

Evomics 2025

R & ggplot2



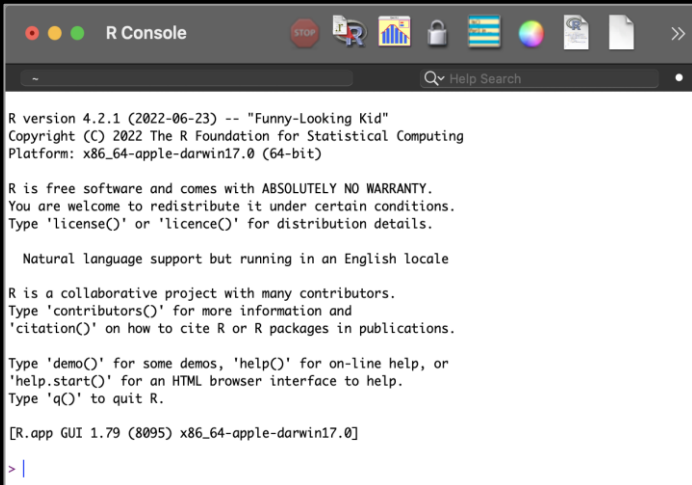
Outline

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

What is R?

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

<http://www.r-project.org>



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)

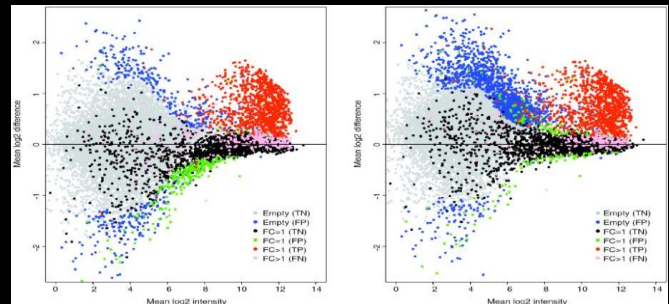
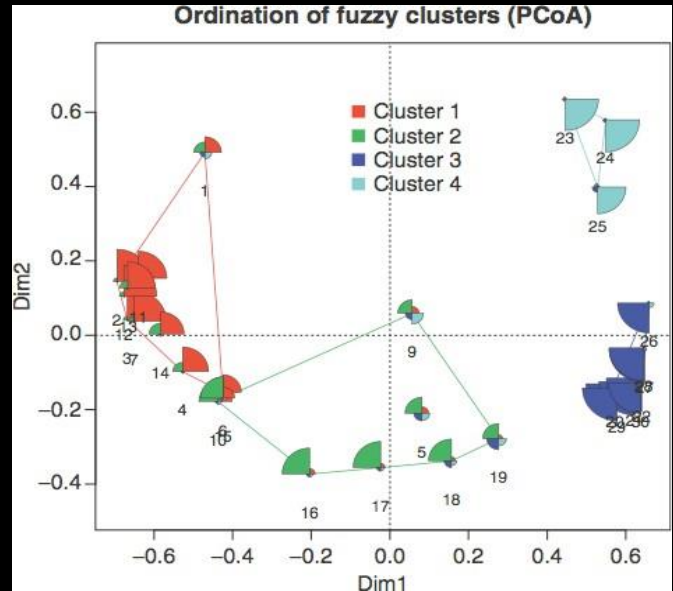
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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 (8095) 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
(Wednesday)

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



Rachel Steward
(Tuesday)

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

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

Why R and not Excel?



Why R and not Excel?

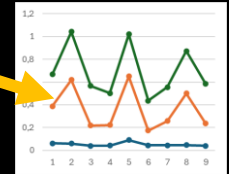


```
"genotype" "cell.width" "XAS3" 28.9213274233043 98  
"XAS3" 18.9921700965613 38  
"XAS3" 40.9197598161176 75  
"XAS3" 33.1389955806546 NA  
"control" 72.1092449936084  
"control" 35.888557069  
"XAS3" 39.8640666861087 58  
"XAS3" 13.141525790561 15.11  
"XAS3" 15.0448761012405 23  
"XAS3" 47.0790477729402 53  
"XAS3" 81.999406393338 74.05  
"XAS3" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

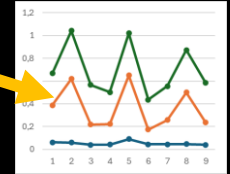
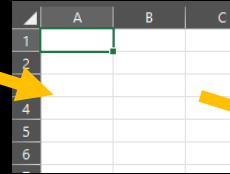
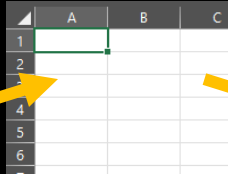
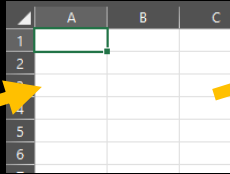


Why R and not Excel?

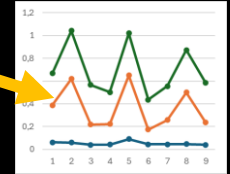
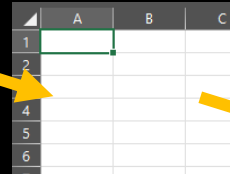
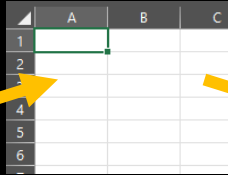
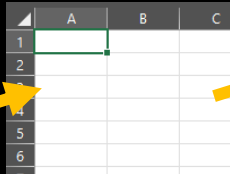
Scenario 1: Data changed



```
"genotype" "cell.width" "6"  
"XA53" 28.9213274233043 94  
"XA53" 18.9921700965613 3  
"XA53" 40.9197598161176 78  
"XA53" 33.1389955806546 NA  
"control" 72.1092449936084  
"control" 35.888557069  
"XA53" 39.8640666861087 58  
"XA53" 13.141525790561 15.11  
"XA53" 15.0448761012405 2  
"XA53" 47.0790477729402 53  
"XA53" 81.999406393338 74.05  
"XA53" 13.9409304767847 7
```



```
"genotype" "cell.width" "6"  
"XA53" 28.9213274233043 94  
"XA53" 18.9921700965613 3  
"XA53" 40.9197598161176 78  
"XA53" 33.1389955806546 NA  
"control" 72.1092449936084  
"control" 35.888557069  
"XA53" 39.8640666861087 58  
"XA53" 13.141525790561 15.11  
"XA53" 15.0448761012405 2  
"XA53" 47.0790477729402 53  
"XA53" 81.999406393338 74.05  
"XA53" 13.9409304767847 7
```

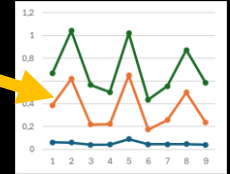
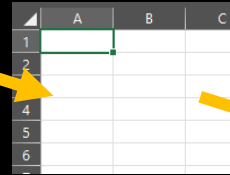
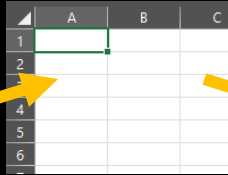
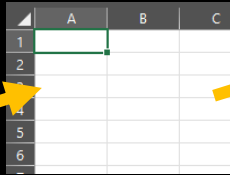


Why R and not Excel?

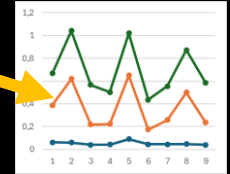
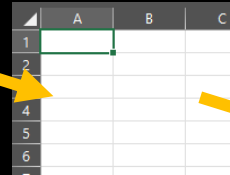
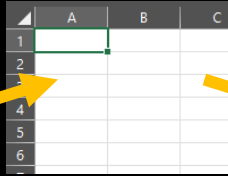
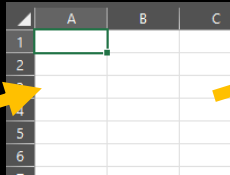
Scenario 1: Data changed



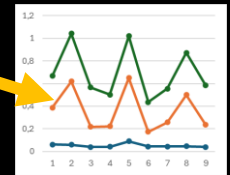
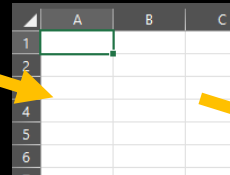
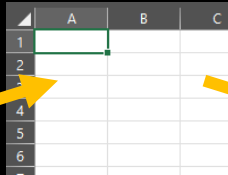
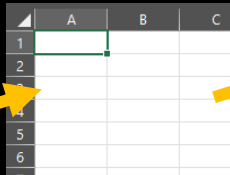
```
"genotype" "cell_width" "N"
"XAS3" 28.9213274233043 98
"XAS3" 18.9921700965613 38
"XAS3" 40.9197598161176 78
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 NA
"control" 35.888557069 NA
"XAS3" 39.8640666861087 58
"XAS3" 13.141525790561 15.111
"XAS3" 15.0448761012405 27
"XAS3" 47.079047729402 53
"XAS3" 81.999406393338 74.053
"XAS3" 13.9409304767847 7
```



```
"genotype" "cell_width" "N"
"XAS3" 28.9213274233043 98
"XAS3" 18.9921700965613 38
"XAS3" 40.9197598161176 78
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 NA
"control" 35.888557069 NA
"XAS3" 39.8640666861087 58
"XAS3" 13.141525790561 15.111
"XAS3" 15.0448761012405 27
"XAS3" 47.079047729402 53
"XAS3" 81.999406393338 74.053
"XAS3" 13.9409304767847 7
```



```
"genotype" "cell_width" "N"
"XAS3" 28.9213274233043 98
"XAS3" 18.9921700965613 38
"XAS3" 40.9197598161176 78
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 NA
"control" 35.888557069 NA
"XAS3" 39.8640666861087 58
"XAS3" 13.141525790561 15.111
"XAS3" 15.0448761012405 27
"XAS3" 47.079047729402 53
"XAS3" 81.999406393338 74.053
"XAS3" 13.9409304767847 7
```

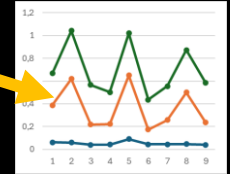
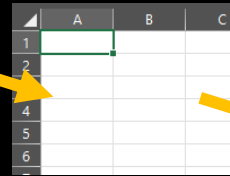
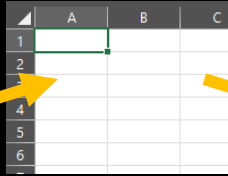
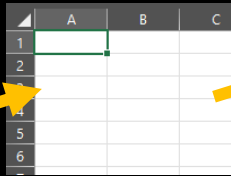


Why R and not Excel?

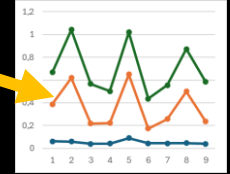
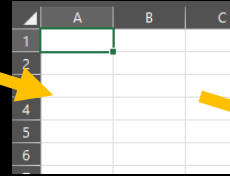
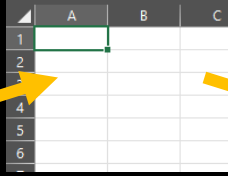
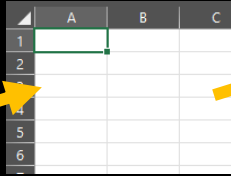
Scenario 1: Data changed



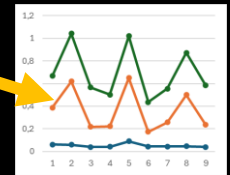
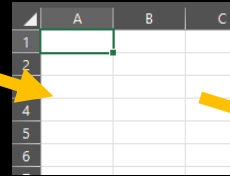
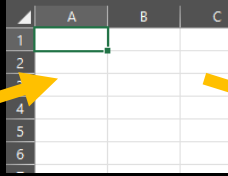
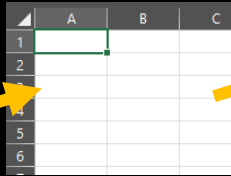
```
"genotype" "cell_width" "NA"
"XAS3" 28.9213274233043 98.1111111111111
"XAS3" 18.9921700965613 33.3333333333333
"XAS3" 40.9197598161176 75.7575757575758
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 100.000000000000
"control" 35.8885570690000 50.0000000000000
"XAS3" 39.8640666861087 58.3333333333333
"XAS3" 13.141525790561 15.1111111111111
"XAS3" 15.0448761012405 27.2727272727273
"XAS3" 47.079047729402 50.0000000000000
"XAS3" 81.999406393338 74.0500000000000
"XAS3" 13.9409304767847 27.2727272727273
```



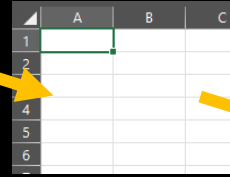
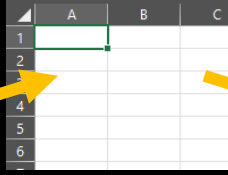
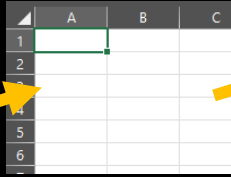
```
"genotype" "cell_width" "NA"
"XAS3" 28.9213274233043 98.1111111111111
"XAS3" 18.9921700965613 33.3333333333333
"XAS3" 40.9197598161176 75.7575757575758
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 100.000000000000
"control" 35.8885570690000 50.0000000000000
"XAS3" 39.8640666861087 58.3333333333333
"XAS3" 13.141525790561 15.1111111111111
"XAS3" 15.0448761012405 27.2727272727273
"XAS3" 47.079047729402 50.0000000000000
"XAS3" 81.999406393338 74.0500000000000
"XAS3" 13.9409304767847 27.2727272727273
```



```
"genotype" "cell_width" "NA"
"XAS3" 28.9213274233043 98.1111111111111
"XAS3" 18.9921700965613 33.3333333333333
"XAS3" 40.9197598161176 75.7575757575758
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 100.000000000000
"control" 35.8885570690000 50.0000000000000
"XAS3" 39.8640666861087 58.3333333333333
"XAS3" 13.141525790561 15.1111111111111
"XAS3" 15.0448761012405 27.2727272727273
"XAS3" 47.079047729402 50.0000000000000
"XAS3" 81.999406393338 74.0500000000000
"XAS3" 13.9409304767847 27.2727272727273
```



```
"genotype" "cell_width" "NA"
"XAS3" 28.9213274233043 98.1111111111111
"XAS3" 18.9921700965613 33.3333333333333
"XAS3" 40.9197598161176 75.7575757575758
"XAS3" 33.1389955806546 NA
"control" 72.1092449936084 100.000000000000
"control" 35.8885570690000 50.0000000000000
"XAS3" 39.8640666861087 58.3333333333333
"XAS3" 13.141525790561 15.1111111111111
"XAS3" 15.0448761012405 27.2727272727273
"XAS3" 47.079047729402 50.0000000000000
"XAS3" 81.999406393338 74.0500000000000
"XAS3" 13.9409304767847 27.2727272727273
```



Why R and not Excel?

Scenario 2: Analysis changed



```
"genotype" "cell_width" "4"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 38
"XA53" 40.9197598161176 78
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888557069
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.11
"XA53" 15.0448761012405 27
"XA53" 47.0790477729402 53
"XA53" 81.999406393338 74.05
"XA53" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

```
"genotype" "cell_width" "4"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 38
"XA53" 40.9197598161176 78
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888557069
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.11
"XA53" 15.0448761012405 27
"XA53" 47.0790477729402 53
"XA53" 81.999406393338 74.05
"XA53" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

```
"genotype" "cell_width" "4"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 38
"XA53" 40.9197598161176 78
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888557069
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.11
"XA53" 15.0448761012405 27
"XA53" 47.0790477729402 53
"XA53" 81.999406393338 74.05
"XA53" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

```
"genotype" "cell_width" "4"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 38
"XA53" 40.9197598161176 78
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888557069
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.11
"XA53" 15.0448761012405 27
"XA53" 47.0790477729402 53
"XA53" 81.999406393338 74.05
"XA53" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

Why R and not Excel?

Scenario 2: Analysis changed



Why R and not Excel?

Scenario 3: Many plots needed

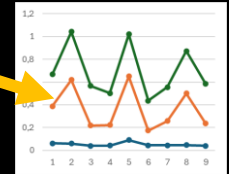


```
"genotype" "cell.width" "0"  
"XAS3" 28.9213274233043 98  
"XAS3" 18.9921700965613 38  
"XAS3" 40.9197598161176 75  
"XAS3" 33.1389955806546 NA  
"control" 72.1092449936084  
"control" 35.888557069  
"XAS3" 39.8640666861087 58  
"XAS3" 13.141525790561 15.11  
"XAS3" 15.0448761012405 21  
"XAS3" 47.0790477729402 53  
"XAS3" 81.999406393338 74.05  
"XAS3" 13.9409304767847 7
```

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

	A	B	C
1			
2			
3			
4			
5			
6			

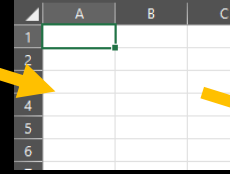
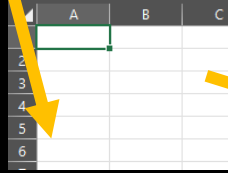
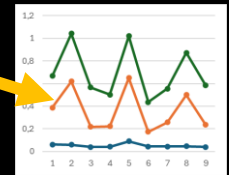
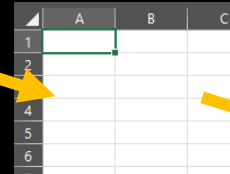
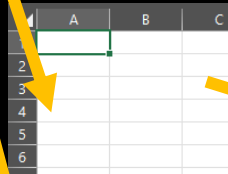
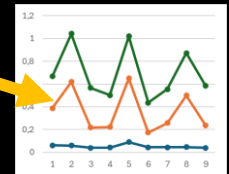
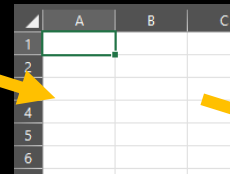
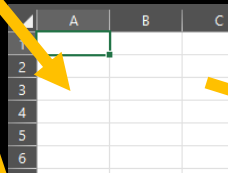
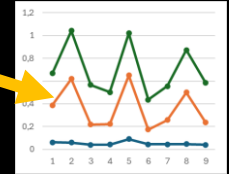
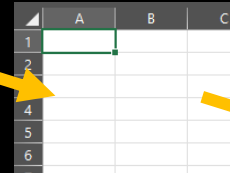
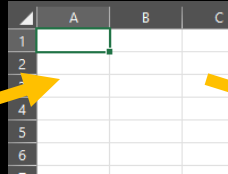
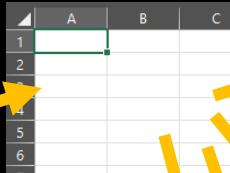


Why R and not Excel?

Scenario 3: Many plots needed



```
"genotype" "cell.width" "..."
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 38
"XA53" 40.9197598161176 75
"XA53" 33.1389955806546 NA
"control" 72.1092449936084 NA
"control" 35.888557069 NA
"XA53" 39.9640666861087 58
"XA53" 13.141525790561 15.11
"XA53" 15.0448761012405 23
"XA53" 47.0790477729402 53
"XA53" 81.999406393338 74.05
"XA53" 13.9409304767847 7
```



Why R and not Excel?

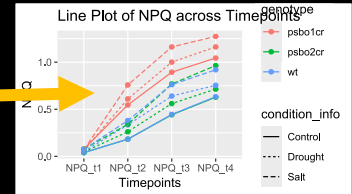


genotype	cell.width	
"XA53"	28.9213274233043	93
"XA53"	18.9921700965613	31
"XA53"	40.9197598161176	79
"XA53"	33.1389955806546	NA
"control"	72.1092449936094	
"control"	35.8885570000000	
"XA53"	39.864066681087	58
"XA53"	13.141525790561	15.111
"XA53"	15.0448761012405	27
"XA53"	47.079047729402	58
"XA53"	81.999406393338	74.052
"XA53"	13.8409304767847	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	NPQ
*XA53	28.9213274233043	9.0
*XA53	18.9921700965613	3.0
*XA53	40.9197598161176	7.5
*XA53	33.1389955806546	NA
*control	72.1092449936084	0.0
*control	35.888550691898	0.0
*XA53	39.8640666861087	5.0
*XA53	13.141525790561	15.111
*XA53	15.0448761012405	2.0
*XA53	47.079047729402	5.0
*XA53	81.999406393338	74.05
*XA53	13.8409304767847	7.0

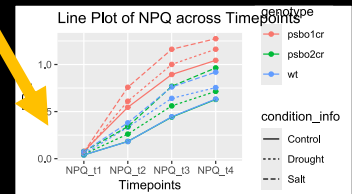
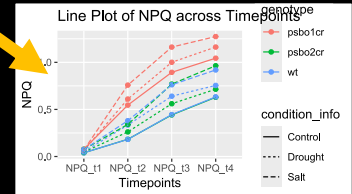
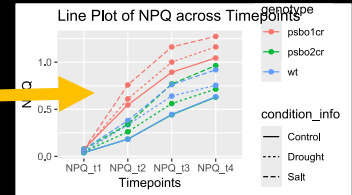
genotype	cell_width	NPQ
*XA53	28.9213274233043	9.0
*XA53	18.9921700965613	3.0
*XA53	40.9197598161176	7.5
*XA53	33.1389955806546	NA
*control	72.1092449936084	0.0
*control	35.888550691898	0.0
*XA53	39.8640666861087	5.0
*XA53	13.141525790561	15.111
*XA53	15.0448761012405	2.0
*XA53	47.079047729402	5.0
*XA53	81.999406393338	74.05
*XA53	13.8409304767847	7.0

genotype	cell_width	NPQ
*XA53	28.9213274233043	9.0
*XA53	18.9921700965613	3.0
*XA53	40.9197598161176	7.5
*XA53	33.1389955806546	NA
*control	72.1092449936084	0.0
*control	35.888550691898	0.0
*XA53	39.8640666861087	5.0
*XA53	13.141525790561	15.111
*XA53	15.0448761012405	2.0
*XA53	47.079047729402	5.0
*XA53	81.999406393338	74.05
*XA53	13.8409304767847	7.0

```
## {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
## "R_analysis/Orthogroups_Species_overlap_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" "N"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 3
"XA53" 40.9197598161176 79
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888550691898
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.111
"XA53" 15.0448761012405 27
"XA53" 47.079047729402 58
"XA53" 81.999406393338 74.05
"XA53" 13.8409304767847 7
```

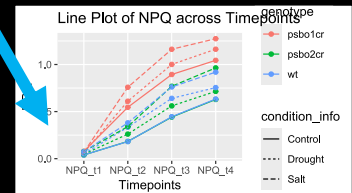
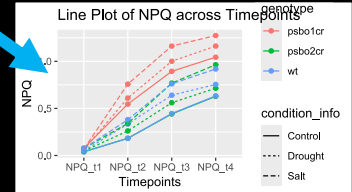
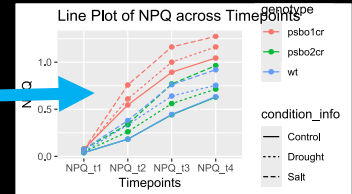
```
"genotype" "cell.width" "N"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 3
"XA53" 40.9197598161176 79
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888550691898
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.111
"XA53" 15.0448761012405 27
"XA53" 47.079047729402 58
"XA53" 81.999406393338 74.05
"XA53" 13.8409304767847 7
```

```
"genotype" "cell.width" "N"
"XA53" 28.9213274233043 98
"XA53" 18.9921700965613 3
"XA53" 40.9197598161176 79
"XA53" 33.1389955806546 NA
"control" 72.1092449936084
"control" 35.888550691898
"XA53" 39.8640666861087 58
"XA53" 13.141525790561 15.111
"XA53" 15.0448761012405 27
"XA53" 47.079047729402 58
"XA53" 81.999406393338 74.05
"XA53" 13.8409304767847 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_Species_overlap_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

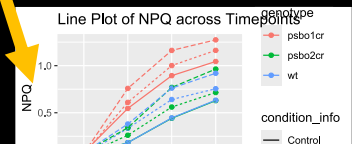
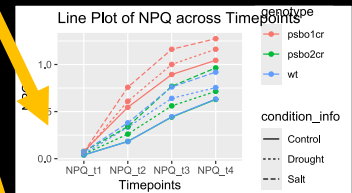
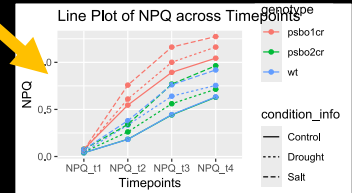
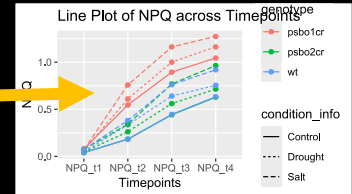


genotype	cell.width	
"XA53"	28.9213274233043	93
"XA53"	18.9921700965613	32
"XA53"	40.9197598161176	79
"XA53"	33.1389955806546	NA
"control"	72.1092449936094	
"control"	35.8885570000000	
"XA53"	39.8640666861087	58
"XA53"	13.141525790561	15.111
"XA53"	15.0448761012405	27
"XA53"	47.079047729402	53
"XA53"	81.999406393338	74.052
"XA53"	13.8409304767847	7

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

spec.overlap <- read.table(file = "orthofinder_res_2/Comparative_genomics_statistics/orthogroups_species_overlap.tsv")

## heatmap with values
pdf("R_analysis/Orthogroups_Species_overlap_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("b:/!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
roups_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
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]
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}
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")

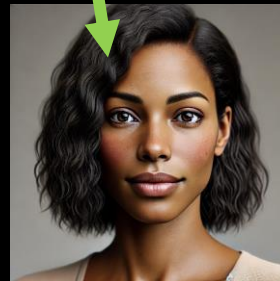
## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```



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")

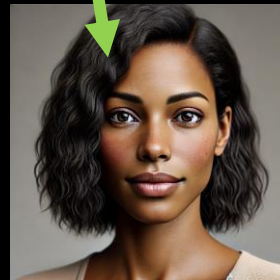
## heatmap with values
pdf ("R_analysis/Orthogroups_SpeciesOverlaps_heatmap.pdf", width=14, height=7,
onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```



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 =
```

Reproducibility

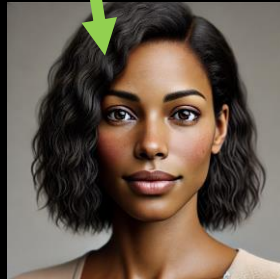
```
pdf ("R_analysis/orthogroups_speciesoverlap_heatmap.pdf", width=14, height=7, onefile = T)
par(mar = c(2, 12, 12, 2) + 0.1)
# input data
gdata <- spec.overlap
```



Me in 2 months



Me in 2 years



Collaborator



Paper reader



BY JAMES MONTGOMERY FLAGG

**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	10.233333	1007.22
psbo1cr	12.566666	71.56
psbo2cr	18.111111	516.33
wt	20.733333	1666.67
psbo1cr	23.166666	72.34

R Studio

Integrated development environment (IDE) for R

Script (enter commands here)

The screenshot displays the R Studio interface with the following components:

- Script Editor:** Contains R code for loading data, summarizing it, and creating a scatter plot.
- Console:** Shows the execution output of the code, including summary statistics for the 'diamonds' dataset.
- Workspace:** Lists the loaded data objects: 'diamonds' (53940 obs. of 10 variables), 'aveSize', 'clarity', and 'p'.
- Plots:** Displays a scatter plot titled 'Diamond Pricing' showing Price vs. Carat, colored by Clarity.

```
1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12           data=diamonds, color=clarity,
13           xlab="Carat", ylab="Price",
14           main="Diamond Pricing")
15
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	4.710	5.700	5.731	6.540	10.740
0.000	4.720	5.710	5.735	6.540	58.900
0.000	2.910	3.530	3.539	4.040	31.800

Summary of diamond pricing:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
326	2401	3933	5324	18820	

Workspace data

View results here

Help and Plots Viewer

Help in R Studio

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

The screenshot shows the R Studio interface. The source editor on the left contains R code for summarizing a dataset. A blue arrow points from the text 'Press F1' to the function name 'rownames' in the code. The console at the bottom left shows the R startup message. The right-hand pane is split into two sections: the top section shows an empty environment, and the bottom section shows the help window for the 'rownames' function, which includes a description and usage examples.

```
96 # Summary of the dataset
97 summary(plantData)
98
99 # What type of data do I have?
100 class(plantData)
101
102 # Column names of the dataset
103 colnames(plantData)
104
105 # Count the number of rows in your dataset
106 nrow(plantData)
107
108 # Printing whole dataset in the console (usable for small
109 datasets)
110 plantData
```

Environment: Environment is empty

R Documentation
Row and Column Names

Description
Retrieve or set the row or column names of a matrix-like object.

Usage

```
rownames(x, do.NULL = TRUE, prefix = "row")
rownames(x) <- value
```

```
colnames(x, do.NULL = TRUE, prefix = "col")
```

The help will open here

Where to write the code?

Console? Not good for reproducibility.

The screenshot shows the RStudio interface. The Source pane on the left contains R code for summarizing a dataset. The Console pane at the bottom shows the R startup message. A blue arrow points from the word 'Console' to the Console pane.

```
# Summary of the dataset
summary(plantData)

# what type of data do I have?
class(plantData)

# Column names of the dataset
colnames(plantData)

# Count the number of rows in your dataset
nrow(plantData)

# Printing whole dataset in the console (usable for small
# datasets)
print(plantData)
```

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.

R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' for how to cite R or R packages in publications.

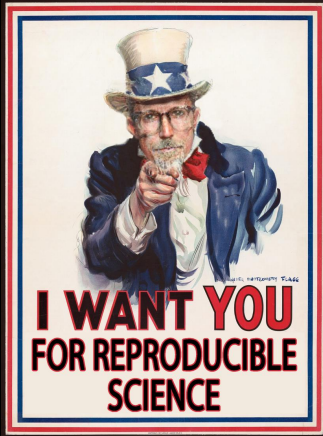
Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R.

> |

Console

Where to write the code?

Console? Not good for reproducibility.



Console



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Source Visual

```
96 # Summary of the dataset
97 summary(plantData)
98
99 # what type of data do I have?
100 class(plantData)
101
102 # Column names of the dataset
103 colnames(plantData)
104
105 # Count the number of rows in your dataset
106 nrow(plantData)
107
108 # Printing whole dataset in the console (usable for small
109 plantData
110
```

1033 Chunk 6: check your data - R Markdown

Console Terminal Background Jobs

R 4.2.1 ~ /

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.

R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' for 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.

>

Environment History Connections Tutorial

R - Global Environment

Environment is empty

Files Plots Packages Help Viewer Presenter

R: Row and Column Names - Find in Topic

row+colnames [base] R Documentation

Row and Column Names

Description

Retrieve or set the row or column names of a matrix-like object.

Usage

```
rownames(x, do.NULL = TRUE, prefix = "row")
rownames(x) <- value

colnames(x, do.NULL = TRUE, prefix = "col")
```

Where to write the code?

R Script / R Markdown

Source



The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main source editor window shows R code for a script named 'Evomics_2024_R_solution_ggplot.rmd'. The code includes comments and functions like `summary(plantData)`, `class(plantData)`, `colnames(plantData)`, and `nrow(plantData)`. The console at the bottom shows the R startup message, including the license and help information. On the right side, the Environment pane shows 'Global Environment' with an empty environment. Below it, the Documentation pane is open to the 'Row and Column Names' section, showing the `rownames` and `colnames` functions.

```
96 # Summary of the dataset
97 summary(plantData)
98
99 # what type of data do I have?
100 class(plantData)
101
102 # Column names of the dataset
103 colnames(plantData)
104
105 # Count the number of rows in your dataset
106 nrow(plantData)
107
108 # Printing whole dataset in the console (usable for small
109 datasets)
110 plantData
111
```

Console: R 4.2.1 ~ /

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.

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.

> |

Environment: Global Environment

Environment is empty

Row and Column Names

Description: Retrieve or set the row or column names of a matrix-like object.

Usage: `rownames(x, do.NULL = TRUE, prefix = "row")`
`rownames(x) <- value`
`colnames(x, do.NULL = TRUE, prefix = "col")`

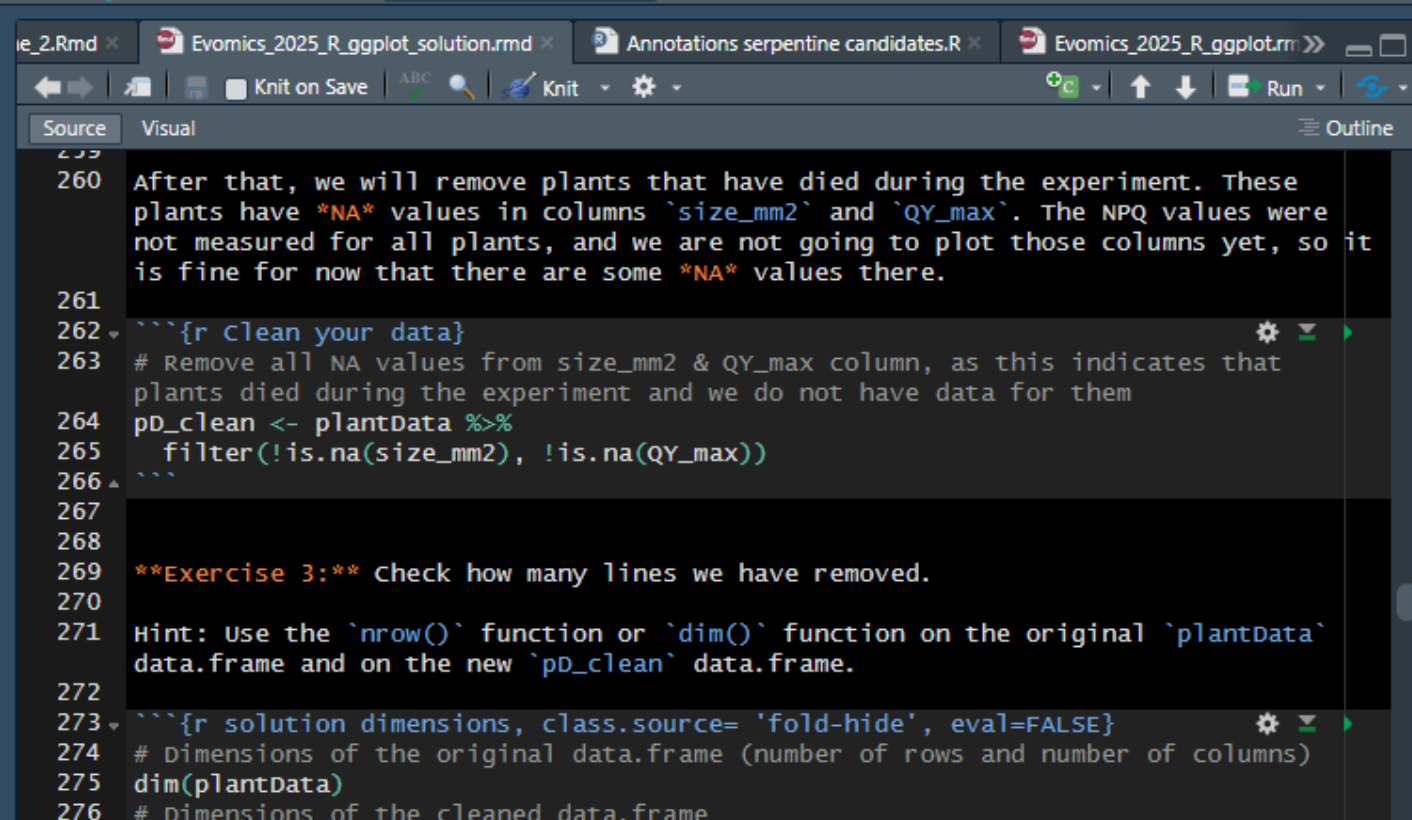
Where to write the code?

R Script: Code + # Comments

```
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````



The screenshot shows an R Markdown editor window with several tabs. The active tab is 'Evomics_2025_R_ggplot_solution.rmd'. The editor is in 'Source' view. The code chunk is as follows:

```
260 After that, we will remove plants that have died during the experiment. These
plants have *NA* values in columns `size_mm2` and `QY_max`. The NPQ values were
not measured for all plants, and we are not going to plot those columns yet, so it
is fine for now that there are some *NA* values there.
261
262 ```{r clean your data}
263 # Remove all NA values from size_mm2 & QY_max column, as this indicates that
plants died during the experiment and we do not have data for them
264 pd_clean <- plantData %>%
265   filter(!is.na(size_mm2), !is.na(QY_max))
266 ```
267
268
269 Exercise 3: check how many lines we have removed.
270
271 Hint: Use the `nrow()` function or `dim()` function on the original `plantData`
data.frame and on the new `pd_clean` data.frame.
272
273 ```{r solution dimensions, class.source= 'fold-hide', eval=FALSE}
274 # Dimensions of the original data.frame (number of rows and number of columns)
275 dim(plantData)
276 # Dimensions of the cleaned data.frame
```

R Markdown

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

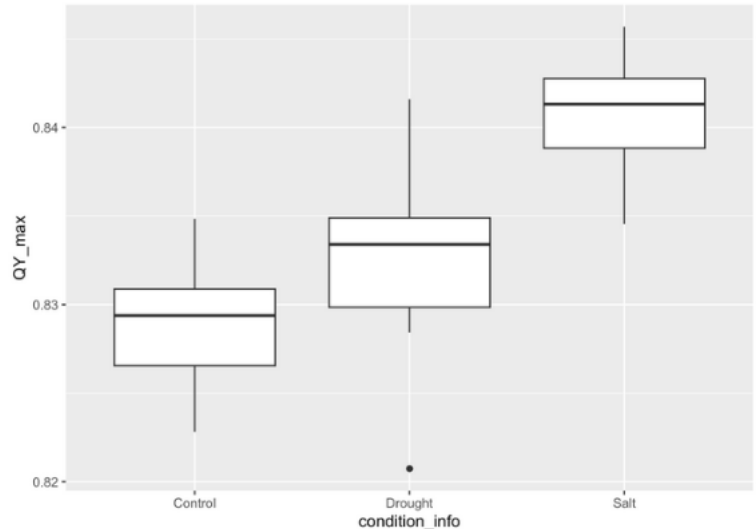
4.3 Modify your graph aesthetics

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 sequentially 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 markdown?

- .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!**

1. New File

2. Embed Code

3. Write Text

4. Set Output Format(s) and Options

5. Save and Render

6. Share

Annotations include: 'set preview location', 'insert code chunk', 'go to code chunk', 'run code chunk(s)', 'show outline', 'modify chunk options', 'run all previous chunks', 'run current chunk'.

3. Write Text

4. Set Output Format(s) and Options

Annotations include: 'add/edit attributes', 'style options', 'insert citations'.

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 to knit and then include
purf	TRUE	include or exclude a code chunk when extracting source code with <code>knitr::purf()</code>

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

RENDERED OUTPUT

Document Title

Author Name

- R Markdown
- Including Plots

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

summary(mtcars)

	mpg	cyl	displ
##	mean	11.4	101.4
##	1st Qu.	12.0	116.0
##	Median	15.0	168.0
##	Mean	15.4	147.9
##	3rd Qu.	19.0	266.0
##	Max.	44.0	473.0

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 Citations 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)`.

```
{r}
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.

"Italic" and "bold"
`superscript2`/`subscript2`
`--strikethrough--`
escaped: `\ \ \`
endash: `--`, emdash: `---`

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

Also end with a backslash! to make a new line.

Italic and bold
`superscript/subscript`
`strikethrough`
endash: `--`, emdash: `---`

Header 1

Header 2

Header 3

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="http://www.posit.co/">`
[This is a link.](link url)
[This is another link!]: [id]: link url

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

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

```
verbatim code
...
multiple lines of verbatim code
> block quotes
```

equation: `SeA∫a∫b + 1 = 0$`
equation block: `$$E = mc^2$$`
horizontal rule: `---`

HTML Tabssets

```
## Results (tabsets)
## Plots
text

## Tables more text
```

Results

Plots Tables

text



General data structures

- **Vector** - ordered collection of data

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

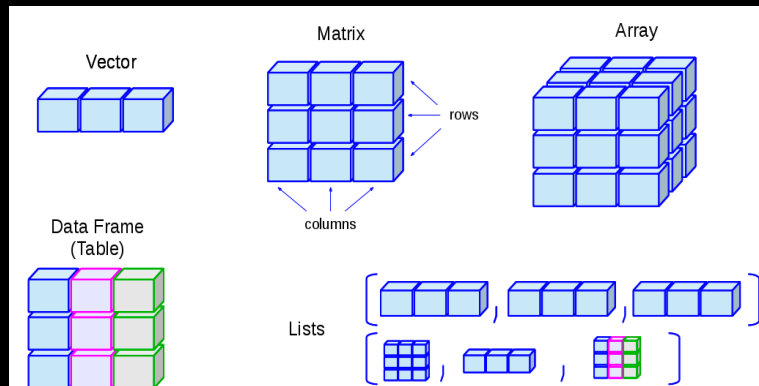
```
vector_2 <- c("potato", "lemonade", "avocado")
```

- **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 try R?

Arabidopsis thaliana mutants *psbo1* and *psbo2*

WT



psbo1

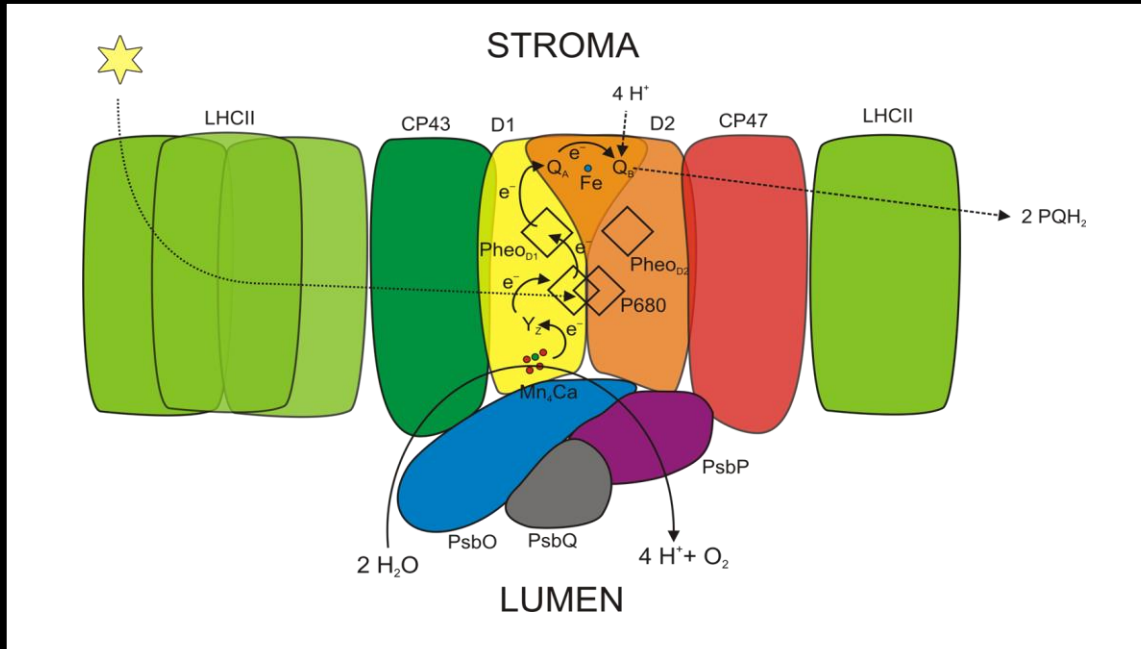


psbo2



PsbO protein

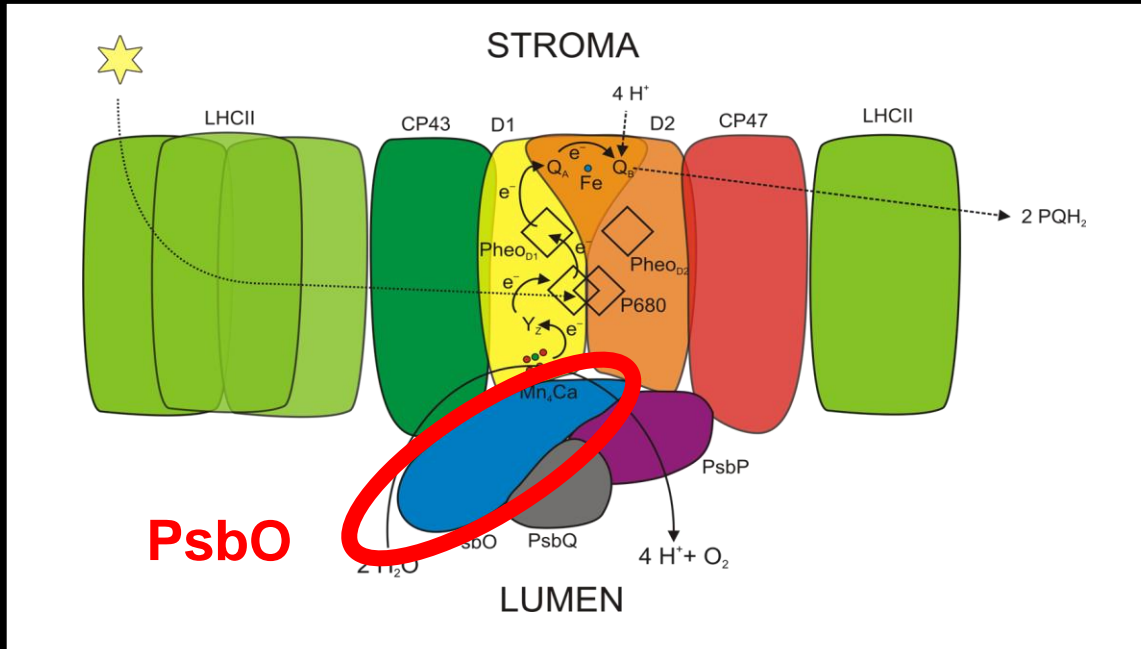
- Subunit of photosystem II
- Important for water splitting
- *Arabidopsis*: PsbO1 and PsbO2



Photosystem II

PsbO protein

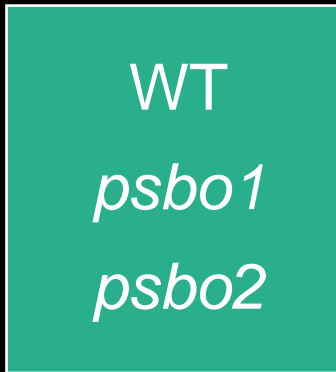
- Subunit of photosystem II
- Important for water splitting
- *Arabidopsis*: PsbO1 and PsbO2



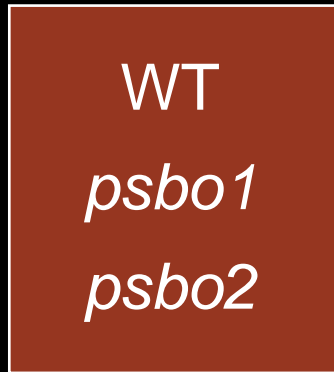
Photosystem II

Experimental design

Control

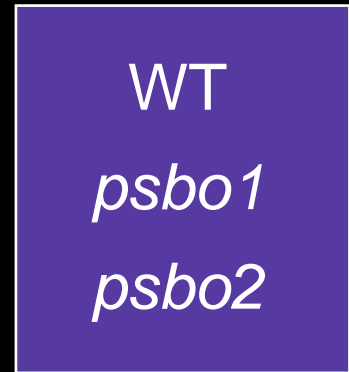


Drought



- water

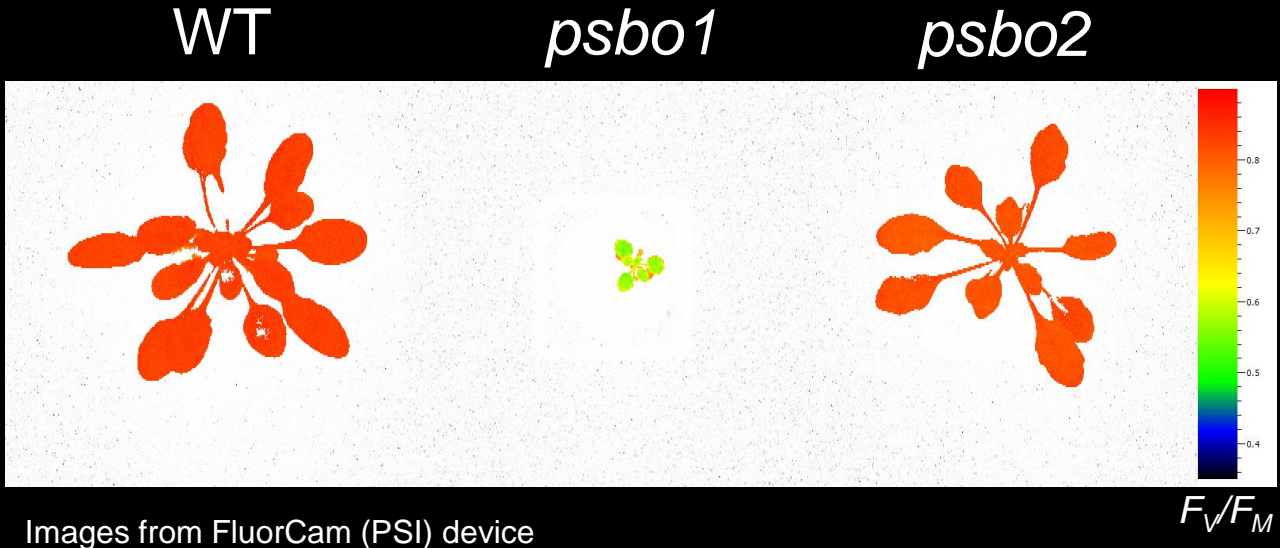
Salt stress



+ NaCl

Measurement – chlorophyll fluorescence

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



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.