

Multilevel Models: ~~Longitudinal Data~~

Andrew Zieffler

March 29, 2016

Load Libraries

```
library(AICcmodavg)
library(dplyr)
library(foreign)
library(ggplot2)
library(lme4)
library(sm)
```

Load and Prepare Data

```
nbaL1 = read.spss(file = "~/Documents/EPsy-8252/data/nbaLevel1.sav", to.data.frame = TRUE)
```

```
## re-encoding from CP1252
```

```
nbaL2 = read.spss(file = "~/Documents/EPsy-8252/data/nbaLevel2.sav", to.data.frame = TRUE)
```

```
## re-encoding from CP1252
```

```
nba = left_join(nbaL1, nbaL2, by = "Team_ID")
head(nba)
```

```
##   Team_ID Shots_on_five Life_Satisfaction Coach_Experience
## 1      01             3             18.804                2
## 2      01             3             18.000                2
## 3      01             4             21.000                2
## 4      01             4             20.500                2
## 5      01             3             19.000                2
## 6      01             2             12.100                2
```

Fit Models

Here we will fit three models we considered in the last set of notes (using REML). These are chosen to illustrate several concepts, not because they were necessarily adopted.

```
lmer.0 = lmer(Life_Satisfaction ~ 1 + (1 | Team_ID), data = nba)
lmer.1 = lmer(Life_Satisfaction ~ 1 + Shots_on_five + (1 | Team_ID), data = nba)
lmer.2 = lmer(Life_Satisfaction ~ 1 + Shots_on_five + (1 + Shots_on_five | Team_ID), data = nba)
```

Variance–Covariance Matrix of the Random Effects

It is common to provide covariances (rather than correlations) when reporting results from multilevel models. To obtain the variance–covariance matrix of the random effects, we use the `varCorr()` function. Without accessing the `Team_ID` component (the cluster variable), this function would just return the RE part of the `summary()` output.

```
# Get variance-covariance matrix of the random effects
VarCorr(lmer.2)$Team_ID
```

```
##              (Intercept) Shots_on_five
## (Intercept)   0.09279452   0.09590839
## Shots_on_five 0.09590839   0.09912676
## attr(,"stddev")
##   (Intercept) Shots_on_five
##     0.3046219   0.3148440
## attr(,"correlation")
##              (Intercept) Shots_on_five
## (Intercept)              1              1
## Shots_on_five            1              1
```

If you want only the variance–covariance matrix, then we use indexing to extract only those elements that are related to this matrix, in this case the first two rows and the first two columns.

```
VarCorr(lmer.2)$Team_ID[1:2, 1:2]
```

```
##              (Intercept) Shots_on_five
## (Intercept)   0.09279452   0.09590839
## Shots_on_five 0.09590839   0.09912676
```

- $\sigma_0^2 = 0.093$
- $\sigma_1^2 = 0.099$
- $\sigma_{01} = 0.096$

All three estimates should be reported in a summary or write-up of results.

In Model 0 and Model 1, the only RE is the intercept, so the variance–covariance matrix of the RE would be a 1x1 matrix.

```
VarCorr(lmer.0)$Team_ID
```

```
##              (Intercept)
## (Intercept)   14.9592
## attr(,"stddev")
## (Intercept)
##     3.867713
## attr(,"correlation")
##              (Intercept)
## (Intercept)              1
```

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

```
##              (Intercept)
## (Intercept)  0.8958596
## attr(,"stddev")
## (Intercept)
##    0.9464986
## attr(,"correlation")
##              (Intercept)
## (Intercept)          1
```

Assumptions

In fixed-effects regression models we looked at two plots to evaluate whether assumptions were satisfied: (1) density plot of the residuals, and (2) scatterplot of the residuals vs. the fitted values. We are going to look at these same plots with multilevel models, but now there is a set of fitted values and residuals for each equation in the multilevel model. Consider a model which includes random effects for both intercepts and slopes, and a level-2 predictor for both:

$$\begin{aligned}\text{Life Satisfaction}_i &= \beta_0^* + \beta_1^*(\text{SO5}) + \epsilon_i \\ \beta_0^* &= \beta_{00} + \beta_{01}(\text{Coach}) + \eta_{0j} \\ \beta_1^* &= \beta_{10} + \beta_{11}(\text{Coach}) + \eta_{1j}\end{aligned}$$

```
lmer.3 = lmer(Life_Satisfaction ~ 1 + Shots_on_five + Coach_Experience +
              Coach_Experience:Shots_on_five + (1 + Shots_on_five | Team_ID),
              data = nba)
```

Level-1 Assumptions

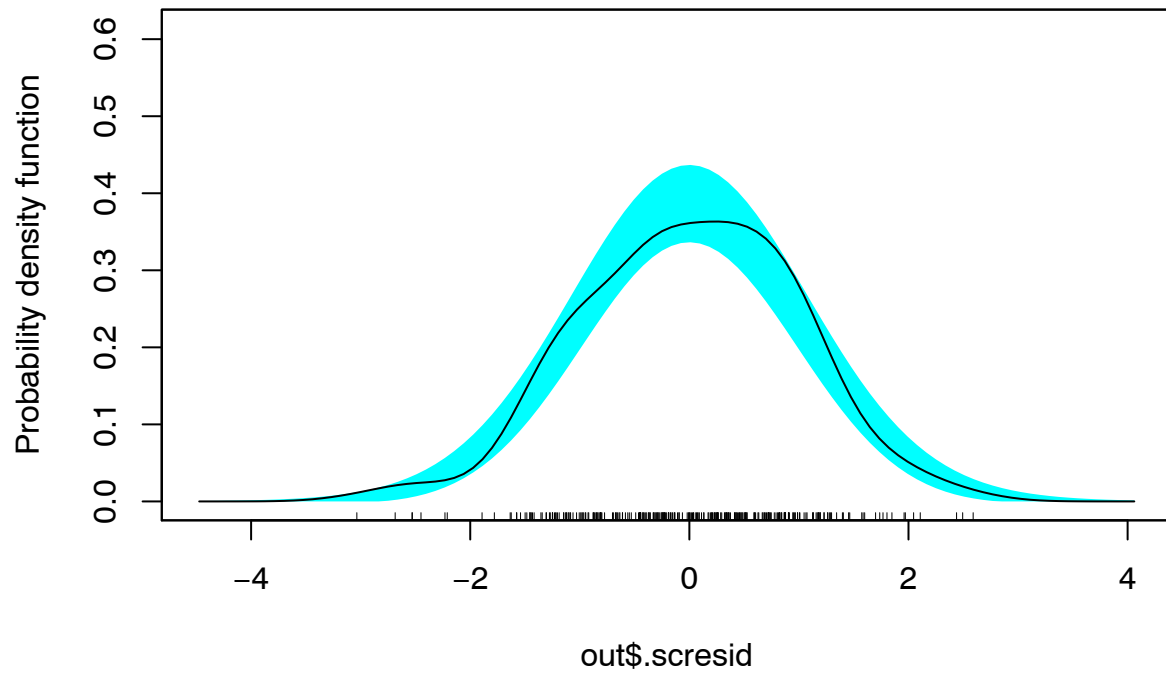
For the level-1 model we will examine (1) a density plot of the level-1 residuals, and (2) a scatterplot of the level-1 residuals vs. the fitted life satisfaction values. We will look for the same things in this plot as we did in the fixed-effects regression. The `fortify()` function works on a model fitted using `lmer()` as well. Here you get the typical columns. The `.scredid` column are *scaled residuals*.

```
out = fortify(lmer.3)
head(out)
```

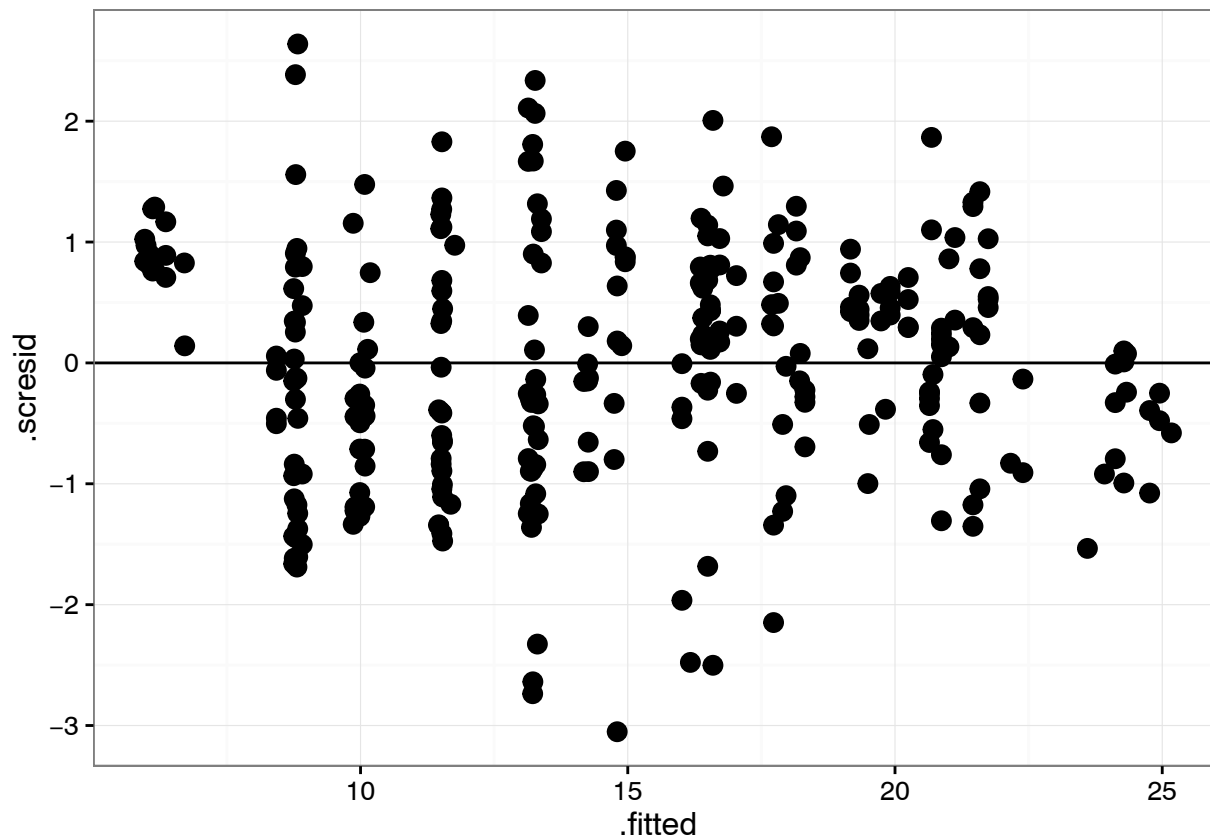
```
##   Team_ID Shots_on_five Life_Satisfaction Coach_Experience  .fitted
## 1      01             3           18.804             2 16.49623
## 2      01             3           18.000             2 16.49623
## 3      01             4           21.000             2 19.73788
## 4      01             4           20.500             2 19.73788
## 5      01             3           19.000             2 16.49623
## 6      01             2           12.100             2 13.25458
##      .resid  .scredid
## 1 2.3077720 1.0507979
## 2 1.5037711 0.6847122
## 3 1.2621225 0.5746823
```

```
## 4  0.7621225  0.3470173
## 5  2.5037711  1.1400422
## 6 -1.1545803 -0.5257151
```

```
sm.density(out$.screid, model = "normal")
```



```
ggplot(data = out, aes(x = .fitted, y = .screid)) +  
  geom_point(size = 3) +  
  theme_bw() +  
  geom_hline(yintercept = 0)
```



Level-2 Residuals: Intercept

For the level-2 intercept model we will examine (1) a density plot of the level-2 residuals from the intercept model, and (2) a scatterplot of the level-2 residuals from the intercept model vs. the fitted intercept ($\hat{\beta}_0^*$) values. We will again look for the same things in these plots as we did before.

The level-2 residuals and fitted values do not exist in a data frame, so we will compute them. The residuals are just the REs for intercept, so they are easy to access. The fitted values at level-2 we can compute by using the level-2 fitted model: $\hat{\beta}_0^* = \hat{\beta}_{00} + \hat{\beta}_{01}(\text{Coach})$. This is at the team level ($n = 30$), so we will use the level-2 data set to compute these.

```
# Compute the residuals
```

```
res0 = ranef(lmer.3)$Team_ID[ , 1]
```

```
# Get fixed effects
```

```
fixef(lmer.3)
```

```
##              (Intercept)              Shots_on_five
##              4.575960              2.676017
##      Coach_Experience Shots_on_five:Coach_Experience
##              1.229287              0.204538
```

```
# Compute the fitted values
```

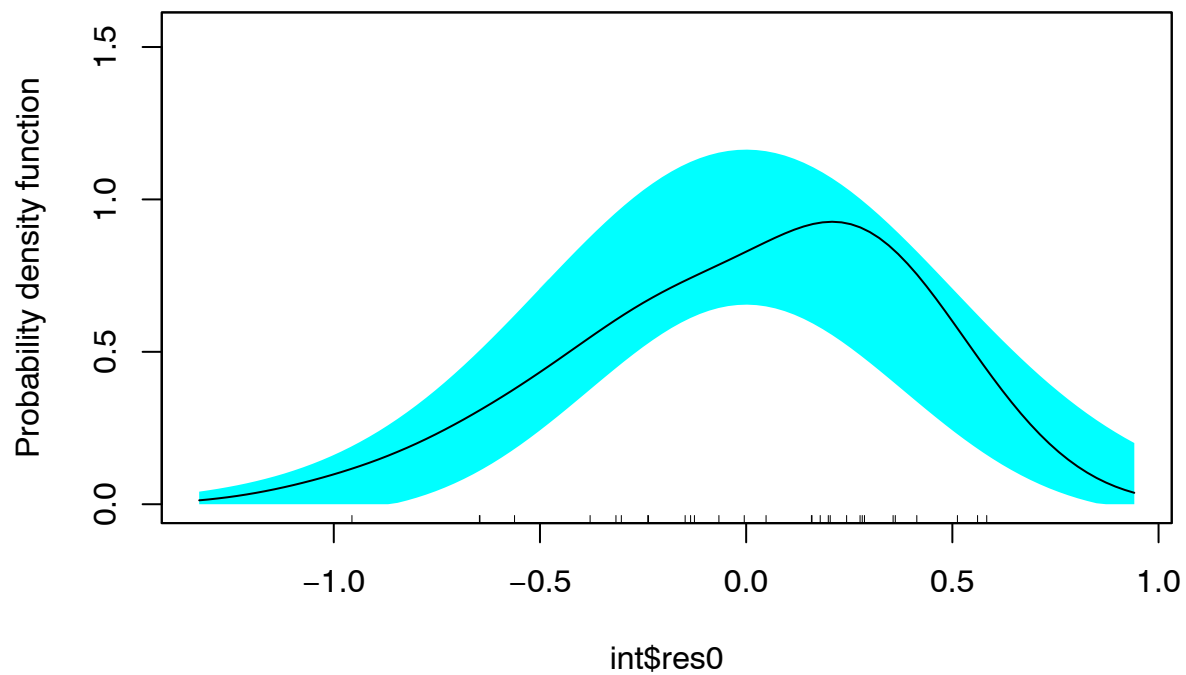
```
fit0 = 4.575960 + nbaL2$Coach_Experience*1.229287
```

```
# Put in data frame
```

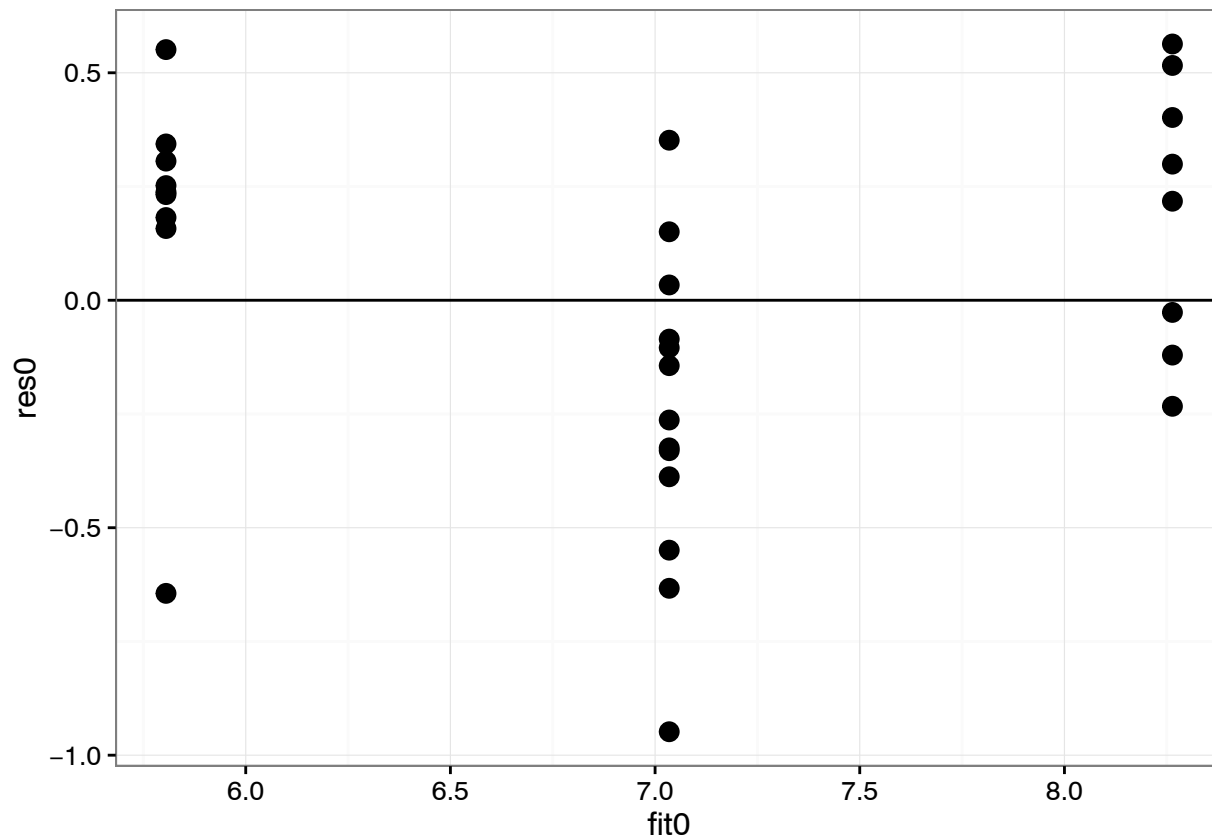
```
int = data.frame(fit0, res0)
head(int)
```

```
##      fit0      res0
## 1 7.034534 -0.26325027
## 2 8.263821 -0.12052607
## 3 7.034534 -0.94848444
## 4 7.034534  0.03363163
## 5 5.805247  0.25200660
## 6 7.034534 -0.63315005
```

```
# Density plot
sm.density(int$res0, model = "normal")
```



```
# Scatterplot of residuals vs fitted values
ggplot(data = int, aes(x = fit0, y = res0)) +
  geom_point(size = 3) +
  theme_bw() +
  geom_hline(yintercept = 0)
```



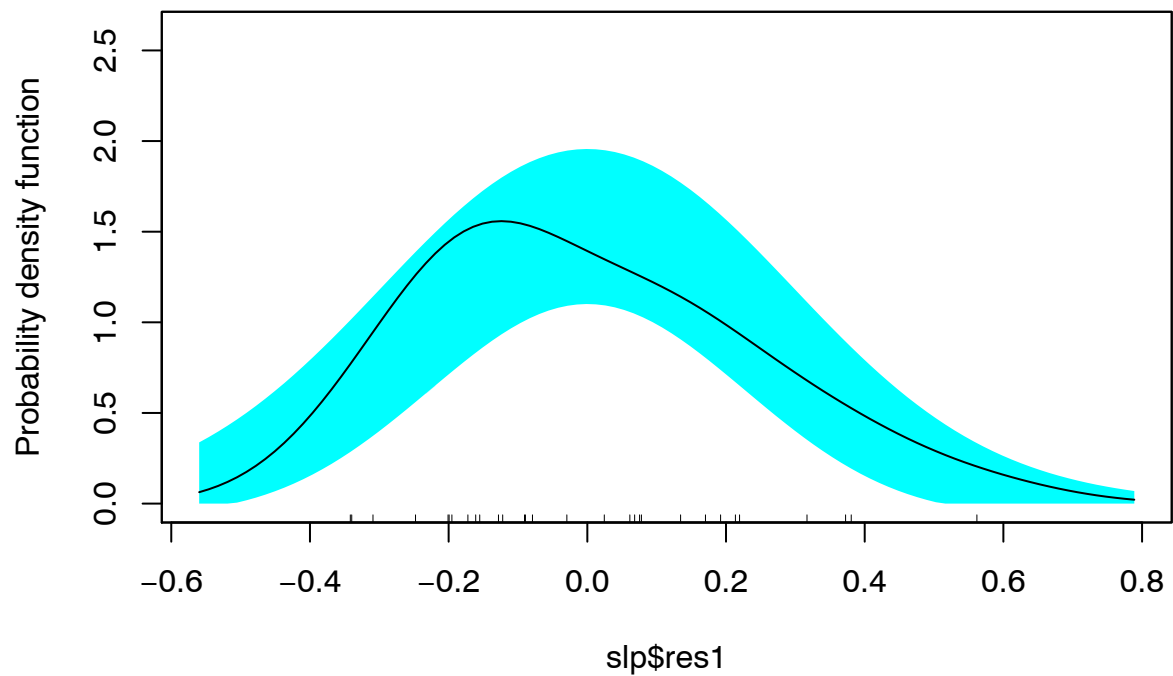
Level-2 Residuals: Slope

We can repeat this process to examine the residuals from the slope model.

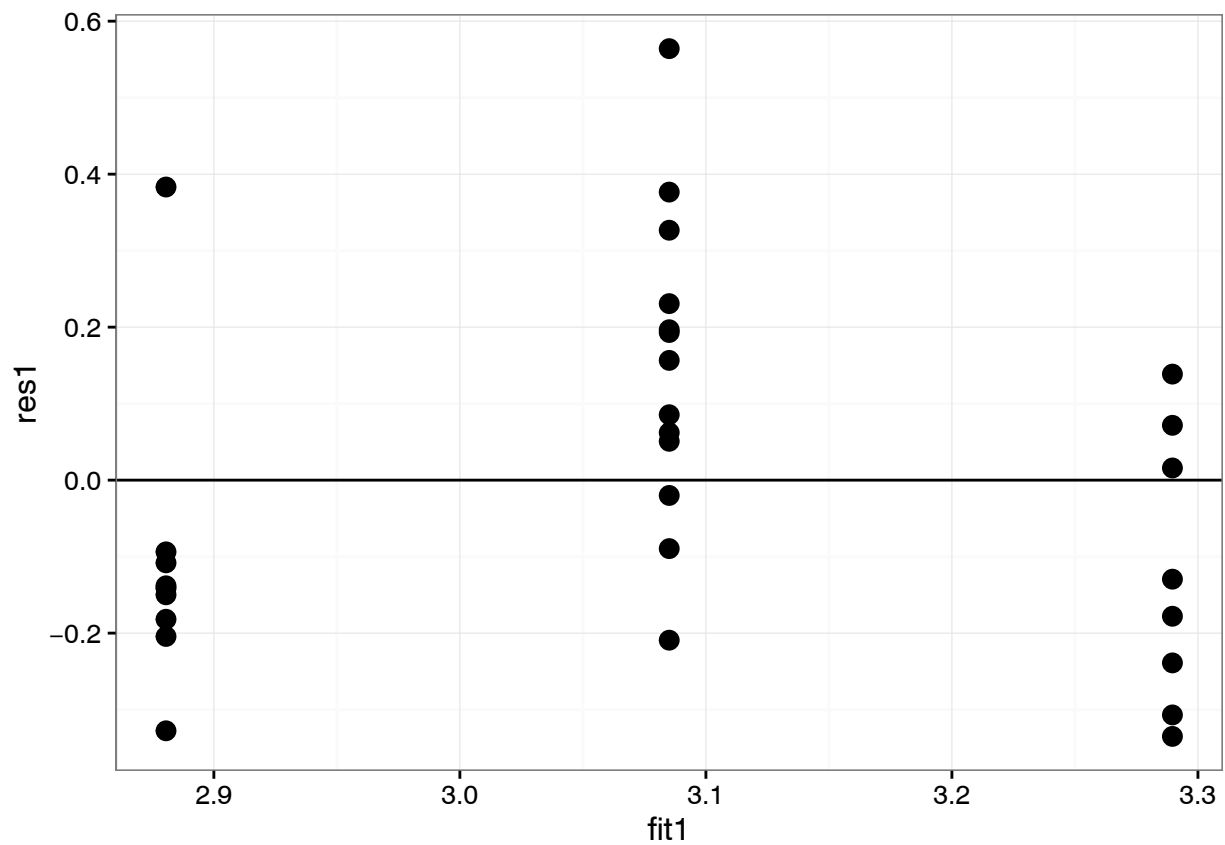
```
# Compute the residuals and fitted values
slp = data.frame(
  res1 = ranef(lmer.3)$Team_ID[ , 2],
  fit1 = 2.676017 + nbaL2$Coach_Experience*0.204538
)
head(slp)
```

```
##      res1    fit1
## 1  0.15655587 3.085093
## 2  0.07167729 3.289631
## 3  0.56406707 3.085093
## 4 -0.02000085 3.085093
## 5 -0.14986923 2.880555
## 6  0.37653659 3.085093
```

```
# Density plot
sm.density(slp$res1, model = "normal")
```



```
# Scatterplot of residuals vs fitted values  
ggplot(data = slp, aes(x = fit1, y = res1)) +  
  geom_point(size = 3) +  
  theme_bw() +  
  geom_hline(yintercept = 0)
```



Pseudo R^2 Values

In fixed effects model, the R^2 value summarized the variation in Y accounted for by the model. There were two ways that we could arrive at this value. The first is to compute the reduction in the residual variation from the intercept only model. The second is to compute the squared correlation between the Y and \hat{Y} values. In fixed effects regression these gave the same result.

We will start by computing the squared correlation between the Y and \hat{Y} values. Again, we can use the fortified data.

```
cor(out$Life_Satisfaction, out$.fitted) ^ 2
```

```
## [1] 0.8417534
```

This we typically interpret as the percentage of variation in Y accounted for by the model.

There are three computations we will need to make to compute the psudo- R^2 values based on the reduction in variance, that for: level-1, level-2 intercept, and level-2 slope residuals. Note in the fixed effects model we always compared the model under study to the intercept only model. For the level-1 and level-2 intercept residuals, we will compare the model under study to the unconditional means model. Consider the RE portion of the `summary()` output for Model 3:

```
Random effects:
  Groups   Name                Variance Std.Dev. Corr
  Team_ID  (Intercept)         0.4202   0.6482
           Shots_on_five     0.1486   0.3855  -1.00
  Residual                    4.8233   2.1962
Number of obs: 300, groups: Team_ID, 30
```

and for the unconditional means model:

```
Random effects:
  Groups   Name                Variance Std.Dev.
  Team_ID  (Intercept)        14.96     3.868
  Residual                    14.61     3.822
Number of obs: 300, groups: Team_ID, 30
```

We can compute the reduction in Level-1 residual variance as:

$$R_{\epsilon}^2 = \frac{14.61 - 4.8233}{14.61} = 0.670$$

The reduction in level-1 variance (from Model 0) was 67.0%. Similarly for the level-2 intercept residuals:

$$R_{\eta_0}^2 = \frac{14.96 - 0.4202}{14.96} = 0.972$$

The reduction in level-2 variance (from Model 0) was 97.2%.

What about the reduction in level-2 residual slope varianbce? The unconditional means model does not include level-2 residuals for slope so we cannot compare a model under study to that model's slope residuals. Instead we compare them to the simplest model that includes slope residuals; namely Model 2.

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
Team_ID	(Intercept)	0.09279	0.3046	
	Shots_on_five	0.09913	0.3148	1.00
Residual		5.10616	2.2597	

Number of obs: 300, groups: Team_ID, 30

$$R_{\eta_1}^2 = \frac{0.09913 - 0.1486}{0.09913} = -.499$$

The reduction in level-2 slope variance (from Model 2) was -49.9% .

In mixed effects models, these different computations for the pseudo- R^2 give give differing results.

$$R_{Y,\hat{Y}}^2 = 0.842$$

$$R_{\epsilon}^2 = 0.670$$

$$R_{\eta_0}^2 = 0.972$$

$$R_{\eta_1}^2 = -.499$$

Which of these you report depends on what you are interested in, level-1 variation or level-2 variation. Also note that these computations can produce negative results; an increase in variance.

Plotting Multilevel Models

There are two plots that you might want to consider: (1) a plot of the fixed effects model; and (2) a plot of each team's model (fixed effects + random effects). Let's consider a model that includes RE for both intercept and slopes, and a level-2 predictor, coaching experience, for the intercept:

$$\begin{aligned}\text{Life Satisfaction}_i &= \beta_0^* + \beta_1^*(\text{SO5}) + \epsilon_i \\ \beta_0^* &= \beta_{00} + \beta_{01}(\text{Coach}) + \eta_{0j} \\ \beta_1^* &= \beta_{10} + \eta_{1j}\end{aligned}$$

```
lmer.4 = lmer(Life_Satisfaction ~ 1 + Shots_on_five + Coach_Experience +  
              (1 + Shots_on_five | Team_ID), data = nba)  
summary(lmer.4)
```

```
## Linear mixed model fit by REML ['lmerMod']  
## Formula: Life_Satisfaction ~ 1 + Shots_on_five + Coach_Experience + (1 +  
##       Shots_on_five | Team_ID)  
## Data: nba  
##  
## REML criterion at convergence: 1341.4  
##  
## Scaled residuals:  
##      Min       1Q   Median       3Q      Max  
## -3.10163 -0.70385  0.08225  0.74557  2.67564  
##  
## Random effects:
```

```
## Groups Name Variance Std.Dev. Corr
## Team_ID (Intercept) 0.3915 0.6257
## Shots_on_five 0.1244 0.3528 -1.00
## Residual 4.8669 2.2061
## Number of obs: 300, groups: Team_ID, 30
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 3.7829 0.3983 9.498
## Shots_on_five 3.0958 0.1395 22.190
## Coach_Experience 1.6593 0.2245 7.392
##
## Correlation of Fixed Effects:
## (Intr) Shts__
## Shots_on_fv -0.249
## Coch_Exprnc -0.653 -0.505
```

The fixed effects part of that model is:

$$\text{Life Satisfaction}_i = 3.78 + 3.10(\text{SO5}_i) + 1.66(\text{Coach}_j)$$

To plot the fixed effects model, we (1) create a dataset with the appropriate predictors; (2) use the model to estimate the Y-hats; and (3) plot.

```
# Create data set
plotData = expand.grid(
  Shots_on_five = seq(from = 0, to = 5, by = 0.1),
  Coach_Experience = 1.967 # set to mean
)

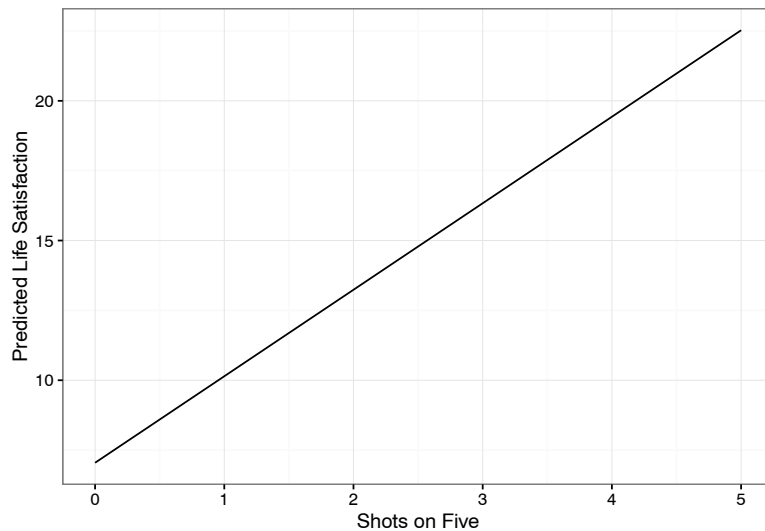
# Compute y-hat values for the fixed-effects (average) model
# Use predict() with re.form=NA argument
plotData$yhat = predict(lmer.4, newdata = plotData, re.form = NA)

# Examine data
plotData
```

```
## Shots_on_five Coach_Experience yhat
## 1 0.0 1.967 7.046669
## 2 0.1 1.967 7.356251
## 3 0.2 1.967 7.665833
## 4 0.3 1.967 7.975416
## 5 0.4 1.967 8.284998
## 6 0.5 1.967 8.594580
## 7 0.6 1.967 8.904163
## 8 0.7 1.967 9.213745
## 9 0.8 1.967 9.523327
## 10 0.9 1.967 9.832910
## 11 1.0 1.967 10.142492
## 12 1.1 1.967 10.452074
## 13 1.2 1.967 10.761657
## 14 1.3 1.967 11.071239
```

## 15	1.4	1.967 11.380821
## 16	1.5	1.967 11.690404
## 17	1.6	1.967 11.999986
## 18	1.7	1.967 12.309568
## 19	1.8	1.967 12.619151
## 20	1.9	1.967 12.928733
## 21	2.0	1.967 13.238315
## 22	2.1	1.967 13.547898
## 23	2.2	1.967 13.857480
## 24	2.3	1.967 14.167062
## 25	2.4	1.967 14.476645
## 26	2.5	1.967 14.786227
## 27	2.6	1.967 15.095809
## 28	2.7	1.967 15.405392
## 29	2.8	1.967 15.714974
## 30	2.9	1.967 16.024556
## 31	3.0	1.967 16.334139
## 32	3.1	1.967 16.643721
## 33	3.2	1.967 16.953303
## 34	3.3	1.967 17.262885
## 35	3.4	1.967 17.572468
## 36	3.5	1.967 17.882050
## 37	3.6	1.967 18.191632
## 38	3.7	1.967 18.501215
## 39	3.8	1.967 18.810797
## 40	3.9	1.967 19.120379
## 41	4.0	1.967 19.429962
## 42	4.1	1.967 19.739544
## 43	4.2	1.967 20.049126
## 44	4.3	1.967 20.358709
## 45	4.4	1.967 20.668291
## 46	4.5	1.967 20.977873
## 47	4.6	1.967 21.287456
## 48	4.7	1.967 21.597038
## 49	4.8	1.967 21.906620
## 50	4.9	1.967 22.216203
## 51	5.0	1.967 22.525785

```
# Plot; coerce coaching experience into a factor (or do it before plotting)
ggplot(data = plotData, aes(x = Shots_on_five, y = yhat)) +
  geom_line() +
  theme_bw() +
  xlab("Shots on Five") +
  ylab("Predicted Life Satisfaction")
```



Plotting the Fixed and Random Effects

The process for plotting these is similar to the previous plot. However, when we set up the initial data frame we use `expand.grid()` to include the level-2 grouping variable (`Team_ID`) and the predictors.

““

The frame slot produces a matrix of the outcome, predictor(s), and level-2 ID for each student. This is what we need in Step 1 of the plotting process.

```
plotData = expand.grid(
  Shots_on_five = seq(from = 0, to = 5, by = 0.1),
  Coach_Experience = 1.967,
  Team_ID = unique(nba$Team_ID)
)

# Compute y-hat values for the fixed-effects (average) model
# Use predict() with re.form=NA argument
plotData$yhat = predict(lmer.4, newdata = plotData)

# Plot
ggplot(data = plotData, aes(x = Shots_on_five, y = yhat, group = Team_ID)) +
  geom_line() +
  theme_bw() +
  xlab("Shots on Five") +
  ylab("Predicted Life Satisfaction")
```

