R codes for \Competing risks regression models with covariates-adjusted censoring weight under the generalized case-cohort design"

By Soyoung Kim and Yayun Xu

There are three R functions including weight functions, beta estimations, and variance estimations for competing risks regression under generalized case-cohort studies.

```
###weight functions
find.weightcc = function(gamma, disease, v, delta , choice = 1,eta) {
  weightcc = c()
  if (choice == 1){#weight function for sigle case-cohort study
    for (i in unique(v)) {
      num = sum(gamma[v == i] * (1 - (disease[v == i] == 1)))
      deno = sum((1 - (disease[v == i] == 1)))
      deno = deno + sum(q[v == i] * (disease[v == i] == 1))
    }else{
    q.est = numq / denoq }
    weightcc[v==i] = (1 - (disease[v==i] == 1)) * gamma[v==i] / alpha.est +
      (disease[v==i]==1)*(gamma[v==i]+ (1-gamma[v==i])*(eta[v==i] == 1))/
      (alpha.est+ (1-alpha.est)*q.est)
  }
else if (choice == 3) { # efficient weight function for multiple case-cohort studies
```

```
  for (i in unique(v)) {
    num = sum(gamma[v == i] * (1 - delta[v == i]))
    deno = sum(1 - delta[v == i])
    deno = deno + (deno == 0)
    alpha.est = num / deno
    alpha.est = alpha.est + (alpha.est == 0)

    denoq1 = sum(gamma[v==i] == 0 &disease[v==i] == 1)
    numq1 = sum(eta[v==i] == 1)
    if (denoq1 == 0){ q1.est = 1
    }else{
    q1.est = numq1 / denoq1 }

    denoq2 = sum(gamma[v==i] == 0 &disease[v==i] == 2)
    numq2 = sum(eta[v==i] == 2)
    if (denoq2 == 0){ q2.est = 1
    }else{
    q2.est = numq2 / denoq2 }
    weightcc[v==i] = (disease[v==i] != 0) * gamma[v==i] + (disease[v==i] == 0)
     * gamma[v==i] / alpha.est +(eta[v==i] == 1) / q1.est +(eta[v==i] == 2) / q2.est
  }
 }else if (choice ==5){ #optimal weight function for multiple case-cohort studies
  for (i in unique(v)){
    deno = sum(disease[v==i] == 0)
    num = sum(gamma[v==i] == 1 &disease[v==i] == 0)
    alpha.est = num / deno
    denoq1 = sum(gamma[v==i] == 0 & disease[v==i] == 1)
    numq1 = sum(eta[v==i] == 1)
    if (denoq1 == 0){ q1.est = 1
    }else{
    q1.est = numq1 / denoq1 }
    denoq2 = sum(gamma[v==i] == 0 & disease[v==i] == 2)
    numq2 = sum(eta[v==i] == 2)
    if (denoq2 == 0){ q2.est = 1
    }else{
      q2.est = numq2 / denoq2 }

      weightcc[v==i] = (disease[v==i] ==1)
      * gamma[v==i]*(1/(alpha.est+(1-alpha.est)*q1.est)) +
      (disease[v==i] ==2) * gamma[v==i]*(1/(alpha.est+(1-alpha.est)*q2.est)) +
      (disease[v==i] == 0) * gamma[v==i] / alpha.est +
      (eta[v==i] == 1) * (1/(alpha.est+(1-alpha.est)*q1.est)) +
      (eta[v==i] == 2) * (1/(alpha.est+(1-alpha.est)*q2.est))
  }
 }

 return(weightcc)

}

###beta estimation function
# choice: weight functions for case-cohort studies
#           1: weight function for single case-cohort study
#           2: optimal weight function for single case-cohort study
# alpha.est + (1-alpha.est)*q1.est
#           3: weight function for multiple
```

```
#           2: covariate-unadjusted weights
# data
# time: observed time
# delta: failarue indicator
# disease: cause types (1: cause 1, 2: cause 2, 0: censored)
# z: covariate
# gamma: subcohort indicator
# v: stratum variable
# eta: case indicator
beta.est = function(data, beta = 0, delta0 = 3, choice = 1,choice.c=1) {

  n = nrow(data)
  time = data[, 2]
  delta = data[, 3]
  disease = data[, 4]
  z = data$Z
  gamma = data$gamma
  v = data$v
  eta = data$eta
  npop = n
  fail = time[!(disease == 0)]
  L = length(fail)
  gg1 = (matrix(rep(time, L), npop, L) <
  matrix(rep(t(fail), each = npop), npop, L)) * 1
  gg1[disease != 1, ] = 0
  gg2 = matrix(1, npop, L)

  for (i in 1:npop) {
    if (delta[i] == 0) {
      for (j in 1:L) {
        if (time[i] < fail[j])
          gg2[i, j] = 0
      }
    }
  }
  Yr = gg2 - gg1
  weight.cc = find.weightcc(gamma, disease ,v, delta ,choice=choice ,eta)
  Gcweight = list()
  faili <- time[disease == 1]
  Li <- length(faili)
  censor<-1-delta
  if (choice.c == 1){ ##COX estimates for censoring distribution
    for (i in unique(v)) {
      i.name = paste(i)
      data.cc<-data.frame(time[v == i],censor[v == i],z[v == i],weight.cc[v == i] )
      data.cc1<-data.cc[data.cc$weight.cc>0,]
      colnames(data.cc1) <- c("time", "censor", "z", "weight.cc")
      ss <- Surv(data.cc1$time, data.cc1$censor)
      fit.cox<-survival::coxph(ss ~ z, weight= weight.cc, data=data.cc1)
      Gc.0 = survfit(fit.cox,newdata=data.frame(z=0))
      Gc.1 = survfit(fit.cox,newdata=data.frame(z=1))

      kmest.0 = stepfun(Gc.0$time, c(1,Gc.0$surv))
      kmest.1 = stepfun(Gc.1$time, c(1,Gc.1$surv))

      gg3 = (matrix(rep(time,Li),npop,Li) >= matrix(rep(t(faili),each=npop),npop,Li))
      foo.0 = matrix(rep(t(kmest.0(faili)),each=npop),npop,Li)
      /(matrix(rep(kmest.0(time),Li),npop,Li))
```

```
foo.1 = matrix(rep(t(kmest.1(faili)),each=npop),npop,Li)
```

```
Sl2 = t(z2expz) %□%
```

```
    for (j in 1:n) f
      if (time[i] < time[j])
        rt[i, j] = 0
    g
  g
g
Yrt = rt - Nrt
phit = find.weightcc(gamma
```

```
    foo = matrix(rep(t(kmest(time[v == i])), each = n), n_l, n) / fff
    foo[gg3] = 1
    Gcweight[[i.name]] = foo
  g
g

wY.all = Yrt

dNr.all = (do.call(cbind, replicate(n, time, simplify = FALSE))
 == do.call(rbind, replicate(n, time, simplify = FALSE))) 1
dNr.all[disease != 1, ] = 0
wdN.all = dNr.all

p = 1
Sl = 0
Sl0 = list()
Ec = list()

for (i in unique(v)) f
  i.name <- paste(i)
  dis1 = (disease[disease == 1])  1
  v1 = (v[disease == 1])  1
  nl = sum(v == i)
  expz  <- exp(z[v == i]  beta)
  zexpz <- expz  z [v == i]
  temp0  <-colSums(expz  (phit[v == i]  wY.all[v == i, ]  Gcweight[[i.name]]))
  Sl0[[i.name]] <- temp0 + (temp0 == 0)
  Sl1 <-colSums (zexpz  (phit[v == i]  wY.all[v == i , ]  Gcweight[[i.name]]))
  z2expz = expz  z[v == i]  z[v == i]
  Sl2 = colSums(z2expz  (phit[v == i]  wY.all[v == i , ]  Gcweight[[i.name]]))
  Sl1overSl0hat <- Sl1 / Sl0[[i.name]]
  Sl2overSl0hat = Sl2 / Sl0[[i.name]]
  Ec[[i.name]] = Sl1overSl0hat

  ipart1 = do.call(rbind, replicate(nl, Sl2overSl0hat, simplify = F))
   wdN.all[v == i,]  Gcweight[[i.name]]phit[v == i]
  Ipartlhat = sum(ipart1)

  E2 = Sl1overSl0hat  Sl1overSl0hat
  ipart2 = do.call(rbind, replicate(nl, E2, simplify = F))
  wdN.all[v == i, ]  Gcweight[[i.name]]phit[v == i]
  IpartIIhat = sum(ipart2)

  Sl <- Sl + (Ipartlhat - IpartIIhat)
g
tempI = Sl
Yt = (do.call(cbind,replicate(n,time,simplify = FALSE))
>= do.call(rbind,replicate(n,time,simplify = FALSE)))1
Nct = (do.call(cbind,replicate(n,time,simplify = FALSE))
<= do.call(rbind,replicate(n,time,simplify = FALSE)))1
Nct[disease!=0, ] = 0

var.1 = 0
expz = exp(z  beta)
for (i in unique(v)) f
  i.name = paste(i)
  nl = sum(v == i)
  dlamb10t = colSums(wdN.all[v == i,]  Gcweight[[i.name]] phit[v == i] )
```

```
/ Sl0[[i.name]]
zminEc = (do.call(cbind, replicate(n, z[v == i], simplify = FALSE)))
 - (do.call(rbind, replicate(sum(v ==i), Ec[[i.name]], simplify = FALSE)))

wdM = wdN.all[v == i,] □ Gcweight[[i.name]] - wY.all[v == i,]
□Gcweight[[i.name]] □ expz[v == i] □ (do.call(rbind, replicate(sum(v ==i),
 dlamb10t, simplify = FALSE)))
eta11 = rowSums(zminEc □ wdM)
eta21 = c()
if (choice.c == 1)f
  WeightedY<-phit[v==i]□Yt[v==i,]
  gamma.hat<-fit.cox$coef
  expg<-exp(z[v==i]□gamma.hat)
  zexpg<-z[v==i]□expg
  s0c<- colSums( WeightedY □expg)
  s0c<- s0c + (s0c==0)
  s1c<- colSums( WeightedY □zexpg)
  ecox <-s1c/s0c
  S1overS02hat<- ecox/s0c
  zc_min_ec<- (do.call(cbind,replicate(n,z[v==i],simplify = FALSE)))
   - (do.call(rbind,replicate(nl,ecox,simplify = FALSE)))
  ilc_mat<-vcov(fit.cox)[1]
  censor.w<-
```

```
        deltaw = do.call(cbind, replicate(n, wlict, simplify = F)) -
        do.call(rbind, replicate(nl, wct, simplify = F))
        ind = do.call(cbind, replicate(n, time[v == i], simplify = F))
        < do.call(rbind, replicate(sum(v == i), time, simplify = F))
        eta21 =   c(eta21, sum(deltaw  zminEc  ind  wdM))
      g
    g

    eta.comb = (eta11 + eta21)
    eta.comb2 = eta.comb eta.comb
    qest1 = sum(eta[v==i]==1)/sum(disease[v==i]==1& gamma[v==i]==0)
    alphaest = sum(         [v =(n)-55(E)-56(c)]TJ7F32 9.9701 Tf 35.227 0 Td [(*)]TJ5.87 7.9701 Tf 5.604 0 Td [(()-270(d)-158(i)-158(s)-
[v =(n)-55(E)-56(c)]TJ7F32 9.9701 Tf 35.227 0 Td [(*)]TJ5.87 7.9701 Tf 5.604 0 Td [(()-270(d)-158(i)-158(s)-158(e)-158(a)-158(s)-158(e Td [([)-210(v
```

32 7.9701 Tf 20.055 0 Td [(()]TJ/F53 7.9701 Tf 4.528 0 Td [(r)-43(b)-42(i)-43(n)-43(d)]TJ/F32 7.9701 Tf 26.429 0 Td [(,)-987(r)-178(e)-178(p)-178(l)-179(i)-178(c)-178(a)-178(t)

```
Q = wY.all[v == i,]  Gcweight[[i.name]]  zminEc    expz[v == i]
if  (choice.c ==1 )f
  eta31 = rowSums((1-delta[v == i] )Q do.call(rbind,replicate(
  sum(v == i),dlamb10t,simplify = FALSE)))
  eta32 = (1-delta[v == i])eta21
  eta3<- eta31-eta32
gelsef
  eta3 = rowSums((1-delta[v == i] )Q do.call(rbind,replicate(
  sum(v == i),dlamb10t,simplify = FALSE)))
g
V331 = sum(eta.combeta.comb
```