TP Statistiques 1

Charantonis Anastase & Brunel Nicolas & Julien Floquet

19 fevrier 2018

Rappel: on considere que vous avez suivi l'introduction ? R: http://tryr.codeschool.com/ dans son integralit? (d'ou le QCM dans 10 minutes)

- Utilitaires et informations importantes:
  nettoyer son espace de travail: rm(list=ls())
  nettoyer sa console: Ctrl +L
  retrouver le r?pertoire de travail: getwd()
  changer d'emplacement de travail: setwd()
- Biblio pratique:
  http://www3.jouy.inra.fr/miaj/public/formation/initiationRv10.pdf
- On travaillera sous Rmarkdown. Utilisez RStudio, et cr?ez un fichier RMarkdown. Pour les rapports on attendera un pdf et un RMarkdown avec le m?me nom, du type `TPSTAT1_NOM_PRENOM_GROUPE_NUMERO.format` ou les formats sont Rmd et pdf. Le depot p?dagogique sera ouvert à partir de la semaine prochaine. Nous allons ?valuer, pour chaque ?tudiant, 2/5 des rendus de TP choisis al?atoirement.

G?n?rer des donn?es et les enregistrer

Dans cette partie on va apprendre ? g?n?rer des echantillons issus d'une loi de probabilit?s.

Un echantillon d'une loi de probabilit? est une suite de r?alisations de cette loi. Il est tr?s utile en statistique de pouvoir g?n?rer des variables al?atoires selon diverses lois de probabilit?.

R peut le faire pour un grand nombre de lois via les fonctions de la forme r*func*(n,p1,p2,...) o? *func* indique la loi de probabilit?, n est le nombre de variables ? g?n?rer et p1, p2, ... sont les param?tres de la loi. Pour ce faire on aura besoin de utiliser `help()` pour les fonctions suivantes:

Lois	Nom sous R
Gaussienne	rnorm(n,mean=0,std=1)
Uniforme	runif(n,min=0,max=1)
Poisson	rpois(n,lambda)
Exponentielle	rexp(n,rate=1)
χ^2	rchisq(n,df)
Binomiale	rbinom(n,size,prob)
Cauchy	rcauchy(n,location=0,scale=1)

Retrouvez ces fonctions dans vos notes de probabilit?s (ou sur internet), ?a va vous ?tre utile.

Pour chaque une de ces fonctions g?n?rer une echantillon de 40 donn?es i.i.d. (independantes et identiquement distribu?es), associez les ? un vecteur inclus dans un `data.frame`, puis utilisez les fonctions `write.csv` et `write.table` pour les enregistrer. Il serrait intelligent de noter les param?tres utilis?s (moyenne,std,...) dans le nom de votre variable/fichier enregistr?.

```
x1<-rnorm(40,mean=0,sd=1)
x2<-runif(40,min=0,max=1)
x3<-rpois(40,lambda=3)
```

```r
x4<-rexp(40,rate=1)
x5<-rchisq(40,df=2)
x6<-rbinom(40,size = 36,prob=0.8)
x7<-rcauchy(40,location=0,scale=1)
data<-data.frame(N=x1,U=x2,P=x3,E=x4,C=x5,B=x6,CAU=x7)
write.csv(data,file="data_frame.csv")
write.table(data,file="data_table.txt")
```

Charger des donn?es depuis un fichier txt (texte) et csv (comma separated variables)

Nettoyez votre espace de travail. Utilisez les fonctions `read.csv` et `read.table`, pour charger la distribution Gaussienne que vous avez g?n?r?. Que remarquez-vous?

Pensez à utiliser header=TRUE.

```r
rm(list = ls())


read.csv("data_frame.csv",header = TRUE)["N"]
```

```
##               N
## 1  -0.818701488
## 2   0.107009993
## 3  -0.928718425
## 4   0.794804004
## 5   0.615834158
## 6   1.746541991
## 7   1.043091379
## 8   0.702229673
## 9   0.029174510
## 10  1.796860909
## 11 -1.361827294
## 12  0.029696818
## 13 -0.593333470
## 14 -0.009648664
## 15 -0.251196458
## 16  2.185299291
## 17  0.249877538
## 18  0.023567343
## 19 -0.709990852
## 20  0.214840043
## 21 -0.558765179
## 22  0.357301647
## 23  1.223106756
## 24 -1.889748761
## 25 -0.021114639
## 26 -1.466023938
## 27  0.360196003
## 28  2.132370147
## 29 -2.394630329
## 30  0.306222571
## 31 -0.850054727
## 32 -0.396388181
## 33  0.783335882
```

```
## 34 -1.080315088
## 35 -1.602684685
## 36  0.297821048
## 37  0.552008591
## 38  0.014491838
## 39 -0.304710797
## 40  0.356885916
```

```r
read.table("data_table.txt",header = TRUE)["N"]
```

```
##               N
## 1  -0.818701488
## 2   0.107009993
## 3  -0.928718425
## 4   0.794804004
## 5   0.615834158
## 6   1.746541991
## 7   1.043091379
## 8   0.702229673
## 9   0.029174510
## 10  1.796860909
## 11 -1.361827294
## 12  0.029696818
## 13 -0.593333470
## 14 -0.009648664
## 15 -0.251196458
## 16  2.185299291
## 17  0.249877538
## 18  0.023567343
## 19 -0.709990852
## 20  0.214840043
## 21 -0.558765179
## 22  0.357301647
## 23  1.223106756
## 24 -1.889748761
## 25 -0.021114639
## 26 -1.466023938
## 27  0.360196003
## 28  2.132370147
## 29 -2.394630329
## 30  0.306222571
## 31 -0.850054727
## 32 -0.396388181
## 33  0.783335882
## 34 -1.080315088
## 35 -1.602684685
## 36  0.297821048
## 37  0.552008591
## 38  0.014491838
## 39 -0.304710797
## 40  0.356885916
```

```r
#fichier de csv n'a pas le permier colone des index
```

On remarque un décalage par rapport aux colonnes entre le fichier .txt et celui .csv

Tracer les donn?es

G?n?rez un vecteur qui contient 10 r?alisations de la loi normale N(0,1). Tracez les points obtenus en utilisant 'plot', et m?ttant sur l'axe des x un vecteur sequentiel de la taille de votre vecteur.

Que remarquez-vous? (Utiliez la commande 'abline(h=0)')

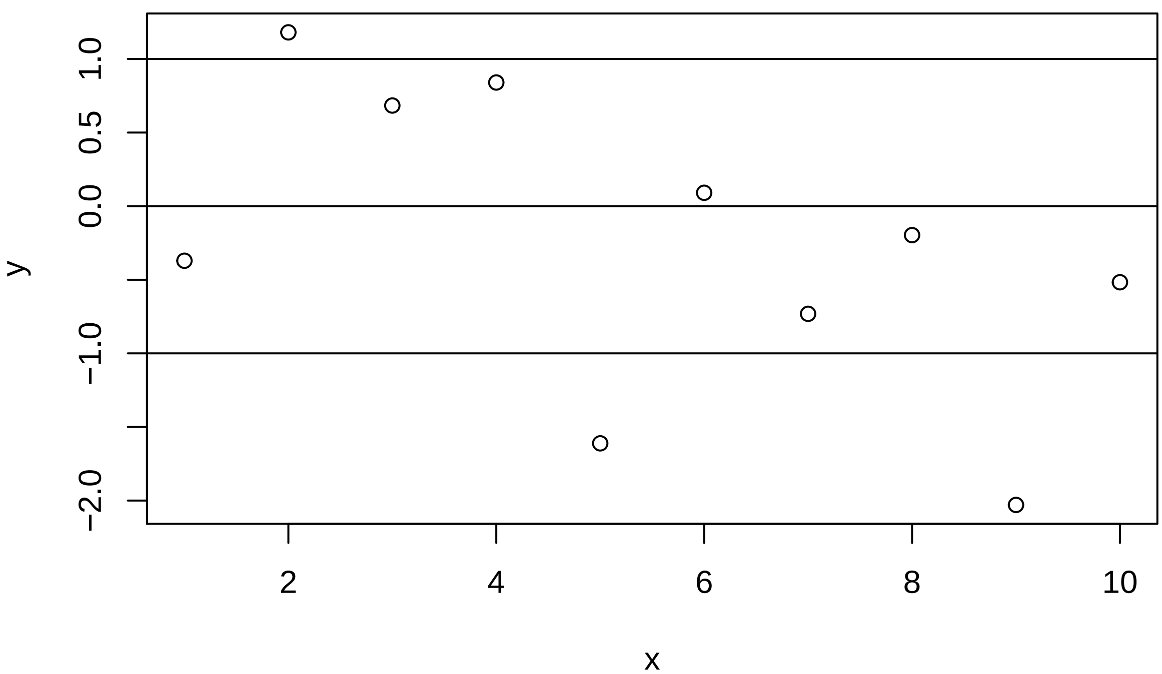
On remarque que les valeurs se situent autour de 0

Tracez ?galement les lignes horizontales 1 et -1. Que remarquez-vous? Combien de points sont en dehors de ces lignes? La m?me chose avec les lignes horizontales 2 et -2, 3 et -3. Que remarquez vous?

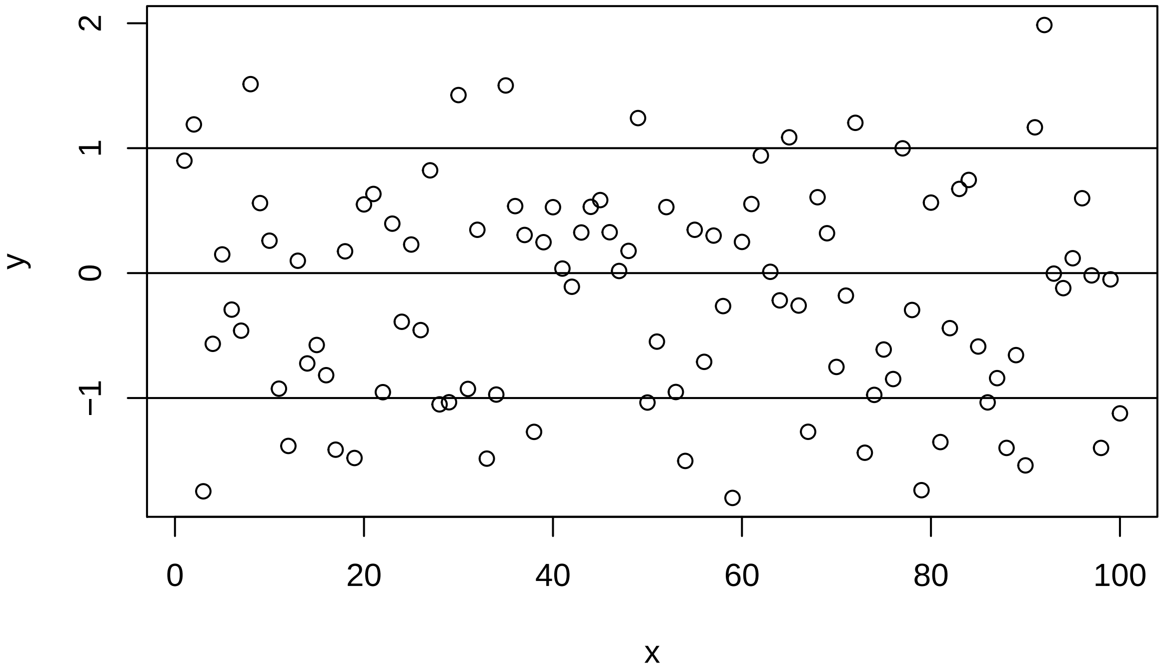
On remarque qu'il y a presque autant de valeurs supérieres à 1 que de valeurs inférieures à -1, de même pour 2 et -2 et 3 et -3.

Effectuer la m?me chose avec des vecteurs contenant 100 et 1000 valeurs. Que remarquez vous?

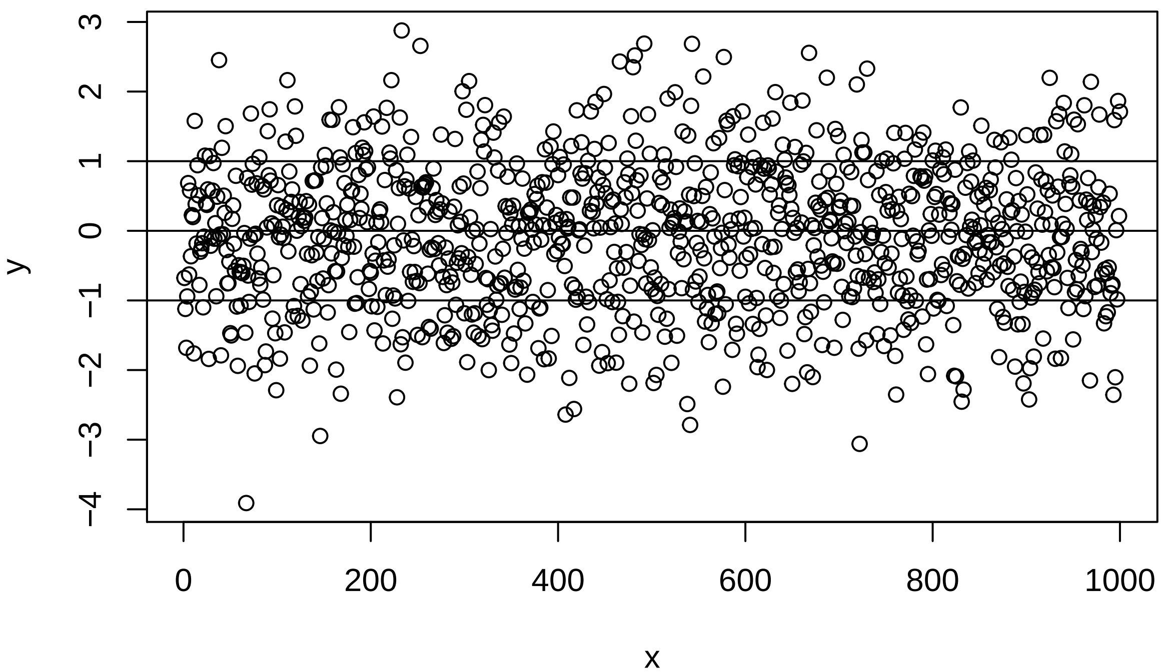
```r
#le function pour generer les vecteurs et les realisations de la loi normale
create_norme <- function(n) {
x<-1:n
y<-rnorm(n,0,1)
plot(x,y)
abline(h=0)
abline(h=1)
abline(h=-1)
}

create_norme(10)
```



```r
create_norme(100)
```




```r
create_norme(1000)
```



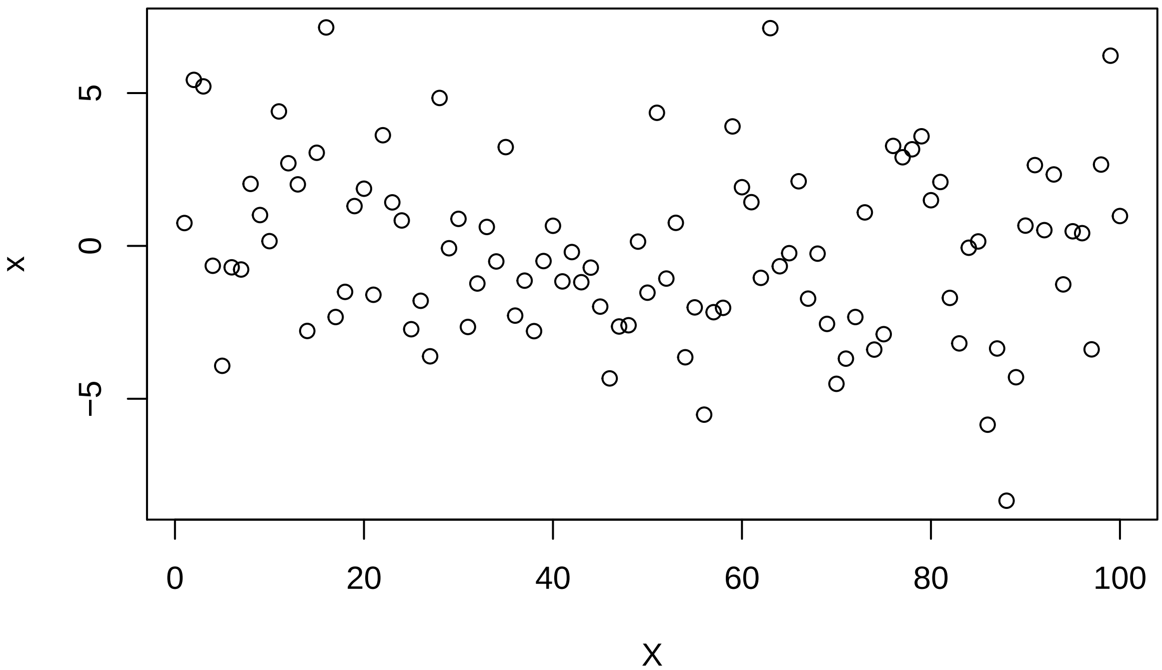

```r
#j'ai esseye ablin = 2 ou 3 , la mojority des points sont pendant [-1,1]ou[-2,2],[-3,3]
```

On remarque que l'on a des résultats similaire quelque soit le nombre de valeurs

Chargez le fichier 'distribution_inconue_1_100_realisations.csv' que vous pouvez trouver dans le m?me emplacement que ce fichier.

Est-ce que vous pouvez conclure quelque chose sur cette distribution, ? partir d'une visualisation?

```r
read.csv("distribution_inconue_1_100_realisations.csv",header = TRUE)->data
plot(data)
```




```r
#pour calcule , on peux calcul la moyenne  , et la variance
moyenne=sum(data[2])/100
m=(data[2]-moyenne)^2
variance=sqrt(sum(m)/100)
```

Testez avec d'autres distributions. Que remarquez-vous?

```r
#tester 200 ,cest mienx qu'avant
   y<-rnorm(200,0,1)
moyenne=sum(y)/200
m=(y-moyenne)^2
variance=sqrt(sum(m)/200)
```

avec les autre distributions suivants , On remarque qu'il semble plus probable que la distribution semble à priori suivre une loi uniforme dans [0, 100] ### Histogrammes

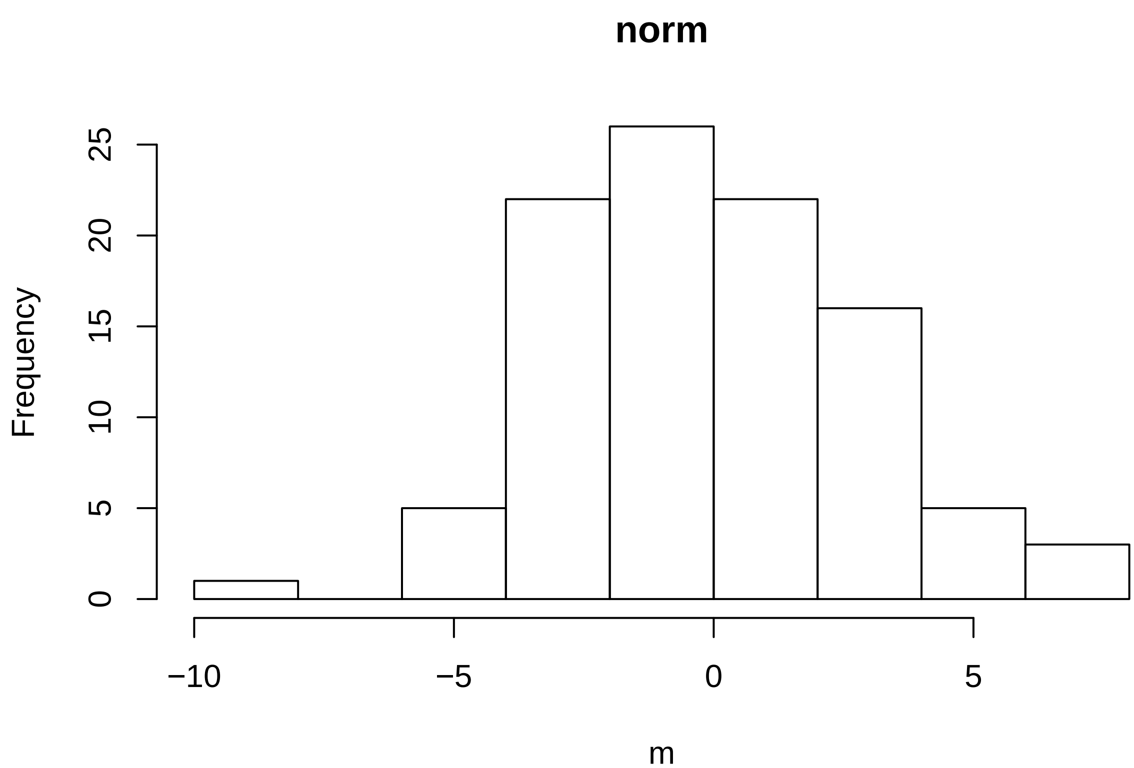
La visualisation des r?sultats precedents nous donnent certaines informations sur la distribution dont ils sont issus.

Les histogrammes sont une autre fa?on d'?valuer visuellement les donn?es d'un echantillon. Ils representent la densit? de distribution de valeurs de r?alisations de notre echantillon par segements.

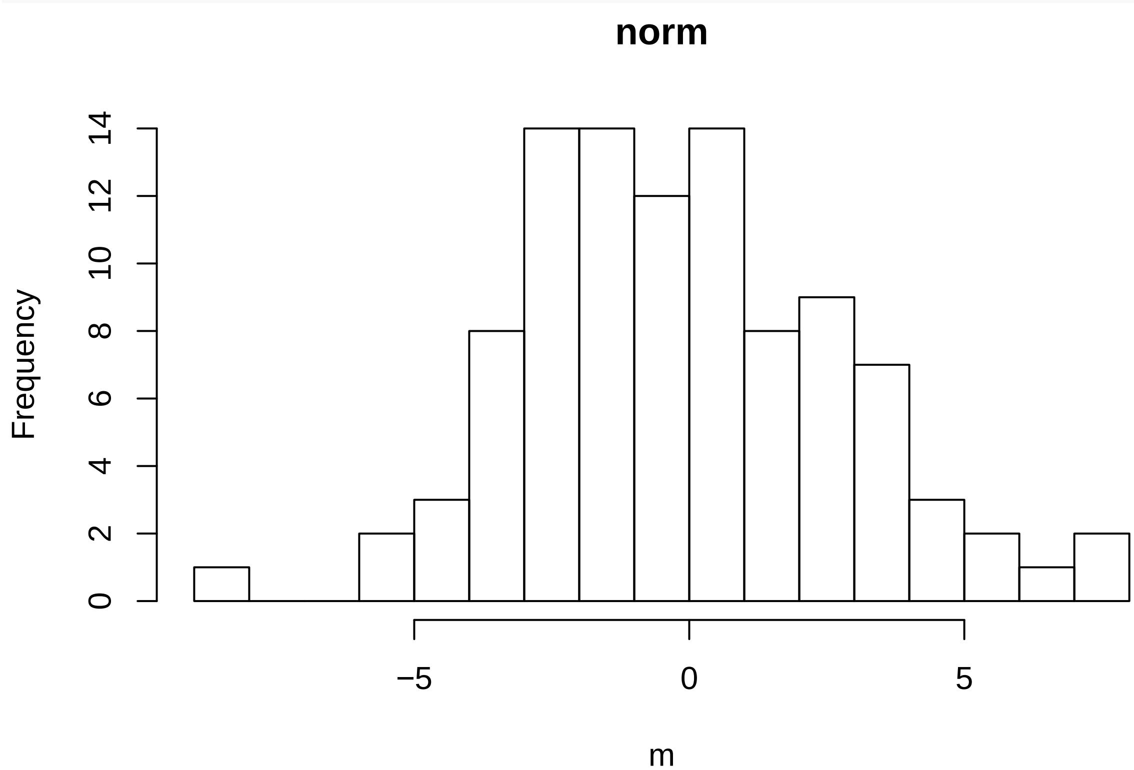
Utilisez `help()` pour la fonction `hist()`.

Appliquez la fonction pour l'echantillon de 100 r?alisations que vous avez cr??, et pour 'distribution_inconue_1_100_realisations.csv'. Que remarquez vous? Testez les differents paramètrages de la fonction: breaks et freq.

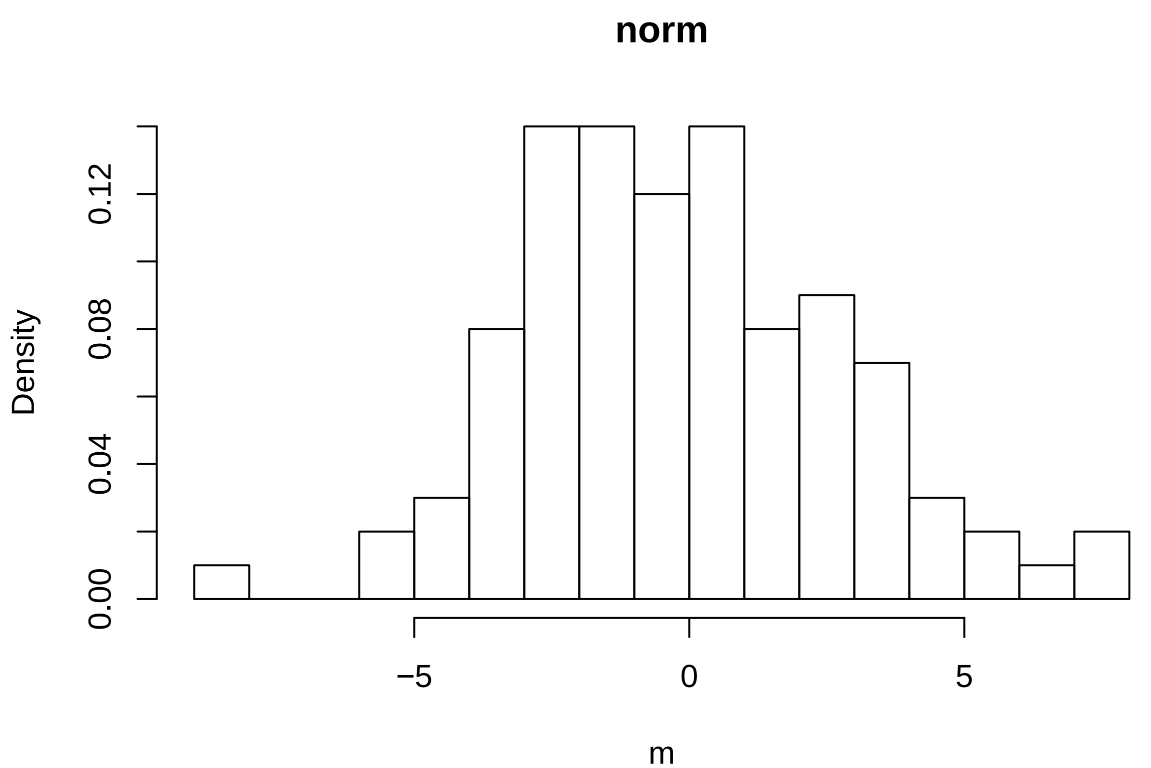
```r
m<-data[,2]
hist(m,main="norm")
```




```r
hist(m,breaks = 12,freq = TRUE,main="norm")
```



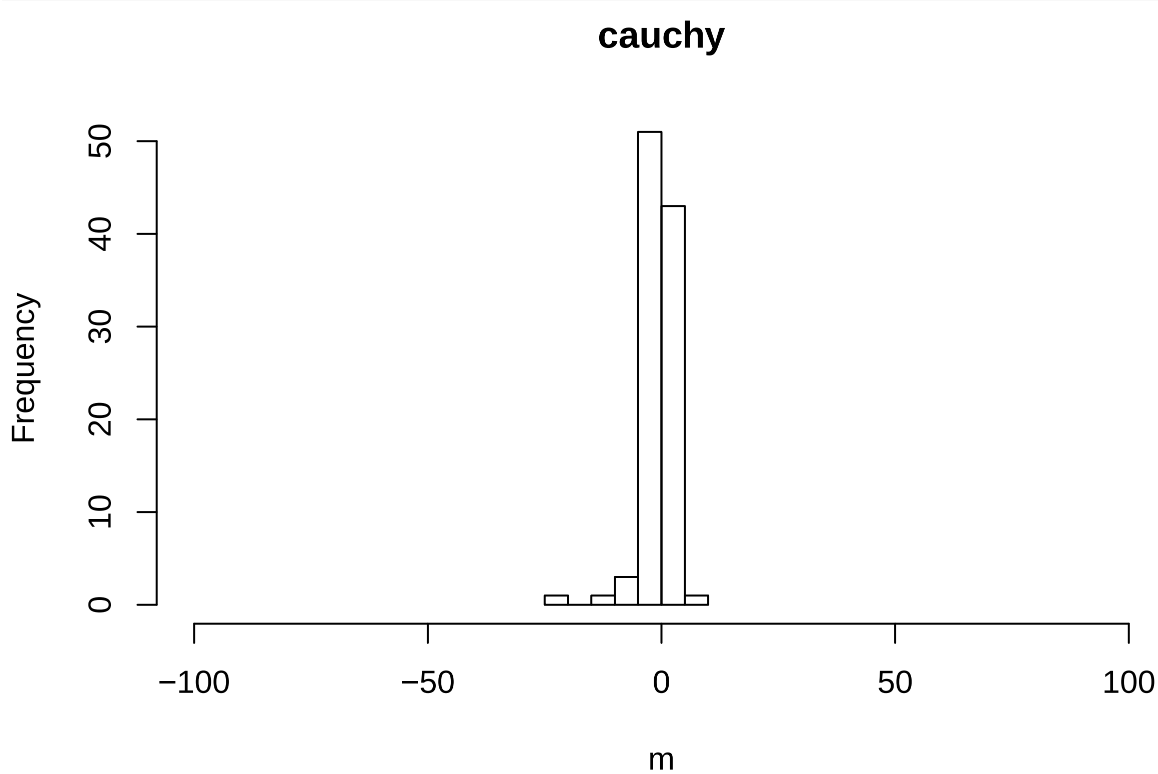

```r
hist(m,breaks = 12,freq = FALSE,main="norm")
```



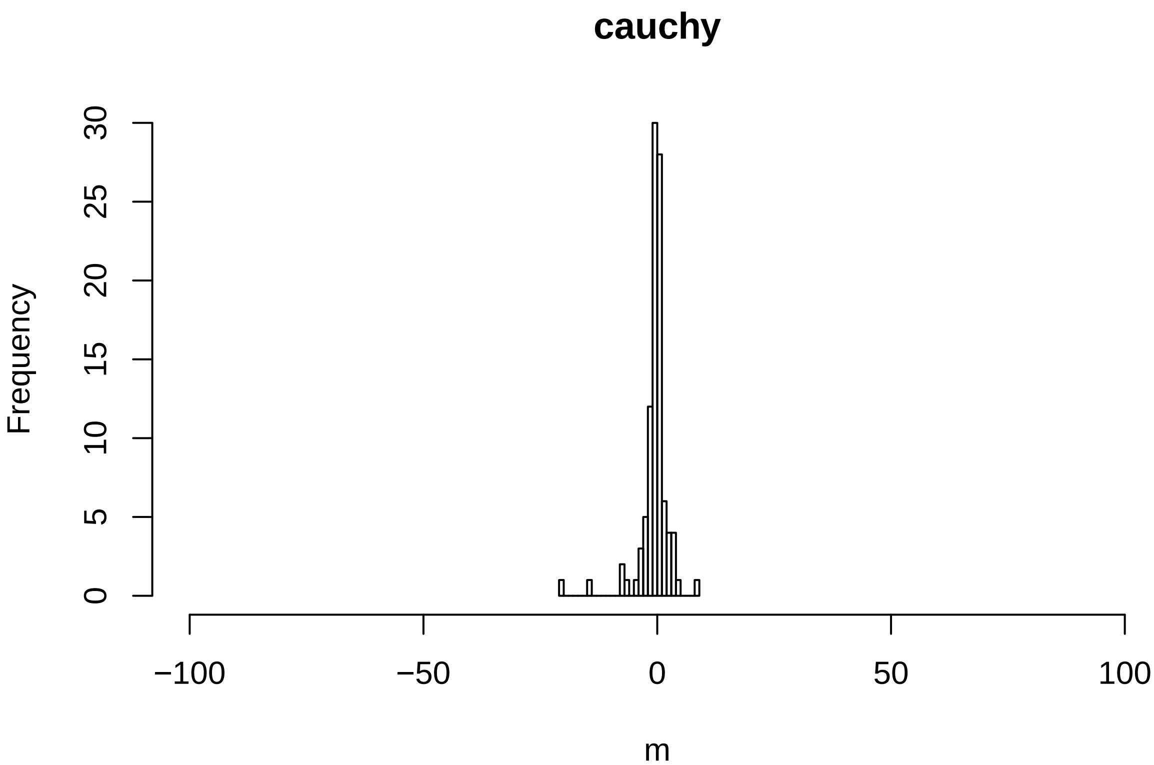
On remarque l'histogramme ressemble un peu à une loi normale centrée réduite.

Effectuez la même chose pour des distributions de Cauchy avec des parametrages differents.

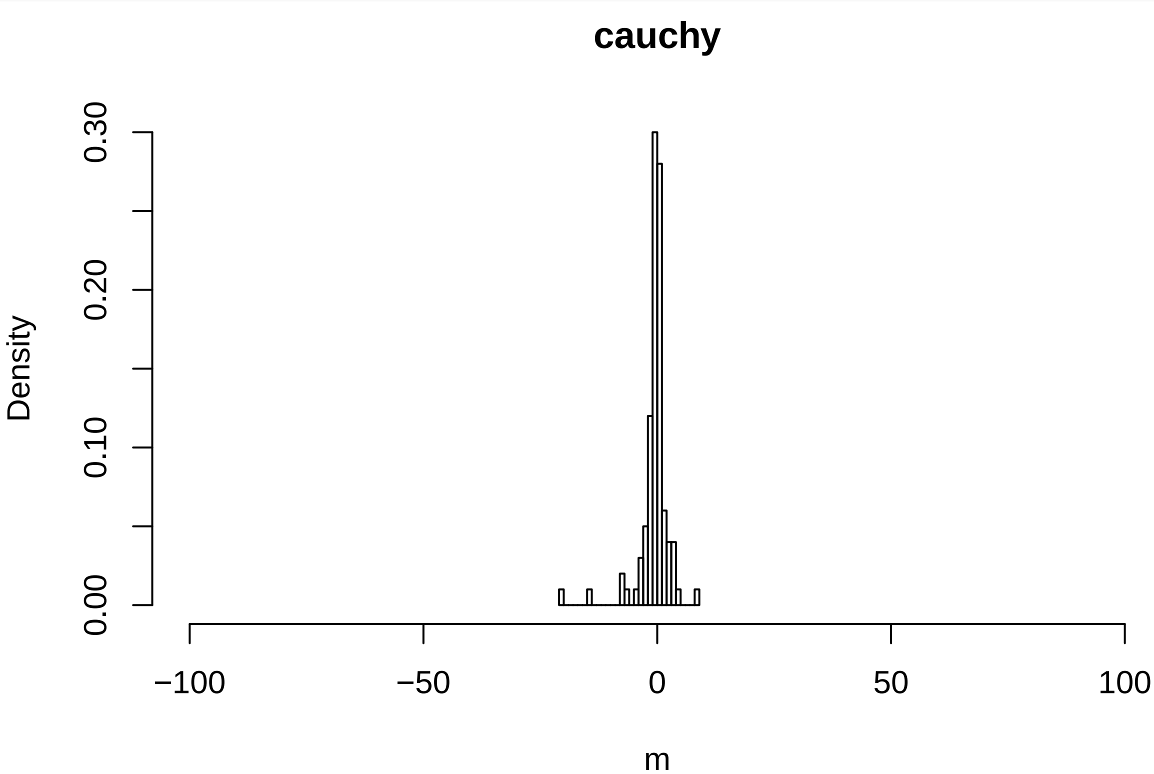
```
m<-rcauchy(100,0,1)
hist(m,main="cauchy",xlim=c(-100,100))
```



```r
hist(m,breaks = 25,freq = TRUE,main="cauchy",xlim=c(-100,100))
```

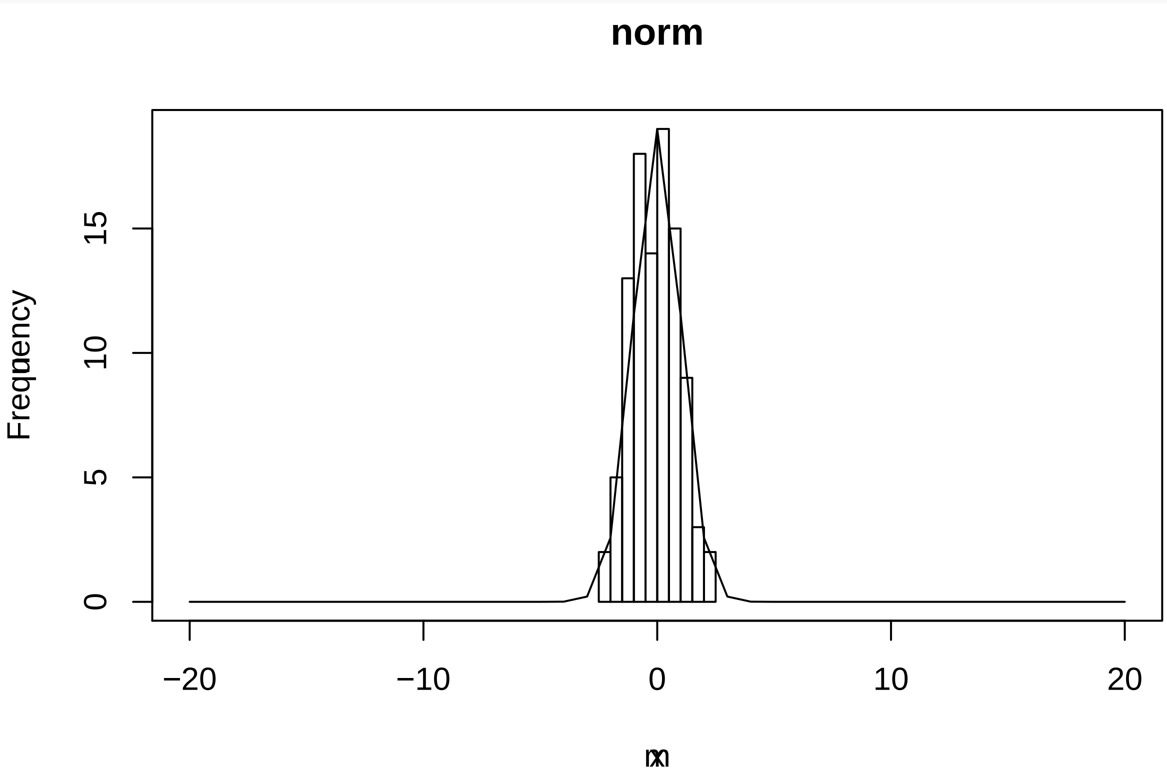



```r
hist(m,breaks = 25,freq = FALSE,main="cauchy",xlim=c(-100,100))
```

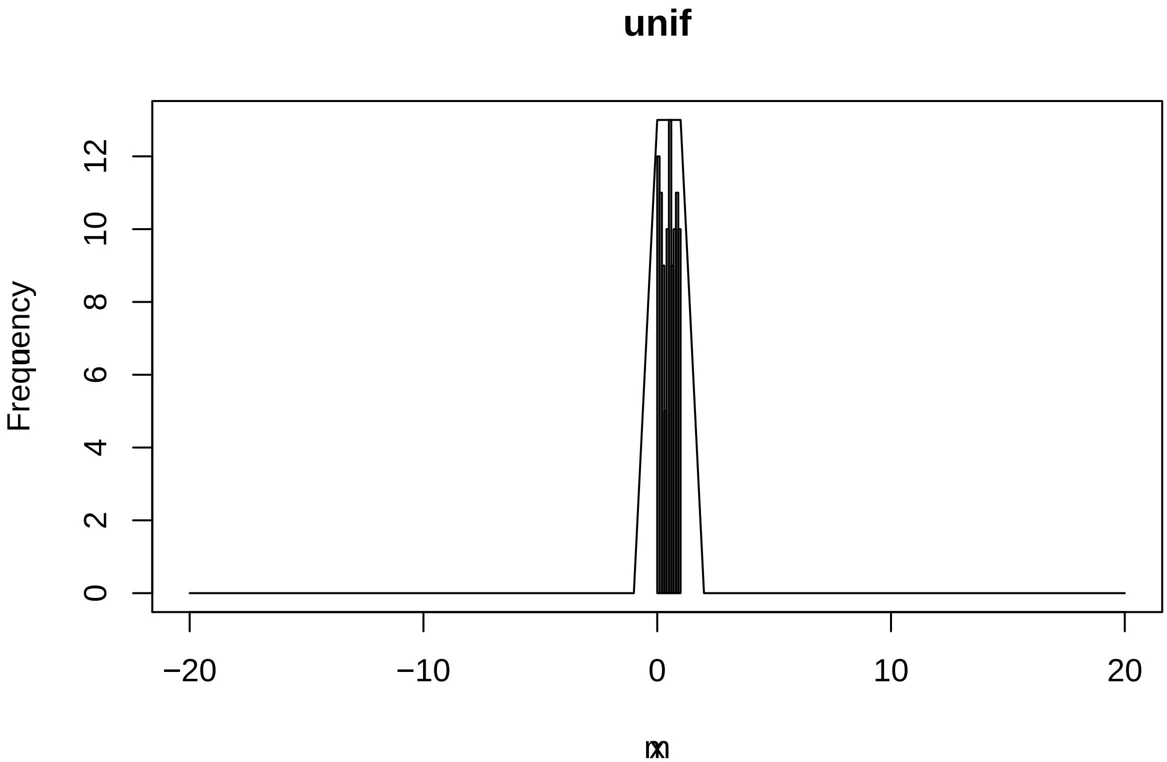


Par ailleurs, regardez les fonctions de type d*func*(n,p1,p2,. . .). Elles peuvent vous donner la distribution théorique que vous devriez obtenir. Superposez deux plots en utilisant `par(new=TRUE)` puis en plottant la distribution correspondante au histogramme que vous visualisez.

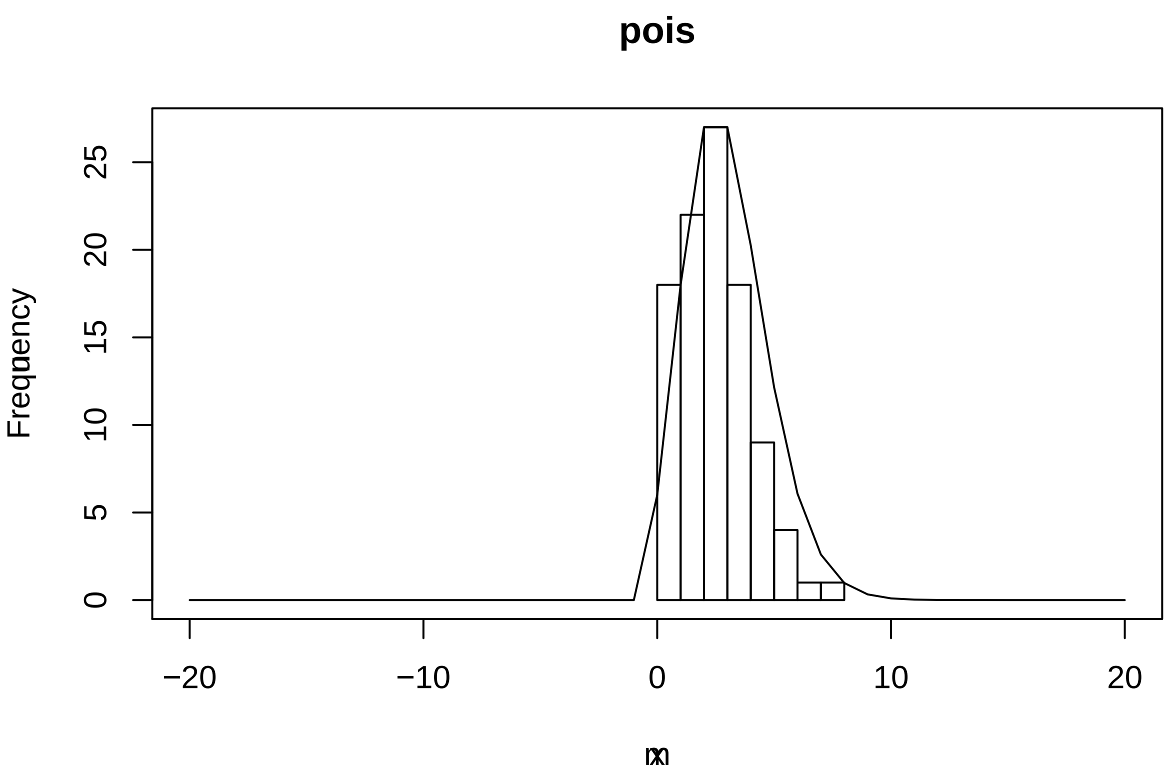
```
x=seq(-20,20)
x1<-rnorm(100,mean=0,sd=1)
x2<-runif(100,min=0,max=1)
x3<-rpois(100,lambda=3)
x4<-rexp(100,rate=1)
x5<-rchisq(100,df=2)
x6<-rbinom(100,size = 36,prob=0.8)
x7<-rcauchy(100,location=0,scale=1)
y1<-dnorm(x,mean=0,sd=1)
y2<-dunif(x,min=0,max=1)
y3<-dpois(x,lambda=3)
y4<-dexp(x,rate=1)
y5<-dchisq(x,df=2)
y6<-dbinom(x,size = 36,prob=0.8)
y7<-dcauchy(x,location=0,scale=1)
#cette function cest pour montrer le hist
dfunc <-function (m,n,s){
x=seq(-20,20)
hist(m,main=s,xlim=c(-20,20))

par(new=TRUE)
plot(x,n,type="l",xaxt="n",yaxt="n")
}

dfunc(x1,y1,"norm")
```



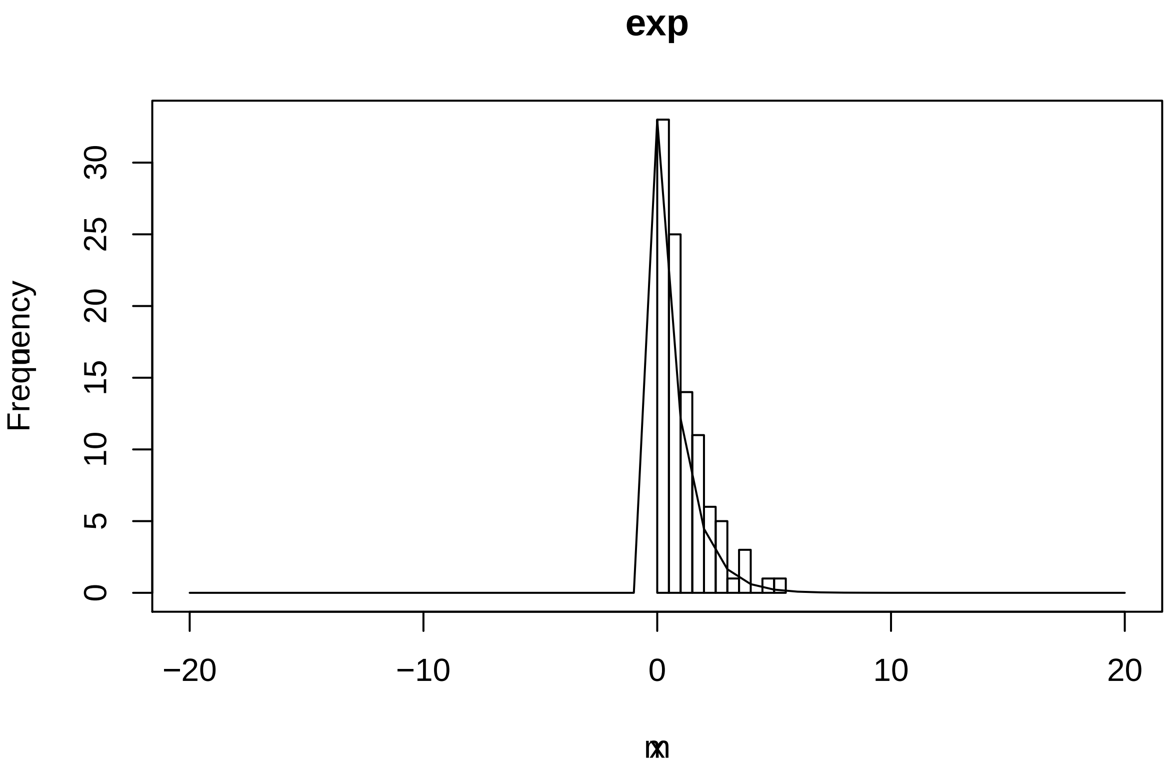
```r
dfunc(x2,y2,"unif")
```



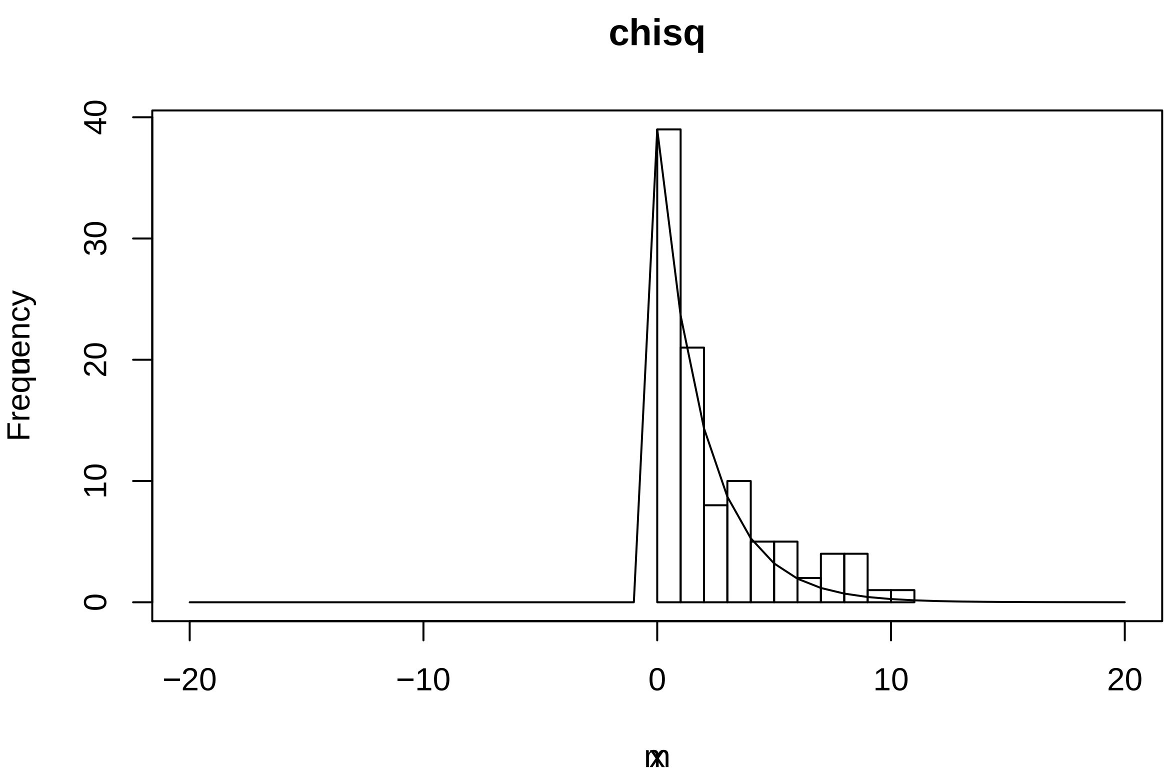

```r
dfunc(x3,y3,"pois")
```



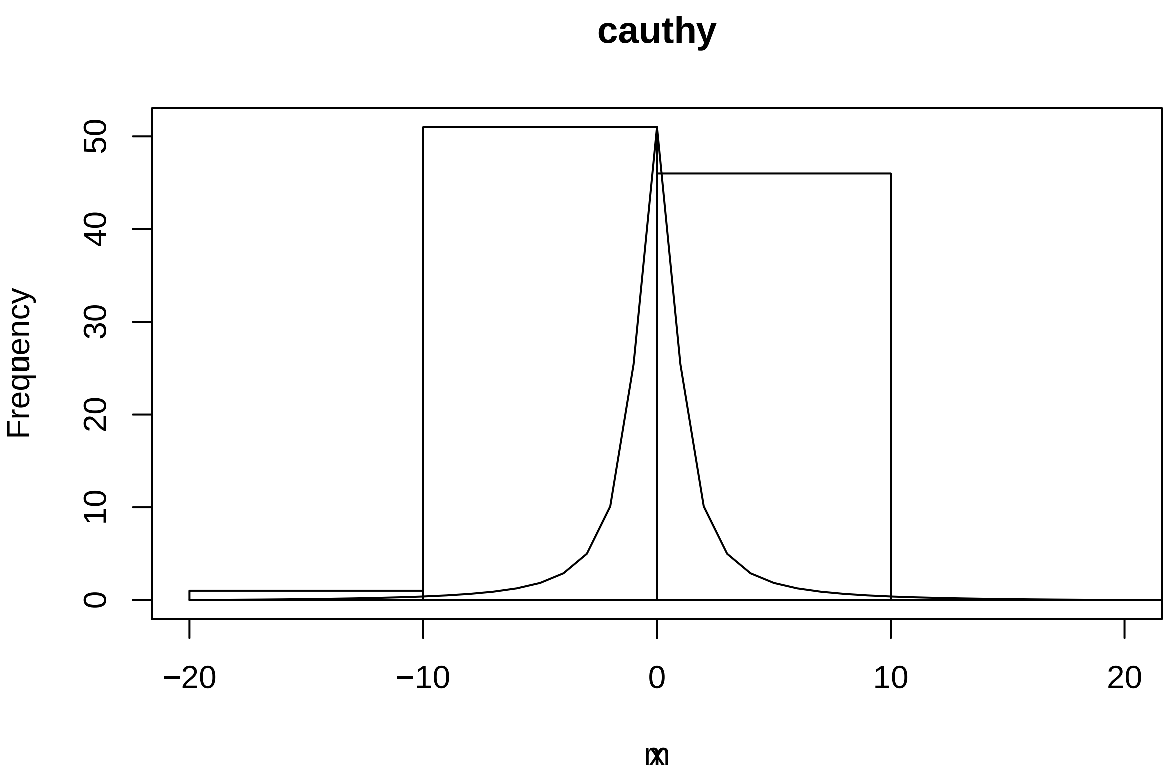
```r
dfunc(x4,y4,"exp")
```




```r
dfunc(x5,y5,"chisq")
```




```r
dfunc(x7,y7,"cauthy")
```




```
rm(list = ls())
```

Moments d'ordre

Les moments d'ordre ?l?v? pour une distribution nous donnent des informations li?s ? la forme des ?cart ? la moyenne. Si on connait notre loi analytiquement, on peut calculer ses moments. Mais quand on a seulement un ?chantillon i.i.d. d'une loi inconnue, nous devons les estimer.

- Empiriquement:
  Skewness n?gatif —> plus notre densit? est dissym?trique vers la gauche.
  Kurtosis petit —> Plus l'extremit? de la densit? va tendre rapidement vers 0.

Sous R il existe les fonctions `skewness()` et `kurtosis()`. Calculez les moments des 4 premiers ordres pours les ?chantillons que vous avez g?n?r? et stockez les r?sultats dans une matrice. Commentez les resultats obtenus et comparez les valeurs th?oriques de ces distributions.\

```
library(fBasics)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
x1<-rnorm(400,mean=0,sd=1)
x2<-runif(400,min=0,max=1)
x3<-rpois(400,lambda=3)
x4<-rexp(400,rate=1)
x5<-rchisq(400,df=2)
x6<-rbinom(400,size = 36,prob=0.8)
x7<-rcauchy(400,location=0,scale=1)

#une fonction pour montrer les 4 resultats:moyenne ,variance,skewness,kurtosis
```

```r
calcul <- function(x) {
moyenne=sum(x)/100
m=(x-moyenne)^2
variance=sqrt(sum(m)/100)
matrix <- matrix(c(moyenne,variance,skewness(x),kurtosis(x)), nrow = 4)
}

y1=calcul(x1)
y2=calcul(x2)
y3=calcul(x3)
y4=calcul(x4)
y5=calcul(x4)
y6=calcul(x4)
y7=calcul(x4)
data<-data.frame(N=y1,U=y2,P=y3,E=y4,C=y5,B=y6,CAU=y7)

write.table(data,file="Analyse_table.txt")
read.table("Analyse_table.txt",header = TRUE)
```

```
##              N           U          P        E        C        B      CAU
## 1  0.459269800  2.01684218 12.0900000 3.876404 3.876404 3.876404 3.876404
## 2  2.095597473  3.07995877 18.4284074 6.117008 6.117008 6.117008 6.117008
## 3 -0.069581043 -0.03461074  0.5152128 2.030869 2.030869 2.030869 2.030869
## 4  0.002777489 -1.21623362  0.2216498 6.324809 6.324809 6.324809 6.324809
```

Moment	Ordre	Formule	Estimateur
Moyenne	1	$E[X] = \int_{-\infty}^{\infty} x dF(x)$	$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
Variance	2	$E[X^2] = \int_{-\infty}^{\infty} x^2 dF(x)$	$\sigma_x^2 = \frac{1}{n} \sum_{i=1}^{n^2} (x_i - \bar{x})^2$
Skewness	3	$E[X^3] = \int_{-\infty}^{\infty} x^3 dF(x)$	$b_1 = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^3}{[\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2]^{3/2}}$
Kurtosis	4	$E[X^4] = \int_{-\infty}^{\infty} x^4 dF(x)$	$g_2 = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^4}{[\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2]^2} - 3$

Quantiles et Boxplot

Les moments (surtout de premier et second ordre) peuvent nous donner beaucoup d'informations sur les lois dont sont issus nos ?chantillons. Une autre fa?on de consid?rer cela correspond ? ordonner nos donn?es dans l'?chantillon et de les ?valuer en estimant quelle quantit? de donn?es sont inferieures ou superieures ? une valeur.

q-Quantile: si on segmente notre distribution de densit? de probabilit?s en q parts de volume ?gal, la valeur en dessous de la quelle se situent p/q des donn?es est nomm?e p-?me quantile. Typiquement on travaille avec des segmentations de notre distribution en quatre ou cent morceaux. Formellement :

Le quantile $x_{\frac{p}{q}}$ d'un variable al?atoire X est d?fini comme: $P(X \leq x_{\frac{p}{q}}) = \frac{p}{q}$
o? de fa?on equivalente: $P(X \geq x_{\frac{p}{q}}) = 1 - \frac{p}{q}$.

Comme avant, entre connaitre la distribution r?elle et essayer de “faire parler les donn?es”, il y a une grande diff?rence. On s’appuie sur notre echantillon pour essayer d’avoir plus d’informations sur nos distributions.

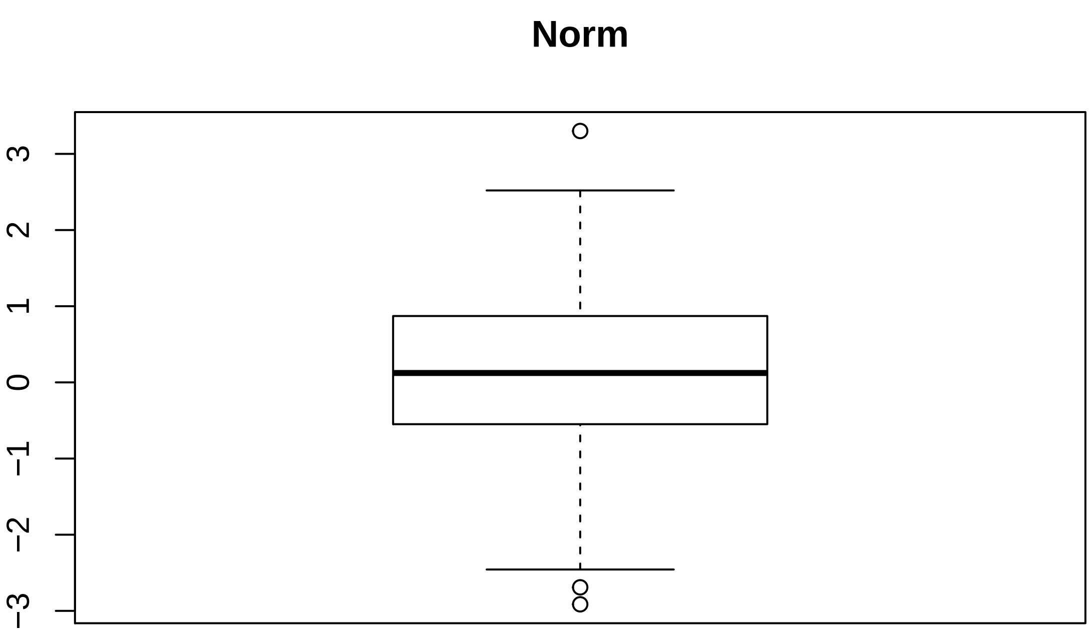
- Quantiles sp?ciaux:
  Q_1: La valeur en dessous de la quelle on a le quart des valeurs de notre echantillon.
  Q_2: La valeur en dessous de la quelle on a la moiti? des valeurs de notre echantillon, aussi connue sous le nom de m?diane.
  Q_3: La valeur en dessous de la quelle on a les trois-quarts des valeurs de notre ?chantillon.

Le boxplot nous permet de voir les valeurs entre Q_1 et Q_2, et Q_2 et Q_3, ainsi que la moyenne et l’?tendue de $+/-3\sigma$. Toute valeur en dehors de ces $+/-3\sigma$ est marqu? avec des points individuels.

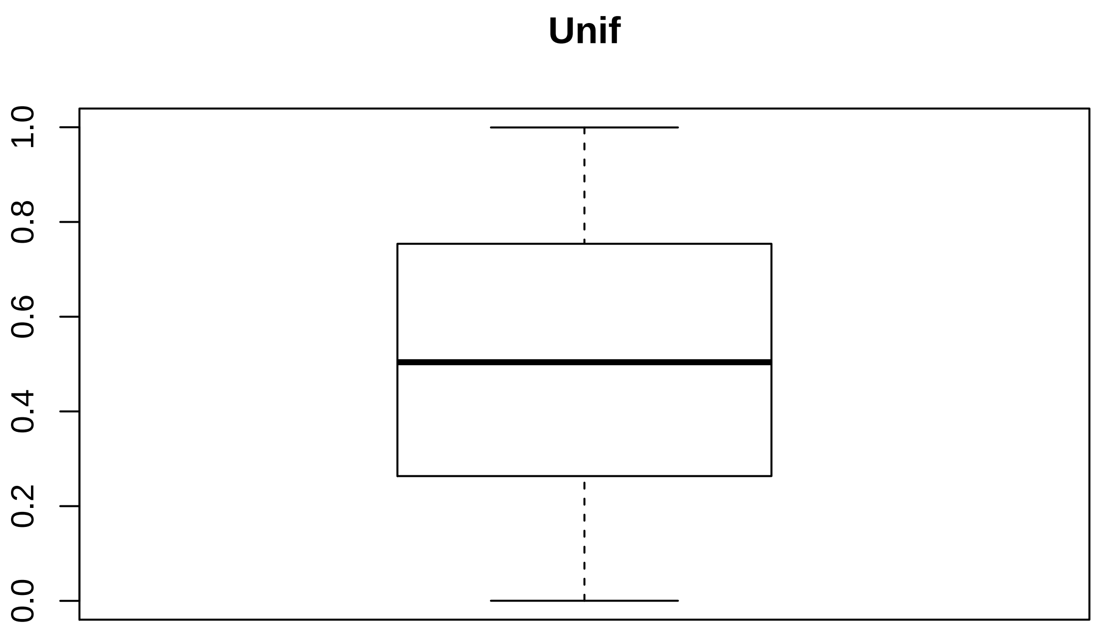
```r
Q1=quantile(x1,c(0.25,0.5,0.75,1))
Q2=quantile(x2,c(0.25,0.5,0.75,1))
Q3=quantile(x3,c(0.25,0.5,0.75,1))
Q4=quantile(x4,c(0.25,0.5,0.75,1))
Q5=quantile(x5,c(0.25,0.5,0.75,1))
Q6=quantile(x6,c(0.25,0.5,0.75,1))
Q7=quantile(x7,c(0.25,0.5,0.75,1))
data2<-data.frame(N=Q1,U=Q2,P=Q3,E=Q4,C=Q5,B=Q6,CAU=Q7)

write.table(data,file="Analyse_table2.txt")
read.table("Analyse_table2.txt",header = TRUE)
```

```
##              N          U          P        E        C        B      CAU
## 1  0.459269800  2.01684218 12.0900000 3.876404 3.876404 3.876404 3.876404
## 2  2.095597473  3.07995877 18.4284074 6.117008 6.117008 6.117008 6.117008
## 3 -0.069581043 -0.03461074  0.5152128 2.030869 2.030869 2.030869 2.030869
## 4  0.002777489 -1.21623362  0.2216498 6.324809 6.324809 6.324809 6.324809
```

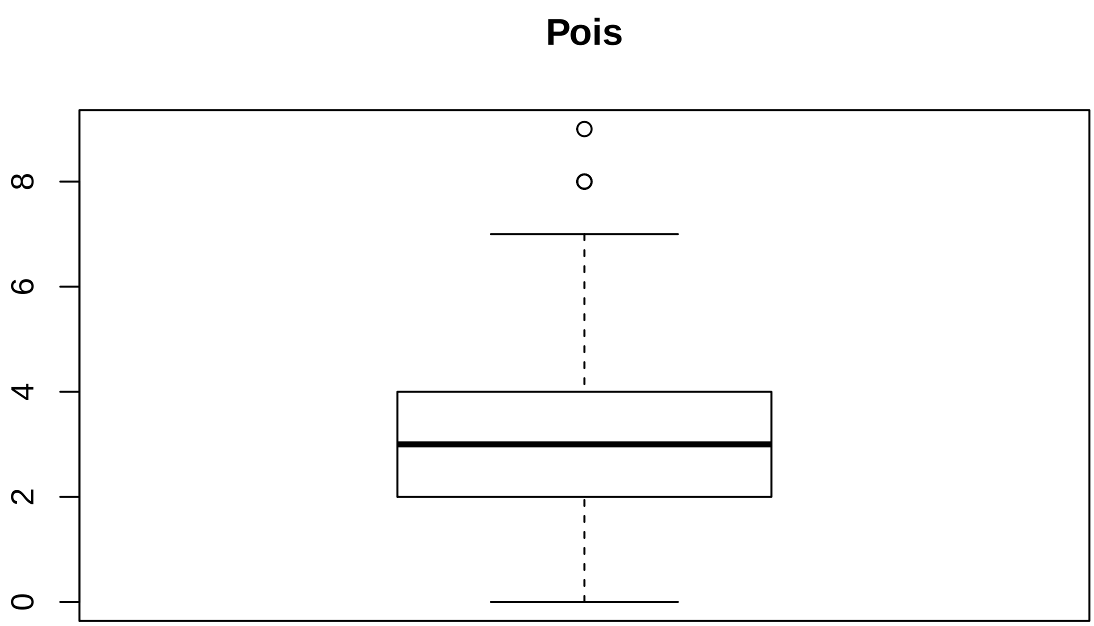
```r
boxplot(x1, main = "Norm")
```



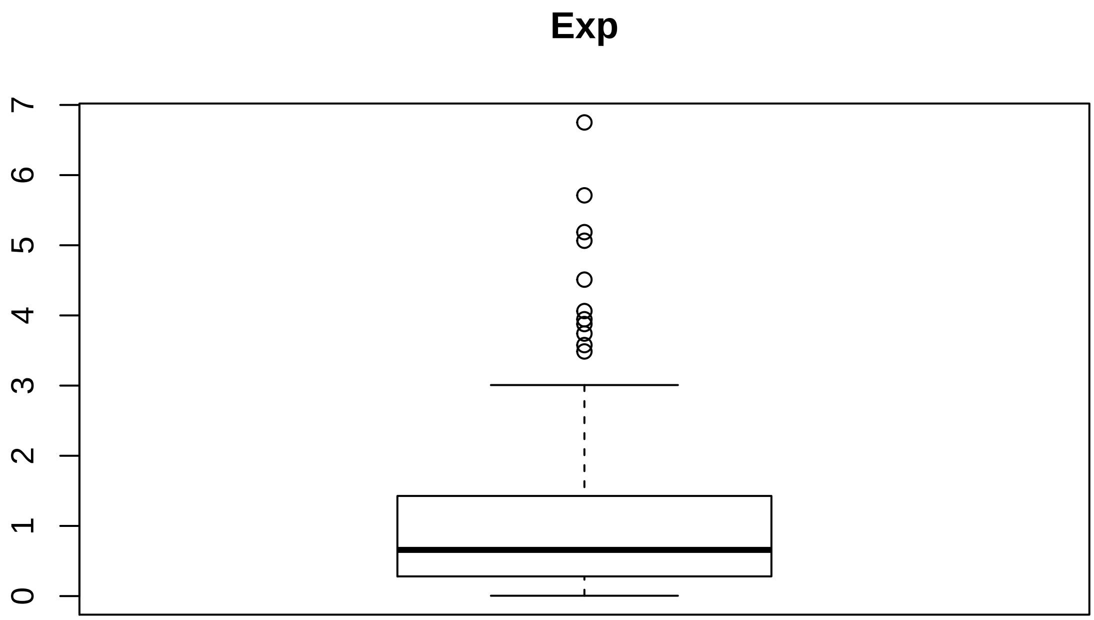

```r
boxplot(x2, main = "Unif")
```



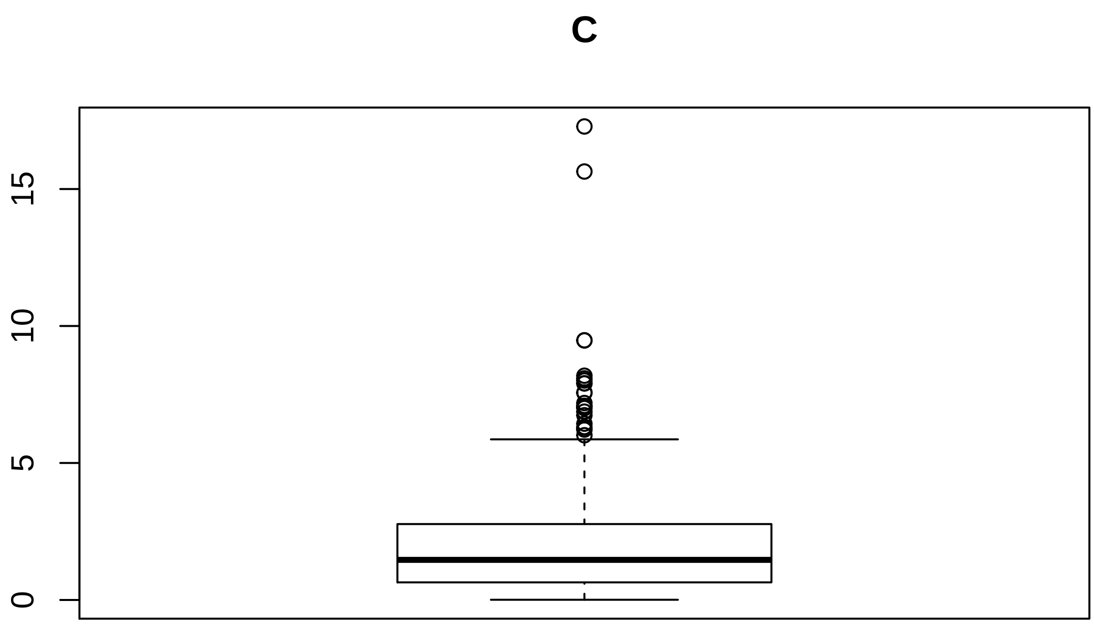

```r
boxplot(x3, main = "Pois")
```



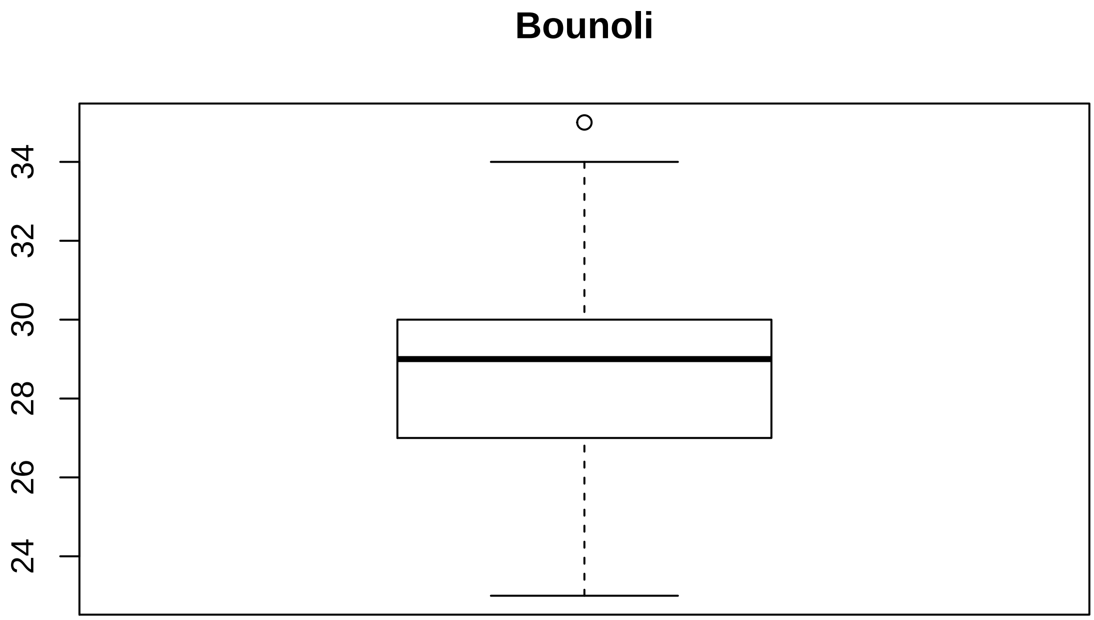

```r
boxplot(x4, main = "Exp")
```



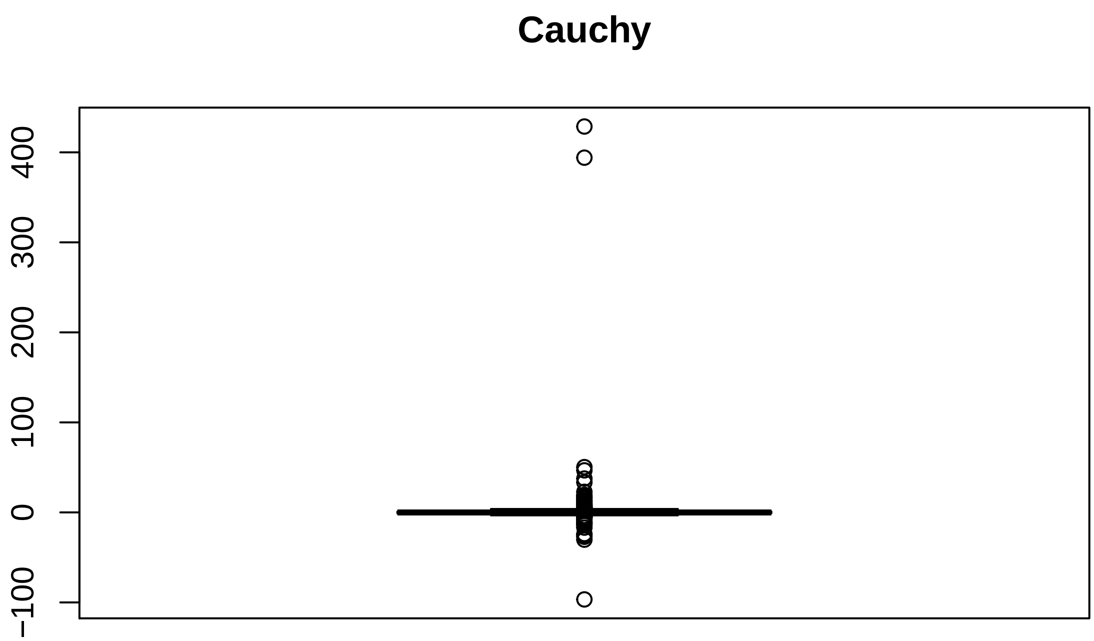

```r
boxplot(x5, main = "C")
```




```r
boxplot(x6, main = "Bounoli")
```



```r
boxplot(x7, main = "Cauchy")
```

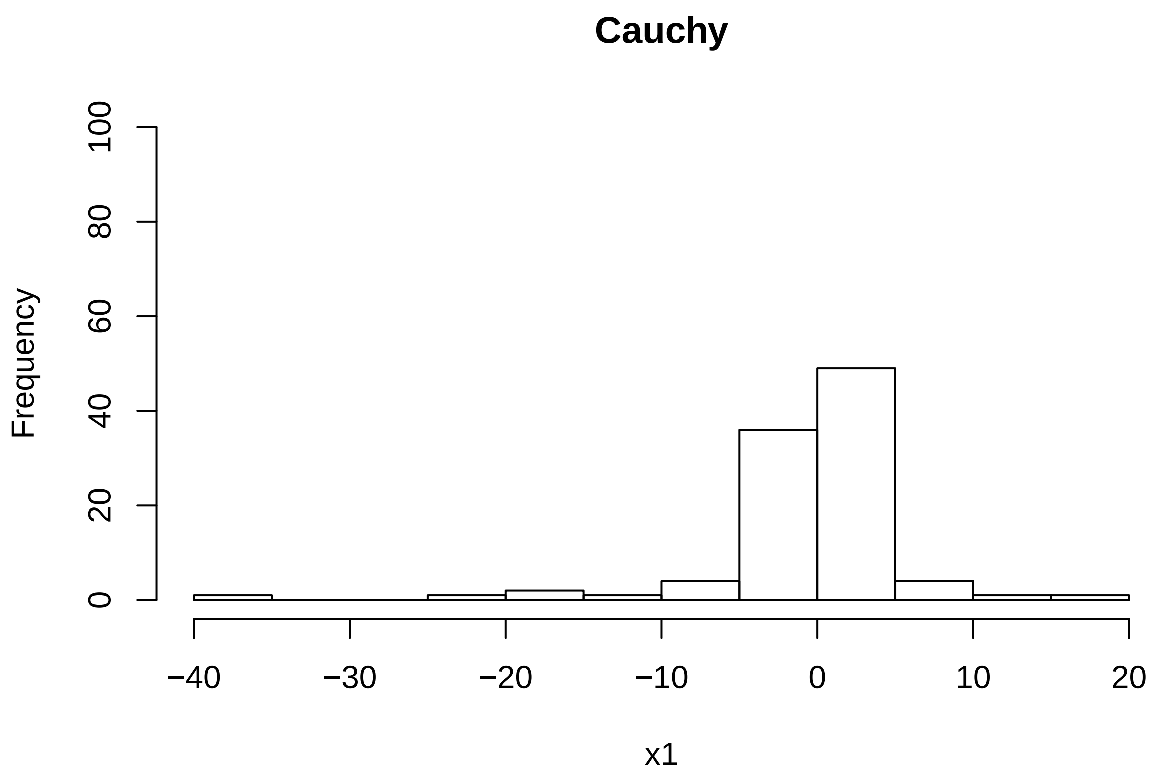


Regardez l'aide de la fonction `boxplot()` et appliquez la sur les different ensembles que vous avez g?n?r?. Pour le tableau precedent, contenant les moments de ordre 1 ? 4, ajoutez 3 colonnes qui contiennent les 3 quantiles.

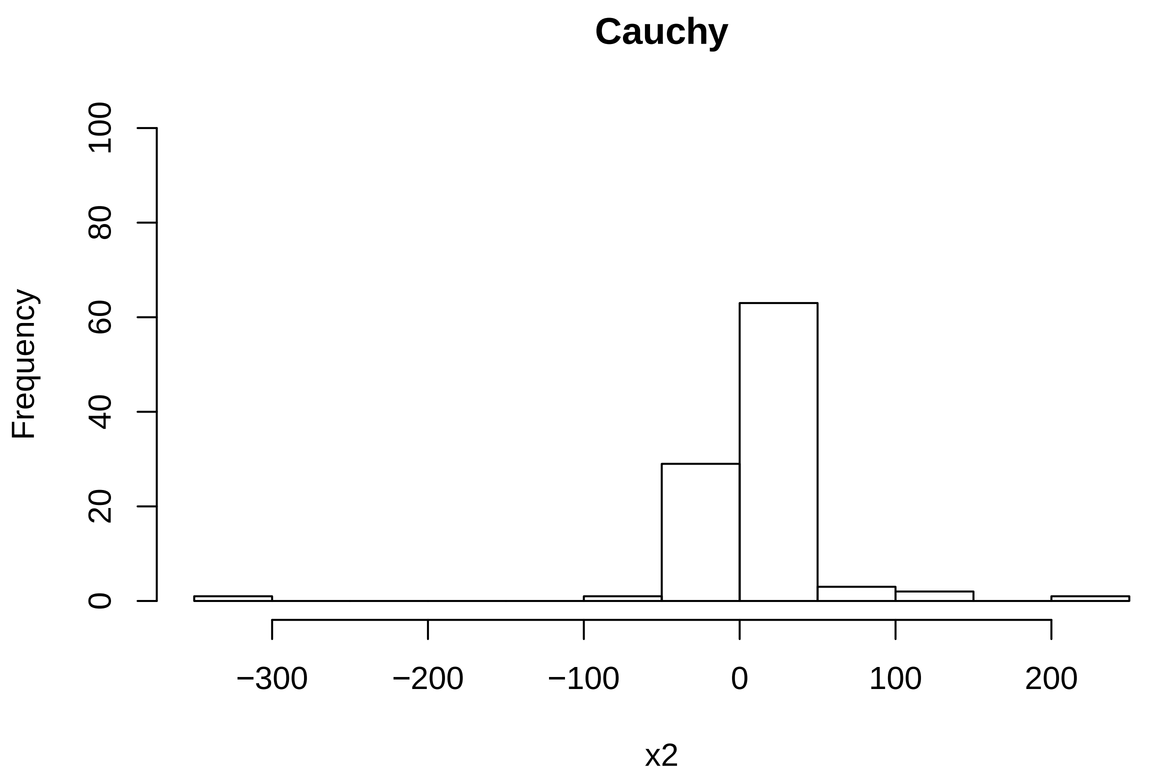
Interpr?tation visuelle

G?n?rez 3 ensembles de 100 individus avec la loi de Cauchy avec des param?trisations differentes. Effectuez toutes les demarches vues dans ce TP. Que remarquez-vous?

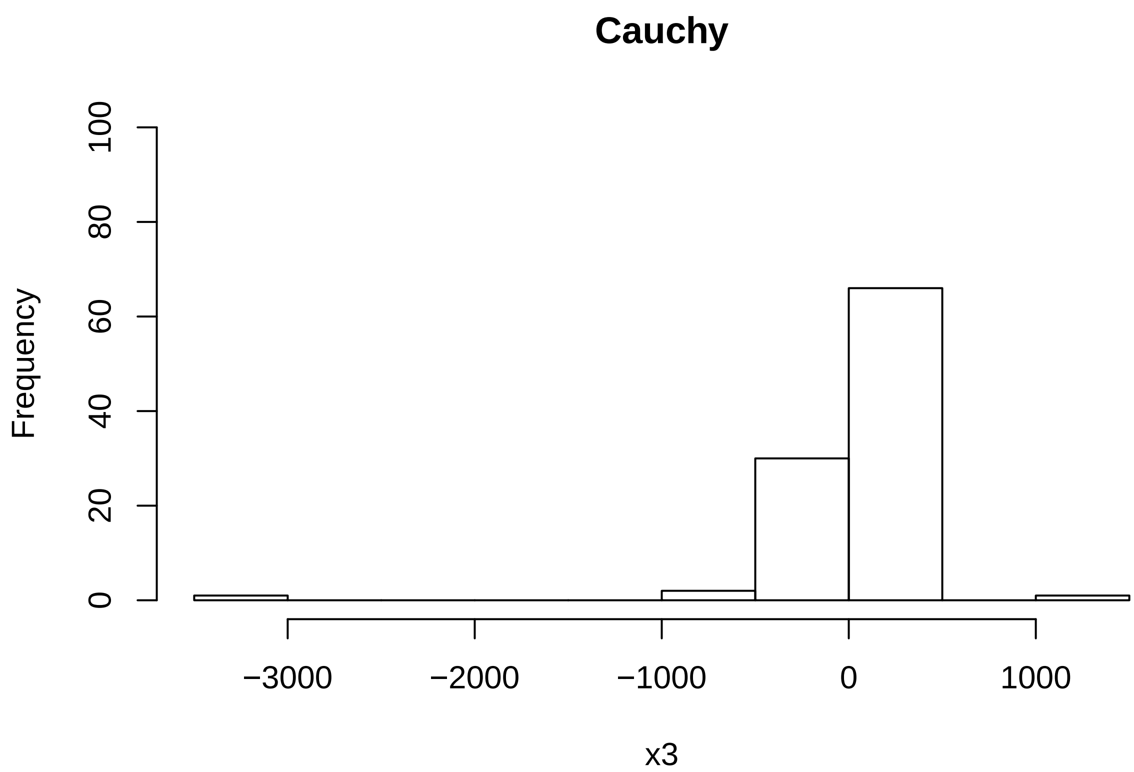
```r
x1<-rcauchy(100,0,1)
hist(x1,main="Cauchy",ylim=c(0,100))
```




```r
x2<-rcauchy(100,3,4)
hist(x2,main="Cauchy",ylim=c(0,100))
```




```r
x3<-rcauchy(100,11,25)
hist(x3,main="Cauchy",ylim=c(0,100))
```




```r
print(calcul(x1))
```

```
##            [,1]
## [1,] -0.5294648
## [2,]  6.0226310
## [3,] -2.8142666
## [4,] 15.2201453
```

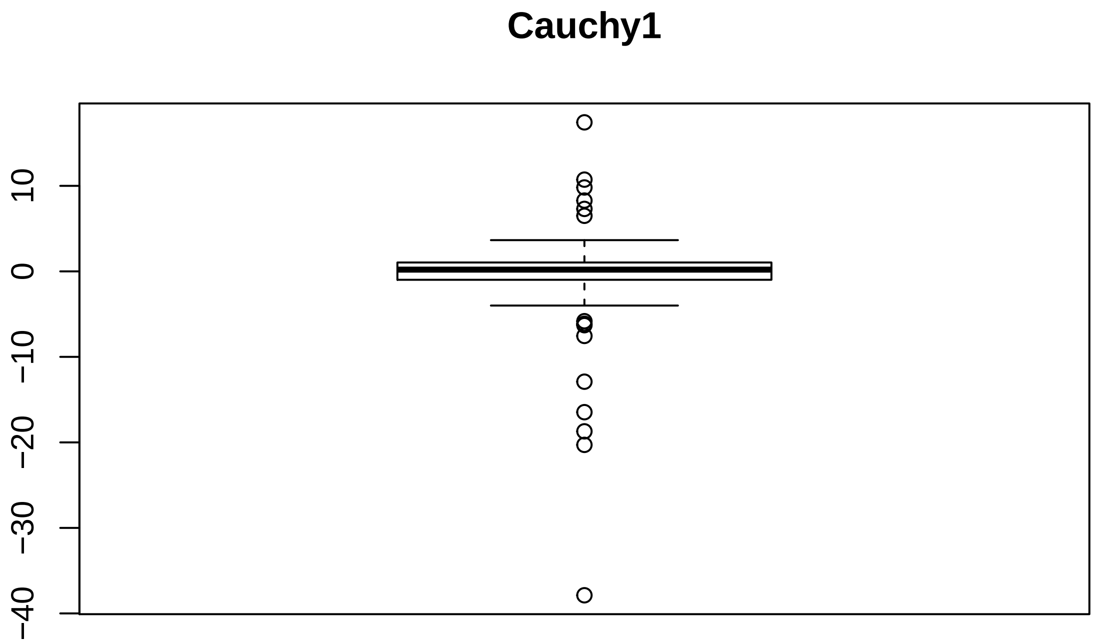
```r
print(calcul(x2))
```

```
##           [,1]
## [1,]  5.135147
## [2,] 46.283789
## [3,] -1.502045
## [4,] 24.877302
```

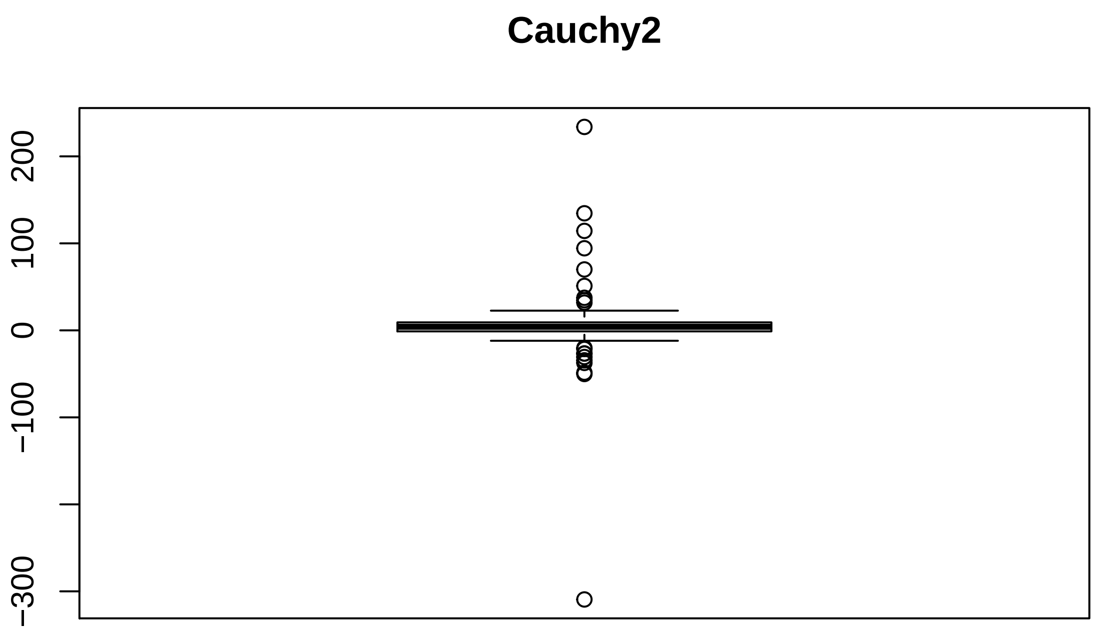
```r
print(calcul(x3))
```

```
##            [,1]
## [1,] -30.401510
## [2,] 371.147792
## [3,]  -5.855804
## [4,]  52.053951
```

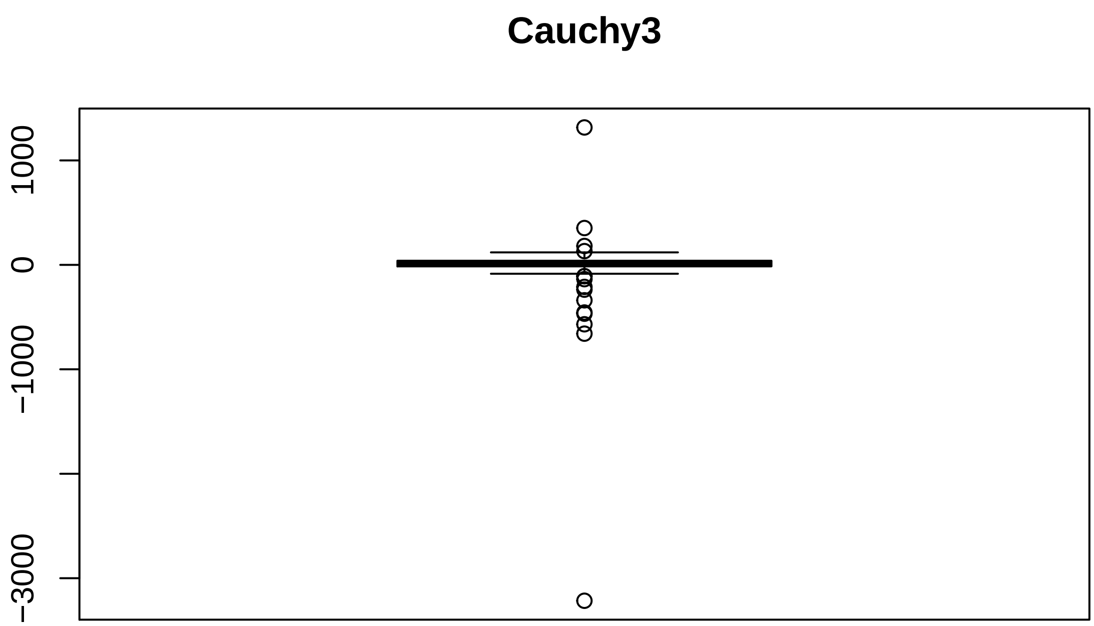
```r
boxplot(x1, main = "Cauchy1")
```




```r
boxplot(x2, main = "Cauchy2")
```




```r
boxplot(x3, main = "Cauchy3")
```




```
Q1=quantile(x1,c(0.25,0.5,0.75,1))
Q2=quantile(x2,c(0.25,0.5,0.75,1))
Q3=quantile(x3,c(0.25,0.5,0.75,1))
print(Q1)
```

```
##        25%        50%        75%       100%
## -0.9802297  0.2025043  1.0242382 17.4233225
```

```
print(Q2)
```

```
##        25%        50%        75%       100%
##  -1.106956   4.129634   9.146870 233.834017
```

```
print(Q3)
```

```
##        25%        50%        75%       100%
##  -14.23380   12.51329   39.52382 1315.49841
```

On remarque qu'une distribution de Cauchy(x, t) a une forte probabilité d'avoir des valeurs dans [x - t, x + t].