Data Mining avec langage R

Département : Informatique
Institut : Mathématique et Informatique
1 Master I2A + STIC

2022/2023

Plan

TP 1: Premiers pas avec R

1. **Présentation générale du R**
2. **Environnement de travail Rstudio**
3. **Gestion des Packages R**

Data mining

À la croisée des chemins

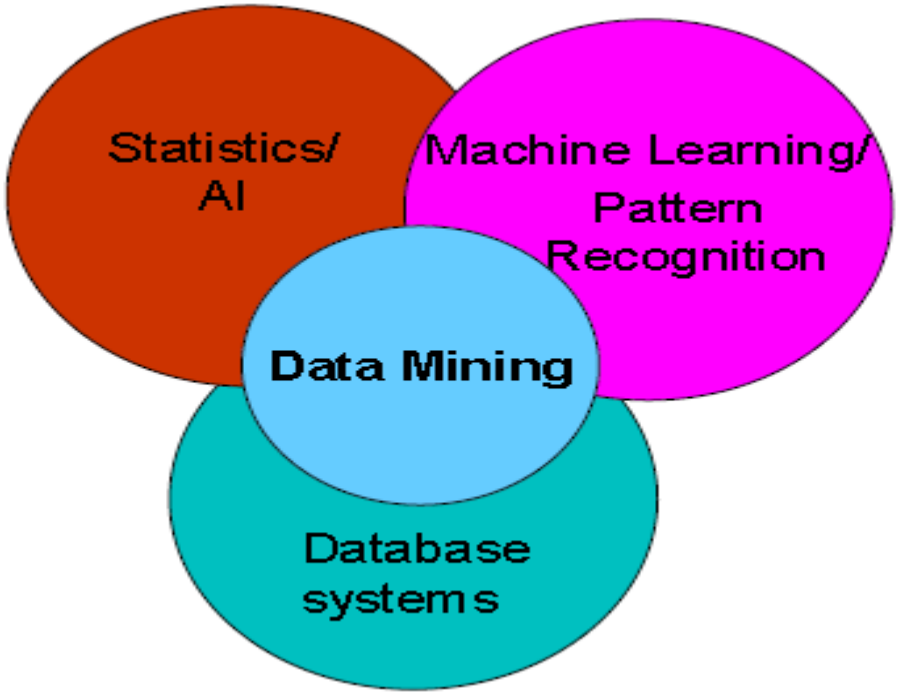


Figure . Origine du Data Mining.
(extraite de ‘’Tan et al, Introduction to Data Mining, 2004’’)

1. Présentation générale du logiciel

1.1, Qu’est-ce que R ?

✓Le projet R naît en 1993 comme un projet de recherche de Ross Ihaka et Robert Gentleman à l'université d'Auckland (Nouvelle-Zélande) Qu’est-ce que R ?

✓Un clone du logiciel S-plus qui est fondé sur le langage de programmation oriente objet S, développé par John Chambers a AT&T Bell Laboratoires en 1988.

✓C'est un logiciel libre distribué selon les termes de la licence GNU GPL et est disponible sous GNU/Linux, Mac OS et Windows.

1. Présentation générale du logiciel

1.2. Utilisation du langage R

- **Analyse statistique**
  - ✓ Régressions (linéaires, logistiques, mixtes).
  - ✓ Séries temporelles (forecast).
  - ✓ Tests statistiques (ANOVA, Chi2, etc.).
- **Data Science**
  - ✓ Préparation, nettoyage et transformation des données.
  - ✓ Exploration et visualisation avancée (tidyverse, ggplot2).
- **Data Mining**
  - ✓ Segmentation et clustering (k-means, hiérarchique).
  - ✓ Détection de motifs, analyse de corrélations.
  - ✓ Exploration de grands jeux de données.

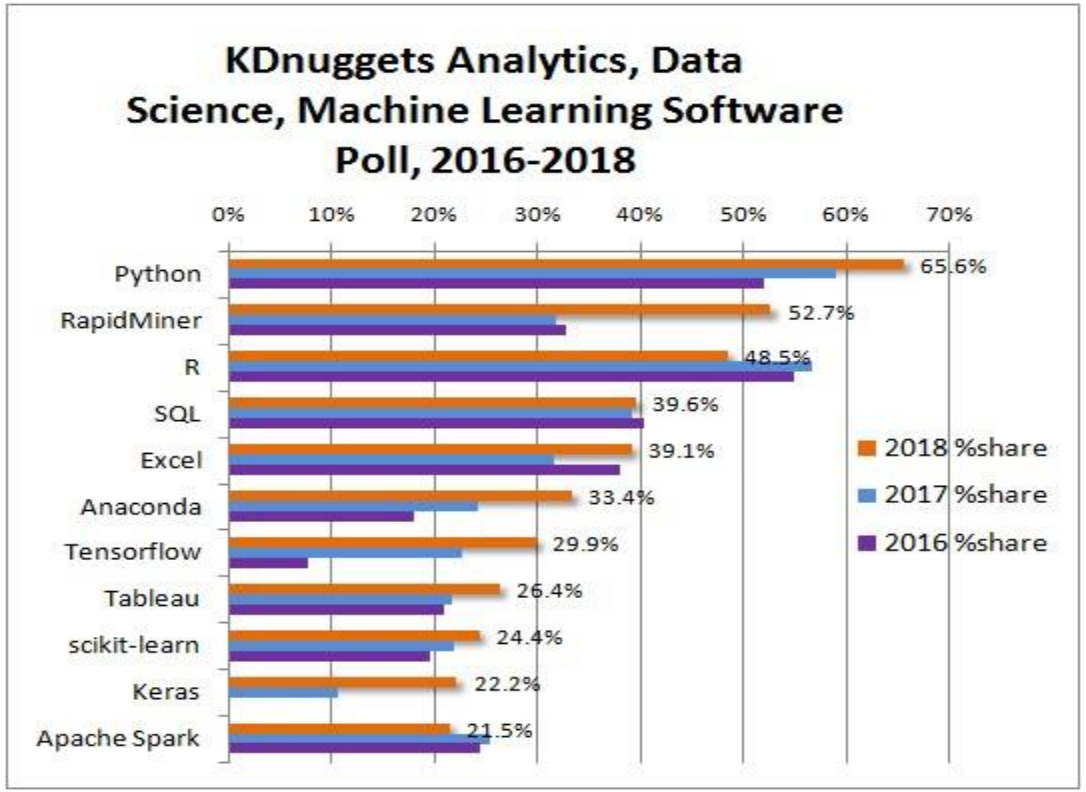
1. Présentation générale du logiciel

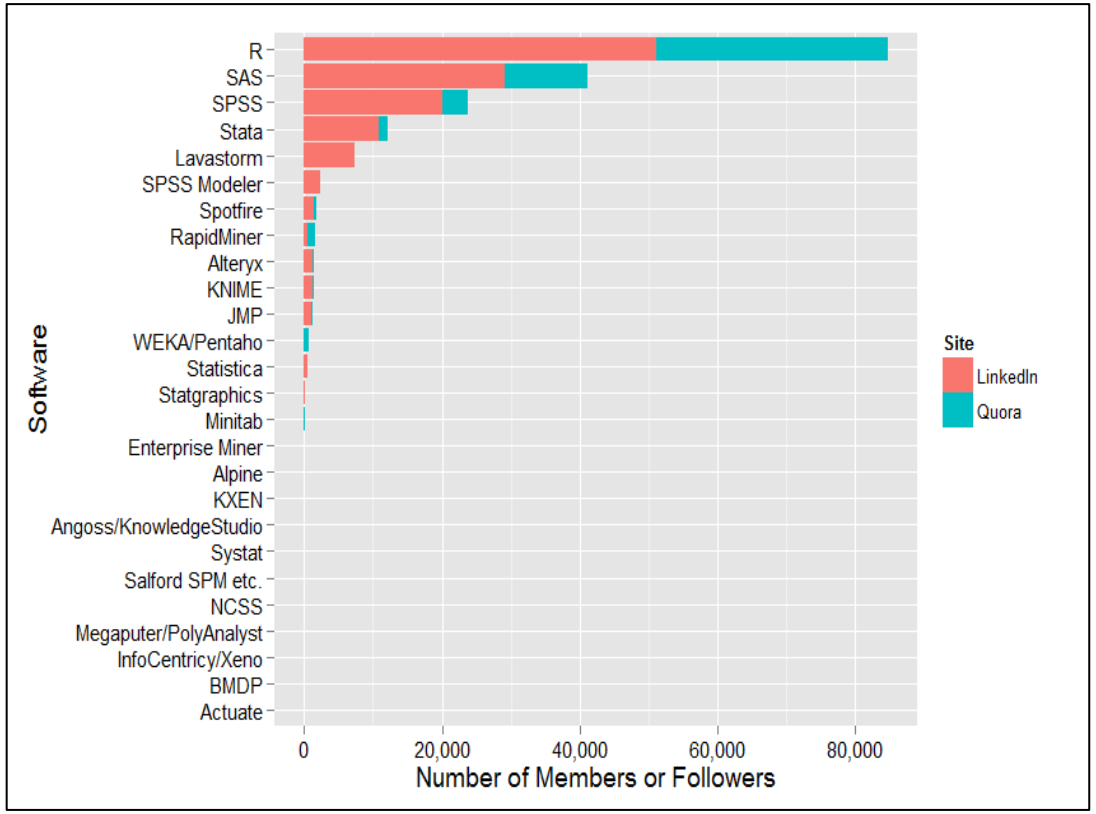
1.2. Utilisation du langage R

- **Machine Learning**
  - ✓ Classification, régression, prédiction (caret, mlr3, xgboost).
  - ✓ Réseaux de neurones (keras, tensorflow pour R).
  - ✓ Évaluation des performances (AUC, ROC, cross-validation).
- **Autres domaines**
  - ✓ Text mining (tm, quanteda).
  - ✓ Bioinformatique (Bioconductor).
  - ✓ Applications interactives et dashboards (shiny).

1. Présentation générale du logiciel

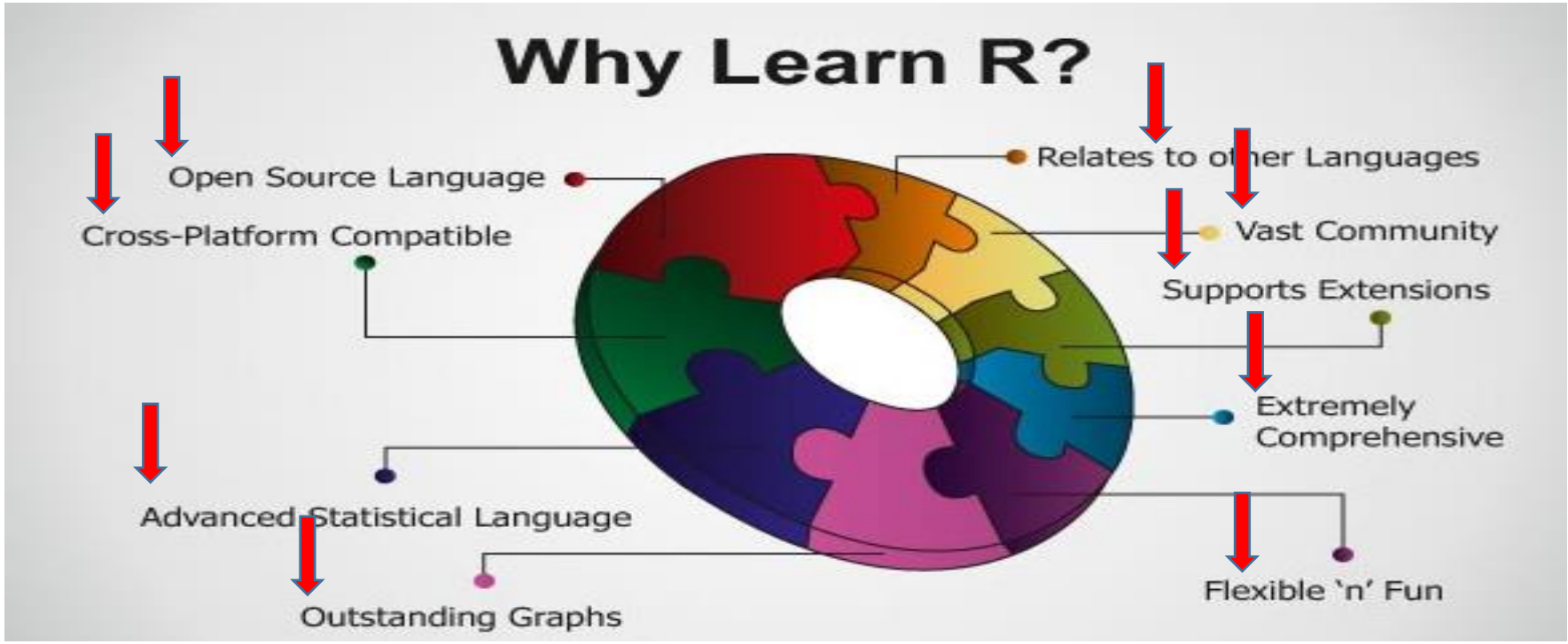
1.3. Popularité du langage R





1. Présentation générale du logiciel

1.4. Pourquoi le langage R



1. Présentation générale du logiciel

1.5.Distributions, installation, et documentation

- ✓ La distribution la plus connue du langage R est celle du R Project et du Comprehensive R Archive Network (CRAN).
- ✓ Il existe d'autres distributions comme la distribution proposée par Microsoft ou encore celle de l'entreprise Oracle, Oracle R Distribution
- ✓ R est gratuit (open source), fonctionne sur plusieurs plateformes (windows,MacOSX, Linux ,…), disponible sur : *http://cran.r-project.org /*
- ✓ Documentation sur le logiciel R :

  www.math.sciences.univ-nantes.fr/:philippe/R.html
- ✓ site consacré aux graphiques :

  addictedtor.free.fr/graphiques/

1. Présentation générale du logiciel

1.6. Editeurs pour le langage R

✓***RGUI***, l'interface graphique installée par défaut sous Windows.

✓***RStudio*** un IDE (Environnement de Developpement integré) multiplateforme,

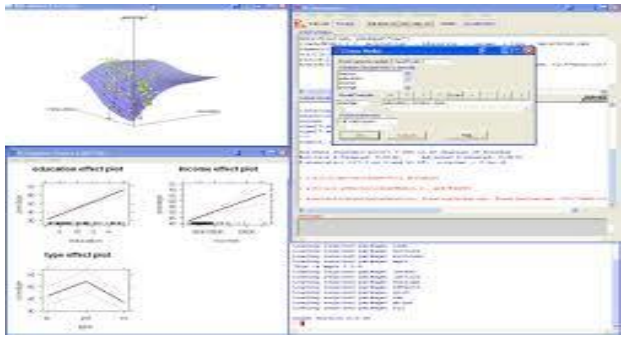
✓***Tinn-R***, un éditeur de texte orienté R sur Windows,

✓ **Jupyter** est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala.

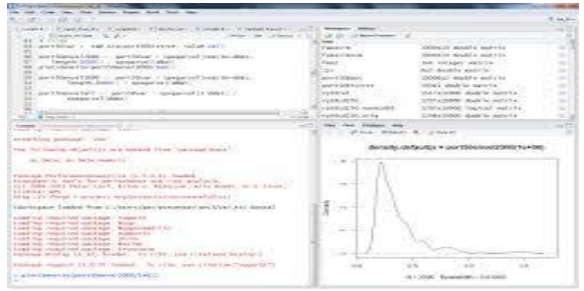
1. Présentation générale du logiciel

1.6. Editeurs pour le langage R

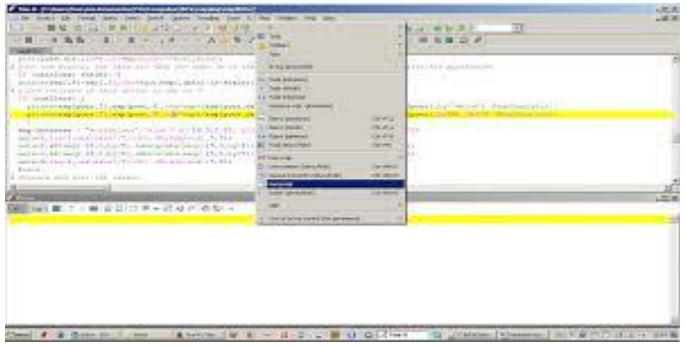
RGUI



RStudio



Tinn-R



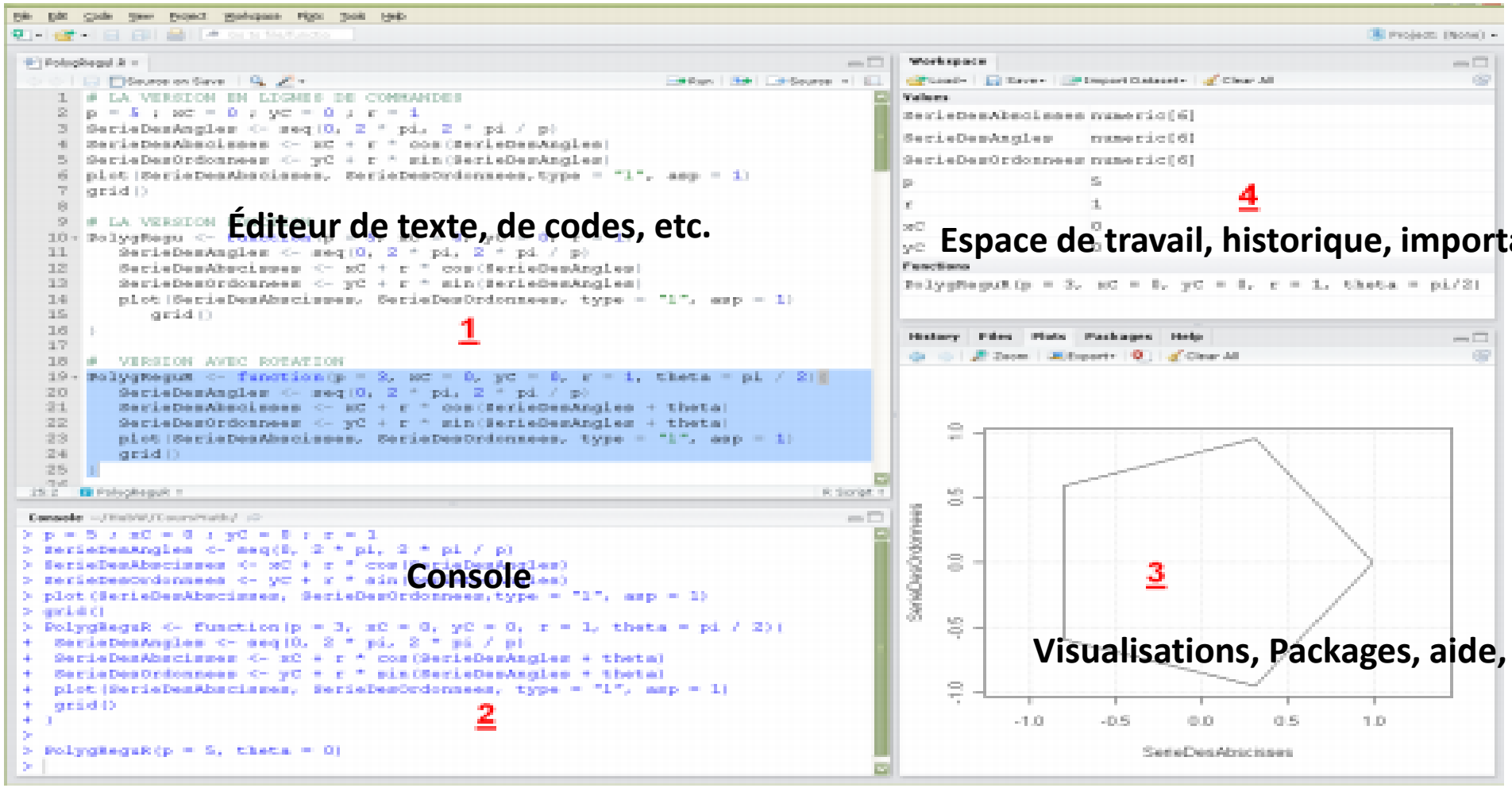
2. Environnement du travail RStudio

2.1. RStudio : Présentation

•**Éditeur de script intelligent :** Coloration syntaxique, complétion automatique, indentation.
•**Console intégrée :** Exécution directe du code R, affichage immédiat des résultats.
•**Gestionnaire d'environnement :** Affiche les variables, jeux de données et objets en mémoire.
•**Onglet Fichiers / Packages / Plots :** Navigation dans les fichiers du projet, Gestion et installation des packages, Visualisation des graphiques générés.
•**Support de projets :**Organisation claire des analyses par dossier.
• **Disponibilité**: en version **Desktop** (locale) et **Server** (utilisable via un navigateur web).
•**Intégration avec Git et GitHub :** Pour le versionnement et le travail collaboratif.
•**Support avancé :** Intégration avec Python, SQL, et création d'applications interactives avec **Shiny**.

2. Environnement du travail RStudio

2.1. RStudio : Présentation



2. Environnement du travail RStudio

2.1. RStudio : Présentation

La fenêtre 1 est celle du traitement de texte à coloration syntaxique.

- **Pour exécuter une ligne de commande(s) :** il suffit de positionner le curseur dans cette ligne (de la fenêtre 1) et de cliquer sur l'icône ***"Run"*** **ou** Ctrl+R**.** Les lignes sont alors copiées dans la console (fenêtre 2) puis exécutées, automatiquement. Le curseur passe alors automatiquement à la prochaine ligne de code.

- **Pour exécuter plusieurs lignes de commandes :** il suffit de les sélectionner et de cliquer sur l'icône ***"Run".*** Les lignes restent sélectionnées et on peut les exécuter à nouveau (pratique pour des simulations).

- **Lorsque l'on veut exécuter l'ensemble des lignes de code du fichier** de la fenêtre de script active, il suffit de cliquer sur l'icône ***"Source".***

2. Environnement du travail RStudio

2.1. RStudio : Présentation

- **Exécution par console** : on peut aussi saisir des commandes directement dans la console, mais ça n'est pas très utile.

```
> 2 + 3
[1] 5
```

```
> x <- exp(2)
> x
[1] 7.389056
```

2. Environnement du travail RStudio

2.1. RStudio : Présentation

- Les objets R (les constantes, vecteurs, fonctions ...), créés en mémoire vive apparaissent dans la fenêtre 4(espace de travail). Cette fenêtre est très utile pour détecter les éventuelles erreurs.
- Les résultats graphiques apparaissent dans la *fenêtre 3*.
- Les derniers graphiques effectués lors d'une session sont enregistrés et on peut les faire défiler en cliquant sur les flèches situées en haut à gauche de la fenêtre.
- Les résultats numériques des lignes de commandes que l'on exécute apparaissent dans la console. Par contre lorsque ces lignes concernent la création d'un fonction, celle-ci est seulement stockée en mémoire, ce que l'on peut voir dans *la fenêtre 4 .*

2. Environnement du travail RStudio

2.2. Répertoire de travail

- A chaque fois qu'on lance R, une nouvelle session est créée. Par défaut R travail dans certain répertoire, typiquement C:\Program Files\R\rw2001.
- La fonction ***getwd()*** permet d'identifier le répertoire de travail par défaut.
- Pour modifier le répertoire de travail, il faut utiliser la fonction ***setwd().***
- En pratique, il faut commencer par lui spécifier un dossier dans le quel on veut travailler, où il va stocker les variables, analyses etc que l'on va utiliser:
  - ✓ menu File/ChangeDir… depuis R, sous windows. Éventuellement copier les données dans ce répertoire
  - ✓ menu Session/Set Working Directory … depuis RStudio

2. Environnement du travail RStudio

2.2. Répertoire de travail

- La commande setwd(dirname(file.choose())) permet d'ouvrir une fenêtre pour choisir le répertoire de travail et permet ainsi d'éviter à avoir à écrire le chemin complet du répertoire dans lequel les fichiers issus de l'analyse sous R seront sauvegardés.
- Bien que la fonction file.choose() soit utilisée dans cette commande, aucun fichier ne sera ouvert mais par contre il faudra s'assurer que le répertoire de travail choisi contienne au moins un fichier avant d'utiliser cette commande.
- Pendant qu'on travaille, R stocke les données produites dans un "espace de travail ("workspace"), et stocke les ordres qu'on lui a passés dans un historique.

2. Environnement du travail RStudio

2.3. Gestion des scripts R

- Notons qu'il n'est pas toujours très pratique de travailler sur la console. La saisie est réalisée ligne à ligne. En cas d'erreur de saisie, vous devez soit saisir une nouvelle commande modifiée, soit la rappeler en la recherchant dans l'historique des commandes pour pouvoir la modifier.
- Un atout important d'un environnement de développement intégré comme RStudio (et, dans une moindre mesure, comme l'interface RGui), est de pouvoir manipuler des scripts.
- Un *script* est un fichier de type texte dans lequel il est possible de saisir directement une séquence d'instruction pour l'exécuter par la suite.

2. Environnement du travail RStudio

2.3. Gestion des scripts R

- Il est possible à tout moment d'exécuter une ligne d'instruction en se positionnant sur cette ligne (sans nécessairement la sélectionner), puis en demandant l'exécution :
  - ✓ Par le bouton *Run*
  - ✓ Par le raccourci *Ctrl+R*
- On peut également exécuter plusieurs ligne à la fois. Pour cela, il faut sélectionner les lignes de commandes désirées et demander l'exécution
- **Exemple:**
  - ✓ Pour cela, cliquez sur ***File > New File > R Script*** ou vous pouvez utiliser le raccourci clavier ***CTRL + MAJ + N***.
  - ✓ Tapez par exemple les instructions suivantes dans l'éditeur :

2. Environnement du travail RStudio

2.3. Gestion des scripts R

```
# Exemple simple de script R
# 1. Créer un vecteur de données (par exemple les notes d'étudiants)
notes <- c(12, 15, 9, 18, 14, 10, 16, 13)

# 2. Afficher les données
print(notes)

# 3. Calculer quelques statistiques de base
moyenne <- mean(notes)      # Moyenne
mediane <- median(notes)    # Médiane
ecart_type <- sd(notes)     # Écart-type
# 2. Afficher les resultats
print(paste("Moyenne :", moyenne))
print(paste("Médiane :", mediane))
print(paste("Écart-type :", ecart_type))
```

3. Gestion des Packages

3.1, Qu’est-ce qu'un package R ?

- Un package R regroupe un ensemble de commandes relatives à domaine particulier d’analyse (représentations graphiques, manipulation de données, modèles spécifiques, etc.).
- R dispose d’un certain nombre de packages de base qui sont installés lors de l’installation de R lui-même.
- Il est possible d’installer des packages additionnels depuis le site principal CRAN à partir de : http://cran.r-project.org/ RStudio.
- Les packages R sont développés et publiés par la plus grande communauté R.

3. Gestion des Packages

3.2. Installation d'un package R ?

1) Installation à partir d'un fichier situé sur le disque avec *RGUI*:

- Vous pouvez par exemple télécharger depuis le site http ***://cran-r-project.org***, le fichier : *R2HTML-numero.zip* permet la création de pages web comportant statistiques, graphiques... et l'enregistrer sur le bureau de windows.
- Allez dans le *menu packages*, puis dans le sous menu Installe package(s) from local file, depuis des fichiers zip.
- sélectionner alors le fichier *R2HTML-numero.zip* sur le bureau de windows ; puis cliquer "ouvrir".

3. Gestion des Packages

3.2. Installation d'un package R ?

2)Installation à partir d'un fichier situé sur le disque avec Rstudio :

- Vous pouvez par exemple télécharger depuis le site http ***://cran-r-project.org***, le fichier : *R2HTML-numero.zip* et l'enregistrer sur le bureau de windows.
- Allez dans le *menu Tools*, puis dans le sous menu Installe package(s) from local file, depuis des fichiers zip.
- sélectionner alors le fichier *R2HTML-numero.zip* sur le bureau de windows ; puis cliquer "ouvrir".

3. Gestion des Packages

3.2. Installation d'un package R ?

3) Installations directement depuis l'internet par commande :

- Les packages peuvent être installés avec la fonction *install.packages()* .
  - ✓ **Exemple :** Le code suivant installe le paquage « tseries »:

    install.packages("tseries")
  - ✓ Cette commande télécharge le package « tseries » du CRAN et l'installe sur votre ordinateur. Tous les packages dont dépend ce package seront également téléchargés et installés.
- Vous pouvez installer plusieurs packages R en même temps avec un seul appel à

  install.packages()
  - ✓ **Exemple :** Placer les noms des packages R dans un vecteur de caractères

    install.packages(c("tseries", "ggplot2", "devtools"))

3. Gestion des Packages

3.2. Installation d'un package R ?

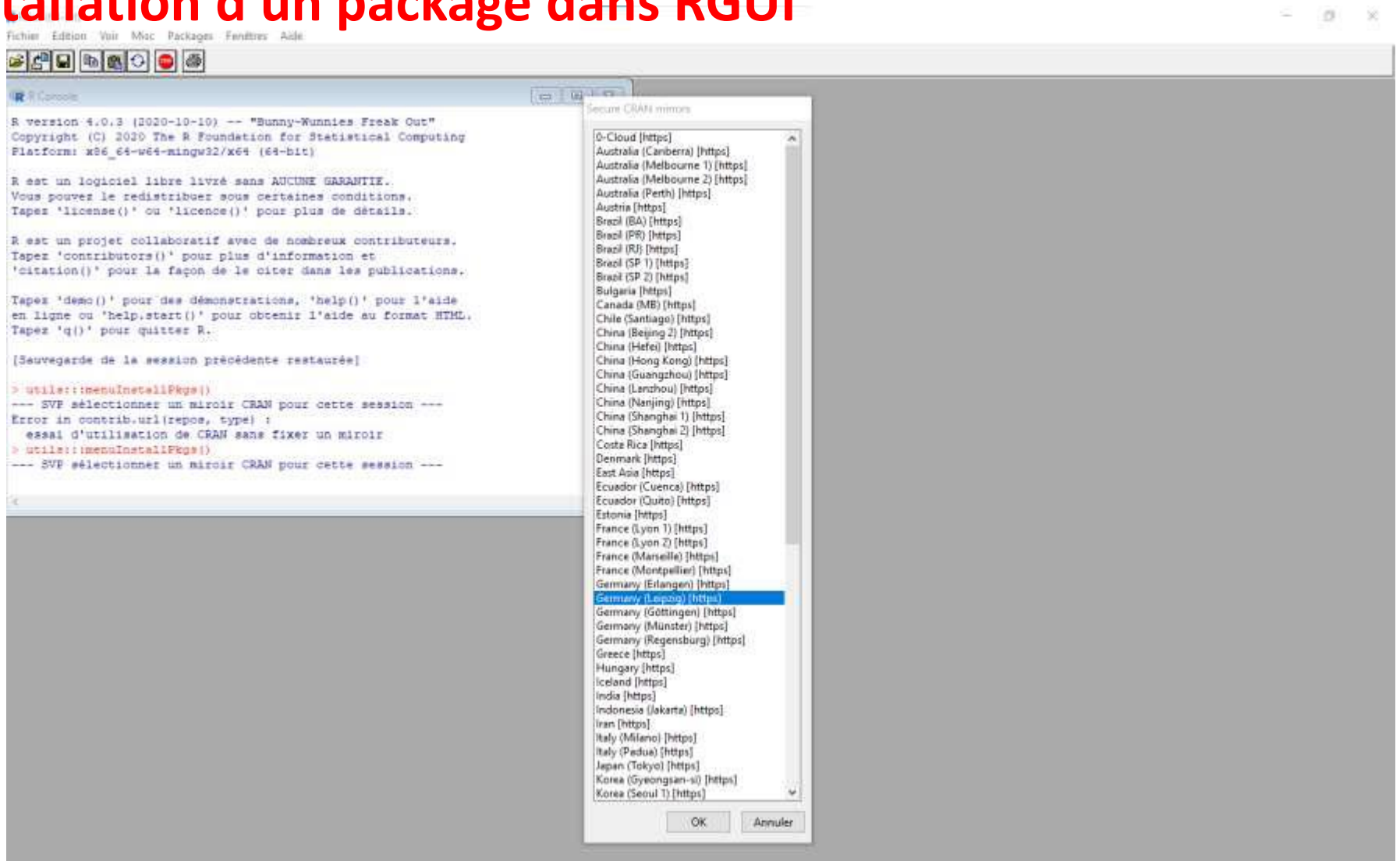
4) Installation d'un package dans RGUI

- Cliquer sur le menu packages, puis dans le sous-menu Installer le(s) package(s), sélectionner un miroir (CRAN mirror) proche de votre situation géographique et cliquer sur « OK » .
- **Exemple :**
  - ✓ Pour installer par exemple les packges ***car*** et ***Rcmdr:***
  - ✓ Sélectionner les entrées (car) et (Rcmdr). Pour cela, cliquer d'abord sur (car) puis glisser l'ascenseur vers le bas et cliquer sur (Rcmdr) tout en maintenant la touche de Ctrl en foncées.
  - ✓ vous devez vérifier que les deux entrées sont bien sélectionnées.

3. Gestion des Packages

3.2. Installation d'un package R ?

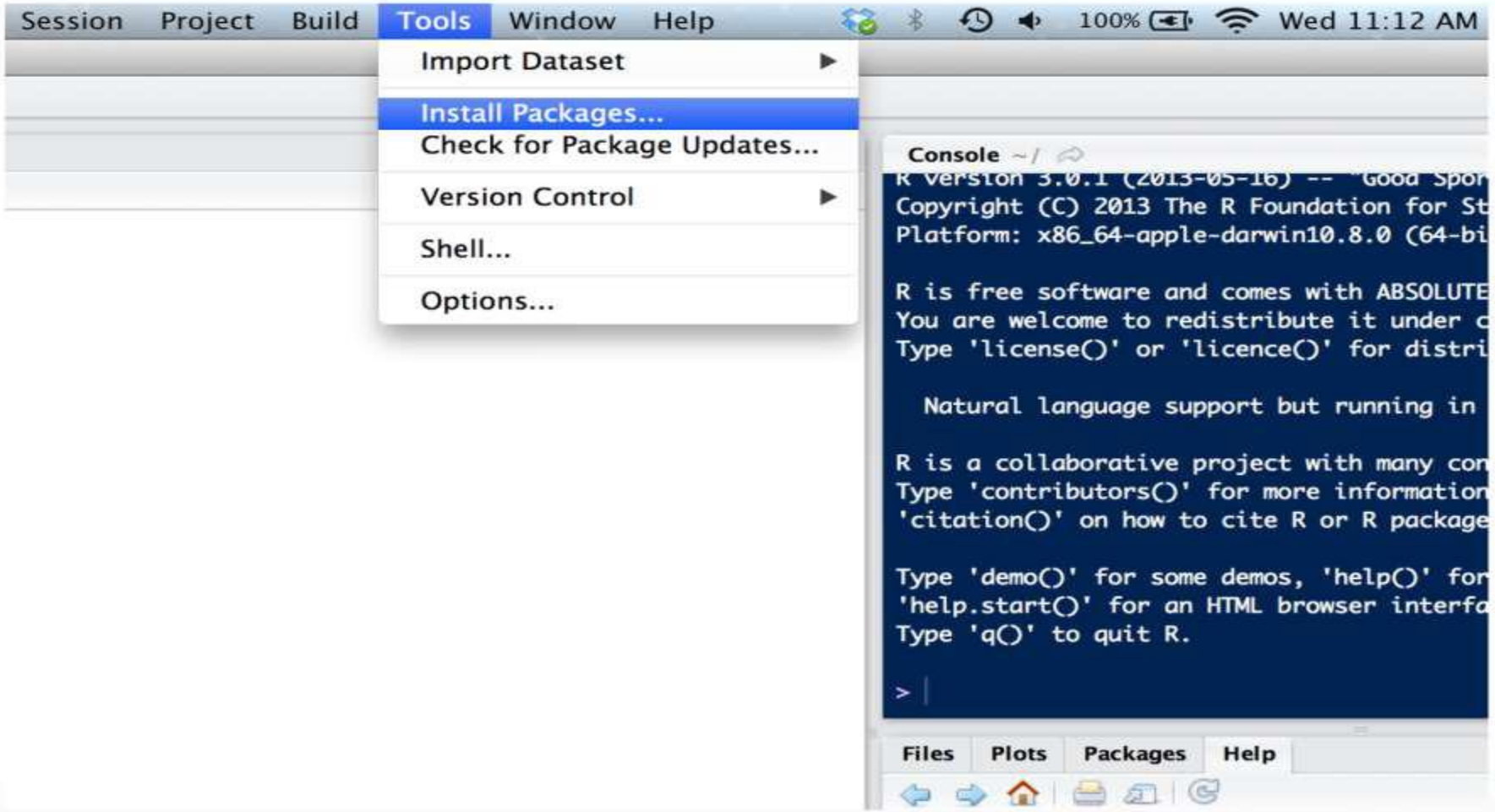
4) Installation d'un package dans RGUI



3. Gestion des Packages

3.2. Installation d'un package R ?

5) Installation d'un package dans Rstudio



3. Gestion des Packages

3.3. Chargement d'un package R ?

- L'installation d'un package ne le met pas immédiatement à votre disposition dans R ;vous devez charger le package

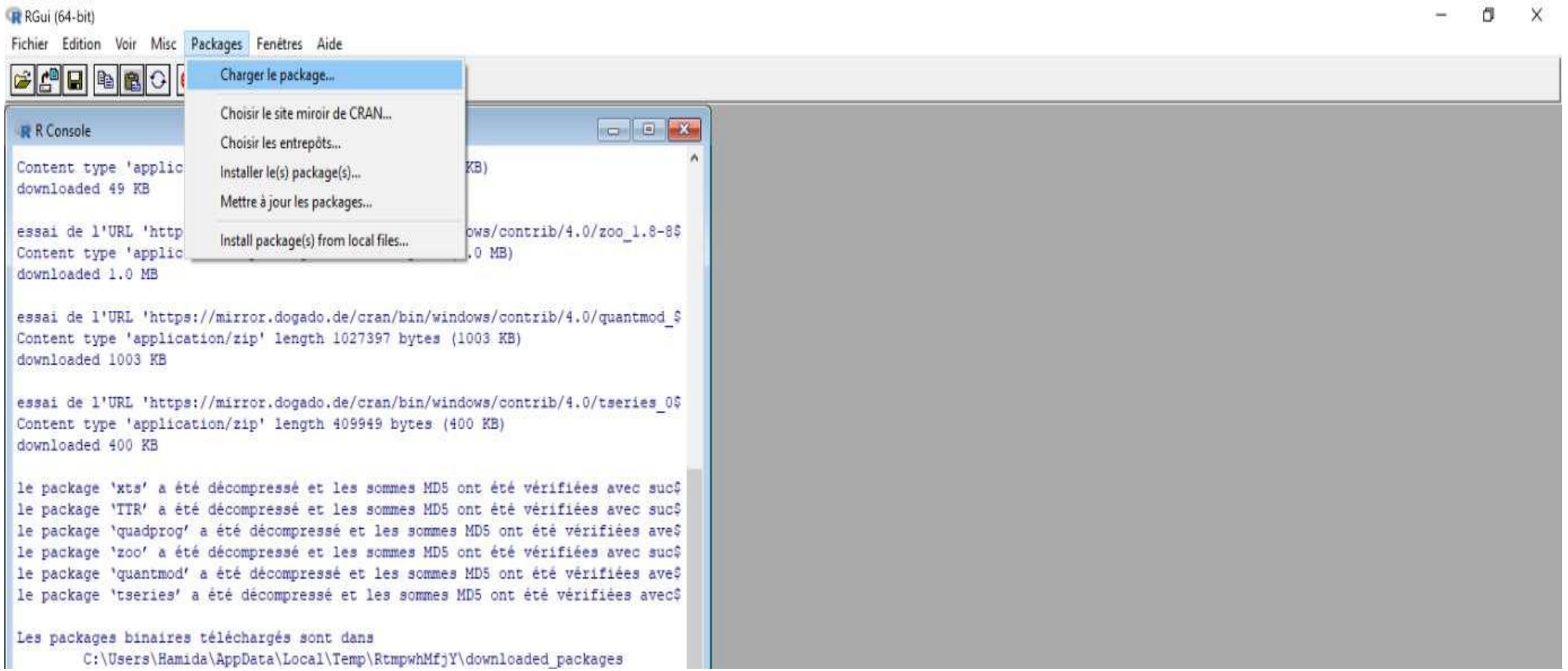
1) Chargement par ligne de commande:

- La fonction ***library()*** est utilisé pour charger les packages dans R
- ***Exemples:***
  - ✓ Le code suivant est utilisé pour charger le package ggplot2 dans R : *library (ggplot2)*
  - ✓ pour charger le package MASS, on tapera : *library(MASS)*
- Tous les packages qui doivent être chargés en tant que dépendances seront chargés en premier, avant que le package nommé ne soit chargé

3. Gestion des Packages

3.3. Chargement d'un package R ?

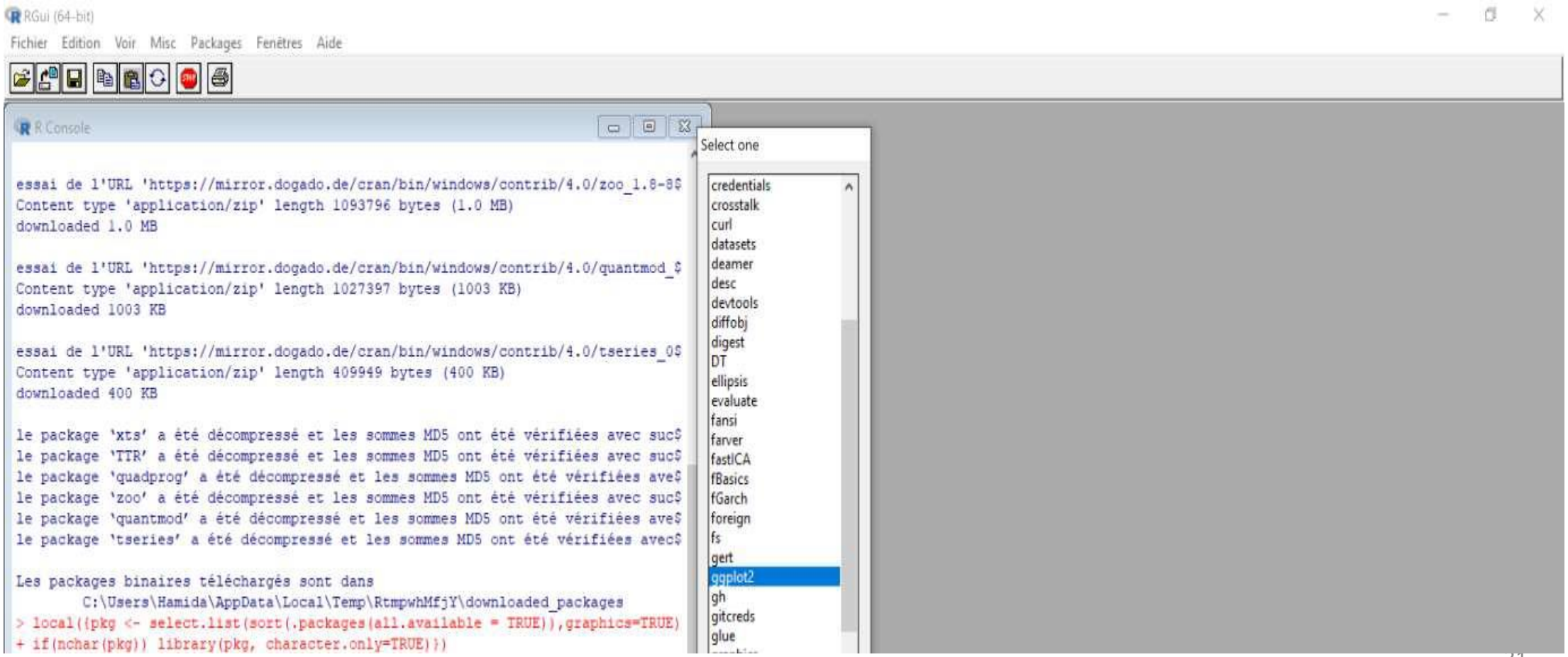
2) Chargement dans RGUI:



3. Gestion des Packages

3.3. Chargement d'un package R ?

2) Chargement dans RGUI:



3. Gestion des Packages

3.4. Commandes d'un package

• Pour connaître l'ensemble des commandes disponibles dans un package, il suffit de taper la commande suivante :

> *help(package=nom)*

> *help(package=FactoMineR)*

• Autres commandes :

.libPaths() # Pour obtenir la location de la library contenant les packages installées
library() # Pour voir tous les packages installés
search() # Pour voir les packages couramment chargés
update.packages()# Pour mettre à jour les packages

3. Gestion des Packages

3.5. Packages R pour Data Mining

Package(s)	Utilisation principale	Exemple d'application
dplyr, tidyr, data.table, readr, readxl	Importation, nettoyage, transformation, préparation des datasets	Charger un CSV, nettoyer données manquantes, filtrer des variables
ggplot2, plotly, corrplot	Visualisation statique et interactive, exploration graphique	Histogrammes, scatter plots, corrélations
caret, mlr3, rpart, randomForest, e1071, xgboost	Entraînement de modèles supervisés (arbres, forêts, SVM, boosting)	Prédire une variable cible (fraude, churn, réussite étudiante)
cluster, factoextra	Segmentation et regroupement de données	K-means, clustering hiérarchique, visualisation de clusters
FactoMineR, factoextra	ACP, AFC, ACM, visualisation factorielle	Résumé d'un dataset avec ACP et représentation graphique
nnet, keras, tensorflow	Modèles neuronaux simples et profonds	Classification d'images, prédiction complexe