The RStudio IDE

using RStudio to manage code, data, and files

by Martin Frigaard

Written: September 30 2021

Updated: November 30 2021

Created using the "λέξις" theme

The RStudio IDE

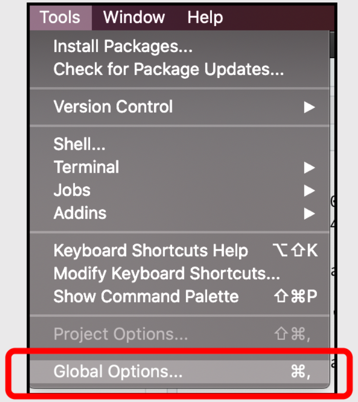
Managing your code, data, and files with RStudio

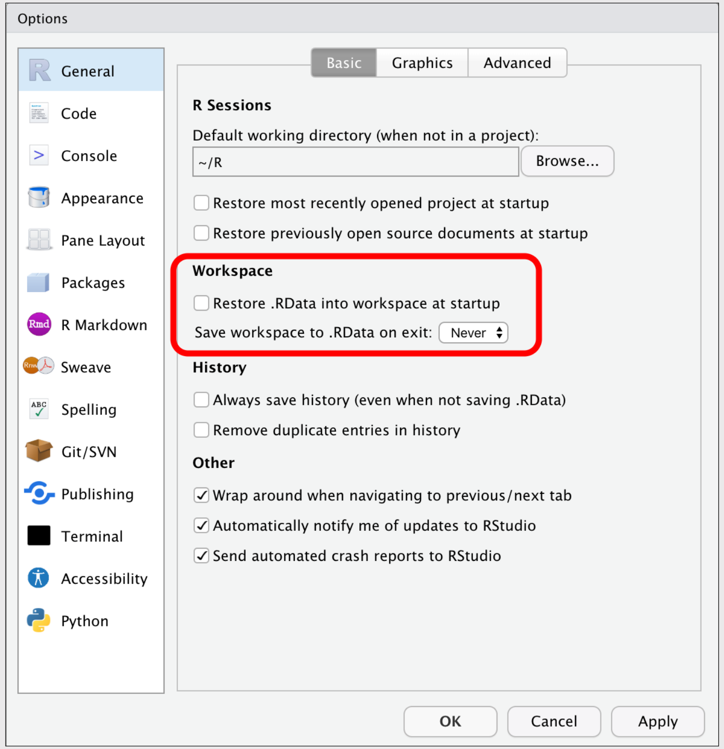
Change RStudio's Default Settings

Click on Tools > Global Options...

- We want to uncheck "*Restore .RData into work space at start up*"

- We also want to make sure we change "*Save work space to .Rdata on exit*" to "*Never*"





R Sessions

Default working directory (when not in a project):

~/R Browse...

☐ Restore most recently opened project at startup

☐ Restore previously open source documents at startup

Workspace

☐ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

History

☐ Always save history (even when not saving .RData)

☐ Remove duplicate entries in history

Other

☑ Wrap around when navigating to previous/next tab

☑ Automatically notify me of updates to RStudio

☑ Send automated crash reports to RStudio

Customize RStudio

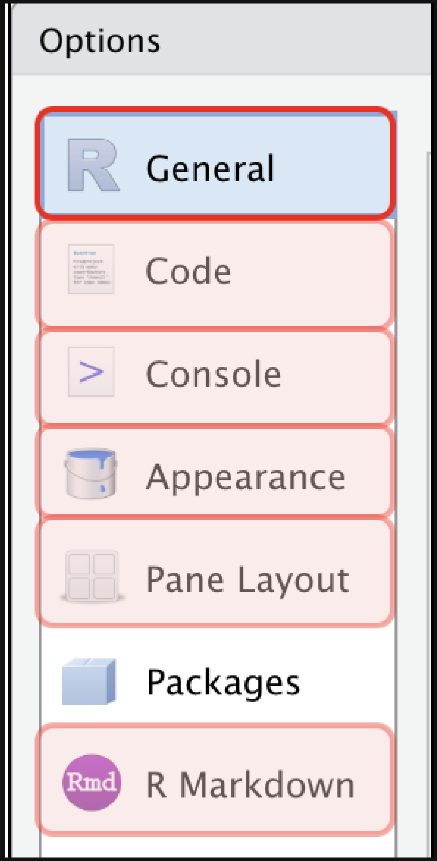
Code

Console

Appearance

Pane Layout

R Markdown

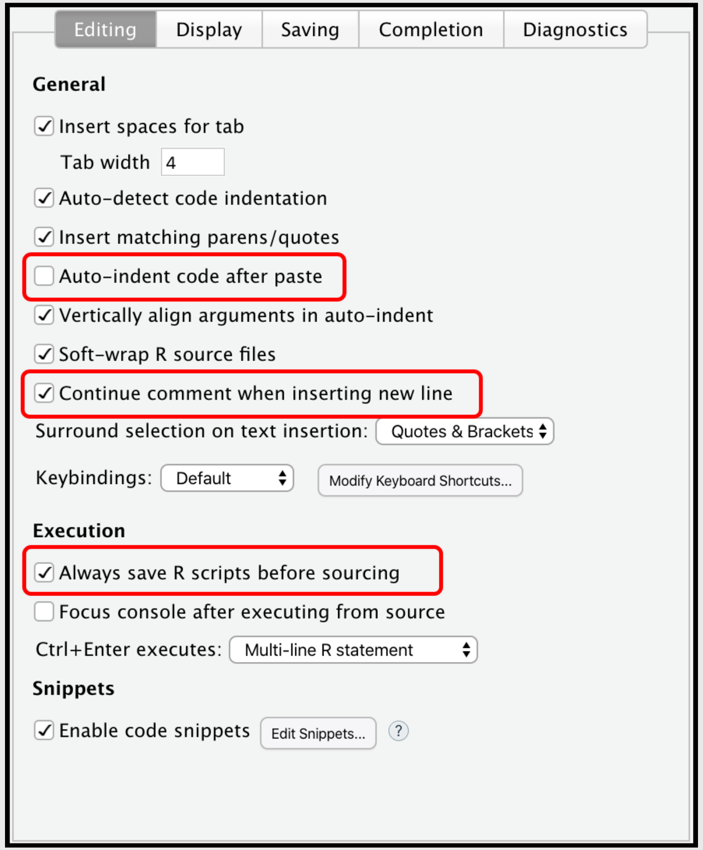


Code Editing

- Auto indent?

- Continue comment lines?

- Save R scripts before sourcing?



General

☑ Insert spaces for tab

Tab width 4

☑ Auto-detect code indentation

☑ Insert matching parens/quotes

□ Auto-indent code after paste

☑ Vertically align arguments in auto-indent

☑ Soft-wrap R source files

☑ Continue comment when inserting new line

Surround selection on text insertion: Quotes & Brackets

Keybindings: Default

Execution

☑ Always save R scripts before sourcing

□ Focus console after executing from source

Ctrl+Enter executes: Multi-line R statement

Snippets

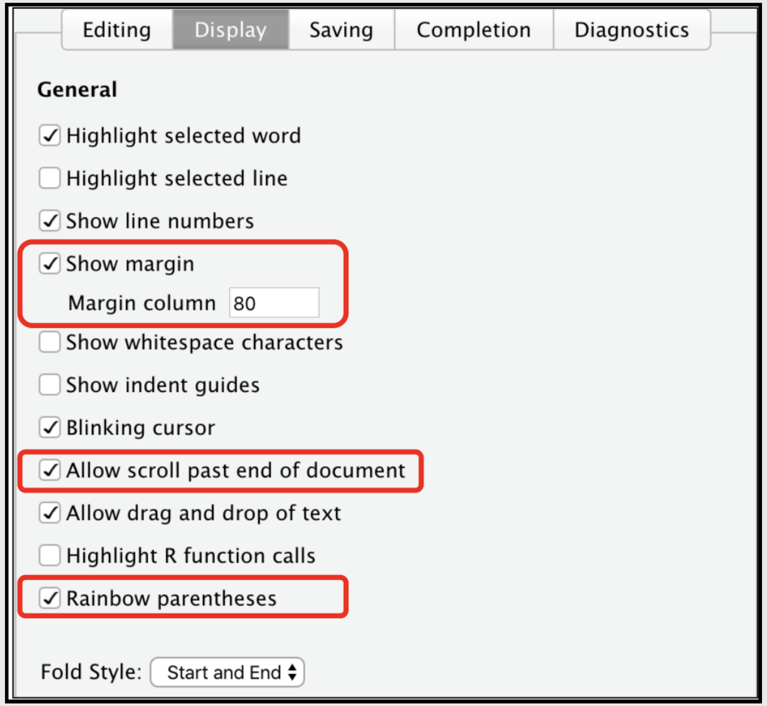
☑ Enable code snippets

Code Display

- Margins?

- Scrolling?

- Rainbow parentheses?



General

☑ Highlight selected word

□ Highlight selected line

☑ Show line numbers

☑ Show margin

Margin column 80

□ Show whitespace characters

□ Show indent guides

☑ Blinking cursor

☑ Allow scroll past end of document

☑ Allow drag and drop of text

□ Highlight R function calls

☑ Rainbow parentheses

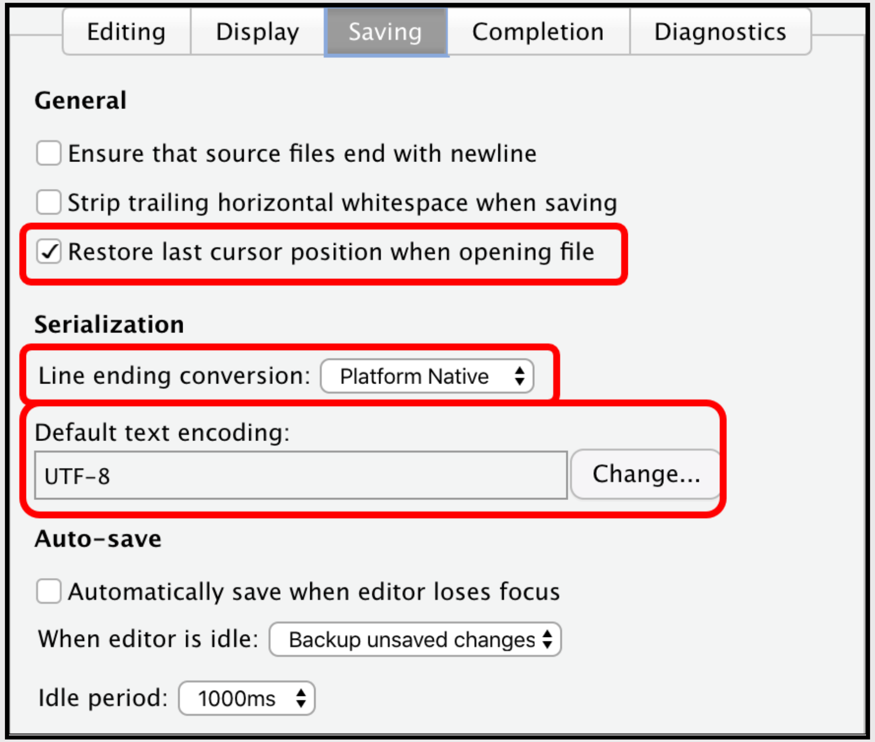
Fold Style: Start and End

Code Saving

- Cursor position?

- Line endings?

- Text encoding?



General

□ Ensure that source files end with newline

□ Strip trailing horizontal whitespace when saving

☑ Restore last cursor position when opening file

Serialization

Line ending conversion: Platform Native

Default text encoding:

UTF-8

Auto-save

□ Automatically save when editor loses focus

When editor is idle: Backup unsaved changes

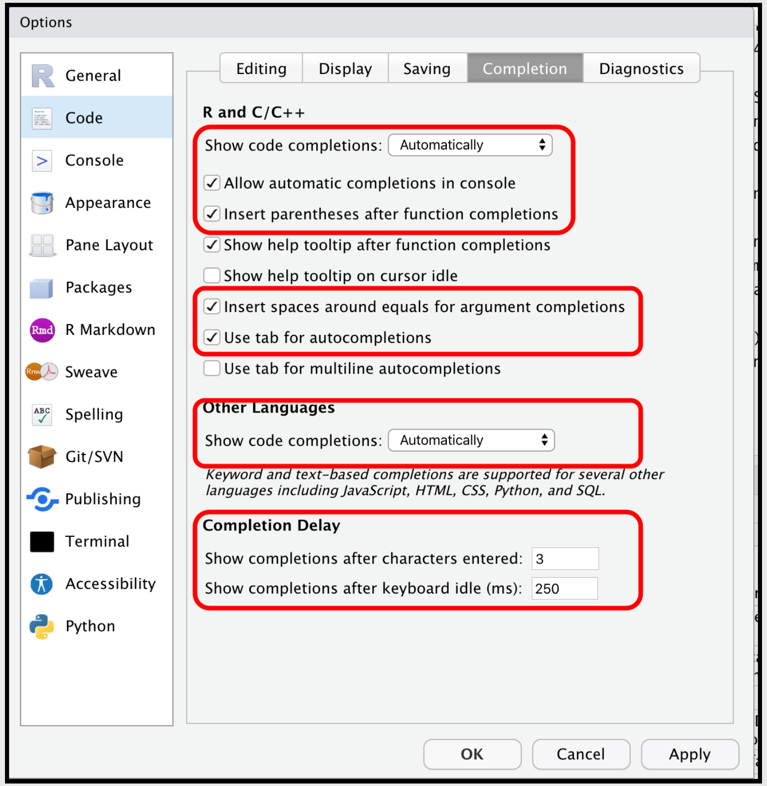
Idle period: 1000ms

Code Completion

- Insert parentheses?

- Insert spaces?

- Completion delay setting?



R and C/C++

Show code completions: Automatically

☑ Allow automatic completions in console

☑ Insert parentheses after function completions

☑ Show help tooltip after function completions

□ Show help tooltip on cursor idle

☑ Insert spaces around equals for argument completions

☑ Use tab for autocompletions

□ Use tab for multiline autocompletions

Other Languages

Show code completions: Automatically

Keyword and text-based completions are supported for several other languages including JavaScript, HTML, CSS, Python, and SQL.

Completion Delay

Show completions after characters entered: 3

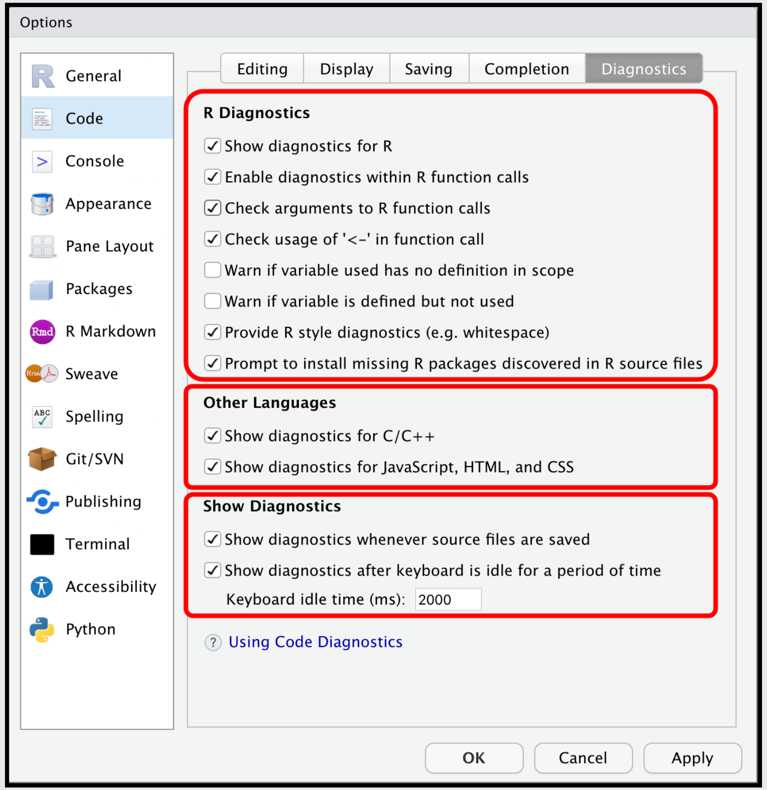
Show completions after keyboard idle (ms): 250

Code Diagnostics

- Check your R Code?

- Check other languages?

- How long?



R Diagnostics

☑ Show diagnostics for R

☑ Enable diagnostics within R function calls

☑ Check arguments to R function calls

☑ Check usage of '<-' in function call

☐ Warn if variable used has no definition in scope

☐ Warn if variable is defined but not used

☑ Provide R style diagnostics (e.g. whitespace)

☑ Prompt to install missing R packages discovered in R source files

Other Languages

☑ Show diagnostics for C/C++

☑ Show diagnostics for JavaScript, HTML, and CSS

Show Diagnostics

☑ Show diagnostics whenever source files are saved

☑ Show diagnostics after keyboard is idle for a period of time

Keyboard idle time (ms): 2000

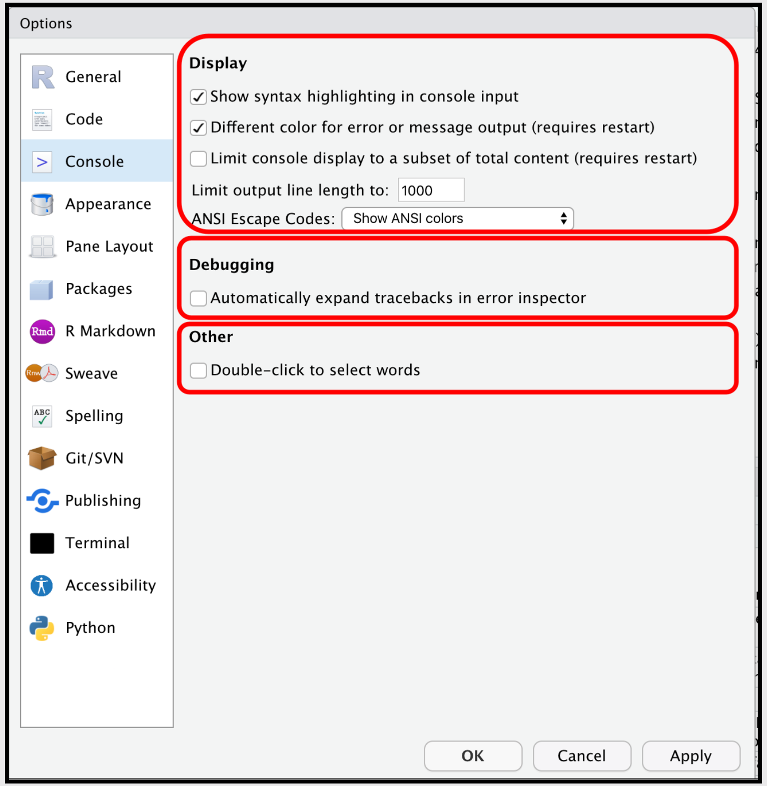
Using Code Diagnostics

Console

- Display?

- Debugging?

- Other?



Display

☑ Show syntax highlighting in console input

☑ Different color for error or message output (requires restart)

□ Limit console display to a subset of total content (requires restart)

Limit output line length to: 1000

ANSI Escape Codes: Show ANSI colors

Debugging

□ Automatically expand tracebacks in error inspector

Other

□ Double-click to select words

Appearance

- **RStudio theme?**

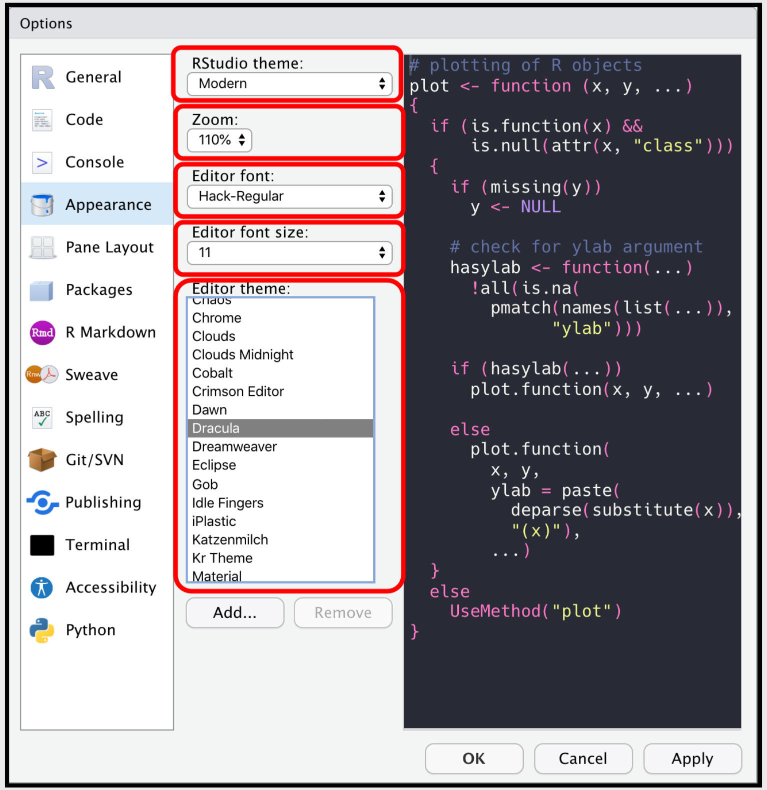
- **Zoom?**

- Also hold ⌘ and press + on macOS

- Also hold `ctrl` and press + on Windows

- **Font?**

- **Editor theme?**



RStudio theme: Modern

Zoom: 110%

Editor font: Hack-Regular

Editor font size: 11

Editor theme: Dracula

```
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
            "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)"),
        ...)
  }
  else
    UseMethod("plot")
}
```

Pane layout

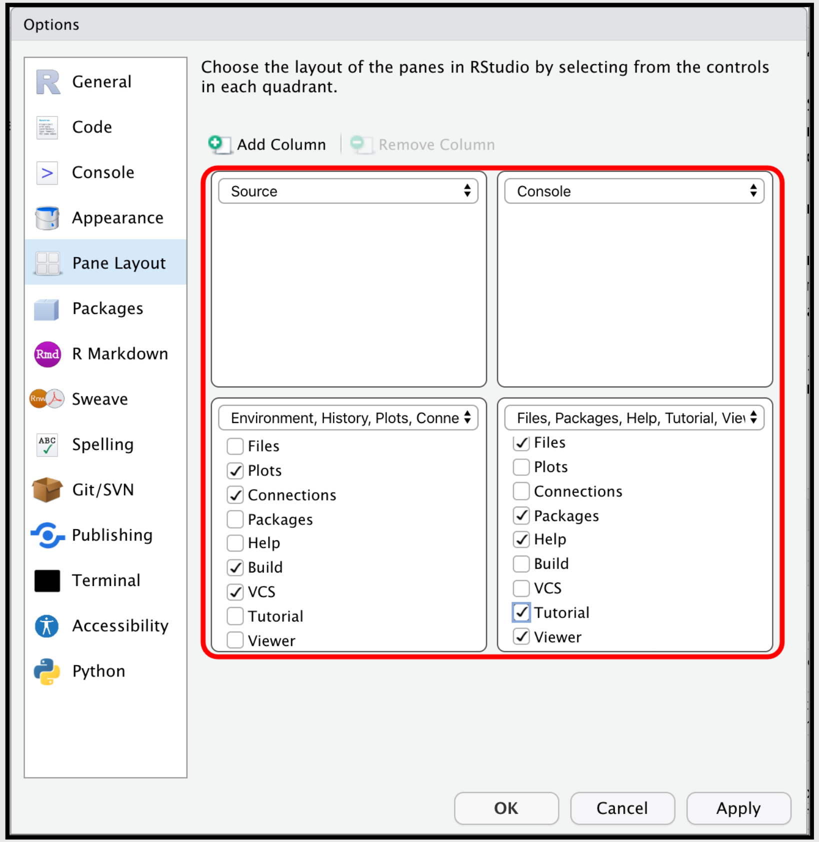
- Source?

- Console?

Combining pane elements?

- Plots, Connections, Build, VCS, Presentation

- Files, Packages, Help, Tutorial, Viewer



Options

Choose the layout of the panes in RStudio by selecting from the controls in each quadrant.

Source

Console

Environment, History, Plots, Conne

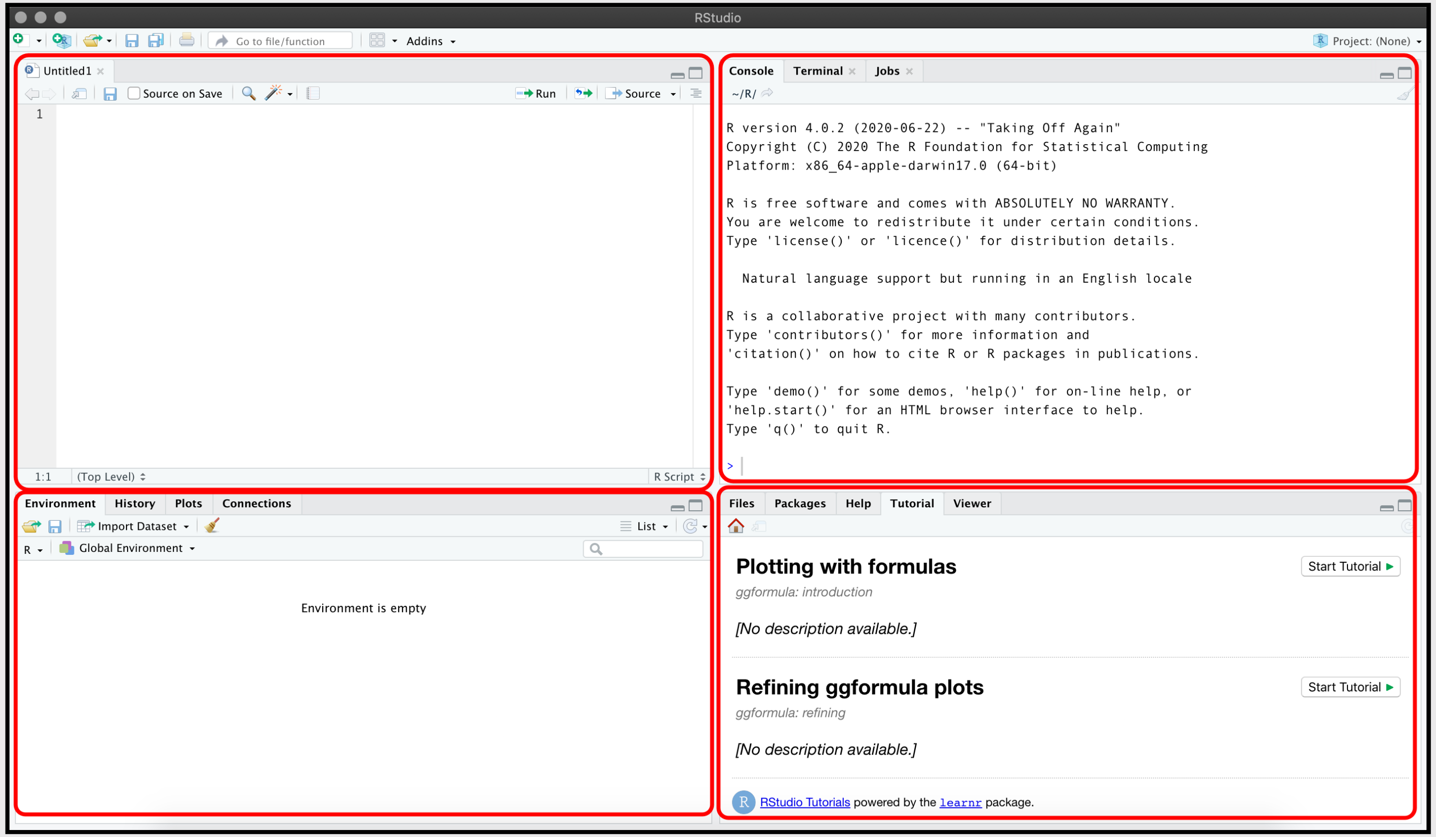
☐ Files
☑ Plots
☑ Connections
☐ Packages
☐ Help
☑ Build
☑ VCS
☐ Tutorial
☐ Viewer

Files, Packages, Help, Tutorial, Viev

☑ Files
☐ Plots
☐ Connections
☑ Packages
☑ Help
☐ Build
☐ VCS
☑ Tutorial
☑ Viewer

Pane layout view

Standard layout options



```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

>
```

Plotting with formulas

ggformula: introduction

[No description available.]

Refining ggformula plots

ggformula: refining

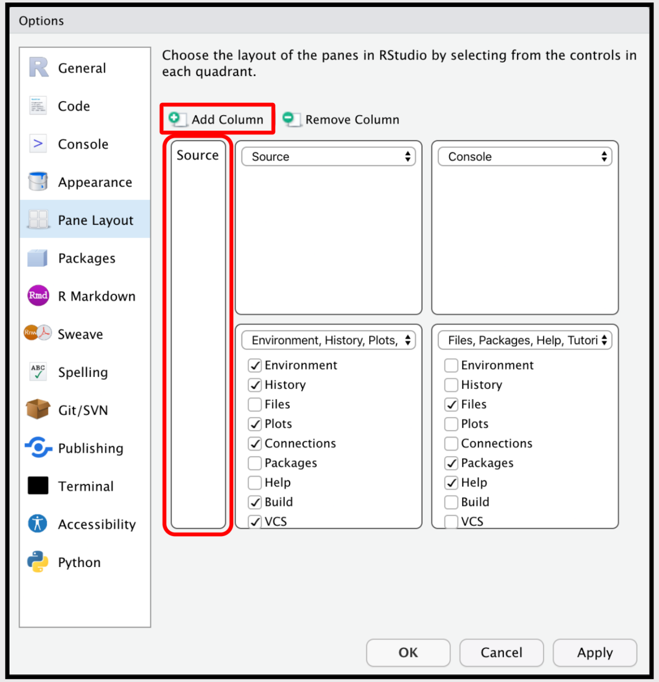
[No description available.]

RStudio Tutorials powered by the `learnr` package.

Pane layout: add column

Two screens?

add a Source column and rearrange the panes



Choose the layout of the panes in RStudio by selecting from the controls in each quadrant.

Source

Source

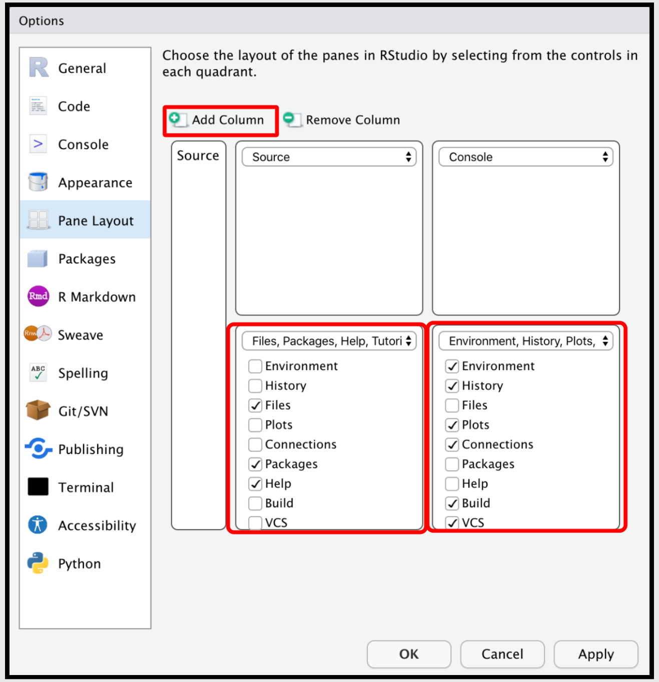
Console

Environment, History, Plots,

☑ Environment
☑ History
□ Files
☑ Plots
☑ Connections
□ Packages
□ Help
☑ Build
☑ VCS

Files, Packages, Help, Tutori

□ Environment
□ History
☑ Files
□ Plots
□ Connections
☑ Packages
☑ Help
□ Build
□ VCS



Choose the layout of the panes in RStudio by selecting from the controls in each quadrant.

Source

Source

Console

Files, Packages, Help, Tutori

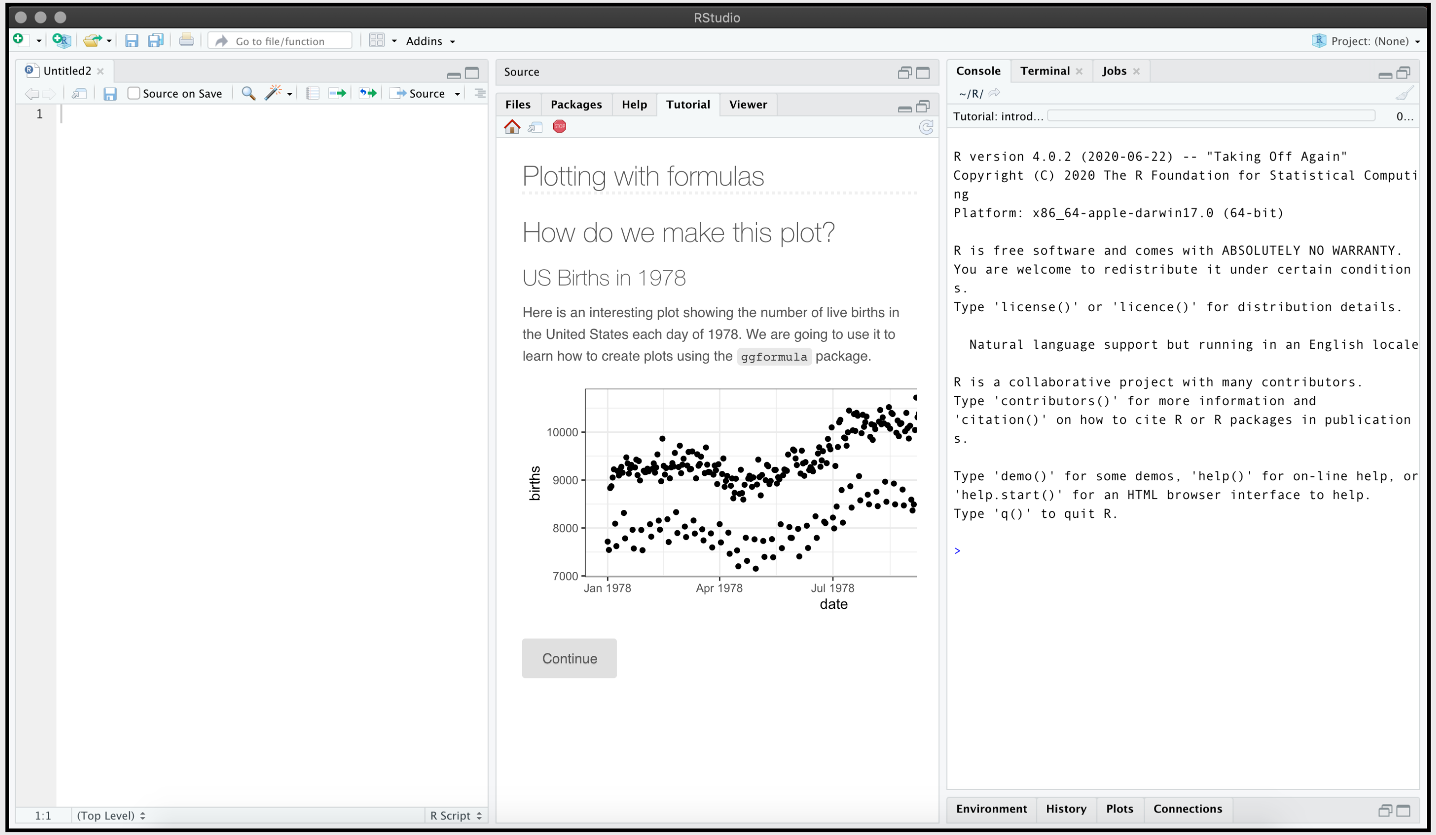
□ Environment
□ History
☑ Files
□ Plots
□ Connections
☑ Packages
☑ Help
□ Build
□ VCS

Environment, History, Plots,

☑ Environment
☑ History
□ Files
☑ Plots
☑ Connections
□ Packages
□ Help
☑ Build
☑ VCS

Pane layout: add column view

Now you see **Source**, **Tutorial**, and **Console** panes on a single screen!



Plotting with formulas

How do we make this plot?

US Births in 1978

Here is an interesting plot showing the number of live births in the United States each day of 1978. We are going to use it to learn how to create plots using the `ggformula` package.

```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computi
ng
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain condition
s.
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 publication
s.

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.

>
```

RStudio Projects

Why RStudio Projects?

Keep track of all your files with RStudio project files (`.Rproj`).

Self contained

> Using R projects keeps track or your current working directory!

Project orientated

> .Rproj files make bundling and shipping files and folders easier!

Why RStudio Projects?

Keep track of all your files with RStudio project files (`.Rproj`).

Avoid removing all the files

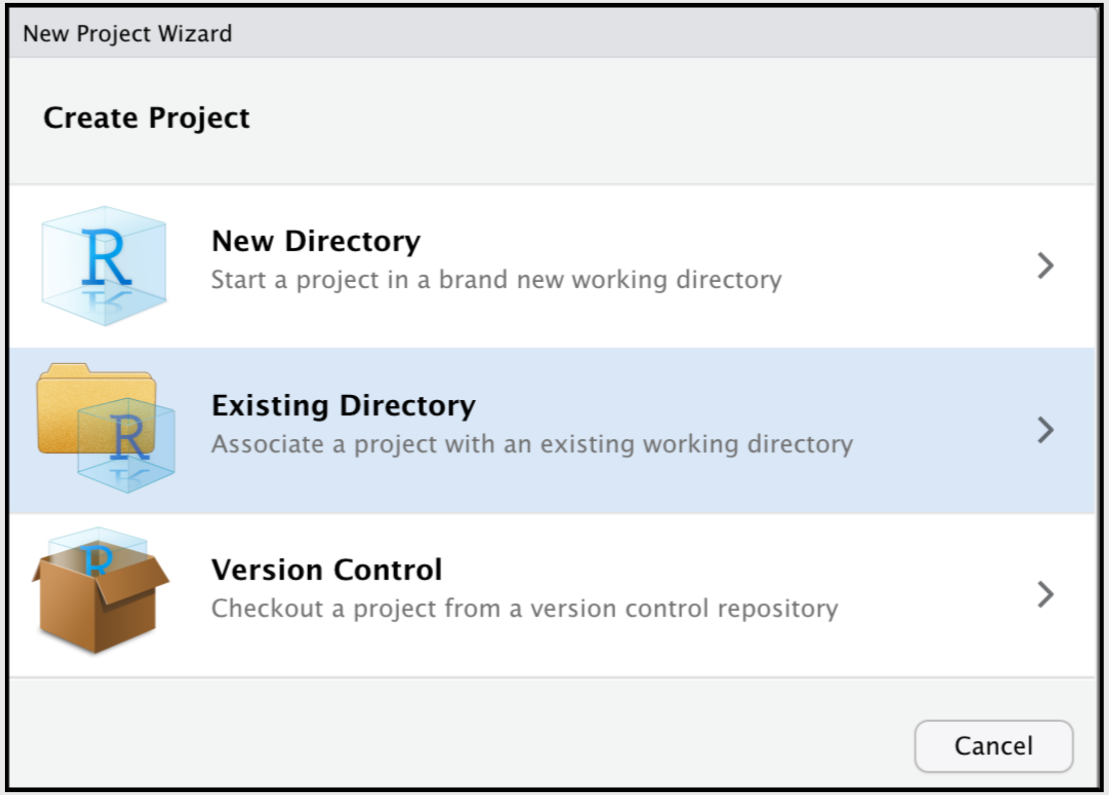
Don't do this!

```
rm(list = ls())
```

> `.Rproj` files keep all the files associated with a project together — input data, R scripts, analytic results, figures.

Creating RStudio project in existing folder

Click on *'Project: (None)'* > *'New Project'*



New Project Wizard

Create Project

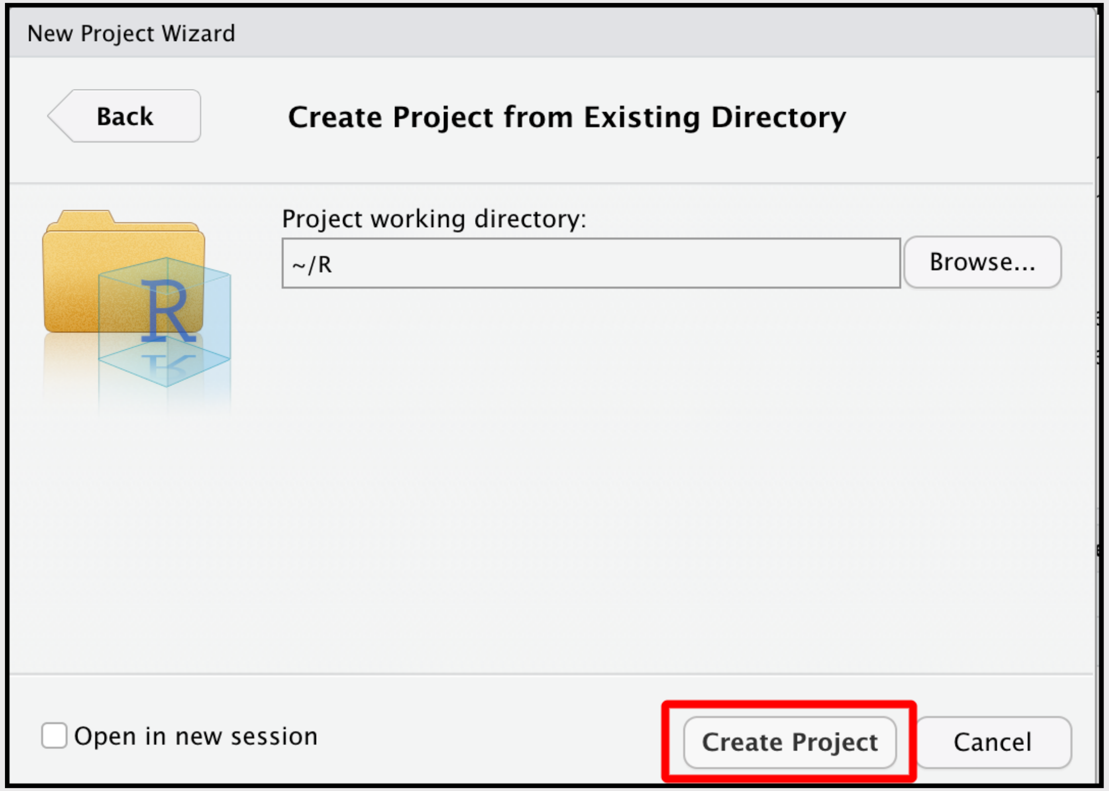
New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Creating RStudio project in existing folder

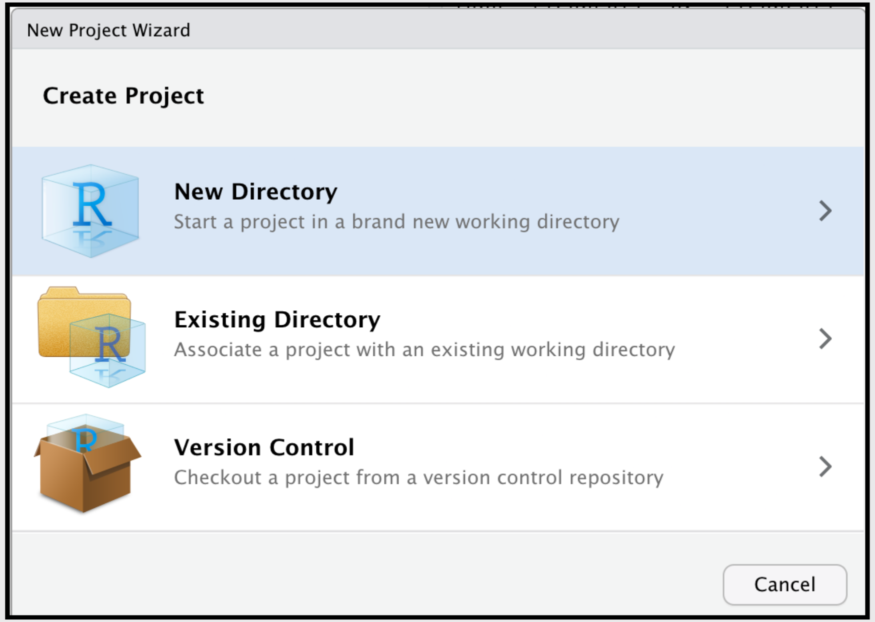
Click on *'Browse > 'Create Project'*



Project working directory: ~/R

Creating RStudio projects in new folder

Click on Project: (None) > New Project



New Project Wizard

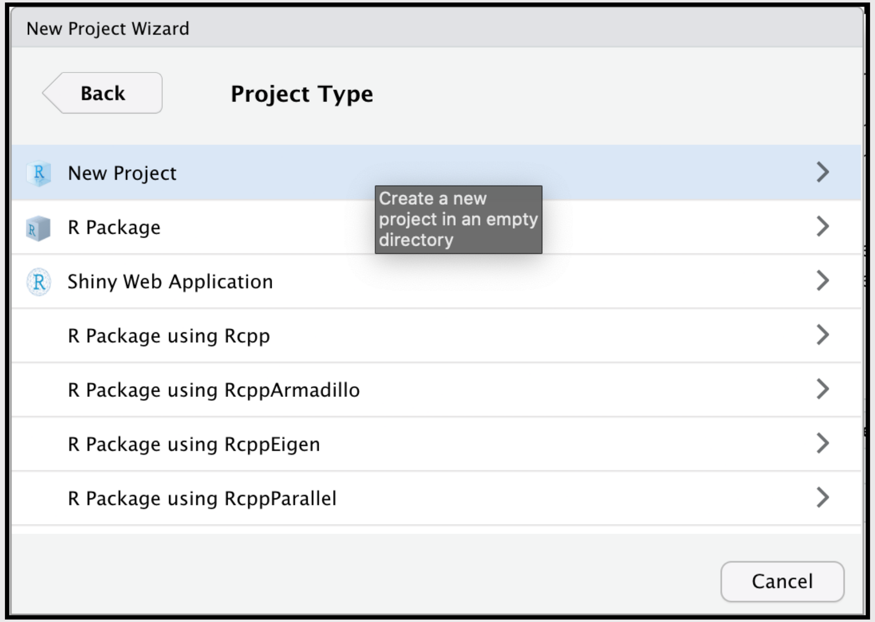
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Select project type



New Project Wizard

Project Type

New Project

R Package

Shiny Web Application

R Package using Rcpp

R Package using RcppArmadillo

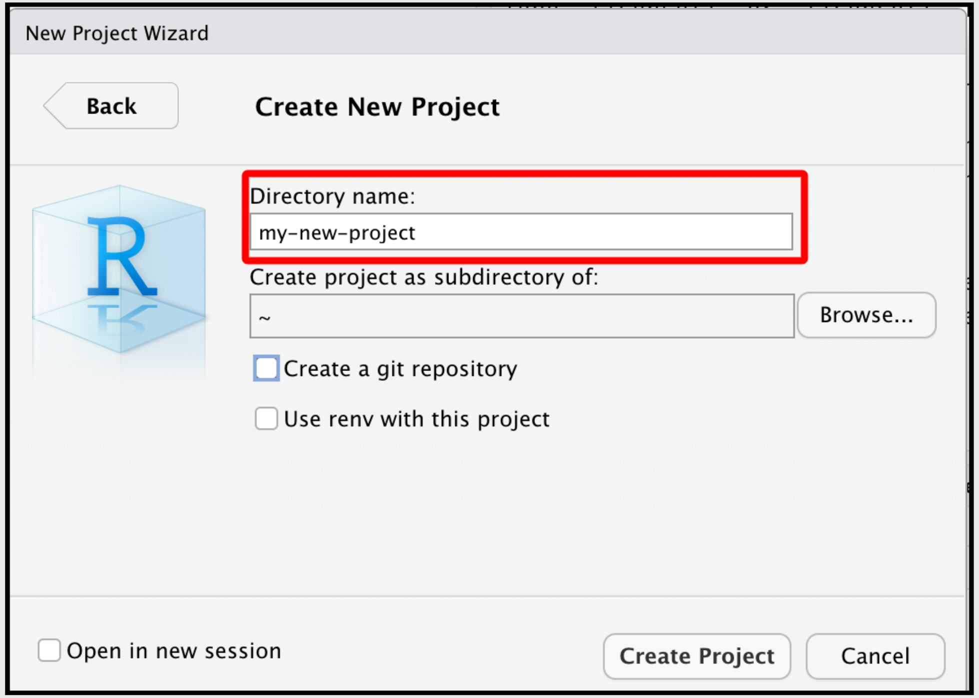
R Package using RcppEigen

R Package using RcppParallel

Create a new project in an empty directory

Creating RStudio projects in new folder

Create new folder name



Directory name: my-new-project

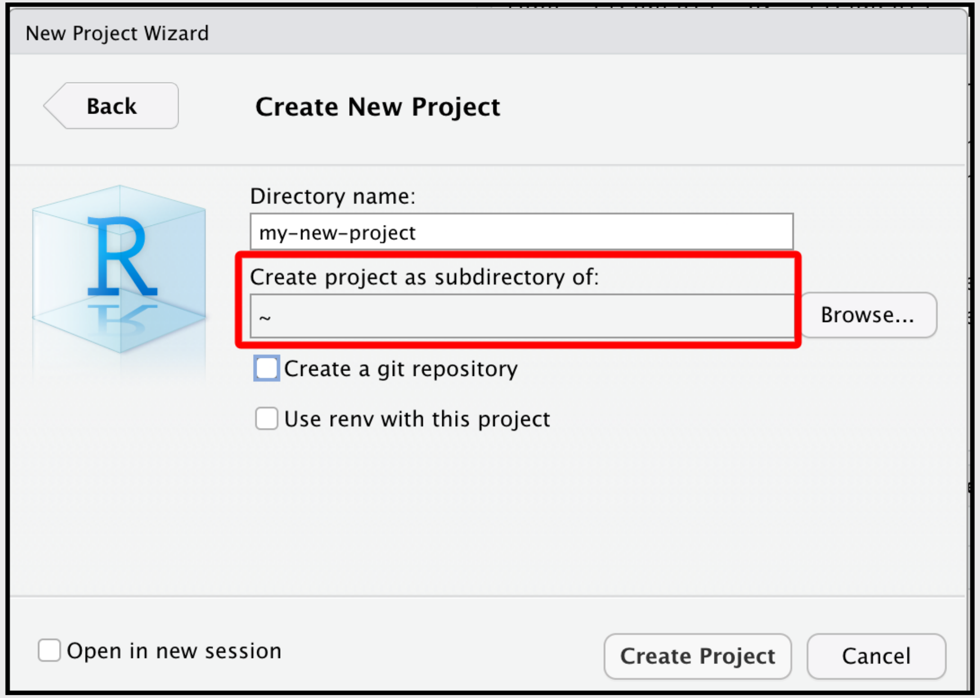
Create project as subdirectory of: ~

□ Create a git repository

□ Use renv with this project

□ Open in new session

Choose parent folder



Directory name: my-new-project

Create project as subdirectory of: ~

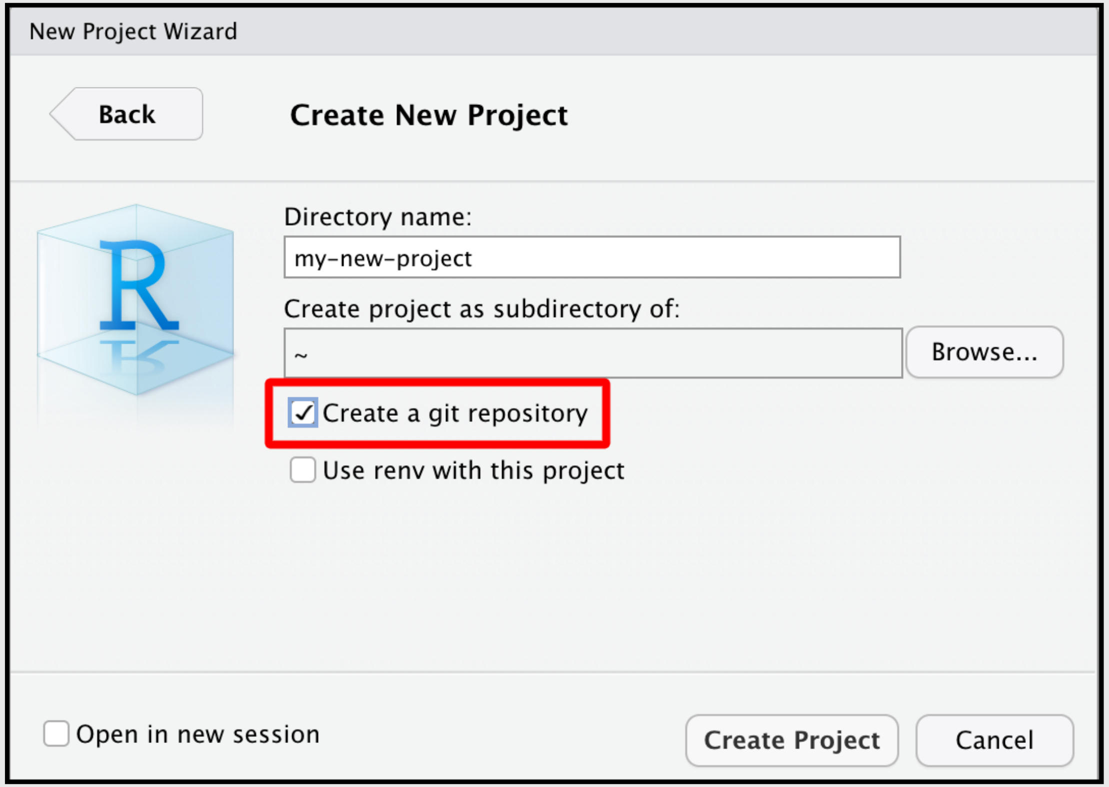
□ Create a git repository

□ Use renv with this project

□ Open in new session

Creating RStudio projects in new folder

If you have Git installed, select '*Create a git repository*'



New Project Wizard

Create New Project

Directory name: my-new-project

Create project as subdirectory of: ~

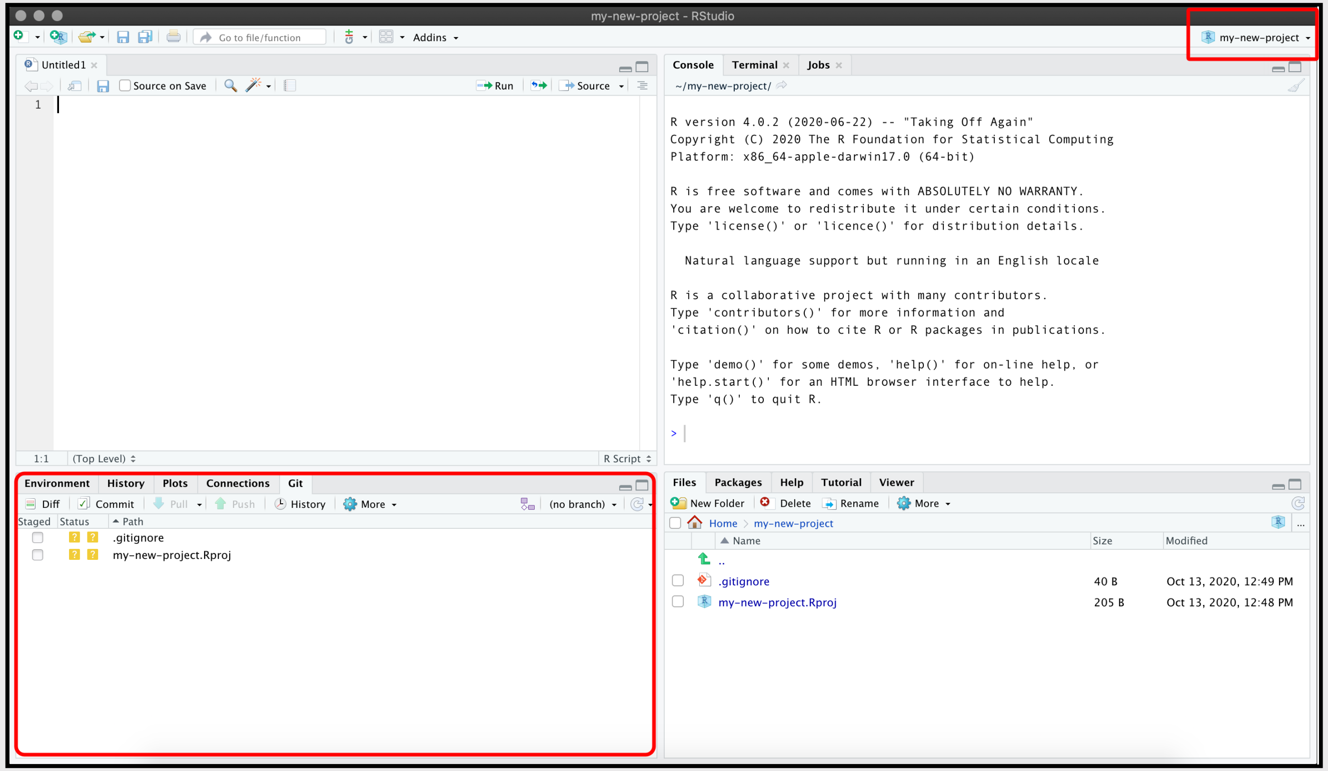
☑ Create a git repository

□ Use renv with this project

□ Open in new session

Creating RStudio projects in new folder

Check for new project name & *Git* pane



```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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.

>
```

Staged	Status	Path
□	? ?	.gitignore
□	? ?	my-new-project.Rproj

Name	Size	Modified
..		
.gitignore	40 B	Oct 13, 2020, 12:49 PM
my-new-project.Rproj	205 B	Oct 13, 2020, 12:48 PM

Folder Structure

- separate raw and cleaned data
- keep documents and code separate
- keep figures separate
- name files appropriately (preferably 2 digit prefix)
- structure is reusable and easy to understand

```
project-name/
    |-- CITATION
    |-- project-name.Rproj
    |-- README.md
    |-- LICENSE
    |-- requirements.txt
    |--data/
       |--raw/
            |--raw-birds-data.csv
       |--processed/
            |--processed-birds-data.csv
    |--doc/
        |-- notebook.Rmd
        |-- manuscript.Rmd
        |-- changelog.txt
    |--results/
        |-- summarized-results.csv
    |--code/
        |-- 01-sightings-import.R
        |-- 02-sightings-wrangle.R
        |-- 03-sightings-model.R
        |-- runall.R
```

Folder structure

Adapted from from 'Good enough practices in scientific computing'

Naming things

Naming files

File names should be:

1. **human readable -> (makes sense)**

2. **machine readable -> (regex)**

3. **sort/order well -> (ISO 8601 date)**

2020-10-12-270-301- **central-lab-metrics**.csv

2020-10-12- **270-301**-central-lab-metrics.csv

2020-10-12-270-301-central-lab-metrics.csv

Naming files*

We can perform regular expression searches for files like this:

Find 270-301 files

```
grepl(pattern = "270-301",
      x = "2020-10-12-270-301-central-lab-metrics.csv")
```

```
 [1] TRUE
```

*Adapted from Jenny Byran's slides

Naming files*

Also acceptable: Logical order and underscores _

```
files
```

```
 [1] "01.0_import_lab-data.R"  "02.0_wrangle_lab-data.R"
 [3] "03.0_eda_lab-data.R"
```

```
stringr::str_split_fixed(string = files, pattern = "_", 3)
```

```
      [,1]   [,2]      [,3]
 [1,] "01.0" "import"  "lab-data.R"
 [2,] "02.0" "wrangle" "lab-data.R"
 [3,] "03.0" "eda"     "lab-data.R"
```

*Adapted from Jenny Byran's slides

File paths

Use relative rather than absolute file paths

Absolute paths are specific to a system

`/project-name/data` → absolute path in macOS

`\\project-name\\data` → absolute path in Windows

Relative paths are specific to a folder

`project-name/data` → relative path in macOS

`project-name\\data` → relative path in Windows

Or use the `here` package

The `here::set_here()` function solves a lot of file path problems (*especially if you're not using R projects*)

```r
library(here)
here::set_here(".")
list.files(all.files = TRUE, pattern = "here")
```

```
[1] ".here"
```

This creates a `.here` file (similar to `.Rproj` files)

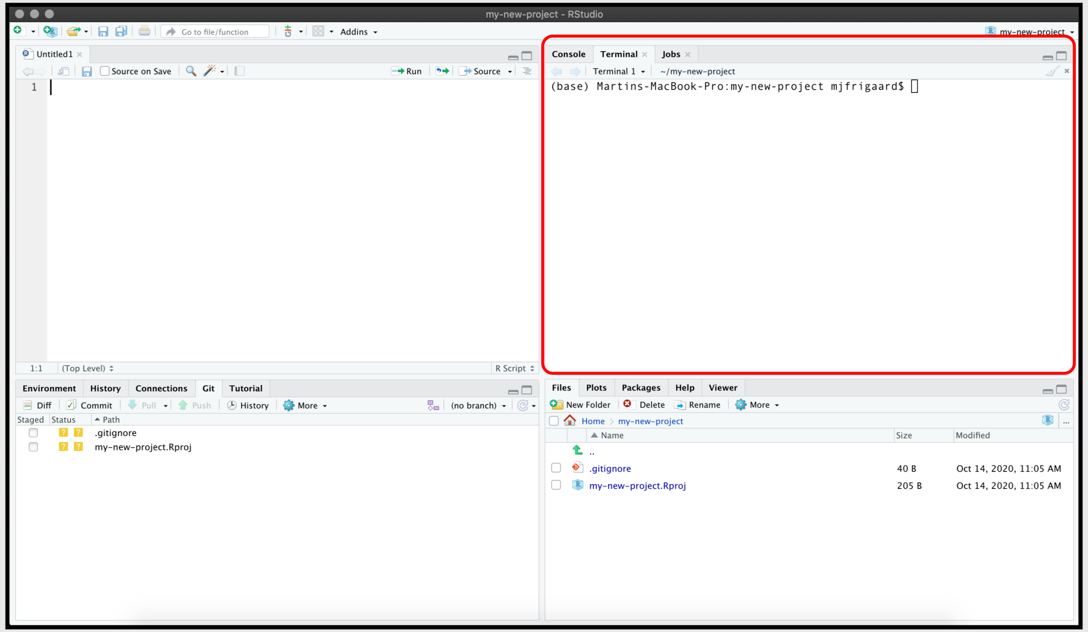
As long as the `.here` file stays in the referenced folder, you can include simply include `here::here()` in the top of your code files.

Terminal pane

Learn a handful of command-line tools to make life easier

`cd`, `pwd`, `mkdir`, `rm`, `ls`, etc.

RStudio comes with a Terminal pane for quick access to the command-line



```
(base) Martins-MacBook-Pro:my-new-project mjfrigaard$ 
```

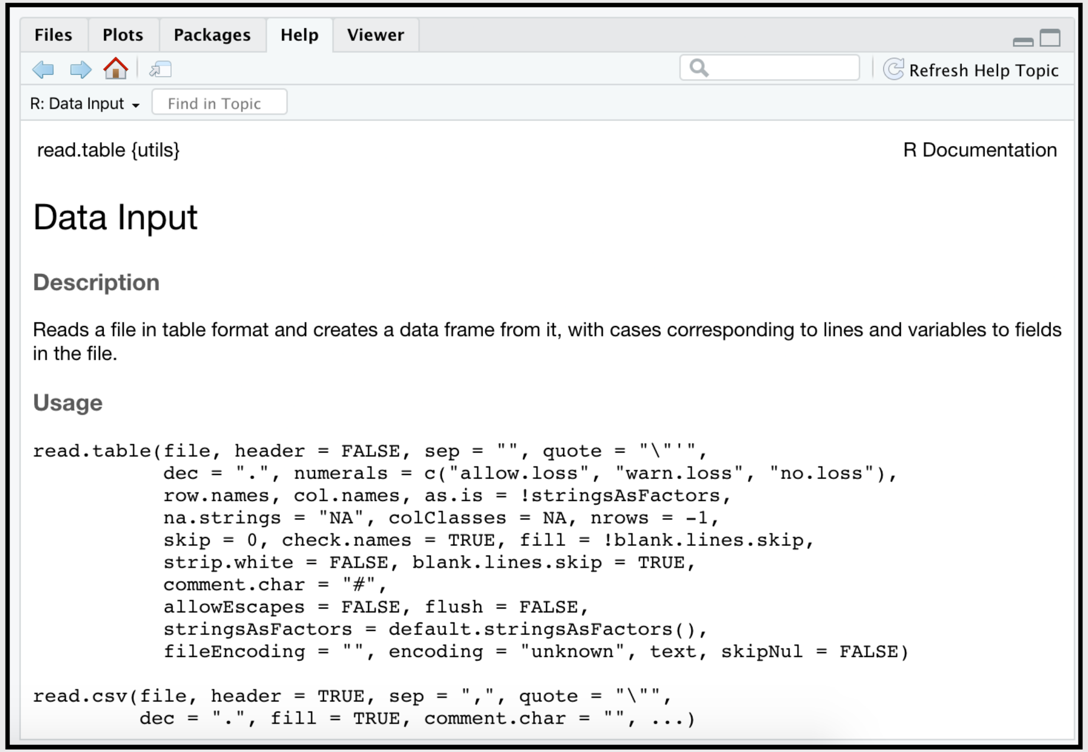
Staged	Status	Path
□	? ?	.gitignore
□	? ?	my-new-project.Rproj

	Name	Size	Modified
	..		
□	.gitignore	40 B	Oct 14, 2020, 11:05 AM
□	my-new-project.Rproj	205 B	Oct 14, 2020, 11:05 AM

Getting help

R comes with a *ton* of accessible help files

```
?read.csv
```



read.table {utils}

R Documentation

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Getting help online

R also help an incredible community! Click on the links below to see some of the common places for Q & A.

1) Dedicated forum on RStudio Community

2) Questions tagged R on StackOverflow

3) Twitter topics with #rstats hashtag

Asking good questions (reproducible examples)

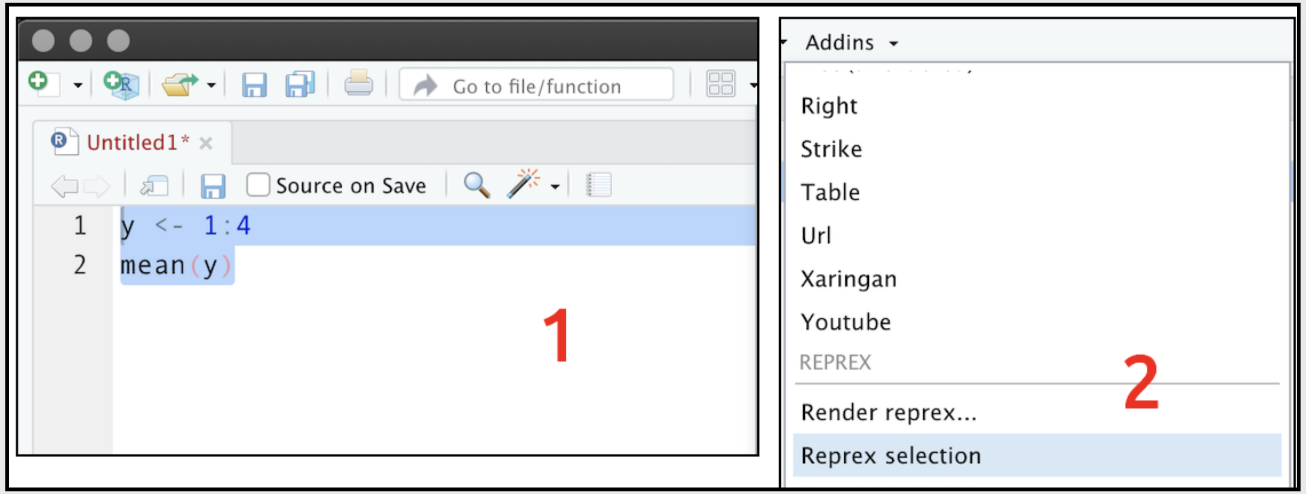
You'll get better results if you ask a question with a reproducible example. The `reprex` package was designed to help you create one!

```r
install.packages("reprex")
library(reprex)
```

Use the RStudio Addin to create a reproducible example from code you've copied onto your clipboard!

Reprex Addin 1

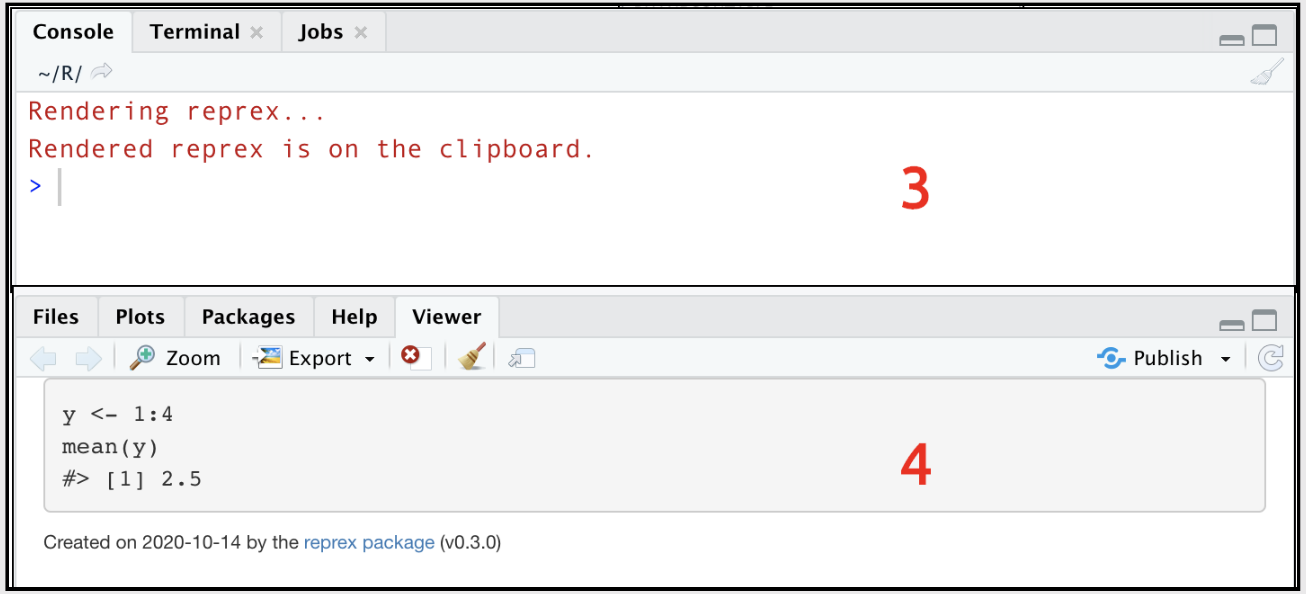
1. Copy code
2. Select Addin > Render selection



```r
y <- 1:4
mean(y)
```

Reprex Addin 2

1. Copy code
2. Select Addin > Render selection
3. Wait for console
4. Paste reprex



```
Rendering reprex...
Rendered reprex is on the clipboard.
>
```

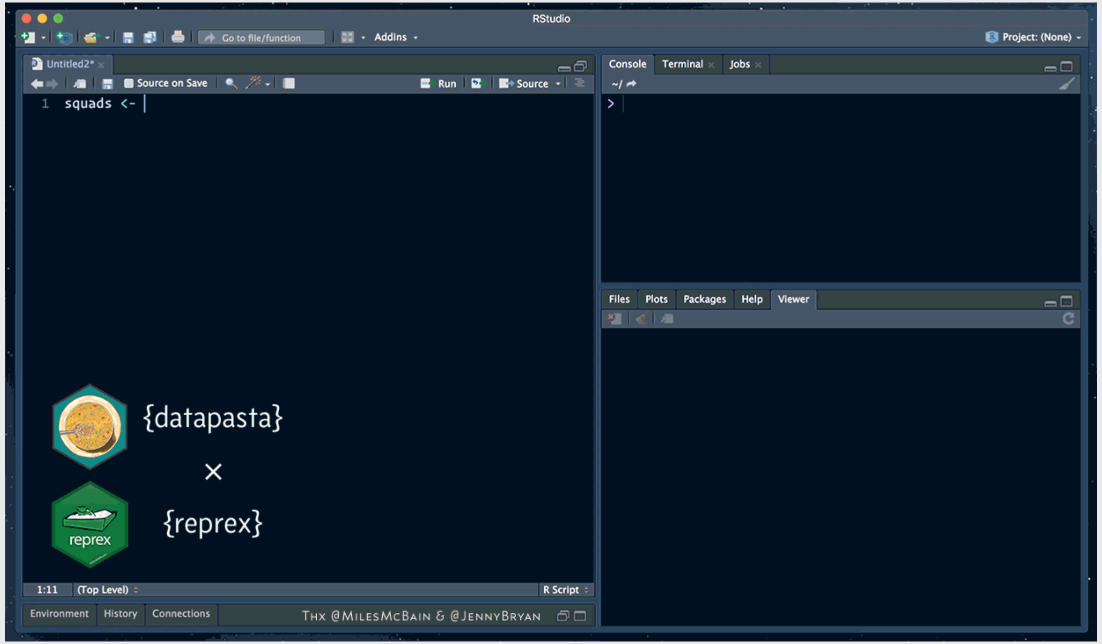
```
y <- 1:4
mean(y)
#> [1] 2.5
```

Created on 2020-10-14 by the reprex package (v0.3.0)

Reprex + datapasta

To copy + paste actual data in a reproducible example, try `datapasta`!

https://reprex.tidyverse.org/articles/articles/datapasta-reprex.html



```
squads <- 
```

Learn more about R best practices:

1. R for Data Science
2. Tidyverse
3. RViews Community Blog

THANK YOU!

Feedback

@mjfrigaard on Twitter and Github

mjfrigaard@gmail.com