TP1 stats

Romain PEREIRA

19 février 2018

Se rendre dans le dossier de travail

```
# setwd("/home/rpereira/ENSIIE/UE/S2/R/TP1")
```

Sauvegarder des données vers un fichier ".txt" ou ".csv"

```
n <- 40
df <- data.frame("Gaussienne" = rnorm(n),     "Uniforme"      = runif(n),
                 "Poisson"    = rpois(n, 1),  "Exponentielle" = rexp(n),
                 "Chi"        = rchisq(n, 1), "Binomiale"     = rbinom(n, 1, 0.5),
                 "Cauchy"     = rcauchy(n))
write.csv(df, file="./samples_40.csv")
# ou:
write.table(df, file="./samples_40.txt")
```

Charger des données depuis un fichier ".txt" ou ".csv"

```
df <- read.csv(file="./samples_40.csv", header=TRUE)
df["Gaussienne"]
```

```
##     Gaussienne
## 1   0.92516988
## 2   1.56917631
## 3   0.87429476
## 4   0.54757850
## 5  -1.35467393
## 6   0.75491835
## 7  -0.05464900
## 8  -0.62841210
## 9   0.47232467
## 10 -0.29017758
## 11 -1.41378808
## 12 -1.29947146
## 13 -1.62944426
## 14 -0.15367028
## 15  1.01093759
## 16  0.16853841
## 17  1.47123861
## 18  0.91925260
## 19  0.55111711
## 20  1.64735128
## 21  0.19425018
## 22  0.37526860
```

```
## 23 -0.72492431
## 24  0.72815473
## 25 -0.43745971
## 26 -0.59658891
## 27 -1.15183725
## 28  0.18460406
## 29  0.73684238
## 30  0.36923130
## 31 -0.02779139
## 32  0.03164971
## 33 -0.19058030
## 34  0.39376050
## 35  0.18958280
## 36  1.72842985
## 37 -0.30397492
## 38 -0.92121557
## 39 -0.14213157
## 40  0.31732747
```

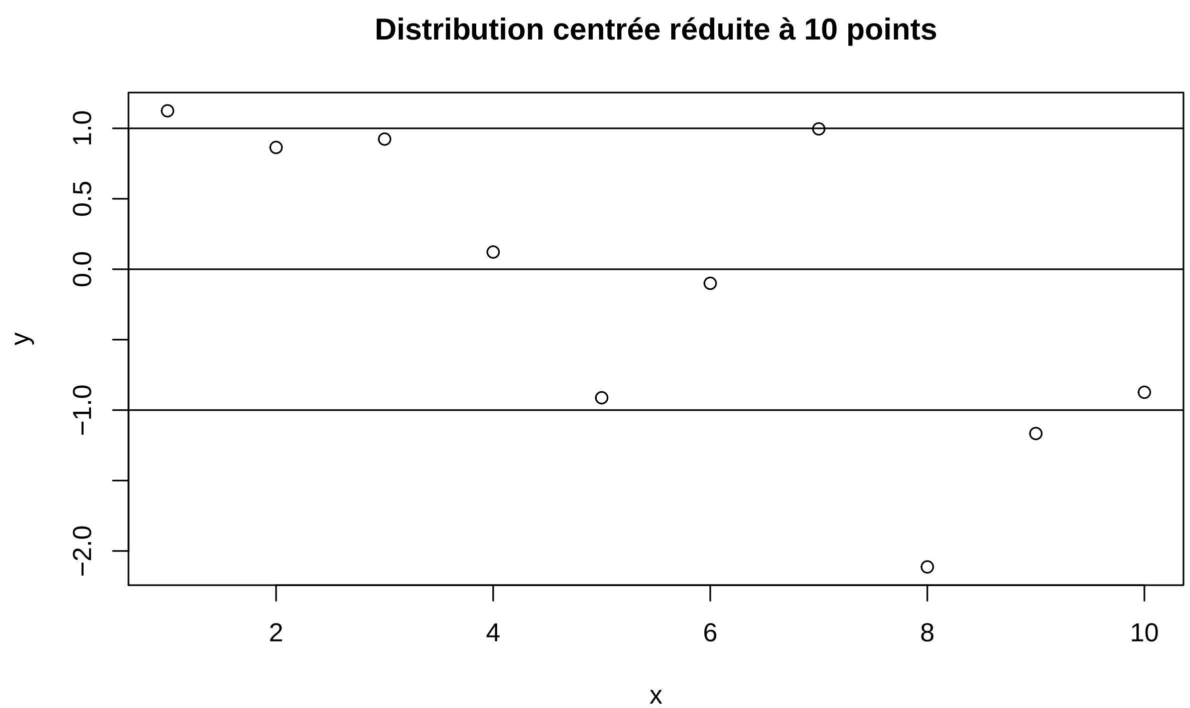
```r
# ou
df2 <- read.table(file="./samples_40.txt", header=TRUE)
df2["Gaussienne"]
```

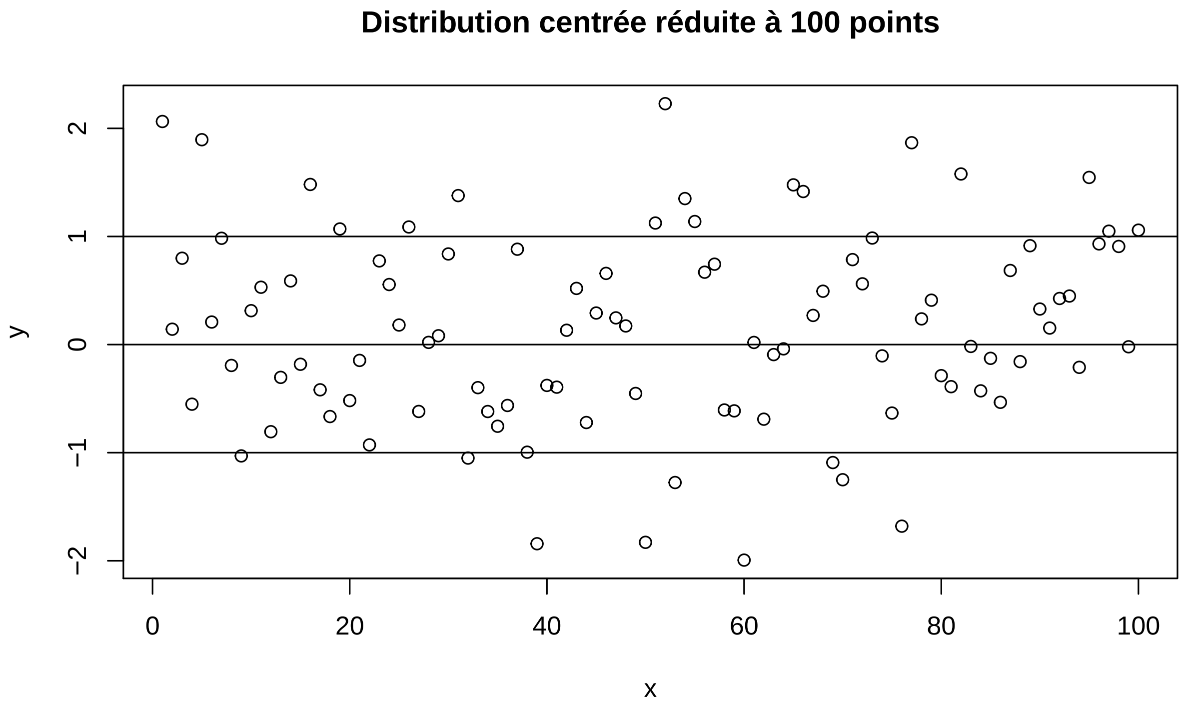
```
##     Gaussienne
## 1   0.92516988
## 2   1.56917631
## 3   0.87429476
## 4   0.54757850
## 5  -1.35467393
## 6   0.75491835
## 7  -0.05464900
## 8  -0.62841210
## 9   0.47232467
## 10 -0.29017758
## 11 -1.41378808
## 12 -1.29947146
## 13 -1.62944426
## 14 -0.15367028
## 15  1.01093759
## 16  0.16853841
## 17  1.47123861
## 18  0.91925260
## 19  0.55111711
## 20  1.64735128
## 21  0.19425018
## 22  0.37526860
## 23 -0.72492431
## 24  0.72815473
## 25 -0.43745971
## 26 -0.59658891
## 27 -1.15183725
## 28  0.18460406
## 29  0.73684238
## 30  0.36923130
## 31 -0.02779139
```

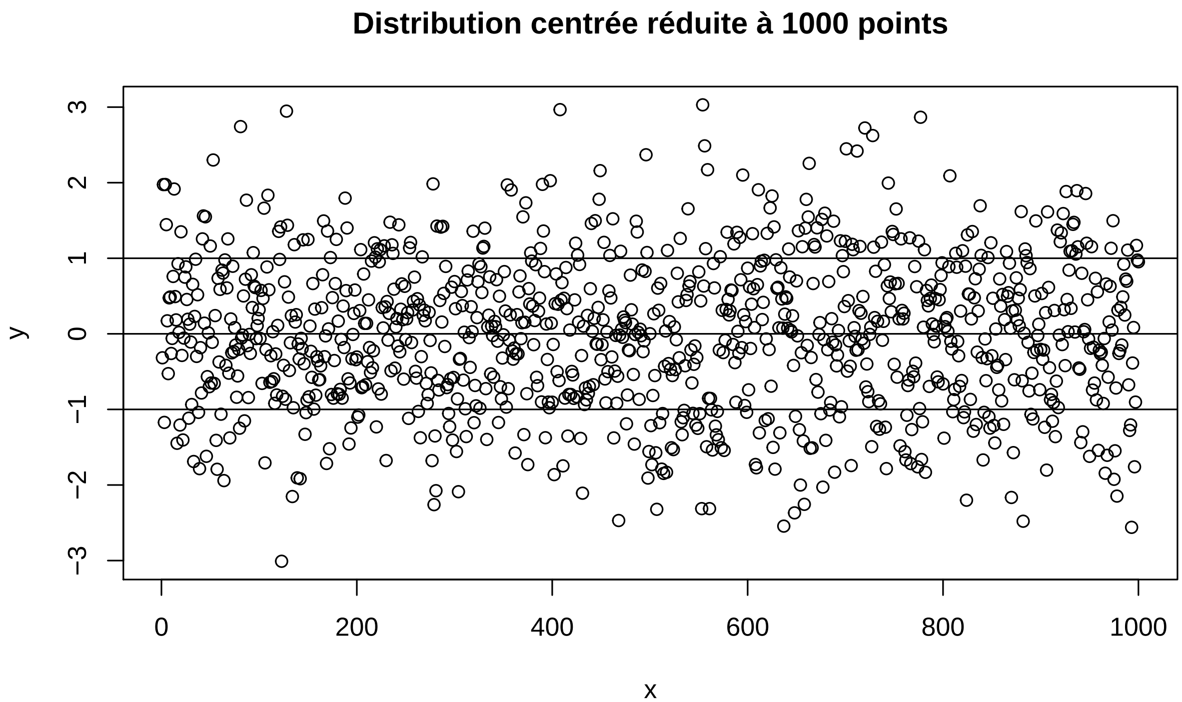
```
## 32  0.03164971
## 33 -0.19058030
## 34  0.39376050
## 35  0.18958280
## 36  1.72842985
## 37 -0.30397492
## 38 -0.92121557
## 39 -0.14213157
## 40  0.31732747
```

Tracer d’un échantillon de 10 points pour la loi normal N(0, 1)

```r
ns <- c(10, 100, 1000)
for (n in ns) {
  x <- 1:n
  y <- rnorm(n, 0, 1)
  plot(x, y, main=paste("Distribution centrée réduite à", n, "points"))
  abline(h=0)
  abline(h=-1)
  abline(h=1)
}
```





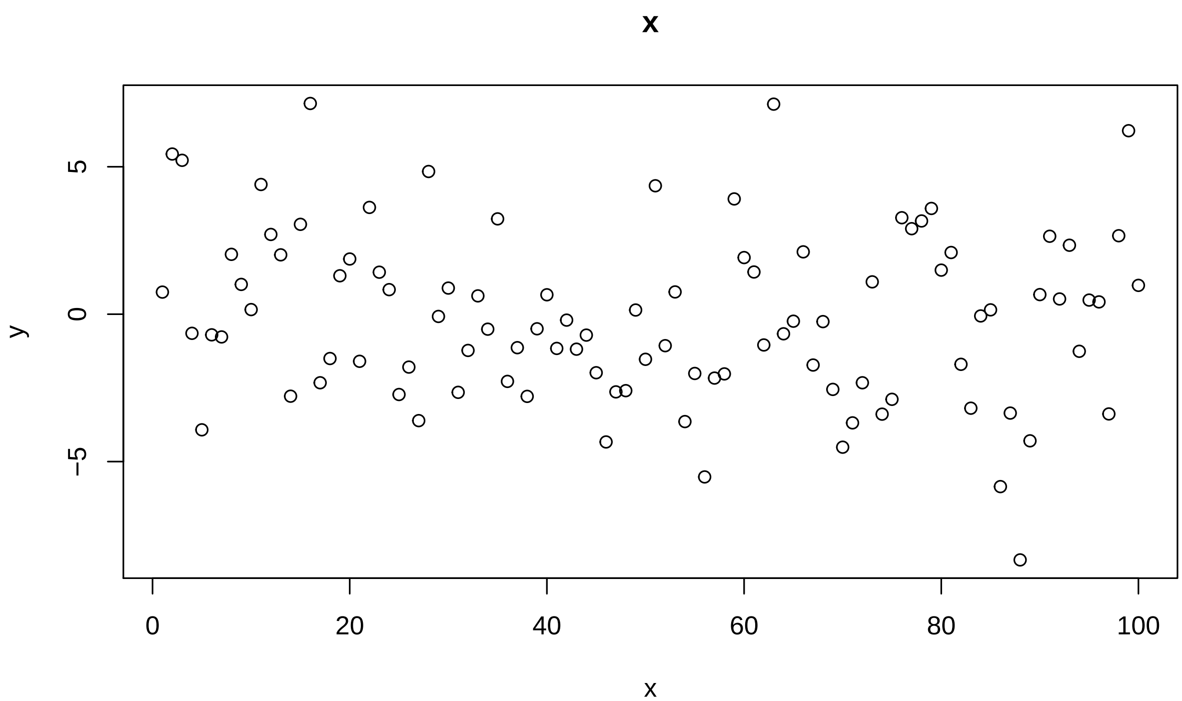


On remarque qu'il y a environ autant de valeurs positives que négatives, et que la répartition est d'autant plus dense que l'on se rapproche de l'axe y=0.

Je définie une fonction permettant de tracer un "data.frame", afin d'étudier la distribution qui nous est fournie.

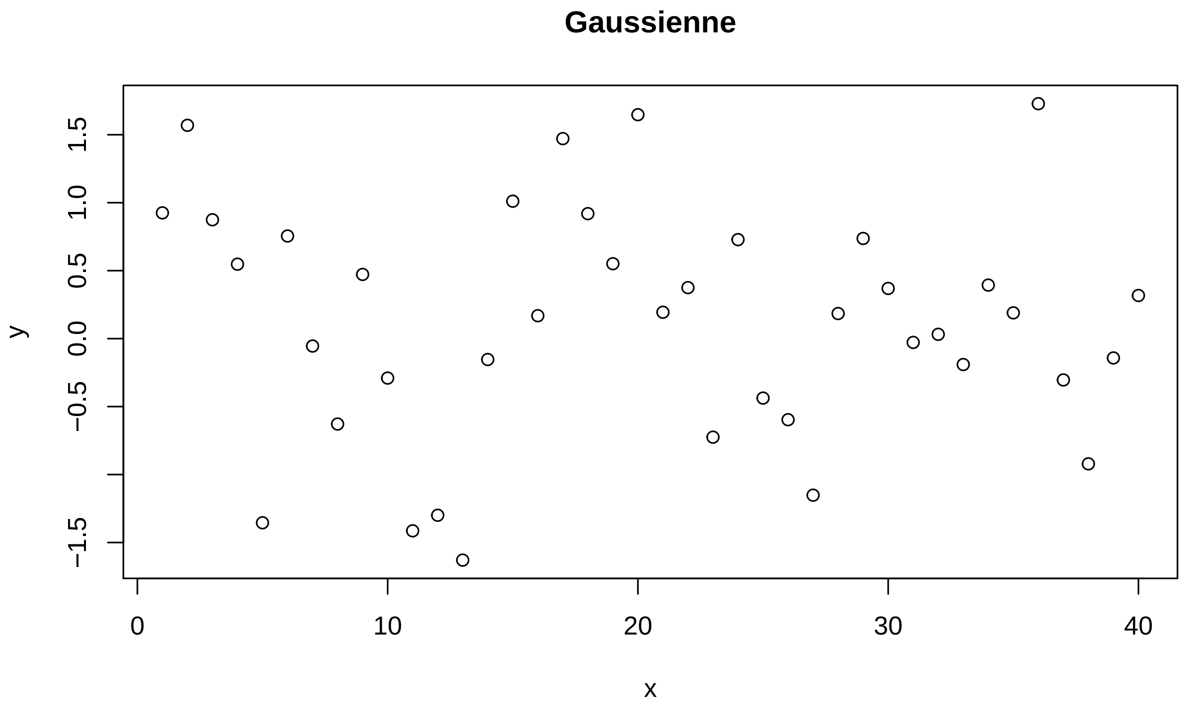
Traçons la distribution inconnue:

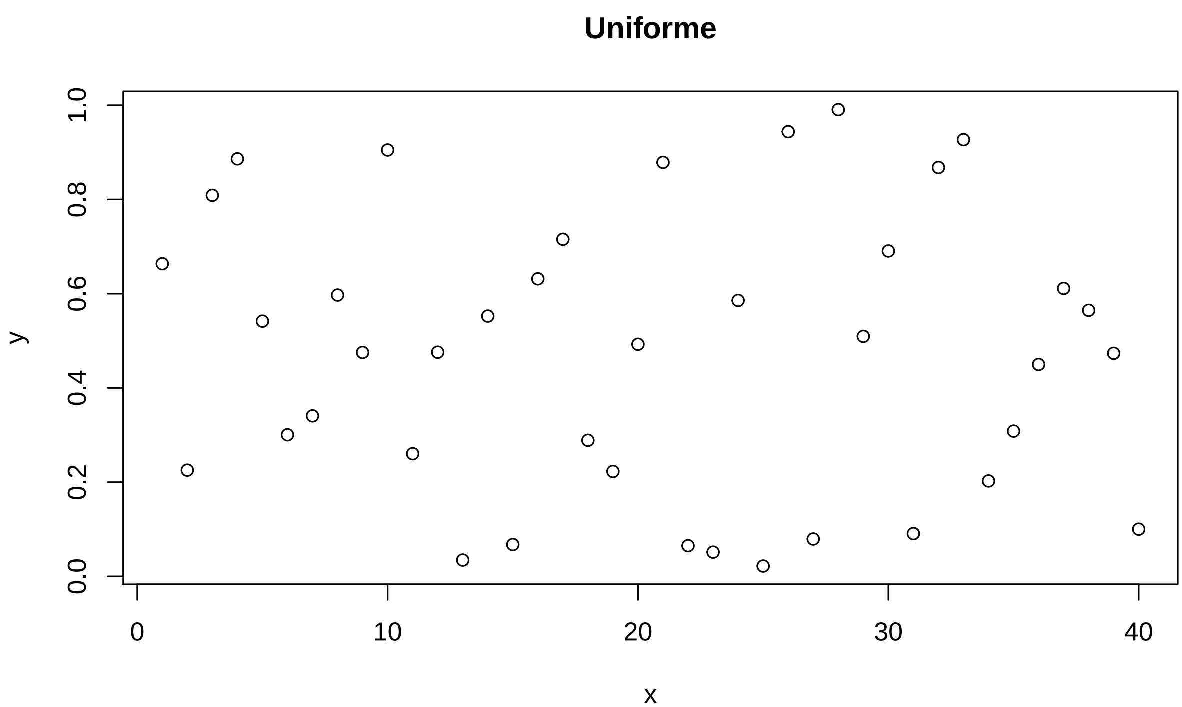
```
tracer <- function(data, xrow, yrow) {
  x <- unlist(data[xrow])
  y <- unlist(data[yrow])
  plot(x, y, main=yrow)
}

df_inconnu <- read.csv("./distribution_inconnue_1_100_realisations.csv")
tracer(df_inconnu, "X", "x");
```

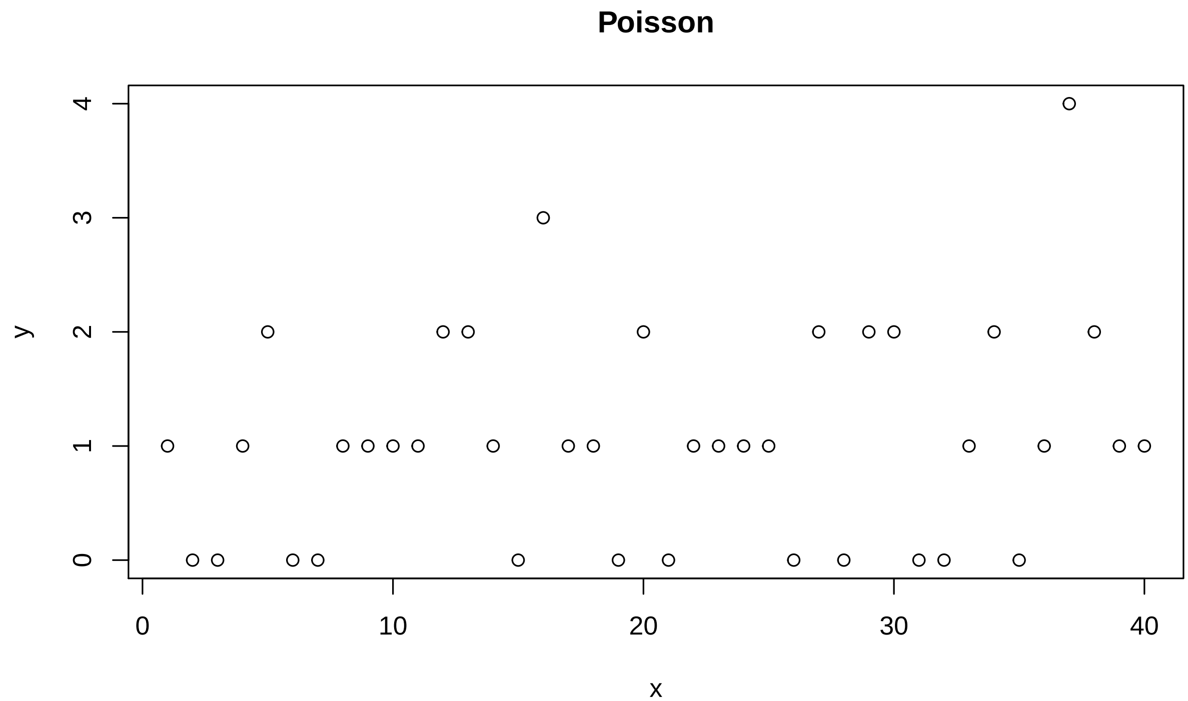


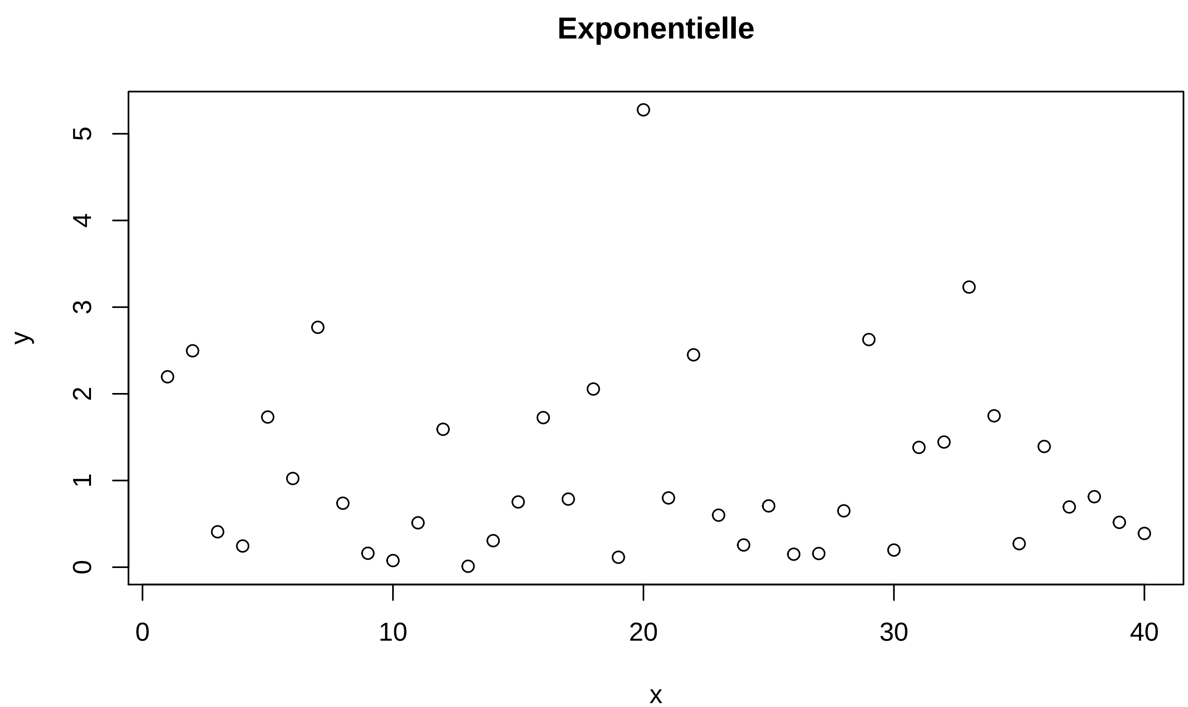
... afin de les comparer avec les distributions générés précdémment:

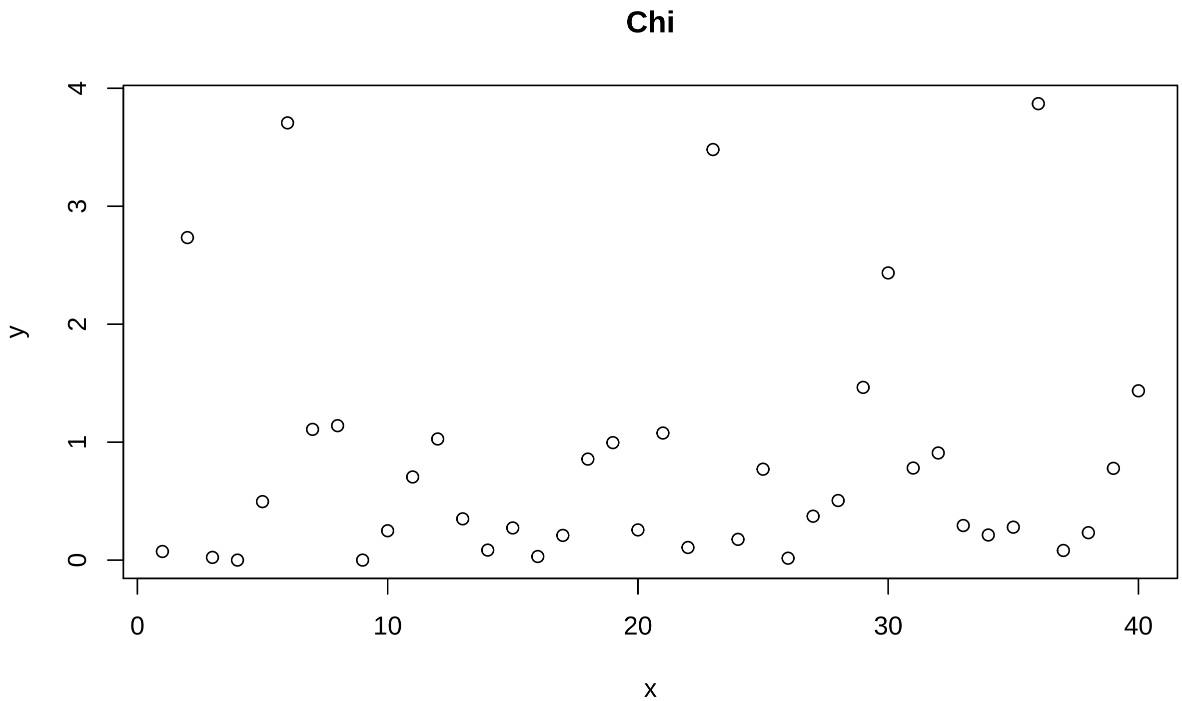
```
distributions <- c("Gaussienne", "Uniforme", "Poisson", "Exponentielle", "Chi", "Binomiale", "Cauchy");
for (distri in distributions) {
  tracer(df, "X", distri);
}
```

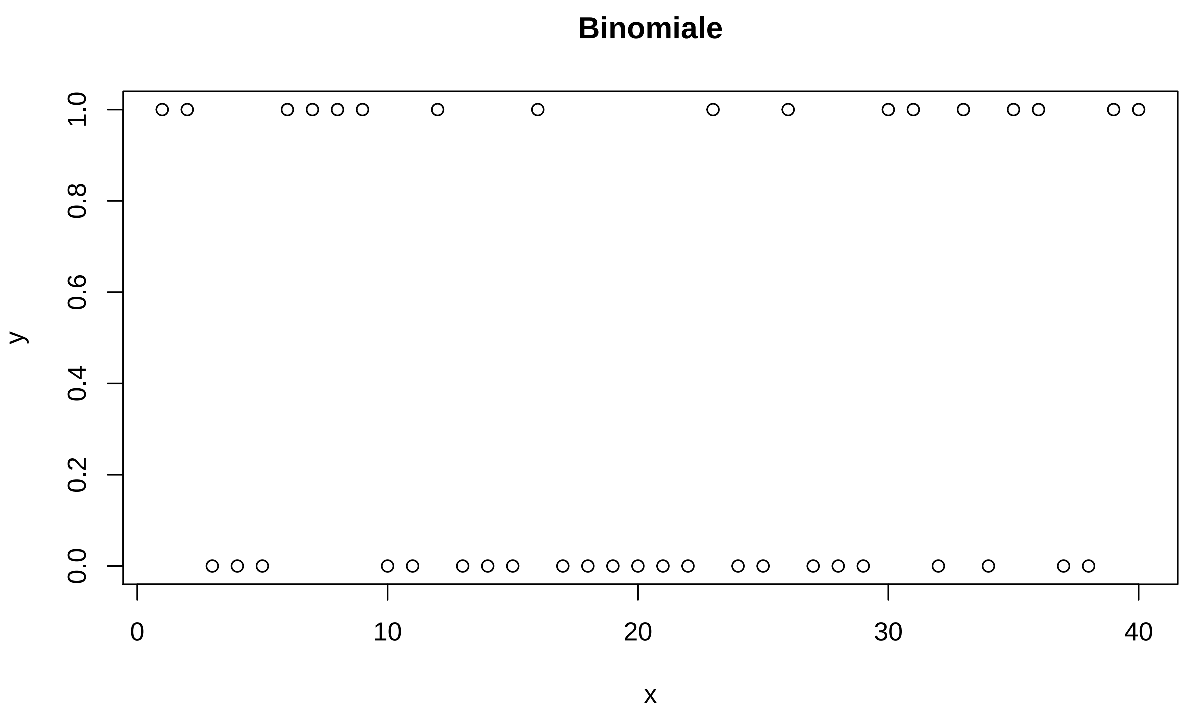

Gaussienne
1.5
1.0
0.5
0.0
−0.5
−1.5
y
0
10
20
30
40
x

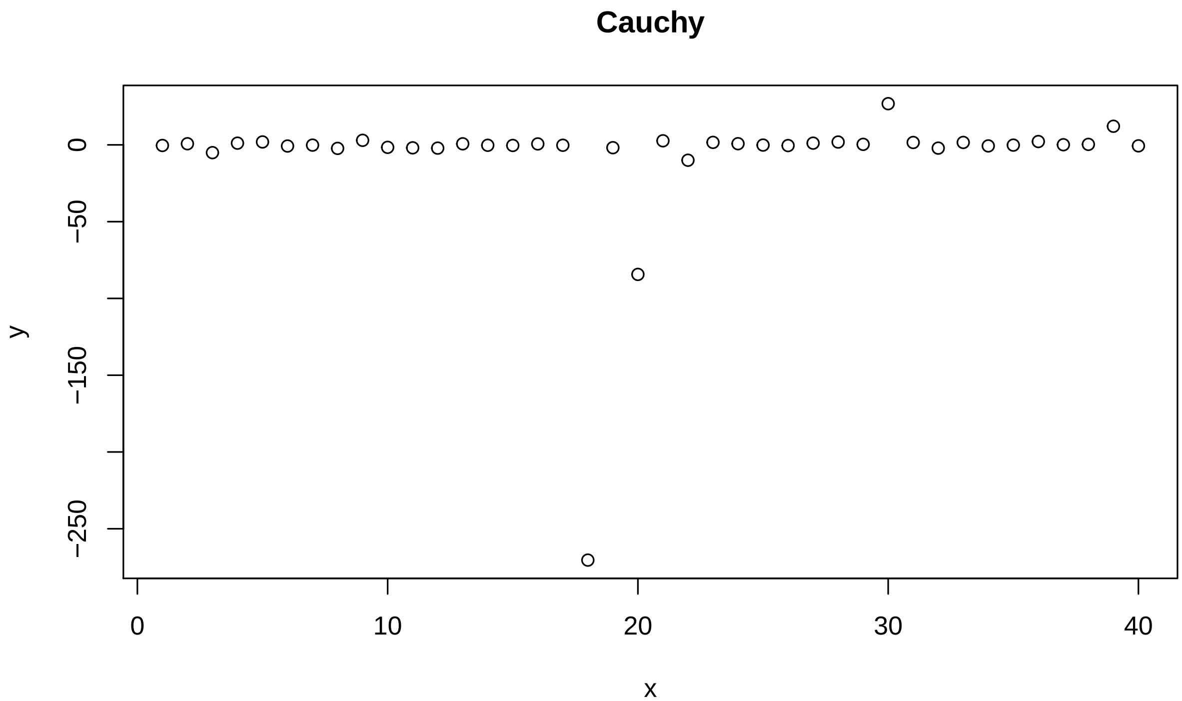

Uniforme
1.0
0.8
0.6
0.4
0.2
0.0
y
0
10
20
30
40
x


Poisson
y
x
0
1
2
3
4
0
10
20
30
40


Exponentielle
y
x
0
1
2
3
4
5
0
10
20
30
40


Chi
y
0
1
2
3
4
0
10
20
30
40
x

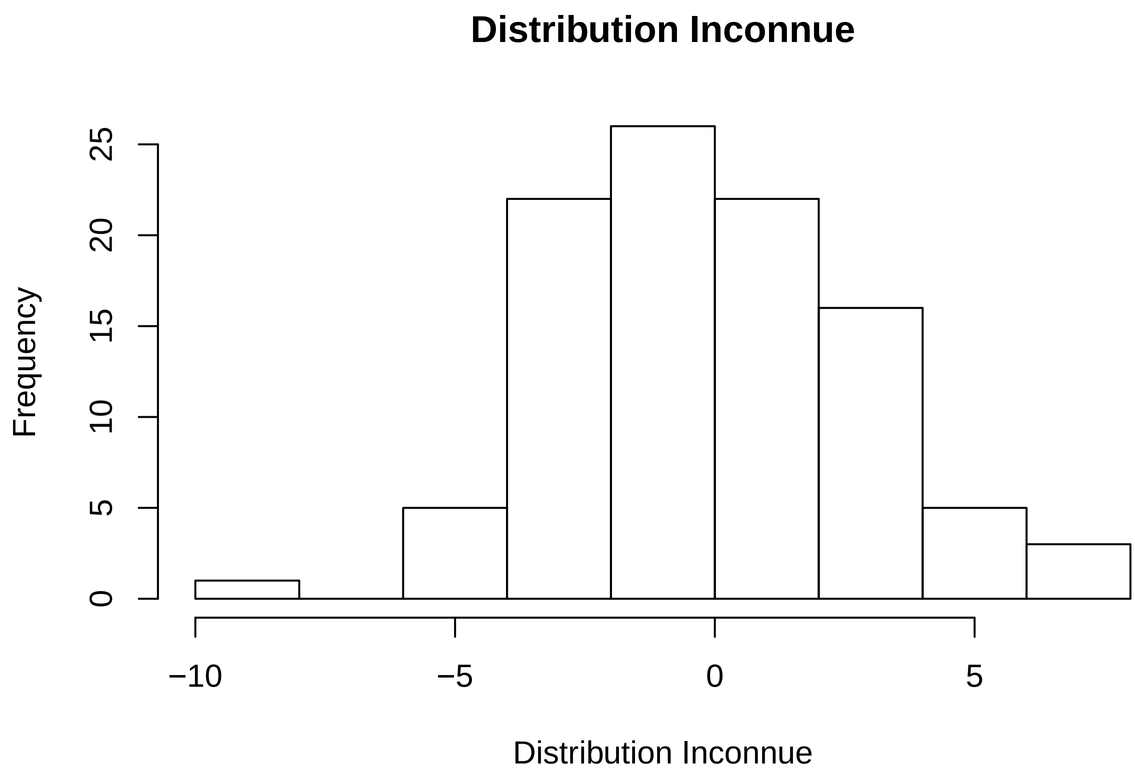

Binomiale
y
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
x


Cauchy
0
−50
−150
−250
y
0
10
20
30
40
x

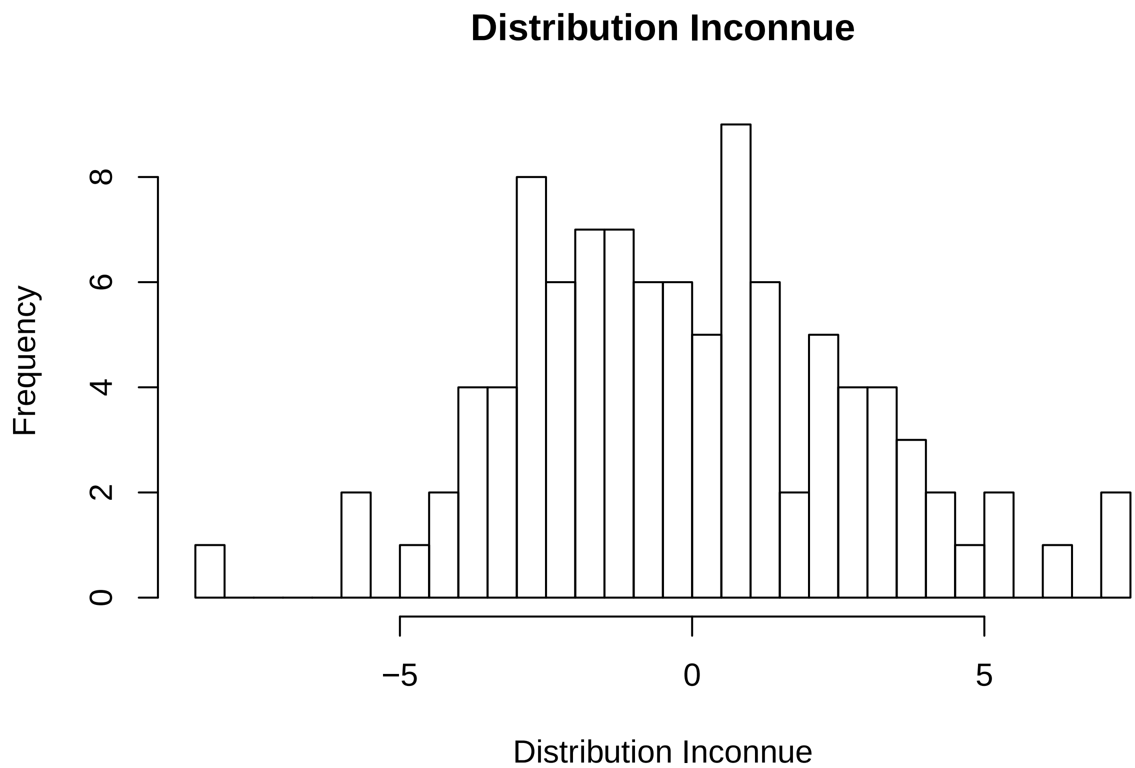
Histogramme

Je définis une fonction qui trace l'histogramme correspondant à la colonne 'row' du dataframe 'df'

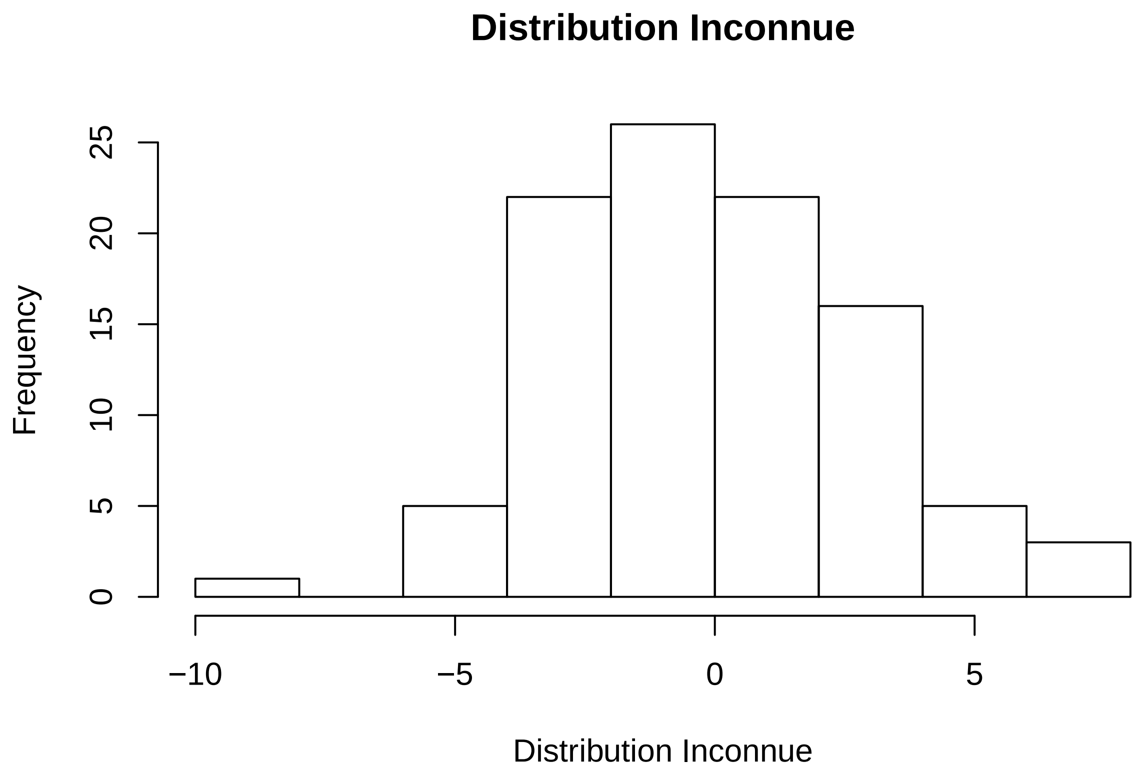
```
hist(unlist(df_inconnu["x"]), xlab="Distribution Inconnue", main="Distribution Inconnue")
```




```
hist(unlist(df_inconnu["x"]), xlab="Distribution Inconnue", breaks=50, main="Distribution Inconnue")
```

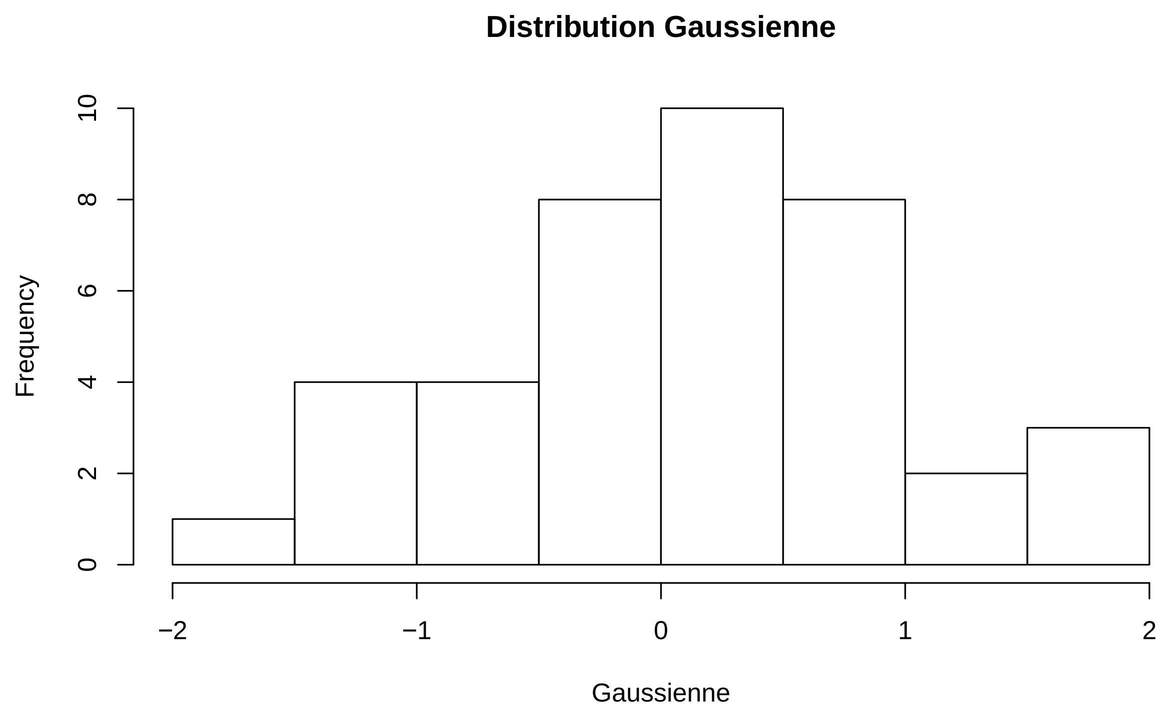


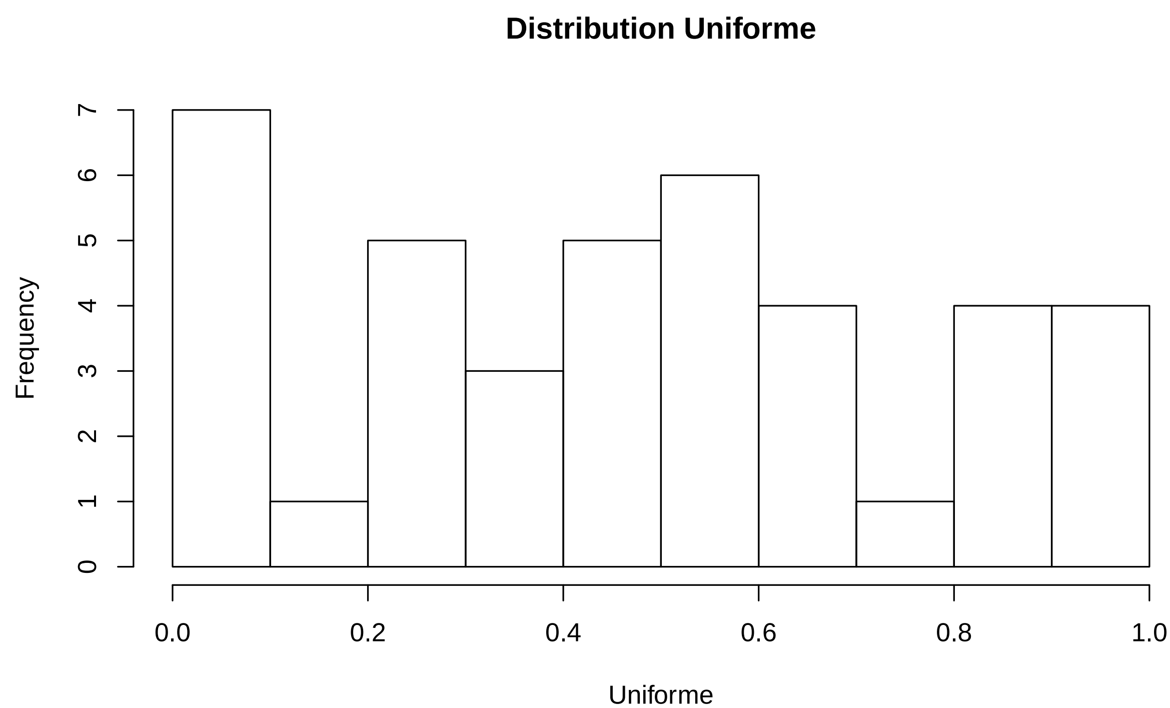

```
hist(unlist(df_inconnu["x"]), xlab="Distribution Inconnue", freq=TRUE, main="Distribution Inconnue")
```

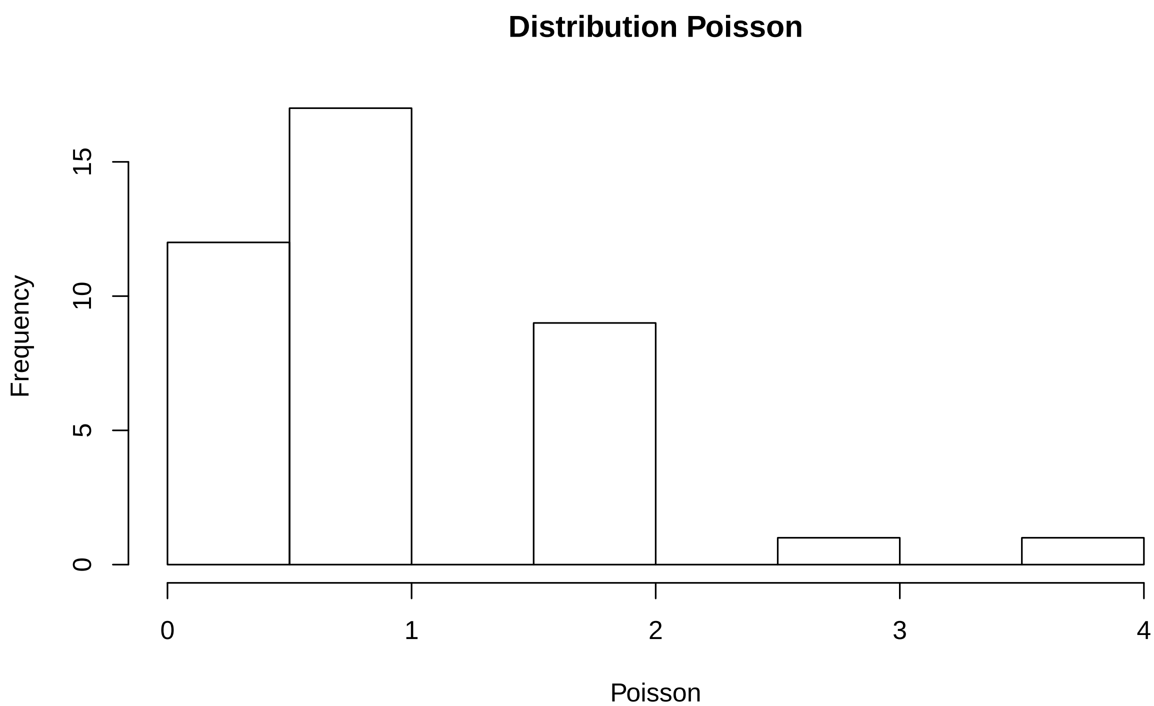


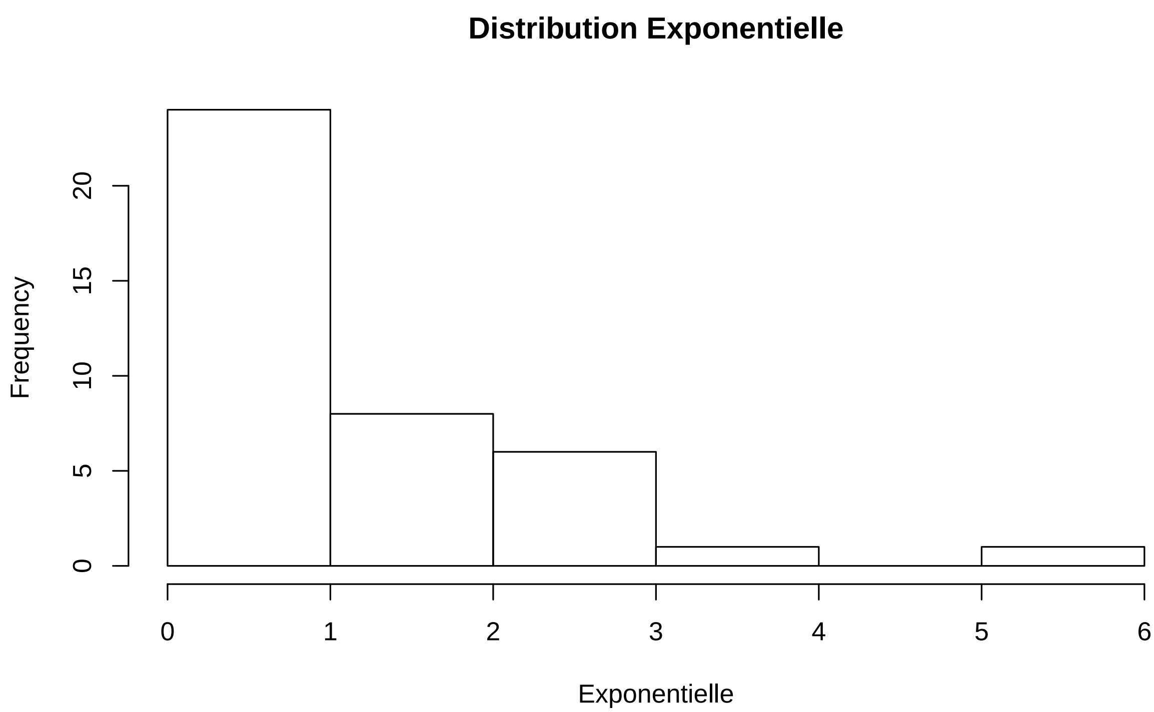
On remarque qu'il est probable que la distribution suit une loi normale $\mu = 0, \sigma^2 = 5$.

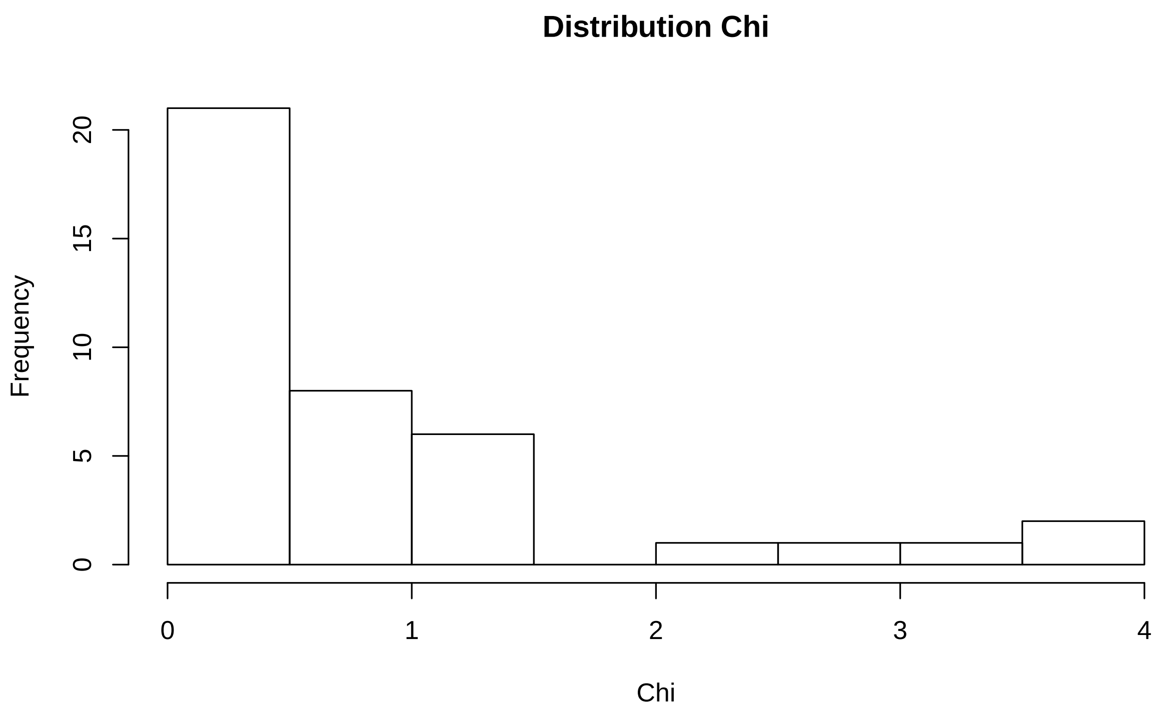
```r
for (distri in distributions) {
  hist(unlist(df[distri]), xlab=distri, main=paste("Distribution", distri))
}
```

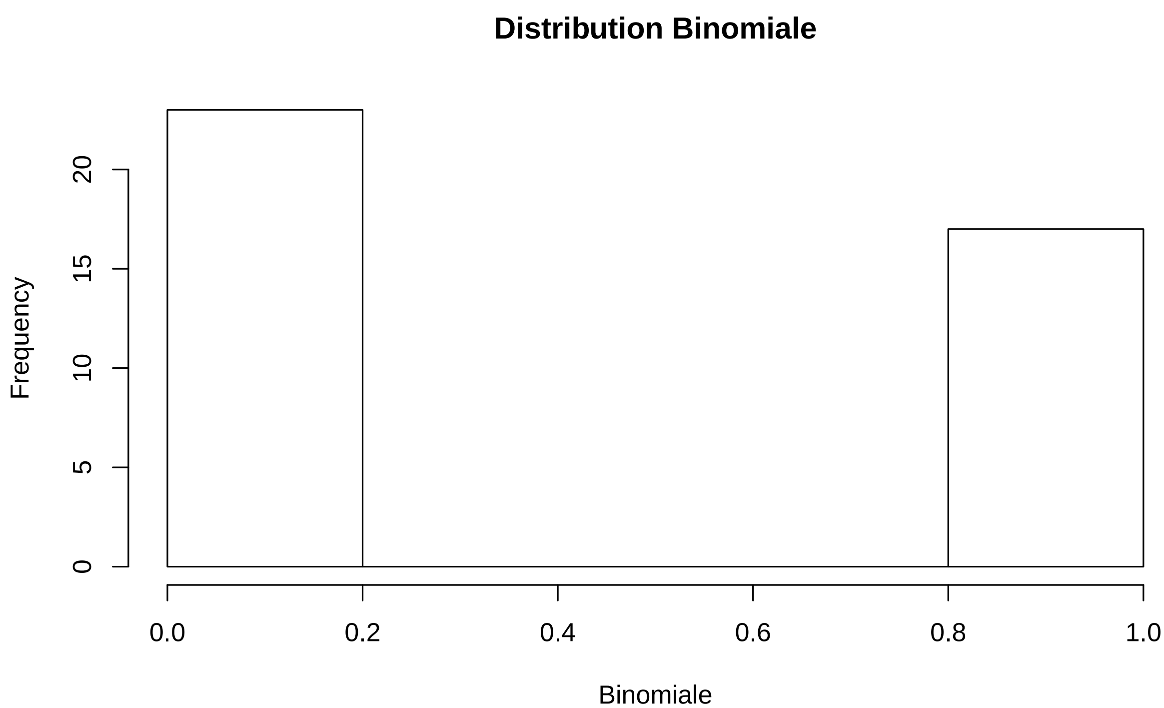


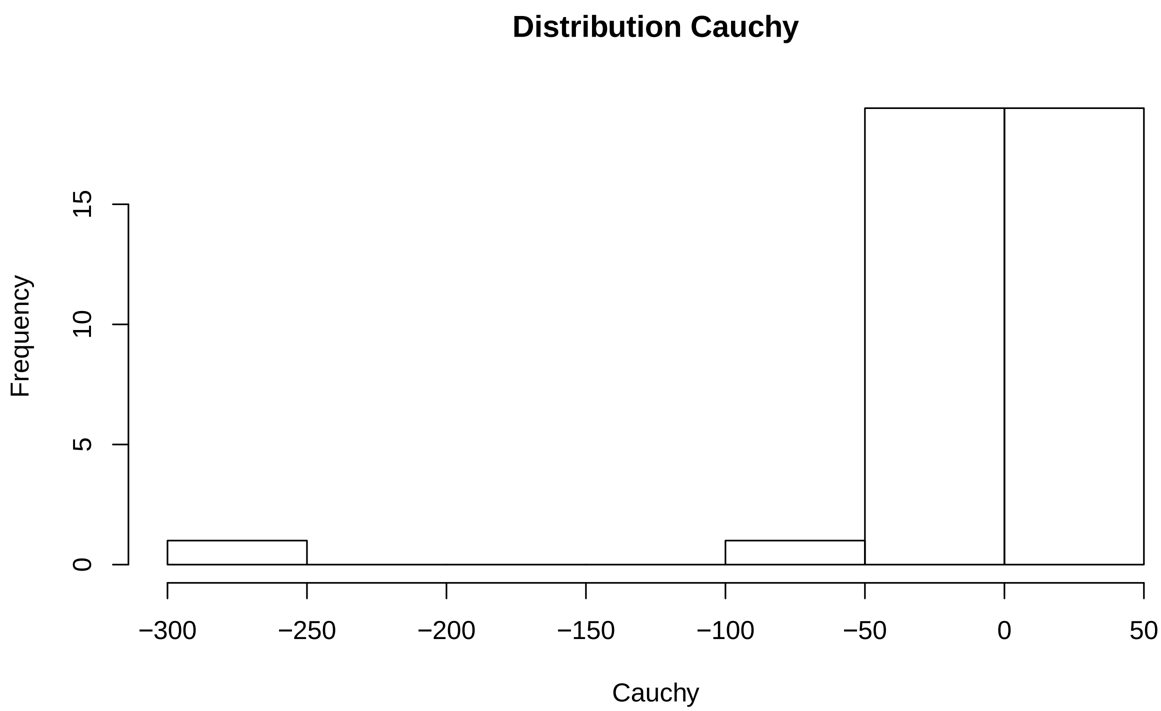



Distribution Poisson
Frequency
0
5
10
15
0
1
2
3
4
Poisson


Distribution Exponentielle
Frequency
0
5
10
15
20
0
1
2
3
4
5
6
Exponentielle


Distribution Chi
Frequency
0
5
10
15
20
0
1
2
3
4
Chi


Distribution Binomiale
Frequency
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
Binomiale


Distribution Cauchy
Frequency
0
5
10
15
−300
−250
−200
−150
−100
−50
0
50
Cauchy

Moment d'ordre

Générons une matrice (sous forme de data.frame) contenant les moments des ordres 1, 2, 3 et 4 de nos distributions.

```
library("moments")
ajouter_ligne <- function(matrice, valeurs, nom) {
  v <- unlist(valeurs)
  m  <- data.frame(nom, mean(v), var(v), skewness(v), kurtosis(v))
  names(m) <- c("Distribution", "Esperance", "Variance", "Skewness", "Kurtosis")
  return (rbind(matrice, m))
}

matrice <- ajouter_ligne(data.frame(), df_inconnu["x"], "Inconnue")
for (distri in distributions) {
  matrice <- ajouter_ligne(matrice, df[distri], distri)
}
print(matrice, digits=5)
```

```
##    Distribution Esperance   Variance Skewness Kurtosis
## 1      Inconnue  -0.11439 8.3443e+00  0.19117   3.0793
## 2    Gaussienne   0.12101 7.2951e-01 -0.15442   2.5186
## 3      Uniforme   0.47246 8.7639e-02  0.07009   1.8756
## 4       Poisson   1.05000 8.6923e-01  0.86227   3.9021
## 5 Exponentielle   1.13629 1.2071e+00  1.63374   6.2219
## 6           Chi   0.83984 1.0552e+00  1.78218   5.3136
## 7     Binomiale   0.42500 2.5064e-01  0.30343   1.0921
## 8        Cauchy  -8.12151 2.0171e+03 -5.27967  30.8299
```

Pour les distributions suivantes, les valeurs théorique des moments sont:

- Gaussienne $(\mu = 0, \sigma = 1)$
  - Espérance : 0
  - Variance : 1
  - Skewness : 0
  - Kurtosis : 3
- Uniforme $(a = 0, b = 1)$
  - Espérance : $\frac{1}{2} = 0.5$
  - Variance : $\frac{1}{12} = 0.084$
  - Skewness : 0
  - Kurtosis : 1.8 => l'extrémité de la densité tend rapidement vers 0.
- Poisson $(\lambda = 1)$
  - Espérance : 1
  - Variance : 1
  - Skewness : 1
  - Kurtosis : 4
- Exponentielle $(\lambda = 1)$
  - Espérance : 1
  - Variance : 1
  - Skewness : 2 => notre densité est dissymétrique vers la droite.
  - Kurtosis : 9
- χ^2 (Chi carré) ($df = 1$ (degree of freedom <=> degrée de liberté))
  - Espérance : 1
  - Variance : 2
  - Skewness : $\sqrt{8} = 2.8$ => notre densité est dissymétrique vers la droite.
  - Kurtosis : 15

- Binomiale ($n = 1, p = 0.5$)
  - Espérance : 0.5
  - Variance : 0.25
  - Skewness : 0 => notre densité est symétrique.
  - Kurtosis : 1
- Cauchy : les moments sont non-définis.

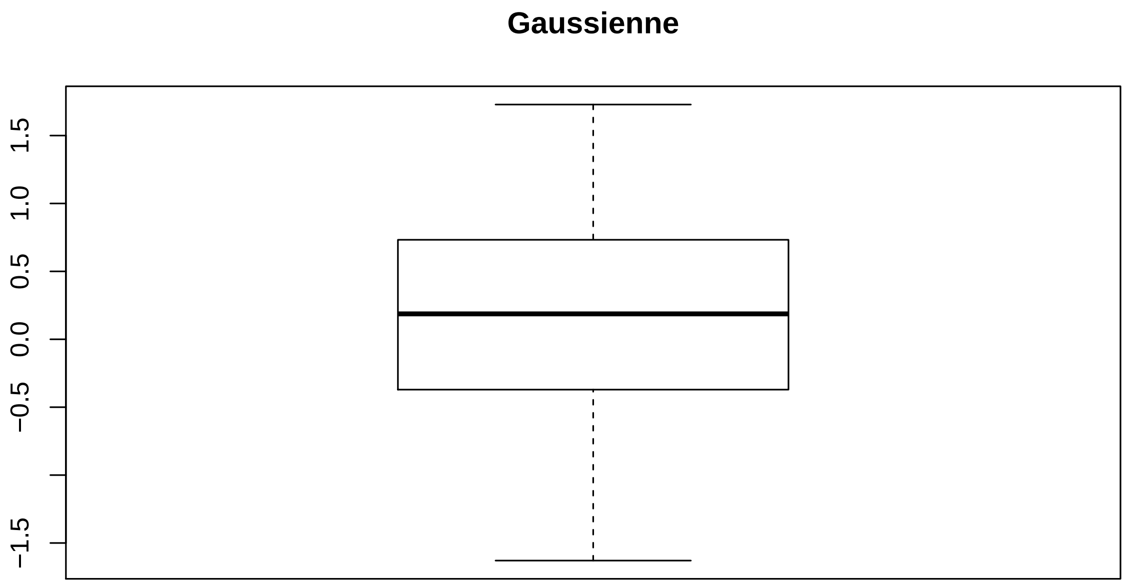
NB: Les Kurtosis utilisés sont 'non-normalisés' : j'ai ajouté '+ 3' aux valeurs théoriques normalisés ("Excess Kurtosis").

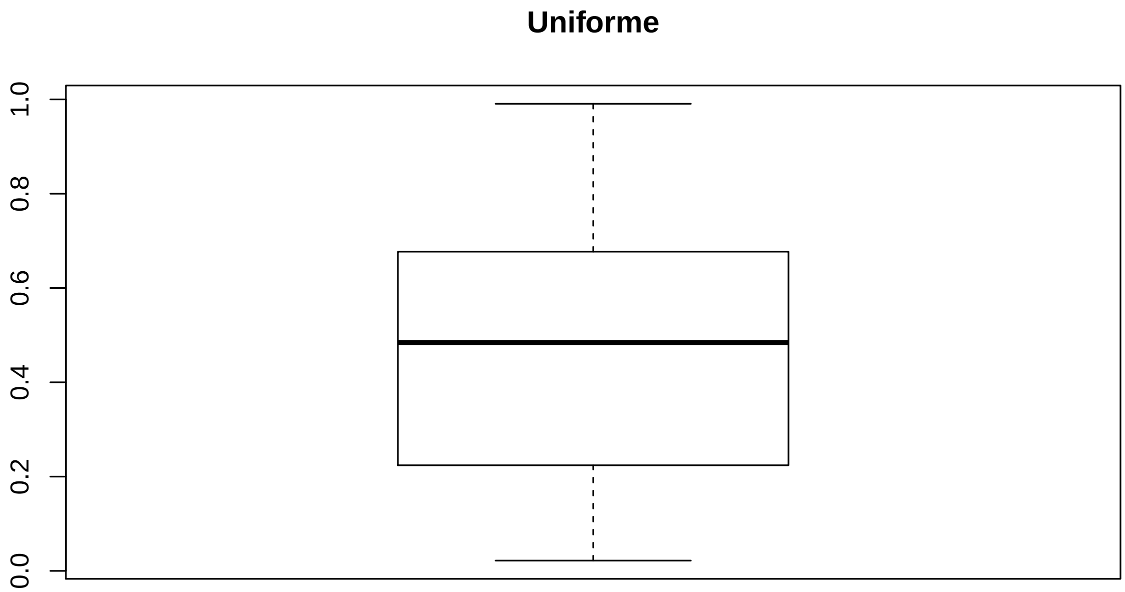
Les résultats obtenus suivent les valeurs théoriques des différents moments, mais peuvent parfois s'en éloigner selon les échantillons générés.

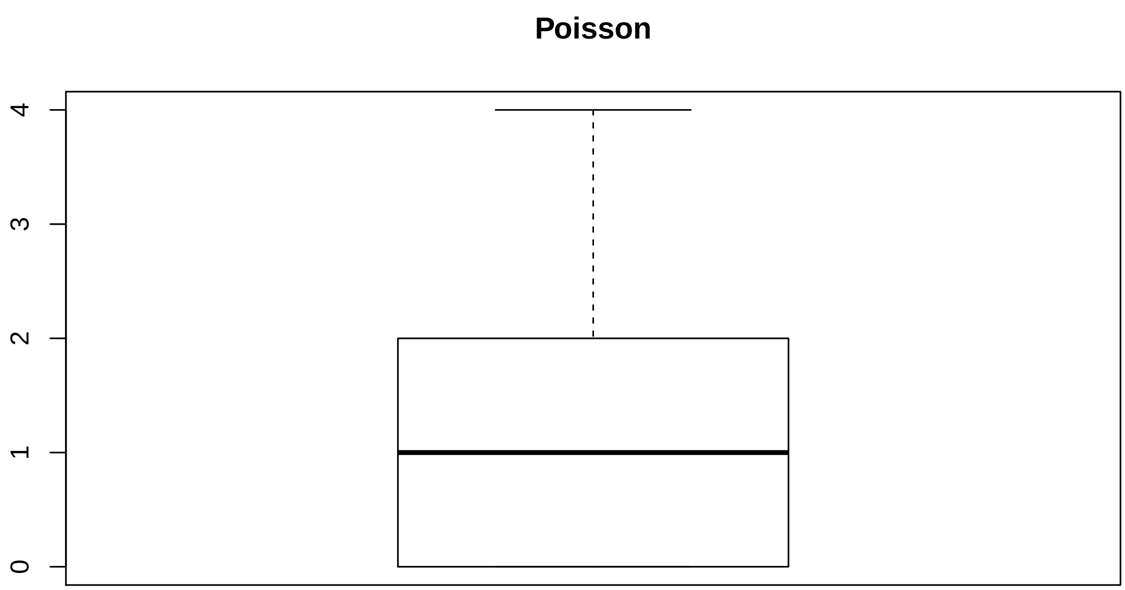
L'hypothèse précèdente ('la distribution inconnue suit une loi normal $\mu = 0, \sigma = 5$') semble d'autant plus probable, car les Kurtosis de la distribution inconnue sont égal à ceux d'une telle distribution normale.

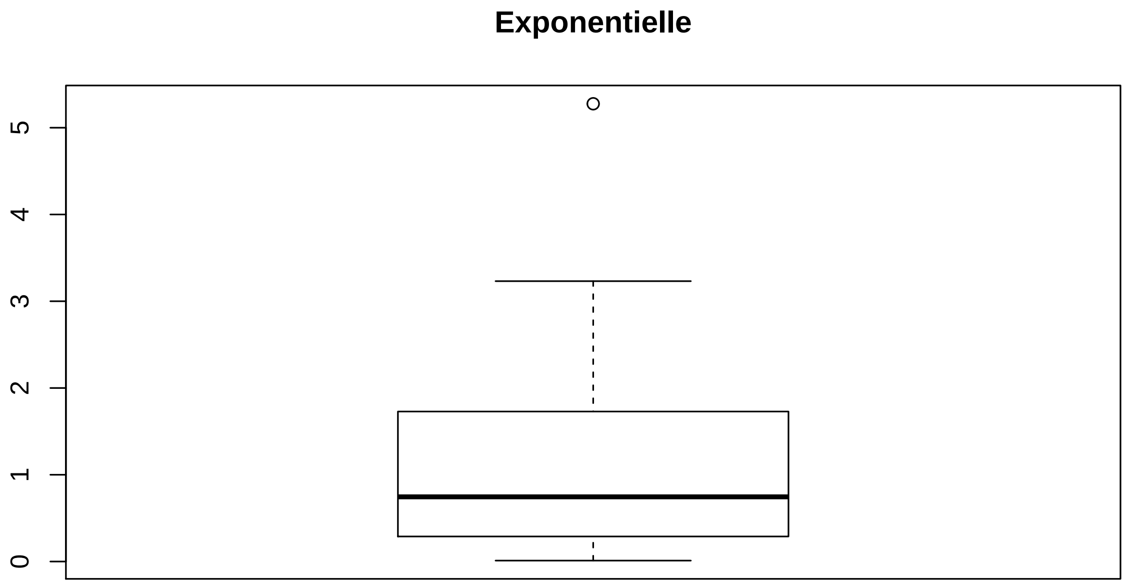
Quantiles et Boxplot

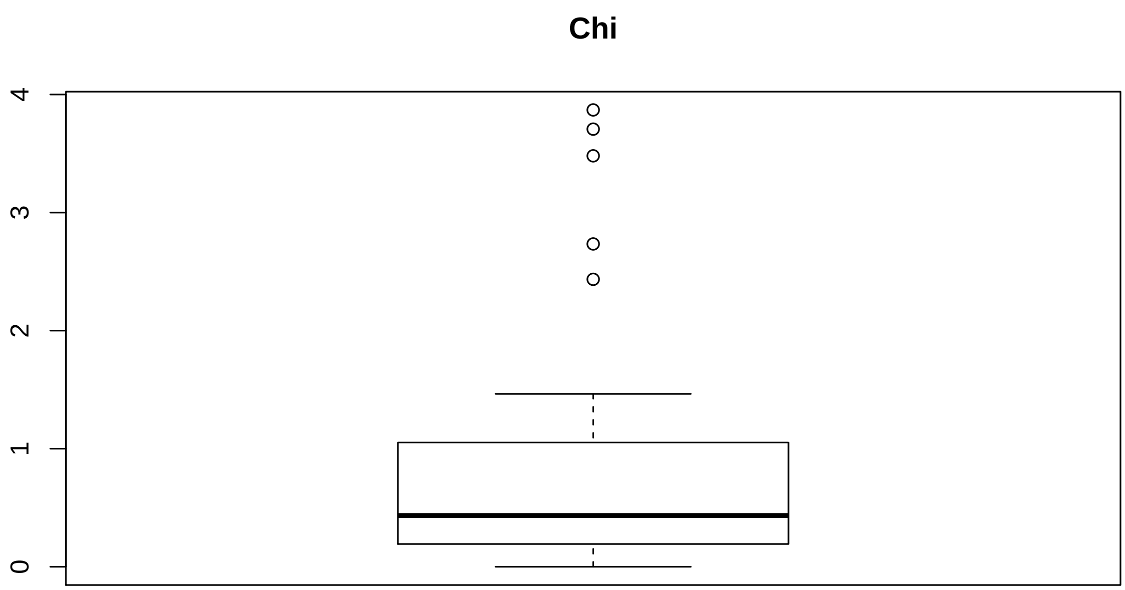
```r
for (distri in distributions) {
  x <- unlist(df[distri])
  boxplot(x, main=distri)
}
```

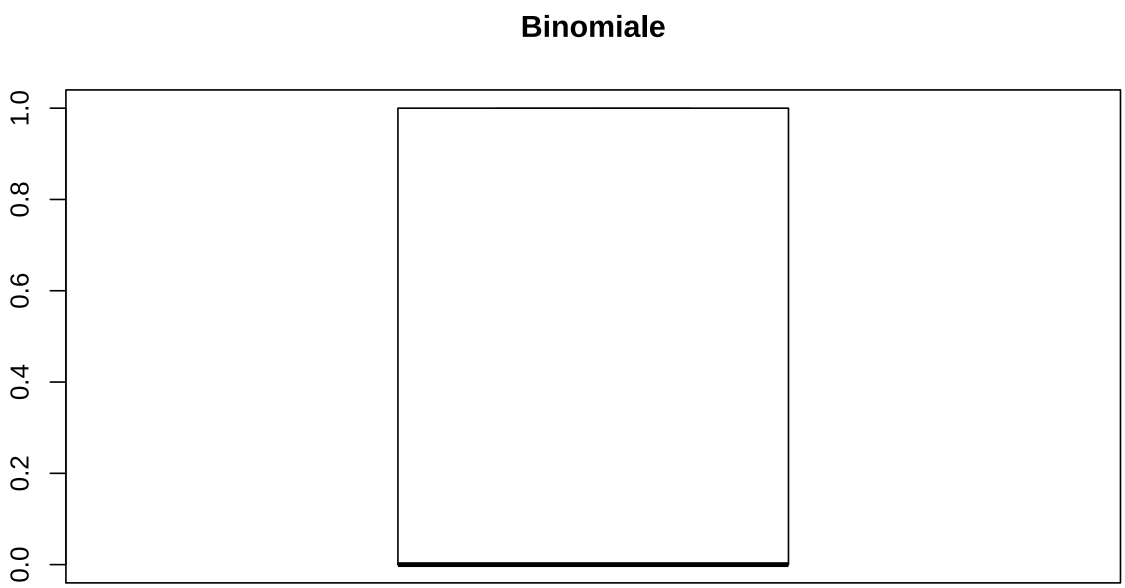


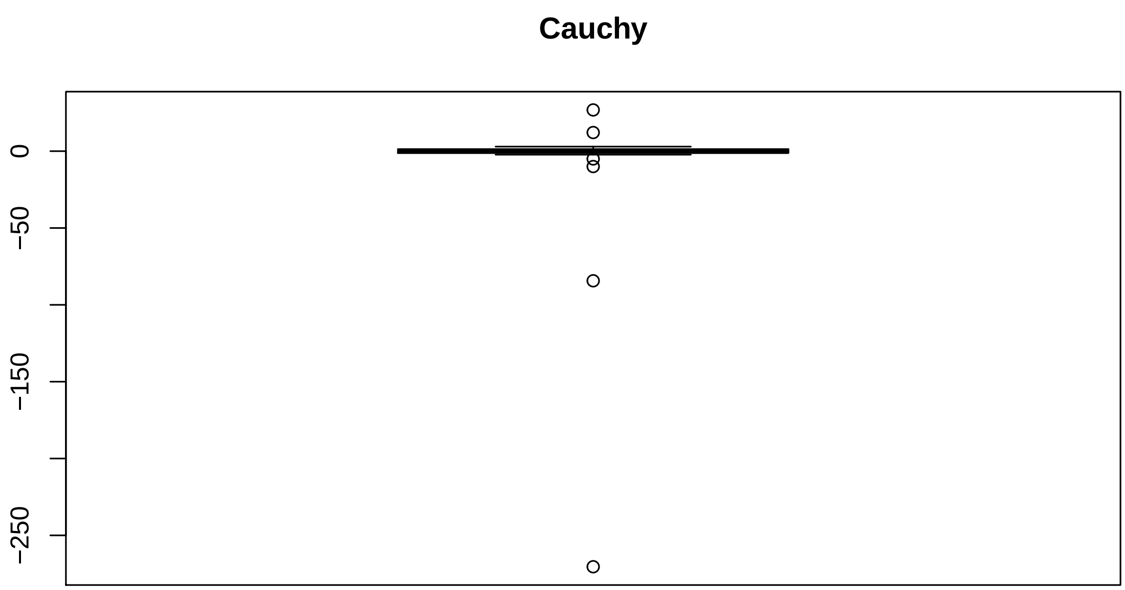

Uniforme
1.0
0.8
0.6
0.4
0.2
0.0


Poisson
4
3
2
1
0


Exponentielle
0
1
2
3
4
5


Chi
0
1
2
3
4






```r
# genere les colonnes Q1, Q2 et Q3
Q  <- quantile(unlist(df_inconnu["x"]), c(0.25, 0.5, 0.75))
Q1 <- c(Q[1])
Q2 <- c(Q[2])
```

```r
Q3 <- c(Q[3])
for (distri in distributions) {
  Q  <- quantile(unlist(df[distri]), c(0.25, 0.5, 0.75))
  Q1 <- c(Q1, Q[1])
  Q2 <- c(Q2, Q[2])
  Q3 <- c(Q3, Q[3])
}

# ajoute les colonnes au data frame
matrice <- cbind(matrice, Q1)
matrice <- cbind(matrice, Q2)
matrice <- cbind(matrice, Q3)
print(matrice, digits=5)
```

```
##    Distribution Esperance   Variance Skewness Kurtosis       Q1       Q2
## 1      Inconnue  -0.11439 8.3443e+00  0.19117   3.0793 -2.19593 -0.24533
## 2    Gaussienne   0.12101 7.2951e-01 -0.15442   2.5186 -0.33735  0.18709
## 3      Uniforme   0.47246 8.7639e-02  0.07009   1.8756  0.22473  0.48425
## 4       Poisson   1.05000 8.6923e-01  0.86227   3.9021  0.00000  1.00000
## 5 Exponentielle   1.13629 1.2071e+00  1.63374   6.2219  0.29739  0.74549
## 6           Chi   0.83984 1.0552e+00  1.78218   5.3136  0.20117  0.43396
## 7     Binomiale   0.42500 2.5064e-01  0.30343   1.0921  0.00000  0.00000
## 8        Cauchy  -8.12151 2.0171e+03 -5.27967  30.8299 -0.96595 -0.14649
##        Q3
## 1 1.88328
## 2 0.73033
## 3 0.67032
## 4 2.00000
## 5 1.72721
## 6 1.03952
## 7 1.00000
## 8 1.23301
```

Interpretation visuelle

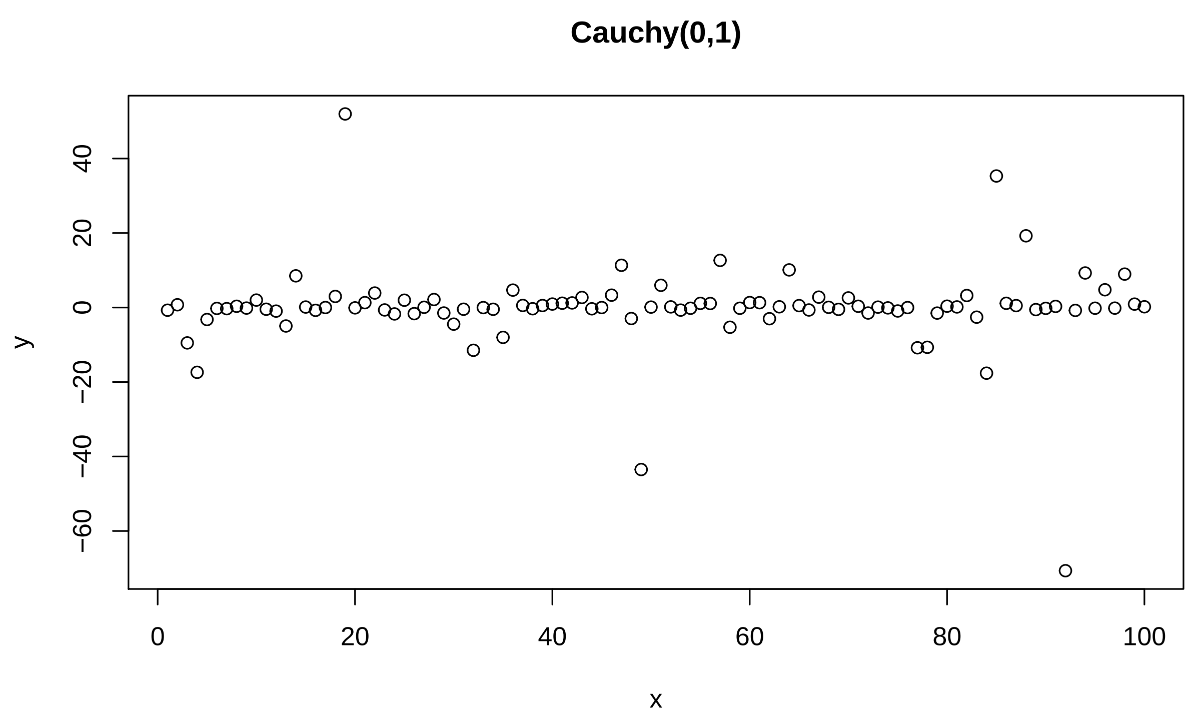
```r
# genere des distriutions de cauchy, avec n=100, et (x0, a) dans {(0, 1), (1, 1), (0, 2)}
n <- 100
params <- list(c(0, 1), c(50, 1), c(0, 4))

# genere le nom des colonnes
noms <- c()
for (p in params) {
  x0   <- p[1]
  a    <- p[2]
  noms <- c(noms, paste("Cauchy(", x0, ",", a, ")", sep=""))
}

# genere le data frame
df <- data.frame(matrix(ncol=length(params), nrow=n))
names(df) <- noms
for (i in 1:length(params)) {
  nom <- noms[i]
  x0  <- params[[i]][1]
  a   <- params[[i]][2]
  df[nom] <- rcauchy(n, location=x0, scale=a)
}
print(df)
```

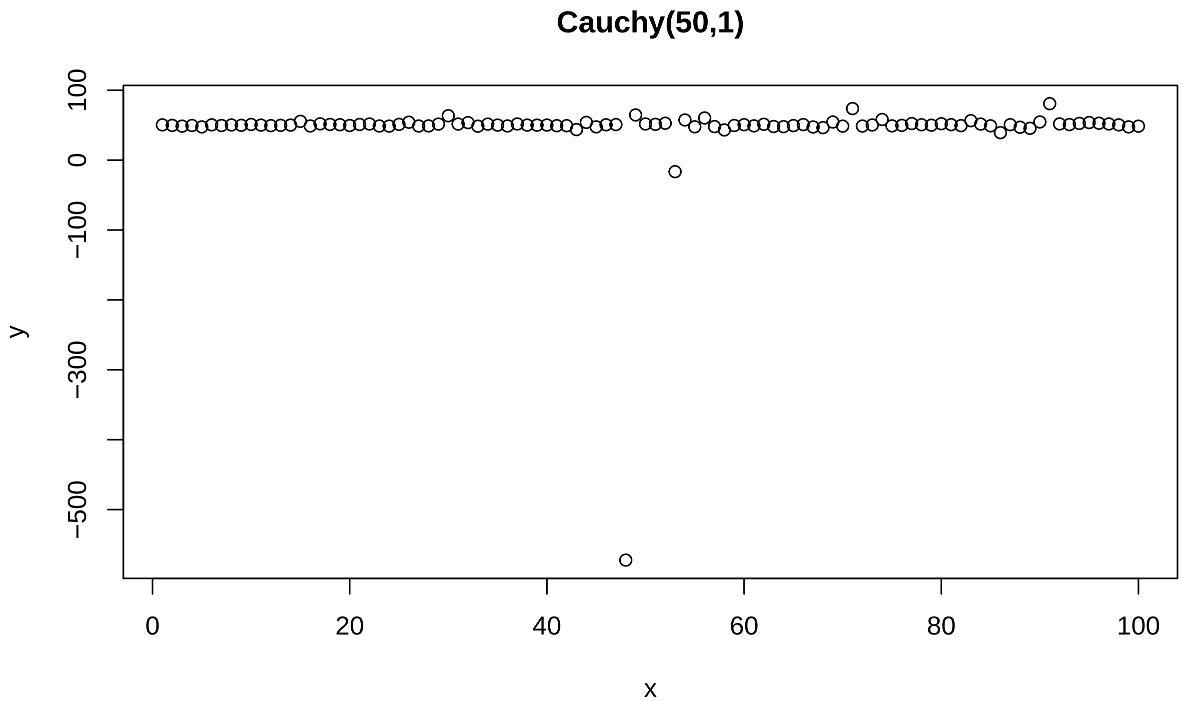
```
##       Cauchy(0,1) Cauchy(50,1)   Cauchy(0,4)
## 1    -0.722434406     50.45835  1.405308e+00
## 2     0.736969950     49.65282 -3.721663e+00
## 3    -9.500719308     48.58686 -1.148245e+02
## 4   -17.396932879     49.57415 -1.326968e+00
## 5    -3.224079477     47.53600 -9.902497e+00
## 6    -0.242917062     50.47118 -7.756598e+00
## 7    -0.303148674     49.51728  1.757066e+00
## 8     0.337222074     50.32713  1.046850e+00
## 9    -0.151346854     49.81820 -4.359583e+00
## 10    1.994210472     51.03253 -2.769538e+00
## 11   -0.460901244     50.10195 -8.256229e+00
## 12   -0.968260781     49.32479  2.975372e+00
## 13   -4.963838176     49.76627 -3.101148e+00
## 14    8.501325570     50.11697  1.085298e+01
## 15    0.126495596     55.51870 -1.571849e-01
## 16   -0.756980836     48.86053  3.290011e+00
## 17   -0.003150307     51.88068 -2.186620e+00
## 18    2.974198596     51.04688  8.581582e+02
## 19   51.972689646     50.68387 -7.450363e-02
## 20   -0.095403258     49.50102  1.087292e+01
## 21    1.306930626     51.02715 -1.268223e+00
## 22    3.887625700     51.66227 -3.899892e+01
## 23   -0.655723729     48.81510  2.039568e+01
## 24   -1.727011055     48.66801  4.497310e+00
## 25    1.961816300     51.10935  3.319762e+00
## 26   -1.652574338     54.25397  1.435610e+01
## 27    0.063637090     48.79199 -2.993896e+00
## 28    2.147987111     48.94344 -2.515899e+00
```

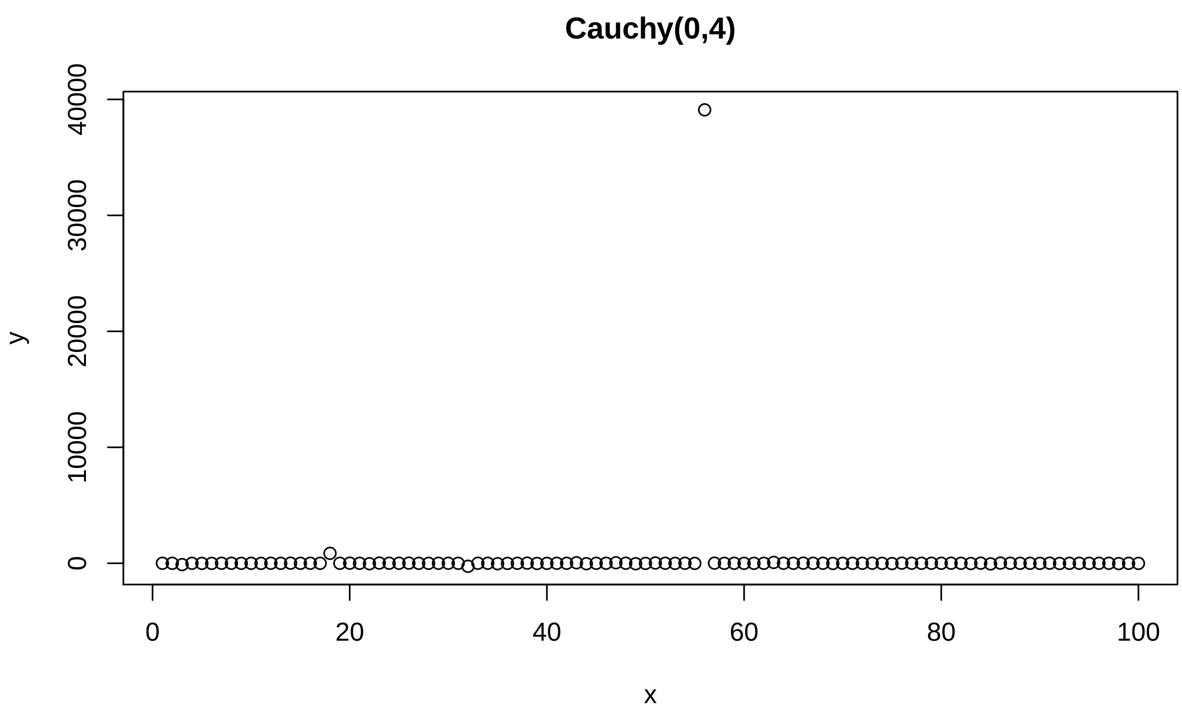
```
## 29   -1.476752305     51.53147  7.119189e-01
## 30   -4.470362238     63.47772  6.587001e-01
## 31   -0.464585351     51.56923 -4.922464e-01
## 32  -11.486549871     53.64846 -2.575454e+02
## 33   -0.026336318     48.54878  2.591953e+00
## 34   -0.475513407     51.48022  2.216228e+00
## 35   -8.016446855     50.13048 -3.375364e+01
## 36    4.669260514     48.87765 -8.926862e+00
## 37    0.555881059     51.73740  7.084191e-01
## 38   -0.330995663     50.10526  1.187313e+01
## 39    0.528029069     50.01170 -1.374174e+01
## 40    0.926558521     50.16541 -6.619017e+00
## 41    1.156749325     49.27240  8.758556e-01
## 42    1.229388047     49.24997  1.231089e+00
## 43    2.700475230     43.65876  4.402125e+01
## 44   -0.346175747     53.99778 -3.601777e+01
## 45   -0.016572626     47.57792 -5.531757e-01
## 46    3.315625211     50.56834  1.550669e+00
## 47   11.339078203     50.87167  4.548662e+01
## 48   -2.957939531   -572.29705  4.941179e+00
## 49  -43.498149898     64.59138 -4.002572e+01
## 50    0.106683101     51.71214 -9.731411e+00
## 51    5.966755068     51.21911  2.348412e+01
## 52    0.165905523     52.79544  2.375064e+00
## 53   -0.701750547    -16.51888 -6.064437e+00
## 54   -0.199381112     57.45446  7.925701e-01
## 55    1.114979098     47.66564 -8.868526e+00
## 56    1.074294951     60.32588  3.909997e+04
## 57   12.650881548     48.00244  1.255371e+01
## 58   -5.298142678     43.19309 -2.515935e+00
## 59   -0.202976294     49.55482  1.518856e+00
## 60    1.329263130     50.70075 -3.695879e+00
## 61    1.293526611     49.01118 -2.541565e+00
## 62   -3.007918987     51.27275 -2.894158e+00
## 63    0.188589960     48.06863  7.518160e+01
## 64   10.100079129     47.86366  1.385164e+00
## 65    0.510503696     49.38664 -1.125051e+00
## 66   -0.665012758     50.82720  1.106289e+01
## 67    2.783194215     47.42995  1.655652e+00
## 68    0.054339606     46.47566  1.932451e+00
## 69   -0.499064706     54.51684 -1.674297e+01
## 70    2.576205120     48.62160 -2.768068e+00
## 71    0.332271297     73.70657 -2.368635e+00
## 72   -1.478739262     48.69455  1.714397e+00
## 73    0.104128936     50.27584  6.683931e+00
## 74   -0.102565089     58.15950 -5.346524e+00
## 75   -0.935205604     48.96589 -2.564434e+01
## 76   -0.023111650     49.71042  1.170839e+01
## 77  -10.812122034     52.28636  3.516192e+00
## 78  -10.676337957     50.60554 -1.578607e+00
## 79   -1.498333857     49.96825  1.833411e+00
## 80    0.355910075     52.04157  1.813602e+00
## 81    0.157004477     50.72101  6.304870e+00
## 82    3.225187403     49.32663 -1.853917e+00
```

```
## 83   -2.593293409     56.40850 -1.881113e+01
## 84  -17.632040861     51.56285  9.931270e+00
## 85   35.311241518     49.08717 -5.265185e+01
## 86    1.130247874     39.30331  1.504350e+01
## 87    0.508624416     50.71984  2.096531e+00
## 88   19.248244845     47.02306 -2.297537e+00
## 89   -0.559149693     45.51769 -2.874770e-01
## 90   -0.222310133     54.60302 -6.484855e+00
## 91    0.302208131     80.76574  9.348008e+00
## 92  -70.664970936     51.80258 -1.581939e+01
## 93   -0.777082951     50.87561 -2.111660e+00
## 94    9.270387570     52.55762 -5.908774e-01
## 95   -0.187523033     53.75657 -1.935299e+00
## 96    4.762963065     52.80401  9.944273e-01
## 97   -0.166856413     51.75442  4.528344e-02
## 98    8.965538288     50.44184 -2.507949e+01
## 99    0.908587619     47.56555 -2.328381e+00
## 100   0.196582037     48.56957 -9.369490e+00
```

```r
# export en .csv
write.csv(df, file="./cauchy_100.csv")

# trace les distributions en nuage de points
x <- 1:n
for (nom in noms) {
  y <- unlist(df[nom])
  plot(x, y, main=nom)
}
```

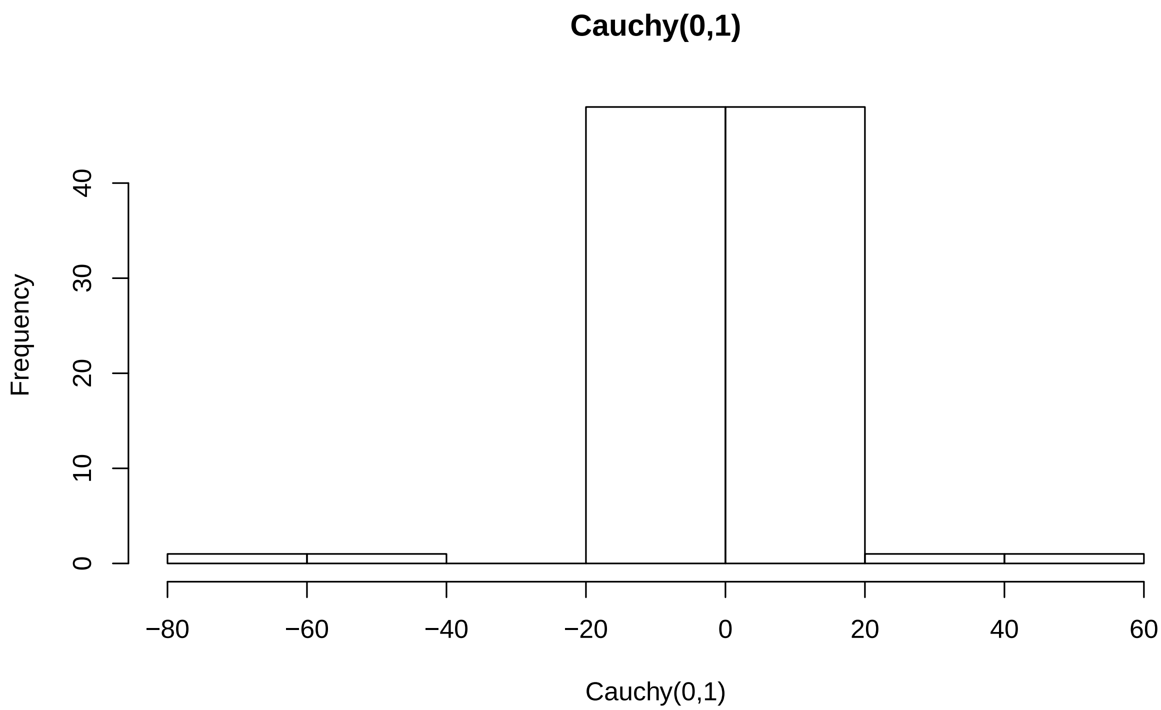




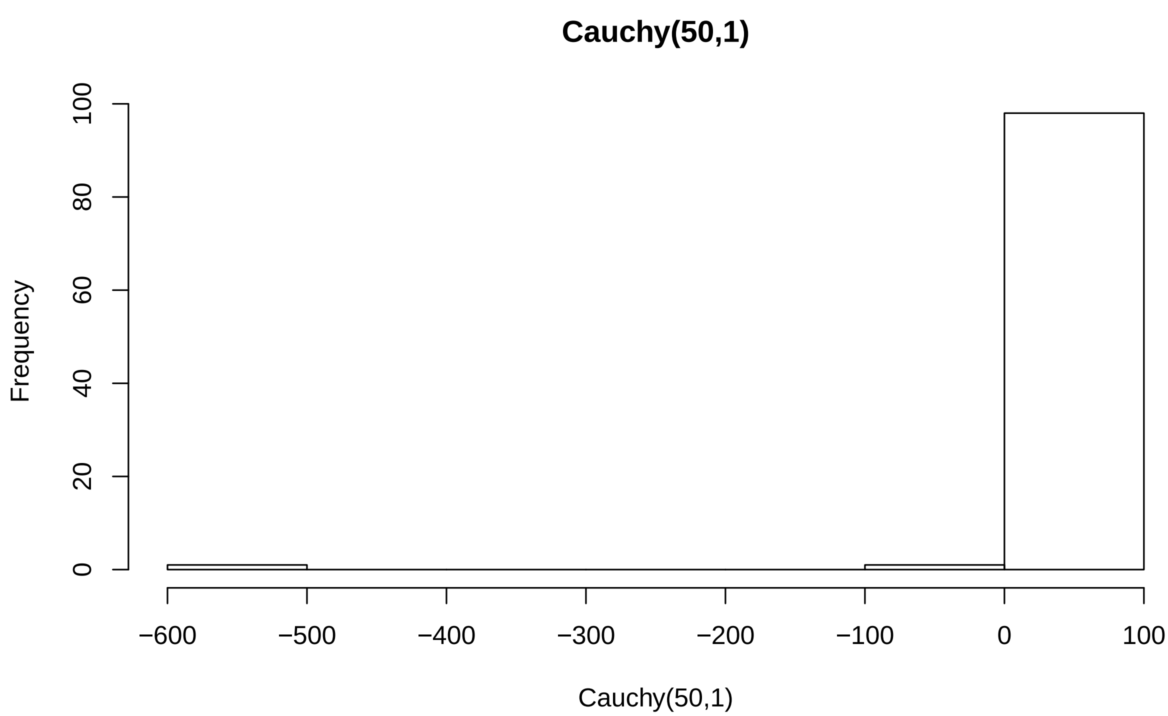



```
# trace les distributions en histogramme
for (nom in noms) {
  y <- unlist(df[nom])
  hist(y, breaks="Sturges", xlab=nom, main=nom)
```

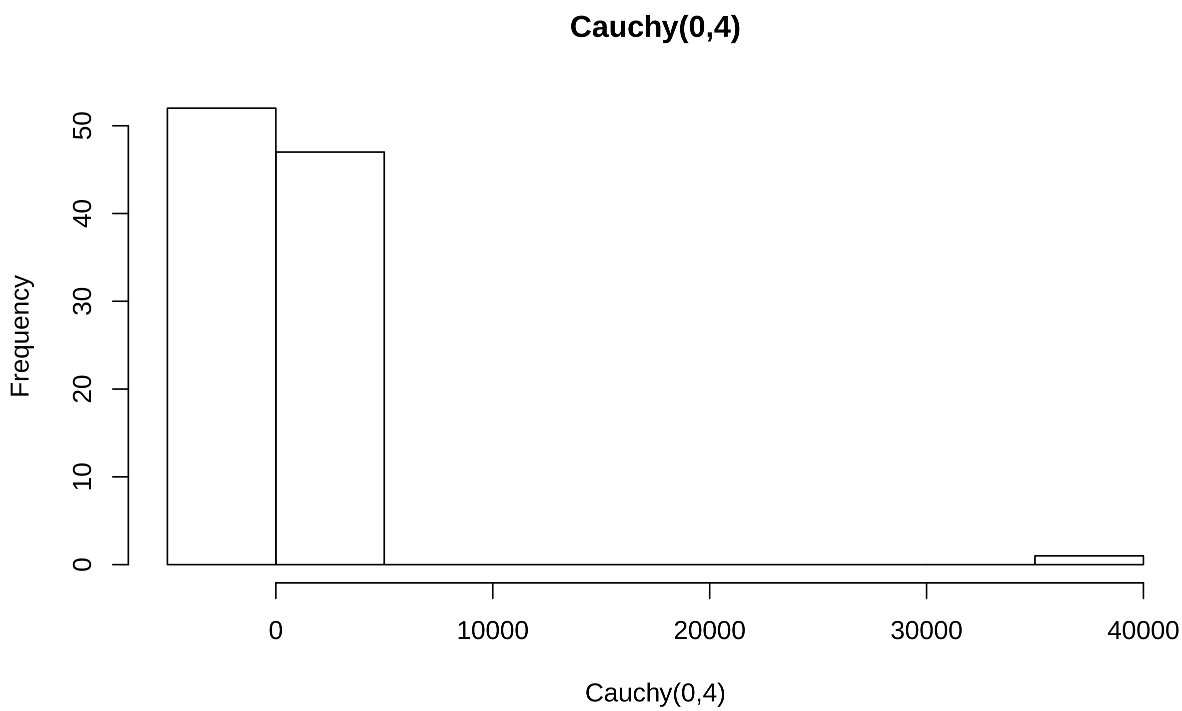
```
}
```



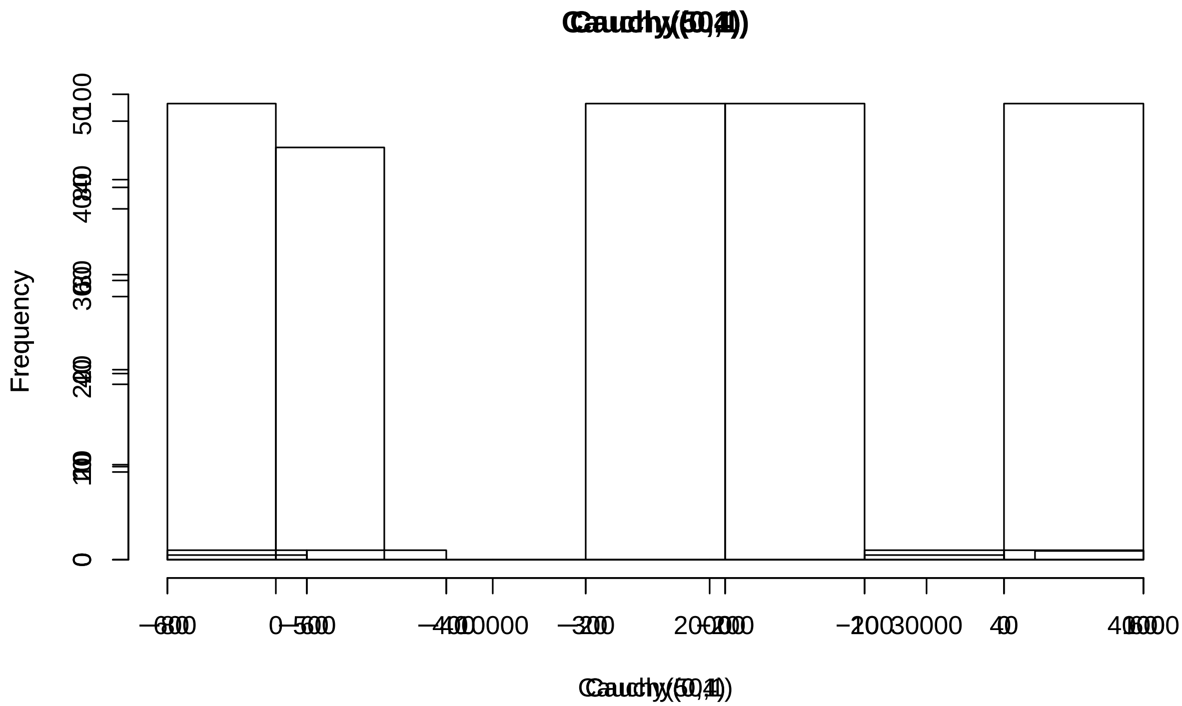
Cauchy(0,1)
Frequency
0
10
20
30
40
−80
−60
−40
−20
0
20
40
60
Cauchy(0,1)



Cauchy(50,1)
Frequency
0
20
40
60
80
100
−600
−500
−400
−300
−200
−100
0
100
Cauchy(50,1)



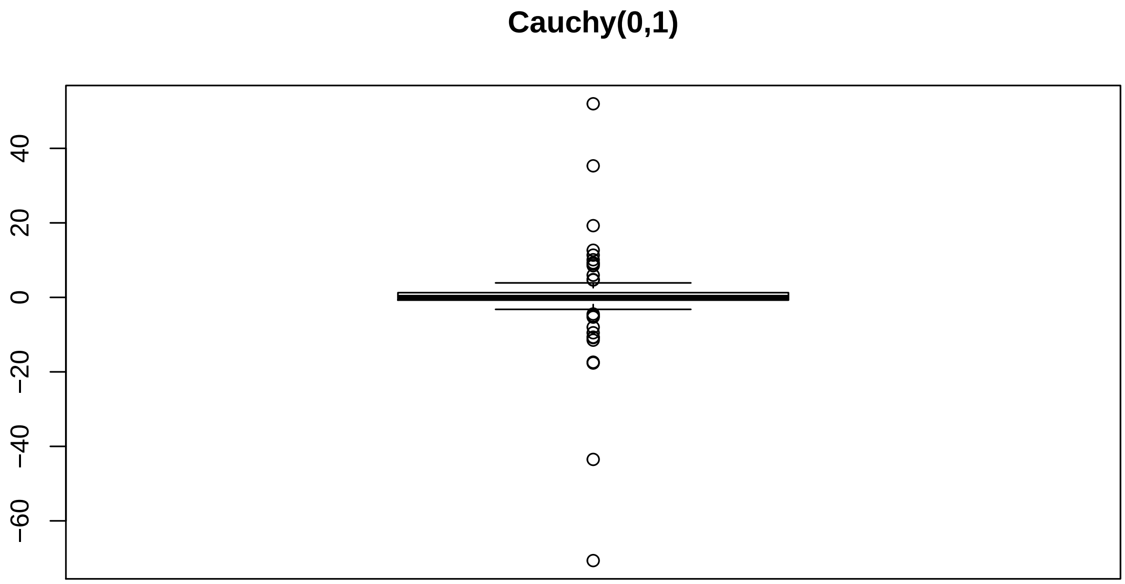

```
# trace les distributions en histogramme sur la meme figure ...
for (nom in noms) {
  y <- unlist(df[nom])
  hist(y, breaks="Sturges", xlab=nom, main=nom)
  par(new=TRUE)
}
par(new=FALSE)
```

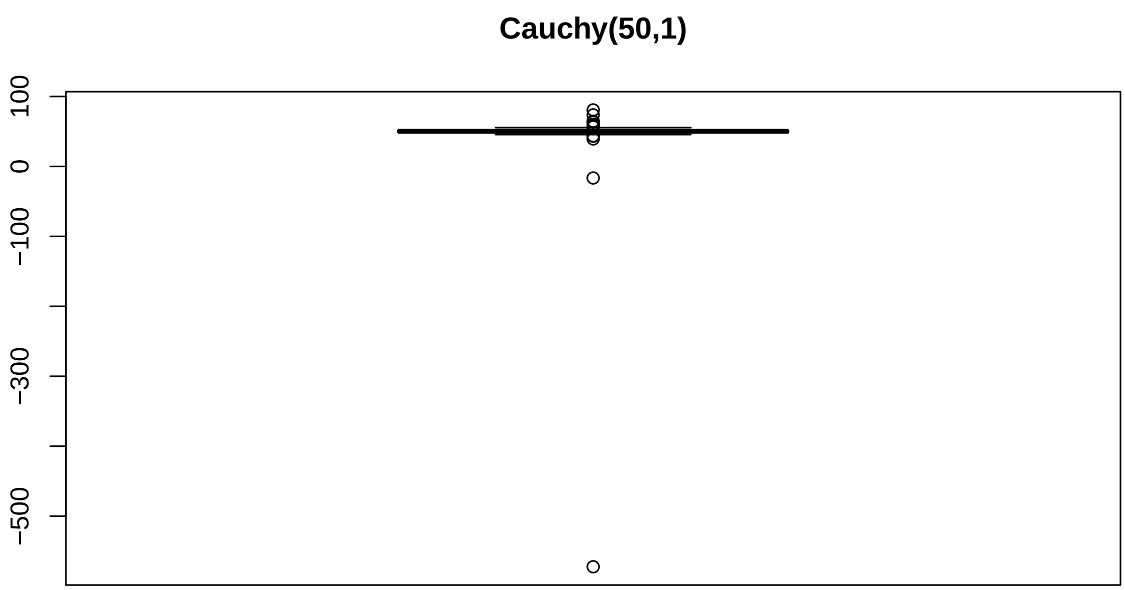


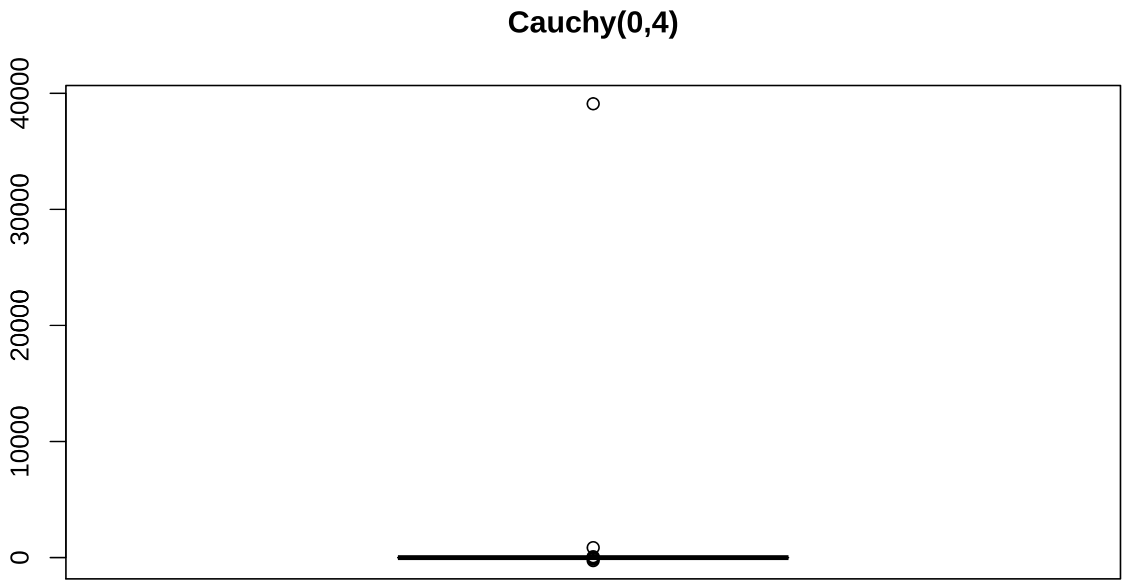

```r
# genere les moments et les quantiles
library("moments")
ajouter_ligne <- function(matrice, valeurs, nom) {
  x <- unlist(valeurs)
  Q <- quantile(x, c(0.25, 0.5, 0.75))
  m  <- data.frame(nom, mean(x), var(x), skewness(x), kurtosis(x), Q[[1]][1], Q[[2]][1], Q[[3]][1])
  names(m) <- c("Distribution", "Esperance", "Variance", "Skewness", "Kurtosis", "Q1", "Q2", "Q3")
  return (rbind(matrice, m))
}

matrice <- data.frame()
for (nom in noms) {
  matrice <- ajouter_ligne(matrice, df[nom], nom)
}
print(matrice, digits=3)
```

```
##   Distribution Esperance Variance Skewness Kurtosis     Q1      Q2    Q3
## 1  Cauchy(0,1)    -0.182      134    -1.61     21.4 -0.731  0.0256  1.25
## 2 Cauchy(50,1)    44.250     3950    -9.59     94.5 48.873 50.2206 51.67
## 3  Cauchy(0,4)   395.193 15293231     9.84     97.9 -3.881 -0.2223  2.69
```

```r
# trace les distributions avec boxplot
for (nom in noms) {
  y <- unlist(df[nom])
  boxplot(y, main=nom)
}
```


Cauchy(0,1)
40
20
0
−20
−40
−60


Cauchy(50,1)
100
0
−100
−300
−500



On remarque qu'une distribution de Cauchy(x0, a) a une forte probabilité d'avoir des valeurs dans [x0 - a, x0 + a], mais que certains tirages peuvent rapidement s'éloigner de cette intervalle.