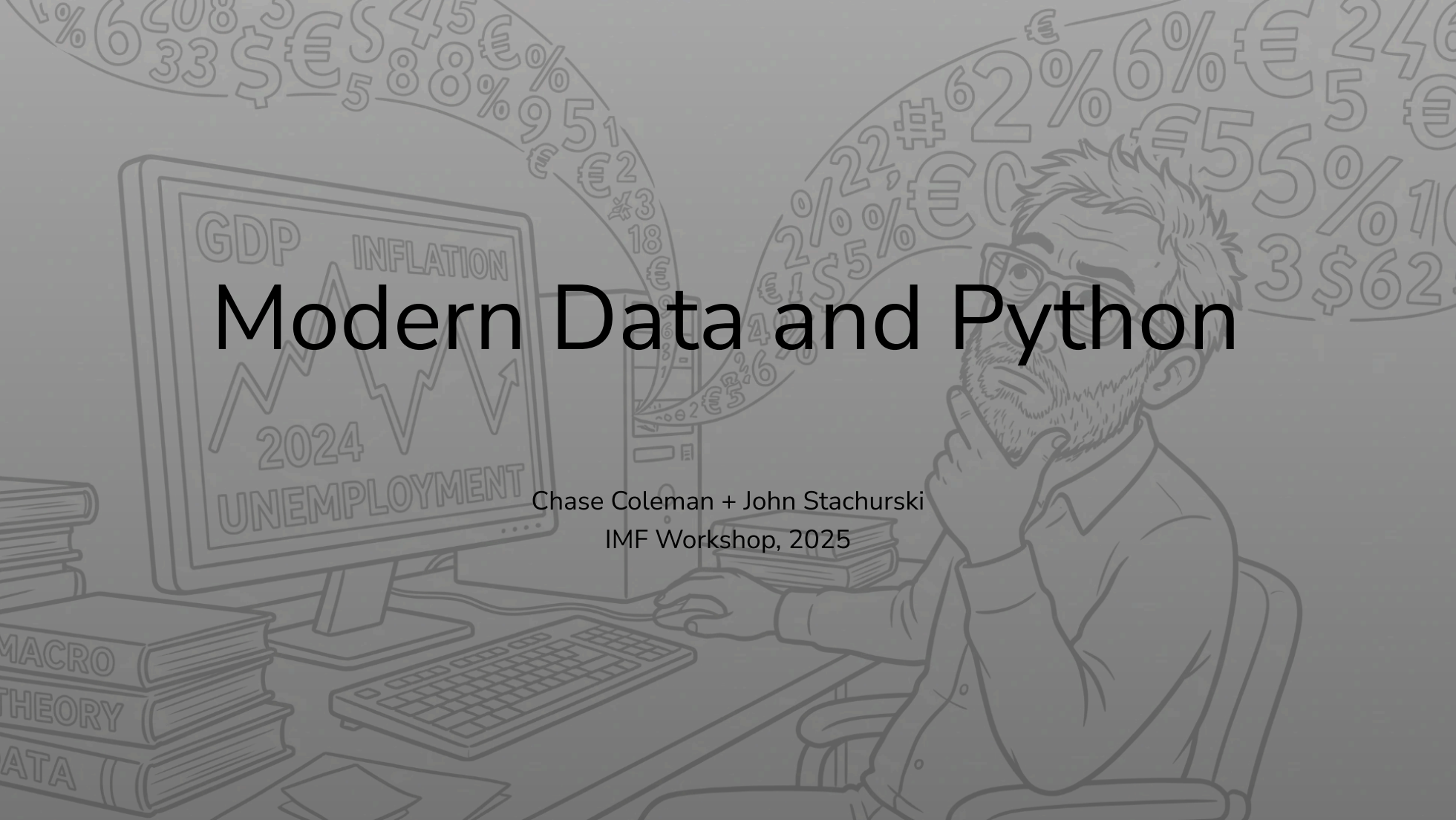


Modern Data and Python

Chase Coleman + John Stachurski

IMF Workshop, 2025

Commonly used data tools

Raise your hand if you have used one of these tools for "data work" sometime this year

- 🦆 DuckDB
- 💾 Microsoft excel
- 🐍 Python
- ❄️ R
- 🔨 Stata
- 🧑‍💻 SQL

Each of these tools is a reasonable choice to make for certain types of problems but today I'm going to try to convince you that (1) if you didn't raise your hand for Python that you should be using Python some of the time and (2) that you could benefit from replacing some of existing workflows with Python

A world full of data

The data revolution

In economics we've historically used relatively "small" amounts of data.

If you think about the "core" economic datasets (GDP, unemployment, imports, etc...), they are typically measured at a monthly/quarterly/annual frequency which means we only have, at best, 500-1,000 useful data points.

Even survey data, like the Current Population Survey in the US, only collects data for ~60,000 households per month which means that with 10 years of data we have ~7,000,000 total household observations.

For many problems, having 7,000,000 observations is now considered to be "relatively small data". As the amount of data produced in the world expands, we need to be able to use tools that can keep up with the ever changing state of the world.

Alternative data

One of the trends that is driving the expansion of data produced is "alternative data"

> Alternative data refers to non-traditional data sets used by investors, financial institutions, and businesses to gain a competitive edge and develop unique insights before they are reflected in traditional sources like company filings, press releases, or official economic statistics. – Gemini

This type of data can provide more frequent, even real-time, snapshots into broader economic conditions which can empower policy makers or investors to make more informed decisions about how to react

Examples:

- Social media posts
- Mobile app usage
- Job postings
- Credit card transactions
- Satellite images

Python "data stack"

Today we will learn about a few key packages for Python data work

`pandas`

`pandas` is the foundational library for data manipulation and analysis in Python.

- Originally created by Wes McKinney in 2008 while working at AQR.
- `pandas` comes from `pan` el `da` ta.
- The development of `pandas` is a big reason why Python has pulled ahead over the last 20 years for data work.

`matplotlib`

`matplotlib` is a comprehensive library for creating static, animated, and interactive visualizations in Python.

- Originally written by John Hunter in the early 2000s with the original motivation being to provide a plotting tool that could emulate the plotting features of MATLAB in Python.
- Provides both a relatively high level interface for making visualizations but it also has an incredibly flexible low level interface that allows users to customize every aspect of a visualization.

`polars`

`polars` is meant to be an alternative to `pandas` and is built with the idea of "this is how we would have build `pandas` if we had the benefit of 10 years of hindsight.

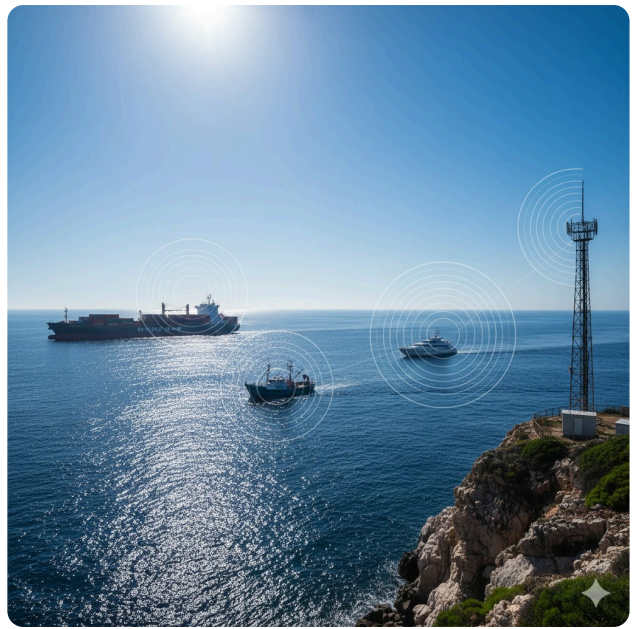
- Originally began in 2020 as a pet project by Ritchie Vink.
- Underlying functionality is all written in Rust for performance reasons.
- It is smart about leveraging multiple cores when possible and also takes advantage of lazy evaluation rather than eager evaluation (like `pandas`)

Data for our exercises

We will be working with some alternative data today – We will be examining vessel transmitter (AIS) data.

Three reasons for this:

1. I have wanted an excuse to explore and play with this data.
2. After choosing the data, I felt extra good about the choice because Andras has a great paper that uses this data.
3. Somewhere in the "multiverse" exists a universe where John and me are out sailing the high seas together instead of giving this workshop today.



AIS data

The AIS data contains a few columns that we'll care most about today:

#	Field Name	Description	Example	Unit
1	mmsi	Maritime Mobile Service Identity value	477220100	integer
2	base_date_time	Full UTC date and time	2017-02-01T05:02	-
3	longitude	Longitude	-71.04182	decimal degree
4	latitude	Latitude	42.35137	decimal degree
5	sog	Speed Over Ground	5.9	knot
6	cog	Course Over Ground	47.5	degree NAz
11	vessel_type	Vessel type as defined in NAIS specifications	70	scalar


"Proof" of another Chase+John sailing the seas

The plan

The plan for this morning is as follows:

1. Learn the basics of `pandas` and `matplotlib`
2. Apply what we learned to the AIS data
3. Learn how to do split-apply-combine and merging in `pandas`
4. Apply what we learned to the AIS data
5. See a failure mode of `pandas` and how `polars` solves it
6. Learn unique pieces of `polars` syntax
7. Apply what we learned to the AIS data