Generalized Estimating Equations

Alexander McLain

Contents

1 Fitting GEE models in R **1**

2 Health effects of air pollution **1**

3 Epilepsy randomized clinical trial **5**

1 Fitting GEE models in R

The packages `gee` and `geepack` are used for GEE models in `R`.

The major difference between gee and geepack is that geepack contains an `anova` method that allows us to compare models and perform Wald tests.

Basic Syntax for `geeglm()` from the geepack package; has a syntax very similar to `glm()`

```r
library(geepack)
```

```
geeglm(formula, family=gaussian, data, id, constr, std.err="san.se")
```

- `formula` Symbolic description of the model to be fitted
- `family` Description of the error distribution and link function
- `data` Optional dataframe
- `id` Vector that identifies the clusters (subjects)
- `constr` Working **correlation** structure: "independence", "exchangeable", "ar1", "unstructured", "userdefined"
- `offset` Offset variable
- `std.err` Type of standard error to be calculated. Default "san.se" is the robust (sandwich) estimate; use "jack" for approximate jackknife variance estimate

2 Health effects of air pollution

Here, we'll look at the Ohio dataset from geepack. Children were followed for four years, wheeze status recorded annually

```r
data(ohio) # Load the dataset
head(ohio)
```

resp	id	age	smoke
0	0	-2	0
0	0	-1	0
0	0	0	0
0	0	1	0
0	1	-2	0

resp	id	age	smoke
0	1	-1	0

```
str(ohio)
```

```
## 'data.frame':    2148 obs. of  4 variables:
##  $ resp : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ id   : int  0 0 0 0 1 1 1 1 2 2 ...
##  $ age  : int  -2 -1 0 1 -2 -1 0 1 -2 -1 ...
##  $ smoke: int  0 0 0 0 0 0 0 0 0 0 ...
```

Response is binary - fit a logistic GEE model. Treat time (age) as continuous

```
form_gee <- resp~age+smoke
fit.exch <- geeglm(form_gee, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
fit.unstr <- geeglm(form_gee, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "unstructured", std.err="san.se")

coef(fit.exch)
```

```
## (Intercept)         age       smoke
##  -1.8804253  -0.1133850   0.2650758
```

```
vcov(fit.exch)
```

	(Intercept)	age	smoke
(Intercept)	0.0129716	0.0013320	-0.0119605
age	0.0013320	0.0019233	0.0001242
smoke	-0.0119605	0.0001242	0.0315938

```
coef(summary(fit.exch))
```

	Estimate	Std.err	Wald	Pr(>|W|)
(Intercept)	-1.8804253	0.1138927	272.596505	0.0000000
age	-0.1133850	0.0438553	6.684474	0.0097256
smoke	0.2650758	0.1777465	2.224015	0.1358793

```
coef(summary(fit.unstr))
```

	Estimate	Std.err	Wald	Pr(>|W|)
(Intercept)	-1.8885638	0.1139600	274.636558	0.0000000
age	-0.1148972	0.0442384	6.745579	0.0093980
smoke	0.2534880	0.1781843	2.023840	0.1548472

```
fit.exch$geese$alpha
```

```
##     alpha
## 0.3543049
```

```
fit.unstr$geese$alpha
```

```
## alpha.1:2 alpha.1:3 alpha.1:4 alpha.2:3 alpha.2:4 alpha.3:4
## 0.3504378 0.3083144 0.3029799 0.4695527 0.3185429 0.3763820
```

Let's look at some measure of model fit. (see `?QIC` for details)

```
QIC(fit.exch)
```

```
##         QIC        QICu   Quasi Lik         CIC      params        QICC
## 1825.947681 1825.892655 -909.946328    3.027513    3.000000 1825.966347
```

```
QIC(fit.unstr)
```

```
##         QIC        QICu   Quasi Lik         CIC      params        QICC
## 1825.789976 1825.947443 -909.973722    2.921266    3.000000 1825.874167
```

```
anova(fit.exch, fit.unstr)
```

```
## Models are identical
```

```
## NULL
```

Now we'll treat time (age) as categorical

```
form_gee <- resp ~ factor(age) + smoke
fit <- geeglm(form_gee, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
summary(fit)
```

```
##
## Call:
## geeglm(formula = form_gee, family = binomial(link = "logit"),
##     data = ohio, id = id, corstr = "exchangeable", std.err = "san.se")
##
##  Coefficients:
##              Estimate  Std.err    Wald Pr(>|W|)
## (Intercept)   -1.74344  0.13740 160.995   <2e-16 ***
## factor(age)-1  0.05401  0.13230   0.167   0.6831
## factor(age)0  -0.02776  0.13878   0.040   0.8415
## factor(age)1  -0.37552  0.14670   6.552   0.0105 *
## smoke          0.27121  0.17809   2.319   0.1278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)   0.9998  0.1148
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha   0.3544  0.0636
## Number of clusters:   537  Maximum cluster size: 4
```

Test the effect of smoke using anova()

```r
fit1 <- geeglm(form_gee, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "exchangeable", std.err="san.se")

form_gee_nosmoke <- resp ~ factor(age)

fit2 <- geeglm(form_gee_nosmoke, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
anova(fit1, fit2)
```

Df	X2	P(>|Chi|)
1	2.319	0.1278

- For a geeglm object returned by `geeglm()`, the functions `drop1()`, `confint()` and `step()` do not apply; however `anova()` does apply.
- The function `esticon()` in the `doBy` package computes CI's and tests linear functions of regression parameters.

```
esticon(obj, cm, beta0, joint.test=FALSE)
```

- `obj` Model object
- `cm` Matrix specifying linear functions of the regression parameters (one linear function per row and one column for each parameter)
- `beta0` Vector of numbers
- `joint.test` If TRUE joint Wald test of the hypothesis Lbeta=beta0 is made, default is one test for each row, (Lbeta).i=beta0.i

Individual Wald test and confidence interval for each parameter

```r
library(doBy)
est <- esticon(fit, diag(5))
# Odds ratio and confidence intervals
OR.CI <- exp(cbind(est$estimate, est$lwr, est$upr))
rownames(OR.CI) <- names(coef(fit))
colnames(OR.CI) <- c("OR", "Lower OR", "Upper OR")
OR.CI
```

	OR	Lower OR	Upper OR
(Intercept)	0.1749	0.1336	0.2290
factor(age)-1	1.0555	0.8144	1.3680
factor(age)0	0.9726	0.7410	1.2767
factor(age)1	0.6869	0.5153	0.9158
smoke	1.3116	0.9251	1.8594

The referent age is -2.

Let's test for an interaction between age and smoking

```r
form_gee_inter <- resp ~ factor(age) + smoke + factor(age)*smoke
fit3 <- geeglm(form_gee_inter, family=binomial(link="logit"),
     data=ohio, id=id, corstr = "exchangeable", std.err="san.se")
coef(summary(fit3))
```

	Estimate	Std.err	Wald	Pr(>|W|)
(Intercept)	-1.6582	0.1458	129.3468	0.0000
factor(age)-1	-0.0876	0.1697	0.2667	0.6056
factor(age)0	-0.1335	0.1780	0.5626	0.4532
factor(age)1	-0.4771	0.1896	6.3291	0.0119
smoke	0.0424	0.2448	0.0299	0.8626
factor(age)-1:smoke	0.3698	0.2710	1.8620	0.1724
factor(age)0:smoke	0.2809	0.2837	0.9798	0.3222
factor(age)1:smoke	0.2696	0.2988	0.8142	0.3669

```r
anova(fit3)
```

	Df	X2	P(>|Chi|)
factor(age)	3	10.019	0.0184
smoke	1	2.319	0.1278
factor(age):smoke	3	1.974	0.5779

3 Epilepsy randomized clinical trial

Taken from: Hothorn, T., & Everitt, B. S. (2014). **A handbook of statistical analyses using R.** CRC press.

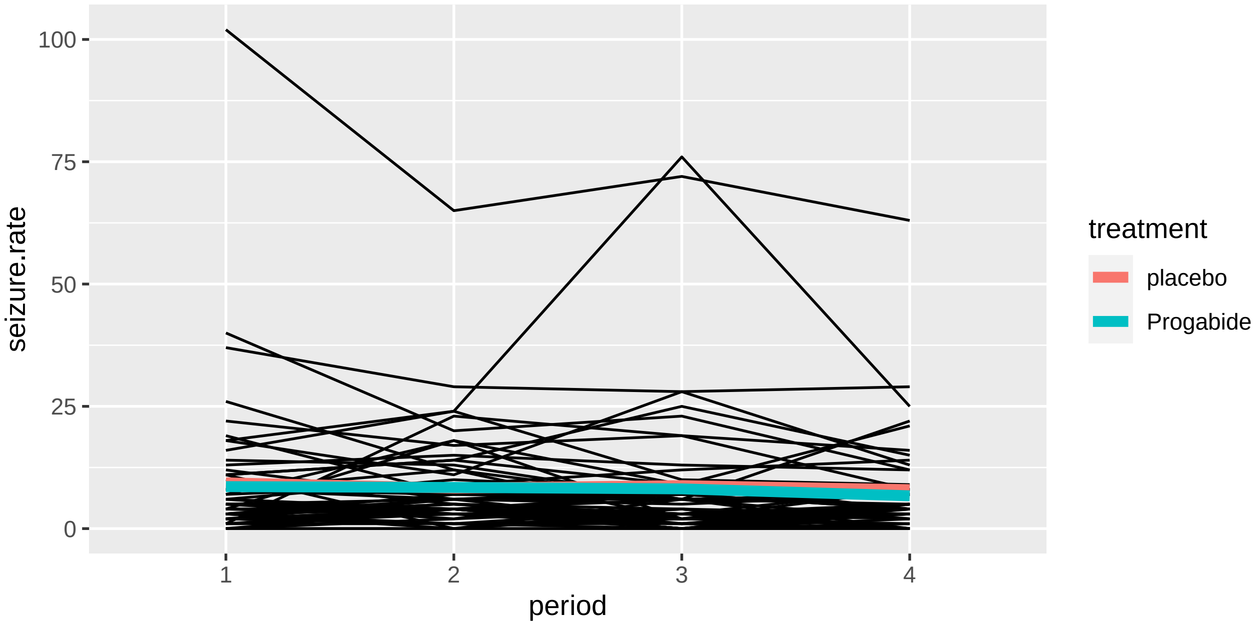
```r
library(tidyverse)
data("epilepsy", package = "HSAUR2")
head(epilepsy)
```

	treatment	base	age	seizure.rate	period	subject
1	placebo	11	31	5	1	1
110	placebo	11	31	3	2	1
112	placebo	11	31	3	3	1
114	placebo	11	31	3	4	1
2	placebo	11	30	3	1	2
210	placebo	11	30	5	2	2

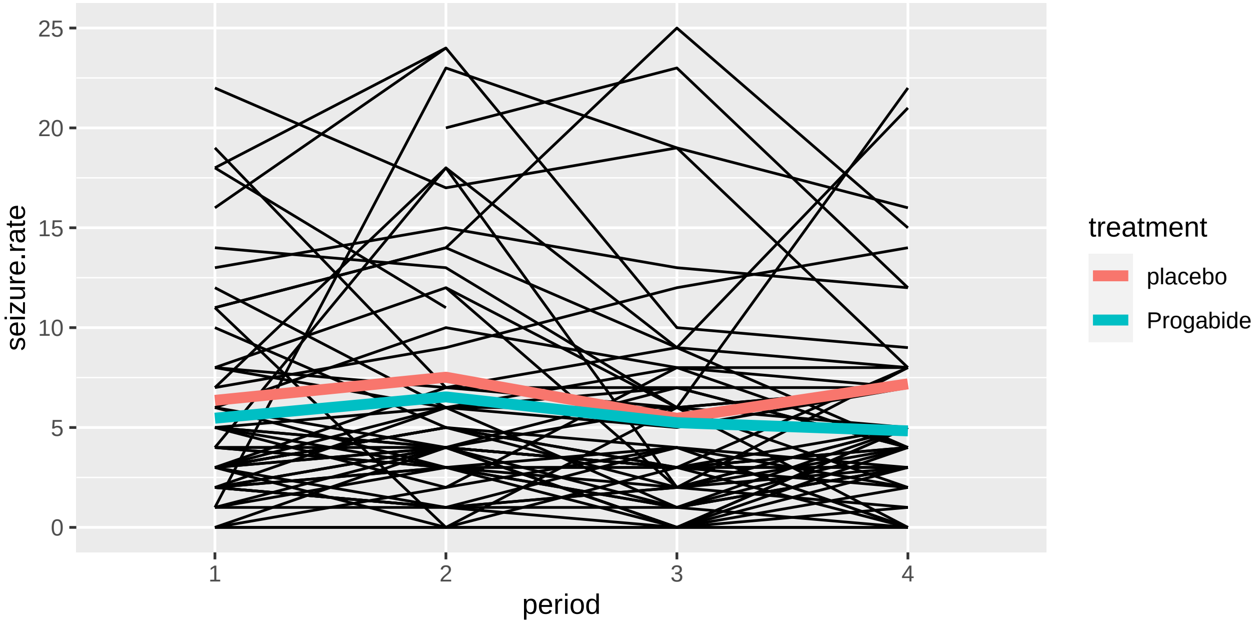
```r
str(epilepsy)
```

```
## 'data.frame':    236 obs. of  6 variables:
##  $ treatment   : Factor w/ 2 levels "placebo","Progabide": 1 1 1 1 1 1 1 1 1 1 ...
##  $ base        : int  11 11 11 11 11 11 11 11 6 6 ...
##  $ age         : int  31 31 31 31 30 30 30 30 25 25 ...
##  $ seizure.rate: int  5 3 3 3 3 5 3 3 2 4 ...
##  $ period      : Ord.factor w/ 4 levels "1"<"2"<"3"<"4": 1 2 3 4 1 2 3 4 1 2 ...
##  $ subject     : Factor w/ 59 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
```

```r
p <- ggplot(epilepsy, aes(x = period, y = seizure.rate, group = subject))
p + geom_line() + stat_summary(aes(group = treatment, color = treatment), geom = "line",
                 fun = mean, size = 2)
```

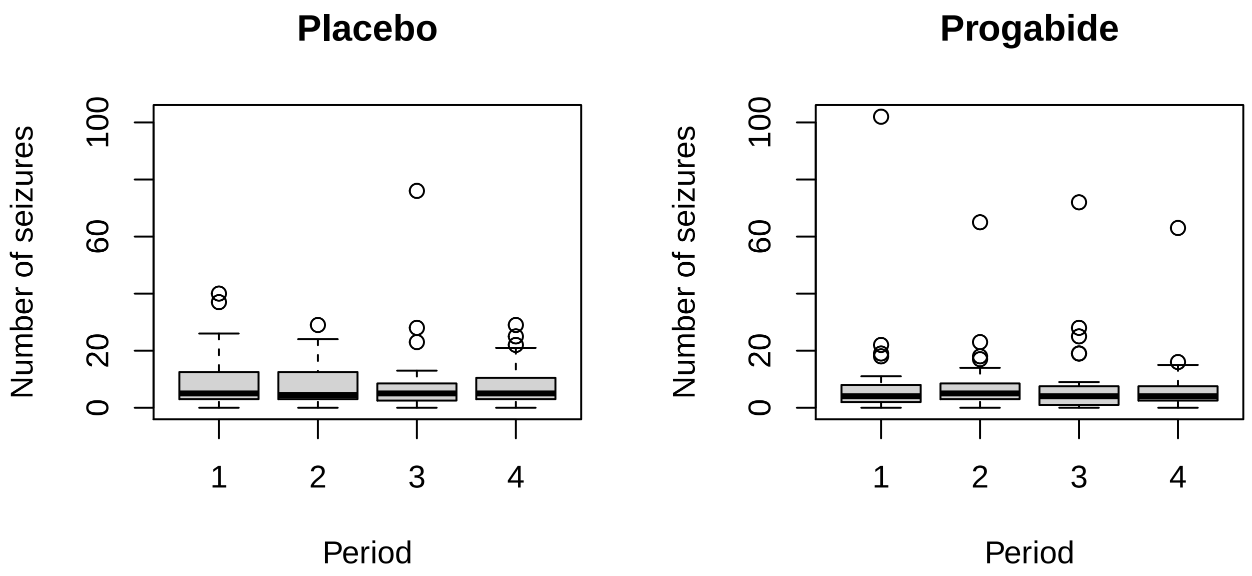


```
p + geom_line() + stat_summary(aes(group = treatment, color = treatment), geom = "line",
                  fun = mean, size = 2) + ylim(0,25)
```



```
layout(matrix(1:2, nrow = 1))

ylim <- range(epilepsy$seizure.rate)
placebo <- subset(epilepsy, treatment == "placebo")
progabide <- subset(epilepsy, treatment == "Progabide")
boxplot(seizure.rate ~ period, data = placebo,
        ylab = "Number of seizures",
        xlab = "Period", ylim = ylim, main = "Placebo")
boxplot(seizure.rate ~ period, data = progabide,
        main  = "Progabide", ylab = "Number of seizures",
        xlab = "Period", ylim = ylim)
```



```
epilepsy <- epilepsy %>% mutate( per = log(2), base_sc = base/5 ) %>%
  rename( trt = treatment)
fm <- seizure.rate ~ base + age + trt + period + trt*period
epilepsy_glm <- glm(fm, data = epilepsy, family = "poisson", offset = per)
epilepsy_gee1 <- geeglm(fm, data = epilepsy, family = "poisson",
                        id = subject, corstr = "independence", offset = per)
epilepsy_gee2 <- geeglm(fm, data = epilepsy, family = "poisson",
                        id = subject, corstr = "exchangeable", offset = per)
summary(epilepsy_glm)
```

```
##
## Call:
## glm(formula = fm, family = "poisson", data = epilepsy, offset = per)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -4.484  -1.487  -0.454   0.493  12.210
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -0.132478   0.135633   -0.98   0.3287
## base                      0.022652   0.000509   44.48  < 2e-16 ***
## age                       0.022740   0.004024    5.65  1.6e-08 ***
## trtProgabide             -0.155468   0.047947   -3.24   0.0012 **
## period.L                 -0.095016   0.064454   -1.47   0.1404
## period.Q                  0.011724   0.064569    0.18   0.8559
## period.C                 -0.075345   0.064684   -1.16   0.2441
## trtProgabide:period.L -0.077825   0.091648   -0.85   0.3958
## trtProgabide:period.Q -0.098181   0.090889   -1.08   0.2800
## trtProgabide:period.C  0.043888   0.090123    0.49   0.6263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2521.75  on 235  degrees of freedom
## Residual deviance:  946.05  on 226  degrees of freedom
## AIC: 1732
##
## Number of Fisher Scoring iterations: 5
```

```
# summary(epilepsy_gee1) Results very similar to the exchangeable model.
summary(epilepsy_gee2)
```

```
##
## Call:
## geeglm(formula = fm, family = "poisson", data = epilepsy, offset = per,
##     id = subject, corstr = "exchangeable")
##
##  Coefficients:
##                         Estimate  Std.err   Wald Pr(>|W|)
## (Intercept)             -0.16165  0.37354   0.19    0.665
## base                     0.02272  0.00125 331.73   <2e-16 ***
## age                      0.02360  0.01180   4.00    0.045 *
## trtProgabide            -0.15317  0.17108   0.80    0.371
## period.L                -0.09502  0.12706   0.56    0.455
## period.Q                 0.01172  0.14031   0.01    0.933
## period.C                -0.07535  0.15180   0.25    0.620
## trtProgabide:period.L -0.07782  0.14678   0.28    0.596
## trtProgabide:period.Q -0.09818  0.17991   0.30    0.585
## trtProgabide:period.C  0.04389  0.17609   0.06    0.803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)     4.88    1.46
##   Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha    0.413  0.0674
## Number of clusters:   59  Maximum cluster size: 4
```

```
anova(epilepsy_gee2)
```

	Df	X2	P(>|Chi|)
base	1	581.387	0.000
age	1	4.642	0.031
trt	1	0.796	0.372
period	3	6.701	0.082
trt:period	3	1.501	0.682

What if we change to an unordered factor?

```
epilepsy <- epilepsy %>% mutate( per_un_ord = factor( period, ordered = FALSE))
str(epilepsy)
```

```
## 'data.frame':    236 obs. of  9 variables:
##  $ trt         : Factor w/ 2 levels "placebo","Progabide": 1 1 1 1 1 1 1 1 1 1 ...
##  $ base        : int  11 11 11 11 11 11 11 11 6 6 ...
##  $ age         : int  31 31 31 31 30 30 30 30 25 25 ...
##  $ seizure.rate: int  5 3 3 3 3 5 3 3 2 4 ...
##  $ period      : Ord.factor w/ 4 levels "1"<"2"<"3"<"4": 1 2 3 4 1 2 3 4 1 2 ...
##  $ subject     : Factor w/ 59 levels "1","2","3","4",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ per         : num  0.693 0.693 0.693 0.693 0.693 ...
##  $ base_sc     : num  2.2 2.2 2.2 2.2 2.2 2.2 2.2 2.2 1.2 1.2 ...
##  $ per_un_ord  : Factor w/ 4 levels "1","2","3","4": 1 2 3 4 1 2 3 4 1 2 ...
```

```
fm <- seizure.rate ~ base + age + trt + per_un_ord + trt*per_un_ord
epilepsy_gee3 <- geeglm(fm, data = epilepsy, family = "poisson",
                     id = subject, corstr = "exchangeable", offset = per)
coef( summary(epilepsy_gee3))
```

	Estimate	Std.err	Wald	Pr(>|W|)
(Intercept)	-0.075	0.340	0.049	0.825
base	0.023	0.001	331.730	0.000
age	0.024	0.012	4.003	0.045
trtProgabide	-0.160	0.182	0.769	0.380
per_un_ord2	-0.122	0.128	0.905	0.341
per_un_ord3	-0.063	0.271	0.054	0.816
per_un_ord4	-0.161	0.154	1.092	0.296
trtProgabide:per_un_ord2	0.103	0.223	0.212	0.645
trtProgabide:per_un_ord3	0.009	0.314	0.001	0.977
trtProgabide:per_un_ord4	-0.085	0.191	0.197	0.657

```
anova( epilepsy_gee3)
```

	Df	X2	P(>|Chi|)
base	1	581.387	0.000
age	1	4.642	0.031
trt	1	0.796	0.372
per_un_ord	3	6.701	0.082
trt:per_un_ord	3	1.501	0.682