Bioestatística

Introdução ao R

Caio Graco-Roza and Marcelo Manzi Marinho

Universidade do Estado do Rio de Janeiro

2021-02-23

Teaching staff

Marcelo Manzi Marinho

@Marcelomanzi2

Caio Graco-Roza

@GracoRoza

Introduction to statistics

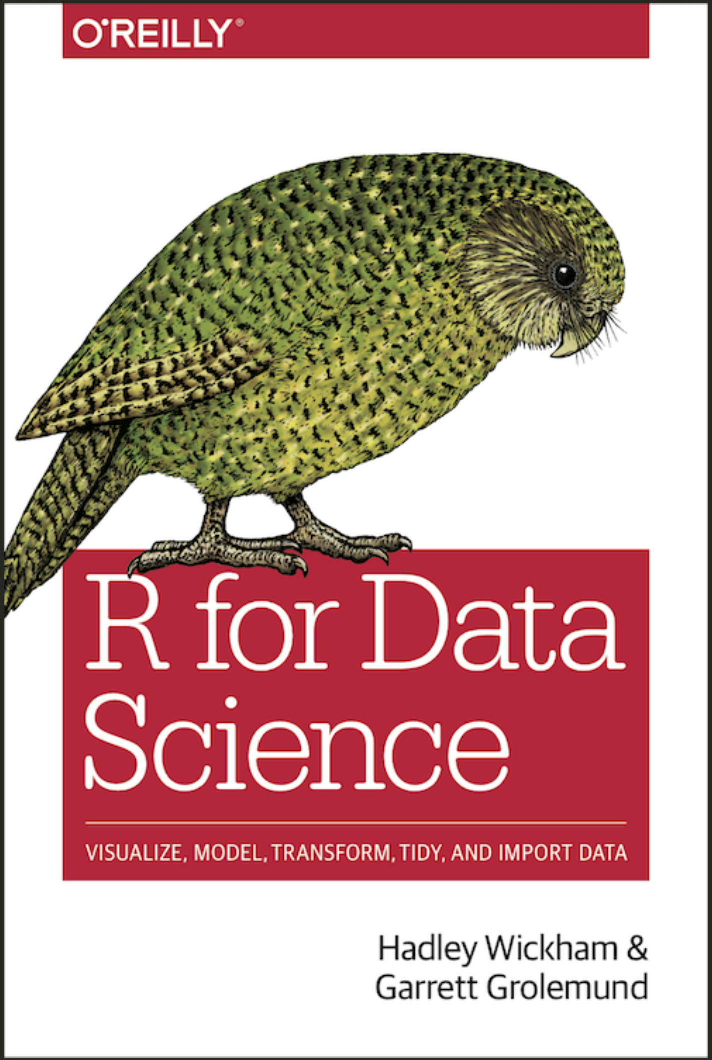
Using the R environment.

Ready?

These classes are based on

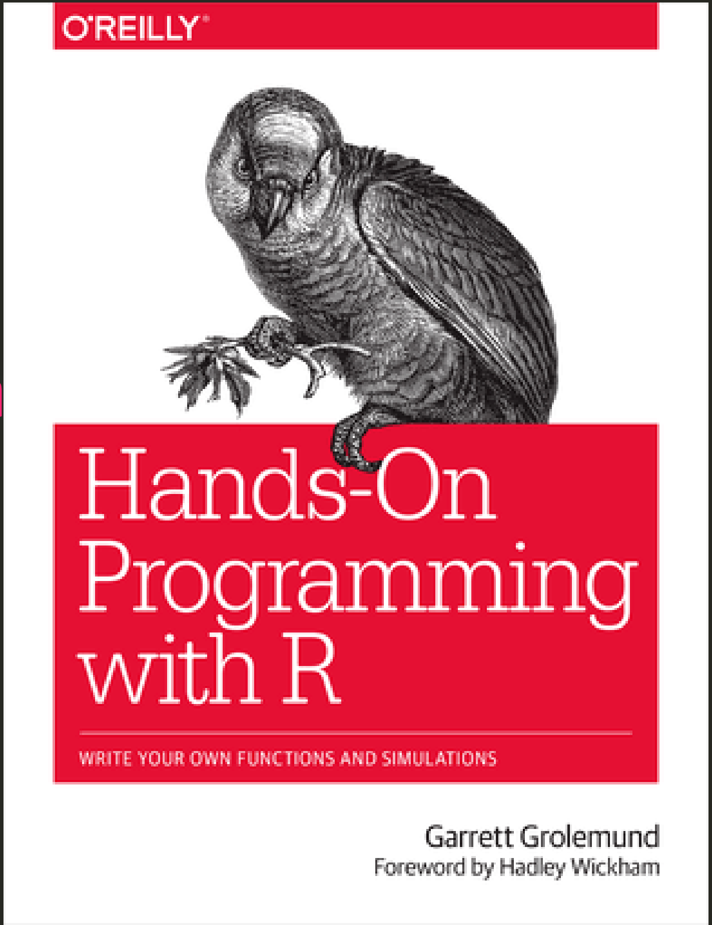
R for data Science

https://r4ds.had.co.nz/



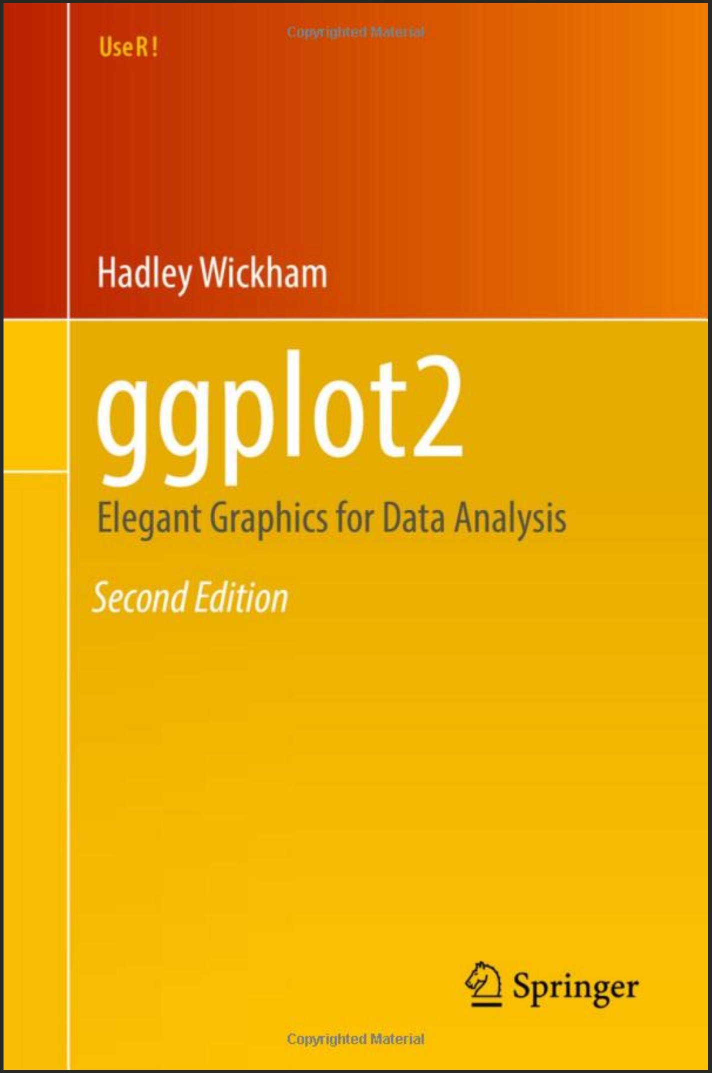
Hands on Programming with R

https://rstudio-education.github.io/hopr/basics.htm


O'REILLY
Hands-On Programming with R
WRITE YOUR OWN FUNCTIONS AND SIMULATIONS
Garrett Grolemund
Foreword by Hadley Wickham

ggplot2: Elegant Graphics for Data Analysis.

https://ggplot2-book.org/



UseR!

Copyrighted Material

Hadley Wickham

ggplot2

Elegant Graphics for Data Analysis

Second Edition

Springer

Copyrighted Material

Why are we all here?

"Vocês devem estar se perguntando por que eu reuni todos vocês aqui hoje".

We are here to study **Statistics** and practice some **Data science**.

Statistics

The practice or science of collecting and analysing numerical data in large quantities, especially for the purpose of inferring **proportions** in a whole from those in a **representative** sample.

Data science

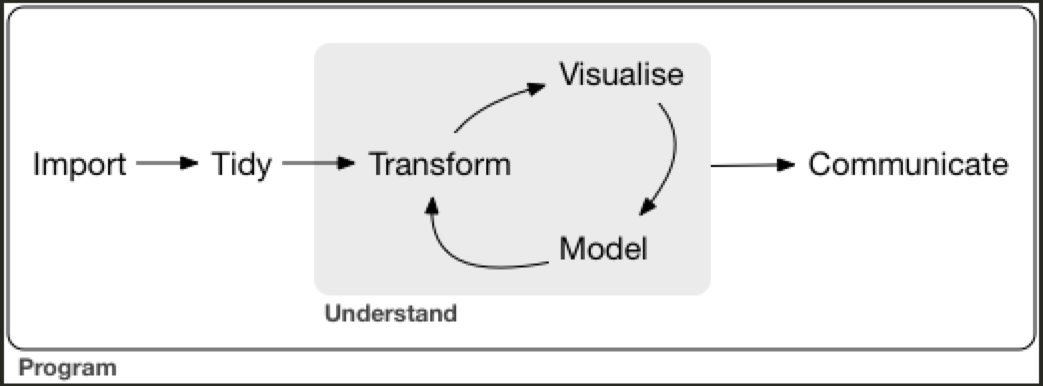
Aims to turn raw data into understanding, insight, and knowledge.

Data science combines multi-disciplinary fields and computing to interpret data for decision making whereas statistics refers to mathematical analysis which use quantified models to represent a given set of data.

Statistics + Data Science

How do we work?

Here's a usual framework of a data science project.



In biological sciences we often:

- Substitute the `Import` by `Collect` as we generate our own data.
- Do the whole `Tidying` using excel or even additional notebooks.
- Perform the `Transform`, `Visualise` and `Model` using multiple software that our labs may or may not have money to buy a license.
- `Communicate` our findings as reports or papers published in indexed journals.

Drawbacks of the actual approach

- Several tutorials with different guidelines leaving no time for social media.
- Several premium licenses leaving no extra money for traveling (Ok, for buying reagents).
- You really have no idea of what's going on behind the curtains (meaning closed-source code).

How can we solve such huge problems?

R R stackoverflow

What is R?

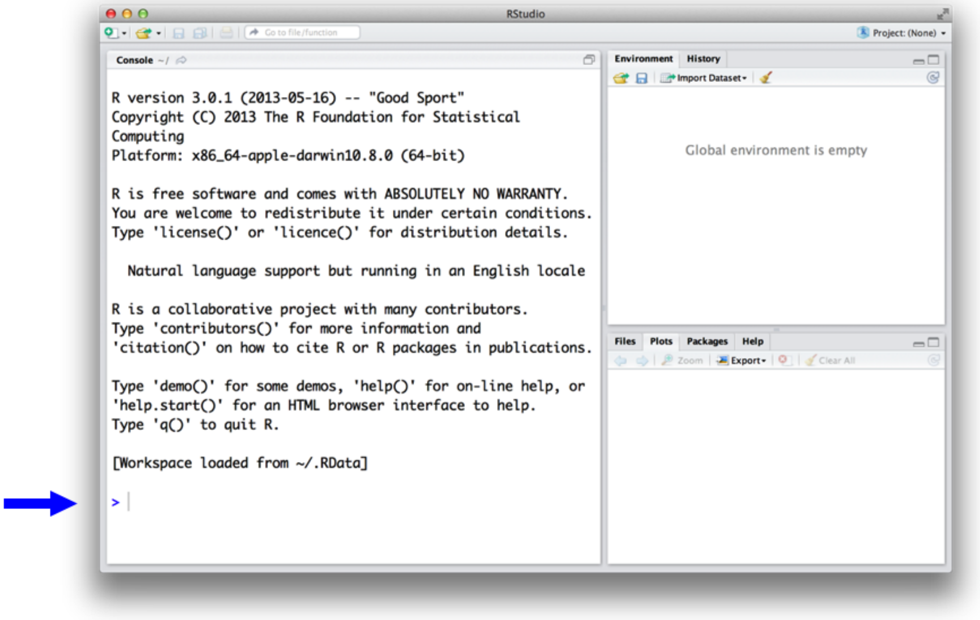
- R is both a programming language and software environment for statistical computing, which is *free* and *open-source*.
- R is an implementation of the S programming language, which was created by Ross Ihahka and Robert Gentlemen at the University of Auckland”

Why R Studio?

RStudio gives you a way to talk to your computer. R gives you a language to speak in.

R User Interface

The `RStudio` interface is simple. You type R code into the bottom line of the **RStudio console** pane and then click Enter to run it. The code you type is called a command, because it will command your computer to do something for you. The line you type it into is called the command line.



```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical
Computing
Platform: x86_64-apple-darwin10.8.0 (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.

[Workspace loaded from ~/.RData]

> |
```

The interface is divided also into a **Workspace Browser** and **Plots**. These can also be adjusted under the `Tools (or Preferences) > Global options > Pane layout` menu. Let's now inspect each pane more closely.

R User Interface

Console

This is where you can type code that executes immediately. This is also known as the command line.

Entering code in the command line is intuitive and easy. For example, we can use R as a calculator by typing into the Console (and pressing Enter after each line):

```r
3+2
```

```
## [1] 5
```

You’ll notice that a [1] appears next to your result. R is just letting you know that this line begins with the first value in your result.

R User Interface

Console

Sometimes R outputs may encompass several lines. In that case, each line will begin with the brackets [] indicating the position of the first object in the output. e.g.:

```
100:150
```

```
##  [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 1
## [20] 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 1
## [39] 138 139 140 141 142 143 144 145 146 147 148 149 150
```

> The colon operator (:) returns every integer between two integers*.
> It is an easy way to create a sequence of numbers.

[*] Whole-valued positive or negative number or 0.

R User Interface

Console

Nothing is perfect, so there are also some situations where you tap Enter before finishing the code:

> 5+
>
> +

R shows you then the + in the next line of the console indicating that you may finnish the command.

```
5+
+
1
```

```
## [1] 6
```

R User Interface

Console

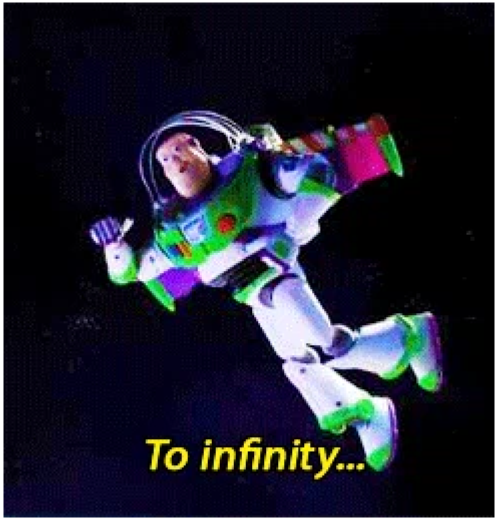
In other situations you may find real Errors:

```
> 3 % 5
Error: unexpected input in "3 % 5"
>
```

R User Interface

Console

The sky is the limit!!



With all the things we learned until now, let's practice:

- Choose any number and add 2 to it.
- Multiply the result by 3.
- Subtract 6 from the answer.
- Divide what you get by 3.

> 10 minutes exercise.

Break! 15 minutes 🕐 ☕

Solution

If you did all correctly you may have in the final step the value you chose to start with.

```
5+2
```

```
## [1] 7
```

```
7*3
```

```
## [1] 21
```

```
21-6
```

```
## [1] 15
```

```
15/3
```

```
## [1] 5
```

R User Interface

Console

In addition, the **Console** title bar has a few useful features:

- It displays the current R working directory (more on this later)
- It provides the ability to interrupt R during a long computation (a stop sign will appear whilst code is running)
- It allows you to minimise and maximise the Console in relation to the Source pane using the buttons at the top-right or by double-clicking the title bar)

R User Interface

Environment and history panes

The Environment pane is very useful as it shows you what objects **(i.e., dataframes, arrays, values and functions)*** you have in your environment (workspace). You can see the values for objects with a single value and for those that are longer R will tell you their class.

Typing the following into the Console will list everything you’ve loaded into the Environment:

```
ls()
```

```
## character(0)
```

Conclusion: There are no objects in the Environment.

[*] We will get to this later, don't worry.

R User Interface

Files, Plots, Packages, Help, and Viewer panes

- *Files*: Allows you to navigate in the folders just like using the operational system.

- *Plots*: This is where the graphics will appear.

- *Packages*: hows you the packages that are installed and those that can be installed.

- *Help*: Allows you to search the R documentation for help and is where the help appears when you ask for it from the *Console*.

  > Try typing `?mean` or `help(mean)` in the console.

R User Interface

Source Editor

Generally we will want to write codes that are longer than a couple of lines. We want this (yes, ***you want it too!***) because we believe we will repeat the same steps many many times (**rountinely**) that would be boring to write it all over again always when we need it .

These files are commonly called scripts and they work like a recipe of a cake. We need to include all the ingredients in the right order in order to get a tastemade-like cake.

You may create a new *R script* using `File > New File > R script`.

Note scripts files are simply standard text files and can be created in any text editor and saved with a .R (or .r) extension, but the Source editor in RStudio has the advantage of providing syntax highlighting, code completion, and smart indentation.

R scripts

By using R scripts you may be interested in storing the results of e.g., equations, transformations or functions to be used later on. This can be easily done by assigning values to *objects*

```
result <- 5+6
```

By running the line above you will not get any results in the console. However, there is an simple way to show it. Simply type the name of the object you assigned and click CTRL + Enter.

```
result
```

```
## [1] 11
```

> The symbol <-shows the direction to which the objects are assigned. It can be substituted by =but we avoid it for good coding* practices.

[*] See Hadley Wickham Style Guide for more details on writing practices.

Objects

R lets you save data by storing it inside an R object. What is an object? Just a name that you can use to call up stored data. For example, you can save data into an object like a or b. Wherever R encounters the object, it will replace it with the data saved inside.

```r
result + 2
```

```
## [1] 13
```

You can name an object in R almost anything you want, but there are a few rules. First, a name cannot start with a number. Second, a name cannot use some special symbols, like `^`, `!`, `$`, `@`, `+`, `-`, `/`, or `*`.

> Note that R is case sensitive so writing names e.g., `Results` and `results` will create two different objects.

Functions

R comes with many functions that you can use to do sophisticated tasks like taking a log or the square root of a number. Some functions involve more than one commands, for example, you can round a number with the `round` function, or calculate its factorial with the `factorial` function.

```r
round(2.6)
```

```
## [1] 3
```

```r
factorial(3) # 3*2*1 (3!)
```

```
## [1] 6
```

> R treats the hashtag character, #, in a special way; R will not run anything that follows a hashtag on a line

R Packages

R comes with a number of built-in functions and datasets, but one of the main strengths of R as an open-source project is its package system. That allows many professors, programmers, and statisticians to design and release tools using R to help people analyze data.

To install a package, use the install.packages() function. Think of this as buying a recipe book from the store, bringing it home, and putting it on your shelf.

```
install.packages("tidyverse") #To install

library("tidyverse") # To load
```

Packages must be installed only once per computer prior using, while loading a package should be done once per R session.

Practice

What about exploring the R universe a little bit before we go deeper into data analysis?

Go to the **Environment ans History pane** click on the `Tutorial` tab and go through the Tutorials (Except for "Tutorial Quiz" and "Slidly demo").

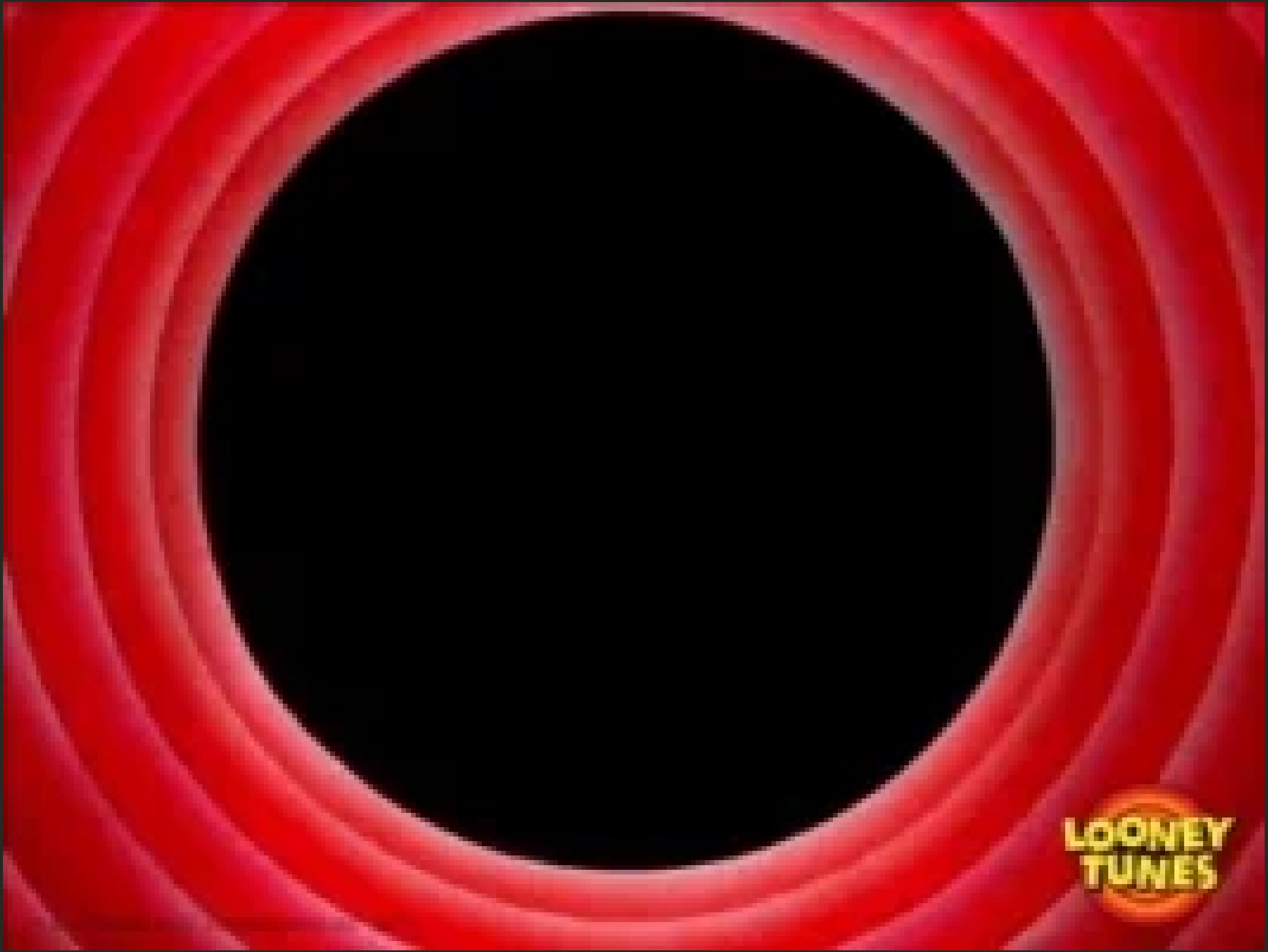
Make notes on your doubts and let's check together tomorrow.

Resources in portuguese

Estatística aplicada ao R

Introdução ao R

Principios de Estatistica em Ecologia



LOONEY TUNES