TP6.Rmd

2022-05-19

Question1

Posons

$$Y_i = ln(X_i)$$

Le rapport de vraisemblance entre les 2 lois normales:

$$\begin{aligned} Z_n = \frac{L(Y;\mu_1)}{L(Y;\mu_0)} &= e^{(\frac{-[\sum_{i=1}^{n}(Y_i-\mu_1)^2)-\sum_{i=1}^{n}(Y_i-\mu_0)^2)]}{2\sigma_0^2})} \\ &= e^{\frac{(\mu_1-\mu_0)\sum_{i=1}^{n}(Y_i)}{\sigma_0^2}} e^{\frac{-n(\mu_1^2-\mu_0^2)}{\sigma_0^2}} \end{aligned}$$

est une variable aléatoire continue sous $\mathbb{P}(\mu_0)$

D'après le lemme de Neyman-Pearson, la région critique optimale au seuil α est

$$\begin{aligned} W &= \left\{ (y_1,...,y_n); e^{\frac{[\sum_{i=1}^{n}(y_i-\mu_0)^2-\sum_{i=1}^{n}(y_i-\mu_1)^2]}{2\sigma_0^2})} > k \right\} \\ &= \left\{ (y_1,...,y_n); e^{\frac{(\mu_1-\mu_0)\sum_{i=1}^{n}(y_i)}{\sigma_0^2}} e^{\frac{-n(\mu_1^2-\mu_0^2)}{\sigma_0^2}} > k \right\} \\ &= \left\{ (y_1,...,y_n); \frac{1}{n}\sum_{i=1}^{n} y_i > c \right\} \end{aligned}$$

On a donc la statistique de test:

$$T(Y) = \frac{1}{n}\sum_{i=1}^{n} y_i$$

D'où:

$$\boxed{T(X) = \frac{1}{n}\sum_{i=1}^{n} ln(x_i)}$$

Déterminons K_α :

Sous l'hypothèse H_{0}, T suit une loi

$$\mathbb{N}(\mu_0, \frac{\sigma_0^2}{n})$$

Donc, avec phi la fonction de répartition associée à la loi des Yi=ln(Xi):

$$\mathbb{P}_{H_0}(W) = \mathbb{P}_{H_0}(\frac{1}{n}\sum_{i=1}^{n} ln(X_i) > K_\alpha) = \mathbb{P}_{H_0}(\frac{\sqrt{n}(\bar{Y_n} - \mu_0)}{\sigma_0} > \frac{\sqrt{n}(K_\alpha - \mu_0)}{\sigma_0}) = \boxed{1 - \phi(\frac{\sqrt{n}(K_\alpha - \mu_0)}{\sigma_0}) = \alpha}$$

D'où , en fixant un premier alpha:

$$\boxed{K_\alpha = \mu_0 + \frac{\sigma_0}{\sqrt{n}}\phi^{-1}(1-\alpha)}$$

Ainsi que:

$$\mathbb{P}_{H_1}(W) = \mathbb{P}_{H_1}(\bar{Y_n} > K_\alpha) = \boxed{1 - \phi(\frac{\sqrt{n}(\mu_0 - \mu_1)}{\sigma_0} + \phi^{-1}(1-\alpha)) = \beta}$$

```
Dl_norm<-function(x){
  r=0
     for(l in x){
       r=r+log(l)
     }
  return(r/length(x))
 }
```

```
n<-10
sig_0<-1
mu_0<-0
mu_1<-0.1
X_0 <- rlnorm(n,mu_0,sig_0)
X_1 <- rlnorm(n,mu_1,sig_0)

L_0<-Dl_norm(X_0)
L_1<-Dl_norm(X_1)
div<-(1/n)*(L_1/L_0)

alpha_test<-0.05

#K_alpha=mu_0+(sig_0/sqrt(n))*qnorm(1-alpha_test,mu_0,sig_0*sig_0/n)
K_alpha<-(1/sqrt(n))*(1/qlnorm(1-alpha_test,mu_0,1/n))
print(K_alpha)
```

```
## [1] 0.2682656
```

```
alpha<-pnorm(K_alpha,mu_0,1/n)
print(alpha)
```

```
## [1] 0.996348
```

```
beta<-1-pnorm(K_alpha,mu_0,1/n)
print(beta)
```

```
## [1] 0.003652009
```

α est le quantile associé à K_α = taux d'erreur de première espèce (accepter à tort l'hypothèse nulle)≈ 0.3 et

$$\beta = 1 - \alpha$$

le taux d'erreur de seconde espèce (rejeter à tort l'hypothèse nulle)

Question 2

```
list<-1:100
for( l in 1:100){
  lognorm<-rlnorm(10,0,1)
  list[l]<-Dl_norm(lognorm)
}
list_trie<-sort(list)
K_alpha_moy<-list_trie[90]-0.0001
beta_moy<-1-pnorm(K_alpha_moy,0,1/10)
alpha_moy<-pnorm(K_alpha_moy,0,1/10)

print(K_alpha_moy)
```

```
## [1] 0.3497301
```

```
print(alpha_moy)
```

```
## [1] 0.999765
```

```
print(beta_moy)
```

```
## [1] 0.0002349959
```

On retrouve des valeurs similaires malgré une marge d'erreur pour K_α.

```
marge_erreur<-(K_alpha_moy-K_alpha)/K_alpha_moy
print(marge_erreur)
```

```
## [1] 0.2329354
```

Question 3

On utilise encore la fonction de répartition pour trouver la p_value:

```
lognorm<-rlnorm(10,0,1)
p_value<-pnorm(Dl_norm(lognorm),0,1/10)
print(p_value)
```

```
## [1] 0.9948979
```

La p-value sert à jouer sur les zones de rejet et d'acceptation. Lorsque les données sont collectées, la p_value est calculée et la décision suivante est prise :

-si elle est inférieure à α, on rejette l'hypothèse nulle au profit de l'hypothèse alternative

-si elle est supérieure à α, on rejette l'hypothèse alternative au profit de l'hypothèse nulle

Question 4

```
n<-c(20,50,100)
for(x in n){
  lognorm<-rlnorm(x,0,1)
  p_value<-pnorm(Dl_norm(lognorm),0,1/x)
  print(p_value)
}
```

```
## [1] 0.0104018
## [1] 6.289395e-08
## [1] 1
```

Plus la taille augmente, plus la p_value diminue et est incertaine

Question 5

D'après le lemme de Neyman-Pearson, la région de rejet W optimale est définie par l'ensemble des points $(x_1, x_2, .., x_n)$ de R^n tels que : $\frac{\mathcal{L}(x,\theta_1)}{\mathcal{L}(x,\theta_0)} > k_\alpha$. On note S_n^2 l'estimateur sans biais de σ^2.

On choisit les hypothèses

$$Test = \begin{cases} H_0 : \mu = \mu_0 \\ H_1 : \mu = \mu_1 \end{cases}$$

On a $W = \Lambda_n > k_\alpha$ où $\Lambda_n = \frac{\sqrt{n}(\bar{X_n} - \mu_0)}{S_n}$ suit une loi de Student à $n-1$ degrés de liberté notée T_{n-1}. Ainsi $k_\alpha = F^{-1}Tn - 1(1-\alpha)$.

Ainsi la région de rejet est de la forme $W > F^{-1}Tn - 1(1-\alpha)$ avec $F_{T_{n-1}}$ la fonction de répartion de T_{n-1}.

Alors:

$$\mathbb{P}_{H_0}(W) = \boxed{1 - F_{T_{n-1}}(\frac{\sqrt{n}(K_\alpha - \mu_0)}{\sigma_0}) = \alpha}$$

D'où , en fixant un premier alpha:

$$\boxed{K_\alpha = \mu_0 + \frac{S_n}{\sqrt{n}} F^{-1}_{T_{n-1}}(1-\alpha)}$$

Ainsi que:

$$\mathbb{P}_{H_1}(W) = \boxed{1 - F_{T_{n-1}}(\frac{\sqrt{n}(\mu_0 - \mu_1)}{S_n} + F^{-1}_{T_{n-1}}(1-\alpha)) = \beta}$$

```
est_var <- function(echant, mu) {
  n = length(echant);
  return ((1 / n) * (sum((echant - mu) ^ 2)));
}

decision <- function(alpha, Sn, mu0, mu1) {
  mean = mean(Sn);
  n = length(Sn);
  varEstimated = est_var(Sn, mu0);
  val = mu0 + (sqrt(varEstimated) / sqrt(n)) * qt(p = 1 - alpha, df = n - 1);
  if (mean > val) {
    return (mu1);
  }
  return (mu0);
}
```

On simule 100 fois le test

```
matrix = matrix(0, 100, 20);
for (i in seq(1, 100)) {
  matrix[i,] = rnorm(n = 20, mean = 0, sd = 1);
}
```

On applique notre règle de décision:

```
vec_mu = seq(1, 100);
for (i in seq(1, 100)) {
  echant = matrix[i,];
  vec_mu[i] = decision(0.05, echant, 1, 1.5);
}

count <-function(mu_vec) {
  count_1 = 0;
  count_1_5 = 0;
  for (i in seq(1, length(mu_vec))) {
    if (mu_vec[i] == 1) {
      count_1 = count_1 + 1;
    } else {
      count_1_5 = count_1_5 + 1;
    }
  }
  return (c(count_1, count_1_5));
}
```

δ est une règle de décision et prend deux valeurs : 0 et 1. Cette variable aléatoire suit une loi de Bernoulli de paramètre $1 - \alpha$.

La région de rejet est de la forme $W > F^{-1}Tn - 1(1 - \alpha)$. Comme la fonction de répartion d'une variable aléatoire qui suit une loi de Student est une fonction croissante indépendante du degré de liberté de la loi de Student, l'inverse de celle ci est une fonction décroissante. Ainsi lorsque α diminue : $F^{-1}Tn - 1(1 - \alpha)$ augmente.

```
alpha_vec = c(0.2, 0.1, 0.05, 0.01);
n = 20;
counted = matrix(0, length(alpha_vec), 1);
for (j in seq(1, length(alpha_vec))) {
  alpha = alpha_vec[j];
  counted[j] = qt(p = 1 - alpha, df = n - 1);
}
counted;
```

```
##           [,1]
## [1,] 0.8609506
## [2,] 1.3277282
## [3,] 1.7291328
## [4,] 2.5394832
```

```
alpha_vec = c(0.2, 0.1, 0.05, 0.01);
counted = matrix(0, length(alpha_vec), 2);
vec_mu = seq(1,100);
for (j in seq(1, length(alpha_vec))) {
  alpha = alpha_vec[j];
  for (i in seq(1, 100)) {
    echant = matrix[i,];
    vec_mu[i] = decision(alpha, echant, 1, 1.5);
    compt = count(vec_mu);
    counted[j, 1] = compt[1];
    counted[j, 2] = compt[2];
  }
}
```

```
counted;
```

```
##      [,1] [,2]
## [1,]  100    0
## [2,]  100    0
## [3,]  100    0
## [4,]  100    0
```

Ici, la p-value est donnée par $1 - F_{n-1}^{T}(t)$ et correspond au vu de la zone de rejet, à la frontière entre acceptation et rejet d'hypothèse.

```
test_matrix = matrix(0, 100, 20);
for (i in seq(1, 100)) {
  test_matrix[i,] = rnorm(n = 20, mean = 1, sd = sqrt(2));
}
alpha_vec = c(0.2, 0.1, 0.05, 0.01);
```

```
test = c(1, 2, 3, 4);
for (i in 1:length(test)) {
  alpha = alpha_vec[i];
  test[i] = t.test(x = test_matrix, conf.level = 1 - alpha)$p.value;
}
test;
```

```
## [1] 7.827125e-188 7.827125e-188 7.827125e-188 7.827125e-188
```

Les résultats sont cohérents, une p-value basse vient du fait qu'il faille une zone d'acceptation agrandie et dans ces conditions la p-value est inférieure à α.

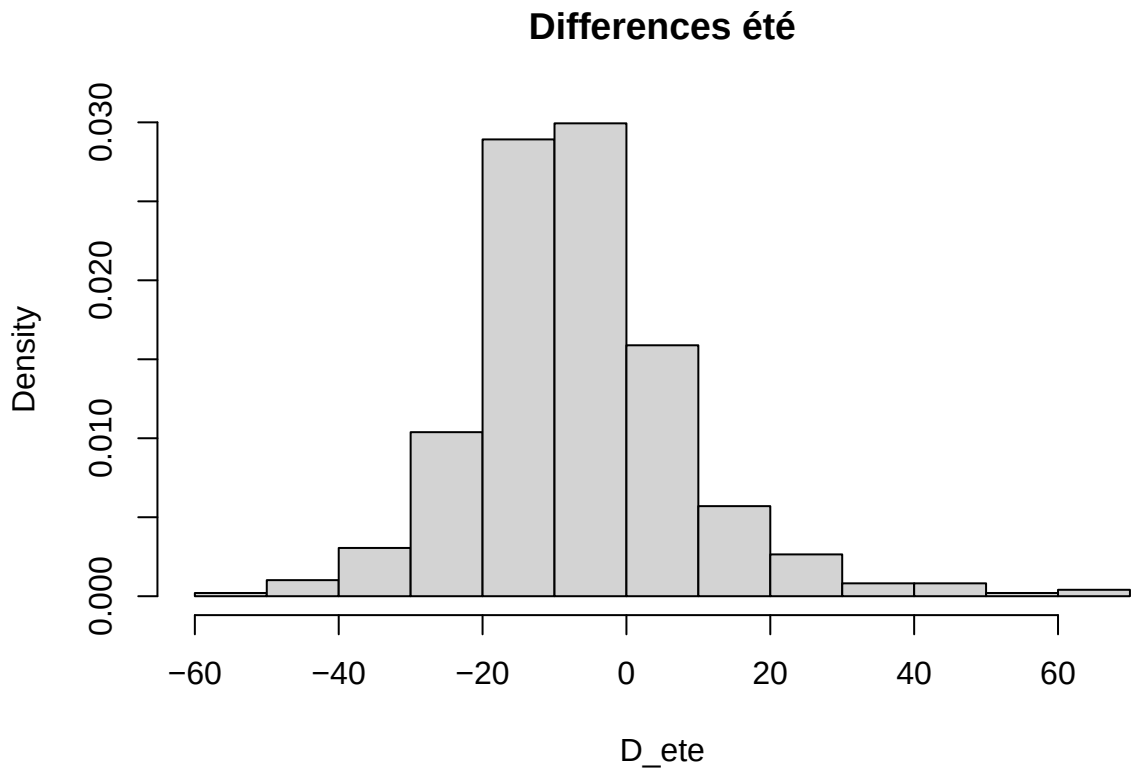
Question 6

On désigne les données sur l'ozone du site urbain par $x_1, \ldots, x_n$ et le site rural par $y_1, \ldots, y_n$, l'indice indiquant les n jours différents pour lesquels nous avons des mesures. L'histogramme ci-dessous montre la différence $d_i = x_i - y_i$ pour $i = 1, \ldots, n$ pour les jours d'été.

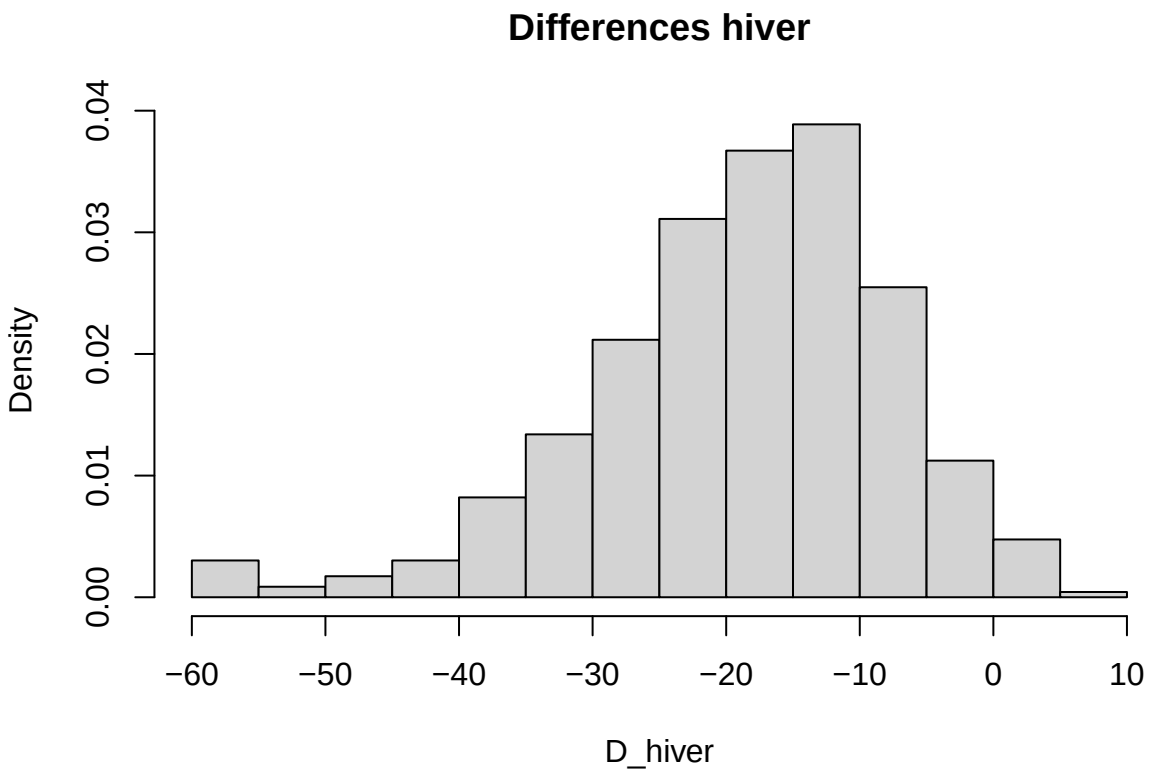
```
ozone.summer = read.csv("summer_ozone.csv")
ozone.winter = read.csv("winter_ozone.csv")

D_ete<-ozone.summer$NEUIL - ozone.summer$RUR.SE
D_hiver<-ozone.winter$NEUIL - ozone.winter$RUR.SE
```

```
hist(D_ete, prob=TRUE, main = "Differences été")
```




```
hist(D_hiver, prob=TRUE, main = "Differences hiver")
```



I) Supposons d'abord que les $D_i \sim \mathbb{N}(\mu, \sigma^2)$

On reprend les calculs avec

$$\boxed{T(X) = \frac{1}{n}\sum_{i=1}^{n} D_{i,ete}}$$

Cas1, Di été

```
D_norm<-function(x){
  r=0
     for(l in x){
       r=r+l
     }
  return(r/length(x))
 }
```

```
str(D_ete)
```

```
##  int [1:491] 3 -13 18 -5 -11 3 -18 -5 -2 -5 ...
```

```
n<-491
mu_ete<-mean(D_ete)
sig_ete<-sd(D_ete)
norm_ete=dnorm(-60:70, mu_ete,sig_ete)

Lete_0<-D_norm(D_ete)
```

```
Lete_1<-D_norm(norm_ete)

div<-(1/n)*(Lete_1/Lete_0)

alpha_test<-0.05

K_alpha<-(1/sqrt(n))*(1/qnorm(1-alpha_test,mu_ete,1/n))
print(K_alpha)
```

```
## [1] -0.007264243
```

```
alpha<-pnorm(K_alpha,mu_ete,1/n)
print(alpha)
```

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

```
beta<-1-pnorm(K_alpha,mu_ete,1/n)
print(beta)
```

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

Cas2, Di hiver

```
str(D_hiver)
```

```
##  int [1:463] -28 -11 -20 -26 -14 -27 -19 -33 -6 -7 ...
```

```
n<-463
mu_hiver<-mean(D_hiver)
sig_hiver<-sd(D_hiver)
norm_hiver=dnorm(-60:10, mu_hiver,sig_hiver)

Lhiver_0<-D_norm(D_hiver)
Lhiver_1<-D_norm(norm_hiver)

div<-(1/n)*(Lhiver_1/Lhiver_0)

alpha_test<-0.05

K_alpha<-(1/sqrt(n))*(1/qnorm(1-alpha_test,mu_hiver,1/n))
print(K_alpha)
```

```
## [1] -0.00251479
```

```
alpha<-pnorm(K_alpha,mu_hiver,1/n)
print(alpha)
```

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

```
beta<-1-pnorm(K_alpha,mu_hiver,1/n)
print(beta)
```

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

II) Supposons d'abord que les $D_i \sim \ll og(N(\mu, \sigma^2))$

Les valeurs des Di en été comme en hiver prennent souvent des valeurs négatives, on ne peut donc pas considérer Di comme suivant une loi log-normale directement.

```
D_ete_log<-log(abs(D_ete))
str(abs(D_ete_log))
```

```
##  num [1:491] 1.1 2.56 2.89 1.61 2.4 ...
```

```
n<-491
mu_ete_log<-mean(D_ete_log)
sig_ete_log<-sd(D_ete_log)
norm_ete_log=dnorm(0:70, mu_ete_log,sig_ete_log)

#Cf la fonction définie en question 1
Lete_0<-Dl_norm(D_ete_log)
```

```
## Warning in log(l): production de NaN

## Warning in log(l): production de NaN

## Warning in log(l): production de NaN

## Warning in log(l): production de NaN

## Warning in log(l): production de NaN

## Warning in log(l): production de NaN

## Warning in log(l): production de NaN
```

```
Lete_1<-Dl_norm(norm_ete_log)

div<-(1/n)*(Lete_1/Lete_0)

alpha_test<-0.05

K_alpha<-(1/sqrt(n))*(1/qlnorm(1-alpha_test,mu_ete_log,1/n))
print(K_alpha)
```

```
## [1] Inf
```

```
alpha<-pnorm(K_alpha,mu_ete_log,1/n)
print(alpha)
```

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

```
beta<-1-pnorm(K_alpha,mu_ete_log,1/n)
print(beta)
```

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

On ne peut donc pas appliquer le modèle de la loi log normale. Pour la loi normale , on constate des valeurs $K_{alpha,été} \approx -0.007$ et $K_{alpha,hiver} \approx -0.002$ proches de 0 notamment en hiver

Question 7

```
nonParamBootstrap <- function(echantillon) {
  M <- 100
  opt <- optim(c(2,1), Dl_norm, x = echantillon, method = "L-BFGS-B", lower=c(-100,10^-4))
  mus <- c()
  for (i in 1:M) {
    xsim <- sample(echantillon, size = 10, replace=TRUE)
    opt <- optim(c(2,1), nlogL_Norm, x = xsim, method = "L-BFGS-B", lower=c(-100,10^-4))
    mus <- append(mus, max(opt$par[1],0))
  }
  inf <- quantile(mus, 0.025)
  sup <- quantile(mus, 0.975)

  return (list(inf=inf,sup=sup))
}
```