret <dbl>, mtlivret <dbl>,
## #   nbeparlt <dbl>, mteparlt <dbl>, nbeparte <dbl>, mteparte <dbl>,
## #   nbbon <dbl>, mtbon <dbl>, nbpaiecb <dbl>, nbcb <dbl>, nbcbptar <dbl>, ...
```

C'est un client relativement atypique (et riche), gardons le en tête pour nos analyses futures.

Il faut enfin coder la variable cartevp en catégorielle :

```
VisaPremier$cartevp=as.factor(VisaPremier$cartevp)
```

Les deux variables cartevpr et sexer sont redondantes (recodage en 0/1 de variables binaires). On les supprime:

```
VisaPremier$cartevpr=NULL
VisaPremier$sexer=NULL
```

Notre jeu de données est maintenant propre, on a supprimé les variables non informatives (constantes ou redondantes), on a imputé les données manquantes, corrigé les erreurs de codage (déclarer les variables catégorielles comme telles)... on a même mis le doigt sur un individu potentiellement atypique (le n°2)

Introduction à la notion d'estimation

On s'intéresse à la distribution d'une variable aléatoire X (ex : âge) sur une population donnée (ex : l'ensemble des clients de la banque). La distribution de X est notamment caractérisée par son espérance $E[X]$ (valeur moyenne théorique sur l'ensemble de la population). On ne dispose des données pour l'ensemble de la population, mais uniquement d'un échantillon de celle-ci (que l'on suppose représentatif).

La meilleure façon d'obtenir un échantillon représentatif, est de choisir les individus de façon aléatoire, indépendante et identiquement distribuée (tout le monde a la même chance d'être tiré.)

A partir de cet échantillon, on va chercher à tirer de l'information sur $E[X]$ dans la population totale, et notamment à en donner une estimation. On parle d'**inférence statistique**

Tirons un sous-échantillon de taille 20 de notre base de données (pour la variable âge)

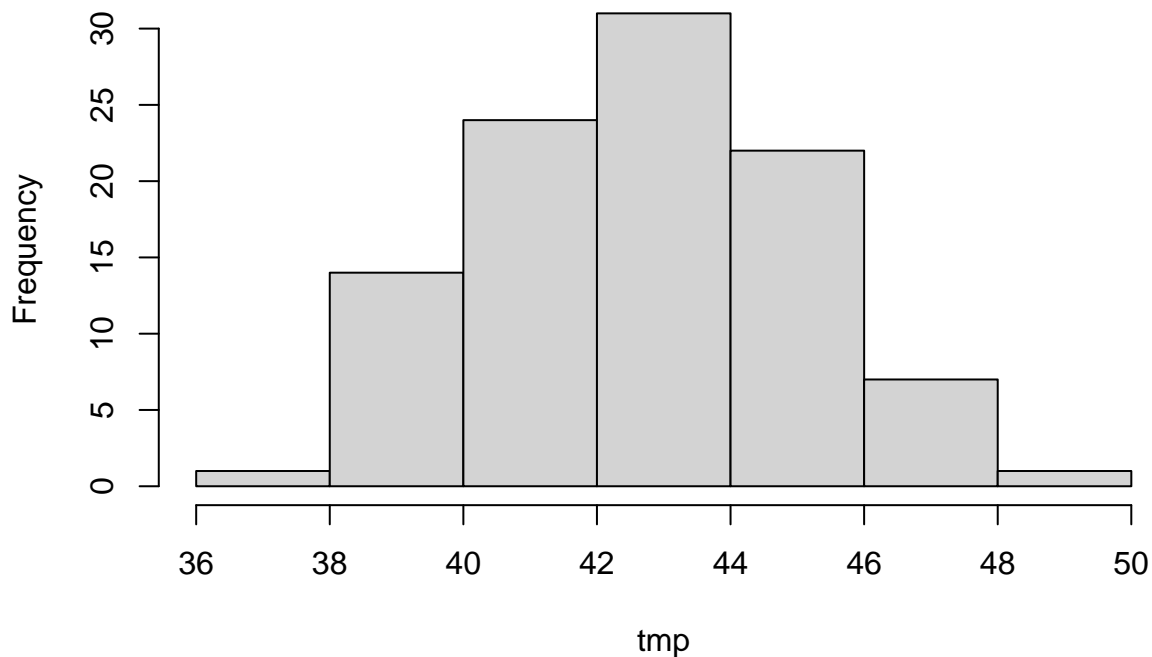
```
x = VisaPremier$age[sample(1:1073,20)]
mean(x)
```

```
## [1] 45.6
```

Si on répète cela plusieurs fois, on constate que l'estimation de l'âge de la population à partir des données de l'échantillon varie

```
tmp=NULL
for (i in 1:100){
  x = VisaPremier$age[sample(1:1073,20)]
  tmp=c(tmp,mean(x))
}
hist(tmp,main='distribution de l âge moyen estime pour 100 echantillons de taille 20')
```

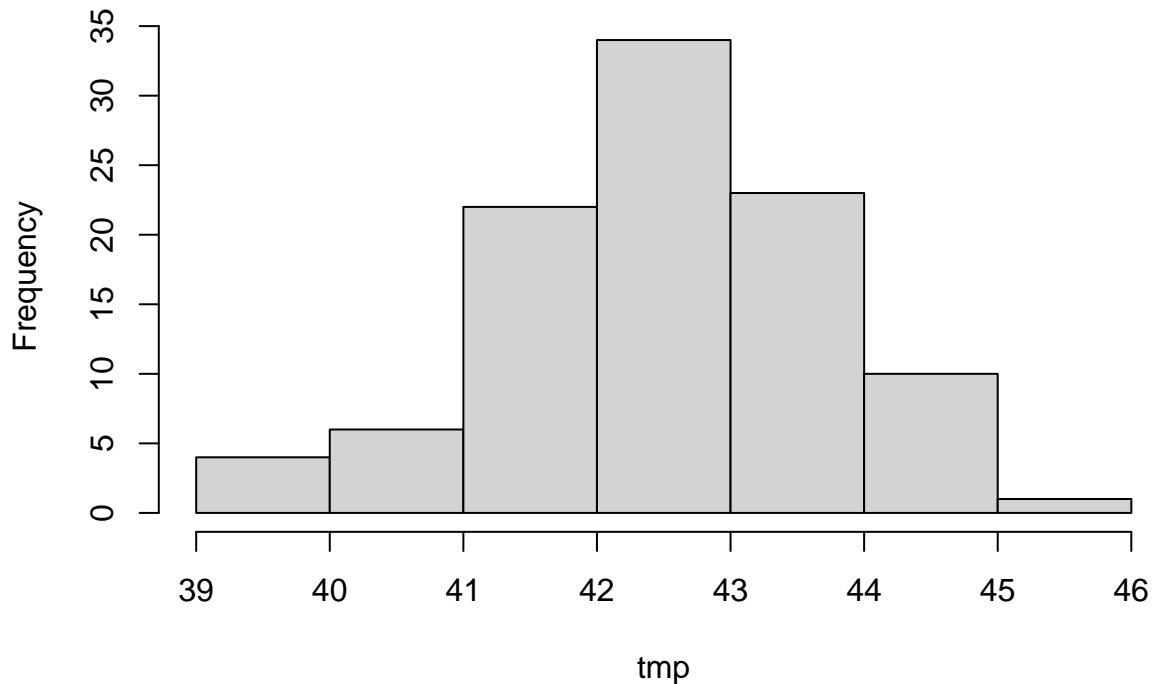
distribution de l âge moyen estime pour 100 echantillons de taille 20



Faisons de même avec des échantillons de taille 100

```
tmp=NULL
for (i in 1:100){
  x = VisaPremier$age[sample(1:1073,100)]
  tmp=c(tmp,mean(x))
}
hist(tmp,main='distribution de l âge moyen estime pour 100 echantillons de taille 100')
```

distribution de l'âge moyen estimé pour 100 échantillons de taille 100



L'estimation que l'on fait varie moins lorsque la taille d'échantillon grandit.

Revenons à notre client de la banque. Si on veut estimer l'âge moyen des clients de la population de la banque, on peut simplement utiliser la moyenne des âges de l'échantillon

```
mean(VisaPremier$age)
```

```
## [1] 42.53215
```

Question : estimer l'âge moyen des hommes et des femmes pour l'ensemble des clients de la banque.

```
summary(VisaPremier$sexe)
```

```
## Sfem Shom
```

```
## 405 668
```

```
mean(VisaPremier$age[ VisaPremier$sexe=='Shom'])
```

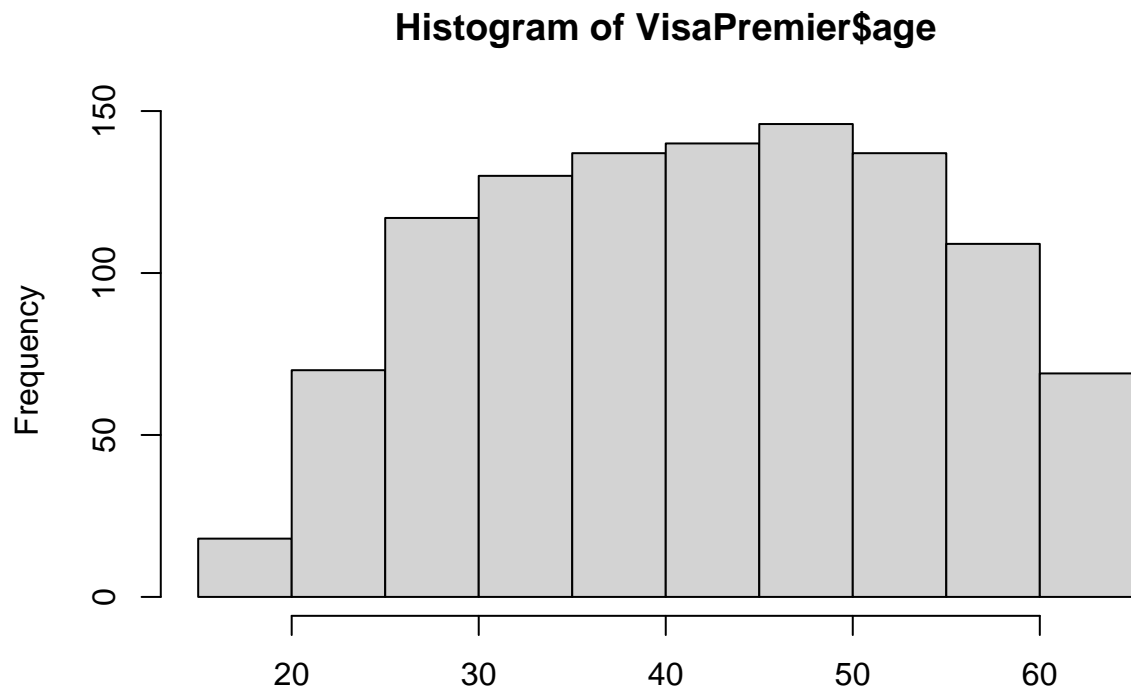
```
## [1] 42.91317
```

```
mean(VisaPremier$age[ VisaPremier$sexe=='Sfem'])
```

```
## [1] 41.9037
```

On peut aussi estimer la distribution de probabilité de la variable âge, en utilisant un histogramme, qui est une estimation de la densité de probabilité d'une variable quantitative

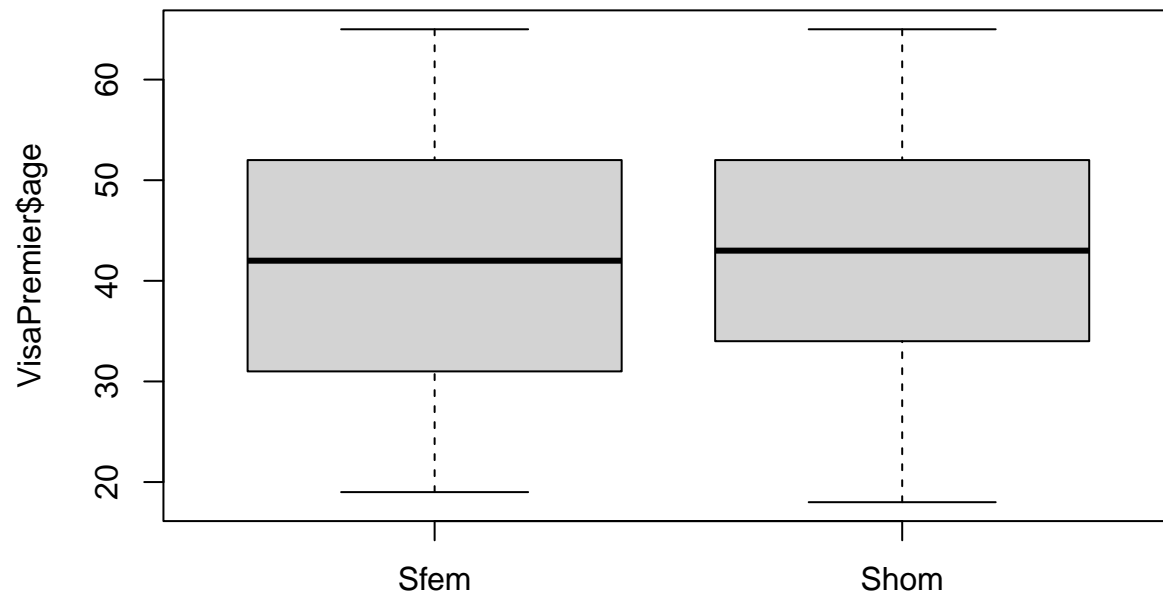
```
hist(VisaPremier$age)
```



VisaPremier\$age

On utilise aussi parfois les boxplots, surtout pour regarder si la distribution change suivant les catégories d'une variable qualitative

```
boxplot(VisaPremier$age~VisaPremier$sexe)
```



VisaPremier\$sexe

Lorsqu'on estime la moyenne théorique à partir de la moyenne de l'échantillon, on réalise une estimation ponctuelle, mais qui ne contient pas d'information sur l'incertitude liée à cet estimation.

Estimation par intervalle de confiance

On va définir un intervalle auquel la moyenne théorique (inconnue) appartient avec un certain niveau de confiance

```
tmp=t.test(VisaPremier$age,conf.level = 0.95)
print(tmp$conf.int)
```

```
## [1] 41.81902 43.24529
## attr(,"conf.level")
## [1] 0.95
```

Cet intervalle de confiance nous dit que l'âge moyen de la population de client appartient à l'intervalle [41.8,43.2] avec probabilité 0.95.

Remarques : 1. plus la taille de l'échantillon est grande, plus l'intervalle est précis (de petite longueur). En effet, avec un échantillon de taille 20, la largeur est beaucoup plus grande, l'estimation moins précise :

```
tmp=t.test(VisaPremier$age[sample(1:1073,30)],conf.level = 0.95)
print(tmp$conf.int)
```

```
## [1] 38.46478 46.20189
## attr(,"conf.level")
## [1] 0.95
```

Rq : l'erreur standard (de la moyenne empirique) est son écart-type (égal à l'écart-type des données initiales divisé par la taille d'échantillon)

2. Lorsqu'on augmente le niveau de confiance

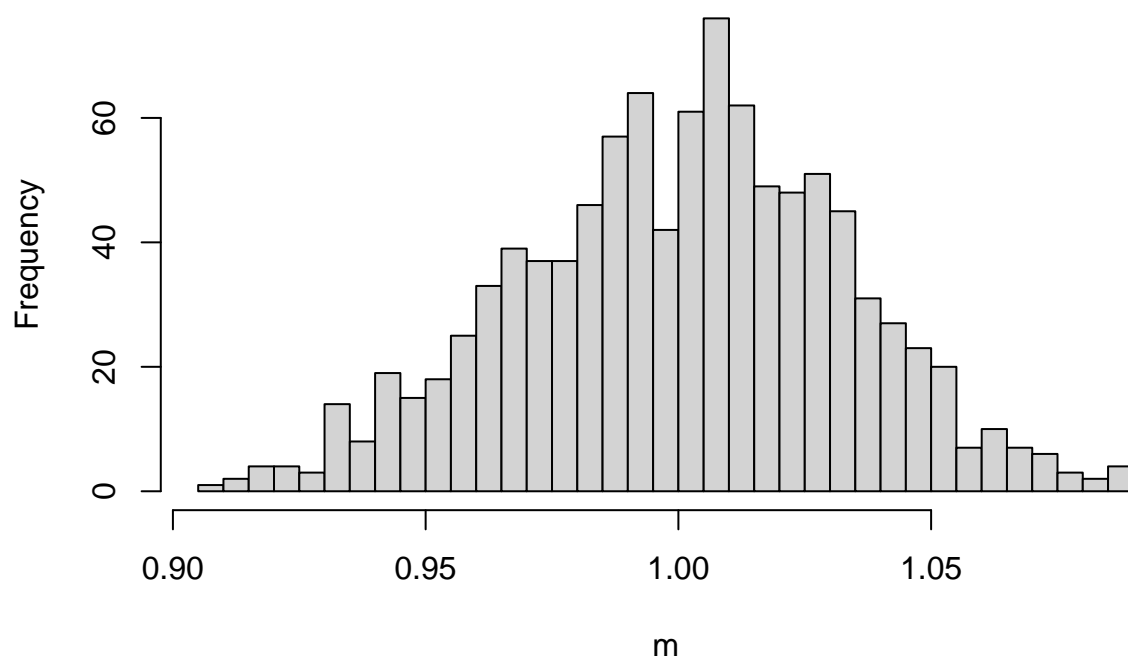
```
tmp=t.test(VisaPremier$age,conf.level = 0.999)
print(tmp$conf.int)
```

```
## [1] 41.33293 43.73137
## attr(,"conf.level")
## [1] 0.999
```

3. Ces intervalles de confiance sont valides dès lors que la taille d'échantillon est suffisamment grande (typiquement au moins 30), et ce quel que soit la nature des données de départ (en particulier quelque soit leur distribution). En effet le théorème central limite nous assure que les moyennes empiriques sont distribuées suivant des lois normales, peu importe la nature de la distribution des données elle-même (ci-dessous un exemple avec des données simulées suivant une loi exponentielle)

```
m=NULL
for (i in 1:1000){
  m=c(m,mean(rexp(1000)))
}
hist(m,breaks=30)
```

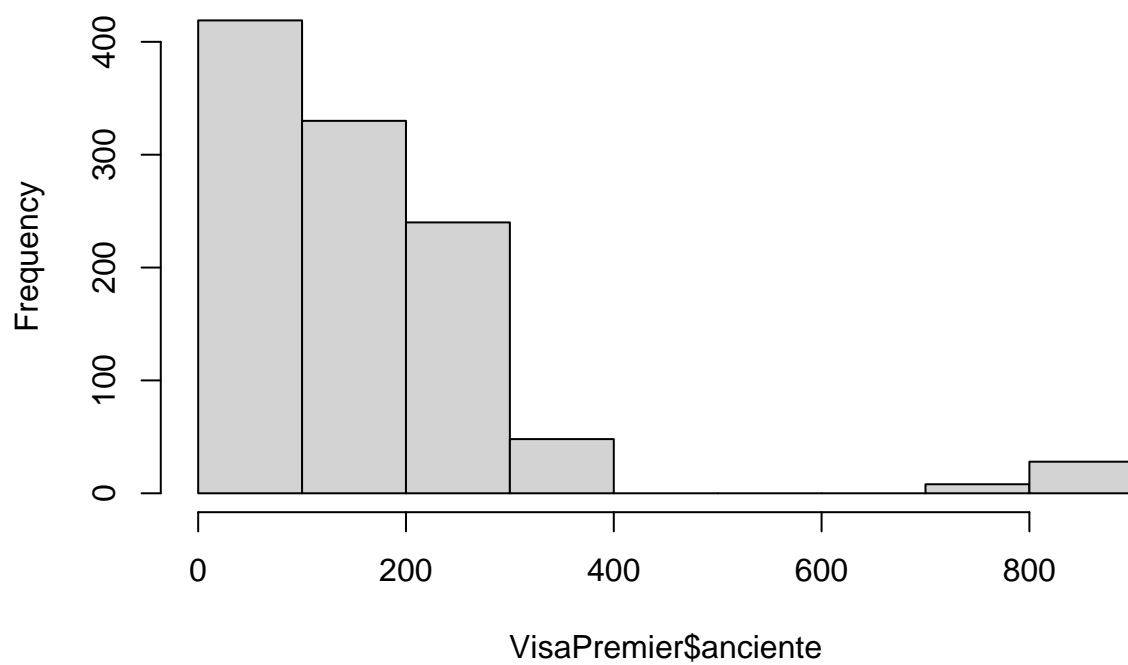
Histogram of m



Exercice sur l'ancienneté

```
hist(VisaPremier$anciennete)
```

Histogram of VisaPremier\$anciennete



```
mean(VisaPremier$anciante)
```

```
## [1] 157.1174
```

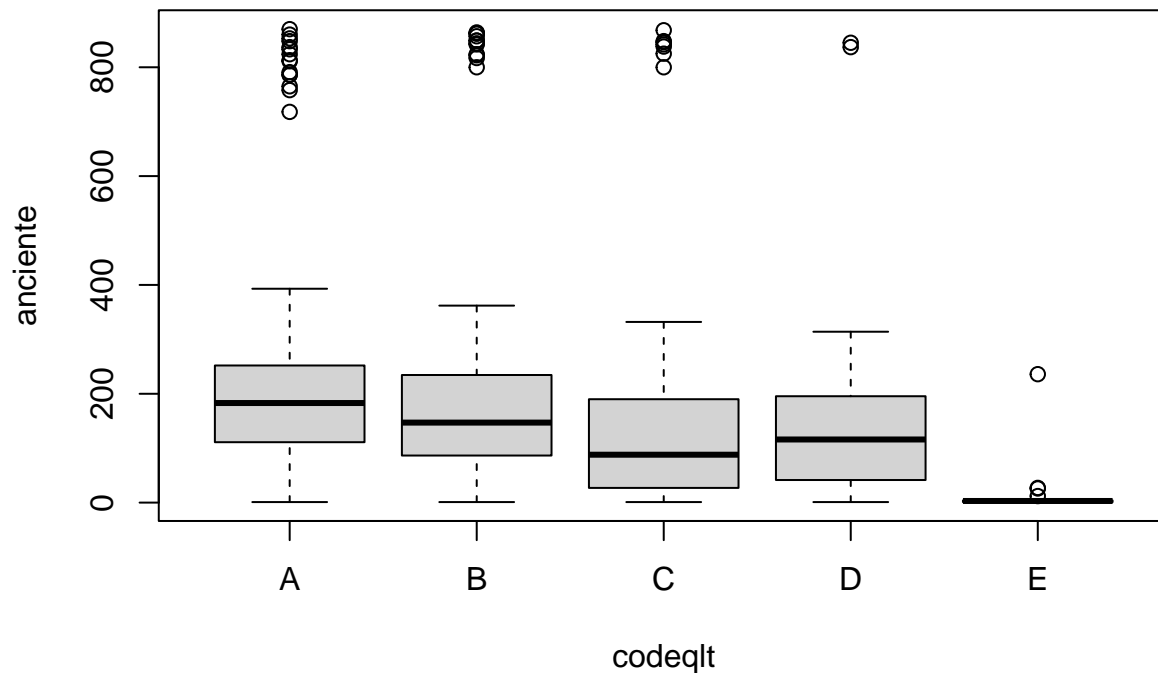
```
t.test(VisaPremier$anciante,conf.level=0.95)$conf.int
```

```
## [1] 147.7689 166.4660
```

```
## attr("conf.level")
```

```
## [1] 0.95
```

```
boxplot(anciante~codeqlt,data=VisaPremier)
```



```
cat('IC ancienneté pour le codqlt A :',t.test(VisaPremier$anciante[VisaPremier$codeqlt=="A"],  
conf.level=0.95)$conf.int,'\n')
```

```
## IC ancienneté pour le codqlt A : 188.2953 233.9748
```

```
cat('IC ancienneté pour le codqlt B :',t.test(VisaPremier$anciante[VisaPremier$codeqlt=="B"],  
conf.level=0.95)$conf.int,'\n')
```

```
## IC ancienneté pour le codqlt B : 161.8177 195.026
```

```
cat('IC ancienneté pour le codqlt C :',t.test(VisaPremier$anciante[VisaPremier$codeqlt=="C"],  
conf.level=0.95)$conf.int,'\n')
```

```
## IC ancienneté pour le codqlt C : 108.4762 144.8
```

```
cat('IC ancienneté pour le codqlt D :',t.test(VisaPremier$anciante[VisaPremier$codeqlt=="D"],  
conf.level=0.95)$conf.int,'\n')
```

```
## IC ancienneté pour le codqlt D : 115.0906 147.7367
```

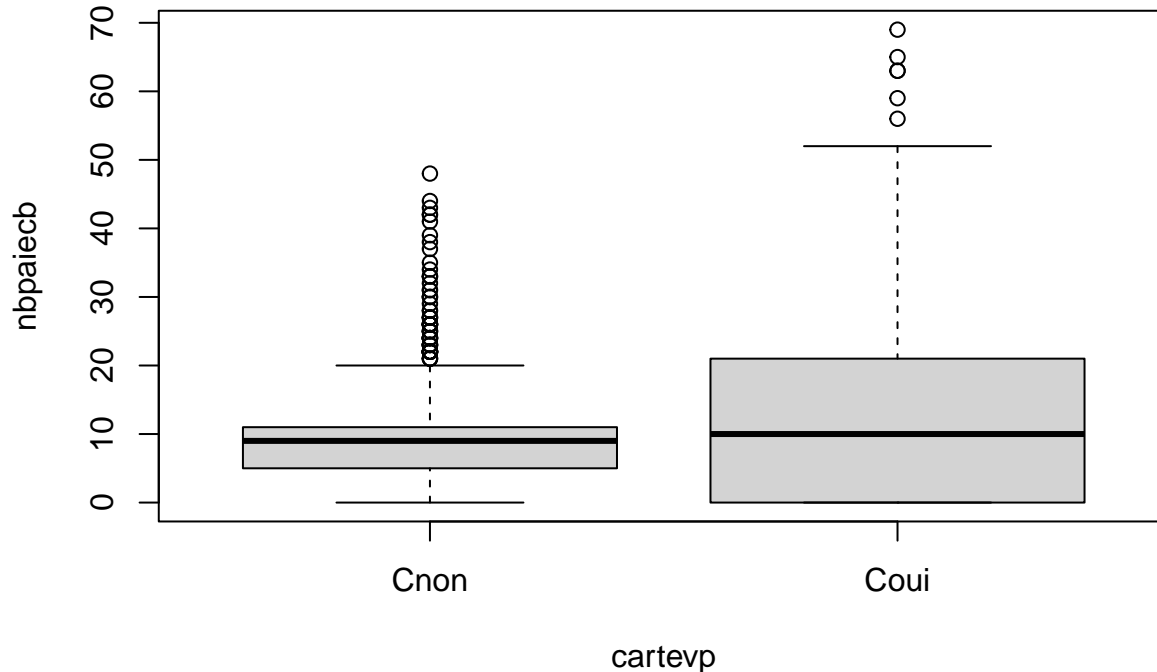
```
cat('IC ancienneté pour le codqlt E :',t.test(VisaPremier$anciante[VisaPremier$codeqlt=="E"],  
conf.level=0.95)$conf.int,'\n')
```

```
## IC ancienneté pour le codqlt E : -1.130522 18.15093
```

Test statistique de comparaison de deux populations

Question : le nombre de paiements par CB est-il en moyenne différent pour les personnes qui possèdent la carte VP et celles qui ne la possèdent pas ?

```
boxplot(nbpaiecb~cartevp,data=VisaPremier)
```



```
t.test(VisaPremier$nbpaiecb[VisaPremier$cartevp=="Coui"])$conf.int
```

```
## [1] 12.18306 15.13170
## attr("conf.level")
## [1] 0.95
```

```
t.test(VisaPremier$nbpaiecb[VisaPremier$cartevp=="Cnon"])$conf.int
```

```
## [1] 8.86166 10.02349
## attr("conf.level")
## [1] 0.95
```

Pour répondre à cette question, on va utiliser un test statistique, qui consiste à choisir entre deux hypothèses: H0 (hypothèse nulle) : moyenne (cartvp=oui) = moyenne (cartvp=non) H1 (hypothèse alternative) : moyenne (cartvp=oui) <> moyenne (cartvp=non)

qui s'appelle le test de Student (qui peut être utilisé uniquement si les échantillons sont suffisamment grands, i.e. $n \geq 30$)

```
t.test(nbpaiecb ~ cartevp, data=VisaPremier)
```

```
##
## Welch Two Sample t-test
##
## data: nbpaiecb by cartevp
## t = -5.2296, df = 472.47, p-value = 2.557e-07
## alternative hypothesis: true difference in means between group Cnon and group Coui is not equal to 0
## 95 percent confidence interval:
## -5.798502 -2.631107
```

```
## sample estimates:
## mean in group Cnon mean in group Coui
##          9.442577          13.657382
```

La **p-value** est la probabilité de se tromper en rejetant l'hypothèse H_0 au profit de H_1 . Autrement dit, c'est la probabilité de se tromper en concluant à une différence entre les deux populations.

Ici la p-value est très faible ($2.5e-7$), on va pouvoir conclure qu'il existe une différence significative entre les nombres de paiements moyens des personnes qui possèdent la carte VP et ceux qui ne l'ont pas. Généralement on convient de rejeter H_0 quand cette pvalue est $<$ à 0.05 (mais attention, ça dépend des domaines d'applications...)

Remarque : - on a fait un test bilatéral ($H_1 : \mu_1 <> \mu_2$), mais en pratique, on est plutôt intéressé par des tests unilatéraux : ici, on veut vérifier si ce que l'on voit sur les moyennes (moyenne de l'échantillon avec CB plus grand que l'autre) est significatif (et non du au hasard). Le test unilatéral va consister à tester : H_0 (hypothèse nulle) : moyenne (cartvp=non) = moyenne (cartvp=oui) H_1 (hypothèse alternative) : moyenne (cartvp=non) $<$ moyenne (cartvp=oui)

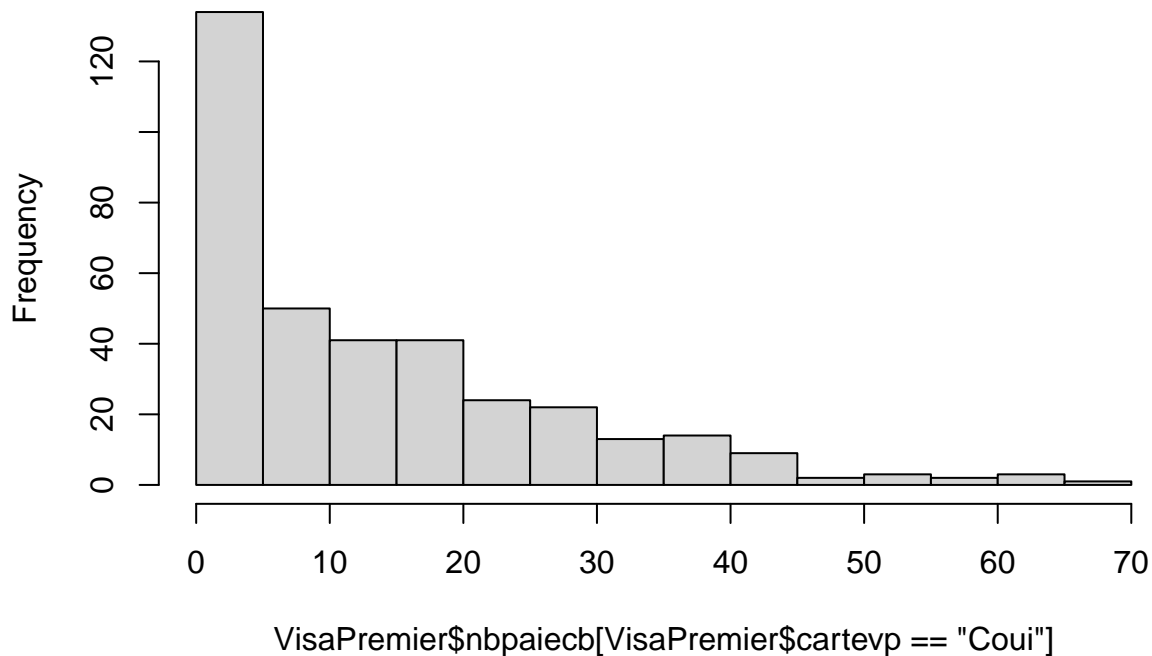
```
t.test(nbpaiecb ~ cartevp, data=VisaPremier,alternative="less")
```

```
##
## Welch Two Sample t-test
##
## data: nbpaiecb by cartevp
## t = -5.2296, df = 472.47, p-value = 1.279e-07
## alternative hypothesis: true difference in means between group Cnon and group Coui is less than 0
## 95 percent confidence interval:
##      -Inf -2.886524
## sample estimates:
## mean in group Cnon mean in group Coui
##          9.442577          13.657382
```

2. si les échantillons sont petits (<30) mais qu'ils suivent une loi normale, alors le test de Student reste valide. La question est comment que les données (de chaque échantillon) suivent bien une loi normale ? Il existe un test pour cela, le test de Shapiro-Wilk H_0 : l'échantillon suit une loi normale H_1 : l'échantillon ne suit pas une loi normale

```
hist(VisaPremier$nbpaiecb[VisaPremier$cartevp=="Coui"])
```

Histogram of VisaPremier\$nbpaiecb[VisaPremier\$cartevp == "Coui"]



```
shapiro.test(VisaPremier$nbpaiecb[VisaPremier$cartevp=="Coui"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  VisaPremier$nbpaiecb[VisaPremier$cartevp == "Coui"]
## W = 0.86518, p-value < 2.2e-16
```

ici la p-value très faible nous dit que l'échantillon n'est pas du tout Gaussien. Heureusement, l'échantillon étant de grande taille on n'en a pas besoin ici. Rq sur la fiabilité du test de Shapiro :

```
x=rnorm(20)
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.96849, p-value = 0.7228
```

Attention, si l'échantillon est de trop petite taille (<15), le test de Shapiro est trop peu fiable, il conclut trop souvent à tort que l'échantillon est gaussien (il est trop peu puissant, il accepte trop souvent H_0 à tort)

3. Enfin, si l'échantillon n'est ni de grande taille, ni Gaussien (ou que la taille d'échantillon ne permet pas de tester la normalité). On fait alors un test de Wilcoxon

```
wilcox.test(nbpaiecb ~ cartevp, data=VisaPremier, alternative="less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  nbpaiecb by cartevp
## W = 114292, p-value = 0.00165
```

```
## alternative hypothesis: true location shift is less than 0
```

Le test de Wilcoxon est moins puissant que le test de Student (il aura plus de mal à rejeter H_0 , i.e. à conclure à une différence significative), donc on utilisera lorsque cela est possible de préférence le test de Student.

Exercice 3

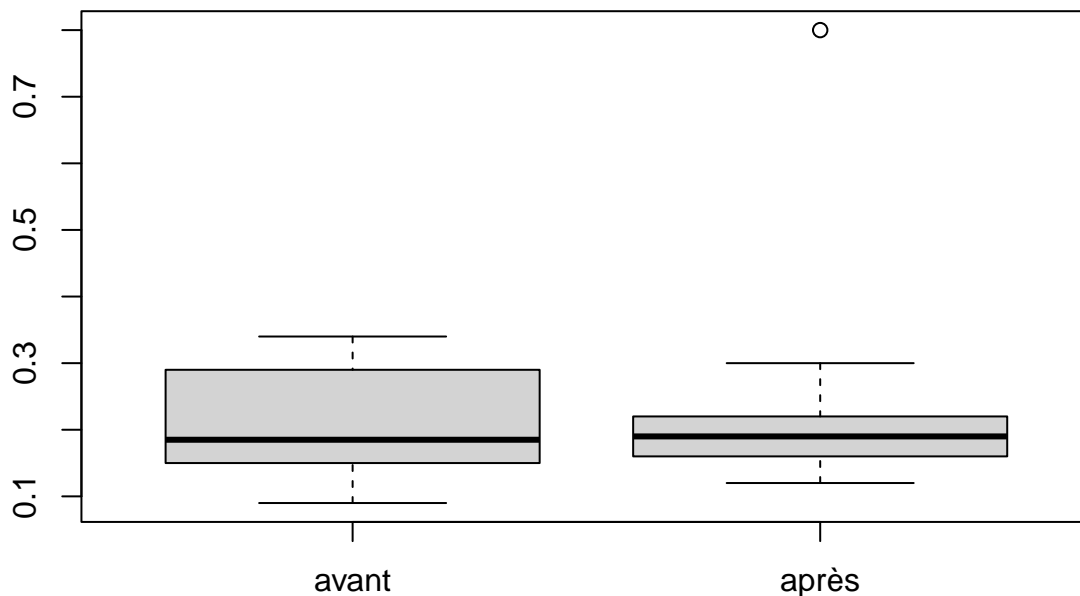
On souhaite tester l'effet d'un médicament censé réduire le taux de le cholestérol. On a mesuré le taux de cholestérol (g/l) chez 10 patients, avant la prise de ce médicament, et une semaine après l'avoir pris. Voici les taux obtenus :

```
Avant=c(0.1,0.2,0.15,0.3,0.34,0.16,0.09,0.24,0.17,0.29)
Après=c(0.8,0.18,0.12,0.2,0.3,0.21,0.12,0.16,0.17,0.22)
cholesterol=data.frame(avant=Avant,après=Après)
```

Le médicament a-t-il un effet ?

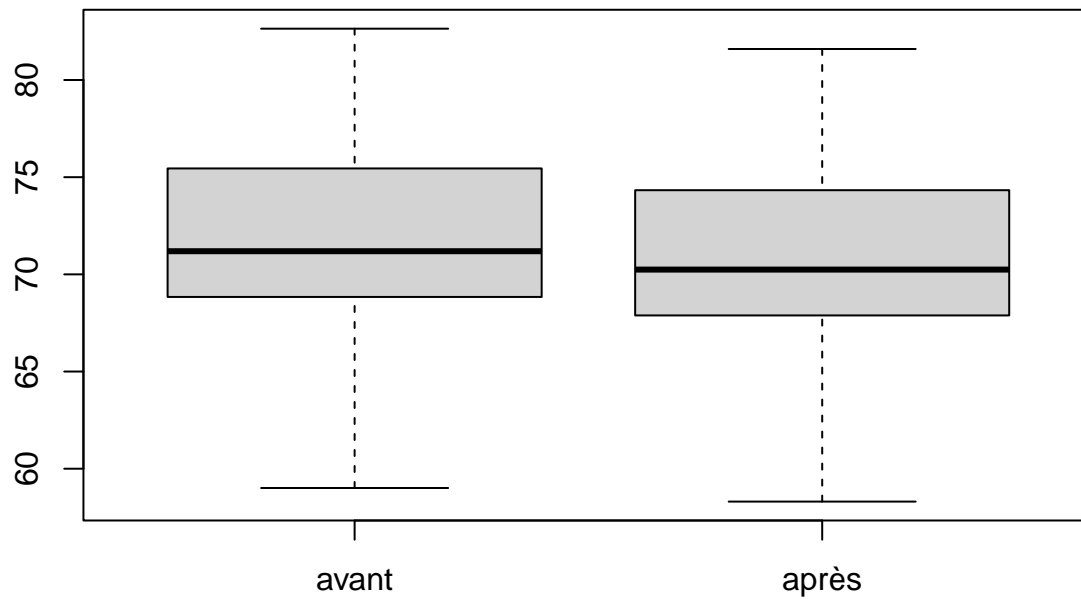
La première chose est de représenter graphiquement les données

```
boxplot(cholesterol)
```



Avant de faire le test, regardons ce qui se passe sur une simulation. On s'intéresse au poids d'individus avant et après un régime

```
poids_avant=rnorm(50,70,5)
poids_après=poids_avant-rnorm(50,1,0.1)
poids=data.frame(avant=poids_avant,après=poids_après)
boxplot(poids)
```

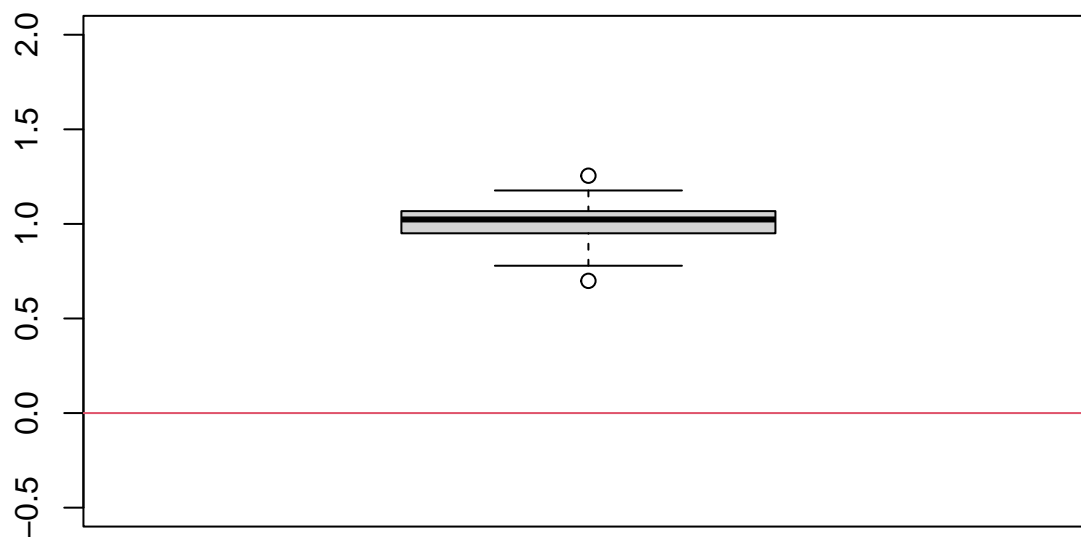


```
t.test(poids$avant,poids$après,alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  poids$avant and poids$après
## t = 0.89081, df = 97.996, p-value = 0.1876
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.8698408      Inf
## sample estimates:
## mean of x mean of y
##  71.39025  70.38361
```

Comme les données sont appariées (ce sont les mêmes individus avant et après), il nous faut travailler avec la différence avant - après :

```
boxplot(poids_avant-poids_après,ylim=c(-0.5,2))
abline(h=0,col=2)
```



Il suffit de

tester si la différence avant-après est significativement positive (à l'aide d'un test de Student à un unique échantillon, en testant si la moyenne théorique de la différence est significativement positive)

```
t.test(poids_avant-poids_après,mu=0,alternative="greater")
```

```
##
## One Sample t-test
##
## data: poids_avant - poids_après
## t = 64.778, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  0.9805902      Inf
## sample estimates:
## mean of x
##  1.006644
```

C'est ce que fait le test de Student de comparaison de deux populations lorsqu'on lui indique que les échantillons sont appariés

```
t.test(poids$avant,poids$après,alternative="greater",paired=TRUE)
```

```
##
## Paired t-test
##
## data: poids$avant and poids$après
## t = 64.778, df = 49, p-value < 2.2e-16
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  0.9805902      Inf
## sample estimates:
## mean difference
##      1.006644
```

Revenons à notre exercice sur le cholestérol, en prenant en compte le fait que les échantillons sont appariés. On veut comparer deux populations (avant et après la prise du médicament). Quel test utiliser ? Comme les échantillons sont petits (taille 10), on ne peut pas utiliser un test de Student. La taille est même trop petite pour utiliser un test de Shapiro pour vérifier s'ils sont gaussiens

```
wilcox.test(cholesterol$avant,cholesterol$après,paired=TRUE)
```

```
## Warning in wilcox.test.default(cholesterol$avant, cholesterol$après, paired =
## TRUE): cannot compute exact p-value with ties

## Warning in wilcox.test.default(cholesterol$avant, cholesterol$après, paired =
## TRUE): cannot compute exact p-value with zeroes

##
## Wilcoxon signed rank test with continuity correction
##
## data: cholesterol$avant and cholesterol$après
## V = 28.5, p-value = 0.5143
## alternative hypothesis: true location shift is not equal to 0
```

On peut comparer avec un test de Student (même si ses conditions d'applications ne sont pas valides ici)

```
t.test(cholesterol$avant,cholesterol$après,paired=TRUE)
```

```
##
```

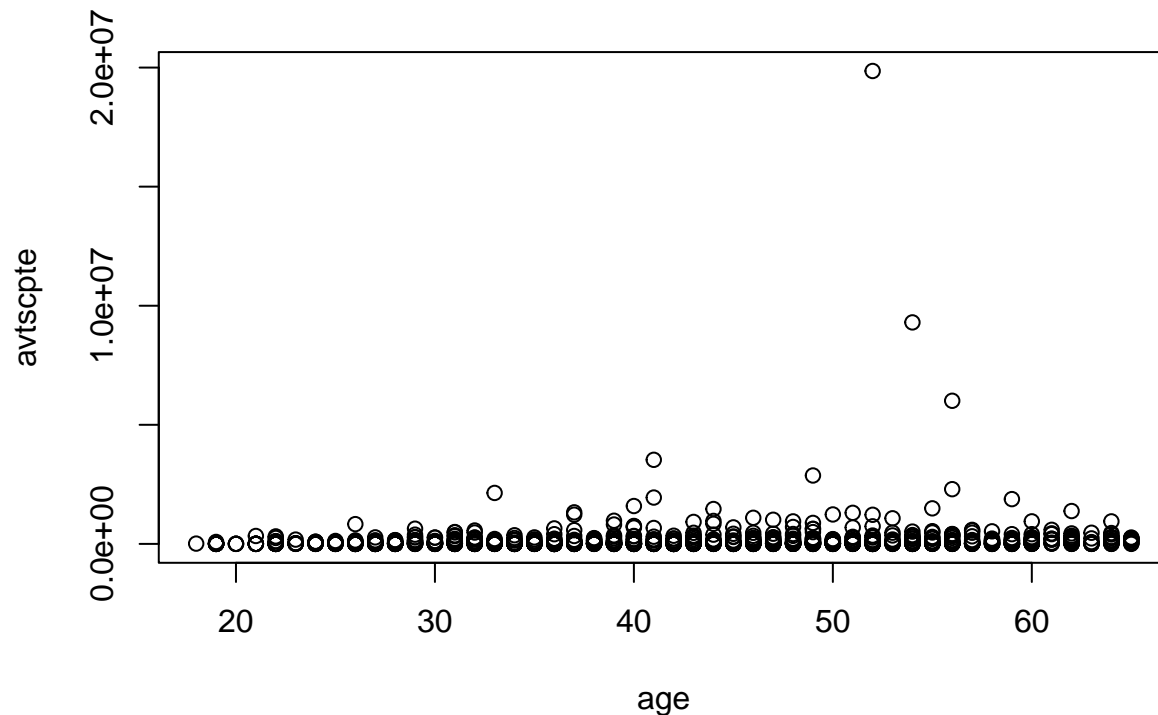
```
## Paired t-test
##
## data: cholesterol$avant and cholesterol$après
## t = -0.59136, df = 9, p-value = 0.5688
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## -0.2123142 0.1243142
## sample estimates:
## mean difference
## -0.044
```

Les deux donnent le même résultats : la pvalue étant grand ($>5\%$), on ne peut pas conclure à une différence significative. On ne peut pas affirmer que le médicament à un effet: - soit il n'a pas d'effet - soit la taille est trop petite pour réussir à montrer qu'il a un effet. On peut rien conclure dans ce cas.

Dépendance entre deux variables aléatoires quantitatives

Représentons graphiquement (nuage de point, scatter plot) le montant des avoirs en fonction de l'âge

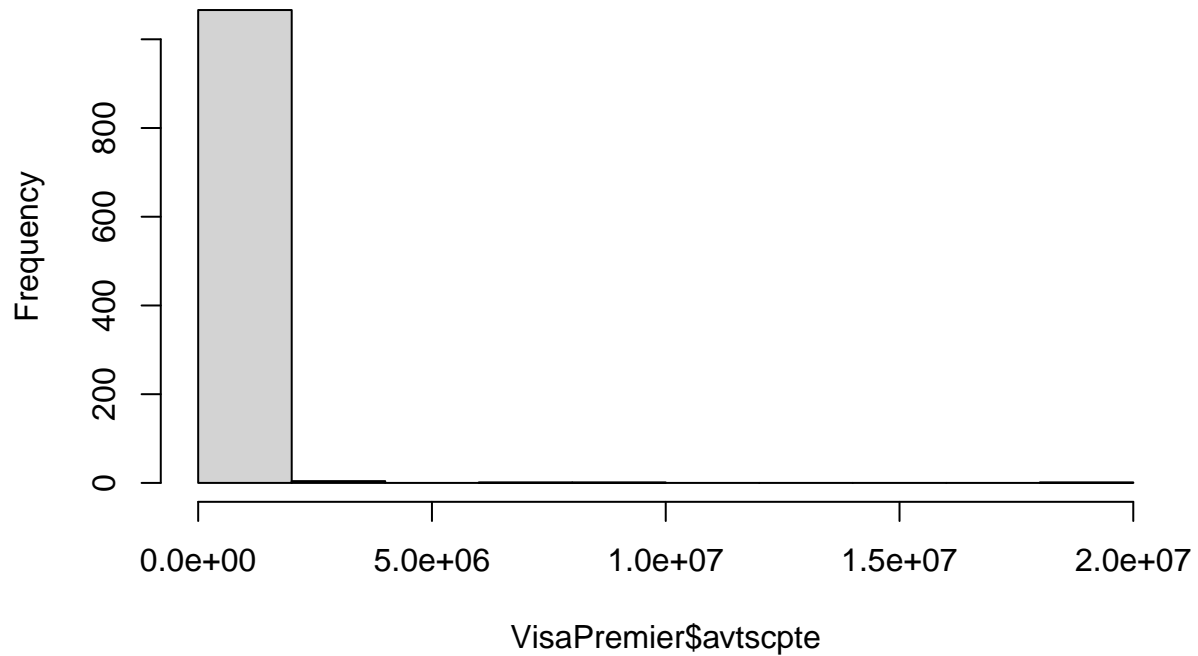
```
plot(avtscpte~age,data=VisaPremier)
```



On ne voit rien, à cause de la distribution du montant des avoirs qui est très asymétrique (très étalé vers les grands valeurs) :

```
hist(VisaPremier$avtscpte)
```

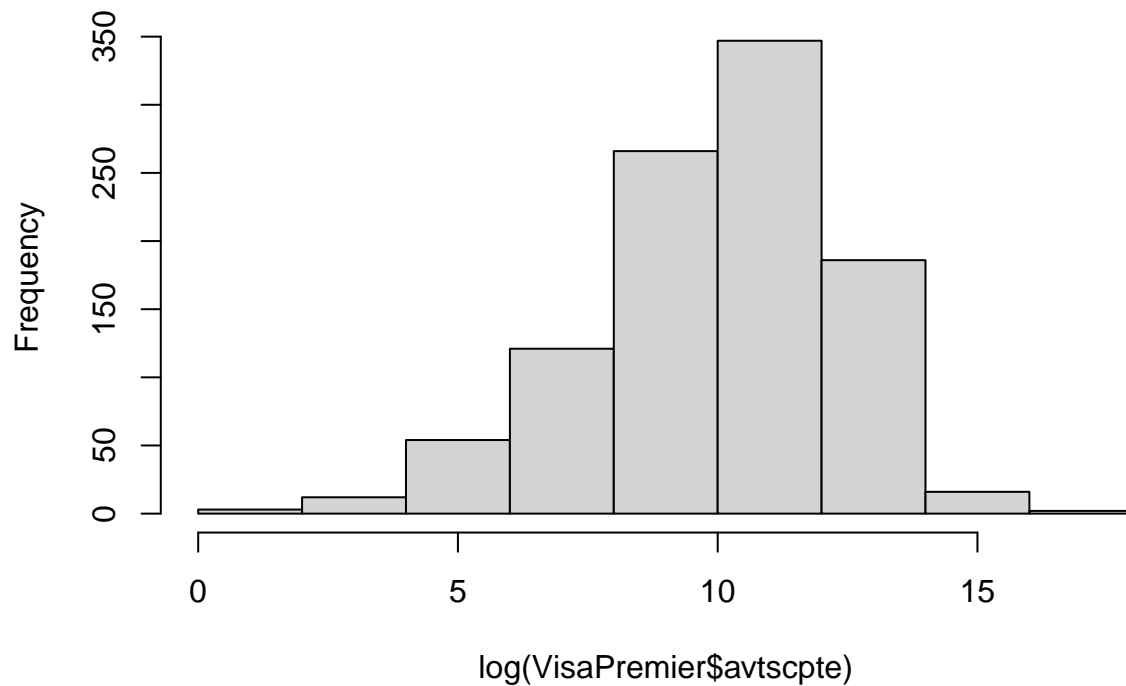
Histogram of VisaPremier\$avtscpte



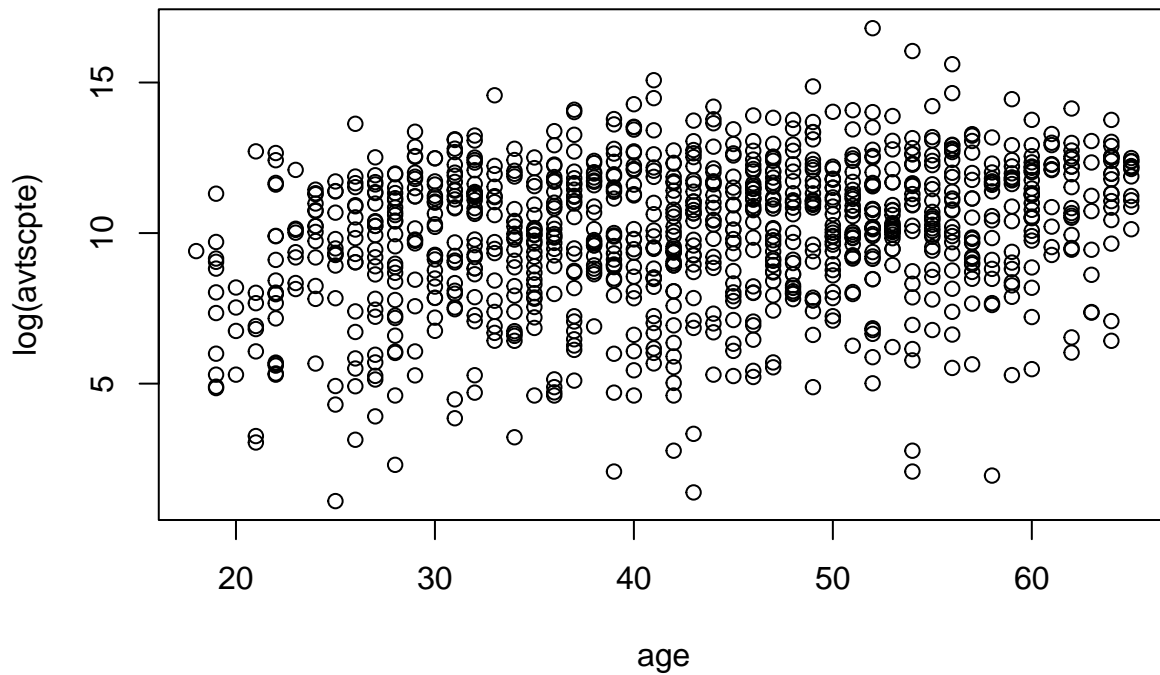
travaillant avec le logarithme des montants, on va gommer cette dispersion vers les grandes valeurs

```
hist(log(VisaPremier$avtscpte))
```

Histogram of log(VisaPremier\$avtscpte)



```
plot(log(avtscpt)~age,data=VisaPremier)
```



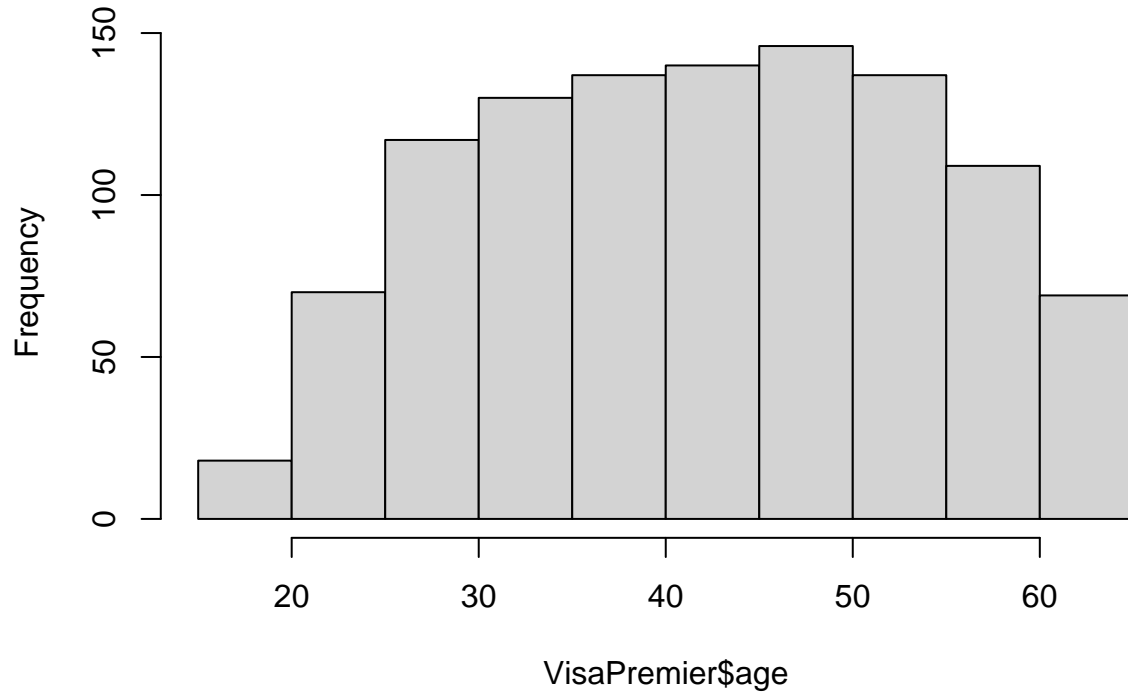
Il semblerait qu'il y ait une légère tendance croissante...

Numériquement, pour quantifier la dépendance entre deux variables quantitatives, on utilise le **coefficient de corrélation linéaire**.

Attention : la liaison linéaire est optimale si les variables sont gaussiennes. Donc avant d'utiliser une corrélation linéaire, on vérifie que les variables sont approximativement gaussiennes, quitte à utiliser des transformations.

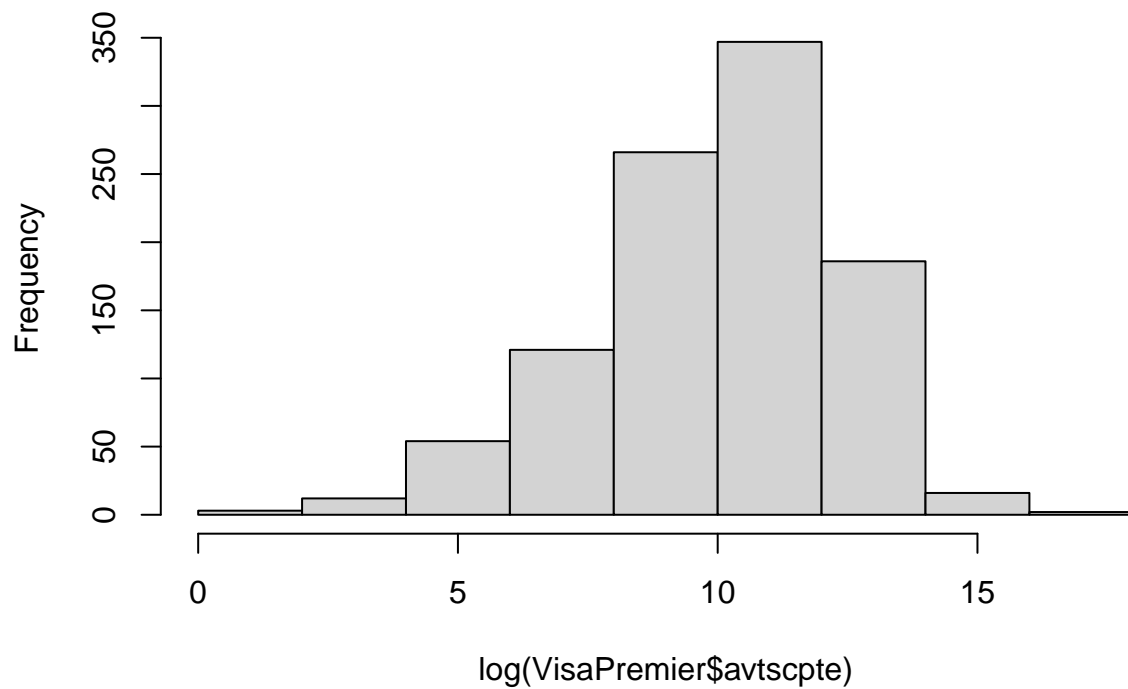
```
hist(VisaPremier$age)
```

Histogram of VisaPremier\$age



```
hist(log(VisaPremier$avtscpte))
```

Histogram of log(VisaPremier\$avtscpte)



On peut calculer le coefficient de corrélation linéaire

```
cor(VisaPremier$age, log(VisaPremier$avtscpte))
```

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

Le soucis vient du fait que certaines personnes ont une variable avtscpte nulle, et la transformation log nous donne une valeur -Inf. Pour éviter ce soucis, on va modifier le data set en ajoutant une petite valeur (typiquement la plus petite non nulle observée divisée par 10) à ceux qui avaient un avoir nul:

```
VisaPremier$avtscpte[VisaPremier$avtscpte==0]=  
min(VisaPremier$avtscpte[VisaPremier$avtscpte>0])/10  
cor(VisaPremier$age,log(VisaPremier$avtscpte))
```

```
## [1] 0.2475014
```

La première que l'on se pose est : est-ce que ce coefficient est significatif, c'est à dire significativement différent de 0. Pour cela, on va faire un **test de corrélation linéaire**, qui va décider entre : H0 : pas de dépendance linéaire $\Leftrightarrow R_{xy} = 0$ H1 : dépendance linéaire $\Leftrightarrow R_{xy} \neq 0$

```
cor.test(VisaPremier$age,log(VisaPremier$avtscpte))
```

```
##  
## Pearson's product-moment correlation  
##  
## data: VisaPremier$age and log(VisaPremier$avtscpte)  
## t = 8.3599, df = 1071, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1904765 0.3028617  
## sample estimates:  
## cor  
## 0.2475014
```

La p-value est significative (<0.05) (et même très significative), on peut conclure à une dépendance linéaire entre l'âge et le log montant des avoirs.

On peut même aller plus loin en vérifiant qu'il est bien significativement positif:

```
cor.test(VisaPremier$age,log(VisaPremier$avtscpte),alternative = "greater")
```

```
##  
## Pearson's product-moment correlation  
##  
## data: VisaPremier$age and log(VisaPremier$avtscpte)  
## t = 8.3599, df = 1071, p-value < 2.2e-16  
## alternative hypothesis: true correlation is greater than 0  
## 95 percent confidence interval:  
## 0.199743 1.000000  
## sample estimates:  
## cor  
## 0.2475014
```

On peut conclure que le log-montant croît linéairement avec l'âge, et donc que le montant des avoirs croît avec l'âge (mais pas linéairement).

Puisqu'il y a une dépendance linéaire significative, on peut estimer un **modèle de régression linéaire** :

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

avec $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

```
modele=lm(log(avtscpte)~age,data=VisaPremier)  
summary(modele)
```

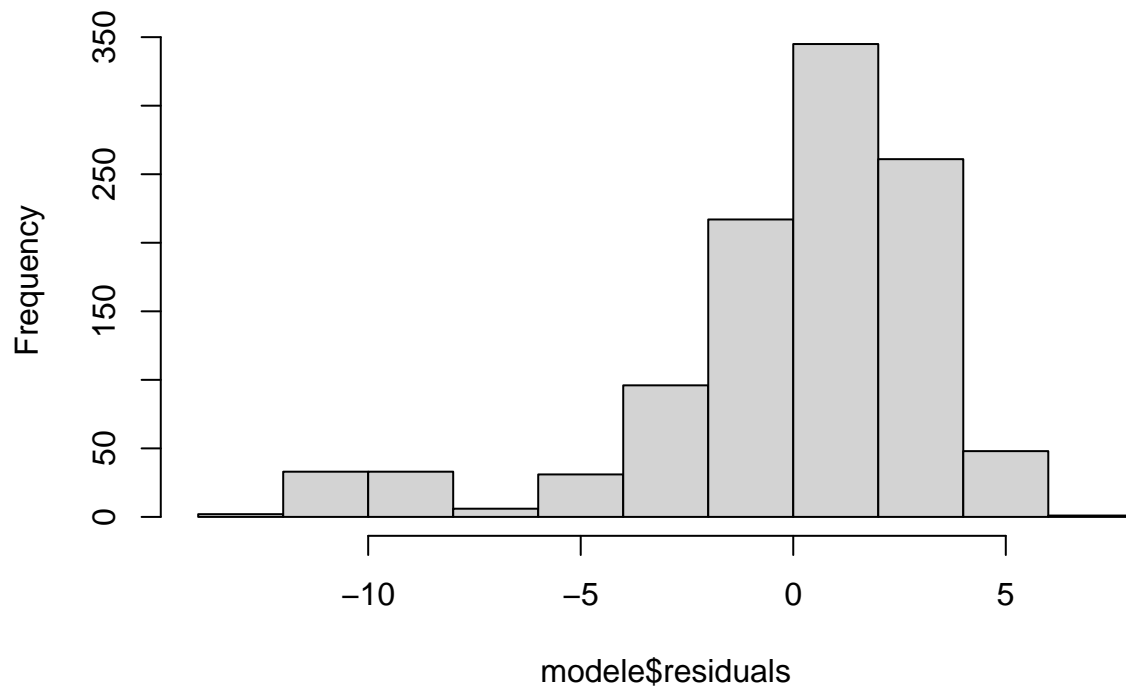
```
##
## Call:
## lm(formula = log(avtscpte) ~ age, data = VisaPremier)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.162  -1.160   0.740   2.203   6.801
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.183791   0.388069   15.94  <2e-16 ***
## age          0.073456   0.008787    8.36  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.425 on 1071 degrees of freedom
## Multiple R-squared:  0.06126,    Adjusted R-squared:  0.06038
## F-statistic: 69.89 on 1 and 1071 DF,  p-value: < 2.2e-16
```

De cette sortie, on peut extraire : - l'estimation de l'intercept β_0 : $\hat{\beta}_0 \simeq 6.18$ - l'estimation du coefficient de régression de l'âge β_1 : $\hat{\beta}_1 \simeq 0.07$. Quand l'âge augmente d'une année, le log-montant augmente en moyenne de 0.07. - on peut lire aussi les Std. Error, qui sont les écarts-types des estimateurs de β_0 et β_1 - on peut aussi lire les pvalues ($\Pr(>|t|)$) des tests de significativités de β_0 et β_1 . Ici les deux paramètres sont bien significativement différent de 0. - le coefficient de détermination R^2 (Multiple R-squared) est un coefficient entre $[0,1]$, qui est d'autant plus proche de 1 que la covariable X explique bien le Y. Précisément, c'est la part de variance de Y que l'on arrive à expliquer à partir du modèle linéaire. - sa version ajustée (Adjusted R-squared), qui est une correction du coefficient R^2 qui prend en compte la complexité du modèle (en nombre de variable.)

Mais avant d'interpréter tout cela, il nous aurait fallu vérifier que les résidus du modèles de régression ϵ_i sont bien gaussiens :

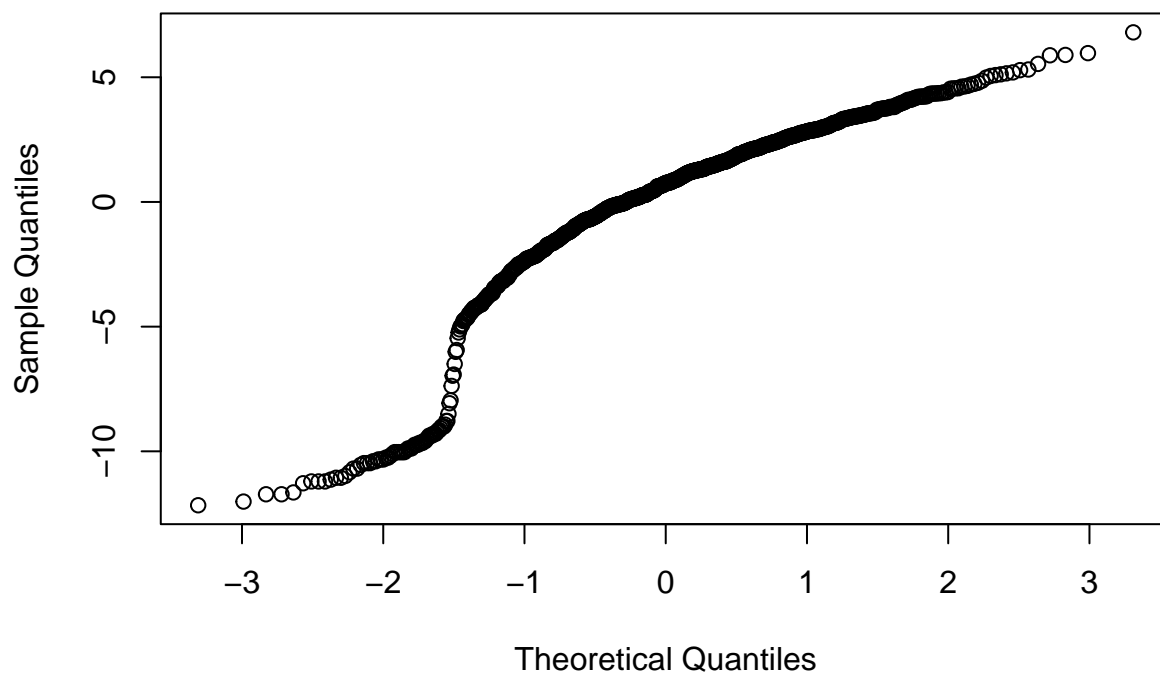
```
hist(modele$residuals)
```

Histogram of modele\$residuals



```
qqnorm(modele$residuals)
```

Normal Q-Q Plot



```
shapiro.test(modele$residuals)
```

```
##  
## Shapiro-Wilk normality test
```



```
##
## data: modele$residuals
## W = 0.86173, p-value < 2.2e-16
```

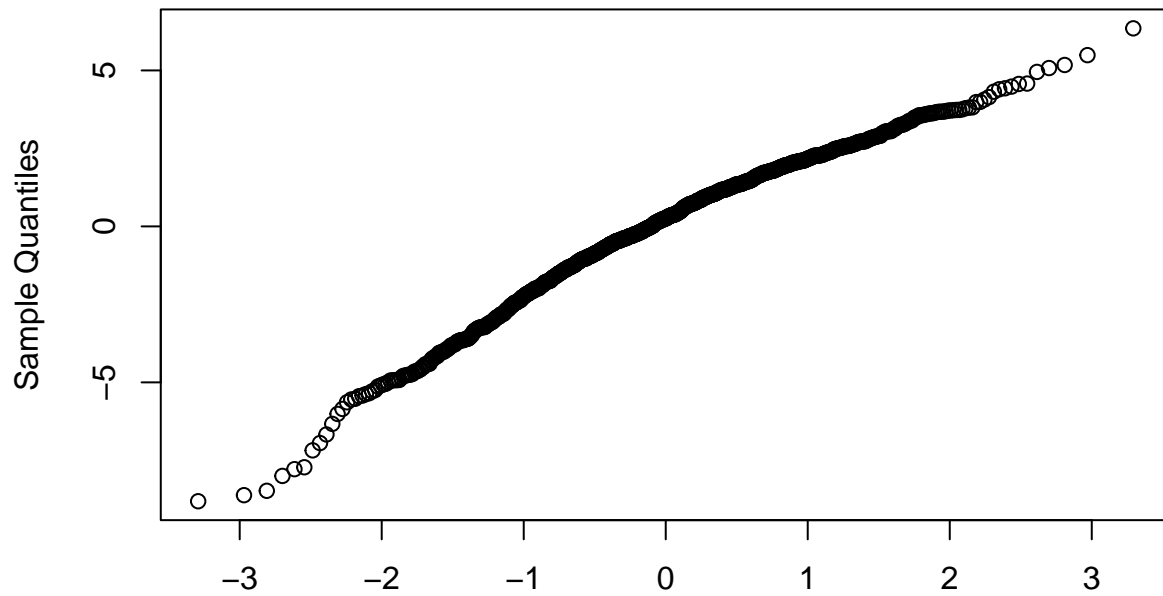
Ici ce n'est pas du tout le cas. Donc on ne peut pas valider ce modèle, et donc ni l'utiliser, ni interpréter les valeurs des coefficients.

En regardant de plus près, le soucis vient du remplacement par 0.3 de toutes les valeurs nulles avant le passage au log. Si on refait l'analyse sans ces individus :

```
VP2=VisaPremier[-which(VisaPremier$avtscpte==0.3),]
modele2=lm(log(avtscpte)~age,data=VP2)
summary(modele2)
```

```
##
## Call:
## lm(formula = log(avtscpte) ~ age, data = VP2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.8106 -1.3050  0.2617  1.6747  6.3491
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.840834   0.270443  28.993 < 2e-16 ***
## age          0.050271   0.006077   8.272 4.16e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.287 on 1005 degrees of freedom
## Multiple R-squared:  0.06374,    Adjusted R-squared:  0.06281
## F-statistic: 68.42 on 1 and 1005 DF,  p-value: 4.16e-16
qqnorm(modele2$residuals)
```

Normal Q-Q Plot



Theoretical Quantiles

Vi-

suellement, cela semble approximativement gaussien, même si le test de Shapiro dit le contraire (car la taille d'échantillon relativement grande lui permet de détecter une déviation même très faible par rapport à l'hypothèse gaussienne.)