Computational Bootcamp 5: LaTeX, Markdown, and Formatting Documents for Social Science

Ankushi Mitra

Department of Government

Georgetown University

August 18, 2023

What We'll Be Covering Overall

1. Software installation, file management
2. Basics of R: data structures, writing code, creating objects, packages
3. R: working with datasets
4. More R: data manipulation, visualization
5. LaTeX: producing documents with Markdown and Overleaf

What We'll Be Covering Today

1. Why document preparation tools?

What We'll Be Covering Today

1. Why document preparation tools?
2. RMarkdown

What We'll Be Covering Today

1. Why document preparation tools?
2. RMarkdown
3. LaTeX in Overleaf

What We'll Be Covering Today

1. Why document preparation tools?
2. RMarkdown
3. LaTeX in Overleaf
4. LaTeX in RMarkdown

Why document preparation tools?

- In academia, we often want to use complicated symbols and formatting in our documents like:

$$\begin{cases} \frac{\partial}{\partial b_0} \sum_{i=1}^{n} \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \\ \frac{\partial}{\partial b_1} \sum_{i=1}^{n} \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \\ \quad \vdots \\ \frac{\partial}{\partial b_{10}} \sum_{i=1}^{n} \{y_i - (b_0 + b_1 x_{1i} + b_2 x_{2i} + \cdots + b_{10} x_{10i})\}^2 = 0 \end{cases}$$

- This is something normal text editors like word and google docs don't do well and aren't really meant for.
- LaTeX and RMarkdown are document preparation tools that can help you produce nice-looking, professional documents.

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio
  - Click on "File" > "New File" > "R Markdown"

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio
  - Click on "File" > "New File" > "R Markdown"
  - Choose the output format (HTML, PDF, Word, etc.).

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio
  - Click on "File" > "New File" > "R Markdown"
  - Choose the output format (HTML, PDF, Word, etc.).
  - Give your document a title and author name and save it.

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio
  - Click on "File" > "New File" > "R Markdown"
  - Choose the output format (HTML, PDF, Word, etc.).
  - Give your document a title and author name and save it.
- Document structure

RMarkdown

- RMarkdown comes from an earlier language called Markdown, which was originally designed for writing simple HTML.
- RMarkdown allows you to create documents that integrate narrative text, code chunks, and the output generated by running the code. Install the package *rmarkdown*.
- To create an R Markdown document,
  - Open RStudio
  - Click on "File" > "New File" > "R Markdown"
  - Choose the output format (HTML, PDF, Word, etc.).
  - Give your document a title and author name and save it.
- Document structure
  - An R Markdown document consists of text written in a combination of plain text and markdown, interspersed with code chunks.

RMarkdown

- At the top of your document, The YAML header serves as a "settings" section. It provides instructions on how to process and render the document.
  - It's enclosed by three hyphens at the top of the file.

RMarkdown

- At the top of your document, The YAML header serves as a "settings" section. It provides instructions on how to process and render the document.
  - It's enclosed by three hyphens at the top of the file.
  - Common YAML settings include:
    - title: Specifies the title of your document.
    - author: Specifies the author's name.
    - output: Specifies the desired output format

RMarkdown

- At the top of your document, The YAML header serves as a "settings" section. It provides instructions on how to process and render the document.
  - It's enclosed by three hyphens at the top of the file.
  - Common YAML settings include:
    - title: Specifies the title of your document.
    - author: Specifies the author's name.
    - output: Specifies the desired output format
- Code chunks are blocks of code that you can include within your document to execute and display the results directly in the document output. Code chunks are enclosed in three backticks (```) and specify the programming language (e.g., r for R code).

RMarkdown

- At the top of your document, The YAML header serves as a "settings" section. It provides instructions on how to process and render the document.
  - It's enclosed by three hyphens at the top of the file.
  - Common YAML settings include:
    - title: Specifies the title of your document.
    - author: Specifies the author's name.
    - output: Specifies the desired output format
- Code chunks are blocks of code that you can include within your document to execute and display the results directly in the document output. Code chunks are enclosed in three backticks (```) and specify the programming language (e.g., r for R code).

````
```{r}
x <- c(1:5)
mean(x)
```
````

RMarkdown

- The first code chunk is used to configure your document. This setup code is used to set a default behavior for all subsequent code chunks in your document. You can also call packages needed throughout the document here.

RMarkdown

- The first code chunk is used to configure your document. This setup code is used to set a default behavior for all subsequent code chunks in your document. You can also call packages needed throughout the document here.

````
```{r setup, include=FALSE, warnings=FALSE,
messages=FALSE, error=FALSE,
pagenumbering="arabic"}
knitr::opts_chunk$set(message = FALSE, warning=FALSE)
library(tidyverse)
library(ggplot2)
library(stargazer)
```
````

RMarkdown

- The code chunk is enclosed in three backticks (‘r setup‘), indicating that it’s a code chunk. ‘setup‘ is a label for this code chunk.

RMarkdown

- The code chunk is enclosed in three backticks (‘r setup‘), indicating that it’s a code chunk. ‘setup‘ is a label for this code chunk.
- This is followed by chunk options. include=FALSE indicates that the code and its output within this chunk should not be included in the final rendered document. warnings=FALSE, messages=FALSE, error=FALSE suppress warnings, messages, and errors from appearing in the output. pagenumbering=”arabic” sets the page numbering style.

RMarkdown

- The code chunk is enclosed in three backticks (‘r setup‘), indicating that it’s a code chunk. ‘setup‘ is a label for this code chunk.
- This is followed by chunk options. include=FALSE indicates that the code and its output within this chunk should not be included in the final rendered document. warnings=FALSE, messages=FALSE, error=FALSE suppress warnings, messages, and errors from appearing in the output. pagenumbering=”arabic” sets the page numbering style.
- `knitr::opts_chunk` sets global chunk options which will apply to all code chunks within the document. In this case, it suppresses messages and warnings.

RMarkdown

- The code chunk is enclosed in three backticks (‘r setup‘), indicating that it’s a code chunk. ‘setup‘ is a label for this code chunk.
- This is followed by chunk options. include=FALSE indicates that the code and its output within this chunk should not be included in the final rendered document. warnings=FALSE, messages=FALSE, error=FALSE suppress warnings, messages, and errors from appearing in the output. pagenumbering=”arabic” sets the page numbering style.
- `knitr::opts_chunk` sets global chunk options which will apply to all code chunks within the document. In this case, it suppresses messages and warnings.
- By including this setup code at the beginning of your RMarkdown document, you ensure consistent behavior for code chunks throughout the document.

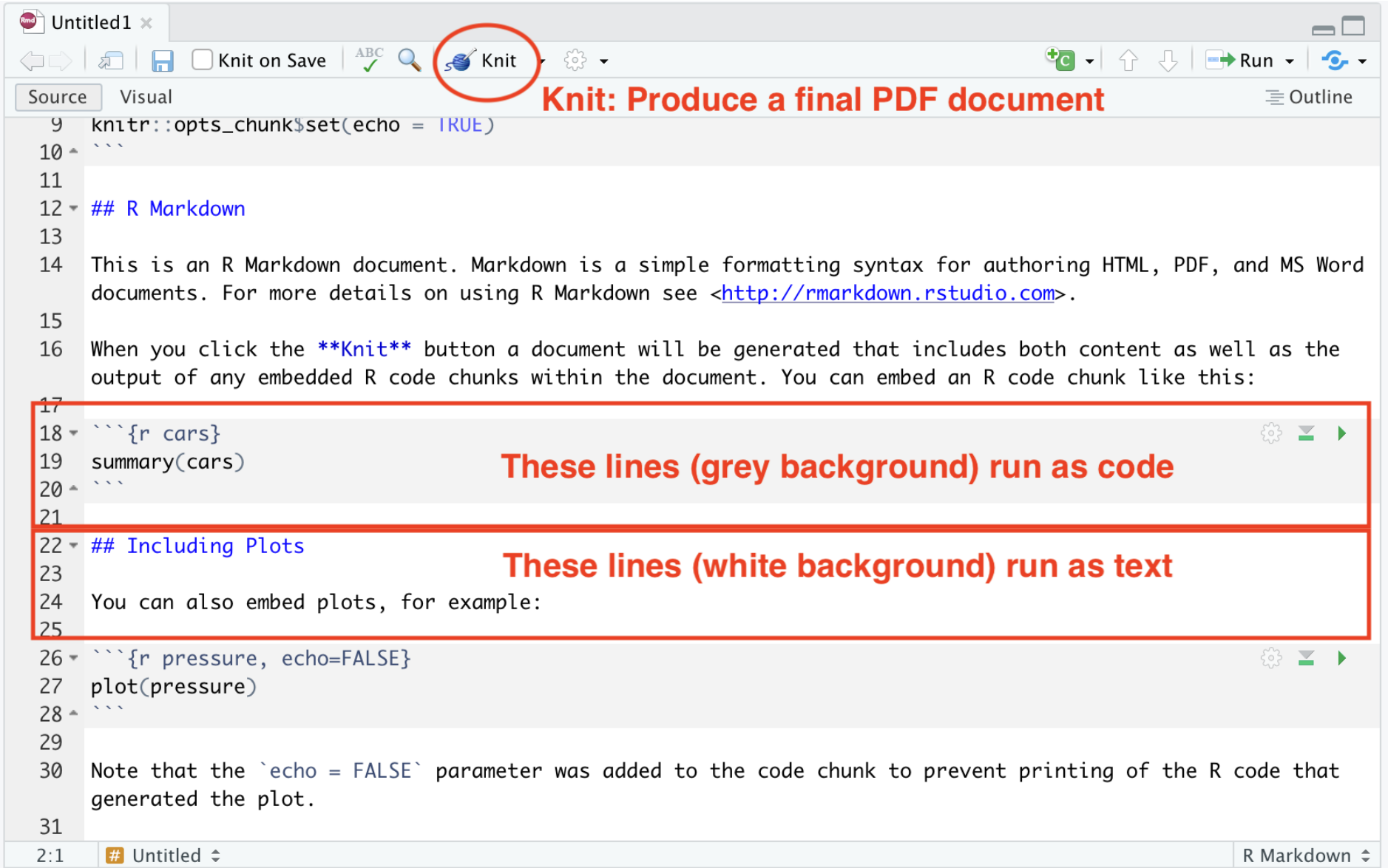
RMarkdown

- Write plain text outside the code chunks.

RMarkdown

- Write plain text outside the code chunks.
- Click the "Knit" button in RStudio to compile your RMarkdown document. R Markdown will execute the code chunks, generate outputs like plots and tables, and combine everything into the final document.

RMarkdown


Untitled1
Knit on Save
Knit
Run
Source
Visual
Outline
Knit: Produce a final PDF document
knitr::opts_chunk$set(echo = TRUE)
```
## R Markdown
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>.
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 cars}
summary(cars)
```
These lines (grey background) run as code
Including Plots
These lines (white background) run as text
You can also embed plots, for example:
```{r pressure, echo=FALSE}
plot(pressure)
```
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
2:1
Untitled
R Markdown
```

Working with RMarkdown

- Exercise:
  - Create a new RMarkdown document. Specify the output format as PDF.

Working with RMarkdown

- Exercise:
  - Create a new RMarkdown document. Specify the output format as PDF.
  - Save the document.

Working with RMarkdown

- Exercise:
  - Create a new RMarkdown document. Specify the output format as PDF.
  - Save the document.
  - Configure the document using the first code chunk.

Working with RMarkdown

- Exercise:
  - Create a new RMarkdown document. Specify the output format as PDF.
  - Save the document.
  - Configure the document using the first code chunk.
  - Create another code chunk. In this, set working directory for your RMarkdown document using setwd(). Take your R code from yesterday and add it here.

Working with RMarkdown

- Exercise:
  - Create a new RMarkdown document. Specify the output format as PDF.
  - Save the document.
  - Configure the document using the first code chunk.
  - Create another code chunk. In this, set working directory for your RMarkdown document using setwd(). Take your R code from yesterday and add it here.
  - Knit the document. Examine the PDF output.

LaTeX in Overleaf

- LaTeX is a typesetting system commonly used for creating documents with complex structures like academic papers, theses, and presentations. It's great for producing professional-looking documents with consistent formatting.

LaTeX in Overleaf

- LaTeX is a typesetting system commonly used for creating documents with complex structures like academic papers, theses, and presentations. It's great for producing professional-looking documents with consistent formatting.
- Overleaf is an online platform that allows users to write, compile, and collaborate on LaTeX documents without needing to install anything on their computers.

LaTeX in Overleaf

- LaTeX is a typesetting system commonly used for creating documents with complex structures like academic papers, theses, and presentations. It's great for producing professional-looking documents with consistent formatting.
- Overleaf is an online platform that allows users to write, compile, and collaborate on LaTeX documents without needing to install anything on their computers.
    - Open overleaf.com. Create an account or log in.

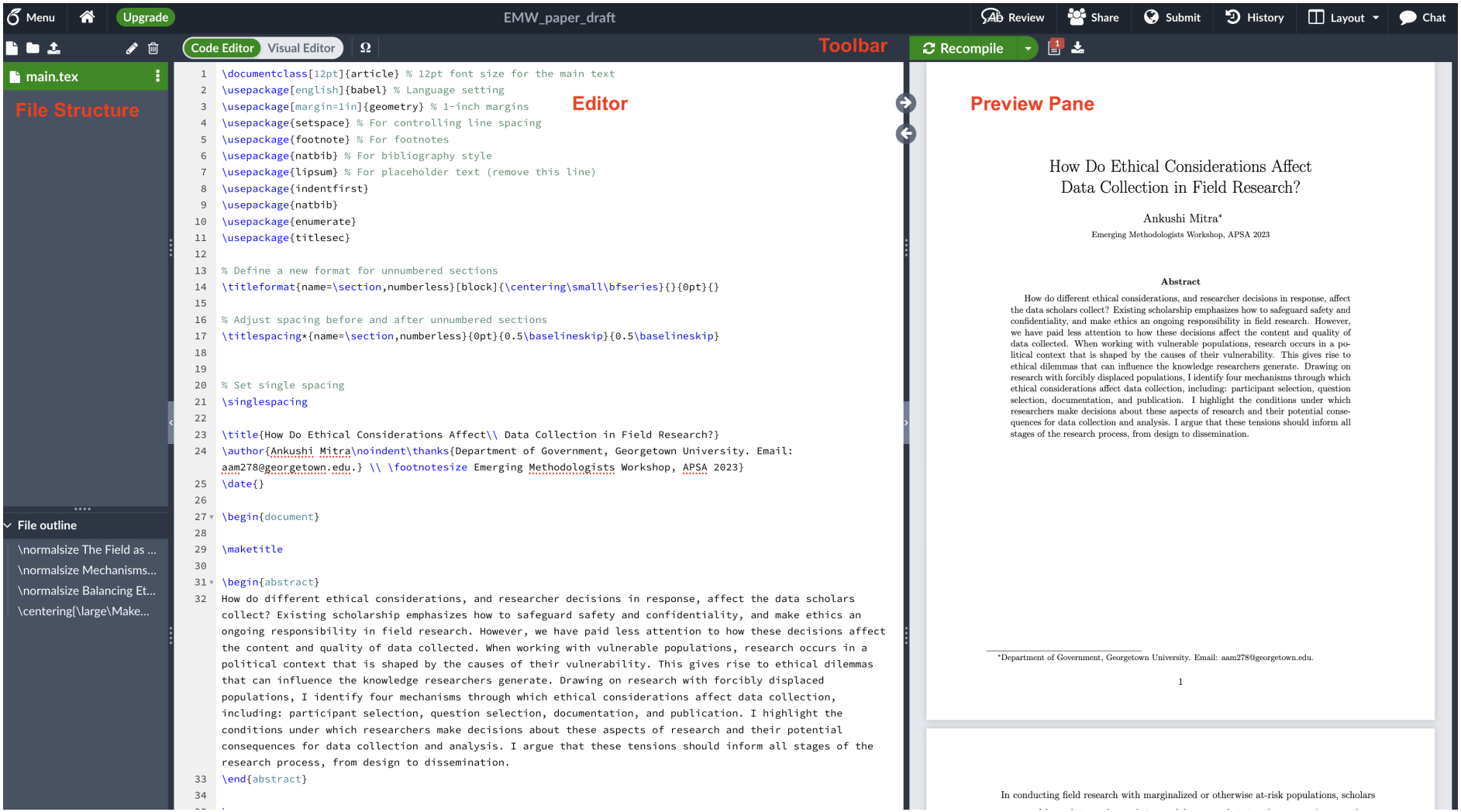
LaTeX in Overleaf

- LaTeX is a typesetting system commonly used for creating documents with complex structures like academic papers, theses, and presentations. It's great for producing professional-looking documents with consistent formatting.
- Overleaf is an online platform that allows users to write, compile, and collaborate on LaTeX documents without needing to install anything on their computers.
  - Open overleaf.com. Create an account or log in.
  - Click on "New Project" or "Create" to start a new LaTeX document. Give the project a name.

LaTeX in Overleaf

- LaTeX is a typesetting system commonly used for creating documents with complex structures like academic papers, theses, and presentations. It's great for producing professional-looking documents with consistent formatting.
- Overleaf is an online platform that allows users to write, compile, and collaborate on LaTeX documents without needing to install anything on their computers.
  - Open overleaf.com. Create an account or log in.
  - Click on "New Project" or "Create" to start a new LaTeX document. Give the project a name.
- Understanding the interface

LaTeX in Overleaf




```
\documentclass[12pt]{article} % 12pt font size for the main text
\usepackage[english]{babel} % Language setting
\usepackage[margin=1in]{geometry} % 1-inch margins
\usepackage{setspace} % For controlling line spacing
\usepackage{footnote} % For footnotes
\usepackage{natbib} % For bibliography style
\usepackage{lipsum} % For placeholder text (remove this line)
\usepackage{indentfirst}
\usepackage{natbib}
\usepackage{enumerate}
\usepackage{titlesec}

% Define a new format for unnumbered sections
\titleformat{name=\section,numberless}[block]{\centering\small\bfseries}{}{0pt}{}

% Adjust spacing before and after unnumbered sections
\titlespacing*{name=\section,numberless}{0pt}{0.5\baselineskip}{0.5\baselineskip}


% Set single spacing
\singlespacing

\title{How Do Ethical Considerations Affect\\ Data Collection in Field Research?}
\author{Ankushi Mitra\noindent\thanks{Department of Government, Georgetown University. Email:
aam278@georgetown.edu.} \\ \footnotesize Emerging Methodologists Workshop, APSA 2023}
\date{}

\begin{document}

\maketitle

\begin{abstract}
How do different ethical considerations, and researcher decisions in response, affect the data scholars
collect? Existing scholarship emphasizes how to safeguard safety and confidentiality, and make ethics an
ongoing responsibility in field research. However, we have paid less attention to how these decisions affect
the content and quality of data collected. When working with vulnerable populations, research occurs in a
political context that is shaped by the causes of their vulnerability. This gives rise to ethical dilemmas
that can influence the knowledge researchers generate. Drawing on research with forcibly displaced
populations, I identify four mechanisms through which ethical considerations affect data collection,
including: participant selection, question selection, documentation, and publication. I highlight the
conditions under which researchers make decisions about these aspects of research and their potential
consequences for data collection and analysis. I argue that these tensions should inform all stages of the
research process, from design to dissemination.
\end{abstract}
```

Basic LaTeX Document

- LaTeX has its own syntax and rules, so it will take time to become familiar with it.

Basic LaTeX Document

- LaTeX has its own syntax and rules, so it will take time to become familiar with it.
- Set up a document:

Basic LaTeX Document

- LaTeX has its own syntax and rules, so it will take time to become familiar with it.
- Set up a document:

```
\documentclass{article}
\begin{document}
Hello, world.
\end{document}
```

Basic LaTeX Document

- LaTeX has its own syntax and rules, so it will take time to become familiar with it.
- Set up a document:

```
\documentclass{article}
\begin{document}
Hello, world.
\end{document}
```

- Add sections to your document:

Basic LaTeX Document

- LaTeX has its own syntax and rules, so it will take time to become familiar with it.
- Set up a document:

```
\documentclass{article}
\begin{document}
Hello, world.
\end{document}
```

- Add sections to your document:

```
\section{Introduction}
```

Basic LaTeX Document

- Format text and create lists:

Basic LaTeX Document

- Format text and create lists:

```
\textbf{Bold text}
\textit{Italic text}

\begin{itemize}
    \item Item 1
    \item Item 2
\end{itemize}
```

Basic LaTeX Document

- Format text and create lists:

```
\textbf{Bold text}
\textit{Italic text}

\begin{itemize}
    \item Item 1
    \item Item 2
\end{itemize}
```

- Add equations:

Basic LaTeX Document

- Format text and create lists:

```
\textbf{Bold text}
\textit{Italic text}

\begin{itemize}
    \item Item 1
    \item Item 2
\end{itemize}
```

- Add equations:

```
\begin{equation}
    E=mc^2
\end{equation}
```

Formatting Regression Tables

- *stargazer* is a popular R package that helps you create and customize nice-looking tables of regression results, which can be used in research papers and presentations.

Formatting Regression Tables

- *stargazer* is a popular R package that helps you create and customize nice-looking tables of regression results, which can be used in research papers and presentations.
- Install and load the package *stargazer* in R.

Formatting Regression Tables

- *stargazer* is a popular R package that helps you create and customize nice-looking tables of regression results, which can be used in research papers and presentations.
- Install and load the package *stargazer* in R.
- The basic syntax of the *stargazer()* function is *stargazer(model)*.

Formatting Regression Tables

- *stargazer* is a popular R package that helps you create and customize nice-looking tables of regression results, which can be used in research papers and presentations.
- Install and load the package *stargazer* in R.
- The basic syntax of the *stargazer()* function is *stargazer(model)*.
- Exercise: Pass the regression model you created from the *economics* dataset as an argument to the *stargazer()* function in your RMarkdown document.

Formatting Regression Tables

- *stargazer* is a popular R package that helps you create and customize nice-looking tables of regression results, which can be used in research papers and presentations.
- Install and load the package *stargazer* in R.
- The basic syntax of the *stargazer()* function is *stargazer(model)*.
- Exercise: Pass the regression model you created from the *economics* dataset as an argument to the *stargazer()* function in your RMarkdown document.
- Copy and paste the output from the R console to the editor window in Overleaf. Make sure it is enclosed within `\begin{document}` and `\end{document}`. Click Recompile.

Regressions in R

```
> model <- lm(uempmed ~ pce + psavert, data = economics)
> summary(model)

Call:
lm(formula = uempmed ~ pce + psavert, data = economics)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6236 -1.3653 -0.1258  0.9355 10.3775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.451e+00  6.342e-01  -10.17   <2e-16 ***
pce          1.459e-03  4.354e-05   33.50   <2e-16 ***
psavert      9.372e-01  5.225e-02   17.94   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.259 on 571 degrees of freedom
Multiple R-squared:  0.6984,    Adjusted R-squared:  0.6974
F-statistic: 661.2 on 2 and 571 DF,  p-value: < 2.2e-16
```

Output in R Console

Table 1:

	Dependent variable:
	uempmed
pce	0.001^{***}
	(0.00004)
psavert	0.937^{***}
	(0.052)
Constant	-6.451^{***}
	(0.634)
Observations	574
R^2	0.698
Adjusted R^2	0.697
Residual Std. Error	2.259 (df = 571)
F Statistic	661.223^{***} (df = 2; 571)
Note:	*p<0.1; **p<0.05; ***p<0.01

Output in Overleaf

LaTeX in RMarkdown

- You can also use LaTeX in RMarkdown. One way is to use the distribution TinyTex.

```
install.packages("tinytex")
tinytex::install_tinytex()
```

LaTeX in RMarkdown

- You can also use LaTeX in RMarkdown. One way is to use the distribution TinyTex.

```
install.packages("tinytex")
tinytex::install_tinytex()
```

- Once TinyTeX is installed, you can use it to compile your RMarkdown documents that include LaTeX code. When you render your RMarkdown document, TinyTeX will automatically handle the LaTeX compilation process to generate the PDF output.

Formatting Regression Tables

- LaTeX code generated by *stargazer* is rendered in RMarkdown using *tinytex*.

Formatting Regression Tables

- LaTeX code generated by *stargazer* is rendered in RMarkdown using *tinytex*.
- Exercise:
  - Load the packages *tidyverse*, *stargazer* and *tinytex* in RMarkdown.

Formatting Regression Tables

- LaTeX code generated by *stargazer* is rendered in RMarkdown using *tinytex*.
- Exercise:
  - Load the packages *tidyverse*, *stargazer* and *tinytex* in RMarkdown.
  - Specify results='asis' as a chunk option to indicate that the output of the R code within the chunk should be directly inserted into the document.

````
```{r, results='asis'}
# your code here
```
````

Formatting Regression Tables

- LaTeX code generated by *stargazer* is rendered in RMarkdown using *tinytex*.
- Exercise:
  - Load the packages *tidyverse*, *stargazer* and *tinytex* in RMarkdown.
  - Specify results='asis' as a chunk option to indicate that the output of the R code within the chunk should be directly inserted into the document.

    ```
    ```{r, results='asis'}
    # your code here
    ```
    ```

- Pass the regression model you created from the *economics* dataset as an argument to the *stargazer()* function within this code chunk.

Formatting Regression Tables

- LaTeX code generated by *stargazer* is rendered in RMarkdown using *tinytex*.
- Exercise:
  - Load the packages *tidyverse*, *stargazer* and *tinytex* in RMarkdown.
  - Specify results='asis' as a chunk option to indicate that the output of the R code within the chunk should be directly inserted into the document.

    ```
    ```{r, results='asis'}
    # your code here
    ```
    ```

- Pass the regression model you created from the *economics* dataset as an argument to the *stargazer()* function within this code chunk.
- Knit the RMarkdown document. Examine the PDF output.

Formatting Regression Tables

- Try modifying the chunk option to also include echo=FALSE. Knit the RMarkdown document. Examine the PDF output.

Formatting Regression Tables

- Try modifying the chunk option to also include echo=FALSE. Knit the RMarkdown document. Examine the PDF output.

```
```{r, results='asis', echo=FALSE}
#Run regression
model <- lm(uempmed ~ pce + psavert, data = economics)
#Create stargazer table
stargazer(model)
```
```

Resources for RMarkdown and LaTeX

- R Markdown: The Definitive Guide

Resources for RMarkdown and LaTeX

- R Markdown: The Definitive Guide
- R Markdown: R For Data Science

Resources for RMarkdown and LaTeX

- R Markdown: The Definitive Guide
- R Markdown: R For Data Science
- Overleaf: Learn LaTeX in 30 Minutes

Resources for RMarkdown and LaTeX

- R Markdown: The Definitive Guide
- R Markdown: R For Data Science
- Overleaf: Learn LaTeX in 30 Minutes
- Overleaf Tutorials