homework 07

Name

November 1, 2018

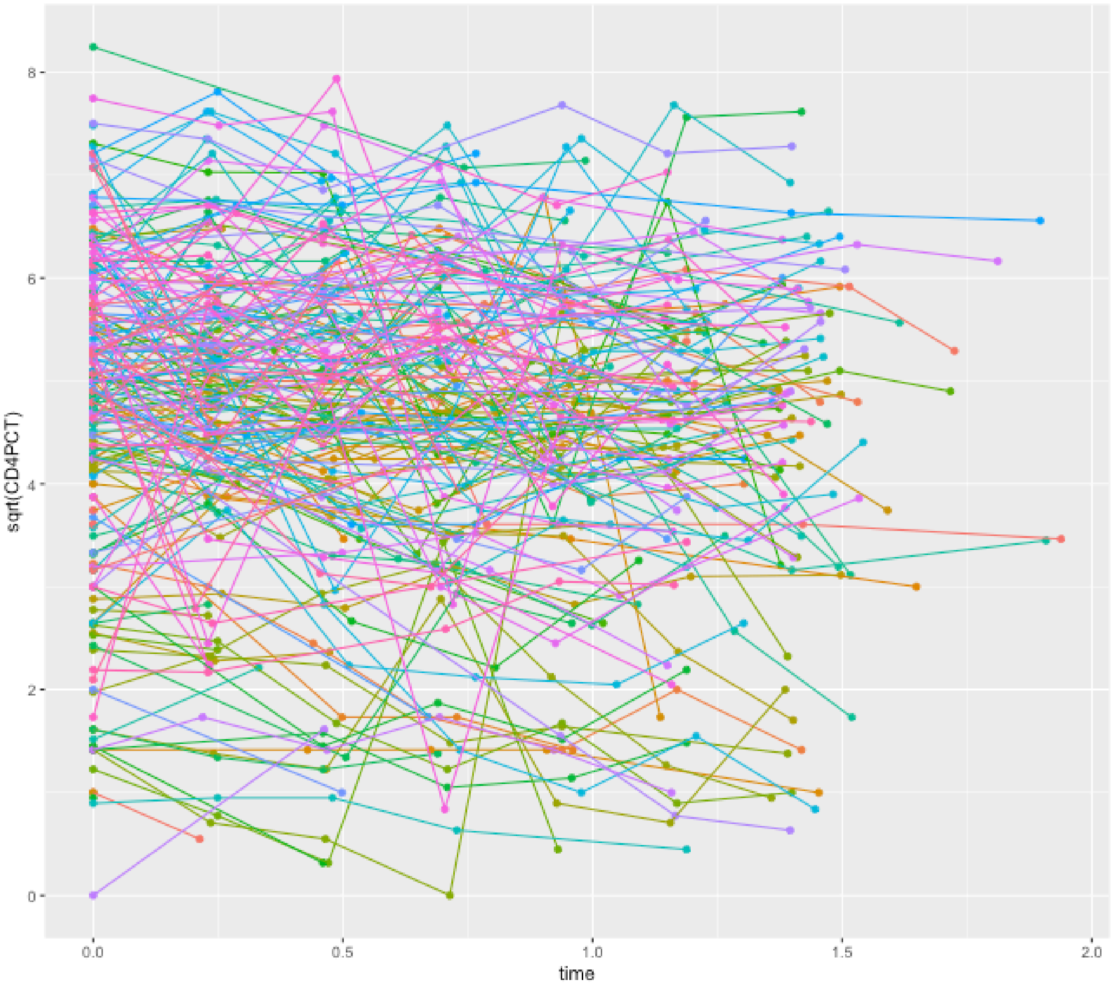
Data analysis

CD4 percentages for HIV infected kids

The folder `cd4` has CD4 percentages for a set of young children with HIV who were measured several times over a period of two years. The dataset also includes the ages of the children at each measurement.

1. Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

```
hiv.data<-na.omit(hiv.data)
ggplot(aes(x=time,y=sqrt(CD4PCT),color= factor(newpid)),data=hiv.data) + geom_point() + geom_line() + th
```



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

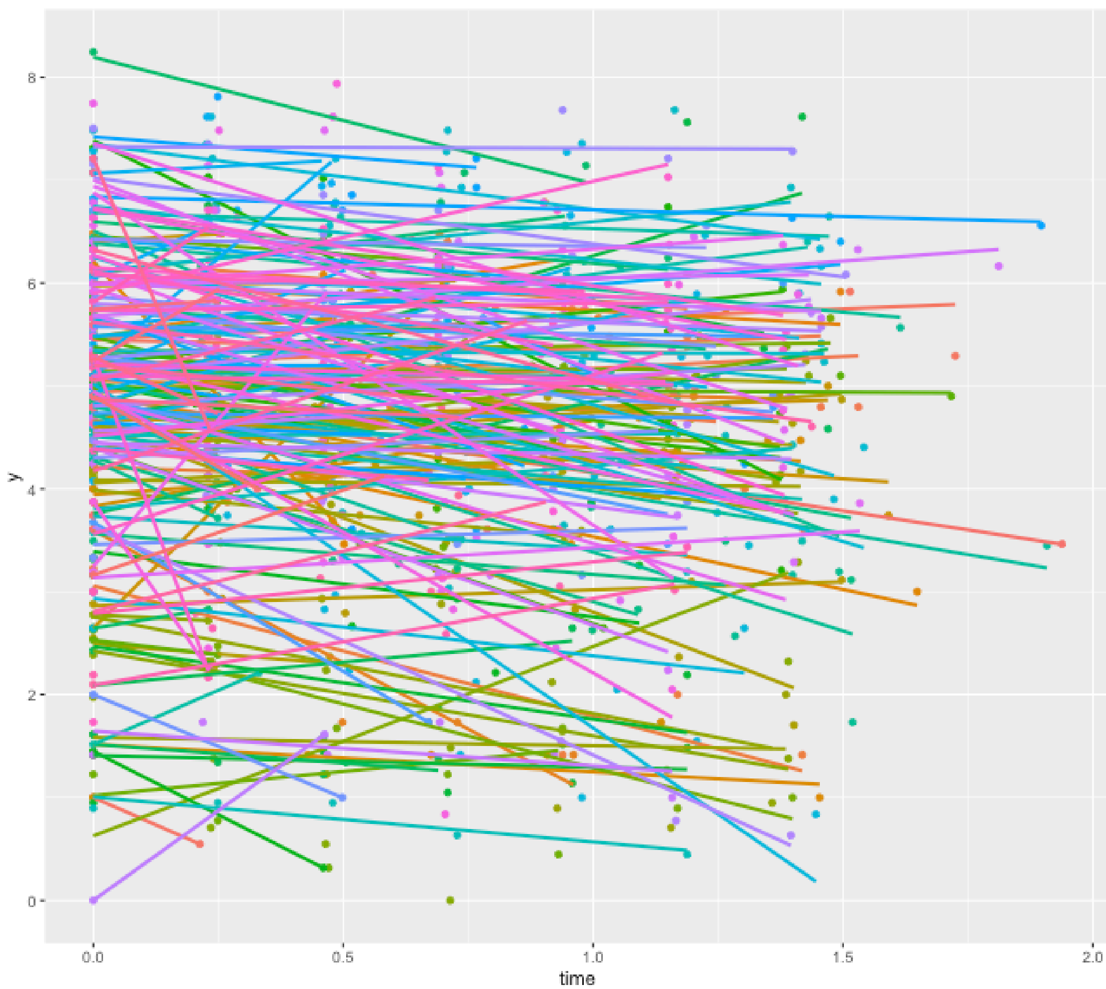
```r
#no pooling
model1<-lm(y~factor(newpid) -1 + time, data = hiv.data)
coef(model1)
```

```
##   factor(newpid)1   factor(newpid)2   factor(newpid)3   factor(newpid)4
##         4.5778674         0.8166729         5.9627893         5.6227401
##   factor(newpid)5   factor(newpid)6   factor(newpid)7   factor(newpid)8
##         4.0000000         5.3799806         5.6298435         5.1579338
##   factor(newpid)9  factor(newpid)11  factor(newpid)12  factor(newpid)13
##         6.2271324         2.4555459         4.3728760         5.3461569
##  factor(newpid)15  factor(newpid)16  factor(newpid)17  factor(newpid)18
##         5.2510840         2.4073179         6.1132998         6.0328538
##  factor(newpid)19  factor(newpid)20  factor(newpid)21  factor(newpid)22
##         4.1186510         5.0133408         5.0000000         6.1644140
##  factor(newpid)23  factor(newpid)24  factor(newpid)25  factor(newpid)26
##         1.6096466         4.8289285         4.7713542         4.6414515
##  factor(newpid)27  factor(newpid)28  factor(newpid)29  factor(newpid)30
##         4.3938479         5.6631989         4.5370888         1.0000000
##  factor(newpid)31  factor(newpid)32  factor(newpid)33  factor(newpid)34
##         4.4635186         4.6623258         5.0457230         6.4936075
##  factor(newpid)35  factor(newpid)38  factor(newpid)39  factor(newpid)40
##         4.9465337         6.1656207         4.8543692         3.6055513
##  factor(newpid)41  factor(newpid)42  factor(newpid)43  factor(newpid)44
##         5.0000000         3.2723948         4.9460030         2.4946444
##  factor(newpid)45  factor(newpid)46  factor(newpid)47  factor(newpid)48
##         5.1743719         3.5126686         4.8714644         4.4634394
##  factor(newpid)49  factor(newpid)50  factor(newpid)51  factor(newpid)52
##         5.4091643         4.3381169         3.9528535         1.8076757
##  factor(newpid)53  factor(newpid)54  factor(newpid)55  factor(newpid)56
##         4.8258783         4.4795522         2.3878391         2.7937471
##  factor(newpid)57  factor(newpid)58  factor(newpid)59  factor(newpid)61
##         2.1593161         2.0254072         5.1381910         5.2478803
##  factor(newpid)62  factor(newpid)63  factor(newpid)64  factor(newpid)65
##         5.6693731         1.9357830         5.4327034         1.4283518
##  factor(newpid)66  factor(newpid)67  factor(newpid)68  factor(newpid)69
##         6.4702137         2.5086474         5.8736701         5.3937991
##  factor(newpid)70  factor(newpid)71  factor(newpid)72  factor(newpid)74
##         5.0550094         2.6457513         3.8019804         5.1632484
##  factor(newpid)75  factor(newpid)76  factor(newpid)78  factor(newpid)79
##         5.8481297         4.9334932         6.0032139         4.9064905
##  factor(newpid)81  factor(newpid)82  factor(newpid)83  factor(newpid)84
##         0.9749953         3.2670178         0.9486833         2.2685773
##  factor(newpid)85  factor(newpid)86  factor(newpid)87  factor(newpid)88
##         1.5996833         6.4494141         6.1078775         4.8348380
##  factor(newpid)89  factor(newpid)90  factor(newpid)91  factor(newpid)93
##         5.0279712         5.8480766         2.5572348         1.5297026
##  factor(newpid)94  factor(newpid)95  factor(newpid)96  factor(newpid)97
##         4.9553806         2.7832455         4.8989795         7.7174633
##  factor(newpid)98  factor(newpid)99 factor(newpid)100 factor(newpid)101
##         4.7958315         6.5945750         6.5528328         5.6568542
## factor(newpid)104 factor(newpid)105 factor(newpid)106 factor(newpid)107
##         3.5680793         4.6798277         3.8071522         5.7979370
## factor(newpid)109 factor(newpid)110 factor(newpid)112 factor(newpid)113
##         4.0463452         5.3341657         4.0582707         6.3554248
```

```
## factor(newpid)116 factor(newpid)117 factor(newpid)118 factor(newpid)119
##          4.2623783          4.4257320          5.3215109          1.9316435
## factor(newpid)120 factor(newpid)121 factor(newpid)124 factor(newpid)126
##          6.8452484          6.1389160          3.1622777          4.4922540
## factor(newpid)127 factor(newpid)128 factor(newpid)129 factor(newpid)131
##          5.2631554          4.7623070          0.9866116          4.2655313
## factor(newpid)132 factor(newpid)133 factor(newpid)134 factor(newpid)135
##          4.7490697          3.7838142          6.7357827          5.6187174
## factor(newpid)136 factor(newpid)137 factor(newpid)138 factor(newpid)139
##          6.6605757          5.6839449          7.4833148          4.8636167
## factor(newpid)140 factor(newpid)141 factor(newpid)142 factor(newpid)143
##          5.4836980          7.1786924          2.8339836          2.8919592
## factor(newpid)144 factor(newpid)145 factor(newpid)146 factor(newpid)147
##          6.0589466          5.5603552          5.4722189          6.1902277
## factor(newpid)148 factor(newpid)149 factor(newpid)150 factor(newpid)151
##          5.3475360          5.6772848          4.3997551          5.6932329
## factor(newpid)152 factor(newpid)153 factor(newpid)154 factor(newpid)155
##          4.6151923          7.2174723          5.7176598          6.2742749
## factor(newpid)156 factor(newpid)157 factor(newpid)158 factor(newpid)159
##          6.3461388          6.4167536          6.0959660          5.3031815
## factor(newpid)160 factor(newpid)161 factor(newpid)162 factor(newpid)163
##          5.0490022          5.1470057          4.6970999          7.4258817
## factor(newpid)164 factor(newpid)165 factor(newpid)166 factor(newpid)167
##          7.0879246          4.4113960          5.6525188          4.9412759
## factor(newpid)168 factor(newpid)169 factor(newpid)170 factor(newpid)171
##          5.8105385          2.8377653          4.5294184          6.7082039
## factor(newpid)172 factor(newpid)173 factor(newpid)174 factor(newpid)175
##          5.2771029          1.6000053          3.8177552          5.8757897
## factor(newpid)176 factor(newpid)177 factor(newpid)178 factor(newpid)179
##          5.7184992          4.6630738          6.6511617          5.4320916
## factor(newpid)181 factor(newpid)182 factor(newpid)184 factor(newpid)185
##          7.5928045          6.8832553          4.6904158          5.3327255
## factor(newpid)186 factor(newpid)187 factor(newpid)188 factor(newpid)189
##          2.2775956          5.9730167          5.6564711          0.8990401
## factor(newpid)191 factor(newpid)192 factor(newpid)194 factor(newpid)196
##          4.7345753          4.6453538          1.6826857          4.2974059
## factor(newpid)198 factor(newpid)199 factor(newpid)200 factor(newpid)201
##          6.1260746          3.5907409          6.3438603          4.9002543
## factor(newpid)203 factor(newpid)204 factor(newpid)205 factor(newpid)206
##          6.3268453          5.4475963          3.6737075          5.9995784
## factor(newpid)207 factor(newpid)208 factor(newpid)209 factor(newpid)210
##          6.0930789          4.1806671          6.4394679          5.2218418
## factor(newpid)212 factor(newpid)213 factor(newpid)214 factor(newpid)215
##          5.2241669          4.6796107          6.5524917          5.0533475
## factor(newpid)216 factor(newpid)217 factor(newpid)218 factor(newpid)219
##          3.7566805          3.1011837          4.7787566          5.4772256
## factor(newpid)220 factor(newpid)222 factor(newpid)223 factor(newpid)224
##          6.3554030          5.2839068          5.3572043          3.8099416
## factor(newpid)225 factor(newpid)226 factor(newpid)227 factor(newpid)228
##          6.4843988          6.8615075          6.2248228          4.6817961
## factor(newpid)229 factor(newpid)230 factor(newpid)231 factor(newpid)232
##          5.2655656          5.9698426          5.9603842          6.1815757
## factor(newpid)233 factor(newpid)234 factor(newpid)235 factor(newpid)236
##          4.3689529          6.2241301          3.2178233          2.8461214
```

```
## factor(newpid)237 factor(newpid)238 factor(newpid)239 factor(newpid)240
##          5.4434274          5.0629095          5.5450783          3.5181685
## factor(newpid)241 factor(newpid)242 factor(newpid)243 factor(newpid)244
##          6.1155539          5.1737754          5.9052043          5.9434971
## factor(newpid)245 factor(newpid)246 factor(newpid)247 factor(newpid)248
##          4.9302980          5.0574571          4.7853944          5.6432166
## factor(newpid)249 factor(newpid)250 factor(newpid)251 factor(newpid)252
##          5.5946403          5.8370910          3.7416574          4.5153795
## factor(newpid)253 factor(newpid)254               time
##          3.6055513          3.7595875         -0.4013592
```

```r
ggplot(data = hiv.data,aes(x=time,y=y,col=factor(newpid))) + geom_point() + geom_smooth(se=F,method = "l
```



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.

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------ tidyverse 1.2.1 --
## √ tibble  1.4.2     √ purrr   0.2.5
## √ tidyr   0.8.1     √ dplyr   0.7.4
## √ readr   1.1.1     √ forcats 0.3.0
```

```
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::combine()   masks gridExtra::combine()
## x tidyr::expand()    masks Matrix::expand()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x dplyr::recode()    masks car::recode()
## x dplyr::select()    masks MASS::select()
## x purrr::some()      masks car::some()
## x purrr::transpose() masks data.table::transpose()
```

```r
coef<-coef(model1)
coef_matrix<-matrix(0,nrow = length(coef)-1, ncol=5)
colnames(coef_matrix)<-c("newpid","intercept","slop","treatment","age")
newpid<-unique(hiv.data$newpid)
coef_matrix[,1]<-factor(newpid)
coef_matrix[,2]<-coef[-length(coef)]
coef_matrix[,3]<-rep(coef[length(coef)],length(coef)-1)
a<-hiv.data %>%
  group_by(factor(newpid)) %>%
  filter(row_number() == 1) %>%     #????????????????????????
  ungroup %>%
  dplyr::select(newpid,treatment,age.baseline)

coef<-merge(coef_matrix[,1:3],a,by="newpid")

model_intercept<-lm(intercept~treatment + age.baseline,data=coef)
model_slop<-lm(slop~treatment + age.baseline,data=coef)

summary(model_intercept)
```

```
##
## Call:
## lm(formula = intercept ~ treatment + age.baseline, data = coef)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9511 -0.6352  0.2292  0.9586  2.8988
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.18363    0.34684  14.945   <2e-16 ***
## treatment    -0.15222    0.20044  -0.759    0.449
## age.baseline -0.03323    0.04390  -0.757    0.450
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.408 on 195 degrees of freedom
## Multiple R-squared:  0.005778,   Adjusted R-squared:  -0.004419
## F-statistic: 0.5666 on 2 and 195 DF,  p-value: 0.5684
```

```r
summary(model_slop)
```

```
##
## Call:
## lm(formula = slop ~ treatment + age.baseline, data = coef)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.072e-16 -7.310e-17 -6.430e-17  1.600e-18  7.734e-15
##
## Coefficients:
##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  -4.014e-01  1.371e-16 -2.928e+15   <2e-16 ***
## treatment    -7.484e-17  7.922e-17 -9.450e-01    0.346
## age.baseline  4.006e-18  1.735e-17  2.310e-01    0.818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.566e-16 on 195 degrees of freedom
## Multiple R-squared:  0.4985, Adjusted R-squared:  0.4933
## F-statistic:  96.9 on 2 and 195 DF,  p-value: < 2.2e-16
```

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
model_2<-lmer(data = hiv.data, sqrt(CD4PCT)~(1|newpid) + time)
display(model_2)
```

```
## lmer(formula = sqrt(CD4PCT) ~ (1 | newpid) + time, data = hiv.data)
##             coef.est coef.se
## (Intercept)  4.81     0.10
## time        -0.39     0.06
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.40
##  Residual             0.76
## ---
## number of obs: 978, groups: newpid, 226
## AIC = 2852.6, DIC = 2831.1
## deviance = 2837.9
```

The average model for all children (complete pooling) is written as:

$y = 4.81 - 0.39 * time$

With 1% increase in time, the square root scale of CD4 percentage will decrease by 0.39%

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
model_3<-lmer(data = hiv.data, sqrt(CD4PCT)~(1|newpid) + time + treatment + age.baseline)
display(model_3)
```

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

```
## (Intercept)   4.76     0.33
## time         -0.38     0.06
## treatment     0.29     0.19
## age.baseline -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.37
##  Residual             0.76
## ---
## number of obs: 978, groups: newpid, 226
## AIC = 2852.5, DIC = 2815
## deviance = 2827.8
```

The average model for all children (complete pooling) is written as:

$y = 4.76 - 0.38 * time + 0.29 * treatment - 0.12 * age.baseline$

With 1% increase in time, the square root scale of CD4 percentage will decrease by 0.38%.

With 1 increase in treatment, the square root scale of CD4 percentage will increase by 0.29

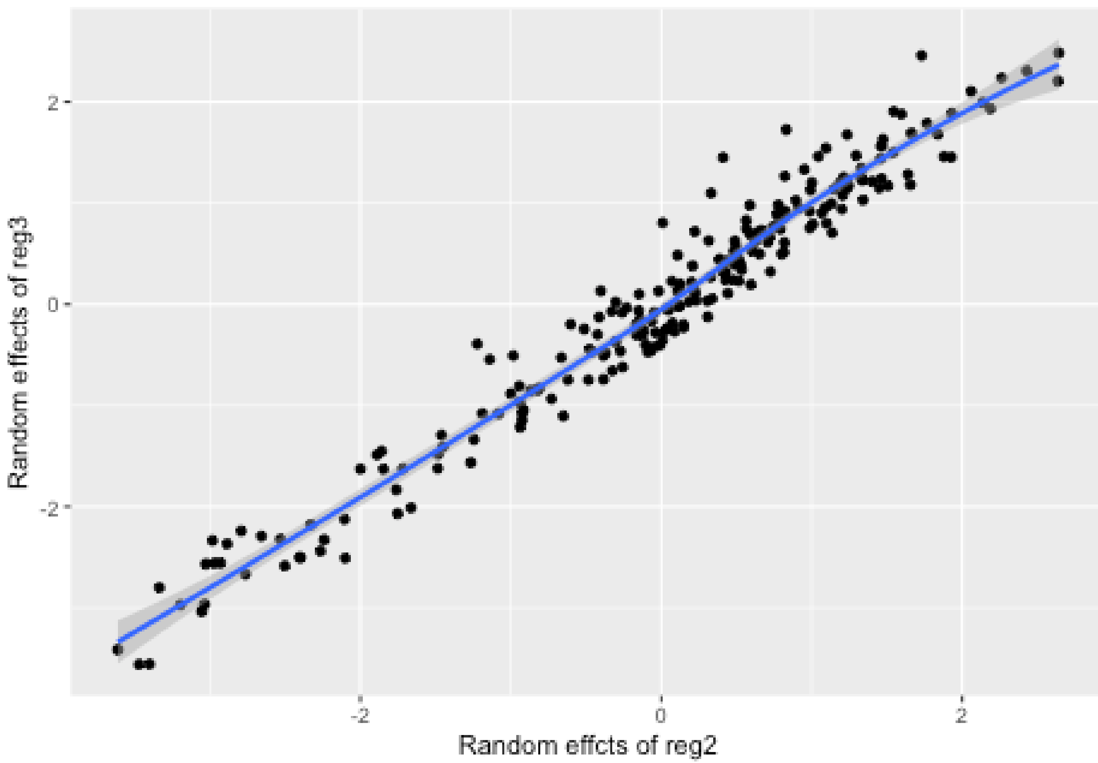
With 1% increase in age.baseline, the square root scale of CD4 percentage will decrease by 0.12%

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

```
data_plot <- as.data.frame(cbind(unlist(ranef(model_2)),unlist(ranef(model_3))))
colnames(data_plot) <- c("model_2","model_3")

ggplot(data=data_plot,aes(x=model_2,y=model_3))+geom_point()+geom_smooth()+
  xlab("Random effcts of reg2")+
  ylab("Random effects of reg3")
```

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



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

```
coef<-coef(model_3)
add_time<-0.3 #set the next time point is 0.3 bigger than last time point
newdata<-hiv.data %>%
  group_by(newpid) %>%
  arrange(desc(time),.by_group=TRUE) %>%
  filter(row_number() == 1) %>%
  ungroup %>%
  mutate(CD4PCT=sqrt(CD4PCT)) %>%
  select(newpid,time,treatment,age.baseline) %>%
  mutate(time=time+add_time)

CD4PCT_sqrt_new<-diag(as.matrix(cbind(rep(1,dim(newdata)[1]),newdata[,2:4])) %*% t(coef$newpid))

result<-cbind(newdata,CD4PCT_sqrt_new^2)
colnames(result)[5]<-"CD4PCT"
kable(result)
```

newpid	time	treatment	age.baseline	CD4PCT
1	2.2383333	1	3.9100000	13.7383221
2	0.5133333	2	3.5650000	1.3798355
3	2.0250000	1	6.1241667	26.0485143
4	1.4883333	1	2.3025000	25.0164524
5	0.3000000	1	0.6541667	16.9652919
6	1.8308333	2	2.9183333	21.6146226
7	1.7558333	2	6.4425000	24.0213894
8	1.5458333	1	5.0266667	20.2860776

newpid	time	treatment	age.baseline	CD4PCT
9	1.4883333	1	1.4975000	30.9845863
11	1.7183333	2	3.0583333	3.5888577
12	1.6033333	2	5.7383333	14.1636167
13	1.1241667	1	2.5266667	23.5515277
15	1.6716667	1	1.3275000	20.9911241
16	1.2608333	1	1.2783333	4.3968387
17	1.7950000	1	1.6233333	28.6257466
18	1.2258333	2	4.5775000	30.0763146
19	1.9483333	2	0.4791667	11.8802376
20	0.8016667	1	0.6650000	22.0854477
21	0.3000000	2	0.4466667	24.5694797
22	0.3000000	1	6.8775000	31.2794769
23	1.7533333	1	6.2125000	1.1808924
24	1.2391667	1	9.5825000	18.1285481
25	1.7150000	2	1.3033333	17.0152421
26	1.4358333	1	1.8675000	16.7463864
27	1.4683333	2	9.3775000	14.5506842
28	0.7791667	1	2.4175000	27.4691146
29	1.2966667	2	6.2700000	16.2950917
30	0.3000000	2	8.4325000	2.8364651
31	1.0008333	1	1.4675000	17.0746230
32	1.8908333	1	2.9625000	15.4094020
33	1.7700000	1	1.3883333	18.9197253
34	0.5575000	1	3.1841667	36.4043915
35	1.3350000	2	3.4058333	19.5795050
38	1.0258333	1	2.1983333	31.7806038
39	1.2658333	2	3.8825000	19.0474729
40	0.3000000	2	2.0641667	14.8128210
41	0.3000000	2	0.7750000	24.4779183
42	1.7975000	2	4.6158333	6.9565131
43	1.7975000	1	2.8500000	17.9433625
44	0.7733333	1	1.8675000	5.8368023
45	1.6716667	2	2.1325000	20.4119587
46	1.7016667	1	1.3116667	8.5081824
47	1.6991667	2	3.6933333	17.7244477
48	1.4141667	2	6.9925000	15.3386037
49	1.7258333	1	3.7566667	22.1122185
50	1.7150000	1	2.0891667	13.6155141
51	1.2766667	1	1.9716667	12.3022812
52	1.6858333	1	6.6775000	1.5769835
53	1.6658333	2	1.4483333	17.5715804
54	1.7100000	2	1.7000000	14.7605271
55	1.6583333	1	3.4575000	3.3787708
56	0.5300000	2	1.4400000	8.4800439
57	1.6908333	1	3.9425000	2.6314385
58	1.6908333	1	5.3608333	2.2099639
59	1.7316667	2	2.2425000	19.8995790
61	1.4641667	1	11.3975000	21.1739156
62	1.7750000	2	1.9908333	24.5410916
63	1.6991667	1	2.9483333	1.9375282
64	1.6875000	1	3.0858333	22.4998621
65	1.2308333	1	4.0741667	1.2874048

newpid	time	treatment	age.baseline	CD4PCT
66	0.9983333	2	1.5525000	35.5287814
67	0.5491667	1	5.6650000	6.5060290
68	0.3000000	1	7.1758333	28.7604677
69	2.0166667	2	3.4275000	20.8941313
70	1.1783333	1	6.6641667	20.6063166
71	0.3000000	2	0.5666667	9.9830984
72	1.3208333	2	4.3700000	11.3160094
74	1.6741667	1	8.9308333	19.9021062
75	1.6791667	2	2.5958333	26.6191109
76	1.6750000	1	5.5383333	18.0532796
78	1.6766667	2	0.8625000	28.2767359
79	0.7383333	2	1.2758333	21.6082271
81	0.7600000	1	7.1866667	1.2490833
82	1.3925000	1	3.6333333	7.8701981
83	0.3000000	2	8.8483333	2.6679064
84	1.4883333	1	1.2100000	3.3989256
85	1.4883333	2	5.1966667	1.4484385
86	1.6416667	1	1.8041667	32.6936529
87	1.7183333	1	1.6016667	29.0023209
88	0.5491667	2	1.8208333	21.7508989
89	1.2966667	1	2.2966667	20.3339207
90	0.3000000	2	1.3500000	31.1085453
91	1.2583333	2	0.8600000	5.1155968
93	0.9900000	1	2.2175000	1.9093668
94	1.7700000	1	2.6833333	18.0181098
95	0.5300000	1	1.3441667	8.2066158
96	0.3000000	1	3.8633333	22.2170630
97	1.2858333	2	0.7141667	48.7906914
98	0.3000000	2	0.2025000	23.1175685
99	1.4500000	2	0.3475000	35.2737374
100	1.2441667	2	1.4650000	35.8586288
101	0.3000000	1	0.4766667	29.0122599
104	1.5650000	2	5.5933333	9.0861235
105	1.8166667	1	0.8950000	15.8670460
106	1.3900000	2	2.0916667	11.2972378
107	1.5258333	1	2.4808333	26.3281284
109	0.5491667	1	3.4333333	15.3372701
110	1.7925000	1	2.8775000	21.2452294
112	1.8191667	2	2.8200000	11.4930439
113	1.9150000	2	2.7108333	30.7506457
116	1.0358333	2	4.9558333	15.4335268
117	2.2083333	1	1.1800000	12.9226557
118	1.3350000	2	2.5050000	22.9077332
119	0.6316667	1	1.1800000	4.3727405
120	1.7725000	1	1.8016667	36.6859756
121	1.7291667	2	2.2391667	28.8108332
124	0.3000000	2	1.1525000	12.5080163
126	1.7183333	1	5.0866667	14.6031107
127	1.4691667	2	3.7016667	21.8276821
128	1.4691667	2	4.3175000	17.6562437
129	1.4883333	1	3.8441667	0.3856578
131	1.6991667	1	4.5666667	13.1370576

newpid	time	treatment	age.baseline	CD4PCT
132	1.5650000	1	1.1908333	17.2286249
133	1.4883333	1	6.3491667	10.4339418
134	1.6958333	2	3.2691667	35.9901006
135	1.7625000	1	3.2358333	23.9267596
136	1.7566667	1	2.3766667	34.7580865
137	1.7566667	1	10.0808333	24.2188823
138	0.3000000	1	2.8833333	45.0265101
139	1.8416667	2	2.1875000	17.2669141
140	1.7816667	1	2.7925000	22.6277591
141	1.7541667	2	4.4625000	40.8881415
142	1.6025000	2	8.1591667	5.2286912
143	1.7458333	1	4.0491667	5.2147614
144	1.7125000	1	2.2891667	28.4860411
145	1.5291667	1	1.3441667	24.3246593
146	1.4883333	1	2.1275000	23.6155517
147	1.5075000	2	6.2366667	30.3735101
148	0.7600000	1	1.2733333	25.1162738
149	1.2550000	1	2.5575000	26.3664898
150	1.0450000	2	1.3441667	16.4137945
151	1.0091667	2	2.7433333	27.5935825
152	0.3000000	2	7.3316667	19.9782891
153	0.7575000	2	2.3216667	45.1883246
154	0.8041667	2	5.5766667	28.1899445
155	0.7766667	1	2.3108333	34.0226797
156	0.5325000	1	1.1225000	35.3878606
157	1.1625000	1	2.0258333	33.8153408
158	1.7950000	2	3.4908333	28.4134913
159	0.8333333	2	2.8091667	24.4243341
160	0.5491667	1	3.8250000	22.8216810
161	1.1625000	2	0.6875000	22.1308191
162	0.8558333	2	1.8591667	19.4141044
163	1.0666667	1	1.0841667	46.6570112
164	2.1966667	2	3.4116667	37.2854988
165	1.6033333	2	3.5041667	14.5656694
166	1.3866667	1	1.6566667	25.0033800
167	1.6608333	2	3.3783333	18.4673983
168	1.6800000	1	5.5275000	25.9636677
169	0.9708333	1	2.9458333	7.3666043
170	1.4491667	2	2.9025000	15.9444180
171	0.3000000	1	3.4466667	37.2555964
172	1.6600000	2	4.6000000	21.2092395
173	0.7983333	2	9.8616667	2.6904215
174	1.4883333	1	3.3400000	10.8016529
175	1.5458333	1	3.9891667	27.0349327
176	0.9700000	2	5.3391667	27.5458817
177	1.4308333	1	2.1108333	16.9469406
178	1.5266667	1	8.7500000	34.8217455
179	0.7566667	1	4.6300000	25.5235947
181	1.6991667	1	0.8075000	46.2477313
182	1.8058333	2	0.7908333	36.3584415
184	0.3000000	2	0.3891667	22.3019851
185	1.7241667	1	3.4000000	21.4018257

newpid	time	treatment	age.baseline	CD4PCT
186	1.6958333	1	1.1391667	3.1388713
187	1.7566667	2	2.0041667	27.5079390
188	1.7366667	2	4.2358333	24.3488782
189	0.7625000	1	0.4241667	1.3433327
191	0.8175000	2	0.9750000	19.9375920
192	1.6966667	2	3.4166667	15.9776135
194	1.4583333	1	7.8491667	1.5276496
196	1.4691667	2	2.2200000	14.2954690
198	1.7566667	2	8.6266667	28.4965583
199	1.4500000	1	7.5425000	9.4091806
200	2.1116667	2	1.9908333	29.8129840
201	1.7016667	2	2.3000000	18.0518095
203	1.7316667	2	2.9516667	30.8726257
204	1.2200000	1	5.7083333	23.9713489
205	1.8333333	1	2.1275000	9.1554656
206	1.6825000	1	0.9500000	28.0778712
207	1.6883333	2	5.0483333	28.8186585
208	1.6858333	1	5.7658333	12.4988617
209	1.6800000	1	5.3825000	32.3355520
210	1.6800000	2	5.1008333	20.6751669
212	1.4608333	2	5.1416667	21.4507007
213	0.7675000	1	0.8600000	19.4335473
214	1.7100000	1	2.4258333	33.7543063
215	1.4575000	2	3.6225000	20.0651112
216	1.4583333	1	3.1566667	10.5284861
217	0.5325000	1	0.7008333	9.9162188
218	1.6800000	2	2.4966667	17.1046920
219	0.3000000	2	1.5058333	27.9899452
220	1.6858333	2	6.0641667	31.6373183
222	1.6825000	2	1.6700000	21.3495663
223	1.4450000	2	4.0025000	22.7620701
224	0.5300000	2	4.7116667	13.9775653
225	1.6800000	1	4.6733333	33.0065722
226	1.4500000	2	0.4108333	38.4910601
227	1.4500000	2	1.9000000	31.4262584
228	1.4391667	1	0.5366667	17.1130544
229	1.4608333	1	1.2075000	21.8788074
230	1.4500000	2	3.4500000	28.5820804
231	1.2200000	1	0.6650000	29.3322959
232	1.2833333	2	4.0108333	30.9230609
233	0.9708333	1	4.7308333	16.0098430
234	0.5300000	2	3.8250000	34.1151443
235	1.4883333	2	4.9883333	7.7646299
236	1.4633333	1	5.7716667	5.5974902
237	1.7375000	2	1.0733333	22.6272918
238	1.1675000	1	0.3833333	21.2107408
239	0.9841667	1	2.7683333	25.8747558
240	1.2058333	1	9.5716667	9.4446624
241	0.3000000	2	2.3491667	33.0989691
242	0.9958333	1	1.5933333	22.6494828
243	1.5041667	1	3.4691667	27.0762016
244	0.5325000	1	1.5116667	31.2963371

newpid	time	treatment	age.baseline	CD4PCT
245	1.0308333	1	2.3108333	20.4248414
246	0.5491667	1	1.5116667	23.2467141
247	0.3000000	2	6.2391667	21.4382995
248	0.5516667	2	0.7258333	28.9385984
249	0.3000000	1	1.2950000	28.2583567
250	0.5458333	1	3.4883333	29.8715833
251	0.3000000	1	1.7225000	15.1474669
252	0.6283333	2	1.0325000	19.0897175
253	0.3000000	2	2.7950000	14.6548493
254	0.8825000	1	0.8216667	12.8153440

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.

```r
coef<-fixef(model_3)
time<-seq(0,2,0.1)#set time
X<-cbind(rep(1,length(time)),time,rep(1,length(time)),rep(4,length(time)))#set treatment=1
pred<-X %*% as.matrix(coef)
result<-cbind(rep(1,length(time)),time,rep(4,length(time)),pred^2)
colnames(result)<-c("treatment","time","age.baseline","CD4PCT")
kable(result)
```

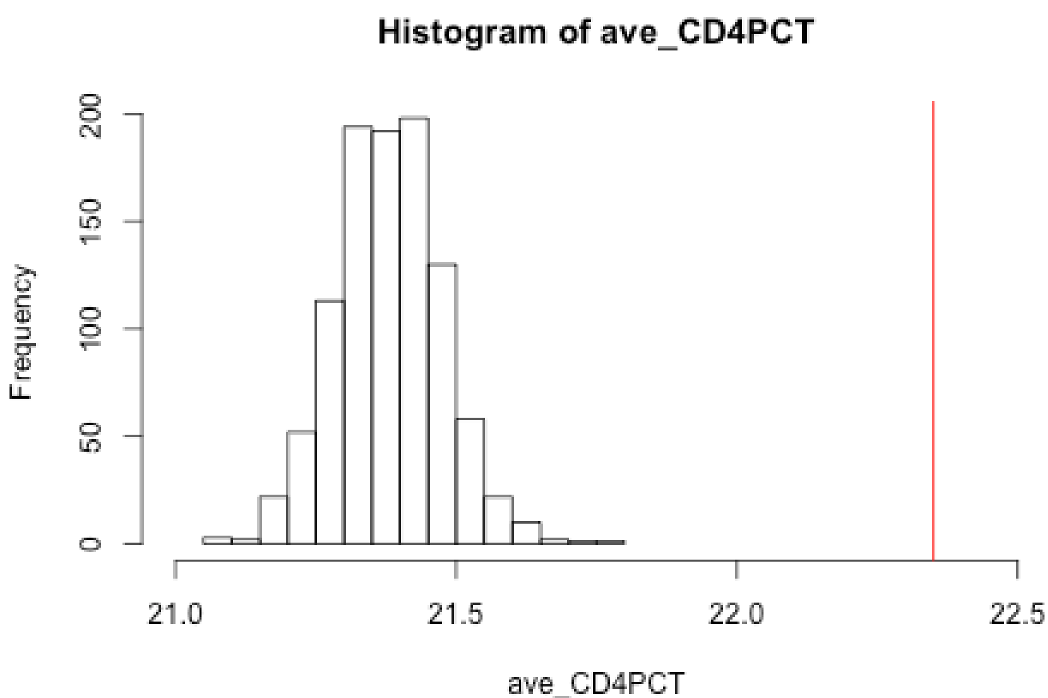
treatment	time	age.baseline	CD4PCT
1	0.0	4	21.02477
1	0.1	4	20.67577
1	0.2	4	20.32969
1	0.3	4	19.98654
1	0.4	4	19.64630
1	0.5	4	19.30898
1	0.6	4	18.97459
1	0.7	4	18.64311
1	0.8	4	18.31456
1	0.9	4	17.98893
1	1.0	4	17.66622
1	1.1	4	17.34643
1	1.2	4	17.02956
1	1.3	4	16.71561
1	1.4	4	16.40458
1	1.5	4	16.09648
1	1.6	4	15.79129
1	1.7	4	15.48902
1	1.8	4	15.18968
1	1.9	4	14.89326
1	2.0	4	14.59975

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
newdata<-hiv.data %>%
  group_by(newpid) %>%
  arrange(desc(time),.by_group=TRUE) %>%
  filter(row_number() == 1) %>%
  select(newpid,treatment,time,age.baseline,CD4PCT)
newdata_original_mean<-mean(newdata$CD4PCT)

for(i in 1:1000) {
  newdata$treatment<-rbernoulli(dim(newdata)[1], p = sum(hiv.data$treatment==1)/dim(hiv.data)[1])
  newdata$treatment[newdata$treatment==0]<-2
  model_sim<-lmer(data = hiv.data, sqrt(CD4PCT)~(1|newpid) + time + treatment + age.baseline)
  re<-predict(model_sim,newdata=newdata)^2
  if(i==1)
    result<-re
  else
    result<-cbind(result,re)
}

ave_CD4PCT<-apply(result,2,mean)
hist(ave_CD4PCT, xlim = c(21,22.5))
abline(v=newdata_original_mean,col="red")
```



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

```r
model_4<-lmer(y~time+(1+time|newpid),data=hiv.data)
display(model_4)
```

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

```
## (Intercept)  4.80     0.10
## time        -0.37     0.07
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  newpid   (Intercept) 1.39
##           time        0.59     -0.10
##  Residual             0.71
## ---
## number of obs: 978, groups: newpid, 226
## AIC = 2826.6, DIC = 2801.9
## deviance = 2808.2
```

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_5<-lmer(y ~ factor(time) + treatment + (1 | newpid),data = hiv.data)
display(model_5)
```

```
## lmer(formula = y ~ factor(time) + treatment + (1 | newpid), data = hiv.data)
##                                 coef.est coef.se
## (Intercept)                      4.27     0.31
## factor(time)0.205               -1.24     0.67
## factor(time)0.209999999999999  0.17     0.89
## factor(time)0.213333333333333  0.19     0.95
## factor(time)0.213333333333334 -1.25     0.95
## factor(time)0.215833333333332  1.50     0.90
## factor(time)0.215833333333334 -0.25     0.88
## factor(time)0.216666666666667 -0.33     0.81
## factor(time)0.219166666666667 -0.46     0.86
## factor(time)0.221666666666667  0.22     0.95
## factor(time)0.224166666666667 -1.45     0.85
## factor(time)0.226666666666667  1.43     0.59
## factor(time)0.227499999999999 -1.57     0.89
## factor(time)0.2275              -0.01     0.46
## factor(time)0.229999999999999 -0.39     0.59
## factor(time)0.23                -0.13     0.13
## factor(time)0.2325              -0.54     0.45
## factor(time)0.233333333333333  0.06     0.84
## factor(time)0.235000000000001 -1.94     0.80
## factor(time)0.235833333333333  0.03     0.30
## factor(time)0.235833333333334  0.17     0.62
## factor(time)0.2375               1.77     0.95
## factor(time)0.238333333333333 -0.24     0.49
## factor(time)0.238333333333334  0.84     0.82
## factor(time)0.240833333333333 -0.20     0.79
## factor(time)0.240833333333334  0.33     0.59
## factor(time)0.243333333333333 -0.50     0.89
## factor(time)0.244166666666667  0.07     0.48
## factor(time)0.245833333333333  0.10     0.43
## factor(time)0.245833333333334 -0.24     0.61
## factor(time)0.246666666666666 -0.26     0.67
## factor(time)0.246666666666667  0.37     0.86
## factor(time)0.249166666666666  0.17     0.39
## factor(time)0.249166666666667 -0.33     0.20
```

```
## factor(time)0.251666666666667  0.25     0.43
## factor(time)0.251666666666668  0.29     0.81
## factor(time)0.254166666666667 -0.60     0.84
## factor(time)0.256666666666667  0.30     0.65
## factor(time)0.257499999999999  0.07     0.85
## factor(time)0.2575             0.46     0.63
## factor(time)0.2625            -0.05     0.85
## factor(time)0.265             -0.18     0.85
## factor(time)0.265833333333333 -0.35     0.84
## factor(time)0.268333333333333 -0.41     0.60
## factor(time)0.268333333333334  0.04     0.50
## factor(time)0.2875             0.48     0.53
## factor(time)0.293333333333333 -0.33     0.95
## factor(time)0.304166666666667  0.25     0.89
## factor(time)0.306666666666666 -0.28     0.95
## factor(time)0.306666666666667  0.46     0.83
## factor(time)0.325833333333334 -0.13     0.90
## factor(time)0.328333333333333 -0.92     0.95
## factor(time)0.331666666666667  0.07     0.95
## factor(time)0.358333333333333  0.67     0.87
## factor(time)0.364166666666667 -0.05     0.61
## factor(time)0.429166666666666 -0.26     0.95
## factor(time)0.429166666666667 -0.41     0.90
## factor(time)0.438333333333333 -0.88     0.90
## factor(time)0.440833333333333 -0.10     0.78
## factor(time)0.443333333333332  0.12     0.85
## factor(time)0.449166666666667 -0.11     0.80
## factor(time)0.454166666666666  0.30     0.90
## factor(time)0.455             -1.46     0.89
## factor(time)0.456666666666667  0.27     0.95
## factor(time)0.4575            -0.17     0.49
## factor(time)0.459166666666667  0.23     0.62
## factor(time)0.459999999999999  0.25     0.57
## factor(time)0.46              -0.27     0.17
## factor(time)0.460000000000001 -0.23     0.34
## factor(time)0.462499999999999 -1.00     0.59
## factor(time)0.4625             0.43     0.41
## factor(time)0.463333333333333 -0.71     0.81
## factor(time)0.465             -0.84     0.49
## factor(time)0.465833333333333  0.33     0.59
## factor(time)0.465833333333334 -0.60     0.83
## factor(time)0.4675             2.33     0.95
## factor(time)0.468333333333335 -0.78     0.86
## factor(time)0.470833333333333 -0.33     0.47
## factor(time)0.470833333333334 -0.48     0.61
## factor(time)0.473333333333333 -0.17     0.62
## factor(time)0.473333333333334  0.47     0.79
## factor(time)0.474166666666666 -0.66     0.82
## factor(time)0.474166666666667 -0.86     0.81
## factor(time)0.475833333333333 -0.44     0.81
## factor(time)0.475833333333334 -1.38     0.59
## factor(time)0.476666666666667  0.62     0.41
## factor(time)0.479166666666666 -0.22     0.58
## factor(time)0.479166666666667 -0.02     0.25
```

```
## factor(time)0.481666666666666  0.14      0.59
## factor(time)0.481666666666667  0.24      0.43
## factor(time)0.484166666666667 -2.71      0.81
## factor(time)0.485               0.89      0.62
## factor(time)0.487499999999999 -0.17      0.77
## factor(time)0.4875              1.80      0.82
## factor(time)0.487500000000001  1.75      0.81
## factor(time)0.489999999999999  0.08      0.86
## factor(time)0.495              -0.11      0.60
## factor(time)0.495833333333333 -0.07      0.90
## factor(time)0.498333333333333 -0.78      0.44
## factor(time)0.498333333333334 -0.38      0.43
## factor(time)0.500833333333333 -0.25      0.80
## factor(time)0.500833333333334  0.12      0.82
## factor(time)0.501666666666667 -1.15      0.62
## factor(time)0.503333333333334  0.61      0.84
## factor(time)0.504166666666666  0.23      0.52
## factor(time)0.505833333333333 -1.56      0.82
## factor(time)0.511666666666667 -1.14      0.88
## factor(time)0.514166666666666  0.12      0.63
## factor(time)0.515               0.19      0.95
## factor(time)0.5175             -0.50      0.42
## factor(time)0.517500000000001 -1.02      0.83
## factor(time)0.533333333333333 -0.31      0.65
## factor(time)0.533333333333334 -1.09      0.87
## factor(time)0.534166666666667  0.15      0.63
## factor(time)0.536666666666667 -0.41      0.51
## factor(time)0.555833333333333 -0.17      0.95
## factor(time)0.558333333333333  1.76      0.95
## factor(time)0.564166666666667 -0.97      0.90
## factor(time)0.575              -0.23      0.64
## factor(time)0.5825              0.67      0.95
## factor(time)0.594166666666667 -0.40      0.90
## factor(time)0.610833333333333 -1.13      0.95
## factor(time)0.6375              1.81      0.90
## factor(time)0.648333333333333 -2.09      0.89
## factor(time)0.651666666666666 -1.12      0.89
## factor(time)0.67               -0.38      0.85
## factor(time)0.670833333333333 -0.98      0.49
## factor(time)0.673333333333333  0.08      0.95
## factor(time)0.675833333333333 -0.12      0.67
## factor(time)0.684166666666667  0.50      0.90
## factor(time)0.685               0.59      0.84
## factor(time)0.6875             -1.55      0.61
## factor(time)0.689166666666666 -0.13      0.89
## factor(time)0.689166666666667  0.14      0.81
## factor(time)0.69               -0.16      0.17
## factor(time)0.6925              0.56      0.56
## factor(time)0.692500000000001  0.90      0.84
## factor(time)0.693333333333334 -0.46      0.86
## factor(time)0.695               0.37      0.88
## factor(time)0.695833333333333 -0.30      0.86
## factor(time)0.695833333333334 -1.67      0.84
## factor(time)0.695833333333335  0.81      0.60
```

```
## factor(time)0.698333333333332  0.07      0.84
## factor(time)0.698333333333333 -0.63      0.59
## factor(time)0.700833333333333  1.59      0.49
## factor(time)0.703333333333333 -0.90      0.58
## factor(time)0.703333333333334 -1.08      0.82
## factor(time)0.704166666666667 -4.82      0.81
## factor(time)0.705833333333333 -0.46      0.50
## factor(time)0.706666666666667  2.37      0.80
## factor(time)0.709166666666666 -0.13      0.80
## factor(time)0.709166666666667  0.26      0.28
## factor(time)0.711666666666666  0.12      0.84
## factor(time)0.711666666666667 -0.45      0.82
## factor(time)0.711666666666668 -0.88      0.57
## factor(time)0.714166666666667 -1.43      0.58
## factor(time)0.714999999999999 -0.71      0.80
## factor(time)0.715000000000001 -0.64      0.86
## factor(time)0.7175            -1.00      0.81
## factor(time)0.72              -4.30      0.78
## factor(time)0.725             -0.61      0.60
## factor(time)0.725833333333332  0.33      0.85
## factor(time)0.725833333333333  0.73      0.83
## factor(time)0.725833333333334  0.36      0.95
## factor(time)0.728333333333333 -0.27      0.29
## factor(time)0.730833333333333  0.06      0.51
## factor(time)0.733333333333333 -0.75      0.85
## factor(time)0.734166666666666 -2.87      0.82
## factor(time)0.735833333333333 -1.27      0.95
## factor(time)0.736666666666666  0.02      0.85
## factor(time)0.736666666666667  1.52      0.85
## factor(time)0.7425            -0.52      0.95
## factor(time)0.744166666666667  0.17      0.58
## factor(time)0.745             -0.40      0.83
## factor(time)0.7475            -0.33      0.37
## factor(time)0.752500000000001 -1.64      0.84
## factor(time)0.758333333333334  0.32      0.87
## factor(time)0.761666666666667 -0.64      0.81
## factor(time)0.763333333333335 -0.04      0.87
## factor(time)0.764166666666666 -0.05      0.90
## factor(time)0.765833333333333 -1.25      0.88
## factor(time)0.766666666666667 -0.39      0.39
## factor(time)0.775             -1.69      0.82
## factor(time)0.78               1.02      1.59
## factor(time)0.783333333333333  1.00      0.89
## factor(time)0.785             -0.19      0.83
## factor(time)0.788333333333333 -0.71      0.95
## factor(time)0.794166666666667 -0.70      0.82
## factor(time)0.8025            -0.57      0.82
## factor(time)0.805             -1.47      0.83
## factor(time)0.805000000000001  0.09      0.79
## factor(time)0.807500000000001 -0.52      0.90
## factor(time)0.824166666666667  0.14      0.64
## factor(time)0.8625            -0.46      0.61
## factor(time)0.8675             0.57      0.90
## factor(time)0.878333333333334 -0.21      0.89
```

```
## factor(time)0.895833333333333 -0.73     0.89
## factor(time)0.900833333333333 -0.41     0.85
## factor(time)0.900833333333334  0.27     0.59
## factor(time)0.903333333333333 -1.06     0.77
## factor(time)0.903333333333334  2.28     0.95
## factor(time)0.905833333333334  1.29     0.90
## factor(time)0.908333333333334 -0.41     0.90
## factor(time)0.909166666666667 -0.30     0.81
## factor(time)0.911666666666667 -0.26     0.79
## factor(time)0.9175            -0.51     0.84
## factor(time)0.919166666666667  0.04     0.48
## factor(time)0.919999999999998 -0.85     0.79
## factor(time)0.92              -0.98     0.28
## factor(time)0.920000000000001 -0.93     0.56
## factor(time)0.9225            -0.52     0.59
## factor(time)0.925833333333333 -1.74     0.59
## factor(time)0.925833333333334  0.33     0.84
## factor(time)0.928333333333333  0.03     0.79
## factor(time)0.928333333333334 -0.93     0.83
## factor(time)0.930833333333333 -0.70     0.56
## factor(time)0.930833333333334 -1.99     0.81
## factor(time)0.933333333333332  0.43     0.83
## factor(time)0.933333333333333  0.01     0.59
## factor(time)0.934166666666664 -0.80     0.84
## factor(time)0.934166666666667 -1.09     0.81
## factor(time)0.935833333333333 -0.51     0.59
## factor(time)0.935833333333334 -0.69     0.85
## factor(time)0.936666666666667  0.14     0.79
## factor(time)0.938333333333334  0.02     0.81
## factor(time)0.939166666666666 -0.17     0.34
## factor(time)0.939166666666667  0.06     0.33
## factor(time)0.941666666666666 -0.29     0.41
## factor(time)0.941666666666667 -0.48     0.82
## factor(time)0.944166666666667  0.15     0.88
## factor(time)0.9475             0.65     0.59
## factor(time)0.952500000000001 -0.43     0.79
## factor(time)0.955              1.58     0.84
## factor(time)0.955000000000001 -0.41     0.82
## factor(time)0.955833333333333 -1.09     0.85
## factor(time)0.9575             0.37     0.79
## factor(time)0.958333333333333  0.11     0.36
## factor(time)0.958333333333334 -0.14     0.46
## factor(time)0.960833333333333 -1.52     0.83
## factor(time)0.964166666666666 -0.04     0.85
## factor(time)0.964166666666667  0.50     0.83
## factor(time)0.965833333333333  0.02     0.85
## factor(time)0.976666666666667 -0.48     0.84
## factor(time)0.977499999999999 -3.28     0.82
## factor(time)0.9775            -0.37     0.36
## factor(time)0.977500000000001  0.56     0.86
## factor(time)0.982499999999999 -0.29     0.82
## factor(time)0.9825            -0.10     0.85
## factor(time)0.983333333333333 -0.06     0.88
## factor(time)0.985833333333333 -0.45     0.95
```

```
## factor(time)0.996666666666666 -0.53      0.57
## factor(time)0.996666666666667 -0.94      0.42
## factor(time)0.999166666666667 -1.00      0.78
## factor(time)1                 -0.95      0.84
## factor(time)1.00166666666667  -0.35      0.84
## factor(time)1.0025            -1.29      0.85
## factor(time)1.01083333333333  -0.37      0.81
## factor(time)1.01583333333333   0.16      0.90
## factor(time)1.02083333333333  -1.32      0.86
## factor(time)1.0325             0.73      0.89
## factor(time)1.035             -0.68      0.44
## factor(time)1.04833333333333  -1.33      0.88
## factor(time)1.05333333333333  -0.10      0.83
## factor(time)1.05416666666667  -0.47      0.59
## factor(time)1.07583333333333  -1.02      0.85
## factor(time)1.08666666666667   0.27      1.59
## factor(time)1.09              -1.57      0.95
## factor(time)1.0925            -0.30      0.63
## factor(time)1.11416666666667   0.07      0.86
## factor(time)1.13083333333333   0.36      0.85
## factor(time)1.13583333333333  -2.77      0.95
## factor(time)1.13916666666667   0.12      0.79
## factor(time)1.14166666666667  -0.48      0.58
## factor(time)1.145             -0.43      0.89
## factor(time)1.1475            -1.37      0.84
## factor(time)1.14916666666667  -0.67      0.58
## factor(time)1.15              -0.41      0.23
## factor(time)1.1525            -0.99      0.55
## factor(time)1.15583333333333  -1.18      0.49
## factor(time)1.1575            -0.05      0.56
## factor(time)1.15833333333333  -1.63      0.60
## factor(time)1.16083333333333  -1.04      0.41
## factor(time)1.16333333333333   0.85      0.60
## factor(time)1.16416666666667   0.34      0.84
## factor(time)1.16583333333333  -1.56      0.83
## factor(time)1.16833333333333  -0.33      0.81
## factor(time)1.16916666666667  -0.47      0.34
## factor(time)1.17166666666667  -1.38      0.58
## factor(time)1.17666666666667  -1.27      0.85
## factor(time)1.1775            -0.08      0.81
## factor(time)1.18              -0.47      0.80
## factor(time)1.18833333333333  -0.10      0.22
## factor(time)1.19666666666667   0.31      0.63
## factor(time)1.20166666666667   0.09      0.48
## factor(time)1.20416666666667  -0.58      0.60
## factor(time)1.20666666666667   0.28      0.79
## factor(time)1.2075            -1.10      0.48
## factor(time)1.2125             0.09      0.85
## factor(time)1.22416666666667  -0.33      0.86
## factor(time)1.22583333333333   0.03      0.87
## factor(time)1.22666666666667   0.04      0.60
## factor(time)1.22916666666667  -0.11      0.59
## factor(time)1.23166666666667  -1.40      0.81
## factor(time)1.2325            -0.11      0.80
```

```
## factor(time)1.24583333333333  -0.64      0.59
## factor(time)1.24833333333333  -1.64      0.85
## factor(time)1.265             -0.22      0.47
## factor(time)1.2675            -0.22      0.84
## factor(time)1.28416666666667  -1.90      0.90
## factor(time)1.3025            -0.73      0.88
## factor(time)1.30333333333333  -0.32      0.57
## factor(time)1.31166666666667  -1.99      0.82
## factor(time)1.34166666666667  -1.10      0.79
## factor(time)1.35              -0.39      0.89
## factor(time)1.35833333333333  -1.69      0.84
## factor(time)1.36              -0.13      0.89
## factor(time)1.36083333333333  -0.02      0.90
## factor(time)1.36583333333333  -0.75      0.79
## factor(time)1.37166666666667  -0.34      0.57
## factor(time)1.37416666666667  -0.60      0.79
## factor(time)1.375             -0.81      0.79
## factor(time)1.37666666666667  -3.03      0.77
## factor(time)1.37916666666667   0.05      0.78
## factor(time)1.38              -0.55      0.36
## factor(time)1.3825            -1.54      0.55
## factor(time)1.38583333333333  -0.78      0.59
## factor(time)1.38583333333334   0.17      0.83
## factor(time)1.3875             0.15      0.79
## factor(time)1.38833333333333  -1.48      0.80
## factor(time)1.39083333333333  -0.56      0.57
## factor(time)1.39583333333333  -0.58      0.60
## factor(time)1.39666666666667   0.29      0.77
## factor(time)1.39833333333333  -0.27      0.83
## factor(time)1.39916666666667  -0.30      0.37
## factor(time)1.40166666666667  -2.16      0.61
## factor(time)1.41              -1.28      0.55
## factor(time)1.4125            -0.65      0.85
## factor(time)1.415             -0.18      0.58
## factor(time)1.41833333333333  -0.13      0.46
## factor(time)1.42083333333333  -0.71      0.95
## factor(time)1.42416666666667   0.40      0.79
## factor(time)1.42583333333333  -0.13      0.82
## factor(time)1.42916666666667   0.68      0.88
## factor(time)1.43166666666667  -0.13      0.58
## factor(time)1.43666666666667   0.43      0.81
## factor(time)1.4375            -0.85      0.79
## factor(time)1.44583333333333  -3.44      0.82
## factor(time)1.45333333333333  -0.83      0.90
## factor(time)1.45416666666667  -0.46      0.86
## factor(time)1.45583333333333  -0.58      0.79
## factor(time)1.45666666666667  -0.19      0.41
## factor(time)1.4625             0.33      0.80
## factor(time)1.47               0.20      0.59
## factor(time)1.4725             0.15      0.82
## factor(time)1.475              0.26      0.85
## factor(time)1.48166666666667  -1.51      0.85
## factor(time)1.4925            -2.56      0.81
## factor(time)1.495              0.17      0.48
```

```
## factor(time)1.4975              0.42     0.63
## factor(time)1.50583333333333  -0.75     0.90
## factor(time)1.51416666666667   0.26     0.79
## factor(time)1.51666666666667  -1.90     0.84
## factor(time)1.51916666666667  -2.73     0.90
## factor(time)1.53                0.48     0.85
## factor(time)1.53083333333333   0.05     0.89
## factor(time)1.53333333333333   0.25     0.82
## factor(time)1.54166666666667  -1.03     0.82
## factor(time)1.59083333333333  -1.12     0.89
## factor(time)1.615              -0.69     0.83
## factor(time)1.64833333333333  -1.56     0.85
## factor(time)1.71666666666667   0.00     0.85
## factor(time)1.725              -0.36     0.79
## factor(time)1.81166666666667   0.32     0.85
## factor(time)1.89666666666667  -0.35     0.83
## factor(time)1.90833333333333  -0.64     0.87
## factor(time)1.93833333333333  -0.86     0.95
## treatment                       0.36     0.20
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.41
##  Residual             0.71
## ---
## number of obs: 978, groups: newpid, 226
## AIC = 2719.6, DIC = 2455.3
## deviance = 2208.4
```

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

```r
data_plot2_inter <- as.data.frame(cbind(unlist(ranef(model_4))[1:226],unlist(ranef(model_5))[1:226]))
colnames(data_plot2_inter) <- c("model_4","model_5")

ggplot(data=data_plot2_inter,aes(x=model_4,y=model_5))+geom_point()+geom_smooth()+
  xlab("Random effects of model_4 intercepts")+
  ylab("Random effects of model_5 intercepts")
```

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

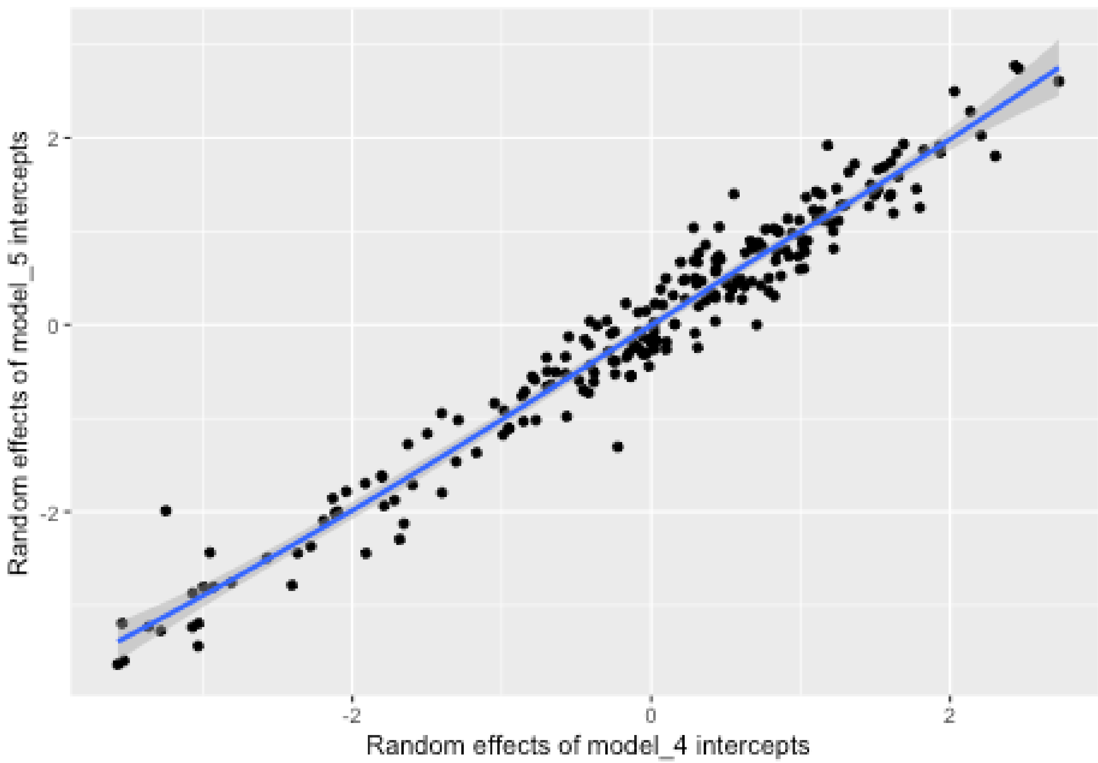


Figure skate in the 1932 Winter Olympics

The folder olympics has seven judges' ratings of seven figure skaters (on two criteria: "technical merit" and "artistic impression") from the 1932 Winter Olympics. Take a look at http://www.stat.columbia.edu/~gelman/arm/examples/olympics/olympics1932.txt

1. Construct a 7 × 7 × 2 array of the data (ordered by skater, judge, and judging criterion).

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

pair	criterion	judge	score
1	Program	judge_1	5.6
1	Performance	judge_1	5.6
2	Program	judge_1	5.5
2	Performance	judge_1	5.5
3	Program	judge_1	6.0
3	Performance	judge_1	6.0
4	Program	judge_1	5.6
4	Performance	judge_1	5.6
5	Program	judge_1	5.4
5	Performance	judge_1	4.8
6	Program	judge_1	5.2
6	Performance	judge_1	4.8
7	Program	judge_1	4.8
7	Performance	judge_1	4.3

pair	criterion	judge	score
1	Program	judge_2	5.5
1	Performance	judge_2	5.5
2	Program	judge_2	5.2
2	Performance	judge_2	5.7
3	Program	judge_2	5.3
3	Performance	judge_2	5.5
4	Program	judge_2	5.3
4	Performance	judge_2	5.3
5	Program	judge_2	4.5
5	Performance	judge_2	4.8
6	Program	judge_2	5.1
6	Performance	judge_2	5.6
7	Program	judge_2	4.0
7	Performance	judge_2	4.6
1	Program	judge_3	5.8
1	Performance	judge_3	5.8
2	Program	judge_3	5.8
2	Performance	judge_3	5.6
3	Program	judge_3	5.8
3	Performance	judge_3	5.7
4	Program	judge_3	5.8
4	Performance	judge_3	5.8
5	Program	judge_3	5.8
5	Performance	judge_3	5.5
6	Program	judge_3	5.3
6	Performance	judge_3	5.0
7	Program	judge_3	4.7
7	Performance	judge_3	4.5
1	Program	judge_4	5.3
1	Performance	judge_4	4.7
2	Program	judge_4	5.8
2	Performance	judge_4	5.4
3	Program	judge_4	5.0
3	Performance	judge_4	4.9
4	Program	judge_4	4.4
4	Performance	judge_4	4.8
5	Program	judge_4	4.0
5	Performance	judge_4	4.4
6	Program	judge_4	5.4
6	Performance	judge_4	4.7
7	Program	judge_4	4.0
7	Performance	judge_4	4.0
1	Program	judge_5	5.6
1	Performance	judge_5	5.7
2	Program	judge_5	5.6
2	Performance	judge_5	5.5
3	Program	judge_5	5.4
3	Performance	judge_5	5.5
4	Program	judge_5	4.5
4	Performance	judge_5	4.5
5	Program	judge_5	5.5
5	Performance	judge_5	4.6

pair	criterion	judge	score
6	Program	judge_5	4.5
6	Performance	judge_5	4.0
7	Program	judge_5	3.7
7	Performance	judge_5	3.6
1	Program	judge_6	5.2
1	Performance	judge_6	5.3
2	Program	judge_6	5.1
2	Performance	judge_6	5.3
3	Program	judge_6	5.1
3	Performance	judge_6	5.2
4	Program	judge_6	5.0
4	Performance	judge_6	5.0
5	Program	judge_6	4.8
5	Performance	judge_6	4.8
6	Program	judge_6	4.5
6	Performance	judge_6	4.6
7	Program	judge_6	4.0
7	Performance	judge_6	4.0
1	Program	judge_7	5.7
1	Performance	judge_7	5.4
2	Program	judge_7	5.8
2	Performance	judge_7	5.7
3	Program	judge_7	5.3
3	Performance	judge_7	5.7
4	Program	judge_7	5.1
4	Performance	judge_7	5.5
5	Program	judge_7	5.5
5	Performance	judge_7	5.2
6	Program	judge_7	5.0
6	Performance	judge_7	5.2
7	Program	judge_7	4.8
7	Performance	judge_7	4.8

2. Reformulate the data as a 98×4 array (similar to the top table in Figure 11.7), where the first two columns are the technical merit and artistic impression scores, the third column is a skater ID, and the fourth column is a judge ID.

```
reformu_data<-spread(melt_olympics,criterion,score)
```

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.

```
pairs<-data.frame(pairs=c(1,2,3,4,5,6,7), country=c("France","United States","Hungary","Hungary","Canada
judges<-data.frame(judges=c(1,2,3,4,5,6,7), country=c("Hungary","Norway","Austria","Finland","France","(

pair_country<-merge(reformu_data,pairs,by.x="pair",by.y="pairs")
colnames(pair_country)[5]<-"pair_country"
judge_country<-merge(pair_country,judges,by.x="pair",by.y="judges")
colnames(judge_country)[6]<-"judge_country"
judge_country$pair_country<-as.character(judge_country$pair_country)
judge_country$judge_country<-as.character(judge_country$judge_country)

country_data<-judge_country %>%
```

```
  mutate(country = 1*(pair_country==judge_country)) %>%
  dplyr::select(pair,judge,Performance,Program,country)

kable(country_data)
```

pair	judge	Performance	Program	country
1	judge_1	5.6	5.6	0
1	judge_2	5.5	5.5	0
1	judge_3	5.8	5.8	0
1	judge_4	4.7	5.3	0
1	judge_5	5.7	5.6	0
1	judge_6	5.3	5.2	0
1	judge_7	5.4	5.7	0
2	judge_1	5.5	5.5	0
2	judge_2	5.7	5.2	0
2	judge_3	5.6	5.8	0
2	judge_4	5.4	5.8	0
2	judge_5	5.5	5.6	0
2	judge_6	5.3	5.1	0
2	judge_7	5.7	5.8	0
3	judge_1	6.0	6.0	0
3	judge_2	5.5	5.3	0
3	judge_3	5.7	5.8	0
3	judge_4	4.9	5.0	0
3	judge_5	5.5	5.4	0
3	judge_6	5.2	5.1	0
3	judge_7	5.7	5.3	0
4	judge_1	5.6	5.6	0
4	judge_2	5.3	5.3	0
4	judge_3	5.8	5.8	0
4	judge_4	4.8	4.4	0
4	judge_5	4.5	4.5	0
4	judge_6	5.0	5.0	0
4	judge_7	5.5	5.1	0
5	judge_1	4.8	5.4	0
5	judge_2	4.8	4.5	0
5	judge_3	5.5	5.8	0
5	judge_4	4.4	4.0	0
5	judge_5	4.6	5.5	0
5	judge_6	4.8	4.8	0
5	judge_7	5.2	5.5	0
6	judge_1	4.8	5.2	0
6	judge_2	5.6	5.1	0
6	judge_3	5.0	5.3	0
6	judge_4	4.7	5.4	0
6	judge_5	4.0	4.5	0
6	judge_6	4.6	4.5	0
6	judge_7	5.2	5.0	0
7	judge_1	4.3	4.8	1
7	judge_2	4.6	4.0	1
7	judge_3	4.5	4.7	1
7	judge_4	4.0	4.0	1
7	judge_5	3.6	3.7	1

pair	judge	Performance	Program	country
7	judge_6	4.0	4.0	1
7	judge_7	4.8	4.8	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().

```
model1<-lmer(Program~(1|pair) + (1|judge),data=country_data)
display(model1)
```

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

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

```
model2<-lmer(Performance~(1|pair) + (1|judge),data=country_data)
display(model2)
```

```
## lmer(formula = Performance ~ (1 | pair) + (1 | judge), data = country_data)
## coef.est  coef.se
##     5.09     0.20
##
## Error terms:
##  Groups   Name        Std.Dev.
##  pair     (Intercept) 0.45
##  judge    (Intercept) 0.28
##  Residual             0.27
## ---
## number of obs: 49, groups: pair, 7; judge, 7
## AIC = 54.2, DIC = 43.4
## deviance = 44.8
```

6. Display your results for both outcomes graphically.

```
#wrong
coef_pair_2<-coef(model2)$pair
coef_judge_2<-coef(model2)$judge
for(i in 1:7){
  s<-coef_pair_2[i,1] + coef_judge_2[,1]
  if(i==1)
    score<-s
  else
    score<-c(score,s)
}
```

```r
pair<-country_data$pair
judge<-country_data$judge
est<-cbind(pair,judge,score)
```

7. (optional) Use posterior predictive checks to investigate model fit in (4) and (5).

Different ways to write the model:

Using any data that are appropriate for a multilevel model, write the model in the five ways discussed in Section 12.5 of Gelman and Hill.

```r
lmer(formula=hiv.data$y~hiv.data$time+hiv.data$age.baseline+hiv.data$treatment+(1|hiv.data$newpid))
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## hiv.data$y ~ hiv.data$time + hiv.data$age.baseline + hiv.data$treatment +
##     (1 | hiv.data$newpid)
## REML criterion at convergence: 2840.543
## Random effects:
##  Groups          Name        Std.Dev.
##  hiv.data$newpid (Intercept) 1.3670
##  Residual                    0.7641
## Number of obs: 978, groups:  hiv.data$newpid, 226
## Fixed Effects:
##           (Intercept)          hiv.data$time  hiv.data$age.baseline
##                4.7639                -0.3822                -0.1183
##    hiv.data$treatment
##                0.2944
```

Original formula of the multilevel model:

$$y = \beta_{0[j]i} + X_{i1} * \beta_{1[j]i} + X_{i2} * \beta_{2[j]i} + X_{i3} * \beta_{3[j]i} + \epsilon_i$$

$$y = \alpha_{j[i]} + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \epsilon_i$$

$$\alpha_j \sim \ N(\mu_i, \sigma_i^2)$$

$$X1 = time,\ X2 = age.baseline,\ X3 = treatment$$

.

** #1 Method: Allowing regression coefficeints to vary accross groups**

$$y = 4.91 + X_{i1} * (-0.36) + X_{i2} * (-0.12) + X_{i3} * 0.18 + 0.77\ ,\ for\ i = 1, ..., n_{250}$$

$$\alpha_j \sim \ \mathrm{N}(0, 1.37^2)$$

** #2 Method: Combining separate local regressions**

$$y \sim \ N(4.91 + X_{i1} * (-0.36) + X_{i2} * (-0.12) + X_{i3} * 0.18,\ 0.77^2)\ ,\ for\ i = 1, ..., n_{250}$$

$$\alpha_j \sim \text{ N}(random\ intercept, 1.37^2)$$

** #3 Method: Modeling the coefficients of a large regression model**

$$y_i \sim\ N(4.91 + X_{i1} * (-0.36) + X_{i2} * (-0.12) + X_{i3} * 0.18,\ 0.77^2)$$

$$\beta_j \sim\ N(0, 1.37^2)$$

** #4 Method: Regression with multiple error terms**

$$y_i \sim\ N(4.91 + X_{i1} * (-0.36) + X_{i2} * (-0.12) + X_{i3} * 0.18 + 1.37^2,\ 0.77^2)$$

** #5 Method: Large regression with correlated errors**

$$y_i \sim\ N(4.91 + X_{i1} * (-0.36) + X_{i2} * (-0.12) + X_{i3} * 0.18,\ 1.37^2 + 0.77^2)$$

Models for adjusting individual ratings:

A committee of 10 persons is evaluating 100 job applications. Each person on the committee reads 30 applications (structured so that each application is read by three people) and gives each a numerical rating between 1 and 10.

lmer(rating_scores~applicants_ID+raters_ID+(1|raters_ID))

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

lmer(rating_scores~applicants_ID+raters_ID+(1+raters_ID|raters_ID))