MA678 Homework 7

Paul Moon

November 14, 2024

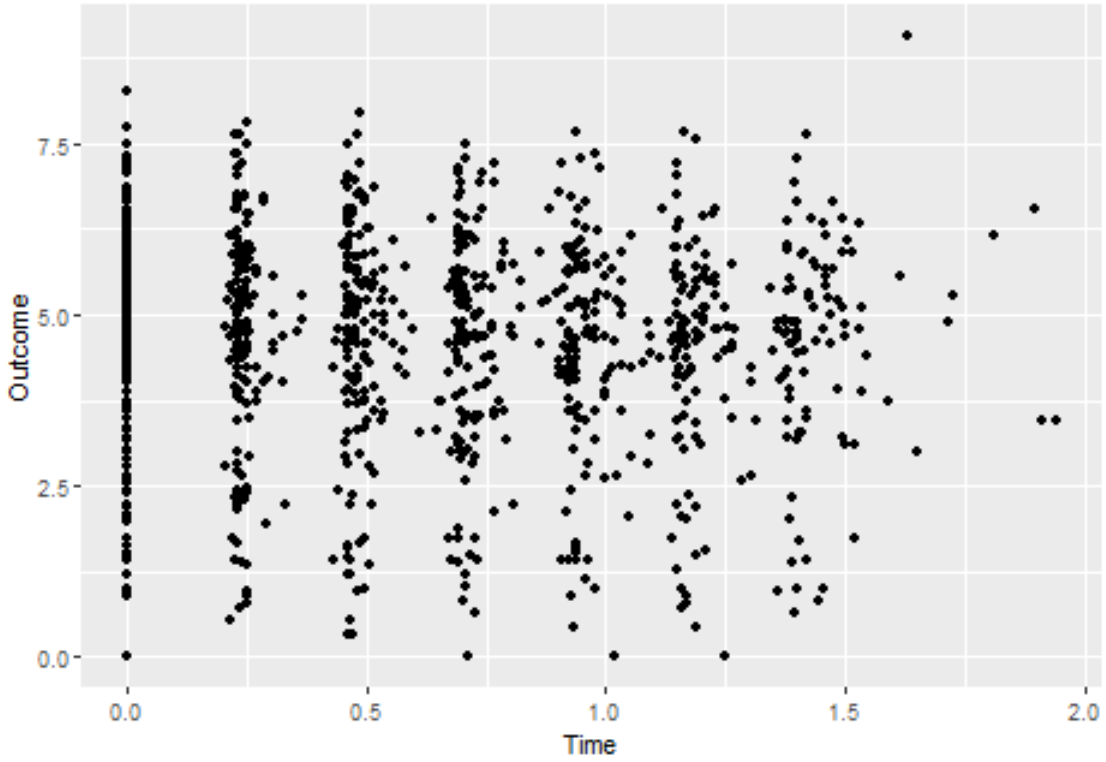
Data analysis

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.

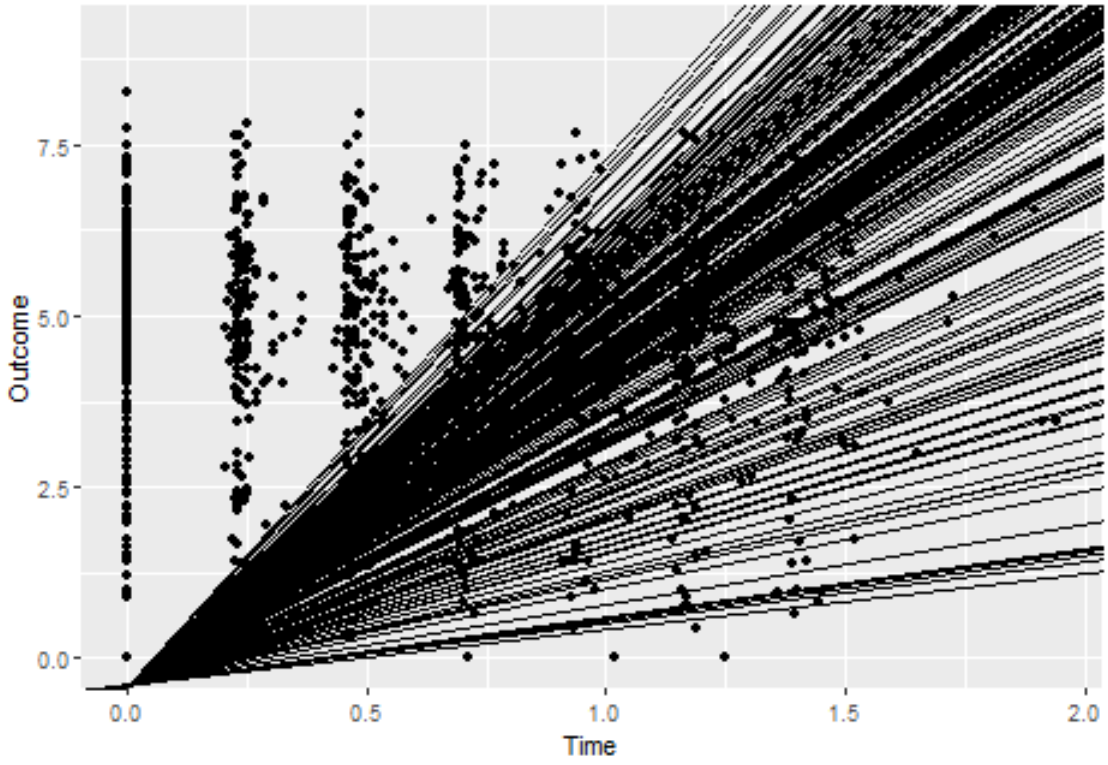
```
ggplot(hiv.data) +
  geom_point(aes(x = time, y = y)) +
  xlab("Time") +
  ylab("Outcome")
```



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.

```
r_np <- lm(y ~ time + factor(newpid) - 1,
           data = hiv.data)

ggplot(aes(x = time, y = y), data = hiv.data) +
  geom_point() +
  geom_abline(intercept = coef(r_np)[1],
              slope = coef(r_np)[2 : length(coef(r_np))]) +
  xlab("Time") + ylab("Outcome")
```



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.

```
r1 <- lm(y ~ time + factor(newpid) - 1, data = hiv.data)

child <- hiv.data %>%
  select(newpid, age.baseline, treatment)
child <- unique(child)
r1.coef <- data.frame(child, r1$coefficients[2 : length(r1$coefficients)])
colnames(r1.coef) <- c("newpid", "age.baseline", "treatment","coef.id")
rownames(r1.coef) <- 1 : 250

r1_coef.id <- lm(coef.id ~ age.baseline + factor(treatment), data = r1.coef)
summary(r1_coef.id)
```

```
##
## Call:
## lm(formula = coef.id ~ age.baseline + factor(treatment), data = r1.coef)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1594 -0.7039  0.2265  1.1215  2.7256
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.10627    0.18728  27.265  < 2e-16 ***
## age.baseline        -0.12088    0.04023  -3.005  0.00293 **
## factor(treatment)2   0.14558    0.18421   0.790  0.43012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.455 on 247 degrees of freedom
## Multiple R-squared:  0.03753,    Adjusted R-squared:  0.02974
## F-statistic: 4.816 on 2 and 247 DF,  p-value: 0.008875
```

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.

```
M0 <- lmer (y ~ time + (1 | newpid), data = hiv.data)
display(M0)
```

```
## 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
```

```
M0.coef <- data.frame(unique(hiv.data$newpid) ,coef(M0)$newpid)
colnames(M0.coef) <- c("newpid", "intercept", "time")
head(coef(M0)$newpid)
```

```
##   (Intercept)       time
## 1    4.557250 -0.3660932
## 2    1.335566 -0.3660932
## 3    5.884129 -0.3660932
## 4    5.561130 -0.3660932
## 5    4.178397 -0.3660932
## 6    5.326751 -0.3660932
```

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.

```
M1 <- lmer (y ~ time + factor(treatment) + age.baseline +
              (1 | newpid), data = hiv.data)
display(M1)
```

```
## lmer(formula = y ~ time + factor(treatment) + age.baseline +
##     (1 | newpid), data = hiv.data)
##                     coef.est coef.se
## (Intercept)          5.09     0.19
## time                -0.36     0.05
## factor(treatment)2   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
```

```
head(coef(M1)$newpid)
```

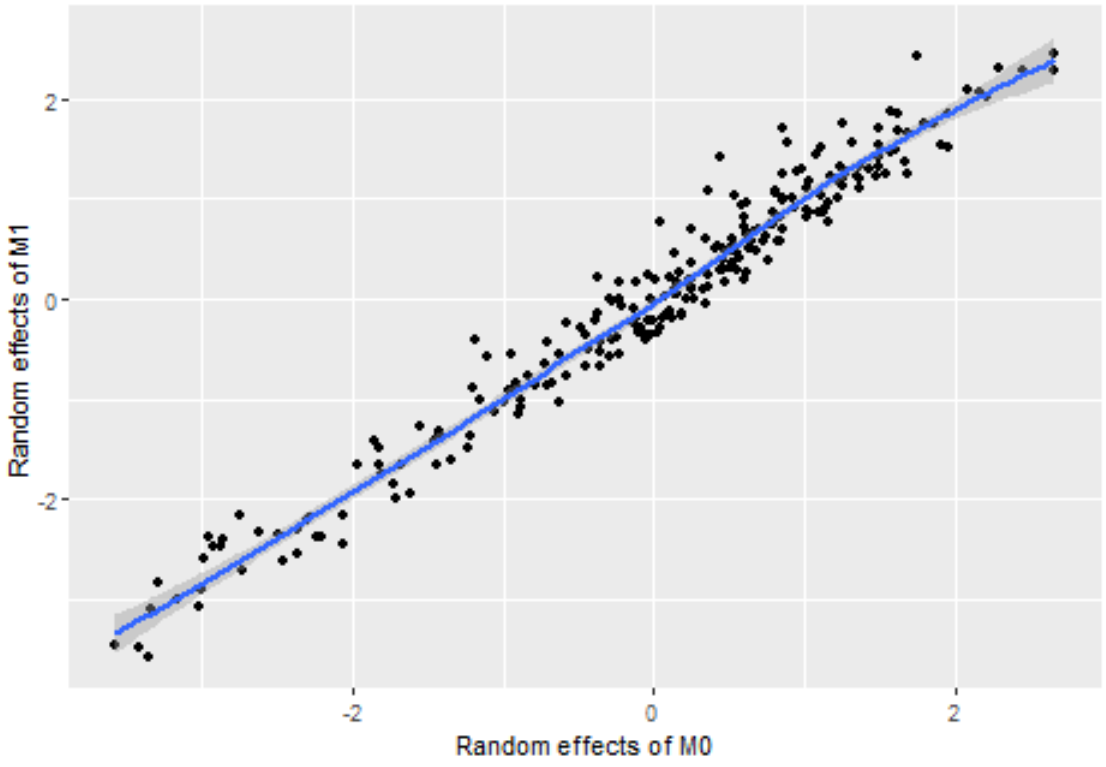
```
##   (Intercept)       time factor(treatment)2 age.baseline
## 1    5.012677 -0.3621573          0.1800822   -0.1194538
## 2    1.607624 -0.3621573          0.1800822   -0.1194538
## 3    6.593175 -0.3621573          0.1800822   -0.1194538
## 4    5.834945 -0.3621573          0.1800822   -0.1194538
## 5    4.320103 -0.3621573          0.1800822   -0.1194538
## 6    5.499405 -0.3621573          0.1800822   -0.1194538
```

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

```
data_plot <- as.data.frame(cbind(unlist(ranef(M0)), unlist(ranef(M1))))
colnames(data_plot) <- c("M0", "M1")

ggplot(data = data_plot, aes(x = M0,y = M1)) + geom_point() + geom_smooth() +
  xlab("Random effects of M0")+
  ylab("Random effects of M1")
```

```
## `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.

```
library(dplyr)
predict_data <- hiv.data %>%
  filter(is.na(hiv.data$treatment) == FALSE) %>%
  filter(is.na(age.baseline) == FALSE) %>%
  select(time,treatment,age.baseline,newpid,y)
predict_new <- predict(M1,newdata = predict_data)
predict_cmb <- cbind(predict_data, predict_new)
colnames(predict_cmb)[1] <- c("prediction")
```

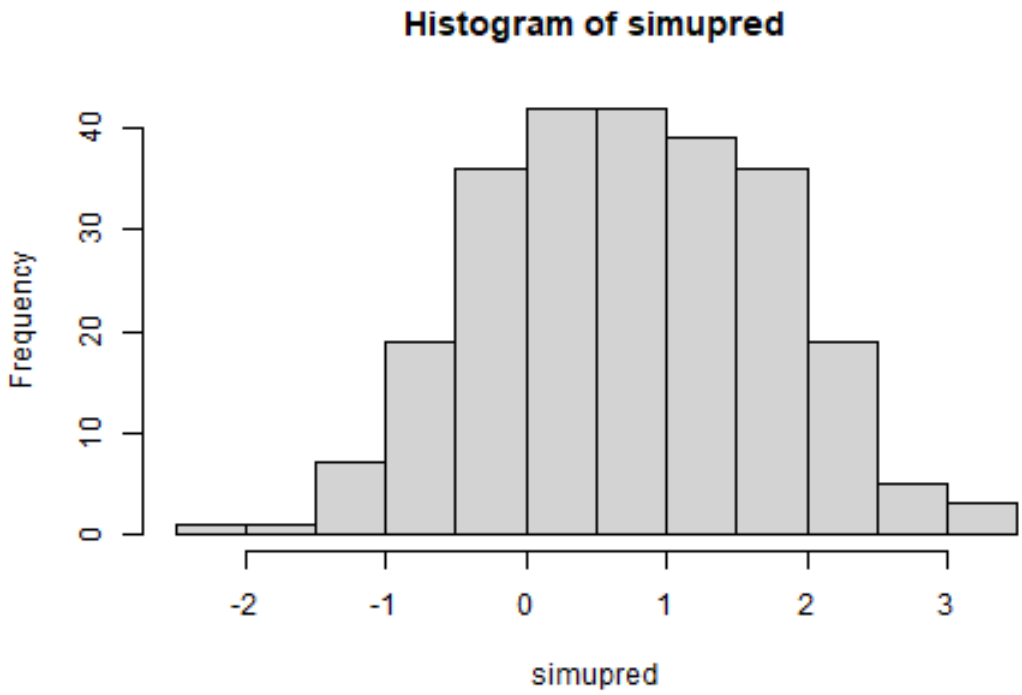
8. Use the same model fit to generate simulations of CD4 percentages at each of the time periods for a new child who was 4 years old at baseline.

```
predict_data2 <- hiv.data %>%
  filter(is.na(hiv.data$treatment)==FALSE) %>%
  filter(is.na(age.baseline)==FALSE) %>%
  select(time,treatment,age.baseline,newpid,y) %>%
  filter(round(age.baseline)==4)
predict_new2 <- predict(M1,newdata=predict_data2)
predict_cmb2 <- cbind(predict_data2, predict_new2)
colnames(predict_cmb2)[1] <- c("prediction")
```

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.

```
pred <- hiv.data[, list(time = max(time), age.baseline = unique(age.baseline),
                        treatment = unique(treatment)), by = newpid]
cm <- coef(M1)$newpid
sigy <- sigma.hat(M1)$sigma$data
predy <- cm[, 1]+cm[, 2] * pred$time +
  cm[, 3] * pred$age.baseline + cm[, 4] * (pred$treatment - 1)
avg.pred.CD4PCT <- NULL
simupred <- matrix(NA, nrow(pred), 1000)
for (i in 1 : 1000){
  ytilde <- rnorm(predy, sigy)
  simupred[, 1] <- ytilde
}
hist(simupred)
```



10. Extend the modelto allow for varying slopes for the time predictor.

```
M2 <- lmer(hiv.data$y ~ hiv.data$time + (1 + hiv.data$time | hiv.data$newpid))
```

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).

```
M3 <- lmer(hiv.data$y ~ factor(hiv.data$time) + (1 | hiv.data$newpid))
```

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

```
data_plot2_inter <- as.data.frame(cbind(unlist(ranef(M2))[1 : 250],unlist(ranef(M3))[1 : 250]))
colnames(data_plot2_inter) <- c("M2", "M3")

ggplot(data = data_plot2_inter,aes(x = M2, y = M3)) +
  geom_point() +
  geom_smooth() +
  xlab("Random effcts of M2 intercepts") +
  ylab("Random effects of M3 intercepts")
```

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

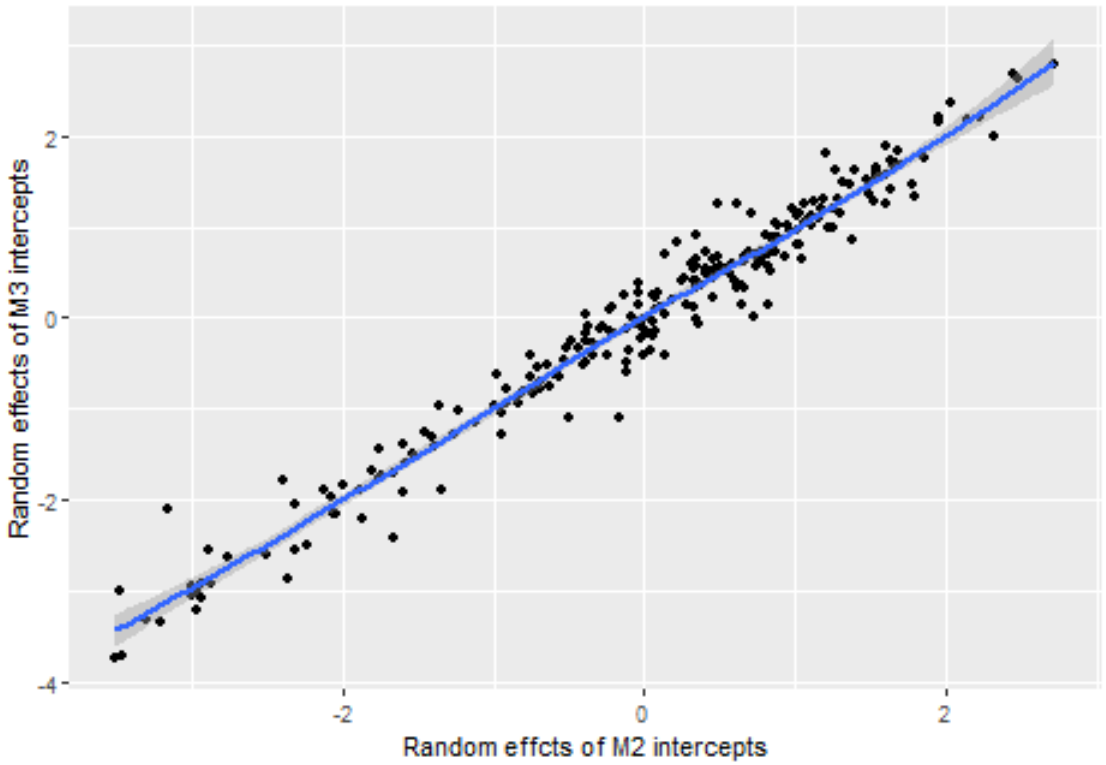


Figure skate in the 1932 Winter Olympics

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 $7 \times 7 \times 2$ array of the data (ordered by skater, judge, and judging criterion).

```
arr_olym <- melt(data = olympics1932, id.vars = c("pair", "criterion"),
                 measure.vars = c(colnames(olympics1932)[3 : 9]))
arr_olym
```

```
##    pair   criterion variable value
## 1     1     Program  judge_1   5.6
```

```
## 2     1 Performance  judge_1   5.6
## 3     2     Program  judge_1   5.5
## 4     2 Performance  judge_1   5.5
## 5     3     Program  judge_1   6.0
## 6     3 Performance  judge_1   6.0
## 7     4     Program  judge_1   5.6
## 8     4 Performance  judge_1   5.6
## 9     5     Program  judge_1   5.4
## 10    5 Performance  judge_1   4.8
## 11    6     Program  judge_1   5.2
## 12    6 Performance  judge_1   4.8
## 13    7     Program  judge_1   4.8
## 14    7 Performance  judge_1   4.3
## 15    1     Program  judge_2   5.5
## 16    1 Performance  judge_2   5.5
## 17    2     Program  judge_2   5.2
## 18    2 Performance  judge_2   5.7
## 19    3     Program  judge_2   5.3
## 20    3 Performance  judge_2   5.5
## 21    4     Program  judge_2   5.3
## 22    4 Performance  judge_2   5.3
## 23    5     Program  judge_2   4.5
## 24    5 Performance  judge_2   4.8
## 25    6     Program  judge_2   5.1
## 26    6 Performance  judge_2   5.6
## 27    7     Program  judge_2   4.0
## 28    7 Performance  judge_2   4.6
## 29    1     Program  judge_3   5.8
## 30    1 Performance  judge_3   5.8
## 31    2     Program  judge_3   5.8
## 32    2 Performance  judge_3   5.6
## 33    3     Program  judge_3   5.8
## 34    3 Performance  judge_3   5.7
## 35    4     Program  judge_3   5.8
## 36    4 Performance  judge_3   5.8
## 37    5     Program  judge_3   5.8
## 38    5 Performance  judge_3   5.5
## 39    6     Program  judge_3   5.3
## 40    6 Performance  judge_3   5.0
## 41    7     Program  judge_3   4.7
## 42    7 Performance  judge_3   4.5
## 43    1     Program  judge_4   5.3
## 44    1 Performance  judge_4   4.7
## 45    2     Program  judge_4   5.8
## 46    2 Performance  judge_4   5.4
## 47    3     Program  judge_4   5.0
## 48    3 Performance  judge_4   4.9
## 49    4     Program  judge_4   4.4
## 50    4 Performance  judge_4   4.8
## 51    5     Program  judge_4   4.0
## 52    5 Performance  judge_4   4.4
## 53    6     Program  judge_4   5.4
## 54    6 Performance  judge_4   4.7
## 55    7     Program  judge_4   4.0
```

```
## 56    7 Performance  judge_4   4.0
## 57    1     Program  judge_5   5.6
## 58    1 Performance  judge_5   5.7
## 59    2     Program  judge_5   5.6
## 60    2 Performance  judge_5   5.5
## 61    3     Program  judge_5   5.4
## 62    3 Performance  judge_5   5.5
## 63    4     Program  judge_5   4.5
## 64    4 Performance  judge_5   4.5
## 65    5     Program  judge_5   5.5
## 66    5 Performance  judge_5   4.6
## 67    6     Program  judge_5   4.5
## 68    6 Performance  judge_5   4.0
## 69    7     Program  judge_5   3.7
## 70    7 Performance  judge_5   3.6
## 71    1     Program  judge_6   5.2
## 72    1 Performance  judge_6   5.3
## 73    2     Program  judge_6   5.1
## 74    2 Performance  judge_6   5.3
## 75    3     Program  judge_6   5.1
## 76    3 Performance  judge_6   5.2
## 77    4     Program  judge_6   5.0
## 78    4 Performance  judge_6   5.0
## 79    5     Program  judge_6   4.8
## 80    5 Performance  judge_6   4.8
## 81    6     Program  judge_6   4.5
## 82    6 Performance  judge_6   4.6
## 83    7     Program  judge_6   4.0
## 84    7 Performance  judge_6   4.0
## 85    1     Program  judge_7   5.7
## 86    1 Performance  judge_7   5.4
## 87    2     Program  judge_7   5.8
## 88    2 Performance  judge_7   5.7
## 89    3     Program  judge_7   5.3
## 90    3 Performance  judge_7   5.7
## 91    4     Program  judge_7   5.1
## 92    4 Performance  judge_7   5.5
## 93    5     Program  judge_7   5.5
## 94    5 Performance  judge_7   5.2
## 95    6     Program  judge_7   5.0
## 96    6 Performance  judge_7   5.2
## 97    7     Program  judge_7   4.8
## 98    7 Performance  judge_7   4.8
```

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.

```
olym_984 <- rename(arr_olym, c("pair" = "skater_ID", "variable" = "judge_ID"))
olym_984 <- olym_984[order(olym_984$judge_ID), ]
olym_984 <- olym_984[c("criterion", "value", "skater_ID", "judge_ID")]
```

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.

```
olym_984$SameCountry <- ifelse(olym_984[, 3] == "1"&olym_984[, 4] == "judge_5", 1,
  ifelse(olym_984[, 3] == " 2"&olym_984[, 4] == "judge_7", 1,
  ifelse(olym_984[, 3] == " 3"&olym_984[, 4] == "judge_1", 1,
  ifelse(olym_984[, 3] == " 4"&olym_984[, 4] == "judge_1", 1,
  ifelse(olym_984[, 3] == " 7"&olym_984[, 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().

```
data_tech <- olym_984 %>%
  filter(criterion == "Program")
data_art <- olym_984 %>%
  filter(criterion == "Performance")

reg_tech <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID), data = data_tech)
summary(reg_tech)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: value ~ 1 + (1 | skater_ID) + (1 | judge_ID)
##    Data: data_tech
##
## REML criterion at convergence: 60
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.51025 -0.45646 -0.05459  0.63866  1.89709
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  skater_ID (Intercept) 0.17488  0.4182
##  judge_ID  (Intercept) 0.07664  0.2768
##  Residual              0.11057  0.3325
## Number of obs: 49, groups:  skater_ID, 7; judge_ID, 7
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   5.1347     0.1954   26.28
```

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

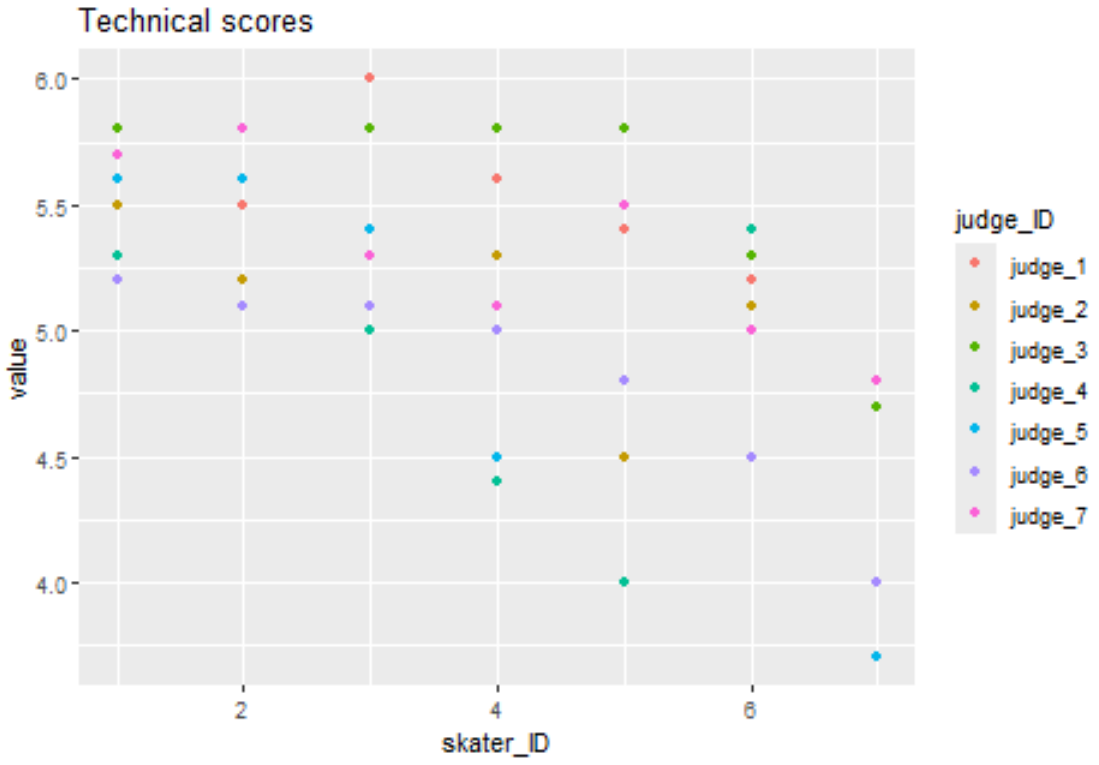
```
reg_art <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID), data = data_art)
summary(reg_tech)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: value ~ 1 + (1 | skater_ID) + (1 | judge_ID)
##    Data: data_tech
##
## REML criterion at convergence: 60
##
## Scaled residuals:
```

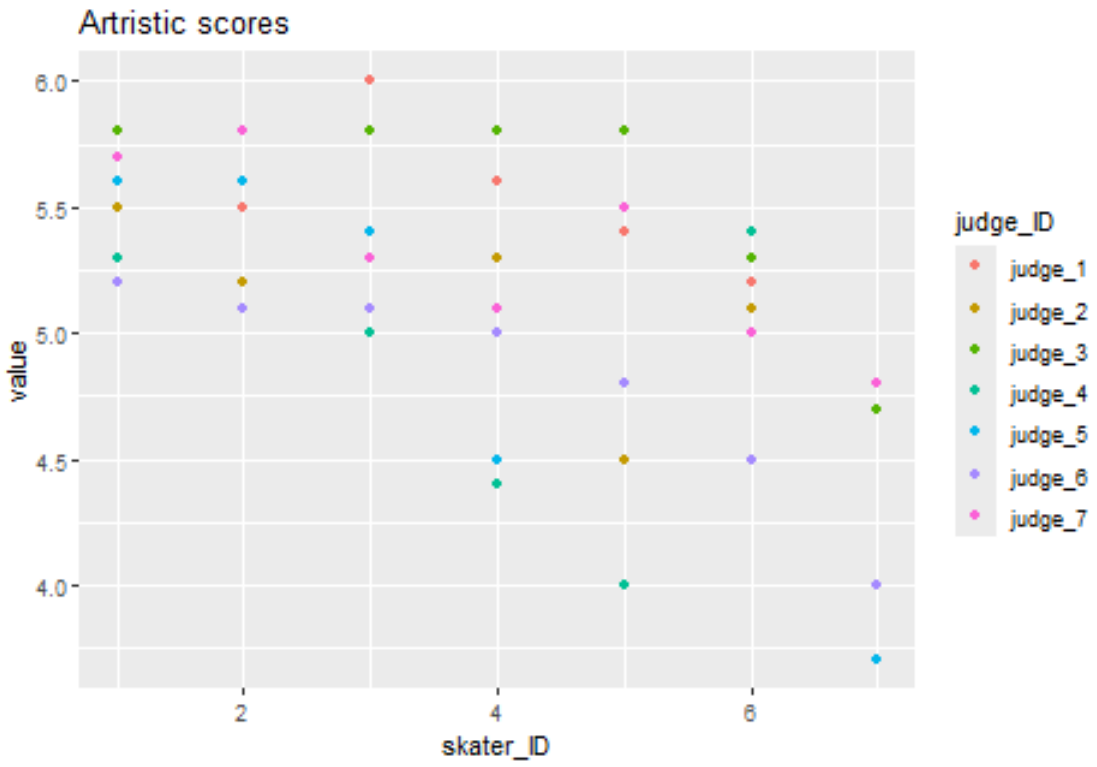
```
##      Min       1Q   Median       3Q      Max
## -2.51025 -0.45646 -0.05459  0.63866  1.89709
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  skater_ID (Intercept) 0.17488  0.4182
##  judge_ID  (Intercept) 0.07664  0.2768
##  Residual              0.11057  0.3325
## Number of obs: 49, groups:  skater_ID, 7; judge_ID, 7
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   5.1347     0.1954   26.28
```

6. Display your results for both outcomes graphically.

```
ggplot(data_tech, aes(x = skater_ID, y = value, color = judge_ID)) +
  geom_point() +
  ggtitle("Technical scores")
```

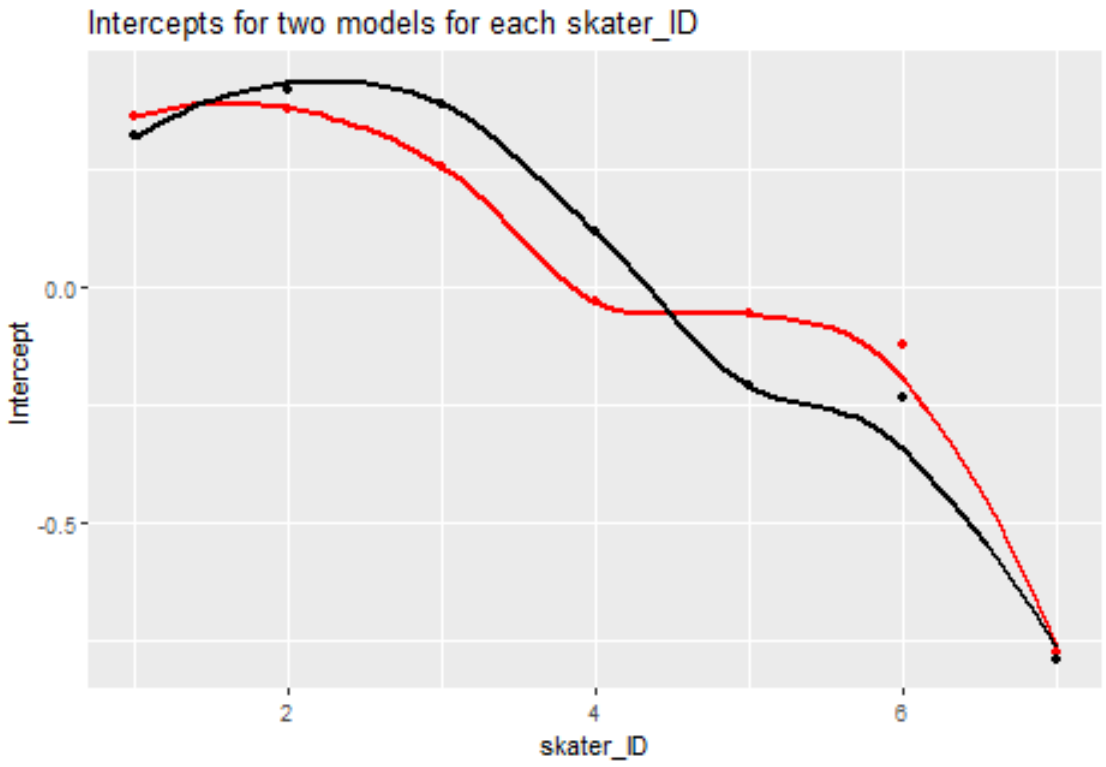



```
ggplot(data_tech, aes(x = skater_ID, y = value, color = judge_ID)) +
  geom_point() +
  ggtitle("Artristic scores")
```



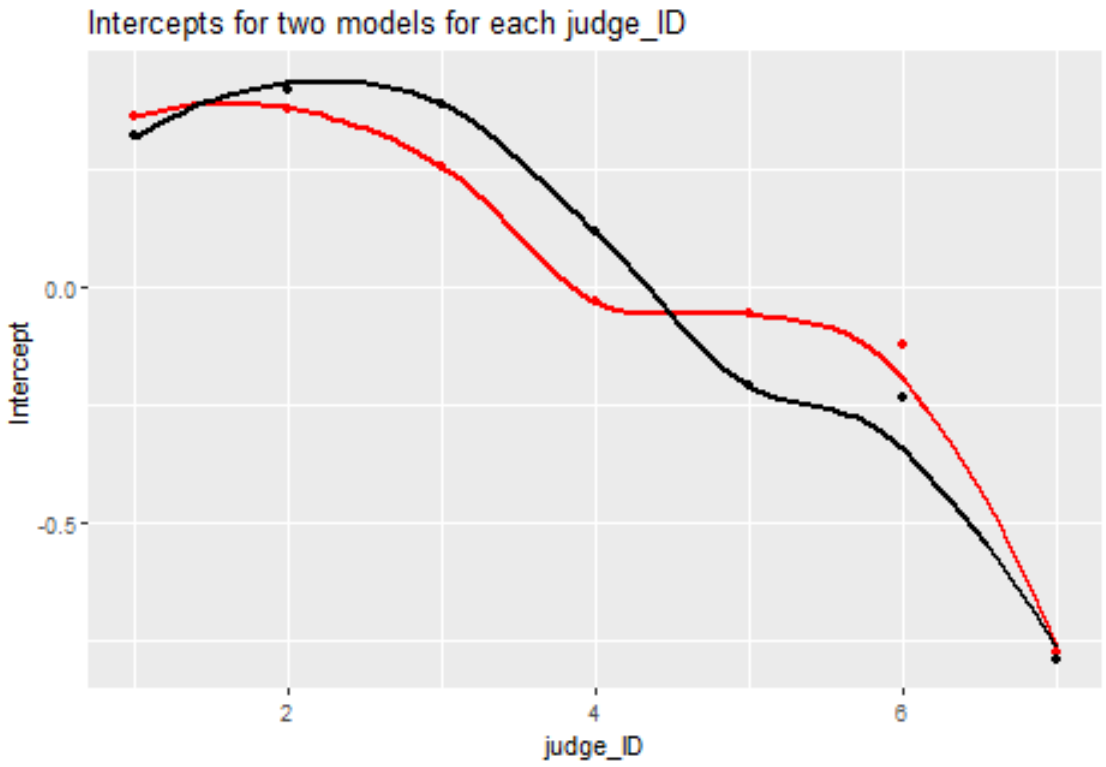

```
inter_skate <- as.data.frame(cbind(unlist(ranef(reg_tech))[1 : 7], unlist(ranef(reg_art))[1 : 7]))
inter_skate$skater_ID <-c(1 : 7)
ggplot(data = inter_skate) +
  geom_point(col = "red", aes(x = skater_ID, y = V1)) +
  geom_smooth(col = "red", aes(x = skater_ID, y = V1), se = FALSE) +
  geom_point(col = "black", aes(x = skater_ID, y = V2)) +
  geom_smooth(col = "black", aes(x = skater_ID, y = V2), se = FALSE) +
  ggtitle("Intercepts for two models for each skater_ID") +
  ylab("Intercept")
```

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




```
inter_judge <- as.data.frame(cbind(unlist(ranef(reg_tech))[1 : 7], unlist(ranef(reg_art))[ 1 :7]))
inter_judge$judge_ID <-c(1 : 7)
ggplot(data = inter_judge) +
  geom_point(col = "red", aes(x = judge_ID, y = V1)) +
  geom_smooth(col = "red", aes(x = judge_ID, y = V1), se = FALSE) +
  geom_point(col = "black", aes(x = judge_ID, y = V2)) +
  geom_smooth(col = "black", aes(x = judge_ID, y = V2), se = FALSE) +
  ggtitle("Intercepts for two models for each judge_ID") +
  ylab("Intercept")
```

```
## `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).

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).

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.

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 $r_{ij1}, \ldots, r_{ij6}$ 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<-fread("http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/Speed%20Dating%20Data.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.
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.
3. Expand further to allow varying intercepts for the persons being rated. Discuss the fitted model.
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 $r_{ij1}, \ldots, r_{ij6}$. (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.)
5. Fit a multilevel model, allowing the intercept and the coefficients for the 6 ratings to vary by the rater i.
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.