

Homework 4

Adeline Shin

11/24/2020

Uploading the Data

```
# Upload the toenail df
toenail_df = read.delim("./toenail.txt", header = TRUE, sep = " ", col.names = c("Subject ID", "Response"))

# Clean up toenail_df
toenail_df =
  toenail_df[, 1:5] %>%
  janitor::clean_names()

toenail_df = as.data.table(toenail_df)
toenail_df$treatment = as.factor(toenail_df$treatment)
```

Question 1

Part 1

```
# Add response at lag 1
toenail_df[, response_1 := shift(response, n = 1, type = "lag", fill = NA), by = "subject_id"]

# Transition Probabilities
tab1 = table(toenail_df$response, toenail_df$response_1)
tab1

##
##      0      1
## 0 1203  112
## 1   28  271

round(prop.table(tab1, margin = 1), 2)

##
##      0      1
## 0 0.91 0.09
## 1 0.09 0.91

# Association b/w treatment and response
tab2 <- table(toenail_df$response, toenail_df$treatment)
round(prop.table(tab2, margin = 1), 2)

##
##      0      1
## 0 0.48 0.52
## 1 0.52 0.48
```

```

# Stratified by previous response
temp <- split(toenail_df, toenail_df$response_1)
tab3 <- lapply(temp, function(z){table(z$response, z$treatment)})
lapply(tab3, function(z){round(prop.table(z,margin = 1),2)})

## $`0`
##
##      0      1
## 0 0.48 0.52
## 1 0.68 0.32
##
## $`1`
##
##      0      1
## 0 0.50 0.50
## 1 0.52 0.48

# Model
model_lag_1 <- gee(response ~ treatment * (response_1 + month), corstr = "independence",family = binomial)

## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate

##      (Intercept)      treatment1      response_1
##      -2.91754387      -0.58731398      4.20408170
##      month treatment1:response_1 treatment1:month
##      -0.09042707      0.69205101      -0.05921630

round(summary(model_lag_1)$coeff,2)

##      Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)      -2.92      0.32      -9.05      0.30      -9.58
## treatment1      -0.59      0.53      -1.10      0.48      -1.21
## response_1       4.20      0.31      13.40      0.33      12.57
## month          -0.09      0.04      -2.24      0.04      -2.31
## treatment1:response_1  0.69      0.52      1.33      0.49      1.40
## treatment1:month     -0.06      0.07      -0.91      0.07      -0.81

The interaction term is insignificant according to the z-score, so we will use a model without the interaction term.

model_lag_1b <- gee(response ~ treatment + (response_1 + month), corstr = "independence",family = binomial)

## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
## running glm to get initial regression estimate

## (Intercept) treatment1 response_1      month
## -3.0094361 -0.3090397  4.4906918 -0.1152287

round(summary(model_lag_1b)$coeff,2)

##      Estimate Naive S.E. Naive z Robust S.E. Robust z
## (Intercept)      -3.01      0.27     -11.24      0.25     -12.01
## treatment1      -0.31      0.21      -1.46      0.18      -1.74
## response_1       4.49      0.24      18.64      0.25      18.26
## month          -0.12      0.03      -3.73      0.03      -3.41

```

Part 2

```
toenail_df2 = toenail_df[, response_1 := shift(response, n = 2, type = "lag", fill = NA), by = "subject"]
```

```
# Transition Probabilities
```

```
tab1 = table(toenail_df$response, toenail_df$response_1)
```

```
tab1
```

```
##
```

```
##      0    1
```

```
## 0 934 189
```

```
## 1  33 169
```

```
round(prop.table(tab1, margin = 1), 2)
```

```
##
```

```
##      0    1
```

```
## 0 0.83 0.17
```

```
## 1 0.16 0.84
```

```
# Association b/w treatment and response
```

```
tab2 <- table(toenail_df$response, toenail_df$treatment)
```

```
round(prop.table(tab2, margin = 1), 2)
```

```
##
```

```
##      0    1
```

```
## 0 0.48 0.52
```

```
## 1 0.52 0.48
```

```
# Stratified by previous response
```

```
temp <- split(toenail_df, toenail_df$response_1)
```

```
tab3 <- lapply(temp, function(z){table(z$response, z$treatment)})
```

```
lapply(tab3, function(z){round(prop.table(z, margin = 1), 2)})
```

```
## $`0`
```

```
##
```

```
##      0    1
```

```
## 0 0.48 0.52
```

```
## 1 0.61 0.39
```

```
##
```

```
## $`1`
```

```
##
```

```
##      0    1
```

```
## 0 0.49 0.51
```

```
## 1 0.54 0.46
```

```
# Model
```

```
model_lag_1b <- gee(response ~ treatment + (response_1 + month), corstr = "independence", family = binomial)
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
```

```
## (Intercept) treatment1 response_1      month
```

```
## -2.3094697 -0.3286715  3.1178039 -0.1465387
```

```
round(summary(model_lag_1b)$coeff, 2)
```

```
##              Estimate Naive S.E. Naive z Robust S.E. Robust z
```

## (Intercept)	-2.31	0.27	-8.70	0.30	-7.68
## treatment1	-0.33	0.20	-1.63	0.22	-1.47
## response_1	3.12	0.23	13.82	0.27	11.72
## month	-0.15	0.03	-4.82	0.04	-3.71

Part 3

The following interpretations can be made for the coefficients in the first-order transition model:

- Intercept: -3.01 is the log odds of having moderate or severe onycholysis for those who had no treatment and did not have moderate or severe onycholysis in the previous month
- treatment_1: -0.31 is the log odds ratio of respiratory infection comparing those with and without moderate or severe onycholysis who had an identical treatment status in the previous month.
- response_1: 4.49 is the log odds ratio of moderate or severe onycholysis comparing those with and without treatment in the previous month who currently have an identical status for moderate or severe onycholysis.
- month: -0.12 is the log odds ratio of moderate or severe onycholysis with an increase in month for those who have an identical status for moderate or severe onycholysis.

Part 4

The interpretations from the transition model differ from the ones in HW 2 and HW 3 because they compare the current status to the status at the previous month, while the previous models (GEE and mixed effects) do not account for the direct previous month

Question 2

Setup

```
toenail <- fread("toenail.txt")
colnames(toenail) <- c("id", "response", "treatment", "month", "visit")
toenail2 <- tidyr::complete(toenail, id, visit) %>%
tidyr::fill(treatment)
toenail2 <- as.data.table(toenail2)
```

Part 1

```
# Complete case analysis
count = toenail2[, j = list(n = sum(!is.na(response))), by = "id"]
table(count$n)
```

```
##
##  1  2  3  4  5  6  7
##  5  3  7  6 10 39 224
```

```
count = count[n==7]
toenail3 = toenail2[id %in% count$id]
table(toenail3$response, useNA = "always")
```

```
##
##      0      1 <NA>
## 1266  302      0
```

```
table(toenail3$visit, toenail3$response, useNA = "always")
```

```
##
##           0      1 <NA>
##      1      144  80      0
##      2      152  72      0
##      3      161  63      0
##      4      180  44      0
##      5      207  17      0
##      6      211  13      0
##      7      211  13      0
##    <NA>      0      0      0
```

```
gee1 = geeglm(response ~ treatment + (visit + I(visit^2)) , id = id, data = toenail3, family = binomial)
summary(gee1)
```

```
##
## Call:
## geeglm(formula = response ~ treatment + (visit + I(visit^2)),
##       family = binomial(link = "logit"), data = toenail3, id = id,
##       corstr = "unstructured")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept) -0.55054   0.22593  5.938   0.0148 *
## treatment    -0.08481   0.25414  0.111   0.7386
## visit        -0.11827   0.11014  1.153   0.2829
## I(visit^2)   -0.03384   0.01808  3.506   0.0612 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)    1.041   0.3283
## Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha.1:2      0.8960  0.28370
## alpha.1:3      0.7745  0.25781
## alpha.1:4      0.5556  0.20552
## alpha.1:5      0.2432  0.12187
## alpha.1:6      0.1470  0.09672
## alpha.1:7      0.1529  0.10101
## alpha.2:3      0.8882  0.29160
## alpha.2:4      0.6492  0.23421
## alpha.2:5      0.2984  0.14228
## alpha.2:6      0.2046  0.11634
## alpha.2:7      0.1828  0.11207
```

```
## alpha.3:4 0.8334 0.28901
## alpha.3:5 0.3074 0.14938
## alpha.3:6 0.2004 0.12192
## alpha.3:7 0.2248 0.12759
## alpha.4:5 0.4036 0.17758
## alpha.4:6 0.2764 0.14615
## alpha.4:7 0.3070 0.15330
## alpha.5:6 0.5167 0.21708
## alpha.5:7 0.5128 0.21372
## alpha.6:7 0.7330 0.26623
## Number of clusters: 224 Maximum cluster size: 7
```

Part 2

```
# Available case analysis
table(toenail2$response, useNA = "always")
```

```
##
##      0      1 <NA>
## 1500  408  150
```

```
table(toenail2$visit, toenail2$response, useNA = "always")
```

```
##
##           0      1 <NA>
## 1      185  109      0
## 2      191   97      6
## 3      199   84     11
## 4      214   58     22
## 5      241   22     31
## 6      226   18     50
## 7      244   20     30
## <NA>      0      0      0
```

```
gee2 = geeglm(response ~ treatment + (visit + I(visit^2)) , id = id, data = toenail2, family = binomial)
summary(gee2)
```

```
##
## Call:
## geeglm(formula = response ~ treatment + (visit + I(visit^2)),
##        family = binomial(link = "logit"), data = toenail2, id = id,
##        corstr = "unstructured")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  -0.3773   0.1862  4.11   0.043 *
## treatment    -0.1495   0.2142  0.49   0.485
## visit        -0.1235   0.0913  1.83   0.176
## I(visit^2)   -0.0302   0.0149  4.10   0.043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
```

```
##
##              Estimate Std.err
## (Intercept)    1.01   0.208
##   Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha.1:2      0.895  0.1880
## alpha.1:3      0.716  0.1611
## alpha.1:4      0.545  0.1405
## alpha.1:5      0.263  0.0941
## alpha.1:6      0.153  0.0789
## alpha.1:7      0.131  0.0804
## alpha.2:3      0.832  0.1848
## alpha.2:4      0.648  0.1629
## alpha.2:5      0.285  0.1016
## alpha.2:6      0.240  0.0947
## alpha.2:7      0.154  0.0870
## alpha.3:4      0.850  0.2027
## alpha.3:5      0.296  0.1073
## alpha.3:6      0.213  0.0961
## alpha.3:7      0.187  0.0968
## alpha.4:5      0.397  0.1256
## alpha.4:6      0.294  0.1120
## alpha.4:7      0.258  0.1128
## alpha.5:6      0.493  0.1502
## alpha.5:7      0.438  0.1500
## alpha.6:7      0.607  0.1842
## Number of clusters: 294 Maximum cluster size: 7
```

Part 3

```
# LOCF
toenail4 = lapply(unique(toenail2$id), function(z){tidyr::fill(toenail2[id == z], treatment)})
toenail4 = rbindlist(toenail4)
table(toenail4$visit, toenail4$response, useNA = "always")

##
##           0    1 <NA>
## 1      185 109    0
## 2      191  97    6
## 3      199  84   11
## 4      214  58   22
## 5      241  22   31
## 6      226  18   50
## 7      244  20   30
## <NA>    0    0    0

gee3 = geeglm(response ~ treatment + (visit + I(visit^2)), id = id, data = toenail4, family = binomial())
summary(gee3)

##
## Call:
## geeglm(formula = response ~ treatment + (visit + I(visit^2)),
```

```

##      family = binomial(link = "logit"), data = toenail4, id = id,
##      corstr = "unstructured")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  -0.3773  0.1862 4.11   0.043 *
## treatment    -0.1495  0.2142 0.49   0.485
## visit        -0.1235  0.0913 1.83   0.176
## I(visit^2)    -0.0302  0.0149 4.10   0.043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)      1.01   0.208
## Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha.1:2      0.895  0.1880
## alpha.1:3      0.716  0.1611
## alpha.1:4      0.545  0.1405
## alpha.1:5      0.263  0.0941
## alpha.1:6      0.153  0.0789
## alpha.1:7      0.131  0.0804
## alpha.2:3      0.832  0.1848
## alpha.2:4      0.648  0.1629
## alpha.2:5      0.285  0.1016
## alpha.2:6      0.240  0.0947
## alpha.2:7      0.154  0.0870
## alpha.3:4      0.850  0.2027
## alpha.3:5      0.296  0.1073
## alpha.3:6      0.213  0.0961
## alpha.3:7      0.187  0.0968
## alpha.4:5      0.397  0.1256
## alpha.4:6      0.294  0.1120
## alpha.4:7      0.258  0.1128
## alpha.5:6      0.493  0.1502
## alpha.5:7      0.438  0.1500
## alpha.6:7      0.607  0.1842
## Number of clusters: 294 Maximum cluster size: 7

```

Part 4

```

# MI
toenail5 = toenail2
pred = make.predictorMatrix(toenail5)
pred

```

```

##              id visit response treatment month
## id          0     1         1           1     1

```



```
## visit      1      0      1      1      1
## response   1      1      0      1      1
## treatment  1      1      1      0      1
## month      1      1      1      1      0
```

```
pred["response", "id"] = -2
pred
```

```
##          id visit response treatment month
## id        0      1      1      1      1
## visit     1      0      1      1      1
## response  -2      1      0      1      1
## treatment 1      1      1      0      1
## month     1      1      1      1      0
```

```
pred = pred["response",,drop = FALSE]
pred
```

```
##          id visit response treatment month
## response -2      1      0      1      1
```

```
toenail5$id <- as.integer(toenail5$id)
imp = mice(toenail5, method = "2l.bin", pred = pred, seed = 1, maxit = 1, m = 5, print = FALSE, blocks =
table(mice::complete(imp)$response, useNA = "always")
```

```
##
##      0      1 <NA>
## 1500  408  150
```

```
# GEE
implist = mids2mitml.list(imp)
gee4 = with(implist, geeglm(response ~ treatment + (visit + I(visit^2)), id = id, family = binomial, co
testEstimates(gee4)
```

```
##
## Call:
##
## testEstimates(model = gee4)
##
## Final parameter estimates and inferences obtained from 5 imputed data sets.
##
##          Estimate Std.Error   t.value      df    P(>|t|)      RIV      FMI
## (Intercept)  -0.377    0.186   -2.026    Inf    0.043    0.000    0.000
## treatment    -0.150    0.214   -0.698    Inf    0.485    0.000    0.000
## visit        -0.124    0.091   -1.353    Inf    0.176    0.000    0.000
## I(visit^2)   -0.030    0.015   -2.025    Inf    0.043    0.000    0.000
##
## Unadjusted hypothesis test as appropriate in larger samples.
```

Part 5

```
# ME
lme1 = mice::complete(imp, "all") %>%
  purrr::map(lme4::glmer,
    formula = response ~ treatment + (visit + I(visit^2)) + (1 | id),
    family = binomial) %>%
```

```
pool()
```

```
summary(lme1)
```

##	term	estimate	std.error	statistic	df	p.value
## 1	(Intercept)	-1.5468	0.8562	-1.81	1901	0.0710
## 2	treatment	-0.7008	0.6830	-1.03	1901	0.3050
## 3	visit	-0.5330	0.2282	-2.34	1901	0.0196
## 4	I(visit^2)	-0.0521	0.0305	-1.71	1901	0.0875