STAT 563 Project #2 Report

Bryan Nsoh

November 1, 2025

Introduction

In this project I rebuilt the instructor's MATLAB workflow with Python so that every figure and numerical summary is generated from reproducible scripts. The assignment ties together large-sample theory, likelihood-based inference, and resampling using the logistic distribution, whose CDF provides the canonical logit link, drives logistic regression, and underpins cross-entropy and softmax losses in machine learning. I therefore examine how the logistic density's shape responds to location and scale changes, estimate the parameters with Fisher scoring, compare asymptotic and bootstrap confidence intervals, and study correlation uncertainty through both bootstrap resampling and slice sampling.

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.

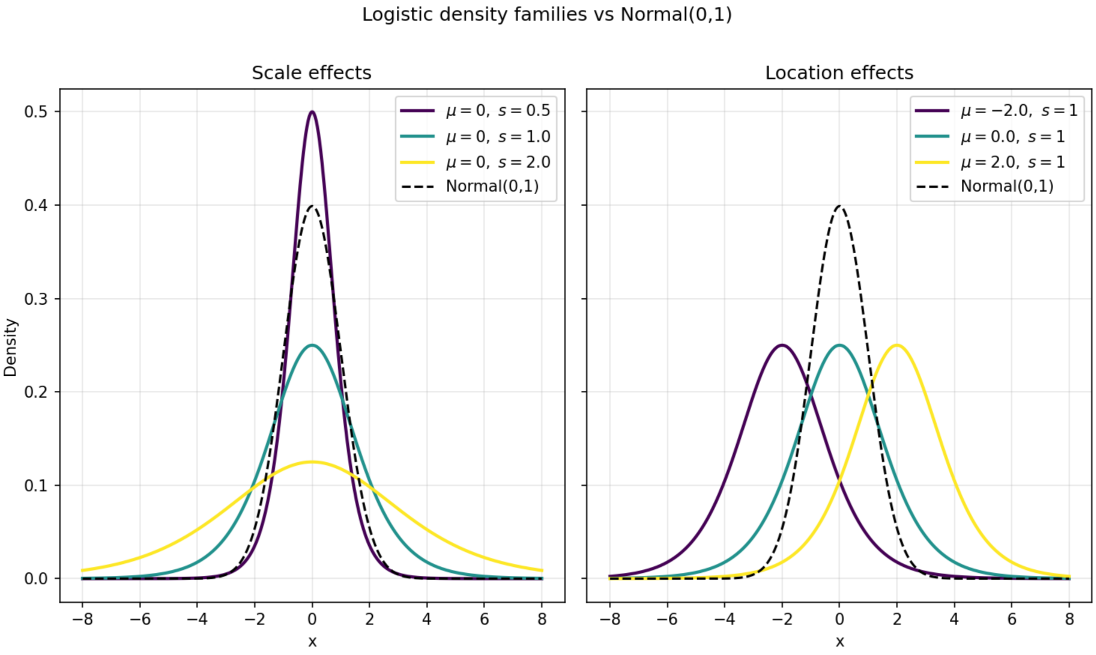

Figure 1: Logistic density families with varying scale (left) and location (right), compared to the Normal$(0, 1)$ curve.

The logistic cumulative distribution translates directly into the logit link $\operatorname{logit}(p) = \log\left(\frac{p}{1-p}\right)$, so the same curves describe how probabilities respond to linear predictors in generalized linear models. Because the tails decay more slowly than Gaussian tails, logistic likelihoods and the cross-entropy loss they induce in classification are more forgiving of outlying observations, which is why the distribution is so widely used in modern machine learning.

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. The Fisher scoring updates numerically satisfy the score equations $\sum_i \tanh\big((X_i - \hat{\mu})/(2\hat{s})\big) = 0$ and the weighted-average form for $\hat{s}$, so the algorithm recovers exactly the analytic solution implied by the derivations in the project handout.

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. Figure 2 displays the bootstrap mean distribution with both intervals marked for visual comparison.

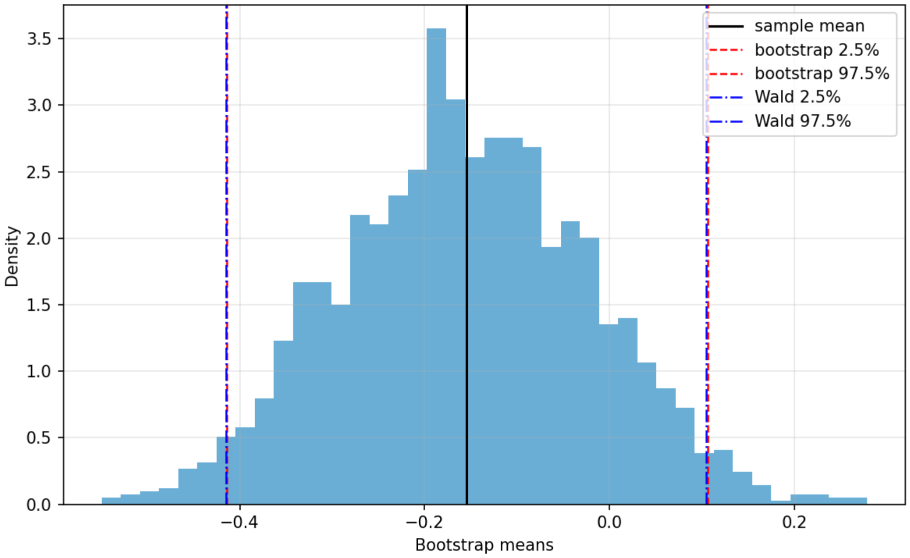

Figure 2: Bootstrap distribution of the sample mean with percentile (red dashed) and Wald (blue dash-dot) interval markers, plus the sample mean (black).

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 $r_{1,3}$. Under the Gaussian model $(n-1)\mathbf{S}$ is Wishart, so the sample correlation matrix has the analytical density described by Muirhead (1982). Figure 3 contrasts that theory with empirical diagnostics. 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 that the distribution is roughly symmetric with only slightly heavier tails than the Fisher-z curve predicts.

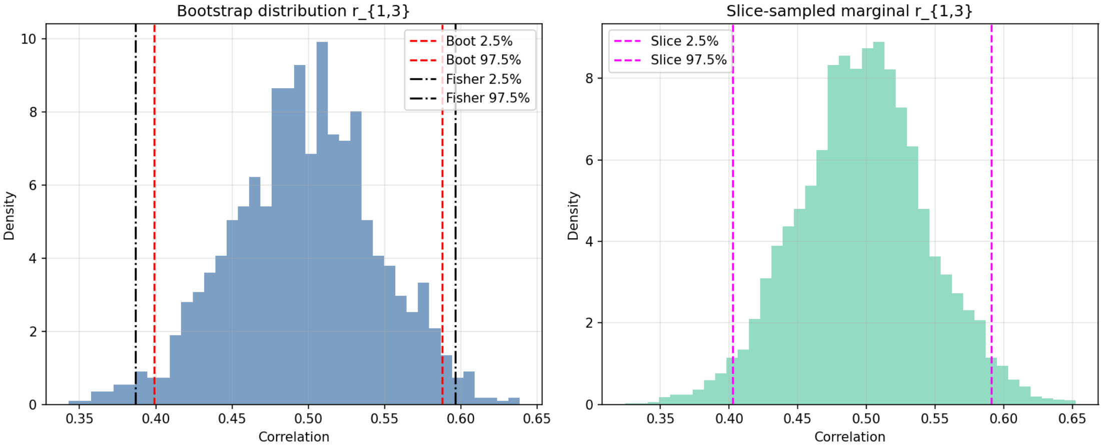

Figure 3: Bootstrap and slice-sampled uncertainty for correlation $r_{1,3}$.

The bootstrap draws also make it easy to revisit the Project 1 question of which simple distribution matches the sampling variability. Fitting a Gaussian to the Fisher-z transform yielded $\mu = 0.548$ and $\sigma = 0.064$ with Kolmogorov–Smirnov $D = 0.018$ and $p = 0.702$. Mapping correlations to $(0, 1)$ and fitting Beta(α, β) gave $\alpha = 246.828$ and $\beta = 82.891$ with $D = 0.019$ and $p = 0.624$. Both fits are acceptable, but the Beta model captures the mild skew a bit better (slightly larger p-value), consistent with the shape seen in the histogram.

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.