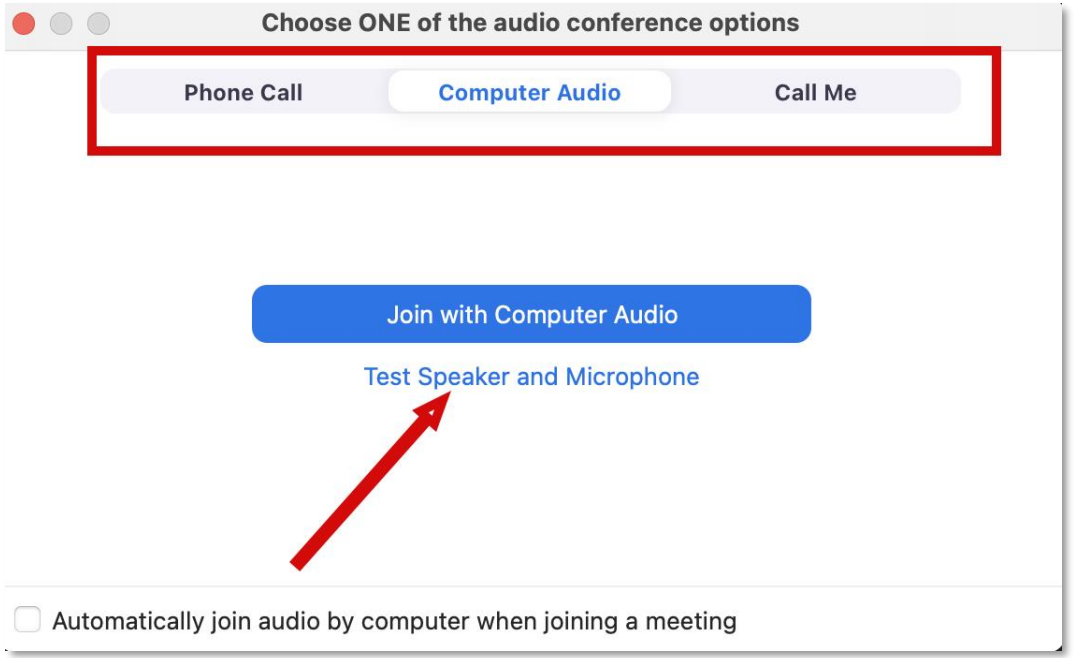
Audio Connection and Chat (Zoom)

You will not hear any sound until the webinar starts.

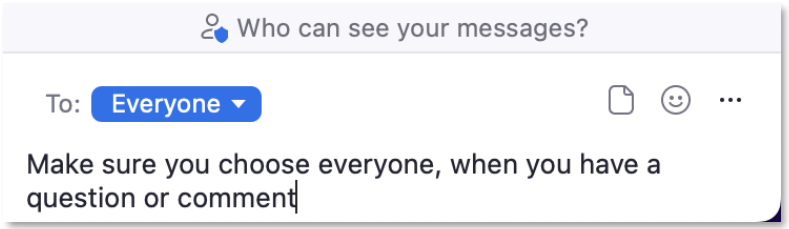
Connect Audio

1. When you join Zoom, the ***Join Audio*** preferences box pops-up (Phone Call, Computer Audio, or Call Me)
2. Choose an option that works best for you
3. Join using that option
4. Use Test Speakers and Microphone option to optimize your webinar experience



Chat

Please send your chat to ***Everyone*** to make sure the monitor sees your question



National Institutes of Health
Office of Management

Introduction to R and RStudio

Joelle Mornini, MLS

January 14, 2025

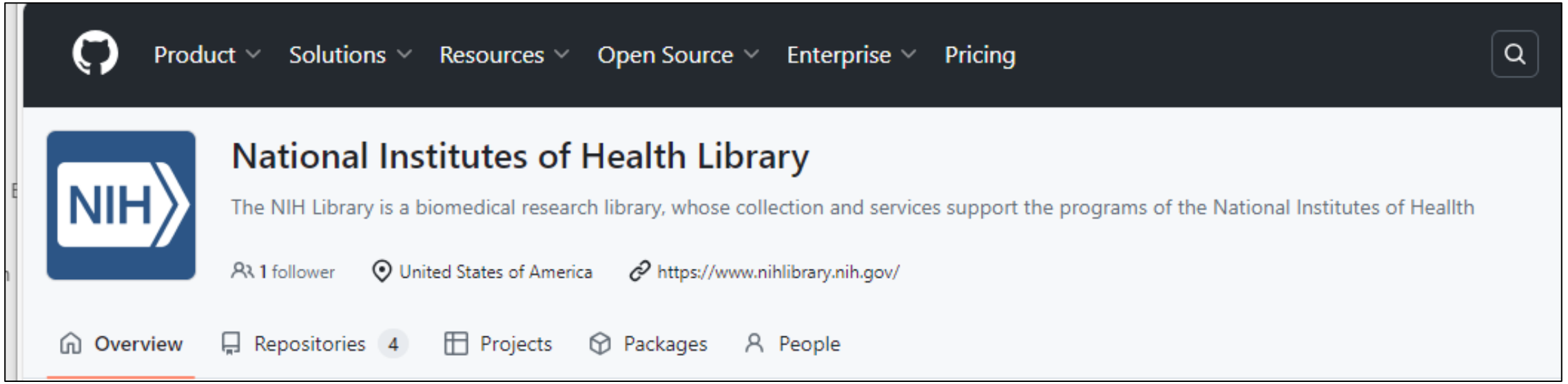
Office of Research Services
National Institutes of Health
U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES

Other Upcoming R Classes

- **Other upcoming R classes on the NIH Library Training Calendar:**
  - Data Wrangling Workshop - Jan 16, 2025 (1:00 pm - 3:00 pm)
  - Data Visualization in R: Introduction to ggplot Part 1 of 2 - Jan 21, 2025 (1:00 pm - 2:00 pm)
  - Data Visualization in R: Customizations Part 2 of 2 - Jan 23, 2025 (1:00 pm - 2:30 pm)
  - Introduction to Quarto for Scholarly Publishing - Feb 10, 2025 (10:00 am - 11:30 am)

Class Materials on NIH Library GitHub

- Visit the NIH Library GitHub to find class materials for R and Python classes
  - Intro to R and RStudio class materials



Objectives

- **After completing this training, you will be able to:**
  - Describe the purpose of R and RStudio
  - Organize files and directories for a set of analyses as an R Project
  - Define key terms as they relate to R: object, assign, comment, call, function, and arguments
  - Find help and learning resources related to R and RStudio

Elements of this training are from the Introduction to R episode of the Data Analysis and Visualization in R for Ecologists lesson from Data Carpentry. (Copyright (c) Data Carpentry)

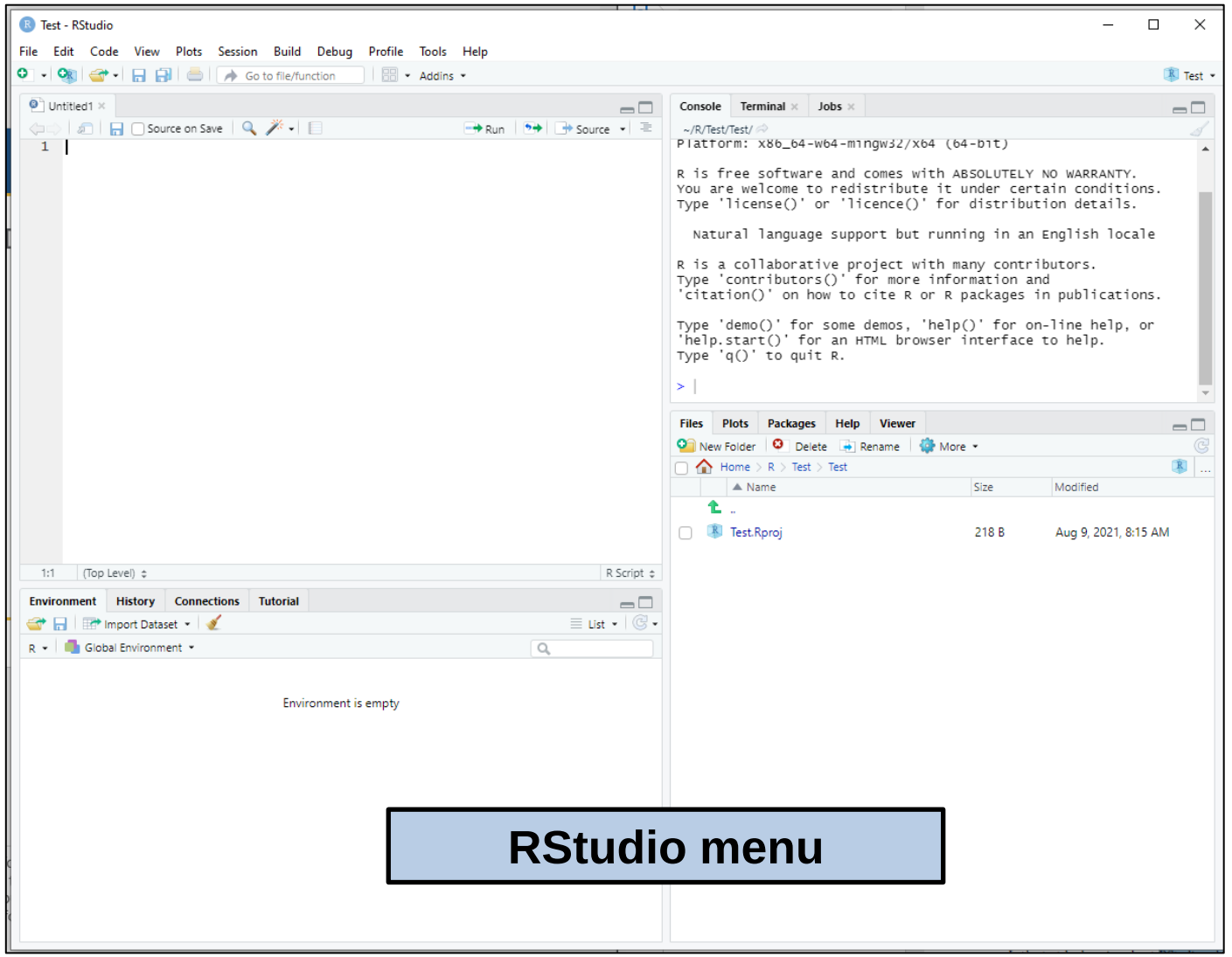
What is R and RStudio?

What is R?

- **R:** Both the **programming language** and the **software** that interprets the scripts
- A language and environment for statistical computing and graphics
  - Similar to S language
  - Source: What is R? from the R Foundation

What is RStudio?

- **RStudio:** An Integrated Development Environment (IDE) for working with R and Python, distributed by Posit
- RStudio needs R to function correctly, so both R (recommended 4.4.0 or later) and RStudio should be installed together



Why Learn R?

- R doesn’t involve lots of pointing and clicking
- R code is great for reproducibility
- R is interdisciplinary and extensible
- R works on data of all shapes and sizes
- R produces high-quality graphics
- R is free, open-source and cross-platform

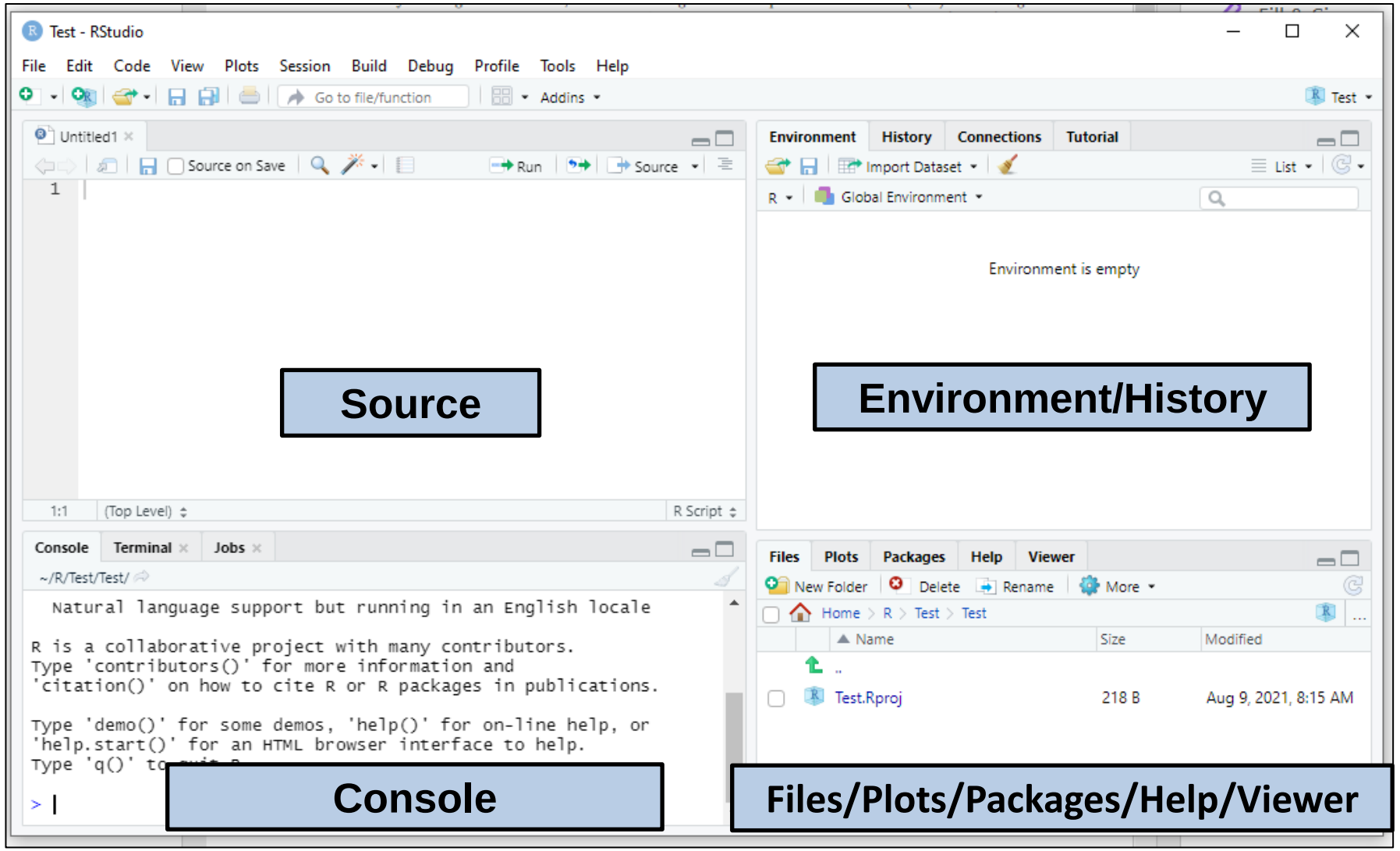
Poll: Why do you want to learn R?

A. To create visualizations
B. For bioinformatics analysis
C. For statistical analysis
D. For data cleaning
E. For formatting/publishing
F. Other (type in chat)

RStudio Overview

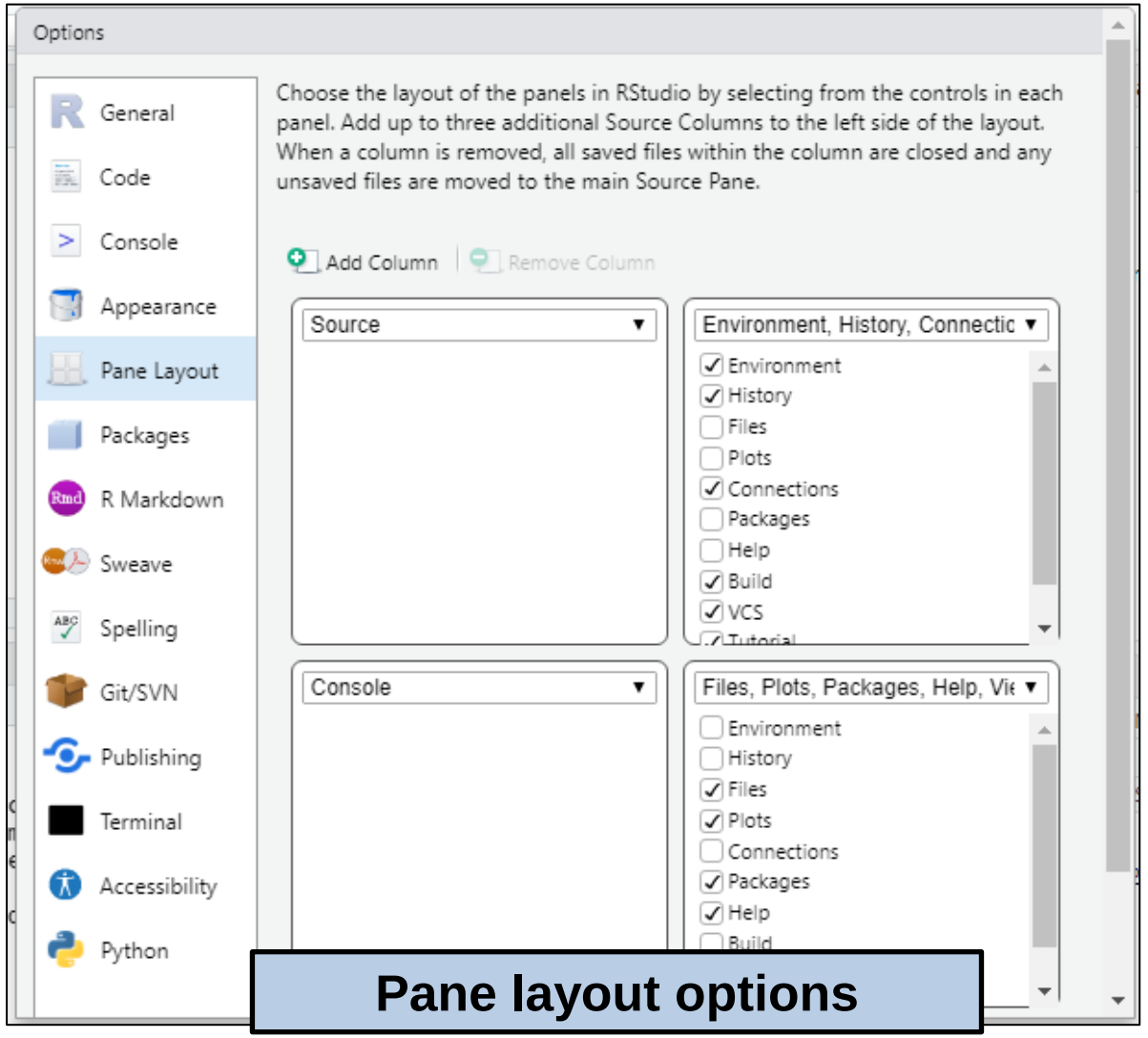
4 Panes of RStudio

- Reference document (PDF): [RStudio IDE Cheat sheet]



4 Panes of RStudio: Customizing Pane Layout

- The placement of these panes and their content can be customized
  - See menu Tools -> Global Options -> Pane Layout

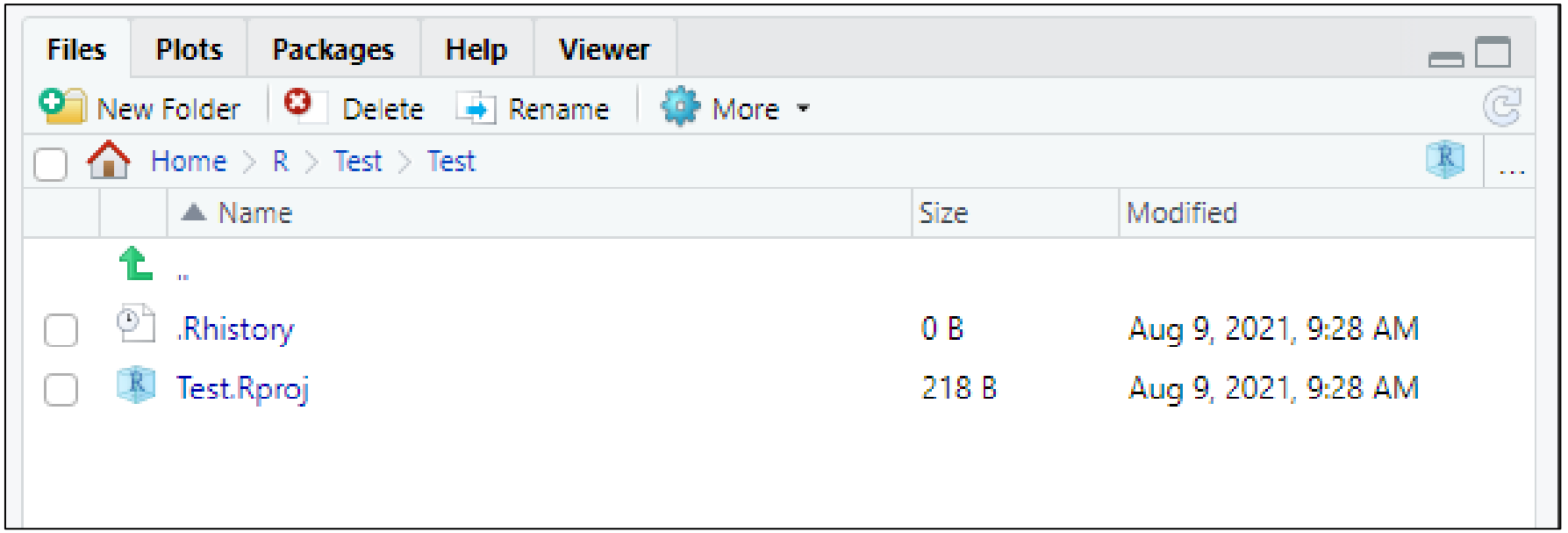


Pane layout options

RStudio: The Working Directory

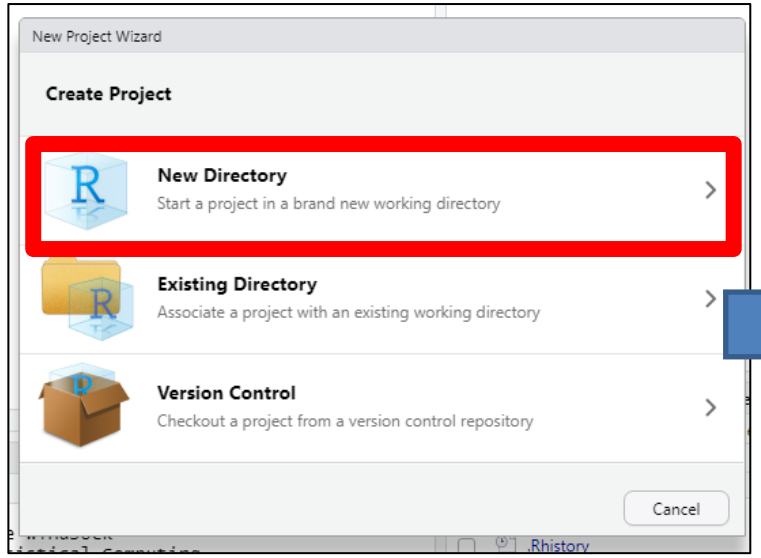
- **Working Directory:** A set of related data, analyses, and text self-contained in a single folder
- Use **Projects** feature in RStudio to create a working directory

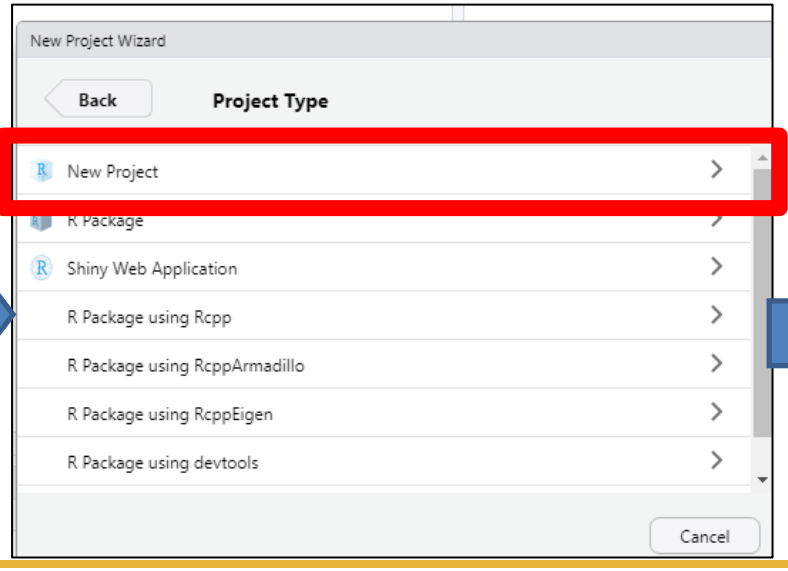
Working directory created for a Project in RStudio

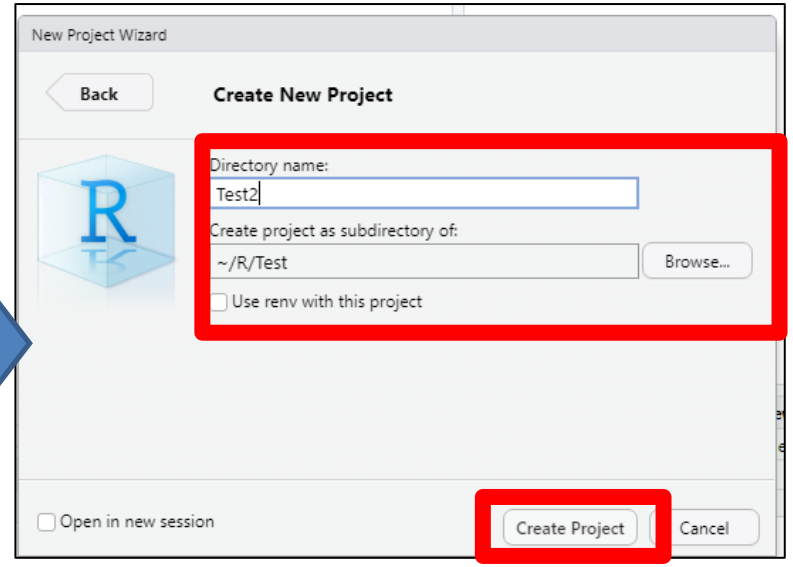


RStudio: Create a New Project

- Steps for creating a new R Project in RStudio:
  - Choose File -> New Project
  - Choose New Directory, then New Project
  - Choose name and location for new directory, and select Create Project

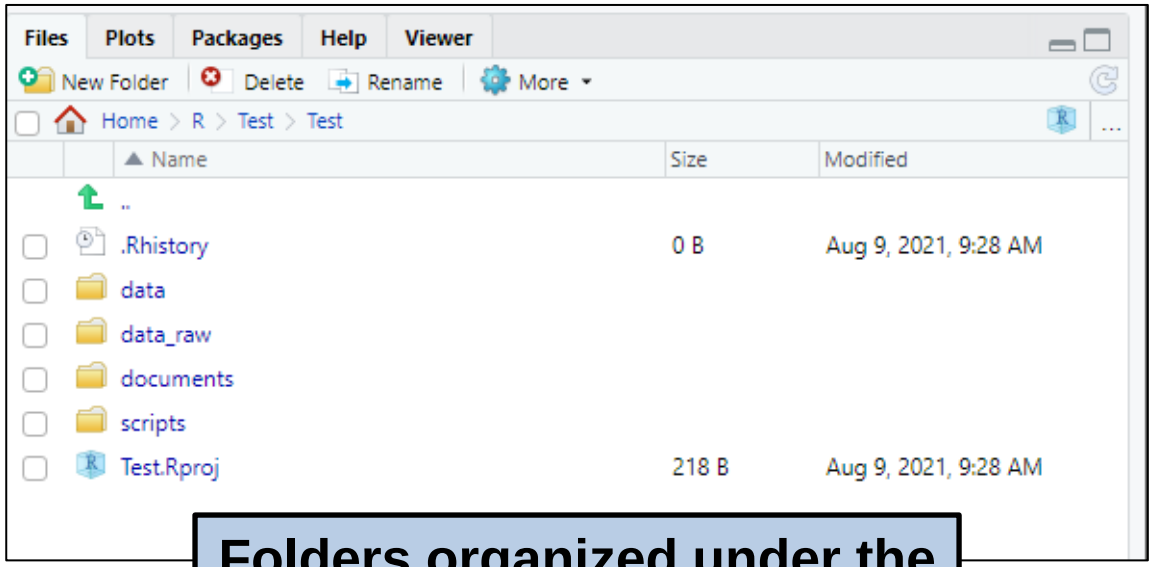






RStudio: Organize Working Directory

- **Best practice:** Create folders for scripts, data, and document. For example –
  - **data_raw/** and **data/** – Folders to store raw data and intermediate datasets
  - **documents/** – Keep outlines, drafts, and other text
  - **scripts/** – Keep your R scripts for different analyses or plotting

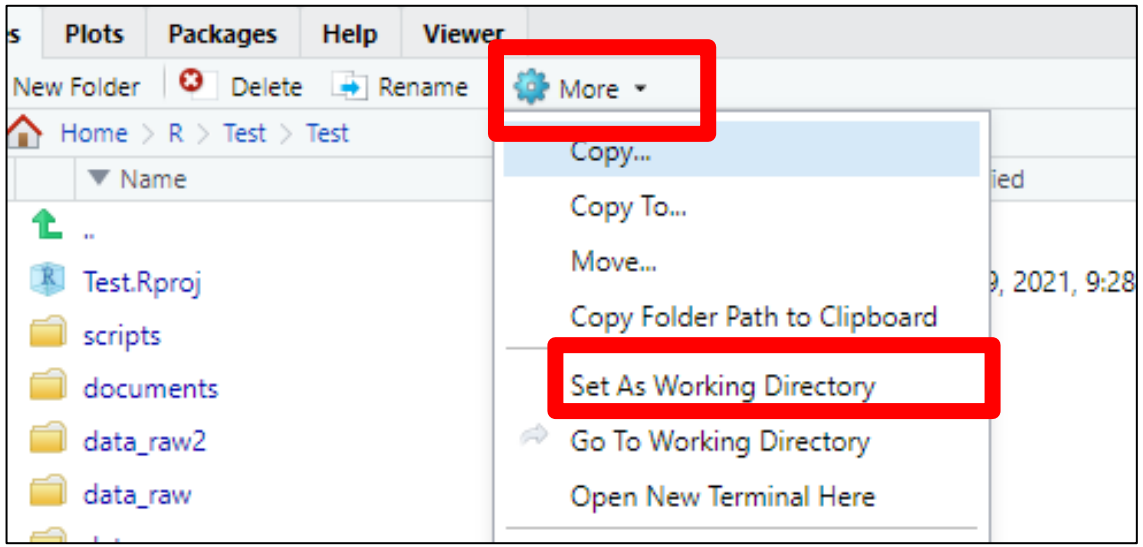


Folders organized under the working directory created for a Project in RStudio

RStudio: Check Working Directory

- To check if your working directory is set properly, use **getwd()**
- If working directory is not correct:
  - Change in RStudio interface in File browser using **More-> Set As Working Directory**
  - Alternatively, you can use **setwd("/path/to/working/directory")** to reset your working directory

Setting the working directory in RStudio



Knowledge Check 1: How can you create a working directory in RStudio?

A. Choose File -> New Project

B. Choose View -> Panes

C. Choose Session -> New Session

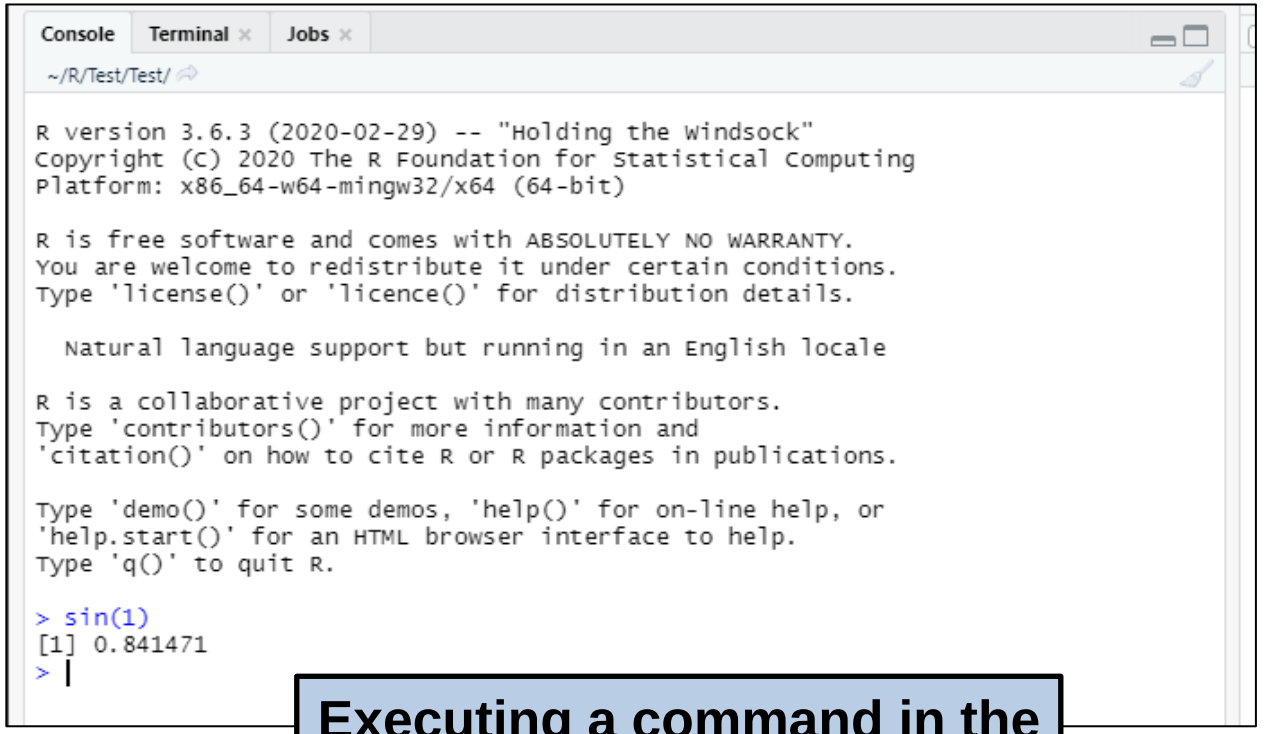
D. Choose Tools -> Install Packages

E. Choose Tools -> Global Options

Interacting with R

Console Pane

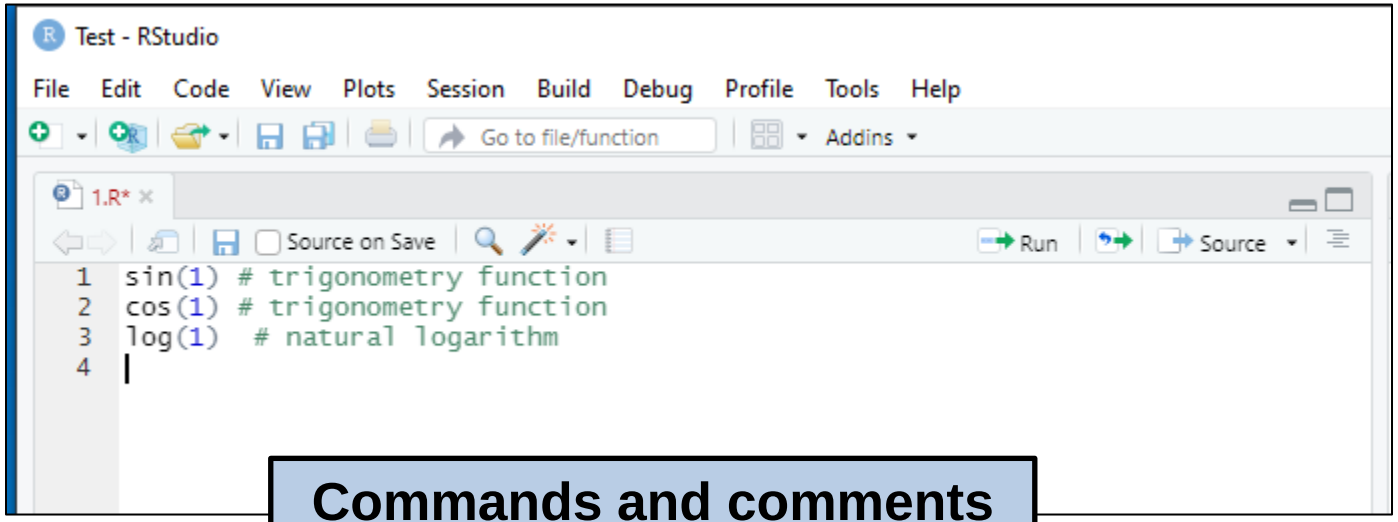
- **Console Pane:** Where commands written in the R language can be typed and executed immediately
  - Type commands directly into the console and press Enter to execute those commands
  - Commands will be forgotten when you close the session



Executing a command in the R console

Script Editor

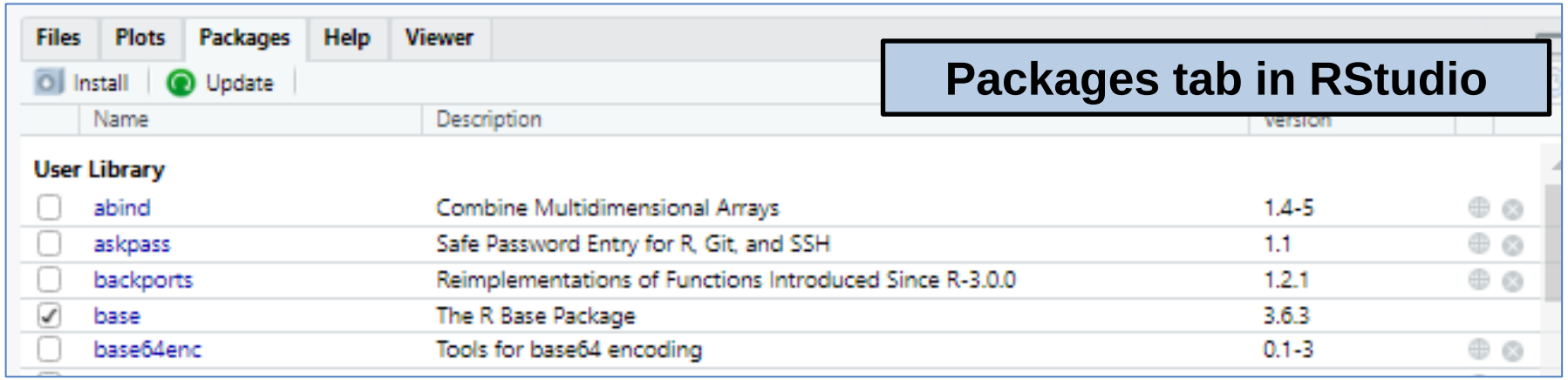
- **Script Editor**: Type commands in script editor to run and save the script
  - Makes workflow easy to replicate later



Commands and comments in script editor

R Packages

- **Use packages to add functions to R**
  - Over 21,000 packages available on [the Comprehensive R Archive Network (CRAN)]
  - Choose **Packages tab** in RStudio to view installed packages, install new packages, update packages, and make a package available for use
  - Choose the checkmark beside an installed package or enter **library(packagename)** in the console to make a package available for use



Packages tab in RStudio

Key R Terms

Creating Objects in R and Assigning Value

- R uses **object-oriented programming**
- What are known as **objects** in R are known as **variables** in many other programming languages
- Assign **values** to objects
- To create object, give it a name followed by **assignment operator <-** (shortcuts: Alt + - (Windows) or Option + - (Mac)) , and then the value

```
> weight_kg <- 55
> |
```

Assigning value 55 to the object weight_kg

Naming Objects in R (1)

- Best practices for naming objects in R:
  - Precise and short
  - Cannot start with number (e.g., 2x is not valid but x2 is)
  - R is case sensitive (e.g., weight_kg is different from Weight_kg)
  - Cannot use names from fundamental functions in R

Naming Objects in R (2)

- Best practices for naming objects in R:
  - Even if allowed, best not to use other function names (e.g., c, T, mean, data, df, weights)
  - Best to avoid dots (.) within an object name
  - Recommended to use nouns for object names and verbs for function names

Assignment

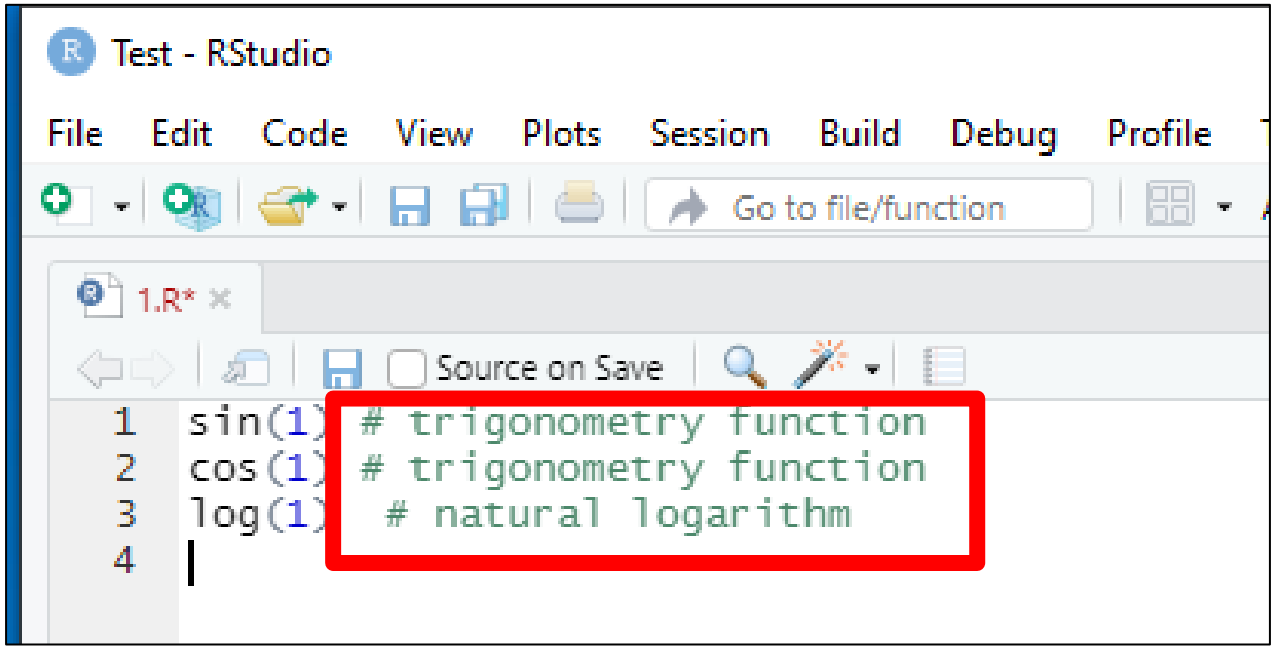
- R will display value of an object if you enter object name as command
  - Alternatively, enter assignment command and put in parentheses to display value as output
- Objects can be assigned new values
- Assigning a value to one object does not change the values of other objects

```
> weight_kg<-55
> (weight_kg<-55)
[1] 55
> weight_kg
[1] 55
> weight_kg<-57.5
> 2.2*weight_kg
[1] 126.5
> weight_lb<-2.2*weight_kg
> weight_kg<-100
```

What do you think is the current content of the object weight_lb? 126.5 or 220?

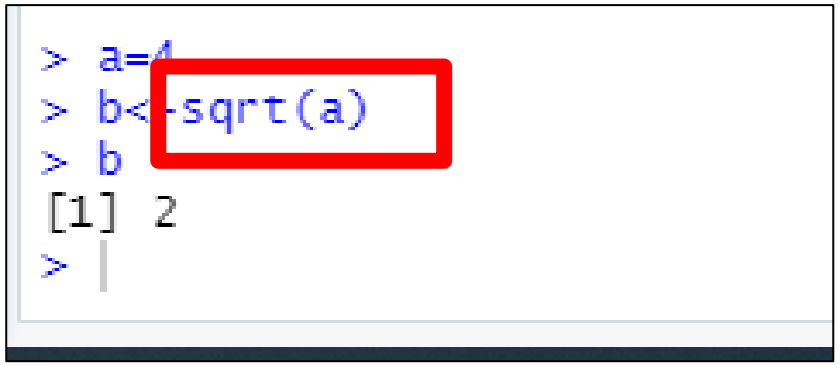
Comments

- The **comment character** in R is #
- Anything to the right of # in a script will be ignored by R
- Useful for leaving notes and explanations in your script



Functions and their Arguments (1)

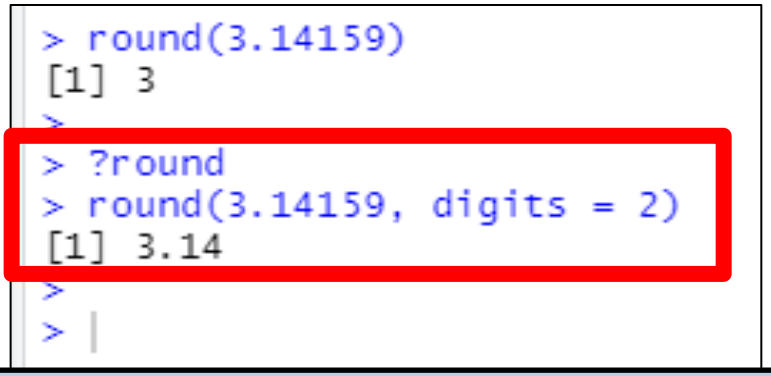
- **Functions:** "canned scripts" that automate more complicated sets of commands
  - Many predefined or made available by importing R packages
- Functions usually make one or more inputs called **arguments**
- Functions often return a **value**
- Executing a function known as **calling the function**



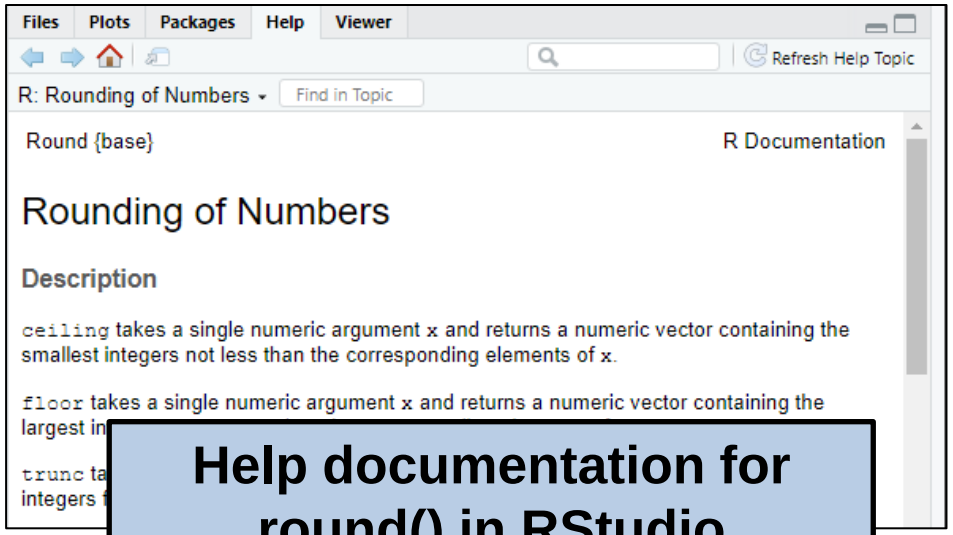
sqrt() function: argument is 4, output value is 2

Functions and their Arguments (2)

- Arguments can be anything: numbers, filenames, other objects, etc.
- Exactly what each argument means differs per function
  - Look up in documentation – Use command **?functionname**
- Some functions take arguments which may either be specified by the user or take on a default value, called **options**



Looking up help documentation for round() function, and using option to define number of digits

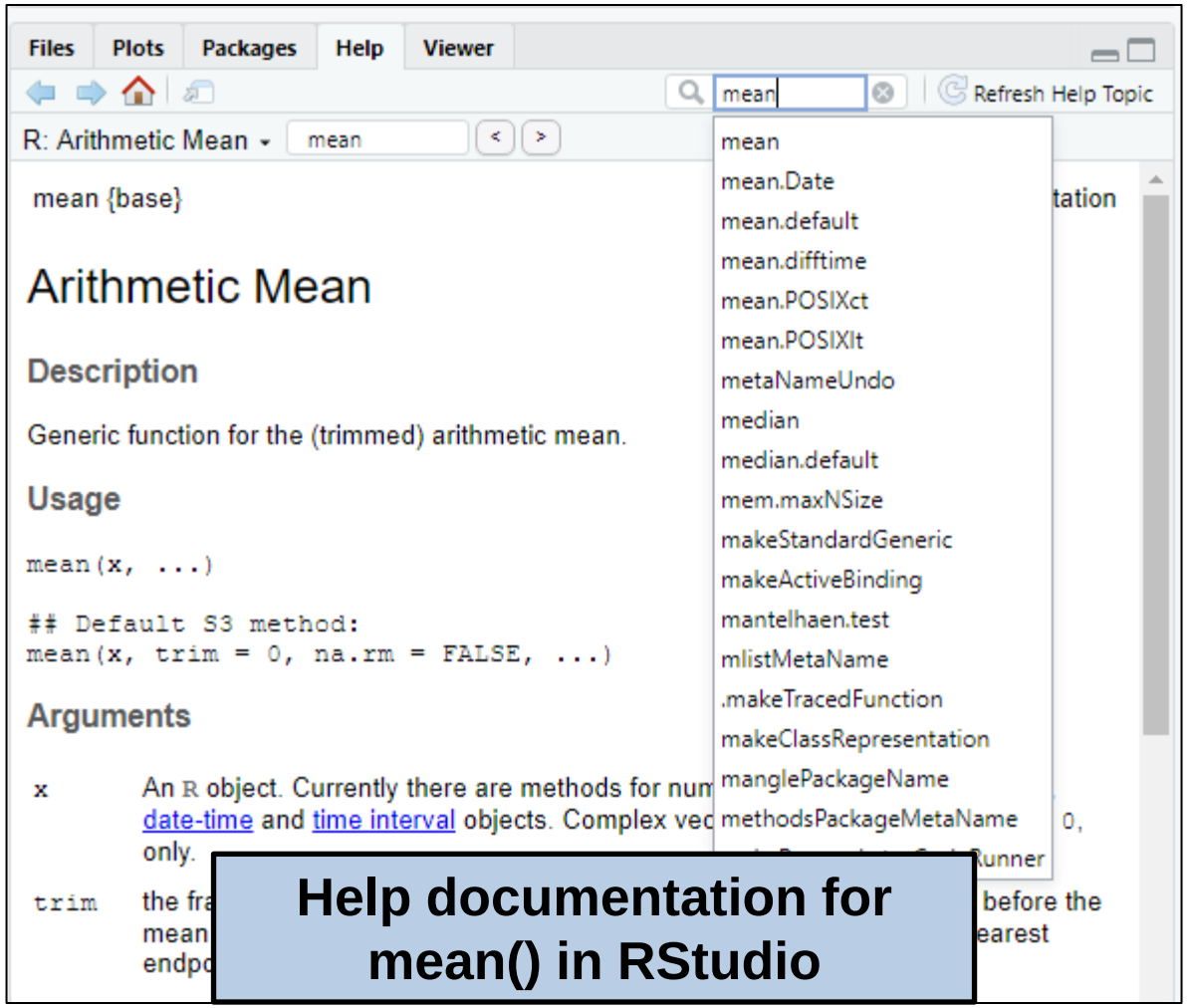


Help documentation for round() in RStudio

Getting Help

RStudio Help Interface

- **Help menu in RStudio:** Lower right pane (under Help tab)
- Search for help topics
- Alternatively, use command **?functionname** to find help information on a specific function
  - Use **??** To search available help pages



Help documentation for mean() in RStudio

Finding Help Information Online

- Try search of **RDocumentation**: Searches all R packages on CRAN and Bioconductor
- Browse **CRAN package repository**: Find reference manuals and vignettes
- **Google search**: Try “R [task]”, or try searching the error message and function or package name
- Check **Stack Overflow** and **search using the [r] tag**
- Search **Posit Community forums**

Getting Help from the R User Community (1)

- Best practices to request help from R user community (on Stack Overflow, Posit Community forums, etc.):
  - Use the correct words to describe your problem, and be precise
  - Provide raw file and script up to the point of the error, when possible
    - Save R object to file, i.e.: saveRDS(weight_kg, file="~/R/Test/Test/weight_kg.rds")
  - Always include the output of **sessionInfo()**

```
> sessionInfo()
R version 3.6.3 (2020-02-29)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 18363)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
[1] compiler_3.6.3 tools_3.6.3    yaml_2.2.1
>
```

Output from sessionInfo()

Getting Help from the R User Community: More Resources

- Posting Guide: How to ask good questions that prompt useful answers (The R Foundation)
- How to Ask for R Help (Revolutions blog)
- A blog post by Jon Skeet: Advice on how to ask programming questions
- The reprex package: Create reproducible examples when asking for help (see presentation from rOpenSci Blog)

Resources to Learn More about R

- An Introduction to R (and other manuals, FAQ, and R Journal available through cran.r-project.org)
- Finding Your Way to R (RStudio Education from Posit, as well as other learning resources for Beginners)
- R for Reproducible Scientific Analysis (from Software Carpentry)
- List of online books for learning R in NIH Library catalog
- Communities to join for learning R and data science at NIH:
  - NIH Data Science Microsoft Team
  - NIH-DATASCIENCE-L

Wrap-Up

- R is a language for statistical computing and graphics, with RStudio Integrated Development Environment from Posit
- R is great for reproducibility
- R is extensible through over 21,000+ packages
- In RStudio, type commands in the script editor to run and save the script
- Find help resources built into RStudio, through sites like RDocumentation, or ask for help through various online forums

Knowledge Check 2: What help search options are available in RStudio interface?

A. Use command **?functionname** to find information on a function
B. Use **??** To search available help pages
C. Use search form under Help tab
D. Type out your question in the console
E. Use command **sessionInfo()**

Anaconda Now Available at NIH

- NIH employees and contractors now have access to **Anaconda Business** licenses through the NIH Center for Information Technology
- Getting a license is simple, and there's no associated cost
- Just fill out this form, and you'll receive a license within a few business days
- With its secure Python and R distributions for data science, this tool is a great fit for anyone engaged in scientific computing, artificial intelligence, or machine learning
- To learn more and review current restrictions on use of Anaconda Cloud, see this NIH product listing
- Please reach out to Anaconda@nih.gov with any questions

Have a Question?

- Joelle Mornini, biomedical librarian
- Office: 301-451-9333
- Email: joelle.mornini@nih.gov

Questions & Comments

National Institutes of Health
Office of Management