Name Jerry

Student Number

STA 312 s2023 Quiz 6

1. The Kapan-Meier estimate of the survival function is based on discrete time. Accordingly, let the survival time T be a discrete random variable with non-zero probability on the points $t_1, t_2, \ldots$. Let $p_j =$ the probability of surviving past time t_j, given survival to time t_{j-1}. That is, $p_j = P(T > t_j | T > t_{j-1})$.

   (a) (3 points) Prove $p_j = \frac{S(t_j)}{S(t_{j-1})}$.

$$P_j = P(T > t_j | T > t_{j-1}) = \frac{P(T > t_j \cap T > t_{j-1})}{P(T > t_{j-1})}$$

$$= \frac{P(T > t_j)}{P(T > t_{j-1})} = \frac{S(t_j)}{S(t_{j-1})}$$

   (b) (3 points) Assuming $t_0 = 0$ and $P(T > 0) = 1$ (which is very reasonable), prove $S(t_3) = p_1 p_2 p_3$. This is a special case of something on the formula sheet, and of course you cannot use what you are proving.

$$P_1 P_2 P_3 = \frac{S(t_1)}{S(t_0)} \frac{S(t_2)}{S(t_1)} \frac{S(t_3)}{S(t_2)}$$

$$= \frac{S(t_3)}{S(t_0)} = \frac{S(t_3)}{1} = S(t_3)$$

added the maximum likelihood estimate of S(t) to

2. (4 points) In Question 12 of Assignment 6, you were asked to plot the Kaplan-Meir and maximum likelihood estimates of $S(t)$ for a small data set. Attach the plot and the R code that produced it to this quiz. On your printout, mark the code that ~~produced~~ the plot, and and write "Question 2" beside it. Make sure your name and student number appear on the printout and plot.

```
R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.79 (8198) x86_64-apple-darwin17.0]

[Workspace restored from /Users/brunner/.RData]
[History restored from /Users/brunner/.Rapp.history]

> # A6
> 
> rm(list=ls()); options(scipen=999)
> exdata = read.table("http://www.utstat.utoronto.ca/brunner/data/legal/expo.data2.txt")
> head(exdata); Time = exdata$Time; Uncensored = exdata$Uncensored
   Time Uncensored
1 0.179          0
2 1.024          1
3 0.189          1
4 0.345          1
5 0.977          1
6 0.241          1
> 
> # 11(a)
> # install.packages("survival",dependencies=TRUE) # Only need to do this once
> library(survival)
> y = Surv(Time,Uncensored)
> km = survfit(y ~ 1); km
Call: survfit(formula = y ~ 1)

      n events median 0.95LCL 0.95UCL
[1,] 50     40  0.351   0.284   0.758
> sumkm = summary(km); sumkm
Call: survfit(formula = y ~ 1)

  time n.risk n.event survival std.err lower 95% CI upper 95% CI
 0.026     50       2   0.9600  0.0277      0.90719        1.000
 0.032     47       1   0.9396  0.0338      0.87557        1.000
 0.058     44       1   0.9182  0.0392      0.84448        0.998
 0.062     43       1   0.8969  0.0437      0.81511        0.987
 0.100     41       1   0.8750  0.0478      0.78610        0.974
```

```
 0.101     40       1   0.8531  0.0514      0.75811        0.960
 0.109     39       1   0.8312  0.0545      0.73095        0.945
 0.117     38       1   0.8094  0.0573      0.70448        0.930
 0.118     37       1   0.7875  0.0598      0.67860        0.914
 0.165     36       1   0.7656  0.0620      0.65324        0.897
 0.173     35       1   0.7437  0.0640      0.62835        0.880
 0.179     34       1   0.7219  0.0657      0.60388        0.863
 0.189     32       1   0.6993  0.0674      0.57888        0.845
 0.239     31       1   0.6768  0.0689      0.55428        0.826
 0.241     30       1   0.6542  0.0702      0.53007        0.807
 0.265     29       1   0.6316  0.0713      0.50621        0.788
 0.284     28       1   0.6091  0.0723      0.48270        0.769
 0.318     27       1   0.5865  0.0730      0.45951        0.749
 0.338     26       1   0.5640  0.0736      0.43665        0.728
 0.345     25       1   0.5414  0.0741      0.41409        0.708
 0.350     24       1   0.5188  0.0743      0.39184        0.687
 0.351     23       1   0.4963  0.0744      0.36988        0.666
 0.450     21       1   0.4727  0.0745      0.34697        0.644
 0.466     20       1   0.4490  0.0745      0.32441        0.622
 0.478     19       1   0.4254  0.0742      0.30220        0.599
 0.499     18       1   0.4018  0.0738      0.28035        0.576
 0.514     17       1   0.3781  0.0731      0.25886        0.552
 0.515     16       1   0.3545  0.0723      0.23774        0.529
 0.634     15       1   0.3309  0.0712      0.21701        0.504
 0.758     13       1   0.3054  0.0701      0.19473        0.479
 0.864     10       1   0.2749  0.0694      0.16752        0.451
 0.977      8       1   0.2405  0.0687      0.13736        0.421
 1.024      7       1   0.2061  0.0670      0.10907        0.390
 1.027      6       1   0.1718  0.0640      0.08277        0.357
 1.068      5       1   0.1374  0.0597      0.05864        0.322
 1.172      4       1   0.1031  0.0538      0.03708        0.287
 1.188      3       1   0.0687  0.0455      0.01876        0.252
 1.601      2       1   0.0344  0.0333      0.00514        0.230
 1.836      1       1   0.0000     NaN           NA           NA
>
> # (b) S-hat(0:062) = 0.8969
>
> # (c) and (d)
> # p-hat     1        2        3        4
>          48/50 * 46/47 * 43/44 * 42/43
[1] 0.8968665
>
>
> # (e) This way of doing it requires you to realize sumkm is a list.
> #     The numbers could also be entered by hand.
> #     Try sumkm[1], sumkm[2] etc. to find out.
> n = sumkm$n.risk; d = sumkm$n.event; Shat = sumkm$surv
> Shat[4] # Another way to answer (b)
[1] 0.8968665
>
> se4 = Shat[4] * sqrt(sum(d[1:4]/(n[1:4]*(n[1:4]-d[1:4])))); se4 # 0.04373656
[1] 0.04373656
>
> # This should agree with the hand" calculation. Get 0.04373819
> 0.8969 * sqrt(sum(d[1:4]/(n[1:4]*(n[1:4]-d[1:4]))))
```

```
[1] 0.04373819
>
> # (f)
> plot(km)
>
>
> # 12
> # (a)
> # First, from last week (Assignment 5),
>
> # A5 Q2a) MLE
> lambdahat = sum(Uncensored)/sum(Time);  lambdahat
[1] 1.717107
>
> # A5 Q2b Estimated asymptotic variance
> vhat = lambdahat^2 / sum(Uncensored);  vhat # Estimated asymptotic variance
[1] 0.07371138
> se = sqrt(vhat); se
[1] 0.2714984
>
> # A5 Q2c) 95% CI for lambda
> lower95 = lambdahat - 1.96*se; upper95 = lambdahat + 1.96*se
> c(lower95,upper95)
[1] 1.184970 2.249244
>
> # Now get to 12a from A6
> # Median of an exponential is log(2)/lambda
> medhat = log(2)/lambdahat; medhat
[1] 0.4036716
>
> # There are two ways to get a CI ...
>
>
> # (b)  Add MLE of S(t) to plot
> t = seq(from=0,to=1.8,length=101)
> Shat = exp(-lambdahat*t)
> lines(t,Shat)
> title('Kaplan-Meier and MLE (MLE is smooth)')
>
>
>
>
>
```

Question 2

Kaplan-Meier and MLE (MLE is smooth)