TP 3 Statistiques

Adib Habbou, Adel Kebli, Clark Ji

11/03/2022

Echantillon, Théorème Central Limite, Estimation Monte Carlo

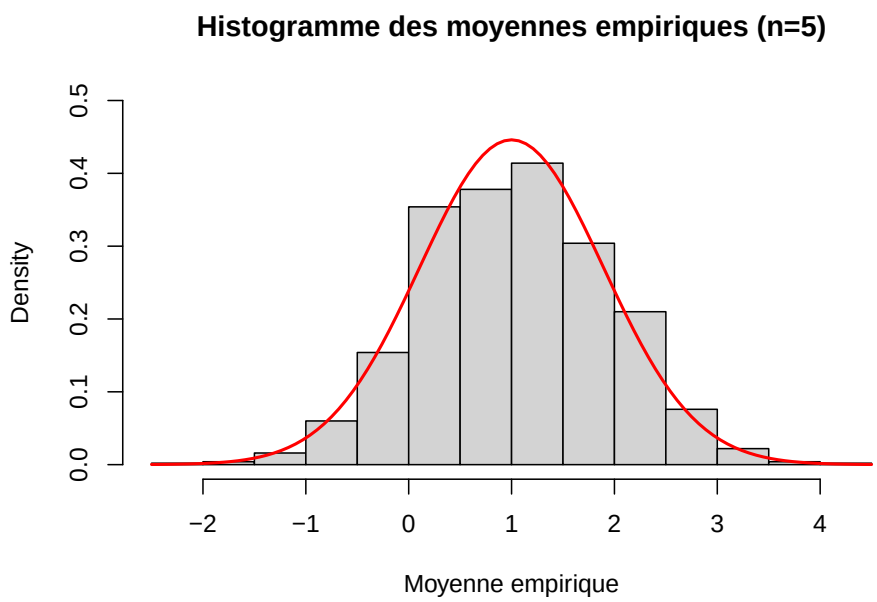
Question 1 :

```
gauss_5 = matrix(rnorm(5*1000, 1, 2), ncol=5, nrow=1000)
gauss_30 = matrix(rnorm(30*1000, 1, 2), ncol=30, nrow=1000)
gauss_100 = matrix(rnorm(100*1000, 1, 2), ncol=100, nrow=1000)
```

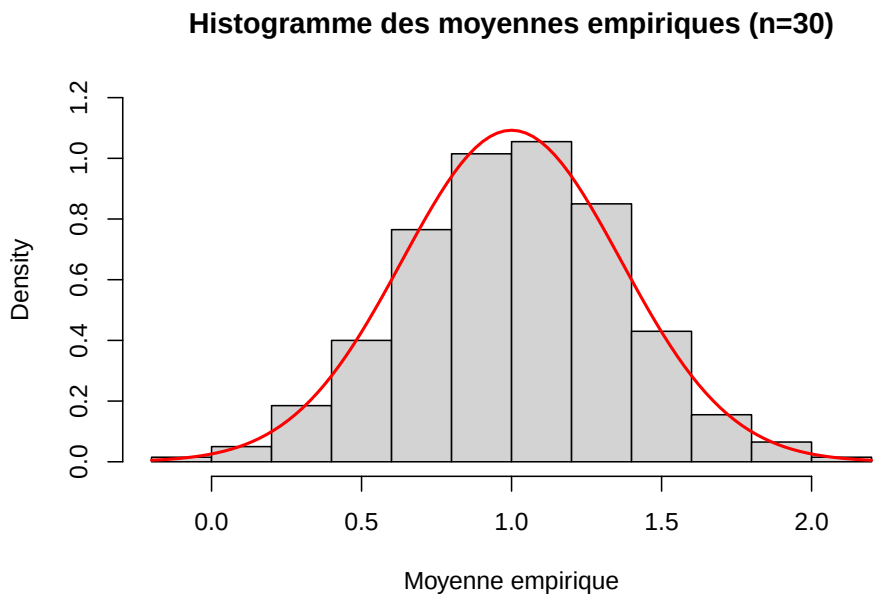
```
mean_gauss_5 = apply(gauss_5, 1, mean)
mean_gauss_30 = apply(gauss_30, 1, mean)
mean_gauss_100 = apply(gauss_100, 1, mean)
```

```
var_gauss_5 = apply(gauss_5, 1, var)
var_gauss_30 = apply(gauss_30, 1, var)
var_gauss_100 = apply(gauss_100, 1, var)
```

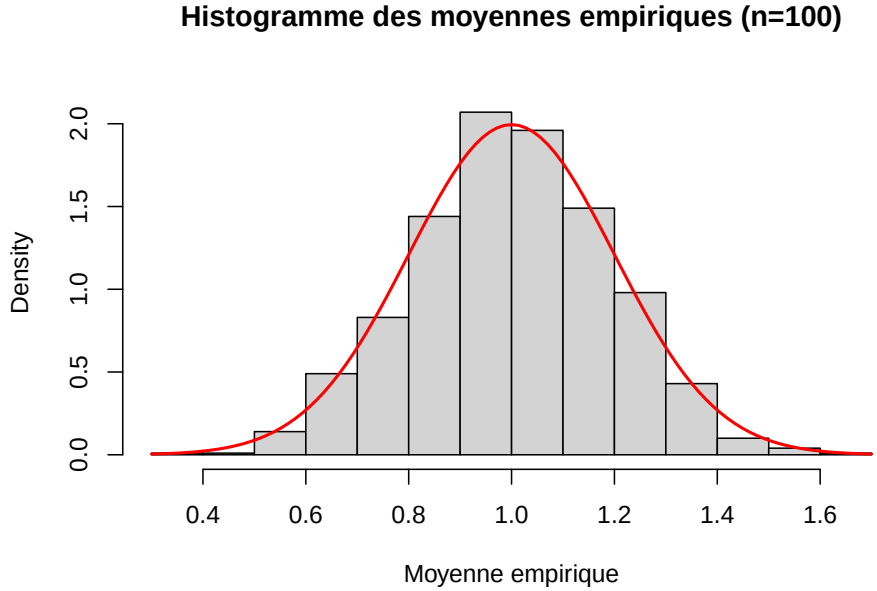
```
hist(mean_gauss_5, main="Histogramme des moyennes empiriques (n=5)",
     xlab="Moyenne empirique", ylim=c(0,0.5), prob=TRUE)
curve(dnorm(x, 1, sqrt(4/5)), add=TRUE, col="red", lwd=2)
```



```r
hist(mean_gauss_30, main="Histogramme des moyennes empiriques (n=30)",
     xlab="Moyenne empirique", ylim=c(0,1.2), prob=TRUE)
curve(dnorm(x, 1, sqrt(4/30)), add=TRUE, col="red", lwd=2)
```




```r
hist(mean_gauss_100, main="Histogramme des moyennes empiriques (n=100)",
     xlab="Moyenne empirique", ylim=c(0,2.2), prob=TRUE)
curve(dnorm(x, 1, sqrt(4/100)), add=TRUE, col="red", lwd=2)
```



La loi théorique de la moyenne empirique est la loi gaussienne

$$\mathcal{N}(1, \frac{\sigma}{\sqrt{n}})$$

On utilisera comme paramètres pour la renormalisation

$$a_n = 1$$

et

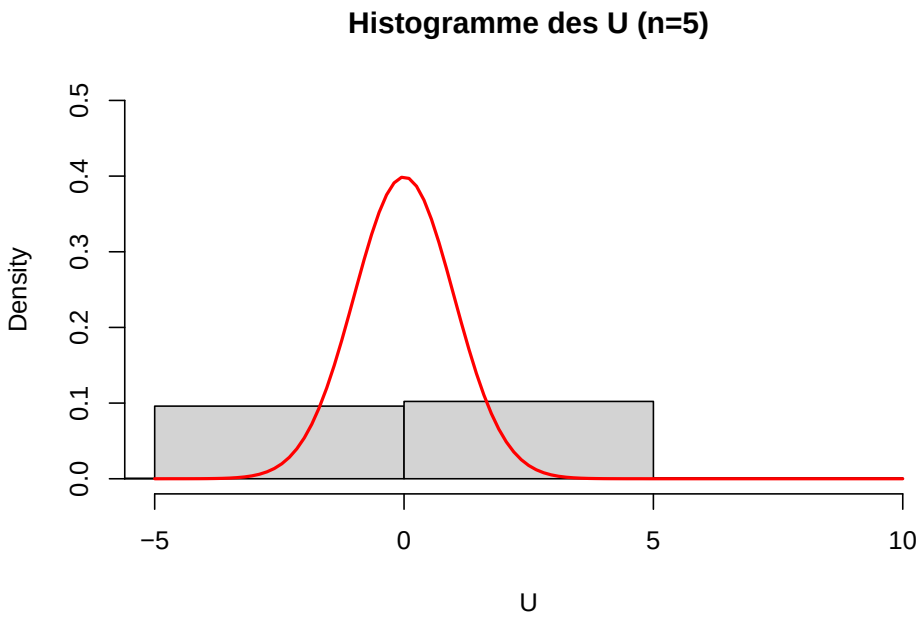
$$b_n = \frac{\sigma_{n,i}}{\sqrt{n}}$$

D'après le théorème central limite, on peut dire que la variable aléatoire

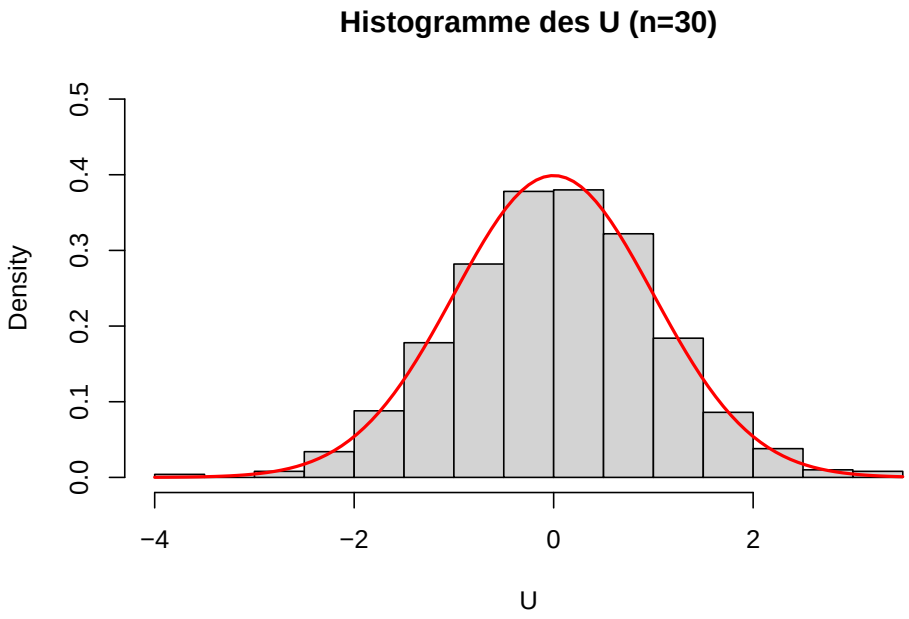
$$U_{n,i} = \frac{\bar{X}_{n,i} - a_n}{b_n} \xrightarrow[n \to +\infty]{loi} \mathcal{N}(0,1)$$

```
sd_gauss_5 = apply(gauss_5, 1, sd)
sd_gauss_30 = apply(gauss_30, 1, sd)
sd_gauss_100 = apply(gauss_100, 1, sd)
```

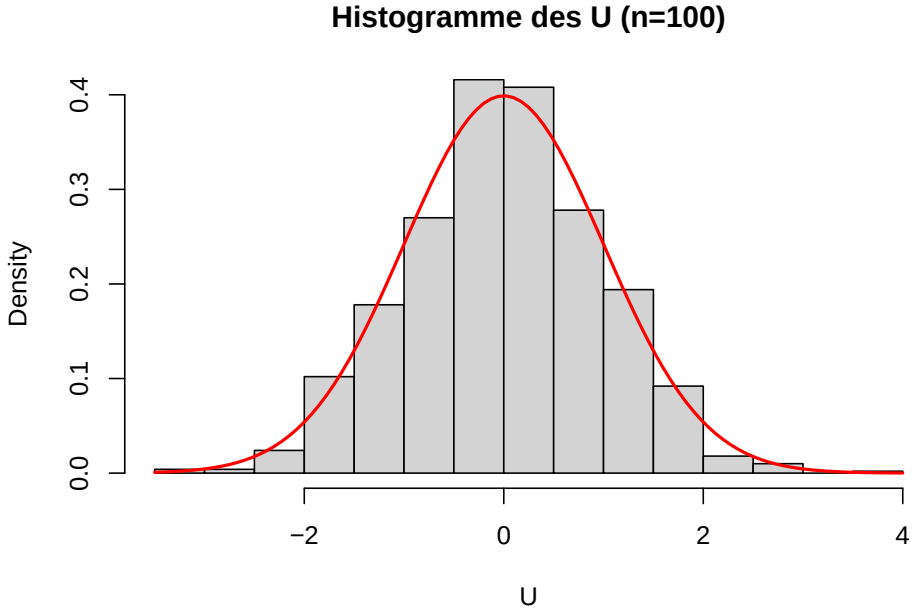
```
u_gauss_5 = (mean_gauss_5 - 1) / (sd_gauss_5 / sqrt(5))
hist(u_gauss_5, main="Histogramme des U (n=5)", xlab="U", ylim=c(0,0.5), xlim=c(-5,10), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```




```
u_gauss_30 = (mean_gauss_30 - 1) / (sd_gauss_30 / sqrt(30))
hist(u_gauss_30, main="Histogramme des U (n=30)", xlab="U", ylim=c(0,0.5), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```




```
u_gauss_100 = (mean_gauss_100 - 1) / (sd_gauss_100 / sqrt(100))
hist(u_gauss_100, main="Histogramme des U (n=100)", xlab="U", ylim=c(0,0.4), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```



On remarque que l'histogramme des $U_{n,i}$ correspond à la densité de la loi normale centrée réduite $\mathcal{N}(0,1)$. Plus la taille de l'échantillon n est grande plus on peut approcher la variable aléatoire $U_{n,i}$ par une loi normale centrée réduite $\mathcal{N}(0,1)$. Ce qui est cohérent avec les résultats du théorème central limite.

Question 2 :

```
pareto_5 = matrix(rpareto(5*1000, 3, 3), ncol=5, nrow=1000)
pareto_30 = matrix(rpareto(30*1000, 3, 3), ncol=30, nrow=1000)
pareto_100 = matrix(rpareto(100*1000, 3, 3), ncol=100, nrow=1000)
```

```
mean_pareto_5 = apply(pareto_5, 1, mean)
mean_pareto_30 = apply(pareto_30, 1, mean)
mean_pareto_100 = apply(pareto_100, 1, mean)
```

```
var_pareto_5 = apply(pareto_5, 1, var)
var_pareto_30 = apply(pareto_30, 1, var)
var_pareto_100 = apply(pareto_100, 1, var)
```

Vérification de l'espérance théorique d'une loi de Pareto :

$$f(x,a,\alpha) = \frac{\alpha a^\alpha}{x^{\alpha+1}} \mathbb{I}_{[a,+\infty[}(x)$$

$$\mathbb{E}[X] = \int_a^{+\infty} x f(x,a,\alpha) dx$$

$$\mathbb{E}[X] = \int_a^{+\infty} x \times \frac{\alpha a^\alpha}{x^{\alpha+1}} dx$$

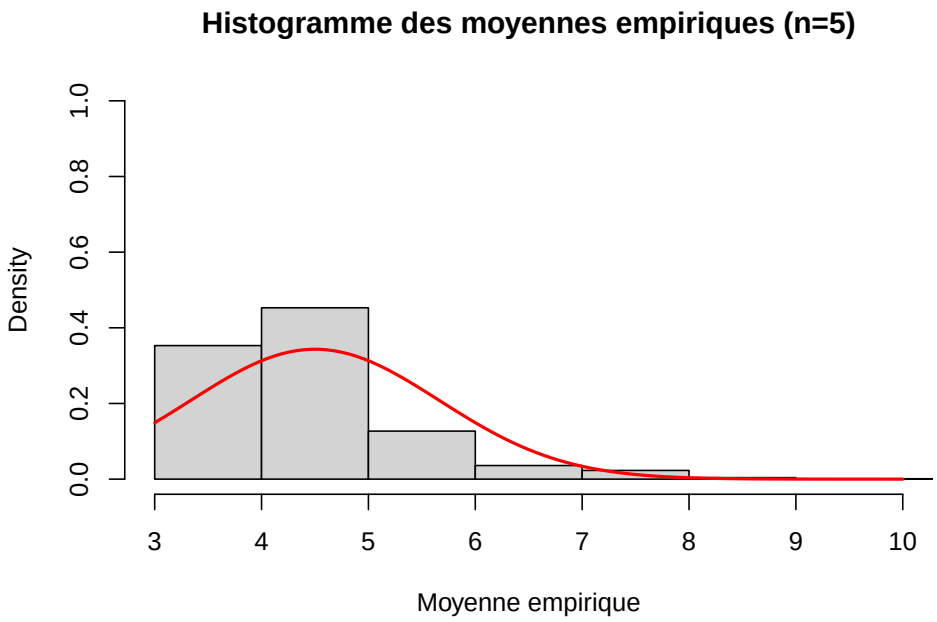
$$\mathbb{E}[X] = \int_a^{+\infty} \frac{\alpha a^\alpha}{x^\alpha} dx$$

$$\mathbb{E}[X] = \alpha a^\alpha \left[\frac{x^{1-\alpha}}{1-\alpha} \right]_a^{+\infty}$$

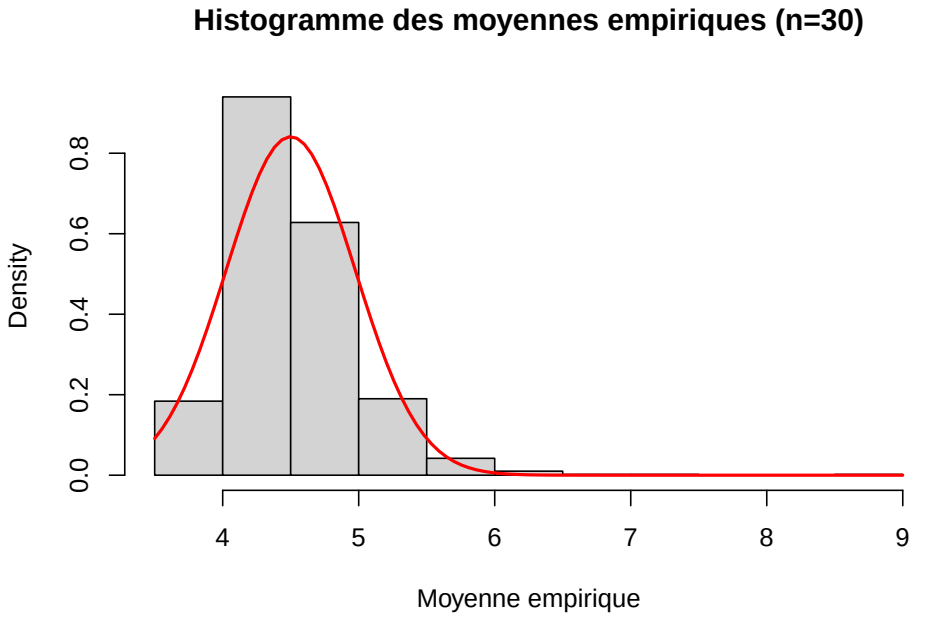
$$\mathbb{E}[X] = \frac{\alpha a^\alpha}{1-\alpha} \times \left(\frac{-1}{a^{\alpha-1}} \right)$$

$$\mathbb{E}[X] = \frac{\alpha a}{\alpha - 1}$$

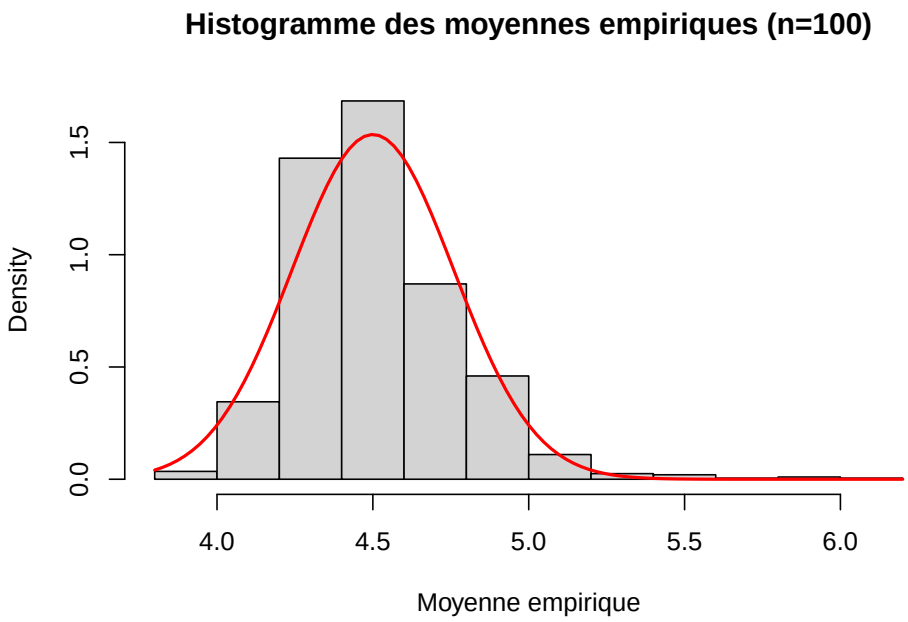
```r
hist(mean_pareto_5, main="Histogramme des moyennes empiriques (n=5)",
     xlab="Moyenne empirique", xlim=c(3,10), ylim=c(0,1), prob=TRUE)
curve(dnorm(x, 4.5, sqrt(6.75/5)), add=TRUE, col="red", lwd=2)
```




```r
hist(mean_pareto_30, main="Histogramme des moyennes empiriques (n=30)",
     xlab="Moyenne empirique", prob=TRUE)
curve(dnorm(x, 4.5, sqrt(6.75/30)), add=TRUE, col="red", lwd=2)
```



```r
hist(mean_pareto_100, main="Histogramme des moyennes empiriques (n=100)",
     xlab="Moyenne empirique", prob=TRUE)
curve(dnorm(x, 4.5, sqrt(6.75/100)), add=TRUE, col="red", lwd=2)
```



La loi théorique de la moyenne empirique est la loi gaussienne

$$\mathcal{N}(\frac{a\alpha}{\alpha-1}, \frac{\sigma}{\sqrt{n}})$$

On utilisera comme paramètres pour la renormalisation

$$a_n = \frac{a\alpha}{\alpha-1}$$

et

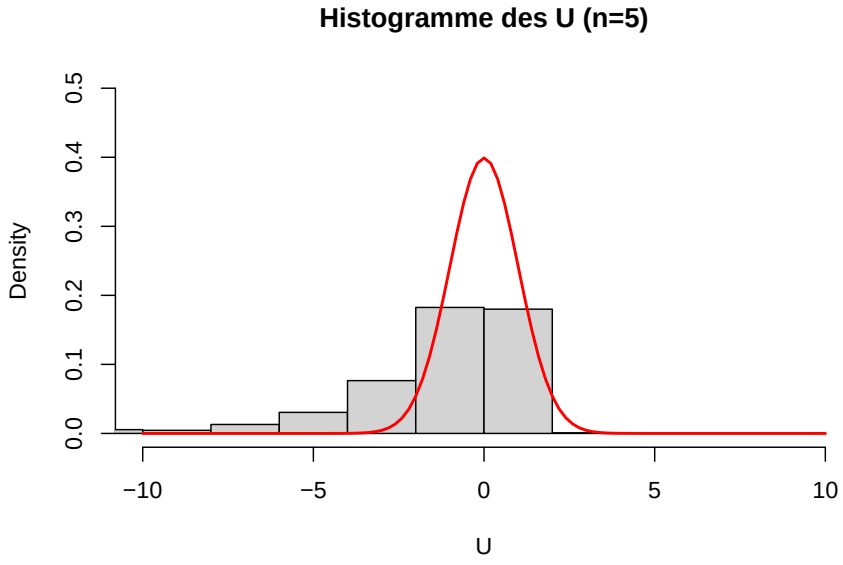
$$b_n = \frac{\sigma_{n,i}}{\sqrt{n}}$$

D'après le théorème central limite, on peut dire que la variable aléatoire

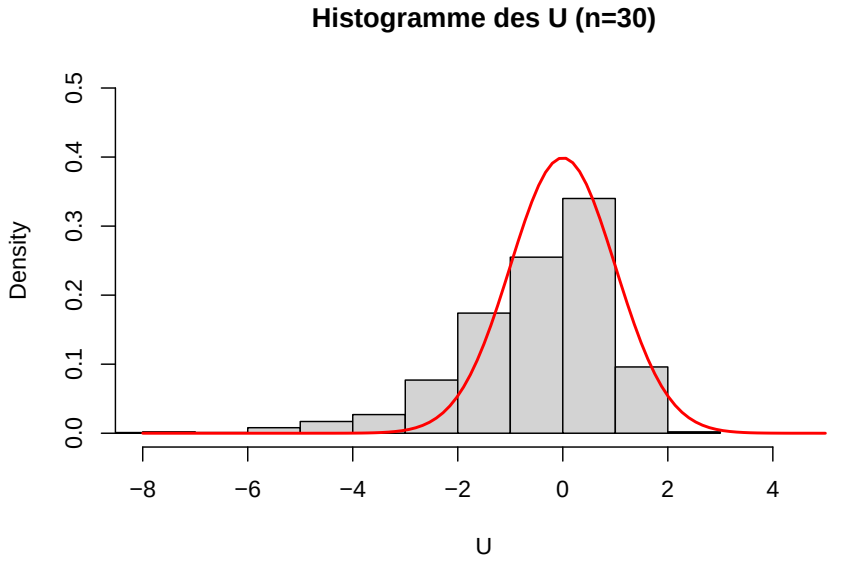
$$U_{n,i} = \frac{\bar{X}_{n,i} - a_n}{b_n} \underset{n\to+\infty}{\overset{loi}{\longrightarrow}} \mathcal{N}(0,1)$$

```r
sd_pareto_5 = apply(pareto_5, 1, sd)
sd_pareto_30 = apply(pareto_30, 1, sd)
sd_pareto_100 = apply(pareto_100, 1, sd)
```

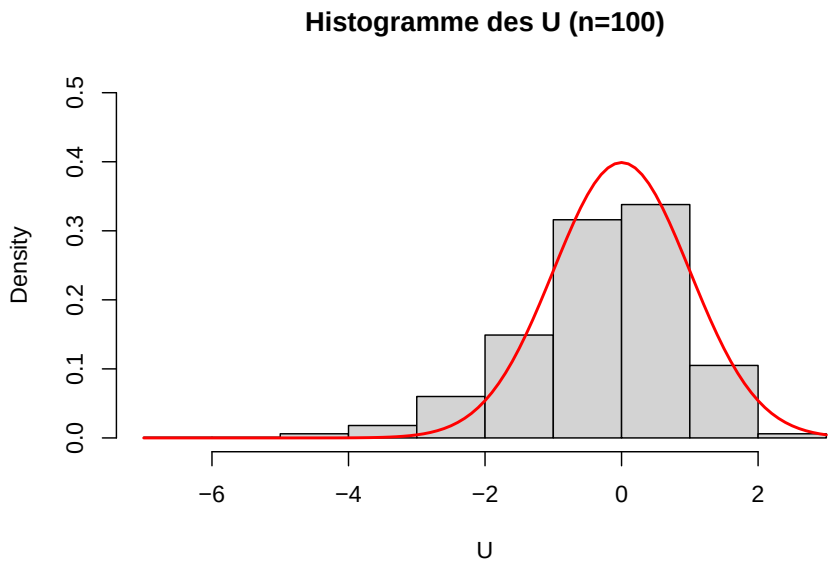
```r
u_pareto_5 = (mean_pareto_5 - 4.5) / (sd_pareto_5 / sqrt(5))
hist(u_pareto_5, main="Histogramme des U (n=5)", xlab="U", ylim=c(0,0.5), xlim=c(-10,10),prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```




```r
u_pareto_30 = (mean_pareto_30 - 4.5) / (sd_pareto_30 / sqrt(30))
hist(u_pareto_30, main="Histogramme des U (n=30)", xlab="U", ylim=c(0,0.5), xlim=c(-8,5), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```



```r
u_pareto_100 = (mean_pareto_100 - 4.5) / (sd_pareto_100 / sqrt(100))
hist(u_pareto_100, main="Histogramme des U (n=100)", xlab="U", ylim=c(0,0.5), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```



On remarque que l'histogramme des $U_{n,i}$ correspond à la densité de la loi normale centrée réduite $\mathcal{N}(0,1)$. Plus la taille de l'échantillon n est grande plus on peut approcher la variable aléatoire $U_{n,i}$ par une loi normale centrée réduite $\mathcal{N}(0,1)$. Ce qui est cohérent avec les résultats du théorème central limite.

Question 3 :

```r
pois_5 = matrix(rpois(5*1000, 10), ncol=5, nrow=1000)
pois_30 = matrix(rpois(30*1000, 10), ncol=30, nrow=1000)
pois_100 = matrix(rpois(100*1000, 10), ncol=100, nrow=1000)
```

```r
mean_pois_5 = apply(pois_5, 1, mean)
mean_pois_30 = apply(pois_30, 1, mean)
mean_pois_100 = apply(pois_100, 1, mean)
```

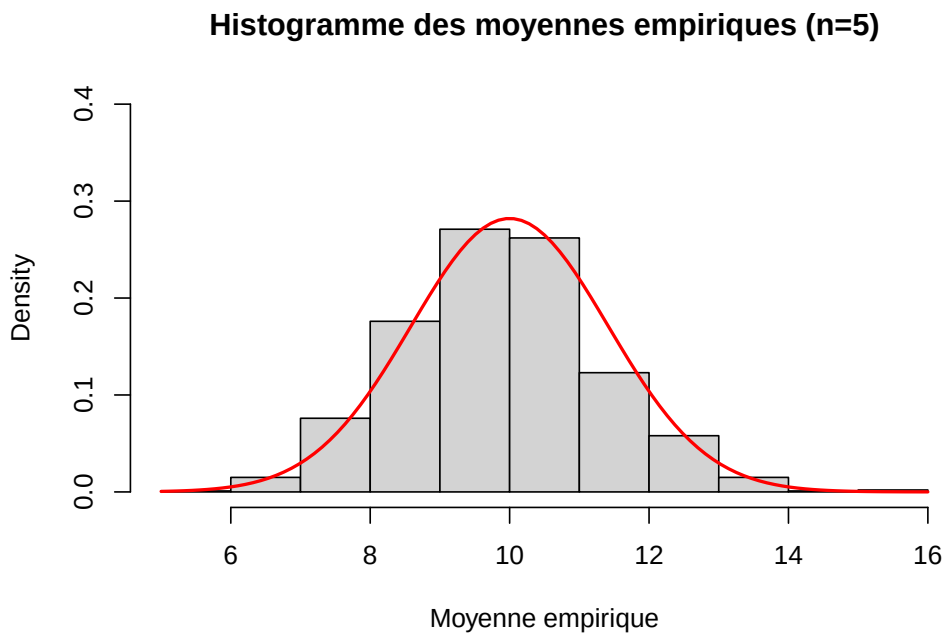
```r
var_pois_5 = apply(pois_5, 1, var)
var_pois_30 = apply(pois_30, 1, var)
var_pois_100 = apply(pois_100, 1, var)
```

Soit X une variable aléatoire qui suit une loi de poisson $\mathcal{P}(\lambda)$ alors

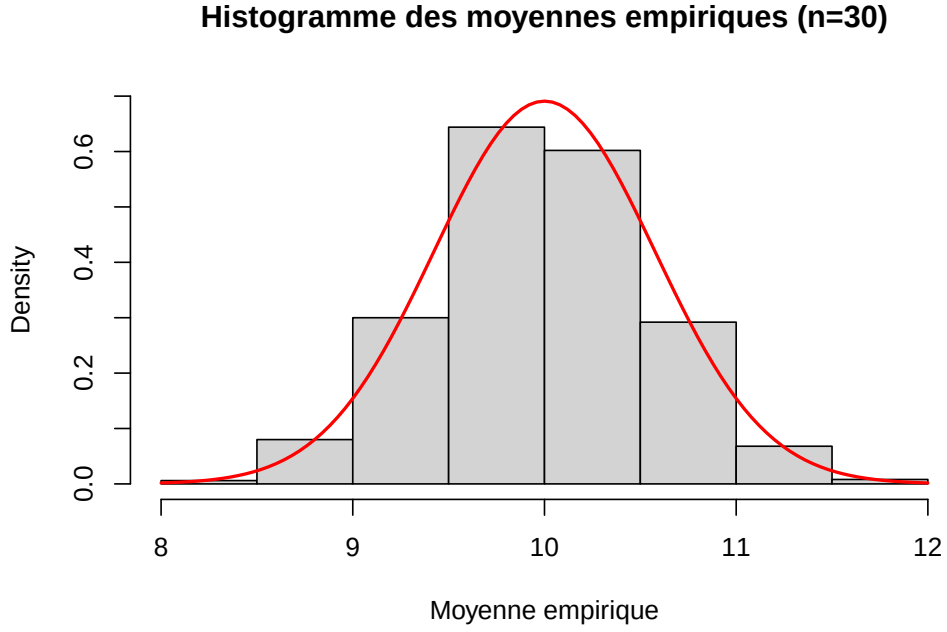
$$\mathbb{E}[X] = \lambda$$

$$\mathbb{V}[X] = \lambda$$

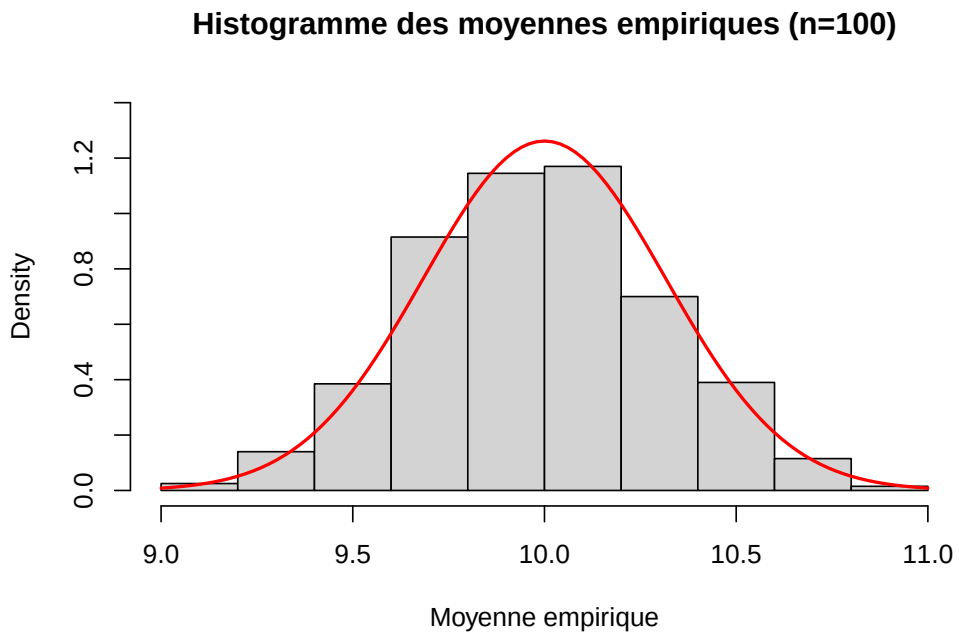
```
hist(mean_pois_5, main="Histogramme des moyennes empiriques (n=5)",
     xlab="Moyenne empirique", ylim=c(0,0.4), prob=TRUE)
curve(dnorm(x, 10, sqrt(10/5)), add=TRUE, col="red", lwd=2)
```




```
hist(mean_pois_30, main="Histogramme des moyennes empiriques (n=30)",
     xlab="Moyenne empirique", ylim=c(0,0.7), prob=TRUE)
curve(dnorm(x, 10, sqrt(10/30)), add=TRUE, col="red", lwd=2)
```



```r
hist(mean_pois_100, main="Histogramme des moyennes empiriques (n=100)",
     xlab="Moyenne empirique", ylim=c(0,1.4), prob=TRUE)
curve(dnorm(x, 10, sqrt(10/100)), add=TRUE, col="red", lwd=2)
```



La loi théorique de la moyenne empirique est la loi normale

$$\mathcal{N}(\lambda, \sqrt{\frac{\lambda}{n}})$$

On utilisera comme paramètres pour la renormalisation

$$a_n = \lambda$$

et

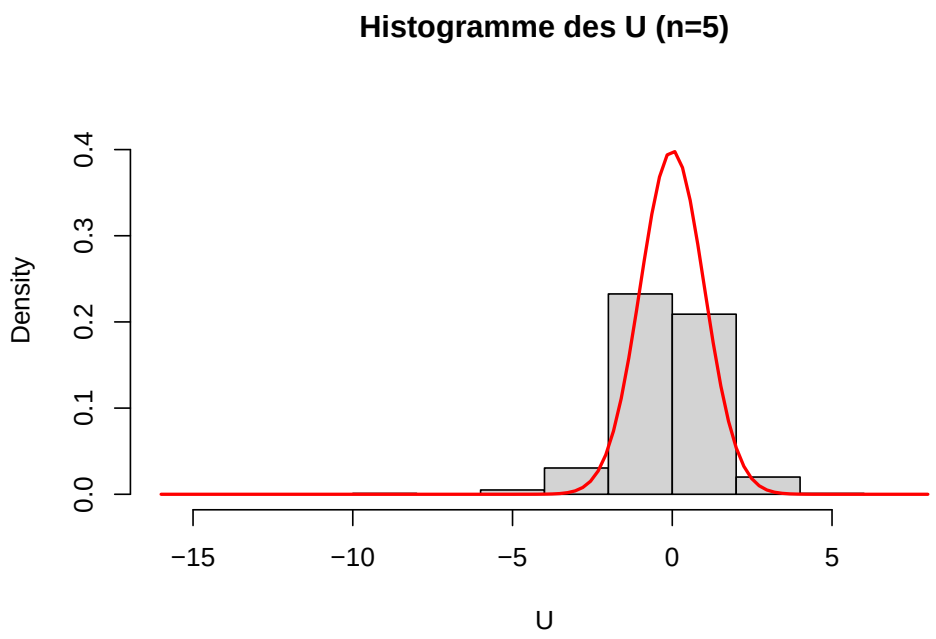
$$b_n = \sqrt{\frac{\lambda}{n}}$$

Daprès le théorème central limite, on peut dire que la variable aléatoire

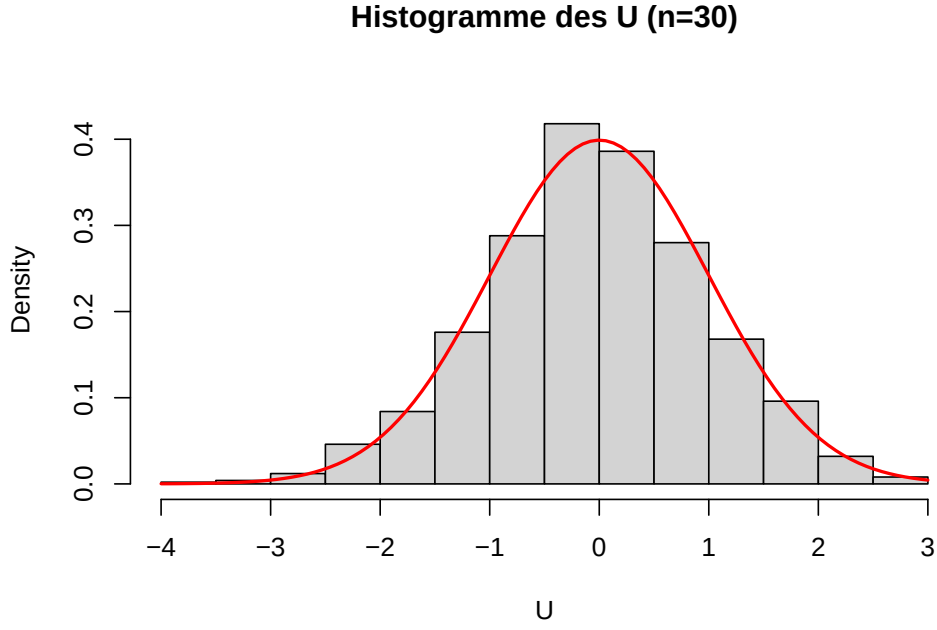
$$U_{n,i} = \frac{\bar{X}_{n,i} - a_n}{b_n} \xrightarrow[n \to +\infty]{loi} \mathcal{N}(0,1)$$

```r
sd_pois_5 = apply(pois_5, 1, sd)
sd_pois_30 = apply(pois_30, 1, sd)
sd_pois_100 = apply(pois_100, 1, sd)
```

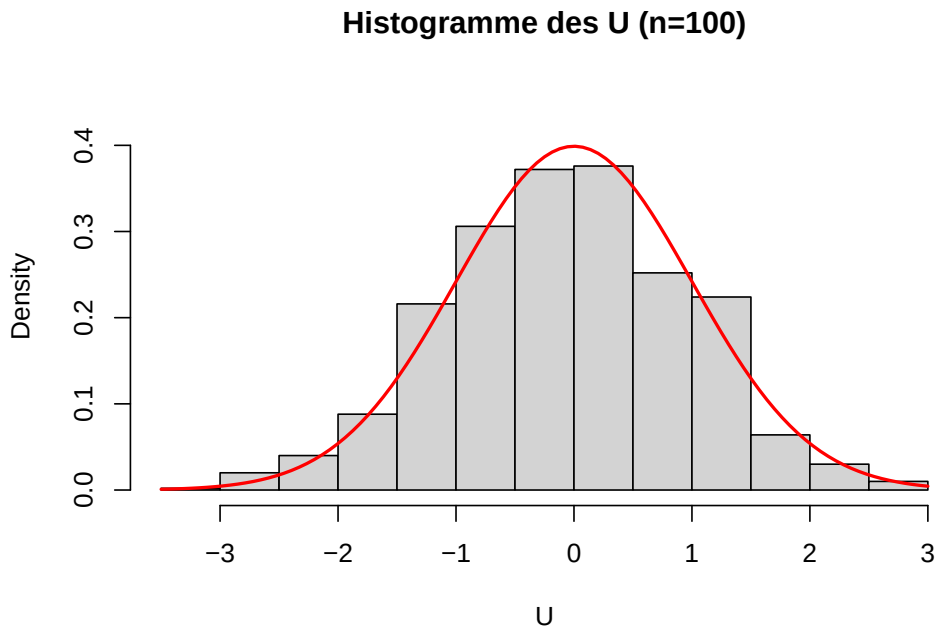
```r
u_pois_5 = (mean_pois_5 - 10) / (sd_pois_5 / sqrt(5))
hist(u_pois_5, main="Histogramme des U (n=5)", xlab="U", ylim=c(0,0.45), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```




```r
u_pois_30 = (mean_pois_30 - 10) / (sd_pois_30 / sqrt(30))
hist(u_pois_30, main="Histogramme des U (n=30)", xlab="U", ylim=c(0,0.45), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```



```r
u_pois_100 = (mean_pois_100 - 10) / (sd_pois_100 / sqrt(100))
hist(u_pois_100, main="Histogramme des U (n=100)", xlab="U", ylim=c(0,0.45), prob=TRUE)
curve(dnorm(x, 0, 1), add=TRUE, col="red", lwd=2)
```



On remarque que l'histogramme des $U_{n,i}$ correspond à la densité de la loi normale centrée réduite $\mathcal{N}(0,1)$. Plus la taille de l'échantillon n est grande plus on peut approcher la variable aléatoire $U_{n,i}$ par une loi normale centrée réduite $\mathcal{N}(0,1)$. Ce qui est cohérent avec les résultats du théorème central limite.

Question 4 :

On déduit de ce qui précède que la méthode pour estimer l'espérance d'une statistique d'un échantillon est d'approcher la loi théorique par la loi gaussienne suivante :

$$\mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$$

Avec μ qui représente l'espérance, σ l'écart-type et n la taille de l'échantillon.

Plus la taille de l'échantillon est grande, plus l'approximation est bonne. C'est ce qu'on remarque lorsque l'on compare les résultats entre les échantillons de taille 5, 30 et 100.

Maximum de vraisemblance

Ajuster une loi de Bernoulli

```
binom_10 = rbinom(10, 1, 0.7)
binom_10
```

```
 [1] 0 0 1 1 1 0 1 1 1 1
```

Question 5 :

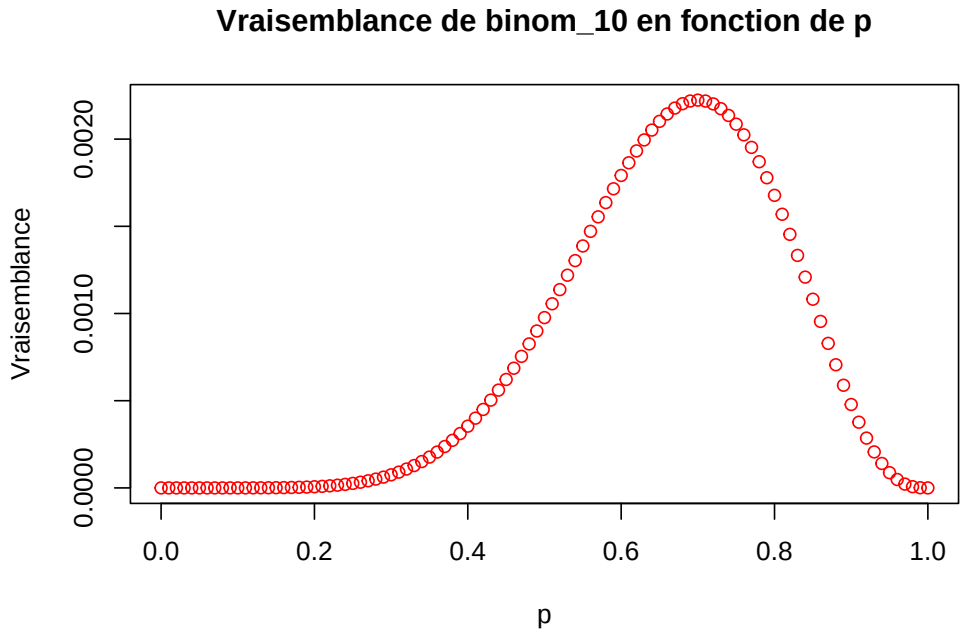
On peut estimer p en faisant le rapport entre le nombre de réussite et le nombre d'essai.

Question 6 :

```
# @param p probabilité de réussite
# @param x échantillon
# @return vraisemblance de l'échantillon x pour la valeur p
L_bern = function(p, x) return(p^sum(x)*(1-p)^(length(x)-sum(x)))
```

Question 7 :

```
range_p = seq(from=0, to=1, by=0.01)
vrai_binom_10 = lapply(range_p, function(range_p) {L_bern(range_p, binom_10)})
plot(range_p, vrai_binom_10, main="Vraisemblance de binom_10 en fonction de p",
     xlab="p", ylab="Vraisemblance", col="red")
```



Question 8 :

```
value_p = optimize(function (range_p) L_bern(range_p, binom_10), c(0, 1), maximum=TRUE)$maximum
value_p
```

```
[1] 0.6999843
```

Question 9 :

```
ecart_bernoulli = data.frame()
for (n in seq(from=100, to=2000, by=100)) {
  echantillon = rbinom(n, 1, 0.7)
  value_p = optimize(function (range_p) L_bern(range_p, echantillon), c(0, 1), maximum=TRUE)$maximum
  ecart_bernoulli = rbind(ecart_bernoulli, c(n, 0.7 - value_p))
}
colnames(ecart_bernoulli) = c("Taille", "Ecart")
kable(ecart_bernoulli)
```

Taille	Ecart
100	0.0199909
200	0.0399983
300	-0.0033318
400	-0.0199984
500	0.0300081
600	0.0033301
700	0.0128688
800	-0.0124920
900	0.0022249
1000	0.0060184
1100	0.0100008
1200	-0.0191528
1300	-0.2999339
1400	-0.2999339
1500	-0.2999339
1600	-0.2999339
1700	-0.2999339
1800	-0.2999339
1900	-0.2999339
2000	-0.2999339

On remarque que l'écart ne diminue pas forcément lorsque la taille de l'échantillon augmente. De plus, à partir d'une certaine taille d'échantillon l'écart devient constant même si l'on augmente la taille.

Pour combattre l'instabilité numérique due aux multiplications de probabilités, il suffit d'utiliser la fonction logarithme pour éviter de calculer les puissances directement et passer plutôt par des sommes.

Ajuster une loi normale d'écart-type connu

```
norm_10 = rnorm(10, 2, 1)
norm_10
```

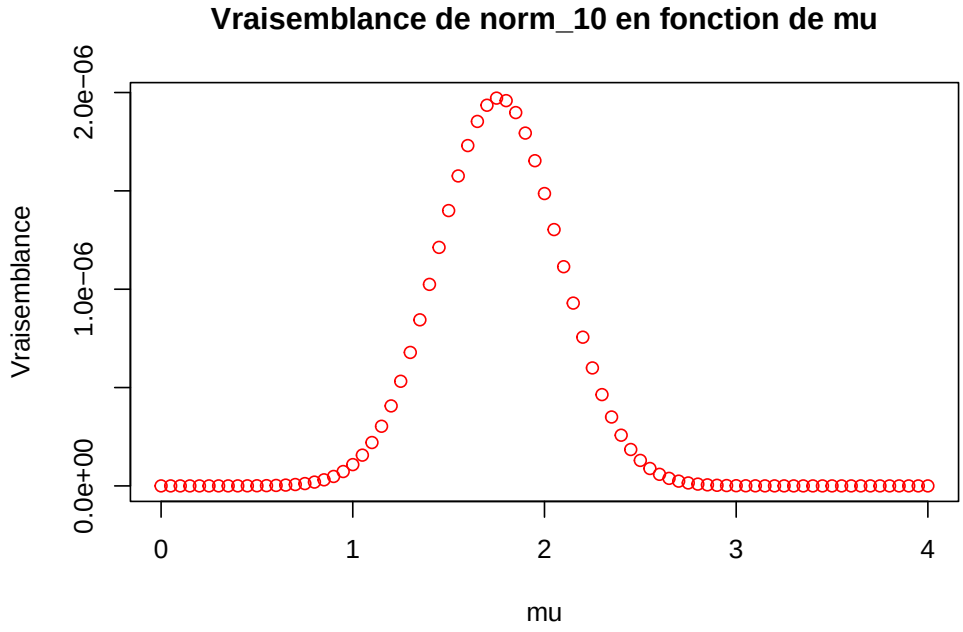
```
 [1]  1.3242471  2.2471998  1.3490273  2.4147256  2.1728357  1.4726527
 [7]  1.3626692 -0.2280453  2.2348205  3.2699031
```

Question 10 :

```
# @param mu espérance
# @param x échantillon
# @return vraisemblance de l'échantillon x pour la valeur mu
L_norm <- function(mu, x) {
  vrai = 1
  for (i in 1:length(x)) vrai = vrai * dnorm(x[i], mu)
  return(vrai)
}
```

Question 11 :

```
range_mu = seq(from=0, to=4, by=0.05)
vrai_norm_10 = lapply(range_mu, function(range_mu) {L_norm(range_mu, norm_10)})
plot(range_mu, vrai_norm_10, main="Vraisemblance de norm_10 en fonction de mu",
     xlab="mu", ylab="Vraisemblance", col="red")
```



Question 12 :

```
value_mu = optimize(function (range_mu) L_norm(range_mu, norm_10), c(0, 4), maximum=TRUE)$maximum
value_mu
```

```
[1] 1.762009
```

Question 13 :

```
L_log_norm <- function(mu, x){
  vrai = 1
  for (i in 1:length(x)){
    vrai = vrai + log(dnorm(x[i], mu))
  }
  return(vrai)
}
ecart_normale = data.frame()
for (n in seq(from=100, to=2000, by=100)) {
  echantillon = rnorm(n, 2, 1)
  value_mu = optimize(function (range_mu) L_log_norm(range_mu, echantillon),
                      c(0, 4), maximum=TRUE)$maximum
  ecart_normale = rbind(ecart_normale, c(n, 2 - value_mu))
}
colnames(ecart_normale) = c("Taille", "Ecart")
kable(ecart_normale)
```

Taille	Ecart
100	-0.0510496
200	-0.0964333
300	0.0815083
400	0.0459360
500	-0.0747452
600	-0.0397652
700	-0.0059830
800	0.0001537
900	-0.0492477
1000	-0.0076969
1100	-0.0470012
1200	0.0109054
1300	-0.0059874
1400	0.0304548
1500	0.0191635
1600	0.0187210
1700	0.0279724
1800	0.0198280
1900	-0.0066634
2000	0.0023672

On modifie la fonction L_norm en L_log_norm en utilisant la fonction logarithme afin de remplacer les multiplications par des sommes pour éviter les instabilités liée s à la puissance de calcul de la machine.

Question 14 :

```
summer_ozone = read.csv("summer_ozone.csv")
winter_ozone = read.csv("winter_ozone.csv")
summer_neuil = summer_ozone$NEUIL
summer_rur = summer_ozone$RUR.SE
winter_neuil = winter_ozone$NEUIL
winter_rur = winter_ozone$RUR.SE
```

```
L_norm_bis <- function(mu, sd, x){
  vrai = 1
  for (i in 1:length(x)){
    vrai = vrai + log(dnorm(x[i], mu, sd))
  }
  return(vrai)
}
```

```
range_param = seq(from=10, to=100, by=0.5)
moy_sd = mean(c(sd(summer_neuil), sd(summer_rur), sd(winter_neuil), sd(winter_rur)))

value_sn = optimize(function (range_param) L_norm_bis(range_param, moy_sd, summer_neuil),
                    c(0, 100), maximum=TRUE)$maximum
value_sr = optimize(function (range_param) L_norm_bis(range_param, moy_sd, summer_rur),
                    c(0, 100), maximum=TRUE)$maximum
value_wn = optimize(function (range_param) L_norm_bis(range_param, moy_sd, winter_neuil),
                    c(0, 100), maximum=TRUE)$maximum
value_wr = optimize(function (range_param) L_norm_bis(range_param, moy_sd, winter_rur),
                    c(0, 100), maximum=TRUE)$maximum

max_vrai = t(data.frame(value_sn, value_sr, value_wn, value_wr))
rownames(max_vrai) = c("Summer NEUIL", "Summer RUR.SE", "Winter NEUIL", "Winter RUR.SE")
colnames(max_vrai) = "Estimateur du maximum de vraisemblance"
kable(max_vrai)
```

	Estimateur du maximum de vraisemblance
Summer NEUIL	86.70265
Summer RUR.SE	92.91853
Winter NEUIL	39.07127
Winter RUR.SE	57.55508

En faisant l'hypothèse que l'écart-type est le même pour les quatres jeux de données, on remarque que l'on retrouve bien les bonnes moyennes empiriques. On en déduit qu'il est raisonnable de fixer le paramètre d'écart-type à la moyenne des écart-type de nos jeux de données. En effet, le calcul de l'estimateur du maximum de vraisemblance est peu sensible à la variation de ce dernier, sachant que l'on retrouve exactement les mêmes valeurs pour des écart-type contenu dans l'intervalle [5, 200].

L'écart-type étant une mesure de la dispersion des valeurs d'un jeu de donnée, il semble logique de prendre la même valeur pour l'écart-type utilisé dans la fonction L_norm_bis, puisque les valeurs des écart-type des jeux de données sont assez proches les unes des autres. Cela est cohérent avec la réalité étant donnée que l'on compare des concentrations maximales en ozone dont la disparité n'est pas si différente selon la saison ou la régoin (rurale ou urbaine). (cf résultats du TP 2)