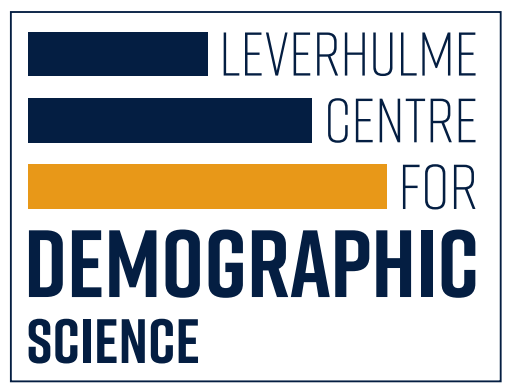
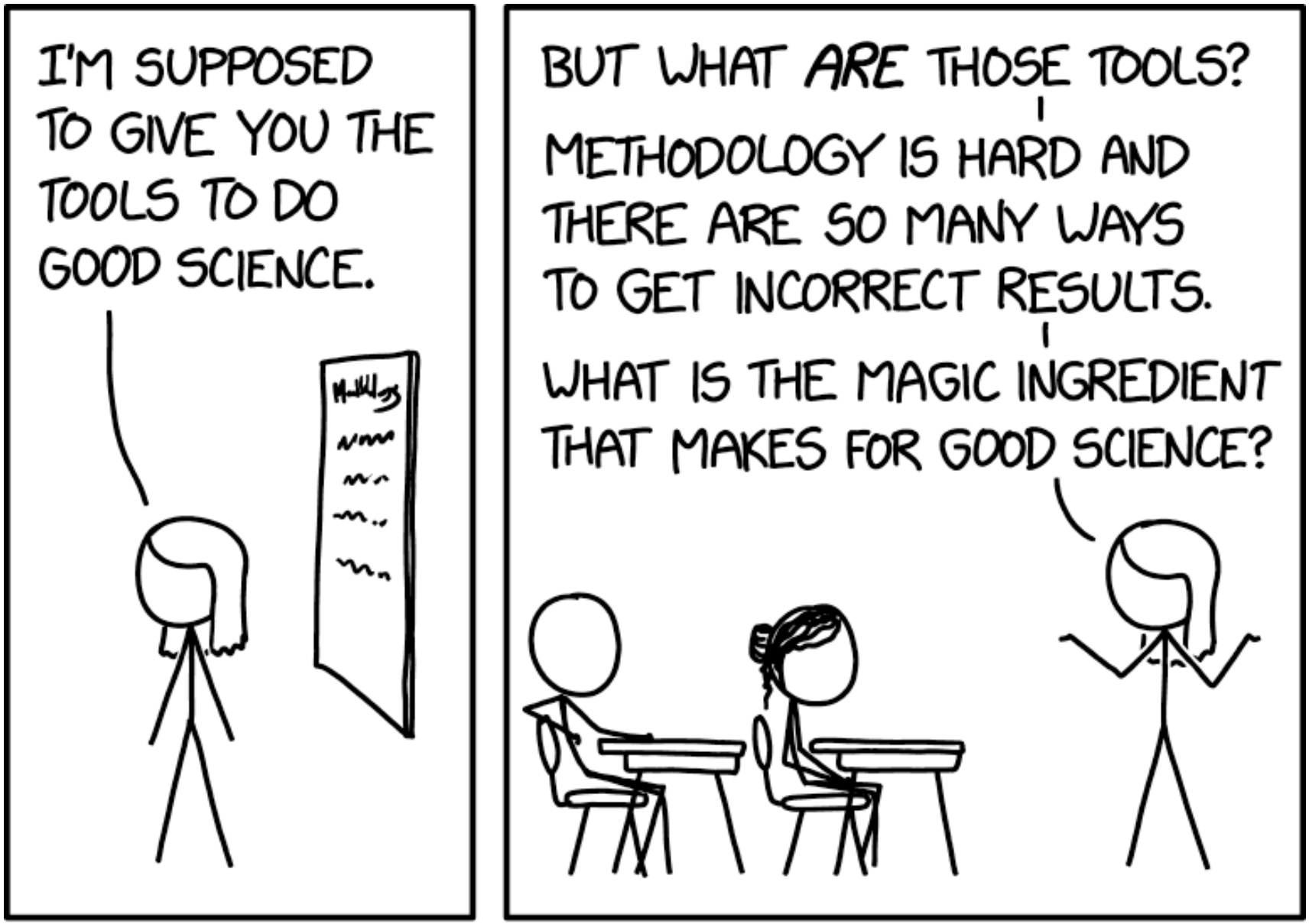
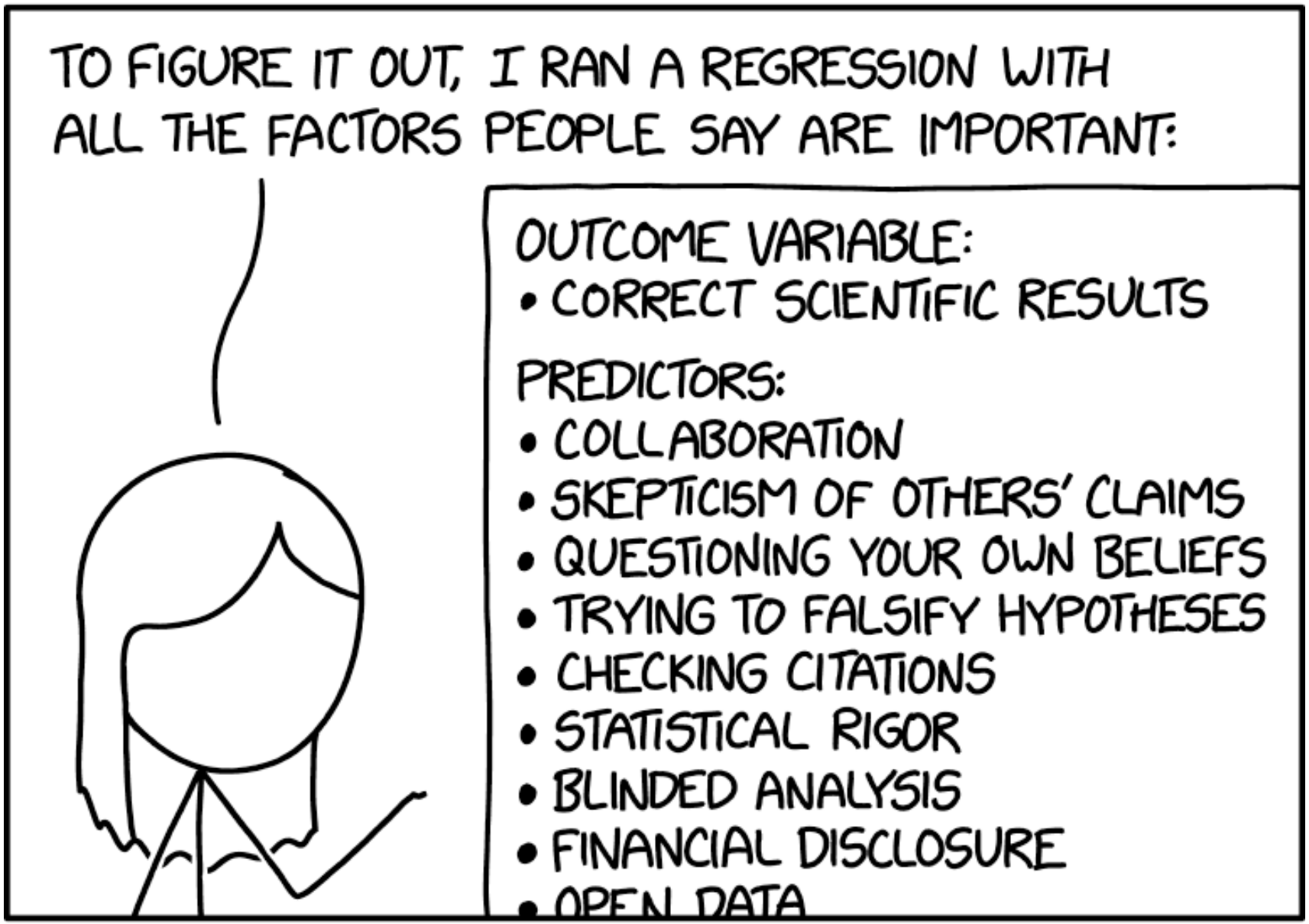
RobustiPy is finished!
A visual celebration

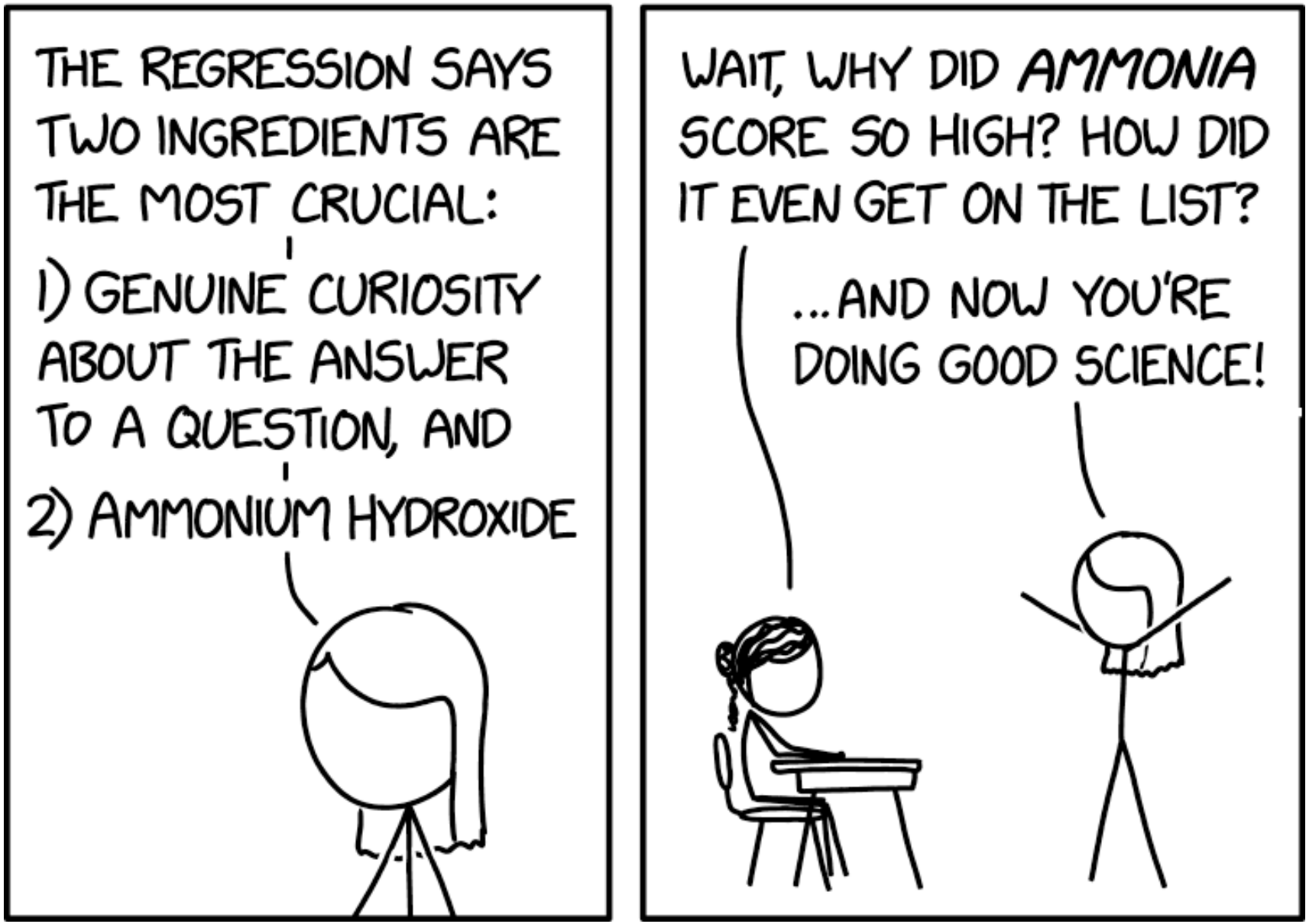
Daniel Valdenegro Ibarra, Jiani Yan, Duiyi Dai, Charlie Rahal
Leverhulme Centre for Demographic Science

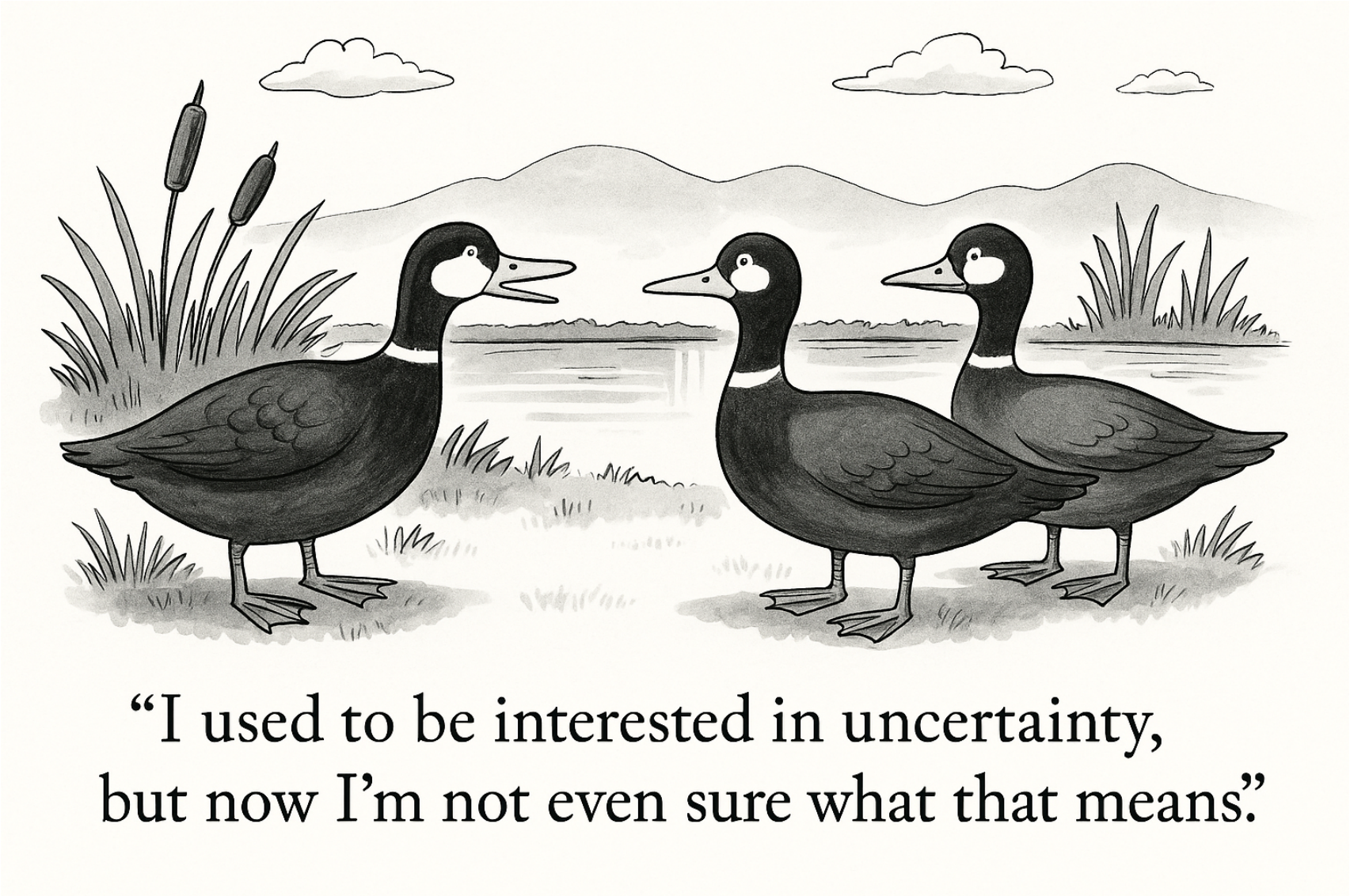
Away day flash research talk; Trinity Term 2025




I'M SUPPOSED TO GIVE YOU THE TOOLS TO DO GOOD SCIENCE.
BUT WHAT ARE THOSE TOOLS?
METHODOLOGY IS HARD AND THERE ARE SO MANY WAYS TO GET INCORRECT RESULTS.
WHAT IS THE MAGIC INGREDIENT THAT MAKES FOR GOOD SCIENCE?


TO FIGURE IT OUT, I RAN A REGRESSION WITH ALL THE FACTORS PEOPLE SAY ARE IMPORTANT:
OUTCOME VARIABLE:
• CORRECT SCIENTIFIC RESULTS
PREDICTORS:
• COLLABORATION
• SKEPTICISM OF OTHERS' CLAIMS
• QUESTIONING YOUR OWN BELIEFS
• TRYING TO FALSIFY HYPOTHESES
• CHECKING CITATIONS
• STATISTICAL RIGOR
• BLINDED ANALYSIS
• FINANCIAL DISCLOSURE
• OPEN DATA


THE REGRESSION SAYS TWO INGREDIENTS ARE THE MOST CRUCIAL:
1) GENUINE CURIOSITY ABOUT THE ANSWER TO A QUESTION, AND
2) AMMONIUM HYDROXIDE
WAIT, WHY DID AMMONIA SCORE SO HIGH? HOW DID IT EVEN GET ON THE LIST?
...AND NOW YOU'RE DOING GOOD SCIENCE!


“I used to be interested in uncertainty,
but now I’m not even sure what that means.”

Introducing RobustiPy: An efficient next generation multiversal library with model selection, averaging, resampling, and explainable artificial intelligence

Daniel Valdenegro Ibarra[1, 2], Jiani Yan[1, 2, 3], Duiyi Dai[1, 2], and Charles Rahal[1, 2, 4]

[1] Leverhulme Centre for Demographic Science, University of Oxford
[2]ESRC Centre for Care, University of Oxford
[3]Wolfson College, University of Oxford
[4]Nuffield College, University of Oxford

June 24, 2025

Abstract

We present `RobustiPy`, a next generation Python-based framework for model-uncertainty quantification and multiverse analysis, released under the GNU GPL v3.0. Through the integration of efficient bootstrap-based confidence intervals, combinatorial exploration of dependent-variable specifications, model selection and averaging, and two complementary joint-inference routines, `RobustiPy` transcends existing uncertainty-quantification tools. Its design further supports rigorous out-of-sample evaluation and apportions the predictive contribution of each covariate. We deploy the library across five carefully constructed simulations and ten empirically grounded case studies drawn from high-impact literature and teaching examples, including a novel re-analysis of 'unexplained discrepancies' in famous prior work. To illustrate its performance, we time-profile `RobustiPy` over roughly 672 million simulated linear regressions. These applications showcase how `RobustiPy` not only accelerates robust inference but also deepens our interpretive insight into model sensitivity across the vast analytical multiverse within which scientists operate.

Keywords: *Model Uncertainty*, *Statistical Software*, *Open Science*

README Code of conduct GPL-3.0 license

RobustiPy

Purpose Research | Python 3.11 | License GNU3.0 | DOI 10.5281/zenodo.15700698

Welcome to the home of `RobustiPy`, a Python library for the creation of a more robust and stable model space. RobustiPy does a large number of things, included but not limited to: high dimensional visualisation, Bayesian Model Averaging, bootstrapped resampling, (in- and)out-of-sample model evaluation, model selection via Information Criterion, explainable AI (via SHAP), and joint inference tests (as per Simonsohn et al. 2019).

Full documentation is available on Read the Docs. The first release is indexed onto Zenodo here.

`RobustiPy` performs Multiversal/Specification Curve Analysis which attempts to compute most or all reasonable specifications of a statistical model, understanding a specification as a single attempt to create an estimand of interest, whether through a particular choice of covariates, hyperparameters, data cleaning decisions, and so forth.

More formally, lets assume we have a general model of the form:

$$\hat{y} = \hat{f}(x, \mathrm{z}) + \epsilon \,.$$

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 bootstrapping, model selection, averaging, resampling, in-and-out-of-sample analysis (/explainable AI), and joint inference tests. It analyses various output spaces, in addition to the control variable space (e.g. multiple dependent variables, estimands of interest, etc). Developed for Python, 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:

```
git clone https://github.com/RobustiPy/robustipy.git
cd robustipy
pip install .
```

and – as we get towards more stable releases – as:

```
pip install robustipy
```

Examples: A series of simulated examples 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.

RobustiPy

latest

Search docs

CONTENTS:

robustipy package

/ Welcome to RobustiPy's documentation! View page source

Welcome to RobustiPy's documentation!

Contents:

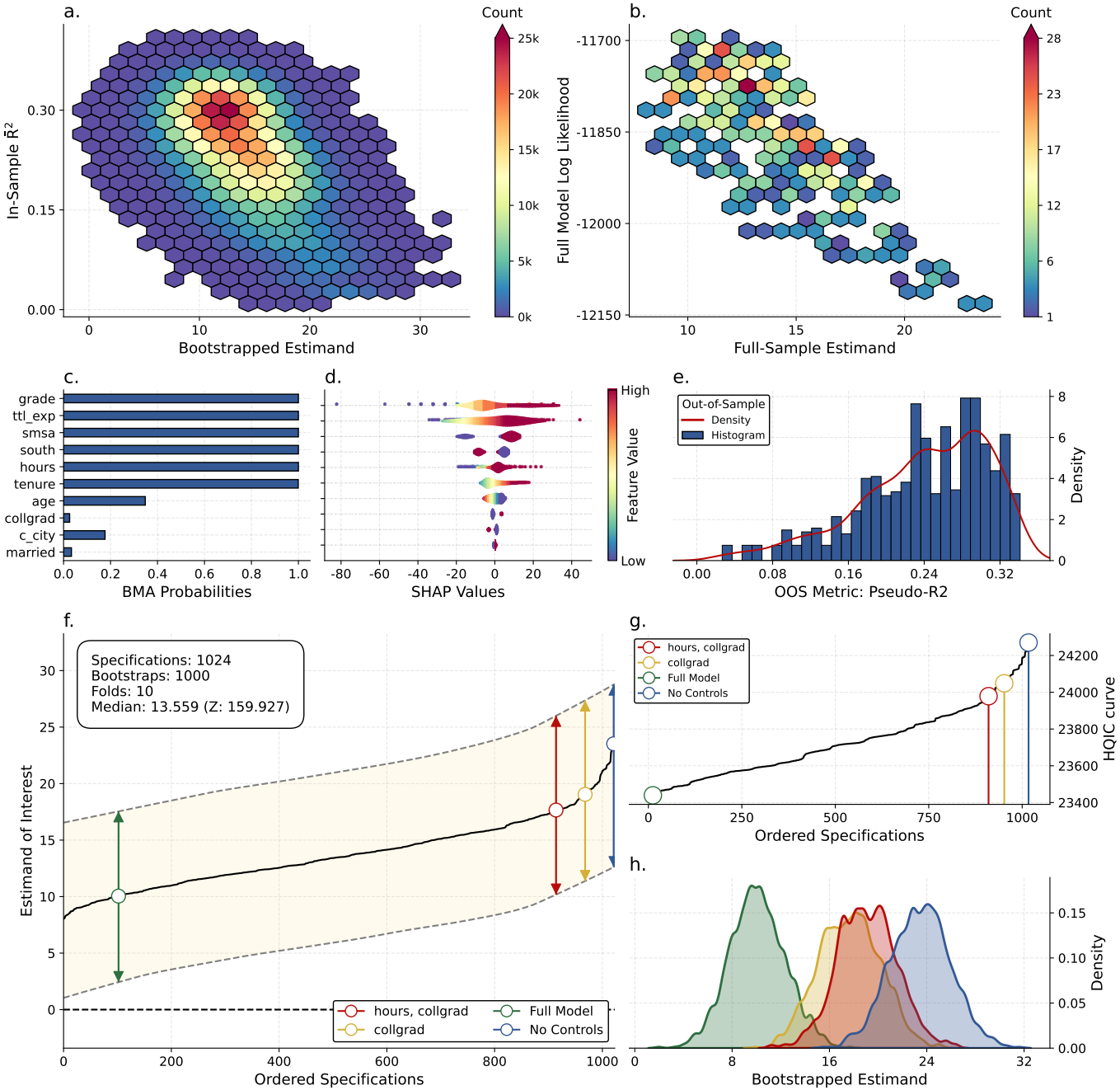
- robustipy package
  - Submodules
  - robustipy.bootstrap_utils module
    - `stripped_ols()`
  - robustipy.figures module
  - robustipy.models module
  - robustipy.prototypes module
    - `BaseRobust`
    - `MissingValueWarning`
    - `Protomodel`
    - `Protoresult`
  - robustipy.utils module
    - `all_subsets()`

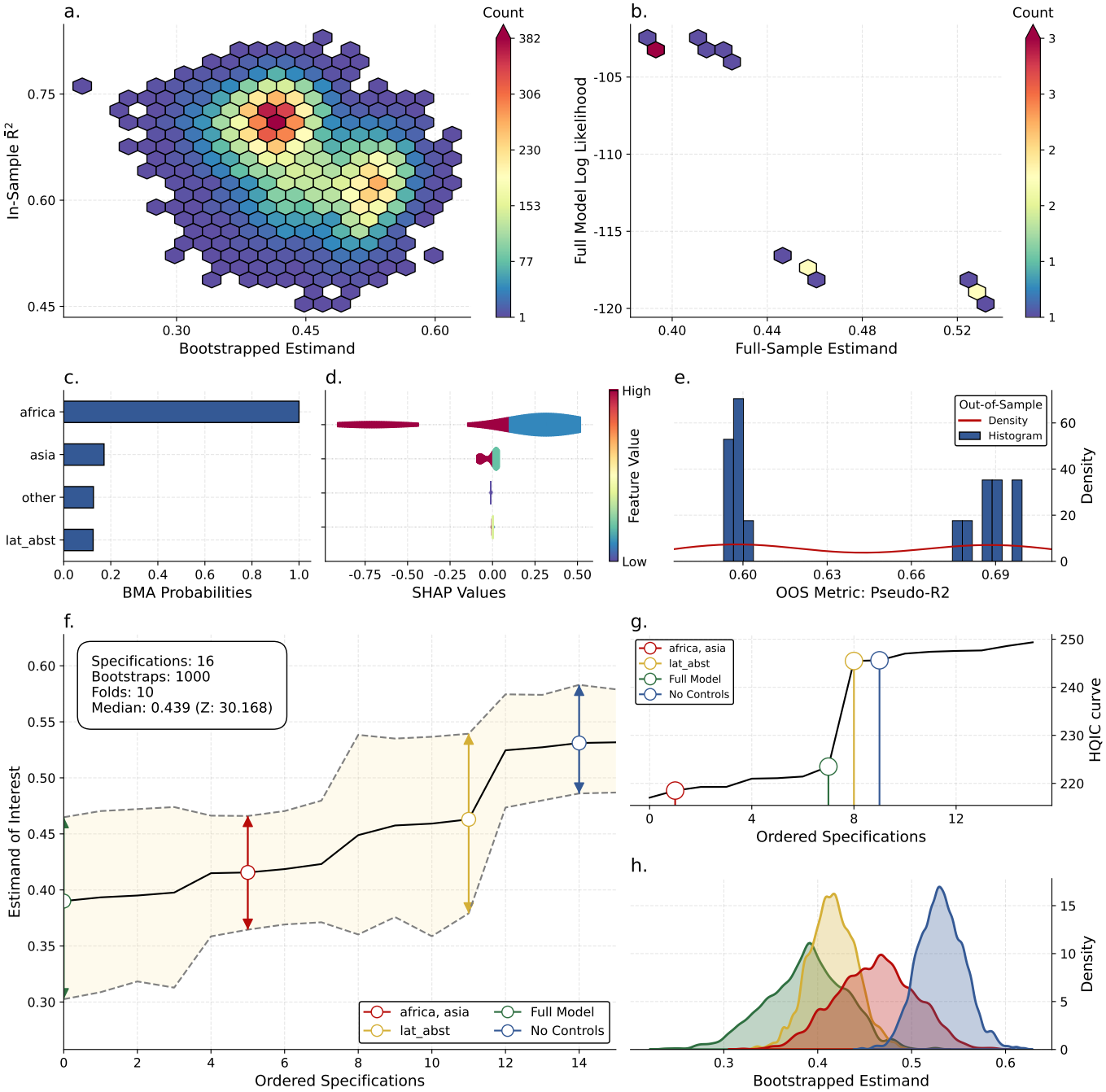
Comparison of Toolkits for Multiverse and Specification Curve Analysis

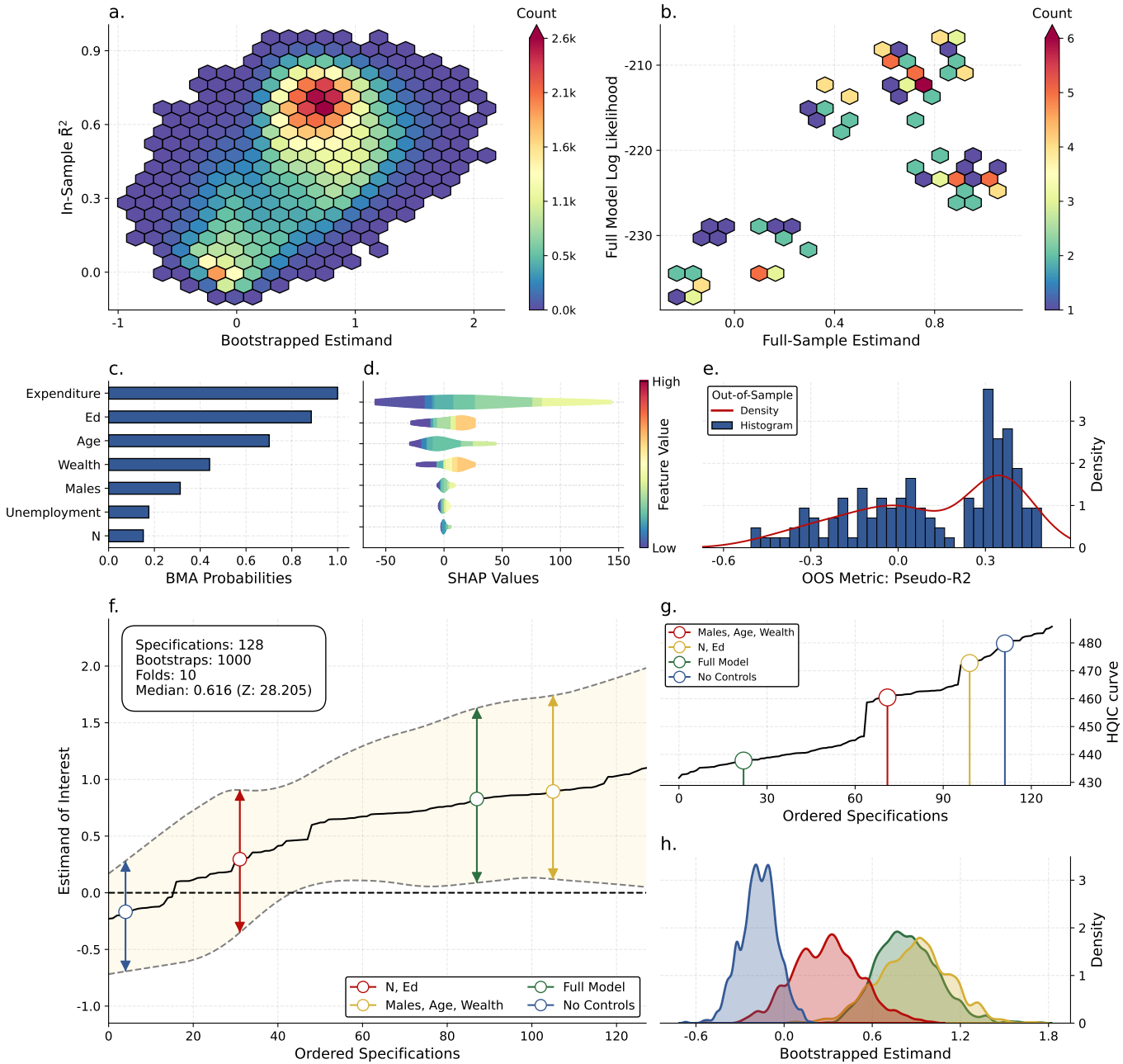
Feature / Toolkit	`multiverse` **(R)**	`specr` **(R)**	`MULTIVRS` **(Stata)**	`spec_curve` **(Python)**	`RobustiPy` **(Python)**
Multiverse analysis	✓	–	✓	–	✓
Specification curve analysis	–	✓	–	✓	✓
High-abstraction syntax	✓	—(partial)	—(limited)	–	✓
Visualisation of results	✓	✓(built-in)	✓(kdensity)	✓(basic)	✓(custom)
Joint inference	–	✓	–	✓	✓
Influence analysis	–	–	✓	–	✓
Bootstrapping	–	✓	–	✓	✓
Out-of-sample	–	–	–	–	✓
Multiple estimands	–	✓(limited)	–	✓(partial)	✓

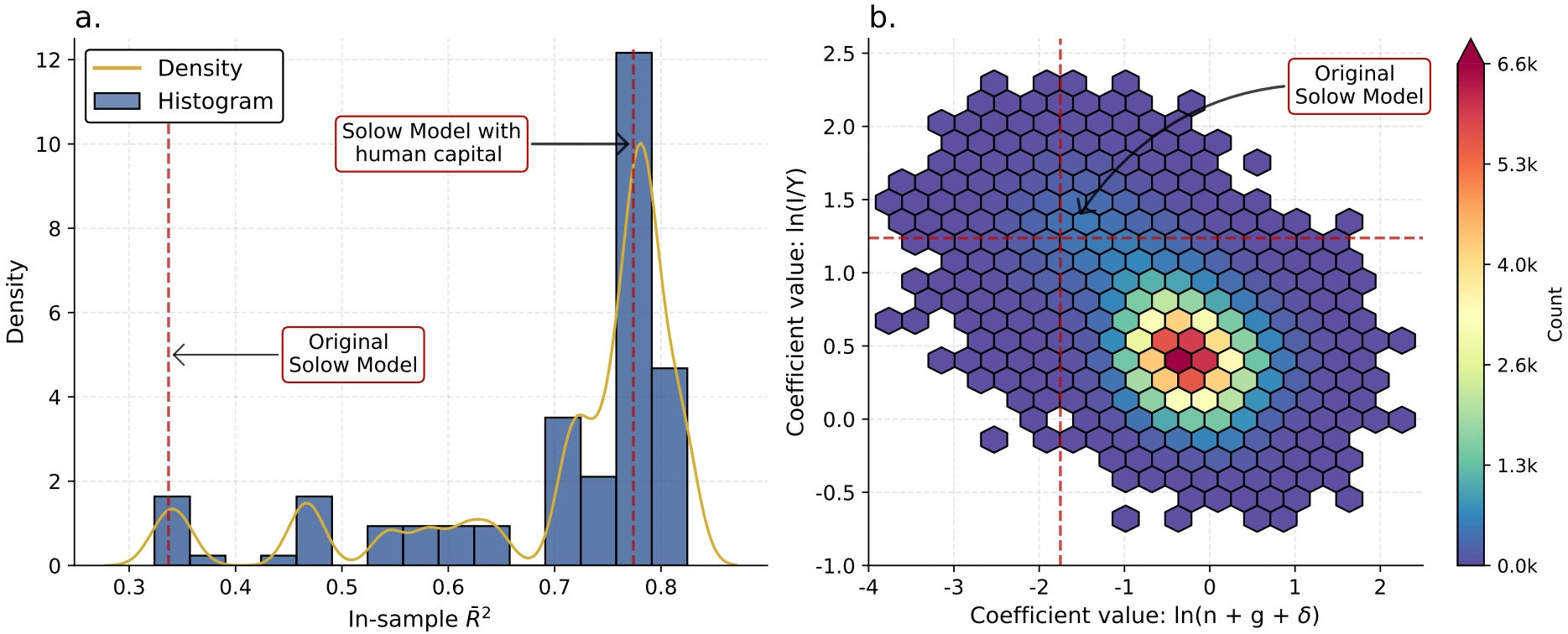
Code Block 1: Vanilla computation

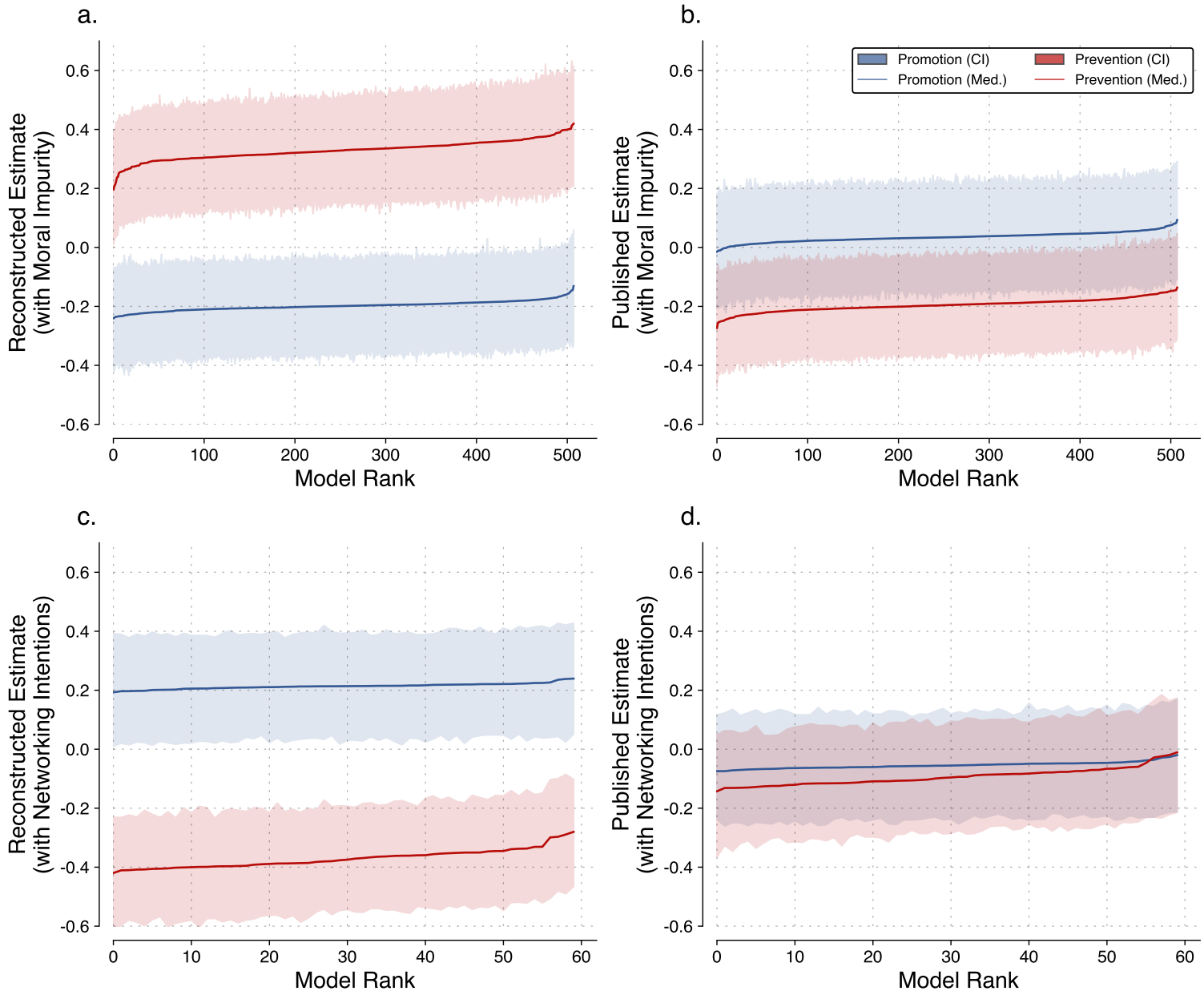
```python
import numpy as np
import pandas as pd
from robustipy.models import OLSRobust


def sim1(project_name):
    # 1. Setup
    np.random.seed(192735)
    n = 100
    rng = np.random.default_rng(0)
    z = [f"z{i}" for i in range(1, 5)]

    # 2. Simulate covariates
    X = rng.standard_normal((n, 1))
    Z = rng.standard_normal((n, len(z)))
    df = pd.DataFrame(np.hstack([X, Z]), columns=["x1"] + z)

    # 3. Generate outcome y
    df["y1"] = (1 + 2 * df["x1"] + 0.5 * df[z].sum(axis=1)
                + rng.standard_normal(n) * 0.5)

    # 4. Fit robust OLS with cross-validation
    vanilla_obj = OLSRobust(y=["y1"], x=["x1"], data=df)
    vanilla_obj.fit(controls=z, draws=1000, kfold=10)

    # 5. Retrieve and plot results
    vanilla_res = vanilla_obj.get_results()
    vanilla_res.plot(specs=[z[1:3]], figsize=(16, 16), ci=0.95,
                     ic="bic", ext="svg", figpath='../figures',
                     project_name=project_name)
    vanilla_res.summary()


if __name__ == "__main__":
    sim1('sim1_example')
```

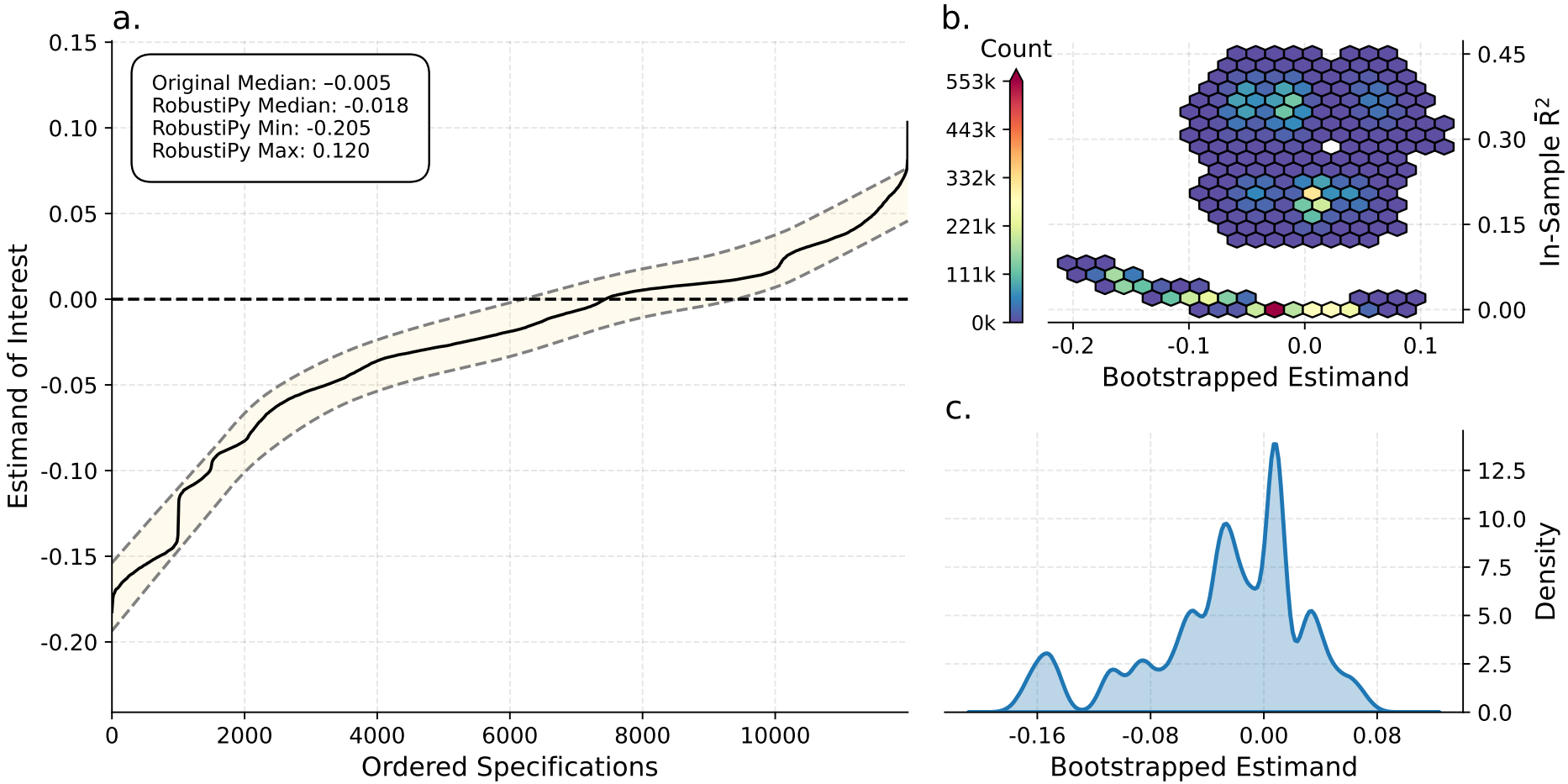

a.
Count
25k
20k
15k
10k
5k
0k
In-Sample $\bar{R}^2$
0.00
0.15
0.30
0
10
20
30
Bootstrapped Estimand
b.
Count
28
23
17
12
6
1
Full Model Log Likelihood
-11700
-11850
-12000
-12150
10
15
20
Full-Sample Estimand
c.
grade
ttl_exp
smsa
south
hours
tenure
age
collgrad
c_city
married
0.0
0.2
0.4
0.6
0.8
1.0
BMA Probabilities
d.
High
Feature Value
Low
-80
-60
-40
-20
0
20
40
SHAP Values
e.
Out-of-Sample
Density
Histogram
0.00
0.08
0.16
0.24
0.32
OOS Metric: Pseudo-R2
Density
0
2
4
6
8
f.
Specifications: 1024
Bootstraps: 1000
Folds: 10
Median: 13.559 (Z: 159.927)
Estimand of Interest
0
5
10
15
20
25
30
hours, collgrad
collgrad
Full Model
No Controls
0
200
400
600
800
1000
Ordered Specifications
g.
hours, collgrad
collgrad
Full Model
No Controls
23400
23600
23800
24000
24200
HQIC curve
0
250
500
750
1000
Ordered Specifications
h.
0.00
0.05
0.10
0.15
Density
0
8
16
24
32
Bootstrapped Estimand

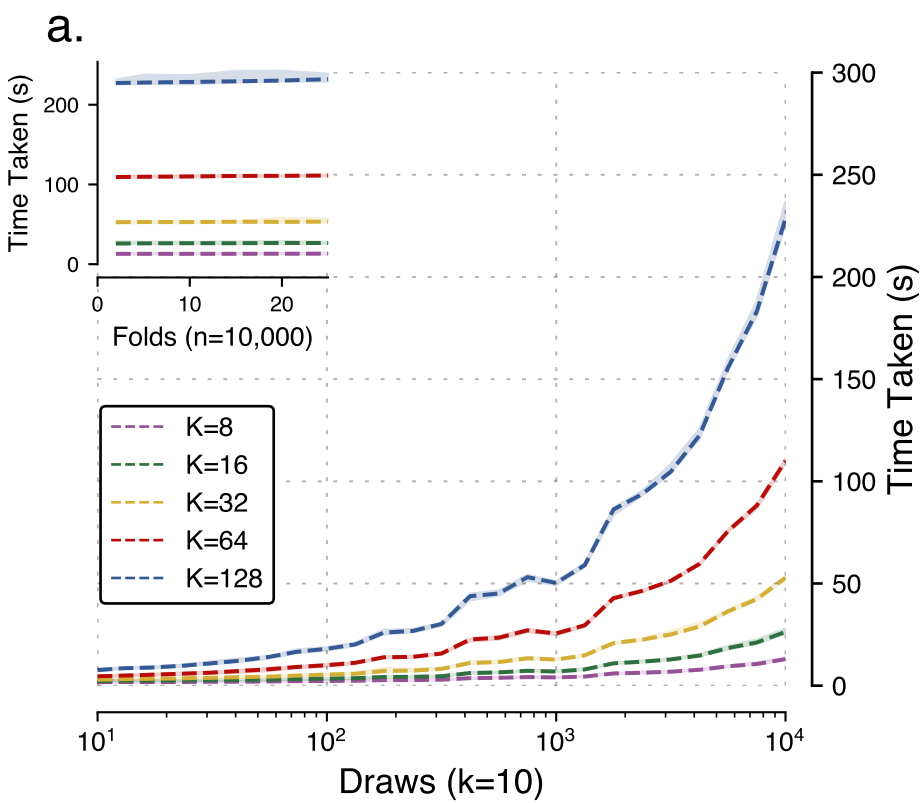

a.
Count
382
306
230
153
77
1
In-Sample $\bar{R}^2$
0.75
0.60
0.45
0.30
0.45
0.60
Bootstrapped Estimand
b.
Count
3
3
2
2
1
1
Full Model Log Likelihood
-105
-110
-115
-120
0.40
0.44
0.48
0.52
Full-Sample Estimand
c.
africa
asia
other
lat_abst
0.0
0.2
0.4
0.6
0.8
1.0
BMA Probabilities
d.
High
Feature Value
Low
-0.75
-0.50
-0.25
0.00
0.25
0.50
SHAP Values
e.
Out-of-Sample
Density
Histogram
60
40
20
0
Density
0.60
0.63
0.66
0.69
OOS Metric: Pseudo-R2
f.
Specifications: 16
Bootstraps: 1000
Folds: 10
Median: 0.439 (Z: 30.168)
Estimand of Interest
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0
2
4
6
8
10
12
14
Ordered Specifications
africa, asia
lat_abst
Full Model
No Controls
g.
africa, asia
lat_abst
Full Model
No Controls
250
240
230
220
HQIC curve
0
4
8
12
Ordered Specifications
h.
15
10
5
0
Density
0.3
0.4
0.5
0.6
Bootstrapped Estimand

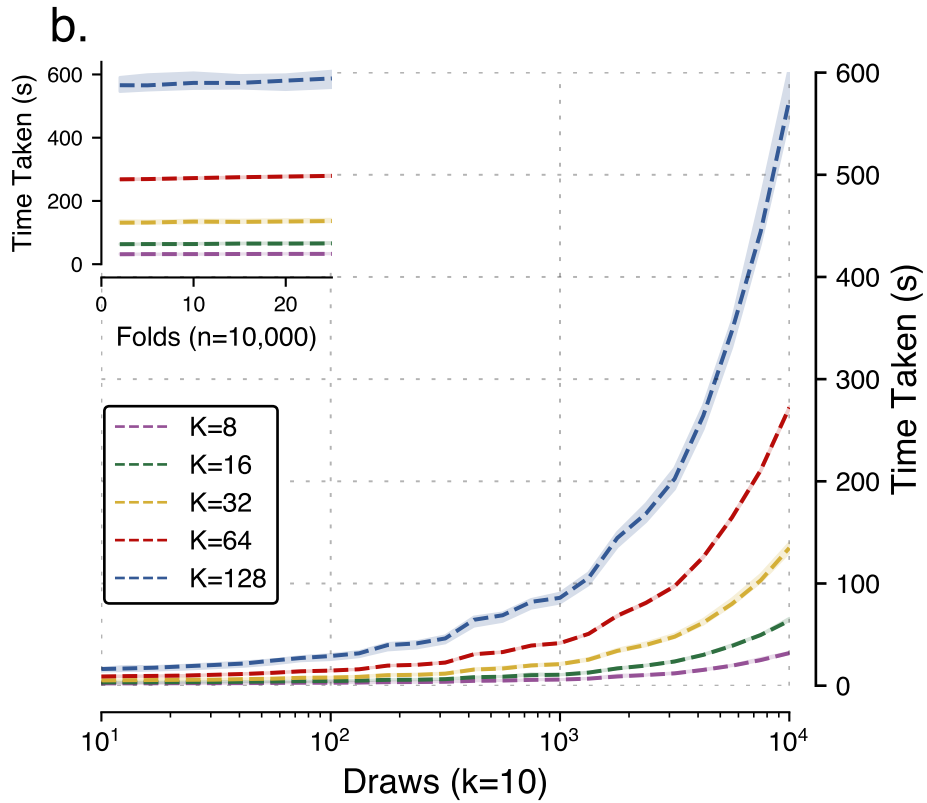

a.
Count
2.6k
2.1k
1.6k
1.1k
0.5k
0.0k
In-Sample $\bar{R}^2$
Bootstrapped Estimand
b.
Count
Full Model Log Likelihood
Full-Sample Estimand
c.
Expenditure
Ed
Age
Wealth
Males
Unemployment
N
BMA Probabilities
d.
High
Feature Value
Low
SHAP Values
e.
Out-of-Sample
Density
Histogram
Density
OOS Metric: Pseudo-R2
f.
Specifications: 128
Bootstraps: 1000
Folds: 10
Median: 0.616 (Z: 28.205)
Estimand of Interest
N, Ed
Males, Age, Wealth
Full Model
No Controls
Ordered Specifications
g.
Males, Age, Wealth
N, Ed
Full Model
No Controls
HQIC curve
Ordered Specifications
h.
Density
Bootstrapped Estimand


a.
Density
Histogram
Solow Model with human capital
Original Solow Model
Density
In-sample $\bar{R}^2$
b.
Original Solow Model
Coefficient value: ln(I/Y)
Coefficient value: ln(n + g + δ)
Count
6.6k
5.3k
4.0k
2.6k
1.3k
0.0k

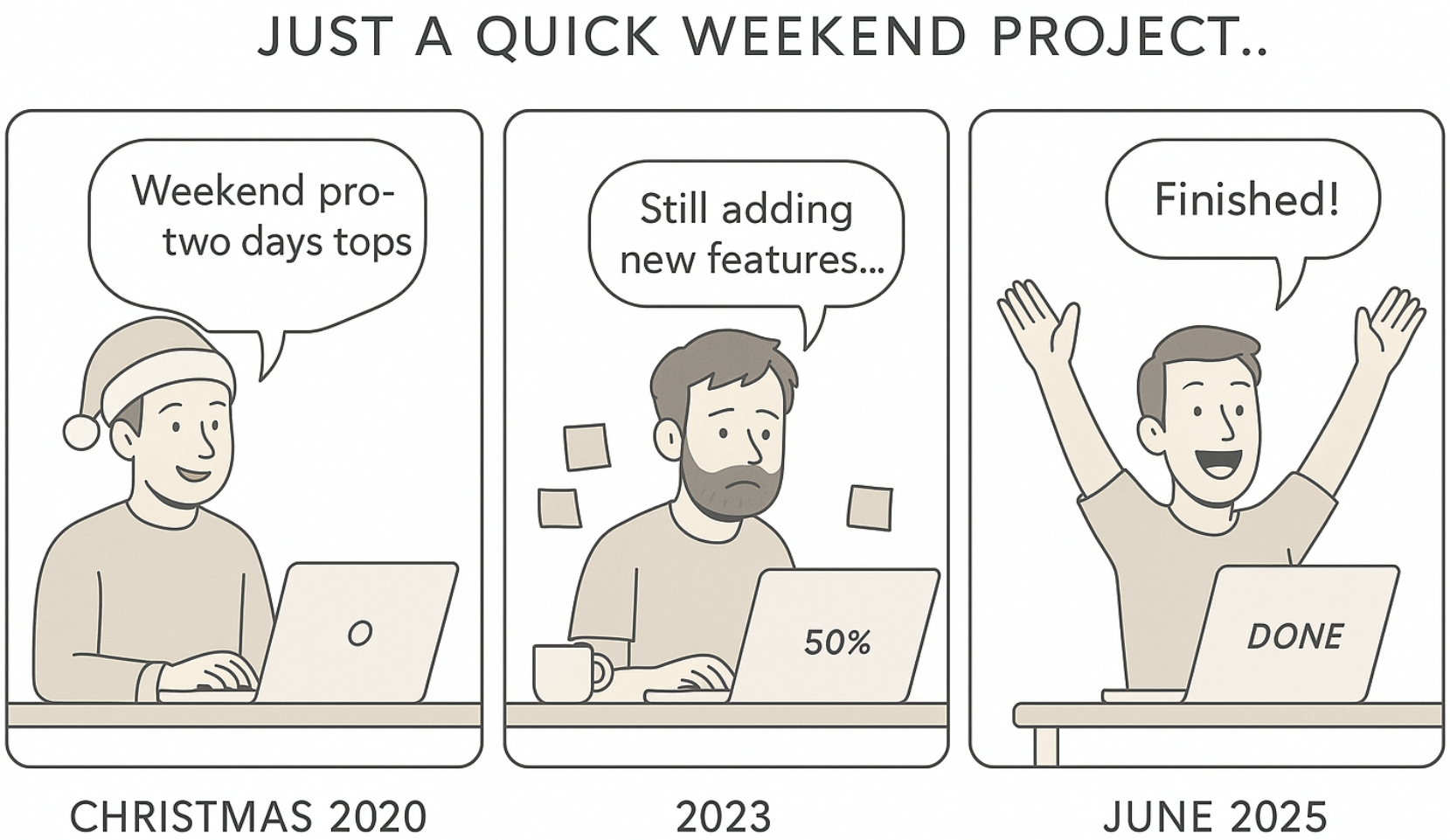

a.
b.
c.
d.
Promotion (CI)
Promotion (Med.)
Prevention (CI)
Prevention (Med.)
Reconstructed Estimate (with Moral Impurity)
Published Estimate (with Moral Impurity)
Reconstructed Estimate (with Networking Intentions)
Published Estimate (with Networking Intentions)
Model Rank


a.
Original Median: –0.005
RobustiPy Median: -0.018
RobustiPy Min: -0.205
RobustiPy Max: 0.120
Estimand of Interest
Ordered Specifications
b.
Count
553k
443k
332k
221k
111k
0k
In-Sample $\bar{R}^2$
Bootstrapped Estimand
c.
Density
Bootstrapped Estimand


a.
Time Taken (s)
0
100
200
0
10
20
Folds (n=10,000)
K=8
K=16
K=32
K=64
K=128
300
250
200
150
100
50
0
Time Taken (s)
10^1
10^2
10^3
10^4
Draws (k=10)


b.
Time Taken (s)
0
200
400
600
0
10
20
Folds (n=10,000)
K=8
K=16
K=32
K=64
K=128
600
500
400
300
200
100
0
Time Taken (s)
10^1
10^2
10^3
10^4
Draws (k=10)

	All specs, with resamples			All specs, no resamples				Stouffers		OOS $R^{2‡}$		
Replication	Min	Median	Max	Min	Median	Max	BIC†	Z	p	Min	Max	**Notes**
Union Dataset	-0.6	13.6	23.7	8.1	13.5	32.8	9.8	159.9	0.0	0.0	0.3	See Young and Cumberworth, 2025
Acemoglu et al. (2001)	0.19	0.44	0.62	0.39	0.44	0.53	0.42	30.17	0.0	0.59	0.7	Constant in $\overleftrightarrow{\mathbf{X}}$
Acemoglu et al. (2001)	0.23	0.44	0.62	0.39	0.81	1.23	0.42	44.32	0.0	-0.43	0.7	Constant in $\overleftrightarrow{\mathbf{Z}}$
Vandervale (1987)	-0.93	0.62	2.11	-0.23	0.70	1.09	0.73	28.21	0.0	-0.51	0.49	See also Erlich (1973)
Mankiew et al. (1982)	-0.68	0.47	2.29	0.26	0.41	1.31	0.30	25.28	0.0	0.31	0.80	$\overleftrightarrow{\mathbf{X}} = [\log\frac{I}{Y}, \log(n+g+\delta)]$
Mankiew et al. (1982)	-3.76	-0.32	2.41	-1.82	-0.21	0.27	-0.17	1.73	0.04	0.31	0.8	$\overleftrightarrow{\mathbf{X}} = [\log(n+g+\delta), \log\frac{I}{Y}]$
ISER (2024)	0.01	0.01	0.01	0.01	0.01	0.01	0.01	91.93	0.0	0.01	0.01	Grouping variable
Zhang et al. (2021)	-0.01	0.20	0.37	0.17	0.20	0.24	0.18	14.37	0.0	0.0	0.01	Grouping / multiple $\overleftrightarrow{\mathbf{X}}$
Zhang et al. (2021)	-0.241	0.14	0.46	-0.099	0.14	0.35	0.20	17.79	0.0	-0.00	0.01	Resampling $\overleftrightarrow{\mathbf{Z}}$
Dawber et al. (1951)	-0.06	0.03	0.1	0.0	0.02	0.06	0.01	60.88	0.0	0.01	0.1	`LRobust`
Dawber et al. (1951)	-0.01	0.00	0.01	-0.0	0.00	0.01	0.00	46.67	0.0	0.01	0.09	`OLSRobust`
Passenger Satisfaction	0.01	0.01	0.02	0.01	0.01	0.02	0.01	32.5	0.0	0.02	0.23	`LRobust`
Passenger Satisfaction	0.00	0.00	0.00	0.00	0.00	0.00	0.00	32.5	0.0	0.44	0.49	`OLSRobust`
Gino et al. (2020)	-0.61	-0.2	0.22	-0.24	-0.2	-0.14	n/a	43.54	0.0	0.02	0.07	Promotion, published
Gino et al. (2020)	-0.19	0.33	0.85	0.20	0.33	0.42	n/a	77.69	0.0	0.86	0.99	Prevention, published
Gino et al. (2020)	-0.41	0.03	0.49	-0.01	0.03	0.09	n/a	-13.05	1.0	-0.00	0.02	Promotion, original
Gino et al. (2020)	-0.63	-0.2	0.23	-0.2	-0.27	-0.14	n/a	40.78	0.00	-0.00	0.02	Prevention, original
Gino et al. (2020)	0.21	-0.21	0.64	0.20	0.21	0.24	n/a	15.39	0.00	0.04	0.06	Promotion, published*
Gino et al. (2020)	-0.77	-0.37	0.11	-0.41	-0.37	-0.28	n/a	28.8	0.0	0.04	0.06	Prevention, published*
Gino et al. (2020)	-0.05	-0.44	0.38	-0.06	-0.07	-0.03	n/a	-1.53	0.94	-0.01	0.01	Promotion, original*
Gino et al. (2020)	-0.54	-0.09	0.37	-0.14	-0.09	-0.01	n/a	2.96	0.00	-0.01	0.01	Prevention, original*
Orben and Przybylski (2020)	-0.20	-0.02	0.12	-0.18	-0.02	0.1	n/a	472.71	0.0	0.38	0.77	Multiple Dependents


JUST A QUICK WEEKEND PROJECT..
Weekend pro-
two days tops
Still adding
new features...
Finished!
50%
DONE
CHRISTMAS 2020
2023
JUNE 2025