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, anciente, 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            anciente         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.   :0.000
##  1st Qu.:      0   1st Qu.: 0.0000   1st Qu.:      0   1st Qu.:0.000
##  Median :      0   Median : 0.0000   Median :      0   Median :1.000
##  Mean   :  52888   Mean   : 0.2395   Mean   :  35915   Mean   :1.473
##  3rd Qu.:      0   3rd Qu.: 0.0000   3rd Qu.:      0   3rd Qu.:2.000
##  Max.   :3472938   Max.   :13.0000   Max.   :5449561   Max.   :9.000
##
##     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
##       nbcb          nbcbptar          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 tord 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

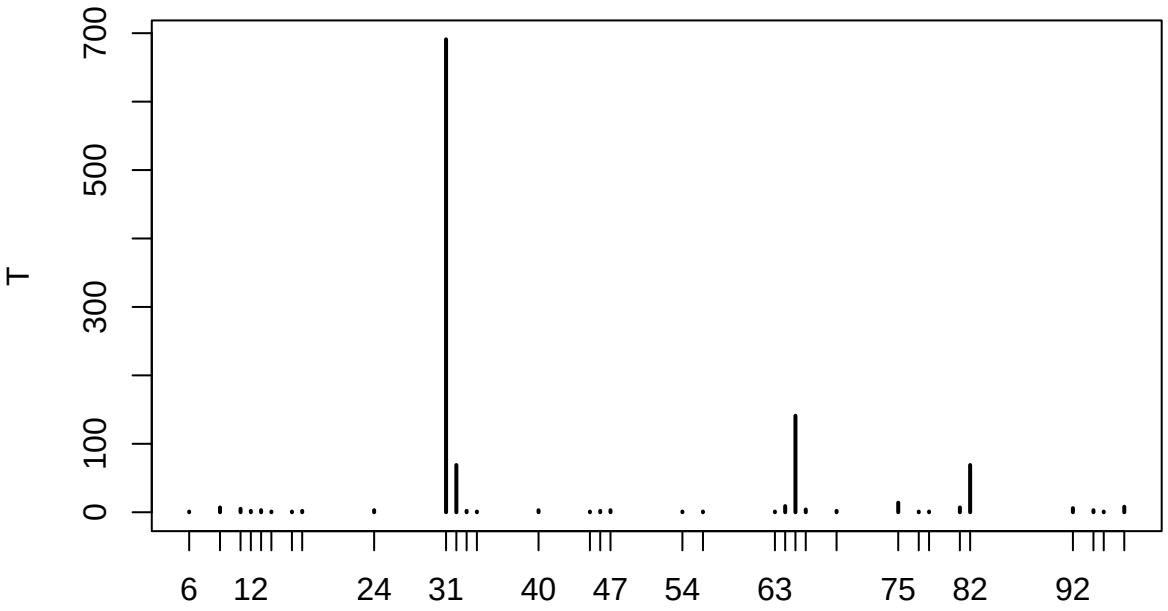
```
VisaPremier$departem = as.factor(VisaPremier$departem)
```

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 codqlt 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 regénérera 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 extrèmes). 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 ptvente sexe    age sitfamil anciente 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>, nblivret <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 supprimer les variables non informative (constantes ou redondantes), on a imputé les données manquantes, corriger les erreurs de codage (déclarer les variables catégorielles comme telles)... on a même mis le doit 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, indépendente 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**

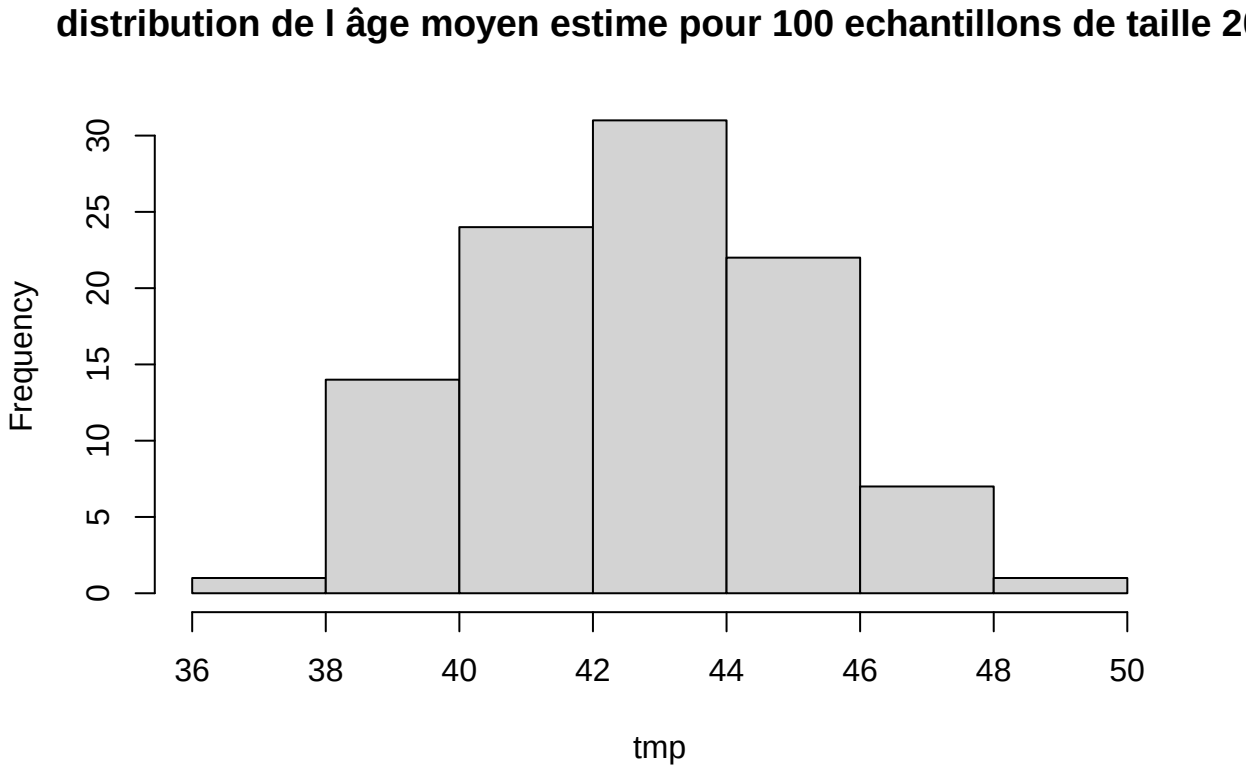
Tirons un sous-échantillon de taille 20 de notre basée de données (pour la variable âge)

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

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

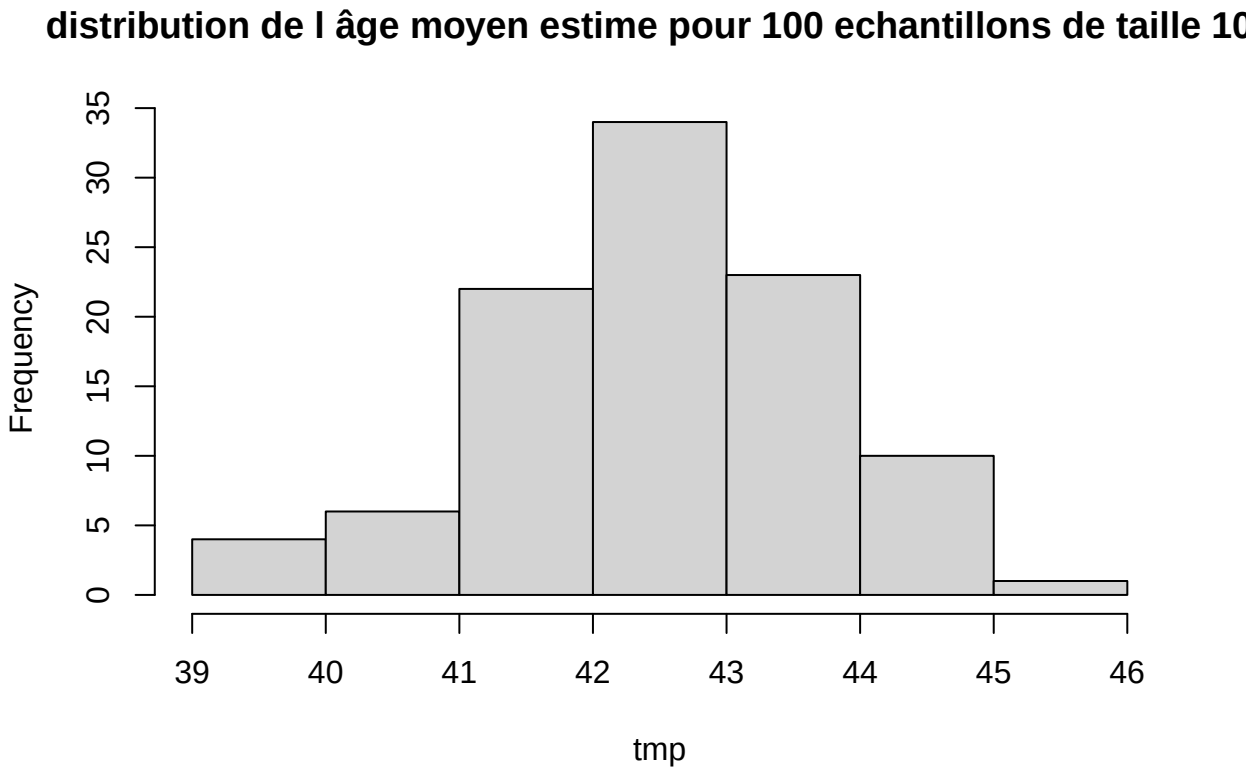
Si on répète cela plusieurs fois, on constate que l'estimation de l'âge de la population à partir des données de l'échantillon varie

```r
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')
```



Faisons de même avec des échantillons de taille 100

```r
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')
```



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'])
```

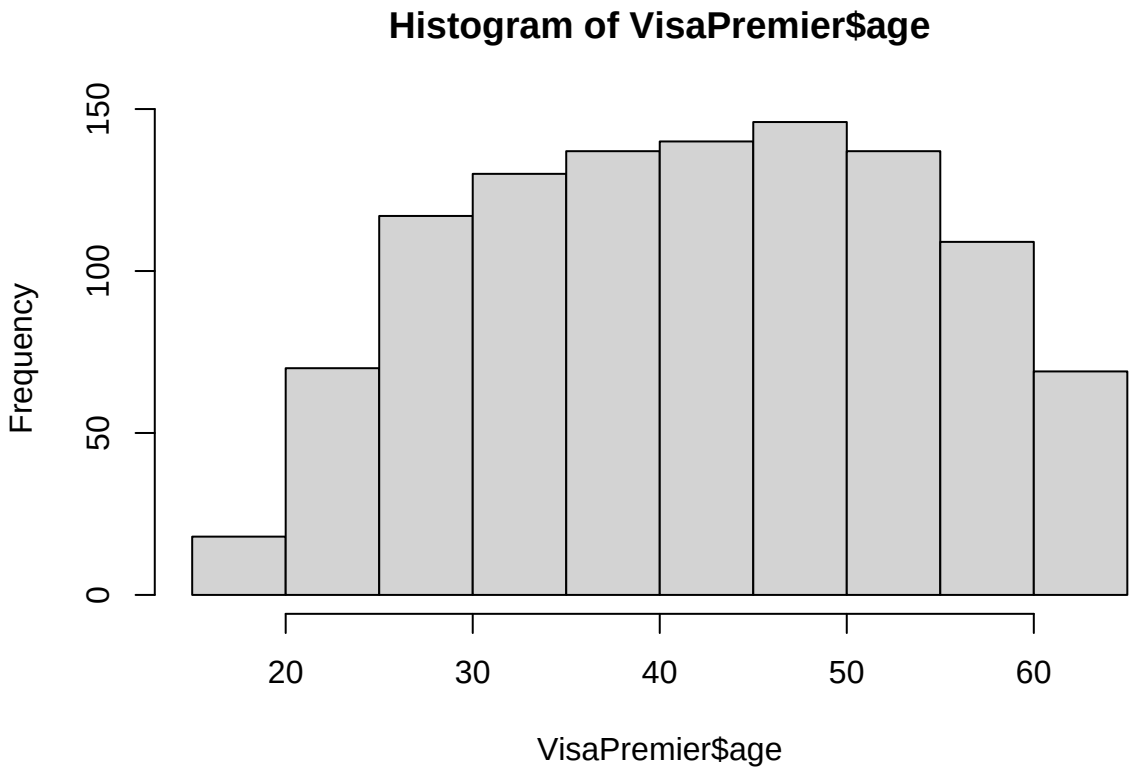
```
## [1] 42.91317
```

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

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

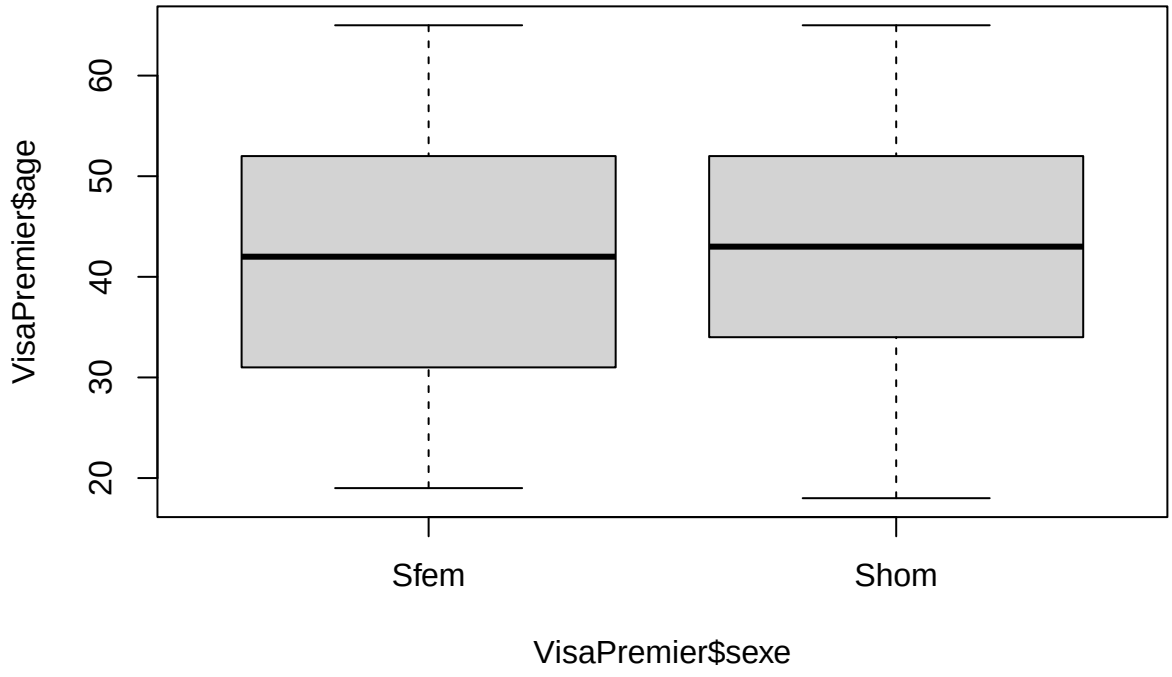
On peut aussi estimer la distribution de probabilité de la variable âge, en utilisant un histogramme, qui est une estimation de la densité de probabilité d'une variable quantitative

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



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

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



Lorsqu'on estime la moyenne théorique à partir de la moyenne de l'échantillon, on réalise un 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 (inconue) 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)

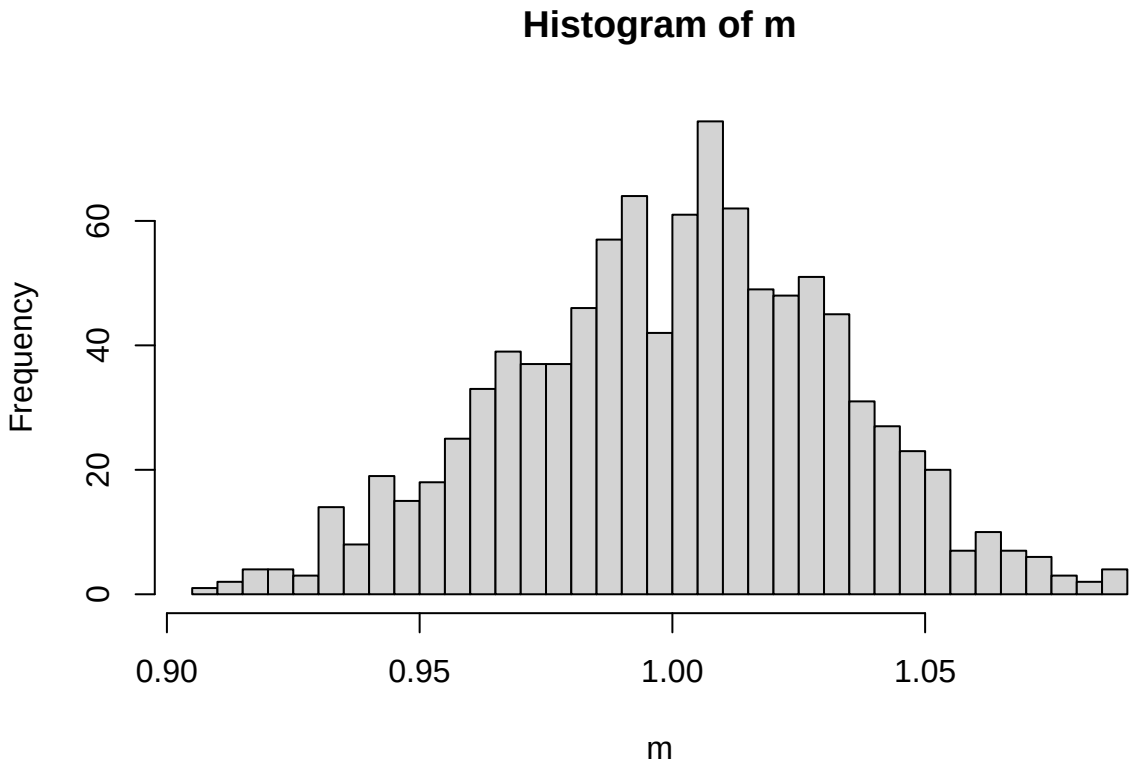
2. Lorsqu'on augmente le niveau de confiance

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

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

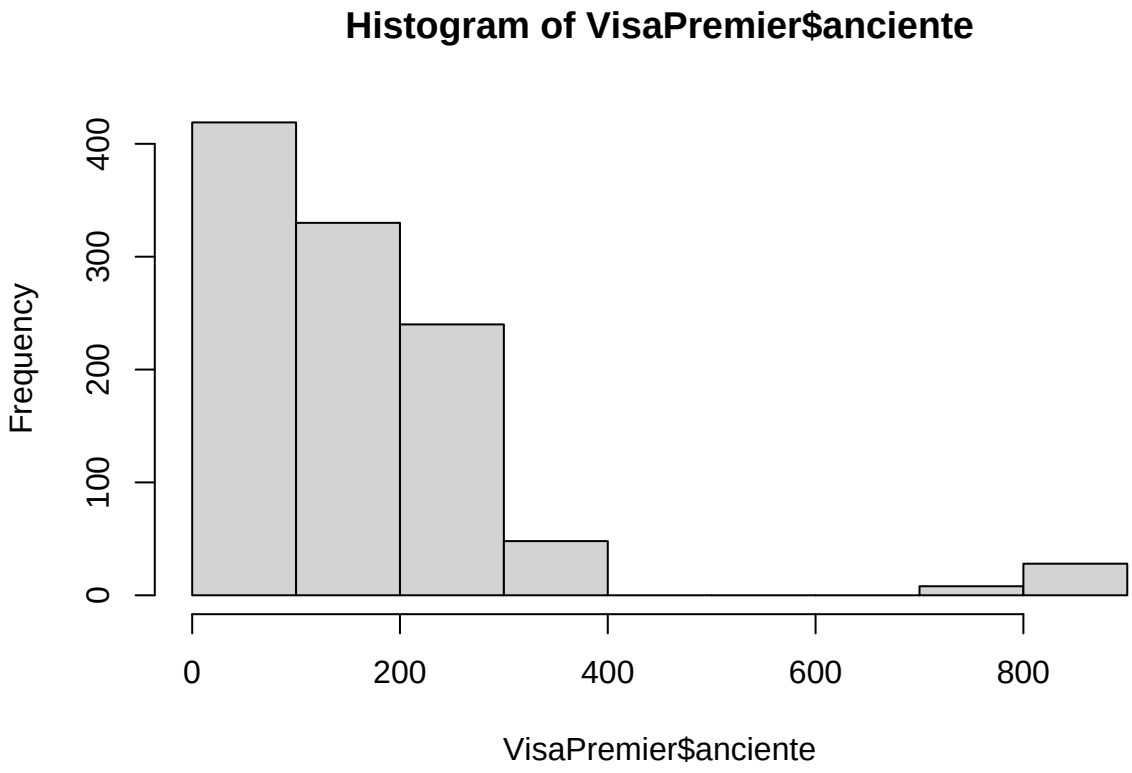
3. Ces intervalles de confiance sont valides dès lors que la taille d'échantillon est suffisamment grande (typiquement au moins 30), et ce quelques soit la nature des données de départ (en particulier quelque soit leur distribution). En effet le théorème central limite nous assure que les moyennes empiriques sont distribuées suivant des lois normales, peut 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)
```



Exercice sur l'ancienneté

```
hist(VisaPremier$anciente)
```



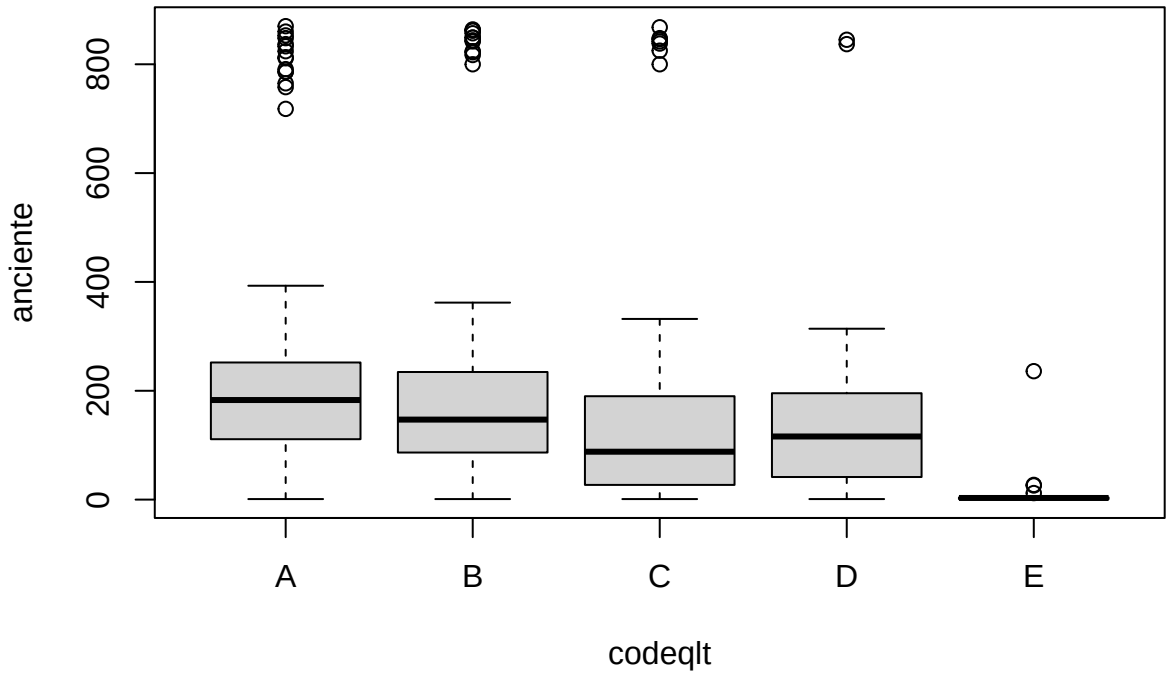
```
mean(VisaPremier$anciente)
```

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

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

```
## [1] 147.7689 166.4660
## attr(,"conf.level")
## [1] 0.95
```

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




```
cat('IC ancienneté pour le codqlt A :',t.test(VisaPremier$anciente[VisaPremier$codeqlt=="A"],conf.level=
```

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

```
cat('IC ancienneté pour le codqlt B :',t.test(VisaPremier$anciente[VisaPremier$codeqlt=="B"],conf.level=
```

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

```
cat('IC ancienneté pour le codqlt C :',t.test(VisaPremier$anciente[VisaPremier$codeqlt=="C"],conf.level=
```

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

```
cat('IC ancienneté pour le codqlt D :',t.test(VisaPremier$anciente[VisaPremier$codeqlt=="D"],conf.level=
```

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

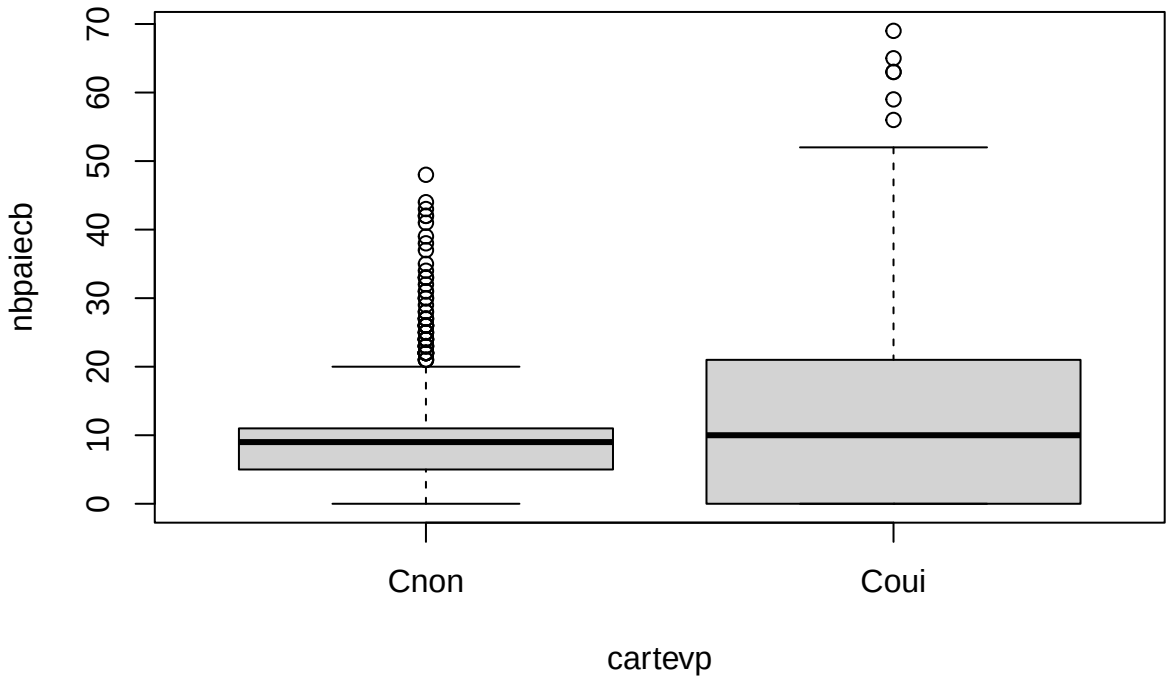
```
cat('IC ancienneté pour le codqlt E :',t.test(VisaPremier$anciente[VisaPremier$codeqlt=="E"],conf.level=
```

```
## 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 ?

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




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

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

```r
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>=30)

```r
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 rejettant l'hypothèse H0 au profit de H1. 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 rejetter H0 quand cette pvalue est $<$ à 0.05 (mais attention, ça dépend des domaines d'applications...)

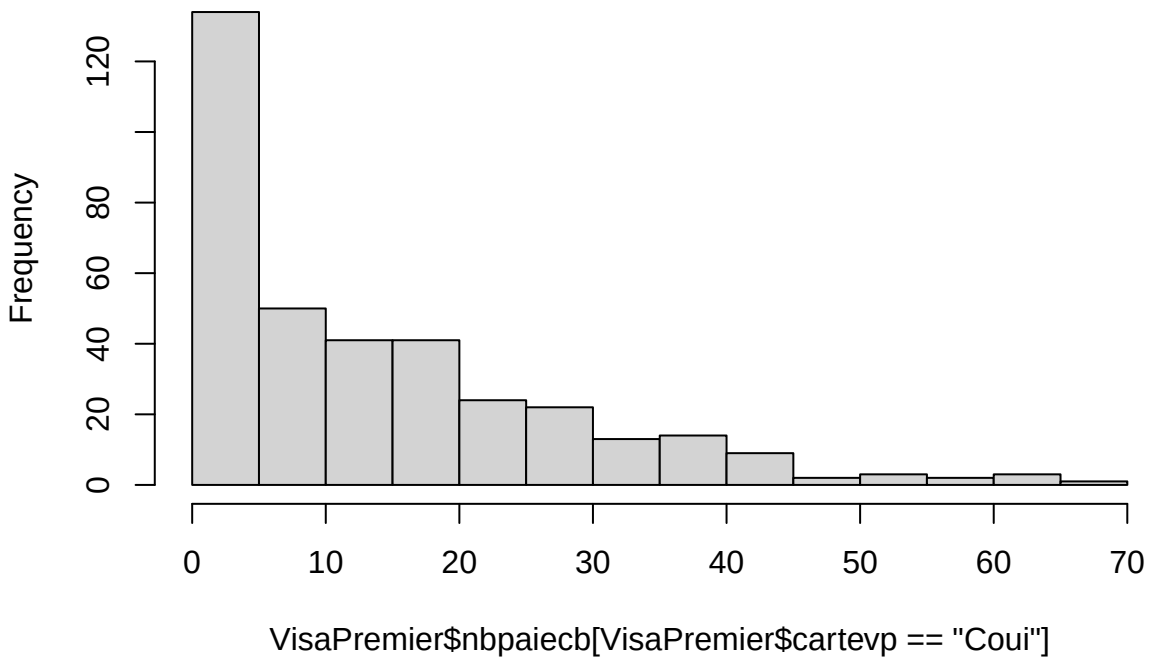
Remarque : - on a fait un test bilatéral (H1 : mu1 <> mu2), mais en pratique, on est plutôt intéressé par des tests unilatéraux : ici, on veut vérifier si ce que l'on voit sur les moyennes (moyenne de l'échantillon avec CB plus grand que l'autre) est significatif (et non du au hasard). Le test unilatéral va consister à tester : H0 (hypothèse nulle) : moyenne (cartvp=non) = moyenne (cartvp=oui) H1 (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 H0 : l'échantillon suit une loi normale H1 : l'échantillon ne suit pas une loi normale

```
hist(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 tord que l'échantillon est gaussien (il est trop peu puissant, il accepte trop souvent H0 à tord)

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 rejetter H0, i.e. à conclure à une différence significative), donc on utilisera lorsque cela est possible de préférence le test de Student.