

# ML MODELS

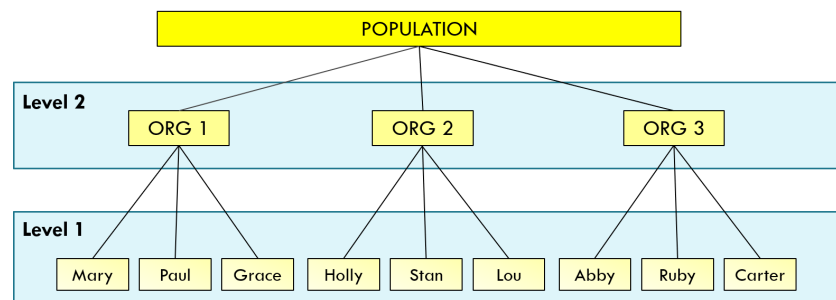
Research Methods in Psychology I & II ■ Department of Psychology ■ Colorado State University

## BY THE END OF THIS UNIT YOU WILL:

1. Know about hierarchical or nested designs.
2. Have a sense of how multilevel variables vary at multiple levels.
3. Understand the role of fixed and random effects in multilevel models.
4. Know how to handle two common types of nesting in behavioral sciences: people nested in groups, and repeated measures nested in person.
5. Know how to fit simple multilevel models using the lmer package in R.
6. Know how to interpret simple multilevel models.
7. Know how to plot the results of simple multilevel models using ggplot.

## What is a multilevel model?

A multilevel model (also commonly called a random effects model, a mixed effects model, and a hierarchical linear model) is used to model hierarchical data. In data that arise from a hierarchical design, the upper level units (e.g., organizations) are selected from the population. Then, cases (e.g., employees) are selected from within these upper level units. In this way, employees (Level 1 of the hierarchy) are nested in organizations (Level 2 of the hierarchy).



**In behavioral science, hierarchical designs are very common:**

**Employees nested in companies**

**Students nested in schools**

**Siblings nested in families**

**Partners nested in romantic couples (dyad)**

**Patients nested in therapists**

**Peers nested in friendship groups**

**Multilevel modeling is a complex topic, and requires much more than a few sessions to master. In this unit, you will receive an introduction to multilevel modeling, and hopefully become interested in learning more in a future class or through the many wonderful books on the subject.**

## Example Dataset Description — Promoting Participation in Data Science Careers

A research team at a large University sought to determine if an 8 week summer program designed to encourage female high school students to pursue Data Science education and career paths was more effective if the program was team-focused vs. individual-focused. 500 females who were recruited to participate in the program completed an application packet that included a high school transcript, an online assessment of their current skills in math and computer science, and psychological assessments of self-efficacy for STEM disciplines. Using these data, the researchers created an index that binned the females into quintiles based on the likelihood of success in the summer program, the index ranged from 0 to 4, where 0 designated the highest likelihood of success and 4 designated the lowest likelihood of success. Once this “risk index” was created, one female from each quintile was randomly assigned to a team of 5 students such that each team had one female who had a risk index of 0, one who had a risk index of 1, and so forth. This created a total of 100 teams, each with 5 team members. Next, each team was randomly assigned to participate in either a team-focused version of the summer program, or an individual-focused version of the summer program. Following random assignment, the teams participated in the 8 week program. Throughout the program, a series of measures and assessments were collected. The data are in a file called `mlm_teams.csv`, and below is a summary of the variables.

**team\_id:** The team number, values of 1 to 50 denote team-focused teams (the treatment condition), and values of 51 to 60 denote individual-focused teams (the control condition).

**kid\_id:** The personal ID number of the student.

**txcond:** Condition indicator (0 = control condition, 1 = treatment condition).

**risk:** The student’s risk quintile (ranges from 0 to 4, where 0 = lowest risk quintile, 4 = highest risk quintile)

**score:** The student’s score on a final comprehensive exam to measure knowledge gained during the summer program. It ranges from 0 to 100, where a higher score denotes more knowledge.

**comafrd:** At the end of week 4, all students completed a measure of belief in the communal affordances of a Data Science career (i.e., that a career in Data Science would afford the opportunity to reach communal goals, such as having a positive impact on society, developing close relationships with co-workers, altruism). The scale ranged from 1 to 9, where a higher score denoted a stronger belief that a career in Data Science would allow for communal goals to be met.

## Summary of Data Structure

### Level 2: Teams

team\_id 1

...

team\_id 100

Variables = txcond

kid\_id1, kid\_id2...kid\_id5

kid\_id1, kid\_id2...kid\_id5

Variables = risk, comafrd, score

### Level 1: Students

## **Prepare Data and Get Descriptive Statistics**

To begin, please set up a new notebook called `MultilevelModels_TeamsNotebook`, and save it in your `MyClassActivities` folder.

Load libraries

```
library(tidyverse)
library(broom)
library(modelr)
library(lme4)
```

Import data

```
teams <- read_csv("mlm_teams.csv")
```

Format data and create subsets by treatment condition

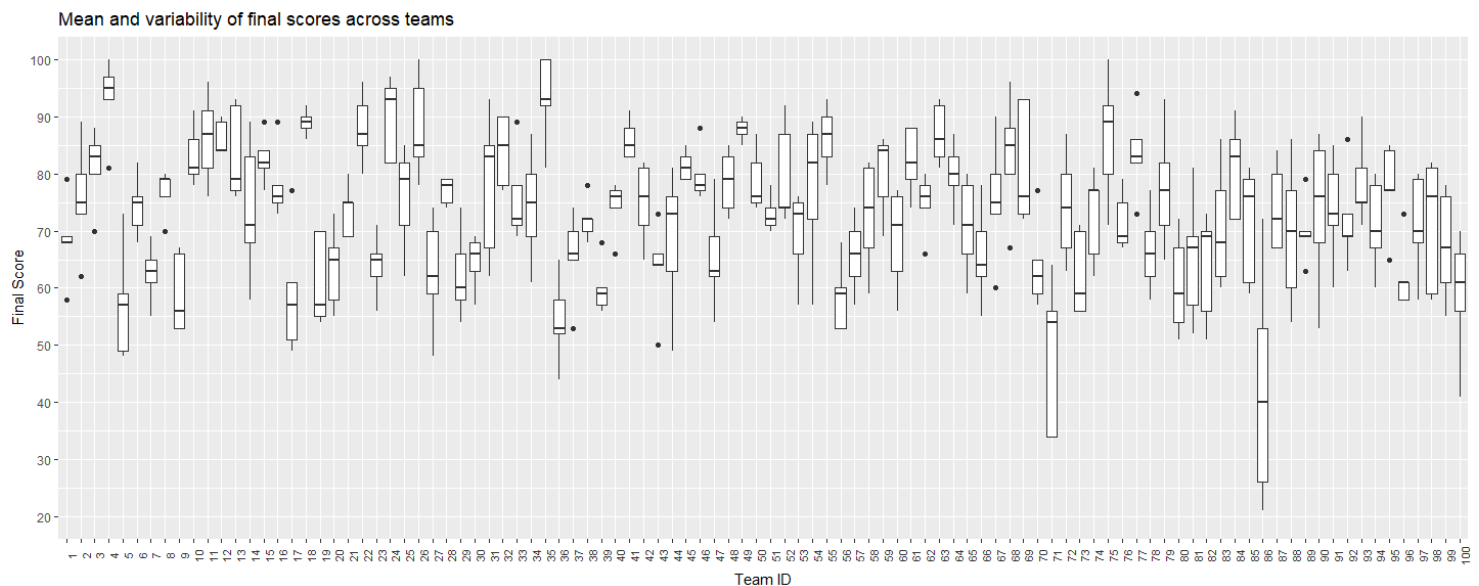
```
teams <- teams %>%
  mutate(txcond.f = factor(txcond, levels = c(0,1), labels = c("control", "treatment")),
         team_id.f = factor(team_id))

c <- filter(teams, txcond == 0)
t <- filter(teams, txcond == 1)
```

## Let's Begin by Examining the Variability of Final Test Scores

Create a plot of scores by team

```
ggplot(data = teams, aes(x = team_id.f, y = score)) +
  geom_boxplot() +
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +
  labs(title = "Mean and variability of final scores across teams", x = "Team ID", y = "Final Score") +
  theme(axis.text.x = element_text(colour="grey20", size=8, angle=90, hjust=.5))
```



This plot presents a boxplot for each team, we can see substantial variability in the final test score both within teams (the height of each box) as well as between teams.

Calculate a mean for each team, and calculate the mean of all team means

```
team_means <- teams %>%
  group_by(team_id) %>%
  summarize(mean_score = mean(score))
team_means

meanofmeans <- team_means %>%
  summarize(meanofmeans = mean(mean_score))
meanofmeans
```

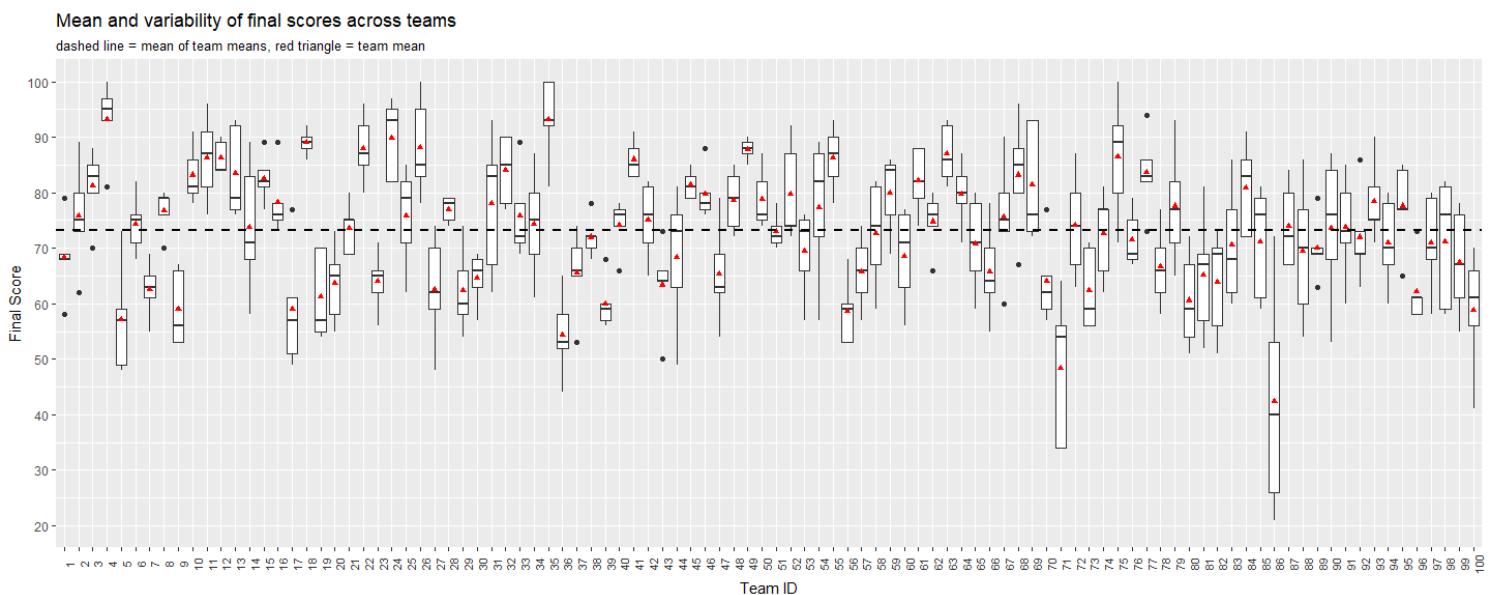
team_id	mean_score
1	68.4
2	75.8
3	81.2
4	93.2
5	57.2
6	74.4
7	62.6
8	76.8
9	59.0
10	83.2

meanofmeans  
<dbl>  
73.216

## An Enhanced Plot

Enhance prior plot with the teams means and the mean of means

```
ggplot(data = teams, aes(x = team_id.f, y = score)) +
  geom_boxplot() +
  stat_summary(aes(y = score, group = team_id.f), fun.y = mean, color = "red", geom = "point", pch = 17, size = 1.5) +
  geom_hline(yintercept = meanofmeans$meanofmeans, linetype="dashed", color = "black", size = 1) +
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +
  labs(title = "Mean and variability of final scores across teams",
       subtitle = "dashed line = mean of team means, red triangle = team mean",
       x = "Team ID", y = "Final Score") +
  theme(axis.text.x = element_text(colour="grey20", size=8, angle=90, hjust=.5))
```



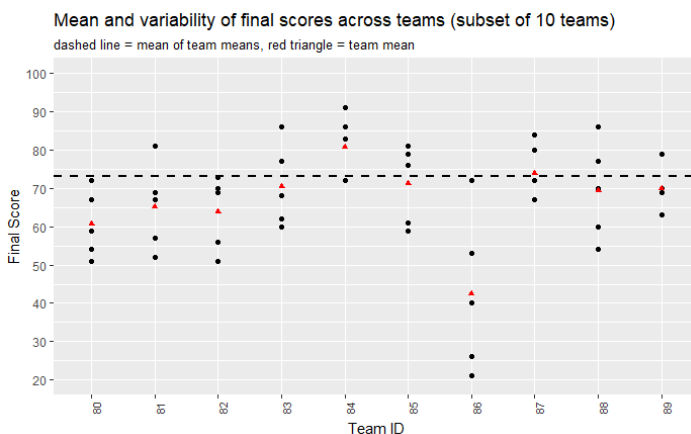
This enhanced plot shows the data with a little more detail. First, the red triangles (the team means) allow the variability of the group means across teams to become more apparent (i.e., some teams have a mean that is quite high, while others have a mean that is quite low). This plot also shows the mean of the team means (the dashed line), which gives us a benchmark for seeing how the teams means vary around the mean of means.

## A Little More Detail Within Teams

Let's look at just a subset of 10 teams (see below) to get even more details, and swap the boxplot for the individual points.

Look at subset for more detail

```
ggplot(data = subset(teams, team_id >= 80 & team_id < 90), aes(x = team_id.f, y = score)) +
  geom_point() +
  stat_summary(aes(y = score, group = team_id.f), fun.y = mean, color = "red", geom = "point", pch = 17, size = 1.5) +
  geom_hline(yintercept = team_mean$meanofmeans, linetype="dashed", color = "black", size = 1) +
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +
  labs(title = "Mean and variability of final scores across teams (subset of 10 teams)",
       subtitle = "dashed line = mean of team means, red triangle = team mean", x = "Team ID", y = "Final Score") +
  theme(axis.text.x = element_text(colour="grey20", size=8, angle=90, hjust=.5))
```



There are three key aspects of this figure:

1. The dashed line, which represents the mean of team means (73.216).
2. The variability of the red triangles (the team means) from the dashed line (the **between group variability**).
3. The variability of the team members within each team (the **within group variability**).

## Mapping these Key Aspects to an Equation—Partitioning the Variability of Scores

$y$  is the dependent variable (e.g., the final test score — the individual black dots on our plot) and the  $ij$  subscripts indicate that each  $y$  value belongs to a Level 1 unit ( $i$ ) that is nested in a Level 2 unit ( $j$ ) — e.g. a student (Level 1) on a team (Level 2).

These effects are called random effects — they carry subscripts to denote the units over which they vary. The  $u_0$  term represents the between group variability in  $y$  (notice the  $j$  subscript). The  $r$  term represents the within group variability in  $y$  (notice the  $ij$  subscript).

$$y_{ij} = \beta_0 + u_{0j} + r_{ij}$$

This is the intercept of the model — it is represented in the graph above as the black dashed line (the grand mean, that is, the mean of the team means). In the language of a multi-level model, we call this a fixed effect because it is constant for all individuals and upper level units.

## The Intraclass Correlation (ICC)

A very useful statistic in multilevel models is the Intraclass Correlation (ICC). The ICC may be interpreted in two ways:

1. The proportion of variance in a Level 1 variable that is due to between group differences.
2. The average correlation of a Level 1 variable between 2 people in the same Level 2 group.

We can calculate the ICC by estimating an unconditional (i.e., no predictors) multilevel with a random intercept, which is the equation that we studied on the previous page.

lmer is the function used to specify a multilevel model (it stands for linear mixed effects regression).

Similar to a lm, the dependent variable is listed, then a tilde. Since, this is an unconditional model, there are no predictors, but we include a 1 to denote the intercept. This is called the fixed effects part of the model and will provide us with the mean of means across the groups.

```
mod1 = lmer(score ~ 1 + (1 | team_id), REML = TRUE, data = teams)
summary(mod1)
```

After the fixed effects, we provide the random effects. Here we list the effects that we want to denote as random. In this case it is just the intercept (1), which will capture the between group variability. The bar (|) and then team\_id denotes the Level 2 grouping variable.

REML stands for Restricted Estimation Maximum Likelihood. This is one of the most common estimators for multilevel models, and for our intro, we will use this one exclusively.

## Results of the Unconditional Random Intercept Model

```
Linear mixed model fit by REML ['lmerMod']
Formula: score ~ 1 + (1 | team_id)
Data: teams
```

REML criterion at convergence: 3741.2

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.07460	-0.59606	-0.01014	0.70455	2.95746

Random effects:

Groups	Name	Variance	Std.Dev.
team_id	(Intercept)	81.58	9.032
	Residual	71.48	8.455

Number of obs: 500, groups: team\_id, 100

Fixed effects:

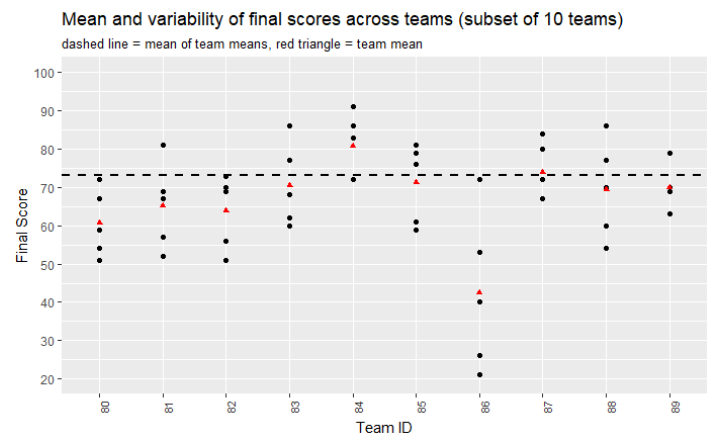
	Estimate	Std. Error	t value
(Intercept)	73.2160	0.9792	74.77

$$y_{ij} = \beta_0 + u_{0j} + r_{ij}$$

The coefficients outlined in blue represent the intercept.  $\beta_0$  is the grand mean of  $y$  (i.e., the mean of means — the dashed line),  $u_{0j}$  is the unique increment to the intercept associated with Level 2 unit  $j$  (the difference between the team mean (red triangle) and the dashed line), and  $r_{ij}$  is the unique increment to the intercept associated with case  $i$  nested in Level 2 unit  $j$  (the difference between an individual data point and the red triangle representing their respective team mean). Note that under the random effects heading in the output, the variance of  $u_{0j}$  and  $r_{ij}$  is presented.

The ICC is calculated using the formula to the right. In this model, we see that about 53% of the variance in the final test scores exists between teams, and about 47% exists within teams. In other words, about 53% of the variance in test scores may be attributed to differences between teams. This also indicates that the average correlation of the score between two people on the same team is .53.

$$ICC = \frac{\text{var}(u_{0j})}{\text{var}(u_{0j}) + \text{var}(r_{ij})} = \frac{81.58}{81.58 + 71.48} = .53$$



A function to automate ICC

```
icc.lmer <- function(modl) {
  vars <- as.data.frame(VarCorr(modl))[4]
  total <- sum(vars)
  tau00 <- vars[1,1]
  icc <- tau00/total
  return(icc)
}
icc.lmer(mod1)
```

[1] 0.53298

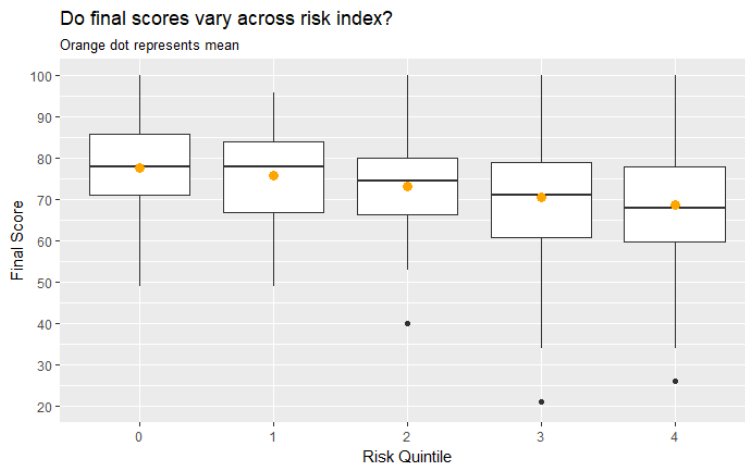


## Adding Predictors to the Multilevel Model

We can add predictors to the model in a similar fashion to traditional, single level modeling. However, we need to be cognizant of which level the predictor was measured on (e.g., Level 1 (students) or Level 2 (teams)). Let's start with a Level 1 variable — namely the risk quintile in our example study. Before adding it as a predictor to our multilevel model, let's create some descriptive plots.

Explore the risk variable

```
ggplot(data = teams, aes(x = factor(risk), y = score)) +  
  geom_boxplot() +  
  stat_summary(aes(y = score, group = factor(risk)), fun.y = mean, color = "orange", geom = "point", size = 3) +  
  guides(color = FALSE) +  
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +  
  labs(title = "Do final scores vary across risk index?", subtitle = "Orange dot represents mean",  
       x = "Risk Quintile", y = "Final Score")
```



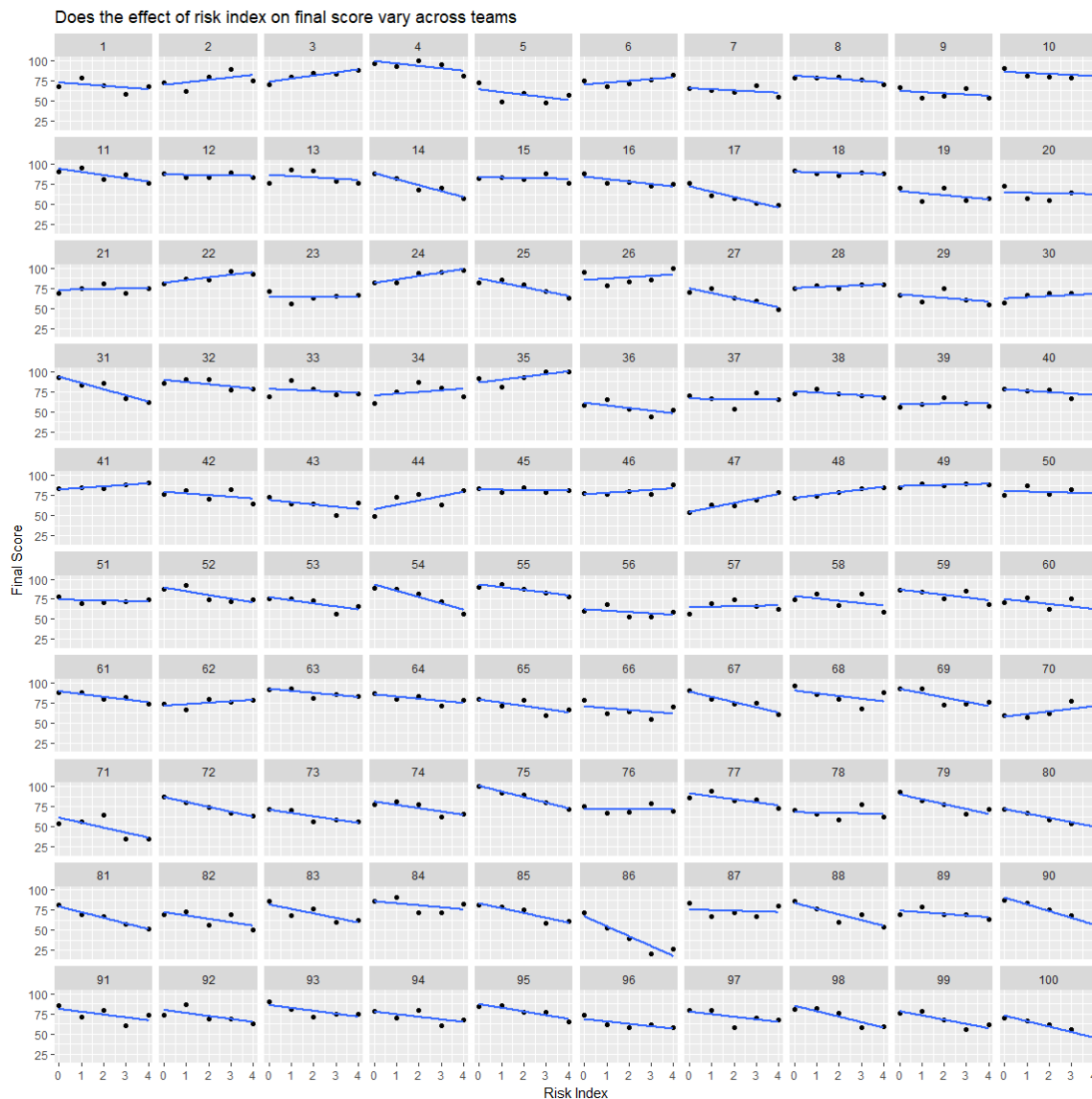
Averaging across all teams, there seems to be some evidence that students with more risk performed poorer on the final exam; however, there is clearly a lot of variability within each risk quintile.

## How Does Risk (a Level 1 Variable) Affect the Outcome?

Let's give some more thought to the nature of the risk variable. Clearly, it describes the individual. It doesn't vary at all at Level 2 (team level) because of the design of the study (i.e., all teams have exactly one person from each risk quintile). As we consider adding a Level 1 variable to the multilevel model, we need to be aware that the effect of the variable may vary across Level 2 units. Let's explore that possibility.

Plot the relationship between risk and score across teams

```
ggplot(data = teams, aes(x = risk, y = score)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE) +  
  facet_wrap(~team_id.f) +  
  labs(title = "Does the effect of risk index on final score vary across teams", x = "Risk Index", y = "Final Score")
```



## Estimate the Effect of Risk on Score Across Teams

We can use the modelr package to efficiently estimate the effect of risk on score within each team.

Group dataset by team

```
by_team <- teams %>%
  group_by(txcond.f, team_id.f)
```

Fit a SLR for each team

```
estimates <- do(by_team,
  tidy(lm(score ~ risk, data = .), conf.int = TRUE, conf.level = .95))
```

This function creates a dataset that has two rows of data for each team,. That is, one row for each of the parameter estimates of the SLR: an intercept (predicted score when risk = 0, and a slope (predicted change in score for each 1 unit increase in risk).

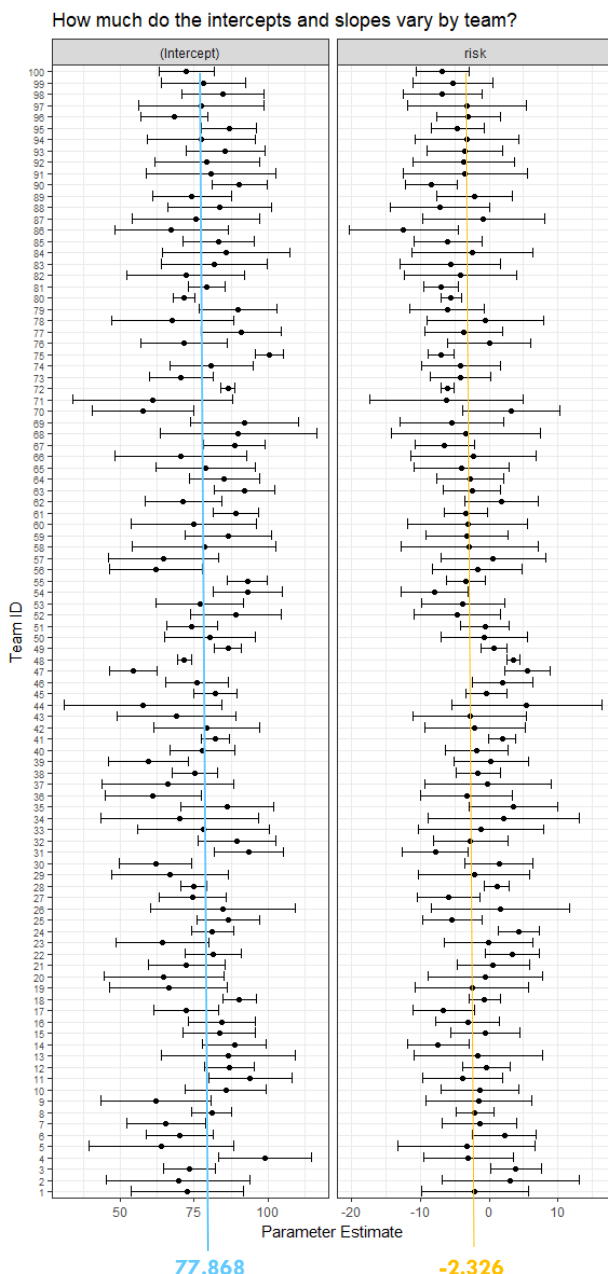
txcond.f	team_id.f	term	estimate	std.error	statistic	p.value	conf.low	conf.high
treatment	1	(Intercept)	7.260e+01	5.9514704	1.219867e+01	1.186112e-03	53.6597650	91.5402350
treatment	1	risk	-2.100e+00	2.4296776	-8.643122e-01	4.509915e-01	-9.8323186	5.6323186
treatment	2	(Intercept)	6.960e+01	7.6772391	9.065759e+00	2.835013e-03	45.1675988	94.0324012
treatment	2	risk	3.100e+00	3.1342197	9.890819e-01	3.955415e-01	-6.8744860	13.0744860
treatment	3	(Intercept)	7.340e+01	2.7820855	2.638309e+01	1.194682e-04	64.5461621	82.2538379
treatment	3	risk	3.900e+00	1.1357817	3.433759e+00	4.142418e-02	0.2854358	7.5145642
treatment	4	(Intercept)	9.920e+01	4.9558047	2.001693e+01	2.725147e-04	83.4284177	114.9715823
treatment	4	risk	-3.000e+00	2.0231988	-1.482800e+00	2.347532e-01	-9.4387215	3.4387215
treatment	5	(Intercept)	6.380e+01	7.6928538	8.293411e+00	3.672778e-03	39.3179058	88.2820942
treatment	5	risk	-3.300e+00	3.1405944	-1.050756e+00	3.705425e-01	-13.2947731	6.6947731
treatment	6	(Intercept)	7.000e+01	3.6000000	1.944444e+01	2.971422e-04	58.5431933	81.4568067
treatment	6	risk	2.200e+00	1.4696938	1.496910e+00	2.313268e-01	-2.4772217	6.8772217
treatment	7	(Intercept)	6.540e+01	4.1856899	1.562466e+01	5.697328e-04	52.0792666	78.7207334
treatment	7	risk	-1.400e+00	1.7088007	-8.192880e-01	4.726494e-01	-6.8381666	4.0381666
treatment	8	(Intercept)	8.100e+01	2.1307276	3.801518e+01	4.004227e-05	74.2190739	87.7809261
treatment	8	risk	-2.100e+00	0.8698659	-2.414165e+00	9.466265e-02	-4.8683015	0.6683015
treatment	9	(Intercept)	6.200e+01	5.8566202	1.058631e+01	1.800774e-03	43.3616207	80.6383793
treatment	9	risk	-1.500e+00	2.3909552	-6.273643e-01	5.749227e-01	-9.1090865	6.1090865
treatment	10	(Intercept)	8.580e+01	4.3335897	1.979883e+01	2.815646e-04	72.0085834	99.5914166
treatment	10	risk	-1.300e+00	1.7691806	-7.348034e-01	5.156871e-01	-6.9303223	4.3303223

## Plot the Intercept and Slopes

It can be informative to plot the intercepts and slopes across teams to see how much they vary.

Plot intercept and slope for each team

```
ggplot(data = estimates, aes(x = estimate, y = team_id.f)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  facet_wrap(~ term, scales = "free_x") +
  theme_bw() +
  theme(axis.text.y = element_text(size = 7)) +
  labs(title = "How much do the intercepts and slopes vary by team?", x = "Parameter Estimate", y = "Team ID")
```



Calculate the average intercept and slope across teams

```
avg_estimates <- estimates %>%
  group_by(term) %>%
  summarize(avg_estimate = mean(estimate))
```

avg\_estimates

term	avg_estimate
(Intercept)	77.868
risk	-2.326

From the plot, we can see quite a bit of variability in the intercept and slope across teams. We see the average intercept (i.e., the average predicted score for team members in the lowest quintile of risk (i.e., risk = 0) is 77.9 across the 100 teams. The average slope is -2.3 across the 100 teams, indicating that, on average, the final exam score is about 2.3 points lower for each 1 unit increase in risk.

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0j} + u_{1j} + r_{ij}$$

The coefficients outlined in blue represents the intercept.  $\beta_0$  is the average intercept (i.e., predicted score when risk = 0) across the Level 2 units, and  $u_{0j}$  is the unique increment to the intercept associated with the Level 2 unit  $j$ .

The coefficients outlined in orange represents the slope (i.e., the effect of risk).  $\beta_1$  is the average regression slope across the Level 2 units, and  $u_{1j}$  is the unique increment to the slope associated with the Level 2 unit  $j$ .

**Estimate a Multilevel Model, Allow Both the Intercept & Slope to Vary Across Teams**

Fit a multilevel model with a fixed and random effect for risk

```
library(lmerTest)
mod2_a = lmer(score ~ 1 + risk + (1 + risk | team_id), REML = TRUE, data = teams)
summary(mod2_a)
```

Denotes that we want to include a random effect (i.e., allow the effect to vary across teams) for both the intercept (1) and the slope (effect of risk).

lmer does not provide p-values, if you load lmerTest first (see library statement), then this add on package will interface with lmer and provide p-values.

Linear mixed model fit by REML

t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']

Formula: score ~ risk + (1 + risk | team\_id)

Data: teams

REML criterion at convergence: 3605.3

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.51237	-0.55470	0.02743	0.58197	2.14968

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
team_id	(Intercept)	80.647	8.980	
	risk	7.213	2.686	-0.22
	Residual	40.207	6.341	

Number of obs: 500, groups: team\_id, 100

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	77.8680	1.0236	99.0000	76.07	< 2e-16 ***
risk	-2.3260	0.3352	99.0000	-6.94	4.12e-10 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)
risk	-0.392

Confidence intervals via bootstrap

```
confint(mod2_a, method = "boot", nsim = 5000)
```

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0j} + u_{1j} + r_{ij}$$

Notice that we now automatically get a correlation between the random intercept and slope. The negative value indicates that, comparing across teams, having a higher intercept (a team where the student with the lowest risk quintile scores high) is associated with a steeper decline in scores as the risk quintile increases—i.e., more disparity between the highest and lowest risk team member).

Computing bootstrap confidence intervals ...

	2.5 %	97.5 %
.sig01	7.3350232	10.54225036
.sig02	-0.4591191	0.07869888
.sig03	2.0664188	3.26473830
.sigma	5.8330865	6.83337646
(Intercept)	75.8817993	79.89723809
risk	-2.9934080	-1.66017092

## A Log Likelihood Test to Determine if the Random Effect is Needed

Our graph of the intercepts and slopes across teams made it look like the slope (ie., the effect of risk) varied across teams in a substantial way. We can conduct a formal test to determine if the inclusion of the random effect significantly improves model fit.

Fit a model that excludes the random slope for risk

```
mod2_b = lmer(score ~ 1 + risk + (1 | team_id), REML = TRUE, data = teams)
summary(mod2_b)
```

Conduct the likelihood ratio test to compare the model with and without the random slope

```
anova(mod2_a, mod2_b, refit = FALSE)
```

When comparing two models the default is to refit the models using ML rather than REML. However, when comparing models that differ only in their random effects, REML is what we want, so we indicate refit = FALSE to prevent the refit.

### mod2\_a: random intercept & slope

Formula: score ~ risk + (1 + risk | team\_id)  
Data: teams

REML criterion at convergence: 3605.3

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.51237	-0.55470	0.02743	0.58197	2.14968

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
team_id	(Intercept)	80.647	8.980	
	risk	7.213	2.686	-0.22
	Residual	40.207	6.341	

Number of obs: 500, groups: team\_id, 100

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	77.8680	1.0236	99.0000	76.07	< 2e-16 ***
risk	-2.3260	0.3352	99.0000	-6.94	4.12e-10 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

(Intr)
risk -0.392

### mod2\_b: random intercept only

Formula: score ~ risk + (1 | team\_id)  
Data: teams

REML criterion at convergence: 3658.3

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.2319	-0.5752	-0.0008	0.6402	2.7829

Random effects:

Groups	Name	Variance	Std.Dev.
team_id	(Intercept)	84.26	9.179
	Residual	58.10	7.623

Number of obs: 500, groups: team\_id, 100

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	77.868	1.091	150.600	71.35	<2e-16 ***
risk	-2.326	0.241	399.000	-9.65	<2e-16 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

(Intr)
risk -0.442

These two models differ by 2 parameters — the random slope, and the correlation between the random intercept and slope. The log likelihood ratio test, which can be used on two nested models, computes the difference in the deviance ( $-2 * \log\text{Lik}$ ) between the two models. This difference (labeled Chisq) is chi-squared distributed with degrees of freedom equal to the difference in the number of parameters between the two models (2 in this case). The null hypothesis for this test is that the larger model is not better than the smaller model.

```
Data: teams
Models:
mod2_b: score ~ risk + (1 | team_id)
mod2_a: score ~ risk + (1 + risk | team_id)
      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
mod2_b  4 3666.3 3683.1 -1829.2   3658.3
mod2_a  6 3617.3 3642.6 -1802.7   3605.3 52.949    2 3.18e-12 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The chi-square (52.949) is large here and exceeds the critical value of chi-square for 2 df (5.99). You can obtain the critical value with this code: `qchisq(.95, df=2)`. Therefore the p-value is very small and we reject the null hypothesis. The random slope significantly improves the model fit.

## Plot the Results of the Two Models

Plot the fitted models

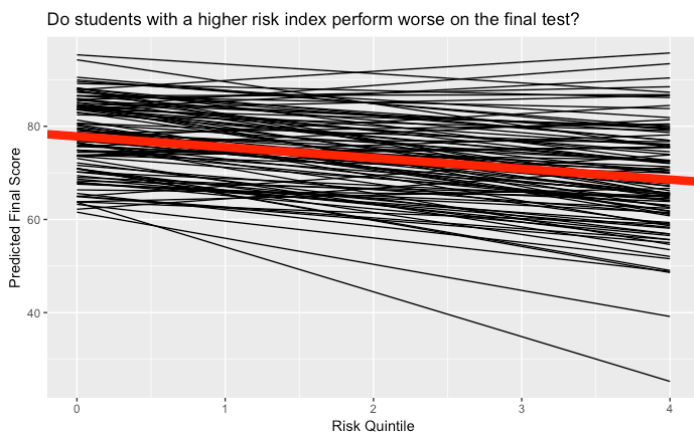
```
# Model with random slope
mod2_a.plot <- add_predictions(data = teams, model = mod2_a)

ggplot(data = mod2_a.plot, aes(x = risk, y = pred, group = team_id.f)) +
  geom_line() +
  geom_abline(intercept = 77.868, slope = -2.326, color="red", size=3) +
  labs(title = "Do students with a higher risk index perform worse on the final test?", x = "Risk Quintile", y = "Predicted Final Score")

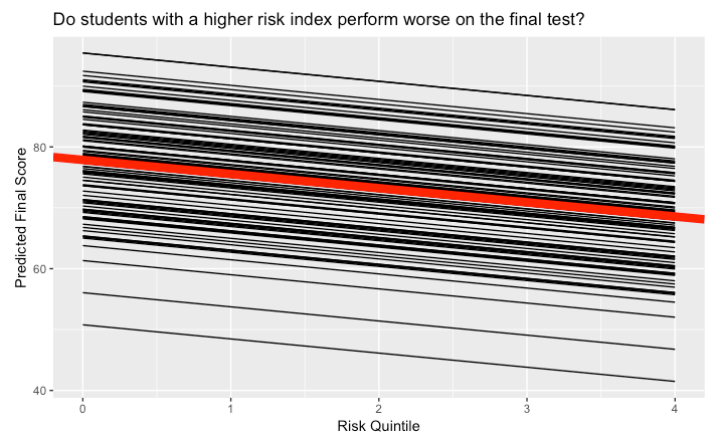
# Model without random slope
mod2_b.plot <- add_predictions(data = teams, model = mod2_b)

ggplot(data = mod2_b.plot, aes(x = risk, y = pred, group = team_id.f)) +
  geom_line() +
  geom_abline(intercept = 77.868, slope = -2.326, color="red", size=3) +
  labs(title = "Do students with a higher risk index perform worse on the final test?", x = "Risk Quintile", y = "Predicted Final Score")
```

Random intercept and slope



Random intercept only



## Add a Level 2 Covariate

The multilevel model can also include Level 2 predictors. Level 2 predictors have the capability of predicting between group variability. Our primary Level 2 predictor is treatment condition. Let's consider treatment condition as a predictor of test scores.

Plot variability in scores across teams, color boxes by condition

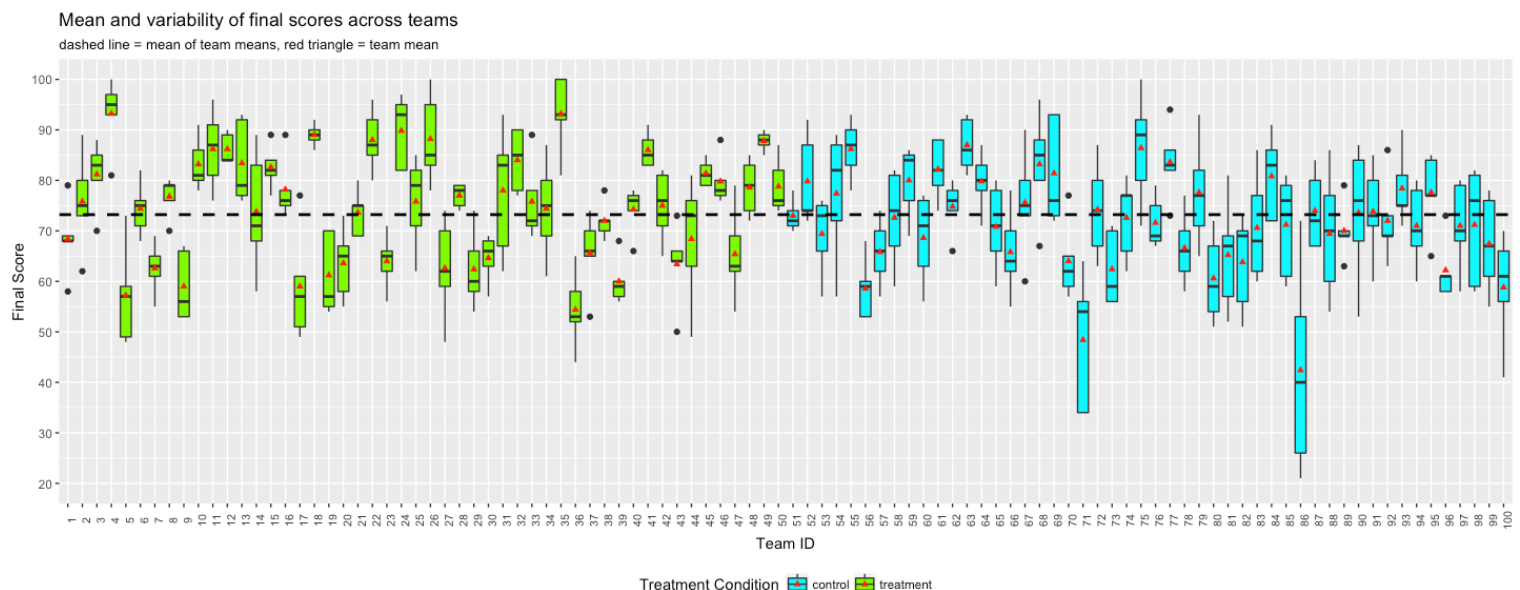
```
ggplot(data = teams, aes(x = team_id.f, y = score, fill = txcond.f)) +
  geom_boxplot() +
  stat_summary(aes(y = score, group = team_id.f), fun.y = mean, color = "red", geom = "point", pch = 17, size = 1.5) +
  geom_hline(yintercept = meanofmeans$meanofmeans, linetype="dashed", color = "black", size = 1) +
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +
  scale_fill_manual(values=c("turquoise1", "chartreuse1")) +
  labs(title = "Mean and variability of final scores across teams",
       subtitle = "dashed line = mean of team means, red triangle = team mean",
       x = "Team ID", y = "Final Score", fill = "Treatment Condition") +
  theme(axis.text.x = element_text(color="grey20", size=8, angle=90, hjust=.5), legend.position = "bottom")
```

Calculate mean of means across condition

```
team_means_bycond <- teams %>%
  group_by(team_id, txcond.f) %>%
  summarize(mean_score = mean(score)) %>%
  group_by(txcond.f) %>%
  summarize(mean_score_bycond = mean(mean_score))
```

team\_means\_bycond

txcond.f	mean_score_bycond
control	71.688
treatment	74.744



There doesn't seem to be strong evidence that treatment condition is related to average test score of the members in the group, but let's fit a multilevel model to determine if this is the case.



## Fit Multilevel Model with Treatment Condition as a Level 2 Predictor

The effect of treatment condition

```
mod3 = lmer(score ~ 1 + txcond + (1 | team_id), REML = TRUE, data = teams)
summary(mod3)
```

```
Linear mixed model fit by REML
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']
Formula: score ~ txcond + (1 | team_id)
Data: teams
```

REML criterion at convergence: 3735.6

```
Scaled residuals:
    Min       1Q   Median       3Q      Max
-3.05532 -0.60621 -0.02285  0.70949  2.97674
```

```
Random effects:
 Groups   Name      Variance Std.Dev.
team_id (Intercept) 80.18     8.954
Residual              71.48     8.455
Number of obs: 500, groups: team_id, 100
```

```
Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)   71.688      1.375 98.000  52.153  <2e-16 ***
txcond         3.056      1.944 98.000   1.572   0.119
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Correlation of Fixed Effects:
      (Intr)
txcond -0.707
```

Since txcond is a Level 2 variable, it can only have a fixed effect because it cannot vary within teams (i.e., all people on the same team received the same treatment).

The intercept of this model (71.688) is the grand mean of final score (mean of teams means) for the control teams, and the slope for txcond is the difference in this grand mean for treatment teams.

The estimate for txcond is not significant. Based on this model, treatment condition does not seem to be associated with final test scores.

txcond.f	mean_score_bycond
control	71.688
treatment	74.744

## Could the Effect of the Treatment Condition Differ by Risk Quintile?

Our last model suggests that there is no effect of the treatment on final test scores. However, this model did not take into account risk quintile. Let's imagine that the research team believed that the team-focused intervention would be more important for students who were less likely to succeed. Let's look at a few plots to get a sense of this possibility.

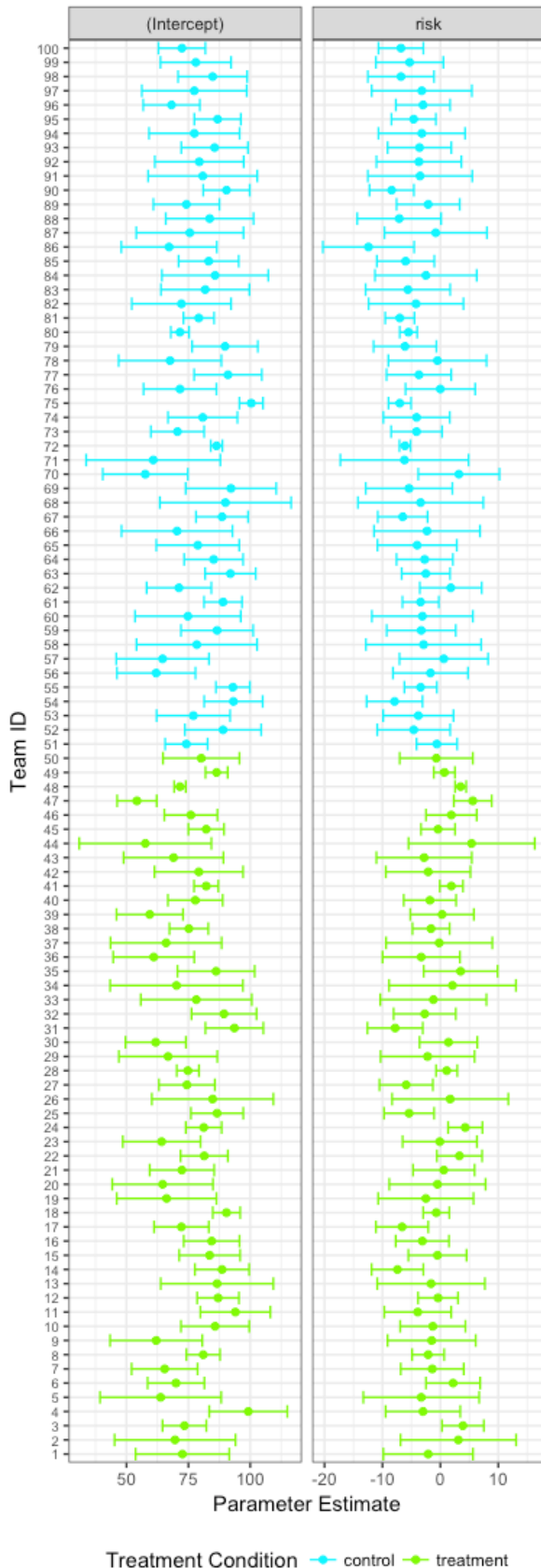
Recreate the random intercept and slope plot, but now color effects by treatment condition

```
ggplot(data = estimates, aes(x = estimate, y = team_id.f, group = txcond.f, color = txcond.f)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  facet_wrap(~ term, scales = "free_x") +
  scale_color_manual(values=c("turquoise1", "chartreuse1")) +
  theme_bw() +
  theme(axis.text.y = element_text(size = 7)) +
  labs(title = "Are there condition differences in the intercepts and slopes across teams?",
       x = "Parameter Estimate", y = "Team ID", color = "Treatment Condition")
```

Boxplot of score by risk quintile across conditions

```
ggplot(data = teams, aes(x = factor(risk), y = score)) +
  geom_boxplot() +
  stat_summary(aes(y = score, group = factor(risk)), fun.y = mean, color = "orange", geom = "point", size = 3) +
  facet_wrap(~ txcond.f) +
  guides(color = FALSE) +
  scale_y_continuous(limits = c(20,100), breaks = seq(20, 100, 10)) +
  labs(title = "Do final scores vary across risk index by treatment condition?",
       subtitle = "Orange dot represents mean",
       x = "Risk Quintile", y = "Final Score")
```

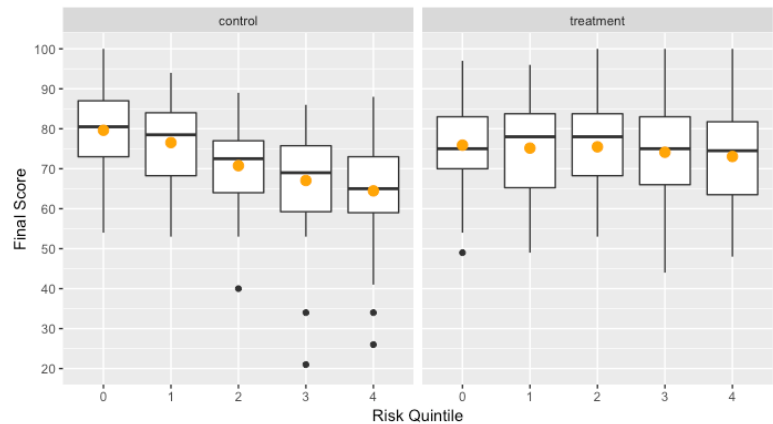
Are there condition differences in the intercepts and slopes across teams?



It appears that there is an interaction between risk quintile and treatment condition, where the effect of risk quintile is different across treatment conditions. First, you can see clearly from the boxplot that in the control condition there is a large, negative effect of risk quintile on final score — more risk equals poorer performance on the test. On the other hand, in the treatment condition, risk quintile doesn't seem to be related to final score. The team-focused intervention seems to have mitigated the expected negative effect of risk quintile on test score. Also notice that when you compare people in the lowest quintile for risk across conditions there isn't much of a difference in test score — in fact, the students in the control condition might have done a little better. However, among students in the highest risk quintile, those in the treatment condition performed substantially better on the final test than those in the control condition.

Do final scores vary across risk index by treatment condition?

Orange dot represents mean

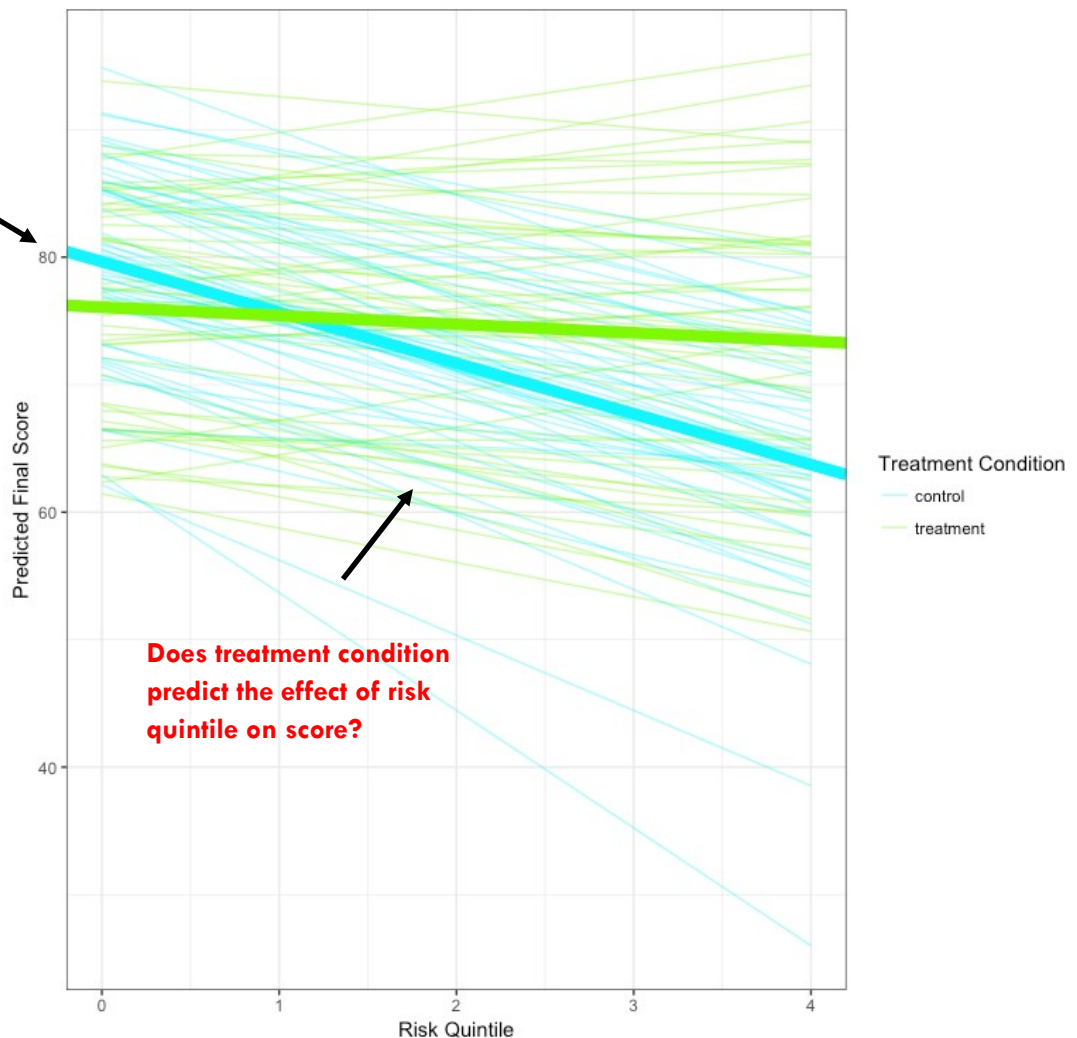


### What Variability Can a Level 2 Predictor Predict?

In thinking about the ways in which treatment condition can influence final score in the context of risk, we can imagine that it could have an effect on both the intercept (i.e., the predicted final score when risk quintile = 0) and the slope (the effect of risk quintile on final test score). The former tells us about whether or not the scores of students in the lowest risk quintile differ as a function of treatment condition, and the latter tells us about whether or not the treatment condition changed the effect of risk quintile on final test score. That is, whether the treatment was effective in attenuating the expected negative relationship between risk quintile and final test score.

Do students with a higher risk index perform worse on the final test?

Does treatment condition predict final score when risk quintile = 0?



## Map Estimates Onto a Multilevel Model

Let's compute the average intercept and slope (for the effect of risk) by treatment condition, and then use these estimates to map onto the equation for a multilevel model with a Level 1 predictor, a random intercept and slope, and a Level 2 predictor of the intercept and slope.

Compute average of intercepts and slopes by condition

```
txcond_avg_estimates <- estimates %>%
  group_by(term, txcond.f) %>%
  summarize(avg_estimate = mean(estimate))

txcond_avg_estimates
```

term	txcond.f	avg_estimate	
(Intercept)	control	79.648	$79.648 - 76.088 = -3.56$
(Intercept)	treatment	76.088	
risk	control	-3.980	$-3.980 + -.672 = 3.31$
risk	treatment	-0.672	

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 Z_j + \beta_3 x_{ij} Z_j + u_{0j} + u_{1j} + r_{ij}$$

Notice that using a Level 2 variable (txcond—represented as Z in the equation above) to predict a random slope means specifying an interaction term between the random slope (i.e., the effect of risk within each of the teams) and the Level 2 predictor (txcond). This is called a **cross-level interaction** because one variable is measured at Level 2 and one variable is measured at Level 1.

## Estimate Multilevel Model with Cross Level Interaction

Let's specify our multilevel model with a cross level interaction using lmer, and then map these estimates onto the values that we obtained in the prior step.

Estimate the model with a cross level interaction for risk and condition

```
mod4 = lmer(score ~ 1 + risk + txcond + risk*txcond + (1 + risk | team_id), REML = TRUE, data = teams)
summary(mod4)
```

```
Linear mixed model fit by REML
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']
Formula: score ~ risk + txcond + risk * txcond + (1 + risk | team_id)
Data: teams
```

REML criterion at convergence: 3573.5

```
Scaled residuals:
    Min       1Q   Median       3Q      Max
-2.53021 -0.55905  0.02051  0.62477  2.15423
```

```
Random effects:
 Groups   Name      Variance Std.Dev. Corr
team_id  (Intercept) 78.483    8.859
          risk       4.536    2.130   -0.14
Residual          40.207    6.341
Number of obs: 500, groups: team_id, 100
```

```
Fixed effects:
              Estimate Std. Error    df t value Pr(>|t|)
(Intercept)  79.6480    1.4325 98.0000  55.600 < 2e-16 ***
risk         -3.9800    0.4137 98.0000 -9.621 8.88e-16 ***
txcond       -3.5600    2.0259 98.0000 -1.757  0.082 .
risk:txcond   3.3080    0.5850 98.0000  5.654 1.54e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Correlation of Fixed Effects:
      (Intr) risk  txcond
risk   -0.357
txcond -0.707  0.253
risk:txcond 0.253 -0.707 -0.357
```

The intercept is the grand mean of y for an individual with a risk quintile score of 0 among teams in the control condition (txcond = 0).

The slope for risk is the average change in y for a 1-unit increase in risk among teams in the control condition (txcond = 0).

The slope for txcond is the difference in the grand mean of y for an individual with a risk quintile score of 0 for treatment condition teams as compared to control condition teams.

The slope for the interaction is the difference in the average change in y for a 1-unit increase in risk for treatment condition teams as compared to control condition teams.

term	txcond.f	avg_estimate
(Intercept)	control	79.648
(Intercept)	treatment	76.088
risk	control	-3.980
risk	treatment	-0.672

$$79.648 - 76.088 = -3.56$$

$$-3.980 + -0.672 = -3.31$$

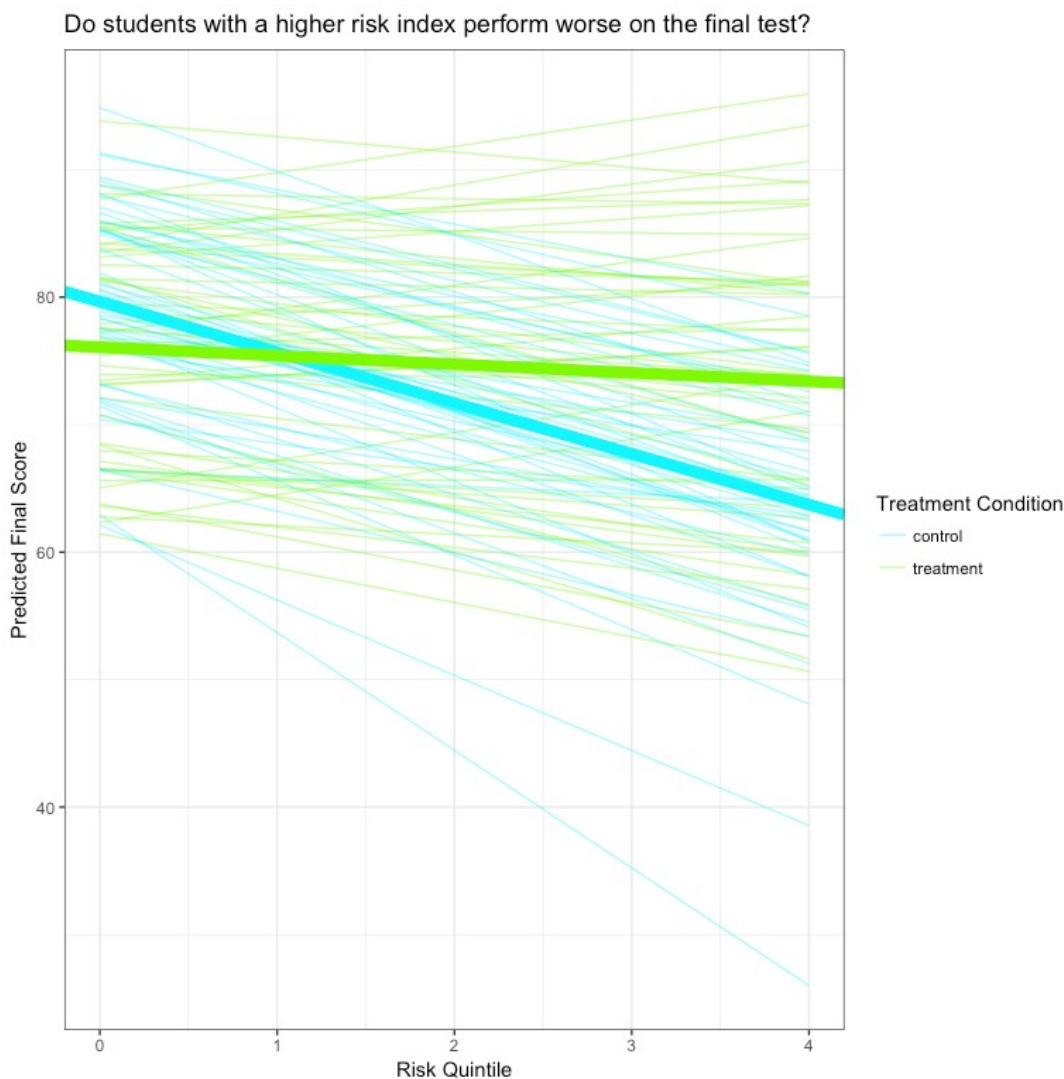
$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 Z_j + \beta_3 x_{ij} Z_j + u_{0j} + u_{1j} + r_{ij}$$

## Plot the Fitted Model with a Cross Level Interaction

Plot fitted model

```
# Get predicted values
mod4.plot <- add_predictions(data = teams, model = mod4)

ggplot(data = mod4.plot, aes(x = risk, y = pred, group = team_id.f, color = txcond.f)) +
  geom_line(alpha = .4) +
  geom_abline(intercept = 79.648, slope = -3.98, color="turquoise1", size=3) +
  geom_abline(intercept = (79.648 + (-3.56)), slope = (-3.98 + 3.308), color="chartreuse1", size=3) +
  theme_bw() +
  scale_color_manual(values=c("turquoise1", "chartreuse1")) +
  labs(title = "Do students with a higher risk index perform worse on the final test?",
       x = "Risk Quintile", y = "Predicted Final Score", color = "Treatment Condition")
```





## Centering of Level 1 Predictors

Similar to the traditional, single-level regression models, choice of centering for predictor variables has important implications for the definition of the intercept and for interpretation of slopes of variables involved in an interaction with another variable in a multilevel model. But, with multilevel models it's a little more complex for additional reasons.

For Level 1 variables that vary at both Level 1 and Level 2, an  $x_{ij}$  can be centered in two ways:

1. Grand mean centering – from each  $x_{ij}$  we subtract the grand mean of  $x$  (or some other constant).
2. Group mean centering – from each  $x_{ij}$  we subtract the respective group mean of  $x$  (i.e., the team mean).

As you will soon see, depending on which method we choose, the regression coefficients for the variable of interest will capture different types of effects.

Let's consider a new Level 1 predictor that varies at both Level 1 and Level 2: belief in communal affordances of STEM (comafrd). **For simplicity, we will restrict our analysis to only teams in the treatment condition.**

Create group and grand mean centered versions of comafrd

```

t <- t %>%
  mutate(grndmc_comafrd = comafrd - mean(comafrd)) %>%
  group_by(team_id.f) %>%
  mutate(team_comafrd = mean(grndmc_comafrd),
         grpmc_comafrd = grndmc_comafrd - team_comafrd) %>%
  ungroup()

```

	kid_id	team_id	txcond	risk	score	comafrd	txcond.f	team_id.f	grndmc_comafrd	team_comafrd	grpmc_comafrd
1	1	1	1	0	68	2.217525	treatment	1	-2.815710387	-0.71984057	-2.0958698183
2	2	1	1	1	79	5.366212	treatment	1	0.332975917	-0.71984057	1.0528164860
3	3	1	1	2	69	6.354152	treatment	1	1.320916081	-0.71984057	2.0407566501
4	4	1	1	3	58	3.143147	treatment	1	-1.890088579	-0.71984057	-1.1702480104
5	5	1	1	4	68	4.485940	treatment	1	-0.547295876	-0.71984057	0.1725446926
6	6	2	1	0	73	3.817667	treatment	2	-1.215568594	-1.15658199	-0.0589866046
7	7	2	1	1	62	3.451067	treatment	2	-1.582168583	-1.15658199	-0.4255865930
8	8	2	1	2	80	4.066355	treatment	2	-0.966880947	-1.15658199	0.1897010432
9	9	2	1	3	89	3.885438	treatment	2	-1.147798040	-1.15658199	0.0087839494
10	10	2	1	4	75	4.162742	treatment	2	-0.870493785	-1.15658199	0.2860882050
11	11	3	1	0	70	4.184316	treatment	3	-0.848920124	0.26764877	-1.1165688971
12	12	3	1	1	80	5.130441	treatment	3	0.097204786	0.26764877	-0.1704439879
13	13	3	1	2	85	5.798342	treatment	3	0.765106290	0.26764877	0.4974575165
14	14	3	1	3	83	5.776268	treatment	3	0.743032656	0.26764877	0.4753838827
15	15	3	1	4	88	5.615056	treatment	3	0.581820259	0.26764877	0.3141714858

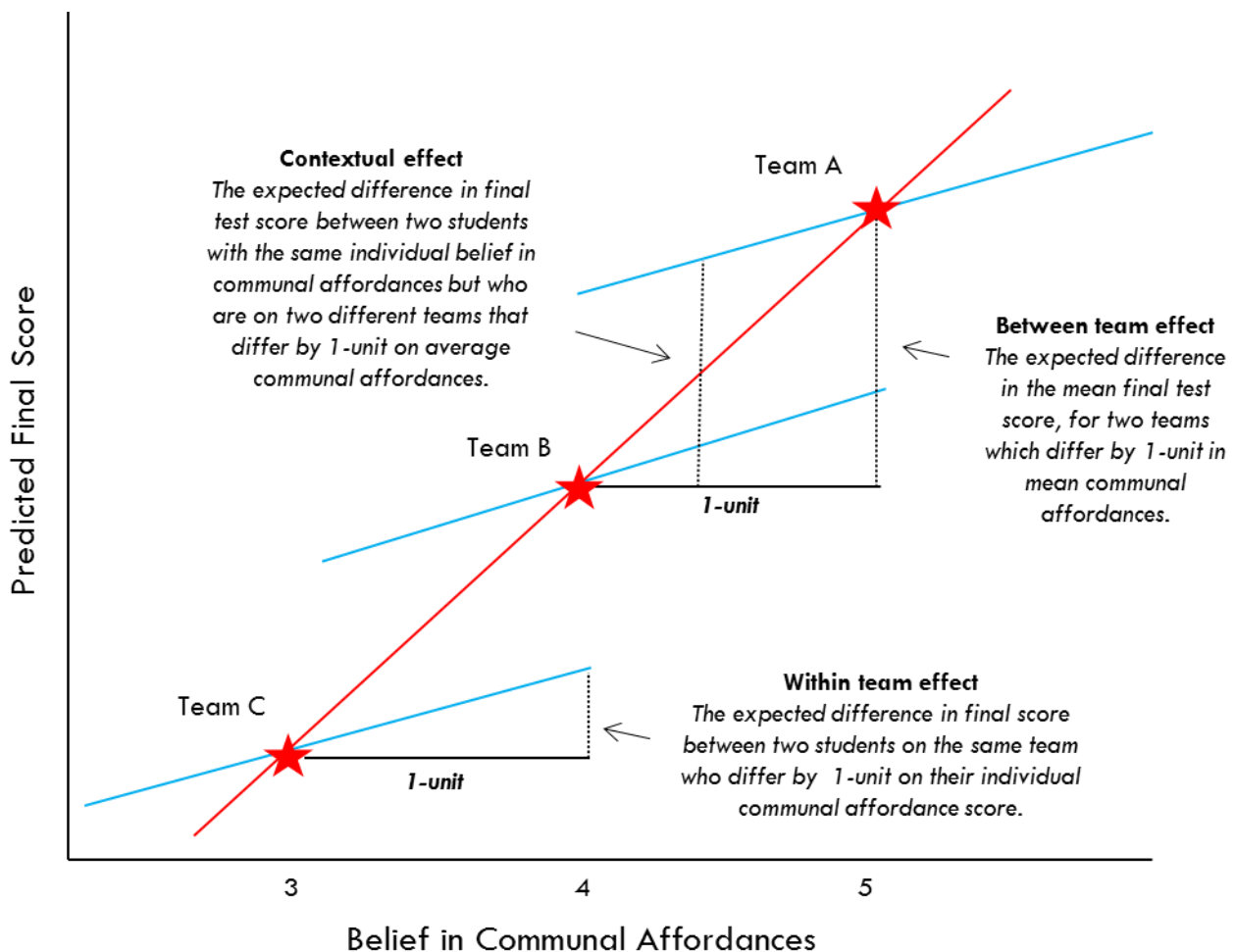


## Separation of Within Group & Between Group Effects for a Level 1 Predictor

In addition to this complexity, for a Level 1 variable that varies at both levels, we can separate the effect of the variable into two components:

1. A **within group effect**: Within a team, what is the expected difference in final test score for a 1-unit increase in the student's communal affordances score?
2. A **between group effect**: What is the expected difference in the average final test score for two teams who differ by one unit on the average communal affordances score?

The difference between the within group effect and the between group effect is called the **contextual effect**: What is the expected difference in final test score for two students who have the same personal score for communal affordances but are on two different teams that differ by 1-unit on the average communal affordances score?



★ = average communal affordance for the team

## A Naïve Model

Let's begin with a naïve model that specifies a single effect of communal affordances. In this model, the within and between team effect of communal affordances are conflated, and we make the assumption that the within and between group effects are equivalent. If in the population this assumption is not true, then with this model we would commit an ecological fallacy, that is, we would draw conclusions about individuals nested within groups based on (at least in part) between group differences.

Initial assessment of the effect of communal affordances on score

```
mod5_a = lmer(score ~ 1 + comafrd + (1 | team_id), REML = TRUE, data = t)
summary(mod5_a)
```

```
Linear mixed model fit by REML
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']
Formula: score ~ comafrd + (1 | team_id)
Data: t
```

REML criterion at convergence: 1800

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.78536	-0.52864	-0.02209	0.61550	2.29057

Random effects:

Groups	Name	Variance	Std.Dev.
team_id	(Intercept)	84.51	9.193
	Residual	51.23	7.158

Number of obs: 250, groups: team\_id, 50

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	64.5306	2.7377	226.3400	23.571	< 2e-16 ***
comafrd	2.0292	0.4701	230.5800	4.316	2.36e-05 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)
comafrd	-0.864

## How Much does Communal Affordances Vary Within & Between Teams?

Let's dig into the assumption of equal within and between group effects. We'll start by exploring the mean and variability of communal affordances within and between teams.

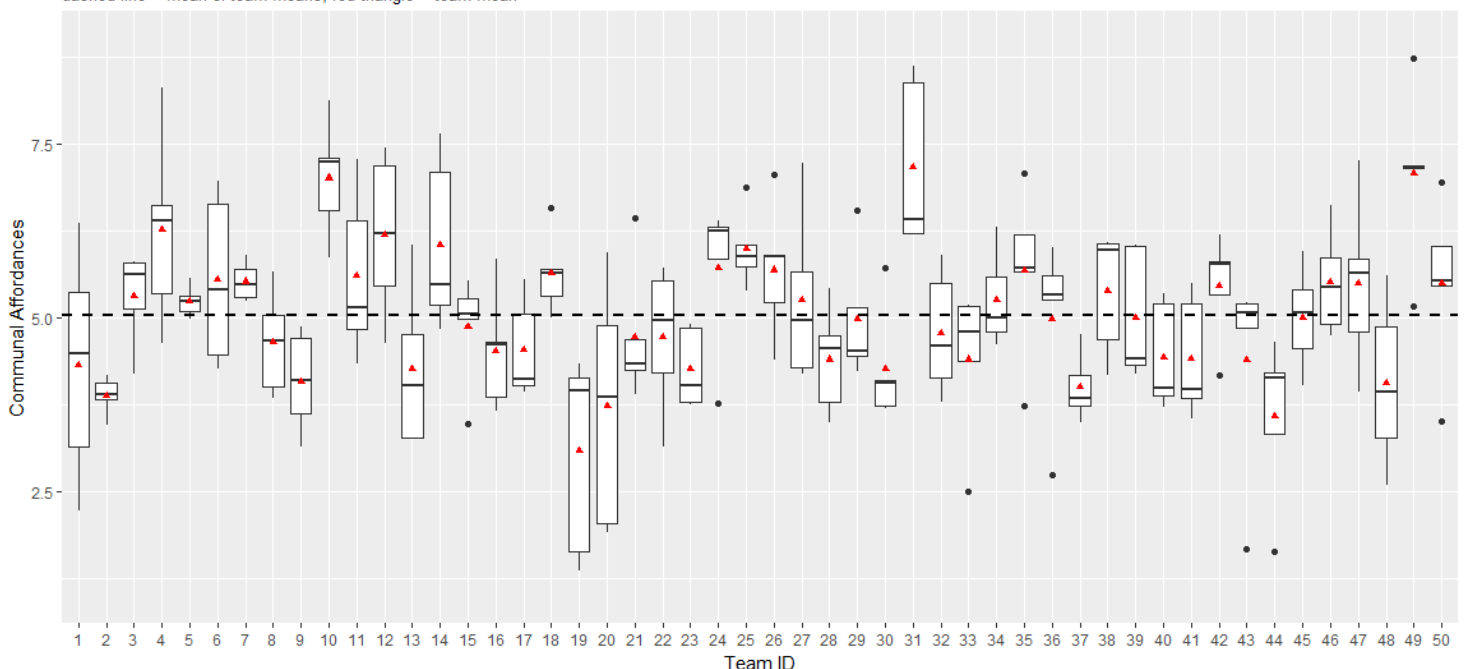
Take a look at the mean and variability of communal affordances within and between teams

```
# Create team means
tx_mean <- t %>%
  group_by(team_id.f) %>%
  mutate(mean_comafrd = mean(comafrd)) %>%
  ungroup () %>%
  summarize(meanofmeans = mean(mean_comafrd))

ggplot(data = t, aes(x = team_id.f, y = comafrd)) +
  geom_boxplot() +
  stat_summary(aes(y = comafrd, group = team_id.f), fun.y = mean, color = "red", geom = "point", pch = 17, size = 1.5) +
  scale_y_continuous(limits = c(1,9)) +
  geom_hline(yintercept = tx_mean$meanofmeans, linetype="dashed", color = "black", size = 1) +
  labs(title = "Mean and variability of belief in communal affordances of STEM careers across TREATMENT teams",
       subtitle = "dashed line = mean of team means, red triangle = team mean",
       x = "Team ID", y = "Communal Affordances")
```

Mean and variability of belief in communal affordances of STEM careers across TREATMENT teams

dashed line = mean of team means, red triangle = team mean

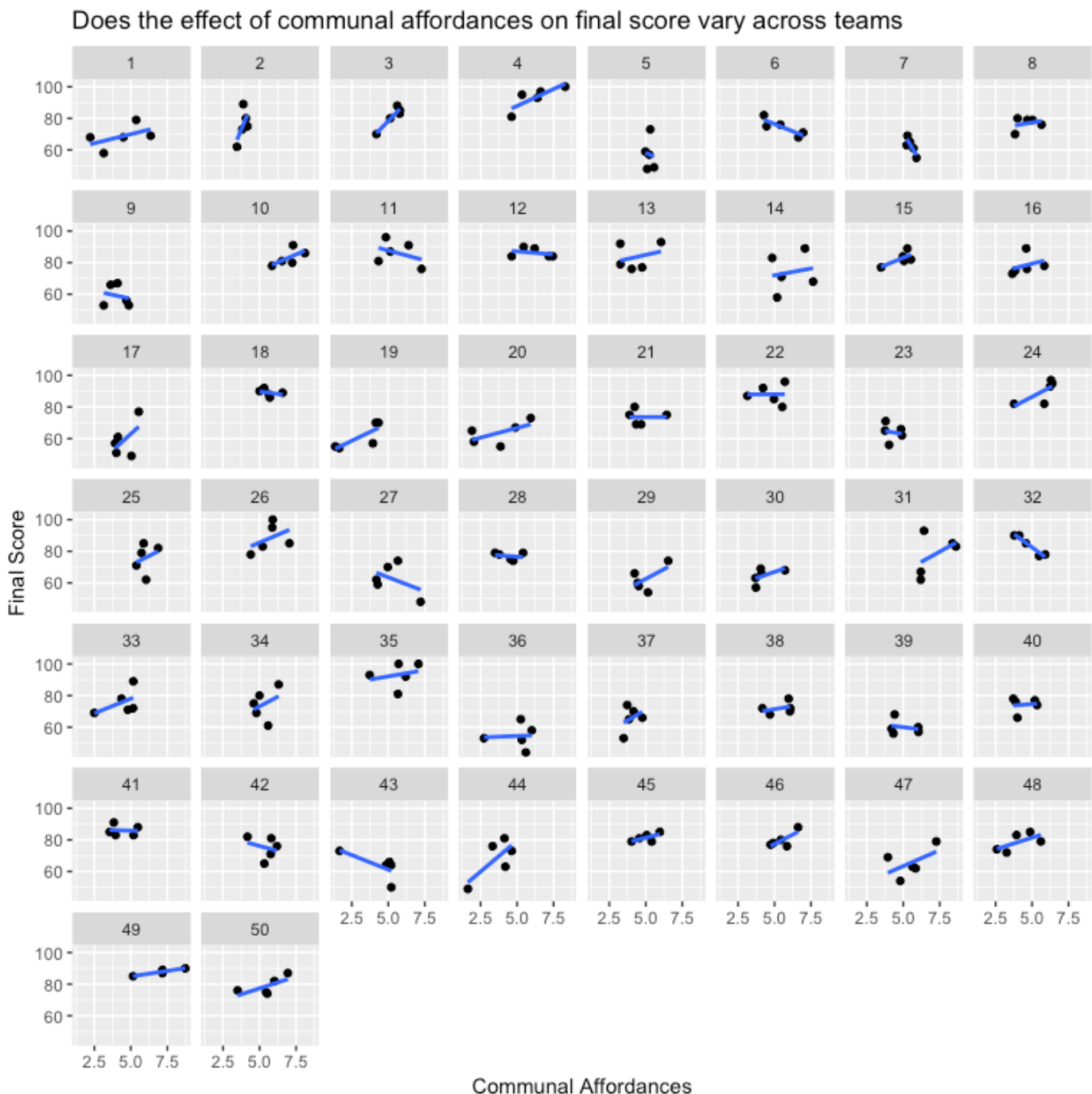


## Is Belief in Communal Affordances Associated with Score Within Teams?

Now, let's take a look at how communal affordances is related to final test score within each of the teams.

Within team effect of comafrd

```
ggplot(data = t, aes(x = comafrd, y = score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~team_id.f) +
  labs(title = "Does the effect of communal affordances on final score vary across teams", x = "Communal Affordances", y = "Final Score")
```



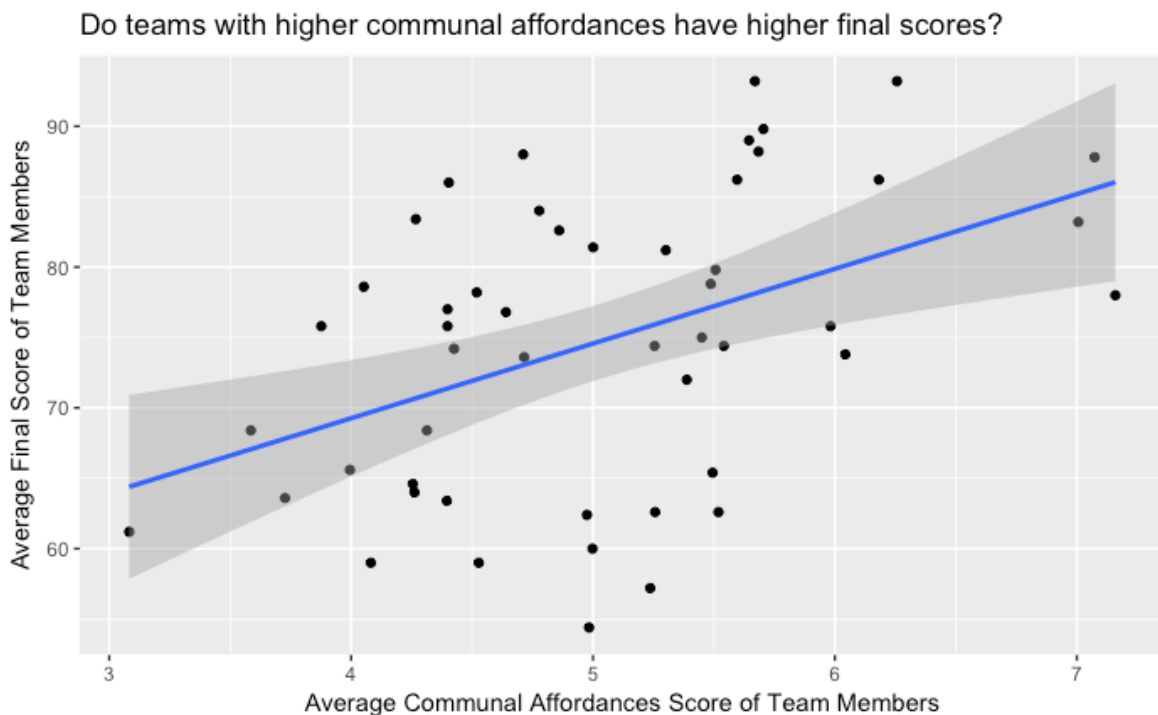
## Is Mean Communal Affordances Associated with Mean Test Score Between Teams?

By aggregating both communal affordances and final test score to the mean for each team, we can take a look at how average communal affordances is related to average final test score across teams.

Scatterplot of team means for comafrd and score

```
# create an aggregated dataset - one row per team
team_avg <- t %>%
  group_by(team_id.f) %>%
  summarize(mean_score = mean(score), mean_comafrd = mean(comafrd)) %>%
  ungroup()

ggplot(data = team_avg, aes(x = mean_comafrd, y = mean_score)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Do teams with higher communal affordances have higher final scores?",
       x = "Average Communal Affordances Score of Team Members", y = "Average Final Score of Team Members")
```



## Group Mean Center Communal Affordances & Fit Multilevel Model

Now, we will fit a new multilevel model that explicitly separates the within and between group effect of communal affordances. To accomplish this task, we will use the group mean (i.e., team mean) centered version of `comafrd` at Level 1. When included in the model, this group mean centered version of `comafrd` (a Level 1 predictor) will ONLY be able to account for within team effects. To estimate the between team effect, we will add the team means of `comafrd` to the model as a Level 2 predictor.

Group mean center `comafrd` at Level 1

```
mod5_b = lmer(score ~ 1 + grpmc_comafrd + team_comafrd + (1 | team_id), REML = TRUE, data = t)
summary(mod5_b)
```

```
Linear mixed model fit by REML
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']
Formula: score ~ grpmc_comafrd + team_comafrd + (1 | team_id)
Data: t
```

REML criterion at convergence: 1792.3

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.72630	-0.54074	-0.04404	0.65195	2.14489

Random effects:

Groups	Name	Variance	Std.Dev.
team_id	(Intercept)	77.81	8.821
Residual		51.15	7.152

Number of obs: 250, groups: team\_id, 50

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	74.7440	1.3270	48.0000	56.327	< 2e-16 ***
grpmc_comafrd	1.7098	0.4921	199.0000	3.474	0.000628 ***
team_comafrd	5.3060	1.5205	48.0000	3.490	0.001047 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)	grpmc_
grpmc_cmfrd	0.000	
team_cmfrd	0.000	0.000

In this model, the slope associated with `grpmc_comafrd` (a Level 1 variable) is the within team effect. The value of 1.7098, which is statistically significant, indicates that, for two students on the same team who differ by 1-unit on `comafrd`, we expect the student with the higher `comafrd` score to have a final test score that is 1.7 points higher.

The slope associated with `team_comafrd` (a Level 2 variable) is the between team effect. The value of 5.3060, which is statistically significant, indicates that we expect two teams who have an average `comafrd` score that differs by 1 unit, to have a average for the final test score that differs by 5.3 points (i.e., the team with higher average `comafrd` will have a higher average test score).

By taking the difference of the between team and within team effect ( $5.3060 - 1.7098 = 3.60$ ), we arrive at the contextual effect. This value indicates the expected difference in final test score for two students who have the same score on `comafrd`, but are on two different teams that differ by 1 unit on their average final test score (i.e., the student on the team with the higher average `comafrd` score will have a higher average test score).

## Grand Mean Center Communal Affordances & Fit Multilevel Model

In the previous group mean centered model, we were able to obtain the contextual effect by subtracting the within team effect from the between team effect; however, we didn't obtain a significance test for this value. We can re-specify the multilevel model to obtain this significance test by swapping the group mean centered version of `comafrd` with a grand mean centered version of `comafrd`. We created this version (`grndmc_comafrd`) in our previous step, rather than subtracting the team mean from each student's score, we subtracted the grand mean.

Grand mean center `comafrd` at Level 1

```
mod5_c = lmer(score ~ 1 + grndmc_comafrd + team_comafrd + (1 | team_id), REML = TRUE, data = t)
summary(mod5_c)
```

Linear mixed model fit by REML

t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']

Formula: `score ~ grndmc_comafrd + team_comafrd + (1 | team_id)`

Data: `t`

REML criterion at convergence: 1792.3

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.72630	-0.54074	-0.04404	0.65195	2.14489

Random effects:

Groups	Name	Variance	Std.Dev.
team_id	(Intercept)	77.81	8.821
	Residual	51.15	7.152

Number of obs: 250, groups: team\_id, 50

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	74.7440	1.3270	48.0000	56.327	< 2e-16 ***
grndmc_comafrd	1.7098	0.4921	199.0000	3.474	0.000628 ***
team_comafrd	3.5962	1.5982	58.4300	2.250	0.028217 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr) grndm_
grndmc_cmfr	0.000
team_comfrd	0.000 -0.308

Notice that the Level 1 effect (labeled `grndmc_comafrd`) is exactly the same as in the previous model. By adjusting for the team mean, we force the Level 1 grand mean centered version of `comafrd` to pick up only within team effects. Also note that the Level 2 effect (labeled `team_comafrd`) has changed, and in fact is now equal to the contextual effect that we computed by hand from the previous model. This slope for `team_comafrd` is the contextual effect, and we now have a significance test for this effect.

With the Level 1 version of the variable centered at the grand mean, the between group effect is obtained by ADDING the within group effect and the contextual effect ( $1.7098 + 3.5962 = 5.31$ ).

## Growth Modeling—A ML Model of Measurement Occasions Nested in Person

We can use the same techniques that we learned thus far to analyze longitudinal data. In this type of model measurement occasions (Level 1) are nested in person (Level 2). We need at least 3 measurement occasions to specify a growth model. In this second section of our Multilevel Modeling Unit, we will consider an extension of the Data Science Intervention.

## Example Dataset Description — Promoting Data Science Careers, Longitudinal Data

The research team was keenly interested in the skill development of the students least likely to do well. They desired to determine if those at highest risk of low performance (highest risk quintile) showed more skill improvement over the course of the summer program if they were on one of the team-focused teams (as opposed to an individual-focused team). During each week of the program, all participants completed a coding challenge. Each challenge had a set of coding skills that had to be employed to solve the challenge, but each challenge focused on solving some substantive problem (e.g., mapping social networks of users on an online forum, developing an algorithm to recommend new music based on a user's Spotify history, etc.). Each student's performance on the challenge was graded by the research team using a valid and reliable rubric able to detect growth in skills over time. The primary hypothesis was that students in the team-focused condition would acquire more from the program than students in the individual-focused condition.

As a secondary aim, the research team sought to determine if a student's interest in the substantive topic of the coding challenge would be predictive of their performance. Therefore, prior to the start of the intervention, each student rated their level of interest in the substantive topic of each week's challenge. It was hypothesized that having more interest in the substantive topic of the coding challenge would be associated with better performance.

The dataset called `mlm_grow.csv` **represents the students in the highest risk quintile (so only one student per team)**. The following variables are included:

**kid\_id**: The student's ID number.

**team\_id**: The team ID of the student.

**txcond**: The treatment condition, 0 = individual-focused condition, 1 = team-focused condition.

**perf\_0 - perf\_6**: Performance on the coding challenge each week of the program, starting at baseline (before the program began (`perf_0`) and ending at week 6 (`perf_6`). The score ranges from 0 to 10.

**inter\_0 - inter\_6**: Level of interest in the coding challenge, measured prior to the start of the program and rated on a scale ranging from -4 (absolutely no interest in the topic) to +4 (maximum interest in the topic).

## Summary of Data Structure

### Level 2: Students

kid\_id 5

...

kid\_id 500

Variables = txcond

week0 ... week6

week0 ... week6

Variables = perform, interest

### Level 1: Measurement occasions



## Set Up a New Notebook

Please set up a new notebook called: MultilevelModels\_GrowthNotebook in your MyClassActivities Folder, and add the following code chunks.

Load libraries

```
library(tidyverse)
library(broom)
library(modelr)
library(lme4)
library(lmerTest)
```

Import data

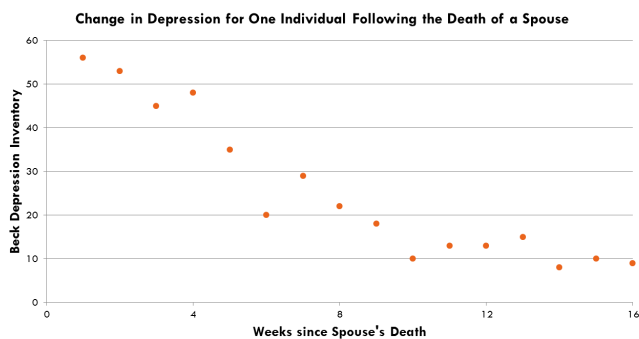
```
wide <- read_csv("mlm_grow.csv")
```

## Aims of Growth Modeling

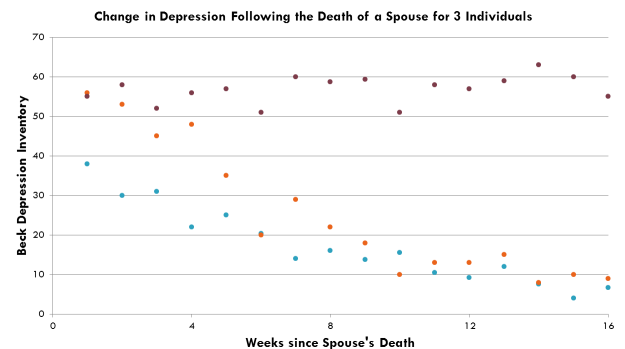
In a series of papers in the 1970's and 1980's, John Nesselroade & Paul Baltes proposed five primary objectives of longitudinal research.

1. Identification of intra-individual change
2. Identification of inter-individual differences in intra-individual change
3. Analysis of causes (determinants) of intra-individual change
4. Analysis of causes (determinants) of inter-individual differences in intra-individual change
5. Analysis of interrelationships in intra-individual change

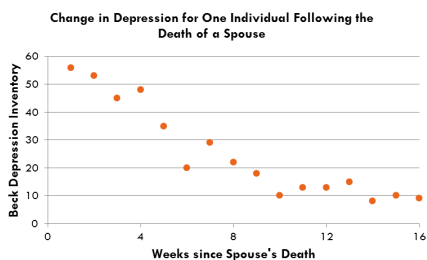
### 1. How does an individual change over time?



### 2. How do individuals differ in the way that they change?

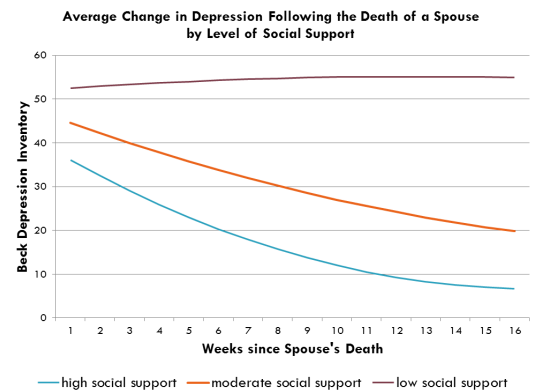


### 3. What changes WITHIN PERSON to cause the intraindividual change?

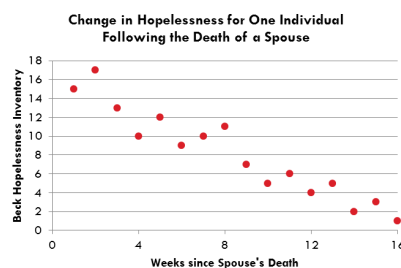
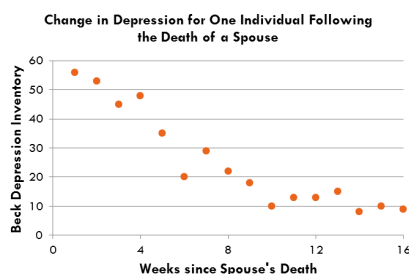


- Movement through the stages of grief.
- Return to normal daily functions.
- Meet a new partner

### 4. Why do some people change differently than others?



### 5. Does change in one process correspond with changes in another process?



## Flipping the Data from Wide to Long

In our longitudinal dataset, there is one row of data per person. The longitudinal measures of performance and interest are each in a separate column. In order to analyze these data as a multilevel model, we must transpose (flip) the data from a wide format to a long format. In the long format, we will have multiple rows of data for each student — specifically, one row per measurement occasion (7 in our example).

We explored flipping data from wide to long, and long to wide at the beginning of PSY 652 using the spread and gather function in tidy. Let's revisit the code to accomplish this task.

Transpose data from wide format to long format

```
long_perf <- wide %>%
  select(kid_id, txcond, perf_0:perf_6) %>%
  gather(key = measure, value = perform, perf_0:perf_6) %>%
  separate(measure, c("temp", "week")) %>%
  select(-temp) %>%
  mutate(week = as.numeric(week))

long_inter <- wide %>%
  select(kid_id, inter_0:inter_6) %>%
  gather(key = measure, value = interest, inter_0:inter_6) %>%
  separate(measure, c("temp", "week")) %>%
  select(-temp) %>%
  mutate(week = as.numeric(week))

grow <- full_join(long_perf, long_inter, by = c("kid_id", "week"))
```

Format data and create subsets by condition

```
grow <- grow %>%
  mutate(txcond.f = factor(txcond, levels = c(0,1), labels = c("control", "treatment")),
         kid_id.f = factor(kid_id))

c <- filter(grow, txcond == 0)
t <- filter(grow, txcond == 1)
```

# RESEARCH METHODS IN PSYCHOLOGY I & II

## Wide and Long Data

kid_id	team_id	txcond	perf_0	perf_1	perf_2	perf_3	perf_4	perf_5	perf_6	inter_0	inter_1	inter_2	inter_3	inter_4	inter_5	inter_6
5	1	1	6.332146	6.507988	6.888969	5.563810	6.152750	5.984029	4.932417	0.36266696	1.41140464	1.367544315	-0.407176147	0.76254173	-0.651200823	-1.47365578
10	2	1	4.531640	5.602623	6.851863	6.076729	5.586186	6.255608	8.042116	-1.20166580	-0.14872048	1.797062429	0.104808718	-0.80283797	0.230374964	0.69797519
15	3	1	4.902634	6.072946	4.830770	6.623872	7.695781	8.171289	9.632302	-1.29520173	-1.05468022	-1.941607490	-1.269560593	-0.88743881	-0.291915763	1.41009405
20	4	1	5.164947	6.640575	5.485294	6.583001	6.021457	5.097809	5.640439	0.88864143	0.27297432	-0.566641466	-0.040206937	0.54761225	-0.504033322	1.05907013
25	5	1	5.130913	4.438954	5.311976	6.173483	3.726906	6.246301	4.886199	1.09745357	-1.16650463	-0.746911310	1.206738733	-1.68721988	0.419709895	0.23349876
30	6	1	6.014382	2.939461	3.540898	6.626733	6.748707	6.872268	7.002733	3.19590120	-2.72968034	-0.840795801	0.674028669	1.67432651	0.839687084	-1.42836844

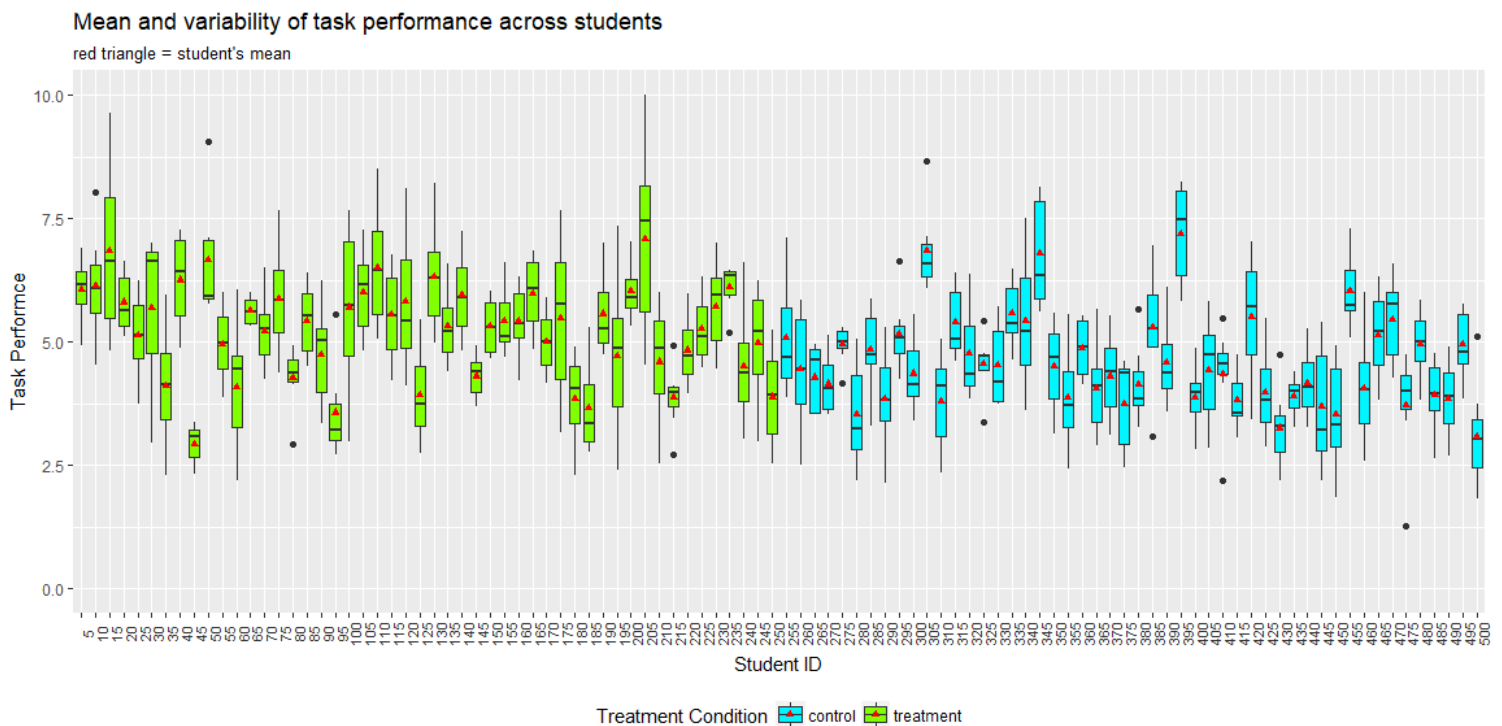
kid_id	txcond	week	perform	interest
5	1	0	6.332146	0.362666956
5	1	1	6.507988	1.411404645
5	1	2	6.888969	1.367544315
5	1	3	5.563810	-0.407176147
5	1	4	6.152750	0.762541731
5	1	5	5.984029	-0.651200823
5	1	6	4.932417	-1.473655779
10	1	0	4.531640	-1.201665796
10	1	1	5.602623	-0.148720478
10	1	2	6.851863	1.797062429
10	1	3	6.076729	0.104808718
10	1	4	5.586186	-0.802837973
10	1	5	6.255608	0.230374964
10	1	6	8.042116	0.697975193
15	1	0	4.902634	-1.295201732
15	1	1	6.072946	-1.054680218
15	1	2	4.830770	-1.941607490
15	1	3	6.623872	-1.269560593
15	1	4	7.695781	-0.887438811
15	1	5	8.171289	-0.291915763
15	1	6	9.632302	1.410094053

## Plot Variability in Task Performance Across Students

Let's begin by assessing the degree of variability in the task performance scores across students.

Variability in task performance across students

```
ggplot(data = grow, aes(x = kid_id.f, y = perform, fill = txcond.f)) +
  geom_boxplot() +
  stat_summary(aes(y = perform, group = kid_id.f, fun.y = mean, color = "red", geom = "point", pch = 17, size = 1.5)) +
  scale_y_continuous(limits = c(0,10)) +
  scale_fill_manual(values=c("turquoise1", "chartreuse1")) +
  labs(title = "Mean and variability of task performance across students", subtitle = "red triangle = student's mean",
       x = "Student ID", y = "Task Performance", fill = "Treatment Condition") +
  theme(axis.text.x = element_text(color="grey20", size=8, angle=90, hjust=.5), legend.position = "bottom")
```



## Estimate ICC

In this setting, the intraclass correlation (ICC) tells us about the amount of variability that is due to interindividual differences as compared to intraindividual differences.

ICC for performance

```
mod1 = lmer(perform ~ 1 + (1 | kid_id), REML = TRUE, data = grow)
summary(mod1)

icc.lmer <- function(mod1) {
  vars <- as.data.frame(VarCorr(mod1))[4]
  total <- sum(vars)
  tau00 <- vars[1,1]
  icc <- tau00/total
  return(icc)
}

icc.lmer(mod1)
```

```
summary from lme4 is returned
some computational error has occurred in lmerTest
Linear mixed model fit by REML ['lmerMod']
Formula: perform ~ 1 + (1 | kid_id)
Data: grow
```

REML criterion at convergence: 2166.1

scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.6469	-0.6424	-0.0085	0.5990	3.2746

Random effects:

Groups	Name	Variance	Std.Dev.
kid_id	(Intercept)	0.8013	0.8951
	Residual	0.9821	0.9910

Number of obs: 700, groups: kid\_id, 100

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	4.90455	0.09703	50.54

Notice that if lmerTest is loaded, it will give the error in red when you estimate an empty model. You can ignore this, it's just confused because there are no predictors in the model for which to compute p-values.

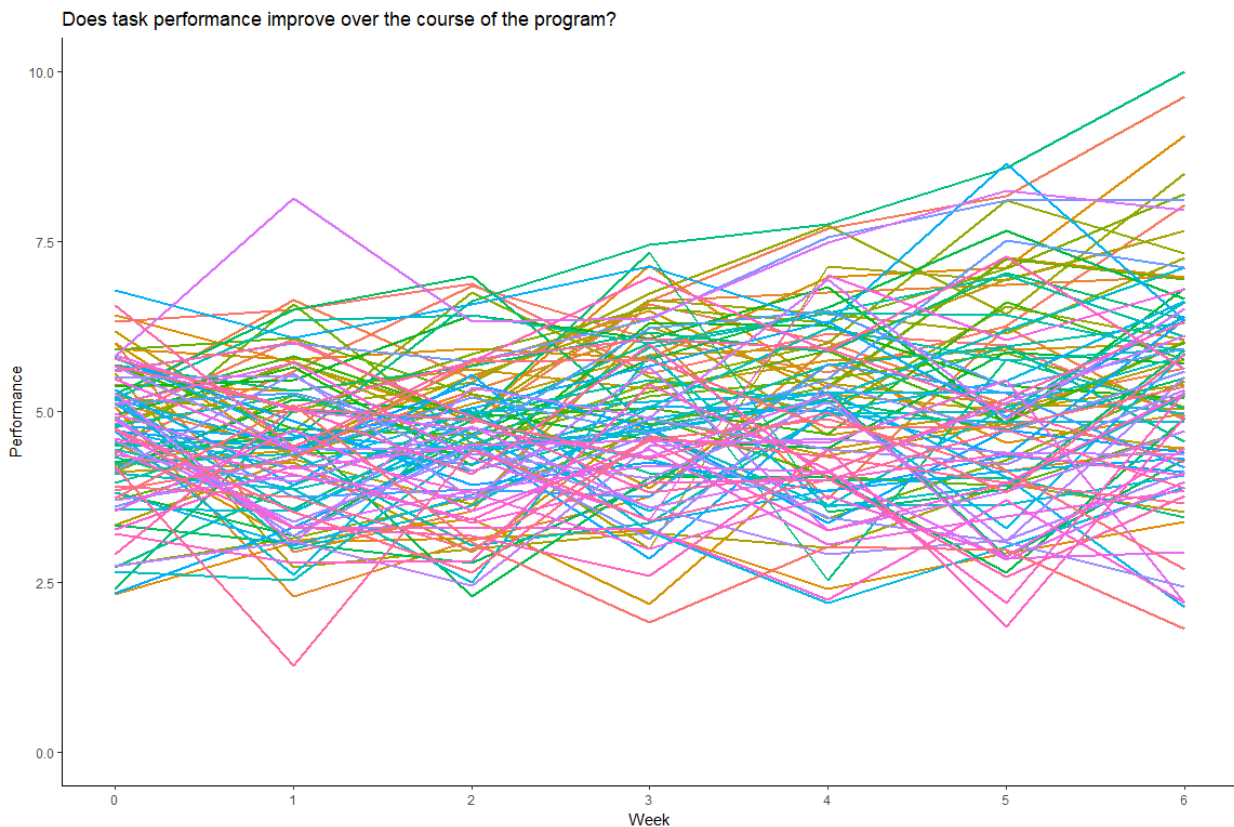
[1] 0.4492853

## How do Students Change over Time on Task Performance?

As we begin to assess change over time, a useful starting point is a spaghetti plot — that is a plot that shows the trajectory of scores across time for all people in the study.

Create a spaghetti plot to determine if change in performance over time

```
ggplot(data = grow, aes(x = week, y = perform, group = kid_id.f, color = kid_id.f)) +  
  geom_line(size=1) +  
  theme_classic() +  
  scale_y_continuous(limits = c(0,10)) +  
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +  
  guides(color = FALSE) +  
  labs(title = "Does task performance improve over the course of the program?",  
        x = "Week", y = "Performance")
```

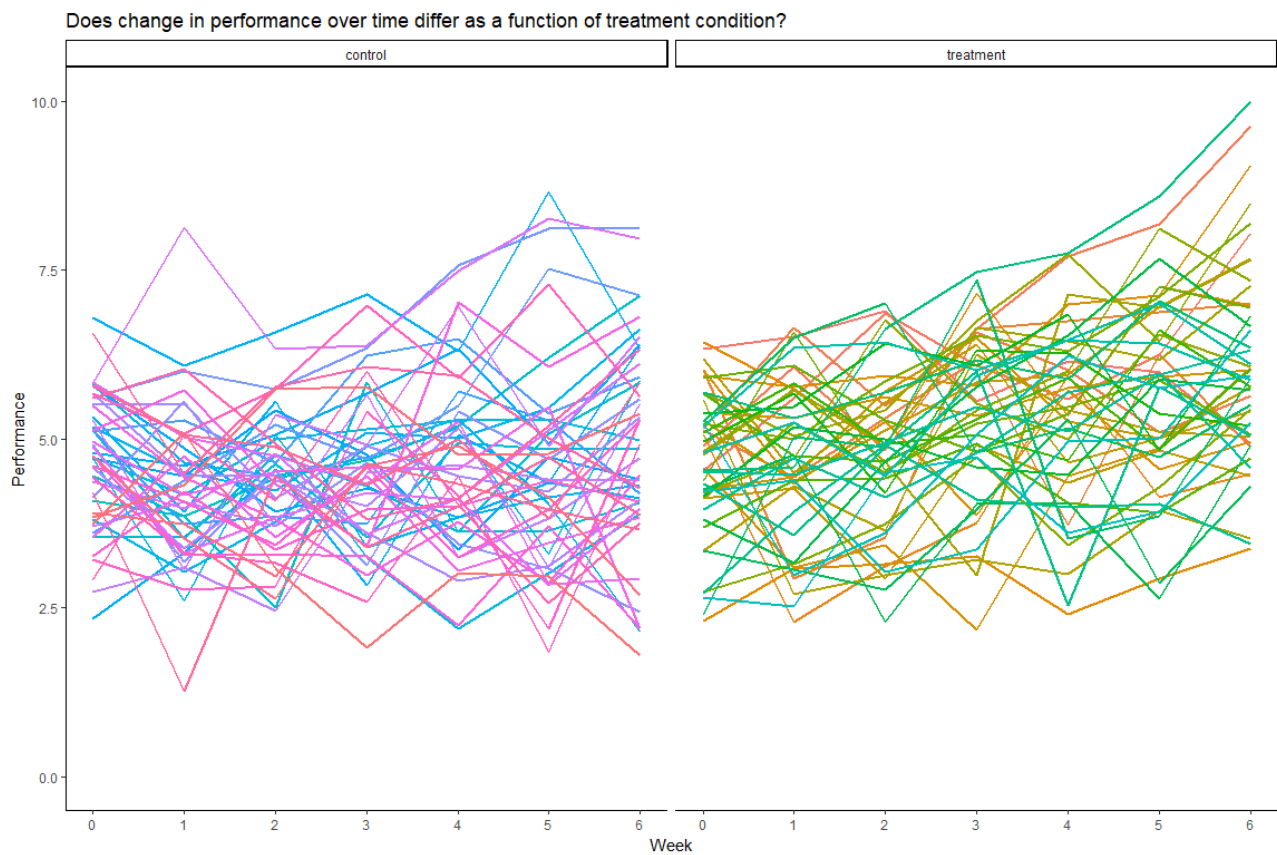


## Consider that the Growth Trajectories Might Differ by Condition

Since condition differences are an important aspect of the study, we can also examine the trajectories by treatment condition.

Create a spaghetti plot to determine if growth trajectories differ by condition

```
ggplot(data = grow, aes(x = week, y = perform, group = kid_id.f, color = kid_id.f)) +
  geom_line(size=1) +
  facet_wrap(~txcond.f) +
  theme_classic() +
  scale_y_continuous(limits = c(0,10)) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  guides(color = FALSE) +
  labs(title = "Does change in performance over time differ as a function of treatment condition?",
       x = "Week", y = "Performance")
```



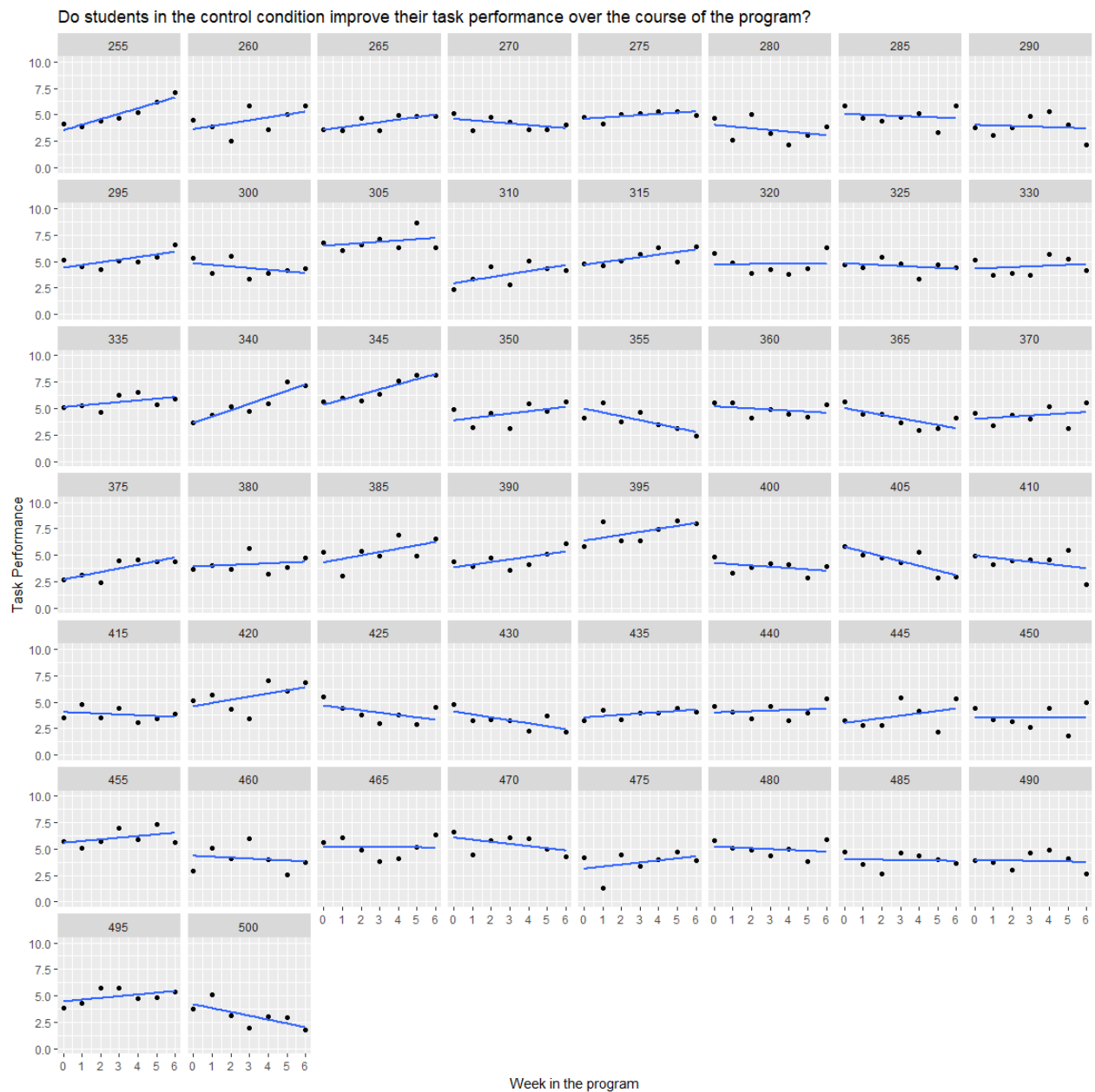


## Examine Growth in Task Performance for the Control Condition

Individual plots are also a useful preliminary assessment of the data, let's create individual plots by condition.

Individual growth curves for students in the control condition

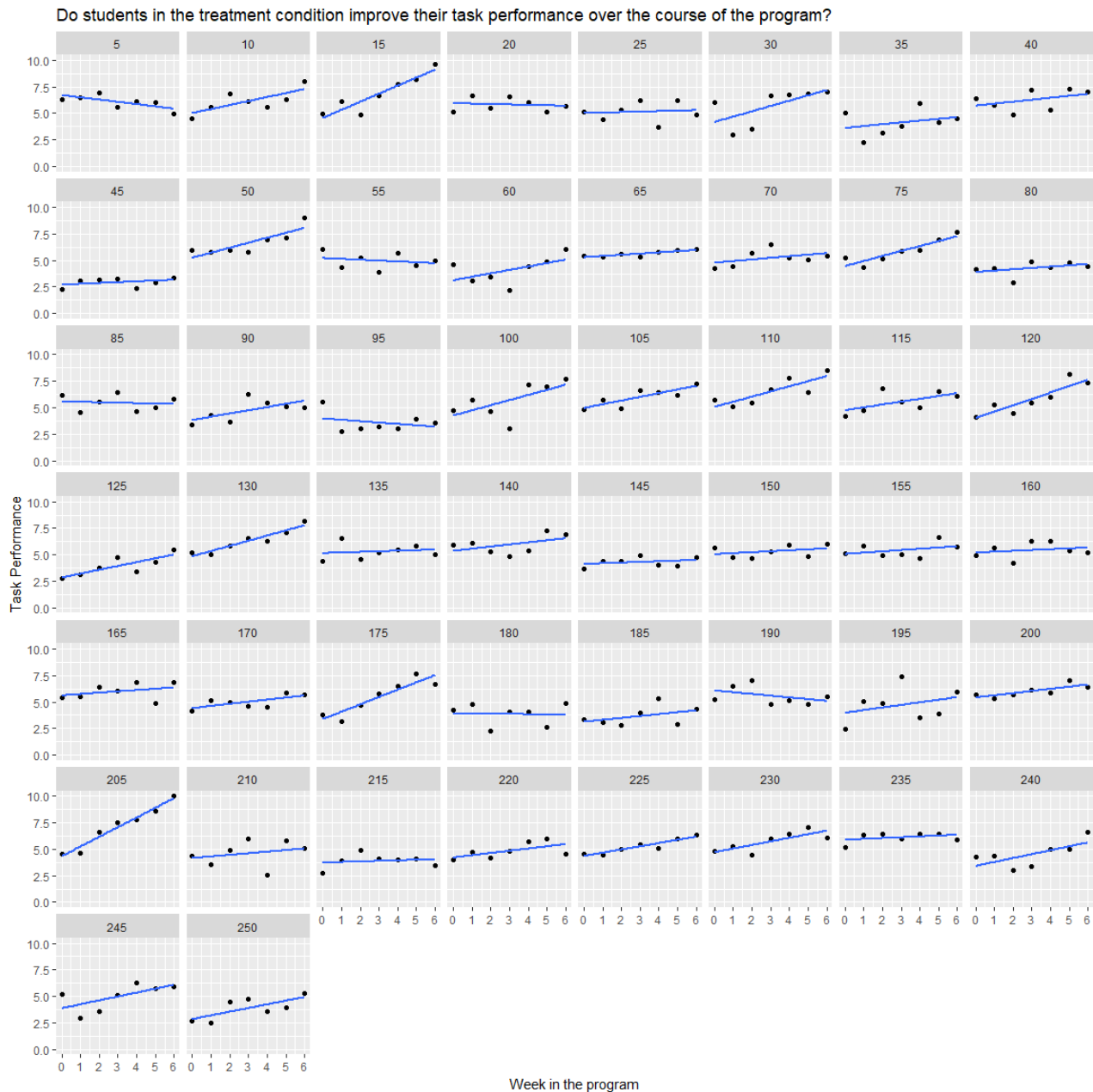
```
ggplot(data = c, aes(x = week, y = perform)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(limits = c(0,10)) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  facet_wrap(~kid_id) +
  labs(title = "Do students in the control condition improve their task performance over the course of the program?",
       x = "Week in the program", y = "Task Performance")
```



## Examine Growth in Task Performance for the Treatment Condition

Individual growth curves for students in the treatment condition

```
ggplot(data = t, aes(x = week, y = perform)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(limits = c(0,10)) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  facet_wrap(~kid_id) +
  labs(title = "Do students in the treatment condition improve their task performance over the course of the program?",
       x = "Week in the program", y = "Task Performance")
```



## A Few Additional Descriptive Summaries

Another useful descriptive technique is to fit a linear regression model to each person's data individually, and then plot the individual intercepts and slopes.

Group data set by student

```
by_kid <- grow %>%
  group_by(txcond.f, kid_id.f)
```

Fit SLR for each student and get parameter estimates

```
estimates <- do(by_kid,
  tidy(lm(perform ~ week, data = .), conf.int = TRUE, conf.level = .95))
```

kid_id.f	term	estimate	std.error	statistic	p.value	conf.low	conf.high
255	(Intercept)	3.528144631	0.29377368	12.00973684	7.061624e-05	2.772975336	4.28331393
255	week	0.517786255	0.08147816	6.35490855	1.425371e-03	0.308339977	0.72723253
260	(Intercept)	3.628095089	0.80427266	4.51102628	6.334917e-03	1.560646408	5.69554377
260	week	0.275385852	0.22306510	1.23455373	2.718489e-01	-0.298021242	0.84879295

Fit SLR for each student and get model summaries

```
model_summaries <- do(by_kid,
  glance(lm(perform ~ week, data = .)))
```

kid_id.f	r.squared	adj.r.squared	sigma	statistic	p.value
255	0.8898311088	0.867797331	0.4311419	4.038486e+01	0.001425371
260	0.2336134589	0.080336151	1.1803496	1.524123e+00	0.271848852

Fit SLR for each student and get case level statistics

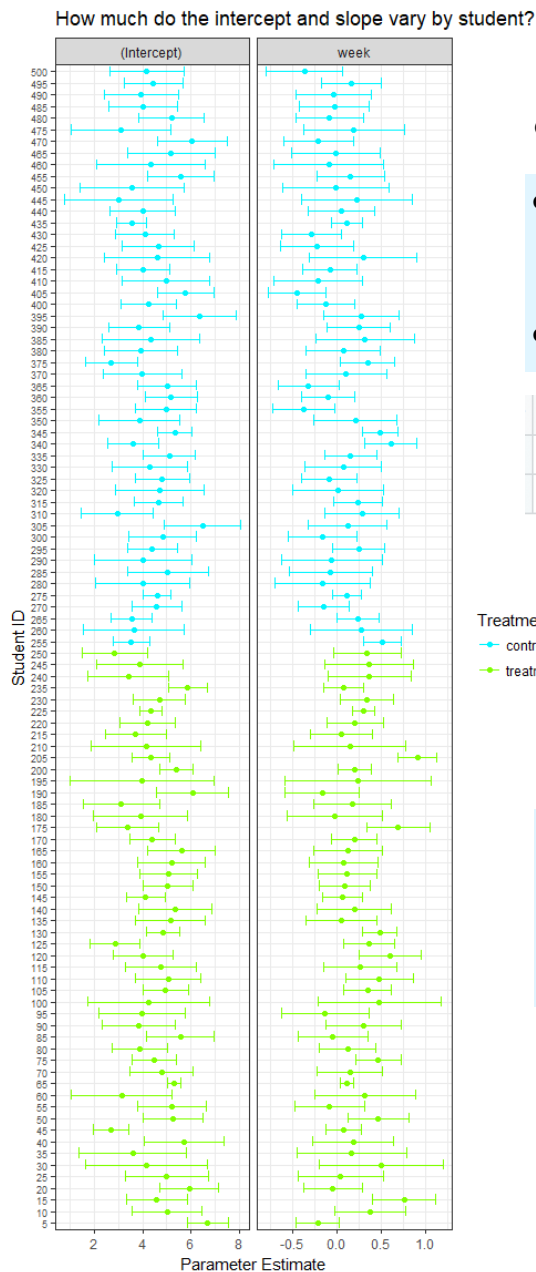
```
predicted <- do(by_kid,
  augment(lm(perform ~ week, data = .)))
```

kid_id.f	perform	week	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid
255	4.096027	0	3.528145	0.2937737	0.567882768	0.4642857	0.2861091	1.403349e+00	1.799582862
255	3.864601	1	4.045931	0.2304550	-0.181330252	0.2857143	0.4699425	4.952883e-02	-0.497638594
255	4.413985	2	4.563717	0.1821907	-0.149731805	0.1785714	0.4749008	1.595990e-02	-0.383185379
255	4.690634	3	5.081503	0.1629563	-0.390869048	0.1428571	0.4333518	7.990752e-02	-0.979229409
255	5.196467	4	5.599290	0.1821907	-0.402822950	0.1785714	0.4277485	1.155129e-01	-1.030882281
255	6.193611	5	6.117076	0.2304550	0.076534916	0.2857143	0.4799000	8.823417e-03	0.210040672
255	7.115199	6	6.634862	0.2937737	0.480336371	0.4642857	0.3531054	1.004013e+00	1.522154131

## Plot Intercept and Slope for Growth Across Students

Plot individual intercepts and slopes across students

```
ggplot(data = estimates, aes(x = estimate, y = kid_id.f, group = txcond.f, color = txcond.f)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  facet_wrap(~ term, scales = "free_x") +
  scale_color_manual(values=c("turquoise1", "chartreuse1")) +
  theme_bw() +
  theme(axis.text.y = element_text(size = 7)) +
  labs(title = "How much do the intercept and slope vary by student?",
       x = "Parameter Estimate", y = "Student ID", color = "Treatment Condition")
```



Calculate the average intercept and slope

```
avg_estimates <- estimates %>%
  group_by(term) %>%
  summarize(avg_estimate = mean(estimate))
```

**avg\_estimates**

term	avg_estimate
(Intercept)	4.4731029
week	0.1438155

Treatment Condition  
 — control  
 — treatment

Calculate the average intercept and slope by treatment condition

```
txcond_avg_estimates <- estimates %>%
  group_by(term, txcond.f) %>%
  summarize(avg_estimate = mean(estimate))
```

**txcond\_avg\_estimates**

term	txcond.f	avg_estimate	var_estimate
(Intercept)	control	4.41380692	0.69983185
(Intercept)	treatment	4.53239884	0.82063674
week	control	0.05191842	0.05558445
week	treatment	0.23571256	0.05483272

## Fit an Unconditional Growth Model

Now that we've explored the data, let's fit an unconditional linear growth model. This is a growth model with time as the only predictor. We will allow a random effect for both the intercept (predicted performance at week 0) and the slope (rate of change in performance over time).

Unconditional growth model for task performance

```
mod2 = lmer(perform ~ 1 + week + (1 + week | kid_id), REML = TRUE, data = grow)
summary(mod2)
```

```
Linear mixed model fit by REML
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']
Formula: perform ~ week + (1 + week | kid_id)
Data: grow
```

REML criterion at convergence: 2038.4

Scaled residuals:

```
      Min       1Q   Median       3Q      Max
-3.1997 -0.5740  0.0449  0.6342  3.0608
```

Random effects:

```
Groups   Name             Variance Std.Dev. Corr
kid_id   (Intercept)  0.42532   0.6522
          week         0.03774   0.1943   0.10
Residual             0.71247   0.8441
Number of obs: 700, groups: kid_id, 100
```

Fixed effects:

```
      Estimate Std. Error    df t value Pr(>|t|)
(Intercept)  4.47310    0.08695 99.00000  51.442 < 2e-16 ***
week         0.14382    0.02514 99.00000   5.722 1.13e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Correlation of Fixed Effects:

```
(Intr)
week -0.292
```

The intercept (4.47) is the average performance score at the very start of the program (week = 0). The slope for week (.14) is the average change in performance for each additional week of the program. Both are significantly different from zero. We also see quite a lot of variability both in where people start (.43) and how people change (.04).

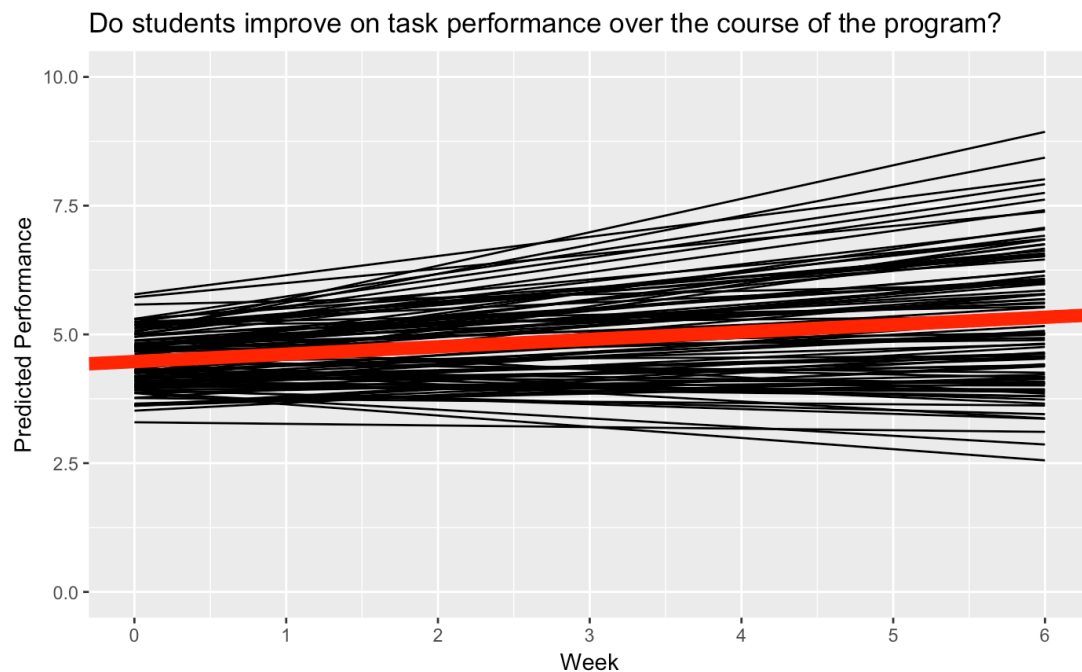
term	avg_estimate
(Intercept)	4.4731029
week	0.1438155

## Plot Model Fitted Growth Trajectories

Model fitted growth trajectories

```
# Get predicted values
mod2.plot <- add_predictions(data = grow, model = mod2)

# Make plot
ggplot(data = mod2.plot, aes(x = week, y = pred, group = kid_id.f)) +
  geom_line() +
  geom_abline(intercept = 4.4731, slope = .1438, color="red", size=3) +
  scale_y_continuous(limits = c(0,10)) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  labs(title = "Do students improve on task performance over the course of the program?",
       x = "Week", y = "Predicted Performance")
```



## Fit a Model to Determine if the Growth Trajectories Differ by Condition

We noted substantial variability in both the intercept and slope. We might then wonder if there are Level 2 predictors that can account for this variability. We will consider the role that treatment condition might play.

Conditional growth model with condition as a predictor

```
mod3 = lmer(perform ~ 1 + week + week*txcond + (1 + week | kid_id), REML = TRUE, data = grow)
summary(mod3)
```

Linear mixed model fit by REML  
t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']  
Formula: perform ~ week + week \* txcond + (1 + week | kid\_id)  
Data: grow

REML criterion at convergence: 2026.2

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.2149	-0.5673	0.0397	0.6348	3.0318

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
kid_id	(Intercept)	0.42944	0.6553	
	week	0.02976	0.1725	0.06
	Residual	0.71247	0.8441	

Number of obs: 700, groups: kid\_id, 100

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	4.41381	0.12331	98.00000	35.795	< 2e-16 ***
week	0.05192	0.03323	98.00000	1.562	0.121408
txcond	0.11859	0.17438	98.00000	0.680	0.498066
week:txcond	0.18379	0.04699	98.00000	3.911	0.000169 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr) week	txcond
week	-0.342	
txcond	-0.707	0.242
week:txcond	0.242	-0.707

term	txcond.f	avg_estimate	var_estimate
(Intercept)	control	4.41380692	0.69983185
(Intercept)	treatment	4.53239884	0.82063674
week	control	0.05191842	0.05558445
week	treatment	0.23571256	0.05483272

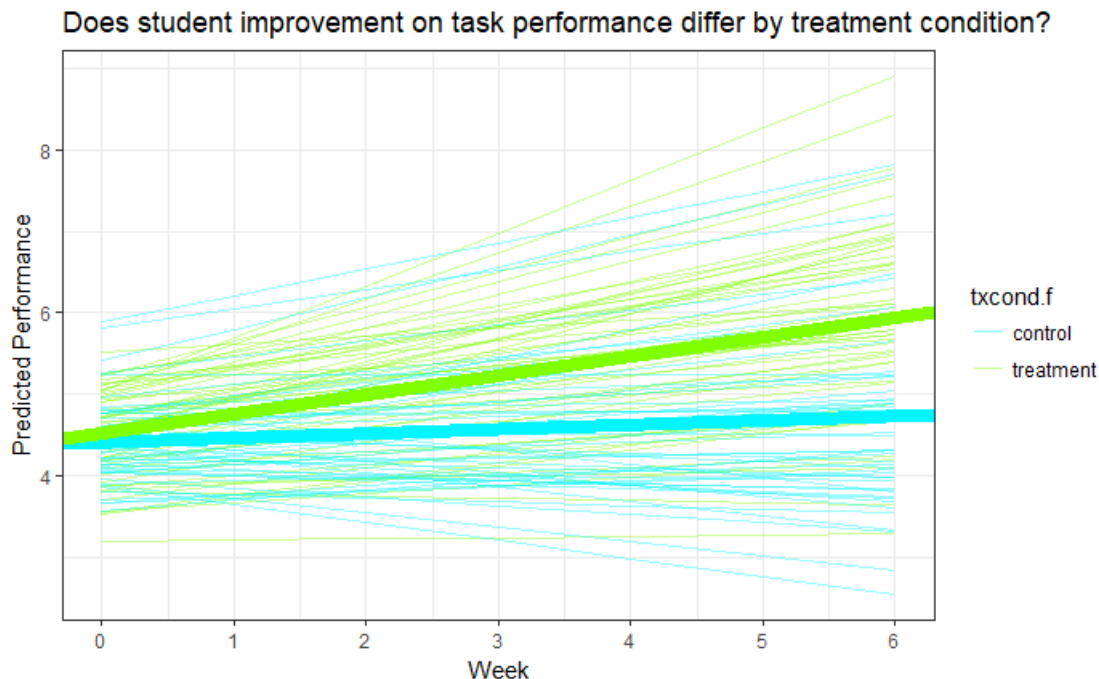
Can you map the parameter estimates from the fixed effects above onto these means?

## Plot Model Fitted Growth Trajectories

Model fitted growth trajectories

```
# Get predicted values
mod3.plot <- add_predictions(data = grow, model = mod3)

# Make plot
ggplot(data = mod3.plot, aes(x = week, y = pred, group = kid_id.f, color = txcond.f)) +
  geom_line(alpha = .4) +
  geom_abline(intercept = 4.41381, slope = .05192, color="turquoise1", size=3) +
  geom_abline(intercept = (4.41381 + .11859), slope = (.05192 + .18379), color="chartreuse1", size=3) +
  theme_bw() +
  scale_color_manual(values=c("turquoise1", "chartreuse1")) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  labs(title = "Does student improvement on task performance differ by treatment condition?",
       x = "Week", y = "Predicted Performance")
```



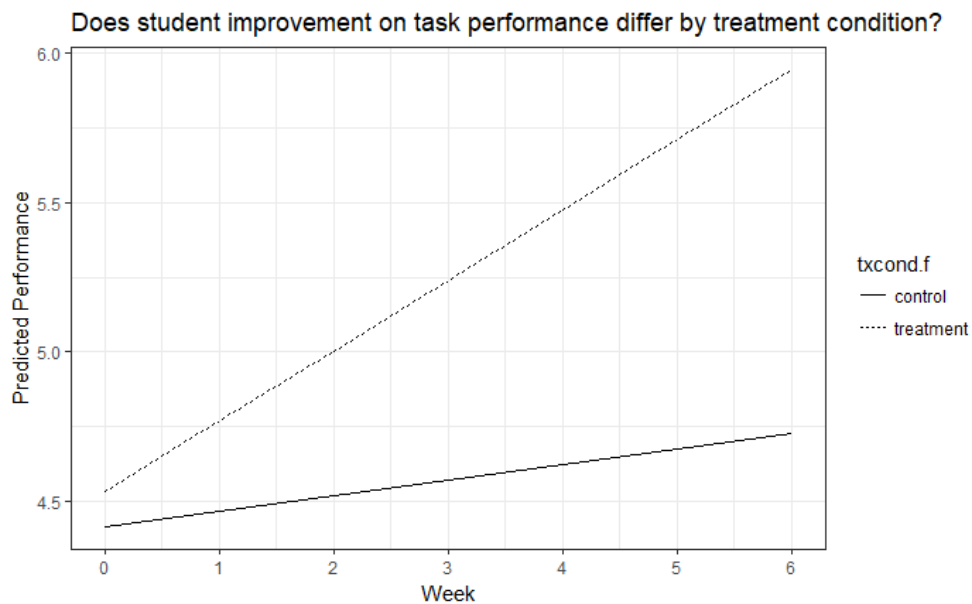


## Simple Plot with Just Fixed Effects

Last, here is a simple plot to consider.

Simple plot of trajectories by condition

```
ggplot(data = grow, aes(x = week, y = perform, shape = txcond.f)) +  
  stat_summary(aes(y = fitted(mod3), linetype = txcond.f), fun.y = mean, geom = "line") +  
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +  
  theme_bw() +  
  labs(title = "Does student improvement on task performance differ by treatment condition?",  
       x = "Week", y = "Predicted Performance")
```

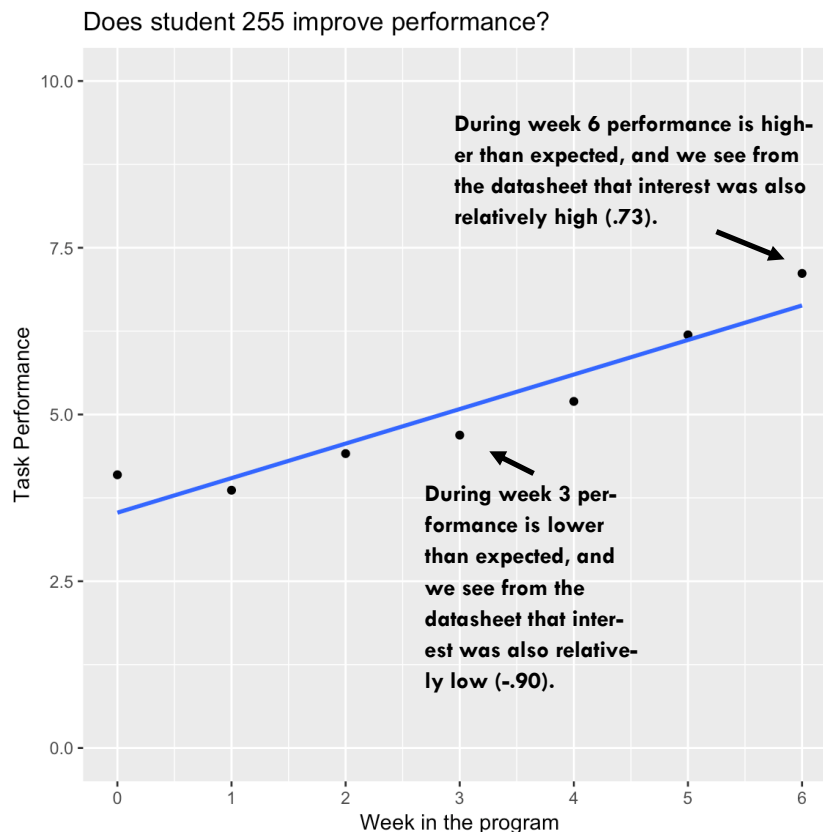


## Time-Varying Covariates in a Growth Model

In addition to Level 2 predictors of the growth trajectories, we might also be interested in considering Level 1 predictors. For example, we might want to know if interest in the substantive subject of the task is associated with better performance. In this example, we will consider only students in the control condition.

Plot of one student's data

```
ggplot(data = subset(c, kid_id == 255), aes(x = week, y = perform)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(limits = c(0,10)) +
  scale_x_continuous(limits = c(0,6), breaks = c(0,1,2,3,4,5,6)) +
  labs(title = "Does student 255 improve performance?",
       x = "Week in the program", y = "Task Performance")
```



With a time-varying covariate, we seek to determine if the covariate can predict the residual between the best fit line and the predicted score at any given measurement occasion.

kid_id	txcond	week	perform	interest
255	0	0	4.096027	-0.159077985
255	0	1	3.864601	-0.997083675
255	0	2	4.413985	-1.465740094
255	0	3	4.690634	-0.900745194
255	0	4	5.196467	-0.524227659
255	0	5	6.193611	0.318003583
255	0	6	7.115199	0.728927321

## Is Interest Associated with Task Performance After Adjusting for Linear Growth

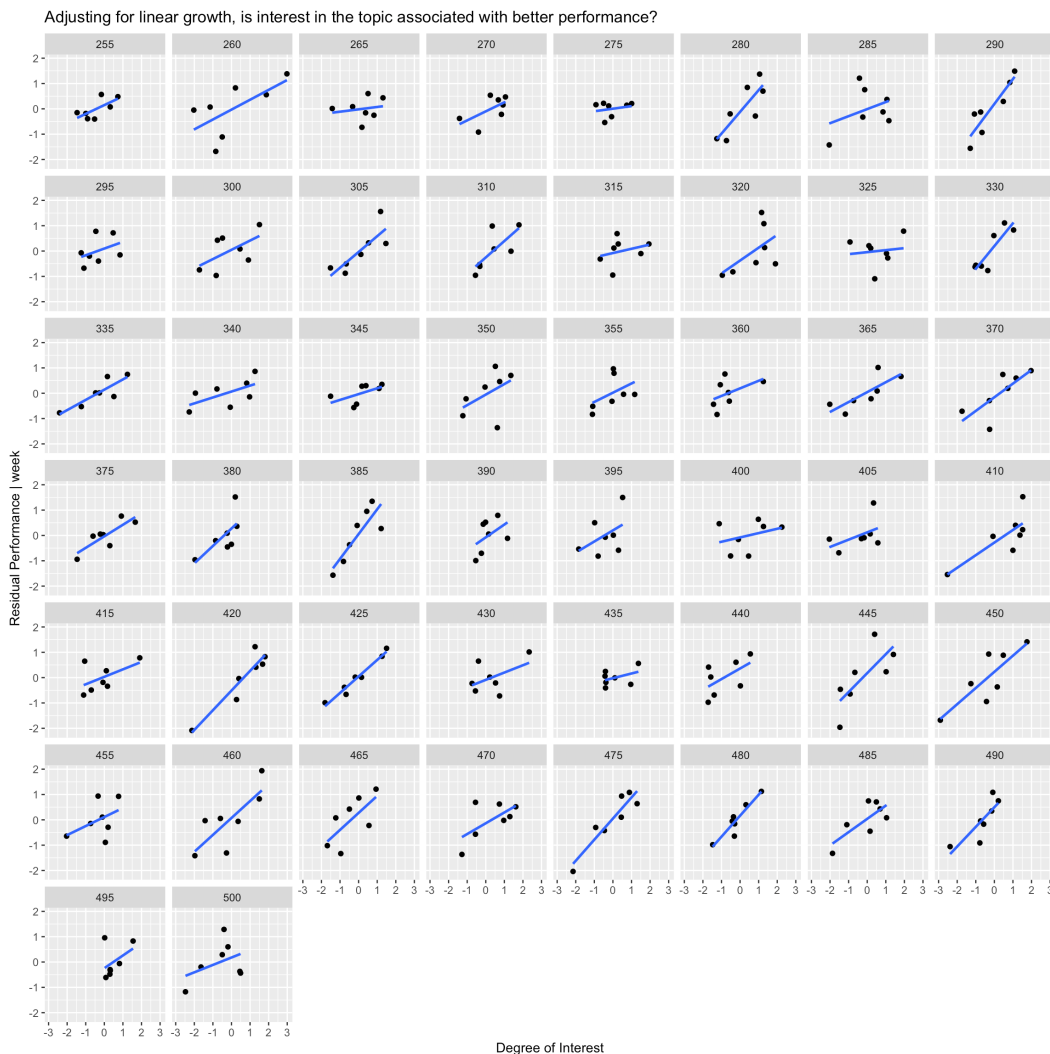
In essence, we're predicting the residuals after time is accounted for by the model. Let's compute the residuals manually and see if they seem to be correlated with interest.

Scatterplot of residualized performance (adjusting for linear growth) and interest

```
resid <- predicted %>%
  filter(txcond.f == "control") %>%
  select(kid_id.f, week, txcond.f, .resid, .fitted)

tvc_eff <- full_join(resid, c, by = c("kid_id.f", "week"))

ggplot(data = tvc_eff, aes(x = interest, y = .resid)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~kid_id) +
  labs(title = "Adjusting for linear growth, is interest in the topic associated with better performance?",
       x = "Degree of Interest", y = "Residual Performance | week")
```



**It does appear that students performed better on tasks that were of interest to them.**

## Include a Time Varying Covariate

Let's build on our linear growth model by including interest as a Level 1 predictor.

Fit growth model for control condition with interest (a time-varying covariate)

```
mod4 = lmer(perform ~ 1 + week + interest + (1 + week | kid_id), REML = TRUE, data = c)
summary(mod4)
```

Linear mixed model fit by REML

t-tests use Satterthwaite approximations to degrees of freedom ['lmerMod']

Formula: perform ~ week + interest + (1 + week | kid\_id)

Data: c

REML criterion at convergence: 842.3

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-2.72298	-0.56758	-0.05115	0.56755	2.47939

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
kid_id	(Intercept)	0.51270	0.7160	
	week	0.02949	0.1717	0.01
	Residual	0.37448	0.6119	

Number of obs: 350, groups: kid\_id, 50

Fixed effects:

	Estimate	Std. Error	df	t value	Pr(> t )
(Intercept)	4.44437	0.11720	48.83000	37.922	<2e-16 ***
week	0.04754	0.02928	48.99000	1.623	0.111
interest	0.55340	0.03509	284.02000	15.771	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr) week
week	-0.227
interest	0.017 -0.009

Interest is a positive and significant predictor, adjusting for week in the program, we see that each 1-unit increase in interest is associated with a .55 unit increase in performance.