Compte-rendu d'analyse

valant évaluation dans le cadre de :

Diplôme : Master statistique pour l'évaluation et prévision, 1ère année
Année universitaire : 2022-2023
Module d'enseignement : MA0814
Responsable : Djamal LOUANI
Comptant pour : 25 %

Compte-rendu de TP Statistique Inferentielle

Elif ERTAS elif.ertas@etudiant.univ-reims.fr
Fatima AAGOUR fatima.aagour@etudiant.univ-reims.fr

Métriques

Finalisé le : 28 avril 2023
Page(s) : 27
Références(s) : 0
Figure(s) : 0
Table(s) : 0
Théorème(s) : 0

Résumé : L'objet de ce TP est de voir des aspects pratiques de l'inférence statistique. Des estimations et des tests sur des données simulées et des données réelles sont au programme. Les programmes informatiques seront faits avec le logiciel R.

Mots-clés : Statistique, Probabilité, Estimation.

Matériel supplémentaire :
git: https://github.com/pregnault/urcadown
: https://cours.univ-reims.fr/course/view.php?id=16431

Table des matières

1. Definitions et tirages d'échantillons

1.1. Loi de Bernouilli

La loi de Bernoulli est une loi de probabilité qui décrit le résultat d'une expérience aléatoire ayant deux issues possibles, généralement appelées "succès" et "échec". Elle est utilisée pour calculer la probabilité d'obtenir un certain nombre de succès dans un nombre fixe de tentatives indépendantes les unes des autres, chacune ayant la même probabilité de succès.

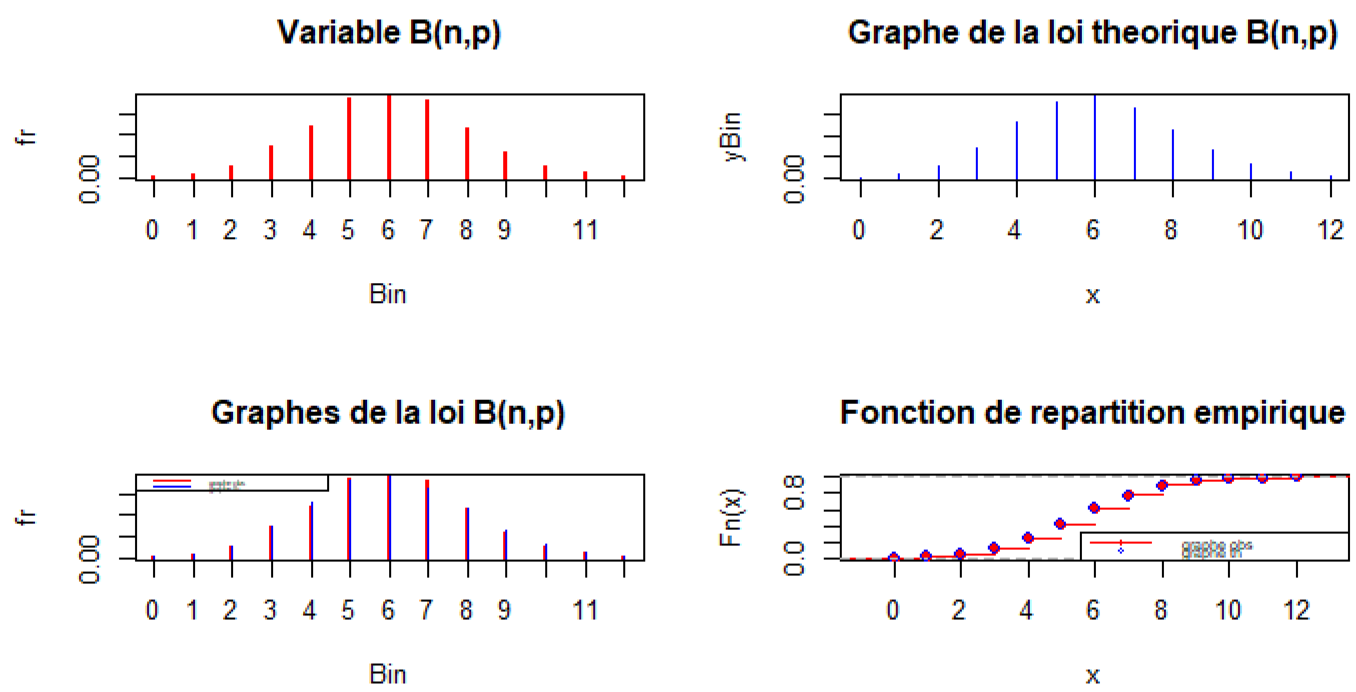
Définition : La loi de Bernoulli est définie mathématiquement comme suit :

Soit X une variable aléatoire qui prend deux valeurs possibles : 1 pour un succès et 0 pour un échec, avec une probabilité de succès p et une probabilité d'échec $q = 1 - p$. La fonction de masse de probabilité de X est alors donnée par :

$$p_X(x) \begin{cases} p \qquad si\ x = 1 \\ 1 - p = q \qquad si\ x = 0 \end{cases}$$

Cette loi permet de calculer la probabilité d'obtenir k succès dans n essais indépendants, avec une probabilité de succès p, en utilisant la formule de la loi binomiale, qui est basée sur la loi de Bernoulli.
La fonction de répartition d'une variable X ~ B(p):

$$F_X(x) = \begin{cases} 0\ si\ x < 0 \\ 1 - p\ si\ 0 \leq x < 1 \\ 1\ si\ x \geq 1 \end{cases}$$



1.2. Loi de Poisson

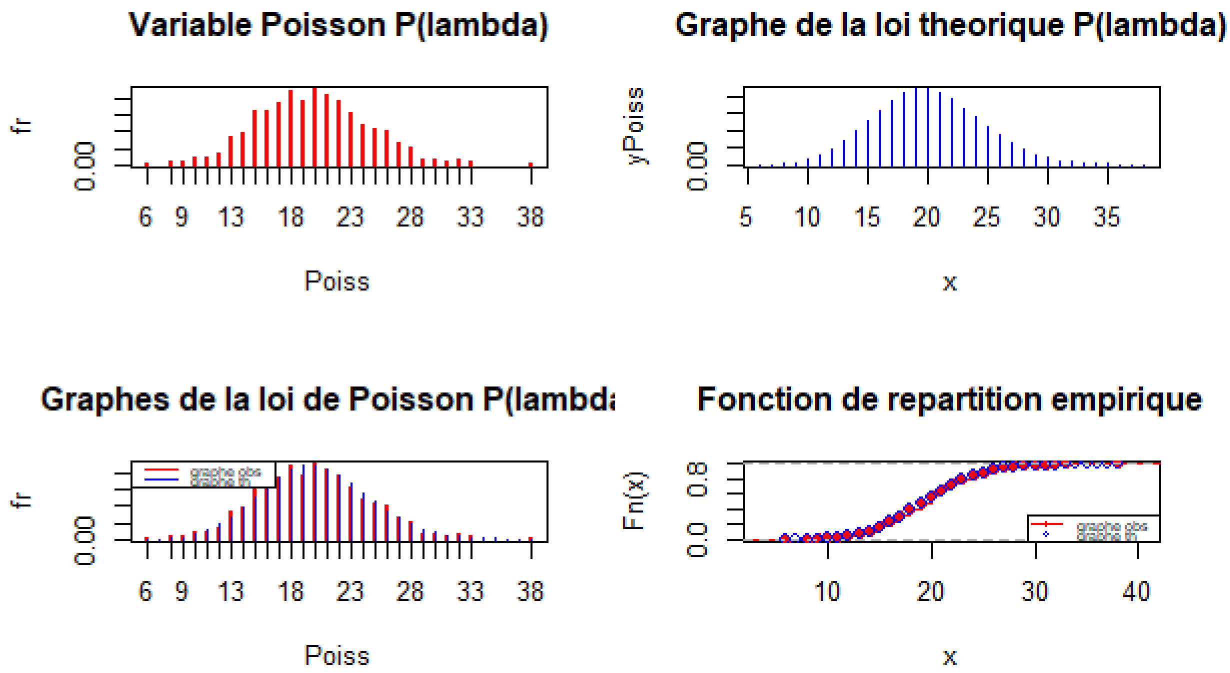
La loi de Poisson est une distribution de probabilité discrète qui permet de modéliser le nombre d'événements rares qui se produisent dans une période de temps donnée, sachant que la moyenne de ces événements est constante.

Définition : La loi de Poisson est caractérisée par un seul paramètre, appelé "lambda", qui représente la moyenne et la variance de la distribution.
Pour un événement rare ayant une fréquence moyenne λ par unité de temps (ou d'espace), la probabilité que k événements se produisent dans cette unité de temps (ou d'espace) est donnée par la formule :

$$P(X = k) = \frac{(e^{(-\lambda)} * \lambda^k)}{k!}$$

La fonction de répartition d'une variable X ~ P(λ):

$$F(k, \lambda) = P(X \leq k) = \sum_{i=0}^{k} \frac{e^{-\lambda} \lambda^i}{i!}$$



1.3. Loi Exponentielle

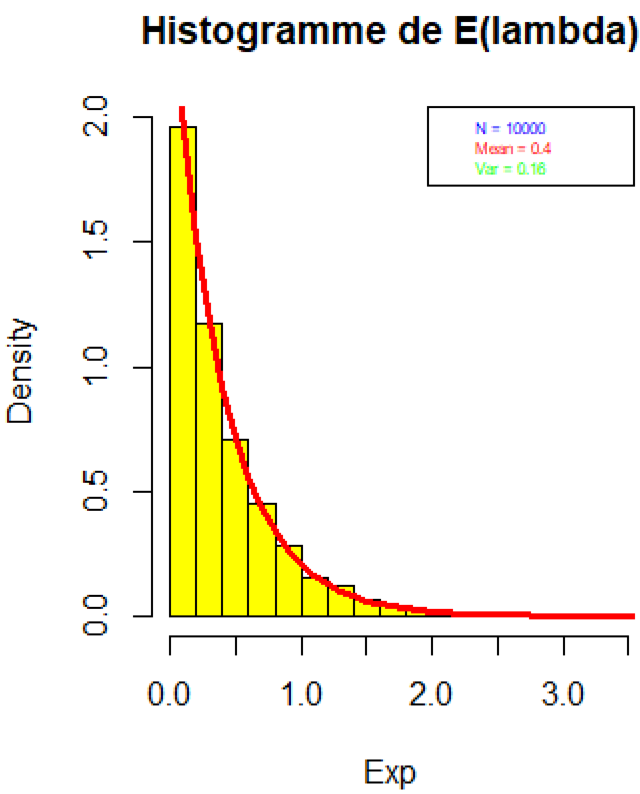
La loi exponentielle est une loi de probabilité continue qui modélise le temps entre deux événements rares et indépendants, tels que les temps d'attente ou les temps de réponse. Elle est caractérisée par un seul paramètre λ qui représente la fréquence moyenne des événements.

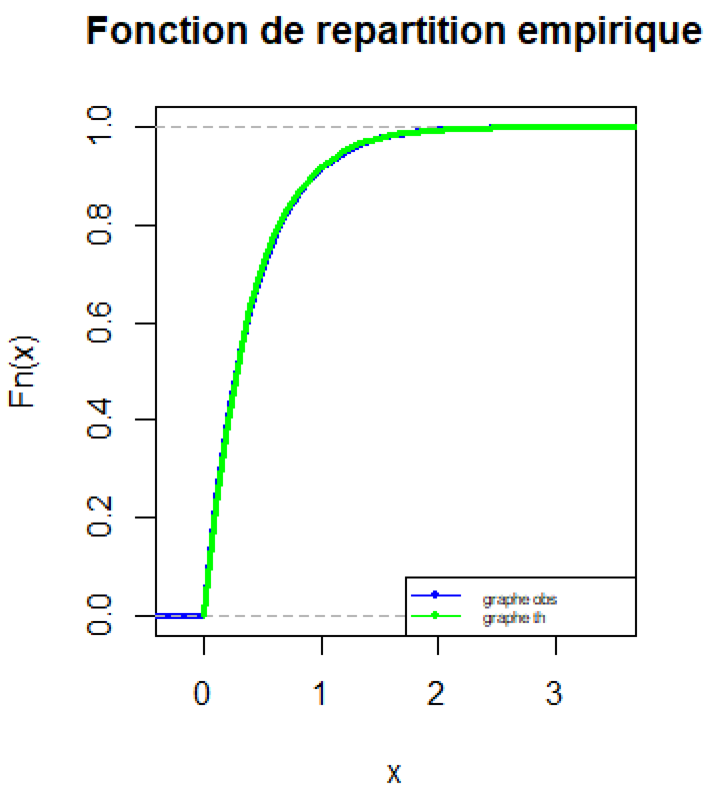
Définition : La densité de probabilité de la loi exponentielle est donnée par la formule :

$$f(x, \lambda) = \lambda * e^{(-\lambda x)}$$

La fonction de répartition de la loi exponentielle est donnée par la formule :

$$F(x, \lambda) = 1 - e^{(-\lambda x)} \quad \forall x \geq 0$$



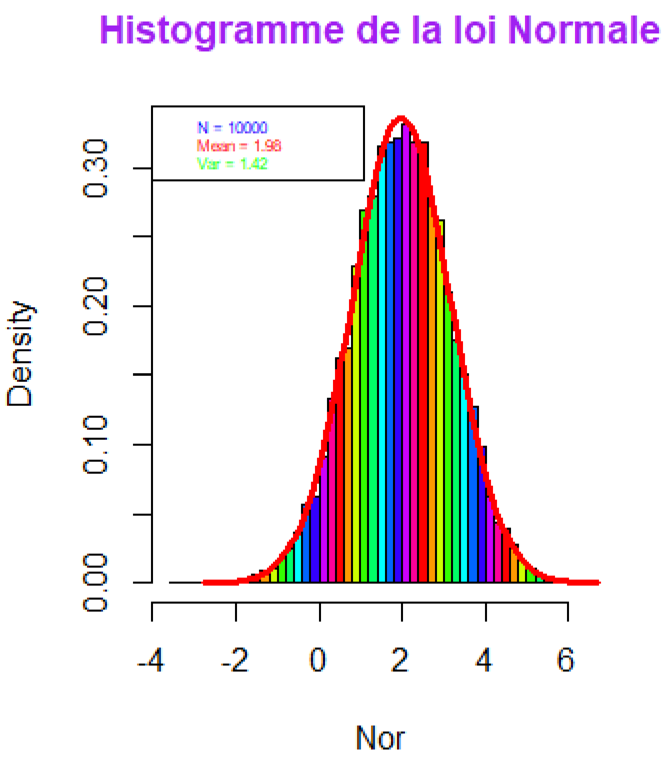


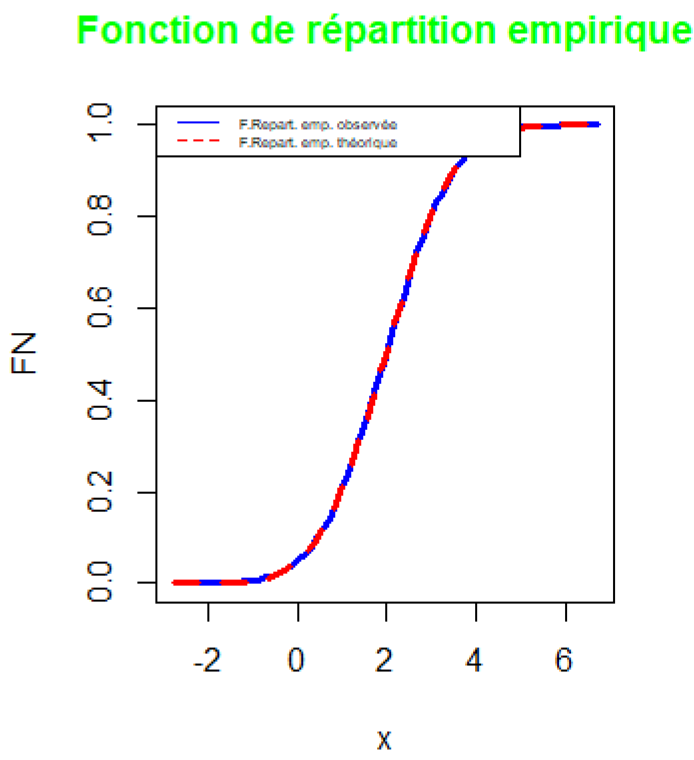
1.4. Loi Normale

La loi normale est une distribution continue en forme de cloche, caractérisée par sa moyenne et son écart type, et souvent utilisée pour modéliser des phénomènes naturels.

Définition : La densité de probabilité de la loi normale est donnée par la formule :

$$f(x;\mu,\sigma) = \frac{1}{\sigma * \sqrt{2\pi}} * exp(\frac{-(x-\mu)^2}{2\sigma^2})$$



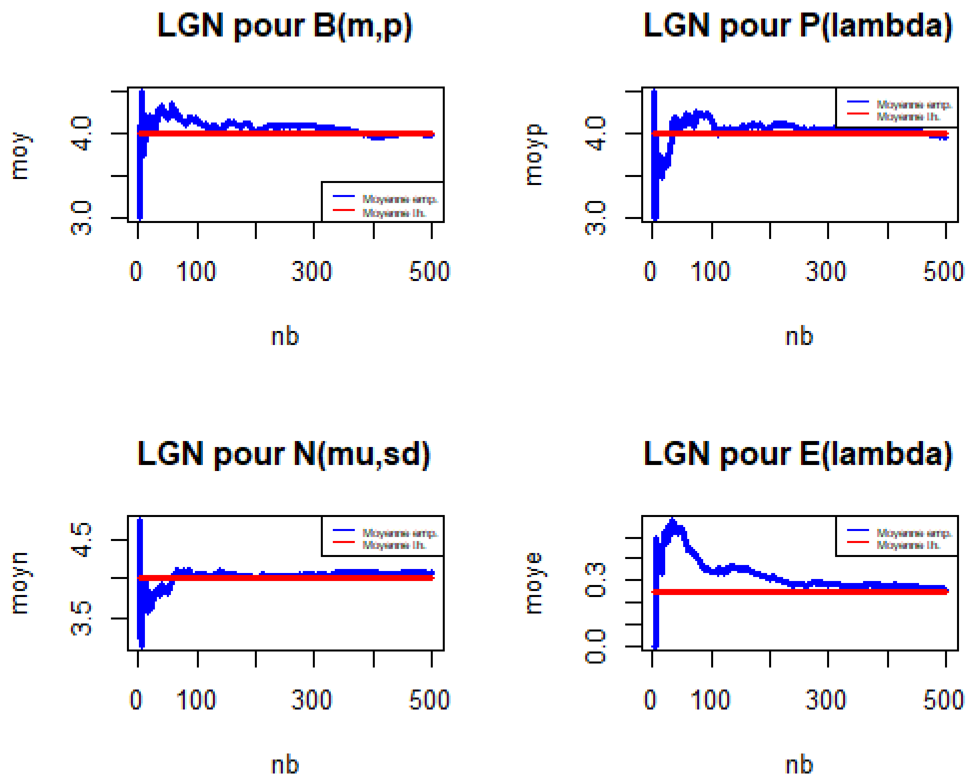


2. Loi des grands nombres

Définition : La loi des grands nombres est un théorème de probabilité qui affirme que, lorsque le nombre d'observations d'une expérience aléatoire est suffisamment grand, la moyenne empirique des résultats converge presque sûrement vers la moyenne théorique attendue. En d'autres termes, plus on répète une expérience aléatoire, plus la moyenne des résultats s'approche de la moyenne théorique attendue.

Si $X_1, X_2, ..., X_n$ sont n échantillons indépendants et identiquement distribués, avec une espérance finie μ et une variance finie σ^2, alors :

$$\lim_{n \to \infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| > \varepsilon\right) = 0 \quad \forall \varepsilon > 0$$



On remarque qu'au début du graphique les oscillations sont plus fortes mais avec l'effet dû loi des grands nombres, la moyenne empirique converge vers l'espérance.

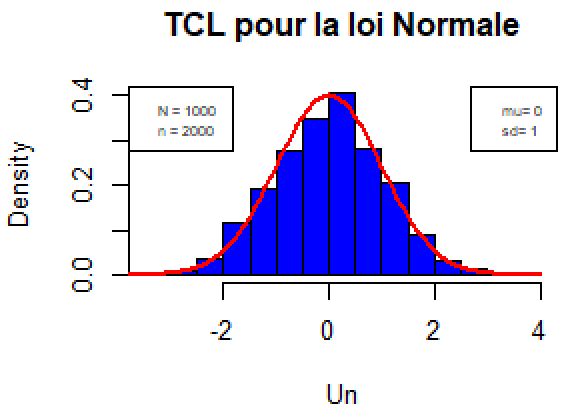
3. Théorème central limite

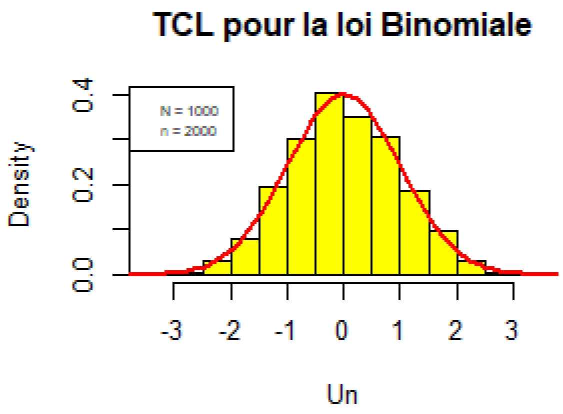
Le théorème central limite est un concept clé en statistique qui énonce que la somme de variables aléatoires indépendantes et identiquement distribuées tend vers une distribution normale, indépendamment de leur distribution initiale. En d'autres termes, si vous ajoutez un grand nombre de variables aléatoires ayant la même distribution, leur somme aura une distribution normale, même si les variables individuelles ne suivent pas une loi normale. Ce théorème est largement utilisé en statistique pour comprendre la distribution d'échantillons et pour estimer des paramètres de population à partir d'échantillons.

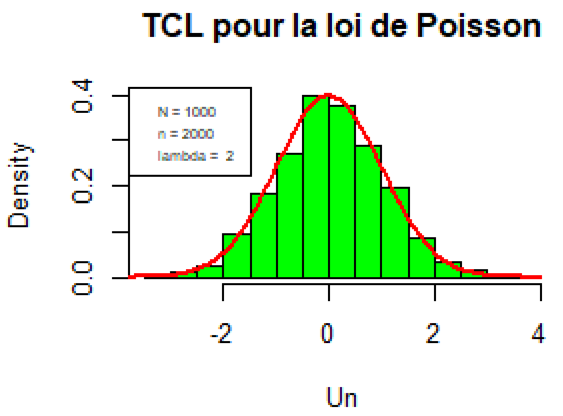
Définition : Soit $X_1, X_2, ..., X_n$ une suite de variables aléatoires indépendantes et identiquement distribuées avec une moyenne μ et un écart type σ finis. Alors, pour n suffisamment grand, la somme des variables aléatoires suit approximativement une loi normale de moyenne $n * \mu$ et d'écart type $\sigma\sqrt{n}$, c'est-à-dire :

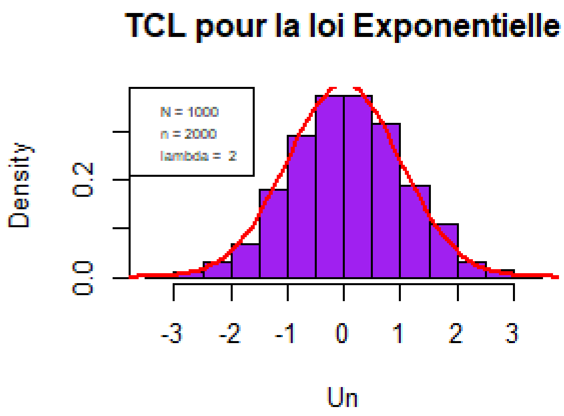
$$\frac{S_n - n\mu}{\sqrt{n}\sigma} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

où $S_n = X_1 + X_2 + ... + X_n$ est la somme des variables aléatoires, et $\mathcal{N}(0,1)$ est une loi normale standardisée.









On remarque que les histogrammes ont la même forme que la courbe rouge représentant la densité d'une loi normale centrée réduite. On a donc bien une vérification visuelle de la définition.

4. Estimation ponctuelles des paramètres

L'estimation ponctuelle des paramètres d'une loi normale consiste à estimer les valeurs des paramètres de la loi normale, tels que la moyenne μ et l'écart type σ, à partir d'un échantillon de données observées.

L'estimateur de la moyenne empirique de l'échantillon est simplement la moyenne arithmétique des observations dans l'échantillon, soit :

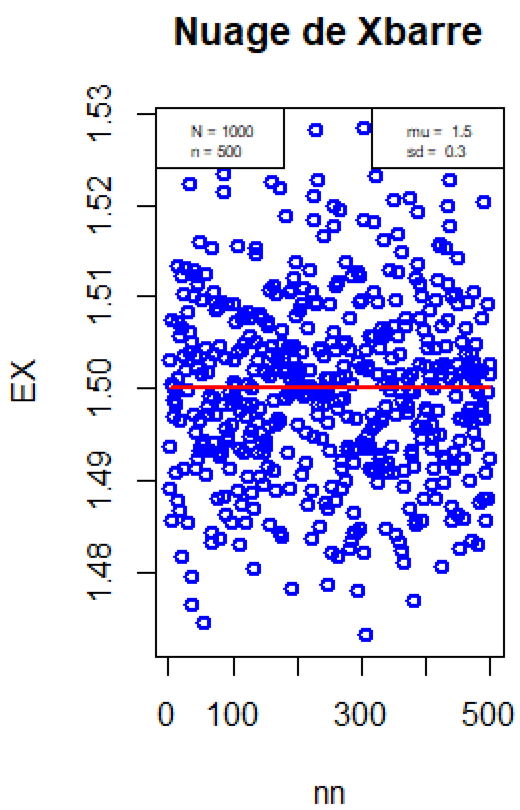
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

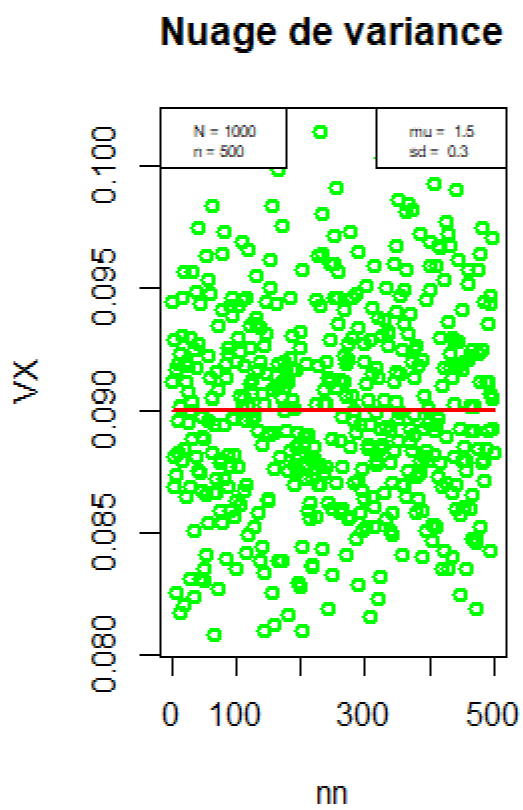
où $\hat{\mu}$ est l'estimateur de la moyenne, n est la taille de l'échantillon, et X_i est la i-ème observation dans l'échantillon.

L'estimateur de l'écart type empirique corrigé est calculé à partir de la variance empirique de l'échantillon, qui est donnée par :

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

où $\hat{\sigma}^2$ est l'estimateur de la variance, n est la taille de l'échantillon, X_i est la i-ème observation de l'échantillon, et $\bar{X}$ est la moyenne empirique de l'échantillon.





On constate que plus N est grand est plus le nuage de point se reserre sur la droite rouge, c'est-à-dire que l'estimation s'approche plus de la valeur théorique.

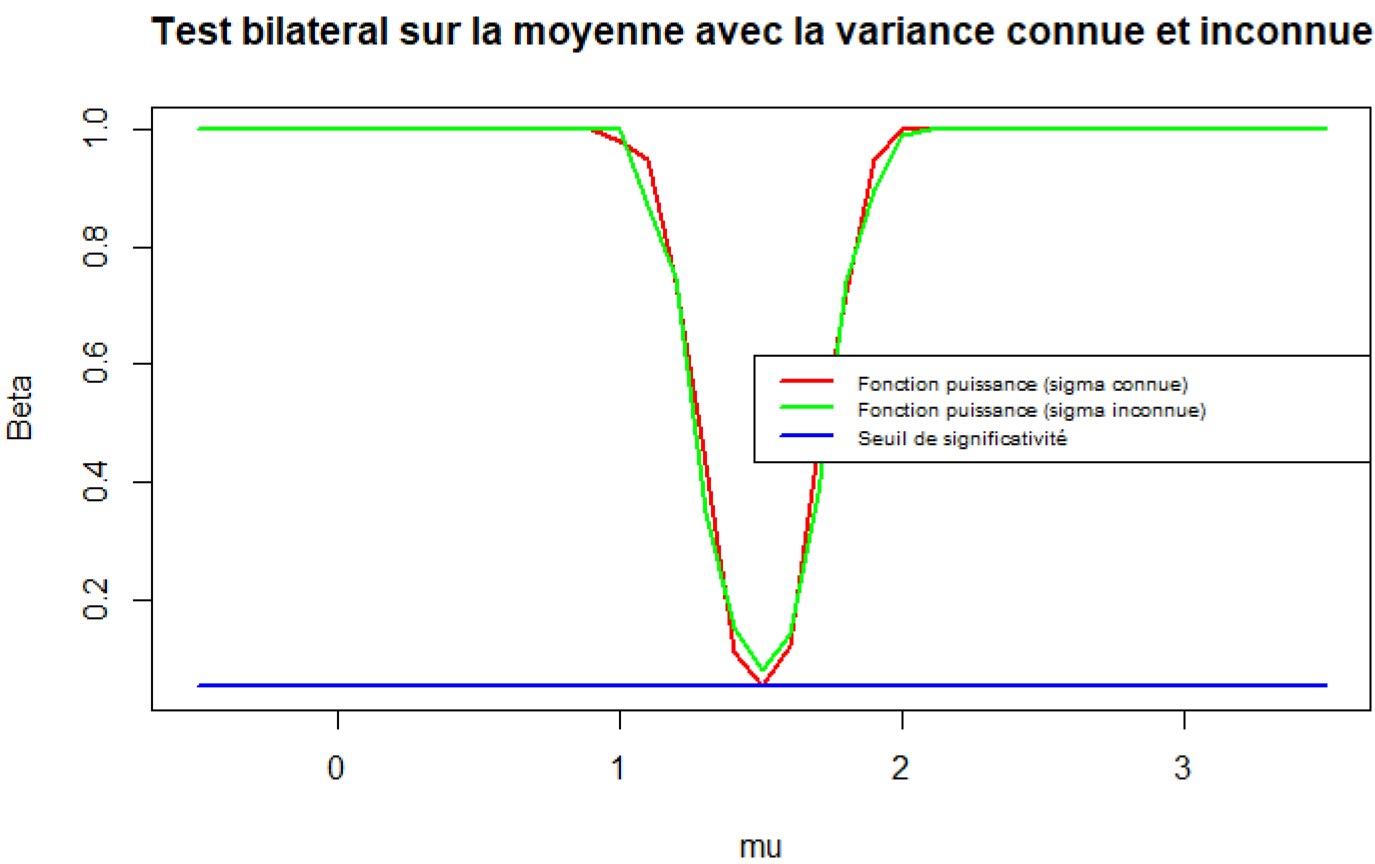
5. Tests

Soient H_0 et H_1 deux parties disjointes de $\mathcal{H}$ ne formant pas nécessairement une partition de $\mathcal{H}$. On appelle *test* entre H_0 et H_1 toute application de $\mathbb{R}^n$ dans $\{H_0, H_1\}$.

L'erreur de première espèce correspond à la probabilité de rejeter H_0 alors qu'elle est vraie ($P_{H_0}(R)$ où R est la région de rejet).

5.1. Test bilatéral

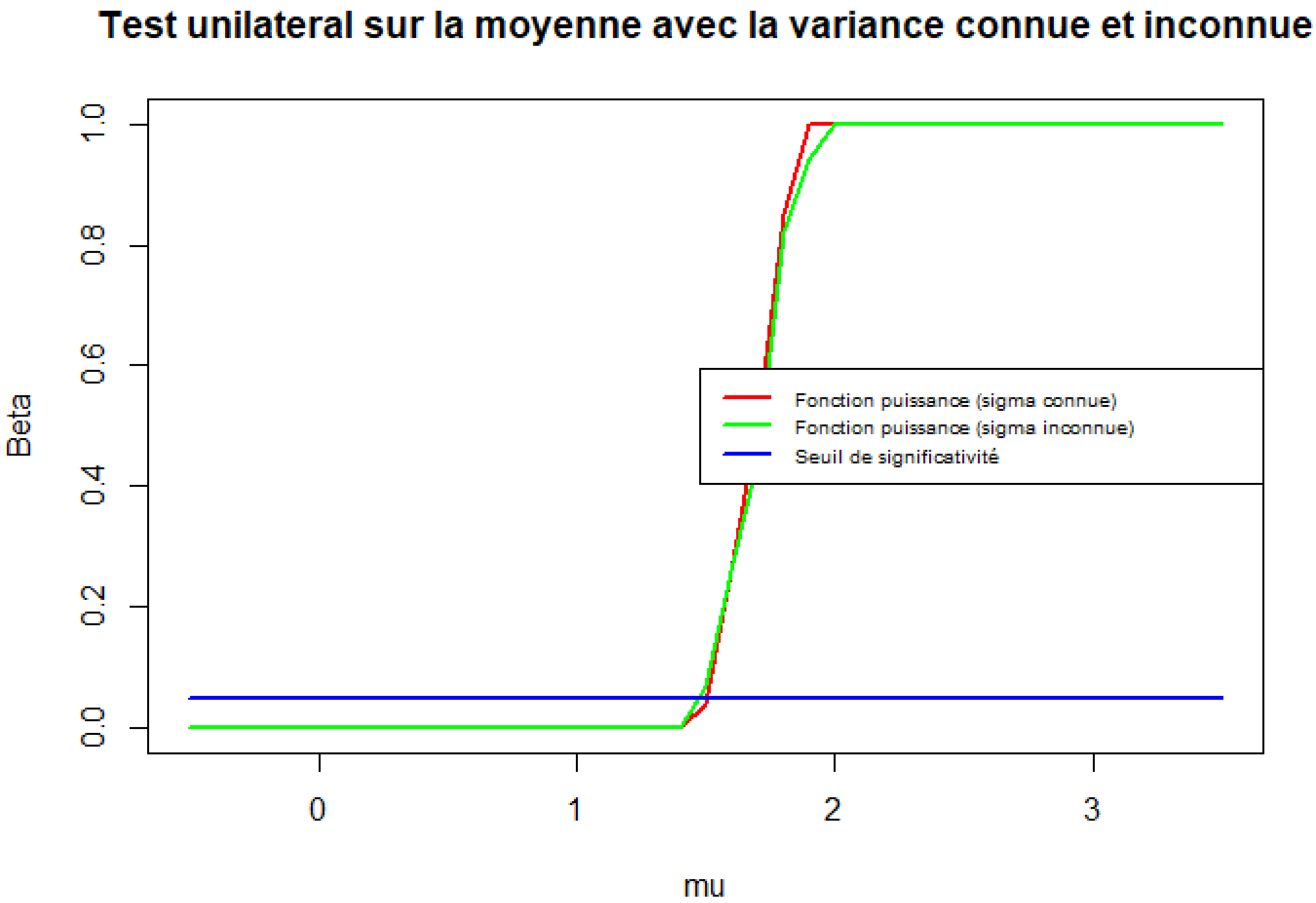
On teste $H_0 : \mu = \mu_0 \quad contre \quad H_1 : \mu \neq \mu_0$.



On observe que lorsqu'on connait sigma la convergence est plus rapide (pente + raide).

5.2. Test unilatéral

On teste $H_0 : \mu \leq \mu_0 \quad contre \quad H_1 : \mu > \mu_0$.



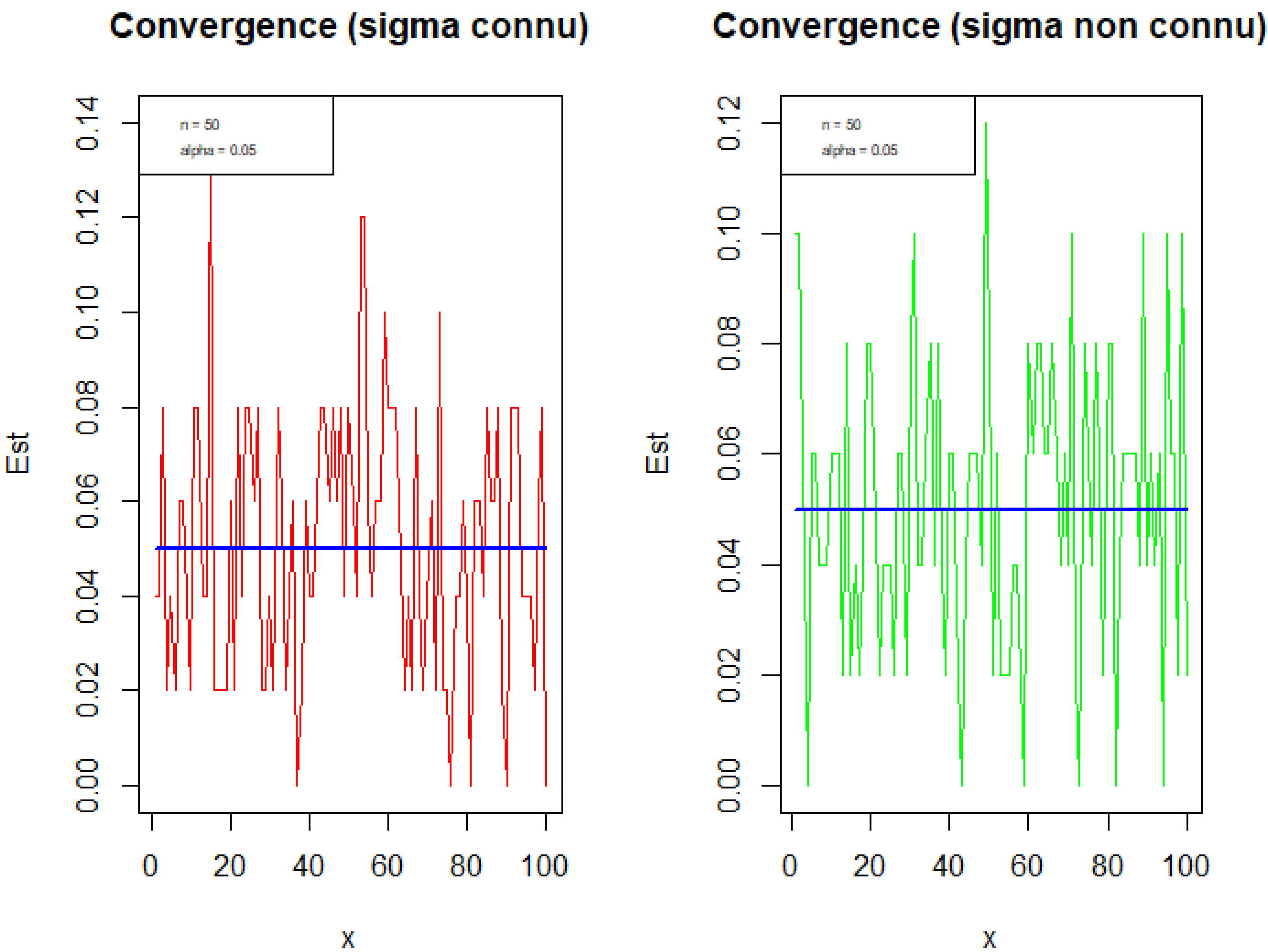
On observe que lorsqu'on connait sigma la convergence est plus rapide (pente + raide).

6. Intervalles de confiance

Un intervalle de confiance est une plage de valeurs dans laquelle on estime qu'un paramètre inconnu est susceptible de se situer avec une certaine probabilité, basée sur les données d'un échantillon. Plus l'intervalle est large, moins notre estimation est précise. Le niveau de confiance est souvent fixé à 95%, ce qui signifie qu'en répétant l'expérience un grand nombre de fois, 95% des intervalles de confiance obtenus contiendront le vrai paramètre inconnu. Les intervalles de confiance sont largement utilisés en statistique pour quantifier l'incertitude dans les estimations de paramètres et pour évaluer la précision des résultats de l'échantillon.

$$\text{Intervalle de confiance} = \text{estimation ponctuelle} \pm (\text{valeur critique} \times \text{erreur standard})$$

où la valeur critique dépend du niveau de confiance et de la distribution utilisée, et l'erreur standard est généralement calculée à partir de l'écart-type de l'échantillon et de la taille de l'échantillon.



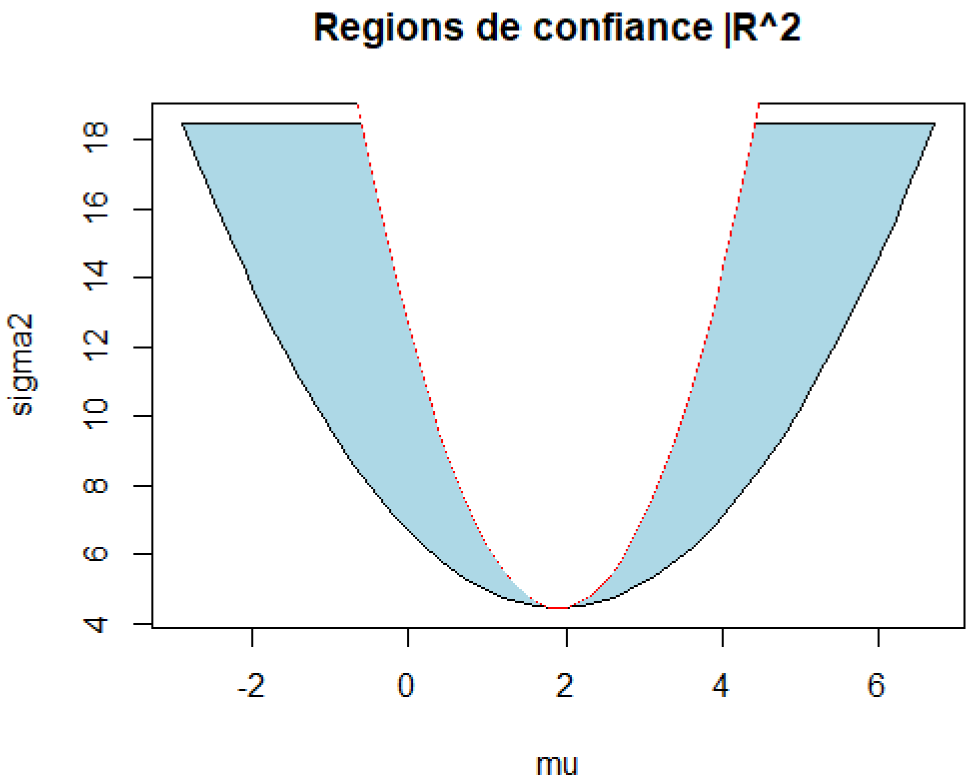
On observe que plus n augmente, plus l'intervalle se ressert autour d'alpha, l'alpha estimée va converger vers l'alpha théorique.

On appelle région de confiance de niveau $1-\alpha$ ($0<\alpha<1$) pour $g(\theta)$, une famille de parties de $g(\Theta)$ indicées par $\omega \in \Omega$, soit $\{S(\omega) : \omega \in \Omega\}$ tel que :

$$1)\forall\theta \in \Theta, (\omega : g(\theta) \in S(\omega)) \in \mathcal{F}$$

$$2)\forall\theta \in \Theta, P_\theta(\omega : g(\theta) \in S(\omega)) = 1-\alpha$$

Voici par exemple la région de confiance pour le vecteur (μ, σ^2) de la loi normale.

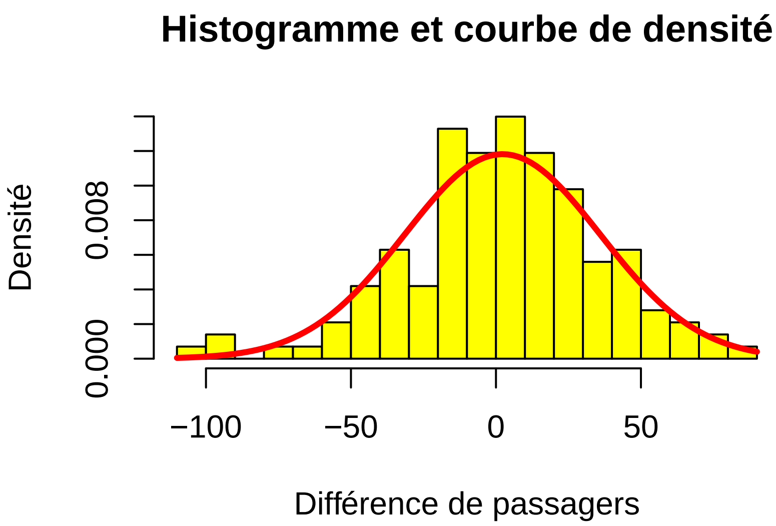


7. Etude globale

La base de données *airpassengers* est un jeu de données qui contient les données mensuelles de passagers aériens internationaux, de janvier 1949 à décembre 1960.
La base de données est constituée de deux colonnes :
Month: une variable de type “date”, qui correspond à la date de chaque observation dans le format “yyyy-mm-dd”. Passengers: une variable numérique, qui correspond au nombre de passagers aériens internationaux pour chaque mois.

7.1. Distribution

Les comportements humains, tels que les voyages, peuvent souvent être approximés par des lois normales. En effet, les tendances saisonnières ou les fluctuations économiques peuvent influencer les habitudes de voyage des gens, et il est possible que ces habitudes suivent une distribution normale.



En visualisant les données, on peut voir une tendance à la hausse, mais avec des fluctuations autour d’une moyenne. Cette forme de distribution peut ressembler à une distribution normale. En effet avec la courbe de densité d’une loi normale, on a une tendance assez similaire.
Cependant, pour nous assurez qu’on peut supposer que ces données suivent une loi normale nous allons appliquer le test de Shapiro-Wilk.

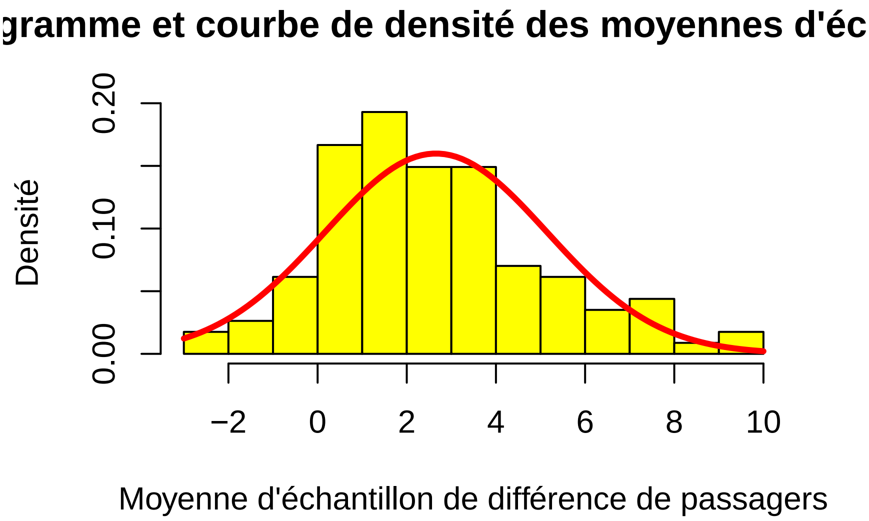
```
    Shapiro-Wilk normality test

data:  diff_passengers
W = 0.98576, p-value = 0.1469
```

On a une p-value > 0.05, ce qui suggère que les données ne s’écartent pas significativement d’une distribution normale. Donc on conclut que ces données suivent une loi normale.

7.2. Loi des grands nombres et théorème centrale limite

On peut appliquer la loi des grands nombres pour vérifier si les moyennes des échantillons de différences de passagers suivent une loi normale.



Le théorème central limite est déjà appliqué dans la construction de l'histogramme et de la courbe de densité normale dans le code précédent. En effet, la courbe de densité normale est tracée en utilisant les paramètres de la distribution normale des différences de passagers, qui sont estimés à partir des données. Cela montre que les différences de passagers suivent une distribution normale.

7.3. Estimation ponctuelles des paramètres

Estimation des paramètres de la population à partir de l'échantillon :

```
Estimation de la moyenne :  280.2986
```

```
Estimation de l'écart type :  119.9663
```

7.4. Test

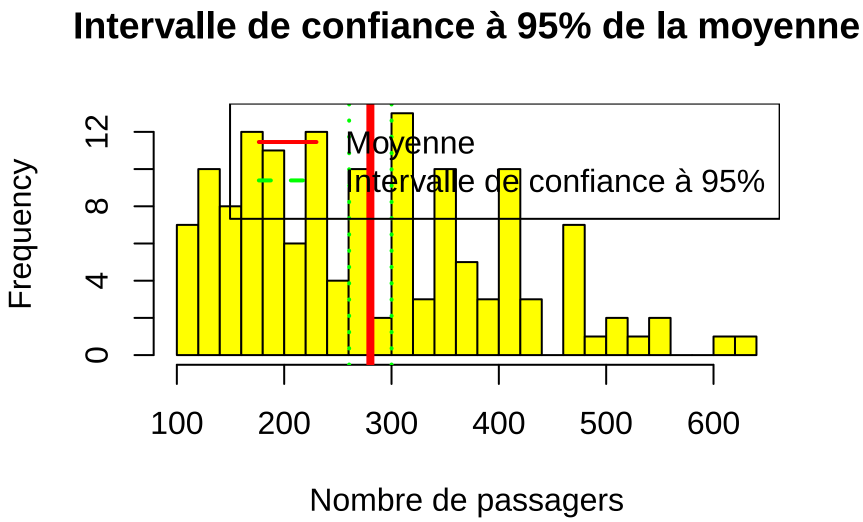
On va tester $H_0 : \mu = 0$ contre $H_1 : \mu \neq 0$

```
L'hypothèse nulle n'est pas rejetée avec un risque de première espèce de 0.05
```

```
La fonction puissance est estimée à 0.767
```

7.5. Intervalles de confiance

```
Intervalle de confiance à 95 % : [ 260.7 , 299.89 ]
```



Remerciements

Merci à M. Louani pour son enseignement et ses conseils.

A. Codes R

A.1. Tirage d'échantillons et diagrammes

```r
## Loi de Bernouilli -----

N = 1000  #nb de nombre au hasard a generer
print(N)
n = 20  #parametre de la loi B(n,p)
print(n)
p = 0.3  #parametre de la loi B(n,p)
print(p)

Bin = rbinom(N, n, p)  #genere des realisations aleatoires
print(Bin)
fa = table(Bin)  #calcul des frequences absolus
print(fa)
fr = prop.table(fa)  #calcul des frequences relatives
print(fr)
res = summary(Bin)
print(res)

par(mfrow = c(2, 2))
# graphique de la loi generee
plot(fr, main = "Variable B(n,p)", col = "red")
# graphique de la loi theorique
x = seq(min(Bin), max(Bin), by = 1)  #abscisses
yBin = dbinom(x, n, p)  #ordonnees
plot(x, yBin, main = "Graphe de la loi theorique B(n,p)", col = "blue", type = "h")
# superposer les deux graphes
plot(fr, main = "Graphes de la loi B(n,p)", col = "red")
lines(x, yBin, col = "blue", type = "h")
legend("topleft", c("graphe obs", "graphe th"), col = c("red", "blue"), lty = 1:1,
    cex = 0.3)
# graphe de la fct de repartition empirique
plot(ecdf(Bin), main = "Fonction de repartition empirique", col = "red")
y = pbinom(x, n, p)  #ordonnee pour la fct de rep
lines(x, y, col = "blue", type = "p")
legend("bottomright", c("graphe obs", "graphe th"), col = c("red", "blue"), lty = c(1,
    0), cex = 0.5, pch = c(19, 1))
```

```r
## Loi de Poisson -----
```

```r
N = 1000  #nb de nombre au hasard a generer
print(N)
lambda = 20  #parametre de la loi P(lambda)
print(lambda)

Poiss = rpois(N, lambda)  #genere des realisations aleatoires
print(Poiss)
fa = table(Poiss)  #calcul des frequences absolus
print(fa)
fr = prop.table(fa)  #calcul des frequences relatives
print(fr)
res = summary(Poiss)
print(res)

par(mfrow = c(2, 2))
# graphique de la loi generee
plot(fr, main = "Variable Poisson P(lambda)", col = "red")
# graphique de la loi theorique
x = seq(min(Poiss), max(Poiss), by = 1)  #abscisses
yPoiss = dpois(x, lambda)  #ordonnees
plot(x, yPoiss, main = "Graphe de la loi theorique P(lambda)", col = "blue", type = "h")
# superposer les deux graphes
plot(fr, main = "Graphes de la loi de Poisson P(lambda)", col = "red")
lines(x, yPoiss, col = "blue", type = "h")
legend("topleft", c("graphe obs", "graphe th"), col = c("red", "blue"), lty = 1:1,
    cex = 0.5)
# graphe de la fct de repartition empirique
plot(ecdf(Poiss), main = "Fonction de repartition empirique", col = "red")
y = ppois(x, lambda)  #ordonnee pour la fct de rep
lines(x, y, col = "blue", type = "p")
legend("bottomright", c("graphe obs", "graphe th"), col = c("red", "blue"), lty = c(1,
    0), cex = 0.5, pch = c(19, 1))
```

```r
## Loi Exponentielle -----

N = 10000
lambda = 2.5
Exp = rexp(N, lambda)
summary(Exp)
par(mfrow = c(1, 2))
hist(Exp, freq = FALSE, main = "Histogramme de E(lambda)", col = "yellow", nclass = 20)
# Courbe de la densite de la exponentielle
x = seq(0, lambda + 10, by = 0.1)
yExp = dexp(x, lambda)
lines(x, yExp, type = "l", col = "red", lwd = 3)
legend("topright", paste(c("N = ", "Mean = ", "Var = "), round(c(N, mean(Exp), var(Exp)),
    2), sep = ""), text.col = c("blue", "red", "green"), cex = 0.5)
# courbe de la fonction de r?partition empirique
plot(ecdf(Exp), main = "Fonction de repartition empirique", col = "blue", lwd = 3)
# courbe de la fonction de r?partition th?orique
FrtExp = pexp(x, lambda)
```

```
lines(x, FrtExp, main = "Fonction de repartition theorique", type = "l", col = "green",
    lwd = 3)
legend("bottomright", c("graphe obs", "graphe th"), col = c("blue", "green"), lty = c(1,
    1), cex = 0.5, pch = c(19, 19))
```

```
## Loi Normale -----

# generer des nbs au hasard issu de la loi N(mu,sigma)
N = 10000
mu = 2
sigma = 1.2
Nor = rnorm(N, mu, sigma)
resnorm = summary(Nor)
print(resnorm)
# Histogramme de la loi normale
par(mfrow = c(1, 2))
hist(Nor, freq = FALSE, main = "Histogramme de la loi Normale", col.main = "purple",
    nclass = 70, col = rainbow(10))
# Courbe de la densite de la loi Normale N(mu,sigma)
MNor = mean(Nor)  #moyenne de Nor
VNor = var(Nor)  #variance de Nor
x = seq(MNor - 4 * sd(Nor), MNor + 4 * sd(Nor), by = 0.1)  # sd (standart deviation) = ecart t
dNor = dnorm(x, MNor, sd(Nor))  #densite de la loi Normale pour tout les points x
lines(x, dNor, col = "red", type = "l", lwd = 3)
legend("topleft", paste(c("N = ", "Mean = ", "Var = "), round(c(N, mean(Nor), var(Nor)),
    2), sep = ""), text.col = c("blue", "red", "green"), cex = 0.5)
# Fonction de repartition empirique F_n(x) = 1/n (somme de 1 à n de
# l'indicatrice (X_i <= x))
m = length(x)
FN = rep(0, times = m)  #m fois que des zeros
for (i in 1:m) {
    for (j in 1:N) {
        if (Nor[j] <= x[i]) {
            FN[i] = FN[i] + 1
        }
    }
    FN[i] = FN[i]/N
}
plot(x, FN, main = "Fonction de répartition empirique", col.main = "green", col = "blue",
    type = "l", lwd = 3)
y = pnorm(x, MNor, sd(Nor))
lines(x, y, col = "red", type = "l", lwd = 3, lty = 2)
legend("topleft", c("F.Repart. emp. observée", "F.Repart. emp. théorique"), col = c("blue",
    "red"), lty = c(1, 2), cex = 0.45)
```

A.2. Loi des grands nombres

```r
# Loi Bernouilli
par(mfrow = c(2, 2))
N = 500
m = 10
p = 0.4
nb = 1:N
moy = rep(0, times = N)
Bin = rbinom(N, m, p)
moy = cumsum(Bin)/nb
esp = rep(m * p, times = N)
plot(nb, moy, type = "l", col = "blue", lwd = 3, main = "LGN pour B(m,p)")
lines(nb, esp, type = "l", col = "red", lwd = 3)
legend("bottomright", c("Moyenne emp.", "Moyenne th."), col = c("blue", "red"), lty = c(1,
    1), cex = 0.45)

# Loi de Poisson
N = 500
lambda = 4
nb = 1:N
moyp = rep(0, times = N)
Poi = rpois(N, lambda)
moyp = cumsum(Poi)/nb
espp = rep(lambda, times = N)
plot(nb, moyp, type = "l", col = "blue", lwd = 3, main = "LGN pour P(lambda)")
lines(nb, espp, type = "l", col = "red", lwd = 3)
legend("topright", c("Moyenne emp.", "Moyenne th."), col = c("blue", "red"), lty = c(1,
    1), cex = 0.45)

# Loi normale
N = 500
mu = 4
sd = 1.4
nb = 1:N
moyn = rep(0, times = N)
Nor = rnorm(N, mu, sd)
moyn = cumsum(Nor)/nb
espn = rep(mu, times = N)
plot(nb, moyn, type = "l", col = "blue", lwd = 3, main = "LGN pour N(mu,sd)")
lines(nb, espn, type = "l", col = "red", lwd = 3)
legend("topright", c("Moyenne emp.", "Moyenne th."), col = c("blue", "red"), lty = c(1,
    1), cex = 0.45)

# Loi exponentielle
N = 500
lambda = 4
nb = 1:N
moye = rep(0, times = N)
Exp = rpois(N, 1/lambda)
moye = cumsum(Exp)/nb
espe = rep(1/lambda, times = N)
plot(nb, moye, type = "l", col = "blue", lwd = 3, main = "LGN pour E(lambda)")
```

```r
lines(nb, espe, type = "l", col = "red", lwd = 3)
legend("topright", c("Moyenne emp.", "Moyenne th."), col = c("blue", "red"), lty = c(1,
    1), cex = 0.45)
```

A.3. Théorème centrale limite

```r
par(mfrow = c(2, 2))
# Pour la vitesse de convergence de la loi des GN Loi Normale -----
N = 1000  #Taille de lechantillon
n = 2000  #Nombre de Un
mu = 0
sd = 1
# construire un echantillon de Un
Un = rep(0, times = n)
for (i in 1:n) {
    Nor = rnorm(N, mu, sd)
    Un[i] = sqrt(N) * (mean(Nor) - mu)/sqrt(sd)
}
# histogramme avec courbe
hist(Un, freq = FALSE, main = "TCL pour la loi Normale", nclass = 16, col = "blue")
x = seq(-4, 4, by = 0.1)
y = dnorm(x, 0, 1)
lines(x, y, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n ="), c(N, n)), cex = 0.5)
legend(x = "topright", legend = paste(c("mu=", "sd="), c(mu, sd)), cex = 0.5)

## Loi Binomiale -----
N = 1000  #Taille de lechantillon
n = 2000  #Nombre de Un
m = 10
p = 0.4
# construire un echantillon de Un
Un = rep(0, times = n)
for (i in 1:n) {
    Bin = rbinom(N, m, p)
    Un[i] = (mean(Bin) - (m * p)) * (sqrt(N/(m * p * (1 - p))))
}
# histogramme avec courbe
hist(Un, freq = FALSE, main = "TCL pour la loi Binomiale", nclass = 16, col = "yellow")
x = seq(-4 * sd(Un), 4 * sd(Un), by = 0.1)
y = dnorm(x, 0, 1)
lines(x, y, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n ="), c(N, n)), cex = 0.5)

## Loi Poisson -----
N = 1000  #Taille de lechantillon
n = 2000  #Nombre de Un
lambda = 2
# construire un echantillon de Un
```

```
Un = rep(0, times = n)
for (i in 1:n) {
    Poi = rpois(N, lambda)
    Un[i] = sqrt(N) * (mean(Poi) - lambda)/sqrt(lambda)
}
# histogramme avec courbe
hist(Un, freq = FALSE, main = "TCL pour la loi de Poisson", nclass = 16, col = "green")
x = seq(-4 * sd(Un), 4 * sd(Un), by = 0.1)
y = dnorm(x, 0, 1)
lines(x, y, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n =", "lambda = "), c(N, n, lambda)),
    cex = 0.5)

## Loi Exponentielle -----
N = 1000  #Taille de lechantillon
n = 2000  #Nombre de Un
lambda = 2
# construire un echantillon de Un
Un = rep(0, times = n)
for (i in 1:n) {
    Exp = rexp(N, lambda)
    Un[i] = sqrt(N) * (mean(Exp) - 1/lambda)/sqrt(1/(lambda^2))
}
# histogramme avec courbe
hist(Un, freq = FALSE, main = "TCL pour la loi Exponentielle", nclass = 16, col = "purple")
x = seq(-4 * sd(Un), 4 * sd(Un), by = 0.1)
y = dnorm(x, 0, 1)
lines(x, y, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n =", "lambda = "), c(N, n, lambda)),
    cex = 0.5)
```

A.4. Estimation ponctuelle des paramètres

```
## Loi Normale ----
par(mfrow = c(1, 2))
N = 1000  #taille de lechantillon
n = 500  #nb dechantillon
mu = 1.5
sd = 0.3
# construire un echantillon
EX = rep(0, times = n)
VX = rep(0, times = n)
for (i in 1:n) {
    Nor = rnorm(N, mu, sd)
    EX[i] = mean(Nor)
    VX[i] = var(Nor)
}
DE = rep(mu, times = n)
DV = rep(sd^2, times = n)
```

```
nn = 1:n
plot(nn, EX, main = "Nuage de Xbarre", type = "p", col = "blue", lwd = 2)
lines(nn, DE, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n ="), c(N, n)), cex = 0.5)
legend(x = "topright", legend = paste(c("mu = ", "sd = "), c(mu, sd)), cex = 0.5)
plot(nn, VX, main = "Nuage de variance", type = "p", col = "green", lwd = 2)
lines(nn, DV, type = "l", col = "red", lwd = 2)
legend(x = "topleft", legend = paste(c("N =", "n ="), c(N, n)), cex = 0.5)
legend(x = "topright", legend = paste(c("mu = ", "sd = "), c(mu, sd)), cex = 0.5)
```

A.5. Tests

```
# 1. On teste H0 : 'mu = mu0' contre H1 : 'mu != mu0'
sigma = 0.5
mu0 = 1.5
N = 20
n = 100
alpha = 0.05
mu = seq(mu0 - 4 * sigma, mu0 + 4 * sigma, by = 0.1)
m = length(mu)
Beta = rep(0, time = m)
Beta1 = rep(0, time = m)
for (i in 1:m) {
    for (j in 1:n) {
        Nor = rnorm(N, mu[i], sigma)
        if (abs((mean(Nor) - mu0) * sqrt(N)/sigma) > qnorm((1 - alpha/2))) {
            Beta[i] = Beta[i] + 1
        }
    }
    Beta[i] = Beta[i]/n
}
for (i in 1:m) {
    for (j in 1:n) {
        Nor = rnorm(N, mu[i], sigma)
        if (abs((mean(Nor) - mu0) * sqrt(N)/sd(Nor)) > qt((1 - alpha/2), N - 1)) {
            Beta1[i] = Beta1[i] + 1
        }
    }
    Beta1[i] = Beta1[i]/n
}
Calpha = rep(alpha, times = m)
plot(mu, Beta, main = "Test bilateral sur la moyenne avec la variance connue et inconnue",
    col = "red", type = "l", lwd = 2)
lines(mu, Beta1, col = "green", type = "l", lwd = 2)
lines(mu, Calpha, col = "blue", type = "l", lwd = 2)
legend(x = "right", legend = c("Fonction puissance (sigma connue)", "Fonction puissance (sigma
    "Seuil de significativité"), col = c("red", "green", "blue"), lty = c(1, 1),
    cex = 0.6, lwd = 2)
```

```r
# 2. On teste H0 : 'mu <= mu0' contre H1 : 'mu > mu0'
sigma = 0.5
mu0 = 1.5
N = 20
n = 100
alpha = 0.05
mu = seq(mu0 - 4 * sigma, mu0 + 4 * sigma, by = 0.1)
m = length(mu)
Beta = rep(0, time = m)
Beta1 = rep(0, time = m)
for (i in 1:m) {
    for (j in 1:n) {
        Nor = rnorm(N, mu[i], sigma)
        if ((mean(Nor) - mu0) * sqrt(N)/sigma > qnorm((1 - alpha))) {
            Beta[i] = Beta[i] + 1
        }
    }
    Beta[i] = Beta[i]/n
}
for (i in 1:m) {
    for (j in 1:n) {
        Nor = rnorm(N, mu[i], sigma)
        if ((mean(Nor) - mu0) * sqrt(N)/sd(Nor) > qt((1 - alpha), N - 1)) {
            Beta1[i] = Beta1[i] + 1
        }
    }
    Beta1[i] = Beta1[i]/n
}
Calpha = rep(alpha, times = m)
plot(mu, Beta, main = "Test unilateral sur la moyenne avec la variance connue et inconnue",
    col = "red", type = "l", lwd = 2)
lines(mu, Beta1, col = "green", type = "l", lwd = 2)
lines(mu, Calpha, col = "blue", type = "l", lwd = 2)
legend(x = "right", legend = c("Fonction puissance (sigma connue)", "Fonction puissance (sigma
    "Seuil de significativité"), col = c("red", "green", "blue"), lty = c(1, 1),
    cex = 0.6, lwd = 2)

# 3. Cas général
lambda0 = 1.5
N = 1000
n = 100
alpha = 0.05
mu = seq(0.05, 4/lambda0, by = 0.05)
m = length(mu)
Beta = rep(0, time = m)
for (i in 1:m) {
    for (j in 1:n) {
        Exp = rexp(N, mu[i])
        if (abs((mean(Exp) - 1/lambda0) * sqrt(N)/sd(Exp)) > qnorm((1 - alpha/2))) {
            Beta[i] = Beta[i] + 1
```

```
        }
    }
    Beta[i] = Beta[i]/n
}
Calpha = rep(alpha, times = m)
plot(mu, Beta, main = "Test bilateral sur la moyenne avec la variance inconnue",
    col = "red", type = "l", lwd = 2)
lines(mu, Calpha, col = "blue", type = "l", lwd = 2)
legend(x = "right", legend = c("Fonction puissance", "Seuil de significativité"),
    col = c("red", "blue"), lty = c(1, 1), cex = 0.6, lwd = 2)

lambda0 = 1.5
N = 1000
n = 100
alpha = 0.05
mu = seq(0.05, 4/lambda0, by = 0.05)
m = length(mu)
Beta = rep(0, time = m)
for (i in 1:m) {
    for (j in 1:n) {
        Exp = rexp(N, mu[i])
        if (((mean(Exp) - 1/lambda0) * sqrt(N)/sd(Exp)) > qt((1 - alpha), N - 1)) {
            Beta[i] = Beta[i] + 1
        }
    }
    Beta[i] = Beta[i]/n
}
Calpha = rep(alpha, times = m)
plot(mu, Beta, main = "Test unilateral sur la moyenne avec la variance inconnue",
    col = "red", type = "l", lwd = 2)
lines(mu, Calpha, col = "blue", type = "l", lwd = 2)
legend(x = "right", legend = c("Fonction puissance", "Seuil de significativité"),
    col = c("red", "blue"), lty = c(1, 1), cex = 0.6, lwd = 2)
```

A.6. Intervalle de confiance

```
par(mfrow = c(1, 2))
alpha = 0.05
n = 50  #nb dechantillons
N = 20  #taille
# loi normale de variance connue
mu = 1.5
sigma = 0.5
k = 100
Est = rep(0, times = k)
for (l in 1:k) {
    alphaest = 0
    for (i in 1:n) {
        Nor = rnorm(N, mu, sigma)
```

```
        if ((abs(mean(Nor) - mu)) > qnorm(1 - alpha/2) * sigma/sqrt(N)) {
            # quantile loi normale
            alphaest = alphaest + 1
        }
        Est[l] = alphaest/n
    }
}
x = 1:k
plot(x, Est, main = "Convergence de l'estimateur alpha (sigma connu)", col = "red",
    type = "l")
xalpha = rep(alpha, times = k)
lines(x, xalpha, col = "blue", type = "l", lwd = 2)
legend(x = "topleft", legend = paste(c("n =", "alpha ="), c(n, alpha)), cex = 0.5)
for (l in 1:k) {
    alphaest = 0
    for (i in 1:n) {
        Nor = rnorm(N, mu, sigma)
        if ((abs(mean(Nor) - mu)) > qt(1 - alpha/2, N - 1) * sd(Nor)/sqrt(N)) {
            # quantile loi poisson : variance pas connue
            alphaest = alphaest + 1
        }
        Est[l] = alphaest/n
    }
}
x = 1:k
plot(x, Est, main = "Convergence de l'estimateur alpha (sigma non connu)", col = "green",
    type = "l")
xalpha = rep(alpha, times = k)
lines(x, xalpha, col = "blue", type = "l", lwd = 2)
legend(x = "topleft", legend = paste(c("n =", "alpha ="), c(n, alpha)), cex = 0.5)
alphaest = alphaest/n  # On obtient une valeur proche de 0.05 (seuil fixé)

alpha = 0.05
N = 20
mu1 = 2
sd = 1.2
Nor = rnorm(N, mu1, sigma)
B = mean(Nor)
mu = seq(B - 4 * sd, B + 4 * sd, by = 0.1)
NN = length(mu)
sigma1 = rep(0, NN)
sigma2 = rep(0, NN)
t1 = qchisq(alpha/2, N - 1)
t2 = qchisq(1 - alpha/2, N - 1)
mn = rep(mean(Nor), times = N)
for (i in 1:NN) {
    a = mu[i]
    sigma1[i] = sum((Nor - mn)^2) + N * (mean(Nor) - a)^2/t1
    sigma2[i] = sum((Nor - mn)^2) + N * (mean(Nor) - a)^2/t2
}
plot(mu, sigma2, main = "Regions de confiance |R^2", type = "l", col = "blue")
```

```r
polygon(mu, sigma2, border = TRUE, col = "lightblue")
lines(mu, sigma1, type = "l", col = "red")
polygon(mu, sigma1, border = FALSE, col = "white")
```