# homework 07

*Weiling Li*

*November 8, 2019*

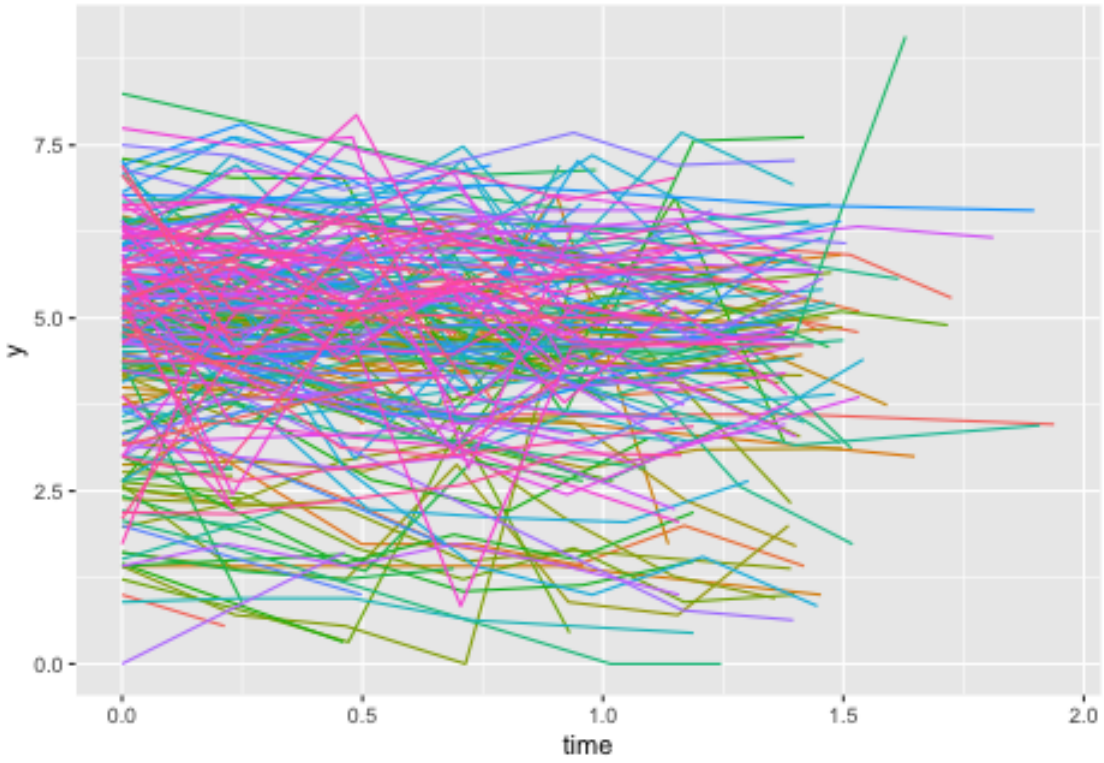
## Data analysis

### CD4 percentages for HIV infected kids

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

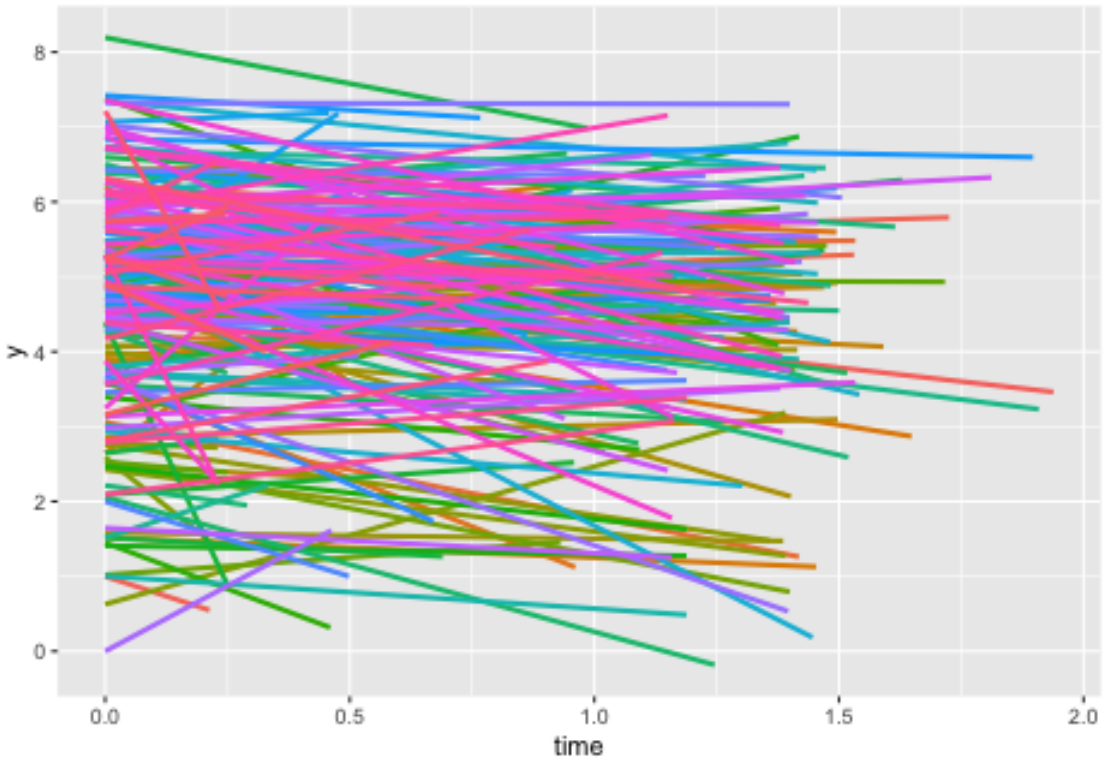
1. Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

```
ggplot(hiv.data)+aes(x = time, y = y, color = factor(newpid))+geom_line(show.legend = FALSE)
```




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.

```
ggplot(hiv.data)+aes(x = time, y = y, color = factor(newpid))+
  #geom_point(show.legend = FALSE)+
  geom_smooth(method = "lm",se = FALSE,show.legend = FALSE)
```




```
fit.lm <- lm(data = hiv.data,y~time+factor(newpid)-1)
lm.coef <- data.frame(coef(fit.lm)[-1])
lm.coef.time <- coef(fit.lm)[1]
```

3. Set up a model for the children's slopes and intercepts as a function of the treatment and age at baseline. Estimate this model using the two-step procedure–first estimate the intercept and slope separately for each child, then fit the between-child models using the point estimates from the first step.

```
fit.1 <- lm(data = hiv.data, y~ time +factor(newpid)-1  )
#summary(fit.1)
id <- hiv.data$newpid%>%unique()
alphas <- coef(fit.1)[-1]

group.alpha <- data.frame(cbind(id,alphas))
colnames( group.alpha) <- c("newpid","alpha")
group.data <- hiv.data%>%dplyr::select(newpid,treatment,age.baseline)%>%dplyr::distinct()
group.data.1 <- dplyr::left_join(group.alpha,group.data)
```

```
## Joining, by = "newpid"
```

```
fit.2 <- lm(data = group.data.1,alpha~factor(treatment)+age.baseline)
summary(fit.2)
```

```
##
## Call:
## lm(formula = alpha ~ factor(treatment) + age.baseline, data = group.data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1594 -0.7039  0.2265  1.1215  2.7256
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.10627    0.18728  27.265  < 2e-16 ***
## factor(treatment)2   0.14558    0.18421   0.790  0.43012
## age.baseline        -0.12088    0.04023  -3.005  0.00293 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.455 on 247 degrees of freedom
## Multiple R-squared:  0.03753,    Adjusted R-squared:  0.02974
## F-statistic: 4.816 on 2 and 247 DF,  p-value: 0.008875
```

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.

```
display(fit.3 <- lmer(data = hiv.data, y~ time + (1|newpid)))
```

```
## lmer(formula = y ~ time + (1 | newpid), data = hiv.data)
##             coef.est coef.se
## (Intercept)  4.76     0.10
## time        -0.37     0.05
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.40
##  Residual             0.77
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3148.8, DIC = 3126.9
## deviance = 3133.9
```

```
lmer.coef.1 <- coef(fit.3)$newpid
#hiv.data1 <- hiv.data%>%dplyr::mutate(newpid = factor(newpid))
#summary(fit.4 <- lmer(data = hiv.data1, y~ time + (1|newpid)))
```

The coefficient of time stats that for any child within or outside the groups, 1 unit increase in `time` will the sqrt of `y` is expected to decrease -0.36609.

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.

```
#hiv.data.1 <- hiv.data1%>%dplyr::mutate(treatment = factor(treatment))
display(fit.4 <- lmer(data = hiv.data, y~ time + treatment + age.baseline +  (1|newpid)))
```

```
## lmer(formula = y ~ time + treatment + age.baseline + (1 | newpid),
##     data = hiv.data)
##              coef.est coef.se
## (Intercept)   4.91     0.32
## time         -0.36     0.05
## treatment     0.18     0.18
## age.baseline -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.37
##  Residual             0.77
```

```
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3149.2, DIC = 3110.9
## deviance = 3124.1
```

```
lmer.coef.2 <- coef(fit.4)$newpid
```

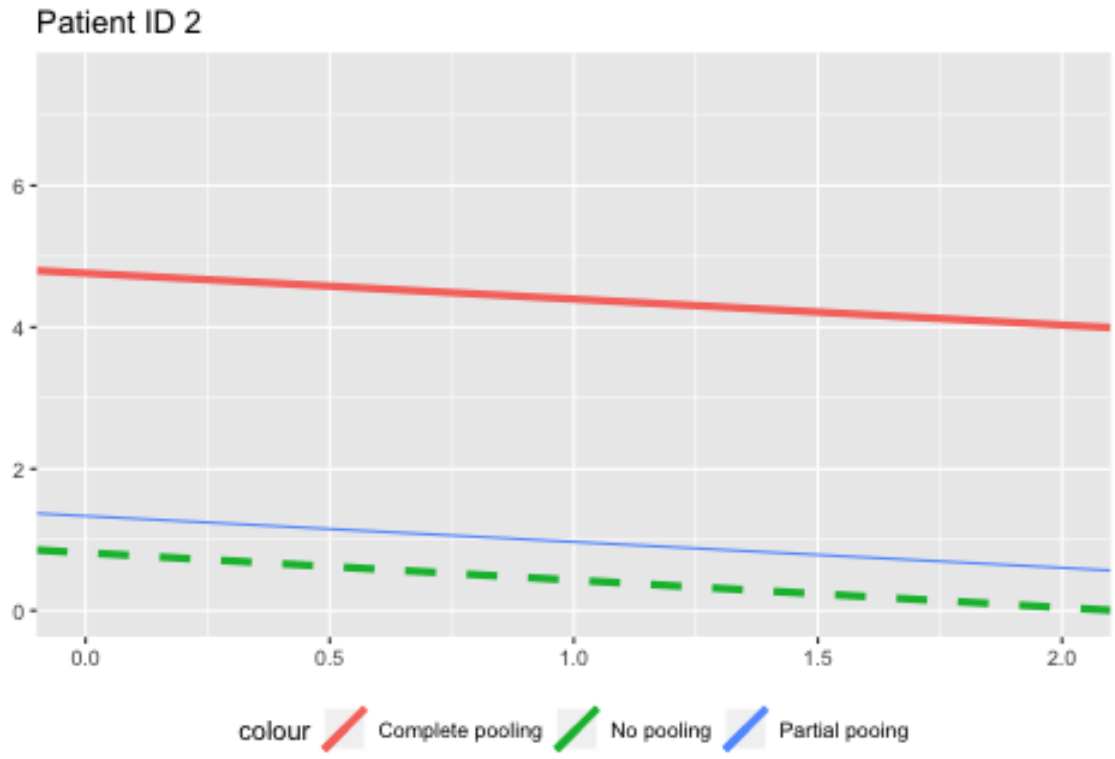
The coefficient of time stays the same as before with little value changed to -0.36216, and need to hold other variables at constant. The treatment coef states that by holding time and age.baseline constant, for treatment 2 patient, the square root of CD4 percentage will increase 0.18, while holding other variables as constant. The age.baseline's coef stats that, by holding other vairables at constant, 1 unit increase in age.baseline is associated with 0.11945 decrease of square root CD4 percentage.

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

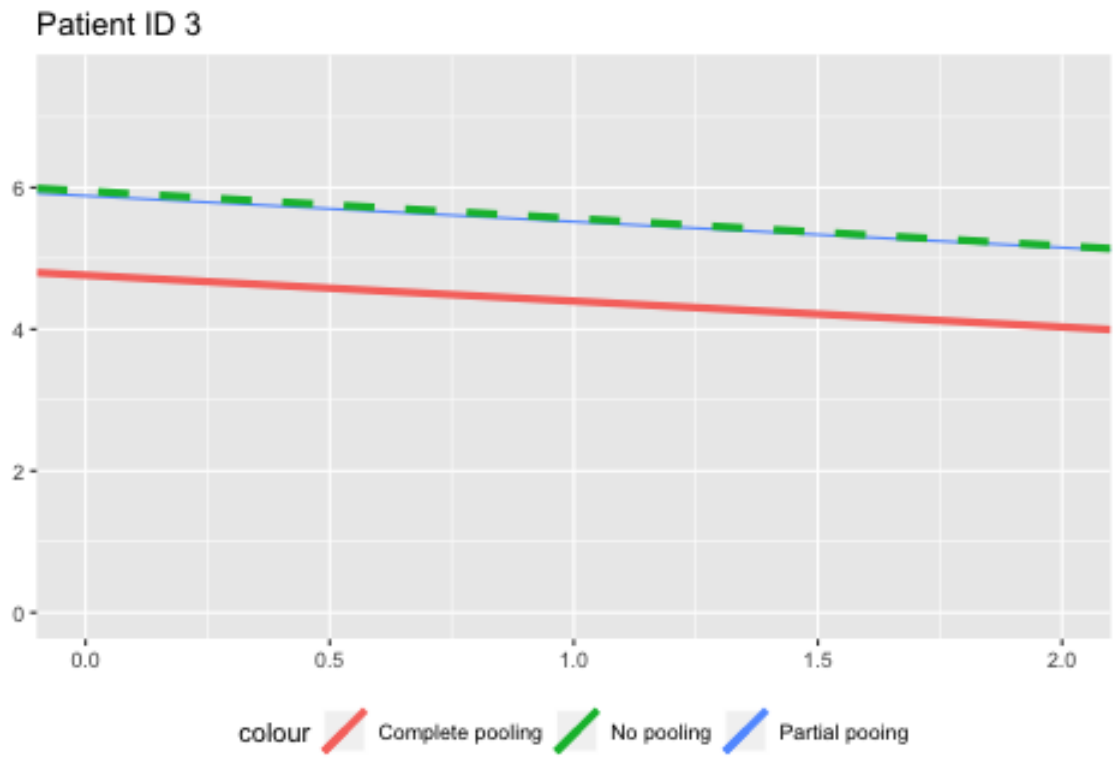
Numerically, the pooling effect does not change much, model from question (4) has $\sigma_\alpha^2 : \sigma_y^2 = 1.40^2 : 0.77^2 = 3.3$ which means that the group level estimate is less meaningful than $1/3.3 = 0.30$ individual observation. In another word, individual level prediction is more reliable. model from question (5) has $\sigma_\alpha^2 : \sigma_y^2 = 1.37^2 : 0.77^2 = 3.16$ which is essentially saying the same thing, while the pooling effect is stronger than (4), but the actual effect is miniscule.

graphically, for almost all patient, we should expect that there will be so little pooling effect, 1st is the time and varing intercept plot for two of the patient

```
#y_hat_1 <- predict(fit.3)
#y_hat_2 <- predict(fit.4)


newpid <- hiv.data%>%dplyr::select(newpid)%>%dplyr::mutate(newpid = as.character(newpid))%>%dplyr::dist

lmer.coef.1.add <- dplyr::mutate(lmer.coef.1,newpid = as.numeric(newpid))
lmer.coef.1.pool <- fixef(fit.3)
lmer.coef.2.add <- dplyr::mutate(lmer.coef.2,newpid = as.numeric(newpid))
lmer.coef.2.pool <- fixef(fit.4)

newpid.info <- dplyr::select(group.data.1,newpid,treatment,age.baseline)
colnames(newpid.info) <- c("newpid","treatment.pid","age.baseline.pid")
colnames(lmer.coef.1.add) <- c("intercept","time","newpid")
intercept <- lmer.coef.1.add$intercept

ggplot(lmer.coef.1.add)+
  geom_abline(aes(intercept = intercept[2] ,slope = lmer.coef.1.add$time[2],color = "Partial pooing"))+
  geom_abline(aes(intercept = lmer.coef.1.pool[1],slope = lmer.coef.1.pool[2],color = "Complete pooling
  geom_abline(aes(intercept = lm.coef$coef.fit.lm...1.[2],slope = lm.coef.time,color = "No pooling"),li
  ylim(c(0,7.5))+
  xlim(c(0,2))+
  theme(legend.position="bottom")+
  ggtitle("Patient ID 2")
```

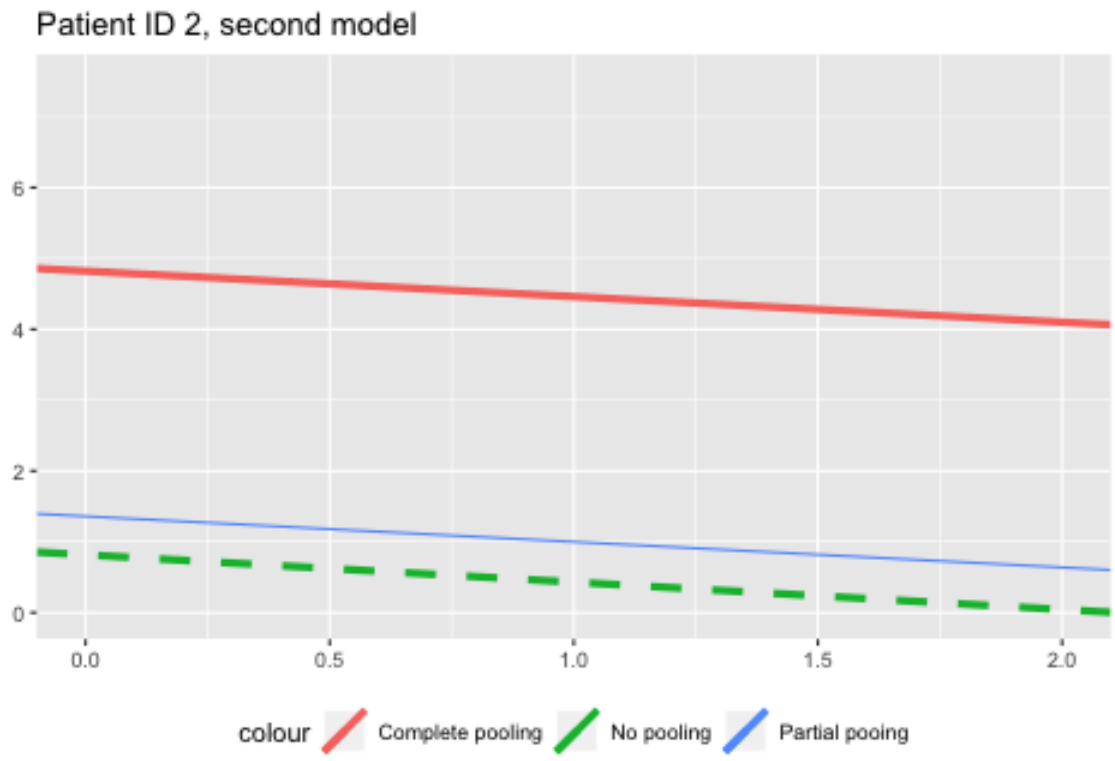



```
ggplot(lmer.coef.1.add)+
  geom_abline(aes(intercept = intercept[3] ,slope = lmer.coef.1.add$time[3],color = "Partial pooing"))+
  geom_abline(aes(intercept = lmer.coef.1.pool[1],slope = lmer.coef.1.pool[2],color = "Complete pooling
  geom_abline(aes(intercept = lm.coef$coef.fit.lm...1.[3],slope = lm.coef.time,color = "No pooling"),li
  ylim(c(0,7.5))+
  xlim(c(0,2))+
  theme(legend.position="bottom")+
  ggtitle("Patient ID 3")
```

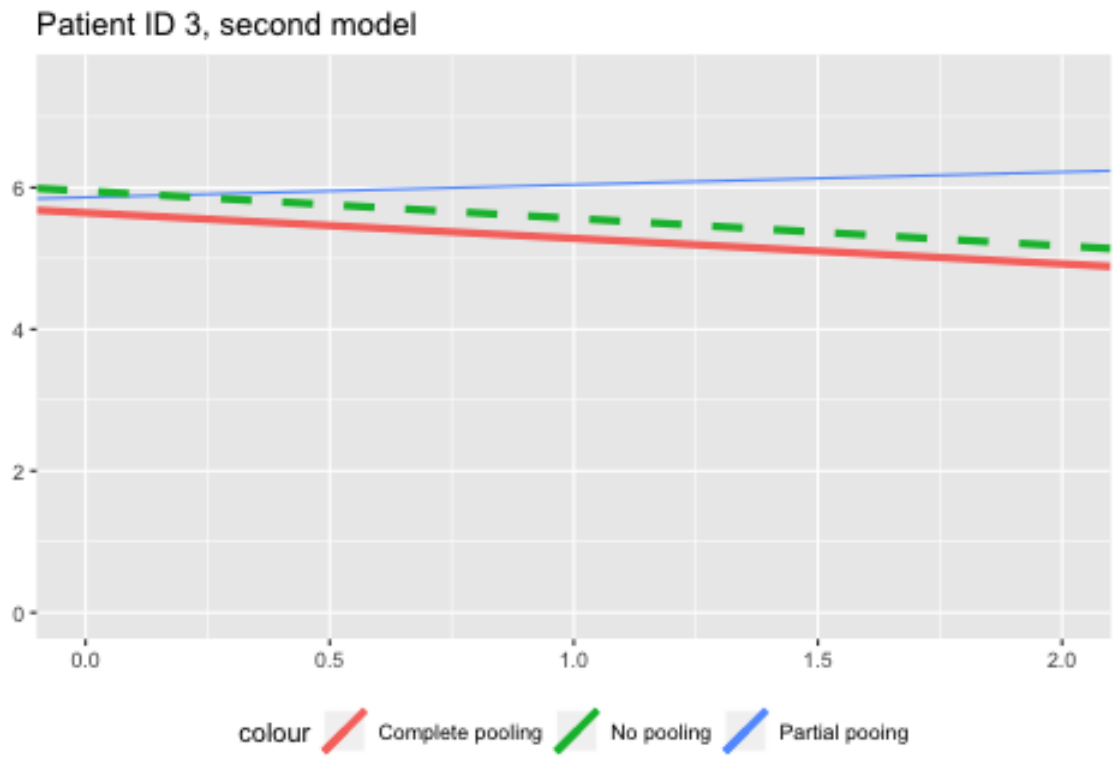



The same happened to the later model as well

```
lmer.coef.2.tot <- dplyr::left_join(lmer.coef.2.add,newpid.info,by = "newpid")%>%dplyr::mutate(intercep

intercept.2 <- lmer.coef.2.tot%>%dplyr::select(intercept)%>%dplyr::pull()
intercept.3 <- lmer.coef.2.tot%>%dplyr::select(intercept.pool)%>%dplyr::pull()

ggplot(lmer.coef.1.add)+
  geom_abline(aes(intercept = intercept.2[2] ,slope = lmer.coef.2.pool[2],color = "Partial pooing"))+
  geom_abline(aes(intercept = intercept.3[2],slope = lmer.coef.2.pool[2],color = "Complete pooling"),si
  geom_abline(aes(intercept = lm.coef$coef.fit.lm...1.[2],slope = lm.coef.time,color = "No pooling"),li
  ylim(c(0,7.5))+
  xlim(c(0,2))+
  theme(legend.position="bottom")+
  ggtitle("Patient ID 2, second model")
```




```
ggplot(lmer.coef.1.add)+
  geom_abline(aes(intercept = intercept.2[3] ,slope = lmer.coef.2.pool[3],color = "Partial pooing"))+
  geom_abline(aes(intercept = intercept.3[3],slope = lmer.coef.2.pool[2],color = "Complete pooling"),si
  geom_abline(aes(intercept = lm.coef$coef.fit.lm...1.[3],slope = lm.coef.time,color = "No pooling"),li
  ylim(c(0,7.5))+
  xlim(c(0,2))+
  theme(legend.position="bottom")+
  ggtitle("Patient ID 3, second model")
```




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.

Assume the next hypothetical time point is 2(because the largest time point is 1.9x)

```
new.pred <- newpid.info%>%dplyr::mutate(time = 2)
colnames(new.pred) <- c("newpid","treatment","age.baseline","time")

new.pred <- new.pred%>%dplyr::mutate(y.hat = predict(fit.4,new.pred))

## Calculate the point of prediction of each patient at time = 2, print the 1st 10
y.time2.mean = predict(fit.4,new.pred)
print(y.time2.mean[1:5])
```

```
##         1         2         3         4         5
## 3.8212979 0.6375377 5.1373054 4.8355881 3.5176450
```

```
sigma.y = sigma.hat(fit.4)$sigma$data
y.sim = c()
for (i in 1:length(y.time2.mean)){
 y.sim[i] =rnorm(1,y.time2.mean[i],sigma.y)
}

## sample of simulated y
print(y.sim[1:10])
```

```
##  [1]  3.317463 -1.154003  5.645243  5.890527  2.851529  5.196439  4.762933
##  [8]  4.182227  5.805442  5.423467
```

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.

Assuming treatment is 1.

```
time.Q8 = seq(0,2,.5)
age.baseline.Q8 = rep(4,length(time.Q8))
treatment.Q8 <- rep(1,length(time.Q8))
Q8.table <- matrix(cbind(time.Q8,treatment.Q8,age.baseline.Q8),ncol = 3)
Q8.table
```

```
##      [,1] [,2] [,3]
## [1,]  0.0    1    4
## [2,]  0.5    1    4
## [3,]  1.0    1    4
## [4,]  1.5    1    4
## [5,]  2.0    1    4
```

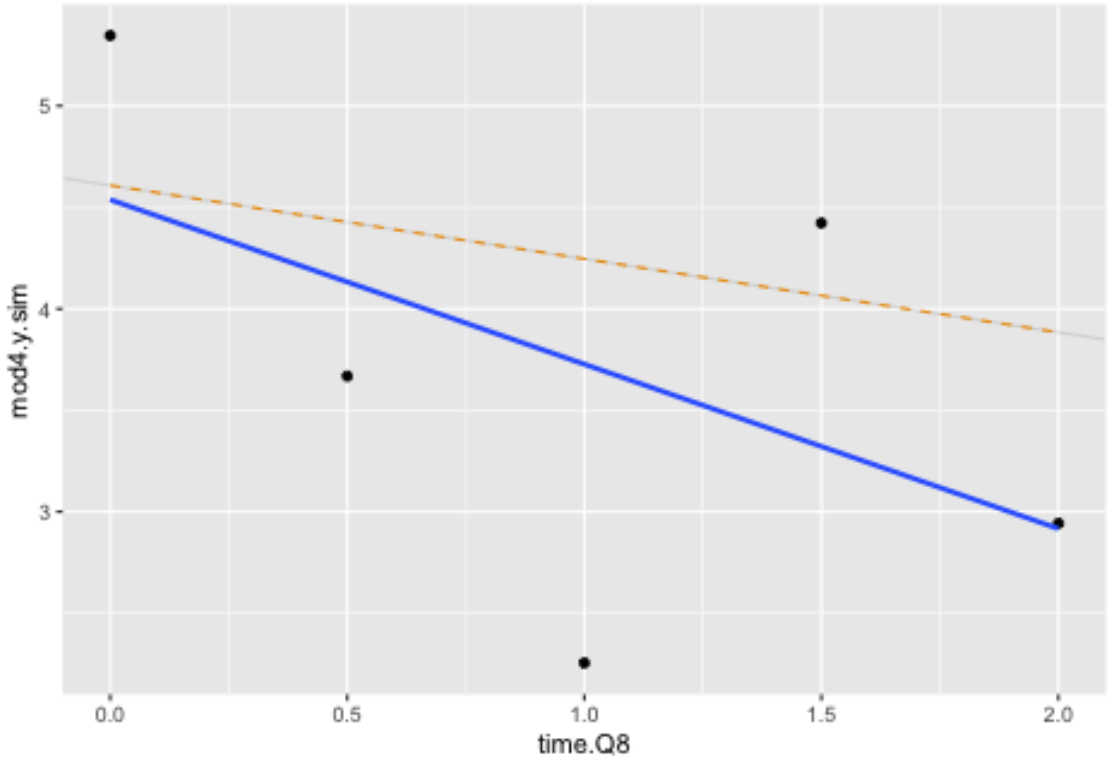
```
#colnames(Q8.table) <- c("time","age.baseline","treatment")
fixef(fit.4)
```

```
##  (Intercept)         time    treatment age.baseline
##    4.9060559   -0.3621573    0.1800822   -0.1194538
```

```
mod4.intercept <- as.numeric(fixef(fit.4)[1])
mod4.coef <- matrix(fixef(fit.4)[-1],ncol = 1)
mod4.coef
```

```
##            [,1]
## [1,] -0.3621573
## [2,]  0.1800822
## [3,] -0.1194538
```

```
mod4.y.hat <- mod4.intercept+Q8.table%*%mod4.coef

mod4.y.sim <- c()


for (i in 1:length(mod4.y.hat)){
 mod4.y.sim[i] =rnorm(1,mod4.y.hat[i],sigma.y)
}

ggplot()+
  aes(x = time.Q8,y = mod4.y.sim)+
  geom_point()+geom_smooth(method = "lm",se = FALSE)+ #The no pooling line estimated line
  geom_line(aes(y = mod4.y.hat),color = "orange",linetype = "dashed")+ # the real generating line
  geom_abline(intercept = mod4.y.hat[1],slope = lmer.coef.2.pool[2],alpha = .1) # complete pooled
```

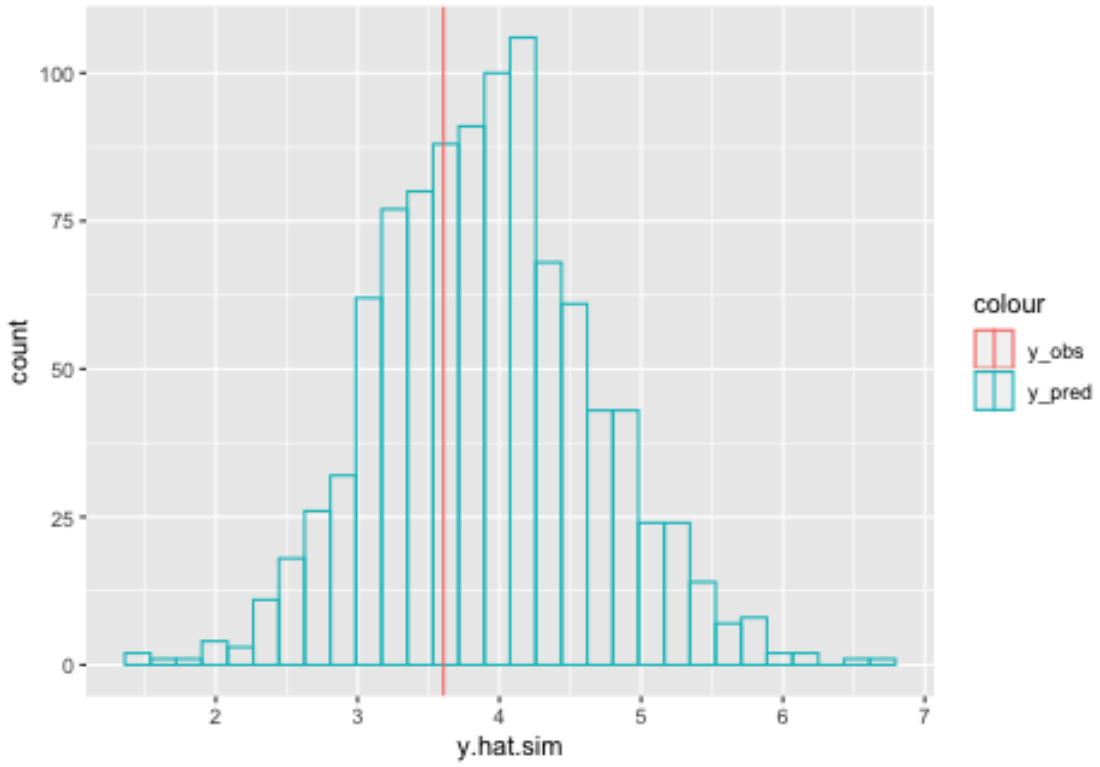



9. Posterior predictive checking: continuing the previous exercise, use the fitted model from (5) to simulate a new dataset of CD4 percentages (with the same sample size and ages of the original dataset) for the final time point of the study, and record the average CD4 percentage in this sample. Repeat this process 1000 times and compare the simulated distribution to the observed CD4 percentage at the final time point for the actual data.

```
Q9.tabe <- hiv.data%>%dplyr::mutate(flag = dplyr::lead(newpid)-newpid)%>%
  dplyr::filter(is.na(flag)|flag != 0)%>%
  dplyr::select(y,newpid,time,treatment,age.baseline)

Q9.tabe.1 <- lmer.coef.2.tot%>%dplyr::select(newpid,intercept)

Q9.tabe.1 <- dplyr::left_join(Q9.tabe,Q9.tabe.1)%>%dplyr::mutate(y.hat = intercept - time*lmer.coef.2.p
```

```
## Joining, by = "newpid"
```

```
## pick a patient
pid <- sample(Q9.tabe.1$newpid,1)
## gather patient info
info.pid <- Q9.tabe.1%>%dplyr::filter(newpid == pid)
y.pid = info.pid$y
y.hat.base <- info.pid$y.hat
y.hat.sim <- c()
for(i in 1:1000){
  y.hat.sim[i] <- rnorm(1,y.hat.base,sigma.y)
}
ggplot()+aes(x = y.hat.sim)+
  geom_histogram(bins = 30, alpha = .7,aes(color = "y_pred"),fill = NA)+ ## distribution of y_pred
  geom_vline(aes(xintercept = y.pid,color = "y_obs"))
```




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

```
display(fit.5 <- lmer(data = hiv.data, y~ time + treatment + age.baseline +  (1+time|newpid)))
```

```
## lmer(formula = y ~ time + treatment + age.baseline + (1 + time |
##     newpid), data = hiv.data)
##              coef.est coef.se
## (Intercept)   4.95     0.31
## time         -0.35     0.07
## treatment     0.16     0.18
## age.baseline -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev. Corr
##  newpid   (Intercept) 1.36
##           time        0.58     -0.04
##  Residual             0.72
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3123, DIC = 3081.6
## deviance = 3094.3
```

```
display(fit.7 <- lmer(data = hiv.data, y~ time + treatment + age.baseline +  (time-1|newpid)+(1|newpid)))
```

```
## lmer(formula = y ~ time + treatment + age.baseline + (time -
##     1 | newpid) + (1 | newpid), data = hiv.data)
##              coef.est coef.se
## (Intercept)   4.96     0.31
## time         -0.36     0.07
## treatment     0.16     0.18
```

```
## age.baseline -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   time        0.57
##  newpid.1 (Intercept) 1.35
##  Residual             0.72
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3121.1, DIC = 3081.6
## deviance = 3094.4
```

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).

```
display(fit.6 <- lmer(data = hiv.data, y~ factor(round(time,2)) + treatment + age.baseline +  (1|newpid)
```

```
## lmer(formula = y ~ factor(round(time, 2)) + treatment + age.baseline +
##     (1 | newpid), data = hiv.data)
##                             coef.est coef.se
## (Intercept)                  4.90     0.32
## factor(round(time, 2))0.2   -1.13     0.69
## factor(round(time, 2))0.21  -0.34     0.52
## factor(round(time, 2))0.22   0.01     0.29
## factor(round(time, 2))0.23  -0.08     0.12
## factor(round(time, 2))0.24   0.02     0.18
## factor(round(time, 2))0.25  -0.18     0.14
## factor(round(time, 2))0.26  -0.11     0.34
## factor(round(time, 2))0.27  -0.07     0.36
## factor(round(time, 2))0.29   0.17     0.42
## factor(round(time, 2))0.3   -0.04     0.86
## factor(round(time, 2))0.31   0.02     0.52
## factor(round(time, 2))0.33  -0.52     0.57
## factor(round(time, 2))0.36   0.12     0.51
## factor(round(time, 2))0.43  -0.40     0.61
## factor(round(time, 2))0.44  -0.24     0.51
## factor(round(time, 2))0.45  -0.28     0.50
## factor(round(time, 2))0.46  -0.22     0.13
## factor(round(time, 2))0.47  -0.01     0.21
## factor(round(time, 2))0.48  -0.07     0.17
## factor(round(time, 2))0.49   0.32     0.32
## factor(round(time, 2))0.5   -0.35     0.20
## factor(round(time, 2))0.51  -0.63     0.38
## factor(round(time, 2))0.52  -0.41     0.34
## factor(round(time, 2))0.53  -0.27     0.41
## factor(round(time, 2))0.54  -0.61     0.45
## factor(round(time, 2))0.56   0.21     0.45
## factor(round(time, 2))0.57  -0.73     0.90
## factor(round(time, 2))0.58   0.00     0.56
## factor(round(time, 2))0.59  -0.42     0.96
## factor(round(time, 2))0.61  -0.71     0.89
## factor(round(time, 2))0.64   1.80     0.96
## factor(round(time, 2))0.65  -1.54     0.65
## factor(round(time, 2))0.66  -2.01     0.92
## factor(round(time, 2))0.67  -0.61     0.40
```

```
## factor(round(time, 2))0.68  0.02     0.39
## factor(round(time, 2))0.69 -0.14     0.15
## factor(round(time, 2))0.7  -0.32     0.22
## factor(round(time, 2))0.71 -0.19     0.19
## factor(round(time, 2))0.72 -1.99     0.48
## factor(round(time, 2))0.73 -0.34     0.21
## factor(round(time, 2))0.74  0.00     0.34
## factor(round(time, 2))0.75 -0.52     0.32
## factor(round(time, 2))0.76 -0.11     0.39
## factor(round(time, 2))0.77 -0.65     0.33
## factor(round(time, 2))0.78  0.55     0.67
## factor(round(time, 2))0.79 -0.32     0.44
## factor(round(time, 2))0.8  -0.60     0.60
## factor(round(time, 2))0.81 -0.42     0.50
## factor(round(time, 2))0.82  0.12     0.67
## factor(round(time, 2))0.86 -0.50     0.64
## factor(round(time, 2))0.87  0.56     0.96
## factor(round(time, 2))0.88  0.45     0.68
## factor(round(time, 2))0.9   0.32     0.35
## factor(round(time, 2))0.91  0.29     0.33
## factor(round(time, 2))0.92 -0.55     0.20
## factor(round(time, 2))0.93 -0.69     0.24
## factor(round(time, 2))0.94 -0.13     0.18
## factor(round(time, 2))0.95  0.45     0.42
## factor(round(time, 2))0.96 -0.21     0.23
## factor(round(time, 2))0.97 -0.05     0.86
## factor(round(time, 2))0.98 -0.56     0.26
## factor(round(time, 2))0.99 -0.19     0.92
## factor(round(time, 2))1    -0.85     0.27
## factor(round(time, 2))1.01 -0.43     0.60
## factor(round(time, 2))1.02 -0.88     0.43
## factor(round(time, 2))1.03  0.21     0.51
## factor(round(time, 2))1.04 -0.58     0.50
## factor(round(time, 2))1.05 -0.55     0.43
## factor(round(time, 2))1.08 -0.71     0.87
## factor(round(time, 2))1.09 -0.64     0.42
## factor(round(time, 2))1.11  0.26     0.88
## factor(round(time, 2))1.12  0.76     0.96
## factor(round(time, 2))1.13  0.61     0.88
## factor(round(time, 2))1.14 -0.48     0.35
## factor(round(time, 2))1.15 -0.48     0.20
## factor(round(time, 2))1.16 -0.51     0.22
## factor(round(time, 2))1.17 -0.73     0.26
## factor(round(time, 2))1.18 -0.41     0.48
## factor(round(time, 2))1.19 -0.14     0.23
## factor(round(time, 2))1.2  -0.23     0.32
## factor(round(time, 2))1.21 -0.33     0.37
## factor(round(time, 2))1.22 -0.95     0.83
## factor(round(time, 2))1.23 -0.40     0.33
## factor(round(time, 2))1.25 -1.19     0.39
## factor(round(time, 2))1.26 -0.20     0.43
## factor(round(time, 2))1.27 -0.23     0.83
## factor(round(time, 2))1.28 -2.05     0.83
## factor(round(time, 2))1.3  -0.39     0.50
```

```
## factor(round(time, 2))1.31 -1.69     0.85
## factor(round(time, 2))1.34 -1.07     0.85
## factor(round(time, 2))1.35 -0.62     0.90
## factor(round(time, 2))1.36 -0.80     0.50
## factor(round(time, 2))1.37 -0.36     0.42
## factor(round(time, 2))1.38 -0.72     0.26
## factor(round(time, 2))1.39 -0.41     0.30
## factor(round(time, 2))1.4  -0.72     0.24
## factor(round(time, 2))1.41 -0.49     0.38
## factor(round(time, 2))1.42 -0.21     0.38
## factor(round(time, 2))1.43  0.07     0.43
## factor(round(time, 2))1.44 -0.37     0.60
## factor(round(time, 2))1.45 -1.37     0.49
## factor(round(time, 2))1.46 -0.33     0.34
## factor(round(time, 2))1.47  0.01     0.50
## factor(round(time, 2))1.48 -0.79     0.48
## factor(round(time, 2))1.49 -0.98     0.60
## factor(round(time, 2))1.5  -0.05     0.39
## factor(round(time, 2))1.51 -0.20     0.62
## factor(round(time, 2))1.52 -2.40     0.59
## factor(round(time, 2))1.53 -0.13     0.43
## factor(round(time, 2))1.54 -0.73     0.85
## factor(round(time, 2))1.59 -1.35     0.90
## factor(round(time, 2))1.61 -0.68     0.84
## factor(round(time, 2))1.63  3.54     0.83
## factor(round(time, 2))1.65 -1.30     0.87
## factor(round(time, 2))1.72 -0.35     0.61
## factor(round(time, 2))1.81  0.02     0.84
## factor(round(time, 2))1.9  -0.47     0.85
## factor(round(time, 2))1.91 -0.99     0.86
## factor(round(time, 2))1.94 -1.01     0.86
## treatment                   0.19     0.18
## age.baseline               -0.12     0.04
##
## Error terms:
##  Groups   Name        Std.Dev.
##  newpid   (Intercept) 1.38
##  Residual             0.76
## ---
## number of obs: 1072, groups: newpid, 250
## AIC = 3213.2, DIC = 2977.3
## deviance = 2971.2
```

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

```
print("varing intercept and slope, with correlated slope and intercept")
```

```
## [1] "varing intercept and slope, with correlated slope and intercept"
```

```
sigma.hat(fit.5)
```

```
## $sigma
## $sigma$data
## [1] 0.7172995
##
## $sigma$newpid
```

```
## (Intercept)        time
##   1.3588105   0.5808426
##
##
## $cors
## $cors$data
## [1] NA
##
## $cors$newpid
##             (Intercept)       time
## (Intercept)   1.0000000 -0.0410524
## time         -0.0410524  1.0000000
```

```
print("varing intercept and slope, with no correlation")
```

```
## [1] "varing intercept and slope, with no correlation"
```

```
sigma.hat(fit.7)
```

```
## $sigma
## $sigma$data
## [1] 0.7181354
##
## $sigma$newpid
##      time
## 0.5732483
##
## $sigma$newpid.1
## (Intercept)
##    1.352246
##
##
## $cors
## $cors$data
## [1] NA
##
## $cors$newpid
## [1] NA
##
## $cors$newpid.1
## [1] NA
```

```
print("factorized time")
```

```
## [1] "factorized time"
```

```
sigma.hat(fit.6)
```

```
## $sigma
## $sigma$data
## [1] 0.7608682
##
## $sigma$newpid
## (Intercept)
##    1.375155
##
##
```

```
## $cors
## $cors$data
## [1] NA
##
## $cors$newpid
## [1] NA
```

## 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).

```
olypics1932.split <- split(olympics1932, olympics1932$criterion)
```

2. Reformulate the data as a $98 \times 4$ 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.

```
Q2.table <- tidyr::pivot_longer(data = olympics1932,cols = 3:9,names_to = "judge_num",values_to = "score

kable(Q2.table,format = "latex",align = "c",caption = "Judge scores table")
```

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.

```
Q3.table <- Q2.table%>%dplyr::mutate(indicator = dplyr::if_else(pair == 3 & judge_num %in% "judge_1",1,0
```

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

$$Rating_{tech} \sim N(\overline{rating} + Skater_{j\{i\}} + Judge_{k\{i\}}), i = 1, ..., 49$$
$$Skater_{j\{i\}} \sim N(0, \sigma_S^2)$$
$$Judge_{k\{i\}} \sim N(0, \sigma_J^2)$$

```
display(fit.8 <- lmer(data  = Q2.table,Performance~1+(1|pair)+(1|judge_num)))
```

```
## lmer(formula = Performance ~ 1 + (1 | pair) + (1 | judge_num),
##     data = Q2.table)
## coef.est  coef.se
##     5.09     0.20
##
## Error terms:
##  Groups    Name        Std.Dev.
##  pair      (Intercept) 0.45
##  judge_num (Intercept) 0.28
##  Residual              0.27
## ---
## number of obs: 49, groups: pair, 7; judge_num, 7
## AIC = 54.2, DIC = 43.4
## deviance = 44.8
```

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

Table 1: Judge scores table

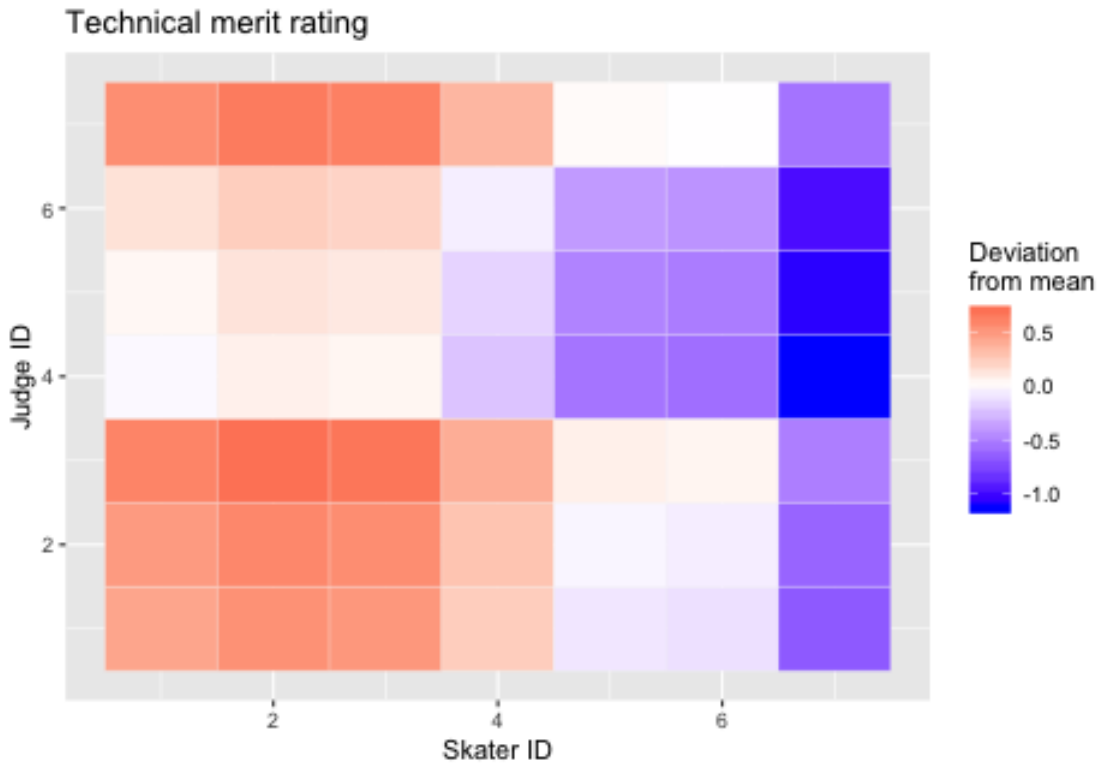
| Performance | Program | pair | judge_num |
|---|---|---|---|
| 5.6 | 5.6 | 1 | judge_1 |
| 5.5 | 5.5 | 1 | judge_2 |
| 5.8 | 5.8 | 1 | judge_3 |
| 4.7 | 5.3 | 1 | judge_4 |
| 5.7 | 5.6 | 1 | judge_5 |
| 5.3 | 5.2 | 1 | judge_6 |
| 5.4 | 5.7 | 1 | judge_7 |
| 5.5 | 5.5 | 2 | judge_1 |
| 5.7 | 5.2 |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

```
display(fit.9 <- lmer(data  = Q2.table,Program~1+(1|pair)+(1|judge_num)))
```

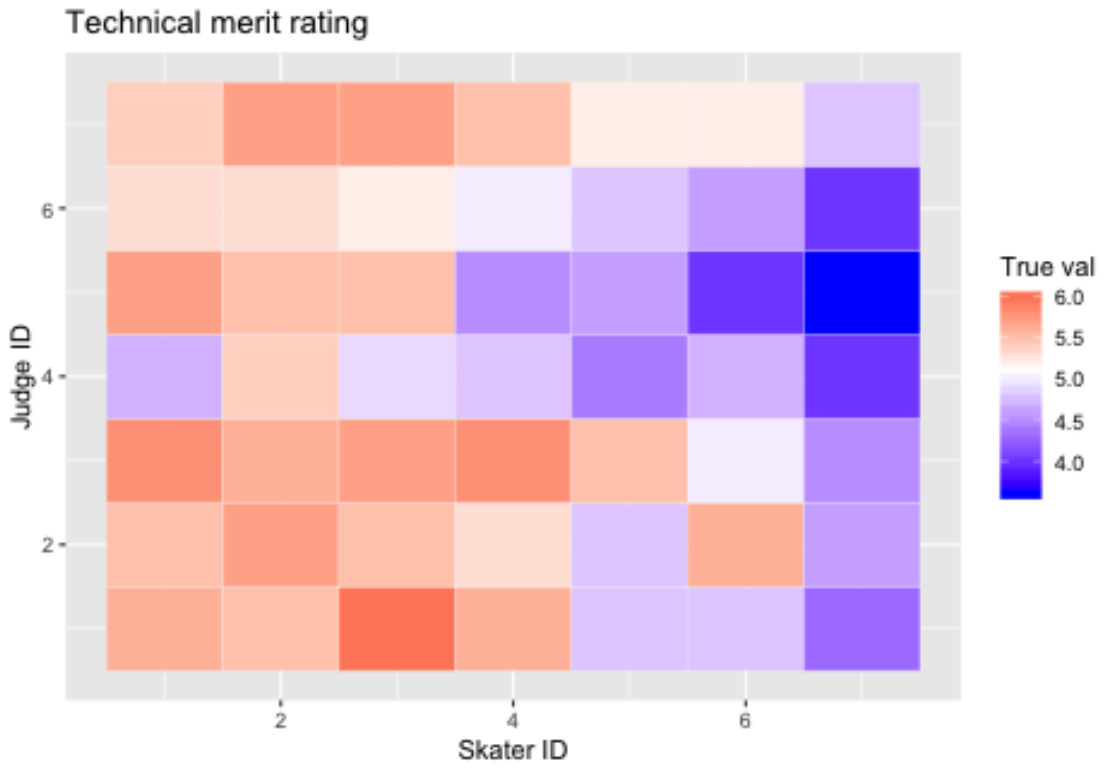
```
## lmer(formula = Program ~ 1 + (1 | pair) + (1 | judge_num), data = Q2.table)
## coef.est  coef.se
##     5.13     0.20
##
## Error terms:
##  Groups    Name        Std.Dev.
##  pair      (Intercept) 0.42
##  judge_num (Intercept) 0.28
##  Residual              0.33
## ---
## number of obs: 49, groups: pair, 7; judge_num, 7
## AIC = 68, DIC = 57
## deviance = 58.5
```

6. Display your results for both outcomes graphically.

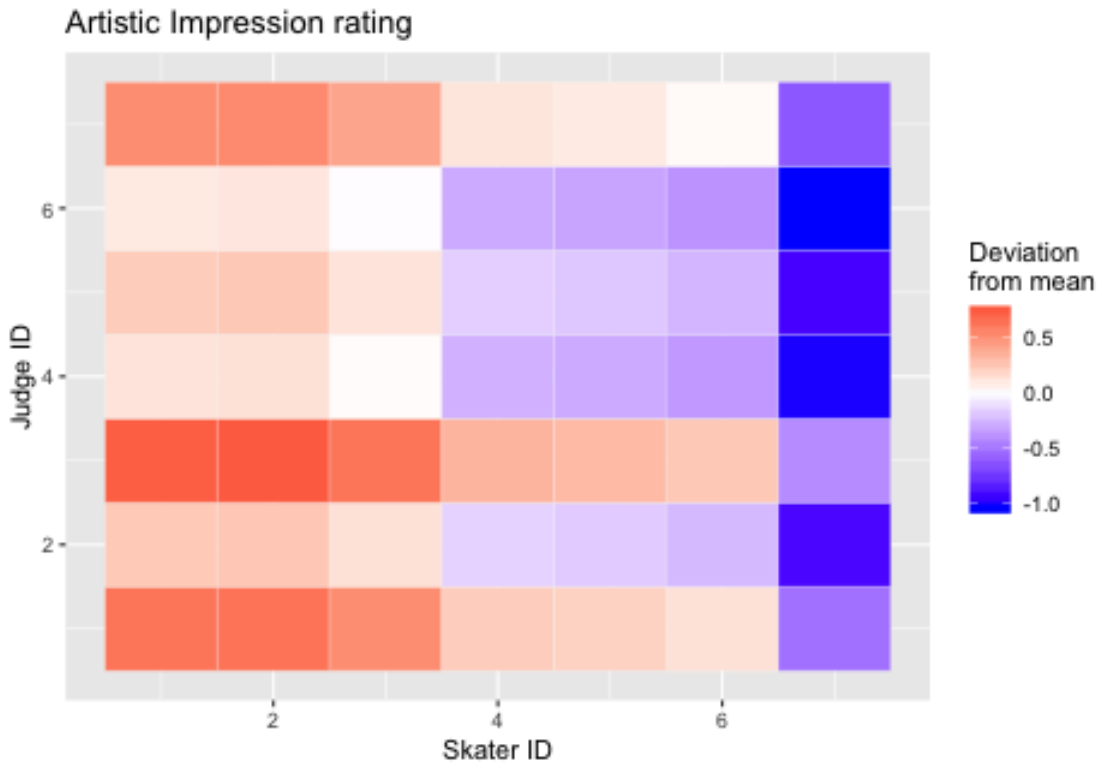
```
judge.id <- rep(c(1,2,3,4,5,6,7),7)

resid.1 <- predict(fit.8)-fixef(fit.8)
resid.2 <- Q2.table$Performance-predict(fit.8)
Q6.table <- Q2.table%>%dplyr::mutate(judge.id = judge.id,resid.1)


ggplot()+
  aes(x = Q2.table$pair,y = judge.id,fill = resid.1 )+
  geom_tile(color = "white")+
  xlab("Skater ID")+
  ylab("Judge ID")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, space = "Lab",
                       name="Deviation\nfrom mean")+
  labs(title = "Technical merit rating")
```



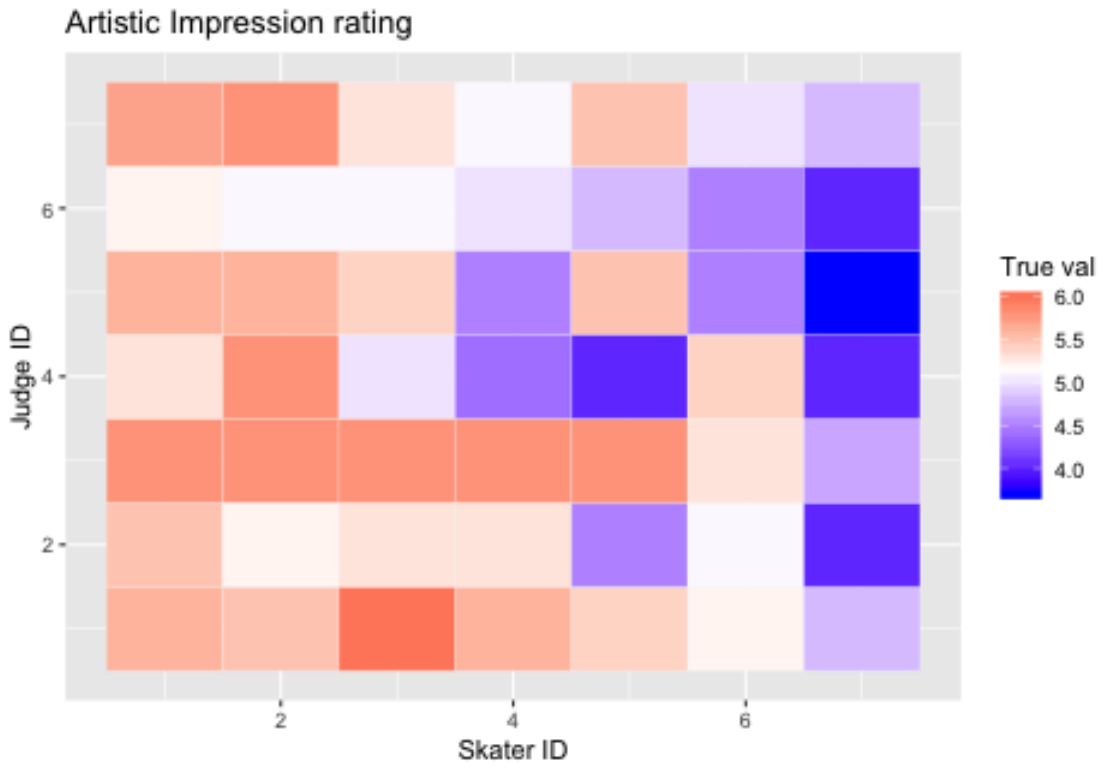

```
ggplot()+
  aes(x = Q2.table$pair,y = judge.id,fill = Q2.table$Performance )+
  geom_tile(color = "white")+
  xlab("Skater ID")+
  ylab("Judge ID")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = fixef(fit.8), space = "Lab",
                       name="True val")+
  labs(title = "Technical merit rating")
```




```
resid.3 <- predict(fit.9)-fixef(fit.9)
resid.4 <- Q2.table$Program-predict(fit.9)

ggplot()+
  aes(x = Q2.table$pair,y = judge.id,fill = resid.3 )+
  geom_tile(color = "white")+
  xlab("Skater ID")+
  ylab("Judge ID")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, space = "Lab",
                       name="Deviation\nfrom mean")+
  labs(title = "Artistic Impression rating")
```




```
ggplot()+
  aes(x = Q2.table$pair,y = judge.id,fill = Q2.table$Program )+
  geom_tile(color = "white")+
  xlab("Skater ID")+
  ylab("Judge ID")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = fixef(fit.9), space = "Lab",
                       name="True val")+
  labs(title = "Artistic Impression rating")
```




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.

$$$$

## Models for adjusting individual ratings:

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

1. It would be natural to rate the applications based on their combined scores; however, there is a worry that different raters use different standards, and we would like to correct for this. Set up a model for the ratings (with parameters for the applicants and the raters).
2. It is possible that some persons on the committee show more variation than others in their ratings. Expand your model to allow for this.