ACRM 2018 Longitudinal Data Analysis Workshop

Keith Lohse[1], and Al Kozlowski[2], 2018-09-07
[1] University of Utah; [2] Michigan State University, Mary Free Bed Rehabilitation Hospital

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 connect.

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

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.

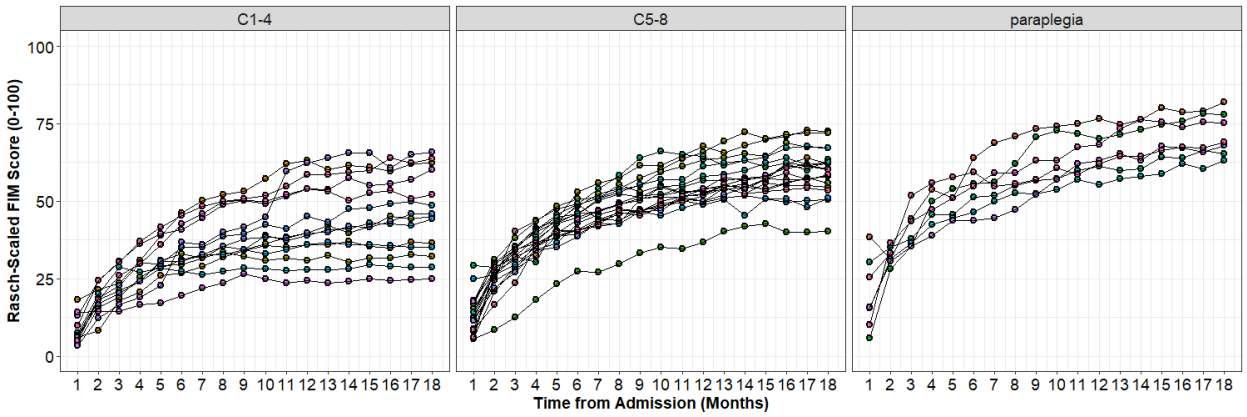
```
Setting the Directory ----------------------------------------------------
getwd()
setwd("C:/Users/Folder/SubFolder/SubSubFolder/")
list.files()
# Make sure that the file data_session1.csv is saved in your working
directory.

# 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=",",
               na.strings=c("NA","NaN"," ",""))

# Use the head() function to check the structure of the data file.
head(DATA)

# Alternately you can also download the data file from the web here:
# DATA <-
read.csv("https://raw.githubusercontent.com/keithlohse/LMER_Clinical_Science/
master/data/data_session1.csv")
```

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).

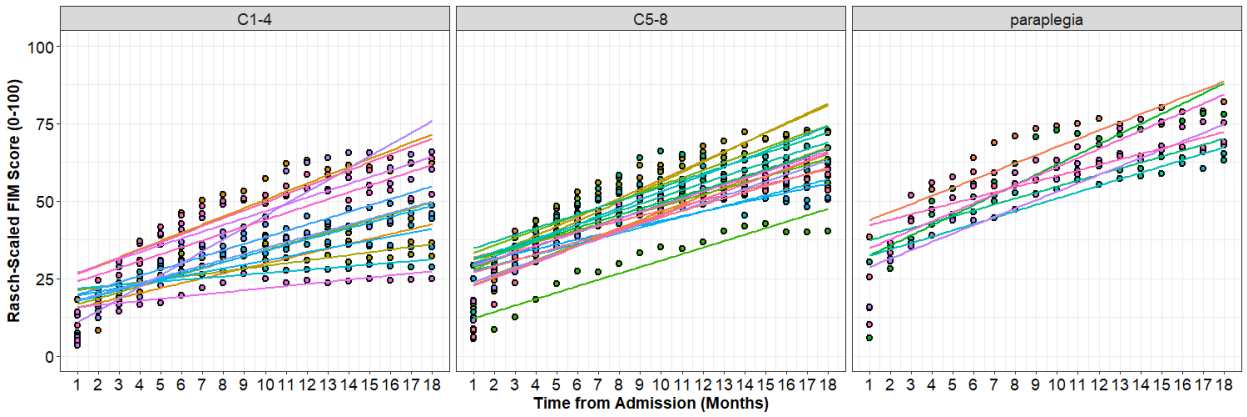
We will build this plots in series, by creating an object called g1, adding to it to create an object called g2, and adding to that to create an object called g3. Using the plot() function, we can make the g3 object appear in our plot window. You can also plot g1 and g2 to see the effects of our different 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 gender and time --------------------------------------------
g1<-ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_point(aes(fill=as.factor(subID)), pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) +
  facet_wrap(~AIS_grade)
g2<-g1+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))
g3 <- g2 + 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")
```

```
plot(g3)
## ------------------------------------------------------------------------
```

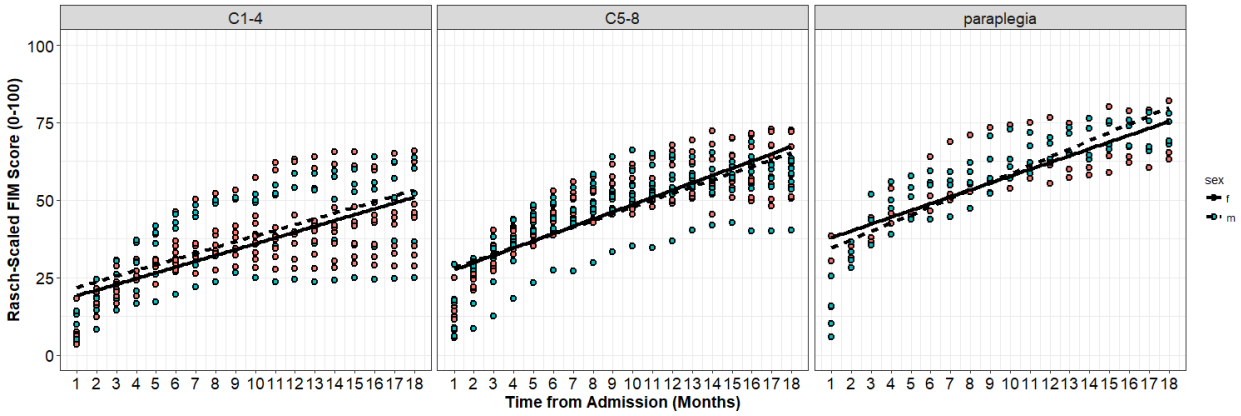



```
## Linear Fit for each person ----------------------------------------------
g1<-ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_point(aes(fill=as.factor(subID)), pch=21, size=2, stroke=1.25) +
  stat_smooth(aes(col=subID), se=FALSE, method="lm") +
  facet_wrap(~AIS_grade)
g2<-g1+scale_x_continuous(name = "Time from Admission (Months)",
breaks=c(0:18)) +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100))
g3 <- g2 + 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")

plot(g3)
## ------------------------------------------------------------------------
```




```
## Linear Fit by gender ----------------------------------------------------
g1<-ggplot(DATA, aes(x = month, y = rasch_FIM)) +
  geom_point(aes(fill=sex), pch=21, size=2, stroke=1.25) +
  stat_smooth(aes(lty=sex), col="black", lwd=1.5,se=FALSE, method="lm") +
  facet_wrap(~AIS_grade)
```

```
g2<-g1+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))
g3 <- g2 + 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))

plot(g3)
## ------------------------------------------------------------------------
```



1.2 Random-Effects in Our Models

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

$$y_i = B_0 + B_1(TIME_i) + \epsilon_i$$

Each subject's actual score (y_i) is the result of an intercept (B_0) and that constant is modified based on Time (the slope, B_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:

$$DATA = MODEL + ERROR$$

The code for LMER is essentially the same as GLM, except that we will be using lmer() instead of lm(). (Note that you need lme4 installed to have access to the lmer() function.) Conceptually, LMER is a lot like GLM but we need to 'partition' our variance into different sources.

$$y_{ij} = B_0 + U_{0j} + B_1(TIME_{ij}) + U_{1j}(TIME_{ij}) + \epsilon_{ij}$$

In LMER, we now have data indexed by time point (i) and by participant (j). Each data-point, y_{ij}, can still be described by the overall intercept and slope, B_0 and B_1, plus a random effect for each subject, U_{0j} and U_{1j}.

Note that these random-effects could be positive or negative, because they represent how this participant deviates from the norm. Thus, in LMER our *MODEL* is the combination of our fixed-effects (all of the B'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 in our DATA:

- The *MODEL* includes fixed effects and random effects.
- *Fixed-Effects* are the group-level B's, these effects parallel the traditional main-effects and interactions that you have probably encountered in other statistical analyses.
- *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 not including the appropriate random-effects like running a between-subjects ANOVA when you should be running a repeated-measures ANOVA.)

The *ERRORS*, or more specifically *Random Errors*, are the difference between our *MODEL*'s predictions and the actual *DATA*.

1.2.1 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.)

$$y_{ij} = B_0 + U_{0j} + \epsilon_{ij}$$

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 mean for each participant, because the mean is the value that will produce the smallest errors. Note that the R syntax in the model below is a little bit different, but conceptually the same, and follows the form:

$$\hat{y}_{ij} = B_0 + (U_0|Subject)$$

In R we are constructing a model that will predict a value for each person at each time, $\hat{y}_{ij}$, based on a single fixed effect, B_0, and a random-effect, U_0, for each subject.

```
# Understanding basic random-effects ----------------------------------------
# Random Intercepts Model ----
raneff_00<-lmer(rasch_FIM~
                  # Fixed-effects
                  1+
                  # Random-effects
                  (1|subID), data=DATA, REML=FALSE)
```

Once or 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_00)
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
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 t value
(Intercept)   44.904      1.587    28.3
```

```
# 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.
```

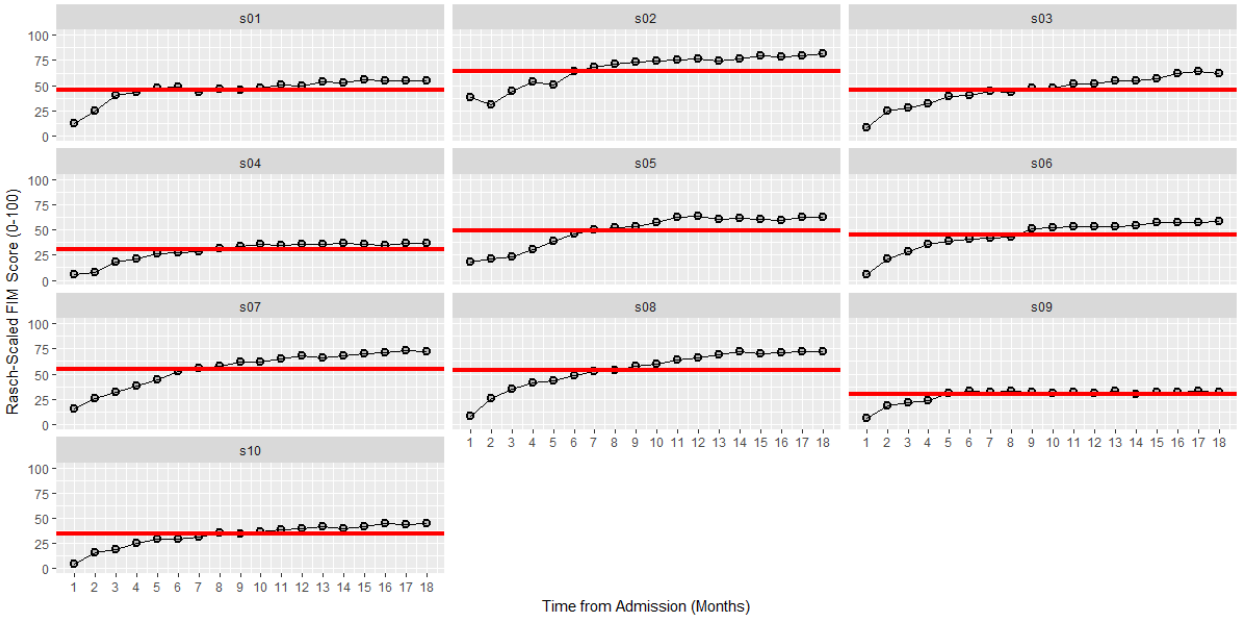
```
ranef(raneff_00)

# 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_00)

# We could do this manually, by adding the fixef() output to the ranef()
output, but we can also get the individual values using the coef() function:
coef(raneff_00)$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 givesus a prediction for each person
at each point.
fitted(raneff_00)
```

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(raneff_00)$subID[c(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:

```
g1<-ggplot(first10, aes(x = month, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) +
  facet_wrap(~subID, ncol=3)
g2<-g1+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))
print(g2)
g3<-g2+geom_abline(aes(intercept=Intercepts, slope=0), col="red", lwd=1.5,
PRED)
plot(g3)
```



1.2.2 The Fixed Slope 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).

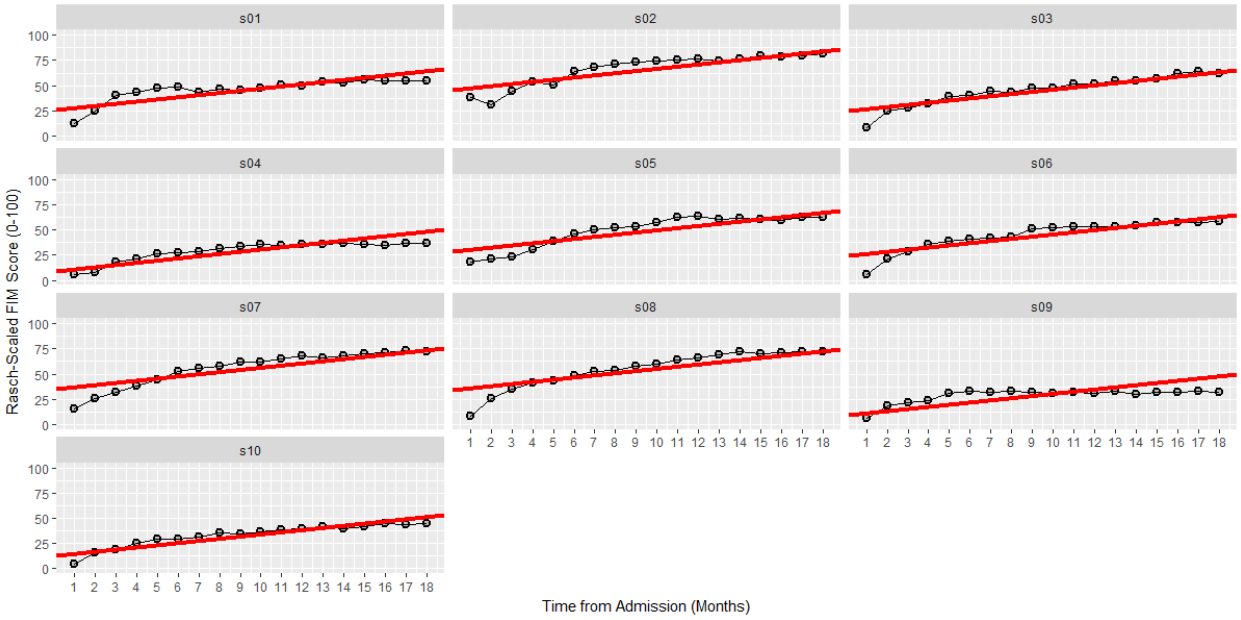
$$y_{ij} = B_0 + U_{0j} + B_1 * (TIME_{ij}) + \epsilon_{ij}$$

```
# Fixed Slope Model ----
raneff_01<-lmer(rasch_FIM~
                  # Fixed-effects
                  1+month+
                  # Random-effects
                  (1|subID), data=DATA, REML=FALSE)
summary(raneff_01)
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
approximations to
  degrees of freedom [lmerMod]
Formula: rasch_FIM ~ 1 + month + (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) 24.43435    1.65002  46.70000   14.81   <2e-16 ***
month        2.15474    0.04755 680.00000   45.31   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      (Intr)
month -0.274
```

```r
# Note that when we run the ranef() function, there is still only a random effect of the intercept:
ranef(raneff_01)

# When we run the fixef() function, however, now have both a slopes and intercept:
fixef(raneff_01)

# When we run the coef() function, you can see that each person has a unique intercept, but everyone has the same slope:
coef(raneff_01)$subID


# 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(coef(raneff_01)$subID[c(1:10),1]),
                  Slopes=c(coef(raneff_01)$subID[c(1:10),2]))
PRED

# We can now plot the predictions for each person, note that although each person has a different intercept, they all have the same slope.
g1<-ggplot(first10, aes(x = month, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) +
  facet_wrap(~subID, ncol=3)
g2<-g1+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))
g3<-g2+geom_abline(aes(intercept=Intercepts, slope=Slopes), col="red",
lwd=1.5, PRED)
plot(g3)
```



1.2.3 The Random-Slopes 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).

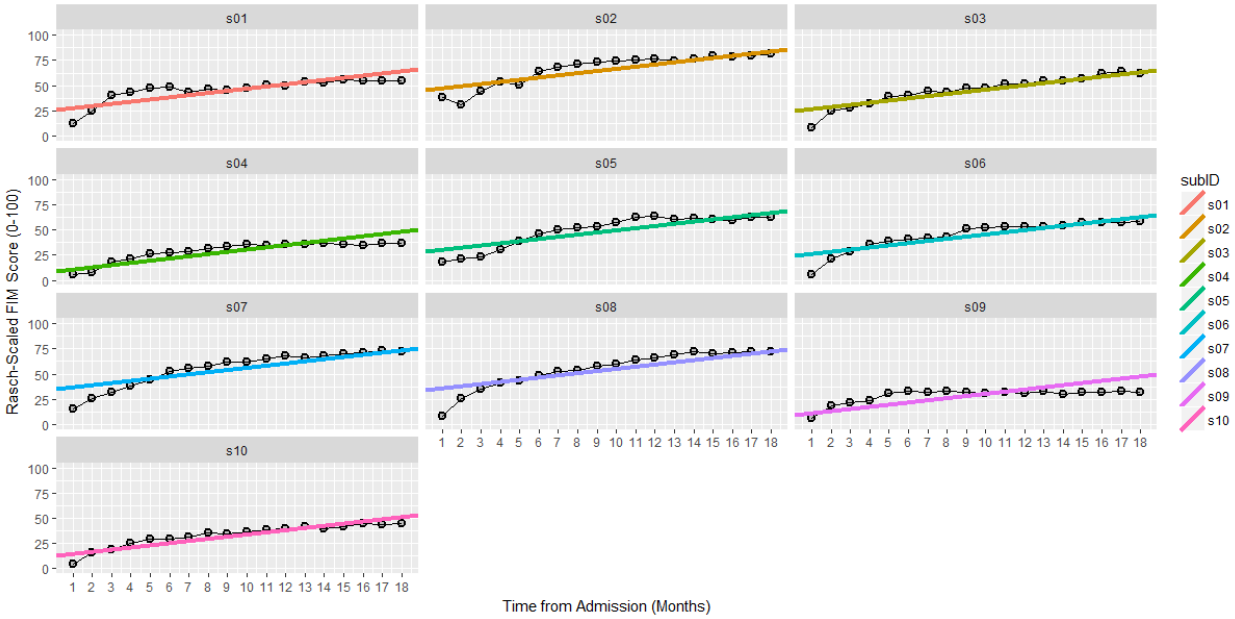
$$y_{ij} = B_0 + U_{0j} + (B_1 + U_{1j}) * (TIME_{ij}) + \epsilon_{ij}$$

```
# Random Slopes and Intercepts Model ----
raneff_02<-lmer(rasch_FIM~
                  # Fixed-effects
                  1+month+
                  # Random-effects
                  (1+month|subID), data=DATA, REML=FALSE)
summary(raneff_02)
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
approximations to
  degrees of freedom [lmerMod]
Formula: rasch_FIM ~ 1 + month + (1 + month | 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) 45.1762  6.7213
```

```
          month        0.3529  0.5941    0.29
 Residual             33.7651  5.8108
Number of obs: 720, groups:  subID, 40

Fixed effects:
            Estimate Std. Error      df t value Pr(>|t|)
(Intercept)  24.4343     1.1548 40.0000   21.16   <2e-16 ***
month         2.1547     0.1028 40.0000   20.96   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
      (Intr)
month 0.103
```

```
# Now when we run the ranef() function, there is random effect for both the
# slope and the intercept:
ranef(raneff_02)

# When we run the fixef() function, it returns the group-level intercept and
slope:
fixef(raneff_02)

# When we run the coef() function, the random-effects (deviates) are combined
# with the group-level fixed-effects to get a unique slope and intercept
# for each person:
coef(raneff_02)$subID


# 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(coef(raneff_02)$subID[c(1:10),1]),
                 Slopes=c(coef(raneff_02)$subID[c(1:10),2]))
PRED

# Now we can plot the predictions for each person, note that each person has
a
# different intercept and slope.
g1<-ggplot(first10, aes(x = month, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  geom_line(aes(group=subID)) +
  facet_wrap(~subID, ncol=3)
g2<-g1+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))
g3<-g2+geom_abline(aes(intercept=Intercepts, slope=Slopes, col=subID),
lwd=1.5, PRED)
plot(g3)
```



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 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…

$$DATA = MODEL + ERROR$$

… 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 (ϵ) left behind by our estimates. Thus, if we delve a little deeper, the *Deviance* is:

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

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 size). 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.**

1.3.1. 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, and 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
# ... 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_00,raneff_01,raneff_02)
```

The anova() function gives a number of valuable pieces of information. First, it explicitly lists the models that are being tested. Below, it gives us the model name and the degrees of freedom for each model. It then gives us the AIC, the related Bayesian Information Criterion (or BIC), the log of the Likelihood (or logLik), and the Deviance.

```
> anova(raneff_00,raneff_01,raneff_02)
Data: DATA
Models:
object: rasch_FIM ~ 1 + (1 | subID)
..1: rasch_FIM ~ 1 + month + (1 | subID)
..2: rasch_FIM ~ 1 + month + (1 + month | subID)
       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
object  3 5865.9 5879.6 -2929.9   5859.9
..1     4 4921.9 4940.2 -2456.9   4913.9 945.98      1  < 2.2e-16 ***
..2     6 4794.8 4822.3 -2391.4   4782.8 131.06      2  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The Random Intercepts model acts as our "benchmark" against which other models can be compared. For instance, the reduction in deviance from the Random Intercepts (Deviance = 5859.9) to the Fixed

Slopes model (Deviance = 4913.9) is 945.98. The change in deviance follows a χ^2 distribution, which allows us to test the statistical significance of the difference between these two models using a classic null hypothesis significance test.

The Fixed Slopes model (df = 4) has one more parameter than the Random Intercepts model (df =3) so the degrees are freedom for the $\chi^2 = 1$ and the p-value for $\chi^2(1) = 945.98$ is < 0.001. Thus, we would conclude that the Fixed Slope model is **significantly** better than the Random Intercepts model. Similarly, the Random Slopes model reduces the deviance by 131.06 compared to the Fixed-Slopes model and the p-value for $\chi^2(2) = 131.06$ is <0.001. Thus, we would conclude that the Random Slopes model is also a **statistically** better model than the Fixed Slopes model (assuming α = 0.05).

1.3.2. The Akaike Information Criterion (AIC)

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. As mentioned, we won't get into why the magic 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 Random Slopes model has the smallest AIC (4794.8). 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 and 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.

```
# A complementary step we can take is calculate how much residual variance
# we might have left to  explain.
# In our multilevel model, keep in mind that we have variance at two levels:
# 1. Level One: which are the data-points within subjects.
# 2. Level Two: variation between subjects slopes and intercepts.

# We can get a summary of the random-effects from the "null" model and the
```

```r
# best fitting model by using the VarCorr() function.
VarCorr(raneff_00)
VarCorr(raneff_02)

# These values will help us keep track of how much variance is "available" to
# be explained in our subsequent models. Similarly, these values will inform
us
# how much variance our fixed-effects explain at Level 1 and Level 2 (sort of
# like an R^2 that has been divided into two parts).
```

1.4 Conditional Models

These three models (Random Intercepts, Fixed Slope, and Random Slopes) are our starting models that you will probably end up building in almost every LMER analyses that you run. These models contain valuable information about the variation within and between participants and essentially become the "models to beat" as you add other variables to your model. For instance, the Random Slopes model accounts for how participants change over time but I might be interested in much more than that. (Additionally, sometimes we might also be interested in testing non-linear effects of time and random slopes are not enough! You can model time non-linearly, but it is an advanced topic for the future!) Below are several types of hypotheses you might be interested in testing in your models:

- Is a participant's AIC grade related to their impairment at the begin of therapy? (This hypothesis asks about the effects of a *categorical* variable on the *intercepts*.)
- Is a participant's AIC grade related to their rate of recovery over the course of of therapy? (This hypothesis asks about the effects of a *categorical* variable on the *slopes*.)
- Do older participants show slower improvements over the course of therapy? (This hypothesis asks about the effects of a *continuous* variable on the *slopes*.)

And naturally there are many more hypotheses that you might be interested in testing! In general, however, when we are talking about data with two-levels (i.e., time nested within participants) we are usually interested in the effects that Level 2 variables (i.e., between-participant variables) have on the intercepts (i.e., does this variable affect where someone starts) and on the slopes (i.e., does this variable affect how someone changes over time). In the code below, we will walk through how to build these models using an example of a categorical predictor, AIS grade, to see how this variable affects the intercepts and slopes (but the same procedure could be done with a continuous predictor as well).

To do this, we will first rescale our time variable, Months, into a new variable with a shorter scale, Years. Additionally, we will "center" Years in two different places. One variable, Year.0, will be centered on the first time point (i.e., Year.0 = 0 is the first time point). The other variable, Year.c, is centered on the average time (i.e., Year.c = 0 is the average time in therapy).

```
## --------------- Unconditional and Conditional Models ---------------------
## Creating our Time variable -----------------------------------------------

summary(DATA$month)
# Note the wide range in the month variable (1-18).
# There are two issues with this:
# 1. Time starts at 1 instead of 0
# 2. This scale is a lot larger than other variables we might want to
include.

# To address these issues, we will convert Months to Years and try
"centering"
# the time variable in different locations.

# The first time variable, year0 will set the first assessment equal to 0
DATA$year<-DATA$month/12
DATA$year.0<-DATA$year-min(DATA$year)
summary(DATA$year.0)

# Next, we will create a mean centered time variable, year.c:
mean(DATA$year)
```

```
DATA$year.c<-DATA$year-mean(DATA$year)
summary(DATA$year.c)
```

Let's look at the effects these two different time variables have on the fixed-effects and random-effects.

```
# Time 0 = the first time time point ----
time_00<-lmer(rasch_FIM~
              # Fixed-effects
              1+year.0+
              # Random-effects
              (1+year.0|subID), data=DATA, REML=FALSE)
summary(time_00)

# Time 0 = the mean time of 0.79 years ----
time_01<-lmer(rasch_FIM~
                # Fixed-effects
                1+year.c+
                # Random-effects
                (1+year.c|subID), data=DATA, REML=FALSE)
summary(time_01)
```

Compare the fixed-effects of the intercept and the slope between the two models. Compare the random-effects between the two models.

Centering the time-variable on the first time point makes a lot of sense conceptually, but it is important to remember what that means when interpreting other variables. As you might recall from multiple regression in previous courses, when interaction terms are included in a model, the effect of one variable is being interpreted when the other variable is equal to 0. Thus, if we had year.0 in our model, that would mean the effects of other variables are being evaluated when time = 0, at the initial assessment. Conversely, if we use year.c in our model, that would mean the effect of other variables are being evaluated when time = 0, which is at 0.79 yrs (or 9.5 months).

Let's stick with year.0 in our models and test the effects of AIS grade. First, we will see if AIS grade has an effect on the intercepts (i.e., the beginning of therapy). Second, we will see if AIS grade has an effect on the slopes (i.e., progression through therapy).

```
## Conditional Model: Does the slope change as a function of AIS grade? ----
# With our new variable of time in years, we want to see if we can explain
the
# residual variation in subject's slopes and intercepts.

# Main-Effect Model: Equivalent to Conditional Intercepts
cond_00<-lmer(rasch_FIM~
                # Fixed-effects
                1+year.0+AIS_grade+
                # Random-effects
                (1+year.0|subID), data=DATA, REML=FALSE)
summary(cond_00)
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
approximations to degrees of
  freedom [lmerMod]
Formula: rasch_FIM ~ 1 + year.0 + AIS_grade + (1 + year.0 | subID)
   Data: DATA
```

```
     AIC      BIC   logLik deviance df.resid
  4766.4   4803.0  -2375.2   4750.4      712

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.0007 -0.4949  0.1381  0.6563  2.0919

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subID    (Intercept) 16.87    4.108
          year.0      50.82    7.129    0.09
 Residual             33.77    5.811
Number of obs: 720, groups:  subID, 40

Fixed effects:
                    Estimate Std. Error     df t value Pr(>|t|)
(Intercept)           19.853      1.297 40.240  15.301  < 2e-16 ***
year.0                25.857      1.233 40.000  20.963  < 2e-16 ***
AIS_gradeC5-8          8.128      1.706 40.000   4.765 2.49e-05 ***
AIS_gradeparaplegia   16.427      2.242 40.000   7.328 6.58e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) year.0 AIS_C5
year.0      -0.070
AIS_grdC5-8 -0.757  0.000
AIS_grdprpl -0.576  0.000  0.438
```

```r
# Interaction Model: Conditional Intercepts and Slopes
cond_01<-lmer(rasch_FIM~
                # Fixed-effects
                1+year.0*AIS_grade+
                # Random-effects
                (1+year.0|subID), data=DATA, REML=FALSE)
summary(cond_01)
```

```
Linear mixed model fit by maximum likelihood t-tests use Satterthwaite
approximations to degrees of
  freedom [lmerMod]
Formula: rasch_FIM ~ 1 + year.0 * AIS_grade + (1 + year.0 | subID)
   Data: DATA

     AIC      BIC   logLik deviance df.resid
  4765.2   4811.0  -2372.6   4745.2      710

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.9523 -0.4970  0.1272  0.6515  2.0934

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subID    (Intercept) 16.83    4.103
          year.0      43.36    6.585    0.12
 Residual             33.77    5.811
```

```
Number of obs: 720, groups:  subID, 40

Fixed effects:
                           Estimate Std. Error     df t value Pr(>|t|)
(Intercept)                  20.113      1.302 40.000  15.444  < 2e-16 ***
year.0                       22.348      1.953 40.000  11.443 3.44e-14 ***
AIS_gradeC5-8                 7.779      1.716 40.000   4.532 5.18e-05 ***
AIS_gradeparaplegia          15.890      2.256 40.000   7.044 1.63e-08 ***
year.0:AIS_gradeC5-8          4.714      2.574 40.000   1.831   0.0745 .
year.0:AIS_gradeparaplegia    7.257      3.383 40.000   2.145   0.0381 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) year.0 AIS_C5 AIS_gr y.0:AIS_C
year.0      -0.111
AIS_grdC5-8 -0.759  0.084
AIS_grdprpl -0.577  0.064  0.438
y.0:AIS_C5-  0.084 -0.759 -0.111 -0.049
yr.0:AIS_gr  0.064 -0.577 -0.049 -0.111  0.438
```

Importantly though, before we invest too much in any one of these models, lets make a model comparison using the anova() function to see which model is the better explanation of the data according to the AIC.

```
anova(cond_00,cond_01)
```

```
Data: DATA
Models:
object: rasch_FIM ~ 1 + year.0 + AIS_grade + (1 + year.0 | subID)
..1: rasch_FIM ~ 1 + year.0 * AIS_grade + (1 + year.0 | subID)
       Df    AIC  BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
object  8 4766.4 4803 -2375.2   4750.4
..1    10 4765.2 4811 -2372.6   4745.2 5.2287      2    0.07321 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

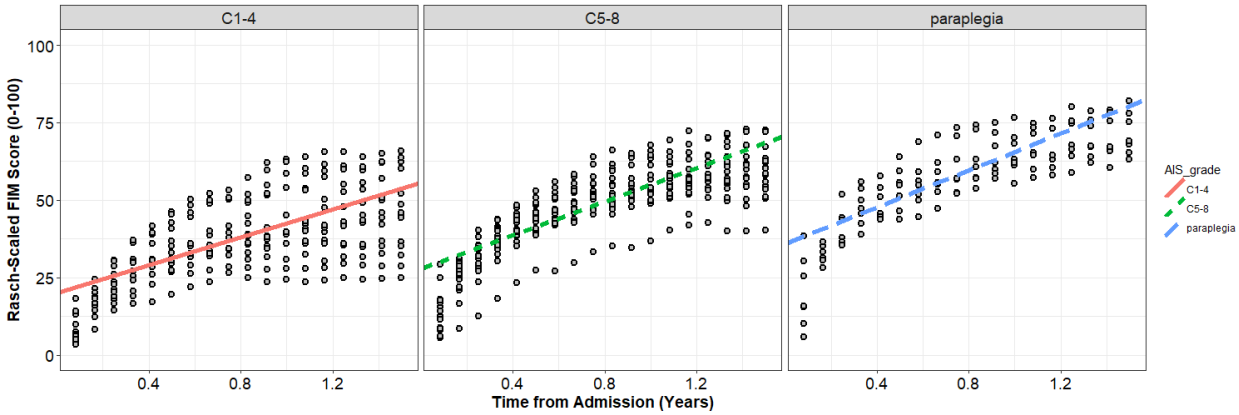
So close! Although the addition of the AIC by Year.0 interaction did improve the model fit slightly, it did not improve our AIC by more than 2 points. Indeed, if we look at the less conservative Wald Test for the change in deviance, we can see that it is also not statistically significant (p = 0.073). Thus, if we were to stop our model building here, we would select “cond_00” as the more parsimonious explanation of the data.

Although the interaction model was not statistically better than the main-effects model, it can useful to present the full model to show the non- significant interactions, especially if those interactions are relevant to theory or clinical practice.

We can also plot the fixed-effects estimates from our models "on top" of the raw data to show the conditional effects.

```
## Linear Fit by AIS Category ----------------------------------------------
summary(cond_01)
FIXED<-data.frame(AIS_grade=c("C1-4","C5-8","paraplegia"),
                  Intercepts=c(20.113, (20.113+7.779), (20.113+15.89)),
```

```
                  Slopes=c(22.348, (22.348+4.714), (22.348+7.257)))
FIXED


g1<-ggplot(DATA, aes(x = year, y = rasch_FIM)) +
  geom_point(fill="grey", pch=21, size=2, stroke=1.25) +
  facet_wrap(~AIS_grade)
g2<-g1+scale_x_continuous(name = "Time from Admission (Years)") +
  scale_y_continuous(name = "Rasch-Scaled FIM Score (0-100)",limits=c(0,100))
g3 <- g2 + 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))

plot(g3)

g4<-g3+geom_abline(aes(intercept=Intercepts, slope=Slopes, col=AIS_grade,
                       lty = AIS_grade), lwd=2, FIXED)
plot(g4)

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



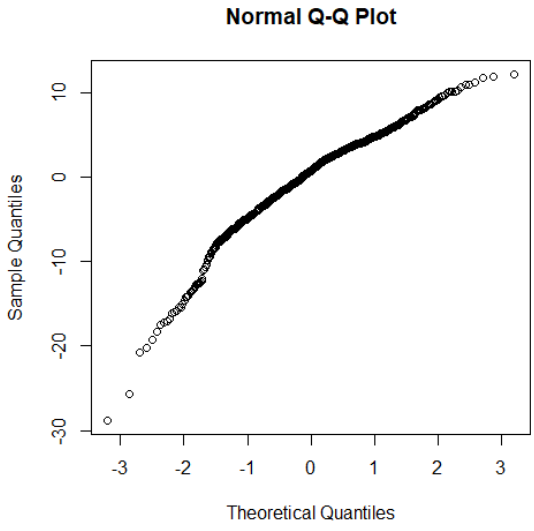
1.5 Checking the Assumptions of our Model

Much like an ordinary least squares regression model, there are a number of diagnostics we want to check to make sure that our statistical inferences are reasonable (e.g., normality) and that our model fits all of the data well (e.g., checking for influential data points). The biggest difference between checking these diagnostics in a simple regression and checking these diagnostics in multi-level model is that we have errors at multiple levels. We have residuals at Level 1 and random-effects at Level 2, but both of these components are a type of error and we want to check each of them for normality, homoskedasticity, and influence.

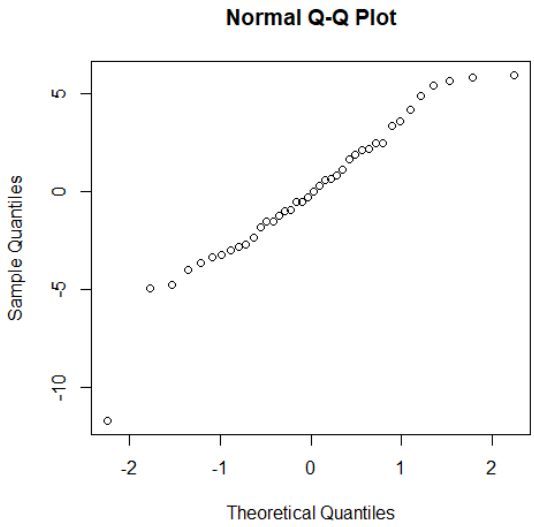
```
# The DATA data frame is what we might call our Level 1 matrix. That is, each
# person and time point is represented. For checking out statistical
assumptions,
# it will also be helpful for us to create a Level 2 matrix. The Level 2
matrix
# will preserve participant information, but ignore time information.

LVL2<-summarize(group_by(DATA, subID),
                  AIS_grade = AIS_grade[1])
LVL2

# At the moment, our LVL2 dataframe contains subject IDs and group
information.
# Next, lets add the random-effects from our models:
LVL2$RE_int<-ranef(cond_01)$subID$`(Intercept)`
LVL2$RE_year<-ranef(cond_01)$subID$year
# Note that we can add this data on directly because the matrix is sorted by
# subID and the the array we are adding is sorted by subID.

LVL2
```

1.5.1 Normality

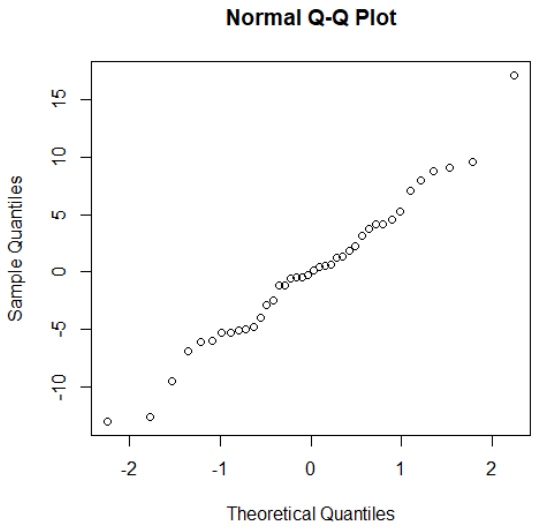
```
# Residuals at Level 1
qqnorm(resid(cond_01))
```




```
# Residuals at Level 2
qqnorm(LVL2$RE_int)
```

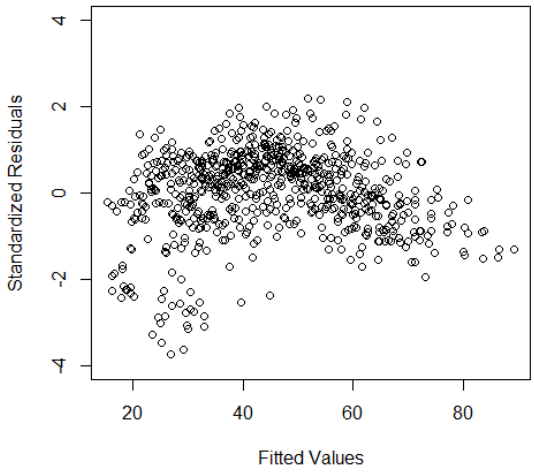



```
qqnorm(LVL2$RE_year)
```

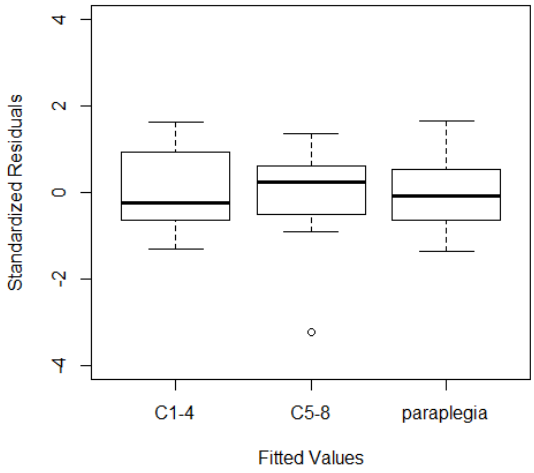


1.5.2 Homoscedasticity

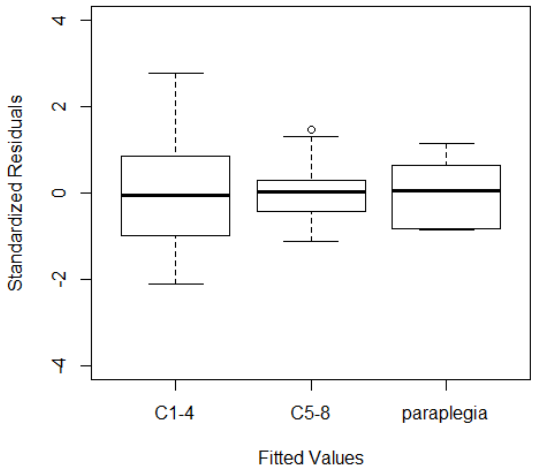
```
# Comparable Variances at Level 1
x <- fitted(cond_01)
y <- resid(cond_01)/sd(resid(cond_01))
plot(x=x, y=y, xlab = "Fitted Values", ylab="Standardized Residuals",
     ylim=c(-4,4))
```




```
# Comparable Variances at Level 2
# Intercepts
LVL2$STD_int<-LVL2$RE_int/sd(LVL2$RE_int)
plot(x=LVL2$AIS_grade, y=LVL2$STD_int,
     xlab = "Fitted Values", ylab="Standardized Residuals",
     ylim=c(-4,4))
```

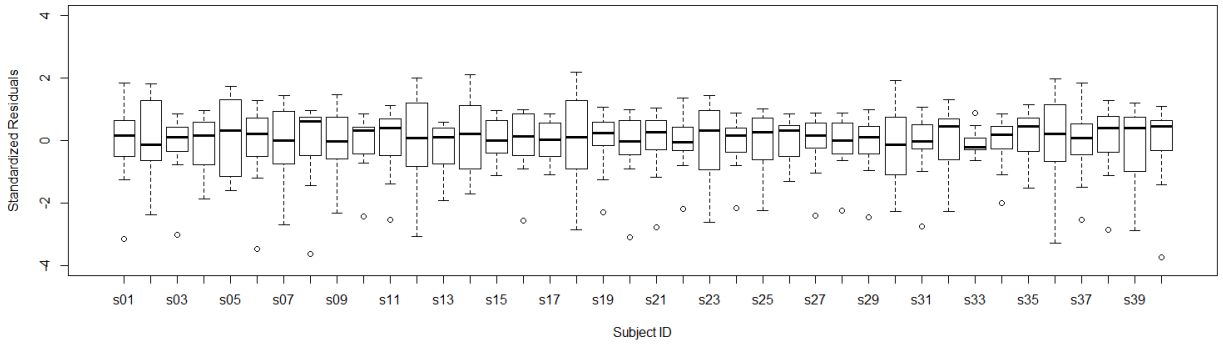



```
# Slopes
LVL2$STD_year<-LVL2$RE_year/sd(LVL2$RE_year)
plot(x=LVL2$AIS_grade, y=LVL2$STD_year,
     xlab = "Fitted Values", ylab="Standardized Residuals",
     ylim=c(-4,4))
```

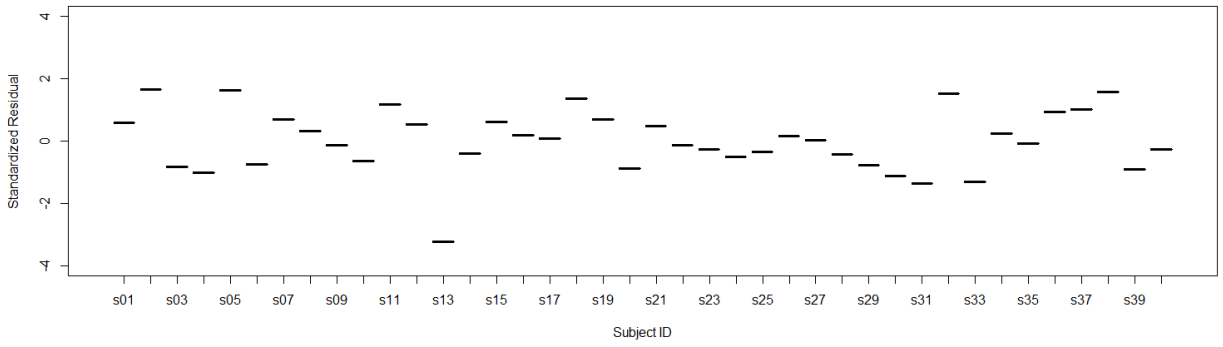


1.5.3 Influential Participants

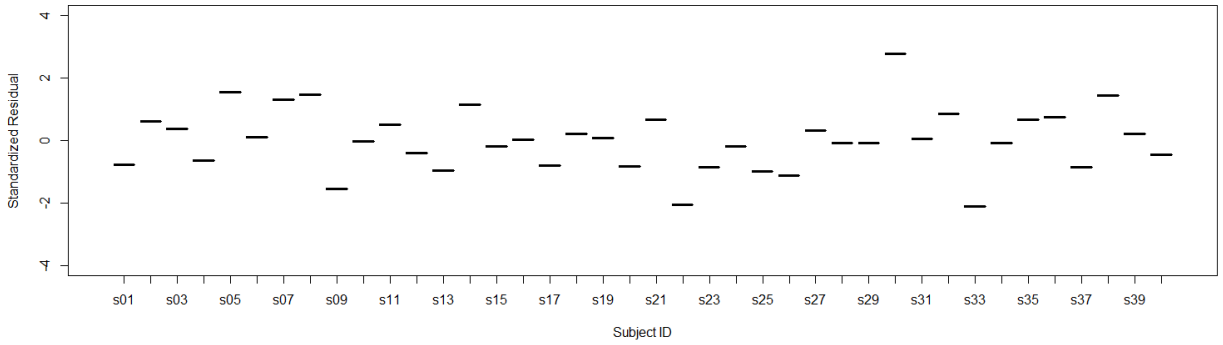
```
# Residuals at Level 1
plot(DATA$subID, resid(cond_03)/sd(resid(cond_03)),
     ylab="Standardized Residuals", xlab="Subject ID",
     ylim=c(-4,4))
```




```
# Residuals at Level 2
# Variation in Intercepts
plot(x=LVL2$subID, y=LVL2$STD_int,
     xlab="Subject ID", ylab="Standardized Residual",
     ylim=c(-4,4))
```




```
# Variation in Slopes
plot(x=LVL2$subID, y=LVL2$STD_year,
     xlab="Subject ID", ylab="Standardized Residual",
     ylim=c(-4,4))
```



Finally, we will save our updated data file with our transformed variables so that we can use it again in the next session.

```
## -------------- Exporting the Revised Data File ----------------------------
# Finally, we want to save our revised data file to include the different
# variables we created in a new .csv file.

write.csv(DATA, "./data_session2.csv")
```