



# **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.

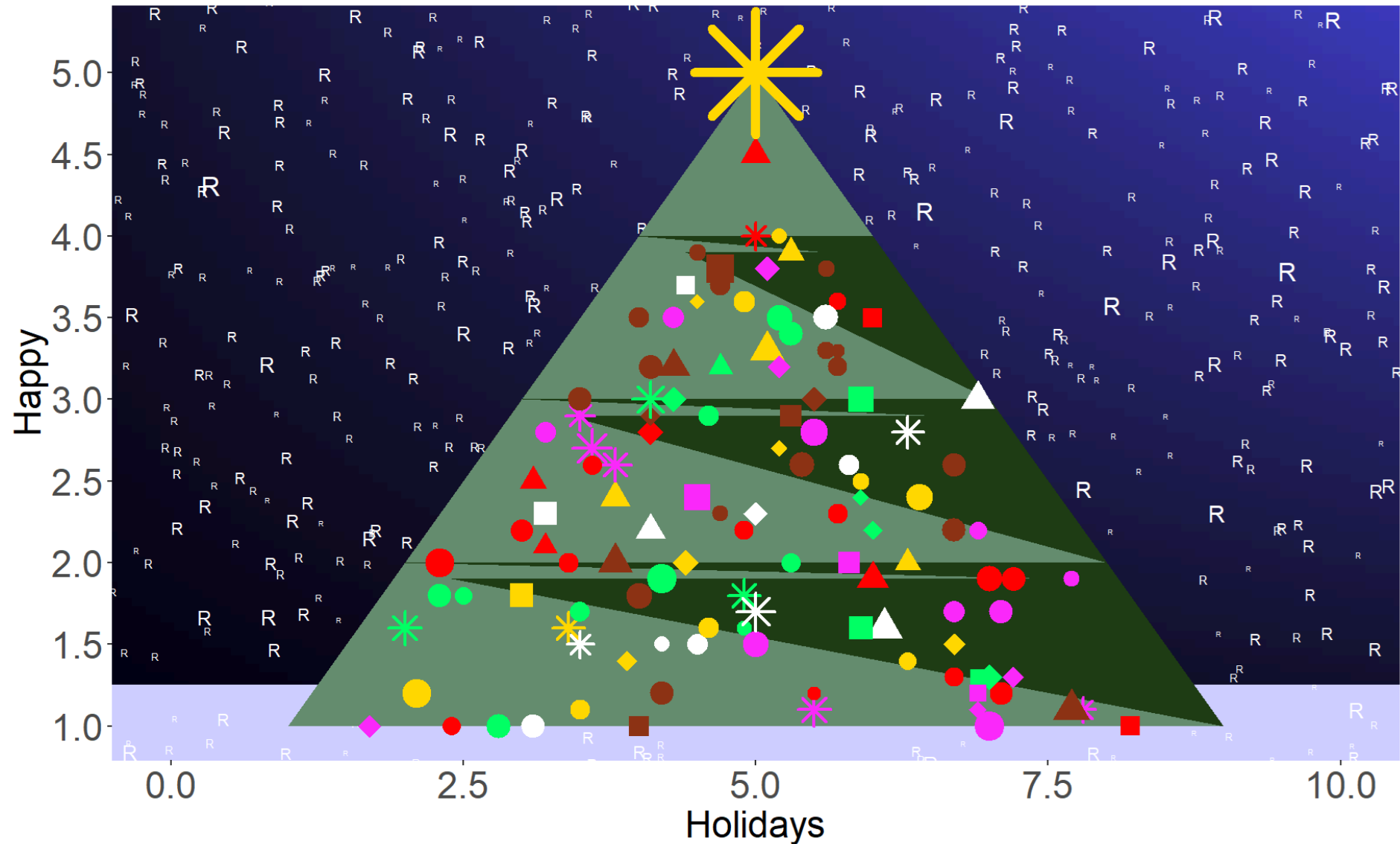
# What you may expect to learn in this course:

- 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.

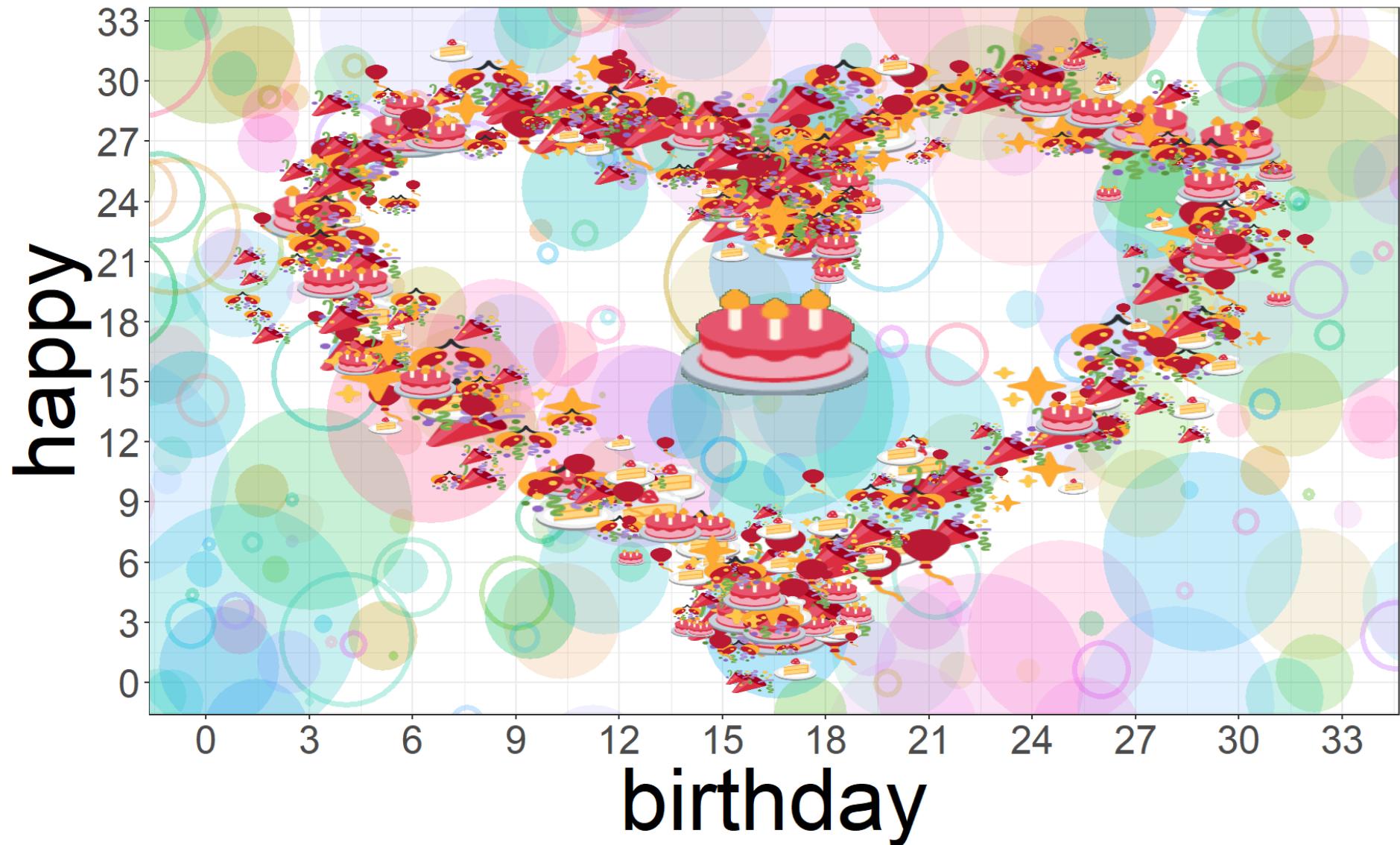
Over the next two years, following this PhD program, you will or may have the opportunity to use R to perform at least some fundamentals about:

- 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**.

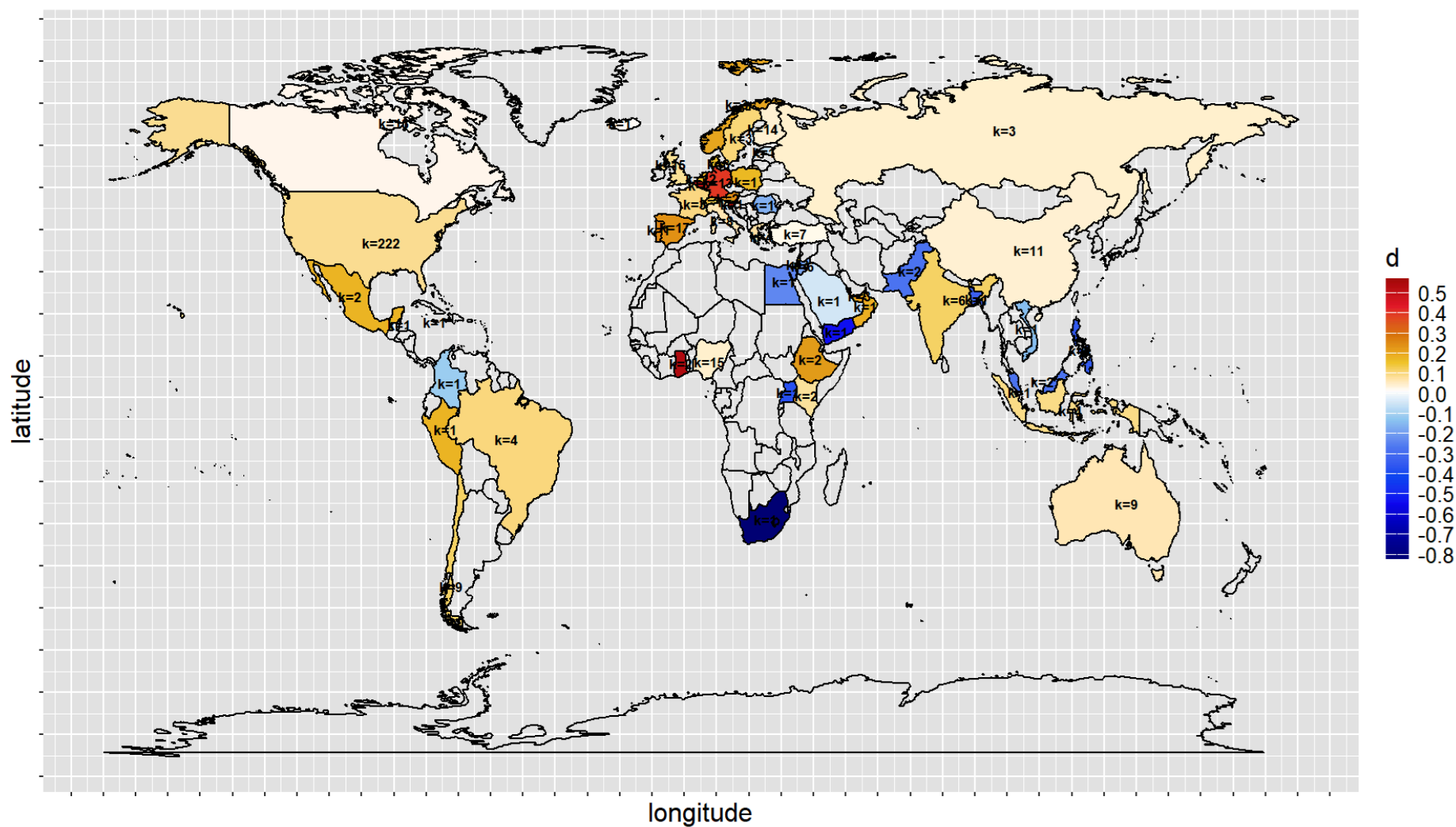
# you may even create greeting cards



you may even create greeting cards

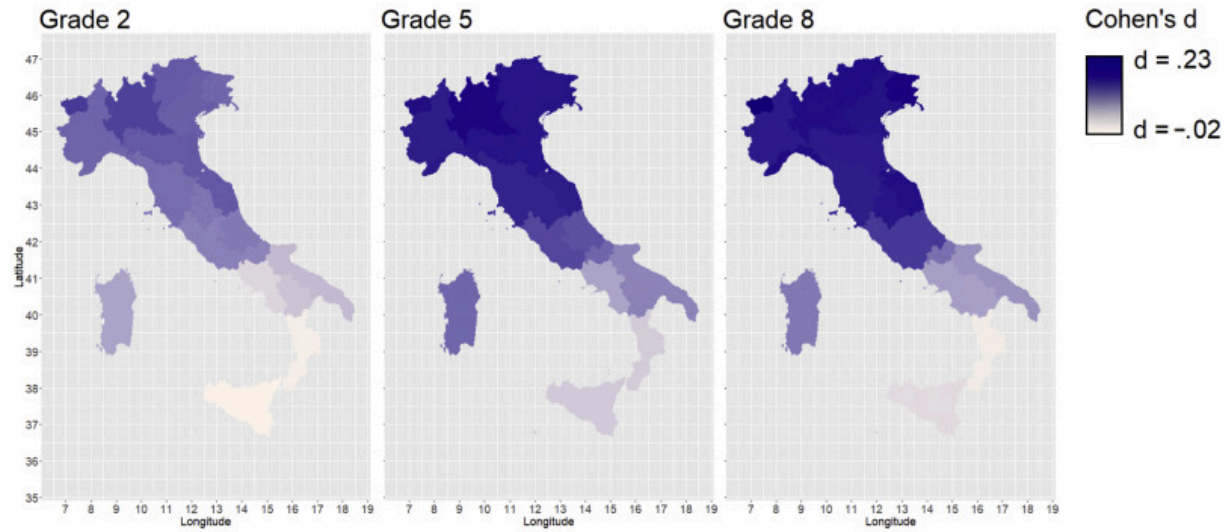


or like fancy infographics

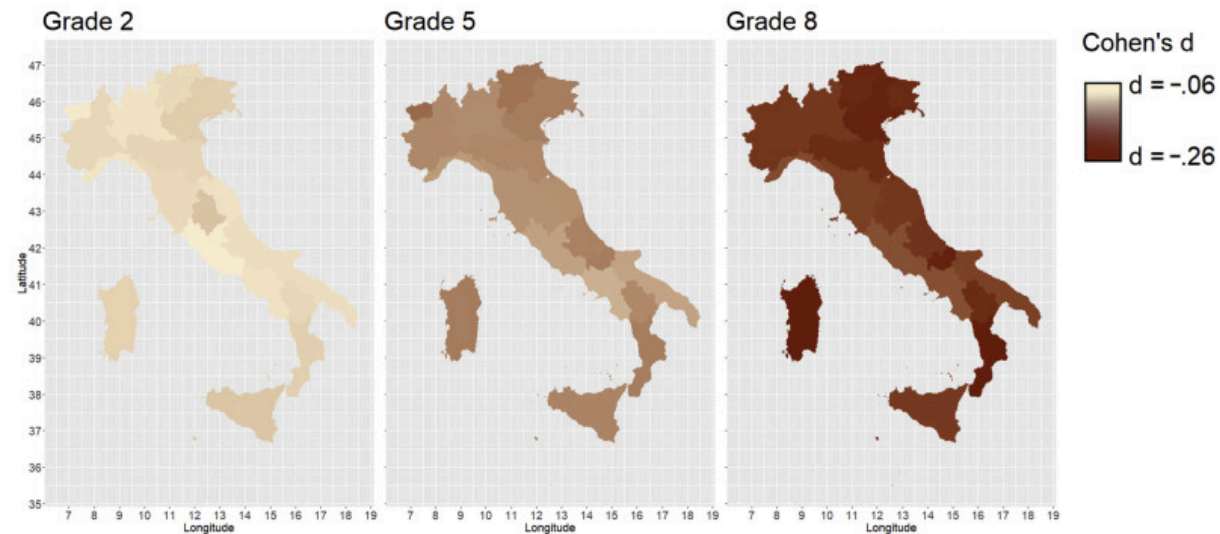


# or like fancy infographics

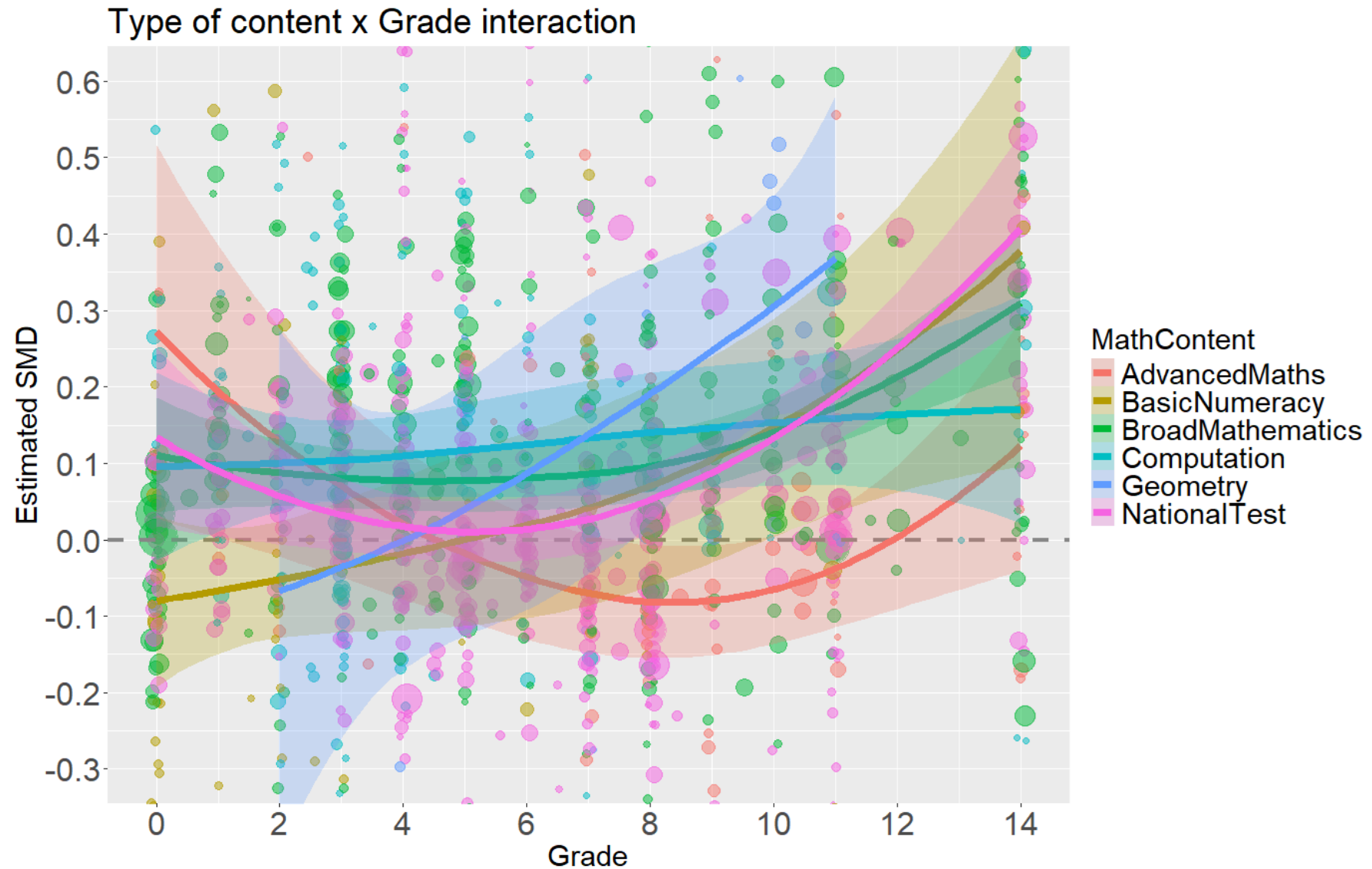
**A.**



**B.**



# perform fancy moderated meta-analyses



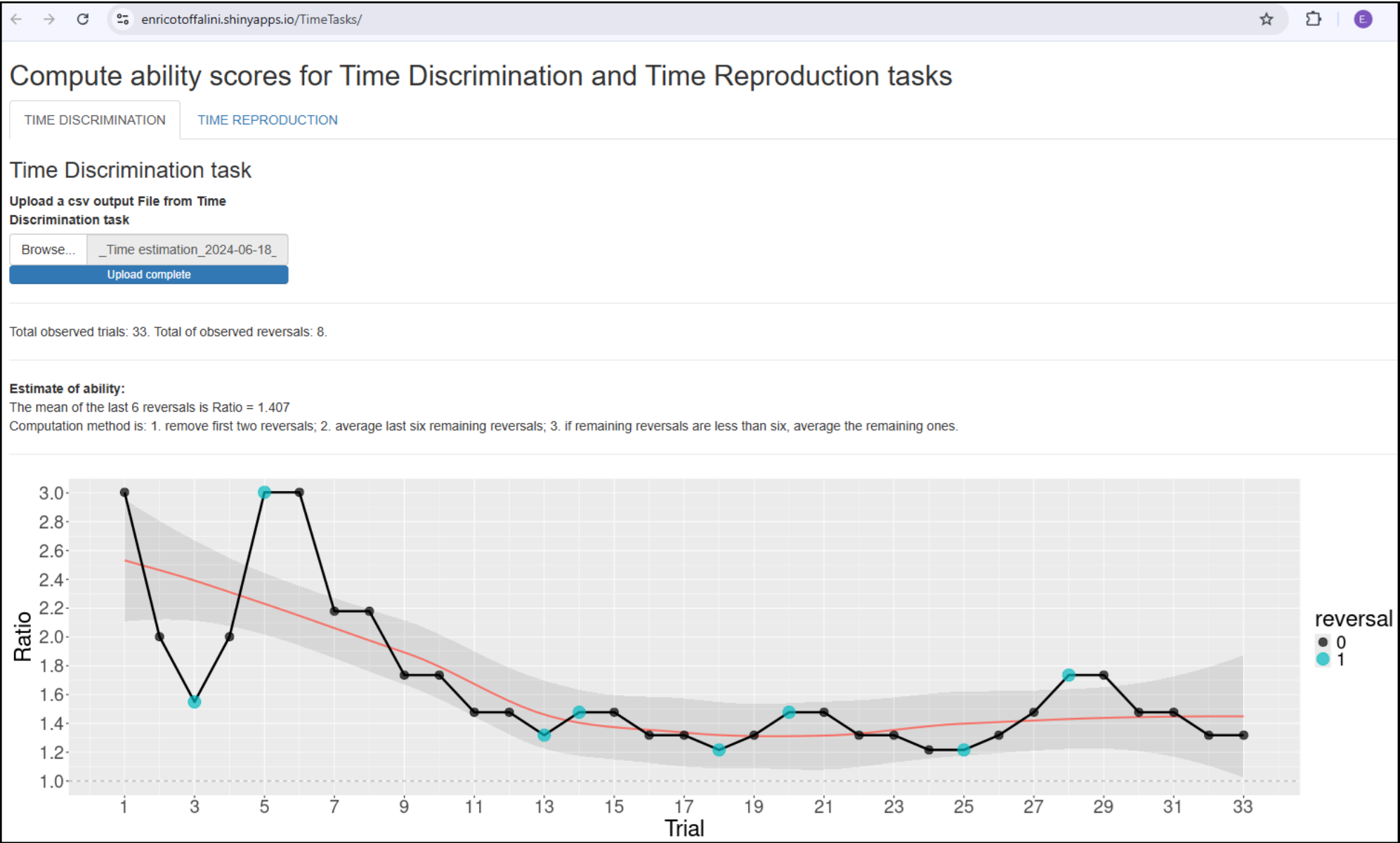
# you may create interactive webapps with Shiny

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)*

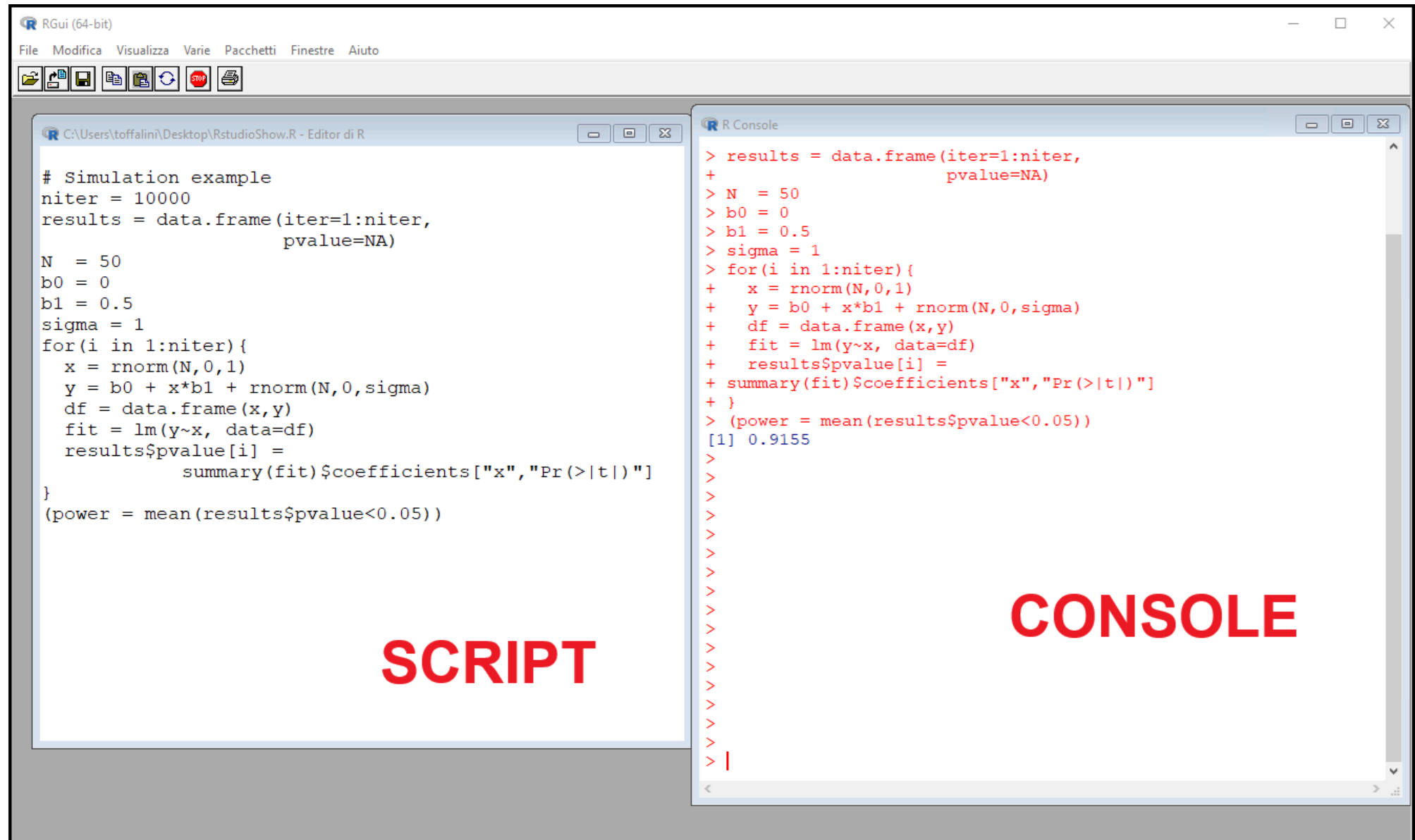
Make sure you install:

- **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 Code)
- **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 an R script for a simulation example. The code defines parameters (N=50, niter=10000, b0=0, b1=0.5, sigma=1) and performs a linear regression simulation using `rnorm` and `lm` functions. The script calculates the power of the test as the mean of p-values less than 0.05.
- CONSOLE**: The top-right pane shows the execution output of the script. It displays the calculated power value of 0.9105.
- ENVIRONMENT**: The bottom-left pane shows the current environment variables. It lists `df` (50 obs. of 2 variables), `fit` (List of 12), and `results` (10000 obs. of 2 variables). The values for `b0`, `b1`, `i`, `N`, `niter`, `power`, and `sigma` are also displayed.
- FILE EXPLORER, ETC.**: The bottom-right pane shows the file explorer view. It displays the current directory structure and lists files such as `faviconpsicostat2.png`, `psicostatLogo.png`, and `RstudioShow.R`.

# 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 area.
- paragraph**: Points to the text 'Finally, compute and inspect empirical power'.
- R chunk with final result**: Points to the third code cell, which is highlighted with a blue border and contains the command to calculate and display the empirical power.
- 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] 0s
niter = 10000
results = data.frame(iter = 1:niter, pvalue = NA)
N = 50; b0 = 0; b1 = 0.5; sigma = 1
```

```
[5] 13s
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] 0s
(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]  0.63285715 -0.09089180  0.05989258 -1.98495842 -1.39771385 -1.76343919  
[7] -1.37670755 -0.97808745  0.72135449  0.59570175
```

```
?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] 108 103  81  89 101 121 100 104  97 101
```

```
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
```