

### HW#3 Survival analysis II, Spring 2015

Name: Shih Jia-Han

Data: 36.3, 41.7, 43.9, 49.9, 50.1, 50.8, 51.9, 52.1, 52.3, 52.3,

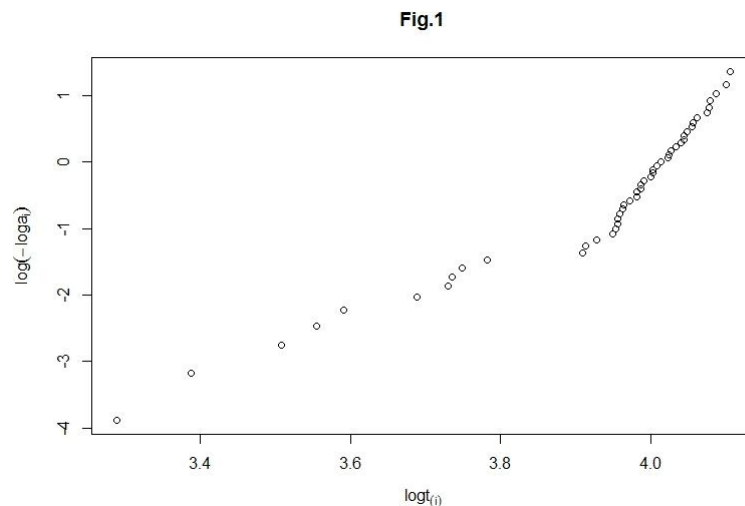
52.4, 52.6, 52.7, 53.1, 53.6, 53.6, 53.9, 53.9, 54.1, 54.6,

54.8, 54.8, 55.1, 55.4, 55.9, 56.0, 56.1, 56.5, 56.9, 57.1,

57.1, 57.3, 57.7, 57.8, 58.1, 58.9, 59.0, 59.1, 59.6, 60.4,

60.7, 26.8, 29.6, 33.4, 35.0, 40.0, 41.9, 42.5 (the last 7 are right-censored)

To estimate the MLE, we have to check whether the Weibull distribution can fit the data or not. We can check it by plot the basic Weibull probability plot.



In Fig.1, the points plotted lie near a straight line. Hence this data set can be fitted by the Weibull distribution.

The log-likelihood function of Weibull distribution is:

$$\log L(\xi, \nu) = \sum_{i=1}^n c_i (\log \nu - \nu \log \xi) + (\nu - 1) \sum_{i=1}^n c_i \log t_i - \sum_{i=1}^n \left( \frac{t_i}{\xi} \right)^\nu$$

where  $c_i$  is the censoring indicator.

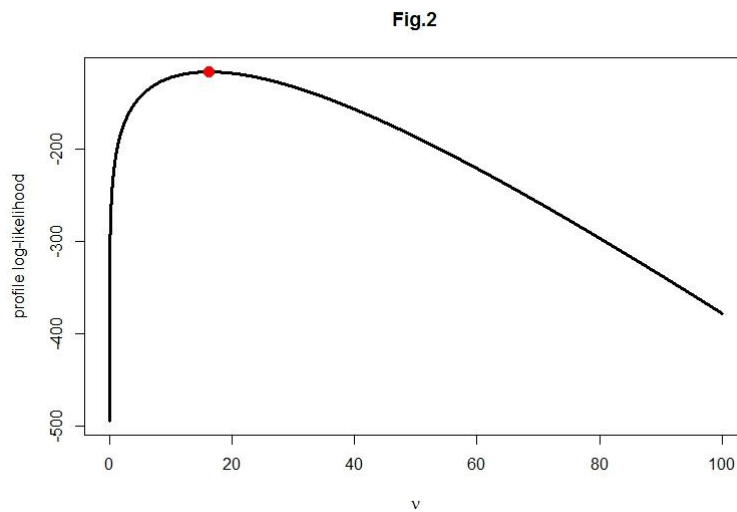
That means  $c_i = \begin{cases} 1 & \text{if } T_i = t_i \\ 0 & \text{if } T_i > t_i \end{cases}$

Then set  $\frac{\partial \log L}{\partial \xi} = 0$  to produce  $\xi^\nu = \frac{1}{n_c} \sum_{i=1}^n t_i$ , where  $n_c = \sum_{i=1}^n c_i$

Since the MLE  $\hat{\xi}_\nu = \left( \frac{1}{n_c} \sum_{i=1}^n t_i \right)^{\frac{1}{\nu}}$  is in terms of  $\nu$ . So,  $\xi$  in  $\log L$  can be replaced

by  $\hat{\xi}_\nu$  to produce the profile log-likelihood  $\log L(\hat{\xi}_\nu, \nu)$  for  $\nu$ .

Then we can perform grid search by R to find the MLE  $\hat{\nu}$ , and it is **16.309**.



The point in Fig.2 is the MLE  $\hat{\nu}$ .

In section 3.2.6, we can apply routine `survreg` from the `survival` package. The output is:

```
Call:
survreg(formula = sobj ~ 1, dist = "weibull")
      value Std. Error      z      p
(Intercept)  4.03    0.00999 402.9 0.00e+00
Log(scale)  -2.79    0.12515 -22.3 3.19e-110

Scale= 0.0613

weibull distribution
Loglik(model)= -115.8  Loglik(intercept only)= -115.8
Number of Newton-Raphson Iterations: 11
n= 48
```

Since the scale here is different. So, we need to transform by  $\nu = 1/\text{scale}$ .

Therefore,  $1/0.0613 \approx 16.31321$  and it is very close to the MLE that I produced from the profile log-likelihood.

## R code

```
##### Data set #####
obs_data=c(36.3,41.7,43.9,49.9,50.1,50.8,51.9,52.1,52.3,52.3,
           52.4,52.6,52.7,53.1,53.6,53.6,53.9,53.9,54.1,54.6,
           54.8,54.8,55.1,55.4,55.9,56.0,56.1,56.5,56.9,57.1,
           57.1,57.3,57.7,57.8,58.1,58.9,59.0,59.1,59.6,60.4,
           60.7)
cens_data=c(26.8,29.6,33.4,35.0,40.0,41.9,42.5)
str_data=c(obs_data,cens_data)

nobs=length(obs_data)
ncens=length(cens_data)
nd=nobs+ncens
cens=c(rep(1,nobs),rep(0,ncens))

##### Data checking #####
ordered_data=sort(str_data,decreasing=F)
a=c()
n=length(ordered_data)
for (i in 1:n){

  a[i]=(n-i+1)/(n+1)

}

plot(log(ordered_data),main="Fig.1",log(-log(a)),xlab = expression(logt[(i)]),ylab =
expression(log(-loga[i])))

##### Profile log-likelihood function #####
ll_func=function(nu){

  k=1/nu
  xi=(sum(str_data^nu)/nobs)^k

  sum(cens*(log(nu)-nu*log(xi)))+(nu-1)*sum(cens*log(str_data))-sum((str_data/xi)^n
  u)

}
```

```

##### Plot profile log-likelihood function #####
q=seq(0.001,100,by=0.001)
ll=c()

for(i in 1:length(q)){

  nu=q[i]
  ll[i]=ll_func(nu)

}

c=which.max(ll)
nu_mle=0.001*c
nu_mle

plot(q,ll,type="l",main="Fig.2",xlab = expression(nu),ylab = "profile
log-likelihood",lwd = 3)
points(nu_mle,max(ll),cex=1.5,col=2,pch=16)

##### Analysis #####
library(survival)
sobj=Surv(str_data,cens,type="right",origin=0)
sfit=survreg(sobj~1,dist="weibull")
summary(sfit)

```