

Introduction to R and RStudio



Session outline

1. What is R?
2. Why should we use R?
3. R and RStudio
4. Packages
5. Using the console
6. Scripts

What is R?

- R is a free, **open source** language for statistical computing.

1992 Development began as research project in Auckland, by Robert Gentleman and Ross Ihaka, based on the S (1976) and S-PLUS (1988) languages.

1993 First release of R

2000 Version 1.0 released

2004 First UseR! conference in Vienna, Austria

2013 Version 3.0 released

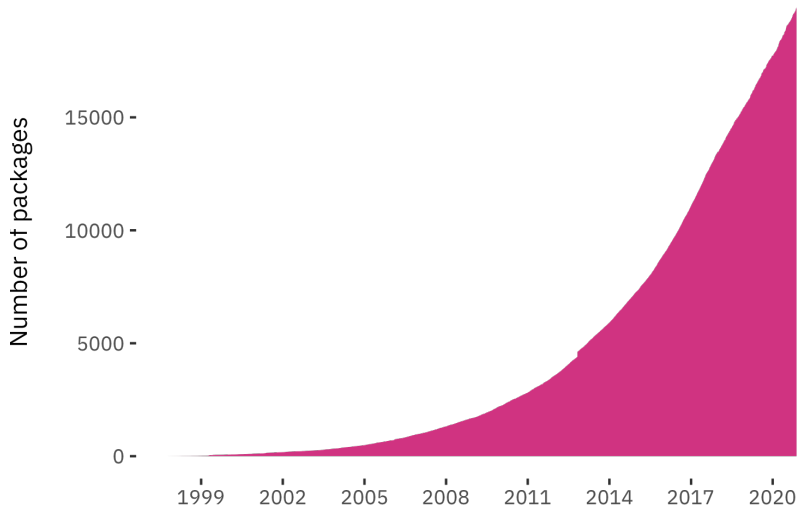
2015 R Consortium founded

2017 10,000 packages published on CRAN

2020 Version 4.0 released

- R is one of the **fastest growing** programming languages, especially for statistics and data science.
- A large (and growing) ecosystem, with over 18,000 packages.

There are over 18,000 packages for R



Why use R?

Strengths

- Free and open source
- Big (and friendly) user community
- Excellent resources
- Very flexible
- Over 18,000 packages
- Rapid development
- Industry adoption
- It's a programming language.

Why use R?

Strengths

- Free and open source
- Big (and friendly) user community
- Excellent resources
- Very flexible
- Over 18,000 packages
- Rapid development
- Industry adoption
- It's a programming language.

Weaknesses

- 18,000 packages.
Which should I use?
- Rapid development
- Package management
- R is slow?
 - R used to be slow; less true today
 - Many functions now written in C++ via Rcpp
- It's a programming language.



Licence

Open

Open

Open

Closed

Open

Version 1.0

2000

1994

2018

1985

1985

Easy of use*

Moderate

Hard

Moderate

Easy

Very hard

Extensibility

Excellent

Good

Good

Limited

Very limited

Industry support

Excellent

Excellent

Limited

Limited

Good

Performance

OK

Good

Very good

Good

Excellent

Execution

Interpreted

Interpreted

JIT

Interpreted

Compiled

*For statistical computing.

AS SEEN BY USERS OF ...

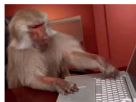
STATA



sas



STATA



sas



Should I learn R or Python?



RStudio

Installing R and RStudio

R is a language.

RStudio is an environment for working with R.

Installing R and RStudio

R is a language.

RStudio is an environment for working with R.

Installing R

Installing RStudio



CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2018-07-02 Feather Spray) [R-3.5.1.tar.gz](#), read [what's](#)

Installing R and RStudio

R is a language.

RStudio is an environment for working with R.

Installing R

Installing RStudio



Products

Resources

Pricing

About Us

Blogs



RStudio

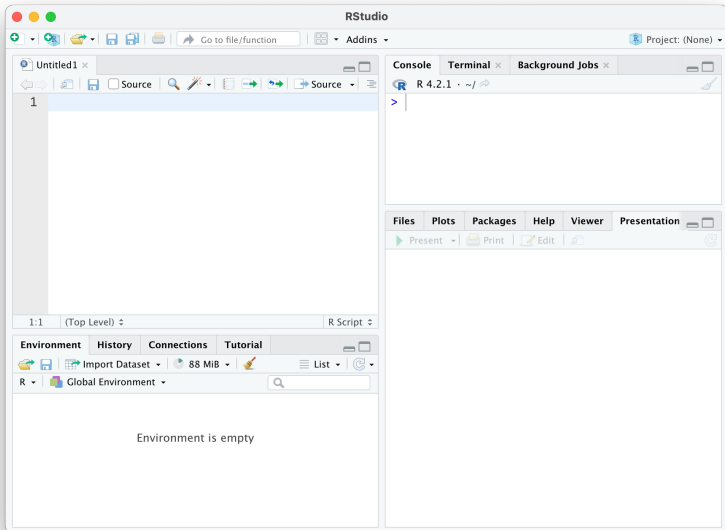
Open source and enterprise-ready
professional software for R

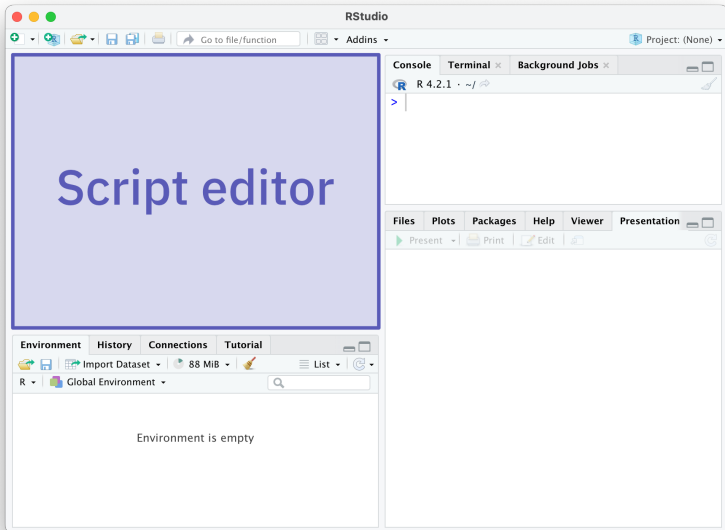
Download RStudio

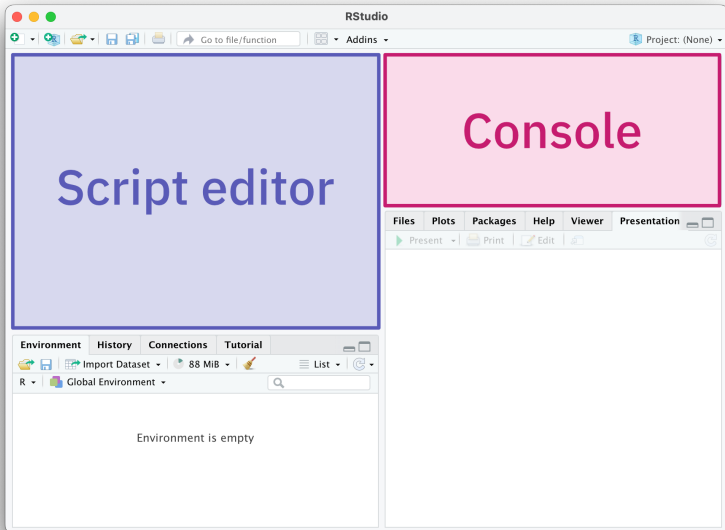
Discover Shiny

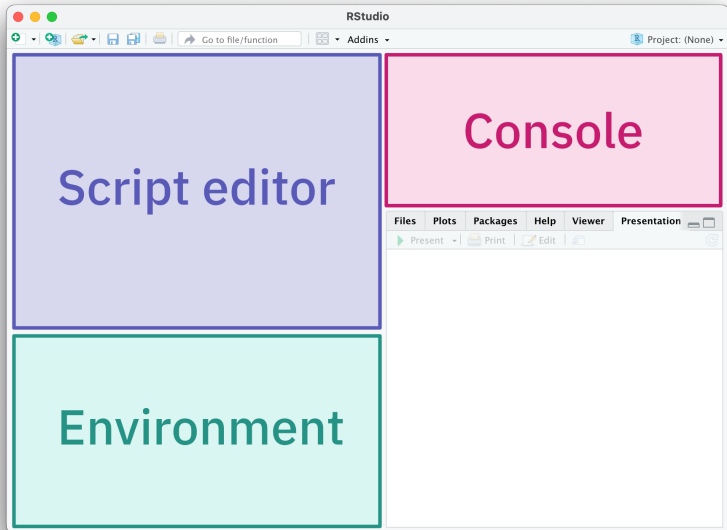
shinyapps.io Login

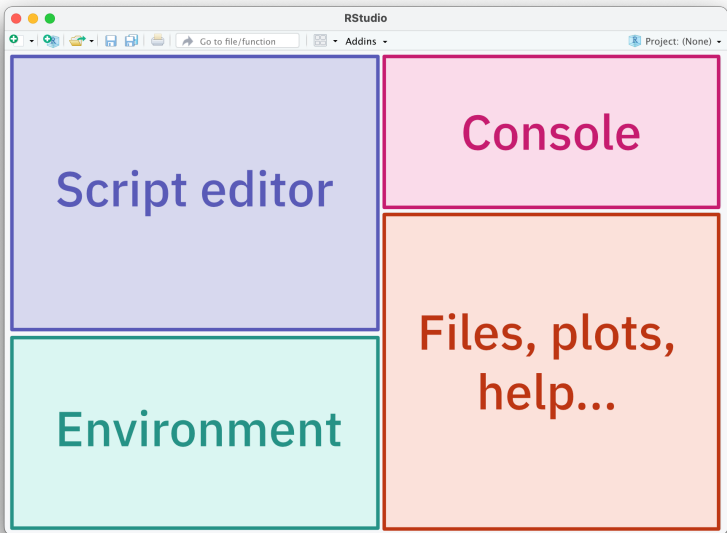
Discover RStudio Connect

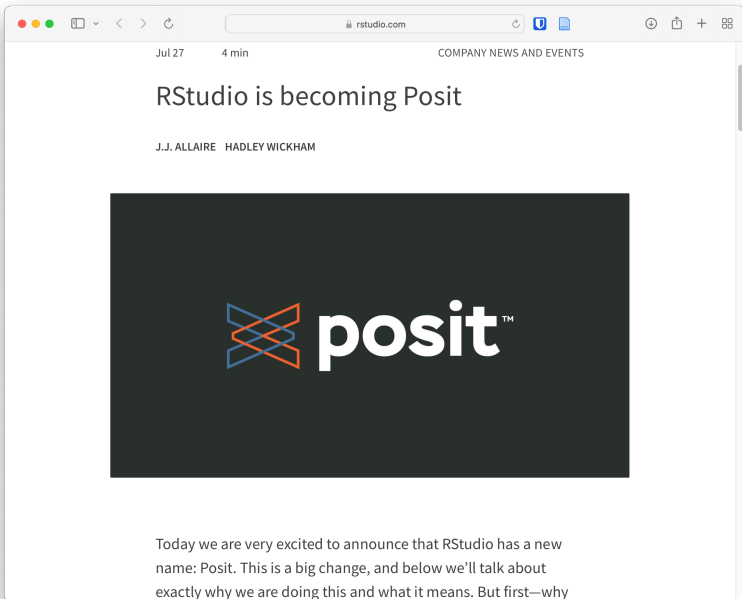












R packages

R packages

We use packages to **add functionality** to R.

A collection of scripts designed to implement a particular function or method.

They represent one of the principle strengths of R: if a method exists, someone has probably written a package.

R comes with many packages out-of-the-box, such as `datasets`, `graphics`, and `stats`. These are referred to as **'base'** packages.

Installing packages

We can install packages by typing:

```
install.packages("lme4")
```

To load this package, we'd then type:

```
library(lme4)
```

(Note the use of quotation marks.)



You should **install** packages once. But you need to **load** them every time you use them.

How to choose which packages to use?

My preferences:

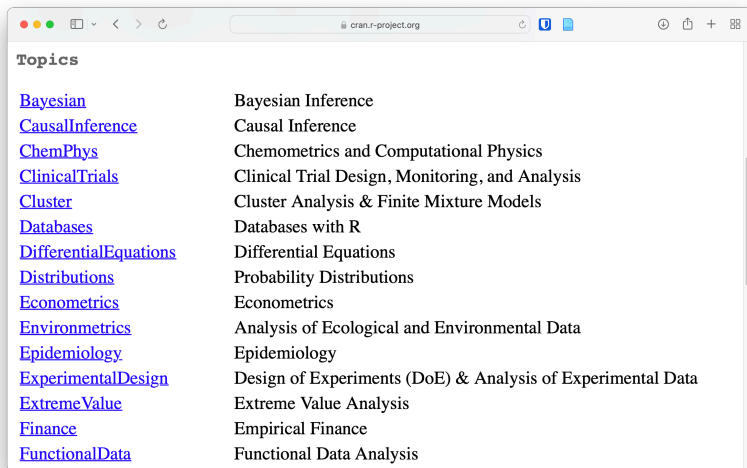
1. Use as **few** packages as possible (but as many as needed).
2. Use big **general** packages (rather than small specialised ones).
3. Use packages that have been **around for a long time** and are **regularly updated**.
4. Use **stable, reliable** packages.
5. At times, ignore all these rules.

Your preferences may differ!

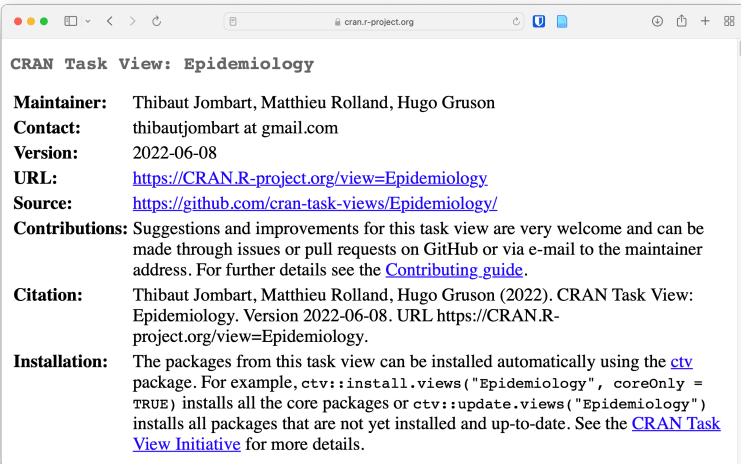
For example: `ggplot2`, `dplyr`, `tidyr`, `data.table`, `lme4`

Use the CRAN Task Views

cran.r-project.org/web/views



cran.r-project.org/web/views/Epidemiology



The image is a screenshot of a web browser window. The address bar shows the URL "cran.r-project.org". The page content is titled "CRAN Task View: Epidemiology". It lists the maintainers as Thibaut Jombart, Matthieu Rolland, and Hugo Gruson. The contact email is thibautjombart at gmail.com. The version is 2022-06-08. The URL is https://CRAN.R-project.org/view=Epidemiology. The source is https://github.com/cran-task-views/Epidemiology/. The contributions section states that suggestions and improvements are welcome and can be made through issues or pull requests on GitHub or via e-mail to the maintainer address. For further details, see the Contributing guide. The citation is Thibaut Jombart, Matthieu Rolland, Hugo Gruson (2022). CRAN Task View: Epidemiology. Version 2022-06-08. URL https://CRAN.R-project.org/view=Epidemiology. The installation section states that the packages from this task view can be installed automatically using the ctv package. For example, ctv::install.views("Epidemiology", coreOnly = TRUE) installs all the core packages or ctv::update.views("Epidemiology") installs all packages that are not yet installed and up-to-date. See the CRAN Task View Initiative for more details.

CRAN Task View: Epidemiology

Maintainer: Thibaut Jombart, Matthieu Rolland, Hugo Gruson

Contact: thibautjombart at gmail.com

Version: 2022-06-08

URL: <https://CRAN.R-project.org/view=Epidemiology>

Source: <https://github.com/cran-task-views/Epidemiology/>

Contributions: Suggestions and improvements for this task view are very welcome and can be made through issues or pull requests on GitHub or via e-mail to the maintainer address. For further details see the [Contributing guide](#).

Citation: Thibaut Jombart, Matthieu Rolland, Hugo Gruson (2022). CRAN Task View: Epidemiology. Version 2022-06-08. URL <https://CRAN.R-project.org/view=Epidemiology>.

Installation: The packages from this task view can be installed automatically using the [ctv](#) package. For example, `ctv::install.views("Epidemiology", coreOnly = TRUE)` installs all the core packages or `ctv::update.views("Epidemiology")` installs all packages that are not yet installed and up-to-date. See the [CRAN Task View Initiative](#) for more details.

Bonus

pak

A Fresh Approach to R Package Installation

pak installs R packages from CRAN, Bioconductor, GitHub, URLs, git repositories, local files and directories. It is an alternative to `install.packages()` and `devtools::install_github()`. pak is fast, safe and convenient.

-  [Short tour](#)
-  [Quick links \(start here if in doubt!\)](#)
-  [Features](#)
-  [Installation](#)
-  [License](#)

LINKS

[View on CRAN](#)

[Browse source code](#)

[Report a bug](#)

LICENSE

[GPL-3](#)

COMMUNITY

[Code of conduct](#)

CITATION

[Citing pak](#)

DEVELOPERS

[Gábor Csárdi](#)

Author, maintainer

pak.r-lib.org

Scripts

Scripts are plain text files containing R code

Similar to Stata do-files or SPSS syntax files.

```
# Title:      Example script
# Author:     Author's name
# Date:       Today's date
# Description: This script...
```

Header

```
library(tidyverse)
library(here)
library(lme4)
```

Load packages

```
# Import raw data
raw <- read.dta("source_data.dta")
```

Comments

```
# Fit linear model
model_1 <- lm(y ~ x1 + x2, data = raw)
```

Scripts (cont.)

- You should annotate your scripts with **comments**.

```
# This is a comment.
```

What makes a good comment?

- Save your scripts with the **.R** file extension
e.g., 1-Data-prep.R.
- Install packages once, load them **every time you use them**. Place at the top of your script.

```
library(tidyverse)  
library(here)
```

Scripts (cont.)

- You should annotate your scripts with **comments**.

```
# This is a comment.
```

What makes a good comment?

- Save your scripts with the **.R** file extension
e.g., 1-Data-prep.R.
- Install packages once, load them **every time you use them**. Place at the top of your script.

```
library(tidyverse)  
library(here)
```

Explore at the console; save anything that matters in a script.

Organising your scripts

You can organise your scripts with [sections](#) and [headers](#).

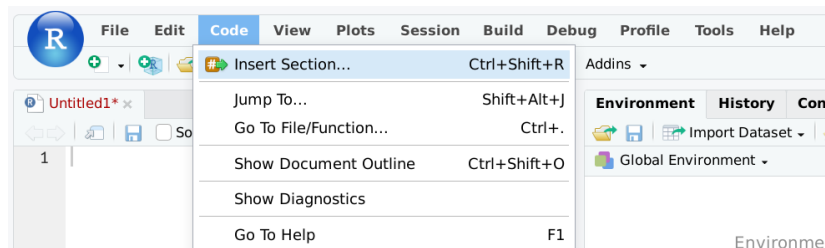
```
# Section One -----  
  
# Section Two =====  
  
### Section Three #####
```

Organising your scripts

You can organise your scripts with **sections** and **headers**.

```
# Section One -----  
# Section Two =====  
### Section Three #####
```

These can be inserted in RStudio with **Ctrl** + **↑** + **R** or
Code > **Insert Section...**



Combining several scripts

- Most analyses won't fit in a single script/file.
 - ~> You might have separate scripts for data import, cleaning, analysis, plotting...
- We can use the `source` command to execute code contained in another file.

```
source("1-Import-raw-data.R")  
source("2-Data-cleaning.R")
```

The R console

Console ~/ ↗

```
R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.
```

```
normal, x[x ≤ -0.01], x[abs(x)<0.01], x[x ≥ 0.01], 19/01/2038 03:14:07,
"string", NA/NaN/NULL, FALSE, TRUE, Inf, stderr, warn, error.
> |
```

Tips for the R console

- Access previous commands with  / .
- Search previous commands with  + .
- Clear the screen with  + .

Operators

Symbols that tell the compiler to perform specific mathematical or logical manipulations.

Arithmetic

`+` addition

`-` subtraction

`*` multiplication

`/` division

`^` or `**` raising to a power

Relational

`x < y` less than

`x > y` greater than

`x <= y` less than or equal to

`x == y` equal

`x != y` not equal

For more, see <https://www.statmethods.net/management/operators.html>

Getting help

You can get help for any function with `?`. For example:

```
?rnorm    # Help for the 'rnorm' function  
?install.packages
```

All help files have a common structure:

Description → Usage → Arguments → Details → See Also → Examples

- I find the 'Arguments' and 'Examples' sections the most useful.
- It's worth familiarising yourself with this structure.

Resources

1. A Succinct Introduction to R

Steve Haroz

<http://r-guide.steveharoz.com>

2. R for Data Science

Hadley Wickham and Garrett Golemund

<https://r4ds.hadley.nz>

3. Advanced R

Hadley Wickham

<https://adv-r.hadley.nz>