ROOT Project - Part 2: Unbinned vs. Binned Fits

Project Goal: The objective of this analysis is to compare the unbinned maximum-likelihood estimate for an exponential decay to several binned (histogram) fits, utilizing ROOT's binned-likelihood option. This approach aims to illustrate the impact of binning on the estimator and to demonstrate how sufficiently fine binning can recover the result obtained from the unbinned case.

Data: The following 40 decay times were analyzed:

$\{4.99, 4.87, 2.59, 3.04, 3.39, 6.20, 10.61, 7.64, 3.92, 5.33, 4.85, 2.39, 4.16, 6.74, 3.53, 5.86, 5.41, 26.25, 4.40, 10.79, 7.08, 2.86, 33.9$

The total number of points is $N = 40$, and the sum of decay times is $\sum_i t_i = 270.4$.

Unbinned Analysis (Analytical ML)

For the exponential probability density function $f(t|\lambda) = \lambda e^{-\lambda t}$, the log-likelihood is given by

$$\ln L(\lambda) = N \ln \lambda - \lambda \sum_i t_i,$$

with the maximum-likelihood estimator (MLE) expressed as $\hat{\lambda} = N / \sum_i t_i$.

Computed values (analytical):

- $\hat{\lambda}_{\mathrm{ML}} = \dfrac{40}{270.4} = 0.147929.$
- Analytical standard error (approximate): $\sigma_{\hat{\lambda}} \approx \dfrac{\hat{\lambda}}{\sqrt{N}} = 0.0233896.$

Log-Likelihood Plot and Graphical 1-Sigma

The function $\ln L(\lambda) = 40 \ln \lambda - 270.4\lambda$ was evaluated and plotted around its maximum, and the graphical 1σ interval (where $\Delta \ln L = -0.5$) was determined.

Values obtained from the log-likelihood analysis:

- $\ln L_{\max} = -116.441.$
- Target for 1σ: $\ln L_{\max} - 0.5 = -116.941.$
- Graphically determined bounds: $\lambda_- = 0.125756,\ \lambda_+ = 0.172567.$
- Asymmetric errors: -0.0221735 and $+0.0246383$. Average (approximate symmetric) error: 0.0234059.
- The graphical results are found to be consistent with the analytical error of 0.0233896.

Comment: The peak of the likelihood and the graphical 1σ band exhibit close agreement with the analytical estimate, supporting the robustness of the unbinned ML result.

Binned Fits (Histograms) and Diagnostics

Three binned-likelihood fits were performed using the following configurations (employing ROOT with option "L"):

1. **3 bins, range** $0 - 15$ (bin width $= 5$)
2. **5 bins, range** $0 - 15$
3. **350 bins, range** $0 - 35$ (bin width $= 0.1$; includes all data)

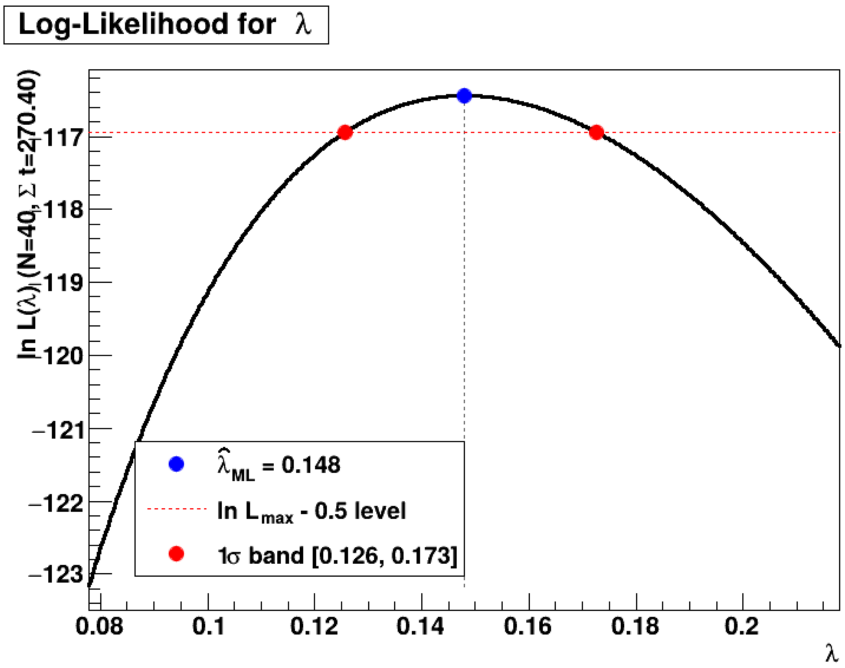

Figure 1: Profile log-likelihood $\ln L(\lambda)$ as a function of λ. The maximum and the horizontal line at $\ln L_{\max} - 0.5$ are indicated.

3-bin Fit (0–15, 3 bins)

Results and Diagnostics:

- Fitted $\lambda_{\text{binned},3} = 0.16122 \pm 0.0468606$.
- The fit reports (approximate) $\chi^2/\text{NDF} = 1.45395/1$; note that χ^2 interpretation is limited in cases of low counts and likelihood fits.
- Integral of fitted function over [0,15]: 190.047.
- Bin width = 5. Therefore $Q = \text{Integral}/\text{bin width} = 190.047/5 = 38.0094$.
- The sum of histogram bin contents in the fitted range is 37 (out of 40 total entries).

Observation: The normalization of the 3-bin fit is consistent with the observed counts ($Q \approx 38$ vs. 37 in range). The binned value for λ is somewhat higher than the unbinned estimate and exhibits a larger uncertainty, as is expected when coarse binning is used.

5-bin Fit (0–15, 5 bins)

Result:

- $\lambda_{\text{binned},5} = 0.14603 \pm 0.0435447$.

Observation: Increasing the number of bins from 3 to 5 leads the estimate to move closer to the unbinned ML value (0.147929), and the uncertainty decreases slightly. This is consistent with expectations: moderate bin refinement can recover more information from the data.

350-bin Fit (0–35, 350 bins)

Result:

- $\lambda_{\text{binned},350} = 0.142986 \pm 0.0248022$.

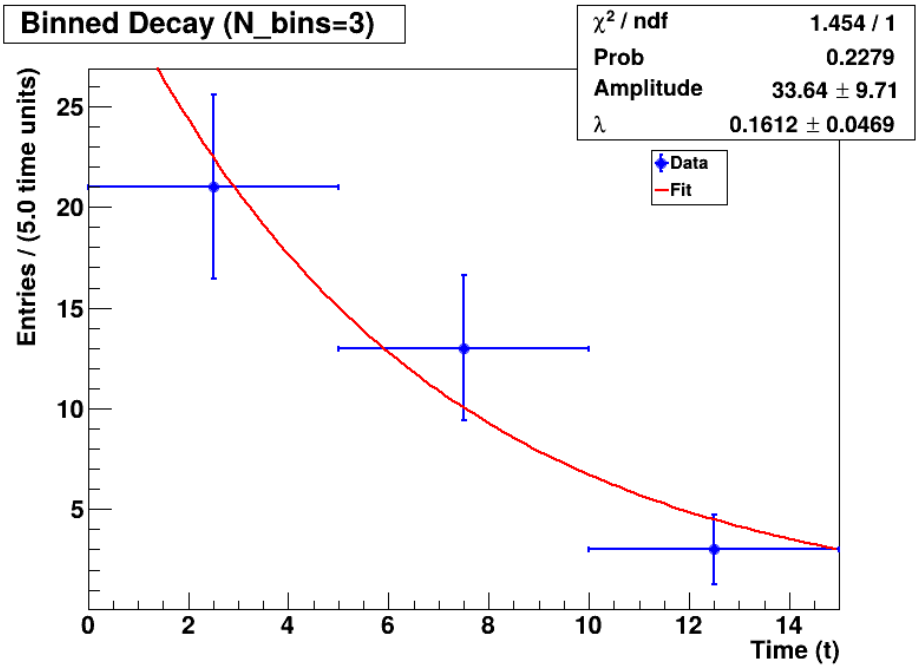

Figure 2: Binned ML fit to the histogram with 3 bins covering $t \in [0, 15]$.

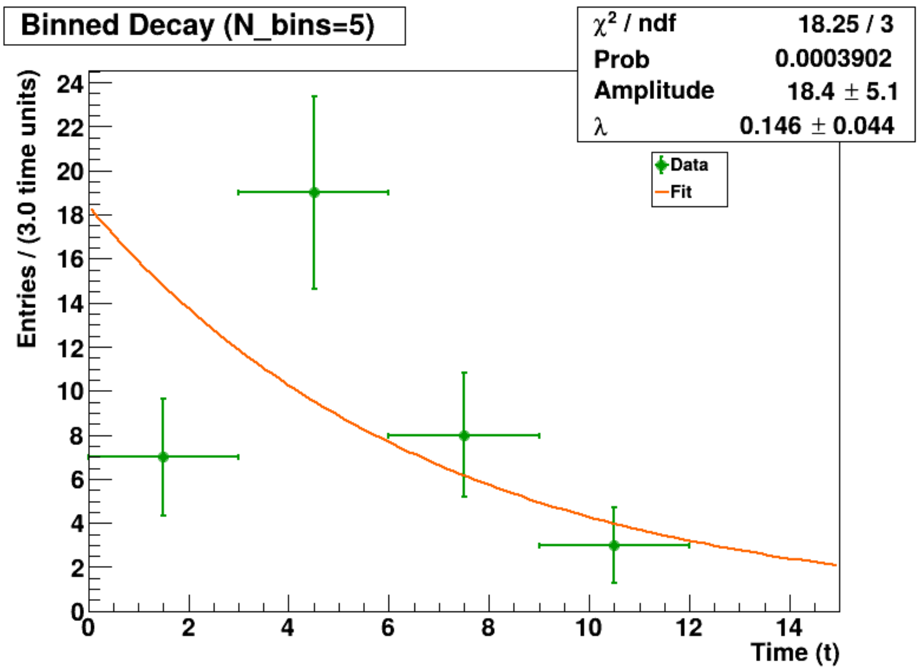

Figure 3: Binned ML fit to the histogram with 5 bins covering $t \in [0, 15]$.

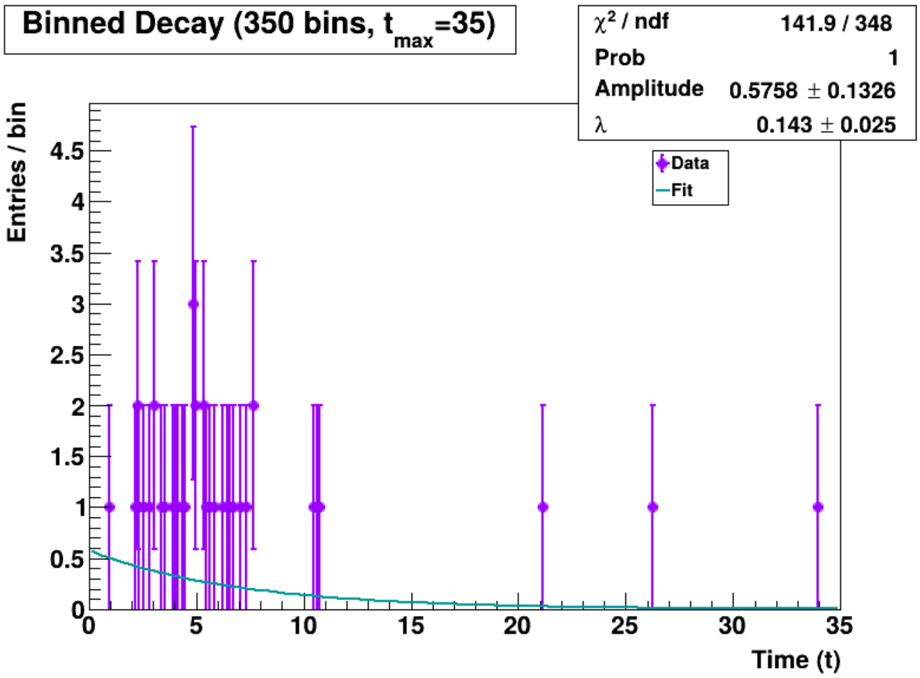

Figure 4: Binned ML fit to the histogram with 350 bins over $t \in [0, 35]$. Most bins are empty or contain a single event.

Observation: When very fine binning is employed and the full range is included, the binned ML result converges toward the unbinned result (0.147929), with an uncertainty comparable to the unbinned analytical error. This demonstrates the expected limiting behavior: as bin width decreases, the binned likelihood approaches the unbinned likelihood.

Conclusions

- The preferred estimate is provided by the unbinned MLE:

$$\boxed{\hat{\lambda} = 0.147929 \pm 0.02339.}$$

- The graphical log-likelihood interval (0.125756, 0.172567) yields an average error ≈ 0.02341, which is in very good agreement with the analytical calculation.
- Coarse binning tends to increase the uncertainty and can result in a shift of the central value, while finer binning (or use of the unbinned ML method) allows for more accurate recovery of the underlying information in the sample.
- The normalization check for the 3-bin fit ($Q = 38.0094$ vs. observed 37) indicates consistency between the fit and the observed data in the fitted range.

Appendix: Code Snippets (provided for reproducibility)

```
// Example: compute analytical unbinned ML
double sum_t = 0.0;
for (double t_val : times) sum_t += t_val;
lambda_ml_unbinned = static_cast<double>(N_data) / sum_t;
err_lambda_ml_unbinned = lambda_ml_unbinned / TMath::Sqrt(static_cast<double>(N_data));
```

```cpp
// Example: create log-likelihood TF1 and find 1-sigma band
auto logLikelihoodFunc = [&](double current_lambda) {
    if (current_lambda <= 0) return -1e10;
    return N_data * TMath::Log(current_lambda) - current_lambda * sum_t;
};
TF1* fLogL = new TF1("fLogL", [&](double*x, double*p){ return logLikelihoodFunc(x[0]); },
                     lambda_plot_min, lambda_plot_max, 0);
fLogL->SetNpx(500);
double logL_max = fLogL->GetMaximum();
double lambda_at_max = fLogL->GetMaximumX();
double target_logL_for_sigma = logL_max - 0.5;
double lambda_sigma_minus = fLogL->GetX(target_logL_for_sigma, lambda_plot_min, lambda_at_max)
    ;
double lambda_sigma_plus  = fLogL->GetX(target_logL_for_sigma, lambda_at_max, lambda_plot_max)
    ;
```