MA678 Homework 7

Chang Lu

November 14, 2024

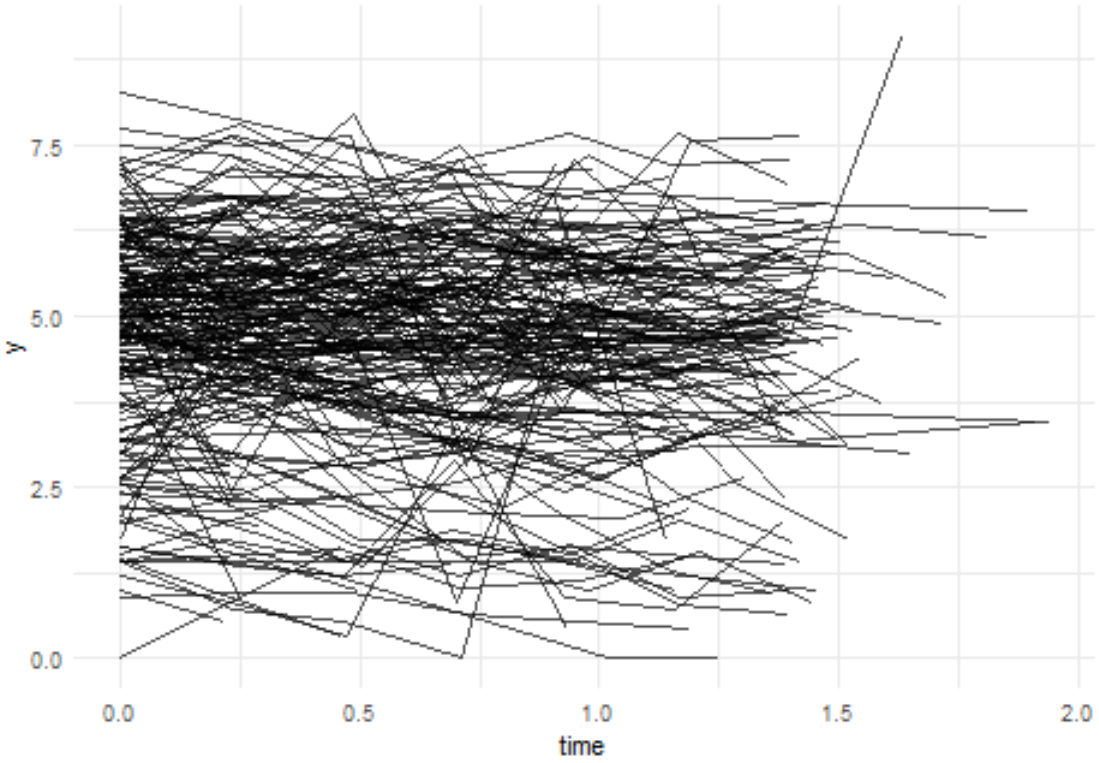
Data analysis

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(data = hiv.data, aes(x = time, y = y, group = newpid)) +
  geom_line(alpha = 0.7)+
  theme_minimal()
```



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.

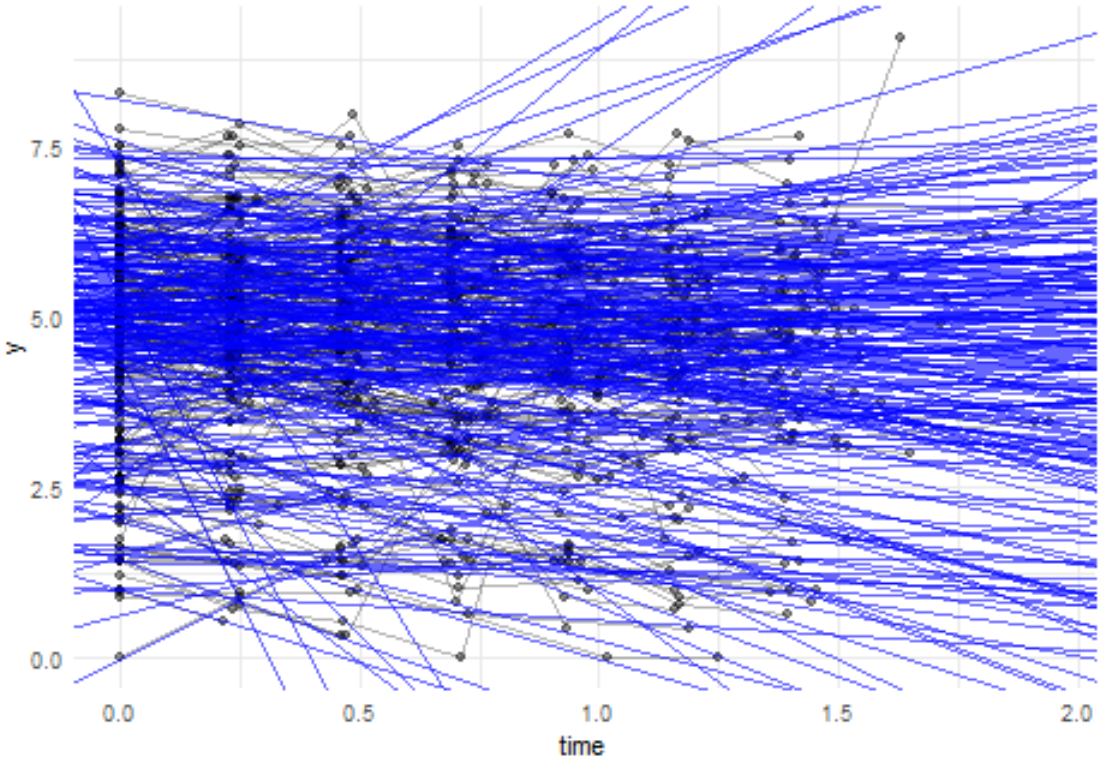
```
reg.1 <- hiv.data[, {
  model <- lm(y ~ time)
  list(intercept = coef(model)[1], slope = coef(model)[2])
}, by = newpid]
summary(reg.1)
```

```
##      newpid         intercept         slope        
##  Min.   :  1.00   Min.   :0.000   Min.   :-13.9961  
##  1st Qu.: 64.25   1st Qu.:3.991   1st Qu.: -0.8959  
##  Median :129.50   Median :4.990   Median : -0.2942  
##  Mean   :128.17   Mean   :4.764   Mean   : -0.4762  
##  3rd Qu.:191.75   3rd Qu.:5.758   3rd Qu.:  0.2778  
##  Max.   :254.00   Max.   :8.194   Max.   :  5.7003  
##                                   NA's   :26
```

```
hiv.data <- merge(hiv.data, reg.1, by = "newpid", all.x = TRUE)

ggplot(hiv.data, aes(x = time, y = y, group = newpid)) +
  geom_point(alpha = 0.4) +
  geom_line(alpha = 0.3) +
  geom_abline(aes(intercept = intercept, slope = slope), color = "blue", alpha = 0.6) +
  theme_minimal()
```

```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_abline()`).
```



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.

```
reg.2 <- merge(reg.1, unique(hiv.data[, .(newpid, treatment, age.baseline)]), by = "newpid", all.x = TRU
intercept_model <- lm(intercept ~ treatment + age.baseline, data = reg.2)
slope_model <- lm(slope ~ treatment + age.baseline, data = reg.2)
summary(intercept_model)
```

```
##
## Call:
## lm(formula = intercept ~ treatment + age.baseline, data = reg.2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0665 -0.7762  0.1892  1.0817  3.0391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.99423    0.32382  15.423  < 2e-16 ***
## treatment     0.12364    0.18736   0.660  0.50992
## age.baseline -0.12100    0.04092  -2.957  0.00341 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.48 on 247 degrees of freedom
## Multiple R-squared:  0.03577,    Adjusted R-squared:  0.02796
## F-statistic: 4.581 on 2 and 247 DF,  p-value: 0.01112
```

```
summary(slope_model)
```

```
##
## Call:
## lm(formula = slope ~ treatment + age.baseline, data = reg.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3917  -0.4547   0.2103   0.7651   6.0022
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.12642    0.46131  -0.274    0.784
## treatment    -0.13926    0.26936  -0.517    0.606
## age.baseline -0.04223    0.06016  -0.702    0.483
##
## Residual standard error: 2.009 on 221 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.003496,   Adjusted R-squared:  -0.005523
## F-statistic: 0.3876 on 2 and 221 DF,  p-value: 0.6791
```

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

```r
reg.3 <- lmer(y ~ time + (1|newpid),data = hiv.data)
summary(reg.3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3140.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7379 -0.4379  0.0024  0.4324  5.0017
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.9569   1.3989
##  Residual             0.5968   0.7725
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  4.76341    0.09648  49.372
## time        -0.36609    0.05399  -6.781
##
## Correlation of Fixed Effects:
##      (Intr)
## time -0.278
```

The coefficient for time represents with every one increasing unit in time the CD4 percentage decrease 0.36609 unit.

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

```r
reg.4 <- lmer(y ~ time + factor(treatment) + age.baseline + (1|newpid),data = hiv.data)
summary(reg.4)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + factor(treatment) + age.baseline + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3137.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7490 -0.4392  0.0097  0.4282  5.0141
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.8897   1.3747
##  Residual             0.5969   0.7726
```

```
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##                     Estimate Std. Error t value
## (Intercept)          5.08614    0.18793  27.064
## time                -0.36216    0.05399  -6.708
## factor(treatment)2   0.18008    0.18262   0.986
## age.baseline        -0.11945    0.04000  -2.986
##
## Correlation of Fixed Effects:
##             (Intr) time   fct()2
## time        -0.135
## fctr(trtm)2 -0.462  0.010
## age.baselin -0.727 -0.017 -0.003
```

The coefficient for time represents with every one increasing unit in time, the CD4 percentage decrease 0.36216 unit. The coefficient for treatment represents corresponding to treatment 1, the CD4 percentage increase 0.18008 unit. The coefficient for age.baseline represents with every one increasing unit in age.baseline, the CD4 percentage decrease 0.11945 unit.

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

```
var_reg.3 <- as.data.frame(VarCorr(reg.3))$vcov[1]
var_reg.4 <- as.data.frame(VarCorr(reg.4))$vcov[1]
cat("Variance of random intercepts in reg.3:", var_reg.3, "\n")
```

```
## Variance of random intercepts in reg.3: 1.956905
```

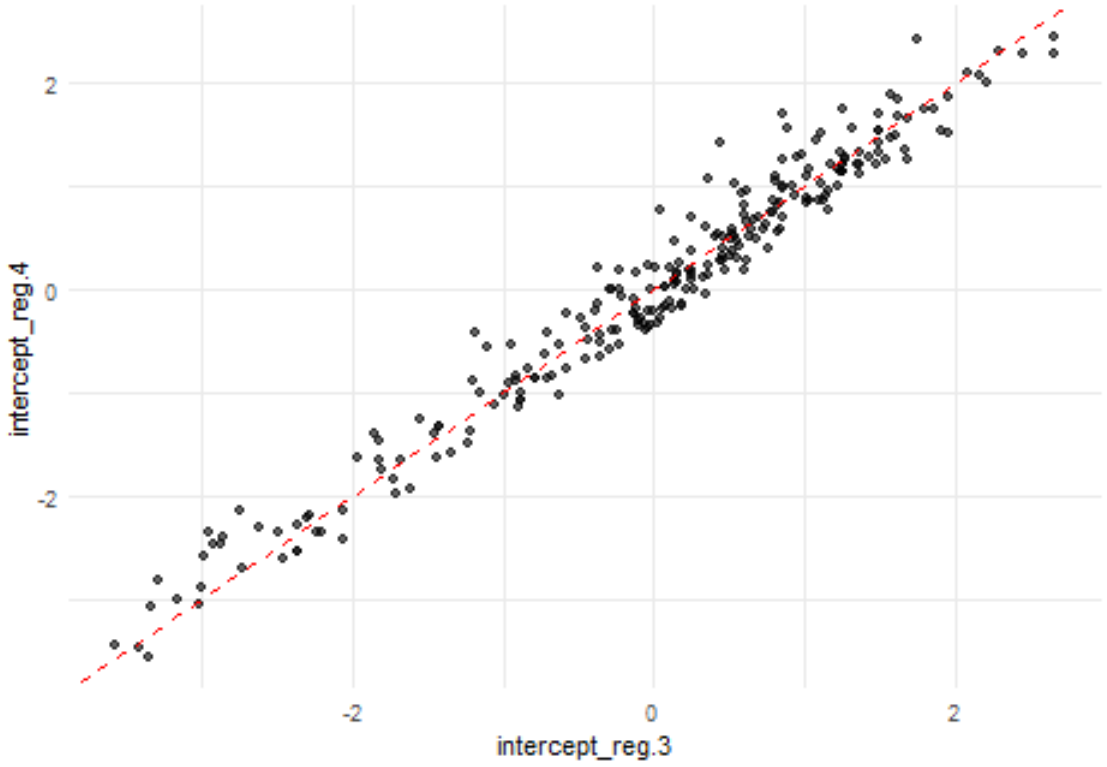
```
cat("Variance of random intercepts in reg.4:", var_reg.4, "\n")
```

```
## Variance of random intercepts in reg.4: 1.889682
```

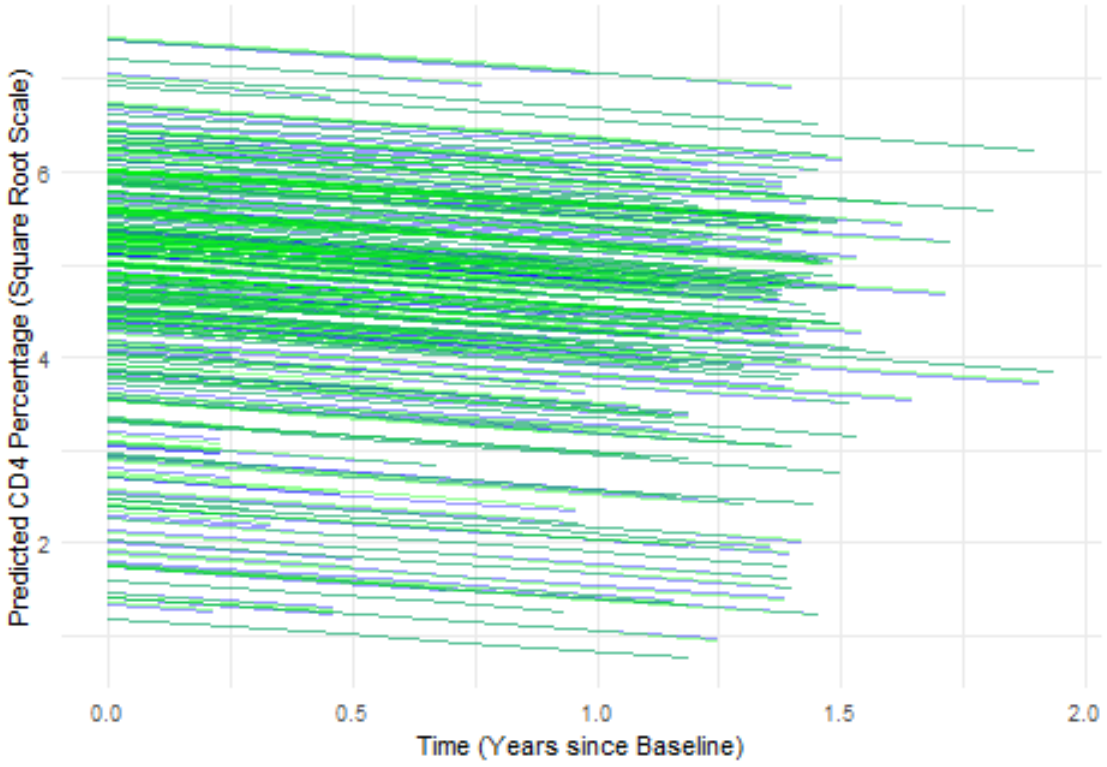
```
cat("The variance of random intercepts is lower in reg.4 than in reg.3, it suggests that treatment and a
```

```
## The variance of random intercepts is lower in reg.4 than in reg.3, it suggests that treatment and ag
```

```
random_intercepts_3 <- ranef(reg.3)$newpid[, "(Intercept)"]
random_intercepts_4 <- ranef(reg.4)$newpid[, "(Intercept)"]
random_intercepts <- data.frame(
  intercept_reg.3 = random_intercepts_3,
  intercept_reg.4 = random_intercepts_4
)
ggplot(random_intercepts, aes(x = intercept_reg.3, y = intercept_reg.4)) +
  geom_point(alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  theme_minimal()
```




```
hiv.data$predicted_reg.3 <- predict(reg.3)
hiv.data$predicted_reg.4 <- predict(reg.4)
ggplot(hiv.data, aes(x = time)) +
  geom_line(aes(y = predicted_reg.3, group = newpid), color = "blue", alpha = 0.4) +
  geom_line(aes(y = predicted_reg.4, group = newpid), color = "green", alpha = 0.4) +
    labs(
    x = "Time (Years since Baseline)",
    y = "Predicted CD4 Percentage (Square Root Scale)"
  ) +
  theme_minimal()
```



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.

```
set.seed(123)
future_data <- hiv.data[, .(newpid, treatment, age.baseline)]
future_data <- unique(future_data)
future_data[, time := max(hiv.data$time) + 1]
future_data$predicted_cd4 <- predict(reg.4, newdata = future_data, re.form = ~(1 | newpid))
future_data[, simulated_cd4 := predicted_cd4 + rnorm(.N, mean = 0, sd = sigma(reg.4))]
head(future_data[, .(id, treatment, age.baseline, time, predicted_cd4, simulated_cd4)])
```

```
##               id treatment age.baseline     time predicted_cd4 simulated_cd4
##           <list>     <int>        <num>    <num>         <num>         <num>
## 1: <function[1]>         1    3.9100000 2.938333     3.4814737     3.0484630
## 2: <function[1]>         2    3.5650000 2.938333     0.2977147     0.1198849
## 3: <function[1]>         1    6.1241667 2.938333     4.7974810     6.0017034
## 4: <function[1]>         1    2.3025000 2.938333     4.4957639     4.5502370
## 5: <function[1]>         1    0.6541667 2.938333     3.1778213     3.2777060
## 6: <function[1]>         2    2.9183333 2.938333     4.2667427     5.5917626
```

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.

```
set.seed(123)
time_points <- unique(hiv.data$time)
new_child_data <- data.frame(
  newpid = "new_child",
```

```
  treatment = 1,
  age.baseline = 4,
  time = time_points
)
new_child_data$predicted_cd4 <- predict(reg.4, newdata = new_child_data, re.form = NA)
new_child_data$simulated_cd4 <- new_child_data$predicted_cd4 + rnorm(nrow(new_child_data), mean = 0, sd
print(new_child_data[, c("time", "predicted_cd4", "simulated_cd4")])
```

```
##          time predicted_cd4 simulated_cd4
## 1   0.0000000      4.608323      4.175312
## 2   0.5583333      4.406118      4.228289
## 3   0.7883333      4.322822      5.527045
## 4   1.4208333      4.093758      4.148231
## 5   1.9383333      3.906341      4.006226
## 6   0.2133333      4.531063      5.856083
## 7   0.2300000      4.525027      4.881120
## 8   0.4600000      4.441731      3.464373
## 9   0.8050000      4.316786      3.786140
## 10  1.1883333      4.177959      3.833651
## 11  1.5141667      4.059956      5.005654
## 12  1.7250000      3.983602      4.261586
## 13  0.4791667      4.434789      4.744416
## 14  0.7283333      4.344552      4.430063
## 15  0.9583333      4.261256      3.831825
## 16  0.3041667      4.498167      5.878695
## 17  0.5341667      4.414871      4.799498
## 18  0.7833333      4.324633      2.805270
## 19  1.0325000      4.234396      4.776247
## 20  1.5308333      4.053920      3.688653
## 21  0.2300000      4.525027      3.700051
## 22  0.9575000      4.261557      4.093155
## 23  1.2066667      4.171320      3.378652
## 24  1.4558333      4.081082      3.517958
## 25  0.2491667      4.518085      4.035194
## 26  0.5750000      4.400082      3.096982
## 27  0.8241667      4.309845      4.957100
## 28  1.2458333      4.157135      4.275628
## 29  0.2300000      4.525027      3.645728
## 30  0.4600000      4.441731      5.410399
## 31  0.7091667      4.351493      4.680970
## 32  0.9583333      4.261256      4.033290
## 33  1.1883333      4.177959      4.869513
## 34  0.3066667      4.497261      5.175687
## 35  0.5366667      4.413965      5.048700
## 36  0.7858333      4.323728      4.855755
## 37  1.0350000      4.233490      4.661434
## 38  1.5333333      4.053015      4.005183
## 39  0.2300000      4.525027      4.288647
## 40  0.4408333      4.448672      4.154729
## 41  0.6708333      4.365376      3.828661
## 42  0.9391667      4.268197      4.107565
## 43  1.1691667      4.184901      3.207284
## 44  1.4183333      4.094663      5.770349
```

```
## 45  0.2683333      4.511144      5.444388
## 46  0.4983333      4.427848      3.560160
## 47  0.7475000      4.337610      4.026351
## 48  0.9966667      4.247373      3.886846
## 49  1.3033333      4.136311      4.738894
## 50  0.5941667      4.393141      4.328732
## 51  0.2491667      4.518085      4.713793
## 52  0.6900000      4.358434      4.336380
## 53  0.9200000      4.275138      4.242017
## 54  1.1500000      4.191842      5.249193
## 55  1.3716667      4.111564      3.937138
## 56  0.4983333      4.427848      5.599438
## 57  0.7283333      4.344552      3.148021
## 58  0.9608333      4.260350      4.712009
## 59  0.7616667      4.332480      4.428167
## 60  1.0108333      4.242242      4.409074
## 61  1.2650000      4.150194      4.443495
## 62  1.4950000      4.066898      3.678814
## 63  0.2358333      4.522914      4.265486
## 64  0.6925000      4.357529      3.570601
## 65  0.9258333      4.273026      3.444984
## 66  0.2300000      4.525027      4.759526
## 67  0.9558333      4.262161      4.608438
## 68  1.6483333      4.011367      4.052317
## 69  0.2383333      4.522009      5.234531
## 70  0.5016667      4.426641      6.010489
## 71  0.4291667      4.452897      4.073538
## 72  0.6758333      4.363565      2.579554
## 73  0.9083333      4.279363      5.056374
## 74  1.4533333      4.081988      3.534075
## 75  0.4816667      4.433884      3.902344
## 76  0.7091667      4.351493      5.143826
## 77  0.9391667      4.268197      4.048188
## 78  0.2300000      4.525027      3.581928
## 79  0.7091667      4.351493      4.491564
## 80  1.1800000      4.180977      4.073673
## 81  1.4150000      4.095870      4.100324
## 82  0.2133333      4.531063      4.828722
## 83  0.4291667      4.452897      4.166534
## 84  0.6733333      4.364470      4.862301
## 85  0.9033333      4.281174      4.110831
## 86  1.1358333      4.196973      4.453300
## 87  0.7141667      4.349682      5.197075
## 88  0.9383333      4.268499      4.604710
## 89  1.1683333      4.185202      3.933395
## 90  0.4791667      4.434789      5.322332
## 91  0.5333333      4.415172      5.182731
## 92  0.7633333      4.331876      4.755555
## 93  0.9966667      4.247373      4.431811
## 94  0.7008333      4.354511      3.869405
## 95  0.6516667      4.372317      5.423526
## 96  1.0925000      4.212666      3.748919
## 97  1.3500000      4.119411      5.809294
## 98  1.5908333      4.032191      5.216251
```

```
## 99  0.2408333      4.521103      4.339007
## 100 0.9333333      4.270309      3.477320
## 101 1.2016667      4.173131      3.624287
## 102 1.4700000      4.075952      4.274414
## 103 0.2575000      4.515067      4.324479
## 104 0.2275000      4.525932      4.257429
## 105 0.5641667      4.404006      3.668807
## 106 0.8075000      4.315881      4.281094
## 107 1.0350000      4.233490      3.627091
## 108 0.2525000      4.516878      3.228264
## 109 0.5150000      4.421812      4.128058
## 110 0.7258333      4.345457      5.055453
## 111 0.4433333      4.447767      4.003267
## 112 0.7366667      4.341534      4.811233
## 113 0.9658333      4.258539      3.008600
## 114 0.2575000      4.515067      4.472142
## 115 0.5041667      4.425735      4.827017
## 116 0.7258333      4.345457      4.578121
## 117 0.9641667      4.259143      4.340786
## 118 1.1966667      4.174941      3.679947
## 119 1.4975000      4.065992      3.409530
## 120 0.7258333      4.345457      3.554239
## 121 0.9641667      4.259143      4.350034
## 122 1.1966667      4.174941      3.442944
## 123 1.4975000      4.065992      3.686999
## 124 0.2433333      4.520198      4.322347
## 125 0.4733333      4.436902      5.861427
## 126 0.4633333      4.440523      3.936842
## 127 0.6900000      4.358434      4.540289
## 128 0.9091667      4.279062      4.339292
## 129 1.1416667      4.194860      3.451752
## 130 1.3716667      4.111564      4.056473
## 131 0.7116667      4.350588      5.466615
## 132 0.9416667      4.267291      4.616113
## 133 1.1716667      4.183995      4.215851
## 134 1.4016667      4.100699      3.774288
## 135 0.4766667      4.435695      2.849403
## 136 1.3991667      4.101605      4.975650
## 137 0.4708333      4.437807      3.309350
## 138 0.9008333      4.282080      4.853746
## 139 1.1141667      4.204819      5.679749
## 140 0.5008333      4.426942      3.311424
## 141 0.7250000      4.345759      4.887941
## 142 0.9550000      4.262463      4.059895
## 143 1.2041667      4.172225      2.957623
## 144 1.4258333      4.091947      2.921749
## 145 0.2458333      4.519293      3.281982
## 146 0.4758333      4.435996      4.025830
## 147 0.7058333      4.352700      3.223381
## 148 0.9358333      4.269404      4.800873
## 149 1.1775000      4.181883      5.804379
## 150 1.4150000      4.095870      3.101540
## 151 0.2541667      4.516275      5.124864
## 152 0.4816667      4.433884      5.028028
```

```
## 153 0.9766667      4.254616      4.511268
## 154 0.2408333      4.521103      3.742055
## 155 0.4658333      4.439618      4.347332
## 156 0.6958333      4.356322      4.139695
## 157 0.9283333      4.272120      4.707073
## 158 1.1558333      4.189729      3.901992
## 159 1.3858333      4.106433      4.861221
## 160 0.2325000      4.524121      4.234729
## 161 0.4625000      4.440825      5.254126
## 162 0.9116667      4.278156      3.467586
## 163 1.3658333      4.113676      3.140109
## 164 0.2358333      4.522914      7.026867
## 165 0.9308333      4.271215      3.949160
## 166 1.1608333      4.187919      4.418322
## 167 1.4100000      4.097681      4.589480
## 168 0.4575000      4.442636      4.068878
## 169 0.6850000      4.360245      4.759561
## 170 0.9175000      4.276044      4.561097
## 171 1.1475000      4.192747      4.026349
## 172 1.3583333      4.116393      4.166836
## 173 0.4658333      4.439618      4.413298
## 174 0.7150000      4.349380      5.993774
## 175 1.3908333      4.104622      3.531883
## 176 0.2350000      4.523216      3.676474
## 177 0.4650000      4.439920      4.469114
## 178 1.3908333      4.104622      4.344493
## 179 0.2491667      4.518085      4.855333
## 180 0.4791667      4.434789      4.080667
## 181 0.7441667      4.338818      3.517316
## 182 0.9775000      4.254314      5.230222
## 183 1.2075000      4.171018      3.900886
## 184 1.4316667      4.089834      3.421159
## 185 0.2241667      4.527139      4.344595
## 186 0.2491667      4.518085      4.365752
## 187 0.6983333      4.355416      5.212915
## 188 0.9341667      4.270008      4.335474
## 189 1.1641667      4.186711      4.769276
## 190 0.2458333      4.519293      4.133551
## 191 0.5141667      4.422114      4.587789
## 192 0.7333333      4.342741      4.091896
## 193 0.9825000      4.252503      4.325576
## 194 1.2125000      4.169207      3.477470
## 195 1.4750000      4.074141      3.061446
## 196 0.2491667      4.518085      6.061087
## 197 0.4875000      4.431771      4.895865
## 198 0.9391667      4.268197      3.301493
## 199 1.1691667      4.184901      3.712728
## 200 1.3991667      4.101605      3.185730
## 201 0.6925000      4.357529      6.056280
## 202 0.9416667      4.267291      5.281232
## 203 1.1575000      4.189126      3.984281
## 204 1.3875000      4.105830      4.525489
## 205 0.4708333      4.437807      4.117698
## 206 0.7008333      4.354511      3.986574
```

```
## 207 0.9308333      4.271215      3.661958
## 208 0.6983333      4.355416      3.896029
## 209 1.2266667      4.164077      5.439530
## 210 1.4950000      4.066898      4.025157
## 211 1.7166667      3.986620      4.078746
## 212 0.3641667      4.476437      4.664705
## 213 0.6483333      4.373524      5.325707
## 214 0.8783333      4.290228      3.891529
## 215 0.2875000      4.504203      3.737414
## 216 0.5333333      4.415172      5.709777
## 217 1.0208333      4.238621      3.897788
## 218 0.4600000      4.441731      3.883106
## 219 0.6875000      4.359340      3.404223
## 220 0.9200000      4.275138      3.282596
## 221 1.1500000      4.191842      3.748403
## 222 1.3741667      4.110658      4.588100
## 223 0.4758333      4.435996      5.293440
## 224 0.9191667      4.275440      4.822107
## 225 1.1491667      4.192144      3.911191
## 226 1.3791667      4.108848      4.155009
## 227 0.6900000      4.358434      3.814079
## 228 1.1500000      4.191842      3.637736
## 229 1.3750000      4.110357      4.793817
## 230 0.6900000      4.358434      3.573811
## 231 0.9200000      4.275138      5.785754
## 232 0.2300000      4.525027      4.455248
## 233 0.4600000      4.441731      4.607478
## 234 0.6900000      4.358434      3.787865
## 235 0.9391667      4.268197      3.824437
## 236 1.1500000      4.191842      3.174346
## 237 1.3766667      4.109753      3.968429
## 238 0.2050000      4.534081      4.857777
## 239 0.4383333      4.449577      4.700127
## 240 0.5175000      4.420907      3.817109
## 241 0.8050000      4.316786      3.707515
## 242 0.7091667      4.351493      3.963506
## 243 1.1883333      4.177959      5.333782
## 244 0.4600000      4.441731      3.563076
## 245 1.3416667      4.122429      3.984097
## 246 0.9200000      4.275138      5.744859
## 247 1.1883333      4.177959      4.099948
## 248 1.4183333      4.094663      3.044081
## 249 0.2491667      4.518085      4.004500
## 250 0.7475000      4.337610      4.712666
## 251 0.5058333      4.425132      4.134950
## 252 0.7283333      4.344552      3.910459
## 253 0.9583333      4.261256      3.995553
## 254 0.4600000      4.441731      4.511646
## 255 1.4700000      4.075952      5.310923
## 256 0.7425000      4.339421      4.270998
## 257 0.9858333      4.251296      5.086297
## 258 0.2491667      4.518085      5.005392
## 259 0.4708333      4.437807      4.350012
## 260 0.2158333      4.530157      3.345872
```

```
## 261 0.4650000      4.439920      4.037316
## 262 0.6950000      4.356624      3.978161
## 263 0.9441667      4.266386      4.302817
## 264 0.2550000      4.515973      5.520477
## 265 0.5558333      4.407024      6.178604
## 266 0.8025000      4.317692      5.513317
## 267 1.0925000      4.212666      4.109797
## 268 1.6291667      4.018308      2.661255
## 269 0.2358333      4.522914      4.222552
## 270 0.7283333      4.344552      4.413471
## 271 0.9991667      4.246467      4.899305
## 272 1.2650000      4.150194      4.893821
## 273 0.2333333      4.523820      5.052501
## 274 0.7666667      4.330669      3.252712
## 275 1.0000000      4.246166      4.902580
## 276 1.2675000      4.149289      3.804289
## 277 1.5166667      4.059051      4.194100
## 278 0.2933333      4.502090      4.559687
## 279 0.6108333      4.387105      4.717897
## 280 1.0900000      4.213571      4.232635
## 281 0.2566667      4.515369      3.227116
## 282 0.5141667      4.422114      4.991114
## 283 1.2258333      4.164378      4.462614
## 284 1.0158333      4.240431      4.035195
## 285 0.2166667      4.529855      4.621131
## 286 0.7475000      4.337610      4.441166
## 287 1.2316667      4.162266      4.333020
## 288 1.4925000      4.067803      5.335483
## 289 0.2900000      4.503297      4.334064
## 290 0.2466667      4.518991      4.648834
## 291 0.4958333      4.428753      5.331420
## 292 0.7641667      4.331574      5.146011
## 293 1.0158333      4.240431      5.125236
## 294 1.2841667      4.143253      3.697114
## 295 1.5191667      4.058146      5.605218
## 296 0.4950000      4.429055      4.480587
## 297 0.7850000      4.324029      5.766316
## 298 1.0533333      4.226851      3.183174
## 299 1.6150000      4.023439      4.039650
## 300 0.4983333      4.427848      5.393503
## 301 0.7633333      4.331876      3.779296
## 302 1.0125000      4.241639      3.660128
## 303 1.2616667      4.151401      3.426308
## 304 1.5000000      4.065087      3.251939
## 305 0.2183333      4.529252      4.191512
## 306 0.5091667      4.423924      4.679786
## 307 0.7358333      4.341836      2.785703
## 308 1.0233333      4.237715      4.401487
## 309 1.2533333      4.154419      5.109846
## 310 1.4833333      4.071123      5.645306
## 311 0.3583333      4.478550      5.483809
## 312 0.7583333      4.333687      4.918354
## 313 1.3991667      4.101605      2.767572
## 314 1.9083333      3.917206      3.452496
```

```
## 315 0.8025000      4.317692      4.045709
## 316 0.3316667      4.488207      5.031734
## 317 0.9825000      4.252503      4.170864
## 318 1.4725000      4.075046      3.102643
## 319 0.5366667      4.413965      5.715322
## 320 1.4291667      4.090740      4.794860
## 321 0.6575000      4.370204      4.553638
## 322 0.9091667      4.279062      5.220144
## 323 0.4600000      4.441731      3.407424
## 324 1.1691667      4.184901      4.695435
## 325 1.1691667      4.184901      3.780911
## 326 0.2325000      4.524121      5.052367
## 327 0.4625000      4.440825      4.393836
## 328 1.1525000      4.190937      4.679948
## 329 0.4983333      4.427848      5.459638
## 330 1.0350000      4.233490      4.239122
## 331 0.7116667      4.350588      5.136730
## 332 0.2383333      4.522009      3.603852
## 333 0.4708333      4.437807      3.880312
## 334 0.7033333      4.353606      5.527318
## 335 0.9475000      4.265179      4.556740
## 336 1.1633333      4.187013      2.601513
## 337 1.3958333      4.102812      3.048988
## 338 0.7066667      4.352398      4.197280
## 339 0.9775000      4.254314      4.923196
## 340 1.2325000      4.161964      4.083251
## 341 1.4625000      4.078668      4.560901
## 342 0.2266667      4.526234      5.267140
## 343 0.4733333      4.436902      5.727921
## 344 0.7091667      4.351493      4.394770
## 345 0.9525000      4.263368      4.223208
## 346 1.2016667      4.173131      2.818620
## 347 1.4566667      4.080780      4.157519
## 348 0.2658333      4.512049      4.070251
## 349 0.5033333      4.426037      3.673540
## 350 0.7525000      4.335800      4.196808
## 351 1.0016667      4.245562      5.029684
## 352 1.2291667      4.163171      2.623619
## 353 0.2683333      4.511144      4.181038
## 354 0.5175000      4.420907      4.511018
## 355 0.7750000      4.327651      3.637579
## 356 1.3116667      4.133293      4.391259
## 357 1.5416667      4.049997      4.367858
## 358 0.2516667      4.517180      4.491657
## 359 0.4850000      4.432677      2.527580
## 360 0.7366667      4.341534      6.328183
## 361 1.0025000      4.245260      4.086651
## 362 1.2483333      4.156230      4.659327
## 363 1.4816667      4.071727      4.283232
## 364 0.4850000      4.432677      5.224316
## 365 0.7150000      4.349380      4.981085
## 366 0.9775000      4.254314      4.092233
## 367 1.2241667      4.164982      4.457146
## 368 1.4541667      4.081686      3.351285
```

```
## 369 0.2683333      4.511144      5.173183
## 370 0.5116667      4.423019      4.066831
## 371 0.7658333      4.330971      6.198115
## 372 1.0483333      4.228661      2.953099
## 373 1.3025000      4.136613      3.778147
## 374 0.7341667      4.342439      4.980109
## 375 0.9775000      4.254314      4.648431
## 376 1.4458333      4.084704      3.629284
## 377 0.2241667      4.527139      3.757049
## 378 0.9358333      4.269404      4.381023
## 379 1.1766667      4.182185      4.171131
## 380 1.4125000      4.096776      2.713645
## 381 0.2441667      4.519896      4.546590
## 382 0.5008333      4.426942      4.573910
## 383 0.7475000      4.337610      4.472600
## 384 0.9775000      4.254314      3.439232
## 385 1.2291667      4.163171      4.531021
## 386 0.4841667      4.432978      5.498030
## 387 0.7175000      4.348475      4.700953
## 388 0.9475000      4.265179      3.387849
## 389 0.2300000      4.525027      4.188457
## 390 0.4816667      4.433884      4.701276
## 391 0.7116667      4.350588      3.850695
## 392 0.9550000      4.262463      2.595515
## 393 0.4791667      4.434789      5.117941
## 394 0.7450000      4.338516      3.697680
## 395 0.2466667      4.518991      4.075871
## 396 0.4900000      4.430866      5.592745
## 397 0.4575000      4.442636      3.844549
## 398 0.2875000      4.504203      5.157596
## 399 0.8625000      4.295962      3.321988
## 400 0.9966667      4.247373      3.973461
## 401 1.4950000      4.066898      4.010070
## 402 0.2683333      4.511144      3.608271
## 403 0.8625000      4.295962      3.805570
## 404 0.3066667      4.497261      4.474979
## 405 0.5558333      4.407024      4.925188
## 406 0.5175000      4.420907      3.145732
## 407 0.7666667      4.330669      4.060457
## 408 0.7666667      4.330669      4.915051
## 409 1.3983333      4.101906      3.685635
## 410 1.8966667      3.921431      4.097032
## 411 1.0541667      4.226549      4.606833
## 412 0.7800000      4.325840      4.532763
## 413 1.0866667      4.214779      4.719471
## 414 0.3258333      4.490320      4.395518
## 415 0.5750000      4.400082      4.080485
## 416 1.3608333      4.115487      2.073451
## 417 0.9583333      4.261256      4.189451
## 418 1.3800000      4.108546      4.440974
## 419 0.6708333      4.365376      4.779013
## 420 0.6891667      4.358736      3.929741
## 421 1.1491667      4.192144      5.566947
## 422 0.2100000      4.532270      4.753555
```

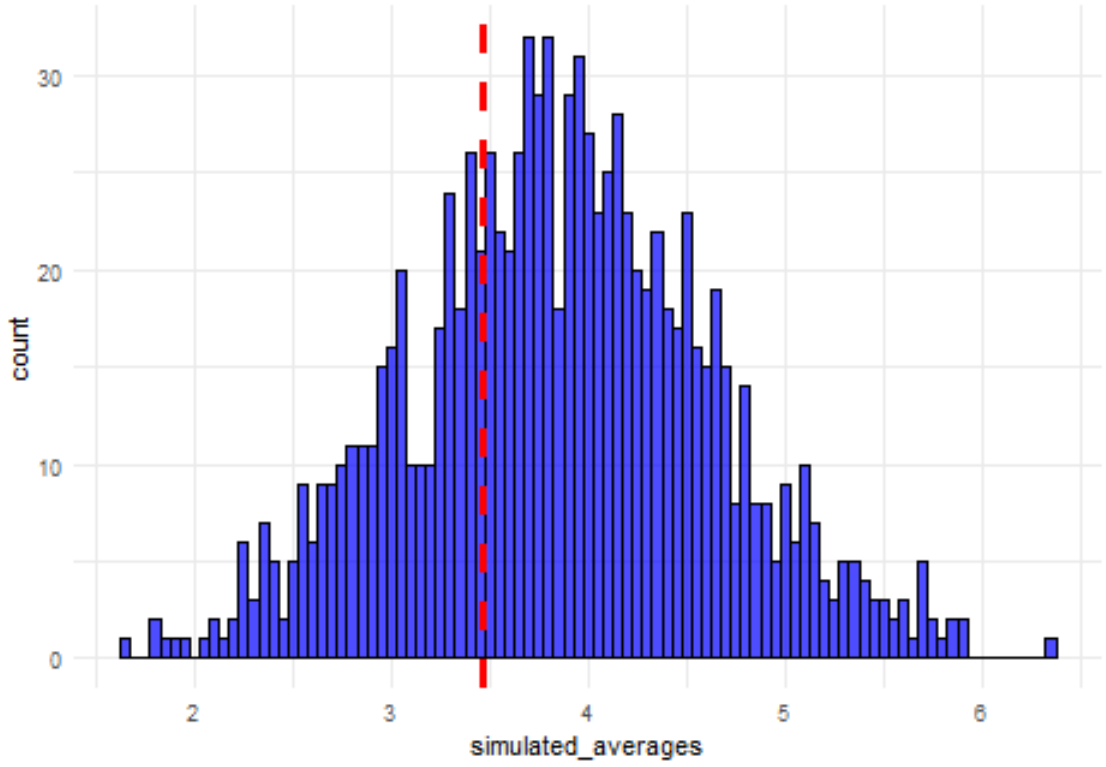
```
## 423 0.4591667      4.442032      4.539621
## 424 0.6891667      4.358736      5.341660
## 425 1.3600000      4.115789      3.560718
## 426 1.2458333      4.157135      3.809214
## 427 0.6700000      4.365678      6.217894
## 428 0.9008333      4.282080      4.290678
## 429 1.1308333      4.198783      5.460841
## 430 0.2216667      4.528045      3.416687
## 431 0.4566667      4.442938      4.295749
## 432 0.9200000      4.275138      4.567500
## 433 0.2300000      4.525027      4.756830
## 434 0.2466667      4.518991      3.742059
## 435 1.5058333      4.062974      4.077854
## 436 0.4958333      4.428753      3.596363
## 437 0.9583333      4.261256      4.811874
## 438 1.1883333      4.177959      5.016032
## 439 1.4241667      4.092551      2.373576
## 440 0.2266667      4.526234      5.480902
## 441 1.1658333      4.186108      3.227305
## 442 1.4566667      4.080780      4.432125
## 443 0.4591667      4.442032      4.951858
## 444 0.9191667      4.275440      4.121010
## 445 1.4366667      4.088024      3.589623
## 446 0.4625000      4.440825      4.568548
## 447 0.2491667      4.518085      4.857107
## 448 0.5175000      4.420907      5.103326
## 449 0.4575000      4.442636      2.857047
## 450 1.3966667      4.102510      2.838281
## 451 0.7058333      4.352700      5.457796
## 452 0.9141667      4.277251      5.085852
## 453 0.2191667      4.528950      4.865244
## 454 0.4683333      4.438713      4.991243
## 455 0.6933333      4.357227      5.065816
## 456 0.9225000      4.274233      2.218465
## 457 1.1583333      4.188824      5.046599
## 458 0.8816667      4.289021      3.914331
## 459 1.1166667      4.203914      4.382083
## 460 0.9391667      4.268197      4.040165
## 461 0.4491667      4.445654      5.119314
## 462 0.7941667      4.320710      4.051488
## 463 0.2650000      4.512351      4.912935
## 464 0.7308333      4.343646      4.041812
## 465 1.5300000      4.054222      3.209960
## 466 1.8116667      3.952215      4.887041
## 467 0.7308333      4.343646      4.916049
## 468 1.0758333      4.218702      5.550828
## 469 1.4016667      4.100699      4.151036
## 470 0.5808333      4.397970      5.267121
## 471 0.2625000      4.513257      6.039420
## 472 1.4316667      4.089834      3.872368
## 473 1.0541667      4.226549      3.204467
## 474 0.2358333      4.522914      4.337997
## 475 0.4625000      4.440825      4.275462
## 476 0.6925000      4.357529      4.474714
```

```
## 477 0.9200000      4.275138      5.598026
## 478 1.3825000      4.107640      3.855669
## 479 0.9258333      4.273026      4.561200
## 480 1.3883333      4.105528      3.929624
## 481 0.4658333      4.439618      4.455418
## 482 0.6958333      4.356322      4.598956
## 483 1.3858333      4.106433      5.132582
## 484 0.9033333      4.281174      4.374902
## 485 0.4675000      4.439014      4.989740
## 486 0.6975000      4.355718      4.957448
## 487 1.1575000      4.189126      4.895859
## 488 1.3875000      4.105830      3.662066
## 489 0.4733333      4.436902      5.693793
## 490 0.7033333      4.353606      4.059287
## 491 0.9308333      4.271215      4.189488
## 492 1.1608333      4.187919      5.272656
## 493 0.2375000      4.522311      5.522090
## 494 0.4675000      4.439014      3.596912
## 495 0.7200000      4.347570      3.673055
## 496 1.1716667      4.183995      3.134774
## 497 1.4100000      4.097681      4.238172
## 498 0.4708333      4.437807      4.565159
## 499 0.9225000      4.274233      4.555539
## 500 1.1575000      4.189126      4.615710
## 501 0.6983333      4.355416      3.890408
## 502 1.1583333      4.188824      3.421115
## 503 0.4741667      4.436600      5.229871
## 504 0.7041667      4.353304      4.933557
## 505 0.9341667      4.270008      3.104060
## 506 1.1608333      4.187919      4.114410
## 507 1.3800000      4.108546      3.416357
## 508 0.2516667      4.517180      2.917365
## 509 0.4875000      4.431771      4.547751
## 510 1.3858333      4.106433      4.045236
## 511 0.2241667      4.527139      4.451914
## 512 0.4541667      4.443843      4.610838
## 513 0.6841667      4.360547      5.042319
## 514 0.9141667      4.277251      4.436091
## 515 1.1441667      4.193955      3.717710
## 516 1.3825000      4.107640      3.539951
## 517 0.2275000      4.525932      4.424104
## 518 0.4550000      4.443541      4.683053
## 519 0.8958333      4.283890      3.480657
## 520 1.1450000      4.193653      4.051260
## 521 1.3800000      4.108546      4.855834
## 522 0.4600000      4.441731      4.358076
## 523 0.9283333      4.272120      3.732536
## 524 0.4600000      4.441731      4.228542
## 525 0.6900000      4.358434      5.219586
## 526 0.9366667      4.269102      4.694054
## 527 1.1391667      4.195765      5.151193
## 528 0.7033333      4.353606      4.461069
## 529 0.2458333      4.519293      4.836262
## 530 1.1500000      4.191842      3.760391
```

```
## 531 0.9833333      4.252202      4.719897
## 532 0.4741667      4.436600      4.045418
## 533 0.6708333      4.365376      3.267879
## 534 0.7058333      4.352700      4.451585
## 535 0.9333333      4.270309      5.773630
## 536 1.1633333      4.187013      4.805781
## 537 0.9775000      4.254314      5.154562
## 538 1.4375000      4.087722      4.364966
## 539 0.2050000      4.534081      4.063923
## 540 0.6375000      4.377448      4.221201
## 541 0.8675000      4.294151      4.083046
## 542 0.2158333      4.530157      4.168051
## 543 0.4541667      4.443843      4.987867
## 544 0.9058333      4.280269      3.355213
## 545 0.2358333      4.522914      5.192249
## 546 0.6958333      4.356322      5.023946
## 547 1.2041667      4.172225      3.246197
## 548 0.4875000      4.431771      4.925828
## 549 0.2516667      4.517180      6.394717
## 550 0.3283333      4.489415      4.058923
## 551 0.5825000      4.397366      5.050120
```

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.

```r
set.seed(123)
final_time <- max(hiv.data$time)
observed_final <- hiv.data[time == final_time]
observed_avg_cd4 <- mean(observed_final$y)
n_sim <- 1000
simulated_averages <- numeric(n_sim)
for (i in 1:n_sim) {
  sim_data <- observed_final[, .(newpid, treatment, age.baseline, time = final_time)]
  sim_data$predicted_cd4 <- predict(reg.4, newdata = sim_data, re.form = ~(1 | newpid))
  sim_data$simulated_cd4 <- sim_data$predicted_cd4 + rnorm(nrow(sim_data), mean = 0, sd = sigma(reg.4))
  simulated_averages[i] <- mean(sim_data$simulated_cd4)
}
ggplot(data.frame(simulated_averages), aes(x = simulated_averages)) +
  geom_histogram(binwidth = 0.05, fill = "blue", color = "black", alpha = 0.7) +
  geom_vline(xintercept = observed_avg_cd4, color = "red", linetype = "dashed", linewidth = 1.2)+
  theme_minimal()
```




```
cat("Observed average CD4 percentage at final time point (square root scale):", observed_avg_cd4, "\n")
```

```
## Observed average CD4 percentage at final time point (square root scale): 3.464102
```

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

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ time + factor(treatment) + age.baseline + (1 + time | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 3107
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.0998 -0.4057  0.0174  0.4030  5.0157
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  newpid   (Intercept) 1.8464   1.3588
##           time        0.3374   0.5808   -0.04
##  Residual             0.5145   0.7173
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
```

```
##                    Estimate Std. Error t value
## (Intercept)         5.10850    0.18594  27.474
## time               -0.35258    0.06763  -5.214
## factor(treatment)2  0.15952    0.18137   0.880
## age.baseline       -0.12423    0.03971  -3.128
##
## Correlation of Fixed Effects:
##             (Intr) time   fct()2
## time        -0.114
## fctr(trtm)2 -0.463  0.010
## age.baselin -0.729 -0.013 -0.004
```

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).

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

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ factor(time) + factor(treatment) + age.baseline + (1 | newpid)
##    Data: hiv.data
##
## REML criterion at convergence: 2166.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9719 -0.2686  0.0000  0.2514  4.2168
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  newpid   (Intercept) 1.9133   1.3832
##  Residual             0.4949   0.7035
## Number of obs: 1072, groups:  newpid, 250
##
## Fixed effects:
##                                Estimate Std. Error t value
## (Intercept)                    5.093552   0.193091  26.379
## factor(time)0.205             -1.292134   0.667019  -1.937
## factor(time)0.209999999999999  0.214367   0.888625   0.241
## factor(time)0.213333333333333  0.138503   0.942856   0.147
## factor(time)0.213333333333334 -1.245963   0.942795  -1.322
## factor(time)0.215833333333332  1.618313   0.896572   1.805
## factor(time)0.215833333333334 -0.265597   0.843673  -0.315
## factor(time)0.216666666666667 -0.327638   0.802723  -0.408
## factor(time)0.218333333333334  0.127950   0.895394   0.143
## factor(time)0.219166666666667 -0.410890   0.852475  -0.482
## factor(time)0.221666666666667  0.247805   0.942822   0.263
## factor(time)0.224166666666666  1.701423   0.855625   1.989
## factor(time)0.224166666666667 -1.532244   0.628760  -2.437
## factor(time)0.226666666666667  1.407242   0.584977   2.406
## factor(time)0.227499999999999 -1.544964   0.886969  -1.742
## factor(time)0.2275             0.050140   0.460200   0.109
## factor(time)0.229999999999999 -0.324847   0.585278  -0.555
```

```
## factor(time)0.23              -0.098256   0.123445  -0.796
## factor(time)0.2325            -0.613329   0.397372  -1.543
## factor(time)0.233333333333333 -0.009334   0.838570  -0.011
## factor(time)0.235000000000001 -1.925107   0.798427  -2.411
## factor(time)0.235833333333333  0.047157   0.294522   0.160
## factor(time)0.235833333333334  0.201800   0.617894   0.327
## factor(time)0.2375             1.410038   0.887285   1.589
## factor(time)0.238333333333333 -0.277099   0.485627  -0.571
## factor(time)0.238333333333334  0.855297   0.816774   1.047
## factor(time)0.240833333333333 -0.228851   0.784205  -0.292
## factor(time)0.240833333333334  0.372320   0.591803   0.629
## factor(time)0.243333333333333 -0.530701   0.888705  -0.597
## factor(time)0.244166666666667  0.073106   0.478082   0.153
## factor(time)0.245833333333333  0.077782   0.432348   0.180
## factor(time)0.245833333333334 -0.237661   0.604154  -0.393
## factor(time)0.246666666666666 -0.553123   0.647008  -0.855
## factor(time)0.246666666666667  0.368539   0.854061   0.432
## factor(time)0.249166666666666  0.174936   0.385011   0.454
## factor(time)0.249166666666667 -0.477748   0.189891  -2.516
## factor(time)0.251666666666667  0.237187   0.429688   0.552
## factor(time)0.251666666666668  0.352242   0.803251   0.439
## factor(time)0.2525            -0.101240   0.942820  -0.107
## factor(time)0.254166666666667 -0.616649   0.840058  -0.734
## factor(time)0.255              0.322723   0.800218   0.403
## factor(time)0.256666666666667  0.298075   0.630792   0.473
## factor(time)0.257499999999999  0.091622   0.842749   0.109
## factor(time)0.2575             0.463916   0.630178   0.736
## factor(time)0.2625            -0.053095   0.842248  -0.063
## factor(time)0.265             -0.195879   0.844804  -0.232
## factor(time)0.265833333333333 -0.255151   0.838540  -0.304
## factor(time)0.268333333333333 -0.374167   0.590115  -0.634
## factor(time)0.268333333333334  0.093972   0.489715   0.192
## factor(time)0.2875             0.493046   0.524727   0.940
## factor(time)0.289999999999999 -0.770748   0.942891  -0.817
## factor(time)0.293333333333333 -0.342827   0.942861  -0.364
## factor(time)0.304166666666667  0.280144   0.888270   0.315
## factor(time)0.306666666666666 -0.322891   0.888118  -0.364
## factor(time)0.306666666666667  0.252771   0.585668   0.432
## factor(time)0.325833333333334 -0.165857   0.890475  -0.186
## factor(time)0.328333333333333 -0.956430   0.943009  -1.014
## factor(time)0.331666666666667 -0.007752   0.942953  -0.008
## factor(time)0.358333333333333  0.551850   0.856358   0.644
## factor(time)0.364166666666667 -0.018138   0.607220  -0.030
## factor(time)0.429166666666666 -0.309272   0.942856  -0.328
## factor(time)0.429166666666667 -0.346737   0.895911  -0.387
## factor(time)0.438333333333333 -0.921356   0.895515  -1.029
## factor(time)0.440833333333333 -0.103654   0.771890  -0.134
## factor(time)0.443333333333332  0.152679   0.847005   0.180
## factor(time)0.449166666666667 -0.032719   0.791948  -0.041
## factor(time)0.454166666666666  0.412911   0.896572   0.461
## factor(time)0.454166666666667  0.143984   0.855625   0.168
## factor(time)0.455             -1.434524   0.886969  -1.617
## factor(time)0.456666666666667  0.294270   0.942822   0.312
## factor(time)0.4575            -0.148945   0.490014  -0.304
```

```
## factor(time)0.459166666666667  0.272475   0.618656   0.440
## factor(time)0.459999999999999 -0.137135   0.458245  -0.299
## factor(time)0.46              -0.271257   0.170741  -1.589
## factor(time)0.460000000000001 -0.240391   0.313719  -0.766
## factor(time)0.462499999999999 -0.758341   0.480477  -1.578
## factor(time)0.4625             0.397775   0.411097   0.968
## factor(time)0.463333333333333 -0.726602   0.810202  -0.897
## factor(time)0.465             -0.871397   0.479905  -1.816
## factor(time)0.465833333333333  0.355263   0.585812   0.606
## factor(time)0.465833333333334 -0.552752   0.827826  -0.668
## factor(time)0.4675             1.614401   0.608492   2.653
## factor(time)0.468333333333335 -0.728727   0.852475  -0.855
## factor(time)0.470833333333333 -0.366554   0.463189  -0.791
## factor(time)0.470833333333334 -0.433132   0.607082  -0.713
## factor(time)0.473333333333333 -0.176119   0.619723  -0.284
## factor(time)0.473333333333334  0.462437   0.791417   0.584
## factor(time)0.474166666666666 -0.639233   0.819819  -0.780
## factor(time)0.474166666666667 -0.823716   0.802298  -1.027
## factor(time)0.475833333333333 -0.469903   0.802735  -0.585
## factor(time)0.475833333333334 -1.412621   0.576916  -2.449
## factor(time)0.476666666666667  0.620029   0.404078   1.534
## factor(time)0.479166666666666 -0.179506   0.579477  -0.310
## factor(time)0.479166666666667 -0.031073   0.251550  -0.124
## factor(time)0.481666666666666  0.100449   0.586169   0.171
## factor(time)0.481666666666667  0.264713   0.427007   0.620
## factor(time)0.484166666666667 -2.717399   0.805034  -3.376
## factor(time)0.485              0.909098   0.621346   1.463
## factor(time)0.487499999999999 -0.222888   0.769079  -0.290
## factor(time)0.4875             1.786911   0.820393   2.178
## factor(time)0.487500000000001  1.808756   0.803251   2.252
## factor(time)0.489999999999999  0.081538   0.854061   0.095
## factor(time)0.495             -0.156852   0.597513  -0.263
## factor(time)0.495833333333333 -0.503907   0.859101  -0.587
## factor(time)0.495833333333334 -0.061541   0.943207  -0.065
## factor(time)0.498333333333333 -0.557042   0.387244  -1.438
## factor(time)0.498333333333334 -0.571204   0.376738  -1.516
## factor(time)0.500833333333333 -0.260243   0.792828  -0.328
## factor(time)0.500833333333334  0.072409   0.812906   0.089
## factor(time)0.501666666666667 -1.210842   0.613996  -1.972
## factor(time)0.503333333333334  0.698344   0.838540   0.833
## factor(time)0.504166666666666  0.253028   0.517090   0.489
## factor(time)0.505833333333333 -1.572085   0.818311  -1.921
## factor(time)0.509166666666666 -0.361330   0.895394  -0.404
## factor(time)0.511666666666667 -1.041414   0.873097  -1.193
## factor(time)0.514166666666666  0.102916   0.622159   0.165
## factor(time)0.515              0.158761   0.942822   0.168
## factor(time)0.5175            -0.438584   0.383818  -1.143
## factor(time)0.517500000000001 -1.116373   0.820098  -1.361
## factor(time)0.533333333333333 -0.290568   0.642683  -0.452
## factor(time)0.533333333333334 -1.036047   0.865528  -1.197
## factor(time)0.534166666666667  0.200595   0.616180   0.326
## factor(time)0.536666666666667 -0.477109   0.436387  -1.093
## factor(time)0.555833333333333 -0.244567   0.614939  -0.398
## factor(time)0.558333333333333  1.765597   0.942776   1.873
```

```
## factor(time)0.564166666666667 -0.795695   0.892551  -0.891
## factor(time)0.575             -0.333839   0.632638  -0.528
## factor(time)0.580833333333333 -0.323665   0.943074  -0.343
## factor(time)0.5825             0.585045   0.943018   0.620
## factor(time)0.594166666666667 -0.471202   0.890549  -0.529
## factor(time)0.610833333333333 -1.146051   0.942861  -1.216
## factor(time)0.6375             1.740300   0.895599   1.943
## factor(time)0.648333333333333 -2.026702   0.887243  -2.284
## factor(time)0.651666666666666 -1.272276   0.873699  -1.456
## factor(time)0.6575            -1.106509   0.943130  -1.173
## factor(time)0.67              -0.405685   0.849442  -0.478
## factor(time)0.670833333333333 -0.983938   0.490981  -2.004
## factor(time)0.673333333333333  0.030663   0.942856   0.033
## factor(time)0.675833333333333 -0.032160   0.667980  -0.048
## factor(time)0.684166666666667  0.566554   0.628950   0.901
## factor(time)0.685              0.605481   0.837083   0.723
## factor(time)0.6875            -1.550321   0.606921  -2.554
## factor(time)0.689166666666666 -0.082805   0.888625  -0.093
## factor(time)0.689166666666667  0.149056   0.806546   0.185
## factor(time)0.69              -0.175690   0.159597  -1.101
## factor(time)0.6925             0.523619   0.552935   0.947
## factor(time)0.692500000000001  0.925629   0.834721   1.109
## factor(time)0.693333333333334 -0.410890   0.852475  -0.482
## factor(time)0.695              0.352319   0.843673   0.418
## factor(time)0.695833333333333 -0.326870   0.855293  -0.382
## factor(time)0.695833333333334 -1.629083   0.831279  -1.960
## factor(time)0.695833333333335  0.861029   0.594380   1.449
## factor(time)0.697500000000001 -1.458074   0.788234  -1.850
## factor(time)0.698333333333332  0.161046   0.836385   0.193
## factor(time)0.698333333333333 -0.618066   0.587940  -1.051
## factor(time)0.700833333333333  1.542814   0.485884   3.175
## factor(time)0.703333333333333 -0.888965   0.575976  -1.543
## factor(time)0.703333333333334 -1.064303   0.816774  -1.303
## factor(time)0.704166666666667 -4.782887   0.802298  -5.961
## factor(time)0.705833333333333 -0.468670   0.488070  -0.960
## factor(time)0.705833333333334 -0.419614   0.770478  -0.545
## factor(time)0.706666666666667  2.369534   0.797067   2.973
## factor(time)0.709166666666666 -0.136343   0.795902  -0.171
## factor(time)0.709166666666667  0.243583   0.276708   0.880
## factor(time)0.711666666666666  0.092769   0.839377   0.111
## factor(time)0.711666666666667 -0.485203   0.814432  -0.596
## factor(time)0.711666666666668 -0.841206   0.568230  -1.480
## factor(time)0.714166666666667 -1.382843   0.576246  -2.400
## factor(time)0.714999999999999 -0.704722   0.793005  -0.889
## factor(time)0.715000000000001 -0.599784   0.857813  -0.699
## factor(time)0.7175            -1.014832   0.805034  -1.261
## factor(time)0.72              -4.315189   0.779603  -5.535
## factor(time)0.725             -0.639835   0.599156  -1.068
## factor(time)0.725833333333332  0.352790   0.842749   0.419
## factor(time)0.725833333333333  0.729226   0.828534   0.880
## factor(time)0.725833333333334  0.325444   0.942822   0.345
## factor(time)0.728333333333333 -0.254339   0.289676  -0.878
## factor(time)0.730833333333333  0.051792   0.504796   0.103
## factor(time)0.733333333333333 -0.768564   0.842276  -0.912
```

```
## factor(time)0.734166666666666 -2.791350   0.805687  -3.465
## factor(time)0.735833333333333 -0.886552   0.666804  -1.330
## factor(time)0.736666666666666  0.047263   0.847005   0.056
## factor(time)0.736666666666667  1.509047   0.842005   1.792
## factor(time)0.7425            -0.552922   0.943070  -0.586
## factor(time)0.744166666666667  0.145617   0.573084   0.254
## factor(time)0.745             -0.420181   0.822511  -0.511
## factor(time)0.7475            -0.353974   0.365231  -0.969
## factor(time)0.752500000000001 -1.546654   0.838540  -1.844
## factor(time)0.758333333333334  0.202448   0.856358   0.236
## factor(time)0.761666666666667 -0.587183   0.807295  -0.727
## factor(time)0.763333333333333 -0.111766   0.814320  -0.137
## factor(time)0.763333333333335  0.018127   0.865528   0.021
## factor(time)0.764166666666666 -0.481204   0.859101  -0.560
## factor(time)0.765833333333333 -1.156161   0.873097  -1.324
## factor(time)0.766666666666667 -0.455389   0.385068  -1.183
## factor(time)0.775             -1.600095   0.809136  -1.978
## factor(time)0.78               0.770175   1.559216   0.494
## factor(time)0.783333333333333  1.024707   0.888270   1.154
## factor(time)0.785             -0.245883   0.828143  -0.297
## factor(time)0.785833333333333  0.492534   0.801169   0.615
## factor(time)0.788333333333333 -0.711614   0.942776  -0.755
## factor(time)0.794166666666667 -0.597034   0.816692  -0.731
## factor(time)0.8025            -0.569540   0.576253  -0.988
## factor(time)0.805             -1.567362   0.820098  -1.911
## factor(time)0.805000000000001  0.117510   0.788428   0.149
## factor(time)0.807500000000001 -0.347919   0.892551  -0.390
## factor(time)0.824166666666667  0.012834   0.632754   0.020
## factor(time)0.8625            -0.476116   0.607547  -0.784
## factor(time)0.8675             0.498449   0.895599   0.557
## factor(time)0.878333333333334 -0.147175   0.887243  -0.166
## factor(time)0.881666666666666  0.956310   0.942861   1.014
## factor(time)0.895833333333333 -0.706733   0.886969  -0.797
## factor(time)0.900833333333333 -0.499919   0.849446  -0.589
## factor(time)0.900833333333334  0.335472   0.583736   0.575
## factor(time)0.903333333333333 -1.014054   0.764566  -1.326
## factor(time)0.903333333333334  2.230417   0.942856   2.366
## factor(time)0.905833333333334  1.405521   0.896572   1.568
## factor(time)0.908333333333334 -0.346737   0.895911  -0.387
## factor(time)0.909166666666666  2.362936   0.943130   2.505
## factor(time)0.909166666666667 -0.321142   0.810202  -0.396
## factor(time)0.911666666666667 -0.292842   0.787125  -0.372
## factor(time)0.914166666666667  0.616320   0.585809   1.052
## factor(time)0.9175            -0.498101   0.837083  -0.595
## factor(time)0.919166666666667  0.044167   0.476171   0.093
## factor(time)0.919999999999998 -0.880261   0.785367  -1.121
## factor(time)0.92              -0.948462   0.266568  -3.558
## factor(time)0.920000000000001 -0.118121   0.461046  -0.256
## factor(time)0.9225            -0.493899   0.586445  -0.842
## factor(time)0.925833333333333 -1.695097   0.589778  -2.874
## factor(time)0.925833333333334  0.360968   0.834721   0.432
## factor(time)0.928333333333333  0.013803   0.786717   0.018
## factor(time)0.928333333333334 -0.883069   0.827826  -1.067
## factor(time)0.930833333333333 -0.733821   0.557137  -1.317
```

```
## factor(time)0.930833333333334 -1.958921   0.809318  -2.420
## factor(time)0.933333333333332  0.464232   0.824467   0.563
## factor(time)0.933333333333333 -0.013478   0.587594  -0.023
## factor(time)0.934166666666664 -0.710901   0.836385  -0.850
## factor(time)0.934166666666667 -1.047895   0.802298  -1.306
## factor(time)0.935833333333333 -0.540877   0.590517  -0.916
## factor(time)0.935833333333334 -0.730638   0.826065  -0.884
## factor(time)0.936666666666667  0.121388   0.785334   0.155
## factor(time)0.938333333333334  0.096023   0.803182   0.120
## factor(time)0.939166666666666 -0.190159   0.333277  -0.571
## factor(time)0.939166666666667  0.058213   0.330615   0.176
## factor(time)0.939166666666668 -1.676498   0.815358  -2.056
## factor(time)0.941666666666666 -0.264353   0.406437  -0.650
## factor(time)0.941666666666667 -0.513717   0.814432  -0.631
## factor(time)0.944166666666667  0.129977   0.590379   0.220
## factor(time)0.9475             0.658158   0.584757   1.126
## factor(time)0.952500000000001 -0.438147   0.791417  -0.554
## factor(time)0.955              1.552441   0.839377   1.850
## factor(time)0.955000000000001 -0.454099   0.812906  -0.559
## factor(time)0.955833333333333 -1.108780   0.847094  -1.309
## factor(time)0.9575             0.407155   0.788686   0.516
## factor(time)0.958333333333333  0.111838   0.361390   0.309
## factor(time)0.958333333333334 -0.129922   0.460003  -0.282
## factor(time)0.960833333333333 -1.477970   0.819054  -1.804
## factor(time)0.964166666666666 -0.012527   0.842749  -0.015
## factor(time)0.964166666666667  0.504631   0.828534   0.609
## factor(time)0.965833333333333  0.047263   0.847005   0.056
## factor(time)0.976666666666667 -0.503498   0.840058  -0.599
## factor(time)0.977499999999999 -3.205564   0.805687  -3.979
## factor(time)0.9775            -0.389680   0.361388  -1.078
## factor(time)0.977500000000001  0.599189   0.857813   0.699
## factor(time)0.982499999999999 -0.320085   0.820896  -0.390
## factor(time)0.9825            -0.115201   0.842276  -0.137
## factor(time)0.983333333333333 -0.028930   0.876641  -0.033
## factor(time)0.985833333333333 -0.482562   0.943070  -0.512
## factor(time)0.996666666666666 -0.568988   0.565726  -1.006
## factor(time)0.996666666666667 -0.930475   0.419337  -2.219
## factor(time)0.999166666666667 -0.970567   0.780275  -1.244
## factor(time)1                 -1.016832   0.838570  -1.213
## factor(time)1.00166666666667  -0.255151   0.838540  -0.304
## factor(time)1.0025            -1.296051   0.842005  -1.539
## factor(time)1.01083333333333  -0.315493   0.807295  -0.391
## factor(time)1.0125            -0.728607   0.814320  -0.895
## factor(time)1.01583333333333  -0.962893   0.604641  -1.593
## factor(time)1.02083333333333  -1.298261   0.851367  -1.525
## factor(time)1.02333333333333  -0.512261   0.895394  -0.572
## factor(time)1.0325             0.757370   0.888270   0.853
## factor(time)1.035             -0.371367   0.386338  -0.961
## factor(time)1.04833333333333  -1.228092   0.873097  -1.407
## factor(time)1.05333333333333  -0.147769   0.828143  -0.178
## factor(time)1.05416666666667  -0.545614   0.579549  -0.941
## factor(time)1.07583333333333  -1.036185   0.846072  -1.225
## factor(time)1.08666666666667   0.012300   1.559216   0.008
## factor(time)1.09              -1.588709   0.942861  -1.685
```

```
## factor(time)1.0925            -0.627146   0.499396  -1.256
## factor(time)1.11416666666667   0.141895   0.853404   0.166
## factor(time)1.11666666666667   0.956310   0.942861   1.014
## factor(time)1.13083333333333   0.275955   0.849446   0.325
## factor(time)1.13583333333333  -2.819862   0.942856  -2.991
## factor(time)1.13916666666667   0.099822   0.785334   0.127
## factor(time)1.14166666666667  -0.506914   0.574740  -0.882
## factor(time)1.14416666666667   1.110558   0.855625   1.298
## factor(time)1.145             -0.404614   0.886969  -0.456
## factor(time)1.1475            -1.354511   0.837083  -1.618
## factor(time)1.14916666666667  -0.682233   0.573739  -1.189
## factor(time)1.15              -0.399037   0.218855  -1.823
## factor(time)1.1525            -0.916084   0.457753  -2.001
## factor(time)1.15583333333333  -1.122385   0.489200  -2.294
## factor(time)1.1575            -0.123252   0.462317  -0.267
## factor(time)1.15833333333333  -1.596844   0.592974  -2.693
## factor(time)1.16083333333333  -1.031120   0.408666  -2.523
## factor(time)1.16333333333333   0.878301   0.593912   1.479
## factor(time)1.16416666666667   0.426656   0.836385   0.510
## factor(time)1.16583333333333  -1.601390   0.831168  -1.927
## factor(time)1.16833333333333  -0.253007   0.803182  -0.315
## factor(time)1.16916666666667  -0.517884   0.308196  -1.680
## factor(time)1.17166666666667  -1.400220   0.574056  -2.439
## factor(time)1.17666666666667  -1.305213   0.826065  -1.580
## factor(time)1.1775            -0.109843   0.802735  -0.137
## factor(time)1.18              -0.476278   0.795902  -0.598
## factor(time)1.18833333333333  -0.095017   0.219807  -0.432
## factor(time)1.19666666666667   0.328206   0.628384   0.522
## factor(time)1.20166666666667   0.079774   0.477936   0.167
## factor(time)1.20416666666667  -0.599325   0.592699  -1.011
## factor(time)1.20666666666667   0.319447   0.788686   0.405
## factor(time)1.2075            -1.051552   0.471721  -2.229
## factor(time)1.2125             0.070202   0.842276   0.083
## factor(time)1.22416666666667  -0.290789   0.857813  -0.339
## factor(time)1.22583333333333   0.018406   0.859156   0.021
## factor(time)1.22666666666667   0.016875   0.591859   0.029
## factor(time)1.22916666666667  -0.067404   0.588863  -0.114
## factor(time)1.23166666666667  -1.400130   0.802723  -1.744
## factor(time)1.2325            -0.114744   0.797067  -0.144
## factor(time)1.24583333333333  -1.076772   0.478915  -2.248
## factor(time)1.24833333333333  -1.650868   0.842005  -1.961
## factor(time)1.25333333333333  -0.930956   0.895394  -1.040
## factor(time)1.26166666666667  -0.398745   0.814320  -0.490
## factor(time)1.265             -0.193192   0.470869  -0.410
## factor(time)1.2675            -0.283020   0.838570  -0.338
## factor(time)1.28416666666667  -2.328037   0.859101  -2.710
## factor(time)1.3025            -0.631730   0.873097  -0.724
## factor(time)1.30333333333333  -0.332187   0.568895  -0.584
## factor(time)1.31166666666667  -1.892114   0.809136  -2.338
## factor(time)1.34166666666667  -1.112609   0.785311  -1.417
## factor(time)1.35              -0.541798   0.873699  -0.620
## factor(time)1.35833333333333  -1.670738   0.837083  -1.996
## factor(time)1.36              -0.082805   0.888625  -0.093
## factor(time)1.36083333333333  -0.060441   0.890475  -0.068
```

```
## factor(time)1.36583333333333  -0.780548   0.787125  -0.992
## factor(time)1.37166666666667  -0.363037   0.566179  -0.641
## factor(time)1.37416666666667  -0.630700   0.785367  -0.803
## factor(time)1.375             -0.798496   0.789251  -1.012
## factor(time)1.37666666666667  -3.037860   0.763020  -3.981
## factor(time)1.37916666666667   0.023653   0.778976   0.030
## factor(time)1.38              -0.400030   0.323647  -1.236
## factor(time)1.3825            -1.536614   0.549845  -2.795
## factor(time)1.38583333333333  -0.733306   0.591934  -1.239
## factor(time)1.38583333333334   0.222503   0.827826   0.269
## factor(time)1.3875            -0.199914   0.565499  -0.354
## factor(time)1.38833333333333  -1.431096   0.798678  -1.792
## factor(time)1.39083333333333  -0.558991   0.571358  -0.978
## factor(time)1.39583333333333  -0.585567   0.596986  -0.981
## factor(time)1.39666666666667   0.312198   0.766086   0.408
## factor(time)1.39833333333333  -0.335258   0.824211  -0.407
## factor(time)1.39916666666667  -0.466878   0.309593  -1.508
## factor(time)1.40166666666667  -2.183125   0.602541  -3.623
## factor(time)1.41              -1.290799   0.552447  -2.337
## factor(time)1.4125            -0.688106   0.826065  -0.833
## factor(time)1.415             -0.196216   0.574367  -0.342
## factor(time)1.41833333333333  -0.129626   0.455803  -0.284
## factor(time)1.42083333333333  -0.711614   0.942776  -0.755
## factor(time)1.42416666666667   0.406566   0.785596   0.518
## factor(time)1.42583333333333  -0.169894   0.812906  -0.209
## factor(time)1.42916666666667   0.646504   0.867074   0.746
## factor(time)1.43166666666667  -0.132489   0.582106  -0.228
## factor(time)1.43666666666667   0.457234   0.807064   0.567
## factor(time)1.4375            -0.901376   0.786071  -1.147
## factor(time)1.44583333333333  -3.368904   0.805687  -4.181
## factor(time)1.45333333333333  -0.760950   0.895911  -0.849
## factor(time)1.45416666666667  -0.423625   0.857813  -0.494
## factor(time)1.45583333333333  -0.541576   0.788686  -0.687
## factor(time)1.45666666666667  -0.146474   0.409964  -0.357
## factor(time)1.4625             0.323925   0.797067   0.406
## factor(time)1.47               0.173318   0.582584   0.297
## factor(time)1.4725             0.115333   0.820896   0.140
## factor(time)1.475              0.240710   0.842276   0.286
## factor(time)1.48166666666667  -1.520439   0.842005  -1.806
## factor(time)1.48333333333333  -0.541049   0.895394  -0.604
## factor(time)1.4925            -2.565285   0.802723  -3.196
## factor(time)1.495              0.199727   0.481772   0.415
## factor(time)1.4975             0.430515   0.628384   0.685
## factor(time)1.5               -0.279703   0.814320  -0.343
## factor(time)1.50583333333333  -0.896623   0.892713  -1.004
## factor(time)1.51416666666667   0.289027   0.788428   0.367
## factor(time)1.51666666666667  -1.964369   0.838570  -2.343
## factor(time)1.51916666666667  -3.165033   0.859101  -3.684
## factor(time)1.53               0.471822   0.844804   0.558
## factor(time)1.53083333333333   0.075976   0.888270   0.086
## factor(time)1.53333333333333  -0.110446   0.575733  -0.192
## factor(time)1.54166666666667  -0.937209   0.809136  -1.158
## factor(time)1.59083333333333  -1.272276   0.873699  -1.456
## factor(time)1.615             -0.744419   0.828143  -0.899
```

```
## factor(time)1.62916666666667   3.589487   0.800218   4.486
## factor(time)1.64833333333333  -1.572882   0.847094  -1.857
## factor(time)1.71666666666667   0.007283   0.843349   0.009
## factor(time)1.725             -0.335550   0.788428  -0.426
## factor(time)1.81166666666667   0.311681   0.844804   0.369
## factor(time)1.89666666666667  -0.411069   0.824211  -0.499
## factor(time)1.90833333333333  -0.752458   0.856358  -0.879
## factor(time)1.93833333333333  -0.853064   0.942776  -0.905
## factor(treatment)2             0.213927   0.186125   1.149
## age.baseline                  -0.124873   0.040918  -3.052

##
## Correlation matrix not shown by default, as p = 405 > 12.
## Use print(value, correlation=TRUE)  or
##     vcov(value)        if you need it
```

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

```
AIC(reg.4, model_varying_slopes, model_time_as_factor)
```

```
##                        df      AIC
## reg.4                   6 3149.209
## model_varying_slopes    8 3123.013
## model_time_as_factor  407 2980.115
```

```
BIC(reg.4, model_varying_slopes, model_time_as_factor)
```

```
##                        df      BIC
## reg.4                   6 3179.073
## model_varying_slopes    8 3162.831
## model_time_as_factor  407 5005.869
```

```
predictions <- data.frame(
  newpid = "new_child",
  time = rep(time_points, 3),
  treatment = 1,
  age.baseline = 0,
  model = rep(c("Original", "Varying Slopes", "Time as Factor"), each = length(time_points))
)
predictions$predicted_cd4_original <- predict(reg.4, newdata = predictions, re.form = NA)
predictions$predicted_cd4_varying <- predict(model_varying_slopes, newdata = predictions, re.form = NA)
predictions$predicted_cd4_factor <- predict(model_time_as_factor, newdata = predictions, re.form = NA)
ggplot(predictions, aes(x = time)) +
  geom_line(aes(y = predicted_cd4_original, color = "Original"), size = 1) +
  geom_line(aes(y = predicted_cd4_varying, color = "Varying Slopes"), size = 1) +
  geom_line(aes(y = predicted_cd4_factor, color = "Time as Factor"), size = 1) +
    scale_color_manual(values = c("Original" = "blue", "Varying Slopes" = "green", "Time as Factor" = "
    labs(
    title = "Predicted CD4 Trajectories for Different Models",
    x = "Time (Years since Baseline)",
    y = "Predicted CD4 Percentage (Square Root Scale)"
  )+
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

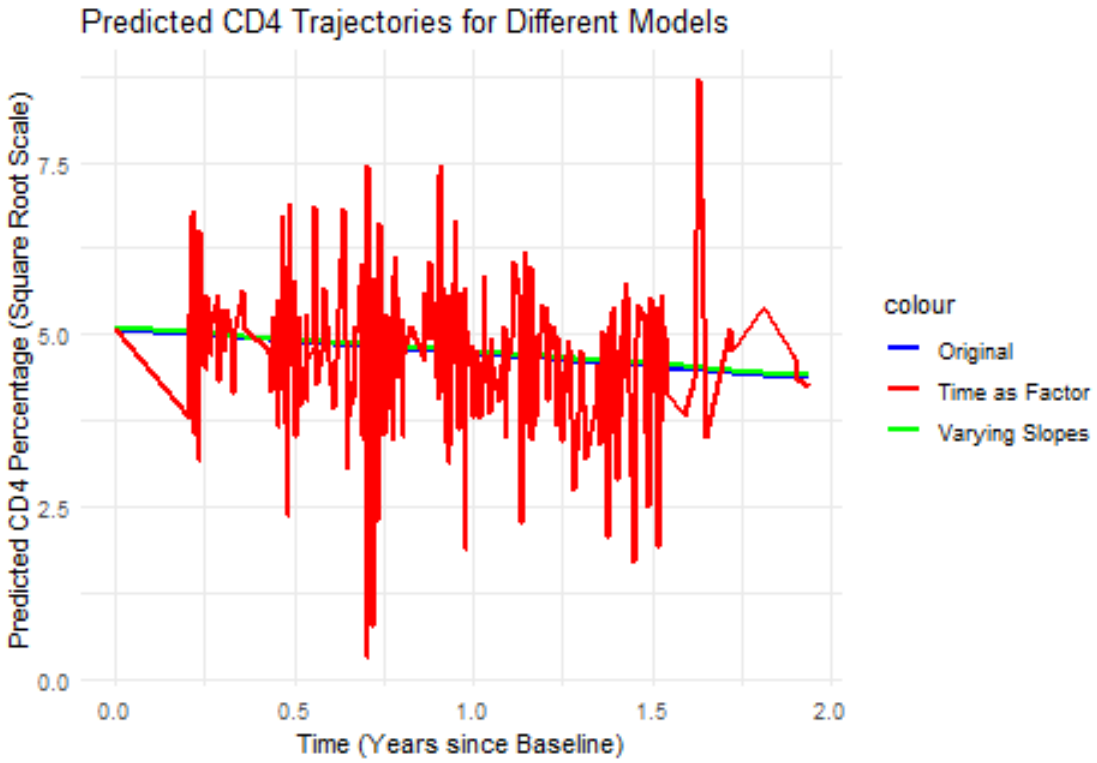


Figure skate in the 1932 Winter Olympics

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).

```
olym_array <- array(0, dim = c(7, 7, 2))
for (i in 1:7) {
  program_scores <- subset(olympics1932, pair == i & criterion == "Program")[, 3:9]
  performance_scores <- subset(olympics1932, pair == i & criterion == "Performance")[, 3:9]

  olym_array[i, , 1] <- as.numeric(program_scores)
  olym_array[i, , 2] <- as.numeric(performance_scores)
}

print(olym_array)
```

```
## , , 1
```

```
##
##      [,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
##
##      [,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. 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.

```r
reformulated_data <- list()
row_index <- 1
for (i in 1:nrow(olympics1932)) {
  for (j in 3:ncol(olympics1932)) {
    reformulated_data[[row_index]] <- c(
      technical_merit = ifelse(olympics1932$criterion[i] == "Program", olympics1932[i, j], NA),
      artistic_impression = ifelse(olympics1932$criterion[i] == "Performance", olympics1932[i, j], NA),
      skater_id = olympics1932$pair[i],
      judge_id = j - 2
    )
    row_index <- row_index + 1
  }
}
reformulated_df <- do.call(rbind, reformulated_data)
reformulated_df <- as.data.frame(reformulated_df, stringsAsFactors = FALSE)
print(reformulated_df)
```

```
##    technical_merit artistic_impression skater_id judge_id
## 1              5.6                  NA         1        1
## 2              5.5                  NA         1        2
## 3              5.8                  NA         1        3
## 4              5.3                  NA         1        4
## 5              5.6                  NA         1        5
## 6              5.2                  NA         1        6
## 7              5.7                  NA         1        7
## 8               NA                 5.6         1        1
## 9               NA                 5.5         1        2
## 10              NA                 5.8         1        3
```

```
## 11    NA  4.7  1  4
## 12    NA  5.7  1  5
## 13    NA  5.3  1  6
## 14    NA  5.4  1  7
## 15   5.5   NA  2  1
## 16   5.2   NA  2  2
## 17   5.8   NA  2  3
## 18   5.8   NA  2  4
## 19   5.6   NA  2  5
## 20   5.1   NA  2  6
## 21   5.8   NA  2  7
## 22    NA  5.5  2  1
## 23    NA  5.7  2  2
## 24    NA  5.6  2  3
## 25    NA  5.4  2  4
## 26    NA  5.5  2  5
## 27    NA  5.3  2  6
## 28    NA  5.7  2  7
## 29   6.0   NA  3  1
## 30   5.3   NA  3  2
## 31   5.8   NA  3  3
## 32   5.0   NA  3  4
## 33   5.4   NA  3  5
## 34   5.1   NA  3  6
## 35   5.3   NA  3  7
## 36    NA  6.0  3  1
## 37    NA  5.5  3  2
## 38    NA  5.7  3  3
## 39    NA  4.9  3  4
## 40    NA  5.5  3  5
## 41    NA  5.2  3  6
## 42    NA  5.7  3  7
## 43   5.6   NA  4  1
## 44   5.3   NA  4  2
## 45   5.8   NA  4  3
## 46   4.4   NA  4  4
## 47   4.5   NA  4  5
## 48   5.0   NA  4  6
## 49   5.1   NA  4  7
## 50    NA  5.6  4  1
## 51    NA  5.3  4  2
## 52    NA  5.8  4  3
## 53    NA  4.8  4  4
## 54    NA  4.5  4  5
## 55    NA  5.0  4  6
## 56    NA  5.5  4  7
## 57   5.4   NA  5  1
## 58   4.5   NA  5  2
## 59   5.8   NA  5  3
## 60   4.0   NA  5  4
## 61   5.5   NA  5  5
## 62   4.8   NA  5  6
## 63   5.5   NA  5  7
## 64    NA  4.8  5  1
```

```
## 65              NA                4.8         5         2
## 66              NA                5.5         5         3
## 67              NA                4.4         5         4
## 68              NA                4.6         5         5
## 69              NA                4.8         5         6
## 70              NA                5.2         5         7
## 71             5.2                 NA         6         1
## 72             5.1                 NA         6         2
## 73             5.3                 NA         6         3
## 74             5.4                 NA         6         4
## 75             4.5                 NA         6         5
## 76             4.5                 NA         6         6
## 77             5.0                 NA         6         7
## 78              NA                4.8         6         1
## 79              NA                5.6         6         2
## 80              NA                5.0         6         3
## 81              NA                4.7         6         4
## 82              NA                4.0         6         5
## 83              NA                4.6         6         6
## 84              NA                5.2         6         7
## 85             4.8                 NA         7         1
## 86             4.0                 NA         7         2
## 87             4.7                 NA         7         3
## 88             4.0                 NA         7         4
## 89             3.7                 NA         7         5
## 90             4.0                 NA         7         6
## 91             4.8                 NA         7         7
## 92              NA                4.3         7         1
## 93              NA                4.6         7         2
## 94              NA                4.5         7         3
## 95              NA                4.0         7         4
## 96              NA                3.6         7         5
## 97              NA                4.0         7         6
## 98              NA                4.8         7         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.

```
reformulated_df$SameCountry <-ifelse(reformulated_df[,3] == 1&reformulated_df[,4] ==5,1,
  ifelse(reformulated_df[,3] == 2&reformulated_df[,4] == 7,1,
  ifelse(reformulated_df[,3] == 3&reformulated_df[,4] == 1,1,
  ifelse(reformulated_df[,3] == 4&reformulated_df[,4] == 1,1,
  ifelse(reformulated_df[,3] == 7&reformulated_df[,4] == 7,1,0
  )))))
print(reformulated_df)
```

```
##    technical_merit artistic_impression skater_id judge_id SameCountry
## 1              5.6                  NA         1        1           0
## 2              5.5                  NA         1        2           0
## 3              5.8                  NA         1        3           0
## 4              5.3                  NA         1        4           0
## 5              5.6                  NA         1        5           1
## 6              5.2                  NA         1        6           0
```

```
## 7          5.7          NA        1        7          0
## 8           NA         5.6        1        1          0
## 9           NA         5.5        1        2          0
## 10          NA         5.8        1        3          0
## 11          NA         4.7        1        4          0
## 12          NA         5.7        1        5          1
## 13          NA         5.3        1        6          0
## 14          NA         5.4        1        7          0
## 15         5.5          NA        2        1          0
## 16         5.2          NA        2        2          0
## 17         5.8          NA        2        3          0
## 18         5.8          NA        2        4          0
## 19         5.6          NA        2        5          0
## 20         5.1          NA        2        6          0
## 21         5.8          NA        2        7          1
## 22          NA         5.5        2        1          0
## 23          NA         5.7        2        2          0
## 24          NA         5.6        2        3          0
## 25          NA         5.4        2        4          0
## 26          NA         5.5        2        5          0
## 27          NA         5.3        2        6          0
## 28          NA         5.7        2        7          1
## 29         6.0          NA        3        1          1
## 30         5.3          NA        3        2          0
## 31         5.8          NA        3        3          0
## 32         5.0          NA        3        4          0
## 33         5.4          NA        3        5          0
## 34         5.1          NA        3        6          0
## 35         5.3          NA        3        7          0
## 36          NA         6.0        3        1          1
## 37          NA         5.5        3        2          0
## 38          NA         5.7        3        3          0
## 39          NA         4.9        3        4          0
## 40          NA         5.5        3        5          0
## 41          NA         5.2        3        6          0
## 42          NA         5.7        3        7          0
## 43         5.6          NA        4        1          1
## 44         5.3          NA        4        2          0
## 45         5.8          NA        4        3          0
## 46         4.4          NA        4        4          0
## 47         4.5          NA        4        5          0
## 48         5.0          NA        4        6          0
## 49         5.1          NA        4        7          0
## 50          NA         5.6        4        1          1
## 51          NA         5.3        4        2          0
## 52          NA         5.8        4        3          0
## 53          NA         4.8        4        4          0
## 54          NA         4.5        4        5          0
## 55          NA         5.0        4        6          0
## 56          NA         5.5        4        7          0
## 57         5.4          NA        5        1          0
## 58         4.5          NA        5        2          0
## 59         5.8          NA        5        3          0
## 60         4.0          NA        5        4          0
```

```
## 61            5.5                NA        5        5           0
## 62            4.8                NA        5        6           0
## 63            5.5                NA        5        7           0
## 64             NA               4.8        5        1           0
## 65             NA               4.8        5        2           0
## 66             NA               5.5        5        3           0
## 67             NA               4.4        5        4           0
## 68             NA               4.6        5        5           0
## 69             NA               4.8        5        6           0
## 70             NA               5.2        5        7           0
## 71            5.2                NA        6        1           0
## 72            5.1                NA        6        2           0
## 73            5.3                NA        6        3           0
## 74            5.4                NA        6        4           0
## 75            4.5                NA        6        5           0
## 76            4.5                NA        6        6           0
## 77            5.0                NA        6        7           0
## 78             NA               4.8        6        1           0
## 79             NA               5.6        6        2           0
## 80             NA               5.0        6        3           0
## 81             NA               4.7        6        4           0
## 82             NA               4.0        6        5           0
## 83             NA               4.6        6        6           0
## 84             NA               5.2        6        7           0
## 85            4.8                NA        7        1           0
## 86            4.0                NA        7        2           0
## 87            4.7                NA        7        3           0
## 88            4.0                NA        7        4           0
## 89            3.7                NA        7        5           0
## 90            4.0                NA        7        6           0
## 91            4.8                NA        7        7           1
## 92             NA               4.3        7        1           0
## 93             NA               4.6        7        2           0
## 94             NA               4.5        7        3           0
## 95             NA               4.0        7        4           0
## 96             NA               3.6        7        5           0
## 97             NA               4.0        7        6           0
## 98             NA               4.8        7        7           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.tech <- reformulated_df[!is.na(reformulated_df$technical_merit), ]
reg.tech <- lmer(technical_merit ~ 1 + (1|skater_id) + (1|judge_id),data=data.tech)
summary(reg.tech)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: technical_merit ~ 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.

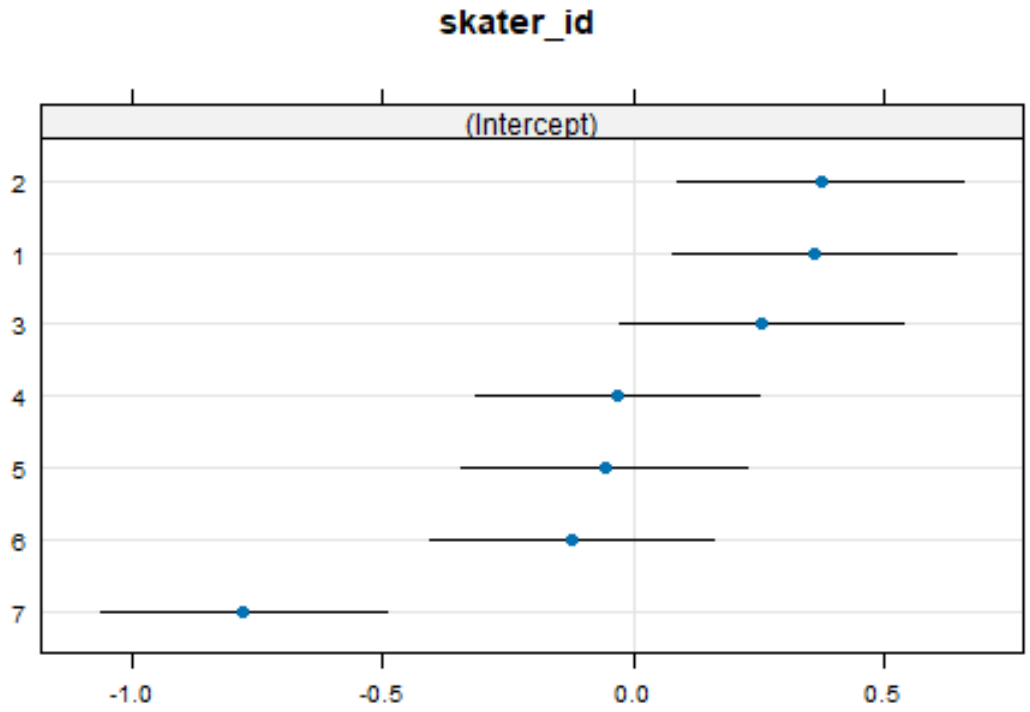
```
data.art <- reformulated_df[!is.na(reformulated_df$artistic_impression), ]
reg.art <- lmer(artistic_impression ~ 1 + (1|skater_id) + (1|judge_id),data=data.art)
summary(reg.art)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: artistic_impression ~ 1 + (1 | skater_id) + (1 | judge_id)
##    Data: data.art
##
## 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
```

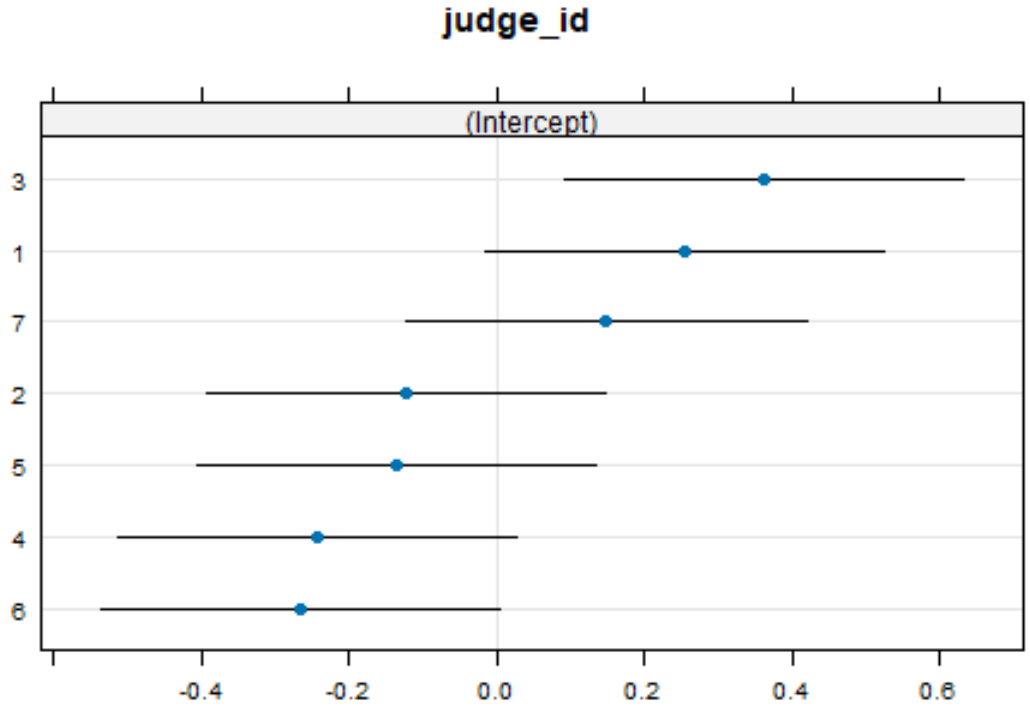
6. Display your results for both outcomes graphically.

```
dotplot(ranef(reg.tech, condVar = TRUE))
```

```
## $skater_id
```

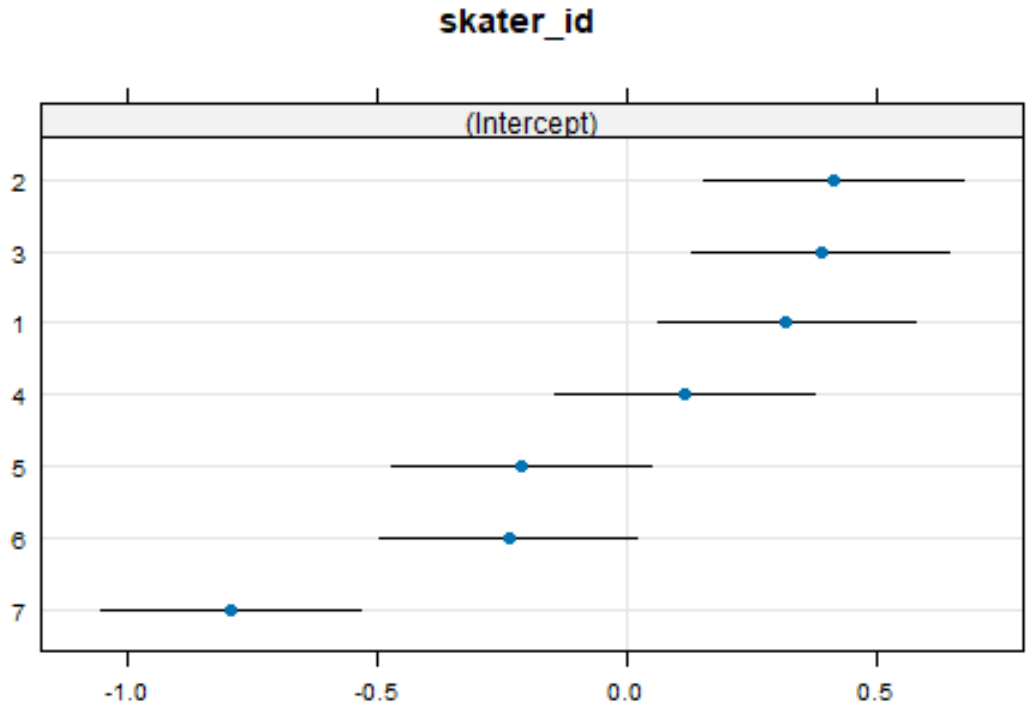

skater_id
(Intercept)
2
1
3
4
5
6
7
-1.0
-0.5
0.0
0.5

```
##
## $judge_id
```

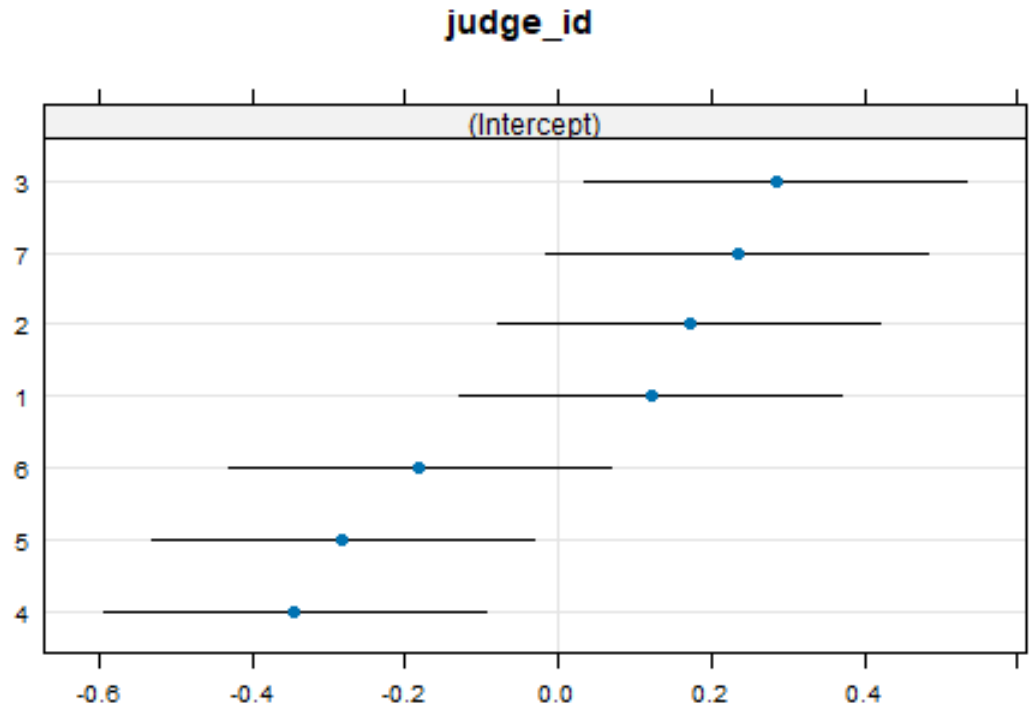

judge_id
(Intercept)
3
1
7
2
5
4
6
-0.4
-0.2
0.0
0.2
0.4
0.6

```
dotplot(ranef(reg.art, condVar = TRUE))
```

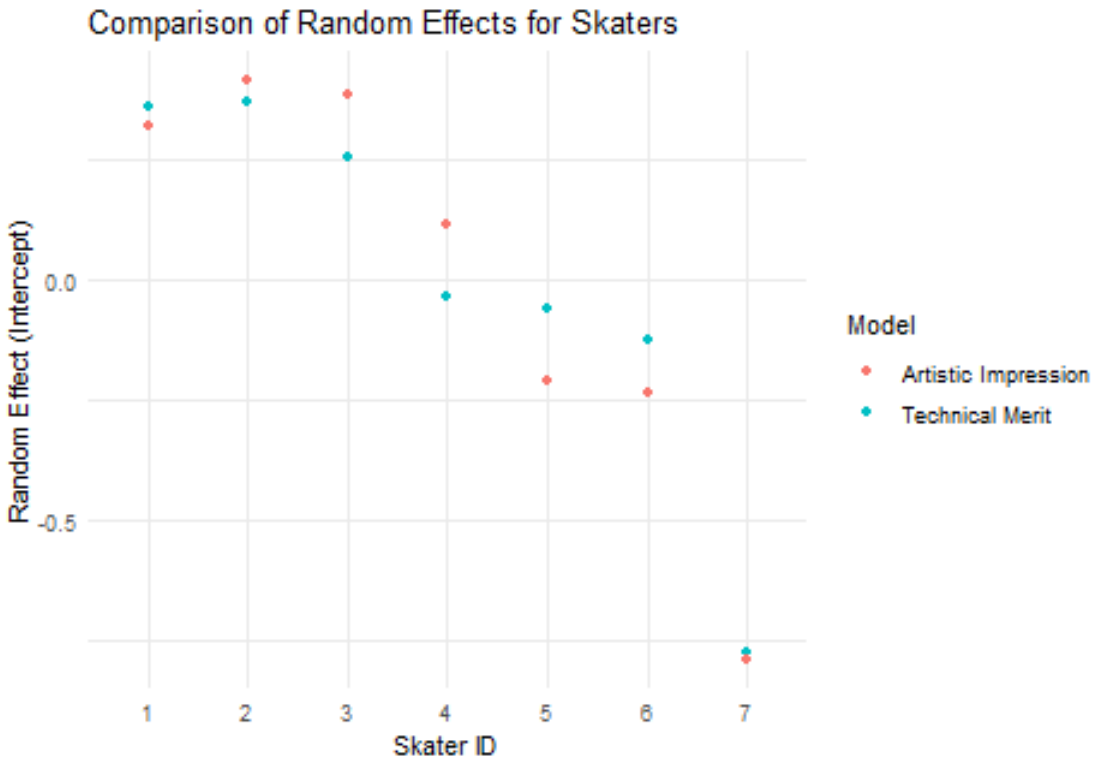
```
## $skater_id
```



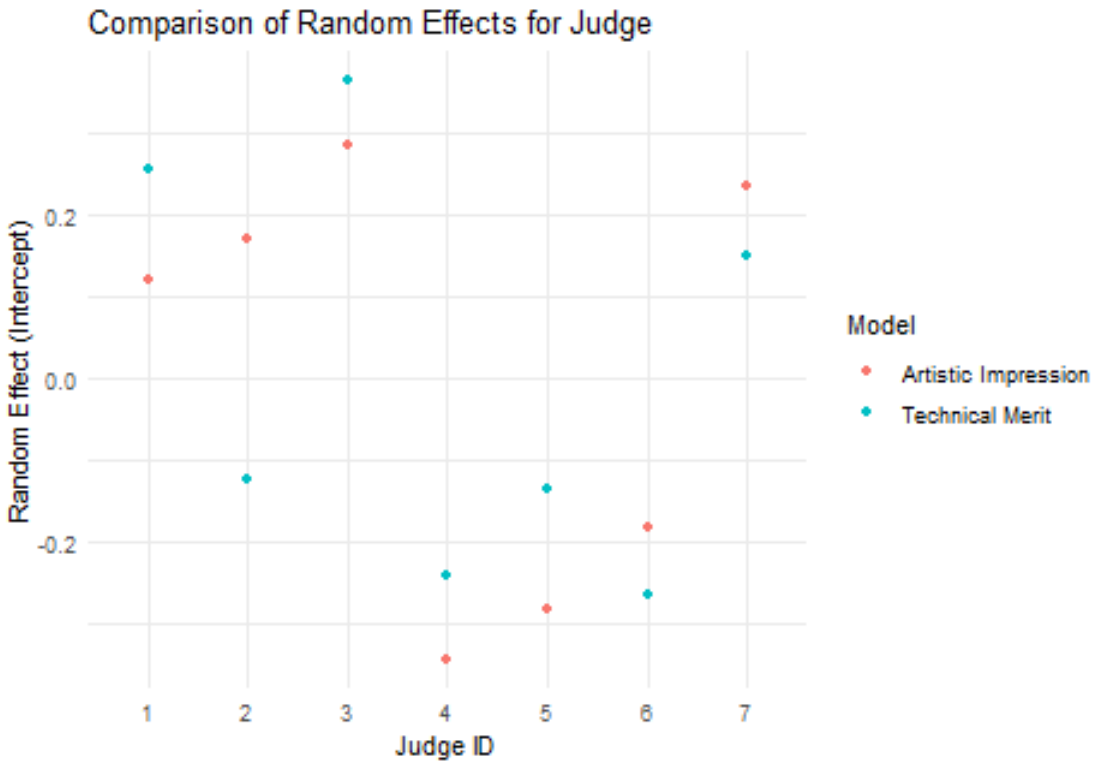

```
##
## $judge_id
```




```
re_tech <- ranef(reg.tech)$skater_id
re_tech$skater_id <- rownames(re_tech)
re_tech$model <- "Technical Merit"
re_art <- ranef(reg.art)$skater_id
re_art$skater_id <- rownames(re_art)
re_art$model <- "Artistic Impression"
combined_re <- rbind(re_tech, re_art)
ggplot(combined_re, aes(x = skater_id, y = `(Intercept)`, color = model)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Comparison of Random Effects for Skaters",
       x = "Skater ID", y = "Random Effect (Intercept)",
       color = "Model")
```




```
re_tech1 <- ranef(reg.tech)$judge_id
re_tech1$judge_id <- rownames(re_tech1)
re_tech1$model <- "Technical Merit"
re_art1 <- ranef(reg.art)$judge_id
re_art1$judge_id <- rownames(re_art1)
re_art1$model <- "Artistic Impression"
combined_re1 <- rbind(re_tech1, re_art1)
ggplot(combined_re1, aes(x = judge_id, y = `(Intercept)`, color = model)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Comparison of Random Effects for Judge",
       x = "Judge ID", y = "Random Effect (Intercept)",
       color = "Model")
```



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).

$$y_{score} = \alpha_{j[i]} + \beta_{cadidate} X_{iCadidate} + \beta_{rater} X_{iRater} + U_{RandomEffect-Rater}$$

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

lmer(rating~applicants+raters+(1+raters|raters))

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.

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

```
##
## Call:
## glm(formula = match ~ attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o, family = binomial, data = dating)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.62091    0.21859 -25.714  < 2e-16 ***
## attr_o       0.22047    0.02388   9.233  < 2e-16 ***
## sinc_o      -0.01996    0.03067  -0.651   0.5152
## intel_o      0.07176    0.03716   1.931   0.0535 .
## fun_o        0.25315    0.02922   8.665  < 2e-16 ***
## amb_o       -0.12099    0.02838  -4.264 2.01e-05 ***
## shar_o       0.21225    0.02209   9.608  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6466.6  on 7030  degrees of freedom
## Residual deviance: 5611.0  on 7024  degrees of freedom
##   (1347 observations deleted due to missingness)
## AIC: 5625
##
## Number of Fisher Scoring iterations: 5
```

Attractiveness (attr_o), fun (fun_o), and shared interests (shar_o) are all highly significant and have positive effects on the probability of a match. These characteristics are the most important predictors in the model. Ambition (amb_o) is also significant but has a negative effect, suggesting that higher ambition may reduce the likelihood of a match. Intelligence (intel_o) has a positive but weak effect, with borderline statistical significance. Sincerity (sinc_o) appears not significant, implying that it has little effect on predicting a match in this particular dataset.

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.038349 (tol = 0.002, component 1)
```

```r
summary(dating_pooled_1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: match ~ gender + attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o + (1 | iid)
##    Data: dating
##
##      AIC      BIC   logLik deviance df.resid
##   5543.3   5605.0  -2762.6   5525.3     7022
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7416 -0.4458 -0.2883 -0.1459 10.3426
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  iid    (Intercept) 0.4268   0.6533
## Number of obs: 7031, groups:  iid, 551
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.02085    0.24369 -24.707  < 2e-16 ***
## gender       0.15329    0.09314   1.646   0.0998 .
## attr_o       0.23540    0.02648   8.888  < 2e-16 ***
## sinc_o      -0.01372    0.03261  -0.421   0.6740
## intel_o      0.07019    0.03967   1.770   0.0768 .
## fun_o        0.26270    0.03140   8.366  < 2e-16 ***
## amb_o       -0.13138    0.03025  -4.343  1.4e-05 ***
## shar_o       0.22389    0.02325   9.629  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) gender attr_o sinc_o intel_ fun_o  amb_o
## gender  -0.192
## attr_o  -0.264  0.109
## sinc_o  -0.163  0.048 -0.120
## intel_o -0.298 -0.055 -0.039 -0.466
## fun_o   -0.112  0.017 -0.246 -0.151 -0.128
## amb_o   -0.038 -0.092 -0.061 -0.015 -0.372 -0.187
## shar_o  -0.056  0.010 -0.099 -0.053 -0.003 -0.265 -0.201
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.038349 (tol = 0.002, component 1)
```

Attractiveness, fun, and shared interests are consistently important for predicting match outcomes. Ambition negatively affects the probability of a match, while intelligence has a weak positive effect. Random effects account for the individual differences in willingness to meet again, which improves model fit.

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

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

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.663627 (tol = 0.002, component 1)
```

```r
summary(dating_pooled_2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: match ~ gender + attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o + (1 | iid) + (1 | pid)
##    Data: dating
##
##      AIC      BIC   logLik deviance df.resid
##   5257.8   5326.4  -2618.9   5237.8     7021
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6959 -0.3825 -0.2195 -0.0921  9.2386
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  iid    (Intercept) 0.6035   0.7769
##  pid    (Intercept) 1.2463   1.1164
## Number of obs: 7031, groups:  iid, 551; pid, 537
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.25462    0.38237 -21.588  < 2e-16 ***
## gender       0.17227    0.14932   1.154  0.24862
## attr_o       0.33684    0.03280  10.270  < 2e-16 ***
## sinc_o       0.02007    0.03896   0.515  0.60642
## intel_o      0.10492    0.04741   2.213  0.02690 *
## fun_o        0.30070    0.03636   8.271  < 2e-16 ***
## amb_o       -0.09327    0.03601  -2.590  0.00959 **
## shar_o       0.26014    0.02844   9.148  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) gender attr_o sinc_o intel_ fun_o  amb_o
## gender  -0.195
## attr_o  -0.336  0.093
## sinc_o  -0.220  0.036 -0.064
## intel_o -0.301 -0.045 -0.025 -0.438
## fun_o   -0.154  0.009 -0.215 -0.123 -0.099
## amb_o   -0.125 -0.071 -0.051  0.011 -0.334 -0.168
## shar_o  -0.100  0.004 -0.075 -0.059 -0.021 -0.237 -0.159
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.663627 (tol = 0.002, component 1)
```

Attractiveness, fun, and shared interests are still the most important traits, significantly increasing the probability of a match. Ambition has a negative effect, while intelligence shows a positive but weaker effect. Gender and sincerity do not significantly influence match outcomes. Including random intercepts for both evaluators and ratees substantially improves model fit by capturing individual-level differences, providing a more nuanced understanding of the factors driving matches.

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.)

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

```
## 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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: 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
```

```
head(dating_no_pool_list,10)
```

```
## [[1]]
##               Estimate Std. Error       z value  Pr(>|z|)
## (Intercept) -659.07830  906455.16 -0.0007270942 0.9994199
## attr_o        45.50864   78057.29  0.0005830160 0.9995348
## shar_o        45.45377   76594.10  0.0005934369 0.9995265
##
## [[2]]
##               Estimate Std. Error    z value  Pr(>|z|)
## (Intercept) -4.5571451  4.7236713 -0.9647464 0.3346718
## attr_o      -0.1083460  0.8580162 -0.1262750 0.8995142
## shar_o       0.5671291  0.7420286  0.7642954 0.4446912
##
## [[3]]
##                   Estimate Std. Error       z value  Pr(>|z|)
## (Intercept) -2.456607e+01  187472.89 -1.310380e-04 0.9998954
## attr_o       4.423449e-15   43072.25  1.026984e-19 1.0000000
## shar_o      -4.455435e-15   34821.85 -1.279494e-19 1.0000000
##
## [[4]]
##               Estimate Std. Error       z value  Pr(>|z|)
## (Intercept) -69.95336  392473.01 -0.0001782374 0.9998578
## attr_o      -46.51372   97293.16 -0.0004780780 0.9996185
## shar_o       46.51120   81318.88  0.0005719606 0.9995436
##
## [[5]]
##               Estimate Std. Error   z value  Pr(>|z|)
## (Intercept) -4.478098  3.5106715 -1.275567 0.2021085
## attr_o      -1.005813  0.8767464 -1.147211 0.2512944
## shar_o       1.274231  0.9215475  1.382707 0.1667546
##
## [[6]]
##               Estimate Std. Error    z value  Pr(>|z|)
## (Intercept)  3.970976  11.294264  0.3515923 0.7251441
## attr_o      -2.393118   2.372022 -1.0088938 0.3130255
## shar_o       1.494276   1.116957  1.3378098 0.1809585
##
## [[7]]
##               Estimate Std. Error       z value  Pr(>|z|)
## (Intercept) -24.35904  259464.89 -9.388184e-05 0.9999251
```

```
## attr_o      -47.09034   84114.89 -5.598336e-04 0.9995533
## shar_o       47.17954   74410.51  6.340441e-04 0.9994941
##
## [[8]]
##               Estimate Std. Error      z value  Pr(>|z|)
## (Intercept) -209.95073 104678.265 -0.002005676 0.9983997
## attr_o        19.88310  12601.222  0.001577871 0.9987410
## shar_o        10.17719   5638.896  0.001804819 0.9985600
##
## [[9]]
##               Estimate Std. Error       z value  Pr(>|z|)
## (Intercept) -790.55336  823825.65 -0.0009596125 0.9992343
## attr_o        45.56010   51057.55  0.0008923283 0.9992880
## shar_o        89.89867   93998.98  0.0009563792 0.9992369
##
## [[10]]
##                 Estimate   Std. Error      z value  Pr(>|z|)
## (Intercept) -125.8438842 3.704808e+04 -0.003396772 0.9972898
## attr_o        17.6958003 5.292583e+03  0.003343509 0.9973323
## shar_o         0.1500004 7.717392e-01  0.194366635 0.8458888
```

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

```
dating_pooled_3 <- glmer(match ~ gender + attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o+(1 + attr_
```

```
## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 1 from bobyqa: bobyqa -- maximum number of function
## evaluations exceeded
```

```
## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations
```

```
## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 4 from Nelder_Mead: failure to converge in 10000 evaluations
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```
summary(dating_pooled_3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: match ~ gender + attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o + (1 + attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o | iid)
##    Data: dating
```

```
##
##      AIC      BIC   logLik deviance df.resid
##   5577.1   5824.0  -2752.6   5505.1     6995
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8945 -0.4260 -0.2779 -0.1410  7.2408
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  iid    (Intercept) 0.50962  0.7139
##         attr_o      0.02611  0.1616    0.82
##         sinc_o      0.02628  0.1621   -0.14 -0.25
##         intel_o     0.01034  0.1017   -0.53 -0.88  0.38
##         fun_o       0.03805  0.1951   -0.40 -0.26 -0.58 -0.16
##         amb_o       0.01802  0.1342    0.09  0.20 -0.38  0.03 -0.40
##         shar_o      0.01594  0.1262   -0.70 -0.27  0.28 -0.09  0.34 -0.34
## Number of obs: 7031, groups:  iid, 551
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.17079    0.27896 -22.121  < 2e-16 ***
## gender       0.12396    0.09672   1.282 0.199987
## attr_o       0.22955    0.03082   7.447 9.53e-14 ***
## sinc_o      -0.03231    0.03847  -0.840 0.400972
## intel_o      0.08420    0.04549   1.851 0.064206 .
## fun_o        0.29793    0.03996   7.456 8.95e-14 ***
## amb_o       -0.13405    0.03632  -3.691 0.000224 ***
## shar_o       0.21808    0.02796   7.798 6.27e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) gender attr_o sinc_o intel_ fun_o  amb_o
## gender  -0.180
## attr_o  -0.151  0.056
## sinc_o  -0.147  0.056 -0.097
## intel_o -0.306 -0.049 -0.101 -0.410
## fun_o   -0.177  0.048 -0.247 -0.216 -0.138
## amb_o   -0.024 -0.100 -0.048 -0.030 -0.320 -0.250
## shar_o  -0.087  0.022 -0.136 -0.083 -0.022 -0.108 -0.236
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
## failure to converge in 10000 evaluations
```

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.

Complete-pooling model captures individual preferences but may be unreliable due to the small sample size for each person. No-pooling model captures individual preferences but may be unreliable due to the small sample size for each person. Multilevel model provides more stable estimates than the no-pooling model while accounting for individual variability in how attributes are valued. The random effects allow you to see how different raters prioritize different attributes.