TP 4 – SY02
Intervalle de confiance
Corrigé

Les questions/sections marquées par un sont des questions qui sont prévues pour être traitées en autonomie en dehors de la séance de TP.

1 Fonctions pivotales

Dans cette section, nous allons vérifier expérimentalement les lois suivies par les 2 principales fonctions pivotales utilisées pour la construction d'intervalles de confiance dans le cadre d'un échantillon iid gaussien, c'est à dire lorsque $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$.

Commençons par le théorème de Fisher qui stipule que :

$$\frac{(n-1)S^{*2}}{\sigma^2} \sim \chi^2_{n-1}. \tag{1}$$

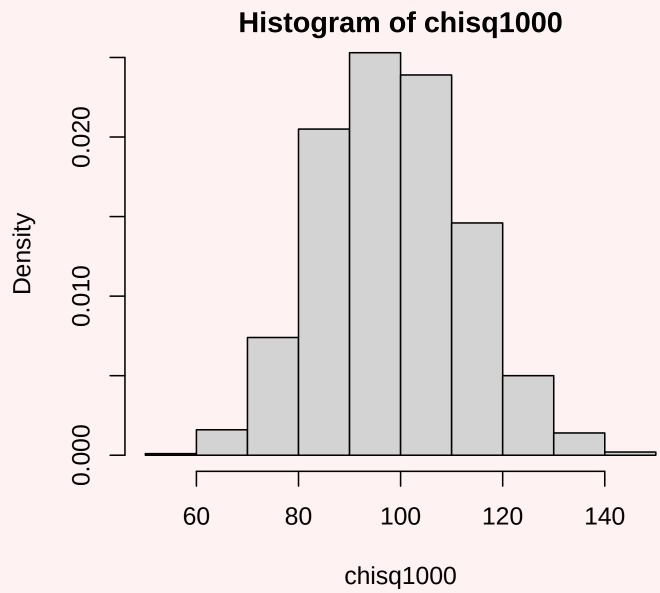
1 On suppose fixés μ, σ et n. Définir une fonction `chisq1` qui renvoie une seule observation de la variable aléatoire

$$\frac{(n-1)S^{*2}}{\sigma^2},$$

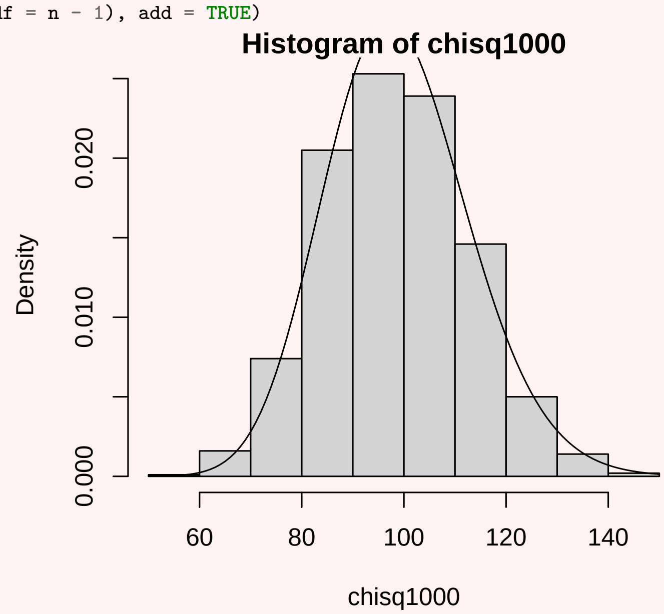
puis illustrer graphiquement la relation (1) (cf TP3).

```
n <- 100
mu <- 3
sigma <- 2
chisq1 <- function() {
    x <- rnorm(n, mean = mu, sd = sigma)
    (n - 1) * sd(x)^2/(sigma^2)
}
```

```
chisq1000 <- replicate(1000, chisq1())
hist(chisq1000, freq = FALSE)
```




```r
hist(chisq1000, freq = FALSE)
curve(dchisq(x, df = n - 1), add = TRUE)
```

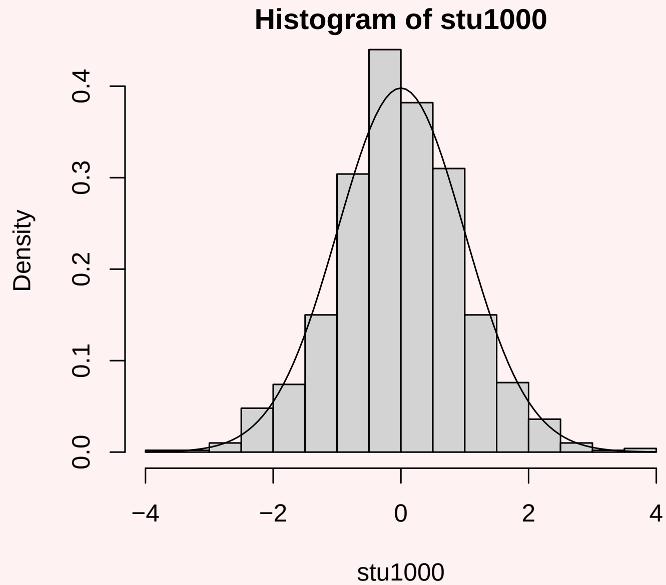


(2) Réaliser le même travail pour illustrer le fait que,

$$\frac{\overline{X} - \mu}{S^*/\sqrt{n}} \sim \mathcal{T}_{n-1}.$$

```r
n <- 100
mu <- 3
sigma <- 2
stu1 <- function() {
    x <- rnorm(n, mean = mu, sd = sigma)
    (mean(x) - mu)/(sd(x)/sqrt(n))
}
```

```
stu1000 <- replicate(1000, stu1())
hist(stu1000, freq = FALSE)
curve(dt(x, df = n - 1), add = TRUE)
```



2 Intervalles de confiance

Dans cette section, nous allons illustrer la notion de "niveau de confiance" d'un intervalle de confiance. Comme dans la section précédente, notre modèle est un n-échantillon iid gaussien, i.e., $X_1, \dots, X_n \overset{\text{iid}}{\sim} X$, avec $X \sim \mathcal{N}(\mu, \sigma^2)$ et nous considérons les cas σ^2 connu et σ^2 inconnu.

3 (σ^2 connu) Après avoir rappelé l'expression de l'intervalle de confiance bilatéral au niveau $1 - \alpha$ sur l'espérance μ d'une variable aléatoire qui suit une loi normale de variance connue σ^2, générer un échantillon de taille n selon une loi normale avec les paramètres de votre choix et donner une réalisation de cet intervalle de confiance ; vérifier la cohérence de votre calcul par rapport au paramètre choisi μ.

Soit $X_1, \dots, X_n$ n variables aléatoires iid de loi parente une loi normale de paramètre μ inconnu et σ^2 connu. L'intervalle de confiance au niveau α pour le paramètre μ s'écrit alors :

$$\left[\overline{X} - \frac{\sigma}{\sqrt{n}} u_{1-\frac{\alpha}{2}}, \overline{X} + \frac{\sigma}{\sqrt{n}} u_{1-\frac{\alpha}{2}}\right]$$

```
n <- 100
x <- rnorm(n, mean = 42, sd = pi)
alpha <- 0.05
mean(x) + c(-1, 1) * qnorm(1 - alpha/2) * pi/sqrt(n)
[1] 41.38146 42.61294
```

On obtient une réalisation de l'intervalle qui contient (avec une probabilité $1 - \alpha$) le paramètre choisi.

On suppose dans la suite de la section que le paramètre σ n'est plus connu. On ne peut donc pas s'en servir dans l'expression de l'intervalle de confiance.

(4) Donner l'expression de l'intervalle de confiance lorsque σ n'est pas connu et calculer une réalisation de l'intervalle de confiance avec les observations précédentes. Retrouver cet intervalle en se servant de la fonction de test que l'on verra prochainement :

```
t.test(x, conf.level = 1 - alpha)$conf.int
```

$$\left[\overline{X} - \frac{S^*}{\sqrt{n}} t_{n-1;1-\frac{\alpha}{2}}, \overline{X} + \frac{S^*}{\sqrt{n}} t_{n-1;1-\frac{\alpha}{2}}\right]$$

```
n <- 100
alpha <- 0.05
x <- rnorm(n)
mean(x) + c(-1, 1) * qt(1 - alpha/2, df = n - 1) * sd(x)/sqrt(n)
[1] -0.1459436  0.2189071
t.test(x, conf.level = 1 - alpha)$conf.int
[1] -0.1459436  0.2189071
attr(,"conf.level")
[1] 0.95
```

(5) Créer une fonction `gen_IC` qui prend en argument un échantillon `x` de taille quelconque et un niveau de signification α et renvoie l'intervalle de confiance sur l'espérance sous forme d'un vecteur de longueur 2 contenant la borne inférieure et la borne supérieure de l'intervalle.

```
gen_IC <- function(x, alpha) {
    n <- length(x)
    mean(x) + c(-1, 1) * qt(1 - alpha/2, df = n - 1) * sd(x)/sqrt(n)
}
```

(6) Utiliser la fonction `replicate` pour créer plusieurs intervalles de confiance. Que contient le résultat de `replicate` ?

```
param <- 3
alpha <- 0.05
ICs <- replicate(100, gen_IC(rnorm(100, mean = param), alpha))
```

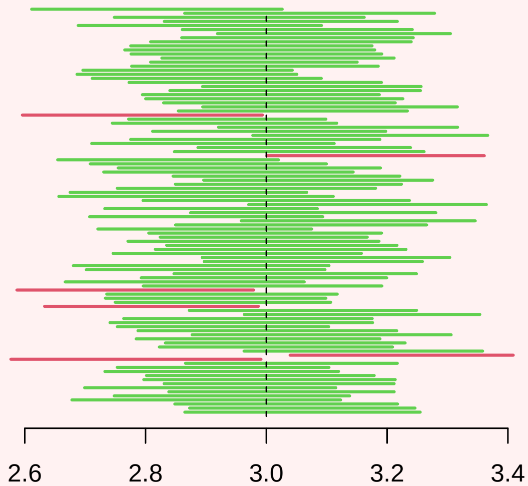
Les intervalles de confiance sont stockés dans les colonnes d'une matrice.

Pour visualiser ces intervalles de confiance, la fonction `plot_ICs` est mise à votre disposition dans le fichier `utils.R` présent dans le sous-dossier `src/`. Pour l'utiliser, il faut charger les définitions présentes de le fichier avec la commande

```
source("src/utils.R")
```

(7) Visualiser les intervalles de confiance avec la fonction `plot_ICs`. Quelle est la relation entre le niveau de l'intervalle de confiance (95%), le nombre d'intervalles verts et le nombre total d'intervalles ?

```
source("src/utils.R")
plot_ICs(ICs, param)
```

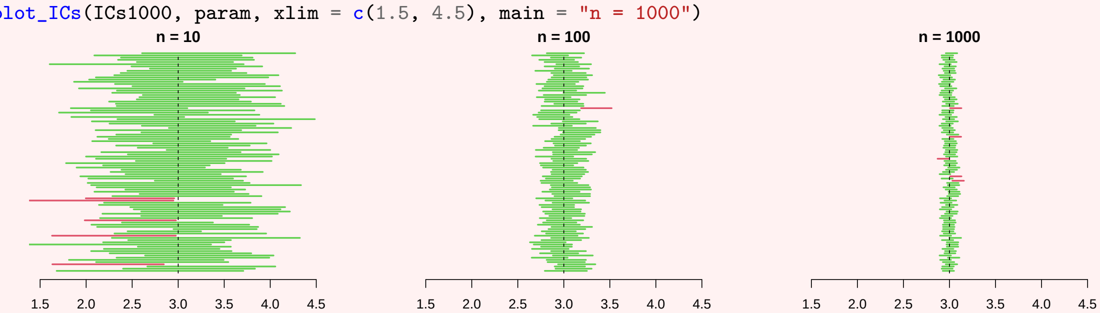


Les intervalles de confiance sont représentés par des segments horizontaux. On voit que la plupart d'entre eux (environ 95%, le niveau du test), dessinés en vert, contiennent le paramètre inconnu $\mu = 3$ en abscisse. Les autres en rouge échouent à encadrer le paramètre inconnu.
On a la relation

$$\frac{\text{nb d'intervalles vert}}{\text{nb d'intervalles total}} \simeq 95\%$$

(8) Étudier l'influence de n sur la largeur moyenne de l'intervalle de confiance. Pour fixer l'échelle des abscisses et éviter que `R` décide, on pourra utiliser le paramètre optionnel `xlim` de la fonction `plot_ICs`.

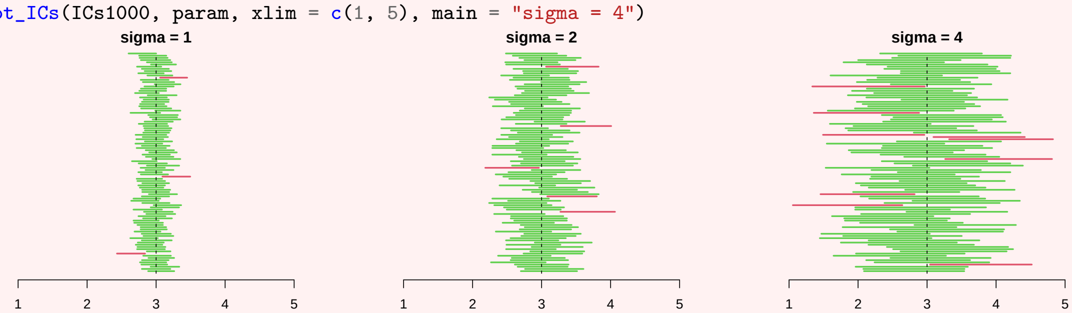
```
alpha <- 0.05
ICs10 <- replicate(100, gen_IC(rnorm(10, mean = param), alpha))
ICs100 <- replicate(100, gen_IC(rnorm(100, mean = param), alpha))
ICs1000 <- replicate(100, gen_IC(rnorm(1000, mean = param), alpha))
plot_ICs(ICs10, param, xlim = c(1.5, 4.5), main = "n = 10")
plot_ICs(ICs100, param, xlim = c(1.5, 4.5), main = "n = 100")
plot_ICs(ICs1000, param, xlim = c(1.5, 4.5), main = "n = 1000")
```



On observe que lorsque n augmente, la largeur moyenne des réalisations des intervalles de confiance diminue. En effet, plus n augmente, plus on dispose d'information et plus l'intervalle est précis.

(9) Étudier l'influence de la dispersion de l'échantillon sur la largeur moyenne de l'intervalle de confiance. Pour fixer l'échelle des abscisses, on pourra utiliser le paramètre optionnel `xlim` de la fonction `plot_ICs`.

```
alpha <- 0.05
ICs10 <- replicate(100, gen_IC(rnorm(100, mean = param, sd = 1), alpha))
ICs100 <- replicate(100, gen_IC(rnorm(100, mean = param, sd = 2), alpha))
ICs1000 <- replicate(100, gen_IC(rnorm(100, mean = param, sd = 4), alpha))
plot_ICs(ICs10, param, xlim = c(1, 5), main = "sigma = 1")
plot_ICs(ICs100, param, xlim = c(1, 5), main = "sigma = 2")
plot_ICs(ICs1000, param, xlim = c(1, 5), main = "sigma = 4")
```



On observe que lorsque la dispersion de la gaussienne augmente, la largeur moyenne des réalisations des intervalles de confiance augmente. En effet, lorsque la dispersion augmente, il est plus difficile d'estimer l'espérance. L'intervalle de confiance est donc plus large.

Pour illustrer plus précisément la relation établie à la question 7 avec un nombre quelconque d'intervalles de confiance, on va créer une fonction qui génère un échantillon, calcule l'intervalle de confiance associé et renvoie **TRUE** si l'intervalle contient le paramètre.

10 Créer cette fonction et calculer la proportion d'intervalles qui contient le paramètre (appelé le taux de recouvrement). Commenter le résultat.

```
hit <- function(n, param, alpha) {
    x <- rnorm(n, mean = param)
    IC <- gen_IC(x, alpha)
    param >= IC[1] & param <= IC[2]
}
n <- 100
alpha <- 0.05
hm <- replicate(10000, hit(n, 3, alpha))
mean(hm)
[1] 0.9491
```

3 Lemme de Slutsky

Le but de cette section est d'illustrer l'application du lemme de Slutsky lors de la recherche d'un intervalle de confiance asymptotique sur la proportion p dans un modèle binomial $B(n, p)$. D'après le polycopié de cours, on a la convergence en loi suivante,

$$\frac{\widehat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1). \tag{2}$$

Il est difficile d'extraire p de l'expression précédente. On choisit donc d'utiliser le lemme de Slutsky. On trouve alors l'intervalle asymptotique suivant,

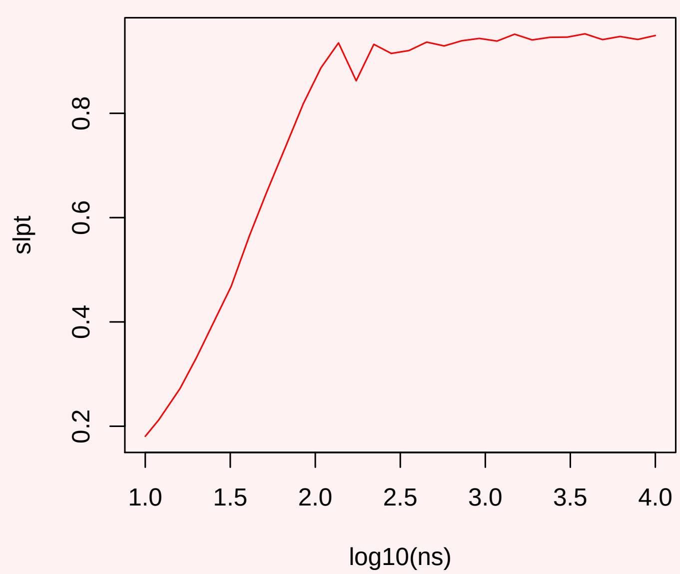
$$IC = \left[\widehat{p} - u_{1-\alpha/2}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}; \widehat{p} + u_{1-\alpha/2}\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}\right]. \tag{3}$$

(11) Écrire une fonction qui prend en argument la proportion recherchée p, la longueur de l'échantillon n, le nombre de fois k où on réitère l'expérience et le niveau $1-\alpha$ des intervalles de confiance et renvoie la proportion de réalisations des intervalles (3) qui contiennent le paramètre p parmi les k expériences.

```
slutsky <- function(p, n, k, alpha) {
    sim <- function() {
        x <- rbinom(n, 1, p)
        phat <- mean(x)
        IC <- phat + c(-1, 1) * qnorm(1 - alpha/2) * sqrt(phat * (1 - phat)/n)
        p >= IC[1] & p <= IC[2]
    }
    mean(replicate(k, sim()))
}
```

(12) Tracer cette proportion en fonction de n. On pourra utiliser une échelle logarithmique pour n.

```
p <- 0.02
k <- 10000
alpha <- 0.05
ns <- floor(10^seq(1, 4, length.out = 30))                    # 30 points en échelle logarithmique
slpt <- sapply(ns, function(n) slutsky(p, n, k, alpha))
plot(log10(ns), slpt, type = "l", col = "red")
```



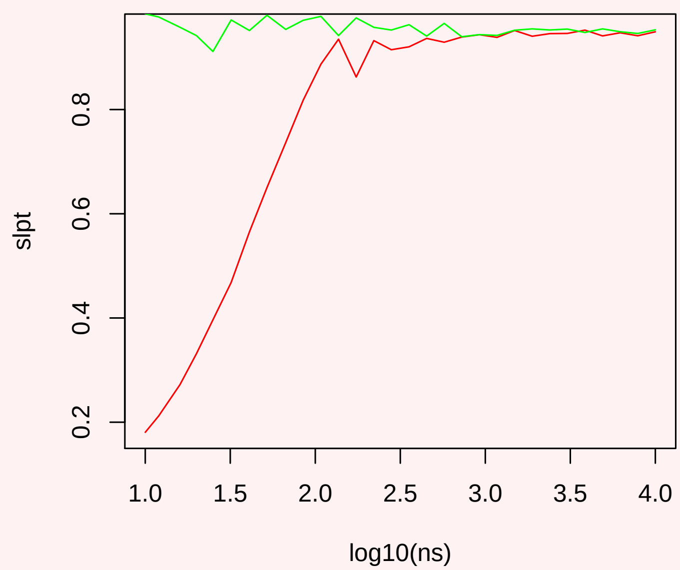
On observe bien que la probabilité que l'intervalle de confiance contienne le paramètre tend vers $1-\alpha$. Vers $n = 10^3$, la convergence commence à être satisfaisante.

Le calcul de l'intervalle de confiance directement issu de (2) sans utiliser le lemme de Slutsky est possible. On trouve

$$IC = \left[\frac{2n\widehat{p} + u_{1-\alpha/2}^2 \pm u_{1-\alpha/2}\sqrt{u_{1-\alpha/2}^2 + 4n\widehat{p}(1-\widehat{p})}}{2n + 2u_{1-\alpha/2}^2}\right].$$

(13) Écrire la même fonction que précédemment avec ce nouvel intervalle de confiance et comparer. Qu'en concluez-vous ?

```
noslutsky <- function(p, n, k, alpha) {
    sim <- function() {
        x <- rbinom(n, 1, p)
        phat <- mean(x)
        u <- qnorm(1 - alpha/2)
        IC <- (2 * n * phat + u^2 + c(-1, 1) * u * sqrt(u^2 + 4 * n * phat * (1 - phat)))/(2 * n + 2
        ↪  * u^2)
        p >= IC[1] & p <= IC[2]
    }
    mean(replicate(k, sim()))
}
nslpt <- sapply(ns, function(n) noslutsky(p, n, k, alpha))
plot(log10(ns), slpt, type = "l", col = "red")
lines(log10(ns), nslpt, col = "green")
```



On observe que la convergence vers $1-\alpha$ est beaucoup plus rapide sans utilisation du lemme de Slutsky. L'utilisation du lemme de Slutsky dégrade considérablement la qualité de l'intervalle de confiance dès que n est inférieur à ≈ 100.

On remarquera que la vitesse de convergence dépend de p. Plus p prend des valeurs extrêmes (proche de 0 ou 1) plus la convergence avec le lemme de Slutsky est lente.