

Lecture 2

Reproducibility

Pierre Biscaye
Université Clermont Auvergne

Data Science for Economics

Note: Materials for this lecture are drawn from Ted Miguel's
Development Economics course at UC Berkeley

Agenda

1. Overview of reproducibility and transparency
2. Organizing files
3. Organizing code
4. Coding transparency and portability
5. Writing code in Python
6. More Python basics

Key references

- Casey, Glennerster, and Miguel. (2012). “Reshaping Institutions: Evidence on Aid Impacts Using a Pre-analysis Plan”, *Quarterly Journal of Economics*, 127(4), 1755-1812.
- Miguel et al. (2014). “Promoting Transparency in Social Science Research”, *Science*, 10.1126/science.1245317.
- Christensen and Miguel. (2018). “Transparency, Reproducibility, and the Credibility of Economics Research”, *Journal of Economic Literature*, 56(3), 920-980.
- Ferguson et al. (2023). “Survey of open science practices and attitudes in the social sciences”, *Nature Communications*, 14.
- Christensen, Freese, and Miguel. (2019). *Transparent and Reproducible Social Science Research: How to Do Open Science*, University of California Press.

Threats to validity of research

- **Fraud:** undermines public trust in science
 - Open data and code can help uncover
- **Publication bias:** missing studies/ “file-drawer” problem
 - May lead to wasted research effort, misguided policy decisions
 - May incentivize author manipulation/“p-hacking”
 - Pre-registration can reduce scope for manipulation, promote publication
- **Failure to replicate:** within study (reproduction) and across settings (replication)
 - Increasing journal data posting requirements
 - Difficulty of getting funding or publishing replications of studies

What do transparency and reproducibility mean in economics research?

- **Transparency:**

- Ensuring that all data, methods, and analyses are openly shared and clearly documented, allowing others to understand and evaluate the research process.

- **Reproducibility:**

- The ability of others to replicate the results of a study using the original data and code provided by the researchers.
- Critical for validating findings, building trust, and advancing knowledge.

- **Benefits:**

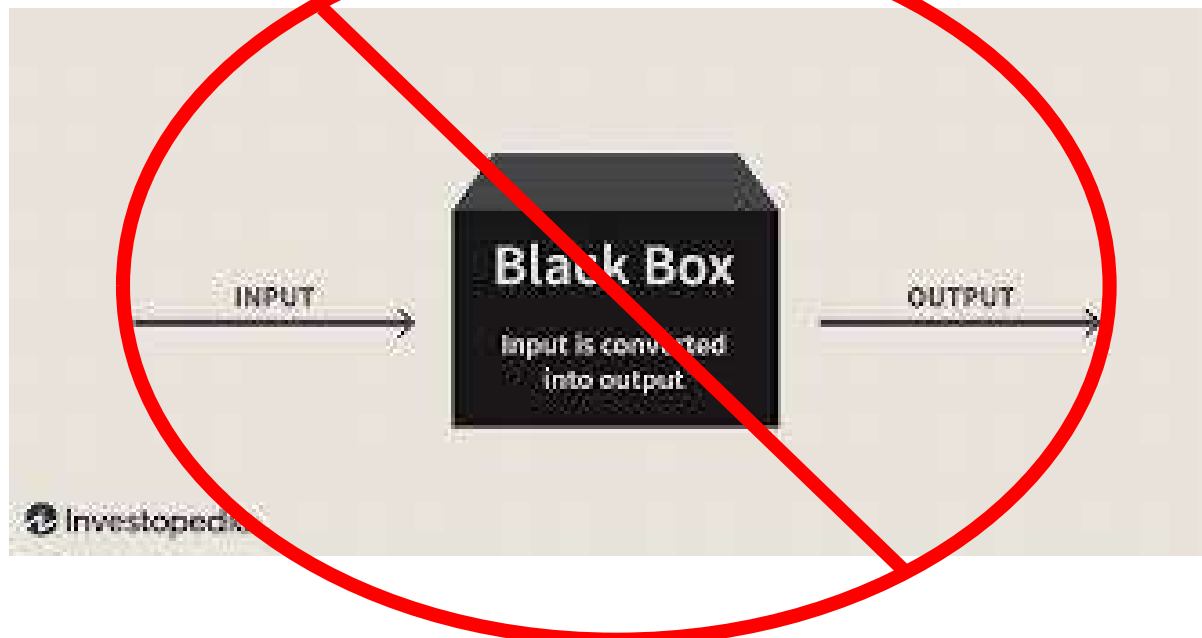
- Results that can be verified and shown to be largely free of investigator bias are more convincing.
- More publication of null or controversial results to broaden policy discourse.

Key principles of reproducibility

- **Accessible Data:** Provide well-documented, cleaned, and publicly available datasets (where ethically and legally possible).
- **Open Code:** Share analysis scripts and code in a version-controlled repository (e.g., GitHub) with clear instructions for execution.
- **Clear Methodology:** Document all steps in data collection, preprocessing, and analysis to ensure clarity and consistency.
- **Replication Workflow:** Design workflows that allow others to re-run analyses seamlessly, ensuring compatibility across systems and tools.

Takeaways for this course

- The quality and clarity of the research process matters
- Focus of this lecture: organizing and documenting research methods



Organizing files

The first step to a clear research process is file organization. There should be a clear structure to your folder hierarchy and file names.

```
README.pdf
data/
  raw/
    cps0001.dat
  analysis/
    combined_data.dta
    combined_data.csv
    combined_data_codebook.pdf
code/
  01_create/
    01_readcps.R
    02_readfred.R
  02_analysis/
    01_table1-5.R
    02_figures1-4.R
results/
  table1.tex
  table2.tex
  ...
  figure1.pdf
  figure2.pdf
```

[Source](#)

Example: Jigawa Floods Project

Basic structure:

1. Code
 1. High-frequency checks for data collection
 2. Analysis: separate scripts for different tasks
2. Data
 1. Raw
 2. Clean
3. Output
 1. Figures
 2. Maps
 3. Tables

Example: Folder of data sources

1. Sub-folders for main data sources/data types
2. Clear organization within folders
3. Example: LSMS-ISA
 1. Country sub-folders
 2. Year/round sub-folders
 3. Data, Questionnaires, Resources sub-folders
 4. Zipped raw data and unzipped folders
 1. Always keep a copy of the original data as a backup!
4. Readme documents
 1. Notes on where/when data were accessed
 2. Example: ARES data

<https://databank.illinois.edu/datasets/IDB-1107366>

Retrieved 2/14/23

1. ARES, crop-specific exposure to temperature and moisture shocks
 - a. 0.25 degree cells by year by crop
 - b. 1961-2014
 - c. /Users/pierrebiscave/Dropbox/Data/Spatial/ARES
 - d. .nc files with crop layers

File organization in this class

- Folders for each section
- Subfolders for data, images
- Separate Jupyter Notebooks for different topics
- Clear file naming conventions

Reproducibility of code

- Comment and document thoroughly
- Use modular design
- Adopt consistent naming conventions
- Version control
- Ensure code is portable

This is important for both your future self and for potential collaborators and reviewers!

Looking back at the code
you wrote last month...



Comment and document thoroughly

- Include clear comments in your code to explain the purpose of each section and the logic behind key steps.
- Best practice: Use a README file to provide an overview of the project, dependencies, and how to run the code.

Looking at code you wrote more than 6 months ago:



[Source](#)

Commented code

```
14 //
15 // Prep spatial datasets
16 //
17
18 * Locust Swarms
19 {
20   * Data from FAO Locust Hub, coordinates and dates of locust swarm observations globally from 1985-2023
21   * Some additional detail included there but lots of missing data so focus just on swarm location and timing
22   import delimited "$data/Locust Hub/Retrieved 2.13.23/Swarms.csv", encoding(UTF-8) clear
23   drop if objectid==.
24   drop if missing(x) | missing(y)
25   destring x, replace
26   destring y, replace
27   gen double lat=round((y+0.125)*4)/4 - 0.125
28   gen double lon=round((x+0.125)*4)/4 - 0.125
29   gen date = date(substr(startdate,1,10), "YMD")
30   format date %td
31   gen year=year(date)
32   gen month=month(date)
33
34   * Output for mapping
35   preserve
36   keep x y year
37   export delimited "$clean/mapping_swarms.csv", replace
38   restore
39
40   * Match to countries, identify countries in Africa and Arabian Peninsula with at least 10 swarms in analysis period
41   1996-2018
42   geoinpoly y x using "$data/Country boundaries/Country raw/UIA_World_Countries_Boundaries/WORLD_coor.dta"
43   merge m:1 _ID using "$data/Country boundaries/Country raw/UIA_World_Countries_Boundaries/WORLD_data.dta", keep(1 3)
44   gen swarm=1 if year>1995 & year<2019
45   egen swarms_1996_2018=sum(swarm),by(COUNTRY)
46   bys COUNTRY: gen first=_n==1
47   br COUNTRY swarms_1996_2018 if swarms_1996_2018>10 & first==1
48   * Well over 10: Algeria, Burkina Faso, Chad, Egypt, Eritrea, Ethiopia, Gambia, Guinea, Guinea-Bissau, Libya, Mali,
49   Mauritania, Morocco/Western Sahara, Niger, Oman, Saudi Arabia, Senegal, Somalia, Sudan, Tunisia, Yemen
50   * Cabo Verde 42, India 127, Iran 19, Israel 11, Kenya 22, Pakistan 108 also meet the swarm count criteria but outside
51   area of interest or are borderline cases (Israel, Kenya)
52
53   * Set target geographic area to trim other datasets, based on countries to target
54   drop if lat<-2.5
55   drop if lat>37.5
56   drop if lon<-17.5
57   drop if lon>60.25
```

Use modular design

- Break the code into smaller, reusable functions or scripts to make it easier to debug, update, and understand. Separate data cleaning, analysis, and visualization steps logically.

Dropbox > Kenya Labor Supply > do files >					Rechercher dans : do files	
	Nom	Modifié le	Type	Taille		
	.ipynb_checkpoints	11/20/2024 1:16 PM	File folder			
	Archive	11/20/2024 1:16 PM	File folder			
ments	1_Data Merge	4/12/2024 12:16 AM	DO File	12 Ko		
s	2_Merged Data Prep	6/25/2024 1:19 AM	DO File	24 Ko		
	3_Paper Figures	5/10/2024 7:56 AM	DO File	51 Ko		
	4_Paper Regressions	9/13/2023 8:29 PM	DO File	72 Ko		

All files / GridWatch Master Folder / Replication

code ⚙

🕒 Recents

☆ Starred

Name ↑

Archive

cleaning

.Rhistory

Balance_reg_fe.do

Balance_reg_iv.do

Balance_reg.do

copy_data.do

figures.do

MapSites_TC.R

prep_survey_data.do

reg_table.do

tables.do

Modular design within scripts

```
3_Paper Figures x
18 // Line Graphs - Kenya COVID-19 Cases
19 // Line Graphs - Kenya COVID-19 Cases
20 // Line Graphs - Kenya COVID-19 Cases
21 // Line Graphs - Kenya COVID-19 Cases
22 {
23 // Line Graphs - Kenya COVID-19 Cases
24 // Line Graphs - Kenya COVID-19 Cases
25 // Line Graphs - Kenya COVID-19 Cases
26 // Line Graphs - Kenya COVID-19 Cases
27 // Line Graphs - Kenya COVID-19 Cases
28 // Line Graphs - Kenya COVID-19 Cases
29 // Line Graphs - Kenya COVID-19 Cases
30 // Line Graphs - Kenya COVID-19 Cases
31 // Line Graphs - Kenya COVID-19 Cases
32 // Line Graphs - Kenya COVID-19 Cases
33 // Line Graphs - Kenya COVID-19 Cases
34 // Line Graphs - Kenya COVID-19 Cases
35 // Line Graphs - Kenya COVID-19 Cases
36 // Line Graphs - Kenya COVID-19 Cases
37 // Line Graphs - Kenya COVID-19 Cases
38 // Line Graphs - Kenya COVID-19 Cases
39 // Line Graphs - Kenya COVID-19 Cases
40 // Line Graphs - Kenya COVID-19 Cases
41 // Line Graphs - Kenya COVID-19 Cases
42 // Line Graphs - Kenya COVID-19 Cases
43 // Line Graphs - Kenya COVID-19 Cases
44 // Line Graphs - Kenya COVID-19 Cases
45 // Line Graphs - Kenya COVID-19 Cases
46 // Line Graphs - Kenya COVID-19 Cases
47 // Line Graphs - Kenya COVID-19 Cases
48 // Line Graphs - Kenya COVID-19 Cases
49 // Line Graphs - Kenya COVID-19 Cases
50 // Line Graphs - Kenya COVID-19 Cases
51 // Line Graphs - Kenya COVID-19 Cases
52 // Line Graphs - Kenya COVID-19 Cases
53 // Line Graphs - Kenya COVID-19 Cases
54 // Line Graphs - Kenya COVID-19 Cases
55 // Line Graphs - Kenya COVID-19 Cases
56 // Line Graphs - Kenya COVID-19 Cases
57 // Line Graphs - Kenya COVID-19 Cases
58 // Line Graphs - Kenya COVID-19 Cases
59 // Line Graphs - Kenya COVID-19 Cases
60 // Line Graphs - Kenya COVID-19 Cases
61 // Line Graphs - Kenya COVID-19 Cases
62 // Line Graphs - Kenya COVID-19 Cases
63 // Line Graphs - Kenya COVID-19 Cases
64 // Line Graphs - Kenya COVID-19 Cases
65 // Line Graphs - Kenya COVID-19 Cases
66 // Line Graphs - Kenya COVID-19 Cases
67 // Line Graphs - Kenya COVID-19 Cases
68 // Line Graphs - Kenya COVID-19 Cases
69 // Line Graphs - Kenya COVID-19 Cases
70 // Line Graphs - Kenya COVID-19 Cases
71 // Line Graphs - Kenya COVID-19 Cases
72 // Line Graphs - Kenya COVID-19 Cases
73 // Line Graphs - Kenya COVID-19 Cases
74 // Line Graphs - Kenya COVID-19 Cases
75 // Line Graphs - Kenya COVID-19 Cases
76 // Line Graphs - Kenya COVID-19 Cases
77 // Line Graphs - Kenya COVID-19 Cases
78 // Line Graphs - Kenya COVID-19 Cases
79 // Line Graphs - Kenya COVID-19 Cases
80 // Line Graphs - Kenya COVID-19 Cases
81 // Line Graphs - Kenya COVID-19 Cases
82 // Line Graphs - Kenya COVID-19 Cases
83 // Line Graphs - Kenya COVID-19 Cases
84 // Line Graphs - Kenya COVID-19 Cases
85 // Line Graphs - Kenya COVID-19 Cases
86 // Line Graphs - Kenya COVID-19 Cases
87 // Line Graphs - Kenya COVID-19 Cases
88 // Line Graphs - Kenya COVID-19 Cases
89 // Line Graphs - Kenya COVID-19 Cases
90 // Line Graphs - Kenya COVID-19 Cases
91 // Line Graphs - Kenya COVID-19 Cases
92 // Line Graphs - Kenya COVID-19 Cases
93 // Line Graphs - Kenya COVID-19 Cases
94 // Line Graphs - Kenya COVID-19 Cases
95 // Line Graphs - Kenya COVID-19 Cases
96 // Line Graphs - Kenya COVID-19 Cases
97 // Line Graphs - Kenya COVID-19 Cases
98 // Line Graphs - Kenya COVID-19 Cases
99 // Line Graphs - Kenya COVID-19 Cases
100 // Line Graphs - Kenya COVID-19 Cases
```

jupyter Section1a_Jupyter Notebook Last Checkpoint: 16 days ago

File Edit View Run Kernel Settings Help

Markdown

Practice

[]: # Please add a cell above here

[]: # Please add a cell below here

[]: # Please delete this cell

[]: ### Please change this code cell to a Markdown cell

Please change this Markdown cell to a code cell

[]: # copy this cell and paste below

[]: # cut this cell and paste it here

[]: # Please run this cell

```
a = a+1 # Adding 1 to a
print(a)
```

[]: # Please split this cell after this line

[]: # Please split this cell

[]: Please toggle comment on this line

[]: Please toggle comment on this line and this line

Use clear headings and labels to organize your code

Jupyter Notebook is built to be modular by default

Adopt consistent naming conventions

- Use meaningful, consistent names for variables, functions, and files to enhance readability and reduce confusion.
- For example, use `clean_data()` instead of `cd()` for function names.

NAMING VARIABLES



Giving them meaningful names, according to their use.



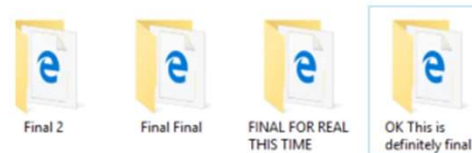
Giving them the most compact names possible, for less storage usage.



Giving them random names like "ahshjdn" or "yeetus".

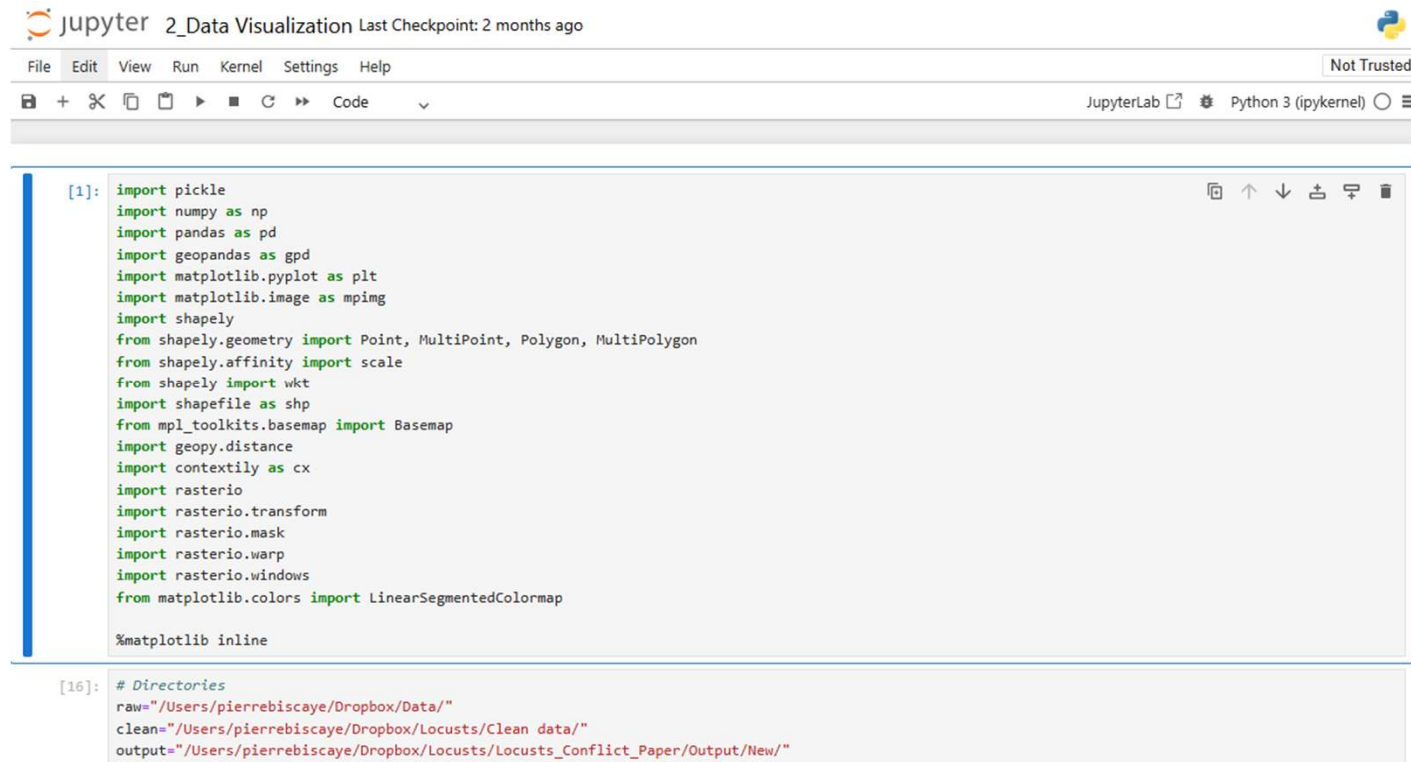
Version control

- Use version control systems to track changes in your code and collaborate efficiently.
- Commit changes with descriptive messages and maintain a well-organized repository structure.
- Gold standard: GitHub repository
- Minimum: clear files names indicating version history, well-structured archive folder
 - Back up your code, don't just always overwrite



Ensure code is portable

- Avoid hardcoding file paths or machine-specific dependencies.
- Use relative paths and specify software environments to ensure others can run the code seamlessly.
- Often useful to have a “0_setup script” with paths and packages to run first.



```
[1]: import pickle
import numpy as np
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import matplotlib.image as mimg
import shapely
from shapely.geometry import Point, MultiPoint, Polygon, MultiPolygon
from shapely.affinity import scale
from shapely import wkt
import shapefile as shp
from mpl_toolkits.basemap import Basemap
import geopy.distance
import contextily as cx
import rasterio
import rasterio.transform
import rasterio.mask
import rasterio.warp
import rasterio.windows
from matplotlib.colors import LinearSegmentedColormap

%matplotlib inline

[16]: # Directories
raw="/Users/pierrebiscaye/Dropbox/Data/"
clean="/Users/pierrebiscaye/Dropbox/Locusts/Clean data/"
output="/Users/pierrebiscaye/Dropbox/Locusts/Locusts_Conflict_Paper/Output/New/"
```

Writing Python code

- Into Jupyter Notebooks!