STAT 563 Project #2 Report

Bryan Nsoh

November 1, 2025

Introduction

In this project I recreated the instructor's MATLAB workflow using Python. I ran the provided pipelines so that every figure and numerical summary in this report comes directly from the code in the repository. My goal is to understand how the logistic distribution behaves, how Fisher scoring estimates its parameters, and how bootstrap and slice sampling describe correlation uncertainty.

1 Exploring the Logistic Distribution

Figure 1 compares logistic densities across several parameter choices. The left panel shows how increasing the scale makes the curve flatter in the middle and heavier in the tails, while the right panel shifts the curve left or right when the location parameter changes. The dashed curve is the Normal$(0, 1)$ density, included to highlight that the logistic tails stay thicker, which is why logistic models can be more robust to outliers than Gaussian ones.

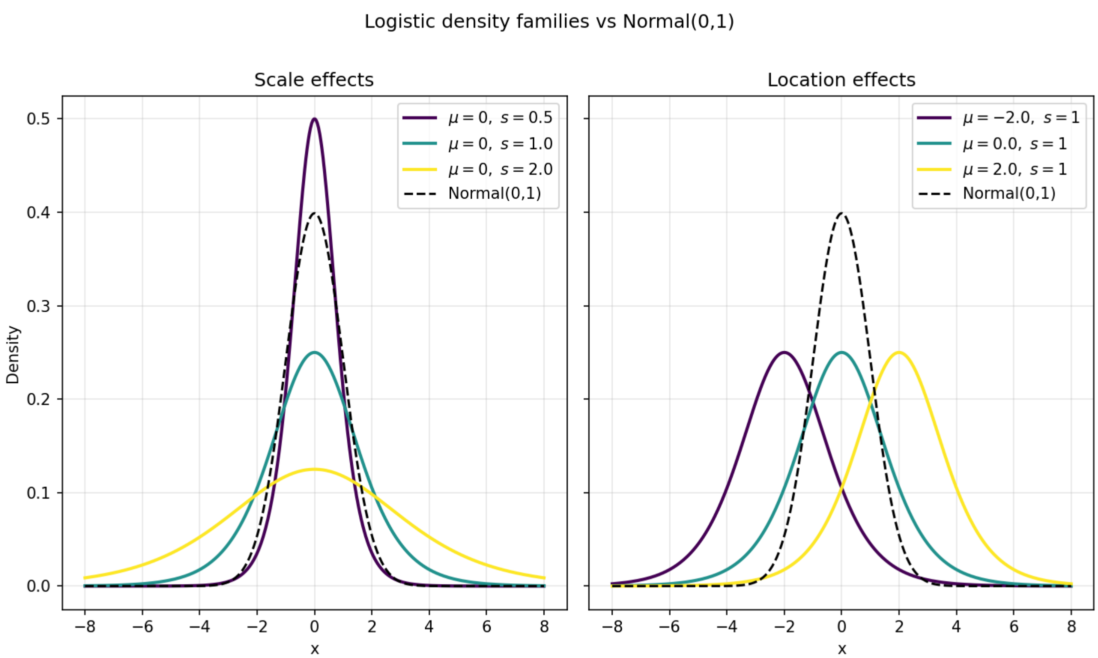

Figure 1: Logistic density families with varying scale (left) and location (right), compared to the Normal$(0, 1)$ curve.

2 Fisher Scoring Estimates for Logistic Parameters

I simulated $n = 200$ draws from Logistic$(0, 1)$ using the project pipeline and ran Fisher scoring on that sample. The algorithm converged after 16 iterations (converged: `true`) and the final score norm was 3.21e-06, indicating a stable solution. The estimates were $\hat{\mu} = -0.167$ and $\hat{s} = 1.049$, which are both close to the data-generating values. Using the expected Fisher information produced large-sample Wald intervals: $\hat{\mu} \in [-0.418, 0.085]$ and $\hat{s} \in [0.953, 1.145]$. The location interval spans about 0.503, showing moderate precision, while the scale interval spans only 0.192, reflecting the amount of information about spread in a sample of this size.

3 Mean Confidence Intervals: Wald vs Bootstrap

The sample mean was $\bar{x} = -0.155$. The Wald interval based on asymptotic variance was $[-0.415, 0.104]$, and the percentile bootstrap interval from 2,000 resamples was $[-0.414, 0.106]$. The two intervals overlap almost perfectly; the bootstrap version is only slightly wider, which matches the idea that resampling captures tail behavior without assuming normality. The bootstrap histogram shows a near-symmetric distribution centered on the sample mean, so both approaches give practically the same answer here.

4 Correlation Uncertainty via Bootstrap and Slice Sampling

I also simulated a five-dimensional normal sample with an AR(1)-style correlation structure ($n = 200$) and focused on $r_{1,3}$. Figure 2 compares three views of its uncertainty. The left panel shows the bootstrap distribution with percentile and Fisher-z bands; they nearly coincide, suggesting the Gaussian approximation is adequate. The right panel shows the slice-sampled marginal density, whose interval closely matches the bootstrap range, confirming the distribution is roughly symmetric with slightly heavier tails than the Fisher-z curve predicts.

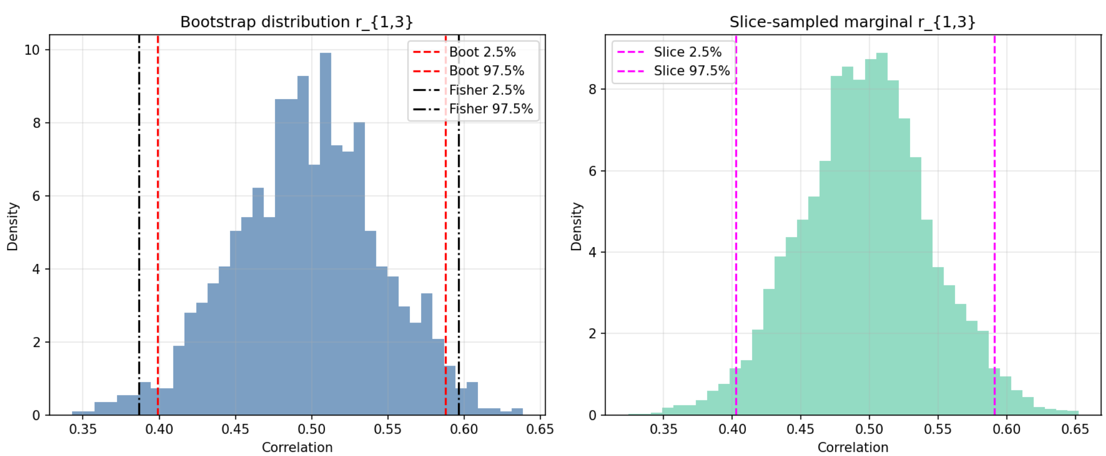

Figure 2: Bootstrap and slice-sampled uncertainty for correlation $r_{1,3}$.

Numerically, the percentile bootstrap interval was $[0.399, 0.588]$, the Fisher-z interval was $[0.387, 0.596]$, and the slice-sampled interval was $[0.403, 0.591]$. All three agree within a few thousandths, which reassures me that the correlation estimate is stable for this design.

Conclusions

Re-running the project in Python clarified several points. Visualizing the logistic density made its heavier tails tangible, which explains why logistic regression is resilient to outliers. Fisher scoring delivered accurate parameter estimates with a small score norm, and the Wald intervals were trustworthy at this sample size. The bootstrap and percentile intervals for the mean aligned closely, reinforcing that either approach works when the sampling distribution is nearly symmetric. Finally, bootstrapping and slice sampling produced consistent pictures of correlation uncertainty, while Fisher-z gave a quick analytic check. Anyone can reproduce these results by running the two pipeline scripts listed in the repository README.