

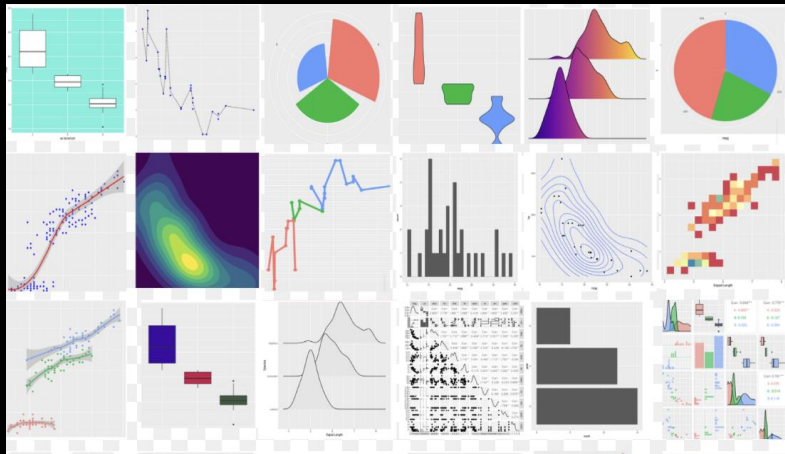
Evomics 2024

R & ggplot2



Outline

- **What is R**
 - Why is it useful?
 - What can be done with R?
 - RStudio
- **Installing packages**
- **General data structures**
- **Dataset**



What is R?

A free software environment
for statistical computing
and graphics

```
R Console

R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

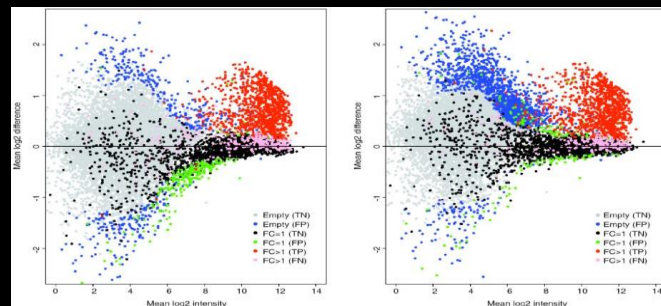
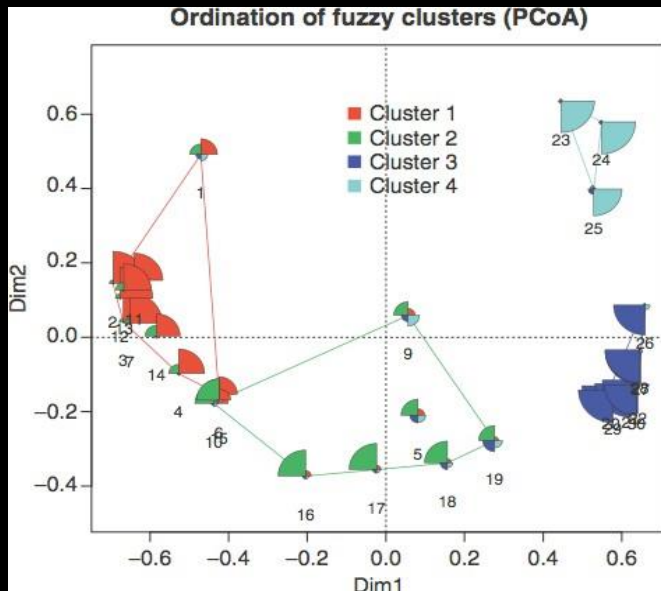
Natural language support but running in an English locale

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 1.79 (8895) x86_64-apple-darwin17.0]

> |
```



Why is R useful?

- **Open source**
- **Data management and manipulation**
 - Importing data in various formats (like text files, excel files, etc.)
 - Manipulating data (subsetting and filtering tables, merging, transposing, etc.)
- Cutting-edge **graphical data visualization**
- Support for rich **statistical simulation and modeling**
- Well established system of **packages and documentation**
- **Active development** and dedicated **community**



Why is R useful?

- Open source

- Data management and manipulation

- Importing data in various formats (like text files, excel files, etc.)
- Manipulating data (subsetting and filtering tables, merging, transposing, etc.)



Stay tuned for our lab session!

- Cutting-edge **graphical data visualization**



David Barnett
(16th Jan)

- Support for rich **statistical simulation and modeling**



Rachel Steward
(today)

- Well established system of **packages and documentation**



Next slides →

- **Active development** and dedicated **community**

Where to learn more about R

- The R Project Homepage: <http://www.r-project.org>
- Quick R Homepage: <http://www.statmethods.net>
- Bioconductor: <http://www.bioconductor.org>
- An Introduction to R (long!): <http://cran.r-project.org/doc/manuals/R-intro.html>
- Guide to downloading and installing R and RStudio: <https://rstudio-education.github.io/hopr/starting.html>
- R Graph gallery (inspiration for R charts) <https://r-graph-gallery.com>
- Google - there are tons of tutorials, guides, demos, packages and more

R for Biologists

● **Bioconductor** <http://bioconductor.org>

- 2,140 packages (05-27-2022):
 - Variant detection: coding changes, PolyPhen database
 - Annotation: pathway analysis, access GO, KEGG, NCBI and many others
 - High-throughput assays: flow cytometry, mass spec
 - Transcription factor binding detection

● **Phylogenetics and Evolution**

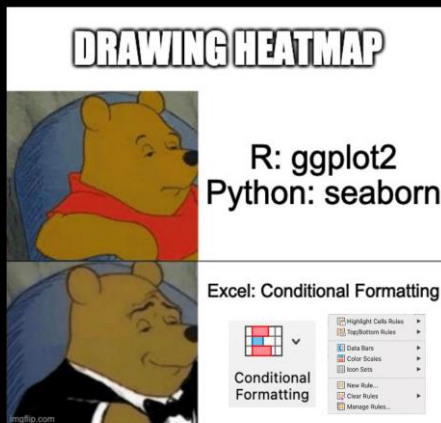
<http://cran.r-project.org/web/views/Phylogenetics.html>

- Ancestral State Reconstruction
- Phylogenetic Inference
- Trait Evolution

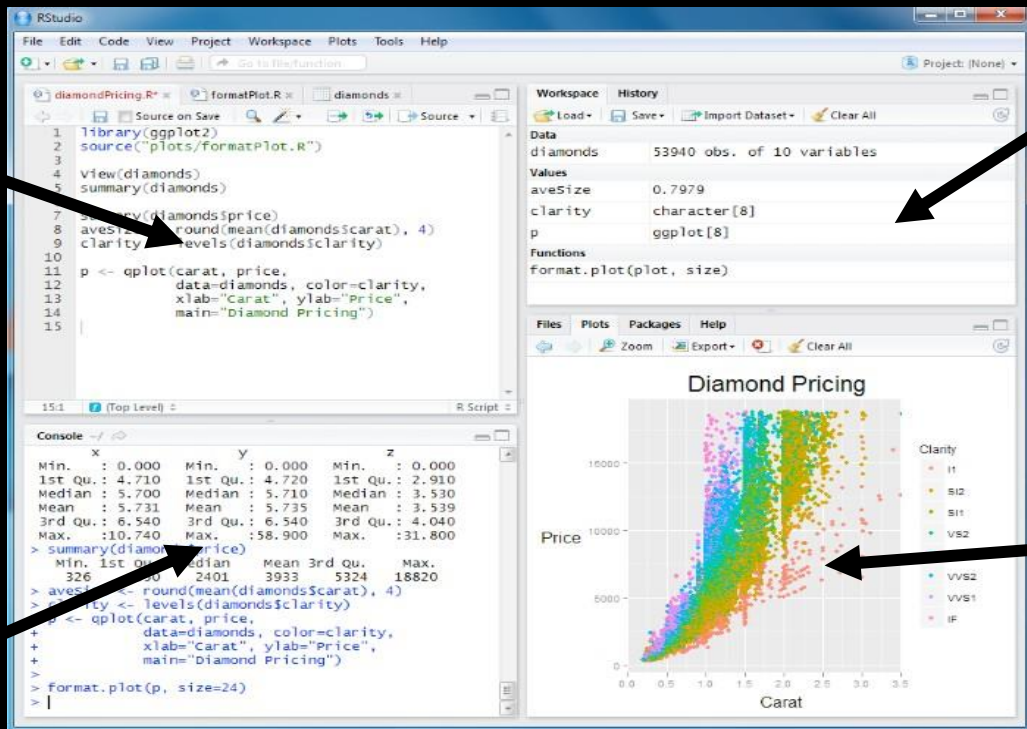
● **Ecology**

<http://cran.r-project.org/web/views/Environmetrics.html>

- Ordination
- Cluster Analysis
- Ecological Theory
- Population Dynamics
- Spatial Data Analysis

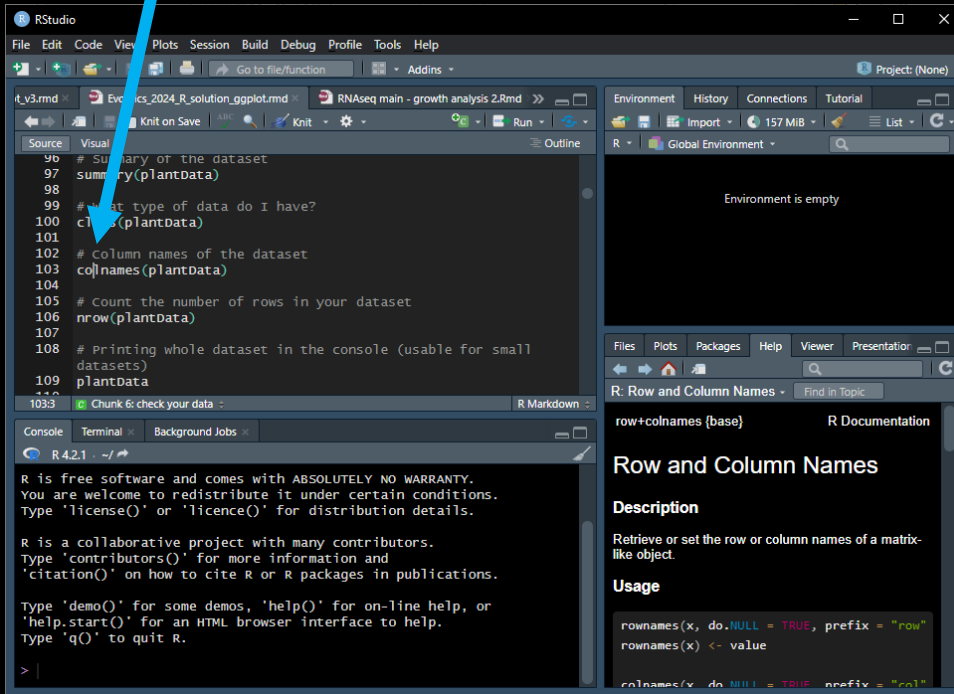


R Studio



Help in R Studio

Press **F1** when the cursor is in the name of the function



The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for summarizing a dataset. The cursor is positioned on the `rownames` function in the line `rownames(plantData)`.
- Environment Panel:** Shows "Global Environment" and "Environment is empty".
- Help Panel:** Displays the documentation for `rownames` from the "R Documentation" package. The title is "Row and Column Names". The description states: "Retrieve or set the row or column names of a matrix-like object." The usage section shows: `rownames(x, do.NULL = TRUE, prefix = "row")` and `rownames(x) <- value`.
- Console:** Shows the R startup message and the prompt `> |`.

A blue arrow points from the text "Press F1" to the cursor in the Source Editor. A blue bracket on the right side of the Help Panel points to the text "The help will open here".

The help
will open
here

rmarkdown :: CHEATSHEET

What is rmarkdown?

- Rmd files** - Develop your code and ideas side-by-side in a single document. Run code as individual chunks or as an entire document.
- Dynamic Documents** - Knit together reports, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS PowerPoint.
- Reproducible Research** - Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

Workflow

1. Open a new **Rmd file** in the RStudio IDE by going to **File > New File > R Markdown**.
2. **Embed code in chunks**. Run code by line, by chunk, or all at once.
3. **Write text** and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
4. **Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. **Save and render** the whole document. Knit periodically to preview your work as you write.
6. **Share your work!**

Embed Code with knitr

CODE CHUNKS

Surround code chunks with ````r` and ````` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after `r`.

```
```{r chunk-label, include=FALSE}
summary(mtcars)
```
```

SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.

```
```{r include=FALSE}
knitr::opts_chunk$set(message = FALSE)
```
```

INLINE CODE

Insert ````r <code>` into text sections. Code is evaluated at render and results appear as text.

*Built with ````r getRversion()```` -> "Built with 4.1.0"



1. New File

2. Embed Code

3. Write Text

4. Set Output Format(s) and Options

5. Save and Render

6. Share

1. New File

2. Embed Code

3. Write Text

4. Set Output Format(s) and Options

5. Save and Render

6. Share

| OPTION | DEFAULT | EFFECTS |
|------------------------|-----------|--|
| echo | TRUE | display code in output document |
| error | FALSE | TRUE (display error messages in doc) |
| eval | TRUE | FALSE (stop render when error occurs) |
| include | TRUE | run code in chunk |
| message | TRUE | include chunk in doc after running |
| warning | TRUE | display code warnings in document |
| results | "markup" | "asis" (pass-through results)
"hide" (don't display results)
"hold" (put all results below all code) |
| fig.align | "default" | "left", "right", or "center" |
| fig.alt | NULL | alt text for a figure |
| fig.cap | NULL | figure caption as a character string |
| fig.path | "figure" | prefix for generating figure file paths |
| fig.width & fig.height | 7 | plot dimensions in inches |
| out.width | | rescales output width, e.g. "75%", "300px" |
| collapse | FALSE | collapse all sources & output into a single block |
| comment | "##" | prefix for each line of results |
| child | NULL | files(s) to knit and then include |
| purrr | TRUE | include or exclude a code chunk when extracting source code with <code>knitr::purrr()</code> |

See more options and defaults by running `str(knitr::opts_chunk$get())`

1. New File

2. Embed Code

3. Write Text

4. Set Output Format(s) and Options

5. Save and Render

6. Share

Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header.

```
title: "My Document"
bibliography: references.bib
link-citations: TRUE
```

- If Zotero is installed locally, your main library will automatically be available.

- Add citations by DOI by searching "from DOI" in the Insert Citation dialog.

INSERT CITATIONS

- Access the **Insert Citations** dialog in the Visual Editor by clicking the **@** symbol in the toolbar or by clicking **Insert > Citation**.
- Add citations with markdown syntax by typing `@cite` or `@cite`.

Insert Tables

Output data frames as tables using `kable(data, caption)`.

```
data <- faithful[1:4, ]
knitr::kable(data,
  caption = "Table with kable")
```

Other table packages include **flextable**, **gt**, and **kableExtra**.

Write with Markdown

The syntax on the left renders as the output on the right.

Plain text.
End a line with two spaces to start a new paragraph.

Also end with a backslash to make a new line.

"italics" and "bold"
superscript²/_{subscript}2-
~ strikethrough ~
escaped: \ \ \ \

endash: ---

Header 1
Header 2

Header 6
unordered list
item 2

ordered list
item 2

link url
[This is a link.](link url)

At the end of the document:
[id]: link url

At the end of the document:
[id]: image.png

verbatim code
multiple lines of verbatim code

block quotes

equation: $e^{\pi i} + 1 = 0$

equation block: $E = mc^2$

horizontal rule

Right | Left | Default | Center
12 12 12 12
123 123 123 123
1 1 1 1

HTML Tagsets
Results (tabset)
Files
Tables
Text

Results
Plots
Tables
Text



Installing packages in R

- **Bioconductor**

<http://bioconductor.org>

- `BiocManager::install("Package Name")`

- **CRAN install** <https://cran.r-project.org/web/packages/>

- `install.packages("Package Name")`

- **GitHub install**

- `devtools::install_github("Package Name")`
- `remotes::install_github("Package Name")`



General data structures

- **Vector** - ordered collection of data

```
vector_1 <- c(2, 3, 4, 10)
```

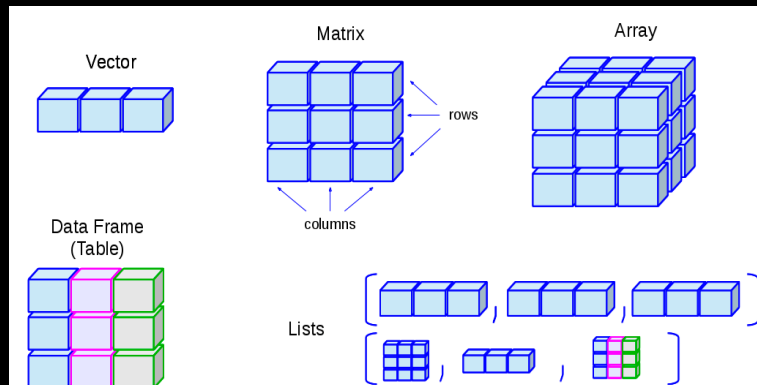
```
vector_2 <- c("potato", "lemonade", "avocado")
```

- **Matrix** - 2D collection of vectors with same data type

- **Array** - multiple dimension collection of vectors

- **Dataframe** - matrix-like with multiple data types (like an excel table with text and numbers)

- **Lists** - ordered collection of any objects (can contain also other lists inside it)



But..

**which dataset should we use to do all
these things?**

Arabidopsis thaliana mutants *psbo1* and *psbo2*

WT



psbo1

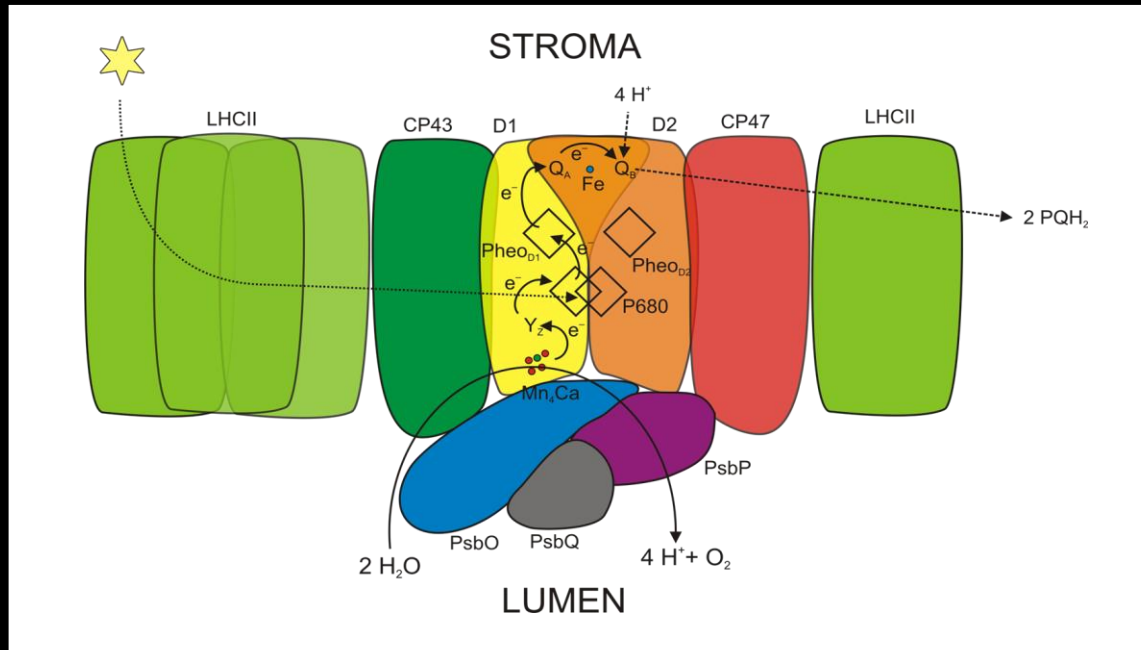


psbo2



PsbO protein

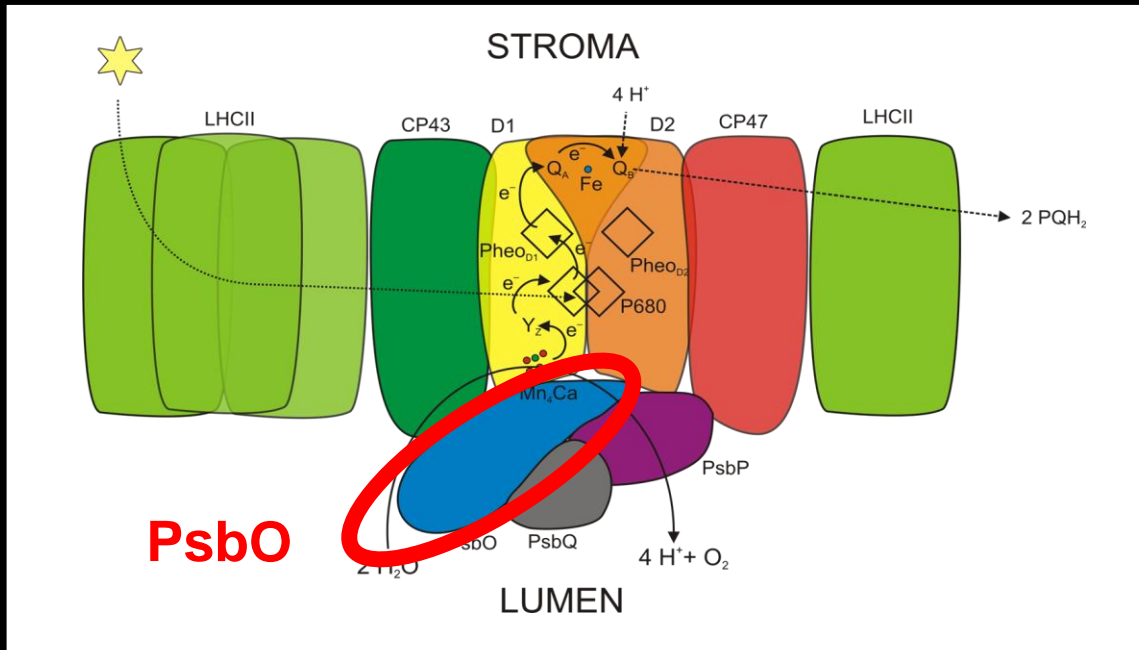
- Subunit of photosystem II
- Important for water splitting
- *Arabidopsis*: PsbO1 and PsbO2



Photosystem II

PsbO protein

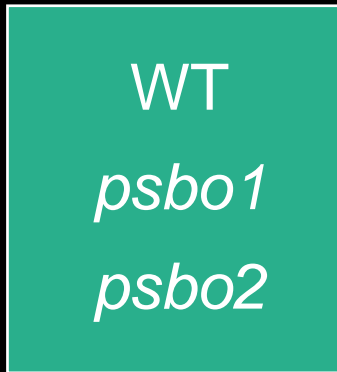
- Subunit of photosystem II
- Important for water splitting
- *Arabidopsis*: PsbO1 and PsbO2



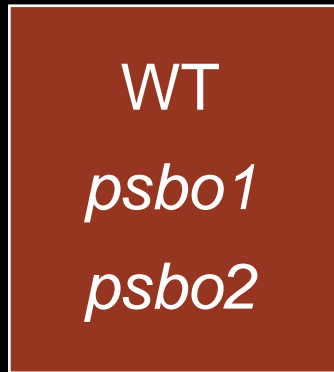
Photosystem II

Experimental design

Control

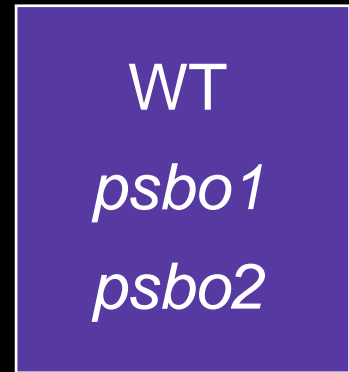


Drought



- water

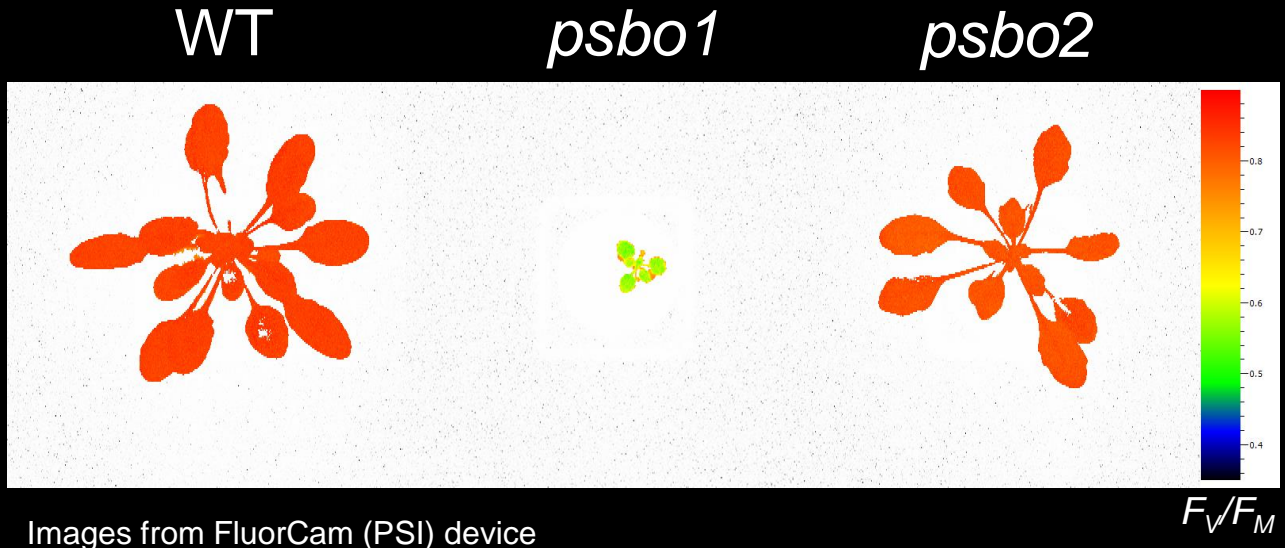
Salt stress



+ NaCl

Measurement – chlorophyll fluorescence

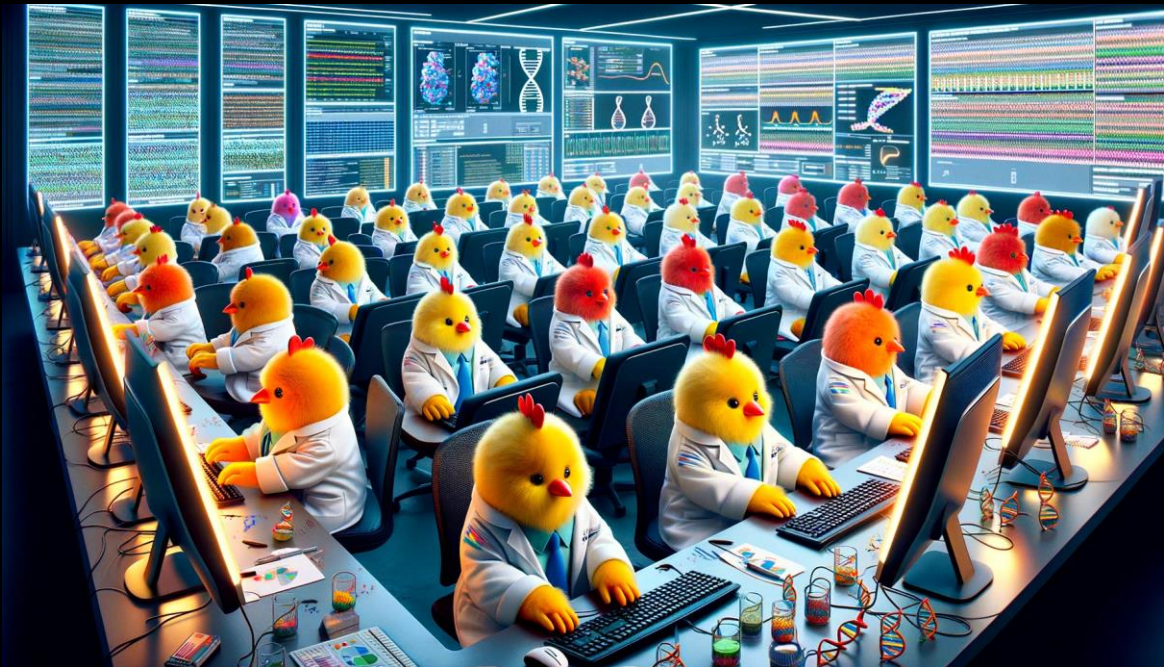
- Leaf rosette area
- F_V/F_M (QY_max) – maximum quantum yield of photosystem II



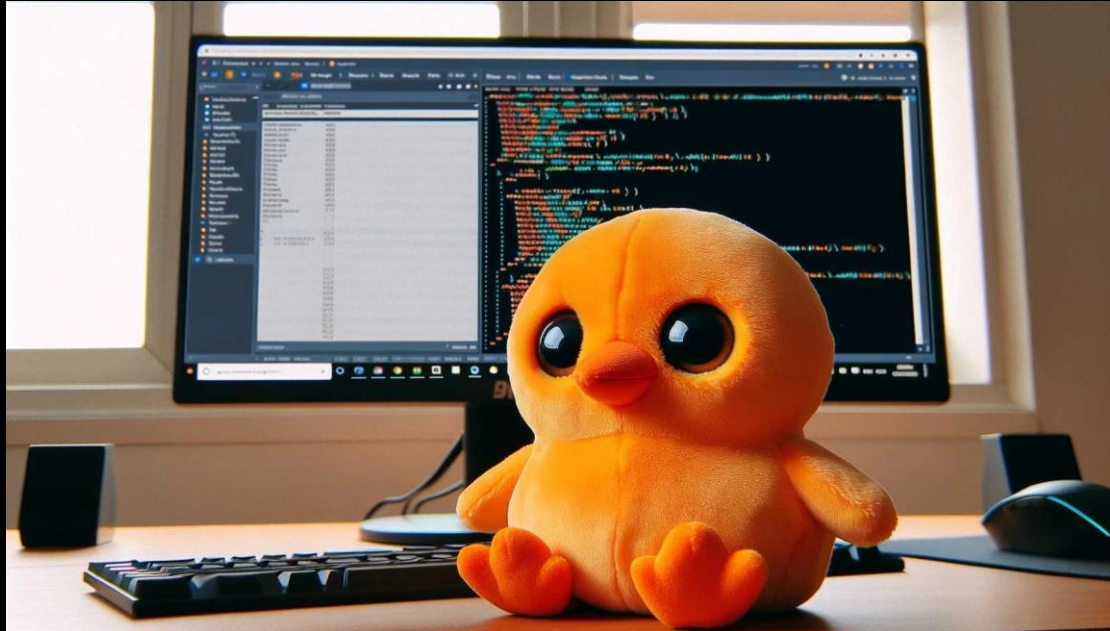
Yey!

Let's start the practical!

Open the Rstudio server by typing in browser:
<your IP>:8787



No matter, if you have never used R, had trouble using it, or feel like an absolute R expert...



Remember: It will all be fine