ECON-UN 4999 — Senior Honors Thesis Seminar

Workshop 2: Project-oriented workflows for economics research

Matthew Alampay Davis

October 16, 2023

Review: foundations of programming for economics research

Review: some useful general-purpose tools

- Quick run-through of data manipulation commands
- The conditional command: the ‘if’ function
  - if (statement) then (command to run)
  - if (statement) then (command to run) else (command to run if false)
- Iterative commands: the for loop
  - Run a command multiple times
  - Iterations defined by an ‘iterator’ (usually *i*) which may or may not alter some computation
  - Condition commands on the iterator
  - Pseudocode for last week’s example:
    1. iterating over values 1 to 200 (“for *i* in 1:200”)
    2. if *i* in this iteration has a remainder of 0 when divided by 20 “if(*i*%%20==0)”
    3. then print the statement “Iteration number [*i*] is a multiple of 20”

Review: data structure and information

- Data: what is it? how was it constructed? what shape is it in?
- Relationship between the shape/structure of data and the model of interest
- Dependence of commands (e.g., plotting) on the shape/structure of the data
- Emphasis on minimizing loss of information if avoidable

Anatomy of a script: an applied example

Some guidelines for writing scripts

1. First decide what you want the script to accomplish
2. Determine what inputs are required
3. Break this down into distinct steps
    - The preamble section
        - Loads relevant packages/functions (R)
        - Define any locals/globals/variables needed (Stata, R)
        - Load datasets used (if size is not a concern)
    - Can use these steps as placeholder comments in your script
    - Ensure these are written in sequence
4. Write the (pseudo)code that accomplishes this

Example: A project requires local precipitation data starting in 1960

Some considerations:

- How was the data constructed?
- Is this source reputable? Is it used in other studies?
- Do I have the skills, tools, and knowledge to work with this data? If not, do I have the resources acquire them?
- What format is it in? What units are the variables in?
- How large is the data?
- Is it publicly available?
    - If so, how do I access it? Can it be automated?
    - If not, can access be requested?
    - If so, when can I expect to gain access?
- What alternatives are there, if any?

Example: A project requires local precipitation data starting in 1960

Source 1:
Global Meteorological Forcing Dataset (GMFD) for Land Surface Modeling

- *Daily* precipitation at a 0.25-degree resolution from 1948 to 2018
- Data available for entire global surface except for Antarctica
- Accessible by online repository
- Each file represents a year, is about 1GB, and is in NetCDF gridded format

Example: outline/pseudo-code for downloading GMFD data

1. Create a directory to download the files into
2. Scrape the online repository for the relevant URLs
3. The download loop
   for every year y from 1960 to 2017:
   - 3.1 Download the file for year y into data/downloads
   - 3.2 Transform the data into a convenient format and shape
   - 3.3 Delete the original download to save space
   - 3.4 Save this processed dataset to data/intermediate

Example: A project requires local precipitation data starting in 1960

Source 2: National Oceanic and Atmospheric Administration (NOAA)

- *Monthly* precipitation at a 0.5-degree resolution from 1900 to 2017
- Data available only for land surface
- Accessible by online repository
- All years available as one file, 250MB, and is in NetCDF gridded format

Example: outline/pseudo-code for downloading UDel data

1. Create a directory to download the precipitation data into
2. Download file directly from URL
3. Transform the data into a convenient format and shape
4. Delete the original download to save space
5. Save this processed dataset to data/intermediate

Example: GMFD vs. UDel

Choose the most sensible data/methods/model/etc. for your research question and timeline!

- The analysis can be done with either dataset
- GMFD had higher resolution, at 30 times the frequency, and with global coverage
- UDel is much simpler but very quick to download (and considered higher quality)
- Make these choices for yourself! For this project, none of GMFD's advantages are relevant for my analysis so I am inclined to use the UDel data

Coding hygiene

- You need to be able read your own handwriting:
  - Comment as if future you will not remember anything you did
  - Break into sections (including visually, Stata users like to use lots of “%%%%” line breaks)
  - Write in a way that emphasizes the input-output nature of the process (a massive advantage of the ‘piping’ grammar of the tidyverse in R)
- At the same time, do not overexplain
- If there are multiple ways to accomplish the same thing, choose the one that is easy to understand from reading. Think of last week’s extreme example:
  1. gen rich = income >= median_income
  2. gen rich_woman = rich == 1 & gender == ‘female’
  3. gen rich_woman = !((!rich & gender == ’male’) | (rich & gender == ‘male’) | (!rich & gender == ‘female’)))

Anatomy of a project

Why care about project-oriented workflows?

1. Legibility
2. Diagnosis
3. Adaptability
4. Replication

Project-oriented workflows: legibility

Avoid having a project folder like this:

- analysis.do
- analysis2.do
- analysis_original.do
- regressions.do
- merge.do
- data_cleaning.do
- survey_data.dta
- survey_data.csv

Project-oriented workflows: diagnosis

- Workflow as an assembly line where data/input flows in one end and the results flow out the other
- Bespoke scripts perform specialized and distinct functions
- A great workflow isn’t just one that avoids mistakes but one that easily identifies and repairs the mistake
- Ideally, a single error can be isolated so that once identified in one script, it causes little to no disruption in the other scripts

Project-oriented workflows: adaptability

- Relatedly, a good workflow is one that is least disrupted when a change is made
- If I have a meeting with my advisor and she suggests removing some control variables or repeating the analysis on analternative dataset, I want these to be as simple to implement as possible
- If my model depends on a particular exogenous parametrization of, say, the savings rate or the separation rate elasticity, I want to only have to adjust one line per parameter at most even if it's used in a hundred different scripts
- Ideally, this even holds for figures and tables: they should update automatically without any manual input

Project-oriented workflows: replication

- If my project requires taking 500 bootstrap samples of a dataset and simulating a stochastic process over a hundred periods, the code should be able to produce the exact same samples and processes no matter
    - who is running the analysis
    - where or when they do it
    - what computer or operating system they are using
    - etc.
- Should be true of errors too! If you run into a coding issue and need help figuring out how to fix it, you want to be able to reproduce it

Setting up your workflow: the project folder

1. Create a project-specific folder and automate setting the working directory (in R this is an .RProject file)
2. Automate the setting of the working directory
   - Benefits: you don't have to start all your scripts setting your working directory; you don't have to edit your scripts if you ever move your project folder; reduces error and frustration; and one less thing to think about
   - In R, all paths will be relative to the .RProject file. A guide here.
   - In Stata, install the here package. Then you automatically have a macro called 'here' which refers to the project folder. A guide here.
3. Optional: make use of project profiles for commands you want run automatically whenever opening your project
   - For R, write a text file with the set of commands you want run. Name it .RProfile and save it to your project folder
   - Similarly, for Stata, write a profile.do script (though I'm not certain whether this can be done project-specifically)

Setting up your workflow: example folder organization 1

1. data
    1.1 input (contents should never be affected by workflow)
    1.2 intermediate
    1.3 output (processed files that can be used to produce the final figures and tables)
2. scripts
    2.1 DownloadData.do
    2.2 ProcessInputs.do
    2.3 MergeDatasets.do
    2.4 EstimatesTables.do
    2.5 Figures.do
3. results
    3.1 table.tex
    3.2 figure.pdf

Setting up your workflow: example folder organization 2

1. build
    1.1 input (contents should never be affected by workflow)
    1.2 code
        1.2.1 DownloadData.do
        1.2.2 ProcessInputs.do
        1.2.3 MergeDatasets.do
    1.3 output
        - final dataset(s)
2. analysis
    2.1 input
        - final dataset(s)
    2.2 code
        - EstimatesTables.do
    2.3 output
        - table.tex
        - figure.pdf

Project-oriented workflows: an applied example

Replicating Corno, Hildebrandt, and Voena (2020)

1. “Age of Marriage, Weather Shocks, and the Direction of Marriage Payments” (Econometrica)
2. Research question: What effect does drought have on the hazard into child marriage and early marriage?
3. CHV find that
    - “droughts *increase* the annual hazard into child marriage by 3% in Sub-Saharan Africa, while in India droughts *reduce* such a hazard by 4%”
    - The sign seems to depend on the direction of marriage payments

Replicating Corno, Hildebrandt, and Voena (2020)

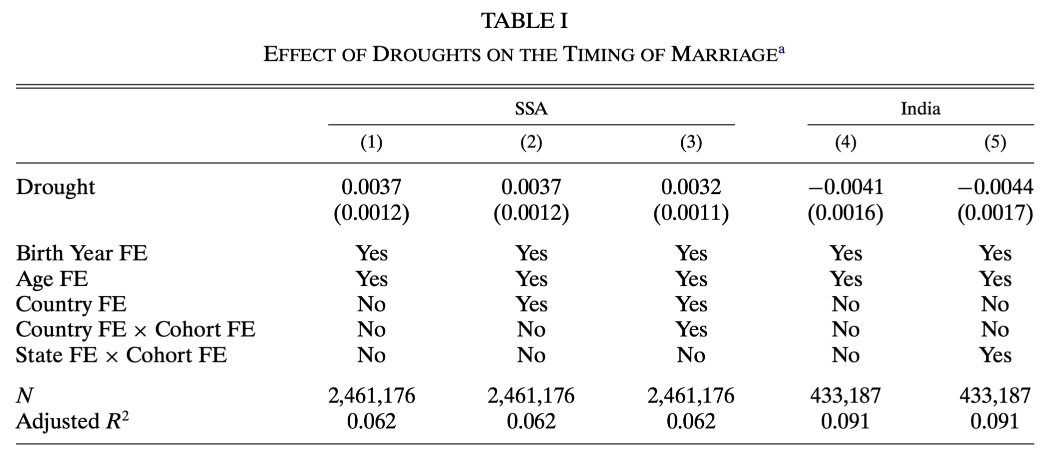
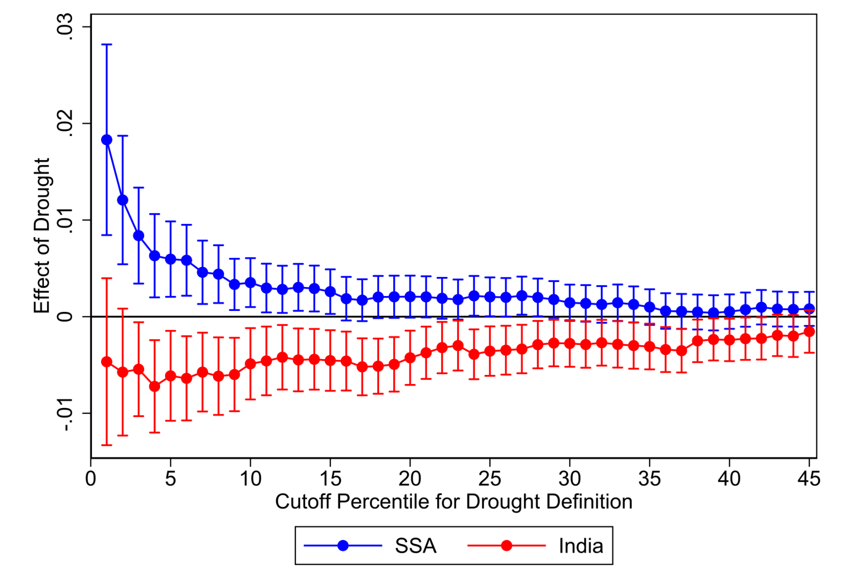


TABLE I

EFFECT OF DROUGHTS ON THE TIMING OF MARRIAGE[a]

	SSA			India	
	(1)	(2)	(3)	(4)	(5)
Drought	0.0037 (0.0012)	0.0037 (0.0012)	0.0032 (0.0011)	−0.0041 (0.0016)	−0.0044 (0.0017)
Birth Year FE	Yes	Yes	Yes	Yes	Yes
Age FE	Yes	Yes	Yes	Yes	Yes
Country FE	No	Yes	Yes	No	No
Country FE × Cohort FE	No	No	Yes	No	No
State FE × Cohort FE	No	No	No	No	Yes
N	2,461,176	2,461,176	2,461,176	433,187	433,187
Adjusted R^2	0.062	0.062	0.062	0.091	0.091

Replicating Corno, Hildebrandt, and Voena (2020)



Replicating Corno, Hildebrandt, and Voena (2020)

1. Replication contribution/differences:
    - The original study used 1999 Demographic and Health Survey (DHS) data for India
    - There have been several more comprehensive rounds of DHS surveys since then
    - Example: The 1999 India survey only interviewed 400,000 married women so conditions on future outcomes
    - The 2020 survey interviewed a representative sample of over 2 million single and married women

Project organization

- First: download DHS survey data
- scripts
  1. Download climate data
  2. Process
  3. Process climate data
  4.
- data
  1. downloads: climate data
  2. input: DHS survey data
  3. intermediate: processed survey data
  4. intermediate: processed climate data
  5. output: merged survey data

Corno, Hildebrandt, and Voena (2020)

1. "Age of Marriage, Weather Shocks, and the Direction of Marriage Payments" (Econometrica)
2. Research question: What effect does drought have on the hazard into child marriage and early marriage?
3. CHV find that
   - "droughts *increase* the annual hazard into child marriage by 3% in Sub-Saharan Africa, while in India droughts *reduce* such a hazard by 4%"
4. Replication contribution/differences:
   - The original study used 1999 Demographic and Health Survey (DHS) data for India
   - There have been several more comprehensive rounds of DHS surveys since then
   - The 1999 survey only interviewed 400,000 married women so conditions on future outcomes
   - The 2016 and 2020 survey interviewed a representative sample of over 2 million single and married women

Other research tips

Documentation and version control

- Task management and documenting research: Trello and Todoist
- Research updates with your advisor
    - Best to not have to run anything during a meeting: have whatever you want to discuss accessible before the meeting, preferably as a document
    - Notebooks (Jupyter and R Notebooks) are great for this
- Save frequently (especially Stata). Consider backing up to the cloud or a hard drive.
- More advanced version control: Git
- Include date produced in the file names

Automation

- Automate as much as possible!
- Automate the production of publication-quality figures and tables
  - R: fixest::etable, stargazer, ggplot2
  - Stata: outreg2, estout, esttab, orth_out
  - Regression tables to .tex
  - Figures to .pdf/.png/etc.
- Profiles: have certain commands automatically run every time you open R/Stata
  - R: .RProfile
  - Stata: profile.do

Computational power

- For computationally demanding tasks, a couple of related options
  - Parallelization (I believe you have to pay extra in Stata)
  - Using the university high-performance cluster
- Idea here is that a demanding task can be broken up into many identical tasks that can be run at the same time
- Probably overkill in most cases but I can help if needed