So you have some data that may be clustered. First off, what does clustering mean? It simply means that the variables you care about are grouped into specific categories and that these categories may have unmeasured characteristics that nonetheless affect the outcome of interest. Using linear methods can control for some of this, but not sufficiently so. For example, lets say you have data on student test scores and you want to know whether an intervention you initiated affected those test scores. If you have one class then no problem, you can directly measure the effect of the intervention. But what if you have multiple classrooms, multiple schools, or even multiple districts? The it becomes trickier. Classroom might vary based on a teacher's years of experience, prior training, or just general ability. Schools might differ on resources, the quality of teachers, school culture and so on. All of these are things that might affect student outcomes but they are not 'of' the student, e.g. they cannot be measured as linear to the student outcomes. To be technical about it, they may under estimate the standard errors of the model and therefore affect the statistical significance of results. This is where HLM comes in.

When we conduct HLM we specify the grouping variables in a level-2 model (I will demonstrate below). This level 2 model is called the random effects term. Essentially, random effects are a special form of interaction term. For example, imagine you measured the ACT scores of 500 students, you could assume (1) that every student in a given class has the same score, or (2) that every student in a given class has a specific average score. In the latter case you could simply fit a linear model with the following R code:

```
> lm( score ~ student)
```

HLM follows a similar intuition but, in the student example, instead of fitting one average per student, it would estimate the amount of variation in the average score between each student.

An Example

We can demonstrate the HLM reasoning with some simple R code:

```
#load libraries
> library(lme4)
> library(ggplot2)
> library(reshape2)
> library(lmtest)
```

```
#load the data

mydata<-read.csv(file = "~/some data.csv")


#A simple model

> fit.1<-lmer(score ~ 1 + (1|class), data=mydata)

> summary(fit.1)
```

This model will estimate the average score across all classrooms but also allow the average score to vary between classrooms. We can find the estimated deviation between each class average and the overall average using the ranef function:

```
> ranef(fit.1)
```

This returns the estimated deviation, if we are interest in the average score per class, we add the overall average to the deviations:

```
> score.class<- fixef(fit.1) + ranef(fit.1)$class

> score.class$class<-rownames(score.class)

> names(score.class)[1]<-"Intercept"

> score.class<-score.class[,c(2,1)]


#plot it


> ggplot(score.class, aes(x=class, y=Intercept))+geom_point()+labs(x="Classroom",
y="ACT Score")
```

We end up with something that looks like this which shows us the average intercept for every class as opposed to a single (biased) coefficient for each class that we would then add to the intercept.

A more realistic example

We may have an idea that a single parameter does not do much to explain test outcomes since student's scores may vary based on a number of factors. For example, student level effects may be whether or not they took an exam prep class and what their parent's education level is. Additionally, they may vary by one or more grouping variable, for example, class. We would expect the slope (effect) of these student level effects to vary between class since classes tend to be clustered based on student ability.

```
#fit a new model
> fit.2 <- lmer(score ~ exam.prep.days + poverty + parent.ed + (1+exam.prep.days | class), data=mydata)
> summary(fit.2)
```

```
#put all of the effects (slope and intercept) into a single dataframe
> fit.2.effect<- as.data.frame(t(apply(ranef(fit.2)$class, 1, function(x) fixef(fit.2) + x)))
```

```
#to get the fitted regression line we need to account for the effects of poverty and parental education that we included in our random effects terms by writing the linear equation: Intercept + Slope*Parent.Ed*Poverty with a difference coefficient for each student
```

```
> pred.slope<- melt(apply(fit.2.effect, 1, function(x) x[1] + x[2]*0:500), value.name ="Effect")
names(pred.slope)[1:2]<-c("Days", "Class")
> pred.slope$Days<-pred.slope$Days-1
> pred.slope$Class<-as.factor(pred.slope$Class)
> pred.slp$class<-factor(pred.slp$Class, levels=c(1,2,3,4,5,6,7), labels=c("Class 1", "Class 2", "Class 3", "Class 4", "Class 5", "Class 6", "Class 7"))
```

```
#Check the data frame
> head(pred.slope)
```

```
  Days  Class      Effect
1   1   Class 1    97.93248
2   2   Class 2    98.58646
3   3   Class 3    99.24044
4   4   Class 4    99.89442
5   5   Class 5   100.54840
6   6   Class 6   101.20238
```

```
#plot the outcomes
> ggplot(pred.slope, aes(x=Days, y=Effect, color=Class))+geom_smooth(method="lm",
aes(linetype=Class))+labs(x="Days of Exam Prep", y="ACT Score")
```

We now can see the average effect of Days of Exam Prep across all students in each Classroom on the average ACT scores for students in that class. This allows us to make interpretations about the classes and potentially the quality of instruction in an unbiased fashion.

If we wanted to see the average effect in descriptive terms, we could plot directly from the data we see a much less informative representation of scores:

```
> Class<-as.factor(mydata$class)
> data<-data.frame(mydata, Class)
> ggplot(data, aes(x=exam.prep.days, y=score, color=Class))+geom_smooth(method="lm",
se=FALSE, aes(linetype=Class))+labs(x="Days of Exam Prep", y="Score")
```

If we wanted to write up the results, we could simply look at the summary, knowing that the effect of Days of Exam Prep on the outcome varies by as much as the Random Effects term, but if we wanted to look at specific estimates of days of exam prep within any given classroom, we can simply examine the pred.slope dataframe since it includes the effects of Days of Exam Prep within each class. This gives us the effect of Class on all fixed effects within the model which can be useful for interpreting differences between classes rather than simply assuming that all students have an average score that is directly affected by both days of exam prep and class.