TP1

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  -1.82365849
## 2   0.74620136
## 3   0.08489944
## 4   1.03753241
## 5   0.14597416
## 6  -0.47905992
## 7   0.43460537
## 8  -0.19090713
## 9   0.78292668
## 10 -0.33938722
## 11  0.06682523
## 12 -0.59267886
## 13 -0.82687490
## 14 -1.60847690
## 15 -1.79352084
## 16  0.39434817
## 17 -0.78058286
## 18 -1.17502847
## 19 -0.60009794
## 20  1.77833460
## 21  0.85400559
## 22 -1.41485668
```

```
## 23  0.04536216
## 24  1.81184492
## 25 -0.50865411
## 26  1.11929913
## 27  0.29113082
## 28 -0.54445865
## 29  0.04489107
## 30 -1.77297465
## 31  0.60300914
## 32 -0.02736943
## 33 -0.74192859
## 34 -2.48970323
## 35 -0.52365540
## 36  1.37508489
## 37 -0.01431349
## 38  2.35695646
## 39 -1.35958819
## 40 -1.02163297
```

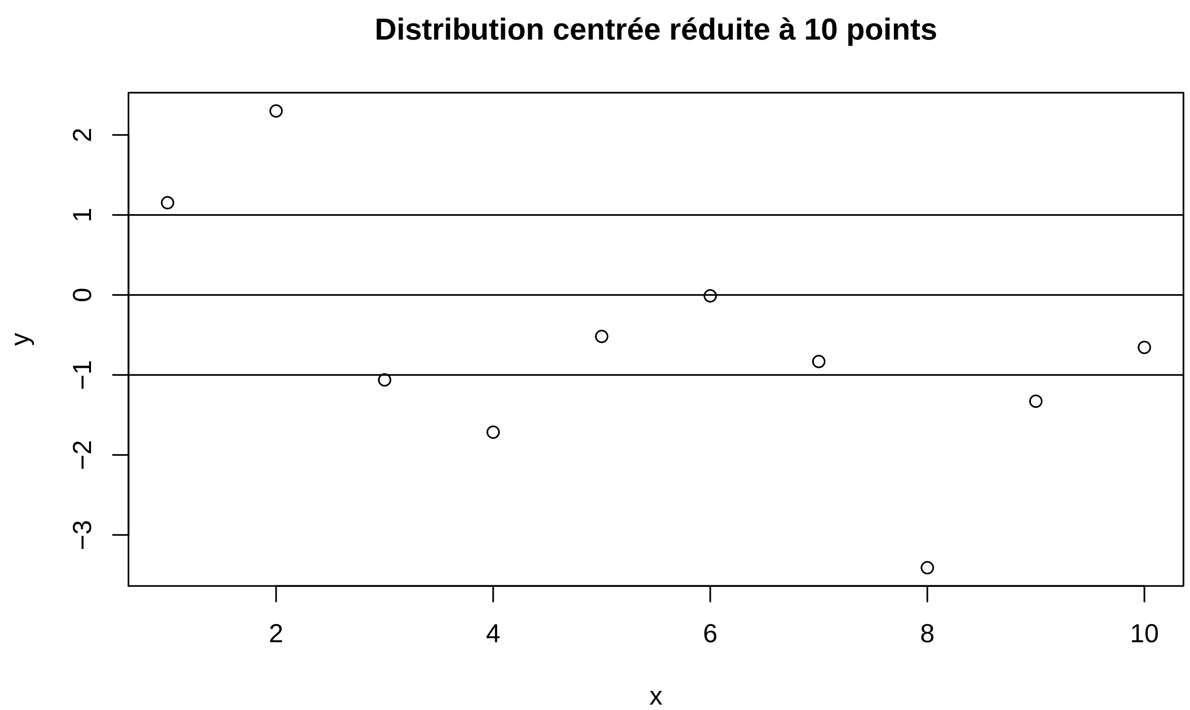
```r
# ou
df2 <- read.table(file="./samples_40.txt", header=TRUE)
df2["Gaussienne"]
```

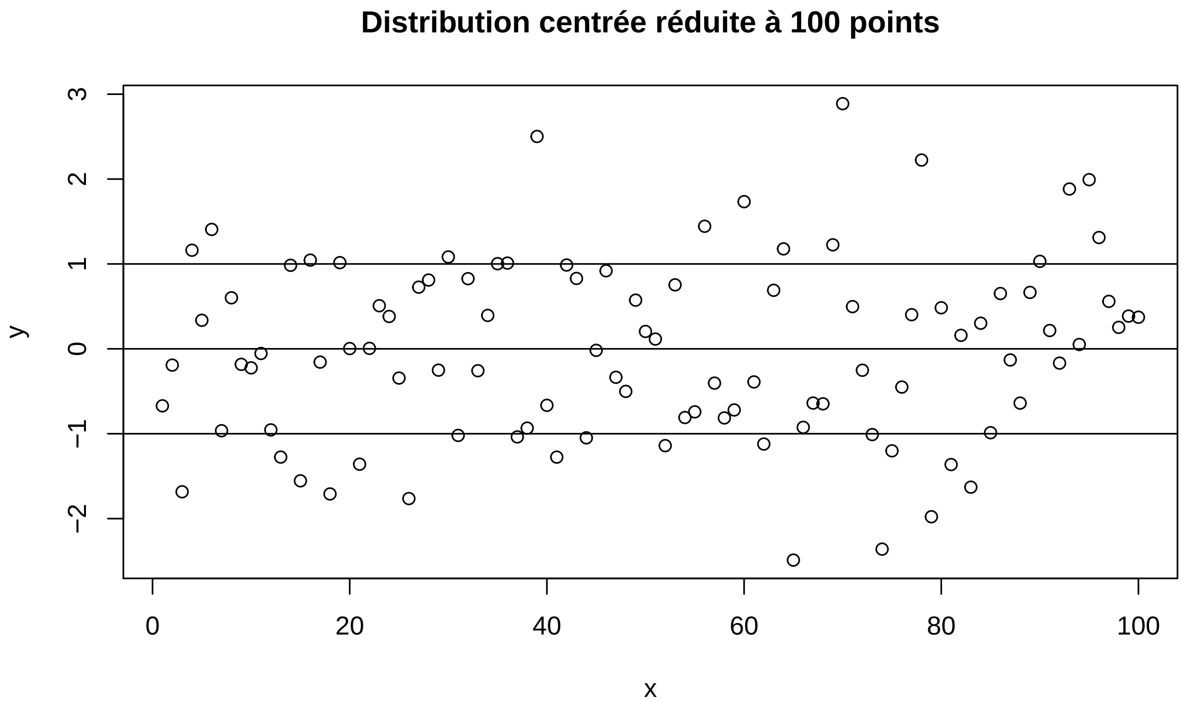
```
##     Gaussienne
## 1  -1.82365849
## 2   0.74620136
## 3   0.08489944
## 4   1.03753241
## 5   0.14597416
## 6  -0.47905992
## 7   0.43460537
## 8  -0.19090713
## 9   0.78292668
## 10 -0.33938722
## 11  0.06682523
## 12 -0.59267886
## 13 -0.82687490
## 14 -1.60847690
## 15 -1.79352084
## 16  0.39434817
## 17 -0.78058286
## 18 -1.17502847
## 19 -0.60009794
## 20  1.77833460
## 21  0.85400559
## 22 -1.41485668
## 23  0.04536216
## 24  1.81184492
## 25 -0.50865411
## 26  1.11929913
## 27  0.29113082
## 28 -0.54445865
## 29  0.04489107
## 30 -1.77297465
## 31  0.60300914
```

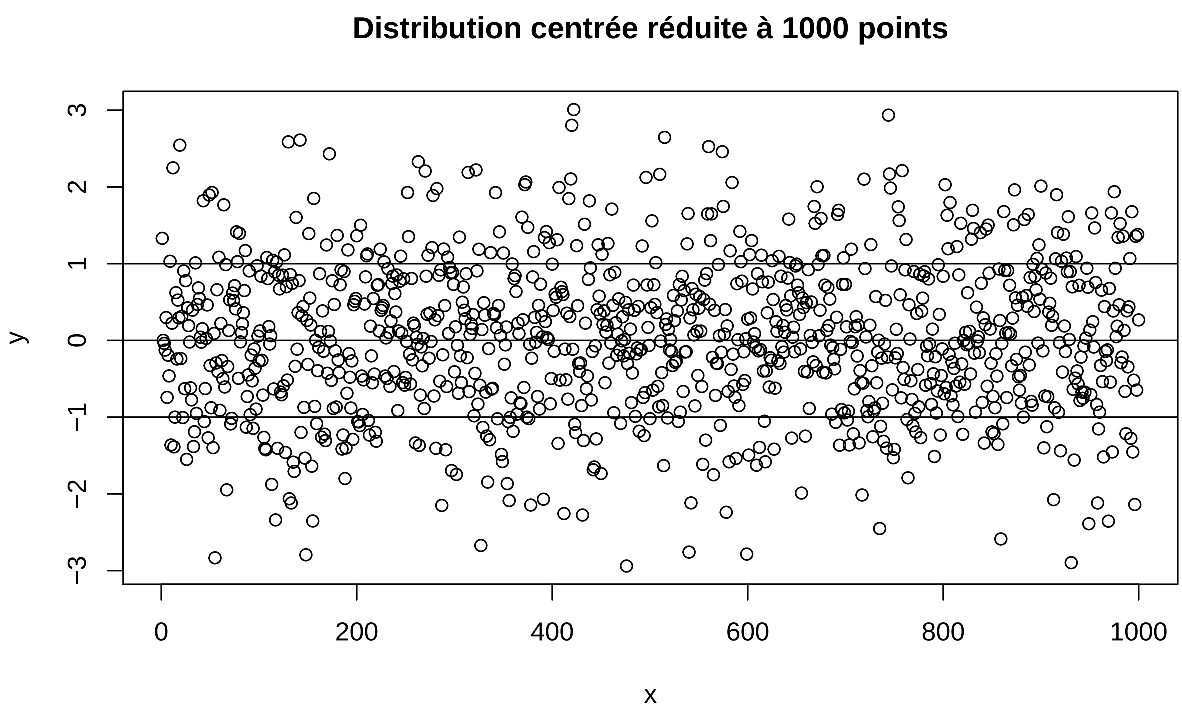
```
## 32 -0.02736943
## 33 -0.74192859
## 34 -2.48970323
## 35 -0.52365540
## 36  1.37508489
## 37 -0.01431349
## 38  2.35695646
## 39 -1.35958819
## 40 -1.02163297
```

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

```
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)
}
```





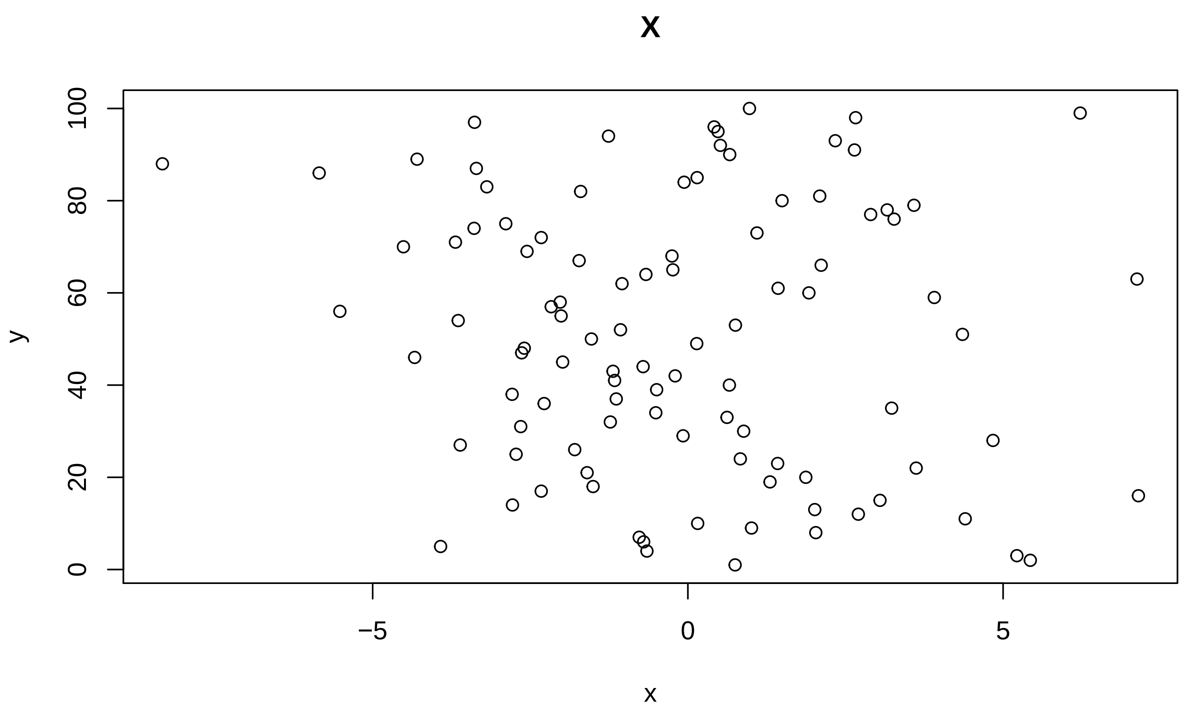


On remarque qu'il y a environ autant de valeurs positives que négatives, et que la répartition est d'autant plus dense que l'on se rapproche de l'axe y=0.

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

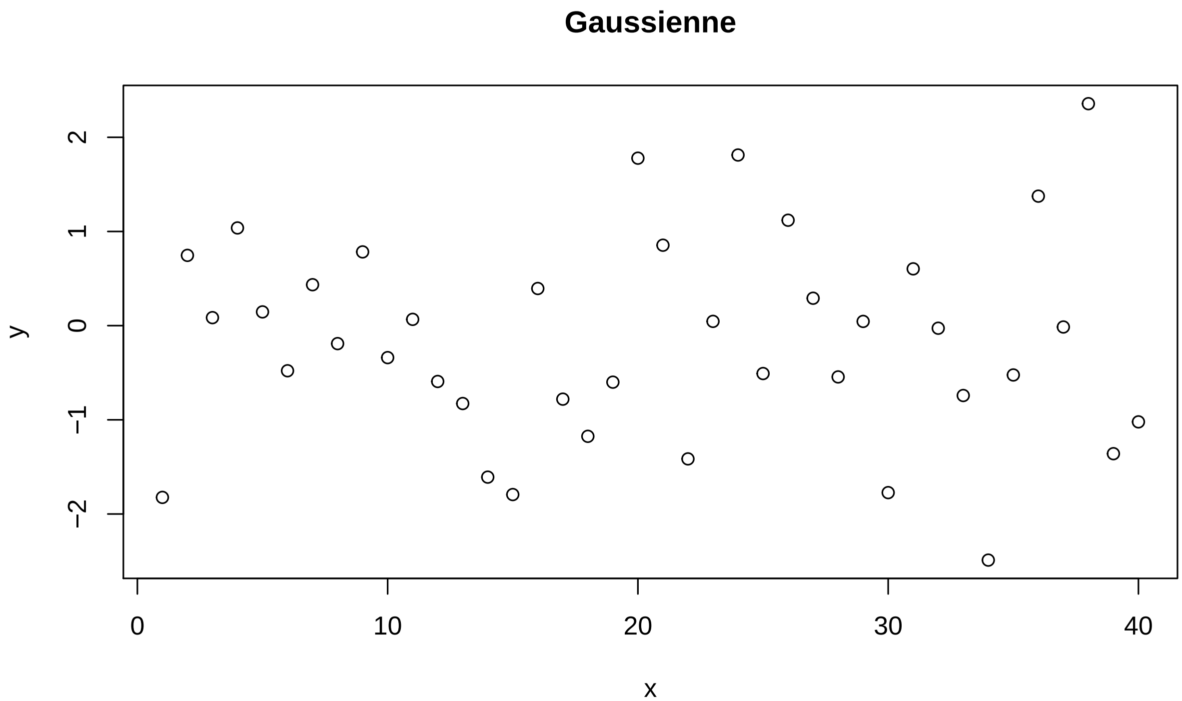
Traçons la distribution inconnue:

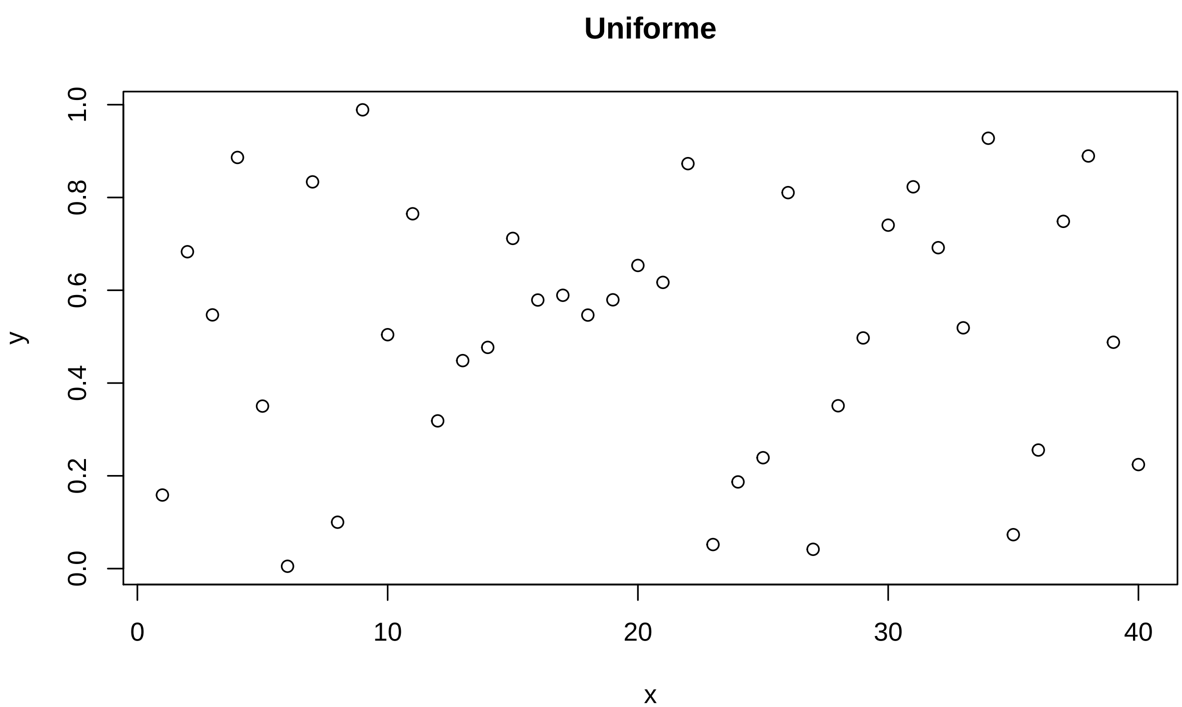
```
tracer <- function(df, xrow, yrow) {
  x <- unlist(df[xrow])
  y <- unlist(df[yrow])
  plot(x, y, main=yrow)
}

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

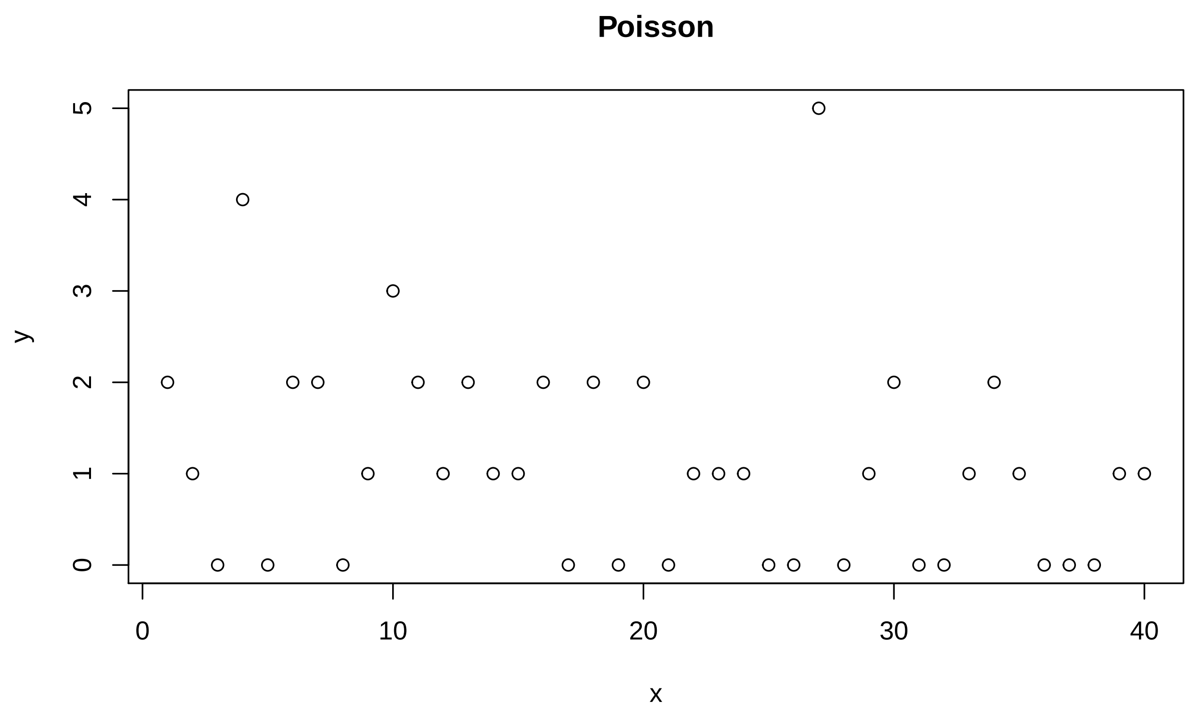


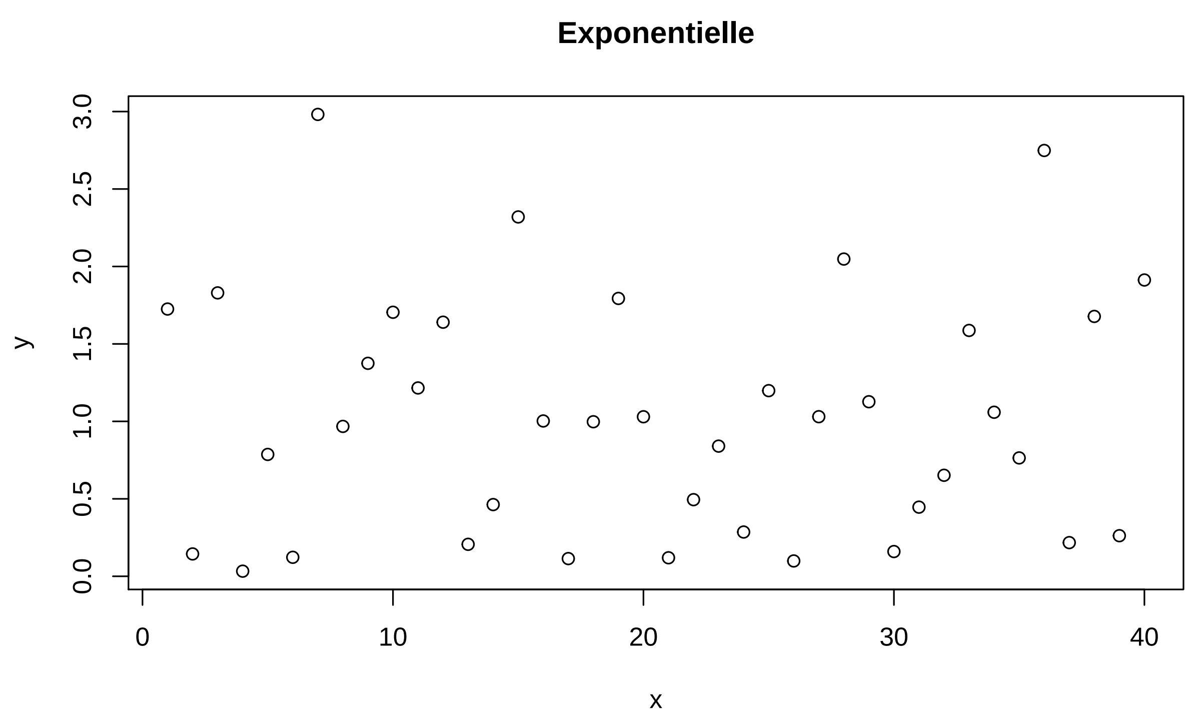
... afin de les comparer avec les distributions générés précdémment:

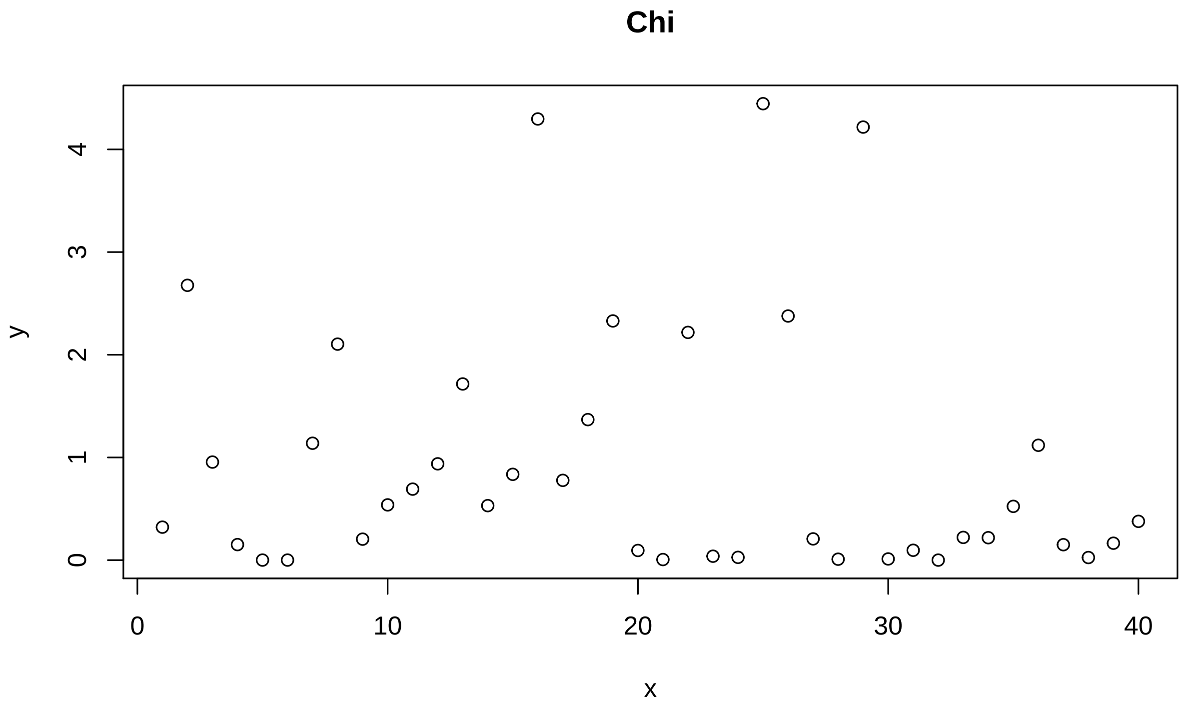
```
distributions <- c("Gaussienne", "Uniforme", "Poisson", "Exponentielle", "Chi", "Binomiale", "Cauchy");
for (distri in distributions) {
  tracer(df, "X", distri);
}
```

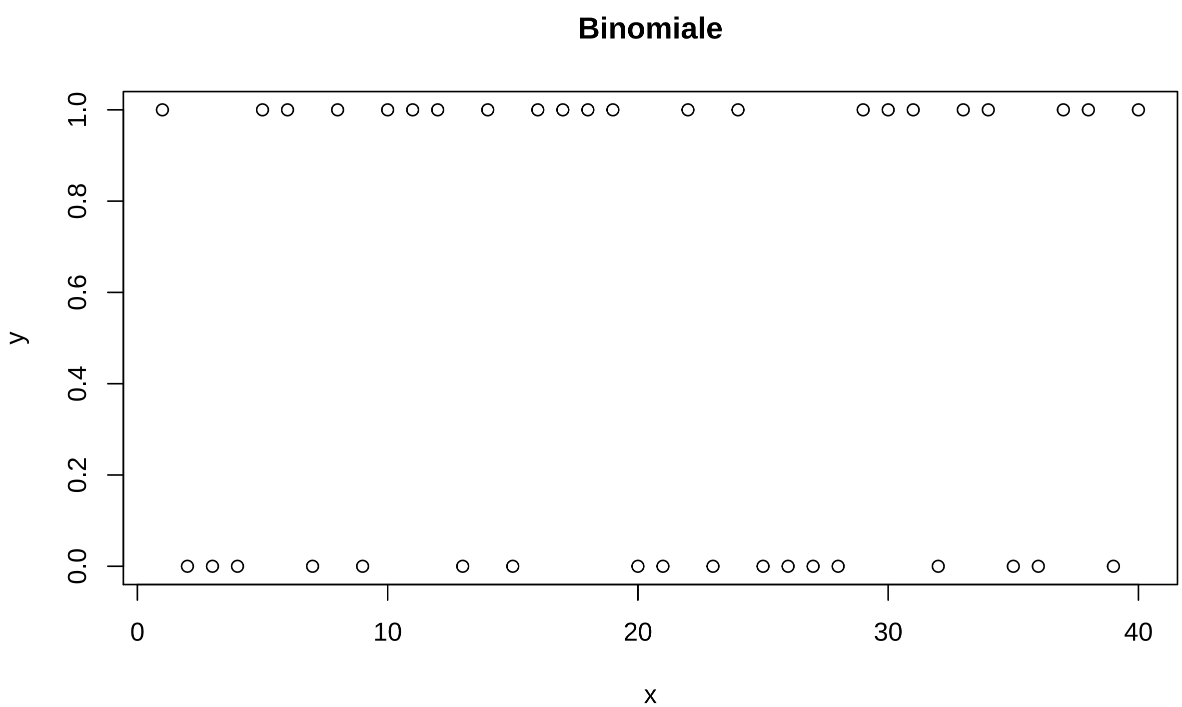

Gaussienne
y
x
0
10
20
30
40
−2
−1
0
1
2

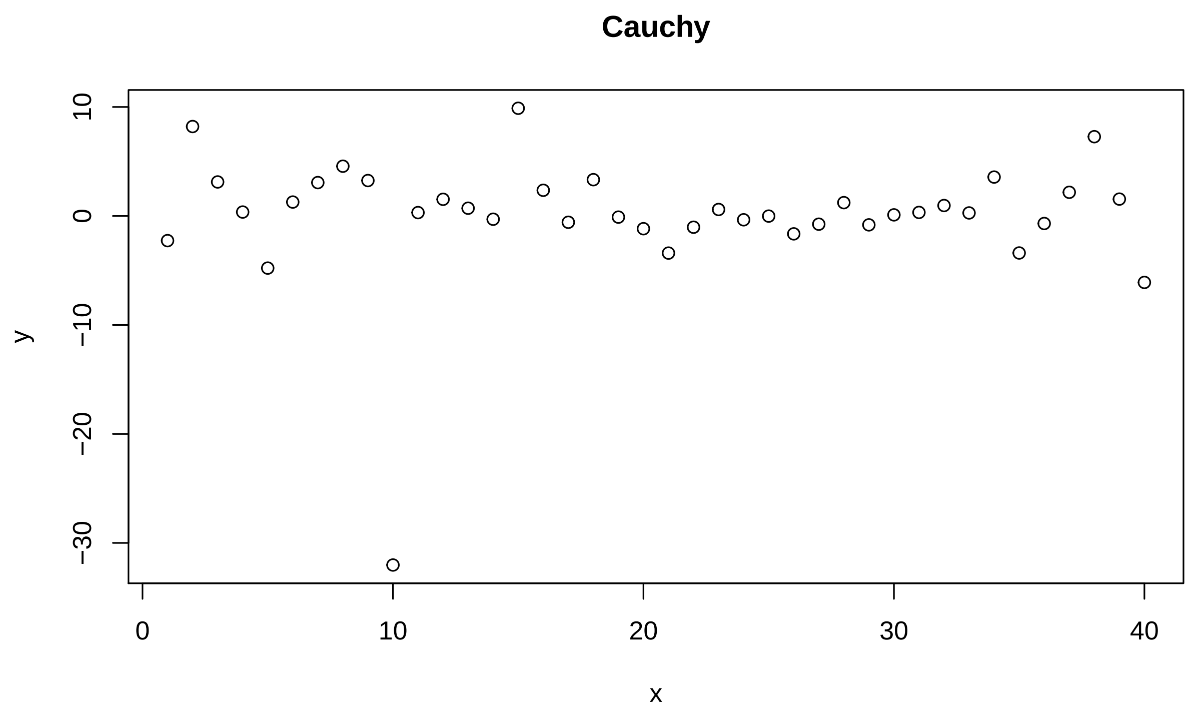

Uniforme
y
x
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0


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


Exponentielle
y
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
10
20
30
40
x


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


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

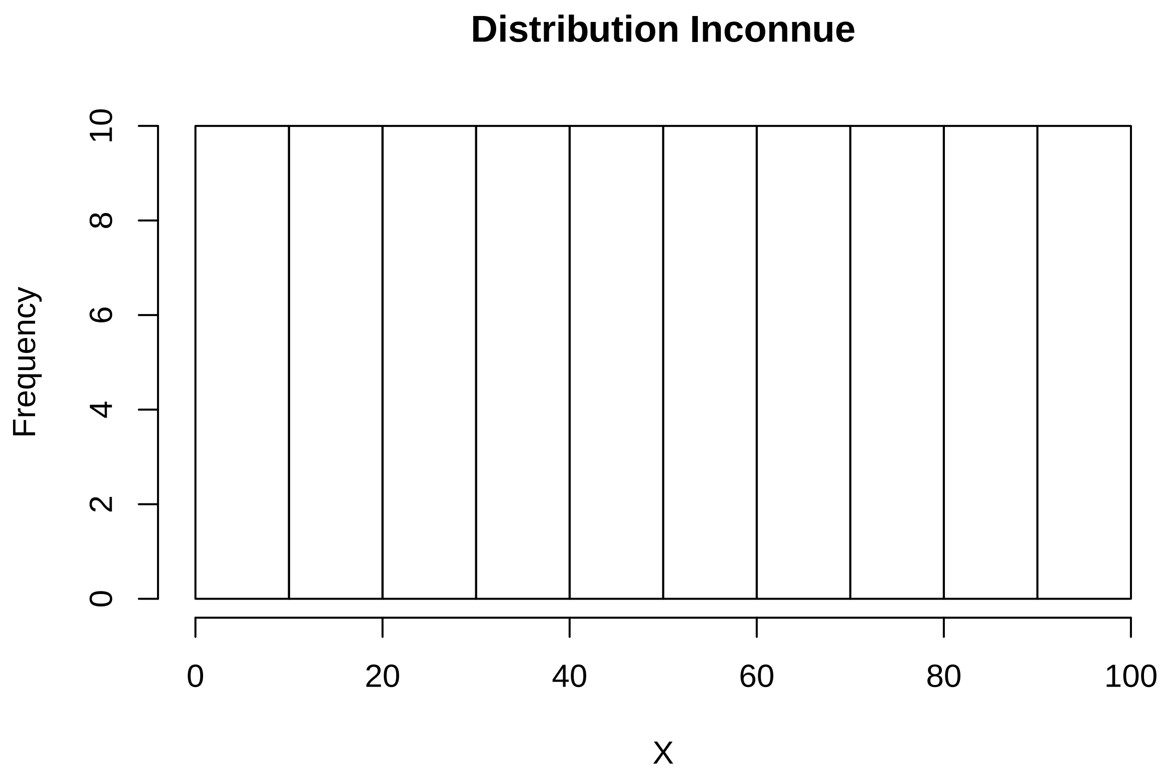


On remarque qu'il semble plus probable que la distribution ne semble à priori pas suivre une loi N(0, 1), mais plutot une loi uniforme dans [0, 100]

Histogramme

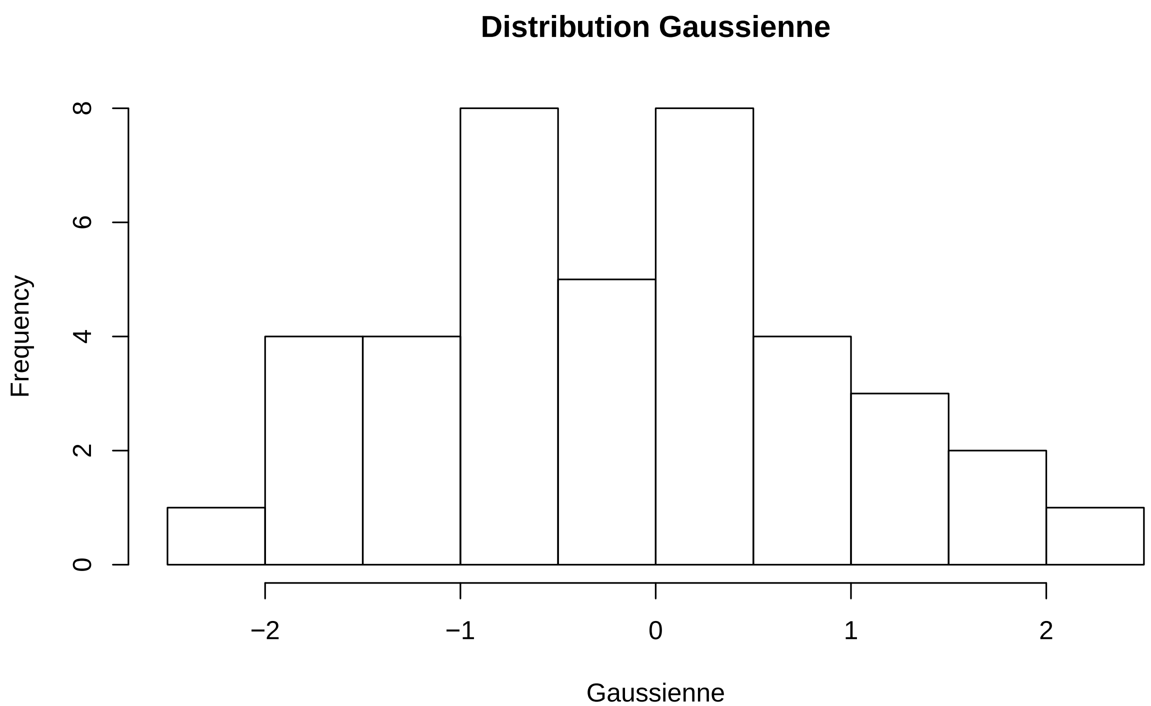
Je définis une fonction qui trace l'histogramme correspondant à la colonne 'row' du dataframe 'df'

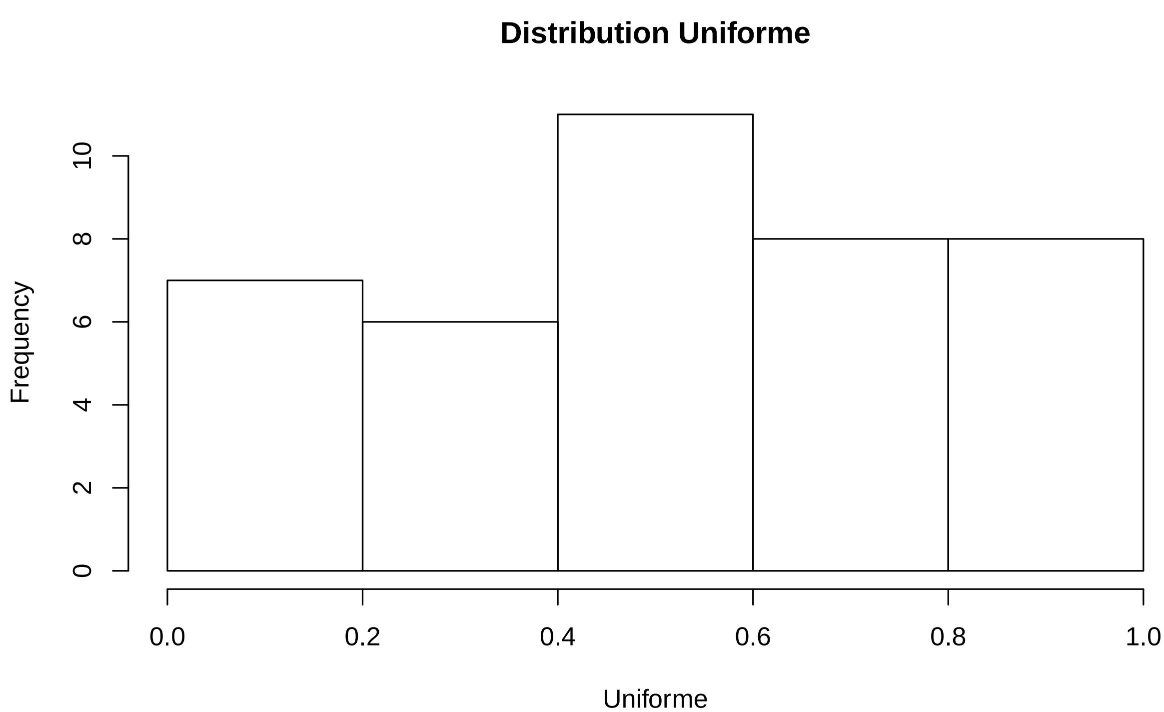
```
tracer <- function(df, row, title) {
  hist(unlist(df[row]), breaks="Sturges", xlab=row, main=paste("Distribution", title));
}
tracer(df_inconnu, "X", "Inconnue");
```

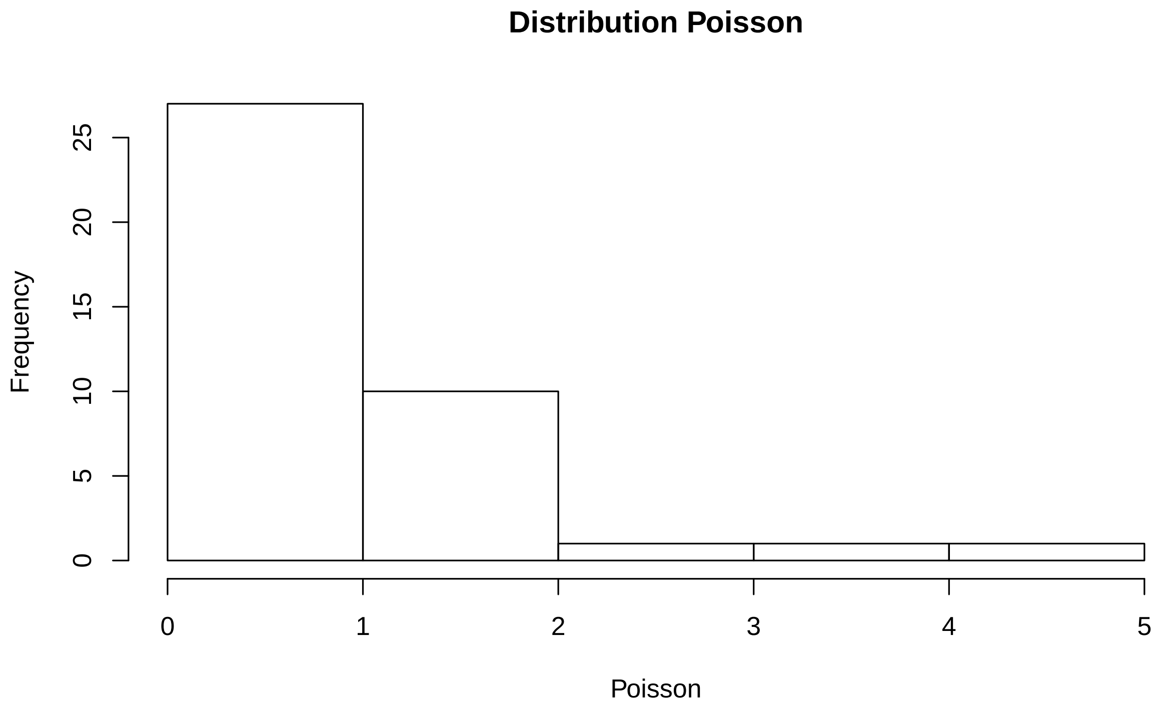


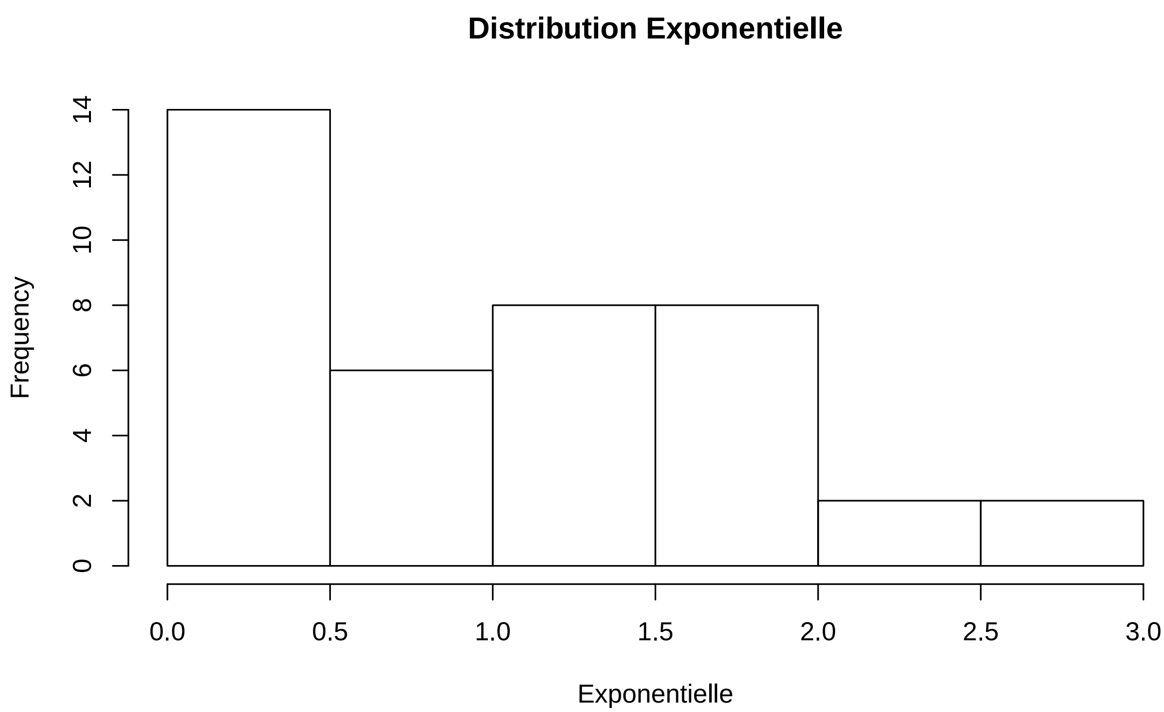
L'histogramme semble indiqué que la distribution inconnue suit une loi uniforme.

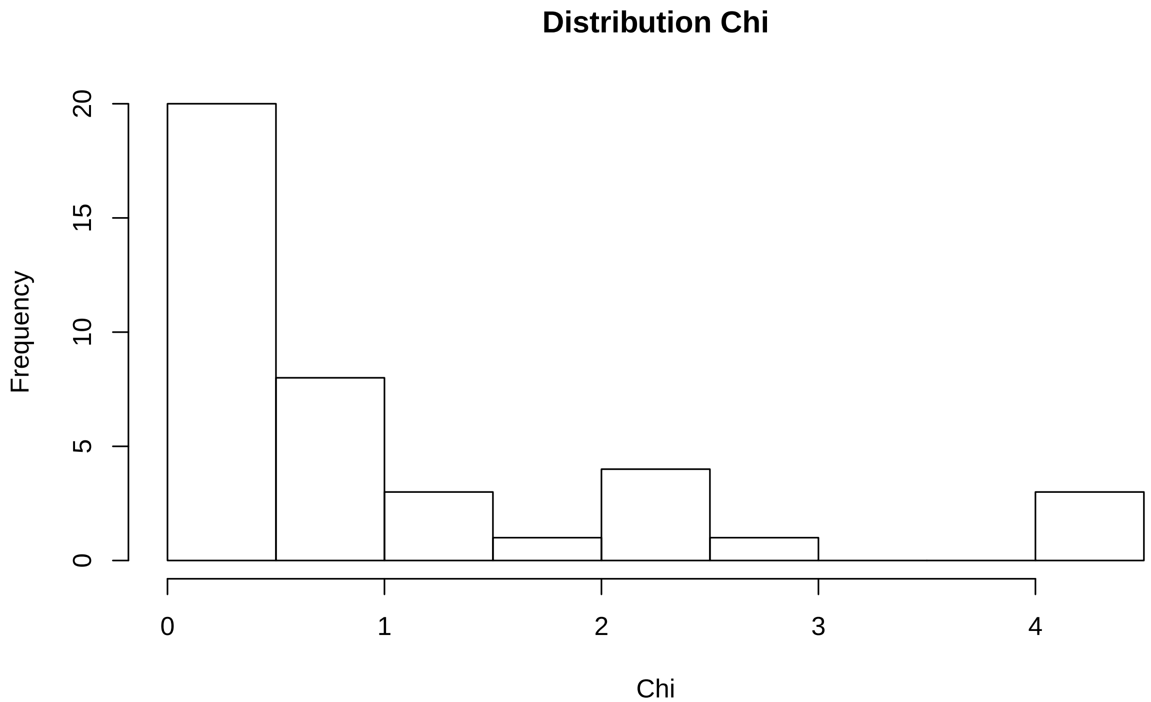
```
for (distri in distributions) {
  tracer(df, distri, distri);
}
```

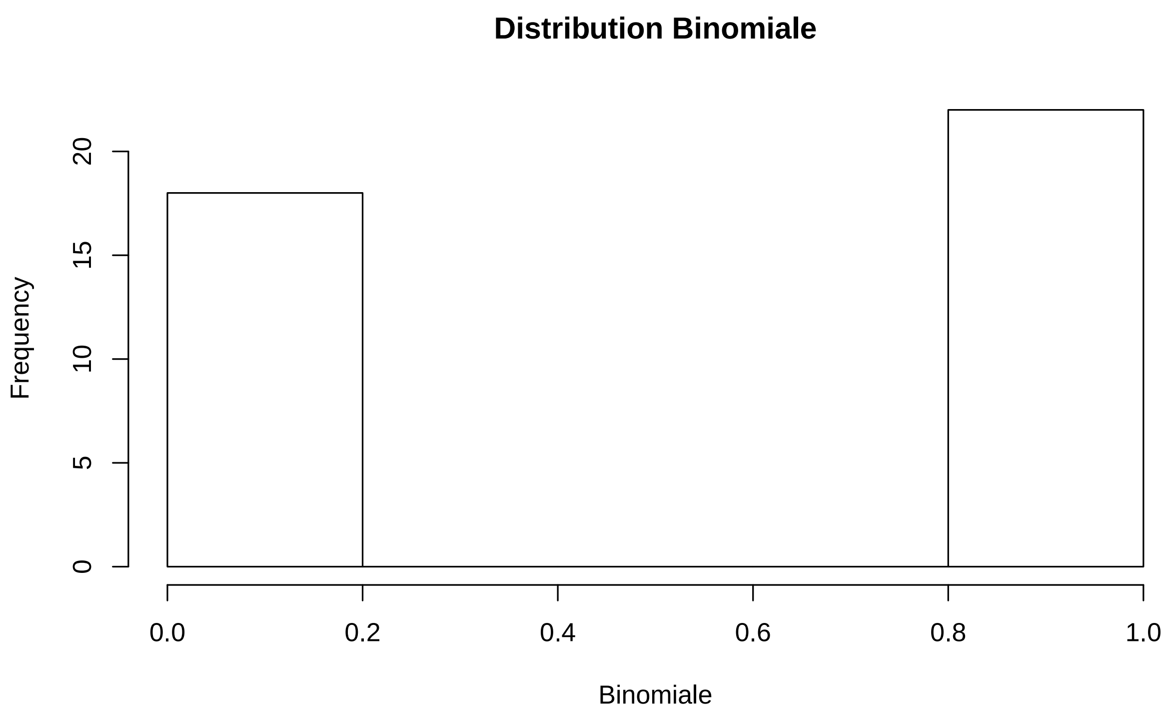


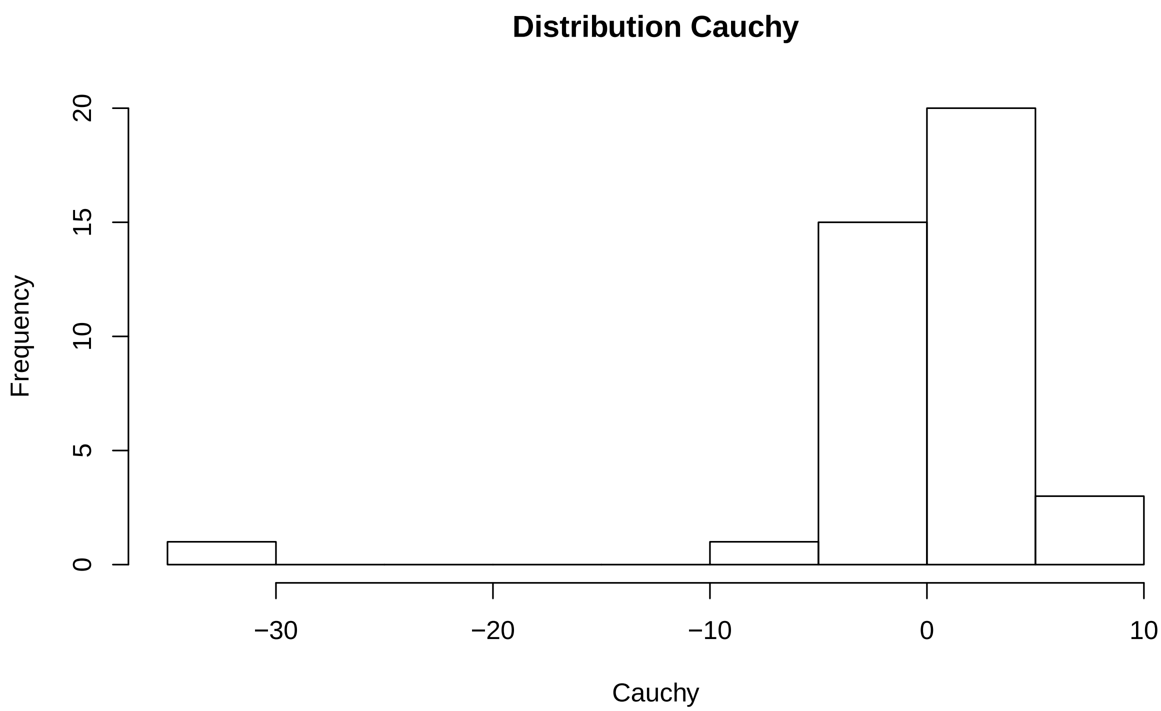



Distribution Poisson
Frequency
0
5
10
15
20
25
0
1
2
3
4
5
Poisson


Distribution Exponentielle
Frequency
0
2
4
6
8
10
12
14
0.0
0.5
1.0
1.5
2.0
2.5
3.0
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
20
−30
−20
−10
0
10
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.

```r
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 50.500000 841.666667  0.00000   1.7998
## 2    Gaussienne -0.166404   1.188583  0.15065   2.7020
## 3      Uniforme  0.519329   0.077525 -0.26414   2.0036
## 4       Poisson  1.125000   1.342949  1.25612   4.8767
## 5 Exponentielle  1.029529   0.598326  0.60277   2.6590
## 6           Chi  0.952927   1.527295  1.60776   4.7278
## 7      Binomiale  0.550000   0.253846 -0.20101   1.0404
## 8        Cauchy  0.014187  36.770937 -3.62654  20.8925
```

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.

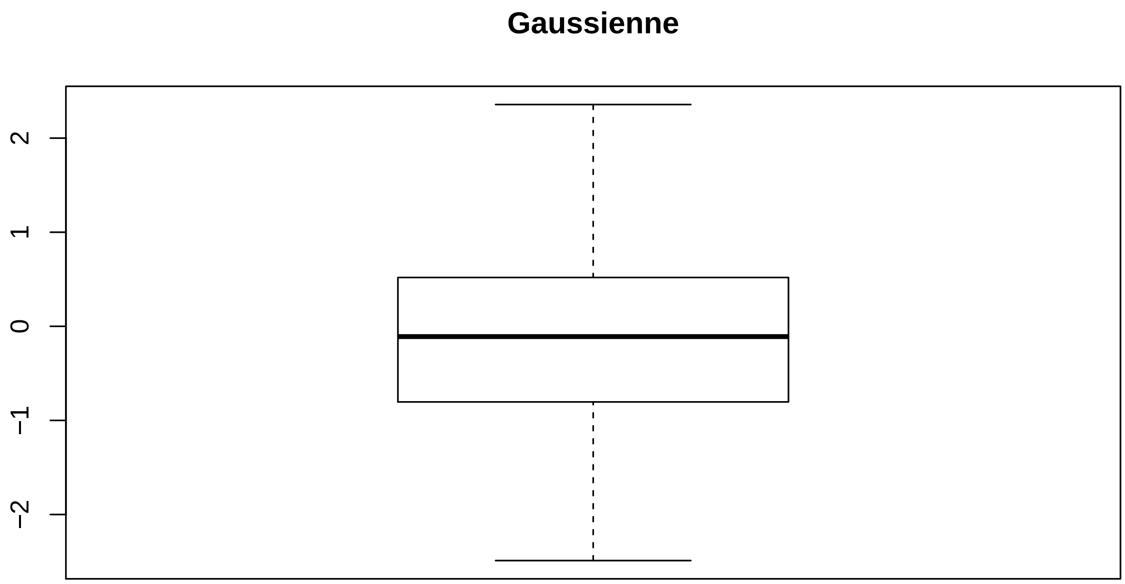
NB: Les Kurtosis utilisés sont 'non-normalisé' : j'ai ajouté '+ 3' aux valeurs théoriques normalisés ("Excess Kurtosis").

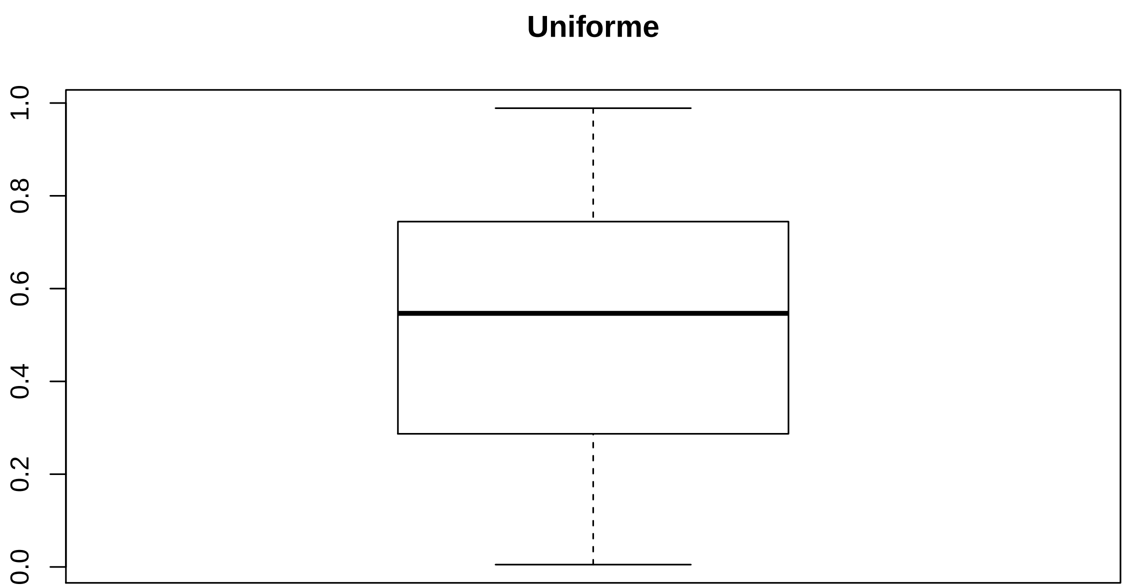
Les résultats obtenus suivent les valeurs théoriques des différents moments, mais peuvent parfois s'en éloigner selon les échantillons générés.

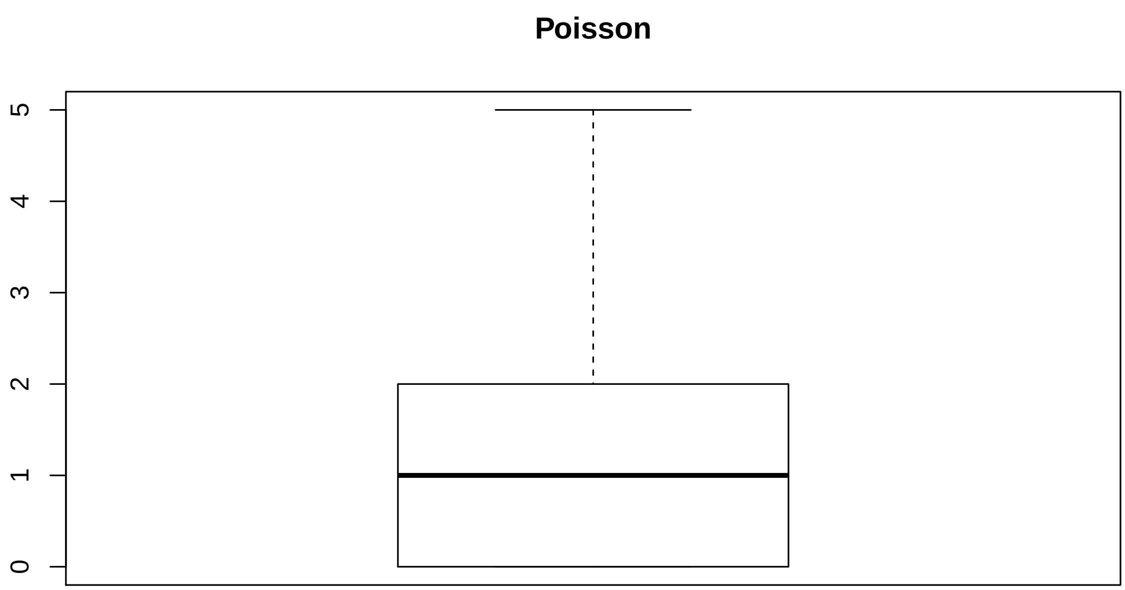
L'hypothèse précèdente semble d'autant plus probable car les Kurtosis de la distribution inconnue sont égal à ceux d'une distribution uniforme.

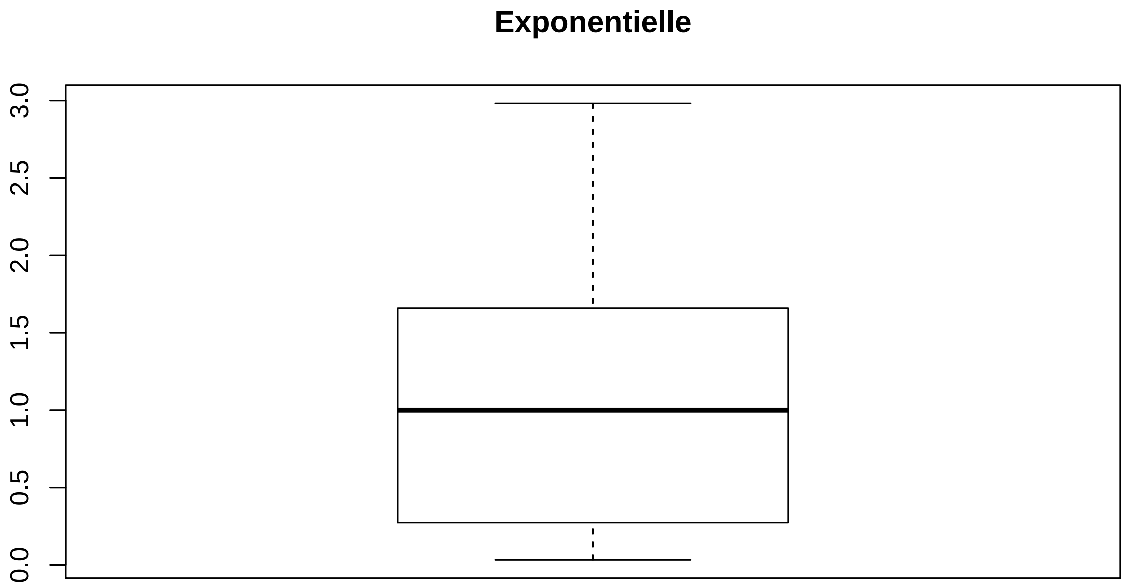
Quantiles et Boxplot

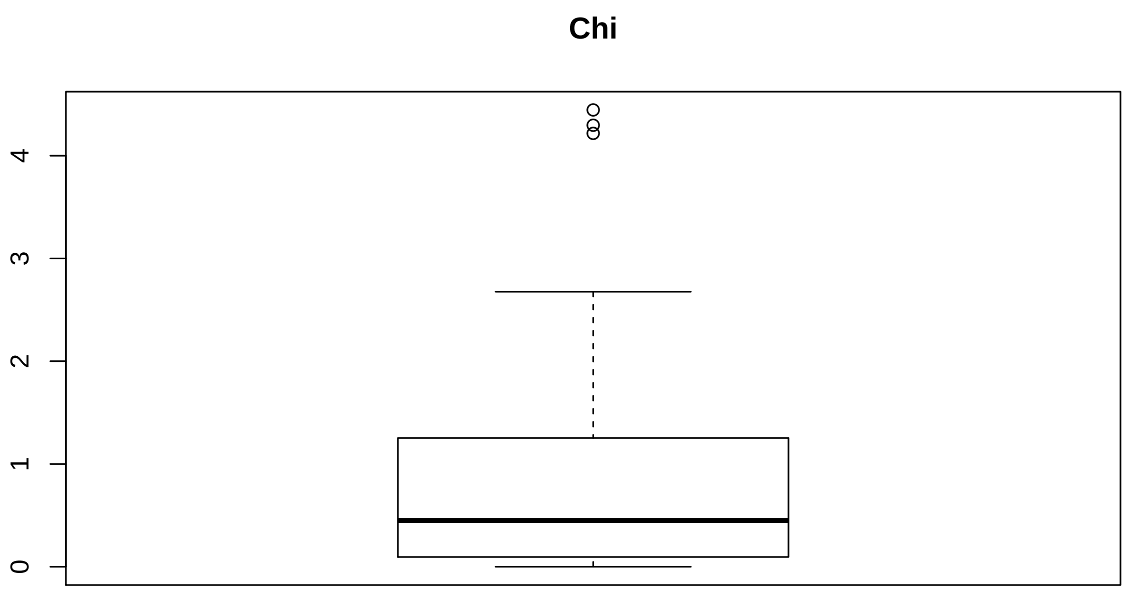
```r
for (distri in distributions) {
  x <- unlist(df[distri])
  boxplot(x, main=distri)
}
```

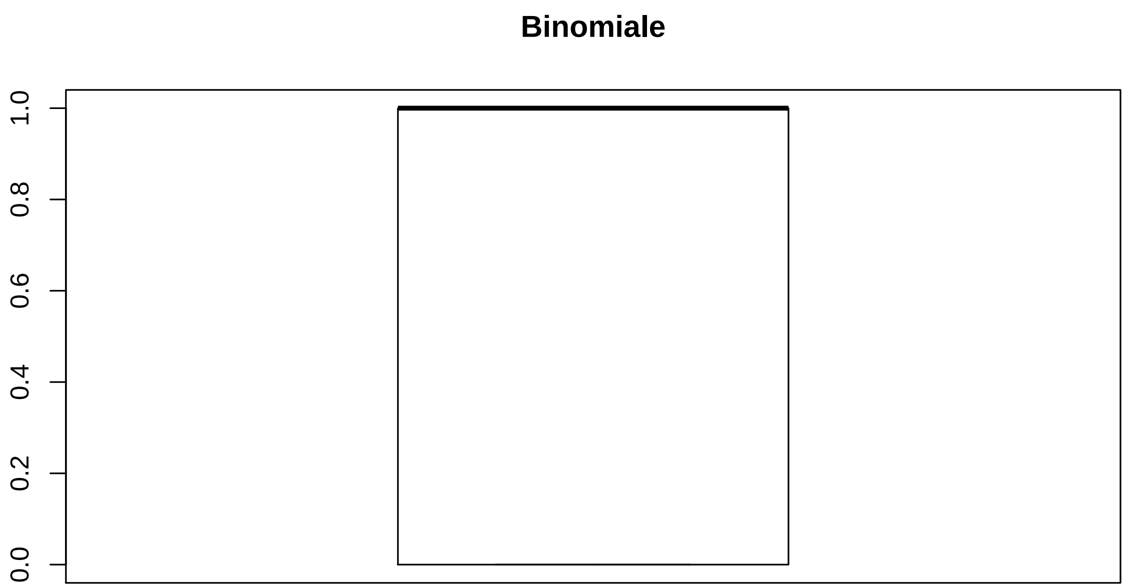


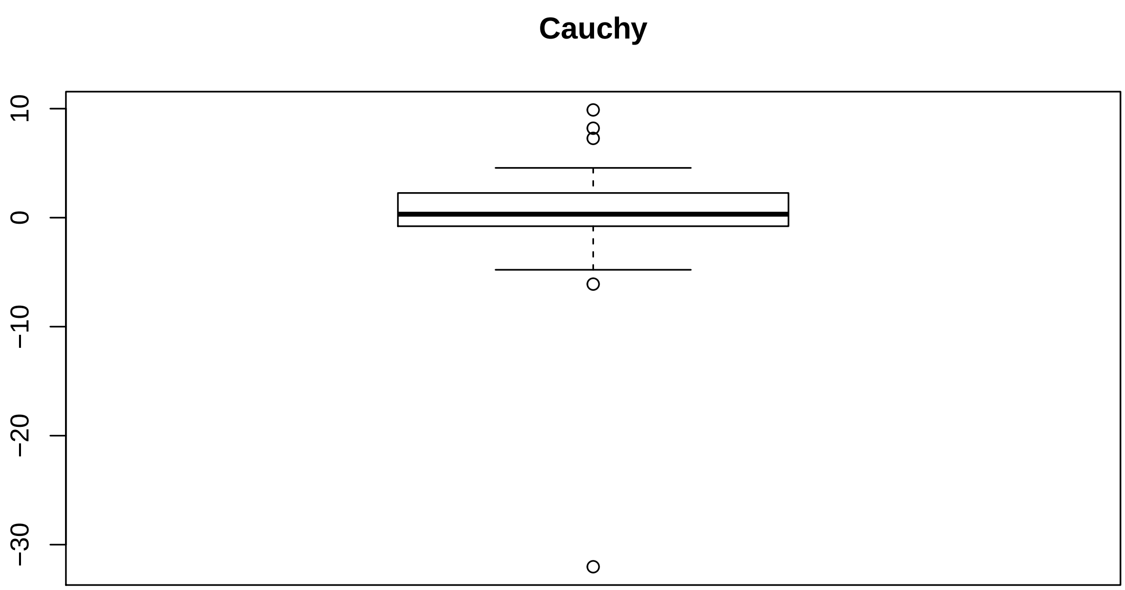

Uniforme
1.0
0.8
0.6
0.4
0.2
0.0


Poisson
5
4
3
2
1
0


Exponentielle
3.0
2.5
2.0
1.5
1.0
0.5
0.0


Chi
4
3
2
1
0






```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 50.500000 841.666667  0.00000   1.7998 25.75000 50.50000
## 2    Gaussienne -0.166404   1.188583  0.15065   2.7020 -0.79216 -0.10914
## 3      Uniforme  0.519329   0.077525 -0.26414   2.0036  0.30273  0.54668
## 4       Poisson  1.125000   1.342949  1.25612   4.8767  0.00000  1.00000
## 5 Exponentielle  1.029529   0.598326  0.60277   2.6590  0.27979  1.00010
## 6           Chi  0.952927   1.527295  1.60776   4.7278  0.09557  0.45070
## 7     Binomiale  0.550000   0.253846 -0.20101   1.0404  0.00000  1.00000
## 8        Cauchy  0.014187  36.770937 -3.62654  20.8925 -0.77044  0.31737
##         Q3
## 1 75.25000
## 2  0.47671
## 3  0.74235
## 4  2.00000
## 5  1.64952
## 6  1.19588
## 7  1.00000
## 8  2.21698
```

Interpretation visuelle

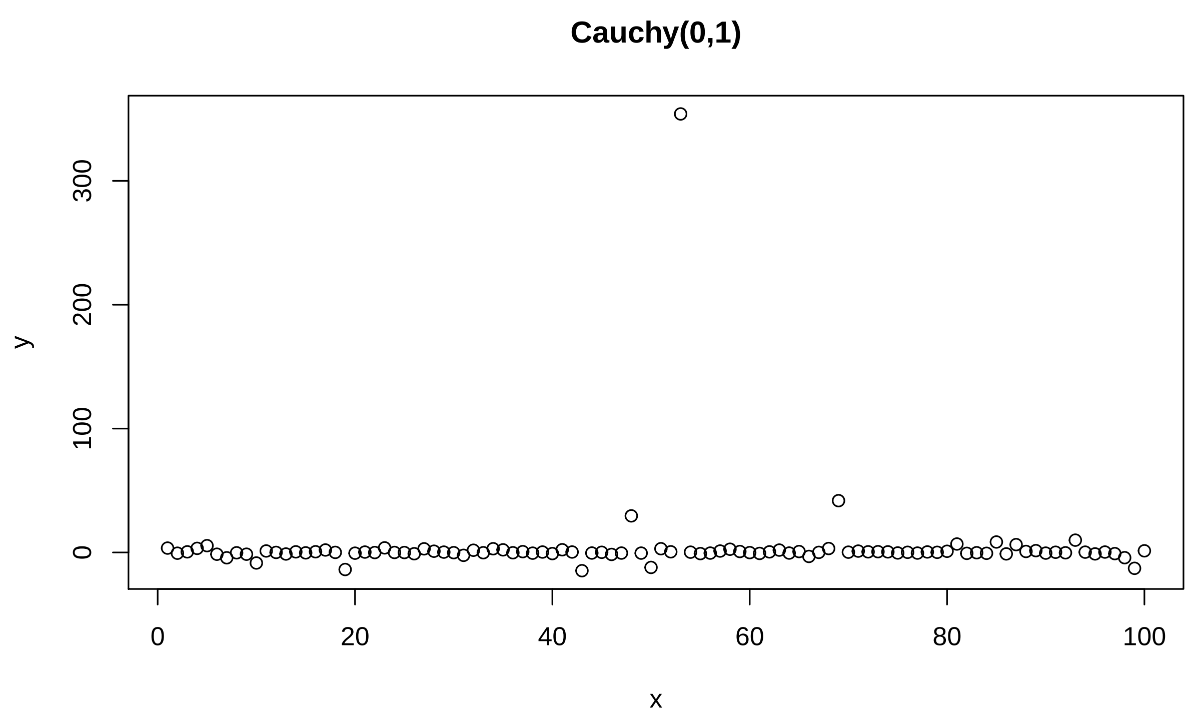
```
# genere des distriutions de cauchy, avec n=100, et (x0, a) dans {(0, 1), (1, 1), (0, 2)}
n <- 100
params <- list(c(0, 1), c(50, 1), c(0, 4))

# 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)
```

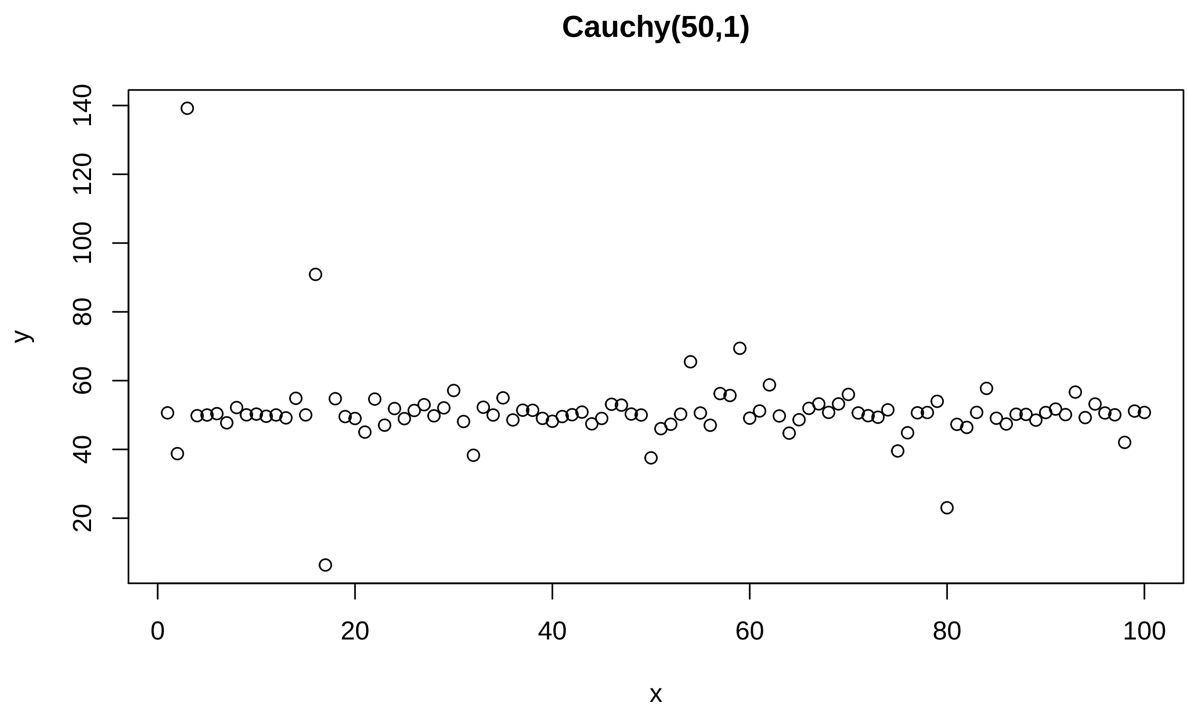
```
##       Cauchy(0,1) Cauchy(50,1)   Cauchy(0,4)
## 1      3.530707013    50.628515   2.86549833
## 2     -0.594255657    38.773577  -9.14016989
## 3      0.503762622   139.199003   0.50600279
## 4      3.314124770    49.806133   4.70090432
## 5      5.500486857    50.001939   1.75289474
## 6     -1.423485958    50.398496 -12.17484965
## 7     -4.279554835    47.734653  -6.65530550
## 8     -0.253155706    52.170746  75.46420606
## 9     -1.391817043    50.047809  -1.20156132
## 10    -8.485618449    50.266968  -0.21304851
## 11     1.203101701    49.637594  29.14383502
## 12     0.029201544    50.011479  -6.91587781
## 13    -1.243594751    49.170956  27.51569288
## 14     0.507912016    54.861819  -9.49215535
## 15    -0.305298450    50.035876  -5.20873298
## 16     0.574129617    90.890374  -1.99118846
## 17     2.006360944     6.385169  -8.68816685
## 18     0.050527554    54.740190   2.58425384
## 19   -13.760334510    49.549828  -3.71338816
## 20    -0.570331432    48.989787   2.74146945
## 21     0.322793775    45.036247  13.35968751
## 22    -0.075674930    54.632208   3.74693981
## 23     3.706571213    47.051565   0.26742431
## 24    -0.002203915    51.883112  33.58065994
## 25    -0.058724483    48.942587   1.89023917
## 26    -1.013545408    51.318842   2.10537067
## 27     2.934428749    53.001984   1.15472718
## 28     1.015774032    49.761718  47.70778375
```

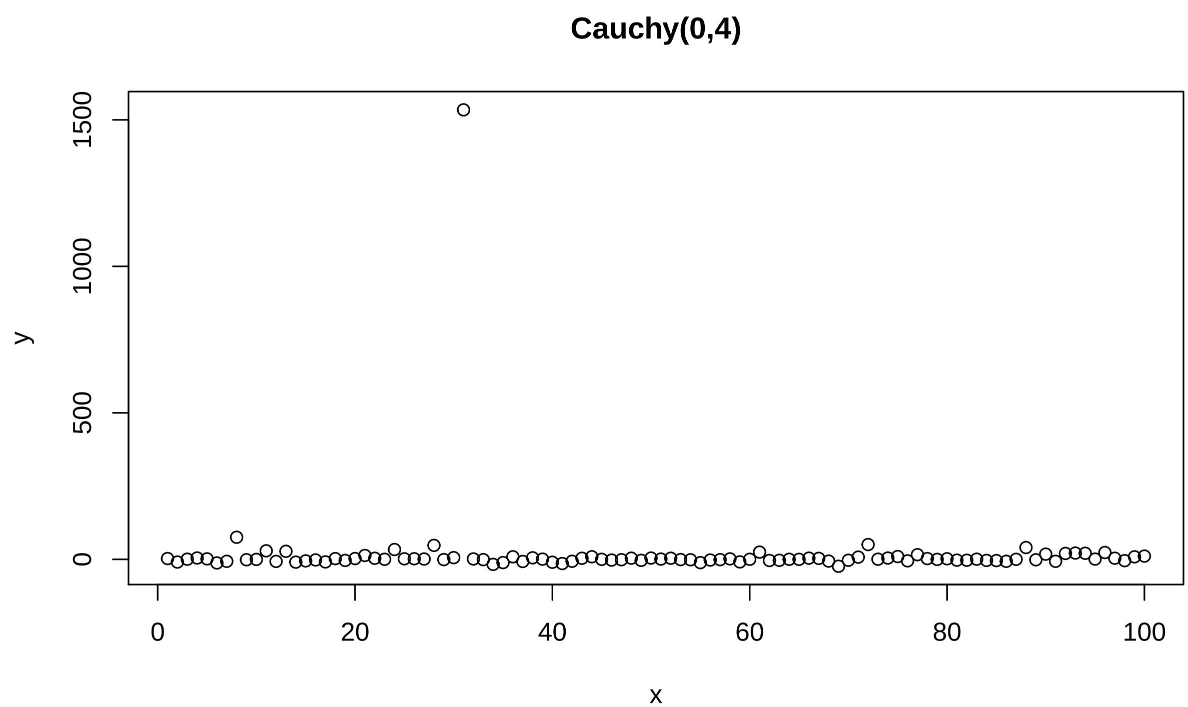
```
## 29    0.280791914    52.077891   -0.76519478
## 30   -0.160721494    57.131364    5.84504864
## 31   -2.309220108    48.111740 1534.24897532
## 32    1.799430490    38.298719    1.45529324
## 33   -0.209576597    52.262371   -1.04801443
## 34    3.011855921    50.001861  -17.16887878
## 35    2.128679579    54.979584  -11.00164422
## 36   -0.227670485    48.565915    8.86957911
## 37    0.666921228    51.403398   -7.24339140
## 38   -0.561280748    51.371225    5.28614485
## 39    0.360297892    49.022637    0.77434677
## 40   -0.913333330    48.178977   -9.81878580
## 41    2.194881906    49.551220  -14.34909078
## 42    0.365582890    50.086830   -6.48365185
## 43  -14.724472831    50.851560    3.69896227
## 44   -0.457661049    47.427391    8.89730292
## 45    0.148485059    49.004516    0.10993560
## 46   -1.588582112    53.124378   -2.52242263
## 47   -0.485472198    52.875751   -1.23801075
## 48   29.544954602    50.282771    4.05691304
## 49   -0.522526764    49.999307   -3.36291112
## 50  -12.081333032    37.538362    4.42366385
## 51    3.034358697    46.032867    0.91300912
## 52    0.539179640    47.316754    3.86252705
## 53  354.031667053    50.247891   -0.76990106
## 54    0.246416987    65.481698   -1.40200026
## 55   -0.927906028    50.581644  -11.24151014
## 56   -0.510554829    47.027652   -2.36028711
## 57    1.155571557    56.237430   -1.01331416
## 58    2.616725075    55.675041    1.63225529
## 59    0.769199497    69.403342   -8.63217288
## 60   -0.123227929    49.090176    0.25651994
## 61   -0.792495910    51.173699   24.70648099
## 62    0.492457496    58.760983   -3.88738303
## 63    1.921887765    49.726161   -3.20664724
## 64   -0.470468572    44.732529    0.27869556
## 65    0.741612775    48.624076    0.08608811
## 66   -3.205237105    51.933940    4.08414546
## 67    0.063738093    53.236217    3.37615416
## 68    3.196476620    50.761376   -5.86776586
## 69   41.814305347    53.226400  -23.59381017
## 70    0.239429705    55.999102   -3.05661133
## 71    1.100963496    50.602973    7.61915119
## 72    0.523329047    49.802330   50.56237223
## 73    0.634347775    49.353312    0.56231595
## 74    0.487348701    51.500440    4.44071705
## 75   -0.400954077    39.539804    9.64114783
## 76    0.146323660    44.838422   -5.10076981
## 77   -0.495912211    50.642915   15.99411802
## 78    0.416855065    50.733167    2.61058349
## 79    0.133243105    53.988896   -0.13383292
## 80    0.973559402    23.028654    1.99959684
## 81    6.829263052    47.286275   -2.69708556
## 82   -0.709585041    46.396488   -2.98574839
```

```
## 83   -0.204144749    50.758592    0.58329796
## 84   -0.671325161    57.746357   -3.63388641
## 85    8.454107243    49.070799   -4.15846399
## 86   -1.145392676    47.387534   -6.67061177
## 87    6.346729732    50.227469    0.13435509
## 88    0.802654569    50.177345   40.32069835
## 89    1.464981321    48.512858   -1.48953993
## 90   -0.520423637    50.722833   17.74410531
## 91    0.176977892    51.725345   -6.61694110
## 92   -0.266373275    50.115504   20.24406041
## 93    9.938508640    56.666116   21.49697553
## 94    0.262829572    49.252704   20.72339993
## 95   -1.145879664    53.195861    0.87556289
## 96    0.339770044    50.588087   23.37430264
## 97   -0.930905793    50.057281    3.87284815
## 98   -4.223045416    42.044409   -4.44797935
## 99  -12.791548761    51.147776    8.69284525
## 100   1.338687644    50.744782   11.11348361
```

```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)
}
```

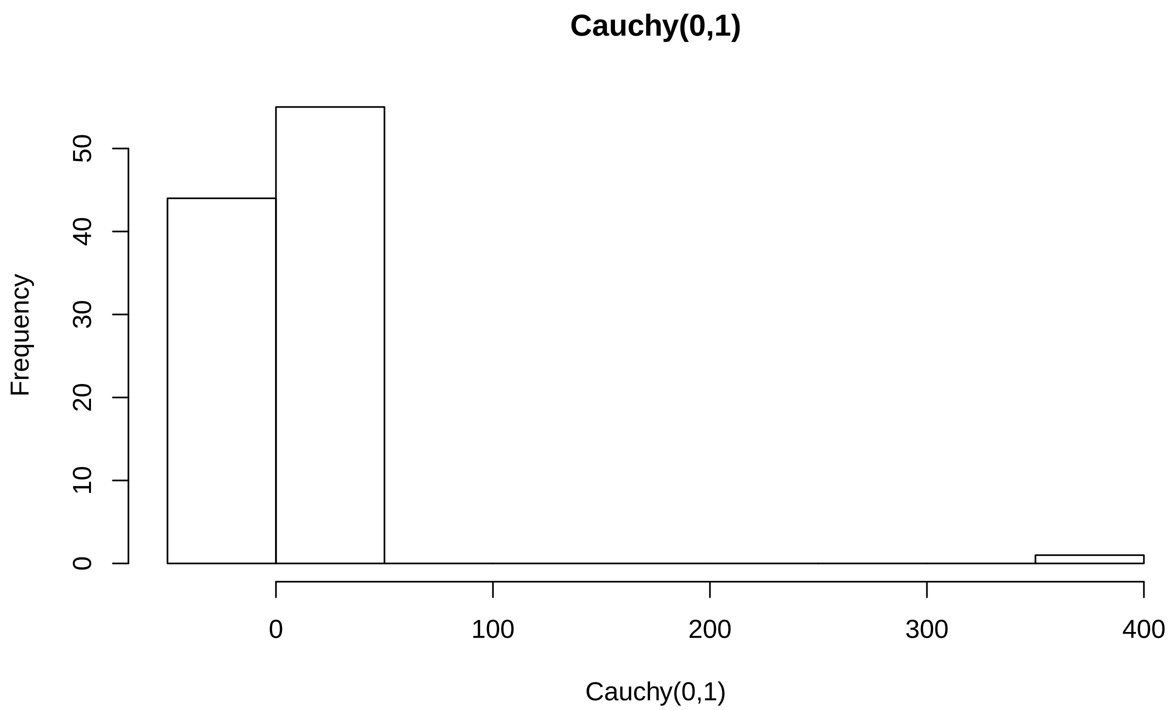




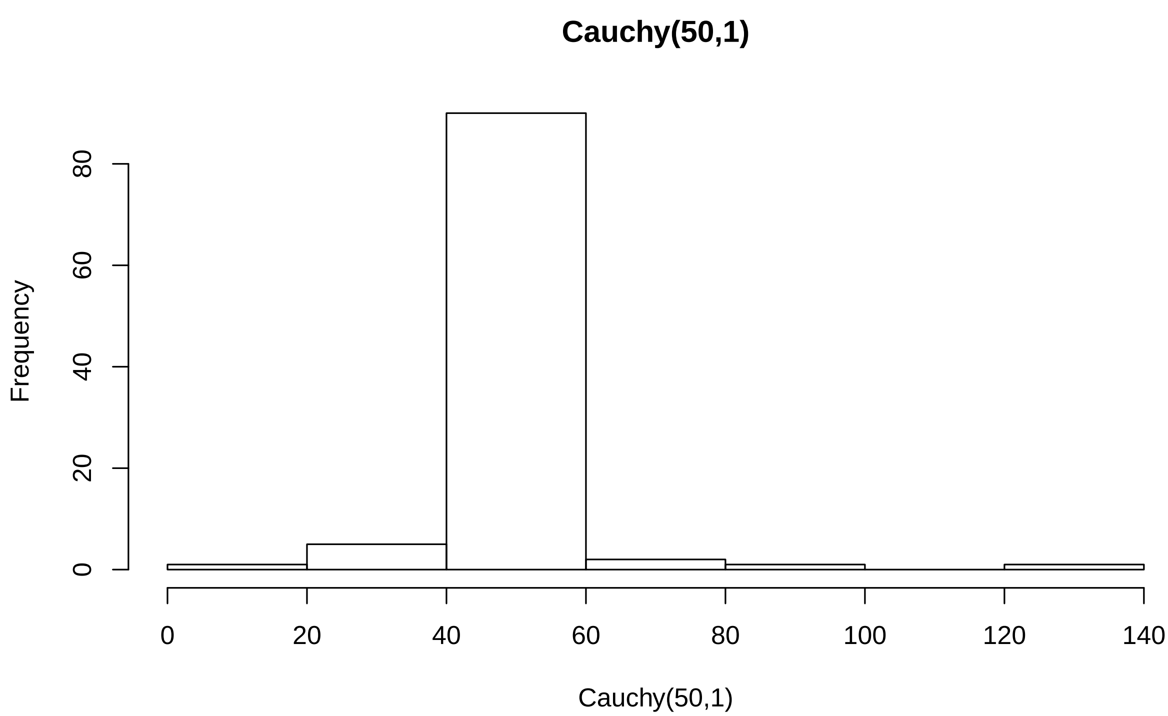



```r
# trace les distributions en histogramme
for (nom in noms) {
  y <- unlist(df[nom])
  hist(y, breaks="Sturges", xlab=nom, main=nom)
```

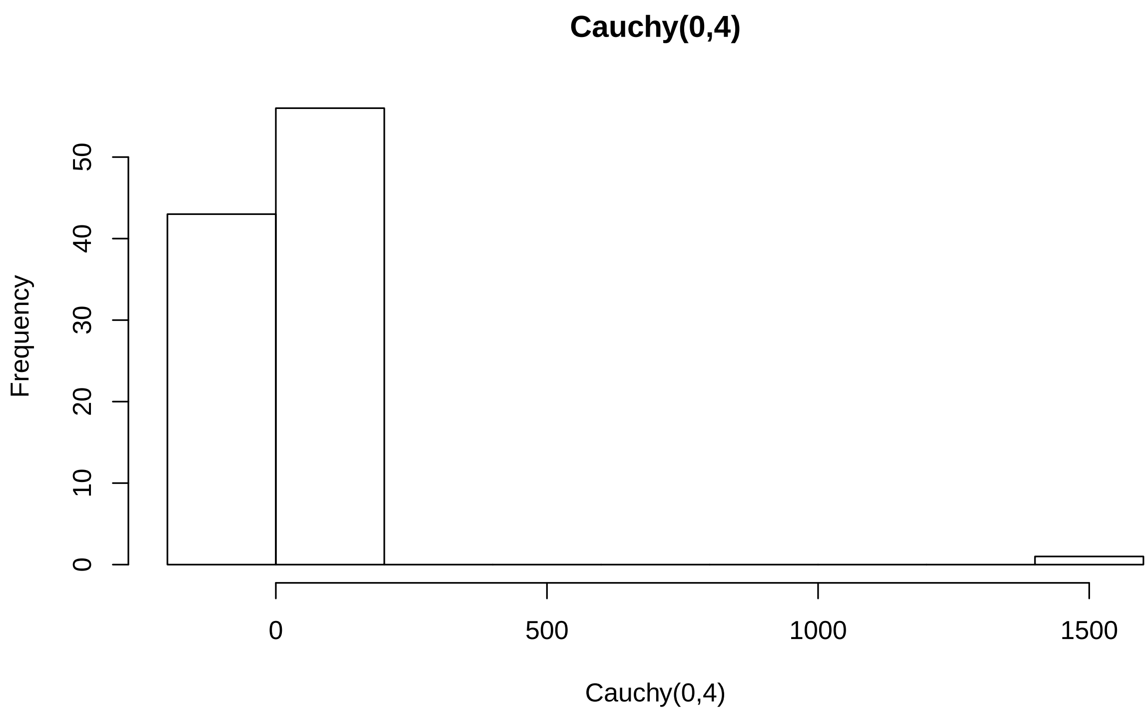
```
}
```



Cauchy(0,1)
Frequency
0
10
20
30
40
50
0
100
200
300
400
Cauchy(0,1)



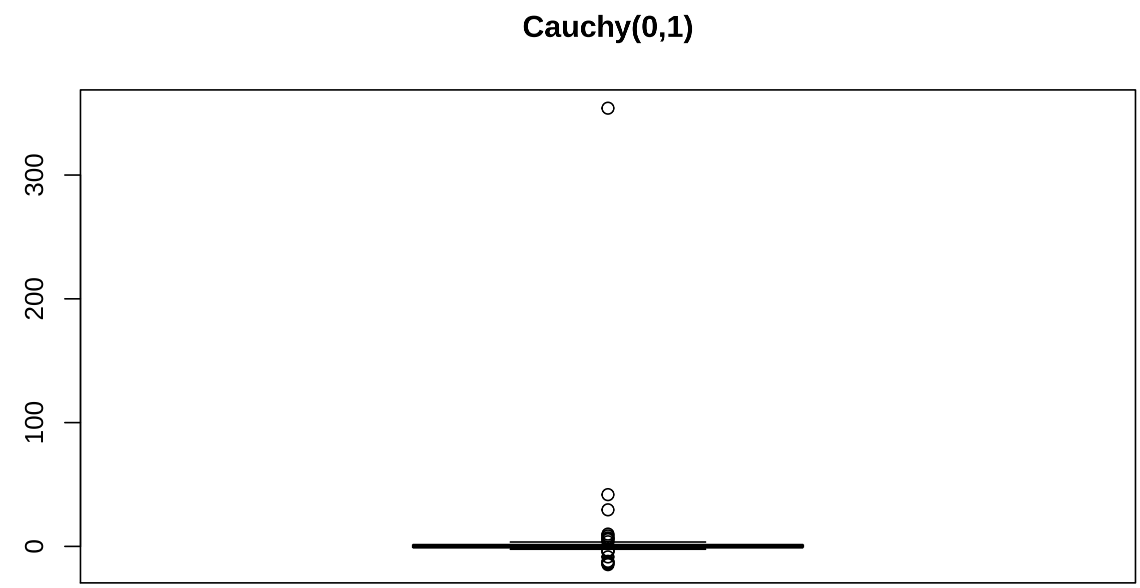
Cauchy(50,1)
Frequency
0
20
40
60
80
0
20
40
60
80
100
120
140
Cauchy(50,1)



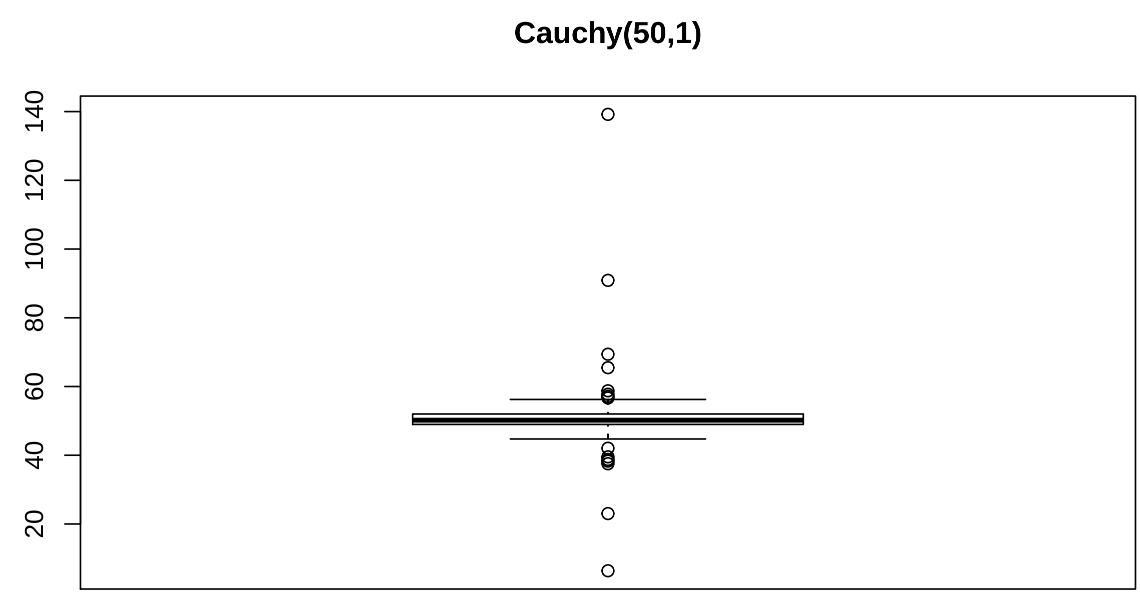

```r
# genere les moments et les quantiles
library("moments")
ajouter_ligne <- function(matrice, valeurs, nom) {
  x <- unlist(valeurs)
  Q <- quantile(x, c(0.25, 0.5, 0.75))
  m  <- data.frame(nom, mean(x), var(x), skewness(x), kurtosis(x), Q[[1]][1], Q[[2]][1], Q[[3]][1])
  names(m) <- c("Distribution", "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)       4.2     1287     9.42     92.2 -0.532  0.163  1.11
## 2 Cauchy(50,1)      51.0      142     3.83     34.2 48.978 50.202 51.97
## 3  Cauchy(0,4)      18.9    23632     9.72     96.3 -3.431  0.534  4.51
```

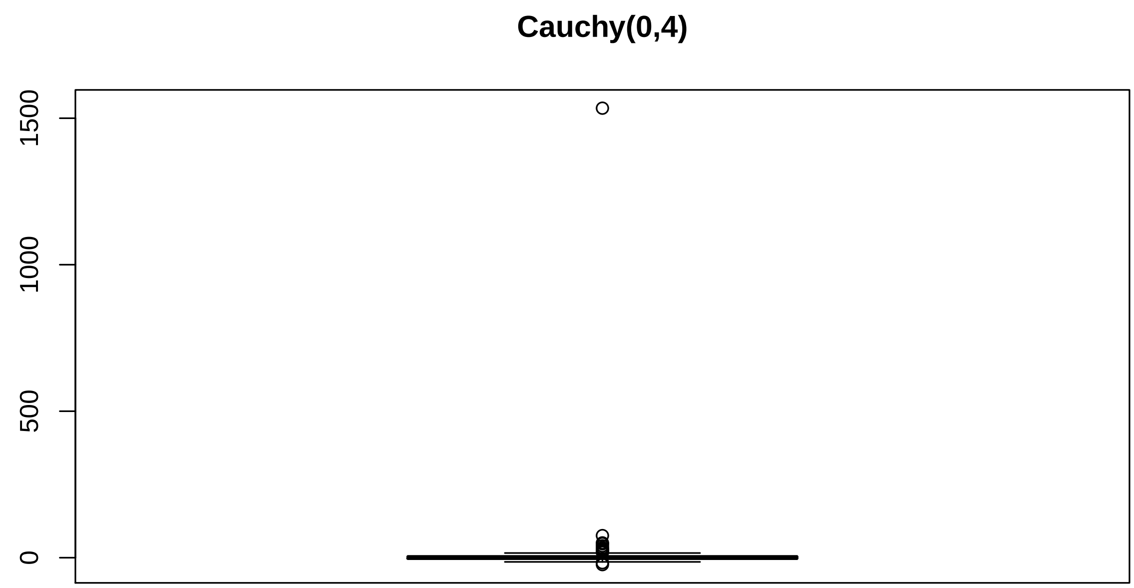
```r
# trace les distributions avec boxplot
for (nom in noms) {
  y <- unlist(df[nom])
  boxplot(y, main=nom)
}
```



Cauchy(0,1)
300
200
100
0



Cauchy(50,1)
140
120
100
80
60
40
20



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.