ST227 Mock Exam Solution

Viet Dang

27/04/2021

Question 1.

Part 1.

First, let us define the mortality function. One naive definition maybe:

```
lambda = 0.1
mu <- function(t){lambda}
```

However, we will need `mu` to be a vectorised function for later steps. That is, the length of the output needs to match that of the input. The simplest way to do it is by replicating `lambda` for a number of n times, where n is the length of the input vector `t`.


```
mu <- function(t){rep(lambda,times=length(t))}
```

Now, we can start defining the survival probability. For this part, it is not critical that ${}_tp_x$ is vectorised, since we are not feeding it into the `integrate` routine.


```
tpx <- function(t,x){
  exp(-integrate(mu,lower=x,upper=x+t)$value)
}
#You can also implement the vectorised version here if you wish
tpx <- function(t,x){
  sapply(
    t,
    FUN = function(t){
      exp(-integrate(mu,lower=x,upper=x+t)$value)
    }
  )
}
```

For a 15-year-old mechanical system, the chance of it surviving the next 5 years is:

```
tpx(t=5,x=15)
```

```
## [1] 0.6065307
```

Part 2.

We consider an updated mortality function with an iterated logarithm term:

$$\tilde{\mu} = \lambda + \gamma \log\log((e+t)), \quad \lambda = 0.1, \gamma = 1.5.$$

Please note that I accidentally uploaded a version on moodle with $\log(\log(t))$. This was an oversight from me, as $\log\log(t)$ can be undefined for small t close to 0. The method is still the same, though.

In order to define the density, we need to first define the survival function.

```
lambda <- 0.1; gamma <- 1.5
mu <- function(t){lambda + gamma*log(log(exp(1) + t))}
tpx <- function(t,x){
  sapply(
    t,
    FUN = function(t){
      exp(-integrate(mu,lower=x,upper=x+t)$value)
    }
  )
}
density <- function(t){
  tpx(t=t,x=15)*mu(t+15)
}
```


For part (b), the expected remaining lifetime is:

```
integrand <- function(t){t*density(t)}
integrate(integrand,lower=0,upper=100)
```



```
## 0.5881207 with absolute error < 2.9e-06
```

The cumulative distribution function ${}_tq_x$ is simply $1 -$ survival function:

```
tqx <- function(t,x){
  1-tpx(t=t,x=x)
}
```


Part (c) asks you to *discuss* how you would find the 95$^{\text{th}}$ percentile - but not to actually solve it. You should give some direction and not simply state some off-the-shelf solution (e.g. external packages or libraries). An approprimate method is *interval bisection*. Note that in this case, the cdf curve will cut through the target level 0.95, so we do not have to worry about the interval bisection method missing out the tangential solution.

Question 2.

For such a small data set, we can simply type it in:

```
lifetimes<-c(64,75,29,45,67,65,77,90,65,55,80,67,72,46,64,28,68,75,49,94)
```

Part 1.

Its first two moments are given by:

$$\mathbb{E}(X) = \frac{\alpha}{\beta}, \quad \text{Var}(X) = \frac{\alpha}{\beta^2}.$$

Then:

$$\frac{\mathbb{E}(X)}{\text{Var}(X)} = \frac{\alpha/\beta}{\alpha/\beta^2} = \beta; \quad \alpha = \beta\mathbb{E}(X) = \frac{\mathbb{E}(X)^2}{\text{Var}(X)}$$

Therefore, if we denote $m_1 = \frac{\sum_i X_i}{n}$ and $m_2 = \frac{\sum_i X_i^2}{n}$, then the method of moment estimators are:

$$\hat{\alpha} = \frac{m_1^2}{m_2 - m_1^2}, \quad \hat{\beta} = \frac{m_1}{m_2 - m_1^2}.$$

We are now ready to define the the negative log likelihood and start the `optim` routine.

```
nLL <- function(params, x){
  alpha <- params[1]; beta <- params[2]
  - sum(
    dgamma(x,shape=alpha,rate=beta,log=TRUE)
```

```
  )
}
alphaMM <- mean(lifetimes)^2/(mean(lifetimes^2)-mean(lifetimes)^2)
betaMM <- mean(lifetimes)/(mean(lifetimes^2)-mean(lifetimes)^2)
optim(par=c(alphaMM,betaMM),nLL, x = lifetimes)
```

```
## Warning in dgamma(x, shape = alpha, rate = beta, log = TRUE): NaNs produced

## Warning in dgamma(x, shape = alpha, rate = beta, log = TRUE): NaNs produced

## Warning in dgamma(x, shape = alpha, rate = beta, log = TRUE): NaNs produced
```

```
## $par
## [1] 11.407096  0.178913
##
## $value
## [1] 86.53808
##
## $counts
## function gradient
##       59       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```


Part 2.

The survival function and thus density are derived as follows:

$$
\begin{aligned}
{}_tp_0 &= \exp\Big(-\int_0^t \alpha\lambda^\alpha s^{\alpha-1} dt\Big) && (1)\\
&= \exp\Big(-\lambda^\alpha s^\alpha\Big|_{s=0}^{s=t}\Big) && (2)\\
&= \exp\Big(-\lambda^\alpha t^\alpha\Big). && (3)\\
f(t) &= \mu(t) \times {}_tp_0 && (4)\\
&= \alpha\lambda^\alpha t^{\alpha-1} \exp\Big(-\lambda^\alpha t^\alpha\Big). && (5)
\end{aligned}
$$

The joint likelihood is:

$$
\begin{aligned}
L(\lambda,\alpha|t) &= \prod_{i=1}^n \alpha\lambda^\alpha t_i^{\alpha-1} \exp\Big(-\lambda^\alpha t_i^\alpha\Big) && (6)\\
&= \alpha^n\lambda^{n\alpha}\Big(\prod_{i=1}^n t_i^{\alpha-1}\Big) \exp\Big(-\sum_{i=1}^n \lambda^\alpha t_i^\alpha\Big). && (7)\\
\ell(\lambda,\alpha|t) &= n\log(\alpha) + n\alpha\log(\lambda) + (\alpha-1)\sum_{i=1}^n \log(t_i) - \sum_{i=1}^n \lambda^\alpha t_i^\alpha. && (8)
\end{aligned}
$$

The question now is how to choose a good initial value for the optimisation algorithm. We can observe when $\alpha = 1$, this reduces to an exponential model. We can use this sub-model as a starting point:

$$\hat{\alpha}^{\text{ini}} = 1, \hat{\lambda}^{\text{ini}} = \frac{1}{\bar{t}}.$$

```
nLL <- function(param, x){
  alpha <- param[1]
  lambda <- param[2]
  n <- length(x)
  - n*log(alpha) - n*alpha*log(lambda) -
    (alpha - 1)*sum(log(x)) +
    sum(lambda^alpha*x^alpha)
}
optim(par = c(1,1/mean(lifetimes)), fn=nLL, x = lifetimes)
```

```
## Warning in log(lambda): NaNs produced

## Warning in log(lambda): NaNs produced

## Warning in log(lambda): NaNs produced

## $par
## [1] 4.39782558 0.01421481
##
## $value
## [1] 84.77866
##
## $counts
## function gradient
##       79       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```


Question 3.

Again, the first task is to import the data:

```
cancer <- readxl::read_excel("C:/Users/Viet Dang/Dropbox/LSE 2021 Lent Term ST227/Mock/mockExamData_can
cancer <- as.data.frame(cancer)
```

The calculation follows pretty much verbatim to the workshop slides.

```
cancer$atRisk <- nrow(cancer):1
#filter the fully observed rows only
cancerObs <- cancer[cancer$fullyObserved, ]
cancerObs$death <- 1
cancerObs$survProb <- (cancerObs$atRisk - cancerObs$death)/cancerObs$atRisk
cancerObs$KM <- cumprod(cancerObs$survProb)
cancerObs$KM
```

```
##  [1] 0.9600000 0.9200000 0.8800000 0.8400000 0.8000000 0.7529412 0.7027451
##  [8] 0.6525490 0.5932264 0.5339037 0.4671658 0.4004278 0.3336898 0.2669519
```

```
## [15] 0.2002139 0.1334759 0.0000000
```

And here's the Greenwood Variance:

```
cancerObs$greenwoodVar <-
  cancerObs$KM^2*cumsum(
    cancerObs$death/(cancerObs$atRisk*(cancerObs$atRisk-cancerObs$death))
  )
cancerObs$greenwoodVar
```

```
##  [1] 0.001536000 0.002944000 0.004224000 0.005376000 0.006400000 0.007753470
##  [7] 0.009105804 0.010191105 0.011621651 0.012580795 0.013529383 0.013757632
## [13] 0.013265541 0.012053111 0.010120342 0.007467234         NaN
```

Question 4.

Note that for this question we need the fully data set, not just the fully observed subset of it.

```
library(survival)
```



```
##
## Attaching package: 'survival'
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     cancer
```



```
survivalObject <- Surv(cancer$time,cancer$event)
coxmodel <- coxph(survivalObject ~ sex, data = cancer)
summary(coxmodel)
```



```
## Call:
## coxph(formula = survivalObject ~ sex, data = cancer)
##
##   n= 25, number of events= 17
##
##       coef exp(coef) se(coef)     z Pr(>|z|)
## sex 0.1526    1.1649   0.5251 0.291    0.771
##
##     exp(coef) exp(-coef) lower .95 upper .95
## sex     1.165     0.8584    0.4162      3.26
##
## Concordance= 0.48  (se = 0.077 )
## Likelihood ratio test= 0.09  on 1 df,   p=0.8
## Wald test            = 0.08  on 1 df,   p=0.8
## Score (logrank) test = 0.08  on 1 df,   p=0.8
```


The Cox Proportional Hazard model has only one parameter, β, which has a point estimate value of 0.1526. With the p-value being approximately 0.80 at all tests, we do not reject the null hypothesis at any reasonable significance level (be careful, one never **accepts** the null hypothesis).