# Simulation Framework

## First Approach at a Python-first General Simulation Package

Matt McAnear

Dept. of Statistics, University of Michigan

December 15, 2025

# Current Simulation Frameworks

Major simulation frameworks are currently all in R.

- simR [3][4]
- simchef [2]
- simulator [1]

This is basically absent from Python.

# Values

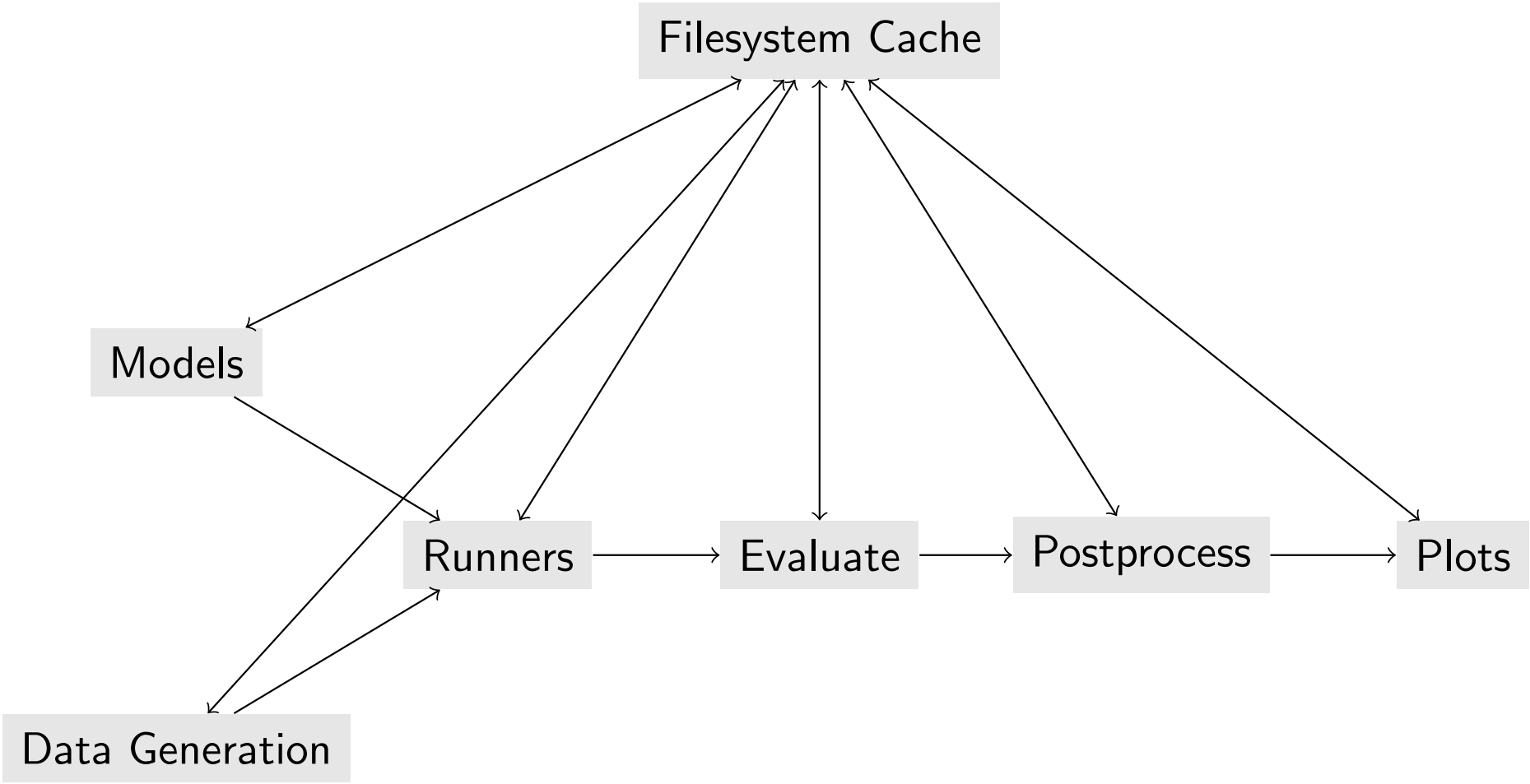
- It should feel like Python
- Composition over Inheritance
- Functional Style
- Modular
- Natural Vectorization

# Design

# Core Components

- **Models**: Data Generating Processes
- **Methods**: Procedures to be applied to data
- **Metrics**: Performance Measures
- **Visualizations**: Orchestration of the above

# Architecture



# Component Details

- Data Generation
  - Tag arbitrary functions as data generators
  - Returns data sets as tensor-like structures for easier vectorization
  - Automatically create an output type
- Models
  - Align inputs with output of data generators
  - Tag method outputs and create output types
  - Run model fitting with a given set of model fitting hyperparameters
- Runners
  - Orchestrate / match data generators and models
  - Run only if results are not present
  - Align randomization
  - Control Caching

# Component Details
(continued)

- Evaluate
  - Provide Standard Metrics
  - Tag arbitrary functions as metrics
  - Align inputs with model outputs
- Postprocess
  - Aggregate results or fit models on results
  - Allow for arbitrary functions of outputs that aren't strictly evaluation
- Plot

# Demonstration

# Simple DGP

```
## Usage
@data_generator("Linear Data", output=("X", "y", "beta"))
def ols_data(N, p):
    X = rng.normal(size=(N, p))
    beta = rng.normal(size=p)
    y = X @ beta + rng.normal(size=N)
    return X, y, beta

## Output
output_tuple = ols_data(N=100, p=10)
assert output_tuple.X.shape == (100, 10)
assert output_tuple.y.shape == (100,)
assert output_tuple.beta.shape == (10,)
```

# Simple Method

```
## Usage
@model("Ridge", output="beta")
def ridge_model(X, y, lam=1.0):
    p = X.shape[1]
    eye = np.eye(p)

    beta = np.linalg.solve(X.T @ X + lam * eye, X.T @ y)
    return beta
```

# Running A Simulation

```
# The Happy Path
Runner = SimRunner(
    data_generator=ols_data,
    method=ridge_model,
    sim_params={
        "Linear Data": {"N": [50, 100], "p": [5, 10, 20]},
        "Ridge": {"lam": [0.1, 0.5, 1.0, 1.5]},
    },
)

results = Runner.run_simulations(N_sims)
```

# Running a Simulation

(continued)

```
# Inside the runner class; the idea is to match the output
# arguments from a class with the input values from the method
# signature using the ‘inspect‘ library
def _run_scenario(self, scenario, N_sims):
  dgp_args, method_args = self._parse_scenario_args(scenario)

  for _ in range(N_sims):
    sim_data = self.data_generator(**dgp_args)

    shared_params = set(sim_data._fields).intersection(
        inspect.signature(self.method.fn).parameters
    )

    method_inputs = {k: sim_data.__getattribute__(k) for k in
         shared_params}

    method_output = self.method(**method_inputs, **method_args)

    return method_output
```

# Conclusion

# Lots left to do

- Play with the R versions to get a better feel for features
- Add filesystem based caching
- Simplify the handoff of function outputs to next stage in simulation pipeline
- Solidify random number generation / state

# Works Cited I

[1] Jacob Bien. *The Simulator: An Engine to Streamline Simulations*. arXiv:1607.00021 [stat]. July 2016. DOI: 10.48550/arXiv.1607.00021. URL: http://arxiv.org/abs/1607.00021 (visited on 11/12/2025).

[2] James Duncan et al. *simChef: High-quality data science simulations in R*. Issue: 95 Pages: 6156 Publication Title: Journal of Open Source Software Volume: 9. 2024.

[3] Peter Green and Catriona J. MacLeod. "SIMR: an R package for power analysis of generalized linear mixed models by simulation". en. In: *Methods in Ecology and Evolution* 7.4 (2016). _eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12504, pp. 493–498. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12504. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.12504 (visited on 12/01/2025).

# Works Cited II

[4] Matthew J. Sigal and R. Philip Chalmers. "Play It Again: Teaching Statistics With Monte Carlo Simulation". In: *Journal of Statistics Education* 24.3 (Sept. 2016). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10691898.2016.1246953, pp. 136–156. ISSN: null. DOI: 10.1080/10691898.2016.1246953. URL: https://doi.org/10.1080/10691898.2016.1246953 (visited on 12/01/2025).