

Formation Statistiques - Concepts de base : exercices

Gilles San Martin - gilles.sanmartin@gmail.com

Septembre 2013

Contents

Exercice 1	2
Exercice 2	6

Le but de cette série d'exercices est de se familiariser avec les concepts de base de l'inférence statistique : Intervalles de confiance, erreur standard, tests d'hypothèse nulle.

On a fait le choix d'utiliser non seulement les approches classiques basées sur des distributions de probabilités théoriques mais aussi des méthodes plus récentes de rééchantillonnage par ordinateur (permutation, bootstrap) qui sont moins basées sur la théorie mathématique.

Exercice 1

Voici une variable "x" représentant le temps de survie d'insectes après la ponte.

- Tracez un histogramme de fréquence de ces valeurs, ensuite calculez la moyenne, la médiane, les quantiles à 2.5% et 97.5% (= 0.9125, 26.37) et l'écart type des données
- Estimez l'erreur standard et l'intervalle de confiance de la médiane avec la méthode du bootstrap vue pendant la partie théorique (NB : il n'existe pas de formule mathématique permettant de calculer ces valeurs comme pour la moyenne)
- Est-ce qu'on pourrait dans ce cas calculer un intervalle de confiance sur la moyenne avec la méthode paramétrique (en utilisant l'erreur standard divisée par la racine carrée du nombre d'observations) ?
- transformez la variable au moyen du logarithme népérien (fonction log). Tracez un histogramme de cette nouvelle variable et calculez la moyenne et la médiane.
- Calculez l'erreur standard (0.121194) et l'intervalle de confiance (1.2707 - 1.798) de la moyenne de cette nouvelle variable avec la méthode paramétrique. Pourquoi peut-on le faire dans ce cas ? Comparez avec les mêmes valeurs obtenues par bootstrap.

```
set.seed(123)
x <- round(exp(rnorm(50, 1.5, 1)),1)
x
```

```
## [1] 2.6 3.6 21.3 4.8 5.1 24.9 7.1 1.3 2.3 2.9 15.2 6.4 6.7 5.0
## [15] 2.6 26.8 7.4 0.6 9.0 2.8 1.5 3.6 1.6 2.2 2.4 0.8 10.4 5.2
## [29] 1.4 15.7 6.9 3.3 11.0 10.8 10.2 8.9 7.8 4.2 3.3 3.1 2.2 3.6
## [43] 1.3 39.2 15.0 1.5 3.0 2.8 9.8 4.1
```

```
mean(x)
```

```
## [1] 7.104
```

```
median(x)
```

```
## [1] 4.15
```

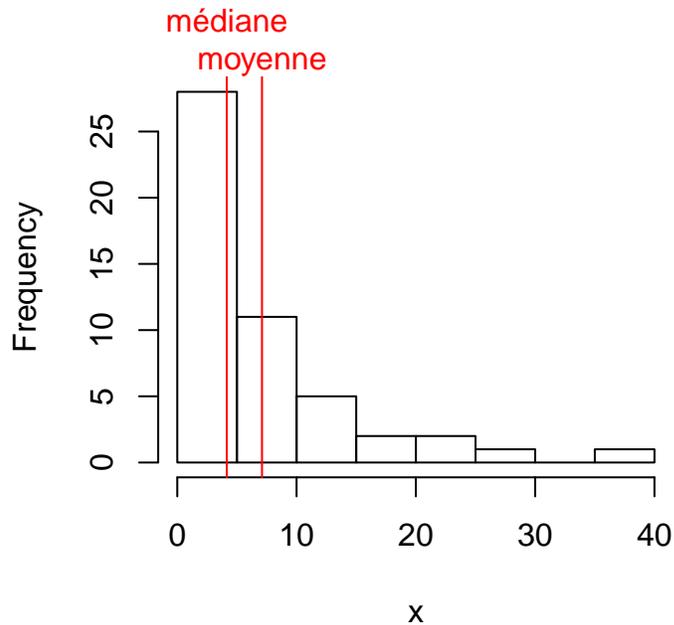
```
quantile(x, probs = c(0.025, 0.975))
```

```
## 2.5% 97.5%
## 0.9125 26.3725
```

```
sd(x)
```

```
## [1] 7.559126
```

```
# graphique
hist(x, main = "")
abline(v = mean(x), col = "red")
mtext(text="moyenne", side=3, line= 0, at=mean(x), col = "red")
abline(v = median(x), col = "red")
mtext(text="médiane", side=3, line= 1, at=median(x), col = "red")
```



```
# bootstrap
boot <- replicate(1000, median(sample(x, replace = TRUE)))
sd(boot)
```

```
## [1] 0.9501914
```

```
quantile(boot, probs = c(0.025, 0.975))
```

```
## 2.5% 97.5%
## 3.1 6.9
```

```
# On ne peut clairement pas utiliser la méthode paramétrique ici pour estimer l'intervalle
# de confiance sur la moyenne car cette méthode suppose que la population est distribuée
# selon une loi normale ce qui n'est clairement pas le cas ici
```

```
# transformation
```

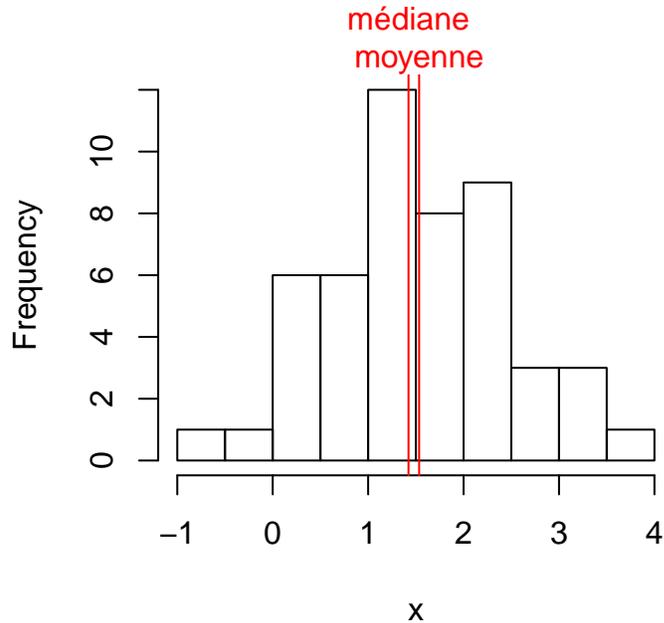
```
x <- log(x)
mean(x)
```

```
## [1] 1.534354
```

```
median(x)
```

```
## [1] 1.423036
```

```
hist(x, main = "")
abline(v = mean(x), col = "red")
mtext(text="moyenne", side=3, line= 0, at=mean(x), col = "red")
abline(v = median(x), col = "red")
mtext(text="médiane", side=3, line= 1, at=median(x), col = "red")
```



```
# se et CI de la variable transformée
t.test(x)# fourni l'intervalle de confiance
```

```
##
## One Sample t-test
##
## data: x
## t = 11.695, df = 49, p-value = 8.66e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 1.270708 1.798000
## sample estimates:
## mean of x
## 1.534354
```

```
(se.x <- sd(x)/sqrt(length(x))) # erreur standard de la moyenne
```

```
## [1] 0.1311948
```

```
mean(x) + qt(p=0.025, df = length(x)-1) * se.x
```

```
## [1] 1.270708
```

```
mean(x) + qt(p=0.975, df = length(x)-1) * se.x
```

```
## [1] 1.798
```

```
# bootstrap
boot <- replicate(1000, mean(sample(x, replace = TRUE)))
sd(boot)
```

```
## [1] 0.1313116
```

```
quantile(boot, probs = c(0.025, 0.975))
```

```
##      2.5%    97.5%
```

```
## 1.279991 1.796112
```

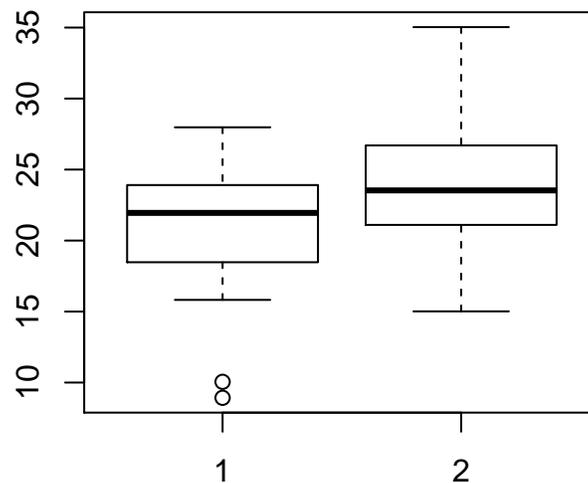
Exercice 2

Voici deux variables contenant la production laitière de 25 vaches contrôles et 25 vaches traitées.

- Faites une représentation graphique des données (par exemple un boxplot)
- Utilisez la fonction de R pour réaliser un test de student comparant les moyennes de ces deux groupes. Utilisez l'option `var.equal = TRUE` (les variances des deux groupes sont considérées comme égales). Comment interprétez-vous les résultats? ($p = 0.02092$)
- Recalculez vous-même les valeurs de t , les degrés de liberté et la p -valeur. La formule du test de student comparant 2 moyennes se trouve dans la présentation, diapo 52.
- recalculez la p -valeur de ce test par permutation

```
n <- 25
set.seed(1)
control <- abs(rnorm(n, 20, 5))
set.seed(12)
treatment <- abs(rnorm(n, 25,5))
```

```
# graphique
boxplot(control, treatment)
```



```
# test de student
t.test(control, treatment, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: control and treatment
## t = -2.3881, df = 48, p-value = 0.02092
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
## -5.921623 -0.508099
## sample estimates:
## mean of x mean of y
## 20.84333 24.05819
```

```
# Calcul à la main
```

```
(tstat <- (mean(control) - mean(treatment))/ sqrt((var(control)/length(control)) +
                                                    (var(treatment)/length(treatment))))
```

```
## [1] -2.38806
```

```
(ddl <- length(control) + length(treatment) - 2)
```

```
## [1] 48
```

```
(p <- 2*pt(tstat, ddl, lower.tail=TRUE))
```

```
## [1] 0.02092059
```

```
# test par permutation - approche 1 avec la fonction replicate
```

```
d <- c(control, treatment)
group <- as.factor(c(rep("control", length(control)), rep("treatment", length(control))))
```

```
perm <- replicate (999, t.test( d ~ sample(group, replace = FALSE), var.equal=TRUE)$statistic)
perm <- c(perm, tstat)
sum(abs(perm) >= abs(tstat))/1000
```

```
## [1] 0.018
```

```
# test par permutation - approche 2 avec une boucle
```

```
d <- c(control, treatment)
```

```
perm <- vector(length = 1000)
for (i in 1: 999) {
  d <- sample(c(control, treatment))
  permcontrol <- d[1:length(control)]
  permtreatment <- d[length(control):length(d)]
  perm [i] <- t.test(permcontrol, permtreatment, var.equal = FALSE)$statistic
}
perm <- c(perm, tstat)
sum(abs(perm) >= abs(tstat))/1000
```

```
## [1] 0.014
```