



M201 Approfondir les systèmes décisionnels



75 heures

SOMMAIRE

1. Approfondir les statistiques

- Rappeler les notions essentielles
- Assimiler les probabilités et distributions avancées
- Maîtriser les mesures de dispersion et de position
 - Vulgariser le langage R

2. Visualiser les données décisionnelles

- Créer divers types de graphiques
- Configurer les graphiques pour une analyse profonde

3. Maîtriser les bases de données décisionnelles

- Introduire les bases de données décisionnelles
- Explorer la modélisation dimensionnelle avancée
- Appliquer le processus ETL dans le contexte décisionnel

PARTIE 1

Approfondir les statistiques

Dans ce module, vous allez :

- Rappeler les notions essentielles
- Maîtriser les mesures de dispersion et de position
- Assimiler les probabilités et distributions avancées
- Vulgariser le langage R



40 heures

CHAPITRE 4

Vulgariser le langage R

Ce que vous allez apprendre dans ce chapitre :

1. Introduction à R et son rôle en statistique
2. Préparation de l'environnement
3. Syntaxe de base et principaux opérateurs en R
4. Structures de données fondamentales en R
5. Manipulations de données en R



20 heures



01 – Introduction à R et son rôle en statistique

1-1 Historique et développement



Historique et développement

Origine : Le langage S a été développé par **John M. Chambers** et son équipe chez Bell Laboratories dans les années **1970**.

Popularité de S : À la fin des années **1980**, S est devenu populaire grâce à **S-PLUS** (une version commerciale du S) pendant près de deux décennies.

Émergence de R : Ross Ihaka et Robert Gentleman ont créé R au milieu des années 1990, inspiré par S et Scheme, et ont intégré R au **projet GNU**, le rendant ainsi **logiciel libre**.

Montée de R : R a rapidement gagné en popularité (vs S-PLUS) dans le milieu académique, et attirant des contributions significatives, y compris celles de **John Chambers**, qui a réorienté ses efforts vers R.



01 – Introduction à R et son rôle en statistique

1-2 Caractéristiques de R en tant que langage



Caractéristiques de R en tant que langage



Le R est un langage particulièrement puissant pour les applications mathématiques et statistiques vu qu'il a été développé dans ce but.

Parmi ses caractéristiques particulièrement intéressantes, on note :

- ❑ **Environnement intégré** : R est un environnement conçu pour la manipulation de données, le calcul et la création de graphiques, tout en étant un langage de programmation autonome.
- ❑ **Inspiration et origine** : R s'inspire principalement des langages **S** et **Scheme**, qui eux-mêmes sont influencés par **APL** et **Lisp**.
- ❑ **Langage interprété** : Contrairement aux langages compilés comme C ou C++, R est un langage interprété, nécessitant un interprète pour exécuter les commandes soumises.

01 – Introduction à R et son rôle en statistique

1-2 Caractéristiques de R en tant que langage



Caractéristiques de R en tant que langage (suite)



- ❑ **langage basé sur la notion de vecteur**, ce qui simplifie les calculs mathématiques et réduit considérablement le recours aux structures itératives (boucles for, while, etc.) ;
- ❑ **pas de typage** ni de déclaration obligatoire des variables ;
- ❑ **programmes courts**, en général quelques lignes de code seulement ;
- ❑ temps de développement très court.

01 – Introduction à R et son rôle en statistique

1-3 Ou R est utilisé ?



Ou R est utilisé ?

R est utilisé dans divers domaines, notamment :



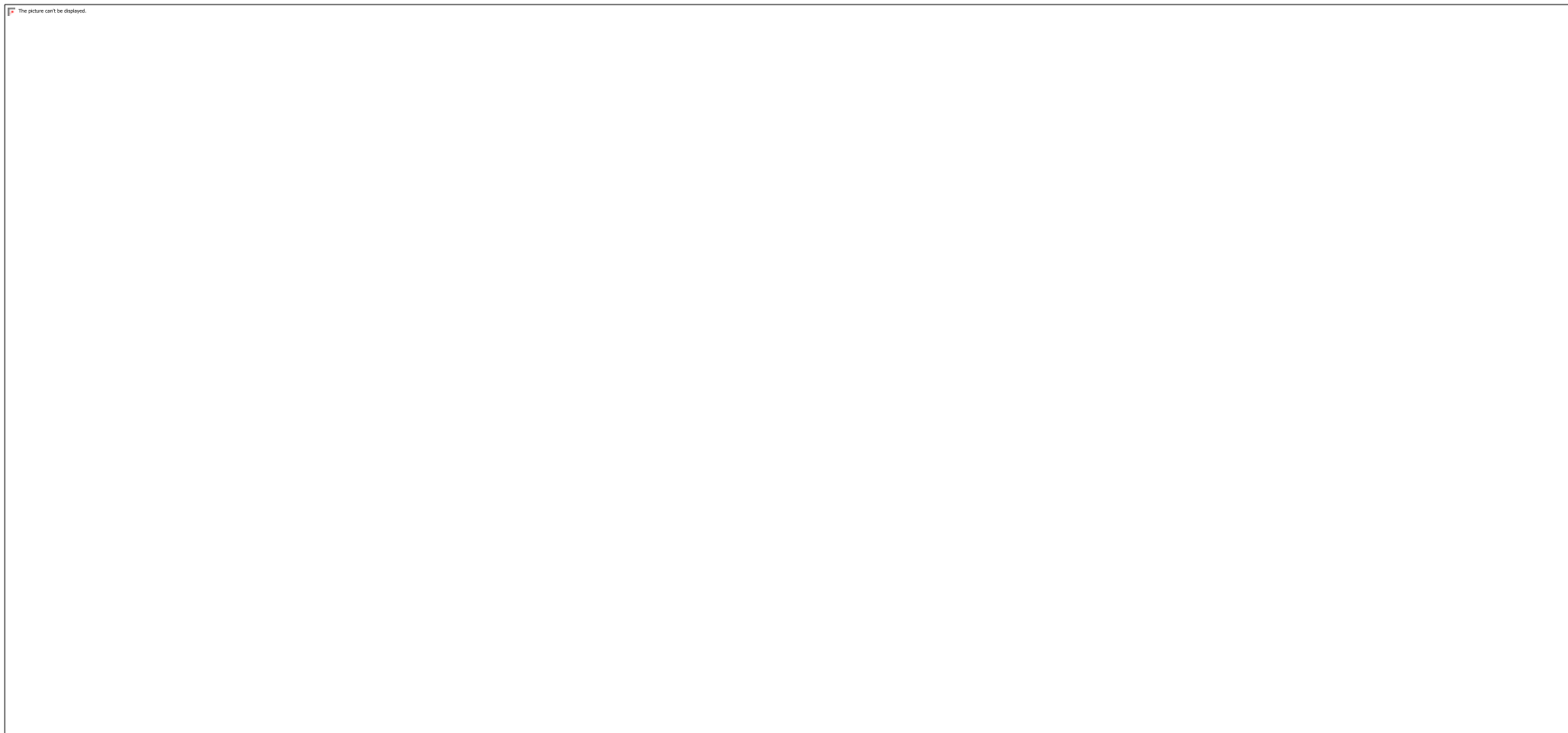
- 1.Statistiques et analyse de données** : R est largement utilisé pour des analyses statistiques avancées, des tests d'hypothèses et des modèles de régression.
- 2.Science des données** : Les scientifiques des données utilisent R pour le nettoyage, la manipulation et l'analyse de grandes quantités de données.
- 3.Bioinformatique** : R est populaire pour l'analyse de données génomiques, l'épidémiologie et d'autres recherches en biologie.
- 4.Finance** : Les analystes financiers utilisent R pour la modélisation statistique, l'analyse de risques et le traitement des données financières.
- 5.Recherche académique** : Dans de nombreux domaines académiques, R est utilisé pour l'analyse de données et la visualisation, facilitant la publication de résultats reproductibles.
- 6.Marketing et analyse commerciale** : R est employé pour l'analyse de tendances de consommation, la segmentation de marché et la prévision des ventes.

01 – Introduction à R et son rôle en statistique

1-3 Ou R est utilisé ?



R et les industries concernées

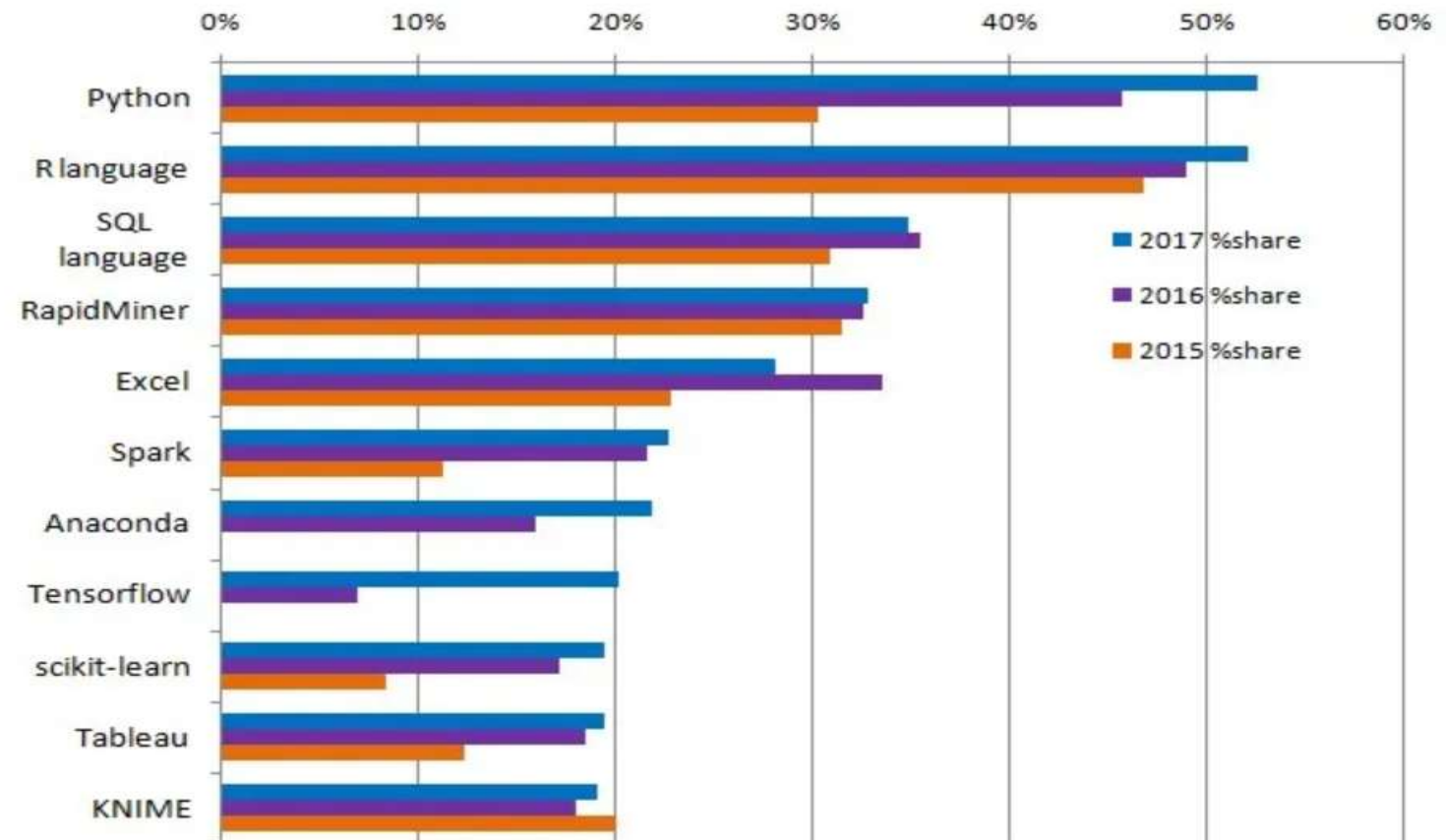


01 – Introduction à R et son rôle en statistique

1-4 R dans le contexte concurrentiel

R dans le contexte concurrentiel

KDnuggets Analytics, Data Science, Machine Learning Software Poll, top tools share, 2015-2017



01 – Introduction à R et son rôle en statistique

1-5 Top des projets utilisant R



Top des projets utilisant R



Voici quatre projets notables de grandes entreprises qui utilisent R :

Google : Utilise R pour l'analyse de données et le machine learning, notamment dans des projets liés à l'analyse de texte et à la recherche d'informations. R est intégré dans des outils comme Google Cloud pour l'analyse statistique.

Facebook : Utilise R pour des analyses statistiques approfondies, y compris pour la compréhension des comportements des utilisateurs et l'optimisation des publicités. Des packages R spécifiques ont été développés pour faciliter ces analyses.

IBM : Avec IBM SPSS et IBM Watson, R est intégré pour fournir des solutions d'analyse prédictive et de data mining. R est utilisé pour améliorer les capacités analytiques des produits de l'entreprise.

Microsoft : A intégré R dans plusieurs de ses produits, y compris SQL Server et Power BI, permettant aux utilisateurs d'exécuter des analyses avancées et des visualisations directement dans leurs environnements de données.

02 – Préparation de l'environnement

2-1 Choix de l'IDE



2-1 Choix de l'IDE

Voici quelques environnements de développement intégrés (IDE) qui intègrent le langage R :

- ❑ Jupyter Notebook
- ❑ Visual Studio Code
- ❑ Eclipse avec le plugin StatET
- ❑ **Rstudio**

02 – Préparation de l'environnement

2-1 Choix de l'IDE

2-1 Choix de l'IDE

Rstudio :

C'est l'IDE le plus populaire pour R, offrant une interface conviviale, des outils de visualisation, et un support pour le développement de packages R et la gestion de projets.

RStudio



RStudio sous Ubuntu 12.10

Informations

Créateur	Joseph J. Allaire, Kevin Ushey (d), Jonathan McPherson (d) et Gary Ritchie (d) 
Développé par	RStudio, Inc.
Première version	28 février 2011
Dernière version	2024.04.2+764 (Chocolate Cosmos, 10 juin 2024) ¹ 
Dépôt	GitHub 

État du projet	En développement actif
Écrit en	C++, Java et JavaScript 
Système d'exploitation	Microsoft Windows, Linux et macOS 
Environnement	Windows, OS X, GNU/Linux
Type	EDI
Licence	GNU Affero General Public License v3 ²
Site web	posit.co/products/open-source/rstudio  

[modifier](#) - [modifier le code](#) - [voir Wikidata](#) ([aide](#))

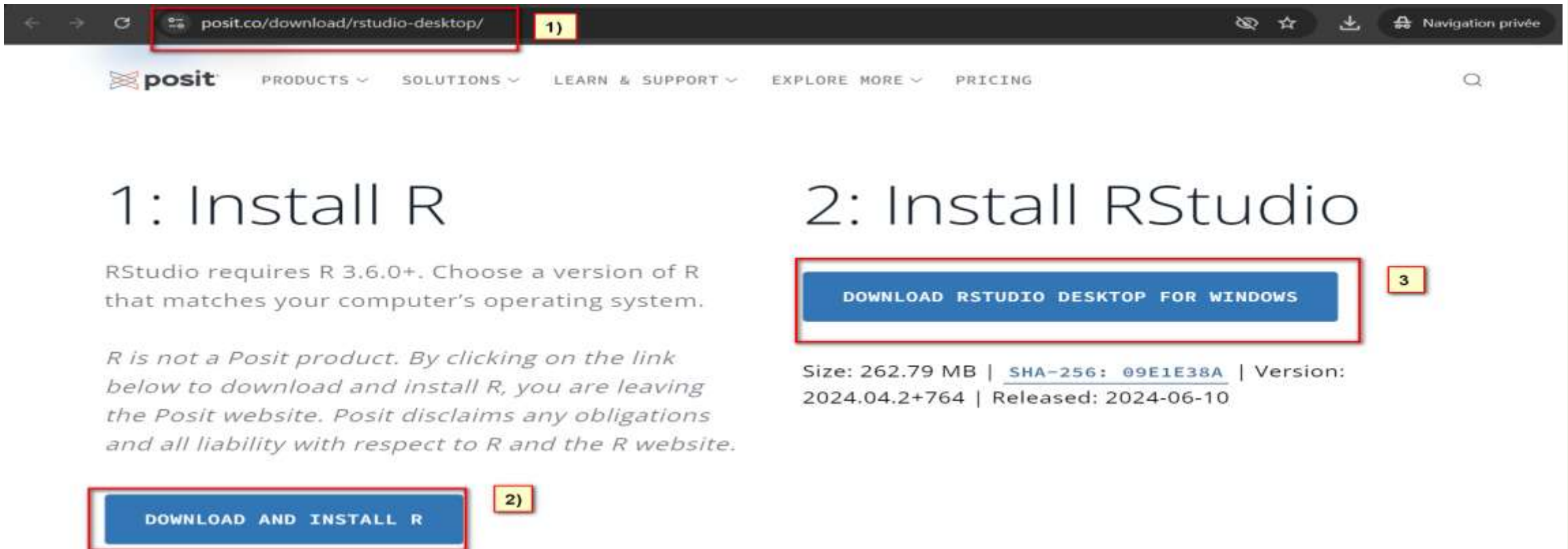


02 – Préparation de l'environnement

2-2 Installation de R et Rstudio

2-2 Installation de R et Rstudio

Le téléchargement peut se faire depuis le site illustré dans la figure. Il faut installer d'abord R puis Rstudio.



The screenshot shows the Posit website with the URL `posit.co/download/rstudio-desktop/` in the browser address bar, marked with a red box and a yellow box labeled "1)". The website has a navigation bar with links: **posit**, PRODUCTS, SOLUTIONS, LEARN & SUPPORT, EXPLORE MORE, and PRICING. The main content is divided into two sections: "1: Install R" and "2: Install RStudio".

1: Install R

RStudio requires R 3.6.0+. Choose a version of R that matches your computer's operating system.

R is not a Posit product. By clicking on the link below to download and install R, you are leaving the Posit website. Posit disclaims any obligations and all liability with respect to R and the R website.

A blue button labeled "DOWNLOAD AND INSTALL R" is highlighted with a red box and a yellow box labeled "2)".

2: Install RStudio

A blue button labeled "DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS" is highlighted with a red box and a yellow box labeled "3)".

Size: 262.79 MB | [SHA-256: 09E1E38A](#) | Version: 2024.04.2+764 | Released: 2024-06-10

02 – Préparation de l'environnement

2-3 Accueil de R et de Rstudio



2-3 Accueil de R et de Rstudio

R comme Python peut être utilisé sans Environnement de Développement Intégré (EDI).

Pour ce faire, dans le **menu Démarrer**, vous pouvez chercher **R 4.4.1**

Si vous aimez travailler avec **l'EDI Rstudio**, dans ce cas vous le lancez également depuis le **menu Démarrer** comme se voit dans la figure.



02 – Préparation de l'environnement

2-3 Accueil de R et de Rstudio



2-3 Accueil de R

Vous pouvez commencer la programmation via l'invite de commandes de R.

Exemples de commandes :

getwd() donne le dossier courant de tes fichiers, projets, et données d'entrées à R .

read.csv(« data.csv ») essaie de lire le fichier depuis le dossier courant. Une erreur est renvoyée car le fichier data.csv n'existe pas dans **"C:/Users/pc/OneDrive - OFPPT/Documents"**

```
RGui
Fichier Edition Voir Div Paquets Fenêtres Aide

R Console

R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

[Sauvegarde de la session précédente restaurée]

> getwd()
[1] "C:/Users/pc/OneDrive - OFPPT/Documents"
> read.csv("mydata.csv")
```

02 – Préparation de l'environnement

2-3 Accueil de R et de Rstudio



2-3 Accueil de R (suite)

Vous pouvez commencer la programmation via l'invite de commandes de R.

Exemples de commandes :

- Opérations arithmétiques
- Affectation avec `<-` ou `=` (préféré est `<-`)
- Et bien d'autres

```
R Console

> getwd()
[1] "C:/Users/pc/OneDrive - OFPPT/Documents"
> read.csv("mydata.csv")
Erreur dans file(file, "rt") : impossible d'ouvrir la connexion
De plus : Message d'avis :
Dans file(file, "rt") :
  impossible d'ouvrir le fichier 'mydata.csv' : No such file or directory
> 2/2
[1] 1
> 2+3
[1] 5
> 2/0
[1] Inf
> 0/2
[1] 0
> 0/0
[1] NaN
> x<-0
> x
[1] 0
> z = 3
> z
[1] 3
> |
```

02 – Préparation de l'environnement

2-3 Accueil de R et de Rstudio



2-3 Accueil de R (suite)

- ✓ Nous pouvons créer un fichier R.
- ✓ Le mettre dans le répertoire courant
- ✓ Y écrire un code R
- ✓ Importer le fichier R
- ✓ Pour l'exécuter ou
- ✓ Pour exécuter une fonction particulière

The screenshot shows the RGui window with two panes. The left pane is the R Console, and the right pane is a script editor titled 'C:\Users\pc\OneDrive - OFPPT\Documents\r1.R - Edit...'. The console shows the execution of several R commands, including a warning about character encoding, a directory listing, and the sourcing and execution of a function 'f1' that prints 'hello world'. The script editor shows the source code for the 'f1' function.

```
pattern <- sub("\\[<-", "\\\\\\\[<-$", pattern, all.names, value = TRUE)
warning("replaced '[<-' by '\\\\\\\\[<-$")
}
}
grep(pattern, all.names, value = TRUE)
}
else all.names
}
}
<bytecode: 0x000001e1056a24c8>
<environment: namespace:base>
> dir()
[1] "Desktop.ini"           "Disque OFPPT" $
[3] "Fichiers Outlook"     "IAM"          $
[5] "Modèles Office personnalisés" "r1.R"         $
> source("r1.R")
[1] "hello world"
> source("r1.R")
[1] "hello world"
> f1
function() {
  print ("hello world")
}
> f1()
[1] "hello world"
>
```

```
# hello world
print ("hello world")

f1 <- function() {
  print ("hello world")
}
```

02 – Préparation de l'environnement

2-3 Accueil de R et de Rstudio



2-3 Accueil de RStudio

Vous pouvez commencer la programmation via l'invite de commandes de R intégré dans RStudio.

The screenshot shows the RStudio interface. The console window on the left displays the R startup message and the results of several commands. The file explorer on the right shows the contents of the 'Home' directory.

```
R version 4.4.1 (2024-06-14 ucrt) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> getwd()
[1] "C:/Users/pc/OneDrive - OFPPT/Documents"
> read.csv("data.csv")
Erreur dans file(file, "rt") : impossible d'ouvrir la connexion
De plus : Message d'avis :
Dans file(file, "rt") :
  impossible d'ouvrir le fichier 'data.csv' : No such file or directory
> 2+2
[1] 4
>
> x <- 5
> 2/0
[1] Inf
> 0/2
[1] 0
> 0/0
[1] NaN
> |
```

The file explorer on the right shows the following files and folders:

Name	Size	Modified
.RData	49 B	Sep 23, 2024, 7:15 PM
.Rhistory	5 B	Sep 23, 2024, 7:15 PM
desktop.ini	418 B	Sep 22, 2024, 8:21 PM
Disque OFPPT		
Fichiers Outlook		
IAM		
Modèles Office personnalisés		

02 – Préparation de l'environnement

2-4 Blocs de l'interface principale de RStudio



2-3 Accueil de RStudio

Vous pouvez commencer la programmation via l'invite de commandes de R intégré dans RStudio.

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Les variables

Dans R les variables ne sont pas déclarées comme un type de données. Les objets R sont affectés aux variables et le type de données de l'objet R devient le type de données de la variable.

Il existe de nombreux types d'objets R. Les plus fréquemment utilisés sont :

- Vecteurs
- Listes
- Matrices
- Tableaux
- Facteurs
- Trames de données

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Les vecteurs

Le plus simple de ces objets est l'**objet vecteur** et il existe **six types de données** de ces vecteurs atomiques, également appelés **six classes** de vecteurs. Les autres objets R sont construits sur les vecteurs atomiques.

Data Type	Exemple	Verification
Logical	TRUE , FALSE	<pre>v <- TRUE print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "logical"</pre>
Numeric	12.3, 5, 999	<pre>v <- 23.5 print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "numeric"</pre>

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Les vecteurs

Data Type	Exemple	Verification
Integer	2L, 34L, 0L	<pre>v <- 2L print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "integer"</pre>
Complex	3 + 2i	<pre>v <- 2+5i print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "complex"</pre>
Character	'a' , '"good", "TRUE", '23.4'	<pre>v <- "TRUE" print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "character"</pre>

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Les vecteurs

Data Type	Exemple	Verification
Raw	"Hello" is stored as 48 65 6c 6c 6f	<pre>v <- charToRaw("Hello") print(class(v))</pre> <p>Cela produit :</p> <pre>[1] "raw"</pre>

Note :

Veuillez noter que dans R, le nombre de classes ne se limite pas aux six types ci-dessus. Par exemple, nous pouvons utiliser de nombreux vecteurs atomiques et créer un tableau dont la classe deviendra un tableau.

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Les vecteurs à plusieurs éléments

Lorsque vous souhaitez créer un vecteur avec plusieurs éléments, vous devez utiliser la fonction `c()` qui consiste à combiner les éléments dans un vecteur.

```
# Create a vector.  
apple <- c('red','green','yellow')  
print(apple)  
# Get the class of the vector.  
print(class(apple))
```

Lorsque vous exécutez le code, cela donne :

```
[1] "red" "green" "yellow"  
[1] "character"
```

03 – Syntaxe de base et opérateurs en R

3-2 Les variables



Nomination des variables

Ci-dessous des règles de nomination des variables

Nom de la variable	Validité	Raison
var_name2.	valide	Contient des lettres, des chiffres, un point et un underscore.
var_name%	invalide	Contient le caractère '%'. Seuls un point (.) et un underscore sont autorisés.
2var_name	invalide	Commence par un chiffre.
.var_name, var.name	valide	Peut commencer par un point (.) mais ne doit pas être suivi d'un chiffre.
.2var_name	invalide	Le point de départ est suivi d'un chiffre, ce qui le rend invalide.
_var_name	invalide	Commence par un underscore (_), ce qui n'est pas valide.

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Principaux opérateurs en R

Un opérateur est un symbole qui indique au compilateur d'effectuer des manipulations mathématiques ou logiques spécifiques. Le langage R est riche en opérateurs intégrés et fournit les types d'opérateurs suivants.

- Opérateurs arithmétiques
- Opérateurs relationnels
- Opérateurs logiques
- Opérateurs d'affectation
- Opérateurs divers

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs arithmétiques

Operator	Description	Example
+	Additionne deux vecteurs	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v+t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 10.0 8.5 10.0</pre>
-	Soustrait le deuxième du premier	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v-t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] -6.0 2.5 2.0</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs arithmétiques

Operator	Description	Example
/	Division réelle de v sur t	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v/t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 0.250000 1.833333 1.500000</pre>
%%	Donne le reste de v sur t	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%%t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 2.0 2.5 2.0</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs arithmétiques

Operator	Description	Example
%/%	Division entière de v sur t (quotient)	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%/%t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 0 1 1</pre>
^	v à la puissance t	<pre>v <- c(2,5.5,6) t <- c(8, 3, 4) print(v^t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 256.000 166.375 1296.000</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs relationnels

Operator	Description	Example
>	Vérifie si chaque élément du premier vecteur est supérieur à l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] FALSE TRUE FALSE FALSE</pre>
<	Vérifie si chaque élément du premier vecteur est inférieur à l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v < t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE FALSE TRUE FALSE</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs relationnels

Operator	Description	Example
==	Vérifie si chaque élément du premier vecteur est égale à l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v==t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] FALSE FALSE FALSE TRUE</pre>
<=	Vérifie si chaque élément du premier vecteur est inférieur ou égale à l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v<=t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE FALSE TRUE TRUE</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs relationnels

Operator	Description	Example
<code>>=</code>	Vérifie si chaque élément du premier vecteur est supérieur ou égal à l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>=t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] FALSE TRUE FALSE TRUE</pre>
<code>!=</code>	Vérifie si chaque élément du premier vecteur est différent de l'élément correspondant du deuxième vecteur.	<pre>v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v!=t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE TRUE TRUE FALSE</pre>

Opérateurs logiques

Ils s'appliquent uniquement aux vecteurs de type **logique, numérique ou complexe**. Tous les nombres supérieurs à 1 sont considérés comme des valeurs logiques (VRAI). Chaque élément du premier vecteur est comparé à l'élément correspondant du deuxième vecteur. Le résultat de la comparaison est une valeur booléenne.

Operator	Description	Example
&	Il s'agit de l'opérateur logique AND élément par élément. Il combine chaque élément du premier vecteur avec l'élément correspondant du deuxième vecteur et donne une sortie VRAI si les deux éléments sont VRAI.	<pre>v <- c(3,1,TRUE,2+3i) t <- c(4,1,FALSE,2+3i) print(v&t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE TRUE FALSE TRUE</pre>
!	On l'appelle opérateur NON logique. Il prend chaque élément du vecteur et donne la valeur logique opposée.	<pre>v <- c(3,0,TRUE,2+2i) print(!v)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] FALSE TRUE FALSE FALSE</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs logiques

L'opérateur logique && et || ne considère que le premier élément des vecteurs et donne un vecteur d'un seul élément en sortie.

Operator	Description	Exemple
&&	Appelé opérateur logique AND. Il prend le premier élément des deux vecteurs et donne la valeur VRAI uniquement si les deux sont VRAI.	<pre>v <- c(3,0,TRUE,2+2i) t <- c(1,3,TRUE,2+3i) print(v&& t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE</pre>
	Appelé opérateur logique OU. Il prend le premier élément des deux vecteurs et donne la valeur VRAI uniquement si les deux sont VRAI.	<pre>v <- c(0,0,TRUE,2+2i) t <- c(0,3,TRUE,2+3i) print(v t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] FALSE</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs d'affectation

Ces opérateurs sont utilisés pour attribuer des valeurs aux vecteurs.

Operator	Description	Exemple
<code><-</code> or <code>=</code> or <code><<-</code>	Appelée affectation à gauche	<pre>v1 <- c(3,1,TRUE,2+3i) v2 <<- c(3,1,TRUE,2+3i) v3 = c(3,1,TRUE,2+3i) print(v1) print(v2) print(v3) Cela produit après l'exécution : [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i</pre>
<code>-></code> Or <code>->></code>	Appelée affectation à droite	<pre>c(3,1,TRUE,2+3i) -> v1 c(3,1,TRUE,2+3i) ->> v2 print(v1) print(v2) Cela produit après l'exécution : [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs divers

Ces opérateurs sont utilisés à des fins spécifiques et non pour des calculs mathématiques ou logiques généraux.

Operator	Description	Exemple
:	Opérateur deux points. Il crée la série de nombres en séquence pour un vecteur.	<pre>v <- 2:8 print(v)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] 2 3 4 5 6 7 8</pre>
%in%	Cet opérateur est utilisé pour identifier si un élément appartient à un vecteur.	<pre>v1 <- 8 v2 <- 12 t <- 1:10 print(v1 %in% t) print(v2 %in% t)</pre> <p>Cela produit après l'exécution :</p> <pre>[1] TRUE [1] FALSE</pre>

03 – Syntaxe de base et opérateurs en R

3-3 Principaux opérateurs en R



Opérateurs divers

Ces opérateurs sont utilisés à des fins spécifiques et non pour des calculs mathématiques ou logiques généraux.

Operator	Description	Exemple									
%*%	Cet opérateur est utilisé pour multiplier une matrice par sa transposée.	<pre>M = matrix(c(2,6,5,1,10,4), nrow=2,ncol=3,byrow = TRUE) t = M %*% t(M) print(t)</pre> <p>Cela produit après l'exécution :</p> <table><tr><td></td><td>[,1]</td><td>[,2]</td></tr><tr><td>[1,]</td><td>65</td><td>82</td></tr><tr><td>[2,]</td><td>82</td><td>117</td></tr></table>		[,1]	[,2]	[1,]	65	82	[2,]	82	117
	[,1]	[,2]									
[1,]	65	82									
[2,]	82	117									