

# Intro to RStudio



Angela He

Stanford Software and Services for Data Science

# What are R and RStudio?

- R is a programming language
- We will be using RStudio with R, which is a user interface for working with R



# Why R and RStudio?

- Compared to Stata:
  - Free and open-source
    - Anyone can open and edit files you share
      - Vs. Stata: you need Stata to open and edit Stata files
    - Lots of cool packages that people create and share regularly
  - R can hold multiple datasets at a time
  - Steeper learning curve but used more widely than Stata and will help you learn other programming languages
- Compared to Python:
  - More gradual learning curve, good first introduction to programming
  - Easier to use for data analysis

# Set-up Steps

1. Create a new project
2. Create a markdown file
3. Install/load the packages you need (you can add more later too)
4. Read in your dataset(s)
5. Now you can begin your data cleaning and analysis!

# 1. Create a new project

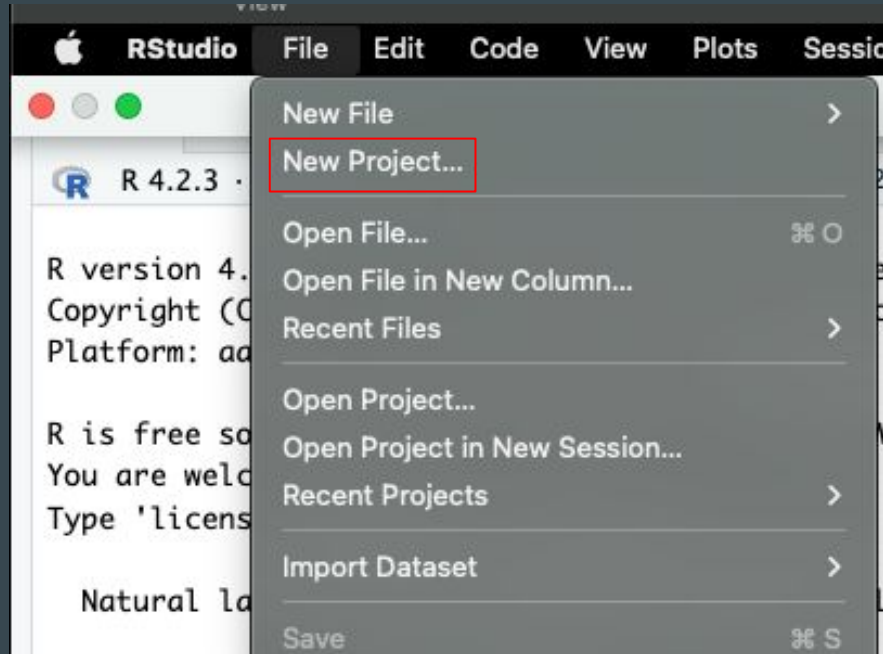


Why use a project file?

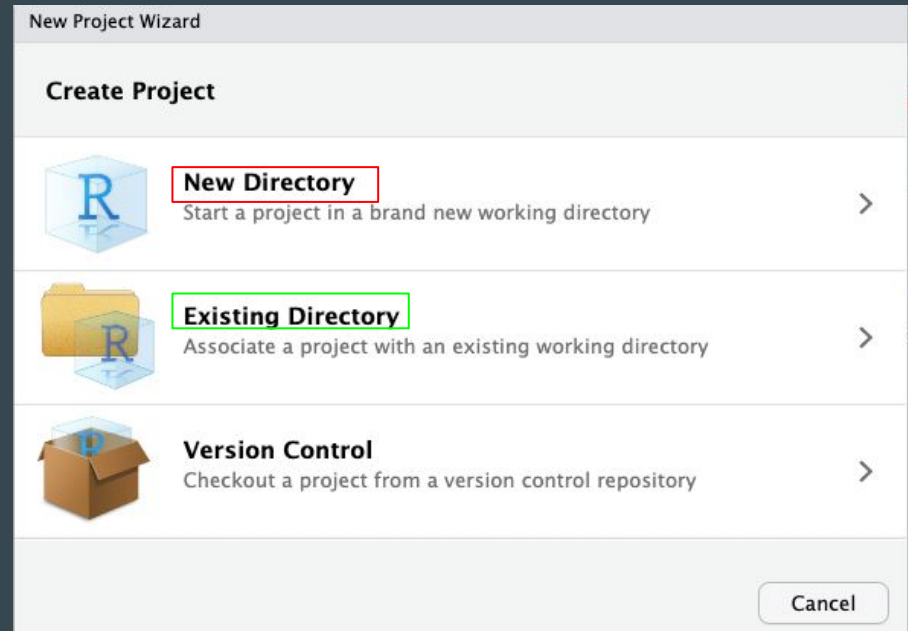
- All of your work associated with the project (e.g. data, scripts) are stored in the same portable folder
- You won't need to worry about configuring your working directory no matter which computer you are on (helps with reproducibility too!)

# File > New Project

- 



- You can choose New Directory or Existing Directory
- Choose **New Directory** if you want to create a new folder to put your project in
- Choose **Existing Directory** if you have an existing folder you want to put your project in



## New Project Wizard

Back

### Create New Project



Directory name:

SSDS Intro to R Workshop Demo

Create project as subdirectory of:

/Users/angelahe/Library/CloudStorage/Dropbox

Browse...

- ☒ Create a git repository
- ☐ Use renv with this project

☒ Open in new session

Create Project

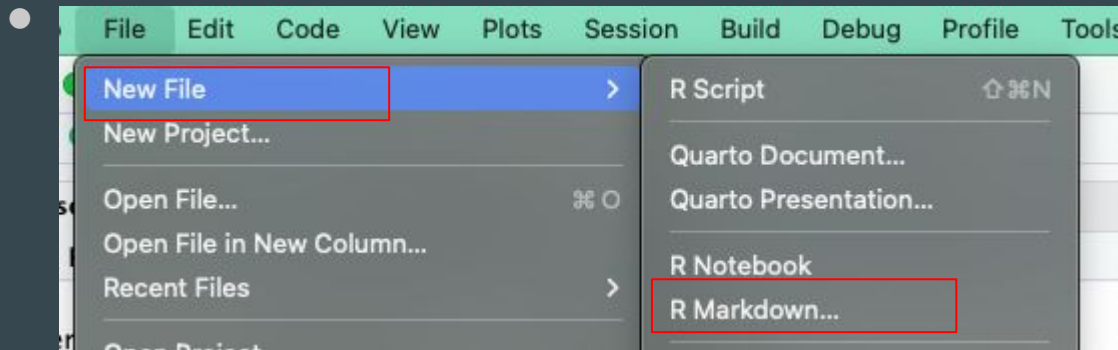
Cancel



## 2. Create a new markdown file


...


# File > New File > R Markdown





# You can add a title and author to your markdown file

New R Markdown

 Document

 Presentation

 Shiny

 From Template

**Title:** preliminary analysis

**Author:**

**Date:** 2024-01-25

☒ Use current date when rendering document

**Default Output Format:**

☒ HTML  
Recommended format for authoring (you can switch to PDF or Word output anytime).

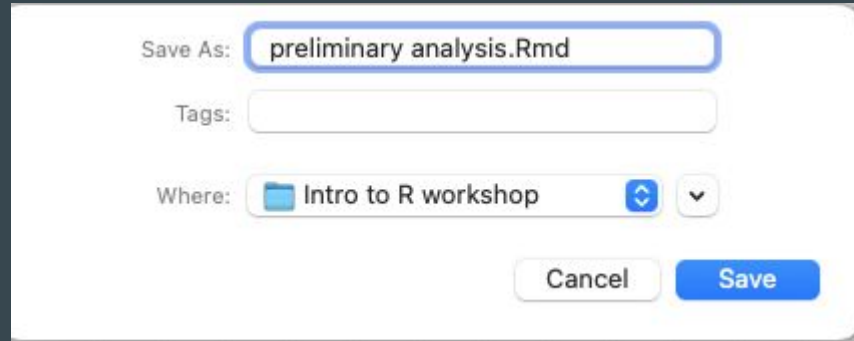
☐ PDF  
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

☐ Word  
Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

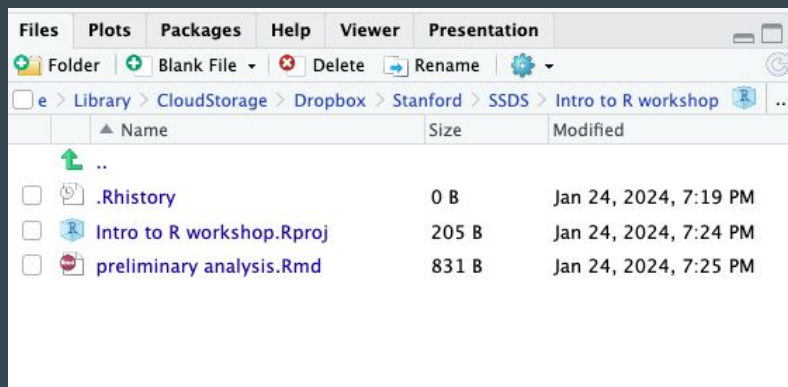
Create Empty Document

OK Cancel

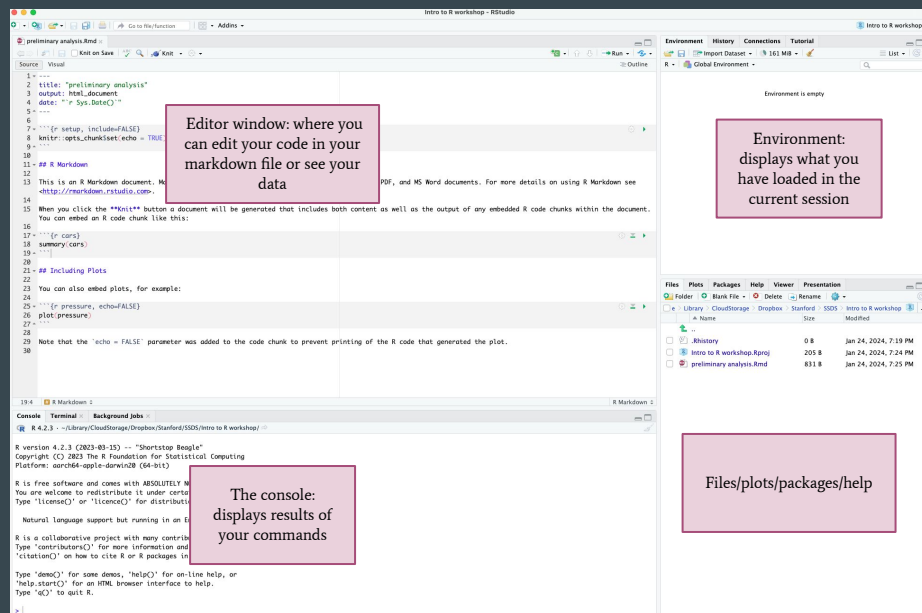
After the new markdown file is created, make sure you press ctrl + s to save the file to your project folder



# You should now see your markdown file in the bottom right quadrant

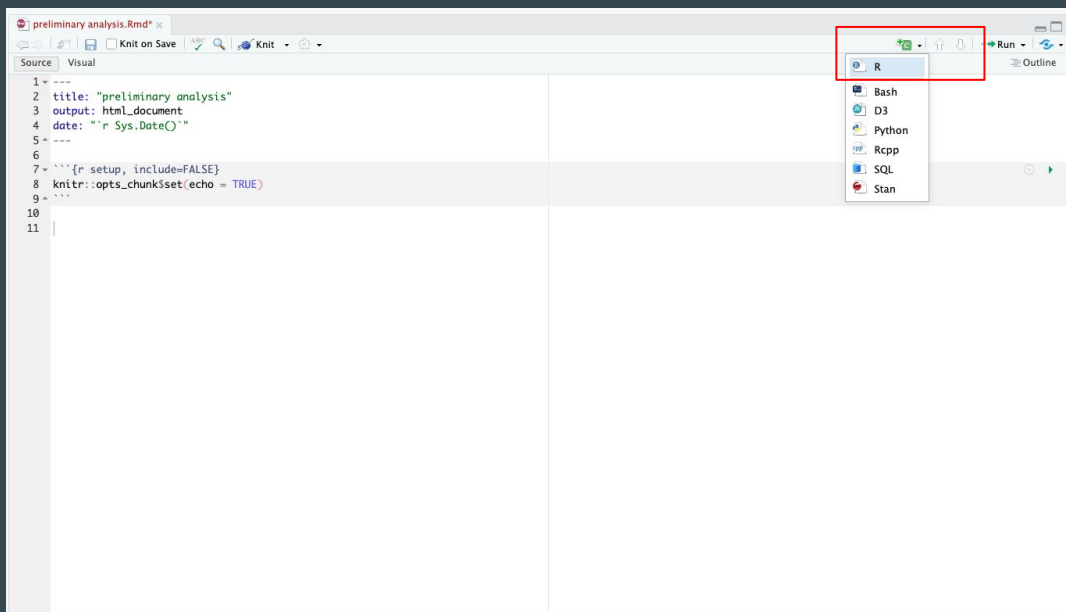


# The four quadrants in RStudio



# How to write code in your markdown file

- You can organize your code into “chunks”
- To create a new chunk:

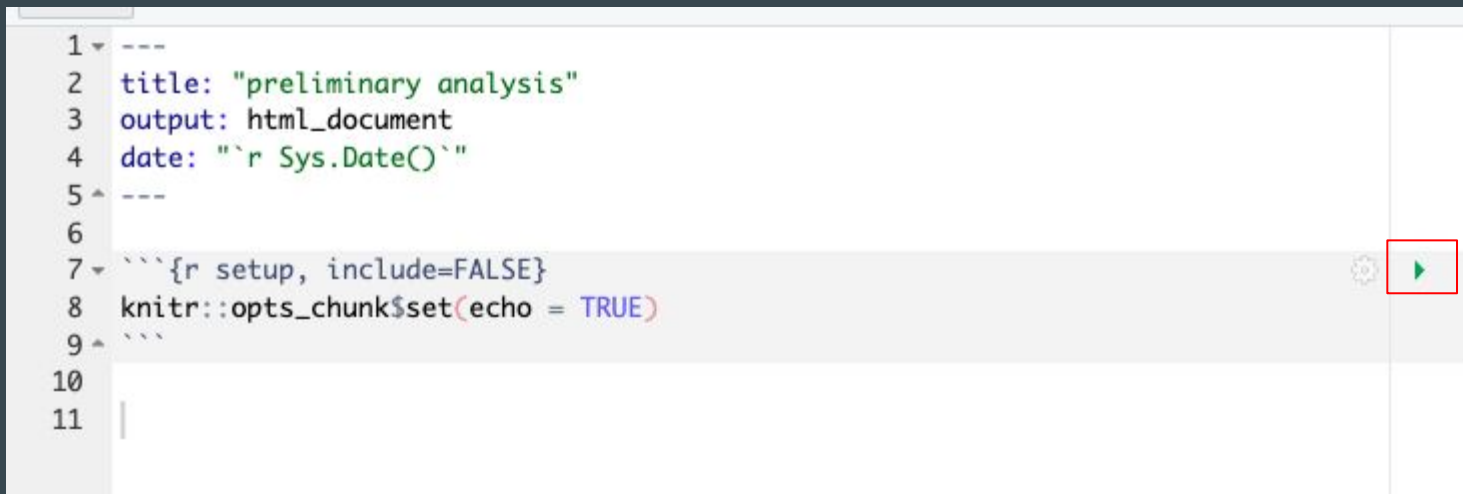


# How to run code

- Press the green play button

OR

- Highlight code and “ctrl + enter” to run that line of code



The screenshot shows a code editor with the following R code:

```
1 ---  
2 title: "preliminary analysis"  
3 output: html_document  
4 date: "`r Sys.Date()`"  
5 ---  
6  
7 ```{r setup, include=FALSE}  
8 knitr::opts_chunk$set(echo = TRUE)  
9 ```  
10  
11 |
```

The code chunk starting at line 7 is highlighted. On the right side of the editor, there is a gear icon and a green play button icon, which is highlighted with a red square, indicating the button used to run the code.



# How to write comments

- Put a **#** before your comment → will make that text green
- Anything outside the chunks will be treated as text too

```
```{r}
#load libraries
library(tidyverse)
library(janitor)
```
```

**3. Install/load packages you need**  
**...**

# Packages

- Packages are essentially extensions (think of Chrome extensions) you can add to your RStudio
  - Make your life easier: you don't need to write code yourself for tasks like predictive modeling (carat) or cleaning variable names (janitor)
- To install a package:
  - Type and run `install.packages("packagename")` either into your console or R markdown file
  - You just need to do this once, which is when you first download the package onto your computer
- To load the package:
  - Type `library(packagename)`
    - No quotes!
  - Do this every time you want to use a specific package for your project

# Some packages I (almost) always use

- tidyverse
  - “A coherent system of packages for data manipulation, exploration, and visualization that share a common design philosophy”
  - Includes the following packages: dplyr (data manipulation), ggplot2, (data visualization), ...
- janitor
  - Cleans variable names
- Stargazer
  - Pretty(ish) regression tables

## 4. Read in your dataset(s)

...

There are many different types of files you can read in. I like to read in csv files because they are universal.

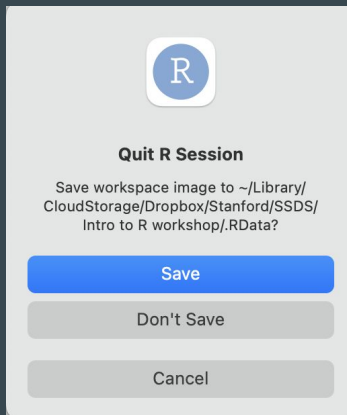
# Let's switch to the R markdown file now to dive into the code!

...

Download the .zip file here: <http://tinyurl.com/ssds-intro-to-r>

# Helpful Tips and Tricks

- Enable rainbow parentheses to improve readability
- To export your file to share with someone else
- Click “don’t save” when R asks you if you want to save your workspace image when you exit out of R.



# Resources

- [The ultimate guide: R for Data Science](#)
  - Written by the creator of Tidyverse!
- Help files! (? or ??)
- [Book an appointment with SSDS](#)