Dyadic Data Analysis

Contents

Read in the individual data (or a pairwise dataset)

```r
library(tidyr)

#install.packages("devtools")
#library(devtools)
#install_github("RandiLGarcia/dyadr")
library(dyadr)

#install.packages("nlme")
library(nlme)
library(dplyr)

acitelli_ind <- read.csv("/Users/randigarcia/Desktop/Data/acitelli.csv", header=TRUE)
```

Convert individual data to pairwise. If you imported a pairwise set, skip this chunk. I also create a gender variable that's a **factor** and has labels `hus` and `wife`. This variable will be useful later.

```r
tempA <- acitelli_ind %>%
  mutate(genderE = gender, partnum = 1) %>%
  mutate(gender = ifelse(gender == 1, "A", "P")) %>%
  gather(variable, value, self_pos:genderE) %>%
  unite(var_gender, variable, gender) %>%
  spread(var_gender, value)

tempB <- acitelli_ind %>%
  mutate(genderE = gender, partnum = 2) %>%
  mutate(gender = ifelse(gender == 1, "P", "A")) %>%
  gather(variable, value, self_pos:genderE)%>%
  unite(var_gender, variable, gender) %>%
  spread(var_gender, value)

acitelli_pair <- bind_rows(tempA, tempB) %>%
  arrange(cuplid) %>%
  mutate(gender_A = ifelse(genderE_A == 1, "hus", "wife"), gender_A = as.factor(gender_A

rm(tempA, tempB)
```

Indistinguishable Dyads

```r
apim_in <- gls(satisfaction_A ~ other_pos_A + other_pos_P,
               data = acitelli_pair,
               correlation = corCompSymm(form=~1|cuplid),
               na.action = na.omit)

smallsummary(apim_in)
```

```
## Correlation structure of class corCompSymm representing
##        Rho
## 0.4693414
##
## NULL
## Residual standard error: 0.4173659
##
##              Value Std.Error t-value p-value
## (Intercept) 0.6698    0.3224  2.0772  0.0387
## other_pos_A 0.4004    0.0473  8.4631  0.0000
## other_pos_P 0.2880    0.0473  6.0864  0.0000

##              2.5 % 97.5 %
## (Intercept) 0.0378 1.3017
```

```
## other_pos_A 0.3077 0.4932
## other_pos_P 0.1952 0.3807
```

Effect Sizes

To compute effect sizes, we use partial correlations.

```r
#install.packages("ppcor")
library(ppcor)

pcor.test(acitelli_pair$other_pos_A, acitelli_pair$satisfaction_A, acitelli_pair$other_p
```

```
## [1] 0.4232524
```

```r
pcor.test(acitelli_pair$other_pos_P, acitelli_pair$satisfaction_A, acitelli_pair$other_p
```

```
## [1] 0.3184723
```

The effect size for actor is .423, a moderate effect, and for partner .318, a moderate effect.

Multiple Correlation

How much variance in the response variable does the actor and partner effects explain together? First we run the empty model so that we can get the total variance in the response—which we need to calculate the pseudo-R^2.

```r
# Empty Model
apimie <- gls(satisfaction_A ~ 1 ,
            na.action=na.omit,
            correlation=corCompSymm (form=~1|cuplid),
            data=acitelli_pair)

summary(apimie)
```

```
## Generalized least squares fit by REML
##   Model: satisfaction_A ~ 1
##   Data: acitelli_pair
##        AIC      BIC    logLik
##   364.4574 375.5183 -179.2287
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | cuplid
##  Parameter estimate(s):
##     Rho
## 0.61847
##
```

```
## Coefficients:
##                Value  Std.Error  t-value p-value
## (Intercept) 3.60473 0.03674615 98.09815       0
##
## Standardized residuals:
##         Min         Q1        Med         Q3        Max
## -4.9061352 -0.5461333  0.4600210  0.7954058  0.7954058
##
## Residual standard error: 0.4969417
## Degrees of freedom: 296 total; 295 residual
```

```r
# sd of errors for the model
esd <- as.numeric(apim_in[6])

# sd of errors for the empty model
esd0 <- as.numeric(apimie[6])

# the R squared, using the crsp function
crsp(esd, esd0)
```

```
## [1] 0.29462
```

```r
#This will perform a likilihood ratio test for the set of all fixed effects in the mod
anova(apim_in, apimie)
```

```
##         Model df      AIC      BIC   logLik    Test L.Ratio p-value
## apim_in     1  5 306.7183 325.1191 -148.3591
## apimie      2  3 364.4574 375.5183 -179.2287 1 vs 2 61.73909  <.0001
```

Rho: ICC = .618470
Residual SE^2 empty model = .246951
Residual SE^2 standard model = .174194
Pseudo-R^2 = 1 - (.174194 / .246951) = .295

Called the "pseudo R2"—29.5% of the variance in satisfaction is explained by other positivity of the actor and the partner. Set it to zero if it's negative.

Interpretation of Model Estimates

Fixed Effects

Intercept: Predicted level of satisfaction for those scoring zero on the actor and partner variables. Because these variables are not centered, it is not all that meaningful.

other_pos_A or the Actor Variable: If you see your partner positively, are you satisfied in the relationship? Yes!

other_pos_P or the Partner effect: If your partner sees you positively, are you satisfied in the relationship? (Or: If you see your partner positively, is your partner satisfied in the relationship?) Yes!

Random Effects

`Residual SE`^{2} is the error or unexplained variance.
The partial ICC, or `Rho`, is .469.

Distingushable Dyads

Interaction Approach

1. Add distinguishing variable as a covariate. Note its coding.

2. Have the distinguishing variable interact with the actor and the partner effects.

3. These interactions evaluate whether actor and partner effects are the same for the two people.

4. Add a `weights =` argument to allow for different error variances for the two members.

```r
apim_di_int <- gls(satisfaction_A ~ other_pos_A + other_pos_P + genderE_A
                   + other_pos_A*genderE_A + other_pos_P*genderE_A,
                   data = acitelli_pair,
                   correlation = corCompSymm(form=~1|cuplid),
                   weights = varIdent(form=~1|genderE_A),
                   na.action = na.omit)

summary(apim_di_int)
```

```
## Generalized least squares fit by REML
##   Model: satisfaction_A ~ other_pos_A + other_pos_P + genderE_A + other_pos_A *
##   Data: acitelli_pair
##        AIC      BIC    logLik
##   322.3805 355.4094 -152.1902
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | cuplid
##  Parameter estimate(s):
##       Rho
## 0.4751092
## Variance function:
```

```
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | genderE_A
##  Parameter estimates:
##        1        -1
## 1.000000 1.203894
##
## Coefficients:
##                        Value Std.Error    t-value p-value
## (Intercept)            0.6508537 0.3250050  2.002596  0.0462
## other_pos_A            0.4010433 0.0472656  8.484896  0.0000
## other_pos_P            0.2915640 0.0482245  6.045967  0.0000
## genderE_A              0.0396052 0.1958905  0.202180  0.8399
## other_pos_A:genderE_A  0.0233433 0.0528291  0.441865  0.6589
## other_pos_P:genderE_A -0.0299142 0.0536888 -0.557177  0.5778
##
##   Correlation:
##                        (Intr) oth__A oth__P gndE_A o__A:E
## other_pos_A            -0.786
## other_pos_P            -0.797  0.263
## genderE_A              -0.207  0.101  0.226
## other_pos_A:genderE_A -0.006 -0.088  0.089 -0.411
## other_pos_P:genderE_A  0.182 -0.003 -0.278 -0.444 -0.631
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -4.6821686 -0.4599787  0.1298148  0.6321989  1.9431082
##
## Residual standard error: 0.3783372
## Degrees of freedom: 296 total; 290 residual
```

Interpretation of Effects

Intercept = .650854—The predicted score for husbands and wives who have a 0 on how positively they see the spouse (We should have centered!)

genderE_A = .039605—Husband are very slightly more satisfied (about .08 points more) than wives when you control for how they both view their spouse. (Recall wives are -1 on Gender_A and Husbands are +1; the difference between husbands and wives is then twice the difference of the effect of Gender_A.)

other_pos_A = .401043—Actor Effect: The more positively you view your spouse, the more satisfied you are in the marriage.

other_pos_P = .291564—Partner Effect: The more positively your partner views you, the more satisfied you are in the marriage.

genderE_A X other_pos_A = .023343—The actor effect is stronger for husbands.

genderE_A X other_pos_P = -.029914—The partner effect is stronger H -> W than

W -> H.

Actor Effect for Husbands = .401043 + .023343 = 0.424386
Actor Effect for Wives = .401043 - .023343 = 0.37770
Partner Effect for W -> H = .291564 + (-.029914) = 0.261650
Partner Effect for H -> W = .291564 - (-.029914) = 0.321478

.207460, error variance for Wives
.143139, error variance for Husbands

Or have R do it like so:

```r
sep_coef <- data.frame(
  othpos_a_men = coef(apim_di_int)[2] + 1*coef(apim_di_int)[5],
  othpos_a_wom = coef(apim_di_int)[2] + -1*coef(apim_di_int)[5],
  othpos_p_men = coef(apim_di_int)[3] + 1*coef(apim_di_int)[6],
  othpos_p_wom = coef(apim_di_int)[3] + -1*coef(apim_di_int)[6],
  row.names = "Randi's nums")

sep_coef
```

```
##              othpos_a_men othpos_a_wom othpos_p_men othpos_p_wom
## Randi's nums    0.4243866       0.3777    0.2616498    0.3214782
```

Getting Separate Error Variances

Notice is the model summary that we now get weights for the error variances. That is, we actually have two separate error variances. We can use the function `getevd` in the `dyadr` package to retrieve them.

```r
error_vars <- data.frame(ev_wom = getevd(apim_di_int)[1],
                         ev_men = getevd(apim_di_int)[2])

error_vars
```

```
##      ev_wom    ev_men
## 1 0.455478 0.3783372
```

0.455 is the error SD for women, and 0.378 is the error SD for men.

Two-Intercept Approach

This involves a trick by which one equation becomes two. We create two dummy variables: H_{ij} which equals 1 for husbands and 0 for wives and W_{ij} which equals 1 for wives and zero for husband. We then estimate the following equation:

$$Y_{ij} = b_H H_{ij} + a_H H_{ij} A_{ij} + p_H H_{ij} P_{ij} + H_{ij} e_{ij} + b_W W_{ij} + a_W W_{ij} A_{ij} + p_W W_{ij} P_{ij} + W_{ij} e_{ij}$$

Note that the equation has no ordinary intercept, but rather, in some sense, two intercepts, b_H and b_W. Note that when $H_{ij} = 1$ and $W_{ij} = 0$, the above becomes

$$Y_{ij} = b_H + a_H A_{ij} + p_H P_{ij} + e_{ij}$$

and when $H_{ij} = 0$ and $W_{ij} = 1$, the above becomes

$$Y_{ij} = b_W + a_W A_{ij} + p_W P_{ij} + e_{ij}$$

Thus, one equals becomes two and we have actor and partner for both members.

To implement this in R, we do the following:

1. Add distinguishing variable as a factor, using `gender_A` created above.
2. Have no intercept in the fixed model by adding `-1` to the formula.
3. Have the distinguishing variable (`gender_A`) interact with actor and partner effect, but no actor and partner main effects. We need to use `:` for this instead of `*`. Separate actor and partner effects will be estimated for each member.

4. Keep the `weights =` argument to allow for different error variances for the two members.

```
apim_di_two <- gls(satisfaction_A ~ gender_A + other_pos_A:gender_A + other_pos_P:gender
                   data = acitelli_pair,
                   correlation = corCompSymm(form=~1|cuplid),
                   weights = varIdent(form=~1|genderE_A),
                   na.action = na.omit)

summary(apim_di_two)

## Generalized least squares fit by REML
##   Model: satisfaction_A ~ gender_A + other_pos_A:gender_A + other_pos_P:gender_A -
##   Data: acitelli_pair
##        AIC      BIC    logLik
##   318.2216 351.2505 -150.1108
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | cuplid
##  Parameter estimate(s):
##        Rho
## 0.4751092
## Variance function:
##  Structure: Different standard deviations per stratum
```

```
##  Formula: ~1 | genderE_A
##  Parameter estimates:
##        1          -1
## 1.000000 1.203894
##
## Coefficients:
##                               Value Std.Error  t-value p-value
## gender_Ahus                0.6904589 0.3429034 2.013567  0.0450
## gender_Awife               0.6112485 0.4128195 1.480668  0.1398
## gender_Ahus:other_pos_A    0.4243866 0.0677157 6.267184  0.0000
## gender_Awife:other_pos_A   0.3777000 0.0739222 5.109428  0.0000
## gender_Ahus:other_pos_P    0.2616498 0.0614025 4.261221  0.0000
## gender_Awife:other_pos_P   0.3214782 0.0815225 3.943427  0.0001
##
##  Correlation:
##                           gndr_Ah gndr_Aw gndr_Ah:__A gndr_Aw:__A
## gender_Awife               0.475
## gender_Ahus:other_pos_A   -0.667  -0.317
## gender_Awife:other_pos_A  -0.267  -0.562  -0.111
## gender_Ahus:other_pos_P   -0.562  -0.267  -0.234         0.475
## gender_Awife:other_pos_P  -0.317  -0.667   0.475        -0.234
##                           gndr_Ah:__P
## gender_Awife
## gender_Ahus:other_pos_A
## gender_Awife:other_pos_A
## gender_Ahus:other_pos_P
## gender_Awife:other_pos_P  -0.111
##
## Standardized residuals:
##         Min         Q1        Med        Q3        Max
## -4.6821686 -0.4599787  0.1298148  0.6321989  1.9431082
##
## Residual standard error: 0.3783372
## Degrees of freedom: 296 total; 290 residual
```

Notice that these actor and partner estimates are the same as what we calculated above. But now we have p-values for them. So that's cool.

```r
rbind(sep_coef,
      model = coef(summary(apim_di_two))[3:6,1],
      p_values = coef(summary(apim_di_two))[3:6,4])
```

```
##                 othpos_a_men othpos_a_wom othpos_p_men othpos_p_wom
## Randi's nums 4.243866e-01 3.777000e-01 0.2616497695 0.3214781786
## model        4.243866e-01 3.777000e-01 0.2616497695 0.3214781786
## p_values     1.327103e-09 5.869872e-07 0.0000275178 0.0001007801
```

R² for Men and Women

We could also get R^2 for men and women separately.

```
apim_di_empty <- gls(satisfaction_A ~ gender_A,
                     data = acitelli_pair,
                     correlation = corCompSymm(form=~1|cuplid),
                     weights = varIdent(form=~1|genderE_A),
                     na.action = na.omit)

R_sqs <- data.frame(R_sq_wom = crsp(getevd(apim_di_two)[1],
                                    getevd(apim_di_empty)[1]),
                    R_sq_men = crsp(getevd(apim_di_two)[2],
                                    getevd(apim_di_empty)[2]))

R_sqs
```

```
##     R_sq_wom  R_sq_men
## 1 0.2615171 0.3287662
```

Pseudo R^2 for women is 0.26, or 26% of the variance in satisfaction for women is explained by the model.
Pseudo R^2 for men is 0.33, or 33% of the variance in satisfaction for men is explained by the model.

Descriptive Statistics by Gender

There are MANY ways to do this. I use the `dyplr` package.

```
acitelli_pair %>%
  select(genderE_A, other_pos_A, satisfaction_A) %>%
  na.omit() %>% #omits only if missing on the variables included in select()
  group_by(genderE_A) %>%
  summarise(other_pos_A_mean = mean(other_pos_A),
            other_pos_A_sd = sd(other_pos_A),
            satisfaction_A_mean = mean(satisfaction_A),
            satisfaction_A_sd = sd(satisfaction_A),
            n = n())
```

```
## # A tibble: 2 x 6
##    genderE_A other_pos_A_mean other_pos_A_sd satisfaction_A_mean
##        <dbl>            <dbl>          <dbl>               <dbl>
## 1         -1         4.245946      0.5227320            3.591216
## 2          1         4.281081      0.4739975            3.618243
## # ... with 2 more variables: satisfaction_A_sd <dbl>, n <int>
```
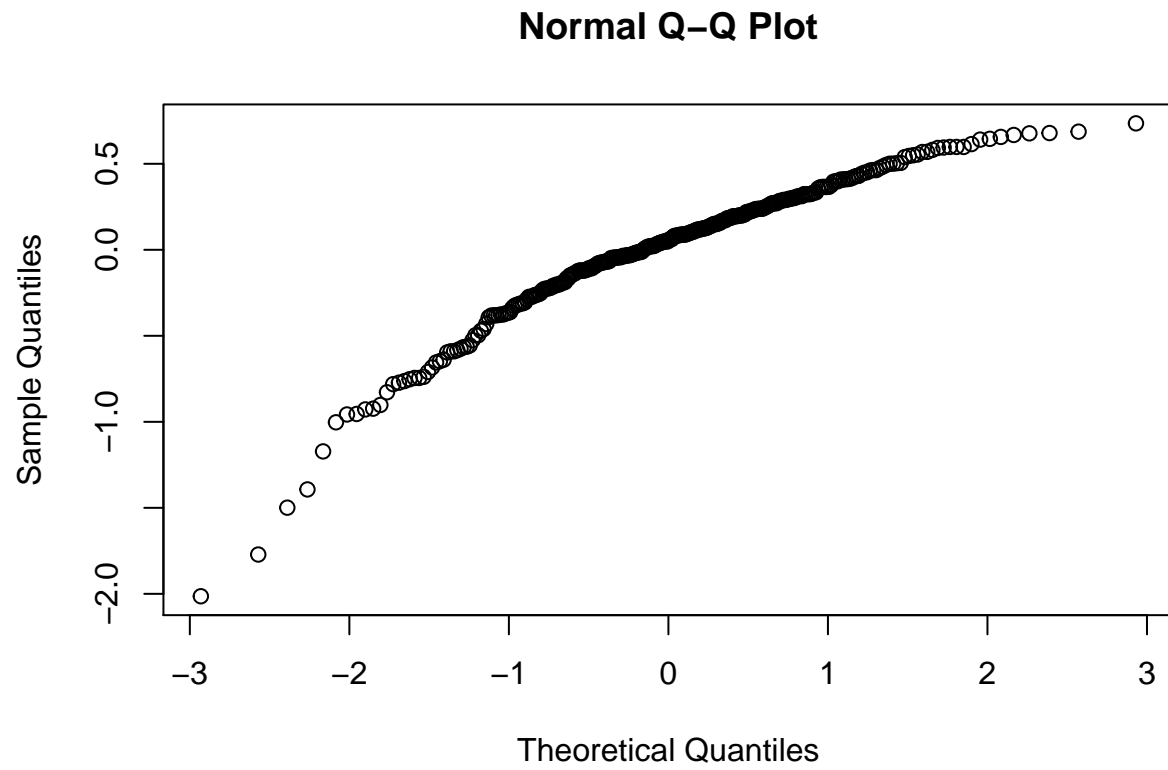
Model Diagnostics

We should also check that our assumptions are not violated.

```
plot(apim_di_int)
```
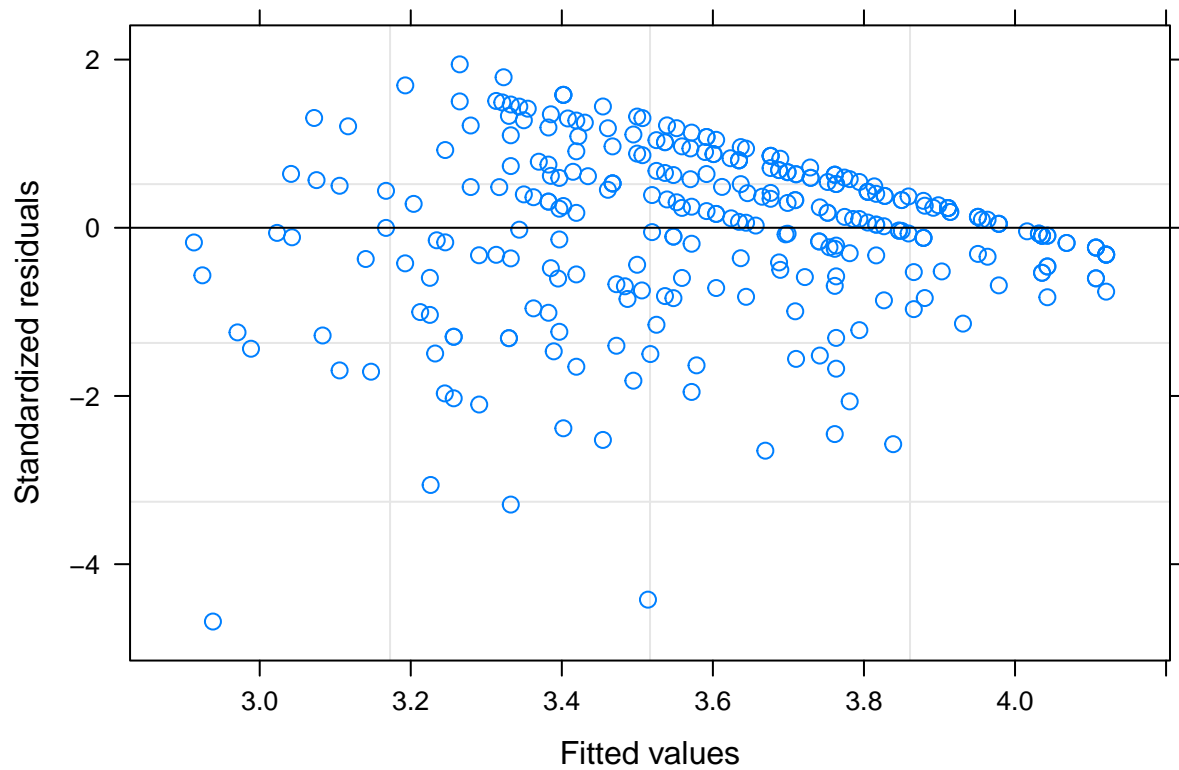


```
qqnorm(apim_di_int$residuals)
```

Normal Q–Q Plot

Satisfaction is skewed to the left.