

Projet_PMS_Zouggari_Zivkovic

2024-03-23

I - Analyse statistique d'un échantillon de loi de Poisson

1.

- a) Soit X une variable aléatoire suivant une loi de Poisson. $\forall x \in \mathbb{N}, P(X = x) = e^{-\lambda} \frac{\lambda^x}{x!}$ Donc, $\ln[x!P(X = x)] = e^{-\lambda} \lambda^x = x \ln(\lambda) - \lambda$.
- b) D'après la question précédente, il existe une relation affine entre $\ln(x!P(X = x))$ et λ . Pour estimer graphiquement λ , on peut réaliser un graphe de probabilités avec, x en abscisses et $\ln(x!P(X=x))$ en ordonnées. Et voir si la courbe correspondante correspond à une droite. Si oui, alors la loi de Poisson est un modèle approprié.

```
lambda <- 3

n<-100

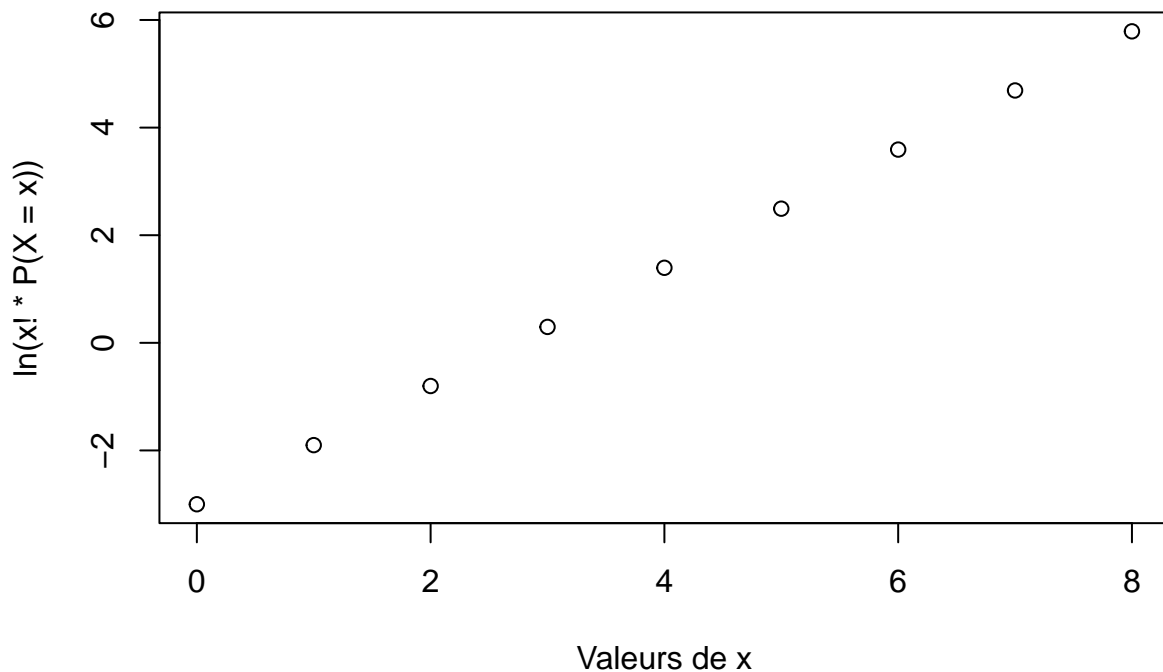
realisations <- rpois(n,lambda)

frequences <- table(realisations)
abscisses_uniques <- unique(realisations)

ln_x_fact_poisson_prob <- sapply(abscisses_uniques, function(x) {
  return(log(factorial(x)*dpois(x,lambda)))
})

ordonnees_uniques <- unique(ln_x_fact_poisson_prob)
plot(abscisses_uniques,ordonnees_uniques, xlim = range(abscisses_uniques), ylim = range(ordonnees_uniques),
     xlab = "Valeurs de x", ylab = "ln(x! * P(X = x))", main = "Graphique de dispersion de ln(x! * P(X = x))")
```

Graphique de dispersion de $\ln(x! * P(X = x))$



Ici un graphe de probabilités pour la loi de Poisson, avec des réalisations suivant la loi de Poisson (normal que ça aie une allure de droite !)

c)

```
lambda <- 5

n_values<-c(10,50,100,1000000)

for (n in n_values){
  realisations_poisson <- rpois(n,lambda)
  realisations_geo <- rgeom(n,1/lambda)

  abscisses_poisson_uniques <- unique(realisations_poisson)
  abscisses_geo_uniques <- unique(realisations_geo)

  ln_x_fact_poisson_prob <- sapply(abscisses_poisson_uniques, function(x) {
    return(log(factorial(x)*dpois(x,lambda)))
  })
  ln_x_fact_geo_prob <- sapply(abscisses_geo_uniques, function(x) {
    return(log(factorial(x)*dgeom(x,1/lambda)))
  })

  ordonnees_poisson_uniques <- unique(ln_x_fact_poisson_prob)
  ordonnees_geo_uniques <- unique(ln_x_fact_geo_prob)

  courbe <-smooth.spline(abscisses_geo_uniques,ordonnees_geo_uniques)
```

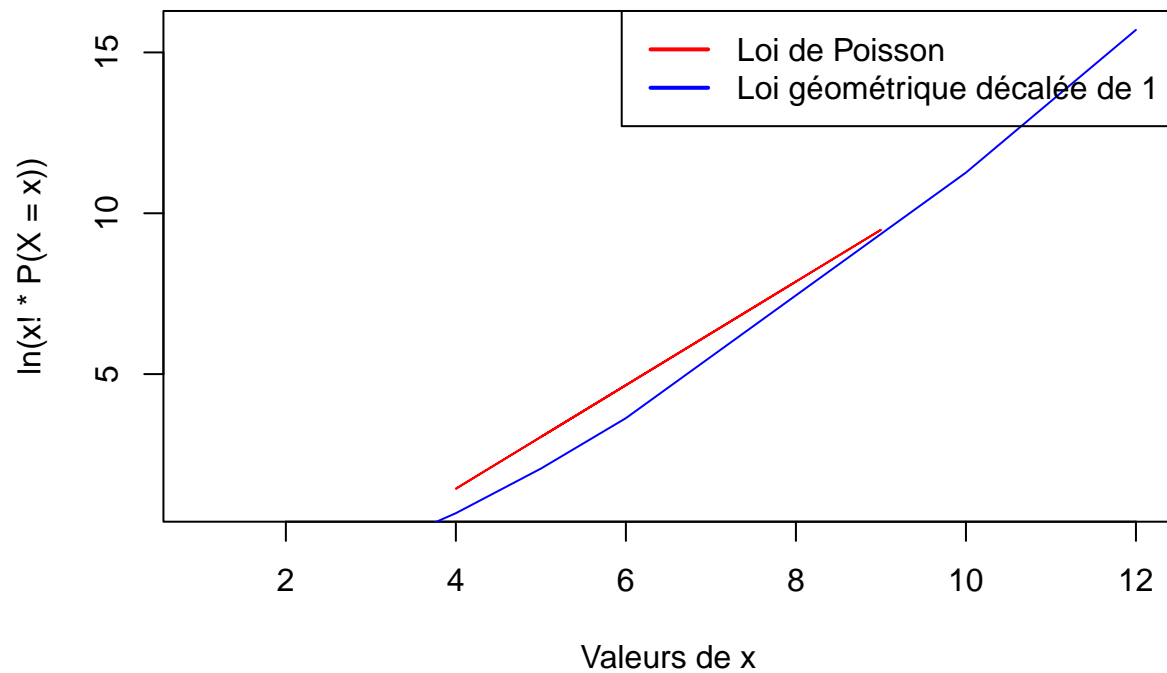
```

plot(NULL, xlim = c(min(abscisses_poisson_uniques,abscisses_geo_uniques),max(abscisses_poisson_uniques,
  xlab = "Valeurs de x", ylab = "ln(x! * P(X = x))", main = ("Graphe de probabilités pour une loi de

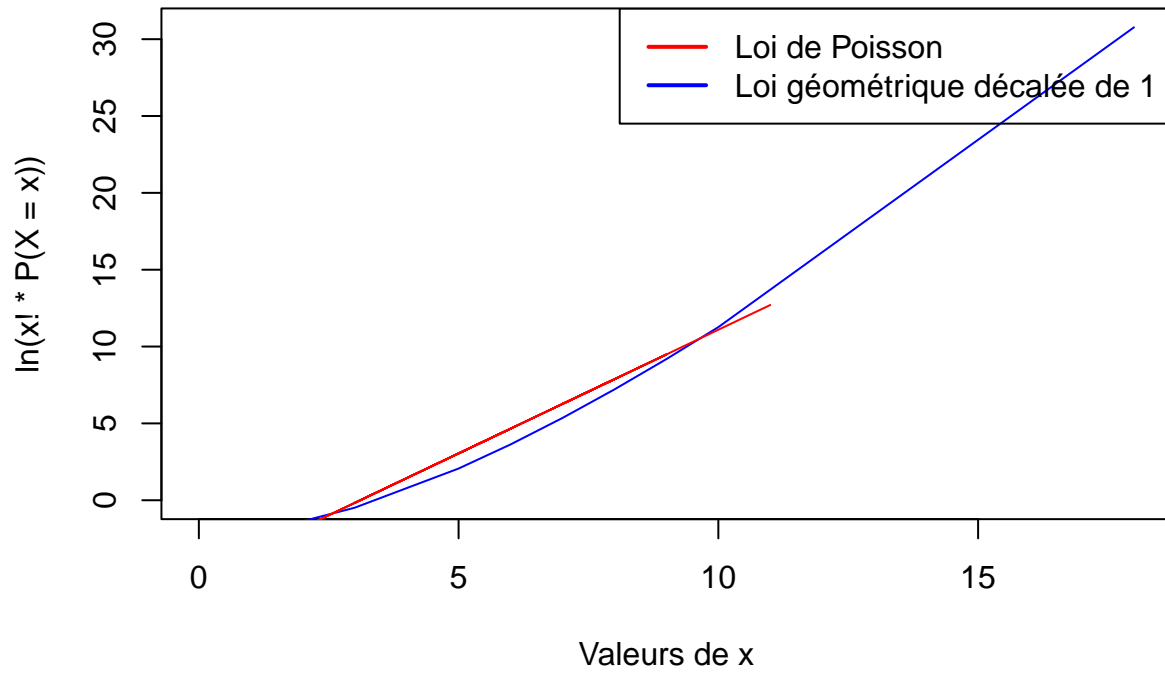
lines(courbe, col='blue')
lines(abscisses_poisson_uniques, ordonnees_poisson_uniques,col='red')
legend("topright", legend = c("Loi de Poisson", "Loi géométrique décalée de 1"), col = c("red", "blue")
}

```

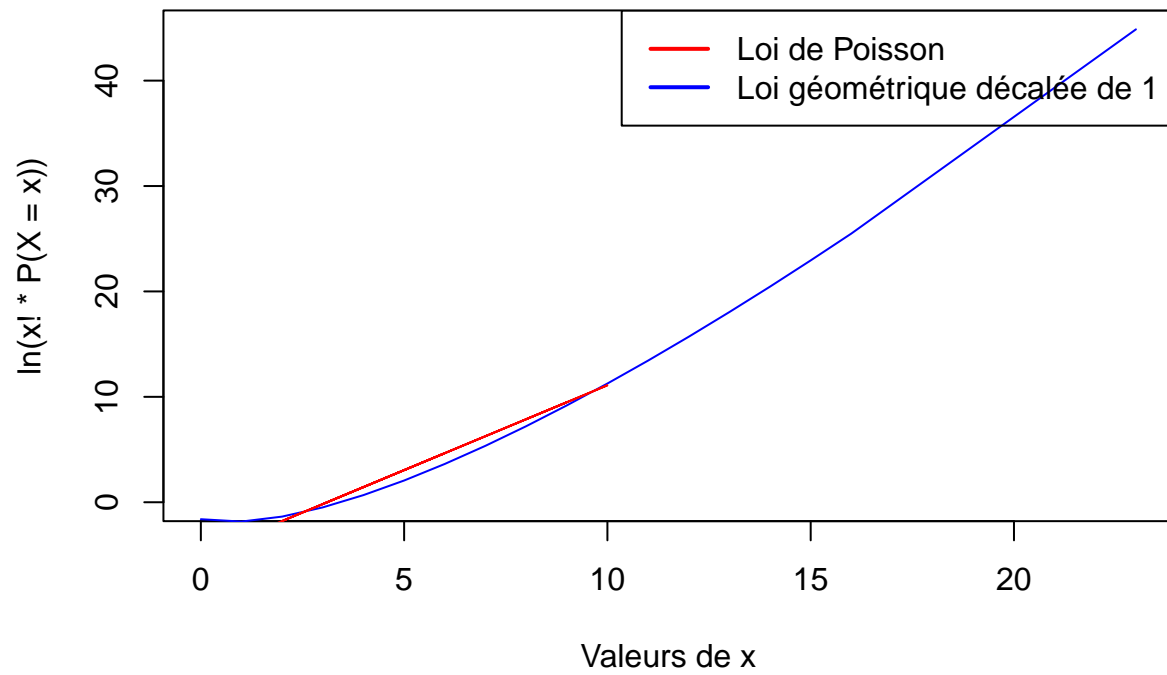
Graphe de probabilités pour une loi de Poisson



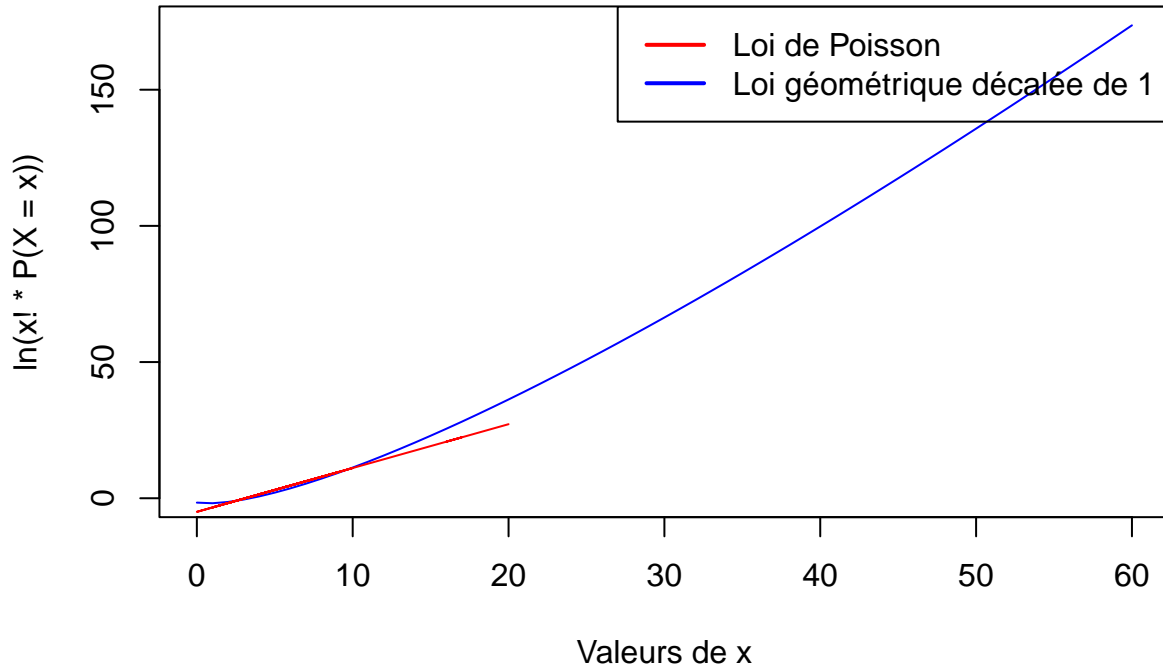
Graphe de probabilités pour une loi de Poisson



Graphe de probabilités pour une loi de Poisson



Graphe de probabilités pour une loi de Poisson



Le graphe de probabilités pour la loi de Poisson, avec des réalisations suivant une loi géométrique décalée de 1, ne correspond pas à une droite. Contrairement aux réalisations suivant une loi de Poisson (logique !). Ainsi, on aboutit au résultat attendu qui est que l'échantillon suivant une loi géométrique ne suit pas une loi de poisson. A noter qu'il est plus simple de conclure quand le nombre de réalisations est grand.

2.

a) Commençons par calculer la fonction de vraisemblance pour des réalisations x_1, \dots, x_n de variables aléatoires X_1, \dots, X_n suivant une loi de Poisson. $L(\lambda; x_1, \dots, x_n) = \prod_{i=1}^n P(X_i = x_i) = \prod_{i=1}^n e^{-\lambda} \frac{\lambda^{x_i}}{x_i!}$

$$L(\lambda; x_1, \dots, x_n) = \frac{e^{-n\lambda} \lambda^{\sum_{i=1}^n x_i}}{x_1! \dots x_n!} \text{ Donc, } \ln(L(\lambda; x_1, \dots, x_n)) = -n\lambda + \ln(\lambda) \left(\sum_{i=1}^n x_i \right) - \sum_{i=1}^n \ln(x_i!) \text{ Donc,}$$

$$\frac{\partial}{\partial \lambda} \ln(L(\lambda; x_1, \dots, x_n)) = -n + \frac{\sum_{i=1}^n x_i}{\lambda}, \text{ Et } \frac{\partial^2}{\partial^2 \lambda} \ln(L(\lambda; x_1, \dots, x_n)) = -\frac{\sum_{i=1}^n x_i}{\lambda^2} I_n(\lambda) = -E\left(\frac{\partial^2}{\partial^2 \lambda} \ln(L(\lambda; x_1, \dots, x_n))\right) =$$

$$\frac{E\left(\sum_{i=1}^n X_i\right)}{\lambda^2} = \frac{\sum_{i=1}^n E(X_i)}{\lambda^2} I_n(\lambda) = \frac{\sum_{i=1}^n \lambda}{\lambda^2} = \frac{n\lambda}{\lambda^2} = \frac{n}{\lambda}$$

b) Tous les X_i sont indépendantes et sont de même loi de Poisson de paramètre λ Calculons le biais de $\hat{\lambda}_n$
 biais($\hat{\lambda}_n$) = $E(T_n) - \lambda$ Avec $E(T_n) = E(\bar{X}_n) = E\left(\frac{\sum_{i=1}^n X_i}{n}\right) = \frac{\sum_{i=1}^n E(X_i)}{n} = \lambda$

Donc, biais($\hat{\lambda}_n$) = $E(T_n) - \lambda = 0$

Donc $\hat{\lambda}_n$ est un estimateur sans biais de λ .

$I_n(\lambda) = \frac{1}{\text{var}(X)}$ Or, on sait que la borne inférieure de la variance (borne de Cramer-Rao) pour un estimateur sans biais est : $\frac{1}{\text{var}(X)}$. Donc, cet estimateur $\hat{\lambda}_n$ est bien de variance minimale pour λ .

Donc, $\hat{\lambda}_n$ est l'estimateur sans biais de variance minimale de λ .

c) On cherche un intervalle de confiance bilatéral de la forme $[\bar{X}_n - \epsilon, \bar{X}_n + \epsilon]$ On cherche $P(|\hat{\lambda}_n - \lambda| > \epsilon) = \alpha$
 Donc, on cherche $P(|\bar{X}_n - \lambda| > \epsilon) = \alpha$

On applique le théorème central-limite. On a $\sqrt{n} \frac{\bar{X}_n - E(X)}{\sigma(X)}$ converge en loi vers une loi normale de paramètres $N(0, 1)$.

Avec $E(X) = \lambda$ Comme $V(X) = E(X)$ (X suit une loi de Poisson), on peut estimer $\sigma(X)$ par $\sqrt{\bar{X}_n}$.

Donc, $\sqrt{n} \frac{\bar{X}_n - \lambda}{\sqrt{\bar{X}_n}}$ converge en loi vers une loi normale de paramètres $N(0, 1)$. Donc, $P(|\sqrt{n} \frac{\bar{X}_n - \lambda}{\sqrt{\bar{X}_n}}| < u_\alpha) = P(|\bar{X}_n - \lambda| < \frac{u_\alpha \sqrt{\bar{X}_n}}{\sqrt{n}}) = P(|\bar{X}_n - \lambda| < \epsilon)$ avec $\epsilon = \frac{u_\alpha \sqrt{\bar{X}_n}}{\sqrt{n}}$

Donc, l'intervalle de confiance bilatéral asymptotique de seuil α pour λ s'écrit : $[\bar{X}_n - \frac{u_\alpha \sqrt{\bar{X}_n}}{\sqrt{n}}, \bar{X}_n + \frac{u_\alpha \sqrt{\bar{X}_n}}{\sqrt{n}}]$.

II - Statistiques sur le football

```
library(worldfootballR)
n <- 382
Saison22 <- fb_match_results(country = c("FRA"), gender = "M", season_end_year = 2022)
saison22_goals <- Saison22$HomeGoals + Saison22$AwayGoals
sum(saison22_goals)
```

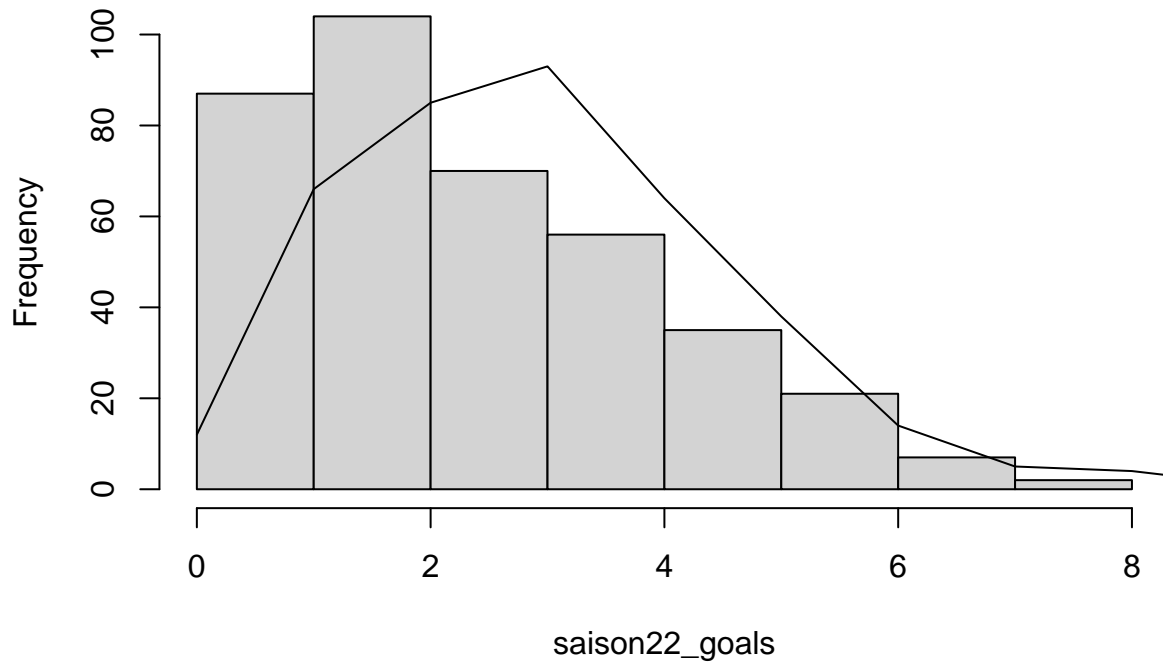
```
## [1] 1071
```

```
summary(saison22_goals)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   2.000   2.500   2.804   4.000   8.000
```

```
hist(saison22_goals)
poisson <- rpois(n, mean(saison22_goals))
tab <- table(poisson)
num_buts <- names(tab)
occ <- as.numeric(tab)
lines(num_buts, occ)
```

Histogram of saison22_goals



On peut admettre que la loi de Poisson est un modèle plausible pour ces données car l'histogramme représentant ces données a une forme ressemblante à celle d'une courbe représentative de données suivant une loi de Poisson.

c) $\lambda=2.804$ car on suppose que les données suivent une loi de Poisson x , et que l'estimateur de λ est sa moyenne expérimentale.

d)

```
library(worldfootballR)
n <- 382
par(mfrow = c(1,2))
Saison22 <- fb_match_results(country = c("FRA"), gender = "M", season_end_year = 2022)
saison22_goals <- Saison22$HomeGoals + Saison22$AwayGoals
sum(saison22_goals)
```

```
## [1] 1071
```

```
summary(saison22_goals)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000   2.000   2.500   2.804   4.000   8.000
```

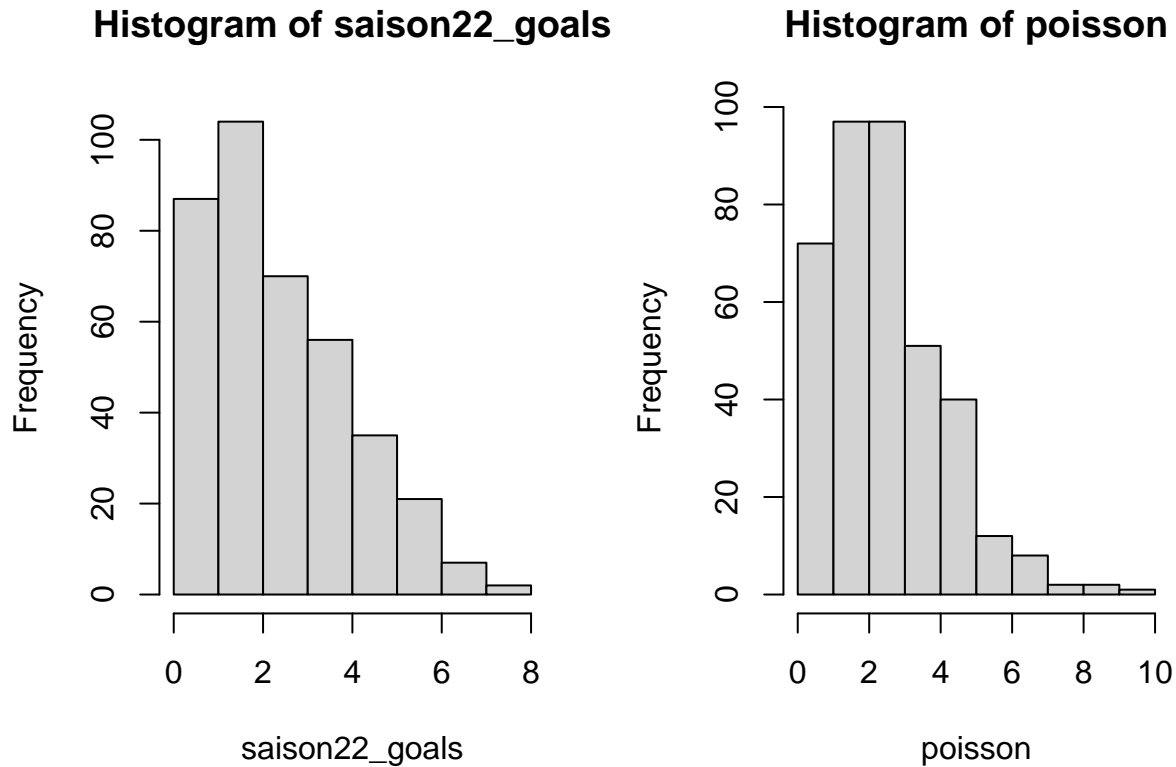
```
cat("variance : ",sd(saison22_goals))
```

```
## variance :  1.702457
```

```
hist(saison22_goals)
poisson <- rpois(n, mean(saison22_goals))
tab <- table(poisson)
```



```
num_buts <- names(tab)
occ <- as.numeric(tab)
hist(poisson)
```



Les fréquences relatives observées ressemblent aux fréquences relatives prédites à l'aide de la loi de Poisson. On en conclut que la loi de Poisson est un modèle adapté à la modélisation du nombres de buts marqués

e)

```
somme = sum(saison22_goals)

q <- qnorm(0.975)
n <- 382
cat("Intervalle de confiance à 5%, [", mean(saison22_goals) - (sqrt(mean(saison22_goals))*q/sqrt(n)),

## Intervalle de confiance à 5%, [ 2.635754 , 2.971576 ]
cat("Intervalle de confiance à 5%, [", n*( mean(saison22_goals) - (sqrt(mean(saison22_goals))*q/sqrt(n)),

## Intervalle de confiance à 5%, [ 1006.858 , 1135.142 ]
2 -

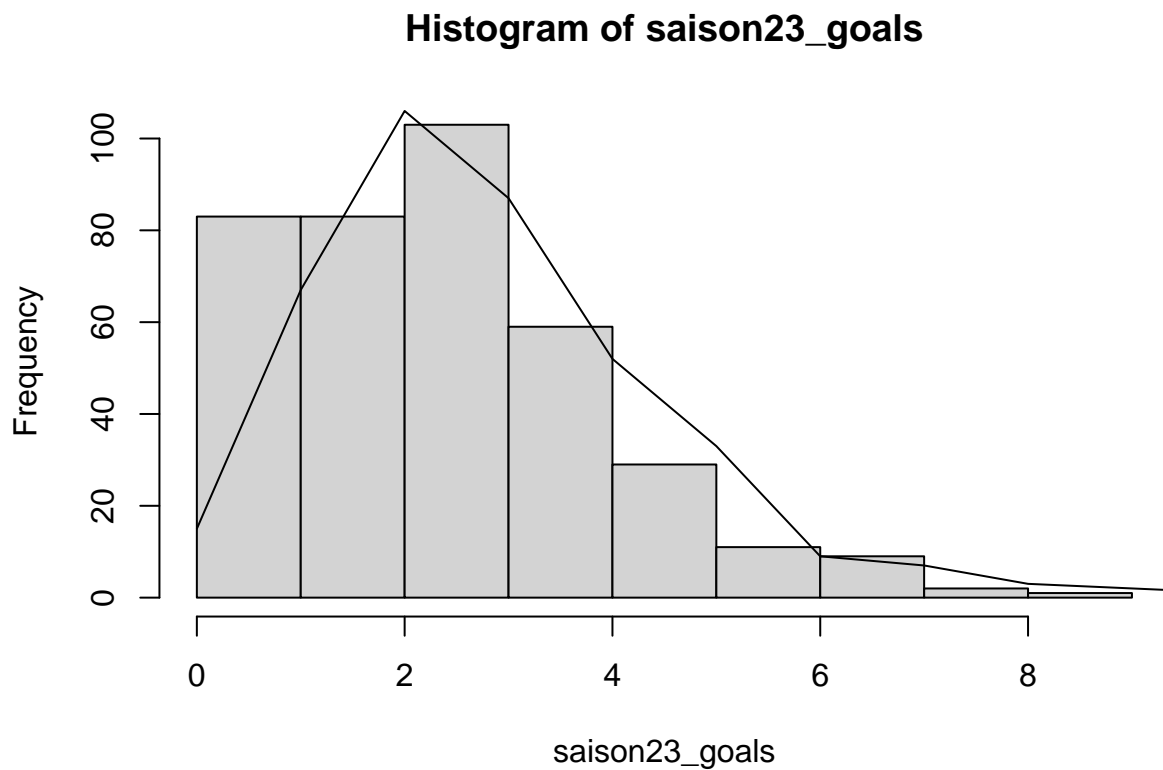
n <- 380
Saison23 <- fb_match_results(country = c("FRA"), gender = "M", season_end_year = 2023)
saison23_goals <- Saison23$HomeGoals + Saison23$AwayGoals
sum(saison23_goals)

## [1] 1067
```

```
summary(saison23_goals)
```

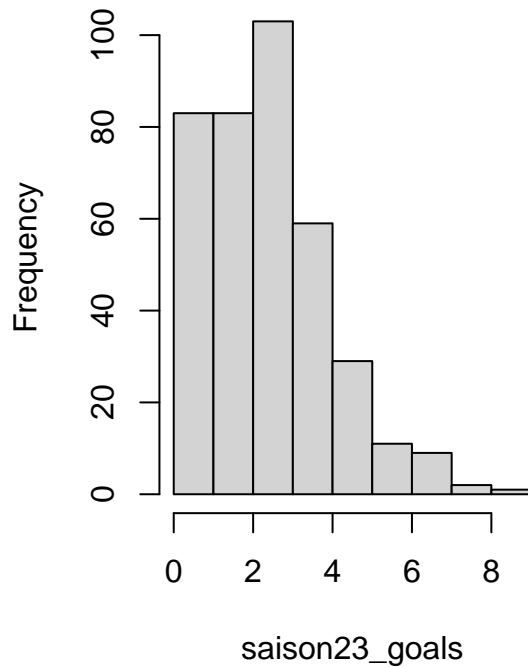
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   2.000   3.000   2.808   4.000   9.000
```

```
hist(saison23_goals)
poisson <- rpois(n, mean(saison23_goals))
tab <- table(poisson)
num_buts <- names(tab)
occ <- as.numeric(tab)
lines(num_buts, occ)
```

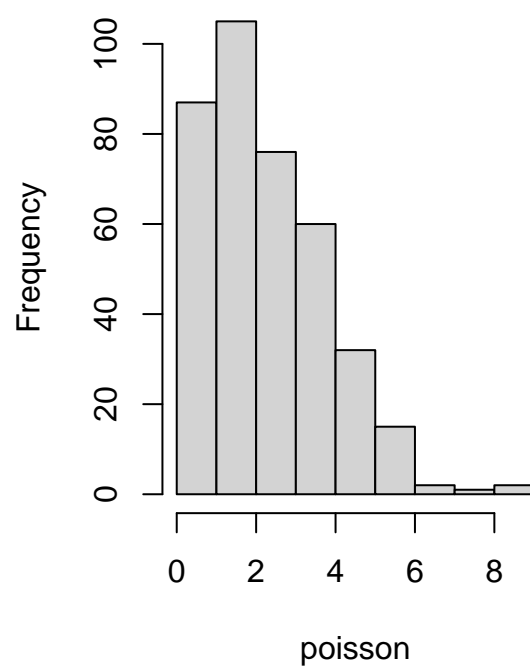


```
par(mfrow = c(1,2))
hist(saison23_goals)
poisson <- rpois(n, mean(saison23_goals))
tab <- table(poisson)
num_buts <- names(tab)
occ <- as.numeric(tab)
hist(poisson)
```

Histogram of saison23_goals



Histogram of poisson



```

somme = sum(saison23_goals)
cat("Intervalle de confiance Ã 5%, [", mean(saison23_goals) - (sqrt(mean(saison23_goals))*q/sqrt(n)),

## Intervalle de confiance Ã 5%, [ 2.639415 , 2.976374 ]
cat("Intervalle de confiance Ã 5%, [", n*( mean(saison23_goals) - (sqrt(mean(saison23_goals))*q/sqrt(n)),

## Intervalle de confiance Ã 5%, [ 1002.978 , 1131.022 ]

```

On retrouve quasiment la même chose pour la saison 2022-2023 donc on aurait pu utiliser les resultats de la saison précédente pour prévoir le nombre de but de la saison.

3 -

```

n <- 380
Saison03 <- fb_match_results(country = c("FRA"), gender = "M", season_end_year = 2003)
saison03_goals <- Saison03$HomeGoals + Saison03$AwayGoals
sum(saison03_goals)

```

```
## [1] 837
```

```
summary(saison03_goals)
```

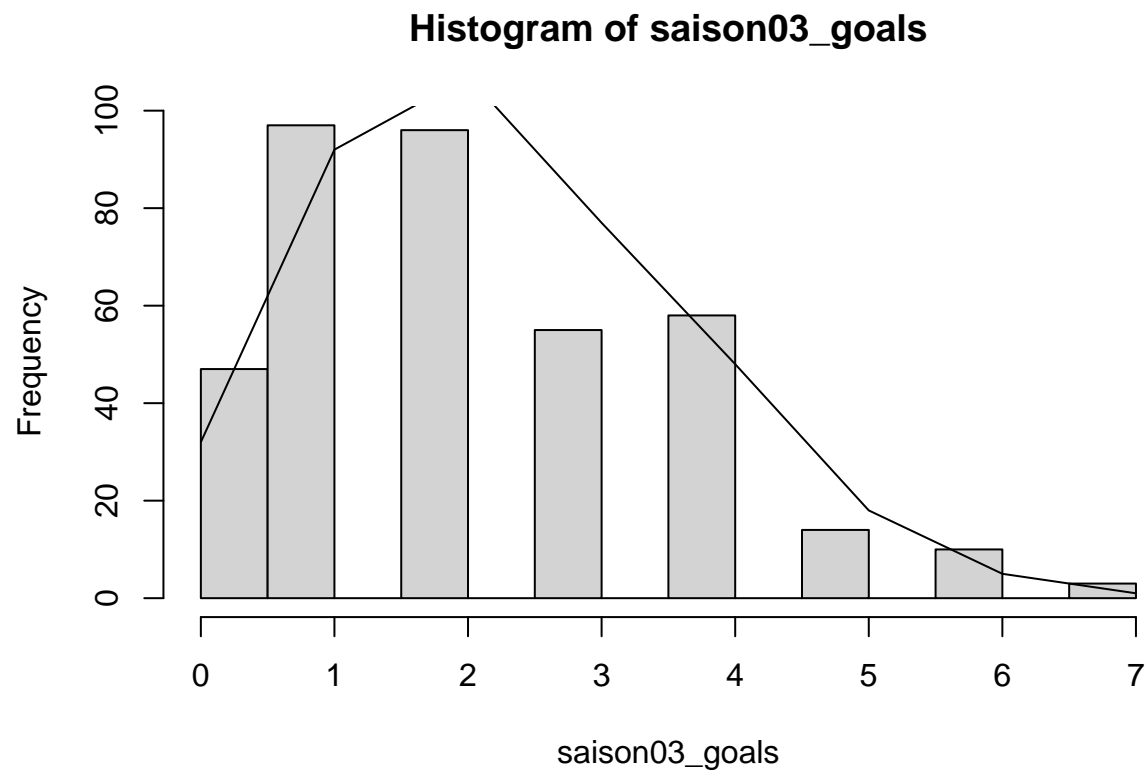
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.000   1.000   2.000   2.203   3.000   7.000
```

```

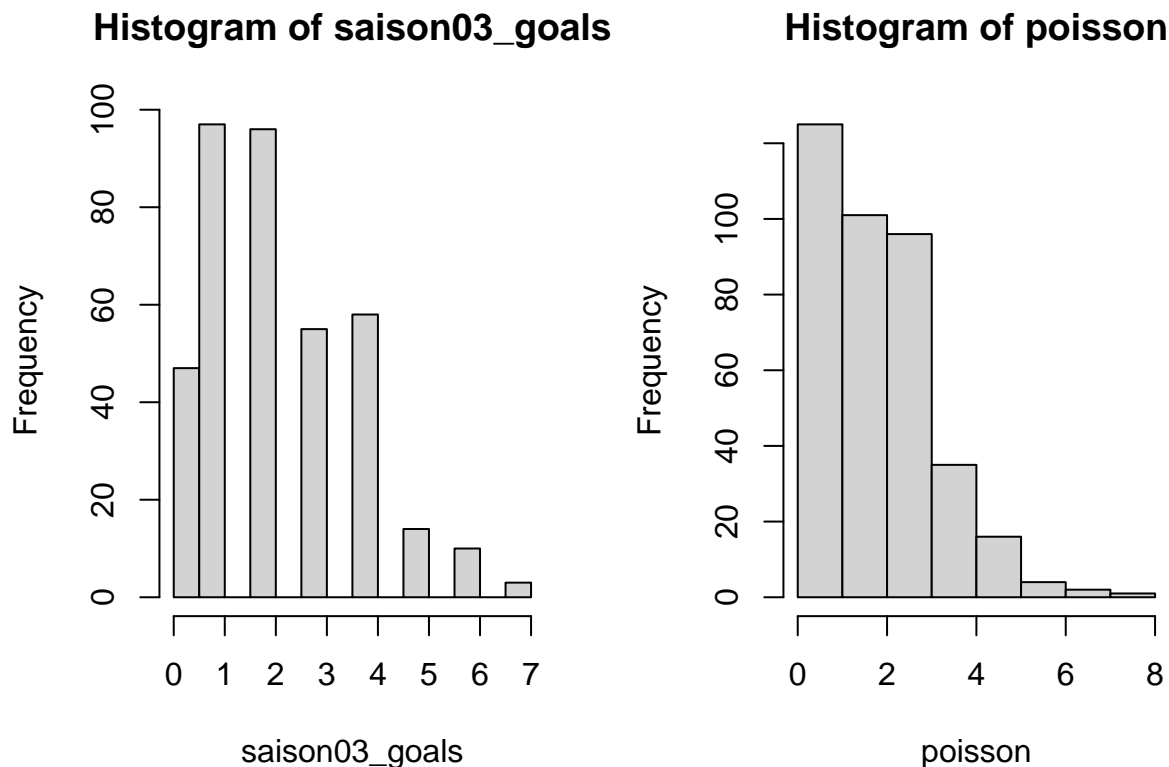
hist(saison03_goals)
poisson <- rpois(n, mean(saison03_goals))
tab <- table(poisson)
num_buts <- names(tab)

```

```
occ <- as.numeric(tab)
lines(num_buts, occ)
```



```
par(mfrow = c(1,2))
hist(saison03_goals)
poisson <- rpois(n, mean(saison03_goals))
tab <- table(poisson)
num_buts <- names(tab)
occ <- as.numeric(tab)
hist(poisson)
```



```
somme = sum(saison03_goals)
cat("Intervalle de confiance Ã 5%, [", mean(saison03_goals) - (sqrt(mean(saison03_goals))*q/sqrt(n)),

## Intervalle de confiance Ã 5%, [ 2.053412 , 2.351852 ]
cat("Intervalle de confiance Ã 5%, [", n*( mean(saison03_goals) - (sqrt(mean(saison03_goals))*q/sqrt(n)),

## Intervalle de confiance Ã 5%, [ 780.2964 , 893.7036 ]
```

Non, le football il a changé. Comme le dit si bien Mbappé depuis la saison 2002-2003, il y a eu de nombreux changements dans le foot donc les résultats de cette saison n'ont rien à voir avec les résultats des années 2000. On observe que le nombre de buts moyen et total sont bien différents pendant la saison 2002-2003 que pendant la saison 2022-2023.

4)a) c'est une loi binomiale de paramètre $(90, p)$ car successions iid de loi de Bernoulli (p) . $E = 90 \cdot p$, la moyenne expérimentale vaut 2.804 en 2022, donc une estimation de p est $2.804/90$, car la moyenne expérimentale est une estimation de l'espérance. Donc le nombre de minutes à attendre pour voir le 1er but marqué est $90/2.804$. Attendre 32 minutes pour avoir un but.

4)b)

```
matchFra2022 <- fb_match_urls(country = "FRA", gender = "M", season_end_year = 2022, tier = "1st")
n=0
for (i in (1:50)){
  vecteur <- fb_match_summary(match_url = matchFra2022[i])
  if(length(vecteur$Event_Time[vecteur$Event_Type=="Goal"])==0){
    n=n+1
  }
}
```

```

y<-numeric(50-n)
j=1
for (i in (1:50)){
  vecteur <- fb_match_summary(match_url = matchFra2022[i])
  if(length(vecteur$Event_Time[vecteur$Event_Type=="Goal"])>0){
    y[j]<-vecteur$Event_Time[vecteur$Event_Type=="Goal"][1]
    j=j+1
  }
}
y

```

```

## [1] 14 43 10 14 82 57 52 23 34 1 3 19 53 84 12 5 1 34 23 51 38 21 19 7 21
## [26] 10 58 35 7 23 40 24 57 24 58 46 16 7 20 37 32 26 39 50 75 8 26

```

```
summary(y)
```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00  14.00   24.00   30.62  44.50   84.00

```

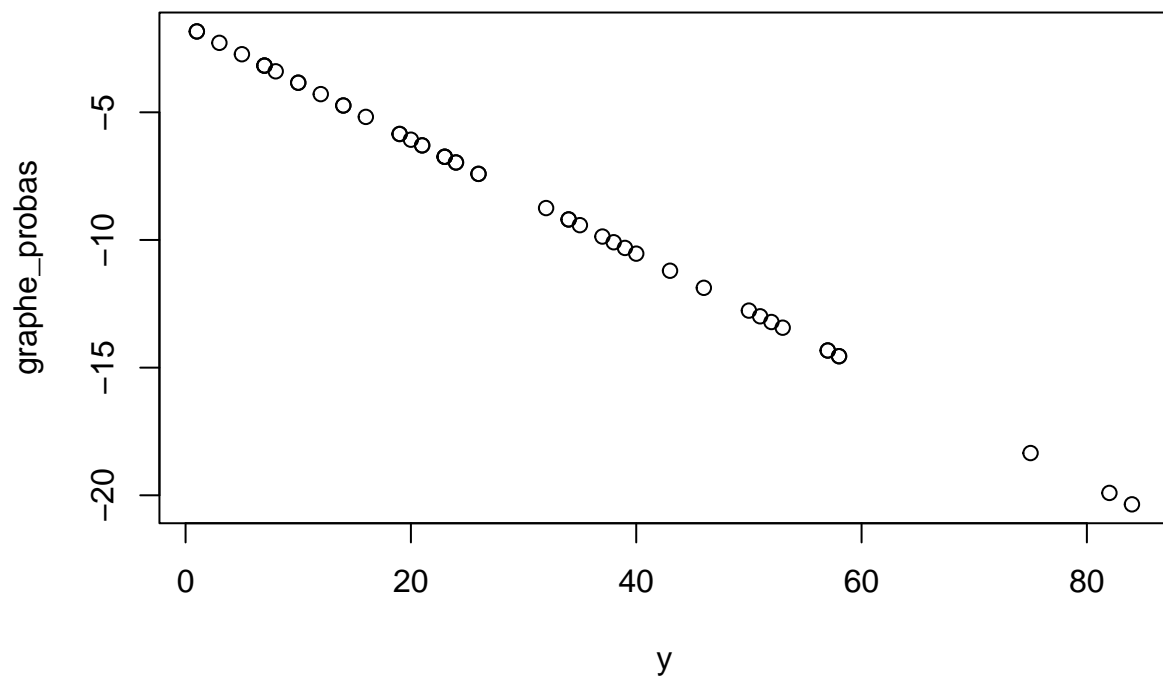
4)c)

```

graphe_probas <- sapply(y, function(x) {
  return(log(dgeom(x,1/lambda)))
})

```

```
plot(y,graphe_probas)
```



La courbe est décroissante car $\ln(P(X = x)) = (x - 1)\ln(1 - p) + \ln(p)$ avec $1 - p < 1$, donc le coefficient $\ln(1 - p)$ est négatif. Le graphe de probabilité a l'allure d'une droite. Ce qui veut dire que la loi géométrique est un bon modèle pour la distribution des minutes des premiers buts marqués dans les 50 premiers matches de la saison 2021-2022.

4)d) Calculons la fonction de vraisemblance pour des réalisations x_1, \dots, x_n de variables aléatoires X_1, \dots, X_n suivant une loi géométrique.

$L(p; x_1, \dots, x_n) = \prod_{i=1}^n P(X_i = x_i) = \prod_{i=1}^n (1-p)^{x_i-1} p = p^n (1-p)^{(\sum_{i=1}^n x_i) - n} = (\frac{p}{1-p})^n (1-p)^{\sum_{i=1}^n x_i}$ Donc,
 $\ln(L(p; x_1, \dots, x_n)) = n * \ln(\frac{p}{1-p}) + (1-p) \sum_{i=1}^n x_i$ Et, $\frac{\partial}{\partial p} \ln(L(p; x_1, \dots, x_n)) = \frac{n * (1-p) - p * (\sum_{i=1}^n x_i) - n}{p(1-p)^2} = \frac{n - p * (\sum_{i=1}^n x_i)}{p(1-p)^2}$. Cette dérivée s'annule pour $p = \frac{n}{\sum_{i=1}^n x_i}$. Ainsi, l'estimation du maximum de vraisemblance de p sous l'hypothèse de loi géométrique est : $\hat{p} = \frac{n}{\sum_{i=1}^n x_i}$.

```
n<- length(saison22_goals)
somme<-sum(saison22_goals)
cat("L'estimation numérique du maximum de vraisemblance de p sous l'hypothèse de loi géométrique est : ", n/somme)

## L'estimation numérique du maximum de vraisemblance de p sous l'hypothèse de loi géométrique est : 0.3210084

cat ("L'estimation du nombres de minutes à attendre pour voir le premier but marqué est : ", 90*n/somme)

## L'estimation du nombres de minutes à attendre pour voir le premier but marqué est : 32.10084 min
```

L'estimation du nombres de minutes à attendre pour voir le premier but marqué correspond très bien à celle prévue à la question 4)d) (32 minutes) ! En fait, ici, l'estimation de p est la même que ce soit pour une loi géométrique ou pour une loi de Bernoulli.

4)e) Pour comparer les deux lois de probabilités, on peut comparer leurs intervalles de confiances à un seuil $\alpha = 0.05$ par exemple. Par exemple, comparons les intervalles de confiances à un seuil $\alpha = 0.05$ du nombre de buts total dans une saison.

```
p <- n/somme

cat("Intervalle de confiance Ã 5%, [", (mean(saison22_goals) - (sqrt(mean(saison22_goals))*q/sqrt(n))), "]", 1/(p+qnorm(0.975)*sqrt(p*(1-p)/n)), ", ", 1/(p-qnorm(0.975)*sqrt(p*(1-p)/n)))

## Intervalle de confiance Ã 5%, [ 2.635754 , 2.971703 ]

cat("Intervalle de confiance Ã 5%, [", 1/(p+qnorm(0.975)*sqrt(p*(1-p)/n)), ", ", 1/(p-qnorm(0.975)*sqrt(p*(1-p)/n)))

## Intervalle de confiance Ã 5%, [ 2.470892 , 3.240022 ]
```

On observe que l'intervalle de confiance en supposant que la loi de Poisson est un modèle adapté pour les réalisations, est plus resserré que celui en supposant que la loi géométrique est un modèle adapté pour les réalisations. On peut en conclure qu'on peut être plus précis en considérant que les données suivent une loi de Poisson.

III - Vérifications expérimentales à base de simulations

1-

```
proportions<- function(m,n,lambd,a,alpha){
x <- rpois(n*m,lambd)

X <-matrix(x,nrow=n,ncol=m)
q <- qnorm(1-(alpha/2))

b_inf <- apply(X,2,FUN = function(x) mean(x) - (sqrt(mean(x))*q/sqrt(n)))
b_sup <- apply(X, 2, FUN = function(x) mean(x) + (sqrt(mean(x))*q/sqrt(n)))
```

```

test_inf <- (lambda > b_inf)
test_sup <- (lambda < b_sup)
cat("Proportion de",m,"échantillons de taille",n," contenant lambda = ", lambda,"pour un seuil ",alpha,
}

cat("On fait varier alpha : \n")

## On fait varier alpha :
proportions(1000,1000,2,0.01)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 2 pour un seuil 0.01 : 0.987
proportions(1000,1000,2,0.05)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 2 pour un seuil 0.05 : 0.945
proportions(1000,1000,2,0.10)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 2 pour un seuil 0.1 : 0.899
proportions(1000,1000,2,0.20)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 2 pour un seuil 0.2 : 0.777
cat("On fait varier lambda : \n")

## On fait varier lambda :
proportions(1000,1000,1,0.05)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 1 pour un seuil 0.05 : 0.944
proportions(1000,1000,10,0.05)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 10 pour un seuil 0.05 : 0.952
proportions(1000,1000,100,0.05)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 100 pour un seuil 0.05 : 0.955
proportions(1000,1000,1000,0.05)

## Proportion de 1000 échantillons de taille 1000  contenant lambda = 1000 pour un seuil 0.05 : 0.96
cat("on fait varier n : \n")

## on fait varier n :
proportions(1000,5,2,0.05)

## Proportion de 1000 échantillons de taille 5  contenant lambda = 2 pour un seuil 0.05 : 0.923
proportions(1000,10,2,0.05)

## Proportion de 1000 échantillons de taille 10  contenant lambda = 2 pour un seuil 0.05 : 0.95
proportions(1000,50,2,0.05)

## Proportion de 1000 échantillons de taille 50  contenant lambda = 2 pour un seuil 0.05 : 0.937
proportions(1000,10000,2,0.05)

## Proportion de 1000 échantillons de taille 10000  contenant lambda = 2 pour un seuil 0.05 : 0.948

```



```
cat("On fait varier m : \n")
```

```
## On fait varier m :
```

```
proportions(10,1000,2,0.05)
```

```
## Proportion de 10 échantillons de taille 1000 contenant lambda = 2 pour un seuil 0.05 : 0.9
```

```
proportions(100,1000,2,0.05)
```

```
## Proportion de 100 échantillons de taille 1000 contenant lambda = 2 pour un seuil 0.05 : 0.97
```

```
proportions(1000,1000,2,0.05)
```

```
## Proportion de 1000 échantillons de taille 1000 contenant lambda = 2 pour un seuil 0.05 : 0.941
```

```
proportions(10000,1000,2,0.05)
```

```
## Proportion de 10000 échantillons de taille 1000 contenant lambda = 2 pour un seuil 0.05 : 0.9511
```

D'après les observations, on peut conclure qu'en simulant un grand nombre d'échantillons de taille n de loi de Poisson de paramètre λ , on a un bien une proportion d'approximativement $1 - \alpha$ des intervalles de confiances du seuil α qui contient la vraie valeur λ .

En faisant varier α , on fait varier la proportion, qui vaut toujours approximativement $1 - \alpha$. En faisant varier λ , la proportion ne change pas en moyenne. Quand n et/ou m tendent vers l'infini, la proportion converge vers $1 - \alpha$.

2-

```
proba <- function(n,m,lambda,alpha,epsilon){
```

```
x <- rpois(n*m,lambda)
```

```
X <-matrix(x,nrow=n,ncol=m)
```

```
q <- qnorm(1-(alpha/2))
```

```
Tn <- apply(X,2,FUN = function(x) mean(x))
```

```
test_inf <- (abs(Tn-lambda)>epsilon)
```

```
mean(test_inf)
```

```
}
```

```
x<-seq(1,1000,5)
```

```
y <- numeric(length(x))
```

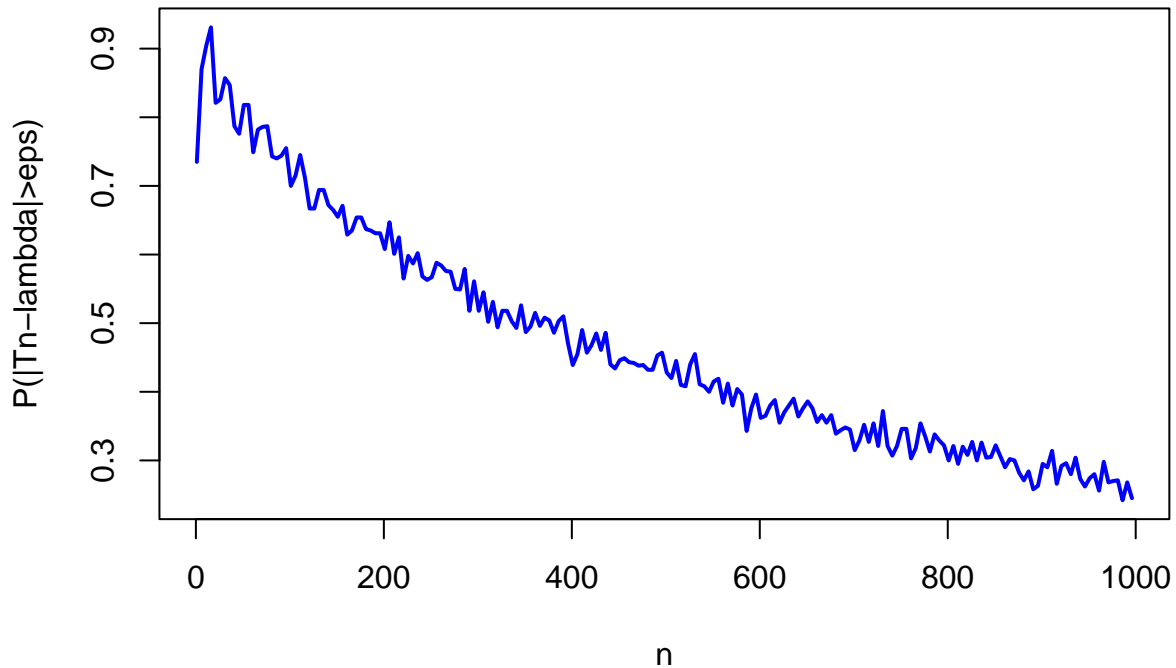
```
for (i in seq_along(x)) {
```

```
  y[i] <- proba(x[i], 1000, 2, 0.05, 0.05)
```

```
}
```

```
plot(x,y,type = "l", col = "blue", lwd = 2, xlab = "n", ylab = "P(|Tn-lambda|>eps)", main = "Courbe de l
```

Courbe de $P(|T_n - \lambda| > \epsilon)$, $\epsilon = 0.05$



Cette courbe de l'évolution de $P(|T_n - \lambda| > \epsilon)$ en fonction de n montre bien que l'estimateur converge en probabilité.

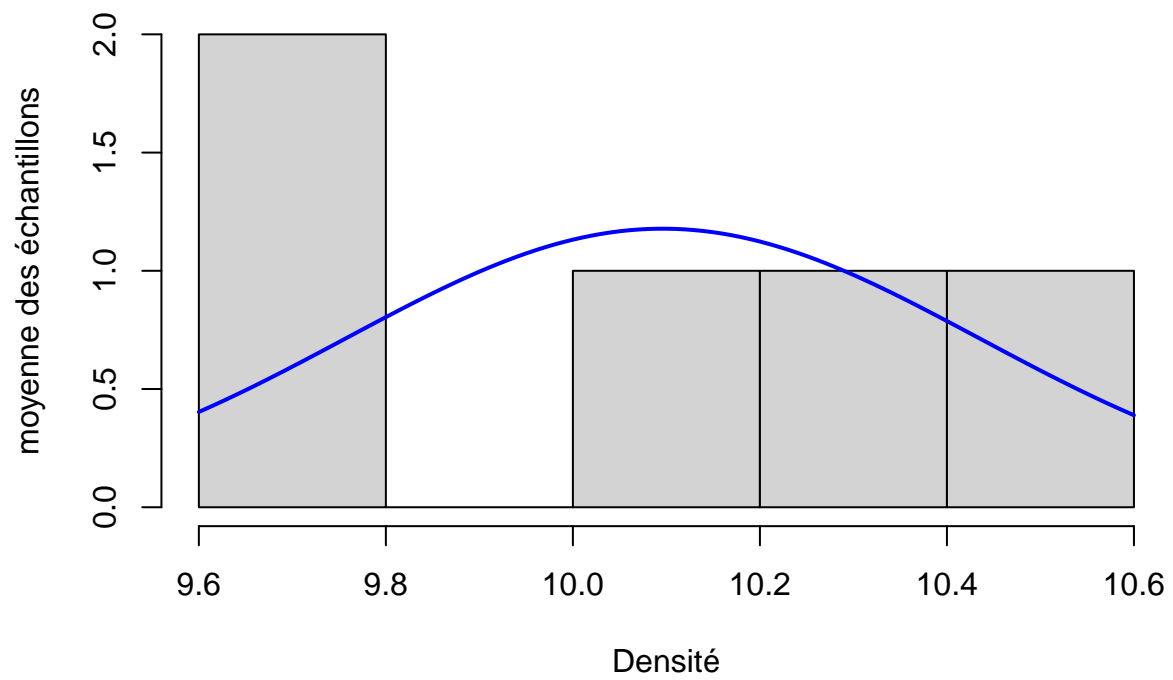
3-

```
histo <- function(n,m,lambda){
  x <- rpois(n*m,lambda)
  X <- matrix(x,nrow=n,ncol=m)

  moyenne_des_echantillons <- rowMeans(X)
  loi_normale <- pnorm(n,mean=mean(moyenne_des_echantillons),sd=sd(moyenne_des_echantillons))
  hist(moyenne_des_echantillons, prob = TRUE, xlab = "Densité", ylab = "moyenne des échantillons", main="")
  curve(dnorm(x,mean(moyenne_des_echantillons),sd(moyenne_des_echantillons)), add = TRUE, col = "blue", lty=2)
}

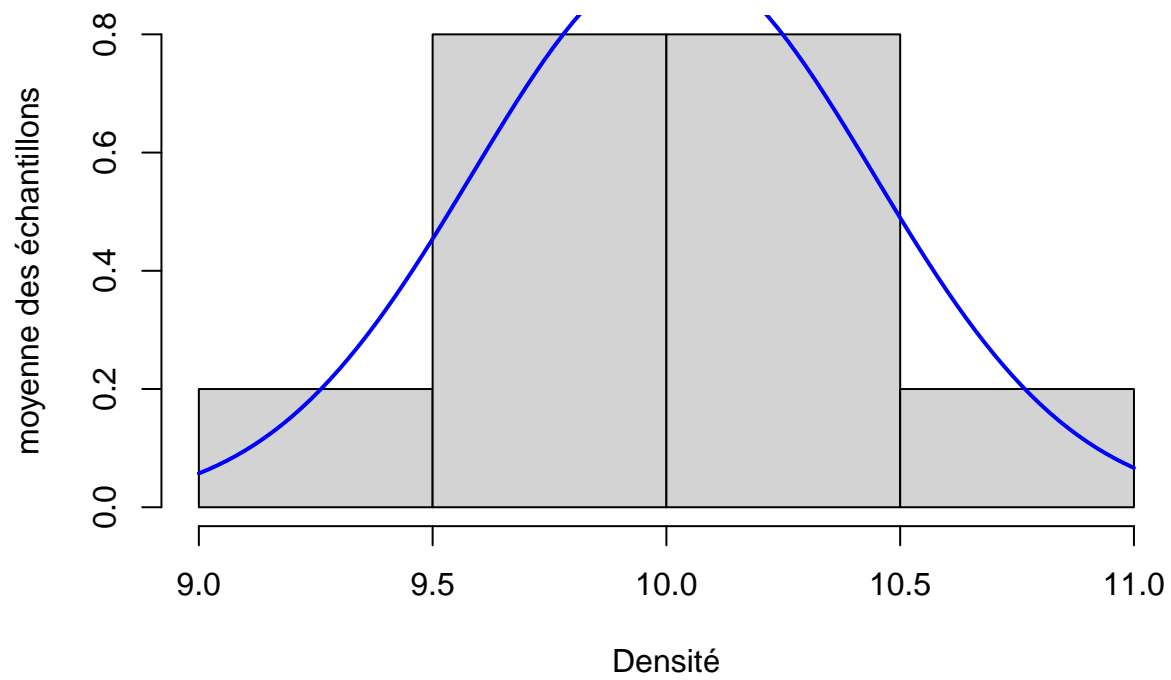
graphe <- function(n,m,lambda){
}
histo(5,50,10)
```

Histogramme de la moyenne des échantillons



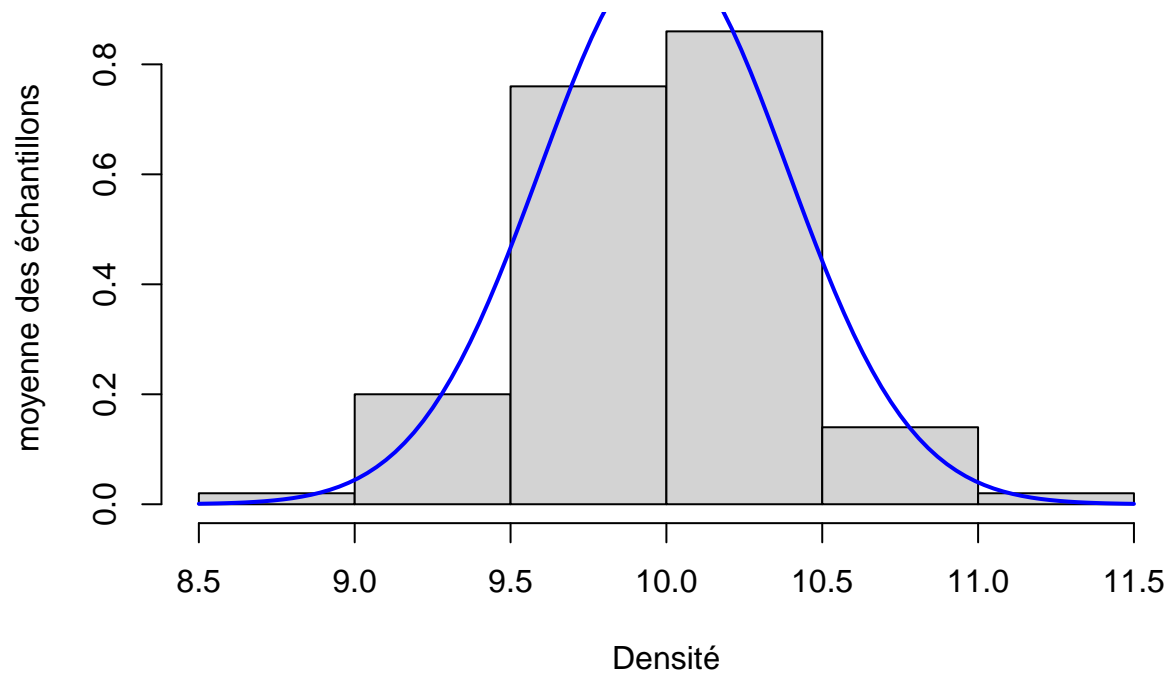
```
histo(10,50,10)
```

Histogramme de la moyenne des échantillons



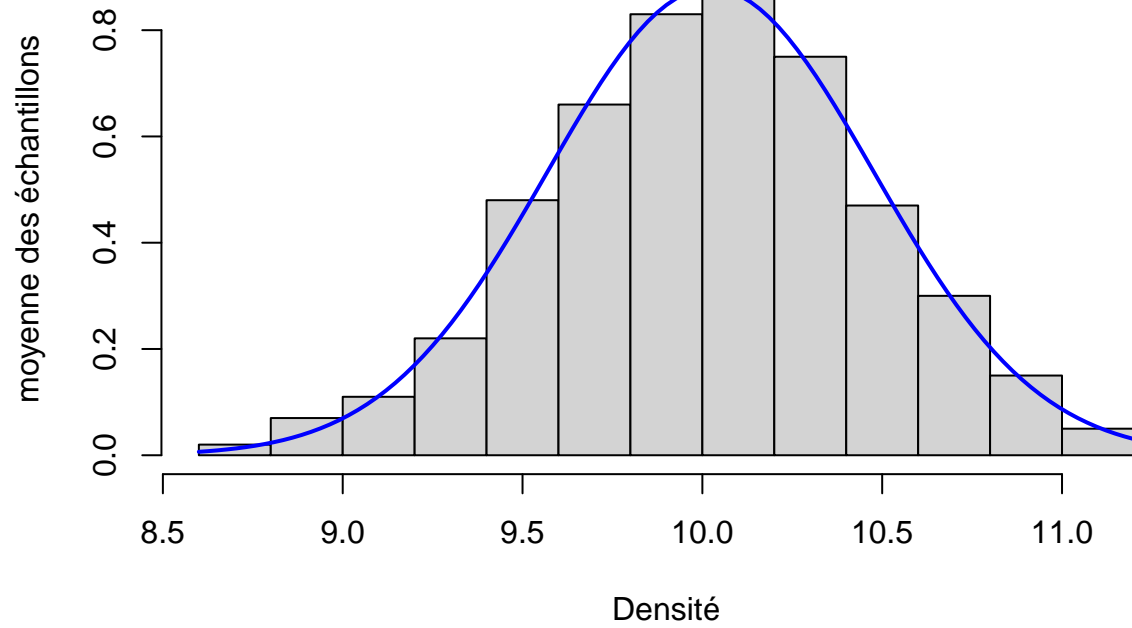
```
histo(100,50,10)
```

Histogramme de la moyenne des échantillons



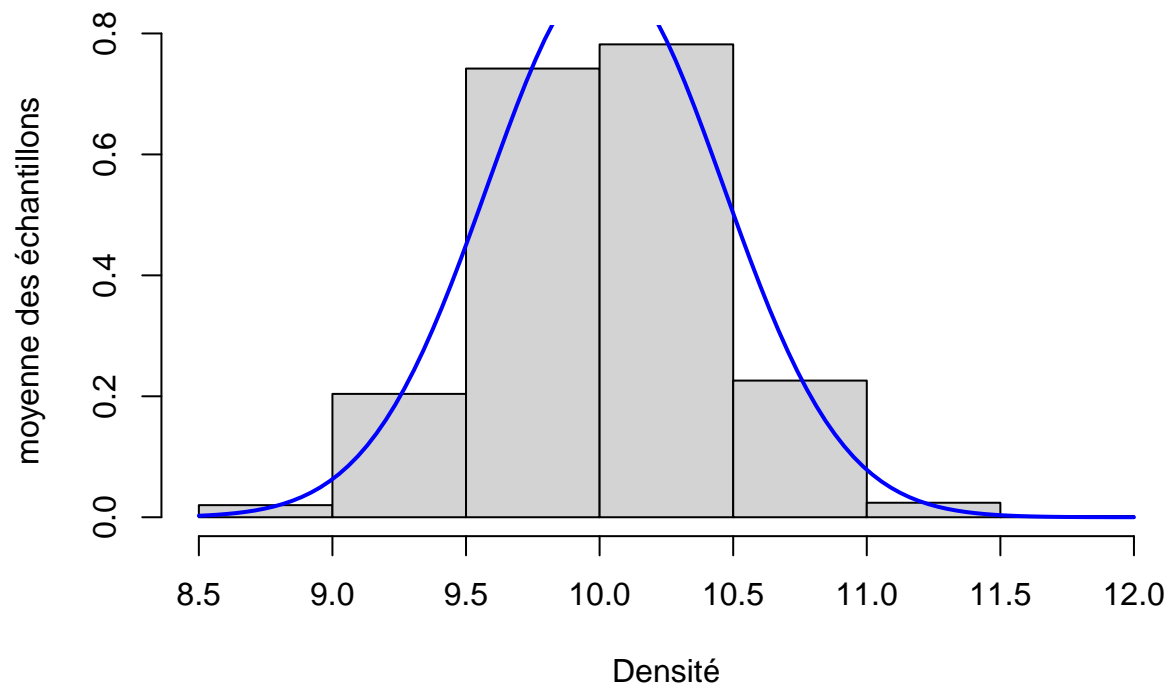
```
histo(500,50,10)
```

Histogramme de la moyenne des échantillons



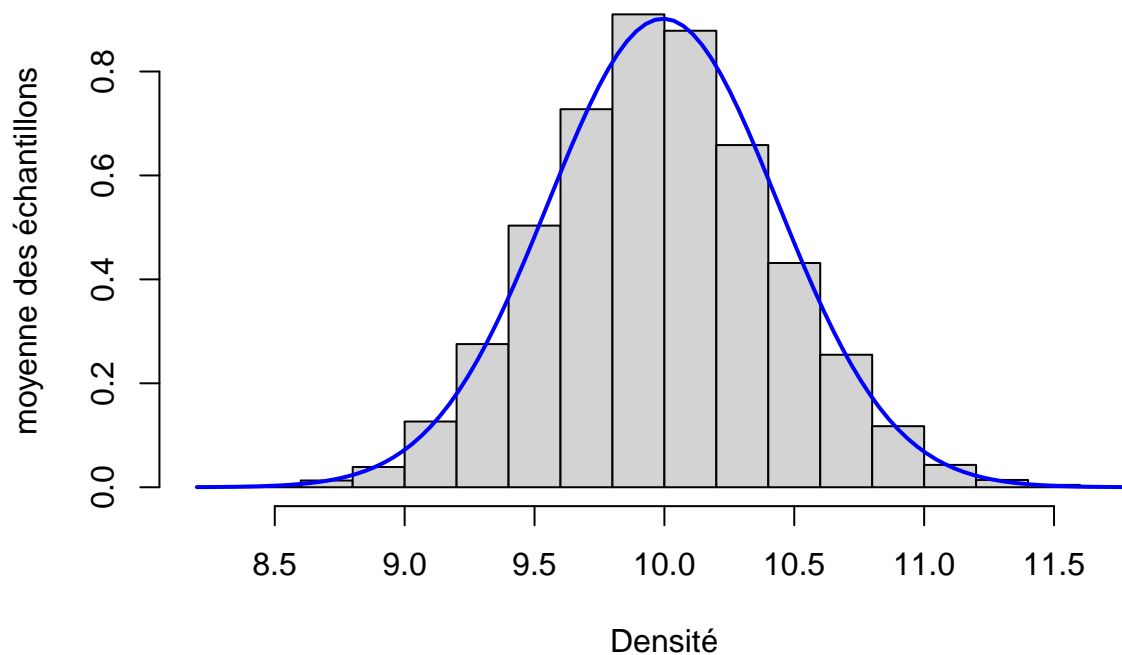
```
histo(1000,50,10)
```

Histogramme de la moyenne des échantillons



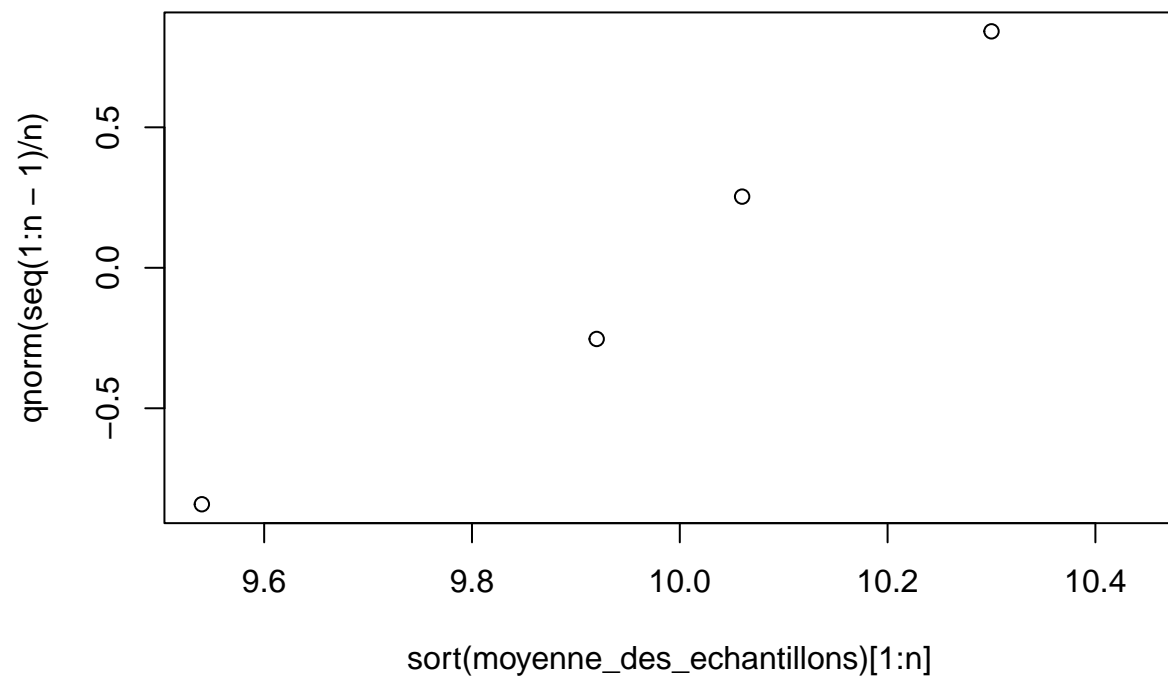
```
histo(10000,50,10)
```

Histogramme de la moyenne des échantillons

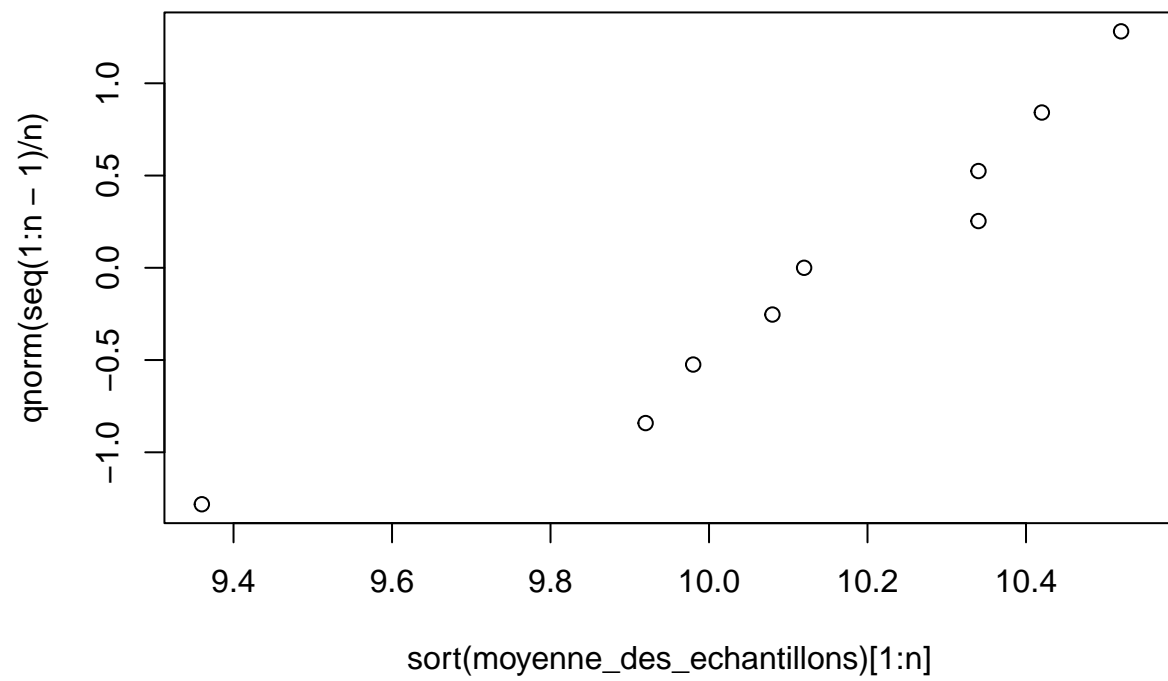


On peut voir à travers les histogrammes que la loi de l'estimateur tend asymptotiquement vers une loi normale lorsque n augmente (la densité d'une loi normale de paramètres bien choisis épouse bien la forme de l'histogramme).

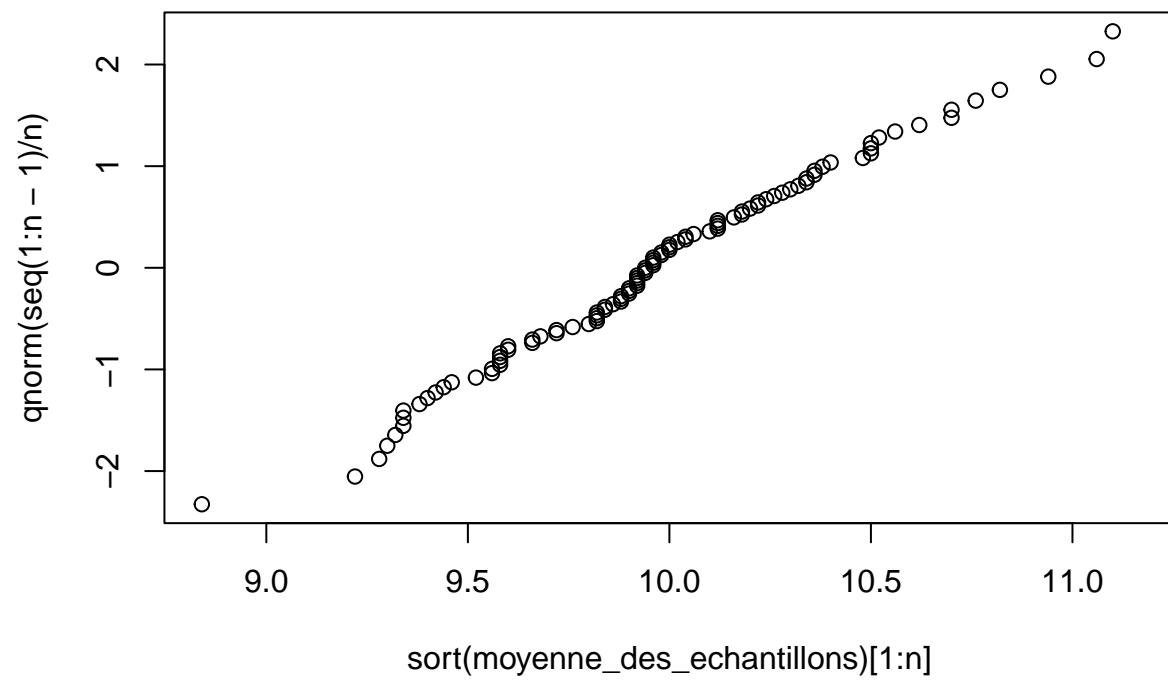
```
graphe_de_probas<- function(n,m,lambda){  
  x <- rpois(n*m,lambda)  
  X <-matrix(x,nrow=n,ncol=m)  
  
  moyenne_des_echantillons <- rowMeans(X)  
  xx<-sort(moyenne_des_echantillons)[1:n]  
  y<-qnorm(seq(1:n-1)/n)  
  plot(sort(moyenne_des_echantillons)[1:n],qnorm(seq(1:n-1)/n))  
}  
graphe_de_probas(5,50,10)
```

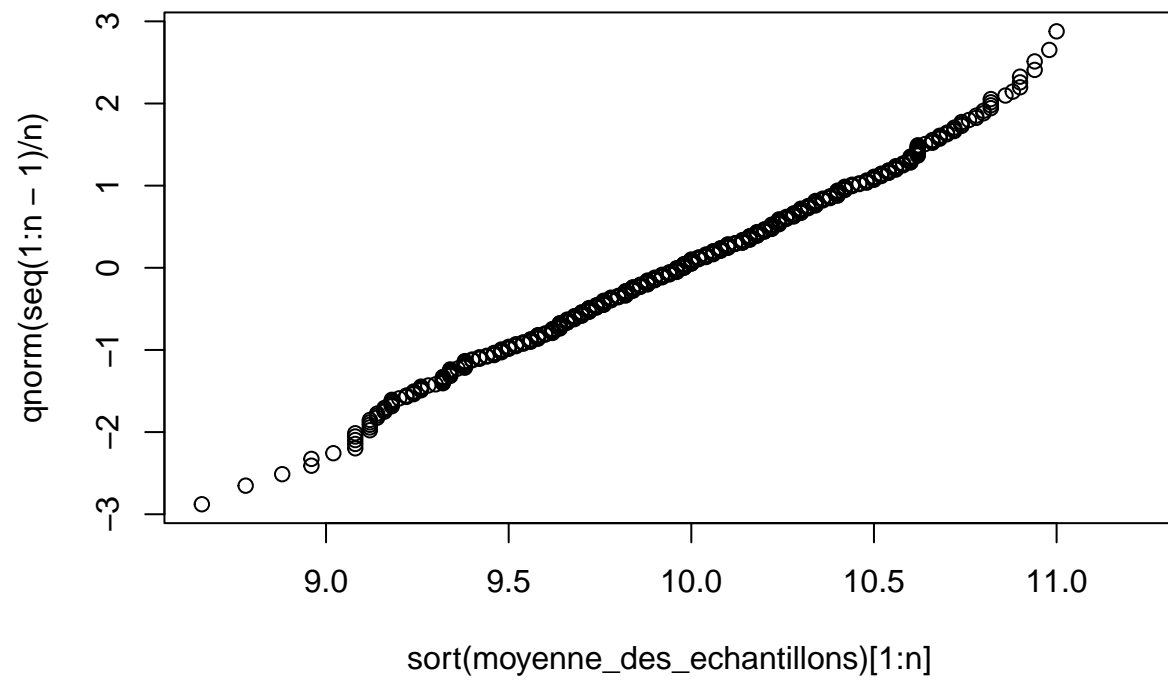
```
graphe_de_probab(10,50,10)
```



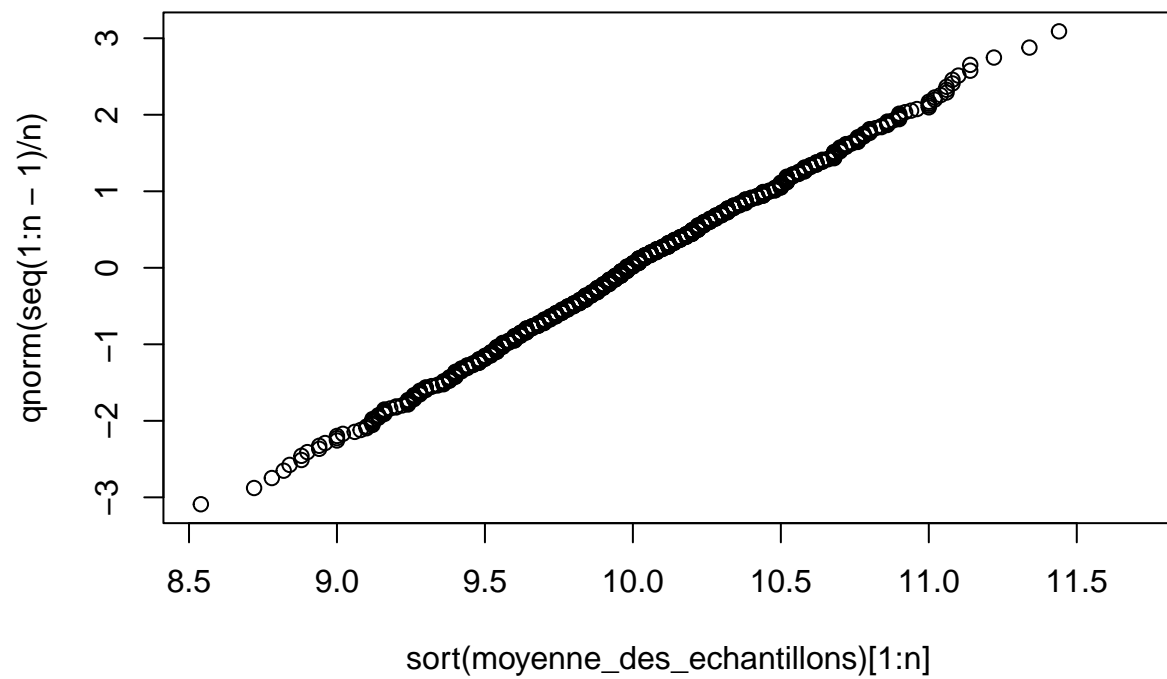
```
graphe_de_probas(100,50,10)
```



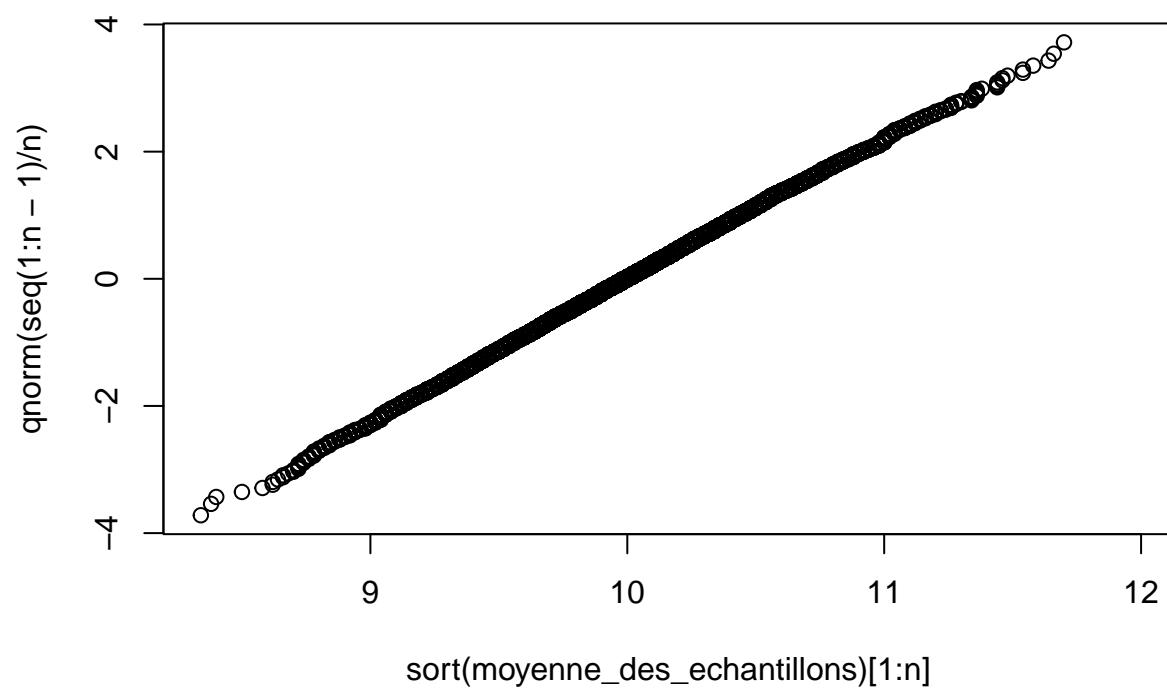
```
graphe_de_probas(500,50,10)
```



```
graphe_de_probab(1000,50,10)
```



```
graphe_de_probas(10000,50,10)
```



On arrive à la même conclusion en observant les graphes de probabilités. La courbe du graphe de probabilités ressemble de plus en plus à une droite ce qui montre que la distribution de l'estimateur tend asymptotiquement vers une loi normale.