Repeated Categorical Outcome Analysis

Note updated April 28, 2019

Wan Nor Arifin
Unit of Biostatistics and Research Methodology,
Universiti Sains Malaysia.

Email: wnarifin@usm.my
Website: wnarifin.github.io

Contents

1 Introduction

Dealing with repeated categorical outcomes, there are several options of analysis. The choice of analysis depends on the data:

- Dependent/outcome variable (DV): binary, multinomial, ordinal.
- Repetition/measure: 2 times, $>$ 2 times.

The options are:

1. DV: Binary, Measure: 2 times. Analysis: McNemar's Test
2. DV: Binary, Measure: $>$ 2 times. Analysis: Cochran's Q Test
3. DV: Multinomial, Measure: 2 times. Analysis: Stuart-Maxwell's Test.
4. DV: Binary/Multinominal, Measure: $\geq$ 2 times. Analysis: Generalized Estimating Equations (GEE).

Basically GEE can do 1-3, but we go through each basic analysis.

2 McNemar's Test

For DV: Binary, Measure: 2 times.

We use PM rating from Agresti (2002), pg409, n = 1600.

```
Data:
            second
first        approve disapprove
  approve        794        150
  disapprove      86        570
```

Prepare the data,

```
first = factor(rep(c(1,2), c(794+150, 86+570)),
               labels = c("approve", "disapprove"))
second = factor(rep(c(1,2,1,2), c(794, 150, 86, 570)),
                labels = c("approve", "disapprove"))
tab = table(first, second); tab
```

```
##             second
## first         approve disapprove
##   approve         794        150
##   disapprove       86        570
```

Then, McNemar's test,

```
mcnemar.test(tab)
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  tab
## McNemar's chi-squared = 16.818, df = 1, p-value = 4.115e-05
```

What's your interpretation?

3 Cochran's Q Test

For DV: Binary, Measure: > 2 times.

Dataset `lect.csv` consists of measuments of understanding $= \{1, 0\}$ for $n = 10$ students taught by 3 different lecturers. Read the data,

```
lect = read.csv("lect.csv")
```

Prepare `student` and `lecturer` as factors,

```
lect$student = as.factor(lect$student)
lect$lecturer = as.factor(lect$lecturer)
str(lect)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ student      : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ lecturer     : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
##  $ understanding: int  1 0 0 0 0 0 0 0 0 0 ...
```

```
lect
```

```
##    student lecturer understanding
## 1        1        1             1
## 2        1        2             0
## 3        1        3             0
## 4        2        1             0
## 5        2        2             0
## 6        2        3             0
## 7        3        1             0
## 8        3        2             0
## 9        3        3             0
## 10       4        1             0
## 11       4        2             0
## 12       4        3             1
## 13       5        1             1
## 14       5        2             1
## 15       5        3             1
## 16       6        1             0
## 17       6        2             0
## 18       6        3             1
## 19       7        1             0
## 20       7        2             0
## 21       7        3             1
## 22       8        1             1
## 23       8        2             1
## 24       8        3             0
## 25       9        1             0
## 26       9        2             0
## 27       9        3             0
## 28      10        1             1
## 29      10        2             1
## 30      10        3             0
```

Load `RVAideMemoire` package.

```r
library(RVAideMemoire)
```

```
## *** Package RVAideMemoire v 0.9-73 ***
```

Then Cochran's Q test,

```r
cochran.qtest(understanding ~ lecturer | student, data = lect)
```

```
##
##  Cochran's Q test
##
## data:  understanding by lecturer, block = student
## Q = 0.3333, df = 2, p-value = 0.8465
## alternative hypothesis: true difference in probabilities is not equal to 0
## sample estimates:
## proba in group 1 proba in group 2 proba in group 3
##              0.4              0.3              0.4
```

Can also be done using `coin` package,

```r
# set outcome as factor for mh_test() function
lect$understanding = as.factor(lect$understanding)
str(lect)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ student      : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ lecturer     : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
##  $ understanding: Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
```

```r
library(coin)
```

```
## Loading required package: survival
```

```r
mh_test(understanding ~ lecturer | student, data = lect)
```

```
##
##  Asymptotic Marginal Homogeneity Test
##
## data:  understanding by lecturer (1, 2, 3)
##   stratified by student
## chi-squared = 0.33333, df = 2, p-value = 0.8465
```

What's your interpretation?

4 Stuart-Maxwell's Test

For DV: Multinomial, Measure: 2 times. When DV = Binary, this test is identical to McNemar's test.

Now we use the following data (hypothetical data), on understanding level of my statistics lectures, n = 200,

```
Data:
              after.lecture
before.lecture confused so-so understand
    confused         12     8         80
    so-so            10    10         20
    understand        5     8         47
```

Prepare the data,

```r
before.lecture = factor(rep(c(1,2,3), c(12+8+80, 10+10+20, 5+8+47)),
                        labels = c("confused", "so-so", "understand"))
after.lecture = factor(rep(c(1,2,3,1,2,3,1,2,3), c(12,8,80, 10,10,20, 5,8,47)),
                       labels = c("confused", "so-so", "understand"))
tabq = table(before.lecture, after.lecture); tabq
```

```
##               after.lecture
## before.lecture confused so-so understand
##     confused         12     8         80
##     so-so            10    10         20
##     understand        5     8         47
```

For the analysis, load `coin` package.

```r
library(coin)
```

Now, perform marginal homogeneity test whenever the categories are treated as **nominal**,

```r
mh_test(tabq)  # nominal
```

```
##
##  Asymptotic Marginal Homogeneity Test
##
## data:  response by
##   conditions (before.lecture, after.lecture)
##   stratified by block
## chi-squared = 68.444, df = 2, p-value = 1.332e-15
```

and **ordinal**, i.e. Stuart-Birch's test (Hothorn, Winell, Hornik, van de Wiel, & Zeileis, 2019),

```
mh_test(tabq, scores = list(response = 1:3))  # ordinal
```

```
##
##  Asymptotic Marginal Homogeneity Test for Ordered Data
##
## data:  response (ordered) by
##   conditions (before.lecture, after.lecture)
##   stratified by block
## Z = -8.1438, p-value = 3.831e-16
## alternative hypothesis: two.sided
```

What's your interpretation for these two tests?

5 Generalized Estimating Equations (GEE)

For DV: Binary/Multinominal, Measure: ≥ 2 times.

In short, about GEE (Agresti, 2002; Hardin & Hilbe, 2002):

- It is a quasi-likelihood method.
- An extension of GLM to include correlation structure – i.e. specify dependent subjects.
- GLM assumes specific distributions for the outcomes, GEE relaxes this assumption.
- Computational simplicity for categorical data as compared to ML.
- Performs well even with misspecified correlation structure.
- However, it does not have a likelihood function, thus cannot use likelihood-based methods for testing model fit, model comparison and inference.

Correlation structures available for GEE (details in Hardin & Hilbe (2002)):

1. Exchangeable*.
2. Autoregressive.
3. Stationary.
4. Nonstationary.
5. Unstructured.
6. Fixed.
7. Free.

Steps in GEE:

1. Specify the model.
2. Specify correlation structure. Usually just set to "exchangeable" – flexible.
3. Fit the model.
4. Hypothesis testing – Wald's test (LR test not applicable)
5. Model comparison by quasi information criterion (QIC).

For practical session, we use depression data from Agresti (2002), pg459, Table 11.2.

The data orginally comes from a longitudinal study by Koch et al. (1977). The study compared a *new drug* vs *standard* on depressive patients. The patients were also grouped by their initial diagnosis as *mild* or *severe* depression. The comparison were done at *week 1, 2*, and *4* (scored as 0, 1, 2). At each week, their response were classified as *normal* = 1 or *abnormal* = 0.

Now, prepare the data,

```
ID = rep(1:340, each = 3)
Diagnosis = factor( rep(c(1,2), c(150*3, 190*3)), labels = c("Mild", "Severe") )
```

```r
Treatment = factor( rep(c(1,2,1,2), c(80*3, 70*3, 100*3, 90*3)),
                    labels = c("Standard", "New Drug") )
Time = rep(0:2, times = 340)
Response = unlist( c(
  rep( list( c(1,1,1), c(1,1,0), c(1,0,1), c(1,0,0),
             c(0,1,1), c(0,1,0), c(0,0,1), c(0,0,0) ), c(16,13,9,3, 14,4,15,6) ),
  rep( list( c(1,1,1), c(1,1,0), c(1,0,1), c(1,0,0),
             c(0,1,1), c(0,1,0), c(0,0,1), c(0,0,0) ), c(31,0,6,0, 22,2,9,0) ),
  rep( list( c(1,1,1), c(1,1,0), c(1,0,1), c(1,0,0),
             c(0,1,1), c(0,1,0), c(0,0,1), c(0,0,0) ), c(2,2,8,9, 9,15,27,28) ),
  rep( list( c(1,1,1), c(1,1,0), c(1,0,1), c(1,0,0),
             c(0,1,1), c(0,1,0), c(0,0,1), c(0,0,0) ), c(7,2,5,2, 31,5,32,6) )
) )
# 1 = normal, 0 = abnormal
depress = data.frame(ID, Diagnosis, Treatment, Time, Response)
head(depress); tail(depress)
```

```
##   ID Diagnosis Treatment Time Response
## 1  1      Mild  Standard    0        1
## 2  1      Mild  Standard    1        1
## 3  1      Mild  Standard    2        1
## 4  2      Mild  Standard    0        1
## 5  2      Mild  Standard    1        1
## 6  2      Mild  Standard    2        1

##       ID Diagnosis Treatment Time Response
## 1015 339    Severe  New Drug    0        0
## 1016 339    Severe  New Drug    1        0
## 1017 339    Severe  New Drug    2        0
## 1018 340    Severe  New Drug    0        0
## 1019 340    Severe  New Drug    1        0
## 1020 340    Severe  New Drug    2        0
```

```r
str(depress)
```

```
## 'data.frame':    1020 obs. of  5 variables:
##  $ ID       : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ Diagnosis: Factor w/ 2 levels "Mild","Severe": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Treatment: Factor w/ 2 levels "Standard","New Drug": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Time     : int  0 1 2 0 1 2 0 1 2 0 ...
##  $ Response : num  1 1 1 1 1 1 1 1 1 1 ...
```

Load these packages,

```r
library(geepack)  # geeglm
library(MuMIn)  # QIC
library(doBy)  # esticon(), confint doesn't work for GEE
```

As usual for any modeling approach, we fit several models from univariable to multivariable ones. The only different to GLM is that we include here the ID identifier to indicate clusters of subjects. We are going to fit GEE model with *exchangeable* correlation structure.

Intercept only model,

```r
gee.model0 = geeglm(Response ~ 1, family = binomial, data = depress,
                    id = ID, corstr = "exchangeable")
summary(gee.model0)
```

```
##
## Call:
## geeglm(formula = Response ~ 1, family = binomial, data = depress,
##     id = ID, corstr = "exchangeable")
##
##  Coefficients:
##             Estimate Std.err  Wald Pr(>|W|)
## (Intercept)  0.09026 0.06522 1.915    0.166
##
## Estimated Scale Parameters:
##             Estimate  Std.err
## (Intercept)        1 0.002947
##
## Correlation: Structure = exchangeable  Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha  0.04119 0.03237
## Number of clusters:   340   Maximum cluster size: 3
```

```r
QIC(gee.model0)
```

```
##  QIC
## 1414
```

One variable models,

```r
gee.model1 = geeglm(Response ~ Diagnosis, family = binomial, data = depress,
                    id = ID, corstr = "exchangeable")
summary(gee.model1)
```

```
##
## Call:
## geeglm(formula = Response ~ Diagnosis, family = binomial, data = depress,
##     id = ID, corstr = "exchangeable")
##
##  Coefficients:
##                 Estimate Std.err Wald Pr(>|W|)
## (Intercept)        0.713    0.103 48.3  3.7e-12 ***
## DiagnosisSevere   -1.097    0.130 71.6  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)        1  0.0195
##
## Correlation: Structure = exchangeable  Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha  -0.0288  0.0303
## Number of clusters:   340   Maximum cluster size: 3
```

```r
QIC(gee.model1)
```

```
##  QIC
## 1343
```

```
gee.model1a = geeglm(Response ~ Treatment, family = binomial, data = depress,
                     id = ID, corstr = "exchangeable")
summary(gee.model1a)
```

```
##
## Call:
## geeglm(formula = Response ~ Treatment, family = binomial, data = depress,
##     id = ID, corstr = "exchangeable")
##
##  Coefficients:
##                    Estimate Std.err  Wald Pr(>|W|)
## (Intercept)         -0.2457  0.0899  7.47   0.0063 **
## TreatmentNew Drug   0.7211  0.1285 31.51    2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)        1  0.0122
##
## Correlation: Structure = exchangeable  Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha  0.00933  0.0309
## Number of clusters:   340   Maximum cluster size: 3
```

```
QIC(gee.model1a)
```

```
##  QIC
## 1384
```

```
gee.model1b = geeglm(Response ~ Time, family = binomial, data = depress,
                     id = ID, corstr = "exchangeable")
summary(gee.model1b)
```

```
##
## Call:
## geeglm(formula = Response ~ Time, family = binomial, data = depress,
##     id = ID, corstr = "exchangeable")
##
##  Coefficients:
##             Estimate Std.err Wald Pr(>|W|)
## (Intercept)  -0.6944  0.1095 40.2  2.2e-10 ***
## Time          0.7937  0.0817 94.3  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)        1  0.0241
##
## Correlation: Structure = exchangeable  Link = identity
##
```

```
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha   0.0957  0.0337
## Number of clusters:   340   Maximum cluster size: 3
```

```
QIC(gee.model1b)
```

```
##  QIC
## 1317
```

Then three variable model,

```
gee.model3 = geeglm(Response ~ Diagnosis + Treatment + Time, family = binomial,
                    data = depress, id = ID, corstr = "exchangeable")
summary(gee.model3)
```

```
##
## Call:
## geeglm(formula = Response ~ Diagnosis + Treatment + Time, family = binomial,
##     data = depress, id = ID, corstr = "exchangeable")
##
##  Coefficients:
##                    Estimate Std.err  Wald Pr(>|W|)
## (Intercept)         -0.4815  0.1568  9.43   0.0021 **
## DiagnosisSevere     -1.2862  0.1442 79.52  < 2e-16 ***
## TreatmentNew Drug    0.8848  0.1392 40.41  2.1e-10 ***
## Time                 0.9013  0.0932 93.62  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)    0.994  0.0564
##
## Correlation: Structure = exchangeable  Link = identity
##
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha  -0.0221  0.0307
## Number of clusters:   340   Maximum cluster size: 3
```

```
QIC(gee.model3)
```

```
##  QIC
## 1201
```

```
gee.est3 = esticon(gee.model3, diag(4)); gee.est3
```

```
##          beta0  Estimate Std.Error  X2.value        DF Pr(>|X^2|)     Lower Upper
## [1,]  0.00e+00 -4.81e-01  1.57e-01  9.43e+00  1.00e+00   2.13e-03 -7.89e-01 -0.17
## [2,]  0.00e+00 -1.29e+00  1.44e-01  7.95e+01  1.00e+00   0.00e+00 -1.57e+00 -1.00
## [3,]  0.00e+00  8.85e-01  1.39e-01  4.04e+01  1.00e+00   2.06e-10  6.12e-01  1.16
## [4,]  0.00e+00  9.01e-01  9.32e-02  9.36e+01  1.00e+00   0.00e+00  7.19e-01  1.08
```

Following Agresti (2002), we fit Treatment*Time interaction,

```
gee.model4 = geeglm(Response ~ Diagnosis + Treatment*Time, family = binomial,
                    data = depress, id = ID, corstr = "exchangeable")
```

```
summary(gee.model4)
```

```
## 
## Call:
## geeglm(formula = Response ~ Diagnosis + Treatment * Time, family = binomial, 
##     data = depress, id = ID, corstr = "exchangeable")
## 
##  Coefficients:
##                         Estimate Std.err  Wald Pr(>|W|)    
## (Intercept)              -0.0281  0.1742  0.03     0.87    
## DiagnosisSevere          -1.3139  0.1460 81.03  < 2e-16 ***
## TreatmentNew Drug        -0.0593  0.2286  0.07     0.80    
## Time                      0.4825  0.1199 16.18  5.8e-05 ***
## TreatmentNew Drug:Time    1.0172  0.1877 29.37  6.0e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)    0.981  0.0653
## 
## Correlation: Structure = exchangeable  Link = identity 
## 
## Estimated Correlation Parameters:
##       Estimate Std.err
## alpha -0.00343  0.0303
## Number of clusters:   340   Maximum cluster size: 3
```

```
QIC(gee.model4)
```

```
##  QIC 
## 1172 
```

```
gee.est4 = esticon(gee.model4, diag(5)); gee.est4
```

```
##          beta0  Estimate Std.Error  X2.value        DF Pr(>|X^2|)     Lower Upper
## [1,]  0.00e+00 -2.81e-02  1.74e-01  2.60e-02  1.00e+00   8.72e-01 -3.69e-01  0.31
## [2,]  0.00e+00 -1.31e+00  1.46e-01  8.10e+01  1.00e+00   0.00e+00 -1.60e+00 -1.03
## [3,]  0.00e+00 -5.93e-02  2.29e-01  6.72e-02  1.00e+00   7.95e-01 -5.07e-01  0.39
## [4,]  0.00e+00  4.82e-01  1.20e-01  1.62e+01  1.00e+00   5.76e-05  2.47e-01  0.72
## [5,]  0.00e+00  1.02e+00  1.88e-01  2.94e+01  1.00e+00   5.99e-08  6.49e-01  1.39
```

```
with(gee.est4, cbind(OR = exp(Estimate), LL = exp(Lower), UL = exp(Upper)))
```

```
##         OR    LL    UL
## [1,] 0.972 0.691 1.368
## [2,] 0.269 0.202 0.358
## [3,] 0.942 0.602 1.475
## [4,] 1.620 1.281 2.049
## [5,] 2.765 1.914 3.995
```

Which model is the best? Why? What's your interpretation?

References

Agresti, A. (2002). *Categorical data analysis (2nd ed.* Hoboken, New Jersey: John Wiley & Sons.

Hardin, J., & Hilbe, J. (2002). *Generalized estimating equations.* CRC Press.

Hothorn, T., Winell, H., Hornik, K., van de Wiel, M. A., & Zeileis, A. (2019). *Coin: Conditional inference procedures in a permutation test framework.* Retrieved from https://CRAN.R-project.org/package=coin

Uebersax, J. (2006). *McNemar tests of marginal homogeneity.* Retrieved from http://john-uebersax.com/stat/mcnemar.htm