Unit 2 Project - Simulation Study (due by 11pm on 10-17)

Goal. Implement a simulation study to evaluate statistical methods or validate theoretical results. Write a well-structured, reproducible pipeline with clear reporting.

Learning Objectives. By completing this project, you will:

- Reflect on study design using the ADEMP framework
- Implement efficient, modular simulation code
- Create informative visualizations that clearly communicate simulation results
- Critically evaluate simulation design choices and their implications

Note: You may work with a partner if both are reproducing the same paper (or working together on a new simulation). In this case, divide responsibilities clearly and document each person's contributions in the README. Both partners must submit individual ANALYSIS.md documents with their own critical perspectives.

Project Requirements

Core Requirements

1. Simulation Selection

Choose **one** of the following options:

Option A: Reproduce an Existing Study

- Select a paper from the provided list or choose your own (subject to approval)
- The paper must include simulation results you can attempt to reproduce
- Before starting, document the paper's simulation design and results

Option B: Original Simulation Study

- Design simulations relevant to your research or a methodological question of interest
- Must address a clear statistical question (e.g., method comparison, property verification, robustness assessment)
- Consult with instructor to ensure appropriate scope

2. ADEMP Framework

Create a document (ADEMP.md) that clearly describes your simulation using the ADEMP framework:

- **Aims**: What questions are you trying to answer? What hypotheses are you testing?
- **Data-generating mechanisms**: What are the data-generating processes? What parameters vary across simulation conditions?
- **Estimands/targets**: What quantities are you trying to estimate or evaluate?
- **Methods**: What statistical methods/estimators are being compared or evaluated?
- **Performance measures**: What metrics will you use to evaluate performance? (e.g., bias, variance, MSE, coverage, power, Type I error)
  Include a table summarizing your simulation design matrix showing all combinations of conditions. Write a description of the simulation that would be sufficient for someone else to reproduce it.

3. Project Structure

Organize your project following this structure:

```
simulation-study/
├── data/
│   └── simulated/          # cache simulation replicates if needed
├── src/
│   ├── dgps.py             # data-generating functions
│   ├── methods.py          # statistical methods being evaluated
```

```
│   ├── metrics.py          # performance measure calculations
│   └── simulation.py       # main simulation orchestration
├── results/
│   ├── figures/            # visualizations
│   └── raw/                # raw simulation output (*.csv, *.pkl)
├── tests/
├── Makefile                # automated workflow
├── ADEMP.md                # simulation design document
├── README.md
├── requirements.txt
└── .gitignore
```

4. Implementation Requirements

- **Modular code**: Separate DGP, methods, and evaluation into distinct, testable functions
- **Reproducibility**: Set and document random seeds appropriately
- **Parameter configuration**: Use configuration files or clear parameter dictionaries (not hardcoded values scattered throughout)
- **Progress tracking**: Include logging or progress indicators for long-running simulations
- **Intermediate outputs**: Save raw simulation results before summarization

5. Makefile Automation

Create a Makefile with targets including:

- `make all`: Run complete simulation pipeline and generate all outputs
- `make simulate`: Run simulations and save raw results
- `make analyze`: Process raw results and generate summary statistics
- `make figures`: Create all visualizations
- `make clean`: Remove generated files
- `make test`: Run test suite

6. Visualization Requirements

Create **at least 2 visualizations (one diagnostic and one publication-quality)** that effectively communicate your results:

- Use appropriate plot types for your performance measures (e.g., boxplots, line plots, heatmaps)
- Include confidence intervals or uncertainty measures where appropriate
- Use clear labels, legends, and titles
- Facet by simulation conditions in the right order for your study
- Make figures interpretable without reading the full report

Note: Even if reproducing a paper with poor visualizations, create informative plots that better communicate the results!

7. Testing

Implement **at least 3 tests**, such as

- DGP verification (e.g., generated data has expected properties)
- Method correctness (e.g., method recovers true values in simple cases)
- Output validation (e.g., performance measures are in valid ranges, especially compared to existing results if you are reproducing a paper)
- Reproducibility check (e.g., rerunning with same seed gives identical results)

8. Critical Analysis Document

Create ANALYSIS.md (1-2 pages) addressing:

For Option A (Reproducing Existing Paper):

- How well did you reproduce the original results? What differed?
- Evaluate the **neutrality** of the simulation design:

    - Were the simulation conditions fair to all methods being compared? Did they ignore important predecessors?
    - Were there any design choices that favored certain methods?
    - Were realistic scenarios included, or only idealized conditions?
- What would you change about the simulation design? Why?
- Did you recreate the visualizations or make your own? What did the visualizations reveal that the original paper missed or undersold?
- What was surprising or unexpected in the results?

For Option B (Original Simulation):

- Justify your design choices: Why these DGPs? Why these sample sizes/conditions?
- How did you ensure your simulation design was fair and unbiased?
- What are the limitations of your simulation study?
- What scenarios did you *not* include, and why might they matter?
- How do your results inform practice or theory?
- What would you investigate next if you had more time/resources?

Both Options:

- Which aspects of the implementation were most challenging?
- How confident are you in your results? What could undermine that confidence?

Submission Requirements

1. **Repository URL** with public access
2. **ANALYSIS.md**: Critical analysis (~1 page, uploaded separately)
3. **ADEMP.md**: Simulation design document
4. **README.md** with:
    - Brief project description
    - Setup instructions
    - How to run the complete analysis (`make all`)
    - Estimated runtime
    - Summary of key findings (1-2 sentences)
5. **Working Makefile** that successfully runs end-to-end
6. **Test suite** that passes
7. **Note:** Don't worry about runtime for this project (we'll optimize in Project 3)

Suggestions

Simulation Design Tips

- Start simple: Get one condition working end-to-end before expanding
- Include a "sanity check" condition where you know the right answer
- Consider both favorable and unfavorable conditions for each method
- Think about practical, not just theoretical, scenarios

Implementation Tips

- Cache intermediate results to avoid rerunning expensive simulations
- Use `tqdm` for progress bars in loops
- Test DGPs by visualizing generated data before running full simulations

Paper Selection (Option A)

Good papers to reproduce have:

- Complete description of simulation setup
- Some tabular or graphical results you can compare against

- Moderate complexity (not totally toy, not impossibly complex)

Papers Ideas

Multiple testing:

- Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. _JRSS(B).
- Barber, R. F., & Candès, E. J. (2015). Controlling the false discovery rate via knockoffs. *AoS*.

High-dimensional statistics

- Hastie, T., Montanari, A., Rosset, S., & Tibshirani, R. J. (2022). Surprises in high-dimensional ridgeless least squares interpolation. *AoS*.
- Wainwright, M. J. (2009). Sharp thresholds for High-Dimensional and noisy sparsity recovery using L*1-Constrained Quadratic Programming (Lasso). _IEEE transactions on information theory*.

Empirical Bayes

- Greenland, S. (1993). Methods for epidemiologic analyses of multiple exposures: a review and comparative study of maximum-likelihood, preliminary-testing, and empirical-Bayes regression. *Statistics in medicine*.
- Gu, J., & Koenker, R. (2017). Unobserved heterogeneity in income dynamics: An empirical Bayes perspective. *Journal of Business & Economic Statistics*.

Causal inference

- Bang, H., & Robins, J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics*.

Timeline

- **Week 1**:
  - Select paper/topic
  - Draft ADEMP document
  - Implement most of DGP, methods, etc.
- **Week 2**:
  - Complete simulation design
  - Implement Makefile
  - Run simulations
  - Create visualizations
  - Write analysis document
  - Finalize documentation

Tools

- **Build automation**: `make`
- **Visualization**: `matplotlib` preferred, or `plotnine`, `seaborn` or `plotly`
- **Testing**: `pytest` preferred
- **Version control**: `git` with GitHub