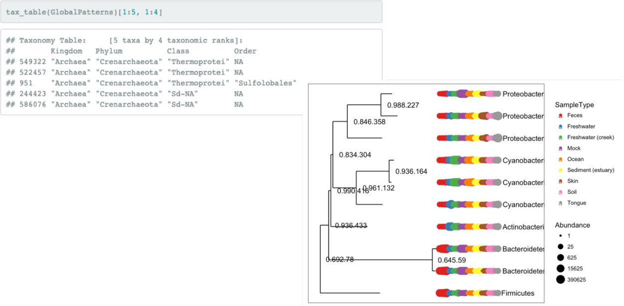
R and Rstudio :Introduction and getting started

Marc GAREL (PhD)
Mediterranean Institute of Oceanography

September 4th 2023



```
tax_table(GlobalPatterns)[1:5, 1:4]
```

```
## Taxonomy Table:     [5 taxa by 4 taxonomic ranks]:
##        Kingdom   Phylum          Class          Order
## 549322 "Archaea" "Crenarchaeota" "Thermoprotei" NA
## 522457 "Archaea" "Crenarchaeota" "Thermoprotei" NA
## 951    "Archaea" "Crenarchaeota" "Thermoprotei" "Sulfolobales"
## 244423 "Archaea" "Crenarchaeota" "Sd-NA"        NA
## 586076 "Archaea" "Crenarchaeota" "Sd-NA"        NA
```

What is cran R?

R?

Statistic
Data science

Interpreted languages

Simple syntax

Free
Muti-OS

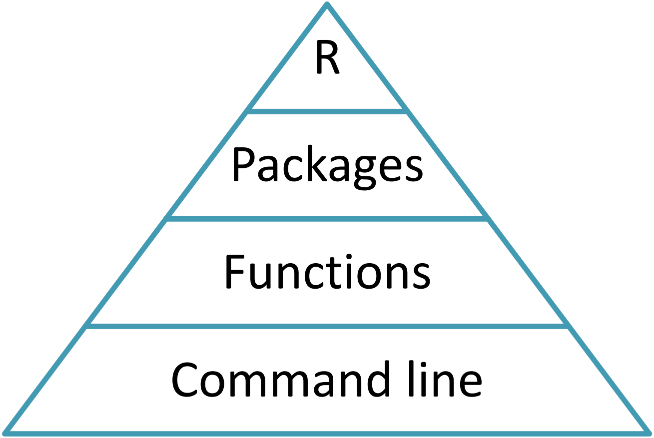
Gorgeous graph output

Reproducibility

Large community

Institut Méditerranéen d'Océanologie

What is cran R?



R

Packages

Functions

Command line

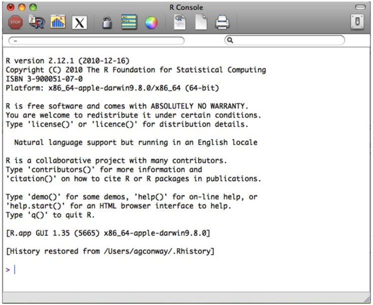
What is cran R?

Here are some examples of the core packages that are commonly included with R:

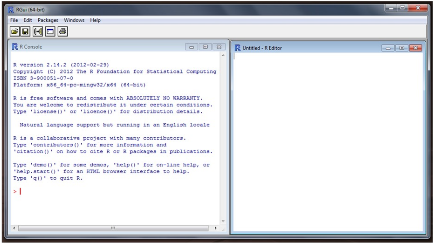
1. **base** - This is the base R package that provides essential functions and data types.
2. **datasets** - This package contains various datasets that are frequently used for examples and testing.
3. **graphics** - Provides functions for creating graphical plots and charts.
4. **stats** - Includes statistical functions and models, such as linear regression and hypothesis testing.
5. **…**

Institut Méditerranéen d'Océanologie

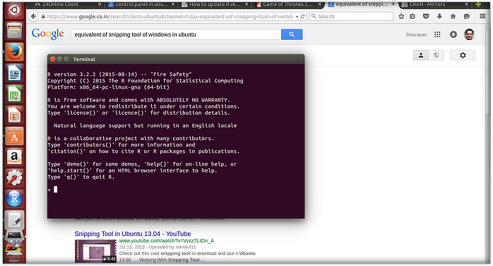
What is cran R?



```
R version 2.12.1 (2010-12-16)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (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.

[R.app GUI 1.35 (5665) x86_64-apple-darwin9.8.0]

[History restored from /Users/agconway/.Rhistory]

> |
```



```
R version 2.14.2 (2012-02-29)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (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.

> |
```



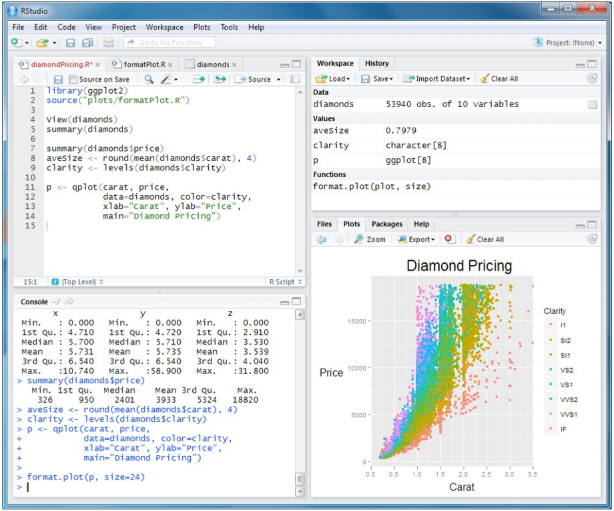
equivalent of snipping tool of windows in ubuntu

```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 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.

> 
```

Snipping Tool in Ubuntu 13.04 - YouTube
www.youtube.com/watch?v=Vvzz7LIOn_A
Jul 13, 2013 - Uploaded by teklek411
Check out this cool **snipping tool** to download and use n **Ubuntu** 13.04. ... Working With **Snipping Tool** ...

Rstudio

Rstudio is an integrated development environment (IDE)



```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")
```

Data

diamonds 53940 obs. of 10 variables

Values

aveSize 0.7979

clarity character[8]

p ggplot[8]

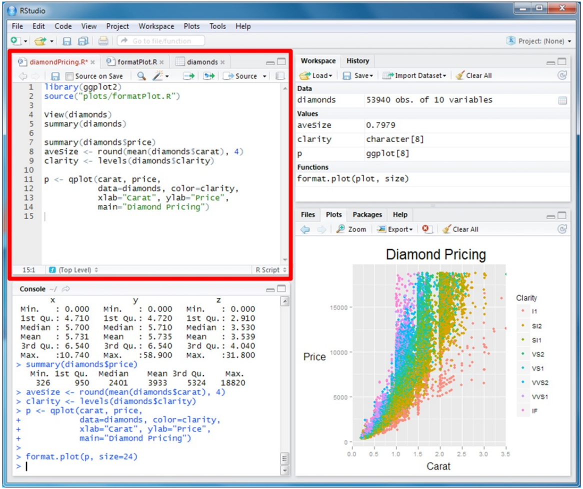
Functions

format.plot(plot, size)

```
       x                y                 z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

Rstudio

An editing window opens where you can write and save a series of instructions in a file: your **script!**



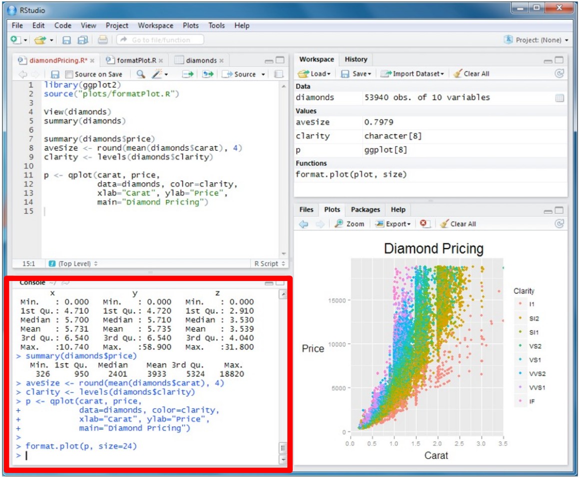
```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

```
Data
diamonds    53940 obs. of 10 variables
Values
aveSize     0.7979
clarity     character[8]
p           ggplot[8]
Functions
format.plot(plot, size)
```

```
       x                y                z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

Rstudio

A terminal window giving all the output from the script



```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

Data

diamonds	53940 obs. of 10 variables

Values

aveSize	0.7979
clarity	character[8]
p	ggplot[8]

Functions

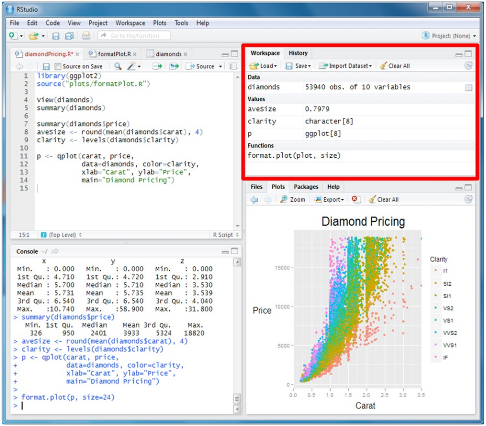
format.plot(plot, size)

```
       x               y               z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> 
```

Institut Méditerranéen d'Océanologie

Rstudio

A Workspace in which all the objects created/generated by the code are displayed (*i.e.* Matrices, Vectors, Functions, Constants...). It includes a History tab which returns all the executed instructions.



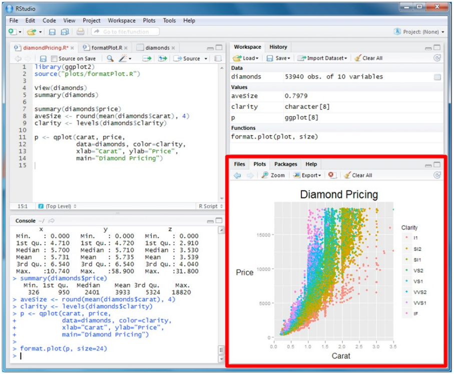
```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

```
       x               y               z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    326     950    2401    3933    5324   18820 
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
> 
> format.plot(p, size=24)
> 
```

```
Data
diamonds      53940 obs. of 10 variables
Values
aveSize       0.7979
clarity       character[8]
p             ggplot[8]
Functions
format.plot(plot, size)
```

Rstudio

This last window contains 4 tabs :
Files (locates your working directory and allows you to navigate),
Plots (returns all graphic outputs),
Packages (package management)
Help (this is R's 112 or 911, contains all of R's help)



```r
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

```
       x                 y                 z         
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 5.700   Median : 5.710   Median : 3.530  
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539  
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :10.740   Max.   :58.900   Max.   :31.800  
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    326     950    2401    3933    5324   18820 
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
> 
> format.plot(p, size=24)
> 
```

Data

diamonds	53940 obs. of 10 variables

Values

aveSize	0.7979
clarity	character[8]
p	ggplot[8]

Functions

format.plot(plot, size)

The commandes on R:

An R command is either an expression or an assignment :

- Expression is directly evaluated and the result is displayed on terminal

  Exemple :

```
>2+3
[1]5
>sqrt(25)
[1]5
```

Mathematical calcul

The commandes on R:

An R command is either an expression or an assignment :

- An assignement is an expression stored in object or variable

Exemple :

```
>a<-5
>a
[1]5
```

```
a <-  5
```

Object is named a

Value stored is 5

Sign used for d'affectation

<- ou =

Why store?

To manipulate later

The commandes on R:

An R command is either an expression or an assignment :

- An assignement is an expression stored in object or variable

Exemple :

```
>a<-5
>a
[1]5
```

```
a <-  5
class(a)
[1] Num
```

It is very important to know what type of object

The commandes on R:

An R command is either an expression or an assignment :

- Object can be :

  Character : (”A”, ”B”, “C”, ..)

  Numeric : (1,2, 5.5, 6.7, …)

  Logic : (TRUE, FALSE)

- But also more complex:

  Vector

  Matrix

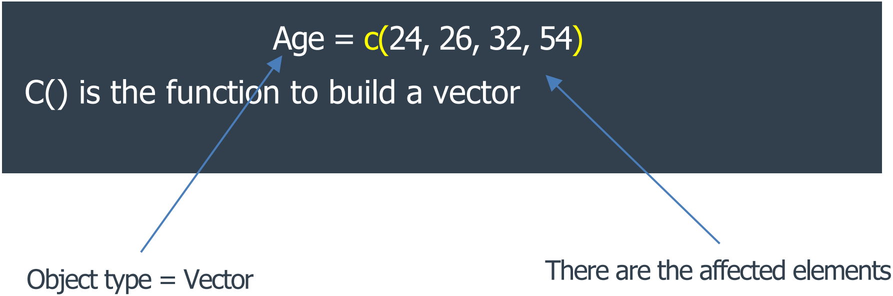
  Data frame

  List

The commandes on R:

Object Vector

- Vector contain only element with same mode (ex : numeric or character or logic ..)



```
Age = c(24, 26, 32, 54)
```

C() is the function to build a vector

Object type = Vector

There are the affected elements

The commandes on R:

What is the structure of my object ?

Function class() and str()

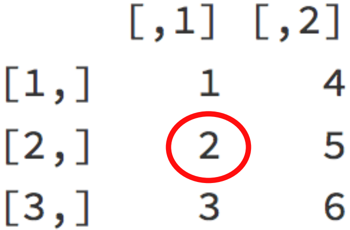
```
Age = c(24, 26, 32, 54)
str(Age)
Num[1:4]
class(Age)
Numeric
```

During the training we will see, how can we choose element inside a Vector or filter a vector

The commandes on R:

Object Matrix

- Matrix contains several vectors that build a table with columns and rows. The vectors belong to the same mode.



```
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

The commandes on R:

Object data frame

- Data frame contains several vectors that build a table with columns and rows. The vectors can belong to different modes

colnames

rownames

	Nom	Prenom	Age	Region_naissance
A	Keïta	Modibo	34	1
B	Traoré	Moussa	23	2
C	Konaré	Oumar	21	2
D	Touré	Toumani	43	4
E	Keïta	Boubacar	54	3

row.names() : row name (i.e. sample name)
colnames() : header(Nom, Prenom, etc)

Institut Méditerranéen d'Océanologie

The Workflow

Create Project
- Allow to save working environment automtically
- Keep the working environment open (scripts …)

Check/Install packages
- install.packages(“your packages”)
- BiocManager::install(“dada2”)

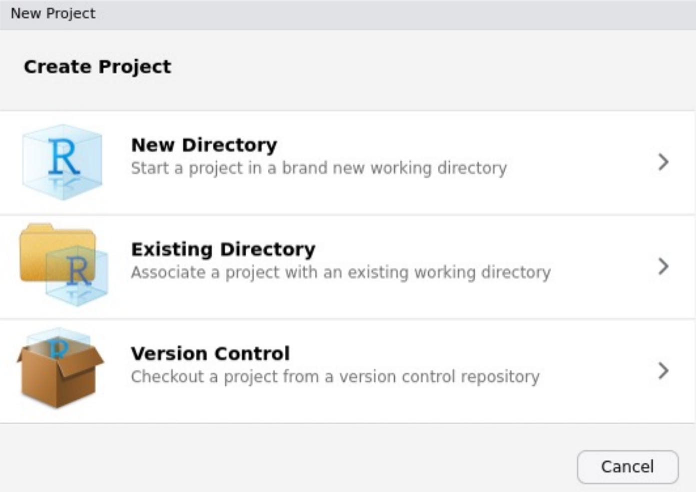
Load library
- library(“dada2”)
- dada2::function

Execute function
- merge.Pairs(dadaFs, derepFS, verbose = TRUE)

Work with projects with RStudio

Environment is empty

Work with projects with RStudio



New Project

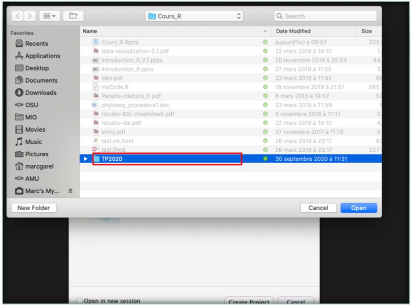
Create Project

New Directory
Start a project in a brand new working directory

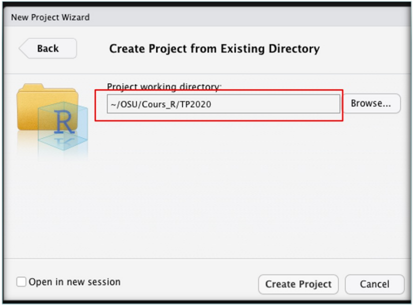
Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Work with projects with RStudio



Name	Date Modified	Size
Cours_R.Rproj	aujourd'hui à 09:57	205
data-visualization-2.1.pdf	22 mars 2018 à 14:10	1
Introduction_R_V2.pptx	20 novembre 2019 à 20:54	44
Introduction_R.pptx	27 mars 2018 à 11:05	3
labs.pdf	23 mars 2018 à 11:42	68
myCode.R	19 novembre 2019 à 21:51	565
Paradis-rdebuts_fr.pdf	6 mars 2013 à 19:57	57
phyloseq_procedure1.doc	23 mars 2018 à 11:59	
rstudio-IDE-cheatsheet.pdf	4 novembre 2016 à 11:17	3
rstudio-ide.pdf	22 mars 2018 à 14:10	4
shiny.pdf	27 novembre 2017 à 11:18	5
test.nb.html	26 mars 2018 à 23:17	83
test.Rmd	26 mars 2018 à 23:17	227
TP2020	30 septembre 2020 à 11:31	

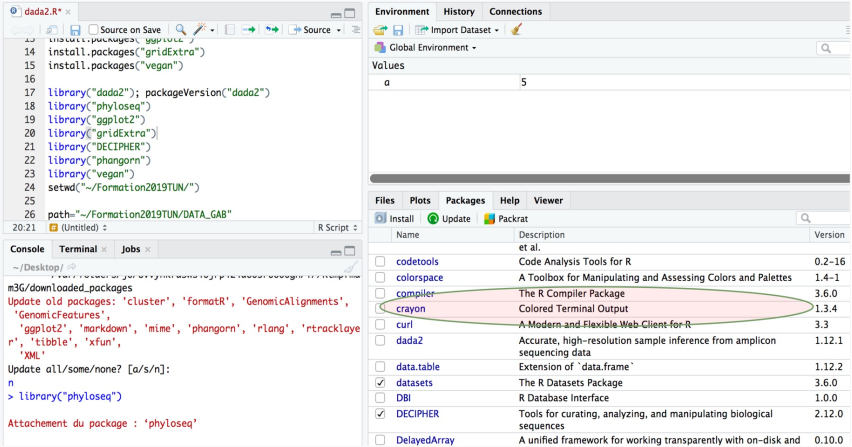


Create Project from Existing Directory

Project working directory: ~/OSU/Cours_R/TP2020

☐ Open in new session

Check install packages



```r
install.packages("gridExtra")
install.packages("vegan")

library("dada2"); packageVersion("dada2")
library("phyloseq")
library("ggplot2")
library("gridExtra")
library("DECIPHER")
library("phangorn")
library("vegan")
setwd("~/Formation2019TUN/")

path="~/Formation2019TUN/DATA_GAB"
```

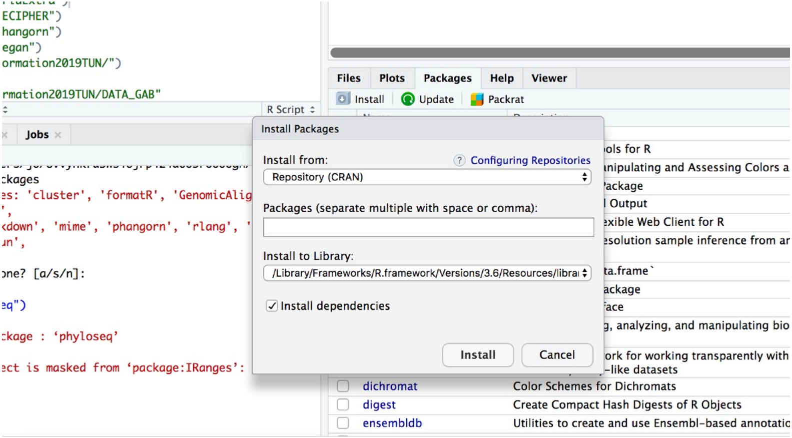
```
m3G/downloaded_packages
Update old packages: 'cluster', 'formatR', 'GenomicAlignments',
 'GenomicFeatures',
  'ggplot2', 'markdown', 'mime', 'phangorn', 'rlang', 'rtracklaye
r', 'tibble', 'xfun',
  'XML'
Update all/some/none? [a/s/n]:
n
> library("phyloseq")

Attachement du package : ‘phyloseq’
```

Values

a	5

	Name	Description	Version
		et al.	
☐	codetools	Code Analysis Tools for R	0.2-16
☐	colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	1.4-1
☐	compiler	The R Compiler Package	3.6.0
☐	crayon	Colored Terminal Output	1.3.4
☐	curl	A Modern and Flexible Web Client for R	3.3
☐	dada2	Accurate, high-resolution sample inference from amplicon sequencing data	1.12.1
☐	data.table	Extension of \`data.frame\`	1.12.2
☑	datasets	The R Datasets Package	3.6.0
☐	DBI	R Database Interface	1.0.0
☑	DECIPHER	Tools for curating, analyzing, and manipulating biological sequences	2.12.0
☐	DelayedArray	A unified framework for working transparently with on-disk and	0.10.0

Check install packages :CRAN package



```
ECIPHER")
hangorn")
egan")
ormation2019TUN/")

rmation2019TUN/DATA_GAB"
```

```
ckages
es: 'cluster', 'formatR', 'GenomicAlig
',
kdown', 'mime', 'phangorn', 'rlang', '
un',

one? [a/s/n]:

eq")

ckage : ‘phyloseq’

ect is masked from ‘package:IRanges’:
```

Install Packages

Install from: Repository (CRAN)

Packages (separate multiple with space or comma):

Install to Library: /Library/Frameworks/R.framework/Versions/3.6/Resources/librai

☑ Install dependencies

		ols for R
		nipulating and Assessing Colors a
		ackage
		Output
		exible Web Client for R
		esolution sample inference from ar
		ta.frame`
		ackage
		face
		g, analyzing, and manipulating bio
		ork for working transparently with -like datasets
☐	dichromat	Color Schemes for Dichromats
☐	digest	Create Compact Hash Digests of R Objects
☐	ensembldb	Utilities to create and use Ensembl-based annotatic

Check install packages :Bioconductor package

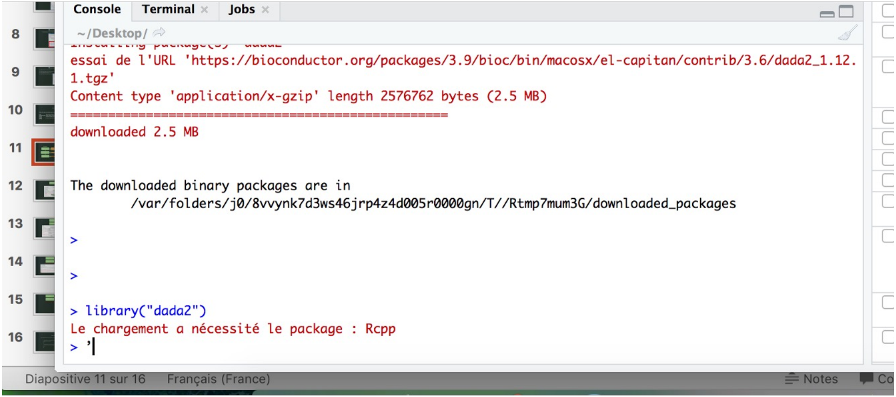
Command line is mandatory

```
library("ggplot2")
library("gridExtra")
library("DECIPHER")
library("phangorn")
library("vegan")
setwd("~/Formation2019TUN/")

path="~/Formation2019TUN/DATA_GAB"
```

```
>
>
> BiocManager::install("dada2", version = "3.9")
Bioconductor version 3.9 (BiocManager 1.30.4), R 3.6.0 (2019-04-26)
Installing package(s) 'dada2'
essai de l'URL 'https://bioconductor.org/packages/3.9/bioc/bin/macosx/el-capitan/contrib/3.6/dada2_1.12.
1.tgz'
Content type 'application/x-gzip' length 2576762 bytes (2.5 MB)
==================================================
downloaded 2.5 MB
```

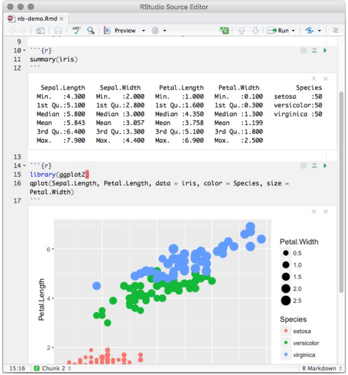
	Name	Description
☐	fastmatch	Fast match() fu
☐	foreach	Provides Foreac for R
☐	foreign	Read Data Store 'SAS', 'SPSS', 'Sta 'dBase', ...
☐	formatR	Format R Code
☐	Formula	Extended Mode
☐	futile.logger	A Logging Utilit
☐	futile.options	Futile Options
☐	gdata	Various R Progr Data Manipulati
☐	GenomeInf...	Utilities for mar chromosome na

Load library to execute function



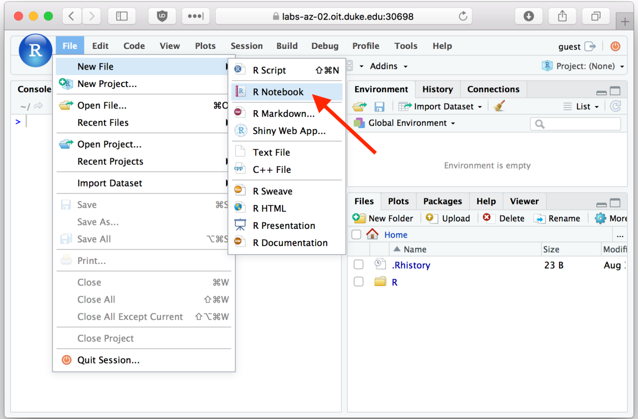
```
essai de l'URL 'https://bioconductor.org/packages/3.9/bioc/bin/macosx/el-capitan/contrib/3.6/dada2_1.12.
1.tgz'
Content type 'application/x-gzip' length 2576762 bytes (2.5 MB)
==================================================
downloaded 2.5 MB


The downloaded binary packages are in
	/var/folders/j0/8vvynk7d3ws46jrp4z4d005r0000gn/T//Rtmp7mum3G/downloaded_packages

>

>

> library("dada2")
Le chargement a nécessité le package : Rcpp
> ’
```

R Notebook



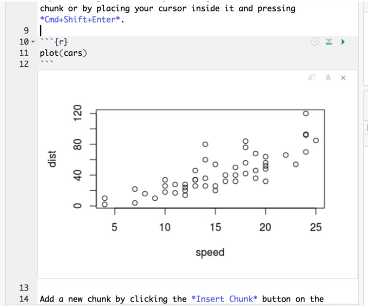
~~~
```{r}
summary(iris)
~~~
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

```
```{r}
library(ggplot2)
qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size =
Petal.Width)
```
```

An R Notebook, is a document with chunks that can be executed independently and interactively, with output visible immediately beneath the input
```

# R Notebook


labs-az-02.oit.duke.edu:30698
File
Edit
Code
View
Plots
Session
Build
Debug
Profile
Tools
Help
guest
Addins
Project: (None)
New File
New Project...
Open File... ⌘O
Recent Files
Open Project...
Recent Projects
Import Dataset
Save ⌘S
Save As...
Save All ⌥⌘S
Print...
Close ⌘W
Close All ⇧⌘W
Close All Except Current ⇧⌥⌘W
Close Project
Quit Session...
R Script ⇧⌘N
R Notebook
R Markdown...
Shiny Web App...
Text File
C++ File
R Sweave
R HTML
R Presentation
R Documentation
Console
~/
>
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Upload
Delete
Rename
More
Home
Name
Size
Modifi
.Rhistory
23 B
Aug
R

# Creating a Notebook instead of the classical script

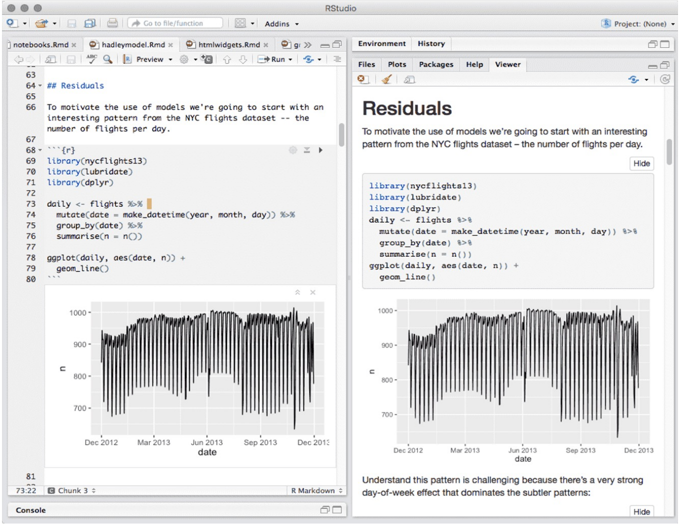
- You can create a new notebook in RStudio with the menu commandFile → *NewFile* → *RNotebook.*
- Notebook chunks can be inserted quickly using the keyboard shortcut Ctrl + Alt + I (macOS: Cmd + Option + I), or via the Insert menu in the editor toolbar.
- Between chunks, you can comment all your results.



~~~
chunk or by placing your cursor inside it and pressing
*Cmd+Shift+Enter*.

```{r}
plot(cars)
~~~

Add a new chunk by clicking the *Insert Chunk* button on the
```
```

R Notebook



~~~
## Residuals

To motivate the use of models we're going to start with an
interesting pattern from the NYC flights dataset -- the
number of flights per day.

```{r}
library(nycflights13)
library(lubridate)
library(dplyr)

daily <- flights %>%
  mutate(date = make_datetime(year, month, day)) %>%
  group_by(date) %>%
  summarise(n = n())

ggplot(daily, aes(date, n)) +
  geom_line()
```
~~~

Residuals

To motivate the use of models we're going to start with an interesting pattern from the NYC flights dataset – the number of flights per day.

```
library(nycflights13)
library(lubridate)
library(dplyr)
daily <- flights %>%
  mutate(date = make_datetime(year, month, day)) %>%
  group_by(date) %>%
  summarise(n = n())
ggplot(daily, aes(date, n)) +
  geom_line()
```

Understand this pattern is challenging because there's a very strong day-of-week effect that dominates the subtler patterns:

A demo before practice. . .