Lab 1: Workflow Workshop

TA: Ray Caraher

UMass Amherst – Econ 755

Spring 2025

Outline

Introduction to Workflow

A good **workflow** is a **systemic, organized, and efficient** process for writing, managing, and editing code-based projects. It helps ensure **clarity, reproducibility, and efficiency** when working with projects with lots of data, code, and results.

Econometric Workflow

An **econometric workflow** is slightly different from a more general coding workflow, since our **products are research papers and presentations**, but the overall **characteristics of a good workflow** are the same:

- **Structured & Organized**: Uses a well-defined directory structure (e.g., data/, scripts/, results/).

Econometric Workflow

An **econometric workflow** is slightly different from a more general coding workflow, since our **products are research papers and presentations**, but the overall **characteristics of a good workflow** are the same:

- **Structured & Organized**: Uses a well-defined directory structure (e.g., data/, scripts/, results/).
- **Reproducible**: Code should be easily re-runnable by anyone (most importantly, future you)

Econometric Workflow

An **econometric workflow** is slightly different from a more general coding workflow, since our **products are research papers and presentations**, but the overall **characteristics of a good workflow** are the same:

- **Structured & Organized**: Uses a well-defined directory structure (e.g., data/, scripts/, results/).
- **Reproducible**: Code should be easily re-runnable by anyone (most importantly, future you)
- **Modular & Scalable**: Uses functions and scripts instead of long, repetitive, copy-pasted code.

Econometric Workflow

An **econometric workflow** is slightly different from a more general coding workflow, since our **products are research papers and presentations**, but the overall **characteristics of a good workflow** are the same:

- **Structured & Organized**: Uses a well-defined directory structure (e.g., data/, scripts/, results/).
- **Reproducible**: Code should be easily re-runnable by anyone (most importantly, future you)
- **Modular & Scalable**: Uses functions and scripts instead of long, repetitive, copy-pasted code.
- **Version Controlled**: Tracks changes systematically using Git/GitHub.

Econometric Workflow

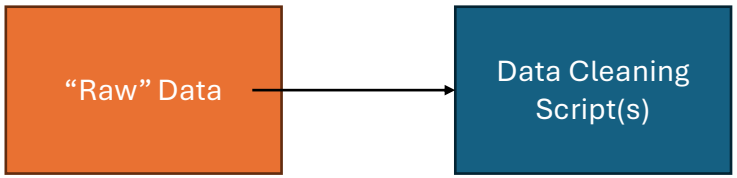
An **econometric workflow** is slightly different from a more general coding workflow, since our **products are research papers and presentations**, but the overall **characteristics of a good workflow** are the same:

- **Structured & Organized**: Uses a well-defined directory structure (e.g., data/, scripts/, results/).
- **Reproducible**: Code should be easily re-runnable by anyone (most importantly, future you)
- **Modular & Scalable**: Uses functions and scripts instead of long, repetitive, copy-pasted code.
- **Version Controlled**: Tracks changes systematically using Git/GitHub.
- **Documented**: Includes comments, README files, and clear instructions.

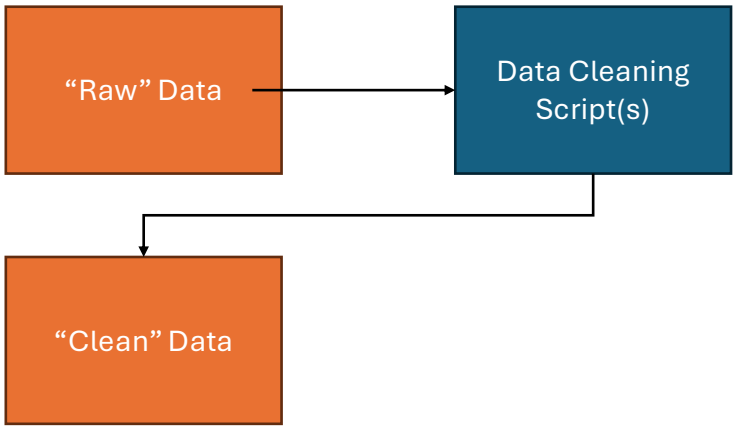
These characteristics should hold for **ALL** aspects of your project, including LaTex!

A General Workflow

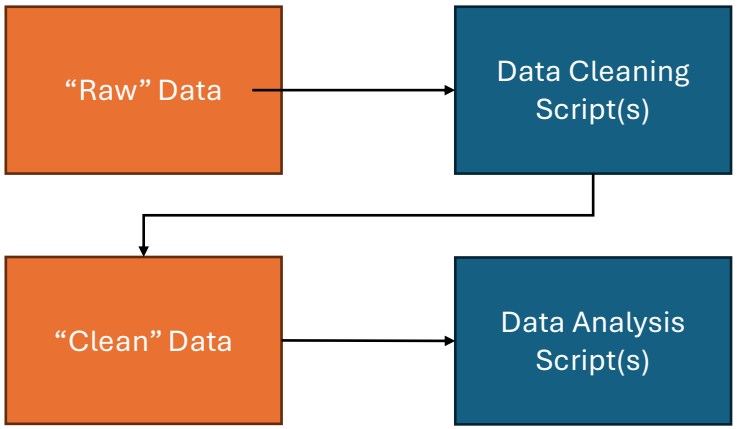
A General Workflow



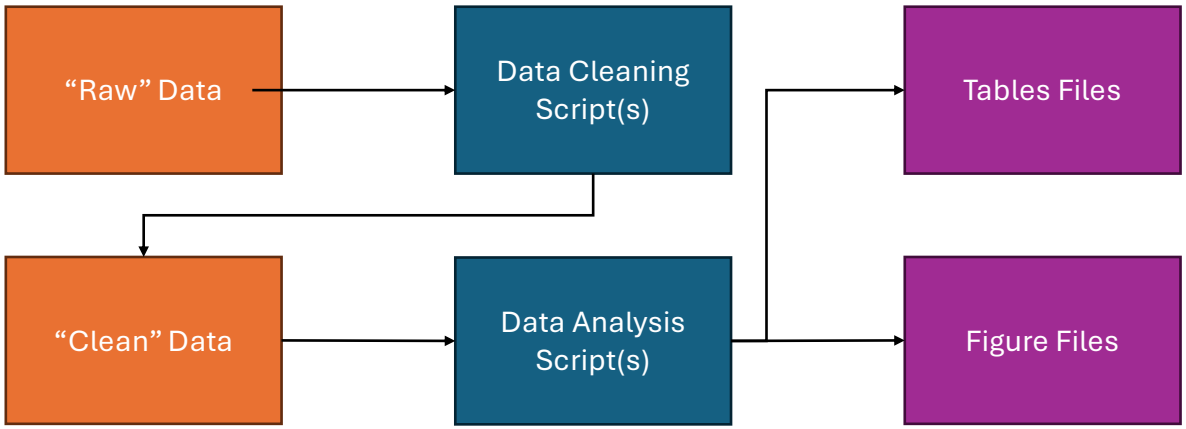
A General Workflow



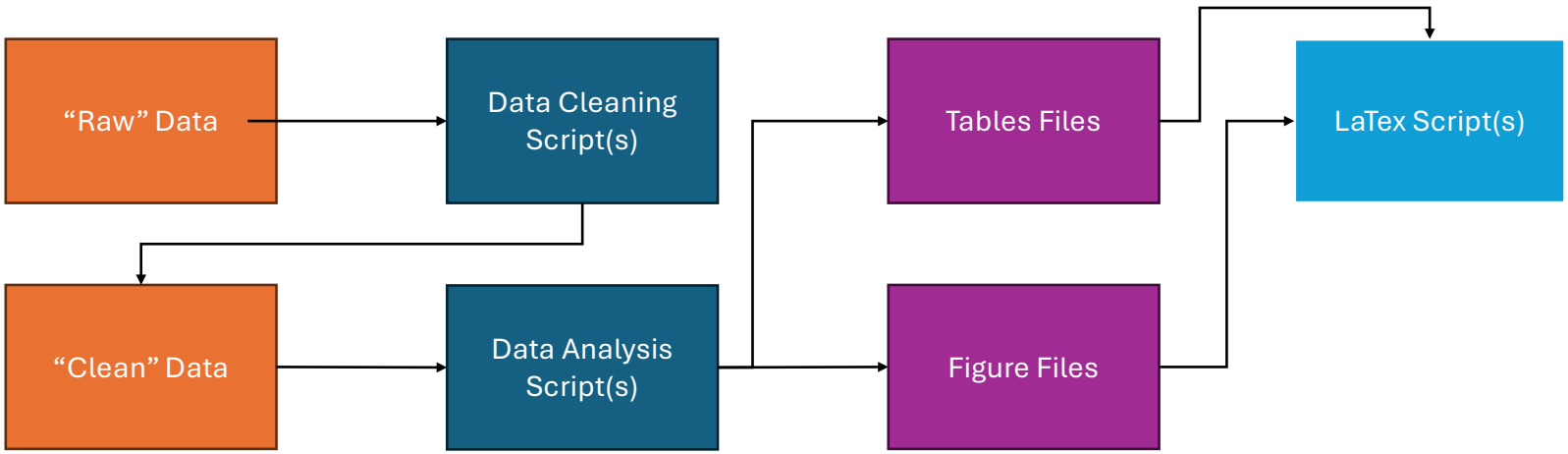
A General Workflow



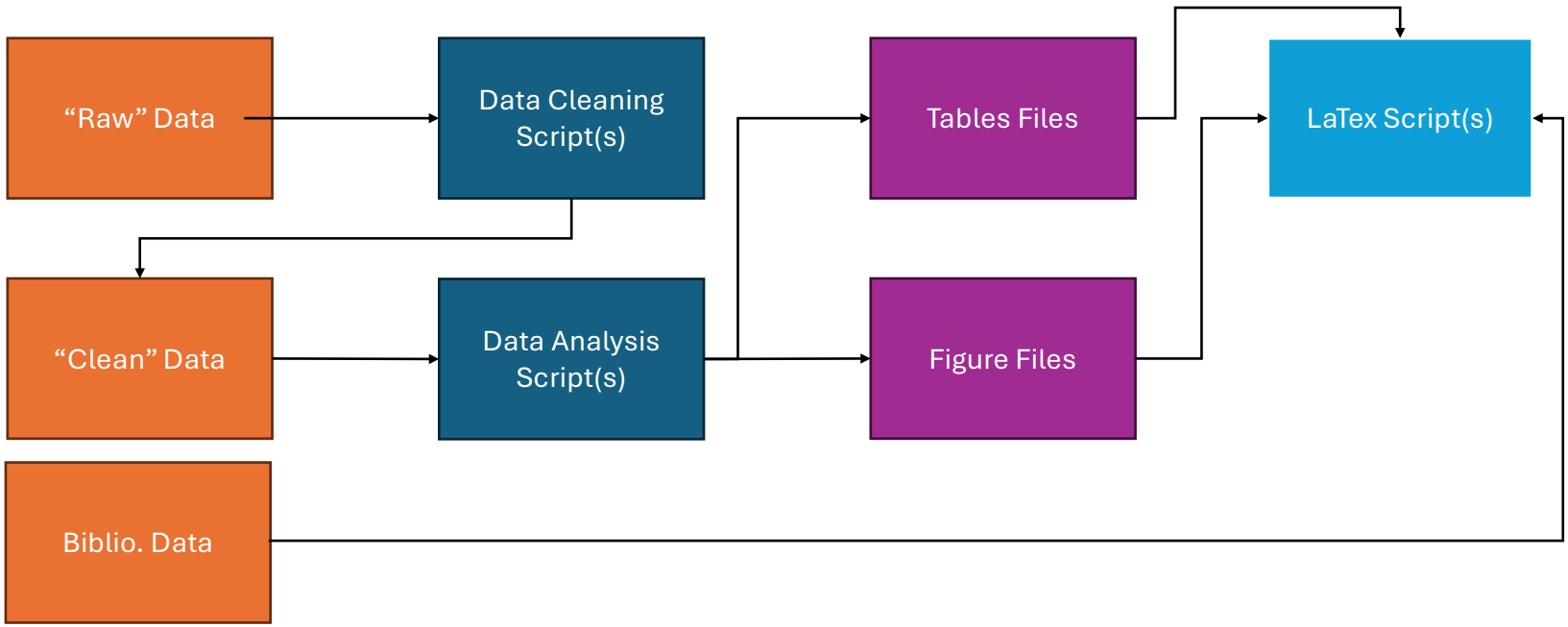
A General Workflow



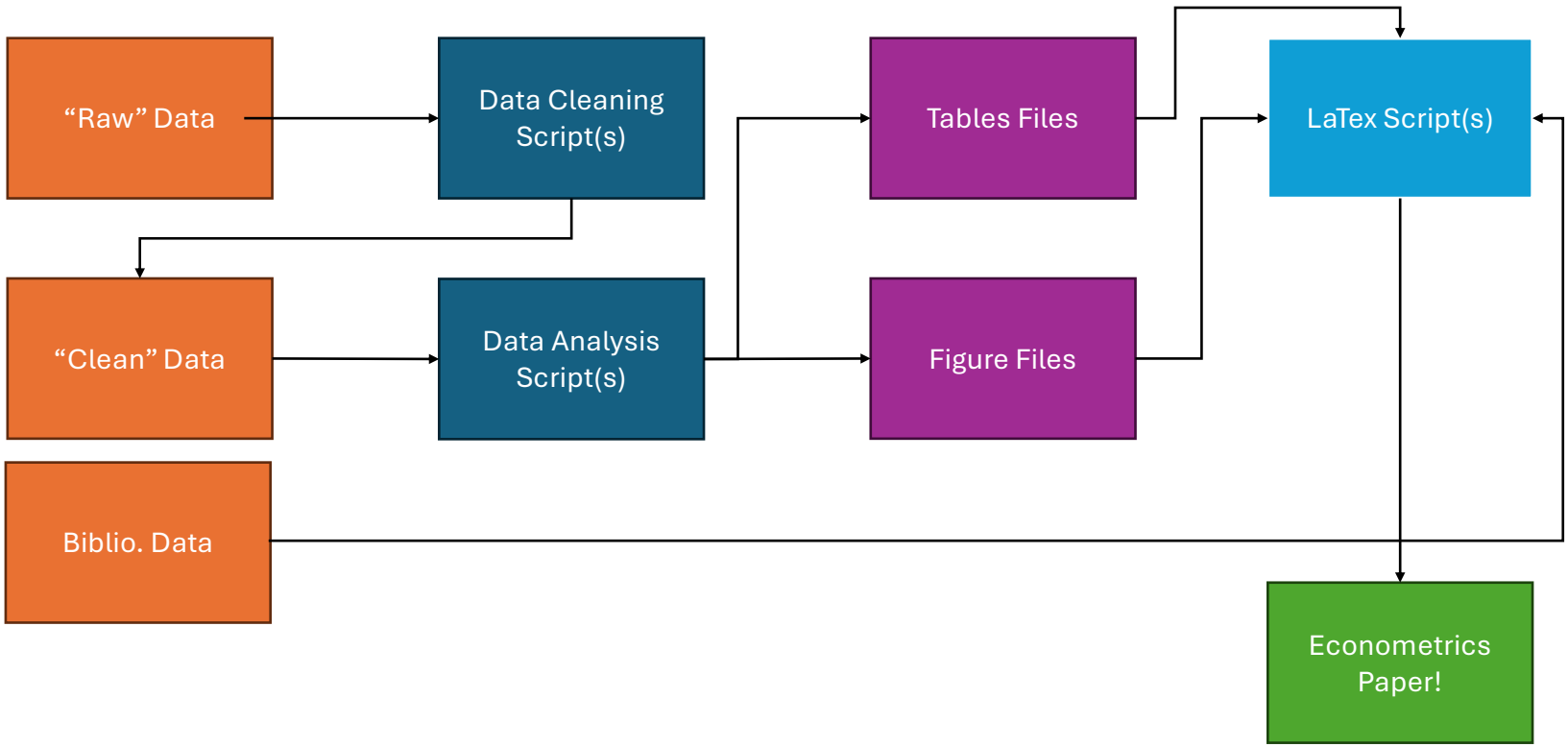
A General Workflow



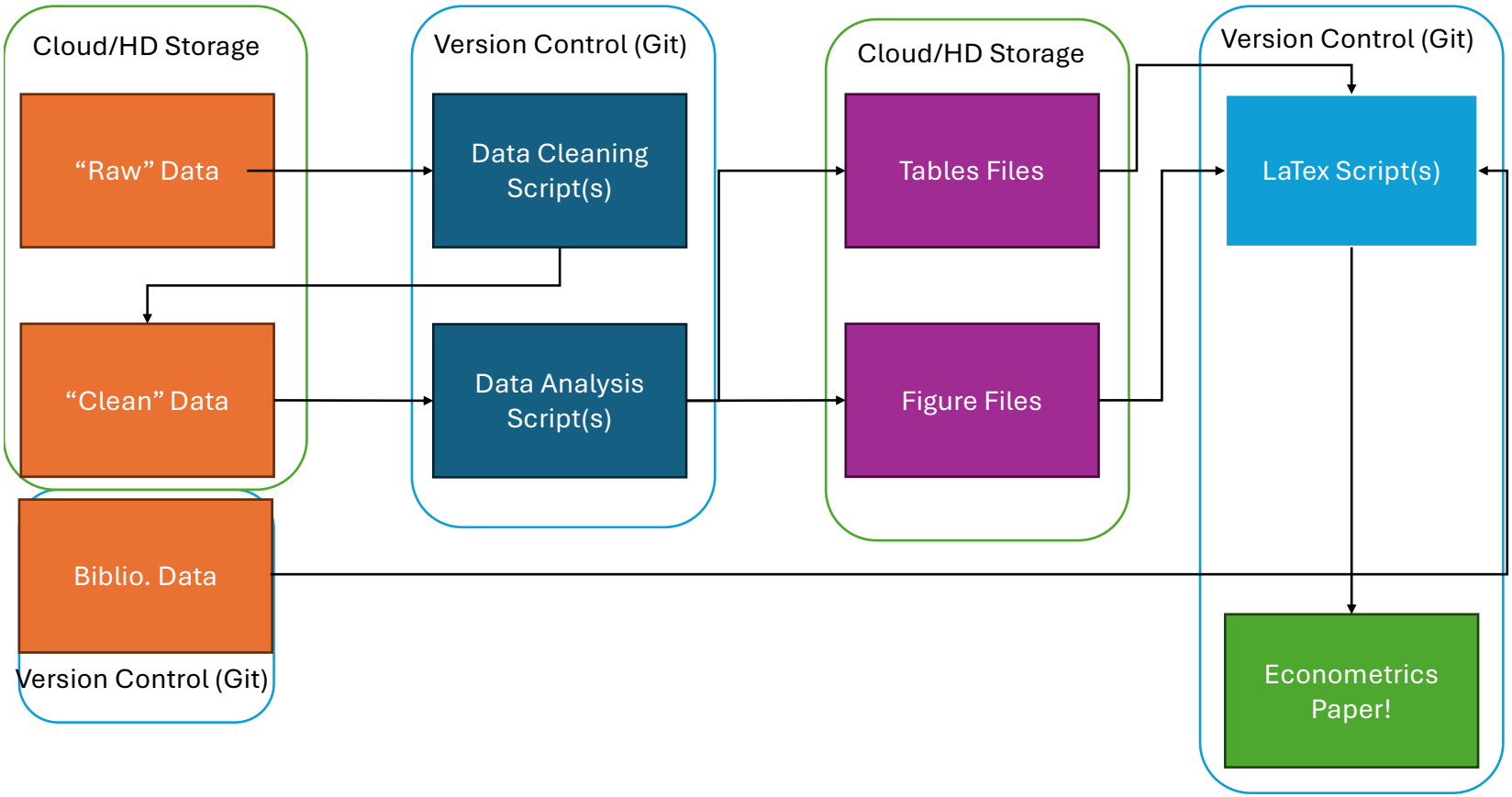
A General Workflow



A General Workflow



How Should I Store My Files?



Outline

What is Version Control?

- **Version Control:** A system for tracking changes to files over time.
- **Benefits:**
  - Keeps a history of all changes.
  - Allows easy rollback to previous versions.
  - Enables collaboration without overwriting work.
  - Helps ensure reproducability of code.
- **Popular Tools:** Git, GitHub, GitLab, Bitbucket.

What is Git?

- **Git** is a distributed version control system managed via web on GitHub.
- Allows users to **track changes, revert versions, and collaborate**.
- Works well with **R, Python, Stata, and LaTeX**.

Basic Git Workflow

Git commands are executed in the **system terminal**, but most IDEs have built-in Git commands so you usually don't need to worry about these!

1. **Initialize a Git repository:**

   ```
   git init
   ```

2. **Check file status:**

   ```
   git status
   ```

3. **Stage changes (prepare for commit):**

   ```
   git add my_script.R
   ```

4. **Commit changes (save a snapshot):**

   ```
   git commit -m "Updated regression model"
   ```

5. **Push to GitHub (backup and collaboration):**

   ```
   git push origin main
   ```

Storing Data

- Most data is too large to store using version control.
- The best option is to store it on the **Cloud** (e.g., UMass OneDrive) as well as on your local hard drive.
- Certain datasets may have **storage restrictions**, so ensure compliance with data policies.
- Consider using **data repositories** (e.g., Dataverse, Zenodo) for long-term storage and sharing.

Outline

Introduction to R

- R is a widely used open-source language for statistical computing.
- Highly customizable with thousands of packages available via CRAN.
- Download R from CRAN.

RStudio: The IDE for R

- RStudio is an Integrated Development Environment (IDE) for R.
- Features include:
  - Code editor with syntax highlighting.
  - Interactive console.
  - Built-in package manager.
- Download RStudio from here.

Resources for Learning R

- Data Science in a Box: Free introductory course.
- Swirl: Interactive R tutorials inside R.
- R for Data Science: Essential for learning tidyverse and data wrangling.
- freeCodeCamp's R Course: Comprehensive YouTube tutorial.

Introduction to Stata

- Stata is a statistical programming language widely used in social sciences.
- Unlike R, it is not free, but UMass provides access.
- No separate IDE is required—Stata has a built-in interface.
- Different versions available with student discounts at Stata's official site.
- Can access fpr free via UMass's Virtual Desktop (need to register).

Resources for Learning Stata

- Stata User Guide and Documentation.
- Stata's YouTube Channel: Video tutorials and guides.
- UCLA Stata Learning Modules: Comprehensive introduction.

Which One Should I Use? Pros and Cons of R

Both are widely used in statistical computing and have their own pros and cons. Here are some of my thoughts:

- **R** Pros:
  - Open-sourced (free and thousands of packages are available)
  - Widely used across disciplines and across industry/academia
  - Object-orientated language (better for more complicated projects/easier to translate to Python)
  - At the cutting-edge of data science *more broadly*
  - Marx would have used R
- **R** Cons:
  - Steeper (relative) learning curve due to more obtuse syntax
  - Not as "custom-made" for econometrics
  - More coding required for complex methods (not always a con)
  - Sometimes difficult to find packages to do the "latest" technique

Which One Should I Use? Pros and Cons of Stata

- **Stata** Pros:
  - More intuitive language with point-and-click implementations through GUIs
  - Purpose-built for econometrics and (probably) the most commonly used software within our field
  - At the cutting-edge of econometrics *more specifically*
- **Stata** Cons:
  - Expensive
  - Not object orientated (very "hacky" solutions when working on complex projects)
  - Sometimes make things too easy

Outline

What is LaTeX

- LaTeX is a typesetting coding language used to create professional-looking documents.
- It is widely used in the social sciences, especially in economics.
- Unlike WYSIWYG software (e.g., Microsoft Word), LaTeX uses codes and commands to generate a clean document.
- Benefits:
  - Automatic updating of figures and tables.
  - Simple reference management.
  - Version control with Git.

Getting Started with LaTeX

- LaTeX takes a plain-text document and compiles it into a professional PDF using a **TeX Engine**.
- The easiest way to get started is Overleaf, a web-based LaTeX editor.
- For larger projects, consider installing a local TeX Engine:
  - Windows: MiKTeX or TeX Live
  - Mac: MacTeX
  - Linux: TeX Live

LaTeX Editors

- Once you have a TeX Engine, you need a text editor to write your LaTeX code.
- Popular LaTeX editors:
  - Overleaf: Web-based, great for collaboration.
  - TeXstudio: Free and open-source, feature-rich.
  - TeXworks: Simple editor included with many TeX distributions.
  - TeXmaker: Clean interface with many useful features.
  - Visual Studio Code with LaTeX Workshop: Highly customizable.

Learning Resources for LaTeX

- Overleaf's LaTeX Guides: Beginner-friendly tutorials.
- LaTeX Wikibook: Comprehensive, open-source guide.
- Learn LaTeX in 30 Minutes (Overleaf): Quick-start guide.
- LaTeX-Tutorial.com: Step-by-step guides.
- ShareLaTeX's YouTube Channel: Video tutorials.

Outline

Example

We will write a "test" econometrics paper using R, LaTex, and good workflow habits. In this lab, we will use a sample of ACS data to estimate gender and racial wage gaps in R or Stata and report our results using LaTex.

First, navigate to the GitHub repo that I will use for this class: https://github.com/rpcaraher/econ755_labs.

Here, I will post the code and slides from labs, as well as the sourcecode used to compile them (including these slides!).

If you have Git on your computer, I recommend you "clone" this repo. Otherwise, you can just download it!