Introduction and R Basics

STA 032: Gateway to data science Lecture 1

Jingwei Xiong

April 3, 2023

Summary

- Course overview
- Course logistics
- R and R studio
- Some examples

Statistical data science

- Statistics is the study of how to collect, analyze, and draw conclusions from data.
- Data science is typically thought of as an interdisciplinary field, combining statistical thinking with elements more traditionally thought of as coming from other fields, such as programming, database management and optimization.
- There is a stronger focus on the practical aspects of working with data, in particular computing, as well as applications in different domains, such as the sciences, business, sports, and government.
- This is a course on introduction to data science, no prior programming experience needed. However, we will focus on R programming.

Examples of Data Science in Practice

- Predicting Grades: Using past grades and study habits to predict future academic performance.
- Analyzing Sports Data: Analyzing data from sports games to identify patterns and make predictions, such as predicting the outcome of a game or identifying which player is likely to make the next play.
- Social Media Analysis: Analyzing social media data to understand user behavior and preferences, such as identifying popular topics or predicting which products or services are likely to be successful.
- Customer Segmentation: Grouping customers based on shared characteristics, such as demographics, buying habits, or preferences, to create targeted marketing campaigns.

These examples demonstrate how data science can be applied to a variety of domains, from academia to sports to marketing to security.

Data Science Workflow

Data Science Workflow refers to the step-by-step process of data analysis, from data collection to communicating insights. The workflow is iterative, meaning that each step builds on the previous one, and changes may be made at any point based on new discoveries or insights.

Steps in the Data Science Workflow

1. **Data Collection**: Collecting relevant data from various sources.
2. **Data Cleaning and Preparation**: Cleaning and preparing the data for analysis by handling missing data, outliers, and other issues.
3. **Data Exploration and Visualization**: Exploring the data visually to identify patterns, trends, and relationships.
4. **Data Analysis and Modeling**: Analyzing the data to generate insights and develop models that can predict outcomes or identify trends.
5. **Communication of Results**: Communicating the insights and findings to stakeholders through reports, visualizations, and presentations.

Course content

1. Fundamentals of R
    - Overview of data types and structures
    - Data manipulation and data visualization tools
    - Functions, iterations
    - R simulation
2. Descriptive statistics for numerical and categorical data
3. Probability
    - Rules of probability computation; conditional probability
    - Basic probability models: Binomial, Normal and Poisson
4. Statistical inference
    - Sampling distributions of sample mean and sample proportion
    - Hypothesis testing and confidence intervals for population mean and population proportion
    - Simple linear regression

- No statistics, data science or programming knowledge presumed
- R will be used throughout this course.

Course logistics

- Lectures Monday, Wednesday and Friday
- Thursday lab
- Office hours
  - TA: TBD
  - Jingwei Xiong: TBD
- Course website: https://xjw1001001.github.io/
  - Lecture notes, homework, supplementary materials, etc.
- Canvas for lab materials, turning homework (through Gradescope), solutions and grade-book
- Piazza for announcements and discussion

Grading

- Grade Distribution:

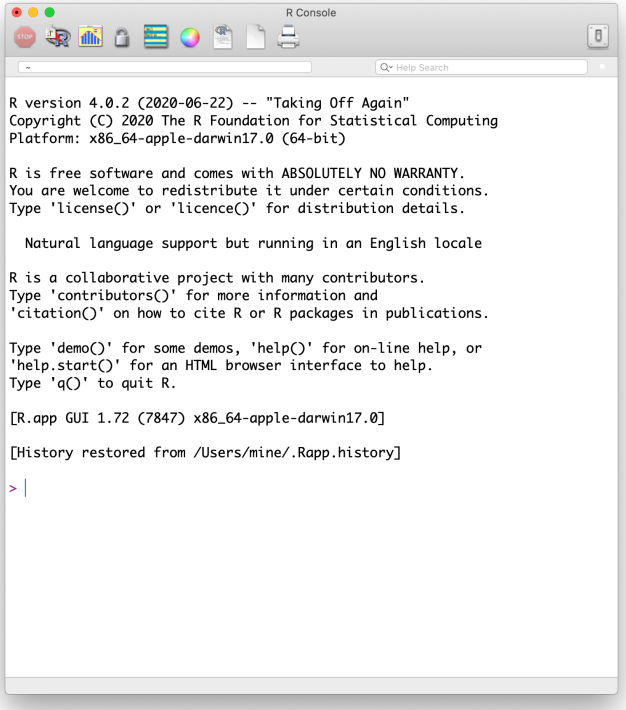
Assignment	Percentage
Homework Level 1	35%
Homework Level 2	5+%
Participation	5%
Exam I	15%
Exam II	20%
Final	25%

- Homework: Released on Fridays, due on Thursdays at 12 PM. Use R markdown to generate the homework and submit in PDF format on Gradescope. Level 2 homework problems are available for extra credit.
- Participation:Participation and L2 homeworks will be evaluated with the higher one.
- Exams: Exam 1 is a take-home exam on R programming, Exam 2 is an in-person exam on statistical concepts, and the Final Exam is a take-home programming exam.

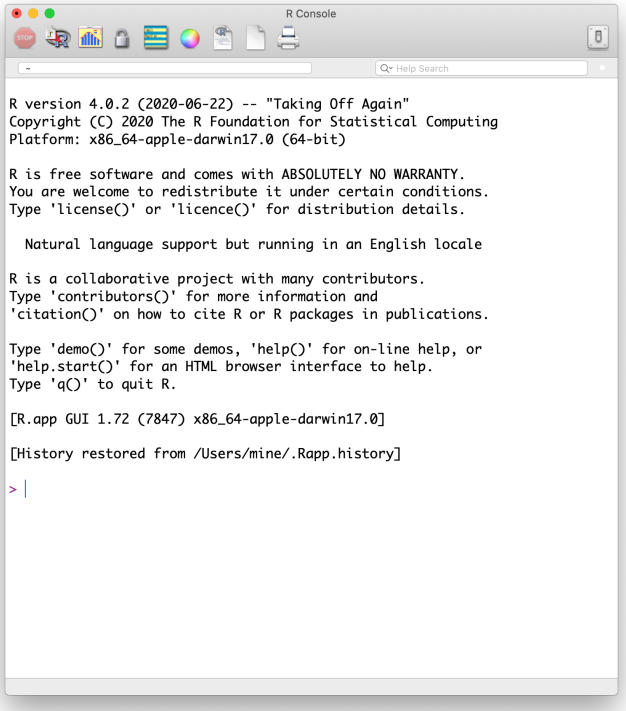
Set up

- You will need regular, reliable access to a computer either with a working browser, or running an up-to-date version of R and RStudio
- It's strongly recommended to install R and RStudio on your own computers
- If this is a problem, please let us know right away. There are resources available to support you.
- Labs will be at TLC 2212; either use computers available in the lab, or your own laptops (make sure your laptop is charged before class)
- For lectures, it's strongly recommended to repeat all coding examples on your own computer.

Software: R



- R is a free, open-source statistical programming language for statistical computing
- It is also an interactive environment for doing data science
- Data science teams often use a mix of languages, including R, Python, Julia, ...



- R Console: Basic interaction with R is by typing in the console, a.k.a. terminal or command-line
- You type in commands, R gives back answers (or errors)
- It is easily extensible with packages
- Menus and other graphical interfaces are extras built on top of the console

Installing R

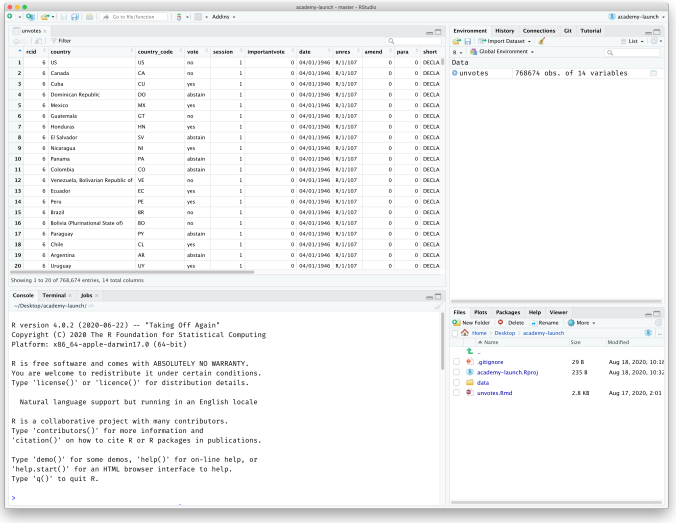
To install R on Windows OS:

1. Go to the **CRAN** website.
2. Click on **"Download R for Windows"**.
3. Click on "install R for the first time" link to download the R executable (.exe) file.
4. Run the R executable file to start installation, and allow the app to make changes to your device.
5. Follow the installation instructions.

To install R on Mac OS:

1. Go to the **CRAN** website.
2. Click on **"Download R for macOS"**.
3. Download the latest version of the R GUI under (.pkg file) under **"Latest release"**. You can download much older versions by following the "old directory" or "CRAN archive" links.
4. Run the .pkg file, and follow the installation instructions.

R studio



- RStudio is a free, open-source R programming environment
- It is called an integrated development environment, or IDE, for R programming
- It contains a built-in code editor, many features to make working with R easier, and works the same way across different operating systems.

Installing RStudio Desktop

To install RStudio Desktop on your computer, do the following:

1. Go to the **RStudio** website.
2. Go to the step 2, Install RStudio Desktop
3. Download RStudio Desktop recommended for your computer.
4. Run the RStudio Executable file (.exe) for Windows OS or the Apple Image Disk file (.dmg) for macOS X.

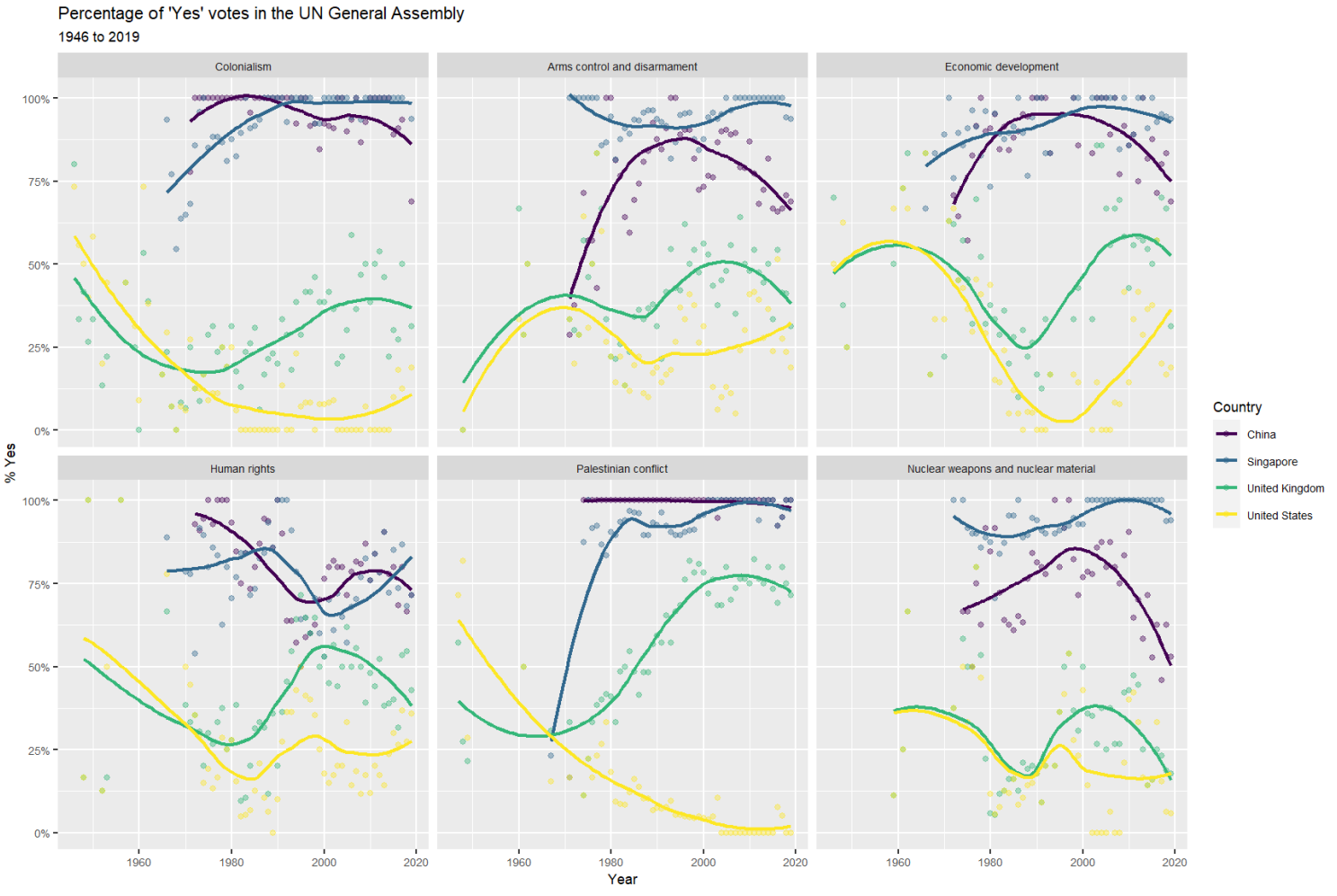
Example of a data visualization

R Code | Plot

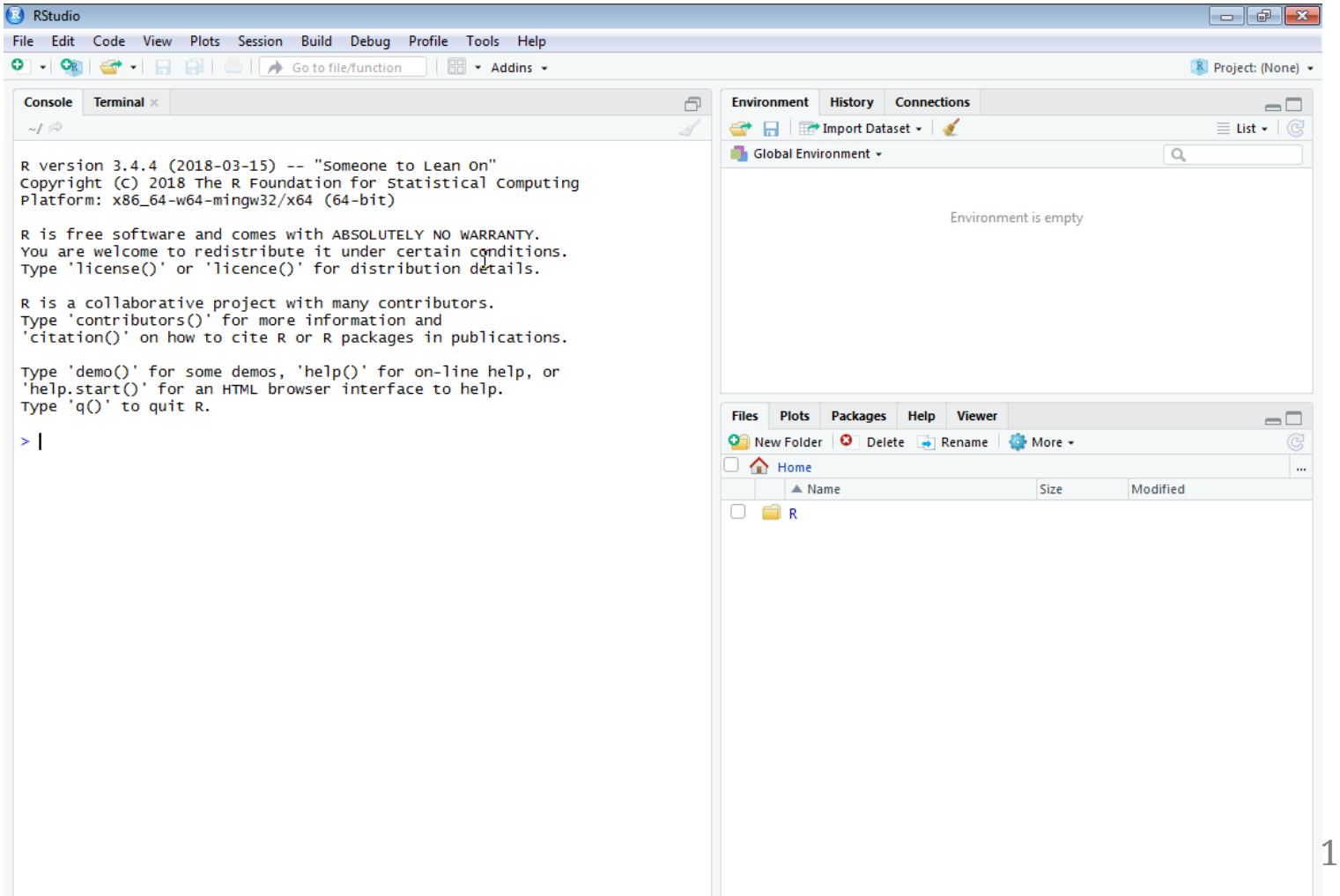
```
un_votes %>%
  filter(country %in% c("United States", "United Kingdom", "China", '
  inner_join(un_roll_calls, by = "rcid") %>%
  inner_join(un_roll_call_issues, by = "rcid") %>%
  mutate(year = lubridate::year(date)) %>%
  group_by(country, year, issue) %>%
  summarize(votes = n(),
            percent_yes = mean(vote == "yes")) %>%
  filter(votes > 5) %>%  # Only use records where there are more thar
  ggplot(mapping = aes(x = year, y = percent_yes, color = country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = scales::percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembly",
    subtitle = "1946 to 2019",
    y = "% Yes",
    x = "Year",
    color = "Country"
  ) +
  scale_color_viridis_d() +
  theme(text = element_text(size = 9))
```

Example of a data visualization

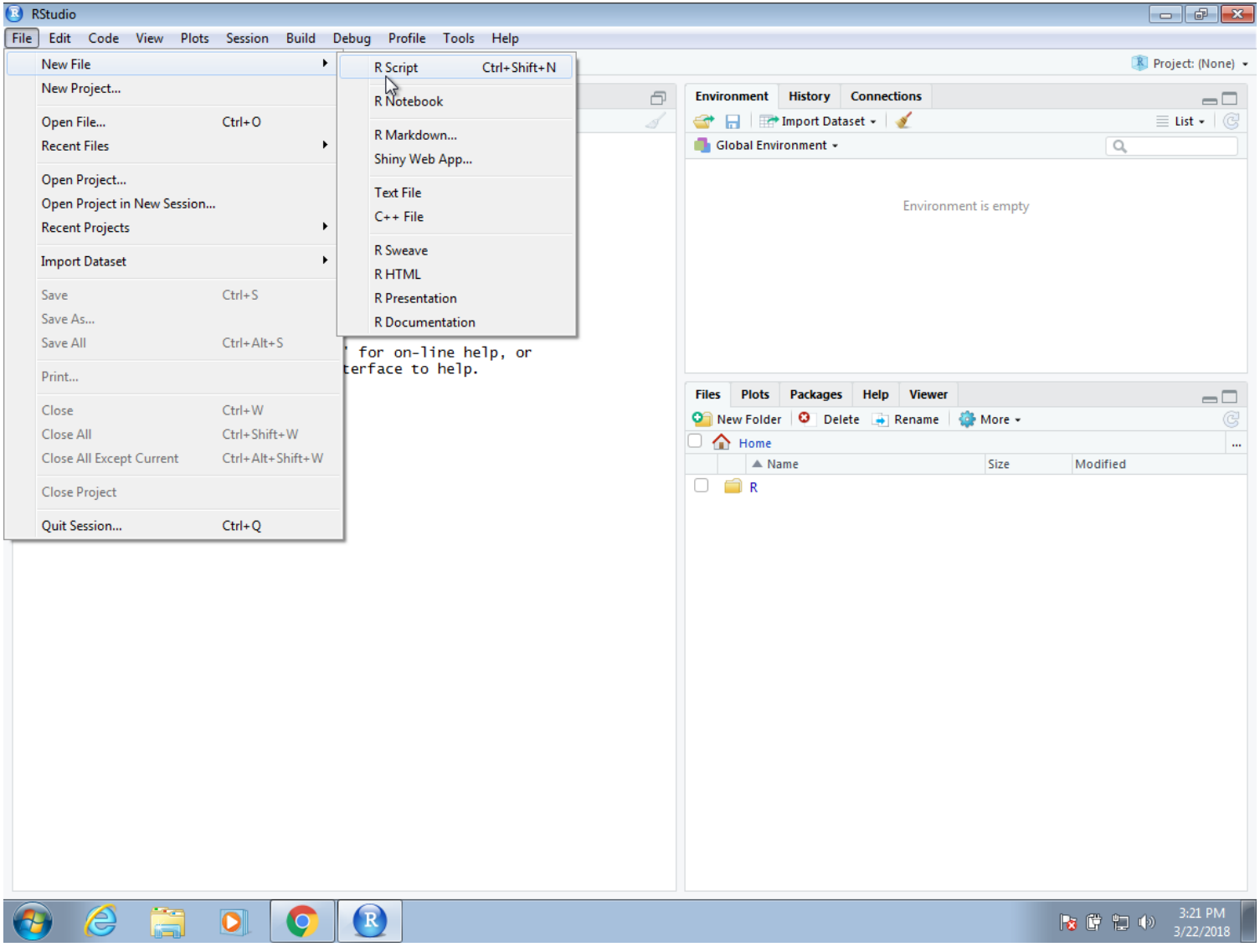
R Code | Plot



When you start RStudio for the first time, you will see three panes. The left pane shows the R console. On the right, the top pane includes tabs such as Environment and History, while the bottom pane shows five tabs: File, Plots, Packages, Help, and Viewer (these tabs may change in new versions). You can click on each tab to move across the different features.



To start a new script, you can click on File, then New File, then R markdown.

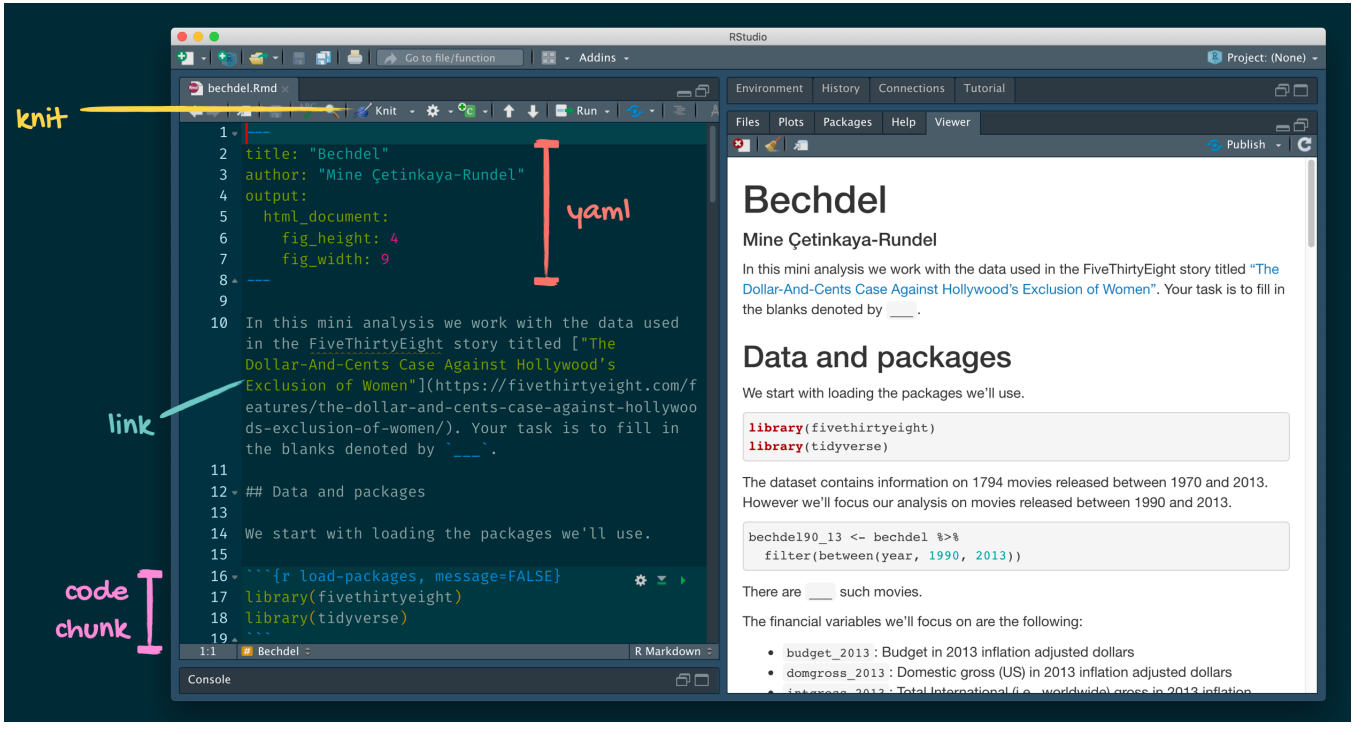


This starts a new pane on the left and it is here where you can start writing your R markdown file. The R markdown file will ended with **.Rmd**.

R Markdown

- R Markdown is a tool to integrate code and written prose in reproducible computational documents
- R Markdown files have the `Rmd` extension. Each time you "knit," the analysis is run from the beginning.
- To learn more, go to rmarkdown.rstudio.com
- Homework and take home exams will be completed in R Markdown
- Code goes in chunks, defined by three backticks, narrative goes outside of chunks

Tour: R Markdown



- Example: Create your first R markdown file and knit it.

In RStudio, you can start an R markdown document by clicking on **File, New File, the R Markdown**.

You will then be asked to enter a title and author for your document.

You can also decide what format you would like the final report to be in: HTML, PDF, or Microsoft Word.

It will generate a template file.

As a convention, we use the **Rmd suffix** for these files.

In the template, you will see several things to note.

The header

At the top you see:

```
---
title: "Untitled"
author: 'Jingwei Xiong'
date: "2023/1/10"
output: html_document
---
```

The things between the `---` is the header. We actually don't need a header, but it is often useful. You can define many other things in the header than what is included in the template. We don't discuss those here, but much information is available online. The one parameter that we will highlight is `output`. By changing this to, say, `pdf_document`, we can control the type of output that is produced when we compile.

R code chunks

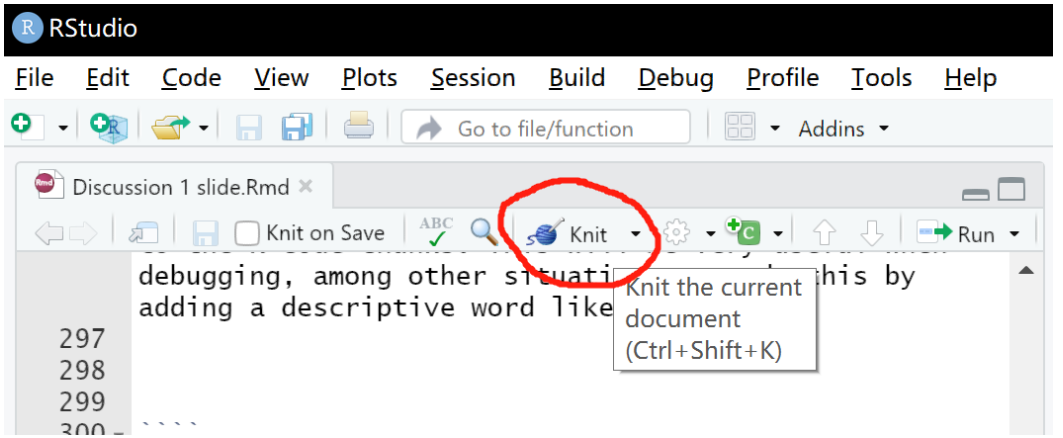
In various places in the document, we see something like this:

```
```{r}
summary(pressure)
```
```

These are the code chunks. When you compile the document, the R code inside the chunk, in this case `summary(pressure)`, will be evaluated and the result included in that position in the final document.

This applies to plots as well; the plot will be placed in that position. We can write something like this:

```
```{r}
plot(pressure)
```
```

By default, the code will show up as well. To avoid having the code show up, you can use an argument. To avoid this, you can use the argument `echo=FALSE`. For example:

````
```{r, echo=FALSE}
summary(pressure)
```
````

If you want to only show the code but not run the code, you can use the argument `eval=FALSE`.

````
```{r, eval=FALSE}
summary(pressure)
```
````

By default, the code will run and the output will be shown.

Knit your first rmd file

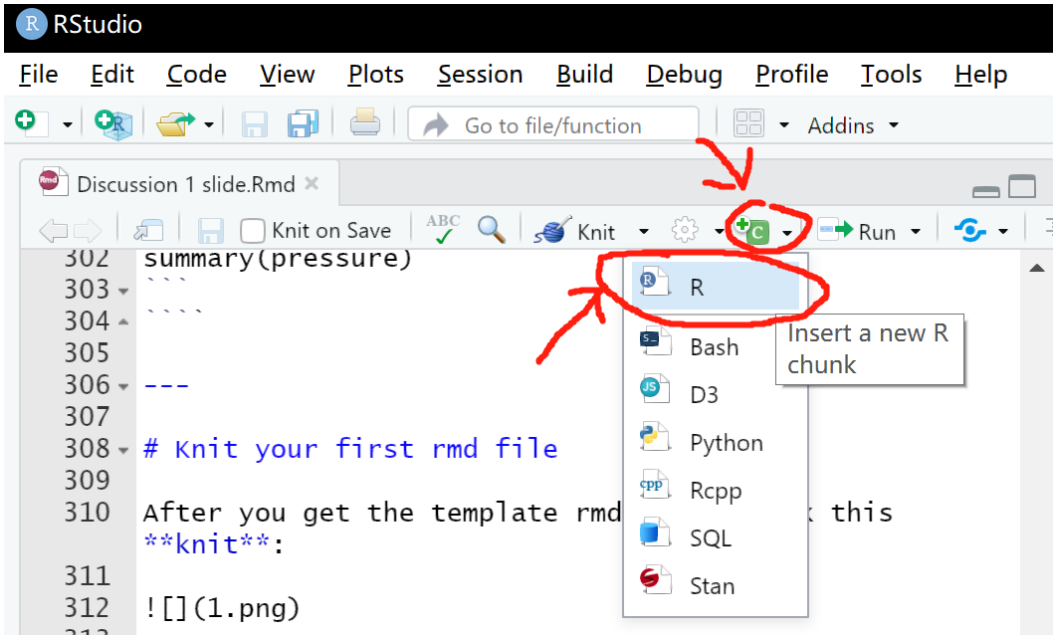
After you get the template rmd file, click this **knit**:



This button will process your source code into the final document, if your code has no error.

Insert a new code chunk

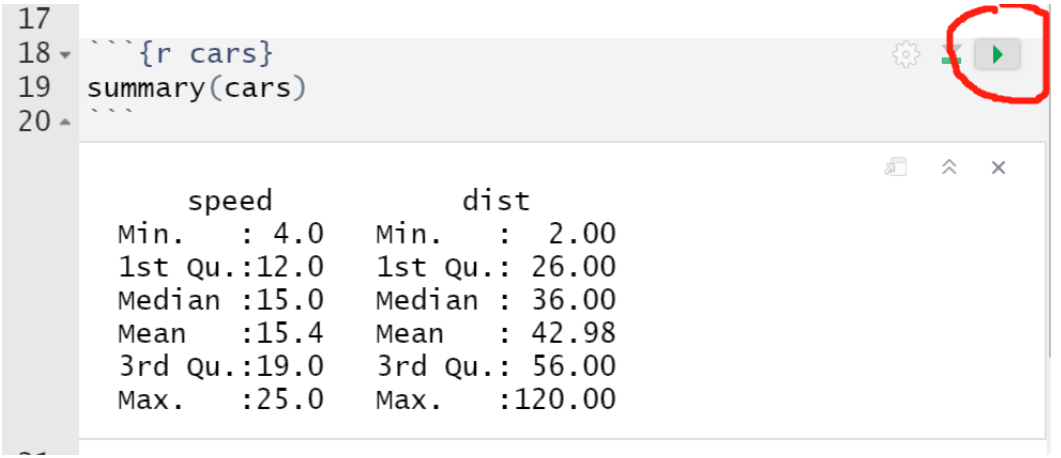
To insert a new code chunk, click this:



This button will insert a new code chunk in the current cursor line.

Run scripts in the code chunk

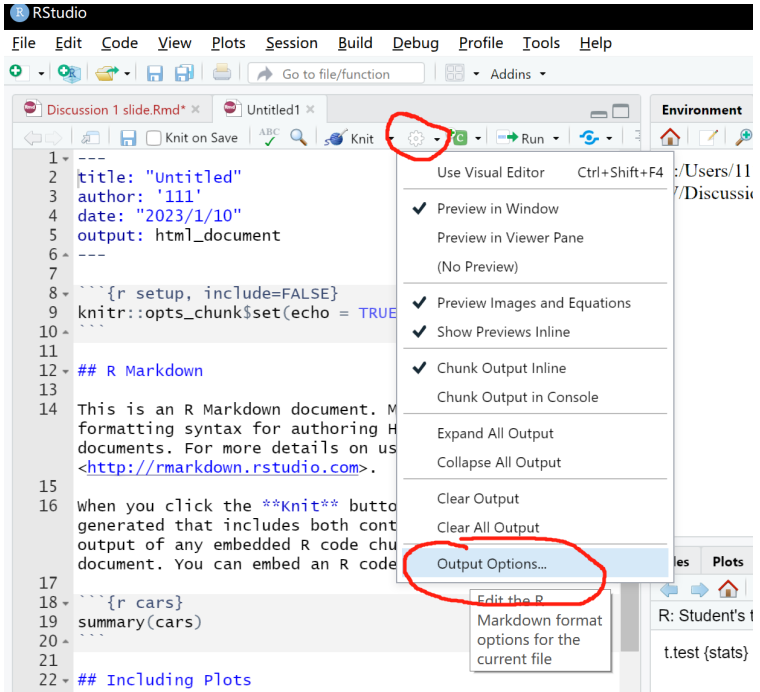
To run scripts in a code chunk, click this:



This button will copy all of the codes inside that code chunk into the console, and run it.

Knit settings

You can find the knit settings here:



You can change to word using the output format.

Try it yourself

1. Go to File -> New file -> R markdown to generate a new R markdown file. Change the title into "My project" and author as your name. Select the html document.

2. Save your rmd file into a place you know with a new name.

3. Knit the project as a html file. You should find the html file in the same location with the rmd file you saved.

4. Directly change the header in the rmd file into

```
title: "My first R markdown"
author: 'Jingwei Xiong'
date: "2023-03-22"
output: pdf_document
```

Check list

- Install R and R studio on your own computer.
- Try it yourself, knit your first rmd file.

Reading:

Getting started with R and RStudio

R markdown