HW_4

Wenjuan Bian

2023-04-15

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
### 8.2, 9.1, 10.4
####
# 8.2. Consider the following synthetic time dependent data:

## id wait.time futime fustat transplant
## 1 12 58 1 1
## 2 - 8 1 0
## 3 - 37 1 0
## 4 18 28 1 1
## 5 - 35 1 0
## 6 17 77 1 1
# First model the data ignoring the wait time. Then transform the data
into startstop format, then use that form of the data to model "transpl
ant" as a time dependent covariate. Write out the partial likelihood fo
r these data, and use this partial likelihood to find the maximum parti
al likelihood estimate of the coefficient for transplant. Compare your
answer to the results of "coxph".

library(survival)
library(asaur)
library(coxme)

## Warning: package 'coxme' was built under R version 4.2.3

## Loading required package: bdsmatrix

##
## Attaching package: 'bdsmatrix'

## The following object is masked from 'package:base':
##
##     backsolve
```

```r
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.2.3
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
id <- c(1, 2, 3, 4, 5, 6)
wait.time <- c(12, NA, NA, 18, NA, 17)
futime <- c(58, 8, 37, 28, 35, 77)
fustat <- c(1, 1, 1, 1, 1, 1)
transplant <- c(1, 0, 0, 1, 0, 1)
ndata<- data.frame(id, wait.time, futime, fustat, transplant)

cox.ndata <- coxph(Surv(futime, fustat) ~ transplant, data=ndata)

summary(cox.ndata)
```

```
## Call:
## coxph(formula = Surv(futime, fustat) ~ transplant, data = ndata)
##
##   n= 6, number of events= 6
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## transplant -1.395     0.248    1.164 -1.198    0.231
##
##            exp(coef) exp(-coef) lower .95 upper .95
## transplant     0.248      4.033   0.02531     2.429
##
## Concordance= 0.667  (se = 0.122 )
## Likelihood ratio test= 1.69  on 1 df,   p=0.2
## Wald test            = 1.43  on 1 df,   p=0.2
## Score (logrank) test = 1.67  on 1 df,   p=0.2
```

```r
sdata <- tmerge(ndata, ndata, id=id,
                death=event(futime, fustat),
                transpl=tdc(wait.time))
sdata.counting <- sdata[,-(2:5)]
sdata.counting
```

```
##   id tstart tstop death transpl
## 1  1      0    12     0       0
## 2  1     12    58     1       1
## 3  2      0     8     1       0
## 4  3      0    37     1       0
## 5  4      0    18     0       0
## 6  4     18    28     1       1
```

```
## 7  5      0    35     1       0
## 8  6      0    17     0       0
## 9  6     17    77     1       1
```

```
coxph.sdata <- coxph(Surv(tstart, tstop, death) ~ transpl,
data=sdata.counting)
summary(coxph.sdata)
```

```
## Call:
## coxph(formula = Surv(tstart, tstop, death) ~ transpl, data = sdata.c
ounting)
##
##   n= 9, number of events= 6
##
##           coef exp(coef) se(coef)      z Pr(>|z|)
## transpl -1.073     0.342    1.235 -0.869    0.385
##
##         exp(coef) exp(-coef) lower .95 upper .95
## transpl     0.342      2.924   0.03036     3.851
##
## Concordance= 0.567  (se = 0.113 )
## Likelihood ratio test= 0.81  on 1 df,   p=0.4
## Wald test            = 0.75  on 1 df,   p=0.4
## Score (logrank) test = 0.83  on 1 df,   p=0.4
```

```
coef_partial_likelihood <- coef(coxph.sdata)["transpl"]
```

```
coef_coxph <- coef(cox.ndata)["transplant"]
```

```
cat("Coefficient estimate from partial likelihood:", coef_partial_likel
ihood, "\n")
```

```
## Coefficient estimate from partial likelihood: -1.073068
```

```
cat("Coefficient estimate from Cox model ignoring wait time:", coef_cox
ph, "\n")
```

```
## Coefficient estimate from Cox model ignoring wait time: -1.394494
```

The estimated coefficient for "transplant" from partial likelihood is -1.073068, which is smaller than the coefficient estimate of -1.394494 for "transplant" from the Cox proportional hazards model ignoring wait time.

```
####
#9.1. Using the “ashkenazi” data of Sect. 9.1, use “coxme” to fit a ran
dom effects model without the “mutant” fixed effect term. How does the
estimate of the variance of the random effect from this model compare t
o that from the model that includes “mutant” as a fixed effect?
####
```

```
ashkenazi[ashkenazi$famID %in% c(1, 9, 94), ]
```

```
##    famID brcancer age mutant
## 1      1        0  73      0
## 2      1        0  40      0
## 7      9        0  89      0
## 8      9        1  60      0
## 87    94        1  44      1
## 88    94        0  45      1
```

```
result.coxme <- coxme(Surv(age, brcancer) ~ mutant + (1|famID),
data=ashkenazi)
summary(result.coxme)
```

```
## Cox mixed-effects model fit by maximum likelihood
##   Data: ashkenazi
##   events, n = 473, 3920
##   Iterations= 10 63
##                     NULL Integrated    Fitted
## Log-likelihood -3579.707  -3564.622 -3411.522
##
##                    Chisq    df          p   AIC     BIC
## Integrated loglik  30.17   2.0 2.8100e-07 26.17   17.85
##  Penalized loglik 336.37 150.1 2.2204e-16 36.16 -588.13
##
## Model:  Surv(age, brcancer) ~ mutant + (1 | famID)
## Fixed coefficients
##            coef exp(coef)  se(coef)    z       p
## mutant 1.236609  3.443914 0.2205358 5.61 2.1e-08
##
## Random effects
##  Group Variable  Std Dev   Variance
##  famID Intercept 0.5912135 0.3495334
```

```
result.coxme2 <- coxme(Surv(age, brcancer) ~ (1|famID),
data=ashkenazi)
summary(result.coxme2)
```

```
## Cox mixed-effects model fit by maximum likelihood
##   Data: ashkenazi
##   events, n = 473, 3920
##   Iterations= 25 103
##                     NULL Integrated    Fitted
## Log-likelihood -3579.707  -3577.011 -3403.742
##
##                    Chisq    df          p   AIC     BIC
## Integrated loglik   5.39   1.0 2.0242e-02  3.39   -0.77
##  Penalized loglik 351.93 168.6 5.3291e-15 14.72 -686.51
##
## Model:  Surv(age, brcancer) ~ (1 | famID)
```

```
##
## Random effects
##  Group Variable  Std Dev   Variance
##  famID Intercept 0.6311421 0.3983403
```

The estimated variance of the random effect for famID is larger in the model without "mutant" fixed effect, 0.3983 vs 0.3495, indicating higher variability in the hazard ratio among families. When the fixed effect is included in the model, it can explain some of the variability in the outcome, which may result in a smaller estimated variance of the random effect. In the model without the fixed effect, the estimated variance of the random effect may be larger as it has to capture all the unexplained variability in the outcome.

```
###
# 10.4. Using the "ashkenazi" data in the "asaur" package, fit a Weibul
l distribution to the women with the "wild type" (non-mutant) BRCA geno
type, matching the Kaplan-Meier survival curve at ages 45 and 65. Then
predict the probability that a woman with the wild type BRCA genotype w
ill develop breast cancer before the age of 70.


library(asaur)
ashkenazi[ashkenazi$famID %in% c(1, 9, 94), ]

##    famID brcancer age mutant
## 1      1        0  73      0
## 2      1        0  40      0
## 7      9        0  89      0
## 8      9        1  60      0
## 87    94        1  44      1
## 88    94        0  45      1

wild_type <- subset(ashkenazi, mutant == "0")

head(wild_type)

##   famID brcancer age mutant
## 1     1        0  73      0
## 2     1        0  40      0
## 3     7        0  48      0
## 4     7        0  25      0
## 5     8        0  56      0
## 6     8        0  55      0

fit.weibull <- survreg(Surv(age, brcancer) ~ 1, data = wild_type, dist
= "weibull")
fit.weibull
```

```
## Call:
## survreg(formula = Surv(age, brcancer) ~ 1, data = wild_type,
##     dist = "weibull")
##
## Coefficients:
## (Intercept)
##    4.773509
##
## Scale= 0.2810826
##
## Loglik(model)= -2899.2   Loglik(intercept only)= -2899.2
## n= 3830
```

```
result.km.b <- survfit(Surv(age, brcancer) ~ 1,data = wild_type)

print(result.km.b)
```

```
## Call: survfit(formula = Surv(age, brcancer) ~ 1, data = wild_type)
##
##         n events median 0.95LCL 0.95UCL
## [1,] 3830    446     NA      NA      NA
```

```
result.summ <- summary(result.km.b, time=c(45, 65))
t.vec <- result.summ$time
t.vec
```

```
## [1] 45 65
```

```
s.vec <- result.summ$surv
s.vec
```

```
## [1] 0.9652765 0.8676570
```

```
data.frame(t.vec, s.vec)
```

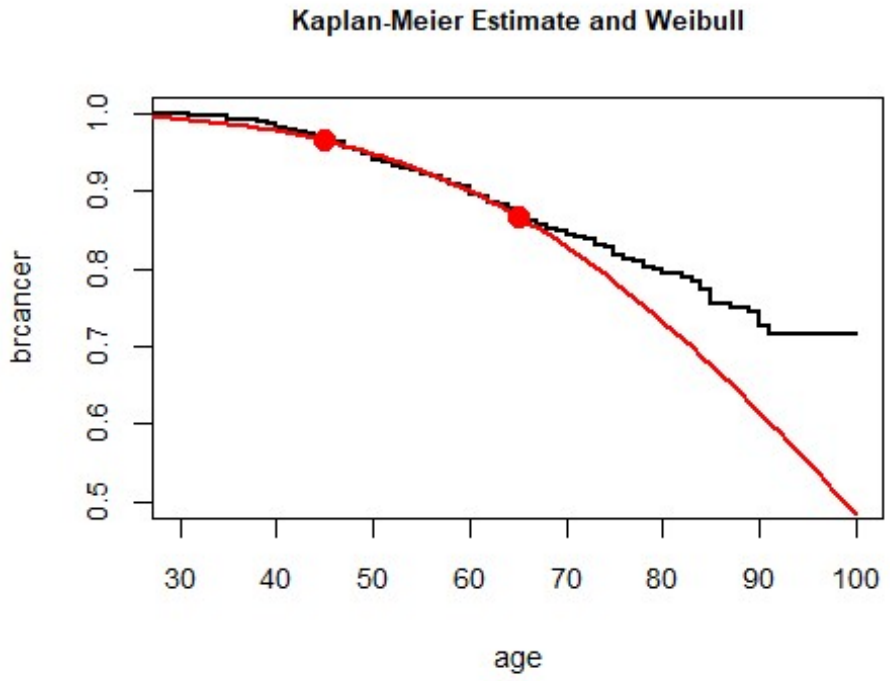
```
##   t.vec     s.vec
## 1    45 0.9652765
## 2    65 0.8676570
```

```
brWeib <- Weibull2(t.vec, s.vec)
t.vals <- 0:100
s.vals <- brWeib(t.vals)
brWeib
```

```
## function (times = NULL, alpha = 1.98092148681984e-08, gamma = 3.7813
6944778285)
## {
##     exp(-alpha * (times^gamma))
## }
## <environment: 0x000001b5609c0b58>
```

```
# Plot the KM estimate

plot(result.km.b, main = "Kaplan-Meier Estimate", xlab = "age", ylab =
"brcancer", xlim= c(30, 100), ylim=c(0.5, 1),conf.int=F, lwd=2, cex.axi
s=0.9, cex.lab=0.9)
lines(s.vals ~ t.vals, col="red", lwd=2)
points(t.vec, s.vec, col="red", pch=16, cex=1.5)
```




```
h_vals <- -log(s.vals)


t_val <- 70
h_val.70 <- h_vals[t.vals == t_val]
h_val.70
```

```
## [1] 0.1878731
```

```
cat("The predicted probability that a woman with the wild type BRCA gen
otype will develop breast cancer before the age of 70:", h_val.70)
```

```
## The predicted probability that a woman with the wild type BRCA genot
ype will develop breast cancer before the age of 70: 0.1878731
```