homework 07

Wendy Liang

October 26, 2020

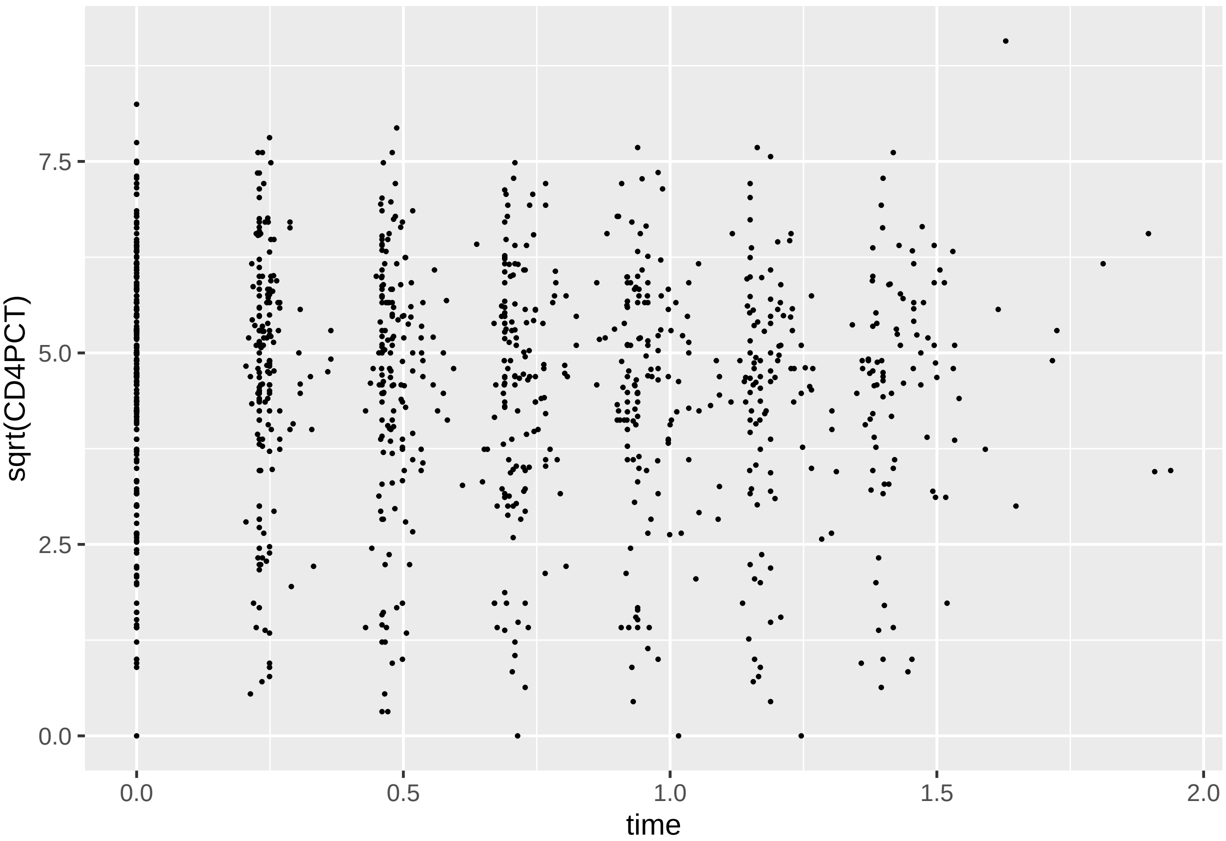
Data analysis

CD4 percentages for HIV infected kids

The folder `cd4` has CD4 percentages for a set of young children with HIV who were measured several times over a period of two years. The dataset also includes the ages of the children at each measurement.

1. Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

```
ggplot(hiv.data, aes(y = sqrt(CD4PCT), x = time)) +
  geom_point(size=0.3)
```



2. Each child's data has a time course that can be summarized by a linear fit. Estimate these lines and plot them for all the children.

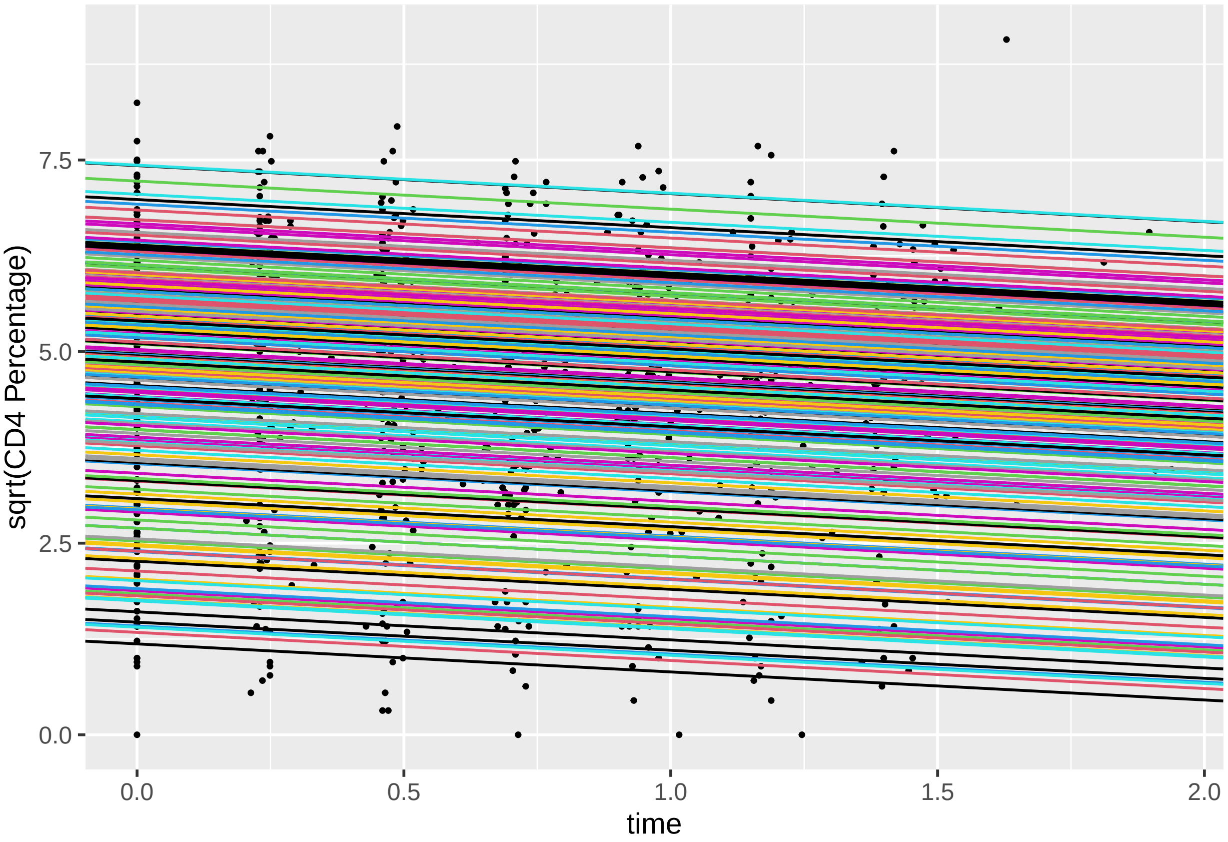
```
#We add "-1" to the regression formula to remove the constant term, so that all 85 counties are include
#lf1 <- lm (y ~ time+factor(newpid)-1, data = hiv.data)
```

```
# display(m1)
#coef_lf1 <- data.frame(coef(lf1))
#ggplot(hiv.data,aes(x=time, y=y,color=factor(newpid))) +
#  geom_smooth(method="lm",se=FALSE,size=0.5) +
#  theme(legend.position="none")

lf1 <- lmer(y ~ 1 + time + (1|newpid),data = hiv.data)
display(lf1)
```

```
## lmer(formula = y ~ 1 + time + (1 | newpid), data = hiv.data)
##             coef.est coef.se
## (Intercept)  4.76     0.10
## time        -0.37     0.05
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.40
##  Residual             0.77
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3148.8, DIC = 3126.9
## deviance = 3133.9
```

```
lf1_coef <- coef(lf1)
lf1_coef <- data.frame(lf1_coef$newpid)
colnames(lf1_coef)[1] <- c("intercept")
## extract the data frame 0 column!
A=as.numeric(rownames(lf1_coef))
lf1_coef$newpid <- A
ggplot(data=hiv.data) +
  geom_point(aes(x=time, y=y),size=0.5) +
  geom_abline(intercept = lf1_coef$intercept,
              slope=lf1_coef$time, col=lf1_coef$newpid)+
  labs(y="sqrt(CD4 Percentage)")
```



3. Set up a model for the children's slopes and intercepts as a function of the treatment and age at baseline. Estimate this model using the two-step procedure–first estimate the intercept and slope separately for each child, then fit the between-child models using the point estimates from the first step.

```
child<- matrix(0,nrow=254,ncol = 3)
colnames(child) <- c("newpid","intercept","slope")
for (i in unique(hiv.data$newpid)){
  child_lm <- lm(y ~ time, hiv.data[newpid == i,c("y","time")])
  child[i,1] <- i
  child[i,2] <- coef(child_lm)[1]
  child[i,3] <- coef(child_lm)[2]
}

hiv.data.use <- hiv.data[,list(age.baseline=unique(age.baseline),treatment=unique(treatment)), by=newpi
#Merge two data frames by common columns or row names, or do other versions of database join operations
hiv.data.use <- merge(child,hiv.data.use,by="newpid")

lm(intercept~ age.baseline+factor(treatment),data = hiv.data.use)
```

```
##
## Call:
## lm(formula = intercept ~ age.baseline + factor(treatment), data = hiv.data.use)
##
## Coefficients:
##       (Intercept)       age.baseline  factor(treatment)2
##            5.1179            -0.1210              0.1236
```

```r
lm(slope~ age.baseline+factor(treatment),data=hiv.data.use)
```

```
##
## Call:
## lm(formula = slope ~ age.baseline + factor(treatment), data = hiv.data.use)
##
## Coefficients:
##        (Intercept)        age.baseline  factor(treatment)2
##           -0.26568            -0.04223            -0.13926
```

4. Write a model predicting CD4 percentage as a function of time with varying intercepts across children. Fit using `lmer()` and interpret the coefficient for time.

```r
# for varying intercept
lf2 <- lmer(y ~ time + (1|newpid),data = hiv.data)
display(lf2)
```

```
## lmer(formula = y ~ time + (1 | newpid), data = hiv.data)
##             coef.est coef.se
## (Intercept)  4.76     0.10
## time        -0.37     0.05
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.40
##  Residual             0.77
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3148.8, DIC = 3126.9
## deviance = 3133.9
```

5. Extend the model in (4) to include child-level predictors (that is, group-level predictors) for treatment and age at baseline. Fit using `lmer()` and interpret the coefficients on time, treatment, and age at baseline.

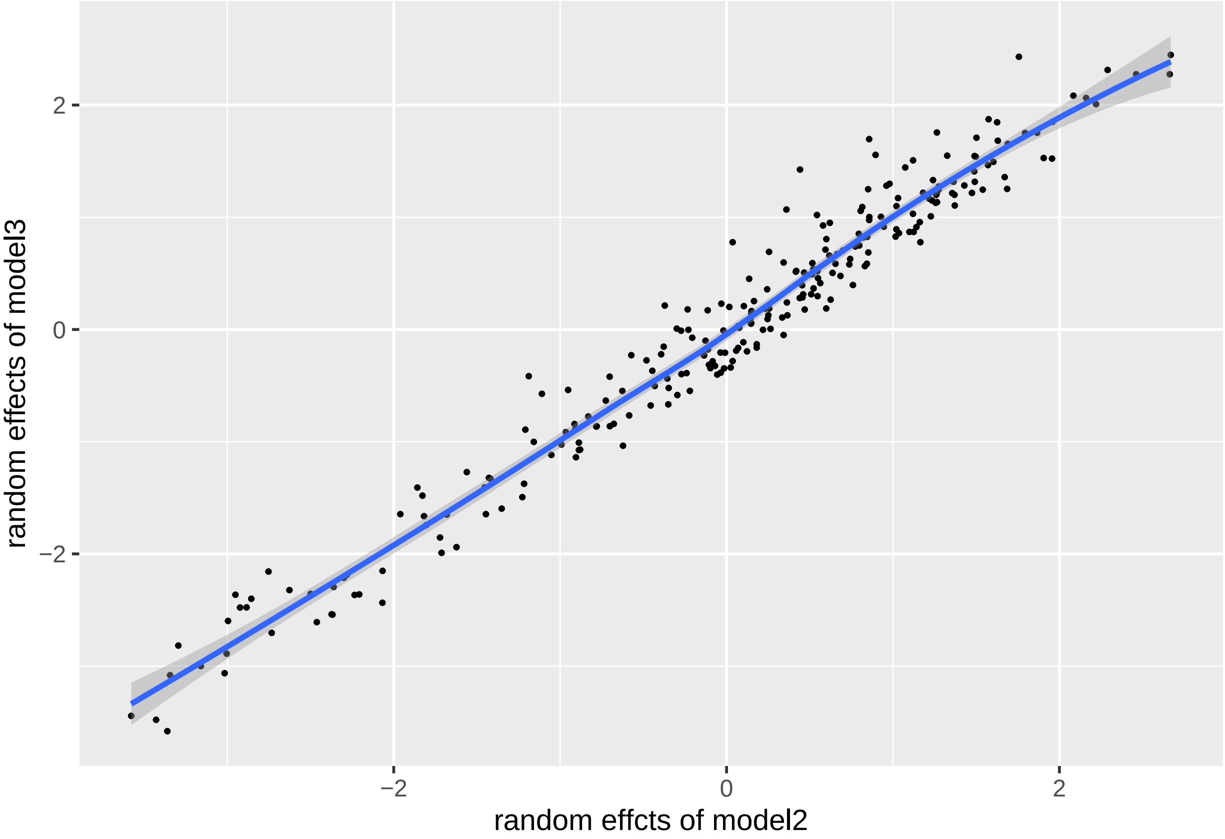
```r
lf3 <- lmer(y ~ time + treatment + age.baseline + (1|newpid),data = hiv.data)
display(lf3)
```

```
## lmer(formula = y ~ time + treatment + age.baseline + (1 | newpid),
##     data = hiv.data)
##              coef.est coef.se
## (Intercept)   4.91     0.32
## time         -0.36     0.05
## treatment     0.18     0.18
## age.baseline -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.37
##  Residual             0.77
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3149.2, DIC = 3110.9
## deviance = 3124.1
```

6. Investigate the change in partial pooling from (4) to (5) both graphically and numerically.

```r
change <- as.data.frame(cbind(unlist(ranef(lf2)),unlist(ranef(lf3))))
colnames(change) <- c("lf2","lf3")
ggplot(change,aes(x=lf2,y=lf3))+geom_point(size=0.5)+geom_smooth()+
  xlab("random effcts of model2")+
  ylab("random effects of model3")
```

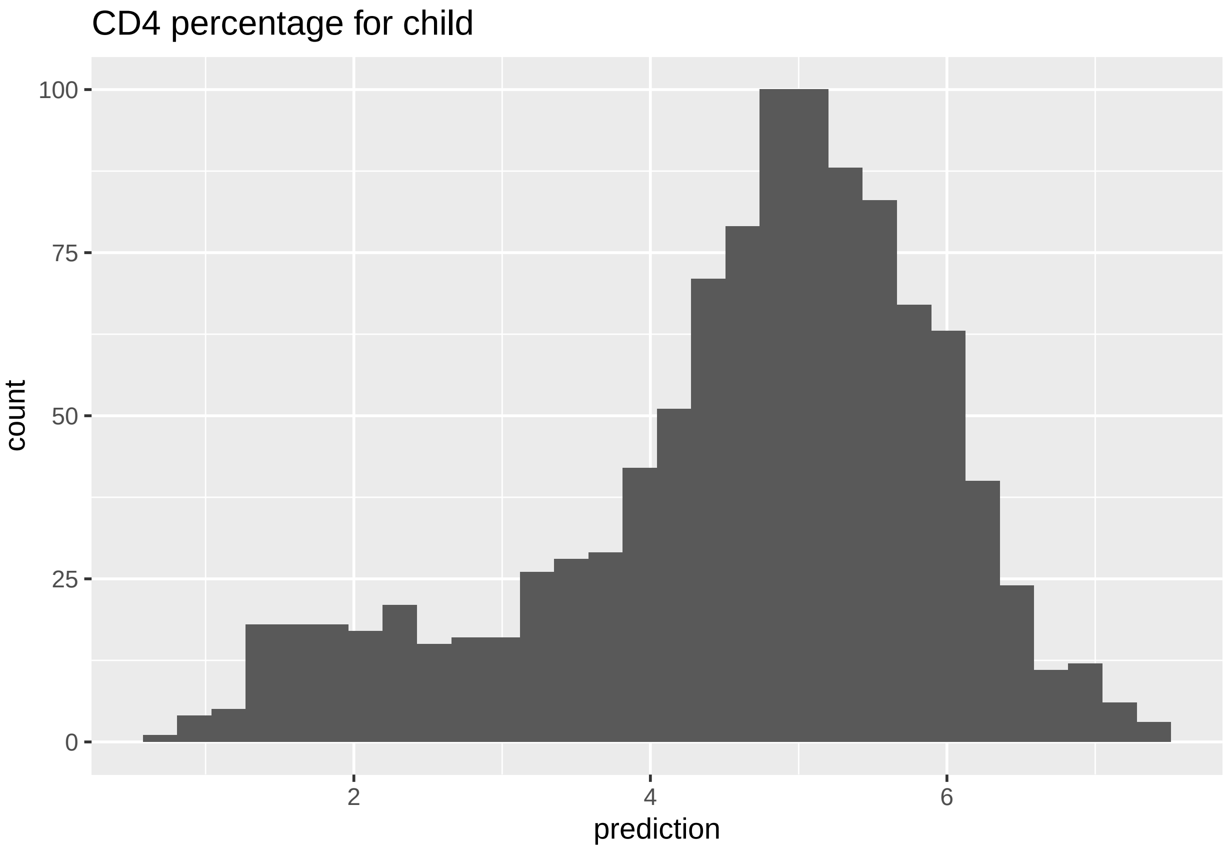
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



7. Use the model fit from (5) to generate simulation of predicted CD4 percentages for each child in the dataset at a hypothetical next time point.

```r
hiv.pred <- subset(hiv.data, !is.na(treatment) & !is.na(baseage))
hiv.pred=hiv.pred[,c(2,7,12,14)]
hiv.pred.result <- predict(lf3,newdata=hiv.pred)
pred.compare <- cbind(hiv.pred.result,hiv.pred)
colnames(pred.compare)[1] <- c("prediction")
ggplot(pred.compare,aes(x=prediction))+
  geom_histogram()+
  ggtitle("CD4 percentage for child")
```

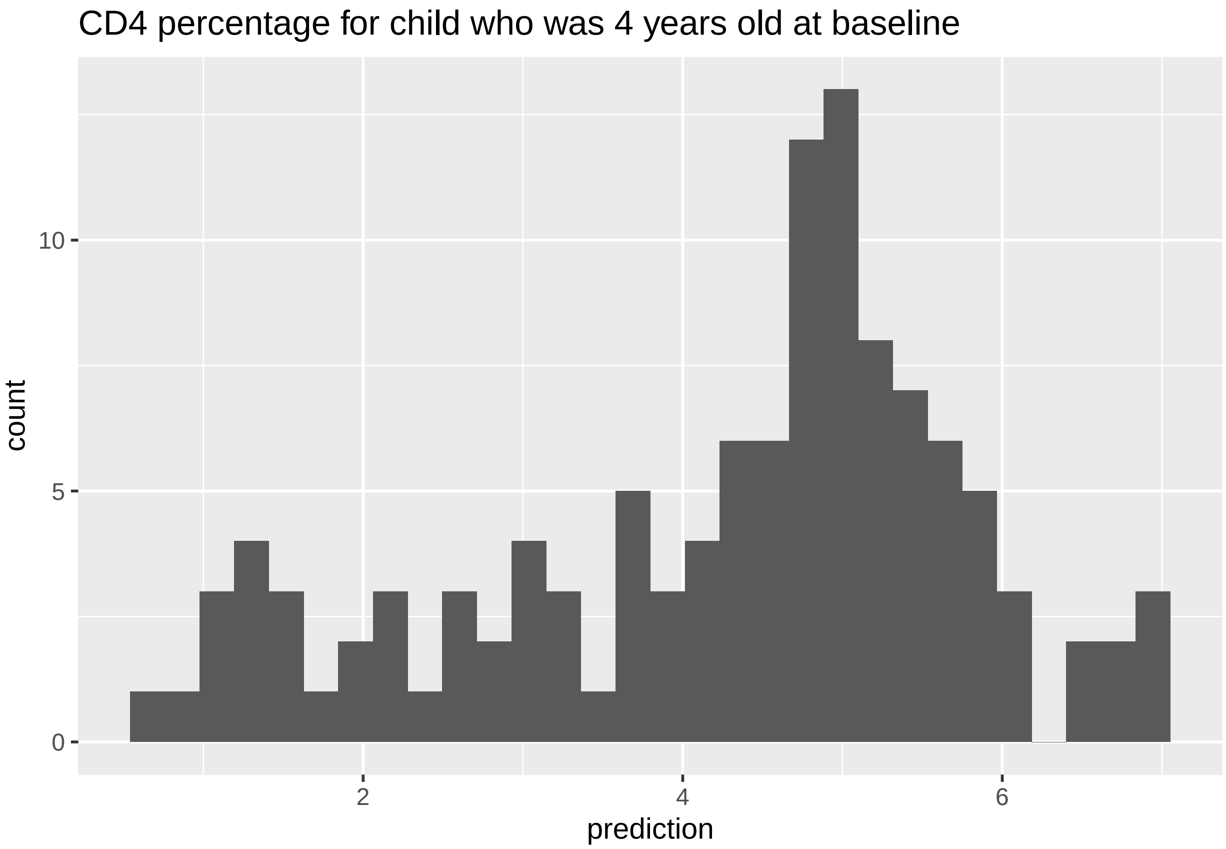
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



8. Use the same model fit to generate simulations of CD4 percentages at each of the time periods for a new child who was 4 years old at baseline.

```
hiv.pred2=hiv.pred[round(hiv.pred$age.baseline)==4,]
hiv.pred.result2 <- predict(lf3,newdata=hiv.pred2)
pred.compare2 <- cbind(hiv.pred.result2,hiv.pred2)
colnames(pred.compare2)[1] <- c("prediction")
ggplot(pred.compare2,aes(x=prediction))+
  geom_histogram()+
  ggtitle("CD4 percentage for child who was 4 years old at baseline")
```

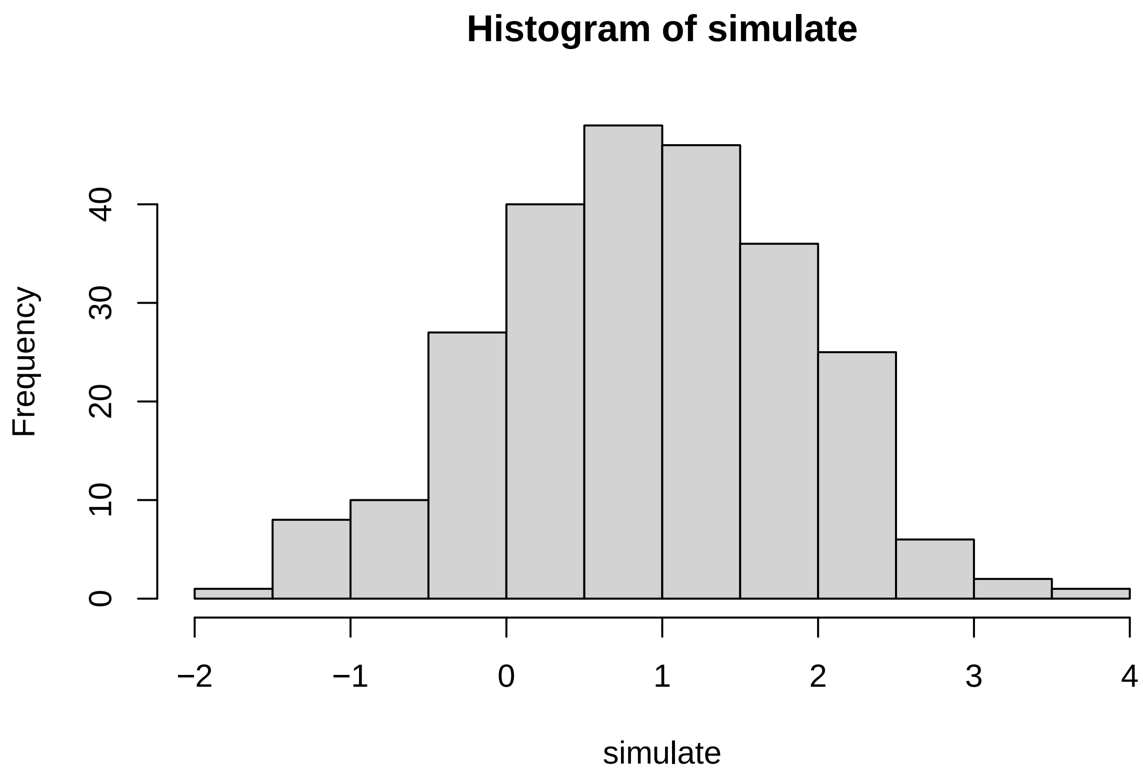
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



9. Posterior predictive checking: continuing the previous exercise, use the fitted model from (5) to simulate a new dataset of CD4 percentages (with the same sample size and ages of the original dataset) for the final time point of the study, and record the average CD4 percentage in this sample. Repeat this process 1000 times and compare the simulated distribution to the observed CD4 percentage at the final time point for the actual data.

```
hiv.pred3 <- hiv.data[,list(time=max(time),age.baseline=unique(age.baseline),
                       treatment=unique(treatment)),by =newpid]
lf3_coef <- coef(lf3)$newpid
est3 <- sigma.hat(lf3)$sigma$data
pred3 <- lf3_coef[,1]+lf3_coef[,2]*hiv.pred3$time+lf3_coef[,3]*hiv.pred3$age.baseline+lf3_coef[,4]*(hiv

simulate <- matrix(NA,nrow(hiv.pred3),1000)
for (i in 1:1000){
  y<-rnorm(pred3,est3)
  simulate[,1]<-y
}
 hist(simulate)
```



10. Extend the model to allow for varying slopes for the time predictor.

```
lf4 <- lmer(y~time+factor(treatment)+age.baseline+(1+time|newpid), data = hiv.data)
display(lf4)
```

```
## lmer(formula = y ~ time + factor(treatment) + age.baseline +
##     (1 + time | newpid), data = hiv.data)
##                    coef.est coef.se
## (Intercept)         5.11     0.19
## time               -0.35     0.07
## factor(treatment)2  0.16     0.18
## age.baseline       -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  newpid   (Intercept) 1.36
##           time        0.58     -0.04
##  Residual             0.72
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3123, DIC = 3081.6
## deviance = 3094.3
```

11. Next fit a model that does not allow for varying slopes but does allow for different coefficients for each time point (rather than fitting the linear trend).

```
lf5 <- lmer(y~factor(time)+(1|newpid), data = hiv.data)
#display(lf5)
```

12. Compare the results of these models both numerically and graphically.

```
anova(lf1,lf2,lf3,lf4,lf5)
```

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

## Data: hiv.data
## Models:
## lf1: y ~ 1 + time + (1 | newpid)
## lf2: y ~ time + (1 | newpid)
## lf3: y ~ time + treatment + age.baseline + (1 | newpid)
## lf4: y ~ time + factor(treatment) + age.baseline + (1 + time | newpid)
## lf5: y ~ factor(time) + (1 | newpid)
##     npar    AIC    BIC  logLik deviance    Chisq  Df Pr(>Chisq)
## lf1    4 3141.9 3161.8 -1566.9   3133.9
## lf2    4 3141.9 3161.8 -1566.9   3133.9   0.0000   0   1.000000
## lf3    6 3136.1 3165.9 -1562.0   3124.1   9.7956   2   0.007463 **
## lf4    8 3110.3 3150.1 -1547.1   3094.3  29.7893   2  3.399e-07 ***
## lf5  405 3244.5 5260.3 -1217.3   2434.5 659.7525 397  2.261e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure skate in the 1932 Winter Olympics

The folder olympics has seven judges' ratings of seven figure skaters (on two criteria: "technical merit" and "artistic impression") from the 1932 Winter Olympics. Take a look at http://www.stat.columbia.edu/~gelman/arm/examples/olympics/olympics1932.txt

1. Construct a 7 × 7 × 2 array of the data (ordered by skater, judge, and judging criterion).

```
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:data.table':
##
##     melt

## The following object is masked from 'package:Matrix':
##
##     expand
```

```
array <- melt(data = olympics1932,id.vars=c("pair","criterion"),measure.vars=c(colnames(olympics1932)[3
#array
```

2. Reformulate the data as a 98 × 4 array (similar to the top table in Figure 11.7), where the first two columns are the technical merit and artistic impression scores, the third column is a skater ID, and the fourth column is a judge ID.

```
array2 <- rename(array, c("pair"="skater_ID", "variable"="judge_ID"))
array2 <- array2 [order(array2 $judge_ID),]
array2 <- array2 [c("criterion", "value", "skater_ID", "judge_ID")]
#array2
```

3. Add another column to this matrix representing an indicator variable that equals 1 if the skater and judge are from the same country, or 0 otherwise.

pairs
1 Andree Brunet, Pierre Brunet, France
2 Beatrix Loughran, Sherwin Badger, United States

3 Emilia Rotter, Laszlo Szollas, Hungary
4 Olva Oronista, Sandor Szalay, Hungary
5 Constance Wilson-Samuel, Montgomery Wilson, Canada
6 Frances Claudet, Chauncey Bangs, Canada
7 Gertrude Meredith, Joseph K. Savage, United States

judges 1 Jeno Minich, Hungary
2 Yngvar Bryn, Norway
3 Hans Grunauer, Austria
4 Walter Jakobsson, Finland
5 George Torchon, France
6 Herbert J. Clarke, Great Britain
7 Charles M. Rotch, United States

```
array2$countryind=ifelse(array2[,3] == " 1"& array2[,4] == "judge_5",1,
  ifelse(array2[,3] == " 2" & array2[,4] == "judge_7",1,
  ifelse(array2[,3] == " 3" & array2[,4] == "judge_1",1,
  ifelse(array2[,3] == " 4" & array2[,4] == "judge_1",1,
  ifelse(array2[,3] == " 7" & array2[,4] == "judge_7",1,0
  )))))
```

4. Write the notation for a non-nested multilevel model (varying across skaters and judges) for the technical merit ratings and fit using lmer().

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:reshape':
##
##     rename

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
tech <- array2 %>%
  filter(criterion=="Program")
```

```r
lm_tech <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID),data=tech)
display(lm_tech)
```

```
## lmer(formula = value ~ 1 + (1 | skater_ID) + (1 | judge_ID),
##     data = tech)
## coef.est  coef.se
##     5.13     0.20
##
## Error terms:
##  Groups    Name        Std.Dev.
##  skater_ID (Intercept) 0.42
##  judge_ID  (Intercept) 0.28
##  Residual              0.33
## ---
## number of obs: 49, groups: skater_ID, 7; judge_ID, 7
## AIC = 68, DIC = 57
## deviance = 58.5
```

5. Fit the model in (4) using the artistic impression ratings.

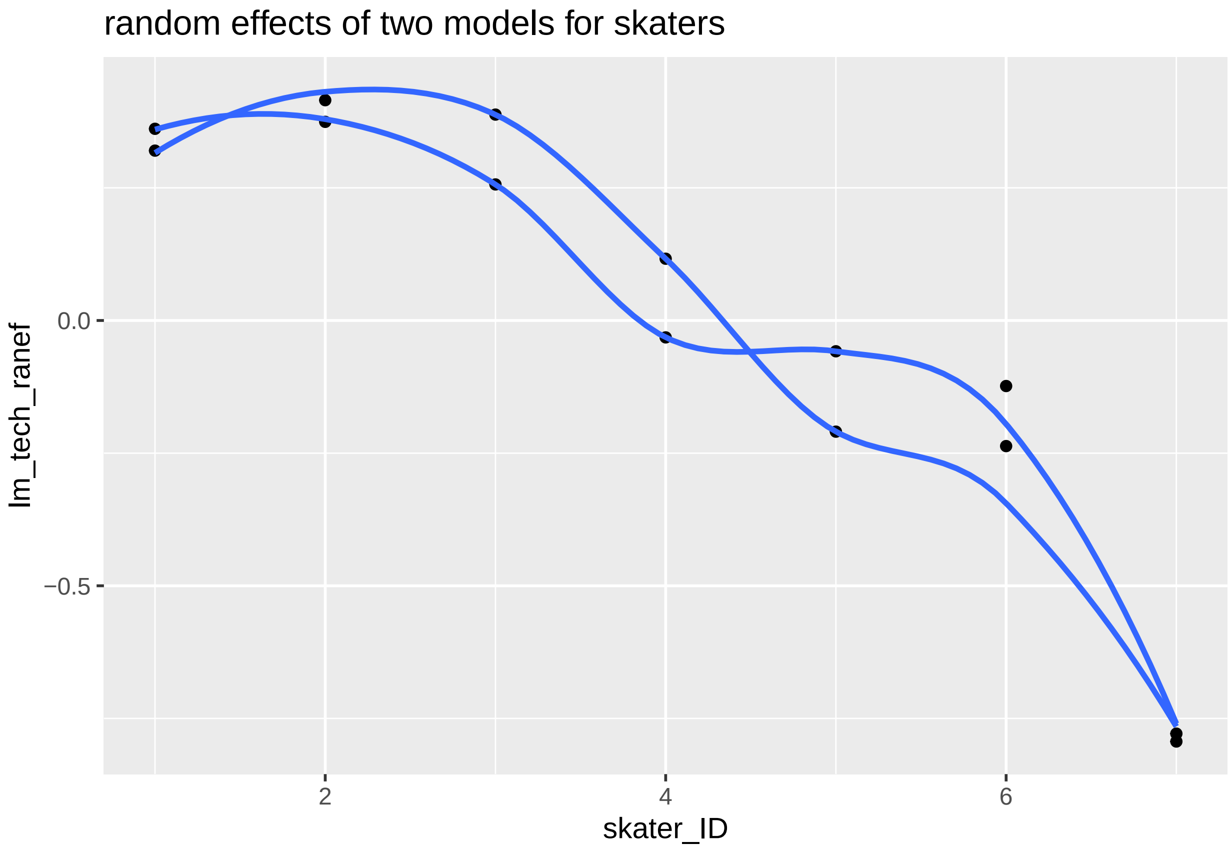
```r
art <- array2 %>%
  filter(criterion=="Performance")
lm_art <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID),data=art)
display(lm_art)
```

```
## lmer(formula = value ~ 1 + (1 | skater_ID) + (1 | judge_ID),
##     data = art)
## coef.est  coef.se
##     5.09     0.20
##
## Error terms:
##  Groups    Name        Std.Dev.
##  skater_ID (Intercept) 0.45
##  judge_ID  (Intercept) 0.28
##  Residual              0.27
## ---
## number of obs: 49, groups: skater_ID, 7; judge_ID, 7
## AIC = 54.2, DIC = 43.4
## deviance = 44.8
```

6. Display your results for both outcomes graphically.

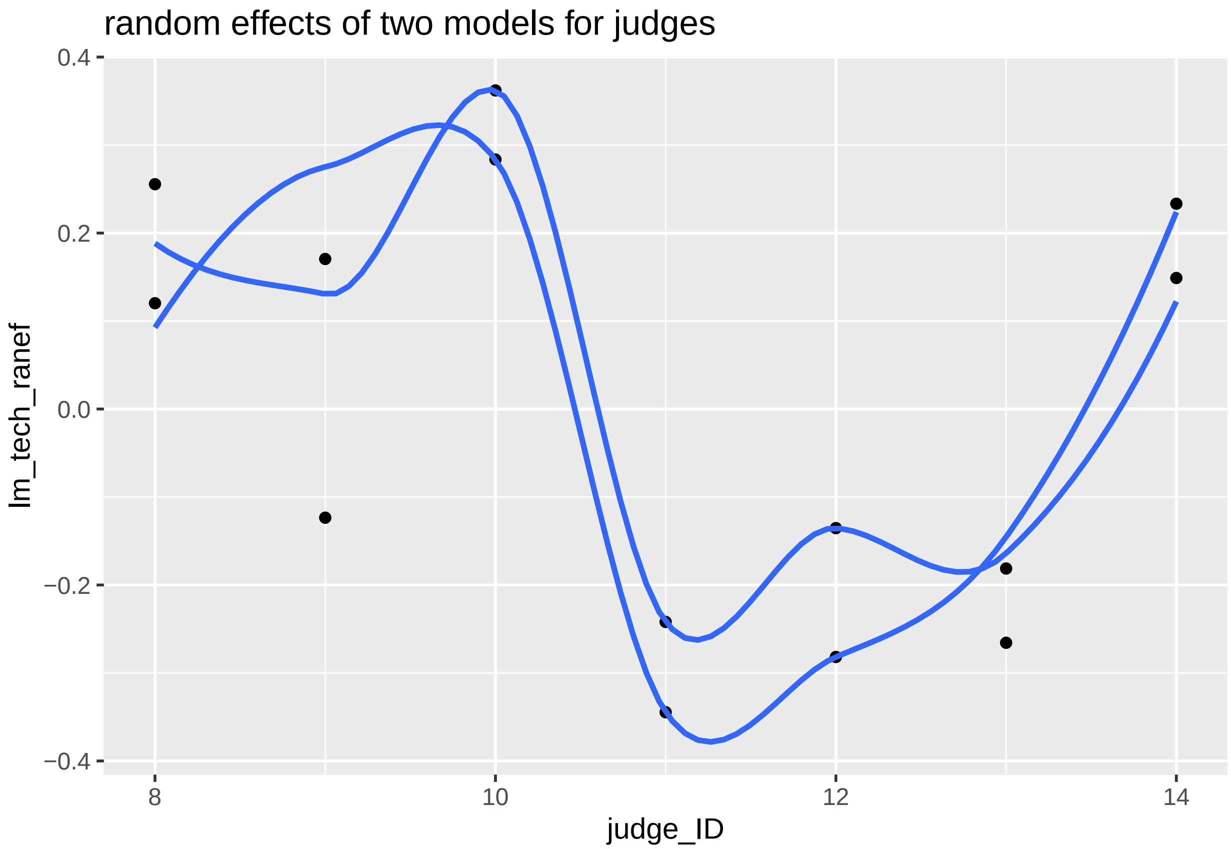
```r
#for stake
skate <- as.data.frame(cbind(unlist(ranef(lm_tech))[1:7],unlist(ranef(lm_art))[1:7]))
skate$skater_ID <-c(1:7)
colnames(skate)[1]="lm_tech_ranef"
colnames(skate)[2]="lm_art_ranef"
ggplot(data=skate)+
  geom_point(aes(x=skater_ID,y=lm_tech_ranef))+
  geom_smooth(aes(x=skater_ID,y=lm_tech_ranef),se=FALSE)+
  geom_point(aes(x=skater_ID,y=lm_art_ranef))+
  geom_smooth(aes(x=skater_ID,y=lm_art_ranef),se=FALSE)+
  ggtitle("random effects of two models for skaters")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
judge <- as.data.frame(cbind(unlist(ranef(lm_tech))[8:14],unlist(ranef(lm_art))[8:14]))
judge$judge_ID <-c(8:14)
colnames(judge)[1]="lm_tech_ranef"
colnames(judge)[2]="lm_art_ranef"
ggplot(data=judge)+
  geom_point(aes(x=judge_ID,y=lm_tech_ranef))+
  geom_smooth(aes(x=judge_ID,y=lm_tech_ranef),se=FALSE)+
  geom_point(aes(x=judge_ID,y=lm_art_ranef))+
  geom_smooth(aes(x=judge_ID,y=lm_art_ranef),se=FALSE)+
  ggtitle("random effects of two models for judges")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



7. (optional) Use posterior predictive checks to investigate model fit in (4) and (5).

Different ways to write the model:

Using any data that are appropriate for a multilevel model, write the model in the five ways discussed in Section 12.5 of Gelman and Hill.

I use the hiv data.

- Allowing regression coefficeints to vary accross groups

$$y = 4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * 0.18 + 0.77$$

- Combining separate local regressions

$$y \sim\ N(4.91{+}time_i{*}(-0.36){+}treatment_i{*}(-0.12){+}age.baseline_i{*}(0.18), 0.77^2)\alpha_j \sim\ \mathrm{N}(RandomIntercept, 1.37^2)$$

- Modeling the coefficients of a large regression model

$$y_i \sim\ N(4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * (0.18), 0.77^2)\alpha_j \sim\ \mathrm{N}(0, 1.37^2)$$

- Regression with multiple error terms

$$y_i \sim\ N(4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * (0.18) + 1.37^2, 0.77^2)$$

#5th method: Large regression with correlated errors

$$y_i \sim\ N(4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * (0.18), 1.37^2 + 0.77^2)$$

Models for adjusting individual ratings:

A committee of 10 persons is evaluating 100 job applications. Each person on the committee reads 30 applications (structured so that each application is read by three people) and gives each a numerical rating between 1 and 10.

1. It would be natural to rate the applications based on their combined scores; however, there is a worry that different raters use different standards, and we would like to correct for this. Set up a model for the ratings (with parameters for the applicants and the raters).

```
lmer(scores~applicant_id+rater_id+(1|rater_id))
```

2. It is possible that some persons on the committee show more variation than others in their ratings. Expand your model to allow for this.

```
lmer(scores~applicant_id+rater_id+(1+rater_id|rater_id))
```

Multilevel logistic regression

The folder `speed.dating` contains data from an experiment on a few hundred students that randomly assigned each participant to 10 short dates with participants of the opposite sex (Fisman et al., 2006). For each date, each person recorded several subjective numerical ratings of the other person (attractiveness, compatibility, and some other characteristics) and also wrote down whether he or she would like to meet the other person again. Label $y_{ij} = 1$ if person i is interested in seeing person j again 0 otherwise. And $r_{ij1}, \dots, r_{ij6}$ as person i's numerical ratings of person j on the dimensions of attractiveness, compatibility, and so forth. Please look at http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/Speed%20Dating%20Data%20Key.doc for details.

```
dating<-read.csv("Speed Dating Data.csv")
```

1. Fit a classical logistic regression predicting $Pr(y_{ij} = 1)$ given person i's 6 ratings of person j. Discuss the importance of attractiveness, compatibility, and so forth in this predictive model.

```
dating_complete_pool <- glm(match~attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o,data=dating,family=binom
display(dating_complete_pool)
```

```
## glm(formula = match ~ attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o, family = binomial, data = dating)
##             coef.est coef.se
## (Intercept) -5.62     0.22
## attr_o       0.22     0.02
## sinc_o      -0.02     0.03
## intel_o      0.07     0.04
## fun_o        0.25     0.03
## amb_o       -0.12     0.03
## shar_o       0.21     0.02
## ---
##   n = 7031, k = 7
##   residual deviance = 5611.0, null deviance = 6466.6 (difference = 855.6)
```

2. Expand this model to allow varying intercepts for the persons making the evaluation; that is, some people are more likely than others to want to meet someone again. Discuss the fitted model.

```
dating_pooled_1 <- glmer(match~gender + attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o+(1|iid),data=datin
display(dating_pooled_1)
```

```
## glmer(formula = match ~ gender + attr_o + sinc_o + intel_o +
##     fun_o + amb_o + shar_o + (1 | iid), data = dating, family = binomial)
##             coef.est coef.se
## (Intercept) -6.02     0.24
## gender       0.15     0.09
## attr_o       0.24     0.03
## sinc_o      -0.01     0.03
## intel_o      0.07     0.04
## fun_o        0.26     0.03
## amb_o       -0.13     0.03
## shar_o       0.22     0.02
##
## Error terms:
##  Groups   Name        Std.Dev.
##  iid      (Intercept) 0.65
##  Residual             1.00
## ---
## number of obs: 7031, groups: iid, 551
## AIC = 5543.3, DIC = 4599.4
## deviance = 5062.3
```

3. Expand further to allow varying intercepts for the persons being rated. Discuss the fitted model.

```
dating_pooled_2 <- glmer(match~gender + attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o+(1|iid)+(1|pid),da
display(dating_pooled_2)
```

```
## glmer(formula = match ~ gender + attr_o + sinc_o + intel_o +
##     fun_o + amb_o + shar_o + (1 | iid) + (1 | pid), data = dating,
##     family = binomial)
##             coef.est coef.se
## (Intercept) -8.26     0.38
## gender       0.17     0.15
## attr_o       0.34     0.03
## sinc_o       0.02     0.04
## intel_o      0.11     0.05
## fun_o        0.30     0.04
## amb_o       -0.09     0.04
## shar_o       0.26     0.03
##
## Error terms:
##  Groups   Name        Std.Dev.
##  iid      (Intercept) 0.78
##  pid      (Intercept) 1.12
##  Residual             1.00
## ---
## number of obs: 7031, groups: iid, 551; pid, 537
## AIC = 5257.8, DIC = 2699
## deviance = 3968.4
```

4. You will now fit some models that allow the coefficients for attractiveness, compatibility, and the other attributes to vary by person. Fit a no-pooling model: for each person i, fit a logistic regression to the data y_{ij} for the 10 persons j whom he or she rated, using as predictors the 6 ratings $r_{ij1}, \dots, r_{ij6}$. (Hint: with 10 data points and 6 predictors, this model is difficult to fit. You will need to simplify it in some way to get reasonable fits.)

```
uiid<-unique(dating$iid)
dating_no_pool_list<-vector("list",length(uiid))
for(i in 1:length(uiid)){
#  attr_o +sinc_o +intel_o +fun_o +amb_o+shar_o,
dating_no_pool_list[[i]] <- summary(glm(match~attr_o+shar_o,
                       data=dating,
                       subset = dating$iid==uiid[i],
                       family=binomial))$coefficients
}
#dating_no_pool_list
```

5. Fit a multilevel model, allowing the intercept and the coefficients for the 6 ratings to vary by the rater i.

```
dating_pooled_3 <- stan_glmer(match~gender + attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o+(1+attr_o +si
#display(dating_pooled_3)
```

6. Compare the inferences from the multilevel model in (5) to the no-pooling model in (4) and the complete-pooling model from part (1) of the previous exercise.

```
anova(dating_pooled_3,dating_complete_pool)
```