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 integralité (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 serrat ouvert ? partir de la semaine prochaine. Nous allons ?valuer, pour chaque ?tudiant, 2/5 des rendus de TP choisis al?atoirement.

Générer des données 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 générer une echantillon de 40 donn?es i.i.d. (independantes et identiquement distribu?es), associez les à 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?.

```
#Generer des donnees et les enregistrer
x1 <- rnorm(n = 40,mean=0,sd=1)
x2 <- runif(n = 40,min=0,max=1)
```

```
x3 <- rpois(n = 40,lambda = 1)
x4 <- rexp(n = 40, rate = 1)
x5 <- rchisq(n = 40, df=1)
x6 <- rbinom(n=40,size=100,prob = 0.5)
x7 <- rcauchy(n=40,location=0,scale=1)
table <- data.frame(Gaus=x1,Unif=x2,Pois=x3,Exp=x4,X2=x5,Binom=x6,Cauchy=x7)
write.csv(table,file = "data_lois.txt")
write.table(table,file = "table_lois.txt")
```

Charger des données 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 genere. Que remarquez-vous?

Pensez ? utiliser header=TRUE.

```
#Charger des donn??es depuis un fichier txt (texte) et csv (comma separated variables)
read.csv("data_lois",header = TRUE)
```

```
##     X         Gaus        Unif Pois        Exp           X2 Binom
## 1   1 -2.228868934 0.247995576    2 0.79786548 1.5617828169    51
## 2   2  0.908721294 0.093036140    0 0.63683605 0.0260895737    44
## 3   3 -1.051628341 0.841059783    0 0.03464805 0.9537782182    50
## 4   4 -0.525118785 0.990921913    1 0.24510577 1.1958655367    52
## 5   5  0.685228455 0.938198705    6 0.20054450 0.8367666393    55
## 6   6  0.256227309 0.786460882    1 1.69939544 2.6047733862    48
## 7   7 -0.098132848 0.048130236    3 0.51424575 1.8203204859    44
## 8   8 -1.283168389 0.019140818    1 0.16007431 0.0092037689    40
## 9   9  0.009362496 0.957550047    0 3.41333091 1.8389479200    53
## 10 10 -0.835433535 0.643353435    1 0.25023203 8.8022177292    45
## 11 11 -1.041441615 0.176544573    0 2.08874663 0.0117207012    55
## 12 12  0.306095916 0.119803840    2 1.29389027 0.5396117805    49
## 13 13  0.352313317 0.116436709    1 0.41187973 0.4819901175    50
## 14 14 -1.389996191 0.005084943    2 1.19406126 2.7187165586    55
## 15 15  0.049227193 0.209420965    1 0.71986785 0.4119801107    45
## 16 16  0.756708248 0.317003883    0 0.54803382 0.2577920238    47
## 17 17  0.829702761 0.794511886    2 0.07749803 2.0054031216    49
## 18 18  0.664739641 0.639865536    1 1.59505588 0.0115381643    47
## 19 19  0.935674302 0.510153611    1 1.05843008 0.6252343602    49
## 20 20  0.629676781 0.964004045    1 3.72422027 0.1878691079    55
## 21 21  0.299217923 0.757314341    1 0.07940264 0.0453727651    51
## 22 22 -1.084006039 0.376431122    0 5.29677455 1.2688167985    47
## 23 23 -0.114739193 0.122295106    1 0.96985305 0.0040966674    50
## 24 24 -0.938622085 0.907518429    0 0.73154425 0.0800121451    52
## 25 25  0.319173813 0.486942126    1 2.08433234 0.0189747882    53
## 26 26  0.079417371 0.977081300    1 1.16148740 4.2128039660    54
## 27 27  0.774205602 0.284898989    2 2.14170399 1.4512905679    48
## 28 28 -0.166453310 0.175979580    1 4.79693659 1.0936768307    59
## 29 29  1.585286284 0.494086588    1 0.44535271 3.1650373468    50
## 30 30 -0.520959005 0.055898370    1 0.63433364 3.5860950223    52
## 31 31 -2.844184604 0.382720170    0 0.84994370 0.4336873709    51
## 32 32  1.740987165 0.189425094    0 0.38251150 0.0006922237    44
## 33 33 -1.755336750 0.661384995    0 3.00414363 0.3448986815    51
## 34 34 -0.805662397 0.954114778    4 3.54012160 0.0472298031    49
```

```
## 35 35  0.668190822 0.229574132    1 0.14056230 4.6830013228    48
## 36 36  0.831036215 0.725698940    0 0.02251692 0.2457625441    43
## 37 37  0.211070411 0.353101630    1 1.96754159 0.0661977086    44
## 38 38  0.420408476 0.296922152    1 0.63338295 1.4797658615    48
## 39 39 -0.600749033 0.795569010    1 0.35407606 2.6692185695    56
## 40 40 -2.413129186 0.856200660    0 0.23249515 0.7347440077    58
##          Cauchy
## 1    1.16315726
## 2   -3.00571324
## 3   -0.78941402
## 4    0.64063649
## 5   -1.76193592
## 6    0.22560521
## 7    1.31106714
## 8   -0.81207933
## 9    0.38964118
## 10   1.73276639
## 11   0.53438741
## 12  -1.46150740
## 13  -0.56088739
## 14   3.73126369
## 15   3.12771761
## 16 -11.39943011
## 17   0.66846214
## 18   0.07623950
## 19   1.67341615
## 20  -2.72292985
## 21  -0.57529690
## 22   1.39793799
## 23  -0.32314444
## 24  -0.14125489
## 25   0.16775494
## 26  -0.48811745
## 27   1.61484641
## 28   2.50056176
## 29  -0.05380677
## 30  -3.98479654
## 31   0.56716772
## 32   0.34614380
## 33   1.50888925
## 34  -1.18601747
## 35   0.03298123
## 36  -0.56009778
## 37  -0.39055069
## 38  -0.30461188
## 39  -0.45299059
## 40   0.50792347
```

```r
read.table("table_lois",header = TRUE)
```

```
##            Gaus        Unif Pois        Exp           X2 Binom
## 1  -2.228868934 0.247995576    2 0.79786548 1.5617828169    51
## 2   0.908721294 0.093036140    0 0.63683605 0.0260895737    44
## 3  -1.051628341 0.841059783    0 0.03464805 0.9537782182    50
## 4  -0.525118785 0.990921913    1 0.24510577 1.1958655367    52
```

```
## 5   0.685228455 0.938198705    6 0.20054450 0.8367666393    55
## 6   0.256227309 0.786460882    1 1.69939544 2.6047733862    48
## 7  -0.098132848 0.048130236    3 0.51424575 1.8203204859    44
## 8  -1.283168389 0.019140818    1 0.16007431 0.0092037689    40
## 9   0.009362496 0.957550047    0 3.41333091 1.8389479200    53
## 10 -0.835433535 0.643353435    1 0.25023203 8.8022177292    45
## 11 -1.041441615 0.176544573    0 2.08874663 0.0117207012    55
## 12  0.306095916 0.119803840    2 1.29389027 0.5396117805    49
## 13  0.352313317 0.116436709    1 0.41187973 0.4819901175    50
## 14 -1.389996191 0.005084943    2 1.19406126 2.7187165586    55
## 15  0.049227193 0.209420965    1 0.71986785 0.4119801107    45
## 16  0.756708248 0.317003883    0 0.54803382 0.2577920238    47
## 17  0.829702761 0.794511886    2 0.07749803 2.0054031216    49
## 18  0.664739641 0.639865536    1 1.59505588 0.0115381643    47
## 19  0.935674302 0.510153611    1 1.05843008 0.6252343602    49
## 20  0.629676781 0.964004045    1 3.72422027 0.1878691079    55
## 21  0.299217923 0.757314341    1 0.07940264 0.0453727651    51
## 22 -1.084006039 0.376431122    0 5.29677455 1.2688167985    47
## 23 -0.114739193 0.122295106    1 0.96985305 0.0040966674    50
## 24 -0.938622085 0.907518429    0 0.73154425 0.0800121451    52
## 25  0.319173813 0.486942126    1 2.08433234 0.0189747882    53
## 26  0.079417371 0.977081300    1 1.16148740 4.2128039660    54
## 27  0.774205602 0.284898989    2 2.14170399 1.4512905679    48
## 28 -0.166453310 0.175979580    1 4.79693659 1.0936768307    59
## 29  1.585286284 0.494086588    1 0.44535271 3.1650373468    50
## 30 -0.520959005 0.055898370    1 0.63433364 3.5860950223    52
## 31 -2.844184604 0.382720170    0 0.84994370 0.4336873709    51
## 32  1.740987165 0.189425094    0 0.38251150 0.0006922237    44
## 33 -1.755336750 0.661384995    0 3.00414363 0.3448986815    51
## 34 -0.805662397 0.954114778    4 3.54012160 0.0472298031    49
## 35  0.668190822 0.229574132    1 0.14056230 4.6830013228    48
## 36  0.831036215 0.725698940    0 0.02251692 0.2457625441    43
## 37  0.211070411 0.353101630    1 1.96754159 0.0661977086    44
## 38  0.420408476 0.296922152    1 0.63338295 1.4797658615    48
## 39 -0.600749033 0.795569010    1 0.35407606 2.6692185695    56
## 40 -2.413129186 0.856200660    0 0.23249515 0.7347440077    58
##          Cauchy
## 1    1.16315726
## 2   -3.00571324
## 3   -0.78941402
## 4    0.64063649
## 5   -1.76193592
## 6    0.22560521
## 7    1.31106714
## 8   -0.81207933
## 9    0.38964118
## 10   1.73276639
## 11   0.53438741
## 12  -1.46150740
## 13  -0.56088739
## 14   3.73126369
## 15   3.12771761
## 16 -11.39943011
## 17   0.66846214
```

```
## 18   0.07623950
## 19   1.67341615
## 20  -2.72292985
## 21  -0.57529690
## 22   1.39793799
## 23  -0.32314444
## 24  -0.14125489
## 25   0.16775494
## 26  -0.48811745
## 27   1.61484641
## 28   2.50056176
## 29  -0.05380677
## 30  -3.98479654
## 31   0.56716772
## 32   0.34614380
## 33   1.50888925
## 34  -1.18601747
## 35   0.03298123
## 36  -0.56009778
## 37  -0.39055069
## 38  -0.30461188
## 39  -0.45299059
## 40   0.50792347
```

```
# si on utilise read.csv, il y aura des virgules entre chque data on a stocké.
```

Tracer les données

Genrrez un vecteur qui contient 10 rralisations 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)')

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?

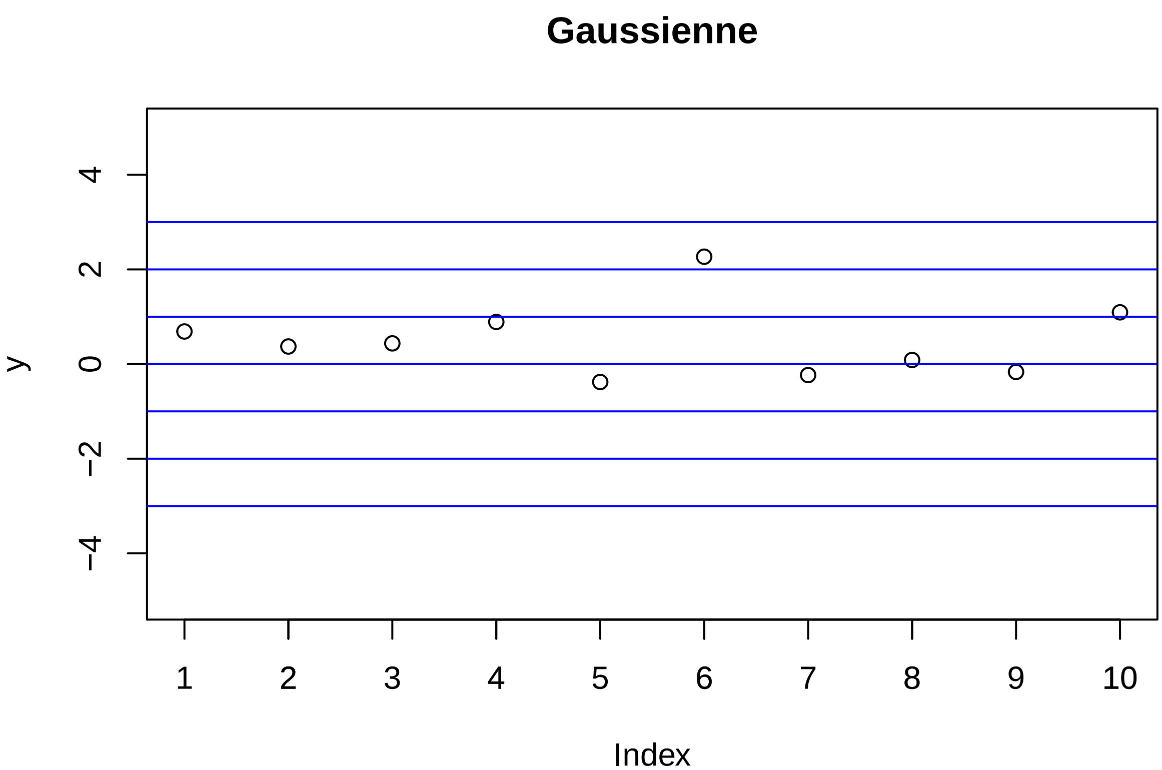
Effectuer la même chose avec des vecteurs contenant 100 et 1000 valeurs. Que remarquez vous?

Chargez le fichier 'distribution_inconue_1_100_realisations.csv' que vous pouvez trouver dans le même emplacement que ce fichier.

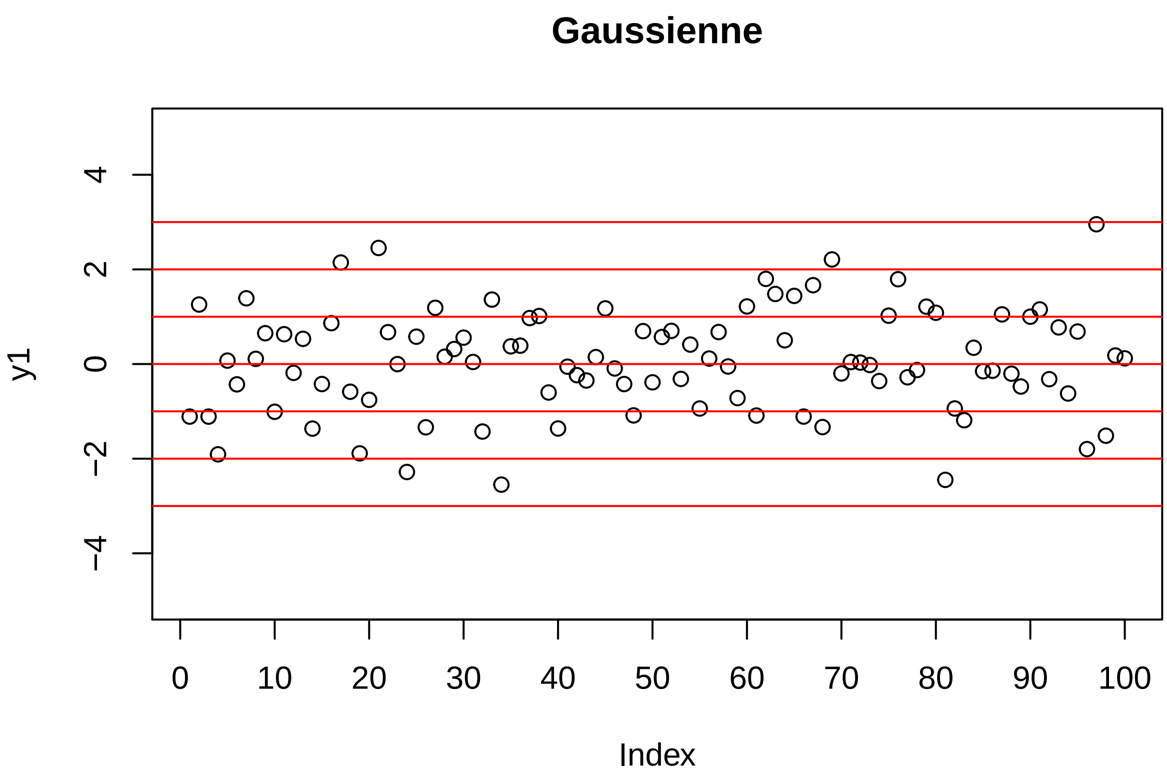
Est-ce que vous pouvez conclure quelque chose sur cette distribution, à partir d'une visualisation?

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

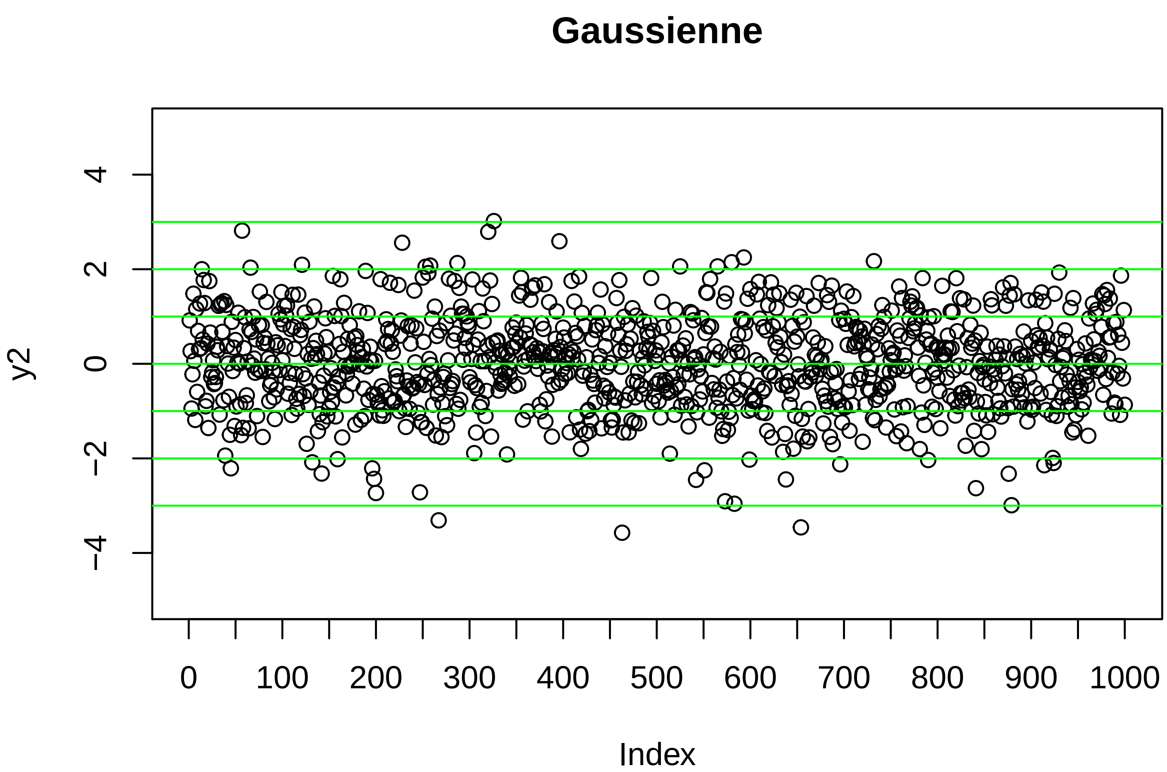
```
#Tracer les données
y <- rnorm(n = 10,mean=0,sd=1)
plot(y,main ="Gaussienne",ylim = c(-5,5))
axis(1,at=seq(1,10,1))
abline(h=c(0,1,-1,2,-2,3,-3),col = ("blue"))
```



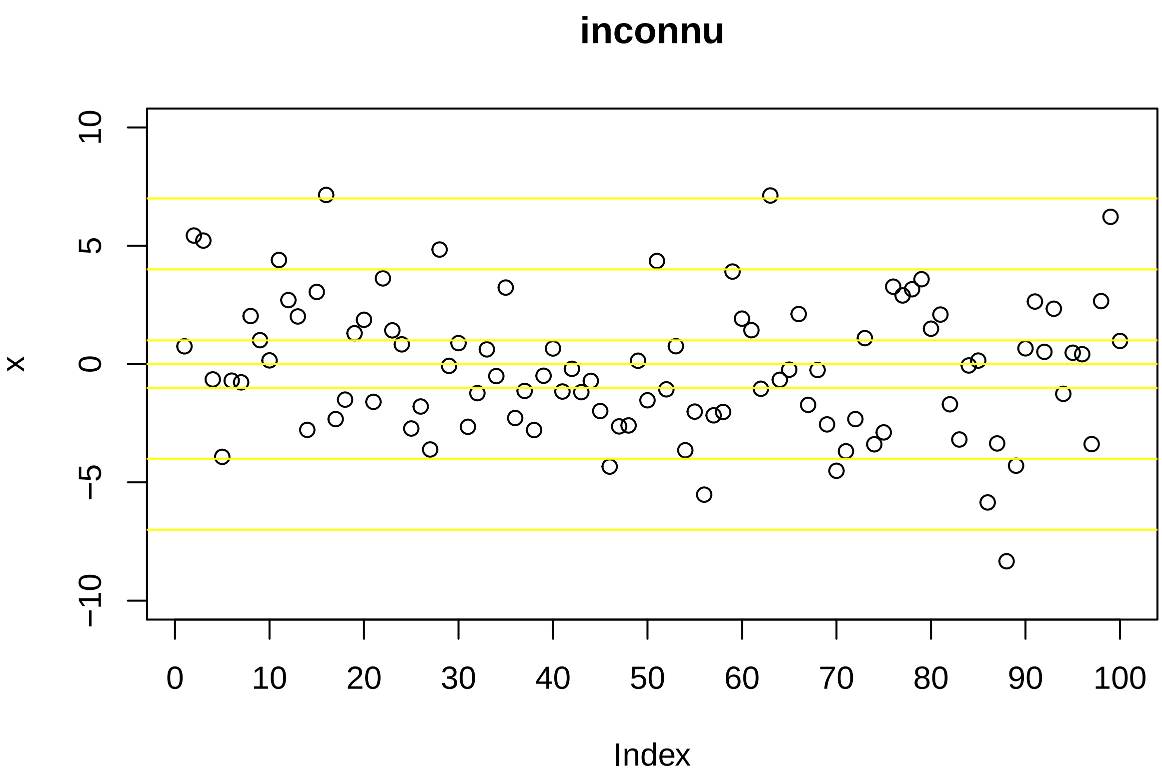

```r
# tous les 10 points sont proche de 0,en plus ils sont tous entre [-1,1],et il n'y a pas
# beaucoup de donneé aberrantes
y1 <- rnorm(n = 100,mean = 0,sd = 1)
plot(y1,main ="Gaussienne",ylim = c(-5,5),xaxt="n")
axis(1,at=seq(0,100,10))
abline(h=c(0,1,-1,2,-2,3,-3),col = ("red"))
```



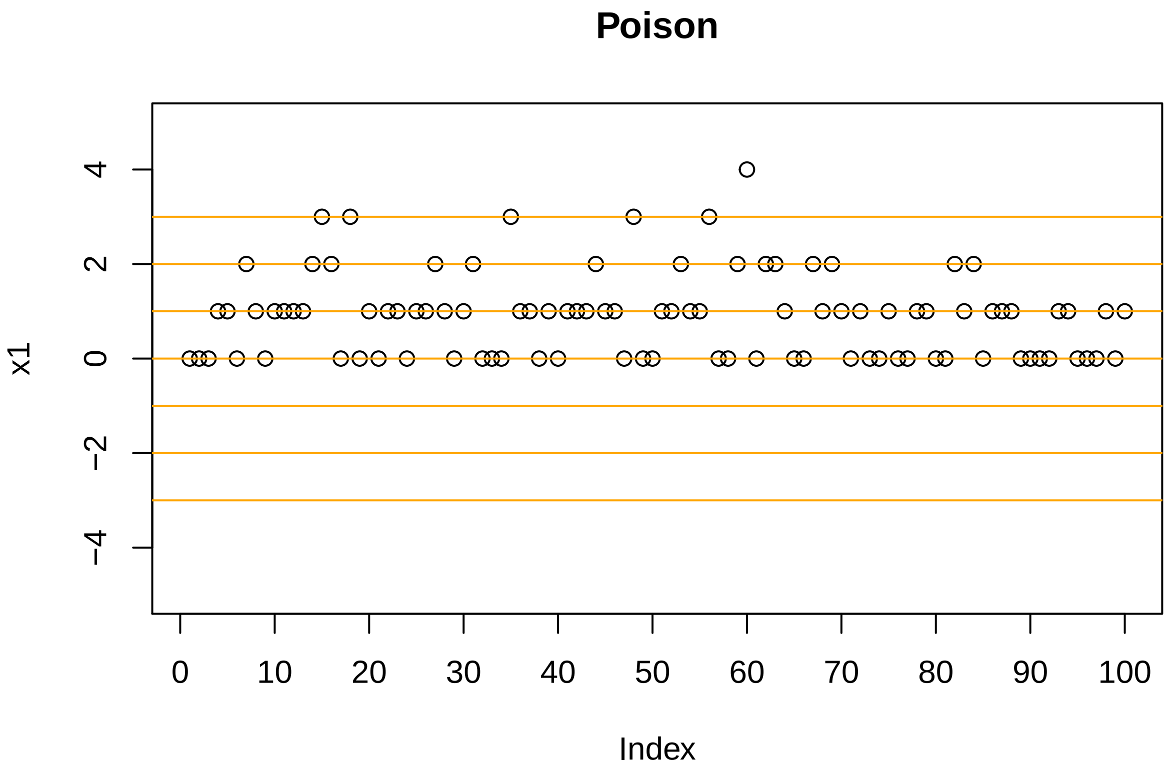

```
# tous les 100 sont tous entre [-3,3], leur degree de dispersion est plus grand que celle de
# 10 point
y2 <- rnorm(n = 1000,mean = 0,sd = 1)
plot(y2,main ="Gaussienne",ylim = c(-5,5),xaxt="n")
axis(1,at=seq(0,1000,50))
abline(h=c(0,1,-1,2,-2,3,-3),col = ("green"))
```



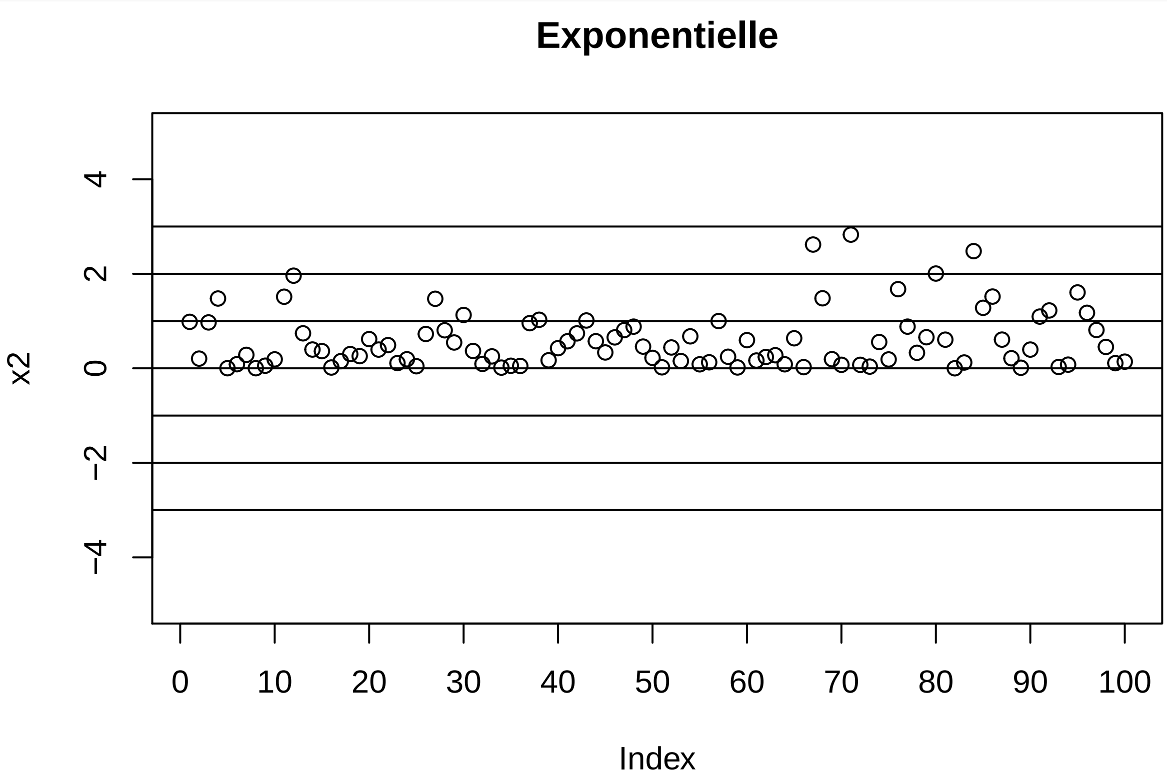

```
# il y a des point dont valeur est superieur que 3 ou moins que -3, c'est à dire si on
# effectue beaucoup de fois d'echantillon , il y aura surement des point aberrantes.
# par contre , pluspart de point se concentrent sur l'esperecne
Data <- read.csv("distribution_inconue_1_100_realisations.csv", header=TRUE)
x <- unlist(Data[2])
plot(x,main ="inconnu",ylim = c(-10,10),xaxt="n")
axis(1,at=seq(0,100,10))
abline(h=c(0,1,-1,4,-4,7,-7),col = ("yellow"))
```



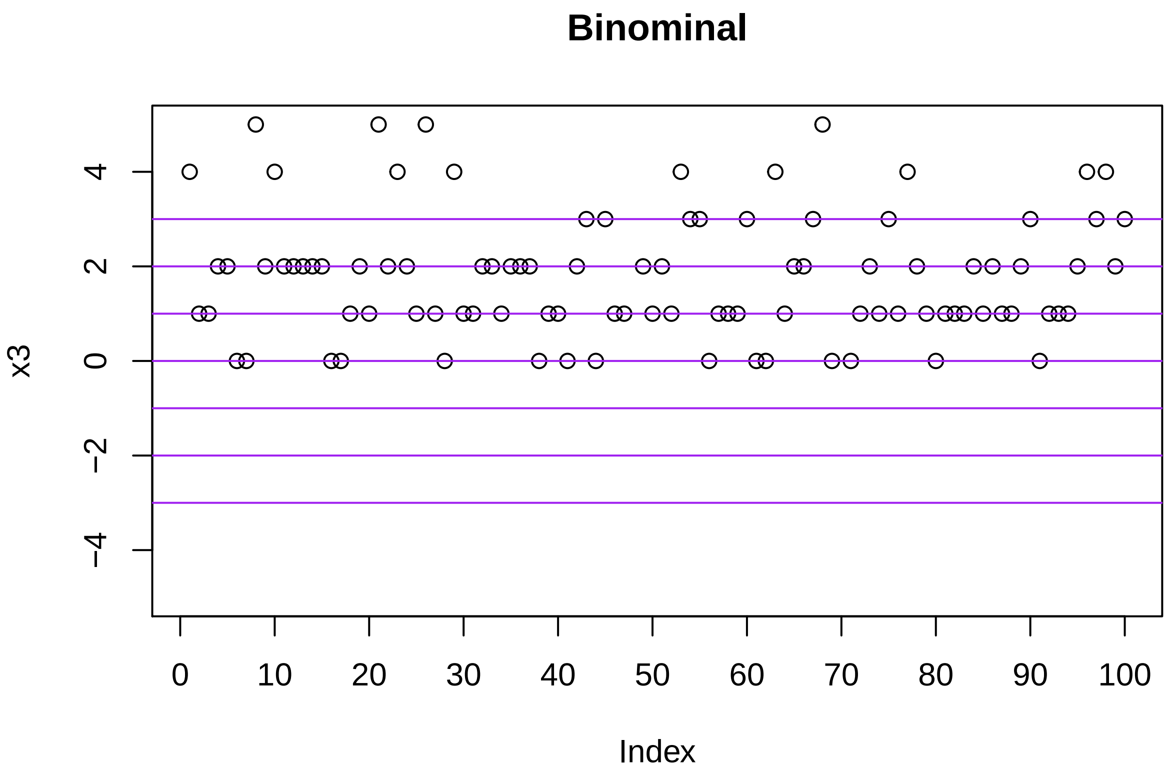

```
# j'ai remarqué pluspart presque 80% de point se situent entre [-4,4], il n'y a pas beaucoup
# de point aberrantes,sa distribution remsemble à celle de Gaussien
x1 <- rpois(n = 100,lambda = 1)
plot(x1,main ="Poison",ylim = c(-5,5),xaxt="n")
axis(1,at=seq(0,100,10))
abline(h=c(0,1,-1,2,-2,3,-3),col = ("orange"))
```




```
x2 <- rexp(n = 100, rate = 2)
plot(x2,main ="Exponentielle",ylim = c(-5,5),xaxt="n")
axis(1,at=seq(0,100,10))
abline(h=c(0,1,-1,2,-2,3,-3),col = ("black"))
```



```r
x3 <- rbinom(n=100,size=100,prob = 0.02)
plot(x3,main ="Binominal",ylim = c(-5,5),xaxt="n")
axis(1,at=seq(0,100,10))
abline(h=c(0,1,-1,2,-2,3,-3),col= ("purple"))
```




```r
#j'ai remarqué que pluspart de point sont proche de leur esperence, et plus la variance est grande, plu
```

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?

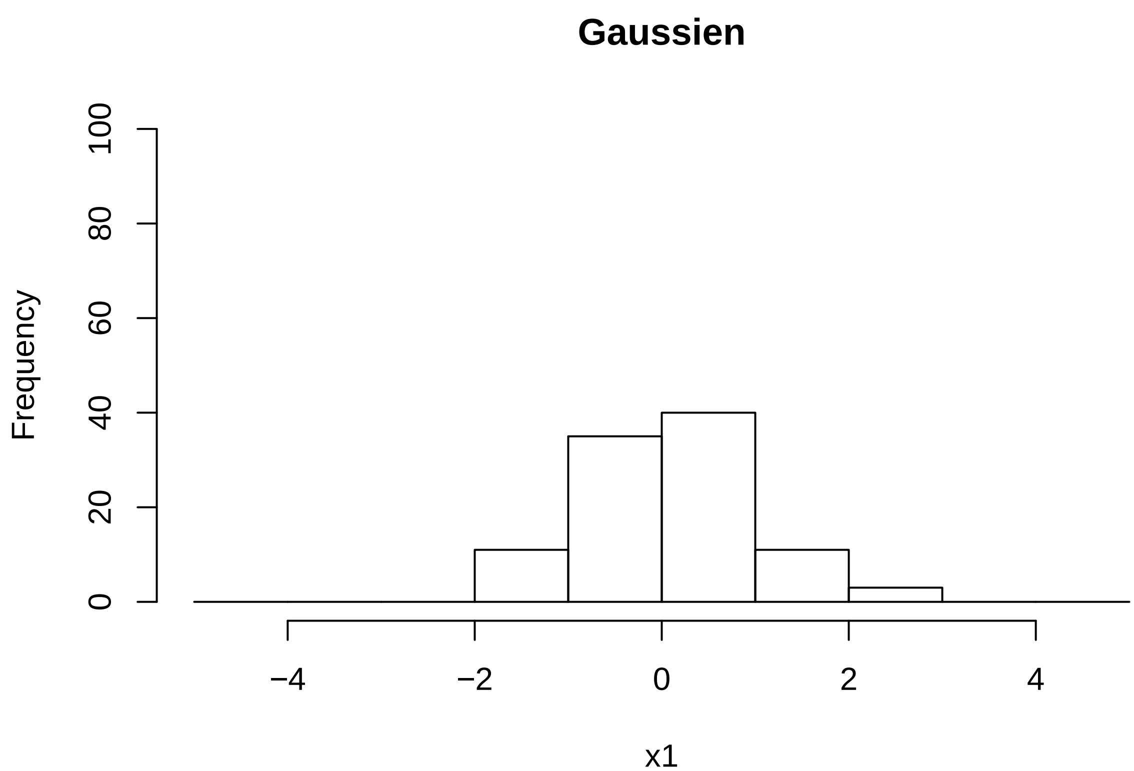
Testez les differents param??trages de la fonction: breaks et freq.

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

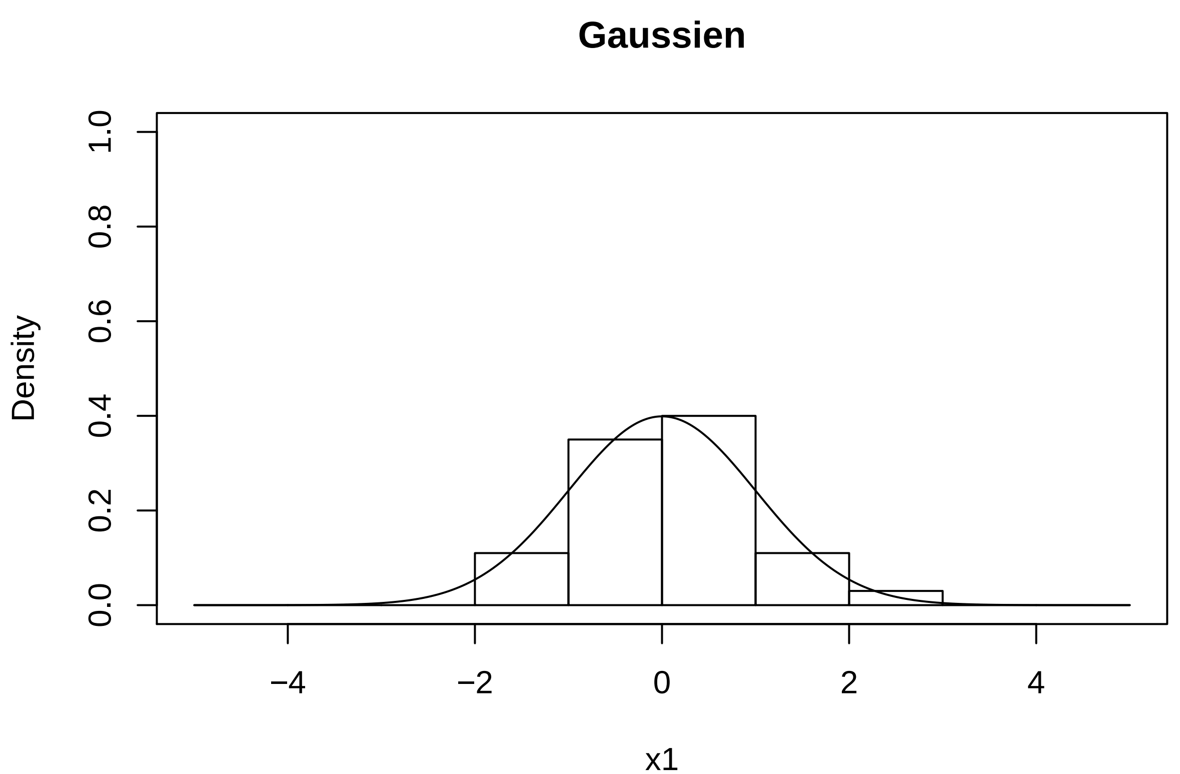
Par ailleurs, regardez les fonctions de type d*func*(n,p1,p2,. . .). Elles peuvent vous donner la distribution th?orique que vous devriez obtenir. Superposez deux plots en utilisant `par(new=TRUE)` puis en plottant la distribution correspondante au histogramme que vous visualisez.

```r
#Histogrammess
# quand j'ai mis seq(-5,5,2), les deux Histogrammes
# ont pas la meme form , pourquoi?
x1 <- rnorm(n = 100,mean=0,sd=1)
```

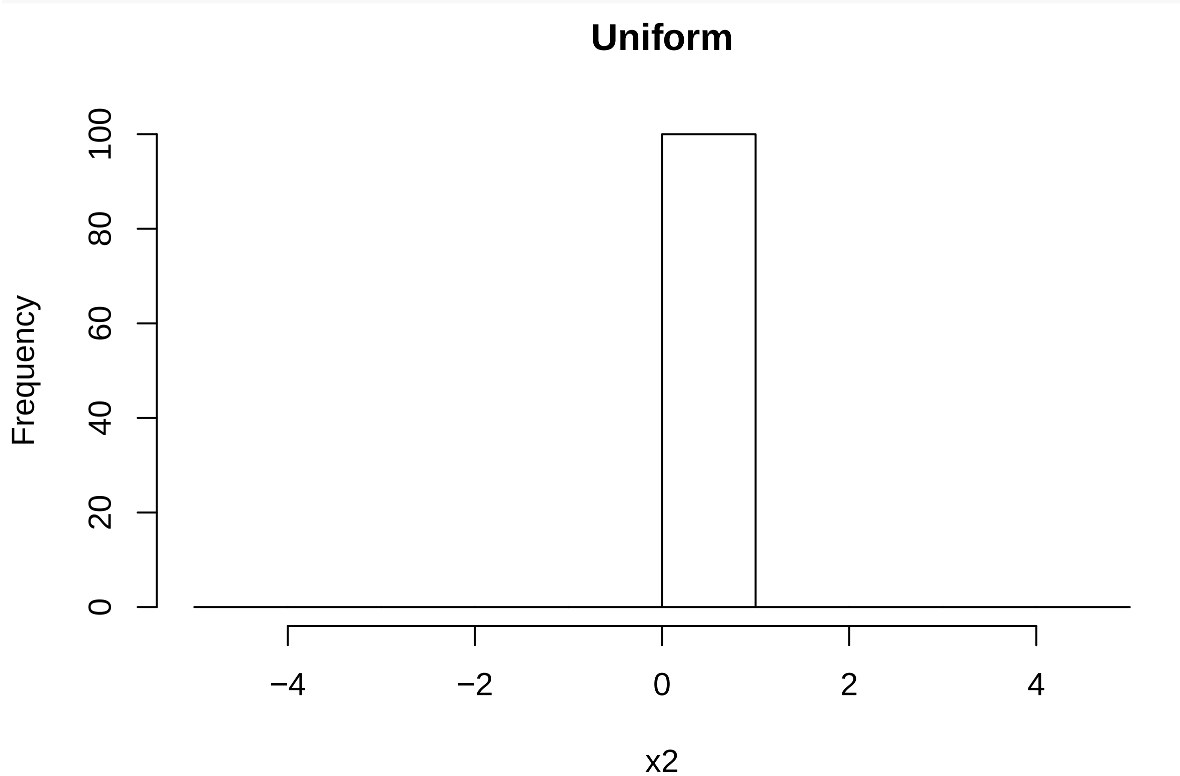
```r
x2 <- runif(n = 100,min=0,max=1)
x3 <- rpois(n = 100,lambda = 1)
x4 <- rexp(n = 100, rate = 1)
x5 <- rchisq(n = 100, df=1)
x6 <- rbinom(n=100,size=100,prob = 0.5)
x7 <- rcauchy(n=100,location=0,scale=1)
Data <- read.csv("distribution_inconue_1_100_realisations.csv", header=TRUE)
x8 <- unlist(Data[2])
table <- data.frame(Gaus=x1,Unif=x2,Pois=x3,Exp=x4,X2=x5,Binom=x6,Cauchy=x7,Inconnu=x8)
write.table(table,file = "data_8_lois.txt")
x <- seq(-5,5,0.01)
z1 <- dnorm(x,0,1)
hist(x1, seq(-5,5,1),freq = T,main ="Gaussien",xlim = c(-5,5),ylim = c(0,100))
```



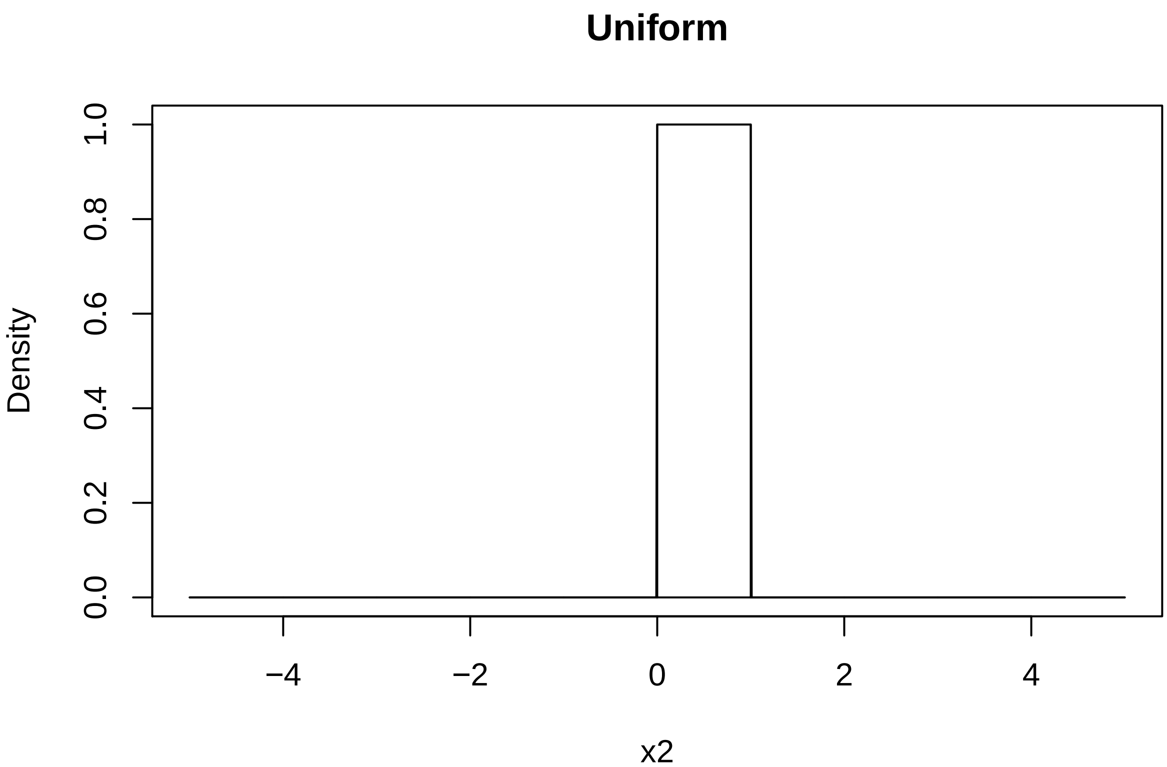

```r
hist(x1, seq(-5,5,1),freq = F,main ="Gaussien",xlim = c(-5,5),ylim = c(0,1))
par(new=TRUE)
plot(x,z1,xlim = c(-5,5),type='l',ylim = c(0,1),xlab="",ylab="",yaxt="n",xaxt="n")
```



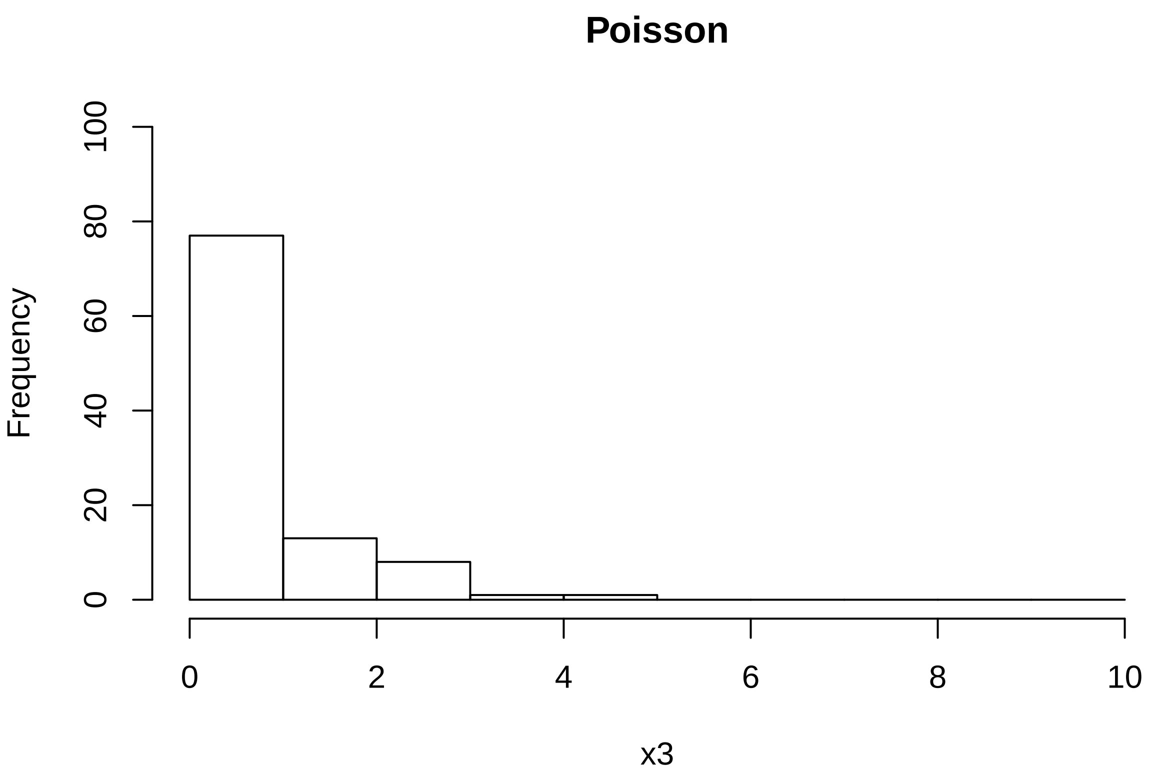

```
hist(x2,breaks = seq(-5,5,1),freq = T,ylim = c(0,100),main ="Uniform")
```




```
hist(x2,breaks = seq(-5,5,1),freq = F,ylim = c(0,1),main ="Uniform")
par(new=TRUE)
z2 <- dunif(x,0,1)
plot(x,z2,xlim = c(-5,5),type='l',ylim = c(0,1),xlab="",ylab="",yaxt="n",xaxt="n")
```

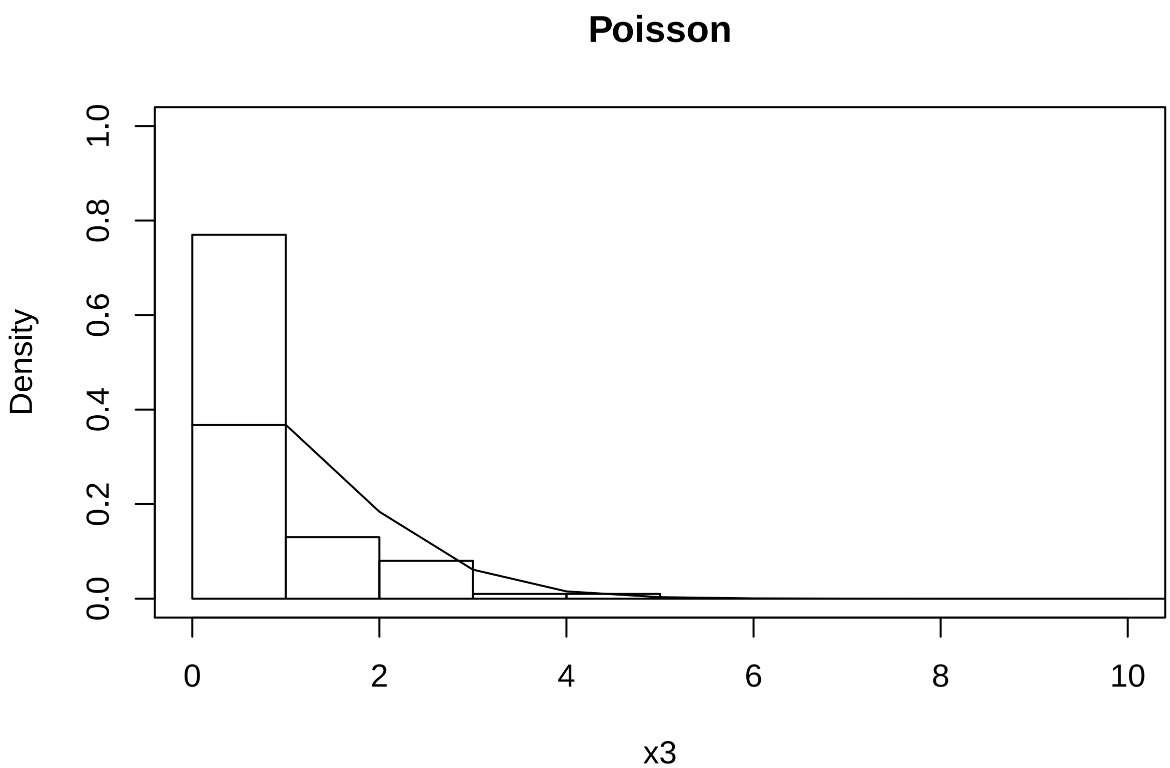



```r
hist(x3, breaks = seq(0,10,1),freq = T,ylim = c(0,100),main ="Poisson")
```

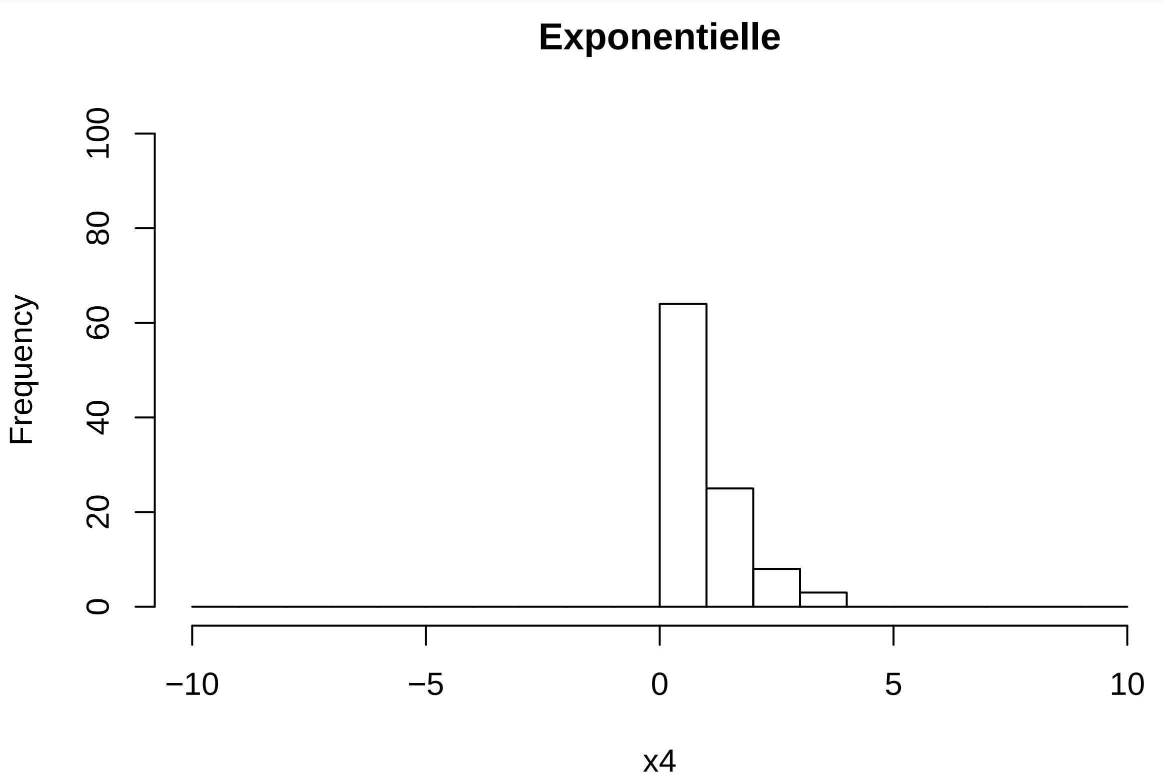



```r
hist(x3, breaks = seq(0,10,1),freq = F,ylim = c(0,1),main ="Poisson")
par(new=TRUE)
```

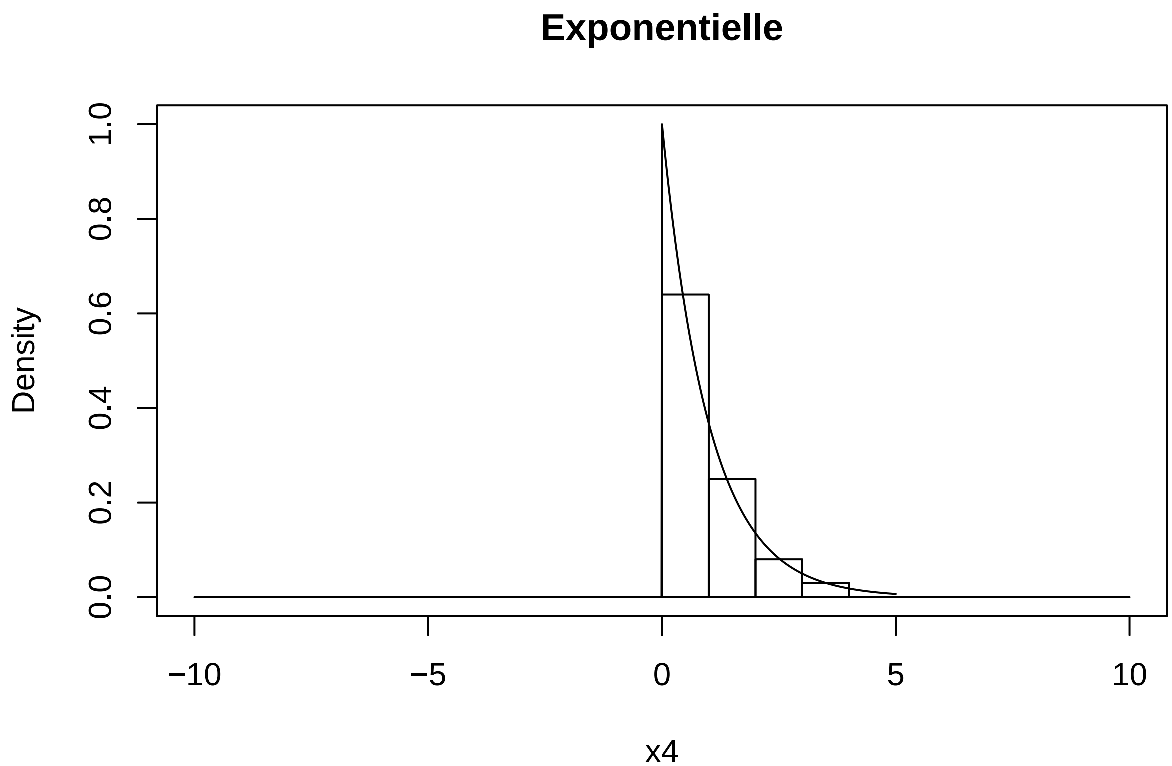
```r
k <- seq(0,100,1)
z3 <- dpois(k,lambda = 1)
plot(k,z3,xlim = c(0,10),type='l',ylim = c(0,1),xlab="",ylab="",yaxt="n",xaxt="n")
```



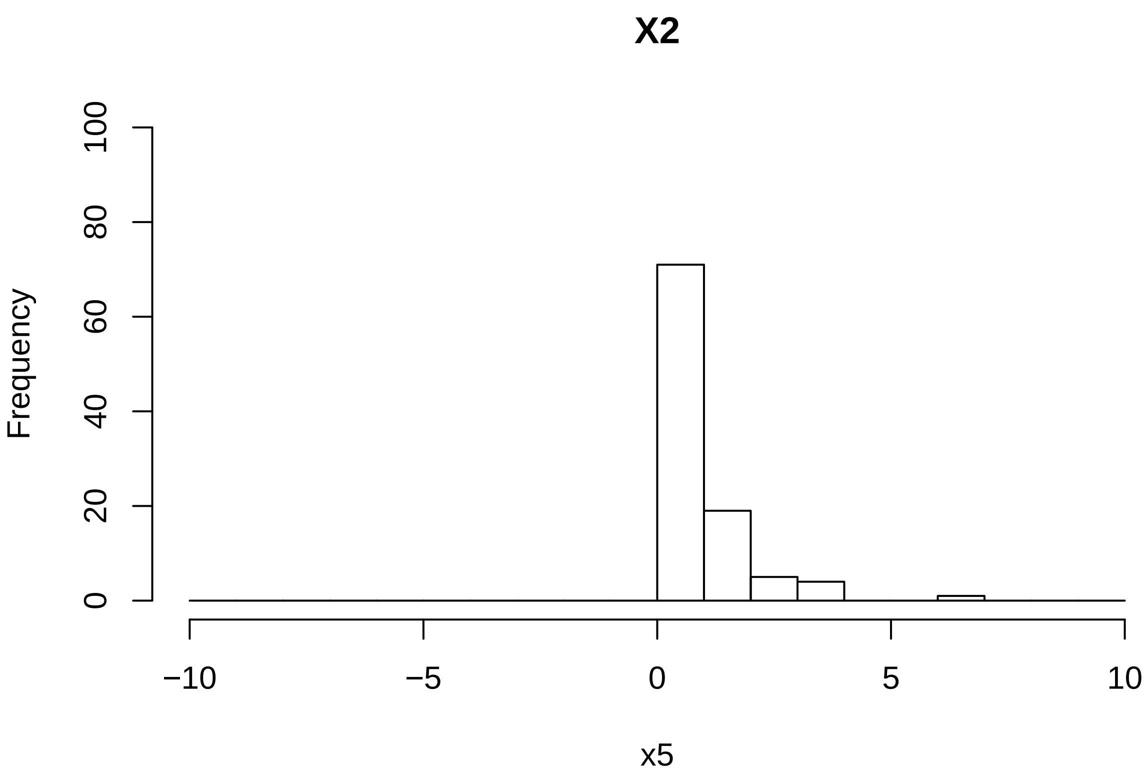

```r
hist(x4, breaks = seq(-10,10,1),freq = T,ylim = c(0,100),main ="Exponentielle")
```



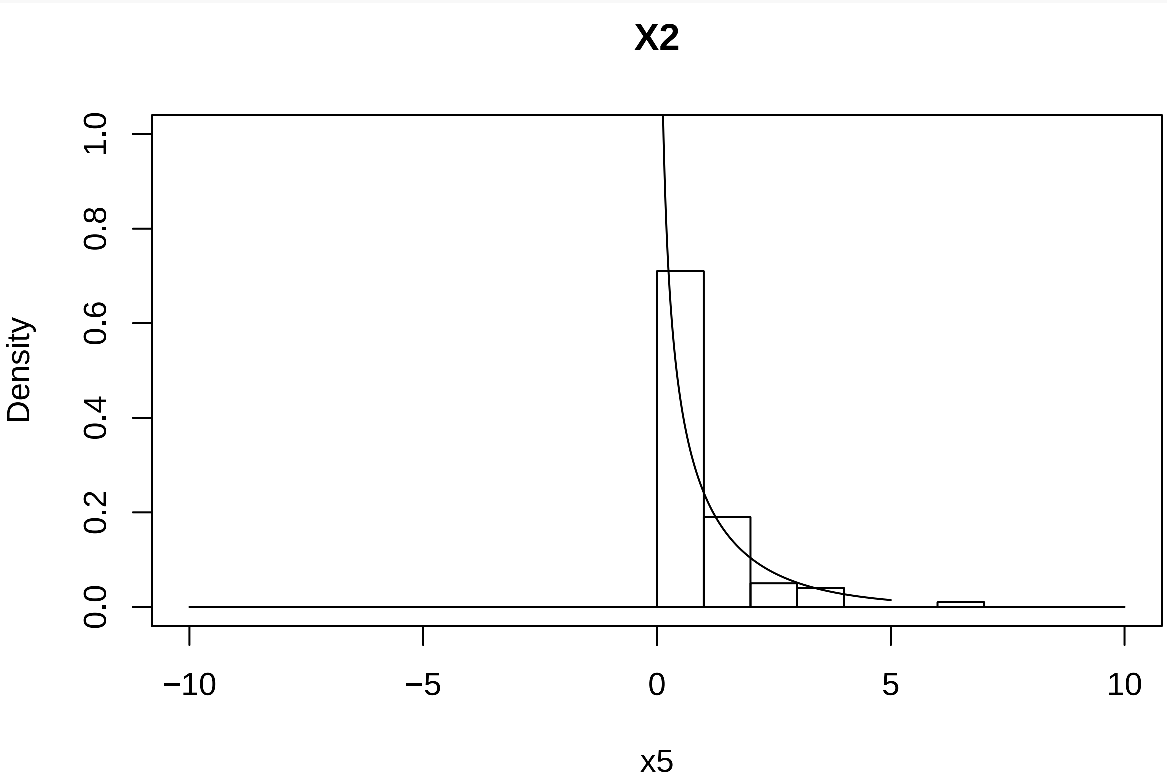
```r
hist(x4, breaks = seq(-10,10,1),freq = F,ylim = c(0,1),main ="Exponentielle")
par(new=TRUE)
z4 <- dexp(x,rate = 1)
plot(x,z4,xlim = c(-10,10),type='l',ylim = c(0,1),xlab="",ylab="",yaxt="n",xaxt="n")
```



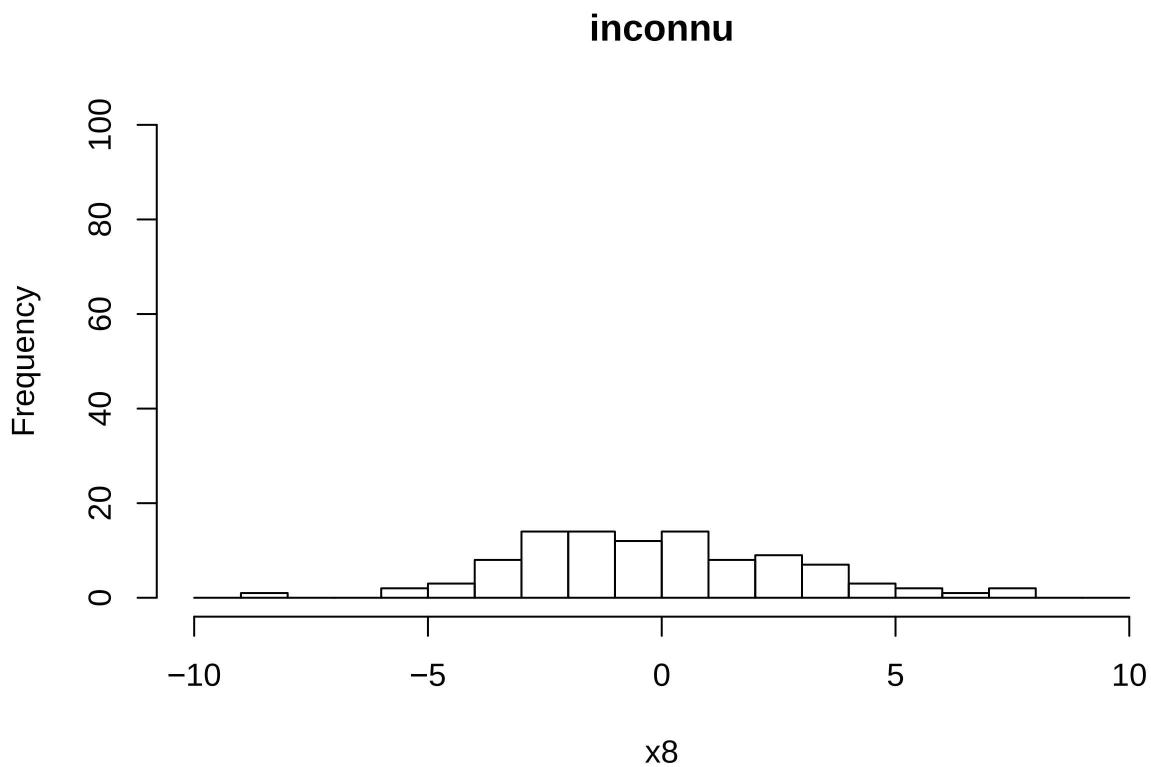

```r
hist(x5, breaks = seq(-10,10,1),freq = T,ylim = c(0,100),main ="X2")
```



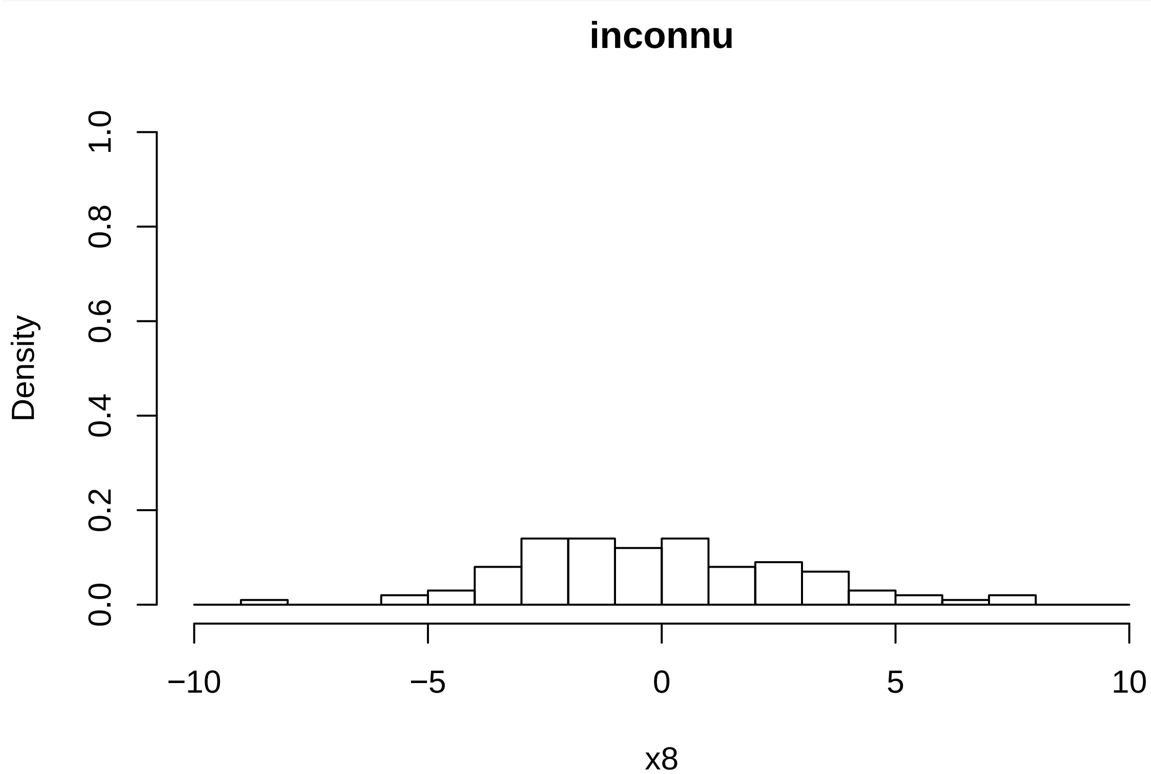

```r
hist(x5, breaks = seq(-10,10,1),freq = F,ylim = c(0,1),main ="X2")
par(new=TRUE)
z5 <- dchisq(x,df=1)
plot(x,z5,xlim = c(-10,10),type='l',ylim = c(0,1),xlab="",ylab="",yaxt="n",xaxt="n")
```



```r
hist(x8, breaks = seq(-10,10,1),freq = T,ylim = c(0,100),main ="inconnu")
```




```r
hist(x8, breaks = seq(-10,10,1),freq = F,ylim = c(0,1),main ="inconnu")
```



```
# Le histogramme de "inconnu" resemble beaucoup à celui de Gaussien
```

Moments d'ordre

Les moments d'ordre élevé pour une distribution nous donnent des informations liés à la forme des écart à 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 négatif —> plus notre densité est dissymétrique vers la gauche.
  Kurtosis petit —> Plus l'extremit? de la densité 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.\

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	$E[X^4] = \int_{-\infty}^{\infty} x^4 dF(x)$	$g_2 = \frac{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^4}{[\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2]^2} - 3$

```
library(fBasics)
```

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

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

```
table <- read.table("data_8_lois.txt",header = TRUE)

x1 <- unlist(table[1])
x2 <- unlist(table[2])
x3 <- unlist(table[3])
x4 <- unlist(table[4])
x5 <- unlist(table[5])
x6 <- unlist(table[6])
x7 <- unlist(table[7])
x8 <- unlist(table[8])
noms <- c("Gus","Unif","Pois","Exp","X2","Binom","Cauchy","Inconnu")
Mean <- c(mean(x1),mean(x2),mean(x3),mean(x4),mean(x5),mean(x6),mean(x7),mean(x8));
Var <- c(var(x1),var(x2),var(x3),var(x4),var(x5),var(x6),var(x7),var(x8));
Skewness <- c(skewness(x1),skewness(x2),skewness(x3),skewness(x4),skewness(x5),skewness(x6),skewness(x7
Kurtosis <- c(kurtosis(x1),kurtosis(x2),kurtosis(x3),kurtosis(x4),kurtosis(x5),kurtosis(x6),kurtosis(x7


m <- data.frame(noms,Mean,Var,Skewness,Kurtosis)
print(m)
```

```
##      noms       Mean         Var   Skewness   Kurtosis
## 1     Gus  0.1087425  0.72736061  0.2557791 -0.1934165
## 2    Unif  0.5113430  0.07672447 -0.0954954 -1.1078875
## 3    Pois  0.9700000  1.10010101  1.2029706  1.4047146
## 4     Exp  0.9669590  0.65984079  1.1072391  0.8239690
## 5      X2  0.8367383  0.95480864  2.3496152  7.5459274
## 6   Binom 49.5200000 25.90868687  0.2677199  0.4852014
## 7  Cauchy -0.4856826 39.07056759 -1.8903425 15.0152373
## 8 Inconnu -0.1143862  8.34430091  0.1883106  0.0180116
```

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.

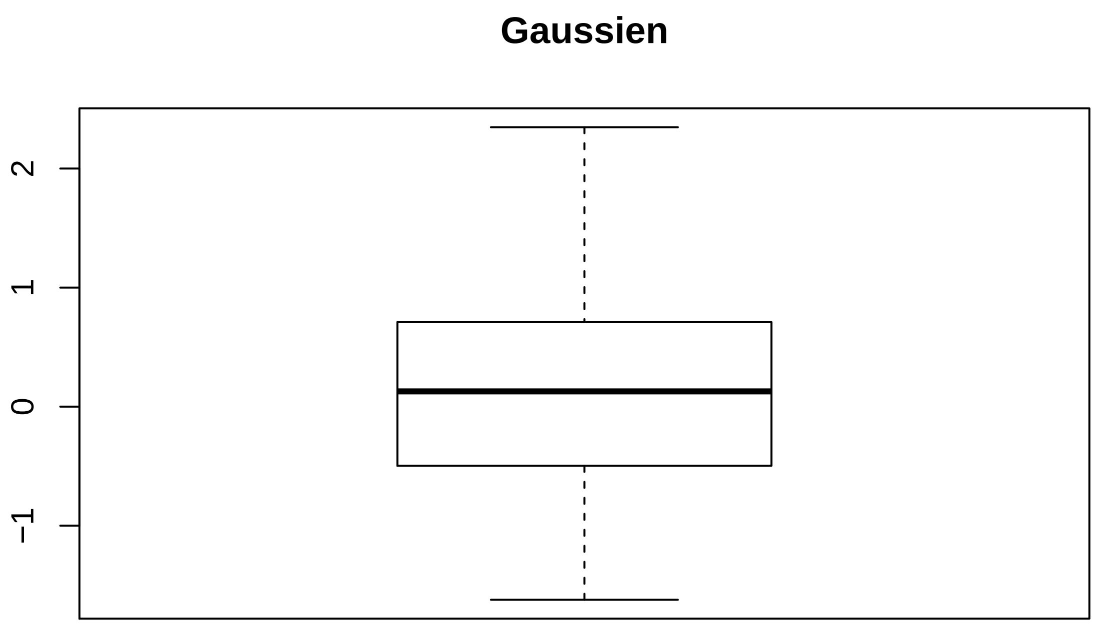
- Quantiles sp?ciaux:
  Q_1: La valeur en dessous de la quelle on a le quart des valeurs de notre echantillion.
  Q_2: La valeur en dessous de la quelle on a la moiti? des valeurs de notre echantillion, aussi connue sous le nom de m?diane.
  Q_3: La valeur en dessous de la quelle on a les trois-quarts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

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.

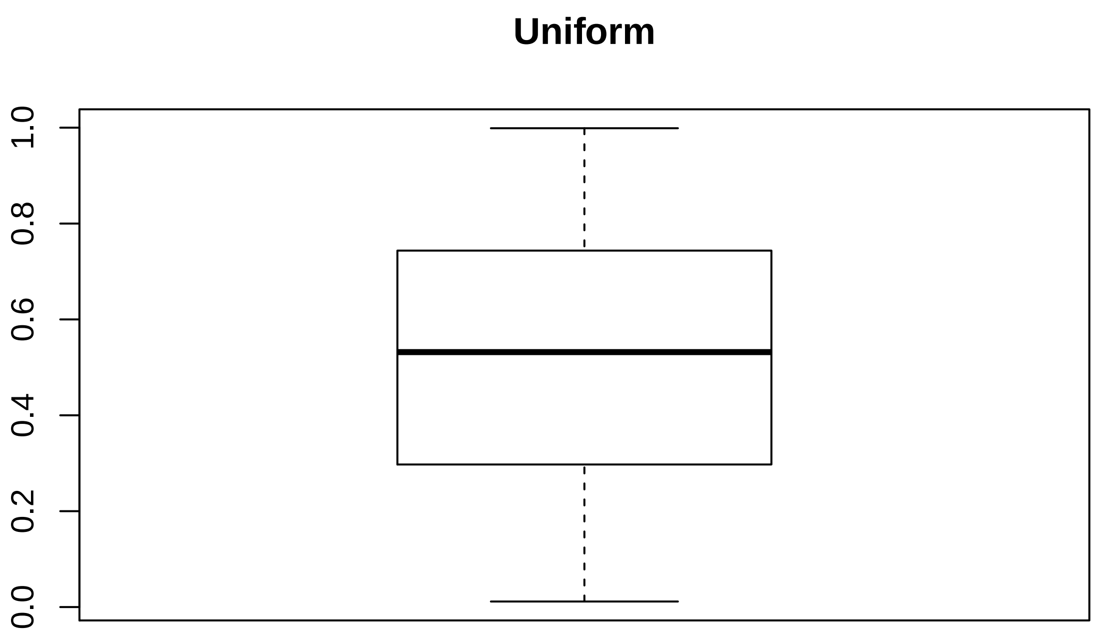
```r
#Quantiles et Boxplot
table <- read.table("data_8_lois.txt",header = TRUE)

x1 <- table[1]
x2 <- table[2]
x3 <- table[3]
x4 <- table[4]
x5 <- table[5]
x6 <- table[6]
x7 <- table[7]
x8 <- table[8]
boxplot(unlist(x1),main ="Gaussien")
```



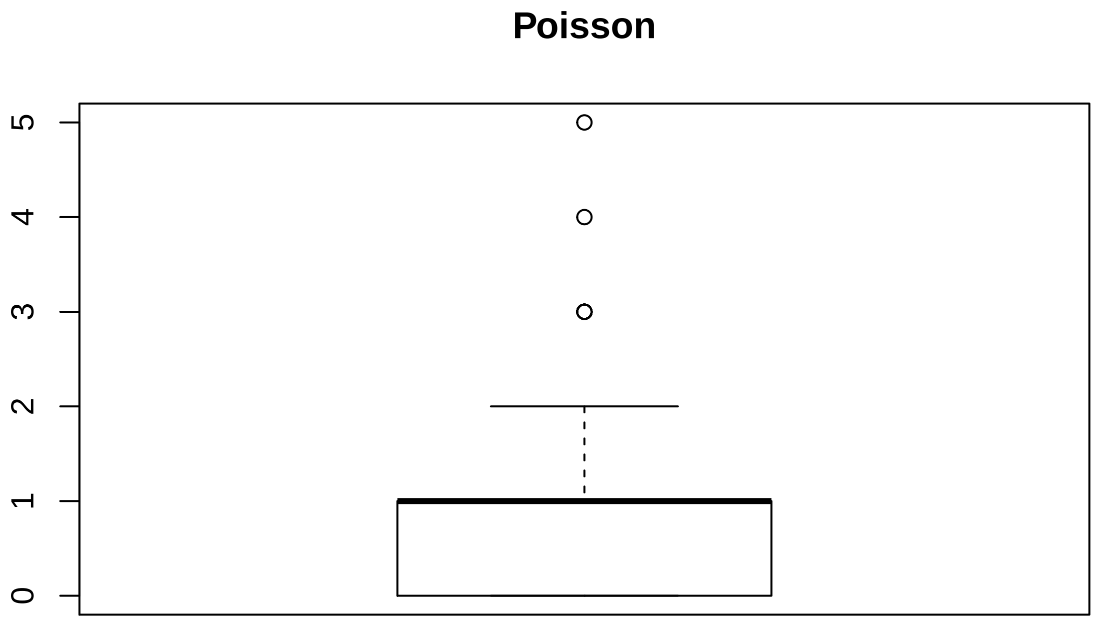

```
c1<-quantile(unlist(x1),c(0.25,0.5,0.75))

boxplot(unlist(x2),main ="Uniform")
```



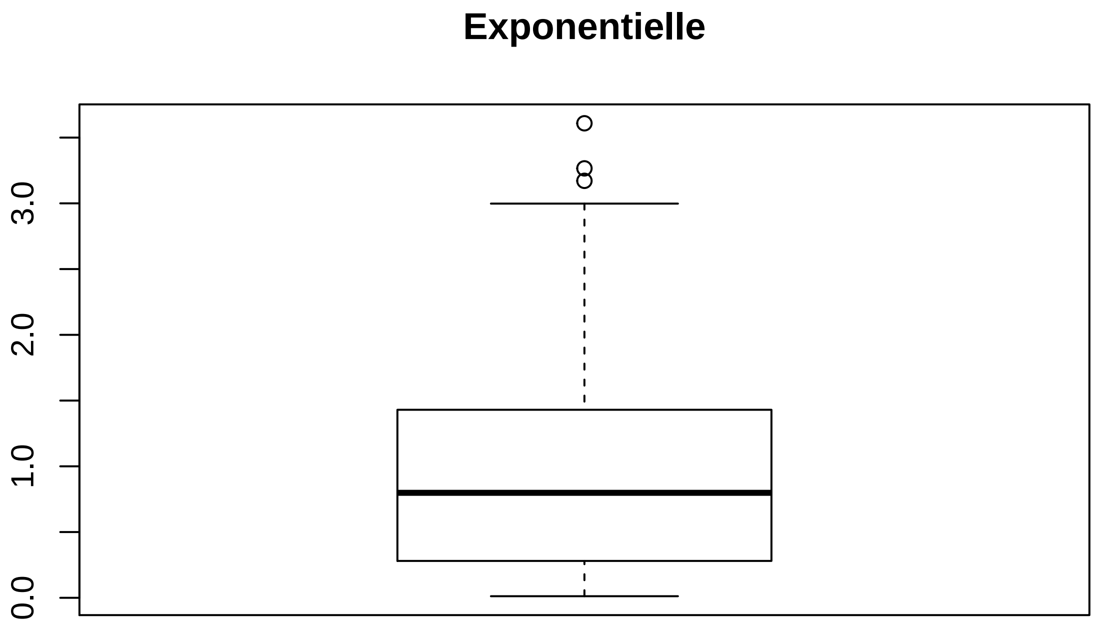

```
c2<-quantile(unlist(x2),c(0.25,0.5,0.75))

boxplot(unlist(x3),main ="Poisson")
```



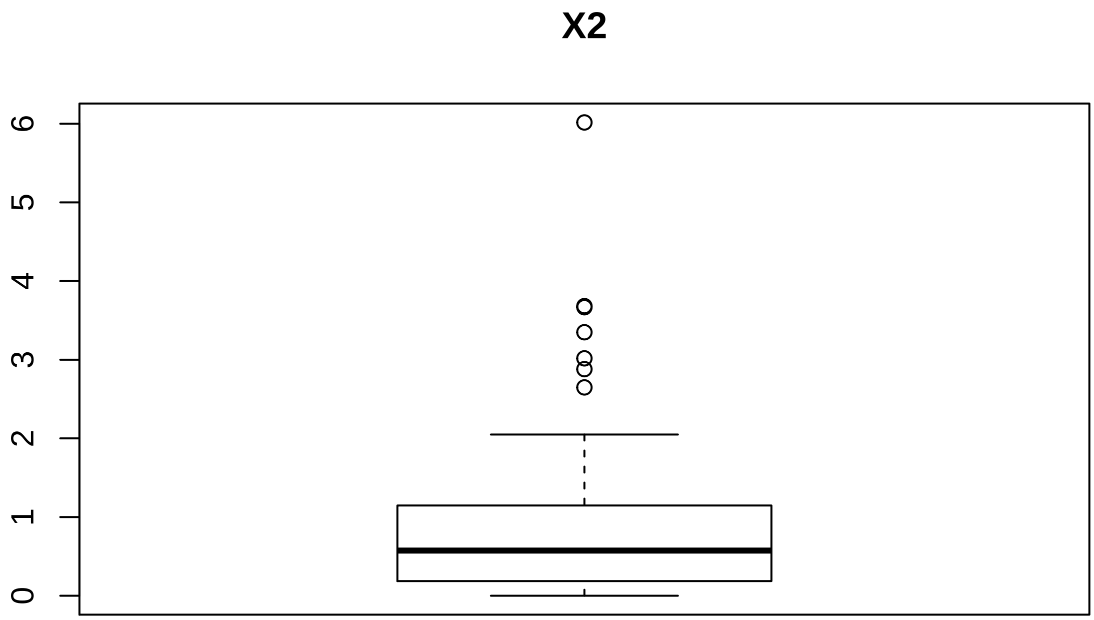

```
c3<-quantile(unlist(x3),c(0.25,0.5,0.75))

boxplot(unlist(x4),main ="Exponentielle")
```



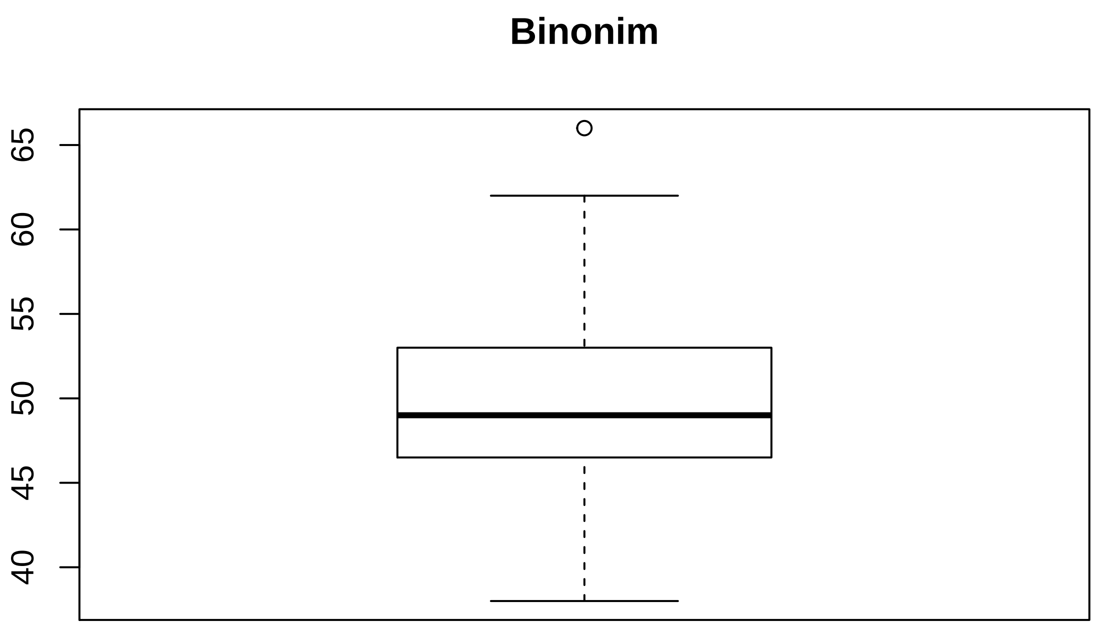

```
c4<-quantile(unlist(x4),c(0.25,0.5,0.75))

boxplot(unlist(x5),main ="X2")
```



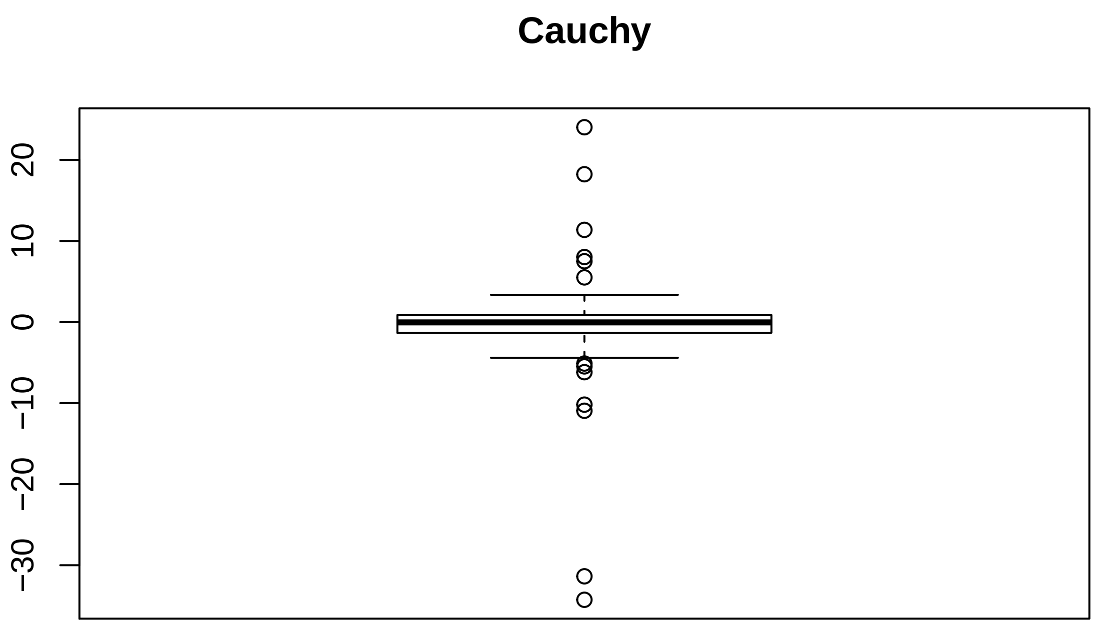

```r
c5<-quantile(unlist(x5),c(0.25,0.5,0.75))

boxplot(unlist(x6),main ="Binonim")
```



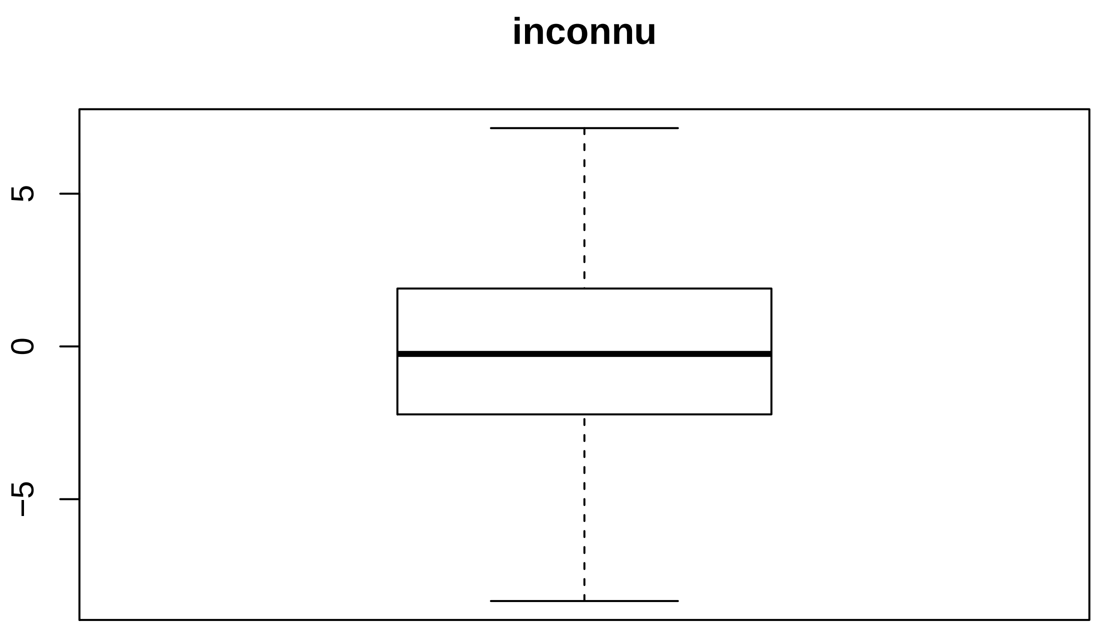

```r
c6<-quantile(unlist(x6),c(0.25,0.5,0.75))

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




```r
c7<-quantile(unlist(x7),c(0.25,0.5,0.75))

boxplot(unlist(x8),main="inconnu")
```




```r
c8<-quantile(unlist(x8),c(0.25,0.5,0.75))
q<-data.frame(Gaus=c1,Unif=c2,Pois=c3,Exp=c4,X2=c5,Binom=c6,Cauchy=c7,Inconnu=c8)
rownames(q)<- c("Q1","Q2","Q3")
Q <-t(q)
Table_info_loi <-cbind(m,Q)
print(Table_info_loi)
```

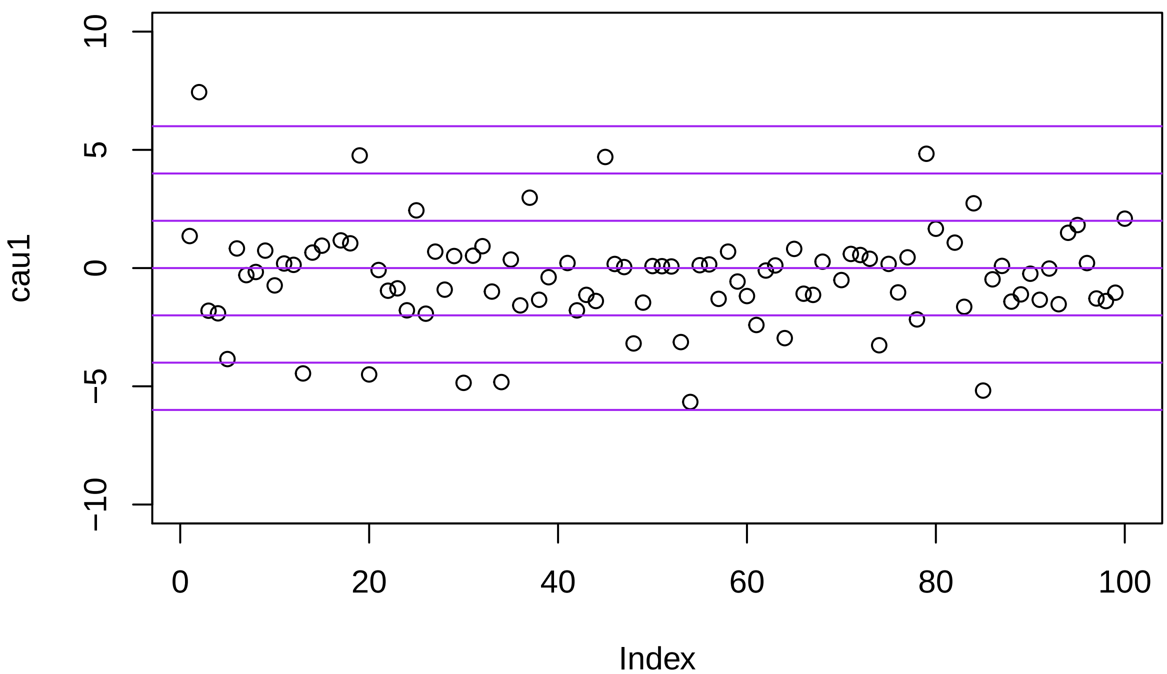
```
##           noms      Mean        Var   Skewness   Kurtosis         Q1
## Gaus       Gus 0.1087425 0.72736061  0.2557791 -0.1934165 -0.4953322
## Unif      Unif 0.5113430 0.07672447 -0.0954954 -1.1078875  0.3091710
## Pois      Pois 0.9700000 1.10010101  1.2029706  1.4047146  0.0000000
```

```
## Exp         Exp  0.9669590  0.65984079  1.1072391  0.8239690  0.2818774
## X2           X2  0.8367383  0.95480864  2.3496152  7.5459274  0.1874066
## Binom     Binom 49.5200000 25.90868687  0.2677199  0.4852014 46.7500000
## Cauchy   Cauchy -0.4856826 39.07056759 -1.8903425 15.0152373 -1.2818906
## Inconnu Inconnu -0.1143862  8.34430091  0.1883106  0.0180116 -2.1959296
##                   Q2         Q3
## Gaus      0.12820361  0.6989595
## Unif      0.53182033  0.7415278
## Pois      1.00000000  1.0000000
## Exp       0.79845034  1.4256436
## X2        0.57375450  1.1432138
## Binom    49.00000000 53.0000000
## Cauchy   -0.04141784  0.8435802
## Inconnu -0.24532650  1.8832830
```

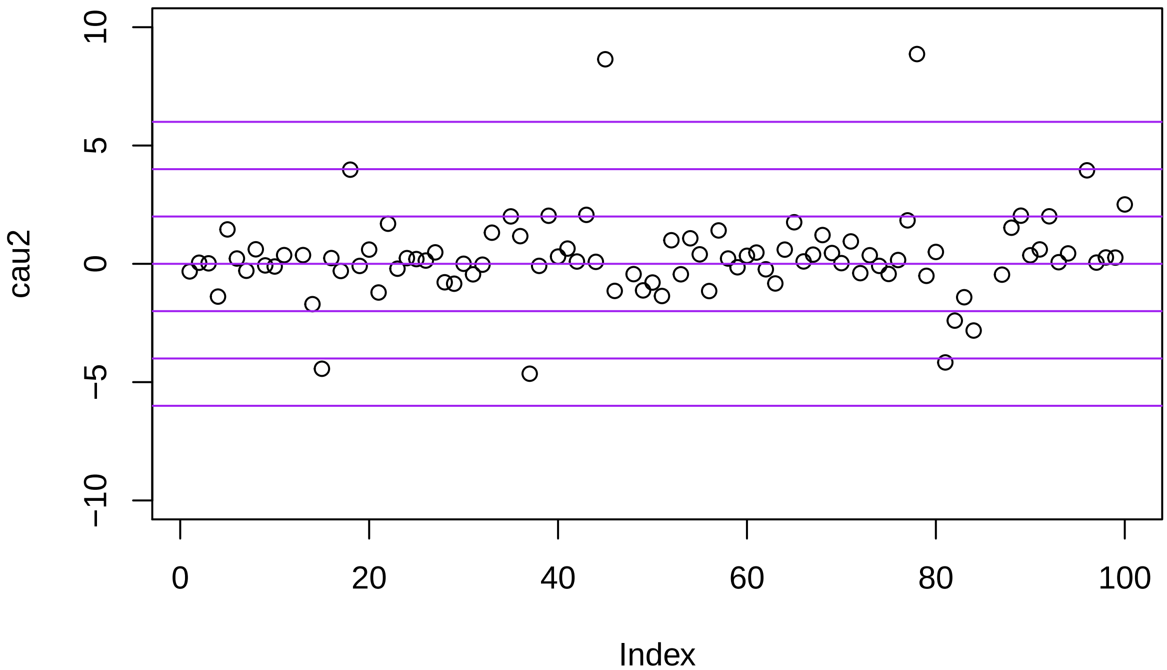
Interprétation visuelle

Générez 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?

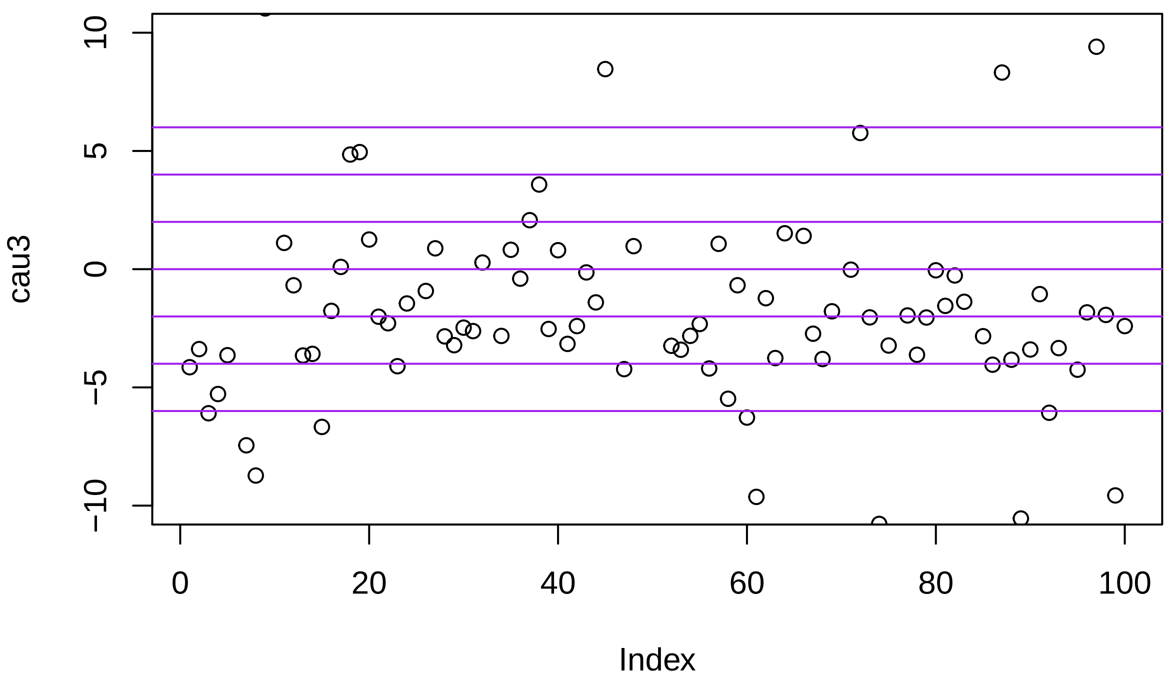
```
#Interprétation visuelle
cau1 <- rcauchy(n=100,location=0,scale=1)
cau2 <- rcauchy(n=100,location=0,scale=0.5)
cau3 <- rcauchy(n=100,location=-2,scale=2)
x <- seq(-5,4.9,0.1)
plot(cau1,ylim=c(-10,10))
abline(h=c(0,2,-2,4,-4,6,-6),col= ("purple"))
```



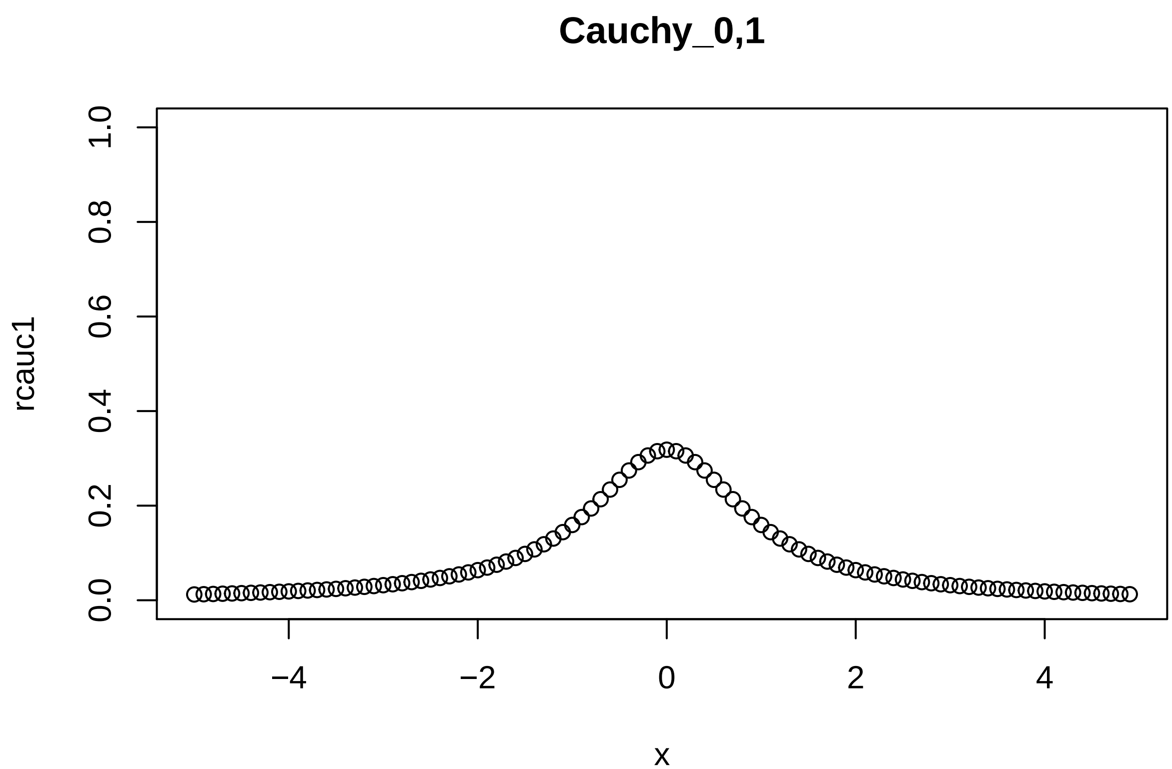

```
plot(cau2,ylim=c(-10,10))
abline(h=c(0,2,-2,4,-4,6,-6),col= ("purple"))
```



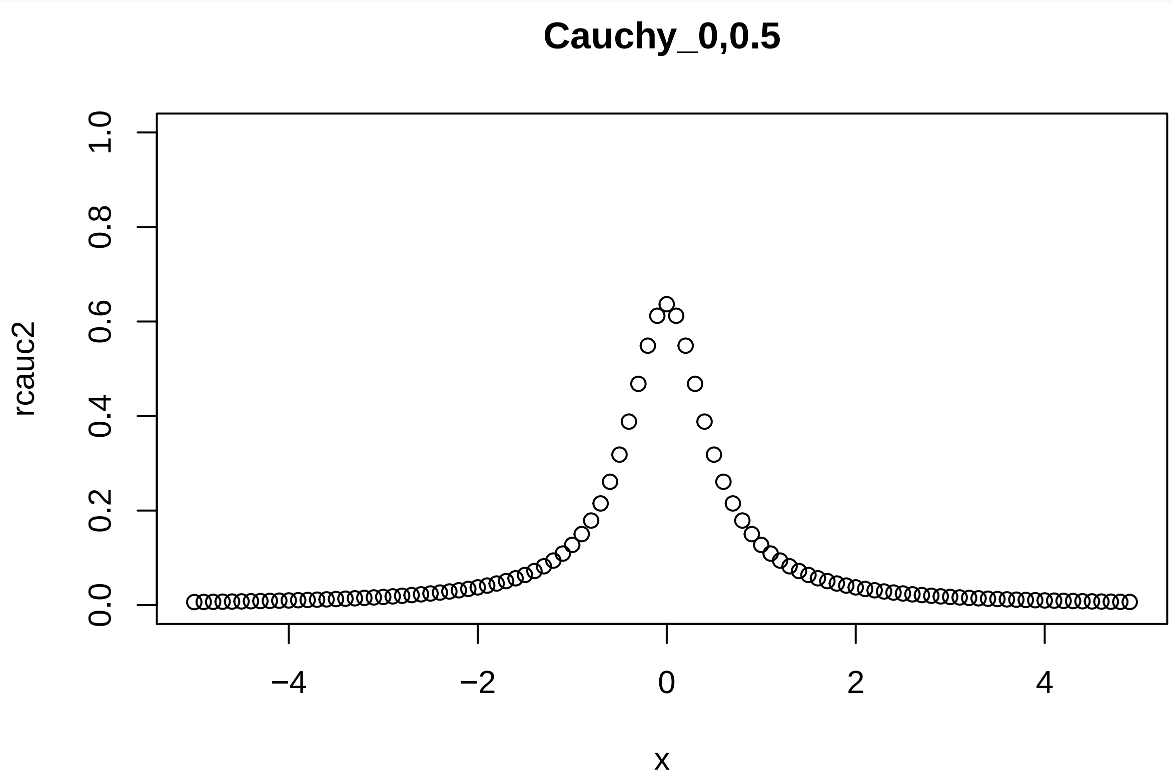

```
plot(cau3,ylim=c(-10,10))
abline(h=c(0,2,-2,4,-4,6,-6),col= ("purple"))
```



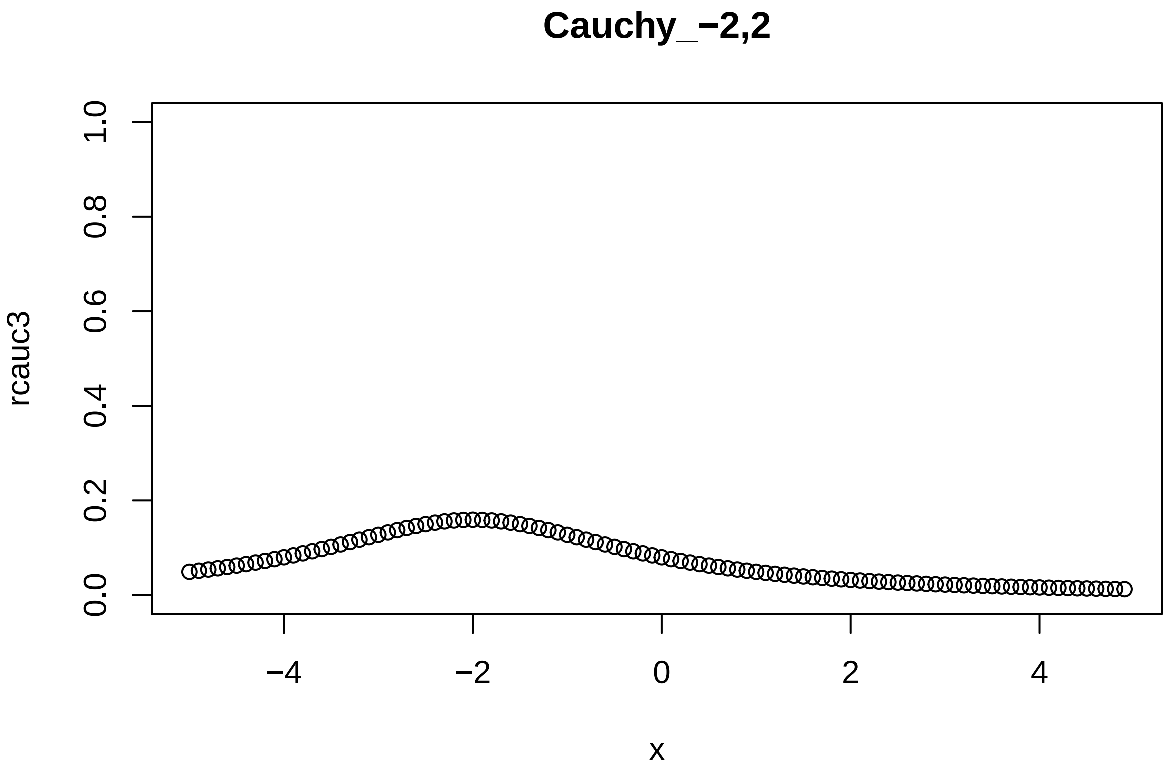

```
rcauc1 <-dcauchy(x,location = 0,scale=1)
plot(x,rcauc1,ylim = c(0,1),main ="Cauchy_0,1")
```



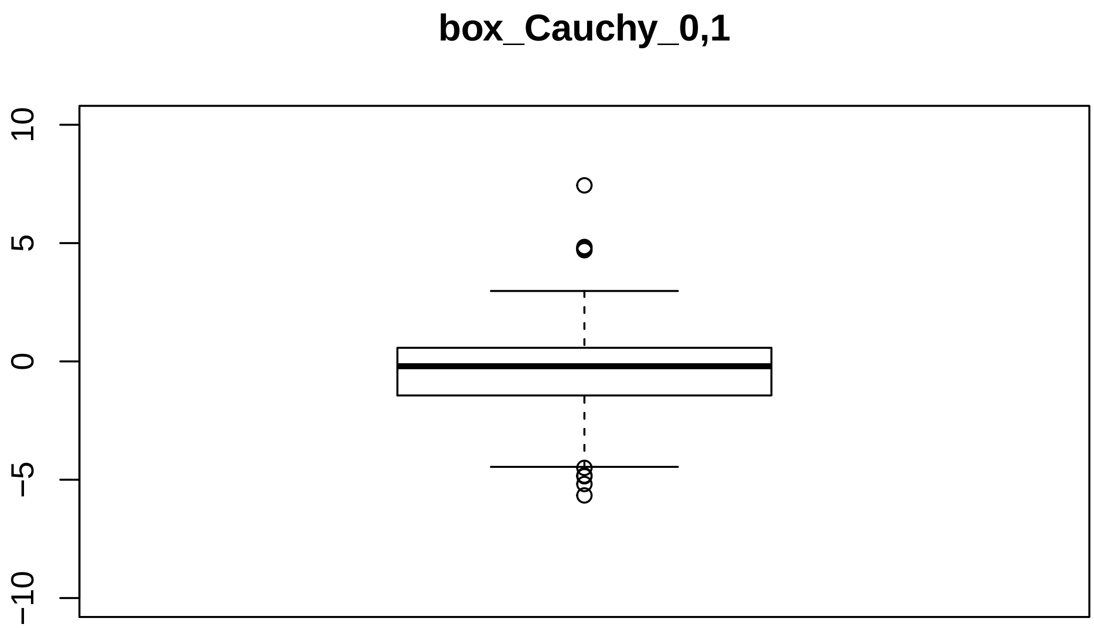

```r
rcauc2 <-dcauchy(x,location = 0,scale=0.5)
plot(x,rcauc2,ylim = c(0,1),main ="Cauchy_0,0.5")
```



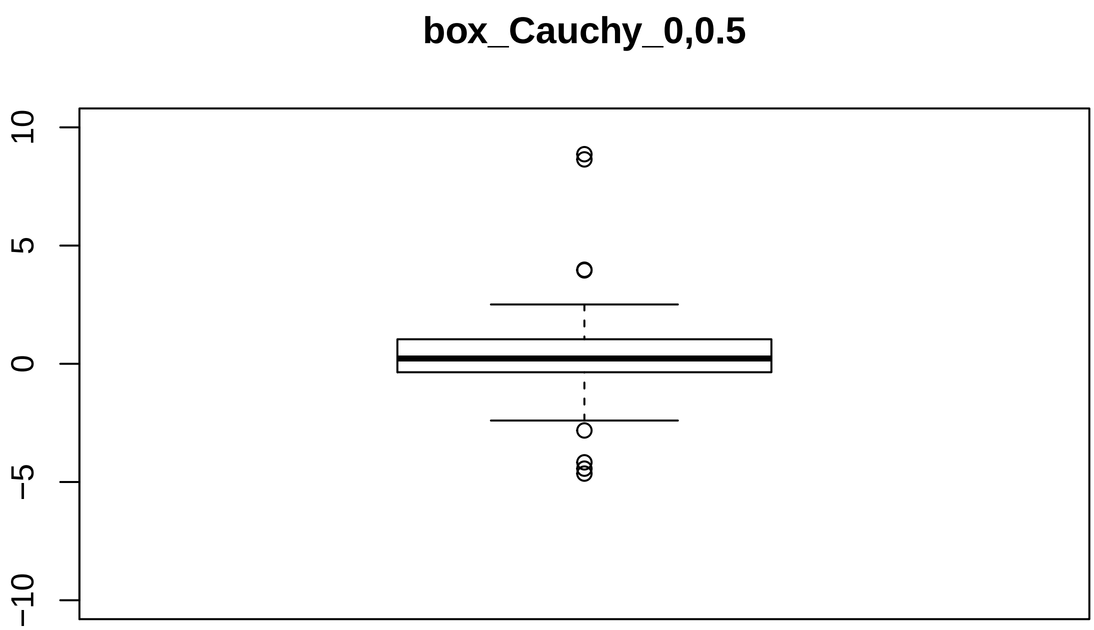

```r
rcauc3 <-dcauchy(x,location = -2,scale=2)
plot(x,rcauc3,ylim = c(0,1),main ="Cauchy_-2,2")
```



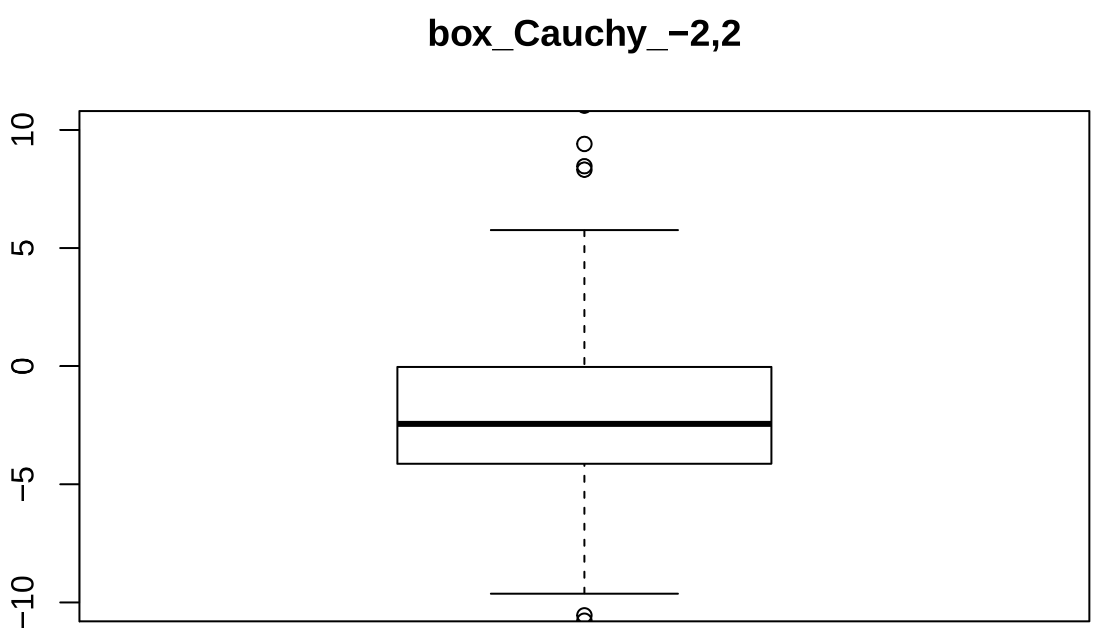

```r
boxplot(cau1,ylim=c(-10,10),main="box_Cauchy_0,1")
```




```r
boxplot(cau2,ylim=c(-10,10),main="box_Cauchy_0,0.5")
```




```
boxplot(cau3,ylim=c(-10,10),main="box_Cauchy_-2,2")
```




```
library(fBasics)
noms = c("cau1 ", "cau2 ", "cau3 ")
moyenne <- c(mean(cau1), mean(cau2), mean(cau3))
variance<- c(var(cau1), var(cau2), var(cau3))
Skewness <- c(skewness(cau1), skewness(cau2), skewness(cau3))
Kurtosis<- c(kurtosis(cau1), kurtosis(cau2), kurtosis(cau3))
f <- data.frame(noms,moyenne, variance, Skewness, Kurtosis)
print(f)
```

```
##    noms    moyenne  variance  Skewness Kurtosis
## 1 cau1  -0.9118985  23.42896 -3.510952 21.92019
## 2 cau2   3.6057151 503.30601  6.655213 49.13554
## 3 cau3  -2.9407181 336.16021  2.610133 29.36037
```

```
#j'ai remarqué que le degrée de dispersion des points de #distribution Cauchy sont tres élevés,surtout p
#les distribution qui ont un petite sacle .Par contre,
# plus le sacle est petite, plus la distribution Cauchy est #centrée sur 0, c'est à dire que sa Var est
```