



UNIVERSITE DE PARAKOU
ECOLE NATIONALE DE STATISTIQUE DE PLANIFICATION
ET DE DEMOGRAPHIE (ENSPD)



MASTER : 1(SA/PSE)

COURS : Introduction aux logiciels statistiques (SPSS, Stata, R)

Thème:

Les Packages dans R

GROUPE : 21

OLACHIMI M. Samuel

GNIMAVO Senan Fiacre Aurelle

Sous la supervision de:

Dr. SODJINOU Epiphane

INTRODUCTION

R est un langage de programmation et un environnement destiné à l'analyse statistique et à la visualisation de données. Bien que le noyau de R offre déjà plusieurs fonctionnalités de base, sa véritable puissance réside dans l'utilisation des packages. Ces derniers permettent d'étendre considérablement les capacités de R, en apportant des outils spécifiques à divers domaines d'analyse. Dans cet exposé, nous allons définir ce qu'est un package dans R, expliquer comment le gérer, puis présenter quelques exemples de packages les plus couramment utilisés.

QU'EST-CE QU'UN PACKAGE DANS R?

Un package (ou bibliothèque) dans R est une collection de **fonctions**, de **jeux de données** et de **documentation** qui ont été développés pour accomplir une tâche précise. Il s'agit en quelque sorte d'un module complémentaire qui vient enrichir l'environnement R. Chaque package contient un ensemble structuré de fichiers qui permettent à l'utilisateur d'effectuer des analyses spécifiques ou de manipuler les données plus efficacement. Les packages permettent ainsi d'étendre les fonctionnalités de base de R et de s'adapter aux besoins évolutifs des utilisateurs, qu'ils soient statisticiens, économistes, biologistes ou spécialistes des données.

CATÉGORIES DE PACKAGES DANS R

❑ LES PACKAGES DE BASE

Un **package de base** dans R (ou **base package**) est un ensemble de fonctions, de données et de fichiers d'aide qui sont **inclus automatiquement avec R dès son installation**. Ces packages sont **essentiels** pour le fonctionnement de R car ils contiennent les outils fondamentaux pour écrire du code, effectuer des calculs, créer des graphiques, etc.

❑ LES PACKAGES RECOMMANDÉS

Les **packages recommandés** dans R (ou *recommended packages*) sont des **extensions supplémentaires** fournies avec R, mais qui ne sont **pas toujours chargées automatiquement** au démarrage. Ils sont considérés comme **utiles, stables et largement utilisés**, même s'ils ne font pas partie du **noyau fondamental** du langage R (comme les packages de base).

❑ LES PACKAGES EXTERNES

Les **packages externes** (ou *packages utilisateurs / packages additionnels*) sont des **extensions créées par la communauté R** qui ne sont **pas incluses** par défaut avec R. Il faut **les installer soit-même** selon besoins à l'aide de la fonction `install.packages()` par exemple.

OÙ TROUVER LES PACKAGES?

- **CRAN(Comprehensive R Archive Network)** (<https://cran.r-project.org/>) est le dépôt officiel qui héberge des packages validés.
- **Bioconductor** (<https://www.bioconductor.org/>) est spécialisé dans la bioinformatique.
- **GitHub, GitLab, Bitbucket** permettent de récupérer des versions de développement ou expérimentales via des outils comme devtools.

GESTION DES PACKAGES DANS R

❑ INSTALLATION

- **`install.packages("nom_du_package")`**
- Pour installer plusieurs packages à la fois:
`install.packages(c("nom_du_package1","nom_du_package2","..."))`
- **Installer directement à l'aide des onglets (R et Rstudio)**
- On peut aussi installer un package déjà téléchargé: `packages:....fichiers locaux`
(Windows: .zip, macOS: .tgz; Linux/Unix: .tar.gz)
- Afficher les packages installés: **`rownames(installed.packages())`**

GESTION DES PACKAGES DANS R

❑ CHARGEMENT

Une fois installé, le package doit être chargé pour que ses fonctions soient disponibles. Cela se fait à l'aide de la commande :

library(nom_du_package)

NB: Pour installer et charger en même temps: **pacman::p_load()**

❑ MISE À JOUR

- Pour bénéficier des dernières fonctionnalités il est conseillé de mettre régulièrement à jour les packages: **update.packages()**
- Pour afficher la liste des packages non à jour: **old.packages()**

❑ SUPPRESSION

Si un package devient inutile, il est possible de le supprimer avec:

remove.packages("nom_du_package")

EXEMPLES DE PACKAGES POPULAIRES

- **ggplot2**: pour la création de graphiques avancés.
- **dplyr**: pour manipuler les données (filtrer, trier, résumer, etc.).
- **tidyr**: pour transformer et organiser les données.
- **readr / readxl**: pour importer des données au format CSV ou Excel.
- **shiny**: pour créer des applications web interactives avec R.
- **caret**: pour les méthodes d'apprentissage automatique.

AVANTAGES DE L'UTILISATION DES PACKAGES

- Gain de temps: des fonctions prêtes à l'emploi.
- Communauté active: des milliers de développeurs partagent leurs outils.
- Flexibilité: adaptation à des besoins très spécifiques.
- Reproductibilité: utilisation de packages standards pour documenter et répéter les analyses.

CONCLUSION

Les packages jouent un rôle central dans l'utilisation de R. Ils permettent aux utilisateurs d'adapter R à une grande variété de contextes d'analyse. Maîtriser leur installation, leur chargement et leur gestion est essentiel pour travailler efficacement avec ce langage. Enfin, grâce à une communauté dynamique, l'écosystème R ne cesse d'évoluer, offrant ainsi toujours plus d'outils performants aux utilisateurs.