

# RAMS Workshop: Reliability Modeling Toolkit for Reliability and Accelerated Testing Analysis

---

Sunday, January 18th and 1:30pm – 5:30pm

Reuel Smith (PhD)

University of Maryland Center for Risk and Reliability



# Objectives and Goals

- To demonstrate the Reliability Modeling Toolkit's functions using real-world based example problems
- To teach attendees how to apply this toolkit for analyzing probability modeling and accelerated life and degradation testing problems
- To give attendees hands-on training in downloading, installation, and application of this toolkit



# What is the Reliability Modeling Toolkit?

- An R library made up of computer scripts
- Solves computationally involved and challenging reliability modeling and analytics including:
  - Life probability distribution analysis
  - Probabilistic physics of failure (PPoF)
  - Accelerated testing: Accelerated life testing (ALT) and accelerated degradation testing (ADT) data analysis
- Free of charge, and is governed by a permissive open-source use license for students, practitioners, and the public

# RMT

RELIABILITY MODELING TOOLKIT



# Background of the RMT

- RMT began as a collection of MATLAB, R, and BUGs scripts in 2015
- R was ultimately chosen as RMT's platform for,
  - Open access to the public
  - High use in the engineering industry
  - Extensive function libraries and catalogs
  - Large data analysis and repeated calculations



# Background of the RMT

- First released to UMD students in August 2022 as part of reliability engineering course **Probabilistic Physics of Failure and Accelerated Testing** (part of annual curriculum)
- Also made available in other reliability engineering courses, summer programs, and within the RAMS Tutorial program from 2024 through 2025
- Sought testing and feedback which assisted improvement and further development of RMT tools
- Most recent version (v1.5.0.1) of RMT currently available for download on GitHub

<https://github.com/Center-for-Risk-and-Reliability/RMT>



# RMT Feature Comparisons

Item	RMT	Weibull++	Minitab	JMP-SAS
Cost	Free/License GPLv3	Outside institution or workplace	Outside institution or workplace	Outside institution or workplace
Least Squares Estimation	✓	✓	✓	✓
Maximum Likelihood Estimation	✓	✓	✓	✓
Bayesian Analysis	✓	✓	×	✓
ALT	✓	✓	✓	✓
ADT	✓	✓	×	✓
Step-Stress ALT	✓	✓	×	✓
Step-Stress ADT	×	✓	×	✓
Life Distribution Models	11	11	5	Multiple
Life-Stress Models	12	9	4	Multiple
Degradation-Life Models	8	6	×	Multiple
Visualization	Good	Good	Good	Extensive



# RMT Improvements and Additions since RAMS 2025

- Improvements made to probability plotting tools including visualization updates, improved labeling, and some set parameter features
- Improved Life-Stress relationship plots
- Improvements to MLE and Bayesian ADT tools
- Some new models and refinement to other models
- Additional output and plot visualization improvements
- Several additional bug fixes



# Installation Instructions

1. Sign into the RAMS or Hotel WiFi service.
2. Download and install latest version of R (<https://www.r-project.org/>).
3. RStudio recommended as an additional download for interface reasons (<https://posit.co/download/rstudio-desktop/>). If installing on a Mac, use Anaconda Navigator to install RStudio.



# Installation Instructions

4. Setup R and RStudio then install the “devtools” library

```
install.packages("devtools")
```

5. Install “cmdstanr” library

```
install.packages("cmdstanr", repos = c('https://stan-dev.r-universe.dev',  
getOption("repos")))
```



# Installation Instructions

## 6. Activate the “devtools” library to connect with GitHub.

```
library(devtools)
```

### Then Install “RMT”

- If Rtools is installed, type the following to build from source

```
devtools::install_github("Center-for-Risk-and-Reliability/RMT", INSTALL_opts = "--install-tests")
```

- If Rtools is not installed, type the following instead

```
devtools::install_github("Center-for-Risk-and-Reliability/RMT", build = FALSE, INSTALL_opts = "--install-tests")
```



# Installation Instructions

## 7. Finally, load RMT library in R

```
library(reliabilityRMT)
```

Installation includes additional libraries that may need updating



# CHECKPOINT

- ✓ Introduced the RMT and learned of its authors, its background, and its function
- ✓ Learned how to install the RMT

# RMT

RELIABILITY MODELING TOOLKIT

NEXT:

RMT Basics – Probability Plotting,  
MLE, and Bayesian Estimation



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

## Nonparametric modeling:

- Strictly based on raw data (failure and censored) and rank of data
- No defined model

## Parametric modeling:

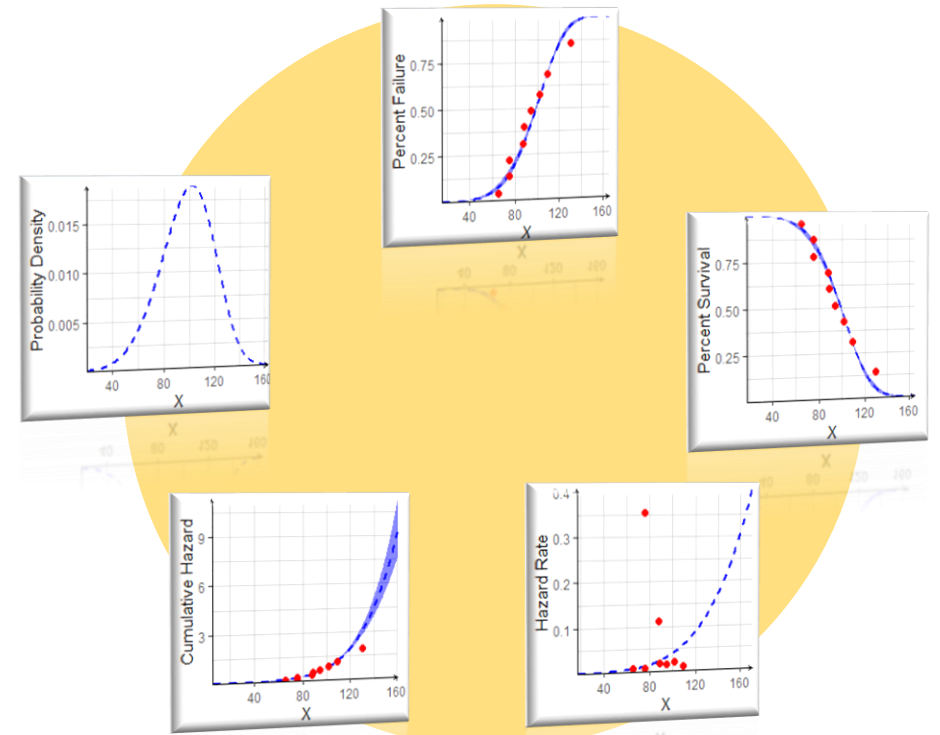
- Requires statistical checks to determine fitness
- May adhere to *many* defined models

- **Definition:** To go from raw data to a defined reliability model, requires raw estimators for reliability metrics called **probability plotting positions**



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

- Nonparametric estimates exist for:
  - Failure probability (cumulative density function, CDF),  $F(x)$
  - Reliability (Survival) function,  $R(x)$  (or complementary CDF  $1 - F(x)$ )
  - Hazard function (failure rate),  $h(x)$
  - Cumulative hazard function,  $H(x)$
  - Probability density (PDF),  $f(x)$
- However there is a *large* selection of probability plotting positions to choose from



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

Plotting Position Model	Plotting Position for CDF, $F_i(x_i)$	Plotting Position Model	Plotting Position for CDF, $F_i(x_i)$
Blom	$F_i(x_i) = \frac{i - 0.375}{n + 0.25}$	Jenkinson's (Beard's)	$F_i(x_i) = \frac{i - 0.31}{n + 0.38}$
Mean	$F_i(x_i) = \frac{i}{n + 1}$	Bernard & Bos-Levenbach	$F_i(x_i) = \frac{i - 0.3}{n + 0.2}$
Median	$F_i(x_i) = \frac{i - 0.3}{n + 0.4}$	Tukey	$F_i(x_i) = \frac{i - 1/3}{n + 1/3}$
Midpoint	$F_i(x_i) = \frac{i - 0.5}{n}$	Grigorten	$F_i(x_i) = \frac{i - 0.44}{n + 0.12}$
Nelson-Aalen	$F_i(x_i) = 1 - \exp\left(-\sum_{x_i \leq x} \frac{d_i}{n_{i-1} - d_{i-1} - c_{i-1}}\right)$	Kaplan-Meier	$F_i(x_i) = 1 - \prod_{x_i \leq x} \left(1 - \frac{d}{j}\right)$

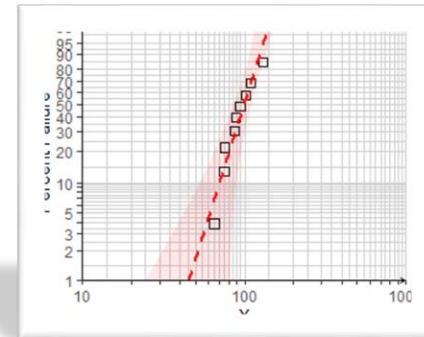
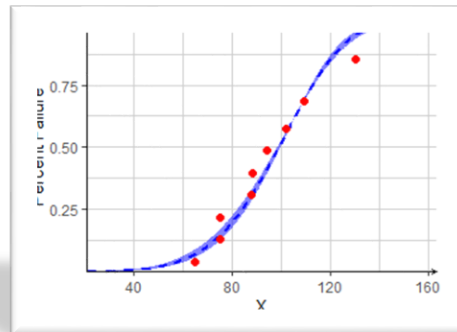
- $i$  – data ranking at failure time  $x_i$
- $n$  – total number of data in set
- $d$  – number of failed units

- $c$  – number of censored units
- $j$  – reverse-rank or the reverse order of cumulative ranking metric  $i$



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

- Regression analysis is an important tool in reliability analysis because it is often an initial procedure used to assign the best distribution to a set of data
- The basis of probability plotting where X-axis and Y-axis are based on the linearization of a CDF



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

**Example:** Take the *Weibull CDF*,

$$F(x) = 1 - \exp\left[-\left(\frac{x}{\alpha}\right)^\beta\right]$$

where  $\alpha$  is the scale parameter and  $\beta$  is the shape parameter

- By log-linearization we get,

$$\ln\left\{-\ln[1 - F(x)]\right\} = \beta \ln x - \beta \ln \alpha$$

Least Squares Parameter Estimates:

$$\hat{\beta} = \text{slope}, \quad \hat{\alpha} = \exp\left(-\frac{\text{intercept}}{\text{slope}}\right)$$

- There are however some distributions where linearization is not 100% feasible



# Overview of Probability Plotting: Non-parametric vs. Parametric Modeling

**Example:** Now take the *Three-Parameter Weibull CDF*,

$$F(x) = 1 - \exp \left[ - \left( \frac{x - \gamma}{\alpha} \right)^\beta \right]$$

which adds  $\gamma$  as the location parameter

- A partial linearization is required with focus on minimizing the sum of square error (SSE).

$$\ln \left\{ -\ln [1 - F(x)] \right\} = \beta \ln (x - \gamma) - \beta \ln \alpha$$

$$[\hat{\alpha}, \hat{\beta}, \hat{\gamma}] = \min \left\langle \sum_{i=1}^n \left( \ln \left\{ -\ln [1 - F_i(x_i)] \right\} - \beta \ln (x - \gamma) + \beta \ln \alpha \right)^2 \right\rangle$$

- Some CDFs including this are also subject to constraints on the location parameter



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation

## Probability Plotting Tools

```
probplot.DIST(data, pp, xlabel, confid, nobounds, ..., stressunit1, stressunit2)
```

- “*data*” – primary event data, censored status, and stress level in tabular form
- “*pp*” – Named probability plotting position model (ex. Enter “Blom” for Blom probability plotting position model)
- “*xlabel*” – label for x-axis in plot
- “*confid*” – confidence bound between 0 and 1 (0.95 for 95% confidence)
- “*nobounds*” – (NEW) check as “1” to turn off the confidence bounds
- “*stressunit1*” – (NEW) Label for the first stress unit. If left blank or NULL, the default label is “unit”.
- “*stressunit2*” – (NEW) Label for the second stress unit (if applicable). If left blank or NULL, the default label is “unit”.
- **NOTE**: Several additional inputs have been added as well between “*nobounds*” and “*stressunit1*” (see help guide)



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation

RMT currently has the eleven distributions for probability plotting:

Probability Distribution	RMT Designation	Probability Distribution	RMT Designation
Normal	<i>DIST = “nor”</i>	Logistic	<i>DIST = “logist”</i>
Lognormal	<i>DIST = “logn”</i>	Loglogistic	<i>DIST = “loglogist”</i>
Weibull	<i>DIST = “wbl”</i>	Gumbel	<i>DIST = “gumb”</i>
Three-Parameter Weibull	<i>DIST = “wbl3P”</i>	Gamma	<i>DIST = “gam”</i>
Exponential	<i>DIST = “exp”</i>	Generalized or Three-Parameter Gamma	<i>DIST = “gam3P”</i>
Two Parameter Exponential	<i>DIST = “exp2P”</i>		

**Example:** Enter **probplot.wbl** to generate a Weibull probability plot



# Probability Plotting Tools:

## Parameter Estimation by Least Squares Estimation

**Example:** A pair of stress tests were performed on two sets of components. The first set of three was run at 150 °C and the second set of nine was run at 200 °C. All components failed at the listed times.

Test Temperature (°C)	Number of Components in Set	Failure Times (hours)
150	3	2350, 2560, 1980
200	9	220, 250, 330, 370, 380, 460, 460, 510, 610

- We can use the probability plotting tools to show the fit and estimate the probability distribution parameters
- Start by entering the data...



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation

- Most tools that use probability plotting toolset have a specific format for entering test data:

Define data:

```
data_RAMs.Pt1.Ex1 <- cbind(c(2350, 2560, 1980,  
                             220, 250, 330, 370, 380, 460, 460, 510, 610),  
                           c(rep(1,3), rep(1,9)),  
                           c(rep(150+273,3),rep(200+273,9)))
```

- **Column 1** – Primary event (time usually) data
- **Column 2** – Censored status for Column 1 (0 for right censored, 1 for not censored)
- **Column 3 and up** – Stress levels for Column 1



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation

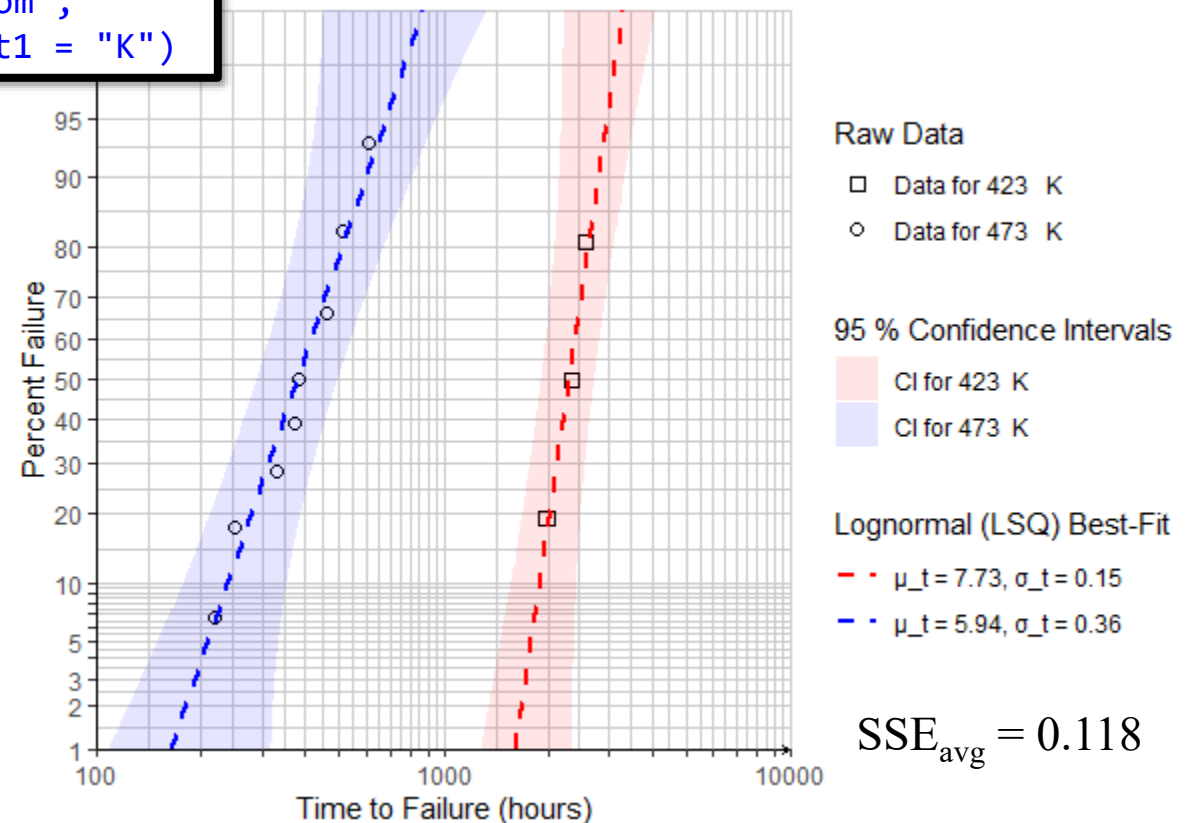
## Input:

```
probplot.logn(data = data_RAMs.Pt1.Ex1, pp = "Blom",  
             xlabel = "Time to Failure (hours)", stressunit1 = "K")
```

## Output includes:

- Probability Parameter estimates
- SSE and coefficient of determination  $R^2$
- Nonparametric tabular estimates

Try this with the Weibull probability plot tool  
**probplot.wbl** or another distribution type



# Probability Plotting Tools:

## Parameter Estimation by Least Squares Estimation

**Example:** Let's try with right censored data now. Take twenty-four aluminum samples that are fatigue tested at different constant cyclic stress amplitudes. Eight units were each tested at 200, 300, and 475 MPa but a few survived at the cycles given.

Test Alternating Stress (MPa)	Number of Components in Set	Failure Cycles	Right-Censored Cycles
200	8	250, 460, 530, 730, 820, 970, 1530	970
300	8	160, 180, 290, 320, 390, 460	500, 500
475	8	23, 90, 100, 150, 180, 220, 230	250

Which distribution would be the best for modeling this data, Weibull or Lognormal? Use the RMT to find out.

Now try entering the data while entering **0** for right-censored observations



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation

- Data would be entered as:

Define data:

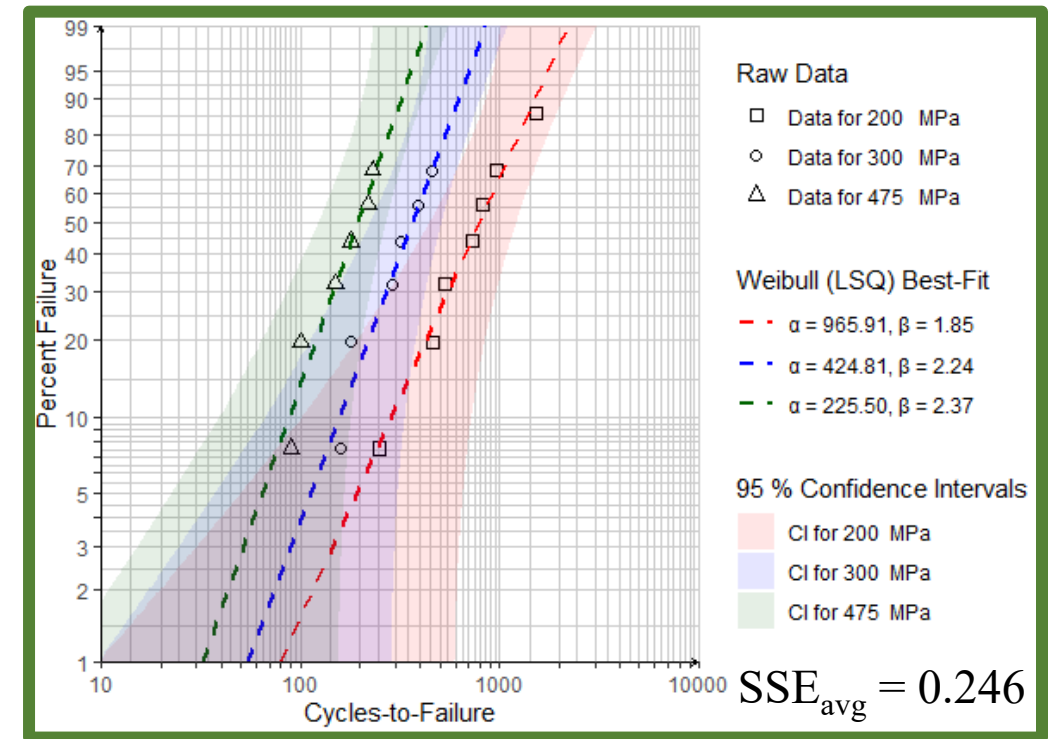
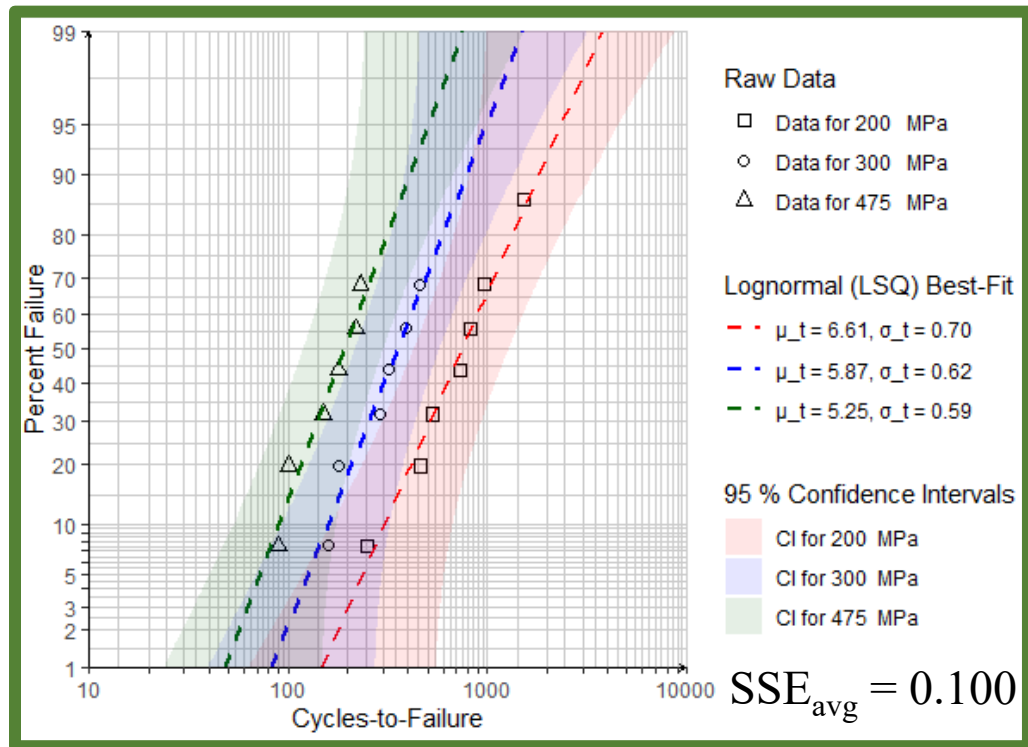
```
data_RAMs.Pt1.Ex2 <- cbind(c(250, 460, 530, 730, 820, 970, 1530,  
                             970,  
                             160, 180, 290, 320, 390, 460,  
                             500, 500,  
                             90, 100, 150, 180, 220, 230, 230,  
                             250),  
                           c(rep(1,7),0,rep(1,6),rep(0,2),rep(1,7),0),  
                           c(rep(200,8),rep(300,8),rep(475,8)))
```

Input:

```
probplot.logn(data = data_RAMs.Pt1.Ex2, pp = "Blom",  
              xlabel1 = "Cycles-to-Failure", stressunit1 = "MPa")
```



# Probability Plotting Tools: Parameter Estimation by Least Squares Estimation



- It's the Lognormal Distribution!

# Overview of Maximum Likelihood Estimation

- Generally, an LSQ estimate is a good start, but *maximum likelihood estimates* are the standard in reliability modeling
- **Definition: Maximum likelihood estimation (MLE)** handles most data analysis purposes including parameter estimation (for parameter set  $\bar{\Theta}$  ), distribution fitting, and confidence intervals of parameter estimates.



# Overview of Maximum Likelihood Estimation

- **Definition**: MLE makes use of a **likelihood**  $\ell$  which is joint distribution that accounts for all  $n$  failure data and all  $m$  right-censored data

$$\ell = \prod_{i=1}^n f(x_i | \bar{\Theta}) \prod_{j=1}^m R(x_j | \bar{\Theta})$$

- **Definition**: The natural log of a likelihood is a **log-likelihood**  $\Lambda$  which in many cases is easier to work with than just the likelihood

$$\Lambda = \sum_{i=1}^n \ln[f(x_i | \bar{\Theta})] + \sum_{j=1}^m \ln[R(x_j | \bar{\Theta})]$$



# Overview of Maximum Likelihood Estimation

- But LSQ estimates are still very necessary in MLE
- R uses optimization analysis to perform MLE (as most platforms) to satisfy partial differential relations of likelihood (or log-likelihood) w.r.t. every parameter equal to zero

$$0 = \frac{\partial \ell}{\partial \theta_1}; 0 = \frac{\partial \ell}{\partial \theta_2} \text{ or } 0 = \frac{\partial \Lambda}{\partial \theta_1}; 0 = \frac{\partial \Lambda}{\partial \theta_2}$$

- where  $\bar{\Theta} = [\theta_1, \theta_2]$

- The built-in *Non-linear Minimization (nlm)* R function requires an initial estimate to operate. Random or faulty initials would likely result in a local minimum thus the LSQ would be a good starting point.



# Maximum Likelihood Estimation Tools: Probability Parameter Estimation by MLE

## Maximum Likelihood Estimator for Probability Distributions Tool

```
distribution.MLEest(LSQest,dist,xi,rc,confid,sided)
```

- “*LSQest*” – vector of the initial parameter estimates
- “*dist*” – named probability distribution
- “*xi*” – vector of failure or primary event data in a given set
- “*rc*” – vector right censored data of a given set
- “*confid*” – confidence bound between 0 and 1 (0.95 for 95% confidence)
- “*sided*” – confidence limits for parameters: two-sided, one-sided high, or one-sided low



# Maximum Likelihood Estimation Tools: Probability Parameter Estimation by MLE

## Probability Plotting Tools (with MLE)

```
probplot.DIST(data, pp, xlabel, confid, nobounds, ..., MLE_i,..., stressunit1. stressunit2)
```

- “ $MLE\_i$ ” – (**NEW**) Activates calculation of the maximum likelihood estimates (MLE) for the parameters and plot for the probability plot. Default is NULL for LSQ. Enter 1 to compute the MLE.

**Example:** Take the probability plot from the *first example* and find the MLE probability parameter estimates for each stress level.



# Maximum Likelihood Estimation Tools: Probability Parameter Estimation by MLE

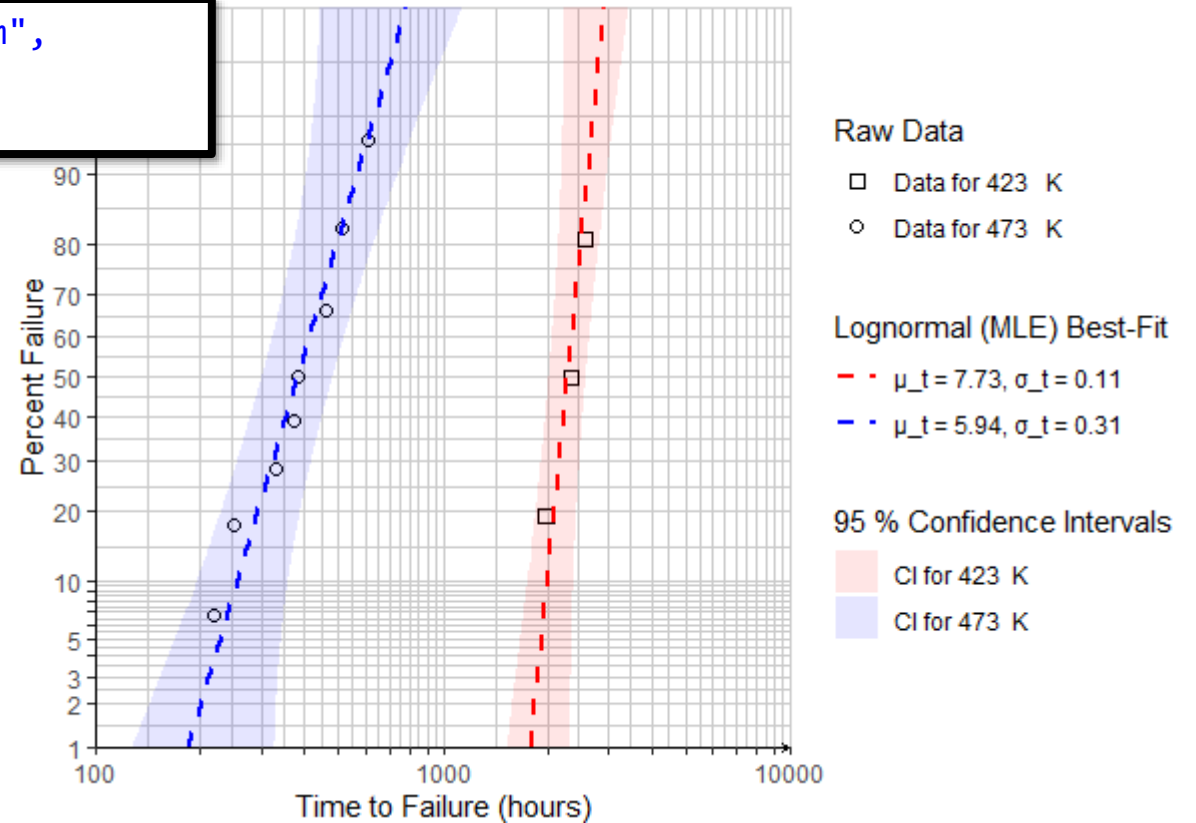
## Input:

```
probplot.logn(data= data_RAMs.Pt1.Ex1, pp = "Blom",  
             xlabel = "Time to Failure (hours)", MLE_i = 1,  
             stressunit1 = "K")
```

## Output includes:

- MLE probability Parameter estimates
- SSE
- Upper and Lower Confidence Intervals of MLE estimates
- Likelihood and log-likelihood,
- AIC and BIC

Try this with the Weibull probability plot tool  
**probplot.wbl** now



# Maximum Likelihood Estimation Tools: Probability Parameter Estimation by MLE

- Lognormal

Stress	$\mu_t$	$\mu_t$ Lower CI	$\mu_t$ Upper CI	$\sigma_t$	$\sigma_t$ Lower CI	$\sigma_t$ Upper CI	loglik	AIC
423 K	7.73	7.61	7.85	0.11	0.05	0.24	-20.75	45.50
473 K	5.94	5.74	6.15	0.31	0.20	0.50	-55.78	115.55

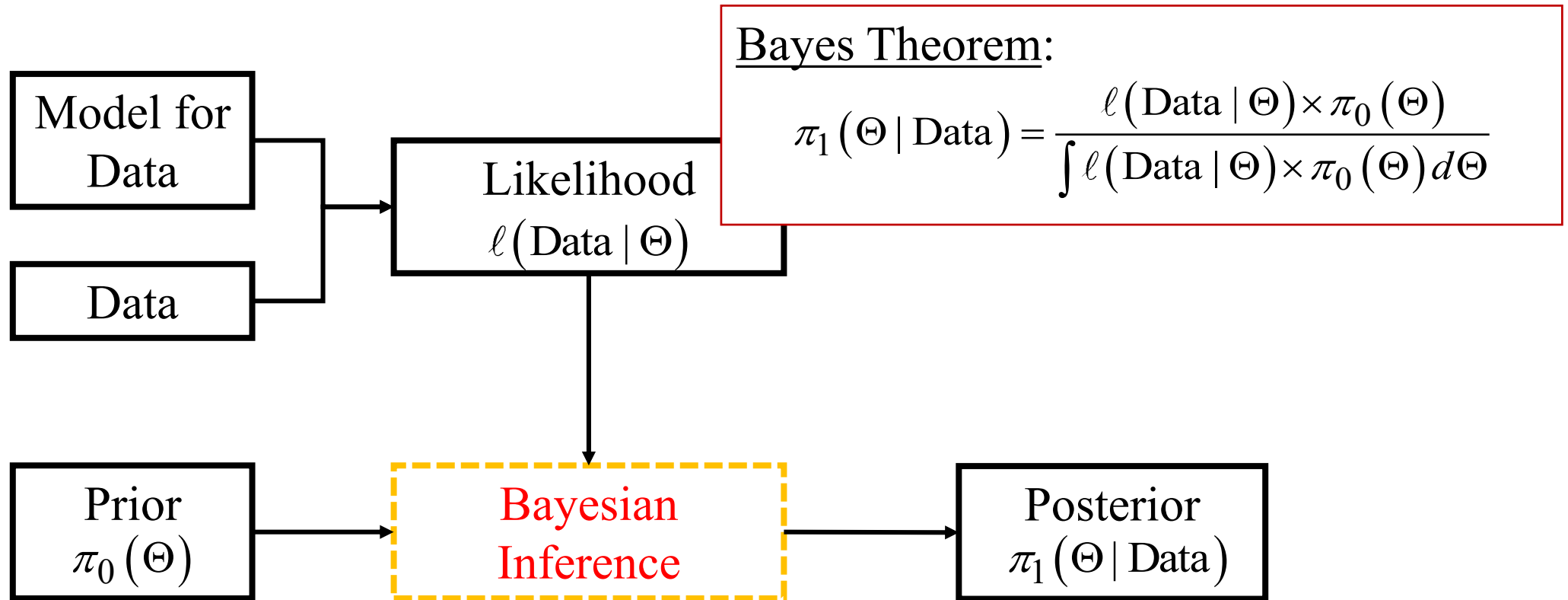
- Weibull

Stress	$\alpha$	$\alpha$ Lower CI	$\alpha$ Upper CI	$\beta$	$\beta$ Lower CI	$\beta$ Upper CI	loglik	AIC
423 K	2402.61	2174.05	2655.21	11.94	4.67	30.51	<u>-20.58</u>	<u>45.16</u>
473 K	442.23	368.44	530.81	3.78	2.27	6.30	<u>-55.61</u>	<u>115.21</u>

**Example:** Now practice with the data from for the *second example*.



# Overview of Bayesian Estimation: Bayesian Inference Approach



# Overview of Bayesian Estimation: Implementation of the RStan Library

- All Bayesian operations on RMT are done with the aid of the RStan computational library
- RStan is an R interface to the Stan code library, widely used for Bayesian statistical inference and sampling
- RStan (and by extension the RMT) makes use of the *No U-Turn Sampler* (aka NUTS) which is an extension of Hamiltonian Monte Carlo, a variant of Metropolis Hastings MCMC that uses Hamiltonian dynamics



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

### Bayesian Updater for Probability Distributions Tool

```
distribution.BAYESest(pt_est,dist,TTF,Tc,confid,priors,nsamples,burnin,nchains)
```

- “*pt\_est*” – vector of the initial parameter estimates
- “*dist*” – Named probability distribution used as the base for the likelihood  $\ell(\text{Data} \mid \Theta)$
- “*TTF*” – vector of failure or primary event data in a given set
- “*Tc*” – vector right censored data of a given set
- “*confid*” – confidence bound between 0 and 1 (0.95 for 95% confidence)
- “*priors*” – vector of prior distributions for each parameter
- “*nsamples*” – number of MCMC samples or iterations per chain
- “*burnin*” – number of initial MCMC iterations the throw out
- “*nchains*” – number of Markov chains



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

**Example:** Consider a prototype transistor for which we don't know what its failure rate is. If Test Facility A had a small batch of two tests conducted at nominal voltage and they got failures at 130 and 1400 hours, use the Bayesian RMT tool to find the failure rate distribution.

- **Definition:** Remember, when we know nothing about a parameter's prior behavior we typically use an uninformative prior. In many cases this is a uniform distribution which is what we will use as our prior input for this solution.



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

- Define the prior as R vector data and place distribution for each parameter of interest in quotes

Define:

```
priorset <- c("uniform(0,1)")
```

- **NOTE**: See Stan documentation to be sure distribution naming and parameter location is correct; otherwise your analysis may be faulty

- The likelihood in this case may be defined by an exponential likelihood  $\ell(\text{Data} = x_{i=1\dots n}, x_{j=1\dots m} \mid \lambda) = \lambda^n \exp \left[ -\lambda \left( \sum_{i=1}^n x_i + \sum_{j=1}^m x_j \right) \right]$

- Now run the tool as follows:

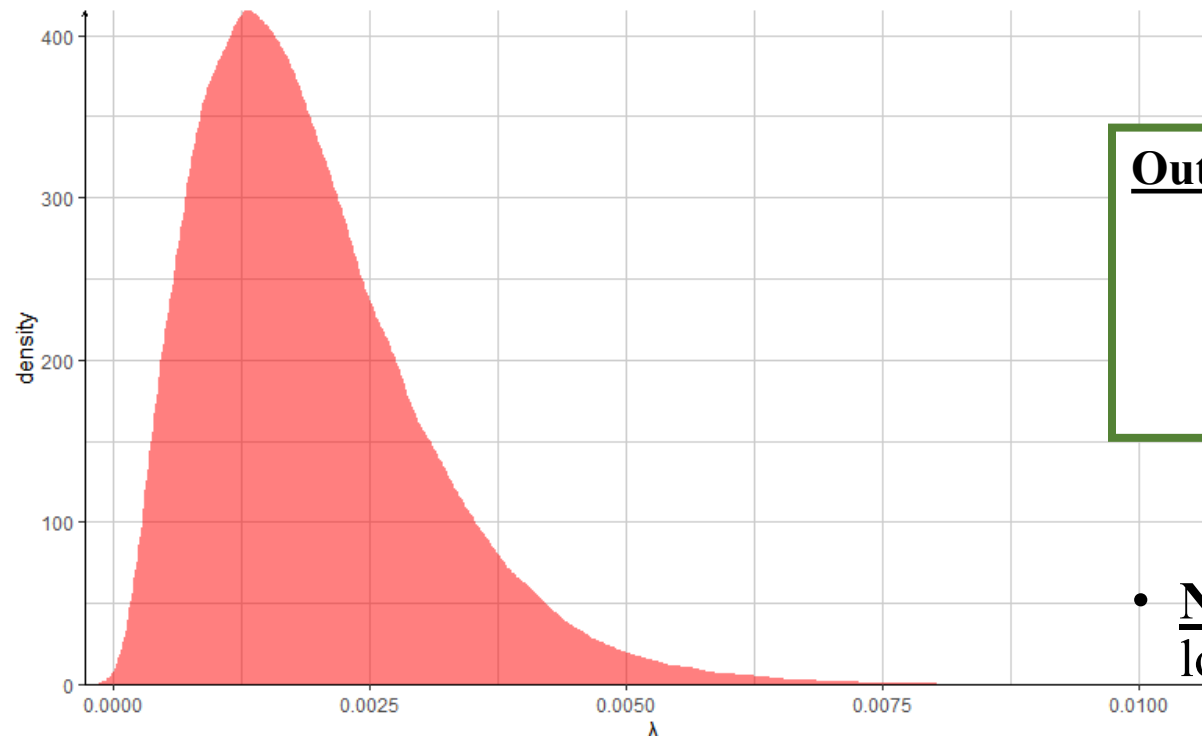
Input:

```
Dat.out <- distribution.BAYEsest(pt_est = c(0.05),  
                                dist = "Exponential", TTF= c(130,1400),  
                                confid = 0.9,  
                                priors = priorset,  
                                nsamples = 20000,  
                                burnin = 1000, nchains = 4)
```



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation



### Output includes:

- Complete posterior MC chains per parameter
- Table of posterior stats
- Plots of posterior MCMC trace, histogram, and density

- **NOTE:** The more complex the likelihood the longer it may take

Posterior	Mean	SDev	Lower 95%	Median	Upper 95%	Rhat
$\hat{\lambda}$	0.00194	0.00113	0.000514	0.00173	0.00408	1.00



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

- **NOTE**: Take care of any warnings generated by RStan and by extension RMT tools that make use of RStan

**Example:** A very common warning deals with the **rhat** parameter of any parameter chain.

- If **rhat** is not equal to or close to 1, RStan will warn you as such as it means that convergence didn't occur
- Model may need to be checked or brought to the attention of the developer
- Usually however this may be remedied by modification of the input (burnin, sample number, initial estimates, etc.)



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

**Example:** Now let's say that we get some additional test data of this transistor from Test Facility B after we have run this Bayesian analysis.

Test Facility B had a batch of ten transistors run for three months or 2160 hours and only observed six failures at 430, 534, 560, 560, 1403, and 2020 hours. If nominal voltage was applied again, use the Bayesian RMT tool to find the *updated* failure rate distribution.

- Here we need to establish a new prior based on the posterior output of the last Bayesian analysis.
- Pull the Markov Chains from the output and fit it to a distribution

Define:

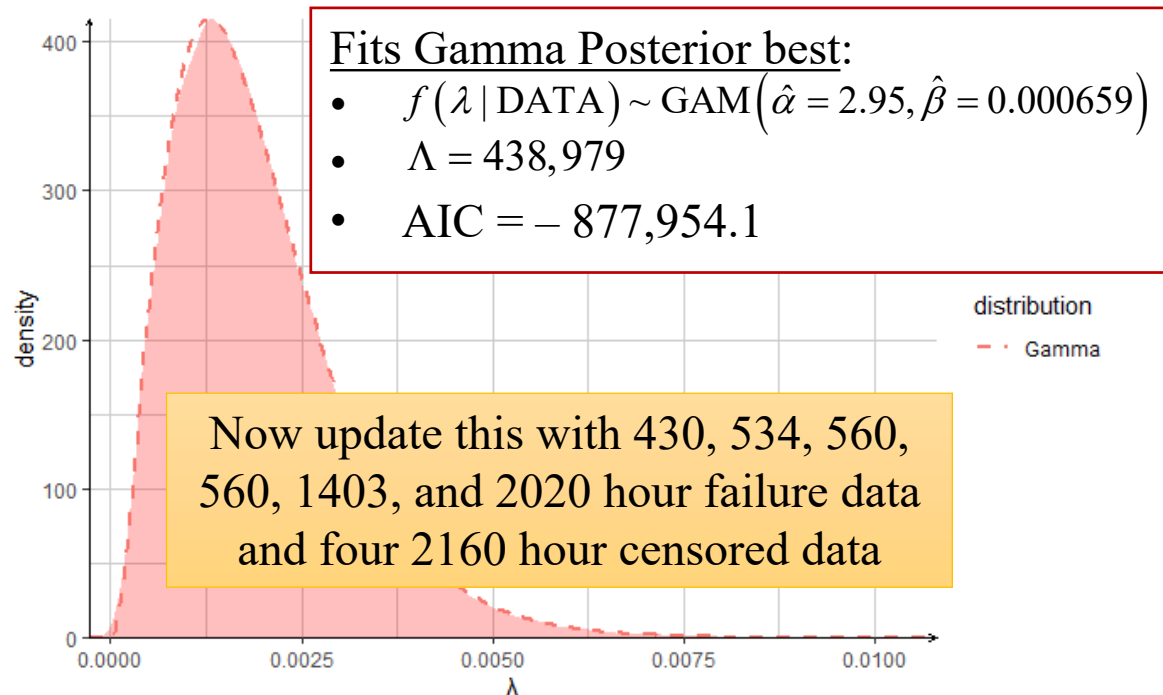
```
Post.dat <-  
extract(Dat.out$posterior.fit,c("lambda"))$lambda
```



# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

- We can fit this with an RMT distribution fitting tool similar to others like it (Example: Python library's `Fit_Everything` tool)



### Distribution Rank and Fit Tool (BETA)

```
distribution.fit(data)
```

- “*data*” – primary data as a vector

#### Output includes:

- MLE parameter estimates
- Log-likelihood
- AIC

- **NOTE:** The more data to process, the longer it takes

# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

- Run the following

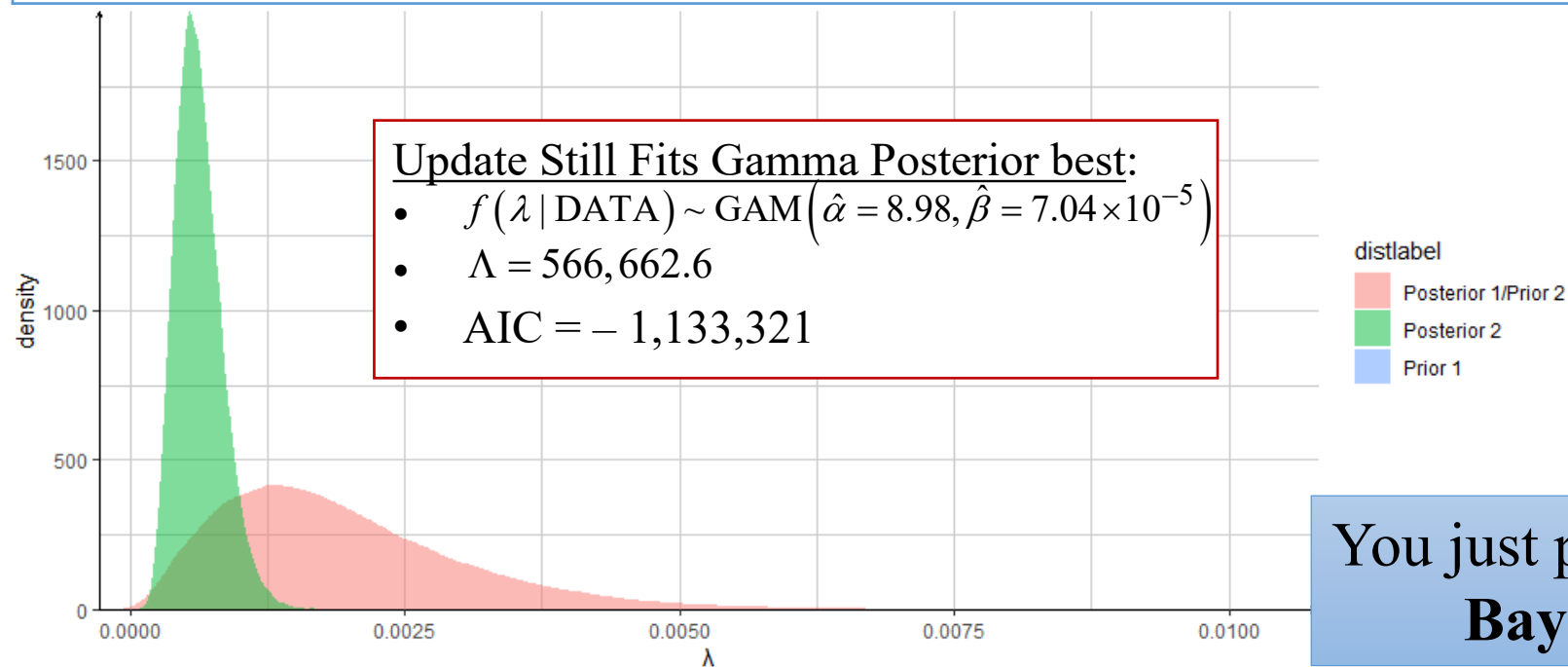
Input:

```
Dat.out2 <- distribution.BAYESest(pt_est = c(0.05),"Exponential",  
                                TTF= c(430,534,560,560,1403,2020), Tc = rep(2160,4),  
                                confid = 0.9,  
                                priors = c("gamma(2.964625, 0.0006598385)"),  
                                nsamples = 20000,  
                                burnin = 1000,nchains = 4)
```

# Bayesian Estimation Tools:

## Parameter Update by Bayesian Estimation

	Mean	SDev	Lower 95%	Median	Upper 95%	Rhat
Last Post	0.00194	0.00113	0.000514	0.00173	0.00408	1.00
<b>New Post.</b>	<b>0.000632</b>	<b>0.000211</b>	<b>0.000480</b>	<b>0.000608</b>	<b>0.000759</b>	<b>1.00</b>



You just performed a **sequential Bayesian procedure!**

# CHECKPOINT

- ✓ Introduced applications of least-squares, MLE, and Bayesian estimation tools used by the RMT
- ✓ Entered time-to-failure data for different stress levels and fit to different distributions
- ✓ Applied least-squares and MLE estimation visualized through probability plots
- ✓ Performed a sequential Bayesian operation
- ✓ Interpreted fitness by SSE, AIC, and loglikelihood

# RMT

RELIABILITY MODELING TOOLKIT

NEXT:

Accelerated Life Testing

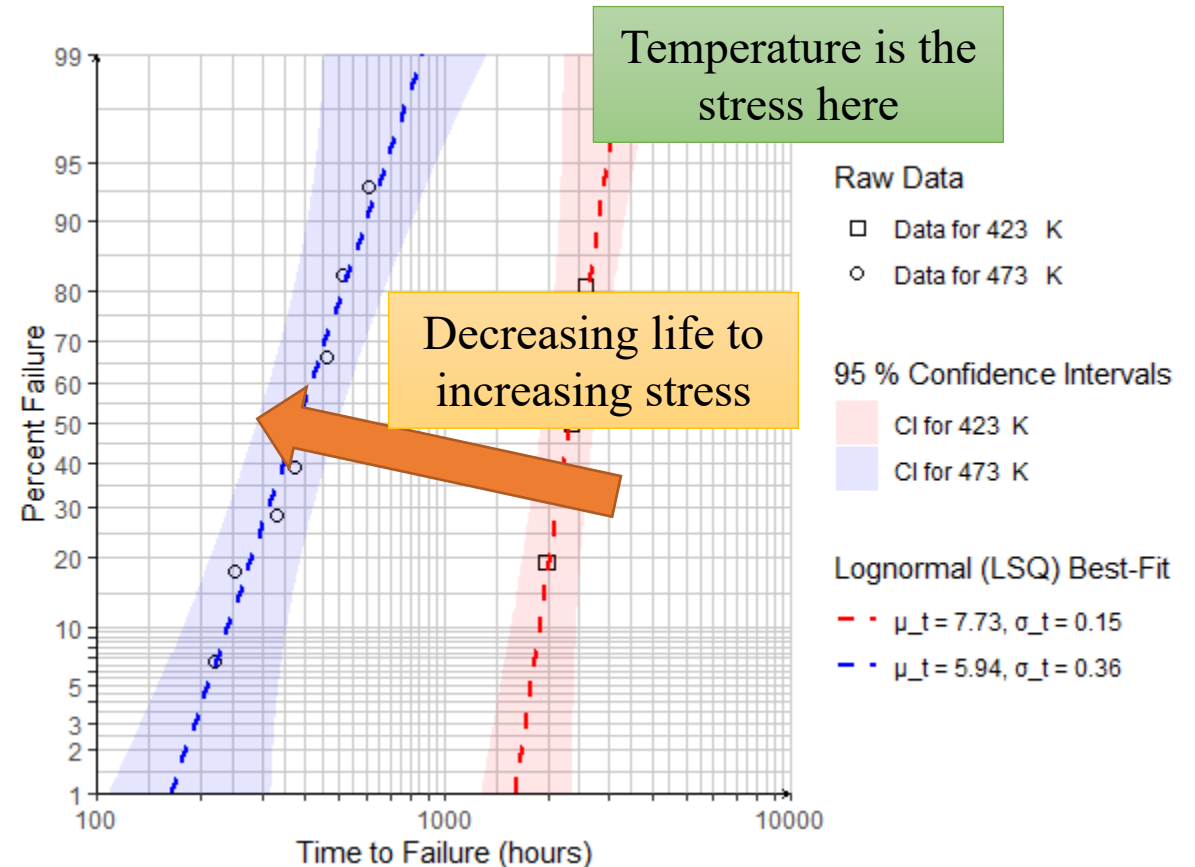


# Overview of Accelerated Life Testing: Life-Stress Relationship

- **QUESTION:** Why have we been separating certain chunks of data by stress?
- **ANSWER:** Because service life is affected by the **physics-of-failure** (or **probabilistic physics-of-failure**) of a unit which relates life (or life distribution) to operational stress or stresses:
  - Temperature
  - Voltage
  - Relative humidity
  - Force or load

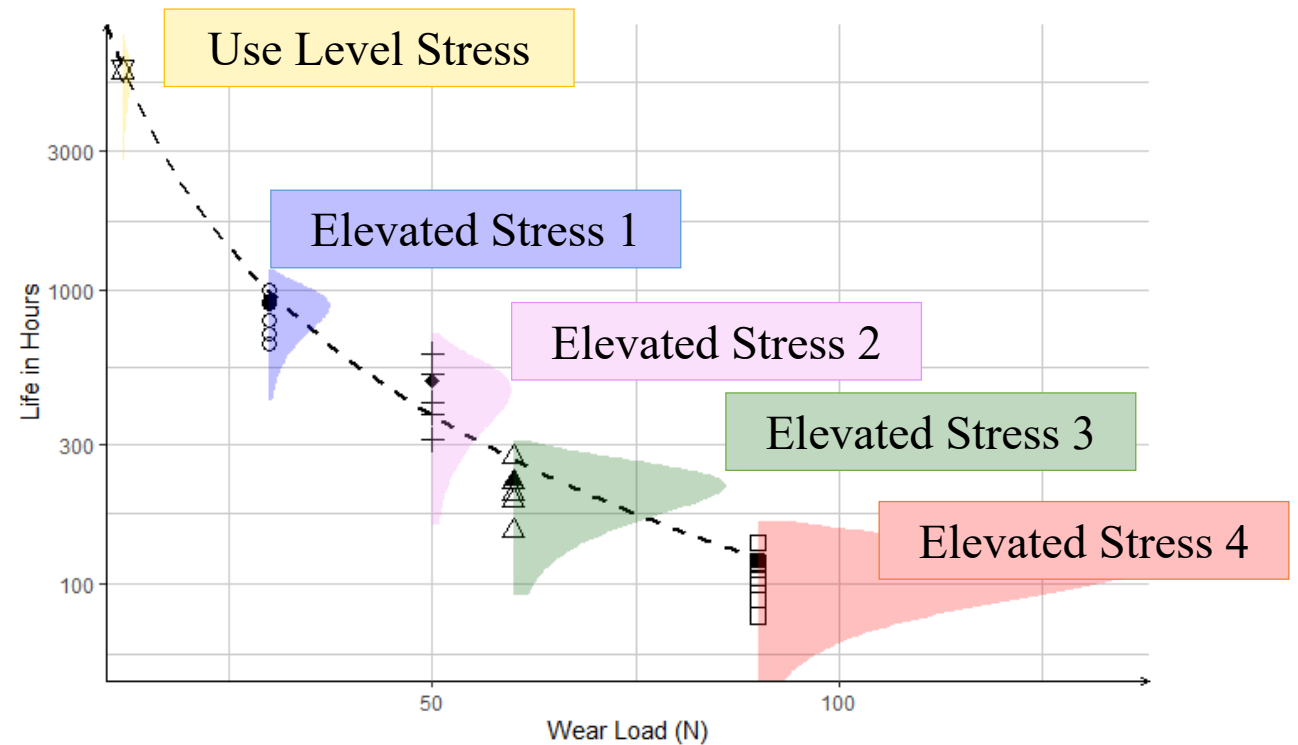
Or to certain physical phenomena:

- Fatigue
- Corrosion
- Creep
- Wear



# Overview of Accelerated Life Testing: Life-Stress Relationship

- **Definition:** Accelerated life testing (ALT) is the methodology where several tests are run at accelerated stress levels  $S_{acc}$  and the accelerated life data therein is used to extrapolate the use life at nominal or use stress levels  $S_{use}$ .
- Many life-stress relations  $l(S)$  are available for modeling such data. The RMT considers at least twelve such models.



# Overview of Accelerated Life Testing: Life-Stress Relationship

Life-Stress Model	Life-Stress function, $l(S)$	Life-Stress Model	Life-Stress function, $l(S)$
Linear	$l(S) = b + aS$	Inverse Power	$l(S) = bS^{-a}$ or $l(S) = \frac{1}{bS^a}$
Exponential	$l(S) = b \exp(aS)$ or $l(S) = b \exp\left(\frac{a}{S}\right)$	Logarithmic	$l(S) = b + a \ln S$
Arrhenius	$l(S) = b \exp\left(\frac{E_a}{K_B S}\right)$	General Exponential Multi-Stress	$l(S_1, \dots, S_n) = \exp(a_0 + a_1 S_1 + \dots + a_n S_n)$
Eyring	$l(S) = \frac{b}{S} \exp\left(\frac{a}{S}\right)$	Temperature-Humidity	$l(S, H) = A \exp\left(\frac{a}{S} + \frac{b}{H}\right)$
(Alt.) Eyring	$l(S) = \frac{1}{S} \exp\left[-\left(a - \frac{b}{S}\right)\right]$	Generalized Eyring	$l(S, U) = \frac{1}{S} \exp\left[\left(a + \frac{b}{S}\right) + \left(c + \frac{d}{S}\right)U\right]$
Power	$l(S) = bS^a$	Power-Exponential	$l(S, U) = \frac{c}{U^b \exp\left(-\frac{a}{S}\right)}$

- $a, b, c, d$  – model parameters
- $E_a$  – Activation energy (eV)
- $k_B$  – Boltzmann constant  $8.617 \times 10^{-5}$  eV/K
- $U$  – nonthermal stress
- $H$  – humidity or relative humidity
- $S$  – thermal or general stress



# Overview of Accelerated Life Testing: Life-Stress Relationship

- All distributions have parameters that are replaced by governing life-stress model to effectively conduct ALT analysis

**Example:** Normal, Lognormal, Weibull, and Exponential distributions require the following replacements

Standard Life Distribution (PDF)	Parameter to replace	Life-Stress Distribution (PDF)
Normal $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$	Mean $\mu = l(S)$	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-l(S)}{\sigma}\right)^2\right]$
Lognormal $f(x) = \frac{1}{\sigma_t x \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln x - \mu_t}{\sigma_t}\right)^2\right]$	Log-mean $\mu_t = \ln l(S)$	$f(x) = \frac{1}{\sigma_t x \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln x - \ln l(S)}{\sigma_t}\right)^2\right]$
Weibull $f(x) = \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left[-\left(\frac{x}{\alpha}\right)^\beta\right]$	Scale parameter $\alpha = l(S)$	$f(x) = \frac{\beta}{l(S)} \left(\frac{x}{l(S)}\right)^{\beta-1} \exp\left[-\left(\frac{x}{l(S)}\right)^\beta\right]$
Exponential $f(x) = \lambda \exp(-\lambda x)$	Failure rate $\lambda = \frac{1}{l(S)}$	$f(x) = \frac{1}{l(S)} \exp\left[-\left(\frac{1}{l(S)}\right)x\right]$



# Accelerated Life Testing Tools: Life-Stress Model Selection

## Life-Stress Model Selector Tool

```
lifestress.select(ls)
```

- “ls” – Named life-stress model

**Example:** Calling the Linear life-stress model pulls a list of functions for life and log-life

Input:

```
lifestress.select("Linear")
```

- All life-stress models can be called as functions for general use in R or in other RMT tools such as...

Output:

```
[[1]]  
function(lparams,s) {  
  lparams[2] + s*lparams[1]  
}  
<bytecode: 0x000001aa0d62e1c0>  
<environment: 0x000001aa0daa4258>  
  
[[2]]  
function(lparams,s) {  
  log(lparams[2] + s*lparams[1])  
}  
<bytecode: 0x000001aa0d63c930>  
<environment: 0x000001aa0daa4258>
```



# Accelerated Life Testing Tools: Acceleration Factor Calculation

## Acceleration Factor Calculator Tool

```
accelfactor(params,ls,S_acc,S_use)
```

- “*params*” – vector of life-stress model parameters (see documentation for order)
- “*ls*” – Named life-stress model
- “*S\_acc*” – accelerated stress or stress vector (for dual or higher stresses)
- “*S\_use*” – use level stress or stress vector (for dual or higher stresses)
- **Definition:** A common metric in ALT is the **acceleration factor**  $AF$ , which is a ratio between use life and accelerated life at a given accelerated stress

$$AF = \frac{l(S_{\text{use}})}{l(S_{\text{accelerated}})}$$



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Least Squares

- Tools for ALT analysis share many of the same input as standard distribution fitting tools with some new input.

### Least-Squares Life-Stress Estimator Tool

```
lifestress.LSQest(data,ls,dist,pp,xlabel1,Suse,Llab,Slab,Slab2, ..., stressunit1. stressunit2)
```

- “*ls*” – Named life-stress model (see `lifestress.select` help file for nomenclature)
- “*dist*” – Life distribution definition
- “*Suse*” – Use level or nominal stress
- “*Llab*” – Life label for life-stress plot
- “*Slab*” – Stress label for life-stress plot
- “*Slab2*” – Second stress label for life-stress plot (if ALT is dual-stress)



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Least Squares

**Example:** Let's demonstrate this tool on the first set of data we set up: That is the twelve component data where three failed at 150 °C and nine was failed at 200 °C (**data\_RAMs.Pt1.Ex1**). If we are given a use level stress of 80 °C (353 K), find the use level life.

### Physics-of-Failure Note:

- When temperature is the stress of note, an exponential-based life-stress model is often the best choice due to the exponential effect witnessed in live data. The RMT has Exponential, Arrhenius, and Eyring as exponential-based options.



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Least Squares

- On Slide 48 we saw two versions of the Exponential life-stress model. RMT has both the direct stress and inverse stress notation defined as "Exponential" and "Exponential2" respectively.

Enter:

- "Exponential"  $l(S) = b \exp(aS)$

- "Exponential2"  $l(S) = b \exp\left(\frac{a}{S}\right)$

- In ALT, inverse or reciprocal temperature is always applied for modeling as it best defines the life stress behavior between temperature and operational life
- Now we enter:

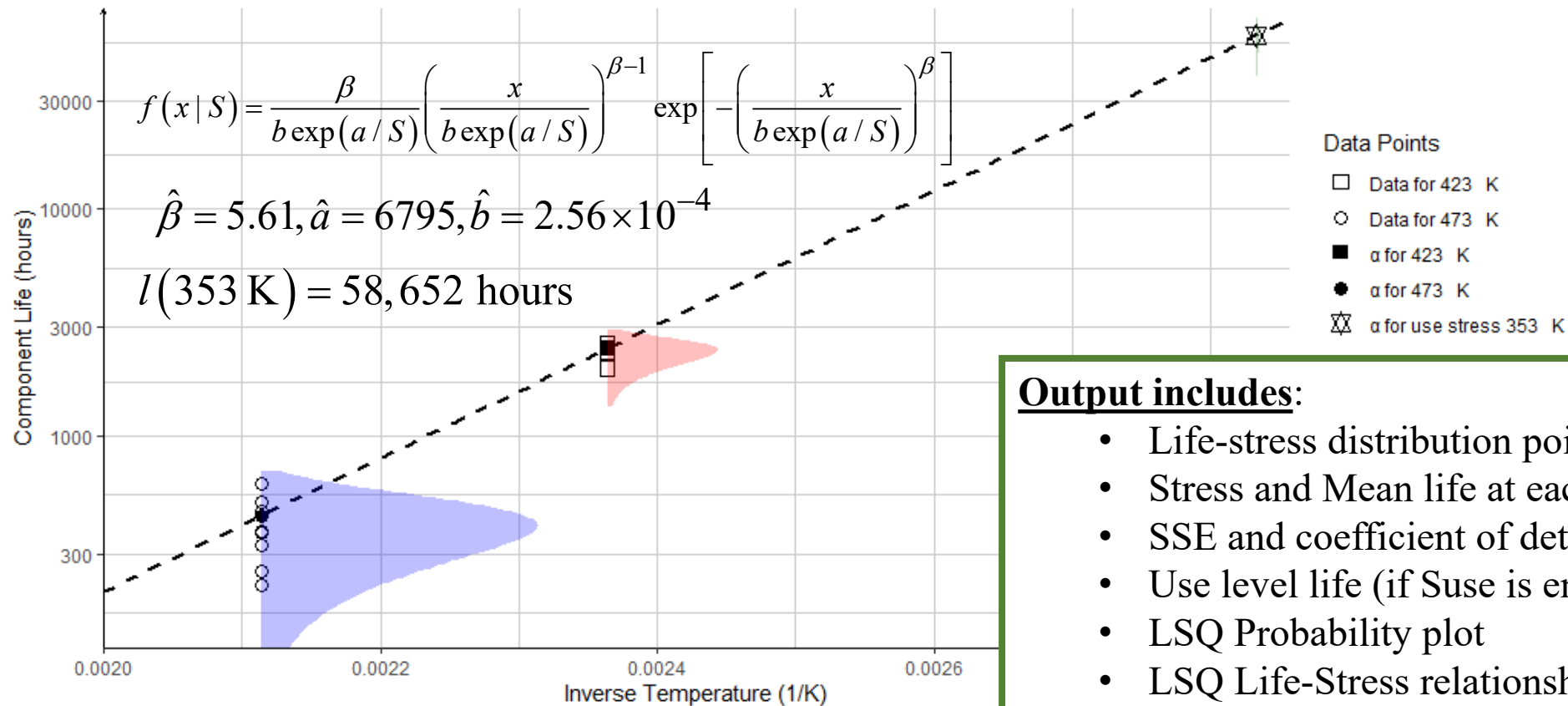
Input:

```
lifestress.LSQest(data_RAMs.Pt1.Ex1,ls="Exponential2", dist = "Weibull", pp = "Blom",  
                  xlabel1 = "Time to Failure (hours)",Suse = 353,  
                  Llab = "Component Life (hours)",Slab = "Inverse Temperature (1/K)",  
                  stressunit1 = "K")
```



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Least Squares



### Output includes:

- Life-stress distribution point estimates
- Stress and Mean life at each
- SSE and coefficient of determination  $R^2$
- Use level life (if Suse is entered)
- LSQ Probability plot
- LSQ Life-Stress relationship plot

# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE

### Maximum Likelihood Life-Stress Estimator Tool

```
lifestress.MLEest(data,ls,dist,confid,sided,xlabel1,Suse,Llab,Slab,Slab2, ..., stressunit1. stressunit2)
```

- The MLE ALT tool carries all of the same input as the LSQ ALT tool with the addition of some MLE input

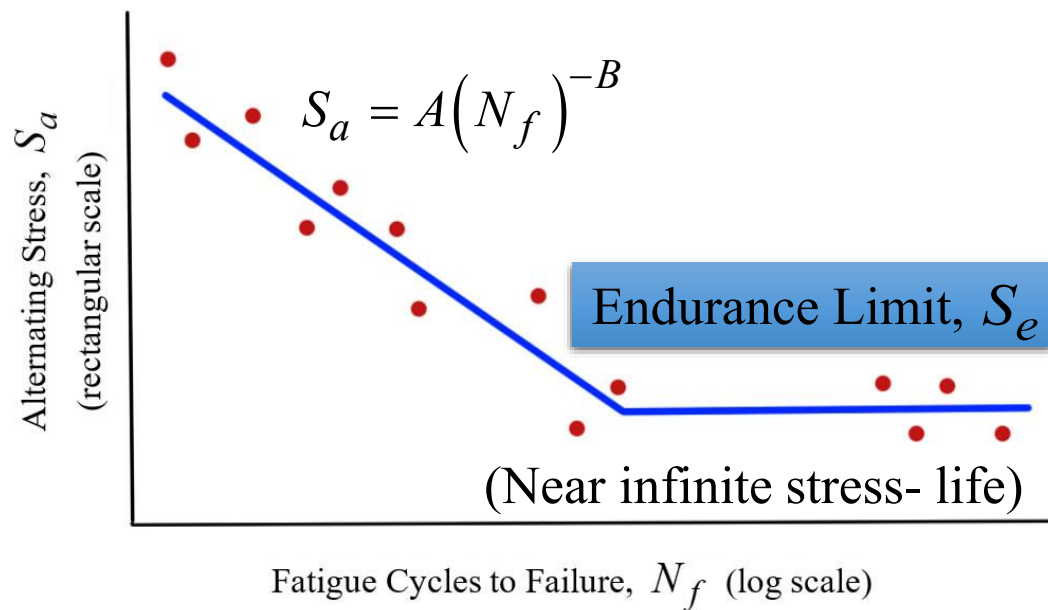
**Example:** Try this tool on the previous example. What use life do you get?



# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE

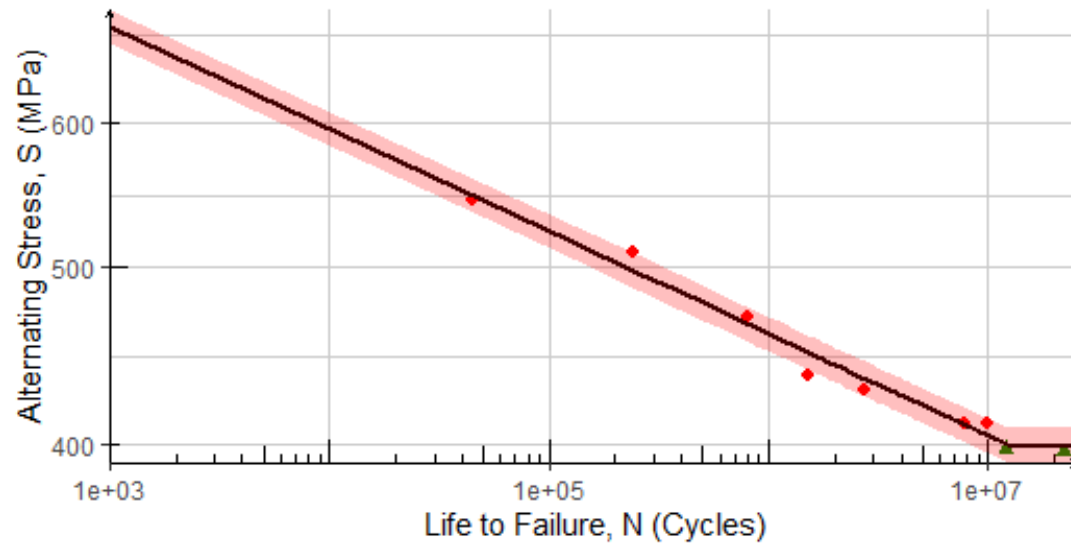
### Physics-of-Failure Note:



- **Definition:** In fracture mechanics, fatigue life to failure can be associated with an **S-N diagram** where failure is expected to occur in the higher stress region vs. the endurance limit region where infinite or near infinite life is expected.
- The RMT has a tool dedicated to generating S-N curves

# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE



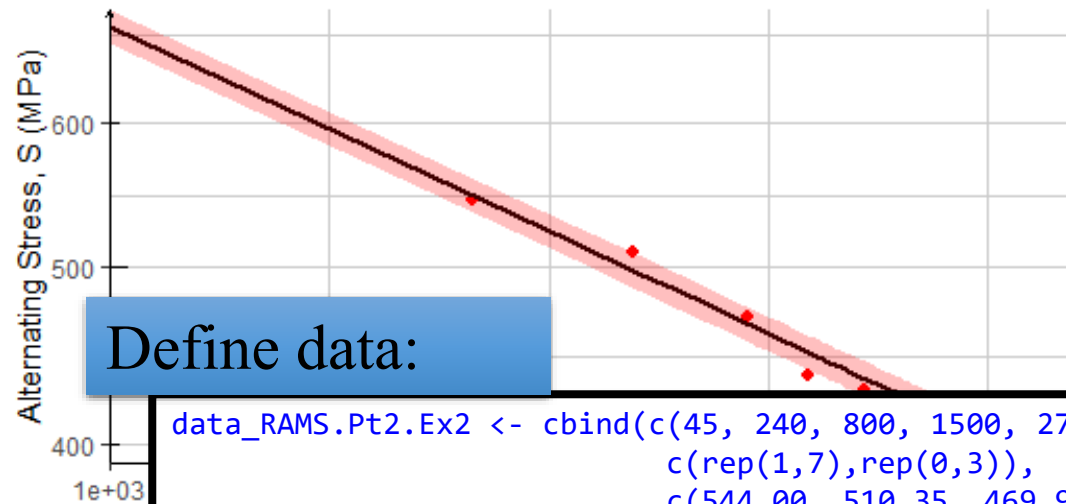
**Example:** Apply the S-N life model to the following data, a set of structural material tests yielded seven failures and three survivors. Use the RMT to determine the ALT parameters by MLE and 90% confidence and the use level life if 379 MPa is the use level stress.

(Note: ‘+’ stands for right-censored data)

Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$	Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$
544.00	45,000	411.00	7,800,000
510.35	240,000	411.00	10,000,000
469.95	800,000	404.24	26,000,000+
436.23	1,500,000	397.48	12,000,000+
427.82	2,700,000	395.83	22,000,000+

# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE



- First enter the data for use in the RMT probability plotting and/or ALT tools

Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$	Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$
544.00	45,000	411.00	7,800,000
510.35	240,000	411.00	10,000,000
469.95	800,000	404.24	26,000,000+
436.23	1,500,000	397.48	12,000,000+
427.82	2,700,000	395.83	22,000,000+



# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE

- Next we'll want to find an appropriate life-stress/distribution model to represent the data. ALT requires some assessment on what life-stress/distribution model to go with (*we can't just pick one*).
- Fortunately, we are given the S-N model for the life-stress component. However we need it in the form of one of the given models.
- This fits InversePower2! Which leaves the distribution to select.
- Unfortunately...

S-N Relation converts as follows:

➤ Step 1:  $S_a = A(N_f)^{-B} \Rightarrow N_f^{-B} = \frac{S_a}{A}$

➤ Step 2:  $N_f^{-B} = \frac{S_a}{A} \Rightarrow N_f = \frac{S_a^{-1/B}}{A^{-1/B}}$

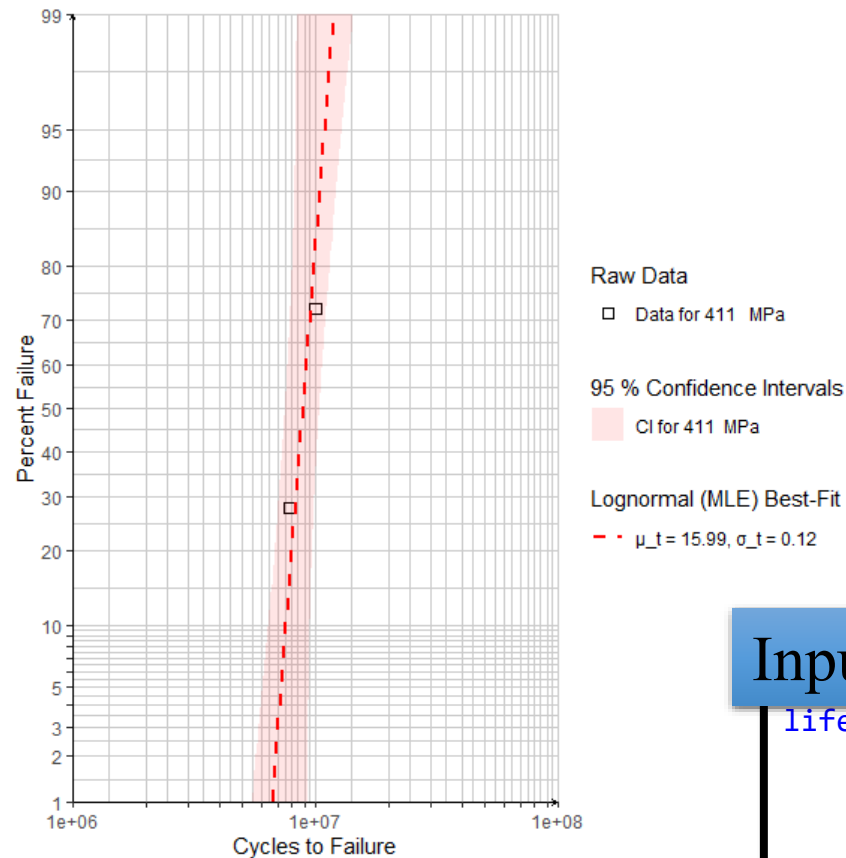
$$N_f = \frac{1}{A^{-1/B} \times S_a^{1/B}} \therefore l(S) = \frac{1}{bS^a}$$

where,  $b = A^{-1/B}$  and  $a = 1 / B$



# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE



- ... the probability plotting tools alone won't help in this case so we use the MLE ALT tool to go through our distribution options (choose between *lognormal* and *Weibull* life distributions).
- Start with the lognormal distribution option.

Input:

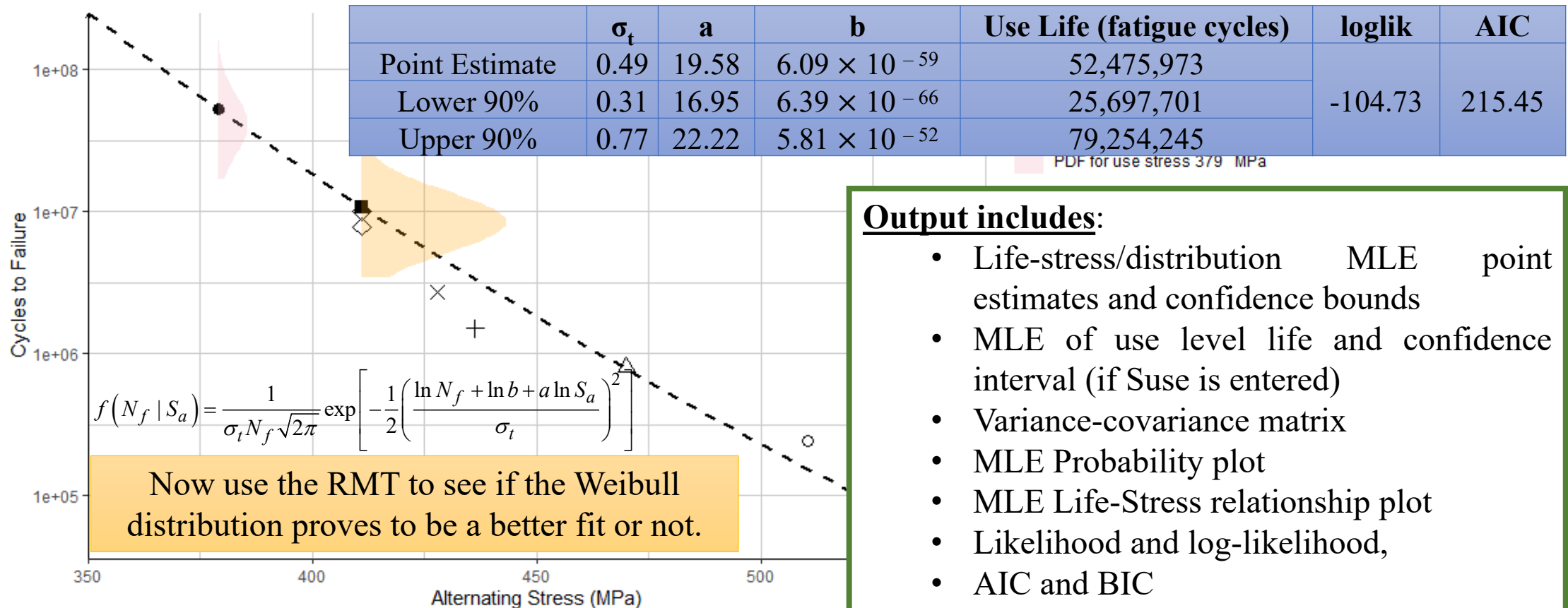
```
lifestress.MLEest(data_RAMs.Pt2.Ex2,ls="InversePower2", dist = "Lognormal",  
confid = 0.9, xlabel1 = "Cycles to Failure", Suse=379,  
Llab = "Cycles to Failure", Slab = "Alternating Stress (MPa)",  
stressunit1 = "MPa")
```



# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE

	A	B
Point Estimate	939.45	0.051
Lower 90%	844.39	0.045
Upper 90%	1034.52	0.059



# Accelerated Life Testing Tools:

## ALT Analysis Tools – MLE

- Lognormal

	$\sigma_t$	a	b	Use Life (fatigue cycles)	loglik	AIC
Point Estimate	0.49	19.58	$6.09 \times 10^{-59}$	52,475,973	<b><u>-104.73</u></b>	<b><u>215.45</u></b>
Lower 90%	0.31	16.95	$6.39 \times 10^{-66}$	25,697,701		
Upper 90%	0.77	22.22	$5.81 \times 10^{-52}$	79,254,245		

- Weibull

	$\beta$	a	b	Use Life (fatigue cycles)	loglik	AIC
Point Estimate	2.36	20.25	$8.13 \times 10^{-61}$	73,124,922	-105.20	216.41
Lower 90%	1.45	18.03	$9.70 \times 10^{-67}$	37,403,761		
Upper 90%	3.85	22.48	$6.81 \times 10^{-55}$	108,846,082		

- By AIC as well as likelihood, we go with the *Lognormal-Inverse-Power model*

# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation

### Bayesian Life-Stress Estimator Tool

```
lifestress.BAYEStest(pt_est,ls,dist,TTF,SF,Tc,Sc,SUSE,SACC,confid,priors,nsamples,burnin,nchains)
```

- “*TTF*” – vector of failure or primary event data in a given set
- “*Tc*” – vector right censored data of a given set
- “*SF*” – stress vector (or list of vectors) corresponding with failure data “*TTF*”
- “*Sc*” – stress vector (or list of vectors) corresponding with right censored data “*Tc*”
- “*SUSE*” – nominal or use level stress
- “*SACC*” – an accelerated stress used to derive posterior accelerated factor



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation

**Example:** A new component series has been developed using the material from the last example. Accelerated tests were performed on six such components where again fatigue was the failure mechanism and the following data was obtained. Use the RMT to update the model parameters using the MLE of the material results as a prior.

(Note: ‘+’ stands for right-censored data)

Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$	Stress Amplitude, $S_a$ (MPa)	Fully Reversed Cycles, $N_f$
404.72	1,400,000	394.38	10,000,000+
410.93	2,900,000	381.28	10,000,000+
387.49	9,000,000	310.95	10,000,000+



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation

- As before, start with defining the priors. We can base this from the upper and lower CI for each parameter to compute the log mean and log standard deviation assuming all three priors fit a lognormal distribution.

For $\sigma_t$	$\mu_{t,\sigma_t} = \frac{\log \sigma_{t,5\%} + \log \sigma_{t,95\%}}{2}$	$\sigma_{t,\sigma_t} = \frac{0.5(\log \sigma_{t,95\%} - \log \sigma_{t,5\%})}{Z_{95\%}}$
For $\mathbf{a}$	$\mu_{t,a} = \frac{\log a_{5\%} + \log a_{95\%}}{2}$	$\sigma_{t,a} = \frac{0.5(\log a_{95\%} - \log a_{5\%})}{Z_{95\%}}$
For $\mathbf{b}$	$\mu_{t,b} = \frac{\log b_{5\%} + \log b_{95\%}}{2}$	$\sigma_{t,b} = \frac{0.5(\log b_{95\%} - \log b_{5\%})}{Z_{95\%}}$

Define:

```
priorset <- c("lognormal(-0.715395,0.358001)",
              "lognormal(2.9654,0.10560)",
              "lognormal(-134.0451,8.199294)")
```



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation

- Now enter the input into **lifestress.BAYESest**. Remember, we only need to enter the data as vectors for Bayesian operations.

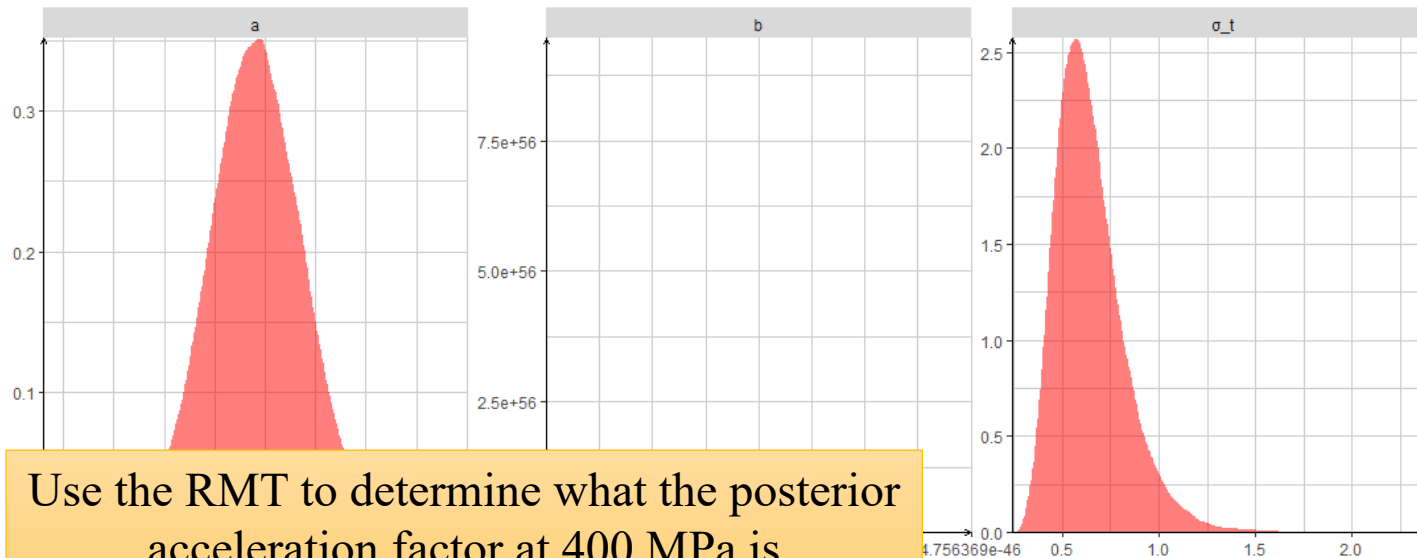
Input:

```
lifestress.BAYESest(c(1,20,0.0001),ls="InversePower2",dist = "Lognormal",  
  TTF = c(2900000, 1400000, 9000000),SF = c(410.93, 404.72, 387.49),  
  Tc = c(10000000, 10000000, 10000000), Sc = c(394.38, 381.28, 310.95),  
  SUSE = 379, confid = 0.9, priors = priorset,  
  nsamples = 30000, burnin = 1000, nchains = 4)
```



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation



### Output includes:

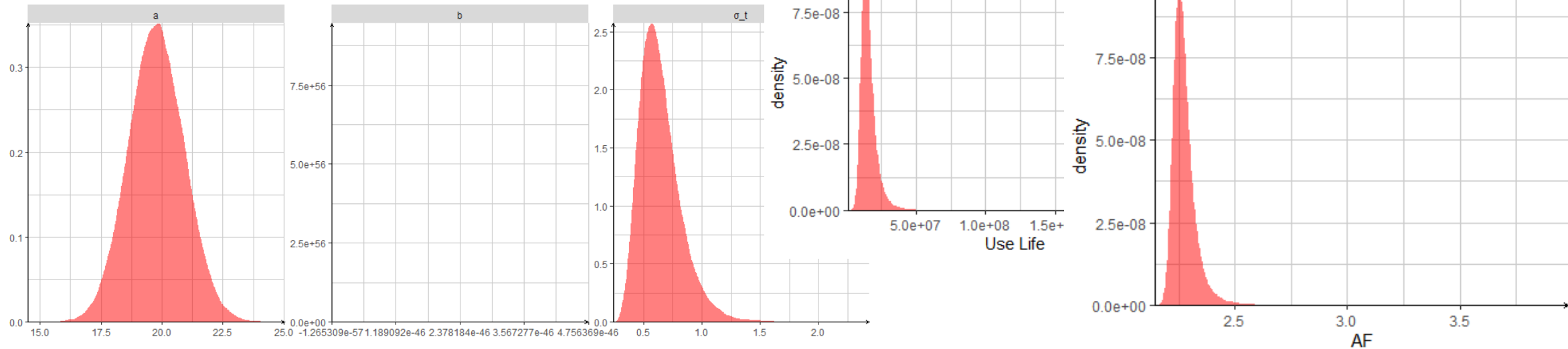
- Complete posterior MC chains per parameter
- Table of posterior stats
- Plots of posterior MCMC trace, histogram, and density
- Use life posterior (if SUSE is given)
- Acceleration Factor posterior (if SUSE and SACC are given)

Posterior	Mean	SDev	Lower 90%	Median	Upper 90%	Rhat
$\hat{\sigma}_t$	0.64	0.17	0.41	0.62	0.97	1.00
$\hat{a}$	19.77	1.15	17.90	19.77	21.68	1.00
$\hat{b}$	$1.52 \times 10^{-48}$	$3.69 \times 10^{-46}$	$7.69 \times 10^{-64}$	$7.06 \times 10^{-59}$	$5.13 \times 10^{-54}$	1.00
Use Level Life (cycles)	$1.57 \times 10^7$	$6.00 \times 10^6$	$8.96 \times 10^6$	$1.46 \times 10^7$	$2.63 \times 10^7$	1.00



# Accelerated Life Testing Tools:

## ALT Analysis Tools – Bayesian Estimation



Posterior	Mean	SDev	Lower 90%	Median	Upper 90%	Rhat
$\hat{\sigma}_t$	0.64	0.17	0.41	0.62	0.97	1.00
$\hat{a}$	19.77	1.15	17.90	19.77	21.68	1.00
$\hat{b}$	$1.52 \times 10^{-48}$	$3.69 \times 10^{-46}$	$7.69 \times 10^{-64}$	$7.06 \times 10^{-59}$	$5.13 \times 10^{-54}$	1.00
Use Level Life (cycles)	$1.57 \times 10^7$	$6.00 \times 10^6$	$8.96 \times 10^6$	$1.46 \times 10^7$	$2.63 \times 10^7$	1.00
AF at 400 MPa	2.91	0.18	2.63	2.90	3.22	1.00

# Accelerated Life Testing Tools:

## ALT Analysis Tools – Practice with RMT

**PROBLEM STATEMENT:** Consider an accelerated test of a component with a linear stress-life relationship.

Stress Units	Failure Times (hours) ('+' stands for right-censored data)					
100	245	312	409	500+	500+	500+
200	110	180	200	222	250+	250+
300	50	70	88	112	140	160

We are interested in developing a pdf of time-to-failure conditioned on the level of stress using the Bayesian approach. In teams do the following:

- Use the probability plot tools to determine the best life distribution for the data
- Use the Bayesian ALT tool to determine the time-to-failure distribution if the use stress is 20 units. Assume we know the prior for parameter  $a$  is uniform(-2.5, -0.5) and for parameter  $b$  uniform(500, 850). The third will be provided based on your choice for part (a).



# CHECKPOINT

- ✓ Introduced concepts of accelerated life testing and probabilistic physics of failure and applications in the RMT
- ✓ Worked through a MLE and Bayesian ALT operation involving physics-of-failure
- ✓ Compared the fitness of one life-stress/distribution model over another by comparing likelihood and/or AIC
- ✓ Ran ALT Bayes tool to determine a posterior acceleration factor at a given accelerated stress

# RMT

RELIABILITY MODELING TOOLKIT

NEXT:

Accelerated Degradation Testing



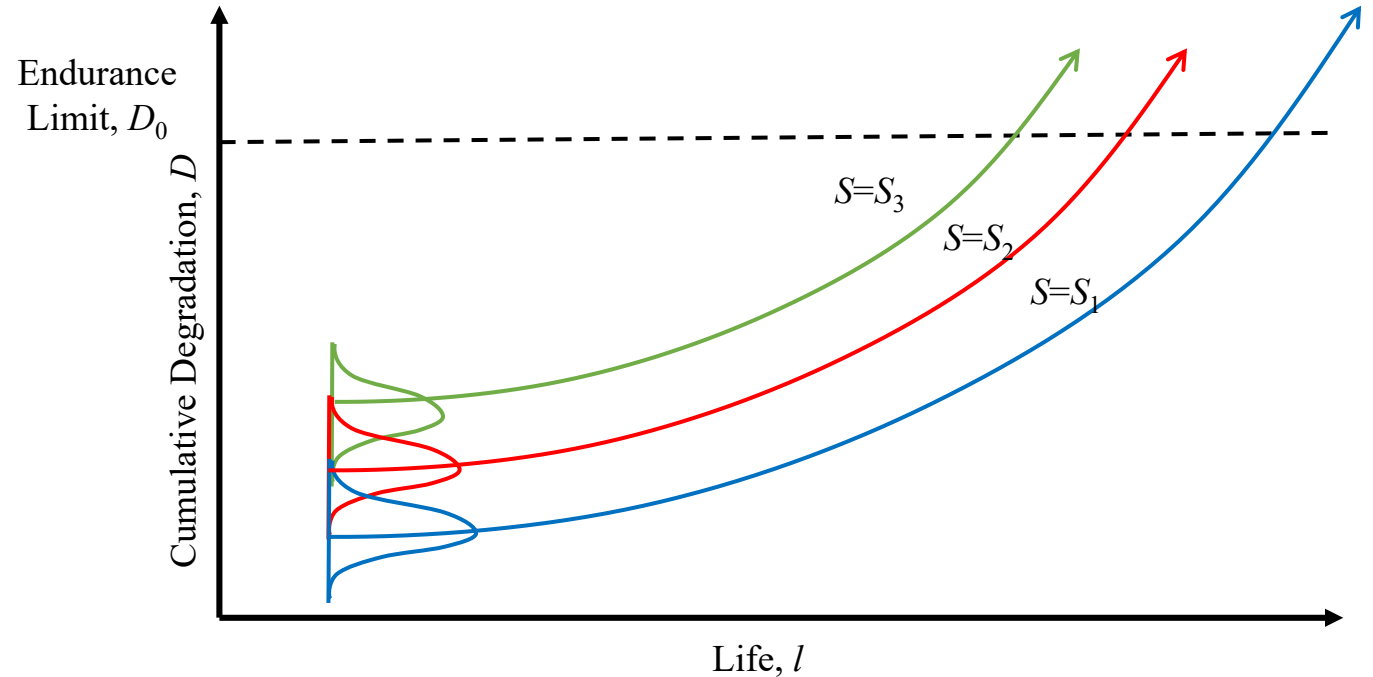
# Overview of Accelerated Degradation Testing: Degradation-Life Relationship

- In accelerated testing, sometimes a non-destructive data gathering strategy is preferred
- **Definition: Accelerated Degradation Testing (ADT)** is such a non-destructive methodology where life is related to a form of degradation (e.g. crack propagation, wear, creep, corrosion, entropy). Life may be then defined as where degradation reaches a known endurance limit  $D_0$ .



# Overview of Accelerated Degradation Testing: Degradation-Life Relationship

- Like ALT, ADT may be conducted at various accelerated stress levels to hasten degradation, but then used to determine the use level life based on both use level stress and the endurance limit.



- The RMT currently has eight degradation-life  $D(l)$  models used for its ADT tools.

# Overview of Accelerated Degradation Testing: Degradation-Life Relationship

Damage-Life Model	Damage-Life function, $D(l)$	Damage-Life Model	Damage-Life function, $D(l)$
Linear	$D(l) = a + bl$	Power	$D(l) = bl^a$
Exponential	$D(l) = b \exp(al)$	Logarithmic	$D(l) = a + b \ln(l)$
Square-Root	$D(l) = (a + bl)^2$	Lloyd-Lipow	$D(l) = a - \frac{b}{l}$
Square-Root .v2	$D(l) = a + b\sqrt{l}$	Mitsuom	$D(l) = \frac{1}{1 + bl^a}$

- $a, b$  – model parameters



# Accelerated Degradation Testing Tools: Degradation-Life Model Selection and Ranking

## Accelerated Degradation Testing Rank System Tool

```
adt.rank(dat)
```

- “*dat*” – Tabular degradation data
- Provides a unit-by-unit rank check of all relevant models as well as an average rank



# Accelerated Degradation Testing Tools: Degradation-Life Model Selection and Ranking

**Example:** The following represents resistance buildup data for a series of five prototype switches run at a constant 5 volts.

Test Duration (Hours)	Resistance (Ohms, $\Omega$ )				
	Switch A	Switch B	Switch C	Switch D	Switch E
10	240.7	255.5	293.3	268.7	295.5
20	316.2	297.9	316.1	305.0	348.1
30	380.7	392.4	343.7	338.2	362.0
40	409.1	436.0	397.5	359.5	400.9
50	419.2	459.8	429.2	442.5	445.3
60	455.8	472.8	444.0	464.6	455.3
70	475.8	479.8	449.3	468.9	470.2

Use the RMT to determine which of its degradation models is the best fit for resistance buildup.

- **NOTE:** Unlike previous examples, the data entry for ADT tools is a bit different...



# Accelerated Degradation Testing Tools: Degradation-Life Model Selection and Ranking

- Different entries for columns

- **Column 1** – Time data
- **Column 2** – Degradation Data
- **Column 3** – Unit or Sample Designation or Name
- **Column 4 and up** – Stress levels for Columns 1 through 3

Test Duration (Hours)	Resistance (Ohms, $\Omega$ )				
	Switch A	Switch B	Switch C	Switch D	Switch E
10	240.7	255.5	293.3	268.7	295.5
20	316.2	297.9	316.1	305.0	348.1
30	380.7	392.4	343.7	338.2	362.0
40	409.1	436.0	397.5	359.5	400.9
50	419.2	459.8	429.2	442.5	445.3
60	455.8	472.8	444.0	464.6	455.3
70	475.8	479.8	449.3	468.9	470.2



# Accelerated Degradation Testing Tools: Degradation-Life Model Selection and Ranking

- Manual entry requires binding of data frame

## Define data:

```
data_RAMSPt3.Ex1 <- cbind(data.frame(rep(c(10, 20, 30, 40, 50, 60, 70),5),  
                                     c(240.7, 316.2, 380.7, 409.1, 419.2, 455.8, 475.8,  
                                     255.5, 297.9, 392.4, 436, 459.8, 472.8, 479.8,  
                                     293.3, 316.1, 343.7, 397.5, 429.2, 444, 449.3,  
                                     268.7, 305, 338.2, 359.5, 442.5, 464.6, 468.9,  
                                     295.5, 348.1, 362, 400.9, 445.3, 455.3, 470.2),  
                                     c(rep("Switch A",7),rep("Switch B",7),  
                                     rep("Switch C",7), rep("Switch D",7),  
                                     rep("Switch E",7))),  
                                     rep(5,35)))
```

Single or double quotes  
around unit names

- This gets tedious for larger data with multiple units so...

Download this data set as a CSV from  
<https://raw.githubusercontent.com/Center-for-Risk-and-Reliability/RMT/main/CSVExampleData/>



# Accelerated Degradation Testing Tools: Degradation-Life Model Selection and Ranking

- The read.csv command feeds CSV data into R

Alt. Define data:

Feed the downloaded CSV to load the data

```
data_RAMs.Pt3.Ex1 <- read.csv("~/Download/Degradation_Data_Session3_Example_1.csv")
```

- Now run **adt.rank** in to get the rank.

Input:

```
adt.rank(data_RAMs.Pt3.Ex1)
```

- We get **square-root.v2**

$$D(l) = a + b\sqrt{l}$$

## Output includes:

- Unit based SSE for each degradation life model
- ADT ranking by unit
- Average ADT ranking



# Accelerated Degradation Testing Tools: ADT Analysis Tools – Least Squares

## Least-Squares Accelerated Degradation Testing Estimator Tool

```
degradationlife.LSQest(dat,dl,dist,pp,D0,modelstress,xlabel,ylabel,Suse)
```

- “*dat*” – Tabular degradation data
- “*dl*” – degradation-life model
- “*dist*” – degradation distribution (“Normal” or “Lognormal”)
- “*D0*” – given endurance limit
- “*modelstress*” – Optional stress-parameter model to relate degradation at different stress levels



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

**Example:** Use the ADT tool to plot the previous data and fit line to the **square-root v2. model** (**d1="SquareRoot2"**) when it is given that the endurance limit is 550  $\Omega$ .

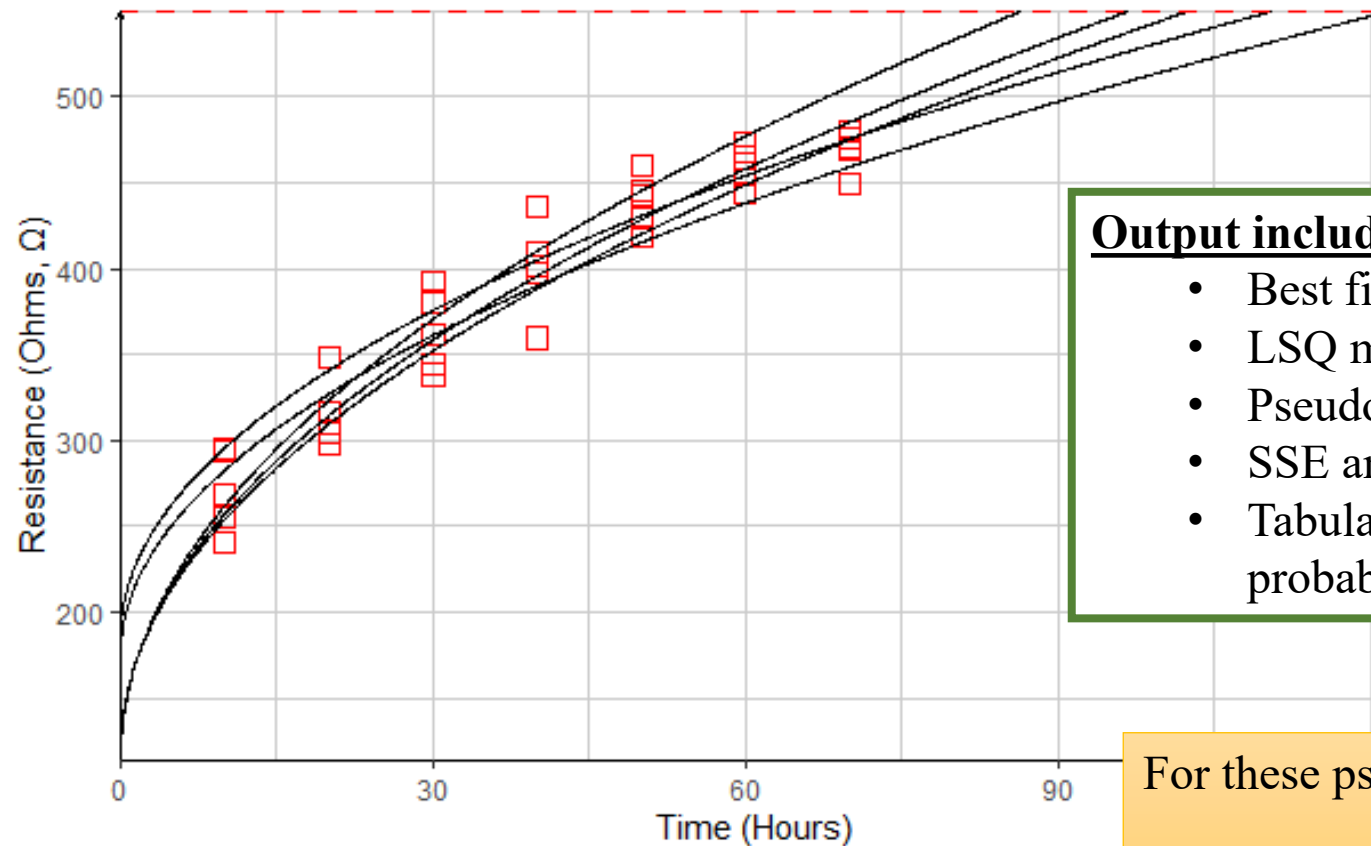
- Enter the following:

Input:

```
degradationlife.LSQest(data=data_RAMs.Pt3.Ex1,d1="SquareRoot2",dist="Normal",  
                        pp="Blom",D0=550,  
                        xlabel = "Time (Hours)",ylabel = "Resistance (Ohms,  $\Omega$ )")
```



# Accelerated Degradation Testing Tools: ADT Analysis Tools – Least Squares



## Output includes:

- Best fit lines for each of the unit
- LSQ model parameters for each unit
- Pseudo-failure times for each unit
- SSE and coefficient of determination  $R^2$
- Tabular time-to-failure data for use in probability plotting or ALT tools

For these pseudo-failure times, find which distribution fits the best

# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

- **Definition**: For lack of a physical failure time, a **pseudo-failure time** is simply the extrapolated time of failure based on the LSQ estimates for degradation-life.

- **QUESTION**: When and how do you use the **modelstress** input?

```
degradationlife.LSQest(dat,dl,dist,pp,D0,modelstress,xlabel,ylabel,Suse)
```

- **ANSWER**: Ideally when you want to apply stress to your degradation-life model to make a degradation-stress-life model.



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

- The RMT has one form for **modelstress** active with another to be added later this year or by RAMS 2027.

**Example:** For the linear degradation-life model  $D(l) = a + bl$

- 1) **Parameter-stress modelstress (NEW)** – Replaces one model parameter (usually the intercept parameter) while the other is represented as a normal distribution.

$$D(l, S) = a(S) + \text{NOR}(\mu_b, \sigma_b)l$$

- 2) **Acceleration Factor-stress modelstress (In development)** – The acceleration factor (of a given life-stress type) is multiplied by the life in the degradation-life equation

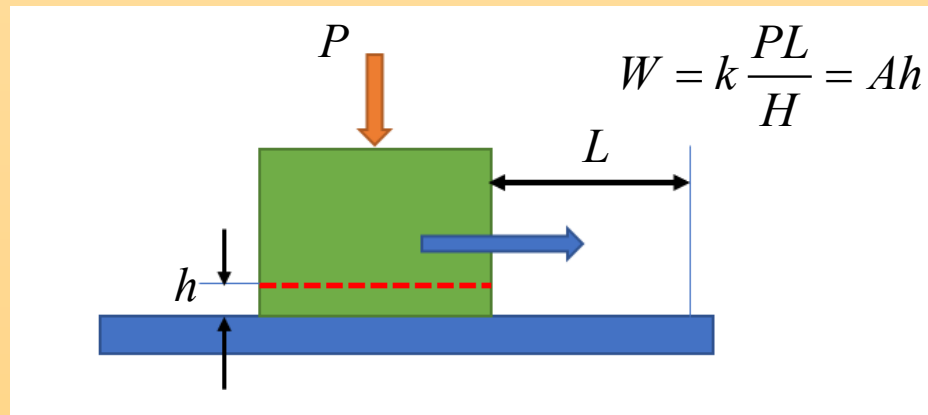
$$D(l, S) = a + b(l \times AF(S))$$



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

### Physics-of-Failure Note:



- $k$  – wear coefficient
- $A$  – constant area of contact between surfaces
- $h$  – wear depth
- $P$  – contact force
- $L$  – sliding distance
- $H$  – material hardness

- **Definition:** The failure mechanism **wear**  $W$  is defined as a volumetric measure of damage produced by repeated contact between two solid surfaces. The material lost is the measure of that damage.

- *Sliding wear is most common*

# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

**Example:** The following is the results of a sliding wear test on a particular metal alloy.

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10	Unit 11	Unit 12
Cycles (x100)	Mass lost from wear ( $\mu\text{g}$ )											
2	3.2	2.7	2.1	2.6	7.5	7.5	7	7.8	12.5	11	13	11.7
5	4.1	3.4	2.7	3.5	7.8	8.1	8.9	8.9	15.4	13.9	15.1	13.7
10	4.5	3.8	3.1	4	8.2	9.8	9.4	10	17.2	16.1	18.6	16.7
20	4.7	3.9	3.3	4	10.6	10.9	11.1	11.5	20.5	18.6	20.2	17.5
50	5.8	5.4	4	5.2	12.6	14.8	12.4	13.7	24.1	22.2	23.9	22.3
100	6.8	5.7	4.6	6.1	13.3	16.1	13.5	16.2	27	27.8	29.7	25.3
200	7.7	6.3	5.7	6.7	12.9	17.3	16.7	16.2	29.4	31	31.5	32
500	9.6	8.4	6.6	8.5	14.8	20.2	17.3	21	37.9	36.6	39.6	38.2
Applied Weight (g)	10	10	10	10	50	50	50	50	100	100	100	100

The test was conducted with three different applied weights in order to study the effect of wear as a failure mechanism.



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Least Squares

- First we download the data.

### Define:

```
data_RAMSPt3.Ex2 <- read.csv("https://raw.githubusercontent.com/Center-for-Risk-and-Reliability/RMT/main/CSVExampleData/Degradation_Data_1_Mass_Loss_by_Weight_gms_Example_5_2.csv")
```

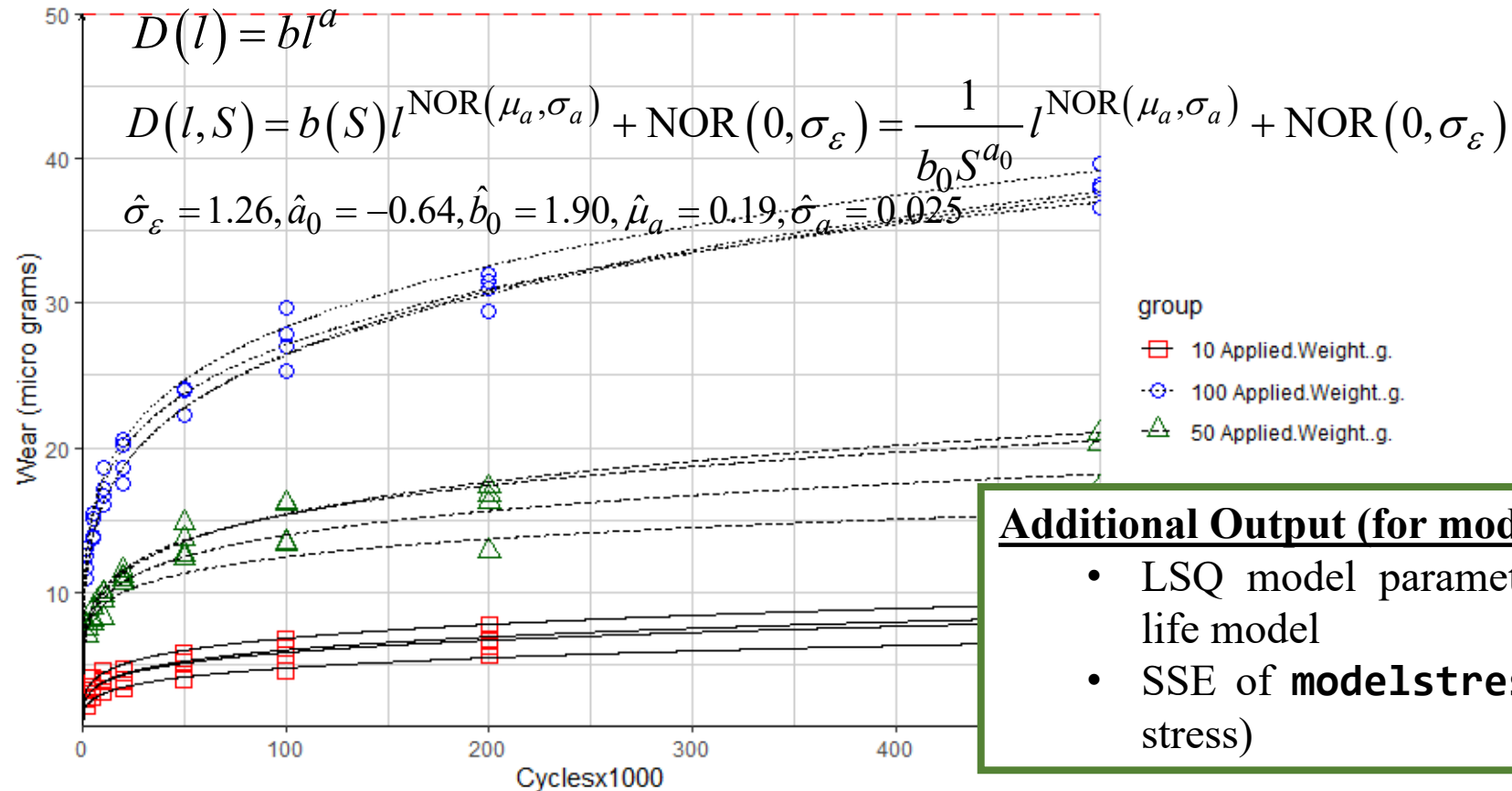
- Then we'll demonstrate the RMT using the Power degradation-life model. Try entering an Inverse Power .v2 parameter-stress modelstress value (that is `modelstress="InversePower2"`). Also set the endurance limit to 50 micrograms.

### Input:

```
degradationlife.LSQest(data=data_RAMSPt3.Ex2,d1="Power",dist="Normal",  
                      pp="Blom",D0=50, modelstress="InversePower2",  
                      xlabel = "Cyclesx100",ylabel = "Wear (micro grams)")
```



# Accelerated Degradation Testing Tools: ADT Analysis Tools – Least Squares



## Additional Output (for modelstress) includes:

- LSQ model parameters for degradation-stress-life model
- SSE of **modelstress** selection (parameter-to-stress)

# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – MLE

- ADT MLE generally follows a random parameter degradation likelihood where, it accounts for  $n$  units of degradation with  $m_i$  readings for each unit  $i$ ,

$$\ell = \prod_{i=1}^n \int \prod_{j=1}^{m_i} \frac{\phi(z_{ij})}{\sigma_\varepsilon} f(\bar{\Theta} | \mu_{\bar{\Theta}}, \Sigma_{\bar{\Theta}}) d\bar{\Theta} \text{ where } z_{ij} = \frac{y_{ij} - D(l_{ij} | \bar{\Theta})}{\sigma_\varepsilon}$$

where,

- $f(\bar{\Theta} | \mu_{\bar{\Theta}}, \Sigma_{\bar{\Theta}})$  – joint distribution (such as a multivariate normal distribution)
- $\bar{\Theta}$  – damage-life model parameters as a vector
- $\mu_{\bar{\Theta}}$  – mean vector for parameters  $\bar{\Theta}$
- $\Sigma_{\bar{\Theta}}$  – variance-covariance matrix for parameters  $\bar{\Theta}$



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – MLE

### Maximum Likelihood Degradation Life Estimator Tool

```
degradationlife.MLEest(dat,dl,dist,pp,D0,modelstress,Suse,confid,sided,xlabel,ylabel,...,stressunit1,stressunit2)
```

- No new input for this tool. Like the ALT MLE tool, there's no need to separate the original data block.

**Example:** Use the RMT ADT tool to compute the model parameters for the sliding wear data (last example) at 90% confidence and the use life at 5 grams of application

- Use much of the same input for entry and execution of this tool.

Input:

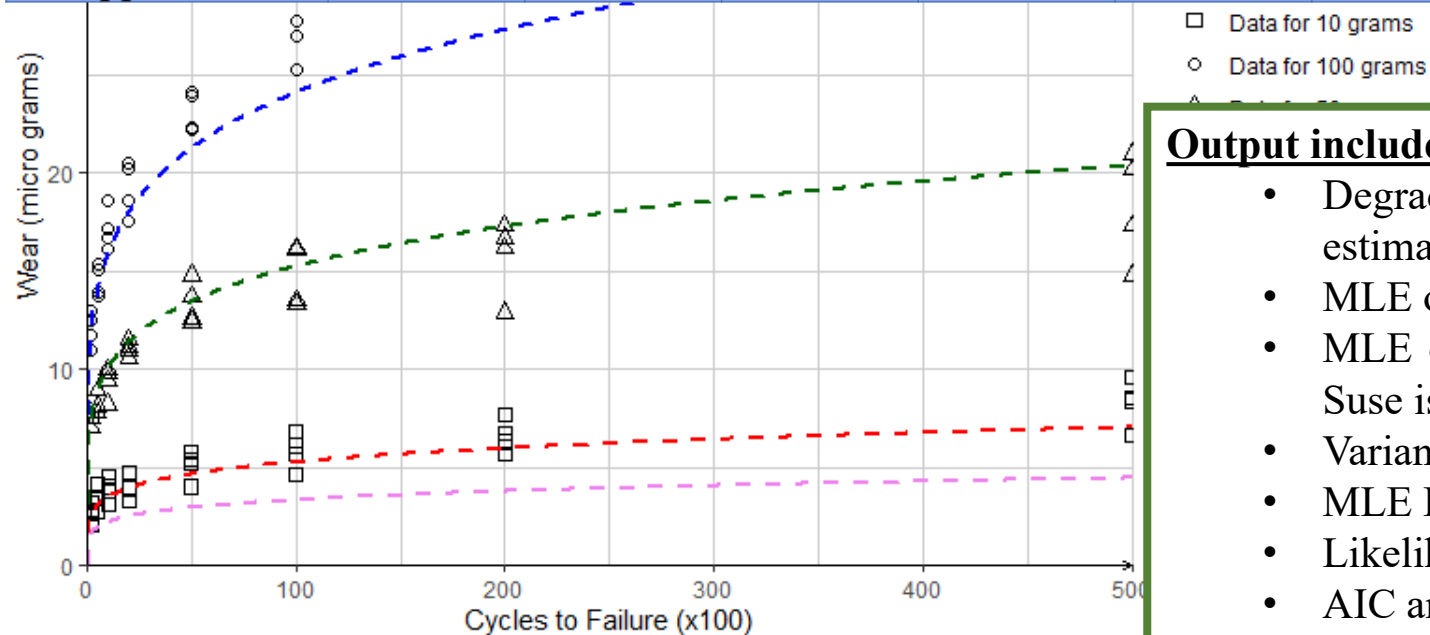
```
degradationlife.MLEest(data=data_RAMs.Pt3.Ex2,dl="Power",dist="Normal",pp="Blom",D0=50,  
    modelstress="InversePower2", confid=0.90, Suse=5,  
    xlabel="Cycles to Failure (x1000)",ylabel="Wear (micro grams)",  
    stressunit1 = "grams")
```



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – MLE

	$\sigma_\varepsilon$	$a_0$	$b_0$	$\mu_a$	$\sigma_a$	Use Life (fatigue cycles x100)	loglik	AIC
Point Estimate	0.63	-0.66	2.01	0.188	0.021	$335 \times 10^6$	-85.61	181.22
Lower 90%	0.52	-0.68	1.82	0.182	0.018	$188 \times 10^6$		
Upper 90%	0.78	-0.64	2.22	0.194	0.024	$627 \times 10^6$		



### Output includes:

- Degradation-Stress-Life/distribution MLE point estimates and confidence bounds
- MLE of stress level life and confidence interval
- MLE of use level life and confidence interval (if Suse is entered)
- Variance-covariance matrix
- MLE Degradation-Stress-Life plot
- Likelihood and log-likelihood
- AIC and BIC

# Accelerated Degradation Testing Tools: ADT Analysis Tools – Bayesian Estimation

## Bayesian Degradation Life Estimator Tool

```
degradationlife.BAYESest(pt_est,dat,dl,dist,D0,modelstress,confid,SUSE,priors,nsamples,burnin,nchains)
```

- Again, nothing new for input. Unlike the other Bayesian tools however, this one takes the data as previously set up for the LSQ and MLE ADT tools.



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Bayesian Estimation

**Example:** Let's say that we get additional wear data from the following: A new device whose failure mechanism is sliding wear subjected on a vital component. In a series of ADT's the wear of four devices' components were monitored and collected on the right.

We want to add this information to the previous MLE results. Use the ADT Bayes tool to apply test data from Devices A, B, C, and D to update the use life.

	Device A	Device B	Device C	Device D
Cycles (x100)	Mass lost from wear (µg)			
10	5.6	5.2	15	14.5
20	5.7	5.5	15.6	14.7
30	6.3	5.7	16.2	15.6
40	6.7	6	16.8	17.4
50	7	6.4	17.5	19.5
100	10.2	9.3	22.3	21.5
1000	18.4	17.3	57.3	59.1
Applied Weight (g)	30	30	70	70

- As we only have four component data, let's manually enter this ADT data

# Accelerated Degradation Testing Tools: ADT Analysis Tools – Bayesian Estimation

## Define data:

```
data_RAMs.Pt3.Ex3 <- cbind(data.frame(c(rep(c(10,20,30,40,50,100,1000),4)),  
                                     c(5.6, 5.7, 6.3, 6.7, 7, 10.2, 18.4,  
                                         5.2, 5.5, 5.7, 6, 6.4, 9.3, 17.3,  
                                         15, 15.6, 16.2, 16.8, 17.5, 22.3, 57.3,  
                                         14.5, 14.7, 15.6, 17.4, 19.5, 21.5, 59.1),  
                                     c(rep("Device A",7),rep("Device B",7),  
                                       rep("Device C",7),rep("Device D",7)),  
                                     c(rep(30,14),rep(70,14))))
```

- Now let's define the prior for the parameters as follows

## Define:

```
priorset <- c("lognormal(-0.456903,0.1232223)",  
              "normal(-0.661706,0.0140145)", "lognormal(0.6961635,0.06053974)",  
              "lognormal(-1.672606,0.01887561)", "lognormal(-3.854416,0.08361823)")
```



# Accelerated Degradation Testing Tools: ADT Analysis Tools – Bayesian Estimation

- Finally, let's run this in the Bayes ADT tool

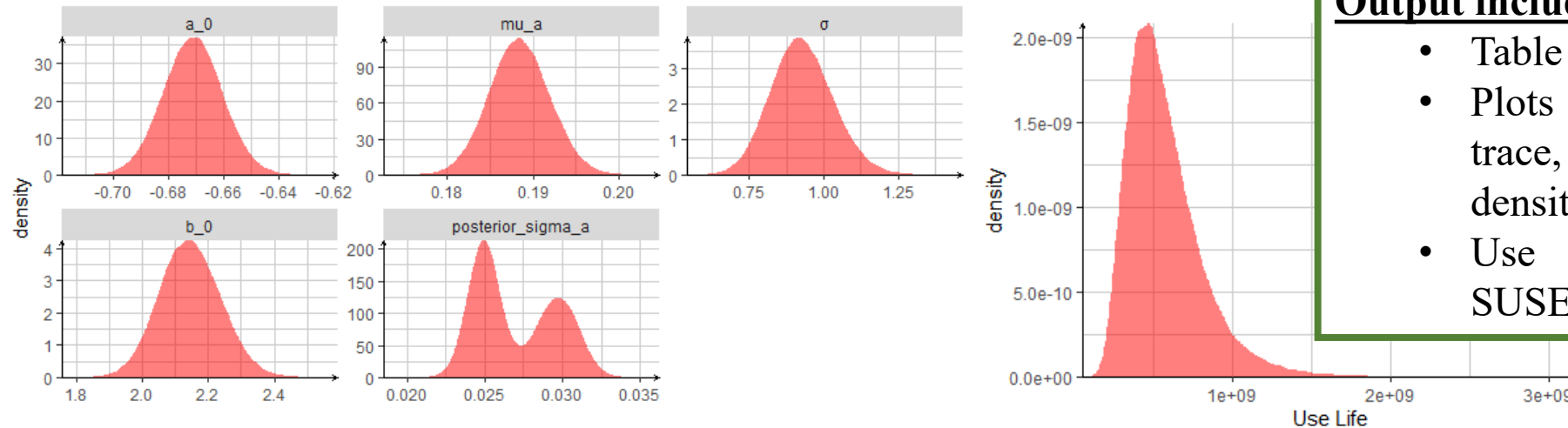
Input:

```
degradationlife.BAYEStest(pt_est = c(0.6,-0.1,2,0.1,0.01), data = data_RAMs.Pt3.Ex3,  
  dl = "Power",dist="Normal", D0=50,  
  modelstress="InversePower2",confid=0.95,  
  SUSE=5,priors = priorset,  
  nsamples = 30000, burnin = 1000, nchains = 4)
```



# Accelerated Degradation Testing Tools:

## ADT Analysis Tools – Bayesian Estimation



### Output includes:

- Table of posterior stats
- Plots of posterior MCMC trace, histogram, and density
- Use life posterior (if SUSE is given)

Posterior	Mean	SDev	Lower 90%	Median	Upper 90%	Rhat
$\hat{\sigma}_\varepsilon$	0.92	0.11	0.72	0.92	1.14	1.02
$\hat{a}_0$	-0.67	0.011	-0.69	-0.67	-0.65	1.00
$\hat{b}_0$	2.14	0.092	1.97	2.14	2.34	1.00
$\hat{\mu}_a$	0.19	0.0035	0.182	0.189	0.196	1.01
$\hat{\sigma}_a$	0.028	0.0026	0.023	0.029	0.032	1.46
Use Level Life (cycles)	$565 \times 10^6$	$238 \times 10^6$	$240 \times 10^6$	$520 \times 10^6$	$117 \times 10^6$	1.00

# CHECKPOINT

- ✓ Introduced concepts of accelerated degradation testing and applications in the RMT
- ✓ Ranked ADT data for fitness to degradation-life models
- ✓ Applied parameter-stress relation to create degradation-stress-life relation
- ✓ Worked through a MLE and Bayesian ADT operation involving sliding wear as our physics-of-failure

# RMT

RELIABILITY MODELING TOOLKIT

NEXT:

Wrap-Up



# Today We Covered...

- The RMT and its tool availability for reliability modeling, PPoF, ALT, and ADT applications
- LSQ and MLE estimation and Bayesian updating methods to perform several examples including a group workshop
- Application of PPoF to run analyses of ALT and ADT problems
- Interpretation of results by SSE, likelihood, and AIC metrics to choose the best distribution model for sample data and life-stress distribution model for an ALT or ADT scenario



# What's Next?

- The RMT is continuously undergoing development, and the development staff is always seeking to bring more tools into the package for education and engineering field applications
- Software reliability and debug procedures to be applied in updating current toolset and creating new tools
- Life-stress and degradation-stress-life model updating and additions with PPoF emphasis
- The Center for Risk and Reliability is seeking opportunities for test and test data driven development opportunities



# Closing Thoughts

- More information on the RMT is available on the Github site and R documentation
- Also feel free to contact the RMT developers for feedback and assistance

<https://github.com/Center-for-Risk-and-Reliability/RMT>



# Thank You!

Contact Information:

Developer: Reuel Smith

[smithrc@umd.edu](mailto:smithrc@umd.edu)

Head and Founder: Mohammad Modarres

[modarres@umd.edu](mailto:modarres@umd.edu)

Please stop by the UMD booth this week (M-W) starting tomorrow at 3PM in the RAMS Exhibit Hall!



# RMT

RELIABILITY MODELING TOOLKIT



CENTER FOR  
RISK AND RELIABILITY