

# M1 SSD - UE Tests - TP1

Lancelot Ravier

2024-10-25

Dans ce TP, on s'intéresse à la problématique de l'obsolescence des appareils électroniques. Certains fabricants sont soupçonnés d'obsolescence programmée : les appareils électroniques auraient une durée de vie programmée dans un certain intervalle de temps (par exemple entre 5 et 6 ans de vie). Au contraire, en l'absence d'obsolescence programmée, on s'attend à avoir des durées de vie aléatoires, prenant des valeurs dans  $\mathbb{R}^+$ . Une modélisation classique des durées de vie est alors d'utiliser la loi de Weibull. Cet exercice a pour but d'étudier cette loi, de proposer un estimateur des paramètres ainsi qu'un intervalle de confiance et un test statistique sur ce paramètre.

1. On admet que la durée de vie est représentée par une variable aléatoire  $X$  suivant une loi de Weibull de paramètres  $\theta$ ,  $a$  et  $c$  positifs. Cette loi, notée  $W(\theta, a, c)$  a pour fonction de répartition

$$F(x) = (1 - \exp(-\frac{(x-a)^c}{\theta}))I\{x > a\},$$

et donc pour densité la fonction

$$f(x) = \frac{c}{\theta}(x-a)^{c-1} \exp(-\frac{(x-a)^c}{\theta})I\{x > a\}.$$

## 1.1. Montrer que la variable aléatoire $Y = \frac{2}{\theta}(X-a)^c$ suit la loi du $\chi^2_2$ .

Posons  $Y = \frac{2}{\theta}(X-a)^c$ . Nous voulons trouver la loi de  $Y$ . Nous allons procéder en utilisant un changement de variable. Soit  $u = (X-a)^c$ , alors  $X = a + u^{\frac{1}{c}}$ . En dérivant cette relation par rapport à  $u$ , :

$$\frac{\partial X}{\partial u} = \frac{1}{c}u^{\frac{1}{c}-1}$$

La densité de  $u$  peut être obtenue en effectuant le changement de variable suivant :

$$f_U(u) = f_X(a + u^{\frac{1}{c}}) \cdot \left| \frac{\partial X}{\partial u} \right|$$

En remplaçant  $f_X(x)$  et  $\frac{dX}{du}$  par leurs expressions, :

$$f_U(u) = \frac{c}{\theta}u^{\frac{c-1}{c}} \exp\left(-\frac{u}{\theta}\right) \cdot \frac{1}{c}u^{\frac{1}{c}-1} = \frac{1}{\theta} \exp\left(-\frac{u}{\theta}\right)$$

Ainsi,  $u$  suit une loi de densité  $f_U(u) = \frac{1}{\theta} \exp\left(-\frac{u}{\theta}\right)$ .

En considérant  $Y = \frac{2}{\theta}u$ ,  $u = \frac{\theta}{2}Y$ . En substituant  $u = \frac{\theta}{2}Y$ , la densité de  $Y$  devient :

$$f_Y(y) = f_U\left(\frac{\theta}{2}y\right) \cdot \left| \frac{\partial}{\partial y} \left(\frac{\theta}{2}y\right) \right|$$

En remplaçant, nous obtenons :

$$f_Y(y) = \frac{1}{\theta} \exp\left(-\frac{\frac{\theta}{2}y}{\theta}\right) \cdot \frac{\theta}{2} = \frac{1}{2} \exp\left(-\frac{y}{2}\right)$$

La densité de la loi de khi-deux étant la suivante :

$$f(x; k) = \frac{1}{2^{k/2} \Gamma\left(\frac{k}{2}\right)} x^{\frac{k}{2}-1} e^{-x/2}, \quad x > 0$$

La densité  $f_Y(y) = \frac{1}{2} \exp\left(-\frac{y}{2}\right)$  correspond bien à celle d'une loi du  $\chi^2$  à deux degrés de liberté.  $Y = \frac{2}{\theta}(X-a)^c$  suit donc la loi du  $\chi^2_2$ .

## 1.2. Que représente le paramètre $a$ ?

Dans le cadre de la distribution de Weibull,  $x > a$ ,  $a$  représente le paramètre de déplacement, et détermine le point de départ de la distribution sur l'axe des  $X$ , c'est à dire le minimum de la variable  $X$  respectant  $X-a > 0$ .

**1.3. Simuler  $n = 100$  variables aléatoires  $X_1, \dots, X_n$  sous la loi de Weibull de paramètres  $a = 0$ ,  $c = 1/2$  et  $\theta = 1$  avec la fonction `rweibull` (les paramètres sont `shape=c`, `scale =  $\theta^{(1/c)}$` , vous expliquerez pourquoi). Calculer les variables  $Y_1, \dots, Y_n$  correspondantes. Proposer des graphiques pour illustrer leurs distributions.**

La distribution de Weibull est souvent présentée sous deux formes de paramétrage différentes. Traditionnellement, on utilise un paramètre de forme  $c$  et un paramètre d'échelle  $\lambda$ , avec la fonction de répartition suivante :

$$F(x) = 1 - \exp\left(-\left(\frac{x}{\lambda}\right)^c\right)$$

Cependant, dans cet exercice, on utilise les paramètres  $\theta$  et  $c$  avec la fonction de répartition suivante :

$$F(x) = 1 - \exp\left(-\left(\frac{(x-a)^c}{\theta}\right)\right)$$

Pour passer de l'un à l'autre, on peut voir que  $\lambda = \theta^{1/c}$  car `rweibull` utilise le paramétrage avec  $\lambda$ . En utilisant cette transformation, les variables aléatoires générées par `rweibull` respectent la loi de weibull vec les paramètres  $\theta$  et  $c$  spécifiés dans l'exercice.

Afin de montrer l'adéquation de  $Y$  par rapport à la loi du  $\chi^2_2$ , nous devons simuler des variables  $X$  suivant une loi de Weibull, générer les variables  $y$  en fonction des variables  $X$  puis vérifier si les variables  $Y$  suivent une loi du  $\chi^2_2$ .

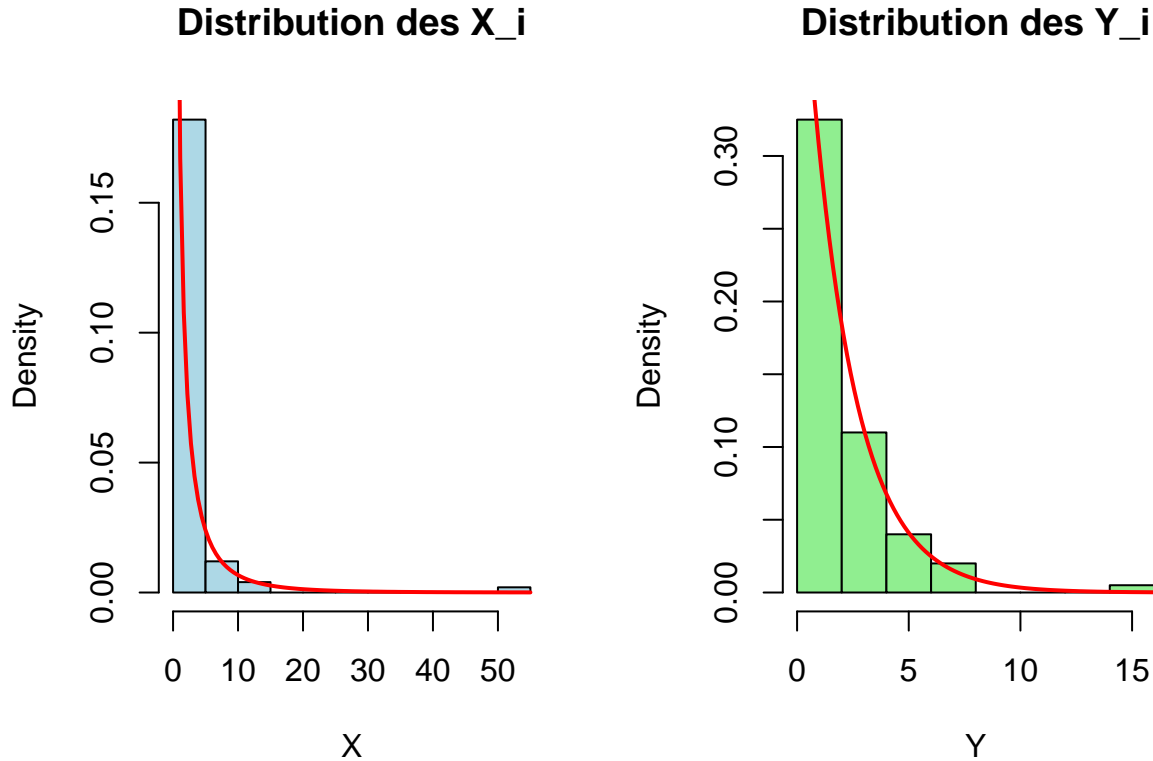
```
# Paramètres divers
set.seed(123)
par(mfrow=c(1,2))

#Définition des paramètres
n <- 100
a <- 0
c <- 1/2
theta <- 1

# Définition des variables
X <- a + rweibull(n, c, theta^(1/c))
Y <- (2/theta) * (X - a)^c

# Traçage des courbes
hist(X, main = "Distribution des X_i", xlab = "X", prob = T, col = "lightblue", border = "black")
curve(dweibull(x, c, theta^(1/c)), add = T, col = "red", lwd = 2)
```

```
hist(Y, main = "Distribution des Y_i", xlab = "Y", prob = T, col = "lightgreen", border = "black")
curve(dchisq(x, 2), add = T, col = "red", lwd = 2)
```



D'après le premier graphique, on voit que les variables  $X$  générés à l'aide de `rweibull` suivent bien une loi de Weibull malgré un  $n = 100$  faible. D'après le second graphique, on voit que les variables  $Y$  générés à l'aide de `rchisq` suivent bien une loi du chi-deux à deux degrés de liberté.

2. Dans la suite on suppose connus les paramètres  $a$  et  $c$  de la loi de  $W(\theta, a, c)$ , le paramètre  $\theta$  étant inconnu. On dispose d'un échantillon  $(X_1, \dots, X_n)$  des durées de vie observées sur  $n$  appareils électroniques,  $X_i$  étant des réalisations i.i.d. de la variable aléatoire  $X$ .

### 2.1. Calculer la vraisemblance du modèle.

En appliquant la fonction de vraisemblance pour notre cas, on obtient :

$$L(\theta; X_1, X_2, \dots, X_n) = \prod_{i=1}^n \frac{c}{\theta} (X_i - a)^{c-1} \exp\left(-\frac{(X_i - a)^c}{\theta}\right) I\{X_i > a\}$$

En simplifiant, la fonction de vraisemblance devient :

$$L(\theta; X_1, X_2, \dots, X_n) = \left(\frac{c}{\theta}\right)^n \prod_{i=1}^n (X_i - a)^{c-1} \cdot \exp\left(-\frac{1}{\theta} \sum_{i=1}^n (X_i - a)^c\right).$$

### 2.2. Montrer que l'estimateur du maximum de vraisemblance de $\theta$ est:

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^n (X_i - a)^c.$$

Pour calculer l'estimateur du maximum de vraisemblance de  $\theta$ , nous allons maximiser la fonction de vraisemblance obtenue à la question précédente par rapport à  $\theta$ . Pour simplifier, utilisons la log-vraisemblance

:

$$\ln L(\theta; X_1, X_2, \dots, X_n) = n \ln(c) - n \ln(\theta) + (c-1) \sum_{i=1}^n \ln(X_i - a) - \frac{1}{\theta} \sum_{i=1}^n (X_i - a)^c$$

Pour maximiser cette fonction, nous devons égaliser sa dérivée à 0 :

$$\frac{d}{d\theta} \ln L(\theta; X_1, X_2, \dots, X_n) = -\frac{n}{\theta} + \frac{1}{\theta^2} \sum_{i=1}^n (X_i - a)^c = 0$$

Ainsi,

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n (X_i - a)^c$$

L'estimateur EMV est donc bien  $\hat{\theta} = \frac{1}{n} \sum_{i=1}^n (X_i - a)^c$ .

### 2.3. Cet estimateur est-il sans biais? convergent?

Préparation : En posant  $Y = (X - a)^c$ ,

$$\mathbb{P}(Y \leq y) = \mathbb{P}((X - a)^c \leq y) = \mathbb{P}(X \leq a + y^{1/c})$$

Ainsi, en utilisant la fonction de répartition de X,  $\mathbb{P}(Y \leq y) = 1 - \exp(-\frac{y}{\theta})$  correspondant à la fonction de répartition d'une loi exponentielle de paramètre  $\theta$ . Ainsi,  $\mathbb{E}[Y_i] = \theta$ .

Concernant le biais : Puisque  $\hat{\theta} = \frac{1}{n} \sum_{i=1}^n Y_i$ , et que les  $Y_i$  sont i.i.d., on a :

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n Y_i\right] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Y_i] = \frac{1}{n} \cdot n \cdot \theta = \theta$$

L'estimateur  $\hat{\theta}$  est donc sans biais.

Concernant la convergence : Étant donné que les  $Y_i = (X_i - a)^c$  sont i.i.d. avec  $\mathbb{E}[Y_i] = \theta$ , selon la loi des grands nombres :

$$\frac{1}{n} \sum_{i=1}^n Y_i \xrightarrow{p} \theta \quad \text{lorsque } n \rightarrow \infty$$

$\hat{\theta}$  converge en probabilité vers  $\theta$ . Ainsi, cet estimateur est un estimateur convergent de  $\theta$ .

### 2.4. On veut illustrer les différentes propriétés de l'estimateur par des simulations. Quel protocole de simulations permet d'illustrer le biais ? la convergence ? l'efficacité ? Mettre en place ces protocoles de simulation.

Le biais est illustré en comparant la moyenne de l'estimateur calculé pour différentes tailles d'échantillons à la vraie valeur de  $\theta$ . Si l'estimateur est sans biais, alors la moyenne des  $\hat{\theta}_n$  sera proche de  $\theta$ . Le biais peut aussi être calculé directement en calculant l'espérance de l'estimateur puis en soustrayant  $\theta$ . Un estimateur sans biais doit donner un résultat proche de 0 à ce calcul. La convergence est illustrée en observant comment l'estimateur se rapproche de  $\theta$  lorsque  $n$  augmente. Pour cela, on calcule l'estimateur pour différentes tailles d'échantillons. L'efficacité d'un estimateur est liée à la variance. Un estimateur efficace a une variance faible. Pour cela, on calcule la variance de  $\hat{\theta}_n$  pour différentes tailles d'échantillons pour vérifier si la variance est bien décroissante selon  $n$ .

```

# Paramètres divers
set.seed(123)
par(mfrow = c(1,3))

# Définition des paramètres
theta <- 1
a <- 0
c <- 1/2
n_sim <- 1000
n <- 100

# Simulation de l'estimateur
simulate_theta_hat <- function(n, theta, a, c) {
  X <- a + rweibull(n, c, theta^(1/c))
  theta_hat <- mean((X - a)^c)
  return(theta_hat)
}

# Calcul du biais
theta_hat_values <- replicate(n_sim, simulate_theta_hat(n, theta, a, c))
mean_theta_hat <- mean(theta_hat_values)
biais <- mean(theta_hat_values) - theta

# Représentation graphique du biais
hist(theta_hat_values, main = "Distribution de l'estimateur", xlab = expression(hat(theta)),
      col = "lightblue", border = "black", prob = TRUE)
abline(v = theta, col = "red", lwd = 2, lty = 2)
abline(v = mean_theta_hat, col = "blue", lwd = 2)
legend("topright", legend = c("Vraie valeur de theta", "Moyenne des estimateurs"),
      col = c("red", "blue"), lty = c(2, 1), lwd = 2)

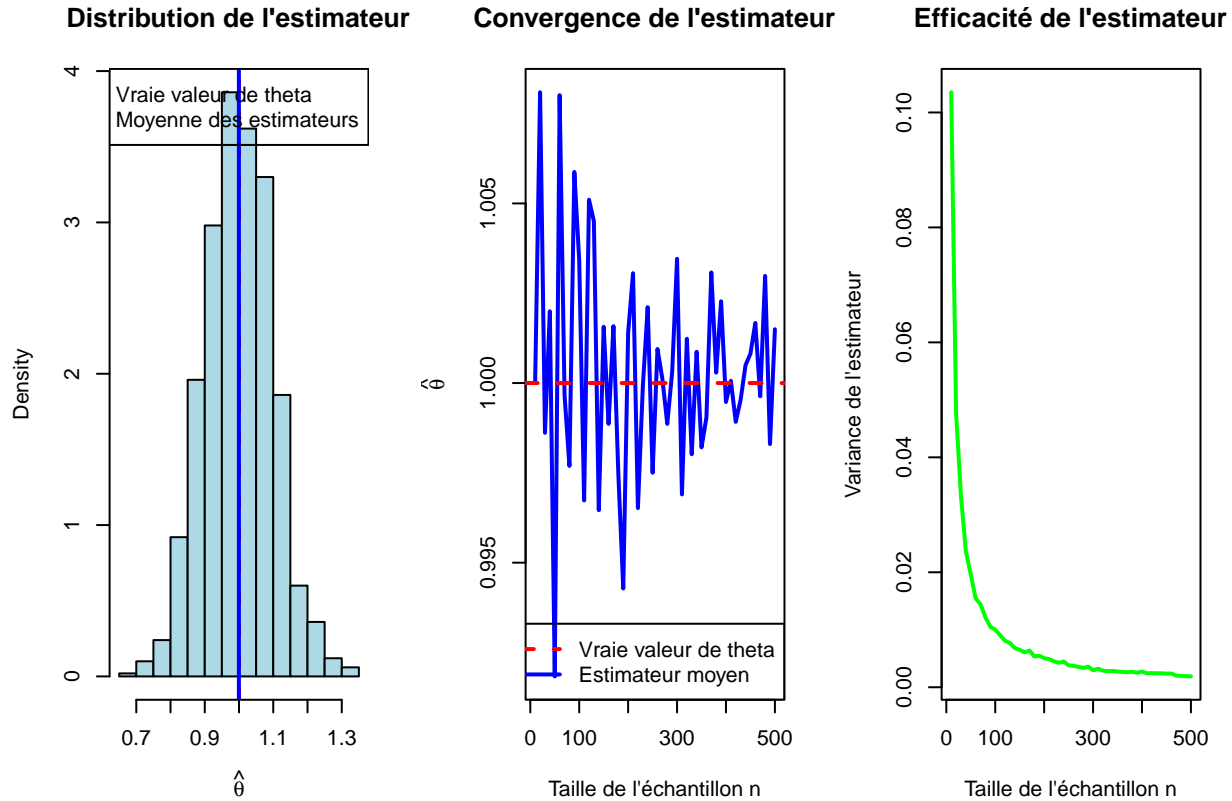
# Représentation de la convergence d l'estimateur
n_values <- seq(10, 500, by = 10)
theta_hat_convergence <- sapply(n_values, function(n) mean(replicate(n_sim, simulate_theta_hat(n, theta, a, c))))

plot(n_values, theta_hat_convergence, type = "l", col = "blue", lwd = 2,
     xlab = "Taille de l'échantillon n", ylab = expression(hat(theta)),
     main = "Convergence de l'estimateur")
abline(h = theta, col = "red", lwd = 2, lty = 2)
legend("bottomright", legend = c("Vraie valeur de theta", "Estimateur moyen"),
     col = c("red", "blue"), lty = c(2, 1), lwd = 2)

# Représentation de la décroissance de la variance
theta_hat_var <- sapply(n_values, function(n) var(replicate(n_sim, simulate_theta_hat(n, theta, a, c))))

plot(n_values, theta_hat_var, type = "l", col = "green", lwd = 2,
     xlab = "Taille de l'échantillon n", ylab = "Variance de l'estimateur",
     main = "Efficacité de l'estimateur")

```



D'après le premier graphique, la moyenne des estimateurs (ligne bleue) est très proche de la vraie valeur de  $\theta$  (ligne rouge), montrant l'absence de biais de l'estimateur. Le second graphique illustre la convergence de l'estimateur et montre que l'estimateur tend bel-et-bien vers  $\theta$  à mesure que  $n$  augmente. Le troisième graphique montre l'efficacité de l'estimateur pour différentes tailles d'échantillons. Il montre bien l'effet expliqué par la loi des grands nombres selon laquelle la variance  $\frac{\sigma^2}{n}$  est décroissante selon la taille de l'échantillon. Ici, la variance (courbe verte) décroît à mesure que  $n$  augmente, ce qui se traduit par une augmentation de l'efficacité de l'estimateur à mesure que  $n$  augmente.

3. On cherche à construire un intervalle de confiance pour  $\theta$ .

**3.1. Quelle est la loi exacte de l'estimateur  $\hat{\theta}_n$  ? Construire un intervalle de confiance pour  $\theta$  de niveau exact 90% (donner la formule).**

Pour trouver la loi exacte de  $\hat{\theta}_n$ , utilisons le fait que  $Y = \frac{2}{\theta}(X - a)^c$  suit la loi du  $\chi^2_2$ . Étant donné notre estimateur :

$$\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^n (X_i - a)^c$$

où chaque  $Y_i = (X_i - a)^c$  suit une loi exponentielle de paramètre  $\theta$  :

$$\sum_{i=1}^n \frac{2Y_i}{\theta} \sim \chi^2_{2n}$$

En posant  $T = \sum_{i=1}^n Y_i$  :

$$\frac{2T}{\theta} = \frac{2n\hat{\theta}_n}{\theta} \sim \chi^2_{2n}$$

Pour construire l'intervalle de confiance pour  $\theta$  de niveau exact 90%, nous devons utiliser les quantiles de la loi  $\chi^2$  à  $2n$  degrés de liberté :  $\chi_{2n,\alpha/2}^2$  et  $\chi_{2n,1-\alpha/2}^2$ . Ainsi, nous avons :

$$P\left(\chi_{2n,\alpha/2}^2 \leq \frac{2n\hat{\theta}_n}{\theta} \leq \chi_{2n,1-\alpha/2}^2\right) = 1 - \alpha$$

En transformant la formule pour trouver l'intervalle pour  $\theta$ , :

$$P\left(\frac{2n\hat{\theta}_n}{\chi_{2n,1-\alpha/2}^2} \leq \theta \leq \frac{2n\hat{\theta}_n}{\chi_{2n,\alpha/2}^2}\right) = 1 - \alpha$$

Pour un niveau de 90%, l'intervalle de confiance pour  $\theta$  est donc :

$$\left[ \frac{2n\hat{\theta}_n}{\chi_{2n,0.95}^2}, \frac{2n\hat{\theta}_n}{\chi_{2n,0.05}^2} \right]$$

**3.2. Donner une loi approchée de l'estimateur  $\hat{\theta}_n$ . Construire l'intervalle de confiance associé, qui est un intervalle de niveau asymptotique 90% (donner la formule).**

Par le théorème central limite, la moyenne  $\hat{\theta}_n$  d'un grand nombre de variables i.i.d.  $Y_i$  suit approximativement une loi normale :

$$\hat{\theta}_n \xrightarrow{loi} \mathcal{N}\left(\theta, \frac{\theta^2}{n}\right) \text{ lorsque } n \rightarrow \infty$$

Ainsi, on peut dire que :

$$\frac{\hat{\theta}_n - \theta}{\theta/\sqrt{n}} \xrightarrow{loi} \mathcal{N}(0, 1) \text{ lorsque } n \rightarrow \infty$$

Pour un niveau de confiance asymptotique de 90%, on utilise les quantiles de la loi  $\mathcal{N}(0, 1)$  :  $z_{1-\alpha/2}$ . Pour  $\alpha = 0.1$ ,  $z_{1-\alpha/2} = z_{0.95}$ . La formule pour l'intervalle de confiance est la suivante :

$$\left[ \bar{X} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}}, \bar{X} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right]$$

Ainsi, dans notre cas, l'intervalle de confiance est la suivante :

$$\left[ \hat{\theta}_n - z_{0.95} \frac{\hat{\theta}_n}{\sqrt{n}}, \hat{\theta}_n + z_{0.95} \frac{\hat{\theta}_n}{\sqrt{n}} \right]$$

**3.3. Quel protocole de simulations permet d'illustrer le niveau de confiance de ces intervalles ? Quel protocole permet d'illustrer l'influence de la taille de l'échantillon  $n$  ? Mettre en place ces illustrations.**

Afin d'illustrer le niveau de confiance de ces intervalles, il faut effectuer un grand nombre de simulations. Ensuite, à chaque iteration, il faut générer un échantillon de taille  $n$  suivant une loi de Weibull (en prenant en compte le paramètre de localisation  $a$  car sinon, le code ne pourra pas s'exécuter si certains  $X_i$  sont plus petits que  $a$ ), calculer l'estimateur et construire l'intervalle de confiance. Enfin, il faut vérifier si  $\theta$  est inclus dans chacune des intervalles de confiance, puis estimer empiriquement le niveau de confiance en divisant le nombre d'intervalles de confiance incluant  $\theta$  par le nombre total de simulations. Finalement, nous allons vérifier si ce niveau de confiance empirique correspond bien au niveau de confiance théorique choisi (ici, 90%). Pour illustrer l'effet de  $n$  sur le niveau de confiance, il suffit de répéter l'opération pour plusieurs tailles d'échantillon et de générer un graphique montrant les valeurs de niveau de confiance selon  $n$ , en traçant, pour référence, la droite de représentation du niveau de confiance théorique (ici, 90%).

```

set.seed(123)

# Définition des paramètres
theta <- 1
a <- 0
c <- 1/2
n_sim <- 1000
alpha <- 0.1

# Définition des différentes tailles d'échantillon
sample_sizes <- c(1, 50, 100, 500, 1000, 5000, 10000)

# Générer des échantillons selon une loi de Weibull
generate_sample <- function(theta, a, c, n) {
  return(a + (rweibull(n, c, theta^(1/c))))
}

# Création des dataframes vides
coverage_results_exact <- data.frame(Sample_Size = sample_sizes, Coverage = NA)
coverage_results_asymptotic <- data.frame(Sample_Size = sample_sizes, Coverage = NA)

# Fonctions de calcul de la couverture selon la taille de l'échantillon
for (n in sample_sizes) {
  count_in_interval <- 0

  for (i in 1:n_sim) {
    X <- generate_sample(theta, a, c, n)

    theta_hat <- mean((X - a)^c)

    lower_bound <- (2 * n * theta_hat) / qchisq(1 - alpha / 2, df = 2 * n)
    upper_bound <- (2 * n * theta_hat) / qchisq(alpha / 2, df = 2 * n)

    if (theta >= lower_bound && theta <= upper_bound) {
      count_in_interval <- count_in_interval + 1
    }
  }

  coverage_exact <- count_in_interval / n_sim
  coverage_results_exact[coverage_results_exact$Sample_Size == n, "Coverage"] <- coverage_exact
}

for (n in sample_sizes) {
  count_in_interval <- 0

  for (i in 1:n_sim) {
    X <- generate_sample(theta, a, c, n)

    theta_hat <- mean((X - a)^c)

    lower_bound <- theta_hat - qnorm(0.95) * (theta_hat / sqrt(n))
    upper_bound <- theta_hat + qnorm(0.95) * (theta_hat / sqrt(n))
  }
}

```



```

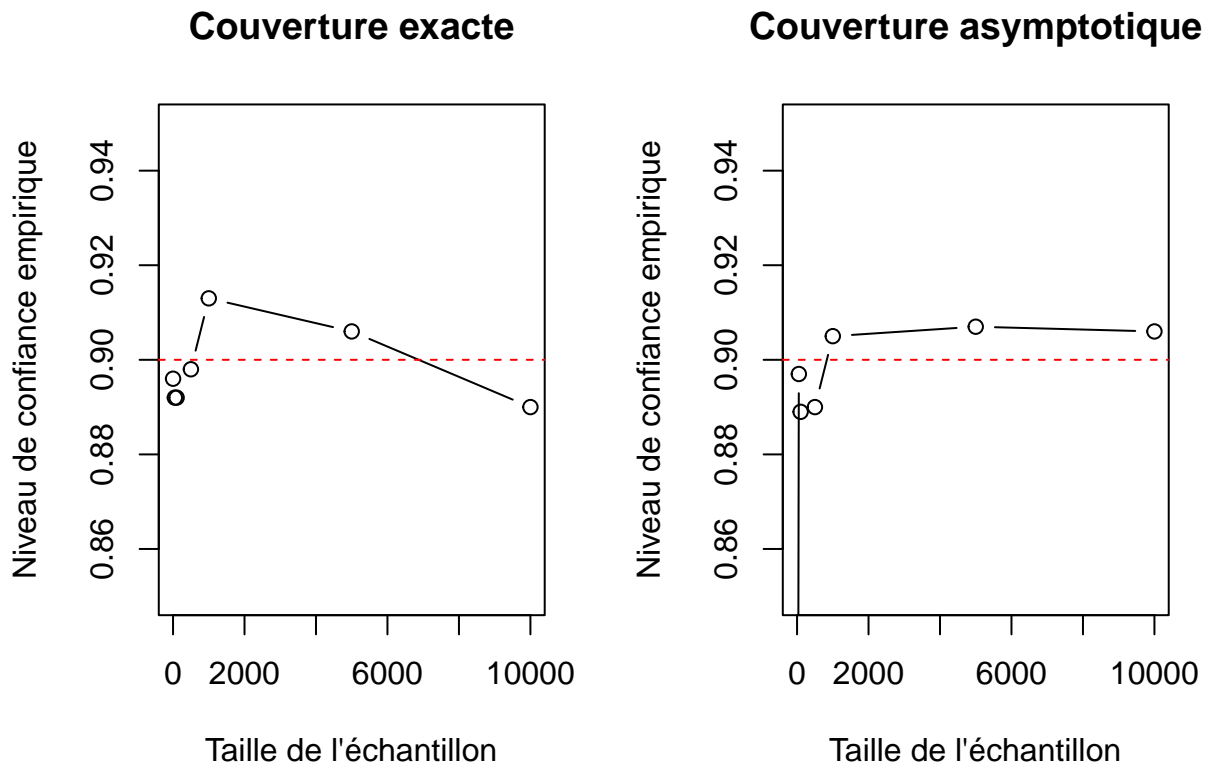
    if (theta >= lower_bound && theta <= upper_bound) {
      count_in_interval <- count_in_interval + 1
    }
  }

  coverage_asymptotic <- count_in_interval / n_sim
  coverage_results_asymptotic[coverage_results_asymptotic$Sample_Size == n, "Coverage"] <- coverage_asymptotic
}

# Représentation graphique de la convergence du niveau de confiance empirique
par(mfrow = c(1,2))
plot(coverage_results_exact$Sample_Size, coverage_results_exact$Coverage, type = "b",
     main = "Couverture exacte",
     xlab = "Taille de l'échantillon", ylab = "Niveau de confiance empirique",
     ylim = c(0.85, 0.95))
abline(h = 1 - alpha, col = "red", lty = 2)

plot(coverage_results_asymptotic$Sample_Size, coverage_results_asymptotic$Coverage, type = "b",
     main = "Couverture asymptotique",
     xlab = "Taille de l'échantillon", ylab = "Niveau de confiance empirique",
     ylim = c(0.85, 0.95))
abline(h = 1 - alpha, col = "red", lty = 2)

```



Pour ce qui est de la couverture exacte, la couverture empirique de l'intervalle de confiance exacte reste globalement proche du niveau de confiance théorique, même lorsque  $n$  est petit. Lorsque  $n$  augmente, la couverture semble se stabiliser autour du niveau de confiance donné malgré certaines fluctuations mineures. Ce résultat est bien celui attendu puisque ces intervalles sont basées sur la loi du  $\chi^2$  et devrait donc atteindre, en théorie, le niveau de confiance souhaité. Concernant la couverture asymptotique, celle-ci est inférieure

à 90% pour les petites tailles d'échantillon, ce qui est attendu car l'intervalle asymptotique repose sur une approximation normale qui n'est pas fiable pour des  $n$  très petits. À mesure que  $n$  augmente, la couverture empirique se rapproche de 90%, ce qui indique la validité asymptotique de cet intervalle. En conclusion, ces graphiques montrent donc bien l'effet de la taille de l'échantillon sur la fiabilité des intervalles de confiance. L'intervalle exact est clairement plus robuste et stable pour les petits échantillons, tandis que l'intervalle asymptotique nécessite des tailles d'échantillon plus grandes pour atteindre un niveau de confiance précis.

# M1 SSD - UE Tests - TP1 Partie 2

Lancelot Ravier

4. On veut maintenant mettre en place un test statistique de l'hypothèse simple  $H_0 : \theta = \theta_0$  contre l'hypothèse alternative simple  $H_1 : \theta = \theta_1$ , vérifiant  $\theta_1 > \theta_0 > 0$ .

1. Le lemme de Neyman-Pearson conduit à une région critique de la forme

$$R = \{(X_1, \dots, X_n) : \sum_{i=1}^n (X_i - a)^c \geq k\},$$

où  $k > 0$  est une constante. Soit  $0 < \alpha < 1$  un niveau de signification fixé et soit  $\chi_{2n}^2(\gamma)$  le quantile d'ordre  $\gamma$  de la loi  $\chi_{2n}^2$ . Déterminer en fonction de  $\theta_0$  et de  $\chi_{2n}^2(\gamma)$ , la région critique d'un test de niveau  $\alpha$ .

le risque de première espèce  $\alpha$  est donné par la probabilité d'accepter  $H_1$  alors que  $H_0$  est vraie :

$$\alpha = \mathbb{P}_{H_0} \left( \sum_{i=1}^n (X_i - a)^c \geq k \right)$$

D'après les questions précédentes, on sait que :

$$\frac{2}{\theta_0} \sum_{i=1}^n (X_i - a)^c \sim \chi^2(2n)$$

Ainsi, dans notre cas, nous pouvons réécrire cette probabilité en utilisant la distribution de la statistique de test :

$$\alpha = \mathbb{P}_{H_0} \left( \frac{2}{\theta_0} \sum_{i=1}^n (X_i - a)^c \geq \chi_{1-\alpha, 2n}^2 \right)$$

Afin de garantir un test de niveau  $\alpha$ , nous choisissons  $k$  tel que :

$$\frac{2k}{\theta_0} = \chi_{2n, 1-\alpha}^2 \iff k = \frac{\theta_0}{2} \chi_{1-\alpha, 2n}^2$$

La région critique du test est donc donnée par :

$$R = \left\{ (X_1, \dots, X_n) : \sum_{i=1}^n (X_i - a)^c \geq \frac{\theta_0}{2} \chi_{1-\alpha, 2n}^2 \right\}$$

2. Exprimer en fonction de  $\theta_0$ ,  $\theta_1$  et de  $\chi_{2n}^2(\gamma)$  et à l'aide de la fonction de répartition  $F$  de la loi  $\chi_{2n}^2$ , le risque de seconde espèce et la puissance de ce test.

Le risque de seconde espèce  $\beta$  représenté est donné par la probabilité de garder  $H_0$  alors que  $H_1$  est vraie :

$$\beta = \mathbb{P}_{H_1} \left( \sum_{i=1}^n (X_i - a)^c < \frac{\theta_0}{2} \chi_{1-\alpha, 2n}^2 \right)$$

Sous  $H_1$  :

$$\frac{2}{\theta_1} \sum_{i=1}^n (X_i - a)^c \sim \chi^2(2n)$$

Ainsi, en remplaçant :

$$\beta = \mathbb{P}_{H_1} \left( \frac{2}{\theta_1} \sum_{i=1}^n (X_i - a)^c < \frac{2}{\theta_1} \frac{\theta_0}{2} \chi_{1-\alpha, 2n}^2 \right)$$

Donc, le risque de seconde espèce  $\beta$  est donné par :

$$\beta = \mathbb{P}_{H_1} \left( \frac{2}{\theta_1} \sum_{i=1}^n (X_i - a)^c < \frac{\theta_0}{\theta_1} \chi_{1-\alpha, 2n}^2 \right)$$

Donc,

$$\beta = \mathbb{P}_{H_1} \left( \sum_{i=1}^n (X_i - a)^c < \frac{\theta_0}{2} \chi_{1-\alpha, 2n}^2 \right)$$

Avec  $\chi_{1-\alpha, 2n}^2$  le quantile d'ordre  $1 - \alpha$  de la loi du  $\chi^2$  à  $2n$  degrés de liberté.

A partir du risque de seconde espèce  $\beta$ , la puissance d'un test est donnée par :

$$\text{Puissance} = 1 - \beta = 1 - F_{\chi_{2n}^2} \left( \frac{\theta_0}{\theta_1} \chi_{1-\alpha, 2n}^2 \right)$$

3. Mettre en place sous R ce test pour un échantillon de taille  $n=100$  avec les paramètres  $a=0$ ,  $c=1/2$  et  $\theta = \theta_0 = 1$ . Illustrer le niveau du risque de première espèce de ce test. Illustrer numériquement la puissance du test en faisant varier  $\alpha$  pour un  $\theta_1$  fixé, puis en faisant varier  $\theta_1$  pour  $\alpha$  fixé.

Afin d'illustrer le niveau du risque de première espèce du test (risque d'accepter  $H_1$  alors que  $H_0$  est vraie), il est pertinent d'observer les variations en fonction de la taille de l'échantillon afin de comprendre les propriétés d'un test correctement conçu et bien calibré. Pour cela, après avoir défini les paramètres et les fonctions pour générer l'échantillon selon une loi de Weibull sous  $H_0$  (en utilisant  $\theta_0$ , car nous devons mesurer le risque de rejeter  $H_0$  alors que  $H_0$  est vraie et, donc, générer les échantillon de sorte à ce que  $H_0$  sont respectée), le test statistique et la valeur critique théorique définie ci-dessus, nous devons créer une fonction qui appliquera le test aux différents échantillons de différentes tailles (avec la fonction `sapply`) et calculera la moyenne des estimateurs pour lesquels on rejette  $H_0$ . Ensuite, nous pouvons créer un graphique représentant, pour chaque points, le risque de première espèce estimé pour chaque échantillon, en traçant une droite représentant le seuil théorique choisi (ici,  $\alpha = 0.05$ ). Ainsi, nous allons pouvoir analyser l'effet de la taille de l'échantillon sur le test.

Pour l'illustration numérique de la puissance du test, il faut générer des échantillons sous  $H_1$  (donc, en utilisant  $\theta_1$ ) car nous devons générer des échantillons respectant  $H_1$  afin d'analyser le nombre de fois que le test accepte  $H_1$ , dans le cas où  $H_1$  est vraie. Ensuite, de la même manière que pour le risque de première espèce, nous allons utiliser la fonction `sapply` pour faire varier, dans une première dataframe, le seuil théorique  $\alpha$  puis, dans un second dataframe, faire varier  $\theta_1$ . Les deux graphiques seront constitués à partir des deux dataframes, d'une courbe représentant la puissance du test (moyenne des estimateurs pour lesquels  $H_0$  est rejeté) pour différents  $\alpha$ , puis pour différents  $\theta_1$ .

```

par(mfrow = c(1,3))
set.seed(123)

n <- 100
a <- 0
c <- 1/2
theta_0 <- 1
theta_1 <- 1.5
alpha <- 0.05
alpha_list <- seq(0.01, 0.2, by = 0.01)
theta_1_list <- seq(1, 2, by = 0.1)
n_list <- seq(10, 1000, by = 10)
n_sim <- 1000

sample <- function(n, theta) {
  a + rweibull(n, c, theta^(1/c))
}

test_stats <- function(sample, a, c) {
  sum((sample - a)^c)
}

critical_value <- function(n, alpha) {
  theta_0 / 2 * qchisq(1 - alpha, 2 * n)
}

# Risque de première espèce

mean_risk_alpha_n <- sapply(n_list, function(n) {
  mean(sapply(1:10000, function(i) {
    sample_H0 <- sample(n, theta_0)
    k <- critical_value(n, alpha)
    test <- test_stats(sample_H0, a, c)
    test >= k
  })))
})

plot(n_list, mean_risk_alpha_n, xlim=c(n_list[1],n_list[100]), ylim=c(0,0.1), xlab = "taille d'échantillon",
abline(alpha, 0, col = "red")
legend("bottomleft", legend = "seuil théorique = 0.05", col = "red", lty = 1, lwd = 2)

# Puissance en fonction de alpha

power_alpha <- sapply(alpha_list, function(alpha) {
  mean(sapply(1:n_sim, function(i) {
    k <- critical_value(n, alpha)
    sample_H1 <- sample(n, theta_1)
    test <- test_stats(sample_H1, a, c)
    test >= k
  })))
})

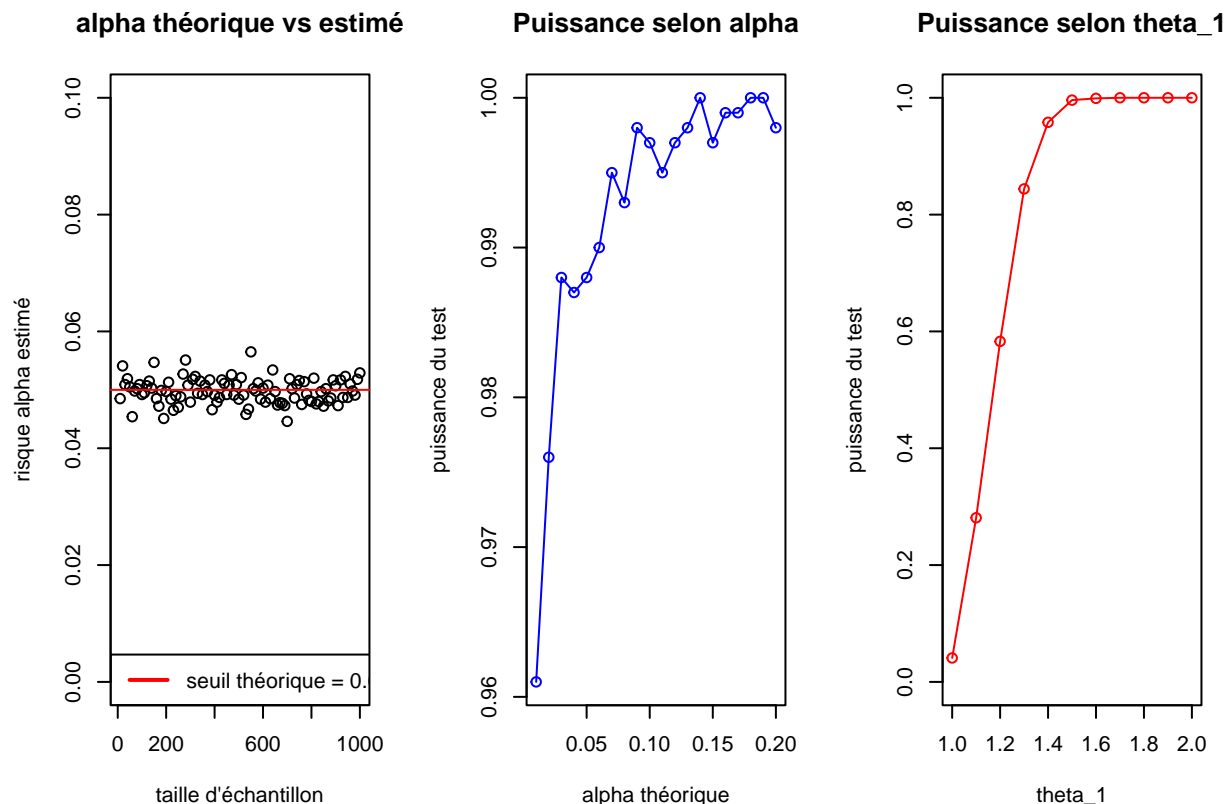
plot(alpha_list, power_alpha, xlim = c(alpha_list[1],alpha_list[20]), ylim = c(power_alpha[1],power_alpha[20]),
ylab = "puissance du test", main = "Puissance selon alpha")

```

```
# Puissance du test en fonction de theta_1 pour alpha fixe

power_beta <- sapply(theta_1_list, function(theta_1) {
  mean(sapply(1:n_sim, function(i) {
    k <- critical_value(n, alpha)
    sample_H1 <- sample(n, theta_1)
    test <- test_stats(sample_H1, a, c)
    test >= k
  })))
})

plot(theta_1_list, power_beta, xlim = c(1,2), ylim = c(0,1), type = "o", col = "red", xlab = "theta_1",
      ylab = "puissance du test", main = "Puissance selon theta_1")
```



Le premier graphique montre le risque de première espèce estimé du test pour différentes tailles d'échantillon. Le test étant constitué de manière exacte, il est normal que les risques estimés restent proches du risque théorique indépendamment de la taille de l'échantillon. En effet, la taille de l'échantillon affecte la puissance et le risque de seconde espèce du test, mais n'influe pas directement le risque de première espèce car celui-ci est déterminé par le seuil critique choisi.

Le second graphique nous montre la puissance du test selon le risque  $\alpha$  théorique. Nous observons une augmentation de la puissance du test à mesure que le risque  $\alpha$  théorique augmente. Cette augmentation est attendue, et expliquée par le fait qu'en augmentant le seuil  $\alpha$  théorique, on élargit la région critique, augmentant ainsi la probabilité de rejeter  $H_0$ , même pour des effets plus faibles ou des résultats moins significatifs. Par exemple, lorsque le seuil est fixé à 20%, la puissance du test est accrue car il devient plus sensible aux écarts par rapport à  $H_0$ . Lorsque le seuil est fixé à 1%, on voit bien que le test est moins puissant car moins permissif.

Le troisième graphique montre la puissance du test en fonction de  $\theta_1$ , avec alpha fixé à 5%. On voit bien

que la puissance du test augmente à mesure que  $\theta_1$  s'éloigne de  $\theta_0$ . En effet, cette convergence est expliquée par le fait que, à mesure que l'écart entre  $\theta_0$  et  $\theta_1$  est important (par exemple pour  $\theta_0 = 1$  et  $\theta_1 = 2$ ), les observations sous  $\theta_1$  deviennent plus distinctes de celle sous  $\theta_0$ . Ainsi, le test est davantage capable de détecter la différence et de rejeter  $H_0$  lorsqu'elle est fausse. Si  $\theta_0$  se rapproche de  $\theta_1$ , la puissance diminue. En effet, lorsque  $\theta_0$  et  $\theta_1$  sont proches (par exemple, pour  $\theta_0 = 1$  et  $\theta_1 = 1.2$ ), les distributions des observations sous  $H_0$  et  $H_1$  se chevauchent davantage, rendant plus difficile, pour le test, de différencier  $H_1$  de  $H_0$ , ce qui augmente le risque de seconde espèce et, donc, diminue la puissance du test.

En conclusion, le test est correctement calibré. En effet, le risque de première espèce estimé reste proche du niveau théorique (ici fixé à 5%) quelque soit la taille de l'échantillon. Aussi, cela montre que la taille de l'échantillon n'affecte pas directement le contrôle du risque de première espèce (car le test est bien conçu). La puissance du test augmente à mesure que le seuil  $\alpha$  théorique augmente. En effet, en acceptant un risque de première espèce plus élevé, le test devient donc davantage de rejeter  $H_0$  lorsque  $H_1$  est vraie. Aussi, la puissance augmente de manière significative avec l'écart entre  $\theta_1$  et  $\theta_0$ . En effet, lorsque cet écart est grand, il devient plus difficile pour le test d'identifier  $H_1$  et, donc, de rejeter  $H_0$ .

5. On considère maintenant le test statistique de l'hypothèse  $H_0 : \theta \leq 1$  contre  $H_1 : \theta > 1$ .

1. Proposer un test uniformément le plus puissant pour un niveau de signification  $0 < \alpha < 1$  fixé.

Le lemme de Neyman-Pearson ne s'applique qu'aux tests d'hypothèses simples (exemple :  $\theta = \theta_0$  contre  $\theta = \theta_1$ ). Dans notre cas, l'hypothèse nulle est composite car elle inclut toutes les valeurs de  $\theta$  tel que  $\theta \leq \theta_0$ . Cependant, si la famille de distribution possède la propriété de rapport de vraisemblance monotone par rapport à une statistique suffisante et que la distribution des observations peut se réécrire sous forme exponentielle, nous pouvons trouver un test UPP monotone pour une hypothèse composite à l'aide du théorème de Karlin-Rubin.

Rappelons que pour chaque  $i$ , la variable  $Y_i = (X_i - a)^c$  suit une loi exponentielle de paramètre  $\theta$  :

$$Y_i \sim \text{Exp}(\theta)$$

La somme  $S = \sum_{i=1}^n Y_i$  suit donc une loi Gamma de paramètre  $n, \theta$  :

$$S \sim \Gamma(n, \theta)$$

La fonction de densité de  $S$  est :

$$f_S(s; \theta) = \frac{1}{\Gamma(n)\theta^n} s^{n-1} e^{-s/\theta}, \quad s > 0$$

Le rapport de vraisemblance est alors :

$$\lambda(s) = \frac{f_S(s, \theta)}{f_S(s, \theta_0)} = \left( \frac{\theta_0}{\theta} \right)^n e^{-s(\frac{1}{\theta} - \frac{1}{\theta_0})}$$

Sous  $H_1$ , nous avons  $\frac{1}{\theta} - \frac{1}{\theta_0} < 0$ , donc le terme  $e^{-s(\frac{1}{\theta} - \frac{1}{\theta_0})}$  est une fonction croissante de  $s$ . Ainsi, le rapport de vraisemblance  $\lambda(s)$  est une fonction croissante de  $s$ , ce qui signifie que le théorème de Karlin-Rubin est applicable dans notre cas pour obtenir un test UPP monotone basé sur la statistique  $s$ .

Pour un niveau de signification  $\alpha$ , nous devons déterminer le seuil  $k_\alpha$  satisfaisant :

$$\mathbb{P}_{\theta_0}(S \geq k_\alpha) = \alpha$$

En sachant que sous  $H_0$ ,  $S \sim \Gamma(n, \theta_0)$ , nous pouvons transformer  $S$  en une variable suivant une loi du  $\chi^2_{2n}$  :

$$\frac{2S}{\theta_0} \sim \chi^2_{2n}$$

Ainsi, le seuil  $k_\alpha$  est donné par :

$$\frac{2k_\alpha}{\theta_0} = \chi^2_{1-\alpha, 2n} \iff k_\alpha = \frac{\theta_0}{2} \chi^2_{1-\alpha, 2n}$$

La statistique de test UPP monotone est donc la suivante :

$$\sum_{i=1}^n (X_i - a)^c$$

Avec comme règle de décision :

$$\text{Rejeter } H_0 \text{ si } \sum_{i=1}^n (X_i - a)^c \geq \frac{\theta_0}{2} \chi_{2n}^2(1 - \alpha). \text{ Accepter } H_0 \text{ sinon.}$$

2. Proposer un test asymptotique de niveau  $\alpha$  de l'hypothèse  $H_0 : \theta \leq 1$  contre  $H_1 : \theta > 1$ .

Partons du résultat trouvé à la question 2.2 :

L'estimateur du maximum de vraisemblance de  $\theta$  est :

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n (X_i - a)^c$$

Puisque les  $Y_i$  sont des variables aléatoires i.i.d. exponentielles d'espérance  $\theta$  et de variance  $\theta^2$ , selon le TCL, la moyenne empirique  $\hat{\theta}$  est approximativement distribuée selon une loi normale (pour  $n$  grand) :

$$\sqrt{n} \cdot (\hat{\theta} - \theta) \xrightarrow[n \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, \theta^2)$$

Sous  $H_0$ , en particulier lorsque  $\theta = \theta_0$ , la variance asymptotique de  $\hat{\theta}$  devient  $\theta_0^2$ . Ainsi, sous  $H_0$  :

$$\sqrt{n} \cdot \frac{\hat{\theta} - \theta_0}{\theta_0} \xrightarrow[n \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, 1)$$

La statistique de test asymptotique est donc la suivante :

$$T = \sqrt{n} \cdot \frac{\frac{1}{n} \sum_{i=1}^n (X_i - a)^c - \theta_0}{\theta_0}$$

Avec comme règle de décision :

$$\text{Rejeter } H_0 \text{ si } T \geq z_{1-\alpha}. \text{ Accepter } H_0 \text{ sinon.}$$

3. Proposer une illustration numérique du niveau des deux tests en faisant varier  $\alpha$ .

Afin de répondre à la question, Pour étudier le comportement des deux tests statistiques (le test UPP monotone et le test asymptotique) en faisant varier  $\alpha$ , nous devons tout d'abord définir les paramètres et les fonctions des tests et des régions critiques, avec la fonction pour générer un échantillon selon la loi de Weibull.

Il est essentiel, pour une comparaison rigoureuse des niveaux des deux tests, d'appliquer ceux-ci aux mêmes échantillons pour éviter les variations dues à l'aléa du caractère aléatoire des générations de données. Ainsi, le calcul des niveaux sera effectué au sein d'une même boucle d'itération. Les résultats seront stockés dans une liste afin de créer un graphique comportant deux courbes représentant les risques de première espèce des deux tests.

Examinons d'abord l'effet du nombre de simulations sur les niveaux des deux tests. Pour ce faire, nous commençons par générer trois graphiques illustrant les niveaux des tests pour trois nombre de simulations différents.

```
set.seed(1)
```

```
# Paramètres
```



```

theta_0 <- 1
a <- 0
c <- 1/2
n <- 10
n_simulations_values <- c(100, 1000, 20000) # différents valeurs pour n_simulations
alpha_levels <- seq(0.01, 0.99, by = 0.01) # différents seuils de significativité

# generation des echantillons selon une loi de weibull
sample <- function(n, theta) {
  a + rweibull(n, c, theta^(1/c))
}

# test statistique UPP monotone
test_1 <- function(sample) {
  sum((sample - a)^c)
}

# test statistique asymptotique
test_2 <- function(sample, theta) {
  sqrt(n) * (1/n * sum((sample - a)^c) - theta) / theta
}

# seuil critique du test UPP monotone
critical_1 <- function(n, alpha, theta) {
  theta / 2 * qchisq(1 - alpha, 2 * n)
}

# seuil critique du test asymptotique
critical_2 <- function(alpha) {
  qnorm(1 - alpha)
}

# creation de la dataframe resultats
results <- data.frame()

# simulation pour chaque valeur de n_simulations
for (n_simulations in n_simulations_values) {
  for (alpha in alpha_levels) {
    reject_test_1 <- 0
    reject_test_2 <- 0

    for (sim in 1:n_simulations) {
      # génération d'un échantillon sous H0
      sample_data <- sample(n, theta_0)

      # calcul des statistiques de test
      T1 <- test_1(sample_data)
      T2 <- test_2(sample_data, theta_0)

      # calcul des seuils critiques
      crit_1 <- critical_1(n, alpha, theta_0)
      crit_2 <- critical_2(alpha)
    }
  }
}

```

```

# tests
if (T1 > crit_1) {
  reject_test_1 <- reject_test_1 + 1
}
if (T2 > crit_2) {
  reject_test_2 <- reject_test_2 + 1
}
}

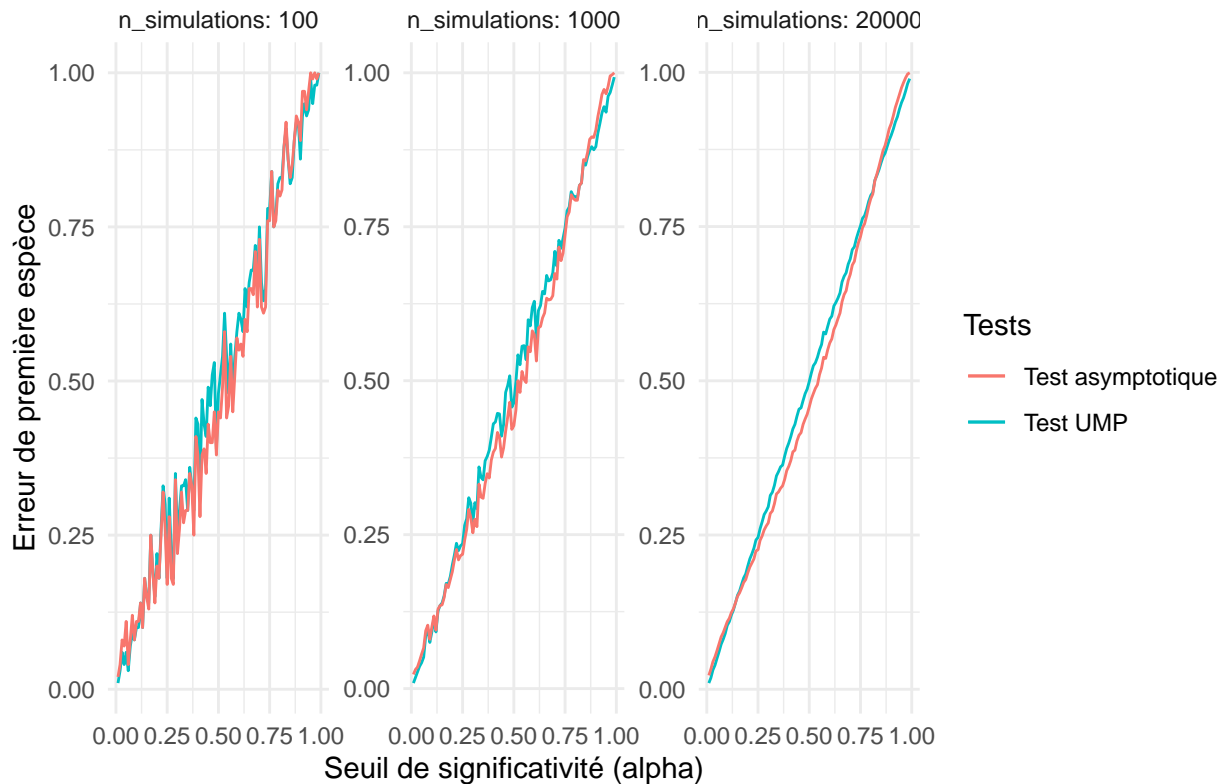
# ajouter les résultats à la dataframe
results <- rbind(results, data.frame(
  n_simulations = n_simulations,
  alpha = alpha,
  error_test_1 = reject_test_1 / n_simulations,
  error_test_2 = reject_test_2 / n_simulations
))
}
}

# partie graphique
library(ggplot2)

ggplot(results, aes(x = alpha)) +
  geom_line(aes(y = error_test_1, color = "Test UMP")) +
  geom_line(aes(y = error_test_2, color = "Test asymptotique")) +
  facet_wrap(~ n_simulations, scales = "free_y", labeller = label_both) +
  labs(title = "Erreur de première espèce des deux tests pour différentes valeurs de n_simulations",
       x = "Seuil de significativité (alpha)",
       y = "Erreur de première espèce",
       color = "Tests") +
  theme_minimal()

```

## Erreur de première espèce des deux tests pour différentes valeurs de $n_{si}$



Le premier graphique montre l'évolution du comportement des deux tests en fonction d'un nombre réduit de simulations (100). Les courbes obtenues sont plus bruitées en raison de la variabilité idue au caractère aléatoire de l'échantillon généré. Cette variabilité entraîne des variations importantes des estimations du risque d'erreur  $\alpha$  des deux tests par rapport au seuil théorique.

Lorsque le nombre de simulations augmente, les courbes deviennent plus lisses, reflétant une meilleure estimation de l'erreur de première espèce. En effet, selon la loi des grands nombres, un grand nombre de simulations permet aux fréquences observées de converger vers les probabilités théoriques. Ainsi, avec un nombre de simulations fixé à 1000 ou 20000, les résultats calculés convergent vers les résultats théoriques. Un nombre élevé de simulations permet donc de réduire l'impact du bruit statistique et de détecter les différences subtiles entre les deux tests, qui pourraient être masquées par un bruit trop important (comme dans le graphique 1) avec un nombre de simulations insuffisant.

Au-delà du nombre de simulations, il serait intéressant d'analyser l'effet de la taille de l'échantillon sur la précision des tests, en particulier la précision du test asymptotique, qui de par sa construction, ne devrait être précis que pour des tailles élevées d'échantillons. Pour analyser cet effet, nous fixons le nombre de simulations à 20000 afin de minimiser le bruit statistique tout en gardant une vitesse de calcul raisonnable.

```
set.seed(1)

theta_0 <- 1
a <- 0
c <- 1/2
n_values <- c(1, 10, 1000) # différentes tailles d'échantillons
n_simulations <- 20000
alpha_levels <- seq(0.01, 0.99, by = 0.01)
```

```

sample <- function(n, theta) {
  a + rweibull(n, c, theta^(1/c))
}

test_1 <- function(sample) {
  sum((sample - a)^c)
}

test_2 <- function(sample, theta) {
  sqrt(length(sample)) * (1/length(sample) * sum((sample - a)^c) - theta) / theta
}

critical_1 <- function(n, alpha, theta) {
  theta / 2 * qchisq(1 - alpha, 2 * n)
}

critical_2 <- function(alpha) {
  qnorm(1 - alpha)
}

results <- data.frame()

for (n in n_values) { # appliquer la fonction aux différents n
  for (alpha in alpha_levels) {
    reject_test_1 <- 0
    reject_test_2 <- 0

    for (sim in 1:n_simulations) {

      sample_data <- sample(n, theta_0)

      T1 <- test_1(sample_data)
      T2 <- test_2(sample_data, theta_0)

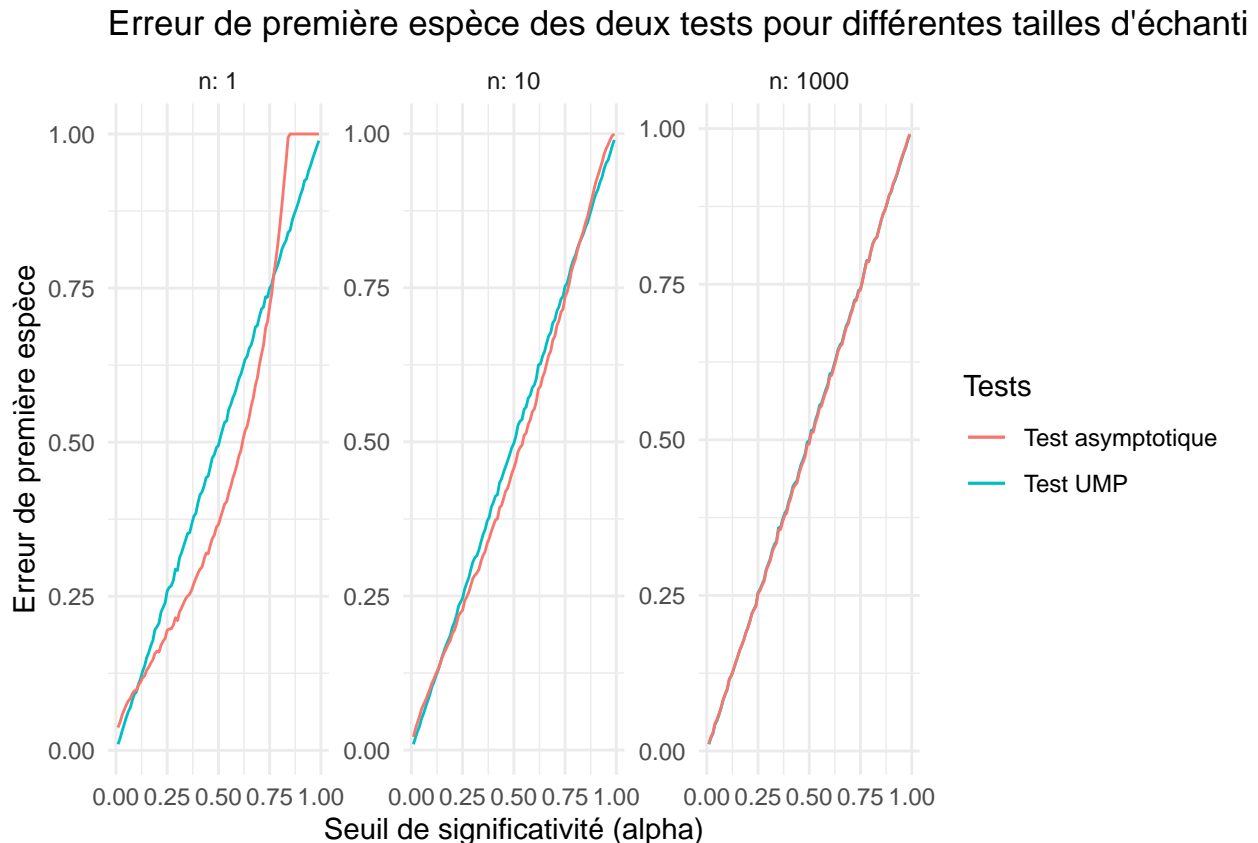
      crit_1 <- critical_1(n, alpha, theta_0)
      crit_2 <- critical_2(alpha)

      if (T1 > crit_1) {
        reject_test_1 <- reject_test_1 + 1
      }
      if (T2 > crit_2) {
        reject_test_2 <- reject_test_2 + 1
      }
    }

    results <- rbind(results, data.frame(
      n = n,
      alpha = alpha,
      error_test_1 = reject_test_1 / n_simulations,
      error_test_2 = reject_test_2 / n_simulations
    ))
  }
}

```

```
ggplot(results, aes(x = alpha)) +
  geom_line(aes(y = error_test_1, color = "Test UMP")) +
  geom_line(aes(y = error_test_2, color = "Test asymptotique")) +
  facet_wrap(~ n, scales = "free_y", labeller = label_both) +
  labs(title = "Erreur de première espèce des deux tests pour différentes tailles d'échantillon",
       x = "Seuil de significativité (alpha)",
       y = "Erreur de première espèce",
       color = "Tests") +
  theme_minimal()
```



Les graphiques obtenus montrent que, pour de petites tailles d'échantillons, les différences entre les deux tests sont significatives. Le test UPP monotone, étant exact par construction, maintient une estimation précise du risque de première espèce indépendamment de la taille des échantillons. En revanche, le test asymptotique repose sur une approximation normale de la statistique de test, qui n'est valide que pour des échantillons suffisamment grands.

Pour de petits échantillons (exemple :  $n = 1$ ), l'approximation normale du test asymptotique est moins précise, ce qui conduit à des déviations par rapport au niveau de significativité théorique. Cela s'explique par le fait que le Théorème Central Limite (TCL), sur lequel se base l'approximation normale, nécessite des échantillons suffisamment grands afin que la distribution de la statistique de test soit bien approchée par une loi normale.

À mesure que la taille d'échantillon augmente, les différences entre les deux tests diminuent. Le test asymptotique converge vers le test UPP monotone lorsque  $n$  tend vers l'infini, conformément aux propriétés asymptotiques des statistiques de test. Ainsi, pour de grandes tailles d'échantillon, les deux tests présentent des performances similaires et respectent le niveau  $\alpha$  théoriques.

```

set.seed(1)

theta_0 <- 1
a <- 0
c <- 1/2
n <- 1000
n_simulations <- 20000
alpha_levels <- seq(0.01, 0.99, by = 0.01)

sample <- function(n, theta) {
  a + rweibull(n, c, theta^(1/c))
}

test_1 <- function(sample) {
  sum((sample - a)^c)
}

test_2 <- function(sample, theta) {
  sqrt(length(sample)) * (1/length(sample) * sum((sample - a)^c) - theta) / theta
}

critical_1 <- function(n, alpha, theta) {
  theta / 2 * qchisq(1 - alpha, 2 * n)
}

critical_2 <- function(alpha) {
  qnorm(1 - alpha)
}

results <- data.frame()

for (alpha in alpha_levels) {
  reject_test_1 <- 0
  reject_test_2 <- 0

  for (sim in 1:n_simulations) {

    sample_data <- sample(n, theta_0)

    T1 <- test_1(sample_data)
    T2 <- test_2(sample_data, theta_0)

    crit_1 <- critical_1(n, alpha, theta_0)
    crit_2 <- critical_2(alpha)

    if (T1 > crit_1) {
      reject_test_1 <- reject_test_1 + 1
    }
    if (T2 > crit_2) {
      reject_test_2 <- reject_test_2 + 1
    }
  }
}

```

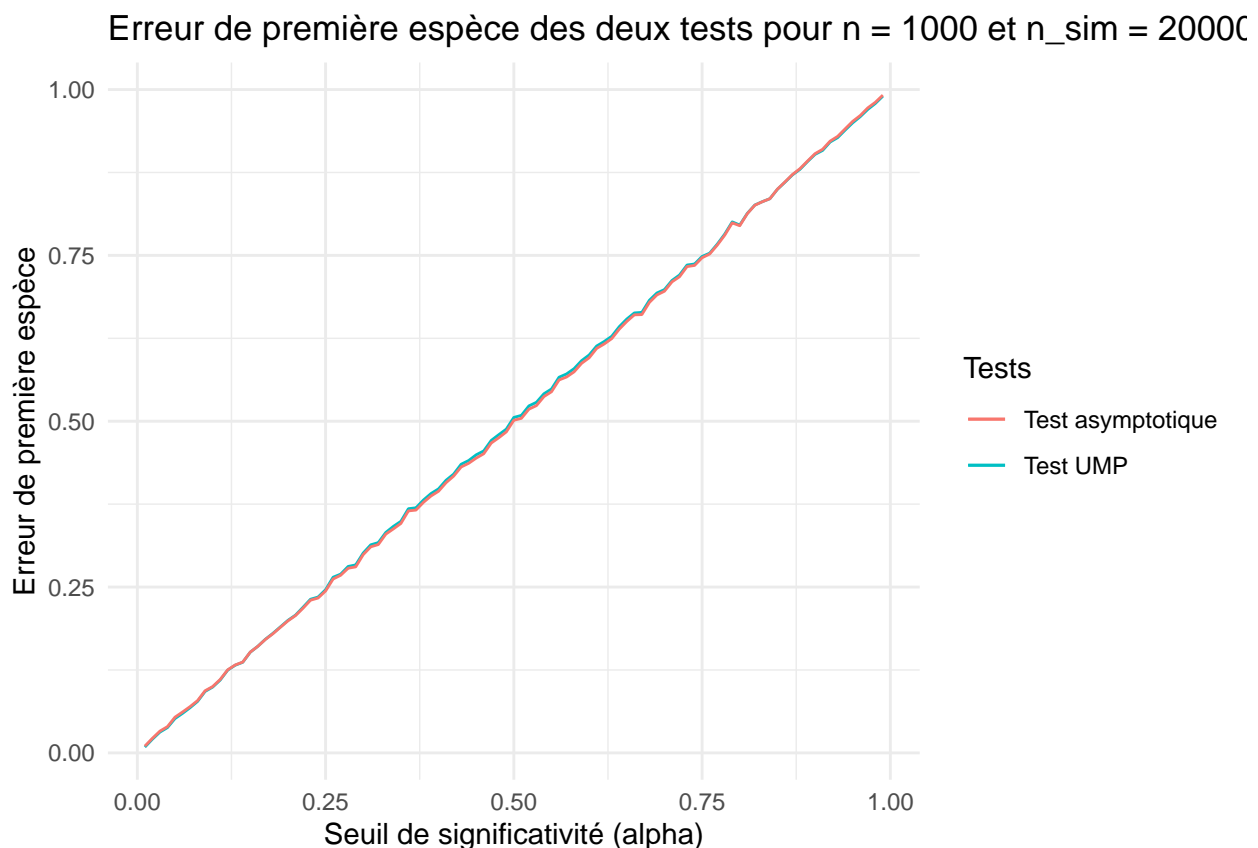
```

results <- rbind(results, data.frame(
  n = n,
  n_simulations = n_simulations,
  alpha = alpha,
  error_test_1 = reject_test_1 / n_simulations,
  error_test_2 = reject_test_2 / n_simulations
))
}

# Visualisation

ggplot(results, aes(x = alpha)) +
  geom_line(aes(y = error_test_1, color = "Test UMP")) +
  geom_line(aes(y = error_test_2, color = "Test asymptotique")) +
  labs(title = "Erreur de première espèce des deux tests pour n = 1000 et n_sim = 20000",
       x = "Seuil de significativité (alpha)",
       y = "Erreur de première espèce",
       color = "Tests") +
  theme_minimal()

```



En examinant les niveaux des deux tests pour des tailles d'échantillons et un nombre de simulations élevées, nous constatons que le test asymptotique suit de manière précise le niveau de significativité théorique. L'approximation normale utilisée dans le test asymptotique devient robuste pour de grandes tailles d'échantillon, grâce au TCL. En standardisant la statistique de test avec la taille de l'échantillon et la variance, le test asymptotique atténue l'impact des asymétries de la distribution de Weibull.

Le test UPP monotone, bien qu'exact pour toutes les valeurs de  $n$ , peut présenter de légères déviations du

niveau  $\alpha$  théorique pour des valeurs  $\alpha$  extrêmes. Ces déviations peuvent s'expliquer par la discrétisation des seuils critiques issus de la loi du  $\chi^2$ , qui peut être moins précise dans les extrémités de la distribution.

Le test asymptotique repose sur le Théorème Central Limite (TCL), selon lequel la distribution de la moyenne empirique tend vers une loi normale lorsque  $n$  tend vers l'infini. Cependant, cette convergence est asymptotique, ce qui signifie qu'elle n'est exacte que lorsque  $n$  est infiniment grand. Même avec  $n = 1000$ , certains écarts peuvent toujours être observés entre la distribution réelle de la statistique de test et son approximation normale. Ces écarts sont dus à la vitesse de convergence, qui dépend de la distribution sous-jacente.

Pour finir, même avec  $n_{simulations} = 20000$ , il existe une variabilité statistique naturelle dans les estimations de l'erreur de première espèce. Par exemple avec  $\hat{p} \approx 0.01015$  et  $n_{simulations} = 20000$ , l'erreur type est environ 0.0007. Ainsi, la différence observée entre les deux tests peut être due, en partie, à cette variabilité.

En conclusion, le test UPP monotone se distingue par son exactitude (du à sa constitution théorique). Le test asymptotique, fondé sur une approximation normale, montre une forte variabilité pour des échantillons de taille réduite (dû aux limitations asymptotiques du TCL). À mesure que les tailles d'échantillons augmentent, le test asymptotique converge vers les performances du test UPP monotone. L'augmentation du nombre de simulation joue un rôle crucial pour les deux tests dans la réduction du bruit. Lorsque le nombre de simulations est grand, les résultats des deux tests deviennent plus fiables. Cependant, des variations peuvent subsister en raison de la variabilité due au caractère aléatoire des simulations. Dans la plupart des cas réels, il est difficile de connaître avec certitude la distribution de nos observations. Ainsi, même s'il est privilégié, le test UPP est souvent impossible à constituer. Ainsi, le test asymptotique peut être aussi utilisé car, comme nous l'avons vu, ses performances se rapprochent de celles du test UPP, dans le cas, bien sûr, où la taille des échantillons est suffisamment grande.