PH C240B: HW 2

Yue You

Problem 1

Consider the following data generating system for (T, C), a random person's survival time and censoring time in days. T follows an exponential distribution with $\lambda = 0.01$ and C follows a weibull distribution with shape and scale parameters, 4 and 100. Simulate 1000 random draws of $n = 1000$ from this distribution and for each draw compute the Kaplan-Meier estimator for survival curve at times 50, 60 and 70, where we consider the study ending at time 100. Form 95% simultaneous confidence bands from each draw and check simultaneous coverage of the true survival. Use the influence curve for the KM estimator to obtain the CI's for each draw. Report the coverage percentage. Note, use foreach for this as shown in lab3sol.Rnw on bcourses to save some time.

```r
S0 = function(t) 1-pexp(t,.01)

sim.cov = function(n) {
  T = rexp(n, .01)
  C = rweibull(n, 4, scale = 100)

  # record the observed right censored data
  Ttilde = pmin(T,C)
  Delta = T <= C & T < 100

  # do a KM fit
  km = survfit(Surv(Ttilde, Delta, type = "right") ~ 1, type = "kaplan-meier",
               conf.int = .95)

  # make step functions
  km_step <- stepfun(km$time, c(1, km$surv))
  km_lower <- stepfun(km$time, c(1, km$lower))
  km_upper <- stepfun(km$time, c(1, km$upper))

  # order the times and corresponding deltas
  T_ord = Ttilde[order(Ttilde)]
  D_ord = Delta[order(Ttilde)]


  ##### CHANGES MADE ######
  # Get all known death times in order and compute the hazard at such times
  # lambda = km$n.event[D_ord]/km$n.risk[D_ord]
  T_death = T_ord[D_ord]
  nrisk = vapply(T_death, FUN = function(x) sum(Ttilde >= x), FUN.VALUE = 1)
  lambda = 1/nrisk
  #########################

  # compute the prob of surviving up to or past each death time
  Pbar = vapply(T_death, FUN = function(time) mean(Ttilde >= time), FUN.VALUE = 1)

  IC_time = function(Ttilde, Delta, time, n) {
    sum((-(T_death <= time)*km_step(time)/(1 - lambda))*
          lambda^2*(1 - lambda)^2/((1 - lambda)^2/n + lambda^2*Pbar)*
          ((Ttilde == T_death & Delta == 1)/lambda - (Ttilde > T_death)/(1 - lambda)))
  }

  # compute the IC matrix with each col an IC
  IC = vapply(c(50,60,70),FUN = function(t) {
    unlist(lapply(1:n, FUN = function(i) {
      IC_time(Ttilde[i], Delta[i], t, n)
    }))
  }, FUN.VALUE = rep(1,n))


  # COMPUTE THE CORRELATION and draw the random three-d normals
  Sigma = cor(IC)
  z = rmvnorm(1e6, c(0,0,0), Sigma)

  # compute the max abs val of each of the 3-d normals then choose the
  # 95th quantile of that vector, which is the simultaneous number of SE's
  z_abs = apply(z, 1, FUN = function(row) max(abs(row)))
  SE_num = quantile(z_abs, .95)

  # Note how the CI is wider when demanding simultaneous coverage
  Cov50 = km_lower(50) <= S0(50) & km_upper(50) >= S0(50)
  Cov60 = km_lower(60) <= S0(60) & km_upper(60) >= S0(60)
  Cov70 = km_lower(70) <= S0(70) & km_upper(70) >= S0(70)

  indy_cov = all(Cov50, Cov60, Cov70)

  # simultaneous for 40, 50, 60
  Sim50 = km_step(50) - SE_num*sd(IC[,1])*sqrt(n-1)/n <= S0(50) &
    S0(50) <= km_step(50) + SE_num*sd(IC[,1])*sqrt(n-1)/n
  Sim60 = km_step(60) - SE_num*sd(IC[,2])*sqrt(n-1)/n <= S0(60) &
    S0(60) <= km_step(60) + SE_num*sd(IC[,2])*sqrt(n-1)/n
  Sim70 = km_step(70) - SE_num*sd(IC[,3])*sqrt(n-1)/n <= S0(70) &
    S0(70) <= km_step(70) + SE_num*sd(IC[,3])*sqrt(n-1)/n

  sim_cov = all(Sim50, Sim60, Sim70)

  return(c(indy_cov = indy_cov, sim_cov = sim_cov))
}

sim.cov(1000)
```

```
## indy_cov  sim_cov
##     TRUE     TRUE
```

```r
registerDoParallel(cores = detectCores())
getDoParWorkers()
```

```
## [1] 4
```

```r
B = 1000
n = 1000
ALL = foreach(i=1:B,.packages=c("mvtnorm"), .errorhandling = "remove") %dopar% {sim.cov(n)}
res = do.call(rbind, ALL)
```

```
colMeans(res)
```

```
## indy_cov  sim_cov
##    0.918    0.947
```

We calculated 'indy_cov' based on the combination of individually based confidence intervals (i.e. a confidence for *one* time point) and we see that, even though each confidence interval probably covers it's correspond truth 95% of the time, these individual confidence intervals only cover the truth simultaneously (i.e. across our set of time points) roughly 91% of the time. Alternatively, we calculated 'sim_cov' based on a 95% simultaneous confidence interval. A simultaneous confidence interval is constructed according to a *set* of points and simultaneous confidence intervals stem from a multivariate normal distribution with mean equal to 0 and variance equal to the correlation matrix of the covariance matrix for the set. They are also wider than individually based confidence intervals because they need to cover a set of points opposed to a singular point. For a simutaneous confidence interval, we expect all intervals to contain their corresponding truth with 95% confidence and, indeed, we see that this is the case as the 'sim_cov' is exactly 95%!