TP Statistiques 1

Aurélien Enfroy, Tram Ngo, Thuy Vo

24 janvier 2025

Introduction à R

`R` (https://www.r-project.org/) est l'un des langages et environnements les plus connus pour le calcul statistique et les graphiques. Il est gratuit et il est très utilisé comme une langue éducative et comme un outil de recherche (similaire à Matlab et à Python).

Quelques sources pour l'installation et l'apprentissage à la maison

- Pour un cours interactif, suivez https://swirlstats.com/students.html.
- Une introduction courte à R (12p): https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf
- R `manuals`: https://cran.r-project.org/manuals.html (en anglais)
- Plus de ressources et en français: `contributed documents`: https://cran.r-project.org/other-docs.html

R fondamentaux

Tout d'abord, nous pouvons utiliser `R` comme une calculatrice avancée.

Operation scalaire

```
x = 3
sin(2*pi*x^2)/5
?Arithmetic
```

Ex1. Calculer le volume d'une sphère de radiux 2.5.

Operation des vecteurs

```
## Créer un vecteur en R
x = c(1, 2, 3, 4, 5)
## Pour acceder à un élément de vecteur x
x[3]
## Addition : faire la somme des éléments
sum_x <- x[1] + x[2] + x[3] + x[4] + x[5]
sum_x
## Vecteurs des charactères
petit_dejeuner <- c("croissant", "pain","cereales", "lait", "jus d'orange",
                    "café", "chocolat chaud")
print(petit_dejeuner)
```

Ex2. Compléter le code `ma_commande <-...` pour créer un vecteur qui contient `céréales`, `café` et `lait`.

```r
ma_commande <-
print(ma_commande)
```

Mes tâches multiples et les répétitions

Les boucles peuvent être créées par `for` ou `while` s'il y a une condition à vérifier.

Ex3. Compléter le code suivant qui calcule la somme des carrés de x.

```r
x <- seq(3, 20, by=4) ## x est une séquence des éléments variant de 3 à 20 avec le pas entre les
sum_x <- 0
n <- length(x)
for (i in 1:n){

}
```

Fonctions

Une fonction est utile lorsque l'on souhaite effectuer plusieurs fois des opérations similaires.

```r
my_smean <- function(x)
{
  sum_x <- 0
  n <- length(x)
  for (i in 1:n){
    sum_x <- sum_x + x[i]
  }
  return(sum_x/n)   ## output
}
my_smean(x) ## fontion my_smean() évaluée en x
```

Ex4. écrire une fonction (`my_stdev`) pour calculer l'écart-type d'une séquence de nombres.

```r
my_stdev <- function(x)
{
  ## x - vector
}
```

Simuler des données

`R` comprend des fonctions qui créent des échantillons aléatoires à partir de nombreuses familles de distribution standard.

```r
x <- rnorm(30, 0, 0.5)
y <- runif(25, -2, 2)
```

Ex5. À l'aide de `help()` trouver ce qu'est le modèle de simulation pour x et y.

Visualisation des données

```r
plot(x); barplot(x); boxplot(x); hist(x)
```

Ex6. Quelles caractéristiques de distribution ces graphiques vous ont-ils permis de dégager ? Essayer avec y et résumez vos résultats.

Ex7. Modifier l'histogramme avec des ruptures entre -2 et 2 avec 10 intervalles (ou bacs) égaux.

Statistiques numériques récapitulatives

Des statistiques sommaires simples peuvent être calculées facilement à l'aide de fonctions intégrées dans R.

```
c(mean(x), var(x), sd(x), min(x), max(x), median(x))
```

Ex8. Expliquer ce que fait chaque fonction. Obtenez la moyenne et l'écart-type de l'échantillon pour x et y. Sont-ils proches des valeurs réelles ?

Attention : la fonction `var()` en R calcule la variance empirique corrigée.

Structure de données

Vecteurs et matrices

```
xmat = matrix(data=c(3,7, 2, 9, 1, 5), ncol=2)
xmat
xmat[,2]
```

Ex9. Quelle est la dimension de `xmat` (?dim) ? Quel est le (2,2)ème élément de la matrice ?

Data frame

Un `DataFrame` est une structure de données bidimensionnelle, c'est-à-dire que les données sont alignées de façon tabulaire en lignes et en colonnes. Contrairement à la matrice, le `DataFrame` de données peut contenir des éléments non numériques.

```
mydf = data.frame(x=runif(10), y=rnorm(10), z=rep(c('f','m'), 5))
mydf
mydf$y
mydf["z"]
str(mydf)
head(mydf)
```

Ex10. Faire un boxplot (boîte à moustache) pour $y|z = f$ et $y|z = m$ séparément (`?boxplot`).

Ex11. Simuler deux échantillons aléatoires de taille 50 à partir de $X \sim \text{Bern}(p = 0.4)$ et $Y \sim \text{Poisson}(\lambda = 2.5)$. Créez un `data frame` appelé `mydf_sim` et imprimer les 5 premières lignes.

Lists

Un `list` est une collection de colonnes, mais contrairement à la matrice ou le DataFrame, elles n'ont pas besoin d'avoir la même longueur.

```
xlist = list(a=1, b=c(2,3), c=seq(0,1,length=5))
xlist
names(xlist)
xlist$b
```

Opérations multiples efficaces

Une boucle plus efficace peut être mise en œuvre à l'aide de fonctions R telles que `apply()` et `lapply()`, `tapply()`, `sapply()`.

```
apply(xmat, 2, mean) ## compute sample mean for each column
lapply(1:n, function(i) {i^2})
```

Ex12. Réviser la fonction (`my_stdev`) sans `for`.

Visualisation avancée

Avec R, nous pouvons rendre les figures plus belles et plus informatives, en utilisant le module `ggplot2`.

```
install.packages("ggplot2") ## install the package
library(ggplot2) ## load the package
?mpg ## dataset
ggplot(data=mpg,
       mapping = aes(x=class, y=hwy)) +
  geom_boxplot()
```

Ex13. Combien y a-t-il de variables dans le data frame `mpg` ? Que montre le graphique ?