STAT 687: Mathematics of Reliability Theory I
Summer 2012 Assignment 2

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

Homework 3:

Do the following exercises from Meeker and Escobar:

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7

As an <u>optional</u> challenge, try 5.4

Adjustments, hints and comments:

- For 4.7, please do this problem analytically using appropriate derivatives. Theory!
- For 4.8, appendix Section B.1 give the gerneal theory for transformation of random variables, but the simple special case on the top of page 618 is all that is needed here. Note that g^{-1} denotes the inverse of function g. That is $g^{-1}\left(g(x)\right) = x$, and if $g(x) \equiv \log(x)$, then $\exp(\log(x)) = x$, because exp is the inverse of log. Also note that usually the CDF is easier to work with in univariate cases like this, but the pdf method is also feasible.
- For 4.11, thinking about the log of T might make this simpler
- For 4.13, To do these, you need to use the well known definitions (assume they are true, you've probably shown them before):

$$E(Y) = \int_{-\infty}^{\infty} y f(y) dy$$

$$Var(Y) = \int_{-\infty}^{\infty} [y - E(Y)]^2 f(y) dy$$

$$= \int_{-\infty}^{\infty} y^2 f(y) dy - [E(Y)]^2$$

Homework 4:

Do the following exercises from Meeker and Escobar:

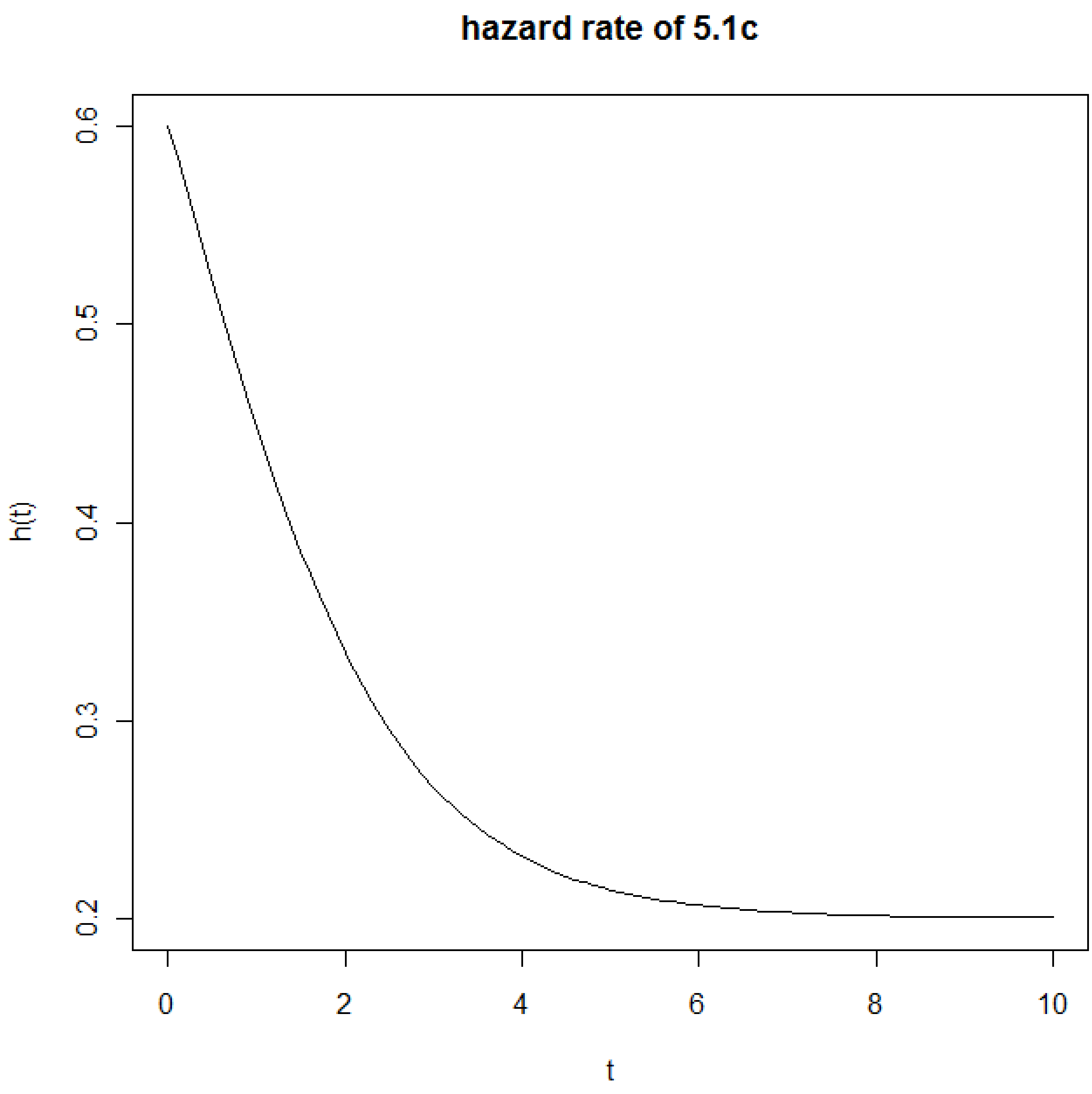
6.5, 6.9, 6.10, 6.11, 6.12, 6.15

Adjustments, hints and comments:

- This is the first homework where you really ought to use RSPLIDA. For 6.5, you might try the following code:
    - plot(lzbearing.ld,"Weibull")
    - plot(lzbearing.ld,"Lognormal")
    - temp<-npprobplot(lzbearing.ld,"Weibull"); print(temp)
    - Note that you have to type this manually – R doesn't recognize the office's quotation marks, and office copies the bullet too when I try and copy and paste.
- For 6.9, we need to import data. Please see the Problem6.9.R file at L:\Courses\STAT\STAT 687\R Stuff\Code. Also note that I'm providing you code to teach you how to do it – don't expect me to do all you coding for you.
- For 6.11, be careful – an adjustment is needed because the scale in the figure is in base 10 logarithms, but the usual definition of the lognormal is based on natural logs.
- For 6.12, your answers should not be too far apart – assume your eyeball of the plot is good. You might want to check against ML estimates to be sure.

STAT 687: Mathematics of Reliability Theory I
Summer 2012 Assignment 2

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15
4.7 – 4.13: See handwritten pages.
5.1a,b,c: See handwritten pages.
5.1d:



5.1e: It is obviously not constant, even though the two parts of the mixture distribution have constant hazard rates. The intuition is that the hazard rate is decreasing because the unreliable parts are generally failing out first, leading to their percentage of the population decreasing and thus reliability increase (decreasing hazard rate).
5.1f: It is improving in that the weak population is dying out more quickly with time, thus reducing the percentage of the population that is weak. Thus if you chose a unit that had been in service for t=6

units, you'd be relatively unlikely to get a weak unit, (hazard rate 1) and much more likely to get a stronger unit (hazard rate 1/5).

5.3: See handwritten notes.

6.3: A) Handwritten notes
B) the scale parameter of your distribution is important because it drives what you really want to estimate (t_p for some p), but in and of itself probably isn't that important, especially in computers computers will be hopelessly outdated by the time $t_{.63}$ is reached (leaving aside Microsoft jokes).
C) Nope – you'd have a log of extrapolation on the plot, which means lots of extrapolation and lots of uncertainty.
D) When we linearize, β becomes the slope of the line – should have a reasonable estimate. However, η is going to have issues because of extrapolation (small uncertainty in β will lead to LARGE uncertainty in η). We can however estimate smaller quantiles $t_{.1}$.

6.7: A) According to the binomial method, $\hat{F}(t) = \frac{\#failed\ at\ time\ t}{n}$, so $F(t)$ simply increases by .01 at each failure time. The Kaplan Meier estimate is given in the excel table below.

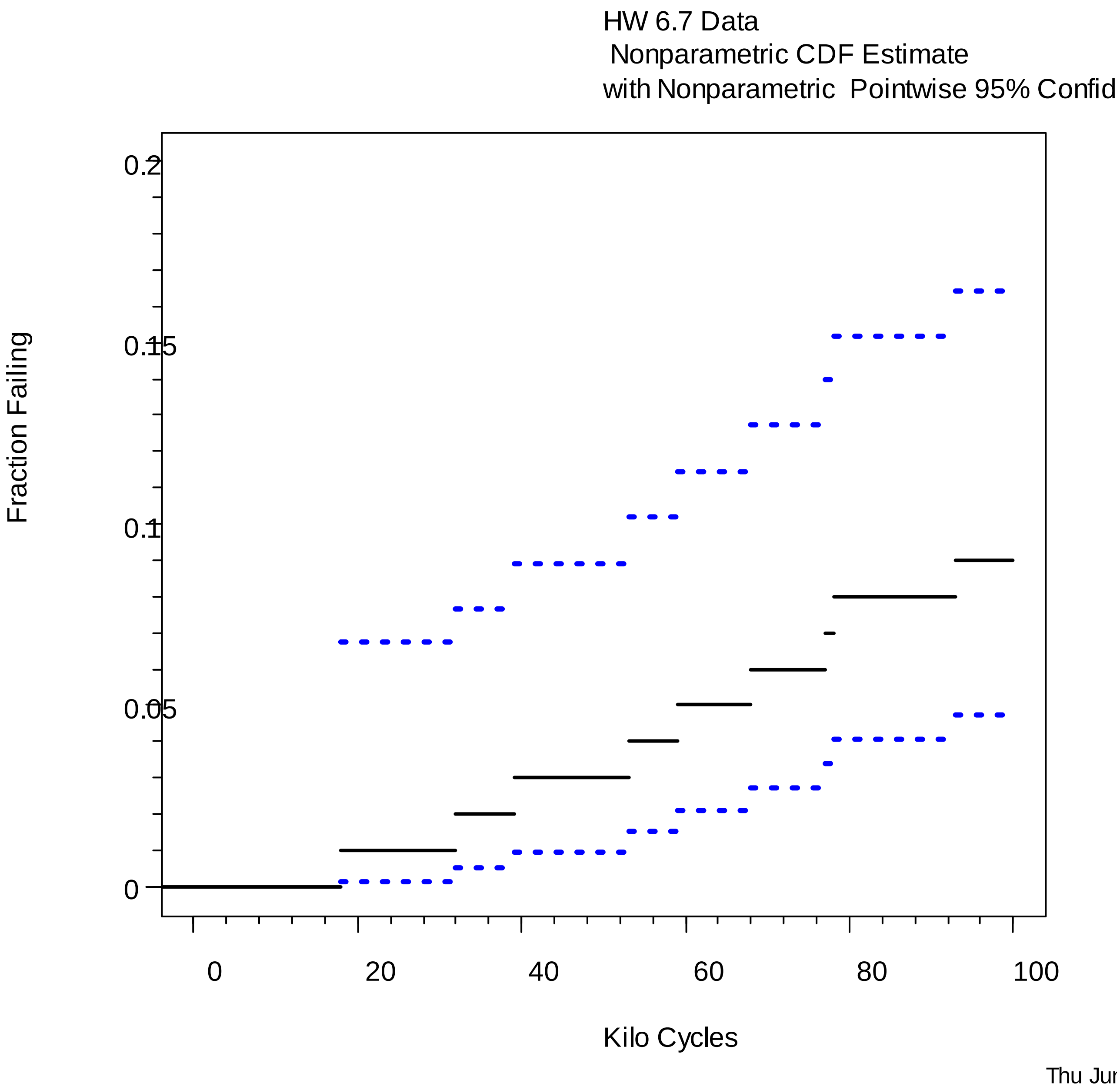
Time	di	Censored	entered	pjhat	1-pjhat	s(ti)hat	F(ti)hat
18	1	0	100	0.01	0.99	0.99	0.01
32	1	0	99	0.010101	0.989899	0.98	0.02
39	1	0	98	0.010204	0.989796	0.97	0.03
53	1	0	97	0.010309	0.989691	0.96	0.04
59	1	0	96	0.010417	0.989583	0.95	0.05
68	1	0	95	0.010526	0.989474	0.94	0.06
77	1	0	94	0.010638	0.989362	0.93	0.07
78	1	0	93	0.010753	0.989247	0.92	0.08
93	1	0	92	0.01087	0.98913	0.91	0.09
100	0	91	91				

B) Now that I've done it in Excel, lets do it in Splida to show why we actually use Splida.
Code:
```
HW6.7Frame<-
      matrix(c(18,32,39,53,59,68,77,78,93,100,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,91),nco
      l=3)
HW6.7Frame<-as.data.frame(HW6.7Frame)
names(HW6.7Frame)<-c("Time","Censor Code","Count")
HW6.7Frame
HW6.7.ld<-frame.to.ld(HW6.7Frame,response.column=1,censor.column=2,case.weight.column =
      3,time.units = "Kilo Cycles",data.title="HW 6.7 Data")
Output<-plot(HW6.7.ld)
Output
> Output


Nonparametric estimates from  HW 6.7 Data
with approximate 95% pointwise confidence intervals
```

STAT 687: Mathematics of Reliability Theory I
Summer 2012 Assignment 2

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

	Kilo Cycles-lower	Kilo Cycles-upper	Fhat	SE_Fhat	95% Lower	95% Upper
1	0	18	0.00	0.00000	0.000000	0.00000
2	18	32	0.01	0.00995	0.001407	0.06753
3	32	39	0.02	0.01400	0.005008	0.07643
4	39	53	0.03	0.01706	0.009708	0.08890
5	53	59	0.04	0.01960	0.015094	0.10176
6	59	68	0.05	0.02179	0.020965	0.11454
7	68	77	0.06	0.02375	0.027204	0.12717
8	77	78	0.07	0.02551	0.033736	0.13961
9	78	93	0.08	0.02713	0.040512	0.15188
10	93	100	0.09	0.02862	0.047494	0.16400

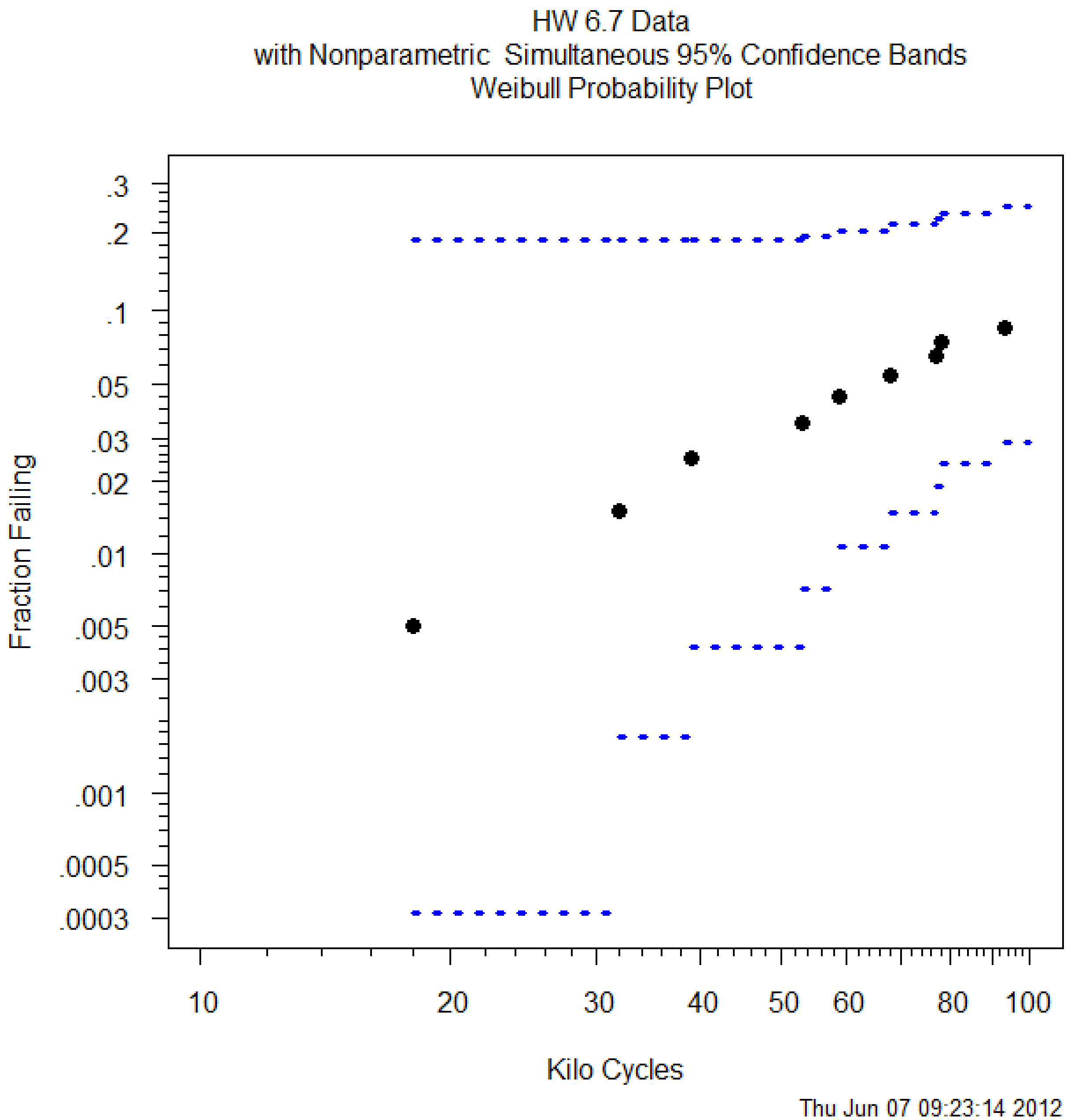


Much easier, isn’t it?
C) You can do this by hand, but why?

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

Code:

plot(HW6.7.ld,distribution="Weibull")



Let us eyeball this chart and estimate the slope to get a rough guess at β. I'm going to guess at 20 kilo Cycles the fraction failing is about .006, and at 100KC its about .9, so slope is rise over run (in their respective scales (qsev and log), is 1.709807.

Code:

```
rise<-qsev(.09)-qsev(.006)
run<-log(100)-log(20)
rise/run
```

D): The plot is very nearly a straight line. This is a good fit.

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

E) We saw 9% fail, which is pretty close to 10% that we want to estimate. Therefore, this is probably adequate to estimate this quantile, though the extrapolation is going to necessarily widen our confidence bounds.

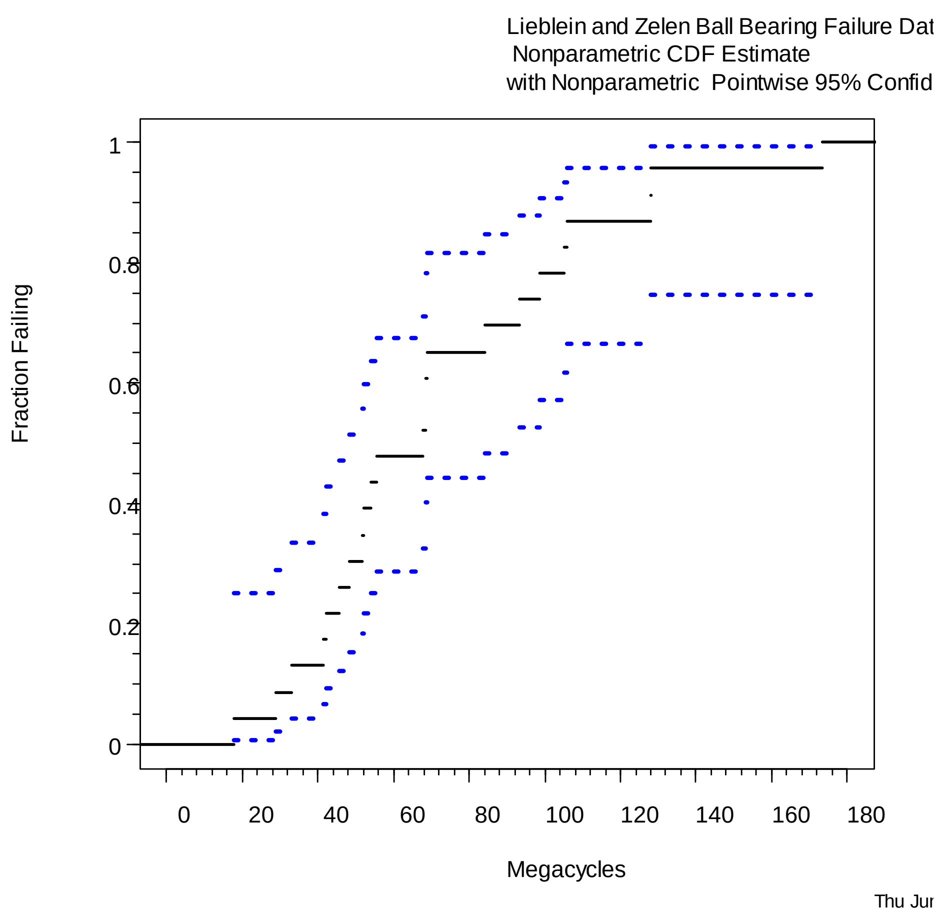
STAT 687: Mathematics of Reliability Theory I
Summer 2012 Assignment 2

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

6.5: A)

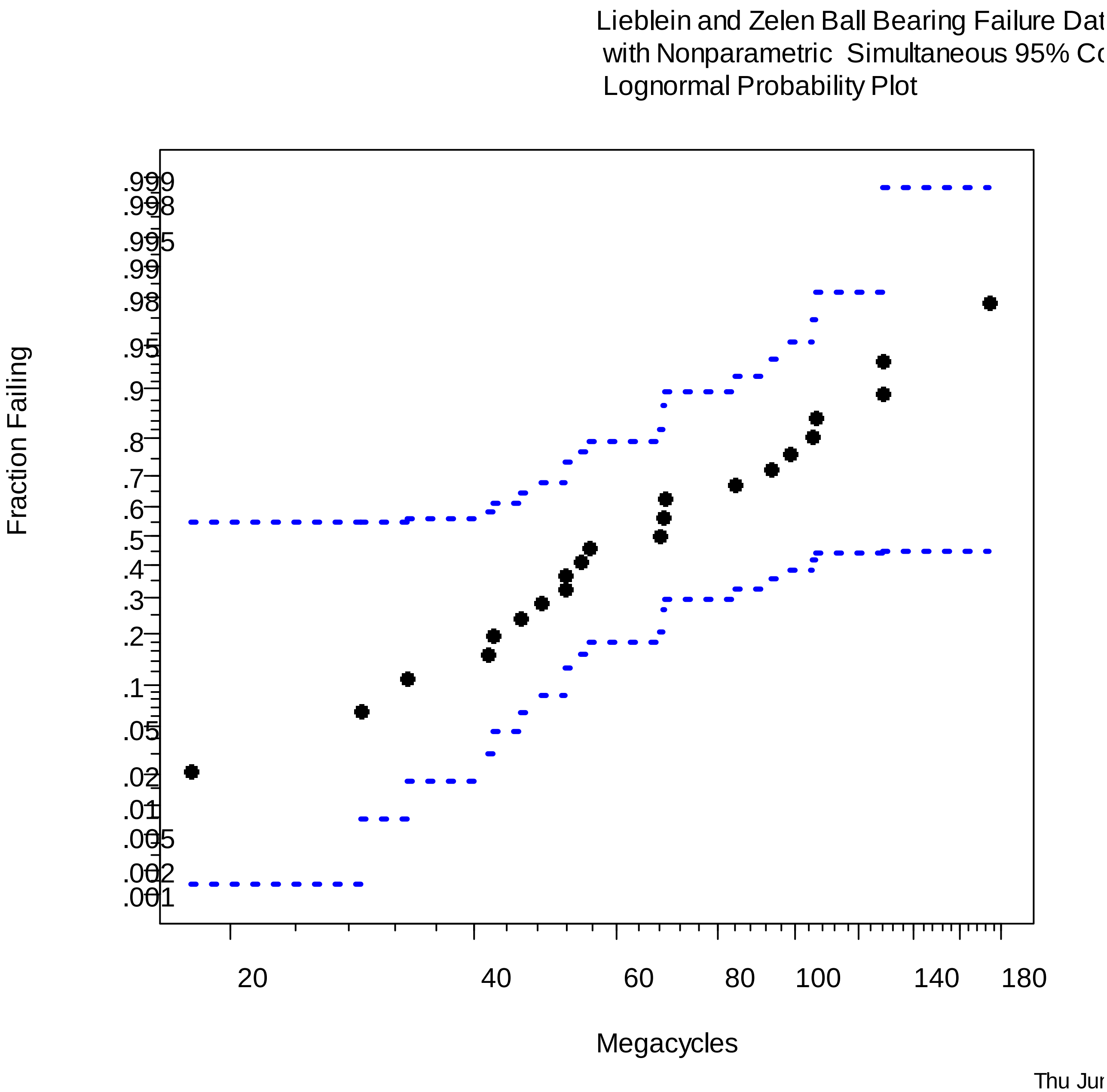
```
Nonparametric estimates from  Lieblein and Zelen Ball Bearing Failure Data
with approximate 95% pointwise confidence intervals

   Megacycles-lower Megacycles-upper    Fhat SE_Fhat 95% Lower 95% Upper
1              0.00        1.788e+01 0.00000 0.00000   0.00000    0.0000
2             17.88        2.892e+01 0.04348 0.04252   0.00609    0.2522
3             28.92        3.300e+01 0.08696 0.05875   0.02184    0.2888
4             33.00        4.152e+01 0.13043 0.07022   0.04267    0.3355
5             41.52        4.212e+01 0.17391 0.07903   0.06683    0.3823
6             42.12        4.560e+01 0.21739 0.08601   0.09349    0.4280
7             45.60        4.840e+01 0.26087 0.09156   0.12216    0.4723
8             48.40        5.184e+01 0.30435 0.09594   0.15253    0.5154
9             51.84        5.196e+01 0.34783 0.09931   0.18442    0.5571
10            51.96        5.412e+01 0.39130 0.10176   0.21768    0.5976
11            54.12        5.556e+01 0.43478 0.10337   0.25223    0.6369
12            55.56        6.780e+01 0.47826 0.10416   0.28799    0.6751
13            67.80        6.864e+01 0.52174 0.10416   0.32495    0.7120
14            68.64        6.888e+01 0.60870 0.10176   0.40238    0.7823
15            68.88        8.412e+01 0.65217 0.09931   0.44289    0.8156
16            84.12        9.312e+01 0.69565 0.09594   0.48463    0.8475
17            93.12        9.864e+01 0.73913 0.09156   0.52766    0.8778
18            98.64        1.051e+02 0.78261 0.08601   0.57203    0.9065
19           105.12        1.058e+02 0.82609 0.07903   0.61773    0.9332
20           105.84        1.279e+02 0.86957 0.07022   0.66455    0.9573
21           127.92        1.280e+02 0.91304 0.05875   0.71115    0.9782
22           128.04        1.734e+02 0.95652 0.04252   0.74782    0.9939
23           173.40        1.000e+29 1.00000 0.00000   1.00000    1.0000
```

Plots:

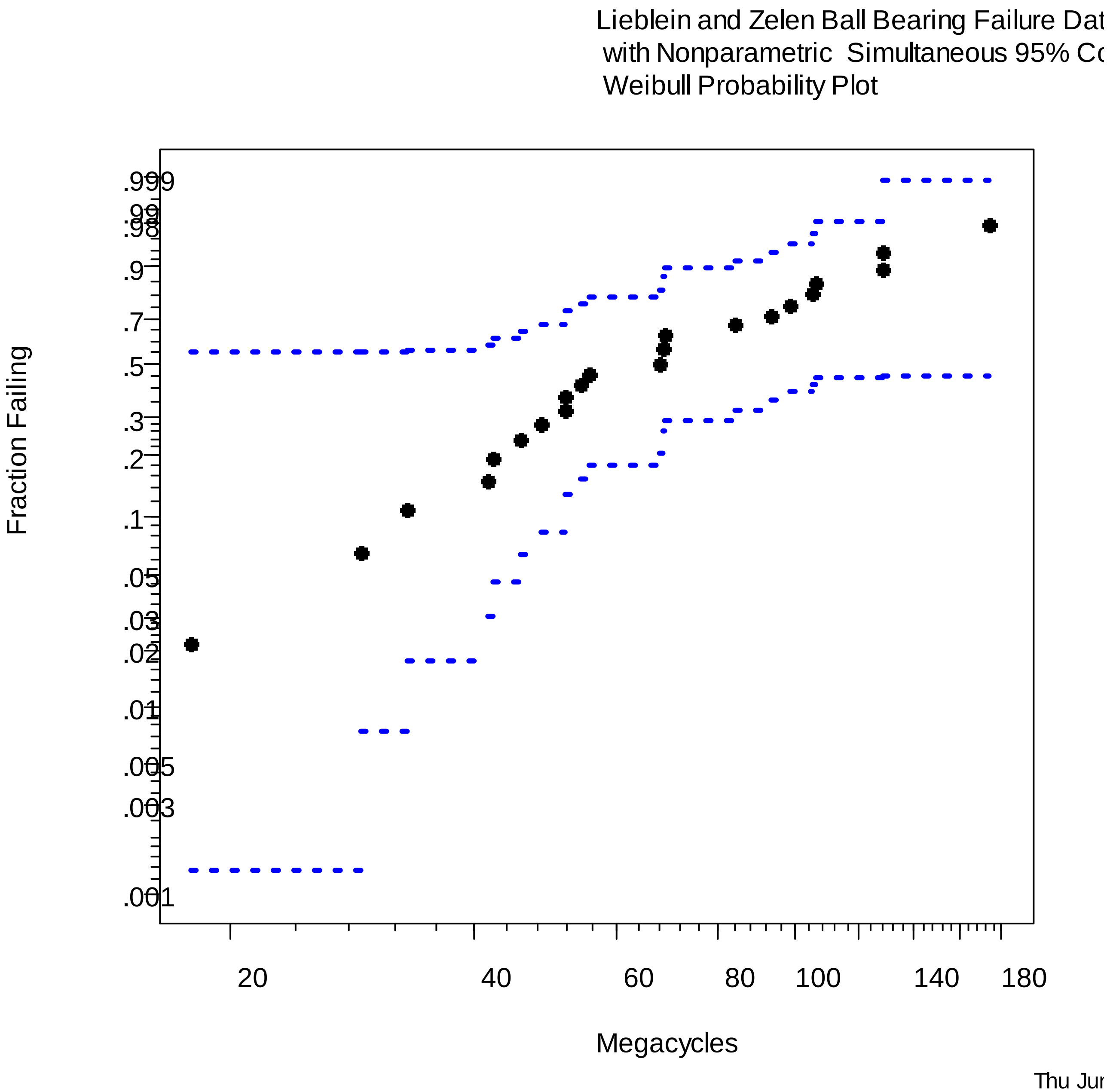


4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

6.5B) Or, you could have Splida do it. I know which one I want to do!



6.5C)



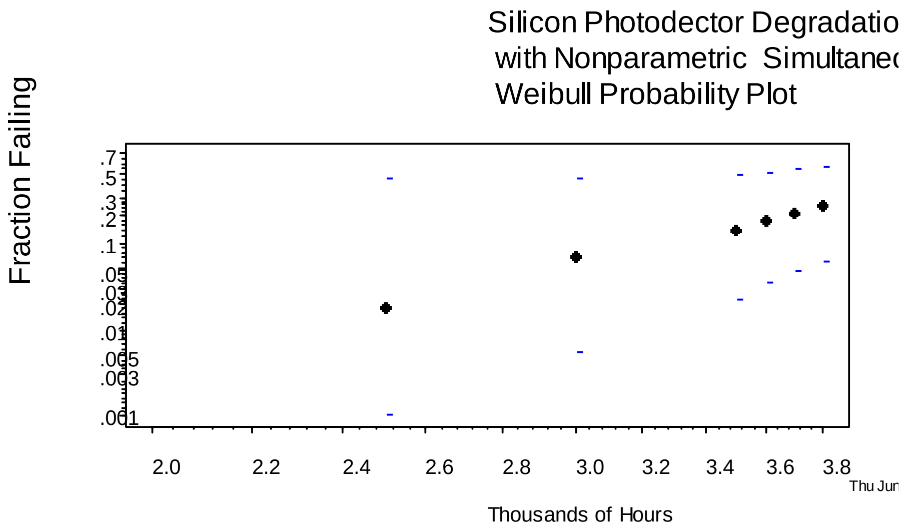
6.5D) Both distributions appear to fit well (both appear to be relatively straight lines). The lognormal might be just a touch better, but it is really close and I'd have a hard time choosing one over the other.

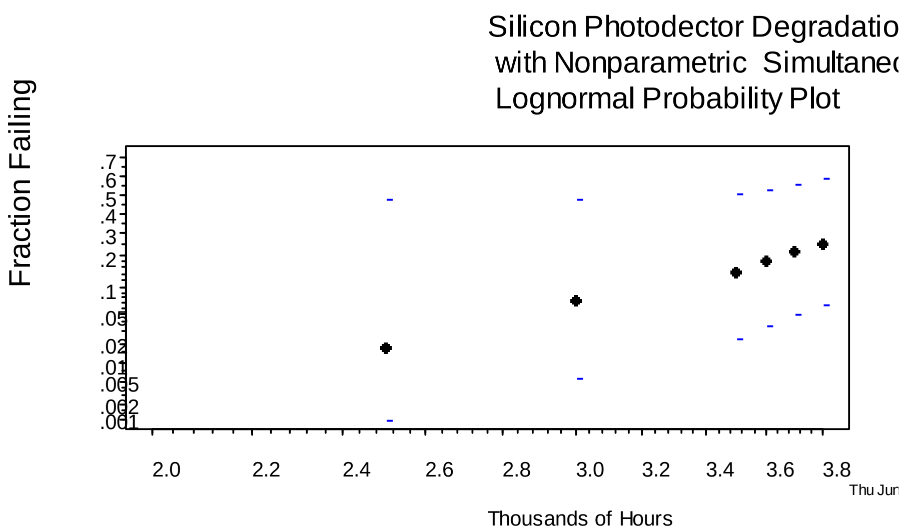
6.5 Code:

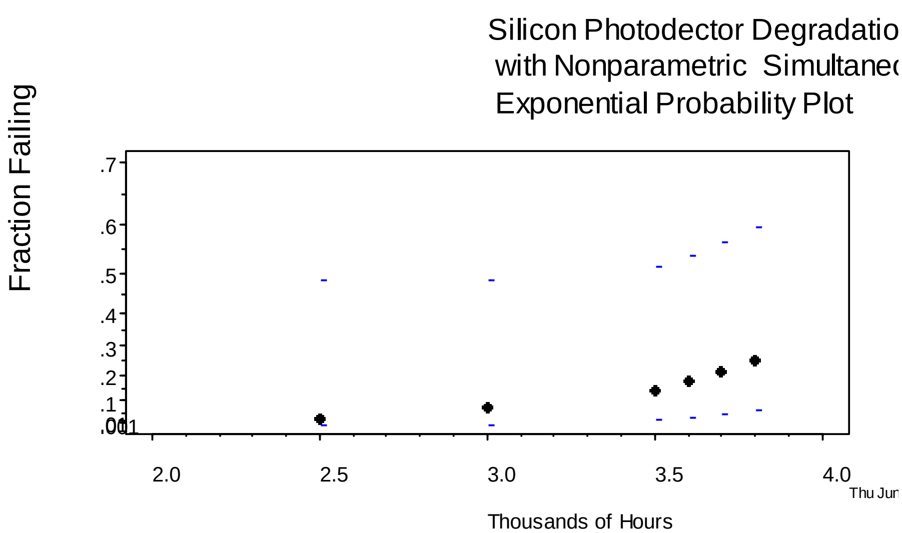
```
#Problem 6.5
#lzbearing.ld should be automatically included with RSplida.
temp<-plot(lzbearing.ld) #plot in linear scale
temp #give non-parametric est to data
plot(lzbearing.ld,'lognormal')
plot(lzbearing.ld,'weibull')
```

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

6.9)







The exponential does NOT fit, others appear okay.

STAT 687: Mathematics of Reliability Theory I
Summer 2012 Assignment 2

4.7, 4.8, 4.11, 4.13, 5.1, 5.3, 6.3, 6.7, 6.5, 6.9, 6.10, 6.11, 6.12, 6.15

Code:

```
photodet.ld<- frame.to.ld(file="I:/RSplidaAlpha/RSplida_text_data/photodetector.txt",
      response.column=c(1,2), censor.column=3, case.weight.column=4, data.title="Silicon
      Photodector Degradation Failure Time", time.units="Thousands of Hours", skip=3)
par(mfrow=c(3,1))
plot(photodet.ld,distribution="Weibull")
plot(photodet.ld,distribution="Lognormal")
plot(photodet.ld,distribution="Exponential")
```

6.10) A) this is a normal consequence of the lognormal distribution where p = F(t) = pnorm((log(t)-\mu)/\sigma). If we reverse this and find z=((log(t)-\mu)/\sigma such that p=.5, we se that z must be zero. (in a standard normal distribution the median = 0).

B) For the weibull distribution, the CDF looks like $p = \Phi_{\text{sev}}\left(\frac{\log(t)-\mu}{\sigma}\right)$, letting $z = \frac{\log(t)-\mu}{\sigma}$, we need to find p such that z=0, which implies that $p = \Phi_{\text{sev}}(z) \rightarrow \mathrm{p} = 1 - \exp(-exp(z)) \rightarrow \mathrm{p} = 1 - \exp(-\exp(0)) = 1 - \exp(1) \approx 0.6321206$.

6.11) So to do this, we recall that with the log normal distribution we estimate sigma using the slope and mu where the line intercepts the 0 standard quantile line. Now, this is complicated because the log-thousand cycles is in base-10 log ☹!. Reading off the chart, I eyeball that the line crosses the 0 line at 2.24 (the first hash mark after 2.2, so $\ln(10^{2.24}) = 5.15779$. For sigma, we see that the line of the data goes through the -1 standard quantile at about 2.09 log cycles, and through 1 standard quantile at roughly 2.38 log cycles. This implies that the slope of the line (rise/run) =(ln(10^2.38)-ln(10^2.09))/2 = 0.3338748. Note that the ln(10^___) is to convert from base 10 log to base-e log.

Now, to estimate F(200), simply put them into the formula using your estimates.
pnorm((log(200)-5.157791)/0.3338748)= 0.6630843.

6.12) Depending on which chart you look at and read the number from (I just used 6.5), it appears that your estimate should be ~.75. They shouldn't be different from each other (6.5, 6.6, 6.7) since they are the same plots on different scales. They are different from 6.11, because of lack of fit – that plot does not look particularly straight on log-normal paper, hence log normal approximation is not as good as it could be. Also, there is a tad bit of variance in "eye-ball the chart" estimates ☺.

6.15) See by hand notes