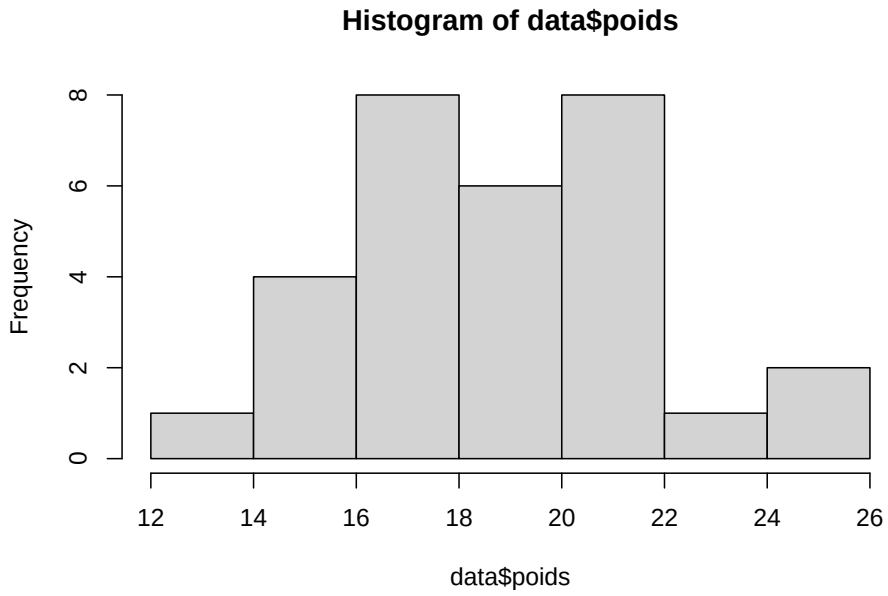
TP 9 : Estimation par intervalles de confiance, introduction aux tests statistiques

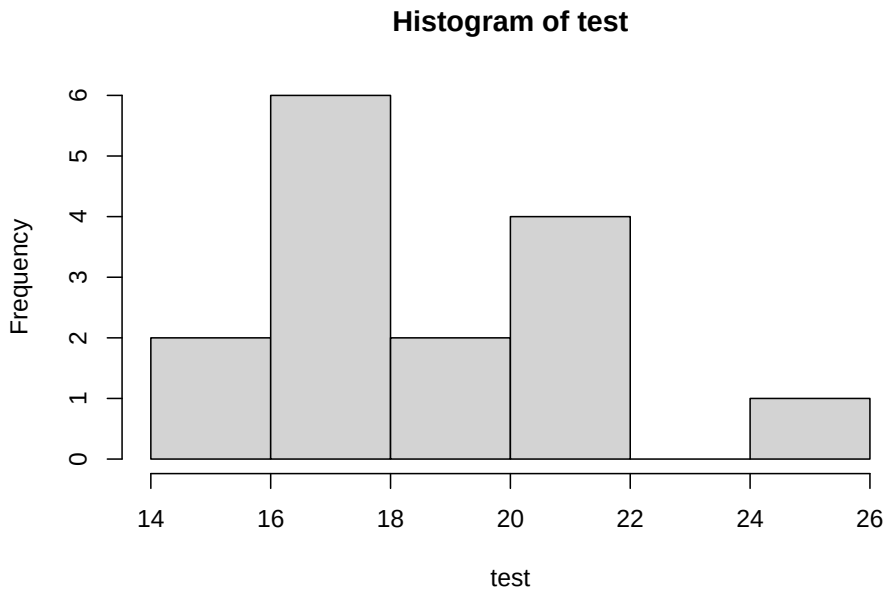
Exercice 1 : Tester l'effet du maïs OGM

1. Voici le code :

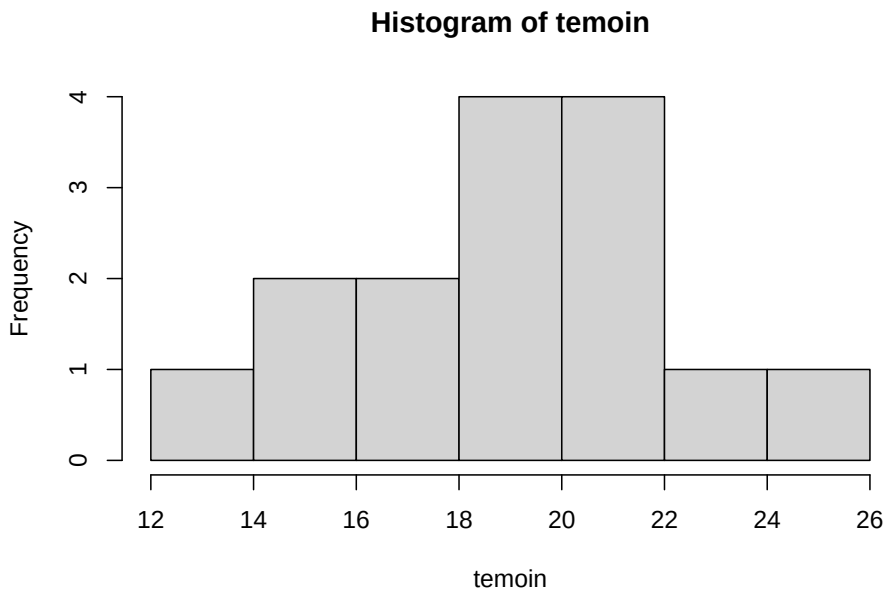
```
data = read.csv("souris30.csv")
hist(data$poids)
```




```
test = data$poids[data$groupe=="test"]
hist(test)
```




```
temoin = data$poids[data$groupe=="temoin"]
hist(temoin)
```



Réponse : au vu des histogrammes, l'hypothèse gaussienne paraît assez cohérente, bien que le jeu de données soit trop petit pour en être plus convaincu.

2. Réponse : en inversant la formule précédente, un intervalle de confiance asymptotique de niveau $1-\alpha$ pour la moyenne est

$$\left[\bar{X}_n - \frac{\sqrt{v_n}}{\sqrt{n}} t_{1-\alpha/2}^{(n-1)}, \bar{X}_n - \frac{\sqrt{v_n}}{\sqrt{n}} t_{\alpha/2}^{(n-1)}\right].$$

```
alpha = 0.05
n = length(test)
ICtest1 = mean(test) - sqrt(var(test))/sqrt(n) * qt(1-alpha/2,n-1)
ICtest2 = mean(test) - sqrt(var(test))/sqrt(n) * qt(alpha/2,n-1)
print(paste("L'IC de niveau 0.95 pour la moyenne test est [",ICtest1,",",ICtest2,"]."))
```

```
## [1] "L'IC de niveau 0.95 pour la moyenne test est [ 17.0682856338293 , 20.0650476995041 ]."
```

```
n = length(temoin)
ICtemoin1 = mean(temoin) - sqrt(var(temoin))/sqrt(n) * qt(1-alpha/2,n-1)
ICtemoin2 = mean(temoin) - sqrt(var(temoin))/sqrt(n) * qt(alpha/2,n-1)
print(paste("L'IC de niveau 0.95 pour la moyenne temoin est [",ICtemoin1,",",ICtemoin2,"]."))
```

```
## [1] "L'IC de niveau 0.95 pour la moyenne temoin est [ 17.1730623547047 , 20.8669376452953 ]."
```

3. Voici le code :

```
t.test(test)
```

```
##
##  One Sample t-test
##
## data:  test
## t = 26.576, df = 14, p-value = 2.214e-13
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  17.06829 20.06505
## sample estimates:
## mean of x
##  18.56667
```

```
t.test(temoin)
```

```
##
##  One Sample t-test
##
## data:  temoin
## t = 22.087, df = 14, p-value = 2.789e-12
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  17.17306 20.86694
## sample estimates:
## mean of x
##     19.02
```

Réponse : les intervalles sont très similaires, il est impossible de conclure sur le fait que les moyennes sont différentes. Le problème est que l'échantillon n'est pas représentatif (15 souris, ce n'est pas assez) pour avoir un intervalle de confiance assez étroit.

4. Voici le code :

```
data = read.csv("souris300.csv")
test = data$poids[data$groupe=="test"]
temoin = data$poids[data$groupe=="temoin"]

alpha = 0.05
n = length(test)
ICtest1 = mean(test) - sqrt(var(test))/sqrt(n) * qt(1-alpha/2,n-1)
ICtest2 = mean(test) - sqrt(var(test))/sqrt(n) * qt(alpha/2,n-1)
print(paste("L'IC de niveau 0.95 pour la moyenne test est [",ICtest1,",",ICtest2,"]."))
```

```
## [1] "L'IC de niveau 0.95 pour la moyenne test est [ 17.9237688378048 , 18.6215644955285 ]."
```

```
n = length(temoin)
ICtemoin1 = mean(temoin) - sqrt(var(temoin))/sqrt(n) * qt(1-alpha/2,n-1)
ICtemoin2 = mean(temoin) - sqrt(var(temoin))/sqrt(n) * qt(alpha/2,n-1)
print(paste("L'IC de niveau 0.95 pour la moyenne temoin est [",ICtemoin1,",",ICtemoin2,"]."))
```

```
## [1] "L'IC de niveau 0.95 pour la moyenne temoin est [ 18.8056922630628 , 19.7623077369372 ]."
```

Réponse : les intervalles sont cette fois-ci disjoints, on peut donc rejeter l'hypothèse "les moyennes sont égales" avec une probabilité d'erreur de moins de 5%. L'échantillon est assez représentatif pour conclure ici. On peut aussi faire le test en question (avec la p-value) avec la fonction `t.test(test,temoin)`:

```
t.test(test,temoin)
```

```
##
##  Welch Two Sample t-test
##
## data:  test and temoin
## t = -3.3755, df = 272.58, p-value = 0.0008441
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.6011809 -0.4214857
## sample estimates:
## mean of x mean of y
##  18.27267  19.28400
```

La p-value est très faible, la probabilité de faire une erreur en rejetant l'hypothèse "les moyennes sont égales" est donc même bien plus faible que 5% ! Il y a donc différence significative entre les souris nourries au maïs OGM (les poids sont plus faibles) et celles nourries normalement (les poids sont plus importants).

Exercice 2 : Intervalle de confiance asymptotique pour le paramètre d'une loi de Poisson.

1. Réponse : on trouve l'intervalle par excès

$$\left[\bar{X}_n - \frac{\sqrt{4\alpha n+1}-1}{2\alpha n}\ ,\ \bar{X}_n + \frac{\sqrt{4\alpha n+1}+1}{2\alpha n}\right].$$

2. Voici le code :

```
IC1 = function(ech,alpha){
  n = length(ech)
  xbar = mean(ech)
  borneinf = xbar - (sqrt(4*alpha*n+1) - 1)/(2*alpha*n)
  bornesup = xbar + (sqrt(4*alpha*n+1) + 1)/(2*alpha*n)
  return(c(borneinf, bornesup))
}

ech = rpois(300,lambda=2)
alpha = 0.05
IC1(ech,alpha)
```

```
## [1] 1.666325 2.187008
```

3. Réponse :

$$\frac{\bar{X}_n - \lambda}{\sqrt{\lambda}} \to \mathcal{N}(0,1).$$

On a de plus

$$\sqrt{\bar{X}_n} \to \sqrt{\lambda}$$

presque sûrement et donc en loi. C'est le Lemme de Slutsky qui permet de conclure à la seconde convergence en loi (en remplacant la moyenne empirique au dénominateur par λ). On trouve alors l'intervalle exact asymptotique

$$\left[\bar{X}_n - q_{1-\alpha/2}\sqrt{\frac{\bar{X}_n}{n}}\ ,\ \bar{X}_n - q_{\alpha/2}\sqrt{\frac{\bar{X}_n}{n}}\right]$$

qui se réexprime, en utilisant la symétrie de la Gaussienne :

$$\left[\bar{X}_n - q_{1-\alpha/2}\sqrt{\frac{\bar{X}_n}{n}}\ ,\ \bar{X}_n + q_{1-\alpha/2}\sqrt{\frac{\bar{X}_n}{n}}\right].$$

4. Voici le code :

```
IC2 = function(ech,alpha){
  n = length(ech)
  xbar = mean(ech)
  borneinf = xbar - qnorm(1-alpha/2)*sqrt(xbar/n)
  bornesup = xbar + qnorm(1-alpha/2)*sqrt(xbar/n)
  return(c(borneinf, bornesup))
}

ech = rpois(30,lambda=2)
alpha = 0.05
IC1(ech,alpha)
```

```
## [1] 1.451416 3.215250
```

```
IC2(ech,alpha)
```

```
## [1] 1.493939 2.506061
```

```
ech = rpois(30,lambda=0.2)
alpha = 0.2
IC1(ech,alpha)
```

```
## [1] -0.2000000  0.6333333
```

```
IC2(ech,alpha)
```

```
## [1] 0.04789656 0.21877010
```

```
ech = rpois(300,lambda=21)
alpha = 0.01
IC1(ech,alpha)
```

```
## [1] 20.10907 21.31093
```

```
IC2(ech,alpha)
```

```
## [1] 19.86928 21.21738
```

Commentaires : pour tous les paramètres choisis, IC2 est toujours plus précis (plus étroit) que IC1.

5. Voici le code :

```
lambda = 3
alpha = 0.05
N = 10000
n = 5000
dansIC1 = replicate(N,0)
dansIC2 = replicate(N,0)

for (i in 1:N){
  ech = rpois(n,lambda)
  if (IC1(ech,alpha)[1] <= lambda & IC1(ech,alpha)[2] >= lambda ){
    dansIC1[i]=1
  }
  if (IC2(ech,alpha)[1] <= lambda & IC2(ech,alpha)[2] >= lambda ){
    dansIC2[i]=1
  }
}

probaIC1 = mean(dansIC1)
probaIC2 = mean(dansIC2)

probaIC1
```

```
## [1] 0.989
```

```
probaIC2
```

```
## [1] 0.9501
```

Réponse : on voit que IC2 est effectivement un intervalle exact, au sens où la proba que le paramètre réel soit dans l'intervalle est estimée très proche de $1-\alpha$. En revanche, IC1 est par excès : il est plus large, car la vraie probabilité d'erreur est en réalité plus faible que α. Cela s'explique par le fait que l'inégalité de BT n'est pas optimale : c'est juste une inégalité qui majore une proba de façon realtivement bonne mais tout de même assez grossière.

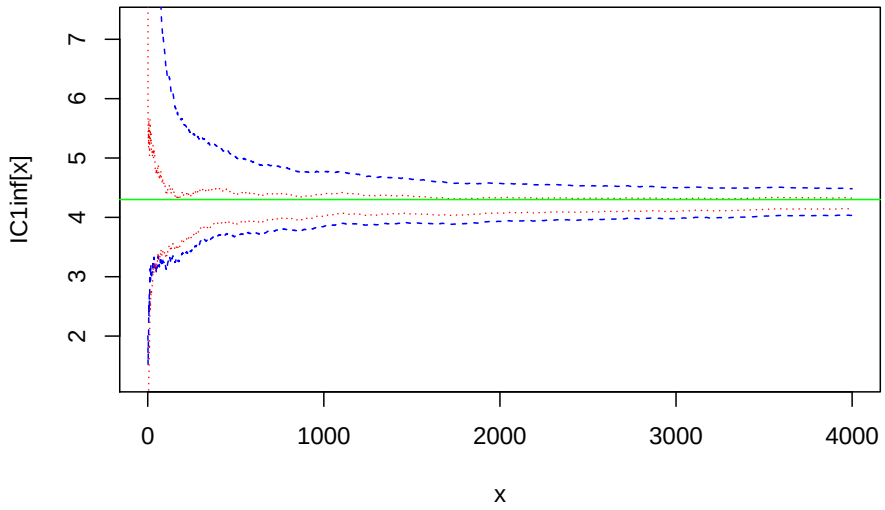
6. Pour aller plus loin Voici le code :

```
alpha = 0.005
lambda = 4.3
n = 4000
ech = rpois(n,lambda)
IC1inf = replicate(n,0)
IC1sup = replicate(n,0)
IC2inf = replicate(n,0)
IC2sup = replicate(n,0)

for (k in 1:n){
  IC1inf[k] = IC1(ech[1:k],alpha)[1]
  IC1sup[k] = IC1(ech[1:k],alpha)[2]
  IC2inf[k] = IC2(ech[1:k],alpha)[1]
  IC2sup[k] = IC2(ech[1:k],alpha)[2]
}

x=1:n
plot(x,IC1inf[x],col="blue",type = "l",lty = 2,ylim=c(lambda-3,lambda+3))
```

```
lines(x, IC1sup[x], col="blue", lty = 2)
lines(x, IC2sup[x], col="red", lty = 3)
lines(x, IC2inf[x], col="red", lty = 3)
abline(a=lambda,b=0,col="green")
```



Commentaires : on visualise bien que IC2 est plus précis, et ce d'autant plus que n est grand.