Week Fourteen

Last Week

- Class Recap
- Introduction to Correlated Data

This Week: More Correlated Data

- Tuesday: Activity
- Thursday: Lab

Next Week: Fall Break

Next-Next Week: Classification Algorithms + Final Exam Review

- Tuesday:
  - Video Lecture
  - In Class Lab
- Thursday:
  - Final Exam Review
  - Take Home Final Assigned

McNemar’s Test

Recall McNemar’s Test where we have repeated measurements on a sampling unit.

We previously framed this in the context of asking a respondent two related policy questions to evaluate whether one would be more amenable.

Another scenario would be where multiple treatments are given to a single patient. For example, we might compare a medical treatment with a placebo.

```r
outcomes <- matrix(c(40, 6, 8, 46), nrow = 2,
               dimnames = list("Treatment" = c("Better", "Not Better"),
                               "Placebo" = c("Better", "Not Better")))
outcomes
```

```
            Placebo
Treatment     Better Not Better
  Better          40          8
  Not Better       6         46
```

```r
mcnemar.test(outcomes)
```

```

    McNemar's Chi-squared test with continuity correction

data:  outcomes
McNemar's chi-squared = 0.071429, df = 1, p-value = 0.7893
```

Q: What are the null hypothesis and alternative hypothesis in this situation? How do and outcome of the test in this situation?

The null hypothesis would be that marginal probabilities (of Better) are the same for the placebo and treatment. The alternative would be that one of the interventions is more effective than the other. In this situation, there is not evidence to reject the null hypothesis.

Now generate data with the following joint probabilities:

- $\pi_{better,better} = .4$
- $\pi_{notbetter,notbetter} = .4$
- $\pi_{better,notbetter} = .15$
- $\pi_{notbetter,better} = .05$

Analyzing Rater Agreement

In a similar setting, we can assess similarities of raters (or tests).

- Cohen's Kappa Measure of Agreement

$$\kappa\frac{\sum \pi_{ii} - \sum \pi_{i+}\pi_{+i}}{1 - \sum \pi_{i+}\pi_{+i}}$$

κ is 0 expected under independence, κ is 1 perfect agreement.

```r
outcomes <- matrix(c(40, 6, 8, 46), nrow = 2,
               dimnames = list("Test 1" = c("Sick", "Not Sick"),
                               "Test 2" = c("Sick", "Not Sick")))
outcomes
```

```
          Test 2
Test 1      Sick Not Sick
  Sick        40        8
  Not Sick     6       46
```

```r
CohenKappa(outcomes)
```

```
[1] 0.7191011
```

```r
statement <- data.frame(
  A=c(2,3,1,3,1,2,1,2,3,3,3,3,3,2,1,3,3,2,2,1,
      2,1,3,3,2,2,1,2,1,1,2,3,3,3,3,3,1,2,1,1),
  B=c(2,2,2,1,1,2,1,2,3,3,2,3,1,3,1,1,3,2,1,2,
      2,1,3,2,2,2,3,2,1,1,2,2,3,3,3,3,2,2,2,3),
  C=c(2,2,2,1,1,2,1,2,3,3,2,3,3,3,3,2,2,2,2,3,
      2,2,3,3,2,2,3,2,2,2,2,3,3,3,3,3,3,2,2,2),
  D=c(2,2,2,1,1,2,1,2,3,3,2,3,3,3,3,3,2,2,2,2,
      3,1,3,2,2,2,1,2,2,1,2,3,3,3,3,3,3,2,2,1)
)

KappaM(statement)
```

```
[1] 0.5036937
```

Q: Which two raters are most in agreement?

```
KappaM(statement[,1:2])
```

```
[1] 0.4328922
```

```
KappaM(statement[,c(1,3)])
```

```
[1] 0.3697686
```

```
KappaM(statement[,c(1,4)])
```

```
[1] 0.5021541
```

```
KappaM(statement[,c(2,3)])
```

```
[1] 0.4701987
```

```
KappaM(statement[,c(2,4)])
```

```
[1] 0.5692609
```

```
KappaM(statement[,c(3,4)])
```

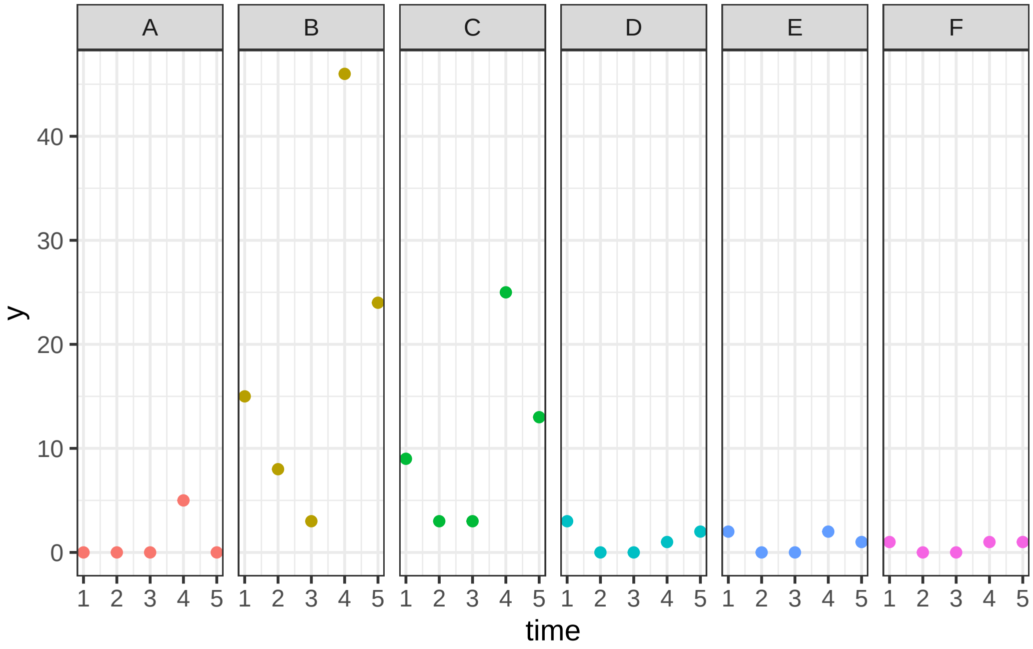
```
[1] 0.6647459
```

Modeling Correlated Data: GLMMs

Modeling Correlated Data:

- longitudinal studies or matched sets (case - control or family clusters)

Recall the bike rental data, now let's assume we have counts at different bike rental stations.



How might we model this data?

$y_{it} \sim Poisson(\mu_{it})$

$log(\mu_{it}) = \alpha_i + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5$

fixed effects

$y_{it} \sim Poisson(\mu_{it})$

$log(\mu_{it}) = \alpha_i + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5$

Maybe if there are a few stations in the dataset, we use a categorical variable in the modeling framework. In general there may be specific interest in those stations.

```
sim_data %>%
  mutate(time = as.factor(time)) %>%
  stan_glm(y ~ time + group, data = ., family = 'poisson', refresh = 0)
```

```
stan_glm
 family:       poisson [log]
 formula:      y ~ time + group
 observations: 30
 predictors:   10
------
            Median MAD_SD
(Intercept) -0.2    0.5  
time2       -1.0    0.4  
time3       -1.6    0.5  
time4        1.0    0.2  
time5        0.3    0.2  
groupB       3.0    0.5  
groupC       2.4    0.5  
groupD       0.2    0.6  
groupE       0.0    0.7  
groupF      -0.6    0.8  

------
* For help interpreting the printed output see ?print.stanreg
* For info on the priors used see ?prior_summary.stanreg
```

```
sim_data %>%
  mutate(time = as.factor(time)) %>%
  glm(y ~ time + group, data = ., family = 'poisson')
```

```

Call:  glm(formula = y ~ time + group, family = "poisson", data = .)

Coefficients:
```

```
(Intercept)       time2       time3       time4       time5      groupB 
 -1.133e-01  -1.003e+00  -1.609e+00   9.808e-01   3.124e-01   2.955e+00 
     groupC      groupD      groupE      groupF 
  2.361e+00   1.823e-01  -1.858e-10  -5.108e-01 

Degrees of Freedom: 29 Total (i.e. Null);  20 Residual
Null Deviance:	    358.4 
Residual Deviance: 17.46 	AIC: 103.5
```

- (Intercept) would be `u[1] + alpha + gamma_1` -0.6777621
- time2 would be an increase associated with time 2 relative to time1
- groupB would be an increase associated with group B relative to group A.

with these values, we can build out the values for each combination.

```
sim_data %>%
  mutate(time = as.factor(time)) %>%
model.matrix(y ~ time + group, data = .) |>
  head()
```

```
  (Intercept) time2 time3 time4 time5 groupB groupC groupD groupE groupF
1           1     0     0     0     0      0      0      0      0      0
2           1     1     0     0     0      0      0      0      0      0
3           1     0     1     0     0      0      0      0      0      0
4           1     0     0     1     0      0      0      0      0      0
5           1     0     0     0     1      0      0      0      0      0
6           1     0     0     0     0      1      0      0      0      0
```

random effects

An alternative approach is to use random effects.

- If there are a large number of stations in the dataset and we are less interested in the stations themselves, we can use a random effect model.
- Instead of directly estimating each we use a categorical variable in the modeling framework. In general there may be specific interest in those stations.

$y_{it} \sim Poisson(\mu_{it})$

$log(\mu_{it}) = \alpha_i + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5$

$\alpha_i \sim N(\alpha, \sigma^2_\alpha)$ and

$\alpha_i = \alpha + u_i$, $u_i \sim N(0, \sigma^2_\alpha)$

- We can use `glmer` in `lme4` or `stan_glmer` in `rstanarm`. We will use the notation `(1|variable)` to denote a random effect on the variable.

```r
sim_data %>%
  mutate(time = as.factor(time)) %>%
stan_glmer(y ~ time  + (1|group), data = ., family = 'poisson', refresh = 0)
```

```
stan_glmer
 family:       poisson [log]
 formula:      y ~ time + (1 | group)
 observations: 30
------
            Median MAD_SD
(Intercept)  0.7    0.7  
time2       -1.0    0.3  
time3       -1.6    0.5  
time4        1.0    0.2  
time5        0.3    0.2  

Error terms:
 Groups Name        Std.Dev.
 group  (Intercept) 1.7     
Num. levels: group 6 

------
* For help interpreting the printed output see ?print.stanreg
* For info on the priors used see ?prior_summary.stanreg
```

```
sim_data %>%
  mutate(time = as.factor(time)) %>%
glmer(y ~ time  + (1|group), data = ., family = 'poisson')
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: poisson  ( log )
Formula: y ~ time + (1 | group)
   Data: .
     AIC      BIC   logLik deviance df.resid
119.9907 128.3978 -53.9953 107.9907       24
Random effects:
 Groups Name        Std.Dev.
 group  (Intercept) 1.341
Number of obs: 30, groups:  group, 6
Fixed Effects:
(Intercept)        time2        time3        time4        time5
     0.7258      -1.0033      -1.6095       0.9808       0.3124
```

GLMMs for Categorical Data

1. Logistic Regression

$y_{it} \sim Bernoulli(n, \pi_{it})$

$logit(\mu_{it}) = \alpha_i + \gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5$

$\alpha_i \sim N(\alpha, \sigma_\alpha^2)$ and

$\alpha_i = \alpha + u_i$, $u_i \sim N(0, \sigma_\alpha^2)$