Introduction to R programming

Anand Osuri (Nature Conservation Foundation)
Akshay Surendra (Yale University)

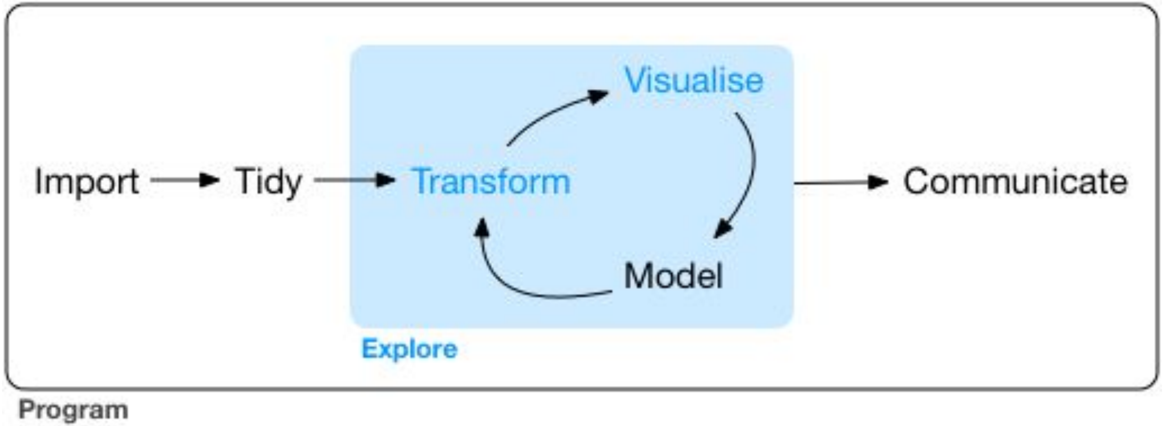
What is R?

A computer software for working with data

A computer programming language

Free and open source

https://en.wikipedia.org/wiki/R_(programming_language)



Source: Wickham & Grolemund (2017)
https://r4ds.had.co.nz/

Objectives of the course

Objectives of the course

Familiarity with the R environment and components

Objectives of the course

Familiarity with the R environment and components

Data and programming essentials

Objectives of the course

Familiarity with the R environment and components

Data and programming essentials

Self sufficiency

Course structure

TWO credits

Course structure

TWO credits

11 Sep to early October

Four weeks with two interactions per week -- Wednesday AM and Friday AM

Course structure

TWO credits

11 Sep to early October

Four weeks with two interactions per week -- Wednesday AM and Friday AM

‘Flipped’ class -- Self-study tutorials and worksheets (four) followed by discussion sessions (with smaller groups)

Course structure

TWO credits

11 Sep to early October

Four weeks with two interactions per week -- Wednesday AM and Friday AM

‘Flipped’ class -- Self-study tutorials and worksheets (four) followed by discussion sessions (with smaller groups)

Continuous assessment based on worksheets (80%) + end of course quiz or short project (20%)

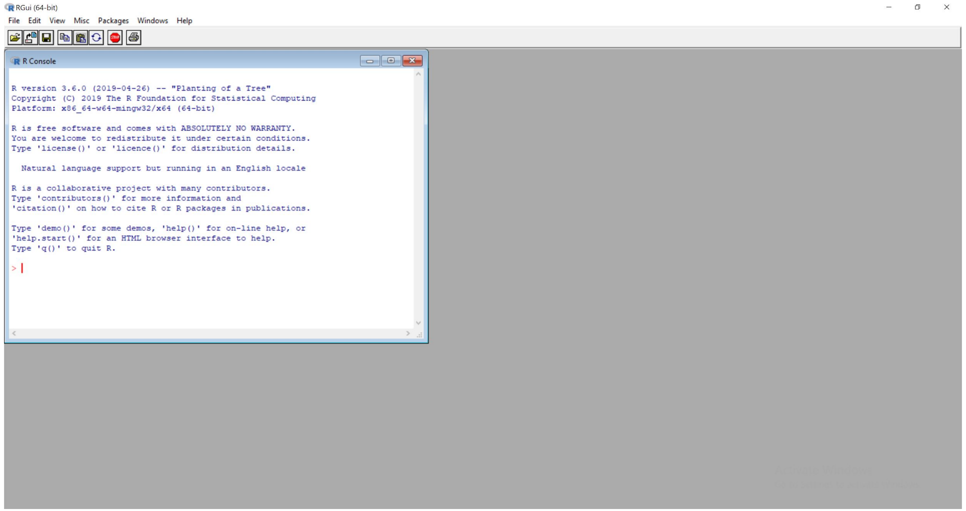
RStudio®

tidyverse

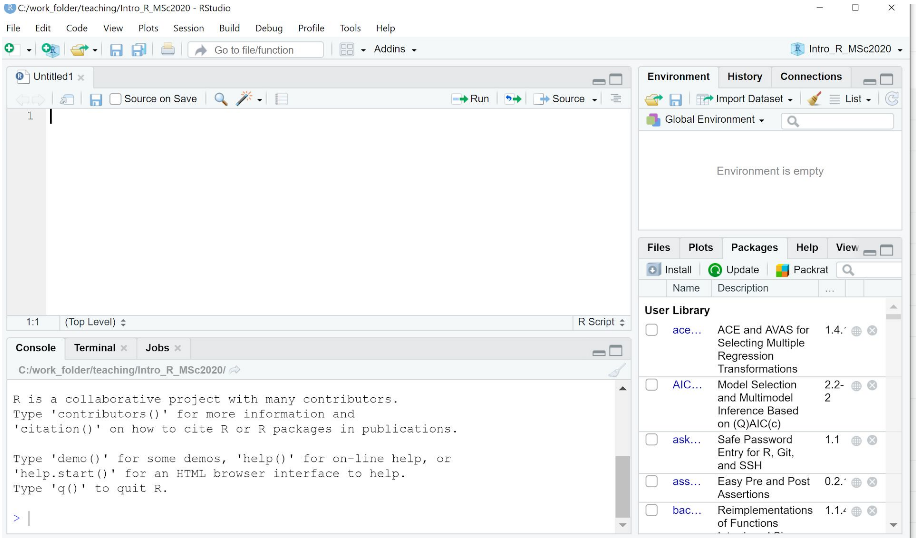
www.rstudio.com

“...makes data science faster, easier and more fun”

Has everyone installed R and RStudio?



```
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

>
```

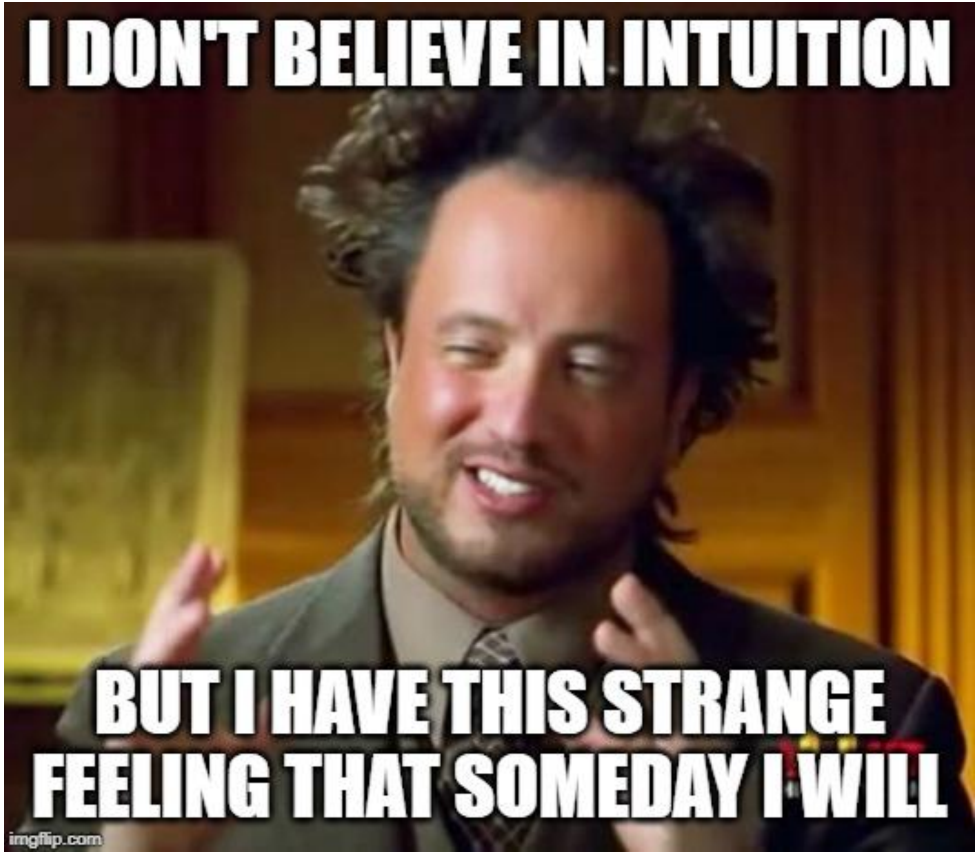


C:/work_folder/teaching/Intro_R_MSc2020/

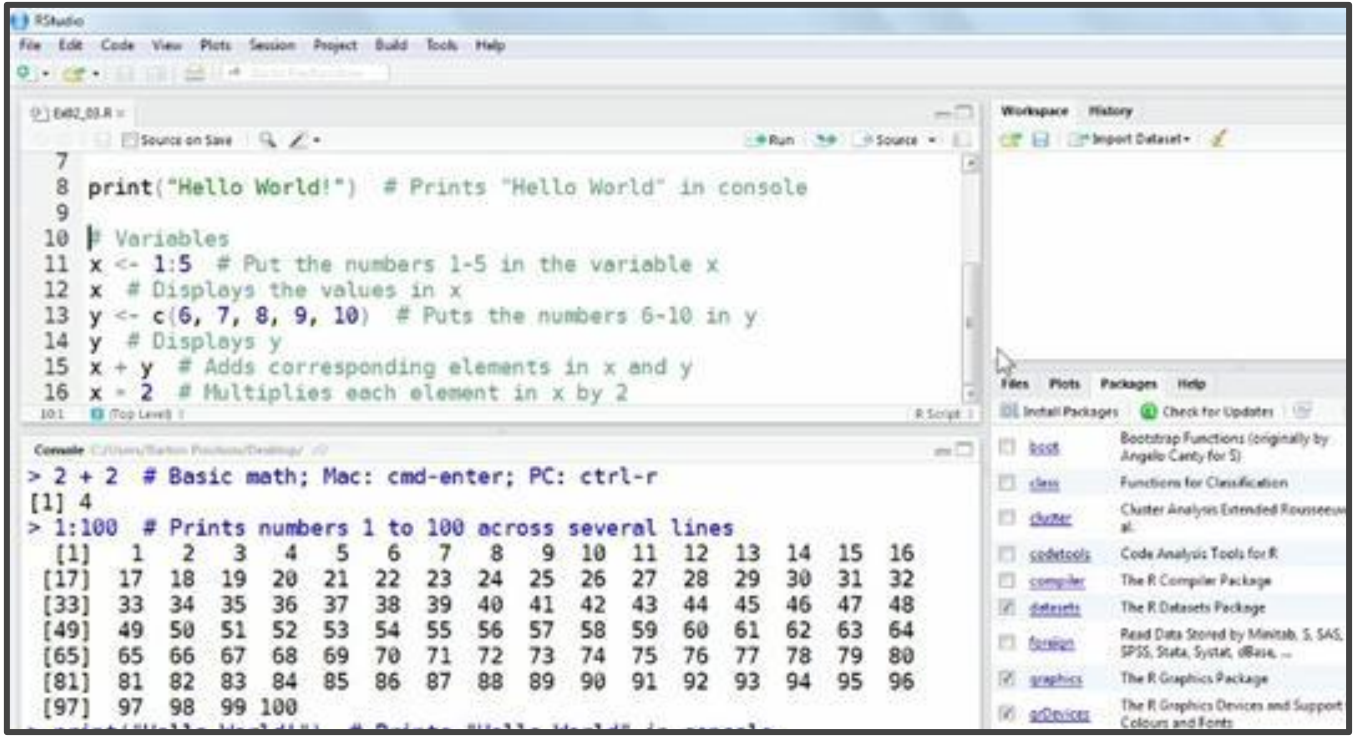
```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

>
```

	Name	Description	...
	User Library		
□	ace…	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.'
□	AIC…	Model Selection and Multimodel Inference Based on (Q)AIC(c)	2.2-2
□	ask…	Safe Password Entry for R, Git, and SSH	1.1
□	ass…	Easy Pre and Post Assertions	0.2.'
□	bac…	Reimplementations of Functions	1.1.‹

An intuition for R through an analogy


I DON'T BELIEVE IN INTUITION
BUT I HAVE THIS STRANGE FEELING THAT SOMEDAY I WILL
imgflip.com

An R environment seen through R Studio -

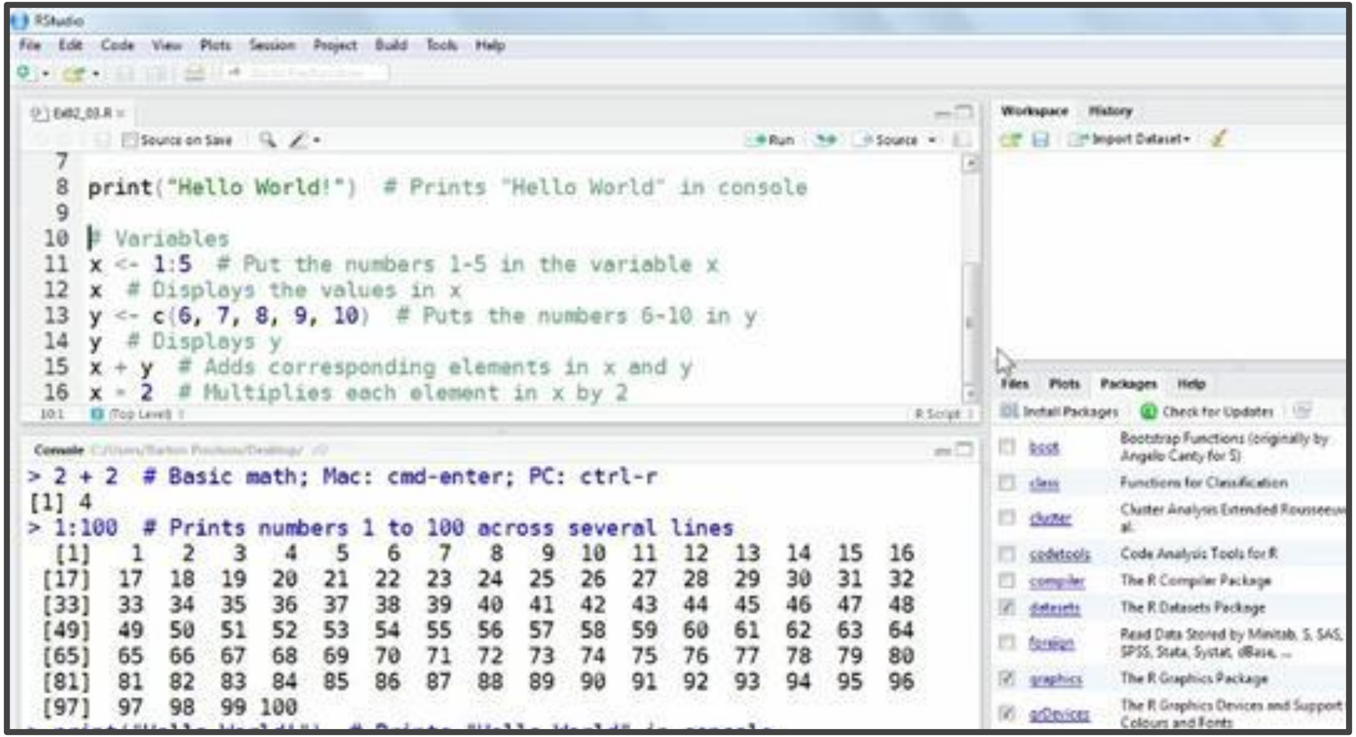


```
print("Hello World!")  # Prints "Hello World" in console

# Variables
x <- 1:5  # Put the numbers 1-5 in the variable x
x  # Displays the values in x
y <- c(6, 7, 8, 9, 10)  # Puts the numbers 6-10 in y
y  # Displays y
x + y  # Adds corresponding elements in x and y
x * 2  # Multiplies each element in x by 2
```

```
> 2 + 2  # Basic math; Mac: cmd-enter; PC: ctrl-r
[1] 4
> 1:100  # Prints numbers 1 to 100 across several lines
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
 [17]  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
 [33]  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48
 [49]  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
 [65]  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
 [81]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
 [97]  97  98  99 100
> print("Hello World!")  # Prints "Hello World" in console
```

□	boot	Bootstrap Functions (originally by Angelo Canty for S)
□	class	Functions for Classification
□	cluster	Cluster Analysis Extended Rousseeuw et al.
□	codetools	Code Analysis Tools for R
□	compiler	The R Compiler Package
☑	datasets	The R Datasets Package
□	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
☑	graphics	The R Graphics Package
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts

- is like an android OS on your phone



```
print("Hello World!")  # Prints "Hello World" in console

# Variables
x <- 1:5  # Put the numbers 1-5 in the variable x
x  # Displays the values in x
y <- c(6, 7, 8, 9, 10)  # Puts the numbers 6-10 in y
y  # Displays y
x + y  # Adds corresponding elements in x and y
x * 2  # Multiplies each element in x by 2
```

```
> 2 + 2  # Basic math; Mac: cmd-enter; PC: ctrl-r
[1] 4
> 1:100  # Prints numbers 1 to 100 across several lines
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
 [17]  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
 [33]  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48
 [49]  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
 [65]  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
 [81]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
 [97]  97  98  99 100
> print("Hello World!")  # Prints "Hello World" in console
```

□	boot	Bootstrap Functions (originally by Angelo Canty for S)
□	class	Functions for Classification
□	cluster	Cluster Analysis Extended Rousseeuw et al.
□	codetools	Code Analysis Tools for R
□	compiler	The R Compiler Package
☑	datasets	The R Datasets Package
□	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
☑	graphics	The R Graphics Package
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts

An android system has a bunch of apps, and each of these apps perform certain sets of tasks

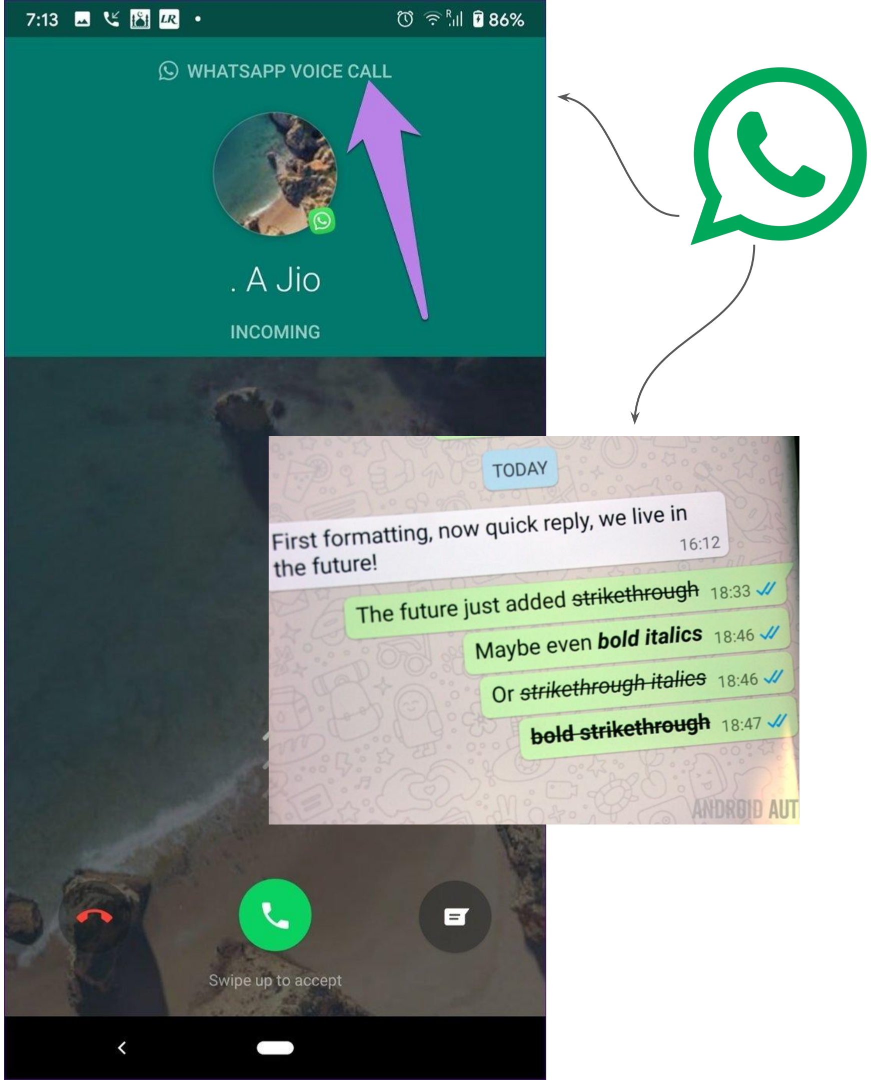
Similarly, R bundles together tasks into packages (like apps)

dplyr
forcats
ggplot2
readr
readxl
stringr

haven
lubridate
%>%
Ceci n'est pas un pipe.
TIBBLE
tidyr
tidyverse

Consider an app (~package), say *Whatsapp*

Two of the many tasks you can do on this app (~package) are *texting()* and *calling()*



First formatting, now quick reply, we live in the future! 16:12

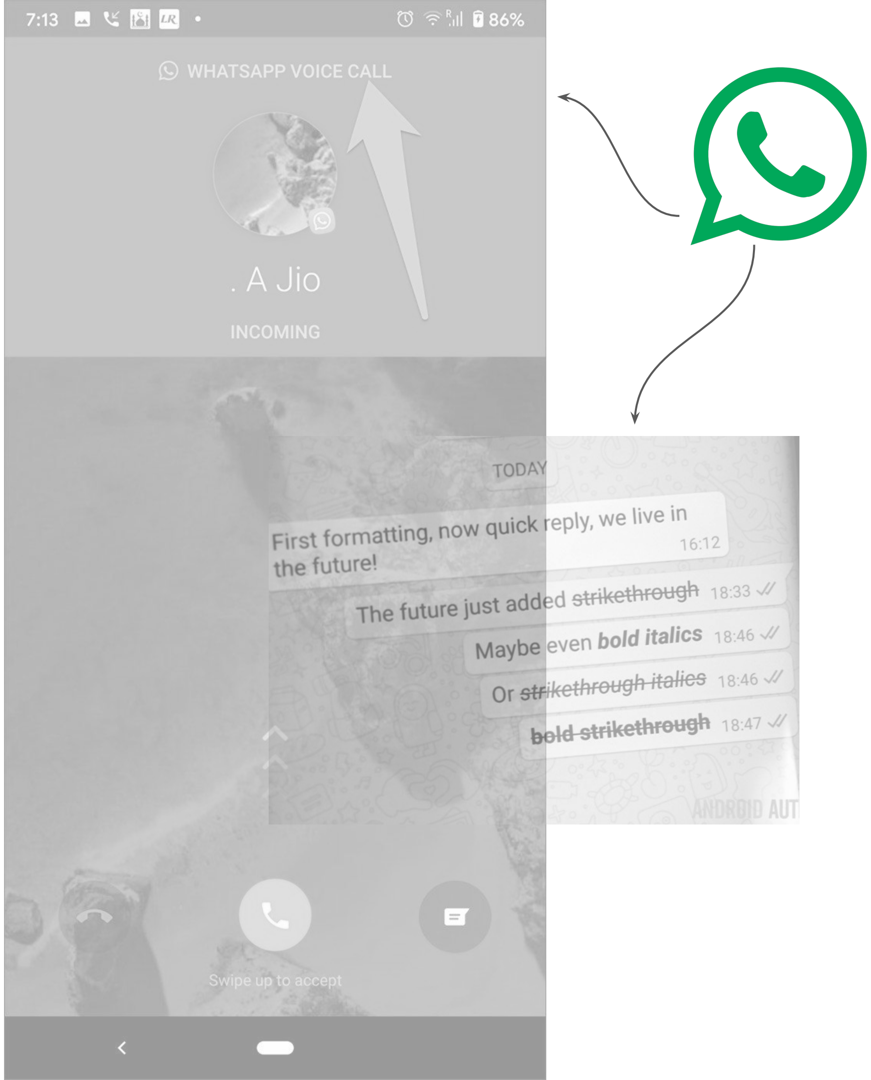
The future just added ~~strikethrough~~ 18:33

Maybe even ***bold italics*** 18:46

Or ~~*strikethrough italics*~~ 18:46

~~**bold strikethrough**~~ 18:47

Just like Whatsapp on Android, there are packages (like dplyr) on R



TODAY

First formatting, now quick reply, we live in the future! 16:12

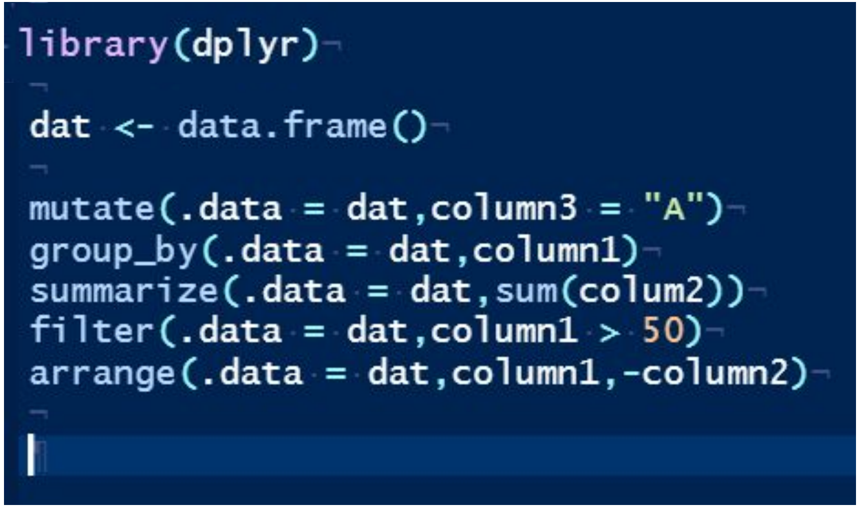
The future just added ~~strikethrough~~ 18:33

Maybe even ***bold italics*** 18:46

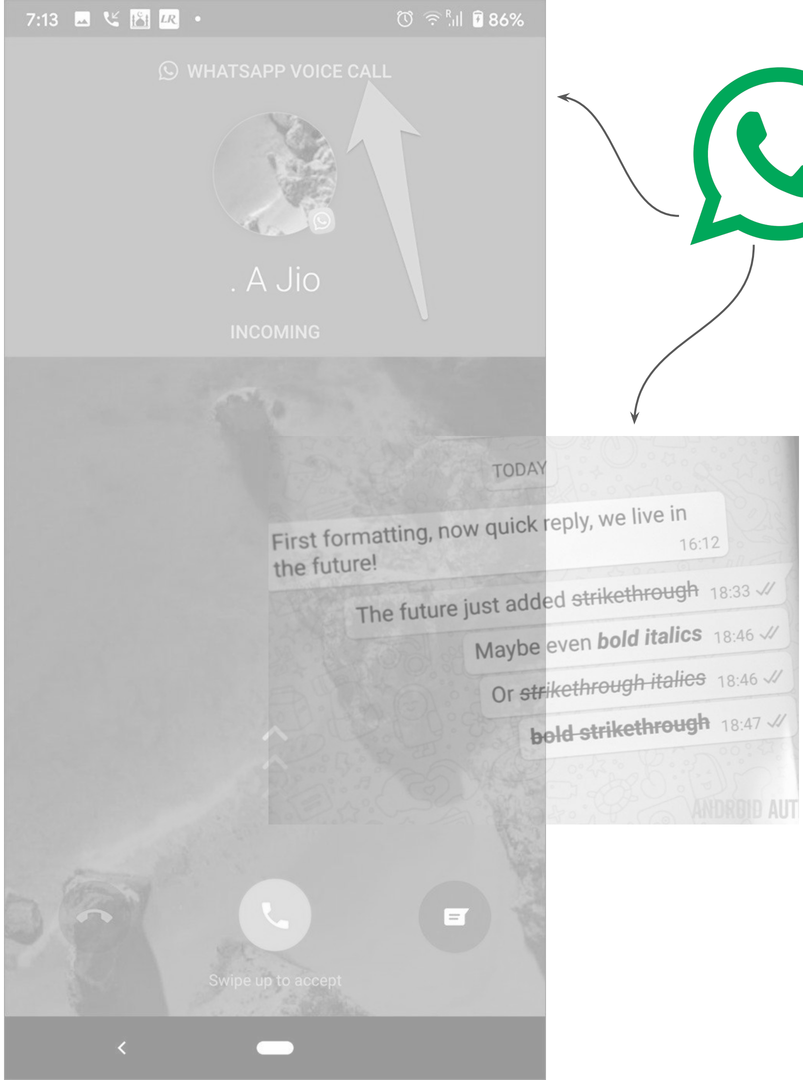
Or ~~*strikethrough italics*~~ 18:46

~~**bold strikethrough**~~ 18:47

Each package can do several tasks, and these are called functions



```
library(dplyr)

dat <- data.frame()

mutate(.data = dat,column3 = "A")
group_by(.data = dat,column1)
summarize(.data = dat,sum(colum2))
filter(.data = dat,column1 > 50)
arrange(.data = dat,column1,-column2)
```



TODAY

First formatting, now quick reply, we live in the future! 16:12

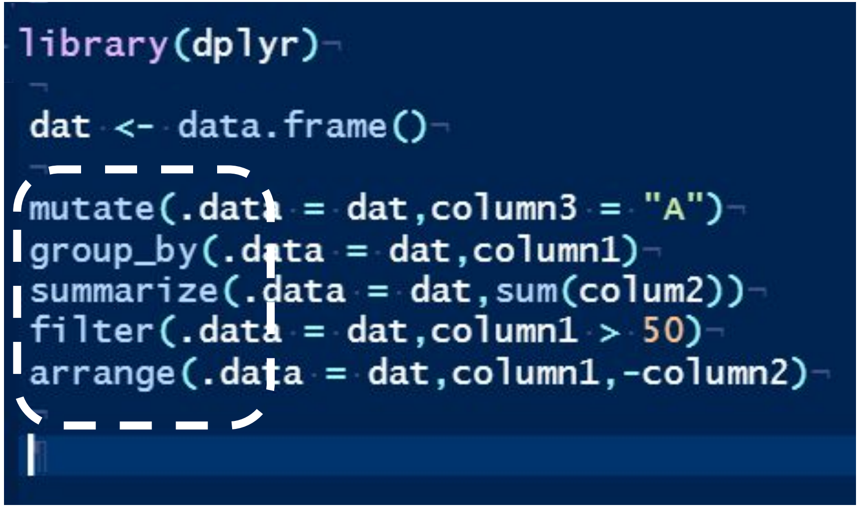
The future just added ~~strikethrough~~ 18:33

Maybe even ***bold italics*** 18:46

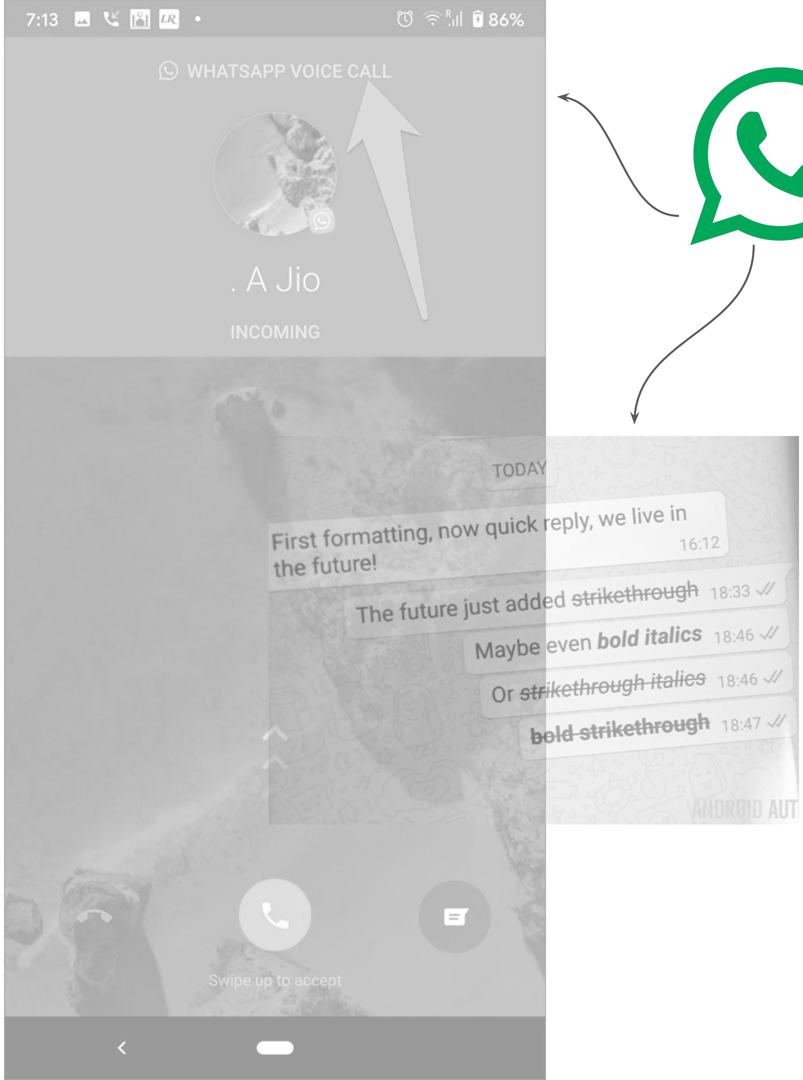
Or ~~*strikethrough italics*~~ 18:46

~~**bold strikethrough**~~ 18:47

Each function is followed by parentheses ()



```
library(dplyr)

dat <- data.frame()

mutate(.data = dat,column3 = "A")
group_by(.data = dat,column1)
summarize(.data = dat,sum(colum2))
filter(.data = dat,column1 > 50)
arrange(.data = dat,column1,-column2)
```



TODAY

First formatting, now quick reply, we live in the future! 16:12

The future just added ~~strikethrough~~ 18:33

Maybe even ***bold italics*** 18:46

Or ~~*strikethrough italics*~~ 18:46

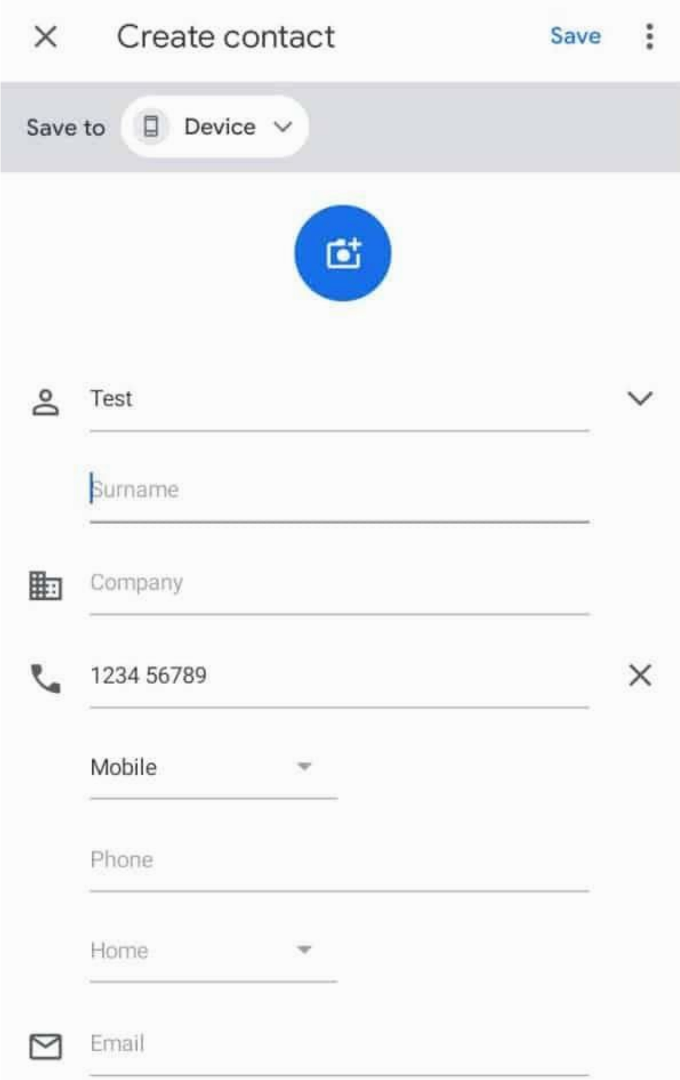
~~**bold strikethrough**~~ 18:47

But what do functions() in an R package work on?

But what do functions() in an R package work on?
Data!

But data needs to be stored somewhere to be able to use it!

In Android, you can store data in different objects



Test

Surname

Company

1234 56789

Mobile

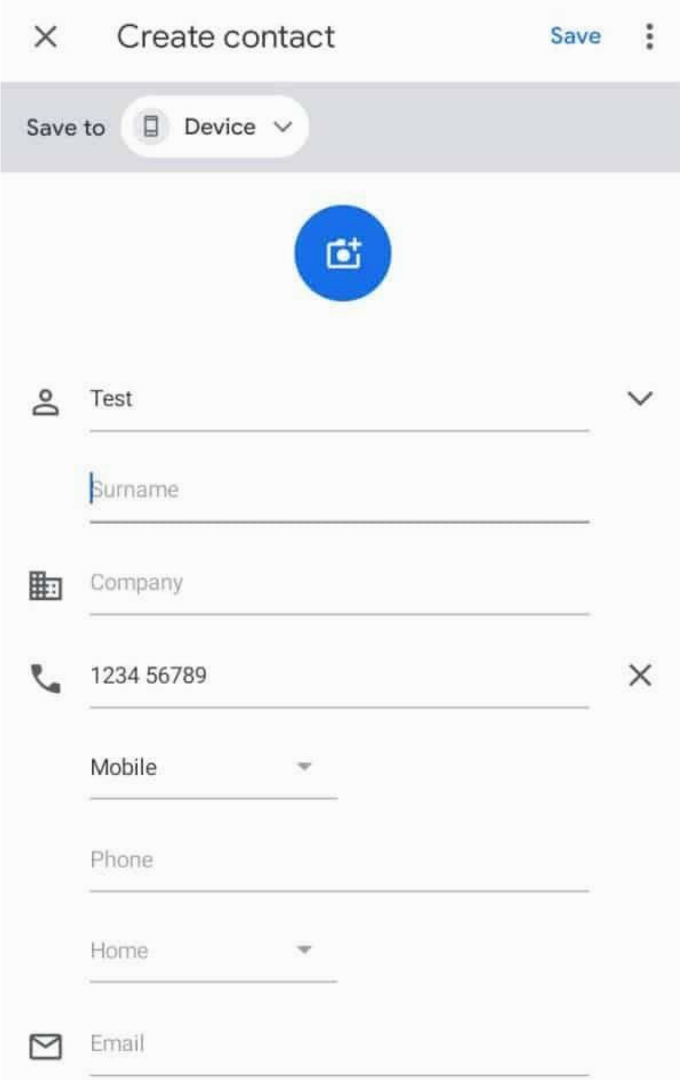
Phone

Home

Email

In Android, you can store data in different objects

In the contacts page, you have an object that stores the name, an object that stores the phone number or another that stores the email ID



Test

Surname

Company

1234 56789

Mobile

Phone

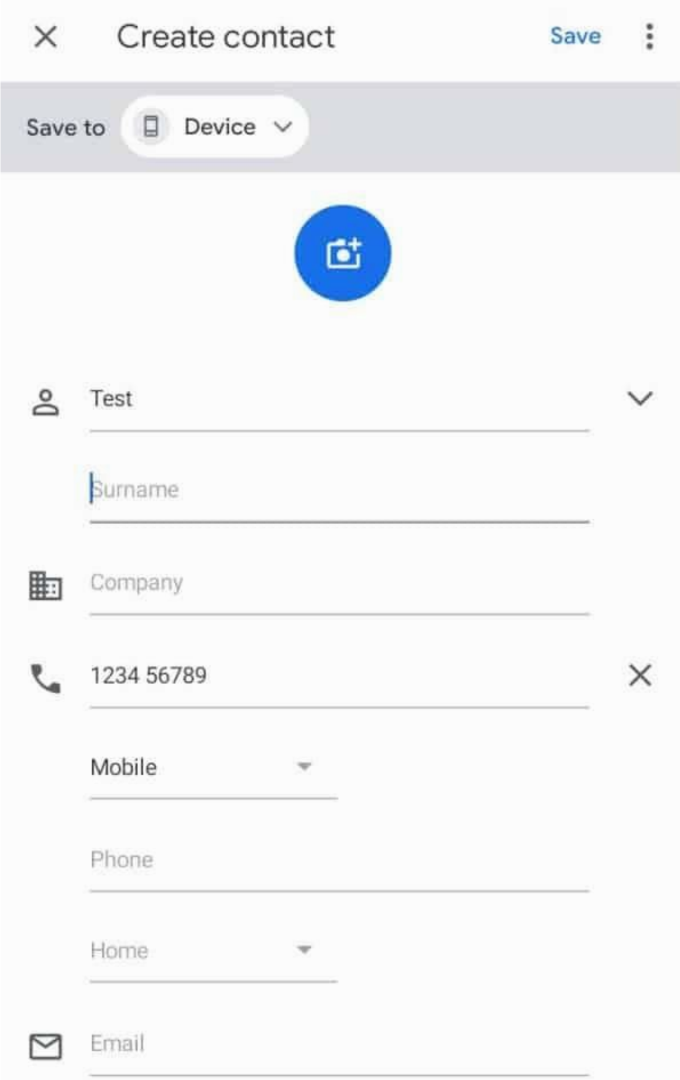
Home

Email

In Android, you can store data in different objects

In the contacts page, you have an object that stores the name, an object that stores the phone number or another that stores the email ID

Objects can either take text or numbers, while some can only store numbers (phone number object)



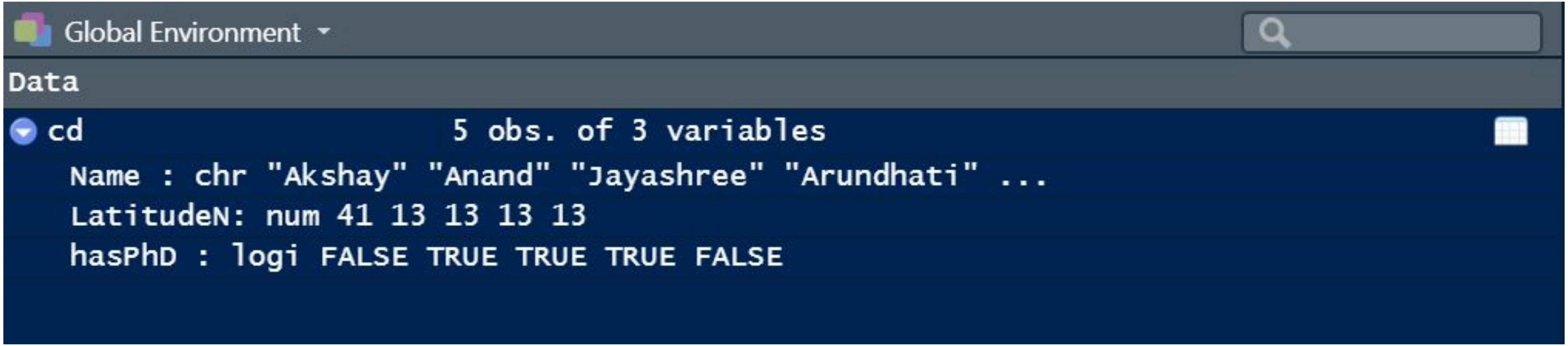
Save to Device

Test

1234 56789

Mobile

Similarly, **R stores data in objects**

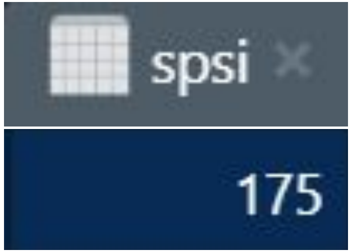


```
cd                      5 obs. of 3 variables
   Name : chr "Akshay" "Anand" "Jayashree" "Arundhati" ...
   LatitudeN: num 41 13 13 13 13
   hasPhD : logi FALSE TRUE TRUE TRUE FALSE
```

There are 4 types of data in R*:

1. character or *char* (text)
2. numeric or *num* (number)
3. logical or *logi* (boolean, TRUE or FALSE)

The simplest object has one element



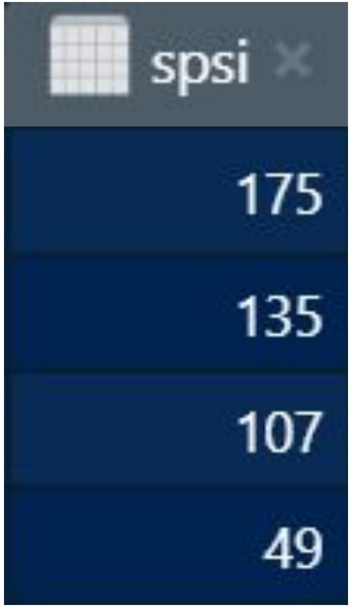
spsi

175

Name of the object

Data stored in the object

Objects with a chain of elements are called **VECTORS**



spsi

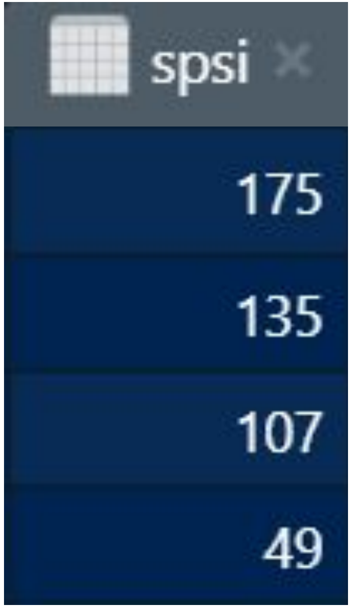
spsi
175
135
107
49

Name of the object

Data stored in the object

Note: all elements of the vector are of the ***same data type*** *(all numbers, all text, etc)*

Objects with a chain of elements are called **VECTORS**



spsi

175

135

107

49

Name of the object

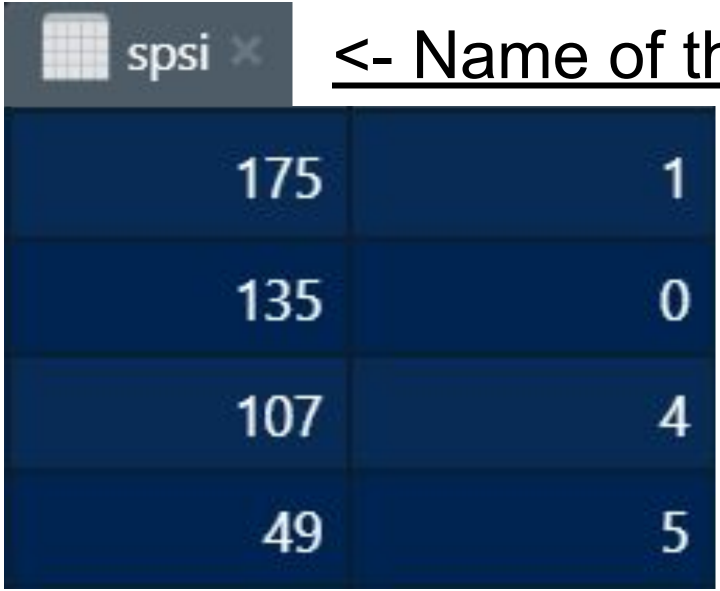
Data stored in the object

Each element of a vector has a unique index or address that refers to an element (using **[index],** pronounced **at**)

Object[3] here refers to _107_

(R: index-1 language)

Objects can also contain a series of vectors, one next to the other - this is called a **MATRIX**



spsi ×

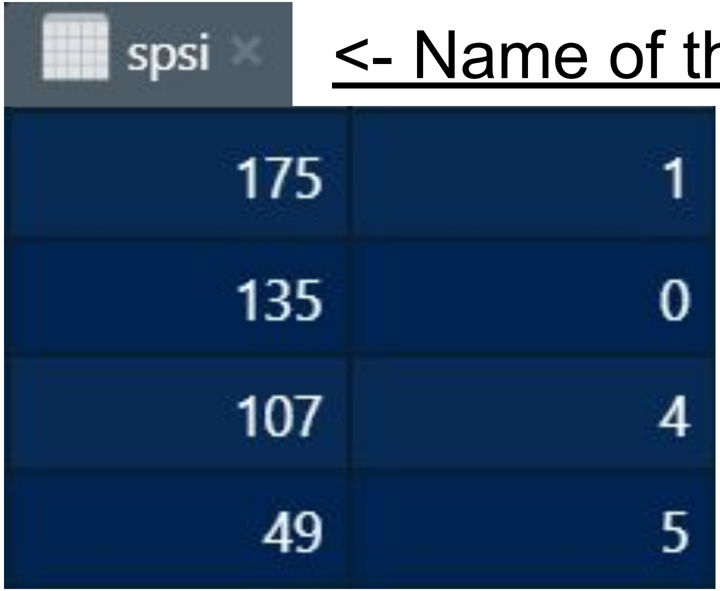
175	1
135	0
107	4
49	5

<- Name of the object

Data stored in the object

Note: all elements of a matrix are of the same data type (all numbers, all text, etc)

Objects can also contain a series of vectors, one next to the other - this is called a **MATRIX**



175	1
135	0
107	4
49	5

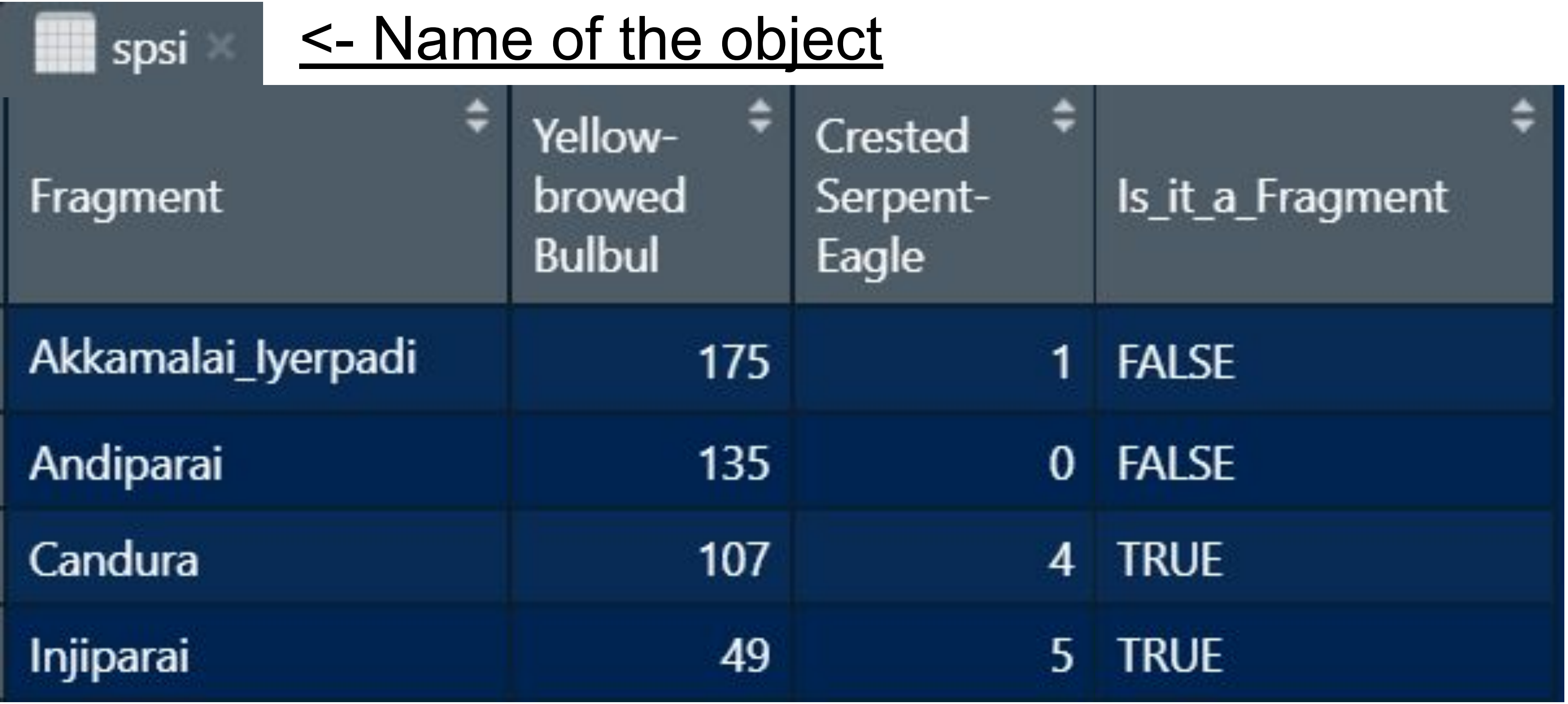
<- Name of the object

Data stored in the object

Each element of a matrix has two indices, first row then column index (**using [row, col]**)

Object[3,2] refers to _4_

A matrix may also contain *different kinds of data types* - these are **DATA FRAMES***



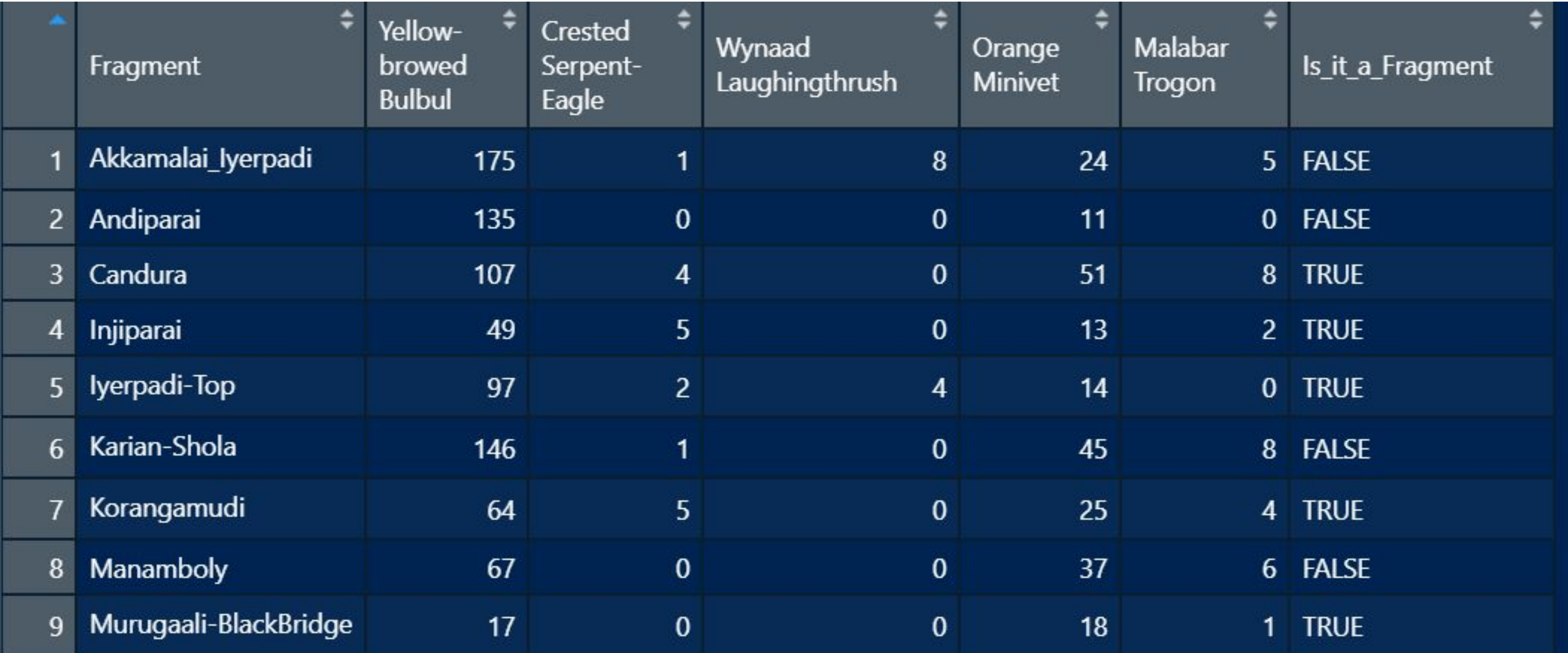
Fragment	Yellow-browed Bulbul	Crested Serpent-Eagle	Is_it_a_Fragment
Akkamalai_Iyerpadi	175	1	FALSE
Andiparai	135	0	FALSE
Candura	107	4	TRUE
Injiparai	49	5	TRUE

<- Name of the object

Column name

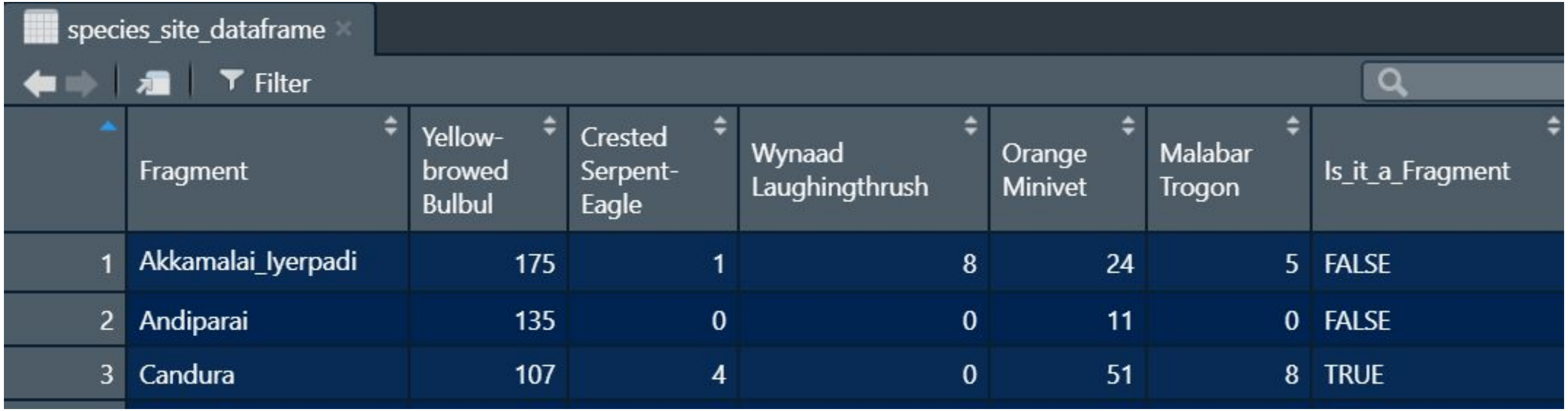
Data

Almost all our objects going forward will be data frames

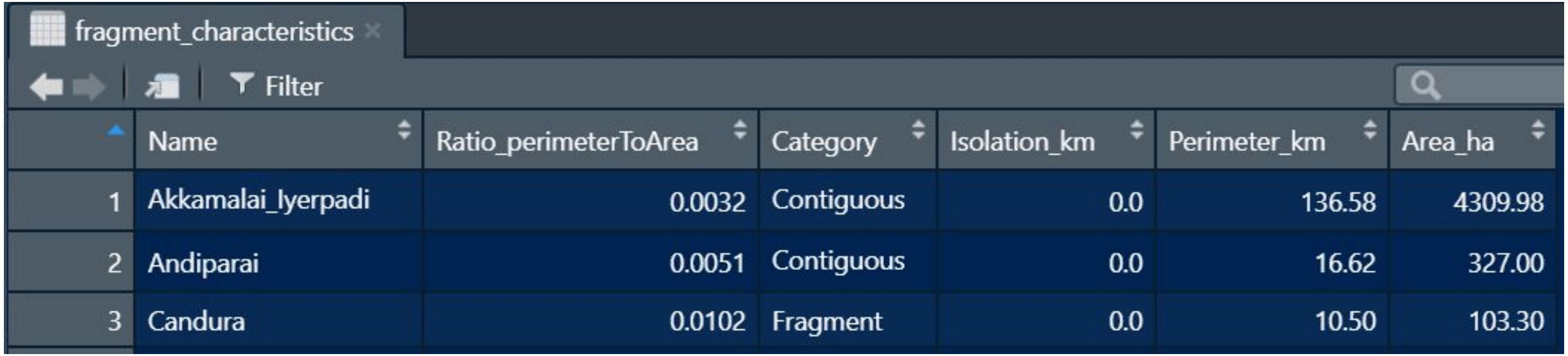


	Fragment	Yellow-browed Bulbul	Crested Serpent-Eagle	Wynaad Laughingthrush	Orange Minivet	Malabar Trogon	Is_it_a_Fragment
1	Akkamalai_Iyerpadi	175	1	8	24	5	FALSE
2	Andiparai	135	0	0	11	0	FALSE
3	Candura	107	4	0	51	8	TRUE
4	Injiparai	49	5	0	13	2	TRUE
5	Iyerpadi-Top	97	2	4	14	0	TRUE
6	Karian-Shola	146	1	0	45	8	FALSE
7	Korangamudi	64	5	0	25	4	TRUE
8	Manamboly	67	0	0	37	6	FALSE
9	Murugaali-BlackBridge	17	0	0	18	1	TRUE

We also have a super-class of objects, a *collection* of data frames, called lists

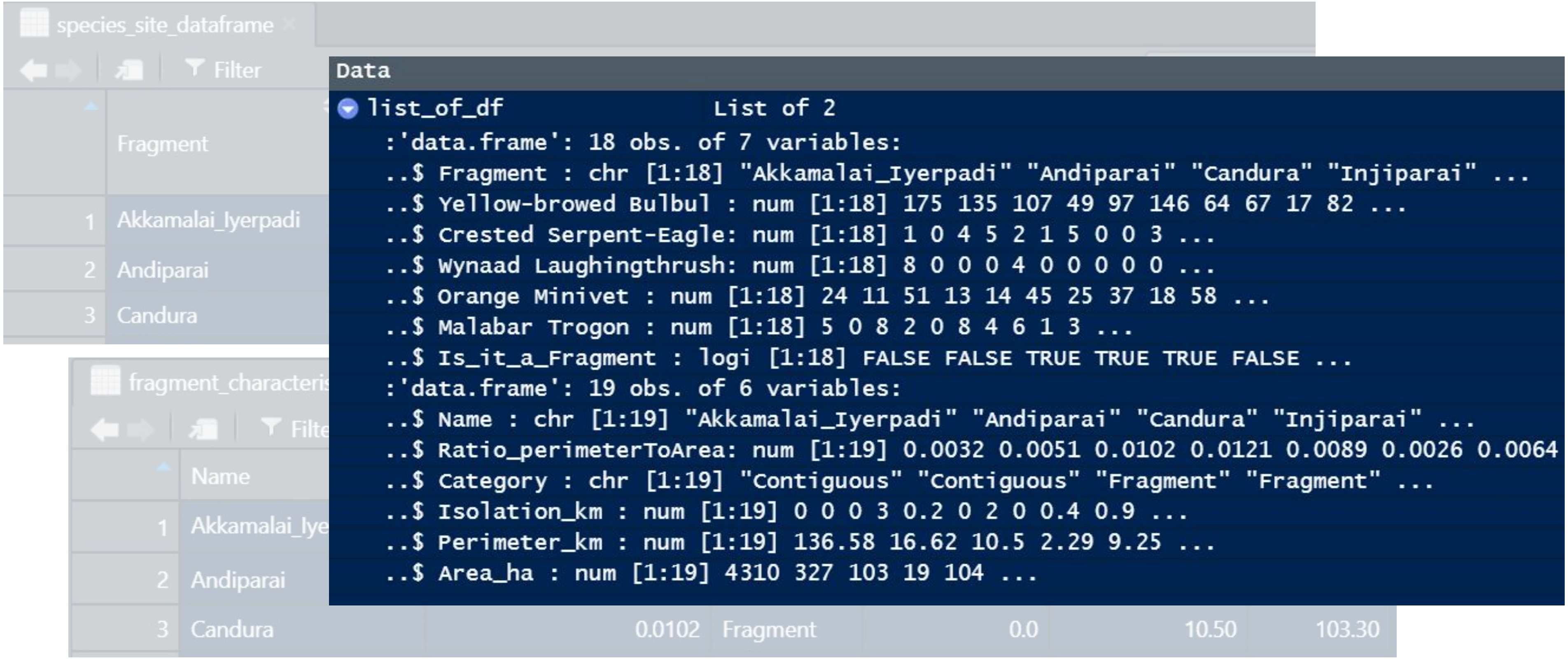


	Fragment	Yellow-browed Bulbul	Crested Serpent-Eagle	Wynaad Laughingthrush	Orange Minivet	Malabar Trogon	Is_it_a_Fragment
1	Akkamalai_Iyerpadi	175	1	8	24	5	FALSE
2	Andiparai	135	0	0	11	0	FALSE
3	Candura	107	4	0	51	8	TRUE



	Name	Ratio_perimeterToArea	Category	Isolation_km	Perimeter_km	Area_ha
1	Akkamalai_Iyerpadi	0.0032	Contiguous	0.0	136.58	4309.98
2	Andiparai	0.0051	Contiguous	0.0	16.62	327.00
3	Candura	0.0102	Fragment	0.0	10.50	103.30

This list has two elements, each element is itself a data frame - don’t worry about lists for now



```
Data
list_of_df                    List of 2
  :'data.frame': 18 obs. of 7 variables:
  ..$ Fragment : chr [1:18] "Akkamalai_Iyerpadi" "Andiparai" "Candura" "Injiparai" ...
  ..$ Yellow-browed Bulbul : num [1:18] 175 135 107 49 97 146 64 67 17 82 ...
  ..$ Crested Serpent-Eagle: num [1:18] 1 0 4 5 2 1 5 0 0 3 ...
  ..$ Wynaad Laughingthrush: num [1:18] 8 0 0 0 4 0 0 0 0 0 ...
  ..$ Orange Minivet : num [1:18] 24 11 51 13 14 45 25 37 18 58 ...
  ..$ Malabar Trogon : num [1:18] 5 0 8 2 0 8 4 6 1 3 ...
  ..$ Is_it_a_Fragment : logi [1:18] FALSE FALSE TRUE TRUE TRUE FALSE ...
  :'data.frame': 19 obs. of 6 variables:
  ..$ Name : chr [1:19] "Akkamalai_Iyerpadi" "Andiparai" "Candura" "Injiparai" ...
  ..$ Ratio_perimeterToArea: num [1:19] 0.0032 0.0051 0.0102 0.0121 0.0089 0.0026 0.0064
  ..$ Category : chr [1:19] "Contiguous" "Contiguous" "Fragment" "Fragment" ...
  ..$ Isolation_km : num [1:19] 0 0 0 3 0.2 0 2 0 0.4 0.9 ...
  ..$ Perimeter_km : num [1:19] 136.58 16.62 10.5 2.29 9.25 ...
  ..$ Area_ha : num [1:19] 4310 327 103 19 104 ...
```

Hierarchy of objects

List

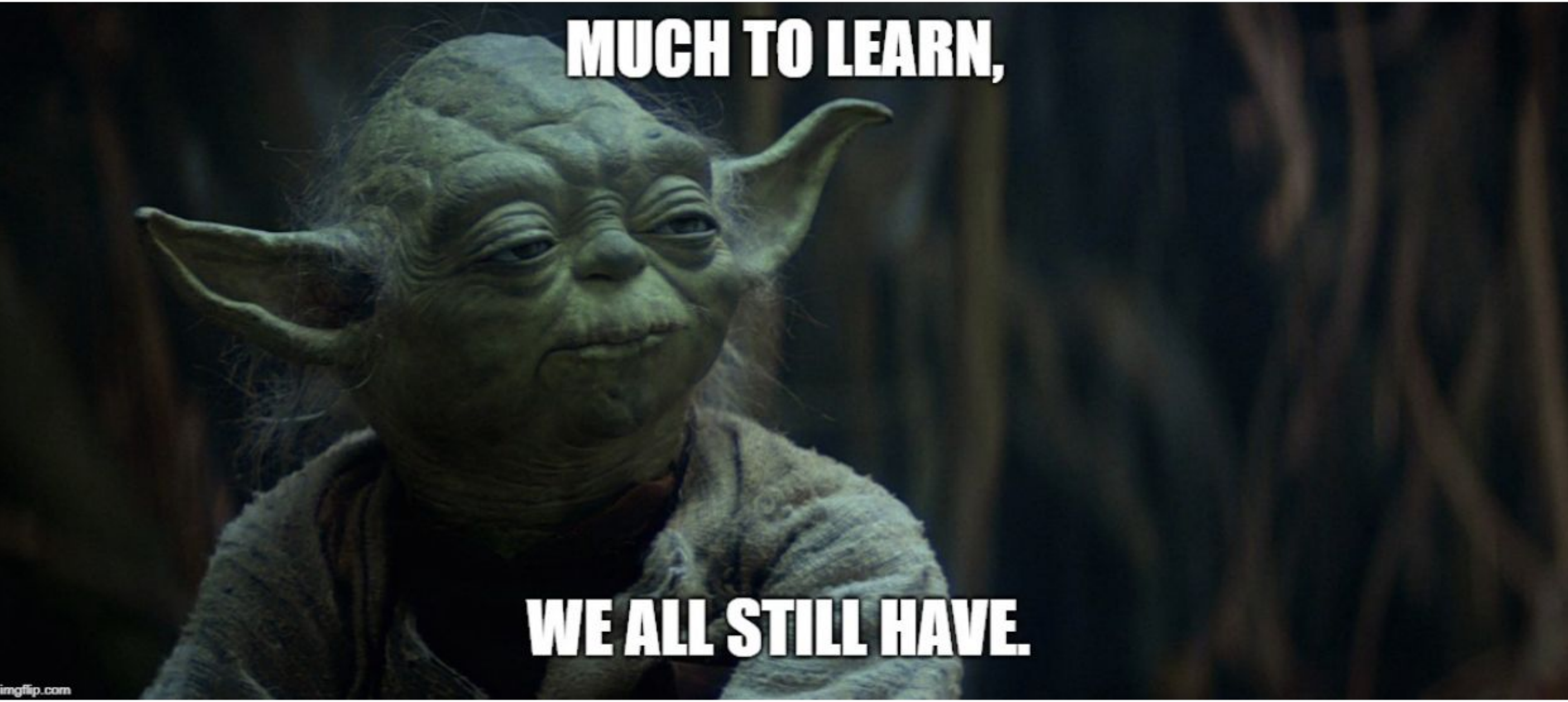
Data frame

Matrix

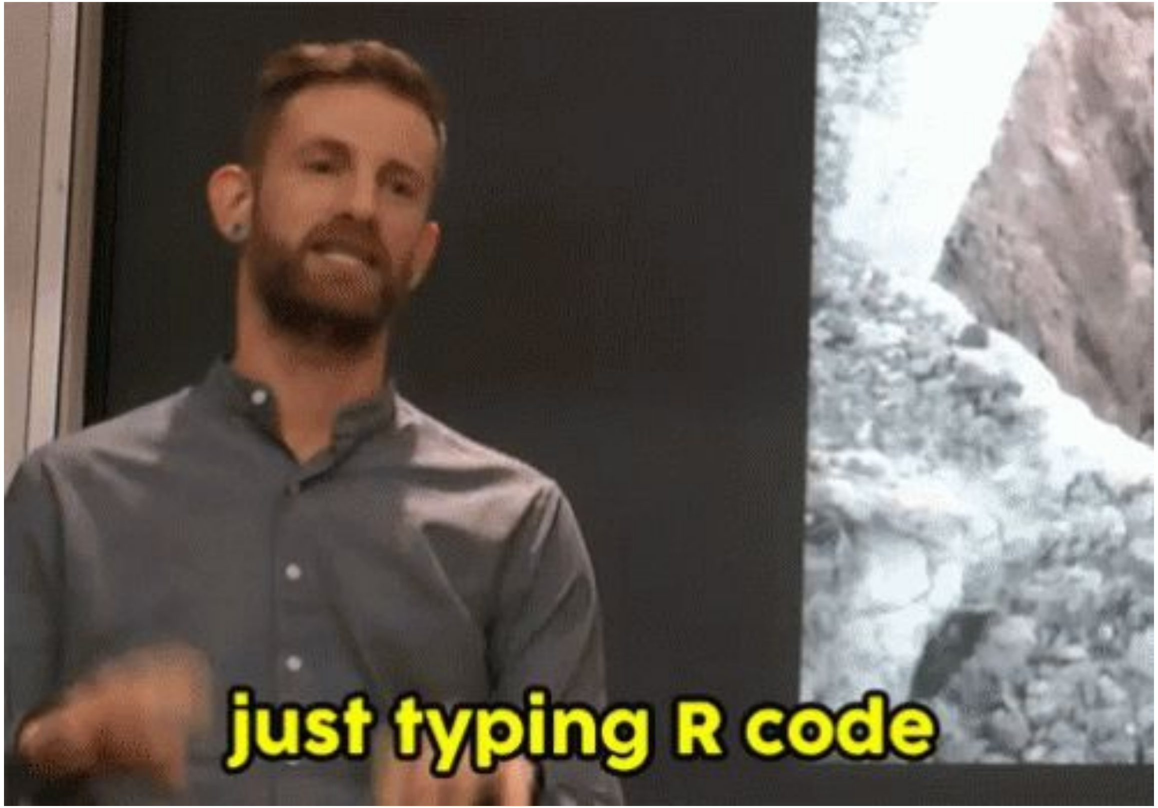
Vector

1-element object
(also a vector)

1. R is (mostly) composed of **objects** that store data and **functions** that perform tasks on **objects**
2. **Functions** can be bundled together as **packages**
3. **Objects** are of various kinds and hierarchical - **vectors, matrices, data frames and lists**


MUCH TO LEARN,
WE ALL STILL HAVE.
imgflip.com

If you've all installed R and RStudio on your computers, let's write our first program!



Let’s install tidyverse!

Contact us for any questions, clarification or feedback!

Anand Osuri - aosuri@ncf-india.org

Akshay Surendra - akshaysurendra1@gmail.com