TD2 : Normalité d'une variable
Estimation de la puissance d'un test
Test de Fischer

BIO5XX - BIOSTATISTIQUE L3

21 septembre 2016

Objectifs de la séance
Réflexions sur les tests de normalité, Test de Fischer

1 Test de normalité : suite

1.1 Génération d''échantillions aléatoires

R permet de simuler des échantillons numériques en utilisant un générateur de nombres aléatoires. Les générateurs aléatoire proposés dans *R* produisent des nombres dont la distribution suit une loi de distribution déterminée. Il est ainsi possible de générer des échantillons suivant par exemple une loi normale ou une loi uniforme. Les fonctions de génération d'échantillons aléatoires ont toute un nom commençant par «r» et se terminant par une abréviation du nom de la loi. Par exemple :

- rnorm pour la loi normale.
- runif pour la loi uniforme.
- rexp pour la loi exponentielle.

D'autres lois sont disponibles. Chacune de ces fonctions accepte comme premier paramètre la taille de l'échantillon à construire. Des paramètres supplémentaires correspondant aux paramètres de la loi de distribution peuvent être rajoutés comme la moyenne et l'écart type pour une loi normale. Pour des informations complémentaires concernant ces fonctions vous pouvez consulter l'aide de *R* en utilisant la commande *help*(*rnorm*) par exemple.

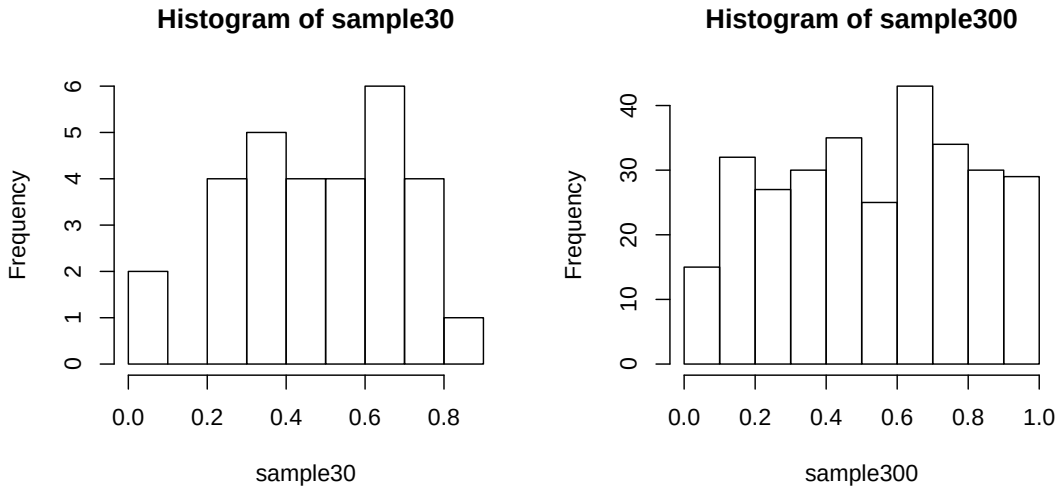
Exemple R - 1 :

Simulation d'un échantillon aléatoire de taille 30 puis de taille 300 suivant une loi de distribution uniforme.

```
sample30 <- runif(30)
sample300 <- runif(300)
sample30
```

```
##  [1] 0.73881433 0.54906313 0.48812867 0.68084468 0.06744743 0.50836045
##  [7] 0.75874227 0.46980454 0.77089733 0.67490089 0.44411849 0.67157975
## [13] 0.24558798 0.34226632 0.64066414 0.09439348 0.60618767 0.57561792
## [19] 0.20172391 0.34252334 0.22481968 0.64473207 0.49055208 0.38128944
## [25] 0.56211156 0.20588044 0.33456736 0.81410804 0.38088866 0.76689255
```

```
par(mfrow=c(1,2))
hist(sample30)
hist(sample300)
```




```
dev.copy(pdf,'sample-unif.pdf',width=12,height=6)

## pdf
##   3

dev.off()

## pdf
##   2
```

Essayer de générer des échantillons aléatoires suivant une loi normale avec la fonction *rnorm* ou une loi exponentielle à l'aide de la fonction *rexp*

La forme de l'histogramme des valeurs de l'échantillon est d'autant plus semblable à celle de la fonction de densité de la loi utilisée que la taille de l'échantillon est grand. Ceci est dû au biais d'échantillonnage et explique que les tests de normalité d'un échantillon sont d'autant plus puissants que la taille de l'échantillon augmente.

1.2 Comparaison de deux tests de normalité

Plusieurs tests existent pour tester la normalité d'un échantillon. *R* propose en standard le test de *Shapiro-Wilk* qui est réalisable par un appel à la fonction *shapiro.test*.

Exemple R - 2 :

Réalisation d'un test de *Shapiro-Wilk* sur les deux échantillons *sample*30 et *sample*300 construit précedement. Les deux dernières commandes montrent qu'il est possible d'obtenir uniquement une des valeurs associées au test, ici la p_{value}.

```
shapiro.test(sample30)

##
##  Shapiro-Wilk normality test
##
## data:  sample30
## W = 0.95865, p-value = 0.286

shapiro.test(sample300)

##
##  Shapiro-Wilk normality test
##
## data:  sample300
## W = 0.96125, p-value = 3.616e-07

shapiro.test(sample30)$p.value

## [1] 0.285957

shapiro.test(sample300)$p.value

## [1] 3.615507e-07
```

Dans la bibliothèque *nortest* plusieurs autres tests sont disponibles. Nous utiliserons le test nommé *pearson.test* qui ce base sur un test du χ^2 de conformité à une loi normale.

la fonction *pearson.test* accepte un argument *x* qui contient l'échantillon à tester. Elle retourne le même type d'information que la fonction *shapiro.test*.

Exemple R - 3 :

Calcul de la p_{value} associée aux tests de normalité des deux échantillons *sample*30 et *sample*300 par la méthode de *Shapiro-Wilk* et celle du test du χ^2 de conformité. La première commande indique à *R* de charger en mémoire la bibliothèque *nortest*.

```
#install.packages("nortest",
#                  repos="http://cran.r-project.org" )
library(nortest)
pearson.test(sample30)

##
##  Pearson chi-square normality test
##
## data:  sample30
## P = 3.6, p-value = 0.6083

pearson.test(sample300)

##
##  Pearson chi-square normality test
##
## data:  sample300
## P = 63.067, p-value = 3.249e-07
```

1.3 Signification de la p_{value} associé à un test

La p_{value} associée à un test est la probabilité de se tromper si l'on rejette l'hypothèse nulle H_0 testée alors que celle-ci est vrai. On parle du risque de première espèce α. Pour bien prendre conscience de cette signification, nous vous proposons de jouer avec.

Pour cela nous allons générer des échantillons aléatoires suivant une loi normale de moyenne $E = 1$ et d'écart type $\sigma = 1$ et tester la normalité de ces échantillons.

1.3.1 construction du générateur d'échantillons aléatoires

Pour permettre les comparaisons dans la suite du TD nous ne travaillerons qu'avec des échantillons de moyenne $\mu = 1$ et d'écart type $\sigma = 1$.

Exemple R - 4 :

Construction d'un générateur aléatoire de moyenne $\mu = 1$ et d'écart type $\sigma = 1$ suivant une loi normale.

```
normalea <- function(x) rnorm(x,1,1)
mean(normalea(10000))
```

```
## [1] 0.9964946
```

```
sd(normalea(10000))
```

```
## [1] 1.00259
```

1.3.2 Construction des fonctions de test

Pour ne récuperer que les P_{values} des tests de normalité de Shapiro et de Pearson, on construit deux fonction *shapiro.normtest.pvalue* et *pearson.normtest.pvalue*

Exemple R - 5 :

Construction de fonctions retournant pour un echantillon x la p_{value} des tests *shapiro.test* et *pearson.test*.

```
shapiro.normtest.pvalue <- function(x) shapiro.test(x)$p.value
pearson.normtest.pvalue <- function(x) pearson.test(x)$p.value
shapiro.normtest.pvalue(sample30)
```

```
## [1] 0.285957
```

```
pearson.normtest.pvalue(sample30)
```

```
## [1] 0.6083133
```

Exemple R - 6 :

La fonction *rep* permet de répéter une même valeur plusieurs fois. Ici la valeur 30 est répétée 20 fois. La fonction *mapply* permet de relancer une même fonction pour toute une série de parametres. La fonction *table* retourne pour chacune des modalités d'une variable discrète l'effectif associé.

```r
rep(30,20)
```

```
##  [1] 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
```

```r
mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20))
```

```
##  [1] 0.181062318 0.605268494 0.215155564 0.292264180 0.477525634
##  [6] 0.360785231 0.977850059 0.778160503 0.178318264 0.008814055
## [11] 0.568829147 0.903745833 0.079551953 0.581175546 0.932921320
## [16] 0.517041665 0.468832557 0.534262579 0.529683665 0.137774740
```

```r
mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20)) > 0.05
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE
```

```r
table(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20)) > 0.05)
```

```
##
## TRUE
##   20
```

```r
table(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20)) > 0.05)
```

```
##
## TRUE
##   20
```

```r
table(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20)) > 0.05)
```

```
##
## FALSE  TRUE
##     1    19
```

```r
table(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),rep(30,20)) > 0.05)
```

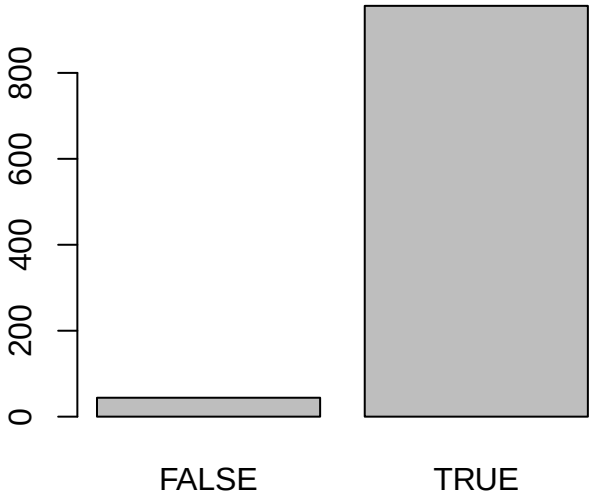
```
##
## FALSE  TRUE
##     1    19
```

Refaite le calcul du nombre de tests positifs et négatifs en passant :
— le nombre d'échantillons testés de 20 à 1000.
— la taille des échantillons de 30 à 300
— le test de shapiro à pearson

```r
rep = factor(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),
                    rep(30,1000)
                   ) > 0.05)
table(rep)
```

```
## rep
## FALSE  TRUE
##    44   956
```

```
plot(rep)
```




```
rep = factor(mapply(function(x) shapiro.normtest.pvalue(normalea(x)),
                    rep(300,1000)
                    ) > 0.05)
table(rep)

## rep
## FALSE  TRUE
##    47   953
```

```
rep = factor(mapply(function(x) pearson.normtest.pvalue(normalea(x)),
                    rep(30,1000)
                    ) > 0.05)
table(rep)

## rep
## FALSE  TRUE
##    56   944
```

```
rep = factor(mapply(function(x) pearson.normtest.pvalue(normalea(x)),
                    rep(300,1000)
```

```
                    ) > 0.05)
table(rep)

## rep
## FALSE  TRUE
##    56   944
```

1.4 Estimation de la puissance d'un test

La p_{value} associée à un test est la probabilité de se tromper si l'on rejette l'hypothèse nulle H_0 testée alors que celle-ci est vraie. Mais ce risque ne dit rien quant à l'acceptation d'H_0 . Si je ne peux pas rejeter H_0 quel est la probabilité que H_1 soit vrai ?

Nous pouvons estimer pour des échantillons où nous savons que H_1 est vrai la probabilité que le test ne permette pas de rejeter H_0. C'est à dire le risque d'accepter H_0 à tord. C'est le risque de seconde espèce β. La puissance d'un test est définie comme $1 - \beta$. C'est donc la probabilité de rejeter H_0 quand H_1 est vrai.

Pour estimer la puissance d'un test, le plus simple est de réaliser une série de tests sur des échantillons pour lesquels nous savons que H_0 est faux. Par exemple nous pouvons générer des échantillons aléatoires suivant une distribution uniforme ou une distribution exponentielle et tester s'ils suivent une loi normale.

Une loi uniforme est une loi de distribution où la probabilité de tirer n'importe quelle valeur entre deux valeurs extrêmes est équivalente. La probabilité de tirer une valeur hors de cette intervalle est nulle.

Une loi uniforme définie sur l'intervalle :

$$[E - \sigma\sqrt{3}, E + \sigma\sqrt{3}] \tag{1}$$

possède une moyenne égale à E et un écart type σ.

La loi exponentielle est définie par la fonction suivante

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & , \ x \geq 0, \\ 0 & , \ x < 0. \end{cases} \tag{2}$$

Sa moyenne et son écart type sont :

$$\mathbf{E}_x = \frac{1}{\lambda} \tag{3}$$

$$\sigma_x = \frac{1}{\lambda^2} \tag{4}$$

1.4.1 construction des générateurs d'échantillons aléatoires

Comme pour l'exercice précédant nous ne travaillerons qu'avec des échantillons de moyenne $\mu = 1$ et d'écart type $\sigma = 1$.

Exemple R - 7 :

Construction de deux générateurs aléatoires supplémentaires de moyenne $\mu = 1$ et d'écart type $\sigma = 1$ suivant respectivement une loi uniforme et exponentielle.

```
unifalea <- function(x) runif(x,1-sqrt(3),1+sqrt(3))
expoalea <- function(x) rexp(x,1)
mean(unifalea(10000))

## [1] 0.9983456

mean(unifalea(10000))

## [1] 0.996568

sd(unifalea(10000))

## [1] 1.000153

mean(expoalea(10000))

## [1] 1.001632

sd(expoalea(10000))

## [1] 0.9906399
```

1.4.2 Réalisation du test de puissance

Afin de tester la puissance des deux tests (*Shapiro-Wilk* et *Pearson*) en fonction de l'hypothèse alternative (loi uniforme ou loi exponentielle) et de la taille des échantillons testés (30 ou 300 individus) les commandes suivantes sont exécutées.

Exemple R - 8 :

On realise une serie de tests sur des jeux de données simulées sous l'hypothèse alternative H1

```
shap.unif.30 <- factor(mapply(function(x) shapiro.normtest.pvalue(unifalea(x)),
                       rep(30,1000)
                             ) > 0.05)
shap.unif.300 <- factor(mapply(function(x) shapiro.normtest.pvalue(unifalea(x)),
                        rep(300,1000)
                       ) > 0.05)

shap.expo.30 <- factor(mapply(function(x) shapiro.normtest.pvalue(expoalea(x)),
                       rep(30,1000)
                      ) > 0.05)
shap.expo.300 <- factor(mapply(function(x) shapiro.normtest.pvalue(expoalea(x)),
                        rep(300,1000)
                       ) > 0.05)

pear.unif.30 <- factor(mapply(function(x) pearson.normtest.pvalue(unifalea(x)),
                       rep(30,1000)
                             ) > 0.05)
pear.unif.300 <- factor(mapply(function(x) pearson.normtest.pvalue(unifalea(x)),
                        rep(300,1000)
                       ) > 0.05)
```

```
pear.expo.30 <- factor(mapply(function(x) pearson.normtest.pvalue(expoalea(x)),
                              rep(30,1000)
                             ) > 0.05)
pear.expo.300 <- factor(mapply(function(x) pearson.normtest.pvalue(expoalea(x)),
                               rep(300,1000)
                              ) > 0.05)
```

Exemple R - 9 :

Les taux de descision sont présentés sous forme de tables de contingeance et d'histogrammes

```
table(shap.unif.30)
```

```
## shap.unif.30
## FALSE  TRUE
##   363   637
```

```
table(shap.unif.300)
```

```
## shap.unif.300
## FALSE
##  1000
```

```
table(shap.expo.30)
```

```
## shap.expo.30
## FALSE  TRUE
##   972    28
```

```
table(shap.expo.300)
```

```
## shap.expo.300
## FALSE
##  1000
```

```
table(pear.unif.30)
```

```
## pear.unif.30
## FALSE  TRUE
##   103   897
```

```
table(pear.unif.300)
```

```
## pear.unif.300
## FALSE  TRUE
##   993     7
```

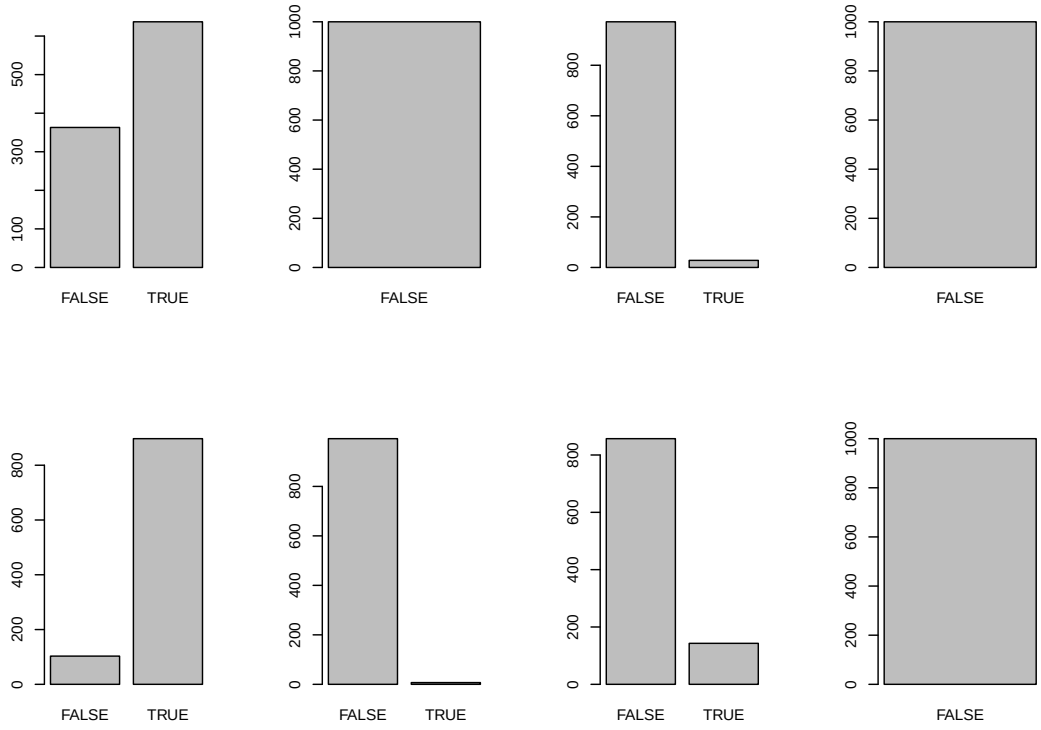
```
table(pear.expo.30)
```

```
## pear.expo.30
## FALSE  TRUE
##   857   143
```

```
table(pear.expo.300)
```

```
## pear.expo.300
## FALSE
##  1000
```

```r
par(mfrow=c(2,4))
plot(shap.unif.30)
plot(shap.unif.300)
plot(shap.expo.30)
plot(shap.expo.300)
plot(pear.unif.30)
plot(pear.unif.300)
plot(pear.expo.30)
plot(pear.expo.300)
```



Discuter l'impact du test, de la taille de l'échantillon et de l'hypothèse alternative (uniforme ou exponentielle) sur la puissance du test.