Heirarchical Linear Modeling

Neba Nfonsang
University of Denver

In [1]:

```
install.packages("lmerTest") # to get p-values
library(lmerTest)
library(tidyverse, quietly=T)
library("lme4", quietly=T)

# useful for computing ICC
library("merTools", quietly=T)
```

```
package 'lmerTest' successfully unpacked and MD5 sums checke
d

The downloaded binary packages are in
        C:\Users\nnfon\AppData\Local\Temp\RtmpcXKKJv\downloa
ded_packages


Loading required package: Matrix
Loading required package: lme4

Attaching package: 'lmerTest'

The following object is masked from 'package:lme4':

    lmer

The following object is masked from 'package:stats':

    step

Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr
Conflicts with tidy packages -------------------------------
--------------------
expand(): tidyr, Matrix
filter(): dplyr, stats
lag():    dplyr, stats

Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
arm (Version 1.9-3, built: 2016-11-21)

Working directory is C:/Users/nnfon/Desktop/INFO 4340 - Data
Mining and Viz
```

In [2]:

```
dat <- read.csv("NEW_NELS_CLEAN.csv")
names(dat) <- tolower(names(dat))
head(dat)
```

stu_id	sch_id	mathscore	family_inc	urban
7898401	81	48.26	11	1
7898402	81	37.66	12	1
7898406	81	60.32	12	1
7898407	81	49.96	8	1
7898418	81	51.64	9	1
7898424	81	57.29	9	1

Introduction to Heirarchical Linear Modeling (HLM)

- In HLM, there is regression equation at level 1, which models a y variable and regression equations at level 2 which model the intercept and slope
- The full model has a predictor at level 1 and a predictor at level 2 and the intercepts ($\beta 0j$) and slopes($\beta 1j$) are allow to vary across level 2 units.
- The intercepts has fixed components $\gamma 00$ = mean of intercepts and a variance component tau00 = var($\beta 0j$) = var(u0j) where u0j is the random variation of the intercept of a unit
- The slope has fixed components $\gamma 10$ = mean of slopes, and a variance component tau11 = var($\beta 1j$) = var(u1j) where u1j is the random variation of the slope of a unit.
- The covariance between slopes and intercepts is tau01 or tau10 = cov($\beta 0j$, $\beta 1j$)

Tips for Formulating a Heirarchical Linear Model

- There is always a fixed intercept $\gamma 00$,
- If a predictor exists at level 1, there is always a fixed slope, $\gamma 10$,
- We can make the intercepts or slopes vary across level 2 units by including the random variation of the intercept u0j or random variation of the slopes u1j

Tips for Formulating a Heirarchical Linear Model in R

- Use 1 to indicate the intercept
- To make the intercept vary, use 1 and the group (or unit) you want the effect to vary across: (1|level2_unit)
- To make the slope vary, use the variable associated with the slope and the unit you want the effect to vary across: (variable_name|level2_unit)

Full Model

A full model has predictors at level 1 and level 2 as well as both fixed and random compoents for the slope and intercept.

Level-1 Model

```
scienceij = β0j + β1j*(family_inc_ij) + rij
```

Level-2 Model

```
β0j = γ00 + γ01*(schtypj) + u0j

β1j = γ10 + γ11*(schtypj) + u1j
```

Model with Predictor at Level 1

- Fixed Slopes only, Random Intercept

Level-1 Model

```
 scienceij = β0j + β1j*(family_inc_ij) + rij
```

Level-2 Model

```
β0j = γ00 + u0j

β1j = γ10
```

In [3]:

```
model.1 <- lmer(mathscore ~ 1 + family_inc + (1|sch_id), data=dat, REML =
summary(model.1)
```

```
Linear mixed model fit by maximum likelihood t-tests use Sat
terthwaite
  approximations to degrees of freedom [lmerMod]
Formula: mathscore ~ 1 + family_inc + (1 | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 59944.7  59972.8 -29968.3  59936.7     8300

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.10428 -0.73893 -0.01012  0.73980  2.91628

Random effects:
 Groups   Name        Variance Std.Dev.
 sch_id   (Intercept) 15.35    3.918
 Residual             72.44    8.511
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 4.069e+01  4.261e-01 5.017e+03   95.50   <2e-16
***
family_inc  1.060e+00  4.053e-02 7.983e+03   26.15   <2e-16
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Correlation of Fixed Effects:
           (Intr)
family_inc -0.903
```

Interpretation

- Random Effect: The random effect estimate for this model is the variance of the intercept across different schools and this variance is 15.40.
- Fixed Effect: The fixed effect of the intercept is 40.68

Coefficients

- We can see that the intercepts vary and the coefficients are fixed.

In [4]:

```
coef <- coef(model.1)$sch_id
head(coef)
```

	(Intercept)	family_inc
81	42.12045	1.059821
391	39.26540	1.059821
421	36.19551	1.059821
461	39.28299	1.059821
615	37.29476	1.059821
750	49.74479	1.059821

In [5]:

```
# change column names
names(coef) <- c("intercept", "slope_family_inc")
head(coef)
```

	intercept	slope_family_inc
81	42.12045	1.059821
391	39.26540	1.059821
421	36.19551	1.059821
461	39.28299	1.059821
615	37.29476	1.059821
750	49.74479	1.059821

Significant Test For a Fixed Effect

Raudenbush and Bryk (2002), and therefore the HLM software, use a *t*-distribution to evaluate this ratio (Fotiu, 1989).

$$t = \frac{\hat{\gamma}_h}{S.E.(\hat{\gamma}_h)}$$

where γ_h is either the intercept or slope coefficient and $S.E.(\gamma_h)$ is the standard error estimate.[1] The fixed effects hypothesis tests (whether for level-1 or level-2 predictors) used by the HLM software use a degrees of freedom based on the number of level-2 units (i.e., number of groups).

$$df = N - q - 1,$$

N= number of groups

q= number of predictors

http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf
(http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf)

- Test of significance for a fixed effect is basically testing whether the mean estimate for the intercept or slope significantly differ from 0

In [6]:

```
intercepts = coef$intercept
t.test(intercepts)
```

```
        One Sample t-test

data:  intercepts
t = 414.03, df = 853, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 40.49337 40.87912
sample estimates:
mean of x
 40.68624
```

In [7]:

```
mean(intercepts)
```

40.6862423006384

Test of Significance for Random Effects

The chi-square test. The chi-square test used in the HLM package is based on the deviation of group means from the grand mean, given in Raudenbush and Bryk (2002, p.64) as:

$$\chi^2 = \frac{\sum_j \left(\hat{\beta}_{qj} - \hat{\gamma}_{q0} - \sum_{s=1}^{S_q} \hat{\gamma}_{qs} W_{sj} \right)^2}{\hat{V}_{qqj}}.$$

In the above formula, β is the group estimate (intercept or slope), γ is the average estimate (grand mean or average slope), and W is a predictor. The numerator in the equation represents the sum of squared deviations from the average value adjusting for the predictors in the model. The denominator, V_{qqj}, is a variance error estimate (i.e., square of the standard error). Degrees of freedom for this test are $J - S_q - 1$, where J is the number of groups and S_q is the number of predictors in the model (in Snijders & Bosker, 2012, this is $N - q - 1$). Small groups are omitted from the computations (the number omitted is noted in the HLM output).

[http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf (http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf)](http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf)

The variance of a random effect is basically the deviation of the slope or intercept from the grand mean or mean of intercept, adjusting for effect of the predictor at level 2.

In [8]:

```
intercepts = coef$intercept
gamma00 = mean(intercepts)
gamma01.predictor = 0  # there is no predictor at level 2 so gamma01*pred
variance_00_error = var(intercepts)/length(intercepts)

deviations = (intercepts - gamma00 - 0)^2
chi_squared = sum(deviations)/variance_00_error
chi_squared
```

728462

In [9]:

```
# df = dree of freedom = number of groups - number of level2 predictors -
pchisq(q = 728462, df =854-0-1, lower.tail = FALSE)
```

0

Model Fit Indices

- Chi-square $-2 * \ell$
- Akaike Information Criteria

$$AIC = -2 * \ell + 2K$$

- Bayesian Informaiton Criteria

$$BIC = -2 * \ell + K * Ln(N)$$

Model comparison

When we compare two models, we find the change in deviance of the models. Normally models with smaller deviance have a better fit. We can find the change in deviance and that gives a chi-statistic with a degree of freedom being equal to the difference between the parameters in the model. We can then find if that chi-squared statistics is significant.
[http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf (http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf)](http://web.pdx.edu/~newsomj/mlrclass/ho_significance.pdf)

Likelihood Ratio Test

This test that compares the deviance of two models is called the likelihood ratio test. The deviance of a model is calculated as deviance = -2*loglikelihood or

$$-2 \log \left(p(y \mid \hat{\theta}_0) \right)$$

In [10]:

```
deviance(model.1)
```

59936.6860974382

Model with Predictor at Level 1

- Random Slopes, Random Intercept

In [11]:

```
model.2 <- lmer(mathscore ~ 1  + family_inc + (family_inc|sch_id),
                data=dat, REML = FALSE)
summary(model.2)
```

```
Linear mixed model fit by maximum likelihood t-tests use Sat
terthwaite
  approximations to degrees of freedom [lmerMod]
Formula: mathscore ~ 1 + family_inc + (family_inc | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 59940.6  59982.7 -29964.3  59928.6     8298

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.14262 -0.74398 -0.01008  0.73675  2.90751

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 sch_id   (Intercept)  6.91135 2.6289
          family_inc   0.01718 0.1311   1.00
 Residual             72.44775 8.5116
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 4.060e+01  4.044e-01 1.458e+03   100.4   <2e-16
***
family_inc  1.061e+00  4.048e-02 7.210e+03    26.2   <2e-16
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Correlation of Fixed Effects:
           (Intr)
family_inc -0.895
```

In [12]:

```
anova(model.1, model.2)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
object	4	59944.69	59972.78	-29968.34	59936.69	NA	NA	NA
..1	6	59940.57	59982.72	-29964.28	59928.57	8.117954	2	0.01726668

In [13]:

```
# check how chi-squared was computed
deviance(model.1) - deviance(model.2)
```

8.11795381792035

In [14]:

```
# we can also get the p-value too
pchisq(q=452.1748, df=2, lower.tail = FALSE)
```

6.47872807522628e-99

There is a significant difference between the two models, the model with lower deviance is better, which is model 2.

Note that, we can generate a model with random slopes and random intercepts without specifying the intercepts in the model. By default, R always generates random intercepts with the lmer() function.

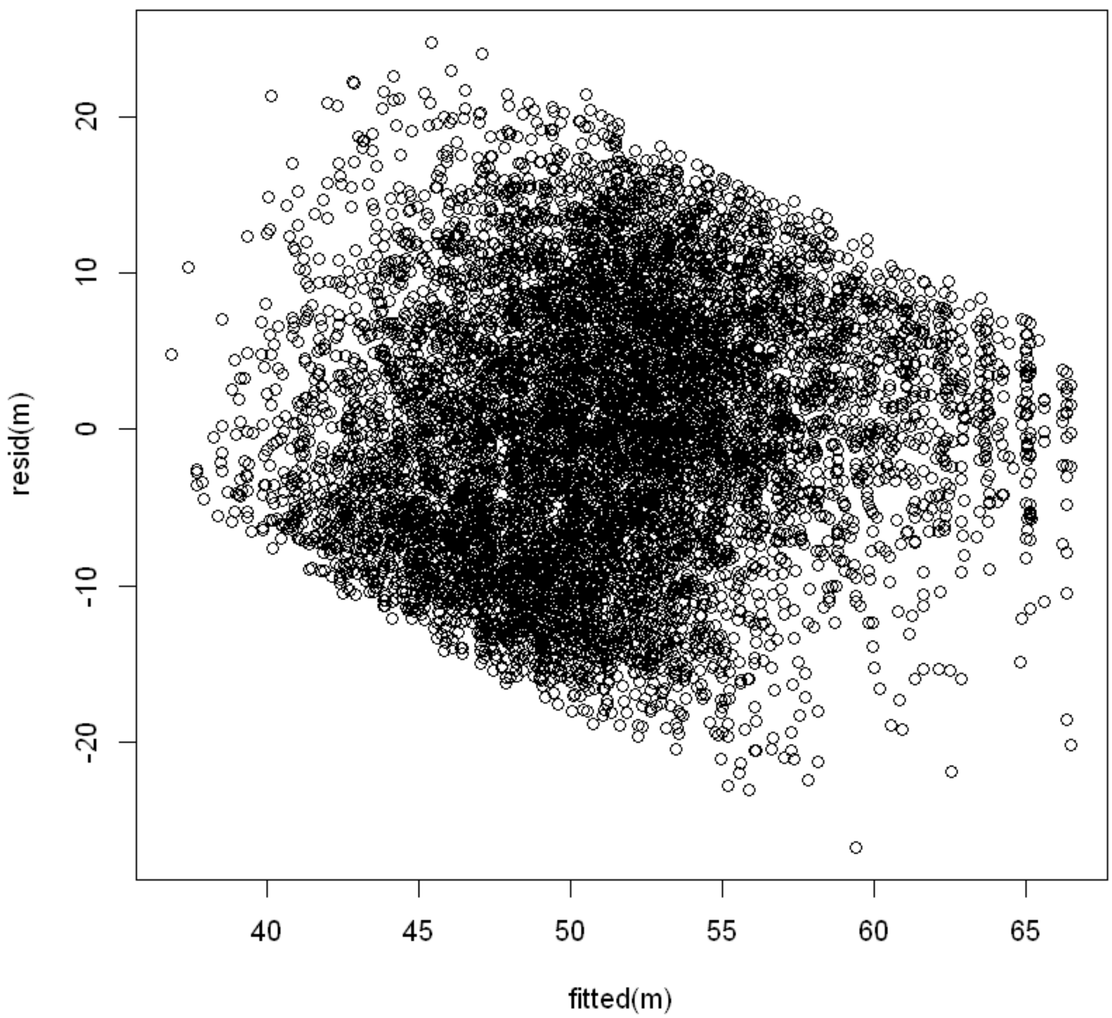
In [15]:

```
m <- lmer(mathscore ~ family_inc + (family_inc|sch_id),
                data=dat, REML = FALSE)
summary(m)
```

```
Linear mixed model fit by maximum likelihood t-tests use Sat
terthwaite
  approximations to degrees of freedom [lmerMod]
Formula: mathscore ~ family_inc + (family_inc | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 59940.6  59982.7 -29964.3  59928.6     8298

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.14262 -0.74398 -0.01008  0.73675  2.90751

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 sch_id   (Intercept)  6.91135 2.6289
          family_inc   0.01718 0.1311   1.00
 Residual             72.44775 8.5116
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 4.060e+01  4.044e-01 1.458e+03   100.4   <2e-16
***
family_inc  1.061e+00  4.048e-02 7.210e+03    26.2   <2e-16
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Correlation of Fixed Effects:
           (Intr)
family_inc -0.895
```

In [16]:

```
# check residuals
plot(resid(m) ~ fitted(m))
```



Null Model

This is a model with no predictor at level 1 or level 2. Intercepts is random. There is no slope since there is no predictor at level 1.

In [17]:

```
model.3 <- lmer(mathscore ~ 1 + (1|sch_id),
                data=dat, REML = FALSE)
summary(model.3)
```

```
summary from lme4 is returned
some computational error has occurred in lmerTest

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mathscore ~ 1 + (1 | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 60548.8  60569.9 -30271.4  60542.8     8301

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.70061 -0.75164 -0.00107  0.70824  2.93238

Random effects:
 Groups   Name        Variance Std.Dev.
 sch_id   (Intercept) 27.49    5.243
 Residual             75.43    8.685
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
            Estimate Std. Error t value
(Intercept)  50.7035     0.2259   224.5
```

Predictor at Level 1, Random Slope, Fixed intercept

In [18]:

```
model.4 <- lmer(mathscore ~ 1 + (family_inc|sch_id),
                data=dat, REML = FALSE)
summary(model.4)
```

```
summary from lme4 is returned
some computational error has occurred in lmerTest


Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mathscore ~ 1 + (family_inc | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 60390.7  60425.9 -30190.4  60380.7     8299

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.93050 -0.74194 -0.00765  0.71750  2.83936

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 sch_id   (Intercept) 92.867   9.637
          family_inc   1.012   1.006    -0.89
 Residual             71.726   8.469
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
            Estimate Std. Error t value
(Intercept)  50.1727     0.2055   244.1
```

In [19]:

```
# compare model 2 and 3 to 1
anova(model.1, model.2, model.3)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
..2	3	60548.84	60569.91	-30271.42	60542.84	NA	NA	NA
object	4	59944.69	59972.78	-29968.34	59936.69	606.151586	1	7.687611e-134
..1	6	59940.57	59982.72	-29964.28	59928.57	8.117954	2	1.726668e-02

Intra class correlation (ICC)

This is a measure of variability between groups. This is important to compute before even trying to apply a model that assumes that groups vary.

$$ICC = \rho = \frac{\tau_0^2}{\sigma^2 + \tau_0^2}$$

- Sigma squared is variance within the groups and tau squared is variance across the groups with respect to the dependent variable.
- So, ICC captures the proportion of variance in the dependent variable due to the groups. The variance in the dependent variable can be partition into variance within groups and variance across groups.

In [20]:

```
ICC(outcome = "mathscore", group = "sch_id", data=dat)
```

0.267488882772605

In [21]:

```
coef2 <- coef(model.2)$sch_id
names(coef2) <- tolower(c("intercept", "family_inc_slope"))
head(coef2)
```

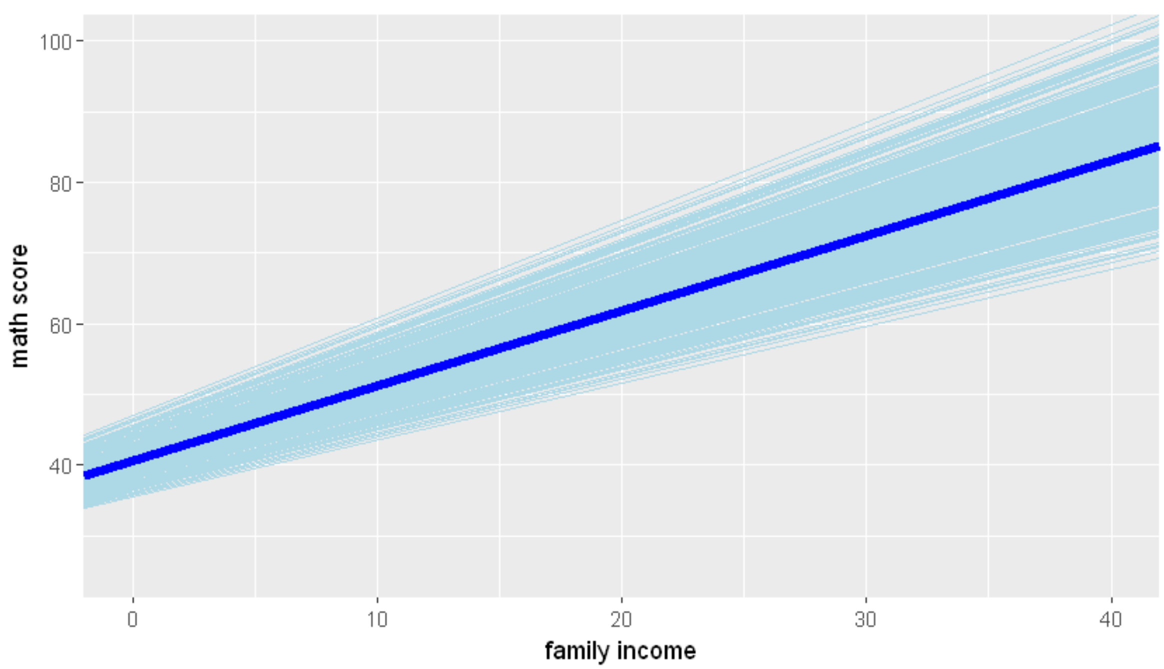
	intercept	family_inc_slope
81	41.61890	1.1114946
391	39.65244	1.0134415
421	37.43830	0.9030384
461	39.62227	1.0119368
615	38.41604	0.9517909
750	46.57756	1.3587465

Visualize Random Intercepts and Random Slopes

- Let's take a look at the random slopes and random intercept model where both the slope and intercept are allow to vary randomly.

In [22]:

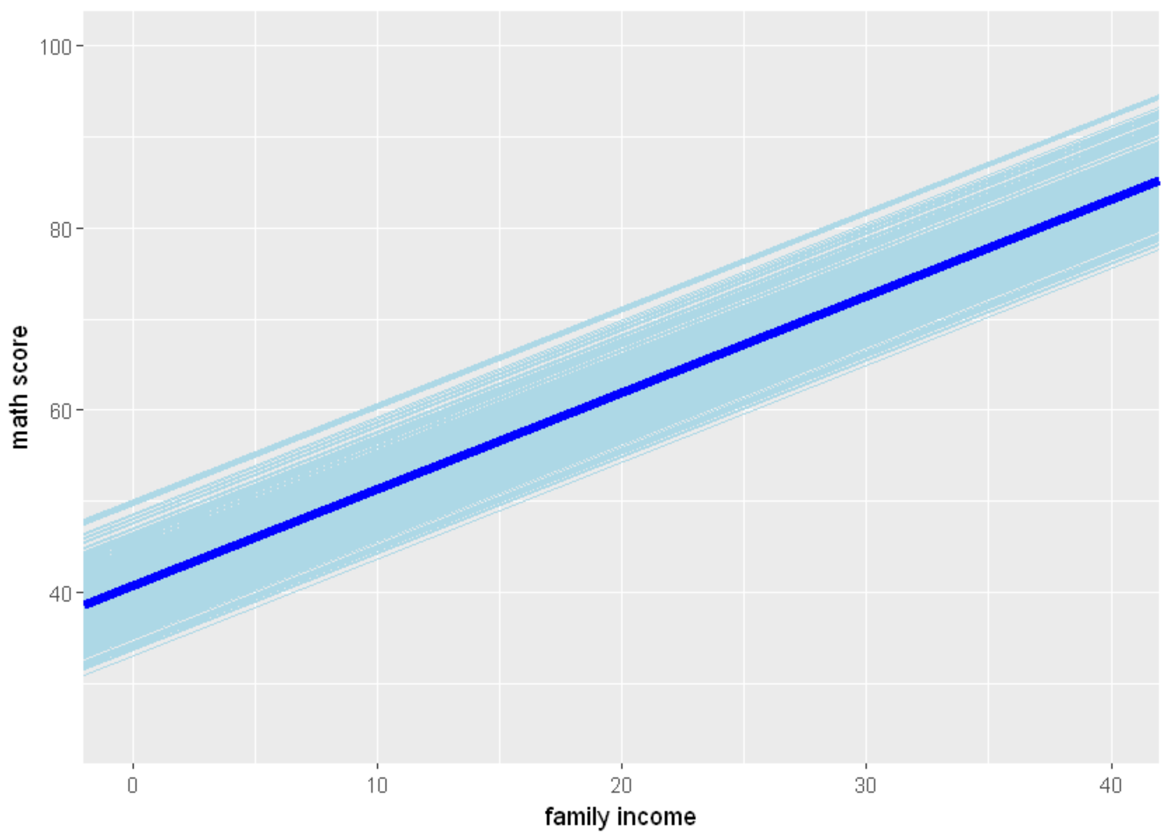
```
options(repr.plot.height=4, repr.plot.width=7)

ggplot() +
scale_x_continuous(name="family income", limits=c(0,40)) +
scale_y_continuous(name="math score", limits=c(25,100)) +
scale_linetype(name="family_inc_slope") +
geom_abline(data=coef2,
            mapping=aes(slope=family_inc_slope,
                        intercept=intercept), color="lightblue") +
geom_abline(slope=mean(coef2$family_inc_slope),
                        intercept=mean(coef2$intercept), size=2, color="b
```



Visualize Fixed Slopes and Random Intercepts

In [23]:

```
options(repr.plot.height=5, repr.plot.width=7)

ggplot() +
scale_x_continuous(name="family income", limits=c(0,40)) +
scale_y_continuous(name="math score", limits=c(25,100)) +
scale_linetype(name="family_inc_slope") +
geom_abline(data=coef,
            mapping=aes(slope=slope_family_inc,
                        intercept=intercept), color="lightblue") +
geom_abline(slope=mean(coef$slope_family_inc),
                        intercept=mean(coef$intercept), size=2, color="bl
```



How to Obtain Means of Each Unit

In [24]:

```
dat %>%
group_by(sch_id) %>%
summarize(math_score_mean=mean(mathscore)) %>%
head()
```

sch_id	math_score_mean
81	51.75667
391	43.16000
421	44.05118
461	51.87000
615	46.70235
750	62.67000

Adding a Level 2 Predict

- The lmer() can identify which variable is a level 2 predictor because there is no within group variability for a level 2 predictor. (If you try to use a level 2 predictor as a level 1 predictor, for exaple if you specify a random slope for a level 2 predictor, the model will fail due to no within group variability in the level 2 predictor).
- So, the predictor at level 2 is added to the right of the formula in the lmer() function.

Predictor at Level 1 and 2: Random Intercept, Fixed Slope Model

Level-1 Model

```
MATHSCORij = β0j + β1j*(FAMILY_Iij) + rij
```

Level-2 Model

```
β0j = γ00 + γ01*(URBANj) + u0j

β1j = γ10
```

Mixed Model

```
MATHSCORij = γ00 + γ01*URBANj + γ10*FAMILY_Iij  + u0j+ rij
```

In [25]:

```
model.5 <- lmer(mathscore ~ family_inc + urban + (1|sch_id),
                data=dat, REML = FALSE)
summary(model.5)
```

```
Linear mixed model fit by maximum likelihood t-tests use Sat
terthwaite
  approximations to degrees of freedom [lmerMod]
Formula: mathscore ~ family_inc + urban + (1 | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 59941.6  59976.7 -29965.8  59931.6     8299

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.10848 -0.74150 -0.01192  0.73562  2.92341

Random effects:
 Groups   Name        Variance Std.Dev.
 sch_id   (Intercept) 15.09    3.885
 Residual             72.47    8.513
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
              Estimate Std. Error         df t value Pr(>|t
|)
(Intercept)   41.11556    0.46980 4165.00000  87.517   <2e-1
6 ***
family_inc     1.05714    0.04057 8010.00000  26.057   <2e-1
6 ***
urban         -0.82167    0.36348  627.00000  -2.261   0.024
1 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Correlation of Fixed Effects:
           (Intr) fmly_n
family_inc -0.841
urban      -0.424  0.055
```

Predictor at Level 1: Fixed Intercept, Random Slope

- We can make the intercept fixed by adding 0 to the predictor in the parenthesis.

In [26]:

```r
model.6 <- lmer(mathscore ~ family_inc +  (0+family_inc|sch_id),
                data=dat, REML = FALSE)
summary(model.6)
```

```
Linear mixed model fit by maximum likelihood t-tests use Sat
terthwaite
  approximations to degrees of freedom [lmerMod]
Formula: mathscore ~ family_inc + (0 + family_inc | sch_id)
   Data: dat

     AIC      BIC   logLik deviance df.resid
 59969.0  59997.1 -29980.5  59961.0     8300

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2009 -0.7568 -0.0189  0.7329  2.8611

Random effects:
 Groups   Name       Variance Std.Dev.
 sch_id   family_inc  0.1464  0.3826
 Residual            73.1022  8.5500
Number of obs: 8304, groups:  sch_id, 854

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 4.014e+01  3.802e-01 8.303e+03  105.59   <2e-16
***
family_inc  1.095e+00  4.223e-02 5.303e+03   25.93   <2e-16
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1

Correlation of Fixed Effects:
           (Intr)
family_inc -0.901
```

In [27]:

```
coef.6 <- coef(model.6)$sch_id
names(coef.6) <- c("intercept", "slope_family_inc")
head(coef.6)
```

	intercept	slope_family_inc
81	40.14411	1.2523836
391	40.14411	0.9626563
421	40.14411	0.6038350
461	40.14411	0.9465261
615	40.14411	0.7981257
750	40.14411	1.9289007

- The coefficient data frame indicates that the intercept is fixed

Visualize the Fixed Intercept, Random Slopes Model

In [28]:

```
options(repr.plot.height=5, repr.plot.width=7)

ggplot() +
scale_x_continuous(name="family income", limits=c(0,40)) +
scale_y_continuous(name="math score", limits=c(25,100)) +
scale_linetype(name="family_inc_slope") +
geom_abline(data=coef.6,
            mapping=aes(slope=slope_family_inc,
                        intercept=intercept), color="lightblue") +
geom_abline(slope=mean(coef.6$slope_family_inc),
                        intercept=mean(coef.6$intercept), size=2, color="
```