Makefile for Economists

An Introduction

Jesse Chieh Chen

2023–06–20

Road Map

1. The Purpose of Makefile
2. What is the Logical Structure of a Makefile?
3. Praxis
   - The Shell
   - Demonstration
4. Learn More

The Purpose of Makefile

Economists and their Computers

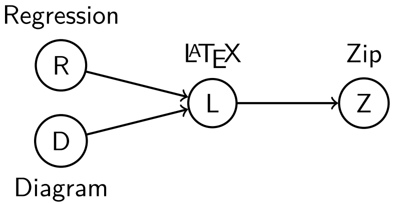
- What do economists (or a student of economics) do all day with the computer?

Economists and their Computers

- What do economists (or a student of economics) do all day with the computer?
- Usually two things:
  1. Run regression or run simulations.
  2. Produce LaTeX (or Beamer) documents.
- These tasks are often tied to each other and involves several, not difficult, but annoying steps.

Example: Homework

- An assignment requires the following workflow:
  1. (R) Run some regression.
  2. (D) Draw some diagram.
  3. (L) Put the regression result and the diagram in the LaTeX document.
  4. (Z) Put the LaTeX pdf and the regression code in a `zip` and upload it.
- Graphically, the workflow looks like this:



The Purpose of **Makefile**

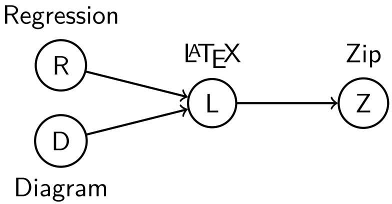
- A **Makefile** offers a way to specify and automate the entire process with a simple text file.
- A **Makefile** is a text file that specifies a workflow for the program `make`.
- The program `make` is a free and open source GNU project.
- It is pre-installed on almost all Unix-like systems, including MacOS.
  - It can be installed on Windows.
- In the following demonstration, I will focus on Unix-like systems.

What is the Logical Structure of a Makefile?

How is a Makefile Structured?

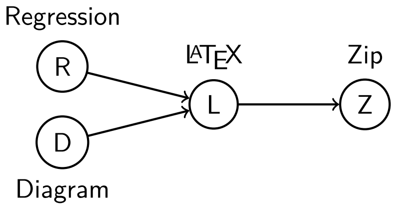
- The logic of a Makefile is to specify a target file, then its dependencies.
- So whenever a dependency is changed, the target file should be regenerated.
- For example, `homework.pdf` is dependent on `homework.tex`. So `homework.pdf` should be recompiled whenever `homework.tex` is changed.

Example: Homework



In this example, the dependency structure is as follows:

Example: Homework



In this example, the dependency structure is as follows:

- regression table depends on the code 1 and data files.
- diagram depends on code 2.
- `homework.pdf` depends on regressions table, diagram, and `homework.tex`.
- `homework.zip` depends on `homework.pdf`.

Praxis

Praxis

The Shell

A Brief Intro to Shell

- You will need some knowledge about the shell to use Makefiles.
- When you open a terminal, you are presented with a shell prompt.
- You can traverse the file system and execute commands with the shell prompt.

Basic Commands

- `pwd`: (print working directory) show where you are currently
- `cd`: (change directory) move to a different directory

`\`	root directory
`~`	home directory
`.`	current directory
`..`	parent directory

- `ls`: (list) list files in the current directory
- `cp`: (copy) copy files or directories
- `mv`: (move) move files or directories
- `rm`: (remove) remove file (CAREFUL! CANNOT UNDO!)
- `which`: show path of a command
- `clear`: clear the terminal screen

Useful Commands

- `pdflatex`: compile `.tex` file with `pdflatex` engine.
- `latexmk`: automate LaTeX compilation process
- `R`: interactive R console
- `Rscript`: run R script
- `zip`: make a `.zip` file
- `make`: to execute a Makefile

Praxis

Demonstration

Demonstration

- Example 1: Homework Example.
- Example 2: This Beamer Presentation.
- Example 3: Beamer Presentation on Chapter 5 of J-SEN.

Learn More

Resources

- A great resource is a lecture series called `The Missing Semester`, in it a lecture on `meta-programming` also introduces the use of Makefiles.
- `The Missing Semester` lecture series introduces a lot of “workflow” knowledge that is not typically covered in schools.
- Here is the documentation of make.
- You can find the demos of this presentation at `github.com/jessekelighine/makefile-for-economists`.