4: Simulation Study

07/07/2020

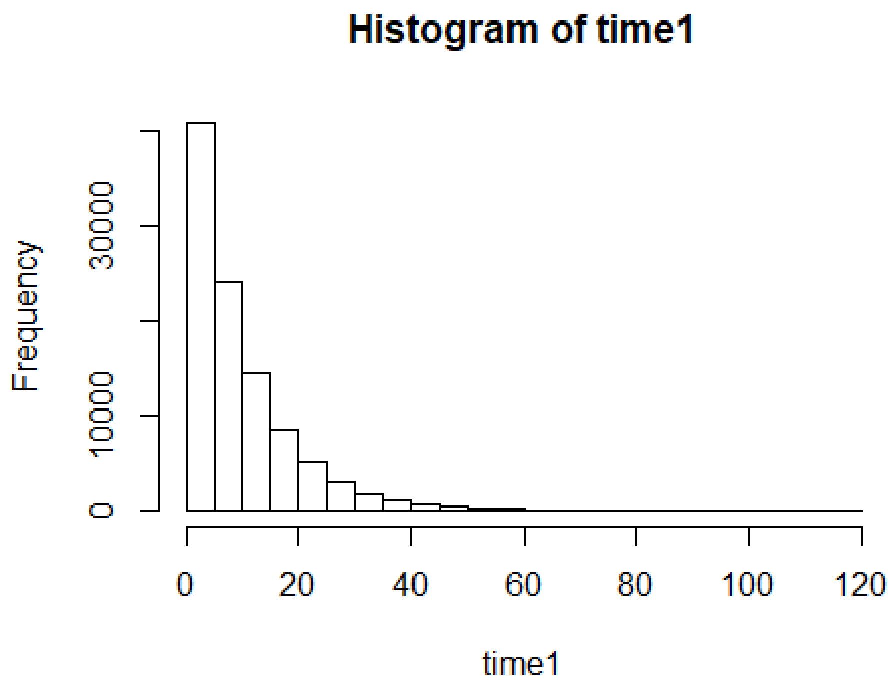
```
# Load packages
library(survival)
library(tidyverse)
library(mstate)
library(cmprsk)
library(gridExtra)
library(pseudo)
```

Data Generation

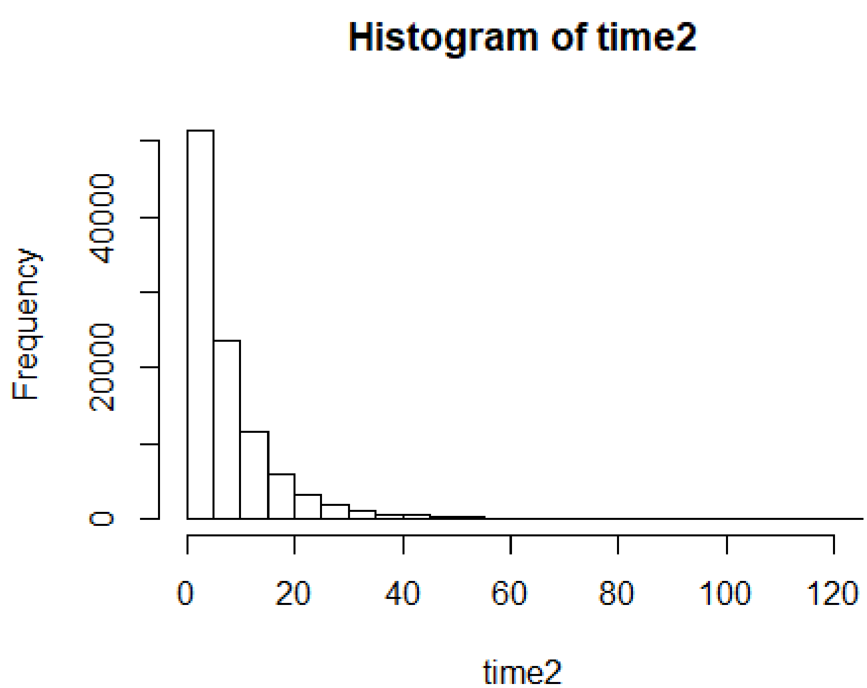
```
set.seed(12345)

# Participants
n <- 100000

### Simulate covariates
gender <- rbinom(n, 1, 0.5)
summary(gender)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  1.0000  0.5008  1.0000  1.0000
```

```
# 0 = female, 1 = male

### Simulate competing risks
# k = 2 causes of failure

Btz1 <- 0.1*gender
Btz2 <- 0.7*gender


# Proportional cs hazards
cshr1 <- 0.1*exp(Btz1)
cshr2 <- 0.1*exp(Btz2)

# Latent failure times
time1 <- rexp(n, rate = cshr1)
time2 <- rexp(n, rate = cshr2)

hist(time1)
```




```
hist(time2)
```




```
summary(time1)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   0.00001   2.75649   6.59839   9.55158  13.23251 115.82263
```

```r
summary(time2)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   0.00007   1.94022   4.78819   7.46882  10.02662 121.16510
```

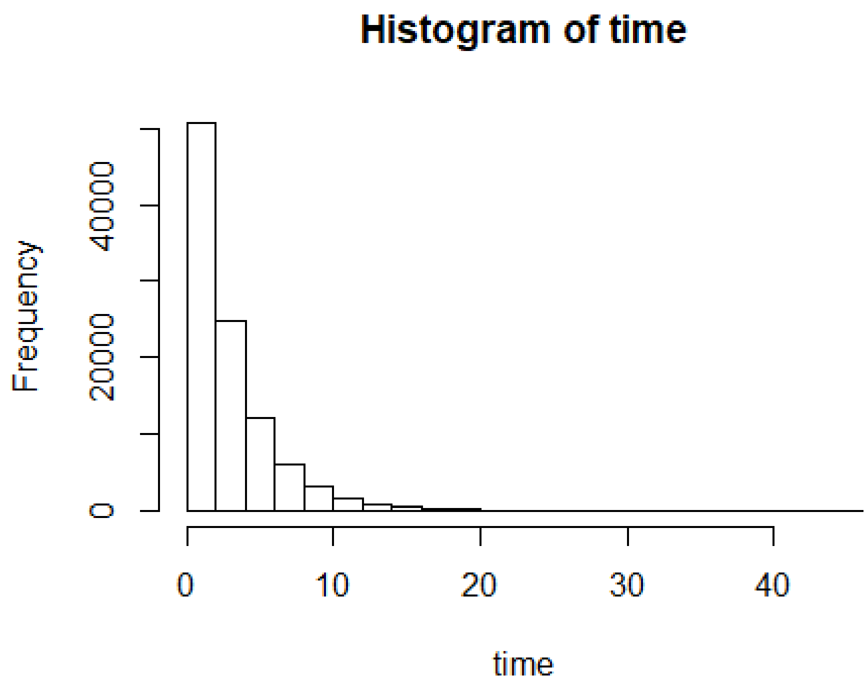
```r
# Status
# 1 if time 1 first, 2 if time 2 is first
epsilon <- 1*(time1<time2) + 2*(time1>time2)


# Calculate the observed times
time <- time1
time[epsilon == 2] <- time2[epsilon == 2]



### Simulate censoring
# Drop out times:
cens <- rexp(n, rate = 0.1)

summary(1*(time > cens)) # Is it censored?
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2904  1.0000  1.0000
```

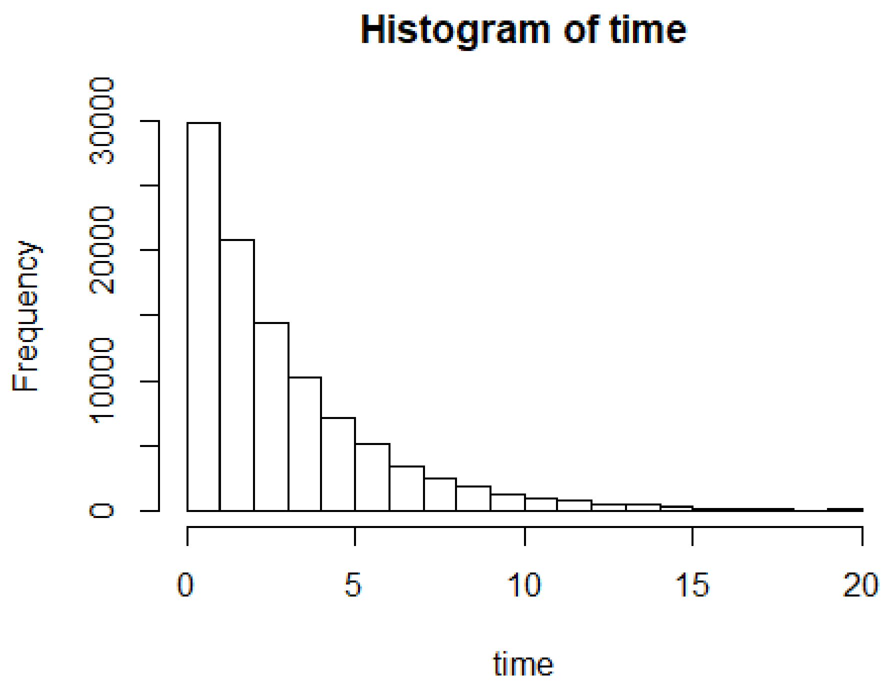
```r
epsilon[time > cens] <- 0
time[time>cens] <- cens[time>cens]


summary(time)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00001  0.80857  1.95829  2.87688  3.94717 45.89500
```

```r
hist(time)
```




```r
# Set max follow up time to 20
tmax <- 20
sum(1*(time > tmax))

## [1] 134

# Censor individuals not dead by tmax
epsilon[time > tmax] <- 0
time[time > tmax] <- tmax

summary(time)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.000006  0.808574  1.958288  2.872574  3.947169 20.000000

hist(time)
```




```
# Generated dataset:
data1 <- data.frame(time, epsilon, gender)

summary(data1)

##        time              epsilon         gender      
##  Min.   : 0.000006   Min.   :0.00   Min.   :0.0000  
##  1st Qu.: 0.808574   1st Qu.:0.00   1st Qu.:0.0000  
##  Median : 1.958288   Median :1.00   Median :1.0000  
##  Mean   : 2.872574   Mean   :1.12   Mean   :0.5008  
##  3rd Qu.: 3.947169   3rd Qu.:2.00   3rd Qu.:1.0000  
##  Max.   :20.000000   Max.   :2.00   Max.   :1.0000

head(data1)

##          time epsilon gender
## 1  0.01376141       0      1
## 2 14.63483371       0      1
## 3  2.45289294       1      1
## 4  2.19906769       2      1
## 5  0.28455770       0      0
## 6  6.80930541       2      0

data1 <- data1 %>% mutate(epsilon = as.integer(epsilon))

causes <- data.frame(epsilon=c(0, 1, 2), cause = c("event-free", "cause1", 
"cause2"))
```

```
data1 <- merge(data1, causes, by = "epsilon")
head(data1)
```

```
##   epsilon        time gender      cause
## 1       0  0.01376141      1 event-free
## 2       0 14.63483371      1 event-free
## 3       0  4.26225636      1 event-free
## 4       0  1.09292492      1 event-free
## 5       0  0.28455770      0 event-free
## 6       0  7.60832017      1 event-free
```

```
table(data1$epsilon)
```

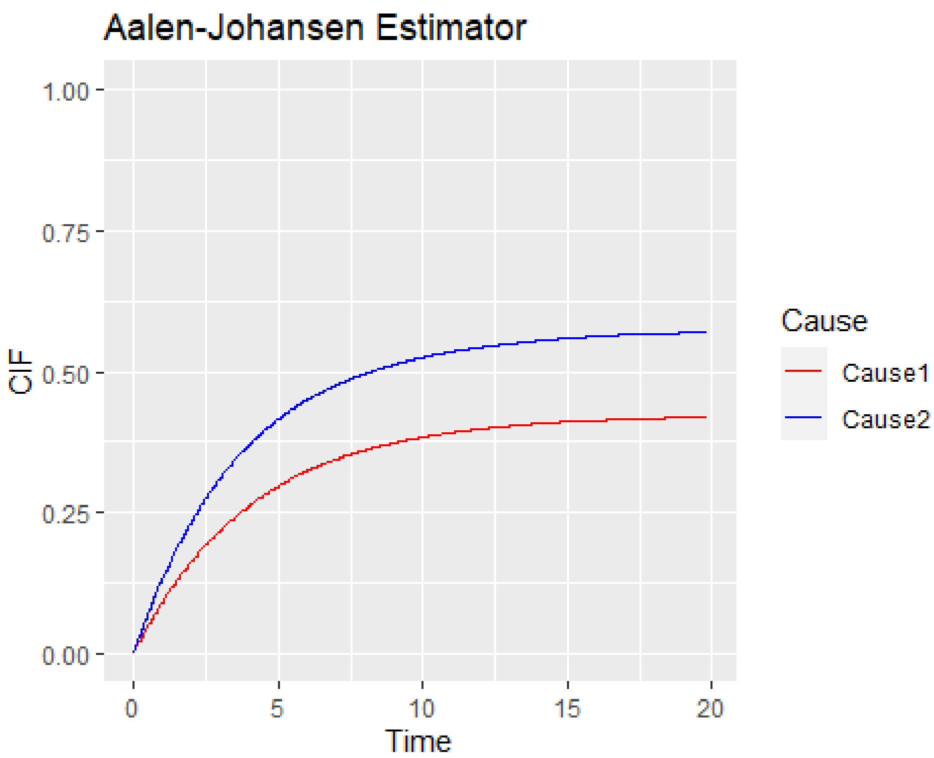
```
##
##     0     1     2
## 29137 29769 41094
```

Non-parametric estimator

```
cif0 <- Cuminc(time = "time", status = "epsilon", data = data1 %>%
select(time, epsilon))

head(cif0)
```

```
##           time    Surv         CI.1         CI.2       seSurv       seCI.1
## 1 6.067408e-06 0.99999 1.00000e-05 0.00000e+00 9.999950e-06 9.999950e-06
## 2 6.653057e-05 0.99998 1.00000e-05 1.00001e-05 1.414206e-05 9.999950e-06
## 3 1.111580e-04 0.99997 2.00003e-05 1.00001e-05 1.732048e-05 1.414221e-05
## 4 1.678483e-04 0.99996 2.00003e-05 2.00005e-05 2.000000e-05 1.414221e-05
## 5 2.512521e-04 0.99995 2.00003e-05 3.00009e-05 2.236066e-05 1.414221e-05
## 6 2.755245e-04 0.99994 3.00007e-05 3.00009e-05 2.449482e-05 1.732065e-05
##         seCI.2
## 1 0.000000e+00
## 2 1.000005e-05
## 3 1.000005e-05
## 4 1.414235e-05
## 5 1.732077e-05
## 6 1.732077e-05
```

```
# Plot
ggplot(cif0) +
      geom_step(aes(x = time, y = CI.1, color = 'Cause1')) +
      geom_step(aes(x = time, y = CI.2, color = 'Cause2'))  +
  labs(title = 'Aalen-Johansen Estimator') + xlab('Time') + ylab('CIF') +
ylim(0,1) +
scale_colour_manual(name="Cause",
    values=c(Cause1="red", Cause2="blue"))
```

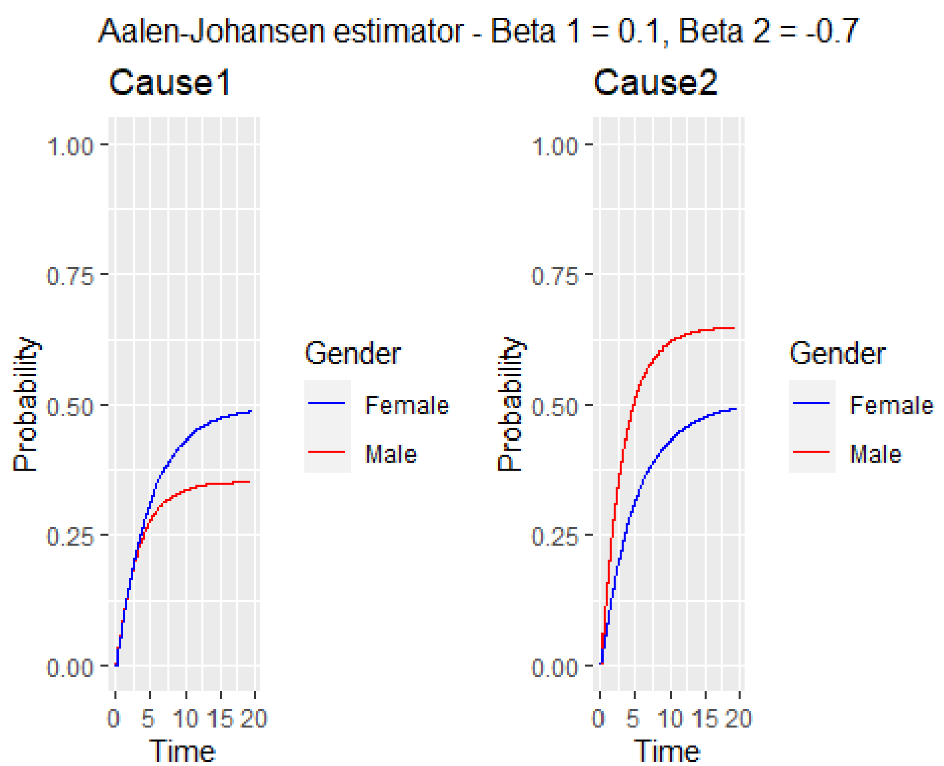



```
## GENDER
cif00 <- Cuminc(data1$time, as.numeric
             (data1$epsilon),
             group = data1$gender)

ci.m <- cif00[cif00$group == 1, ]
ci.f <- cif00[cif00$group == 0, ]
head(ci.f)
```

```
##   group         time      Surv         CI.1         CI.2       seSurv
## 1     0 8.223113e-05 1.0000000 0.000000e+00 0.000000e+00 0.000000e+00
## 2     0 2.755245e-04 0.9999800 2.003446e-05 0.000000e+00 2.003426e-05
## 3     0 3.097719e-04 0.9999599 2.003446e-05 2.003486e-05 2.833272e-05
## 4     0 3.416943e-04 0.9999399 4.006932e-05 2.003486e-05 3.470012e-05
## 5     0 4.984392e-04 0.9999199 4.006932e-05 4.007012e-05 4.006812e-05
## 6     0 5.094291e-04 0.9998998 6.010458e-05 4.007012e-05 4.479725e-05
##         seCI.1       seCI.2
## 1 0.000000e+00 0.000000e+00
## 2 2.003426e-05 0.000000e+00
## 3 2.003426e-05 2.003466e-05
## 4 2.833272e-05 2.003466e-05
## 5 2.833272e-05 2.833329e-05
## 6 3.470035e-05 2.833329e-05
```

```
#Plot
p11 <- ggplot(data = NULL, aes(x= time, y = CI.1)) +
      geom_step(data = ci.m, aes(color = 'Male')) +
```

```
      geom_step(data = ci.f, aes(color = 'Female')) +
  labs(title = 'Cause1') + xlab('Time') + ylab('Probability') + ylim(0, 1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

p22 <- ggplot(data = NULL, aes(x= time, y = CI.2)) +
      geom_step(data = ci.m, aes(color = 'Male')) +
      geom_step(data = ci.f, aes(color = 'Female')) +
  labs(title = 'Cause2') + xlab('Time') + ylab('Probability') + ylim(0, 1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

grid.arrange(p11, p22, nrow=1, ncol=2, top = "Aalen-Johansen estimator - Beta
1 = 0.1, Beta 2 = -0.7")
```



Cox model

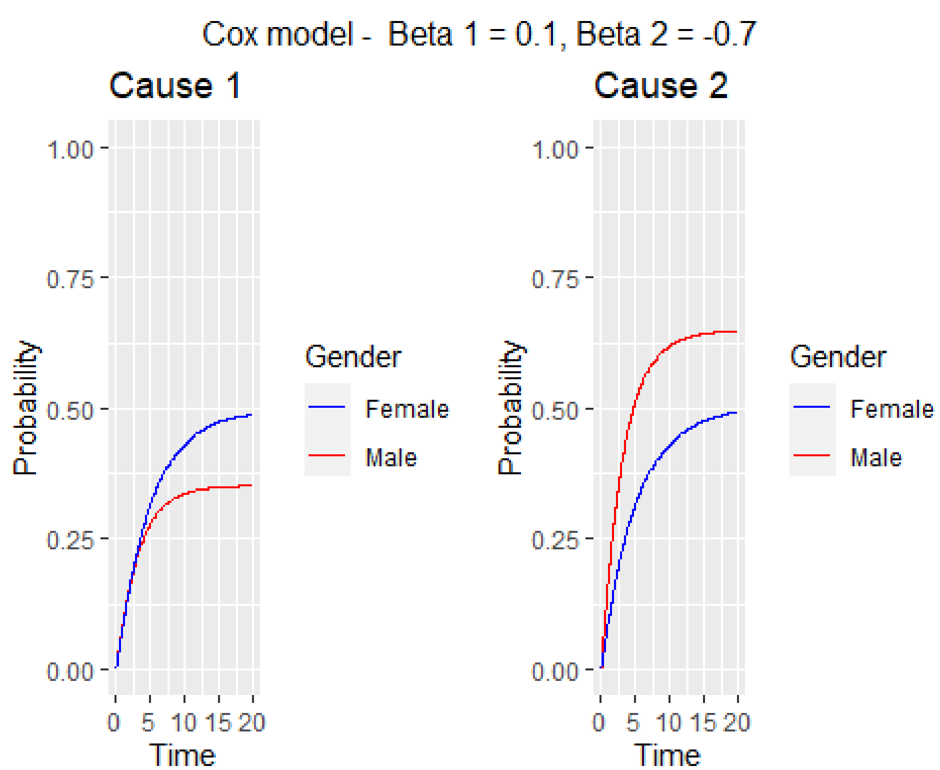
The code below on obtaining Cox model estimates using the `mstate` package is based on a tutorial by H.Putter [1].

```
# Competing risk transition matrix
tmat <- trans.comprisk(2, names = c("event-free", "cause1", "cause2"))

data_wide <- data1
# Indicator columns for each of the 2 causes of deaths
data_wide$stat1 <- as.numeric(data_wide$epsilon == 1)
```

```r
data_wide$stat2 <- as.numeric(data_wide$epsilon == 2)

head(data_wide)

##   epsilon        time gender      cause stat1 stat2
## 1       0  0.01376141      1 event-free     0     0
## 2       0 14.63483371      1 event-free     0     0
## 3       0  4.26225636      1 event-free     0     0
## 4       0  1.09292492      1 event-free     0     0
## 5       0  0.28455770      0 event-free     0     0
## 6       0  7.60832017      1 event-free     0     0

# Convert data into long format using msprep:
data_long <- msprep(time = c(NA, "time", "time"), status = c(NA, "stat1",
"stat2"), data = data_wide, keep = c("gender"), trans = tmat)

tail(data_long)

## An object of class 'msdata'
##
## Data:
##            id from to trans Tstart     Tstop      time status gender
## 199995  99998    1  2     1      0 2.2409850 2.2409850      0      1
## 199996  99998    1  3     2      0 2.2409850 2.2409850      1      1
## 199997  99999    1  2     1      0 0.2821291 0.2821291      0      0
## 199998  99999    1  3     2      0 0.2821291 0.2821291      1      0
## 199999 100000    1  2     1      0 3.0657201 3.0657201      0      1
## 200000 100000    1  3     2      0 3.0657201 3.0657201      1      1

# Check number of events same as before:
events(data_long)

## $Frequencies
##             to
## from          event-free cause1 cause2 no event total entering
##   event-free           0  29769  41094    29137         100000
##   cause1               0      0      0    29769          29769
##   cause2               0      0      0    41094          41094
##
## $Proportions
##             to
## from          event-free  cause1  cause2 no event
##   event-free     0.00000 0.29769 0.41094  0.29137
##   cause1         0.00000 0.00000 0.00000  1.00000
##   cause2         0.00000 0.00000 0.00000  1.00000

# Add cause-specific covariates for regression:
data_long <- expand.covs(data_long, covs = c("gender"))

head(data_long)
```

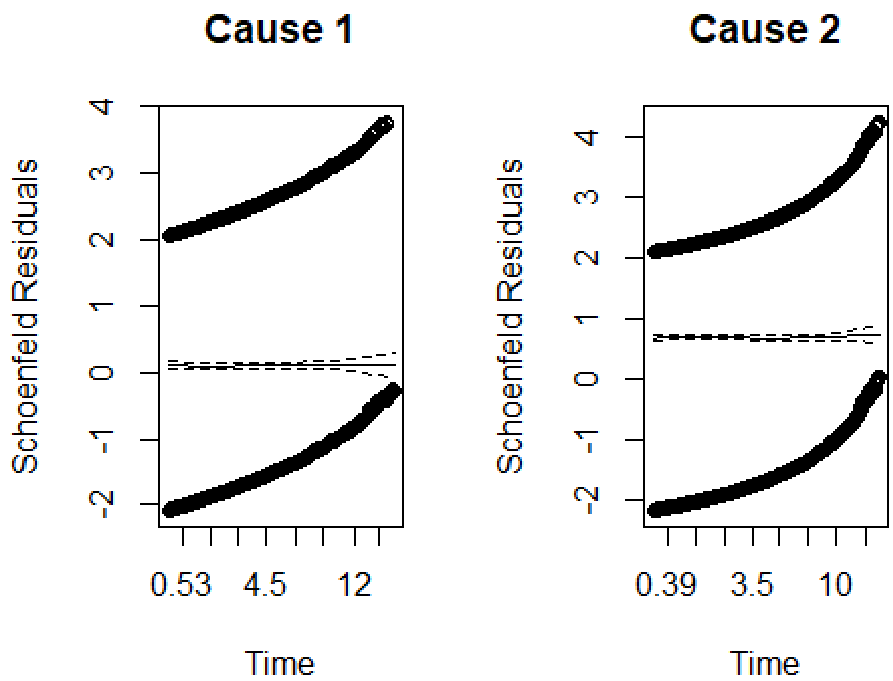
```
## An object of class 'msdata'
##
## Data:
##   id from to trans Tstart       Tstop        time status gender gender.1
## 1  1    1  2     1      0  0.01376141  0.01376141      0      1        1
## 2  1    1  3     2      0  0.01376141  0.01376141      0      1        0
## 3  2    1  2     1      0 14.63483371 14.63483371      0      1        1
## 4  2    1  3     2      0 14.63483371 14.63483371      0      1        0
## 5  3    1  2     1      0  4.26225636  4.26225636      0      1        1
## 6  3    1  3     2      0  4.26225636  4.26225636      0      1        0
##   gender.2
## 1        0
## 2        1
## 3        0
## 4        1
## 5        0
## 6        1

# Fit Cox propotional hazards model
c1 <- coxph(Surv(time, status) ~ gender.1 + gender.2 + strata(trans), data =
data_long)

summary(c1)

## Call:
## coxph(formula = Surv(time, status) ~ gender.1 + gender.2 + strata(trans),
##     data = data_long)
##
##   n= 200000, number of events= 70863
##
##             coef exp(coef) se(coef)      z Pr(>|z|)
## gender.1 0.10174   1.10709  0.01179  8.628   <2e-16 ***
## gender.2 0.70862   2.03119  0.01018 69.599   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## gender.1     1.107     0.9033     1.082     1.133
## gender.2     2.031     0.4923     1.991     2.072
##
## Concordance= 0.556  (se = 0.001 )
## Likelihood ratio test= 5047  on 2 df,   p=<2e-16
## Wald test            = 4919  on 2 df,   p=<2e-16
## Score (logrank) test = 5112  on 2 df,   p=<2e-16

Male <- data.frame(gender.1 = c(1,0), gender.2 = c(0,1),
                   trans = c(1, 2), strata = c(1, 2))
Female <- data.frame(gender.1 = c(0,0), gender.2 = c(0,0),
                     trans = c(1, 2), strata = c(1, 2))
# Estimated cumulative hazards for all event times
```

```r
msf.Male <- msfit(c1, Male, trans = tmat)
msf.Female <- msfit(c1, Female, trans = tmat)
# Caluculates Cumulative Incidence
pt.Male <- probtrans(msf.Male, 0)[[1]]
pt.Female <- probtrans(msf.Female, 0)[[1]]
# Plot
# Cause 1
plot1 <- ggplot(NULL, aes(x = time, y = pstate2)) +
      geom_step(data = pt.Male, aes(color = 'Male')) +
      geom_step(data = pt.Female, aes(color = 'Female')) + labs(title =
'Cause 1') + xlab('Time') + ylab('Probability') + ylim(0,1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

# Cause 2
plot12 <- ggplot(NULL, aes(x = time, y = pstate3)) +
      geom_step(data = pt.Male, aes(color = 'Male')) +
      geom_step(data = pt.Female, aes(color = 'Female')) + labs(title =
'Cause 2') + xlab('Time') + ylab('Probability') + ylim(0,1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

grid.arrange(plot1, plot12, nrow=1, ncol=2, top = "Cox model -  Beta 1 = 0.1,
Beta 2 = -0.7")
```




```r
coxc1 <- coxph(Surv(time,epsilon==1)~  gender, data = data1)
coxc2 <- coxph(Surv(time,epsilon==2)~  gender, data = data1)
```

```r
coxc1

## Call:
## coxph(formula = Surv(time, epsilon == 1) ~ gender, data = data1)
##
##            coef exp(coef) se(coef)     z      p
## gender 0.10174   1.10709  0.01179 8.628 <2e-16
##
## Likelihood ratio test=74.16  on 1 df, p=< 2.2e-16
## n= 100000, number of events= 29769

coxc2

## Call:
## coxph(formula = Surv(time, epsilon == 2) ~ gender, data = data1)
##
##            coef exp(coef) se(coef)    z      p
## gender 0.70862   2.03119  0.01018 69.6 <2e-16
##
## Likelihood ratio test=4973  on 1 df, p=< 2.2e-16
## n= 100000, number of events= 41094

# Proportional hazards assumption
temp01 <- cox.zph(coxc1)

# plot curves

par(mfrow = c(1,2))

plot(temp01, resid = T, se = T, main = 'Cause 1',
                 xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')
temp02 <- cox.zph(coxc2)

plot(temp02, resid = T, se = T, main = 'Cause 2',
                 xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')
```



To check the proportionality assumption in the Cox and Fine-Gray models, the `cox.zph` funtion from the survival package was used [2].

Fine and Gray Model

Data generation

```
# Data for Fine-Gray method - same as before with fewer participants
set.seed(12345)

# Participants
n <- 10000

### Simulate covariates as before
gender <- rbinom(n, 1, 0.5)
summary(gender)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.5041  1.0000  1.0000

# 0 = female, 1 = male

### Simulate competing risks
# k = 2 causes of failure

Btz1 <- 0.1*gender
Btz2 <- 0.7*gender
```

```
# Proportional cs hazards
cshr1 <- 0.1*exp(Btz1)
cshr2 <- 0.1*exp(Btz2)

# Latent failure times
time1 <- rexp(n, rate = cshr1)
time2 <- rexp(n, rate = cshr2)



# Status
# 1 if time 1 first, 2 if time 2 is first
epsilon <- 1*(time1<time2) + 2*(time1>time2)


# Calculate the observed times
time <- time1
time[epsilon == 2] <- time2[epsilon == 2]



### Simulate censoring
# Drop out times:
cens <- rexp(n, rate = 0.1)


epsilon[time > cens] <- 0
time[time>cens] <- cens[time>cens]


# Set max follow up time to 20
tmax <- 20
sum(1*(time > tmax))

## [1] 15

# Censor individuals not dead by tmax
epsilon[time > tmax] <- 0
time[time > tmax] <- tmax


# Generated dataset:
data1 <- data.frame(time, epsilon, gender)


data1 <- data1 %>% mutate(epsilon = as.integer(epsilon))
```

```r
causes <- data.frame(epsilon=c(0, 1, 2), cause = c("event-free", "cause1",
"cause2"))

data1 <- merge(data1, causes, by = "epsilon")
```

To obtain the weights for the Fine-Gray model, the `finegray` funtion was used [3].

Fine-Gray estimates

```r
###### using weighted coxph()

# cause 1

data10 <- data1 %>% mutate(epsilon = factor(epsilon))
data_c1 <- finegray(Surv(time, epsilon) ~ ., data=data10)

fgc1 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ gender,
                     weight=fgwt, data=data_c1)
summary(fgc1)
```

```
## Call:
## coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ gender, data = data_c1,
##     weights = fgwt)
##
##   n= 2924412, number of events= 2933
##
##            coef exp(coef) se(coef)      z Pr(>|z|)
## gender -0.23587   0.78988  0.03709 -6.359 2.03e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## gender    0.7899      1.266    0.7345    0.8494
##
## Concordance= 0.52  (se = 0.005 )
## Likelihood ratio test= 40.63  on 1 df,   p=2e-10
## Wald test            = 40.44  on 1 df,   p=2e-10
## Score (logrank) test = 40.62  on 1 df,   p=2e-10
```

```r
# cause 2

data_c2 <- finegray(Surv(time, epsilon) ~ ., data=data10, etype = '2')

fgc2 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ gender,
                     weight=fgwt, data=data_c2)
summary(fgc2)
```

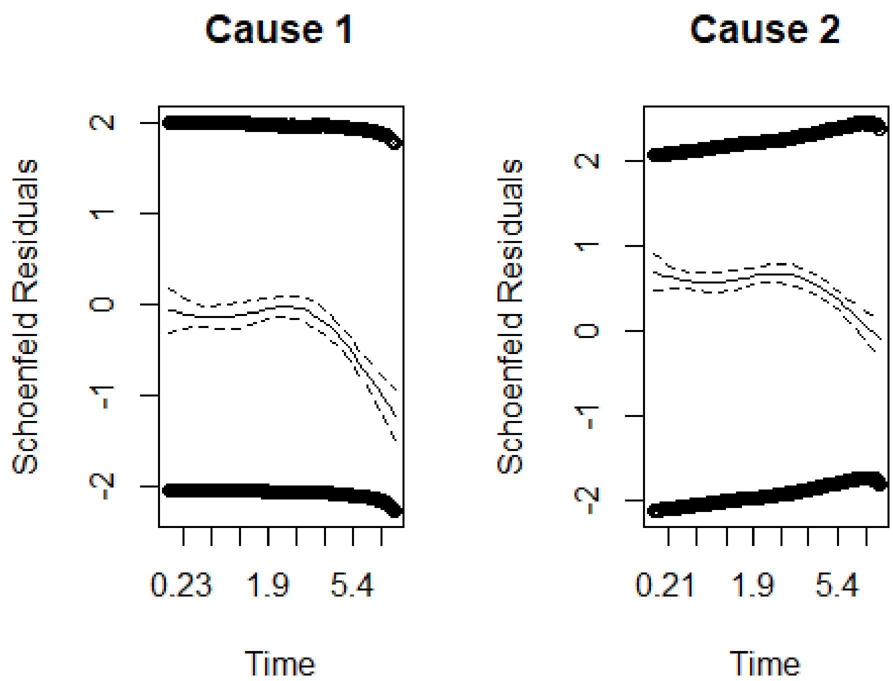
```
## Call:
## coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ gender, data = data_c2,
##     weights = fgwt)
```

```
##
##   n= 2439198, number of events= 4060
##
##           coef exp(coef) se(coef)     z Pr(>|z|)
## gender 0.53346   1.70482  0.03202 16.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##        exp(coef) exp(-coef) lower .95 upper .95
## gender     1.705     0.5866     1.601     1.815
##
## Concordance= 0.572  (se = 0.004 )
## Likelihood ratio test= 283.8  on 1 df,   p=<2e-16
## Wald test            = 277.6  on 1 df,   p=<2e-16
## Score (logrank) test = 284.1  on 1 df,   p=<2e-16
```

```
#plot

ndata <- data.frame(gender=c(1,0))
fgsurv1 <- survfit(fgc1, ndata)
fgsurv2 <- survfit(fgc2, ndata)


cif1 <- data.frame(time = fgsurv1$time, m = 1-fgsurv1$surv[,1], f = 1-
fgsurv1$surv[,2])

cif2 <- data.frame(time = fgsurv2$time, m = 1-fgsurv2$surv[,1], f = 1-
fgsurv2$surv[,2])

p1 <- ggplot(cif1) +
      geom_step( aes(x = time, y = m, color = 'Male')) +
      geom_step(aes(x = time, y = f, color = 'Female')) +
  labs(title = 'Cause 1') + xlab('Time') + ylab('Probability') + ylim(0, 1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

p2 <- ggplot(cif2) +
      geom_step( aes(x = time, y = m, color = 'Male')) +
      geom_step(aes(x = time, y = f, color = 'Female')) +
  labs(title = 'Cause 1') + xlab('Time') + ylab('Probability') + ylim(0, 1) +
scale_colour_manual(name="Gender",
    values=c(Male="red", Female="blue"))

grid.arrange(p1, p2, nrow=1, ncol=2, top = "Fine-Gray model -  Beta 1 = 0.1,
Beta 2 = -0.7")
```

Fine-Gray Diagnostics

```
# Proportional hazards assumption
temp <- cox.zph(fgc1)
print(temp)   # display the results

##          chisq df       p
## gender  31.2  1 2.3e-08
## GLOBAL  31.2  1 2.3e-08

# plot curves

par(mfrow = c(1,2))

plot(temp, resid = T, se = T, main = 'Cause 1',
                 xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')

temp2 <- cox.zph(fgc2)

# plot curves

plot(temp2, resid = T, se = T, main = 'Cause 2',
                 xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')
```



Fine-Gray model - Time-varying covariate

```
# Misspecified Fine-Gray model? Time-varying covariate.


# Cause 1
data_inter1 <- finegray(Surv(time, epsilon) ~ gender + gender*time,
data=data10)

fg_inter1 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ gender + gender*time,
weight=fgwt, data=data_inter1)

summary(fg_inter1)

## Call:
## coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ gender + gender *
##     time, data = data_inter1, weights = fgwt)
##
##   n= 2924412, number of events= 2933
##
##                  coef exp(coef)  se(coef)       z Pr(>|z|)
## gender      -0.426111  0.653044  0.056186  -7.584 3.35e-14 ***
## time        -0.134452  0.874195  0.009257 -14.525  < 2e-16 ***
## gender:time  0.022411  1.022664  0.015350   1.460    0.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##              exp(coef) exp(-coef) lower .95 upper .95
## gender          0.6530     1.5313    0.5849    0.7291
## time            0.8742     1.1439    0.8585    0.8902
## gender:time     1.0227     0.9778    0.9924    1.0539
##
## Concordance= 0.706  (se = 0.004 )
## Likelihood ratio test= 390.4  on 3 df,   p=<2e-16
## Wald test            = 348.3  on 3 df,   p=<2e-16
## Score (logrank) test = 350.5  on 3 df,   p=<2e-16
```

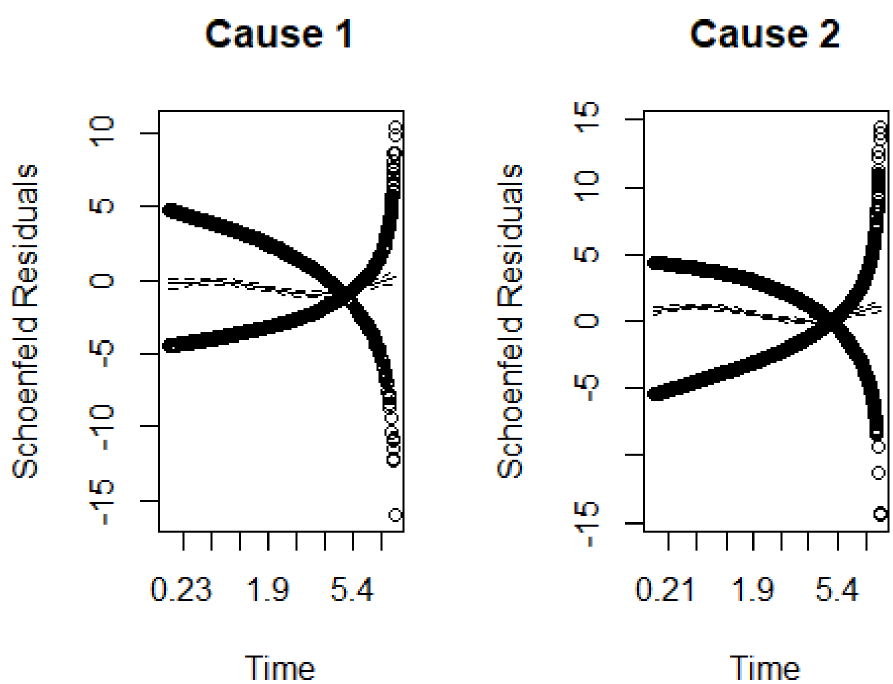
```
# Cause 2
data_inter2 <- finegray(Surv(time, epsilon) ~ gender + gender*time, etype =
'2', data=data10)

fg_inter2 <- coxph(Surv(fgstart, fgstop, fgstatus) ~ gender + gender*time,
weight=fgwt, data=data_inter2)

summary(fg_inter2)
```

```
## Call:
## coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ gender + gender *
##     time, data = data_inter2, weights = fgwt)
##
##   n= 2439198, number of events= 4060
##
##                  coef exp(coef)  se(coef)       z Pr(>|z|)
## gender       0.633244  1.883711  0.049258  12.856  < 2e-16 ***
## time        -0.115843  0.890615  0.008645 -13.400  < 2e-16 ***
## gender:time -0.066017  0.936115  0.013044  -5.061 4.17e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## gender          1.8837     0.5309    1.7104    2.0746
## time            0.8906     1.1228    0.8757    0.9058
## gender:time     0.9361     1.0682    0.9125    0.9604
##
## Concordance= 0.76  (se = 0.003 )
## Likelihood ratio test= 906.4  on 3 df,   p=<2e-16
## Wald test            = 814.9  on 3 df,   p=<2e-16
## Score (logrank) test = 847.4  on 3 df,   p=<2e-16
```

```
# Proportional hazards assumption
temp <- cox.zph(fg_inter1)
print(temp)   # display the results
```

```
##               chisq df       p
## gender         48.3  1 3.7e-12
## time         2233.8  1 < 2e-16
## gender:time   383.9  1 < 2e-16
## GLOBAL       2244.0  3 < 2e-16
```

```
par(mfrow = c(1,2))
# plot curves
plot(temp, residuals = T, se = T, var =1, main = 'Cause 1', xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')

## Warning in plot.window(...): "residuals" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "residuals" is not a graphical
parameter

## Warning in title(...): "residuals" is not a graphical parameter

temp2 <- cox.zph(fg_inter2)
plot(temp2, residuals = T, se = T, var = 1, main = 'Cause 2', xlab = 'Time',
                 ylab = 'Schoenfeld Residuals')

## Warning in plot.window(...): "residuals" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "residuals" is not a graphical
parameter

## Warning in title(...): "residuals" is not a graphical parameter
```



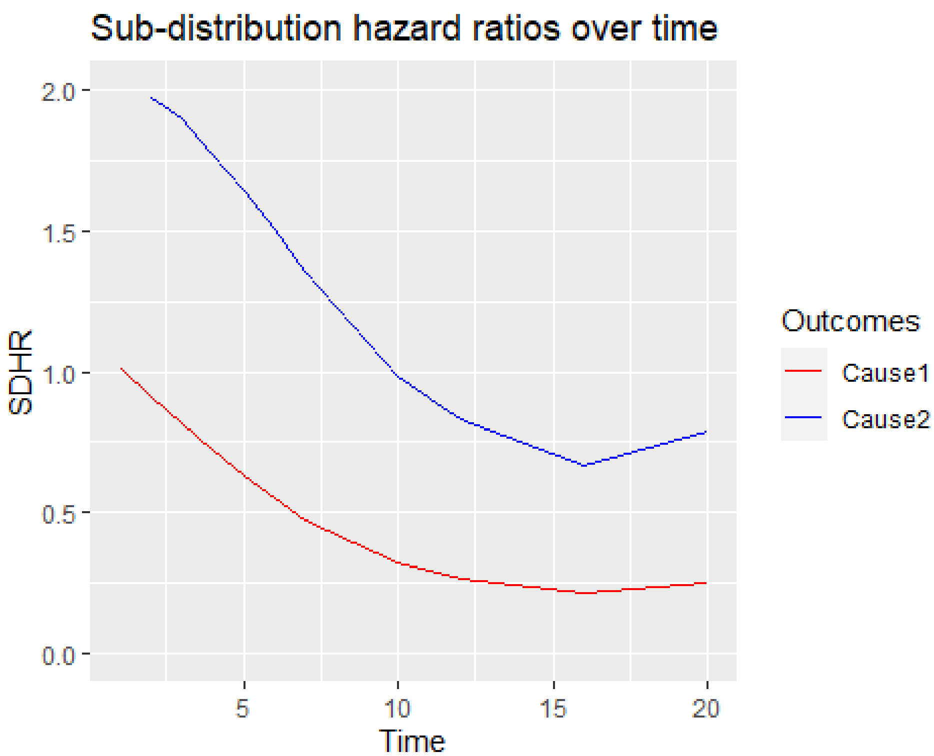
‘Misspecified’ Fine-Gray model

```
# Relationship between SDHR and CSHR:

# Calculate cumulative incidences for z = 0 and z = 1 at set of times
```

```r
cif00 <- cuminc(data1$time, as.numeric
             (data1$epsilon),
             group = data1$gender)

ci_ests <- timepoints(cif00, times = c( 1,2,3,4,5,6,7, 10, 12, 16, 20))$est

ci_ests

##              1         2         3         4         5         6         7
## 0 1 0.09651617 0.1721161 0.2306937 0.2714853 0.3096000 0.3410174 0.3708627
## 1 1 0.08664275 0.1583853 0.2108696 0.2507509 0.2762796 0.2983249 0.3139671
## 0 2 0.09107428 0.1699302 0.2313699 0.2779355 0.3167791 0.3509061 0.3760570
## 1 2 0.16362706 0.2894507 0.3797288 0.4446263 0.5007349 0.5379784 0.5680116
##            10        12        16        20
## 0 1 0.4246653 0.4505346 0.4715983 0.4798019
## 1 1 0.3388530 0.3449980 0.3527122 0.3537710
## 0 2 0.4395455 0.4612800 0.4924255 0.5053170
## 1 2 0.6157633 0.6300528 0.6388176 0.6419939

ci_ests[,2]

##       0 1       1 1       0 2       1 2
## 0.1721161 0.1583853 0.1699302 0.2894507

SDHR <- function(b1 = 0.1, b2 = 0.7, tp=1) {
  # function estimating the SDHR at a time point tp using the CSHR

  # survival for z = 1 and z = 0
  S1 <- 1 - (ci_ests[,tp][2] + ci_ests[,tp][4])
  S0 <- 1 - (ci_ests[,tp][1] + ci_ests[,tp][3])



  # k=1
  num1 <- 1 - ci_ests[,tp][1]
  denom1 <- 1 - ci_ests[,tp][2]

  sdhr1 <- exp(b1)*(S1/S0)*(num1/denom1)

  # k=2
  num2 <- 1 - ci_ests[,tp][3]
  denom2 <- 1 - ci_ests[,tp][4]

  sdhr2 <- exp(b2)*(S1/S0)*(num2/denom2)

  sdhr <- c(sdhr1, sdhr2)
  return(sdhr)
}



SDHR(tp = 1)
```

```r
##      1 1      1 1
## 1.008879 2.019596
SDHR(tp = 2)
##       1 1       1 1
## 0.9123434 1.9742370
SDHR(tp = 3)
##       1 1       1 1
## 0.8199712 1.8991555
SDHR(tp = 4)
##       1 1       1 1
## 0.7264954 1.7700608
SDHR(tp = 5)
##       1 1       1 1
## 0.6292236 1.6446806
SDHR(tp = 6)
##       1 1       1 1
## 0.5515039 1.5032546
SDHR(tp = 7)
##       1 1       1 1
## 0.4726415 1.3563784
SDHR(tp = 8)
##       1 1       1 1
## 0.3214299 0.9817072
cos1 <- c(SDHR(t=1)[1], SDHR(t=2)[1], SDHR(t=3)[1], SDHR(t=4)[1],
SDHR(t=5)[1], SDHR(t=6)[1], SDHR(t=7)[1], SDHR(t=8)[1], SDHR(t=9)[1],
SDHR(t=10)[1],SDHR(t=11)[1])

cos2 <- c(SDHR(t=1)[2], SDHR(t=2)[2], SDHR(t=3)[2], SDHR(t=4)[2],
SDHR(t=5)[2], SDHR(t=6)[2], SDHR(t=7)[2], SDHR(t=8)[2], SDHR(t=9)[2],
SDHR(t=10)[2],SDHR(t=11)[2])

times = c(1,2,3,4,5,6,7,10, 12, 16, 20)

truesdhr <- data.frame(times, cos1, cos2)



ggplot(data = truesdhr) +
```

```
     geom_line(aes(x = times, y = cos1, color = 'Cause1')) +
     geom_line(aes(x = times, y = cos2, color = 'Cause2')) +
     labs(title = 'Sub-distribution hazard ratios over time') + xlab('Time')
+ ylab('SDHR') + ylim(0, 2) + scale_colour_manual(name="Outcomes",
values=c(Cause1="red", Cause2="blue"))

## Warning: Removed 1 row(s) containing missing values (geom_path).
```



Pseudo-value approach

Generate data

```
# Dataset for pseudovalue experiments - same as before but n=1000
# Participants
set.seed(12345)
n <- 1000

### Simulate covariates as before
gender <- rbinom(n, 1, 0.5)
summary(gender)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.531   1.000   1.000

# 0 = female, 1 = male

### Simulate competing risks
# k = 2 causes of failure
```

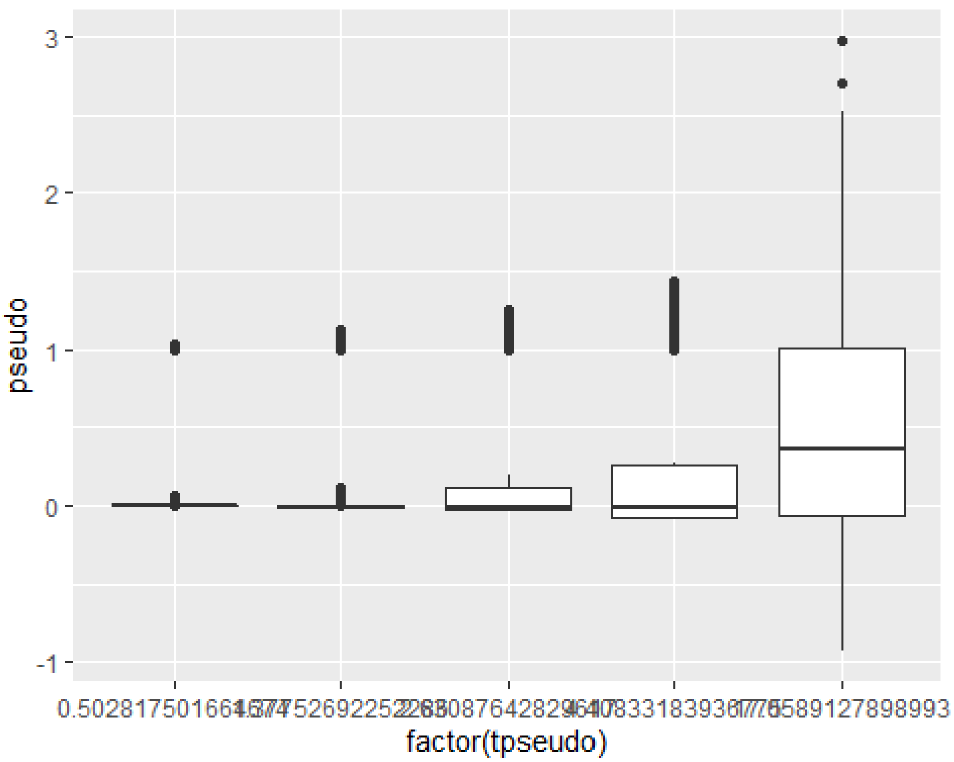
```
Btz1 <- 0.1*gender
Btz2 <- 0.7*gender


# Proportional cs hazards
cshr1 <- 0.1*exp(Btz1)
cshr2 <- 0.1*exp(Btz2)

# Latent failure times
time1 <- rexp(n, rate = cshr1)
time2 <- rexp(n, rate = cshr2)



# Status
# 1 if time 1 first, 2 if time 2 is first
epsilon <- 1*(time1<time2) + 2*(time1>time2)


# Calculate the observed times
time <- time1
time[epsilon == 2] <- time2[epsilon == 2]



### Simulate censoring
# Drop out times:
cens <- rexp(n, rate = 0.1)


epsilon[time > cens] <- 0
time[time>cens] <- cens[time>cens]


# Set max follow up time to 20
tmax <- 20
sum(1*(time > tmax))

## [1] 3

# Censor individuals not dead by tmax
epsilon[time > tmax] <- 0
time[time > tmax] <- tmax


# Generated dataset:
data1 <- data.frame(time, epsilon, gender)
```

```
data1 <- data1 %>% mutate(epsilon = as.integer(epsilon))

causes <- data.frame(epsilon=c(0, 1, 2), cause = c("event-free", "cause1",
"cause2"))

data1 <- merge(data1, causes, by = "epsilon")
```

The code used below to implement the pseudo-value approach was adapted from a tutorial by Klein et al. on producing pseudo-value estimates using the pseudo package in R [4].

```
head(data1)

##   epsilon      time gender      cause
## 1       0 2.1980926      0 event-free
## 2       0 0.8703240      1 event-free
## 3       0 0.7990876      1 event-free
## 4       0 2.2274626      0 event-free
## 5       0 5.9433316      1 event-free
## 6       0 3.4807279      0 event-free

data_pseudo <- data1[data1$epsilon != 0, ]
data_pseudo <- data_pseudo %>%
  mutate(epsilon = as.integer(epsilon))

summary(data_pseudo$epsilon)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   1.599   2.000   2.000

# Vector of 5-10 evenly spaced time points on the event scale - to find
pseudo-values at

# Quantiles
quantile(data_pseudo$time, probs = c(0.2,0.4,0.6,0.8,1) )

##        20%        40%        60%        80%       100%
##  0.5028175  1.3775269  2.6308764  4.4083318 17.0589128

t_pts <- quantile(data_pseudo$time, probs = c(0.2,0.4,0.6,0.8,1) )

data_pseudo <- data1 %>%
  mutate(epsilon = as.integer(epsilon))

pseudo <- pseudoci(time = data_pseudo$time, event = data_pseudo$epsilon, tmax
= t_pts)

# Cause1
b <- NULL
for(it in 1:length(pseudo$time)){
    b <- rbind(b,cbind(data_pseudo,pseudo = pseudo$pseudo$cause1[,it],
```

```
        tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
b <- b[order(b$id),]

b$tpseudo <- factor(b$tpseudo)

ggplot(b, aes(x = factor(tpseudo), y = pseudo)) + geom_boxplot()
```




```
# fit the model
library(geepack)


##### CAUSE 1
fit_c1 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")






#The results using the AJ variance estimate
h1 <- cbind(mean = round(fit_c1$beta,4), SD =
round(sqrt(diag(fit_c1$vbeta.ajs)),4),
    Z = round(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)),4),
    PVal = round(2-2*pnorm(abs(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)))),4))
```

```r
# Logit link
fit_c12 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")


#The results using the AJ variance estimate
h2 <- cbind(mean = round(fit_c12$beta ,4), SD =
round(sqrt(diag(fit_c12$vbeta.ajs)),4),
    Z = round(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)),4),
    PVal = round(2-
2*pnorm(abs(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)))),4))




#### Cause 2

c <- NULL
for(it in 1:length(pseudo$time)){
    c <- rbind(c,cbind(data_pseudo,pseudo = pseudo$pseudo$cause2[,it],
         tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
c <- c[order(c$id),]


fit_c2 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")


#The results using the AJ variance estimate
c1 <- cbind(mean = round(fit_c2$beta,3), SD =
round(sqrt(diag(fit_c2$vbeta.ajs)),3),
    Z = round(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)),3),
    PVal = round(2-2*pnorm(abs(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)))),3))

fit_c22 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")

#The results using the AJ variance estimate
c2 <- cbind(mean = round(fit_c22$beta,3), SD =
round(sqrt(diag(fit_c22$vbeta.ajs)),3),
    Z = round(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)),3),
    PVal = round(2-
2*pnorm(abs(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)))),3))
```

```r
th1 <- data.frame(covariate = h1[,0], estimate = h1[,1], se = h1[,2], p =
h1[,4])

th2 <- data.frame(covariate = h2[,0], estimate = h2[,1], se = h2[,2], p =
h2[,4])

tc1 <- data.frame(covariate = c1[,0], estimate = c1[,1], se = c1[,2], p =
c1[,4])

tc2 <- data.frame(covariate = c2[,0], estimate = c2[,1], se = c2[,2], p =
c2[,4])

print(th1)

##                                    estimate     se     p
## (Intercept)                         -2.4992 0.1306 0e+00
## as.factor(tpseudo)1.37752692252286   0.5846 0.0853 0e+00
## as.factor(tpseudo)2.63087642829617   1.1308 0.1068 0e+00
## as.factor(tpseudo)4.40833183936775   1.5691 0.1166 0e+00
## as.factor(tpseudo)17.0589127898993   2.0216 0.1248 0e+00
## gender                              -0.3974 0.1178 7e-04

print(th2)

##                                    estimate     se      p
## (Intercept)                         -2.4451 0.1383 0.0000
## as.factor(tpseudo)1.37752692252286   0.6171 0.0889 0.0000
## as.factor(tpseudo)2.63087642829617   1.2153 0.1120 0.0000
## as.factor(tpseudo)4.40833183936775   1.7162 0.1235 0.0000
## as.factor(tpseudo)17.0589127898993   2.2615 0.1352 0.0000
## gender                              -0.4561 0.1412 0.0012

print(tc1)

##                                    estimate    se p
## (Intercept)                          -2.936 0.134 0
## as.factor(tpseudo)1.37752692252286    0.922 0.093 0
## as.factor(tpseudo)2.63087642829617    1.428 0.106 0
## as.factor(tpseudo)4.40833183936775    1.868 0.112 0
## as.factor(tpseudo)17.0589127898993    2.405 0.118 0
## gender                                0.768 0.100 0

print(tc2)

##                                    estimate    se p
## (Intercept)                          -3.104 0.155 0
## as.factor(tpseudo)1.37752692252286    1.015 0.101 0
## as.factor(tpseudo)2.63087642829617    1.616 0.116 0
```

```
## as.factor(tpseudo)4.40833183936775    2.179 0.125 0
## as.factor(tpseudo)17.0589127898993    2.957 0.138 0
## gender                                1.028 0.132 0
```

Pseudovalue - Unstructure correlation structure

```
################## Unstructured correlation structure


##### CAUSE 1
fit_c1 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr = 'unstructured')

summary(fit_c1)

##
## Call:
## geese(formula = pseudo ~ as.factor(tpseudo) + gender, id = id,
##     data = b, family = gaussian, mean.link = "cloglog", scale.fix = TRUE,
##     corstr = "unstructured", jack = TRUE)
##
## Mean Model:
##  Mean Link:                 cloglog
##  Variance to Mean Relation: gaussian
##
##  Coefficients:
##                                      estimate     san.se     ajs.se
wald
## (Intercept)                        -2.4809156 0.13102605 0.13009644
358.51612
## as.factor(tpseudo)1.37752692252286  0.5838263 0.08636679 0.08573049
45.69553
## as.factor(tpseudo)2.63087642829617  1.1329352 0.10828445 0.10748850
109.46569
## as.factor(tpseudo)4.40833183936775  1.5762481 0.11828533 0.11741653
177.57726
## as.factor(tpseudo)17.0589127898993  2.0370292 0.12686867 0.12593644
257.80169
## gender                             -0.4845546 0.11899819 0.11821658
16.58077
##                                               p
## (Intercept)                        0.000000e+00
## as.factor(tpseudo)1.37752692252286 1.381373e-11
## as.factor(tpseudo)2.63087642829617 0.000000e+00
## as.factor(tpseudo)4.40833183936775 0.000000e+00
## as.factor(tpseudo)17.0589127898993 0.000000e+00
## gender                             4.662131e-05
##
## Scale is fixed.
##
```

```
## Correlation Model:
##  Correlation Structure:     unstructured
##  Correlation Link:          identity
##
##  Estimated Correlation Parameters:
##             estimate     san.se     ajs.se      wald p
## alpha.1:2 0.1381088 0.01518154 0.01510124  82.75819 0
## alpha.1:3 0.1267161 0.01397244 0.01389643  82.24682 0
## alpha.1:4 0.1130464 0.01259463 0.01252283  80.56438 0
## alpha.1:5 0.0935502 0.01100174 0.01093406  72.30466 0
## alpha.2:3 0.2275223 0.01768507 0.01759335 165.51385 0
## alpha.2:4 0.2029973 0.01606806 0.01597958 159.60751 0
## alpha.2:5 0.1680652 0.01429174 0.01420447 138.28827 0
## alpha.3:4 0.3493802 0.01959094 0.01949132 318.04305 0
## alpha.3:5 0.2877260 0.01797630 0.01786873 256.18731 0
## alpha.4:5 0.4440466 0.02138613 0.02126109 431.11436 0
##
## Returned Error Value:    0
## Number of clusters:   1000   Maximum cluster size: 5
```

```
#The results using the AJ variance estimate
h1 <- cbind(mean = round(fit_c1$beta,4), SD =
round(sqrt(diag(fit_c1$vbeta.ajs)),4),
    Z = round(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)),4),
    PVal = round(2-2*pnorm(abs(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)))),4))


# Logit link
fit_c12 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="unstructured")


#The results using the AJ variance estimate
h2 <- cbind(mean = round(fit_c12$beta ,4), SD =
round(sqrt(diag(fit_c12$vbeta.ajs)),4),
    Z = round(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)),4),
    PVal = round(2-
2*pnorm(abs(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)))),4))




#### Cause 2

c <- NULL
for(it in 1:length(pseudo$time)){
    c <- rbind(c,cbind(data_pseudo,pseudo = pseudo$pseudo$cause2[,it],
        tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
```

```r
}
c <- c[order(c$id),]


fit_c2 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")


#The results using the AJ variance estimate
c1 <- cbind(mean = round(fit_c2$beta,3), SD =
round(sqrt(diag(fit_c2$vbeta.ajs)),3),
    Z = round(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)),3),
    PVal = round(2-2*pnorm(abs(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)))),3))

fit_c22 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")

#The results using the AJ variance estimate
c2 <- cbind(mean = round(fit_c22$beta,3), SD =
round(sqrt(diag(fit_c22$vbeta.ajs)),3),
    Z = round(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)),3),
    PVal = round(2-
2*pnorm(abs(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)))),3))


th1 <- data.frame(covariate = h1[,0], estimate = h1[,1], se = h1[,2], p =
h1[,4])

th2 <- data.frame(covariate = h2[,0], estimate = h2[,1], se = h2[,2], p =
h2[,4])

tc1 <- data.frame(covariate = c1[,0], estimate = c1[,1], se = c1[,2], p =
c1[,4])

tc2 <- data.frame(covariate = c2[,0], estimate = c2[,1], se = c2[,2], p =
c2[,4])
```

Cause 1 Results

```r
print(th1)

##                                   estimate     se p
## (Intercept)                        -2.4809 0.1301 0
## as.factor(tpseudo)1.37752692252286  0.5838 0.0857 0
## as.factor(tpseudo)2.63087642829617  1.1329 0.1075 0
## as.factor(tpseudo)4.40833183936775  1.5762 0.1174 0
## as.factor(tpseudo)17.0589127898993  2.0370 0.1259 0
## gender                             -0.4846 0.1182 0
```

```r
print(th2)
```

```
##                                        estimate     se p
## (Intercept)                             -2.4203 0.1385 0
## as.factor(tpseudo)1.37752692252286       0.6154 0.0905 0
## as.factor(tpseudo)2.63087642829617       1.2187 0.1144 0
## as.factor(tpseudo)4.40833183936775       1.7290 0.1264 0
## as.factor(tpseudo)17.0589127898993       2.2943 0.1391 0
## gender                                  -0.6168 0.1476 0
```

Cause 2 Results

```r
print(tc1)
```

```
##                                        estimate    se p
## (Intercept)                              -2.936 0.134 0
## as.factor(tpseudo)1.37752692252286        0.922 0.093 0
## as.factor(tpseudo)2.63087642829617        1.428 0.106 0
## as.factor(tpseudo)4.40833183936775        1.868 0.112 0
## as.factor(tpseudo)17.0589127898993        2.405 0.118 0
## gender                                    0.768 0.100 0
```

```r
print(tc2)
```

```
##                                        estimate    se p
## (Intercept)                              -3.104 0.155 0
## as.factor(tpseudo)1.37752692252286        1.015 0.101 0
## as.factor(tpseudo)2.63087642829617        1.616 0.116 0
## as.factor(tpseudo)4.40833183936775        2.179 0.125 0
## as.factor(tpseudo)17.0589127898993        2.957 0.138 0
## gender                                    1.028 0.132 0
```

Pseudovalue approach with 1 time point

```r
#### 1 time point

head(data_pseudo)
```

```
##   epsilon      time gender      cause
## 1       0 2.1980926      0 event-free
## 2       0 0.8703240      1 event-free
## 3       0 0.7990876      1 event-free
## 4       0 2.2274626      0 event-free
## 5       0 5.9433316      1 event-free
## 6       0 3.4807279      0 event-free
```

```r
tpseudo <- 1.99

pseudo1 <- pseudoci(time = data_pseudo$time, event = data_pseudo$epsilon,
tmax = tpseudo)

pseudo <- pseudo1$pseudo$cause1
id = seq(1, 100, 1)
```

```r
# Cause1
b1 <- cbind(data_pseudo,pseudo, id)

# fit the model
mod.t1 <- geese(formula = pseudo ~ gender, data = b1, id = id, family =
gaussian, mean.link = 'cloglog', corstr = 'independence')

summary(mod.t1)

##
## Call:
## geese(formula = pseudo ~ gender, id = id, data = b1, family = gaussian,
##     mean.link = "cloglog", corstr = "independence")
##
## Mean Model:
##  Mean Link:                 cloglog
##  Variance to Mean Relation: gaussian
##
##  Coefficients:
##                estimate    san.se        wald         p
## (Intercept) -1.73301069 0.1201505 208.0419968 0.0000000
## gender      -0.06400755 0.1660430   0.1486008 0.6998761
##
## Scale Model:
##  Scale Link:                identity
##
##  Estimated Scale Parameters:
##              estimate      san.se     wald p
## (Intercept) 0.1427284 0.009016342 250.5878 0
##
## Correlation Model:
##  Correlation Structure:     independence
##
## Returned Error Value:    0
## Number of clusters:   1000   Maximum cluster size: 1

mod.t11 <- geese(formula = pseudo ~ gender, data = b1, id = id, family =
gaussian, mean.link = 'logit', corstr = 'independence')

summary(mod.t11)

##
## Call:
## geese(formula = pseudo ~ gender, id = id, data = b1, family = gaussian,
##     mean.link = "logit", corstr = "independence")
##
## Mean Model:
##  Mean Link:                 logit
##  Variance to Mean Relation: gaussian
```

```
##
##  Coefficients:
##                 estimate    san.se        wald         p
## (Intercept) -1.64333359 0.1310816 157.1694616 0.0000000
## gender      -0.06964337 0.1806918   0.1485535 0.6999215
##
## Scale Model:
##  Scale Link:                identity
##
##  Estimated Scale Parameters:
##               estimate      san.se     wald p
## (Intercept) 0.1427284 0.009016342 250.5878 0
##
## Correlation Model:
##  Correlation Structure:     independence
##
## Returned Error Value:    0
## Number of clusters:   1000   Maximum cluster size: 1
```

```
##### CAUSE 1
p.fit.1 <- glm(pseudo ~ gender, data=b, family=gaussian)
#summary(p.fit.1)



#### CAUSE 2

pseudo2 <- pseudo1$pseudo$cause2
b2 <- cbind(data_pseudo,pseudo2, id)

mod.t2 <- geese(formula = pseudo2 ~ gender, data = b2, id = id, family =
gaussian, mean.link = 'cloglog', corstr = 'independence')
summary(mod.t2)
```

```
##
## Call:
## geese(formula = pseudo2 ~ gender, id = id, data = b2, family = gaussian,
##     mean.link = "cloglog", corstr = "independence")
##
## Mean Model:
##  Mean Link:                 cloglog
##  Variance to Mean Relation: gaussian
##
##  Coefficients:
##               estimate    san.se      wald            p
## (Intercept) -1.8668516 0.1281512 212.21420 0.000000e+00
## gender       0.9209549 0.1510856  37.15615 1.090381e-09
##
## Scale Model:
##  Scale Link:                identity
```

```
##
##  Estimated Scale Parameters:
##             estimate      san.se     wald p
## (Intercept) 0.1886259 0.007984474 558.0972 0
##
## Correlation Model:
##  Correlation Structure:     independence
##
## Returned Error Value:    0
## Number of clusters:   1000   Maximum cluster size: 1
```

```
mod.t22 <- geese(formula = pseudo2 ~ gender, data = b2, id = id, family =
gaussian, mean.link = 'logit', corstr = 'independence')
summary(mod.t22)
```

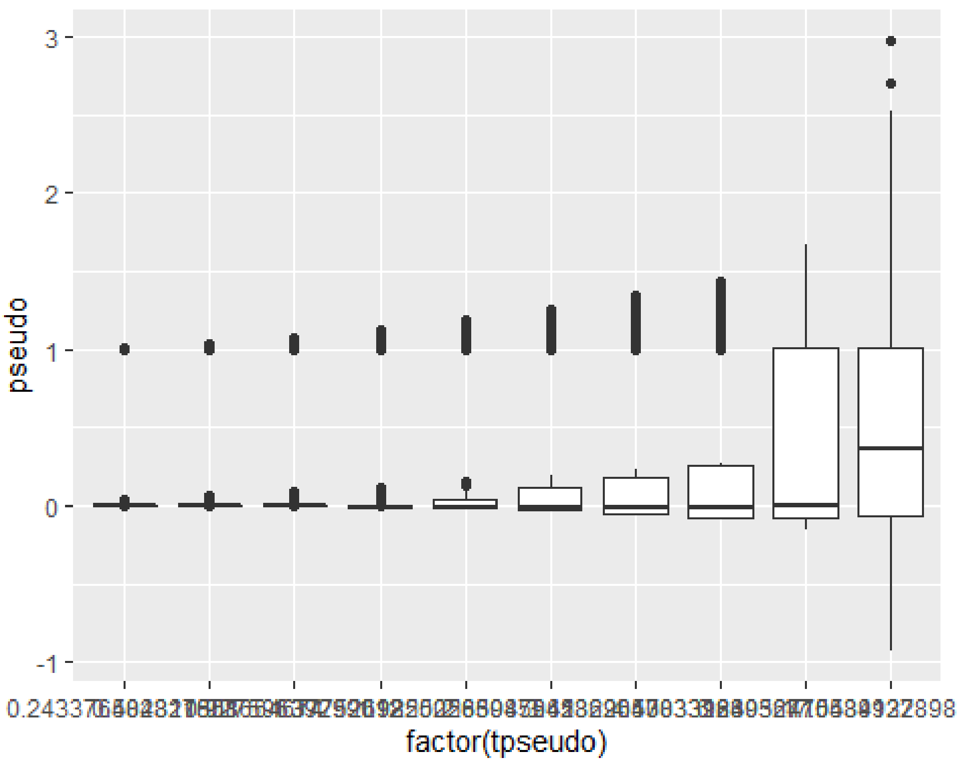
```
##
## Call:
## geese(formula = pseudo2 ~ gender, id = id, data = b2, family = gaussian,
##     mean.link = "logit", corstr = "independence")
##
## Mean Model:
##  Mean Link:                 logit
##  Variance to Mean Relation: gaussian
##
##  Coefficients:
##             estimate    san.se      wald           p
## (Intercept) -1.788551 0.1383130 167.21541 0.00000e+00
## gender       1.043095 0.1686882  38.23652 6.26684e-10
##
## Scale Model:
##  Scale Link:                identity
##
##  Estimated Scale Parameters:
##             estimate      san.se     wald p
## (Intercept) 0.1886259 0.007984474 558.0972 0
##
## Correlation Model:
##  Correlation Structure:     independence
##
## Returned Error Value:    0
## Number of clusters:   1000   Maximum cluster size: 1
```

```
p.fit.2 <- glm(pseudo ~ gender, data=c, family=gaussian)
#summary(p.fit.2)
```

Pseudovalue approach with 10 time points

```
#### 10 time points

data_pseudo <- data1[data1$epsilon != 0, ]
data_pseudo <- data_pseudo %>%
```

```
  mutate(epsilon = as.integer(epsilon))

summary(data_pseudo$epsilon)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   1.599   2.000   2.000

# Vector of 5-10 evenly spaced time points on the event scale - to find
pseudo-values at

# Quantiles
quantile(data_pseudo$time, probs = c(0.1, 0.2, 0.3,0.4, 0.5,
0.6,0.7,0.8,0.9,1) )

##        10%        20%        30%        40%        50%        60%
70%
##  0.2433765  0.5028175  0.9275964  1.3775269  1.9850057  2.6308764
3.4136205
##        80%        90%       100%
##  4.4083318  6.2605241 17.0589128

t_pts <- quantile(data_pseudo$time, probs = c(0.1, 0.2, 0.3,0.4, 0.5,
0.6,0.7,0.8,0.9,1) )

data_pseudo <- data1 %>%
  mutate(epsilon = as.integer(epsilon))

pseudo <- pseudoci(time = data_pseudo$time, event = data_pseudo$epsilon, tmax
= t_pts)

# Cause1
b <- NULL
for(it in 1:length(pseudo$time)){
    b <- rbind(b,cbind(data_pseudo,pseudo = pseudo$pseudo$cause1[,it],
          tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
b <- b[order(b$id),]

b$tpseudo <- factor(b$tpseudo)

ggplot(b, aes(x = factor(tpseudo), y = pseudo)) + geom_boxplot()
```




```r
# fit the model
library(geepack)


##### CAUSE 1
fit_c1 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")



#The results using the AJ variance estimate
h1 <- cbind(mean = round(fit_c1$beta,4), SD =
round(sqrt(diag(fit_c1$vbeta.ajs)),4),
    Z = round(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)),4),
    PVal = round(2-2*pnorm(abs(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)))),4))



# Logit link
fit_c12 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")
```

```
#The results using the AJ variance estimate
h2 <- cbind(mean = round(fit_c12$beta ,4), SD =
round(sqrt(diag(fit_c12$vbeta.ajs)),4),
    Z = round(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)),4),
    PVal = round(2-
2*pnorm(abs(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)))),4))




#### Cause 2

c <- NULL
for(it in 1:length(pseudo$time)){
    c <- rbind(c,cbind(data_pseudo,pseudo = pseudo$pseudo$cause2[,it],
        tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
c <- c[order(c$id),]


fit_c2 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")


#The results using the AJ variance estimate
c1 <- cbind(mean = round(fit_c2$beta,3), SD =
round(sqrt(diag(fit_c2$vbeta.ajs)),3),
    Z = round(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)),3),
    PVal = round(2-2*pnorm(abs(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)))),3))

fit_c22 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")

#The results using the AJ variance estimate
c2 <- cbind(mean = round(fit_c22$beta,3), SD =
round(sqrt(diag(fit_c22$vbeta.ajs)),3),
    Z = round(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)),3),
    PVal = round(2-
2*pnorm(abs(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)))),3))




th1 <- data.frame(covariate = h1[,0], estimate = h1[,1], se = h1[,2], p =
h1[,4])

th2 <- data.frame(covariate = h2[,0], estimate = h2[,1], se = h2[,2], p =
```

```r
h2[,4])

tc1 <- data.frame(covariate = c1[,0], estimate = c1[,1], se = c1[,2], p =
c1[,4])

tc2 <- data.frame(covariate = c2[,0], estimate = c2[,1], se = c2[,2], p =
c2[,4])

print(th1)

##                                      estimate     se      p
## (Intercept)                           -3.1537 0.1697 0.0000
## as.factor(tpseudo)0.502817501664674    0.6374 0.1166 0.0000
## as.factor(tpseudo)0.92759639299012     0.9566 0.1323 0.0000
## as.factor(tpseudo)1.37752692252286     1.2216 0.1422 0.0000
## as.factor(tpseudo)1.98500565945965     1.5253 0.1503 0.0000
## as.factor(tpseudo)2.63087642829617     1.7646 0.1549 0.0000
## as.factor(tpseudo)3.41362053033964     1.9830 0.1583 0.0000
## as.factor(tpseudo)4.40833183936775     2.1989 0.1612 0.0000
## as.factor(tpseudo)6.26052410434932     2.3960 0.1635 0.0000
## as.factor(tpseudo)17.0589127898993     2.6469 0.1669 0.0000
## gender                                -0.3356 0.1181 0.0045

print(th2)

##                                      estimate     se      p
## (Intercept)                           -3.1231 0.1752 0.0000
## as.factor(tpseudo)0.502817501664674    0.6578 0.1194 0.0000
## as.factor(tpseudo)0.92759639299012     0.9915 0.1354 0.0000
## as.factor(tpseudo)1.37752692252286     1.2732 0.1457 0.0000
## as.factor(tpseudo)1.98500565945965     1.6015 0.1543 0.0000
## as.factor(tpseudo)2.63087642829617     1.8660 0.1593 0.0000
## as.factor(tpseudo)3.41362053033964     2.1125 0.1633 0.0000
## as.factor(tpseudo)4.40833183936775     2.3606 0.1669 0.0000
## as.factor(tpseudo)6.26052410434932     2.5939 0.1700 0.0000
## as.factor(tpseudo)17.0589127898993     2.9002 0.1750 0.0000
## gender                                -0.3788 0.1399 0.0068

print(tc1)

##                                      estimate    se p
## (Intercept)                            -3.759 0.182 0
## as.factor(tpseudo)0.502817501664674     0.779 0.126 0
## as.factor(tpseudo)0.92759639299012      1.343 0.148 0
## as.factor(tpseudo)1.37752692252286      1.704 0.156 0
## as.factor(tpseudo)1.98500565945965      1.972 0.160 0
## as.factor(tpseudo)2.63087642829617      2.212 0.163 0
## as.factor(tpseudo)3.41362053033964      2.445 0.166 0
## as.factor(tpseudo)4.40833183936775      2.654 0.167 0
## as.factor(tpseudo)6.26052410434932      2.879 0.169 0
```

```
## as.factor(tpseudo)17.0589127898993     3.197 0.171 0
## gender                               0.821 0.101 0
```

```r
print(tc2)
```

```
##                                     estimate    se p
## (Intercept)                           -3.951 0.199 0
## as.factor(tpseudo)0.502817501664674    0.807 0.131 0
## as.factor(tpseudo)0.92759639299012     1.421 0.155 0
## as.factor(tpseudo)1.37752692252286     1.825 0.163 0
## as.factor(tpseudo)1.98500565945965     2.139 0.168 0
## as.factor(tpseudo)2.63087642829617     2.430 0.172 0
## as.factor(tpseudo)3.41362053033964     2.721 0.176 0
## as.factor(tpseudo)4.40833183936775     2.998 0.178 0
## as.factor(tpseudo)6.26052410434932     3.310 0.181 0
## as.factor(tpseudo)17.0589127898993     3.784 0.188 0
## gender                                 1.072 0.131 0
```

Pseudovalue approach 100 time points

```r
## 100 time points




data_pseudo <- data1[data1$epsilon != 0, ]
data_pseudo <- data_pseudo %>%
  mutate(epsilon = as.integer(epsilon))

summary(data_pseudo$epsilon)
```

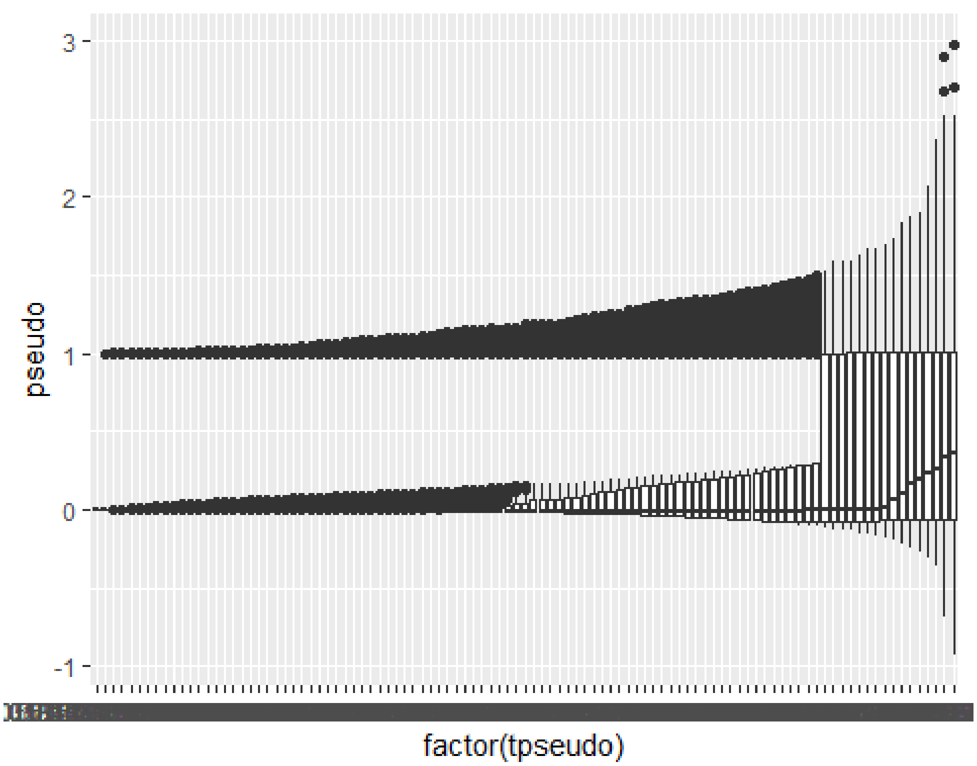
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   1.599   2.000   2.000
```

```r
# Quantiles
quantile(data_pseudo$time, probs = seq(0,1, 0.01) )
```

```
##          0%          1%          2%          3%          4%
5%
##  0.005675404  0.018086188  0.053401015  0.072313979  0.092687655
0.107134132
##          6%          7%          8%          9%         10%
11%
##  0.147373445  0.184132310  0.199110891  0.215204663  0.243376484
0.259026143
##         12%         13%         14%         15%         16%
17%
##  0.284510585  0.320371379  0.337560376  0.353675889  0.382210549
0.440607390
##         18%         19%         20%         21%         22%
23%
##  0.466002432  0.480678821  0.502817502  0.523044874  0.542745998
```

```
0.588999292
##         24%         25%         26%         27%         28%
29%
##  0.625677086  0.666234632  0.710779456  0.771370099  0.813328383
0.848582064
##         30%         31%         32%         33%         34%
35%
##  0.927596393  0.986546117  1.017785432  1.044753077  1.103417569
1.142111883
##         36%         37%         38%         39%         40%
41%
##  1.169768796  1.203696274  1.254623864  1.309545188  1.377526923
1.432349961
##         42%         43%         44%         45%         46%
47%
##  1.521553009  1.593043712  1.633083194  1.703951153  1.741785519
1.791551395
##         48%         49%         50%         51%         52%
53%
##  1.837404171  1.920566351  1.985005659  2.016727821  2.058749811
2.112691827
##         54%         55%         56%         57%         58%
59%
##  2.138583175  2.216793080  2.302139858  2.405363653  2.492961812
2.551991687
##         60%         61%         62%         63%         64%
65%
##  2.630876428  2.727625497  2.822112603  2.871366513  2.972732354
3.076843096
##         66%         67%         68%         69%         70%
71%
##  3.156463676  3.239629995  3.288881908  3.362578783  3.413620530
3.487475462
##         72%         73%         74%         75%         76%
77%
##  3.567850519  3.708096562  3.764394091  3.882526932  4.033185741
4.086828684
##         78%         79%         80%         81%         82%
83%
##  4.210845593  4.315648537  4.408331839  4.496661654  4.710241562
4.934566658
##         84%         85%         86%         87%         88%
89%
##  5.099765464  5.246459890  5.435543577  5.650597553  5.928524900
6.068290132
##         90%         91%         92%         93%         94%
95%
##  6.260524104  6.445983333  6.729223560  7.090335440  7.417239568
8.232918109
```

```
##          96%         97%          98%          99%         100%
##  8.751808993  9.527643956 10.396241727 13.034475587 17.058912790

t_pts <-  quantile(data_pseudo$time, probs = seq(0,1, 0.01) )

data_pseudo <- data1 %>%
  mutate(epsilon = as.integer(epsilon))

pseudo <- pseudoci(time = data_pseudo$time, event = data_pseudo$epsilon, tmax
= t_pts)

# Cause1
b <- NULL
for(it in 1:length(pseudo$time)){
    b <- rbind(b,cbind(data_pseudo,pseudo = pseudo$pseudo$cause1[,it],
          tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
b <- b[order(b$id),]

b$tpseudo <- factor(b$tpseudo)

ggplot(b, aes(x = factor(tpseudo), y = pseudo)) + geom_boxplot()
```




```
# fit the model
library(geepack)
```

```r
##### CAUSE 1
fit_c1 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")


#The results using the AJ variance estimate
h1 <- cbind(mean = round(fit_c1$beta,4), SD =
round(sqrt(diag(fit_c1$vbeta.ajs)),4),
    Z = round(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)),4),
    PVal = round(2-2*pnorm(abs(fit_c1$beta/sqrt(diag(fit_c1$vbeta.ajs)))),4))

# Logit link
fit_c12 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =b, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")


#The results using the AJ variance estimate
h2 <- cbind(mean = round(fit_c12$beta ,4), SD =
round(sqrt(diag(fit_c12$vbeta.ajs)),4),
    Z = round(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)),4),
    PVal = round(2-
2*pnorm(abs(fit_c12$beta/sqrt(diag(fit_c12$vbeta.ajs)))),4))




#### Cause 2

c <- NULL
for(it in 1:length(pseudo$time)){
    c <- rbind(c,cbind(data_pseudo,pseudo = pseudo$pseudo$cause2[,it],
         tpseudo = pseudo$time[it],id=1:nrow(data_pseudo)))
}
c <- c[order(c$id),]


fit_c2 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "cloglog", corstr="independence")


#The results using the AJ variance estimate
c1 <- cbind(mean = round(fit_c2$beta,3), SD =
round(sqrt(diag(fit_c2$vbeta.ajs)),3),
    Z = round(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)),3),
```

```
    PVal = round(2-2*pnorm(abs(fit_c2$beta/sqrt(diag(fit_c2$vbeta.ajs)))),3))

fit_c22 <- geese(pseudo ~ as.factor(tpseudo) + gender, data =c, id=id, jack =
TRUE, scale.fix=TRUE, family=gaussian,
    mean.link = "logit", corstr="independence")

#The results using the AJ variance estimate
c2 <- cbind(mean = round(fit_c22$beta,3), SD =
round(sqrt(diag(fit_c22$vbeta.ajs)),3),
    Z = round(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)),3),
    PVal = round(2-
2*pnorm(abs(fit_c22$beta/sqrt(diag(fit_c22$vbeta.ajs)))),3))



th1 <- data.frame(covariate = h1[,0], estimate = h1[,1], se = h1[,2], p =
h1[,4])

th2 <- data.frame(covariate = h2[,0], estimate = h2[,1], se = h2[,2], p =
h2[,4])

tc1 <- data.frame(covariate = c1[,0], estimate = c1[,1], se = c1[,2], p =
c1[,4])

tc2 <- data.frame(covariate = c2[,0], estimate = c2[,1], se = c2[,2], p =
c2[,4])

print(th1)

##                                      estimate     se      p
## (Intercept)                          -19.5686 0.0550 0.0000
## as.factor(tpseudo)0.018086187573991   13.8426 0.5581 0.0000
## as.factor(tpseudo)0.0534010152901654  14.4880 0.3928 0.0000
## as.factor(tpseudo)0.0723139785345689  14.8031 0.3442 0.0000
## as.factor(tpseudo)0.0926876547919063  15.3096 0.2750 0.0000
## as.factor(tpseudo)0.10713413198684    15.5390 0.2543 0.0000
## as.factor(tpseudo)0.147373444563405   15.7883 0.2211 0.0000
## as.factor(tpseudo)0.18413230959094    15.9641 0.2040 0.0000
## as.factor(tpseudo)0.199110891010409   16.0816 0.1929 0.0000
## as.factor(tpseudo)0.215204662626433   16.3002 0.1827 0.0000
## as.factor(tpseudo)0.243376484321858   16.4023 0.1855 0.0000
## as.factor(tpseudo)0.259026142651327   16.4652 0.1808 0.0000
## as.factor(tpseudo)0.28451058547279    16.5675 0.1751 0.0000
## as.factor(tpseudo)0.320371378967939   16.6352 0.1701 0.0000
## as.factor(tpseudo)0.337560375650488   16.7175 0.1636 0.0000
## as.factor(tpseudo)0.353675888899147   16.8045 0.1581 0.0000
## as.factor(tpseudo)0.382210549216773   16.8430 0.1564 0.0000
## as.factor(tpseudo)0.440607389500042   16.8803 0.1614 0.0000
## as.factor(tpseudo)0.466002432301572   16.9363 0.1523 0.0000
```

```
## as.factor(tpseudo)0.480678820691241   17.0040 0.1507 0.0000
## as.factor(tpseudo)0.502817501664674   17.0364 0.1508 0.0000
## as.factor(tpseudo)0.523044874376719   17.0680 0.1508 0.0000
## as.factor(tpseudo)0.54274599783201    17.1285 0.1479 0.0000
## as.factor(tpseudo)0.588999292096211   17.1538 0.1433 0.0000
## as.factor(tpseudo)0.625677085734893   17.2104 0.1480 0.0000
## as.factor(tpseudo)0.666234631588776   17.2459 0.1438 0.0000
## as.factor(tpseudo)0.710779456363122   17.2576 0.1431 0.0000
## as.factor(tpseudo)0.771370099248708   17.2840 0.1424 0.0000
## as.factor(tpseudo)0.813328383078736   17.3098 0.1400 0.0000
## as.factor(tpseudo)0.848582063568167   17.3243 0.1406 0.0000
## as.factor(tpseudo)0.92759639299012    17.3571 0.1383 0.0000
## as.factor(tpseudo)0.986546116953409   17.3850 0.1362 0.0000
## as.factor(tpseudo)1.01778543185269    17.3956 0.1326 0.0000
## as.factor(tpseudo)1.04475307726515    17.4460 0.1356 0.0000
## as.factor(tpseudo)1.10341756926076    17.4719 0.1367 0.0000
## as.factor(tpseudo)1.14211188260644    17.4917 0.1347 0.0000
## as.factor(tpseudo)1.16976879611643    17.5264 0.1348 0.0000
## as.factor(tpseudo)1.20369627359565    17.5481 0.1315 0.0000
## as.factor(tpseudo)1.25462386421356    17.5813 0.1306 0.0000
## as.factor(tpseudo)1.3095451877122     17.5905 0.1342 0.0000
## as.factor(tpseudo)1.37752692252286    17.6204 0.1329 0.0000
## as.factor(tpseudo)1.43234996111055    17.6320 0.1346 0.0000
## as.factor(tpseudo)1.52155300933151    17.6613 0.1301 0.0000
## as.factor(tpseudo)1.59304371194989    17.7099 0.1298 0.0000
## as.factor(tpseudo)1.63308319406145    17.7377 0.1274 0.0000
## as.factor(tpseudo)1.70395115253662    17.7590 0.1247 0.0000
## as.factor(tpseudo)1.74178551932295    17.8143 0.1296 0.0000
## as.factor(tpseudo)1.79155139543487    17.8442 0.1283 0.0000
## as.factor(tpseudo)1.83740417108823    17.8790 0.1225 0.0000
## as.factor(tpseudo)1.92056635138114    17.9035 0.1292 0.0000
## as.factor(tpseudo)1.98500565945965    17.9225 0.1237 0.0000
## as.factor(tpseudo)2.01672782057449    17.9832 0.1279 0.0000
## as.factor(tpseudo)2.05874981079251    17.9903 0.1278 0.0000
## as.factor(tpseudo)2.11269182742096    18.0045 0.1267 0.0000
## as.factor(tpseudo)2.13858317504994    18.0133 0.1263 0.0000
## as.factor(tpseudo)2.21679308038693    18.0380 0.1294 0.0000
## as.factor(tpseudo)2.30213985790129    18.0555 0.1303 0.0000
## as.factor(tpseudo)2.40536365257866    18.0780 0.1243 0.0000
## as.factor(tpseudo)2.49296181178326    18.1192 0.1291 0.0000
## as.factor(tpseudo)2.55199168734405    18.1410 0.1266 0.0000
## as.factor(tpseudo)2.63087642829617    18.1609 0.1233 0.0000
## as.factor(tpseudo)2.72762549715672    18.1922 0.1228 0.0000
## as.factor(tpseudo)2.82211260254507    18.2052 0.1234 0.0000
## as.factor(tpseudo)2.87136651320306    18.2279 0.1221 0.0000
## as.factor(tpseudo)2.97273235380936    18.2631 0.1222 0.0000
## as.factor(tpseudo)3.0768430956438     18.2837 0.1199 0.0000
## as.factor(tpseudo)3.15646367594812    18.3073 0.1235 0.0000
## as.factor(tpseudo)3.23962999519119    18.3352 0.1223 0.0000
## as.factor(tpseudo)3.28888190835714    18.3505 0.1222 0.0000
```

```
## as.factor(tpseudo)3.36257878339462    18.3644 0.1214 0.0000
## as.factor(tpseudo)3.41362053033964    18.3782 0.1189 0.0000
## as.factor(tpseudo)3.48747546228567    18.3994 0.1218 0.0000
## as.factor(tpseudo)3.56785051868023    18.4204 0.1207 0.0000
## as.factor(tpseudo)3.70809656213344    18.4498 0.1218 0.0000
## as.factor(tpseudo)3.76439409073899    18.4571 0.1249 0.0000
## as.factor(tpseudo)3.88252693207279    18.4691 0.1215 0.0000
## as.factor(tpseudo)4.03318574130535    18.4911 0.1228 0.0000
## as.factor(tpseudo)4.08682868355602    18.5319 0.1240 0.0000
## as.factor(tpseudo)4.2108455930892     18.5650 0.1203 0.0000
## as.factor(tpseudo)4.31564853650456    18.5779 0.1220 0.0000
## as.factor(tpseudo)4.40833183936775    18.5920 0.1208 0.0000
## as.factor(tpseudo)4.49666165357747    18.6178 0.1215 0.0000
## as.factor(tpseudo)4.71024156231433    18.6435 0.1232 0.0000
## as.factor(tpseudo)4.93456665794567    18.6505 0.1228 0.0000
## as.factor(tpseudo)5.09976546400902    18.6694 0.1215 0.0000
## as.factor(tpseudo)5.24645989015698    18.6896 0.1217 0.0000
## as.factor(tpseudo)5.43554357677849    18.7183 0.1198 0.0000
## as.factor(tpseudo)5.65059755308551    18.7255 0.1224 0.0000
## as.factor(tpseudo)5.92852490041405    18.7438 0.1252 0.0000
## as.factor(tpseudo)6.06829013151397    18.7731 0.1215 0.0000
## as.factor(tpseudo)6.26052410434932    18.7878 0.1214 0.0000
## as.factor(tpseudo)6.44598333298888    18.8151 0.1232 0.0000
## as.factor(tpseudo)6.72922355980835    18.8299 0.1218 0.0000
## as.factor(tpseudo)7.09033543994912    18.8438 0.1208 0.0000
## as.factor(tpseudo)7.4172395684572     18.8756 0.1223 0.0000
## as.factor(tpseudo)8.23291810876361    18.8981 0.1213 0.0000
## as.factor(tpseudo)8.75180899334671    18.9143 0.1218 0.0000
## as.factor(tpseudo)9.5276439563827     18.9390 0.1211 0.0000
## as.factor(tpseudo)10.3962417267358    18.9578 0.1220 0.0000
## as.factor(tpseudo)13.0344755865671    19.0119 0.1224 0.0000
## as.factor(tpseudo)17.0589127898993    19.0368 0.1210 0.0000
## gender                                -0.2850 0.1149 0.0132
```

```r
print(th2)
```

```
##                                      estimate     se      p
## (Intercept)                          -17.2065 0.0547 0.0000
## as.factor(tpseudo)0.018086187573991   11.4879 0.5621 0.0000
## as.factor(tpseudo)0.0534010152901654  12.1287 0.3954 0.0000
## as.factor(tpseudo)0.0723139785345689  12.4482 0.3481 0.0000
## as.factor(tpseudo)0.0926876547919063  12.9597 0.2802 0.0000
## as.factor(tpseudo)0.10713413198684    13.1931 0.2592 0.0000
## as.factor(tpseudo)0.147373444563405   13.4423 0.2261 0.0000
## as.factor(tpseudo)0.18413230959094    13.6202 0.2092 0.0000
## as.factor(tpseudo)0.199110891010409   13.7397 0.1995 0.0000
## as.factor(tpseudo)0.215204662626433   13.9641 0.1904 0.0000
## as.factor(tpseudo)0.243376484321858   14.0695 0.1890 0.0000
## as.factor(tpseudo)0.259026142651327   14.1344 0.1884 0.0000
## as.factor(tpseudo)0.28451058547279    14.2392 0.1826 0.0000
```

```
## as.factor(tpseudo)0.320371378967939   14.3085 0.1779 0.0000
## as.factor(tpseudo)0.337560375650488   14.3927 0.1714 0.0000
## as.factor(tpseudo)0.353675888899147   14.4828 0.1710 0.0000
## as.factor(tpseudo)0.382210549216773   14.5225 0.1679 0.0000
## as.factor(tpseudo)0.440607389500042   14.5610 0.1673 0.0000
## as.factor(tpseudo)0.466002432301572   14.6191 0.1665 0.0000
## as.factor(tpseudo)0.480678820691241   14.6893 0.1636 0.0000
## as.factor(tpseudo)0.502817501664674   14.7229 0.1617 0.0000
## as.factor(tpseudo)0.523044874376719   14.7557 0.1604 0.0000
## as.factor(tpseudo)0.54274599783201    14.8187 0.1597 0.0000
## as.factor(tpseudo)0.588999292096211   14.8450 0.1572 0.0000
## as.factor(tpseudo)0.625677085734893   14.9041 0.1550 0.0000
## as.factor(tpseudo)0.666234631588776   14.9412 0.1529 0.0000
## as.factor(tpseudo)0.710779456363122   14.9534 0.1517 0.0000
## as.factor(tpseudo)0.771370099248708   14.9811 0.1520 0.0000
## as.factor(tpseudo)0.813328383078736   15.0083 0.1497 0.0000
## as.factor(tpseudo)0.848582063568167   15.0235 0.1491 0.0000
## as.factor(tpseudo)0.92759639299012    15.0580 0.1482 0.0000
## as.factor(tpseudo)0.986546116953409   15.0874 0.1489 0.0000
## as.factor(tpseudo)1.01778543185269    15.0986 0.1473 0.0000
## as.factor(tpseudo)1.04475307726515    15.1519 0.1478 0.0000
## as.factor(tpseudo)1.10341756926076    15.1793 0.1480 0.0000
## as.factor(tpseudo)1.14211188260644    15.2003 0.1471 0.0000
## as.factor(tpseudo)1.16976879611643    15.2371 0.1465 0.0000
## as.factor(tpseudo)1.20369627359565    15.2602 0.1465 0.0000
## as.factor(tpseudo)1.25462386421356    15.2955 0.1455 0.0000
## as.factor(tpseudo)1.3095451877122     15.3053 0.1458 0.0000
## as.factor(tpseudo)1.37752692252286    15.3373 0.1435 0.0000
## as.factor(tpseudo)1.43234996111055    15.3497 0.1444 0.0000
## as.factor(tpseudo)1.52155300933151    15.3811 0.1438 0.0000
## as.factor(tpseudo)1.59304371194989    15.4334 0.1412 0.0000
## as.factor(tpseudo)1.63308319406145    15.4633 0.1427 0.0000
## as.factor(tpseudo)1.70395115253662    15.4861 0.1427 0.0000
## as.factor(tpseudo)1.74178551932295    15.5459 0.1418 0.0000
## as.factor(tpseudo)1.79155139543487    15.5781 0.1418 0.0000
## as.factor(tpseudo)1.83740417108823    15.6160 0.1419 0.0000
## as.factor(tpseudo)1.92056635138114    15.6427 0.1417 0.0000
## as.factor(tpseudo)1.98500565945965    15.6633 0.1422 0.0000
## as.factor(tpseudo)2.01672782057449    15.7295 0.1412 0.0000
## as.factor(tpseudo)2.05874981079251    15.7374 0.1420 0.0000
## as.factor(tpseudo)2.11269182742096    15.7531 0.1428 0.0000
## as.factor(tpseudo)2.13858317504994    15.7628 0.1411 0.0000
## as.factor(tpseudo)2.21679308038693    15.7898 0.1414 0.0000
## as.factor(tpseudo)2.30213985790129    15.8090 0.1414 0.0000
## as.factor(tpseudo)2.40536365257866    15.8340 0.1408 0.0000
## as.factor(tpseudo)2.49296181178326    15.8793 0.1417 0.0000
## as.factor(tpseudo)2.55199168734405    15.9036 0.1413 0.0000
## as.factor(tpseudo)2.63087642829617    15.9262 0.1390 0.0000
## as.factor(tpseudo)2.72762549715672    15.9609 0.1407 0.0000
## as.factor(tpseudo)2.82211260254507    15.9758 0.1388 0.0000
```

```
## as.factor(tpseudo)2.87136651320306    16.0011 0.1395 0.0000
## as.factor(tpseudo)2.97273235380936    16.0408 0.1406 0.0000
## as.factor(tpseudo)3.0768430956438     16.0642 0.1387 0.0000
## as.factor(tpseudo)3.15646367594812    16.0905 0.1388 0.0000
## as.factor(tpseudo)3.23962999519119    16.1222 0.1395 0.0000
## as.factor(tpseudo)3.28888190835714    16.1394 0.1399 0.0000
## as.factor(tpseudo)3.36257878339462    16.1552 0.1394 0.0000
## as.factor(tpseudo)3.41362053033964    16.1709 0.1400 0.0000
## as.factor(tpseudo)3.48747546228567    16.1950 0.1399 0.0000
## as.factor(tpseudo)3.56785051868023    16.2190 0.1401 0.0000
## as.factor(tpseudo)3.70809656213344    16.2521 0.1404 0.0000
## as.factor(tpseudo)3.76439409073899    16.2605 0.1401 0.0000
## as.factor(tpseudo)3.88252693207279    16.2747 0.1390 0.0000
## as.factor(tpseudo)4.03318574130535    16.2996 0.1412 0.0000
## as.factor(tpseudo)4.08682868355602    16.3466 0.1403 0.0000
## as.factor(tpseudo)4.2108455930892     16.3851 0.1405 0.0000
## as.factor(tpseudo)4.31564853650456    16.4003 0.1395 0.0000
## as.factor(tpseudo)4.40833183936775    16.4165 0.1401 0.0000
## as.factor(tpseudo)4.49666165357747    16.4469 0.1404 0.0000
## as.factor(tpseudo)4.71024156231433    16.4773 0.1414 0.0000
## as.factor(tpseudo)4.93456665794567    16.4854 0.1406 0.0000
## as.factor(tpseudo)5.09976546400902    16.5081 0.1407 0.0000
## as.factor(tpseudo)5.24645989015698    16.5319 0.1414 0.0000
## as.factor(tpseudo)5.43554357677849    16.5653 0.1416 0.0000
## as.factor(tpseudo)5.65059755308551    16.5738 0.1413 0.0000
## as.factor(tpseudo)5.92852490041405    16.5965 0.1414 0.0000
## as.factor(tpseudo)6.06829013151397    16.6309 0.1414 0.0000
## as.factor(tpseudo)6.26052410434932    16.6481 0.1415 0.0000
## as.factor(tpseudo)6.44598333298888    16.6813 0.1404 0.0000
## as.factor(tpseudo)6.72922355980835    16.6988 0.1405 0.0000
## as.factor(tpseudo)7.09033543994912    16.7158 0.1415 0.0000
## as.factor(tpseudo)7.4172395684572     16.7537 0.1413 0.0000
## as.factor(tpseudo)8.23291810876361    16.7813 0.1418 0.0000
## as.factor(tpseudo)8.75180899334671    16.8006 0.1409 0.0000
## as.factor(tpseudo)9.5276439563827     16.8310 0.1414 0.0000
## as.factor(tpseudo)10.3962417267358    16.8537 0.1420 0.0000
## as.factor(tpseudo)13.0344755865671    16.9210 0.1420 0.0000
## as.factor(tpseudo)17.0589127898993    16.9525 0.1422 0.0000
## gender                                -0.3187 0.1344 0.0177
```

```r
print(tc1)
```

```
##                                       estimate    se     p
## (Intercept)                             -7.284 0.950 0.000
## as.factor(tpseudo)0.018086187573991      1.349 0.798 0.091
## as.factor(tpseudo)0.0534010152901654     1.989 0.875 0.023
## as.factor(tpseudo)0.0723139785345689     2.505 0.905 0.006
## as.factor(tpseudo)0.0926876547919063     2.659 0.912 0.004
## as.factor(tpseudo)0.10713413198684       2.879 0.920 0.002
## as.factor(tpseudo)0.147373444563405      3.040 0.924 0.001
```

```
## as.factor(tpseudo)0.18413230959094      3.179 0.928 0.001
## as.factor(tpseudo)0.199110891010409     3.296 0.930 0.000
## as.factor(tpseudo)0.215204662626433     3.351 0.931 0.000
## as.factor(tpseudo)0.243376484321858     3.489 0.933 0.000
## as.factor(tpseudo)0.259026142651327     3.607 0.935 0.000
## as.factor(tpseudo)0.28451058547279      3.702 0.936 0.000
## as.factor(tpseudo)0.320371378967939     3.790 0.937 0.000
## as.factor(tpseudo)0.337560375650488     3.835 0.938 0.000
## as.factor(tpseudo)0.353675888899147     3.925 0.939 0.000
## as.factor(tpseudo)0.382210549216773     4.005 0.940 0.000
## as.factor(tpseudo)0.440607389500042     4.081 0.940 0.000
## as.factor(tpseudo)0.466002432301572     4.162 0.941 0.000
## as.factor(tpseudo)0.480678820691241     4.204 0.941 0.000
## as.factor(tpseudo)0.502817501664674     4.268 0.942 0.000
## as.factor(tpseudo)0.523044874376719     4.360 0.943 0.000
## as.factor(tpseudo)0.54274599783201      4.395 0.943 0.000
## as.factor(tpseudo)0.588999292096211     4.457 0.943 0.000
## as.factor(tpseudo)0.625677085734893     4.511 0.943 0.000
## as.factor(tpseudo)0.666234631588776     4.555 0.944 0.000
## as.factor(tpseudo)0.710779456363122     4.621 0.944 0.000
## as.factor(tpseudo)0.771370099248708     4.685 0.944 0.000
## as.factor(tpseudo)0.813328383078736     4.735 0.945 0.000
## as.factor(tpseudo)0.848582063568167     4.787 0.945 0.000
## as.factor(tpseudo)0.92759639299012      4.833 0.945 0.000
## as.factor(tpseudo)0.986546116953409     4.873 0.945 0.000
## as.factor(tpseudo)1.01778543185269      4.931 0.945 0.000
## as.factor(tpseudo)1.04475307726515      4.949 0.945 0.000
## as.factor(tpseudo)1.10341756926076      4.995 0.946 0.000
## as.factor(tpseudo)1.14211188260644      5.031 0.946 0.000
## as.factor(tpseudo)1.16976879611643      5.061 0.946 0.000
## as.factor(tpseudo)1.20369627359565      5.091 0.946 0.000
## as.factor(tpseudo)1.25462386421356      5.120 0.946 0.000
## as.factor(tpseudo)1.3095451877122       5.165 0.946 0.000
## as.factor(tpseudo)1.37752692252286      5.193 0.946 0.000
## as.factor(tpseudo)1.43234996111055      5.237 0.946 0.000
## as.factor(tpseudo)1.52155300933151      5.272 0.946 0.000
## as.factor(tpseudo)1.59304371194989      5.284 0.947 0.000
## as.factor(tpseudo)1.63308319406145      5.307 0.947 0.000
## as.factor(tpseudo)1.70395115253662      5.344 0.947 0.000
## as.factor(tpseudo)1.74178551932295      5.352 0.947 0.000
## as.factor(tpseudo)1.79155139543487      5.377 0.947 0.000
## as.factor(tpseudo)1.83740417108823      5.402 0.947 0.000
## as.factor(tpseudo)1.92056635138114      5.431 0.947 0.000
## as.factor(tpseudo)1.98500565945965      5.463 0.947 0.000
## as.factor(tpseudo)2.01672782057449      5.463 0.947 0.000
## as.factor(tpseudo)2.05874981079251      5.495 0.947 0.000
## as.factor(tpseudo)2.11269182742096      5.526 0.947 0.000
## as.factor(tpseudo)2.13858317504994      5.557 0.947 0.000
## as.factor(tpseudo)2.21679308038693      5.583 0.947 0.000
## as.factor(tpseudo)2.30213985790129      5.622 0.947 0.000
```

```
## as.factor(tpseudo)2.40536365257866      5.642 0.947 0.000
## as.factor(tpseudo)2.49296181178326      5.655 0.947 0.000
## as.factor(tpseudo)2.55199168734405      5.684 0.947 0.000
## as.factor(tpseudo)2.63087642829617      5.703 0.947 0.000
## as.factor(tpseudo)2.72762549715672      5.719 0.947 0.000
## as.factor(tpseudo)2.82211260254507      5.754 0.948 0.000
## as.factor(tpseudo)2.87136651320306      5.773 0.948 0.000
## as.factor(tpseudo)2.97273235380936      5.786 0.948 0.000
## as.factor(tpseudo)3.0768430956438       5.814 0.948 0.000
## as.factor(tpseudo)3.15646367594812      5.838 0.948 0.000
## as.factor(tpseudo)3.23962999519119      5.856 0.948 0.000
## as.factor(tpseudo)3.28888190835714      5.885 0.948 0.000
## as.factor(tpseudo)3.36257878339462      5.908 0.948 0.000
## as.factor(tpseudo)3.41362053033964      5.938 0.948 0.000
## as.factor(tpseudo)3.48747546228567      5.962 0.948 0.000
## as.factor(tpseudo)3.56785051868023      5.983 0.948 0.000
## as.factor(tpseudo)3.70809656213344      6.000 0.948 0.000
## as.factor(tpseudo)3.76439409073899      6.034 0.948 0.000
## as.factor(tpseudo)3.88252693207279      6.054 0.948 0.000
## as.factor(tpseudo)4.03318574130535      6.073 0.948 0.000
## as.factor(tpseudo)4.08682868355602      6.085 0.948 0.000
## as.factor(tpseudo)4.2108455930892       6.094 0.948 0.000
## as.factor(tpseudo)4.31564853650456      6.121 0.948 0.000
## as.factor(tpseudo)4.40833183936775      6.148 0.948 0.000
## as.factor(tpseudo)4.49666165357747      6.164 0.948 0.000
## as.factor(tpseudo)4.71024156231433      6.179 0.948 0.000
## as.factor(tpseudo)4.93456665794567      6.212 0.948 0.000
## as.factor(tpseudo)5.09976546400902      6.236 0.948 0.000
## as.factor(tpseudo)5.24645989015698      6.254 0.948 0.000
## as.factor(tpseudo)5.43554357677849      6.273 0.948 0.000
## as.factor(tpseudo)5.65059755308551      6.300 0.948 0.000
## as.factor(tpseudo)5.92852490041405      6.320 0.948 0.000
## as.factor(tpseudo)6.06829013151397      6.343 0.948 0.000
## as.factor(tpseudo)6.26052410434932      6.375 0.948 0.000
## as.factor(tpseudo)6.44598333298888      6.392 0.948 0.000
## as.factor(tpseudo)6.72922355980835      6.425 0.948 0.000
## as.factor(tpseudo)7.09033543994912      6.456 0.948 0.000
## as.factor(tpseudo)7.4172395684572       6.475 0.948 0.000
## as.factor(tpseudo)8.23291810876361      6.507 0.948 0.000
## as.factor(tpseudo)8.75180899334671      6.535 0.948 0.000
## as.factor(tpseudo)9.5276439563827       6.563 0.948 0.000
## as.factor(tpseudo)10.3962417267358      6.608 0.948 0.000
## as.factor(tpseudo)13.0344755865671      6.626 0.948 0.000
## as.factor(tpseudo)17.0589127898993      6.697 0.949 0.000
## gender                                  0.864 0.100 0.000
```

```
print(tc2)
```

```
##                                      estimate    se     p
## (Intercept)                            -7.466 0.953 0.000
```

```
## as.factor(tpseudo)0.018086187573991     1.305 0.792 0.099
## as.factor(tpseudo)0.0534010152901654    1.958 0.873 0.025
## as.factor(tpseudo)0.0723139785345689    2.478 0.905 0.006
## as.factor(tpseudo)0.0926876547919063    2.639 0.912 0.004
## as.factor(tpseudo)0.10713413198684      2.863 0.920 0.002
## as.factor(tpseudo)0.147373444563405     3.031 0.925 0.001
## as.factor(tpseudo)0.18413230959094      3.176 0.928 0.001
## as.factor(tpseudo)0.199110891010409     3.292 0.930 0.000
## as.factor(tpseudo)0.215204662626433     3.346 0.931 0.000
## as.factor(tpseudo)0.243376484321858     3.491 0.934 0.000
## as.factor(tpseudo)0.259026142651327     3.611 0.936 0.000
## as.factor(tpseudo)0.28451058547279      3.709 0.937 0.000
## as.factor(tpseudo)0.320371378967939     3.800 0.938 0.000
## as.factor(tpseudo)0.337560375650488     3.844 0.939 0.000
## as.factor(tpseudo)0.353675888899147     3.940 0.940 0.000
## as.factor(tpseudo)0.382210549216773     4.023 0.940 0.000
## as.factor(tpseudo)0.440607389500042     4.101 0.941 0.000
## as.factor(tpseudo)0.466002432301572     4.187 0.942 0.000
## as.factor(tpseudo)0.480678820691241     4.231 0.942 0.000
## as.factor(tpseudo)0.502817501664674     4.298 0.943 0.000
## as.factor(tpseudo)0.523044874376719     4.398 0.943 0.000
## as.factor(tpseudo)0.54274599783201      4.436 0.944 0.000
## as.factor(tpseudo)0.588999292096211     4.503 0.944 0.000
## as.factor(tpseudo)0.625677085734893     4.561 0.944 0.000
## as.factor(tpseudo)0.666234631588776     4.609 0.945 0.000
## as.factor(tpseudo)0.710779456363122     4.681 0.945 0.000
## as.factor(tpseudo)0.771370099248708     4.751 0.945 0.000
## as.factor(tpseudo)0.813328383078736     4.805 0.946 0.000
## as.factor(tpseudo)0.848582063568167     4.863 0.946 0.000
## as.factor(tpseudo)0.92759639299012      4.914 0.946 0.000
## as.factor(tpseudo)0.986546116953409     4.957 0.946 0.000
## as.factor(tpseudo)1.01778543185269      5.022 0.946 0.000
## as.factor(tpseudo)1.04475307726515      5.042 0.946 0.000
## as.factor(tpseudo)1.10341756926076      5.093 0.947 0.000
## as.factor(tpseudo)1.14211188260644      5.133 0.947 0.000
## as.factor(tpseudo)1.16976879611643      5.167 0.947 0.000
## as.factor(tpseudo)1.20369627359565      5.200 0.947 0.000
## as.factor(tpseudo)1.25462386421356      5.234 0.947 0.000
## as.factor(tpseudo)1.3095451877122       5.285 0.947 0.000
## as.factor(tpseudo)1.37752692252286      5.317 0.947 0.000
## as.factor(tpseudo)1.43234996111055      5.368 0.947 0.000
## as.factor(tpseudo)1.52155300933151      5.409 0.948 0.000
## as.factor(tpseudo)1.59304371194989      5.422 0.948 0.000
## as.factor(tpseudo)1.63308319406145      5.449 0.948 0.000
## as.factor(tpseudo)1.70395115253662      5.493 0.948 0.000
## as.factor(tpseudo)1.74178551932295      5.501 0.948 0.000
## as.factor(tpseudo)1.79155139543487      5.531 0.948 0.000
## as.factor(tpseudo)1.83740417108823      5.561 0.948 0.000
## as.factor(tpseudo)1.92056635138114      5.595 0.948 0.000
## as.factor(tpseudo)1.98500565945965      5.633 0.948 0.000
```

```
## as.factor(tpseudo)2.01672782057449      5.633 0.948 0.000
## as.factor(tpseudo)2.05874981079251      5.671 0.948 0.000
## as.factor(tpseudo)2.11269182742096      5.709 0.948 0.000
## as.factor(tpseudo)2.13858317504994      5.747 0.948 0.000
## as.factor(tpseudo)2.21679308038693      5.778 0.948 0.000
## as.factor(tpseudo)2.30213985790129      5.825 0.948 0.000
## as.factor(tpseudo)2.40536365257866      5.849 0.948 0.000
## as.factor(tpseudo)2.49296181178326      5.865 0.949 0.000
## as.factor(tpseudo)2.55199168734405      5.900 0.949 0.000
## as.factor(tpseudo)2.63087642829617      5.925 0.949 0.000
## as.factor(tpseudo)2.72762549715672      5.945 0.949 0.000
## as.factor(tpseudo)2.82211260254507      5.987 0.949 0.000
## as.factor(tpseudo)2.87136651320306      6.012 0.949 0.000
## as.factor(tpseudo)2.97273235380936      6.027 0.949 0.000
## as.factor(tpseudo)3.0768430956438       6.062 0.949 0.000
## as.factor(tpseudo)3.15646367594812      6.092 0.949 0.000
## as.factor(tpseudo)3.23962999519119      6.114 0.949 0.000
## as.factor(tpseudo)3.28888190835714      6.151 0.949 0.000
## as.factor(tpseudo)3.36257878339462      6.181 0.949 0.000
## as.factor(tpseudo)3.41362053033964      6.218 0.949 0.000
## as.factor(tpseudo)3.48747546228567      6.250 0.949 0.000
## as.factor(tpseudo)3.56785051868023      6.277 0.949 0.000
## as.factor(tpseudo)3.70809656213344      6.298 0.949 0.000
## as.factor(tpseudo)3.76439409073899      6.342 0.949 0.000
## as.factor(tpseudo)3.88252693207279      6.371 0.949 0.000
## as.factor(tpseudo)4.03318574130535      6.396 0.949 0.000
## as.factor(tpseudo)4.08682868355602      6.410 0.949 0.000
## as.factor(tpseudo)4.2108455930892       6.423 0.949 0.000
## as.factor(tpseudo)4.31564853650456      6.458 0.950 0.000
## as.factor(tpseudo)4.40833183936775      6.496 0.950 0.000
## as.factor(tpseudo)4.49666165357747      6.516 0.950 0.000
## as.factor(tpseudo)4.71024156231433      6.537 0.950 0.000
## as.factor(tpseudo)4.93456665794567      6.583 0.950 0.000
## as.factor(tpseudo)5.09976546400902      6.613 0.950 0.000
## as.factor(tpseudo)5.24645989015698      6.639 0.950 0.000
## as.factor(tpseudo)5.43554357677849      6.667 0.950 0.000
## as.factor(tpseudo)5.65059755308551      6.710 0.950 0.000
## as.factor(tpseudo)5.92852490041405      6.739 0.950 0.000
## as.factor(tpseudo)6.06829013151397      6.770 0.950 0.000
## as.factor(tpseudo)6.26052410434932      6.810 0.950 0.000
## as.factor(tpseudo)6.44598333298888      6.834 0.950 0.000
## as.factor(tpseudo)6.72922355980835      6.882 0.950 0.000
## as.factor(tpseudo)7.09033543994912      6.923 0.950 0.000
## as.factor(tpseudo)7.4172395684572       6.950 0.950 0.000
## as.factor(tpseudo)8.23291810876361      6.994 0.950 0.000
## as.factor(tpseudo)8.75180899334671      7.042 0.950 0.000
## as.factor(tpseudo)9.5276439563827       7.083 0.950 0.000
## as.factor(tpseudo)10.3962417267358      7.148 0.951 0.000
## as.factor(tpseudo)13.0344755865671      7.178 0.951 0.000
```

```
## as.factor(tpseudo)17.0589127898993      7.288 0.951 0.000
## gender                                  1.099 0.126 0.000
```

References

[1] H. Putter, “Tutorial in biostatistics: Competing risks and multi-state models Analyses using the mstate package”, [Online], May 30, 2014. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.487.3943&rep=rep1&type=pdf

[2] T. Therneau, “A package for survival analysis in R”,[Online], June 12 2020. Available: https://cran.r-project.org/web/packages/survival/vignettes/survival.pdf

[3] T. Therneau, C. Crowson, and E. Atkinson, “Multi-state models and competing risks”,[Online], June 12 2020. Available: https://cran.r-project.org/web/packages/survival/vignettes/compete.pdf

[4] J. P. Klein, M. Gerster, P. K. Andersen, S. Tarima, and M. P. Perme, “SAS and R functions to compute pseudo-values for censored data regression,” Computer Methods and Programs in Biomedicine, vol. 89, no. 3, pp. 289-300, 2007, doi: 10.1016/j.cmpb.2007.11.017.