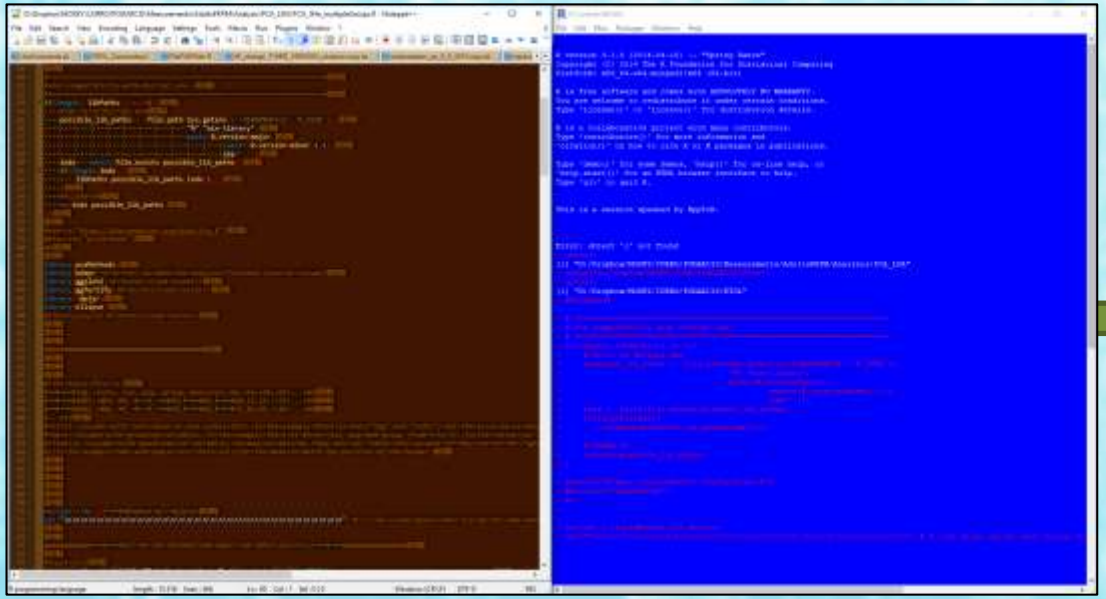
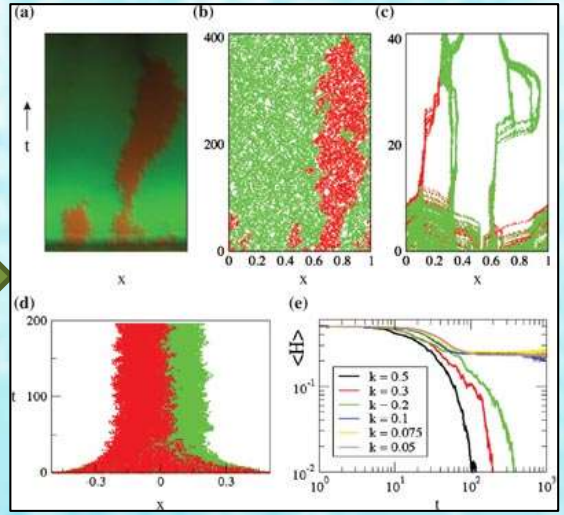
Use of R environment in **Evolutionary Ecology**





Why R??

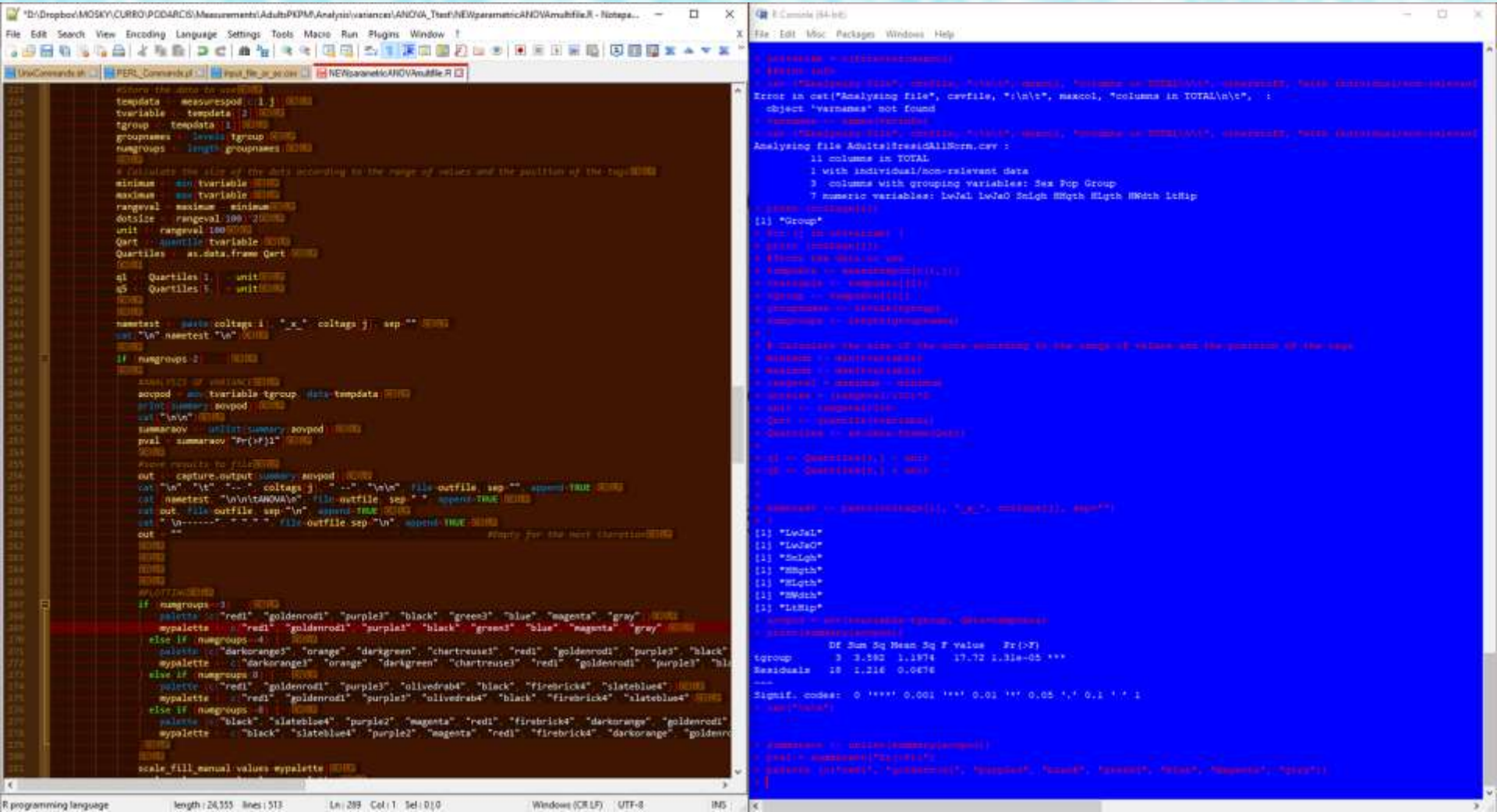
PROS

- Beautiful plots
- Big data
- Loop through files and analysis
- Repeat same analysis any time
- Open free language = big online community to help

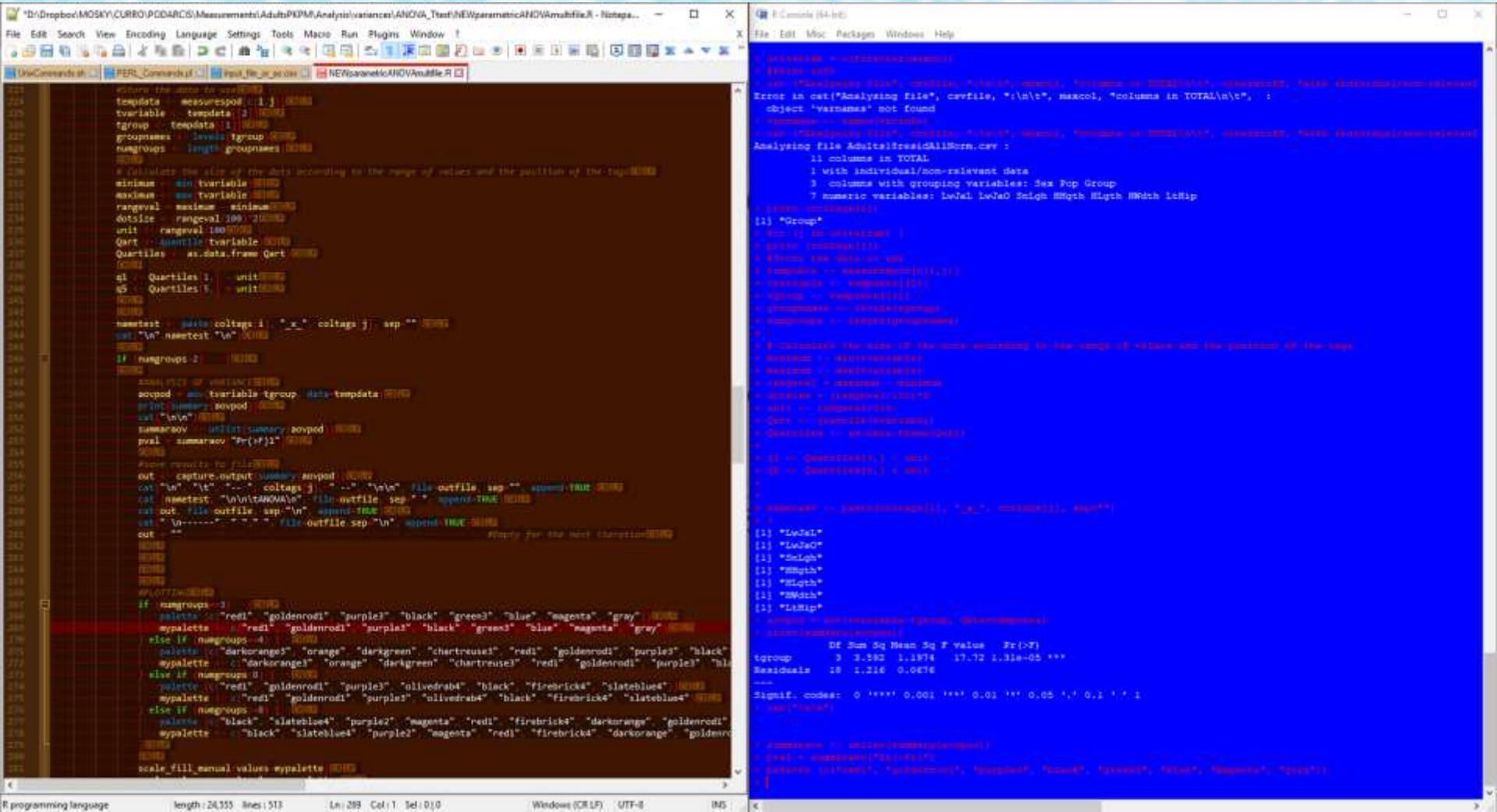
CONS

- Not pushing buttons to get quick results*
- Learn to code*

- **CONS**
- Not pushing buttons to get quick results
- Learn to code

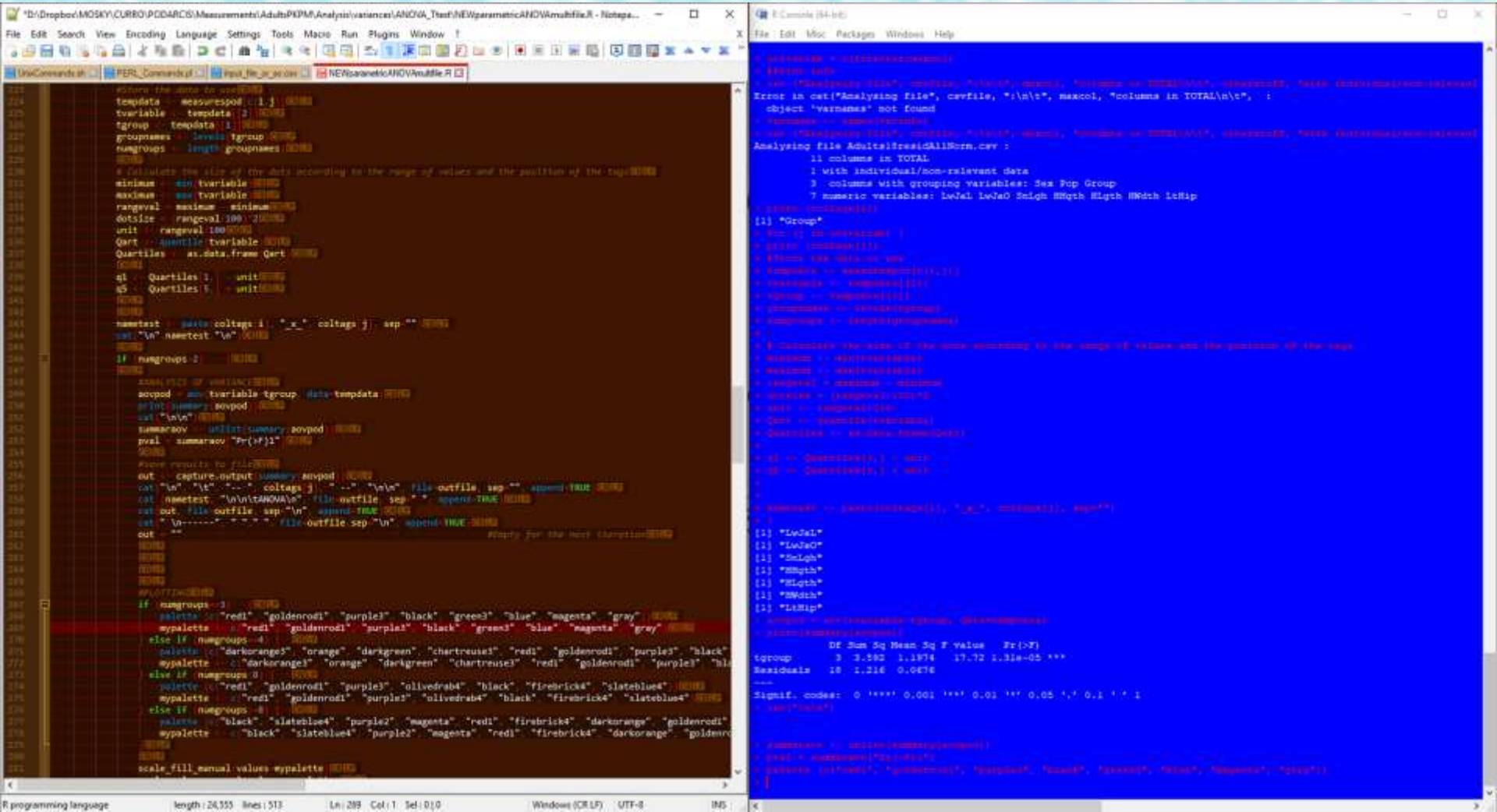


- **PROS**
- Understanding what we are doing
- Learn how to code!



We are not goint to jump blindly into a bunch of code.

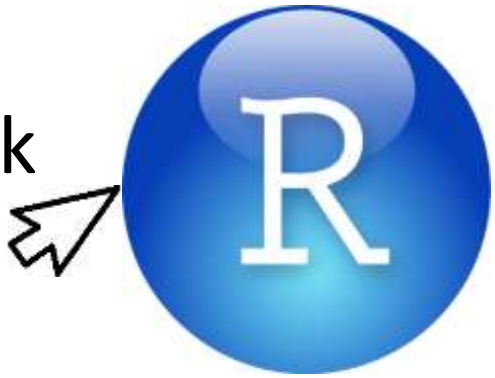
Baby steps



Basic R: How? Where?

RStudio: The Best way to start in R

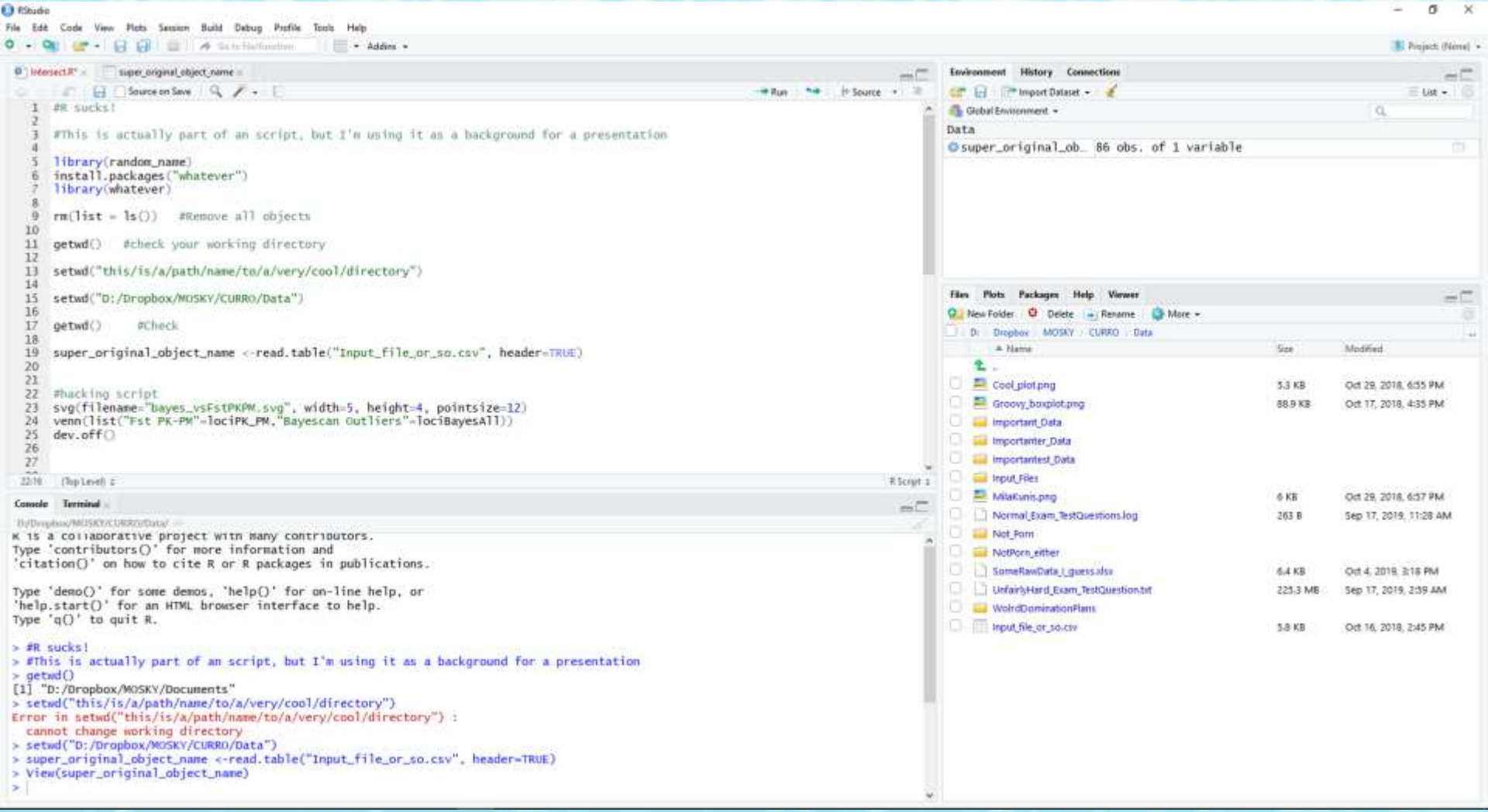
Double click



Basic R : How? Where?

Script (commands you want to run)

objects (data) loaded in R Environment



Console > coded executed already and output messages (including errors)

Directories and files in your computer

Basic R : How? Where?

File > New File > R Script (Ctrl+Shift+N)

Write something nice!

Comment #

 # Notes for yourself that will be ignored by R
 # you should define the purpose of the script
 # kind of input file and describe each step

Basic R

Basic commands

Objects: any data saved in R environment

many kinds of objects can be saved in R memory
variables, vectors, dataframes...

variables
Information stored with a unique name
numerical values or text (string)

Basic R

Basic commands

Declare **variables:** = <-

~~3 <- three~~

~~*three -> 3*~~

two <- 2

three = 3

variable = "text string"

Execute with **Ctrl+Enter**

and will appear at "Environment"

Basic R

Basic commands

R does statistics and basic math

Operators: + - * / ^

Basic R

Basic commands

a = 2

b = 10*4

sum_ab = a+b

```
> a=10*4
> b=2
> sum_ab=a+b
> sum_ab
[1] 42
>
```

you can call an **object** any time

sum_ab *#42*

Basic R

The first rule of the coding club:

"If you are typing a lot your are doing it wrong"

Use 'Tab':

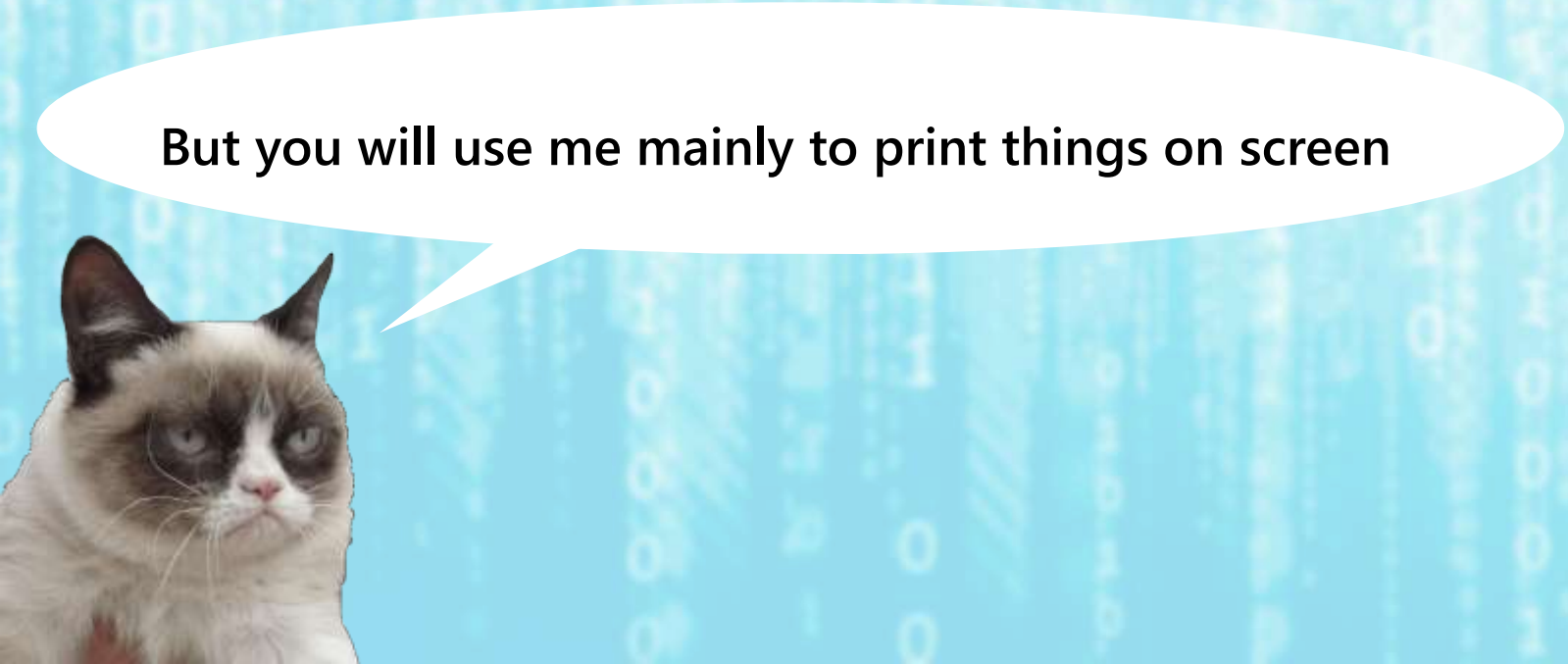
(autocomplete)

Basic R

Basic commands

- **First command**: cat()

It is a powerful tool to con**cat**enate and write files.



Basic R

Basic commands: cat

Try to use cat() to print something in your console

```
cat(Hello World!)
```

Ctrl+Enter

Basic R

Basic commands: cat

Tell R something is text by using "quotation marks"

```
cat("Hello World!")
cat(sum_ab)
```

Basic R

Basic commands: cat

- Commands may have multiple arguments
- Arguments are separated by commas ","
- Concatenate various arguments with cat()

```
cat("The answer is", sum_ab )
```

Basic R

Basic commands: cat

Arguments can modify commands behaviour

sep: how to separate arguments when printing

\n <- will print each argument in a different line

```
cat("The answer is", sum_ab, sep="\n" )
```

Basic R

Basic commands

Summarysing

- commands() can have many arguments
- arguments modify command behaviour
- arguments are separated by commas,
- text is defined with "quotation marks"

```
cat("The answer is", sum_ab, sep="\n" )
```

Basic R

Basic commands

Commands have help information

check other arguments for cat:

?cat

help(cat)

Basic R

Let's analyse our first data in R!

https://tinyurl.com/evolecopract

- Download excel file into a new folder
- Open Excel file

Basic R

Dataset

Head and body measurements from lizards (*Podarcis sicula*) from two Adriatic islands

Basic R

Excel file

Variables in columns

Observations in rows

	CODE	Sex	Popu	Group	Alive	procesed	Spots	JawLe	HeadWi	TailLen	BodyLength
2	PK021	fem	PK	PK-F	yes	TRUE	3	1.392	0.899	9.438	6.150
3	PK022	fem	PK	PK-F	no	TRUE	5	1.290	0.855	9.319	6.340
4	PK023	fem	PK	PK-F	yes	TRUE	6	1.277	0.849	1.060	5.894
5	PK024	fem	PK	PK-F	yes	TRUE	4	1.315	0.852	8.727	5.848
6	PK025	fem	PK	PK-F	yes	TRUE	4	1.408	0.933	6.938	6.367
7	PK026	fem	PK	PK-F	yes	TRUE	4	1.206	0.856	8.804	5.994
8	PK027	fem	PK	PK-F	yes	TRUE	2	1.299	0.830	7.260	6.175
9	PK028	fem	PK	PK-F	no	FALSE	2	1.365	0.897	8.086	6.234
10	PK029	fem	PK	PK-F	no	TRUE	2	1.385	0.912	7.849	6.043
11	PK030	fem	PK	PK-F	yes	FALSE	3	1.282	0.836	0.933	5.573
12	PK031	fem	PK	PK-F	no	TRUE	6	1.122	0.873	6.498	6.053
13	PK032	fem	PK	PK-F	yes	TRUE	5	1.295	0.856	6.462	5.929
14	PK033	fem	PK	PK-F	no	TRUE	9	1.336	0.888	8.849	6.220
15	PK034	fem	PK	PK-F	no	TRUE	3	1.297	0.821	9.233	6.452
16	PK035	fem	PK	PK-F	yes	TRUE	2	1.345	0.871	2.920	6.642
17	PK036	fem	PK	PK-F	yes	TRUE	1	1.234	0.858	2.299	5.662
18	PK037	fem	PK	PK-F	yes	FALSE	5	1.327	0.860	6.129	5.739
19	PK038	fem	PK	PK-F	no	FALSE	5	1.359	0.953	4.977	6.264
20	PK039	fem	PK	PK-F	no	FALSE	6	1.358	0.929	8.960	6.166
21	PK040	fem	PK	PK-F	yes	TRUE	3	1.442	0.942	7.592	6.362
22	PK041	fem	PK	PK-F	no	FALSE	5	1.281	0.892	8.352	6.588
23	PK042	fem	PK	PK-F	yes	TRUE	6	1.359	0.894	8.221	5.881
24	PK043	fem	PK	PK-F	no	FALSE	3	1.311	0.928	8.949	6.311
25	PK044	fem	PK	PK-F	yes	TRUE	4	1.425	0.905	1.922	6.380
26	PK045	fem	PK	PK-F	no	FALSE	6	1.402	0.926	9.174	6.569
27	PK046	fem	PK	PK-F	no	TRUE	4	1.372	0.903	7.839	6.186
28	PK001	mal	PK	PK-M	yes	TRUE	5	1.437	0.994	8.664	6.007
29	PK003	mal	PK	PK-M	no	TRUE	2	1.508	1.091	6.943	6.748
30	PK004	mal	PK	PK-M	yes	TRUE	3	1.608	1.094	1.559	6.456
31	PK005	mal	PK	PK-M	no	TRUE	1	1.405	1.010	8.467	6.024

Export from Excel

- File>Save As: comma separated (.csv)

Basic R

- **Browse files in R**: Working Directory

getwd()

To check the working directory (where R is reading your files from)

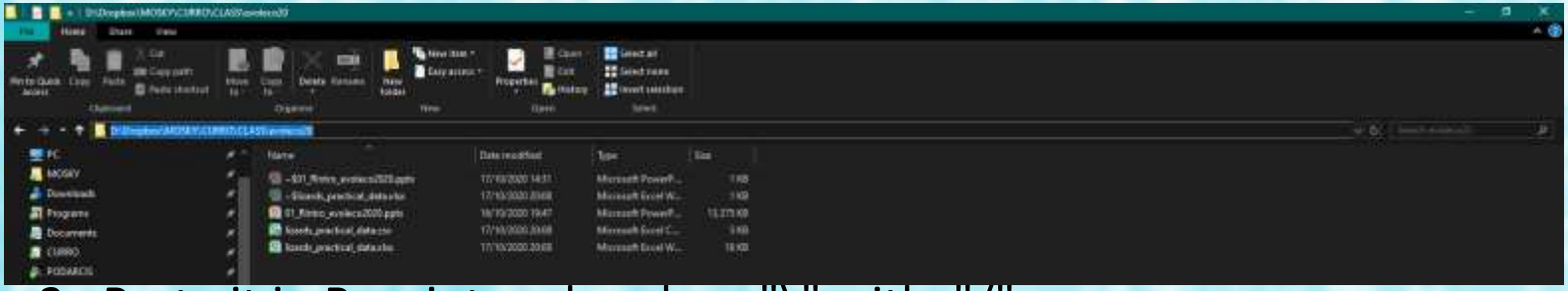
setwd()

To tell R where to find the files to analyse

Basic R

setwd("path/")

1 . Copy the path of the folder where you file is at



2 . Paste it in R script and replace "\" with "/"

setwd("D:/Dropbox/MOSKY/CURRO/CLASS/evoleco20")

getwd() *# check wd*

Basic R

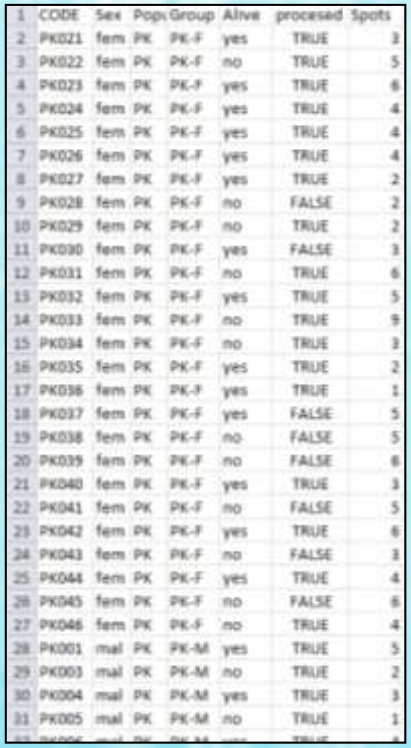
- **Import files to R**

read.table()

Check **?read.table** to see arguments

You need three arguments

- "file name"
- separator of columns (sep=",")
- header=? (TRUE/FALSE)



	CODE	Sex	Pop	Group	Alive	processed	Spots
2	PK021	fem	PK	PK-F	yes	TRUE	3
3	PK022	fem	PK	PK-F	no	TRUE	5
4	PK023	fem	PK	PK-F	yes	TRUE	6
5	PK024	fem	PK	PK-F	yes	TRUE	4
6	PK025	fem	PK	PK-F	yes	TRUE	4
7	PK026	fem	PK	PK-F	yes	TRUE	4
8	PK027	fem	PK	PK-F	yes	TRUE	2
9	PK028	fem	PK	PK-F	no	FALSE	2
10	PK029	fem	PK	PK-F	no	TRUE	2
11	PK030	fem	PK	PK-F	yes	FALSE	3
12	PK031	fem	PK	PK-F	no	TRUE	6
13	PK032	fem	PK	PK-F	yes	TRUE	5
14	PK033	fem	PK	PK-F	no	TRUE	9
15	PK034	fem	PK	PK-F	no	TRUE	3
16	PK035	fem	PK	PK-F	yes	TRUE	2
17	PK036	fem	PK	PK-F	yes	TRUE	1
18	PK037	fem	PK	PK-F	yes	FALSE	5
19	PK038	fem	PK	PK-F	no	FALSE	5
20	PK039	fem	PK	PK-F	no	FALSE	6
21	PK040	fem	PK	PK-F	yes	TRUE	3
22	PK041	fem	PK	PK-F	no	FALSE	5
23	PK042	fem	PK	PK-F	yes	TRUE	6
24	PK043	fem	PK	PK-F	no	FALSE	3
25	PK044	fem	PK	PK-F	yes	TRUE	4
26	PK045	fem	PK	PK-F	no	FALSE	6
27	PK046	fem	PK	PK-F	no	TRUE	4
28	PK001	mal	PK	PK-M	yes	TRUE	5
29	PK003	mal	PK	PK-M	no	TRUE	2
30	PK004	mal	PK	PK-M	yes	TRUE	3
31	PK005	mal	PK	PK-M	no	TRUE	1

Basic R

- **Import files to R**

```
read.table( "lizards_practical_data.csv",
        header=TRUE,
        sep=",")
```

This way we see it, but can't we save it in R environment?

Basic R

Basic commands

Save files to R environment

You can redirect the output of most commands to an object

```
lizards <- read.table(  ... )
```

```
read.table( "lizards_practical_data.csv",
        header=TRUE,
        sep=",")
```

Basic R

Check your data!!

- Read the description in 'Environment'
- head(lizards)
- tail(lizards)
- str(lizards)

Basic R

ERRORS?

Export from Excel

File>Options>Advaced

In "Editing options"

Uncheck: □ Use system separators

Decimal separator [.]

Thousands separator [,]

File>Save As: comma separated (.csv)

lizards = read.table(. . .)

Basic R

ERRORS?

Export from Excel

Open file, replace all "**,**" with "**.**"

Save

Let's go again:

```
lizards = read.table( "lizards_practical_data.csv",
        header=TRUE,
        sep=";")
```

Basic R

Check your data!!

- Read the description in 'Environment'
  86 observarions (individuals)
- head(lizards)
- tail(lizards)
- **str(lizards)**

Basic R

Check your data again!!

str(lizards)

data frame organisation: $

Each column from our excel is a different variable stored in a different $ slot

lizards$Sex

```
'data.frame':   86 obs. of  11 variables:
 $ CODE       : chr  "PK021" "PK022" "PK023" "PK024" ...
 $ Sex        : chr  "fem" "fem" "fem" "fem" ...
 $ Population: chr  "PK" "PK" "PK" "PK" ...
 $ Group      : chr  "PK-F" "PK-F" "PK-F" "PK-F" ...
 $ Alive      : chr  "yes" "no" "yes" "yes" ...
 $ procesed   : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ Spots      : int  3 5 6 4 4 4 2 2 2 3 ...
 $ JawLength : num  1.39 1.29 1.28 1.31 1.41 ...
 $ HeadWidth : num  0.899 0.855 0.849 0.852 0.933 0.856 0.83 0.897 0.912 0.836 ...
 $ TailLength: num  9.44 9.32 1.06 8.73 6.94 ...
 $ BodyLength: num  6.15 6.34 5.89 5.85 6.37 ...
>
```

Basic R

R differentiates some types of variables

Sex and Population "chr"

```
$ Sex       : chr  "fem" "fem" "fem" "fem" ...
$ Population: chr  "PK" "PK" "PK" "PK" ...
```

- Are categorical variables
- We will use them to define groups
- In most analysis they are used as "grouping", "treatment" or "explanatory variables"

Basic R

R differentiates some types of variables

Tail Length and Body Length "num"

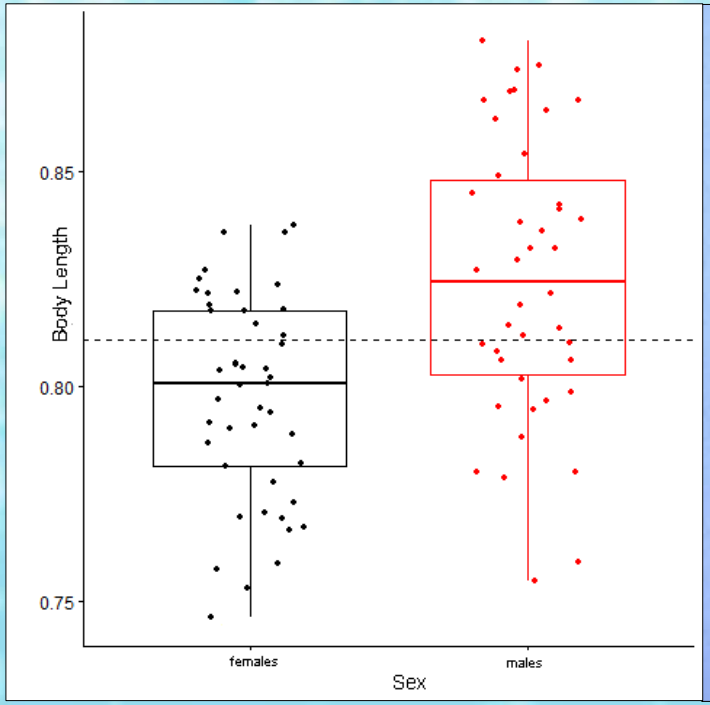
```
$ TailLength: num  9.44 9.32 1.06 8.73 6.94 ...
$ BodyLength: num  6.15 6.34 5.89 5.85 6.37 ...
```

- Continuous numerical variables
- In mostly are used as "response variables"
- We will analyse if their values are different among groups

Basic R

Let's analyse the data

Is there significant differences in Body Lenght between males and females?



Basic R

Let's analyse the data

Is there significant differences in Body Lenght between males and females?

aov() = **an**alysis **o**f **va**riance = ANOVA

2 arguments

formula: response variable ~ grouping variable

data frame name: lizards

Basic R

Let's analyse the data

ANOVA

aov(BodyLength~Sex**,** data=lizards**)**

Summarice: summary()

summary(aov(. . .))

```
   Pr(>F)
1.91e-05 ***
```

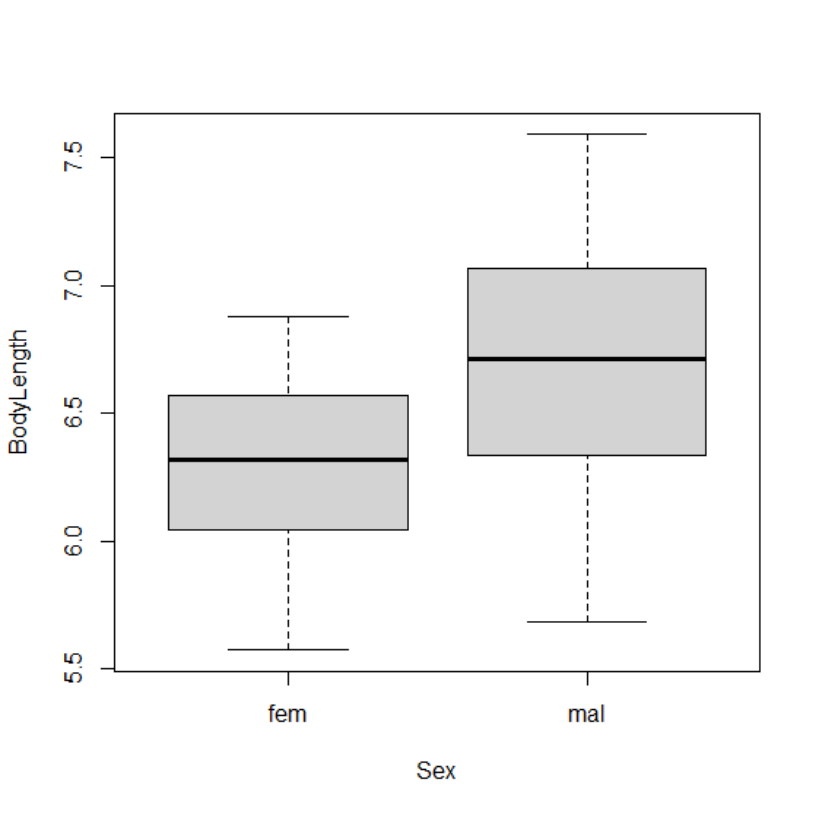
Save for later

results_aov <- summary(aov(. . .))

Basic R

BASIC PLOT

boxplot(BodyLength~Sex, data=lizards)



Basic R

EXERCISE

Now try to do the same for Head Width and Populations

Are there significant differences in Head Width between Populations?

You will need:

aov(), summary(), and boxplot()

Basic R

save results to a file

First check format

str(results_aov)

This is ugly

```
> str(results_aov)
List of 1
 $ :Classes 'anova' and 'data.frame':   2 obs. of  5 variables:
  ..$ Df     : num [1:2] 1 84
  ..$ Sum Sq : num [1:2] 3.76 15.37
  ..$ Mean Sq: num [1:2] 3.761 0.183
  ..$ F value: num [1:2] 20.5 NA
  ..$ Pr(>F) : num [1:2] 1.91e-05 NA
 - attr(*, "class")= chr [1:2] "summary.aov" "listof"
```

We actually want it as the output that appears in Console

```
> results_aov
            Df Sum Sq Mean Sq F value   Pr(>F)
Sex          1  3.761   3.761   20.55 1.91e-05 ***
Residuals   84 15.374   0.183
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Basic R

save results to a file

1.- capture output as seen on screen: capture.output

```
results_aov <- summary(aov( ... ))

capture.output(results_aov)

print_aov <- capture.output(results_aov)
```

2.- print to a file: 'cat' with argument "file" and "sep"

```
cat(print_aov, sep="\n", file="aov_out.txt")
```