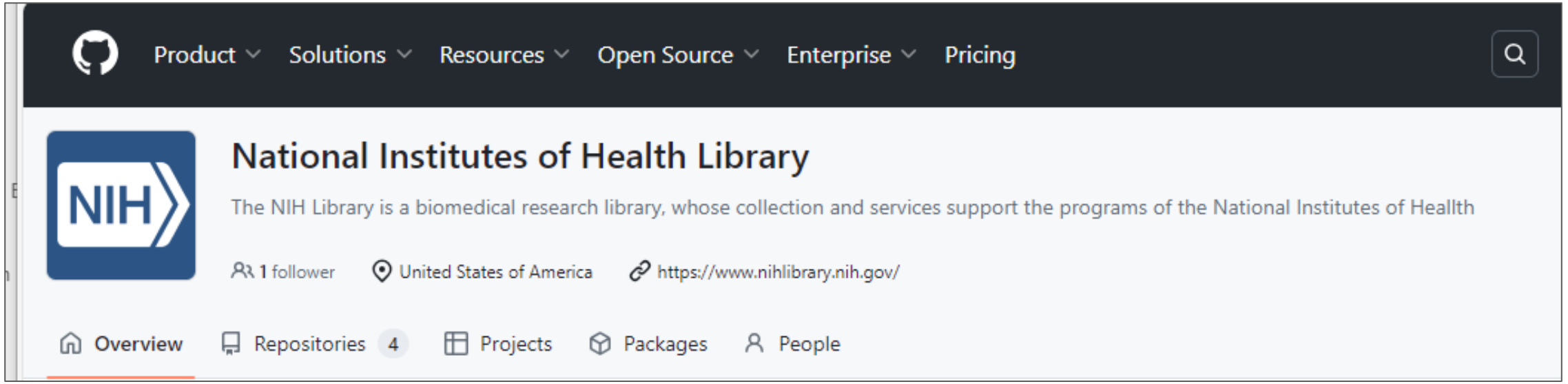
Introduction to R and RStudio

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 and learn how it can support reproducible research
  - Define key terms as they relate to R: object, assign
  - 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
- 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
- R (recommended 4.4.0 or later) needs to be installed before RStudio

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

How Does R Compare? (1)

- **R vs. Python**
  - R excels in statistical analysis and publication-quality graphics
  - Python offers broader general-purpose programming capabilities
  - R has more specialized statistical packages; Python has a stronger machine learning ecosystem
  - Both support reproducible research with notebooks (R Markdown vs. Jupyter)
- **R vs. SAS**
  - R is free and open-source; SAS requires expensive licensing
  - SAS offers enterprise-level support; R relies on community support
  - R's package ecosystem evolves faster than SAS
  - R code is more accessible for sharing and collaboration

How Does R Compare? (2)

- **R vs. SPSS**:
  - R requires more programming knowledge; SPSS has a point-and-click interface
  - R offers greater flexibility for custom analyses
  - R provides superior data visualization capabilities
  - R enables more transparent and reproducible workflows
- **R vs. Excel**:
  - R handles larger datasets more efficiently
  - R provides automated, reproducible analysis workflows
  - Excel is more accessible for simple analyses
  - R offers more statistical rigor and advanced techniques

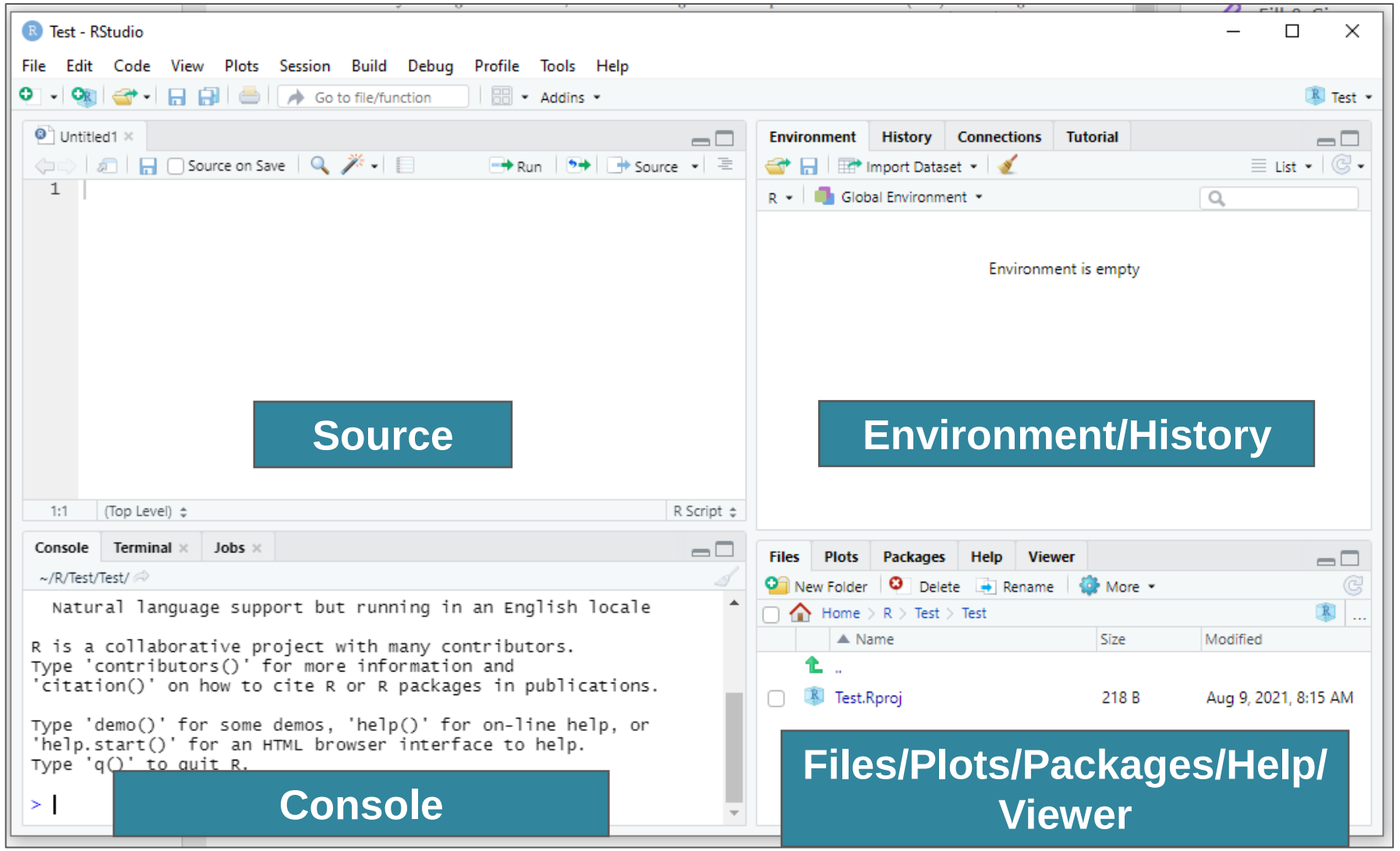
When R Shines

- Complex statistical analyses requiring specialized techniques
- Creating publication-quality visualizations
- Developing reproducible research workflows
- Working in domains with dedicated R packages (bioinformatics, ecology, etc.)
- Situations requiring free, open-source solutions

RStudio Overview

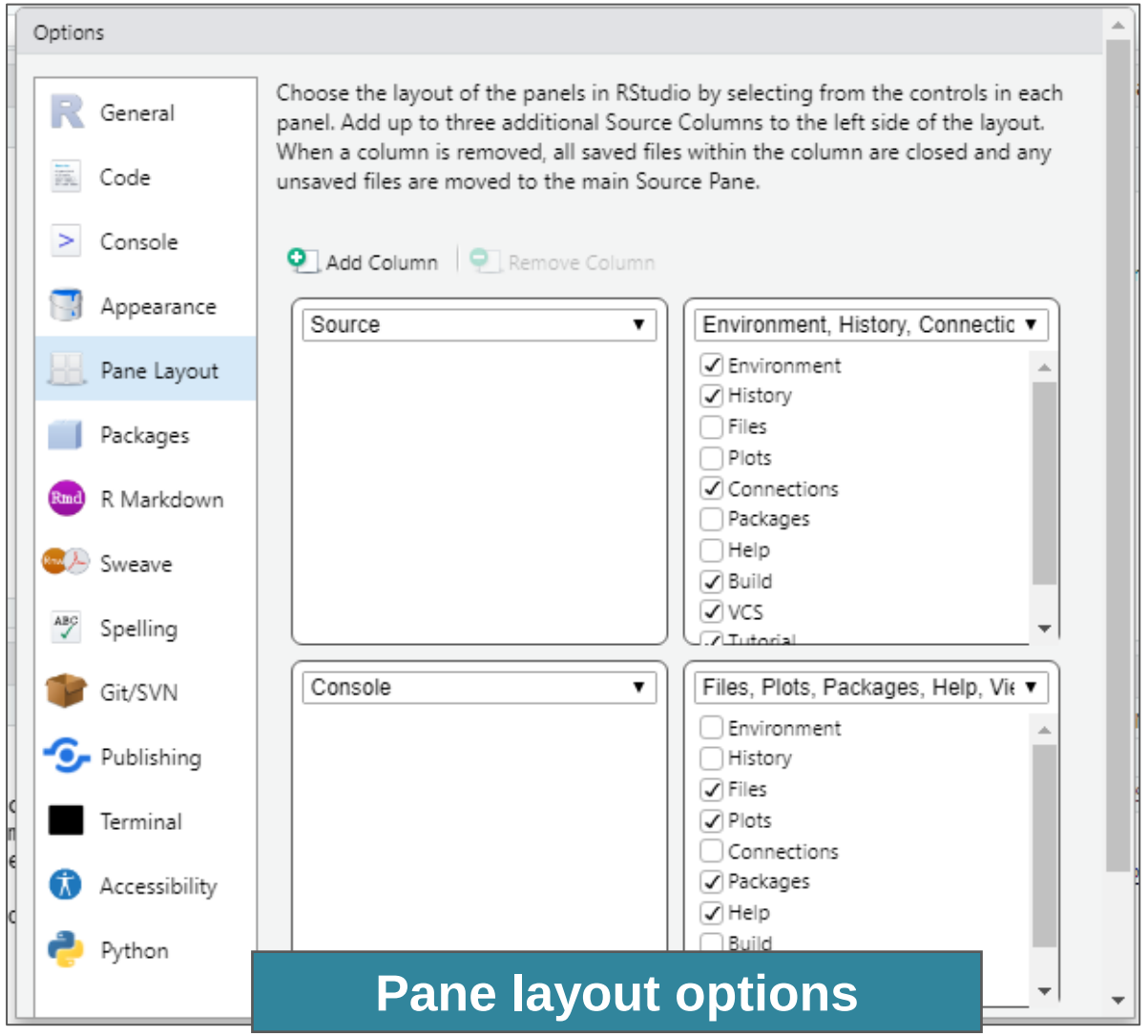
4 Panes of RStudio

- Reference document that includes keyboard shortcuts (PDF): RStudio IDE Cheat sheet
- Learn about additional RStudio keyboard shortcuts (PDF)



4 Panes of RStudio: Customizing Pane Layout

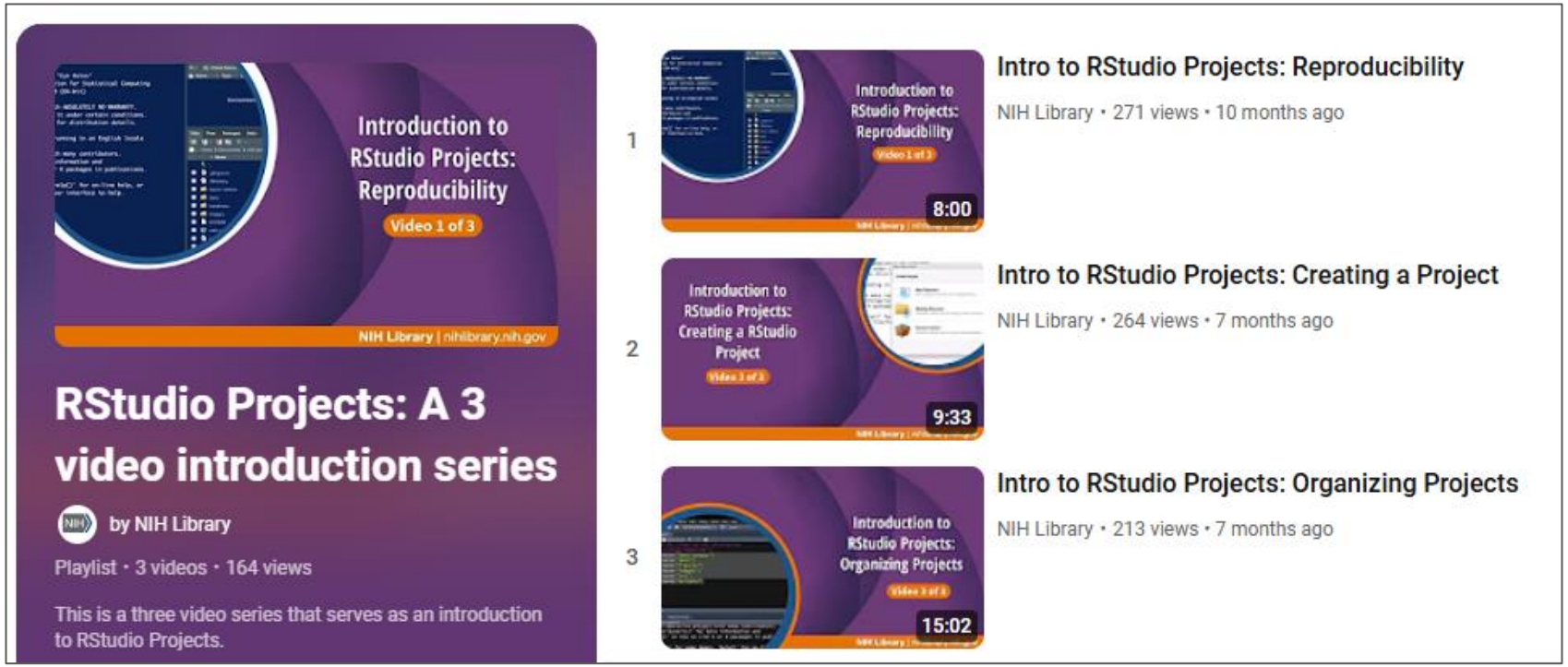
- The placement of these panes and their content can be customized
  - See menu Tools -> Global Options -> Pane Layout



Pane layout options

RStudio Projects

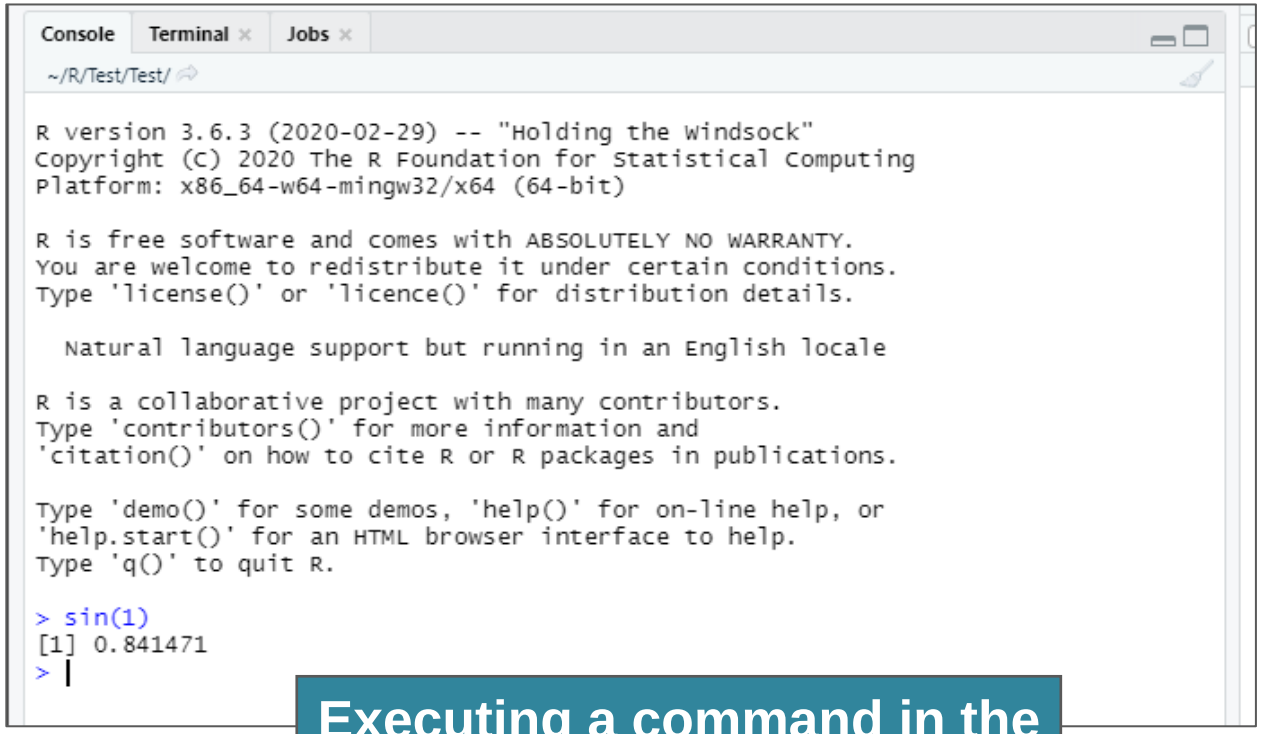
- Watch the on-demand trainings RStudio Projects: A 3 video introduction series to learn how to create and organize projects in RStudio



Interacting with R

Console Pane

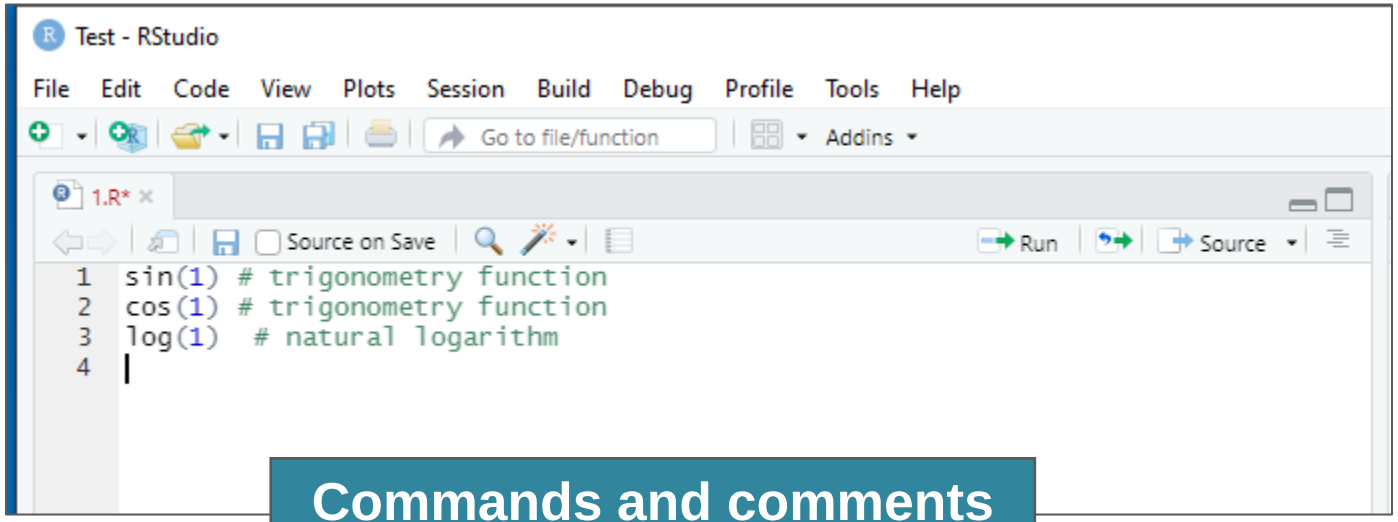
- **Console Pane:** Where commands written in the R language can be typed and executed immediately
  - Type commands directly into the console and press Enter to execute those commands
  - Commands will be forgotten when you close the session



Executing a command in the R console

Script Editor

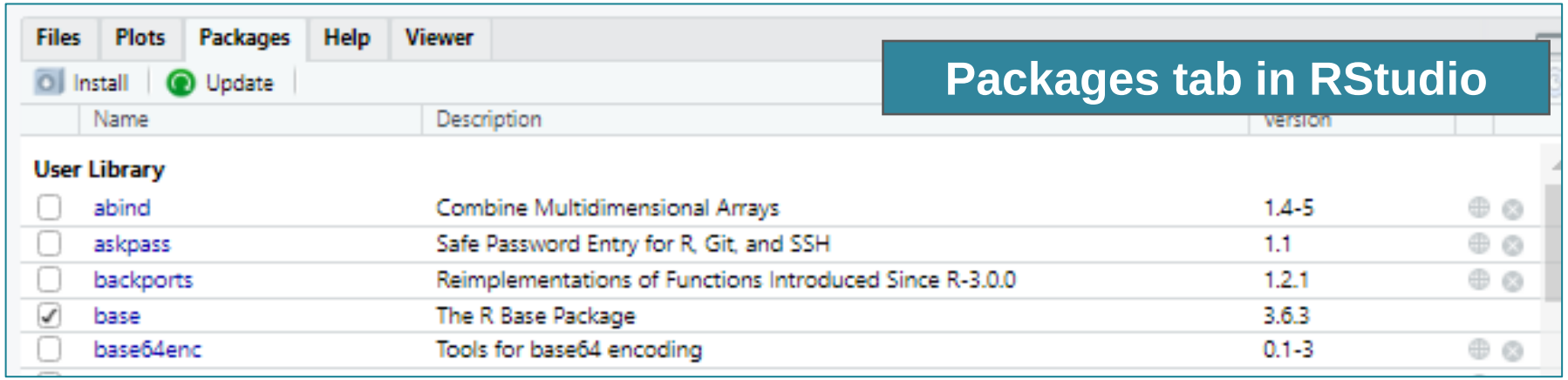
- **Script Editor**: Type commands in script editor to run and save the script
  - Makes code easy to replicate later



Commands and comments in script editor

R Packages

- **Use packages to add functions to R**
  - Over 22,000 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

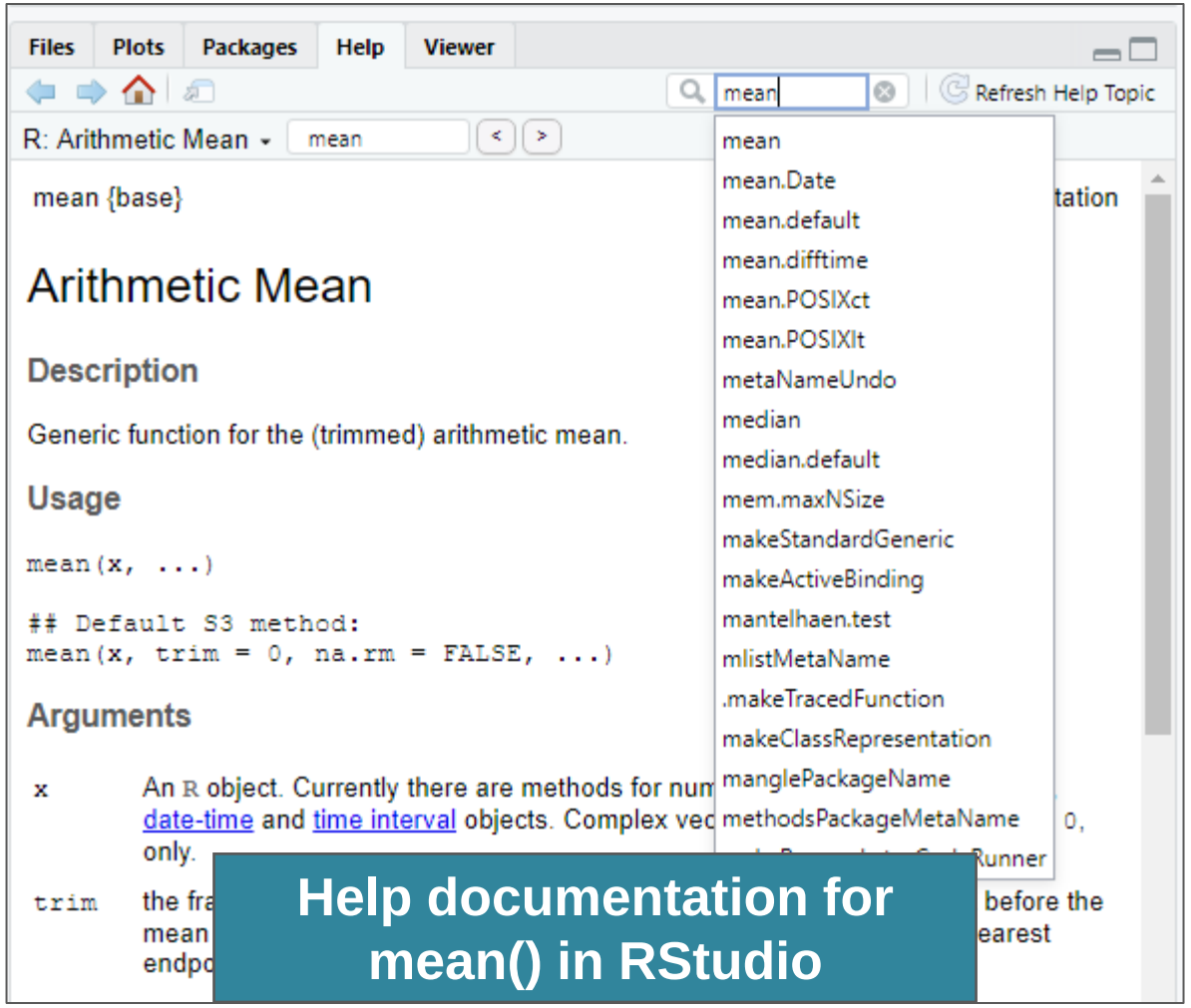
```
> weight_kg <- 55
> 
```

Assigning value 55 to the object weight_kg

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

- 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="~/Demo/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

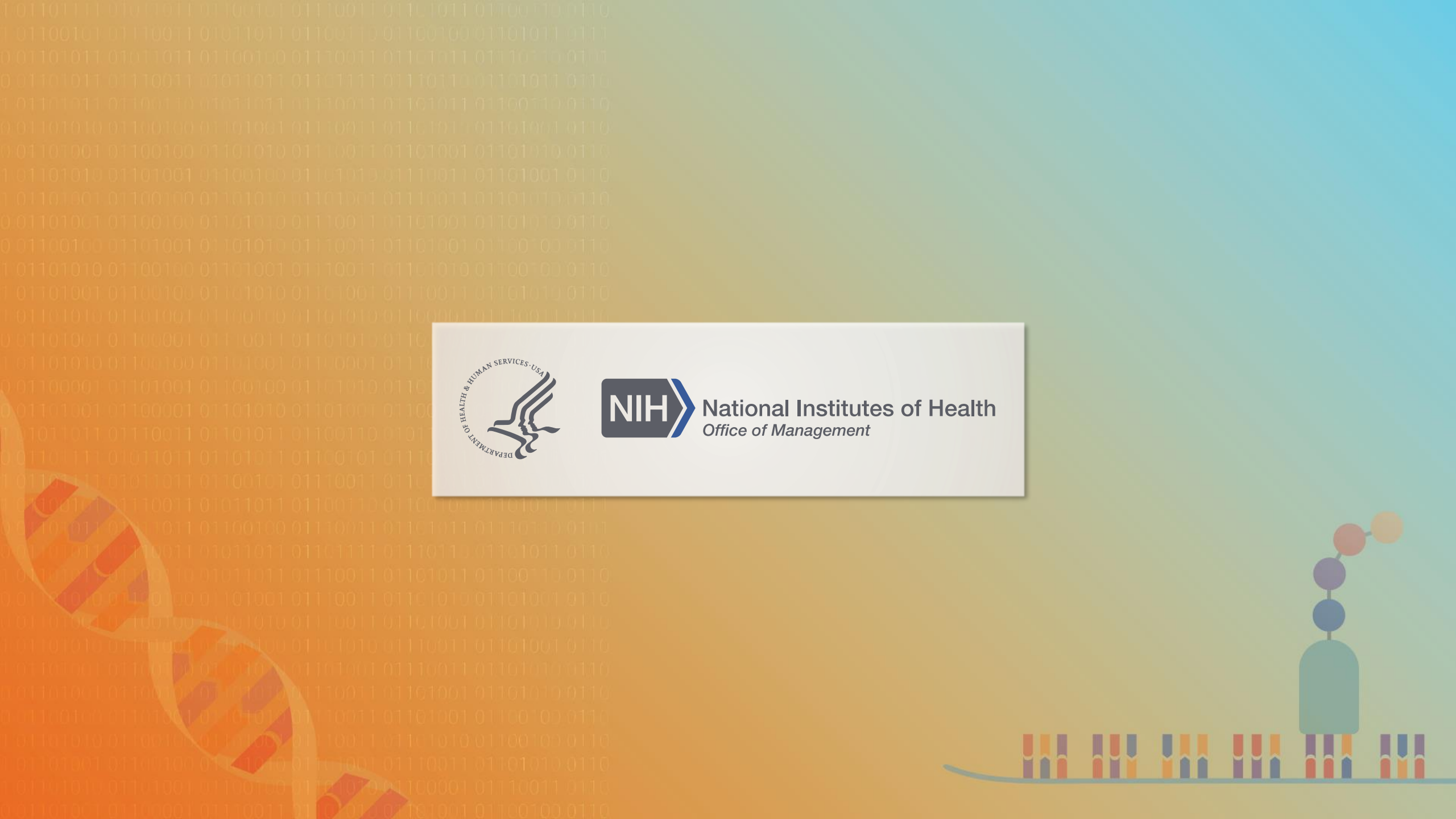
- Posting Guide: How to ask good questions that prompt useful answers (The R Foundation)
- How to Ask for R Help (Revolutions blog)
- A blog post by Jon Skeet: Advice on how to ask programming questions
- The reprex package: Create reproducible examples when asking for help (see presentation from rOpenSci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

Resources to Learn More about R

- An Introduction to R (PDF) (and other manuals, FAQ, and R Journal available through cran.r-project.org)
- Finding Your Way to R (RStudio Education from Posit)
- 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 22,000+ 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


DEPARTMENT OF HEALTH & HUMAN SERVICES · USA
NIH
National Institutes of Health
Office of Management