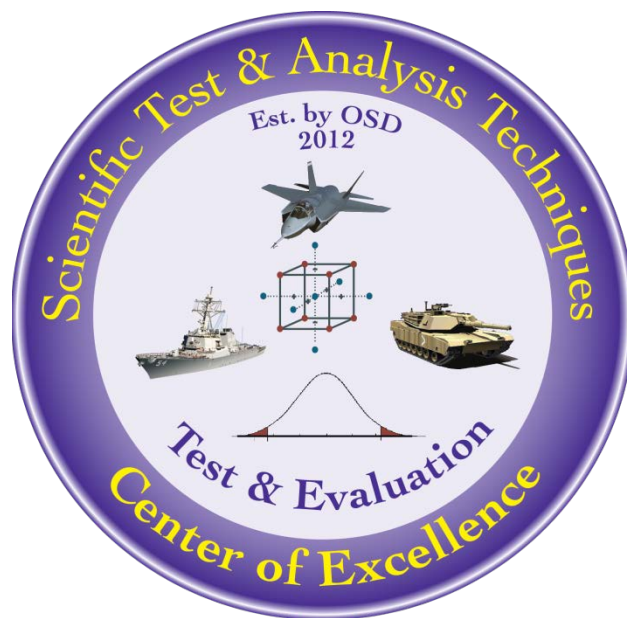


Practical Bayesian Analysis for Failure Time Data Best Practice

*Authored by: Michael Harman
15 June 2017*



The goal of the STAT COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. This and other COE products are available at www.afit.edu/STAT.

Table of Contents

Table of Contents	1
List of Figures	2
List of Tables	2
Executive Summary.....	3
Introduction	3
Analytical Objectives.....	4
Frequentist and Bayesian Differences	4
Steps to Implementing Bayesian Analysis.....	6
Assumptions in the Closed Form Solution	8
Choosing a Prior	9
Matlab Code Details.....	14
R Code Details	14
Conclusions	14
References	16
Appendix A: Example Data Set and R Code CSV File Example	17
Appendix B: Matlab Closed Form Solution Code	18
Appendix C: R MCMC Code.....	22
Appendix D: Zero Failure Data Set Comparison.....	27

List of Figures

Figure 1: Probability Distribution Function.....	5
Figure 2: Sample Matlab Code Output	8
Figure 3: Narrow Prior Impact on the Posterior	10
Figure 4: Vague Prior Impact on the Posterior	11
Figure 5: Posterior from 10 Data Points	12
Figure 6: Posterior from 50 Data Points	13
Figure 7: Closed Form Solution with Zero Failures	27
Figure 8: MCMC Solution with Zero Failures	28
Figure 9: MCMC Solution with 5 Censored Data Points	29

List of Tables

Table 1: Comparison of Results for Various Priors	11
Table 2: Comparison of Results for Various Data Sizes	13
Table 3: Comparison of Uncensored and Censored Data Analysis	29

Executive Summary

Reliability assessment is a typical requirement for defense test programs. A common method is the determination of a mean time between failures (MTBF) and informing decision makers on system suitability by comparing the MTBF lower confidence bound to the MTBF threshold. The accuracy of this method is challenged when data sets are small, have limited test times for the observed failures, and/or contain no failures at all. Using a classical (also called frequentist) approach, confidence intervals are wide when the data sets are small, and when there are zero failures, no estimate of MTBF is possible (dividing by zero). Bayesian analysis can address these issues and provides a more detailed assessment and more intuitive interpretation of the results. But while Bayes' rule is easily described, analysis for real world problems gets complicated quickly and typically requires advanced skills and software to conduct the analysis. This paper provides practical and easy-to-use Matlab code that will support most program reliability assessment needs. Additionally, R code is provided for more flexible applications.

Keywords: Bayes, reliability, prior selection, mean time between failures, conjugate prior, defense, Matlab

Introduction

Generating a reliability assessment is a typical requirement for defense test programs. Following a test period, this requirement is typically assessed using a frequentist approach where the mean time between failures (MTBF) is estimated as the total test time divided by the number of failures. Note, the following methods also apply if the desired metric is mean miles between failure or another similarly continuous measure. A decision on the suitability of the system is determined by comparing the MTBF lower confidence bound to the MTBF threshold. This method runs into difficulty when the data sets are small, have limited test time, and/or contain no failures. Confidence intervals are wide when the data sets are small and when there are zero failures, the MTBF is not estimable (dividing by zero) (Truett, 2017). Even if you assume the system has a high MTBF due to no observed failures, an exponential distribution lower bound calculation may be so low as to provide minimal information to the decision-maker (Morris, 2017).

Bayesian analysis can address these issues and provide a more detailed assessment and more intuitive interpretation of the results (Berger 2006). But while Bayes' rule is easily described, analysis for real world problems gets complicated quickly and typically requires advanced skills and software to conduct the analysis. This paper addresses these topics and provides practical, easy-to-use Matlab code (Appendix B) that will support most program reliability assessment needs. Additionally, R code is provided to perform similar Bayesian analysis (Appendix C). R is free, open-source software and extremely effective at addressing statistical problems, but we know government users may not have administrative privileges to load it onto government computers.

This paper does not cover the determination of the test time for a reliability test plan. Frequentist methods for calculating a test time length can be found in Kensler (2014) as well as numerous other sources. Bayesian test time determination is somewhat more complicated but an overview can be found at the NIST website (section 8.3.1.5) among other sources. Regardless, this paper assumes the data already exists.

Analytical Objectives

Regardless of the selected method, analysts typically have the following objectives:

1. Collect failure time data for a determined test time
2. Estimate the value for MTBF
3. Determine a lower bound on this estimate for evaluation against requirements

This paper addresses these objectives using a Bayesian method and compares the process and output to frequentist methods. This method can be applied during developmental testing (DT) to compare performance before and after correction of deficiencies or in operational testing (OT) to evaluate overall performance.

Frequentist and Bayesian Differences

Frequentist methods treat model parameters as unknown, fixed constants and employ only observed data to estimate the values of parameters. In the case of failure time data, one might assume the data is exponentially distributed:

$$f(x; \lambda) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \text{ for } x \geq 0$$

where λ is the parameter of interest, MTBF. The maximum likelihood estimator for λ in this equation is simply the total test time divided by the number of failures which provides a point estimate for MTBF. To describe the variability in the estimate a confidence interval (more typically a one-sided lower bound) can be calculated (Morris, 2017). When the data set is small or no failures are observed the bounds can be wide and minimally informative. And, in the case of no failures, no point estimate is available (dividing by zero). Moreover, to the confusion of many, confidence bounds do not describe the range of values the parameter occupies, but describes the uncertainty associated with a sampling method (Kensler, Cortes, 2014). We would say that “we expect 90% of the estimated intervals to include the population parameter” (StatTrek.com, 2017).

Bayesian methods treat parameters as unknown random variables whose distribution (the prior) represents the current belief about the parameter. Bayes’ rule is defined as:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

where

1. $P(\theta|D)$ is the Posterior: Probability distribution of the parameter (θ) *given* a data set (D)
2. $P(D|\theta)$ is the Likelihood: A function of the observed data (D) *given* a parameter (θ)
3. $P(\theta)$ is the Prior: Probability distribution of the parameter (θ) which represents our belief on the parameter before the dataset D is observed
4. $P(D)$ is the evidence: Probability distribution of the observed data (D) which acts as a normalizing constant that ensures the Cumulative Posterior Distribution sums to 1.

Many people are familiar with maximum likelihood estimation (MLE). MLE is used when you want to find the parameter values that best fit the dataset using a specified distribution. The likelihood term represents this type of information. The difference is that the likelihood and prior are inputs to Bayesian analysis, not the output. The critical point in Bayesian analysis is that the posterior is a probability distribution function (pdf) of the parameter given the data set, not simply a point estimate. This enables all the properties of a pdf to be employed in the analysis. Figure 1 shows a pdf for a normal distribution with $\mu=80$ and $\sigma=5$.

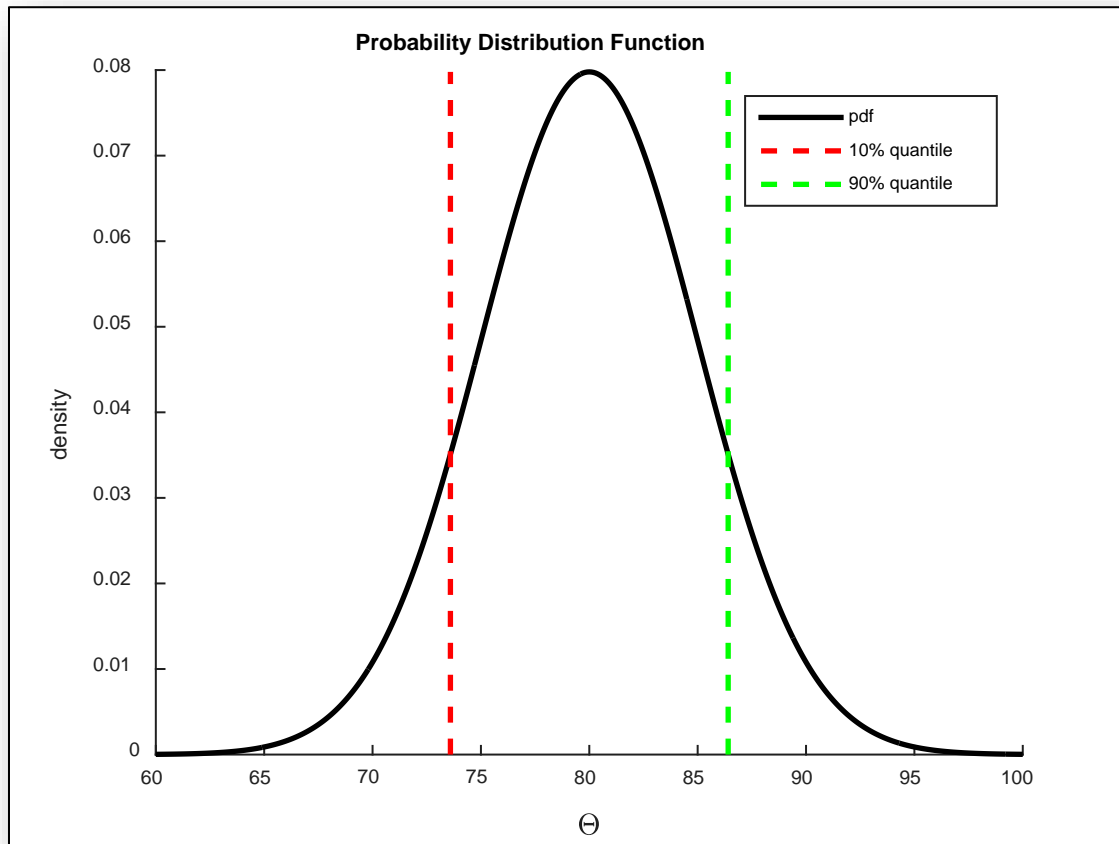


Figure 1: Probability Distribution Function

The pdf shows the probability of θ across a range of values and shows the most likely value at the peak (80). Also, we can state the lower quantile indicates a 10% chance the parameter is below 73.6 and 10% chance it is above 86.4. Since the Bayes posterior is a pdf, we can use credible interval (CI) quantiles instead of confidence intervals (unfortunately both are abbreviated CI). Note the more intuitive interpretation of credible intervals than confidence intervals.

In either case, the differences between frequentist and Bayesian methods become negligible as the sample size increases. However, when the data sets are small, these differences can be significant, with Bayesian interval estimates often narrower than the frequentist methods (Hamada 2008).

Steps to Implementing Bayesian Analysis

1. Choose a prior distribution that describes our belief of the MTBF parameter
2. Collect failure time data and determine the likelihood distribution function
3. Use Bayes' rule to obtain the posterior distribution
4. Use the posterior distribution to evaluate the data

Choose a Prior Distribution That Describes Our Belief of the MTBF Parameter

Any distribution can be chosen as a prior so long as it accurately describes the parameter information known and is determined **before** collecting any new data. In the case of failure times, we choose an inverse gamma distribution as it relates to the specific example below. The inverse gamma distribution is defined as:

$$f(\lambda; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{-\alpha-1} e^{-\frac{\beta}{\lambda}} \text{ for } \alpha > 0, \beta > 0$$

Where

- λ = exponential distribution parameter
- α = shape parameter
- β = rate parameter

There are several ways to arrive at the values of these parameters. First, MLE can be used to determine them directly. Alternatively, if expert opinion or engineering knowledge provide insight to the values of the mean and standard deviation of MTBF, the expected mean (μ) and variance (σ^2) can be used to determine the values of α and β :

$$\mu = \frac{\beta}{\alpha - 1} \text{ for } \alpha > 1, \sigma^2 = \frac{\beta^2}{(\alpha - 1)^2(\alpha - 2)} \text{ for } \alpha > 2$$

This is an option coded into the Matlab code in the Appendix. Finally, there may be expectations that 95% of the MTBF values fall in a certain range and this quantile information can be used to derive the parameters. This method is not covered in this paper (see Cook, 2010), but it is contained in the Matlab code.

Collect Data and Determine the Likelihood Distribution Function

Many defense systems' failure times are assumed to be exponentially distributed (NIST 8.1.6.1); however, this may not always be true (Reliasoft.com, 2001). Ultimately, the collected data should drive the choice of a likelihood distribution. Here we will choose the exponential distribution for the analytical convenience that will be described later.

Use Bayes' Rule to Obtain the Posterior Distribution

The posterior combines the prior and likelihood distributions and (generally) any combination is possible. The general case requires the use of numerical methods, usually Markov Chain Monte Carlo (MCMC). However, certain combinations of distributions result in closed form posteriors with the same form as the

prior. These are called conjugate priors and the case of an exponential likelihood and inverse gamma prior results in an inverse gamma posterior of the form (Fink, 1997 or [Wikipedia conjugate priors](#), 2017):

$$P(\lambda | D) \sim \text{Inverse Gamma} (\alpha_0 + N, \beta_0 + \sum_{i=1}^N x_i)$$

where the parameters are a combination of

- α_0 = prior α
- β_0 = prior β
- N = number of observed failures
- $\sum_{i=1}^N x_i$ = sum of all failure times.

Because the posterior is always a distribution, we can readily evaluate most likely and interval values for any data set, including small samples and those with no observed failures. Also, Bayesian analysis enables you to conduct posterior analysis following every failure data point while testing continues. Frequentist analysis requires the pre-determined test time to be completed before any analysis is conducted.

Example: Use the Posterior Distribution to Evaluate the Data

Consider an example with the following information:

- Inverse gamma prior information
 - $\alpha_0 = 46.9$
 - $\beta_0 = 3147.6$
- Data consisting of ten failure times (x_i) (these are ordered for clarity)
 - 55.8 57.8 70.1 79.1 94.1 96.6 103.9 111.3 122.1 161.0
 - $\sum_{i=1}^{10} x_i = 951.8$
 - $N = 10$

The prior takes the form

$$P(\lambda) \sim \text{Inverse Gamma} (46.9, 3147.6)$$

The posterior takes the form

$$P(\lambda | D) \sim \text{Inverse Gamma} (46.9 + 10, 3147.6 + 951.8)$$

The Matlab code provides the following information in the chart header as in Figure 2: Sample Matlab Code Output

- Max/most likely posterior value of parameter (peak)
- Specified quantile/credible interval (CI) value
- Frequentist MTBF point estimate (total test time/number of failures)
- Total test time
- Number of observed failures (N)
- Prior mean and sigma value

- Inverse gamma prior alpha and beta parameters
- Inverse gamma posterior alpha and beta parameters

Additionally, the chart plots threshold (T) and objective (O) values for evaluation against the lower CI.

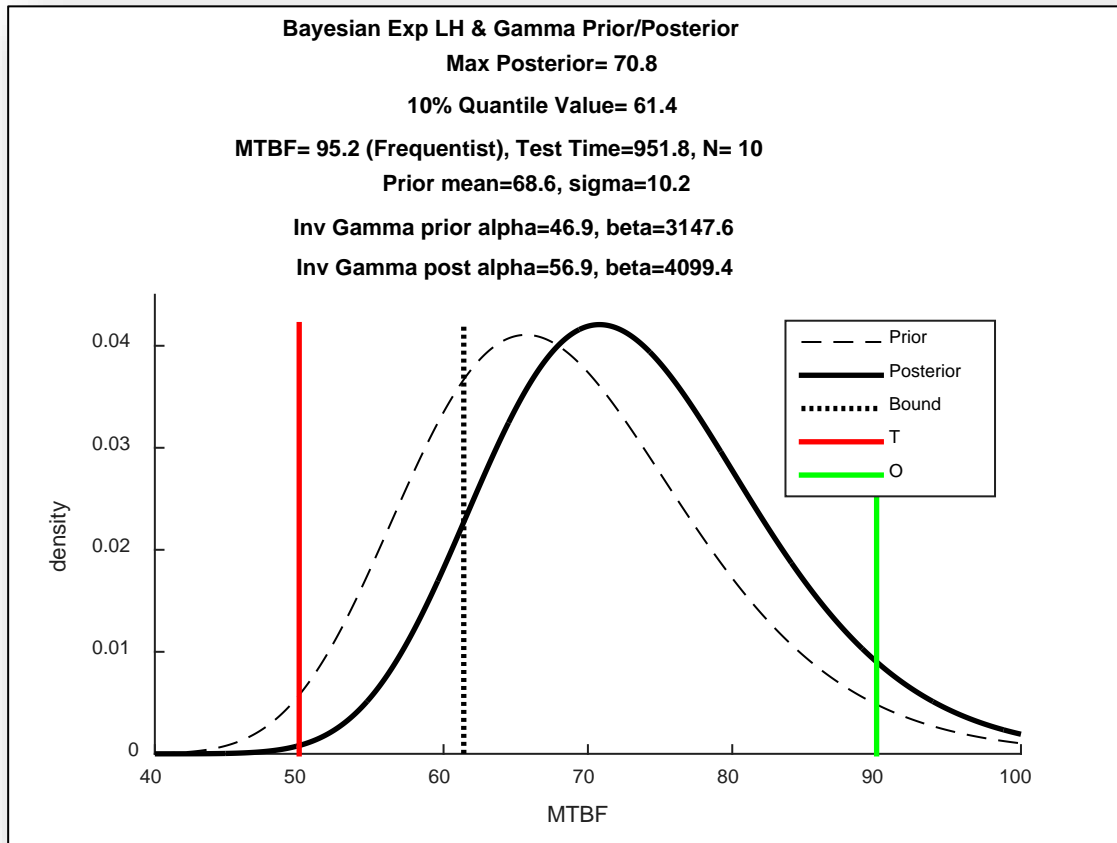


Figure 2: Sample Matlab Code Output

The prior depicts an inverse gamma distribution with 95% of its density between 50 (Threshold) and 90 (Objective). The observed 10 data points result in an inverse gamma posterior shifted slightly to the right. The 10% quantile of the posterior of MTBF (90% lower CI indicated by the dotted black line in Figure 1) is at 54.0 and is above the threshold. This indicates that there is a 90% chance the MTBF is above 54.0. Note this interpretation of the CI is quite intuitive compared to a confidence interval. The most likely MTBF value is at the peak, about 71.

Assumptions in the Closed Form Solution

Most statistical analysis requires the assumption of an underlying distribution and Bayesian analysis is not different. In this case, the data is assumed to be exponentially distributed (likelihood) and the prior on the parameter assumes an inverse gamma distribution. The failure time data should be exponentially

distributed (or at least close) for the results to be accurate. Data varying significantly from this assumption may skew results with no insight as to the extent of the impact.

The parameters of the posterior use failure meta-data (number of failures and sum of the failure times). This implies the test period ends at the last observed failure time, called a failure truncated period. However, most DOD test periods go until a pre-determined test time and the final failure time is censored, called a time truncated period. This closed form solution implemented in Matlab does not address censoring, but the MCMC code implemented in R provided in Appendix C does.

The closed form solution will also not correctly address a test period without any observed failures. You might assume the total test time (for sum of the failure times) and $N=0$ could be inserted into the posterior parameters equations, but this is not correct because the single data point is censored. Again, the MCMC code can handle any number of censored data points and a comparison of the Matlab and R code for this case is provided in Appendix D.

These assumptions and specific distributions may seem to overly constrain the analysis, but they facilitate the creation of closed form solutions using conjugate priors coded into Matlab and made accessible to more T&E practitioners.

Choosing a Prior

Choice of a prior may be a challenging task. A prior should be chosen *before* any data is collected so as not to appear to manipulate the output. However, there are several defensible, objective sources and methods for choosing a prior.

Hamada et al. (2008) outlines various sources for informative priors:

1. Physical/chemical theory
2. Computational analysis
3. Previous engineering and qualification test results from a process development program
4. Industrywide generic reliability data
5. Past experience with similar devices
6. Expert opinion

In DOD testing, one could argue that the most defensible source is data collected on the same systems in a previous test phase. This is primarily due to the complexity and uniqueness of DOD systems and a general lack of trust in data otherwise sourced.

Hamada also points out some concerns to be avoided:

1. Beware of zero values
2. Cognitive biases in the way people think
3. Overly narrow priors
4. Prior information should be relevant to the problem at hand
5. Be careful when assessing prior distributions on parameters that are not directly observable
6. Beware of conservatism. Realism is the desired ideal, not conservatism.

Bayesian analysis is essentially a weighted solution where the prior effect varies with the size of the data set. Posteriors from a small data set will typically not move significantly from the prior. Conversely, increasingly larger data sets will begin to overwhelm the prior and reduce its effect. If you chose Bayesian analysis because of the benefits it provides to small data sets, you may unintentionally impact the results with an unrealistic prior. A prior with a small variance (optimistic) implies a high degree of confidence the parameter only exists over a small range of values and is fair only if the data supports it. Conversely, a large variance (vague) used in an attempt to “be fair” imparts little knowledge to the posterior and may result in unnecessarily wide and uninformative intervals. So, an overly optimistic or needlessly vague prior does not serve the analysis well. Ultimately, the analyst must be able to defend the choice of a prior and allow the data to tell the story. It is acceptable to conduct sensitivity analysis for different priors before data is collected (as in the next section) but this should be avoided after data has been collected.

Impacts of Various Priors

Figure 3 and Figure 4 show the posteriors for the same fictitious 10 point data (true MTBF=80) set using a narrow prior and a vague prior. Note the peak of the vague prior in Figure 4 appears to be shifted left compared to the narrow prior. This is simply a function of the gamma distribution properties. The mean and sigma values are shown in the header for comparison.

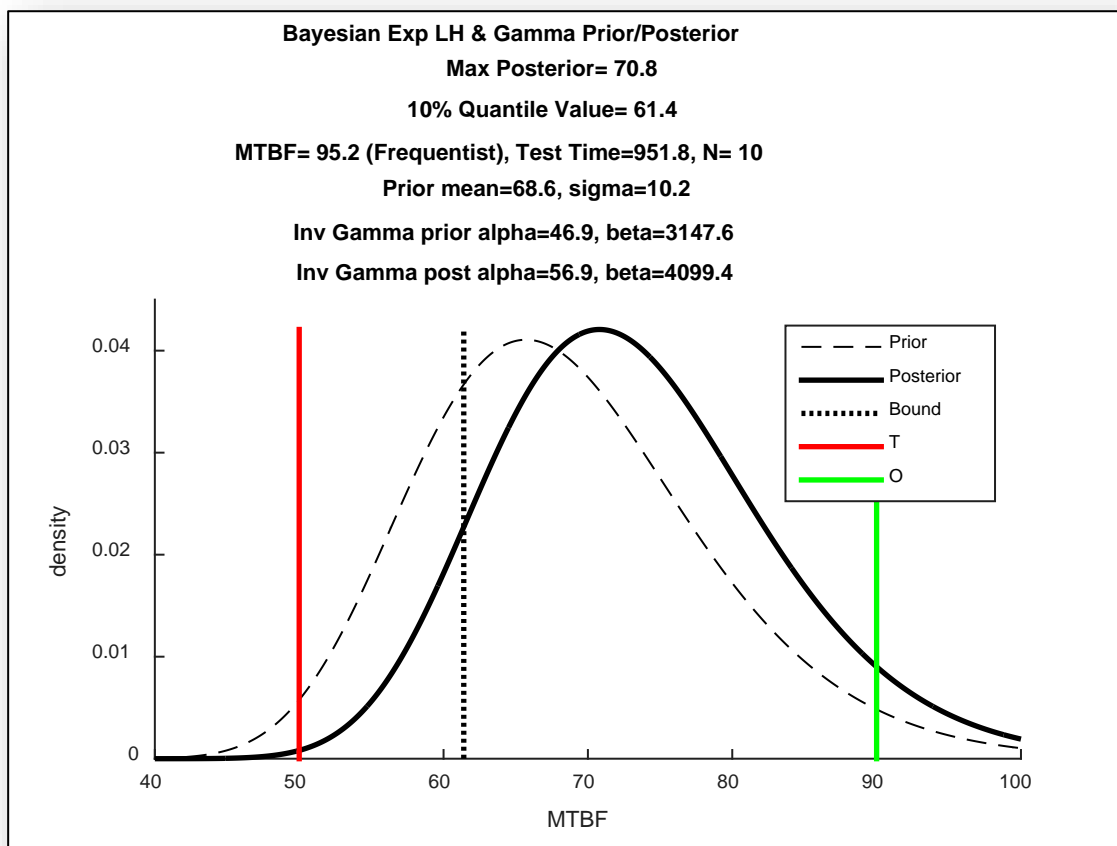


Figure 3: Narrow Prior Impact on the Posterior

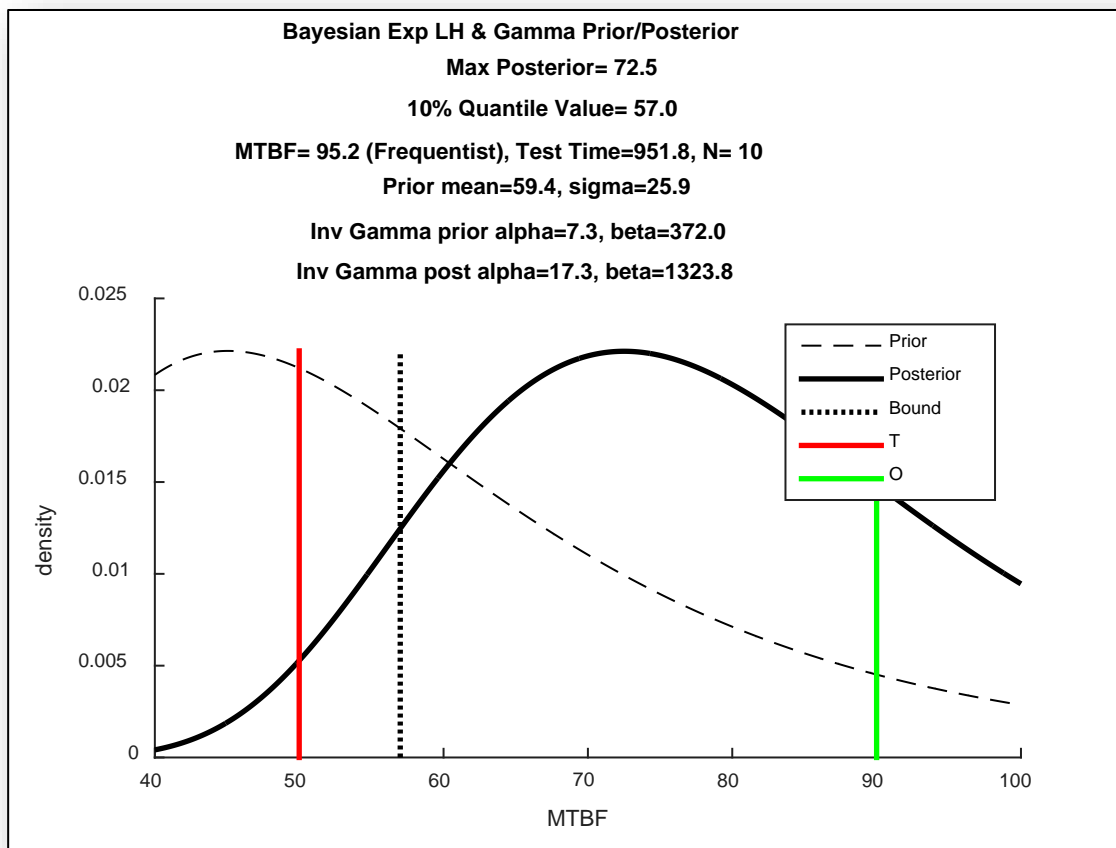


Figure 4: Vague Prior Impact on the Posterior

Takeaways from this example (not necessarily indicative of all Bayesian analysis):

1. The Bayesian posterior peak value is closer to the simulated MTBF (80) than the Frequentist result (these differences would be unknown with real data)
2. The posterior peak values vary slightly (~ 2.5%)
3. The narrow prior results in a narrower and more peaked posterior
4. The prior impacts the location of the CI quantile, the vague prior moving it left
5. In both cases the lower bound is greater than threshold resulting in the same conclusion (passing)

Table 1: Comparison of Results for Various Priors

Mean/Most Likely	Narrow Prior	Vague Prior
Frequentist (% error)	95.2 (19%)	95.2 (19%)
Bayesian (% error)	70.8 (12%)	72.5 (9%)
10% Quantile Value		
Frequentist*	65.1	65.1
Bayesian	61.4	57.0

*Frequentist CI values were calculated using JMP Life Distribution platform (likelihood method)

Impact of Sample Size

Figure 5 and Figure 6 compare small (10 data points) and large (50 data points: same 10 plus 40 more) data sets analyzed with the same prior. The data is included alone in Appendix A and in the Matlab code in Appendix B.

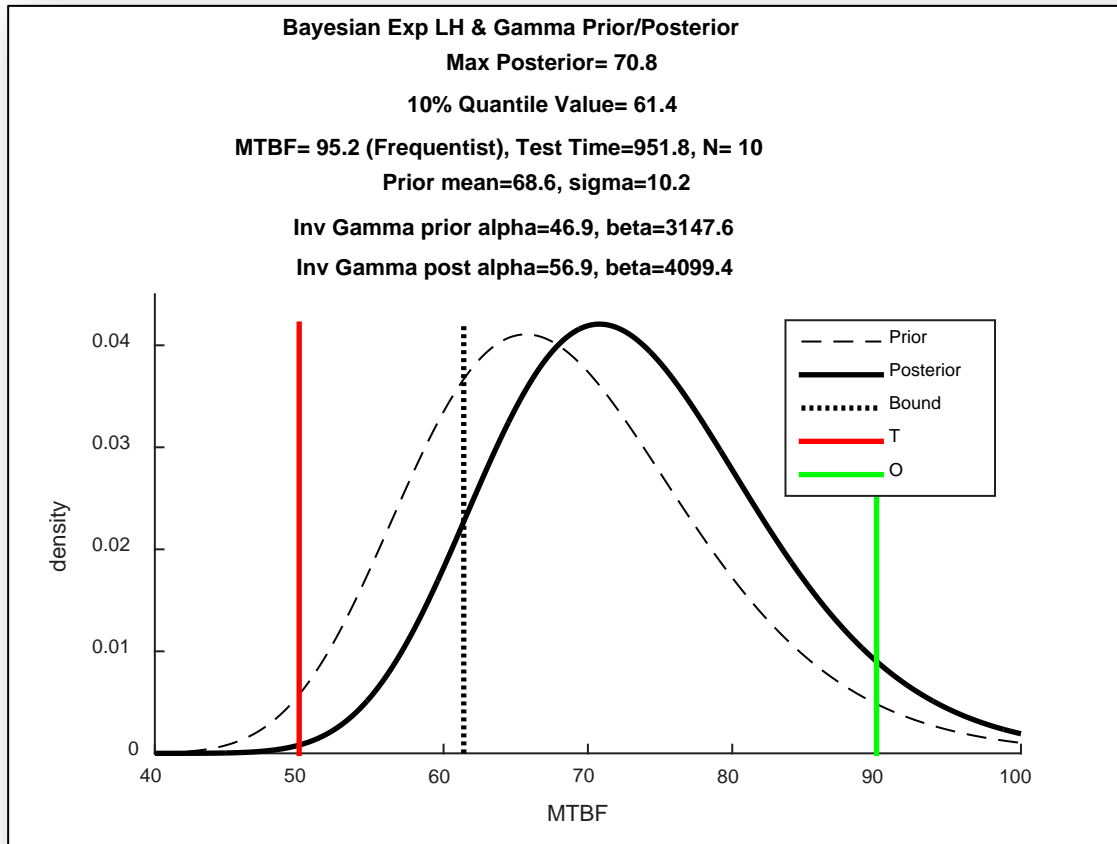


Figure 5: Posterior from 10 Data Points

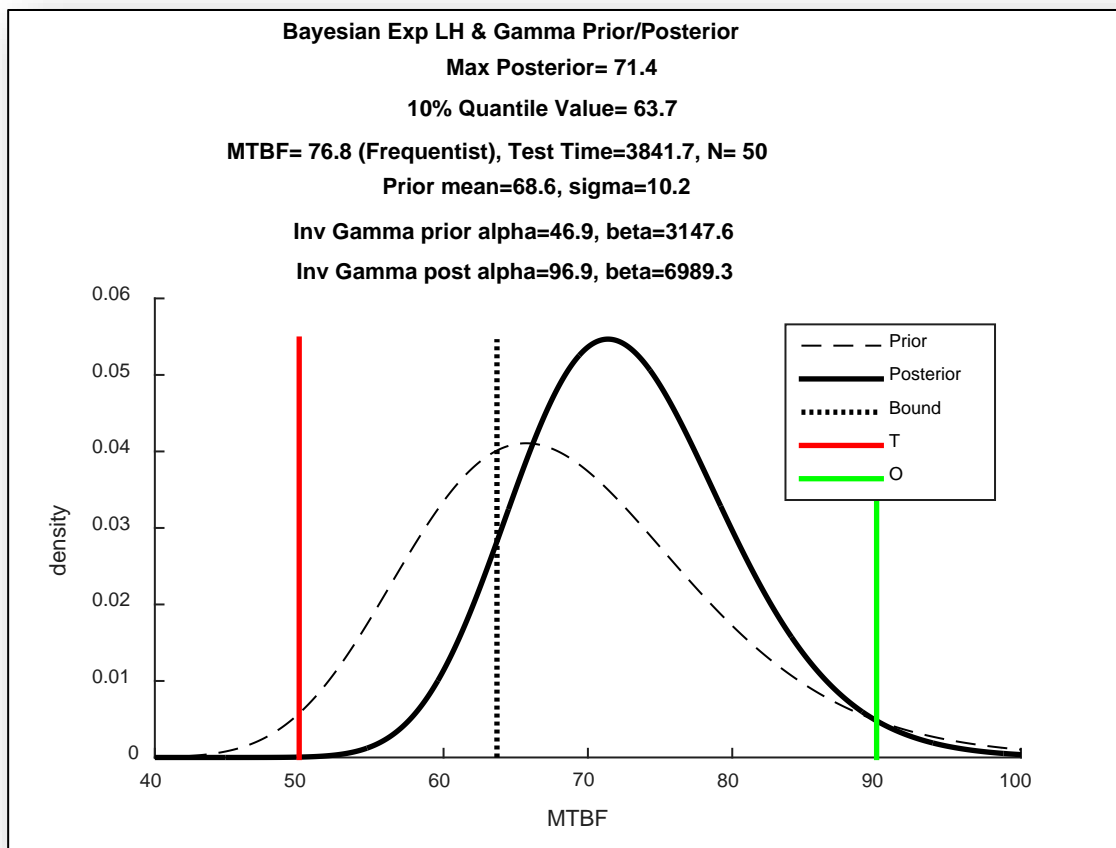


Figure 6: Posterior from 50 Data Points

Takeaways from this example (not necessarily indicative of all Bayesian analysis):

1. Larger difference between MTBF point estimates (~19%) than between peak posterior values (~1%). The exponential distribution mean is sensitive to large value data points, especially in small data sets.
2. The larger data set has a narrower and more peaked posterior
3. The Bayesian CI changes less as N increases (9.4 difference @ 10 points, 7.7 (50 points)) than the Frequentist CI (30.1 (10 points), 12.4 (50 points))
4. In all cases the lower bound is greater than threshold resulting in the same conclusion (passing)

Table 2: Comparison of Results for Various Data Sizes

Mean/Most Likely	10 Data Points	50 Data Points
Frequentist (% error)	95.2 (19%)	76.8 (4%)
Bayesian (% error)	70.8 (12%)	71.4 (11%)
10% Quantile Value		
Frequentist*	65.1	64.4
Bayesian	61.4	63.7

*Frequentist CI values were calculated using JMP Life Distribution platform (likelihood method)

Matlab Code Details

The full code is provided in Appendix B. Matlab was chosen because it is prevalent at government test and evaluation sites. Detailed instructions are commented in the code but additional information is provided below. You may request the code in native format at COE@afit.edu.

1. Enter data and make changes only in the USER DATA ENTRY section.
2. Test times can be in any order. They do not need to be sorted.
3. Threshold and Objective values are input for plotting against the distributions and desired credible interval.
4. Three methods are available to define the prior. Be sure to input the selected method value as well as entering the specific method data.
 - a. Method 1 inputs a mean and standard deviation to determine inverse gamma parameters.
 - b. Method 2 uses lower and upper quantiles to determine inverse gamma parameters.
 - c. Method 3 inputs inverse gamma parameters directly.

R Code Details

The code is provided in Appendix C and the example data file example is in Appendix A. Detailed instructions are commented in the code but additional information is provided below. Note the JAGS package and RJAGS library must be installed to run the code. You may request the code in native format at COE@afit.edu.

1. Working directory needs to match the location of the R file and data file.
2. This code is set up to read a CSV file with two columns: DATA and CENSOR.
 - a. In this code, 1=censored data, 0=observed failure.
 - b. The data file must be in the same working directory.
 - c. Enter the file name on line 34: **Data.Set <- read.csv("FailureData.csv",header=TRUE).**
3. Threshold and Objective values are input for plotting against the distributions and desired credible interval.
4. Three methods are available to define the prior. Be sure to input the selected method value as well as entering the specific method data.
 - a. Method 1 (line 48) inputs a mean and standard deviation to determine gamma parameters. Note R MCMC functionality does not require inverse gamma parameterization.
 - b. Method 2 (line 52) uses lower and upper quantiles to determine gamma parameters.
 - c. Method 3 (line 62) inputs gamma parameters directly.

Conclusions

Bayesian analysis provides a clear and intuitive method to address reliability failure time analysis, especially when frequentist methods fall short. The mathematical fundamentals permit easy calculation and the provided code should enable all DOD practitioners to effectively analyze real world real world problems and provide insightful information to decision makers. If your data violates the assumptions of

the solution presented in this paper you should contact the STAT COE (COE@afit.edu) of your local analyst for specific help.

References

Berger, James. The case for objective Bayesian analysis. *Bayesian Anal.* 1 (2006), no. 3, 385--402. doi:10.1214/06-BA115. <http://projecteuclid.org/euclid.ba/1340371035>.

Cook, John D., Determining distribution parameters from quantiles, 27 January 2010, https://www.johndcook.com/quantiles_parameters.pdf

Fink, Daniel, A Compendium of Conjugate Priors, web, 30 May 2017. <https://www.johndcook.com/CompendiumOfConjugatePriors.pdf>

Hamada, Michael, Alyson Wilson, C. Shane Reese, and Harry Martz. *Bayesian Reliability*. New York: Springer, 2008. Print.

Kensler, Jennifer, Luis Cortes, A Practical Guide for Interpreting Confidence Intervals-Best Practice, www.afit.edu/STAT, 24 December 2014

Kensler, Jennifer, Reliability Test Planning for Mean Time Between Failures-Best Practice, www.afit.edu/STAT, March 21, 2014

Morris, Seymour F., Reliability Analytics Corporation. "Confidence Limits - Exponential Distribution." *Reliability Analytics Corporation*. Web, 30 May 2017.

NIST/SEMATECH e-Handbook of Statistical Methods, 30 May 2017, <http://www.itl.nist.gov/div898/handbook/>

Reliasoft.com, <http://www.reliasoft.com/newsletter/4q2001/exponential.htm>, web, 5 June 2017.

Truett, Leonard, A Quick Guide To Understanding The Impact Of Test Time On Estimation Of MTBF-Best Practice, www.afit.edu/STAT, June 2017

StatTrek.com, web, 5 June 2017, <http://stattrek.com/estimation/confidence-interval.aspx>

Wikipedia, "Conjugate Prior." *Wikipedia*. Wikimedia Foundation, 04 May 2017. Web. 30 May 2017.

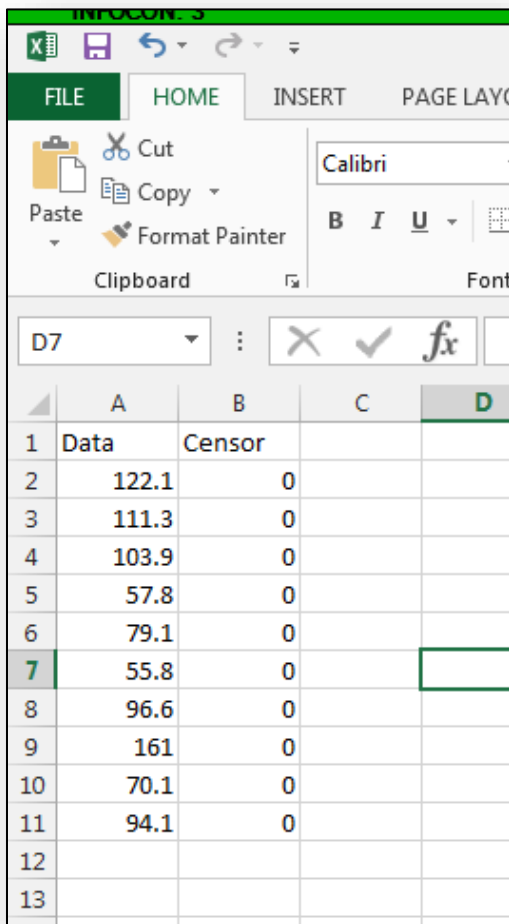
Wikipedia, "Inverse-gamma Distribution." *Wikipedia*. Wikimedia Foundation, 28 May 2017. Web. 30 May 2017.

Appendix A: Example Data Set and R Code CSV File Example

Failures times were simulated using an exponential distribution with MTBF=80.

Failure times:

122.1	111.3	103.9	57.8	79.1	55.8	96.6	161.0	70.1	94.1	67.3	340.3	111.6
41.5	33.3	22.7	95.1	98.4	91.1	59.8	23.4	120.3	0.7	22.3	54.5	8.0
35.8	50.5	40.2	220.6	99.5	79.2	125.0	41.2	1.6	69.7	19.3	34.9	36.7
22.3	39.7	267.6	43.4	151.6	20.8	12.5	57.5	51.0	114.7	64.3		



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Data	Censor		
2	122.1	0		
3	111.3	0		
4	103.9	0		
5	57.8	0		
6	79.1	0		
7	55.8	0		
8	96.6	0		
9	161	0		
10	70.1	0		
11	94.1	0		
12				
13				

Appendix B: Matlab Closed Form Solution Code

```
% MTBF_Bayes_Exponential_Gamma_Posterior_Analysis.m
%
% This code performs Bayesian analysis on a continuous failure time data
% set using an exponential likelihood and inverse gamma conjugate prior.
% The user enters the data set information and the method they want to use
% to select the prior. The program outputs a plot of the prior and
% posterior along with the MTBF point estimate, and selected quantile
% (lower or upper).
%
% Your failure time data should be (at least roughly) exponentially
% distributed. If this is not the case then the results may not be
% accurate. This assumption is built into the conjugate prior closed form
% solution which precludes the use of MCMC simulation. This inherently
% assumes the test time ends at the last failure (no last censored data
% point). In the case you have a test period with no failures, or you have
% at least one censored data point, use the similarly titled R code
% provided by the STAT COE
%
% The closed form solution explanation can be found at
% https://www.johndcook.com/CompendiumOfConjugatePriors.pdf, page 15.
%
% Notes:
% *****
% Method 2 (quantile selection) may not solve when the X1 and X2 values are
% too close. Try moving them apart by small values until it works.
% *****
%
% Created at the Scientific Test and Analysis Techniques (STAT) Center of
% Excellence (COE). Contact email: COE@AFIT.edu Website: www.AFIT.edu/STAT
% DISCLAIMER: This code has been checked using known examples and functions
% correctly to the best of our knowledge.
%
% Michael Harman
% Vers 2.2 5/31/17
%
% Clear and initialize
close all;
clear all;
clc;

% *****
% USER DATA ENTRY *****

% IF TEST TIME HAS AT LEAST 1 FAILURE: ENTER DATA HERE
% Data set is time between failures
Data=[122.1 111.3 103.9 57.8 79.1 55.8 96.6 161.0 70.1 94.1 67.3...
      340.3 111.6 41.5 33.3 22.7 95.1 98.4 91.1 59.8 23.4 120.3 0.7...
      22.3 54.5 8.0 35.8 50.5 40.2 220.6 99.5 79.2 125.0 41.2 1.6...
      69.7 19.3 34.9 36.7 22.3 39.7 267.6 43.4 151.6 20.8 12.5 57.5 51.0
      114.7 64.3];

% Enter Threshold and Objective Requirement Values
```

```

T=50;
O=90;

% Enter credibility interval value (output is one sided)
CI=0.90;
% Enter 1 for lower bound, 0 for upper
bound=1;

% PRIOR SELECTION: Choose method 1, 2, 3 to define the prior
% Enter specific data under the chosen METHOD below
method=2;

% METHOD 1: Define typical (mean) MTBF prior Mean and Standard Deviation
mean=68.6;
sigma=10.2;

% METHOD 2: Enter expected prior quantile information for MTBF
% Lower Quantile (enter as %, e.g. 2.5% = 2.5)
P1=2.5;
% Lower Quantile Value
X1=50;
% Upper Quantile (enter as %, e.g. 2.5% = 2.5)
P2=97.5;
% Upper Quantile Value
X2=90;

% METHOD 3: Enter prior INVERSE GAMMA distribution parameters
% These parameters are typically determined using MLE on a data set
alpha=46.9;
beta=3147.6;

% END USER DATA ENTRY *****
% *****

N=length(Data);
SumXi=sum(Data);
MTBF=SumXi/N;

% Check method and determine which values to use
if method==1 % Mean and Sigma for GAMMA distribution
    % no change to mean and sigma;
    % determine INVERSE GAMMA alpha and beta
    alpha=mean^2/sigma^2+2;
    beta=mean*(alpha-1);

elseif method==2 % Quantile bounds
    % method from https://www.johndcook.com/quantiles\_parameters.pdf
    % solves for GAMMA alpha, beta, mean, sigma and converts to INVERSE
    alpha=0.1;
    incr=.1;
    complete=0;
    Xratio=X2/X1;

```

```

while complete==0
    Calcratio=icdf('Gamma',P2/100,alpha,1)/icdf('Gamma',P1/100,alpha,1);
    if abs(Xratio-Calcratio)<=0.001
        complete=1;
        beta=X1/icdf('Gamma',P1/100,alpha,1);
    elseif Xratio-Calcratio<0
        alpha=alpha+incr;
    elseif Xratio-Calcratio>0
        alpha=alpha-incr;
        incr=incr/2;
    end
end

% determine mean and sigma
mean=alpha*beta;
sigma=sqrt(alpha*beta^2);

% convert alpha and beta to inverse gamma values
alpha=mean^2/sigma^2+2;
beta=mean*(alpha-1);

elseif method==3 % alpha and beta for inverse gamma distribution
    % determine mean and sigma
    mean=beta/(alpha-1);
    sigma=sqrt(beta^2/(alpha-1)^2/(alpha-2));

end

% X values for calcs and plotting
X=.8*T:0.1:O+.2*T;

% Prior is INVERSE GAMMA
Prior=(beta^alpha)/gamma(alpha)*X.^(-alpha-1).*exp(-beta./X);

% Posterior is INVERSE GAMMA
alphapost=alpha+N;
betapost=beta+SumXi;
PostLOG= alphapost*log(betapost) - gammaln(alphapost) + log(X)*(-alphapost-1)
-betapost./X;
Posterior=exp(PostLOG);
MaxPindex = find(Posterior==max(Posterior(:)));

% Calculate desired bound value
z=abs(bound-CI);
complete=0;
i=1;
while complete==0
    density=sum(Posterior(1:i))/sum(Posterior);
    if density<z
        i=i+1;
    else

```

```

        complete=1;
    end
end
BVal=X(i);

% Plotting
plotnum=1;

% Max Y value for plotting
maxY=max([Prior Posterior]);

figure(1)
hold on
plot(X,Prior,'k--')
plot(X,Posterior,'k','linewidth',2)
plot([BVal, BVal],[0, maxY],'k:','LineWidth',2); % Bound value
plot([T, T],[0, maxY],'r','LineWidth',2); % T line
plot([0, 0],[0, maxY],'g','LineWidth',2); % T line
legend('Prior','Posterior','Bound','T','O')
MP=num2str(X(MaxPindex),'%.1f');
MM=num2str(MTBF,'%.1f');
sx=num2str(SumXi,'%.1f');
BV=num2str(BVal,'%.1f');
aa=num2str(alpha,'%.1f');
bb=num2str(beta,'%.1f');
mm=num2str(mean,'%.1f');
ss=num2str(sigma,'%.1f');
ap=num2str(alphapost,'%.1f');
bp=num2str(betapost,'%.1f');
title(['Bayesian Exp LH & Gamma Prior/Posterior'];
    ['Max Posterior= ',MP];
    [num2str(z*100),'% Quantile Value= ',BV];
    ['MTBF= ',MM,' (Frequentist), Test Time=',sx,', N= ',num2str(N)];
    ['Prior mean=',mm,', sigma=',ss];
    ['Inv Gamma prior alpha=',aa,', beta=',bb];
    ['Inv Gamma post alpha=',ap,', beta=',bp]]);
xlabel('MTBF')
ylabel('density')
hold off

% output selections to screen for QA
fprintf('Data set contains %d fails.\n',N)
fprintf('Chosen Prior selection method = %d.\n',method)

% EOF

```

Appendix C: R MCMC Code

```
# MTBF_Bayes_Exp_Gamma_Post_Analysis_Censor.R
#
# This code performs Bayesian analysis on a continuous failure time data
# set using an exponential likelihood and gamma prior.
# The program outputs a plot of the prior and posterior along with the MTBF point estimate
# and selected posterior quantile.
#
# Your failure time data should be (at least roughly) exponentially
# distributed. If this is not the case then the results may not be
# accurate. This assumption is built into the model.

# The code facilitates the use of censored data points.
# Examples: single or multiple systems with test time ending before next observed failure or
# single system with no observed failures.

# Created at the Scientific Test and Analysis Techniques (STAT) Center of Excellence (COE).
# Contact email: COE@AFIT.edu
# Website: www.AFIT.edu/STAT
# DISCLAIMER: This code has been checked using known examples and functions correctly
# to the best of our knowledge.
# Michael Harman
# Vers 3.1 6/20/17

# access RJAGS library
library(rjags)

# USER DATA ENTRY REGION *****

# Change working directory to where your files are located
setwd("I:/HARMAN WORKING/Bayes/1-R")

# Read data set
# Reads a CSV Data set with first column DATA and second column CENSOR
# Censor column =1 if censored, =0 if observed failure
Data.Set <- read.csv("FailureData.csv",header=TRUE)

# Requirements (for plotting)
T<-c(50)
O<-c(90)

# Desired Credibility Interval/Quantile (0.10=10%ile/90% lower CI, 0.90=90%ile/90% upper CI)
CI<-0.1

# PRIOR SELECTION: Choose method 1, 2, 3 to define the prior
```

```
# Enter specific data under the chosen METHOD below
method<-2
```

```
# METHOD 1: Define MTBF prior Mean and Standard Deviation
mean<-80
sigma<-10
```

```
# METHOD 2: Enter expected prior quantile information for MTBF
# Lower Quantile (enter as %, e.g. 2.5% = 2.5)
P1<-2.5
# Lower Quantile Value
X1<-50
# Upper Quantile (enter as %, e.g. 2.5% = 2.5)
P2<-97.5
# Upper Quantile Value
X2<-90
```

```
# METHOD 3: Define prior gamma parameters
gamma.shape<-44.9
gamma.scale<-1.53
```

```
#####
##### END USER DATA ENTRY REGION#####
```

```
# Check method and determine which values to use
# Mean and Sigma for GAMMA distribution
# no change to mean and sigma;
if (method==1) {
  gamma.shape<-mean^2/sigma^2;
  gamma.scale<-mean/gamma.shape
} else if (method==2) {
  # Quantile bounds
  # method from https://www.johndcook.com/quantiles\_parameters.pdf
  # solves for GAMMA alpha, beta, mean, sigma
  alpha<-0.1;
  incr<-0.1;
  complete<-0;
  Xratio<-X2/X1;

  while (complete==0)
  {
    Calcratio<-qgamma(P2/100,alpha,1)/qgamma(P1/100,alpha,1)
    if (abs(Xratio-Calcratio)<=0.001)
    {
      complete<-1;
      beta<-X1/qgamma(P1/100,alpha,1)
```



```

    }
    else if (Xratio-Calcratio<0)
    {
      alpha<-alpha+incr
    }
    else if (Xratio-Calcratio>0)
    {
      alpha=alpha-incr;
      incr=incr/2
    }
  }

  # determine mean and sigma
  gamma.shape<-alpha
  gamma.scale<-beta
  mean<-gamma.shape*gamma.scale
  sigma<-sqrt(gamma.shape*gamma.scale^2)

} else
# Method=3
mean<-gamma.shape*gamma.scale
sigma<-sqrt(gamma.shape*gamma.scale^2)

# End method check and calculations

# Define MODEL
model <-paste(
'model{
# Likelihood (exponential)
for(i in 1:n)
{
  is.censored[i] ~ dinterval(LH[i], Censor.Time[i])
  LH[i] ~ dexp(tau)
}

# Prior (gamma)
tau <- 1/lambda          # rate parameter
lambda ~ dgamma(shape,1/scale)  # jags uses rate=1/scale

# ENTER PRIOR PARAMETERS HERE
shape <- ', gamma.shape,
'scale <- ',gamma.scale,'
}')
# End of MODEL

# Assign csv file information

```

```

Data<-Data.Set$Data
Censor<-Data.Set$Censor
is.na(Data)<-Censor==1    # make data NA where censored
Censor.Time<-Data.Set$Data+1-Censor

#Store data needed for JAGS in list
jags.dat<-list(n=length(Data),LH=Data,is.censored=Censor,Censor.Time=Censor.Time)

#Sets initial values for MCMC
init.Censor<-Data.Set$Data+5
is.na(init.Censor)<-Censor==0 # make data NA where not censored
init.values <- list(LH=init.Censor)

# Run JAGS
jags <- jags.model(textConnection(model), inits = init.values, n.chains = 1, data = jags.dat)

#Update starts sampler at a value n.iter into the chain (burn-in)
update(jags,n.iter = 5000)

#Obtain draws from the MCMC algorithm
posterior <- coda.samples(jags,thin = 5, variable.names = c('lambda'), n.iter = 50000)

# Summary of the posterior parameter (lambda in this case)
summary(posterior)
CIquantile=quantile(posterior[[1]],probs=CI)

# Find most likely posterior value
MaxP <- density(posterior[[1]])
MaxY<-MaxP$y[which.max(MaxP$y)]
MaxX<-MaxP$x[which.max(MaxP$y)]

# ACF displays autocorrelation plots for diagnostics
acf(posterior[[1]])

# Determine plot limits
Xplotmin<-0.8*T
Xplotmax<-O+0.4*T
Yplotmin<-0
Yplotmax<-MaxY*1.1

# Calc number of actual observed fails for MTBF use
obs.fails<-length(Censor)-sum(Censor)
if (sum(Censor)==length(Censor)) {MTBF<-NaN; NN=0} else {MTBF<-sum(Data.Set$Data)/obs.fails;
NN=obs.fails}

# Plot prior and posterior

```

```

par(mar=c(5, 4, 7, 4))
plot(density(posterior[[1]]),xlim=c(Xplotmin,Xplotmax),ylim=c(Yplotmin,Yplotmax),
     main=paste("Bayesian Exp LH & Gamma Prior/Posterior \n",
               "Max Posterior = ", round(MaxX,1)," \n",
               CI*100, "% Quantile = ", round(CIquantile,1),
               "\n MTBF = ", round(MTBF,1), " (Frequentist), Test Time = ", round(sum(Data.Set$Data),1)," , N
(observed)= ",NN, " \n",
               "Gamma prior mean = ",round(mean,1)," , Gamma prior sigma = ", round(sigma,1), " \n",
               "Gamma prior shape = ",round(gamma.shape,1)," , Gamma prior scale = ",
round(gamma.scale,1)),
     xlab="MTBF",lwd=2, cex.main = 1.0)
lines(density(rgamma(10000,gamma.shape,scale=gamma.scale)),lty=2,lwd=2)

abline(v=c(T),col="red",lwd=3)
abline(v=c(O),col="green",lwd=3)
abline(v=c(CIquantile),col="black",lty=3,lwd=2)

legend("topright",lty=c(2,1,3),c("Prior","Posterior","CI"), cex = .75)

# EOF

```

Appendix D: Zero Failure Data Set Comparison

The following two figures show posteriors for the case where the test time (for a single unit) expired before any failures were observed. Figure 7 is the closed form solution (Matlab code) and Figure 8 is the MCMC solution (R code). The R code correctly accounts for the likelihood differences for censored data points.

Significant observations:

1. Posterior peaks are similar (85.6, 85.2)
2. CI values are similar (72.2, 72.8)
3. Closed form solution appears to have more mass in the right tail
4. This data set consists of a single censored data point. A larger data set with more censored points may result in much different posteriors between the two solution methods.

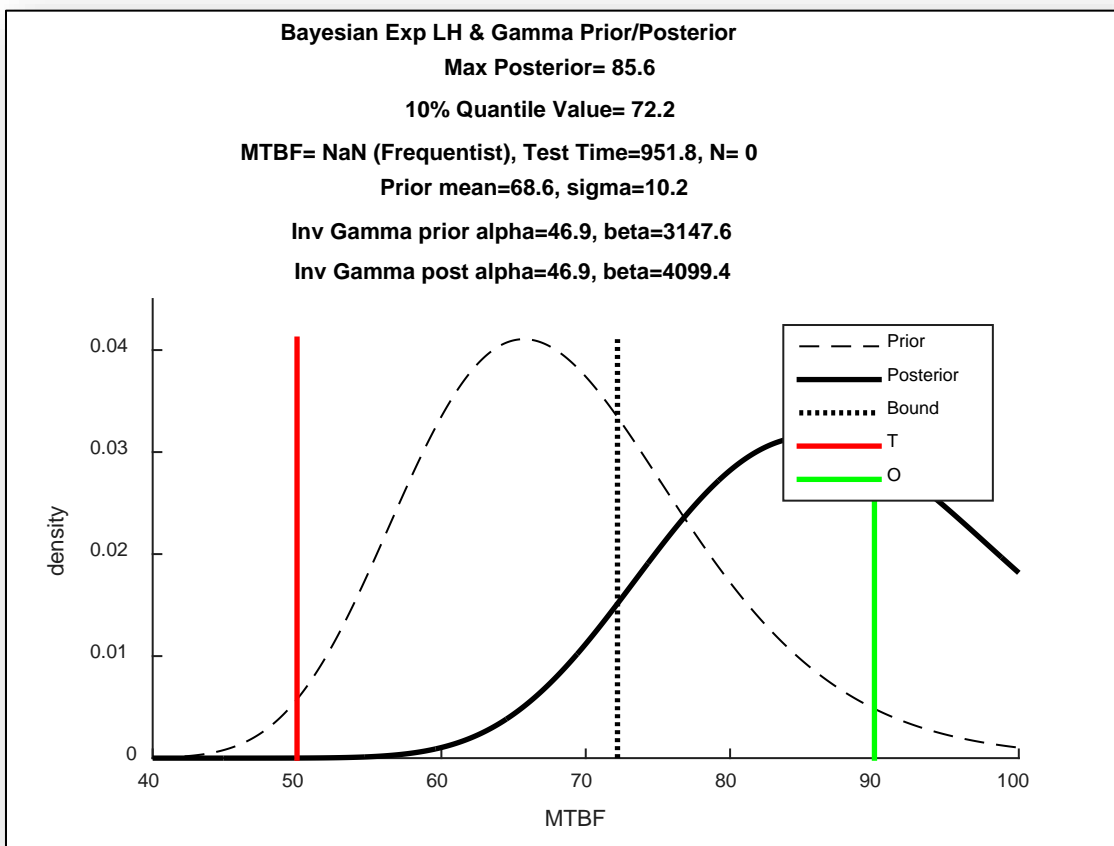


Figure 7: Closed Form Solution with Zero Failures

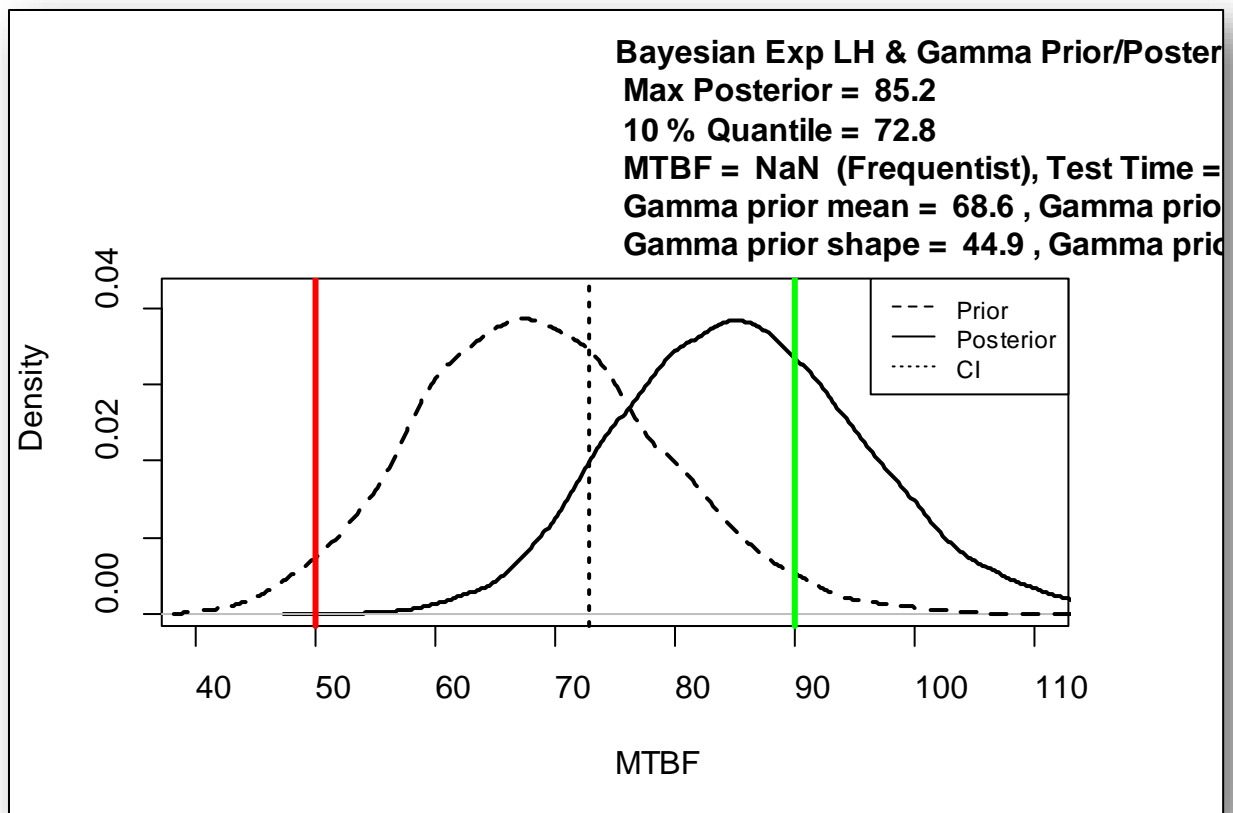


Figure 8: MCMC Solution with Zero Failures

Despite the similarities in this output, the correct analysis method must be used when censored data is present. The previous plots appear similar because there is only 1 point in the data set so the prior is not overwhelmed by the data. However, consider the case where the same ten data points seen in Figure 2 are accrued by five units resulting in 5 censored data points (the plot reflects the 10 point data set with every even data point censored). The closed form Matlab solution has no way to deal with this correctly but the MCMC R code does (Figure 9).

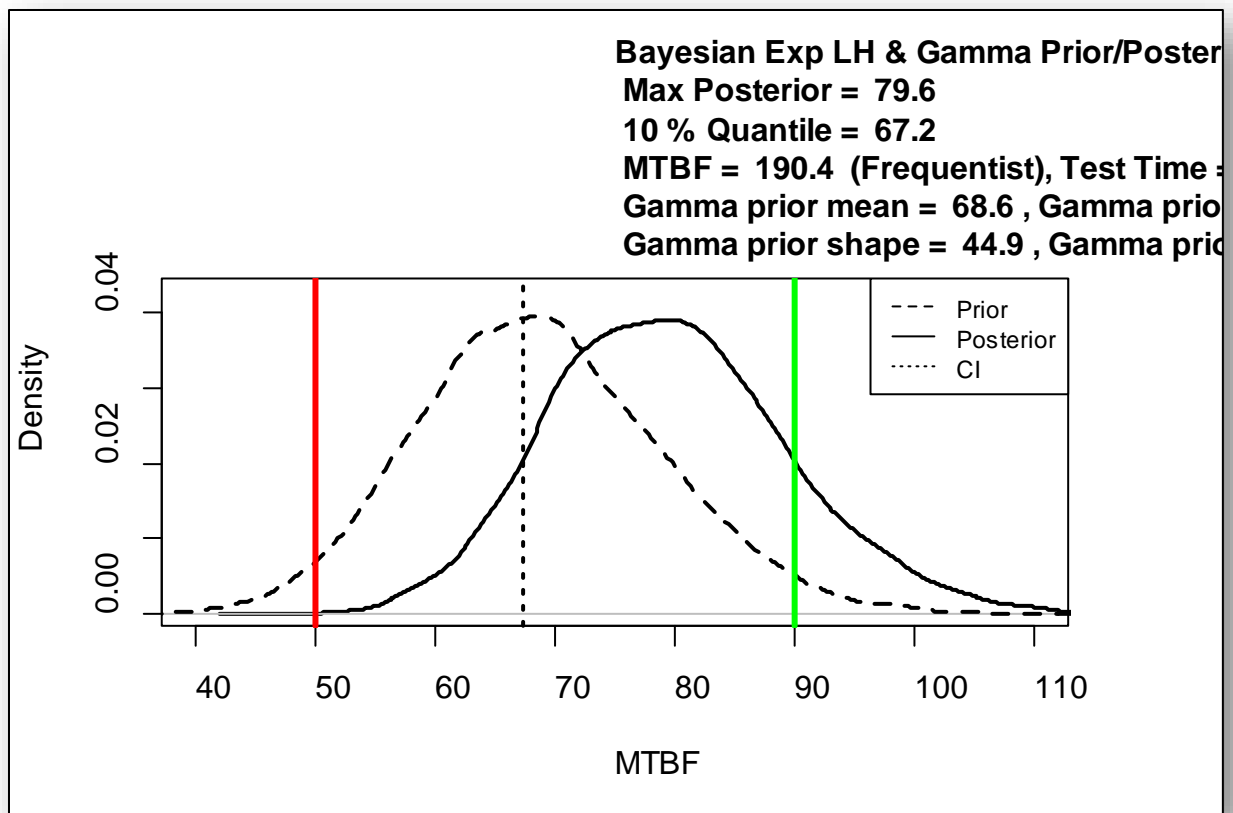


Figure 9: MCMC Solution with 5 Censored Data Points

The comparison yields the following information:

1. Censored data results in higher peak and CI values (as expected)
2. The censored data analysis results in a lower error from the true value
3. Frequentist MTBF is overly optimistic given only 5 observed failures (190.4)

Table 3: Comparison of Uncensored and Censored Data Analysis

	10 Uncensored Points (Figure 2)	5 of 10 Censored Points (Figure 9)
Most Likely (% error)	70.8 (12%)	79.6 (1%)
10% Quantile Value	61.4	67.2

There are frequentist methods that correctly deal with censored data. These are not covered here and there is no implication that method would produce an incorrect result. This appendix simply deals with the comparison between the Bayesian methods when censoring is present.