act 11: Weibull distribution for reliability

Dr. Islam

April 21, 2022

# Review of basic concepts

## Time to Failure

* Let be the time-to-failure of a product or system with the pdf .
* The product fails within the time interval with the probability (CDF)
* The probability that the product is in operation beyond the time interval , called the reliability function , defined by
* The failure rate function, , is the rate at which a time old item fails in the next instant of time. It follows that
* The mean time to failure (MTTF) of a product is
* It also follows that
* The mean residual life of the item at age is defined by

# Weibull distribution

Let be the time-to-failure of a product. We say that follows an Weibull distribution with shape parameter and scale parameter , denoted by , if the pdf of is given by

Due to the popularity of the Weibull model in reliability study, the analysis of life data is often referred to as a Weibull Analysis. This model is popular because by adjusting the distribution parameters, it can be used to model a wide range of life distribution characteristics of different classes of engineered items. For example, it can be noted that if , an Weibull distribution becomes exponential distribution

# Applications of Weibull distribution

* The probability that the product with the failure time fails within the time interval is

**Proof**

By definition,

Let . Then

Also, when , then and when , then . Then,

* The **reliability function** of is given by
* The **failure rate function**, , is the rate at which a time old item fails in the next instant of time. It follows that
* The **mean time to failure (MTTF)** of is given by
* The moment of is given by

**Proof**

Let . Then Also, when , then and when .

Now,

The above results suggest that

Then,

* [**Grad student @ Final Exam**] The mean residual life of the item at age , , is
* # Estimation of Weibull parameters

The parameters of distribution can be estimated via the following procedures:

* MLE (“mle”, maximum likelihood estimators)
* MME (“mme”, methods of moments estimators)
* MMUE (“mmue”, method of moments based on the unbiased estimator of variance).

We implement R to get the parameter estimated for us.

# R function of parameter estimation

* **EnvStats::eweibull()** function estimates parameters by “mle” (maximum likelihood estimators; the default), “mme” (methods of moments estimators), and “mmue” (method of moments based on the unbiased estimator of variance) using options method = “mle”, etc.
* **MASS::fitdistr(x, densfun=“weibull”, lower = 0)** provides MLEs. The option “lower = 0” is added to ensure that estimated Weibull parameters .
* **weibullness::weibull.mle()** provides MLEs of parameters.
* **fitdistrplus::fitdist(ussample,“weibull”)** also provides MLEs of parameters.

**Act 11.1**

Read the *Weib1.csv* dataset uploaded in Canvas into R.

This dataset contains two columns , which are assumed to come from and distributions, respectively.

For each of the data vectors

1. Estimate the true mean and true variance .
2. Estimate the mean and variance using the *var()* functions. How do the estimates compare to the true values in (a)?
3. Estimate the parameters and for each of the data vectors? How do they compare to their true values?
4. Fit an Weibull distribution to the data vector w1 and perform ks.test to assess the goodness of the fit of the data vector to the Weibull model.

require(MASS)

Loading required package: MASS

shape1=6  
scale1=3  
shape2=11  
scale2=4  
dat1=read.csv("reliAct11.csv", header=T)  
names(dat1)

[1] "w1" "w2"

w1=dat1$w1  
w2=dat1$w2  
# (a)  
tmw1=scale1\*gamma((1/shape1)+1)  
tvw1=scale1^2\*(gamma((2/shape1)+1)-(gamma((1/shape1)+1))^2)  
c(tmw1, tvw1)

[1] 2.7831580 0.2908471

tmw2=scale2\*gamma((1/shape2)+1)  
tvw2=scale2^2\*(gamma((2/shape2)+1)-(gamma((1/shape2)+1))^2)  
  
c(tmw2, tvw2)

[1] 3.8203181 0.1763922

#(b)  
emw1=mean(w1)  
evw1=var(w1)  
c(tmw1, emw1, tvw1, evw1)

[1] 2.7831580 2.7754600 0.2908471 0.2762935

emw2=mean(w2)  
evw2=var(w2)  
c(tmw2, emw2, tvw2, evw2)

[1] 3.8203181 3.8306700 0.1763922 0.1682937

#(c) Using package function to estimate parameters  
#EnvStats::eweibull(w1, method = "mle")  
#EnvStats::eweibull(w2, method = "mle")  
fit1=MASS::fitdistr(w1, "weibull",lower=0)  
fit2=MASS::fitdistr(w2, "weibull",lower=0)  
fit1

shape scale   
 6.22962188 2.98643547   
 (0.15472351) (0.01594837)

fit2

shape scale   
 11.04873917 4.00884386   
 ( 0.27097406) ( 0.01209695)

#(d)  
ks.test(w1,"pweibull", scale=fit1$estimate[2], shape=fit1$estimate[1] )

Warning in ks.test(w1, "pweibull", scale = fit1$estimate[2], shape =  
fit1$estimate[1]): ties should not be present for the Kolmogorov-Smirnov test

One-sample Kolmogorov-Smirnov test  
  
data: w1  
D = 0.022094, p-value = 0.7134  
alternative hypothesis: two-sided

#

**Act 11.2**

Read the *reli3.csv* dataset uploaded in Canvas into R.

This dataset contains four columns . We are note sure what distribution can be used to model data columns. We will assume that these data vectors come from an distribution and wish to test which data vector is actually coming from the hypothesized distribution.

At the first step, we estimate parameters via MLE method and test if the underlying data actually fits the hypothesized distribution via goodness of fit test.

To proceed by doing the following tasks:

1. Estimate the shape and scale parameters for each column of the data vectors via MLE method.
2. Estimates mean and variance using the estimated parameters (invariance property of MLE applies here).
3. Use data to estimate mean and variance using *mean()* and *var()* functions and compare the results with the estimates in (b)
4. Use the estimated parameters to fit the data to the hypothesized distribution for each data vectors.
5. Test the fitted distributions if they come from the hypothesized distribution.
6. What is your conclusion in regard to which data vectors come from the hypothesized distribution?

require(MASS)  
dat2=read.csv("reli3.csv", header=T)  
names(dat2)

[1] "x1" "x2" "x3" "x4"

x1=dat2$x1  
x2=dat2$x2  
x3=dat2$x3  
x4=dat2$x4  
  
#EnvStats::eweibull(x4, method = "mle")  
#EnvStats::eweibull(x4, method = "mme")   
  
#(a) fit data to MAAS::fitdistr() function in package MASS  
fit1 <- fitdistr(x1, "weibull",lower=0)  
fit2 <- fitdistr(x2, "weibull",lower=0)  
fit3 <- fitdistr(x3, "weibull",lower=0)  
fit4 <- fitdistr(x4, "weibull",lower=0)  
fit1

shape scale   
 0.95299195 1.34819817   
 (0.02347262) (0.04710929)

fit2

shape scale   
 1.44561223 1.57248152   
 (0.03700042) (0.03608417)

fit3

shape scale   
 5.00324609 2.98705226   
 (0.12411604) (0.01988756)

fit4

shape scale   
 9.938855899 1.994787020   
 (0.242679225) (0.006688298)

#(b)  
scale1=fit1$estimate[2]  
shape1=fit1$estimate[1]  
  
em1=scale1\*gamma((1/shape1)+1)  
ev1=scale1^2\*(gamma((2/shape1)+1)-(gamma((1/shape1)+1))^2)  
c(em1, ev1)

scale scale   
1.377679 2.091313

scale1=fit1$estimate[2]  
shape1=fit1$estimate[1]  
  
em1=scale1\*gamma((1/shape1)+1)  
ev1=scale1^2\*(gamma((2/shape1)+1)-(gamma((1/shape1)+1))^2)  
c(em1, ev1)

scale scale   
1.377679 2.091313

scale2=fit2$estimate[2]  
shape2=fit2$estimate[1]  
  
em2=scale2\*gamma((1/shape2)+1)  
ev2=scale2^2\*(gamma((2/shape2)+1)-(gamma((1/shape2)+1))^2)  
c(em2, ev2)

scale scale   
1.42640 1.00419

scale3=fit3$estimate[2]  
shape3=fit3$estimate[1]  
  
em3=scale3\*gamma((1/shape3)+1)  
ev3=scale3^2\*(gamma((2/shape3)+1)-(gamma((1/shape3)+1))^2)  
c(em3, ev3)

scale scale   
2.7427209 0.3942031

#(c)  
c(mean(x1), var(x1))

[1] 1.377743 2.112809

c(mean(x2), var(x2))

[1] 1.4326660 0.9238261

c(mean(x3), var(x3))

[1] 2.7411988 0.3959423

c(mean(x4), var(x4))

[1] 1.89799706 0.05101364

#(d&e) GOF test to fitted models via ks.test  
ks.test(x1,"pweibull", scale=fit1$estimate[2], shape=fit1$estimate[1] )

One-sample Kolmogorov-Smirnov test  
  
data: x1  
D = 0.01305, p-value = 0.9957  
alternative hypothesis: two-sided

ks.test(x2,"pweibull", scale=fit2$estimate[2], shape=fit2$estimate[1] )

One-sample Kolmogorov-Smirnov test  
  
data: x2  
D = 0.041501, p-value = 0.06383  
alternative hypothesis: two-sided

ks.test(x3,"pweibull", scale=fit3$estimate[2], shape=fit3$estimate[1] )

One-sample Kolmogorov-Smirnov test  
  
data: x3  
D = 0.016771, p-value = 0.9412  
alternative hypothesis: two-sided

ks.test(x4,"pweibull", scale=fit4$estimate[2], shape=fit4$estimate[1] )

One-sample Kolmogorov-Smirnov test  
  
data: x4  
D = 0.013845, p-value = 0.9908  
alternative hypothesis: two-sided

**Act 11.3**

Time to failure of a pipeline valve follows an Weibull distribution with a shape parameter 2 and a scale parameter 4000 hours. Calculate reliabilites of the valve for an operation period of 2000, 4000 and 6000 hours.

**Solution**

Recall that the reliability function is given by

Then,

R2000=exp(-(2000/4000)^2)  
R2000

[1] 0.7788008

#78%  
R4000=exp(-(4000/4000)^2)  
R4000

[1] 0.3678794

#37%  
R6000=exp(-(6000/4000)^2)  
R6000

[1] 0.1053992

#11%  
#F(t) and R(t) can be computed directly using pweibull(t, shape=alpha, scale=beta) and pweilbull(t, shape=alpha, scale=beta, lower.tail=F), functions respectively:  
  
pweibull(2000, shape=2, scale=4000, lower.tail=F)

[1] 0.7788008

pweibull(4000, shape=2, scale=4000, lower.tail=F)

[1] 0.3678794

pweibull(6000, shape=2, scale=4000, lower.tail=F)

[1] 0.1053992

**Act 11.4**

Bearings are highly engineered, precision-made components that enable machinery to move at extremely high speeds and carry remarkable loads with ease and efficiency. A recent accelerated life test of bearings resulted in the Weibull distribution with shape parameter and scale parameter hours.

* Find the reliability of bearings at one year, assuming that 1 year=365 days, or 8,760 hours.
* Find the reliability of bearings at 48,500 hours.
* Find the MTTF

**Solution**

The reliability function for the Weibull distribution is:

Where is the shape parameter and is the scale parameter. Therefore, we have

Thus, given the Weibull parameters, the reliability of bearings at one year is 97.3%.

The reliability of bearings at 48,500 hours is

Thus, the reliability of bearings at 48,500 hours is 36.8%.

* Note that if time equals it doesn’t matter what the value is, the reliability is 36.8% (try it).

pweibull(8760, shape=2.1, scale=48500, lower.tail = F)

[1] 0.9728828

pweibull(48500, shape=2.1, scale=48500, lower.tail = F)

[1] 0.3678794

The MTTF is