MONTE CARLO PIVOTAL CONFIDENCE BOUNDS FOR WEIBULL ANALYSIS, WITH IMPLEMENTATIONS IN R

Jurgen Symynck[1], Filip De Bal[2]

[1]*KAHOSint-Lieven,*jurgen.symynck@kahosl.be
[2]*KAHOSint-Lieven,* filip.debal@kahosl.be

Abstract: To gain expert insight in the inner workings and pitfalls ofcommercial lifetime analysis software, the authors created an open source alternative with asubset of analysis tools and made it freely available as a package for the R statistical software, called the Weibull Toolkit for R. This articlefocuses on creating pivotal confidence bounds using Monte Carlo simulation for B-lives from a Weibull model. These bounds were suggested by Lawless (ref. [1]) and are recommended for small sample sizes of (nearly) complete databy Abernethy and Fulton(ref. [2], [3]). Fully functional and annotated R code is presented, derived from the toolkit's codebase. For the latest version of this document, check ref. [4]. For more in-depth treatment of the Weibull analysis with R, check ref. [5].

Keywords: Weibull, R, pivotals, Monte Carlo, confidence bounds.

1 Introduction

1.1 The FATIMATProject

FATIMAT (FATigue In MATerials) (ref. [6])is a completed PWO project (ref. [7]) supported by KAHOSint-Lieven (ref. [8]) that investigated cheaper alternatives to servo hydraulic dynamic testing machines, like servo pneumatic (ref. [9]) and electromechanical spindle actuator machines.

Products can be tested in various ways: for example an engine mount for fixing car engines to a chassiscan be tested by cyclically compressing it between 10 [mm] and 20 [mm] on a Zwick/Roell EZ020 high speed electromechanical spindle actuator ata load frequency of 3 [Hz]. After some time, the specimen will show signs of wear-out (in this case oil leakage) that are detected by the machine. The test is then halted and the failure time is recorded. This test is repeated (under identical loads and circumstances!) for a number of specimens.

One of the subprojects of FATIMAT was to analyze these failure data using the Weibull lifetime distribution for drawing conclusions on the general reliability of the specimens.Commercial packages such as Reliasoft'sWeibull++ (ref. [10]) and superSMITHWeibull (ref. [3]) were evaluated, but for gaining expert insight the authors created their own software: the WeibullToolkit (ref. [11]) for the statistical software R (ref. [12]).

1.2 The Two-Parameter WeibullDistribution

The two-parameter Weibull model is widely used in the field of reliability engineering, because it allows useful analysis with extremely small sample sizes (two failures or less). The Weibull model covers many other distributions (either exactly or approximately) like the(log-)normal and exponential.Also, an informative graphical plot can be created that helps to convey the analysis results to non-statisticians like engineers and their managers. In many cases, a two-parameter Weibull model is sufficient for accurately describing failure data. Its cumulative distribution function (c.d.f.) is given by:

$$F(t) = 1 - \mathrm{e}^{-\left(\frac{t}{\eta}\right)^{\beta}} \qquad (1)$$

The 'shape' parameterβ indicates the type of failure: $\beta < 1$ is a sign of infant mortality while $\beta > 1$ is a sign of wear out failures. $\beta = 1$ suggests a constant failure rate, not related to lifetime. The 'scale' parameter η, also called the 'characteristic life'represents the age at which 63.2 [%] of the specimen have failed.

1.3 The R Statistical Programming Language

(from the Rhomepage, ref. [12]:) "R is a language and environment for statistical computing and graphics. It is a GNU project which is similar

to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) […] R provides a wide variety of statistical [...] and graphical techniques, and is highly extensible."

R can be downloaded with no cost from its homepage and can be installed on most computers. It is essentially a console-like application where the user enters commands at the prompt. Multiple commands can be scripted and stored in a plain text file, making complete applications possible. Some graphical interfaces for R, and some dedicated R code editors like Tinn-R (ref. [13]) are available, making R easier to use.

1.4 The WeibullToolkit in R

The WeibullToolkit is a package for the R statistical programming language. It was initially conceived for doing analysis on the very simple reliability problem of complete, uncensored data, but its feature set and capabilities are continuously updated.In this paper, its code is demonstrated using simplified toolkit functions: just copy and paste the code in an open R console.More details on the Weibull toolkit can be found under ref. [5] and [14].

The WeibullToolkit is hosted online at Sourceforge (ref. [11]) and can be downloaded as an installable package. The source code can be browsed online (ref. [15]) or downloaded using the 'Git fast version control' system (ref. [16]).

2 Entering Data in the R Console

Let us assume that eight engine mounts were tested under the previously mentioned conditions (cyclic compressive sine wave load alterations between 10 [mm] and 20 [mm], 3 [Hz], same environmental temperature and humidity, …).The recorded observations are 149971, 70808, 133518, 145658, 175701, 50960, 126606 and 82329 cycles. All specimen were tested to failure (in this case: failure by oil leakage) creating a so-called complete dataset. The data is imported in R by opening the R console and entering the following code at the prompt:

```
### Entering failure data ###
d <- data.frame(
  time=c(149971, 70808,133518,
  145658, 175701, 50960,
  126606, 82329), event=1)
```

This creates a table-like variable named `d` of the 'dataframe'class, with two columns: `d$time` and `d$event`.The latter represents the event at the time of observation: here, all the specimens failed, corresponding with event '1' or 'died'. Just type `d` followed by <ENTER> at the prompt to display the contents of `d`(note that help is available for `data.frame` and all other functions by typing `?data.frame` followed by <ENTER> at the prompt).

2.1 Censored Data

In many cases, mechanical-dynamical testing means only a few specimens are tested (3-8) until failure, after which the failure time is recorded. This creates a 'complete' or 'uncensored' dataset. Sometimes an upper test duration limit is enforced, after which unfailed specimen are taken from the machine and labelled as 'survived'. This kind of censored data is called 'right censored' data or 'suspended' data.

The WeibullToolkit can handle this type of censored data, but for simplicity, this (important) subject is not treated in this article and the presented code does not support it; check out ref. [5] for a more detailed treatment.

3 CreatingaWeibullPlot

The goal of the Weibull plot is to provide a useful 2D representation of the observations. On the vertical axis, the 'unreliability' of the specimens is found while on the horizontal axis one finds the observation time. The double logarithmic scale of the Weibull plot's vertical axis together with the horizontal axis' logarithmic scale makes the Weibullc.d.f. (eq. [1])appear as a straight line (fig. [1]).

3.1 Median Rank Regression

To create the straight line representing the sample's β and η, a vertical plotting position is assigned to the ordered observation times ('ranking').These ranksare equivalent to the 'unreliability' of the specimens' population. Then, by means of simple linear least-square regression on transformations of the observations and median ranks, the Weibull parameter estimates are calculated from the data points.Several methods exist for the rank assignment (Hazen's, Bernard's, mean ranks) but here the inverse of incomplete beta function is used, which is considered the best method.

For small and moderate sample sizes (2-100) with few or no censoring it is always best practice to determinethe estimates usingmedian rank regression (MRR) in favour of other techniques like maximum likelihood estimation (MLE) (ref. [2]).Continue by running the following code to assign median ranks to the previously entered failure data (note that for this uncensored data the `d$event` column is not used or needed, simplifying the ranking process):

```
### Ranking failure data ###
mrank.ob <- function(j,f){
    r <- qbeta(0.5,j,
      f-j+1);r}
mrank.data <- function(d){
  n <- nrow(d)
  d$rank  <- rank(d$time,"first")
  d <- d[order(d$rank), ]
  d$mrank <-
    mrank.ob(d$rank, n);d}
print(d <- mrank.data(d))
```

The ranking is done by the`qbeta()` function, the inverse of the incomplete beta function and part of the standard libraries of R. The dataframed now holds the median ranks for each observation in the `d$mrank` column, as shown below.

```
    time event rank      mrank
6  50960     1    1 0.08299596
2  70808     1    2 0.20113119
8  82329     1    3 0.32051897
7 126606     1    4 0.44015520
3 133518     1    5 0.55984480
4 145658     1    6 0.67948103
1 149971     1    7 0.79886881
5 175701     1    8 0.91700404
>
```

After ranking the observations, a line can be fitted through the data points, calculating the Weibull parameter estimates.Continue by running the following code to fit a line through the data points:

```
### Median Rank Regression ###
F0inv<- function(p){
log(log(1/(1 - p)))}
fwb <-
lm(log(d$time)~F0inv(d$mrank),d)
beta <- 1/coef(fwb)[2]
eta <- exp(coef(fwb)[1])
print(paste("beta=",signif(beta)))
print(paste("eta =",signif(eta)))
```

The `lm()` function fits a straight line by fitting `log(d$time)` on a transformation of the median ranks.The `F0inv()`transformation, together with the logarithmic transformation of the observations allow treating the soughtWeibullc.d.f. as a straight line, making a line fit possible (the transformations can be derived from eq. [1] and will result in $F(t) = \beta \log_e(t) - \beta \log_e(\eta)$, which is of the same structure as the standard line equation $y = a\,x + b$).

Note that here it is the observations that are fitted on the ranks (X-on-Y regression), and not vice-versa (Y-on-X, as is standard in most implementations of least square fitting).It is good practice to fit the variable with the most variability (the observations) on those with less variability: the values of the median ranks are exactly defined and do not depend on the actual values of the observations, only on their positions in the list of ordered lifetimes, and the total number of observations!

```
Call:
...
[1] "beta= 2.58128"
[1] "eta = 132512"
>
```

3.2 The WeibullPlot

The WeibullToolkit automates all of the above steps. For convenience however,the stand-alone code for a simplified version of the Weibull plot is presented here. Because R lacks native support for automatically transforming the vertical axis to the double logarithmic scale, this must be done manually with the `F0inv()` function (logarithmic transformations are supported by the `log="x"` parameter).It will be used frequently with most graphical functions in this article.Continue by running the following code to produce aWeibull plot for the given example:

```
### Simplified Weibull plot ###
options(scipen=10) # no scientific
                   # notation
plot(y=F0inv(d$mrank),x=d$time,
  log="x",axes=F,lwd=2,cex=1.2,
  main="Engine mount cyclic test",
  xlim=c(5000,500000),
  ylim=F0inv(c(0.01,0.99)),
  xlab="time",ylab="Unrel. [%]")
```

```
curve(add=TRUE,lwd=2,
  beta*log(x)-beta*log(eta))
ygrid <-
  c(1:9,seq(10,90,10),91:99)
axis(2,F0inv(ygrid/100),
  ygrid,lwd=2);axis(1,lwd=2)
abline(v=c(5000,seq(1e4,1e5,1e4),
  seq(1e5,5e5,1e5)),
  h=F0inv(ygrid/100))
abline(lty=2,
  h=F0inv(ETA <- 1-exp(-1)))
```

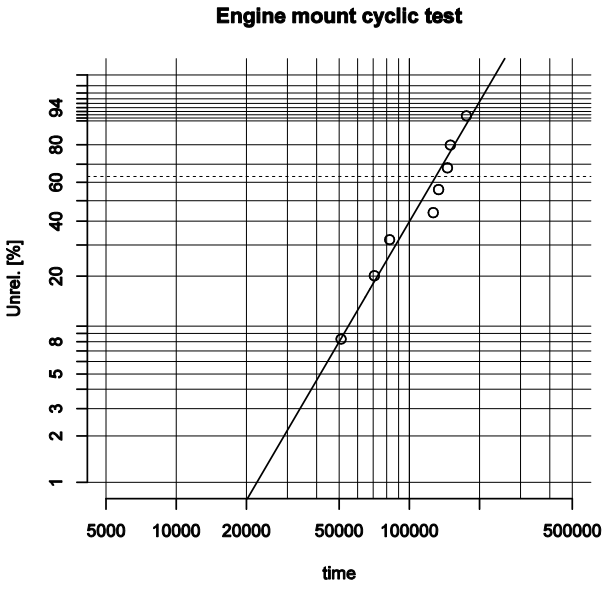

Figure 1:*SimplifiedWeibull plot, showing the median rank regression (MRR) fitted line, representing the Weibull parameters β and η.*

After plotting the median ranks versus the observations, the `curve()`function draws a straight line with$beta = \hat{\beta}$and$eta = \hat{\eta}$. Note that an identical line can be plotted by substituting `curve(...)` with the following code, where in the argument of `F0inv()` one recognises the Weibullc.d.f. from eq. [1].

```
### Alternative curve() method ###
curve(add=TRUE,lwd=2,
F0inv(1-exp(-(x/eta)^beta)))
```

Grid lines are plotted at horizontal and vertical plot position (vertical positions are listed in `ygrid)`. The 63.2 [%] (dashed) unreliability line is also plotted: the point of its intersection with the fitted line provides the characteristic life of the distribution, η.

4 B-life

When the Weibull plot is created, predictions of the failure behaviour of the specimenspopulation can be made.

The B-life is the age at which a certain percentage of the investigated populationis expected to fail (based on the analyzed sample!). For example, the B10-life for the population of engine mounts can be read from the plot and isapprox. 55500 cycles (the number following the capital 'B' indicates the unreliability percentage). The B1-life is approx. 22300 cycles. Just draw a horizontal line at the 10 [%]unreliabilityand find the intersection with the fitted (straight) line.Then, read the age from the horizontal axis.Run the lower code for marking the B10 life:

```
### Marking the B10 B-life ###
abline(lwd=2,lty=2,
v=55500,h=F0inv(0.1))
```

In R, the B-lives are very easily calculated by means of the `qweibull(p,beta,eta)` function which is part of the R standard libraries.It calculates the p^{th}quantile from the Weibull model described by $\hat{\beta}$and $\hat{\eta}$. Executing `qweibull(c(0.1,0.01),beta,eta)`calculates the B10 and B1 life, respectively:

```
[1] 55415.93 22299.16
>
```

5 Confidence in Predicted B-lives

5.1 Definition of Confidence Interval

It is evident that B-livesbased on Weibull parameter estimates from large sample sizes are to be taken more seriously than those based on two or three observations; small samples contain very limited lifetime information.To get an indication of the confidence that one should have on an estimated B-life, a confidence interval can be calculated. A90 [%]confidence interval for a B-life has the following meaning:

"When the B-life would be estimated over and over again from samples similar to the original one, then the real, unknown B-life of the specimens population will, with a 90 [%] frequency, be situated inside the 90 [%] confidence interval."

A 90 [%] confidence interval is limited by two bounds, who can also be described as – at the lower side – thelower confidence bound of a

95 [%] confidence interval with no upper limit (100 [%]), and – at the upper side – the upper confidence bound of a 95 [%] confidence interval with no lower limit (0 [%]).

This means that the real, unknown B-life exceeds the lifetime at the lower confidence boundwith a 95 [%] frequency. Also: the real, unknown B-life will,with a 5 [%] frequency, be smaller than the lifetime at the lower confidence bound. The latter shows that there is still a small chance that the actual B-life turns out to be worse than anticipated by the confidence bounds!

5.2 Usage

According to the method demonstrated in this article the real, unknown B10 life for the provided example lies between approx.24500 and 87000 cycles, with 90 [%] confidence level. The B1-life (estimated at 22300 cycles) liesin a big 4700 – 50200 cycles interval, with 90 [%] confidence level.

Generally, confidence levels of 90 [%] are used and silently implied. In the automotive industry, a confidence level of 50 [%] is often used, meaning that the confidence concept isnot used at all but B-lives are read straight from the fitted line. In this case they rely on large sample sizes for reliable predictions. Much higher confidence levels are used (90 [%] to 99.9 [%] or more) in the medical and aircraft industry.

The presented confidence limits widen dramatically when the sample size decreases. Although still valid, it is clear that the interval will be so wide that it could be of little practical use: increasing the sample size is the only remedy against wide confidence intervals.

5.3 Calculating Confidence Bounds

To obtain a lower bound for a 90 [%] confidence bound for a B10 life, one must find the 5th percentile of the distribution of the B-lives at the given (10 [%]) uncertainty. This distribution is not easily described and it is, except for complete or 'Type II' censored data (where all the unfailed specimens are censored at the time of the latest failure) difficult or impossible to calculate analytically (ref. [1]). Some methods approximate this distribution by a well known distribution or apply transformations to the B-lives.

Confidence bounds come in a great variety: beta-binomial bounds, Fisher's matrixbounds, likelihood ratio bounds, Monte Carlo pivotal bounds and bootstrap bounds are the most popular. The first are very easily calculated (even by hand) but cannot be extrapolated to lower unreliability's; precisely where the reliability engineer needs them. Abernethy (ref. [2]) concludes that Monte Carlo pivotal bounds are best practice when median rank regression is used, given that the needed computer power is available. For larger sample sizes (> 400), likelihood ratio bounds for MLE or MLE-RBA based Weibull estimates are faster to calculate. This article describes how to calculate the Monte Carlo pivotal bounds.

5.4 Straightforward Monte Carlo Bounds

It would make sense to calculate the confidence interval of a B10 life using the following Monte Carlo based method:

- Calculate the parameter estimates $\hat{\beta}$ and $\hat{\eta}$ from the original sample $T = t_1, t_2, \ldots, t_n$ with sample size *n*, as explained earlier.
- Create a lot (2000 <= *R*<= 5000)of synthetic samples (same*n*) by randomly generating synthetic observations based on $\hat{\beta}$ and $\hat{\eta}$, resulting in $T^*_{1\ldots R} = (t^*_1, \ldots, t^*_n)_1, (t^*_1, \ldots, t^*_n)_2, \ldots, (t^*_1, \ldots, t^*_n)_R$. In R this is accomplished by repeating the `rweibull(n,beta,eta)` function*R*times.
- Find the synthetic Weibull parameters estimates$\hat{\beta}^*$ and $\hat{\eta}^*$for all the synthetic samples, $\hat{\beta}^*_{1\ldots R} = \hat{\beta}^*_1, \hat{\beta}^*_2, \ldots, \hat{\beta}^*_Rand\hat{\eta}^*_{1\ldots R} = \hat{\eta}^*_1, \hat{\eta}^*_2, \ldots, \hat{\eta}^*_R$.
- Calculate the B10 life for each synthetic Weibull plot, resulting in $\mathrm{B10}^*_{1..R} = \mathrm{B10}^*_1, \mathrm{B10}^*_2, \ldots, \mathrm{B10}^*_R$by repeating `qweibull(0.1,beta,eta)`for all $\hat{\beta}^*$ and $\hat{\eta}^*$.
- Calculate the 5th and 95thpercentile (for a 90 [%] confidence interval) from the empirical distribution of $B10^*_{1\ldots R}$ using the `quantile()` function. These values represent the lower and upper confidence bounds.

The above straightforward method turns out to provide too optimistic (narrow) intervals, especially for small sample sizes (*n*< 20). The reason is that the distribution of the synthetic B10 lives (from which one calculates the 5th and 95th percentiles) depends too much on the real, unknown values of β and η. The above calculations are based on *estimates*$\hat{\beta}^*$and $\hat{\eta}^*$, who themselves become less accurate with smaller sample sizes. The deviations of these estimates

from theirreal, unknown values should be reflected in the range of the confidence intervals, but they are not. When bigger sample sizes are used, the estimates are closer to their true values, making theseconfidence bounds more realistic, however.

5.5 Monte Carlo Pivotal Confidence Bounds

A better approach is to derive the 5th and 95th percentiles from a 'pivotal' quantity: this is a quantity that does not depend on the underlying true and unknown distribution parametersβ and η. Lawless (ref. [1]) supplies a pivotal quantity for determining B-life distributions:

$$Z_p = \frac{\hat{u}-y_p}{\hat{b}} \quad (2)$$

Where $\hat{u} = \ln(\hat{\eta})$, $y_p = \ln(t_p)$ and $\hat{b} = 1/\hat{\beta}$. $t_p$isthe time where the unreliability is $p = F(t;\beta,\eta)$ from eq. [1]; the Weibullc.d.f. for the real, unknown parameters β and η. Otherwise said, $t_p = F^{-1}(p;\beta,\eta)$ which is the inverse of the Weibull c.d.f..Practically,t_p is the B(100*p)-life where a confidence bound is to be calculated for.

By definition, Z_P does not depend on unknown parameters, meaning that its distribution formstays the same regardless of the actual values of β and η.

The pivotal can therefore be calculated by arbitrary setting $\beta = \eta = 1$ (or $u = 0$ and $b = 1$) for further calculations.

Recall: $$t_p = F^{-1}(p;\beta,\eta) \quad (3)$$

Let $w_p = \ln(t_p)$, and with $\beta = \eta = 1$:

$$w_p = \ln(F^{-1}(p;1,1)) = \ln\left(F_0^{-1}(p)\right) \quad (4)$$

Eq. [4] can now be calculated for it represents the inverse of the standard Weibullc.d.f. (eq. [1]) and results in:

$$w_p = \ln\left(\ln\left(\frac{1}{1-p}\right)\right) \quad (5)$$

To determine the empirical distribution of Z_p, Monte Carlo methods are used again:

- Create a lot (2000<= R <= 5000) of synthetic samples (same size as the original sample) by randomly generating synthetic observations from the standard Weibull model ($\beta = \eta = 1$).
- Find the synthetic standard Weibull parameter estimates$\widehat{u_0}^* = \ln(\widehat{\eta_0}^*)$and $\widehat{b_0}^* = 1/\widehat{\beta_0}^*$ for all the synthetic samples, giving$\widehat{u_0}^*_{(1\ldots R)}$and $\widehat{b_0}^*_{(1\ldots R)}$.
- Calculate all pivotalsz_p^* using a combination of eq. [2] and [5]:

$$Z_p = z_{p(1\ldots R)}^* = \frac{\widehat{u_0}^*_{(1\ldots R)} - w_p}{\widehat{b_0}^*_{(1\ldots R)}} \quad (6)$$

The empirical distribution of these pivotals can now be calculated. To calculate the bounds for a 90 [%] confidence interval for the B10 life, on takes the q=5th and q=95thpercentile of $Z_{(p=0.10)}$ and calculates the corresponding B-life with a rearranged eq. [2]:

$$y_{(p,q)} = \ln(\hat{\eta}) - Z_{(p,q)}/\hat{\beta} \quad (7)$$

$$t_{(p,q)} = e^{y_{(p,q)}} \quad (8)$$

Continue with the next code block to load the `pivotals()` function:

```
### Pivotals function def. ###
MC<- function(n){
  std <- data.frame(
time=rweibull(n,1,1),
event=1)
  std <- mrank.data(std)
  fwb <- lm(log(std$time)~
F0inv(std$mrank),std)
c(u0_hat=coef(fwb)[1],
b0_hat=coef(fwb)[2])}
pivotals<- function(r,R,unrel){
  piv <- as.data.frame(
t(replicate(R,MC(r))))
  wp <- F0inv(unrel)
Zp <- function(wp){
((piv$u0_hat-wp)/piv$b0_hat)}
piv <- cbind(piv,sapply(wp,Zp))
  names(piv) <-c("u0_hat",
"b0_hat",signif(unrel))
piv}
```

The `MC()` function calculates and returns one pair of $\widehat{u_0}^*$ and$\widehat{b_0}^*$ by fitting a Weibull line trough a synthetic sample from the standard Weibull distribution. The `pivotals()` function repeats `MC()` for `R` times (usually `R=2000`),applies eq. [6] with `sapply()` on all $\widehat{u_0}^*$ and$\widehat{b_0}^*$ and binds the pivotalsto the `piv`dataframein an extra column.

Note that `wp<- F0inv(unrel)` gives the same results as `wp<- log(qweibull(unrel,1,1))`, showing the relation with the standard Weibull model and the calculation of B-lives.Proceed by calculating the pivotals for determining the B10 life'sconfidence interval, based on the exemplary samplewith size *n*=8:

```
###B10 pivotals calculation ###
piv <- pivotals(8,2000,0.1)
head(piv,3);tail(piv,3)
```

In the third column of the `piv`dataframe, one can find the 2000 pivotal quantities (only a few are displayed with the `head()` and `tail()` functions).Continue by calculating the pivotal percentiles and the actual confidence intervals:

```
### B10 pivotal conf. bounds ###
tp_low <- exp(log(eta)-quantile(
piv[,'0.1'],0.95)/beta)
tp_upp <- exp(log(eta)-quantile(
piv[,'0.1'],0.05)/beta)
print(paste(
"B10 90[%] CI = (",
signif(tp_low),",",
signif(tp_upp),")"))
points(pch=3,lwd=2,cex=2,
   x=c(tp_low,tp_upp),
   y=rep(F0inv(0.1),2))
```

Eq. [8] is executed twice for the 5^{th} and 95^{th} percentile. Finally, the pivotal confidence interval for the B10 life is calculated and displayed, and added to the plot:

```
[1] "B10 90[%] CI =
( 23931.3 , 86688.7 )"
>
```

Note that recalculated bounds will never be exactly identical because of the Monte Carlo variability. For lower B-lives like B1, many replications like `R=5000` or more may be necessary. If repeatable results are needed, one can run `set.seed(1)` before all code to lock the random number seed value.

Repeating the above steps for a number of unreliability levels and connecting the points so that a curve emerges is the next logical step; these are the pivotal confidence bounds as plotted in the WeibullToolkit:

```
### Plot pivotal conf. bounds ###
pivotal_CB<- function(piv,CL){
unrel <- as.numeric(names(
piv[,c(-1,-2)]))
rdf <- data.frame(unrel=unrel,
row.names=unrel)
  Tp <- function(Zp,conf){
  exp(log(eta)-quantile(
Zp,conf)/beta)}
rdf <- cbind(rdf,
lower =sapply(piv[,c(-1,-2)],
Tp,1-(1-CL)/2),
upper =sapply(piv[,c(-1,-2)],
Tp,(1-CL)/2))
rdf}
piv<- pivotals(8,2000,ygrid/100)
CB <- pivotal_CB(piv,0.90)
lines(lwd=2,
CB$lower,F0inv(CB$unrel))
lines(lwd=2,
CB$upper,F0inv(CB$unrel))
```

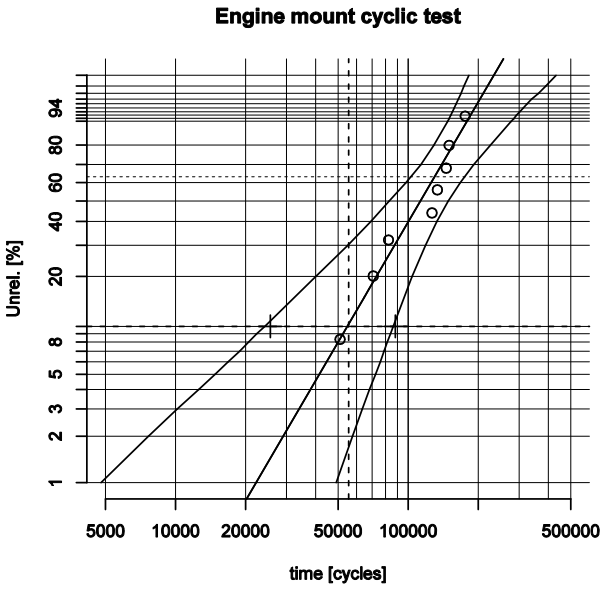

Figure 2:*Weibull plot with Monte Carlo pivotal confidence bounds for a 90 [%] confidence level.*

The B10 confidence intervals can now be read from the graph in the same way as the regular B10 lives.

6 Conclusion

The article demonstrates the inner workings of the Weibull Toolkit for R, an open source reliability and lifetime data analysis package. After demonstrating R code for calculating the Weibull parameters for complete lifetime data

using the median rank regression method,Monte Carlo pivotal confidence bounds are discussed and calculated. Where appropriate, alternative calculation methods are briefly mentioned and discussed. The calculation of the pivotal quantities is clarified in detail and applied in functional R code, culminating in a simplified version of the Weibull plot as generated by the Weibull toolkit.

7 References

[1] Jerald F. Lawless, *Statistical Models and Methods for Lifetime Data*, 2nd edition, Wiley-Interscience, Hoboken N.J., 2003.

[2] Robert B. Abernethy, *The New Weibull Handbook*, 5th Edition, 2008.

[3] http://www.barringer1.com/wins.htm

[4] http://mechanics.kahosl.be/fatimat/index.php/downloads-and-information/40/198

[5] JurgenSymynck, Filip De Bal, *The Weibull model in fatigue and reliability analysis: an introduction & implementation in R', KAHO Sint- Lieven, 2011*
http://mechanics.kahosl.be/fatimat/index.php/downloads-and-information/40/77

[6] http://mechanics.kahosl.be/fatimat

[7] http://www.kahosl.be/site/index.php?p=/nl/page/153/

[8] http://www.kahosl.be

[9] http://www.youtube.com/watch?v=g2IczVyuQkQ

[10] http://www.reliasoft.com

[11] http://sourceforge.net/projects/weibulltoolkit/

[12] http://www.r-project.org/

[13] http://www.sciviews.org/Tinn-R/

[14] JurgenSymynck, Filip De Bal, *Weibull analysis using R, in a nutshell*,New Technologies and Products in Machine Manufacturing Technology, Stefan cel Mare University. of Suceava, 2010
http://mechanics.kahosl.be/fatimat/index.php/downloads-and-information/40/171

[15] http://weibulltoolkit.git.sourceforge.net/git/gitweb-index.cgi

[16] http://git-scm.com/
http://code.google.com/p/msysgit/

8 Further reading

– Chi-Chao Lui, *A Comparison Between The Weibull And Lognormal Models Used To Analyse Reliability Data*, dissertation from University of Nottingham, 1997.

– William Q. Meeker and Luis A. Escobar, *Statistical Methods for Reliability Data*, Wiley-Interscience, New York, 1998.

– http://www.weibull.com/
Reliasoft's 'Reliability subjects' website

– http://cran.r-project.org/web/packages/survival/
The homepage of the 'survival' R package for general survival and reliability analysis.

– http://cran.r-project.org/web/packages/boot/
The homepage of the 'boot' R package, for bootstrapping functions.