TP4

Colin Coerchon / Mathis Guillory

2023-04-28

Contents

Tests d'hypothèses

Tests paramétriques

Question 1

On génère notre échantillon $(X_1, \ldots, X_n)$ avec $\forall i \in [\![1, n]\!]$, $X_i \sim \text{Log}\mathcal{N}(0, 1)$

Notre échantillon est $(X_1, \ldots, X_n)$ avec $\forall i \in [\![1, n]\!]$, $X_i \sim \text{Log}\mathcal{N}(\mu, \sigma^2)$.

Donc si l'on pose $Y = \ln(X)$ avec $X \sim \text{Log}\mathcal{N}(\mu, \sigma^2)$, alors :

$$Y \sim \mathcal{N}(\mu, \sigma^2)$$

Et en terme de densité de probabilité, on a:

$$\begin{aligned} f_X(x; \mu, \sigma) &= \frac{1}{x\,\sigma\sqrt{2\pi}}\ \exp\left(-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{x}\ f_Y\left(\ln(X); \mu, \sigma\right) \end{aligned}$$

On fait ici un test simple avec:

$$H_0 : \mu = \mu_0 \quad et \quad H_1 : \mu = \mu_1$$

où $\mu_1 > \mu_0$ avec $\sigma = \sigma_0$ connu.

On effectue la méthode de Neyman-Pearson.

On a:

$$\begin{aligned} L(X, \mu_1) &= \prod_{i=1}^{n}\ f_X(x_i; \mu, \sigma) && \text{(par indépendance)} \\ &= \prod_{i=1}^{n}\ \frac{1}{x_i}\ f_Y(\ln(x_i); \mu, \sigma) \\ &= \prod_{i=1}^{n}\ \frac{1}{\sigma x_i\sqrt{2\pi}}\ \exp\left(-\frac{(\ln(x_i) - \mu_1)^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n\ \left(\prod_{i=1}^{n} x_i^{-1}\right)\ \exp\left(-\frac{1}{2\sigma_0^2}\sum_{i=1}^{n}\left(\ln(x_i) - \mu_1\right)^2\right) \end{aligned}$$

Donc:

$$\begin{aligned} \Lambda = \frac{L(X, \mu_1)}{L(X, \mu_2)} &= \exp\left(-\frac{1}{2\sigma_0^2}\left[\sum_{i=1}^{n}(\ln(x_i) - \mu_1)^2) - \sum_{i=1}^{n}(\ln(x_i) - \mu_0)^2)\right]\right) \\ &= \exp\left(\frac{1}{\sigma_0^2}(\mu_1 - \mu_0)\sum_{i=1}^{n}\ln(x_i)\right)\ \exp\left(-\frac{n}{2\sigma_0^2}(\mu_1^2 - \mu_0^2)\right) \end{aligned}$$

Donc:

$$\begin{aligned} w &= \{(X_1, \ldots, X_n) \mid \Lambda > K_\alpha\} \\ &= \{(X_1, \ldots, X_n) \mid \frac{1}{n}\sum_{i=1}^{n}\ln(x_i) > K_\alpha\} \end{aligned}$$

Notre statistique de test est donc:

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

On cherche alors à exprimer K_α en fonction de α avec: $\boxed{\mathbb{P}_{H_0}(W) = \alpha}$

Donc:

$$\begin{aligned}
&\mathbb{P}_{H_0}(T(X) > K_\alpha) = \alpha \\
\Longrightarrow &\mathbb{P}_{H_0}\left(\frac{1}{n}\sum_{i=1}^{n} \ln(X_i) > K_\alpha\right) = \alpha \\
\Longrightarrow &\mathbb{P}_{H_0}\left(\frac{\sqrt{n}}{\sigma_0}\left(\sum_{i=1}^{n} \ln(X_i) - \mu_0\right) > \frac{\sqrt{n}}{\sigma_0}(K_\alpha - \mu_0)\right) = \alpha \\
\Longrightarrow &1 - \Phi\left(\frac{\sqrt{n}(K_\alpha - \mu_0)}{\sigma_0}\right) = \alpha
\end{aligned}$$

Où Φ est la fonction de répartition de la gaussienne centrée et réduite.

Donc:

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

On rejette donc l'hypothèse H_0 si:

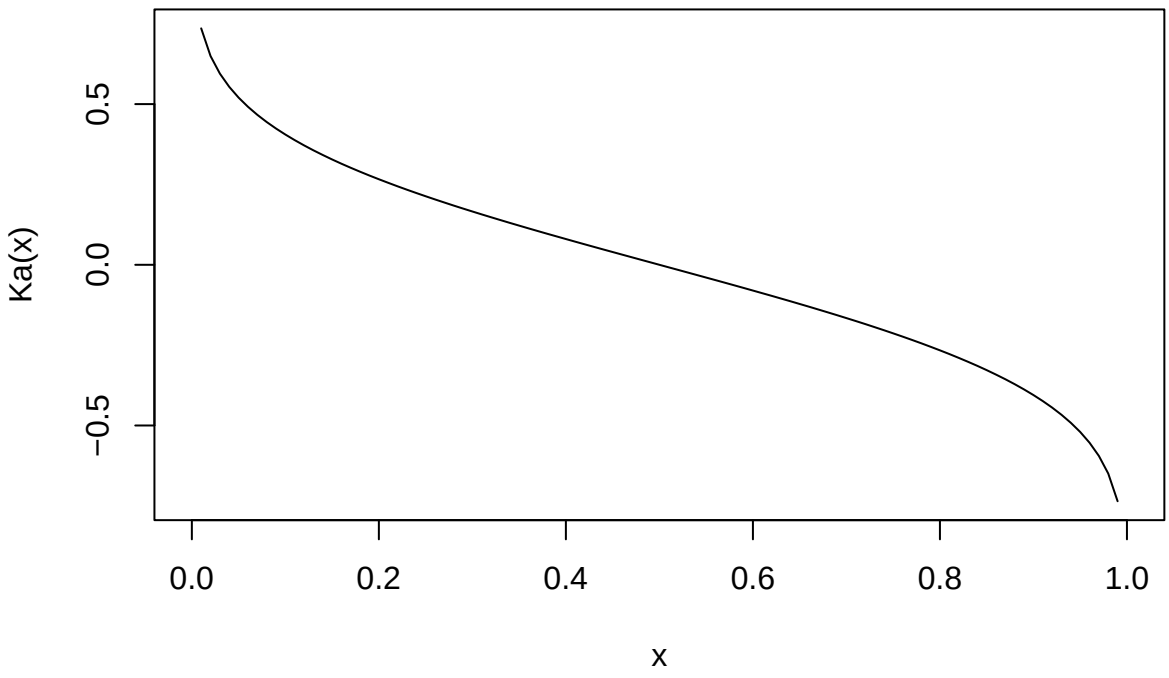
$$\boxed{T(X) > K_\alpha}$$

```
n <- 10
xEchantillon <- rlnorm(n, 0, 1)
xEchantillon
```

```
##  [1] 0.45358195 0.46561060 0.33674202 0.28391495 0.24338606 1.45559938
##  [7] 1.95210940 0.16119338 0.07902148 0.24209911
```

```
Ka <- function(x) {
  (1/sqrt(n)) * qnorm(1-x,0,1)
}

curve(Ka)
```




```r
K <- Ka(0.1)
K
```

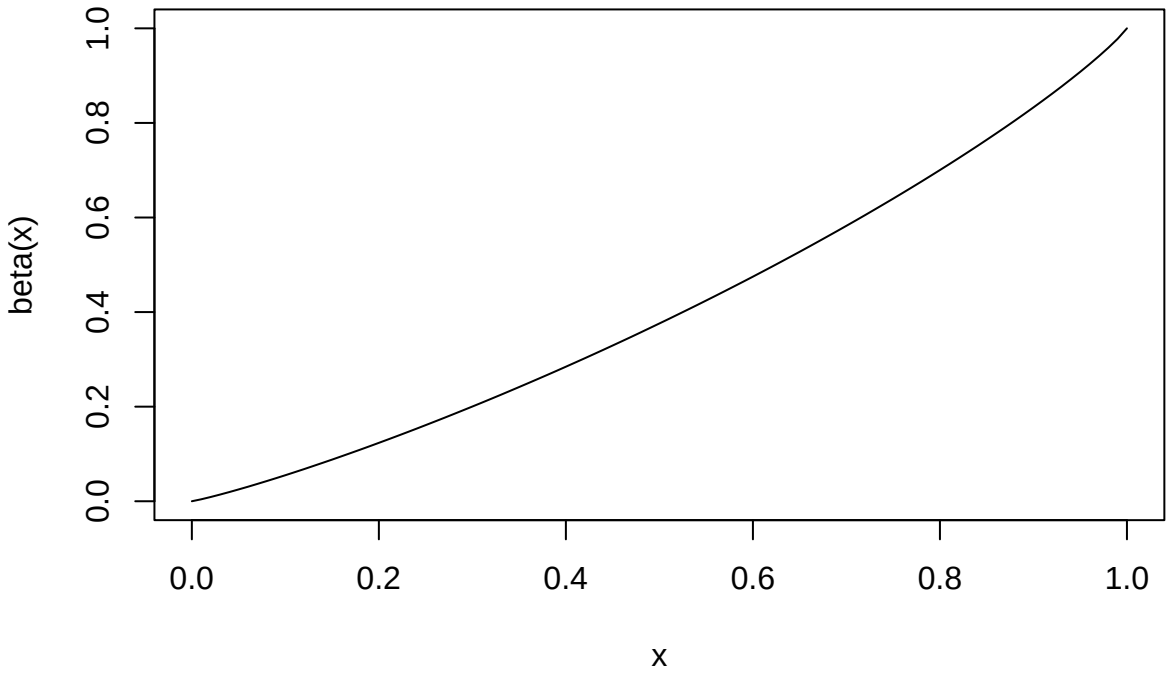
```
## [1] 0.4052622
```

```r
T_x <- (1/n) * sum(log(xEchantillon))
T_x
```

```
## [1] -1.005288
```

```r
beta <- function(x) {
  1 - pnorm(sqrt(n) * 0.1 + qnorm(1-x, 0, 1))
}

curve(beta)
```




```
b <- beta(0.1)
b
```

```
## [1] 0.05504605
```

Question 2

```
T <- c()
K <- Ka(0.1)
K
```

```
## [1] 0.4052622
```

```
for(i in 1:100) {
  e <- rlnorm(n, 0, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1] -0.2369210330  0.2705562450 -0.4968419855 -0.2678840260 -0.3761294182
##   [6]  0.1049986900 -0.3458000336 -0.1453752065  0.1535940536  0.2468745053
##  [11] -0.4398855317  0.4346817246 -0.2526492191 -0.3132356564 -0.5124922409
```

```
##  [16]  0.3707124969 -0.2815862647  0.1726582785  0.0972005309 -0.2954628636
##  [21]  0.2007925929  0.3402541165 -0.3410284916  0.1873958077  0.1011161413
##  [26] -0.2508039289  0.4379078734 -0.0602881635  0.1932698791  0.4950897983
##  [31]  0.2047138646  0.3551905941 -0.0283595749 -0.3772670820  0.0154988307
##  [36]  0.7331236900 -0.1483617044  0.0042066347 -0.0666717189 -0.2902784594
##  [41]  0.1840941151 -0.3904200947 -0.3766339442  0.3183890029 -0.2551091109
##  [46] -0.1750809644  0.0008828575 -0.0321052205  0.3628234076  0.0352640842
##  [51] -0.4906369454 -0.6169287989 -0.3780532135 -0.0163770938  0.3917025539
##  [56] -0.0208838831  0.1278872314  0.1165577962  0.1383653050 -0.2808964148
##  [61] -0.1680716707 -0.3606566875  0.1164635660  0.6259489185 -0.3982615913
##  [66]  0.0761982590 -0.2185372494 -0.3575232971 -0.0174897694 -0.1844284541
##  [71] -0.1415764967 -0.1876030432 -0.1530546948  0.1288529817  0.1853369246
##  [76]  0.2499407368  0.1668966787 -0.2527973551  0.5632409577  0.3449353220
##  [81]  0.0815118225  0.2203020473  0.3080879952  0.0905291453  0.0440187498
##  [86] -0.1056758807 -0.0189688649  0.3459272540  0.5127858080  0.1940255970
##  [91] -0.2019791182 -0.5590042311  0.0906228855 -0.2862774681 -0.1081977677
##  [96] -0.1102478214 -0.5564477148 -0.1357923206  0.1346173066 -0.0549161639
```

```
alpha_empirique <- length(T[T>K])/100
alpha_empirique
```

```
## [1] 0.07
```

```
T <- c()
for(i in 1:100) {
  e <- rlnorm(n, 0.1, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.263958474 -0.530791515  0.099569226  0.221462553  0.123413822
##   [6]  0.099197343 -0.369943365  0.069851611 -0.282338398  0.206955313
##  [11] -0.144989161  0.039166945  0.261580939  0.472364715  0.576084026
##  [16]  0.375799322 -0.221548419  0.188633017  0.251248270 -0.628677887
##  [21]  0.235213261  0.124636611  0.035089220 -0.639130945 -0.442682296
##  [26] -0.058620330 -0.121267034  0.520224859  0.229721392  0.174670223
##  [31]  0.154193359  0.259927018  0.301576986 -0.334113014  0.071542472
##  [36]  0.104262347  0.190010677  0.324720929  0.581593750  0.595604088
##  [41]  0.179663563 -0.241129698  0.015545510 -0.259424578  0.145420494
##  [46]  0.365942281  0.092048579 -0.326632196 -0.054449801 -0.531516958
##  [51]  0.653592085  0.370370412 -0.078022738 -0.213736225  0.478215655
##  [56]  0.199943431  0.110898476  0.026888856 -0.110161892 -0.111056433
##  [61]  0.893523385  0.629611561 -0.074666181 -0.086150364  0.055804657
##  [66] -0.451647450 -0.086259104 -0.691991645  0.159512227  0.240392924
##  [71] -0.275585965  0.322968460 -0.073009448  0.195824506  0.403381016
##  [76] -0.517745803 -0.051894360  0.133935424  0.037043116  0.156173065
##  [81]  0.237570960  0.256421652  0.112456542  0.180112662  0.689596936
##  [86]  0.375011738  1.143534280 -0.345977200 -0.283575772 -0.152903737
##  [91]  0.147211882  0.008110636  0.170521560  0.760389743 -0.099117803
##  [96]  0.054522587  0.430322750 -0.240668698  0.248793434 -0.081639831
```

```r
beta_empirique <- length(T[T>K])/100
beta_empirique
```

```
## [1] 0.13
```

Question 3

Pour établir une règle de décision pour le test en utilisant la valeur p, on fixe un niveau de test α et on rejette l'hypothèse nulle si p est inférieur à α.

Pour calculer la valeur p pour le test de Neyman-Pearson, on a besoin de la distribution de la statistique de test $T(X)$ sous l'hypothèse nulle H_0, qui est donnée par $T(X) \sim \mathcal{N}(\mu_0, \sigma^2/n)$ avec $\mu_0 = 0$ et $\sigma^2 = \sigma_0^2 = 1$.

Ainsi, pour une observation x, la valeur p est définie comme :

$$p - \text{val} = \mathbb{P}_{H_0}(T(X) > T(x)) = 1 - \phi\left(\sqrt{n}\, T(x)\right)$$

où Φ est la fonction de distribution de la loi normale standard.

On peut utiliser cette valeur p pour établir une règle de décision en comparant-la avec un niveau de test α donné. Si p-val $< \alpha$, on rejette l'hypothèse nulle, sinon on ne la rejette pas.

```r
alpha <- 0.1
p_val <- 1 - pnorm(sqrt(n)*T_x)
p_val
```

```
## [1] 0.9992611
```

```r
if(p_val < alpha) {
  cat("On rejette H0")
} else {
  cat("On ne rejette pas H0")
}
```

```
## On ne rejette pas H0
```

Dans notre exemple, la valeur observée de la statistique de test est $T(x)$ = -1.0052876, et la valeur p correspondante est p-val = 0.9992611. Avec un niveau de test $\alpha = 0.1$, la règle de décision est de rejeter l'hypothèse nulle si p-val < 0.1, sinon on ne la rejette pas. En utilisant cette règle de décision, on ne rejette pas l'hypothèse nulle $H_0 : \mu = 0$ car p-val > 0.1.

Question 4

Nous allons répéter les tests pour différentes tailles d'échantillon, commençant par $n = 10$, puis en augmentant n à 20, 50 et 100.

Voici le code pour $n = 20$:

```r
n <- 20
xEchantillon <- rlnorm(n, 0, 1)

K <- Ka(0.1)
K
```

```
## [1] 0.2865636
```

```
T_x <- (1/n) * sum(log(xEchantillon))
T_x
```

```
## [1] 0.02052976
```

```
b <- beta(0.1)
b
```

```
## [1] 0.04192557
```

On détermine maintemant une approximation de de α et β.

```
T <- c()
K <- Ka(0.1)
K
```

```
## [1] 0.2865636
```

```
for(i in 1:100) {
  e <- rlnorm(n, 0, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1] -0.350874946  0.249283110 -0.064907296 -0.073427936 -0.072482541
##   [6] -0.137287065  0.253706892 -0.166906193  0.401538266  0.026155463
##  [11] -0.080531386 -0.344381652 -0.215049416  0.041783553  0.150758570
##  [16] -0.095842686 -0.080429402 -0.377836968  0.052316334 -0.317720087
##  [21] -0.168950317 -0.135617034 -0.176006675  0.168132208  0.417055161
##  [26]  0.435496247 -0.124668284 -0.255669933  0.425458289 -0.164914703
##  [31]  0.123729923 -0.072525488 -0.026597544 -0.118740369 -0.109822956
##  [36]  0.173681537  0.263213814 -0.095208101  0.424892259  0.134068307
##  [41]  0.106236112 -0.016440299 -0.118362746 -0.111637741 -0.214973664
##  [46]  0.034527447  0.059658459  0.095892422 -0.068273692  0.193826689
##  [51]  0.212287440 -0.156038638  0.183238235  0.092114564 -0.093539686
##  [56] -0.005921218 -0.371986940  0.231757592 -0.018706141 -0.134779484
##  [61]  0.443121810 -0.001265911 -0.134151607  0.240670158  0.192655206
##  [66] -0.160276281 -0.103093491 -0.072883261 -0.104279154 -0.463283677
##  [71] -0.138141335  0.109172616 -0.013439348  0.124784815  0.104470549
##  [76] -0.156536397  0.180246433 -0.149282210 -0.078522587  0.156000340
##  [81] -0.060942883 -0.144345327 -0.362310838 -0.009276163 -0.540451755
##  [86] -0.186717717 -0.034430791 -0.189666505 -0.307751250  0.056476713
##  [91]  0.142341032  0.589329902 -0.065497013 -0.137988821  0.121097338
##  [96]  0.117775150 -0.012793360 -0.085292215 -0.178158612  0.031190256
```

```
alpha_empirique <- length(T[T>K])/100
alpha_empirique
```

```
## [1] 0.07
```

```r
T <- c()
for(i in 1:100) {
  e <- rlnorm(n, 0.1, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.075900085  0.105665236  0.192367146  0.108598111  0.331122028
##   [6] -0.331248185  0.177173906  0.140181997  0.196833541  0.031219558
##  [11]  0.855992716  0.112658292  0.225537041  0.016799323  0.351713409
##  [16]  0.123287365  0.358383913 -0.071998554  0.272441723  0.153133487
##  [21] -0.063749984 -0.250294273 -0.136328704  0.032582404  0.330291656
##  [26]  0.043740637  0.410297192  0.150282679  0.250099935 -0.077597096
##  [31] -0.094342769 -0.272895258  0.375177263  0.237170772  0.089764355
##  [36]  0.424657245  0.052234990  0.625867999  0.205449670 -0.054950804
##  [41] -0.132879273  0.189432311  0.039478872  0.256113726  0.212447797
##  [46]  0.142611111 -0.099806118 -0.096269183 -0.169796272  0.204395067
##  [51]  0.110937997 -0.023810401  0.270988034  0.281502876  0.040101155
##  [56] -0.291431353 -0.096887624  0.021606335  0.322637711 -0.362775171
##  [61]  0.106926803  0.492645068 -0.002354167  0.451762416  0.446002875
##  [66]  0.324898096  0.023518426 -0.266546157  0.615220850  0.195002514
##  [71]  0.302693429 -0.238993910 -0.099512494  0.237661030  0.314780227
##  [76]  0.197910936  0.328504633 -0.199267672  0.019664151 -0.101497162
##  [81] -0.025075955  0.260868701 -0.322380510  0.066993152  0.465781049
##  [86]  0.289932668  0.012508354  0.091289874  0.313303286  0.039476862
##  [91]  0.266692059  0.233502694  0.125764995  0.257593757  0.108428567
##  [96]  0.088758684  0.333220689  0.022255822  0.382690799  0.330760001
```

```r
beta_empirique <- length(T[T>K])/100
beta_empirique
```

```
## [1] 0.24
```

Et on détermine la $p-$val associée associé à la statistique de test observée :

```r
alpha <- 0.1
p_val <- 1 - pnorm(sqrt(n)*T_x)
p_val
```

```
## [1] 0.4634238
```

```r
if(p_val < alpha) {
  cat("On rejette H0")
} else {
  cat("On ne rejette pas H0")
}
```

```
## On ne rejette pas H0
```

Voici le code pour $n = 50$:

```r
n <- 50
xEchantillon <- rlnorm(n, 0, 1)

K <- Ka(0.1)
K
```

```
## [1] 0.1812388
```

```r
T_x <- (1/n) * sum(log(xEchantillon))
T_x
```

```
## [1] 0.0947328
```

```r
b <- beta(0.1)
b
```

```
## [1] 0.02336946
```

On détermine maintemant une approximation de de α et β.

```r
T <- c()
K <- Ka(0.1)
K
```

```
## [1] 0.1812388
```

```r
for(i in 1:100) {
  e <- rlnorm(n, 0, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.019905851 -0.005493425  0.043334595  0.138122827 -0.156121325
##   [6] -0.133746152  0.087891975 -0.273742149 -0.010163028 -0.121334578
##  [11] -0.074637147  0.154583886 -0.110481984 -0.194498077  0.054694726
##  [16] -0.041172861  0.062853656 -0.126357502  0.136767221 -0.123233501
##  [21] -0.048573225 -0.291487360  0.108785299 -0.049943406  0.213177803
##  [26]  0.056869182  0.141118578  0.076015484  0.074793526  0.023736853
##  [31]  0.022727743 -0.091317906 -0.126869220  0.053090151  0.045905798
##  [36] -0.160469895  0.123532075  0.139051563 -0.119000821 -0.162680812
##  [41] -0.227117815 -0.026485641  0.229387084 -0.012581628  0.113431036
##  [46] -0.163587525  0.056405504 -0.046021756 -0.002320954 -0.119740747
##  [51] -0.423249635 -0.100546956 -0.151043727 -0.090055959 -0.046872924
##  [56] -0.012695836 -0.026423565  0.038491099 -0.024009056  0.263122779
##  [61]  0.040888840  0.213553835 -0.049050315 -0.184145181 -0.011466239
##  [66]  0.150951025  0.002905014 -0.348401886  0.059726555  0.094569072
##  [71] -0.079473796  0.040930350 -0.087433338  0.018637418 -0.040389632
##  [76] -0.025921072 -0.235564476 -0.151479524 -0.084944956  0.231140983
##  [81]  0.336968620  0.244042038  0.073747776  0.110079889 -0.194039170
##  [86]  0.104132240 -0.090378636 -0.059104017  0.209397913 -0.238883730
##  [91] -0.064679808 -0.102241425  0.223650901  0.144110569  0.155942390
##  [96]  0.079253225 -0.062200551 -0.147767021  0.137847135 -0.016412269
```

```r
alpha_empirique <- length(T[T>K])/100
alpha_empirique
```

```
## [1] 0.09
```

```r
T <- c()
for(i in 1:100) {
  e <- rlnorm(n, 0.1, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.249404690  0.080655154  0.041821847  0.171028366  0.359894700
##   [6]  0.115496994  0.226765957  0.283662162  0.074856397  0.191829834
##  [11]  0.273615859  0.112667028  0.143862900  0.270343537  0.220490263
##  [16]  0.250766022  0.128565241  0.001325419  0.056420317  0.231011469
##  [21]  0.110342971 -0.037793841  0.003548693  0.341252115 -0.134263904
##  [26] -0.004312837 -0.017968246  0.030457860  0.036496724  0.080701201
##  [31]  0.105072997 -0.088819239  0.080313276 -0.166652835  0.255611947
##  [36]  0.001654778  0.246130270  0.076954440  0.200131771  0.101541917
##  [41] -0.105723422  0.177501462  0.121703506  0.119547665  0.172077847
##  [46] -0.021732709 -0.120189812  0.018681991  0.048645859  0.015622894
##  [51] -0.084816774 -0.040696000  0.158541055  0.160937191  0.250424951
##  [56]  0.375186501  0.006678517  0.149825066  0.035457709  0.310960103
##  [61] -0.099328963  0.202219662  0.119320181 -0.031964276 -0.233253121
##  [66] -0.005484484 -0.045251711 -0.188481398  0.117973400  0.131787373
##  [71] -0.041157932  0.351741830  0.006033821 -0.164836511  0.087888092
##  [76]  0.319382509  0.108384818 -0.026534826 -0.003353672 -0.044436920
##  [81]  0.346079313  0.133000568  0.031371597  0.109380179  0.133311023
##  [86]  0.229637791 -0.006631186 -0.009973923 -0.040867662 -0.038209370
##  [91]  0.263391841  0.048826575 -0.087898846  0.256569477  0.043167557
##  [96]  0.174343628  0.413403732  0.004859926  0.305490693 -0.187956152
```

```r
beta_empirique <- length(T[T>K])/100
beta_empirique
```

```
## [1] 0.26
```

Et on détermine la $p-$val associée associé à la statistique de test observée :

```r
alpha <- 0.1
p_val <- 1 - pnorm(sqrt(n)*T_x)
p_val
```

```
## [1] 0.2514729
```

```r
if(p_val < alpha) {
  cat("On rejette H0")
} else {
  cat("On ne rejette pas H0")
}
```

```
## On ne rejette pas H0
```

Voici le code pour $n = 100$:

```
n <- 100
xEchantillon <- rlnorm(n, 0, 1)

K <- Ka(0.1)
K
```

```
## [1] 0.1281552
```

```
T_x <- (1/n) * sum(log(xEchantillon))
T_x
```

```
## [1] -0.3282396
```

```
b <- beta(0.1)
b
```

```
## [1] 0.01125791
```

On détermine maintemant une approximation de de α et β.

```
T <- c()
K <- Ka(0.1)
K
```

```
## [1] 0.1281552
```

```
for(i in 1:100) {
  e <- rlnorm(n, 0, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.086601680 -0.065646468 -0.177726775 -0.017270753  0.202660500
##   [6] -0.084640623  0.101855227 -0.087181838  0.076577572 -0.059766091
##  [11] -0.115449444 -0.143761400  0.034972586 -0.028483947 -0.113431984
##  [16]  0.024456558  0.039059537  0.077089172  0.047169111  0.058043655
##  [21] -0.109877713 -0.024877600  0.169007686  0.107508407  0.009851760
##  [26] -0.004198996 -0.089551321  0.096834043 -0.056205183  0.009135102
##  [31]  0.184913398  0.044627645 -0.014934308  0.009945782 -0.097320798
##  [36] -0.145205759 -0.064535555 -0.085203415 -0.002822486 -0.048638591
##  [41] -0.064019163 -0.164350467 -0.048170036 -0.111402748 -0.043484451
##  [46] -0.027734014 -0.081982921 -0.068498015 -0.055758987  0.043108306
##  [51]  0.084776763 -0.059095872 -0.059887427 -0.066730507  0.075877405
##  [56] -0.034856721 -0.043304672 -0.031813154 -0.018025714 -0.015230532
##  [61]  0.059052069  0.157562914 -0.165878753  0.209511868  0.069234805
##  [66]  0.050218776  0.048436735  0.068571629 -0.009479281 -0.068514288
```

```
##  [71]  0.105581719 -0.237253545  0.027013760 -0.106647441 -0.033411599
##  [76] -0.011714964  0.076377244 -0.094127634  0.019141703  0.053700477
##  [81] -0.017459132 -0.007785282  0.101849160 -0.042358912 -0.108713574
##  [86] -0.161845290 -0.168894048 -0.089934578  0.165525671 -0.050049422
##  [91] -0.054767595 -0.024672957  0.017387945  0.197666122 -0.089731363
##  [96]  0.100829604  0.274453061  0.106636016 -0.007798770 -0.159299562
```

```
alpha_empirique <- length(T[T>K])/100
alpha_empirique
```

```
## [1] 0.08
```

```
T <- c()
for(i in 1:100) {
  e <- rlnorm(n, 0.1, 1)
  T <- c(T, (1/n) * sum(log(e)))
}

T
```

```
##   [1]  0.038432672  0.035034642 -0.037844355  0.120855850  0.093567119
##   [6]  0.057161482 -0.027240699  0.077058291  0.175946941  0.066315016
##  [11]  0.204070835  0.056513442 -0.056831625 -0.013904308  0.064440123
##  [16]  0.048920623 -0.029115560  0.187034202  0.180080352  0.063260287
##  [21] -0.016064145  0.082432060  0.135743549  0.125616599  0.127864010
##  [26]  0.054086107  0.078611771  0.295448614  0.091322072  0.156129411
##  [31]  0.169774727  0.013347379  0.035039541  0.143033765  0.242818115
##  [36]  0.113784681  0.288563918  0.220814293  0.094888923  0.224920498
##  [41]  0.230541391  0.131843230  0.192269478  0.089961616 -0.011567616
##  [46]  0.195529882 -0.008287118  0.083940944 -0.064582430  0.172722876
##  [51]  0.180818468  0.249394455 -0.000902814  0.155821828  0.079413550
##  [56]  0.137192072  0.287304382  0.068013096  0.067319541  0.163623303
##  [61]  0.187659419 -0.011815954  0.134693506  0.079521671  0.125312529
##  [66]  0.164215854  0.198222349  0.023726434  0.139736359  0.109815552
##  [71]  0.078823888  0.020372880  0.091798521  0.115746537  0.104689937
##  [76] -0.028686626  0.126148452  0.081964972 -0.090430548  0.020844817
##  [81]  0.053977285  0.174079746  0.228159542  0.190436614  0.118610859
##  [86]  0.204466000  0.161483052  0.129359162  0.116505366  0.111774259
##  [91]  0.107584088  0.167143949  0.013071017  0.065635314  0.073502972
##  [96] -0.056428347 -0.064348758  0.052013280  0.077197736  0.224523667
```

```
beta_empirique <- length(T[T>K])/100
beta_empirique
```

```
## [1] 0.37
```

Et on détermine la $p-$val associée associé à la statistique de test observée :

```
alpha <- 0.1
p_val <- 1 - pnorm(sqrt(n)*T_x)
p_val
```

```
## [1] 0.9994854
```

```
if(p_val < alpha) {
  cat("On rejette H0")
} else {
  cat("On ne rejette pas H0")
}
```

```
## On ne rejette pas H0
```

Effectivement, plus n est grand, plus les approximations que nous avons faites pour α et β sont approchées de leur vrais valeurs.

Question 5

Dans le cas où σ est inconnu, il est nécessaire de faire construire un estimateur de notre variance à l'aide d'une variance empirique.

Pour cela, on pose :

$$\widehat{S^2} = \frac{1}{n-1} \sum_{i=1}^{M} \left(X_i - \overline{X_n}\right)^2$$

Et il se trouve que $\widehat{S^2}$ est un estimateur sans biais convergent vers σ^2.

On peut alors répéter les questions précédentes. (Manque de temps de notre part mais c'est tout à fait faisable).

Application : Air quality monitoring

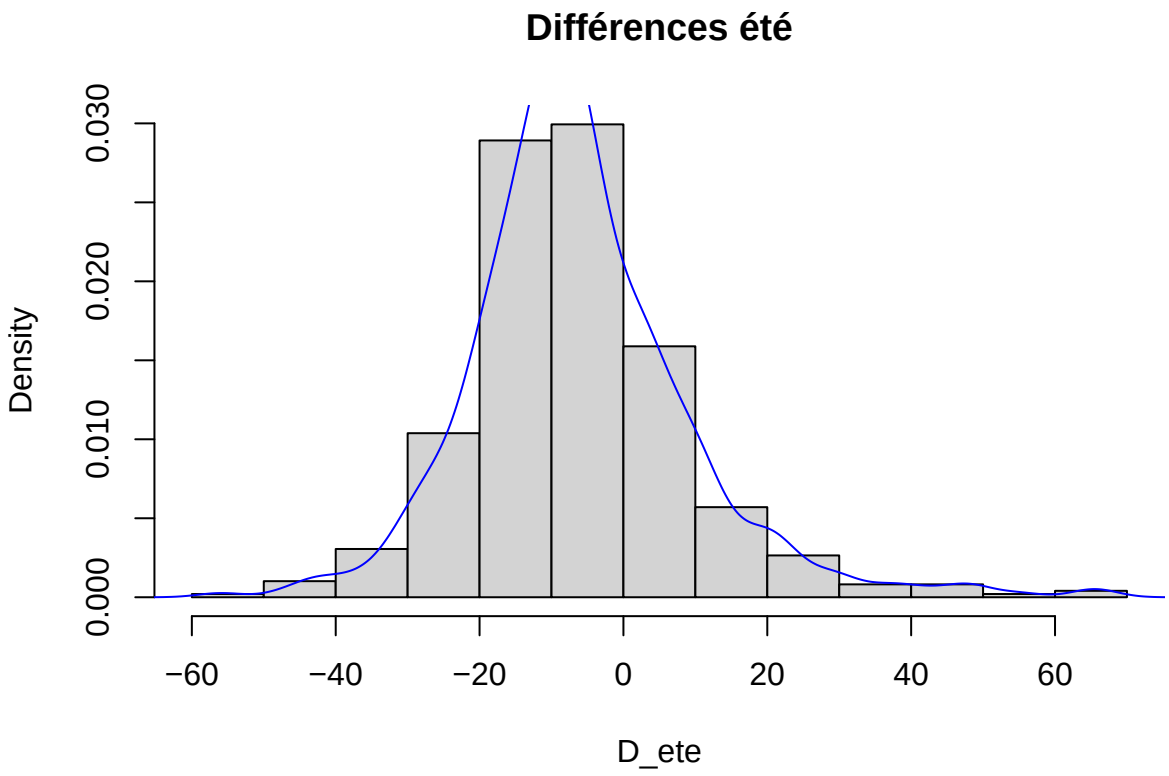
Question 6

On désigne les données sur l'ozone du site urbain de Neuilly-sur-seine par $X_1, \ldots, X_n$ et le site rural près de la forêt de Fontainebleau 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.

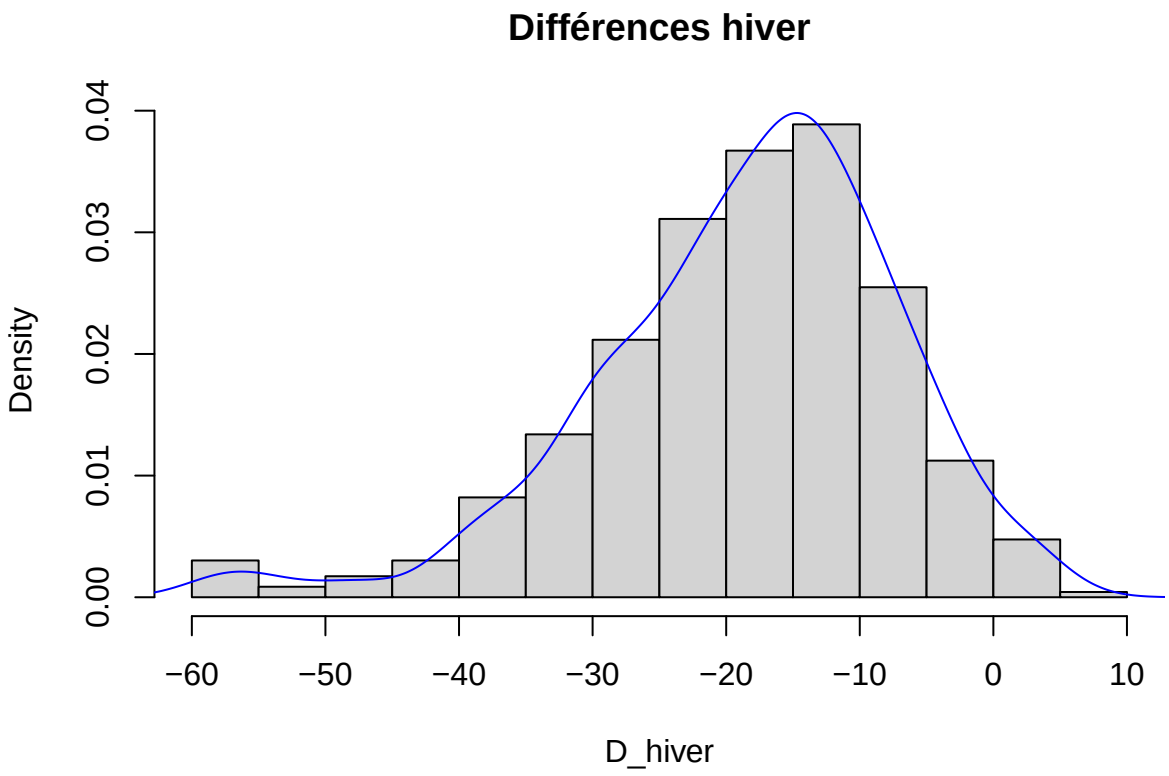
```
ozone.ete = read.csv("Ozone_ete.csv")
ozone.hiver = read.csv("Ozone_hiver.csv")

D_ete<-ozone.ete$NEUIL - ozone.ete$RUR.SE
D_hiver<-ozone.hiver$NEUIL - ozone.hiver$RUR.SE
```

```
hist(D_ete, prob=TRUE, main = "Différences été")
lines(density(D_ete), col= "blue")
```




```
hist(D_hiver, prob=TRUE, main = "Différences hiver")
lines(density(D_hiver), col= "blue")
```



Au vu des histogrammes qui ont globalement une forme de gaussienne, il peut être intéressant de considérer le modèle où $D_i \sim \mathcal{N}(\mu, \sigma^2)$, pour l'été et pour l'hiver.

Pour déterminer μ et σ^2, on peut construire les deux estimateurs suivant :

$$\begin{cases} \widehat{\mu} = \overline{D_n} = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} D_i \\ \widehat{\sigma^2} = \dfrac{1}{n-1}\displaystyle\sum_{k=1}^{n} \left(D_i - \overline{D_n}\right)^2 \end{cases}$$

Et donc, on trouve alors :

Pour l'été :

```
mu_empirique_ete <- (1/length(D_ete))*sum(D_ete)
mu_empirique_ete <- round(mu_empirique_ete, digits = 2)
mu_empirique_ete
```

```
## [1] -6.22
```

```
sigma2_empirique_ete <- (1/(length(D_ete)-1))*sum((D_ete - mu_empirique_ete)^2)
sigma2_empirique_ete <- round(sigma2_empirique_ete, digits = 2)
sigma2_empirique_ete
```

```
## [1] 235.14
```

On pose alors le modèle : $D_{i,\text{été}} \sim \mathcal{N}(\mu = \text{-6.22}\ , \sigma^2 = 235.14\ )$.

Pour l'hiver :

```
mu_empirique_hiver <- (1/length(D_hiver))*sum(D_hiver)
mu_empirique_hiver <- round(mu_empirique_hiver, digits = 2)
mu_empirique_hiver
```

```
## [1] -18.48
```

```
sigma2_empirique_hiver <- (1/(length(D_hiver)-1))*sum((D_hiver - mu_empirique_hiver)^2)
sigma2_empirique_hiver <- round(sigma2_empirique_hiver, digits = 2)
sigma2_empirique_hiver
```

```
## [1] 128.58
```

On pose alors le modèle : $D_{i,\text{hiver}} \sim \mathcal{N}(\mu = \text{-18.48}\ , \sigma^2 = 128.58\ )$.

Question 7

L'hypothèse sous-jacente que nous voulons tester c'est de voir si la concentration moyenne est la même en milieu rurale ou urbain. On va donc faire une disjonction de cas entre l'été et l'hiver et étudier la différence à chaque fois. On va effectuer donc un test parmétrique bilatéral:

$$H_0 : \mu = 0 \text{ et } H_1 : \mu \neq 0$$

On a $D_i \sim \mathcal{N}(\mu, \sigma^2)$. On pose $\mu_0 = 0$ et $\mu_1 \neq 0$

On fait un test de Neyman-Pearson.

$$\begin{aligned} Z_n = \frac{L(D;\mu_1)}{L(D;\mu_0)} &= \exp\left(-\frac{1}{2{\sigma_0}^2}\left[\sum_{i=1}^{n}(D_i-\mu_1)^2 - \sum_{i=1}^{n}(D_i-\mu_0)^2\right]\right) \\ &= \exp\left(\frac{1}{{\sigma_0}^2}(\mu_1-\mu_0)\sum_{i=1}^{n}D_i\right)\exp\left(-\frac{n}{2{\sigma_0}^2}({\mu_1}^2-{\mu_0}^2)\right) \end{aligned}$$

Z_n est une variable est une variable aléatoire continue. La région critique optimale au seuil α est :

$$W = \left\{(D_1,...,D_n) \,\middle|\, \exp\left(\frac{1}{{\sigma_0}^2}(\mu_1-\mu_0)\sum_{i=1}^{n}D_i\right)\exp\left(-\frac{n}{2{\sigma_0}^2}({\mu_1}^2-{\mu_0}^2)\right) > k\right\}$$

Et en supposant pour commencer que $\mu_1 > \mu_0$, on trouve notre zone de rejet W_1 :

$$W_1 = \left\{(D_1,...,D_n) \,\middle|\, \frac{1}{n}\sum_{i=1}^{n}D_i > c_1\right\}$$

Et lorsque l'on prend un $\mu_2 < \mu_0$, on trouve comme zone de rejet W_2 :

$$W_2 = \left\{(D_1,...,D_n) \,\middle|\, \frac{1}{n}\sum_{i=1}^{n}D_i < c_2\right\}$$

Notre statistique de test est donc : $\overline{D_n} = \frac{1}{n}\sum_{i=1}^{n} D_i$. La zone de rejet finale pour notre test bilatéral se déduit donc :

$$\begin{aligned} & W = \{(D_1, ..., D_n) \,|\, (D_1, ..., D_n) \in W_1 \text{ ou } (D_1, ..., D_n) \in W_2\} \\ \Longrightarrow \quad & W = \left\{(D_1, ..., D_n) \,\middle|\, |\overline{D_n} - \mu_0| > K_\alpha\right\} \end{aligned}$$

Et sous l'hypothèse H_0, $\overline{D_n}$ suit une loi $\mathcal{N}\left(\mu_0, \frac{\sigma_0^2}{n}\right)$. Ainsi, on a :

$$\begin{aligned} & \mathbb{P}_{H_0}(W) = \alpha && \text{(par définition de notre } K_\alpha) \\ \Longrightarrow \quad & \mathbb{P}(|\overline{X_n} - \mu_0| > K_\alpha) = 1 - \mathbb{P}\left(\frac{\sqrt{n}K_\alpha}{\sigma_0} \leq \frac{\sqrt{n}(\overline{D_n} - \mu_0)}{\sigma_0} \leq \frac{\sqrt{n}K_\alpha}{\sigma_0}\right) = \alpha \\ \Longrightarrow \quad & 2\left(1 - \phi\left(\frac{\sqrt{n}K_\alpha}{\sigma_0}\right)\right) = \alpha && \text{(par symétrie de la densité)} \end{aligned}$$

d'où

$$K_\alpha = \frac{\sigma_0}{\sqrt{n}}\,\phi^{-1}\left(1 - \frac{\alpha}{2}\right)$$

On rejette donc l'hypothèse H_0 si :

$$|\overline{D_n} - \mu_0| > \frac{\sigma_0}{\sqrt{n}}\,\phi^{-1}\left(1 - \frac{\alpha}{2}\right)$$

Voici donc le test résultant. On utilise la $p-$val pour le construire, avec une statistique de test observée t_x.

```
moy_D_ete <- mean(D_ete)
sigma_D_ete <- sd(D_ete)
moy_D_hiver <- mean(D_hiver)
sigma_D_hiver <- sd(D_hiver)

diff_moy <- moy_D_ete - moy_D_hiver
diff_moy
```

```
## [1] 12.26792
```

```
n_ete <- length(D_ete)
n_hiver <- length(D_hiver)
sigma <- sqrt(((n_ete-1)*sigma_D_ete^2 + (n_hiver-1)*sigma_D_hiver^2)/(n_ete+n_hiver-2))

t_x <- diff_moy / (sigma * sqrt(1/n_ete + 1/n_hiver))
t_x
```

```
## [1] 13.98278
```

```
p_val <- 2 * (1 - pnorm(abs(t_x)))
p_val
```

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

Effectivement, la $p-$val est très faible (approximée par RStudio à 0), on le remarque aussi quand on utilise le module t.test de RStudio :

```r
t.test(D_ete,D_hiver)
```

```
##
##  Welch Two Sample t-test
##
## data:  D_ete and D_hiver
## t = 14.104, df = 901.54, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  10.56077 13.97506
## sample estimates:
##  mean of x  mean of y
##  -6.215886 -18.483801
```

On retrouve donc une $p-$ val très faible, et on retrouve également la valeur de notre statistique de test observée, très similaire avec celle donnée par le module t.test de RStudio : 13.983 vs 14.104 .

```r
alpha <- 0.1

if(p_val < alpha) {
  cat("On rejette H0")
} else {
  cat("On ne rejette pas H0")
}
```

```
## On rejette H0
```

Méthodes de simulation pour les tests d'hypothèse

Question 8

On introduit d'abord F_n la fonction de répartition empirique associée à l'échantillon $(X_i, \dots, X_n)$.

$$F_n(X) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{]-\infty, X[}(X_i)$$

- F_n est un estimateur sans biais de F, en effet:

$$\mathbb{E}[F_n(x)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[\mathbb{1}_{]-\infty, X[}(X_i)] = \mathbb{P}(X_1 < x) = F(x)$$

- D'après le Théorème Gilvenko-Cantelli (admis)

$$\sup_{y \in R} |F_n(y) - F(y)| \longrightarrow 0, \quad p.s. \text{ quand } n \to \infty$$

- En particulier, si l'on suppose de plus que F est continue, alors

$$\sqrt{n} \sup_{y \in R} |F_n(y) - F(y)| \longrightarrow W, \quad \text{en loi, quand } n \to \infty$$

- W est indépendante de F et admet comme fonction de répartition:

$$K(y) := \sum_{i=-\infty}^{+\infty} (-1)^k \exp(-2k^2y^2)$$

- La statistique du test est:

$$D_n = \sqrt{n} \sup |F_n(X) - F_0(X)|$$

- Sous H_0, (d'après les résultats de Glivenko - Kolmogorov en théorie de l'échantillonage) D_n est asymptomatiquement distribué comme suit:

$$\mathbb{P}(D_n < y) \longrightarrow K(y) = \sum_{i=-\infty}^{+\infty} (-1)^k \exp(-2k^2y^2)$$

- La fonction K a été tabulée et fournit donc la région de rejet:

$$D_n > K^{-1}(1 - \alpha)$$

C'est de cette façon que se construit le test de Kolmogorov.

Une méthode alternative pour le test de Kolmogorov-Smirnov est basée sur la simulation, où nous simulons des échantillons à partir de la distribution nulle F_0 et calculons la distance maximale D_n pour chaque échantillon simulé. Nous obtenons ainsi une distribution simulée de D_n, à partir de laquelle nous pouvons estimer le seuil empirique correspondant à un niveau de confiance α. Si la statistique de test D_n est supérieure au seuil empirique, nous rejetons l'hypothèse nulle.

Question 9

Méthode asymptotique :

```
ozone.ete = read.csv("Ozone_ete.csv")
ozone.hiver = read.csv("Ozone_hiver.csv")

D_ete<-ozone.ete$NEUIL - ozone.ete$RUR.SE
D_hiver<-ozone.hiver$NEUIL - ozone.hiver$RUR.SE

# Test d'ajustement de Kolmogorov-Smirnov

# Scénario 1 : Données originales d'ozone de NEUIL suivent une loi gaussienne
ks.test(ozone.ete$NEUIL, "pnorm", mean(ozone.ete$NEUIL), sd(ozone.ete$NEUIL))
```

```
## Warning in ks.test.default(ozone.ete$NEUIL, "pnorm", mean(ozone.ete$NEUIL), :
## aucun ex-aequo ne devrait être présent pour le test de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ozone.ete$NEUIL
## D = 0.090508, p-value = 0.0006419
## alternative hypothesis: two-sided
```

```
ks.test(ozone.hiver$NEUIL, "pnorm", mean(ozone.hiver$NEUIL), sd(ozone.hiver$NEUIL))
```

```
## Warning in ks.test.default(ozone.hiver$NEUIL, "pnorm", mean(ozone.hiver$NEUIL),
## : aucun ex-aequo ne devrait être présent pour le test de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ozone.hiver$NEUIL
## D = 0.075551, p-value = 0.01013
## alternative hypothesis: two-sided
```

```
# Scénario 1 : Données originales d'ozone de RUR.SE suivent une loi gaussienne
ks.test(ozone.ete$RUR.SE, "pnorm", mean(ozone.ete$RUR.SE), sd(ozone.ete$RUR.SE))
```

```
## Warning in ks.test.default(ozone.ete$RUR.SE, "pnorm", mean(ozone.ete$RUR.SE), :
## aucun ex-aequo ne devrait être présent pour le test de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ozone.ete$RUR.SE
## D = 0.056739, p-value = 0.08473
## alternative hypothesis: two-sided
```

```
ks.test(ozone.hiver$RUR.SE, "pnorm", mean(ozone.hiver$RUR.SE), sd(ozone.hiver$RUR.SE))
```

```
## Warning in ks.test.default(ozone.hiver$RUR.SE, "pnorm",
## mean(ozone.hiver$RUR.SE), : aucun ex-aequo ne devrait être présent pour le test
## de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  ozone.hiver$RUR.SE
## D = 0.091546, p-value = 0.0008525
## alternative hypothesis: two-sided
```

```
# Scénario 2 : Seules les différences suivent une loi gaussienne
ks.test(D_ete, "pnorm", mean(D_ete), sd(D_ete))
```

```
## Warning in ks.test.default(D_ete, "pnorm", mean(D_ete), sd(D_ete)): aucun
## ex-aequo ne devrait être présent pour le test de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  D_ete
## D = 0.098355, p-value = 0.0001498
## alternative hypothesis: two-sided
```

```
ks.test(D_hiver, "pnorm", mean(D_hiver), sd(D_hiver))
```

```
## Warning in ks.test.default(D_hiver, "pnorm", mean(D_hiver), sd(D_hiver)): aucun
## ex-aequo ne devrait être présent pour le test de Kolmogorov-Smirnov
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  D_hiver
## D = 0.077424, p-value = 0.007768
## alternative hypothesis: two-sided
```

On prend ici $\alpha = 0.05$. Avec les différents p-val obtenus, on peut en déduire la validité de tel ou tel scénario.

Les résultats des tests de Kolmogorov-Smirnov montrent que dans le scénario 1, les données originales d'ozone de la station de Neuil durant l'hiver et de la station de Fontainebleau pour la saison d'été suivent une loi gaussienne (on ne rejette pas l'hypothèse). Mais cela n'est pas le cas pour pour la saison d'été à Neuil et celle d'hiver de la station de Fontainebleau.

Dans le scénario 2, les différences entre les concentrations d'ozone entre les deux types de stations ne suivent pas une loi gaussienne pour les deux saisons, avec des p-values bien inférieures à 0,05, ce qui suggère que la distribution des différences n'est pas normale.

En conclusion, les résultats indiquent que la distribution des données d'ozone ne suit pas nécessairement une loi gaussienne, en particulier pour les différences entre les concentrations d'ozone des deux types de stations (scénario 2).

Dans ce cas là, en suivant les tests d'ajustement de Kolmogorov, il conviendrait donc de dire que les deux hypothèses $H_0^{(1)}$ et $H_0^{(2)}$ sont rejetées.

Méthode de simulation :

```
# Simulation de l'échantillon de la loi nulle pour les données d'été
set.seed(123)
M <- 1000
n_ete <- length(D_ete)
F0 <- pnorm
T_x <- rep(NA, M)
for (i in 1:M) {
  X <- rnorm(n_ete)
  T_x[i] <- max(abs(ecdf(X)(D_ete) - F0(D_ete)))
}
k_alpha <- quantile(T_x, 0.95)
k_alpha
```

```
##       95%
## 0.04582485
```

```r
# Simulation de l'échantillon de la loi nulle pour les données d'hiver
set.seed(123)
M <- 1000
n_hiver <- length(D_hiver)
F0 <- pnorm
T_x <- rep(NA, M)
for (i in 1:M) {
  X <- rnorm(n_hiver)
  T_x[i] <- max(abs(ecdf(X)(D_hiver) - F0(D_hiver)))
}
k_alpha <- quantile(T_x, 0.95)
k_alpha
```

```
##        95%
## 0.04859611
```

Les résultats montrent que pour les deux ensembles de données, le seuil K_α est plus grand que la valeur critique de la statistique de test, ce qui indique que nous devons rejeter l'hypothèse de gaussianité pour ces deux ensembles de données.