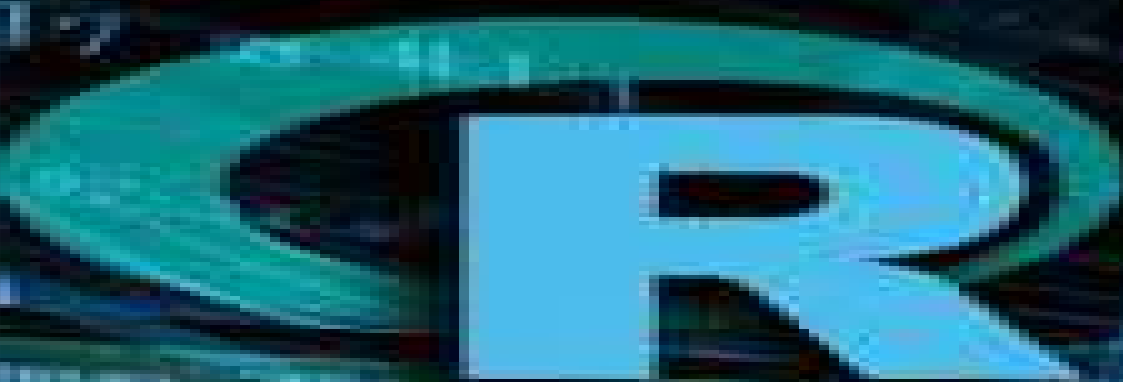




**Université Mohamed Khider - Biskra**  
**Faculté des Sciences Exactes et SNV**  
**Département de Mathématiques**



# Initiation à



**YAHIA Djabrane**

**11/12/2023**



## Plan de la présentation

- Que sont **R** et **CRAN** ?
- Installation
- Avantages de **R**
- Packages
- Fonctions les plus utilisées dans **R**





3



## Que sont R et CRAN ?

- **R** est un langage et un environnement disponible **gratuitement** pour le calcul statistique et le graphisme: **modélisation linéaire et non linéaire, tests statistiques, analyse de séries chronologiques, SNP, etc.**
- **Conçu Par : Ross Ihaka, Robert Gentleman en 1993.**



## Que sont R et CRAN ?

- **CRAN (Comprehensive R Archive Network)** est un réseau de serveurs **Web** à travers le monde qui stockent des versions identiques et à jour du code et de la documentation pour **R**.
- Pour plus d'informations, veuillez consulter la page d'accueil du projet **R** : <https://cran.r-project.org/>





5

# Installation



- Veuillez télécharger **R** depuis :

<https://cran.r-project.org/bin/windows/base/>

Dernière version : 4.3.2 (31 octobre 2023)

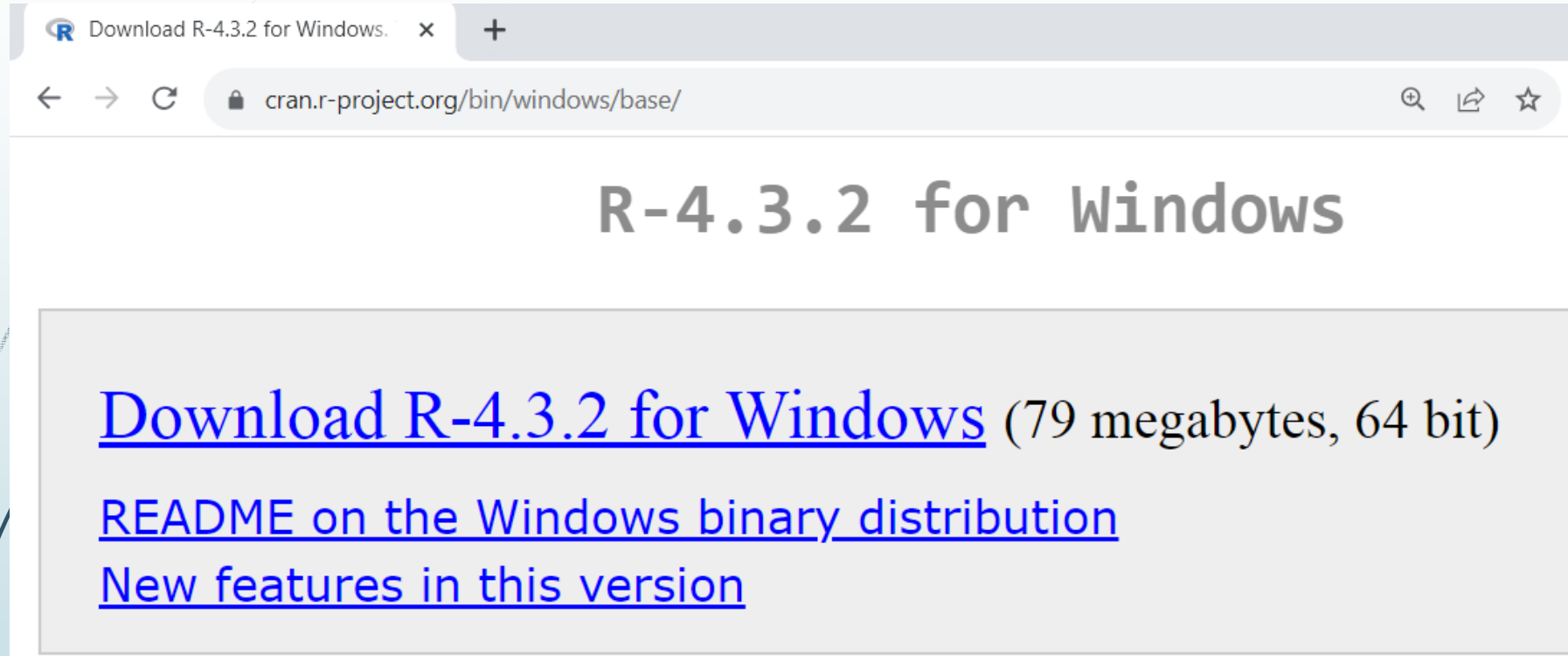
- Après installation (simple et rapide) vous pouvez ajouter : **Rstudio** (espace de travail)





6

# Installation



The screenshot shows a web browser window with the address bar displaying "cran.r-project.org/bin/windows/base/". The main heading is "R-4.3.2 for Windows". Below the heading, there are three blue underlined links: "Download R-4.3.2 for Windows (79 megabytes, 64 bit)", "README on the Windows binary distribution", and "New features in this version".





## Avantages de R

- **R** est un langage activement développé (une nouvelle version majeure tous les 2 ans) et plus de 1000 nouveaux packages publiés par an sur le **CRAN**
- **R** est entièrement gratuit, de code source ouvert.
- Les packages **R** s'enrichit au fil des années.
- Il n'y a pas de **date d'expiration** de **R** (licence,...).





# Packages



➤ Actuellement, le référentiel de packages **CRAN** contient **20141 packages** disponibles (12/2023).

➤ Pour ajouter (**installer**) un package:

Menu → Packages → `install.packages()`

➤ Pour charger (**utiliser**) des packages: `library()`

**Exemple: `library(KernSmooth)`**

**Menu → Packages → charger packages(KernSmooth)**







## Aide et fonctions de base

- **help(texte)** : Affiche l'aide relative à texte
- **help.start( )** : Lance la version **HTML** de l'aide
- Menu → Aide → Fonctions **R** texte
- Menu → Aide → Manuels (**en PDF**)
- **Utilisation du #** : Pour commenter,  
**Exemple : rexp(100,2) # v.a de loi Exp( $\lambda=2$ ) et de taille 100**





## Espace de travail R

- Il est nécessaire de lancer **R** et de sauvegarder vos propres programmes, fonctions résultats, graphes,... dans un fichier

(document de travail):

- `setwd( )` et `getwd( )`

- **Exemple:**

```
setwd("D:/R-YAHIA")
```

```
getwd( )
```

```
[1] "D:/R-YAHIA"
```



# Fonctions les plus utilisées dans R



## 1) Lecture de données :

- **read.csv()** : Pour importer des fichiers CSV.
- **read.table()** : Pour importer des données sous forme de table.
- **read.xlsx()** : Pour importer des fichiers Excel.

- 
- **write.csv()** : Pour exporter des données au format CSV.
  - **write.table()** : Pour exporter des données au format de table.

- 
- **data()** : pour la liste des données existant dans R



# Fonctions les plus utilisées dans R



## 2) Opérations de base

**Addition :  $3+5$**

**Soustraction :  $4 - 1$**

**Multiplication:  $3 * 4$**

**Division :  $(4 - 1)/2$**

**Exposant (puissance) :  $2^{(-1/5)}$**

**$t(x)$  transposée**

**$diag(x)$  diagonale**

**$\%*\%$  produit de matrices**



# Fonctions les plus utilisées dans R



## 3) Création de données

`c(1,-2, 3.09, ...)` ou `c("a", "b", ...)` : Vecteur

`from:to` : génère une séquence (**1:20** donne 1,2,3,...,19,20)

`numeric(n)`: génère un **vecteur** de taille **n**

`matrix(x, nrow=,ncol=)`: crée une matrice **matrix(1:12,3,4)**

---

`seq(from,to, by=h )`: **by** est le **pas** → `seq(0,1, by=0.1)`

`seq(from,to, length= n)`: **length** et la longueur (ou taille)

**Exemple :** `seq(0,1, length =100)` → 0, 0.01, 0.02,...,1



# Fonctions les plus utilisées dans R



## 4) Vecteurs et listes

$x[i]$  : l'éléments **i** de la liste **x**  
 $x[[j]]$  : **j-ième** élément de la liste **x**  
 $x[["A"]]$  : l'élément de **x** nommé "**A**"  
 $x\$A$  : l'élément de **x** nommé "**A**"

---

$x[-k]$  : tous sauf le **k-ième** élément  
 $x[1:n]$  : les **n premier** éléments  
 $x[x > 0]$  : tous les éléments **> 0**  
 $x[x >= 0 \ \& \ x < 10]$  : tous les éléments  **$\geq 3$  et  $< 10$**





# Fonctions les plus utilisées dans R



## 5) Matrices

$x[i, j]$  l'élément de la **ligne i, colonne j**

$x[i, ]$  toute la **ligne i**

$x[, j]$  toute la **colonne j**

$x[1:4, ]$  les lignes de **1 à 4**

$x[, c(1,3)]$  les colonnes **1 et 3**

$x["A", ]$  la ligne nommée **"A"**



# Fonctions les plus utilisées dans R



## 6) Modèles (1)

**density(x)** estimateur à noyaux de la densité

**lm(formula)** ajuste un modèle linéaire

**summary(M)** pour les caractéristiques du modèle **M**

**Exemple:**  $x = \text{rexp}(100)$  ;  $y = 2*x+4+rnorm(100)$   
 $M = \text{lm}(y \sim x)$  ; **summary(M)**

### Coefficients:

	Estimate	Std. Error	t value	Pr(>  t )
(Intercept)	4.05976	0.13030	31.16	<2e-16 ***
x	1.97724	0.09659	20.47	<2e-16 ***

Residual standard error: 0.9557 on 98 degrees of freedom, Multiple R-squared: 0.8105, Adjusted R-squared: 0.8085  
F-statistic: 419 on 1 and 98 DF, p-value: < 2.2e-16







# Fonctions les plus utilisées dans R



## 6) Modèles (2)

- **Optimisation d'une fonction `fn`:**

`optimize(fn, interval)`, `optim(par, fn)`, `which.min(fn)`,...

- **Analyse de la variance : `aov`, `anova`**

- **Tests : `t.test`, `chisq.test`, `ks.test`, `shapiro.test` ....**

- **Prédiction et intervalles de confiance: `predict(fit, ...)`**





# Fonctions les plus utilisées dans R



## 7) Statistique

Les fonctions suivantes peuvent s'utiliser en remplaçant la lettre **r** avec **d**, **p**, **q** pour obtenir, respectivement:

- **r** → tirage de **n réalisations** d'une variable aléatoire,
- **d** → la **densité**, **p** → Fonction de **répartition**, **q** → **quantiles**

**Exemples :**

**pt(1, 8) → Distribution de Student à 8 ddl au point  $x = 1$**

**dexp(3.14, 3) → Densité Expo. de paramètre 3, au point  $x = 3.14$**

**rnorm(100, 0, 2) → 100 valeurs de  $N(0,4)$**

**qf(0.95, 3, 12) → 95% quantile de Fisher à 3 et 12 ddl.**





# Fonctions les plus utilisées dans R



## 8) Fonctions de Calcul

$\text{max}(x)$  = maximum,

$\text{min}(x)$  = minimum,

$\text{sum}(x)$  = somme,

$\text{prod}(x)$  = produit,

$\text{mean}(x)$  = moyenne (espérance),  $\text{sd}(x)$  = écart-type

---

$\text{var}(x)$  = variance empirique **corrigée**

$\text{cov}(x, y)$  = covariance entre x et y

$\text{cor}(x, y)$  = corrélation linéaire entre x et y

$\text{round}(x, k)$  = arrondit les éléments de x à k décimales

**$\text{round}(10.248706, 2)$  donne 10.25**





# Fonctions les plus utilisées dans R



20

## 9) Graphiques et figures (1)

`x11()` ouvre une **nouvelle fenêtre** graphique

`plot(x, y)` trace **y en fonction de x**

`lines(...)` ajoute une **courbe** à plot

`hist(x)` **histogramme** des fréquences de **x**

`curve(f,...)` trace une **fonction** d'expression **f**

`points(x, y)` ajoute des **points** aux coordonnées **x et y**

`abline(h=0.8)` trace une **ligne horizontale** à l'ordonnée **h**

`abline(v=1.25)` trace une **ligne verticale** à l'abscisse **v**





# Fonctions les plus utilisées dans R



21

## 9) Graphiques et figures (2)

**main=**, **xlab=**, **ylab=** **titres** du graphe et des **axes**

**legend(x, y, ...)** ajoute une **légende (étiquète)** au point (x,y)

**text(x, y, texte, ...)** ajoute le **texte** aux point (x,y)

**col=** contrôle la **couleur** des courbes

**lty=** contrôle le **type** de **ligne**

**lwd =** contrôle **l'épaisseur** des lignes





# Fonctions les plus utilisées dans R



## 10) Programmation (1)

**function( arglist ) {expr} : **arglist** est une liste d'arguments, **expr** est une expression exécutée**

**Exemple : Fn=function(X, a) {a\*sd(X)}**

**X=rnorm(100); a=2; Fn(X, a) → 1.997**

**ifelse(cond, cons.expr, alt.expr): si et sinon**

**Exemple:**

**Ind=function(t){ifelse(abs(t)<1, 1, 0)}**

**Ind(X)**





# Fonctions les plus utilisées dans R



## 10) Programmation (2)

`for(var in seq) {expr}` exécute l'expression pour chaque valeur d'une séquence (**boucle**)

**Exemple:**

```
F = numeric(100)
X = rexp(100, 2)
for(j in 1:100) {
  F[j] = sum(X[1:j])
}
```





# Fonctions les plus utilisées dans R



## 10) Programmation (3)

`repeat {expr {break} }` : arrête une boucle **for, while ou repeat**

**Exemple :**

**N=200**

**repeat{ T=rexp(N,1.5) ; X=rnorm(N,2,1)**

**X1=X[X<=T]; T1=T[X<=T]**

**n=length(X1); a=n/N**

**if((a>=.249)&(a<=.251)){break}}**





# Fonctions les plus utilisées dans R



## 10) Programmation (4)

**T-C-L:**  $Z = \sqrt{n} \frac{\bar{X} - \mu}{\sigma} \xrightarrow{Loi} N(0, 1)$ , quand  $n \rightarrow \infty$ .

**N=300**

**Z=numeric(N)**

**m= 3; s=2; n=100**

**for(j in 1:N) {**

**X=rnorm(n,m,s)**

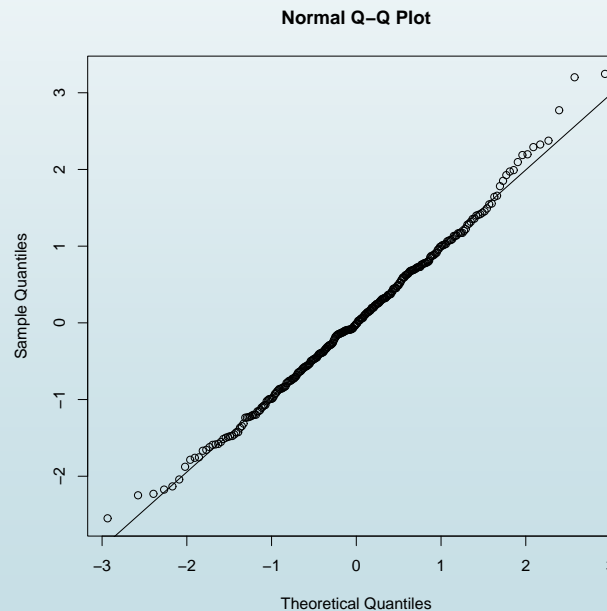
**Z[j]=sqrt(n)\*(mean(X)-m)/s }**

**qqnorm(Z); qqline(Z)**

**chisq.test(Z,rnorm(N))**

**ks.test(Z,rnorm(N))**

**shapiro.test(Z)**



**Pearson's Chi-squared test**  
X-squared = 89700, df = 89401,  
p-value = **0.2395**

**Kolmogorov-Smirnov test**  
D = 0.066667, p-value = **0.5176**

**Shapiro-Wilk normality test**  
W = 0.995, p-value = **0.441**



**Merci pour votre attention  
et j'espère que mon discours a été  
à la hauteur de vos attentes.**

