

Take Home Final Solutions STAT 506 Spring 2014

1. Use these data on Math improvement scores. Use random effects for school, class (within school) as well as for the individual pupil. We want a model for mathgain based on the other variables (or a subset of these). These students were randomly selected.

- (a) Which potential predictor variables were measured on the schools? which on classrooms (or the teacher)? which on individuals?

Variables:

sex, minority status (yes = 1, 0 = no), SES (socio-economic status) are measured on each individual.

mathPrescore was measured in kindergarten on each individual

mathGain is the response of interest: improvement in the last year on each individual.

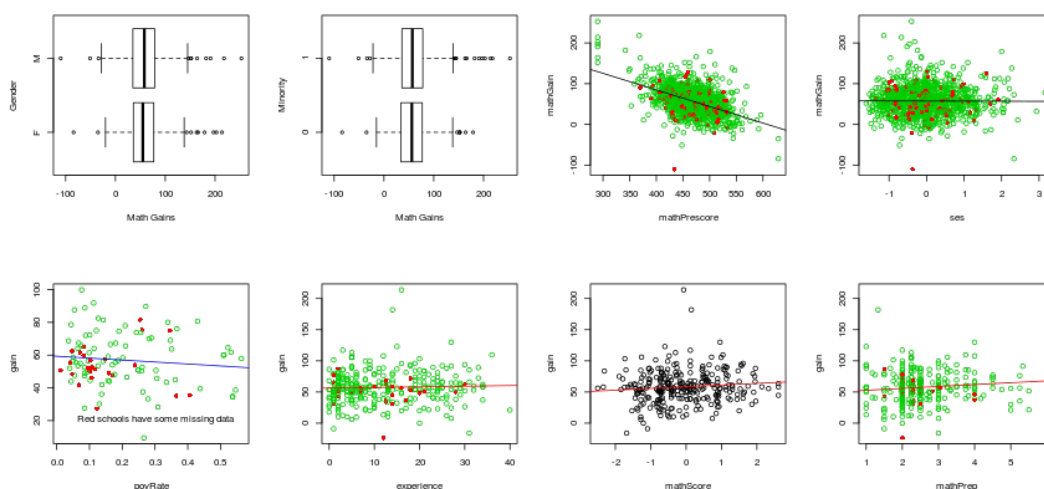
povertyRate is the proportion of kids qualifying for free lunch and is measured on the school.

teachrXper is number of years of experience (measured on the teacher/class).

teachrMath is a score of his/her mathematics knowledge (measured on the teacher/class).

teacherMathprep rates the math classes the teacher has taken (measured on the teacher/class).

- (b) Fit with appropriate mixed effects model. Explain what fixed and random terms you will use. Interpret results from your fit. Show diagnostic plots and discuss what you see.



I began by plotting each predictor against mathGain, and found very weak relationships with the exception of mathPrescore. Those with high prescores tended to get lower gain. That is probably just a limitation of using similar tests. A person who gets a high score at first has a harder time improving the grade because they are already close to the ceiling.

Next I worried about whether or not to use all the data (1190 kids in 312 classes within 107 schools), or to use the subset of complete data (1081 kids in 285 classes within 105 schools). The greater number of rows would be nice, but dropping about 100 kids, 2 schools, and 27 classes has the advantage of allowing us to use the teacher's math scores as a predictor, but it seems to be a variable with little predictive power (third plot in bottom row above). From the plots with green and red, the red points are omitted if we use only complete cases, and that seems OK to me in that they do not stand out as different. Another observation based on the plots: prescores seem to have a lump of observations at 300. I'd guess no one can score below that. Perhaps we should drop those students as coming from another population. I didn't do that because I don't have a way to consult with experts about that.

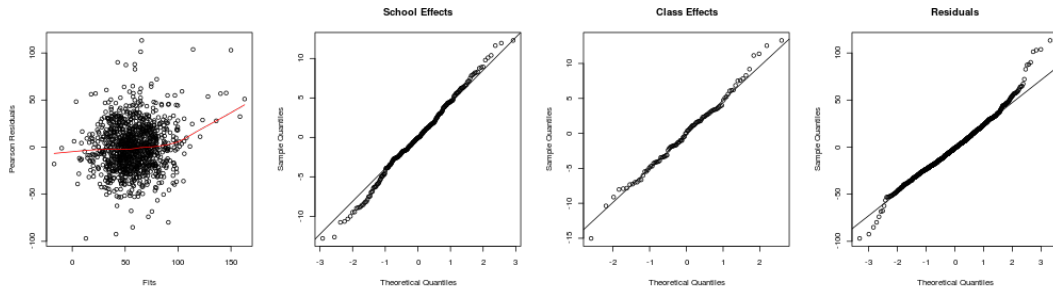
	Estimate	Std. Error	t value
(Intercept)	61.97	2.34	26.45
mathPrescore	-19.98	0.96	-20.90
sexM	1.33	1.72	0.77
minority	-7.51	2.46	-3.05
ses	4.00	0.96	4.17
teachXper	0.31	1.13	0.27
povertyRate	-1.12	1.38	-0.81
teachrMath	1.92	1.17	1.64
teachrMathprep	1.05	1.15	0.92

I first fit a model with all the (fixed) main effects. I included random effects for school and class within school, because those are part of the study design. As noted above, the major player is prescore with slope -0.41 ($SE = 0.044$). To facilitate comparisons, I then centered and rescaled the continuous predictors and refit the model. Minority has a coefficient of -7.51 ($SE = 2.46$) and SES has a coefficient of 3.95 ($SE = 0.95$), which are of the signs I would predict. Gelman says we should look for interactions with the stronger effects, so I refit adding two-way interactions for prescore, minority, and SES. Of these, I decided to keep the minority by SES and minority by prescore interactions which had t -ratios of -1.8 and -1.4 respectively.

```
## Computing profile confidence intervals ...
```

Diagnostics plots (below) show no problem with the assumption of normally distributed random effects. There are a few unusual kids scores, especially on the low end, but I'll not worry about them, just appeal to the central limit theorem (over 1000 kids).

	Estimate	Std. Error	t value
(Intercept)	60.67	2.38	25.51
mathPrescore	-18.19	1.64	-11.07
sexM	1.36	1.72	0.79
minority	-6.77	2.47	-2.75
ses	6.05	1.52	3.97
teachXper	0.31	1.13	0.27
povertyRate	-1.21	1.36	-0.89
teachrMath	1.79	1.17	1.53
teachrMathprep	1.04	1.15	0.91
minority:ses	-3.51	1.95	-1.80
mathPrescore:minority	-2.78	1.99	-1.40



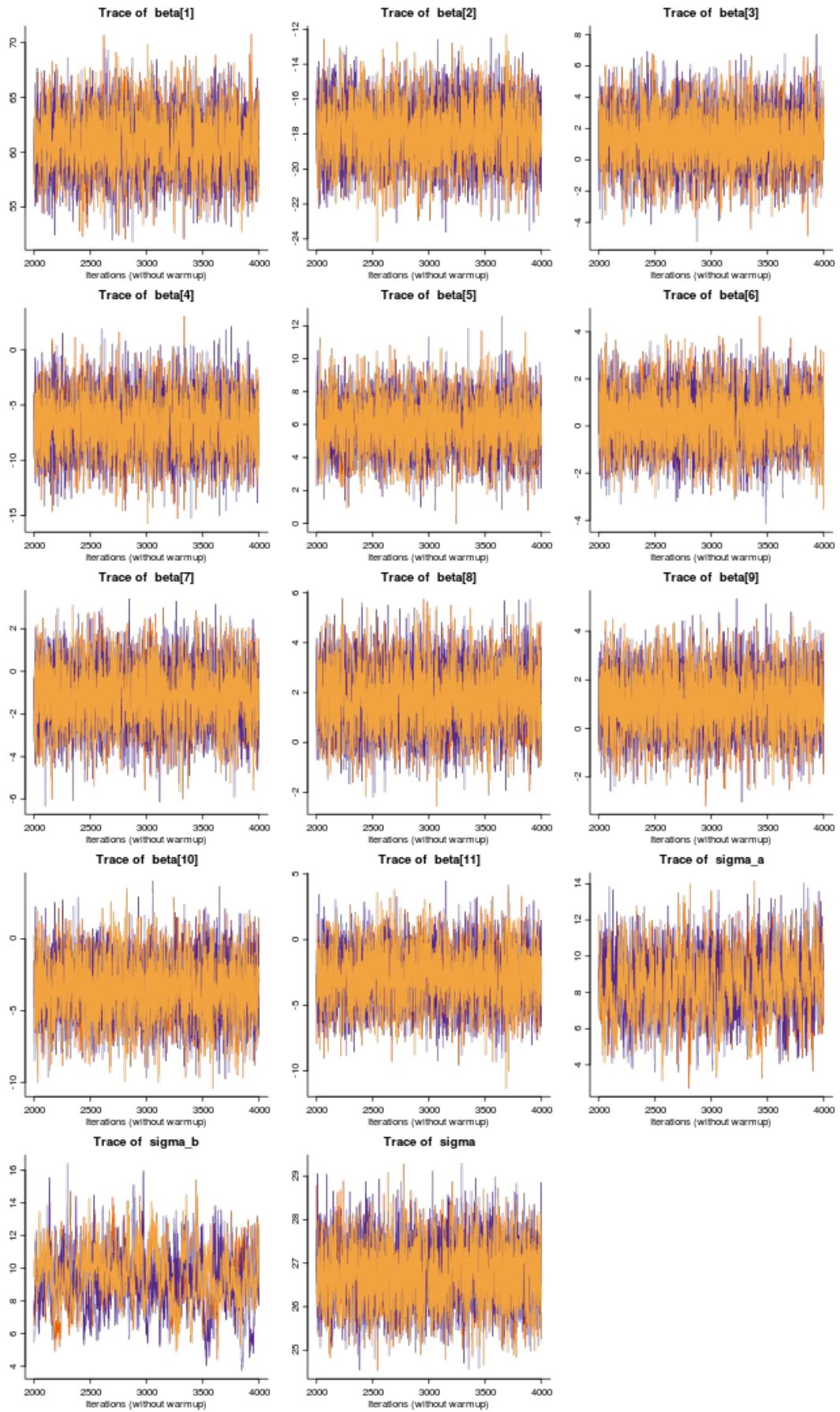
In addition to the underlying standard deviation for one child, estimated to lie in (25.4 ,28) with 95% confidence, we have school to school spread estimate of (4.7 ,11.2) and class within school spread estimated as (5.4 ,12.3).

Coefficient estimates: Each is conditional on all other terms in the model. The 95% CI for math prescore CI of (-21.4 , -14.9) which is saying that kids with higher prescore tend to have lower math gain. Minority has a CI of (-11.6 , -1.9) and SES has CI of (3.1 , 9). The other predictors: gender, poverty rate, teacher experience, teacher math score, teacher preparation, and minority by SES interaction all have 95% CI's that contain 0.

- (c) Fit the same model using MCMC software. Explain how interpretation changes for the Bayesian approach.

Using JAGS, I had a bit of a problem with convergence of the variance terms for class ($\hat{R} = 1.06$) and school ($\hat{R} = 1.23$) after 10 iterations of 1000 runs each on four chains. One school effect had an \hat{R} of 1.07, probably because of an unusual gain of 214. Other \hat{R} 's were less than 1.05.

To save time, I ran the models with rstan.



Compare CIs:

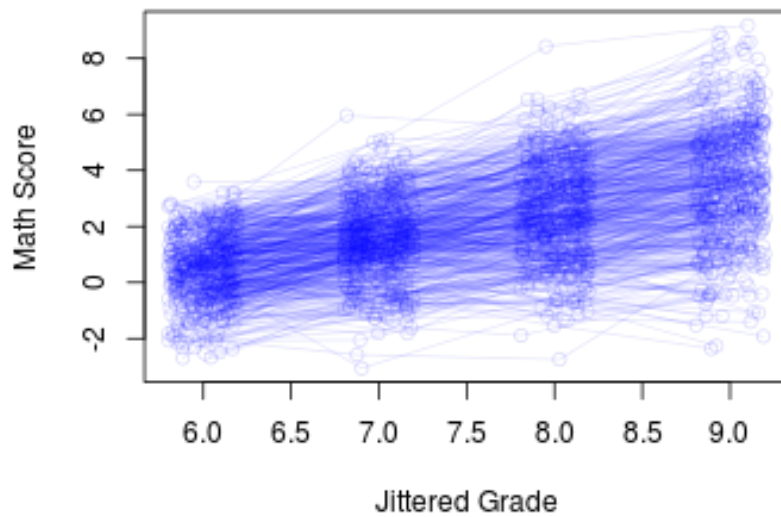
	lmer Lower	lmer upper	MCMC lower	MCMC upper
.sig01	5.4378	12.3190	5.9224	12.5916
.sig02	4.7395	11.2410	5.0275	11.5386
.sigma	25.3733	27.9668	25.4936	28.0938
(Intercept)	56.0103	65.2830	55.6856	66.0026
mathPrescore	-21.3714	-14.9258	-21.0822	-14.7453
sexM	-2.0016	4.7093	-1.9871	4.7788
minority	-11.5536	-1.8799	-11.4919	-1.7944
ses	3.0642	9.0164	3.0312	8.9598
teachXper	-1.8856	2.5173	-1.8907	2.5292
povertyRate	-3.8619	1.4331	-3.7900	1.4922
teachrMath	-0.5783	4.0829	-0.5617	4.1333
teachrMathprep	-1.1869	3.2704	-1.2169	3.3524
minority:ses	-7.3347	0.3084	-7.3209	0.2653
mathPrescore:minority	-6.7361	1.0844	-6.7034	1.0924

The lmer and stan estimates agree very closely. The lmer interval is usually given a frequentist interpretation: We are 95% confident that each (individually) contains its true parameter. (Confidence is the long run probability of inclusion of the fixed parameter in a random interval when sampling repeatedly.) The MCMC interval is a 95% posterior credible interval, meaning that there is a 0.95 posterior probability that the "random" parameter falls in the fixed interval given the prior and the data.

2. Another study followed 500 randomly selected kids through four grades to track their math achievement scores.

Do preliminary exploration in lme or lmer, but I want your report on these data to be based solely on a JAGs fit. That includes diagnostics to evaluate the assumptions of the model.

After explaining your model, derive the correlation between grade i and grade j scores for the same individual from the MCMC output (for each i, j pair). Compare to lme or lmer estimates.



I start again with a plot which shows a generally increasing trend with lots of individual variation in these 500 students.

In lmer we can fit a trend in the fixed effects and can examine the need for random slopes and intercepts for each kid.

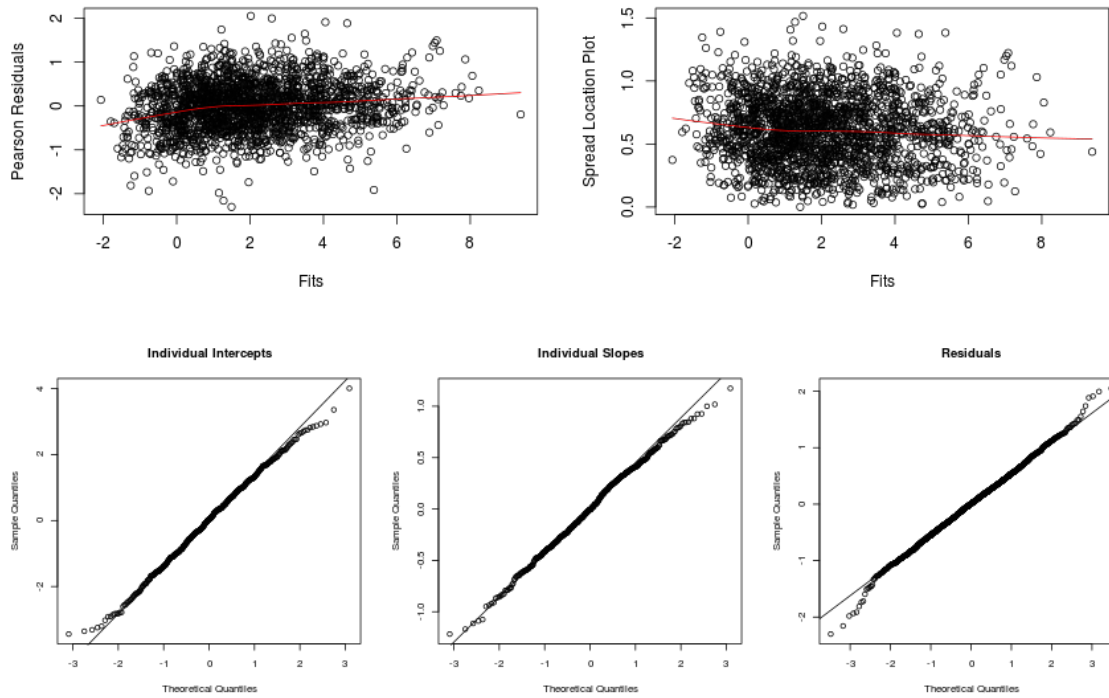
```
## refitting model(s) with ML (instead of REML)
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
mImp1	4	6510.78	6533.18	-3251.39	6502.78			
mImp2	6	6045.39	6079.00	-3016.70	6033.39	469.38	2	0.0000
mImp3	7	6047.31	6086.52	-3016.65	6033.31	0.08	1	0.7707

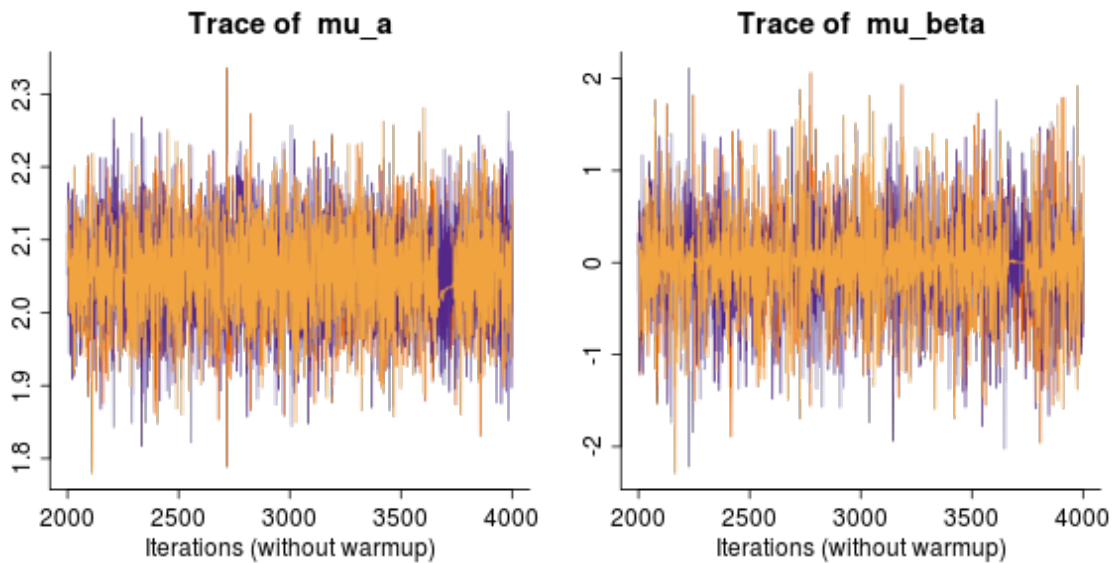
```
## refitting model(s) with ML (instead of REML)
```

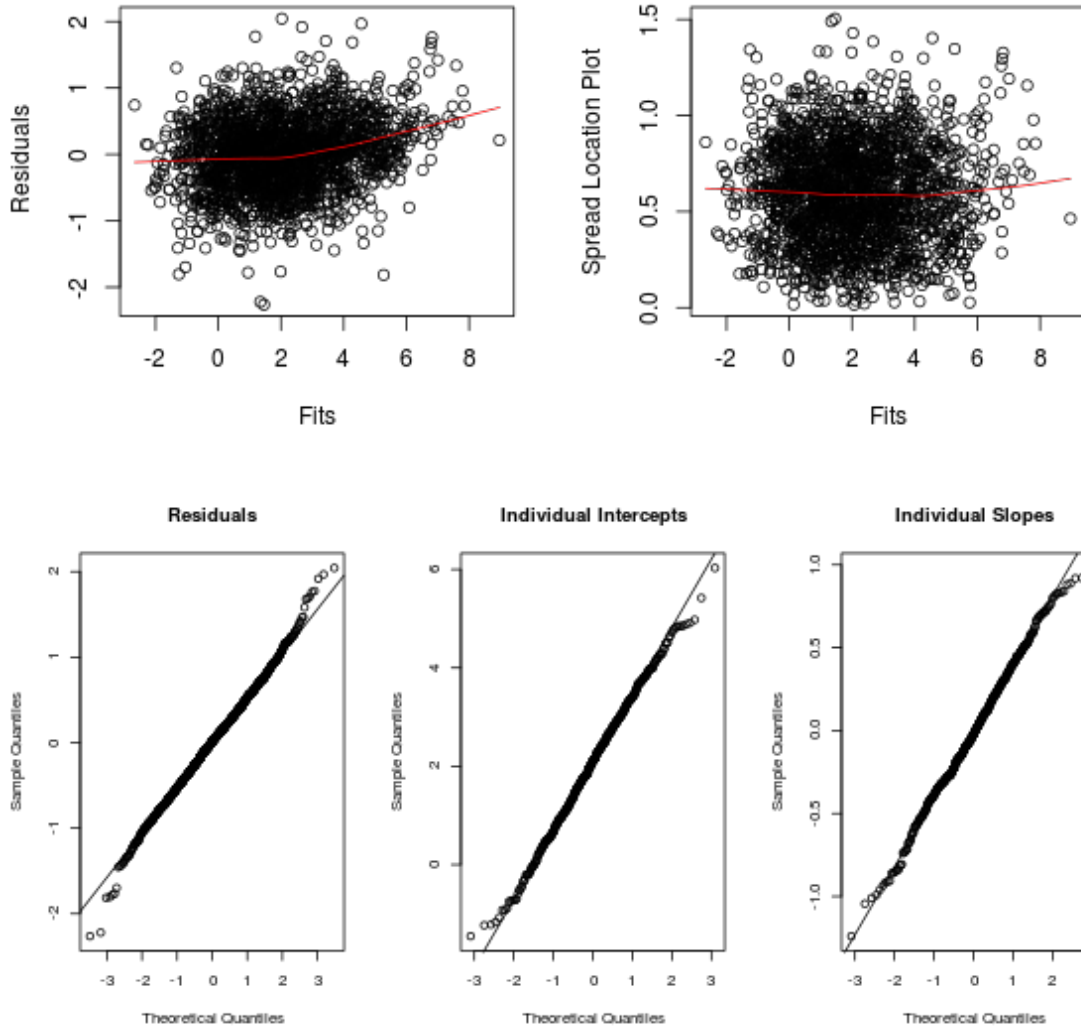
	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
mImp2	6	6045.39	6079.00	-3016.70	6033.39			
mImp4	8	6048.97	6093.78	-3016.49	6032.97	0.42	2	0.8090

The first comparison of models shows that we clearly need random intercept and slope. The addition of a quadratic effect does not improve the model. Several of you treated grade as a factor, but the last AIC comparison/anova shows that is not necessary. The improvement from one grade to the next is just about one for each grade level.



I fit the same model with JAGs (took way too long) and with Stan (much quicker) and ran 4000 draws from 4 chains to obtain convergence. The Gelman \hat{R} diagnostic values were all less than 1.02. To extract what we need from the MCMC runs, I will grab means from the stan summary.





The posterior mean residuals seem close to normally distributed, as are mean random effects.

To compute the inter-student correlation of two measurements taken in grade t_i and grade t_j on the k^{th} kid, we need covariances and variances.

$$\text{cov}(a_k + b_k t_i, a_k + b_k t_j) = \text{var}(a_k) + t_i t_j \text{var}(b_k) + \text{cov}(a_k, b_k)(t_i + t_j) = \sigma_a^2 + t_i t_j \sigma_b^2 + (t_i + t_j) \rho \sigma_a \sigma_b$$

From `lmer` the estimates are: $\hat{\sigma}_a^2 = 1.8958$; $\hat{\sigma}_b^2 = 0.2275$; and $\hat{\rho} = 0.72$. With JAGs I get posterior medians of 1.38; 0.477; and 0.459.

Using posterior means as estimates, the covariance will be

$$1.38 + t_i t_j 0.477 + (t_i + t_j) 0.459(1.38)(0.477)$$

Then correlation is estimated as

$$\frac{1.38 + t_i t_j 0.477 + (t_i + t_j) 0.459(1.38)(0.477)}{\sqrt{1.38 + t_i^2 0.477 + 2t_i 0.459(1.38)(0.477)} \sqrt{1.38 + t_j^2 0.477 + 2t_j 0.459(1.38)(0.477)}}$$

A better way to compute the overall value for this would be to compute it for each MCMC iteration, then summarize the distribution, but we can't do that without picking particular i and j .

R Code

```
math1 <- read.csv("mathComparison.csv")
## School measures
mathSchool <- data.frame(school = 1:107, gain = tapply(math1$mathGain, math1$schoolid,
  mean), povRate = tapply(math1$povertyRate, math1$schoolid, mean), miss = sapply(1:107,
  function(ndx) any(is.na(math1$teachrMath[math1$schoolid == ndx]))))

## class measures
mathClass <- data.frame(class = 1:312, gain = tapply(math1$mathGain, math1$classid, mean),
  experience = tapply(math1$teachXper, math1$classid, mean), mathScore = tapply(math1$teachrMath,
  math1$classid, mean), mathPrep = tapply(math1$teachrMathprep, math1$classid,
  mean), school = sapply(1:312, function(ndx) median(math1$schoolid[math1$classid ==
  ndx])), rowMin = sapply(1:312, function(ndx) min(which(math1$classid == ndx))),
  rowMax = sapply(1:312, function(ndx) max(which(math1$classid == ndx))))

par(mfrow = c(2, 4))
boxplot(mathGain ~ sex, data = math1, horizontal = TRUE, xlab = "Math Gains", ylab = "Gender")
boxplot(mathGain ~ minority, data = math1, horizontal = TRUE, xlab = "Math Gains", ylab = "Minority")
plot(mathGain ~ mathPrescore, data = math1, col = 3 - is.na(teachrMath), pch = 1 + 15 *
  is.na(teachrMath))
abline(lm(mathGain ~ mathPrescore, data = math1))
plot(mathGain ~ ses, data = math1, col = 3 - is.na(teachrMath), pch = 1 + 15 * is.na(teachrMath))
abline(lm(mathGain ~ ses, data = math1))

plot(gain ~ povRate, data = mathSchool, col = 3 - miss, pch = 1 + 15 * miss)
abline(lm(gain ~ povRate, data = mathSchool), col = 4)
text(0.3, 20, "Red schools have some missing data")

plot(gain ~ experience, data = mathClass, col = 3 - is.na(mathScore), pch = 1 + 15 *
  is.na(mathScore))
abline(lm(gain ~ experience, data = mathClass), col = 2)
plot(gain ~ mathScore, data = mathClass)
abline(lm(gain ~ mathScore, data = mathClass), col = 2)
plot(gain ~ mathPrep, data = mathClass, col = 3 - is.na(mathScore), pch = 1 + 15 * is.na(mathScore))
abline(lm(gain ~ mathPrep, data = mathClass), col = 2)

require(lme4)
math2 <- subset(math1, !is.na(teachrMath))
## drop schools 48 & 58, drop classes: 7 12 15 32 51 53 107 109 111 133 135 143 153
## 157 160 164 179 182 194 226 251 266 274 276 278 307 309
math2$schoolid <- as.numeric(as.factor(math2$schoolid))
math2$classid <- as.numeric(factor(math2$classid, levels = unique(math2$classid)))
mathlmerFit1 <- lmer(mathGain ~ mathPrescore + sex + minority + ses + teachXper + povertyRate +
  teachrMath + teachrMathprep + (1 | schoolid/classid), data = math2)
math2[, c(3, 5:9)] <- scale(math2[, c(3, 5:9)])
mathlmerFit1 <- lmer(mathGain ~ mathPrescore + sex + minority + ses + teachXper + povertyRate +
  teachrMath + teachrMathprep + (1 | schoolid/classid), data = math2)
```

```

xtable(summary(mathlmerFit1)$coef)
## school measures
mathSchool <- data.frame(school = 1:105, gain = tapply(math2$mathGain, math2$schoolid,
  mean), povRate = tapply(math2$povertyRate, math2$schoolid, mean), minClass = sapply(1:105,
  function(ndx) min(math2$classid[math2$schoolid == ndx])), maxClass = sapply(1:105,
  function(ndx) max(math2$classid[math2$schoolid == ndx])))

## class measures
classIDs <- 1:285
mathClass <- data.frame(class = classIDs, gain = tapply(math2$mathGain, math2$classid,
  mean), experience = tapply(math2$teachXper, math2$classid, mean), mathScore = tapply(math2$teachrMath,
  math2$classid, mean), mathPrep = tapply(math2$teachrMathprep, math2$classid, mean),
  school = sapply(classIDs, function(ndx) median(math2$schoolid[math2$classid == ndx])),
  rowMin = sapply(classIDs, function(ndx) min(which(math2$classid == ndx))), rowMax = sapply(classIDs,
  function(ndx) max(which(math2$classid == ndx))))

mathlmerFit2 <- lmer(mathGain ~ mathPrescore + sex + minority * ses + mathPrescore:minority +
  teachXper + povertyRate + teachrMath + teachrMathprep + (1 | schoolid/classid), data = math2)
xtable(summary(mathlmerFit2)$coef)
ci2 <- confint(mathlmerFit2)

```

```

par(mfrow = c(1, 4))
plot(fitted(mathlmerFit2), resid(mathlmerFit2, type = "pearson"), xlab = "Fits", ylab = "Pearson Residuals")
lines(lowess(fitted(mathlmerFit2), resid(mathlmerFit2, type = "pearson")), col = 2)
qqnorm(ranef(mathlmerFit2)[[1]][, 1], main = "School Effects")
qqline(ranef(mathlmerFit2)[[1]][, 1])
qqnorm(ranef(mathlmerFit2)[[2]][, 1], main = "Class Effects")
qqline(ranef(mathlmerFit2)[[2]][, 1])
qqnorm(resid(mathlmerFit2), main = "Residuals")
qqline(resid(mathlmerFit2))

```

```

cat('model{
for (i in 1:n){
  y[i] ~ dnorm (beta[1] + beta[2] * mathPrescore[i] + beta[3] * male[i] +
    beta[4] * minority[i] + beta[5] * ses[i] +
    beta[6] * teachXper[i] + beta[7] * povertyRate[i] +
    beta[8] * teachMath[i] + beta[9] * teachMathprep[i] +
    beta[10] * minority[i] * ses[i] +
    beta[11] * mathPrescore[i] * minority[i] +
    b2[class[i]] + b1[school[i]], invsigsq)
}
for (k in 1:K){
  b2.raw[k] ~ dnorm(0, invsigsq2)
  b2[k] <- b2.raw[k] - mean(b2.raw)
}
for (j in 1:J){
  b1.raw[j] ~ dnorm(0, invsigsq1)
  b1[j] <- b1.raw[j] - mean(b1.raw)
}
for(e1 in 1:11){

```

```

    beta[e1] ~ dnorm(0, .001)
  }
  invsigsq1 ~ dgamma(0.001, 0.001)
  sigma1 <- pow(invsigsq1, -.5)
  invsigsq2 ~ dgamma(0.001, 0.001)
  sigma2 <- pow(invsigsq2, -.5)
  invsigsq ~ dgamma(0.001, 0.001)
  sigma <- pow(invsigsq, -.5)
}',
  file = "math1.jags")

```

```

math1.jagsdata <- with(math2, list(n = nrow(math2), J = length(unique(schoolid)), K = length(unique(classid)),
  y = mathGain, school = schoolid, class = classid, mathPrescore = mathPrescore, male = (sex == "M") + 0, minority = minority, ses = ses, teachMath = teachrMath, teachMathprep = teachrMathprep,
  teachXper = teachXper, povertyRate = povertyRate))
math1.JAGs <- jags(model.file = "math1.jags", data = math1.jagsdata, parameters.to.save = c("b1", "b2", "beta", "sigma", "sigma1", "sigma2"), n.chains = 4, n.iter = 6000) ##
math1.JAGs <- autojags(math1.JAGs, n.iter = 4000, n.chains = 4, n.thin = 16, Rhat = 1.05, n.update = 5)
math1.JAGs

```

1 in rstan:

```

cat('data {
  int<lower=0> N;
  int<lower=1,upper=1081> classid[N];
  int<lower=1,upper=1081> school[N];
  vector[N] mathPrescore;
  vector[N] male;
  vector[N] minority;
  vector[N] ses;
  vector[N] povertyRate;
  vector[N] teachXper;
  vector[N] teachMath;
  vector[N] teachMathprep;
  vector[N] y;
}
transformed data {
  vector[N] eth_Prescore;
  eth_ses <- minority * ses;
  eth_Prescore <- mathPrescore * minority;
}
parameters {
  vector[105] a; ## school effects
  vector[285] b; ## class effects
  vector[11] beta;
  real mu_a;
  real mu_b;
  real<lower=0,upper=100> sigma_a;
  real<lower=0,upper=100> sigma_b;
  real<lower=0,upper=100> sigma;
}
transformed parameters {
  vector[N] y_hat;

```

```

for (i in 1:N)
  y_hat[i] <- beta[1] + beta[2] * mathPrescore[i] + beta[3] * male[i] + beta[4] * minority[i] + beta[5]
}

model{
  y ~ normal (y_hat, sigma);
  mu_a ~ normal(0, 1);
  a ~ normal (mu_a, sigma_a);
  mu_b ~ normal(0, 1);
  b ~ normal (0.1 * mu_b, sigma_b);
}', file = "math1.stan")

```

Weird knitr problem: if I include the whole model, the block of code turns blank. Here's the whole model which should be above:

```

y_hat[i] <- beta[1] + beta[2] * mathPrescore[i] + beta[3] * male[i] +
  beta[4] * minority[i] + beta[5] * ses[i] +
  beta[6] * teachXper[i] + beta[7] * povertyRate[i] +
  beta[8] * teachMath[i] + beta[9] * teachMathprep[i] +
  beta[10] * eth_ses[i] +
  beta[11] * eth_Prescore[i] +
  a[school[i]] + b[classid[i]];

```

```

math1.stanData <- with(math2, list(N = nrow(math2), y = mathGain, school = schoolid,
  classid = classid, mathPrescore = mathPrescore, male = (sex == "M") + 0, minority = minority,
  ses = ses, teachMath = teachrMath, teachMathprep = teachrMathprep, teachXper = teachXper,
  povertyRate = povertyRate))
math1.stan <- stan(file = "math1.stan", data = math1.stanData, par = c("a", "b", "beta",
  "sigma", "sigma_a", "sigma_b"), chains = 4, iter = 4000) ##

```

```

rstan::traceplot(math1.stan, pars = c("beta", "sigma_a", "sigma_b", "sigma"), inc_warmup = FALSE,
  ncol = 3, nrow = 5)
ttemp1 <- summary(math1.stan)

```

```

stanCIs <- ttemp1$summary[c(404:402, 391:401), c(4, 8)]
bothCIs <- cbind(ci2, stanCIs)
colnames(bothCIs) <- c("lmer Lower", "lmer upper", "MCMC lower", "MCMC upper")
xtable(bothCIs, digits = 4)

```

```

mathImp <- read.csv("mathImprovement.csv")
mathImp <- with(mathImp, data.frame(id = rep(id, 4), grade = rep(6:9, each = 500) - 7.5,
  score = unlist(mathImp[, 2:5])))
jitgrade <- jitter(mathImp$grade)
plot(jitgrade + 7.5, mathImp$score, col = rgb(0, 0, 1, 0.15), xlab = "Jittered Grade",
  ylab = "Math Score")
for (i in 1:500) lines(y = mathImp$score[mathImp$id == i], x = jitgrade[mathImp$id ==
  i] + 7.5, col = rgb(0, 0, 1, 0.1))

```

```

mImp1 <- lmer(score ~ grade + (1 | id), data = mathImp)
mImp2 <- lmer(score ~ grade + (1 + grade | id), data = mathImp)
mImp3 <- lmer(score ~ grade + I(grade^2) + (1 + grade | id), data = mathImp)
xtable(anova(mImp1, mImp2, mImp3))
mImp4 <- lmer(score ~ factor(grade) + (1 + grade | id), data = mathImp)
xtable(anova(mImp2, mImp4))

```

```

par(mfrow = c(1, 2))
plot(fitted(mImp2), resid(mImp2, type = "pearson"), xlab = "Fits", ylab = "Pearson Residuals")
lines(lowess(fitted(mImp2), resid(mImp2, type = "pearson")), col = 2)
plot(fitted(mImp2), sqrt(abs(resid(mImp2, type = "pearson"))), xlab = "Fits", ylab = "Spread Location Plot")
lines(lowess(fitted(mImp2), sqrt(abs(resid(mImp2, type = "pearson")))), col = 2)
par(mfrow = c(1, 3))
qqnorm(ranef(mImp2)$id[, 1], main = "Individual Intercepts")
qqline(ranef(mImp2)$id[, 1])
qqnorm(ranef(mImp2)$id[, 2], main = "Individual Slopes")
qqline(ranef(mImp2)$id[, 2])
qqnorm(resid(mImp2), main = "Residuals")
qqline(resid(mImp2))

```

```

cat('var B[2,J],B.hat[2,J];
model{
  for (i in 1:n){
    y[i] ~ dnorm (beta[1] + beta[2] * grade[i] +
                  a[id[i]] + b[id[i]] * grade[i], invsigsq)
  }
  for (j in 1:J){
    a[j] <- B[1,j] - mean(B[1,1:J ])
    b[j] <- B[2,j] - mean(B[2, 1:J ])
    B[1:2, j] ~ dnmnorm (B.hat[1:2,j], invSigB)
    B.hat[1, j] <- 0
    B.hat[2, j] <- 0
  }
  for(el in 1:2){
    beta[el] ~ dnorm(0, .001)
  }
  invSigB ~ dwish( W , 3)
  SigmaB <- inverse(invSigB)
  sigma.a <- sqrt(SigmaB[1, 1])
  sigma.b <- sqrt(SigmaB[2, 2])
  rho <- SigmaB[1,2] / sqrt(SigmaB[1, 1] * SigmaB[2, 2])
  invsigsq ~ dgamma(0.001, 0.001)
  sigma <- pow(invsigsq, -.5)
}', file = "math2.jags")

```

```

math2.jagsdata <- with(mathImp, list(n = nrow(mathImp), J = length(unique(id)), y = score,
  id = id, grade = grade))
math2.JAGs <- jags(model.file = "math2.jags", data = math2.jagsdata, parameters.to.save = c("a",
  "b", "beta", "sigma", "sigma.a", "sigma.b", "rho"), n.chains = 4, n.iter = 4000) ##
math2.JAGs <- autojags(math2.JAGs, n.iter = 4000, Rhat = 1.05, n.update = 10)
math2.JAGs

```

2 in rstan:

```
cat('data {  
  int<lower=0> N;  
  int<lower=1,upper=500> id[N];  
  vector[N] grade;  
  vector[N] y;  
}  
parameters {  
  vector[500] a;  
  vector[500] b;  
  real beta;  
  real mu_a;  
  real mu_b;  
  real mu_beta;  
  real<lower=0,upper=100> sigma_a;  
  real<lower=0,upper=100> sigma_b;  
  real<lower=0,upper=100> sigma_beta;  
  real<lower=0,upper=100> sigma_y;  
}  
transformed parameters {  
  vector[N] y_hat;  
  for (i in 1:N)  
    y_hat[i] <- a[id[i]] + grade[i] * b[id[i]] + beta * grade[i];  
}  
model {  
  mu_beta ~ normal(0, 1);  
  beta ~ normal(100 * mu_beta, sigma_beta);  
  
  mu_a ~ normal(0, 1);  
  a ~ normal(mu_a, sigma_a);  
  
  mu_b ~ normal(0, 1);  
  b ~ normal(0.1 * mu_b, sigma_b);  
  
  y ~ normal(y_hat, sigma_y);  
}', file = "mathImp.stan")
```

```
mathImp.stanData <- with(mathImp, list(N = nrow(mathImp), y = score, id = id, grade = grade))  
mathImp.stanfit <- stan(file = "mathImp.stan", data = mathImp.stanData, pars = c("a",  
  "b", "beta", "mu_a", "mu_b", "mu_beta", "sigma_y", "sigma_a", "sigma_b", "sigma_beta",  
  "y_hat"), iter = 4000) ##  
ttemp <- summary(mathImp.stanfit)  
xtable(ttemp$summary[1001:1008, ])  
traceplot(mathImp.stanfit, pars = c("mu_a", "mu_b", "mu_beta", "sigma_a", "sigma_b",  
  "sigma_beta"), inc_warmup = FALSE, ncol = 2, nrow = 3)  
a <- sapply(mathImp.stanfit@sim$samples[[1]][1:500], function(x) unlist(x))  
a <- rbind(a, sapply(mathImp.stanfit@sim$samples[[2]][1:500], function(x) unlist(x)))  
a <- rbind(a, sapply(mathImp.stanfit@sim$samples[[3]][1:500], function(x) unlist(x)))  
a <- rbind(a, sapply(mathImp.stanfit@sim$samples[[4]][1:500], function(x) unlist(x)))  
b <- sapply(mathImp.stanfit@sim$samples[[1]][500 + 1:500], function(x) unlist(x))  
b <- rbind(b, sapply(mathImp.stanfit@sim$samples[[1]][500 + 1:500], function(x) unlist(x)))  
b <- rbind(b, sapply(mathImp.stanfit@sim$samples[[1]][500 + 1:500], function(x) unlist(x)))  
b <- rbind(b, sapply(mathImp.stanfit@sim$samples[[1]][500 + 1:500], function(x) unlist(x)))  
cor(as.numeric(a), as.numeric(b))
```

With stan, convergence was fast, the effective sample sizes and \hat{R} s are all good (< 1.02), but variation in `sigma_beta` is huge: 95% posterior credible interval of roughly (2.0, 100).