More Multilevel Models

2017-03-20

Preparation

We will use two datasets located in the *nbaLevel1.sav* file and the *nbaLevel2.sav* file. These data include player-level attributes for $n = 300$ NBA players, and team-level attributes for $N = 30$ different teams. The source of these data is: Woltman, Feldstein, MacKay, & Rocchi (2012). We will use these data to explore the question of whether how good a player is (`Shots_on_five`) predicts variation in life satisfaction.

The player-level attributes in the *nbaLevel1.sav* file include:

- `Team_ID`: The team ID number for each player
- `Shots_on_five`: A proxy for player quality/success. This indicates the number of successful shots (out of five taken). Higher values indicate a more succesful player.
- `Life_Satisfaction`: Score on a survey of life satisfaction. Scores range from 5 to 25, with higher scores indicating more life satisfaction.

The team-level attributes in the *nbaLevel2.sav* file include:

- `Team_ID`:The team ID number
- `Coach_Experience`: Years of coaching experience in the NBA

We will use the `read_sav()` function from the **haven** library to import the data. We will also merge the two datasets together using the `left_join()` function from the **dplyr** package.

```
# Read in player-level and team-level data
library(haven)
nbaL1 = read_sav(file = "~/Google Drive/Documents/EPsy-8252/data/nbaLevel1.sav")
nbaL2 = read_sav(file = "~/Google Drive/Documents/EPsy-8252/data/nbaLevel2.sav")

# Merge the datasets
library(dplyr)
nba = left_join(nbaL1, nbaL2, by = "Team_ID")
head(nba)
```

Team_ID	Shots_on_five	Life_Satisfaction	Coach_Experience
01	3	18.8	2
01	3	18.0	2
01	4	21.0	2
01	4	20.5	2
01	3	19.0	2
01	2	12.1	2

```
# Load other libraries
library(AICcmodavg)
library(ggplot2)
library(lme4)
library(sm)
```

Multilevel Model Equations

Just like in conventional regression, we can mathematically express the multilevel models. Unlike conventional regression, there are multiple sets of models that we need to express. We need to express a level-1 model (player-level) and at least one level-2 model (team-level). Below is the set of models we use to mathematically express the multilvele model we fitted in the previous set of notes.

Level-1 :

$$\text{Life Satisfaction}_{ij} = \beta_{0j} + \beta_{1j}(\text{Player Success}_{ij}) + \epsilon_{ij}$$

Level-2 :

$$\beta_{0j} = \gamma_{00} + u_{0j}$$
$$\beta_{1j} = \gamma_{10} + u_{1j}$$

In these equations, i indicates that the term can vary across players, and j indicates the term can vary across teams. In general, the level-1 model specifies how the player-level predictors relate to the outcome, and the level-2 models specify how EACH coefficient in the level-1 model is predicted by team-level predictors. In this particular example, the level-1 model indicates that the ith player on the jth team has a life satisfaction score that is a function of both a team-level intercept and the product of that player's success and the team-level slope, and some residual term. The level-2 model for intercept indicates that the team-level intercept is a function of some overall team average and a residual from the specific team-intercept to that overall average. Similarly the team-level slope is a function of some overall team average and a residual from the specific team-slope to that overall average. Expressing the models so that they separate the level-1 and level-2 models is what is referred to as the *multilevel model*.

In the level-1 model, note that for a particular team (a particular j value) the coefficients (intercept and slope) would be the same for all players on that team. Players on a team are allowed to vary in their success and their life satisfaction. They would also vary in their residuals. Because the coefficients would be constant on a team, the only source of variation in Level-1 model is player-to-player (within-team) variation. Any unaccounted for player-to-player variation is absorbed into the residual term (similar to conventional regression). To further explain player-to-player variation we include predictors in the level-1 model.

In the level-2 models, note that the outcomes are the level-1 coefficients. There is one level-2 model for each level-1 coefficient. Secondly, note that there are no i subscripts in these models …only j subscripts. This means that the only source of variation is team-to-team variation. Any covariates we want to include that explain variation between teams would appear in the level-2 models. The residual terms in the level-2 models, recall, are referred to as *random effects*.

Mixed-Effects Model

Some people refer to a multilevel model analysis as a *mixed-effects model* analysis. This is because in order to do the analysis, most software packages require us to express the multilevel models as a mixed-effects model. A mixed-effects model simply expresses the multilevel models in a single equation. To do this, we will substitute the level-2 models into the level-1 equation.

$$\begin{aligned}\text{Life Satisfaction}_{ij} &= \Big[\gamma_{00} + u_{0j}\Big] + \Big[\gamma_{10} + u_{1j}\Big](\text{Player Success}_{ij}) + \epsilon_{ij} \\ &= \gamma_{00} + u_{0j} + \gamma_{10}(\text{Player Success}_{ij}) + u_{1j}(\text{Player Success}_{ij}) + \epsilon_{ij}\end{aligned}$$

We also typically re-arrange the terms from the mixed-effects model to move the level-2 residual terms (random-effect terms) and the level-1 residual term together.

$$\text{Life Satisfaction}_{ij} = \gamma_{00} + \gamma_{10}(\text{Player Success}_{ij}) + \Big[u_{0j} + u_{1j}(\text{Player Success}_{ij}) + \epsilon_{ij}\Big]$$

Here the γ-terms represent the fixed-effects of intercept and player success, respectively. They are akin to that overall average intercept and slope. The u-terms represent the random-effects for intercept and slope, respectively. These are the deviations to the team-specific intercept and slope. Finally, there is a player specific residual term. Since we include both the fixed-effects and the random-effects in the same equation, this is referred to as the *mixed-effects model*.

Expressing the multilevel models in a single mixed-effects model helps us specify the R syntax we will use in the `lmer()` function. Recall that the foirmula syntax we used (using Y and X rather than the variable names) was,

$$\texttt{Y} \sim \texttt{1} + \texttt{X} + (\texttt{1} + \texttt{X} \mid \texttt{Group})$$

This is exactly the specification in the mixed-effects model.

$$Y_{ij} = \gamma_{00}(1) + \gamma_{10}(X_{ij}) + \Big[u_{0j}(1) + u_{1j}(X_{ij}) + \epsilon_{ij}\Big]$$

Another thing that becomes evident once we express the equation as a mixed-effects model is that the main difference between conventional regression and multilevel regression is that we are partitioning the residual term into multiple parts. If I use γ rather than β in the conventional regression model,

$$Y_{ij} = \gamma_{00} + \gamma_{10}(X_{ij}) + \Big[\epsilon_{ij}\Big]$$

The mixed-effects model is

$$Y_{ij} = \gamma_{00} + \gamma_{10}(X_{ij}) + \Big[u_{0j} + u_{1j}(X_{ij}) + \epsilon_{ij}\Big]$$

If we used the same predictors in the conventional regression and in a multilevel regression, the difference would be in the residual terms. The multilevel regression partitions the unexplained part of the outcome (the residual terms) as a function of team-intercept error, team-slope error, AND player-level error. The conventional regression, since it doesn't account for teams, basically attributes all of that error to the player-level.

How does this help us? First recall that all statistical tests (e.g., t-tests, F-tests) use a measure of the player-level error in the denominator. If we can remove the team-level error from the player-level error, all things being equal, the player-level error should get smaller. This means the denominator of our test statistic will get smaller, making the statistic larger. This gives us more statistical power (e.g., smaller p-values) and more precision in our confidence intervals. So, not only are we accounting for the dependency in the residuals correctly, we also get the advantage of more statistical power.

LMER Output

Below we will fit the mixed-effects model using `lmer()`. Then we will examine the output using the `summary()` function.

```r
# Fit multilevel model
lmer.1 = lmer(Life_Satisfaction ~ 1 + Shots_on_five + (1 + Shots_on_five | Team_ID), data = nba)
summary(lmer.1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Life_Satisfaction ~ 1 + Shots_on_five + (1 + Shots_on_five |
##     Team_ID)
##    Data: nba
##
## REML criterion at convergence: 1379
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5373 -0.6504  0.0299  0.7088  2.3887
##
## Random effects:
##  Groups   Name          Variance Std.Dev. Corr
##  Team_ID  (Intercept)   0.0928   0.305
##           Shots_on_five 0.0991   0.315    1.00
##  Residual               5.1062   2.260
## Number of obs: 300, groups:  Team_ID, 30
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)      6.430      0.317    20.3
## Shots_on_five    3.289      0.134    24.6
##
## Correlation of Fixed Effects:
##             (Intr)
## Shots_on_fv -0.728
```

The first line of the `summary()` output tells us `Linear mixed model fit by REML`. This indicates the estimation method used to fit the model. REML is the abbreviation for Restricted Maximum Likelihood. This is different than Maximum Likelihood. For now, just take it at face value. After this, the formula used in the `lmer()` function is provided, followed by the REML value at convergence. The `at convergence` indicates that the estimation was iterative and the estimates converged to particualr values. Then there is some information about the fitted models' scaled residuals (which we will ignore).

The two sections of the output that are of most interest are (1) the fixed-effects, and (2) the random-effects.

Fixed-Effects Output

The fixed-effects section gives the fitted estimates of the γ-terms. These estimates are interpreted in the same manner that we would interpret the fitted coefficients from a conventional regression.

- $\hat{\gamma}_{00} = 6.43$: The predicted average life satisfaction score for all players with a success of zero is 6.43.
- $\hat{\gamma}_{10} = 3.29$: Players with a one-unit difference in success have, on average, a predicted difference in life satisfaction scores of 3.29.

The output also includes standard errors and t-values for these coefficients. There are no p-values given for results from a mixed-effects analysis in R. That is because it is theoretically unclear what value for the df should be used for the t-test. Rather than guess at this, the developers of the **lme4** package chose not to give p-values.

One compromise applied researchers use is to rely on the t-vlaues; any t-value greater than 2 (or less than -2) is likely statistically relevant. We will use information criteria to make decisions about predictors, then just

report the estimates from the final model (recall that p-values and information criteria are two incompatible frameworks for statistical decision making).

Random-Effects Output

The random-effects section reports the variation in all the random-effects terms and in the level-1 residuals. The variances of the random-effect terms are referred to as *variance components* in the language of multilevel modeling. The variance components are typically denoted using the Greek letter tau (τ).

- $\hat{\tau}_{00} = \text{Var}(\hat{u}_{0j}) = 0.093$
- $\hat{\tau}_{11} = \text{Var}(\hat{u}_{1j}) = 0.099$

What do the subscripts on the τ-terms indicate? Well, it turns out that typically we expess the variance components for the random-effects in a matrix:

$$\begin{bmatrix} \text{Var}(u_{0j}) & \text{Cov}(u_{0j}, u_{1j}) \\ \text{Cov}(u_{1j}, u_{0j}) & \text{Var}(u_{1j}) \end{bmatrix}$$

This is called a ⋆variance–covariance matrix of the random-effects⋆⋆ since its elements are the variances and covariances of the different random-effects. The subscripts on the τ-terms indicate the row and column numbers in this matrix (if we start counting at 0).

$$\begin{bmatrix} \tau_{00} & \tau_{01} \\ \tau_{10} & \tau_{11} \end{bmatrix}$$

Note that this matrix is symmetric as the covariance terms, τ_{01} and τ_{10} are equal. We can get all of the τ estimates using the `VarCorr()` function and accessing the `Team_ID` column. (You can figure out wha tthe column is called by using the `names()` function on the `VarCorr()` output.)

```
# names(VarCorr(lmer.1))
VarCorr(lmer.1)$Team_ID
```

	(Intercept)	Shots_on_five
(Intercept)	0.093	0.096
Shots_on_five	0.096	0.099

The output includes a matrix giving the estimates for the variances and covariances, τ-terms. Note that the variance terms are exactly the same as those given in the `summary()` output. However, the covariance is not provided in the `summary()` output.

$$\begin{bmatrix} \hat{\tau}_{00} = 0.093 & \hat{\tau}_{01} = 0.096 \\ \hat{\tau}_{10} = 0.096 & \hat{\tau}_{11} = 0.099 \end{bmatrix}$$

The variance of the level-1 residual is typically denoted as σ^2_ϵ, similar to how we denoted it in the conventional regression.

- $\hat{\sigma}^2_\epsilon = \text{Var}(\hat{\epsilon}_{ij}) = 5.106$

References

Woltman, H., Feldstein, A., MacKay, J. C., & Rocchi, M. (2012). An introduction to hierarchical linear modeling. *Tutorials in Quantitative Methods for Psychology*, *8*(1), 52–69.