ECON-UN 4999 — Senior Honors Thesis Seminar

Workshop 1: Foundations of programming for economics research

Matthew Alampay Davis

October 9, 2023

Introduction

About me

- Grew up in Manila, Jakarta, Ho Chi Minh City
- Undergrad in New York and London
- Academia:
    - Research assistant at Stanford's Earth System Science department
    - Economics master's at Oxford
    - Economics PhD at Columbia
    - Happy to chat about any of these experiences in office hours
- Teaching:
    - Econometrics (x4)
    - Intermediate micro (x2)
    - The global economy

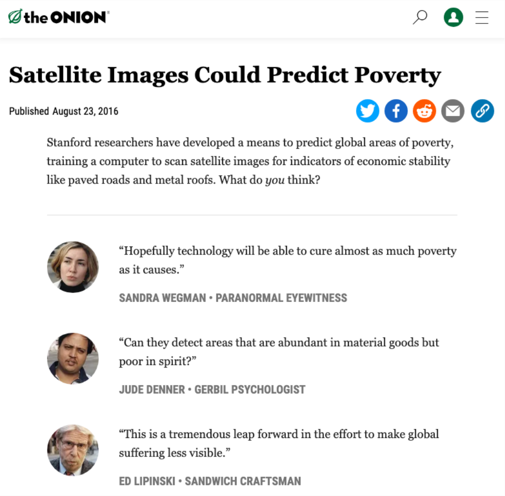
Research: development, environmental economics, political economy

Past work:

1. Using satellite imagery to locate poverty
2. Quantifying benefits of mitigating global warming

Dissertation:

1. Climate change and inequality
2. Weather shocks and political transitions
3. Cultural transmission of gendered violence and norms



the ONION

Satellite Images Could Predict Poverty

Published August 23, 2016

Stanford researchers have developed a means to predict global areas of poverty, training a computer to scan satellite images for indicators of economic stability like paved roads and metal roofs. What do *you* think?

"Hopefully technology will be able to cure almost as much poverty as it causes."

SANDRA WEGMAN • PARANORMAL EYEWITNESS

"Can they detect areas that are abundant in material goods but poor in spirit?"

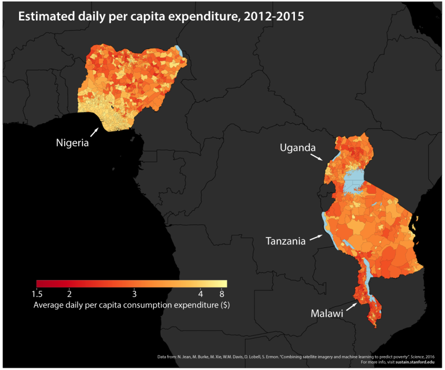
JUDE DENNER • GERBIL PSYCHOLOGIST

"This is a tremendous leap forward in the effort to make global suffering less visible."

ED LIPINSKI • SANDWICH CRAFTSMAN

Learning how to program: my experience

1. Quantitative topics courses
    - Macro-finance with Matlab
    - Time series econometrics with R
    - Quantitative economics and dynamic programming with Python
    - Part-time RA using Python
2. Apply to a bunch of research positions
3. Lie about how good I am at R
4. Move to the other side of the country
5. Spend eight hours a day Googling
6. Get published



These two workshops are for making you as self-sufficient as possible

Goal of these workshops

- Overview the most common and useful data operations
- Instill best practices for responsible and understandable data work
- Get a sense of what programming can do
- Most importantly, get out of your own way

Not the goal:

- Becoming fluent in a programming language
- Memorizing commands
- An exhaustive tutorial of the commands your project will need

Me as a resource

- Office hours
  - Rest of October (2 hours per week)
    Friday: 2:30-4:30 at **Hamilton 408**
  - Starting November (6 hours per week)
    Thursday: 1:15-3:15pm by Zoom
    Friday: 2:30-6:30pm at **IAB 1102**
- Reach out but no guarantees
  - Some flexibility if hours don't work
  - Emails for small things not worth a meeting
  - Recommendations for external resources or references
- Keeping running notes on what people are working on
- General resources periodically updated on my website: wmadavis.com/teaching

Office hours: before the meeting

Go in with a general plan:

1. Book two slots if you need (15 minutes go by fast!)
2. What is the problem/question?
3. Is there anything you want to show me?
4. What do you want to come away from our meeting with?

Anything for me to know or look at in advance?

- For example
  - Important context
  - What kind of problem/method/data, paper you're borrowing from?
  - What software or programming language you're using
- There are lots of things I don't know but can prepare for
- If so, provide by Calendly or email ahead of meeting

Plan for today

1. Reviewing the essentials of data analysis
2. Organization 1: Anatomy of a script

Reviewing the essentials of data analysis

Statistical software and programming languages

$tata: Very efficient at statistical operations on a single dataset

- **Stata**
  reg y x, robust
- **Compared to R**:
  library(fixest)
  model <- feols(y ∼ x, data = dataset, se = 'robust')
  etable(model)

"Object-oriented" programming languages (e.g., R, Python, Julia)

- Environment can contain any number of accessible 'objects': words, scalars, vectors, vectors of words, arrays, matrices, tables, graphs, models, estimates, functions, lists, lists of objects, …
- General purpose: multiple datasets, spatial data, text data, online data, machine learning, parallelized computing, creating documents, creating websites…

The basics of data processing: see R Notebook 01

1. Loading data
2. Summarizing data
3. Basic data transformations
4. Reshaping data
5. Statements, conditions, and loops

Organization 1: Anatomy of a script

First determine the purpose of your script and what you want it to do

- Working backwards
    - Outputs: what should you end up with?
    - Inputs: what data/objects/functions/packages are strictly required to produce them? Typically, these are all loaded at the top of a script in what's called the preamble
- The task of programming is to bridge the two as clearly and efficiently as possible and organize/comment on your script accordingly
- Pseudocode:
    - Break down the process from input to output into intuitive and discrete tasks
    - Describe what the task is in plain English
    - Translate that into real code and keep the plain English as a comment

Scripting: tips and pitfalls

- Make sure commands are written in sequence
    - If you are saving regression estimates in line 30, the command to run the regression should be somewhere between lines 1 and 29
    - Use the console for taste tests and experimenting and the script/do editor for recording your recipe
    - Beginners make the mistake of using scripts like scratch paper and risk being unable to retrace their steps
- Make sure the script is self-contained
    - Conditional on all the inputs having been generated or installed, the script should be able to run successfully even if you were to run it in a fresh Stata/R session

Scripting: tips and pitfalls

- Don’t do too many things in one script
    - Messy and hard to read
    - Easier to make mistakes
    - Harder to locate and fix mistakes
- Don’t do too different things in one script
    - For example, don’t estimate regressions in the same script that you’re processing the data
    - Instead, save your processed data at the ends of your data processing script(s) and load that processed data at the start of your analysis script(s)
    - More on this when we get to project-oriented workflows

Next week

1. Project-oriented workflows
2. From estimation to communication: tables and data visualization
3. Synthesizing example: following a project workflow from start to finish
4. More specific and advanced applications