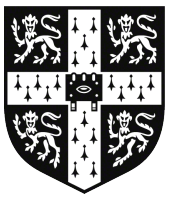


UNIVERSITY OF CAMBRIDGE

S1 Coursework Assignment - Report

Jacob Tutt (JLT67)

Department of Physics, University of Cambridge

December 18, 2024

Word Count: 2996

1 Introduction

The motivation of this report is to compare the statistical power and performance of two common methods for fitting distributions in experimental data analysis. It contrasts a multidimensional Extended Maximum Likelihood Estimate (MLE) with an 'sWeighted' fit, which isolated the Signal distribution in the control variable using fits from the independent variable [2].

1.1 True 2-Dimensional Model

This paper examines a two dimensional truncated statistical model, $X \in [0, 5]$ and $Y \in [0, 10]$, in which the signal $s(X, Y)$ and background $b(X, Y)$ models are independent across X and Y:

$$\begin{aligned} f(X, Y) &= fs(X, Y) + (1 - f)b(X, Y) \\ &= fg_s(X)h_s(Y) + (1 - f)g_b(X)h_b(Y). \end{aligned} \tag{1}$$

where:

- f is the signal fraction,
- $g_s(X)$, $h_s(Y)$ are the signal distributions,
- $g_b(X)$, $h_b(Y)$ are the background distributions.

Signal Distribution:

- X component, $g_s(X)$, uses the Crystal Ball function:

$$p(X; \mu, \sigma, \beta, m) = N \cdot \begin{cases} e^{-Z^2/2} & \text{for } Z > -\beta, \\ \left(\frac{m}{\beta}\right)^m e^{-\beta^2/2} \left(\frac{m}{\beta} - \beta - Z\right)^{-m} & \text{for } Z \leq -\beta. \end{cases} \tag{2}$$

  where $Z = \frac{X-\mu}{\sigma}$, and N is the normalisation constant. μ, σ, β, and m define the mean, width, threshold, and tail exponent.

- Y component, $h_s(Y)$, is an exponential decay:

$$h_s(Y) = \lambda e^{-\lambda Y}. \tag{3}$$

  where λ is the decay constant.

Background Distribution:

- X component, $g_b(X)$, is the uniform distribution:

$$g_b(X) = \begin{cases} \frac{1}{X_{\max} - X_{\min}} & \text{for } X_{\min} \leq X \leq X_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- Y component, $h_b(Y)$, is modeled as a normal distribution:

$$h_b(Y) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{(Y-\mu_b)^2}{2\sigma_b^2}}. \tag{5}$$

  where μ_b, σ_b are the mean and width of the distribution.

2 Part a - Normalisation of Crystal Ball Distribution

This provides a derivation of the normalisation constant, N, for the Crystal Ball function, Eq. (2). The overarching goal is to enforce:

$$\int_{-\infty}^{\infty} p(X; \mu, \sigma, \beta, m)\, dX = 1 \tag{6}$$

First, the integral is rewritten with respect to the transformation, Z:

$$Z = \frac{X - \mu}{\sigma}, \quad dZ = \frac{1}{\sigma} dX, \quad dX = \sigma dZ, \tag{7a}$$

$$\int_{-\infty}^{\infty} p(X; \mu, \sigma, \beta, m)\, dX = \sigma \int_{-\infty}^{\infty} p(Z; \mu, \sigma, \beta, m)\, dZ. \tag{7b}$$

Splitting the integral into the two sides:

$$\int_{-\infty}^{\infty} p(Z; \mu, \sigma, \beta, m)\, dZ = \sigma \left(\int_{-\infty}^{-\beta} p(Z; \mu, \sigma, \beta, m)\, dZ + \int_{-\beta}^{\infty} p(Z; \mu, \sigma, \beta, m)\, dZ \right). \tag{8a}$$

The left-hand side of the integral ($Z \leq -\beta$):

$$\int_{-\infty}^{-\beta} p(Z; \mu, \sigma, \beta, m)\, dZ = \int_{-\infty}^{-\beta} \left(\frac{m}{\beta}\right)^m e^{-\beta^2/2} \left(\frac{m}{\beta} - \beta - Z\right)^{-m} dZ, \tag{9a}$$

$$= \left[\frac{1}{m-1} \left(\frac{m}{\beta}\right)^m e^{-\beta^2/2} \left(\frac{m}{\beta} - \beta - Z\right)^{-(m-1)} \right]_{-\infty}^{-\beta}, \tag{9b}$$

$$= \frac{1}{m-1} \left(\frac{m}{\beta}\right)^m e^{-\beta^2/2} \left(\frac{m}{\beta}\right)^{-(m-1)}, \tag{9c}$$

$$= \frac{1}{m-1} \left(\frac{m}{\beta}\right) e^{-\beta^2/2}. \tag{9d}$$

As $\left(\frac{m}{\beta} - \beta - Z\right)^{-(m-1)} \to 0$ as $Z \to -\infty$.

The right-hand side (Gaussian component) ($Z > -\beta$):

$$\int_{-\beta}^{\infty} p(Z; \mu, \sigma, \beta, m)\, dZ = \int_{-\beta}^{\infty} e^{-Z^2/2} dZ = \sqrt{2\pi}\Phi(\beta), \tag{10a}$$

Where $\Phi(x)$ is the CDF of the standard normal distribution:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt. \tag{11}$$

Recombining the left and right integrals:

$$1 = \sigma \left(\frac{1}{m-1} \left(\frac{m}{\beta}\right) e^{-\beta^2/2} + \sqrt{2\pi}\Phi(\beta) \right). \tag{12a}$$

Rewriting Eq. (12a), N is given by:

$$N = \sigma \left(\frac{m}{\beta(m-1)} e^{-\beta^2/2} + \sqrt{2\pi}\Phi(\beta) \right)^{-1}. \tag{13a}$$

3 Part b - Defining the Distributions

3.1 Object-Oriented Architecture

The program's uses an object-oriented structure, shown in Figure 1, to improve the flexibility for future component distributions by increasing modularity. The majority of analysis is defined within the classes hence documentation with source code and docstrings is provided for easier insights into its methods.

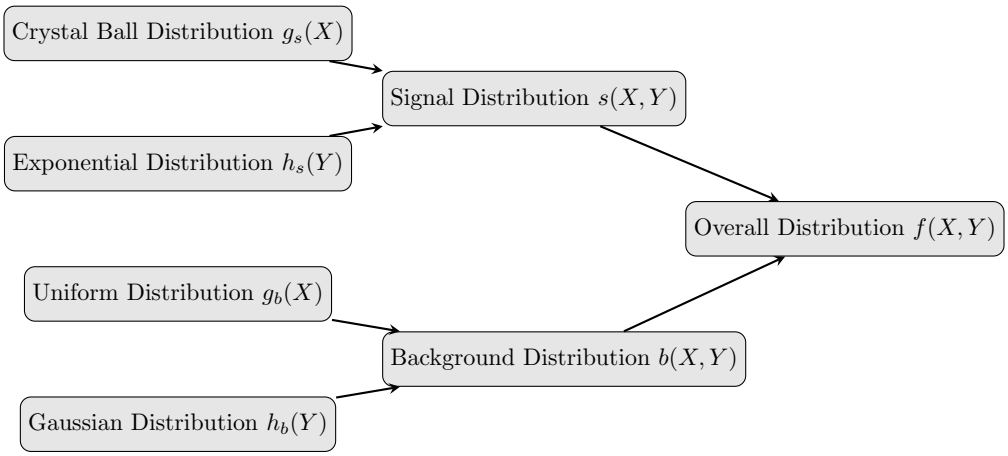


Figure 1: Inheritance structure of the object-oriented architecture

3.2 Applying Truncation

The distributions PDF's and CDF's are implemented using scipy's distributions; truncating and renormalising them for $X \in [0, 5]$ and $Y \in [0, 10]$.

Truncated PDF:

$$f_{\text{trunc}}(x) = \begin{cases} \frac{f(x)}{F(x_{\max}) - F(x_{\min})}, & x \in [x_{\min}, x_{\max}], \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Truncated CDF:

$$F_{\text{trunc}}(x) = \begin{cases} \frac{F(x) - F(x_{\min})}{F(x_{\max}) - F(x_{\min})}, & x \in [x_{\min}, x_{\max}], \\ 0, & x < x_{\min}, \\ 1, & x > x_{\max}. \end{cases} \tag{15}$$

where:

- $f(x)$: Original PDF
- $F(x)$: Original CDF
- $[x_{\min}, x_{\max}]$: Range of truncation.

3.3 Initialising Distribution

The model defined in 1.1 is initialised with the following parameters:

μ	σ	β	m	f	λ	μ_b	σ_b
3.0	0.3	1.0	1.4	0.6	0.3	0.0	2.5

Table 1: True model parameters' values

Plots of each distribution's PDFs and CDF's are shown in below:

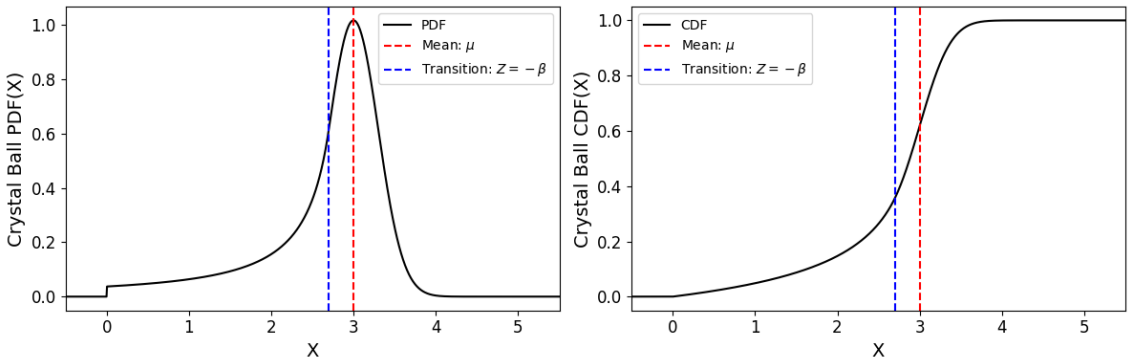


Figure 2: Truncated Crystal Ball Distribution ($g_s(X)$)

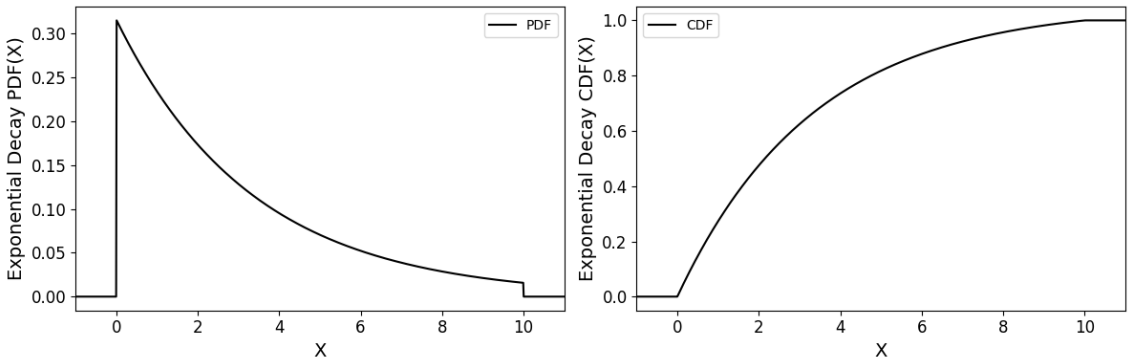


Figure 3: Truncated Exponential Distribution ($h_s(Y)$)

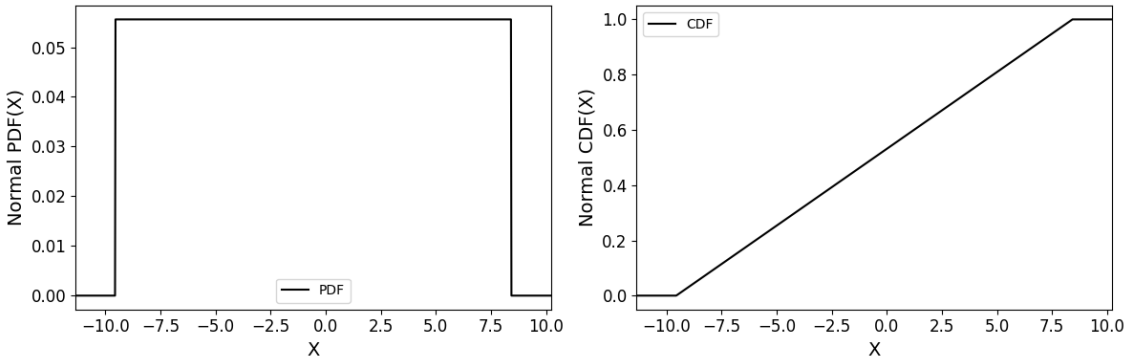


Figure 4: Uniform Distribution ($h_b(Y)$).

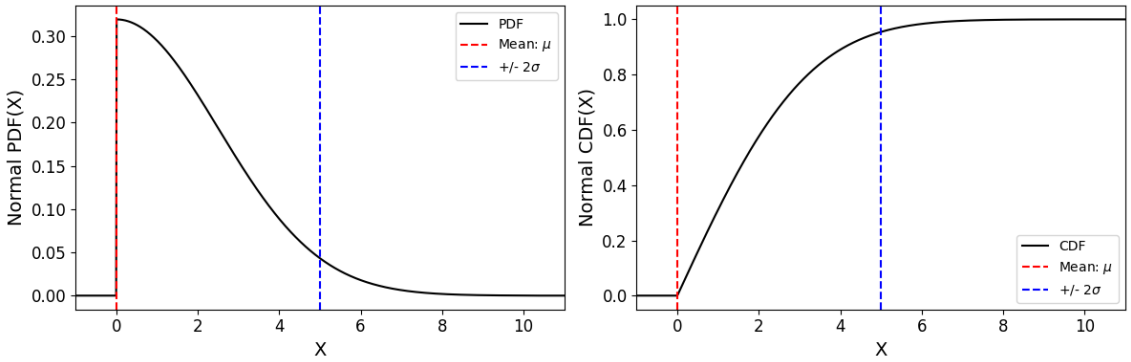


Figure 5: Truncated Normal Distribution($g_b(X)$)

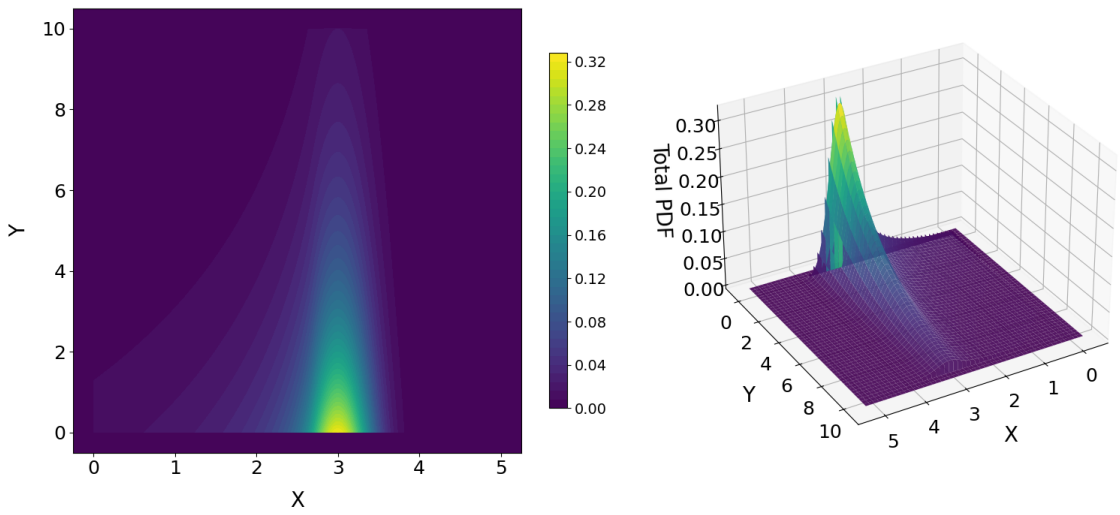


Figure 6: Signal Distribution ($s(X, Y)$)

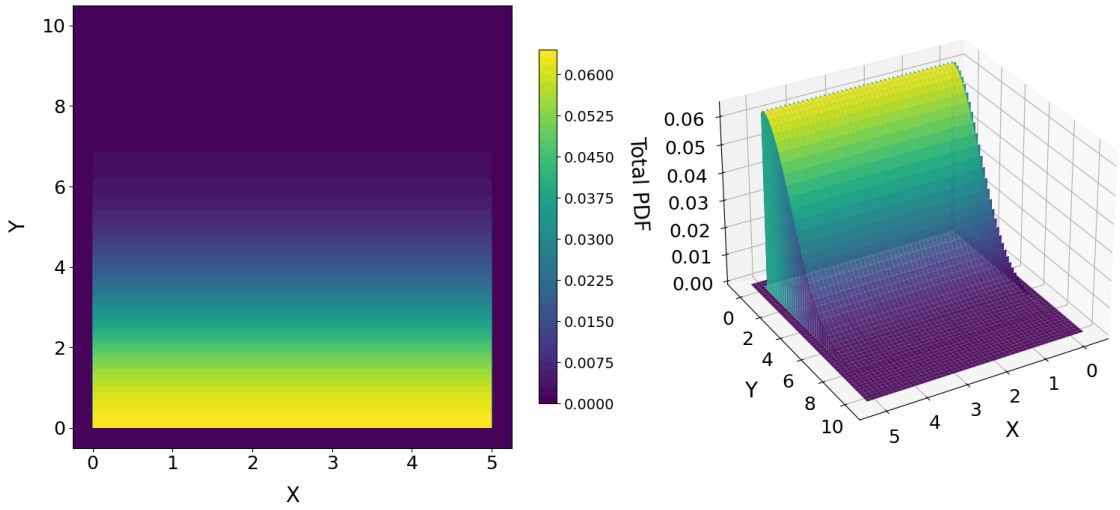


Figure 7: Background Distribution ($b(X, Y)$)

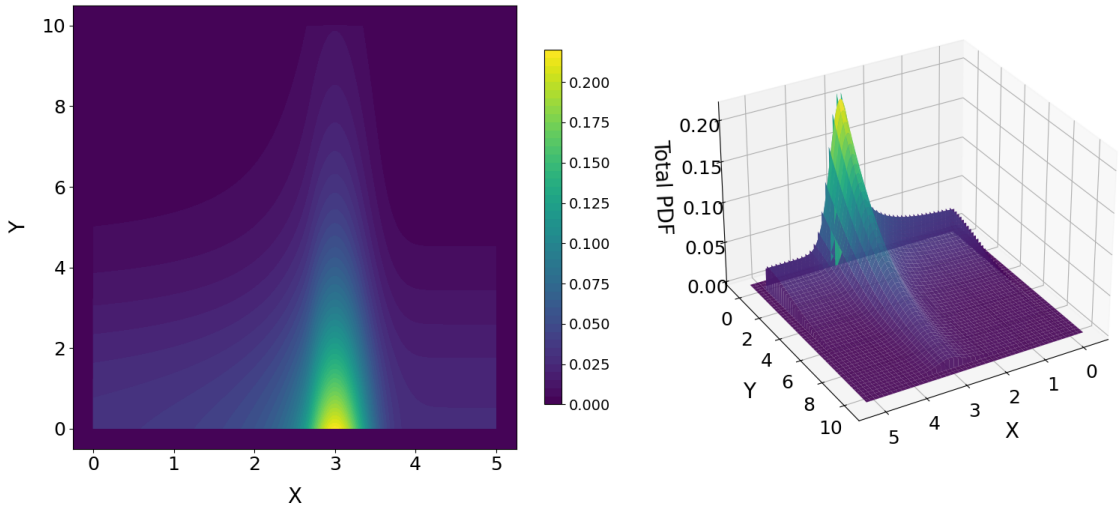


Figure 8: Overall Distribution ($f(X, Y)$)

3.4 Verifying Normalisation

The normalisation of the truncated distributions is verified by integrating the PDFs over their truncated ranges and all real space using scipy's numerical integration functions `quad` for 1D and `dblquad` for 2D distributions. Ensuring:

$$\int_{-\infty}^{\infty} f(x)\,dx = 1, \quad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,dx\,dy = 1. \tag{16}$$

The results are shown in Table 2, showing all integrals lie within the expected numerical integration error. The project's notebook's performs this for multiple other configurations to ensure robust normalisation for all cases.

Distribution	**Interval**	**Numerical Integral**
Crystal Ball $g_s(X)$	$X \in [0, 5]$	$1.0000004 \pm 7.3 \times 10^{-9}$
	$X \in [-\infty, \infty]$	$1.0000000 \pm 4.3 \times 10^{-9}$
Exponential $h_s(Y)$	$Y \in [0, 10]$	$1.0000000 \pm 1.1 \times 10^{-14}$
	$Y \in [-\infty, \infty]$	$1.0000000 \pm 1.1 \times 10^{-14}$
Uniform $g_b(X)$	$X \in [0, 5]$	$1.0000000 \pm 1.1 \times 10^{-14}$
	$X \in [-\infty, \infty]$	$1.0000000 \pm 3.9 \times 10^{-9}$
Normal $h_b(Y)$	$Y \in [0, 10]$	$1.0000000 \pm 4.8 \times 10^{-12}$
	$Y \in [-\infty, \infty]$	$1.0000000 \pm 1.2 \times 10^{-8}$
Signal $s(x, y)$	$X \in [0, 5], Y \in [0, 10]$	$1.0000004 \pm 7.3 \times 10^{-9}$
	$X \in [-\infty, \infty], Y \in [-\infty, \infty]$	$1.0000000 \pm 1.5 \times 10^{-8}$
Background $b(x, y)$	$X \in [0, 5], Y \in [0, 10]$	$1.0000000 \pm 9.7 \times 10^{-13}$
	$X \in [-\infty, \infty], Y \in [-\infty, \infty]$	$1.0000000 \pm 8.3 \times 10^{-9}$
Overall $f(x, y)$	$X \in [0, 5], Y \in [0, 10]$	$1.0000002 \pm 4.4 \times 10^{-9}$
	$X \in [-\infty, \infty], Y \in [-\infty, \infty]$	$1.0000000 \pm 1.5 \times 10^{-8}$

Table 2: Numerical integration results for each distribution and integration interval

4 Part c - Marginal Probabilities

Due to the signal and background distributions being independent in X and Y, the marginal probabilities are simply linear combinations of the component distributions and thus easily implemented by the object-oriented architecture. Derivations of each are outlined below, with the CDF's following the same logic:

Marginal PDF in X $f(X)$

Integral of $f(X, Y)$ over all X:

$$f(X) = \int_{-\infty}^{\infty} f(X,Y)\,dY, \tag{17a}$$

$$= f g_s(X) \int_{-\infty}^{\infty} h_s(Y)\,dY + (1-f) g_b(X) \int_{-\infty}^{\infty} h_b(Y)\,dY, \tag{17b}$$

$$= f g_s(X) + (1-f) g_b(X). \tag{17c}$$

Marginal PDF in Y $f(Y)$

Integral of $f(X, Y)$ over all X:

$$f(Y) = \int_{-\infty}^{\infty} f(X,Y)\,dX, \tag{18a}$$

$$= f h_s(Y) \int_{-\infty}^{\infty} g_s(X)\,dX + (1-f) h_b(Y) \int_{-\infty}^{\infty} g_b(X)\,dX, \tag{18b}$$

$$= f h_s(Y) + (1-f) h_b(Y). \tag{18c}$$

Marginal CDF in X $F(X)$

Integral of $F(X,Y)$ over all X:

$$F(X) = fG_s(X) + (1-f)G_b(X). \tag{19}$$

where:

- $G_s(X)$, $G_b(X)$: the Crystal Ball and Uniform Distribution's CDFs.

Marginal CDF in Y $F(Y)$

Integral of joint CDF $F(X,Y)$ over all Y:

$$F(Y) = fH_s(Y) + (1-f)H_b(Y). \tag{20}$$

where:

- $H_s(Y)$, $H_b(Y)$: the Exponential and Normal Distribution's CDFs.

4.1 Plotting the Marginal Distributions

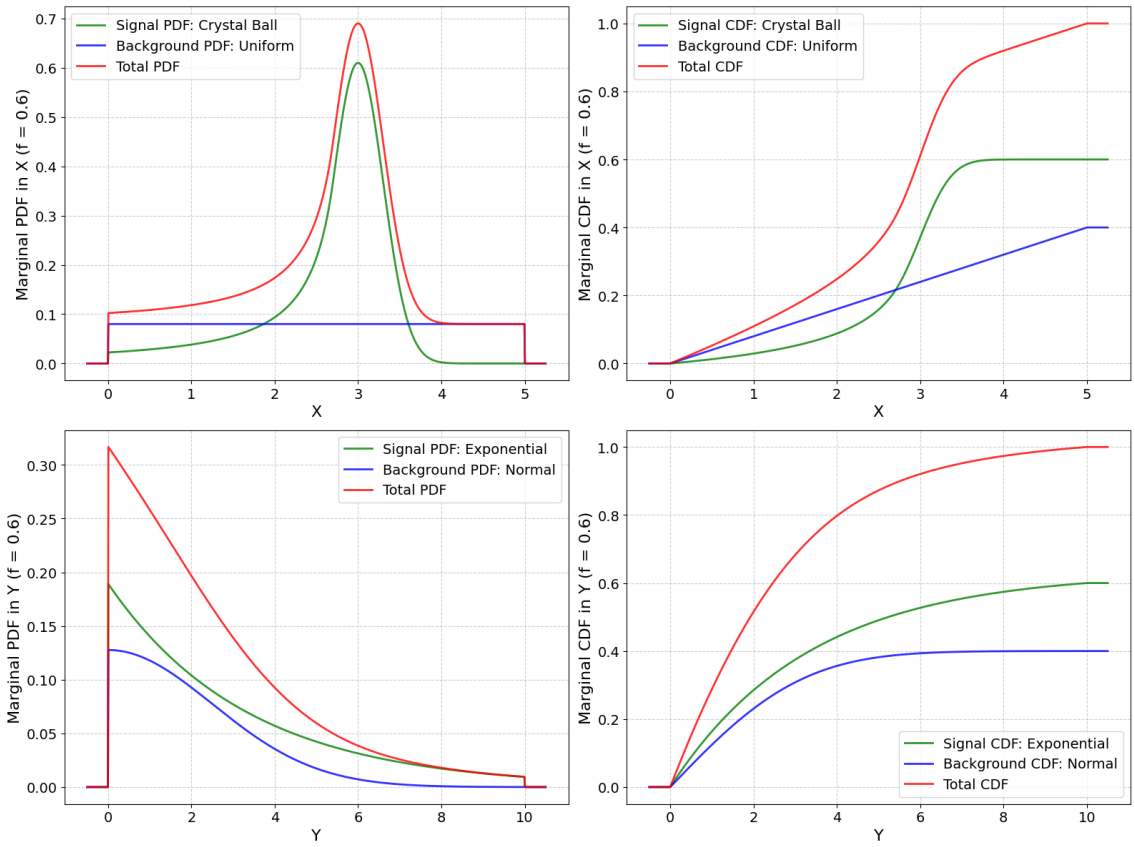


Figure 9: Marginal PDFs and CDFs of $f(X,Y)$: (Top) X marginal PDF and CDF combining Crystal Ball (signal) and Uniform (background). (Bottom) Y marginal PDF and CDF combining Exponential (signal) and Normal (background). Total distributions in red

5 Part d - Generating Samples and MLE estimate

5.1 Generating Samples Algorithm

This paper samples from the true 2D distribution $f(X,Y)$ using an accept-reject algorithm, which aims to generate the desired number of samples in as few vectorised batches as possible. The algorithm also determines the PDF's maximum value upon initialisation allowing it to restrict the randomly generated samples and increase the acceptance rate significantly. The algorithm is outlined below:

1. Generate an initial batch of samples (default 1000) uniformly over $X \in [0, 5]$, $Y \in [0, 10]$ and $Z \in [0, f_{max}]$, and perform an accept-reject step.

2. Estimate the acceptance rate:

$$\text{Acceptance Rate} = \frac{\text{Number of Accepted Samples}}{\text{Total Number of Samples}}. \tag{21}$$

3. Scale the next batch's sample size to generate the remaining number of samples (slightly overestimating to counter statistical fluctuations). A maximum batch size (default 2 million) also prevents memory overflow:

$$\text{Batch Size} = \min\left(\frac{1.1 \times \text{Number of Remaining Samples}}{\text{Acceptance Rate}}, \text{Max Batch Size}\right) \tag{22}$$

4. Perform the accept-reject step on the new batch.

5. Repeat steps 3-4 until the desired number of samples is generated.

This method is tested with 100,000 samples, achieving an acceptance rate of ≈ 0.107, consistent with the maximum PDF value (≈ 0.218). For all tests, the algorithm successfully generated the samples in 2 batches (initial batch inclusive). The generated samples' are plotted in Figure 10, showing strong agreement with the true distribution.

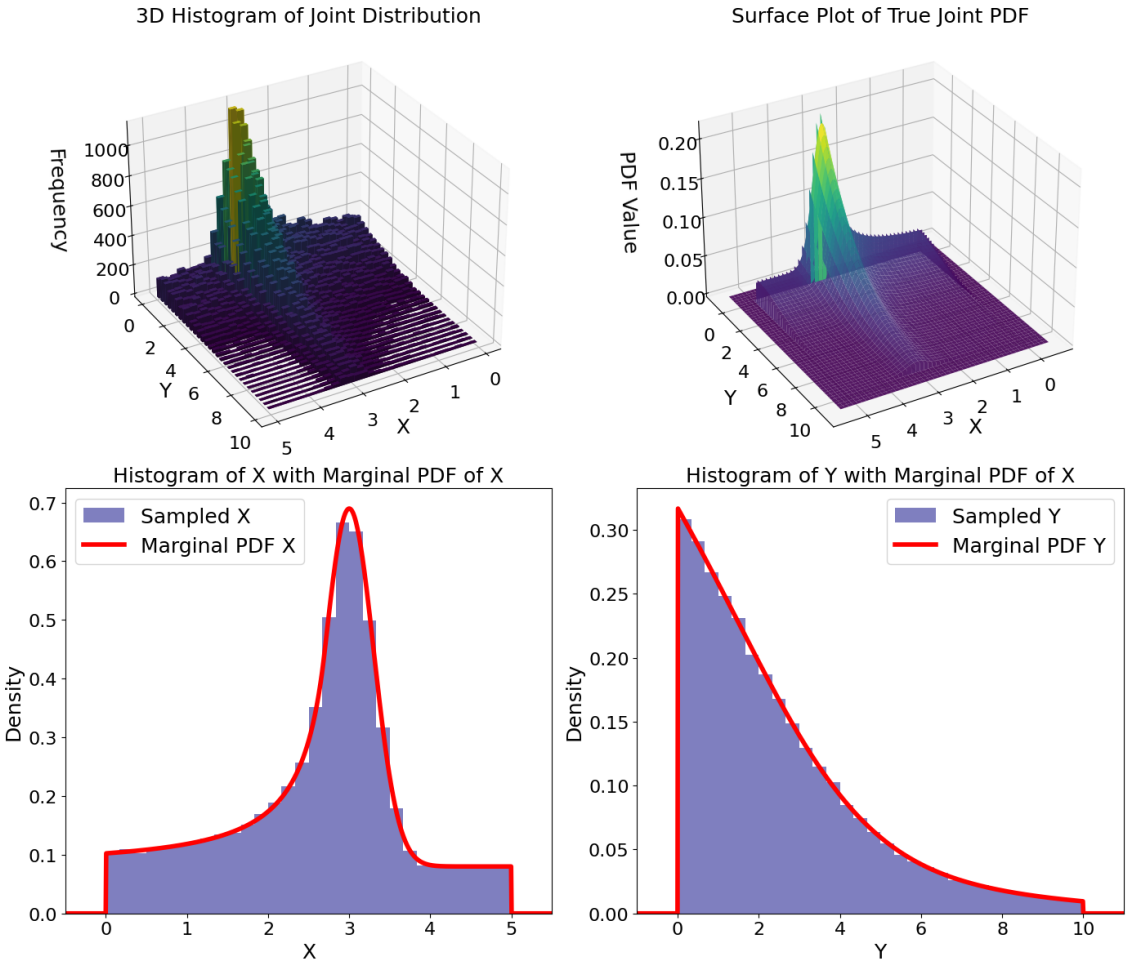


Figure 10: Verification of sampling method: (Top Left) Sampled joint distribution. (Top Right) True joint PDF. (Bottom Left) Sampled X values with marginal PDF. (Bottom Right) Sampled Y values with marginal PDF

5.2 Fitting the Samples using Extended MLE

The extended MLE fitting method is implemented by minimising the negative log-likelihood using the iminuit package's 'migrad' function. Initial parameter guesses are chosen to be offset from the true values for verification of the method's convergence and are kept constant throughout the report to allow the comparability of convergence speeds. Additionally, iminuit is provided with parameter limits to restrict the search space to physically meaningful parameter values and reduce run time.

Parameters	μ	σ	β	m	f	λ	μ_b	σ_b
Initial Value	3.1	0.4	1.1	1.5	0.7	0.4	0.1	265
Parameter Limits	N/A	[0,]	[0,]	[1,]	[0, 1]	[0,]	N/A	[0,]

Table 3: Initial parameter guesses and search limits

The parameters' uncertainties are first evaluated using the 'hesse' function, which exploits the inverse Hessian of the likelihood at its minimum. This method although computationally efficient within the iminuit package, assumes the uncertainties are a multivariate Gaussian distribution. This assumption breaks down for poorly constrained parameters, especially near parameter boundaries. The method hence requires verification using the more robust profiled log-likelihoods to determine uncertainties from ($-\Delta lnL = 0.5$). The results of the fitting and both uncertainty calculations are shown in Table 4.

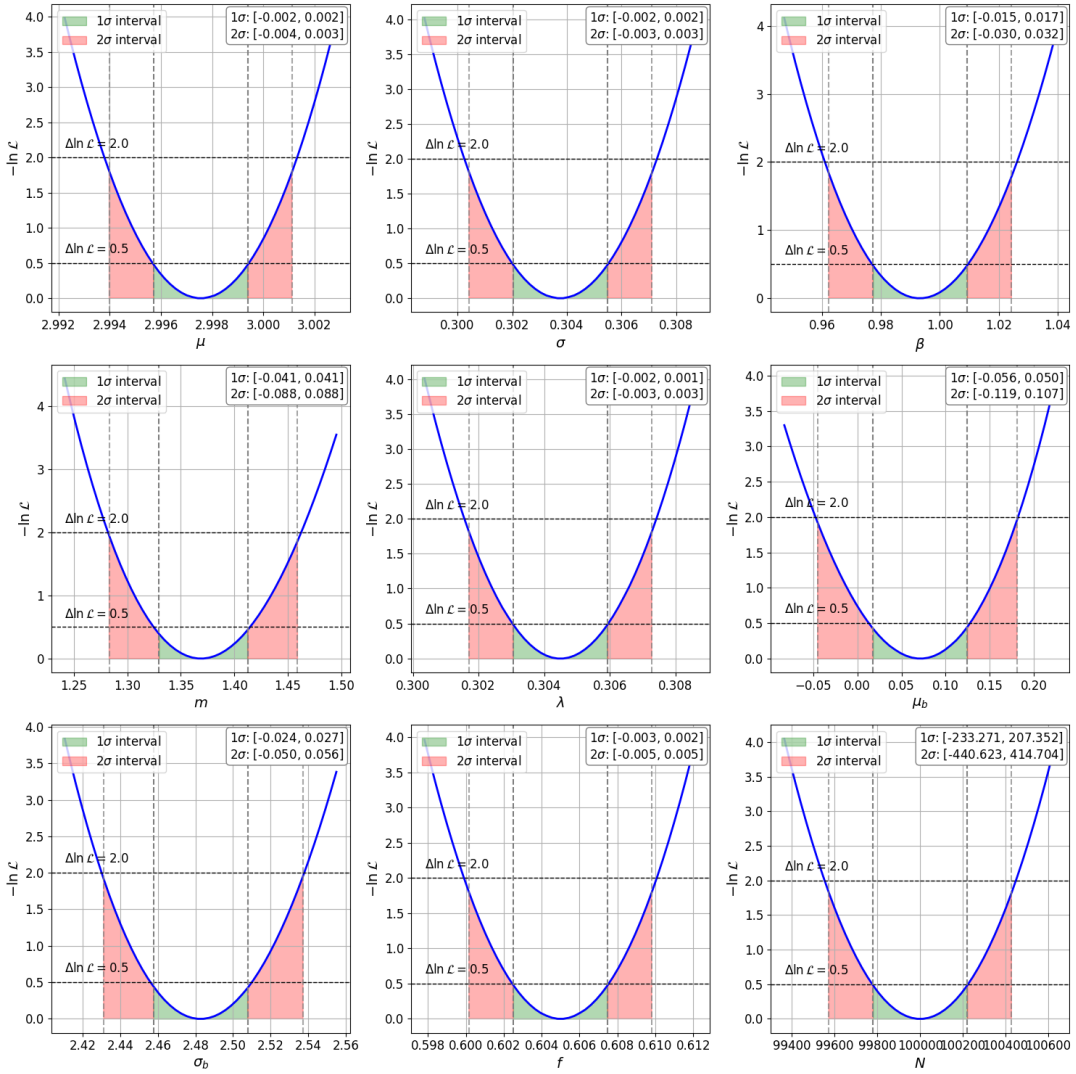


Figure 11: Profile likelihoods for each parameter. Highlighted are the 1σ (green, $\Delta \ln \mathcal{L} = 0.5$) and 2σ (red, $\Delta \ln \mathcal{L} = 2.0$) confidence intervals

Parameter	Value ± Error(Hessian)	True Value	Std Errors Away	Profile Lower Bound	Profile Upper Bound
μ	2.9975 ± 0.0027	3	0.92	-0.002	0.002
σ	0.3038 ± 0.0025	0.3	1.51	-0.002	0.002
β	0.9933 ± 0.0231	1	0.29	-0.015	0.017
m	1.3685 ± 0.0635	1.4	0.50	-0.041	0.041
λ	0.3045 ± 0.0021	0.3	2.17	-0.002	0.001
μ_b	0.0710 ± 0.0769	0	0.92	-0.056	0.050
σ_b	2.4827 ± 0.0362	2.5	0.48	-0.024	0.027
f	0.6050 ± 0.0036	0.6	1.39	-0.003	0.002
N	100000 ± 320	100000	0	-440.63	414.70

Table 4: Parameter fit results, including estimates, uncertainties (Hessian), true values, and bounds

From Table 4, the fitting method provides highly accurate estimates with a sample size of 100,000. The uncertainties derived from the Hessian method, while similar to the profile method, are slightly overestimated for all parameters (except N), likely due to the large sample size tightly constraining parameters. Thus narrowing the likelihood beyond the Hessian's quadratic approximation. Nonetheless, the minimal over-coverage of the Hessian method does not justify the higher computational cost of the profile likelihood method, especially for smaller bootstrap sample sizes (up to 10,000), as used in Section 6. As we move on to analyse the suitability of the fitting method, with a focus on λ, the corrolation matrix from the Hessian method (Fig 12) shows λ is largely uncorrelated with the other parameters, with the strongest being 0.28 with f. However, it should be noted for future analysis the strong correlations among the Crystal Ball and Normal distribution parameters which may cause issues at lower sample sizes.

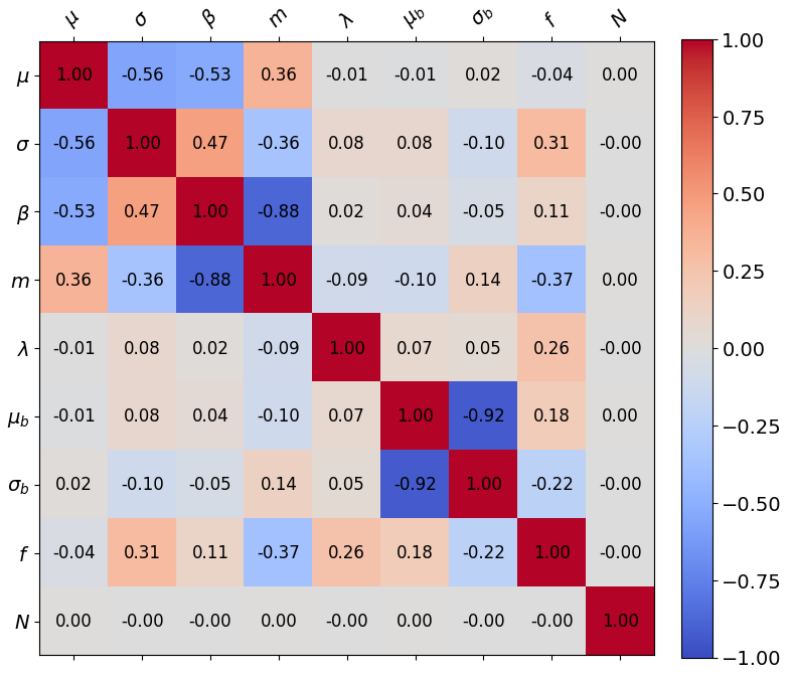


Figure 12: Correlation matrix of the fitted parameters

5.3 Computational Preformance

To determine the computational performance the runtime of generating and fitting 100,000 samples is averaged for 100 calls and reported below. Notably for sample generation, a single initial batch of circa 1 million samples, would result in a slower run time however the algorithm prioritises minimising it for a range of distributions and sample sizes.

Process	Average Time (s)	Relative Runtime
Benchmark Process	0.001542	1
Accept Reject Sampling (100,000)	0.183773	119.202
Parameter Fitting	6.582080	4269.37

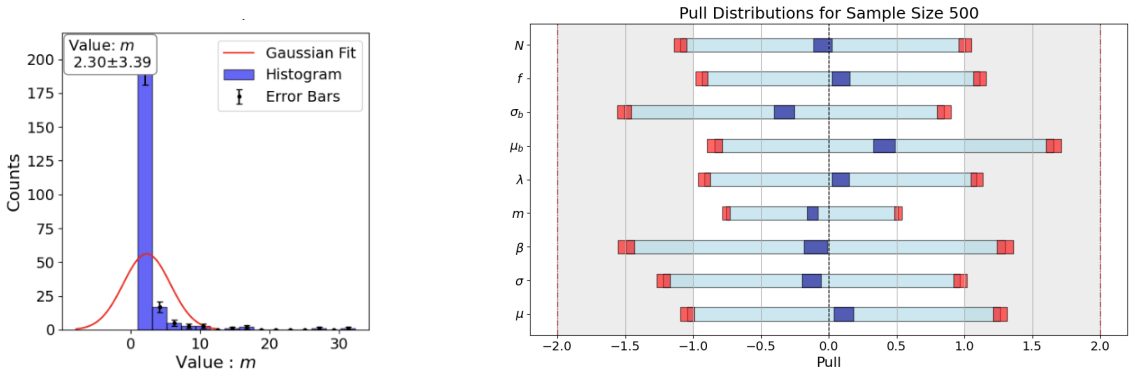
Table 5: Execution times and relative runtimes for 100,000 samples

6 Part e - Parametric Bootstrap

The accuracy and robustness of the Extended MLE fitting method are further evaluated using a parametric bootstrap, generating and refitting samples from the true distribution. This report examines the method's performance across Poisson-varied sample sizes of 500, 1000, 2500, 5000, and 10000, each using 250 toys. Notably, during bootstrapping for the sample size of 500 data points, an average of 12 out of 250 toys were unable to converge to a minimum. This suggests certain parameters were unable to be sufficiently constrained, a topic discussed further within this section.

This section of the paper predominantly focuses on the fitting of λ, however other parameters' plots are within the project's 'Bootstrap' directory. First, we look at the distributions of λ's fitted values, Hessian errors, and pulls across the 250 toys, shown in Figure 13. The parametric bootstrap can be seen to create a normal distribution of fitted values centered around the true value (with bias' discussed later). Importantly, despite λ being constrained ($\lambda \geq 0$), even at a sample size of 500, the fitted values remain sufficiently constrained from the parameter's boundary. Thus, the Hessian's Gaussian assumption is valid and will likely provide an accurate quantification of the uncertainty.

On the other hand, the MLE method struggles to constrain m away from its boundary ($m \geq 1$), even at sample sizes of 10,000, leading to stacking at the boundary. This causes the hessian error for m to overcover as it fails to account for the truncation of the likelihood surface and an overly narrow pull distribution. Figure 14 illustrates this issue for a sample size of 500. In studies more focused on m, the computationally demanding Feldman-Cousins method would provide a greater understanding of coverage near the boundary.

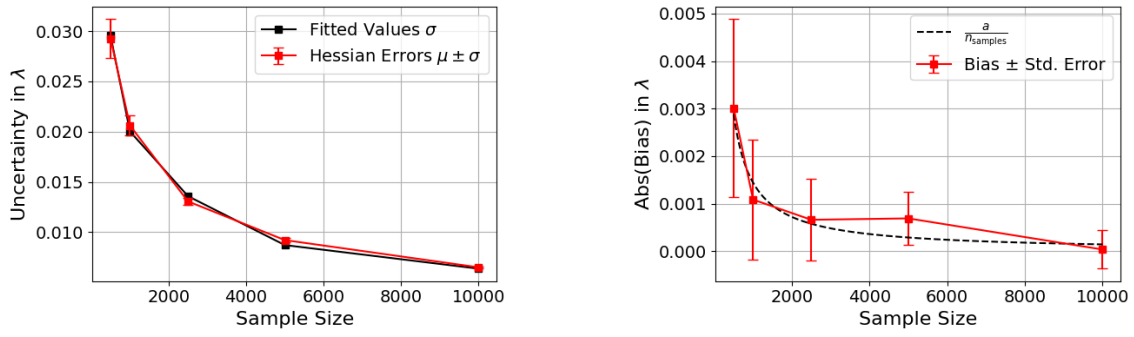


(a) Fitted m distribution

(b) Pull distribution for all parameters

Figure 14: Overcoverage from stacking at the parameter boundary for m (sample size 500)

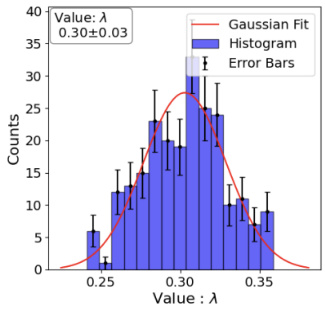
In Figure 13, as the sample size increases, the width of the fitted λ distribution narrows, and the Hessian errors' distribution shifts closer to zero. Finally, to evaluate the Hessian's accuracy in quantifying the true uncertainty of the fit, its mean value (and standard deviation) across all toys is compared to the standard deviation of the fitted values, as shown in Figure 15a.



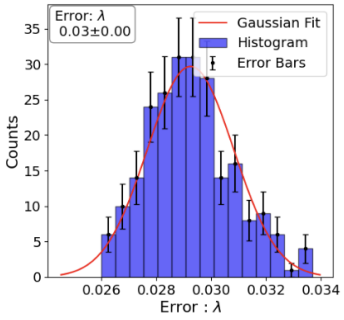
(a) Hessian errors vs. standard deviation of λ

(b) Biases in $\lambda \pm$ Std. Error

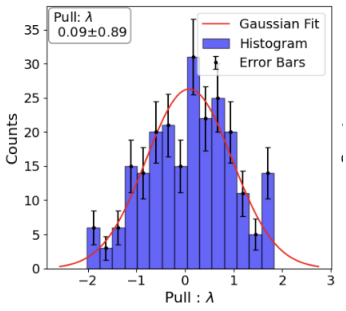
Figure 15: Comparison of bias and uncertainty in fitted λ values across sample sizes



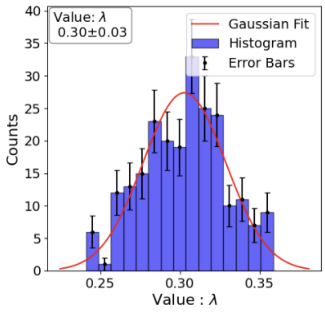
(a) Fitted Values - 500



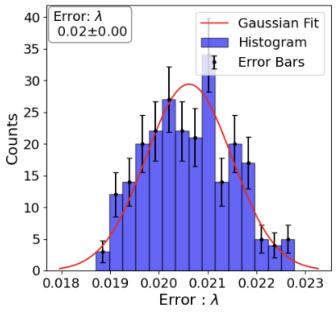
(b) Hessian Errors- 500



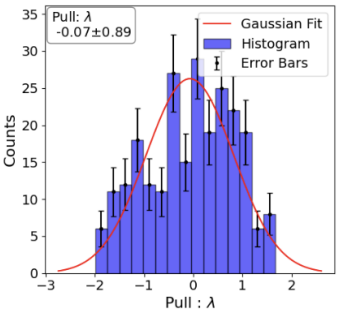
(c) Pulls - 500



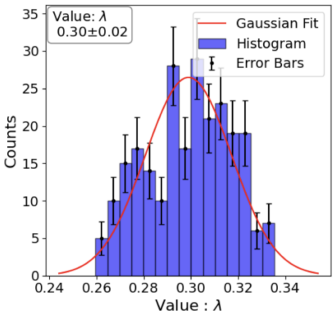
(d) Fitted Values - 1000



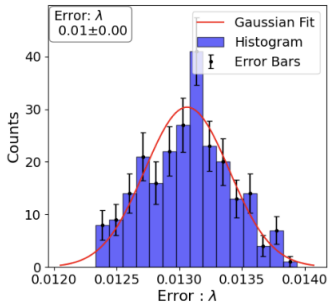
(e) Hessian Errors - 1000



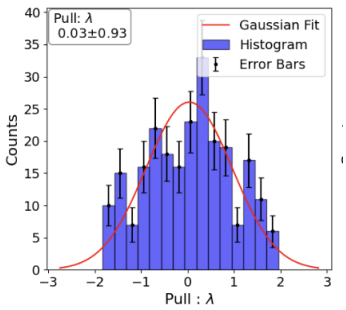
(f) Pulls - 1000



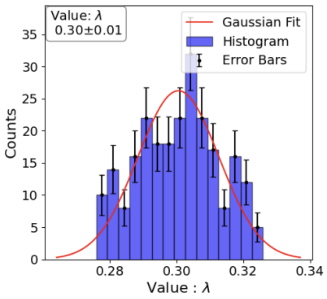
(g) Fitted Values - 2500



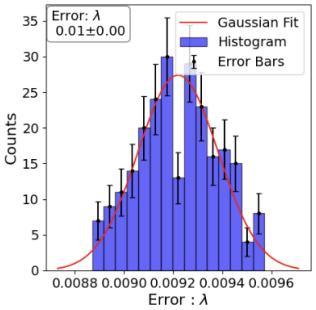
(h) Hessian Errors - 2500



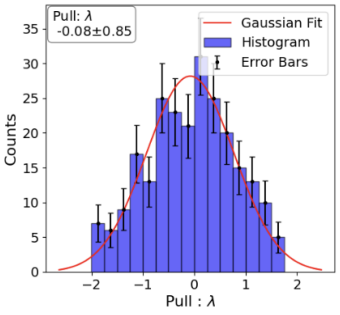
(i) Pulls - 2500



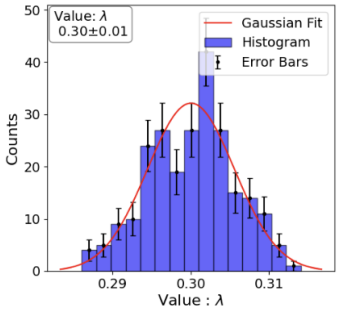
(j) Fitted Values - 5000



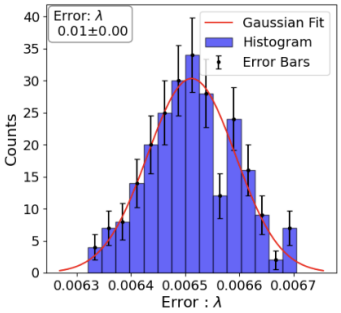
(k) Hessian Errors - 5000



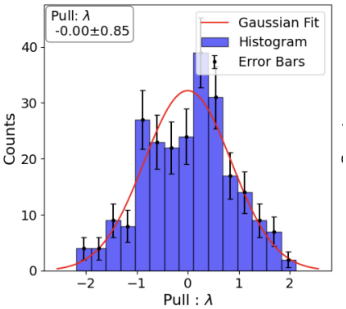
(l) Pulls - 5000



(m) Fitted Values - 10000



(n) Hessian Errors - 10000



(o) Pulls - 10000

Figure 13: Histograms of parameter values, Hessian errors, and pull distributions across all sample sizes

From Figure 15a, the overall fit uncertainty decreases with sample size, as expected. The Hessian errors closely match the standard deviation of the fitted λ values across all sample sizes, with a small spread (σ) indicating reliable uncertainty estimation. Alternatively, for parameters m , μ_b, and σ_b (Figure 22) the Hessian errors, while centered around the standard deviation of the fitted values, exhibit a much greater spread, making them less reliable. This also happens, although to a lesser extent, to the other Crystal Ball parameters, likely as a result of the poor fitting for m having knock-on effects and its previously discussed high correlation with these parameters. Having accessed the uncertainty provided by the Hessian, the MLE's inherent bias is investigated:

$$\text{Bias}(\hat{\theta}) \approx \bar{\hat{\theta}}_{\text{bootstrap}} - \theta = \left(\frac{1}{N} \sum_{i=1}^{N} \hat{\theta}_i \right) - \theta, \quad \text{Std. Error} = \frac{1}{\sqrt{n_{\text{toys}}}} \cdot \text{StdDev}(\hat{\theta}). \tag{23}$$

where:

- $\hat{\theta}$, fitted parameter value,
- θ, true parameter value.

Figure 15b, shows the bias on the fitted $\bar{\hat{\lambda}}$ decreases inversely proportionally, a trend seen for all parameters (Figure 22). This suggests systematic errors in the MLE method, especially for low sample sizes. This agrees with literature, where MLE is documented to be biased for finite samples, with $\mathcal{O}(n^{-1})$ shown by the fitted $\frac{1}{n}$ curve [4]. Despite this, the bias is approximately an order of magnitude smaller than the fit's uncertainty for all sample sizes, and thus not the limiting factor for the accuracy.

Finally, the bias and uncertainty of λ can be summarised using pull distributions (Figure 16) showing decreasing bias but correct coverage for all samples.

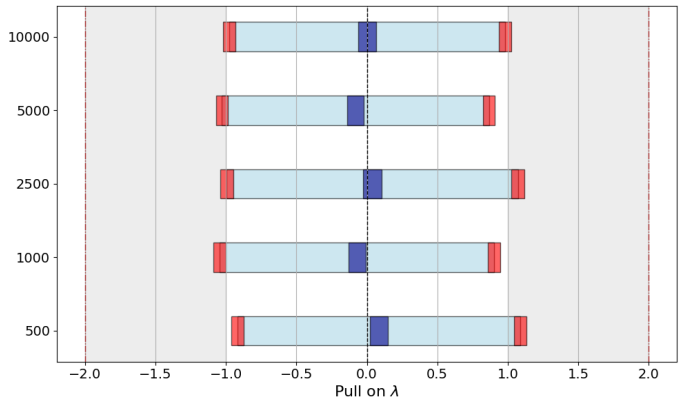


Figure 16: Pull distributions for λ across all sample sizes, with the mean (blue) and standard deviation (red), including uncertainties

7 Part f - sWeight Projection

This report investigates a second method that determines signal weights in the X dimension through a one-dimensional Extended MLE fit. These weights are then used to project the signal component into the Y dimension, isolating it from the background. This approach thus enables an isolated fit of the signal distribution in Y without background interference. This procedure is implemented using a combination of the 'iminuit' and 'sWeights' packages. At first, a single generated sample of 10,000 is used to demonstrate the method's validity before applying it to all bootstrap samples:

Step 1 - Fitting for the overall X dimension

The first step uses an extended unbinned MLE fit to determine the parameters in the X dimension, hence reducing the parameter dimensions to 6. The fitting method is the same as described in Section 5 and for clarity, the relevant parameters, guesses, and limits are shown in Table 6. The additional parameter N is the expected sample count and its initial guess is taken as the sample size generated from.

Parameters	μ	σ	β	m	f
Initial Value	3.1	0.4	1.1	1.5	0.7
Parameter Limits	N/A	[0,]	[0,]	[1,]	[0, 1]

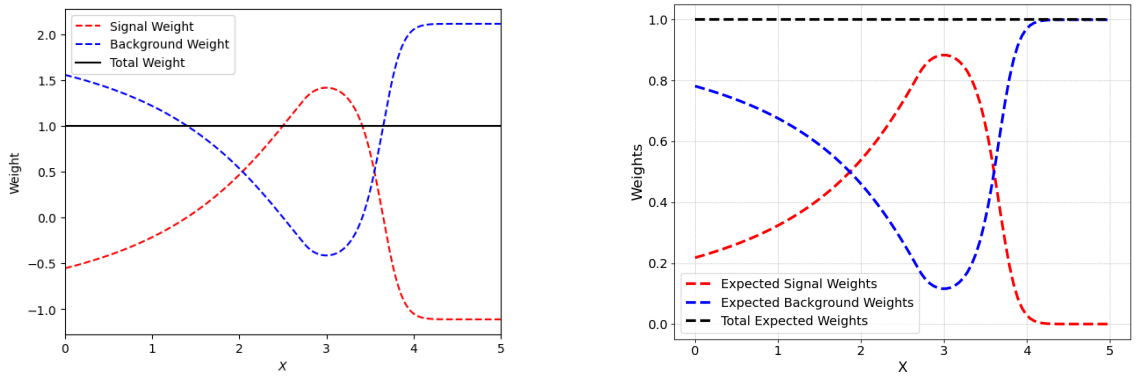
Table 6: Initial parameter values and search limits for sWeights Step 1

The results seen for the respective parameters, Table 8, are very comparable to those in Section 5.

Step 2 - Determining the sWeights and projecting signal in Y

The second step calculates the weight for each data point to be a signal event based on the X dimension's fitted distribution, using the 'sWeights' package. The distribution of weights is verified through comparison with the true fractional signal and background components, calculated from the X dimension's marginal PDFs defined in Section 4. Although exact scaling cannot be compared due to the sWeights being normalised to the fitted sample sizes N_s and N_b, the shape of the distribution can seen to strongly agree (Figure 17).

$$N_s = N_{fitted} \times f_{fitted} \qquad N_b = N_{fitted} \times (1 - f_{fitted}) \tag{24}$$



(a) Weights of signal and background in X-dimension from sWeights

(b) True weights of signal and background in X-dimension from marginal PDFs

Figure 17: Comparison of true and sWeights-determined distributions in the X-dimension

The weights are then applied to each associated Y variable, scaling its contribution to the signal component. The signal projection is plotted below with the true signal distribution (scaled for the samplesize), showing a very strong agreement between them (Figure 18).

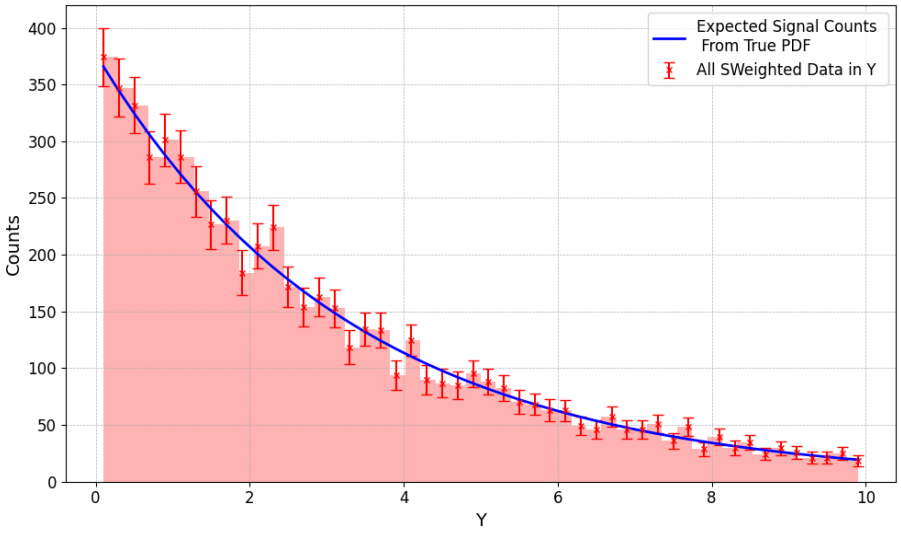


Figure 18: Signal projection in the Y-dimension using sWeights

7.0.1 Step 3 - Fitting the projected signal in Y

The final step uses a binned MLE fit on the projected signal in Y to determine λ, using an initial guess and limits consistent with past fits (Table 7).

Parameter	λ
Initial Value	0.4
Parameter Limits	[0,]

Table 7: Initial guess and search limits for λ in the signal projection

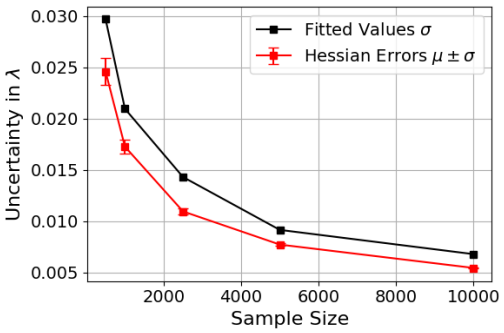
The final fitted parameters show strong agreement with their true values for the 10,000 samples, with the majority falling within 1 std deviation of the calculated Hessian errors(Table 8). One key advantage of this method is the removal of the background distribution's influence in the Y dimension, eliminating the need to fit two additional parameters. This significantly reduces computational overhead, as discussed in Section 8. The method's robustness and consistency are analysed through bootstrapping in Section 7.1.

Parameter	**Value ± Error(Hessian)**	**True Value**	**Std Errors Away**
μ	3.0006 ± 0.0083	3	0.07
σ	0.2931 ± 0.0080	0.3	0.86
β	0.8683 ± 0.0664	1	1.98
m	1.7874 ± 0.3012	1.4	1.29
f	0.5971 ± 0.0125	0.6	0.23
N	10000 ± 100	10000	0
λ	0.2991 ± 0.0055	0.3	0.16

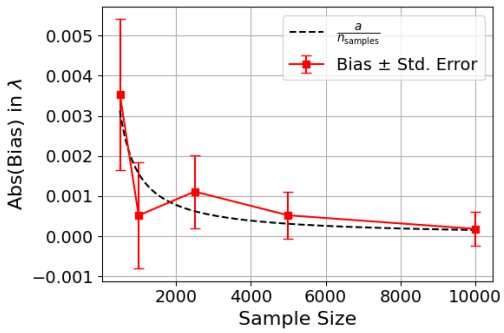
Table 8: Fitted parameters, errors, and deviations from true values using sWeights

7.1 Bootstraping the sWeights Method

The sWeights procedure was then analysed using parameteric bootstraping, using the same generated samples from Section 6, ie 250 toys with sample sizes of 500, 1000, 2500, 5000 and 10000. The results for the Crystal Ball distribution parameters align with those in Section 6, where the fit struggles due to m stacking at the boundary. This in turn effects the accuracy of the other parameters, which are highly correlated with m. Furthermore, the Hessian errors again fail to reliably represent the true uncertainty of the fit, resulting in over coverage and consistently low pull values. Notably, for a sample size of 500, an average of 13 samples failed to converge, consistent with the behavior observed in the original MLE fit.



(a) Hessian errors and standard deviation of fitted λ across sample sizes



(b) Biases in $\lambda \pm$ Std. Error

Figure 19: sWeights fitting method's bias and uncertainty for λ across sample sizes

The key difference is that the fitted estimates for λ now rely on the X dimension's as this determines the data's weights and thus their projection in Y. Before, in Section 6, the dimension's results were largely uncorrelated and thus independent. This presents challenges for uncertainty quantification as binned MLE fit provides the uncertainty based on the projected data, not the true distribution. Hence this uncertainty doesn't account for the compounding effects of X's fit.

As seen in 19a, this results in the hessian error, consistently underestimating the uncertainty of the fit, evident from the overly wide pull distributions, shown in Figure 20.

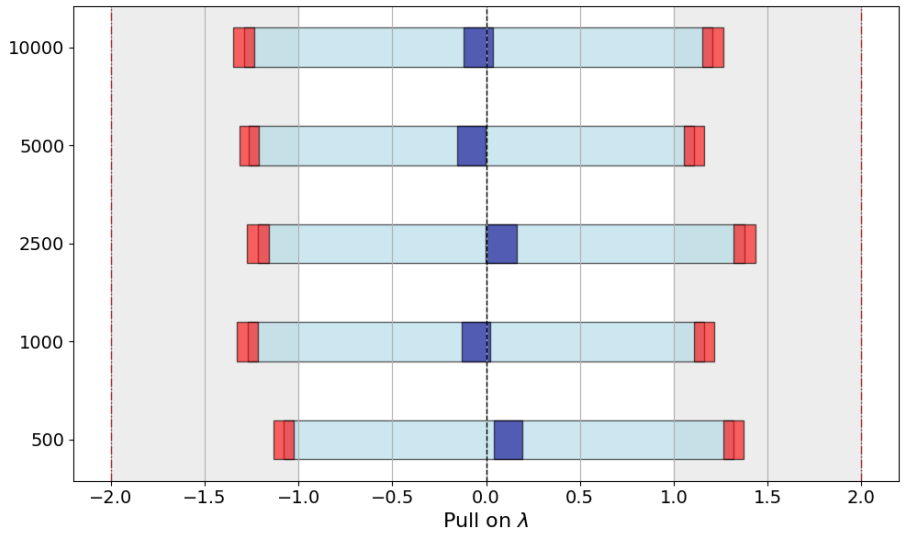


Figure 20: Pull distributions for λ using sWeights for all sample sizes, showing mean (blue) and standard deviation (red)

Finally, the biases are shown to follow a similar trend to Section 6's, with an $\mathcal{O}(n^{-1})$ trend, as the fitting is still being performed by an MLE estimator and hence the inherent bias is still present at finite sample sizes. Once again the bias is consistently an order of magnitude lower than the uncertainty of the fit, and hence accounted for.

8 Part g - Comparison of Methods

Finally, the report aims to compare the two fitting methods, the standard MLE fitting and the sWeights procedure, and highlight their respective advantages and disadvantages in experimental research.

8.1 Computational Performance

The computational performance of the two methods is compared initially through the runtime of fitting 100,000 samples. Secondly, the typical runtime to perform the fitting for all bootstrap samples generated in Section 6 are compared:

Method	**Typical Run Time (s)**	**Rumtime Reduction (%)**
Full MLE	6.6	N/A
Full MLE Bootstrap	496.3	N/A
sWeights MLE	4.3	34.8
sWeights MLE Bootstrap	274.8	44.6

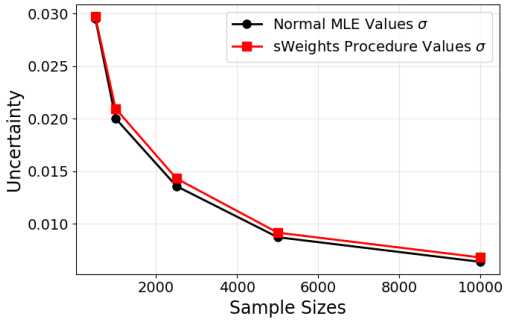
Table 9: Comparison of run times for sWeights and full MLE fitting

The sWeights procedure shows a clear advantage in computational performance, with a reduction in runtime of 34.8% and 44.6% for a single fit and full bootstrapping procedure respectively. A result of it fitting for 2 fewer parameters (those of the Y dimension's background distribution). As we enter the era of exascale experiments, such as the High-Luminosity LHC [1], and the Square Kilometre Array [3], computational performance is increasingly worth consideration.

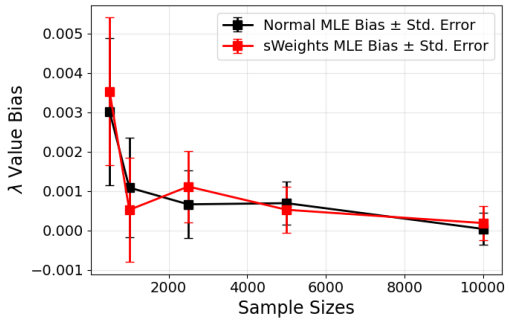
8.2 Uncertainty in Fit

Figure 21 shows very strong similarities between both methods' true uncertainty and bias. The plot shows that despite the compounding errors from the X dimension, the sWeights procedure can fit the signal distribution in Y with a similar level of accuracy. However, the drawback of the

sWeight procedure, as previously discussed, is that the Hessian errors are underestimated due to its inability to account for the propagating effects.



(a) Standard deviation of fitted λ across sample sizes



(b) Bias of fitted λ across sample sizes

Figure 21: Comparison of bias and uncertainty for λ between Normal MLE and sWeights

8.3 Removal of Background Contribution in Y

The main advantage of the sWeights procedure compared to using a full MLE estimate, 'is that one avoids the need to parameterise the background density in the control variable' [2]. This is particularly useful when the distribution of the background (or any component) in the control variable is poorly understood.

8.4 Conclusion

Overall, the sWeights procedure shows clear advantages in computational performance while remaining consistent with its uncertainty and bias. However, its main drawback is its inability to accurately quantify the uncertainty of the fit due to the compounding errors from the X dimension. Therefore, this method requires further refinement to propagate these errors. It is also worth noting that the sWeights procedure is particularly applicable to this distribution where the variables are independent. However, more complicated distributions may require the more generalised Custom Orthogonal Weight Functions (COWs) [2].

9 Appendicies

The repository's Bootstrap folder contains further plots for all parameters and analysis procedures.

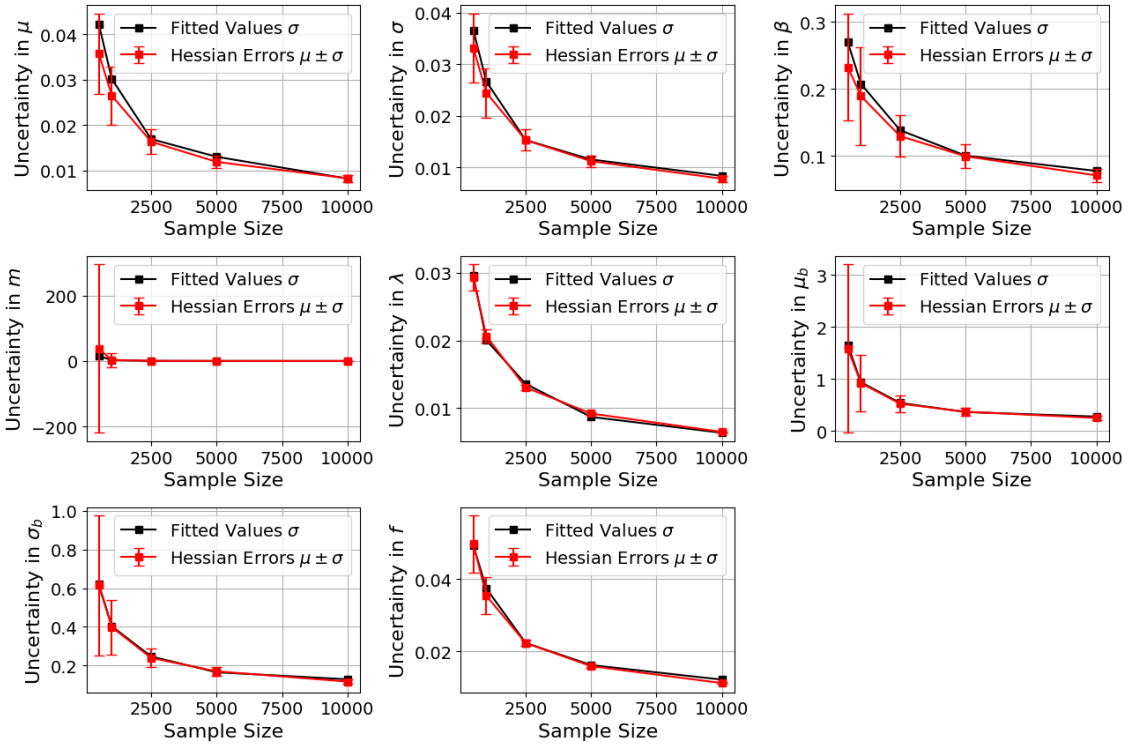


Figure 22: Hessian errors and standard deviation of fitted parameters from original MLE fitting procedure across sample sizes

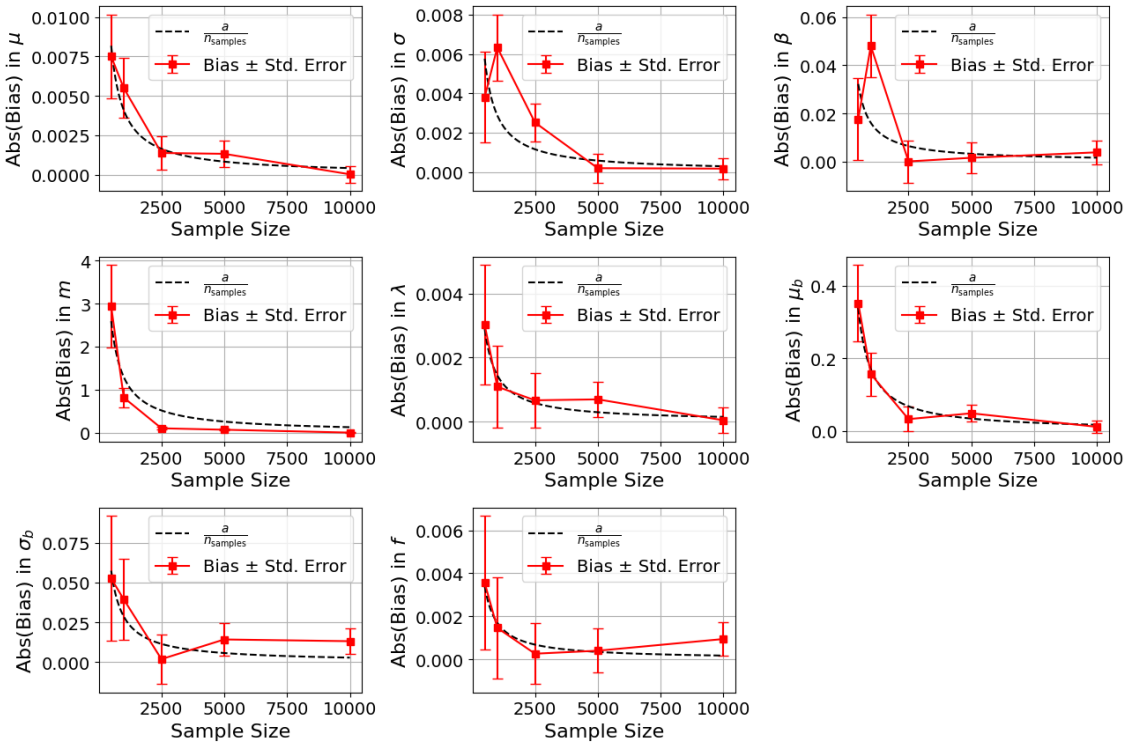


Figure 23: Bias of fitted parameters from original MLE fitting across sample sizes

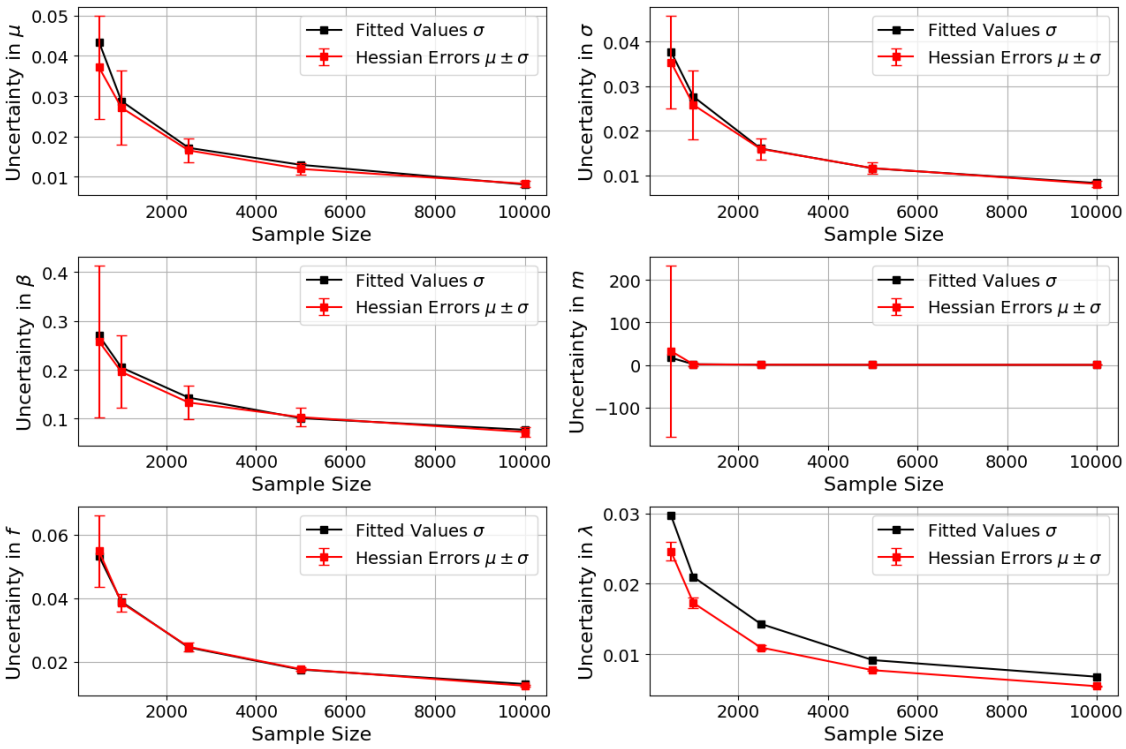


Figure 24: Hessian errors and standard deviation of fitted parameters from sweights procedure across sample sizes

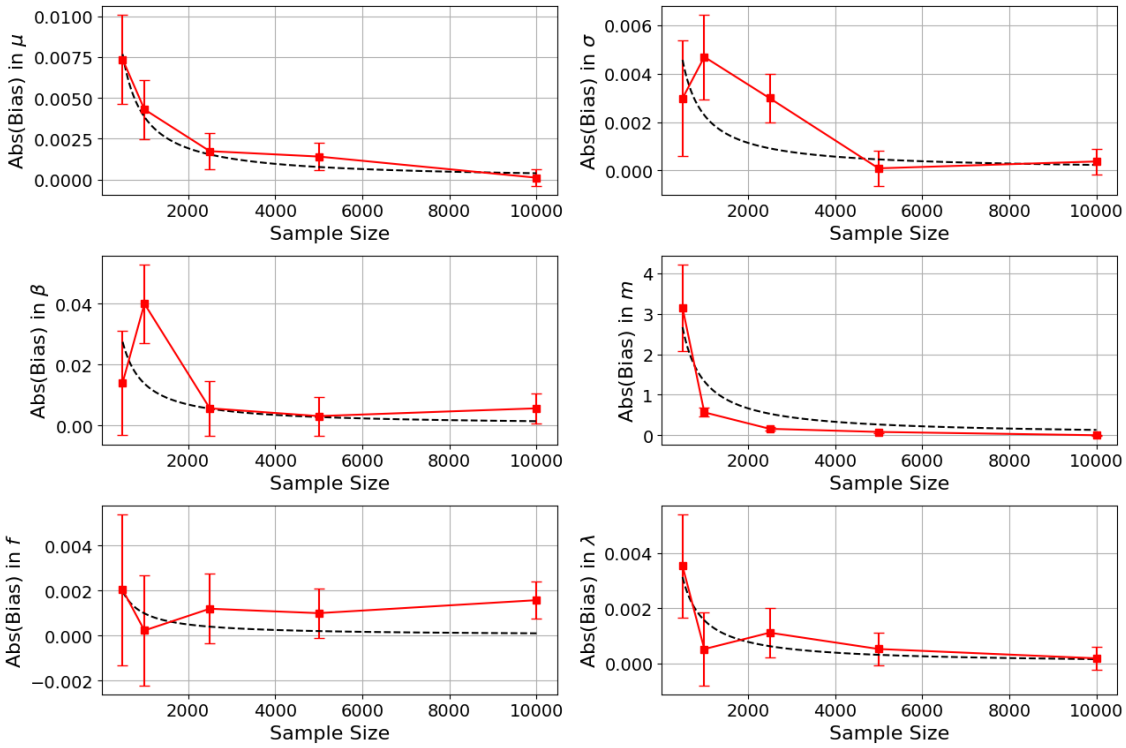


Figure 25: Bias of fitted parameters from sweights procedure across sample sizes

9.1 Declaration of Use of Autogeneration Tools

This report made use of Large Language Models (LLMs), primarily ChatGPT and Co-Pilot, to assist in the development of the statistical analysis pipeline. These tools have been employed for:

- Generating detailed docstrings for the repository's documentation.
- Formatting plots to enhance presentation quality.
- Performing iterative changes to already defined code.
- Debugging code and identifying issues in implementation.
- Helping with Latex formatting for the report.

References

[1] G Apollinari et al. *High Luminosity Large Hadron Collider HL-LHC*. en. 2015. DOI: 10.5170/CERN-2015-005.1. URL: https://cds.cern.ch/record/2120673.

[2] Hans Dembinski et al. "Custom Orthogonal Weight functions (COWs) for event classification". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1040 (Oct. 2022), p. 167270. ISSN: 0168-9002. DOI: 10.1016/j.nima.2022.167270. URL: http://dx.doi.org/10.1016/j.nima.2022.167270.

[3] A. M. M. Scaife. "Big telescope, big data: towards exascale with the Square Kilometre Array". In: *Philosophical Transactions of the Royal Society of London Series A* 378.2166, 20190060 (Mar. 2020), p. 20190060. DOI: 10.1098/rsta.2019.0060.

[4] Ali A. Al-Shomrani. "An improvement in maximum likelihood estimation of the Burr XII distribution parameters". In: *AIMS Mathematics* 7.9 (2022), pp. 17444–17460. DOI: 10.3934/math.2022961.