

TP 1 : Introduction à R et RStudio



R est un logiciel pour l'analyse statistique des données. Il fournit des procédures usuelles et possède des possibilités graphiques performantes pour explorer les données. Pouvant être utilisé aussi bien en mode interactif qu'en mode batch, R est un logiciel libre, dont le code source est disponible et qui peut être recopié et diffusé gratuitement. Des versions compilées de R sont disponibles pour Linux, Windows et MacOS.

Installation du système de base

Le site principal du logiciel R est www.r-project.org. Le téléchargement de R se fait à partir d'un des sites du «Comprehensive R archive Network» (CRAN).

Environnement de développement RStudio / RStudio Cloud

RStudio est un environnement de développement intégré (IDE) pour R. Il facilite l'utilisation de R en fournissant une interface utilisateur conviviale avec des fonctionnalités telles que l'édition de code, la gestion de projets, la visualisation de données et bien plus encore.

RStudio est disponible en deux versions :

- Une version de bureau (RStudio Desktop): Télécharger et installer RStudio Desktop depuis [ce lien](#)

- Une version en ligne (RStudio Cloud) : Créer un compte gratuit sur [RStudio Cloud](#) pour utiliser RStudio directement dans votre navigateur web, sans avoir besoin d'installer quoi que ce soit sur votre ordinateur.

Prise en main de RStudio

L'interface de RStudio est divisée en plusieurs panneaux principaux :

1. **Script Editor (Éditeur de script)** : C'est l'endroit où vous écrivez et éditez votre code R. Vous pouvez créer des scripts R (.R) ou des documents R Markdown (.Rmd) pour organiser votre travail.
2. **Console** : C'est l'endroit où vous pouvez exécuter des commandes R directement. Vous pouvez taper des commandes et voir les résultats immédiatement.
3. **Environment/History (Environnement/Historique)** : Le panneau Environnement affiche les objets (variables, fonctions, etc.) que vous avez créés dans votre session R. L'onglet Historique montre les commandes que vous avez exécutées précédemment.
4. **Files/Plots/Packages/Help (Fichiers/Graphiques/Packages/Aide)** : Ce panneau a plusieurs onglets. L'onglet Fichiers vous permet de naviguer dans vos fichiers et dossiers. L'onglet Graphiques affiche les graphiques que vous créez. L'onglet Packages vous permet de gérer les packages R installés, et l'onglet Aide fournit une documentation sur les fonctions R.

Exécution de code R

Vous pouvez exécuter du code R de plusieurs manières dans RStudio : - **Dans la console** : Tapez une commande R directement dans la console et appuyez sur Entrée pour l'exécuter. - **Dans l'éditeur de script** : Écrivez votre code dans l'éditeur de script, puis sélectionnez la ligne ou le bloc de code que vous souhaitez exécuter et appuyez sur Ctrl + Entrée pour l'exécuter dans la console. - **Exécution de scripts** : Vous pouvez exécuter un script entier en cliquant sur le bouton "Source" dans l'éditeur de script ou en utilisant le raccourci Ctrl + Shift + S.

Installation et gestion des packages

R dispose d'un système de packages qui permet d'étendre ses fonctionnalités. Pour installer un package, vous pouvez utiliser la fonction `install.packages("nom_du_package")`. Par exemple, pour installer le package `ggplot2`, vous pouvez exécuter la commande suivante dans la console :

```
install.packages("ggplot2")
```

Une fois installé, vous devez charger le package dans votre session R en utilisant la fonction `library(nom_du_package)`. Par exemple :

```
library(ggplot2)
```

Vous pouvez gérer vos packages via l'onglet “Packages” dans le panneau inférieur droit de RStudio, où vous pouvez voir les packages installés, les mettre à jour ou en installer de nouveaux.

Aide et documentation

R dispose d'une documentation complète pour toutes ses fonctions et packages. Vous pouvez accéder à l'aide en utilisant la fonction `?nom_de_la_fonction` ou `help("nom_de_la_fonction")`. Par exemple, pour obtenir de l'aide sur la fonction `mean`, vous pouvez exécuter :

```
?mean
```

Vous pouvez également utiliser l'onglet “Help” dans le panneau inférieur droit de RStudio pour rechercher des fonctions et des packages.

Ressources supplémentaires

- [R for Data Science](#) : Un livre en ligne gratuit qui couvre les bases de R et de la science des données.
- [RStudio Cheatsheets](#) : Des fiches pratiques pour R et RStudio.
- [R-bloggers](#) : Un site qui agrège des articles de blog sur R.
- [Stack Overflow](#) : Une communauté en ligne où vous pouvez poser des questions et trouver des réponses sur R.

Premiers pas

L'interaction avec R se fait en tapant des commandes dans la fenêtre R Console. Le principe de R est le suivant : vous entrez une ligne de commande, et quand vous tapez sur « Entrée », R lit cette ligne et effectue l'opération demandée.

Pour vous familiariser avec R, taper les commandes suivantes une par une dans la console ou dans un script R:

```
5+7
```

```
a = 5
```

```
a
```

```
b <- 10
```

```
b
```

```
x = 1 : 20
```

```
y = rnorm(20)
```

```
plot (x, y)
```

```
plot (x, y, pch=16, col=2)
```

Manipulation de base

Objets

L'objet de base en R est le vecteur. Un vecteur peut contenir des valeurs numériques, des valeurs binaires (True ou False), des chaînes de caractères...

Les fonctions les plus utilisées pour créer des vecteurs sont `c`, `rep` et `seq`:

```
c(1,2,3,4,5,6)
c(T,T,F,F)
c('a','b')
rep(55,10)
rep(c(1,2),10)
rep(c('a','b'),c(2,7))
seq(1,10,by=0.1)
```

Accéder aux éléments d'un vecteur

Soit le vecteur `a` composé par 50 valeurs aléatoires :

```
a = rnorm(50)
```

Que donnent les commandes suivantes ?

```
a[1]  
a[2]  
a[c(1,3,5)]  
a>0  
a[a>0]
```

Une particularité de R est que les éléments d'un vecteur peuvent avoir des noms :

```
v=c(1,2,3,4)  
names(v)=c('alpha','beta','gamma','delta')  
v['beta']
```

Arrays, listes et data.frames

D'autres objets de R sont les listes, les arrays (vecteurs multidimensionnels) et les `data.frames`. Les `data.frames` sont des listes de vecteurs qui ont tous la même longueur. Les `data.frames` sont très bien adaptés pour stocker des données présentées sous forme de tableau bi-dimensionnel.

```
a=array(1:20,dim=c(4,5))  
a[2,4]
```

```
b=list(alpha=1:3, beta=c('a','b','c','d'))  
names(b)  
b$alpha  
b$beta
```

```
c=data.frame(a=gl(2,5,10),b=1:10,x=rnorm(10))  
c$a  
c$b  
c$x  
c[1:2,]
```

Les matrices

```
a=matrix(1:9,nrow=3)
a
a=matrix(1:9,nrow=3,byrow=T)
a
```

```
b=matrix(c(5,3,7,4,6,3,1,6,3,2,8,5),nrow=4)
b
b=matrix(c(5,3,7,4,6,3,1,6,3,2,8,5),nrow=4,byrow=T)
b
```

```
v=1:3 # ou v=c(1,2,3)
v
c=cbind(a,v)
d=rbind(a,b)
c
d
```

Utilisation d'une fonction

Dans cet exemple nous utilisons la fonction `mean` pour calculer la moyenne des valeurs stockées dans le vecteur `numbers`. Puis nous utilisons la fonction `print` pour afficher la valeur de la variable `mean_value`.

```
# Exemple de vecteur
numbers <- c(1, 2, 3, 4, 5)
# Calcul de la moyenne
mean_value <- mean(numbers)
print(mean_value)
```