A brief introduction to R with tidyverse

Mauricio Moreira-Soares - May 11th, 2022
University of Oslo, Norway

R

A language from scientists to scientists

- A programming language and a software for statistical computing
- Created by the statisticians Ross Ihaka and Robert Gentleman
- Mainly used for data analysis and development of statistical methods

R

A language from scientists to scientists

- A programming language and a software for statistical computing
- Created by the statisticians Ross Ihaka and Robert Gentleman
- Mainly used for data analysis and development of statistical methods

- Hang on, what about python? 🤔

What is tidy data?

- Simple idea:
  - Every column is a variable
  - Every row is an observation
  - Every cell is a single value

- Any other format is messy!

PatientID	Age	Sex	BP
ous384	20	female	105
ous883	38	male	140
rad200	33	male	130
rik010	33	male	125

What is tidy data?

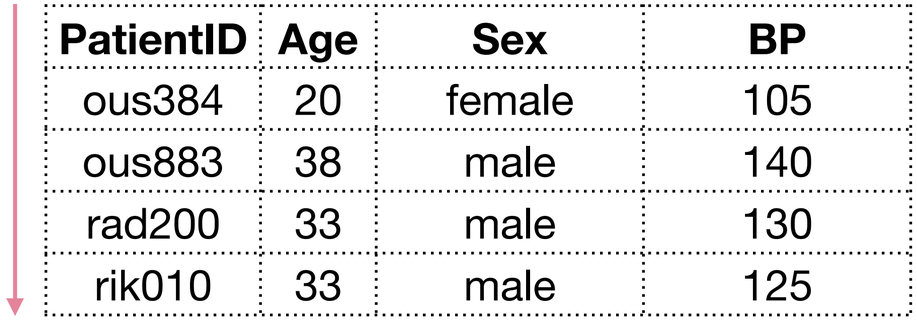
- Simple idea:
  - Every column is a variable
  - Every row is an observation
  - Every cell is a single value

PatientID	Age	Sex	BP
ous384	20	female	105
ous883	38	male	140
rad200	33	male	130
rik010	33	male	125

- Any other format is messy!

What is tidy data?

- Simple idea:
  - Every column is a variable
  - Every row is an observation
  - Every cell is a single value



PatientID	Age	Sex	BP
ous384	20	female	105
ous883	38	male	140
rad200	33	male	130
rik010	33	male	125

- Any other format is messy!

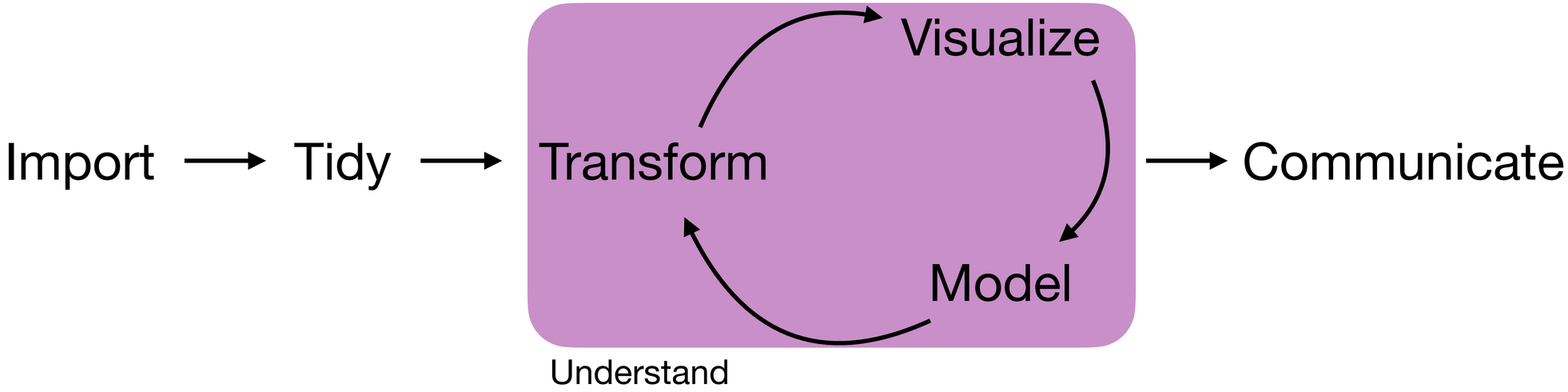
What is tidy data?

- Simple idea:
  - Every column is a variable
  - Every row is an observation
  - Every cell is a single value

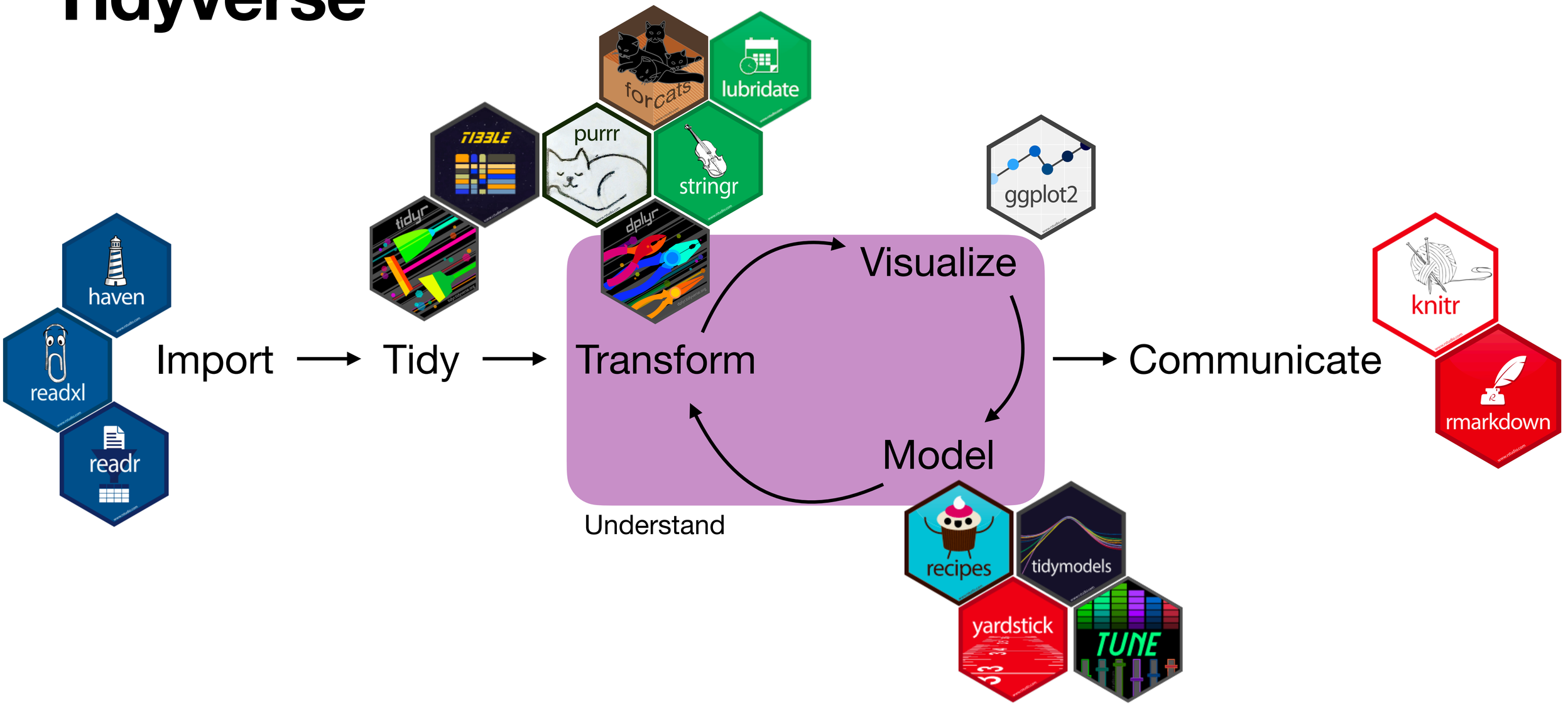
PatientID	Age	Sex	BP
ous384	20	female	105
ous883	38	male	140
rad200	33	male	130
rik010	33	male	125

- Any other format is messy!

Data analysis flow



Tidyverse


haven
readxl
readr
Import
tidyr
tibble
purrr
forcats
lubridate
stringr
dplyr
Tidy
Transform
Visualize
ggplot2
Model
Understand
recipes
tidymodels
yardstick
TUNE
Communicate
knitr
rmarkdown

Tidyverse

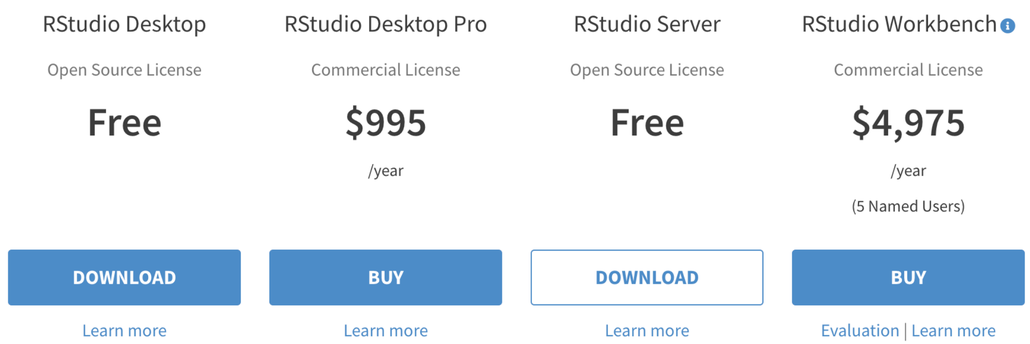
The philosophy behind it

- To facilitate the conversation between a human and a computer about data
- Sits at the heart of every data science project to solve repeated tasks
- Tidyverse will not solve all your problems and it is not the aim
  - It is expected to combine it with specific domain-knowledge packages
  - For instance, high throughput data requires **Bioconductor**

R Studio

Download & Installation

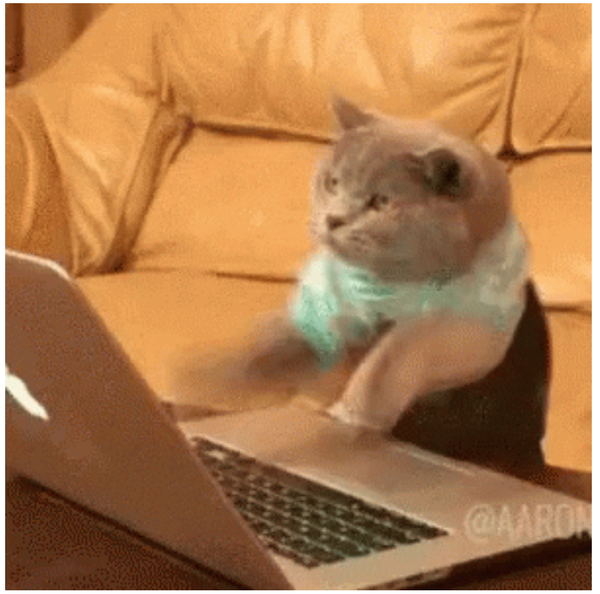
- First you have to install R base from www.r-project.org or Software manager
- A Graphical User Interface (GUI) for R
  - Free, open source and multi-platform (Windows, Linux and MacOSX)
- Access tinyurl.com/bebrstudio and click on *Download* on the Free version



R studio

Download & Installation

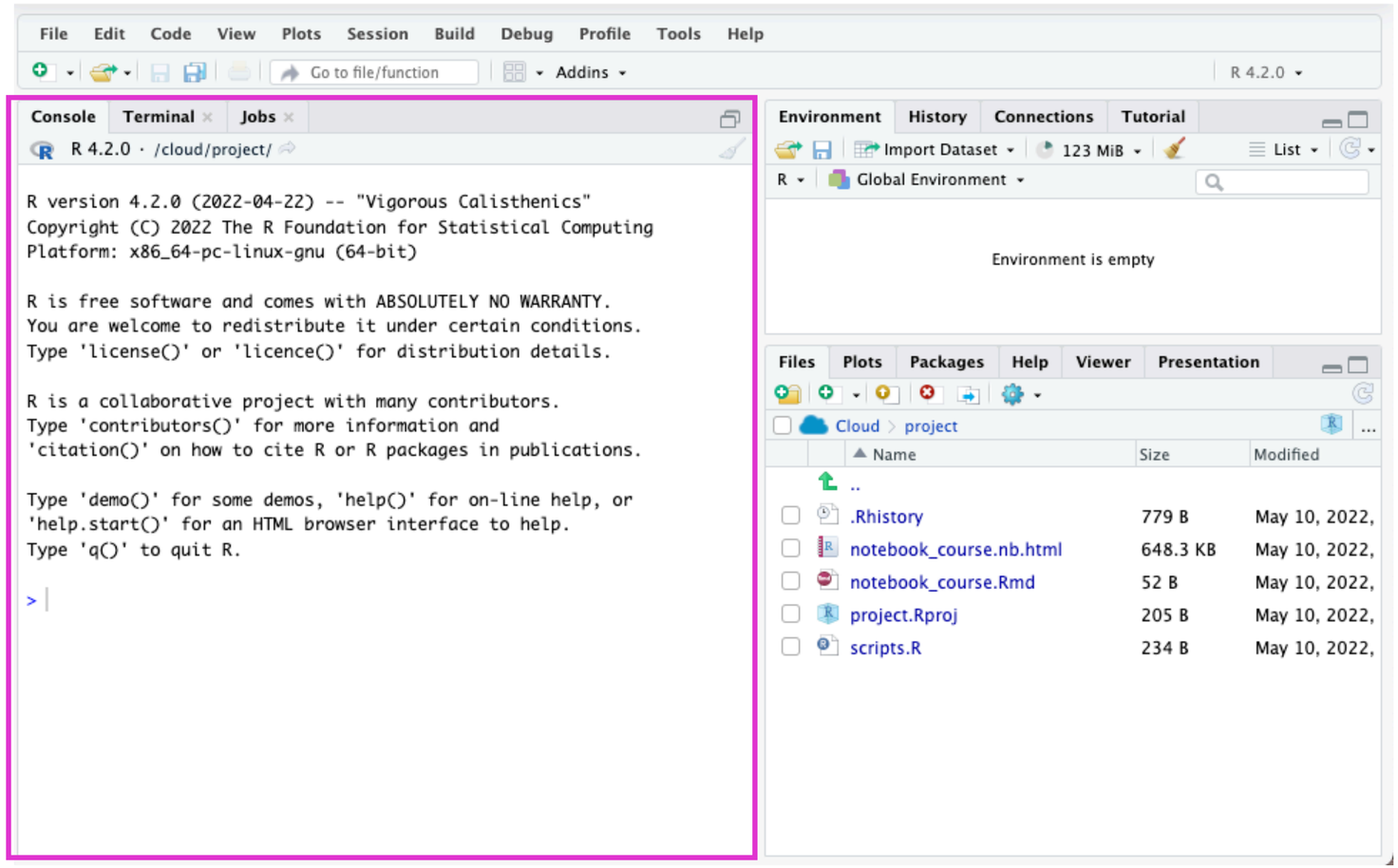
- Or sign up for RStudio cloud and try it online!
  https://login.rstudio.cloud/



Basic concepts

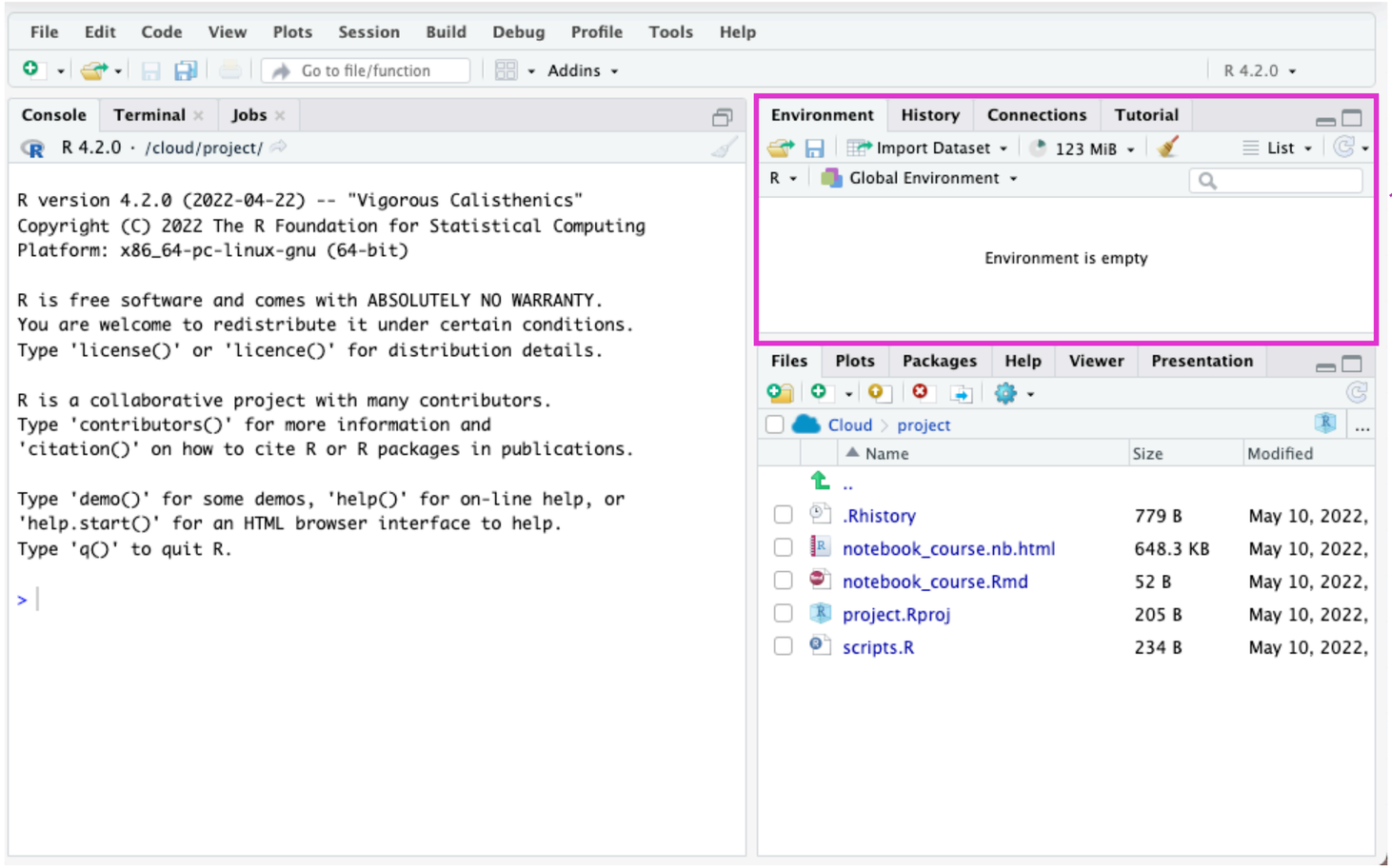
RStudio interface

The console



Basic concepts

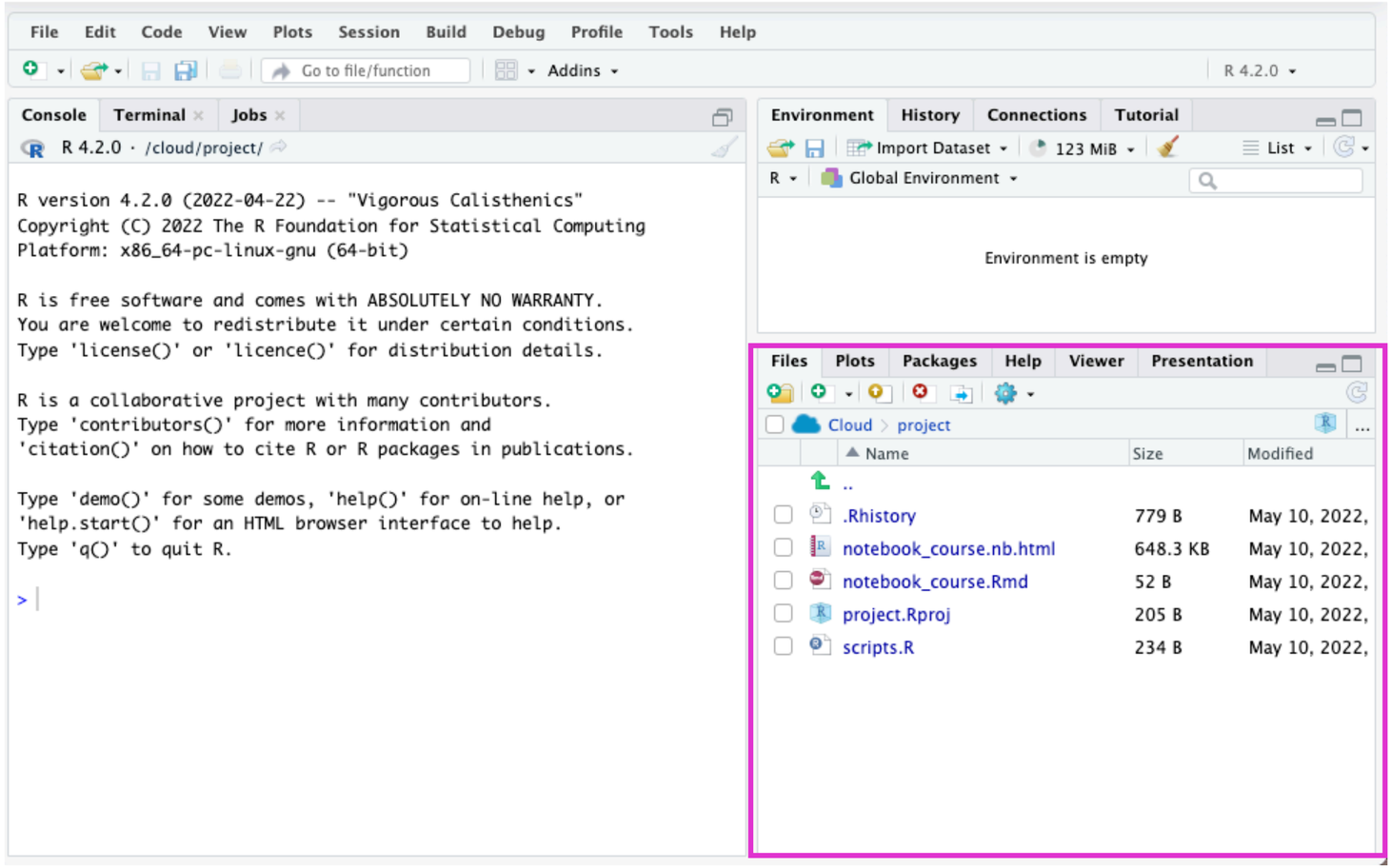
RStudio interface



The environment

Basic concepts

RStudio interface



Files, plots and help

Basic concepts

Variables and operations

- Defining a new variable

```
x <- 1.2
```

```
y <- “hello world”
```

- Arithmetics (+, -, *, /, ^,**)

```
1 + 2*(3/9)
```

R can be used as a fancy calculator

- Variable types:
  - Numeric, integer, character, matrix …

```
class(x)
```

```
class(y)
```

- Vectors

```
c(1,2,3,4,5)
```

Basic concepts

Variables and operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

if `1==1` then

`10>12`

`(1==1) & (10>12)`

`(1==1) & !(10>12)`

`(1==1) | (10>12)`

Basic concepts

Variables and operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

`1==1` ⟶ **TRUE**

if `10>12` then

`(1==1) & (10>12)`

`(1==1) & !(10>12)`

`(1==1) | (10>12)`

Basic concepts

Variables and operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

`1==1` ⟶ **TRUE**

`10>12` ⟶ **FALSE**

if `(1==1) & (10>12)` then

`(1==1) & !(10>12)`

`(1==1) | (10>12)`

Basic concepts

Variables and operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

`1==1` ⟶ **TRUE**

`10>12` ⟶ **FALSE**

`(1==1) & (10>12)` ⟶ **FALSE**

if `(1==1) & !(10>12)` then

`(1==1) | (10>12)`

Basic concepts

Logical operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

```
1==1                    ⟶  TRUE

10>12                   ⟶  FALSE

(1==1) & (10>12)        ⟶  FALSE

(1==1) & !(10>12)       ⟶  TRUE
```

if `(1==1) | (10>12)` then

Basic concepts

Logical operations

- Logical operators

`==`	Equal to
`!=`	Different of
`<`	Less than
`<=`	Less or equal than
`>`	Greater than
`>=`	Greater of equal than
`A|B`	A or B
`A&B`	A and B
`!A`	not A

The result of logical operations are TRUE or FALSE

```
1==1                    ⟶  TRUE

10>12                   ⟶  FALSE

(1==1) & (10>12)        ⟶  FALSE

(1==1) & !(10>12)       ⟶  TRUE

(1==1) | (10>12)        ⟶  TRUE
```

Basic concepts

Loops

- Loops are, in general, completely avoidable with tidyverse
- This reduces the algorithmic and implementation complexity

For loop

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

While loop

```
i <- 1
while (i<=10){
   print(i)
   i <- i + 1
}
```

Basic concepts

Packages

- Packages are available in a central public repository:
  - The Comprehensive R Archive Network (CRAN)
- Simple installation

```
install.packages("tidyverse")
```

- Packages are collections of functions that are loaded into your work environment

```
library(dplyr)
```

Basic concepts

Asking for help

- Programmers have one secret and one special skill

Basic concepts

Asking for help

- Programmers have one secret and one special skill
  - We cannot remember every single command so we often have to search

Basic concepts

Asking for help

- Programmers have one secret and one special skill
  - We cannot remember every single command so we often have to search
  - Our special skill is to be really good at searching and reading documentations

Basic concepts

Asking for help

- Programmers have one secret and one special skill
  - We cannot remember every single command so we often have to search
  - Our special skill is to be really good at searching and reading documentations
- Learning how to search is key

Basic concepts

Asking for help

- Programmers have one secret and one special skill
  - We cannot remember every single command so we often have to search
  - Our special skill is to be really good at searching and reading documentations
- Learning how to search is key
- But gladly you can also ask for help direct to R

```
?mean  ?sd
```

- The autocomplete function **<tab>** is also handy

Importing data

support for several data formats

- **SAS:** `read_sas()`
- **SAV:** `read_sav()`
- **SPSS:** `read_dta()`
- **Excel:** `read_excel()`
- **GraphPad Prism:** `pzfx_tables()*`

* the **pzfx** library is not part of tidyverse

Importing data

- We will work with simulated clinical data from one of my projects
- Comma Separated Values (CSV) file

```
df <- read_csv(“https://tinyurl.com/clindata2022")
```

- For transforming string variables into factors automatically use

```
stringAsFactors = TRUE
```

- … not recommended!

Data frames

- You can access variables by using the $ symbol and column name

```
df$study_id
```

- But also by column number (which is more inconvenient)

```
df[, 1]
```

- Print the top 6 observations

```
head(df)
```

The pipe operator

%>%

- “Pipes” the result of the operation on the left to the next operation on the right

```
mean(c(1,2,3))
```

```
c(1,2,3) %>% mean
```

- Aims to simplify the coding style and reduce mistakes

$f(g(h(a))) \longrightarrow$ `a %>% f %>% g %>% h`

Useful functions

- Center: `mean(), median()`
- Spread: `sd(), IQR(), mad()`
- Range: `min(), max(), quantile()`
- Position: `first(), last(), nth()`
- Count: `n(), n_distinct()`
- Logical: `any(), all()`

Data wrangling

with dplyr

- How to select certain variables according to a condition?

`select()` →

```
df %>% select(study_id, hn1_dv_age_cons)
df %>% select(starts_with("hn1"))
df %>% select(contains("c30"))
df %>% select(ends_with("sex"))
df %>% select(num_range("X", 1:5))
df %>% select(matches(RegEx))
```

Data wrangling

with dplyr

- How to filter observations?

`filter()` →

```
df %>% filter(hn1_dv_age <= 20)
```

```
df %>% filter(hn1_alcohol == “1 - Yes”)
```

Data wrangling

with dplyr

- How to create a new variable based on existing ones?

`mutate()` → `df %>% mutate(half_age = hn1_dv_age_cons/2)`

Data wrangling

with dplyr

- Compute summary statistics - mean, mode, sd, median, IQR, etc

`summarise()` → `df %>% summarise(mean(hn1_dv_age_cons))`

```
df %>%
        group_by(hn2_surgery) %>%
        summarise(mean(hn1_dv_age_cons))
```

Data wrangling

with dplyr

- Order your dataset by a specific variable

`arrange()` → `df %>% arrange(hnl_dv_age_cons)`

Data visualization

R base

- Histograms

```
hist(df$hn1_dv_age_cons)
```

- Scatter plot

```
hist(df$hn1_dv_age_cons, df$hn4_dv_c30_summary)
```

- Pairs

```
pairs(df %>% select(hn2_dv_c30_summary, hn3_dv_c30_summary))
```

Data visualization

R base

- Boxplot

```
boxplot(hn1_dv_age_cons ~ hn1_na8_cb_sex, data=df)
```

- Strip chart

```
stripchart(hn1_dv_age_cons ~ hn2_radio, data=df, vertical=TRUE)
```

Data visualisation

ggplot2

- Every plot with ggplot2 starts with

  `ggplot(data)`

- Then we add components to compose the plot (literally with a + sign)

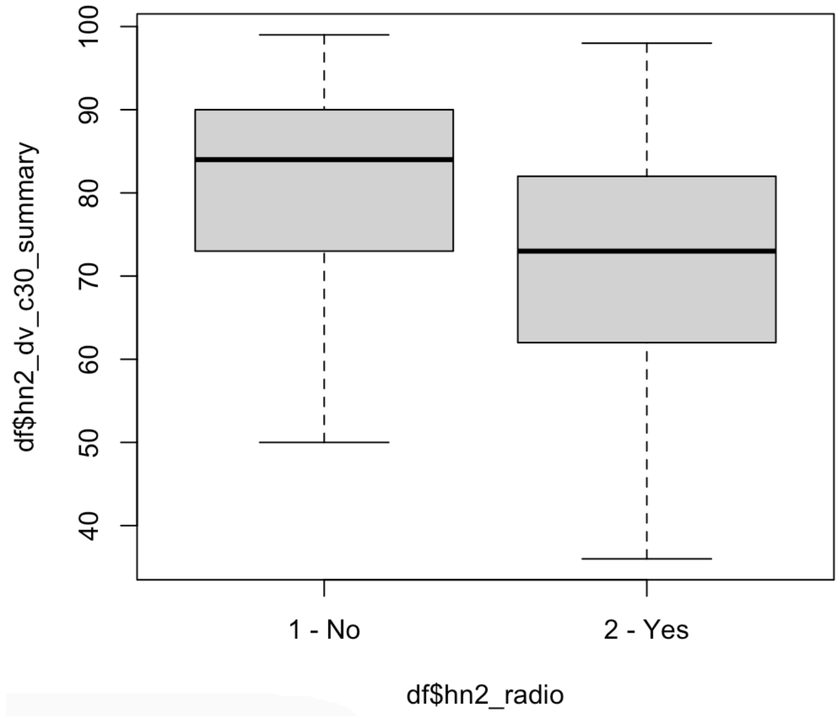
  `ggplot(data=df) + geom_boxplot(aes(x=hn2_radio, y=hn2_dv_c30_summary))`

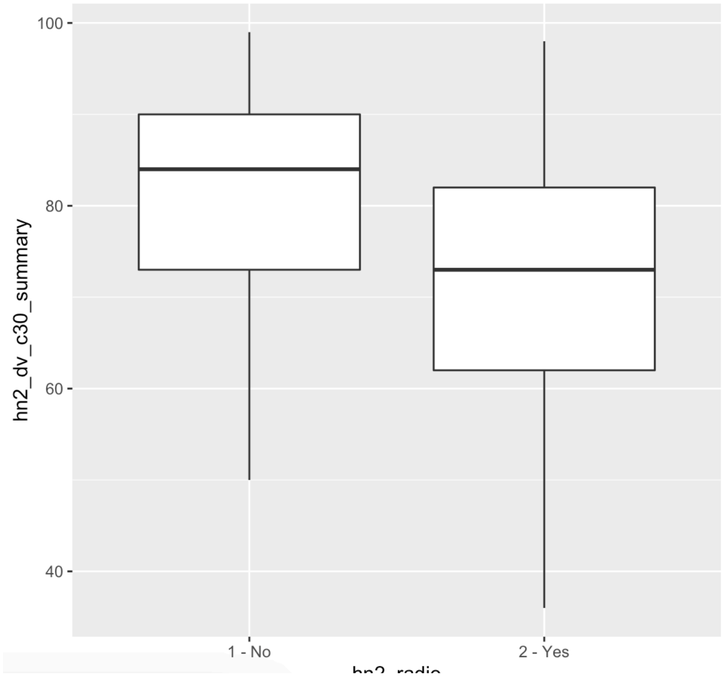
- A common mistake is to use %>% instead of +

```
> ggplot(data=df) %>% geom_boxplot(aes(x=hn2_radio, y=hn2_dv_c30_summary))
Error: `mapping` must be created by `aes()`
Did you use %>% instead of +?
```

- But the error message is helpful 😅

R base vs ggplot2





Statistical analysis

Common tests

- T-student

```
t.test(df$hn2_dv_c30_summary)
```

- Kruskal-wallis

```
kruskal.test(hn1_dv_age_cons ~ hn2_radio, data = df)
```

- Pairwise

```
pairwise.wilcox.test(df$hn2_dv_c30_summary,
                     df$hn1_dv_TNM_stage_best,
                       p.adjust.method = "BH")
```

- Linear model

```
lm(hn2_dv_c30_summary ~ hn1_dv_age_cons, data=df)
```

Communicating

RMarkdown and knitr

- You can perform data analysis, write reports, papers and build dashboards
- Export to several different formats (HTML, pdf, LaTeX)
- Check the RMarkdown gallery for inspiration
  - https://rmarkdown.rstudio.com/gallery.html

Resources

- Official tidyverse website: https://www.tidyverse.org/
- Community R studio: http://community.rstudio.com/
- Stack overflow: https://stackoverflow.com/
- Twitter #rstats: https://twitter.com/search?q=%23rstats&src=typd
- University of Oslo slides: https://github.com/ocbe-uio/public-slides/blob/main/Presentations/20210920-IntroR.md
- Several books are available about data analysis with R and tidyverse

Follow me on twitter:

Thank you!

European Union's Horizon 2020 Research and Innovation programme
under the Marie Skłodowska-Curie Actions Grant agreement No 801133