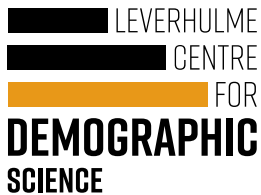


Why Python for (Open) Data Science?

Charlie Rahal

LCDS and Nuffield College, University of Oxford

A ten-minute talk at OxFOS, 6th March 2025



In the spirit of 'Open Scholarship', slides today are available at:

crahal.com/files/why_python.pdf

*"If I have seen further, it is by standing on the
shoulders of giants."*

Source: Isaac Newton to Robert Hooke, in February 1676

Can anyone guess what this code does?

guess_what_this_does.py

```
import json, io, imageio, requests
from PIL import Image

def get_cat_bounded(url):
    api_ret = requests.get(url)
    if api_ret.status_code==200:
        text = json.loads(api_ret.text)
        while (text[0]['width']<=400 or text[0]['width']>=600) or \
            (text[0]['height']<=400 or text[0]['height']>=600):
            api_ret = requests.get(url)
            if api_ret.status_code==200:
                text = json.loads(api_ret.text)
        r = requests.get(api_text[0]['url'])
        return Image.open(io.BytesIO(r.content))

url = 'https://api.thecatapi.com/v1/images/search'
images = []
for _ in range(0, 50):
    images.append(get_cat_bounded(url))
imageio.mimsave('CATS.gif', images, fps=1)
```

Thats right: Cats!

Why show you this? Well, Python is great at lots of useful things:

1. Scientific Computing (SciPy, numpy)
2. Data visualisation (Matplotlib, seaborn)
3. Web scraping (requests)
4. Data wrangling (Pandas)

A natural choice for un/semi-structured data. Python also excels at NLP/ML, but maybe a little less with classical statistics.

The rich ecosystem is just one reason for using Python.

Lets give some more reasons!

1. Intuitive syntax accelerates implementation.
2. Strong community fosters collaboration/innovation.
3. Extensible modules integrate diverse tools.
4. Efficient handling of large-scale data processing.
5. Advanced visualization reveals complex insights.
6. Proven support for ML/AI research.
7. Rapid prototyping expedites development.
8. Immense cross-platform compatibility.

How about some technical reasons?

1. Dynamic typing with flexible data structures.
2. Robust GC supports efficient memory management.
3. Native C API enables efficient module extensions.
4. JIT compilers (e.g., Numba) optimize runtime.

- Lets perhaps see two examples of how Python can be used for 'Open' data science:
 1. An 'Open' approach to statistical modeling
 2. An 'Open' approach to algorithmic uncertainty.
- Both have strong relations to 'Open Scholarship', and explicate Python's immense utility.

- Lets perhaps see two examples of how Python can be used for 'Open' data science:
 1. An 'Open' approach to statistical modeling
 2. An 'Open' approach to algorithmic uncertainty.
- Both have strong relations to 'Open Scholarship', and explicate Python's immense utility.
- This is OxFOS, after all!

Open Modeling

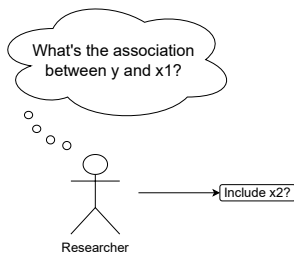
- There are close links between ‘Open Science’ and traditional modeling paradigms!
- By **opening** research, we can examine the certainty of statistical results.
- Does β_1 really equal 7.12452?

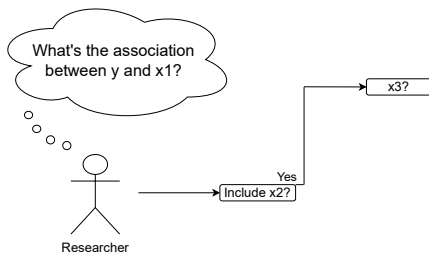
Open Modeling

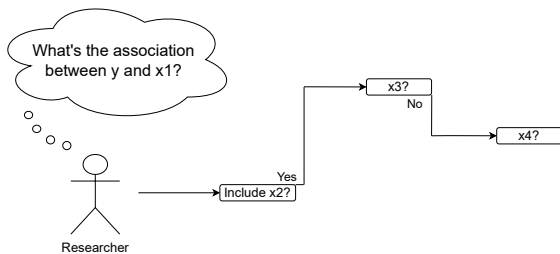
- There are close links between ‘Open Science’ and traditional modeling paradigms!
- By **opening** research, we can examine the certainty of statistical results.
- Does β_1 really equal 7.12452?
- How can we "take the ‘con’ out of econometrics"?

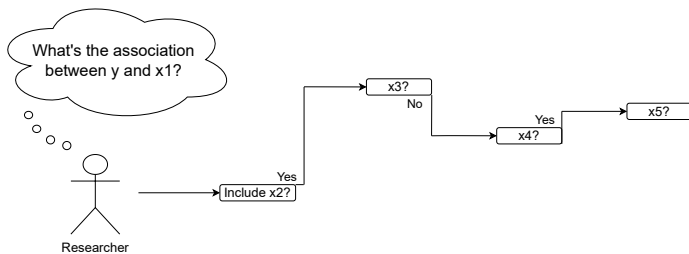
Open Modeling

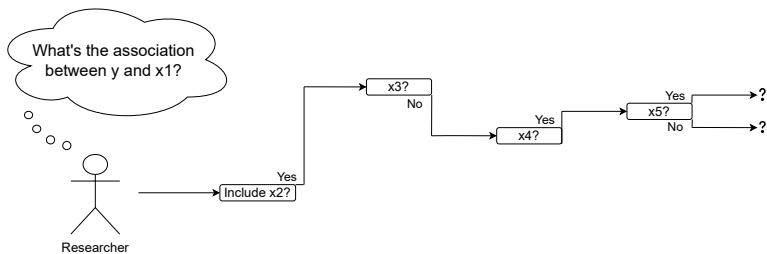
- There are close links between ‘Open Science’ and traditional modeling paradigms!
- By **opening** research, we can examine the certainty of statistical results.
- Does β_1 really equal 7.12452?
- How can we "take the ‘con’ out of econometrics"?
- Python to the rescue!

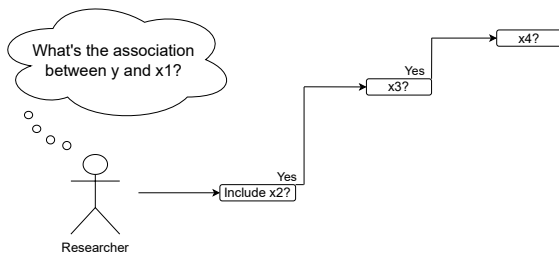


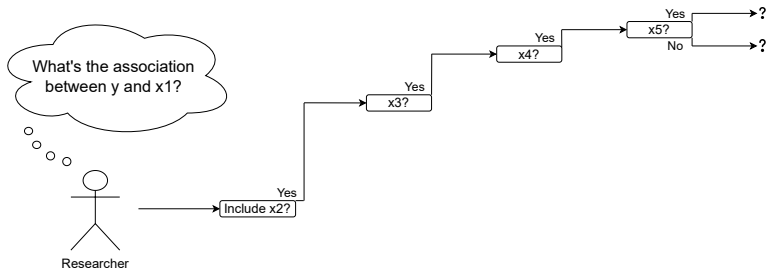


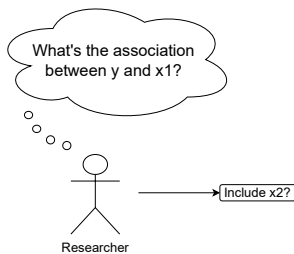


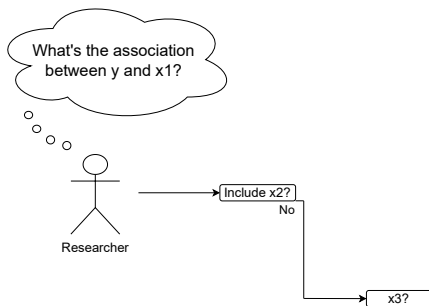


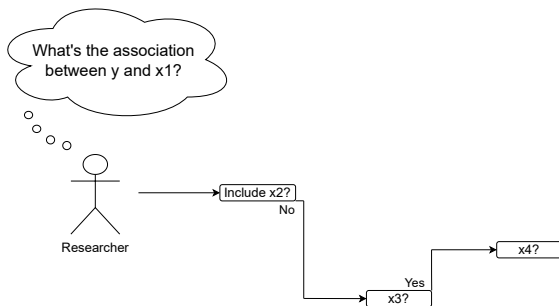


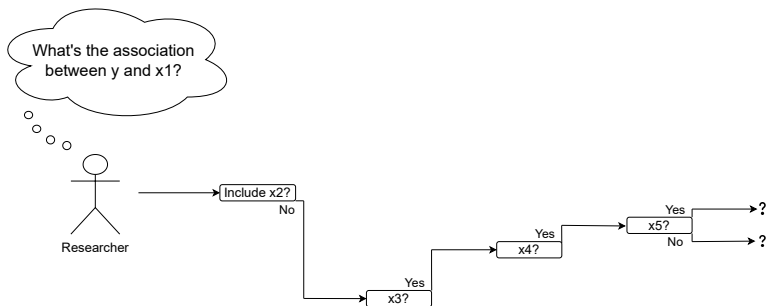


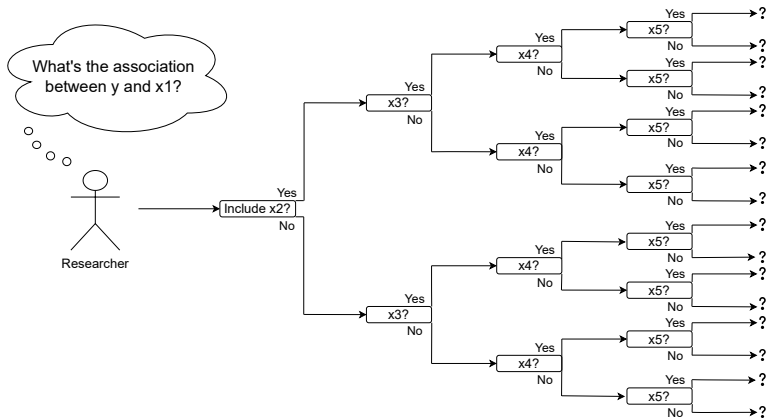




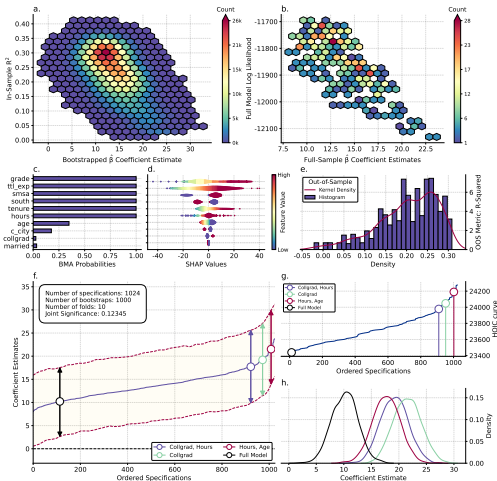








Python allows us to make tools to combat problems like this!



1. Make modelling uncertainty more transparent!
2. Incorporate all multiverse tools in one place!
3. Allow researchers to do more rigorous quantitative research!
4. Allows extensibility and intuitive to use.

From: 'Introducing RobustPy'. With Daniel Valdenegro Ibarra and Jiani Yan



A simple, minimal website for RobustiPy!

[View the Project on GitHub here!](#)

[View the Python Documentation here!](#)

[View the RobustiPy working paper here!](#)

Welcome to the RobustiPy homepage!

RobustiPy is an efficient multiversal library with model selection, averaging, resampling and out-of-sample analysis. It analyses various output spaces, in addition to the control variable space (e.g. multiple dependent variables, estimands of interest, etc). Developed for Python and R, it is designed to be both accessible and computationally efficient.

Note: This site (and project!) are a work in progress!

Installation

Installation is as simple (in Python) as:

```
pip install robustipy
```

and – potentially in the future for R – as:

```
install.packages("robustipy")
```

Examples: A series of toy examples with which might be useful to get you up and running can be found [here](#). A series of empirical examples which might be useful can be found [here](#).

Hackathon: We are holding a hackathon regarding RobustiPy in Oxford during June 2024. Participants can click [here](#) for further information on the day's activities!

Inspiration: We found great inspiration in the following three papers, which you might also enjoy reading:

Open Algorithms

- Python is **really** good at expressing how models work.
- It allows us to consider – for example – reliance on PRNGs .
- PRNGs occur everywhere, often ‘under the hood’ of software.
- As Python is so open/modular, we can take things apart easily.

(PRNGs have an enormous, hidden effect on scientific results)

- Lets actually see an example of why this is a problem!
- And how we can solve it?

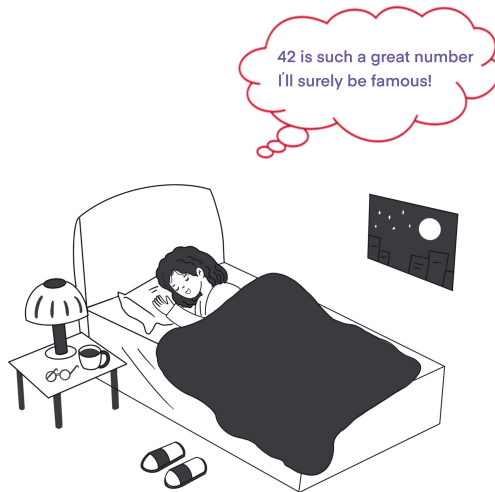
Open Algorithms

- Python is **really** good at expressing how models work.
- It allows us to consider – for example – reliance on PRNGs .
- PRNGs occur everywhere, often ‘under the hood’ of software.
- As Python is so open/modular, we can take things apart easily.

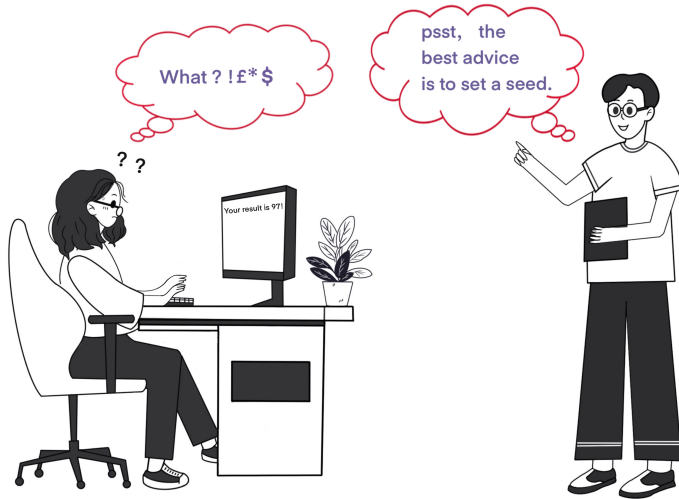
(PRNGs have an enormous, hidden effect on scientific results)

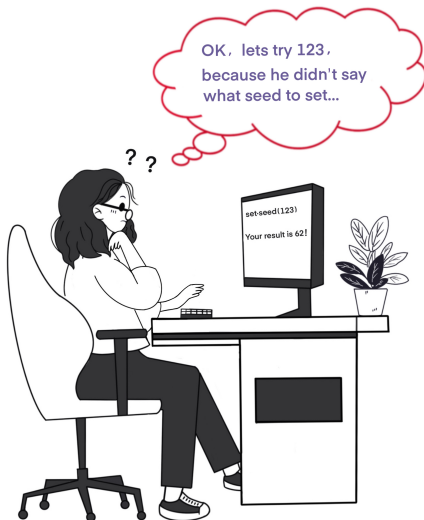
- Lets actually see an example of why this is a problem!
- And how we can solve it? [Python to the rescue!](#)

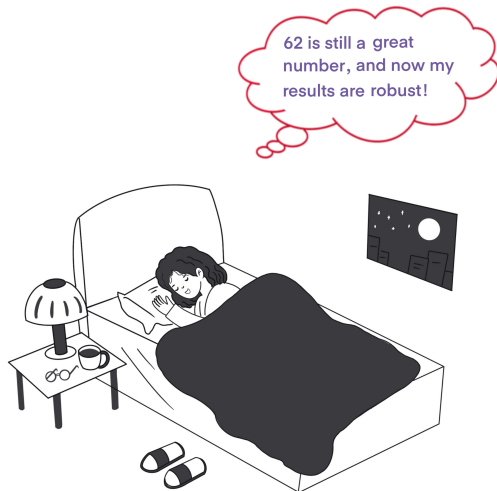








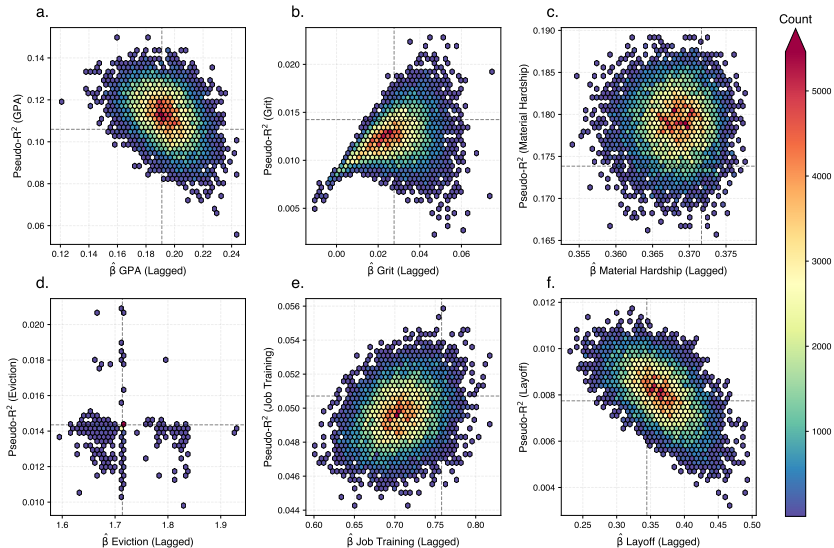




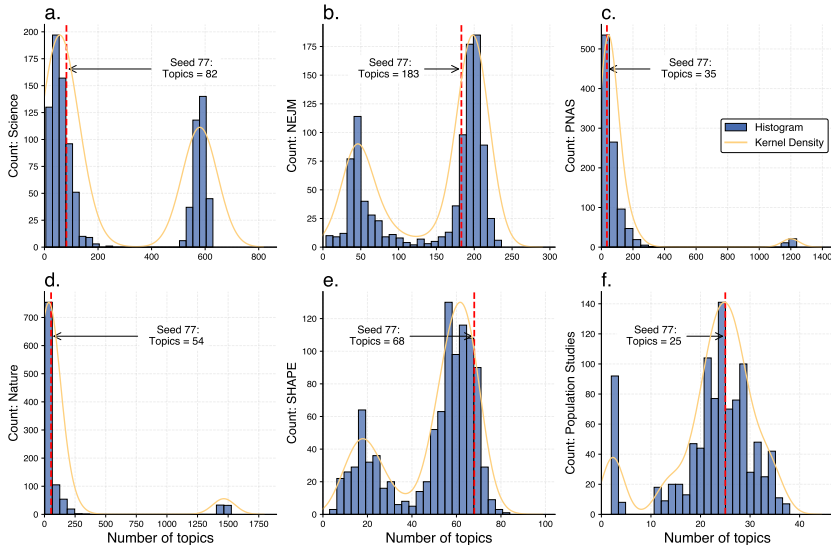








‘From a Seed of Doubt Grows a Forest of Uncertainty’
With Arun Frey, Jiani Yan and Mark Verhagen



‘From a Seed of Doubt Grows a Forest of Uncertainty’
 With Arun Frey, Jiani Yan and Mark Verhagen

- Python uniquely allows interrogation of existing designs.
- Allows unpacking the core of data science.
- It does this in a way that's easy, intuitive, and accessible.
- It also scales really well for computational problems like this.
 - The GIL is being removed in future versions!

- Python uniquely allows interrogation of existing designs.
- Allows unpacking the core of data science.
- It does this in a way that's easy, intuitive, and accessible.
- It also scales really well for computational problems like this.
 - The GIL is being removed in future versions!
- It also – of course – has millions of other use cases which we don't have time to talk about today, such is its versatility!

Conclusion

I hope I've been able confer my excitement, enthusiasm and
inspiration for Python!

If just one more 'open' paper, program, design, or estimator enters
scientific record because of using Python as opposed to a closed
software, everything here will have been worth while 😊.

Thank you for attending!
Thank you to my collaborators!
Don't hesitate to get in touch at:
charles.rahall@demography.ox.ac.uk
and
github.com/crahall!