

ACRM 2021 Longitudinal Data Analysis Workshop

Keith Lohse, PhD, PStat, and Allan Kozlowski, PhD, BsC (PT)

Practical Session 1: Building Linear Mixed-Effects Models

This handout is designed to accompany the script you will be working with in the practical session. A copy of the script file, the data, set, and this handout can be found at: https://github.com/keithlohse/LMER_Clinical_Science. The R code is interspersed with explanations below. All R code is highlighted in grey and color coded to show different functions, arguments, and comments in the code.

First, you will need to open the five packages we will be using for this session using the library function:

```
# Loading the essential libraries.
library("ggplot2"); library("lme4"); library("car"); library("dplyr"); library("lmerTest");
```

If you haven't already installed these packages, you will need to use the `install.packages()` function first. This can take some time and will require an internet connection.

```
# If these packages are not installed already, run the following code:
install.packages("ggplot2"); install.packages("lme4"); install.packages("car"); install.packages("dplyr")
```

1.1 Data Cleaning and Quality Assurance

One of the first steps is to set the working directory. This is a file-pathway that directs R to the folder in which the various data and script files are stored. Make sure the "data_session1.csv" file is saved in that folder and then use the `read.csv()` function to read the data into R.

```
# We will read these data in from the web:
DATA <- read.csv("https://raw.githubusercontent.com/keithlohse/LMER_Clinical_Science/master/ACRM_2018/d
                stringsAsFactors = TRUE, na.strings=c("NA","NaN"," ",""))
# Use the head() function to check the structure of the data file.
head(DATA)
```

```
##   subID month time sex female bmi_z rasch_FIM rasch_FIM_MAR rasch_FIM_MNAR
## 1   s01     1 1.16  f      1 -0.44 12.56000      12.56000      12.56000
## 2   s01     2 2.58  f      1 -0.44 24.85447      24.85447      24.85447
## 3   s01     3 3.69  f      1 -0.44 40.27863           NA      40.27863
## 4   s01     4 4.92  f      1 -0.44 43.86893           NA      43.86893
## 5   s01     5 5.16  f      1 -0.44 47.66534           NA      47.66534
## 6   s01     6 6.63  f      1 -0.44 49.11310      49.11310           NA
##   rasch_FIM_LOCF inpatient_time los AIS_grade therapy_hours PT_hours OT_hours
## 1      12.56000           5wks  32      C5-8      14.6000  7.3000  7.4000
## 2      24.85447           5wks  32      C5-8      14.7000  7.3000  7.4000
## 3      40.27863           5wks  32      C5-8      14.6000  7.3000  7.3000
```

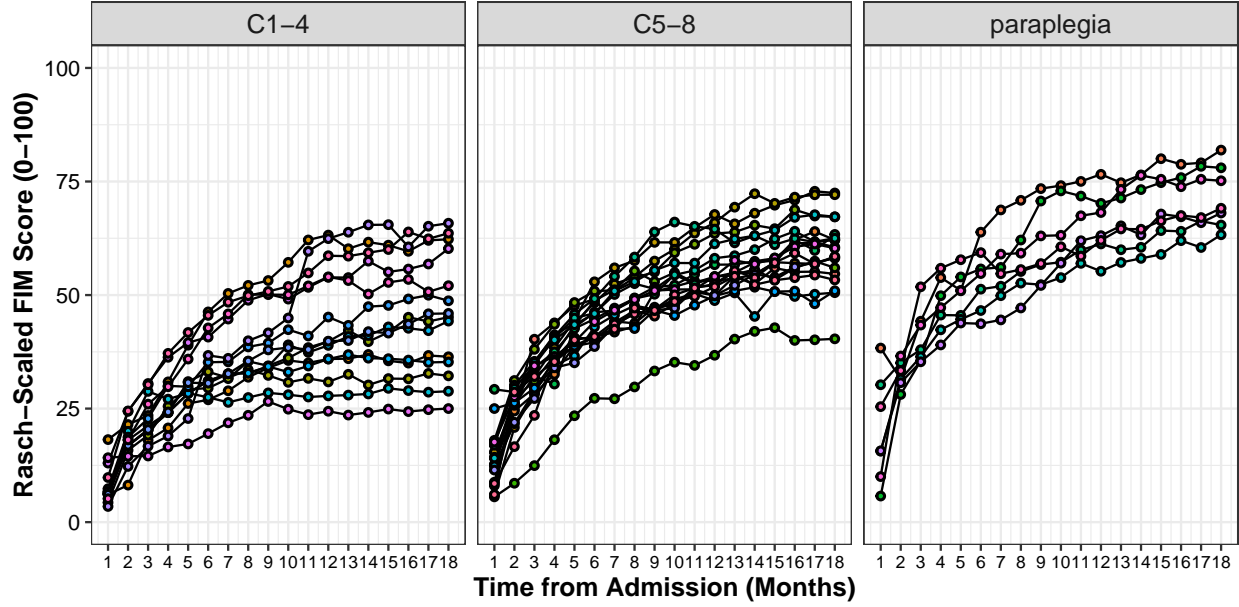
```
## 4      43.86893      5wks 32      C5-8      14.5000      7.2000      7.3000
## 5      47.66534      5wks 32      C5-8      8.3366      4.1683      4.2254
## 6      47.66534      5wks 32      C5-8      NA          NA          NA
##      SUM_PT_hours SUM_OT_hours AVE_PT_hours AVE_OT_hours
## 1      33.2683      33.6254      6.65366      6.72508
## 2      33.2683      33.6254      6.65366      6.72508
## 3      33.2683      33.6254      6.65366      6.72508
## 4      33.2683      33.6254      6.65366      6.72508
## 5      33.2683      33.6254      6.65366      6.72508
## 6      33.2683      33.6254      6.65366      6.72508
```

```
## Alternatively, if you want to download the data to a local folder, you will
## need to set the appropriate directory and then read in the data file, as
## illustrated by the code below:
## Note, if you already downloaded the data from the web, you can skip this step.
# getwd()
# setwd("C:/Users/Folder/SubFolder/SubSubFolder/")
# list.files()
## Make sure that the file data_session1.csv is saved in your working directory.
## Import the .csv file into R.
## We will save this file in the R environment as an object called "DATA".
# DATA<-read.csv("./data_session1.csv", header = TRUE, sep="," ,
## stringsAsFactors=TRUE, na.strings=c("NA","NaN"," ", ""))
## Use the head() function to check the structure of the data file.
# head(DATA)
```

Visualizing the data is an important step in quality assurance. Building some basic plots helps us see the general shape of the data, look for patterns, and identify outliers or anomalous data points. We will do this using the ggplot2 package. The “gg” is short for the “Grammar of Graphics”. Much more detailed treatments of the ggplot2 package are available, but in short the grammar of the package allows you to overlay different types of visual objects called “geoms” (e.g., points, lines, boxes) on top of the basic aesthetics of the plot (e.g., the y-axis values, the x-axis values).

We will build this plots in series, by creating an object called g1, adding to it to create an object called g2, and adding to that to create an object called g3. Using the plot() function, we can make the g3 object appear in our plot window. You can also plot g1 and g2 to see the effects of our different bits of code. From the plot window, you can also export the image in different sizes and formats (e.g., .png, .jpeg) and save it to your working directory.

```
## ----- Basic Data Visualization -----
## FIM scores by group and time -----
ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_line(aes(group=subID)) +
  geom_point(aes(fill=as.factor(subID)), pch=21, size=1, stroke=1) +
  facet_wrap(~AIS_grade) +
  scale_x_continuous(name = "Time from Admission (Months)", breaks=c(0:18)) +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)", limits=c(0,100)) +
  theme_bw() + theme(axis.text.x=element_text(size=8, colour="black"),
                    axis.text.y=element_text(size=10, colour="black"),
                    axis.title=element_text(size=12, face="bold")) +
  theme(strip.text.x = element_text(size = 12))+ theme(legend.position="none")
```



1.2 Random-Effects in Our Models

In a traditional general linear model (GLM), all of our data are independent (i.e., one data point per person). Statistically, we can write this as a linear model like:

$$y_i = \beta_0 + \beta_1(\text{Time}_i) + \epsilon_i$$

Each subject's actual score (y_i) is the result of an intercept (β_0) and that constant is modified based on Time (the slope, β_1 multiplied by the Time variable). The intercept and slope are collectively referred to as our statistical *MODEL*. Our model is not going to be perfect, however, so we need to include an error term (ϵ_i). Good models will have small errors and thus be a better approximation of our *DATA*. As such, we can more generally say that:

$$\text{Data}_i = \text{Model}_i + \text{Error}_i$$

Mixed-effect regressions are an extension of the general linear model, but they include *random-effects* in addition to the more traditional *fixed-effects* of our models. These random-effects allow us to account for statistical dependencies in our data. For instance, consider the data in the spaghetti plot in the figure we generated above (showing FIM Scores as a function of Group and Time).

It would be tempting to model these data using our traditional GLM where there was a fixed-effect of time. However, that would ignore the fact that time varies within each person and this violates one of our primary regression assumptions: that residuals are *independent* of each other.

To ensure that we have independent residuals, we need to account for the fact that we have multiple observations per person. As a step in this direction, we can add a random-effect of subject:

$$y_{ij} = \beta_0 + U_{0j} + \beta_1(\text{Time}_{ij}) + \epsilon_{ij}$$

The random-effect of subject (U_j) allows each subject to have a separate intercept ($\beta_0 + U_{0j}$). As such, we would refer to this model as a *random-intercepts; fixed-slope* model, because even though each subject has a unique intercept all subjects would have the same slope (β_1).

If we wanted to estimate a unique trajectory (i.e, slope) for each subject, then we would need to add a random-slope to our model:

$$y_{ij} = \beta_0 + \beta_1(Time_{ij}) + U_{0j} + U_{1j}(Time_{ij}) + \epsilon_{ij}$$

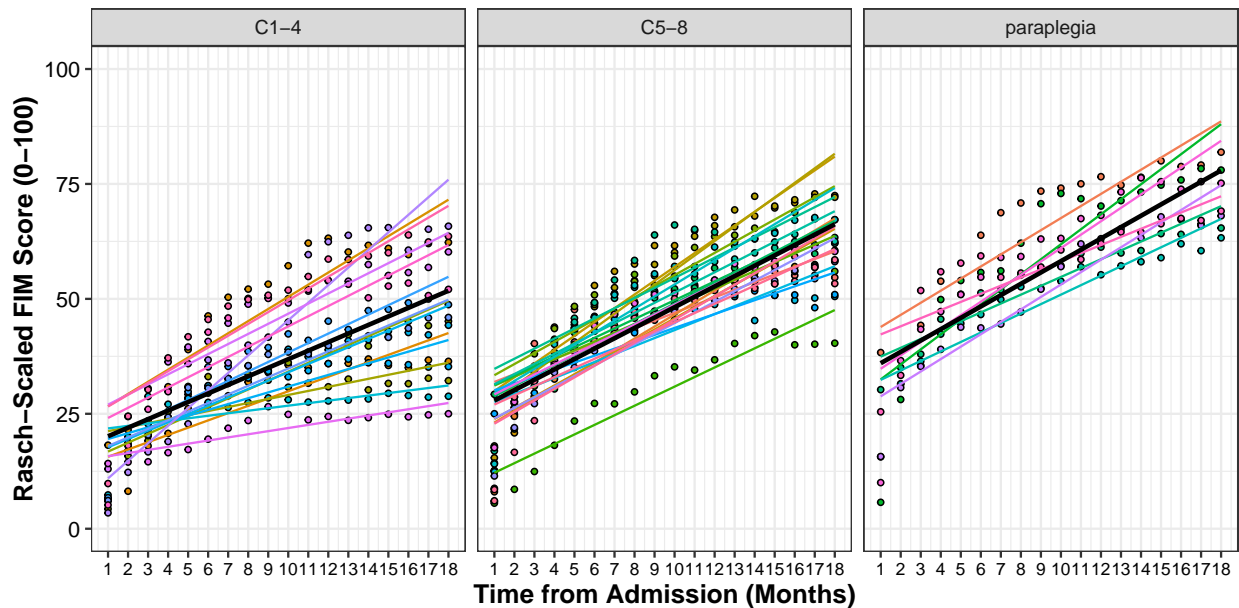
To show the effects specifically on the slopes and intercepts, this equation can equivalently be written as:

$$y_{ij} = (\beta_0 + U_{0j}) + (\beta_1 + U_{1j})(Time_{ij}) + \epsilon_{ij}$$

In this *random-intercepts; random-slopes* model, we estimate a unique trajectory for each person $((\beta_1 + U_{1j})(Time_{ij}))$. Visually, that model would look something like this:

```
## Linear Fit for each person -----
ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_point(aes(fill=as.factor(subID)), pch=21, size=1, stroke=0.5) +
  stat_smooth(aes(col=subID), se=FALSE, method="lm", lwd=0.5) +
  stat_smooth(aes(group=AIS_grade), col="black", se=FALSE, method="lm", lwd=1) +
  facet_wrap(~AIS_grade) +
  scale_x_continuous(name = "Time from Admission (Months)", breaks=c(0:18)) +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)", limits=c(0,100)) +
  theme_bw() + theme(axis.text.x=element_text(size=8, colour="black"),
                     axis.text.y=element_text(size=10, colour="black"),
                     axis.title=element_text(size=12, face="bold")) +
  theme(legend.position="none")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



```
## -----
```

The thick black lines represent the “group-level” trajectories $(\beta_0 + \beta_1(Time_{ij}))$ in each group. The estimated trajectories for each subject are color-coded based on the individual subjects. The intercepts for these lines

are captured by the group-level intercept plus the individual distance from that intercept ($\beta_0 + U_{0j}$). The slopes for these lines are captured by the group-level slope plus the individual distance from that slope ($(\beta_1 + U_{1j})(Time_{ij})$). Note that these random-effects (U 's) could be positive or negative, because they represent how this participant deviates from the norm. Thus, our *mixed-effects* MODEL is the combination of our fixed-effects (all of the β 's) and the random-effects (all of the U_j 's). However, $DATA = MODEL + ERROR$ still applies, so we need to include a random-error term for each data point, ϵ_{ij} .

In summary, we have the following terms to explain our **DATA**:

1. The **MODEL** includes fixed effects and random effects.
2. **Fixed-Effects** are the group-level β 's, these effects parallel the traditional main-effects and interactions that you have probably encountered in other statistical analyses.
3. **Random-Effects** are the participant-level U_j 's that remove statistical dependency from our data. This is bit of a simplification, but you can think of including the appropriate random-effects kind of like correctly specifying between-subject and within-subject variables. If you ignore important sources of statistical dependence, it can really throw off your model!
4. The **ERRORS**, also called the random errors or residuals, are the difference between our **MODEL**'s predictions and the actual **DATA**, ϵ_{ij} 's.

But our model doesn't look very linear?

Correct! In looking at the figures, it certainly doesn't look like a straight line is the best description our data. There appear to be diminishing returns in FIM scores over time. The rate of improvement is certainly slowing down and participants may even reach an asymptote at some point. Mathematically, we could try explain this curvature using **curvilinear** model or a **non-linear** model.

A **curvilinear** model creates a curving line, but is linear in its parameters. The most common way this is accomplished is adding polynomials to our model (e.g., x, x^2, x^3). For instance, in the equation below, our model is linear its parameters, but by raising x to different powers and adding those factors together, we can model a curvilinear relationship between x and y .

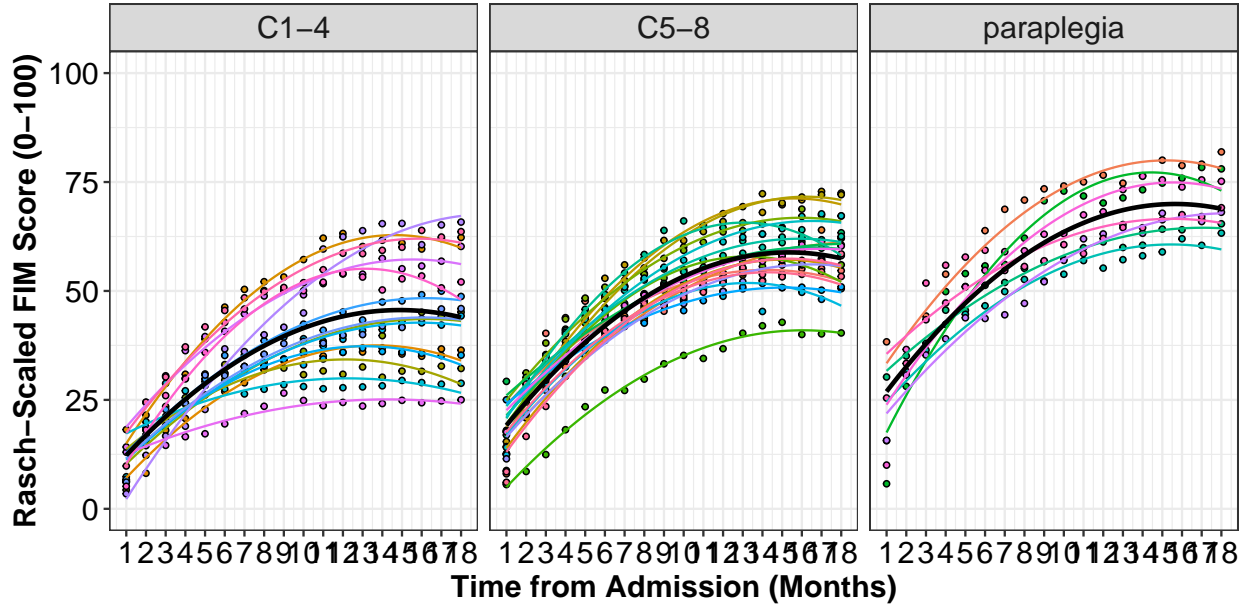
$$y_i = \beta_0 + \beta_1(x_{1i}) + \beta_2(x_{2i}^2) + \epsilon_i$$

In contrast, a *non-linear* model is *not linear* in its parameters. For instance, relationships that follow a power-function or an exponential-function (shown below) do not result from linear combination of the parameters (i.e., addition) and instead have more complex relationships.

$$y_i = \alpha + \beta * e^{(-\gamma/x_i)}$$

We *can* model non-linear relationships using mixed-effects regression, but that is more complicated topic that we will need to save for a later time. For now, let's focus on what a curvilinear model might look like in our data:

```
ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_point(aes(fill=as.factor(subID)), pch=21, size=1, stroke=0.5) +
  stat_smooth(aes(col=subID), se=FALSE, method="lm", formula=y~x+I(x^2), lwd=0.5) +
  stat_smooth(aes(group=AIS_grade), col="black", se=FALSE, method="lm",
              formula=y~x+I(x^2), lwd=1) +
  facet_wrap(~AIS_grade)+
  scale_x_continuous(name = "Time from Admission (Months)", breaks=c(0:18)) +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)", limits=c(0,100)) +
  theme_bw() +
  theme(axis.text=element_text(size=14, colour="black"),
        axis.title=element_text(size=14,face="bold")) +
  theme(strip.text.x = element_text(size = 14))+ theme(legend.position="none")
```



Visually, this curvilinear model looks like it is providing a much better explanation of our data, because there is a closer correspondence between our model estimate (the lines) and the real data (individual data points). Within each group, these lines would come from a mixed-effects model that looks like this:

$$y_{ij} = (\beta_0 + U_{0j}) + (\beta_1 + U_{1j})(Time_{ij}) + (\beta_2 + U_{2j})(Time_{ij}^2) + \epsilon_{ij}$$

The thick black lines correspond to the group-level estimates (β 's) and the thin lines correspond to the estimates for each individual participant ($(\beta + U)$'s). It looks like our curvilinear model has explained a lot of the within-participant variability, because the difference between our estimates and the data (ϵ 's) are very small. However, there does seem to be a fair amount of variability between participants (U 's) that remains to be explained.

1.3 Building Unconditional Models of Time

Now that we have provided a brief explanation of fixed-effects, random-effects, and how they work together to shape the relationship between the dependent variable and the time variable, let's focus on choosing which *shape* of the time variable is appropriate. Doing this requires *model selection*, where we will build a series of progressively more complicated models and compare them to each other to see which model provides the best explanation of the data.

1.3.1 The Random Intercepts Model

In our Random Intercepts model, we estimate a constant (intercept) for each participant and the overall constant. This overall constant is the **fixed-effect**, and individual deviations away from this constant are our **random-effects**. In writing up a study, we would refer to this as a random-effect of subject/ participant.

$$y_{ij} = (\beta_0 + U_{0j}) + \epsilon_{ij}$$

True to its name, the Random Intercepts model is going to estimate a flat line for each person. Each line will be different, however, because our random-effect of subject means that we are estimating a unique intercept for each person. This intercept will be equal to the mean for each participant, because the mean is the value that will produce the smallest errors.

```
DATA$year.0 <- (DATA$month-1)/12

raneff_int<-lmer(rasch_FIM~
                # Fixed-effects
                1+
                # Random-effects
                (1|subID), data=DATA, REML=FALSE)
```

Once our model is completed, we can use the `summary()` function to get a full description of the results, the `raneff()` function to get just the random-effects, the `fixef()` function to get just the fixed effect, and `coef()` function to get coefficient (fixed + random effect) for each participant.

```
summary(raneff_int)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: rasch_FIM ~ 1 + (1 | subID)
## Data: DATA
##
##      AIC      BIC    logLik deviance df.resid
##  5865.9   5879.6  -2929.9   5859.9      717
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.9937 -0.4357  0.2207  0.6874  1.6748
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## subID    (Intercept)    90.95      9.537
## Residual                    176.15    13.272
## Number of obs: 720, groups: subID, 40
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)   44.904      1.587 40.000   28.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Recall that the fixed-effect for the intercept is the overall, "group-level"
# intercept in our model. However, we also have a random-effect of subject for
# the intercept. This means that our model estimates a deviate for each subject
# from the group-level intercept. To see these random-effects using the raneff()
# function.
```

```
raneff(raneff_int)
```

```
# Remember that these are deviates from the fixed-effect, so if we want to see
# what the model is actually estimating for each person, we need to add the
# fixed-effect back in:
```

```
fixef(raneff_int)
```

```
# We could do this manually, by adding the fixef() output to the raneff() output,
# but we can also get the individual values using the coef() function:
```



```
coef(ranef_int)$subID
```

If you want the actual predictions of the model, rather than the estimated effects, you can use the fitted() function. Note the difference in the size of these arrays. The fitted() function gives us a prediction for each person at each point.

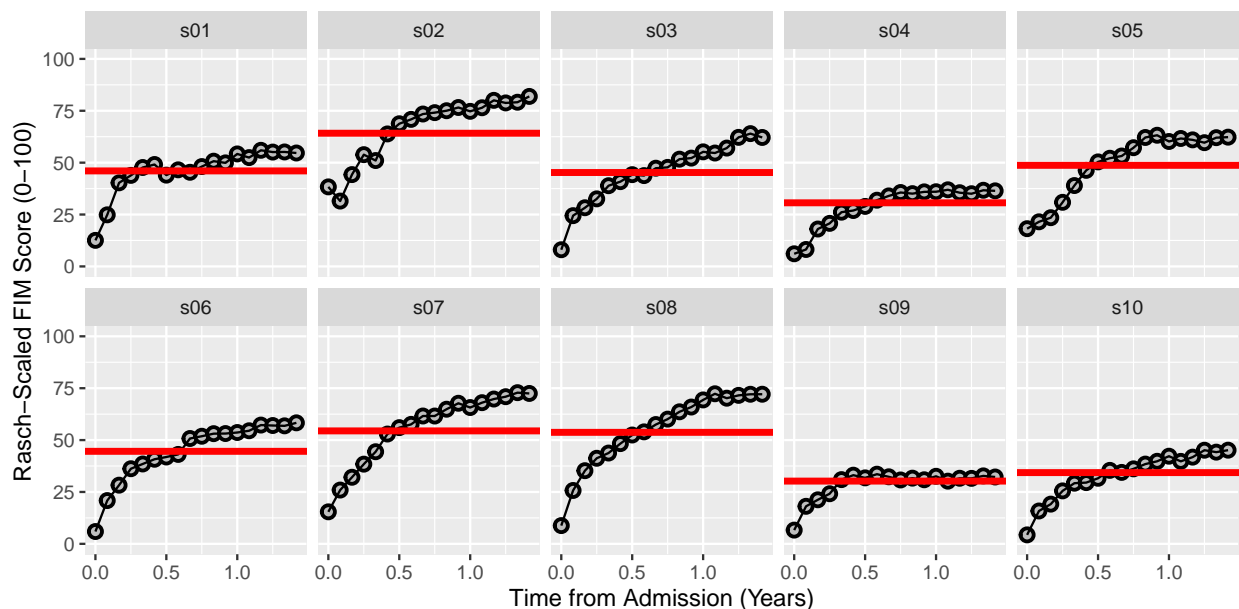
```
fitted(ranef_int)
```

To help us understand the model, we can plot these predictions for each person. As our data set is fairly large we will take a subset of the first 10 people.

```
first10<-DATA[c(1:180),] # Second, we'll make a smaller dataset with the predictions for these 10:
PRED<-data.frame(subID=c("s01","s02","s03","s04","s05","s06","s07","s08","s09","s10"), Intercepts=c(coef(ranef_int)$subID[1:10]), PRED)
```

When you print the “PRED” object to the screen, you should see a column of subject identifiers and the intercepts from our model, one for each person. We can then plot these predictions to show you exactly what the random intercepts model is doing:

```
ggplot(first10, aes(x = year.0, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) + facet_wrap(~subID, ncol=5)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100))+
  geom_abline(aes(intercept=Intercepts, slope=0), col="red", lwd=1.5, PRED)
```



1.3.2 The Fixed Slope/Random Intercept Model

In our Random Intercept - Fixed Slope model, we estimate an intercept for each participant and an overall slope, **but the slope is the same for each person**. The overall intercepts and slopes are the fixed-effects.

The prediction lines for each participant can have different heights (due to the random-intercepts), but all of these lines have the same slope (because there is no random-effect for the slope).

$$y_{ij} = (\beta_0 + U_{0j}) + (\beta_1)Time_{ij} + \epsilon_{ij}$$

```

raneff_slope<-lmer(rasch_FIM~
  # Fixed-effects
  1+year.0+
  # Random-effects
  (1|subID), data=DATA, REML=FALSE)
summary(raneff_slope)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: rasch_FIM ~ 1 + year.0 + (1 | subID)
## Data: DATA
##
##      AIC      BIC   logLik deviance df.resid
##  4921.9   4940.2  -2457.0   4913.9      716
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -5.4092 -0.4540  0.2028  0.6429  2.3953
##
## Random effects:
## Groups   Name            Variance Std.Dev.
## subID    (Intercept)    98.31     9.915
## Residual                  43.82     6.620
## Number of obs: 720, groups: subID, 40
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)  26.5891     1.6376  45.3468   16.24  <2e-16 ***
## year.0       25.8569     0.5706 680.0000   45.31  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## year.0 -0.247

```

As before, let's make a smaller dataset with the predictions for these 10 people to visualize what the random-intercept, fixed-slope model is doing:

```

PRED<-data.frame(subID=c("s01","s02","s03","s04","s05","s06","s07","s08","s09","s10"), Intercepts=c(co
PRED

```

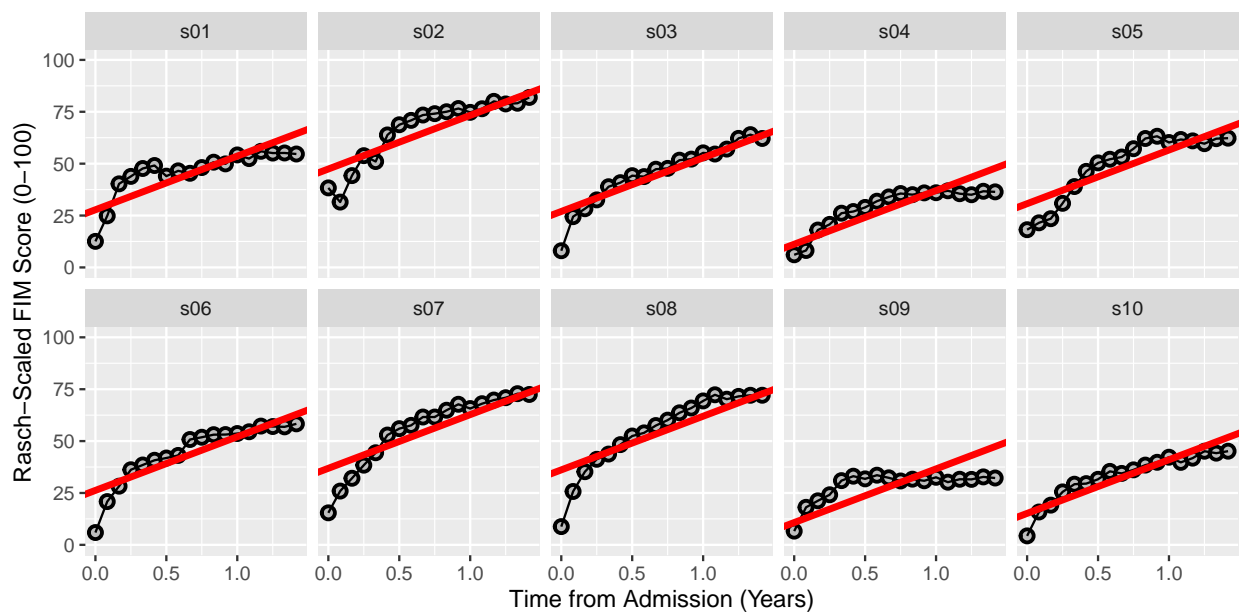
```

##      subID Intercepts  Slopes
## 1      s01  27.78992 25.85693
## 2      s02  47.41957 25.85693
## 3      s03  26.96130 25.85693
## 4      s04  11.17751 25.85693

```

```
## 5    s05    30.70916 25.85693
## 6    s06    26.23937 25.85693
## 7    s07    36.88449 25.85693
## 8    s08    36.12365 25.85693
## 9    s09    10.74127 25.85693
## 10   s10    15.16206 25.85693
```

```
ggplot(first10, aes(x = year.0, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) + facet_wrap(~subID, ncol=5)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)", limits=c(0,100))+
  geom_abline(aes(intercept=Intercepts, slope=Slopes), col="red", lwd=1.5, PRED)
```



1.3.3 The Random Slope/Random Intercept Model

In our Random Slopes model, we estimate an overall intercept and slope and a unique intercept and slope for each participant. These **overall intercepts and slopes** are the **fixed-effects** (B 's), and deviations away from these values are **random-effects** (U 's).

$$y_{ij} = (\beta_0 + U_{0j}) + (\beta_1 + U_{1j})(Time_{ij}) + \epsilon_{ij}$$

```
raneff_lin_rand<-lmer(rasch_FIM~
  # Fixed-effects
  1+year.0+
  # Random-effects
  (1+year.0|subID), data=DATA, REML=FALSE)
summary(raneff_lin_rand)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
```

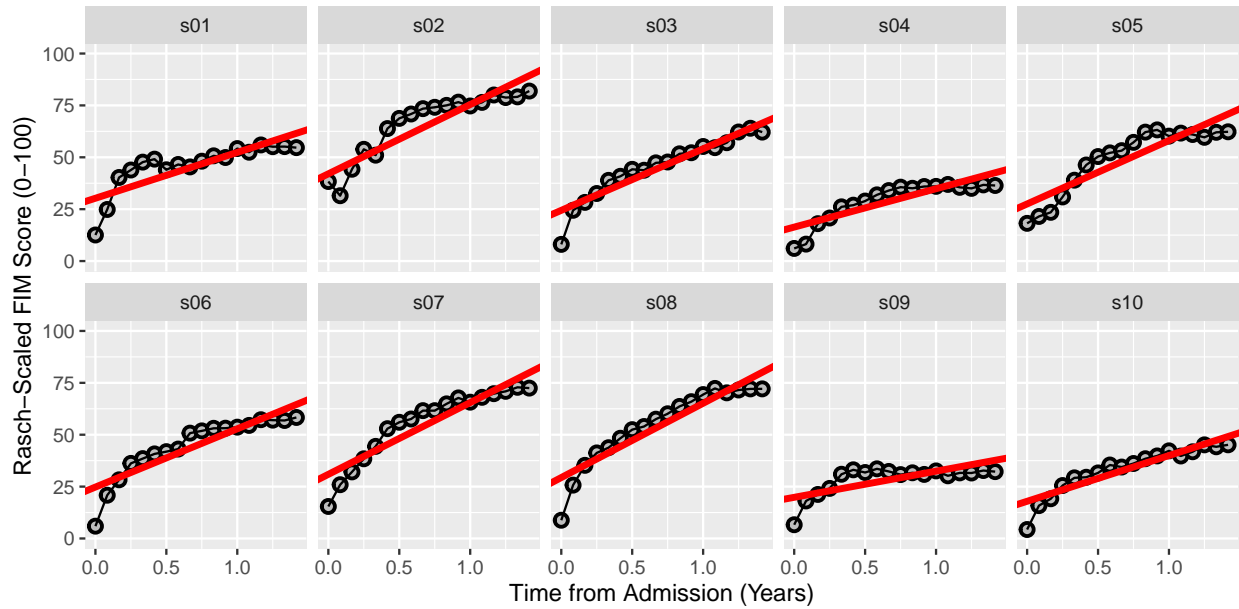
```
## Formula: rasch_FIM ~ 1 + year.0 + (1 + year.0 | subID)
## Data: DATA
##
##      AIC      BIC   logLik deviance df.resid
##  4794.8   4822.3 -2391.4   4782.8     714
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.7756 -0.4913  0.1596  0.6602  2.0907
##
## Random effects:
## Groups   Name      Variance Std.Dev. Corr
## subID    (Intercept) 47.83    6.916
##          year.0      50.82    7.129   0.37
## Residual                33.77    5.811
## Number of obs: 720, groups: subID, 40
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)   26.589      1.170 40.000   22.73  <2e-16 ***
## year.0        25.857      1.233 40.000   20.96  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## year.0 0.189
```

As before, let's make a smaller dataset with the predictions for these 10 people:

```
PRED<-data.frame(subID=c("s01","s02","s03","s04","s05","s06","s07","s08","s09", "s10"), Intercepts=c(co
PRED
```

```
##      subID Intercepts  Slopes
## 1    s01   30.26528 22.08379
## 2    s02   42.02847 33.37995
## 3    s03   24.43125 29.66606
## 4    s04   16.35868 18.49036
## 5    s05   27.51060 30.56442
## 6    s06   24.77084 28.08465
## 7    s07   30.90439 34.58137
## 8    s08   29.35009 35.80196
## 9    s09   19.83044 12.60255
## 10   s10   17.91374 22.04065
```

```
ggplot(first10, aes(x = year.0, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) + facet_wrap(~subID, ncol=5)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100))+
  geom_abline(aes(intercept=Intercepts, slope=Slopes), col="red", lwd=1.5, PRED)
```



Note that in this model I am using a different color for each line to indicate that a unique slope and intercept is being estimated for each participant.

1.3.4 The Quadratic Random Slopes/Intercepts Model

Finally, let's consider making a **curvilinear** model by adding $Time^2$ as a predictor to our model. For the sake of space, I am going to add both a random- and a fixed-quadratic effect at the same time. However, just like we did with the linear effect of time, you could add a fixed-effect by itself and then see if adding the random-effect leads to an improvement in model fit.

$$y_{ij} = (\beta_0 + U_{0j}) + (\beta_1 + U_{1j})(Time_{ij}) + (\beta_2 + U_{2j})(Time_{ij}^2) + \epsilon_{ij}$$

```
ranef_quad_rand<-lmer(rasch_FIM~
  # Fixed-effects
  1+year.0+I(year.0^2)+
  # Random-effects
  (1+year.0+I(year.0^2)|subID), data=DATA, REML=FALSE)
summary(ranef_quad_rand)
```

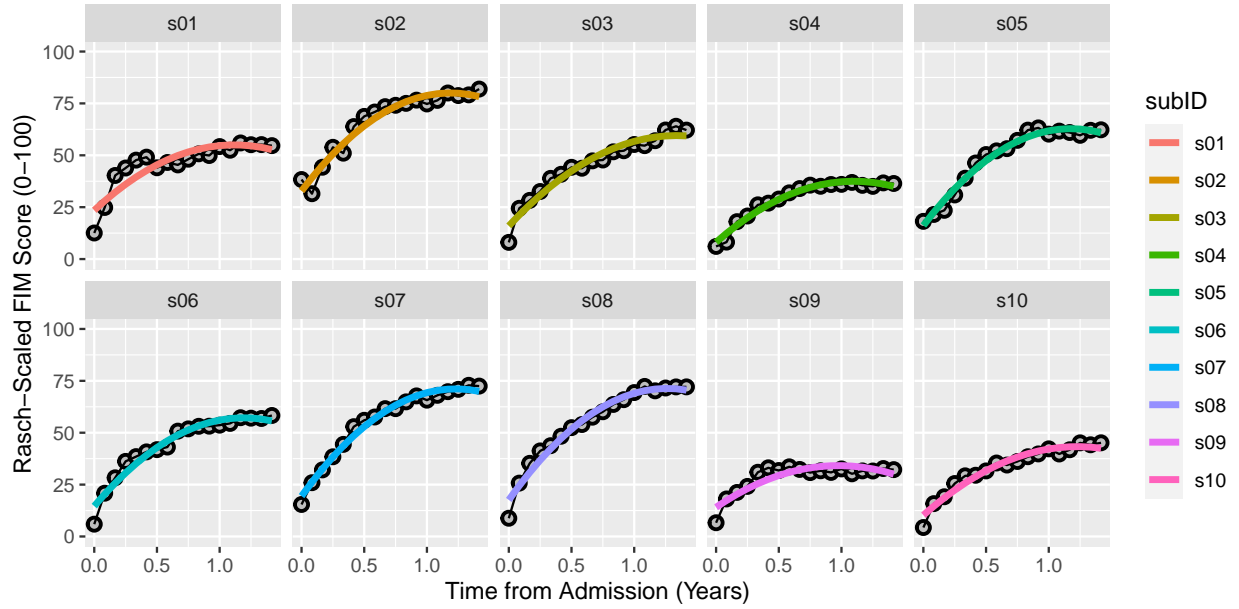
```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: rasch_FIM ~ 1 + year.0 + I(year.0^2) + (1 + year.0 + I(year.0^2) |
## subID)
## Data: DATA
##
##      AIC      BIC   logLik deviance df.resid
## 4039.3   4085.0 -2009.6   4019.3     710
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.4893 -0.5194  0.0455  0.5442  3.0358
```

```
##
## Random effects:
##   Groups   Name      Variance Std.Dev. Corr
##   subID    (Intercept) 46.224   6.799
##           year.0      242.058  15.558   0.07
##           I(year.0^2)  44.137   6.644  -0.08 -0.93
## Residual                9.754   3.123
## Number of obs: 720, groups: subID, 40
##
## Fixed effects:
##           Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   18.103      1.120  39.999   16.17 <2e-16 ***
## year.0         64.044      2.666  39.998   24.03 <2e-16 ***
## I(year.0^2)   -26.956      1.262  39.998  -21.36 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) year.0
## year.0      -0.030
## I(year.0^2)  0.049 -0.921
```

One last time, let's make a smaller dataset with the predictions for these 10 people:

```
DATA$quad_pred <- fitted(raneff_quad_rand) # Save model predictions to data frame
first10<-DATA[c(1:180),] # Second, we'll make a smaller dataset with the predictions for these 10:

ggplot(first10, aes(x = year.0, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) + facet_wrap(~subID, ncol=5)+
  scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100))+
  geom_line(aes(y=quad_pred, col=subID), lwd=1.5)
```



Again, given that this is a random-slopes model, I am using a different color for each line to indicate that a unique intercept, linear slope, and quadratic slope are being estimated for each participant.

Qualitatively, this looks like our **best fitting** model. However, we don't want to make decision based on visual inspection alone, so in the next section we will describe how to choose between competing models.

1.4 Comparing between Models

Now that we have created several different models, how do we decide which models are the best? How do we decide which parameters to include and which parameters are “statistically significant”? When it comes to mixed-effect linear models (or other forms of “multi-level” models), we use a similar criterion to traditional regression. That is, we still rely on the general idea that...

$$Data_i = Model_i + Error_i$$

... and if the error is reduced by a large enough magnitude, then we will call that effect statistically significant (i.e., the parameter reduced error by an unusually larger amount under the null-hypothesis.). However, there are some differences between mixed-effect models and traditional Ordinary Least Squares regression.

For instance, you might have noticed that p-values are conspicuously absent from the LME4 output. The reasons for this are complicated, but it has to do with the fact that these models can have statistical dependencies and unbalanced/missing data that do not make the calculation of traditional p-values tenable. In short, your denominator degrees of freedom can get a little crazy. (You can also read an explanation from Doug Bates, author of the lme4 package, here: <https://stat.ethz.ch/pipermail/r-help/2006-May/094765.html>). However, the main things that you should know are:

1. If you really, really want p-values for individual parameters, you can get them from packages that implement the Welch-Satterthwaite approximation to estimate the appropriate degrees of freedom, like the “lmerTest” package.
2. We are most interested in comparisons between models and less so the individual parameters within a model. All of our models were fit using Maximum Likelihood Estimation (explained below) and we judge models based on a reduction in ERROR that we call Deviance. Fortunately, there are several quantitative and objective methods for evaluating the change in deviance. We will focus on two of these methods, the *Wald Test for the change in Deviance* and the *Akaike Information Criterion (AIC)*.

You can read more about Maximum Likelihood Estimation from other resources, but conceptually the idea of ML estimation is that our computer tests a long series of parameters until it arrives at the specific set of parameters that leads to the smallest error in our data. This is why it is referred to as “maximum likelihood”, because we arrive at the set of values (for the given parameters) that are most likely to have produced the data we observed. The goodness of fit for these parameter estimates is quantified in something called the Deviance, which is a transformation of the likelihood.

$$Deviance = -2 * \log(Likelihood)$$

Where *Likelihood* is defined by the amount of error () left behind by our estimates. Thus, if we delve a little deeper, the *Deviance* is:

$$Deviance = N * \log(2\pi\sigma_e^2 + (1/\sigma_e^2)(\sum_{i=1}^n \epsilon_i)$$

This formula is kind of scary, but there are two things you need to notice about it: 1. Looking at the right side of the equation, notice that the deviance is still largely determined by the sum of errors. *Thus, everything else being equal, smaller errors are going to lead to a smaller deviance.* 2. Notice that size our sample shows up in two different places (the N at the beginning and the fact that we are summing over N on the right hand side). This means that the deviance is sensitive to the amount of data we are using. *Thus, if we want to compare models based on their deviance, those models need to be based on the same amount of data.*

##1.4.1 The Wald Test of the Change in Deviance Now we are ready to make a comparison between some of our different statistical models by comparing the change in the Deviance. Let’s start by comparing the Random Intercepts, Fixed Slopes, Random Slopes, and Quadratic Random Slopes models using the `anova()` function in R.

```
## ----- Comparing between Models -----
# Now that we have our three random-effects models:
# 1. Random Intercepts
# 2. Fixed Slopes and Random Intercepts
# 3. Random Slopes and Random Intercepts
# 4. Quadratic Random Slopes and Intercepts
# How do we decide which one is most appropriate?
# We can do this is by analyzing the the variance explained by each model.
anova(raneff_int,raneff_slope,raneff_lin_rand, raneff_quad_rand)

## Data: DATA
## Models:
## raneff_int: rasch_FIM ~ 1 + (1 | subID)
## raneff_slope: rasch_FIM ~ 1 + year.0 + (1 | subID)
## raneff_lin_rand: rasch_FIM ~ 1 + year.0 + (1 + year.0 | subID)
## raneff_quad_rand: rasch_FIM ~ 1 + year.0 + I(year.0^2) + (1 + year.0 + I(year.0^2) | subID)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## raneff_int      3 5865.9 5879.6 -2929.9   5859.9
## raneff_slope     4 4921.9 4940.2 -2456.9   4913.9 945.98  1 < 2.2e-16 ***
## raneff_lin_rand   6 4794.8 4822.3 -2391.4   4782.8 131.06  2 < 2.2e-16 ***
## raneff_quad_rand 10 4039.3 4085.0 -2009.6   4019.3 763.58  4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on these results, the each model was a statistically significant improvement on the model before it (see the $Pr(> Chisq)$ column). Ultimately, it looks like the quadratic random slopes model was our best model, being a statistically significant improvement over the linear random slopes model ($p < 2.2e - 16$).

##1.4.2 Akaike's Information Criterion The AIC is another method to evaluating model fit that is based on the Deviance. Although we won't get into the details of the math behind it, the importance of the AIC is in the name "Information Criterion". That is, the AIC is all about informativeness which is slightly different from just being the model that produces the smallest deviance. For a model to be informative, the parameters need to generalize to other new datasets, not just provide the best explanation of the current data. Therefore, the AIC introduces a penalty to reduce over-fitting of the model:

$$AIC = Deviance + 2(k)$$

In the formula above, k = number of parameters in the model. Thus, for a parameter to improve the AIC it has to reduce the deviance by >2 times the number of parameters. We won't get into why the number of $2(k)$ seems to work so well, but the key thing to know is that *the AIC imposes a penalty based on the number of parameters and is thus a more conservative test than the Wald Test of the change in Deviance*.

We can see this in action by comparing our three random-effects model in the ANOVA output above. Smaller AIC is an indicator of better fit, and you can see that the quadratic random slopes model has the smallest AIC (4039.3). Importantly though, compare the AIC to the deviance for each model. The AIC for each individual model is larger than the deviance for that model. This difference is due to the penalty the AIC introduces to prevent overfitting!

As a word of caution, there are no fixed cut-offs for a "statistically significant" change in the AIC although some research has been done exploring how the ΔAIC relates to other measures of effect-size (see Long, 2012). In general, it is a good idea to declare your minimum ΔAIC in advance as an integer value (i.e., greater than at least a 1 point change in AIC). I usually use a value of 2, which some researchers have suggested as a plausible rule of thumb for preferring one model over another (Burnham, 2002; Burnham & Anderson, 1998). That is, if an additional parameter reduces the AIC by ≥ 2 points, then I will stick with the more complex model. If the additional parameter reduces the AIC by < 2 points, then I will stick with the simpler model.

Now a smaller AIC always means better model fit, so why pick a $\Delta AIC > 2$? Well, the accuracy of the AIC depends on the model meeting your distributional assumptions (just like p-value does) and those will almost always be violated to some degree (even if the violation is not statistically significant). Additionally, when the sample size is small there is bias in the AIC and it will not sufficiently punish additional parameters (i.e., **the AIC does not do its job to prevent overfitting in small samples**). As a remedy to this, some people recommend using the corrected AIC (AIC_c ; Hurvich & Tsai, 1989):

$$AIC_c = AIC + \frac{2k^2 + 2k}{n - k - 1}$$

The AIC_c correction is great and I definitely recommend people use it. However, much like distributional assumptions, we need to think about sample size continuously rather than categorically. There is no point where sample size is suddenly too small or sufficiently big. You could work out just how much k and n trade-off in the formula above, but that is why I usually recommend committing to an integer value of improvement up front (e.g., 1 or 2 points in the AIC). I like to choose a $\Delta AIC_c > 2$ points to create a bit of a buffer. That is, allowing for some error in my distributional assumptions and some bias from my sample size, I want to stack the deck against myself and really make sure that I am not overfitting the model.

1.5 What comes next?

So far, we have describe the basic components of a mixed-effect model and how we can select the best **unconditional** model using the Wald Test of the Change in Deviance or Akaike's Information Criterion. In the next module, we will explore how to create conditional models where we add in effects like *AIS Grade* to see how the slopes and intercepts differ between groups of participants.

Before that, however, I want to take a moment to reflect on mixed-effect regression models and how they related to another dominant method for analyzing repeated measures data, specifically repeated measures analysis of variance (**RM ANOVA**).

1.5.1 Constrasting Mixed-Effects Regression and RM ANOVA

I hope this brief introduction gives you some sense of what mixed-effects regression is and what it can do. Mixed-effect regression is a very useful analytical tool when it comes the analysis of longitudinal data or in study designs where participants are exposed to different conditions (i.e., repeated measures designs). Although mixed-effects regression is very useful in these study designs, the more commonly used method of analysis is repeated measures analysis of variance (RM ANOVA).

RM ANOVA is a perfectly valid method of analysis for a lot of study designs, but in many contexts, researchers use a RM ANOVA when a mixed-effect regression might be more appropriate or effective. Lohse, Shen, and Kozlowski (2020) provide a more detailed contrast of these two methods, but I have recreated some of the central arguments from that paper below.

- **Model concept**
 - **RM ANOVA:**
 - * Compares means of a continuous outcome stratified by one or more categorical variable(s) to the grand mean.
 - * Individuals are treated as a factor with error aggregated from each individual's mean of repeated measures and partitioned as intra-individual variance from the error term.
 - **Mixed-Effects Regression:**
 - * Accounts for correlations in data with clustered or nested structure.
 - * In longitudinal models, change over time is considered a within-person factor accounting for within-person correlations across time points and estimating error as residuals from each individual's trajectory, and between-person error is accounted for as random effects in a correlation matrix, which can be explained by fixed covariate associations with trajectory parameters.
- **Modeling of the outcome over time**
 - **RM ANOVA:**
 - * Addresses questions about mean difference.
 - * Time is not inherently captured in the repeated measure, instead discrete time points are treated as levels of a categorical variable with a mean for each time point.
 - * Mean differences between time points do not represent change over time, since time is an not explicit part of the model.
 - **Mixed-Effects Regression:**
 - * Time is modeled explicitly for the outcome variable as a trajectory of change.
 - * The model assumes a common pattern of change for the group (fixed effects), but individuals can vary from that pattern (random effects).
 - * The shape of the trajectory is determined by fitting progressively more complex mathematical functions that are likely to fit the pattern of raw data scores, and testing a fit statistic (e.g., Akaike Information Criterion or Bayesian Information Criterion).
 - * Of particular use is the ability to estimate the magnitude and timing of a plateau or other milestone on the trajectory.
- **Variability in timing of data points**
 - **RM ANOVA:**
 - * Requires common, discreet time points; variability in actual timing may contribute to measurement error in categorized time points.
 - * Measurement error may accrue within time points if outcome measurement varies by time within a time point, e.g., measurement at a time point varies by \pm time units around that point. Individuals' scores on an increasing trajectory may be overestimated if captured before the time point or underestimated if captured after.

- **Mixed-Effects Regression:**
 - * Can accommodate variability in spacing of time points and in the actual timing of individual data collection.
 - * Time points can be spaced farther apart where little change is expected, and closer together where more change is expected.
 - * Individual measurement can vary from the target time points. If, for example, 5 weekly measurements are planned over 4 weeks, a time variable defined in days can capture the actual day of measurement, rather than collapsing to the weekly time point.
- **Data missing on the outcome**
 - **RM ANOVA:**
 - * Missing outcome data cannot be accommodated, without complicated statistical adjustments (such as multiple imputation) when data are missing at random.
 - * Including only cases with complete data will reduce statistical power and risk bias to the model if data are missing not at random (MNAR).
 - * Depending on the method employed, imputing missing values may not bias parameter estimates, but may reduce standard errors risking Type I errors in hypothesis tests.
 - **Mixed-Effects Regression:**
 - * Data that is missing at random (MAR) can be accommodated without excluding cases.
 - * However, models can be biased if important time points are missing (e.g., no data where important change occurs).
 - * Models with data that is MNAR can be fit, but models may be biased. For example, an unbalanced data set is one in which later time points are more likely to be missing, which can occur due to drop out, or outcome measurement that is performed during an intervention that varies for individuals.
 - * Imputation of outcome data is generally not recommended.
- **Data missing on covariates**
 - **RM ANOVA:**
 - * Missing between-person covariate data cannot be accommodated.
 - * Cases are either dropped from analysis or retained by imputing missing values.
 - **Mixed-Effects Regression:**
 - * Missing between-person covariate data cannot be accommodated.
 - * Cases are either dropped from analysis or retained by imputing missing values.
- **Time-varying covariates**
 - **RM ANOVA:**
 - * Time varying covariates cannot be accommodated in a RM ANOVA model.
 - **Mixed-Effects Regression:**
 - * Time varying covariates can be included, but you need to be careful about collinearity and variance at both the between- and within-subject levels.

Conclusions

Hopefully, this introduction gives you a sense of what mixed-effect regression is and what it can do. There are a whole host of topics that we haven't covered yet, but mixed-effect regression is an incredibly flexible and powerful method for analyzing your data. That flexibility comes at a cost, however, as analytical flexibility also means greater complexity and there are a lot of choices that an analyst must make that can have significant influence on the results.

That said, in the following modules I want to explain how mixed-effect regression is useful for factorial designs with repeated measures, truly longitudinal designs, and designs that mix repeated measures with time series data.

References

Lohse, K. R., Shen, J., & Kozlowski, A. J. (2020). Modeling Longitudinal Outcomes: A Contrast of Two Methods. *Journal of Motor Learning and Development*, 1(aop), 1-21. doi: <https://doi.org/10.1123/jmld.2019-0007>