PS11: introduction to R

HPC School

Aurélien Ginolhac (aurelien.ginolhac@uni.lu) - Life Sciences Research Unit

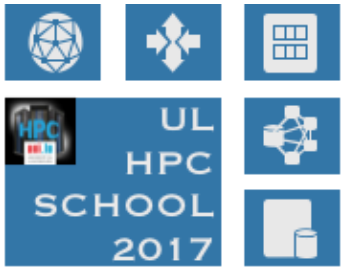
2019-06-21

Objectives

You will learn to:

- quick intro to R
- focus on the tidyverse dialect
- explore `ggplot2` and `dplyr` on *dataSaurus* (beginner only)
- summarise a dataset using different packages and benchmark them
- use R on the clusters
- perform single node parallelisation on `iris`

What is R?



is shorthand for "GNU R":

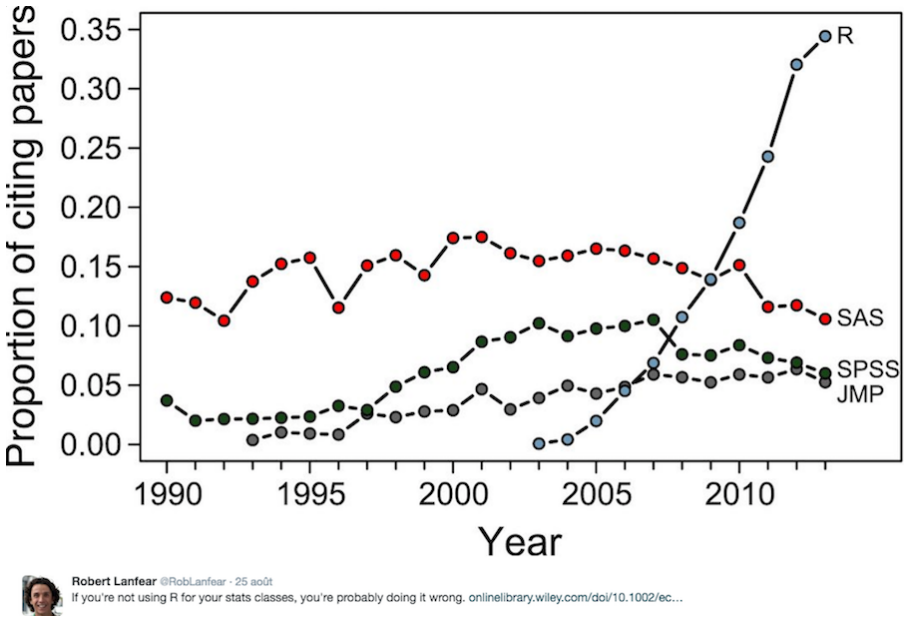
- An **interactive** programming language derived from **S** (**J. Chambers**, Bell Lab, 1976)
- Appeared in 1993, created by **R. Ihaka** and **R. Gentleman**, University of Auckland, NZ
- Focus on data analysis and plotting
- **R** is also shorthand for the ecosystem around this language
  - Book authors
  - Package developers
  - Ordinary useRs

Learning to use **R** will make you **more efficient** and **facilitate the use** of advanced data analysis tools

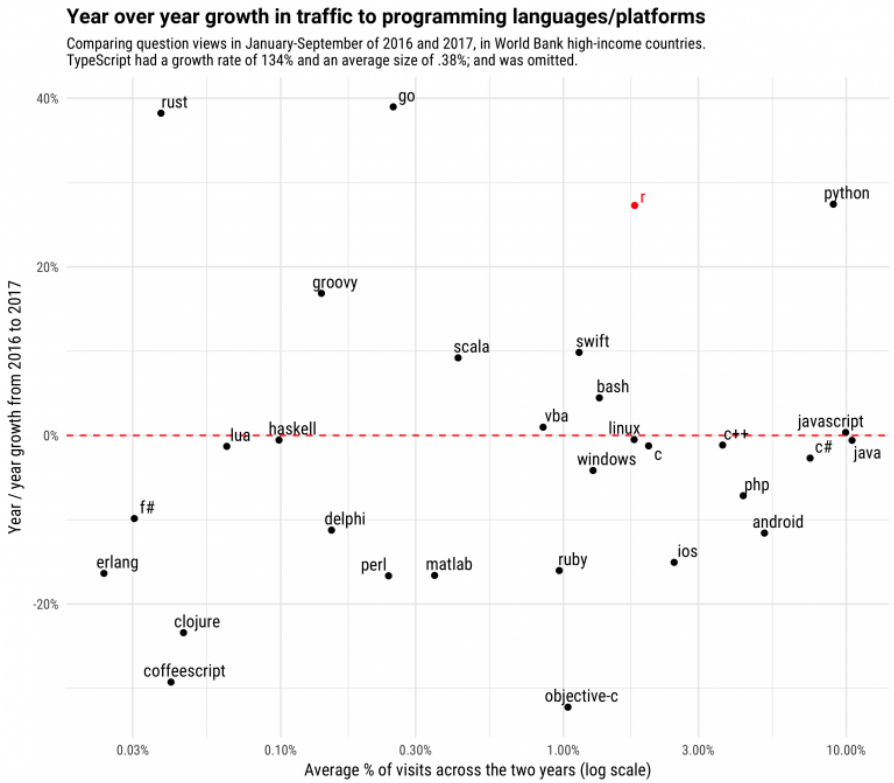
Why use R?

- It's *free!* and **open-source**
- easy to install / maintain
- multi-platform (Windows, macOS, GNU/Linux)
- can process big files and analyse huge amounts of data (db tools)
- integrated data visualization tools, *even dynamic* `shiny`
- fast, and even faster with *C++* integration via Rcpp.
- easy to get help
  - huge R community in the web
  - stackoverflow with a lot of tags like **r**, **ggplot2** etc.
  - rbloggers

Constant trend



Source: Touchon & McCoy. *Ecosphere*. 2016



Source: D. Robinson, StackOverflow blog

Packages

+16,000 in Feb 2019

CRAN

reliable: package is checked during submission process

MRAN for Windows users

bioconductor

dedicated to biology. status

typical install:

```
#
install.packages("BiocManager")
BiocManager::install("limma")
```

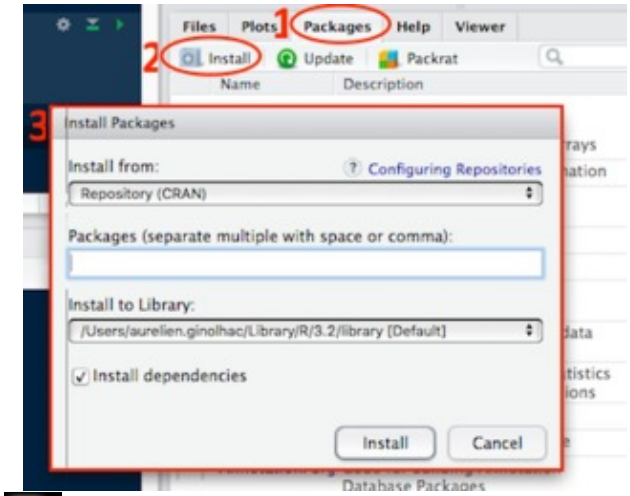
GitHub

easy install thanks to remotes.

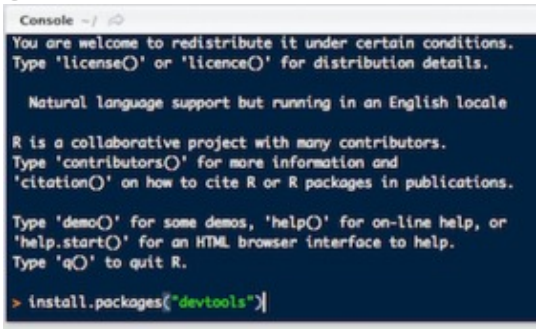
```
#
install.packages("remotes")
remotes::install_github("tidyverse/readr")
```

could be a security issue

CRAN install from Rstudio



github install from Rstudio' console



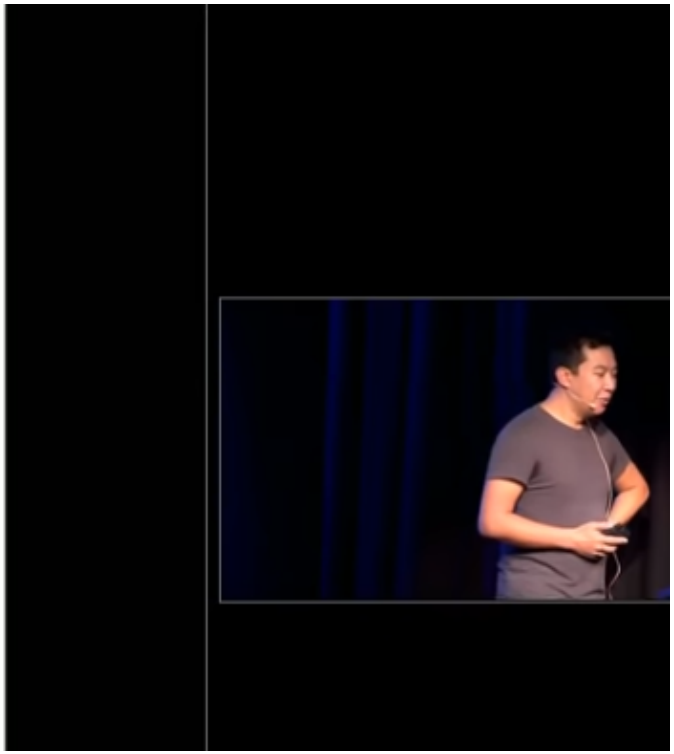
more in the article from David Smith

R ambiguity

Roger D. Peng

"The ambiguity [of the S language] is real and goes to a key objective: we wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important."

John Chambers, "Stages in the Evolution of S"



source: Teaching R to New Users: From tapply to Tidyverse by Roger D. Peng

R is hard to learn

R base is complex, has a long history and many contributors

Why R is hard to learn

- Unhelpful help `?print`
- generic methods `print.data.frame`
- too many commands `colnames, names`
- inconsistent names `read.csv, load, readRDS`
- unstrict syntax, was designed for interactive usage
- too many ways to select variables `df$x, df$"x", df[,"x"], df[[1]]`
- […] see r4stats' post for the full list
- the tidyverse curse

source: Robert A. Muenchen' blog

> Navigating the balance between base R and the tidyverse is a challenge to learn
>
> — *Robert A. Muenchen*

Tidyverse

creator

We think the **tidyverse** is better, especially for beginners. It is

- recent (both an issue and an advantage)
- allows doing powerful things quickly
- unified
- consistent, one way to do things
- give strength to learn base R
- criticisms exist

Hadley Wickham

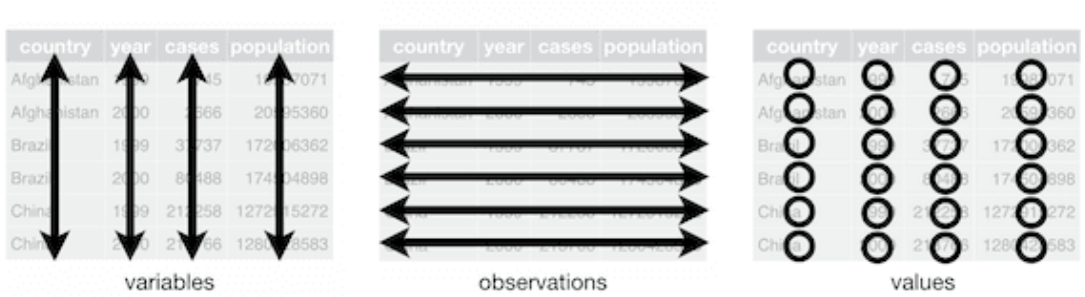
Hadley, Chief Scientist at **Rstudio**

- coined the *tidyverse* at userR meeting in 2016
- developed and maintains most of the core *tidyverse* packages

Tidy data

Definition

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.



Tidy dataset are all alike; every messy dataset is messy in its own way

— Hadley Wickham

16,777
indexed packages

2,419,088
indexed functions

Most downloaded packages

Name	Direct downloads	Indirect downloads	Total
1. pkgconfig	5,644	5,446	11,090
2. ggplot2	4,696	9,714	14,410
3. data.table	4,668	5,579	10,247
4. tidyverse	4,595	183	4,778
5. openssl	4,437	5,110	9,547
6. readxl	4,258	3,255	7,513
7. dplyr	4,197	8,338	12,535
8. R6	4,021	6,826	10,847
9. rlang	3,882	16,420	20,302
10. devtools	3,637	906	4,543

Next >

Most active maintainers

Name	Direct downloads	Indirect downloads	Total
1. Hadley Wickham	43,481	106,471	149,952
2. Gabor Csardi	26,253	81,436	107,689
3. Yihui Xie	15,406	58,680	74,086
4. Kirill Müller	14,559	53,629	68,188
5. Joe Cheng	11,798	46,801	58,599
6. Jeroen Ooms	16,008	36,353	52,361
7. Dirk Eddelbuettel	13,004	36,536	49,540
8. Lionel Henry	8,528	31,887	40,415
9. Jim Hester	6,561	25,202	31,763
10. Winston Chang	10,879	17,111	27,990

Next >

source: rdocumentation (2019-02-19)

16,777
indexed packages

2,419,088
indexed functions

Most downloaded packages

Name	Direct downloads	Indirect downloads	Total
1. pkgconfig	5,644	5,446	11,090
2. ggplot2	4,696	9,714	14,410
3. data.table	4,668	5,579	10,247
4. tidyverse	4,595	183	4,778
5. openssl	4,437	5,110	9,547
6. readxl	4,258	3,255	7,513
7. dplyr	4,197	8,338	12,535
8. R6	4,021	6,826	10,847
9. rlang	3,882	16,420	20,302
10. devtools	3,637	906	4,543

Next >

Most active maintainers

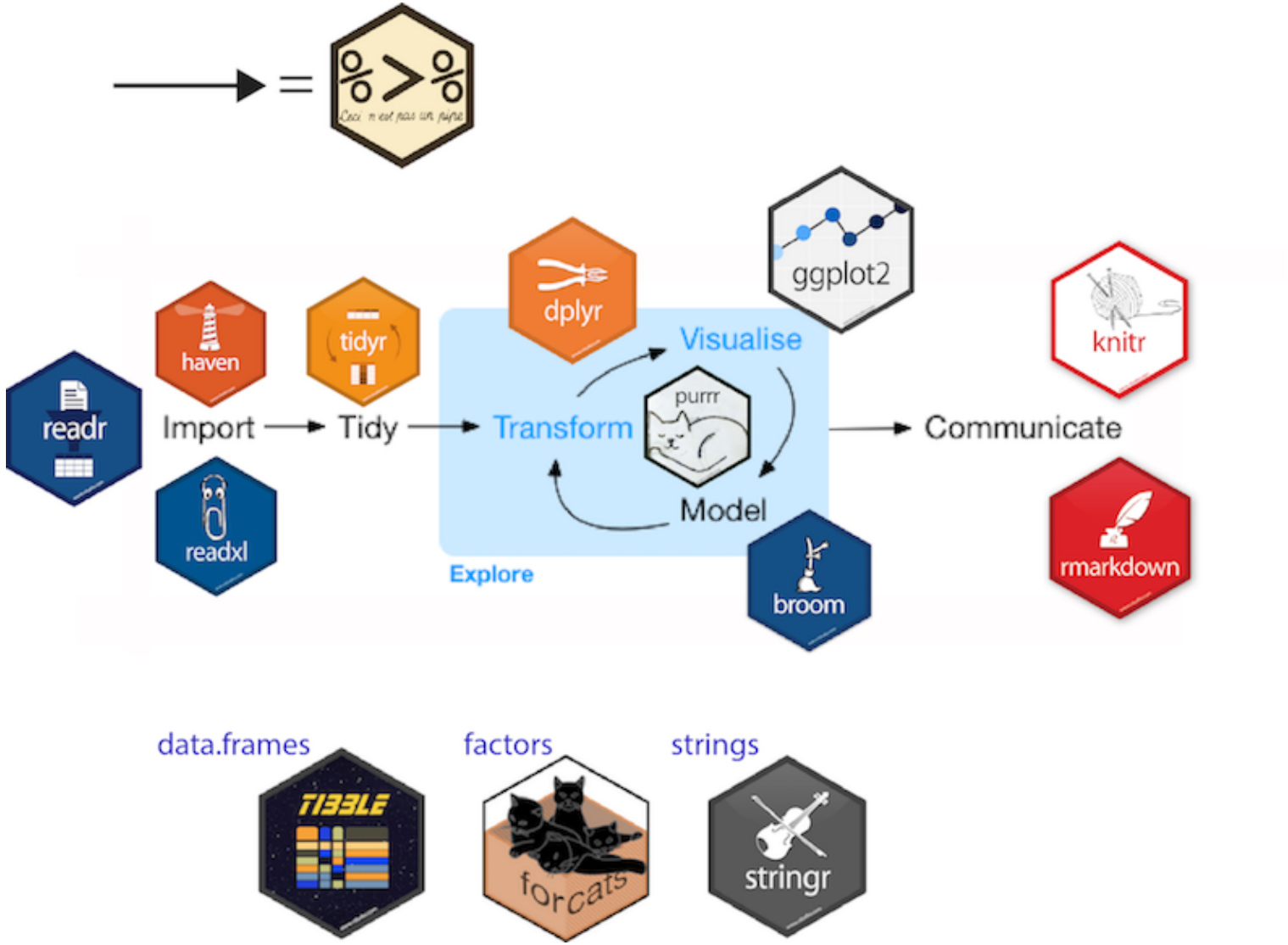
Name	Direct downloads	Indirect downloads	Total
1. Hadley Wickham	43,481	106,471	149,952
2. Gabor Csardi	26,253	81,436	107,689
3. Yihui Xie	15,406	58,680	74,086
4. Kirill Müller	14,559	53,629	68,188
5. Joe Cheng	11,798	46,801	58,599
6. Jeroen Ooms	16,008	36,353	52,361
7. Dirk Eddelbuettel	13,004	36,536	49,540
8. Lionel Henry	8,528	31,887	40,415
9. Jim Hester	6,561	25,202	31,763
10. Winston Chang	10,879	17,111	27,990

Next >

source: rdocumentation (2019-02-19)

Tidyverse

packages in processes



Tidyverse components

core / extended

Core

- `ggplot2`, for data visualization
- `dplyr`, for data manipulation
- `tidyr`, for data tidying
- `readr`, for data import
- `purrr`, for functional programming
- `tibble`, for tibbles, a modern re-imagining of data frames
- `stringr`, for strings
- `forcats`, for factors

source: http://tidyverse.tidyverse.org/. H.Wickham

Extended

- Modelling
  - `modelr`, for modelling within a pipeline
  - `broom`, for models -> tidy data
- Programming
  - `rlang`, low-level API
  - `glue`, alternative to paste

Reproducibility with RMarkdown reports

Why using `rmarkdown` ?

- write detailed reports
- ensure reproducibility
- keep track of your analyses
- comment/describe each step of your analysis
- export a single (Rmd) document to various formats (Pdf, Html…)
- text file that can be managed by a version control system (like git)

Including R code

Rmarkdown document

from the Rmarkdown cheatsheet

Rmarkdown

- extends markdown
- place **R code** in **chunks**
- **chunks** will be **evaluated**
- can also handle bash; python; css; …

Knitr

- extracts R chunks
- interprets them
- formats results as markdown
- reintegrates them into the main document (md)

Pandoc

- pandoc converts markdown to the desired document (Pdf, Html, …)

Rstudio

makes working with R easier

RStudio is an Integrated Development Environment .

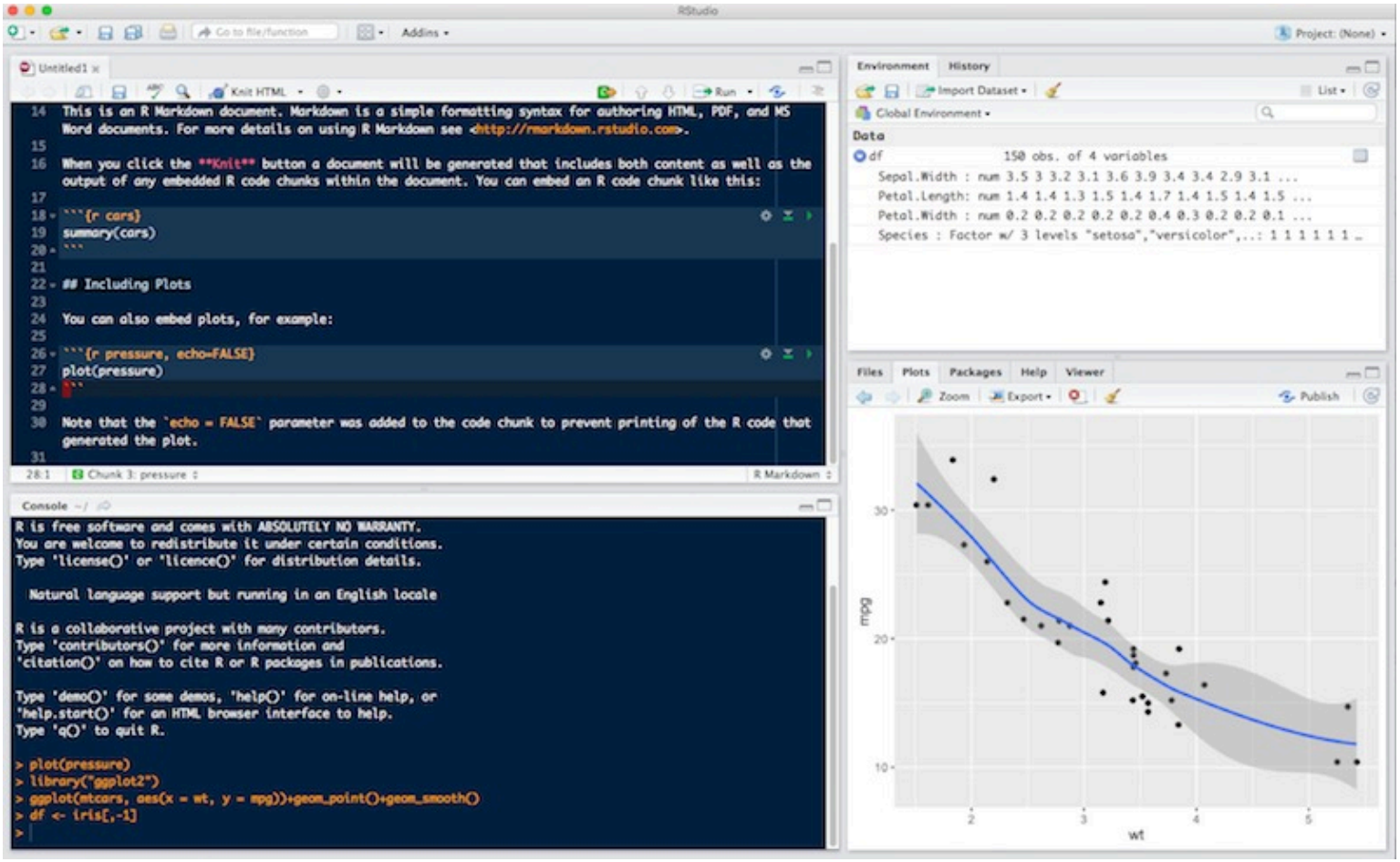
Features

- *Console* to run **R**, with syntax highlighter
- *Editor* to work with scripts
- *Viewer* for data / plots
- *Package management* (including building)
- *Autocompletion* using TAB
- *Cheatsheets*
- *Git* integration for versioning
- *Build* for website / packages
- *Inline* outputs (>= v1.03)
- *Keyboard shortcuts*
- *Notebooks*

Warning

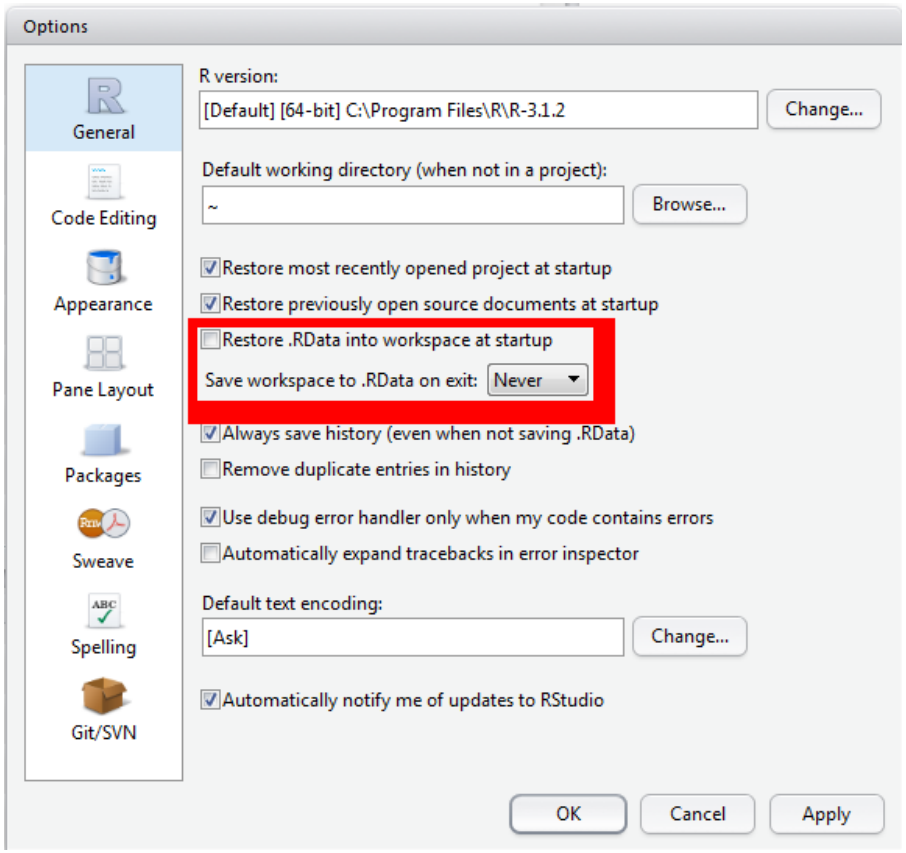
Don't mix up **R** and **RStudio**.
R needs to be installed first.

The 4 panels layout



For reproducibility

options to activate / deactivate



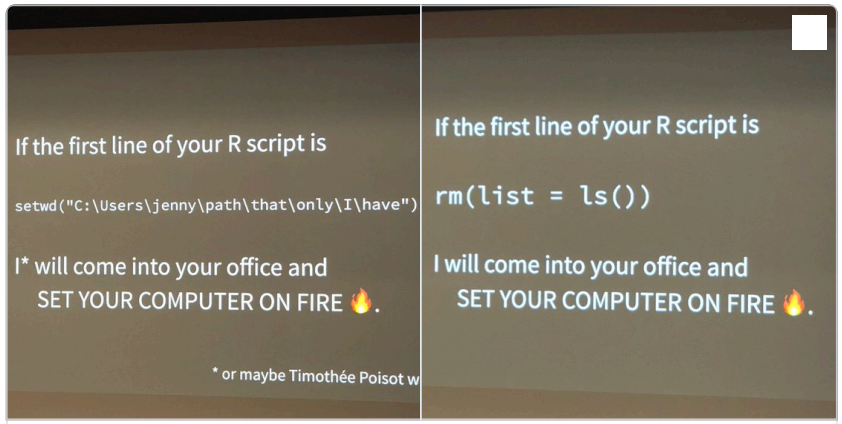
`rm(list = ls())` is not recommended

- does **not** make a fresh R session
  - `library()` calls remain
  - working directory not set!
  - modified functions, evil `==` <- `!=`
- knitting `Rmarkdown` files solves it

source: Jenny Bryan article

Organising files

Use projects



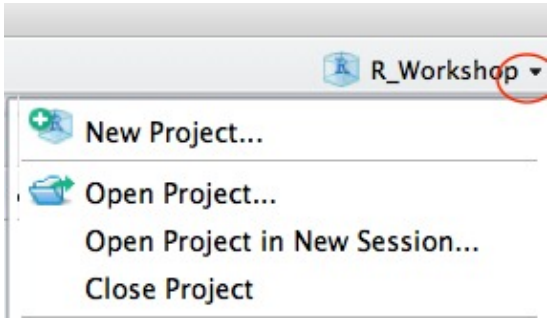
Hadley Wickham
@hadleywickham

The only two things that make @JennyBryan 😤😠🤯.
Instead use projects + here::here() #rstats

993 2:50 AM - Dec 11, 2017

365 people are talking about this

Rstudio projects



Jennifer Bryan's advice

Use `here` package to build paths

- gets the root path of your project:
  - detects Rstudio projects (`.Rproj`)
  - git repository (`.git`)
  - `.here` file

source: Jennifer Bryan's article and test repo

Data types and structures

R base

4 main types

mode( )

Type	Example
numeric	integer (2), double (2.34)
character (strings)	“tidyverse!”
boolean	TRUE / FALSE
complex	2+0i

in the console

```
2L                  [1] 2
typeof(2L)          [1] "integer"
mode(2L)            [1] "numeric"
2.34                [1] 2.34
typeof(2.34)        [1] "double"
"tidyverse!"        [1] "tidyverse!"
TRUE                [1] TRUE
2+0i                [1] 2+0i
```

Special case

```
NA    # not available, missing data
NA_real_
NA_integer_
NA_character_
NA_complex_
NULL # empty
-Inf/Inf # infinite values
NaN # Not a Number
```

Structures

Vectors

`c()` is the function for **concatenate**

```
4                                   [1] 4
c(43, 5.6, 2.90)                    [1] 43.0  5.6  2.9
```

Factors

convert strings to factors,
`levels` is the dictionary

```
factor(c("AA", "BB", "AA", "CC"))   [1] AA BB AA CC
                                    Levels: AA BB CC
```

Lists

very important as it can contain anything

```
list(f = factor(c("AA", "AA")),     $f
     v = c(43, 5.6, 2.90),          [1] AA AA
     s = 4L)                        Levels: AA

                                    $v
                                    [1] 43.0  5.6  2.9

                                    $s
                                    [1] 4
```

Data frames are special lists

`data.frame`

same as list **but** where all objects *must* have the **same** length

Example, 3 elements of same size

```
data.frame(
  f = factor(c("AA", "AA", "BB")),
  v = c(43, 5.6, 2.90),
  s = rep(4, 3))
```

```
   f    v s
1 AA 43.0 4
2 AA  5.6 4
3 BB  2.9 4
```

assignment operator, create object

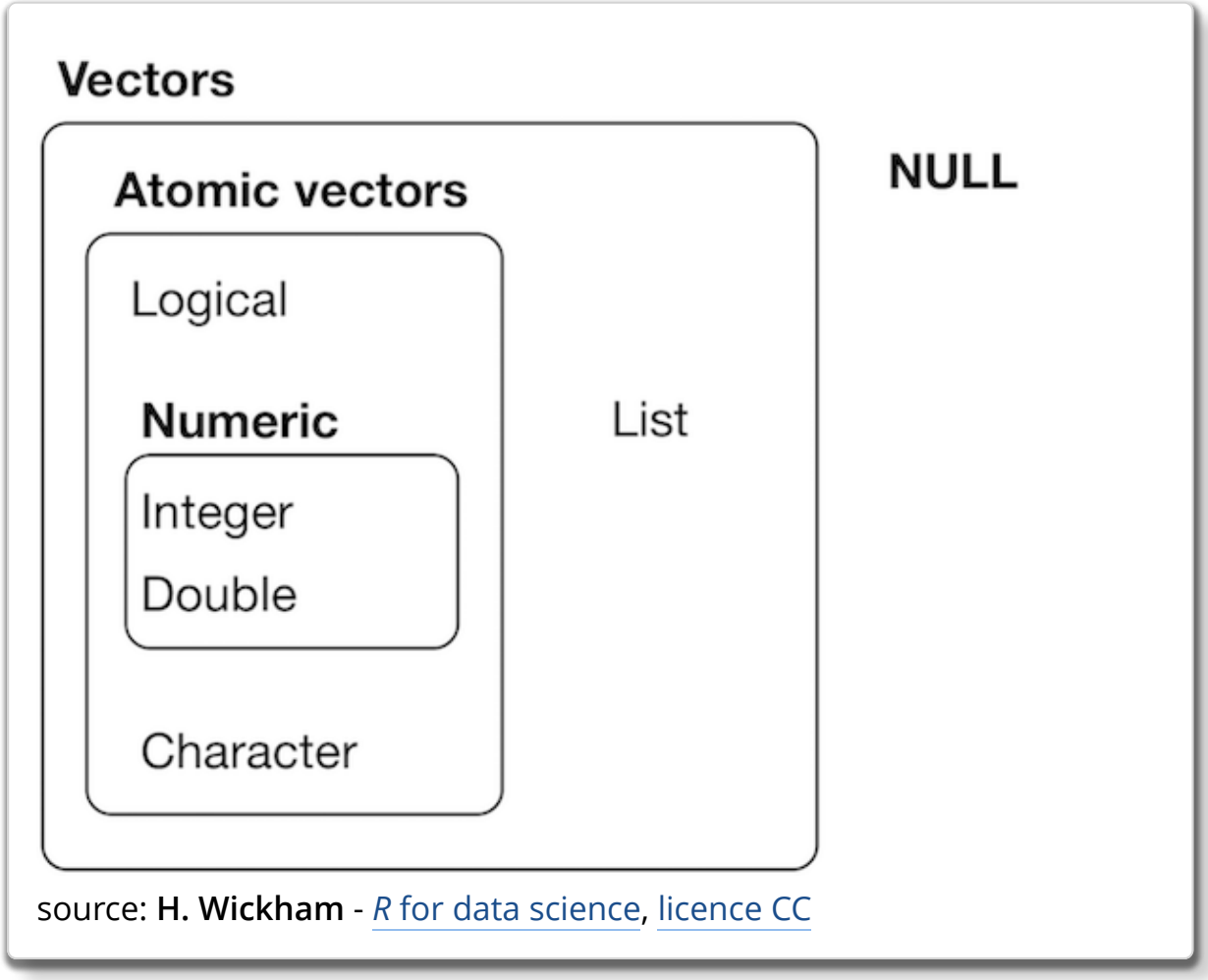
operator is `<-`, associate a *name* to an object

```
my_vec <- c(3, 4, 1:3)
my_vec
[1] 3 4 1 2 3
```

Tip

Rstudio has the built-in shortcut Alt + - for `<-`

hierarchy



source: **H. Wickham** - *R* for data science, licence CC

in console

```
is.vector(c("a", "c"))
[1] TRUE
mode(c("a", "c"))
[1] "character"
is.vector(list(a = 1))
[1] TRUE
is.atomic(list(a = 1))
[1] FALSE
is.data.frame(list(a = 1))
[1] FALSE
```

Vectorized operation

one of the best R feature

```
my_vec <- 10:18
my_vec
[1] 10 11 12 13 14 15 16 17 18
my_vec + 2
[1] 12 13 14 15 16 17 18 19 20
```

warning

- R *recycles* vectors that are too short
- without any warnings:

```
1:10 + c(1, 2)
 [1]  2  4  4  6  6  8  8 10 10 12
```

avoid writing loops, someone else already did (`purrr` , `lapply`)

(still remember not to grow a vector)

```
res <- vector(mode = "numeric", length = length(my_vec))
for (i in seq_along(my_vec)) {
  res[i] <- my_vec[i] + 2
}
res
[1] 12 13 14 15 16 17 18 19 20
```

For loops are fine

growing

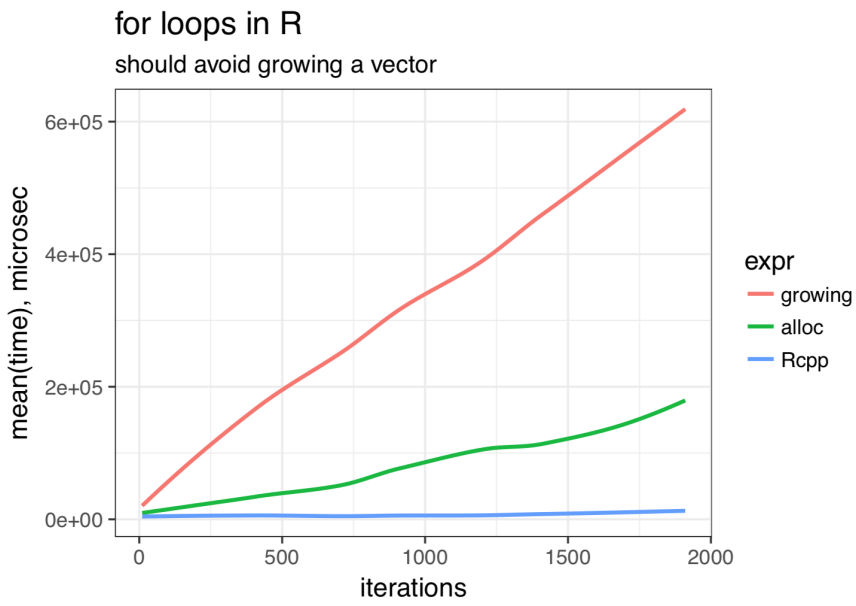
```
for_loop <- function(x) {
  res <- c()
  for (i in seq_len(x)) {
    res[i] <- i
  }
}
```

alloc

```
for_loop <- function(x) {
  res <- vector(mode = "integer",
                length = x)
  for (i in seq_len(x)) {
    res[i] <- i
  }
}
```

Rcpp

```
library(Rcpp)
cppFunction("NumericVector rcpp(int x) {
  NumericVector res(x);
  for (int i=0; i < x; i++) {
    res[i] = i;
  }
}")
```



`purrr::map()` example

type stable

For 3 `cyl` groups on `mtcars`

- fit a linear model (miles per gallon explained by the weight)
  - the equation is then: $mpg = \beta_0 + \beta_1 \times wt,$
  - formula in *R*: `mpg ~ wt`

map the linear model

- `map(YOUR_LIST, YOUR_FUNCTION)`
- `YOUR_LIST` = `spl_mtcars`
- `YOUR_FUNCTION` can be an anonymous function (declared on the fly)

```
spl_mtcars <- group_split(mtcars, cyl)
spl_mtcars %>%
  map(~lm(mpg ~ wt, data = .x)) %>%
  map(summary) %>%
  map_dbl("r.squared") %>%
  str()
 num [1:3] 0.509 0.465 0.423
```

one step per line

- generate 3 tibbles (list)
- run the linear model on each (list)
  - `~` shortcut to foronymous function
- summarised 3 linear models (list)
  - even better with `broom::glance()`
- extract R^2 (doubles)
  - `_dbl()` force a num vector. Error is coercion fails

Acknowledgements

- *HPC* team, especially Valentin Plugaru for setting up the `furrr` runs in `iris`
- Eric Koncina, slides prepared with his iosp R package
- Eric Koncina & Roland Krause for their content in the R workshop
- Joseph Emeras who wrote earlier version of this practical session

- Practical here: https://github.com/ULHPC/tutorials/tree/devel/maths/R
- Slides (html): https://github.com/ULHPC/tutorials/raw/devel/advanced/R/Intro_PS.html
- Slides (pdf): https://github.com/ULHPC/tutorials/raw/devel/advanced/R/Intro_PS.pdf

Practical Session

dataSaurus & furrr