R for Health Data Science

Week 09: Biostats III - GEE

Sam Stewart

2021-04-23

```
library(Hmisc)
library(kableExtra)
library(tidyverse)
library(ggplot2)
library(boomer)
library(table1)
library(HSAUR)
library(ggpubr)
library(gee)

#GEE example data - if you don't want to load a package you can load a
#dataset from it like this (though you'll still need to download the package)
data("BtheB", package = "HSAUR")
```

We're going to wrap up Biostats methods this week with an investigation of Generalized Estimating Equations (GEE). Note that I have added two old lectures to the course site (both Blackboard and GitHub) that cover the modeling methods we've discussed. These are from a summer tutorial session I ran at CH&E a couple of years ago, so they focus on STATA more than R, but they have *some* R code in them, and the methods are useful for us. The names are self-explanatory:

1. LinearLogisticSurvivalPoissonRegression.pdf
2. GEEAndImputation.pdf

1 Generalized Estimating Equations

For analyzing multiple values from a subject over time

- Measuring weight every month for 12 months
- A patient's med count exery time they visit the ER

The challenge is the issue of *dependence* that arises with sequential observations - we need methods to account for that. Note that the mixed effects models we learned last week (and the `lmer()` and `glmer()` functions from `lme4`) can fit this data as well - I prefer the flexibility of GEE, but each have their advantages.

I'm not going to go too deep into it, but in building GEE models you need to specify the *covariance structure* - that is, how the observations within a group are correlated with one another. We'll see through the example below.

The data are from an interactive multimedia program called "Beat the Blues", a CBT delivered to depressed patients online. Patients were randomly assigned to BtB or Treatment as Usual (TAU). The outcome is the "Beck Depression Inventory" with predictors of centre, treatment, duration of depression and medication use.

```
#make it a tibble for ease of printing, and add an ID column
BtheB = BtheB %>% tibble() %>%
  mutate(ID = 1:n()) %>%
  relocate(ID)
#make a long version of the bdi variables
BtheB.long =
BtheB %>%
  pivot_longer(cols=-c(ID, drug,length,treatment),names_to='time',values_to='beck') %>%
  mutate(time=factor(gsub("bdi.","",time),levels=c("pre","2m","4m","6m","8m")))

#some functions use labels for better printing
label(BtheB$drug) = 'On depression medications'
label(BtheB$length) = 'Duration of depressive symptoms'

#table1 will give us a quick summary
table1(~.|treatment,data=BtheB)
```

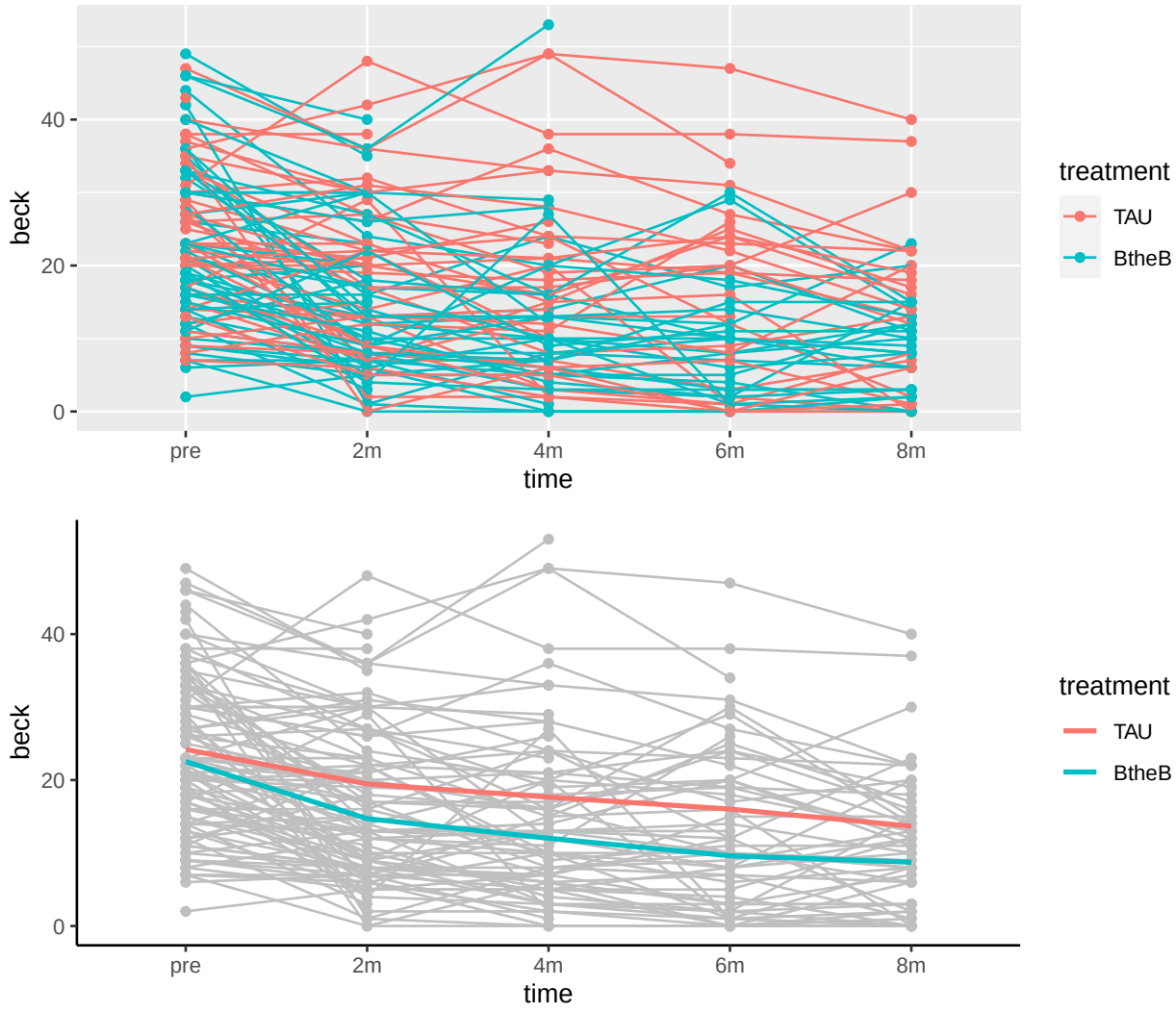
```
## [1] "<table class=\"Rtable1\">\n<thead>\n<tr>\n<th class='rowlabel firstrow lastrow'></th>\n<th clas
```

1.1 Longitudinal Plots

Imbalance in drug-use between groups, many missing values in follow-up. Let's try to plot the trajectories (this is a fun one to build in parts with `ggplot()`)

```
p1 = BtheB.long %>%
  ggplot(aes(x=time,y=beck,color=treatment))+
  geom_point()+
  geom_line(aes(group=ID))

p2 = BtheB.long %>%
  ggplot(aes(x=time,y=beck))+
  geom_point(colour=grey(0.75))+
  geom_line(aes(group=ID),colour=grey(0.75))+
  geom_smooth(aes(colour=treatment,group=treatment),se=FALSE)+
  theme_classic()
#this function from the ggpubr library is useful for combining ggplots
ggarrange(p1,p2,nrow=2)
```



Obviously a lot of variability in the trajectories, but two things of note are that the TAU subjects seem to do worse (with Beck low == good) but that they also start higher, and that the baseline Beck scores seem to vary significantly between groups.

We're going to have to incorporate the baseline Beck score into the model, and we'll experiment with a couple of different correlational structures.

```
#I'm going to move bdi.pre from an outcome measure to a predictor
BtheB.long.model =
BtheB %>%
  pivot_longer(cols=-c(ID, drug,length,treatment,bdi.pre),
               names_to='time',values_to='bdi') %>%
  mutate(time=factor(gsub("bdi.","",time),levels=c("2m","4m","6m","8m")))%>%
  arrange(ID,time)%>%
  drop_na()

BtheB.long.model
```

```
## # A tibble: 280 x 7
##       ID drug  length treatment bdi.pre time    bdi
##    <int> <fct> <fct>  <fct>       <dbl> <fct> <dbl>
```

```
##  1     1 No    >6m    TAU            29 2m        2
##  2     1 No    >6m    TAU            29 4m        2
##  3     2 Yes   >6m    BtheB          32 2m       16
##  4     2 Yes   >6m    BtheB          32 4m       24
##  5     2 Yes   >6m    BtheB          32 6m       17
##  6     2 Yes   >6m    BtheB          32 8m       20
##  7     3 Yes   <6m    TAU            25 2m       20
##  8     4 No    >6m    BtheB          21 2m       17
##  9     4 No    >6m    BtheB          21 4m       16
## 10     4 No    >6m    BtheB          21 6m       10
## # ... with 270 more rows
```

1.2 Fitting GEE Models

```
gee(formula, id,
    data, subset, na.action,
    R = NULL, b = NULL,
    tol = 0.001, maxiter = 25,
    family = gaussian, corstr = "independence",
    Mv = 1, silent = TRUE, contrasts = NULL,
    scale.fix = FALSE, scale.value = 1, v4.4compat = FALSE)
```

We use the same formula structure, but we need to set a couple of other variables.

1. `id` identifies the grouping variable - in our case that is `ID`
2. `corstr` defines the correlation structure
3. `family` defines the type of model we're fitting - you'll use `gausian` (for continuous data) and `binomial` (for logistic data) the most.

```
#to build a gee model we just need to specify the variable that
#co-ordinates the groups (ID), the family, and the correlation structure
mod01 =  gee(bdi ~ bdi.pre + treatment + length + drug,
             data = BtheB.long.model, id = ID, family = gaussian, corstr = "independence")
```

```
##    (Intercept)        bdi.pre treatmentBtheB      length>6m        drugYes
##      3.5686314      0.5818494     -3.2372285      1.4577182     -3.7412982
```

```
mod01
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Identity
##  Variance to Mean Relation: Gaussian
##  Correlation Structure:     Independent
##
## Call:
## gee(formula = bdi ~ bdi.pre + treatment + length + drug, id = ID,
##     data = BtheB.long.model, family = gaussian, corstr = "independence")
##
## Number of observations :  280
##
## Maximum cluster size   :  4
##
##
```

```
## Coefficients:
##    (Intercept)        bdi.pre treatmentBtheB      length>6m        drugYes
##      3.5686314      0.5818494     -3.2372285      1.4577182     -3.7412982
##
## Estimated Scale Parameter:  79.25813
## Number of Iterations:  1
##
## Working Correlation[1:4,1:4]
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
##
## Returned Error Value:
## [1] 0
```

```
summary(mod01)
```

```
##
##  GEE:  GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
##  gee S-function, version 4.13 modified 98/01/27 (1998)
##
## Model:
##  Link:                      Identity
##  Variance to Mean Relation: Gaussian
##  Correlation Structure:     Independent
##
## Call:
## gee(formula = bdi ~ bdi.pre + treatment + length + drug, id = ID,
##     data = BtheB.long.model, family = gaussian, corstr = "independence")
##
## Summary of Residuals:
##         Min          1Q      Median          3Q         Max
## -21.6497810  -5.8485100   0.1131663   5.5838383  28.1871039
##
##
## Coefficients:
##                  Estimate Naive S.E.   Naive z Robust S.E.   Robust z
## (Intercept)     3.5686314  1.4833349  2.405816  2.26947617  1.5724472
## bdi.pre         0.5818494  0.0563904 10.318235  0.09156455  6.3545274
## treatmentBtheB -3.2372285  1.1295569 -2.865928  1.77459534 -1.8242066
## length>6m       1.4577182  1.1380277  1.280916  1.48255866  0.9832449
## drugYes        -3.7412982  1.1766321 -3.179667  1.78271179 -2.0986557
##
## Estimated Scale Parameter:  79.25813
## Number of Iterations:  1
##
## Working Correlation
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

You can see that, like before, we can get a simple and more complete summary of the model that was fit. In this case there is no easy way to get a p-value or confidence interval, so the function below will do it for us.

```
#You need to get CIs and p-values yourself, so I built a function to do
#it for me
mySummary = function(mod){
  sum = summary(mod)
  coef = sum$coefficients[,c(1,4,5)] %>%
    as.data.frame() %>% rownames_to_column() %>% tibble() %>%
    rename(variable = rowname,
           se = `Robust S.E.`,
           z = `Robust z`)%>%
    mutate(LCL = Estimate-qnorm(0.975)*se,
           UCL = Estimate+qnorm(0.975)*se,
           pValue=round(2*(1-pnorm(abs(z))),4)
    ) %>%
    relocate(variable,Estimate,LCL,UCL)
  coef
}

mySummary(mod01) %>%
  kable(digits=c(0,3,2,2,2,2,4)) %>% kable_styling()
```

variable	Estimate	LCL	UCL	se	z	pValue
(Intercept)	3.569	-0.88	8.02	2.27	1.57	0.1158
bdi.pre	0.582	0.40	0.76	0.09	6.35	0.0000
treatmentBtheB	-3.237	-6.72	0.24	1.77	-1.82	0.0681
length>6m	1.458	-1.45	4.36	1.48	0.98	0.3255
drugYes	-3.741	-7.24	-0.25	1.78	-2.10	0.0358

This is the result of the GEE model, but I'll hold off on interpreting it because it was built on an independence correlation structure, which is rarely correct. Let's look at the exchangeable, unstructured and AR-M models.

```
#let's look at the other models
mod02 =  gee(bdi ~ bdi.pre + treatment + length + drug,
             data = BtheB.long.model, id = ID, family = gaussian, corstr = "exchangeable")
```

```
##    (Intercept)        bdi.pre treatmentBtheB      length>6m        drugYes
##      3.5686314      0.5818494     -3.2372285      1.4577182     -3.7412982
```

```
mod03 =  gee(bdi ~ bdi.pre + treatment + length + drug,
             data = BtheB.long.model, id = ID, family = gaussian, corstr = "unstructured")
```

```
##    (Intercept)        bdi.pre treatmentBtheB      length>6m        drugYes
##      3.5686314      0.5818494     -3.2372285      1.4577182     -3.7412982
```

For some reason the `AR-M` correlation structure in R breaks when we have people that have a single observation - that shouldn't happen, but we're stuck with what is programmed. The code below drops the subjects with <2 observations, then fits an AR-1 correlation structure.

```
#this breaks because there are people with only 1 observation
#mod04 =  gee(bdi ~ bdi.pre + treatment + length + drug, data = BtheB.long.model, id = ID, family = gau

#drop the people with a single observation
b2 = BtheB.long.model %>% drop_na() %>% group_by(ID) %>% filter(n()>1)
mod04 =  gee(bdi ~ bdi.pre + treatment + length + drug,
```

Table 1: Exchangeable

variable	Estimate	LCL	UCL	se	z	pValue
(Intercept)	3.023	-1.35	7.40	2.23	1.35	0.1756
bdi.pre	0.648	0.48	0.81	0.08	7.76	0.0000
treatmentBtheB	-2.169	-5.57	1.23	1.74	-1.25	0.2115
length>6m	-0.111	-3.15	2.93	1.55	-0.07	0.9428
drugYes	-3.000	-6.39	0.39	1.73	-1.73	0.0832

Table 2: Unstructured

variable	Estimate	LCL	UCL	se	z	pValue
(Intercept)	3.248	-1.16	7.66	2.25	1.44	0.1490
bdi.pre	0.624	0.46	0.79	0.09	7.30	0.0000
treatmentBtheB	-2.361	-5.76	1.04	1.73	-1.36	0.1736
length>6m	0.259	-2.78	3.30	1.55	0.17	0.8673
drugYes	-3.022	-6.40	0.36	1.72	-1.75	0.0796

```
              data = b2, id = ID, family = gaussian, corstr = "AR-M")
```

```
##    (Intercept)        bdi.pre treatmentBtheB      length>6m        drugYes
##      3.6333471      0.5397869     -3.8591219      2.5816094     -3.7636661
```

```
#another short custom function to avoid copy+paste too much
kableMod = function(mod,cap=''){
  mySummary(mod) %>%
    kable(digits=c(0,3,2,2,2,2,4),caption=cap) %>% kable_styling()
}

kableMod(mod02,cap='Exchangeable')
```

```
kableMod(mod03,cap='Unstructured')
```

```
kableMod(mod04,cap='AR-M*')
```

I'm always biased by the AR-M approach to temporal analysis, but I don't like that I had to reduce the dataset in R, so I'd probably go with the unstructured correlation matrix - you sacrifice the most degrees of freedom with that approach, but I think it's better.

The variability in results based on correlation structure changes reflects a potential sample size issue, which is also present in the width of the CIs.

```
#let's look at the four correlation matrices
cor01 = mod01$working.correlation
```

Table 3: AR-M*

variable	Estimate	LCL	UCL	se	z	pValue
(Intercept)	3.712	-1.01	8.44	2.41	1.54	0.1237
bdi.pre	0.521	0.32	0.72	0.10	5.08	0.0000
treatmentBtheB	-3.721	-7.33	-0.12	1.84	-2.02	0.0431
length>6m	2.666	-0.50	5.83	1.62	1.65	0.0988
drugYes	-2.780	-6.41	0.86	1.85	-1.50	0.1340

Table 4: Independent

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Table 5: Exchangable

1.0000	0.6758	0.6758	0.6758
0.6758	1.0000	0.6758	0.6758
0.6758	0.6758	1.0000	0.6758
0.6758	0.6758	0.6758	1.0000

```
cor02 = mod02$working.correlation
cor03 = mod03$working.correlation
cor04 = mod04$working.correlation

cor01 %>% kable(digits=4,caption='Independent') %>% kable_styling()
```

```
cor02 %>% kable(digits=4,caption='Exchangable') %>% kable_styling()
```

```
cor03 %>% kable(digits=4,caption='Unstructured') %>% kable_styling()
```

```
cor04 %>% kable(digits=4,caption='AR-M') %>% kable_styling()
```

2 Breakout Session

We'll use a second dataset from the same library called `HSAUR::respiratory`, from a multi-centre study of some treatment (details are scarce).

- 111 patients (54 on treatment).
- baseline + 4 follow-ups
- outcome: binary, resp status as poor or good
- predictors: centre, treatment, gender, age

```
data(respiratory)
```

Tasks for you with this dataset

1. Create a summary table, similar to our `table1` command above *(hint: you'll need to use `pivot_wider()` to get the data in a one-row-per-patient format)*
2. Summarize the data graphically - I think there is an interesting line plot and barplot, but anything that communicates the results is worthwhile
3. Fit a GEE model, and pick an appropriate correlation structure

Table 6: Unstructured

1.0000	0.6392	0.5163	0.4181
0.6392	1.0000	0.5746	0.4595
0.5163	0.5746	1.0000	0.5889
0.4181	0.4595	0.5889	1.0000

Table 7: AR-M

1.0000	0.6951	0.4832	0.3359
0.6951	1.0000	0.6951	0.4832
0.4832	0.6951	1.0000	0.6951
0.3359	0.4832	0.6951	1.0000