

TP2 stats

Romain PEREIRA

5 Mars 2018

1. Echantillon, Théorème Central Limite, Estimation Monte Carlo

1.1 Simulation de 1000 échantillon i.i.d gaussien.

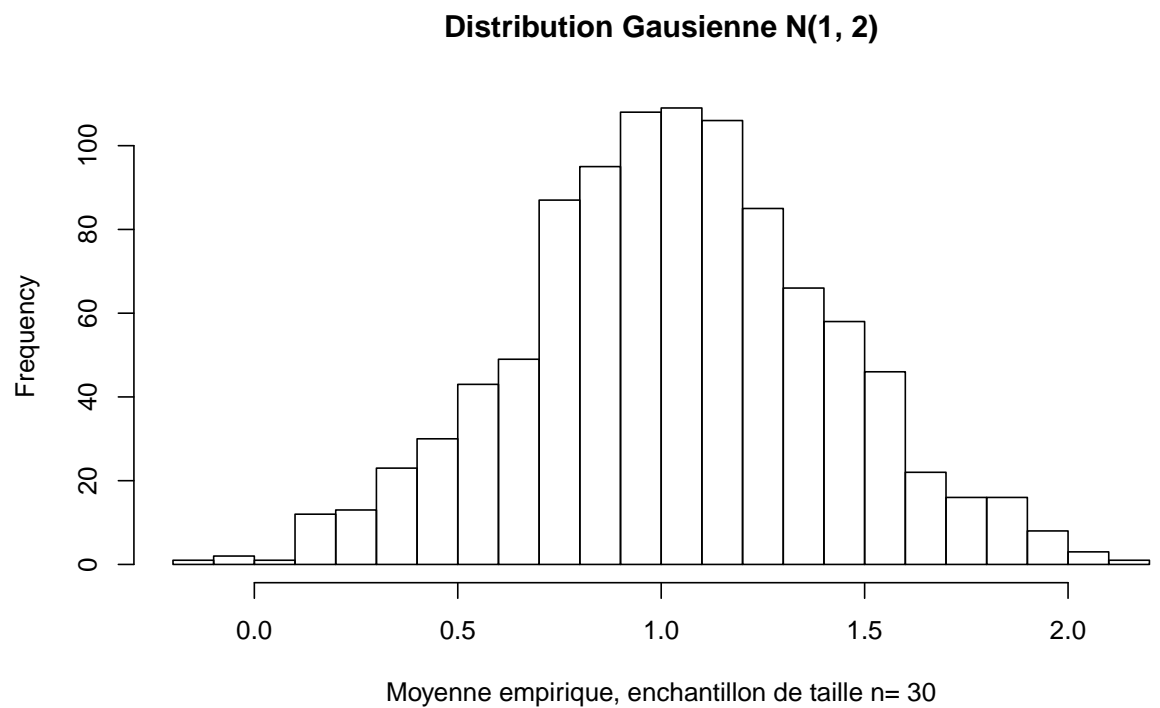
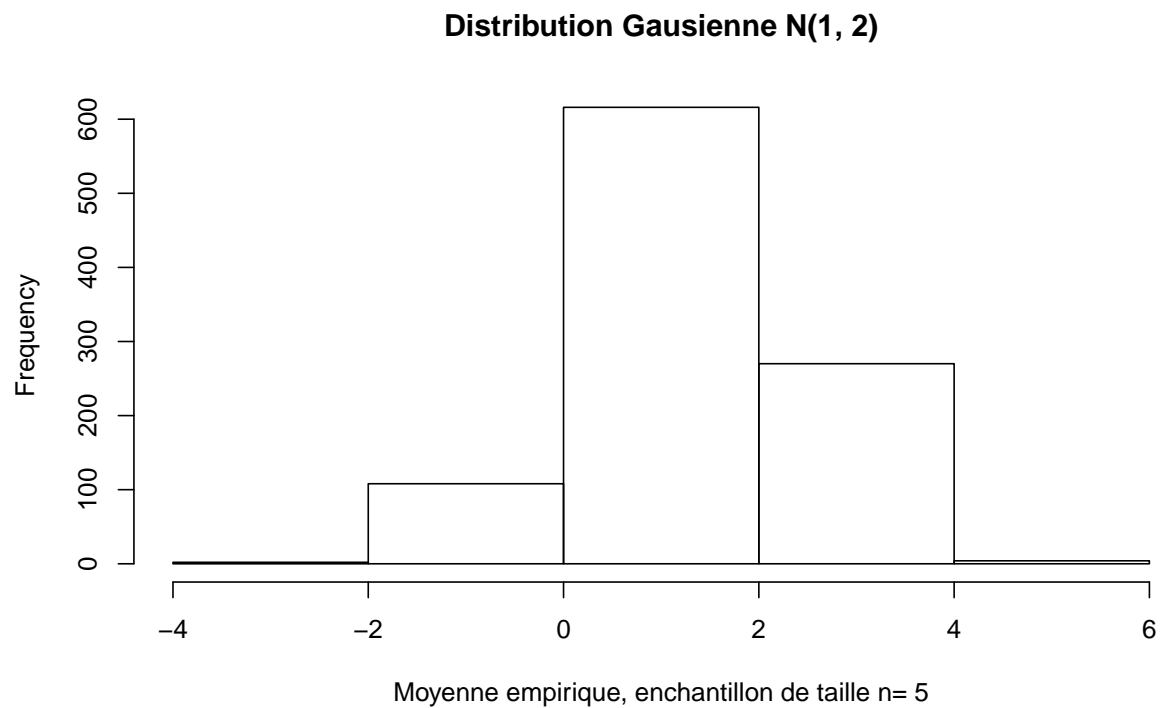
```
N <- 1000
n <- c(5, 30, 100)

empirical_mean <- function(vec) {
  s <- 0
  for (x in vec) {
    s <- s + x
  }
  return (s / (length(vec) - 1))
}

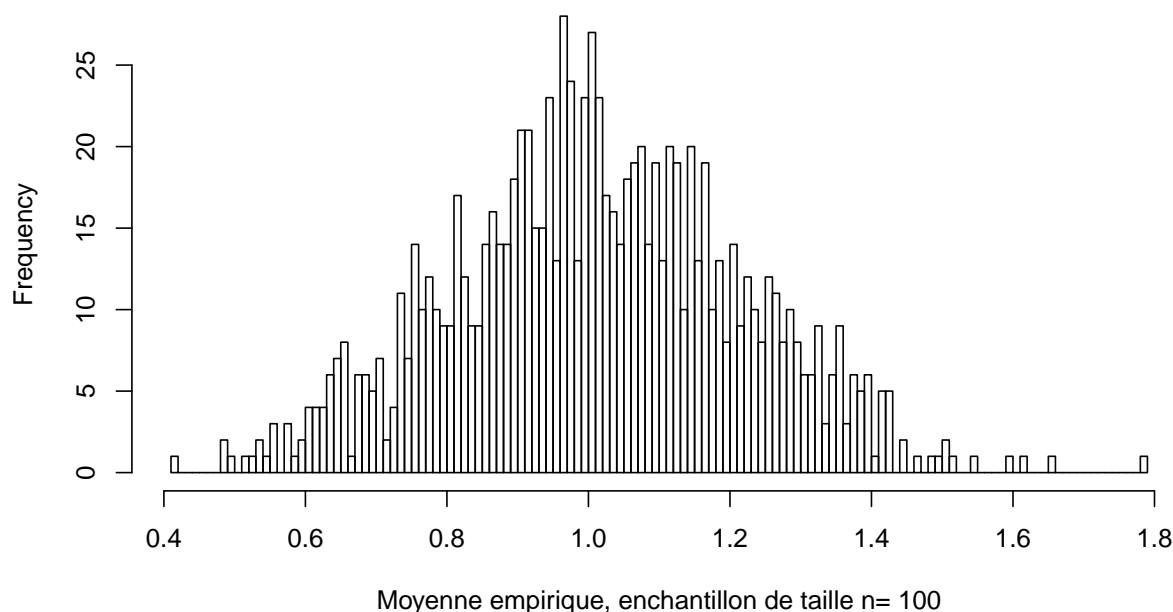
empirical_var <- function(vec) {
  m <- empirical_mean(vec)
  s <- 0
  for (x in vec) {
    s <- s + (x - m) * (x - m)
  }
  return (s / (length(vec) - 1))
}

# fonction 'mean_hist'
#
# 'law' : fonction qui genere un vecteur de taille 'm', e.g: law(42)
# 'title': titre de l'histogramme
#
# La fonction trace 3 histogrammes de la loi de la moyenne empirique
# sur 'N' echantillons de taille dans 'ns'
mean_hist <- function(law, title) {
  for (nj in n) {
    sample <- law(nj * N)
    means <- c()
    for (i in 1:N) {
      subsample <- sample[((i-1)*nj + 1):(i * nj)]
      Xni <- empirical_mean(subsample)
      # uni <- empirical_var(subsample)
      means <- c(means, Xni)
    }
    hist(means, xlab=paste("Moyenne empirique, echantillon de taille n=", nj), main=title, breaks=nj)
  }
}

mean_hist(function(n) { return (rnorm(n, mean=1, sd=2)) }, "Distribution Gausienne N(1, 2)")
```



Distribution Gausienne N(1, 2)



Je pose $S_n = \sum_{i=1}^n X_i$, tel que:: + les X_i i.i.d de même loi + $E[X_i] = \mu + \mathbb{V}[X_i] = \sigma^2$.

D'après le théorème central limite, pour N assez grand, S_n suit approximativement une loi normal $N(n\mu, n\sigma^2)$.

En notant la moyenne empirique $\bar{X}_n = \frac{S_n}{n}$, on a:

$$\mathbb{E}[\bar{X}_n] = \mathbb{E}\left[\frac{S_n}{n}\right] = \frac{1}{n}\mathbb{E}[S_n] = \mu$$

$$\mathbb{E}[\bar{X}_n] = \mathbb{V}\left[\frac{S_n}{n}\right] = \frac{1}{n^2}\mathbb{V}[S_n] = \frac{\sigma^2}{n}$$

Dans notre exemple, les X_i suivent une loi $N(1, 2)$.

En notant $(a_n, b_n) = (\text{moyenne, écart-type}) = (\mu, \sqrt{\frac{\sigma^2}{n}}) = (1, \frac{2}{\sqrt{n}})$, $U_n = \frac{\bar{X}_n - a_n}{b_n}$ suit une loi normal centrée réduite $N(0, 1)$.

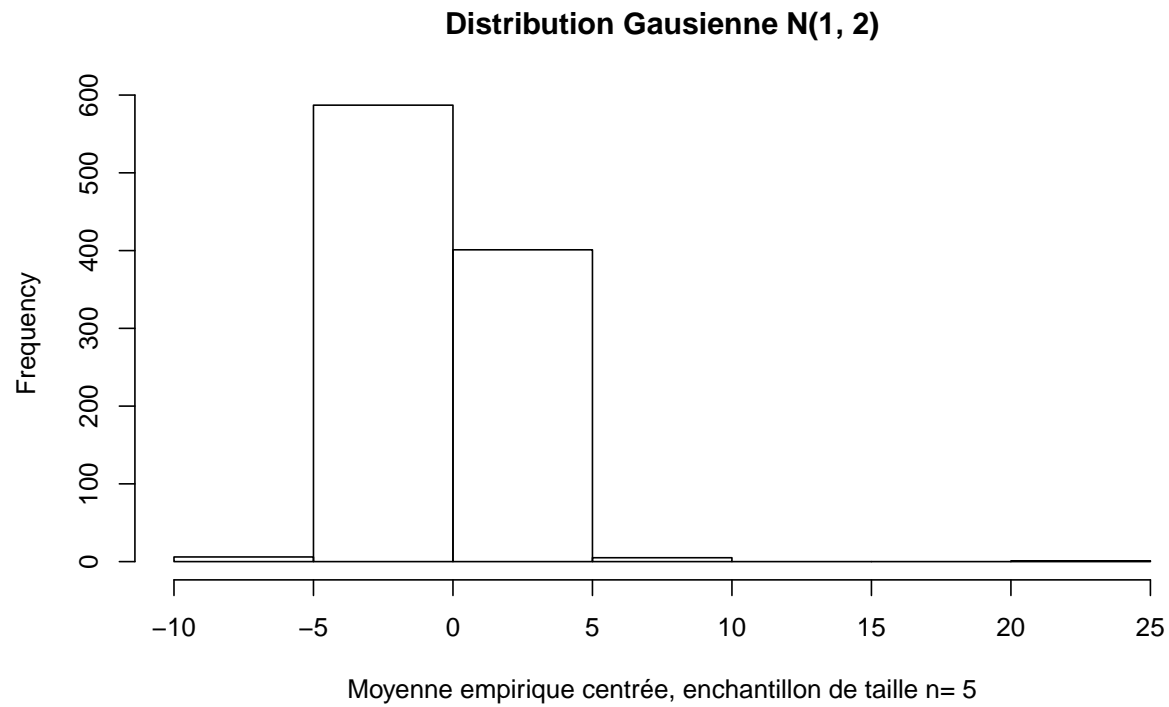
```
# fonction 'mean_norm_hist'
#
# 'law' : fonction qui genere un vecteur de taille 'm', e.g: law(42)
# 'title': titre de l'histogramme
#
# La fonction trace 3 histogrammes de la loi de la moyenne empirique renormalisé
# sur 'N' échantillons de taille dans 'ns'
mean_norm_hist <- function(law, title) {
  for (nj in n) {
    sample <- law(nj * N)
    Xn <- empirical_mean(sample)
    Un <- c()
    for (i in 1:N) {
      subsample <- sample[((i-1)*nj + 1):(i * nj)]
      ani <- empirical_mean(subsample)
      bni <- empirical_var(subsample) / sqrt(nj)
    }
  }
}
```

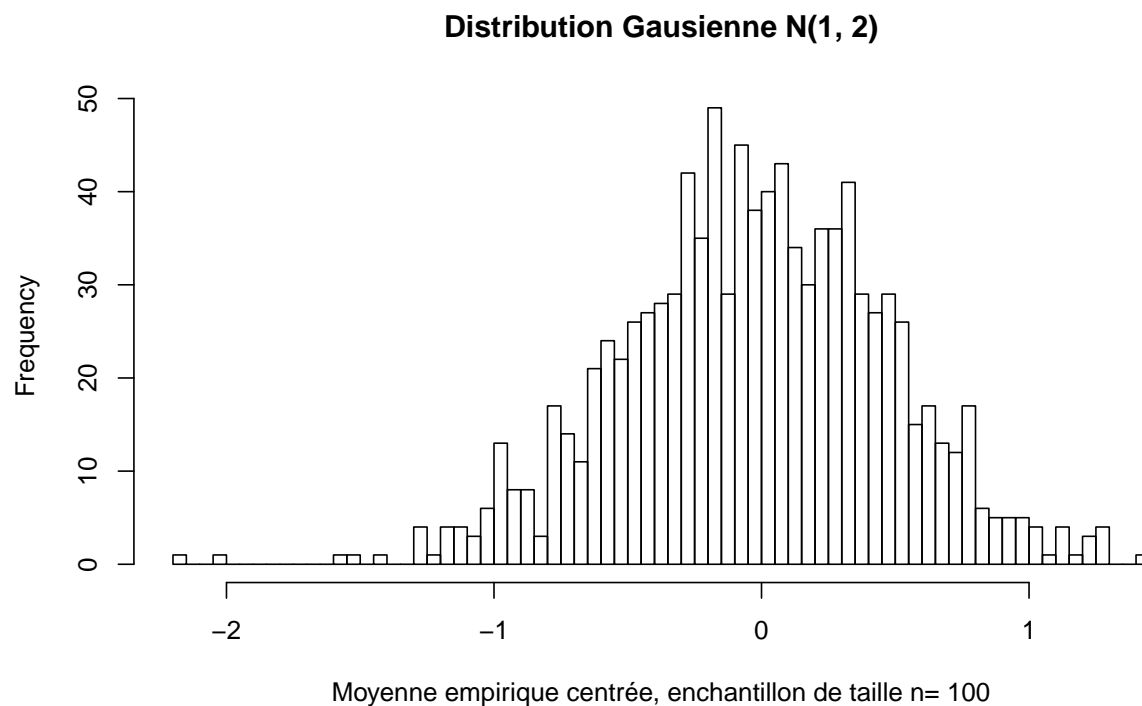
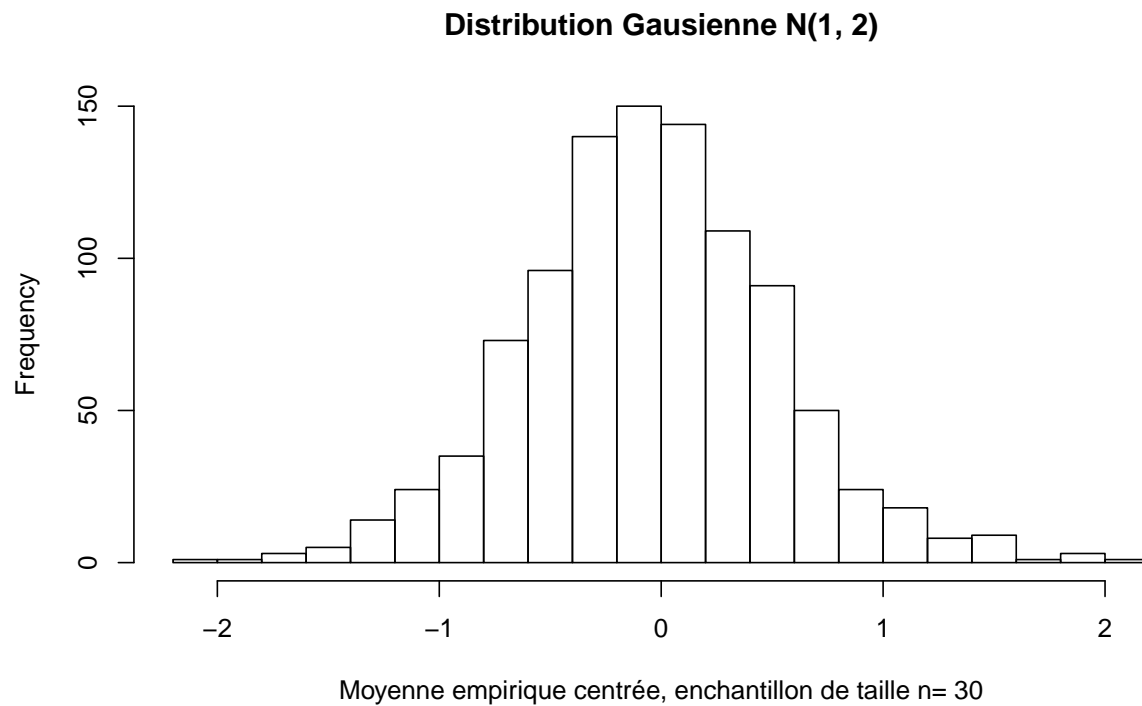
```

    Uni      <- (Xn - ani) / bni
    Un       <- c(Un, Uni)
  }
  hist(Un, xlab=paste("Moyenne empirique centrée, échantillon de taille n=", nj), main=title, breaks=
}
}

mean_norm_hist(function(n) { return (rnorm(n, mean=1, sd=2)) }, "Distribution Gausienne N(1, 2)")

```





Les histogrammes obtenus montrent en effet une loi normale centrée réduite.

Plus n est grand, plus la loi moyenne empirique renormalisée semble suivre une loi $N(0, 1)$. (cf Théorème Central Limite)

1.2 Loi de Pareto

Soit X une variable aléatoire suivant une loi de Pareto $P(a, \alpha), \alpha > 2$. Alors, $\mathbb{E}[X] = \frac{\alpha a}{\alpha - 1}$ et $\mathbb{V}[X] = (\frac{\alpha a}{\alpha - 1})^2 \frac{\alpha}{\alpha - 2}$

```
library("rmutil")
```

```
##
```

```
## Attaching package: 'rmutil'
```

```
## The following object is masked from 'package:stats':
```

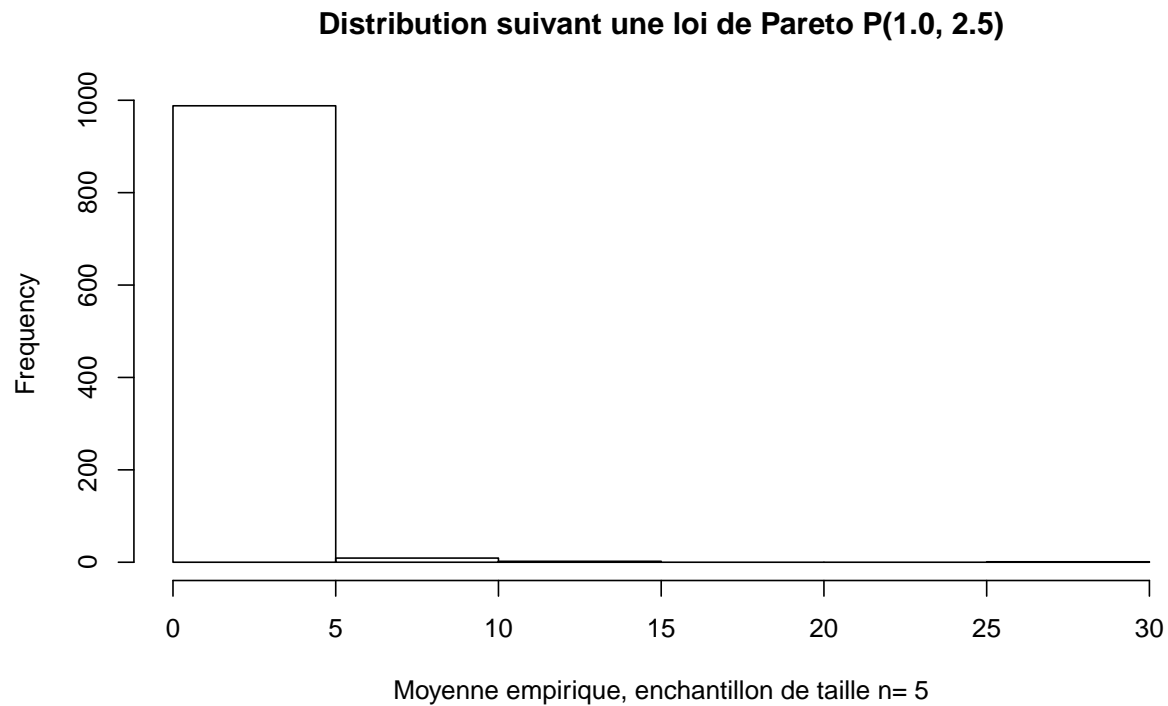
```
##
```

```
##      nobs
```

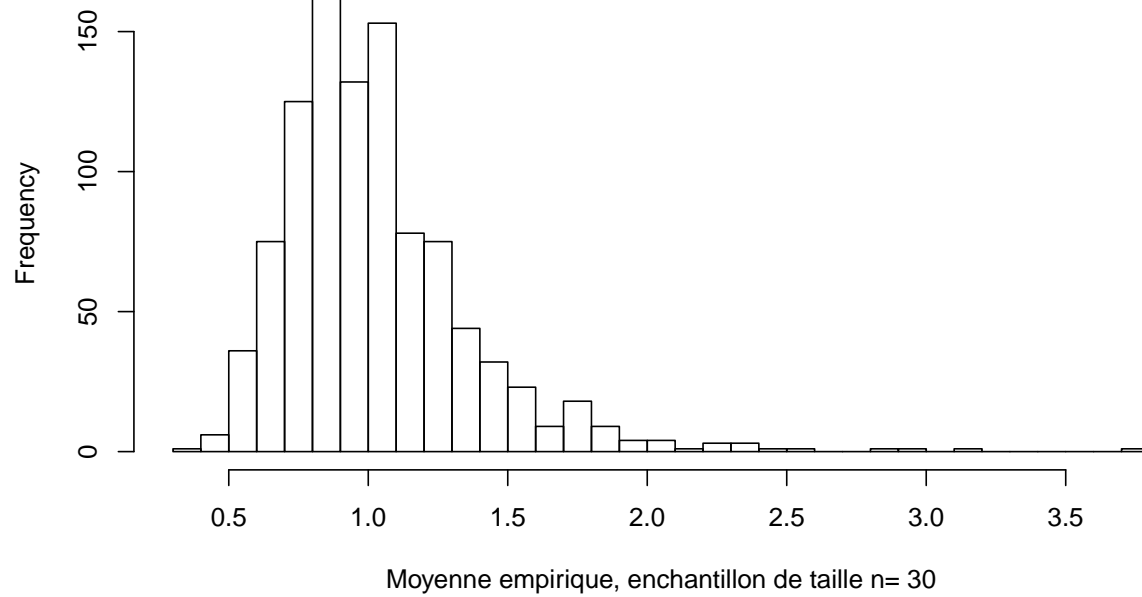
```
a      <- 1.0
```

```
alpha  <- 2.5
```

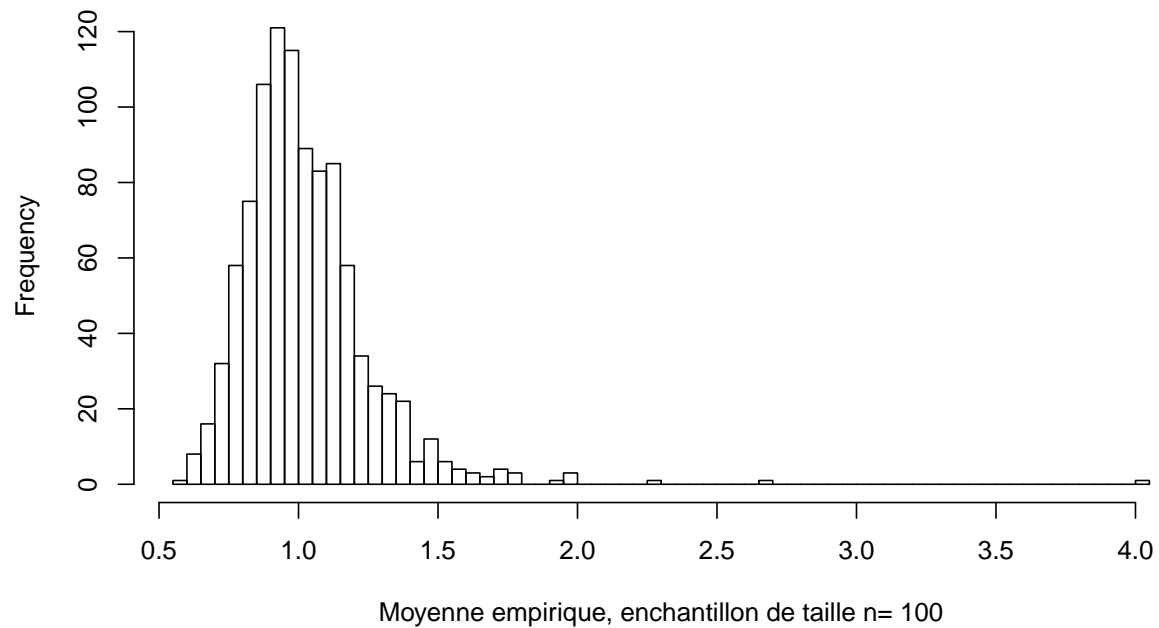
```
mean_hist(function(n) { return (rpareto(n, m=a, s=alpha)) }, "Distribution suivant une loi de Pareto P(
```



Distribution suivant une loi de Pareto P(1.0, 2.5)

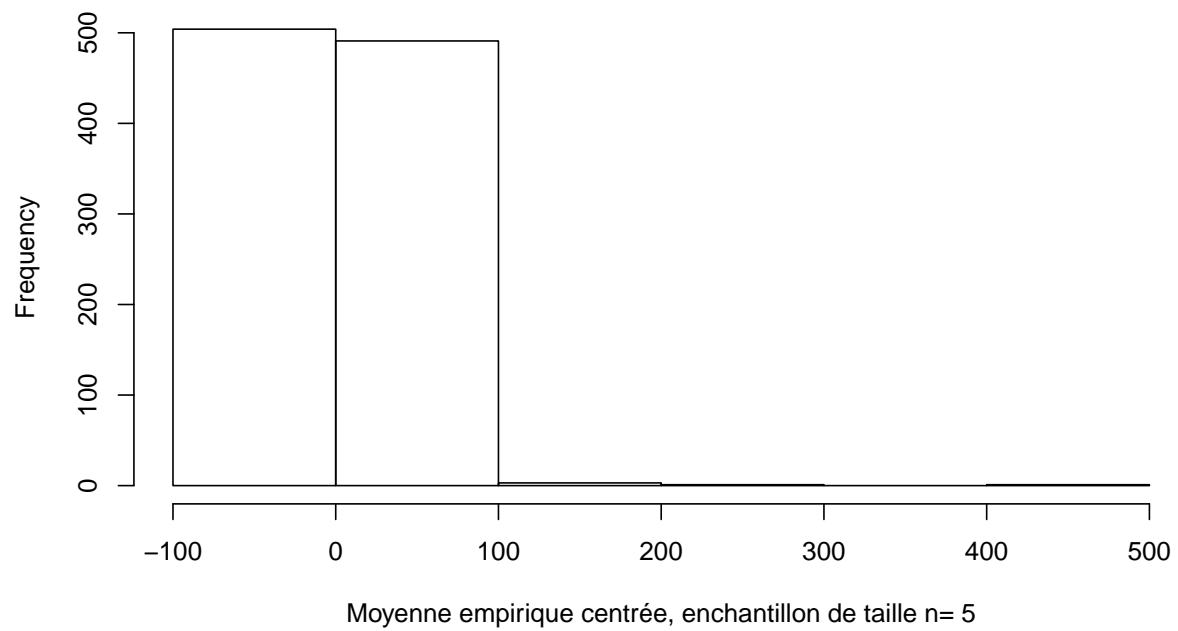


Distribution suivant une loi de Pareto P(1.0, 2.5)

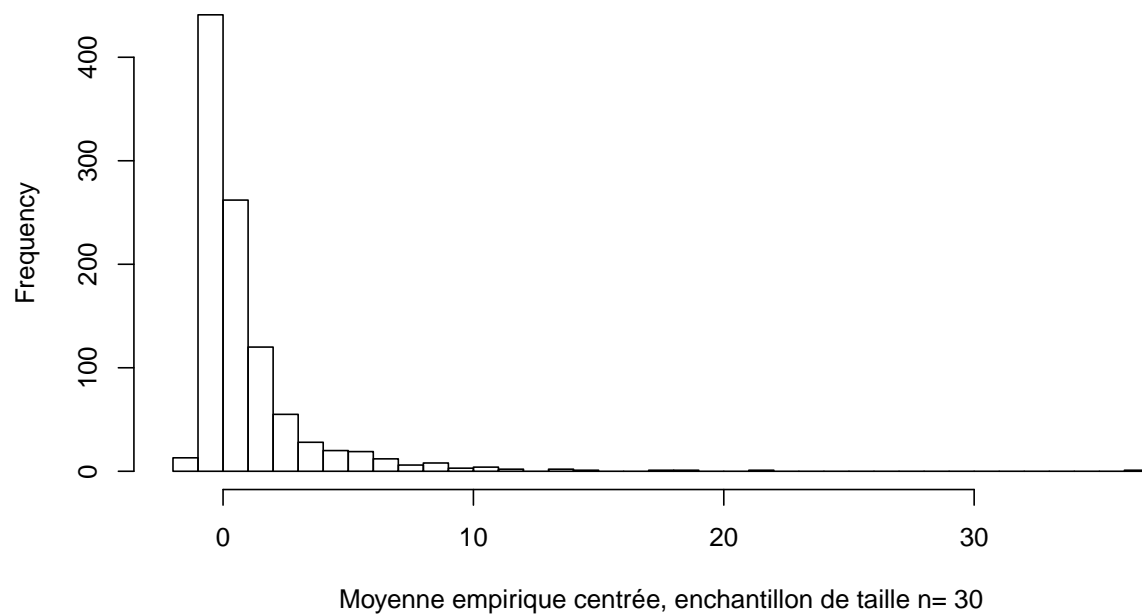


```
mean_norm_hist(function(n) { return (rpareto(n, m=a, s=alpha)) }, "Distribution suivant une loi de Pareto")
```

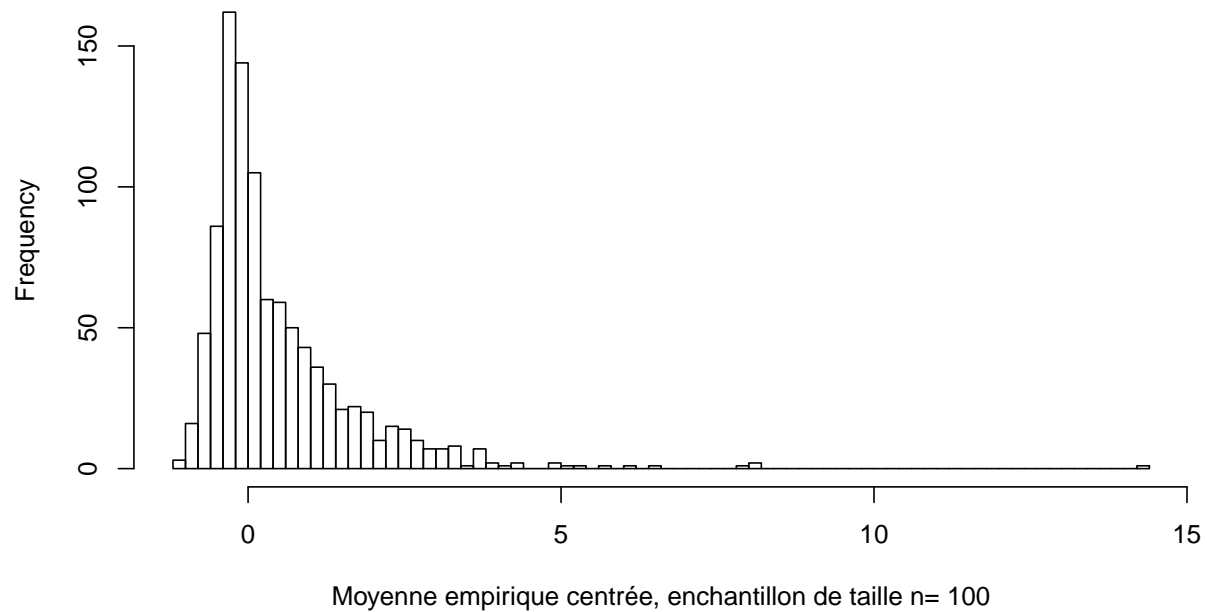
Distribution suivant une loi de Pareto P(1.0, 2.5)



Distribution suivant une loi de Pareto P(1.0, 2.5)



Distribution suivant une loi de Pareto P(1.0, 2.5)



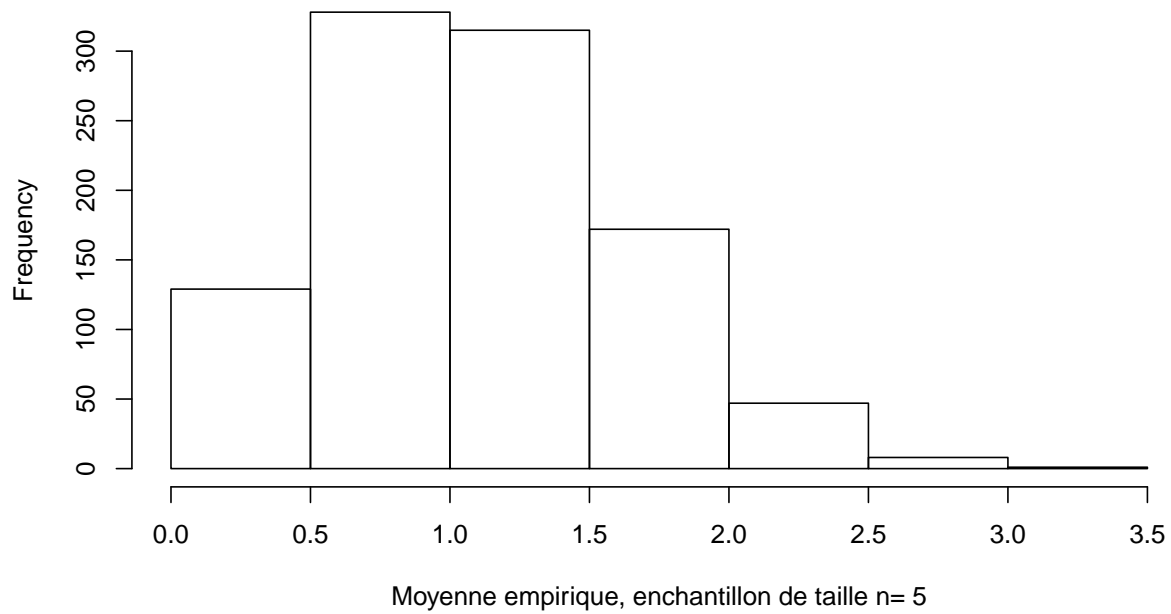
Plus n est grand, plus la loi moyenne empirique renormalisée semble suivre une loi $N(0, 1)$ (bien qu'elle possède de nombreuses valeurs 'à droite' faisant penser à une allure exponentielle).

1.3 Loi de Poisson

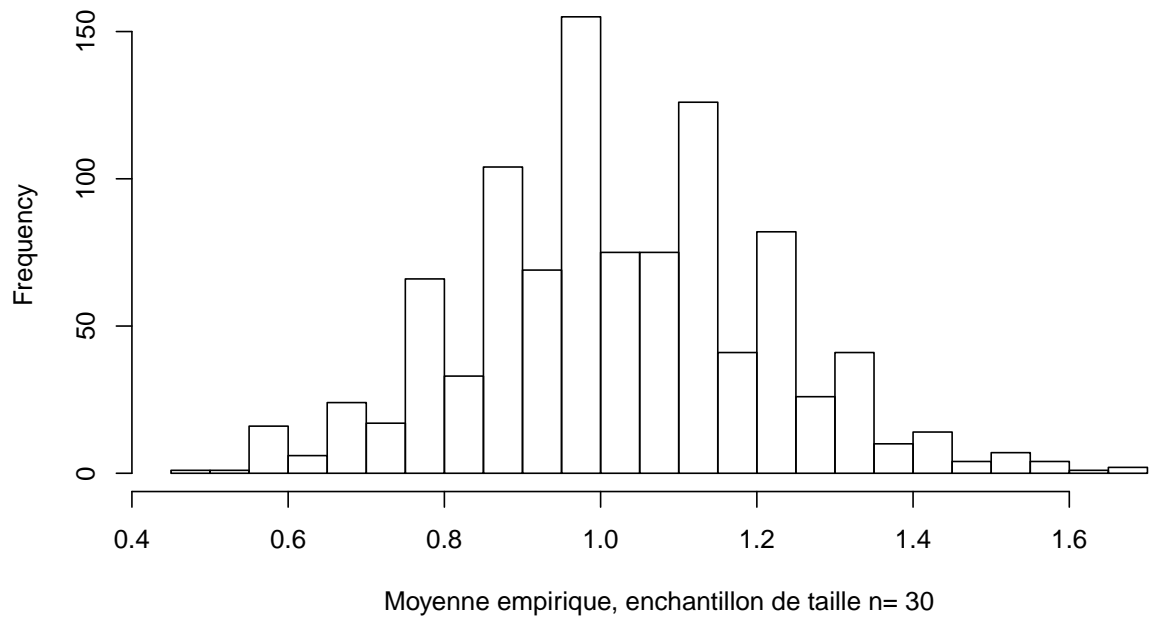
Soit X une variable aléatoire suivant une loi de Poisson $P(\lambda)$. Alors, $\mathbb{E}[X] = \lambda$ et $\mathbb{V}[X] = \lambda$

```
mean_hist(function(n) { return (rpois(n, lambda=1)) }, "Distribution suivant une loi de Poisson P(1)")
```

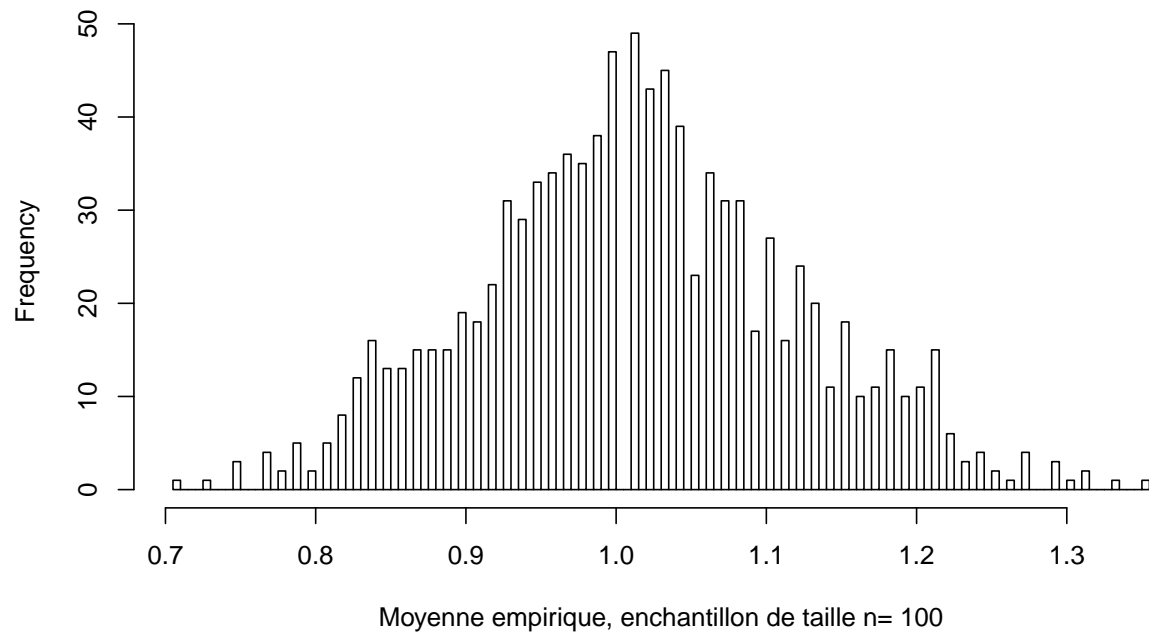
Distribution suivant une loi de Poisson P(1)



Distribution suivant une loi de Poisson P(1)

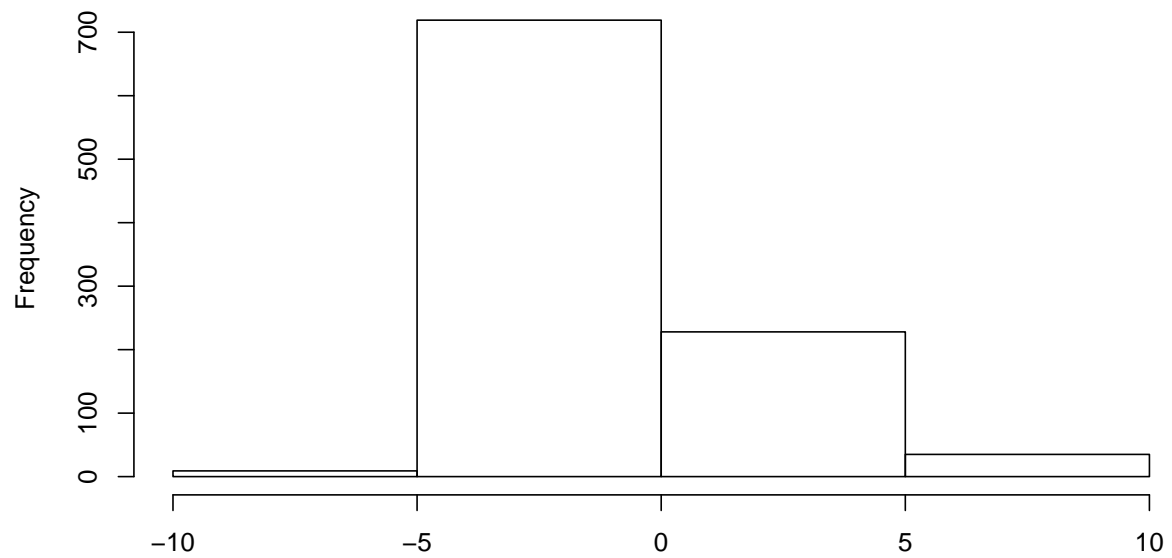


Distribution suivant une loi de Poisson P(1)



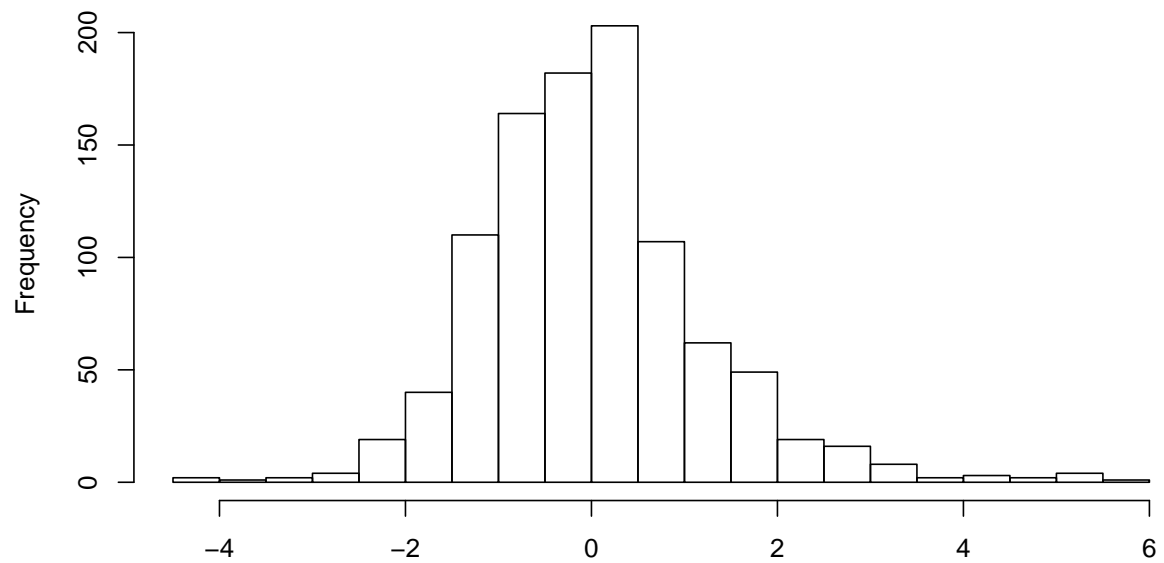
```
mean_norm_hist(function(n) { return (rpois(n, lambda=1)) }, "Distribution suivant une loi de Poisson P(1))
```

Distribution suivant une loi de Poisson $P(1)$

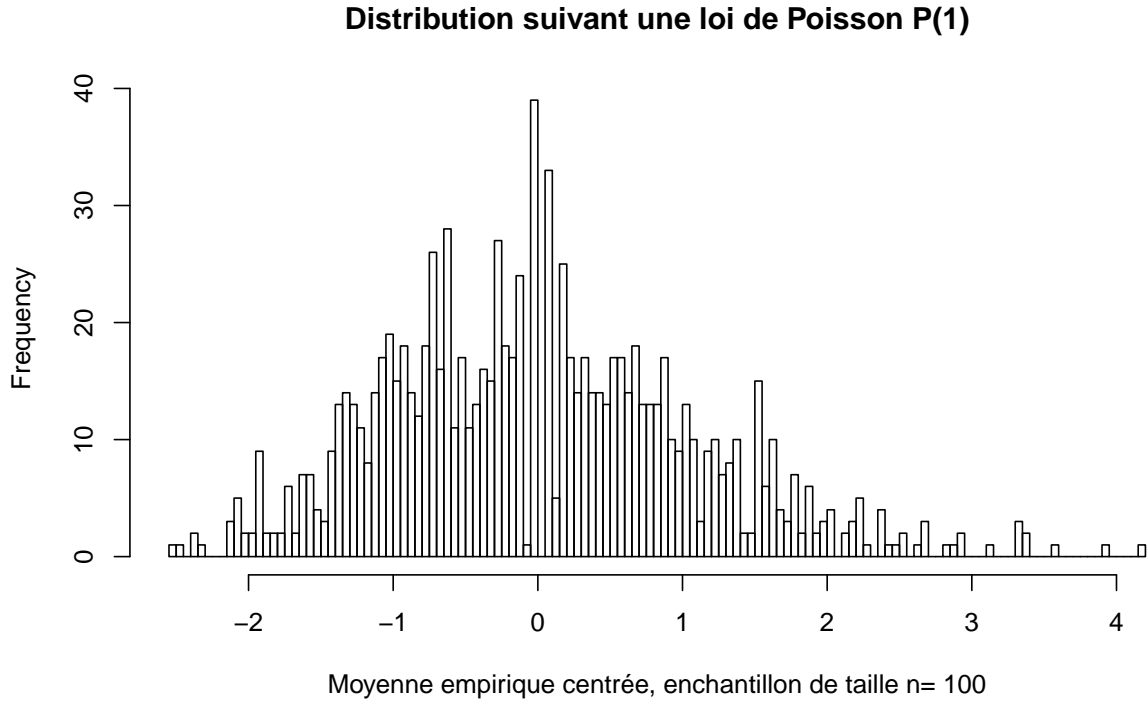


Moyenne empirique centrée, échantillon de taille $n=5$

Distribution suivant une loi de Poisson $P(1)$



Moyenne empirique centrée, échantillon de taille $n=30$



De même, plus n est grand, plus la loi moyenne empirique renormalisé semble suivre une loi $N(0, 1)$.

1.4 Méthode d'expérimentation

On note $X = (X_1, \dots, X_n)$ pour $n \in \mathbb{R}$, un échantillon de taille n (simulable 'facilement')

On suppose que tous les X_i son i.i.d, et suivent la même loi.

Soit $T : \Omega^n \rightarrow \mathbb{R}$ une statistique sur un échantillon de taille n .

On peut trouver une approximation de l'espérance $\mathbb{E}[T(X)]$ en utilisant le protocole suivant:

1. Fixer $N \in \mathbb{N}, N \gg 1$.
2. Générer N échantillons de taille n , notés $X^i = (X_1^i, \dots, X_n^i)$ avec $1 \leq i \leq N$
3. Je pose
 - $\bar{T}_N = \frac{1}{N} \sum_{i=1}^N T(X^i)$
 - Lorsque N devient grand, d'après le théorème central limite (même raisonnement qu'en **1.1**):
 - (1) : $\mathbb{E}[\bar{T}_N] \xrightarrow[N \gg 1]{} \mathbb{E}[T(X)]$
 - (2) : $\mathbb{V}[\bar{T}_N] \xrightarrow[N \gg 1]{} \frac{1}{N} \mathbb{V}[T(X)] \xrightarrow[N \rightarrow +\infty]{} 0$
4. Donc d'après (1) et (2), on a $\bar{T}_n \xrightarrow[N \rightarrow +\infty]{\mathbb{L}^2} \mathbb{E}[T(X)] = cste$.

Autrement dit, la moyenne empirique (entant que variable aléatoire), tends (en norme 2) vers la v.a constante $\mathbb{E}[T(X)]$.

La moyenne empirique est une bonne estimation de l'espérance quand elle est effectué sur un grand nombre d'échantillon.

2. Moyenne et dispersion.

2.1 Inégalité de Bienaymé Tchebychev

Soit X une variable aléatoire admettant un moment d'ordre 2 (et donc un moment d'ordre 1). L'inégalité de Bienaymé-Tchebychev affirme: $\forall \alpha \in \mathbb{R}_+^*, \mathbb{P}(|X - \mathbb{E}[X]| \geq \alpha) \leq \frac{1}{\alpha^2} \mathbb{V}[X]$

Pour une loi Gaussienne $N(\mu, \sigma^2)$, on a : $\mathbb{P}(|X - \mu| \geq \alpha) \leq \frac{\sigma^2}{\alpha^2}$

Pour une loi de Poisson $P(\lambda)$, on a : $\mathbb{P}(|X - \lambda| \geq \alpha) \leq \frac{\lambda}{\alpha^2}$

2.2 Estimation par Monte Carlo.

(a) $\mathbb{P}(|X - \mu| \geq \delta) = \mathbb{E}[1_{\{|X - \mu| \geq \delta\}}] = \mathbb{E}[Z]$, en posant $Z = 1_{\{|X - \mu| \geq \delta\}}$

(b) On estime $\mathbb{E}[Z]$ par la moyenne empirique $\bar{Z}_N = \frac{1}{N} \sum_{i=1}^N T(Z^i)$:

```
# effectue une estimation de Monte Carlo sur une loi Gaussienne, de Pareto, et de Poisson.
#
# N : nombre d'echantillon pour la moyenne
# delta : verifiant (a)
# (mu, sigma) : paramètre de la loi Gausienne
# (a, alpha) : paramètre de la loi de Pareto
# (lambda) : paramètre de la loi de Poisson
#
# renvoie une liste contenant :
#         - les distributions générées
#         - la moyenne empirique de ces distributions
#         - les distributions transformées Z (voir (a))
#         - les espérances empirique de Z, approximation de (a)
estimate_monte_carlo <- function(N, delta, mu, sigma, a, alpha, lambda) {
  # On genere des distributions
  XN      <- list("Gauss" = rnorm(N, mu, sigma), "Pareto" = rpareto(N, a, alpha),
                 "Poisson" = rpois(N, lambda))
  XN_means <- list("Gauss" = mu,                  "Pareto" = alpha*a/(alpha - 1),
                 "Poisson" = lambda)
  ZN      <- list()
  ZN_means <- list()
  # Pour chaque distributions
  for (distrib in names(XN)) {
    # on recupere la distribution
    XNi      <- XN[[distrib]]
    XNi_mean <- XN_means[[distrib]]
    # on génère la variable aléatoire Z correspondante
    ZN[[distrib]] <- unlist(lapply(XNi, function(xi) {
      if (abs(xi-XNi_mean) >= delta) {
        return (1)
      }
      return (0)
    })))
    ZN_means[[distrib]] <- empirical_mean(ZN[[distrib]])
  }
  return (list("XN" = XN, "XN_means" = XN_means, "ZN" = ZN, "ZN_means" = ZN_means))
}
```

On obtient selon les différentes lois: $\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta_{=1}) = \dots$

```
N <- 1e5
estimation <- estimate_monte_carlo(N, delta=1, mu=0, sigma=1, a=1.0, alpha=2.5, lambda=1)
estimation[["ZN_means"]]
```

```
## $Gauss
## [1] 0.3172032
##
## $Pareto
## [1] 0.6775368
##
## $Poisson
## [1] 0.6306863
```

La moyenne empirique est une variable aléatoire, et on a montré que $\mathbb{E}[\bar{Z}_N] = \mathbb{E}[Z]$ et $\mathbb{V}[\bar{Z}_N] = \frac{1}{N} \mathbb{V}[Z]$.

Donc d'après le théorème de Bienaymé Tchebichev, la précision de notre estimation

$$\mathbb{P}(|X - \mu| \geq \delta) = \mathbb{E}[Z] \simeq \bar{Z}_N$$

est donné par:

$$\forall \delta \geq 0, \mathbb{P}[|\bar{Z}_N - \mathbb{E}[Z]| \geq \delta] \leq \frac{1}{\delta * N} \mathbb{V}[Z]$$

(c) Application numérique:

```
markov_sup <- function(Z, N, delta) {
  return (var(Z) / (delta * N));
}
delta <- 1e-4
```

On a fixé plus tôt $N = 10^5$. On fixe $\delta = 10^{-4}$.

En fonction des loi de X précédentes, notre estimation de $\bar{Z}_N \simeq \mathbb{E}[Z]$ vérifie:

$$\mathbb{P}[|\bar{Z}_N - \mathbb{E}[Z]| \geq \delta] = \mathbb{P}(\bar{Z}_N \notin [\mathbb{E}[Z] - \delta; \mathbb{E}[Z] + \delta]) \leq \dots$$

```
XN <- estimation[["XN"]]
ZN <- estimation[["ZN"]]
for (distrib in names(XN)) {
  print(paste(distrib, ":", markov_sup(ZN[[distrib]], N, delta)))
}
```

```
## [1] "Gauss : 0.0216586325863259"
## [1] "Pareto : 0.021848528395284"
## [1] "Poisson : 0.0232925066850668"
```

Remarques:

- Plus δ est 'petit' devant 1, plus notre précision est incertaine. (la probabilité que notre estimation soit dans l'intervalle $[\mathbb{E}[Z] - \delta; \mathbb{E}[Z] + \delta]$ s'éloigne de 1)
- Plus N est 'grand' devant 1, plus notre précision est probable. (la probabilité que notre estimation soit dans l'intervalle $[\mathbb{E}[Z] - \delta; \mathbb{E}[Z] + \delta]$ tends vers 1)
- Plus $\mathbb{V}[Z]$ est 'grande' devant δ , plus notre précision est incertaine. (la probabilité que notre estimation soit dans l'intervalle $[\mathbb{E}[Z] - \delta; \mathbb{E}[Z] + \delta]$ s'éloigne de 1)

Voici les bornes obtenus pour différentes valeurs de δ et σ :

```
# Majoration par l'inégalité de Markov, de la probabilité que
# la moyenne obtenu s'écarte à +- epsilon prêt de l'espérance:
for (delta in c(1e-4, 1e-2, 1)) {
  for (sigma in c(1, 10, 100)) {
    estimation <- estimate_monte_carlo(N, delta, mu=0, sigma, a=1.0, alpha=2.5, lambda=1)
    XN <- estimation[["XN"]]
    ZN <- estimation[["ZN"]]
    print("-----")
    print(paste("Pour delta=", delta, " et sigma=", sigma, sep=""))
    for (distrib in names(XN)) {
      print(paste(distrib, ":", markov_sup(ZN[[distrib]], N, delta)))
    }
  }
}
```

```
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=1"
## [1] "Gauss : 7.99943999439995e-06"
## [1] "Pareto : 1e-06"
## [1] "Poisson : 0.0232180071800718"
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=10"
## [1] "Gauss : 0"
## [1] "Pareto : 4.99979999799998e-06"
## [1] "Poisson : 0.0232477860378604"
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=100"
## [1] "Gauss : 0"
## [1] "Pareto : 3.99987999879998e-06"
## [1] "Poisson : 0.023310493094931"
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=1"
## [1] "Gauss : 7.86719377193772e-06"
## [1] "Pareto : 2.39426394263942e-06"
## [1] "Poisson : 0.000232443424334243"
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=10"
## [1] "Gauss : 9.19162791627916e-07"
## [1] "Pareto : 2.40421594215942e-06"
## [1] "Poisson : 0.000232551915519155"
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=100"
## [1] "Gauss : 9.9990999909999e-08"
## [1] "Pareto : 2.50372493724937e-06"
## [1] "Poisson : 0.0002322467599676"
## [1] "-----"
## [1] "Pour delta=1 et sigma=1"
## [1] "Gauss : 2.16531462814628e-06"
## [1] "Pareto : 2.17573369333693e-06"
## [1] "Poisson : 2.32302681426814e-06"
## [1] "-----"
## [1] "Pour delta=1 et sigma=10"
## [1] "Gauss : 7.43056374563745e-07"
```



```
## [1] "Pareto : 2.178142300423e-06"
## [1] "Poisson : 2.32217432574326e-06"
## [1] "-----"
## [1] "Pour delta=1 et sigma=100"
## [1] "Gauss : 7.78845538455385e-08"
## [1] "Pareto : 2.16859966099661e-06"
## [1] "Poisson : 2.32699555395554e-06"
```

(d) Inégalité de Chernoff.

Soit X une variable aléatoire admettant une fonction génératrice.

L'inégalité de Chernoff donne:

$\forall \delta \in \mathbb{R}, \forall \epsilon \in \mathbb{R}_+^*,$

$\mathbb{P}(X - \mathbb{E}[X] \geq \delta) \leq e^{-\epsilon \delta} \mathbb{E}[e^{(X - \mathbb{E}[X])\epsilon}]$

$\mathbb{P}(X - \mathbb{E}[X] \leq -\delta) \leq e^{-\epsilon \delta} \mathbb{E}[e^{(X - \mathbb{E}[X])\epsilon}]$

Donc,

$\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta) \leq 2e^{-\epsilon \delta} \mathbb{E}[e^{(X - \mathbb{E}[X])\epsilon}]$

Pour une variable Gaussienne $N(\mu, \sigma^2)$, on a: $\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta) \leq 2e^{\frac{-\delta^2}{2\sigma^2}}$

Pour une variable de Poisson $P(\lambda)$, on a: $\mathbb{P}(|X - \mathbb{E}[X]| \geq \delta) \leq 2e^{\frac{-\delta^2}{2(\lambda + \delta)}}$

```
# Majoration par l'inégalité de Chernoff, de la probabilité que
# la moyenne obtenu s'écarte à +- delta prêt de l'espérance:
for (delta in c(1e-4, 1e-2, 1, 10)) {
  for (sigma in c(1, 10, 100)) {
    estimation <- estimate_monte_carlo(N, delta, mu=0, sigma, a=1.0, alpha=2.5, lambda=1)
    XN <- estimation[["XN"]]
    ZN <- estimation[["ZN"]]
    print("-----")
    print(paste("Pour delta=", delta, " et sigma=", sigma, sep=""))
    print(paste("Gaussienne", ":", 2.0 * exp(-delta * delta / (2 * sigma))))
    print(paste("Poisson", ":", 2.0 * exp(-delta * delta / (2 * (delta + 1)))))
  }
}
```

```
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=1"
## [1] "Gaussienne : 1.99999999"
## [1] "Poisson : 1.999999990001"
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=10"
## [1] "Gaussienne : 1.999999999"
## [1] "Poisson : 1.999999990001"
## [1] "-----"
## [1] "Pour delta=1e-04 et sigma=100"
## [1] "Gaussienne : 1.9999999999"
## [1] "Poisson : 1.999999990001"
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=1"
## [1] "Gaussienne : 1.99990000249996"
## [1] "Poisson : 1.99990099254971"
```

```
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=10"
## [1] "Gaussienne : 1.999990000025"
## [1] "Poisson : 1.99990099254971"
## [1] "-----"
## [1] "Pour delta=0.01 et sigma=100"
## [1] "Gaussienne : 1.9999900000025"
## [1] "Poisson : 1.99990099254971"
## [1] "-----"
## [1] "Pour delta=1 et sigma=1"
## [1] "Gaussienne : 1.21306131942527"
## [1] "Poisson : 1.55760156614281"
## [1] "-----"
## [1] "Pour delta=1 et sigma=10"
## [1] "Gaussienne : 1.90245884900143"
## [1] "Poisson : 1.55760156614281"
## [1] "-----"
## [1] "Pour delta=1 et sigma=100"
## [1] "Gaussienne : 1.99002495838536"
## [1] "Poisson : 1.55760156614281"
## [1] "-----"
## [1] "Pour delta=10 et sigma=1"
## [1] "Gaussienne : 3.85749969592784e-22"
## [1] "Poisson : 0.0212306929239533"
## [1] "-----"
## [1] "Pour delta=10 et sigma=10"
## [1] "Gaussienne : 0.0134758939981709"
## [1] "Poisson : 0.0212306929239533"
## [1] "-----"
## [1] "Pour delta=10 et sigma=100"
## [1] "Gaussienne : 1.21306131942527"
## [1] "Poisson : 0.0212306929239533"
```

Pour δ petit devant la variance, la borne tends vers 2... ce qui n'apporte pas d'informations utiles.

2.3.

2.4.

(a)

```
theta <- 0
for (n in c(20, 100, 1000, 10000)) {
  cauchy <- rcauchy(n, location=theta, scale=1)
  m <- empirical_mean(cauchy)
  print(paste("n=", n, " ; la moyenne empirique calculé est: ", m, sep=""))
}
```

```
## [1] "n=20 ; la moyenne empirique calculé est: -0.322681600247578"
## [1] "n=100 ; la moyenne empirique calculé est: 1.70315796697599"
## [1] "n=1000 ; la moyenne empirique calculé est: 0.336946731837079"
## [1] "n=10000 ; la moyenne empirique calculé est: -2.98878886388578"
```

La moyenne empirique donne des valeurs très différentes selon 'n', et ne semble pas converger.

(b) Une variable aléatoire X suivant une loi de Cauchy $C(\theta)$ n'admet pas d'espérance:

$f_X(x, \theta) = \frac{1}{\pi} \frac{1}{1+(x-\theta)^2}$, et quand $x \rightarrow +\infty$, $xf_X(x, \theta) \sim \frac{1}{x}$, donc:

$\mathbb{E}[X] = \int_{-\infty}^{+\infty} |xf_X(x, \theta)|dx$ diverge.

Donc le théorème central limite ne s'applique pas: il n'y a pas d'espérance, donc la moyenne empirique ne converge pas.

Ceci s'explique par le fait que la probabilité d'obtenir une valeur éloigné de θ (la médiane) est trop élevée pour que la moyenne converge.

(c) La médiane d'une loi de Cauchy $C(\theta)$ est θ .

Si l'on sait qu'un phénomène suit une loi de Cauchy, il est possible de déterminer son paramètre θ en suivant ce protocole:

1. Fixer $n \in \mathbb{N}, n \gg 1$.
2. Générer un échantillon de taille n .
3. Trier les valeurs de cette échantillon par ordre croissant. (ou décroissant)
4. La valeur au centre de l'échantillon trié (en $\frac{n}{2}$) est un estimateur de θ .

Application:

```
theta <- 0
for (theta in c(-1, 0, 1)) {
  print("-----")
  print(paste("theta=", theta, sep=""))
  for (n in c(20, 100, 1000, 10000)) {
    cauchy <- rcauchy(n, location=theta, scale=1)
    sorted <- sort(cauchy)
    print(paste("la médiane de l'échantillon n=", n, " vaut:", sorted[n / 2 + 1], sep=""))
  }
}
```

```
## [1] "-----"
## [1] "theta=-1"
## [1] "la médiane de l'échantillon n=20 vaut:-1.0085773619437"
## [1] "la médiane de l'échantillon n=100 vaut:-0.968412784989428"
## [1] "la médiane de l'échantillon n=1000 vaut:-1.06308593380296"
## [1] "la médiane de l'échantillon n=10000 vaut:-1.03062276390339"
## [1] "-----"
## [1] "theta=0"
## [1] "la médiane de l'échantillon n=20 vaut:0.132882341777964"
## [1] "la médiane de l'échantillon n=100 vaut:-0.151914773973725"
## [1] "la médiane de l'échantillon n=1000 vaut:-0.0227109894865033"
## [1] "la médiane de l'échantillon n=10000 vaut:-0.00575708000002459"
## [1] "-----"
## [1] "theta=1"
## [1] "la médiane de l'échantillon n=20 vaut:0.955603243825675"
## [1] "la médiane de l'échantillon n=100 vaut:0.883086179584292"
## [1] "la médiane de l'échantillon n=1000 vaut:0.956111277067289"
## [1] "la médiane de l'échantillon n=10000 vaut:0.976645768137792"
```

Les valeurs obtenus par la simulation sont en accord avec celle attendu par notre protocole.