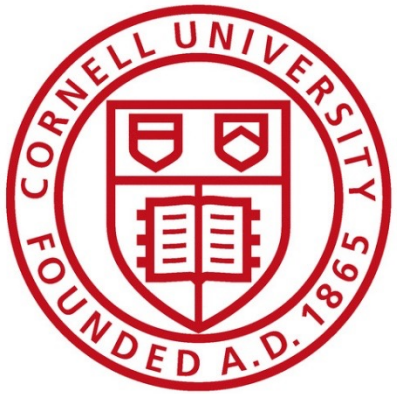
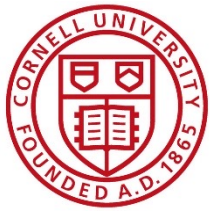


Center for Advanced Computing

www.cac.cornell.edu



Center for Advanced Computing

Introduction to modern R data analysis

Christopher Cameron
Computational Scientist
Cornell University Center for Advanced Computing
https://cac.cornell.edu/Cameron/
cjc73@cornell.edu

Questions

1. What is R and how does it fit in the statistical analysis and data science ecosystems?
2. When is R a good choice for data analysis?
3. What features make R useful for researchers?
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.

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

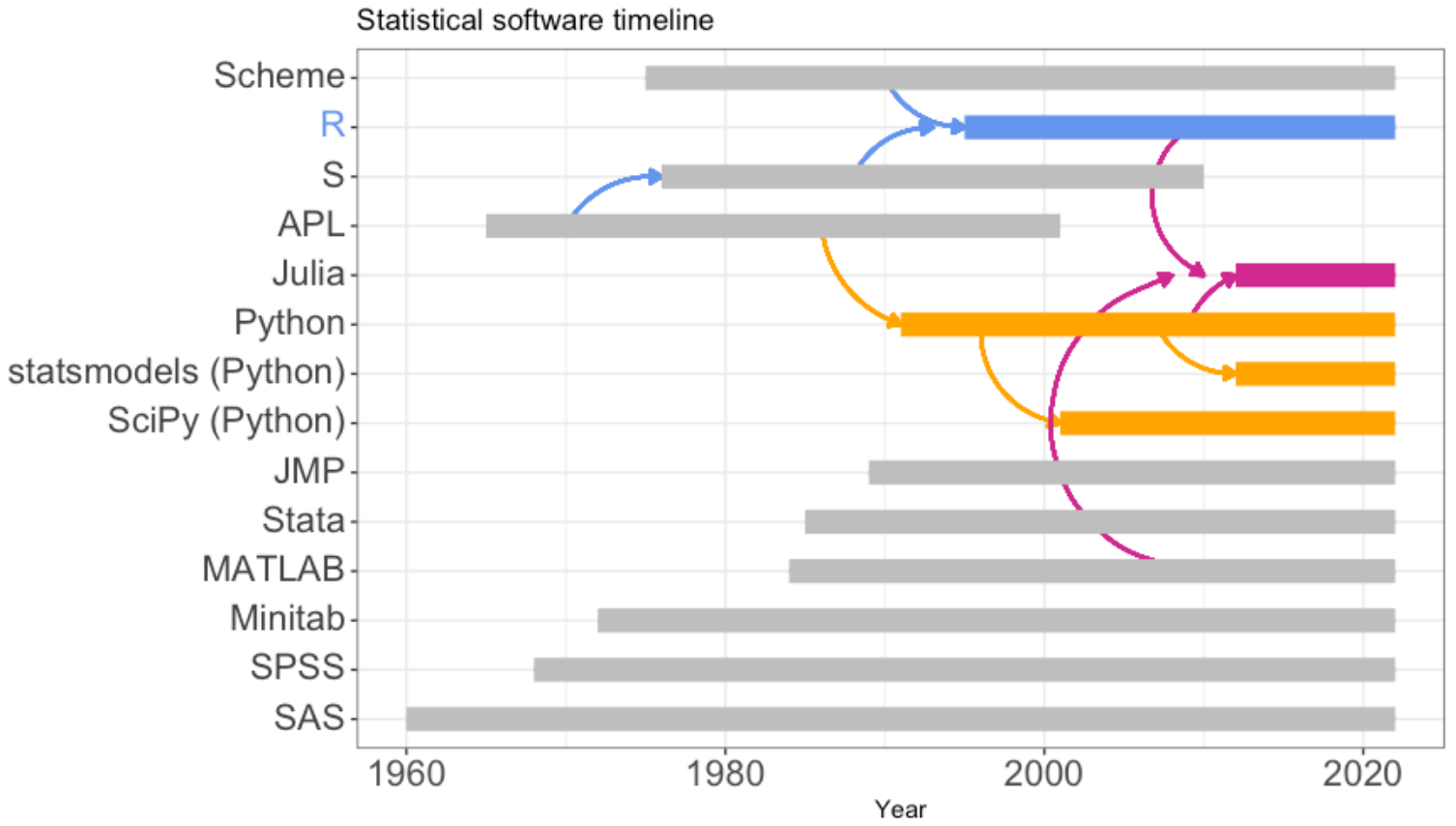
Some people want R to do everything, so packages do exist to make some of these possible!

(Someone also wrote a web-crawler in SAS)

R, Python, and Julia

- Trio of modern open-source computer languages favored by data scientists.
  - Jupyter Lab stands for the **Ju**lia, **Pyt**hon, 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 (but can use R and Python)
  - Keep an eye on it!

Statistics Software Ecosystem



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

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

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: InitalCaps, 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)
    - packages can mask each other's functions
    - variable names can have the same name as functions – mostly works, hard to read

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/
  - https://cvw.cac.cornell.edu/R/
  - https://community.rstudio.com ← community help forum

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.

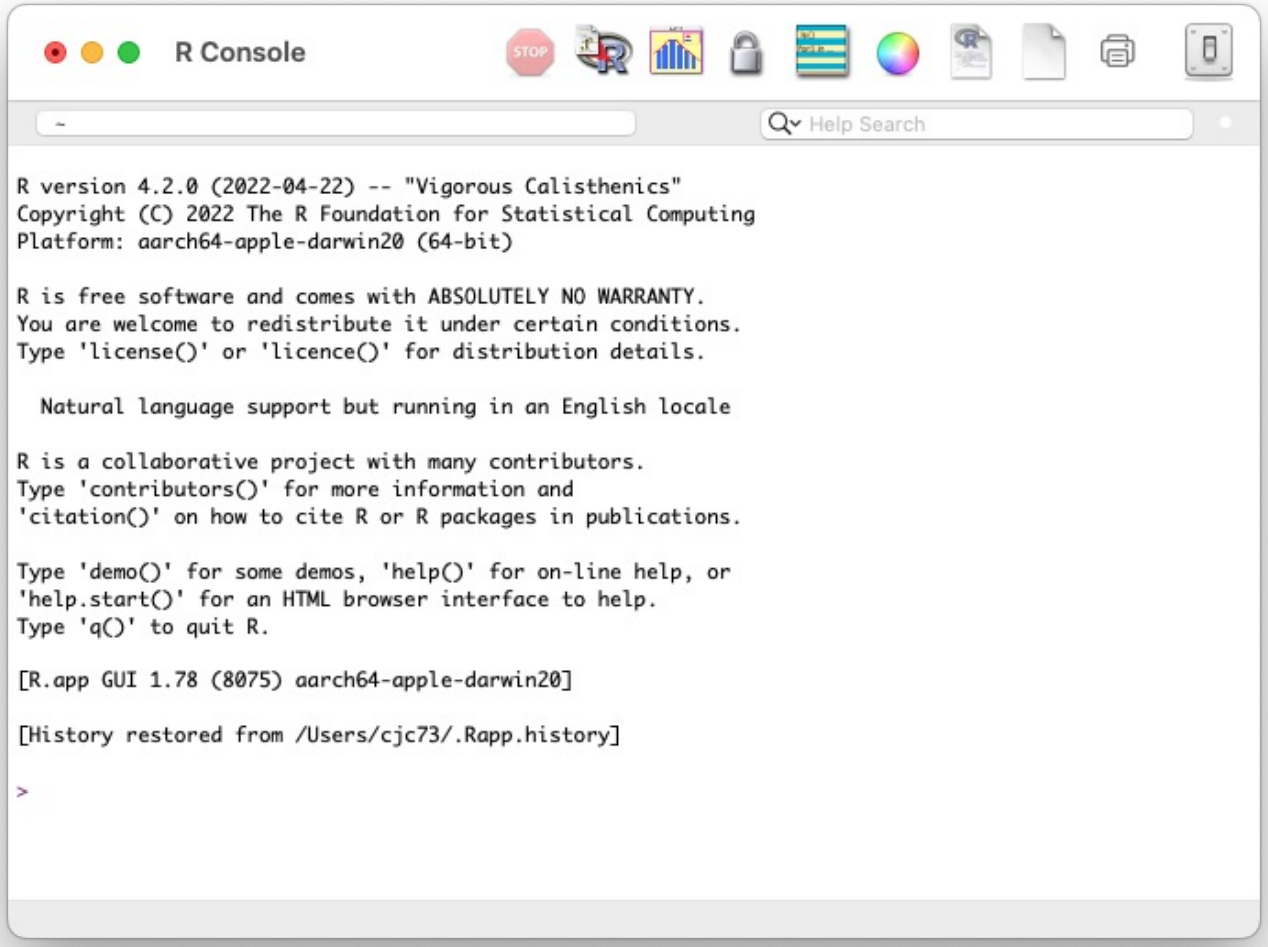
Base R

- 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

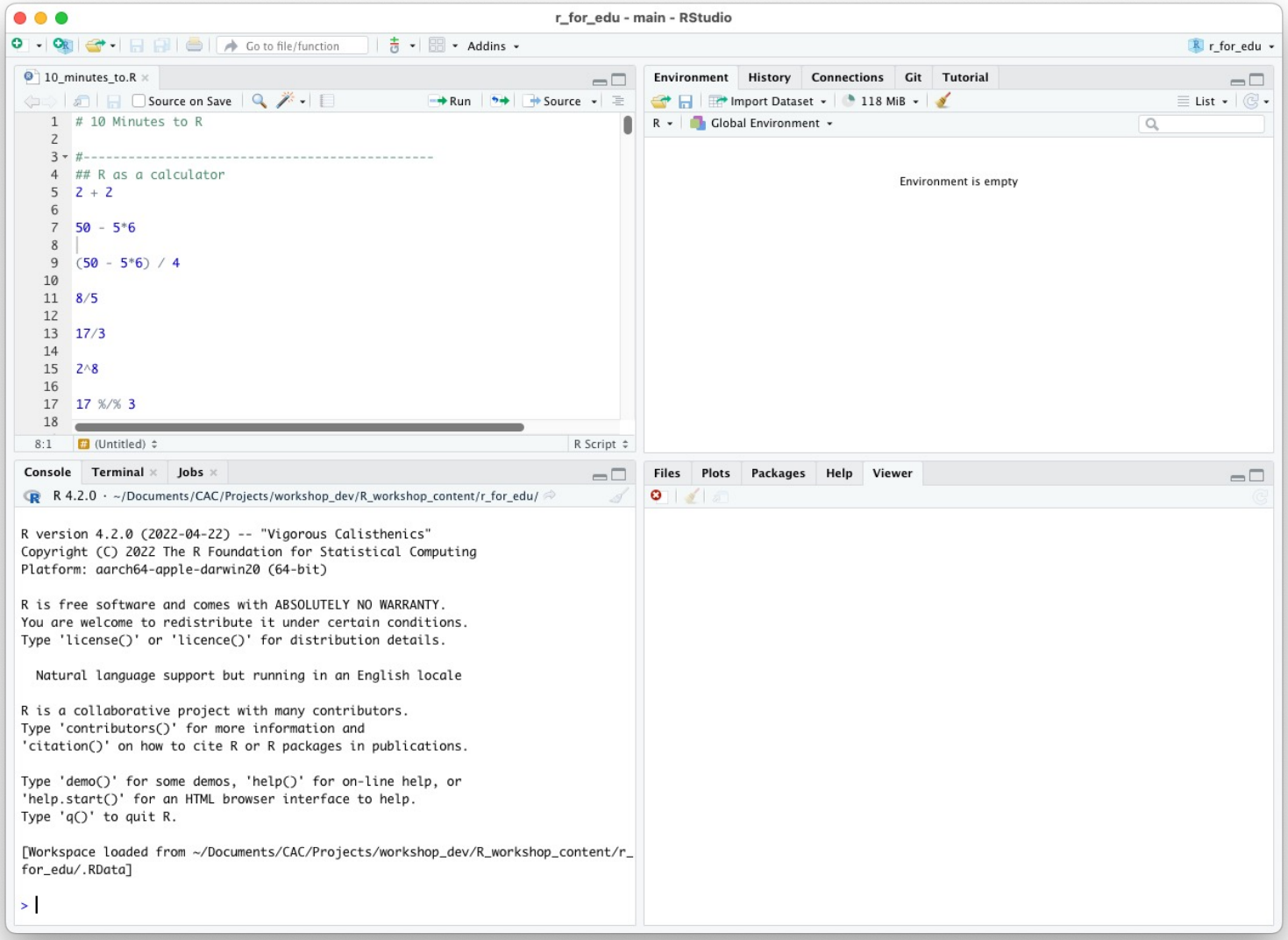
R Console


R Console
STOP
~
Help Search
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 (now Posit)
  - depends on installed R version
  - adds useful development, analysis and authoring features
- RStudio interface incorporates the R Console
  - Posit will incorporate Python compatibility
- Tip: If you want to install RStudio locally, install R and *then* install RStudio
- RStudio Cloud (soon to be Posit Cloud) https://rstudio.cloud is a hosted version of RStudio with the same interface as the desktop application.

RStudio Interface



More information

- Cornell Virtual Workshop in R: https://cvw.cac.cornell.edu/R/
  - CVW offers free self-paced, text-based modules covering a variety of computational focused topics. The CVW R topic complements today's workshop and covers using R on multiple cores and on supercomputer infrastructure.

- RStudio Cheatsheets:
  - https://www.rstudio.com/resources/cheatsheets/
  - Thoughtfully designed, single-page, double-sided reference sheets for major R packages.

More information

- Using R for teaching and research:
  - https://www.chrisbail.net/teaching
  - Chris Bail's work is a good example of incorporating R into teaching and research at undergraduate and graduate levels. Dr. Bail uses R for most aspects of his data collection and analysis.

- eBooks:
  - R for Data Science, Hadley Wickam and Garrett Grolemund - https://r4ds.had.co.nz
  - Advanced R (Programming), Hadley Wickam - https://adv-r.hadley.nz

More information

- Installing R for Jupyter Notebooks:
  - If you already use Jupyter, you can install the R jupyter kernel to use R in the familiar notebook environment. If you are on macOS, read the yellow warning box on the linked page. https://irkernel.github.io/installation/

- R packages on CRAN by area:
  - https://cran.r-project.org/web/views/