

Logiciel R — niveau débutant

Hervé Seitz (email : herve.seitz@igh.cnrs.fr)

IGH (UMR 9002 CNRS et université de Montpellier)

5–6 juin 2023

Ce diaporama est accessible à :

http://www.igh.cnrs.fr/equip/Seitz/R_deb.pdf

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

L'interface

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Le *prompt* (caractère « > » en début de ligne) signifie que **R** attend que vous tapiez une commande.

L'interface

La commande peut être un calcul :

```
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 4*3
[1] 12
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La commande peut être un calcul :

```
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 4*3
[1] 12
>
```

(taper « Entrée » pour exécuter la commande)

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La commande peut être un calcul :

```
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 4*3
[1] 12
>
```

De nombreux opérateurs mathématiques sont disponibles
(+, -, *, /, ^, ...).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La commande peut être un calcul :

```
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 4*3
[1] 12
>
```

De nombreux opérateurs mathématiques sont disponibles
(+, -, *, /, ^, ...).

De nombreuses fonctions (sqrt, cos, log, ...) et la constante
 π (pi).

L'interface

La commande peut être un calcul :

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> 28/3  
[1] 9.333333  
> 9^2  
[1] 81  
> sqrt(81)  
[1] 9  
> cos(20)  
[1] 0.4080821  
> pi  
[1] 3.141593  
>
```

De nombreux opérateurs mathématiques sont disponibles
(+, -, *, /, ^, ...).

De nombreuses fonctions (sqrt, cos, log, ...) et la constante
 π (pi).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La commande peut être un calcul :

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> 28/3  
[1] 9.333333  
> 9^2  
[1] 81  
> sqrt(81)  
[1] 9  
> cos(20)  
[1] 0.4080821  
> pi  
[1] 3.141593  
>
```

De nombreux opérateurs mathématiques sont disponibles (+, -, *, /, ^, ...).

De nombreuses fonctions (sqrt, cos, log, ...) et la constante π (pi). Les arguments d'une fonction sont donnés entre parenthèses (ex. : « sqrt(81) »).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La ligne donnant le résultat commence par « [1] »...

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La ligne donnant le résultat commence par « [1] »...

```
Type 'q()' to quit R.
> 28/3
[1] 9.333333
> 9^2
[1] 81
> sqrt(81)
[1] 9
> cos(20)
[1] 0.4080821
> pt
[1] 3.141593
> 1:100
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
[19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
[55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
[73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
[91] 91 92 93 94 95 96 97 98 99 100
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface

La ligne donnant le résultat commence par « [1] »...

```
Type 'q()' to quit R.
> 28/3
[1] 9.333333
> 9^2
[1] 81
> sqrt(81)
[1] 9
> cos(20)
[1] 0.4080821
> pi
[1] 3.141593
> 1:100
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
[19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
[55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
[73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
[91] 91 92 93 94 95 96 97 98 99 100
>
```

C'est l'indice du premier élément du résultat affiché.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Exécution non-interactive

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Exécution non-interactive

La commande *source()* permet d'exécuter la liste de commandes écrites (en texte brut) dans un fichier externe.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Exécution non-interactive

La commande *source()* permet d'exécuter la liste de commandes écrites (en texte brut) dans un fichier externe.

Exercice : écrire, en texte brut, des commandes R dans un fichier à faire appeler par **R**.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Exécution non-interactive

La commande *source()* permet d'exécuter la liste de commandes écrites (en texte brut) dans un fichier externe.

Exercice : écrire, en texte brut, des commandes R dans un fichier à faire appeler par **R**.

Pour ne pas avoir à taper entièrement les commandes à la main (fastidieux, source de fautes de frappe) : après avoir tapé les premières lettres, la touche « Tabulation » permet de compléter automatiquement le nom de la commande si les premières lettres ne sont pas ambiguës.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Exécution non-interactive

La commande *source()* permet d'exécuter la liste de commandes écrites (en texte brut) dans un fichier externe.

Exercice : écrire, en texte brut, des commandes R dans un fichier à faire appeler par **R**.

Pour ne pas avoir à taper entièrement les commandes à la main (fastidieux, source de fautes de frappe) : après avoir tapé les premières lettres, la touche « Tabulation » permet de compléter automatiquement le nom de la commande si les premières lettres ne sont pas ambiguës.

Si elles sont ambiguës (plusieurs noms de commandes commencent par ces lettres) : taper deux fois « Tabulation » pour afficher les possibilités.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Les nombres sont représentés par des symboles.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Les nombres sont représentés par des symboles. Permet d'assigner une valeur à une variable, puis d'exécuter des calculs dessus.

Calcul symbolique

Les nombres sont représentés par des symboles. Permet d'assigner une valeur à une variable, puis d'exécuter des calculs dessus.

```
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=15
> x/5
[1] 3
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Les nombres sont représentés par des symboles. Permet d'assigner une valeur à une variable, puis d'exécuter des calculs dessus.

```
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=15
> x/5
[1] 3
>
```

Contraintes sur les noms de variables :

- ▶ ne doivent contenir que lettres, chiffres, « . » et « _ » ;
- ▶ ne peuvent pas commencer par un chiffre, un « _ » ou un « . » suivi d'un chiffre.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Les nombres sont représentés par des symboles. Permet d'assigner une valeur à une variable, puis d'exécuter des calculs dessus.

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=15
> x/5
[1] 3
> +e=3
Error in +e = 3 : object 'e' not found
> 4x=5
Error: unexpected symbol in "4x"
> .t=10
> .t
[1] 10
> .2t=30
Error: unexpected symbol in ".2t"
> _x=20
Error: unexpected input in "_"
>
```

Contraintes sur les noms de variables :

- ▶ ne doivent contenir que lettres, chiffres, « . » et « _ » ;
- ▶ ne peuvent pas commencer par un chiffre, un « _ » ou un « . » suivi d'un chiffre.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Calcul symbolique

Assignment d'une valeur à une variable : avec l'opérateur
« = » ou avec « < - ».

Calcul symbolique

Assignment d'une valeur à une variable : avec l'opérateur « = » ou avec « < - ».

Exercice : opérations mathématiques simples sur des variables.

Commandes et fonctions

Logiciel R

H. Seitz



Entrer des
commandes

**Commandes et
fonctions**

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Une commande n'a pas toujours besoin d'un argument.

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Une commande n'a pas toujours besoin d'un argument.

Exemples : *q()*, *list.files()*, *sink()*, ...

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Une commande n'a pas toujours besoin d'un argument.

Exemples : *q()*, *list.files()*, *sink()*, ...

Les commandes de **R** : des fonctions mathématiques, et d'autres commandes.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Une commande n'a pas toujours besoin d'un argument.

Exemples : *q()*, *list.files()*, *sink()*, ...

Les commandes de **R** : des fonctions mathématiques, et d'autres commandes.

Obtenir de l'aide sur une commande : avec « ? » (ex. : *?sink*).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Une fonction prend un (ou : des) argument(s), et retourne un résultat.

Exemples : *cos()*, *sqrt()*, ...

Une commande n'a pas toujours besoin d'un argument.

Exemples : *q()*, *list.files()*, *sink()*, ...

Les commandes de **R** : des fonctions mathématiques, et d'autres commandes.

Obtenir de l'aide sur une commande : avec « ? » (ex. : *?sink*).

Rappeler une commande précédente : avec la touche « flèche du haut ».

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Toutes les fonctions ne viennent pas d'office avec l'installation de base de **R**.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Toutes les fonctions ne viennent pas d'office avec l'installation de base de **R**.

Des « *packages* », à installer, permettent d'ajouter de nouvelles fonctionnalités.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Toutes les fonctions ne viennent pas d'office avec l'installation de base de **R**.

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages('car')
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Toutes les fonctions ne viennent pas d'office avec l'installation de base de **R**.

```
* installing *source* package 'car' ...
** package 'car' successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (car)

The downloaded source packages are in
  '/tmp/RtmpNCbT6r/downloaded_packages'
> library(car)
Loading required package: carData
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Commandes et fonctions

Toutes les fonctions ne viennent pas d'office avec l'installation de base de **R**.

```
* installing *source* package 'car' ...
** package 'car' successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (car)

The downloaded source packages are in
  '/tmp/RtmpNCbT6r/downloaded_packages'
> library(car)
Loading required package: carData
>
```

Installé avec « *install.packages()* » (à faire une fois par ordinateur et par package) ; appelé avec « *library* » (à faire à chaque session **R** faisant appel à ses fonctions).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

- Les valeurs numériques.

Types de variables

- ▶ Les valeurs numériques.
- ▶ Les chaînes de caractères (ex. : annotations de graphes).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

- ▶ Les valeurs numériques.
- ▶ Les chaînes de caractères (ex. : annotations de graphes).
- ▶ Les booléens (ex. : utilisés pour conditionner une commande).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

- ▶ Les valeurs numériques.
- ▶ Les chaînes de caractères (ex. : annotations de graphes).
- ▶ Les booléens (ex. : utilisés pour conditionner une commande).
- ▶ Les facteurs (ex. : utilisés pour catégoriser les données dans un test statistique).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

- ▶ Les valeurs numériques.
- ▶ Les chaînes de caractères (ex. : annotations de graphes).
- ▶ Les booléens (ex. : utilisés pour conditionner une commande).
- ▶ Les facteurs (ex. : utilisés pour catégoriser les données dans un test statistique).

```
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> x=34  
> y='Bonjour'  
> z=x<40  
> a=as.factor('Untreated')  
> x  
[1] 34  
> y  
[1] "Bonjour"  
> z  
[1] TRUE  
> a  
[1] Untreated  
Levels: Untreated  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».

```
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels.

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> y=array(c(1,1,4.5,6,2.3,0,-2.5,0.1),dim=c(4,2))
> y
      [,1] [,2]
[1,] 1.0  2.3
[2,] 1.0  0.0
[3,] 4.5 -2.5
[4,] 6.0  0.1
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels. Extraire un élément d'un vecteur ou tableau : avec « `[]` ».

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels. Extraire un élément d'un vecteur ou tableau : avec « `[]` ».

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> y=array(c(1,1,4.5,6,2.3,0,-2.5,0.1),dim=c(4,2))
> y
      [,1] [,2]
[1,] 1.0  2.3
[2,] 1.0  0.0
[3,] 4.5 -2.5
[4,] 6.0  0.1
> y[3,2]
[1] -2.5
> x[6]
[1] 0
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels. Extraire un élément d'un vecteur ou tableau : avec « `[]` ».
- ▶ Des *lists* et *data.frames* : associent des variables de différentes natures (nombres, chaînes de caractères, ...).

[Entrer des commandes](#)[Commandes et fonctions](#)[Types de variables](#)[Entrée de données](#)[L'interface RStudio](#)[Structuration des calculs](#)[Tests statistiques usuels](#)[Structures de contrôle](#)[Tracer des graphiques](#)[Fichiers de sortie](#)[Conclusion](#)

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels. Extraire un élément d'un vecteur ou tableau : avec « `[]` ».
- ▶ Des *lists* et *data.frames* : associent des variables de différentes natures (nombres, chaînes de caractères, ...).

```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> list_example=list(x,annot)
> list_example
[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1

[[2]]
[1] "treated" "treated" "untreated" "treated" "untreated" "treated"
[7] "untreated" "untreated"
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Structuration des variables :

- ▶ Des « vecteurs » (*i.e.* : liste ordonnée de valeurs) : saisis manuellement par la commande « `c()` ».
- ▶ Des tableaux : vecteurs multidimensionnels. Extraire un élément d'un vecteur ou tableau : avec « `[]` ».
- ▶ Des *lists* et *data.frames* : associent des variables de différentes natures (nombres, chaînes de caractères, ...).

```
[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1

[[2]]
[1] "treated" "treated" "untreated" "treated" "untreated" "treated"
[7] "untreated" "untreated"

> df_example=as.data.frame(list_example)
> names(df_example)=c('Value','Annotation')
> df_example
  Value Annotation
1  1.0    treated
2  1.0    treated
3  4.5  untreated
4  6.0    treated
5  2.3  untreated
6  0.0    treated
7 -2.5  untreated
8  0.1  untreated
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Dans une *list*, les différentes composantes peuvent avoir des nombres d'éléments différents (ex. : 8 éléments dans « x », 9 éléments dans « annot »).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Dans une *list*, les différentes composantes peuvent avoir des nombres d'éléments différents (ex. : 8 éléments dans « x », 9 éléments dans « annot »).

Dans un *data.frame*, les différentes composantes doivent toutes avoir le même nombre d'éléments.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Extraire une composante d'une *list* : avec les doubles crochets « `[[]]` ».

Types de variables

Extraire une composante d'une *list* : avec les doubles crochets « `[[]]` ».

D'un *data.frame* : avec le symbole dollar « `$` ».

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Extraire une composante d'une *list* : avec les doubles crochets « `[[]]` ».

D'un *data.frame* : avec le symbole dollar « `$` ».

```
2  1.0  treated
3  4.5  untreated
4  6.0   treated
5  2.3  untreated
6  0.0   treated
7 -2.5  untreated
8  0.1  untreated
> list_example
[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1

[[2]]
[1] "treated" "treated" "untreated" "treated" "untreated" "treated"
[7] "untreated" "untreated"

> list_example[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> df_example$value
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Extraire une composante d'une *list* : avec les doubles crochets « `[[]]` ».

D'un *data.frame* : avec le symbole dollar « `$` ».

```
2  1.0  treated
3  4.5  untreated
4  6.0   treated
5  2.3  untreated
6  0.0   treated
7 -2.5  untreated
8  0.1  untreated
> list_example
[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1

[[2]]
[1] "treated" "treated" "untreated" "treated" "untreated" "treated"
[7] "untreated" "untreated"

> list_example[[1]]
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> df_example$value
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
>
```

Exercice : extraire le 3^{ème} élément de la 1^{ère} composante d'une *list*, et d'un *data.frame*.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Types de variables

Convertir le type de variable (ex. : le nombre « 2 » à considérer comme le caractère « 2 ») : commandes *as.numeric()*, *as.character()*, *as.factor()*, ...

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Convertir le type de variable (ex. : le nombre « 2 » à considérer comme le caractère « 2 ») : commandes *as.numeric()*, *as.character()*, *as.factor()*, ...

```
> annot=c('Condition_1','Condition_2','Condition_3','Condition_4','Condition_5')
> annot
[1] "Condition_1" "Condition_2" "Condition_3" "Condition_4" "Condition_5"
> sub('^Condition_', '', annot)
[1] "1" "2" "3" "4" "5"
> sub('^Condition_', '', annot)+10
Error in sub("^Condition_", "", annot) + 10 :
  non-numeric argument to binary operator
> as.numeric(sub('^Condition_', '', annot))+10
[1] 11 12 13 14 15
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Convertir le type de variable (ex. : le nombre « 2 » à considérer comme le caractère « 2 ») : commandes *as.numeric()*, *as.character()*, *as.factor()*, ...

Contrôler le type d'une variable : commandes *is.numeric()*, *is.character()*, *is.factor()*, ...

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Types de variables

Convertir le type de variable (ex. : le nombre « 2 » à considérer comme le caractère « 2 ») : commandes *as.numeric()*, *as.character()*, *as.factor()*, ...

Contrôler le type d'une variable : commandes *is.numeric()*, *is.character()*, *is.factor()*, ...

```
> is.numeric(annot)
[1] FALSE
> is.character(annot)
[1] TRUE
>
```

[Entrer des commandes](#)[Commandes et fonctions](#)[Types de variables](#)[Entrée de données](#)[L'interface RStudio](#)[Structuration des calculs](#)[Tests statistiques usuels](#)[Structures de contrôle](#)[Tracer des graphiques](#)[Fichiers de sortie](#)[Conclusion](#)

Types de variables

Donnée manquante (dans un vecteur ou tableau de nature quelconque : numérique, caractère, booléen, ...) : codée par « NA » (mais sans les guillemets ! elle serait interprétée comme une chaîne de caractères sinon) :

Types de variables

Donnée manquante (dans un vecteur ou tableau de nature quelconque : numérique, caractère, booléen, ...) : codée par « NA » (mais sans les guillemets ! elle serait interprétée comme une chaîne de caractères sinon) :

```
> c(1,5,2,NA,4,6)
[1] 1 5 2 NA 4 6
> is.numeric(c(1,5,2,NA,4,6))
[1] TRUE
> c('Allo ?', 'Bonjour !', NA, 'Au revoir !')
[1] "Allo ?"      "Bonjour !"   NA           "Au revoir !"
> is.character(c('Allo ?', 'Bonjour !', NA, 'Au revoir !'))
[1] TRUE
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Entrée de données

Importer des données dans une session **R** :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).
- ▶ d'un fichier CSV, tabulé par des virgules (par défaut) : *read.csv* (et sa variante *read.csv2*).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).
- ▶ d'un fichier CSV, tabulé par des virgules (par défaut) : *read.csv* (et sa variante *read.csv2*).
- ▶ d'autres types de fichiers, d'usage moins courant (*read.DIF*, *read.socket*, *read.dcf*).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).
- ▶ d'un fichier CSV, tabulé par des virgules (par défaut) : *read.csv* (et sa variante *read.csv2*).
- ▶ d'autres types de fichiers, d'usage moins courant (*read.DIF*, *read.socket*, *read.dcf*).

La tabulation par un caractère pré-défini : une façon de formater des tableaux dans un texte brut.

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).
- ▶ d'un fichier CSV, tabulé par des virgules (par défaut) : *read.csv* (et sa variante *read.csv2*).
- ▶ d'autres types de fichiers, d'usage moins courant (*read.DIF*, *read.socket*, *read.dcf*).

La tabulation par un caractère pré-défini : une façon de formater des tableaux dans un texte brut. Caractère d'échappement : permet d'utiliser le caractère de tabulation à l'intérieur d'une colonne.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Entrée de données

Importer des données dans une session **R** :

- ▶ d'un fichier texte, tabulé par des espaces : *read.table* (et ses variantes *read.delim*, *read.delim2*, *read.fwf*).
- ▶ d'un fichier CSV, tabulé par des virgules (par défaut) : *read.csv* (et sa variante *read.csv2*).
- ▶ d'autres types de fichiers, d'usage moins courant (*read.DIF*, *read.socket*, *read.dcf*).

La tabulation par un caractère pré-défini : une façon de formater des tableaux dans un texte brut. Caractère d'échappement : permet d'utiliser le caractère de tabulation à l'intérieur d'une colonne.

Exercice : sous Excel, exporter un fichier aux formats CSV et TSV. Les ouvrir sous **R** (→ un *data.frame*).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

L'interface RStudio

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface RStudio

Offre un environnement graphique en plus de la console.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

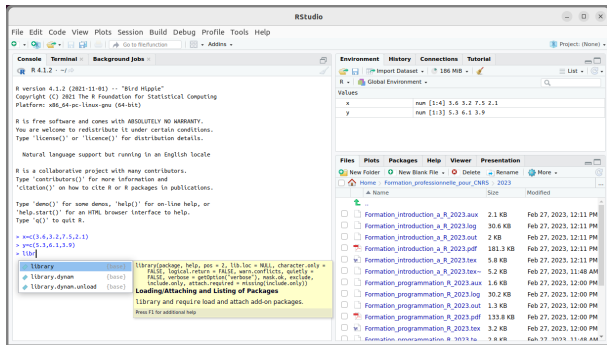
Tracer des
graphiques

Fichiers de sortie

Conclusion

L'interface RStudio

Offre un environnement graphique en plus de la console.



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

L'interface RStudio

Plusieurs panneaux :

- console (pour entrer les commandes) ;

Entrer des commandes

Commandes et fonctions

Type de variables

Entrée de données

L'interface RStudio

Structuration des calculs

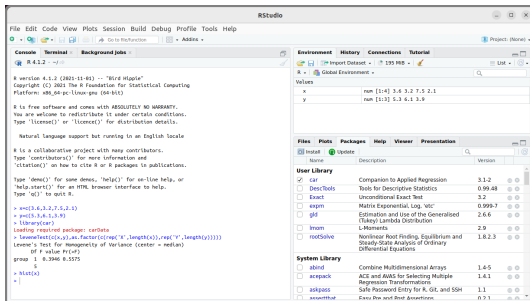
Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

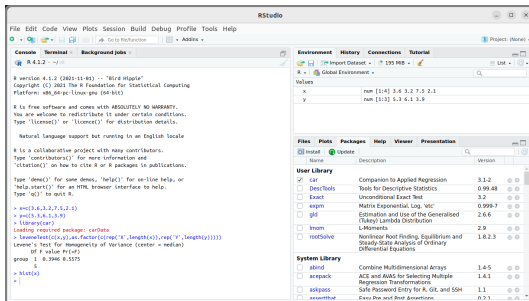
Conclusion



L'interface RStudio

Plusieurs panneaux :

- ▶ console (pour entrer les commandes);
- ▶ historique (pour visualiser les commandes passées) / environnement (pour voir l'état des variables) / ...



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

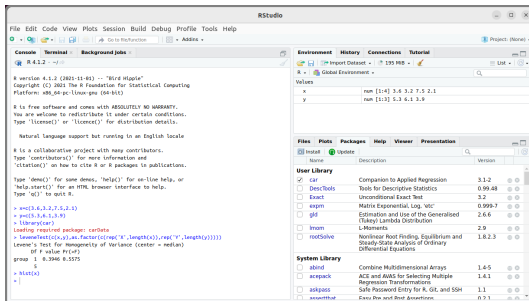
Fichiers de sortie

Conclusion

L'interface RStudio

Plusieurs panneaux :

- ▶ console (pour entrer les commandes);
- ▶ historique (pour visualiser les commandes passées) / environnement (pour voir l'état des variables) / ...
- ▶ arborescence des fichiers / sorties graphiques / *packages* chargés / ...



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ...

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

**Structuration des
calculs**

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ... mais aussi par une condition à tester.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ... mais aussi par une condition à tester.

```
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)  
> x[3]  
[1] 4.5  
> x[x<2]  
[1] 1.0 1.0 0.0 -2.5 0.1  
> x[x<=0]  
[1] 0.0 -2.5  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ... mais aussi par une condition à tester.

La condition peut s'appliquer sur une autre variable.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ... mais aussi par une condition à tester.

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x[3]
[1] 4.5
> x[x<2]
[1] 1.0 1.0 0.0 -2.5 0.1
> x[x<=0]
[1] 0.0 -2.5
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> x[annot=='untreated']
[1] 4.5 2.3 -2.5 0.1
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Dans un vecteur, les crochets « [] » permettent de sélectionner un élément par son indice ... mais aussi par une condition à tester.

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x[3]
[1] 4.5
> x[x<2]
[1] 1.0 1.0 0.0 -2.5 0.1
> x[x<=0]
[1] 0.0 -2.5
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> x[annot=='untreated']
[1] 4.5 2.3 -2.5 0.1
>
```

Danger ! Ne pas confondre « = » (assignation) et « == » (test d'égalité).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Combiner des conditions : « & » pour « et », « | » pour « ou ».

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

**Structuration des
calculs**

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Combiner des conditions : « & » pour « et », « | » pour « ou ».

```
> a=3
> b=1
> if ((a>2) & (b>2)) print("Les deux sont >2")
> if ((a>2) & (b>0)) print("a est >2 et b est >0")
[1] "a est >2 et b est >0"
> if ((a>2) | (b>2)) print("Au moins l'un des deux est >2")
[1] "Au moins l'un des deux est >2"
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Combiner des conditions : « & » pour « et », « | » pour « ou ». Parenthèses pour grouper les conditions par priorité.

```
> a=3
> b=1
> if ((a>2) & (b>2)) print("Les deux sont >2")
> if ((a>2) & (b>0)) print("a est >2 et b est >0")
[1] "a est >2 et b est >0"
> if ((a>2) | (b>2)) print("Au moins l'un des deux est >2")
[1] "Au moins l'un des deux est >2"
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Combiner des conditions : « & » pour « et », « | » pour « ou ». Parenthèses pour grouper les conditions par priorité.

```
> a=3
> b=1
> if ((a>2) & (b>2)) print("Les deux sont >2")
> if ((a>2) & (b>0)) print("a est >2 et b est >0")
[1] "a est >2 et b est >0"
> if ((a>2) | (b>2)) print("Au moins l'un des deux est >2")
[1] "Au moins l'un des deux est >2"
```

```
> a=3
> b=1
> c=10
> if (((a>2) & (b>2)) | (c>8)) print("Soit a et b sont tous les deux >2, soit c est >8")
[1] "Soit a et b sont tous les deux >2, soit c est >8"
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Extraire d'un *data.frame* les lignes qui remplissent une condition : avec la commande *subset()* :

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Extraire d'un *data.frame* les lignes qui remplissent une condition : avec la commande *subset()* :

```
> list_example=list(c('treated','untreated','treated','treated','untreated','untreated'),c(10.3,4.5,9.4,8.3,6.2,7.1))
> names(list_example)=c('Condition','Value')
> df_example=as.data.frame(list_example)
> df_example
  Condition Value
1  treated  10.3
2  untreated   4.5
3  treated   9.4
4  treated   8.3
5  untreated   6.2
6  untreated   7.1
> subset(df_example,df_example$Condition=='treated')
  Condition Value
1  treated  10.3
3  treated   9.4
4  treated   8.3
> subset(df_example,df_example$Value>7)
  Condition Value
1  treated  10.3
3  treated   9.4
4  treated   8.3
6  untreated   7.1
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération à tous les éléments d'un vecteur, ou à certains éléments :

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération à tous les éléments d'un vecteur, ou à certains éléments :

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x[3]
[1] 4.5
> x[x<2]
[1] 1.0 1.0 0.0 -2.5 0.1
> x[x<=0]
[1] 0.0 -2.5
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> x[annot=='untreated']
[1] 4.5 2.3 -2.5 0.1
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> x+2
[1] 3.0 3.0 6.5 8.0 4.3 2.0 -0.5 2.1
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération à tous les éléments d'un vecteur, ou à certains éléments :

```
Type 'q()' to quit R.
> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> x[3]
[1] 4.5
> x[x<2]
[1] 1.0 1.0 0.0 -2.5 0.1
> x[x<=0]
[1] 0.0 -2.5
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> x[annot=='untreated']
[1] 4.5 2.3 -2.5 0.1
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> x+2
[1] 3.0 3.0 6.5 8.0 4.3 2.0 -0.5 2.1
> x[2:4]=x[2:4]+2
> x
[1] 1.0 3.0 6.5 8.0 2.3 0.0 -2.5 0.1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération par lignes, ou par colonnes, dans un tableau : avec la commande *apply()* :

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération par lignes, ou par colonnes, dans un tableau : avec la commande *apply()* :

```
[1] 4.5 2.3 -2.5 0.1
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> x+2
[1] 3.0 3.0 6.5 8.0 4.3 2.0 -0.5 2.1
> x[2:4]=x[2:4]+2
> x
[1] 1.0 3.0 6.5 8.0 2.3 0.0 -2.5 0.1
> y=array(x,dim=c(4,2))
> y
      [,1] [,2]
[1,]  1.0  2.3
[2,]  3.0  0.0
[3,]  6.5 -2.5
[4,]  8.0  0.1
> apply(y,1,sum)
[1] 3.3 3.0 4.0 8.1
> apply(y,2,sum)
[1] 18.5 -0.1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération par lignes, ou par colonnes, dans un tableau : avec la commande *apply()* :

```
[1] 4.5 2.3 -2.5 0.1
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> x+2
[1] 3.0 3.0 6.5 8.0 4.3 2.0 -0.5 2.1
> x[2:4]=x[2:4]+2
> x
[1] 1.0 3.0 6.5 8.0 2.3 0.0 -2.5 0.1
> y=array(x,dim=c(4,2))
> y
      [,1] [,2]
[1,]  1.0  2.3
[2,]  3.0  0.0
[3,]  6.5 -2.5
[4,]  8.0  0.1
> apply(y,1,sum)
[1] 3.3 3.0 4.0 8.1
> apply(y,2,sum)
[1] 18.5 -0.1
>
```

Exercice : Créer un vecteur de valeurs numériques, et créer un nouveau vecteur qui n'en contiendrait que les valeurs > 1 .

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structuration des calculs

Appliquer une même opération par lignes, ou par colonnes, dans un tableau : avec la commande *apply()* :

```
[1] 4.5 2.3 -2.5 0.1
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> x+2
[1] 3.0 3.0 6.5 8.0 4.3 2.0 -0.5 2.1
> x[2:4]=x[2:4]+2
> x
[1] 1.0 3.0 6.5 8.0 2.3 0.0 -2.5 0.1
> y=array(x,dim=c(4,2))
> y
      [,1] [,2]
[1,]  1.0  2.3
[2,]  3.0  0.0
[3,]  6.5 -2.5
[4,]  8.0  0.1
> apply(y,1,sum)
[1] 3.3 3.0 4.0 8.1
> apply(y,2,sum)
[1] 18.5 -0.1
>
```

Exercice : Créer un tableau de valeurs numériques, et calculer la valeur maximale de chaque ligne (avec la commande *max()*), puis la valeur maximale globale.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Le t-test (par défaut : à variances hétérogènes) :

Tests statistiques usuels

Le t-test (par défaut : à variances hétérogènes) :

```
> x=c(3.4,2.1,5.0,3.6,4.1)
> y=c(5.4,4.2,6.0,5.3)
> t.test(x,y)

Welch Two Sample t-test

data: x and y
t = -2.6227, df = 6.9462, p-value = 0.03451
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.0162784 -0.1537216
sample estimates:
mean of x mean of y
 3.640    5.225

> t.test(x,y,var.equal=TRUE)

Two Sample t-test

data: x and y
t = -2.5148, df = 7, p-value = 0.04012
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.07537667 -0.09462333
sample estimates:
mean of x mean of y
 3.640    5.225

>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

One-sided t-test (« less » : la moyenne de la population échantillonnée de la condition x est-elle inférieure à celle de la population échantillonnée de la condition y ?) :

```

3.640    5.225
> t.test(x,y,var.equal=TRUE,alternative='less')

    Two Sample t-test

data:  x and y
t = -2.5148, df = 7, p-value = 0.02006
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -0.390884
sample estimates:
mean of x mean of y
 3.640    5.225

> t.test(x,y,var.equal=TRUE,alternative='greater')

    Two Sample t-test

data:  x and y
t = -2.5148, df = 7, p-value = 0.9799
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -2.779116      Inf
sample estimates:
mean of x mean of y
 3.640    5.225
>

```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

t-test à valeurs appariés (implique que les deux groupes aient le même effectif) :

```
> t.test(x,y,paired=TRUE)
Error in complete.cases(x, y) : not all arguments have the same length
> length(x)
[1] 5
> z=c(4.5,3.2,6.9,5.2,5.6)
> length(z)
[1] 5
> t.test(x,z,paired=TRUE)

      Paired t-test

data:  x and z
t = -9.3736, df = 4, p-value = 0.0007216
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.866526 -1.013474
sample estimates:
mean of the differences
           -1.44
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Un test de normalité : le test de Shapiro-Wilk :

```
> shapiro.test(x)

      Shapiro-Wilk normality test

data:  x
W = 0.97839, p-value = 0.9258

> shapiro.test(y)

      Shapiro-Wilk normality test

data:  y
W = 0.93143, p-value = 0.6028

> shapiro.test(z)

      Shapiro-Wilk normality test

data:  z
W = 0.99408, p-value = 0.9918

>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Extraire la p -value du t-test ou du test de Shapiro-Wilk :

Tests statistiques usuels

Extraire la p -value du t-test ou du test de Shapiro-Wilk :

```
data: x
W = 0.97839, p-value = 0.9258

> shapiro.test(y)

      Shapiro-Wilk normality test

data: y
W = 0.93143, p-value = 0.6028

> shapiro.test(z)

      Shapiro-Wilk normality test

data: z
W = 0.99408, p-value = 0.9918

> t.test(x,y)$p.value
[1] 0.03451026
> shapiro.test(x)$p.value
[1] 0.9258149
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Un test d'homogénéité des variances : le test de Levene
(définir les groupes par un *factor*) :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Un test d'homogénéité des variances : le test de Levene
(définir les groupes par un *factor*) :

```
> leveneTest
Error: object 'leveneTest' not found
> library(car)
Loading required package: carData
> rep('Condition_x',times=length(x))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
> c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y)))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
[6] "Condition_y" "Condition_y" "Condition_y" "Condition_y"
> as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))
[1] Condition_x Condition_x Condition_x Condition_x Condition_x Condition_y
[7] Condition_y Condition_y Condition_y
Levels: Condition_x Condition_y
> leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y)
))))
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.3389 0.5787
      7
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Un test d'homogénéité des variances : le test de Levene
(définir les groupes par un *factor*) :

```
> leveneTest
Error: object 'leveneTest' not found
> library(car)
Loading required package: carData
> rep('Condition_x',times=length(x))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
> c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y)))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
[6] "Condition_y" "Condition_y" "Condition_y" "Condition_y"
> as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))
[1] Condition_x Condition_x Condition_x Condition_x Condition_x Condition_y
[7] Condition_y Condition_y Condition_y
Levels: Condition_x Condition_y
> leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.3389  0.5787
      7
```

(nécessite le chargement du *package* « car »).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Un test d'homogénéité des variances : le test de Levene
(définir les groupes par un *factor*) :

```
> leveneTest
Error: object 'leveneTest' not found
> library(car)
Loading required package: carData
> rep('Condition_x',times=length(x))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
> c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y)))
[1] "Condition_x" "Condition_x" "Condition_x" "Condition_x" "Condition_x"
[6] "Condition_y" "Condition_y" "Condition_y" "Condition_y" "Condition_y"
> as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))
[1] Condition_x Condition_x Condition_x Condition_x Condition_x Condition_y
[7] Condition_y Condition_y Condition_y
Levels: Condition_x Condition_y
> leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.3389  0.5787
      7
>
```

```
> leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))$Pr
[1] 0.5787471      NA
> leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))$Pr[1]
[1] 0.5787471
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Une alternative non-paramétrique au t-test : le test de Wilcoxon (=test de Mann-Whitney) :

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Une alternative non-paramétrique au t-test : le test de Wilcoxon (=test de Mann-Whitney) :

```
> wilcox.test(x,y)

      Wilcoxon rank sum exact test

data:  x and y
W = 1, p-value = 0.03175
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(x,z,paired=TRUE)

      Wilcoxon signed rank test with continuity correction

data:  x and z
V = 0, p-value = 0.05791
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x, z, paired = TRUE) :
  cannot compute exact p-value with ties

>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Une alternative non-paramétrique au t-test : le test de Wilcoxon (=test de Mann-Whitney) :

```
> wilcox.test(x,y)

    Wilcoxon rank sum exact test

data:  x and y
W = 1, p-value = 0.03175
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(x,z,paired=TRUE)

    Wilcoxon signed rank test with continuity correction

data:  x and z
V = 0, p-value = 0.05791
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x, z, paired = TRUE) :
  cannot compute exact p-value with ties

>
```

Tests basés sur les rangs : imprécis en cas d'*ex-aequo* (ici : sur les différences $x - z$).

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Tests de comparaison de distribution catégorielle :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Tests de comparaison de distribution catégorielle :

```
> table=array(c(10,3,12,25,2,15,6,9,13,4),dim=c(5,2))
> table
      [,1] [,2]
[1,]   10   15
[2,]    3    6
[3,]   12    9
[4,]   25   13
[5,]    2    4
> chisq.test(table)

      Pearson's Chi-squared test

data:  table
X-squared = 6.6491, df = 4, p-value = 0.1556

Warning message:
In chisq.test(table) : Chi-squared approximation may be incorrect
> fisher.test(table)

      Fisher's Exact Test for Count Data

data:  table
p-value = 0.1569
alternative hypothesis: two.sided
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

ANOVA unidimensionnelle (« *one-way ANOVA* ») :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

ANOVA unidimensionnelle (« one-way ANOVA ») :

```
> profession=as.factor(c('boulangier','boulangier','boulangier','boulangier','boulangier','boulangier','b
boulangier','professeur','professeur','professeur','professeur','professeur','professeur','professeur
','professeur','professeur','professeur','policier','policier','policier','policier','policier','po
llicier','policier','policier','infirmier','infirmier','infirmier','infirmier','infirmier','infirmie
r','infirmier','infirmier'))
> temps_telerivision=c(4.4,0.8,3.4,5.9,1.3,5.0,7.0,1.5,1.7,0.9,1.9,0.5,0.8,0.2,2.0,2.8,1.7,6.6,6.5,4.
5,5.8,6.8,4.3,4.6,7.1,3.4,4.0,6.5,4.4,4.2,5.6,6.3,1.8)
> summary(aov(temps_telerivision~profession))
              Df Sum Sq Mean Sq F value    Pr(>F)
profession    3   93.17   31.057    14.12 7.39e-06 ***
Residuals    29   63.76    2.199
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Entrer des
commandes

Commandes et
fonctions

Type de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Tests post-hoc après l'ANOVA unidimensionnelle :

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

Tests post-hoc après l'ANOVA unidimensionnelle :

```
> pairwise.t.test(temps_television,profession,p.adjust.method='BH')  
  
Pairwise comparisons using t tests with pooled SD  
  
data: temps_television and profession  
  
      boulangier infirmier policier  
infirmier 0.47648    -          -  
policier  0.03867  0.12304    -  
professeur 0.00290  0.00036 5.2e-06  
  
P value adjustment method: BH  
>
```

Tests statistiques usuels

ANOVA bidimensionnelle (« *two-way ANOVA* ») :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

**Tests statistiques
usuels**

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

ANOVA bidimensionnelle (« *two-way ANOVA* ») :

```
> values=c(6.80,10.10,7.90,6.59,12.70,19.00,23.90,17.80,19.00,20.50,8.81,9.01,9.71,11.20,9.35,7.93,
7.21,12.40,12.60,6.98,29.00,26.90,30.20,29.90,31.10,26.90,29.50,31.20,31.40,32.20,27.50,29.70,28.20
,29.20,27.00,25.90,33.10,30.20,31.60,30.60)
> strain=rep(c(rep('OregonR',5),rep('yw',5),rep('Canton-S',5),rep('Berlin',5)),2)
> age=c(rep(5,20),rep(15,20))
> summary(aov(values~strain+age))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
strain	3	259.1	86.4	9.008	0.000147 ***
age	1	3094.3	3094.3	322.687	< 2e-16 ***
Residuals	35	335.6	9.6		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

ANOVA bidimensionnelle (« *two-way ANOVA* ») avec interaction :

```
> summary(aov(values~strain*age))
      Df Sum Sq Mean Sq F value    Pr(>F)
strain   3  259.1    86.4    18.85 3.17e-07 ***
age       1 3094.3   3094.3   675.25 < 2e-16 ***
strain:age 3   189.0    63.0    13.75 6.25e-06 ***
Residuals 32   146.6     4.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tests statistiques usuels

ANOVA bidimensionnelle (« *two-way ANOVA* ») ne testant que l'interaction :

```
> summary(aov(values~strain:age))
              Df Sum Sq Mean Sq F value    Pr(>F)
strain:age      4   3170    792.6    53.48 1.98e-14 ***
Residuals     35     519     14.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

**Structures de
contrôle**

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Le test *if* : conditionne une commande à un booléen.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

**Structures de
contrôle**

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Le test *if* : conditionne une commande à un booléen.

```
> a=3  
> if (a<2) print("a est plus petit que 2")  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

**Structures de
contrôle**

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Le test *if* : conditionne une commande à un booléen.

```
> a=3  
> if (a<2) print("a est plus petit que 2")  
> if (a>2) print("a est plus grand que 2")  
[1] "a est plus grand que 2"  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

**Structures de
contrôle**

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

```
> x=c(3.4,2.1,5.0,3.6,4.1)
> y=c(5.4,4.2,6.0,5.3)
> p=leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))$Pr[1]
> if (p<0.05)
+ {
+   print("Heterogeneous variances")
+   t.test(x,y,var.equal=FALSE)
+ } else
+ {
+   print("Homogeneous variances")
+   t.test(x,y,var.equal=TRUE)
+ }
[1] "Homogeneous variances"

Two Sample t-test

data: x and y
t = -2.5148, df = 7, p-value = 0.04012
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.07537667 -0.09462333
sample estimates:
mean of x mean of y
 3.640    5.225

>
```


Structures de contrôle

```

> x=c(3.4,2.1,5.0,3.6,4.1)
> y=c(5.4,4.2,6.0,5.3)
> p=leveneTest(c(x,y),as.factor(c(rep('Condition_x',times=length(x)),rep('Condition_y',times=length(y))))))$Pr[1]
> if (p<0.05)
+ {
+   print("Heterogeneous variances")
+   t.test(x,y,var.equal=FALSE)
+ } else
+ {
+   print("Homogeneous variances")
+   t.test(x,y,var.equal=TRUE)
+ }
[1] "Homogeneous variances"

                Two Sample t-test

data:  x and y
t = -2.5148, df = 7, p-value = 0.04012
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.07537667 -0.09462333
sample estimates:
mean of x mean of y
   3.640    5.225
>

```

Les accolades permettent de conditionner plusieurs commandes au *if* (par défaut : seule la première commande est concernée).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Les boucles *for* et *while* : répéter une série d'instructions.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

**Structures de
contrôle**

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Les boucles *for* et *while* : répéter une série d'instructions.

```
> for (i in 1:10)
+ print(paste("now i =",i))
[1] "now i = 1"
[1] "now i = 2"
[1] "now i = 3"
[1] "now i = 4"
[1] "now i = 5"
[1] "now i = 6"
[1] "now i = 7"
[1] "now i = 8"
[1] "now i = 9"
[1] "now i = 10"
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Les boucles *for* et *while* : répéter une série d'instructions.

```
> r=1
> while (r<100)
+ {
+   r=2*r
+   print(paste("now r =",r))
+ }
[1] "now r = 2"
[1] "now r = 4"
[1] "now r = 8"
[1] "now r = 16"
[1] "now r = 32"
[1] "now r = 64"
[1] "now r = 128"
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Structures de contrôle

Les boucles *for* et *while* : répéter une série d'instructions.

```
> r=1
> while (r<100)
+ {
+   r=2*r
+   print(paste("now r =",r))
+ }
[1] "now r = 2"
[1] "now r = 4"
[1] "now r = 8"
[1] "now r = 16"
[1] "now r = 32"
[1] "now r = 64"
[1] "now r = 128"
>
```

Attention ! Pour les opérations à effectuer sur chaque élément d'un vecteur, il est plus efficace d'appliquer l'opération à tout le vecteur, que de faire une boucle *for* pour le parcourir.

► Appliquer une opération à tout un vecteur

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

**Tracer des
graphiques**

Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

```
> x=c(-10:10)
> y=x^2
> plot(x,y)
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

**Tracer des
graphiques**

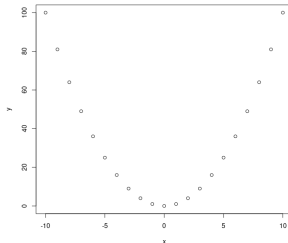
Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

```
> x=c(-10:10)
> y=x^2
> plot(x,y)
>
```



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

```
> y=x^2  
> plot(x,y)  
> plot(x,y,ty='l')  
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

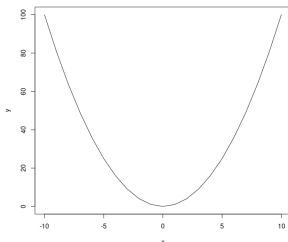
Fichiers de sortie

Conclusion

Tracer des graphiques

La commande *plot* : à partir de deux vecteurs de même taille (les abscisses, et les ordonnées), trace les points et (optionnellement) les relie par des segments de droite.

```
> y=x^2  
> plot(x,y)  
> plot(x,y,ty='l')  
>
```



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

- commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

**Tracer des
graphiques**

Fichiers de sortie

Conclusion

Tracer des graphiques

- commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;

```
> hist(x)
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

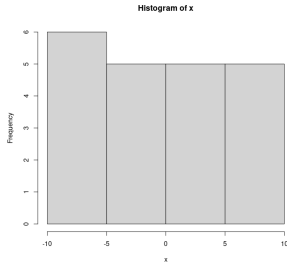
Fichiers de sortie

Conclusion

Tracer des graphiques

- commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;

```
> hist(x)
```



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

- ▶ commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;
- ▶ commande *barplot* : des barres partant de 0 (côte à côte si l'argument est un vecteur ; possibilité de superposer des barres si l'argument est un tableau).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

- ▶ commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;
- ▶ commande *barplot* : des barres partant de 0 (côte à côte si l'argument est un vecteur ; possibilité de superposer des barres si l'argument est un tableau).

```
> barplot(x)
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

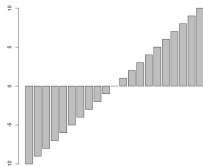
Fichiers de sortie

Conclusion

Tracer des graphiques

- ▶ commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;
- ▶ commande *barplot* : des barres partant de 0 (côte à côte si l'argument est un vecteur ; possibilité de superposer des barres si l'argument est un tableau).

```
> barplot(x)
```



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

- ▶ commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;
- ▶ commande *barplot* : des barres partant de 0 (côte à côte si l'argument est un vecteur ; possibilité de superposer des barres si l'argument est un tableau).

```
> barplot(array(x,dim=c(7,3)))
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

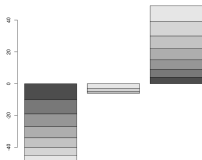
Fichiers de sortie

Conclusion

Tracer des graphiques

- ▶ commande *hist* : trace un histogramme (nombre réglable d'intervalles) ;
- ▶ commande *barplot* : des barres partant de 0 (côte à côte si l'argument est un vecteur ; possibilité de superposer des barres si l'argument est un tableau).

```
> barplot(array(x,dim=c(7,3)))
```



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

Superposer des graphes : avec la commande *par(new=TRUE)*.

Tracer des graphiques

Superposer des graphes : avec la commande *par(new=TRUE)*.

Réglages :

- ▶ gamme de valeurs des axes : avec les options *xlim* et *ylim* ;
- ▶ position et étiquetage des graduations : avec la commande *axis* (après avoir tracé le graphe avec l'option *axes=FALSE* pour éviter les superpositions) ;
- ▶ couleur des courbes ou points : avec l'option *color* ;
- ▶ largeur des traits, taille des marges, ... : cf l'aide de la commande *plot*.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

```
> plot(x,y,xlim=c(-20,20),ylim=c(-20,120),xlab='Abscisse',ylab='Ordonnée',axes=F,ty='b',col='red',main='Exemple')
> axis(1,labels=5*(-4:4),at=5*(-4:4))
> axis(2)
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

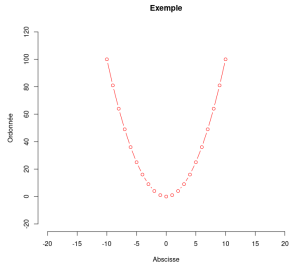
**Tracer des
graphiques**

Fichiers de sortie

Conclusion

Tracer des graphiques

```
> plot(x,y,xlim=c(-20,20),ylim=c(-20,120),xlab='Abscisse',ylab='Ordonnée',axes=F,ty='b',col='red',main='Exemple')  
> axis(1,labels=5*(-4:4),at=5*(-4:4))  
> axis(2)  
>
```



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

Sur un graphe existant : ajout de points (commande *points()*), de texte (*text()*), d'un cadre de légende (*legend()*).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

Sur un graphe existant : ajout de points (commande `points()`), de texte (`text()`), d'un cadre de légende (`legend()`).

```
> plot(x,y,xlim=c(-20,20),ylim=c(-20,120),xlab='Abscisse',ylab='Ordonnée',axes=F,ty='b',col='red',main='Exemple')
> par(new=TRUE)
> plot(x,0.3*y,xlim=c(-20,20),ylim=c(-20,120),xlab='',ylab='',axes=F,ty='b',col='blue',main='')
> axis(1,labels=5*(-4:4),at=5*(-4:4))
> axis(2)
> points(c(-15,-5,5,15),c(-10,30,30,-10),col='green',ty='b')
> arrows(0,25,0,15,length=0.1)
> text(0,25,'Minimum',pos=3)
> legend('topright',c('Condition 1','Condition 2','Condition 3'),col=c('red','blue','green'),pch=1)
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

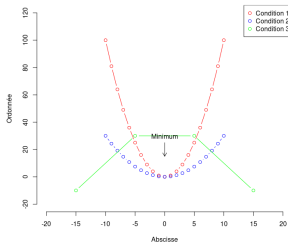
Conclusion

Tracer des graphiques

Sur un graphe existant : ajout de points (commande `points()`), de texte (`text()`), d'un cadre de légende (`legend()`).

```
> plot(x,y,xlim=c(-20,20),ylim=c(-20,120),xlab='Abscisse',ylab='Ordonnée',axes=F,ty='b',col='red',main='Exemple')
> par(new=TRUE)
> plot(x,0.3*y,xlim=c(-20,20),ylim=c(-20,120),xlab='',ylab='',axes=F,ty='b',col='blue',main='')
> axis(1,labels=5*(-4:4),at=5*(-4:4))
> axis(2)
> points(c(-15,-5,5,15),c(-10,30,30,-10),col='green',ty='b')
> arrows(0,25,0,15,length=0.1)
> text(0,25,'Minimum',pos=3)
> legend('topright',c('Condition 1','Condition 2','Condition 3'),col=c('red','blue','green'),pch=1)
>
```

Exemple



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Tracer des graphiques

Panneaux multiples : avec la commande *par(mfrow)*.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

**Tracer des
graphiques**

Fichiers de sortie

Conclusion

Tracer des graphiques

Panneaux multiples : avec la commande `par(mfrow)`.

```
> par(mfrow=c(2,3))
> plot(x,x)
> plot(x,x^2)
> plot(x,x^3)
> plot(x,x^4)
> plot(x,x^5)
> plot(x,x^6)
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

**Tracer des
graphiques**

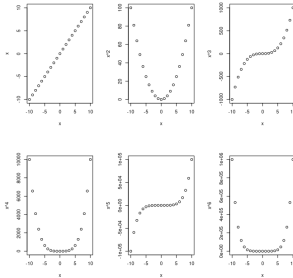
Fichiers de sortie

Conclusion

Tracer des graphiques

Panneaux multiples : avec la commande `par(mfrow)`.

```
> par(mfrow=c(2,3))
> plot(x,x)
> plot(x,x^2)
> plot(x,x^3)
> plot(x,x^4)
> plot(x,x^5)
> plot(x,x^6)
>
```



Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

```
> x=c(1:10)
> q()
Save workspace image? [y/n/c]:
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

La commande *sink()* : affiche le résultat des commandes dans un fichier au lieu de les afficher dans le terminal.

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

La commande *sink()* : affiche le résultat des commandes dans un fichier au lieu de les afficher dans le terminal.

```
> sink('Output.txt')  
> x=c(1:10)  
> print(x)  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

La commande *sink()* : affiche le résultat des commandes dans un fichier au lieu de les afficher dans le terminal.
Mettre fin à cette sortie texte : avec une commande *sink()* sans argument.

```
> sink('Output.txt')  
> x=c(1:10)  
> print(x)  
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

La commande *sink()* : affiche le résultat des commandes dans un fichier au lieu de les afficher dans le terminal.
Mettre fin à cette sortie texte : avec une commande *sink()* sans argument.

```
> sink('Output.txt')
> x=c(1:10)
> print(x)
> sink()
> print(x)
[1] 1 2 3 4 5 6 7 8 9 10
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

À la fin de la session **R**, il est possible d'enregistrer les détails des commandes passées, et leurs résultats (ré-utilisables dans une session future).

La commande *sink()* : affiche le résultat des commandes dans un fichier au lieu de les afficher dans le terminal.
Mettre fin à cette sortie texte : avec une commande *sink()* sans argument.

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Pour une sortie plus facilement utilisable : *write.csv* (ou ses variantes *write.table*, *write.socket*, etc).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Pour une sortie plus facilement utilisable : *write.csv* (ou ses variantes *write.table*, *write.socket*, etc).

```
> x=c(1,1,4.5,6,2.3,0,-2.5,0.1)
> annot=c('treated','treated','untreated','treated','untreated','treated','untreated','untreated')
> list_example=list(x,annot)
> df_example=as.data.frame(list_example)
> names(df_example)=c('Value','Annotation')
> write.csv(df_example,'Output.csv')
>
```

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Pour une sortie plus facilement utilisable : *write.csv* (ou ses variantes *write.table*, *write.socket*, etc).

	A	B	C	D
1		Value	Annotation	
2	1	1,00	treated	
3	2	1,00	treated	
4	3	4,50	untreated	
5	4	6,00	treated	
6	5	2,30	untreated	
7	6	0,00	treated	
8	7	-2,50	untreated	
9	8	0,10	untreated	
10				
11				
12				
13				

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Logiciel R

H. Seitz



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Fichier de sortie graphique : avec les commandes `svg()`, `pdf()`, `postscript()` (et leurs variantes `cairo_pdf()` et `cairo_ps()`) (formats vectoriels), `png()`, `bmp()`, `jpeg()` et `tiff()` (formats *bitmap*).

Entrer des commandes

Commandes et fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des calculs

Tests statistiques usuels

Structures de contrôle

Tracer des graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Fichier de sortie graphique : avec les commandes `svg()`, `pdf()`, `postscript()` (et leurs variantes `cairo_pdf()` et `cairo_ps()`) (formats vectoriels), `png()`, `bmp()`, `jpeg()` et `tiff()` (formats *bitmap*).

Formats vectoriels (description des éléments graphiques) : éditables, pas de problème de zoom, efficaces en espace disque pour des motifs simples (à privilégier!).

[Entrer des commandes](#)[Commandes et fonctions](#)[Types de variables](#)[Entrée de données](#)[L'interface RStudio](#)[Structuration des calculs](#)[Tests statistiques usuels](#)[Structures de contrôle](#)[Tracer des graphiques](#)[Fichiers de sortie](#)[Conclusion](#)

Fichiers de sortie

Fichier de sortie graphique : avec les commandes `svg()`, `pdf()`, `postscript()` (et leurs variantes `cairo_pdf()` et `cairo_ps()`) (formats vectoriels), `png()`, `bmp()`, `jpeg()` et `tiff()` (formats *bitmap*).

Formats vectoriels (description des éléments graphiques) : éditables, pas de problème de zoom, efficaces en espace disque pour des motifs simples (à privilégier!).

Formats *bitmap* : parfois inévitables pour des graphes couverts de nombreux points.

[Entrer des commandes](#)[Commandes et fonctions](#)[Types de variables](#)[Entrée de données](#)[L'interface RStudio](#)[Structuration des calculs](#)[Tests statistiques usuels](#)[Structures de contrôle](#)[Tracer des graphiques](#)[Fichiers de sortie](#)[Conclusion](#)

Fichiers de sortie

Fichier de sortie graphique : avec les commandes `svg()`, `pdf()`, `postscript()` (et leurs variantes `cairo_pdf()` et `cairo_ps()`) (formats vectoriels), `png()`, `bmp()`, `jpeg()` et `tiff()` (formats *bitmap*).

Formats vectoriels (description des éléments graphiques) : éditables, pas de problème de zoom, efficaces en espace disque pour des motifs simples (à privilégier!).

Formats *bitmap* : parfois inévitables pour des graphes couverts de nombreux points.

```
> x
[1] 1.0 1.0 4.5 6.0 2.3 0.0 -2.5 0.1
> pdf('Essai.pdf',width=6,height=6)
> plot(x,x^2,ty='b')
> dev.off()
null device
      1
>
```

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Dans le fichier 'Essai.pdf' :

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

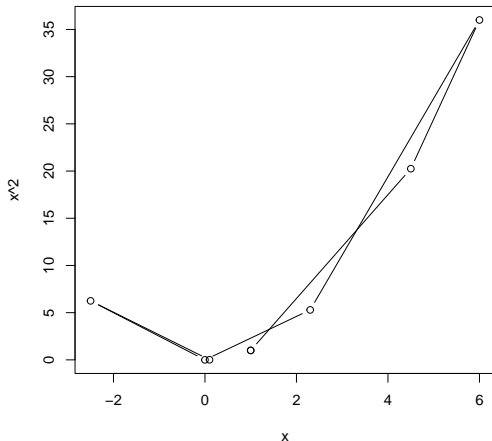
Tracer des
graphiques

Fichiers de sortie

Conclusion

Fichiers de sortie

Dans le fichier 'Essai.pdf' :



Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion

Conclusion

- ▶ Autonomie sur les analyses les plus simples.

Conclusion

- ▶ Autonomie sur les analyses les plus simples.
- ▶ Connaître le vocabulaire et les concepts pour chercher des informations pour les problèmes plus compliqués.

Conclusion

- ▶ Autonomie sur les analyses les plus simples.
- ▶ Connaître le vocabulaire et les concepts pour chercher des informations pour les problèmes plus compliqués.
- ▶ L'archivage des commandes (en plus de celui des données!) permet une reproductibilité intégrale.

Entrer des
commandes

Commandes et
fonctions

Types de variables

Entrée de données

L'interface RStudio

Structuration des
calculs

Tests statistiques
usuels

Structures de
contrôle

Tracer des
graphiques

Fichiers de sortie

Conclusion