

INTRODUCTION AU LOGICIEL R

Club IAME-BioStats

25/01/2024

Présentée par:
Romain Leroux

Basée sur les cours de:
Emmanuelle Comets

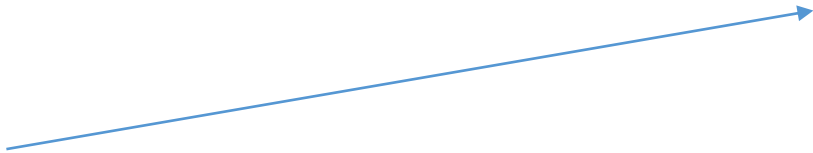
Séance 1 : Prise en main en R et Statistiques descriptives partie 1

- Vue d'ensemble & premiers pas
- Vecteur, Matrice
- Dataframe
- Statistiques descriptives (1)

Présentation

R est :

- une version « libre » du langage S
- un langage de programmation pour l'analyse et la modélisation des données
- un langage statistique contenant un très grand choix de techniques statistiques
- parmi les logiciels statistiques les plus utilisés
- dispose de modules et de librairies spécialisées disponibles sur Internet

- 
- pour executer le programme
 - pour etudier comment programme est construit et l'adapter à vos besoins
 - pour distribuer les copies du programme
 - pour ameliorer le programme et le metre à la disposition de la communauté des utilisateurs

Présentation

Avantages

- R existe sur tous les systèmes d'exploitation les plus fréquents : Windows, MacOS, Linux
- R est « gratuit »
- R est sans doute le logiciel actuel le plus « riche » en outils statistiques
- R fournit des graphiques de grande qualité
- R est sans cesse actualisé et possède la communauté d'utilisateurs très active
- R est « débuggé » très régulièrement
- R permet de construire facilement vos propres fonctions
- R est un logiciel mathématique (calcul matriciel, intégration numérique, optimisation, ...)

Inconvénients

- Interface basique
- R est un langage de programmation peu sécurisé

Installation de R (1)

- Sur le site: www.cran.r-project.org



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Mailing Lists

Bug Tracking

Development Site

Conferences

Search

R Foundation

Foundation

Board

Members

Donors

Donate

Documentation

Manuals

FAQs

The R Journal

Books

Certification

Other

Links

Bioconductor

Related Projects

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- **R version 3.2.4 (Very Secure Dishes)** has been released on Thursday 2016-03-10.
- **R version 3.3.0 (Supposedly Educational) prerelease versions** will appear starting Monday 2016-03-14. Final release is scheduled for Thursday 2016-04-14.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016**, will take place at Stanford University, CA, USA, June 27 - June 30, 2016.
- **The R Journal Volume 7/2** is available.
- **R version 3.2.3 (Wooden Christmas-Tree)** has been released on 2015-12-10.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.

Choisir un CRAN mirror:

France

<http://cran.univ-lyon1.fr/>

<https://mirror.ibcp.fr/pub/CRAN/>

<http://mirror.ibcp.fr/pub/CRAN/>

<http://cran.biotools.fr/>

<http://ftp.igh.cnrs.fr/pub/CRAN/>

<http://cran.irsn.fr/>

<https://cran.univ-paris1.fr/>

<http://cran.univ-paris1.fr/>

Dept. of Biometry & Evol. Biology, University of Lyon

CNRS IBCP, Lyon

CNRS IBCP, Lyon

IBDM, Marseille

Institut de Genetique Humaine, Montpellier

French Nuclear Safety Institute, Paris


SAMM, Université Paris 1 Panthéon-Sorbonne

SAMM, Université Paris 1 Panthéon-Sorbonne

Installation de R (2)

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#) 

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2015-12-10, Wooden Christmas-Tree) [R-3.2.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Subdirectories:

[base](#) 

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

[contrib](#)

Binaries of contributed CRAN packages (for R \geq 2.11.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

[old contrib](#)

Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.11.x; managed by Uwe Ligges).

[Rtools](#)

Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Installation de R (3)

Télécharger l'executable:



R-3.2.3 for Windows (32/64 bit)

[Download R 3.2.3 for Windows](#) (62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

Lancer l'installation par un double clic et suivre les Instructions

Lancer R

Installation de R (4)

Télécharger des extensions :

- Directement dans le R → Packages → Installer le package
- Sur le site cran.r: <https://cran.r-project.org/>
 - dans la rubrique Software → Packages → “MonPackage”
 - télécharger “MonPackage”.zip
 - installer le package dans R
 - charger le package

Editeurs pour R

- Rstudio : RStudio Desktop 1.4.1106

www.rstudio.com

www.rstudio.com/products/RStudio/

- Tinn-R: Tinn-R_04.00.03.05_setup

<https://sourceforge.net/projects/tinn-r/?source=directory>

L'avantage principal de Rstudio et Tinn-R est la coloration syntaxique, qui permet de mieux s'y retrouver.

- Editeur R
- Winedt
- Bloc note

Conseils:

Taper les commandes dans un fichier texte

- permet de corriger facilement ses erreurs sans tout retaper
- permet de garder la trace de son travail d'une session sur l'autre
- premiers pas vers la programmation

Ressources pour R

Manuels en Français :

- <https://cran.r-project.org/>
- “R pour les débutants”: Emmanuel Paradis
https://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf

Sites utiles:

- <https://www.rstudio.com/resources/cheatsheets/>
- <https://www.r-bloggers.com/>
- <https://stackoverflow.com/>

R- cours gratuits en ligne: www.coursera.org

<https://www.coursera.org>

coursera

Établissements

Connexion

S'inscrire

Suivez en ligne les meilleurs cours au monde.

Rechercher tout sujet



[Voir le catalogue complet](#)



Stanford

UC San Diego



R Programming

Université Johns-Hopkins

Partie d'une série de 10 cours, la Spécialisation **Data Science**

Session en cours :

mars 7 - avril 11

Les inscriptions prennent fin le mars

13

S'inscrire

La session suivante commence le
mars 14

<http://mooc-francophone.com/cours/mooc-introduction-statistique-r/>



[Accueil](#) [Cours](#) [Agenda](#) [Emploi](#) [Blog](#) [Contact](#)



Introduction à la statistique avec R

MOOC (Gratuit) FUN Mathématiques / Statistiques Terminée

Le MOOC Introduction à la statistique avec R, est une introduction aux concepts de base en statistique. Le recours aux mathématiques est minimal. L'objectif est de savoir analyser des données, de comprendre ce que l'on fait, et de pouvoir communiquer ses résultats.

Utilisation de l'aide (1)

- Pour éditer la fiche d'aide correspondant à une fonction, par exemple `mean()` :
 - `help(mean)`
 - ou `?mean`
- On obtient une fiche comprenant les sections :
 - **Description** : une brève description de l'usage de la fonction
 - **Usage** : les arguments de la fonction et leurs valeurs par défaut
 - **Arguments** : les paramètres devant être passés à la fonction
 - **Details** : des remarques particulières sur l'usage de la fonction
 - **Value** : le type d'objet retourné par la fonction
 - **See Also** : autres rubriques d'aide proches ou similaires
 - **Examples** : des exemples d'application

Utilisation de l'aide (2)

Comment lire les pages de l'aide:

- Lisez la courte introduction au début.
- Descendez à la fin et répéter les exemples !
- On comprend souvent rapidement l'idée de la fonction.
- Lisez les détails de la fonction plus tard.

```
mean {base}
```

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments

x An R object. Currently there are methods for numeric/logical vectors and [date](#), [date-time](#) and [time interval](#) objects. Complex vectors are allowed for `trim = 0`, only.

trim the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of `trim` outside that range are taken as the nearest endpoint.

na.rm a logical value indicating whether NA values should be stripped before the computation proceeds.

... further arguments passed to or from other methods.

Value

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If `x` is not logical (coerced to numeric), numeric (i

If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

[weighted.mean](#), [mean.POSIXct](#), [colMeans](#) for row and column means.

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

Recherche

Quand on ne sait pas trop ce qu'on cherche... Par exemple, j'aimerais trouver une fonction qui fasse des permutations :

- `help.search("permutation")`
- ou `??permutation`

Help files with alias or concept or title matching 'permutation' using fuzzy matching:

- `base::order` Ordering Permutation
- `base::sample` Random Samples and Permutations
- `combinat::permn` Generates all permutations of the elements
- `e1071::permutations` All Permutations of Integers 1:n
- `gbm::relative.influence` Methods for estimating relative influence
- `gdata::resample` Consistent Random Samples and Permutations
- `gtools::combinations` Enumerate the Combinations or Permutations the Elements of a Vector
- `Matrix::pMatrix-class` Permutation matrices
- `zoo::ORDER` Ordering Permutation

Note: le premier élément est le nom de la librairie (package).

Exercice 0

- Premiers pas
 - lancer R
 - ouvrir un nouveau script
 - donner le résultat du calcul de: $2 * \text{sqrt}(5) + \pi$
- Aide
 - consulter la fiche d'aide de R sur la fonction **log**
 - calculer la valeur du logarithme de 2 en base 10 d'une autre manière en utilisant cette page d'aide

Les bases du langage R (1)

- Comment utiliser R:

- Comme une calculatrice :

2+3

[1] 5

4-3

[1] 1

- Opérateurs simples :

+ - * / ^

2*5

[1] 10

112/4

[1] 28

2^3

[1] 8

round(2.7) (arrondit les éléments de x à n chiffres après la virgule)

[1] 3

round(2.3)

[1] 2

floor(2.7) (l'entier inferieur à la valeur du x)

[1] 2

ceiling(2.7) (l'entier supérieur à la valeur du x)

[1] 3

ceiling(2.3)

[1] 3

trunc(2.3)

[1] 2

trunc(-2.3)

[1] -2

ceiling(-2.3)

[1] -2

floor(-2.3)

[1] -3

Les bases du langage R (2)

- Comment utiliser R:
 - Comme un langage

Créer un objet:

```
x<-10
```

Afficher l'objet:

```
x
```

```
[1] 10
```

Effectuer des calculs :

```
x^2
```

```
[1] 100
```

Vecteurs

- Création d'un vecteur

On utilise les fonctions **c**, **seq** et/ou **rep**, et l'opérateur **:**«

```
a<- c(1,3,5)
```

```
a
```

```
[1] 1 3 5
```

```
b<- seq(1,4, by=0.5)
```

```
b
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0
```

```
colors<- c("Red","Green","Red")
```

```
colors
```

```
[1] "Red" "Green" "Red"
```

```
c<-rep(1,5)
```

```
c
```

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

```
d<-1:10
```

```
d
```

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

Type d'un vecteur (1)

En gros, le type d'un objet est déterminé par la taille mémoire nécessaire pour le stocker. Pour un ordinateur, un entier prend moins de place qu'un nombre réel.

- On utilise la fonction **typeof** pour connaître le type d'un objet :

typeof(a)

[1] "double"

typeof(c)

[1] "character"

Type d'un vecteur (2)

- Le type du vecteur s'ajuste automatiquement aux éléments, en prenant le type permettant de stocker tous les éléments.

```
cha<-c(1:4,1.5,"toto")
```

```
typeof(cha)
```

```
[1] "character"
```

- On peut utiliser la fonction **as.double** pour transformer ces chaînes de caractères en nombre

```
as.double(cha)
```

```
[1] 1.0 2.0 3.0 4.0 1.5 NA
```

Warning message:

NAs introduits lors de la conversion automatique

Opérations sur des vecteurs

1) Sélectionner un élément d'un vecteur:

```
x<-10:16
```

```
x
```

```
[1] 10 11 12 13 14 15 16
```

```
x[2]
```

```
[1] 11
```

2) Sélectionner le 2e et 4e élément de x :

```
x[c(2,4)]
```

```
[1] 11 13
```

3) Sélection négative : tous les éléments sauf le 4^e:

```
x[-4]
```

```
[1] 10 11 12 14 15 16
```

5) Les opérations arithmétiques : $+$, $-$, $*$, $/$, $^$ (ou $**$)

Toutes ces opérations se font élément par élément :

```
x<- seq(1,4)
```

```
x
```

```
[1] 1 2 3 4
```

```
x+2
```

```
[1] 3 4 5 6
```

```
x/3
```

```
[1] 0.3333333 0.6666667 1.0000000 1.3333333
```

6) Les opérations logiques : $==$ (égalité), $!=$ (différence), $<$, $<=$, $>$, $>=$ (infériorité /supériorité)

```
x>0
```

```
[1] TRUE TRUE TRUE TRUE
```

```
x==3
```

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

```
x[x>2]
```

 Utilisé pour sélectionner les indices de x correspondant à un test vrai

```
[1] 3 4
```

Exercice 1 : Vecteurs

A chaque fois, afficher le résultat demandé

- Créer un vecteur x contenant les éléments 1, 4 et 5
- Créer un vecteur y contenant les chiffres de 1 à 9
 - afficher le 2ème élément de y
 - afficher tous les éléments de y sauf le 2ème
 - afficher les éléments 2 à 4 de y
- Créer un vecteur xy contenant le premier, quatrième et cinquième élément de y

Opérations sur deux vecteurs

- Exemple : multiplication:

```
x<-1:5
```

```
x
```

```
[1] 1 2 3 4 5
```

```
x*x
```

```
[1] 1 4 9 16 25
```

Chaque élément de x est multiplié par lui-même \Rightarrow on a calculé le carré de x.

- Vecteurs de longueur inégale :

```
y<-1:3
```

```
y
```

```
[1] 1 2 3
```

```
x*y
```

```
[1] 1 4 9 4 10
```

Message d'avis : la longueur de l'objet le plus long n'est pas un multiple de la longueur de l'objet le plus court in: x * y

Un message d'erreur apparaît ici.

Les fonctions utiles



1) ordre : **sort** (dans l'ordre ascendant), **rev** (dans l'ordre descendant), **order** (dans l'ordre ascendant ou descendant), **rank**

```
sort(1:10,decreasing=T)
```

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

```
x<-c(1,4,3,8,5)
```

```
rank(x)
```

```
[1] 1 3 2 5 4
```

2) agrégation : **cbind**, **rbind**

```
a <- 1:2
```

```
b <- 3:4
```

```
rbind(a,b)
```

```
  [,1] [,2]
```

```
a    1    2
```

```
b    3    4
```

```
cbind(a,b)
```

```
  a    b
```

```
[1,] 1    3
```

```
[2,] 2    4
```

3) propriétés : **length**, **mode**

length(a)

```
[1] 2
```

mode(a)

```
[1] "numeric"      ("integer" ou "double")
```

4) fonctions mathématiques : **exp**, **sqrt**, **cos**, **sum**, ...

sum(x)

```
[1] 15
```

5) fonctions de manipulation : **unique**, **duplicated**

unique(c(1,1,2,2))

```
[1] 1 2
```

duplicated (c(1,1,2,2))

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

Les données manquantes

- Elles sont codées en NA (Not Available) aussi NaN (Not A Number) :

```
y <- c(1:4,NA,rep(0.8,2),NA)
```

```
y
```

```
[1] 1.0 2.0 3.0 4.0 NA 0.8 0.8 NA
```

- Attention, un test impliquant une donnée NA donnera un résultat NA :

```
y[y>0]
```

```
[1] 1.0 2.0 3.0 4.0 NA 0.8 0.8 NA
```

- Pour "tester" la présence de données manquantes :

```
is.na(y)
```

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

```
y[is.na(y)==F] # afficher les données non-manquantes
```

```
[1] 1.0 2.0 3.0 4.0 0.8 0.8
```

```
y[!is.na(y)] #même chose
```

```
[1] 1.0 2.0 3.0 4.0 0.8 0.8
```

- Donc pour afficher uniquement les éléments positifs et non manquants de y, il faut faire un double test ("ET") :

```
y[!is.na(y) & y>0]
```

```
[1] 1.0 2.0 3.0 4.0 0.8 0.8
```

- Attention, certaines fonctions retournent NA si on les applique un vecteur contenant des NA :

```
mean(y)
```

```
[1] NA
```

```
mean(y,na.rm=T)
```

```
[1] 1.933333
```

Exercice 2 : manipulation de vecteurs

- 1) Créer un vecteur **vec** de taille 10 avec la commande `rnorm(10)`.
- 2) Calculer le nombre d'éléments positifs de `vec`.
- 3) Calculer `yvec` dont les éléments sont composés du logarithme des éléments de `vec`.
- 4) Créer le vecteur `vec2` avec seulement les éléments non manquants de `yvec`.
- 5) Afficher le nombre d'éléments de `vec2`.

Matrices

Création d'une matrice (1)

- En partant d'un vecteur, avec remplissage par colonne (par défaut) :

```
x<-matrix(1:9,ncol=3)
```

x

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

- ou avec remplissage par ligne :

```
x<-matrix(1:9,ncol=3,byrow=T)
```

x

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

Création d'une matrice (2)

A partir d'un vecteur `vec`, `dim(vec)<-c(n,m)` crée une matrice à `n` lignes et `m` colonnes :

```
x<-1:12
```

```
dim(x)<-c(3,4)
```

```
x
```

```
 [,1] [,2] [,3] [,4]
```

```
[1,] 1   4   7  10
```

```
[2,] 2   5   8  11
```

```
[3,] 3   6   9  12
```

Ici encore par défaut le vecteur `x` vient se ranger en colonnes de gauche à droite.

Création d'une matrice (3)

A partir de 2 ou plusieurs vecteurs, soit ligne par ligne (**rbind**), soit colonne par colonne (**cbind**).



```
x<-1:4
```

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

```
z<-x^3
```

```
mat<-rbind(x,y,z)
```

```
mat
```

```
  [,1] [,2] [,3] [,4]
```

```
x   1    2    3    4
```

```
y   1    4    9   16
```

```
z   1    8   27   64
```

```
mat<-cbind(x,y,z)
```

```
mat
```

```
  x  y  z
```

```
[1,] 1  1  1
```

```
[2,] 2  4  8
```

```
[3,] 3  9 27
```

```
[4,] 4 16 64
```

Bien sûr il faut que les vecteurs aient la même taille (sinon échec avec message d'erreur).

Création d'une matrice (4)

Matrice diagonale avec `diag(x)`

```
y<-1:3
```

```
diag(y)
```

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

```
[1,] 1  0  0
```

```
[2,] 0  2  0
```

```
[3,] 0  0  3
```

Éléments d'une matrice (1)

```
x<-matrix(1:9,ncol=3,byrow=T)
```

X

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

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

```
[2,]  4  5  6
```

```
[3,]  7  8  9
```

Pour récupérer la deuxième colonne de la matrice x :

```
x[,2]
```

```
[1] 2 5 8
```

Pour la troisième ligne :

```
x[3, ]
```

```
[1] 7 8 9
```

Pour le deuxième élément de la troisième ligne :

```
x[3,2]
```

```
[1] 8
```

Pour avoir toute la deuxième ligne sauf l'élément de la deuxième colonne

```
x[2,-2]
```

Éléments d'une matrice (2)

Comme pour les vecteurs, on peut sélectionner plusieurs lignes à la fois :

`x[1:2,]`

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

On peut sélectionner en même temps sur les lignes et les colonnes, on obtient les cellules situées à l'intersection des 2 conditions :

`x[1:2,1:2]`

[,1] [,2]

[1,] 1 2

[2,] 4 5

A la place de lignes consécutives (ici 1:2), on peut introduire un vecteur d'indices :

`x[c(1,3),]`

[,1] [,2] [,3]

[1,] 1 2 3

Éléments d'une matrice (3)

x

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	4	5	6
[3,]	7	8	9

Le vecteur d'indices peut être défini comme une condition :

x[x[,1] > 1,]

	[,1]	[,2]	[,3]
[1,]	4	5	6
[2,]	7	8	9

Ce qui se lit : "on choisit parmi les lignes de x celles pour lesquelles, dans la première colonne, la valeur est supérieure strictement à 1" (dans la partie ligne) et "toutes les colonnes" (dans la partie colonne).

Exercice 3 - Matrices

- 1) Créer une matrice à 3 lignes et 5 colonnes ayant pour éléments 15 chiffres entre 1 et 20
- 2) Extraire la sous-matrice formée par les 2 dernières lignes et les colonnes 2 et 4

Dataframe

- Format d'un dataframe
- Ecriture et lecture de fichiers de données
- Manipulation des dataframes

Format d'un dataframe

- Format de jeu de données structuré
- Les “dataframes” ressemblent à des matrices, mais sont beaucoup plus flexibles. Ils sont utilisés dans la plupart des techniques statistiques à disposition dans R
- Nombreux jeux de données structurés disponibles sous R
 - Exemple : jeu de données pressure sur la relation entre la température en degrés et la pression en millimètres de mercure

```
> pressure
  temperature pressure
1           0  0.0002
2          20  0.0012
...
19        360 806.0000
```

=> Voir script R : visualiser les premières lignes, récupérer une variable, attacher/détacher un dataframe

Ecriture de fichiers de données

- Créer des dataframes avec
 - `data.frame()`
 - `as.data.frame()`
- Sauver un jeu de données dans un fichier sous différents formats .dat, .txt, .csv
 - `write.table()`

⇒ Voir script R

Lecture de fichiers de données

- Les fonctions `read.table`, `read.csv`

=> Voir script R

Manipulation des dataframes (1)

- Extraction d'une sous-base avec `subset()` etc.
- Transformation de données, création de nouvelles variables
 - `ifelse()` pour variable indicatrice

=> Voir script R

Exercice 5

- Créer à partir de `air1` le dataframe `air2`, où :
 - 1) la variable `Ozone` n'est pas manquante
 - 2) la température est $\leq 94^{\circ}F$
- Créer à partir de `airquality` le dataframe `air3`, où la variable `Ozone` n'est pas manquante.
- Ajouter une colonne à `air3` représentant une variable valant 1 si :
 - 1) on est dans les 6 premiers mois de l'année
 - 2) la température est supérieure à $80^{\circ}F$et 0 sinon.

Manipulation des dataframes (2)

- Tri sur une ou plusieurs variables avec `order()`
- La fonction `match()` teste si un élément est présent dans un vecteur : outil puissant pour manipuler plusieurs bases de données avec un identifiant commun (ex : le numéro d'identification du patient).

=> Voir script R

Exercice 6

- On vous donne un relevé des profondeurs de glace relevées dans une station météo avec les dates correspondantes :

```
dates<-c("1971-01-20", "1971-01-28", "1971-02-03", "1971-02-11", "1971-02-18", "1973-01-17", "1973-01-25", "1973-01-31", "1973-02-17", "1974-01-07", "1974-01-10", "1974-01-15", "1974-01-22", "1974-01-29", "1974-02-05", "1974-02-12", "1974-02-19")
```

```
measure<-c(64, 69, 71, 71, 71, 32, 42, 28, 32, 18, 25, 29, 34, 36, 42, 50, 61)
```

- Extraire de ce vecteur la première incidence de chaque profondeur mesurée, en utilisant la fonction match.
- Trier les mesures de glace dans l'ordre croissant et créer une matrice avec comme première colonne les mesures triées et comme deuxième colonne les dates correspondantes

Statistiques descriptives (1)

- Moyenne et médiane : fonction **mean**
- Médiane : fonction **median**
- Variance : fonction **var**
- Ecart-type : fonction **sd**
- Option **na.rm=T** pour ignorer les valeurs manquantes

Statistiques descriptives (2)

- Corrélation : fonction `cor(x,y)`, par défaut, corrélation de Pearson
- Options
 - `method` : deux autres coefficients de corrélation (association basée sur les rangs)
 - `method="kendall"` : τ de Kendall (coefficient de corrélation des rangs de Kendall)
 - `method="spearman"` : ρ de Spearman
 - `use="c"` : pour ignorer les valeurs manquantes
- Covariance: fonction `cov(x,y)`
 - mêmes options que pour `cor` : `method`, `na.rm=T`
 - `x` et `y` peuvent être des matrices : `cov` renvoie alors la matrice de variance-covariance entre les colonnes de `x` et de `y`

Statistiques descriptives (3)

- On peut obtenir les quantiles empiriques d'un vecteur:

```
> quantile(x)
```

```
0%      25%      50%      75%      100%
```

```
2.525075 4.413246 4.897222 5.528062 7.327059
```

- Par défaut, on a le min, le max, et les 3 quartiles (0.25, 0.5 et 0.75 quantiles)
- Pour obtenir les autres quantiles, par exemple les déciles:

```
> dec<-seq(0,1,0.1)
```

```
> dec
```

```
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

```
> quantile(x,dec)
```

```
0%      10%      20%      30%      40%      50%
2.525075 3.538811 4.211671 4.468975 4.648577 4.897222
60%      70%      80%      90%      100%
5.190315 5.459779 5.785553 6.198171 7.327059
```

Statistiques descriptives (4) : fonctions appliquées à un tableau

- Utiliser `summary()` pour tableaux ou dataframes
- Plus généralement, la fonction `apply()` sert à appliquer une autre fonction de façon répétée sur les lignes ou les colonnes d'un tableau.
Par exemple :

```
> apply(x,2,quantile)
```

calcule les quantiles par défaut pour les colonnes de la matrice x (avec 1 à la place de 2, ce serait pour les lignes). Il est possible de spécifier des options pour la fonction appelée par apply

```
> apply(x,2,quantile,c(0.05,0.95))
```

Statistiques descriptives (5) : exercice 7

- Créez une matrice x portant les chiffres de 1 à 100, en 4 colonnes
- Calculez la moyenne et la variance
 - de x
 - des colonnes de x
 - des 3 premières lignes de x
- Prendre le jeu de données interne à R appelé ToothGrowth
 - le visualiser pour se faire une idée de son contenu (utilisez l'aide)
 - la première colonne représente la longueur des dents de cochons d'inde soumis à différents régimes
 - ranger la première colonne par colonnes dans une matrice à 6 colonnes mat
 - calculer la matrice de corrélations entre les colonnes de mat