



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

Enrico Toffalini

# Why becoming an R user?

- **Mainstream in academia** for statistical computing and data science, increasingly used in business. *Job market advantage!*
- **Free & open-source:** wherever you go, R will be with you at no costs (unlike *MATLAB, MPLUS, 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 (plus *chatGPT*, *Gemini*, *Lucrez-IA*, etc., know it pretty well!).
- **Huge ecosystem**, >23,000 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*]); also, can integrate with *Quarto*, *GitHub*.
- Facilitates **reproducible scientific research** by sharing code and workflows.

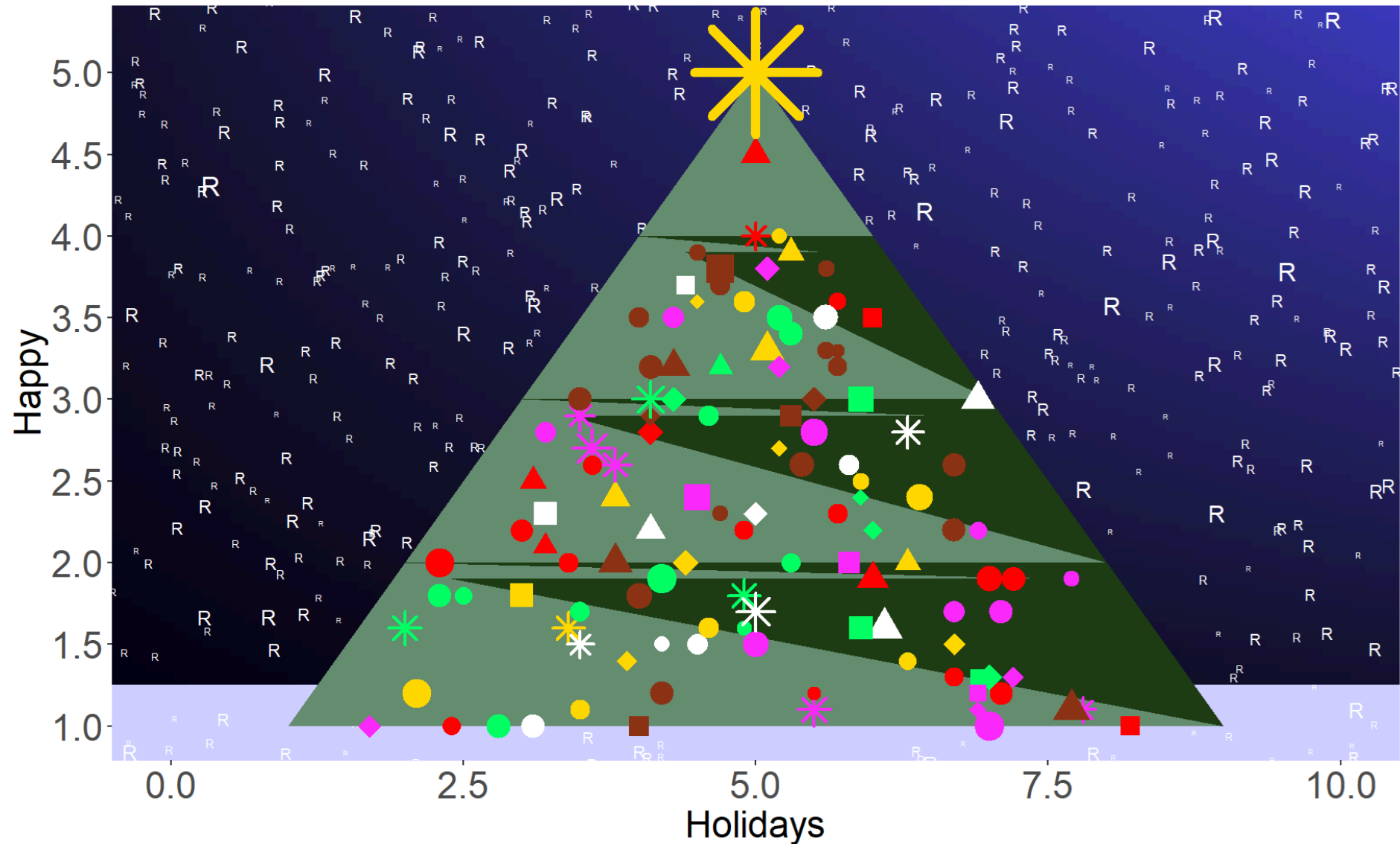
# What you may expect to learn in this course:

- 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.

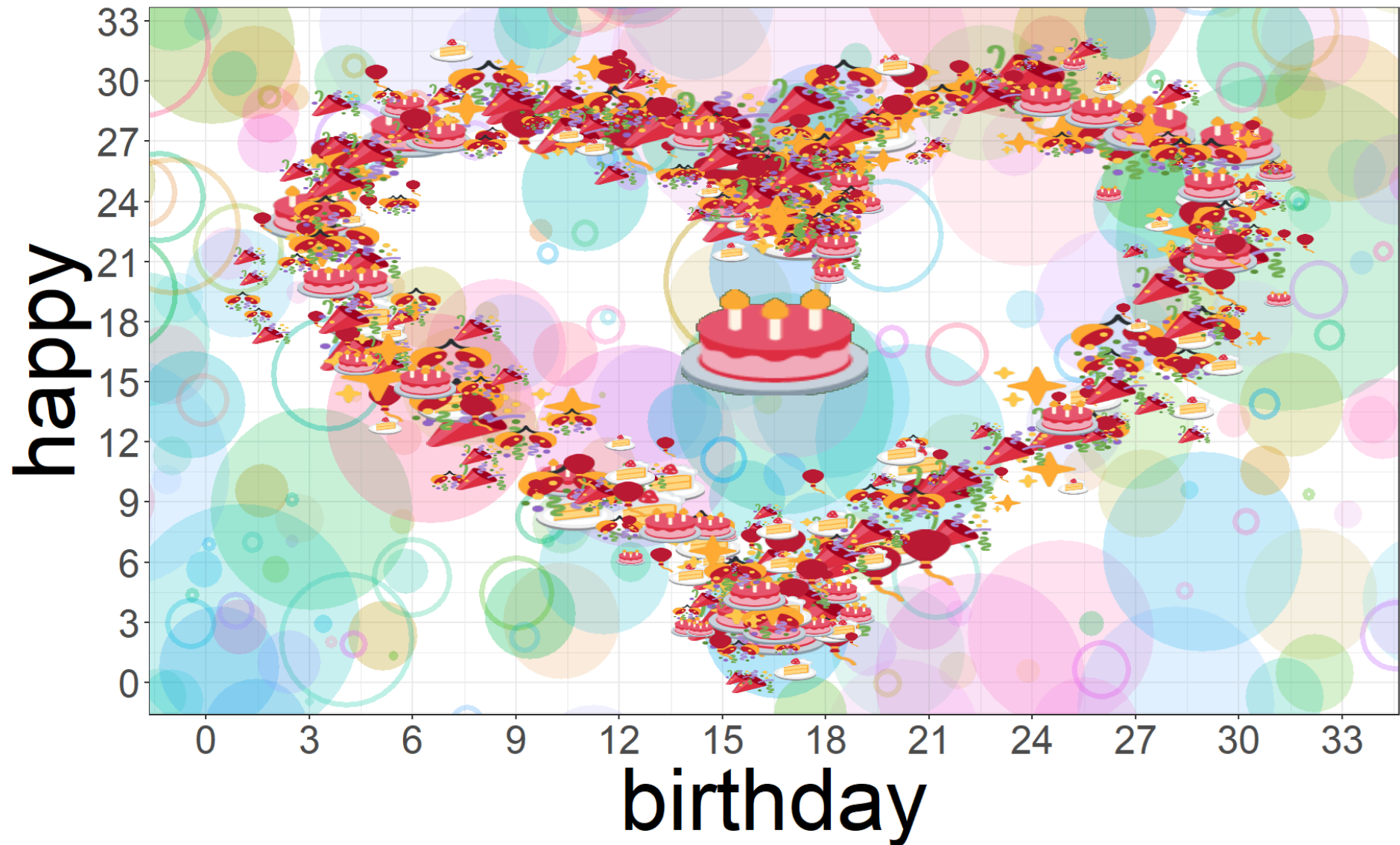
Over the next two years, following this PhD program, you will or may have the opportunity to use R to perform at least some fundamentals about:

- Core statistical inference methods;
- LM/LMM/GLMM: (Generalized) linear (mixed-effects) models;
- Data visualization using *ggplot2*;
- Power analysis & more via **data simulation**;
- **SEM**: Structural Equation Modeling;
- Conducting **meta-analysis**.

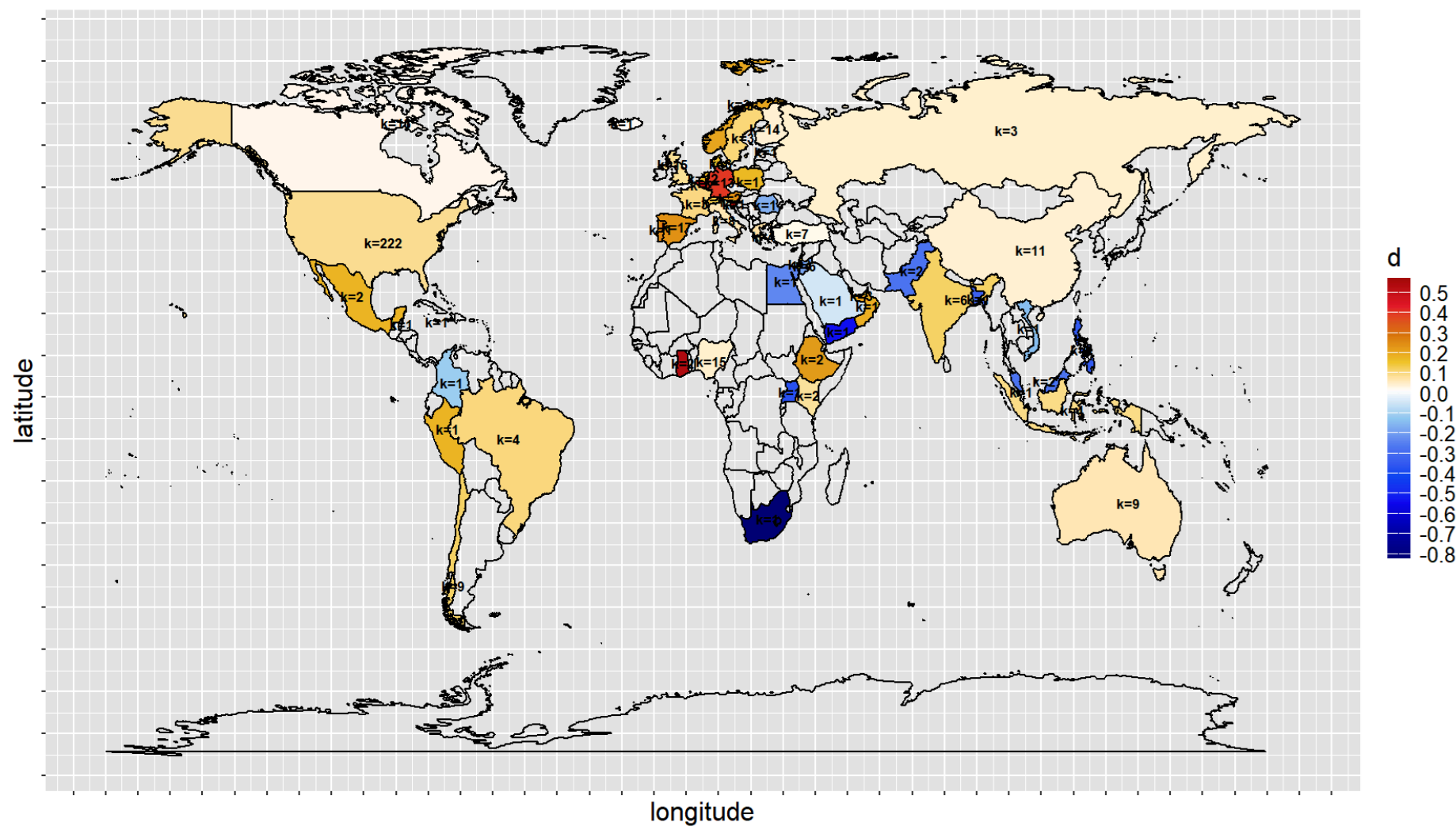
you may even create greeting cards



you may even create greeting cards

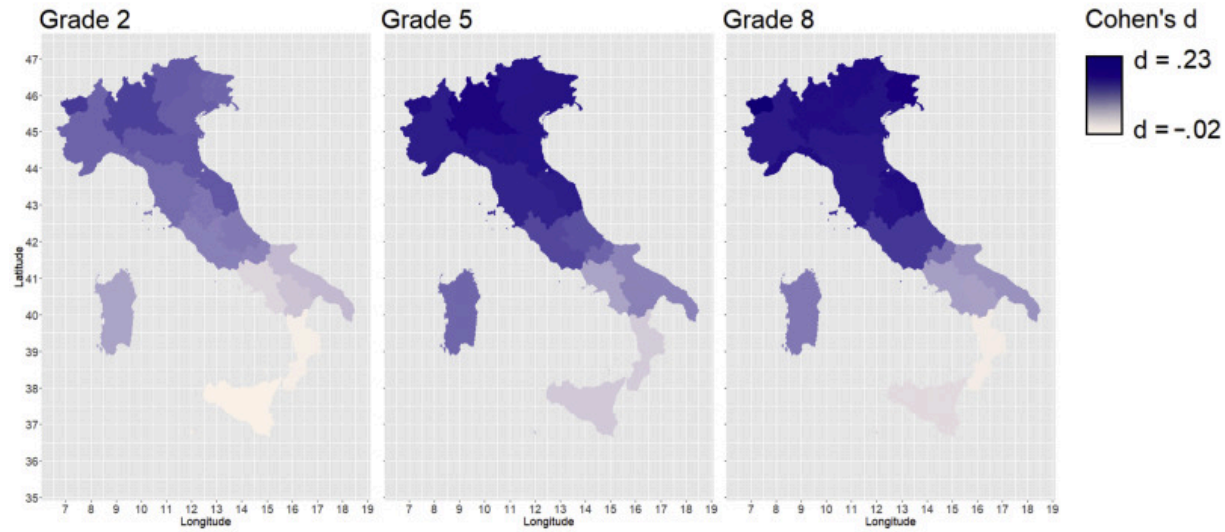


or like fancy infographics

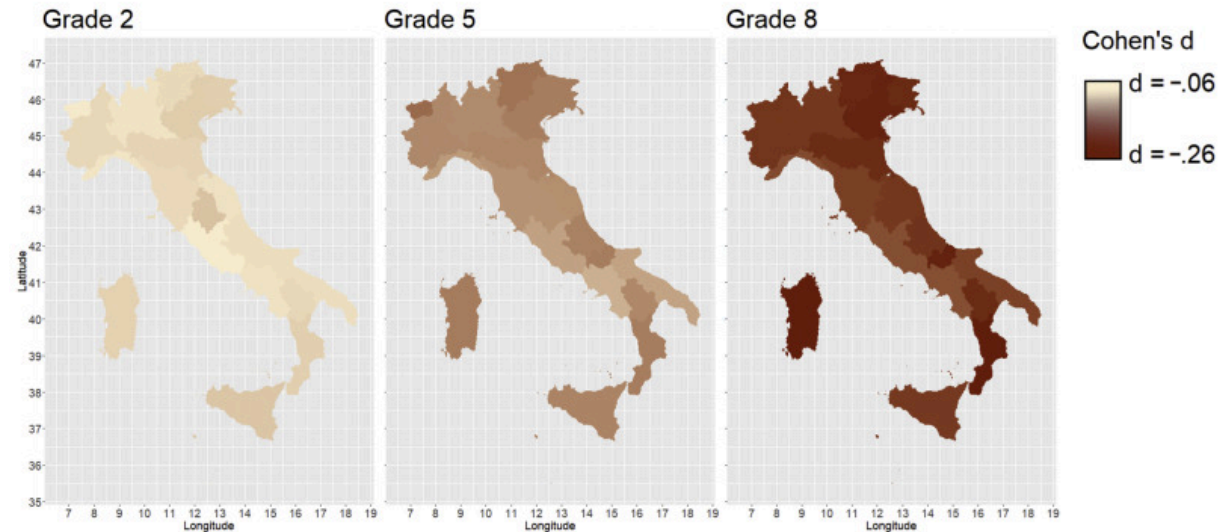


# or like fancy infographics

**A.**

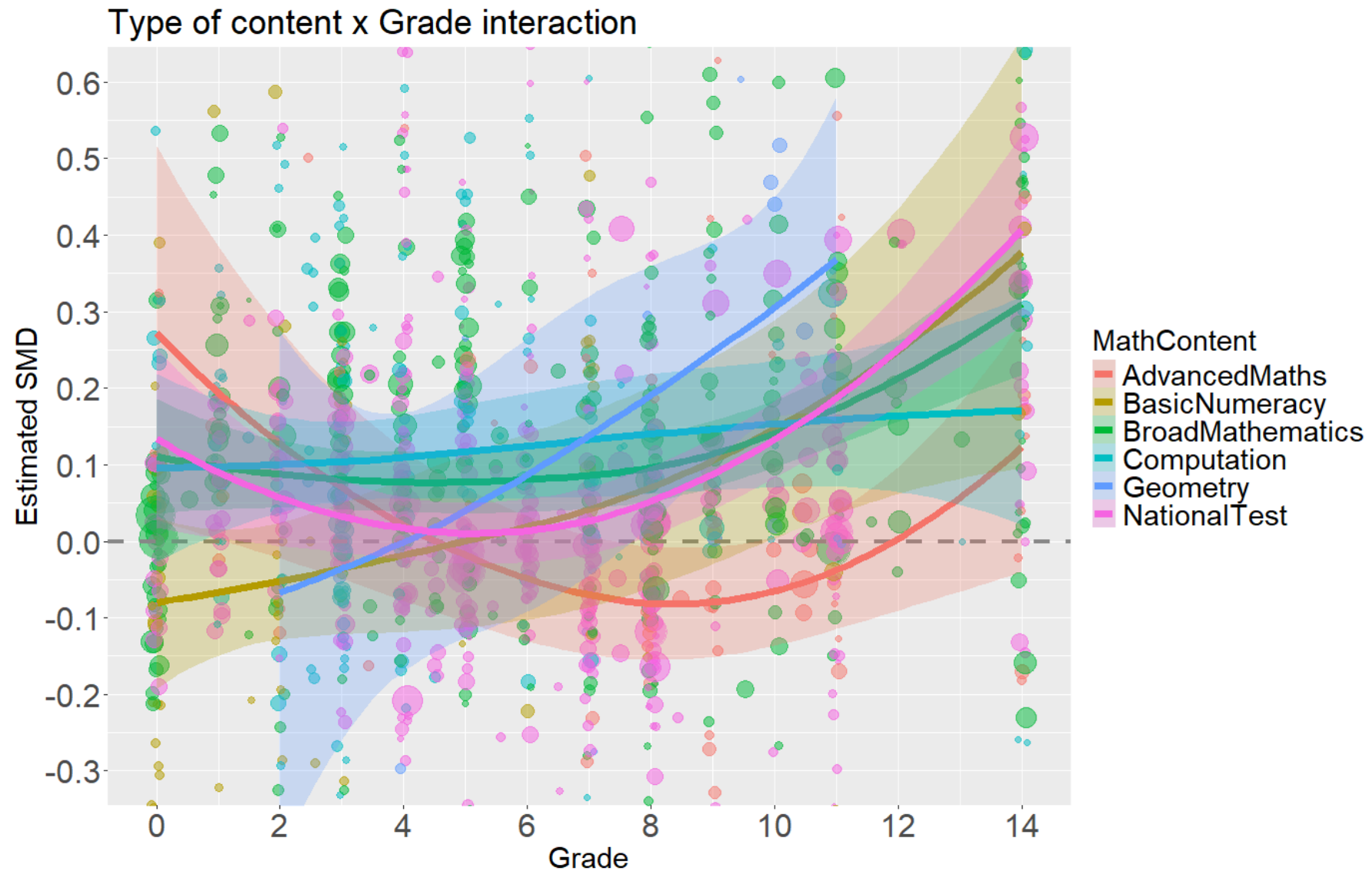


**B.**





# perform fancy moderated meta-analyses



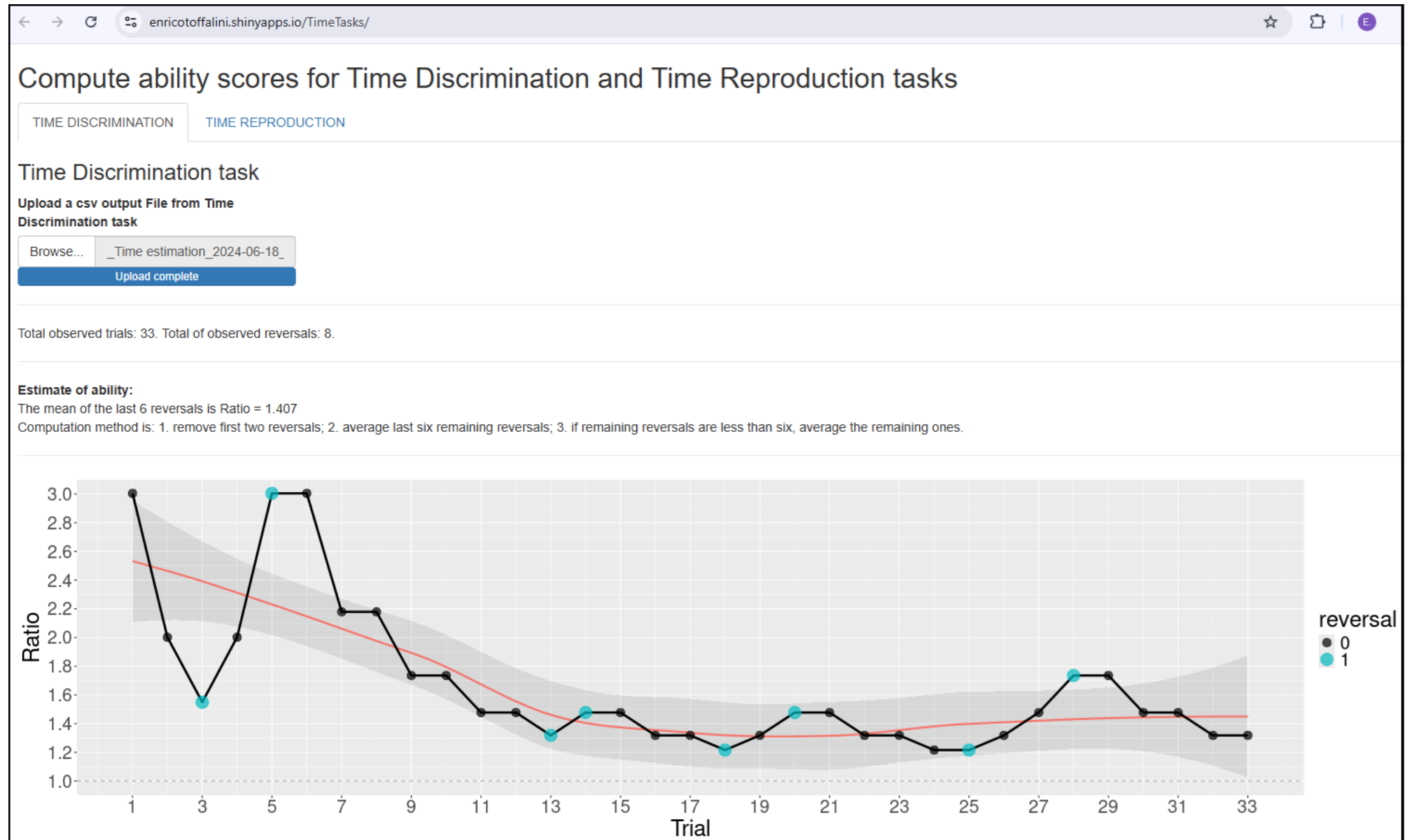
# you may create interactive webapps with Shiny

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

# R + *Integrated Development Environment(s)*

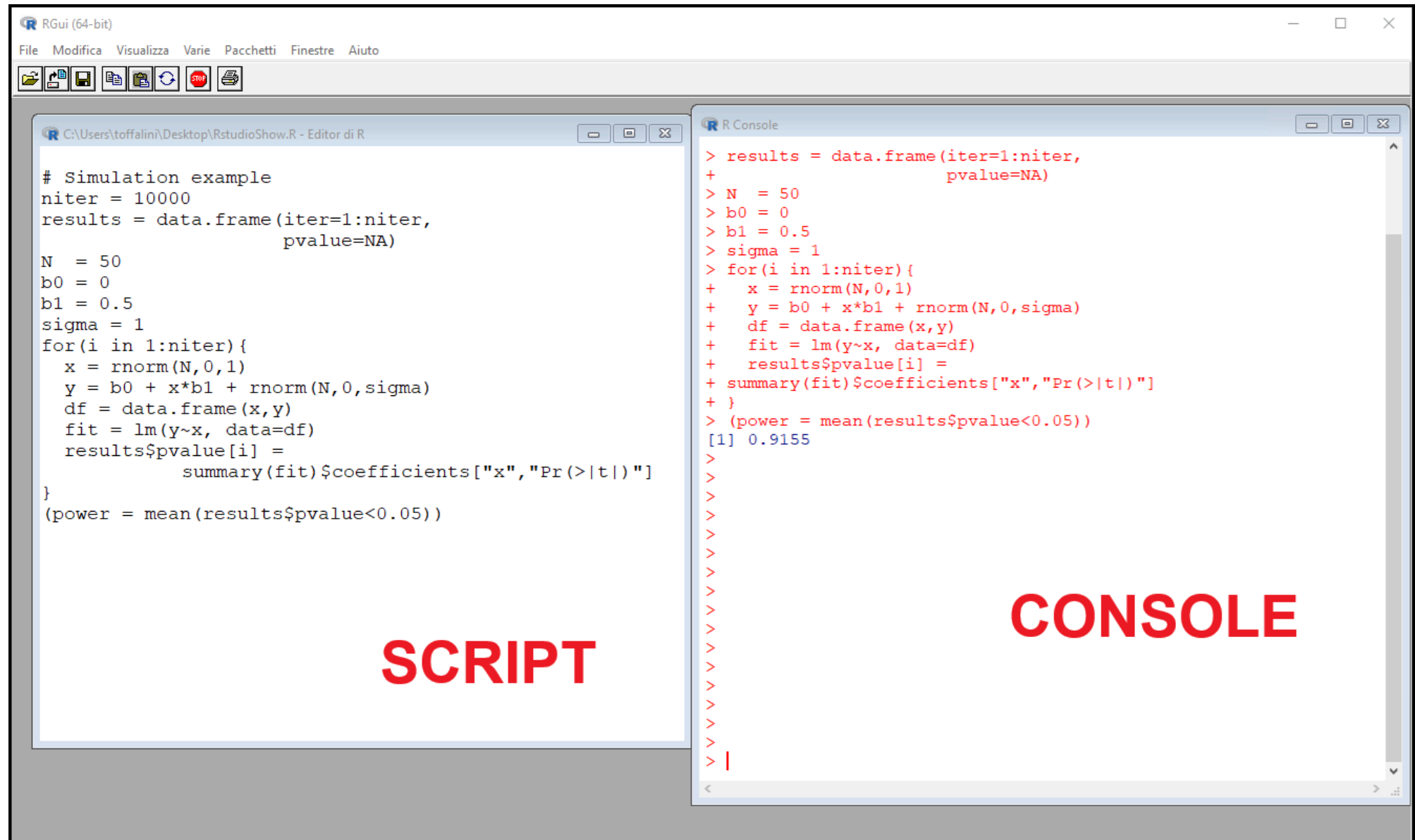
Make sure you install:

- **R** as the programming language interpreter and the basic environment and packages
- **RStudio** is the IDE of election to make writing R code easy

Interesting alternatives to installing RStudio:

- **Positron** (based on MS Visual Studio)
- **Posit.cloud** (fully online, actually RStudio)
- **Google Colab** (fully online, make sure to set *R runtime type*; actually a Jupyter notebook)

# 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 R code for a simulation example. The code defines parameters (N=50, b0=0, b1=0.5, sigma=1, niter=10000) 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 output of the R code, including the calculated power (0.918) and the results of the simulation.
- ENVIRONMENT**: The bottom-left pane shows the current environment with variables: 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), and results (1).
- FILE EXPLORER, ETC.**: The bottom-right pane shows the file explorer with a list of files: faviconpsicostat2.png (79.8 KB, Oct 31, 2024, 10:25 AM), psicostatLogo.png (264 KB, Mar 4, 2024, 2:05 PM), and RstudioShow.R (378 B, Oct 31, 2024, 5:08 PM).

The R Studio interface includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help) and a toolbar with icons for running code, saving files, and other functions. The status bar at the bottom shows the current file (RstudioShow.R) and the system clock (17:08, 31/10/2024).

# Google Colab (online notebook)

The screenshot shows a Google Colab notebook interface with several red annotations:

- shareable via link like any Google document**: Points to the 'Share' button in the top right corner.
- add stuff**: Points to the '+ Code' and '+ Text' buttons in the top toolbar.
- execute**: Points to the 'Run all' button in the top toolbar.
- HEADING**: Points to the notebook title 'Simple Power Simulation Example - R'.
- paragraph**: Points to the text 'First of all, define parameters and pre-allocate dataframe for results.'
- R chunk**: Points to the first code cell containing R code for parameter initialization.
- paragraph**: Points to the text 'Now run the simulation loop, and store result (p-value) at each iteration.'
- R chunk**: Points to the second code cell containing a for-loop for the simulation.
- remember to set R**: Points to the bottom right corner of the notebook, indicating the R environment.
- paragraph**: Points to the text 'Finally, compute and inspect empirical power'.
- R chunk with final result**: Points to the third code cell, which calculates the empirical power and displays the result '0.9166'.
- inspect use of resources**: Points to the 'RAM' and 'Disk' usage indicators in the top right corner.

The notebook content includes the following R code chunks:

```
[4] niter = 10000
     results = data.frame(iter = 1:niter, pvalue = NA)
     N = 50; b0 = 0; b1 = 0.5; sigma = 1
```

```
[5] for(i in 1:niter){
     x = rnorm(N,0,1)
     df = data.frame(
       x = x,
       y = b0 + x*b1 + rnorm(N,0,sigma)
     )
     fit = lm(y ~ x, data = df)
     results$pvalue[i] = summary(fit)$coefficients["x", "Pr(>|t|)"]
   }
```

```
[6] (power = mean(results$pvalue < 0.05))
     ... 0.9166
```



# 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] 2.1893126 -0.3920248 0.6233079 -0.3044554 1.0206581 -0.8628790  
[7] 0.9408118 1.1756534 0.6488329 -0.9160198
```

```
?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] 89 89 75 104 84 59 111 92 94 77
```

```
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
```