



## Initiation au logiciel R

Formation ITA-FAFSEA – 11 février 2014

**François Brun (ACTA)**

**François Piraux (Arvalis – Institut du végétal)**

Florent Duyme (Arvalis – Institut du végétal)

Contacts: [francois.brun@acta.asso.fr](mailto:francois.brun@acta.asso.fr)

[f.piraux@arvalisinstitutduvegetal.fr](mailto:f.piraux@arvalisinstitutduvegetal.fr)

[f.duyme@arvalisinstitutduvegetal.fr](mailto:f.duyme@arvalisinstitutduvegetal.fr)

### Objectifs de cette initiation

- **Objectifs** : Se familiariser avec le langage et l'outil R. Manipuler des données, les représenter graphiquement et réaliser des analyses statistiques de base.
- **pas de pré requis.**
- **Apprendre et pratiquer:**
  - Suivre la présentation
  - Et pratiquer sur son ordinateur :
    - Manipuler/ saisir les commandes
    - Utiliser le script déjà enregistré



## Tour de table

Prénoms	Noms	mails
Mathieu	COUFFIGNAL	m.couffignal@arvalisinstitutduvegetal.fr
Jean-Luc	GOUDOUNECHE	m.goudouneche@arvalisinstitutduvegetal.fr
Bruno	FONTAINE	b.fontaine@arvalisinstitutduvegetal.fr
Sonia	GEOFFROY	s.geoffroy@arvalisinstitutduvegetal.fr
Perrine	MORIS	p.moris@arvalisinstitutduvegetal.fr
Bastien	RAUX	b.raux@arvalisinstitutduvegetal.fr
Béatrice	ORLANDO	b.orlando@arvalisinstitutduvegetal.fr
Séverine	TRUPIN	s.trupin@arvalisinstitutduvegetal.fr
Jean-Michel	ASTRUC	Jean-michel.astruc@idele.fr
Gilles	THOMAS	Gilles.thomas@idele.fr
Emmanuel	MORIN	Emmanuel.morin@idele.fr
Emma	DIELENSEGER	Emma.dielenseger@idele.fr
Nathalie	DUPONT	Nathalie.dupont@ifpc.eu
Elmina	MOTTIN	elmina.mottin@acta.asso.fr

3

## Programme

- **Cours1. Introduction - 0.5h**
- **Cours2. Les types de données et les objets – 1h**
- **Cours3. Les autres structures de données (matrice, data.frame, list, ...) - 1h**
- **Cours4. Lire des données externes - 0.5h**
- **Cours5. Graphiques sous R et paramètres graphiques - 1h**
- **Cours6. Statistiques de base avec R - 1h**
- **Cours7. Sauvegardes de script, graphiques, données et résultats d'analyse - 0.5h**
- **Cours8. Programmation avec R (fonction, loop, conditions,...) - 0.5h**
- **Cours9. Compléments : installer un package, aide, données manquantes - 0.5h**
- **Conclusion et suites à donner en terme de modules de formation avancée – qq mn**

4

## **Cours1. Introduction - 0.5h**

Utilisation de R comme une calculatrice.

Création d'un premier script.

5

## **Qu'est ce que R ?**

- environnement/système d'analyse statistique
  - un langage de programmation ET un logiciel
  - distribué gratuitement « GNU Public Licence »
  - R Development Core Team
  - « dialecte » du langage S (logiciel S-PLUS)
- un langage simple de très haut niveau
  - analyses statistiques
  - graphiques
  - calcul matriciel
  - interprété (un peu lent...)

6

## Où TROUVE-T-ON R ?

<http://www.r-project.org/>

- exe d'installation du logiciel (version actuelle : 3.0.2)
- zip des compléments (=package) au logiciel
- pdf de la documentation
- FAQ
- liens

7

## Support de cours



- Un fichier zip : R\_ITA.zip

À décompresser

à copier (si possible) dans **C:/R\_ITA**

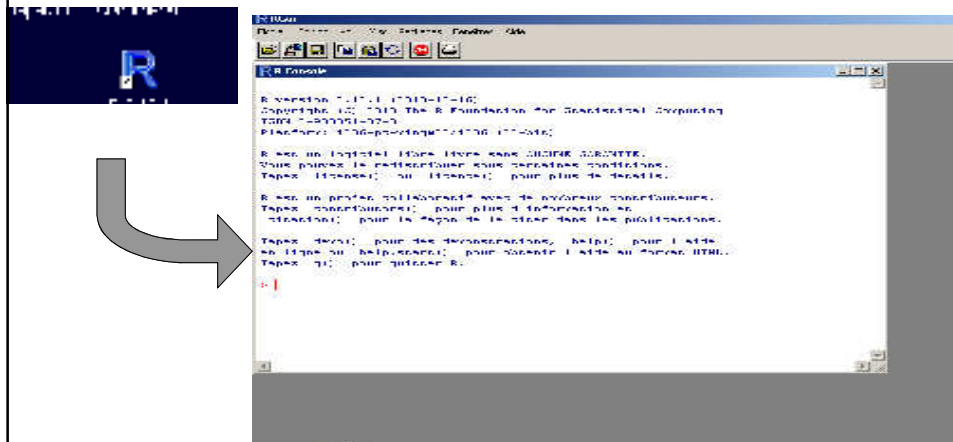
ou D:/R\_ITA

ou ailleurs

=> Il faudra dans certains script adapter  
l'instruction **setwd("C:/R\_ITA/")**

8

## Ouvrir R



- On tape chaque ligne de code (bien connaître la syntaxe !)
- On tape les lignes de code dans un éditeur (R\_fichier\_nouveau script, bloc note, pspad, crimson, tinnR, RSTUDIO ...) puis on les copie/colle dans R

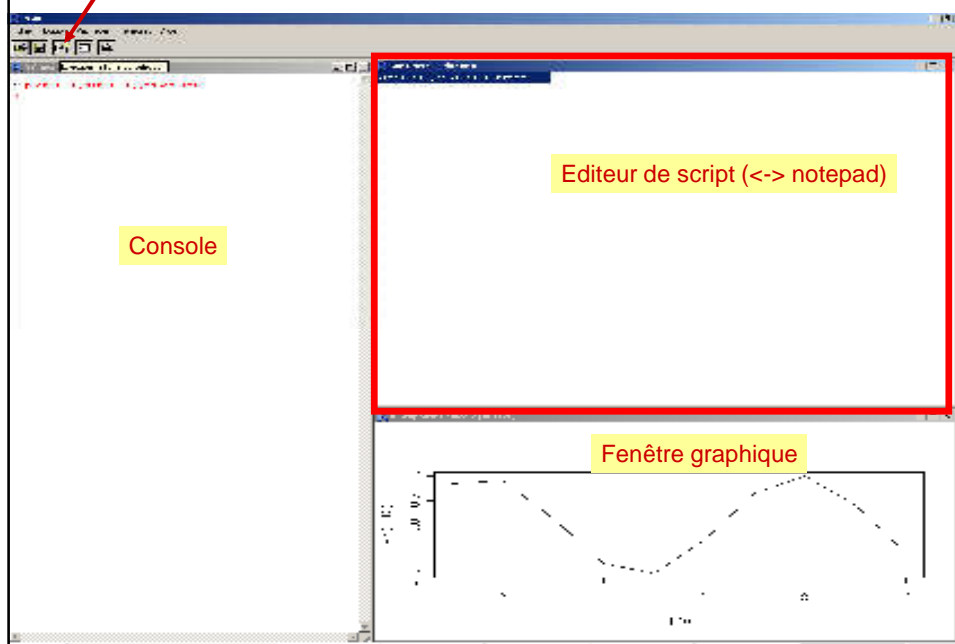
9

## L'interface utilisateur (1)

- Menu et icônes pour les fichiers de gestion, des fenêtres
- Console R: tapez et exécutez votre code, l'affichage des résultats
- Éditeur de texte: utile pour créer des scripts + possibilité d'utiliser un éditeur extérieur
- graphisme

10

## L'interface utilisateur (2)



## Code couleur pour le cours



- En rouge : ce qui est à taper dans la console (après le « > »)
- En bleu : le résultat affiché

Ex :

1+5

[1] 6

## Accéder à l'aide



```
?seq      # ouvre une fenêtre d'aide
help(seq)
# alternative forms
# arguments
# value
# examples
```

13

## Utilisation comme un calculateur



<code>7+12</code>	<code>sqrt(15.7)</code>
<code>[1] 19</code>	<code>[1] 3.962323</code>
<code>6-8</code>	<code>exp(4.6)</code>
<code>[1] -2</code>	<code>[1] 99.48432</code>
<code>3*5</code>	<code>log(10)</code>
<code>[1] 15</code>	<code>[1] 2.302585</code>
<code>4/2</code>	
<code>[1] 2</code>	
<code>2^6</code>	
<code>[1] 64</code>	

14

## L'opérateur <- ou =



- langage "orienté-objet"
  - variables, les données, les matrices, les fonctions, les résultats, etc. : **stockés** dans la mémoire vive de l'ordinateur sous forme d'« **objets** » qui ont un **nom**

```
x<- 4 # ou x=4 # ou 4->x
```

```
X
```

```
[1] 4
```

```
X
```

# attention minuscule/majuscule !

Erreur : objet 'X' introuvable

```
y<- 7
```

```
x+y
```

```
[1] 11
```

-> est équivalent à <- mais pas à =

15

## Afficher les résultats



- Afficher la variable à l'écran

```
print(pi) # ou pi
```

```
[1] 3.141593
```

```
options(digits=2)
```

```
print(pi)
```

```
[1] 3.1
```

16



## Les objects sous R



`objects()` ou `ls()`

`[1] "x" "y"`

`rm(x)` # on supprime x

17

## Mon premier script



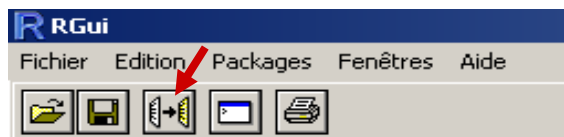
- Dans l'éditeur, écrire les 3 lignes:

`# mon premier script`

`x = 1:10`

`plot(x,x^2)`

- Sélectionner l'ensemble, et cliquer sur l'icône « exécuter la ligne ou la sélection »



- Puis, « file > save to file »
- Puis travail avec TinnR / Rstudio

18

## Cours2. Les types de données et les objets– 1h

Manipulation de vecteurs simples et opération sur ces vecteurs (assignement, sélection).

19

### Types de données

numeric, character, factor, ordered, logical



"a", "b", ...

1,2,3, ...

"T"

"a" < "b" < "c" < ...

F

TRUE

2012

20

## Principaux objets R

- Vecteur
- Matrice : matrix
- Liste d'objets : list
- Tableau de données : data.frame
- Constante

21

## Vecteur numérique



- une collection de valeurs
- valeur manquante notée NA (« not available »)
- mis en place avec c() ou seq():

```
c(10,6,5.7,1)
```

```
x<- c(10,6,5.7,1)
```

```
x
```

```
x= c(10,6,5.7,1)
```

```
x
```

```
y<- 1:10
```

```
y<- seq(1,10, by=1)
```

```
y
```

22

## Vecteur booléen



- permet la manipulation de quantités logiques
- mis en place en utilisant «les conditions»
- Eléments: TRUE (T), FALSE (F) ou NA

```
y<- 2
x<- 2
```

```
w<- x>5
w
[1] FALSE
```

```
w<- T
w2<- c(T,F)
```

```
w1<- (y==3) | (x<=5)
w1
[1] TRUE
```

```
w2<- y>=1
w2
[1] TRUE
```

23

## Vecteur de caractères



- Vecteur de textes
- Chaque élément est entré à l'aide `` ou ‘

```
x<- c("I","like","Montpellier","very","much")
x
[1] "I"      "like"    "Montpellier"  "very"      "much"
```

24

## Manipulation des Vecteurs



- Indices
 

```
y<- c(1,2,5,10,100,200,500)
y[2]
y[2:6]
y[-1]
```
- Modification
 

```
y[2]<- 100
y
y[y>=100]<- 2
y
```
- Arithmétique
 

```
z<- 2*y
y2<- y+z
y2
```

25

## Opérations sur les vecteurs



```
x= c(1, 5, 8, -2, 7)
x
[1] 1 5 8 -2 7
sqrt(x)
log(x)
abs(x)
exp(x)
sin(x)

length(x)
[1] 5
# compter le nombre d'éléments ≥5 dans x
length(x[x>=5]) # ou bien : sum(x>=5)
[1] 3
```

26

## Opérations statistiques

```
mean(x)
sum(x)
var(x)
sd(x)
median(x)
range(x)
min(x)
max(x)
quantile(x, probs=seq(0,1, by=0.1))
```

dénombrement

```
table(x)
```

27

## Trier



```
x<- c(8, 1, -2, 7, 5)
```

```
sort(x)
```

```
[1] -2  1  5  7  8
```

```
sort(x, decreasing=TRUE)
```

28

### Attention : R fait en sorte que cela marche !!

- Ca marche (mais cela pourrait être une erreur...)

```
vector4<- c(2,7,3,4)
vector2<- c(3,6)
vector4+vector2
[1] 5 13 6 10
```

- Ca marche aussi... mais avec un message d'alerte...

```
vector5<- c(2,7,3,4,8)
vector5+vector2
[1] 5 13 6 10 11
Warning message:
In vector5 + vector2 :
  longer object length is not a multiple of
  shorter object length
```

29

## Cours3. Les autres structures de données (matrice, data.frame, list, ...) - 1h

Création et manipulation des structures.

30

## Matrice



- un tableau à deux dimensions
- un seul type de données
- en utilisant matrix()

```
M1 <- matrix(0, nrow=2, ncol=3)
```

```
M1
```

```
      [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
```

31

## Table de données : data.frame



- ressemble à une matrice, mais ses colonnes peuvent être de différents types
- colonnes ont un nom
- commodes pour décrire données expérimentales

```
City<- c("Lille", "Montpellier", "Paris")
```

```
Rank<- c(13,9,1)
```

```
Weather<- c("Rainy", "Sunny", NA)
```

```
TAB<- data.frame(City, Rank, Weather)
```

```
TAB
```

```
      City Rank Weather
1    Lille   13   Rainy
2 Montpellier    9   Sunny
3     Paris    1    <NA>
```

```
TAB$Rank # ou TAB[,2] ou TAB[, "Rank"]
```

```
[1] 13  9  1
```

32



## data.frame



```
TAB[TAB$City=="Paris", ]
```

```
      City Rank Weather
3 Paris    1    <NA>
```

```
TAB[2,]
```

```
      City Rank Weather
2 Montpellier 9   Sunny
```

33

## data.frame



```
str(TAB)
```

```
ncol(TAB) [1] 3
```

```
nrow(TAB) [1] 3
```

```
summary(TAB)
```

City	Rank	Weather
Lille :1	Min. : 1.000	Rainy:1
Montpellier:1	1st Qu.: 5.000	Sunny:1
Paris :1	Median : 9.000	NA's :1
	Mean : 7.667	
	3rd Qu.:11.000	
	Max. :13.000	

```
head(TAB,2)
```

	City	Rank	Weather
1	Lille	13	Rainy
2	Montpellier	9	Sunny

34

## data.frame : tri de données



```
TAB[order(TAB$Rank), ]
```

	City	Rank	Weather
3	Paris	1	<NA>
2	Montpellier	9	Sunny
1	Lille	13	Rainy

```
TAB[order(TAB$Rank, TAB$City), ]
```

35

## Liste



- un ensemble d'éléments de types différents ou non
- pratique pour stocker les résultats de calculs
- définie en utilisant list() et les éléments sont extraits en utilisant [ ]

```
MyList<-list(FALSE,7,M1, TAB)
```

```
MyList
```

```
[[1]]  
[1] FALSE
```

```
[[2]]  
[1] 7
```

```
[[3]]  
      [,1] [,2] [,3]  
[1,]    0    0    0  
[2,]    0    0    0
```

```
[[4]]  
      City Rank Weather  
1      Lille   13  Rainy  
2 Montpellier    9  Sunny  
3        Paris    1   <NA>
```

```
MyList[[3]]  
      [,1] [,2] [,3]  
[1,]    0    0    0  
[2,]    0    0    0
```

36

## **Cours4. Lire des données externes - 0.5h**

Lecture de quelques fichiers dans des formats divers.

37

### **Lire des données externes**

- R lit des fichiers textes, mais il y a d'autres possibilités...

38

## Gestion de vos données / propositions

- En général, fichier XLS
  - **Solution 1. enregistrer les onglets de données brutes sous un fichier texte (TXT, CSV, ...)**
  - **Solution 2. lecture en utilisant une librairie le permettant**
  - (Solution 3. lecture par copier-coller)
- Si multiples fichiers (saisie ou enregistrement)
  - automatiser le travail d'agrégation
- Si utilisation d'un gestionnaire de BD
  - Requête directe à partir de R en sql.

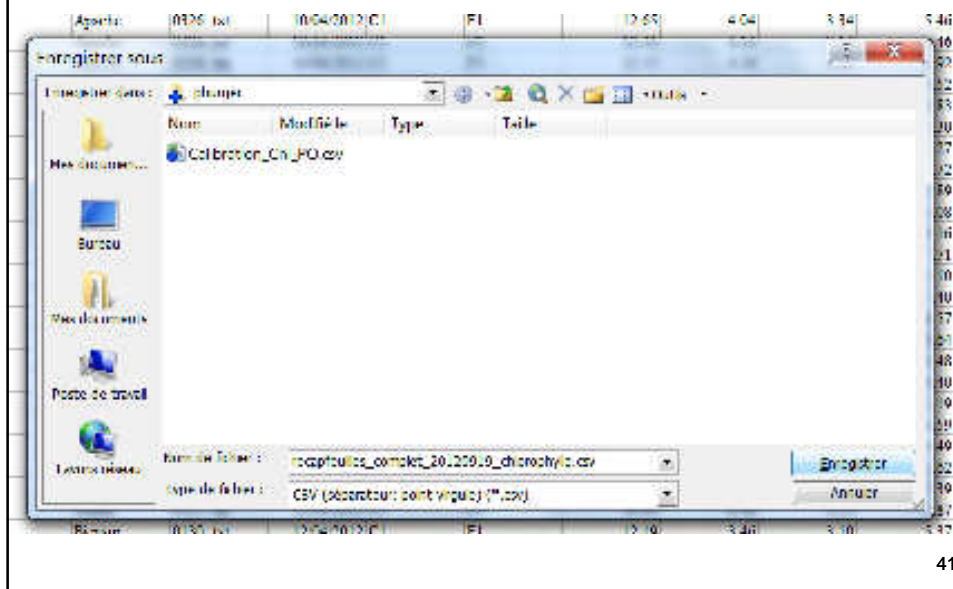
39

## Exemple. Ouvrir un fichier XLS... option (1) → CSV

The screenshot shows an Excel spreadsheet with a table containing data. The table has columns for various identifiers and numerical values. A red box highlights a specific row and column area, likely indicating the data being processed or exported to CSV.

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

## Enregistrer sous... pour chaque onglet



41

## Les nouveaux fichiers CSV

Calibration_Ch_PO.csv	04/10/2012 09:14	Fichier de données SPU	7801 Ko
recapitulatif_complet_20120519.csv	04/10/2012 09:14	Fichier de données SPU	208 Ko
recapitulatif_complet_20120519_pontbeuse.csv	04/10/2012 09:14	Fichier de données SPU	75 Ko
recapitulatif_complet_20120519_chirochyle.csv	04/10/2012 09:14	Fichier de données SPU	18 Ko
recapitulatif_complet_20120519_date1.csv	04/10/2012 09:14	Fichier de données SPU	2 Ko
recapitulatif_complet_20120519_date2.csv	04/10/2012 09:15	Fichier de données SPU	2 Ko

42

Puis, sous R :



- **setwd("C:/R\_ITA/")** Ex : 01\_lecturefichier.r
  - Set working directory
  - On peut adapter le chemin, en prenant le chemin depuis l'explorateur de fichier.
  - Attention au sens des / et à bien mettre " "
- **read.table** (ou read.csv2 ou read.delim)
 

```
statenz <- read.table("data/stat_enzyme.csv", sep=";",  
dec=".", skip=3, header=TRUE, stringsAsFactors=TRUE)
```

**sep** : séparateur colonne  
**dec** : séparateur decimal  
**skip** : on saute n ligne  
**header**: la première ligne est prise pour les noms de colonnes  
**stringsAsFactors** : considérer les chaînes de caractères comme facteurs<sub>43</sub>

option 2 :



librairie pour lire directement du XLS/XLSX

Ex : 02\_lecturefichier.xls2010.r

- **library(xlsx)**
- **setwd("C:/R\_ITA/")**
- **read.xlsx**

```
statenz_1= read.xlsx("data/stat_enzyme.xlsx", sheetName="stat_enzyme",  
startRow=4, header=TRUE, stringsAsFactors = FALSE )
```

**sheetName** : nom de l'onglet  
**startRow** : on commence à la ligne n  
**header**: la première ligne est prise pour les noms de colonnes  
**stringsAsFactors** : considérer les chaînes de caractères comme facteurs

44

### option 3 : par copier/coller

#### Dans Excel

- sélection
- copier

	A	B	C	D	E
	variete	jireca	annee	talia	projet
1	ADWALIS	2013	2013	2013	2013
2	ADWALIS	2013	2013	2013	2013
3	ADWALIS	2013	2013	2013	2013
4	ADWALIS	2013	2013	2013	2013
5	ADWALIS	2013	2013	2013	2013
6	ADWALIS	2013	2013	2013	2013
7	ADWALIS	2013	2013	2013	2013
8	ADWALIS	2013	2013	2013	2013
9	ADWALIS	2013	2013	2013	2013
10	ADWALIS	2013	2013	2013	2013
11	ADWALIS	2013	2013	2013	2013
12	ADWALIS	2013	2013	2013	2013

#### Dans R

```
my_data <- read.delim("clipboard")
```

Attention : intitulés en 1ère ligne

45

### Exercice de sélection

- Sur la table statenz :
  - Sélectionner les lignes correspondant à la variété v2
  - Sélectionner les lignes pour lesquelles les valeurs de PO sont supérieures à la moyenne des PO.

46

## Cours5. Graphiques sous R et paramètres graphiques - 1h

Faire "les graphs classiques" avec R.

47

### graphisme

- De nombreux types de graphiques possibles
- Fonctions de base: `plot`, `lines`, `points`, `hist`, `barplot`
- Fonctions de personnalisation :

`par(mfrow=c(.,.))`

- Des packages avec fonctions avancées

48

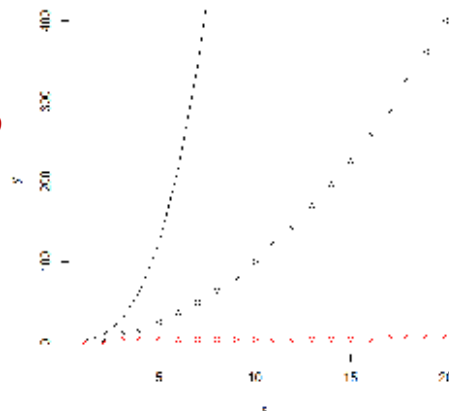


## Nuage de points



```
x<- 1:20
y<- x^2
# crée le graphique avec la première série
plot(x, y)
```

```
# ajoute une série
points(x, sqrt(x), col="red")
lines(x, x^3)
```



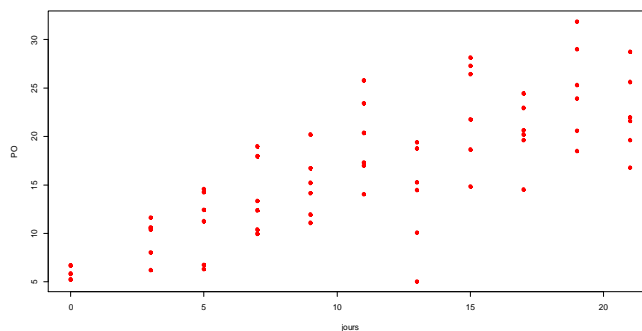
## Nuage de points



Ex : 03\_graphiques.r

PO en fonction de jours, pour v1

```
df1 <- statenz[statenz$cultivar=="v1", c("PO", "jours")]
plot(PO~jours, df1, pch=19, col="red")
```



50

## Autres graphes



# Barplot. Chaque valeur représentée par une barre

```
x <- c(1,5,8,-2,7)
barplot(x)
```



51

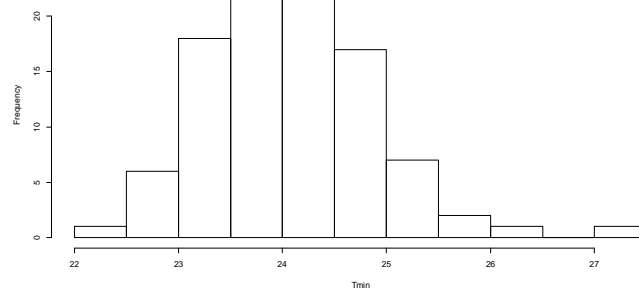
## Autres graphes



# Histogramme. Chaque effectif représenté par une barre

```
weather<- read.delim("data/meteo.dat",dec=',')
hist(weather$tmin, xlab="Tmin", main=" ")
```

Ex : 03\_graphiques.r

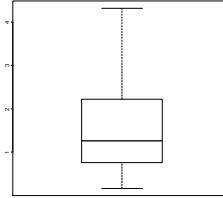


52

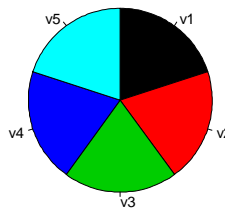
## Autres graphes



```
boxplot(statenz$GST)
```



```
pie(table(statenz$cultivar), col=1:5, clockwise=TRUE, cex=2)
```



53

## Personnalisation

- Personnaliser les graphiques avec **par()** ou **dans les fonctions** graphiques elles-mêmes... Exemple :
  - plusieurs graph sur une page **mfrow = c(2, 2)**
  - les marges : **mar=c(bottom, left, top, right)**
  - taille des caractères et symboles: **cex=1**
  - échelle log : **xlog**
  - ....
  - Voir **help(par)**.

54



```
par(mfrow=c(2,2))
```

Ex : 03\_graphiques.r

```
x<- seq(0,2,by=0.1)*pi
```

```
y<- sin(x)
```

```
z<- cos(x)
```

```
plot(x,y)
```

```
plot(x,y,type="l")
```

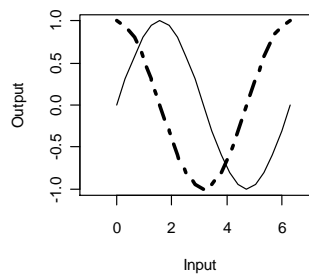
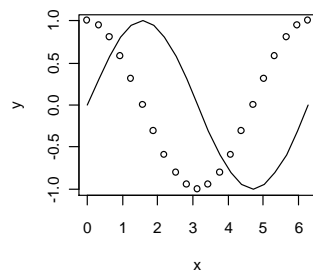
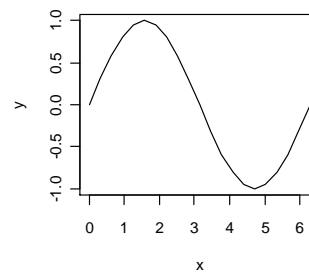
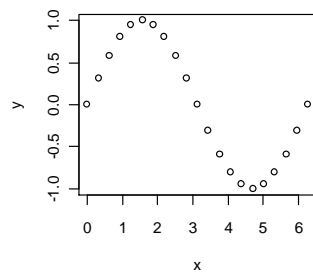
```
plot(x,y,type="l")
```

```
points(x,z)
```

```
plot(x,y,type="l",xlab="Input", ylab="Output",  
xlim=c(-1,7))
```

```
lines(x,z,lwd=3,lty=4)
```

55



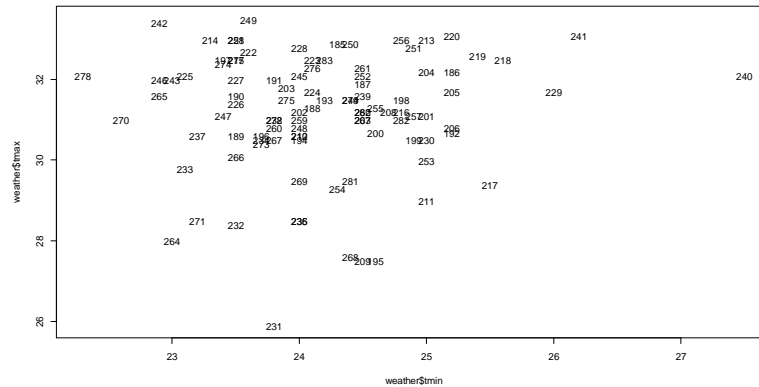
56

## Nuage de points avec nom des individus

Ex : 03\_graphiques.r



```
plot(weather$tmin, weather$tmax, type="n")
text(weather$tmin, weather$tmax, labels=weather$day)
```



57

## Cours6. Statistique de base avec R - 1h

comparaison de moyennes, test du khi2,  
corrélation, régression linéaire, analyse de  
la variance

58

## TESTS STATISTIQUES USUELS

Ex : 04\_stat.r



Comparer 2 groupes indépendants: test de student

```
t.test(LOX~inocule, statenz, var.equal=TRUE)
```

Two Sample t-test

```
data: LOX by inocule
t = -1.0801, df = 328, p-value = 0.2809
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09199371  0.02678158
sample estimates:
mean in group i mean in group ni
    0.6109091      0.6435152
```

59

et comparer les variances ?



```
var.test(LOX~inocule, statenz)
```

Ex : 04\_stat.r

Test du coefficient de corrélation

```
cor(statenz$PO, statenz$jours)
cor.test(~PO+jours,statenz)
```

Pearson's product-moment correlation

```
data: PO and jours
t = 26.1383, df = 328, p-value < 2.2e-16
alternative hypothesis: true correlation is not
equal to 0
95 percent confidence interval:
 0.7835407 0.8541358
sample estimates:
      cor
0.8219707
```

60

## Tableau croisé



```
with(statenz, tapply(GST,cultivar, mean))
```

Ex : 04\_stat.r

```
      v1      v2      v3      v4      v5
1.444697 1.759545 1.520152 1.748182 1.083788
```

```
with(statenz, tapply(GST,list(cultivar, inocule), mean))
```

```
      i      ni
v1 1.551515 1.337879
v2 1.827273 1.691818
v3 1.566667 1.473636
v4 1.828788 1.667576
v5 1.058485 1.109091
```

61

## Croiser 2 variables qualitatives: test du Khi2



Ex : 04\_stat.r

```
temp<- data.frame(
  V1=sample(letters[1:5],300, replace=T),
  V2=sample(LETTERS[1:5],300, replace=T))
table(temp$V1, temp$V2)
chisq.test(table(temp))
```

```
  A  B  C  D  E
a 10 16 13 13  3
b 14  7 11 10 13
c 14 10 15 14 10
d 11 10 10 16  9
e 15 16 15 11 14
```



sample

Pearson's Chi-squared test

```
data: table(temp)
X-squared = 15.3742, df = 16, p-value = 0.4974
```

62

## Test de normalité



Ex : 04\_stat.r

```
W <- statenz$GST
shapiro.test(W)
```

Shapiro-Wilk normality test

```
data: W
W = 0.9367, p-value = 1.202e-10
```

63

## Modèle de régression linéaire simple



Ex : 04\_stat.r

```
y<- weather$tmax
x<- weather$radiation

Fit<- lm(y~x)
print(Fit)
summary(Fit)
COEF <- Fit$coefficients

plot(x,y)
abline(a=COEF[1], b=COEF[2],lty=2)
```

64



## Ex : 04\_stat.r

```
> print(Fit)
Call:
lm(formula = weather$tmax ~ weather$radiation)

Coefficients:
(Intercept)  weather$radiation
    27.3418      0.2223

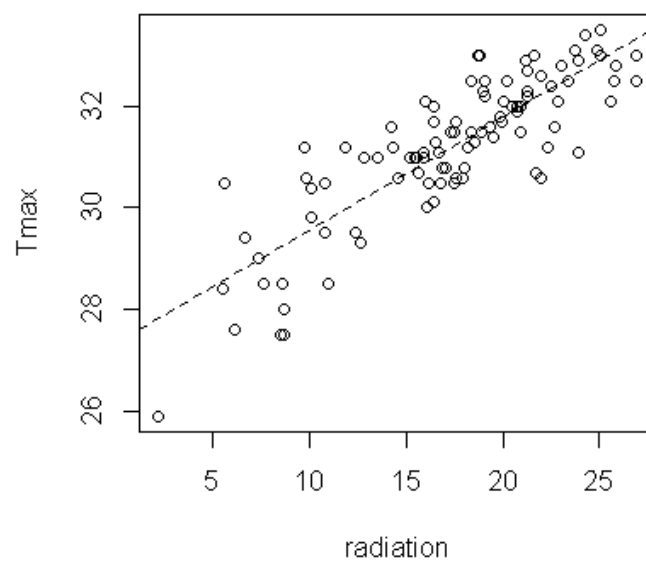
> summary(Fit)
Call:
lm(formula = weather$tmax ~ weather$radiation)

Residuals:
    Min       1Q   Median       3Q      Max
-1.93010 -0.54519  0.05233  0.58254  1.90948

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    27.3418    0.2635  103.78  <2e-16 ***
weather$radiation 0.2223    0.0145   15.33  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7962 on 97 degrees of freedom
Multiple R-squared:  0.7079,    Adjusted R-squared:  0.7049
F-statistic: 235 on 1 and 97 DF, p-value: < 2.2e-16
```

65



66

## ANALYSE D'UN DISPOSITIF équilibré

v1, inoculé => PO



Ex : 04\_stat.r

```
toto <- statenz[statenz$cultivar=="v1" &
  statenz$inocule=="i",]
toto$series <- as.factor(toto$series)
toto$jours <- as.factor(toto$jours)
res_lm <- lm(PO~jours + series, toto)
anova(res_lm)
summary(res_lm)$sigma      # ETR
```

Analysis of Variance Table

Response: PO

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
jours	10	1220.00	122.000	12.1800	1.803e-06	***
series	2	74.62	37.312	3.7251	0.04215	*
Residuals	20	200.33	10.016			

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

67

## Autres modèles

- Régression multiple: lm()
- ANCOVA: lm()
- Generalized linear model: glm()
- Mixed-effect model: lme(), lmer()
- Non linear regression: nls()

68

## Données manquantes



Ex : 07\_manquant.r

quanti 

```
x<- statenz$GST
x[c(10,20,30)]<- NA
sum(is.na(x))
x[!is.na(x)]
x[is.na(x)]<- mean(x, na.rm=TRUE)
```

quali 

```
x<- as.character(statenz$cultivar)
x[c(10,20,30)]<- NA
sum(is.na(x))
x[is.na(x)]<- "inconnu"
```

69

## Cours7. Sauvegardes de scripts, graphiques, données et résultats d'analyse - 0.5h

70

## SAUVEGARDE d'un script

Fichier, enregistrer\_sous ...

71

## SAUVEGARDE d'un graphique

copier en mode vectoriel, coller dans word, powerpoint, ...

ou :

```
png("nom_fichier.png")
  barplot(1:5)
dev.off()
```

En spécifiant une dimension et une résolution (avec png, mais aussi avec tiff ou jpeg)

```
png("nom_fichier.tiff", width = 8, height = 8,
units = "cm", pointsize = 12, res = 300)
  barplot(1:5)
dev.off()
```

Ou, copie fenêtre graphique => fichier

```
dev.print(device =png,file="nom_fichier.png", width = 8, height = 8,
units="cm",res=150)
```

72

## Exportation de données

- Copier/coller depuis la console vers le bloc note, excel, ...
- Un fichier peut être créé à partir de R avec `write.table`; le chemin doit être spécifié

```
write.table(iris, file="iris.txt", row.names=F, sep="\t")
```

73

## SAUVEGARDE de résultats

Copier/coller depuis la console vers word, ...



Ex : 05\_sauvegardes.r

```
sink("synthese.txt")
  anova(res_lm)
  summary(weather)
sink()
```

74

## SAUVEGARDE de tout (ou presque !)

De tout le travail : (=environnement)

Par le menu : fichier, enregistrer l'environnement

```
save.image("nom_fichier.RData")
```

pour charger les précédents résultats

Par le menu : fichier, charger l'environnement

```
load("nom_fichier.RData")
```

75

## Cours8. Programmation avec R (function, loop, conditions,...) - 0.5h

76

## boucles



- pour répéter les instructions
- différentes approches: for, while, repeat, apply

```
for (i in 1:10) { print(i^2) }
```

**Autre exemple**  
**Ex : 06\_prog.r**

77

## Performance : éviter les boucles

- Souvent on peut passer en écriture matricielle ou utiliser des **apply** au lieu des boucles...

```
res = sapply(1:10, function(i) {i^2} )
```

```
res
```

```
[1] 1  4  9 16 25 36 49 64 81 100
```

78

## exécutions conditionnelles

- mise en œuvre des instructions sous certaines conditions
- Syntaxe: **if** (condition) expression\_1 **else** expression\_2
- peut être spécifié en utilisant vecteur logique

```
for (i in 1:10) {
  if (i>=5) {print(i^2) }
  else {print("valeur < 5")}
}
```

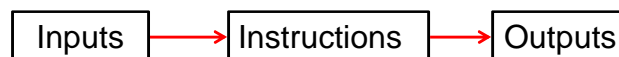
Autre exemple

Ex : 06\_prog.r

79

## fonction (création)

- un utilisateur peut créer des fonctions R pour mettre en œuvre des tâches spécifiques
- Une fonction se caractérise par ses entrées, sorties, et des instructions pour calculer les sorties à partir des entrées



```
functionName<- function(paramètres/arguments) {
  commandes
  return(valeur/résultat de la fonction)
}
```

80



## fonction



```
MeanTemp<- function(Tmin, Tmax) {  
  Tmean<-(Tmin+Tmax)/2  
  return(Tmean)  
}
```

```
MeanTemp(c(10,12,11), c(15,20,16))
```

```
[1] 12.5 16.0 13.5
```

Ex : 06\_prog.r

81

**Cours9. Compléments : installer un  
package, aide, données manquantes -  
0.5h**

82

## COMPLEMENTS

### Utilisation d'un package non chargé, mais installé

```
library(nom_du_package)
library(help=nom_du_package)
remove.packages(nom_du_package)
```

### Quelques noms de packages utiles

utilitaire : xlsx

graphiques : lattice, plotrix, maptools, ggplot2

SIG : maps, maptools, shapefiles

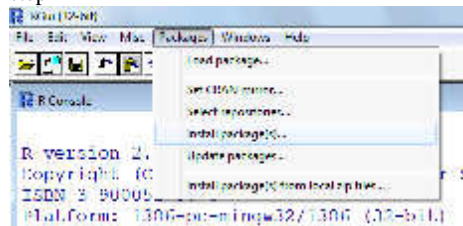
analyse statistique d'essais : agricolae

analyses multidimensionnelles: FactoMineR, rpart, randomForest, ade4

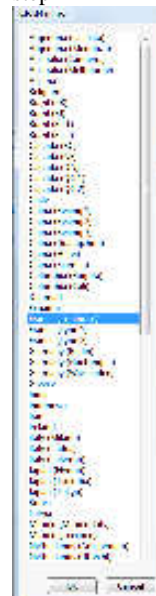
83

## Installation d'une librairie

step 1



step 2



step 3



step 4

```
install.packages("lattice")
# install.packages("lattice", type="source")
# install.packages("lattice", type="binary")
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/")
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/", quiet=FALSE)
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/", quiet=FALSE, verbose=TRUE)
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/", quiet=FALSE, verbose=TRUE, showProgress=TRUE)
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/", quiet=FALSE, verbose=TRUE, showProgress=TRUE, ask=TRUE)
# install.packages("lattice", type="source", repos="http://cran.rstudio.com/", quiet=FALSE, verbose=TRUE, showProgress=TRUE, ask=TRUE, install.args=c("--no-multiarch"))
```

84

## Pour aller plus loin...

- Nombreuses références...
- Venables W.N., Smith D.M. and the R Development Core Team 2010. An introduction to R (available online)
- <http://cran.r-project.org/manuals.html>
- Et la pratique...

85

## Aides

`?nom_fonction`  
`??nom_fonction`

### R site search

<http://finzi.psych.upenn.edu/search.html>

<http://r-project.markmail.org/search/>

### Forums

<http://forums.cirad.fr/logiciel-R/>

pdf gratuits en ligne



86

## Ressources supplémentaires

- Fournies dans le ZIP/clef USB
- Document; D:\R\_ITA\theorie
  - 01\_InitiationR\_ITA\_V4.ppt
  - 02\_Synthese\_fonctions.doc
  - sup\_Introduction\_To\_R\_Language\_EN.pdf
  - sup\_R\_exercices.doc
- Exemples; D:\R\_ITA\rscript\Autres\_resource
  - TP2\_ex.basique.vecteur.r
  - TP4\_bd\_postgreSQL.r
  - TP5\_exemple\_courbes.ggplot.R, TP5\_krigeage.r
  - TP5\_map.carte\_france.r, TP5\_plot.double.axes.r
  - TP5\_plot.NuagePoints\_&\_2axes.r
  - TP9\_ExempleComplet.r
  - TP9\_intallation.libraries.r

87

## Quelques liens utiles ....

<http://www.r-project.org/>

<http://finzi.psych.upenn.edu/search.html>

<http://pbil.univ-lyon1.fr/R/enseignement.html>

<http://forums.cirad.fr/logiciel-R/>

<http://www.oga-lab.net/RGM2/images.php?show=all&pageID=299>

<http://dirk.eddelbuettel.com/cranberries/>

<http://r-project.markmail.org/search/>

<http://research.stowers-institute.org/efg/R/Color/Chart/>

<http://www.springer.com/series/6991>

[http://www.crcpress.com/ecommerce\\_product/browse\\_book\\_categories.jsf?category=STA](http://www.crcpress.com/ecommerce_product/browse_book_categories.jsf?category=STA)

88

## Conclusion et suites à donner en terme de modules de formation avancée – qq mn

89

## Discussion

- Vos questions
- Vos besoins et suites à donner (une fois passée la phase d'appropriation)  
des formations-ateliers ?
  - **Questions statistiques particulières (à préciser)**
  - **Représentation graphique sous R**  
(création de graphiques, fonctionnalités avancées, SIG (librairies ggplot, lattice, maps))
  - **Travail avec les données sous R**  
(lecture des fichiers de données. Bases, problèmes rencontrés et solutions, conversions entre types de données (bases, librairie plyr,...), méthodes de sélection de données([], subset, constrasts))
  - **utilisation de gestionnaires de base de données / requêtes SQL**

90



À vous de jouer !