



# **Intro to This Course; Getting Started with R**

Enrico Toffalini

# Why becoming an R user?

- **Mainstream in academia** for data science, increasingly used in business. *Job market advantage!*
- **Free & open-source:** wherever you go, R will be with you at no costs (unlike *MPLUS, MATLAB, SPSS*, etc.)
- **Real programming language:** difficult at the beginning, but: 1) gives you lots of flexibility; 2) has transfer on other programming languages (e.g., *Python*).
- **Vast community support** thanks to a large and active community (also, *chatGPT, Lucrez-IA*, etc., know it pretty well!).
- **Huge ecosystem**, >20,000 packages on CRAN, more from other sources (e.g., GitHub), to do amazing stuff with statistical data analysis, machine learning, data visualization, developing webapps [*shiny*], writing reports and even entire books [*bookdown, rmarkdown*]), can integrate with *quarto, github*.
- Facilitates **reproducible research** by sharing code and workflows.

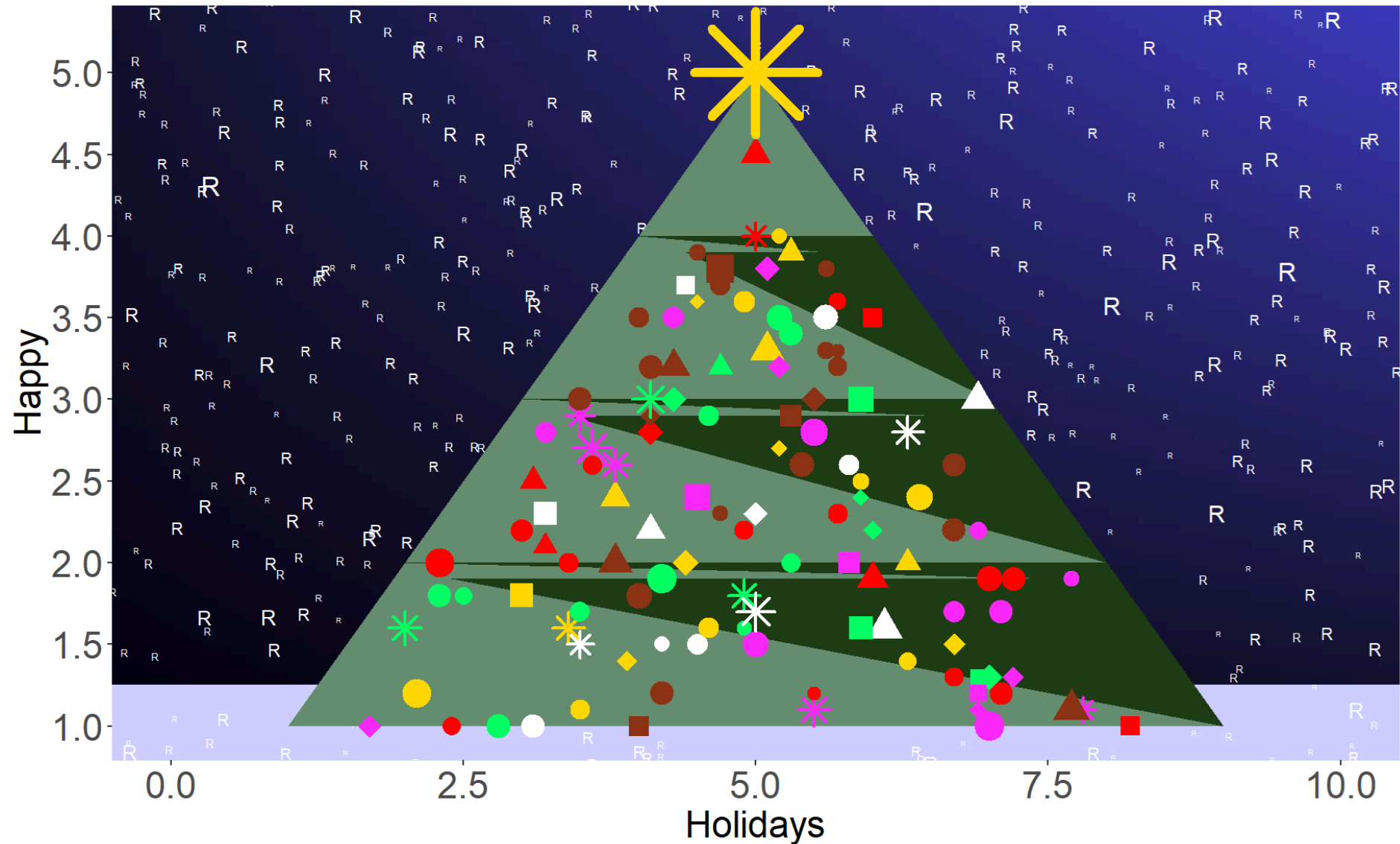
# What you may expect to learn in this course:

- Executing fundamental operations and using basic functions;
- Working with essential data types and structures;
- Gaining some proficiency in managing and manipulating data with vectors and dataframes;
- Understanding some fundamental concepts of programming.

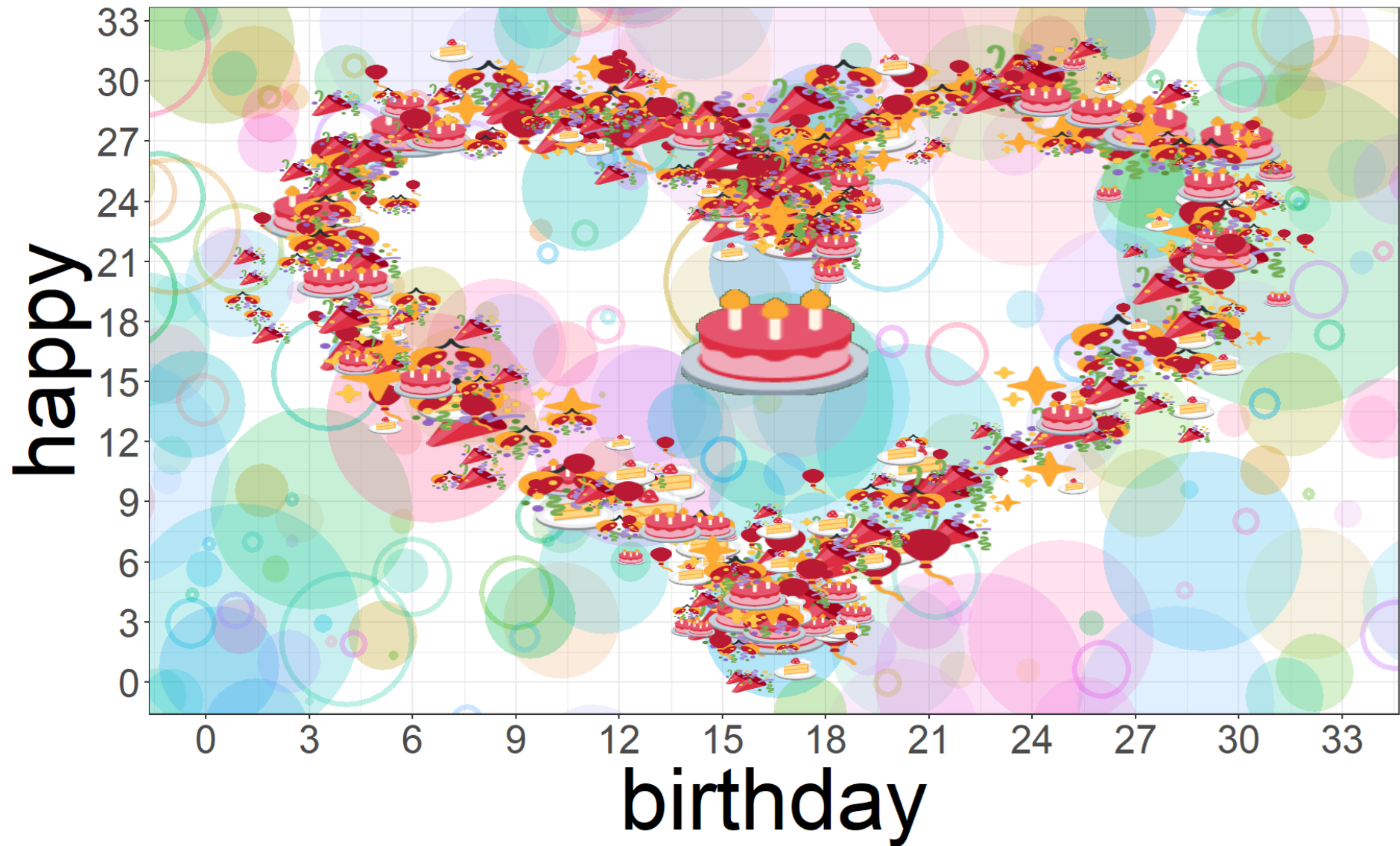
**Over the next *couple of years*, following this PhD program, you may have the opportunity to learn to use R to perform at least some fundamentals about:**

- Core statistical inference methods;
- (Generalized) linear (mixed-effects) modeling;
- Data visualization using *ggplot2*;
- Power analysis via data simulation ;
- Structural Equation Modeling (SEM);
- Conducting Meta-Analysis.

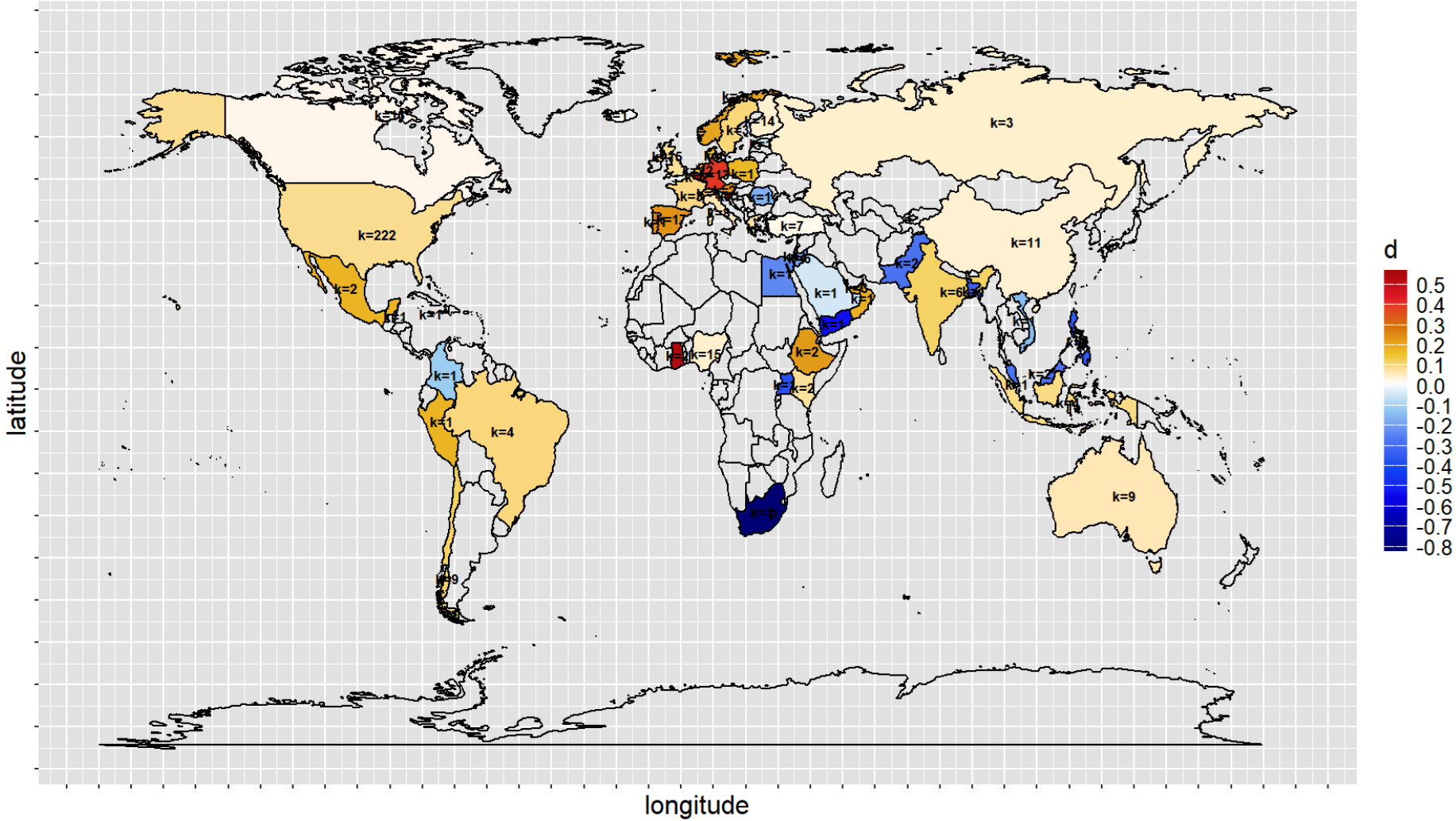
# you may even create greeting cards



you may even create greeting cards

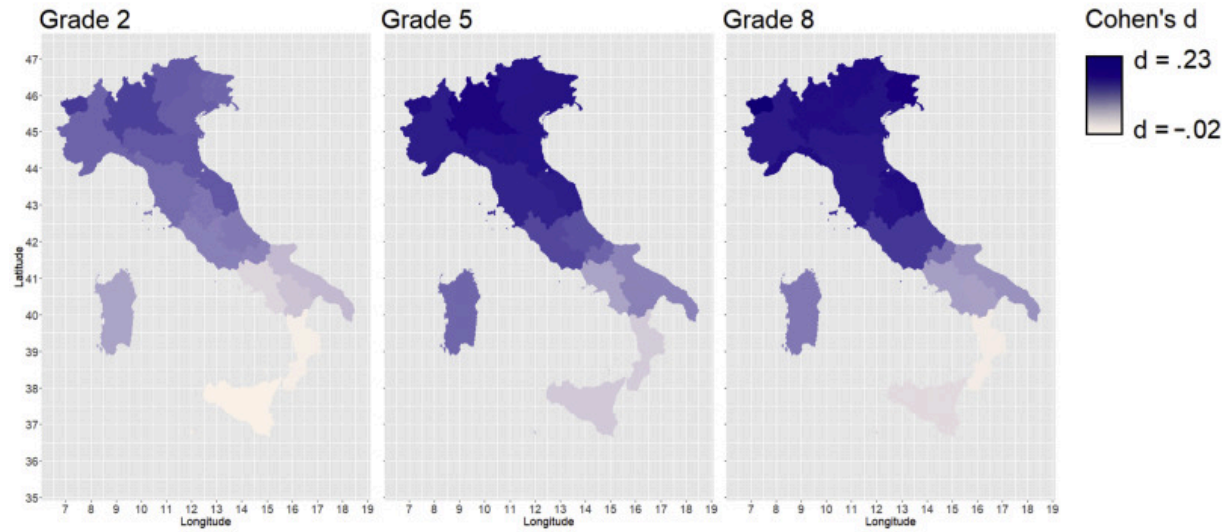


or like fancy infographics

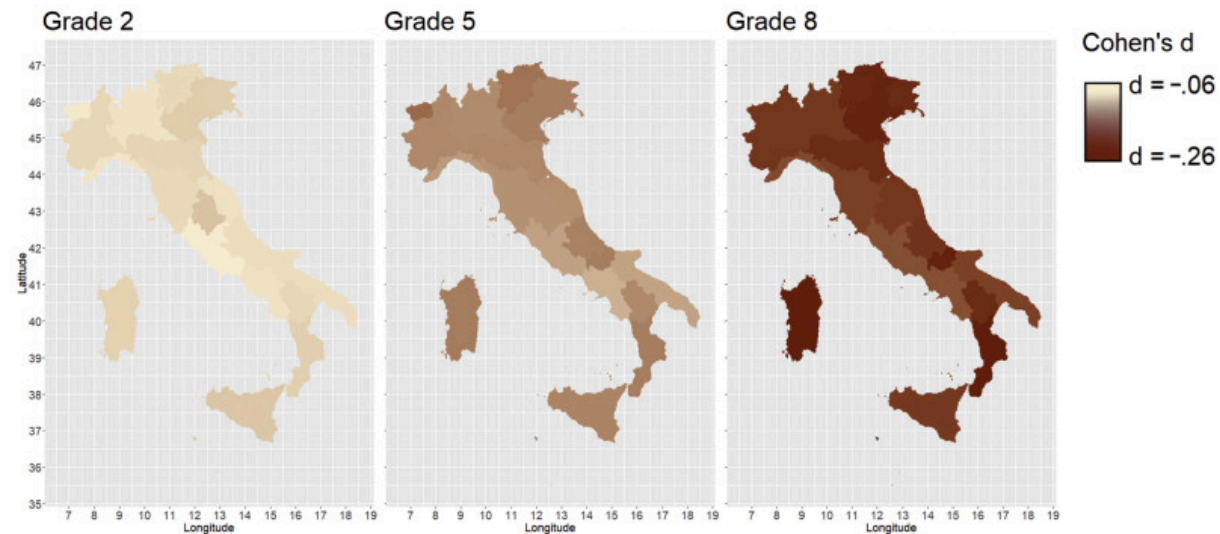


# or like fancy infographics

**A.**

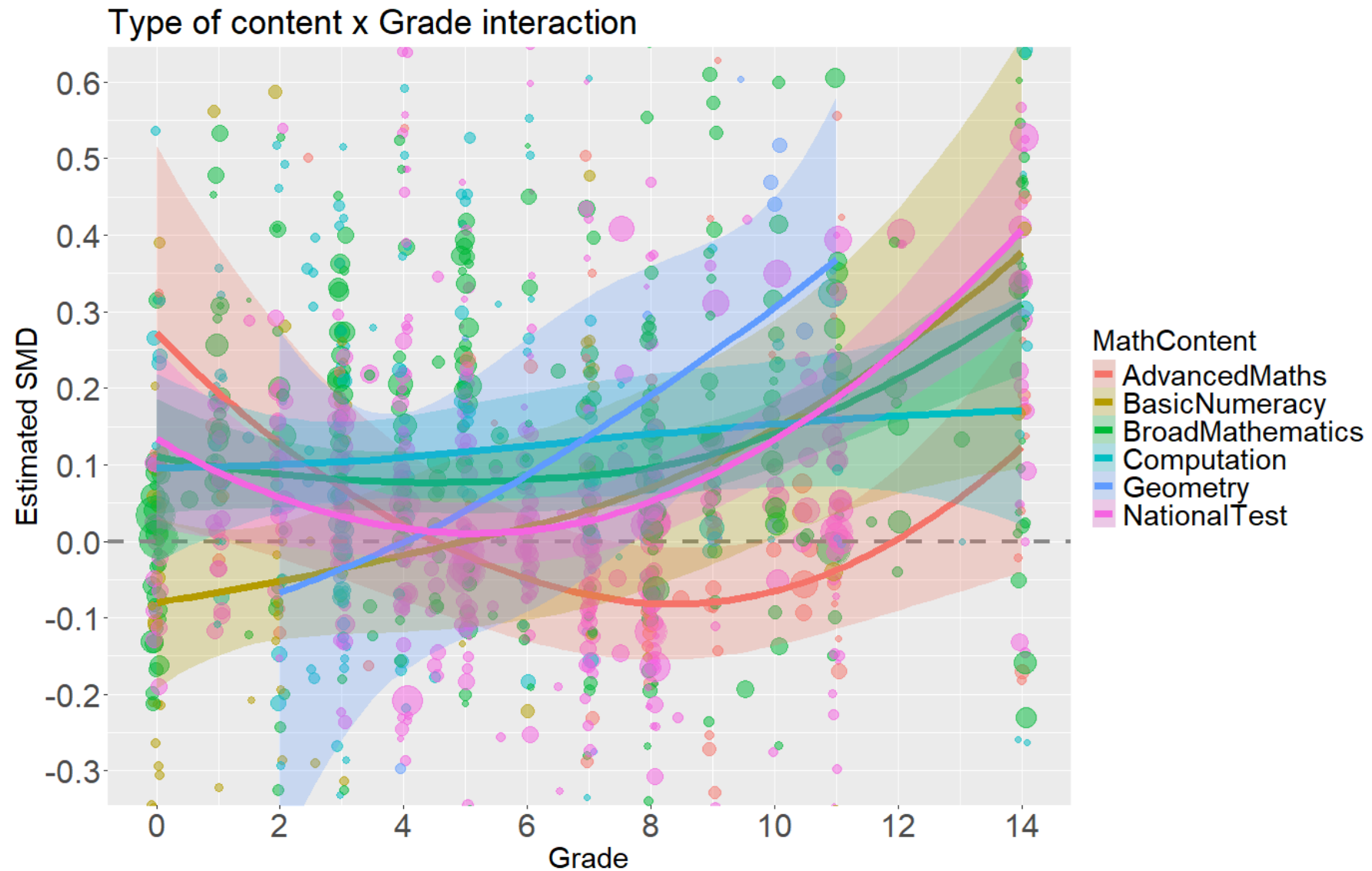


**B.**





# perform classical data analysis



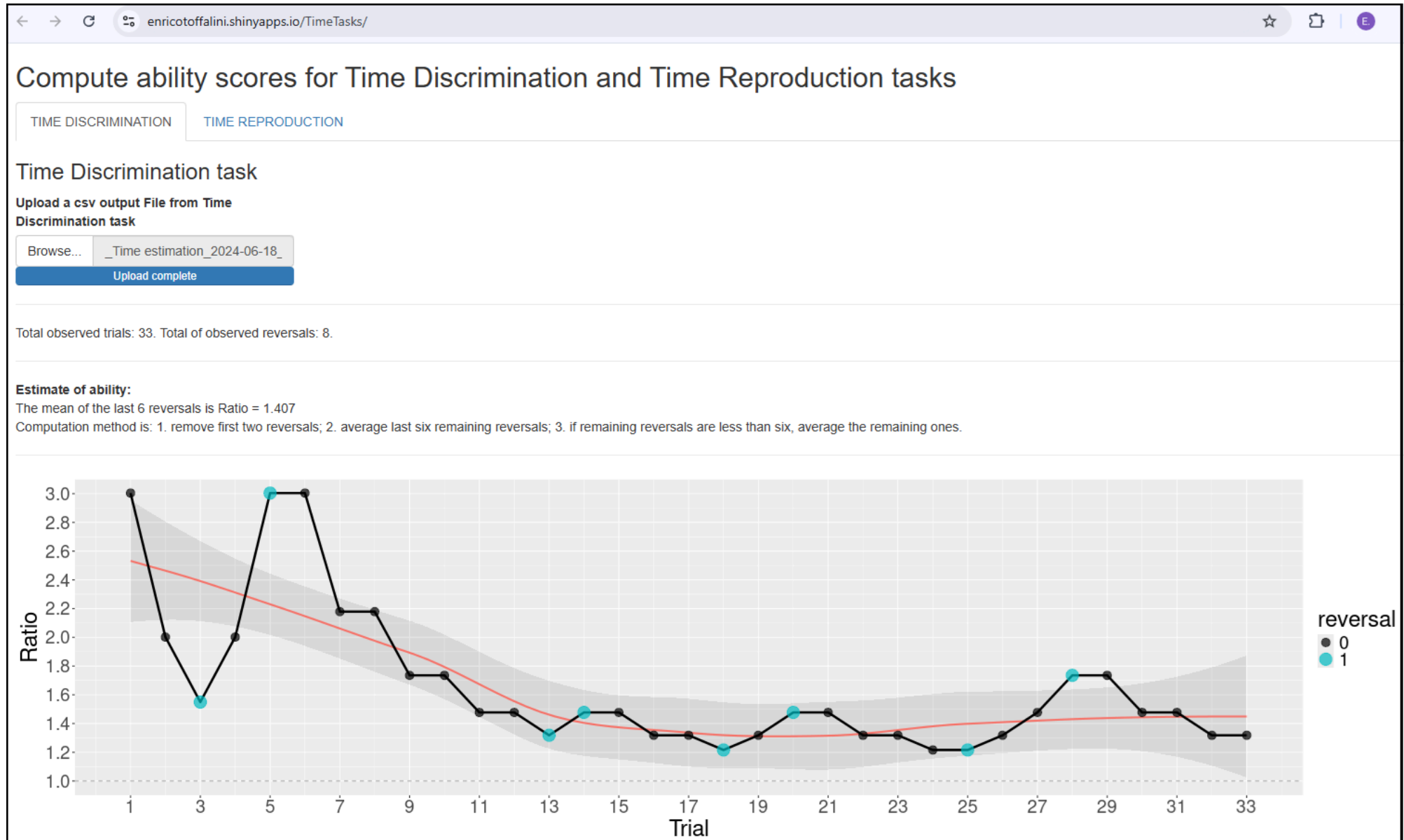
# you may create interactive webapps with Shiny

see [Shiny gallery](#)

here's a couple of recent real examples from **Psicostat** members:

- this [game-like shiny app](#) developed for the science4all event in Padova; see [here](#) some explanation in Italian
- practical [ad-hoc shiny app](#) for scoring experimental data collected by students

# you may create interactive webapps with Shiny



# or entire websites and books

examples of other resources that can be created within the R ecosystem, integrating other tools such as *GitHub* and *Quarto*:

- this very **course support material** is a website in its own right
- this very **course textbook** is a book/website
- **this book** by Daniël Lakens explaining Statistical Inference

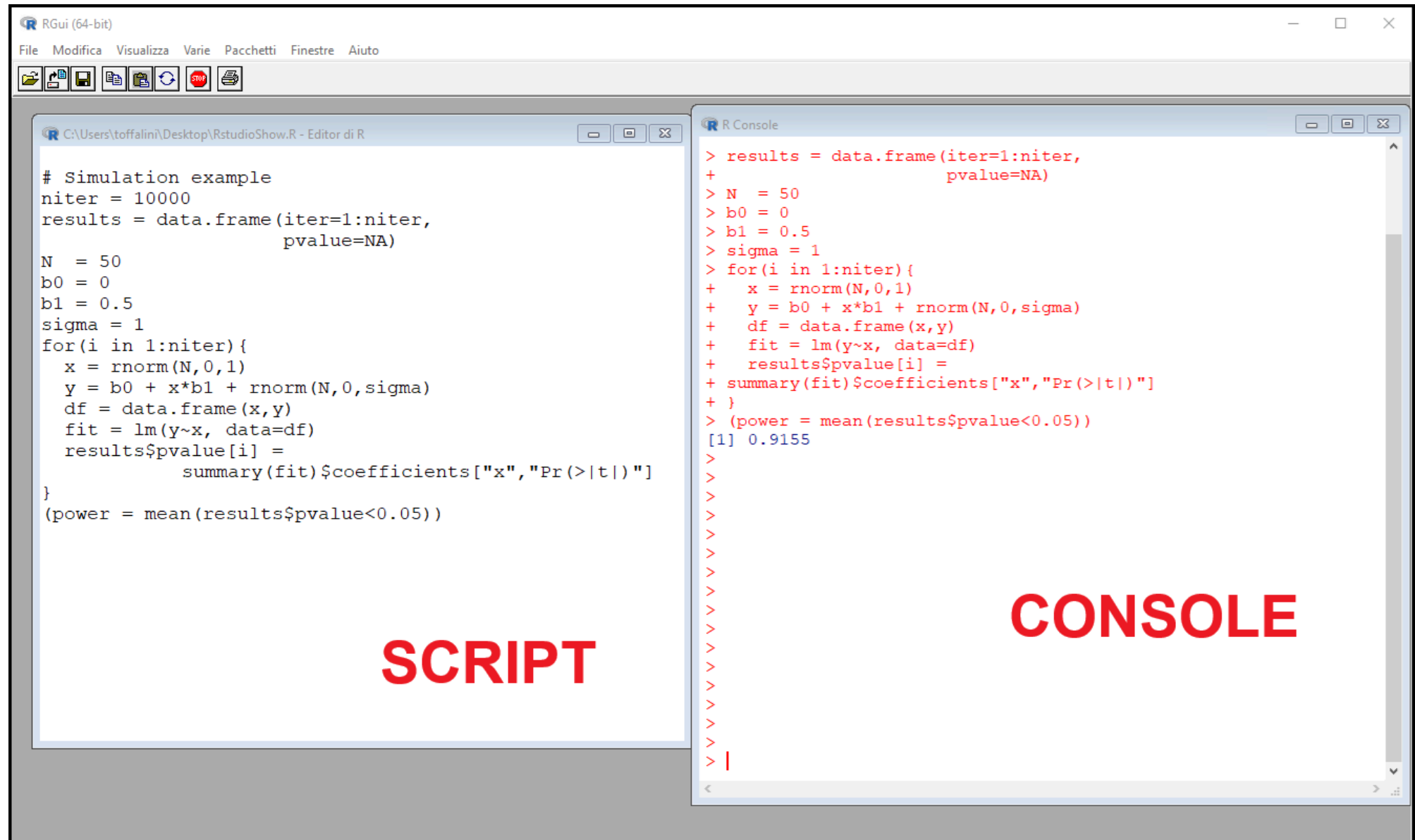
# install R and Rstudio

first of all, for getting started, follow the instructions in *Chapter 1* of *Introduction2R* to ensure that both

- **R** (the basic software itself to run the programming language) and
- **RStudio** (as the IDE of election)

are installed

## R Console (just base R)



# R Studio (full IDE)

The screenshot displays the R Studio IDE interface with four main panes:

- SCRIPT**: The top-left pane shows an R script for a simulation example. The code defines parameters (niter, N, b0, b1, sigma) and a loop that generates data, fits a linear model, and calculates the power of a hypothesis test.
- CONSOLE**: The top-right pane shows the execution output of the script, including the calculated power (0.9105).
- ENVIRONMENT**: The bottom-left pane shows the current environment with variables df, fit, and results, along with their data types and sizes.
- FILE EXPLORER, ETC.**: The bottom-right pane shows the file explorer with a list of files and folders, including faviconpsicostat2.png, psicostatLogo.png, and RstudioShow.R.

The R script in the Script pane is as follows:

```
1  
2 # Simulation example  
3 niter = 10000  
4 results = data.frame(iter=1:niter,  
5                       pvalue=NA)  
6  
7 N = 50  
8 b0 = 0  
9 b1 = 0.5  
10 sigma = 1  
11 for(i in 1:niter){  
12   x = rnorm(N,0,1)  
13   y = b0 + x*b1 + rnorm(N,0,sigma)  
14   df = data.frame(x,y)  
15   fit = lm(y~x, data=df)  
16   results$pvalue[i] = summary(fit)$coefficients["x","Pr(>|t|)"]  
17 }  
18 (power = mean(results$pvalue<0.05))  
19
```

The console output is as follows:

```
> }  
> (power = mean(results$pvalue<0.05))  
[1] 0.918  
> # Simulation example  
> niter = 10000  
> results = data.frame(iter=1:niter,  
+                       pvalue=NA)  
> N = 50  
> b0 = 0  
> b1 = 0.5  
> sigma = 1  
> for(i in 1:niter){  
+   x = rnorm(N,0,1)  
+   y = b0 + x*b1 + rnorm(N,0,sigma)  
+   df = data.frame(x,y)  
+   fit = lm(y~x, data=df)  
+   results$pvalue[i] = summary(fit)$coefficients["x","Pr(>|t|)"]  
+ }  
> (power = mean(results$pvalue<0.05))  
[1] 0.9105  
>
```

The Environment pane shows the following variables:

Variable	Value
df	50 obs. of 2 variables
fit	List of 12
results	10000 obs. of 2 variables
b0	0
b1	0.5
i	10000L
N	50
niter	10000
power	0.9105

The File Explorer pane shows the following files and folders:

Name	Size	Modified
..		
faviconpsicostat2.png	79.8 KB	Oct 31, 2024, 10:25 AM
psicostatLogo.png	264 KB	Mar 4, 2024, 2:05 PM
RstudioShow.R	378 B	Oct 31, 2024, 5:08 PM

## Some R packages that you will or may need in the future (1/3)

Package	Used for what	Example(s) of functions
<code>base</code> (base R)	Basic functions	<code>sum</code> , <code>mean</code> , <code>sqrt</code> , <code>abs</code> , <code>c</code> , <code>data.frame</code> , <code>summary</code> , <code>scale</code> , <code>plot</code> , <code>+</code> , <code>-</code>
<code>stats</code> (base R)	Basic statistical calculations and functions	<code>sd</code> , <code>cor</code> , <code>cor.test</code> , <code>t.test</code> , <code>lm</code> , <code>glm</code> , <code>AIC</code> , <code>rnorm</code> , <code>rbinom</code>
<code>graphics</code> (base R)	Basic statistical calculations and functions	<code>plot</code> , <code>boxplot</code> , <code>hist</code> , <code>barplot</code>

*(You may actually use these “base” packages very often without even realizing that they are packages)*



## Some R packages that you will or may need in the future (2/3)

Package	Used for what	Example(s) of functions
<code>effectsize</code>	Compute different effect sizes	<code>cohens_d</code> , <code>hedges_g</code> , <code>cohens_f</code> , <code>d_to_r</code>
<code>lme4</code>	Fitting (generalized) (non-)linear mixed-effects models	<code>lmer</code> , <code>glmer</code> , <code>ranef</code>
<code>performance</code>	Useful tools for models	<code>check_collinearity</code> , <code>r2_nagelkerke</code> , <code>icc</code>
<code>effects</code>	Display effects for various statistical models	<code>allEffects</code>
<code>emmeans</code>	Estimate marginal means for various models	<code>emmeans</code>

## Some R packages that you will or may need in the future (3/3)

Package	Used for what	Example(s) of functions
<code>ggplot2</code>	Create beautiful plots using The Grammar of Graphics	<code>ggplot</code> , <code>geom_point</code> , <code>geom_line</code> , ...
<code>lavaan</code>	Structural Equation Models (SEM)	<code>sem</code> , <code>cfa</code>
<code>semTools</code>	Useful tools for SEMs	<code>compRelSEM</code> , <code>measEq.syntax</code>
<code>metafor</code>	Perform meta-analysis	<code>rma</code> , <code>rma.mv</code> , <code>forest</code> , <code>funnel</code> , <code>regtest</code>
<code>brms</code>	Fitting practically any Bayesian model via MCMC with STAN	<code>brm</code> , <code>set_prior</code>
<code>blavaan</code>	Fitting Bayesian SEMs	<code>bcfa</code> , <code>bsem</code>

# Let's Test the Environment!

Let's run a few commands in RStudio to familiarize with its console and see if the installation works properly

```
rmnorm(10) # draw 10 random values from a Standard Normal distribution
```

```
[1] -2.9017454  1.6591803 -0.3806936  0.2842570 -2.1319494 -1.1003835  
[7]  1.5540280 -1.5755473 -0.5630551  0.4359361
```

```
?rmnorm # open the help tab for the "rmnorm" function
```

```
round( rmnorm(10, mean=100, sd=15) ) # draw 10 values from IQ distribution, round them
```

```
[1]  78 101  87  86 114  87  91  83 116 111
```

```
install.packages("psych") # install a package from CRAN
```

```
library(psych) # load the newly installed package
```

```
fisherz(rho=0.5) # use it to transform a correlation into a Fisher's z
```

```
[1] 0.5493061
```