

MA678 Homework 7

Your Name

November 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.
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.
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.
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.
6. Investigate the change in partial pooling from (4) to (5) both graphically and numerically.
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.
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.
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).
12. Compare the results of these models both numerically and graphically.

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).
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.
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.
4. Write the notation for a non-nested multilevel model (varying across skaters and judges) for the technical merit ratings and fit using `lmer()`.
5. Fit the model in (4) using the artistic impression ratings.
6. Display your results for both outcomes graphically.
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}, \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=binom
```

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=dati
```

```
## 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),d
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.676556 (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$id)  
dating_no_pool_list<-vector("list",length(uiid))  
for(i in 1:length(uiid)){  
  # attr_o +sinc_o +intel_o +fun_o +amb_o+shar_o,  
  dating_no_pool_list[[i]] <- summary(glm(match~attr_o+shar_o,  
                                          data=dating,  
                                          subset = dating$id==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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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 +s
```

```
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 1).
## Chain 1: Rejecting initial value:
## Chain 1:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 1:   Stan can't start sampling from this initial value.
## Chain 1: Rejecting initial value:
## Chain 1:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 1:   Stan can't start sampling from this initial value.
## Chain 1: Rejecting initial value:
## Chain 1:   Log probability evaluates to log(0), i.e. negative infinity.
## Chain 1:   Stan can't start sampling from this initial value.
## Chain 1:
## Chain 1: Gradient evaluation took 0.00121 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 12.1 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]   (Warmup)
## Chain 1: Iteration:   200 / 2000 [ 10%]   (Warmup)
## Chain 1: Iteration:   400 / 2000 [ 20%]   (Warmup)
## Chain 1: Iteration:   600 / 2000 [ 30%]   (Warmup)
## Chain 1: Iteration:   800 / 2000 [ 40%]   (Warmup)
```

```

## Chain 1: Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 297.888 seconds (Warm-up)
## Chain 1: 122.436 seconds (Sampling)
## Chain 1: 420.324 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 2).
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2: Rejecting initial value:
## Chain 2: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 2: Stan can't start sampling from this initial value.
## Chain 2:
## Chain 2: Gradient evaluation took 0.000914 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 9.14 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration: 1 / 2000 [ 0%] (Warmup)
## Chain 2: Iteration: 200 / 2000 [ 10%] (Warmup)
## Chain 2: Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)

```

```

## Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 2:
## Chain 2: Elapsed Time: 293.054 seconds (Warm-up)
## Chain 2: 241.165 seconds (Sampling)
## Chain 2: 534.219 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000962 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 9.62 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)
## Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)
## Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 3:
## Chain 3: Elapsed Time: 307.34 seconds (Warm-up)
## Chain 3: 119.601 seconds (Sampling)
## Chain 3: 426.941 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'bernoulli' NOW (CHAIN 4).
## Chain 4: Rejecting initial value:
## Chain 4: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 4: Stan can't start sampling from this initial value.
## Chain 4: Rejecting initial value:
## Chain 4: Log probability evaluates to log(0), i.e. negative infinity.
## Chain 4: Stan can't start sampling from this initial value.
## Chain 4:
## Chain 4: Gradient evaluation took 0.000917 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 9.17 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)
## Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)
## Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)

```

```
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 4:
## Chain 4: Elapsed Time: 345.122 seconds (Warm-up)
## Chain 4:           247.871 seconds (Sampling)
## Chain 4:           592.993 seconds (Total)
## Chain 4:
```

```
## Warning: There were 8 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess
```

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.