

# Intro to multilevel modeling in R

Rose Hartman

Office of the Vice President for Research and Innovation  
Center for Assessment Statistics & Evaluation

May 19th, 2016

# Workshop Overview

## 1 MLM overview

- What is MLM?
- The lme4 package

## 2 Running models

- Prepping your data for MLM
- Example: Nested data

## 3 Plotting

# This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind MLM more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!
- We'll be using primarily dplyr and tidyr for data wrangling, and ggplot2 for plotting
- Color-coded content, to help you keep track of the most important pieces

# This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind MLM more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!
- We'll be using primarily dplyr and tidyr for data wrangling, and ggplot2 for plotting
- Color-coded content, to help you keep track of the most important pieces

# This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind MLM more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!
- We'll be using primarily dplyr and tidyr for data wrangling, and ggplot2 for plotting
- Color-coded content, to help you keep track of the most important pieces

# This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind MLM more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!
- We'll be using primarily dplyr and tidyr for data wrangling, and ggplot2 for plotting
- Color-coded content, to help you keep track of the most important pieces

Keep an eye out for...

How would you do this?

What will this do?

Keep an eye out for...

**How would you do this?**

What will this do?



Keep an eye out for...

How would you do this?

What will this do?

Keep an eye out for...

**How would you do this?**

**What will this do?**

**Learn more:** resources to check out

Keep an eye out for...

**How would you do this?**

**What will this do?**

**Learn more:** resources to check out

**Key idea:** the big ideas you need to hold on to

Keep an eye out for...

**How would you do this?**

**What will this do?**

**Learn more:** resources to check out

**Key idea:** the big ideas you need to hold on to

**See also:** other functions or packages that do a similar thing

If you don't already have dplyr, tidyr, haven, and ggplot2 installed, do that now:

```
install.packages("ggplot2")

## Installing package into '/Users/TARDIS/Library/R/3.3/library'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN
## without setting a mirror

install.packages("haven")

## Installing package into '/Users/TARDIS/Library/R/3.3/library'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN
## without setting a mirror

install.packages("dplyr")

## Installing package into '/Users/TARDIS/Library/R/3.3/library'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN
```

# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet

# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet

# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet



# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet

# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet

# What is multilevel modeling?

First, a note on terminology. The statistical technique we'll be discussing today goes by many names:

- multilevel modeling (MLM)
- hierarchical linear modeling (HLM)
- mixed effects modeling
- fixed and random effects modeling
- nested effects models
- probably more I haven't come across yet

# When should you consider using multilevel modeling?

- Nested data
  - Repeated measures / longitudinal data
  - Random effects (when you want to generalize from a sample to a population, for more than just the subjects in your model)

# When should you consider using multilevel modeling?

- Nested data
- Repeated measures / longitudinal data
- Random effects (when you want to generalize from a sample to a population, for more than just the subjects in your model)

# When should you consider using multilevel modeling?

- Nested data
- Repeated measures / longitudinal data
- Random effects (when you want to generalize from a sample to a population, for more than just the subjects in your model)

# Important caveat

Multilevel modeling is powerful and flexible and cool, and really widely applicable. That doesn't mean it's the only right way to analyze these kinds of data. There are often many acceptable analysis techniques for a given problem, each with different strengths and weaknesses depending on your situation.

# Overview

## 1 MLM overview

- What is MLM?
- The lme4 package

## 2 Running models

- Prepping your data for MLM
- Example: Nested data

## 3 Plotting



# Install the lme4 package

```
install.packages("lme4")

## Installing package into '/Users/TARDIS/Library/R/3.3/library'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN
## without setting a mirror

library("lme4")

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
```

# Install the lme4 package

```
install.packages("lme4")

## Installing package into '/Users/TARDIS/Library/R/3.3/library'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN
## without setting a mirror

library("lme4")

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
```

Remember...

- When you run `install.packages()`, R contacts CRAN to get the package you want, so you need an internet connection for it

# A note about packages

Note the message you get when you load lme4:

The following object is masked from 'package:tidyr':  
expand

# Hi, lme4

Let's get to know our new friend.

```
help(package="lme4")
```

# Hi, lme4

Let's get to know our new friend.

```
help(package="lme4")
```

**See also:** nlme package. For a great comparison of differences between the two packages, see the help page for the lme4 package.

What makes running a multilevel model different from regular regression models?

- You can model what would otherwise be non-independent observations. Yay!
- You have several different sources of error variance to think about (and covariances among them!)
- There is no longer an analytic solution — the computer has to brute force a solution by trying millions of combinations of parameter estimates and seeing what yields the best fit. The lme4 package offers a few different options for what estimation procedure to use. It defaults to REstricted Maximum Likelihood (REML), which I recommend.

What makes running a multilevel model different from regular regression models?

- You can model what would otherwise be non-independent observations. Yay!
- You have several different sources of error variance to think about (and covariances among them!)
- There is no longer an analytic solution — the computer has to brute force a solution by trying millions of combinations of parameter estimates and seeing what yields the best fit. The lme4 package offers a few different options for what estimation procedure to use. It defaults to REstricted Maximum Likelihood (REML), which I recommend.

What makes running a multilevel model different from regular regression models?

- You can model what would otherwise be non-independent observations. Yay!
- You have several different sources of error variance to think about (and covariances among them!)
- There is no longer an analytic solution — the computer has to brute force a solution by trying millions of combinations of parameter estimates and seeing what yields the best fit. The lme4 package offers a few different options for what estimation procedure to use. It defaults to REstricted Maximum Likelihood (REML), which I recommend.



# Overview

- 1 MLM overview
  - What is MLM?
  - The lme4 package
- 2 Running models
  - Prepping your data for MLM
  - Example: Nested data
- 3 Plotting

# Getting your data into R

Check your working directory:

```
getwd()  
  
## [1] "/Users/TARDIS/Documents/CASE/Workshop_IntroR"
```

If you want R to find something on your computer, you have three options:

- 1 Put the file in R's working directory
- 2 Move R's working directory to where ever the file is saved using `setwd()`
- 3 Specify the file path when you tell R to look for the file

# Getting your data into R

Check your working directory:

```
getwd()  
  
## [1] "/Users/TARDIS/Documents/CASE/Workshop_IntroR"
```

If you want R to find something on your computer, you have three options:

- 1 Put the file in R's working directory
- 2 Move R's working directory to where ever the file is saved using `setwd()`
- 3 Specify the file path when you tell R to look for the file

# Getting your data into R

I'll use option 3, specifying the file path for the file when I tell R to read it in. Find the file on your computer, get its location, and add that file path to your `read_sav` command:

```
atlas <- read_sav("data/ATLAS.sav")
```

# Getting your data into R

I'll use option 3, specifying the file path for the file when I tell R to read it in. Find the file on your computer, get its location, and add that file path to your `read_sav` command:

```
atlas <- read_sav("data/ATLAS.sav")
```

# Formatting your data

Your data need to be in "long" format, where each row is just one observation (i.e., one participant). If you have repeated measures data, then each row should be one observation (i.e., one timepoint), not one participant.

## How would you do this?

Transform a dataframe from wide to long using dplyr/tidyr commands.

# Formatting your data

Your data need to be in "long" format, where each row is just one observation (i.e., one participant). If you have repeated measures data, then each row should be one observation (i.e., one timepoint), not one participant.

## How would you do this?

Transform a dataframe from wide to long using dplyr/tidyr commands.

```
?gather
```

# Formatting your data

Also, for nested data, make sure your grouping variable is a factor:

```
str(atlas)

## Classes 'tbl_df', 'tbl' and 'data.frame': 1226 obs. of  15 variables
## $ schoolid      : atomic  1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "label")= chr "School ID"
## $ grade         :Class 'labelled'  atomic [1:1226] 12 12 12 12 12 12
## .. ..- attr(*, "label")= chr "Student grade (9-12)"
## .. ..- attr(*, "labels")= Named num [1:4] 9 10 11 12
## .. ..- attr(*, "names")= chr [1:4] "Frosh" "Soph" "Junior" "Sen
## $ stuID         : atomic  1482 1492 1493 1494 1500 ...
## ..- attr(*, "label")= chr "Student ID"
## $ intervention:Class 'labelled'  atomic [1:1226] 0 0 0 0 0 0 0 0 0
## .. ..- attr(*, "label")= chr "School intervention?"
## .. ..- attr(*, "labels")= Named num [1:2] 0 1
## .. ..- attr(*, "names")= chr [1:2] "Control" "Intervention"
## $ STSE0         : atomic   5.83 7 6 6.83 5.67 ...
## ..- attr(*, "label")= chr "Strength training self-efficacy: pretes
## $ STSE1         : atomic   6.17 7 6.5 7 5.33 ...
## ..- attr(*, "label")= chr "Strength training self-efficacy: posttes
```



# Formatting your data

Also, for nested data, make sure your grouping variable is a factor:

```
str(atlas)

## Classes 'tbl_df', 'tbl' and 'data.frame': 1226 obs. of  15 variables
## $ schoolid      : atomic  1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "label")= chr "School ID"
## $ grade         :Class 'labelled'  atomic [1:1226] 12 12 12 12 12 12
## .. ..- attr(*, "label")= chr "Student grade (9-12)"
## .. ..- attr(*, "labels")= Named num [1:4] 9 10 11 12
## .. ..- attr(*, "names")= chr [1:4] "Frosh" "Soph" "Junior" "Sen
## $ stuID         : atomic  1482 1492 1493 1494 1500 ...
## ..- attr(*, "label")= chr "Student ID"
## $ intervention:Class 'labelled'  atomic [1:1226] 0 0 0 0 0 0 0 0 0
## .. ..- attr(*, "label")= chr "School intervention?"
## .. ..- attr(*, "labels")= Named num [1:2] 0 1
## .. ..- attr(*, "names")= chr [1:2] "Control" "Intervention"
## $ STSE0         : atomic  5.83 7 6 6.83 5.67 ...
## ..- attr(*, "label")= chr "Strength training self-efficacy: pretes
## $ STSE1         : atomic  6.17 7 6.5 7 5.33 ...
## ..- attr(*, "label")= chr "Strength training self-efficacy: posttes
```

# Centering

Why center your variables?

- If you have interactions between continuous predictors, not centering can produce multicollinearity :(

# Centering

Why center your variables?

- If you have interactions between continuous predictors, not centering can produce multicollinearity :(
- Centering often makes interpretation easier, if 0 isn't an interesting/meaningful value on your scales.

# Centering

Why center your variables?

- If you have interactions between continuous predictors, not centering can produce multicollinearity :(
- Centering often makes interpretation easier, if 0 isn't an interesting/meaningful value on your scales.
- Sometimes it can make your model converge a little faster, since the intercept won't jump around as much iteration to iteration.

# Centering

## Why center your variables?

- If you have interactions between continuous predictors, not centering can produce multicollinearity :(
- Centering often makes interpretation easier, if 0 isn't an interesting/meaningful value on your scales.
- Sometimes it can make your model converge a little faster, since the intercept won't jump around as much iteration to iteration.
- It doesn't hurt anything, so might as well!

# Centering

## Why center your variables?

- If you have interactions between continuous predictors, not centering can produce multicollinearity :(
- Centering often makes interpretation easier, if 0 isn't an interesting/meaningful value on your scales.
- Sometimes it can make your model converge a little faster, since the intercept won't jump around as much iteration to iteration.
- It doesn't hurt anything, so might as well!
- Centering only makes sense for continuous variables. Be careful to never try to center a categorical predictor.

# Centering

```
atlas$STSE0c <- atlas$STSE0 - mean(atlas$STSE0, na.rm=TRUE)
```

# Subset analysis

One lovely thing about the `lmer()` function is that it includes a subset argument — that means if you only want to run the model on some of your observations and not others, you don't have to save a new data frame for that, you can just sepcify the subset right in your `lmer()` command!



# Subset analysis

One lovely thing about the `lmer()` function is that it includes a subset argument — that means if you only want to run the model on some of your observations and not others, you don't have to save a new data frame for that, you can just sepcify the subset right in your `lmer()` command!

By default, it will use all of the observations in the data frame.

# Overview

- 1 MLM overview
  - What is MLM?
  - The lme4 package
- 2 Running models
  - Prepping your data for MLM
  - Example: Nested data
- 3 Plotting

## Quick review: `lm`

Let's say you want to model the effect of the intervention (intervention) on post-test strength-training self-efficacy (STSE1), controlling for pretest strength-training self-efficacy (STSE0). Allow for the fact that the effect of pretest scores on posttest scores may be different for students who did vs. didn't get the intervention.

**How would you do this?** Write the formula for that model in `lm()`

## Quick review: lm

Let's say you want to model the effect of the intervention (intervention) on post-test strength-training self-efficacy (STSE1), controlling for pretest strength-training self-efficacy (STSE0). Allow for the fact that the effect of pretest scores on posttest scores may be different for students who did vs. didn't get the intervention.

**How would you do this? Write the formula for that model in lm()**

```
lm(STSE1 ~ intervention * STSE0c, data=atlas,  
    na.action = na.exclude)  
  
##  
## Call:  
## lm(formula = STSE1 ~ intervention * STSE0c, data = atlas, na.action  
##  
## Coefficients:  
##              (Intercept)              intervention  
##              STSE0c              intervention:STSE0c
```

## Quick review: lm

Let's say you want to model the effect of the intervention (intervention) on post-test strength-training self-efficacy (STSE1), controlling for pretest strength-training self-efficacy (STSE0). Allow for the fact that the effect of pretest scores on posttest scores may be different for students who did vs. didn't get the intervention.

**How would you do this? Write the formula for that model in lm()**

```
lm(STSE1 ~ intervention * STSE0c, data=atlas,  
    na.action = na.exclude)  
  
##  
## Call:  
## lm(formula = STSE1 ~ intervention * STSE0c, data = atlas, na.action  
##  
## Coefficients:  
##              (Intercept)              interventionyes
```

# Null model for ICC

```
model0 <- lmer(STSE1 ~ 1 +  
               (1|schoolid),  
               data = atlas,  
               na.action = na.exclude)  
  
# the variance components  
varcom <- as.data.frame(VarCorr(model0))
```

# Null model for ICC

```
model0 <- lmer(STSE1 ~ 1 +  
               (1|schoolid),  
               data = atlas,  
               na.action = na.exclude)  
  
# the variance components  
varcom <- as.data.frame(VarCorr(model0))
```

```
L2var <- varcom$vcov[1]  
L1var <- varcom$vcov[2]  
  
icc <- L2var/(L2var+L1var)
```

# Model 1: Random intercepts

First, let's just specify a random intercepts model.



# Model 1: Random intercepts

First, let's just specify a random intercepts model.

```
model1 <- lmer(STSE1 ~ intervention * STSE0c +  
                (1|schoolid),  
                data = atlas,  
                na.action = na.exclude)
```

# Model 1: Random intercepts

First, let's just specify a random intercepts model.

```
model1 <- lmer(STSE1 ~ intervention * STSE0c +  
               (1|schoolid),  
               data = atlas,  
               na.action = na.exclude)
```

The random effects are the last term in the model, in parentheses. In this case, we have one random factor, schoolid. In the parentheses before the fence —, you write out all of the effects in your model that you want to vary by that random factor. In this case, it's just a 1, representing the intercept.

# Model 1: Random intercepts

First, let's just specify a random intercepts model.

```
model1 <- lmer(STSE1 ~ intervention * STSE0c +  
               (1|schoolid),  
               data = atlas,  
               na.action = na.exclude)
```

The random effects are the last term in the model, in parentheses. In this case, we have one random factor, schoolid. In the parentheses before the fence —, you write out all of the effects in your model that you want to vary by that random factor. In this case, it's just a 1, representing the intercept.

**Learn more:** A great answer on stackexchange:  
<http://stats.stackexchange.com/questions/31569/questions-about-how-random-effects-are-specified-in-lme>

# Model 1: Random intercepts

```
summary(model1)

## Linear mixed model fit by REML ['lmerMod']
## Formula: STSE1 ~ intervention * STSE0c + (1 | schoolid)
## Data: atlas
##
## REML criterion at convergence: 3356.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -5.4987 -0.5511  0.2180  0.6486  2.3341
##
## Random effects:
##  Groups   Name      Variance Std.Dev.
## schoolid (Intercept) 0.05027  0.2242
## Residual                0.92150  0.9599
## Number of obs: 1201, groups: schoolid, 31
##
## Fixed effects:
```

	Estimate	Std. Error	t value
--	----------	------------	---------

# Model 1: Random intercepts

If you load the `lmerTest` package, it will add information about significance tests to the `lmer` object, using Satterthwaite's approximations for df.

```
library(lmerTest)

##
## Attaching package: 'lmerTest'
## The following object is masked from 'package:lme4':
##
##      lmer
## The following object is masked from 'package:stats':
##
##      step

model1 <- lmer(STSE1 ~ intervention * STSE0c +
               (1|schoolid),
               data = atlas,
               na.action = na.exclude)

summary(model1)
```

## Model 2: Adding random slopes

When you have more than one random component, you can also estimate the covariance(s) between them, or you can restrict the covariances to be 0.

## Model 2: Adding random slopes

When you have more than one random component, you can also estimate the covariance(s) between them, or you can restrict the covariances to be 0.

This estimates the covariance between slope for STSE0c and the intercept:

```
model2 <- lmer(STSE1 ~ intervention * STSE0c +  
               (1 + STSE0c|schoolid),  
               data = atlas,  
               na.action = na.exclude)
```

```
summary(model2)
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximat  
## to degrees of freedom [lmerMod]  
## Formula: STSE1 ~ intervention * STSE0c + (1 + STSE0c | schoolid)  
## Data: atlas  
##  
## REML criterion at convergence: 3355.8  
##
```

## Model 2: Adding random slopes

And this constrains the covariance to be 0:

```
model2.nocov <- lmer(STSE1 ~ intervention * STSE0c +  
                      (1|schoolid) + (0 + STSE0c|schoolid),  
                      data = atlas,  
                      na.action = na.exclude)  
  
summary(model2.nocov)  
  
## Linear mixed model fit by REML t-tests use Satterthwaite approximat  
##   to degrees of freedom [lmerMod]  
## Formula: STSE1 ~ intervention * STSE0c + (1 | schoolid) + (0 + STSE0  
##       schoolid)  
##   Data: atlas  
##  
## REML criterion at convergence: 3356.1  
##  
## Scaled residuals:  
##      Min      1Q  Median      3Q      Max  
## -5.4568 -0.5443  0.2227  0.6473  2.3551  
##
```



## Model 2: Adding random slopes

And this constrains the covariance to be 0:

```
model2.nocov <- lmer(STSE1 ~ intervention * STSE0c +  
                      (1|schoolid) + (0 + STSE0c|schoolid),  
                      data = atlas,  
                      na.action = na.exclude)  
  
summary(model2.nocov)  
  
## Linear mixed model fit by REML t-tests use Satterthwaite approximat  
##   to degrees of freedom [lmerMod]  
## Formula: STSE1 ~ intervention * STSE0c + (1 | schoolid) + (0 + STSE0  
##       schoolid)  
##   Data: atlas  
##  
## REML criterion at convergence: 3356.1  
##  
## Scaled residuals:  
##      Min      1Q  Median      3Q      Max  
## -5.4568 -0.5443  0.2227  0.6473  2.3551  
##
```

## Model 2: Adding random slopes

```
anova(model1, model2)

## refitting model(s) with ML (instead of REML)

## Data: atlas
## Models:
## object: STSE1 ~ intervention * STSE0c + (1 | schoolid)
## ..1: STSE1 ~ intervention * STSE0c + (1 + STSE0c | schoolid)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object 6 3352.2 3382.7 -1670.1  3340.2
## ..1     8 3355.6 3396.4 -1669.8  3339.6 0.5508      2      0.7593
```

## Model 2: Adding random slopes

```
anova(model1, model2)

## refitting model(s) with ML (instead of REML)

## Data: atlas
## Models:
## object: STSE1 ~ intervention * STSE0c + (1 | schoolid)
## ..1: STSE1 ~ intervention * STSE0c + (1 + STSE0c | schoolid)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  6 3352.2 3382.7 -1670.1   3340.2
## ..1     8 3355.6 3396.4 -1669.8   3339.6 0.5508      2    0.7593
```

Note that the recommendation from the folks who wrote lme4 is to only use deviance change tests like this when the models have all of the same fixed effects (just the random effects changing).

## Model 3: Adding MORE random slopes

```
model3 <- lmer(STSE1 ~ intervention * STSE0c +  
               (1 + intervention * STSE0c | schoolid),  
               data = atlas,  
               na.action = na.exclude)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =  
control$checkConv, : unable to evaluate scaled gradient  
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =  
control$checkConv, : Model failed to converge: degenerate  
Hessian with 2 negative eigenvalues
```

# Printing model tables

The pander package can't handle lme4 objects, unfortunately.

# Printing model tables

The pander package can't handle lme4 objects, unfortunately. But texreg can! It won't work with knitting directly to word, but there's a decent work around:

```
htmlreg(model2, file="model2_table.doc") # one model

## Error in eval(expr, envir, enclos): could not find function
"htmlreg"

htmlreg(list(model1, model2), file="model1and2_table.doc") # compare two

## Error in eval(expr, envir, enclos): could not find function
"htmlreg"
```

And then you can copy the table right into your word doc.

# Overview

- 1 MLM overview
  - What is MLM?
  - The lme4 package
- 2 Running models
  - Prepping your data for MLM
  - Example: Nested data
- 3 Plotting

# Plotting

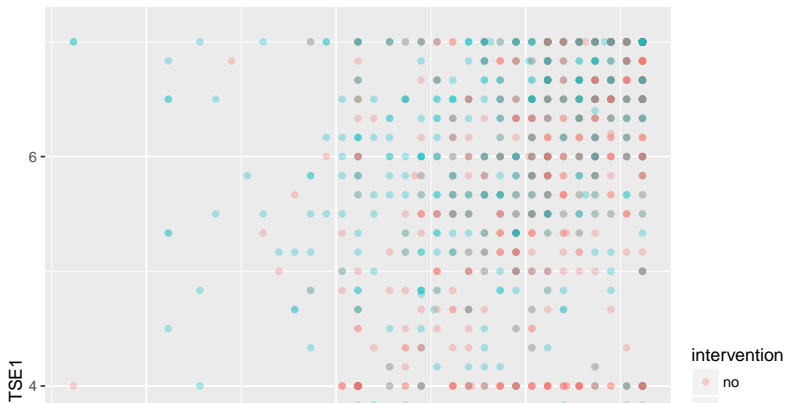
```
atlas$pred1 <- predict(model1)  
atlas$pred2 <- predict(model2)
```



# Plotting

```
ggplot(atlas, aes(y=STSE1, x=STSE0c, color=intervention)) +  
  geom_point(alpha=.3)
```

## Warning: Removed 25 rows containing missing values  
(geom\_point).



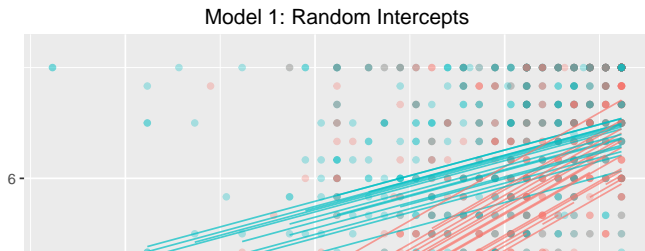
# Plotting

Add lines from the models:

```
ggplot(atlas, aes(y=STSE1, x=STSE0c, color=intervention)) +  
  geom_point(alpha=.3) +  
  geom_line(aes(y=pred1, group=schoolid), alpha=.7) +  
  labs(title="Model 1: Random Intercepts")
```

## Warning: Removed 25 rows containing missing values  
(geom\_point).

## Warning: Removed 17 rows containing missing values  
(geom\_path).



# Plotting

Add lines from the models:

```
ggplot(atlas, aes(y=STSE1, x=STSE0c, color=intervention)) +  
  geom_point(alpha=.3) +  
  geom_line(aes(y=pred1, group=schoolid), alpha=.7) +  
  labs(title="Model 1: Random Intercepts")
```

## Warning: Removed 25 rows containing missing values  
(geom\_point).

## Warning: Removed 17 rows containing missing values  
(geom\_path).

