POLI 30 D: Political Inquiry

TA Sessions

Lab 04 | R Markdown, Packages, and more

Before we start

Announcements:

- Github page: **https://github.com/umbertomig/POLI30Dpublic**
- Piazza forum: For some reason the link in the slides does not work well. Check with TAs / Professor for alternative link.

Before we start

Recap: In the Lab sessions, you learned:

- How to install R and R Studio on your computer.
- How to do basic math operations in R.
- How to do basic vector and data.frame operations in R.

Great job!

- Do you have any questions about these contents?

Plan for Lab 04

- Learn R Markdown
- Install Packages

R Markdown

R Markdown

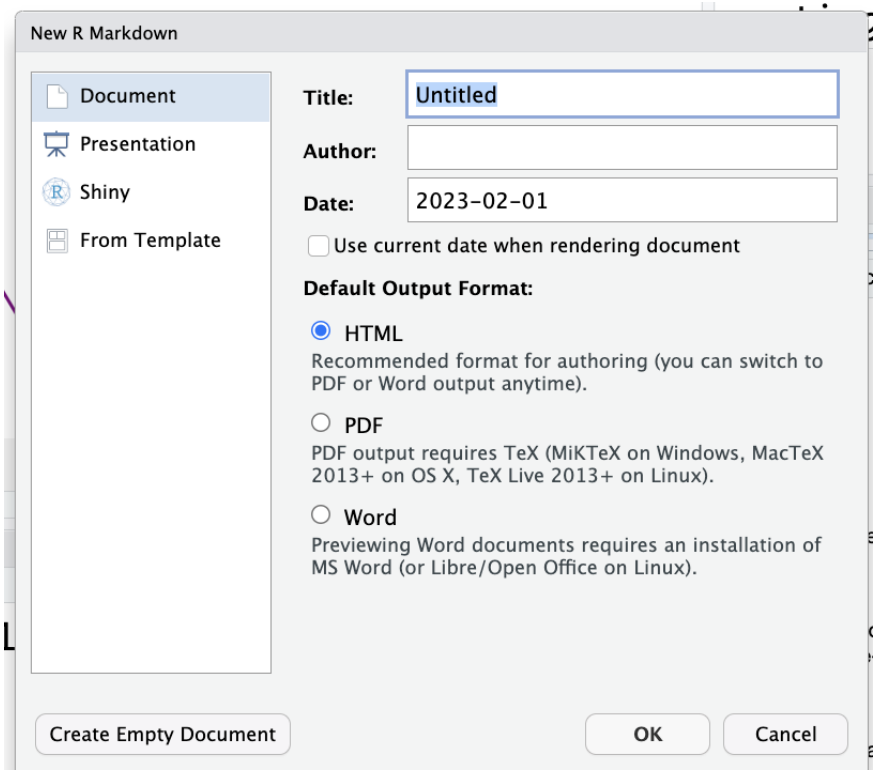
- R Markdown are a great way of combining text and code.
- They look ugly but they compile well.
- The problem-sets and final are all done in R Markdown.
- The main advantage of R Markdown is that it makes the code reproducible:
  - I do not need to believe you to recheck your work.

R Markdown

- Compiles into three formats: `HTML`, `PDF`, or `DOCX`.
  - Of these three formats, you can only do `HTML` without installing extra software.
  - But you only need `HTML`! The `HTML` has also the advantage of being easy to customize.
- If you ever want to build a `PDF`, *Google* how to install something called `tinytex`.
- But how does R Markdown work?

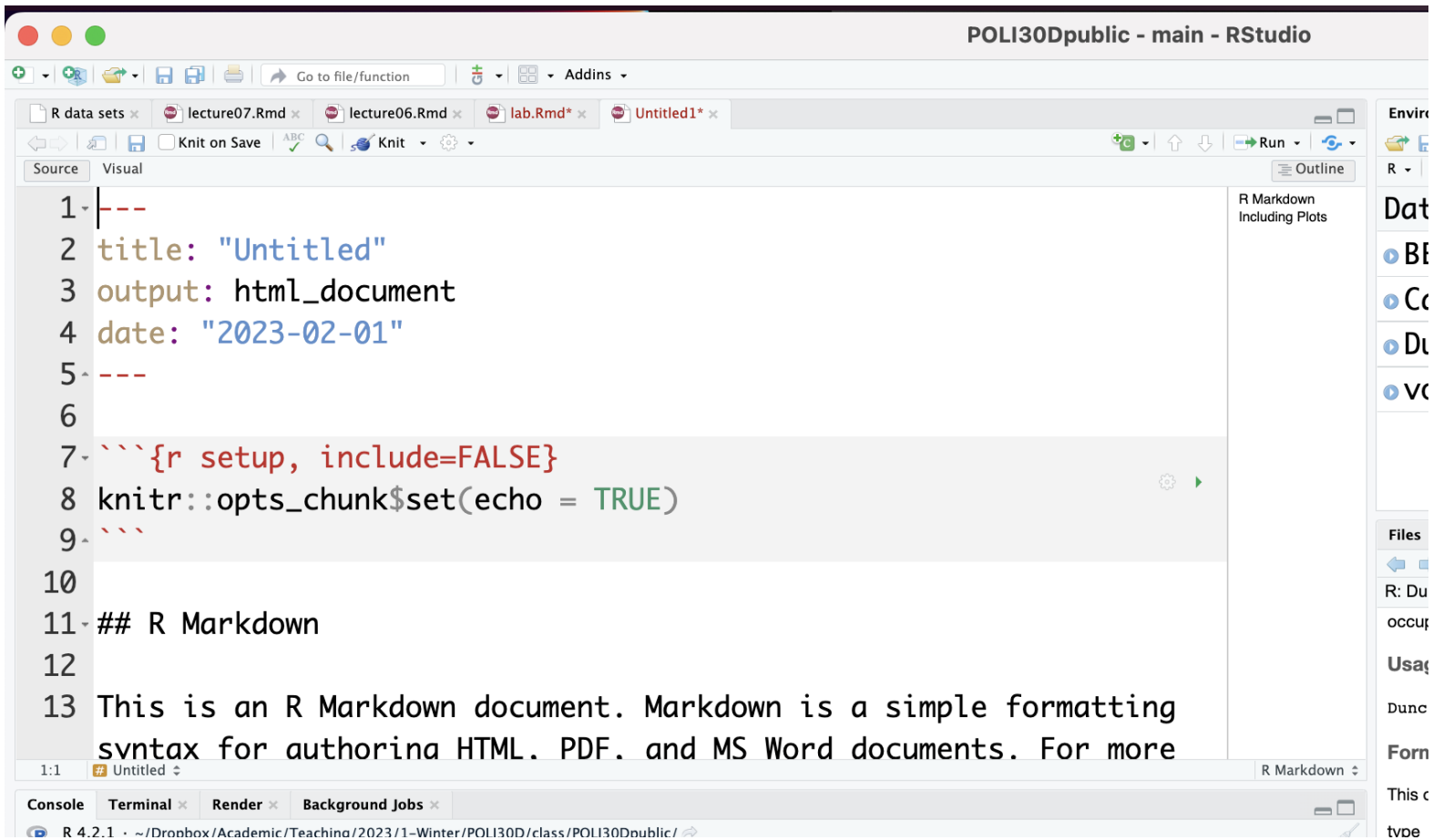
R Markdown

- To get started with a R Markdown, go to `File > New File > R Markdown...`.
- It shows the image below. For now, click `OK` to proceed.



R Markdown

It creates a default R Markdown for you to get started, as you can see below.



R Markdown

- To get started, we are going to compile the markdown.
- Compiling transforms the `.Rmd` into a report in the format you asked it to do.
  - For this class, it is going to be `HTML`.
- We compile by running the Knit command. But don't click there yet!
- Before clicking, a quick detour...

Quick Detour - Packages

Packages

- Your R is good, but not that powerful.
- The power of R is the fact that many people around the world can build R code and share.
- People do that by building `packages`. Think of packages as the R supplements.
- If you go to the gym and take whey to grow muscle, your R does the same by download and running packages.
- Each package was built to solve a particular problem. By having a problem, you can google a package that solves it.

Install Packages

- To install a package you need to run:

```
install.packages('name_of_the_package')
```

- And here is a list of packages in the R repo.
- There are almost 20k packages by now! Go ahead and run:

```
install.packages('tidyverse')
```

- Tidyverse is a family of packages useful for data analysis.

Loading Packages

- Once done installing, the package is downloaded, but not *loaded* in the memory.
- R does not load packages automatically, because for each loaded package, you eat up a little of the computer's memory.
- So you need to ask R to load it. To load the tidyverse family we do:

```
library('tidyverse')
```

- And now you have it there.

Packages

Two important facts about packages:

1. You just need to **install** a package **once**. After installing, it is in your computer. You may need to update it, but that's not frequent.
2. You **always** need to **load** the packages you are using. Get into the habit of running `library(tidyverse)` first thing when you open the computer.

Also important:

Don't be afraid of installing new packages, it is like giving vitamins and minerals to your R.

Back to R Markdown

R Markdown

- Compiling transforms the `.Rmd` into a report in the format you asked it to do.
- We compile by running the Knit button (

)
- Now go ahead and click there! If you did not use the command before, it is going to install a few packages.
- Then it is going to ask to save the file. Save as `lab04.Rmd`, in a folder that you will remember later.
- Once done, you should go to the folder and see the `lab04.Rmd` and another file, `lab04.html`.
  - The `HTML` is a file that you can open in your internet browser!
- Now let's learn each of the parts of an R Markdown file

R Markdown - YAML

- `YAML` (or Yet Another Markup Language, hahaha): This is the head of your document.
- This part gives the instructions regarding what type of document you want and more.
- We are not covering it here, but an advanced user should be aware of these tools.

```
1 ---
2 title: "Untitled"
3 output: html_document
4 date: "2023-02-01"
5 ---
6
```

R Markdown - YAML

- `YAML`: My suggestion is keep it simple:
  - `output: html_document`. Don't change unless required.
  - `title:`: Add the title to your document, unless already done.
  - `date:`: You can write a date. Note that I do an R command that generates the date automatically.
  - `author:`: Your name here!
- Example:

```
---
title: "My nice title goes here"
output: html_document
author: "Smart Student"
date: "`r Sys.Date()`"
---
```

R Markdown - Text Blocks

- The text in the Markdown is going to be displayed as regular text.
- There are several ways to customize it.
- Headings:

```
7
8 # First level heading
9
10 ## Second level heading
11
12 ### Third level heading (and so on...)
13
```

R Markdown - Text Blocks

- The text in the Markdown is going to be displayed as regular text.
- Paragraphs:

```
This is my nice paragraph. It is going to look great!
```

- And you can write as much as you want in a paragraph. It is pretty much up to you.

R Markdown – Text Blocks

- Paragraphs: another example:

```
11 ## R Markdown
12
13 This is an R Markdown document. Markdown is a simple formatting syntax
   for authoring HTML, PDF, and MS Word documents. For more details on using
   R Markdown see <http://rmarkdown.rstudio.com>.
14
15 When you click the **Knit** button a document will be generated that
   includes both content as well as the output of any embedded R code chunks
   within the document. You can embed an R code chunk like this:
```

R Markdown - Text Blocks

- The text in the Markdown is going to be displayed as regular text.
- We are going to display: My work is **great**. It is *nice* to learn ***R Markdown***!
- **Bold**, *Italic*, and ***Bold*** + ***Italic***:

```
13
14 This is my nice paragraph. It is going to look great!
15
16 My work is **great**. It is *nice* to learn ***R Markdown***!
17
```

R Markdown - Text Blocks

- We can do ordered and unordered lists. Ordered start with numbers. Unordered start with star (*) or dash (-). Lower levels are followed by +.

```
Now I am going to build an unordered list:

- First element
- Second element
- Third element
  + Third-and-a-half element
  + Third-and-three-quarters element

And when I go shopping today, I am going to buy:

1. Rice
2. Beans
3. Soda (two types):
  + Diet coke
  + Sprite
4. Impossible burger
```

R Markdown - Text Blocks

- And we can do nice and hard equations.
  - But you would need to learn something called LaTeX to know how to do them.

$$a^2 + b^2 = c^2$$

- And an equation in the same line would be: $a^2 + b^2 = c^2$.
- To do equations in the middle of the text you should use two $ signs ($$) to start and two $ signs ($$) to end.
- To do equations in-line, you should use one $ signs ($) to start and one $ sign ($) to end.

R Markdown: Code chunks

- So far we used R Markdown as a text processor. But the power comes from the fact that we can do coding.
- To do in-line coding we do:

```
two plus two equals `r 2 + 2`!
```

- And it shows two plus two equals 4.
- This symbol (` ) we call the tick mark. When you have one tick mark followed by r, it understands that it is R code ( `r).
- You should end with another tick mark: (`r 2+2`).

R Markdown: Code chunks

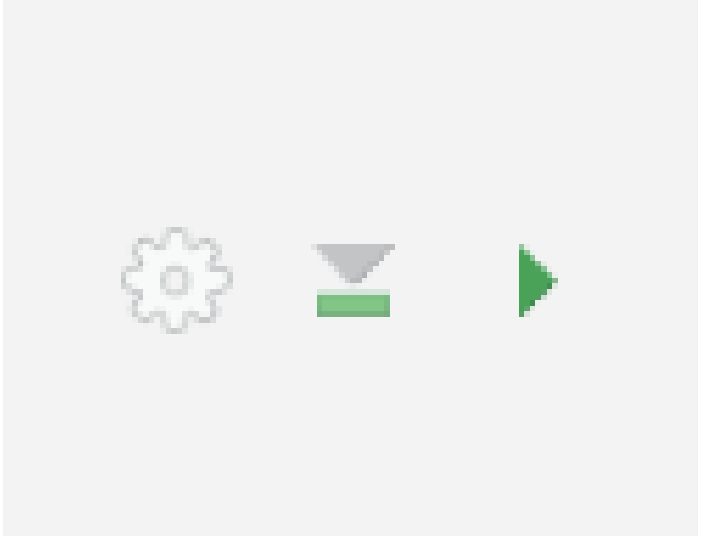
- But sometimes we need big code chunks.
- Here is how we do those:

````
```{r}
# This is a code chunk

# Loading the voting dataset
voting <- read.csv("https://raw.githubusercontent.com/umbertomig/POLI30Dpublic/main/data/voting.csv")

# Mean of voted variable
mean(voting$voted)
```
````

- You open with (` ```{r} `) and close with (` ``` `). In the middle, as many code lines as you need.

R Markdown: Code chunks

- And if you want to run your code chunks, you can use the little green play button.
  - It is located in the right-hand side of each of your chunks.



R Markdown - Good Practices

Good practices:

1. Try to code clean: don't do all in one chunk.
   - It is ok to alternate chunks with paragraphs of plain text. That's the purpose of R markdown!

2. Comment your code as if you were doing a R Script.
   - Remember, you will forget how you did, but you may need to do it again...

3. Always compile! It is not because you run the code and it works that your R Markdown will compile.
   - You should Knit before you submit.

R Markdown - Good Practices

Good practices:

4. As a general rule, never install packages on R Markdown:
   - It is going to get confused where is the place to download the package and will crash.

5. PDFs are great, DOCX also good, but HTML is the best for begginer R coders.

6. The internet is a great source for help.
   - Example, the R Markdown Cheat Sheet.

Today's Lab

- Learn R Markdown
- Install Packages

Next Lab

- Variables within data.frames
- Learn how to explore a dataset
- How to build nice plots
- Elementary data analysis

Questions?

See you in the next lab!