TP_R_Intro: Prise en main de **R** et **RStudio**, création de notebooks

Pavlo Mozharovskyi (Télécom Paris) - Parcours Data Science BPCE

Contents

1 Tutoriel: utilisation de R et Rstudio **1**
1.1 Utilisation basique de **RStudio** 2
1.2 Kit de survie en R 3
1.3 Pour aller plus loin 7

2 Exercice: Théorème Central Limite **7**
2.1 Loi des grands nombres 7
2.2 Théorème central limite 8

3 Exercice : Import et visualisation de données. **9**
3.1 Vitesse de la lumière 9
3.2 Réchauffement climatique 9

Ce TP a pour objectif de vous familiariser avec l'utilisation du logiciel R et de vous donner quelques outils simples permettant de rédiger un rapport dans le domaine des statistiques et de l'apprentissage sous forme de notebook.

On ne vous demande pas de rendre de rapport pour ce TP ; il n'est pas noté, mais destiné à faciliter le démarrage du mini-projet de mi-parcours (courant octobre). **Le jour du démarrage, le contenu de ce document sera considéré comme acquis.**

---

1 Tutoriel: utilisation de R et Rstudio

##Installations

*Si vous travaillez sur les machines de l'école, les logiciels **R** et **Rstudio** sont déjà installés ; vous pouvez donc passer au paragraphe suivant concernant* rmarkdown.

1.0.1 Commencez par installer R et sur votre ordinateur

(Uniquement si vous travaillez sur votre ordinateur personnel)

R est un logiciel libre contenant un grand nombre de librairies ("packages") particulièrement adaptées à l'analyse statistique des données.
Visitez le site https://cran.r-project.org/ pour un aperçu et suivez pas à pas les instructions d'installation. Prenez garde de suivre la procédure adaptée à votre distribution. Si vous avez le choix, il est beaucoup plus pratique de travailler sous linux (par exemple Ubuntu). Il vous sera demandé de choisir un site miroir depuis lequel les librairies seront téléchargées. Vous pouvez par exemple choisir le miroir http://cran.univ-lyon1.fr/.

1.0.2 Installez ensuite Rstudio

Rstudio est une interface graphique très pratique pour débuter avec **R**. Visitez le site https://www.rstudio.com/ pour avoir un aperçu des fonctionnalités de cet outil. Téléchargez l'"installer" qui correspond à votre distribution (windows, linux, mac ...) ici: https://www.rstudio.com/products/rstudio/download/ et installez-le.

1.0.3 Installez le package *rmarkdown*

Ouvrez ensuite **Rstudio** et installez le package *rmarkdown* (un outil de création de document interfaçant texte, code et symboles mathématiques/équations): pour cela, tapez dans la console de **Rstudio** :

```
install.packages("rmarkdown")
```

1.1 Utilisation basique de RStudio

La console **R** (par défaut en bas à gauche) permet d'interagir directement avec **R**. Tapez 1+1 puis "Entrée" dans la console :

```
1 + 1
```

Pour charger le jeu de données "iris" (un jeu de données disponible par défaut dans R sous forme de matrice), tapez maintenant (toujours dans la console) :

```
data("iris")
```

Pour connaître les dimensions du jeu de données, et le nom des colonnes :

```
dim(iris) ; names(iris)
```

```
## [1] 150   5
```

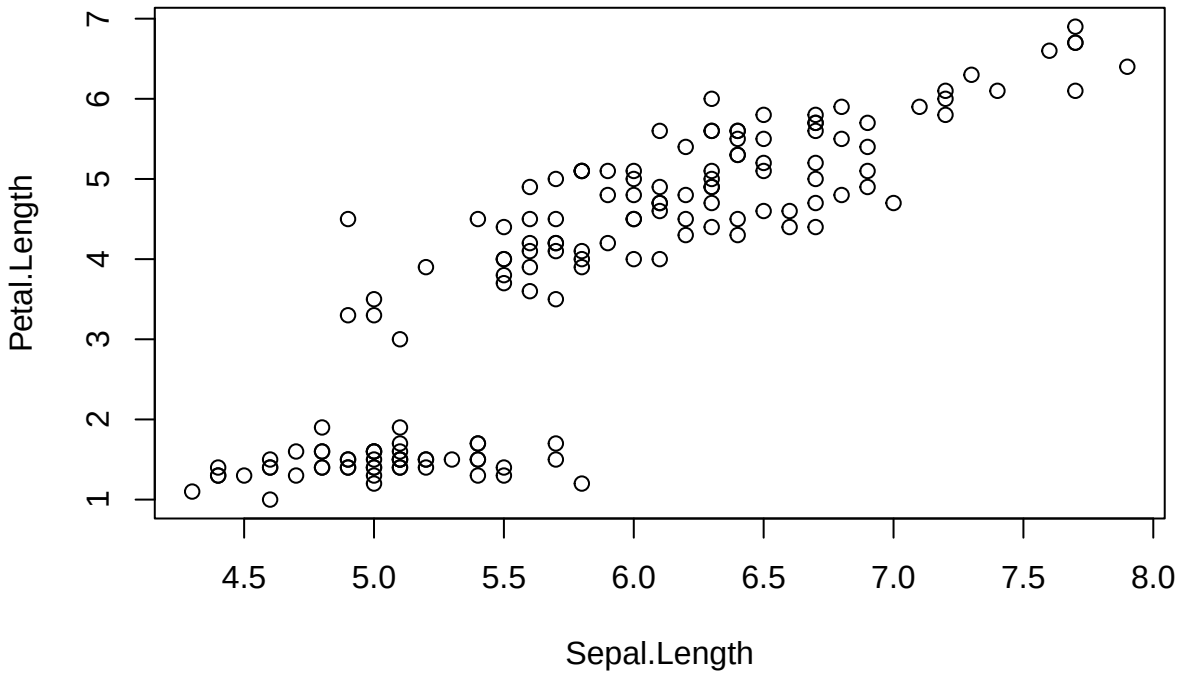
```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```
names(iris)[1]
```

```
## [1] "Sepal.Length"
```

puis, pour afficher le nuage de points formé par les colonnes 1 et 3 :

```
i = 1
j = 3
plot(iris[,i],iris[,j],xlab = names(iris)[i],ylab = names(iris)[j])
```



La figure devrait s'afficher dans la fenêtre "Plots".

1.1.1 Travailler dans un fichier script

Cette étape est indispensable pour faire des choses plus intéressantes. Dans **RStudio**, créez un script (un fichier de type "`monfichierScript.R`") en cliquant sur : `menu -> File -> new File -> R Script`.

Dans la fenêtre du script, écrivez par exemple :

```
print("exemple de calcul");
pommes = 3 ;
poires = 8;
resultat = pommes + poires ;
resultat
## Ceci est une ligne de  commentaire
#' ceci est une autre ligne de commentaire. Quelle différence ?
```

Ensuite, sauvez (`Ctrl+S`) puis exécutez le script (`Ctrl+shift+Entrée` ou bouton `Source -> Source with echo`). Observez le résultat dans la console **R**. Les figures éventuelles (il n'y en a pas ici) s'ouvrent dans la fenêtre 'Plots'.

Pour comprendre la différence entre les deux styles de commentaires : dans la fenêtre de script, tapez `Ctrl+Shift+K` : vous allez créer un notebook. Choisissez le format `html` et observez la différence de rendu entre les deux lignes commentées. Plus de détails sur la création de notebooks sont rassemblés dans la dernière partie du TP création de notebook

---

1.2 Kit de survie en R

1.2.1 Résumé des commandes indispensables

NB1: Dans les exemples ci-dessous, les lignes commençant par `##` sont les résultats affichés dans la console.

NB2: En R, la plupart des commandes sont des appels à des fonctions, déjà existantes ou que vous aurez créées plus haut dans le script.

- Aide sur un objet ou une fonction : tapez dans la console `?` suivi du nom de l'objet ou de la fonction.

Exemple:

```
?dnorm
```

- Affectation et affichage : Au choix, le symbole `=` ou `<-` . Pour afficher, écrire le nom de l'objet.

Exemple:

```
a = 3 ; b <- 6 ; x = a * b
x
```

```
## [1] 18
```

- Créer un vecteur manuellement : la fonction `c()`

```
X = c(2, 3, 0, 2.5)
```

et l'afficher au besoin :

```
X
```

```
## [1] 2.0 3.0 0.0 2.5
```

- Arithmétique sur les vecteurs :

```
2 * X; X + X ; X^2 ; X+10;
```

```
## [1] 4 6 0 5
```

```
## [1] 4 6 0 5
```

```
## [1] 4.00 9.00 0.00 6.25
```

```
## [1] 12.0 13.0 10.0 12.5
```

- Créer une grille :

```
ABSC = seq(-1, 1, length.out=11)
ABSC
```

```
##  [1] -1.0 -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4  0.6  0.8  1.0
```

- Accéder à certains éléments d'un vecteur:

```
ABSC[1] ; ABSC[3] ; ABSC[3:5]
```

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

```
## [1] -0.6
```

```
## [1] -0.6 -0.4 -0.2
```

- Modifier certains éléments d'un vecteur :

```
X ; X[1] = 42 ; X; X[1:3] = c(98,99,100); X
```

```
## [1] 2.0 3.0 0.0 2.5
```

```
## [1] 42.0  3.0  0.0  2.5
```

```
## [1]  98.0  99.0 100.0   2.5
```

- Calculer des statistiques de base sur des vecteurs :

```
mean(ABSC) ## moyenne empirique
```

```
## [1] 1.009294e-17
```

```
sd(ABSC) ## écart type empirique
```

```
## [1] 0.663325
```

```
var(ABSC) ## variance empirique
```

```
## [1] 0.44
```

```
min(ABSC); max(ABSC)
```

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

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

- Ecrire une boucle `for` :

Exemple: calculer le produit des 10 premiers entiers impairs.

```
res=1
for(i in 0 : 9)
  {
    res = res *(2*i+1)
  }
res
```

```
## [1] 654729075
```

- Générer des échantillons *i.i.d.* selon une loi de probabilité donnée. La commande de base pour générer un échantillon est l'appel à une fonction de type **"rNomAbrégé"**: par exemple, **rnorm** est une fonction permettant de générer des échantillons suivant une loi normale, **runif** sous une loi uniforme, **rpois** sous une loi de Poisson, etc. Pour plus d'information, tapez :

```
?distribution
```

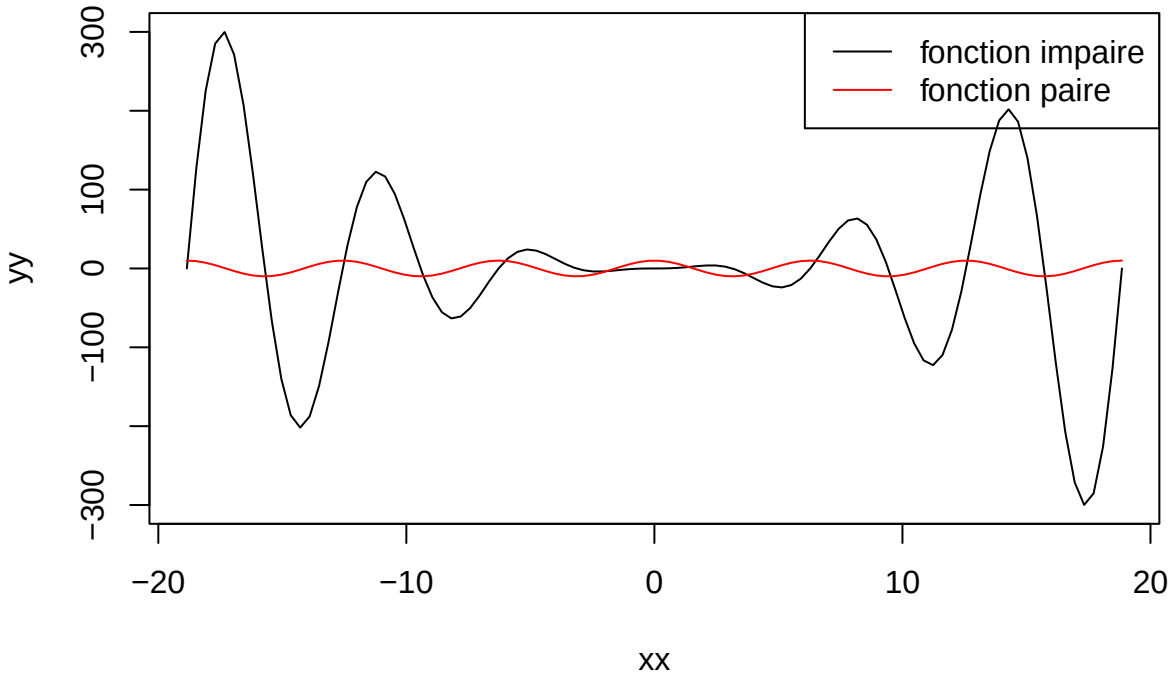
Exemple :

```
set.seed(2) ## pour la reproductibilité des résultats

X = rnorm(n = 100, mean = 5, sd = 0.3) ## échantillon de données gaussiennes de:
##taille n,
## moyenne 5
## écart type 0.3.
```

- Générer des figures :

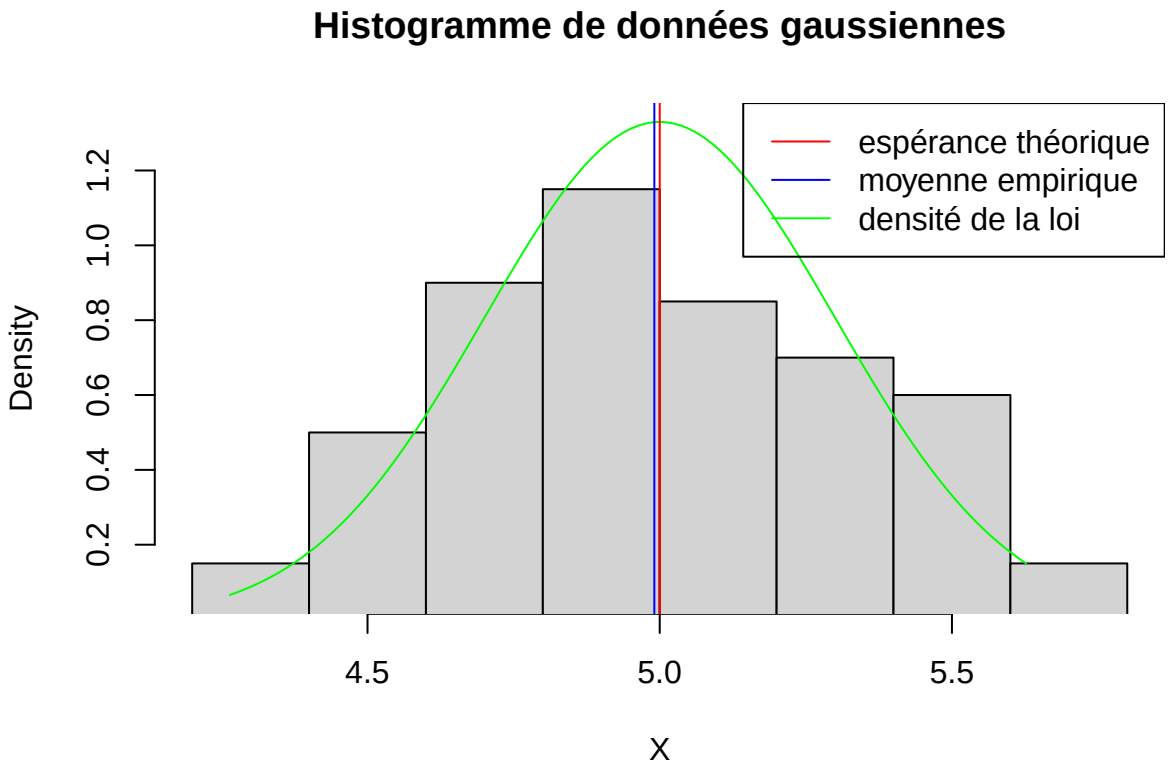
1. Des courbes

```
xx = seq(-6 * pi, 6* pi, length.out = 100) ## la grille d'abscisses
yy= sin(xx)*xx^2  ## la fonction à tracer, évaluée sur la grille.

plot(xx, yy  , type = "l") ## ouvre une fenetre graphique.
##NB: Si vous supprimez l'argument "type = 'l' ", vous aurez des points à la place de la ligne pleine.
lines(xx, 10 * cos(xx), col="red") ## pour rajouter une courbe à une figure existante:
##  commande 'lines'.

## Rajouter une légende
legend("topright", ## position de la légende
       legend = c("fonction impaire", "fonction paire"), ## texte de la légende
       lty=1,## pour avoir des lignes
       col=c("black","red") ## le code couleur
       )
```



2. Tracer l'histogramme d'un jeu de données et/ou afficher une fonction de densité :

```
ABSC = seq(min(X),max(X),length.out=100)
## Crée une "grille" de taille 100, de même plage que les données X.
DENSITY = dnorm(ABSC, mean = 5, sd = 0.3)
## densité de la loi gaussienne évaluée sur la grille

hist(X, ## premier argument: les données dont on veut l'histogramme.
     probability=TRUE, ## pour une échelle en fréquence,
                       ## pas en nombre de points dans chaque case
     main="Histogramme de données gaussiennes", ## le titre
     ylim=range(DENSITY) ## optionnel: régler l'étendue de l'axe vertical
     )
lines(x = ABSC, y = DENSITY, col="green")
abline(v=mean(X), col="blue")
abline(v = 5, col="red")
legend("topright",
       legend =c("espérance théorique",  "moyenne empirique", "densité de la loi"),
       lty=1, ## pour avoir des traits pleins
       col=c("red", "blue","green"))
```



1.3 Pour aller plus loin

Il est fortement recommandé d'apprendre les commandes élémentaires avant le lancement du mini-projet. Le résumé des commandes présenté ci-dessus est volontairement simple (iste). Pour être un peu plus à l'aise, vous pouvez après cette séance de TP :

- (conseillé pour démarrer) Suivre une partie du tutoriel interactif en anglais *"Introduction to R"* sur https://campus.datacamp.com/, **dans sa version gratuite**: chapitres 1,2,3 (basics, vectors, matrices)
- (dans un deuxième temps) Consulter un manuel de référence introductif, par exemple *an introduction to R*: les chapitres 1 et 2, et le début du chapitre 5 sur les matrices et les tableaux (pages 18 à 20) vous suffiront largement.

---

2 Exercice: Théorème Central Limite

L'objectif de cet exercice est de vérifier expérimentalement la loi des grands nombres et le théorème central limite. Toutes les commandes R nécessaires à la résolution de l'exercice sont indiquées.

2.1 Loi des grands nombres

Soit $X_1, X_2, \ldots$ une suite i.i.d. de variables aléatoires d'espérance finie μ. D'après la loi forte des grands nombres, on sait que :

$$\frac{1}{n}\sum_{k=1}^{n} X_k \overset{p.s.}{\to} \mu.$$

1. Générer une suite de $n = 100$ variables aléatoires i.i.d. de loi uniforme sur $[a, b]$, pour des réels a, b tels que $a < b$ de votre choix.

```
?runif
```

2. Tracer l'évolution de la moyenne empirique des k premières variables en fonction de k, ainsi que la valeur limite prédite par la loi des grands nombres.

```
?cumsum
?plot
```

```
## Help on topic 'plot' was found in the following packages:
##
##   Package               Library
##   graphics              /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library
##   base                  /Library/Frameworks/R.framework/Resources/library
##
##
## Using the first match ...
```

```
?abline
```

3. Tracer sur le même graphique, avec une couleur différente, le résultat obtenu avec une autre suite (indépendante) de variables aléatoires i.i.d. de même loi.

```
?line
```

4. Faire le même exercice avec une autre loi de votre choix.

```
?distribution
```

2.2 Théorème central limite

Une question intéressante est celle de la vitesse de convergence de la loi des grands nombres. Plus spécifiquement, on aimerait connaître les variations typiques autour de la valeur limite, ou mieux encore la loi de ces variations. Une réponse (partielle) à cette question est donnée par le théorème central limite. En notant σ^2 la variance (supposée finie) des variables aléatoires $X_1, X_2, \ldots$, alors

$$\sqrt{\frac{n}{\sigma^2}} \left(\frac{1}{n} \sum_{k=1}^{n} X_k - \mu \right) \xrightarrow{\mathcal{L}} Z,$$

où Z désigne une variable aléatoire de loi normale centrée réduite.

1. Tracer l'histogramme de 1000 échantillons indépendants de la moyenne empirique de $n = 100$ variables aléatoires i.i.d. de loi uniforme sur $[a, b]$, pour des valeurs a, b de votre choix.

```
?sum
?hist
# Utiliser l'option probability=TRUE
```

2. Tracer sur le même graphique la densité de la loi normale de mêmes espérance et variance que la moyenne empirique.

```
?dnorm
?lines
```

3. Faire la même expérience pour une autre loi de votre choix.

---

3 Exercice : Import et visualisation de données.

L'objectif de cet exercice est d'apprendre à importer, visualiser et traiter un jeu de données. Toutes les commandes R nécessaires à la résolution de l'exercice sont indiquées.

3.1 Vitesse de la lumière

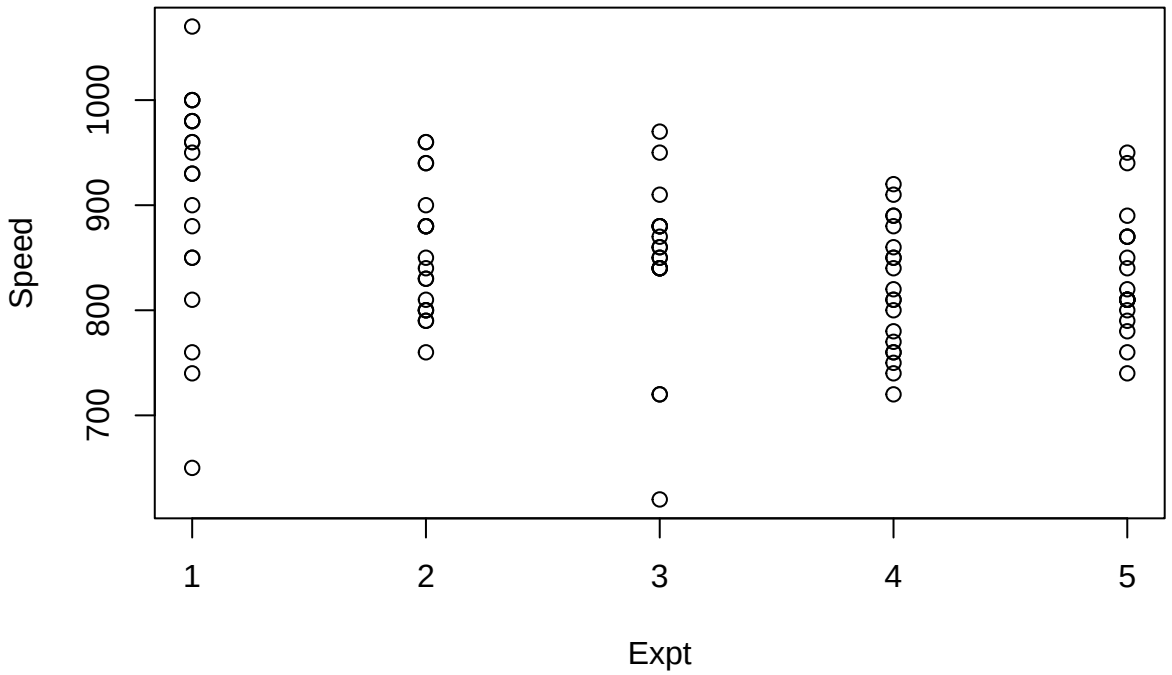
Le premier jeu de données est un ensemble de mesures de la vitesse de la lumière effectuées en 1879, basées sur 5 expériences différentes.

1. Importer le jeu de données, disponible sur R.

```
?datasets
?morley
v=morley
```

2. Tracer les résultats des mesures de chaque expérience sous forme de nuages de points.

```
plot(v[c(1,3)])
```



3. Tracer les résultats des mesures de chaque expérience sous forme de boxplots. Il faudra d'abord convertir les données en une matrice de 5 colonnes, chacune comportant les 20 résultats de l'expérience.

```
?matrix
?boxplot
```

3.2 Réchauffement climatique

Le second jeu de données donne les relevés de température annuelle à différentes latitudes de 1880 à 2015, par rapport à une période de référence de 30 ans, 1951 à 1980. Des informations sur ce jeu de données sont disponibles sur http://data.giss.nasa.gov/gistemp/

1. Télécharger le fichier temperature.csv depuis le site pédagogique puis importer le jeu de données (onglet Tools de R ; attention, le fichier comporte des entêtes !).
2. Tracer l'évolution de la température de la planète sur la période 1880-2015, puis des deux hémisphères sur le même graphique.

```
?plot
```

```
## Help on topic 'plot' was found in the following packages:
##
##   Package               Library
##   graphics              /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library
##   base                  /Library/Frameworks/R.framework/Resources/library
##
##
## Using the first match ...
```

```
?lines
?legend
```

3. Tracer sur le même graphique les histogrammes de la température annuelle de la planète sur les périodes 1880-1950, 1951-1980 et 1981-2015.

```
?subset
# You may use p=data.matrix(p) to convert a list into array
?hist
```