

MA678 Homework 7

JingjianGao

November 15, 2022

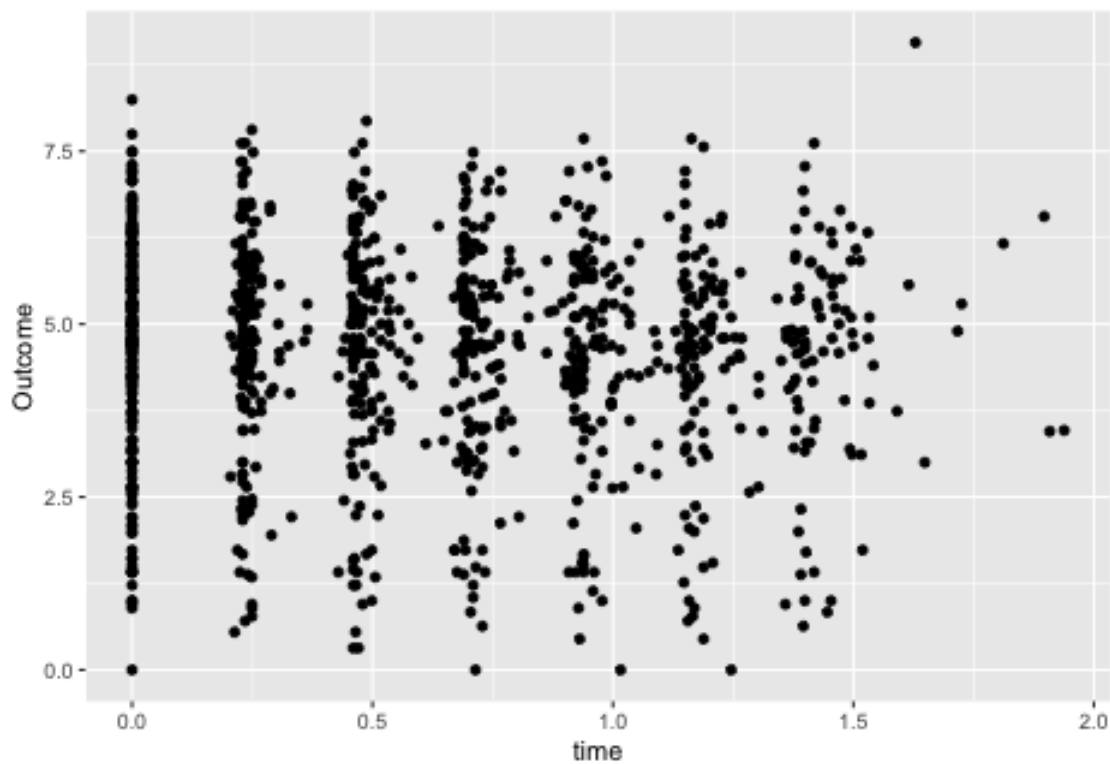
Data analysis

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.

```
reg1 <- glm(CD4PCT ~ time, data = hiv.data)
ggplot(data=hiv.data, aes(x=time,y=y))+geom_point()+ylab("Outcome")
```

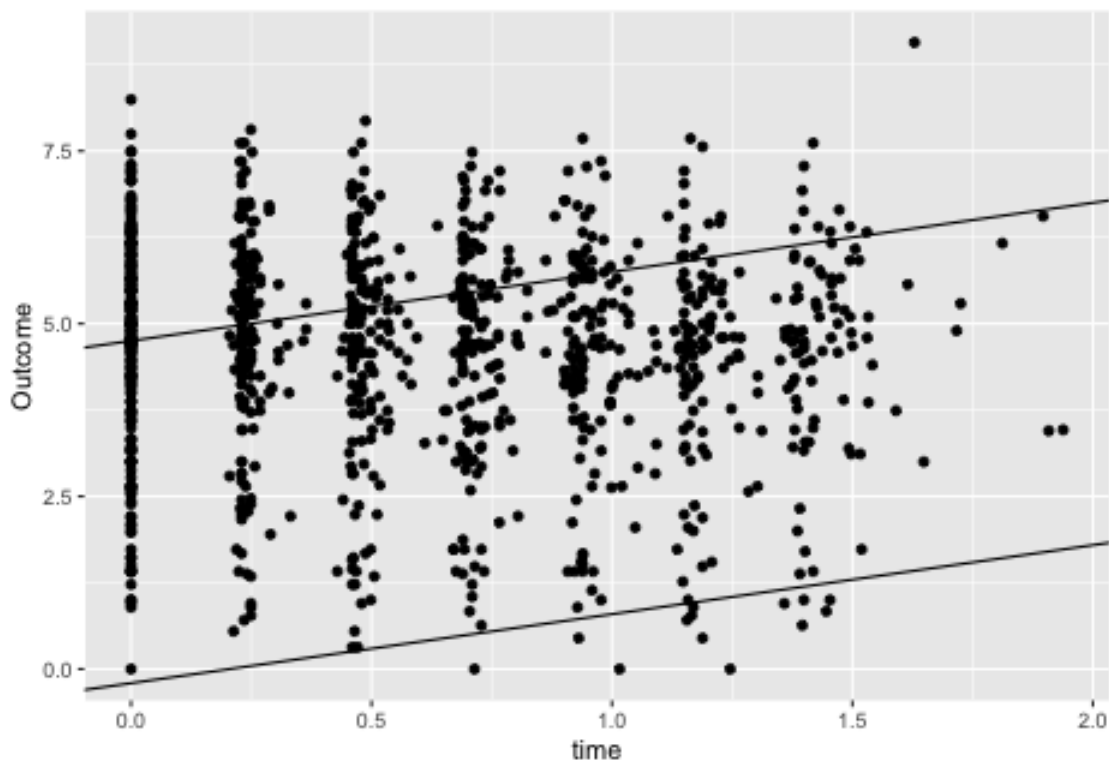


2. Each child's data has a time course that can be summarized by a linear fit. Estimate these lines and plot them for all the children.

```
fit1 <- lm(y~time,data=hiv.data)
summary(fit1)
```

##

```
## Call:
## lm(formula = y ~ time, data = hiv.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.7519 -0.8116  0.2131  1.0624  4.6535
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.75188    0.07575  62.734  <2e-16 ***
## time        -0.20464    0.10014  -2.044   0.0412 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.563 on 1070 degrees of freedom
## Multiple R-squared:  0.003888, Adjusted R-squared:  0.002957
## F-statistic: 4.176 on 1 and 1070 DF, p-value: 0.04123
ggplot(data=hiv.data, aes(x=time,y=y))+geom_point()+ylab("Outcome")+
  geom_abline(intercept=coef(fit1[1]))
```



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.

#Step 1

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.

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

```
## lmer(formula = y ~ time + (1 | treatment), data = hiv.data)
##           coef.est coef.se
## (Intercept)  4.76    0.17
## time        -0.20    0.10
##
## Error terms:
## Groups      Name          Std.Dev.
## treatment (Intercept) 0.22
## Residual              1.55
## ---
## number of obs: 1072, groups: treatment, 2
## AIC = 4004, DIC = 3986.2
## deviance = 3991.1
```

#We can see that the coefficient for time is -0.2, this means whenever there is an increase of 1 unit of time, the mean CD4 value drops by 0.2

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.

```
fit5 <- lmer(y~time+treatment+age.baseline+(1|newpid),data=hiv.data)
display(fit5)
```

```
## 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
```

#The coefficient of time is -0.36, meaning that for each unit increase in time, CD4PCT will drop by 0.36

#The coefficient of treatment is 0.18, meaning that for each unit increase in treatment, CD4PCT will increase by 0.18

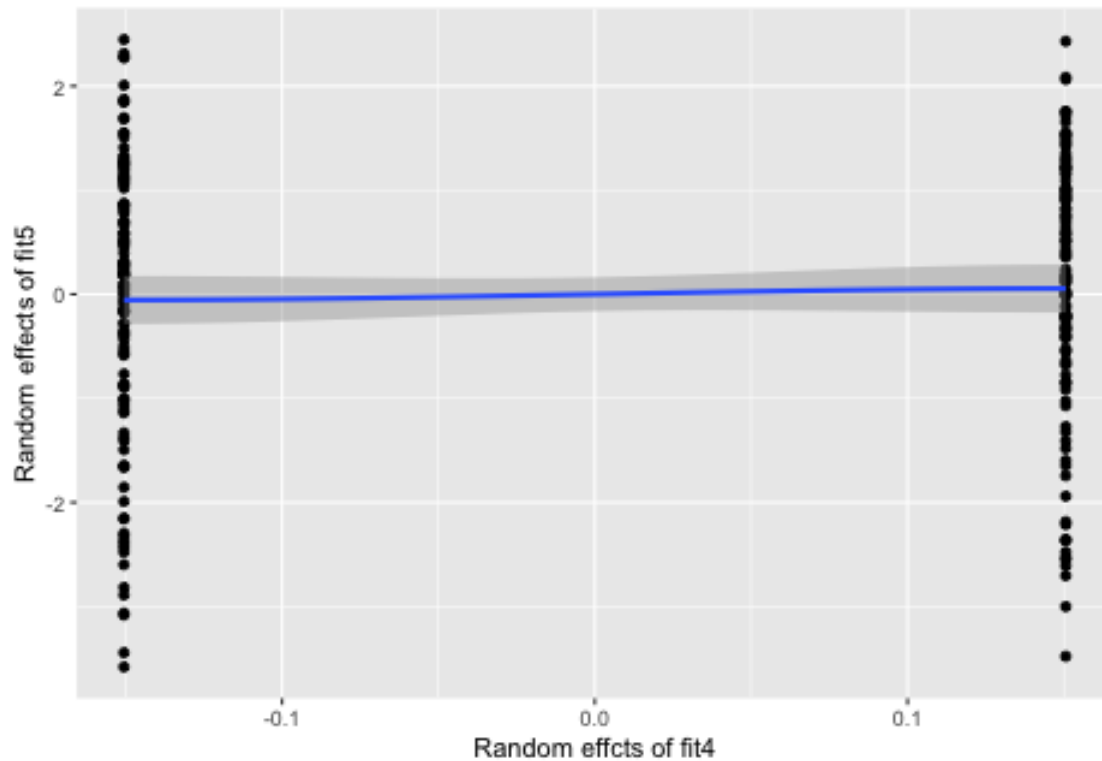
#The coefficient of age.baseline is -0.12, meaning that for each unit increase in age.baseline, CD4PCT will decrease by 0.12

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

```
plot6 <- as.data.frame(cbind(unlist(ranef(fit4)),unlist(ranef(fit5))))
colnames(plot6) <- c("fit4","fit5")
ggplot(data=plot6,aes(x=fit4,y=fit5))+geom_point()+geom_smooth()+
  xlab("Random effects of fit4")+
  ylab("Random effects of fit5")
```

```
ylab("Random effects of fit5")

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.15208
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.30266
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.091602
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : pseudoinverse used at
## -0.15208
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : neighborhood radius
## 0.30266
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : reciprocal condition
## number 0
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object))), : There are other near
## singularities as well. 0.091602
```



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)
```

```
##
## 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
predict_data <- hiv.data %>%
  filter(is.na(hiv.data$treatment)==FALSE) %>%
  filter(is.na(age.baseline)==FALSE) %>%
  select(time,treatment,age.baseline,newpid,y)
new_predict <- predict(fit5,newdata=predict_data)
predict_cmb <- cbind(predict_data, new_predict)
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.

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 model to allow for varying slopes for the time predictor.

```
fit10 <- lmer(y~time+(time|newpid), data=hiv.data)
```

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).

```
fit11 <- lmer(y~factor(time)+(1|newpid), data=hiv.data)
```

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

```
display(fit10)
```

```
## lmer(formula = y ~ time + (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
```

```
display(fit11)
```

```
## lmer(formula = y ~ factor(time) + (1 | newpid), data = hiv.data)
##           coef.est coef.se
## (Intercept)          4.77    0.10
## factor(time)0.205      -1.22    0.69
## factor(time)0.21        0.27    0.88
## factor(time)0.21333333333333333 -0.97    0.66
## factor(time)0.21583333333333331  1.66    0.97
## factor(time)0.21583333333333333 -0.31    0.87
## factor(time)0.21666666666666667 -0.18    0.83
```

```

## factor(time)0.218333333333335 0.22 0.97
## factor(time)0.219166666666667 -0.67 0.91
## factor(time)0.221666666666668 0.20 0.97
## factor(time)0.224166666666667 -0.58 0.54
## factor(time)0.226666666666667 1.40 0.60
## factor(time)0.227499999999999 -1.50 0.91
## factor(time)0.2275 0.19 0.48
## factor(time)0.229999999999999 -0.54 0.37
## factor(time)0.23 -0.07 0.14
## factor(time)0.230000000000001 0.21 0.37
## factor(time)0.2325 -0.57 0.41
## factor(time)0.233333333333333 -0.07 0.87
## factor(time)0.235 -2.41 0.84
## factor(time)0.235833333333333 0.00 0.31
## factor(time)0.235833333333334 0.73 0.88
## factor(time)0.235833333333335 -0.26 0.84
## factor(time)0.2375 1.72 0.97
## factor(time)0.238333333333333 -0.05 0.61
## factor(time)0.238333333333334 0.05 0.60
## factor(time)0.240833333333333 0.28 0.61
## factor(time)0.240833333333335 -0.80 0.90
## factor(time)0.243333333333333 -0.75 0.97
## factor(time)0.244166666666667 0.07 0.49
## factor(time)0.245833333333333 -0.20 0.62
## factor(time)0.245833333333334 0.10 0.44
## factor(time)0.246666666666666 -0.86 0.88
## factor(time)0.246666666666667 0.28 0.65
## factor(time)0.249166666666666 -0.54 0.69
## factor(time)0.249166666666667 -0.36 0.19
## factor(time)0.249166666666668 -0.14 0.58
## factor(time)0.251666666666667 0.46 0.45
## factor(time)0.251666666666669 0.25 0.83
## factor(time)0.2525 -0.07 0.97
## factor(time)0.254166666666667 -0.56 0.87
## factor(time)0.255 0.31 0.84
## factor(time)0.256666666666667 0.28 0.67
## factor(time)0.257499999999999 0.07 0.87
## factor(time)0.2575 0.45 0.65
## factor(time)0.262500000000001 -0.04 0.87
## factor(time)0.265 -0.14 0.90
## factor(time)0.265833333333335 -0.33 0.86
## factor(time)0.268333333333333 -0.31 0.61
## factor(time)0.268333333333334 0.09 0.51
## factor(time)0.2875 0.89 0.67
## factor(time)0.287500000000001 -0.45 0.97
## factor(time)0.29 -0.81 0.97
## factor(time)0.293333333333333 -0.29 0.97
## factor(time)0.304166666666667 0.38 0.97
## factor(time)0.306666666666665 0.01 0.86
## factor(time)0.306666666666667 0.12 0.64
## factor(time)0.325833333333334 -0.18 0.97
## factor(time)0.328333333333333 -0.87 0.97
## factor(time)0.331666666666667 0.00 0.97
## factor(time)0.358333333333333 0.57 0.88

```

```

## factor(time)0.364166666666666 0.01 0.63
## factor(time)0.429166666666666 -1.26 0.88
## factor(time)0.429166666666668 -0.72 0.97
## factor(time)0.438333333333333 -0.84 0.92
## factor(time)0.440833333333334 -0.16 0.80
## factor(time)0.443333333333334 0.17 0.87
## factor(time)0.449166666666668 0.04 0.83
## factor(time)0.454166666666664 0.46 0.97
## factor(time)0.454166666666667 -0.91 0.90
## factor(time)0.455 -1.39 0.91
## factor(time)0.456666666666668 0.25 0.97
## factor(time)0.4575 0.33 0.65
## factor(time)0.457500000000001 -0.73 0.80
## factor(time)0.459166666666667 0.32 0.62
## factor(time)0.459999999999999 0.47 0.82
## factor(time)0.46 -0.27 0.17
## factor(time)0.460000000000001 -0.17 0.38
## factor(time)0.460000000000002 -0.36 0.47
## factor(time)0.462499999999999 -0.32 0.85
## factor(time)0.4625 -0.02 0.35
## factor(time)0.463333333333333 -0.41 0.82
## factor(time)0.465 -1.04 0.50
## factor(time)0.465833333333333 -0.03 0.63
## factor(time)0.465833333333334 -0.95 0.90
## factor(time)0.467499999999999 1.14 0.82
## factor(time)0.4675 2.28 0.97
## factor(time)0.468333333333335 -0.99 0.91
## factor(time)0.470833333333332 -1.38 0.92
## factor(time)0.470833333333333 0.02 0.83
## factor(time)0.470833333333334 -0.33 0.50
## factor(time)0.473333333333332 -0.12 0.86
## factor(time)0.473333333333333 -0.66 0.97
## factor(time)0.473333333333334 0.49 0.82
## factor(time)0.474166666666665 -0.65 0.89
## factor(time)0.474166666666667 -0.82 0.83
## factor(time)0.475833333333333 -0.52 0.84
## factor(time)0.475833333333334 -1.23 0.59
## factor(time)0.476666666666667 0.68 0.42
## factor(time)0.479166666666666 -0.46 0.59
## factor(time)0.479166666666667 0.03 0.28
## factor(time)0.479166666666668 -0.15 0.60
## factor(time)0.481666666666666 0.12 0.61
## factor(time)0.481666666666667 0.41 0.51
## factor(time)0.481666666666669 0.15 1.00
## factor(time)0.484166666666667 -2.33 0.84
## factor(time)0.485 1.66 0.89
## factor(time)0.485000000000001 0.32 0.88
## factor(time)0.487499999999999 -0.29 0.79
## factor(time)0.4875 1.81 0.85
## factor(time)0.487500000000002 1.71 0.83
## factor(time)0.49 0.11 0.86
## factor(time)0.495 -0.09 0.61
## factor(time)0.495833333333333 -0.39 0.67
## factor(time)0.498333333333333 -0.59 0.40

```



```

## factor(time)0.498333333333334 -0.53      0.39
## factor(time)0.500833333333334 -0.14      0.59
## factor(time)0.501666666666667 -1.19      0.64
## factor(time)0.503333333333336  0.63      0.86
## factor(time)0.504166666666666  0.23      0.53
## factor(time)0.505833333333334 -1.66      0.84
## factor(time)0.509166666666666 -0.27      0.97
## factor(time)0.511666666666668 -1.11      0.90
## factor(time)0.514166666666666  0.11      0.63
## factor(time)0.515              0.18      0.97
## factor(time)0.5175             -0.39      0.40
## factor(time)0.517500000000001 -1.13      0.85
## factor(time)0.533333333333333 -0.98      0.97
## factor(time)0.533333333333334 -0.53      0.66
## factor(time)0.534166666666666  0.12      0.84
## factor(time)0.534166666666667  0.38      0.97
## factor(time)0.536666666666665 -0.66      0.86
## factor(time)0.536666666666667 -0.42      0.52
## factor(time)0.555833333333333 -0.29      0.84
## factor(time)0.555833333333334  0.03      0.91
## factor(time)0.558333333333334  1.73      0.97
## factor(time)0.564166666666667 -0.72      0.92
## factor(time)0.575              -0.40      0.97
## factor(time)0.575000000000001 -0.20      0.95
## factor(time)0.580833333333333 -0.26      0.97
## factor(time)0.5825              0.61      0.97
## factor(time)0.594166666666666 -0.31      0.97
## factor(time)0.610833333333333 -1.09      0.97
## factor(time)0.6375              1.80      0.92
## factor(time)0.648333333333333 -2.07      0.91
## factor(time)0.651666666666666 -1.28      0.90
## factor(time)0.6575              -1.70      0.91
## factor(time)0.67                -0.27      0.91
## factor(time)0.670833333333332 -0.50      0.89
## factor(time)0.670833333333333 -1.34      0.62
## factor(time)0.673333333333333 -0.92      0.88
## factor(time)0.675833333333332  0.33      0.97
## factor(time)0.675833333333334 -0.72      0.97
## factor(time)0.684166666666666 -0.43      0.90
## factor(time)0.684166666666667  0.48      0.92
## factor(time)0.685              0.77      0.88
## factor(time)0.6875             -1.47      0.63
## factor(time)0.689166666666667  0.07      0.62
## factor(time)0.689999999999999  0.76      0.81
## factor(time)0.69                -0.15      0.18
## factor(time)0.690000000000001 -0.37      0.48
## factor(time)0.6925              0.61      0.47
## factor(time)0.693333333333334 -0.67      0.91
## factor(time)0.695              0.31      0.87
## factor(time)0.695833333333333  0.18      0.61
## factor(time)0.695833333333335 -2.00      0.88
## factor(time)0.695833333333336  0.71      0.90
## factor(time)0.697499999999999 -1.57      0.82
## factor(time)0.698333333333332 -0.08      0.85

```

```

## factor(time)0.6983333333333333 -0.58      0.61
## factor(time)0.7008333333333332  1.74      0.92
## factor(time)0.7008333333333333  2.90      0.97
## factor(time)0.7008333333333334  0.16      0.82
## factor(time)0.7033333333333332 -1.90      0.86
## factor(time)0.7033333333333334 -0.67      0.58
## factor(time)0.7041666666666667 -4.77      0.83
## factor(time)0.7058333333333333 -0.43      0.50
## factor(time)0.7058333333333334 -0.43      0.80
## factor(time)0.7066666666666667  2.52      0.83
## factor(time)0.7091666666666666 -0.11      0.82
## factor(time)0.7091666666666667  0.36      0.31
## factor(time)0.7091666666666668 -0.77      0.84
## factor(time)0.7116666666666666  0.02      0.88
## factor(time)0.7116666666666667 -0.52      0.83
## factor(time)0.7116666666666669 -0.87      0.59
## factor(time)0.7141666666666666 -3.11      0.84
## factor(time)0.7141666666666667  0.11      0.85
## factor(time)0.7150000000000001 -0.92      0.63
## factor(time)0.7175                -0.63      0.84
## factor(time)0.72                  -4.27      0.80
## factor(time)0.725                 -0.62      0.61
## factor(time)0.7258333333333332  0.33      0.87
## factor(time)0.7258333333333333  0.71      0.85
## factor(time)0.7258333333333334  0.35      0.97
## factor(time)0.7283333333333332 -0.41      0.82
## factor(time)0.7283333333333333 -0.27      0.32
## factor(time)0.7308333333333333  0.09      0.53
## factor(time)0.7333333333333333 -0.76      0.85
## factor(time)0.7341666666666666 -2.61      0.82
## factor(time)0.7358333333333333 -1.23      0.97
## factor(time)0.7358333333333334 -0.62      0.97
## factor(time)0.7366666666666667  1.81      0.89
## factor(time)0.7366666666666668  0.06      0.87
## factor(time)0.7425                -0.43      0.97
## factor(time)0.7441666666666667  0.18      0.59
## factor(time)0.7450000000000001 -0.38      0.85
## factor(time)0.7475                0.04      0.48
## factor(time)0.7475000000000001 -0.84      0.58
## factor(time)0.7525000000000003 -1.62      0.86
## factor(time)0.7583333333333334  0.22      0.88
## factor(time)0.7616666666666667 -0.58      0.83
## factor(time)0.7633333333333333 -0.26      0.85
## factor(time)0.7633333333333334  0.18      0.87
## factor(time)0.7641666666666666 -0.51      0.88
## factor(time)0.7658333333333333 -1.22      0.90
## factor(time)0.7666666666666666 -0.10      0.84
## factor(time)0.7666666666666667 -0.73      0.51
## factor(time)0.7666666666666668  0.19      0.89
## factor(time)0.775                 -1.56      0.83
## factor(time)0.78                  0.89      1.57
## factor(time)0.7833333333333333  1.12      0.97
## factor(time)0.785                 -0.16      0.85
## factor(time)0.7858333333333331  0.36      0.86

```

```

## factor(time)0.7883333333333333 -0.75      0.97
## factor(time)0.7941666666666667 -0.65      0.84
## factor(time)0.8025                -0.58      0.60
## factor(time)0.805                 -1.58      0.85
## factor(time)0.8050000000000001  0.20      0.82
## factor(time)0.8075000000000001 -0.27      0.92
## factor(time)0.8241666666666666  0.37      0.97
## factor(time)0.8241666666666667 -0.10      0.95
## factor(time)0.8625                -0.34      0.63
## factor(time)0.8675                0.56      0.92
## factor(time)0.8783333333333333 -0.19      0.91
## factor(time)0.8816666666666666  0.96      0.97
## factor(time)0.8958333333333333 -0.66      0.91
## factor(time)0.9008333333333332  0.01      0.90
## factor(time)0.9008333333333333 -0.44      0.87
## factor(time)0.9008333333333334  0.57      0.80
## factor(time)0.9033333333333333  0.22      0.60
## factor(time)0.9058333333333332  1.45      0.97
## factor(time)0.9083333333333335 -0.72      0.97
## factor(time)0.9091666666666666  0.88      0.63
## factor(time)0.9116666666666667 -0.31      0.81
## factor(time)0.9141666666666667 -0.53      0.90
## factor(time)0.9141666666666668  0.69      0.80
## factor(time)0.9175                -0.33      0.88
## factor(time)0.9191666666666667  0.08      0.49
## factor(time)0.9199999999999998 -0.77      0.82
## factor(time)0.92                 -0.86      0.33
## factor(time)0.9200000000000001 -0.60      0.41
## factor(time)0.9200000000000002 -0.62      0.59
## factor(time)0.9225                -0.20      0.81
## factor(time)0.9225000000000002 -0.99      0.91
## factor(time)0.9258333333333332 -1.30      0.83
## factor(time)0.9258333333333333 -2.55      0.88
## factor(time)0.9258333333333334  0.24      0.82
## factor(time)0.9283333333333333  0.07      0.81
## factor(time)0.9283333333333335 -1.28      0.90
## factor(time)0.9308333333333332 -1.24      0.92
## factor(time)0.9308333333333333 -0.53      0.80
## factor(time)0.9308333333333334 -2.17      0.86
## factor(time)0.9333333333333332  0.35      0.84
## factor(time)0.9333333333333333 -0.15      0.82
## factor(time)0.9333333333333334  0.03      0.90
## factor(time)0.9341666666666664 -0.95      0.85
## factor(time)0.9341666666666667 -1.04      0.83
## factor(time)0.9358333333333333 -0.59      0.61
## factor(time)0.9358333333333334 -0.45      0.84
## factor(time)0.9366666666666667  0.15      0.81
## factor(time)0.9383333333333334  0.34      0.85
## factor(time)0.9391666666666666 -0.13      0.38
## factor(time)0.9391666666666667 -0.01      0.37
## factor(time)0.9391666666666668 -1.26      0.60
## factor(time)0.9391666666666669 -0.07      1.00
## factor(time)0.9416666666666666 -0.30      0.59
## factor(time)0.9416666666666667 -0.29      0.48

```

## factor(time)0.944166666666667	0.05	0.61
## factor(time)0.9475	1.97	0.84
## factor(time)0.9475000000000002	-0.51	0.82
## factor(time)0.9525000000000001	-0.41	0.82
## factor(time)0.955	1.48	0.88
## factor(time)0.9550000000000001	-0.48	0.82
## factor(time)0.9558333333333333	-1.04	0.87
## factor(time)0.9575	0.37	0.82
## factor(time)0.9583333333333333	0.41	0.48
## factor(time)0.9583333333333334	-0.23	0.38
## factor(time)0.9608333333333334	-1.47	0.84
## factor(time)0.9641666666666665	-0.04	0.87
## factor(time)0.9641666666666667	0.49	0.85
## factor(time)0.9658333333333335	0.06	0.87
## factor(time)0.9766666666666667	-0.45	0.87
## factor(time)0.9774999999999999	-3.03	0.82
## factor(time)0.9775	-0.45	0.41
## factor(time)0.9775000000000001	0.25	0.62
## factor(time)0.9825	-0.18	0.62
## factor(time)0.9833333333333334	0.16	0.92
## factor(time)0.9858333333333333	-0.36	0.97
## factor(time)0.9966666666666666	-0.55	0.58
## factor(time)0.9966666666666667	-0.85	0.43
## factor(time)0.9991666666666666	-1.01	0.82
## factor(time)1	-1.08	0.87
## factor(time)1.0016666666666667	-0.33	0.86
## factor(time)1.0025	-1.00	0.89
## factor(time)1.0108333333333333	-0.31	0.83
## factor(time)1.0125	-0.88	0.85
## factor(time)1.0158333333333333	-0.95	0.63
## factor(time)1.0208333333333333	-1.80	0.97
## factor(time)1.0233333333333333	-0.42	0.97
## factor(time)1.0325	0.85	0.97
## factor(time)1.035	-0.37	0.41
## factor(time)1.0483333333333333	-1.29	0.90
## factor(time)1.0533333333333333	-0.07	0.85
## factor(time)1.0541666666666667	-0.61	0.60
## factor(time)1.0758333333333333	-1.00	0.87
## factor(time)1.0866666666666667	0.13	1.57
## factor(time)1.09	-1.54	0.97
## factor(time)1.0925	-0.64	0.52
## factor(time)1.1141666666666667	0.04	0.90
## factor(time)1.1166666666666667	0.96	0.97
## factor(time)1.1308333333333333	0.34	0.87
## factor(time)1.1358333333333333	-3.77	0.88
## factor(time)1.1391666666666667	0.13	0.81
## factor(time)1.1416666666666667	-0.36	0.59
## factor(time)1.1441666666666667	0.06	0.90
## factor(time)1.145	-0.36	0.91
## factor(time)1.1475	-1.19	0.88
## factor(time)1.1491666666666667	-0.64	0.59
## factor(time)1.15	-0.35	0.23
## factor(time)1.1525	-0.90	0.48
## factor(time)1.1558333333333333	-1.32	0.51

## factor(time)1.1575	-0.14	0.48
## factor(time)1.15833333333333	-1.73	0.63
## factor(time)1.16083333333333	-1.08	0.42
## factor(time)1.16333333333333	0.70	0.60
## factor(time)1.16416666666667	0.19	0.85
## factor(time)1.16583333333333	-1.65	0.86
## factor(time)1.16833333333333	-0.01	0.85
## factor(time)1.16916666666667	-0.53	0.32
## factor(time)1.17166666666667	-1.39	0.59
## factor(time)1.17666666666667	-1.02	0.84
## factor(time)1.1775	-0.16	0.84
## factor(time)1.18	-0.45	0.82
## factor(time)1.18833333333333	-0.02	0.24
## factor(time)1.18833333333334	-0.33	0.84
## factor(time)1.19666666666667	0.31	0.65
## factor(time)1.20166666666667	0.04	0.50
## factor(time)1.20416666666667	-0.61	0.60
## factor(time)1.20666666666667	0.28	0.82
## factor(time)1.2075	-0.92	0.48
## factor(time)1.2125	0.08	0.85
## factor(time)1.22416666666667	-0.42	0.88
## factor(time)1.22583333333333	0.02	0.89
## factor(time)1.22666666666667	0.01	0.61
## factor(time)1.22916666666667	-0.14	0.61
## factor(time)1.23166666666667	-1.25	0.83
## factor(time)1.2325	0.03	0.83
## factor(time)1.24583333333333	-1.01	0.52
## factor(time)1.24833333333333	-1.35	0.89
## factor(time)1.25333333333333	-0.84	0.97
## factor(time)1.26166666666667	-0.55	0.85
## factor(time)1.265	-0.20	0.49
## factor(time)1.2675	-0.34	0.87
## factor(time)1.28416666666667	-2.36	0.88
## factor(time)1.3025	-0.70	0.90
## factor(time)1.30333333333333	-0.32	0.59
## factor(time)1.31166666666667	-1.85	0.83
## factor(time)1.34166666666667	-1.09	0.81
## factor(time)1.35	-0.55	0.90
## factor(time)1.35833333333333	-1.50	0.88
## factor(time)1.36	-0.03	0.88
## factor(time)1.36083333333333	-0.08	0.97
## factor(time)1.36583333333333	-0.80	0.81
## factor(time)1.37166666666667	-0.16	0.58
## factor(time)1.37416666666667	-0.52	0.82
## factor(time)1.375	-0.83	0.82
## factor(time)1.37666666666667	-3.01	0.79
## factor(time)1.37916666666667	0.10	0.80
## factor(time)1.38	-0.39	0.34
## factor(time)1.3825	-1.53	0.57
## factor(time)1.38583333333333	-0.97	0.62
## factor(time)1.38583333333334	-0.17	0.90
## factor(time)1.3875	-0.26	0.59
## factor(time)1.38833333333333	-1.52	0.83
## factor(time)1.39083333333333	-1.00	0.61

```

## factor(time)1.3958333333333333 -0.73      0.61
## factor(time)1.3966666666666667  0.30      0.80
## factor(time)1.3983333333333333 -0.11      0.89
## factor(time)1.3991666666666667 -0.47      0.32
## factor(time)1.4016666666666667 -2.18      0.62
## factor(time)1.41                -1.28      0.57
## factor(time)1.4125               -0.40      0.84
## factor(time)1.415                -0.21      0.60
## factor(time)1.4183333333333333 -0.18      0.47
## factor(time)1.4208333333333333 -0.75      0.97
## factor(time)1.4241666666666667  0.50      0.81
## factor(time)1.4258333333333333 -0.19      0.82
## factor(time)1.4291666666666667  0.71      0.90
## factor(time)1.4316666666666667 -0.13      0.60
## factor(time)1.4366666666666667  0.49      0.83
## factor(time)1.4375               -0.79      0.81
## factor(time)1.4458333333333333 -3.19      0.82
## factor(time)1.4533333333333333 -1.14      0.97
## factor(time)1.4541666666666667 -0.56      0.88
## factor(time)1.4558333333333333 -0.58      0.82
## factor(time)1.4566666666666667 -0.08      0.43
## factor(time)1.4625               0.47      0.83
## factor(time)1.47                 0.09      0.61
## factor(time)1.4725               0.18      0.83
## factor(time)1.475                0.25      0.85
## factor(time)1.4816666666666667 -1.22      0.89
## factor(time)1.4833333333333333 -0.45      0.97
## factor(time)1.4925               -2.42      0.83
## factor(time)1.495                0.22      0.50
## factor(time)1.4975               0.41      0.65
## factor(time)1.5                  -0.43      0.85
## factor(time)1.5058333333333333 -0.47      0.92
## factor(time)1.5141666666666667  0.37      0.82
## factor(time)1.5166666666666667 -2.02      0.87
## factor(time)1.5191666666666667 -3.19      0.88
## factor(time)1.53                 0.52      0.90
## factor(time)1.5308333333333333  0.17      0.97
## factor(time)1.5333333333333333 -0.22      0.61
## factor(time)1.5416666666666667 -0.89      0.83
## factor(time)1.5908333333333333 -1.28      0.90
## factor(time)1.615                -0.66      0.85
## factor(time)1.6291666666666667  3.58      0.84
## factor(time)1.6483333333333333 -1.51      0.87
## factor(time)1.7166666666666667  0.02      0.87
## factor(time)1.725                -0.25      0.82
## factor(time)1.8116666666666667  0.36      0.90
## factor(time)1.8966666666666667 -0.18      0.89
## factor(time)1.9083333333333333 -0.74      0.88
## factor(time)1.9383333333333333 -0.89      0.97
##
## Error terms:
## Groups   Name      Std.Dev.
## newpid   (Intercept) 1.38
## Residual                      0.73

```

```
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 2976.8, DIC = 2727.5
## deviance = 2416.1
#
```

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

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

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.

```
olympic2 <- rename(olympic_array, c("skater_ID"="pair", "judge_ID"="variable"))
olympic2 <- olympic2[order(olympic2$judge_ID),]
olympic2 <- olympic2[c("criterion", "value", "skater_ID", "judge_ID")]
```

#Now we have four columns with criterion value skater_ID and 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.

#But how do I know if they are from the same country

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

```
program <- olympic2 %>%
  filter(criterion=="Program")
Performance <- olympic2 %>%
  filter(criterion=="Performance")

fit24 <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID), data=program)
summary(fit24)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: value ~ 1 + (1 | skater_ID) + (1 | judge_ID)
## Data: program
##
## 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
```

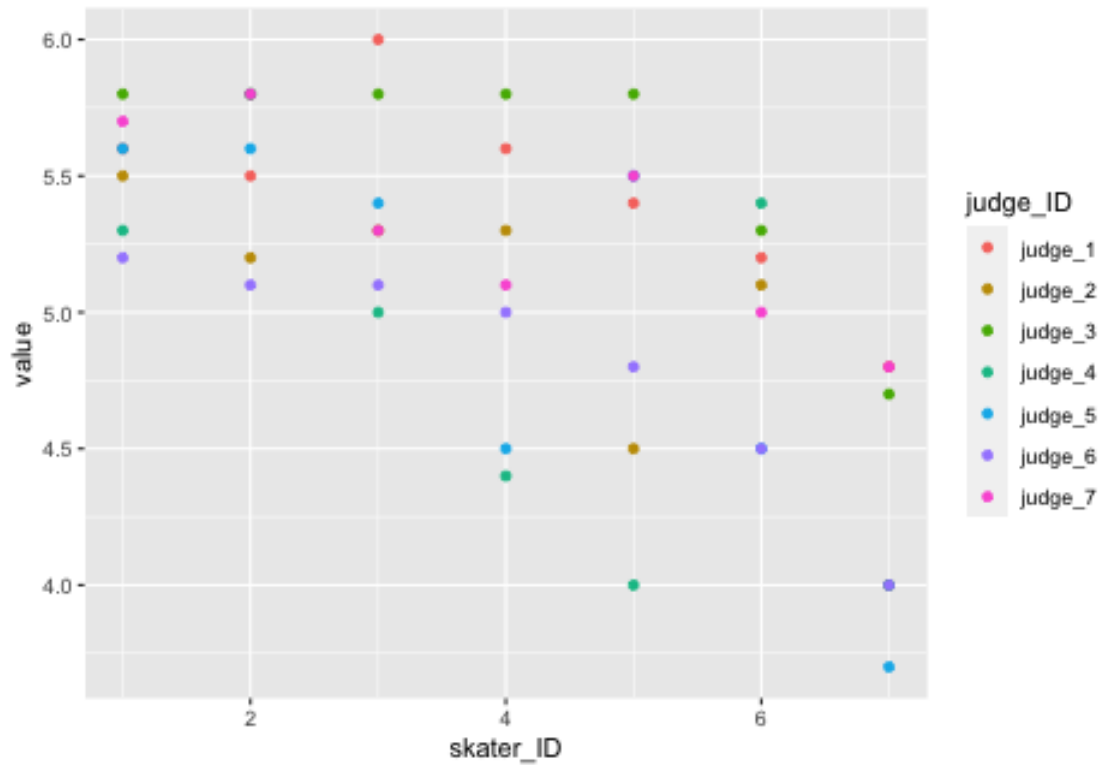
5. Fit the model in (4) using the artistic impression ratings.

```
fit25 <- lmer(value ~ 1 + (1|skater_ID) + (1|judge_ID),data=Performance)
summary(fit25)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: value ~ 1 + (1 | skater_ID) + (1 | judge_ID)
## Data: Performance
##
## REML criterion at convergence: 46.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.10128 -0.50469 -0.09884  0.40875  2.10489
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## skater_ID (Intercept) 0.20486  0.4526
## judge_ID (Intercept) 0.07759  0.2785
## Residual              0.07446  0.2729
## Number of obs: 49, groups: skater_ID, 7; judge_ID, 7
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   5.0918      0.2046   24.88
```

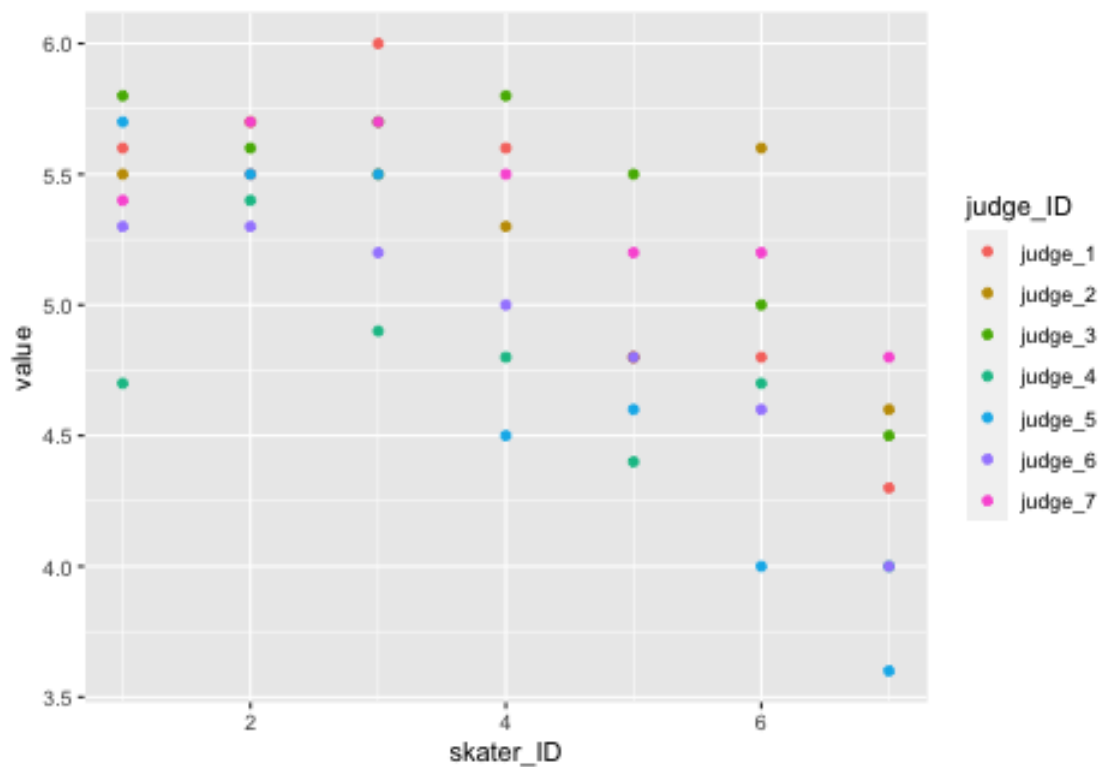
6. Display your results for both outcomes graphically.

```
ggplot(program,aes(x=skater_ID,y=value,color=judge_ID))+geom_point()
```

#Plot based on program scores

```
ggplot(Performance,aes(x=skater_ID,y=value,color=judge_ID))+geom_point()
```



```
#Plot based on performance score
```

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).

```
# fit_committee <- lmer(ratings~Applicants_ID+Raters_ID+(1|Raters_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.

```
# fit_committee2 <- lmer(ratings~Applicants_ID+Raters_ID+(1+Raters_ID|Raters_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<-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.

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

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=dating)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.0383479 (tol = 0.002, component 1)
```

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),data=dating)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.676671 (tol = 0.002, component 1)
```

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

```

```

## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

```

[illegible]

[illegible]

[illegible]

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
```

[illegible]


```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

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 +
```

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.