

```

library(rankreg)

a = read.table("c:\\My Teach\\ST745\\kidney_transplant.txt")

event.time = a[,3]
delta = a[,4]
allo = 2-a[,1]
hodgkins = a[,2]-1
kscore = a[,5]
wtime = a[,6]
x = as.matrix(cbind(allo,hodgkins,kscore,wtime))
y = log(event.time)

####using rankaft function
fit1 = rankaft(x,y,delta)
####output from rankaft
> fit1$beta
      xnewallo xnewhodgkins xnewkscore  xnewwtime
betag 0.3055718    -1.549300 0.06074927 0.01443342  ####Gehan type
estimator
betal 0.3522702    -1.494406 0.07495995 0.01066983  ####log-rank
estimator

####using aft.fun function
aft.fun1 = function (x, y, delta, randomseed = 10, weight = "logrank",
nstep = 3,
      mcsize = 100)
{
  if (any((delta != 0) & (delta != 1)))
    stop("delta must be 0(right-censored) or 1(uncensored)")
  set.seed(randomseed)
  ynew <- 1000 * (length(y))^2
  data1 <- data.frame(y, x)
  options(contrasts = c("contr.treatment", "contr.poly"))
  tempfit <- lm(y ~ ., x = TRUE, y = TRUE, data = data1)
  x <- as.matrix(tempfit$x[, -1])
  xn <- dimnames(x)[[2]]
  y <- tempfit$y
  dimnum <- dim(x)
  n1 <- dimnum[1]
  n2 <- dimnum[2]
  betagw <- betagc <- matrix(0, nrow = mcsize, ncol = n2)
  if (weight == "logrank") {
    nst <- nstep
  }
  else {
    nst <- 0
  }
  betagm <- betalw <- array(0, dim = c(n2, mcsize))
betagc <- betalc <- array(0, dim = c(n2, mcsize))
  covw <- array(0, dim = c(n2, n2, nst + 1))
  residuals <- matrix(0, nrow = n1, ncol = nst + 1)
  yy0 <- rep(y, rep(n1, n1))
  delta1 <- rep(delta, rep(n1, n1))
  yy1 <- rep(y, n1)
  yy <- delta1 * (yy0 - yy1)

```

```

xx0 <- matrix(rep(as.vector(x), rep(n1, n1 * n2))), nrow = n1 *
  n1)
xx1 <- t(matrix(rep(as.vector(t(x)), n1), nrow = n2))
xx <- xx0 - xx1
xxdif <- xx * delta1
xnew <- apply(xxdif, 2, sum)
xnew <- rbind(xxdif, -xnew)
yynew <- c(yy, ynew)
dimnames(xnew) <- list(NULL, xn)
fit <- rq.fit(xnew, yynew, tau = 0.5)
betag <- fit$coef
residn <- fit$resid
residn <- (!(residn > 0))
residn <- residn[-(length(residn))]
betal <- betag
if (weight == "logrank") {
  for (i in 1:nstep) {
    fitted <- x %%% betal
    eb <- y - fitted
    ss0b <- (n1 + 1 - rank(eb))/n1
    ss0b1 <- rep(ss0b, rep(n1, n1))
    xxdif1 <- xxdif/ss0b1
    xnew1 <- apply(xxdif1, 2, sum)
    xnew1 <- rbind(xxdif1, -xnew1)
    yy1 <- c(yy/ss0b1, ynew)
    fit1 <- rq.fit(xnew1, yy1, tau = 0.5)
    betal <- fit1$coef
  }
}
zi <- matrix(rexp(n1 * mcsize), nrow = mcsize, ncol = n1)
for (i in 1:mcsize) {
  zzi <- rep(as.vector(zi[i, ]), rep(n1, n1))
  xm <- xxdif * zzi
  xmnew <- apply(xm, 2, sum)
  xmnew <- rbind(xm, -xmnew)
  ymnew <- c(yy * zzi, ynew)
  fitm <- rq.fit(xmnew, ymnew, tau = 0.5)
  betagm[, i] <- fitm$coef
  betalw[, i] <- betagm[, i]
  betagc[, i] <- betagm[, i] - betag
  if (weight == "logrank") {
    for (j in 1:nstep) {
      fitted <- x %%% betalw[, i]
      eb <- y - fitted
      ss0b <- (n1 + 1 - rank(eb))/n1
      ss0b1 <- rep(ss0b, rep(n1, n1))
      xxdif1 <- xm/ss0b1
      xnew1 <- apply(xxdif1, 2, sum)
      xnew1 <- rbind(xxdif1, -xnew1)
      yy1 <- c(zzi * yy/ss0b1, ynew)
      fitm1 <- rq.fit(xnew1, yy1, tau = 0.5)
      betalw[, i] <- fitm1$coef
    }
    betalc[, i] <- betalw[, i] - betal
  }
}
}
predmatrix <- x - t(matrix(rep(apply(x, 2, mean), n1), ncol = n1))

```

```

covw[, , 1] <- (betagc) %*% t(betagc)/mcsiz
residuals[, 1] <- y - predmatrix %*% as.matrix(betag)
covw[, , 2] <- (betalc) %*% t(betalc)/mcsiz
residuals[, 2] <- y - predmatrix %*% as.matrix(betal)
object <- list(beta = rbind(betag, betal), betacov = covw,
  residuals = residuals, betagm = betagm, betalw = betalw,
  mcsiz = mcsiz, message = fit$message, warning = fit$warning,
  weight = weight)
class(object) <- "AFT"
object
}

fit2 = aft.fun1(x,y,delta)
####output from aft.fun
> fit2$beta
      allo  hodgkins    kscore      wtime
betag 0.3055718 -1.549300 0.06074927 0.01443342 ###Gehan estimator
betal 0.3244511 -1.494337 0.07504969 0.01032466 ###log-rank estimator

> sqrt(apply(fit2$betagm,1,var))
[1] 0.540879687 0.646372329 0.011232689 0.007314381 ###standard error
of Gehan estimator
> sqrt(apply(fit2$betalw,1,var))
[1] 0.694427754 0.821533829 0.010118149 0.006615196 ###standard error
of log-rank estimator

> fit2$beta[1,]/sqrt(apply(fit2$betagm,1,var))
      allo  hodgkins    kscore      wtime
0.5649534 -2.3969163  5.4082572  1.9732936 ###z-statistic of Gehan
estimator
> fit2$beta[2,]/sqrt(apply(fit2$betalw,1,var))
      allo  hodgkins    kscore      wtime
0.4672208 -1.8189603  7.4173339  1.5607492 ###z-statistic of log-rank
estimator

> round(10000*2.0*(1-
pnorm(abs(fit2$beta[1,]/sqrt(apply(fit2$betagm,1,var)))))/10000
      allo hodgkins    kscore      wtime
0.5721  0.0165  0.0000  0.0485 ###p-value of Gehan estimator

> round(10000*2.0*(1-
pnorm(abs(fit2$beta[2,]/sqrt(apply(fit2$betalw,1,var)))))/10000
      allo hodgkins    kscore      wtime
0.6403  0.0689  0.0000  0.1186 ###z-statistic of log-rank
estimator

```