Aaron C. Watt

Berkeley, CA — (503) 327-9232 — aaron.watt@berkeley.edu — acwatt.net — github.com/acwatt

Summary: UC Berkeley PhD with a background in Economics and Physics seeking a Quantitative Researcher position. Expertise in applied econometrics, optimization, probability theory, and machine learning with extensive experience in Python and Julia for large-scale data analysis.

EDUCATION

University of California, Berkeley: Ph.D. Agricultural & Resource Economics (Expected Conferral Dec 2025)
Oregon State University: M.S. Applied Economics (2020); B.S. Physics (2014)

TECHNICAL SKILLS

Programming Languages: Python (NumPy, Pandas, GeoPandas, SciPy, scikit-learn), Julia, Stata, Mathematica, R, SQL, OS Scripting
Statistical Modeling & Econometrics: Maximum Likelihood Estimation, Generalized Method of Moments, Spatial Econometrics, Time Series Analysis, Monte Carlo Simulations, Model Building, Feature Engineering
Machine Learning: Neural Networks, Natural Language Processing, Regression Models, LLMs
Data Management: Google Earth Engine, High-performance Computing Clusters, Amazon Web Services, SQL

RELEVANT PROFESSIONAL EXPERIENCE

Research Assistant, Meredith Fowlie, UC Berkeley **2024–Present**
- Built a data pipeline to automate the downloading and processing of terabytes of hourly meteorological data on a remote server (Requests, GeoPandas).
- Engineered a large-scale simulation of electricity generation (PySAM) for over 300,000 potential sites to create inputs for a firm entry model.
- Leveraged GPU acceleration (CuPy) to estimate spatial autoregressive models of firm entry (SciPy, Pandas, SpatBinary, StatsModels, Concurrent.futures).

Research Consultant, Resources for the Future **2023**
- Developed a framework to quantify prediction uncertainty of black-box air quality projections using satellite, in-situ, and administrative data with Python (GeoPandas, StatsModels) and Google Earth Engine.

Research Assistant, Larry Karp, UC Berkeley **2021–2023**
- Implemented Maximum Likelihood & Generalized Method of Moments estimators for AR(1)/state-space models in Julia (DataFrames, Distributions, Optim, StatsBase, Turing).
- Validated estimator performance using large-scale Monte Carlo simulations on a high-performance computing cluster (Slurm, 10,000 simulated datasets).

Research Consultant, Eugenie Dugoua & Marion Dumas, London School of Economics **2021**
- Applied topic modeling (LDA) and clustering (K-means) to analyze the content of 100,000+ documents in Python (Pandas, SciPy, scikit-learn, Gensim, spaCy).

Research Assistant, Robin Cross, Oregon State University **2018–2022**
- Developed computer vision and econometric models in Python and Stata to analyze agricultural survey data, disease lab results, and satellite data. (TensorFlow, StatsModels, Mata, AWS EC2)

SELECTED PERSONAL PROJECTS

Optimizing Air Quality Monitor Placement with Strategic Local Agents **2022–present**
- Designed and implemented an optimization model to determine the ideal placement of air quality monitors, accounting for strategic agents' behavior. Engineered features of agents' decision-making process.
- Python (SciPy, scikit-learn, StatsModels, PyLogit, Distributed), Julia (QuadGK, JuMP, Turing, Gen, MonteCarloMeasurements), Stata, and Google Earth Engine.

Predicting Commodity Futures Prices **2025–present**
- Engineered a predictive pipeline to forecast commodity futures prices, leveraging AutoML (AutoGluon) and deep learning (Flux.jl, BigQuery) frameworks.

Predicting Supreme Court Decisions using an LLM agent swarm **2025–present**
- Built a hierarchical LLM agent swarm that autonomously researches legal precedents, creates a RAG memory system, debates outcomes, and predicts Supreme Court decisions.
- Python (Google.GenerativeAI, ChromaDB, scikit-learn.metrics, Requests, BeautifulSoup)

Testing for Strategic Timing of Air Quality Monitors **2021–2022**
- Developed a statistical testing framework to detect strategic behavior in environmental data reporting by comparing monitored vs. non-monitored periods using data from consumer products.
- Python (Pandas, StatsModels, Requests, RateLimiter) and Amazon Web Services (lambda functions)