Evomics 2026

R & ggplot2

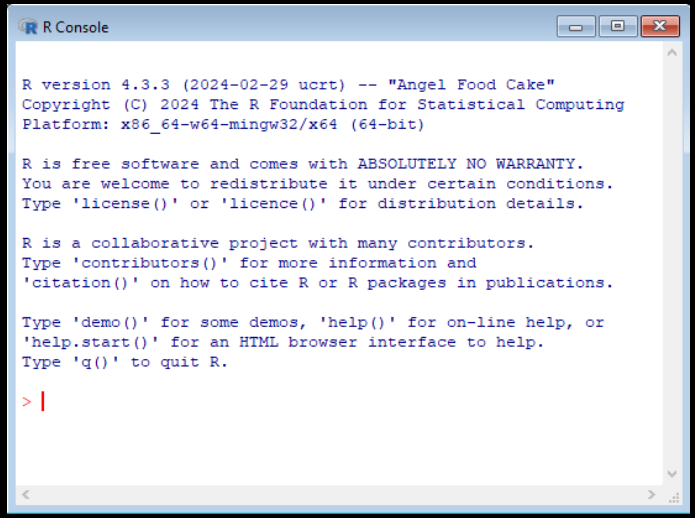
Outline

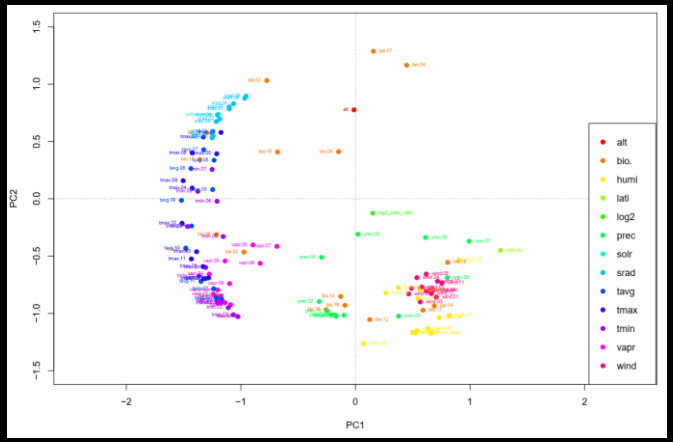
- Short introduction
  - Why is R useful
  - RStudio
  - R Markdown
  - Data structures
- Dataset for practical
- Practical

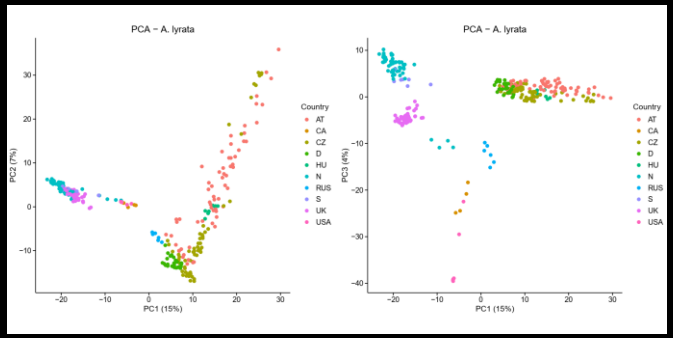
What is R?

A free software environment (and language) for statistical computing and graphics

http://www.r-project.org







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**

When do we use R?

Big data *(fastq etc.)*

UNIX tools

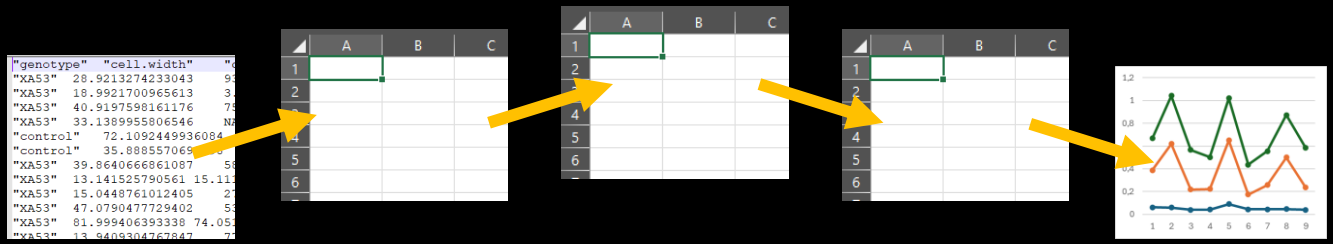
Small data *(count tables etc.)*

R / Python

Final results, visualisations

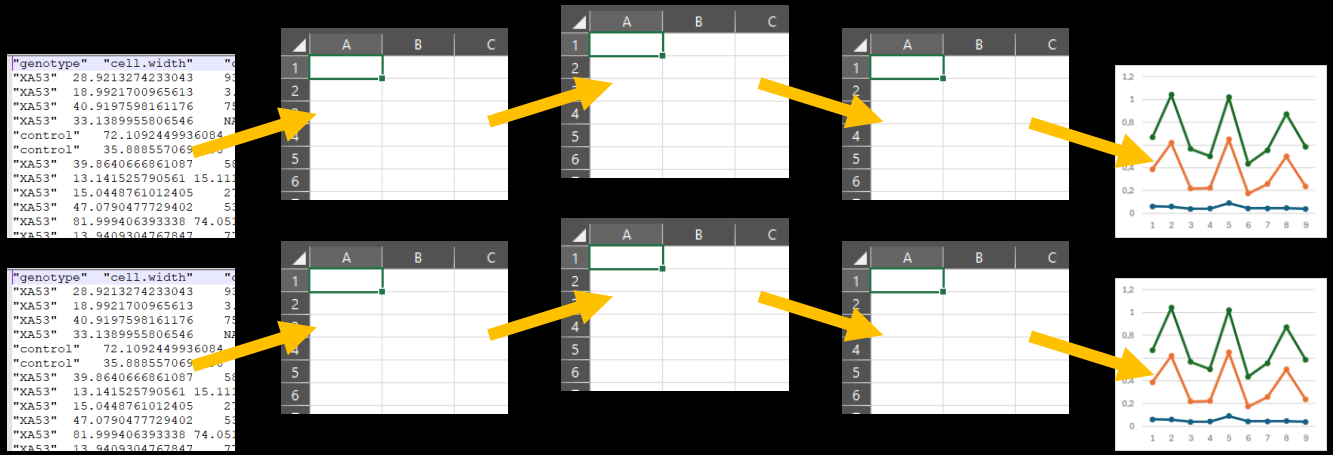
Why R and not Excel?

Why R and not Excel?



Why R and not Excel?

Scenario 1: Data changed

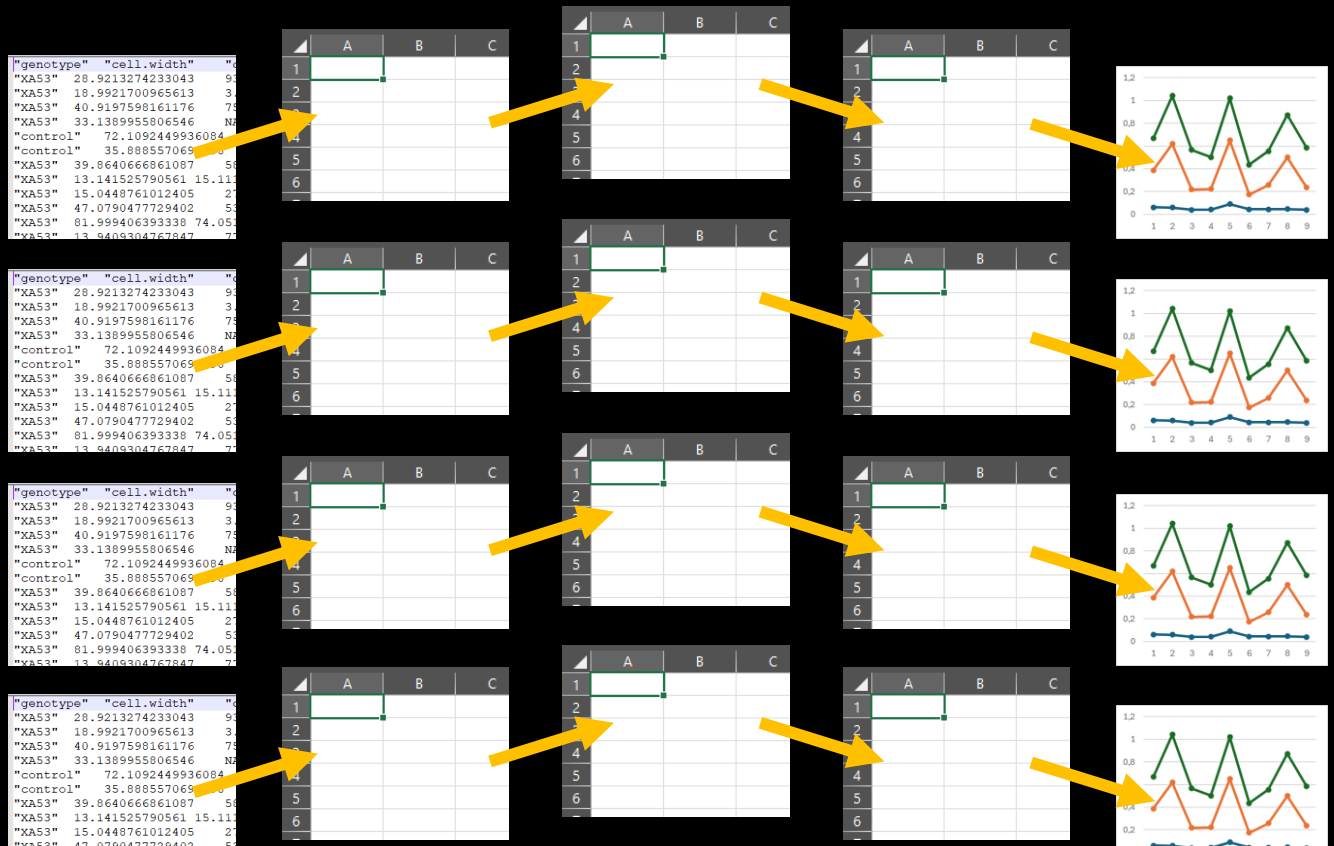


Why R and not Excel?

Scenario 1: Data changed

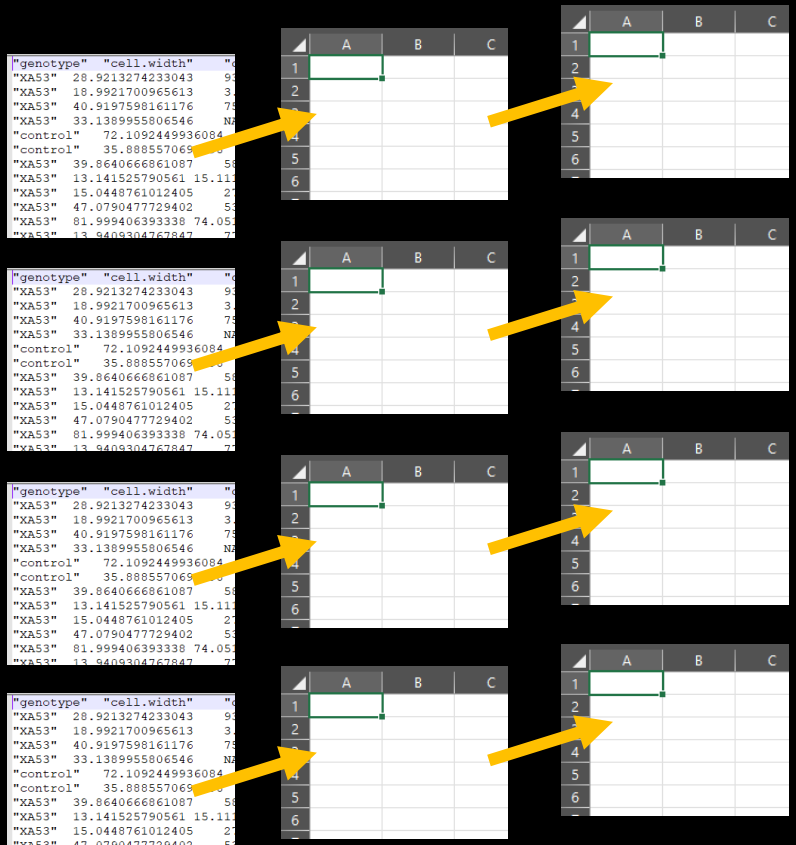
Why R and not Excel?

Scenario 1: Data changed



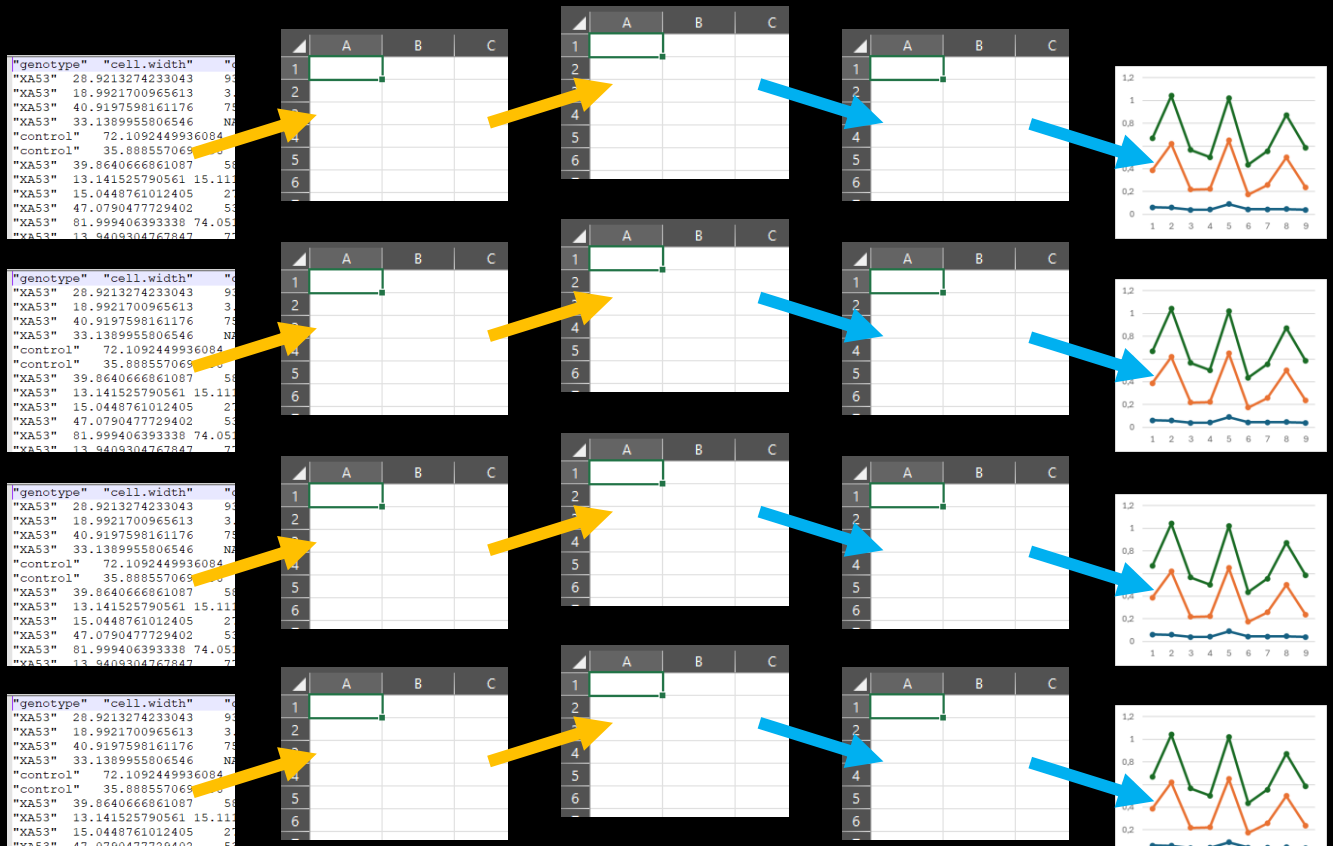
Why R and not Excel?

Scenario 2: Analysis changed



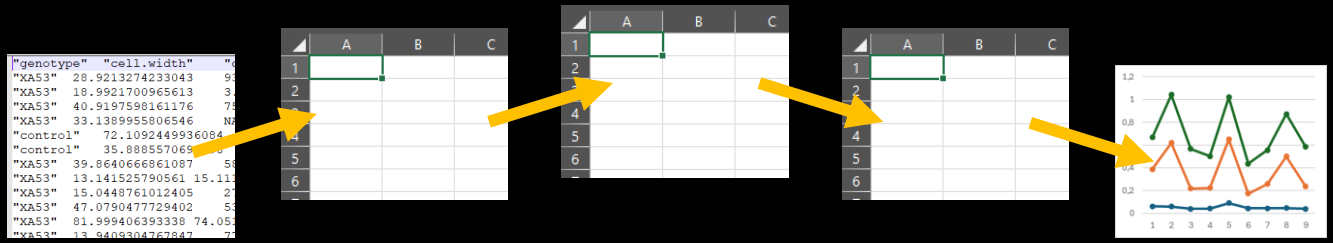
Why R and not Excel?

Scenario 2: Analysis changed



Why R and not Excel?

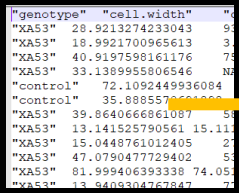
Scenario 3: Many plots needed



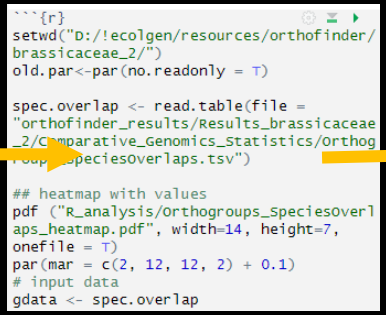
Why R and not Excel?

Scenario 3: Many plots needed

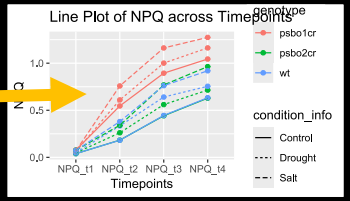
Why R and not Excel?



"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449936084	
"control"	35.888557	
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

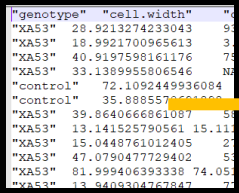


```r
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

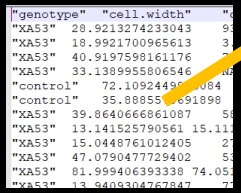


Why R and not Excel?

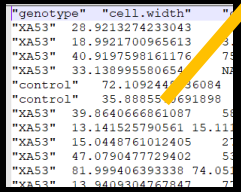
Scenario 1: Data changed



"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449936084	
"control"	35.8885	
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

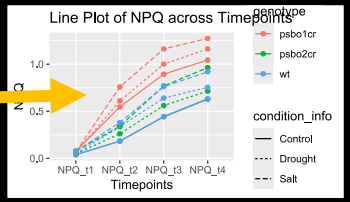


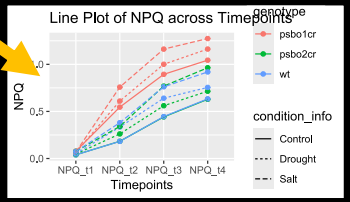
"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449	084
"control"	35.8885	691898
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

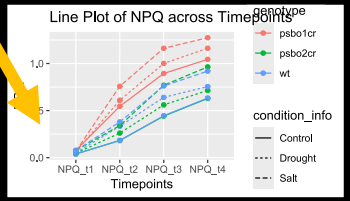


"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.138995580654	NA
"control"	72.109244	36084
"control"	35.88855	691898
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

```r
setwd("D:/!ecolgen/resources/orthofinder/
brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file =
"orthofinder_results/Results_brassicaceae
_2/Comparative_Genomics_Statistics/Orthog
roup_SpeciesOverlaps.tsv")

## Heatmap with values
pdf("R_analysis/Orthogroups_SpeciesOverl
aps_heatmap.pdf", width=14, height=7,
onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# Input data
data <- spec.overlap
```

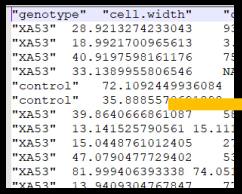




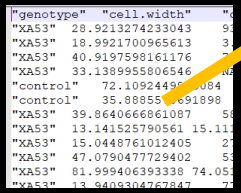


Why R and not Excel?

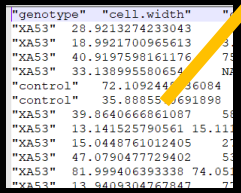
Scenario 2: Analysis changed



"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449936084	
"control"	35.8885	
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

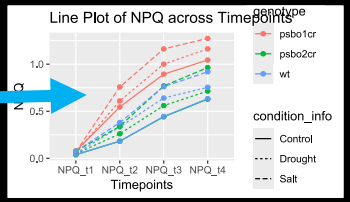


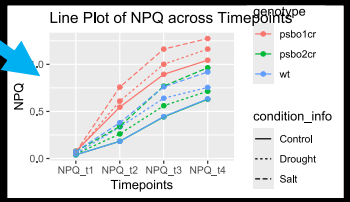
"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449	084
"control"	35.8885	691898
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

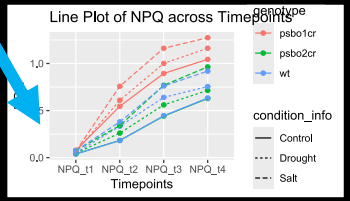


"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449	084
"control"	35.8885	691898
"XA53"	39.8640666861087	5
"XA53"	13.141525790561	15.11
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338	74.05
"XA53"	13.9409304767847	7

```r
setwd("D:/!ecolgen/resources/orthofinder/
brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file =
"orthofinder_results/Results_brassicaceae
_2/Comparative_Genomics_Statistics/Orthog
roup_SpeciesOverlaps.tsv")

## heatmap with values
("R_analysis/Orthogroups_SpeciesOverl
aps_heatmap.pdf", width=14, height=7,
onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
input data
data <- spec.overlap
```

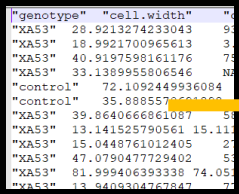




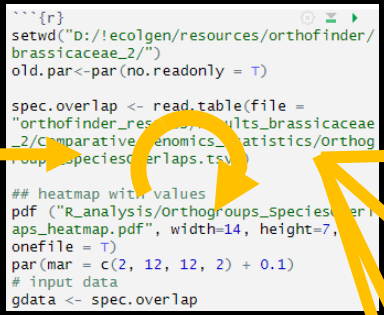


Why R and not Excel?

Scenario 3: Many plots needed



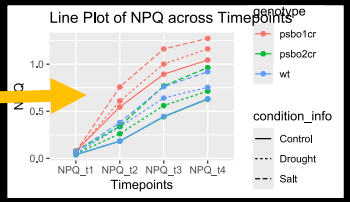
"genotype"	"cell.width"	"c
"XA53"	28.9213274233043	9
"XA53"	18.9921700965613	3
"XA53"	40.9197598161176	7
"XA53"	33.1389955806546	NA
"control"	72.1092449936084	
"control"	35.888557	
"XA53"	39.8640666861087	5
"XA53"	13.141525790561 15.11	
"XA53"	15.0448761012405	2
"XA53"	47.0790477729402	5
"XA53"	81.999406393338 74.05	
"XA53"	13.9409304767847	7

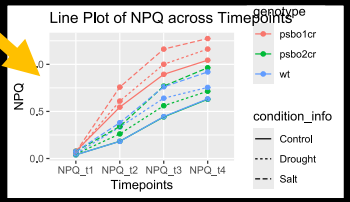


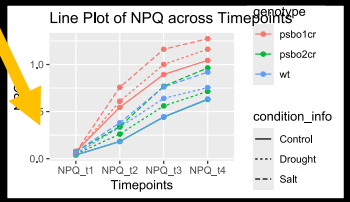
```r
```{r}
setwd("D:/!ecolgen/resources/orthofinder/
brassicaceae_2/")
old.par<-par(no.readonly = T)

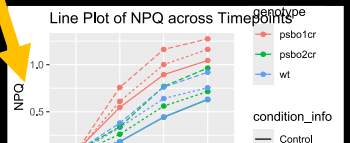
spec.overlap <- read.table(file =
"orthofinder_res      ults_brassicaceae
_2/Comparative_Genomics_Statistics/Orthog
roup_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOver
aps_heatmap.pdf", width=14, height=7,
onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```








```

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Me in 2 months

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
```{r}
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Me in 2 months

Me in 2 years
```

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
```{r}
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Me in 2 months

Me in 2 years

Collaborator
```

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file = "orthofinder_results/Results_brassicaceae_2/Comparative_Genomics_Statistics/Orthogroups_SpeciesOverlaps.tsv")

## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Me in 2 months

Me in 2 years

Collaborator

Paper reader

Why R and not Excel?

Scenario 4: Someone wants to understand or repeat the analysis

```r
setwd("D:/!ecolgen/resources/orthofinder/brassicaceae_2/")
old.par<-par(no.readonly = T)

spec.overlap <- read.table(file =
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```

Reproducibility

Me in 2 months

Me in 2 years

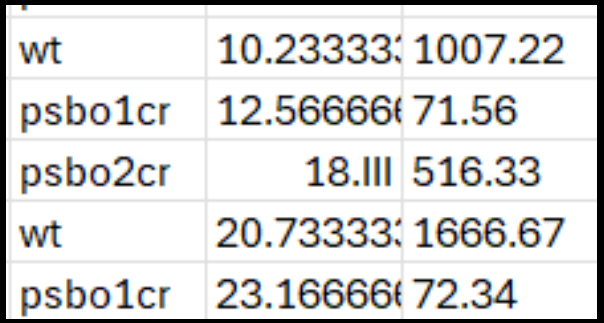
Collaborator

Paper reader

I WANT YOU
FOR REPRODUCIBLE SCIENCE

Why R and not Excel?

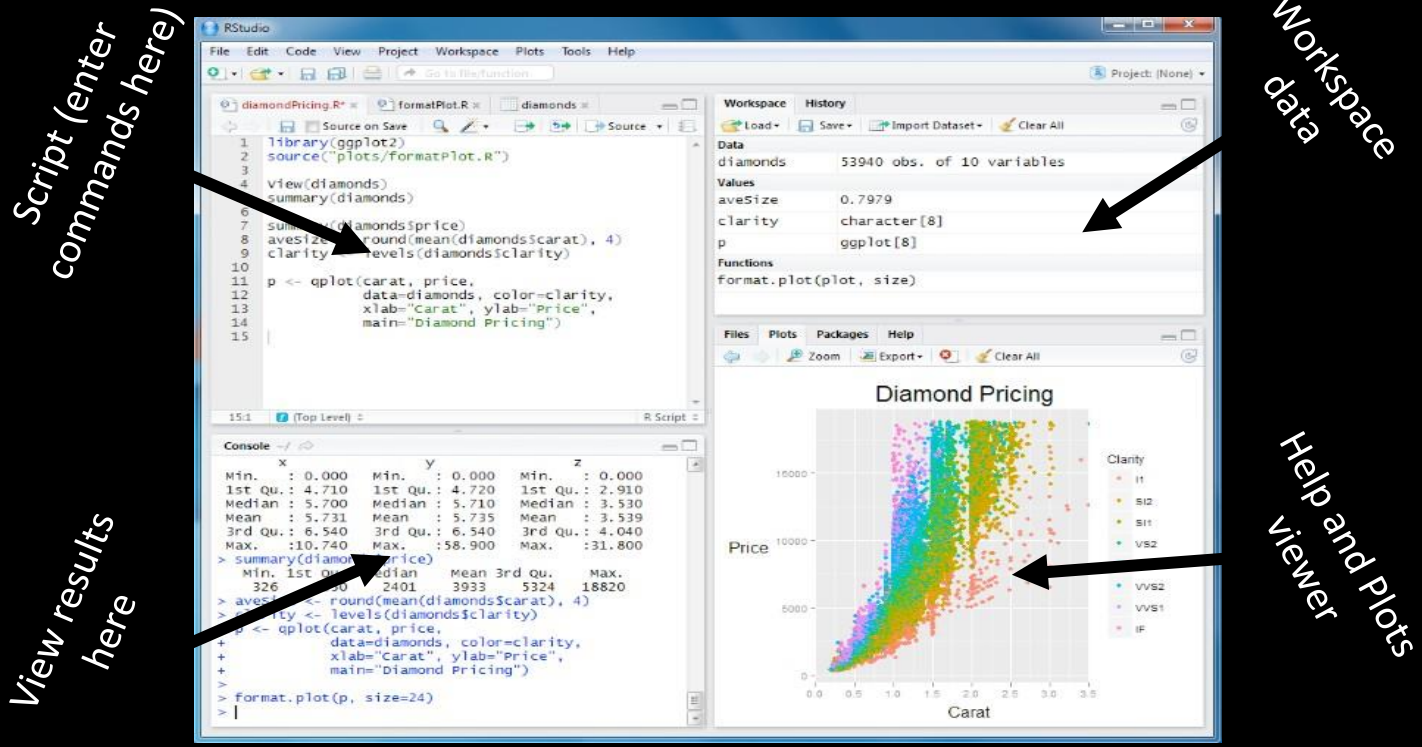
- Automation
  - Many plots in one loop
  - Easily repeated if the data changes
- Reproducibility and transparency
  - You will know later what you did with the data
  - Other people will know what you did with the data
  - You can publish your code with your paper
- Excel tends to change some numbers to dates etc.



wt	10.23333	1007.22
psbo1cr	12.56666	71.56
psbo2cr	18.III	516.33
wt	20.73333	1666.67
psbo1cr	23.16666	72.34

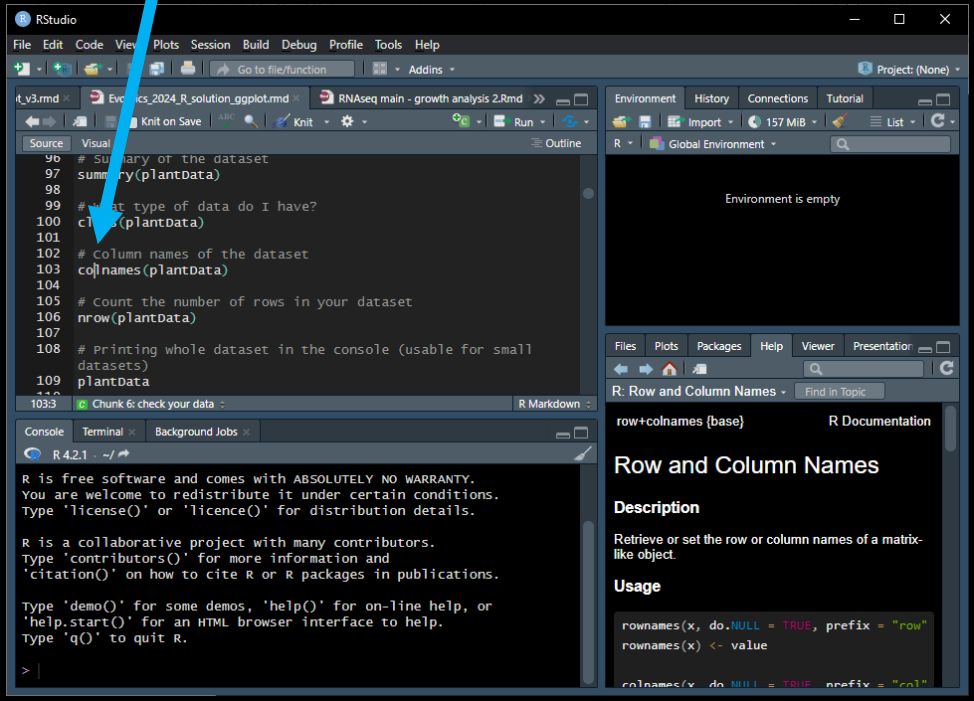
R Studio

Integrated development environment (IDE) for R



Help in R Studio

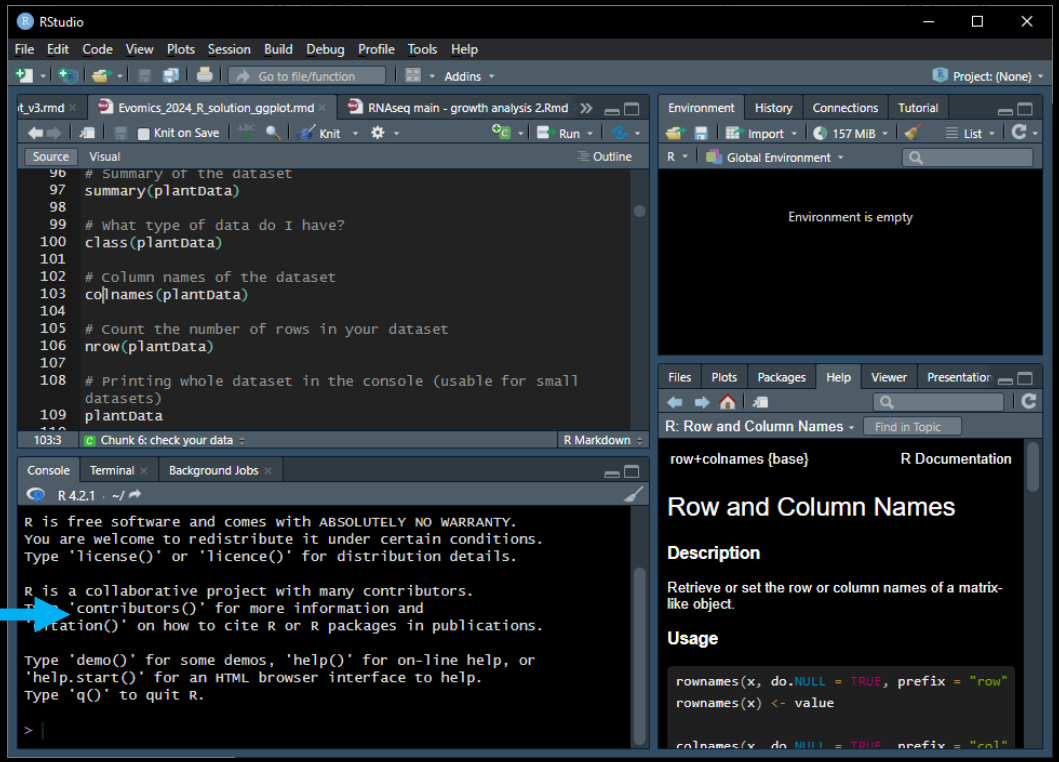
Press **F1** when the cursor is in the name of the function



The help will open here

Where to write the code?

Console? Not good for reproducibility.

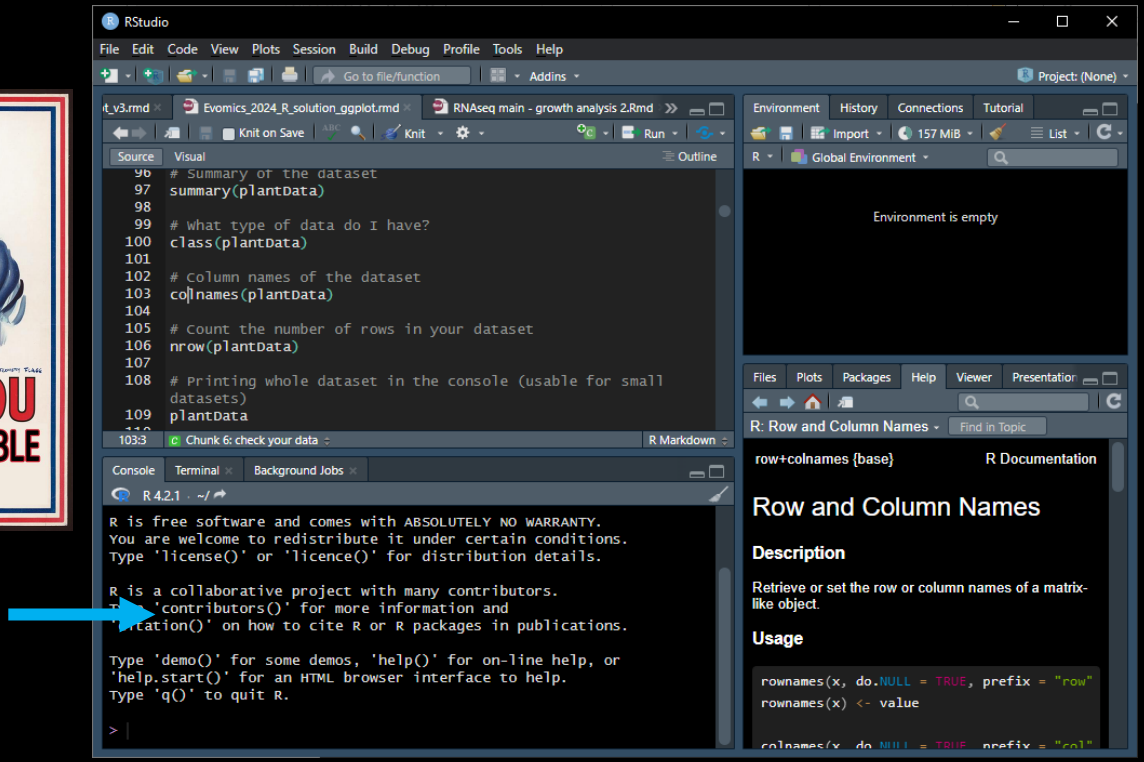


Console

Where to write the code?

Console? Not good for reproducibility.

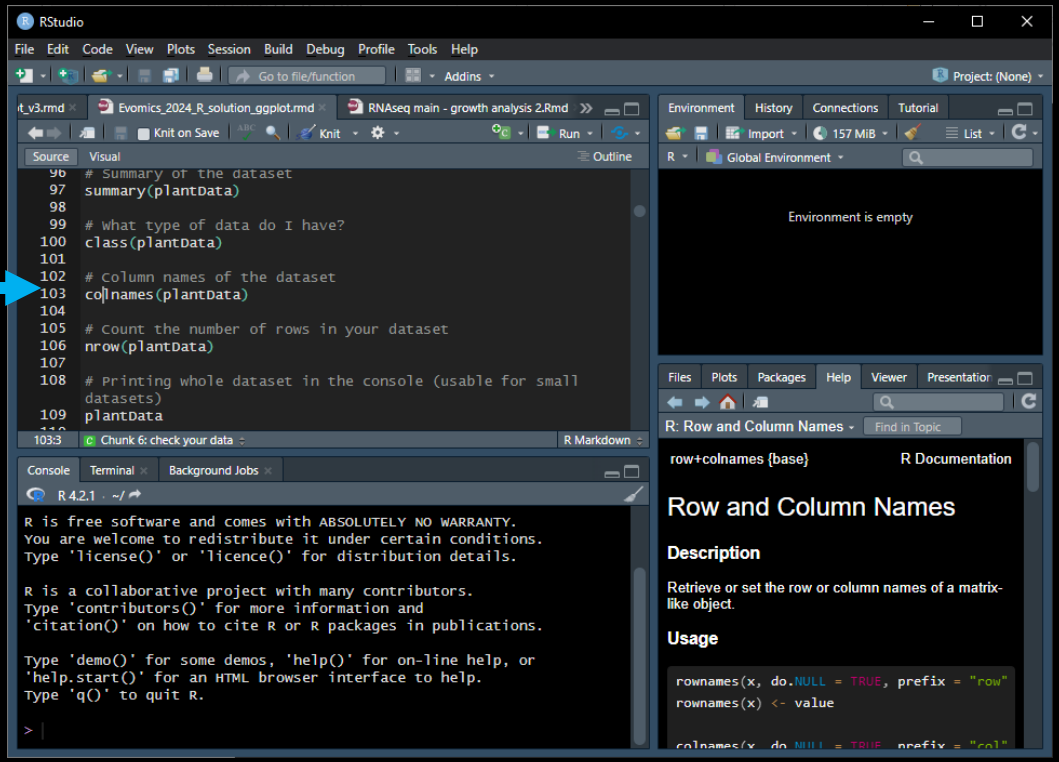
Console



Where to write the code?

R Script / R Markdown

Source



Where to write the code?

R Script: Code + # Comments

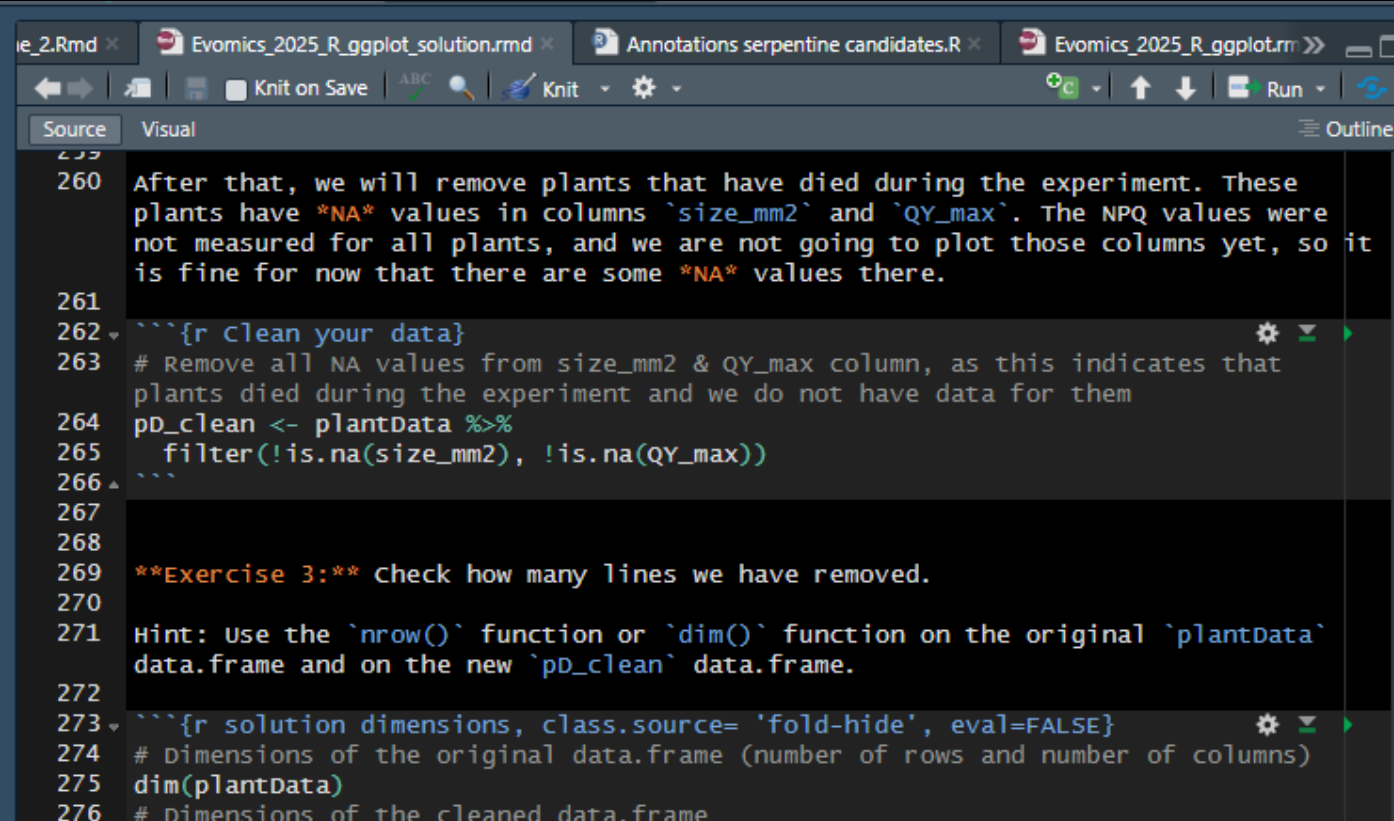
e_2.Rmd | Annotations serpentine candidates.R | Evomics_2025_R_ggplot.rmd | Evomics_2025_R_ggplot_solution.rm

Source on Save | Run | Source

```
66
67
68  ## 1. List of genes (AGI codes; there can be multiple genes per line separated by
    e.g. ";")
69
70  # data (gene numbers)
71  genes.pre0 <- read.csv2("data/Konecna2021_NatCom_Supplementary_data_7_for_R.csv",
72                          stringsAsFactors = F)
73
74  genes <- genes.pre0[!duplicated(genes.pre0$ID), 1:2]
75  colnames(genes)[c(1, 2)] <- c("Al.ids", "ids")
76  head(genes)
77
78  # new orthologs from thaliana (according to Brassicaceae_orthology_0)
79  ortho <- read.table(file = "data/A_lyrata_Rawat__v__A_thaliana.tsv", header = T,
    sep = "\t")
80  head(ortho)
81  ids <- sapply(X = genes$Al.ids, FUN = gene.properties, table = ortho[, 2], feature
    = ortho[, 3], split = ", ", USE.NAMES = F)
82  cbind(genes$Al.ids, genes$ids, ids, genes$ids == ids)
83  ortho[grep(pattern = "AL8G36200", x = ortho$A_lyrata_Rawat), ]
84  ortho[grep(pattern = "AL6G29060", x = ortho$A_lyrata_Rawat), ]
85  # Notes: Some orthologs are missing. Most (except of 2) are the same.
86  # I will use the old homologs for now.
```

Where to write the code?

R Markdown: Formatted text + ```Code chunks```



R Markdown

Can be “knitted” to produce report in html, pdf, docx, GitHub md etc.

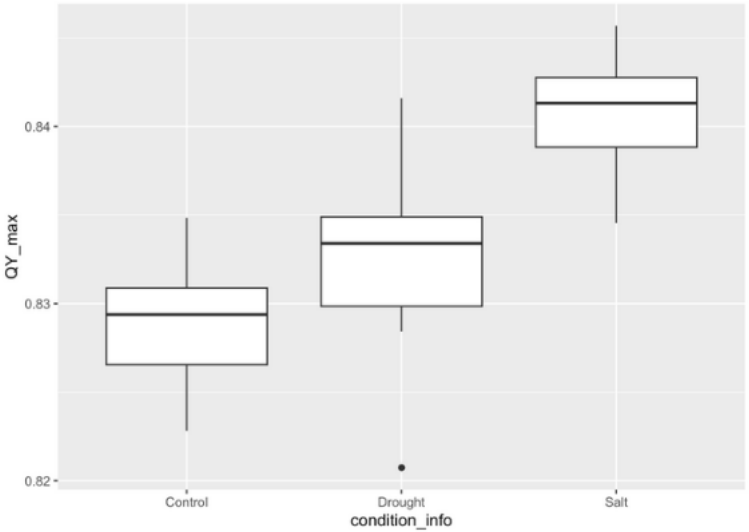
4.3 Modify your graph aesthetics

We will now make our box plot a bit fancier. Although the defaults often work well, you can modify almost everything within the `ggplot2` package.

Here you can see how to modify various things in the plot.

Hide

```
# Original box plot of QY_max by condition_info
p1 <- ggplot(pD_clean_wt, aes(x=condition_info, y=QY_max)) +
  geom_boxplot()
p1
```



Hide

```
# Now let's get fancy with this plot. We'll start with our p1 plot and sequen
tially add layers to it.

p1_fancy <- ggplot(pD_clean_wt, aes(x=condition_info, y=QY_max)) +
  geom_boxplot() + # add a boxplot layer (same as before)
  geom_point() + # add points to the boxplot
```

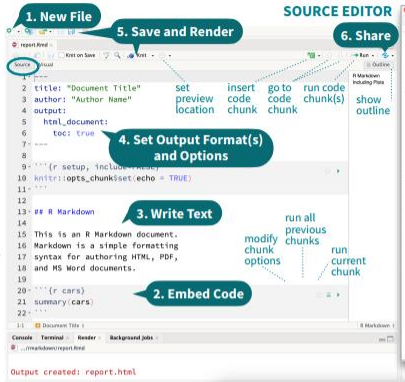
rmarkdown : : CHEATSHEET

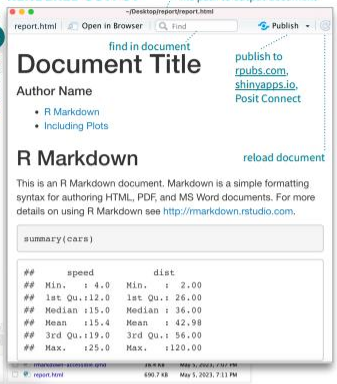
What is rmarkdown?

.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 plots, 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!**

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**

superscript^2^/subscript~2~

~~strikethrough~~

escaped: * _ \\

endash: --, emdash: ---

Header 1
Header 2
...
Header 6

- unordered list
- item 2
  - item 2a (indent 1 tab)
  - item 2b

1. ordered list
2. item 2
   - item 2a (indent 1 tab)
   - item 2b

<link url>

[This is a link.](link url)

[This is another link][id].

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

![Caption](image.png) or ![Caption][id2]

At the end of the document: [id2]: image.png

`verbatim code`

```
multiple lines
of verbatim code
```

> block quotes

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

equation block:
$$E = mc^{2}$$

horizontal rule:

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

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**

superscript^2/subscript$_2$

~~strikethrough~~

escaped: * _ \

endash: –, emdash: —

Header 1
Header 2
...
Header 6

- unordered list
- item 2
  - item 2a (indent 1 tab)
  - item 2b

1. ordered list
2. item 2
   - item 2a (indent 1 tab)
   - item 2b

http://www.posit.co/

This is a link.

This is another link.

Caption.

`verbatim code`

```
multiple lines
of verbatim code
```

> block quotes

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

equation block:

$$E = mc^2$$

horizontal rule:

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

HTML Tabsets

```
## Results {.tabset}
### Plots
text

### Tables
more text
```

Results

Plots | Tables

text

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"

OPTION	DEFAULT	EFFECTS
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after running
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"asis" (passthrough 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
purl	TRUE	include or exclude a code chunk when extracting source code with knitr::purl()

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

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).

Table with kable		
	emptions	waiting
	3.600	79
	1.800	54
	3.333	74
	2.283	62

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

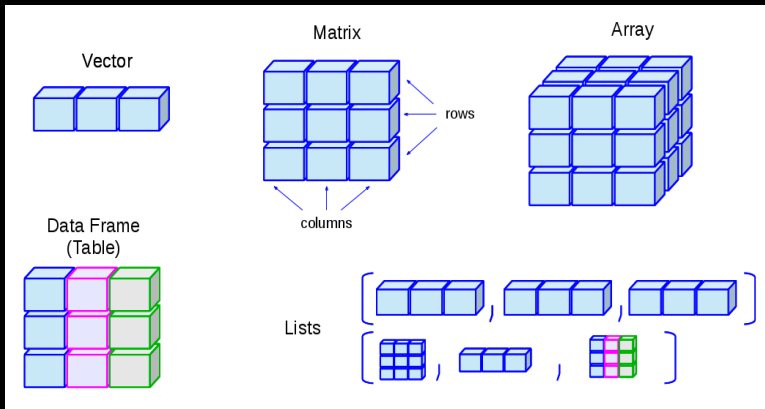
Other table packages include **flextable**, **gt**, and **kableExtra**.

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)



http://venus.ifca.unican.es/Rintro/dataStruct.html

But…

which dataset should we use to try R?

Arabidopsis thaliana mutants *psbo1* and *psbo2*

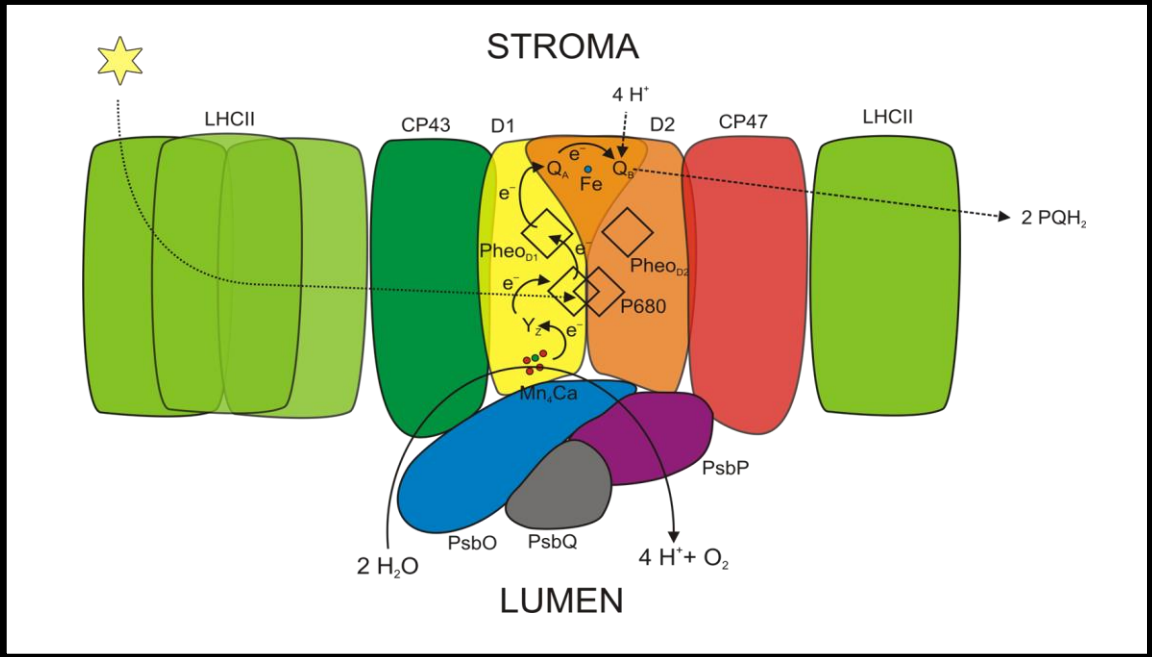
WT

psbo1

psbo2

PsbO protein

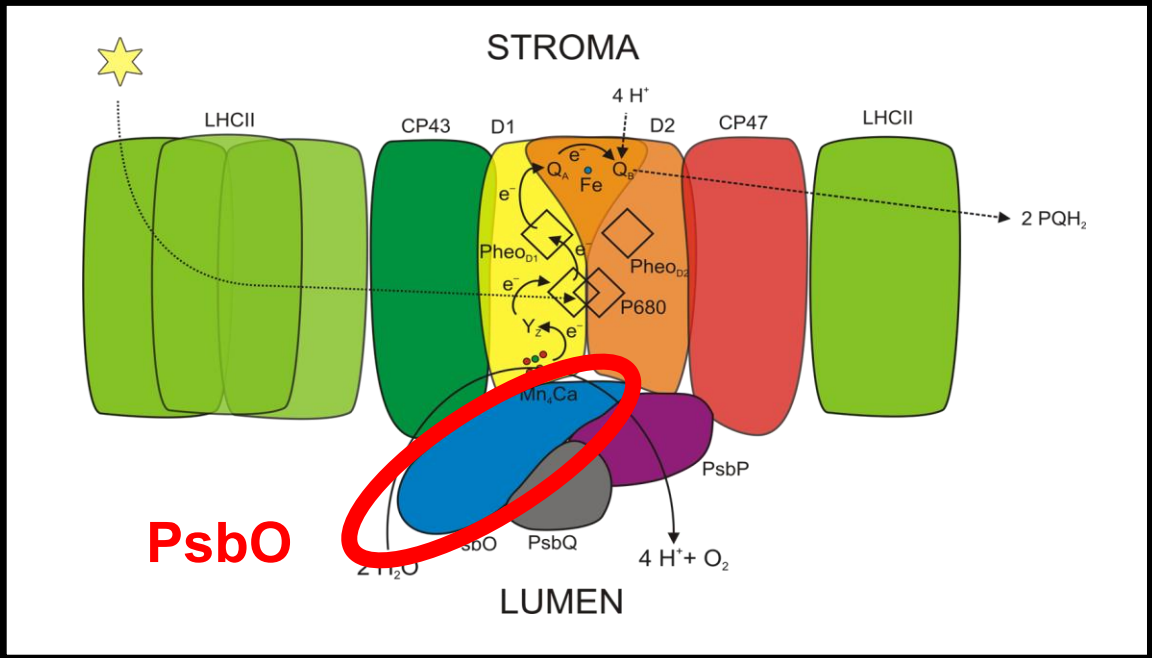
- Subunit of photosystem II
- Important for water splitting
- *Arabidopsis*: PsbO1 and PsbO2



Photosystem II

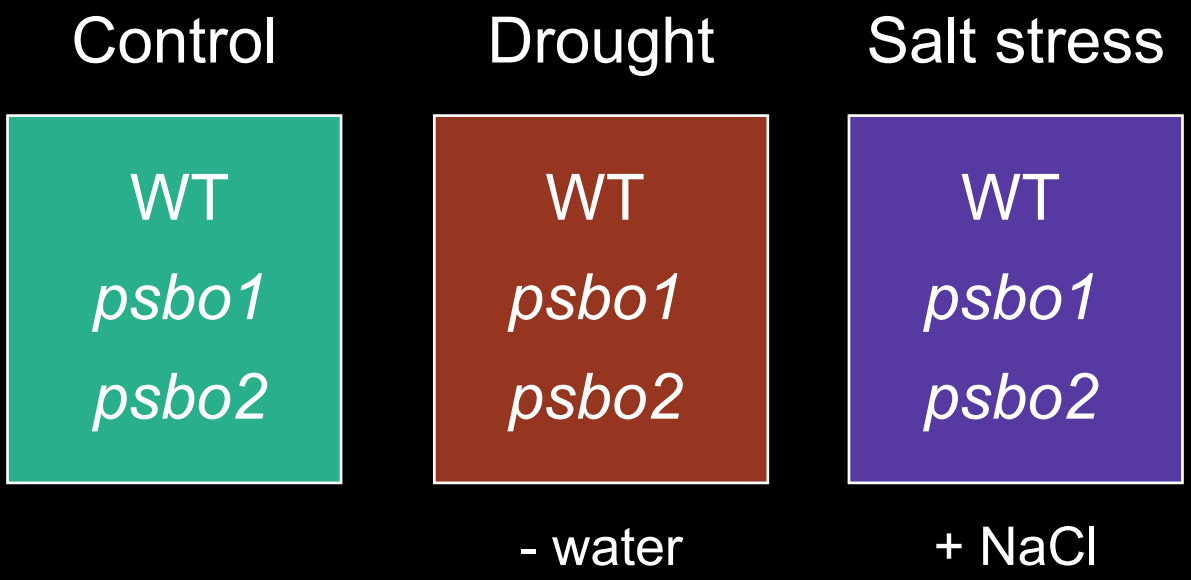
PsbO protein

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



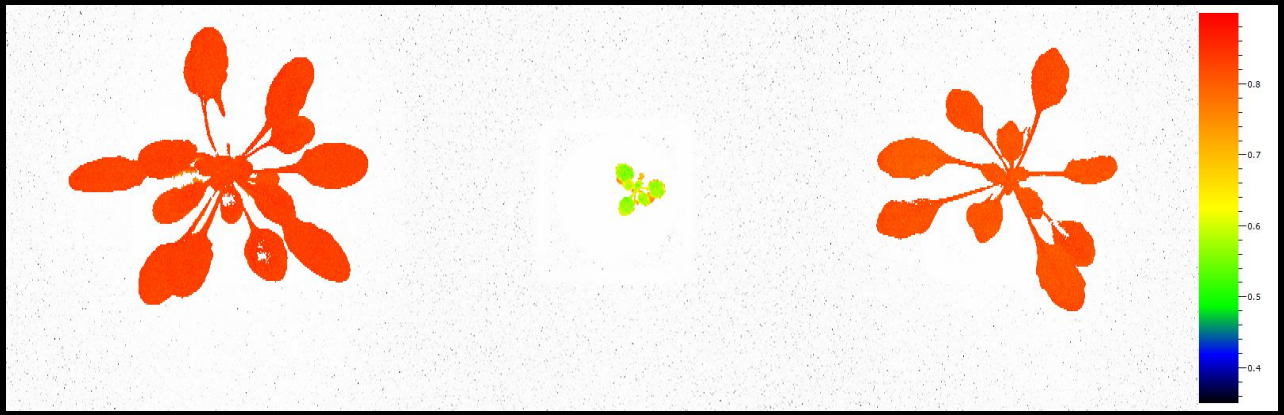
Photosystem II

Experimental design



Measurement – chlorophyl fluorescence

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



Images from FluorCam (PSI) device

Let‘s start the practical!

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

Remember:

- Practise makes the masters.
- Do sanity checks. Always.
- Use AI, but try to understand, check and improve the code.