

# Multilevel Models: Longitudinal Data

Andrew Zieffler

March 29, 2016

## Preparation

```
# Read in the data
mplsWide = read.csv(file = "/Users/andrewz/Documents/EPsy-8252/data/minneapolis.csv")
print(head(mplsWide), row.names = FALSE)
```

```
## studentID read.5 read.6 read.7 read.8 atRisk female minority ell sped
##      1      172    185    179    194      1      1          1  0    0
##      2      200    210    209     NA      1      1          1  0    0
##      3      191    199    203    215      1      0          1  0    0
##      4      200    195    194     NA      1      1          1  0    0
##      5      207    213    212    213      1      1          1  0    0
##      6      191    189    206    195      1      0          1  0    0
##      att
##    0.94
##    0.91
##    0.97
##    0.88
##    0.85
##    0.90
```

```
# Load libraries
library(AICcmodavg)
library(dplyr)
library(ggplot2)
library(lme4)
library(sm)
library(tidyr)
```

## Wide Data versus Long Data

The `mplsWide` data are referred to as wide data. Wide data has the outcome variable (or predictors, or both) in multiple columns. For longitudinal data, it is convenient to enter the data into a spreadsheet in the wide format. Unfortunately, the wide format is not so good for analysis. Functions like `lm()` or `lmer()` require data in the long format. Long formatted data has only one column for the outcome. Additionally, if there are predictors, each is in a single column as well.

There are several functions to convert data back and forth between these two formats. We will use the `gather()` function from the `tidyr` package. `gather()` takes multiple columns, and gathers them into key-value pairs. We can also use the same type of piping we use with `dplyr`. We want to have key-value pairs of time points and reading scores. You can read more about using `tidyr` [here](#) and [here](#).

```

# Create long data from wide data (need tidyr package loaded)
mpls = mplsWide %>% gather(
  key = time, # Name of the variable that delineates the time points
  value = read, # Name of the outcome
  c(read.5, read.6, read.7, read.8) # The old variables you will use as values
)

# Arrange by ID number
mpls = mpls %>% arrange(studentID)
head(mpls, 12)

```

```

##      studentID atRisk female minority ell sped att time read
## 1           1      1      1          1  0   0 0.94 read.5 172
## 2           1      1      1          1  0   0 0.94 read.6 185
## 3           1      1      1          1  0   0 0.94 read.7 179
## 4           1      1      1          1  0   0 0.94 read.8 194
## 5           2      1      1          1  0   0 0.91 read.5 200
## 6           2      1      1          1  0   0 0.91 read.6 210
## 7           2      1      1          1  0   0 0.91 read.7 209
## 8           2      1      1          1  0   0 0.91 read.8  NA
## 9           3      1      0          1  0   0 0.97 read.5 191
## 10          3      1      0          1  0   0 0.97 read.6 199
## 11          3      1      0          1  0   0 0.97 read.7 203
## 12          3      1      0          1  0   0 0.97 read.8 215

```

The last part in our preparation is to create a numeric predictor for grade (note that currently the time predictor is a character vector). It is good practice to keep the character vector (or, better yet, a factor) and a numeric version of the predictor since some functions prefer one to the other. Here we use `mutate()` to create a numeric predictor called `grade`. Before we use `as.integer()` to turn the characters to numbers, however, we first turn `time` into a factor.

```

# Turn time into a factor
mpls$time = as.factor(mpls$time)

# Use as.integer() to turn factor into numbers
mpls = mpls %>% mutate(grade = as.integer(time))
head(mpls)

```

```

##      studentID atRisk female minority ell sped att time read grade
## 1           1      1      1          1  0   0 0.94 read.5 172     1
## 2           1      1      1          1  0   0 0.94 read.6 185     2
## 3           1      1      1          1  0   0 0.94 read.7 179     3
## 4           1      1      1          1  0   0 0.94 read.8 194     4
## 5           2      1      1          1  0   0 0.91 read.5 200     1
## 6           2      1      1          1  0   0 0.91 read.6 210     2

```

Note that the way `as.integer()` works is by using the numeric information in the factor level. The first level in the time factor (after we converted it to a factor) is `read.5`. This was converted to a 1 when we used `as.integer()`; `read.6` was converted to a 2; and so on. Because grade makes sense as a number, we will add four to each value so that `read.5` will be 5; `read.6` will be 6, etc. in the new predictor.

```
# Use as.integer() to turn factor into numbers
mpls = mpls %>% mutate(grade = as.integer(time) + 4)
head(mpls)
```

```
##   studentID atRisk female minority ell sped att   time read grade
## 1         1      1      1         1  0    0 0.94 read.5 172    5
## 2         1      1      1         1  0    0 0.94 read.6 185    6
## 3         1      1      1         1  0    0 0.94 read.7 179    7
## 4         1      1      1         1  0    0 0.94 read.8 194    8
## 5         2      1      1         1  0    0 0.91 read.5 200    5
## 6         2      1      1         1  0    0 0.91 read.6 210    6
```

## Exploratory Analysis

In longitudinal data analysis we are often interested in how the outcome changes over time. Here we want to know if reading scores are changing over time. When we examine this, we typically look at the mean score across time and the variation in scores over time.

```
mpls %>%
  group_by(time) %>%
  summarize(M = mean(read), SD = sd(read))
```

```
## Source: local data frame [4 x 3]
##
##   time      M      SD
##   (fctr)   (dbl) (dbl)
## 1 read.5 205.1364 19.99356
## 2 read.6 211.4545 20.06116
## 3 read.7 215.6818 19.44562
## 4 read.8    NA      NA
```

## Missingness

Since there are some students that do not have a reading score in the 8th grade, the computation of the mean and standard deviation for these grades defaulted to NA. It is typical to report both the count and percentage of missingness at each time point.

```
mpls %>%
  group_by(time) %>%
  summarize(
    Miss = sum(is.na(read)),
    Miss_Percent = sum(is.na(read))/length(read)
  )
```

```
## Source: local data frame [4 x 3]
##
##   time Miss Miss_Percent
##   (fctr) (int)      (dbl)
## 1 read.5    0    0.0000000
```

```
## 2 read.6      0      0.0000000
## 3 read.7      0      0.0000000
## 4 read.8      8      0.3636364
```

This pattern of later time points show more missing data is quite common in longitudinal data.

We will also compute the means and standard deviations for those students that we have data for. Below we use `filter()` to omit any NA values in the `read` variable before we do any computations.

```
mp1s %>%
  filter(!is.na(read)) %>%
  group_by(time) %>%
  summarize(M = mean(read, na.rm = TRUE), SD = sd(read))
```

```
## Source: local data frame [4 x 3]
##
##      time      M      SD
##   (fctr)   (dbl) (dbl)
## 1 read.5 205.1364 19.99356
## 2 read.6 211.4545 20.06116
## 3 read.7 215.6818 19.44562
## 4 read.8 218.0000 19.37881
```

The pattern in the sample data suggests that the mean reading score increases over grade level. The variation in reading scores seems fairly constant over grade levels (equal variances?). We need to be careful about the inferences coming from the last time point. The summary measures are only based on student who had data (recall 36% were missing data at grade 8). This means the estimates are likely biased.

## Correlation of Reading Scores Over Time

It is probably easiest to compute the correlation between the repeated measures using the wide formatted data. Similar to the computation of the means and standard deviations the `cor()` function will produce NAs if there is any missing data. In the function, we specify how to treat missing values. Using pairwise complete observations, the correlation between reading scores at grade 5 and grade 8, is computed only using students that have data at both time points. Because of this the correlations computed between different sets of reading scores may be based on different students. Here, `mp1sWide[2:5]` indicates that the correlation matrix should be based on columns 2 through 5 (the variables `read.5`, `read.6`, `read.7`, `read.8`).

```
cor(mp1sWide[2:5], use = "pairwise.complete.obs")
```

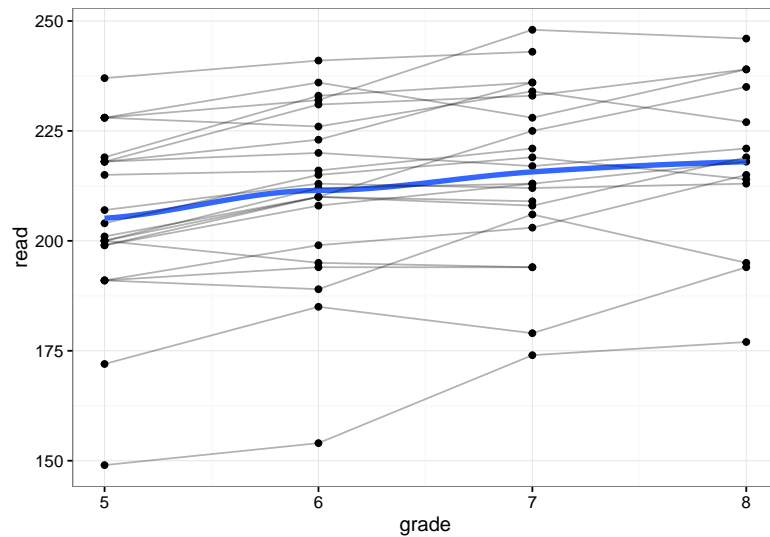
```
##      read.5  read.6  read.7  read.8
## read.5 1.0000000 0.9645804 0.9140704 0.8825860
## read.6 0.9645804 1.0000000 0.9256661 0.9287729
## read.7 0.9140704 0.9256661 1.0000000 0.9227732
## read.8 0.8825860 0.9287729 0.9227732 1.0000000
```

The correlations show that the reading scores are correlated. They also suggest that reading scores that are further apart in time are less correlated (in statistical terms, a pattern of decay). This pattern in the correlations is also quite common in longitudinal data.

## Spaghetti Plots

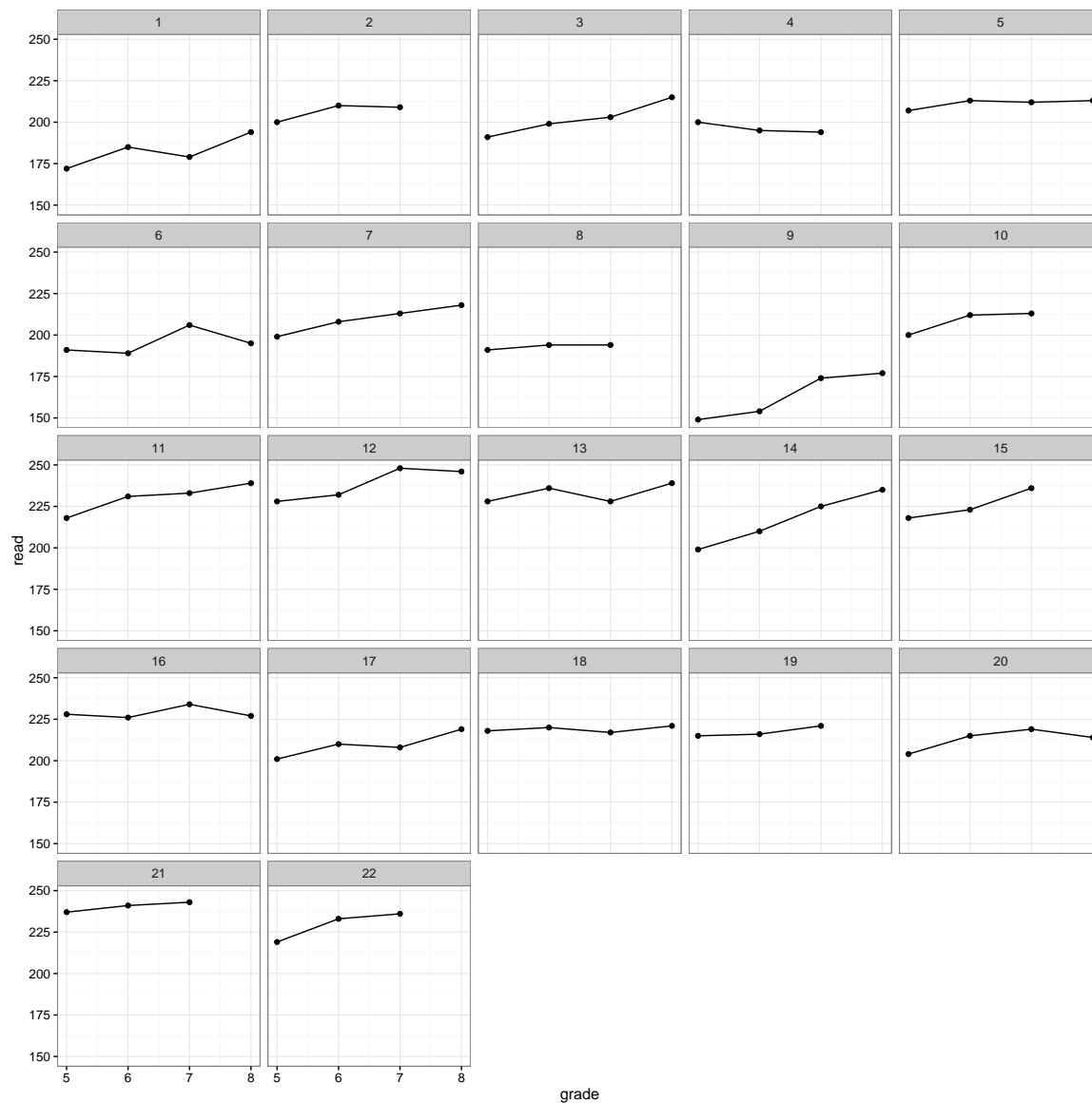
A spaghetti plot is simply a plot showing the trend in the outcome over time for each person. In our case, we will be plotting the reading scores for each student. Here we are looking for the functional form for the relationship between grade level and reading score (e.g., linear, quadratic). We also add the loess smoother to better examine the functional form.

```
ggplot(data = mpls, aes(x = grade, y = read, group = studentID)) +  
  geom_point() +  
  geom_smooth(aes(group = 1), se = FALSE, lwd = 1.5) +  
  geom_line(alpha = 0.3) +  
  theme_bw()
```



We can also facet by student ID to show each student's pattern separately.

```
ggplot(data = mpls, aes(x = grade, y = read, group = studentID)) +  
  geom_point() +  
  geom_line() +  
  theme_bw() +  
  facet_wrap(~studentID)
```



are three features these plots illustrate:

1. Most patterns are approximately linear (but not always) and the reading scores tend to increase over time.
2. Some OLS trajectories fit very well (e.g., Subject #7).
3. Other OLS trajectories show more scatter (e.g., Subject #13)

However, on average, it does seem like we could model the change in reading scores over time using a linear function.

There

## Unconditional Level-1 Model

Before we fit a linear function of time, we will first fit the unconditional level-1 model. The multilevel model for this is:

$$\text{Level-1 Model : } \text{Read}_i = \beta_0^* + \epsilon_i$$

$$\text{Level-2 Model : } \beta_0^* = \beta_{00} + \eta_{0j}$$

where  $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$  and  $\eta_{0j} \sim \mathcal{N}(0, \sigma_0^2)$ . In these models, the  $i$  subscript represents a particular time point, and the  $j$  subscript represents a particular person. The composite model is:

$$\text{Read}_i = \beta_{00} + \eta_{0j} + \epsilon_i$$

Initially we fit this model using REML estimation to get unbiased estimates of the error variances.

```
lmer.0 = lmer(read ~ 1 + (1 | studentID), data = mpls)
summary(lmer.0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: read ~ 1 + (1 | studentID)
## Data: mpls
##
## REML criterion at convergence: 621.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.21398 -0.48096  0.02441  0.52368  2.21040
##
## Random effects:
##  Groups      Name      Variance Std.Dev.
## studentID (Intercept) 335.38   18.313
## Residual              66.21    8.137
## Number of obs: 80, groups: studentID, 22
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  212.244      4.011   52.92
```

The fitted unconditional level-1 model suggests that the mean reading score across students and grade levels is 212.2. Computing the intraclass correlation coefficient (ICC),

$$\begin{aligned}\rho &= \frac{\sigma_0^2}{\sigma_0^2 + \sigma_\epsilon^2} \\ &= \frac{335.38}{335.38 + 66.21} \\ &= 0.83\end{aligned}$$

,

we find that an estimated 83.5% of the total variation in reading scores is attributable to differences between students, and 16.5% of the total variation in reading scores is attributable to differences between time points (within students).

## Building the Unconditional Growth Model (Level-1 Model)

Having partitioned the total variation into within-students and between-students, we can now ask: What role does grade-level (time) play? Based on the exploratory analysis, a linear model relating grade-level and reading scores seemed appropriate. Below we show the composite model that maintains a random-effect of intercept, but also includes a linear predictor of grade-level.

$$\text{Read}_i = \beta_{00} + \beta_{10}(\text{Grade}_i) + \eta_{0j} + \epsilon_i$$

We also consider a quadratic effect of grade level for completeness,

$$\text{Read}_i = \beta_{00} + \beta_{10}(\text{Grade}_i) + \beta_{11}(\text{Grade}_i^2) + \eta_{0j} + \epsilon_i$$

In both models,  $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$  and  $\eta_{0j} \sim \mathcal{N}(0, \sigma_0^2)$ . Since the error structure and random effects are the same as that for the unconditional means model, we fit this model and re-fit the unconditional means model using ML estimation. We then use AICc to select. To double-check the functional form, we also fit a quadratic model for the fixed-effects.

```
##
## Model selection based on AICc:
##
##      K   AICc Delta_AICc AICcWt Cum.Wt      LL
## LMER1 4 587.82      0.00   0.7   0.7 -289.64
## LMER2 5 589.48      1.66   0.3   1.0 -289.34
## LMER0 3 632.82     45.00   0.0   1.0 -313.25
```

Here we adopt the model with only the linear effect of grade level.

## Graphically Examining the Level-1 Residuals: An Alternative Approach to Choosing the Level-1 Structure

Consider Student #1's fitted models for (1) the unconditional means model, (2) the unconditional growth model (linear), and (3) the unconditional growth model (quadratic). To obtain each of these fitted models, substitute in the values of the fixed effects, and of Student #1's random effect. (Base these on models fitted with ML.)

**Model 1 :**  $\hat{\text{Read}}_i = 183.9 + \hat{\epsilon}_i$

**Model 2 :**  $\hat{\text{Read}}_i = 150.8 + 5.0(\text{Grade}_i) + \hat{\epsilon}_i$

**Model 3 :**  $\hat{\text{Read}}_i = 131.3 + 11.3(\text{Grade}_i) - 0.5(\text{Grade}_i^2) + \hat{\epsilon}_i$



Now consider the level-1 residuals for Student #1 for each of these models.

```
# Unconditional means model level-1 residuals
resid.0 = data.frame(
  studentID = lmer.0.ml@frame$studentID,
  residual = resid(lmer.0.ml),
  model = "Model 0"
)

# Unconditional growth model (linear) level-1 residuals
resid.1 = data.frame(
  studentID = lmer.1.ml@frame$studentID,
  residual = resid(lmer.1.ml),
  model = "Model 1"
)

# Unconditional growth model (quadratic) level-1 residuals
resid.2 = data.frame(
  studentID = lmer.2.ml@frame$studentID,
  residual = resid(lmer.2.ml),
  model = "Model 2"
)

# Stack the three sets of residuals
all.resid = rbind(resid.0, resid.1, resid.2)

# Only Student 1
res.1 = all.resid %>% filter(studentID == 1)
res.1
```

```
##   studentID  residual  model
## 1         1 -11.965579 Model 0
## 2         1  1.034421 Model 0
## 3         1 -4.965579 Model 0
## 4         1 10.034421 Model 0
## 5         1 -3.676758 Model 1
## 6         1  4.338689 Model 1
## 7         1 -6.645863 Model 1
## 8         1  3.369584 Model 1
## 9         1 -3.315706 Model 2
## 10        1  3.810648 Model 2
## 11        1 -7.083428 Model 2
## 12        1  4.002065 Model 2
```

Student #1 has four level-1 residuals (one for each grade level) in each of the models. If the linear predictor of grade is necessary, then the residuals for the unconditional growth model (linear) should be smaller than those for the unconditional means model. Similarly, if the quadratic predictor of grade is necessary, then the residuals for the unconditional growth model (quadratic) should be smaller than those for the unconditional growth model (linear).

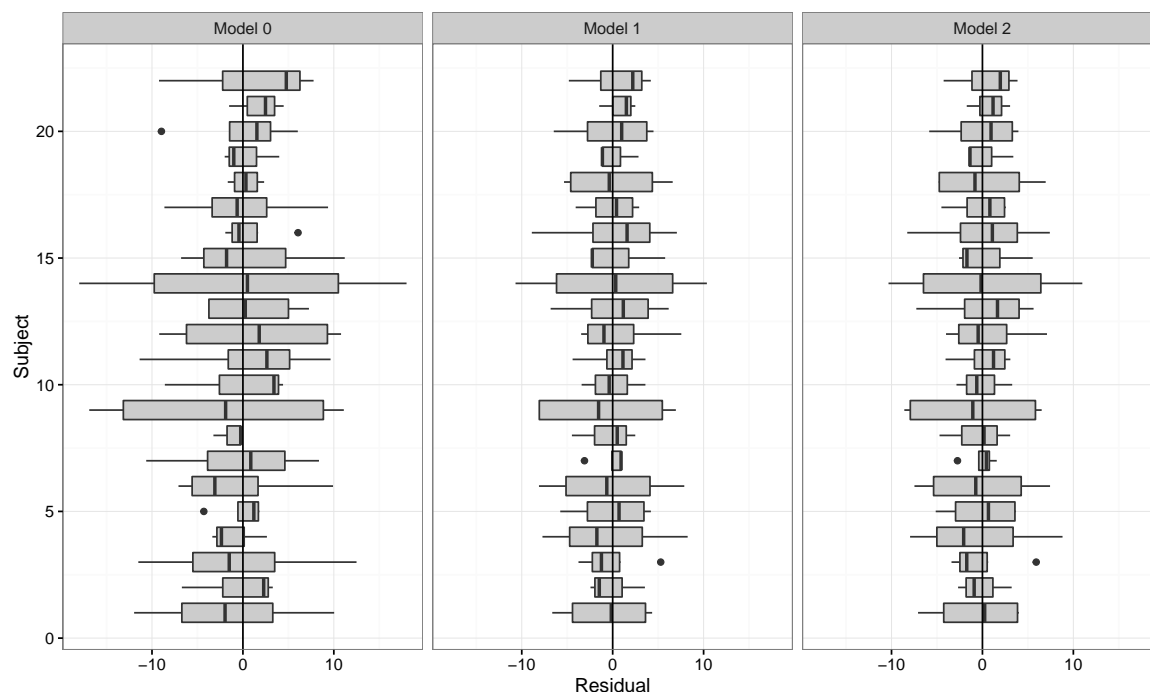
Below we compute the standard deviation for each model's residuals for Student #1.

```
res.1 %>% group_by(model) %>% summarize(SD = sd(residual))
```

```
## Source: local data frame [3 x 2]
##
##   model      SD
##   (fctr)    (dbl)
## 1 Model 0 9.327379
## 2 Model 1 5.358970
## 3 Model 2 5.478261
```

For Student #1, the residuals are definitely smaller in the unconditional growth model (linear) than in the unconditional means model. There doesn't seem to be any improvement in the residuals when we add the quadratic predictor. In practice, we would look at the residuals for ALL the students, not just Student #1. Rather than computing the standard deviation and comparing those for all the students, we just plot the residuals for each student, faceting by model. Then we can visually examine the variation.

```
## Plot the residuals by subject ID to
ggplot(data = all.resid, aes(x = studentID, y = residual, group = studentID)) +
  theme_bw() +
  geom_boxplot(fill = "grey80") +
  geom_hline(yintercept = 0) +
  facet_wrap(~ model) +
  xlab("Subject") +
  ylab("Residual") +
  coord_flip()
```



Adding the grade-level (time) predictor reduces the level-1 residual variation. Again, the quadratic predictor does not seem to really reduce the level-1 residual variation any further. This would suggest that a linear, but not a quadratic predictor should be included in the level-1 model.

This same information is quantified in the residual variance term. Consider the residual variance for each of the three models:

$$\text{Model 0 : } \hat{\sigma}_{\epsilon}^2 = 66.2$$

$$\text{Model 1 : } \hat{\sigma}_{\epsilon}^2 = 29.2$$

$$\text{Model 2 : } \hat{\sigma}_{\epsilon}^2 = 28.9$$

There is a big decrease in level-1 error variance from the unconditional means model to the unconditional growth model (linear). The residual variance from the unconditional growth model (linear) and the unconditional growth model (quadratic) are quite similar.

## Choosing Random Effects Structure

Looking back at the spaghetti plot, it seems that not only are students reading scores different at grade 0 (evidence for random intercepts), but also that their growth trajectories are different (evidence for random slopes). We will examine two different random effects structures: (1) random intercepts only; and (2) random intercepts and random slopes. Because we are keeping the fixed-effects the same, it is better to evaluate these models if we have estimated using REML.

```
# Fit models using REML
lmer.1.int = lmer(read ~ 1 + grade + (1 | studentID), data = mpls)
lmer.1.both = lmer(read ~ 1 + grade + (1 + grade | studentID), data = mpls)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.1.int, lmer.1.both),
  modnames = c("Random Intercepts", "Random Intercepts + Slopes")
)
```

```
## Warning in aictab.AIClmerMod(cand.set = list(lmer.1.int, lmer.1.both), modnames = c("Random Intercep
## Model selection for fixed effects is only appropriate with ML estimation:
## REML (default) should only be used to select random effects for a constant set of fixed effects
```

```
myAIC
```

```
##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt Res.LL
## Random Intercepts + Slopes 6 578.98      0.00  0.85  0.85 -282.91
## Random Intercepts          4 582.45      3.47  0.15  1.00 -286.96
```

The evidence supports adopting a random effects structure that includes random intercepts and random slopes.

## Interpretation of the Model

```
summary(lmer.1.both)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: read ~ 1 + grade + (1 + grade | studentID)
## Data: mpls
##
## REML criterion at convergence: 565.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.59244 -0.52818 -0.04596  0.51542  2.02337
##
## Random effects:
## Groups      Name                Variance Std.Dev. Corr
## studentID (Intercept) 783.658   27.994
##              grade         7.547    2.747  -0.75
## Residual              18.300    4.278
## Number of obs: 80, groups: studentID, 22
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 181.3487    6.7111  27.022
## grade        4.8796     0.7598   6.422
##
## Correlation of Fixed Effects:
##      (Intr)
## grade -0.799
```

- The mean reading score at grade level 0 is predicted to be 181.3.
- Each grade-level difference is associated with a 4.9-point difference, on average, in reading score.
- There is still unexplained within-student residual variance (error variance is not 0). This means we can include time-varying predictors in the model (if we have any).
- There is between-student residual variance in intercepts (reading scores at grade 0) since the RE variance component is not 0.
- There is between-student residual variance in rate of change (slopes are different) since the RE variance component is not 0.
- Estimated residual correlation between intercepts and rate-of-change is negative (students who are lower than average at grade 0 tend to grow faster than average, and vice versa)

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

```
# Get variance-covariance matrix of the random effects
VarCorr(lmer.1.both)$studentID
```

```
##           (Intercept)      grade
## (Intercept)  783.65770 -57.321821
## grade       -57.32182   7.547034
## attr(,"stddev")
## (Intercept)      grade
##  27.993887    2.747187
## attr(,"correlation")
##           (Intercept)      grade
## (Intercept)  1.0000000 -0.7453644
## grade       -0.7453644  1.0000000
```

- $\sigma_0^2 = 783.7$
- $\sigma_1^2 = 7.5$
- $\sigma_{01} = -57.3$

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

## Centering a Level-1 Predictor

One thing we can do to make the intercept interpretable is to center the grade-level predictor. To do this, we will subtract five from the `grade` variable

```
mpls$c.grade = mpls$grade - 5
head(mpls)
```

```
## studentID atRisk female minority ell sped att time read grade c.grade
## 1         1      1      1         1  0    0 0.94 read.5 172    5      0
## 2         1      1      1         1  0    0 0.94 read.6 185    6      1
## 3         1      1      1         1  0    0 0.94 read.7 179    7      2
## 4         1      1      1         1  0    0 0.94 read.8 194    8      3
## 5         2      1      1         1  0    0 0.91 read.5 200    5      0
## 6         2      1      1         1  0    0 0.91 read.6 210    6      1
```

Now we can re-fit the model using the centered predictor.

```
lmer.1.c = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls)
summary(lmer.1.c)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: read ~ 1 + c.grade + (1 + c.grade | studentID)
## Data: mpls
##
## REML criterion at convergence: 565.8
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.59244 -0.52818 -0.04596  0.51542  2.02337
##
## Random effects:
##   Groups      Name      Variance Std.Dev. Corr
## studentID (Intercept) 399.115  19.978
##           c.grade      7.547   2.747  -0.36
## Residual              18.300   4.278
## Number of obs: 80, groups:  studentID, 22
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 205.7469    4.3306   47.51
## c.grade      4.8796    0.7598    6.42
##
## Correlation of Fixed Effects:
##          (Intr)
## c.grade -0.360
```

- The mean reading score at grade 5 is predicted to be 205.7 (no more extrapolation).
- Each grade-level difference is associated with a 4.9-point difference, on average, in reading score (doesn't change since we didn't change the scale).

When we center predictors, the RE for the intercept and the correlation between the REs (that involve intercept) will change.

- There is still unexplained within-student residual variance (it is the same value as before).
- The intercept now pertains to students' grade 5 reading scores ... There is between-student residual variance in initial status (reading scores at grade 5)
- There is between-student residual variance in rate-of-change (slopes are different)
- Estimated residual correlation between initial status and rate-of-change is negative (students who are lower than average starting out at grade 5 tend to grow faster than average, and vice versa)

Because the correlation between the RE changed, we need to re-compute the covariance estimate.

```
VarCorr(lmer.1.c)$studentID
```

```
##              (Intercept)      c.grade
## (Intercept)  399.11545 -19.586674
## c.grade      -19.58667   7.547039
## attr(,"stddev")
## (Intercept)      c.grade
##  19.977874    2.747187
## attr(,"correlation")
##              (Intercept)      c.grade
## (Intercept)  1.0000000 -0.3568808
## c.grade      -0.3568808  1.0000000
```

- $\sigma_0^2 = 399.1$
- $\sigma_1^2 = 7.5$
- $\sigma_{01} = -19.6$

## 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 student's model (fixed effects + random effects). Typically we always plot the fixed effects portion of the model, since that part of the model informs us of the relationship between grade level (time) and the reading scores (outcome).

$$\widehat{\text{Read}}_i = 205.7 + 4.9(\text{c.grade}_i)$$

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 = data.frame(
  c.grade = 0:3
)

# The predict() function WILL NOT WORK to predict y-hats from the
# fixed-effects model. Use fixef() function instead.
fixef(lmer.1.c)

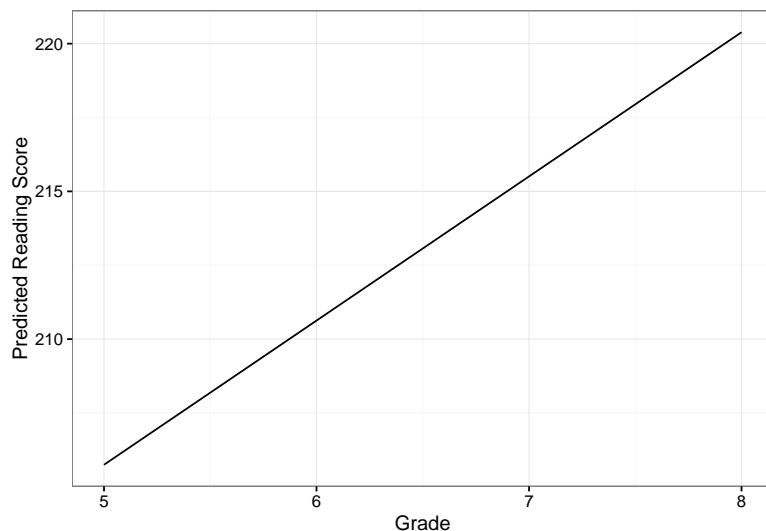
## (Intercept)      c.grade
## 205.746912    4.879632

# Compute y-hat values for the fixed-effects (average) model
# This is essentially: plotData$yhat = 205.745119 + 4.882322 * plotData$c.grade
plotData$yhat = fixef(lmer.1.c)["(Intercept)"] + fixef(lmer.1.c)["c.grade"] * plotData$c.grade

# Examine data
plotData

##   c.grade   yhat
## 1      0 205.7469
## 2      1 210.6265
## 3      2 215.5062
## 4      3 220.3858

# Plot
ggplot(data = plotData, aes(x = c.grade, y = yhat)) +
  geom_line() +
  theme_bw() +
  scale_x_continuous(name = "Grade", breaks = 0:3, labels = c("5", "6", "7", "8")) +
  ylab("Predicted Reading Score")
```



## 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 (`studentID`) and the predictors. An added bonus is that we can use `predict()` to get the y-hat values.

`lmer()` objects are a class of objects called S4 objects. They store output from the model in slots. To find out the names of the slots we use the `slotNames()` function.

```
# What associated output do we get with an lmer() object?
slotNames(lmer.1.c)
```

```
## [1] "resp"    "Gp"      "call"    "frame"   "flist"   "cnms"    "lower"
## [8] "theta"   "beta"    "u"       "devcomp" "pp"      "optinfo"
```

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.

```
head(lmer.1.c@frame)
```

```
##   read c.grade studentID
## 1  172      0          1
## 2  185      1          1
## 3  179      2          1
## 4  194      3          1
## 5  200      0          2
## 6  210      1          2
```

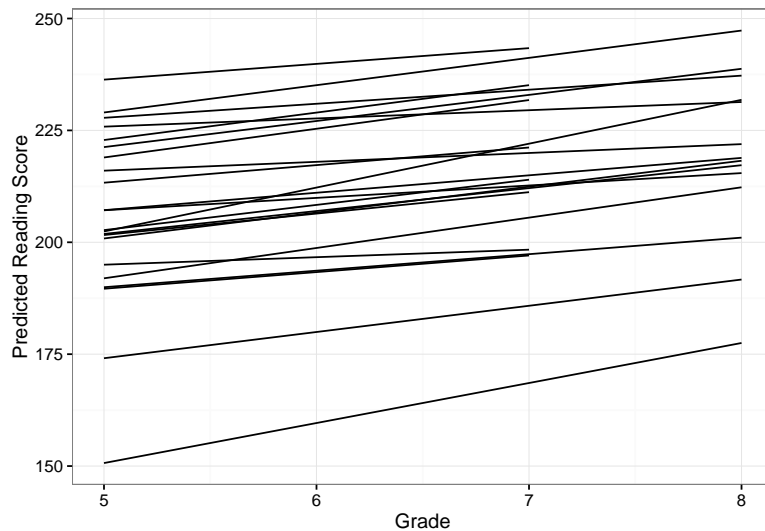
```
# Set up the data for plotting
plotData = lmer.1.c@frame

# Use predict() to add the y-hat values
plotData$yhat = predict(lmer.1.c, newdata = plotData)

# Plot
```



```
ggplot(data = plotData, aes(x = c.grade, y = yhat, group = studentID)) +
  geom_line() +
  theme_bw() +
  scale_x_continuous(name = "Grade", breaks = 0:3, labels = c("5", "6", "7", "8")) +
  ylab("Predicted Reading Score")
```



## Adding Level-2 Predictors

There are several different level-2 predictors we can consider:

- Whether the student is at risk (**atRisk**)
- Whether the student is female (**female**)
- Whether the student is a minority student (**minority**)
- Whether the student is an English Language Learner (**ell**)
- Whether the student is a special education student (**sped**)
- The student's attendance rate (**att**)

In thinking about these predictors, there are several ways to proceed through the model building process. Here, I will first examine each level-2 predictor independent of the other level-2 predictors. For each level-2 predictor, we will examine (1) the unconditional growth model; (2) the unconditional growth model with the level-2 predictor for intercept only; and (3) the unconditional growth model with the level-2 predictor for intercept and slope. Since the effects that are different are the fixed effects, we fit these models using ML.

### At-Risk

```
lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + atRisk + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + atRisk + atRisk:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
```

```

  modnames = c("Unconditional Growth", "atRisk - Intercept", "atRisk - Intercept + Slope")
)

myAIC

```

```

##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## atRisk - Intercept      7 579.46      0.00  0.72  0.72 -281.95
## atRisk - Intercept + Slope 8 581.79      2.33  0.23  0.95 -281.88
## Unconditional Growth      6 584.85      5.39  0.05  1.00 -285.85

```

## Female

```

lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + female + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + female + female:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
  modnames = c("Unconditional Growth", "female - Intercept", "female - Intercept + Slope")
)

myAIC

```

```

##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Unconditional Growth      6 584.85      0.00  0.70  0.70 -285.85
## female - Intercept      7 587.06      2.21  0.23  0.93 -285.75
## female - Intercept + Slope 8 589.53      4.68  0.07  1.00 -285.75

```

## Minority

```

lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + minority + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + minority + minority:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
  modnames = c("Unconditional Growth", "minority - Intercept", "minority - Intercept + Slope")
)

myAIC

```

```
##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## minority - Intercept      7 575.34      0.00  0.68  0.68 -279.89
## minority - Intercept + Slope 8 576.85      1.52  0.32  0.99 -279.41
## Unconditional Growth      6 584.85      9.51  0.01  1.00 -285.85
```

## English Language Learner

```
lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + ell + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + ell + ell:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
  modnames = c("Unconditional Growth", "ell - Intercept", "ell - Intercept + Slope")
)

myAIC
```

```
##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Unconditional Growth      6 584.85      0.00  0.63  0.63 -285.85
## ell - Intercept          7 587.06      2.21  0.21  0.83 -285.75
## ell - Intercept + Slope 8 587.50      2.65  0.17  1.00 -284.74
```

## Special Education

```
lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + sped + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + sped + sped:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
  modnames = c("Unconditional Growth", "sped - Intercept", "sped - Intercept + Slope")
)

myAIC
```

```
##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Unconditional Growth      6 584.85      0.00  0.46  0.46 -285.85
## sped - Intercept          7 585.43      0.58  0.35  0.81 -284.94
## sped - Intercept + Slope 8 586.62      1.77  0.19  1.00 -284.29
```

## Attendance

```
lmer.0 = lmer(read ~ 1 + c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.1 = lmer(read ~ 1 + c.grade + att + (1 + c.grade | studentID), data = mpls, REML = FALSE)
lmer.2 = lmer(read ~ 1 + c.grade + att + att:c.grade + (1 + c.grade | studentID), data = mpls, REML = FALSE)

# AICc selection
myAIC = aictab(
  cand.set = list(lmer.0, lmer.1, lmer.2),
  modnames = c("Unconditional Growth", "att - Intercept", "att - Intercept + Slope")
)

myAIC
```

```
##
## Model selection based on AICc:
##
##           K   AICc Delta_AICc AICcWt Cum.Wt      LL
## att - Intercept + Slope 8 582.04      0.00  0.54  0.54 -282.00
## att - Intercept       7 583.07      1.03  0.32  0.87 -283.76
## Unconditional Growth   6 584.85      2.81  0.13  1.00 -285.85
```

Based on the analyses, we should include:

- The level-2 predictor `atRisk` in the intercept model
- The level-2 predictor `minority` in the intercept model
- The level-2 predictor `att` in the intercept and slope models

Here we fit that model using REML.

```
lmer.final = lmer(read ~ 1 + c.grade + atRisk + minority + att +
  att:c.grade + (1 + c.grade | studentID), data = mpls)
summary(lmer.final)$coefficients
```

```
##           Estimate Std. Error   t value
## (Intercept) 346.57216 102.302142  3.387731
## c.grade     -30.21208  18.532798 -1.630195
## atRisk      -12.74014   7.896095 -1.613473
## minority    -22.35976   7.983893 -2.800609
## att         -126.16844 103.727061 -1.216350
## c.grade:att   36.69194  19.352192  1.896009
```

From here it is a little uncertain as to how we should proceed. The *t*-ratios for the at-risk, attendance, and attendance by grade interaction are all less than two. This suggests that maybe these terms could be dropped. Should we drop them all at once? Or one at a time? This is a good question, and there is no good answer.

Below I will proceed by dropping one term at a time, since we are taking an exploratory view of the analysis. Start by dropping the interaction term, since it is the highest order term.

```
lmer.final2 = lmer(read ~ 1 + c.grade + atRisk + minority + att +
  (1 + c.grade | studentID), data = mpls)
summary(lmer.final2)$coefficients
```

```
##           Estimate Std. Error   t value
## (Intercept) 314.578852 100.999517  3.1146570
## c.grade      4.891834   0.770989  6.3448822
## atRisk      -12.710176   7.895021 -1.6098976
## minority    -22.291750   7.982801 -2.7924723
## att         -92.783593 102.324370 -0.9067595
```

Now we will drop the attendance main effect.

```
lmer.final3 = lmer(read ~ 1 + c.grade + atRisk + minority +
  + (1 + c.grade | studentID), data = mpls)
summary(lmer.final3)$coefficients
```

```
##           Estimate Std. Error   t value
## (Intercept) 222.755346  5.1929257 42.895924
## c.grade      4.876549   0.7675561  6.353345
## atRisk     -10.486052   7.3422277 -1.428184
## minority    -19.100592   7.4351008 -2.568976
```

And, it looks as though the at-risk predictor can be dropped as well.

```
lmer.final4 = lmer(read ~ 1 + c.grade + minority +
  (1 + c.grade | studentID), data = mpls)
summary(lmer.final4)$coefficients
```

```
##           Estimate Std. Error   t value
## (Intercept) 220.459815  5.0393966 43.747264
## c.grade      4.875792   0.7667507  6.359031
## minority    -24.894426   6.4190613 -3.878203
```

## Assumption Checking

Start with the examination of the level-1 residuals.

```
‘ out = fortify(lmer.final4)
```

```
Error in $<-data.frame(*tmp*, "fitted", value = c(174.501800917911, : replacement has 80 rows, data has 88 ‘
```

Unfortunately, we get an error when we try to use `fortify()`. The error is telling us that some values cannot be computed for 8 of the rows. This is because the data has 8 rows where the reading score was missing (hard to compute a residual for those rows). The solution is to drop all rows with missing data after you are in the long format, and then re-fit the final model to the data where there are no missing values.

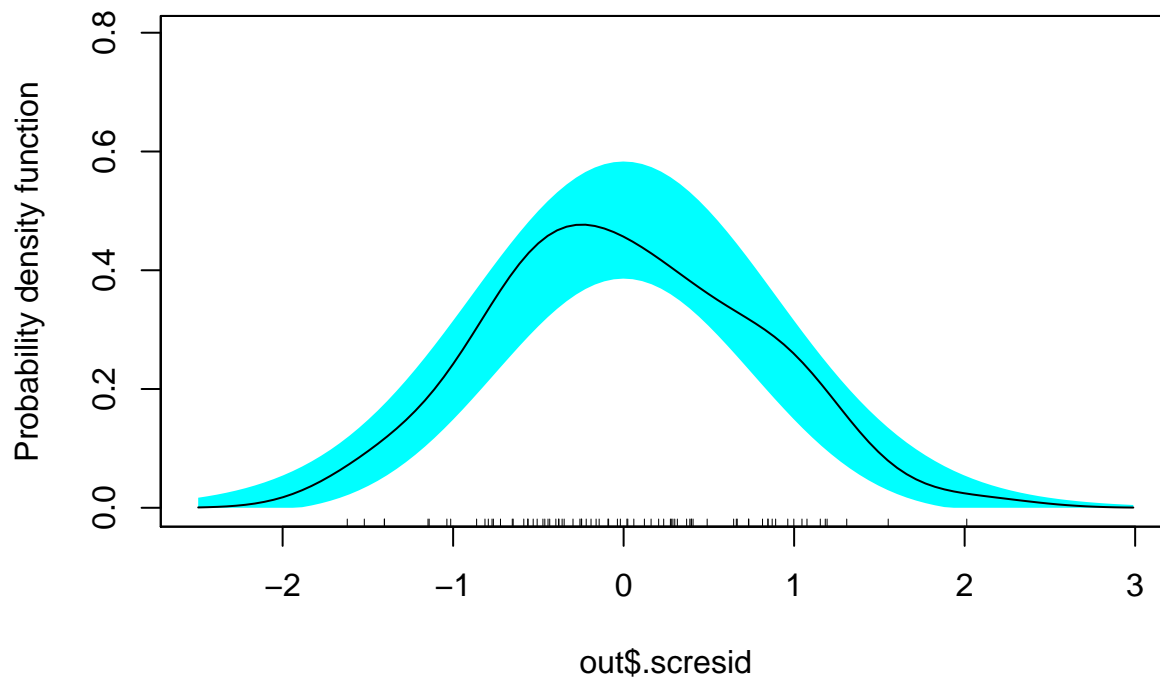
```
# Remove missing values
mpls2 = mpls %>% na.omit()

# Re-fit model
lmer.final5 = lmer(read ~ 1 + c.grade + minority + (1 + c.grade | studentID), data = mpls2)

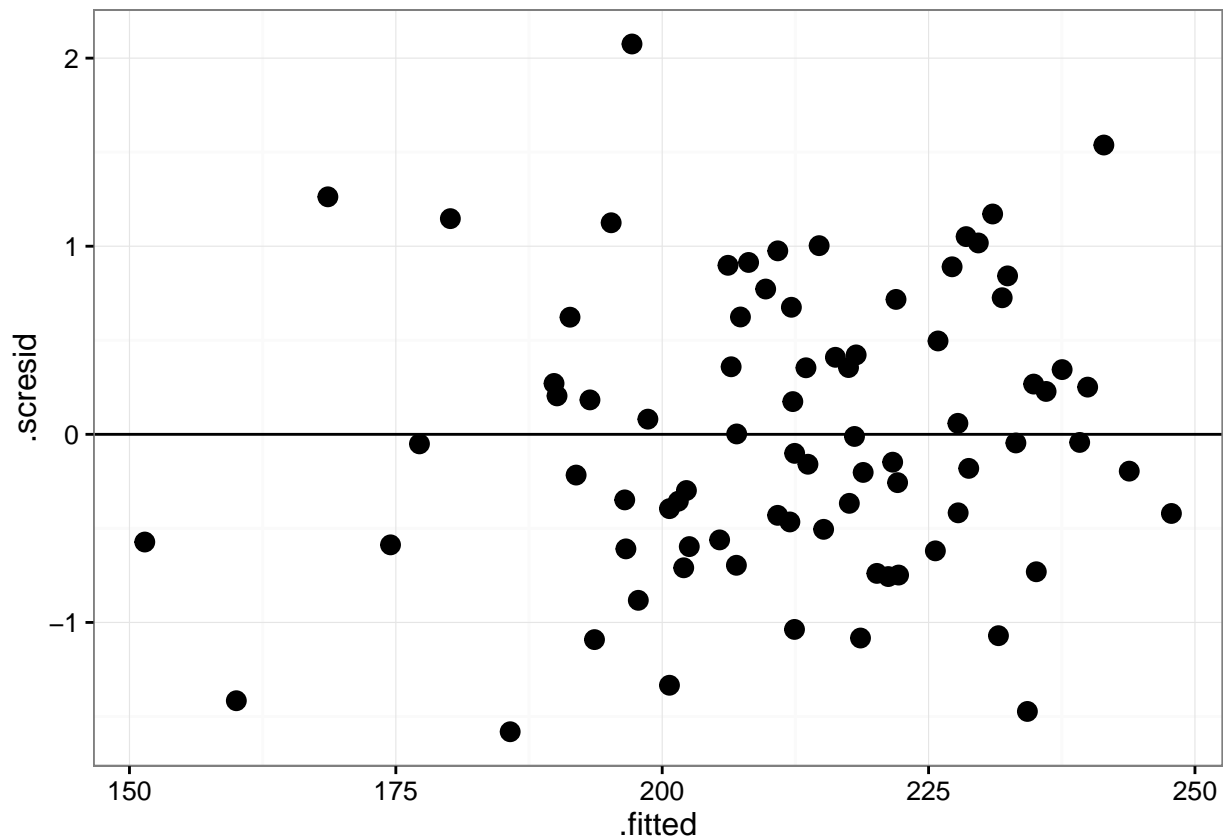
# Fortify
out = fortify(lmer.final5)
head(out)
```

```
## studentID atRisk female minority ell sped att time read grade c.grade
## 1 1 1 1 1 0 0 0.94 read.5 172 5 0
## 2 1 1 1 1 0 0 0.94 read.6 185 6 1
## 3 1 1 1 1 0 0 0.94 read.7 179 7 2
## 4 1 1 1 1 0 0 0.94 read.8 194 8 3
## 5 2 1 1 1 0 0 0.91 read.5 200 5 0
## 6 2 1 1 1 0 0 0.91 read.6 210 6 1
## .fitted .resid .screid
## 1 174.5018 -2.501801 -0.5875248
## 2 180.1172 4.882803 1.1466810
## 3 185.7326 -6.732594 -1.5810874
## 4 191.3480 2.652009 0.6227999
## 5 201.5137 -1.513737 -0.3554872
## 6 206.1736 3.826379 0.8985897
```

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



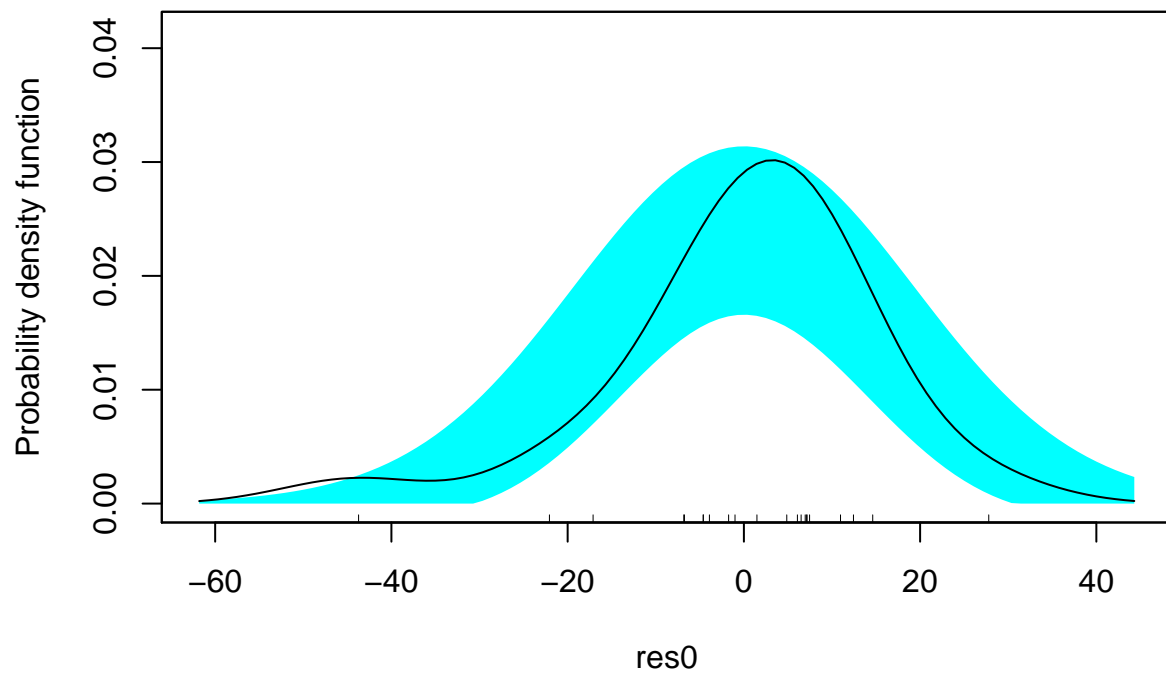
```
ggplot(data = out, aes(x = .fitted, y = .screid)) +
  geom_point(size = 3) +
  theme_bw() +
  geom_hline(yintercept = 0)
```



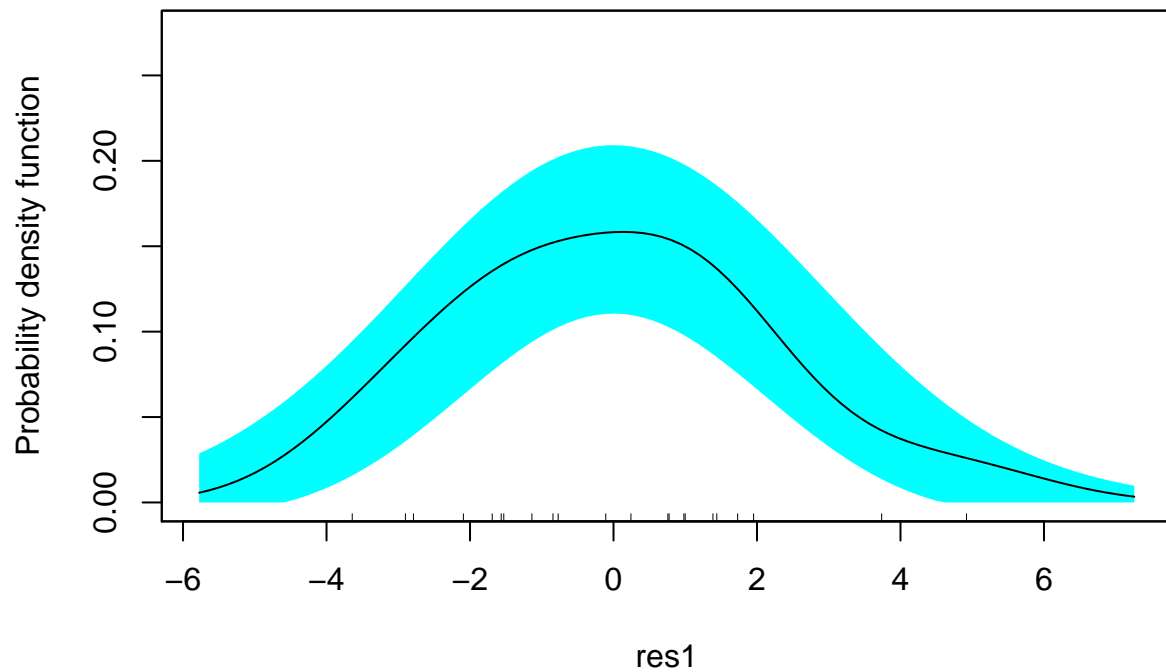
Now we can examine the level-2 residuals for the intercept and slope. In practice, typically we only examine the normality assumptions for the level-2 residuals

```
# Compute the residuals
res0 = ranef(lmer.final5)$studentID[ , 1]
res1 = ranef(lmer.final5)$studentID[ , 2]

# Density plot of the intercept residuals
sm.density(res0, model = "normal")
```



```
# Density plot of the slope residuals
sm.density(res1, model = "normal")
```



All the assumptions seem reasonably satisfied. The last thing to do would be to produce relevant model summaries and report a reasonable taxonomy of models. Here i would report:

- The unconditional means model
- The unconditional growth model
- The “final” model with the minority level-2 predictor