

May 17th 2022

Welcome to Data Analysis with R for Educators

We will begin shortly.

If you haven't done so, please
create an Rstudio Cloud account at
<https://rstudio.cloud/plans/free>

The URL is linked in the Q&A section,
available on the Zoom toolbar.

XSEDE

Extreme Science and Engineering
Discovery Environment



Supported by OAC 15-48562.

Code of Conduct

XSEDE has an external code of conduct which represents our commitment to providing an inclusive and harassment-free environment in all interactions regardless of race, age, ethnicity, national origin, language, gender, gender identity, sexual orientation, disability, physical appearance, political views, military service, health status, or religion. The code of conduct extends to all XSEDE-sponsored events, services, and interactions.

Code of Conduct: <https://www.xsede.org/codeofconduct>

Contact:

- Event organizer: Susan Mehringer (shm7@cornell.edu)
- XSEDE ombudspersons:
 - Linda Akli, Southeastern Universities Research Association (akli@sura.org)
 - Lizanne Destefano, Georgia Tech (lizanne.destefano@ceismc.gatech.edu)
 - Ken Hackworth, Pittsburgh Supercomputing Center (hackworth@psc.edu)
 - Bryan Snead, Texas Advanced Computing Center (jbsnead@tacc.utexas.edu)
- Anonymous reporting form available at <https://www.xsede.org/codeofconduct>



XSEDE

Words Matter!

In line with XSEDE's Code of Conduct, XSEDE is committed to providing training events that foster inclusion and show respect for all. This commitment applies not only to how we interact during the event; it also applies to the training materials and presentation. It is not XSEDE's position to use, condone, or promote offensive terminology. XSEDE has posted a [Terminology Statement](#).

XSEDE instructors strive to keep inclusive language at the forefront. In the event that we have included inappropriate materials, verbal or written, please let us know at terminology@xsede.org

While XSEDE has no control over external third-party documentation, we are taking steps to effect change by contacting the relevant organizations; we hope this will be addressed by all third parties soon.

If you see any terminology concerns in the following presentation or slides, we want to know!

Please contact the Terminology Task Force: terminology@xsede.org



XSEDE

May 17th 2022

Data Analysis with R for Educators

Christopher Cameron, Ph.D.

*Computational Scientist
Cornell Center for Advanced Computing*

cjc73@cornell.edu

Wilbur Ouma, Ph.D.

*Client Engineer
Ohio Supercomputer Center*

wouma@osc.edu

XSEDE

Extreme Science and Engineering
Discovery Environment



Supported by OAC 15-48562.

Polls

1. Are you interested in R to support teaching, research, or both?
2. Have you used R before?
3. Have you used Python before?
4. What program(s) or environments(s) do you use for statistical analysis?

Zoom poll will pop over the screen



XSEDE

Workshop Questions

1. What is R and how does it fit in the data science ecosystem?
2. When is R the right choice for data analysis?
3. What features make R useful for educators?
4. Where can I get more information?

R takes time to learn, and this is the first step. The materials and demonstrations today will help you decide if R is worth the investment.



XSEDE

Agenda

Orientation to the R ecosystem (20 mins)

- Why R?
- Motivation for R
- Brief history of R
- What is R?
- Community
- Documentation and help

Introduction to R software

- R Console
- RStudio
- “10 minutes to R”

Data Analysis Examples

- R Essentials via R Notebook
- Grammar of graphics and data
- Linear Regression
- Multiple Regression



R is...

Good for:

- tabular data
(or vectors or lists)
- statistical analysis
- data visualization
- Integrating custom code in C/C++, Fortran and Java.

Less suitable for

- unstructured data
- file system scripting
- data scraping, cleaning and formatting

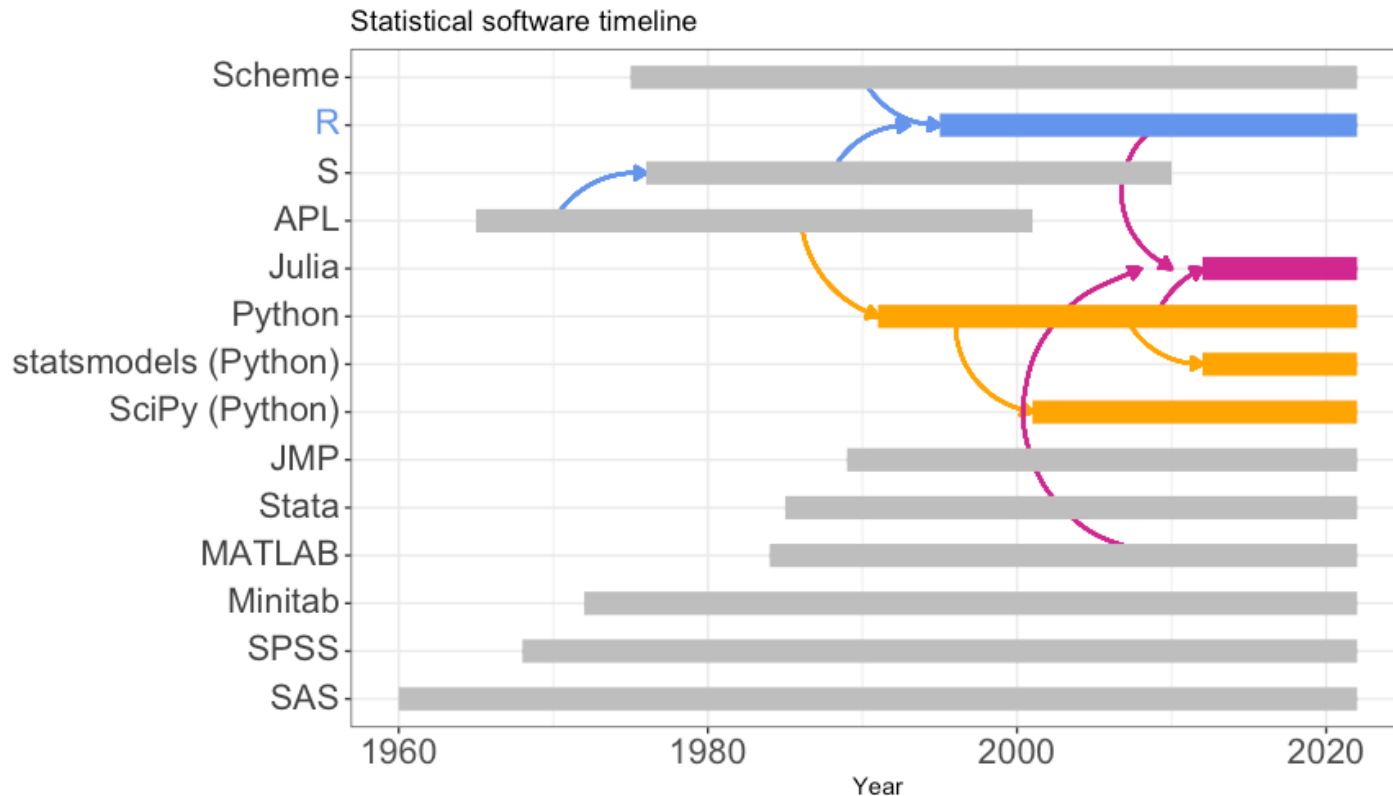
Some people want R to do everything, so packages do exist to make some of these possible!

(Someone also wrote a web-crawler in SAS)



XSEDE

Statistics Software Ecosystem



R is a relative newcomer (as is Python),
but builds on a long legacy (APL, S, Scheme).

R, Python, and Julia

- Trio of modern open-source computer languages favored by data scientists.
 - Jupyter Lab stands for the **J**ulia, **P**ython, and **R** languages
- R and Python have significant overlap and similarity, but
 - Python is more general
 - Python tends to be favored for deep learning
 - R and Python are both popular in machine learning
 - R tends to be favored for statistical analysis
 - Both have huge communities and many add-on packages
- Julia is general purpose language designed at MIT with numerical computing in mind.
 - Only recently reached version 1.0
 - Designed to be more performant but it is still developing
 - Small ecosystem compared to R and Python
 - Keep an eye on it!



Motivation for R

What if we combine things we like into a statistical computing environment and make it free and open source so others could do the same?

- Two faculty members at the University of Auckland wanted a “better software environment [for] their teaching laboratory” (1990s)
 - **did not like** the commercial offerings available
 - **did like** the S statistical programming language
 - **wished** S had some of the modern language features introduced in the Lisp variant called Scheme
- R started as an S implementation with some Scheme features and was distributed via an email list
- A colleague persuaded the authors to open-source R (1995)

Ihaka, Ross. (1998) R : Past and Future History, *A Draft of a Paper for Interface '98*. <https://cran.r-project.org/doc/html/interface98-paper/paper.html>



XSEDE

Collective, eclectic development

- R's developers borrow code conventions and programming styles freely.
 - “object oriented” `object.member` naming is common but has no special meaning in R
 - Many conventions mixed together: InitialCaps, camelCase, snake_case, vars.with.dots (again, R does not assign special meaning)
 - Packages tend to work well with expected input and unpredictably with incorrect input.
 - Many ways to accomplish any given task, inspired by different paradigms.
- Focus on practical, productive use
 - automatic and silent type conversion (casting)
 - convenience features can become gotchas (global namespace, attach)
 - variables names can mask functions
 - packages can mask each other's functions



Community

- R is used and supported by a community of largely academic researchers and developers (and more recently, data scientists).
- R gains new features via *packages* developed by the community
 - Over 10,000 add-on libraries!
 - R packages can target highly specialized research areas.
 - R packages are used to implement and share cutting edge statistical methodology.
 - The official package collection is at <https://cran.r-project.org>
 - Other collections exist: <http://www.bioconductor.org>.
 - Can load packages directly from github
- Active community generating tutorials and demos:
 - <https://www.r-bloggers.com>
 - <https://education.rstudio.com/learn/>



Documentation

R has built-in help and documentation

A typical help entry includes

- *Descriptions* of each function and their arguments.
- *Examples* showing how the functions might be used.
- *References* to relevant manuals and academic papers.

Documentation for packages usually also includes:

- One or more *vignettes* demonstrating how the package can be used to perform an analysis.
- Bundled *data sets* that support the vignette and demonstrate required data formats.



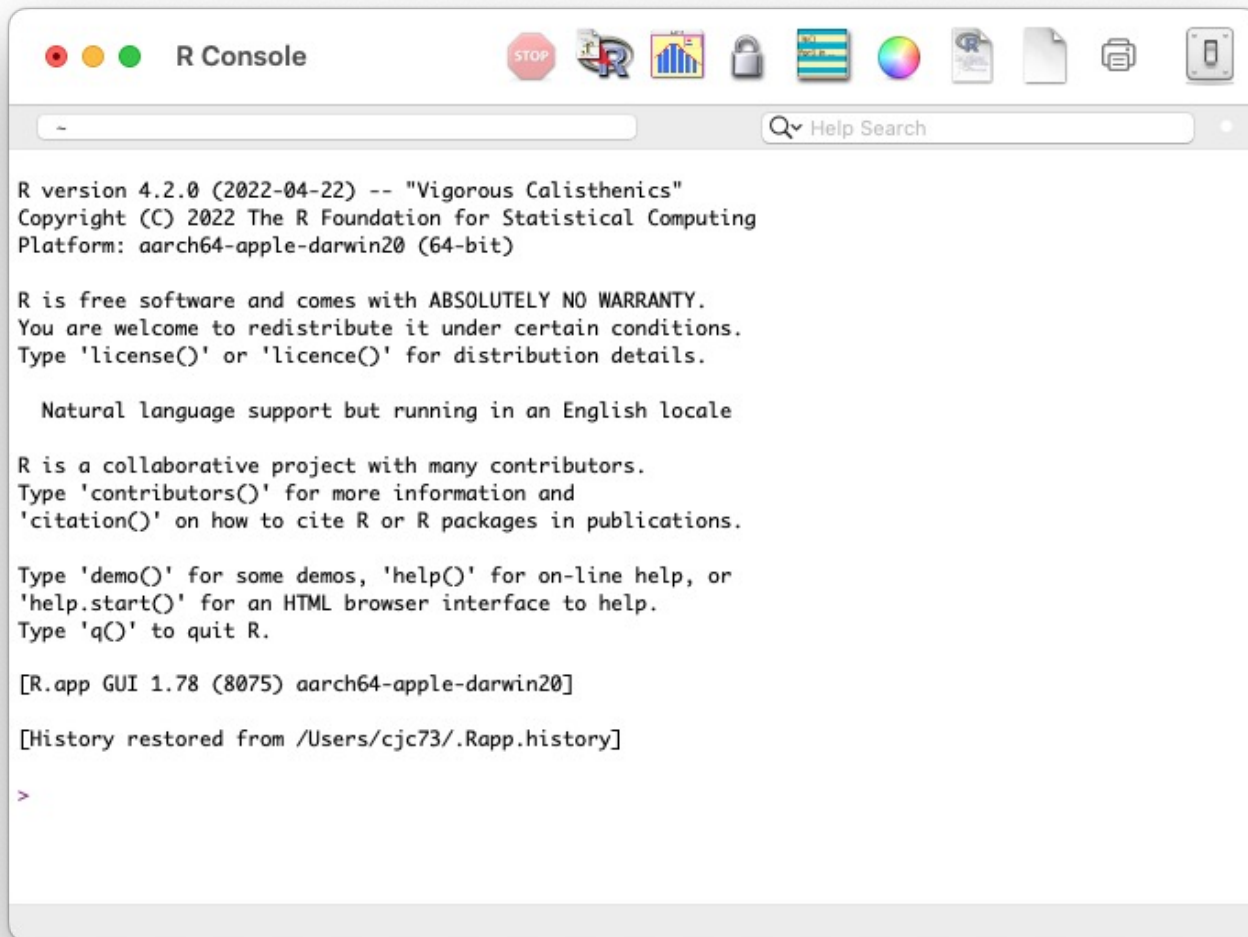
XSEDE

Base R

- The *R Project for Statistical Computing* is maintained by The R Foundation.
 - free and runs on Linux, Windows and MacOS.
 - <https://www.r-project.org>
- Command line interface via R console
 - Creates objects in memory rather than printing to screen
 - You query and manipulate these in-memory objects
 - Interactive, but not in the point-and-click GUI sense.
- Many people that “use R” do not use it directly. Instead, they use something that interfaces with the R environment.
 - RStudio IDE
 - Jupyter Lab notebooks
 - Google CoLab



R Console



The screenshot shows the R Console window with a title bar that includes the text "R Console" and several system icons. The main content area displays the R version 4.2.0 startup screen, which includes the version number, copyright information, platform details, and a list of commands for users to explore the software's features.

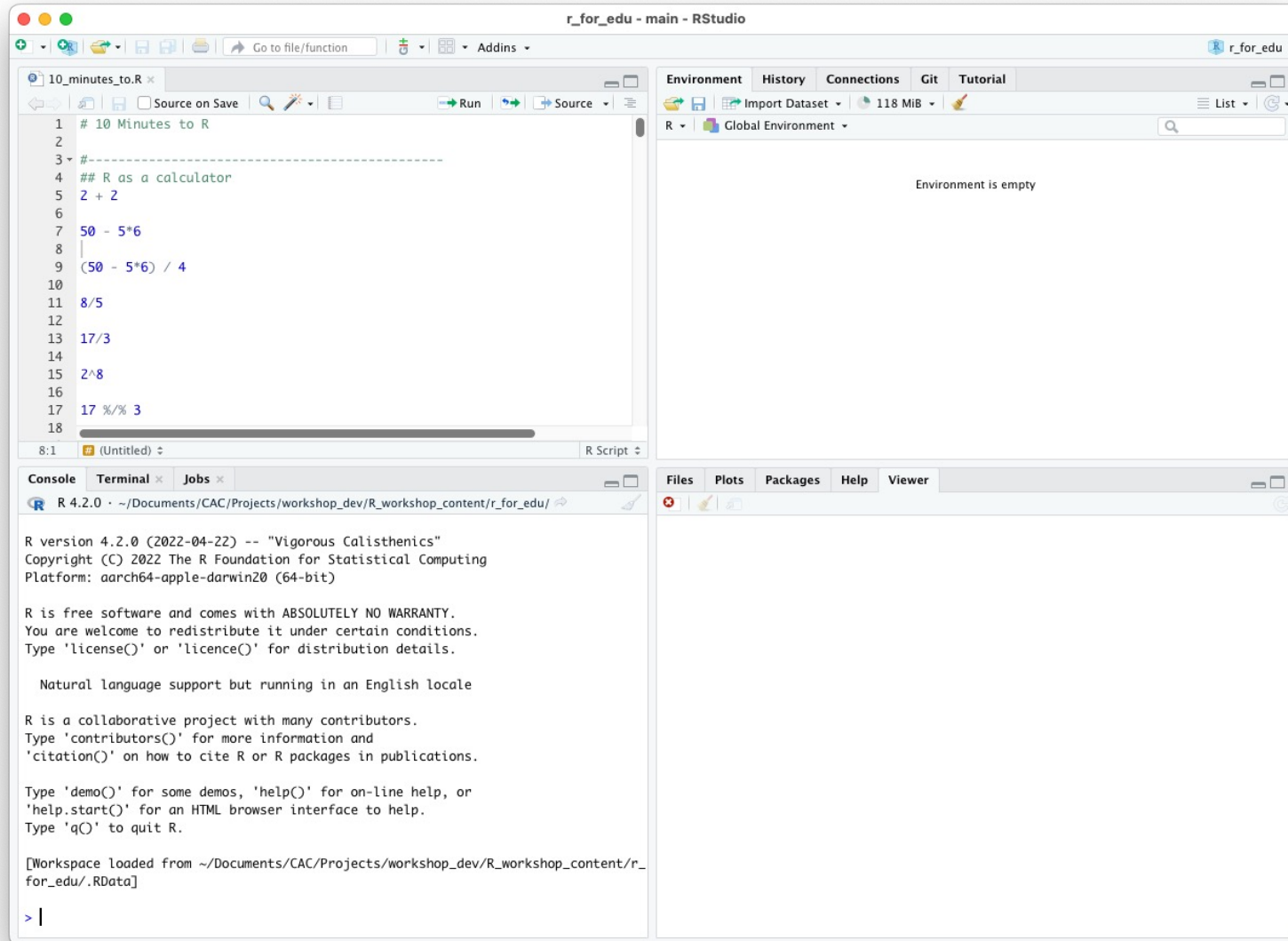
```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: aarch64-apple-darwin20 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
[R.app GUI 1.78 (8075) aarch64-apple-darwin20]  
  
[History restored from /Users/cjc73/.Rapp.history]  
  
>
```

RStudio

- RStudio is an integrated development environment for R
 - developed by RStudio Public Benefit Corporation
 - depends on installed R version
 - adds useful development, analysis and authoring features
- RStudio interface incorporates the R Console
- Tip: If you want to install RStudio locally, install R and *then* install RStudio
- RStudio Cloud <https://rstudio.cloud> is a hosted version of RStudio with the same interface as the desktop application.
 - **We will use RStudio Cloud today, so no installation is needed.**



RStudio Interface



XSEDE

10 Minutes to R

1. Open this rstudio workspace link (will post in chat):

<https://rstudio.cloud/project/4044219>

2. Login (signup if you haven't yet)

3. Project will start up (this takes a few moments)

4. Let us know you are ready by raising your hand in Zoom



While you wait...

1. Locate the tab labeled Console in left pane (unless you have changed the layout)

2. The `>` symbol near the bottom of the console window is the *prompt*.

- Click to the right of the prompt, type `2 + 2`, then press return

```
> 2 + 2  
## [1] 4
```



XSEDE