MA678 Homework 07

Megha Pandit

November 1, 2018

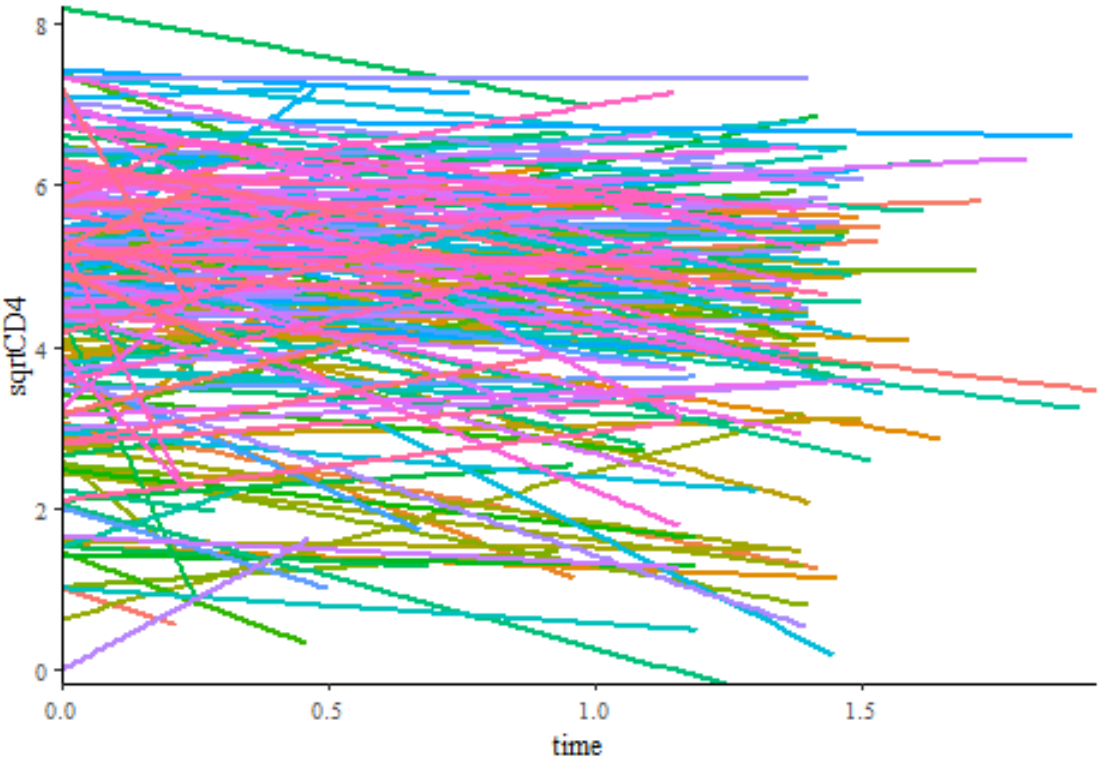
Data analysis

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(x = time, y = y, color = factor(hiv.data$newpid)))+
  theme(legend.position = "none")+
  ylab("sqrtCD4")+
  geom_smooth(method = "lm", se = FALSE)+
  scale_x_continuous(expand = c(0,0))+
  scale_y_continuous(expand = c(0,0))
```



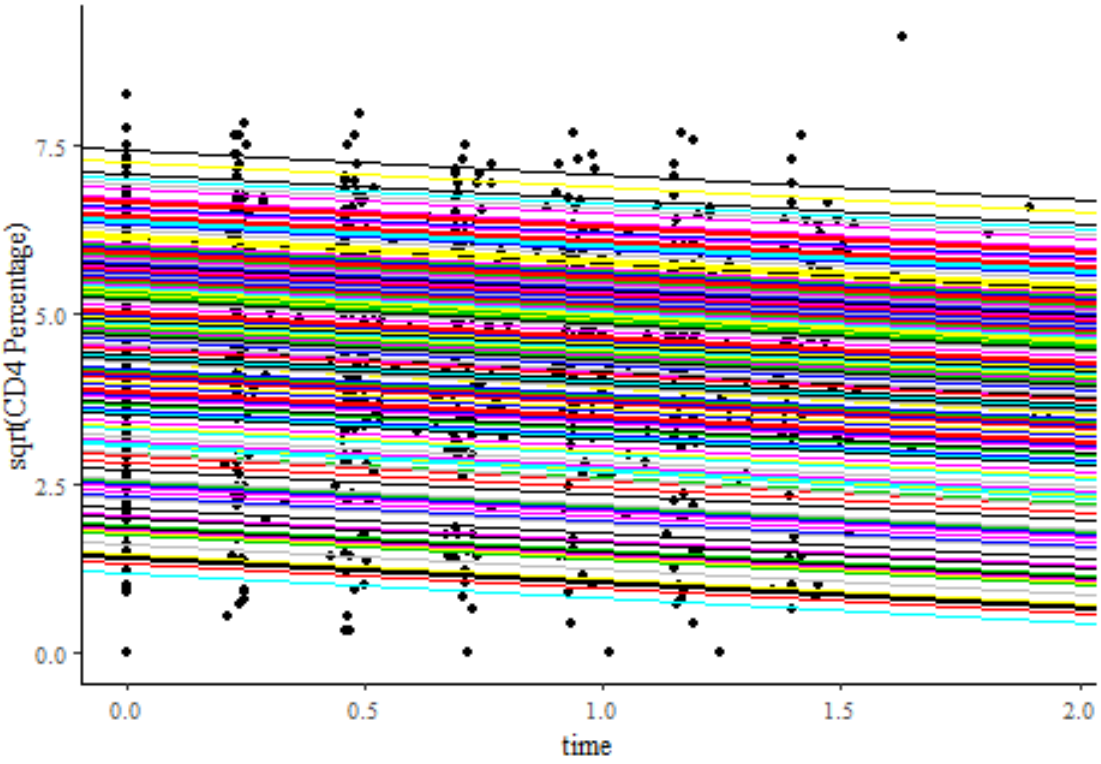
2. Each child's data has a time course that can be summarized by a linear fit. Estimate these lines and plot them for all the children.

```
fit1 <- lmer(y ~ time + (1 | newpid), data = hiv.data)
display(fit1)
```

```
## 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
```

```
coef_fit1 <- data.frame(coef(fit1)$newpid)
coef_fit1$newpid <- c(1:250)
ggplot(data=hiv.data) + geom_point(aes(x=time, y=y)) +
  geom_abline(intercept = coef_fit1$X.Intercept.,
slope= coef_fit1$time, color= coef_fit1$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.

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.

```
fit2 <- lmer(CD4PCT ~ time + (1 | newpid), data = hiv.data)
display(fit2)
```

```
## lmer(formula = CD4PCT ~ time + (1 | newpid), data = hiv.data)
##             coef.est coef.se
```

```
## (Intercept) 25.04     0.81
## time        -3.00     0.51
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 11.37
##  Residual              7.30
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 7887.2, DIC = 7882.8
## deviance = 7881.0
```

The average of time trends is -3.00 (estimated coefficient for time) with a standard error of 0.51. Thus, it can be estimated that most of the children may have declining levels of CD4 levels during this time period.

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.

```
fit3 <- lmer(CD4PCT ~ time + treatment + age.baseline + (1 | newpid), data = hiv.data)
display(fit3)
```

```
## lmer(formula = CD4PCT ~ time + treatment + age.baseline + (1 |
##     newpid), data = hiv.data)
##              coef.est coef.se
## (Intercept)  26.50     2.62
## time         -2.96     0.51
## treatment     1.21     1.51
## age.baseline -0.95     0.33
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 11.19
##  Residual              7.30
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 7880.2, DIC = 7876.3
## deviance = 7872.2
```

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

```
display(fit2)
```

```
## lmer(formula = CD4PCT ~ time + (1 | newpid), data = hiv.data)
##             coef.est coef.se
## (Intercept) 25.04     0.81
## time        -3.00     0.51
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 11.37
##  Residual              7.30
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 7887.2, DIC = 7882.8
## deviance = 7881.0
```

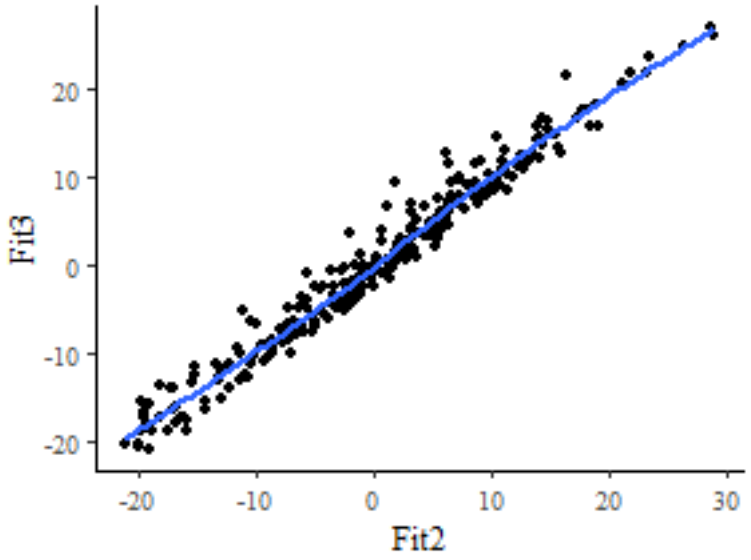
```
display(fit3)
```

```
## lmer(formula = CD4PCT ~ time + treatment + age.baseline + (1 |
##     newpid), data = hiv.data)
##              coef.est coef.se
## (Intercept)  26.50     2.62
## time         -2.96     0.51
## treatment     1.21     1.51
## age.baseline -0.95     0.33
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 11.19
##  Residual              7.30
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 7880.2, DIC = 7876.3
## deviance = 7872.2
```

```
compare <- as.data.frame(cbind(unlist(ranef(fit2)), unlist(ranef(fit3))))

ggplot(data = compare, aes(x = V1, y = V2))+
  geom_point()+
  geom_smooth(se = FALSE)+
  xlab("Fit2")+ ylab("Fit3")
```

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



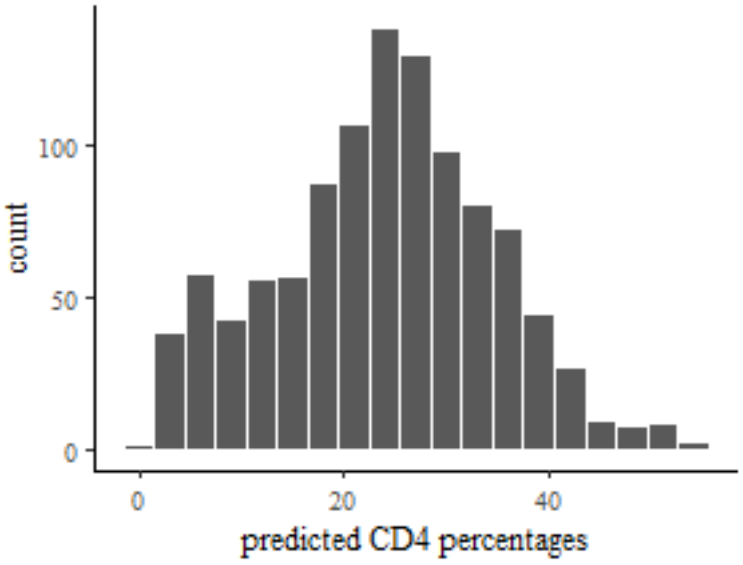
7. Use the model fit from (5) to generate simulation of predicted CD4 percentages for each child in the dataset at a hypothetical next time point.

```
new_data <- hiv.data %>%
  filter(!is.na(treatment))%>%
  filter(!is.na(age.baseline))%>%
  select(time, newpid, treatment, age.baseline)
```

```
pred <- predict(fit3, newdata = new_data)

new_pred <- cbind(pred, new_data)

ggplot(data = new_pred, aes(x = pred))+ geom_histogram(color = "white", binwidth = 3)+
  xlab("predicted CD4 percentages")
```

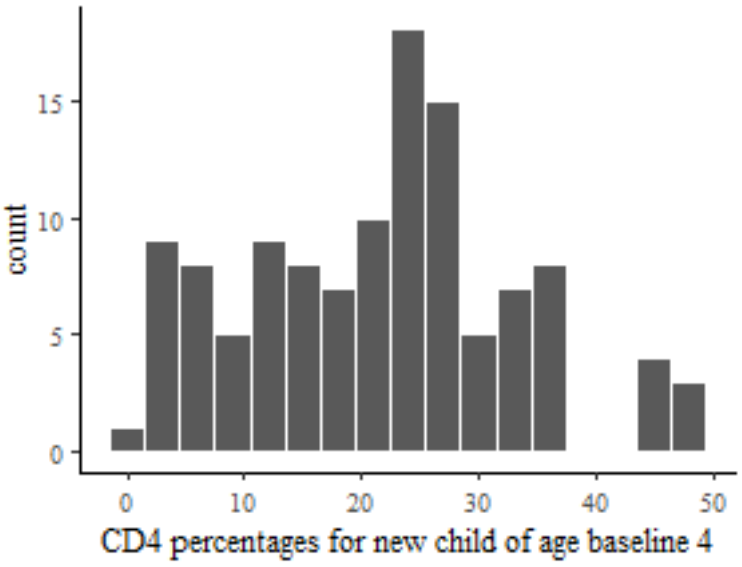


8. Use the same model fit to generate simulations of CD4 percentages at each of the time periods for a new child who was 4 years old at baseline.

```
pred_4 <- new_data%>%
  filter(round(age.baseline) == 4)

pred4 <- predict(fit3, newdata = pred_4)

new_pred4 <- cbind(pred4, pred_4)
ggplot(new_pred4, aes(pred4))+ geom_histogram(color = "white", binwidth = 3)+
  xlab("CD4 percentages for new child of age baseline 4")
```



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.

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

```
fit4 <- lmer(y ~ time + (1 + time | newpid), data = hiv.data)
display(fit4)
```

```
## lmer(formula = y ~ time + (1 + time | newpid), data = hiv.data)
##             coef.est coef.se
## (Intercept)  4.76     0.09
## time        -0.36     0.07
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  newpid   (Intercept) 1.39
##           time        0.58     -0.05
##  Residual             0.72
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3123.2, DIC = 3098.2
## deviance = 3104.7
```

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).

```
fit5 <- lmer(y ~ factor(time) + (1 | newpid), data = hiv.data)
```

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

```
compare1 <- as.data.frame(cbind(unlist(ranef(fit4)), unlist(ranef(fit5))))

ggplot(data = compare1, aes(x = V1, y = V2))+
  geom_point()+
  geom_smooth(se = FALSE)+
```

```
  xlab("Fit4 intercepts - Random effects")+ ylab("Fit5 intercepts - Random effects")
```

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

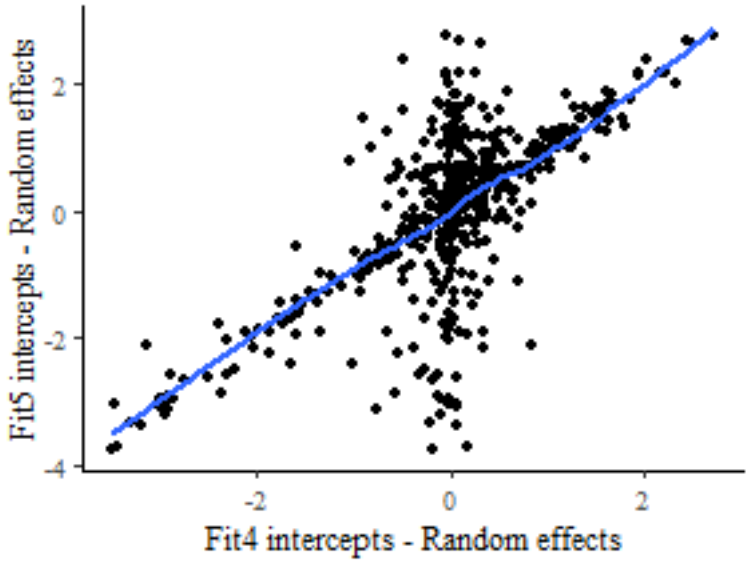


Figure skate in the 1932 Winter Olympics

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).

```
olympics <- olympics1932[,3:9]
array(unlist(lapply(split(olympics, olympics1932$criterion), function(x) as.matrix(x))), c(7,7,2))
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]  5.6  5.5  5.8  4.7  5.7  5.3  5.4
## [2,]  5.5  5.7  5.6  5.4  5.5  5.3  5.7
## [3,]  6.0  5.5  5.7  4.9  5.5  5.2  5.7
## [4,]  5.6  5.3  5.8  4.8  4.5  5.0  5.5
## [5,]  4.8  4.8  5.5  4.4  4.6  4.8  5.2
## [6,]  4.8  5.6  5.0  4.7  4.0  4.6  5.2
## [7,]  4.3  4.6  4.5  4.0  3.6  4.0  4.8
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]  5.6  5.5  5.8  5.3  5.6  5.2  5.7
## [2,]  5.5  5.2  5.8  5.8  5.6  5.1  5.8
## [3,]  6.0  5.3  5.8  5.0  5.4  5.1  5.3
## [4,]  5.6  5.3  5.8  4.4  4.5  5.0  5.1
## [5,]  5.4  4.5  5.8  4.0  5.5  4.8  5.5
## [6,]  5.2  5.1  5.3  5.4  4.5  4.5  5.0
```

```
## [7,]  4.8  4.0  4.7  4.0  3.7  4.0  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.

```
olymp <- data.frame(olympics1932[1:2],stack(olympics1932[3:9]))
```

```
## Warning in data.frame(olympics1932[1:2], stack(olympics1932[3:9])): row
## names were found from a short variable and have been discarded
```

```
olymp <- olymp[order(olymp$pair),]
olymp_split <- data.frame(split(olymp, olymp$criterion))
olymp_final <- olymp_split[,c(3,7,1,4)]
colnames(olymp_final) <- paste(c("Technical Merit Score",
                                 "Artistic Impression Score", "Skater ID", "Judge ID"))
rownames(olymp_final) <- 1:nrow(olymp_final)
olymp_final
```

```
##    Technical Merit Score Artistic Impression Score Skater ID Judge ID
## 1                    5.6                       5.6         1  judge_1
## 2                    5.5                       5.5         1  judge_2
## 3                    5.8                       5.8         1  judge_3
## 4                    4.7                       5.3         1  judge_4
## 5                    5.7                       5.6         1  judge_5
## 6                    5.3                       5.2         1  judge_6
## 7                    5.4                       5.7         1  judge_7
## 8                    5.5                       5.5         2  judge_1
## 9                    5.7                       5.2         2  judge_2
## 10                   5.6                       5.8         2  judge_3
## 11                   5.4                       5.8         2  judge_4
## 12                   5.5                       5.6         2  judge_5
## 13                   5.3                       5.1         2  judge_6
## 14                   5.7                       5.8         2  judge_7
## 15                   6.0                       6.0         3  judge_1
## 16                   5.5                       5.3         3  judge_2
## 17                   5.7                       5.8         3  judge_3
## 18                   4.9                       5.0         3  judge_4
## 19                   5.5                       5.4         3  judge_5
## 20                   5.2                       5.1         3  judge_6
## 21                   5.7                       5.3         3  judge_7
## 22                   5.6                       5.6         4  judge_1
## 23                   5.3                       5.3         4  judge_2
## 24                   5.8                       5.8         4  judge_3
## 25                   4.8                       4.4         4  judge_4
## 26                   4.5                       4.5         4  judge_5
## 27                   5.0                       5.0         4  judge_6
## 28                   5.5                       5.1         4  judge_7
## 29                   4.8                       5.4         5  judge_1
## 30                   4.8                       4.5         5  judge_2
## 31                   5.5                       5.8         5  judge_3
## 32                   4.4                       4.0         5  judge_4
## 33                   4.6                       5.5         5  judge_5
## 34                   4.8                       4.8         5  judge_6
## 35                   5.2                       5.5         5  judge_7
## 36                   4.8                       5.2         6  judge_1
```

```
## 37                    5.6                          5.1         6  judge_2
## 38                    5.0                          5.3         6  judge_3
## 39                    4.7                          5.4         6  judge_4
## 40                    4.0                          4.5         6  judge_5
## 41                    4.6                          4.5         6  judge_6
## 42                    5.2                          5.0         6  judge_7
## 43                    4.3                          4.8         7  judge_1
## 44                    4.6                          4.0         7  judge_2
## 45                    4.5                          4.7         7  judge_3
## 46                    4.0                          4.0         7  judge_4
## 47                    3.6                          3.7         7  judge_5
## 48                    4.0                          4.0         7  judge_6
## 49                    4.8                          4.8         7  judge_7
```

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.

```
skater_country <- c(rep("France",7), rep("United States", 7),
                    rep("Hungary", 7), rep("Hungary", 7), rep("Canada", 7),
                    rep("Canada",7), rep("United States", 7))
judge_country <- c(rep(c("Hungary", "Norway", "Austria","Finland","France",
                         "Great Britain","United States"), 7))
olymp_country <- cbind(olymp_final, skater_country, judge_country)

Same_Country <- c()
for (i in 1:49) {
  if (as.character(olymp_country$skater_country)[i] == as.character(olymp_country$judge_country)[i]){
    Same_Country[i] <- 1
  }
  else{
    Same_Country[i] <- 0
  }
  Same_Country
}
ind_olymp <- cbind(olymp_country, Same_Country)
ind_olymp
```

```
##    Technical Merit Score Artistic Impression Score Skater ID Judge ID
## 1                    5.6                         5.6         1  judge_1
## 2                    5.5                         5.5         1  judge_2
## 3                    5.8                         5.8         1  judge_3
## 4                    4.7                         5.3         1  judge_4
## 5                    5.7                         5.6         1  judge_5
## 6                    5.3                         5.2         1  judge_6
## 7                    5.4                         5.7         1  judge_7
## 8                    5.5                         5.5         2  judge_1
## 9                    5.7                         5.2         2  judge_2
## 10                   5.6                         5.8         2  judge_3
## 11                   5.4                         5.8         2  judge_4
## 12                   5.5                         5.6         2  judge_5
## 13                   5.3                         5.1         2  judge_6
## 14                   5.7                         5.8         2  judge_7
## 15                   6.0                         6.0         3  judge_1
## 16                   5.5                         5.3         3  judge_2
## 17                   5.7                         5.8         3  judge_3
```

```
## 18                  4.9                         5.0         3  judge_4
## 19                  5.5                         5.4         3  judge_5
## 20                  5.2                         5.1         3  judge_6
## 21                  5.7                         5.3         3  judge_7
## 22                  5.6                         5.6         4  judge_1
## 23                  5.3                         5.3         4  judge_2
## 24                  5.8                         5.8         4  judge_3
## 25                  4.8                         4.4         4  judge_4
## 26                  4.5                         4.5         4  judge_5
## 27                  5.0                         5.0         4  judge_6
## 28                  5.5                         5.1         4  judge_7
## 29                  4.8                         5.4         5  judge_1
## 30                  4.8                         4.5         5  judge_2
## 31                  5.5                         5.8         5  judge_3
## 32                  4.4                         4.0         5  judge_4
## 33                  4.6                         5.5         5  judge_5
## 34                  4.8                         4.8         5  judge_6
## 35                  5.2                         5.5         5  judge_7
## 36                  4.8                         5.2         6  judge_1
## 37                  5.6                         5.1         6  judge_2
## 38                  5.0                         5.3         6  judge_3
## 39                  4.7                         5.4         6  judge_4
## 40                  4.0                         4.5         6  judge_5
## 41                  4.6                         4.5         6  judge_6
## 42                  5.2                         5.0         6  judge_7
## 43                  4.3                         4.8         7  judge_1
## 44                  4.6                         4.0         7  judge_2
## 45                  4.5                         4.7         7  judge_3
## 46                  4.0                         4.0         7  judge_4
## 47                  3.6                         3.7         7  judge_5
## 48                  4.0                         4.0         7  judge_6
## 49                  4.8                         4.8         7  judge_7
##    skater_country judge_country Same_Country
## 1          France       Hungary            0
## 2          France        Norway            0
## 3          France       Austria            0
## 4          France       Finland            0
## 5          France        France            1
## 6          France Great Britain            0
## 7          France United States            0
## 8   United States       Hungary            0
## 9   United States        Norway            0
## 10  United States       Austria            0
## 11  United States       Finland            0
## 12  United States        France            0
## 13  United States Great Britain            0
## 14  United States United States            1
## 15        Hungary       Hungary            1
## 16        Hungary        Norway            0
## 17        Hungary       Austria            0
## 18        Hungary       Finland            0
## 19        Hungary        France            0
## 20        Hungary Great Britain            0
## 21        Hungary United States            0
```

```
## 22        Hungary       Hungary            1
## 23        Hungary        Norway            0
## 24        Hungary       Austria            0
## 25        Hungary       Finland            0
## 26        Hungary        France            0
## 27        Hungary Great Britain            0
## 28        Hungary United States            0
## 29         Canada       Hungary            0
## 30         Canada        Norway            0
## 31         Canada       Austria            0
## 32         Canada       Finland            0
## 33         Canada        France            0
## 34         Canada Great Britain            0
## 35         Canada United States            0
## 36         Canada       Hungary            0
## 37         Canada        Norway            0
## 38         Canada       Austria            0
## 39         Canada       Finland            0
## 40         Canada        France            0
## 41         Canada Great Britain            0
## 42         Canada United States            0
## 43  United States       Hungary            0
## 44  United States        Norway            0
## 45  United States       Austria            0
## 46  United States       Finland            0
## 47  United States        France            0
## 48  United States Great Britain            0
## 49  United States United States            1
```

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 <- ind_olymp%>%
  select(`Technical Merit Score`, `Skater ID`, `Judge ID`)
colnames(data) <- paste(c("score", "Skater_ID", "Judge_ID"))

fit6 <- lmer(score ~ 1 + (1 | Skater_ID) + (1 | Judge_ID), data = data)
display(fit6)
```

```
## lmer(formula = score ~ 1 + (1 | Skater_ID) + (1 | Judge_ID),
##     data = data)
## 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
```

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

```
data1 <- ind_olymp%>%
  select(`Artistic Impression Score`, `Skater ID`, `Judge ID`)
colnames(data1) <- paste(c("score", "Skater_ID", "Judge_ID"))

fit7 <- lmer(score ~ 1 + (1 | Skater_ID) + (1 | Judge_ID), data = data1)
display(fit7)
```

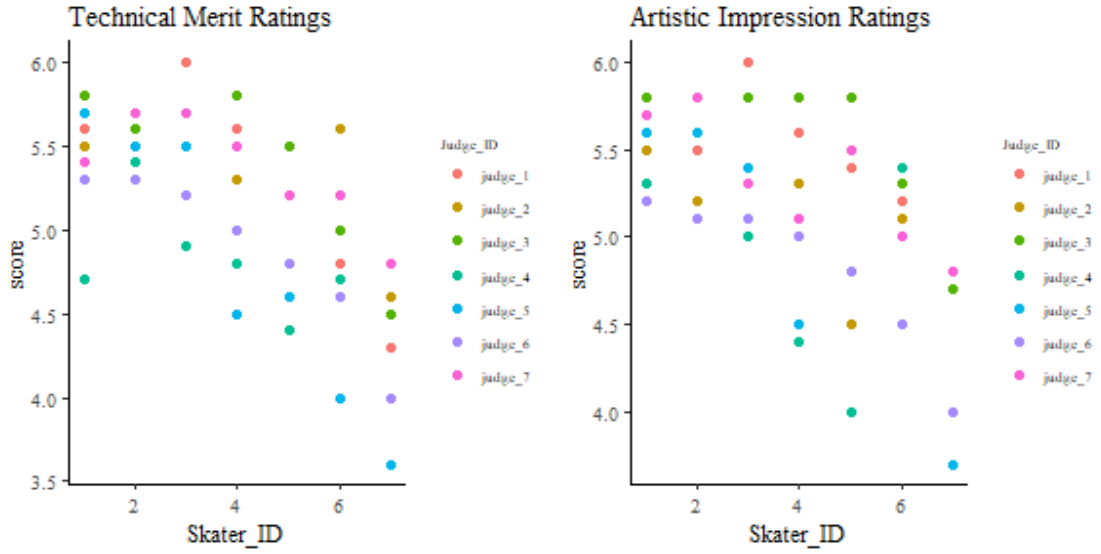
```
## lmer(formula = score ~ 1 + (1 | Skater_ID) + (1 | Judge_ID),
##     data = data1)
## coef.est  coef.se
##     5.13     0.20
##
## Error terms:
##  Groups    Name        Std.Dev.
##  Skater_ID (Intercept) 0.42
##  Judge_ID  (Intercept) 0.28
##  Residual              0.33
## ---
## number of obs: 49, groups: Skater_ID, 7; Judge_ID, 7
## AIC = 68, DIC = 57
## deviance = 58.5
```

6. Display your results for both outcomes graphically.

```
g1 <- ggplot(data = data, aes(x = Skater_ID, y = score, color = Judge_ID))+ geom_point(size = 2)+
  ggtitle("Technical Merit Ratings")+
  theme(legend.text = element_text(size = 7), legend.title = element_text(size = 7))

g2 <- ggplot(data = data1, aes(x = Skater_ID, y = score, color = Judge_ID))+ geom_point(size = 2)+
  ggtitle("Artistic Impression Ratings")+
   theme(legend.text = element_text(size = 7), legend.title = element_text(size = 7))

grid.arrange(g1, g2, ncol = 2)
```



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.

Model1: Regression coefficients varying across groups

$y = 4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * 0.18 + 0.77$ $(i = 1, ..., n_{250})$ $\alpha_j \sim \mathrm{N}(0, 1.37^2)$

Combining local regressions

$y_i \sim \mathrm{N}(4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * 0.18, 0.77^2)$ $(i = 1, ..., n_{250})$ $\alpha_j \sim \mathrm{N}(randomintercept, 1.37^2)$

Modeling coefficients of a large regression model

$y_i \sim \mathrm{N}(4.91 + time_i * (-0.36) + treatment_i * (-0.12) + age.baseline_i * 0.18, 0.77^2)$ $\beta_j \sim \mathrm{N}(0, 1.37^2)(j = 3, ..., J+2)$

Large regression with correlated errors

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

Regression with multiple errors

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

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).
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.