homework 07

Name

November 10, 2017

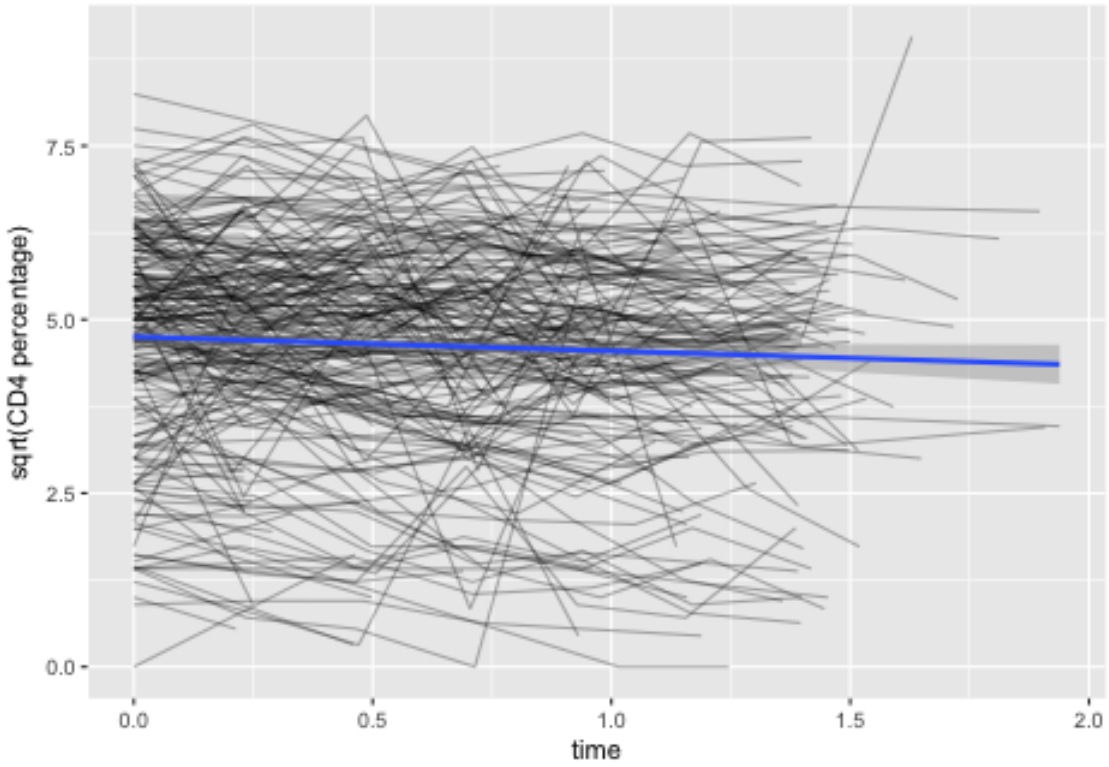
Data analysis

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.

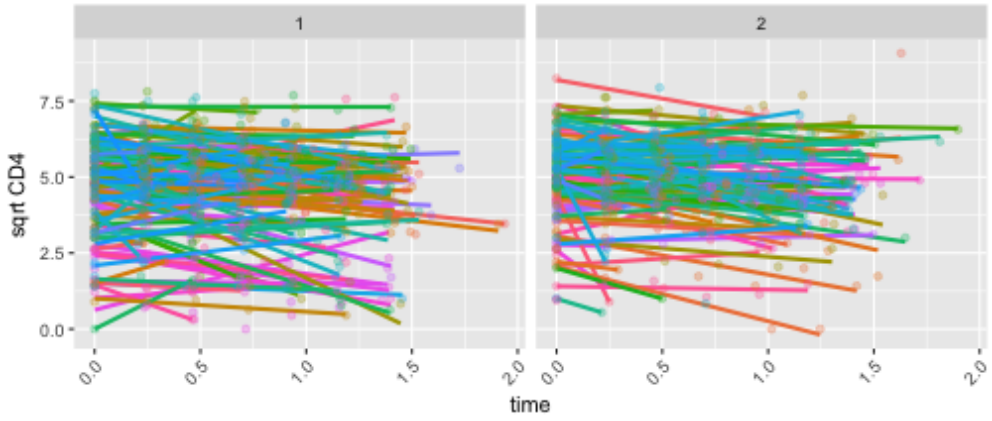
1. Graph the outcome (the CD4 percentage, on the square root scale) for each child as a function of time.

```
ggplot(hiv.data)+aes(x=time,y=y,group=newpid)+geom_line(alpha=0.3)+
  geom_smooth(method="lm",aes(group=1))+ylab("sqrt(CD4 percentage)")
```



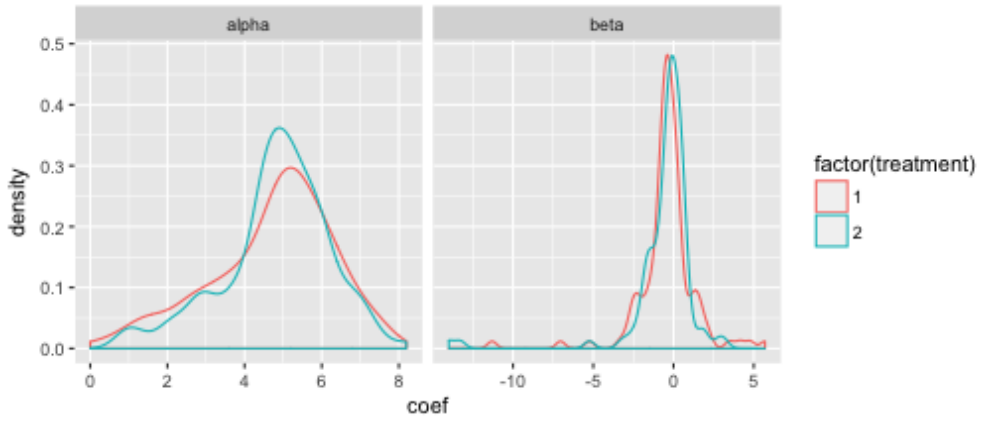

```
#facet_wrap(~treatment)+
```

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. Just using the geom_smooth function

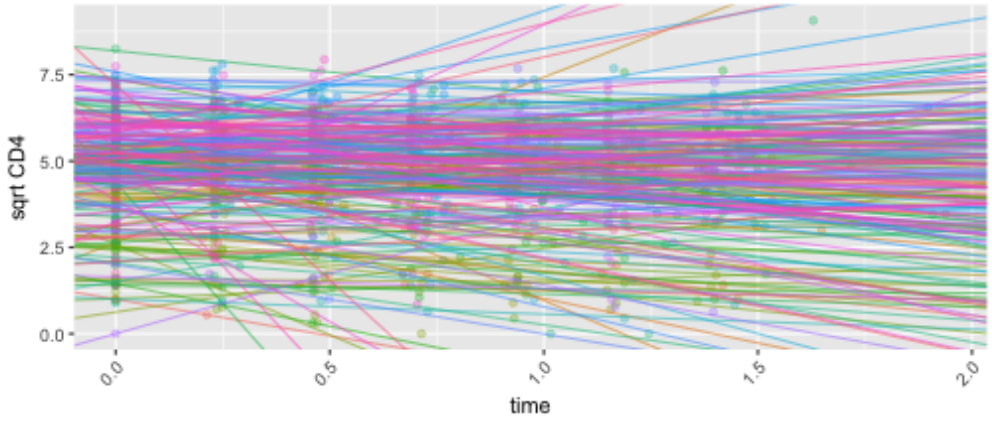


Estimating per patient.

```
## Warning: Removed 26 rows containing non-finite values (stat_density).
```




```
## Warning: Removed 26 rows containing missing values (geom_abline).
```



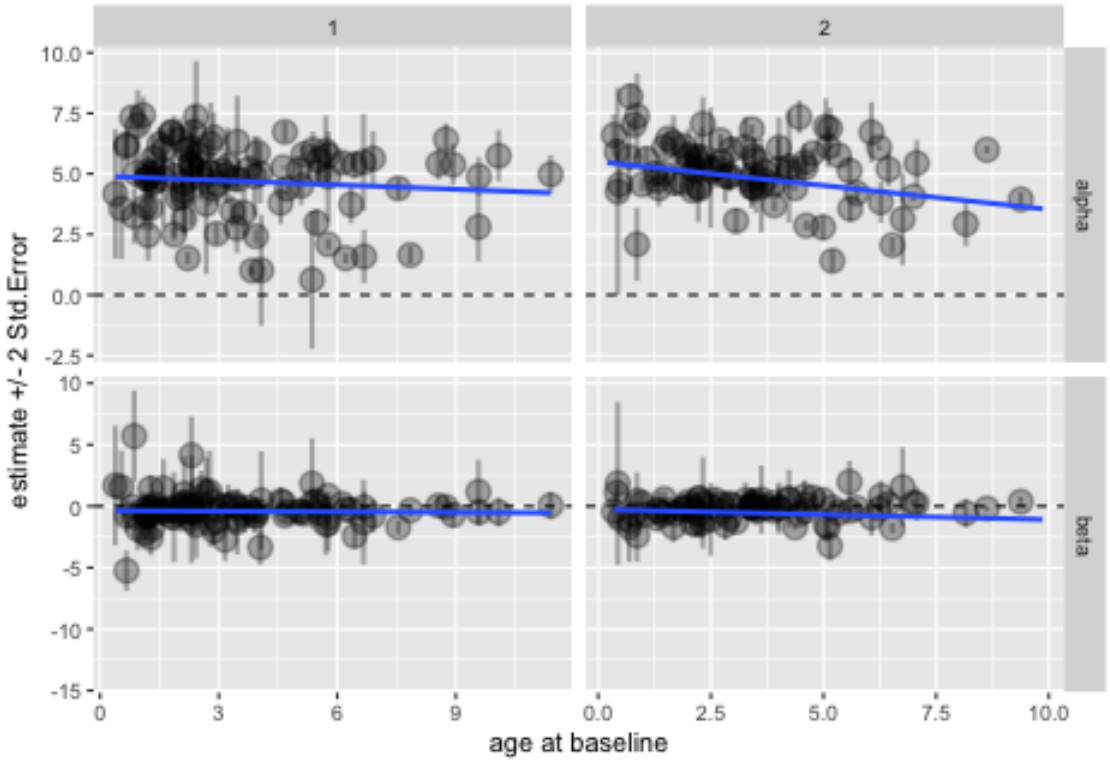
3. Set up a model for the children's slopes and intercepts as a function of the treatment and age at baseline. Estimate this model using the two-step procedure–first estimate the intercept and slope separately for each child, then fit the between-child models using the point estimates from the first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

```
ggplot(nopoolfit) +
    geom_pointrange(aes(ymax = coef + 2*se , ymin=coef - 2*se),lwd=1,alpha=0.3)+
    aes( x=age.baseline,y=coef)+
```

```
    geom_hline(yintercept=0,lty =2)+
  xlab("age at baseline")+
  ylab("estimate +/- 2 Std.Error")+
  facet_grid(Type~treatment,scale="free")+
  geom_smooth(method="lm",se=F)
```

```
## Warning: Removed 26 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 126 rows containing missing values (geom_pointrange).
```




```
lma<-lm(coef~age.baseline+treatment,nopoolfit[Type=="alpha"])
lmb<-lm(coef~age.baseline+treatment,nopoolfit[Type=="beta"])
arm::display(lma)
```

```
## lm(formula = coef ~ age.baseline + treatment, data = nopoolfit[Type ==
##     "alpha"])
##              coef.est coef.se
## (Intercept)   4.99     0.32
## age.baseline -0.12     0.04
## treatment     0.12     0.19
## ---
## n = 250, k = 3
## residual sd = 1.48, R-Squared = 0.04
```

```
arm::display(lmb)
```

```
## lm(formula = coef ~ age.baseline + treatment, data = nopoolfit[Type ==
##     "beta"])
##              coef.est coef.se
## (Intercept)  -0.13     0.46
```

```
## age.baseline -0.04     0.06
## treatment    -0.14     0.27
## ---
## n = 224, k = 3
## residual sd = 2.01, R-Squared = 0.00
```

There is no significant treatment difference but the intercept has a large difference with regards to age at baseline.

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.

For $i = 1, \ldots, n$ observations on $j = 1, \ldots, J$ children

$$y_i = \beta_0 + \beta_1 time_i + \alpha_{j[i]} + \epsilon_i$$

where $\alpha_j \sim N(0, \sigma_a^2)$ and $\epsilon_i \sim N(0, \sigma_y^2)$

```
cd4fit1<-lmer(y~time+(1|newpid),hiv.data)
arm::display(cd4fit1)
```

```
## 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
```

```
mycoef<-coef(cd4fit1)$newpid
mycoef$newpid=unique(hiv.data$newpid)
gp2<-ggplot(hiv.data)+geom_line(alpha=0.3)+
  aes(x=time,y=y,group=newpid,color=factor(newpid))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  theme(legend.position="none")+geom_abline( data=mycoef,aes(slope=time, intercept=`(Intercept)`,color=
```

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.

For $i = 1, \ldots, n$ observations on $j = 1, \ldots, J$ children

$$y_i = \beta_0 + \beta_1 time_i + \gamma_1 age_{j[i]} + \gamma_2 treat_{j[i]} + \alpha_{j[i]} + \epsilon_i$$

where $\alpha_j \sim N(0, \sigma_a^2)$ and $\epsilon_i \sim N(0, \sigma_y^2)$ or

$$y_i \sim N(\beta_0 + \beta_1 time_i + \alpha_{j[i]}, \sigma_y^2)$$

where

$$\alpha_j \sim N(\gamma_0 + \gamma_1 treatment_j + \gamma_2 age_j, \sigma_a^2)$$

.

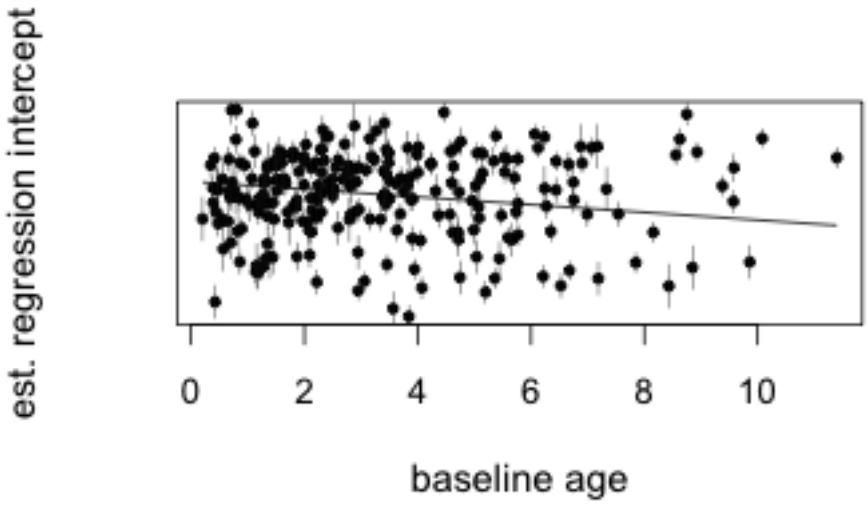
```
cd4fit2<-lmer(y~time+age.baseline+treatment+(1|newpid),hiv.data)
arm::display(cd4fit2)
```

```
## lmer(formula = y ~ time + age.baseline + treatment + (1 | newpid),
##     data = hiv.data)
##              coef.est coef.se
## (Intercept)   4.91     0.32
## time         -0.36     0.05
## age.baseline -0.12     0.04
## treatment     0.18     0.18
##
## 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
```

- With unit increase in time, sqrt CD4 percentage decreases by 0.36 on average.
- With every unit increase on age at baseline, we expect 0.12 decrease in sqrt CD4 percentage on average.
- The treatment group has a 0.18 increase in the outcome but the result is not statistically significantly different from zero.

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

- Shrinkage in the coefficient estimate

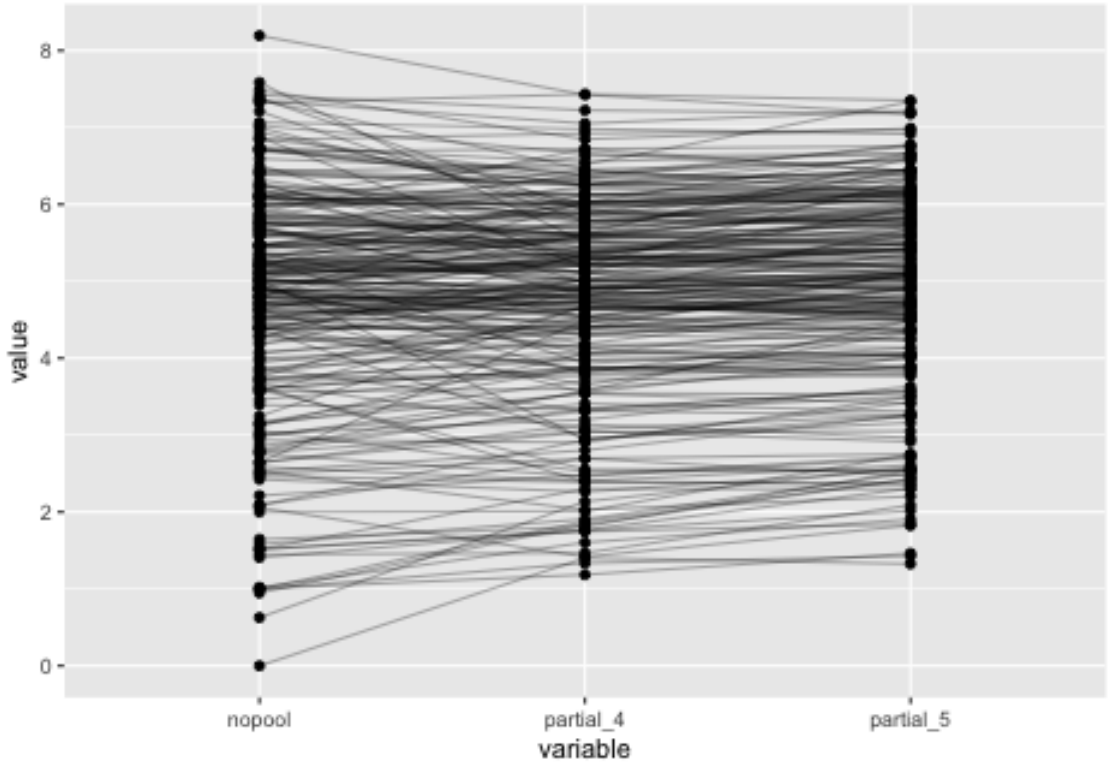



```
coef1<-cbind(coef(cd4fit1)$newpid,se.coef(cd4fit1)$newpid)
coef1$type="no"
coef1$newpid=unique(hiv.data$newpid)

coef2<-cbind(coef(cd4fit2)$newpid[,1:2],se.coef(cd4fit2)$newpid)
coef2$type="yes"
coef2$newpid=unique(hiv.data$newpid)
```

```
coef12<-rbind(coef1,coef2)
coef1122<-merge(coef1,coef2,by="newpid")
```

```
## Warning in merge.data.frame(coef1, coef2, by = "newpid"): column names
## '(Intercept).x', '(Intercept).y' are duplicated in the result
```

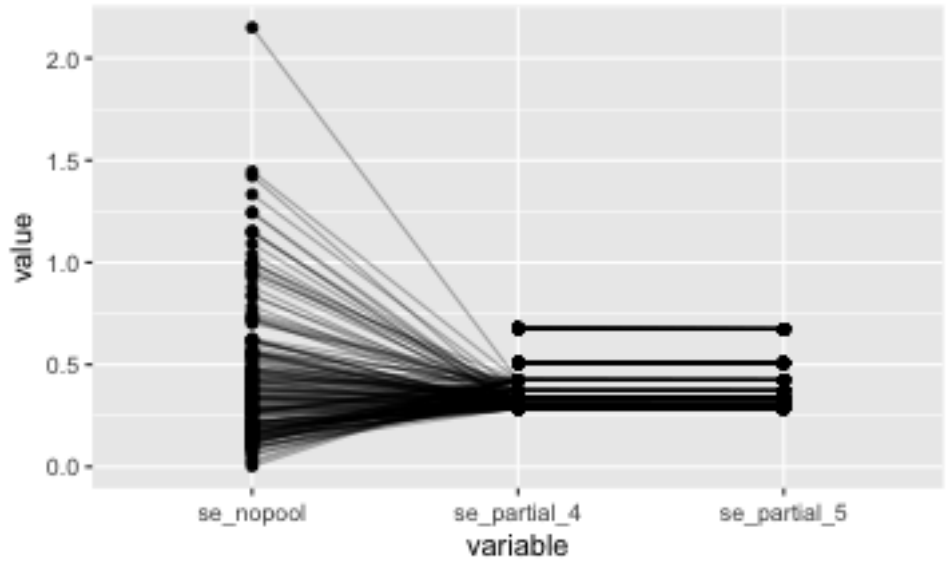
```
colnames(coef12)<-c("ranef","se","type","newpid")
colnames(coef1122)<-c("newpid","b.ranef","b.se","b.type","a.ranef","a.se","a.type")
coef001122<- cbind(nopoolfit[Type=="alpha",list(newpid,coef,se)],coef1[,c(1,3)],coef2[,c(1,3)])
colnames(coef001122)<-c("newpid","nopool","se_nopool","partial_4","se_partial_4","partial_5","se_partia

ggplot(melt(coef001122[,c(1,2,4,6),with=FALSE],id.vars = "newpid"))+
  geom_point()+aes(x=variable,y=value,group=newpid)+geom_line(alpha=0.3)
```



- Shrinkage in the standard error

```
## Warning: Removed 63 rows containing missing values (geom_point).
```

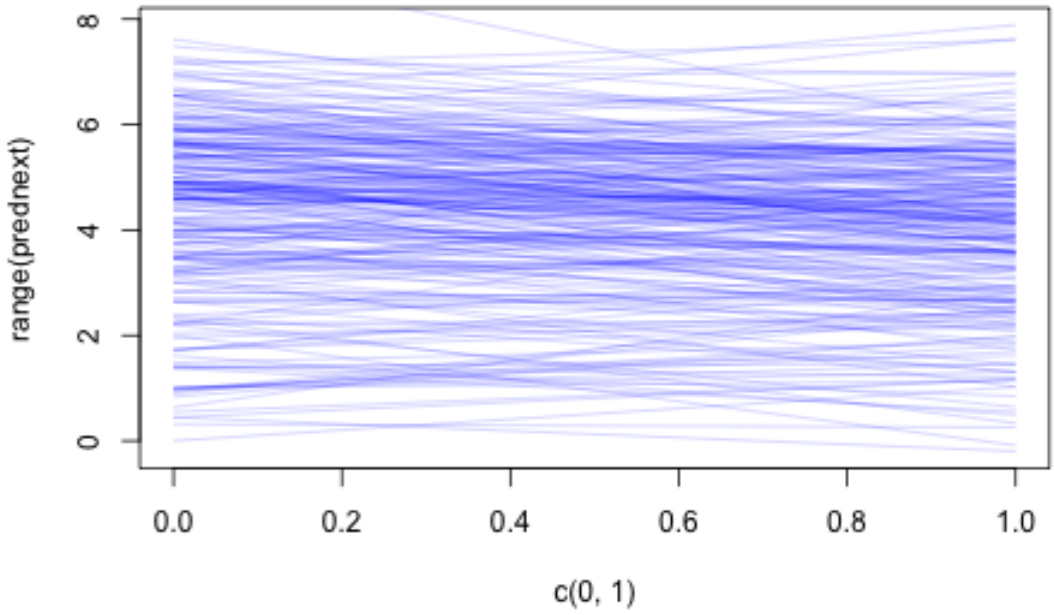
```
## Warning: Removed 63 rows containing missing values (geom_path).
```



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.

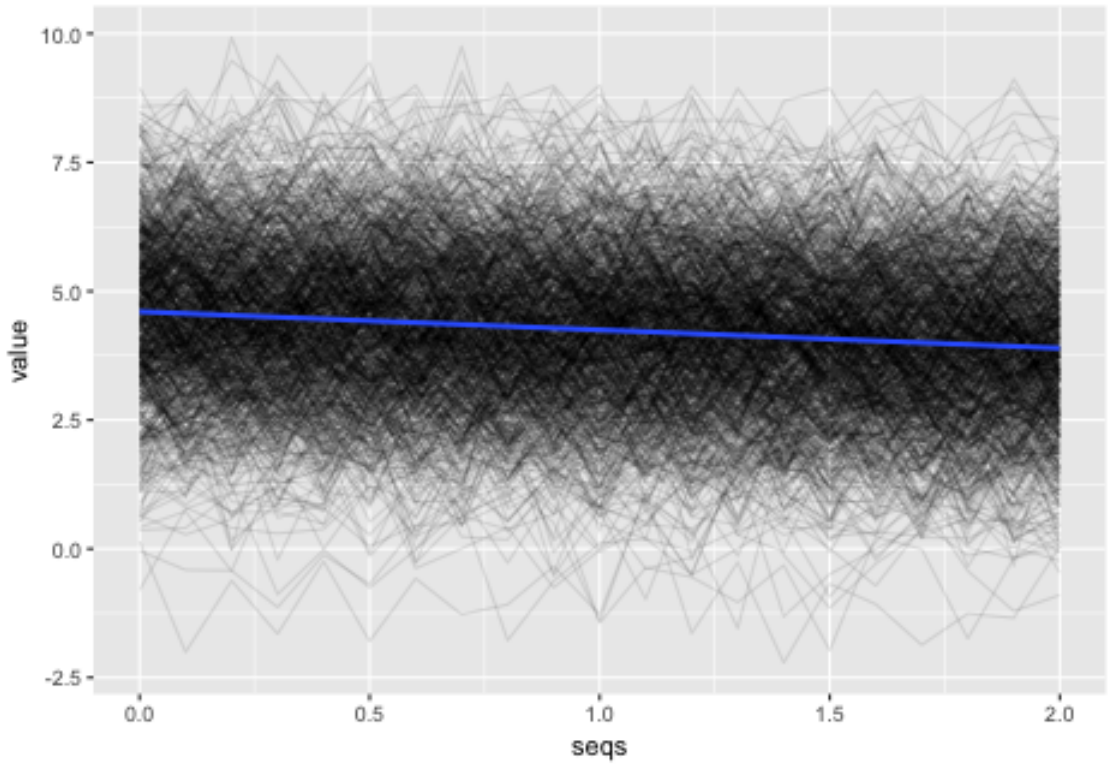
I'll add 1 to time to define the new time point.

```
coefhat<-as.matrix(coef(cd4fit2)$newpid)
preds<-group_by(hiv.data,newpid) %>%
  summarize(age.baseline=last(age.baseline),
            time=last(time),
            treatment=last(treatment))
prednext<-rnorm(250,unlist(coefhat[,"(Intercept)"]+
                      coefhat[,"time"]*(preds[,"time"]+1) +
                        coefhat[,"age.baseline"]*preds[,"age.baseline"] +
                        coefhat[,"treatment"]*preds[,"treatment"]),
                sigma.hat(cd4fit2)$sigma$data)
aa<-group_by(hiv.data,newpid) %>% summarize(last(y))
plot(c(0,1),range(prednext),type="n");
for(i in 1:length(prednext))lines(c(0,1),c(unlist(aa[,2])[i],prednext[i]),col=rgb(0,0,1,alpha=0.2))
```



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.

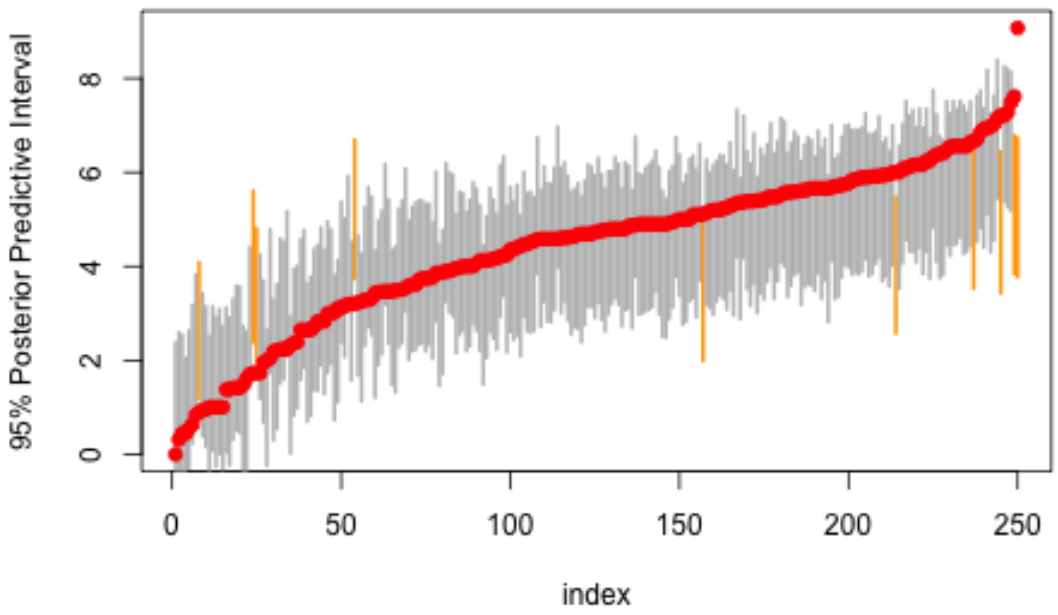
```
cf<-fixef(cd4fit2)
coefhat<-as.matrix(coef(cd4fit2)$newpid)
sigma.y.hat<-sigma.hat(cd4fit2)$sigma$data
sigma.a.hat<-sigma.hat(cd4fit2)$sigma$newpid
simpoints<- matrix(NA,1000,21)
seqs <- seq(0,2,by=0.1)
a.tilde <- rnorm(1000,cf["(Intercept)"]+cf["age.baseline"]*4+cf["treatment"],sigma.a.hat)
for ( i in 1:21){
y.tilde <- rnorm(1000,a.tilde+cf["time"]*seqs[i],sigma.y.hat)
simpoints[,i]<-y.tilde
}
ggplot(melt(data.frame(seqs,t(simpoints)),id.vars="seqs"))+
  geom_line(alpha=0.1)+aes(x=seqs, y=value, group=variable)+geom_smooth(method="lm",aes(group=1),se=FALS
```



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.

```
cf<-fixef(cd4fit2)
coefhat<-as.matrix(coef(cd4fit2)$newpid)
sigma.y.hat<-sigma.hat(cd4fit2)$sigma$data
sigma.a.hat<-sigma.hat(cd4fit2)$sigma$newpid
hiv.data.max <- hiv.data[,mt:=max(time),by=newpid][time==mt]

n <- nrow(hiv.data.max)
simpoints<- matrix(NA,n,1000)
for ( i in 1:1000){
y.tilde <- rnorm(n,coefhat[,1]+coefhat[,2]*hiv.data.max$mt+coefhat[,3]*hiv.data.max$age.baseline+coefha
simpoints[,i]<-y.tilde
}

maxy<-hiv.data.max$y
maxyord<-maxy[order(hiv.data.max$y)]
simpointsord<-simpoints[order(hiv.data.max$y),]
plot(1:n,y=maxyord,col="red",type="n",xlab="index",ylab="95% Posterior Predictive Interval")
pint_ord<- apply(simpointsord,1,quantile,c(0.025,0.975))
miss<-(maxyord<pint_ord[1,]|maxyord>pint_ord[2,])

for(i in 1:n){
lines(c(i,i),pint_ord[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
}
```

```
points(1:n,maxyord,col="red",pch=19)
```

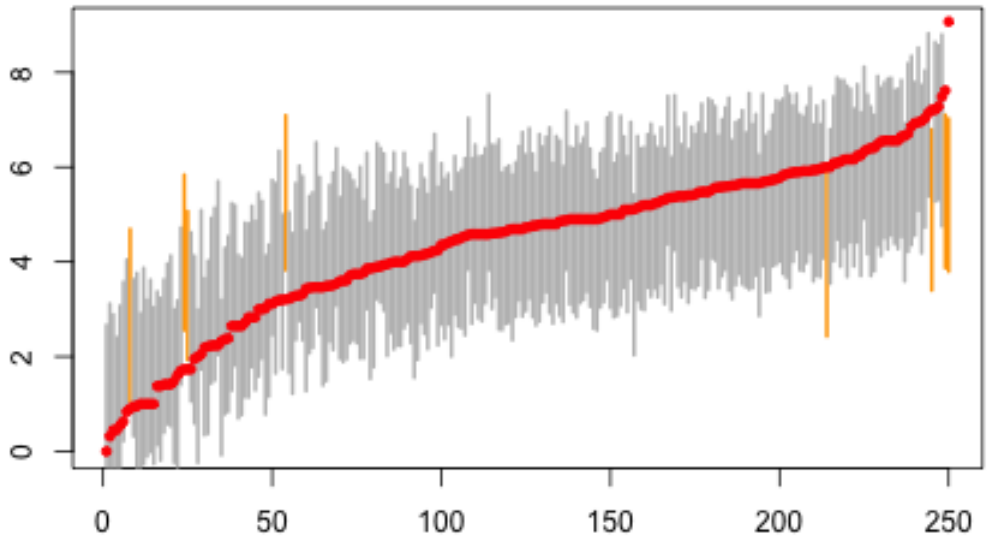



```
# for (i in 1:n) lines(cbind(hiv.data.max$y,hiv.data.max$y)[i,],y=cbind(hiv.data.max$y-2*apply(simpoint
```

The 95% interval covers the observed data fairly well. The intervals that missed the observed data are colored in orange.

- Using Stan

```
#cd4fit2_stan<-stan_lmer(y~time+age.baseline+treatment+(1|newpid),hiv.data)
#saveRDS(cd4fit2_stan,"cd4fit2_stan.rds")
cd4fit2_stan<- readRDS("cd4fit2_stan.rds")

pint <- apply(posterior_predict(cd4fit2_stan)[,hiv.data$time==hiv.data$mt],2,quantile,c(0.025,0.975))
ord <- order(hiv.data.max$y)
pint_ord <- pint[,ord]

miss<-(hiv.data.max$y[ord]<pint_ord[1,]|hiv.data.max$y[ord]>pint_ord[2,])
plot(c(1,250), c(0,9),type="n",ylab="",xlab="")
for(i in 1:250) lines(c(i,i),pint_ord[,i],lwd=2,
                      col=ifelse(miss[i],"orange","gray"))
points(1:250,hiv.data.max$y[ord],col="red",pch=20)
```



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

```
cd4fit3<-lmer(y~time+treatment+age.baseline+(1+time|newpid),hiv.data)
arm::display(cd4fit3)
```

```
## 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
```

```
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

xx<-unique(hiv.data[,list(1,age.baseline,treatment=1*(treatment==2)),by=newpid])


mycoef<-data.frame(intercept =  rowSums( as.matrix(xx[,-1])*(coef(cd4fit3)$newpid[,c(1,3,4)])),
```

```
                    time=unlist(coef(cd4fit3)$newpid[,c(2)]))
mycoef$newpid=unique(hiv.data$newpid)
names(mycoef)[2]<-"time"
gp2<-ggplot(hiv.data)+geom_line(alpha=0.3)+
  aes(x=time,y=y,group=newpid,color=factor(newpid))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  theme(legend.position="none")+
  geom_abline( data=mycoef, aes(slope=time,
                          intercept=intercept,color=factor(newpid),alpha=0.1))+
  geom_smooth(method="lm",se=FALSE,aes(group=1))+ ylab("sqrt CD4")
#cd4fit3<-stan_lmer(y~time+age.baseline+treatment+(1+time|newpid),hiv.data)
#arm::display(cd4fit3)
```

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).

```
hiv.data$time_01<-round(hiv.data$time,2)
cd4fit4<-lmer(y~factor(time_01)+age.baseline+treatment+(1|newpid),hiv.data)
#arm::display(cd4fit4)
```

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

```
display(cd4fit3)
```

```
## 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(cd4fit4)
```

```
cf<-fixef(cd4fit3)
coefhat<-as.matrix(coef(cd4fit3)$newpid)
sigma.y.hat<-sigma.hat(cd4fit3)$sigma$data
sigma.a.hat<-sigma.hat(cd4fit3)$sigma$newpid
hiv.data.max <- hiv.data[,mt:=max(time),by=newpid][time==mt]

n <- nrow(hiv.data.max)
simpoints<- matrix(NA,n,1000)
for ( i in 1:1000){
y.tilde <- rnorm(n,coefhat[,1] + coefhat[,2]*hiv.data.max$mt +
                    coefhat[,3]*hiv.data.max$age.baseline +
                    coefhat[,4]*(hiv.data.max$treatment-1), sigma.y.hat)
```

```
simpoints[,i]<-y.tilde
}

maxy<-hiv.data.max$y
maxyord<-maxy[order(hiv.data.max$y)]
simpointsord<-simpoints[order(hiv.data.max$y),]
plot(1:n,y=maxyord,col="red",type="n",xlab="index",ylab="95% Posterior Predictive Interval")
miss<-(maxyord<pint_ord[1,]|maxyord>pint_ord[2,])
pint_ord<- apply(simpointsord,1,quantile,c(0.025,0.975))
for(i in 1:n){
lines(c(i,i),pint_ord[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
}
points(1:n,maxyord,col="red",pch=19)
```

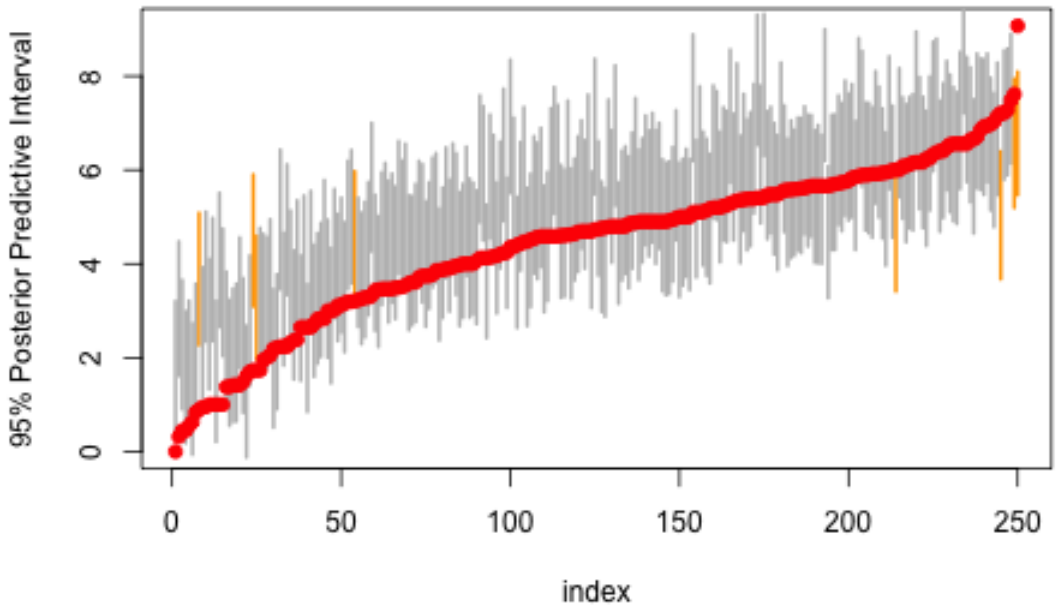
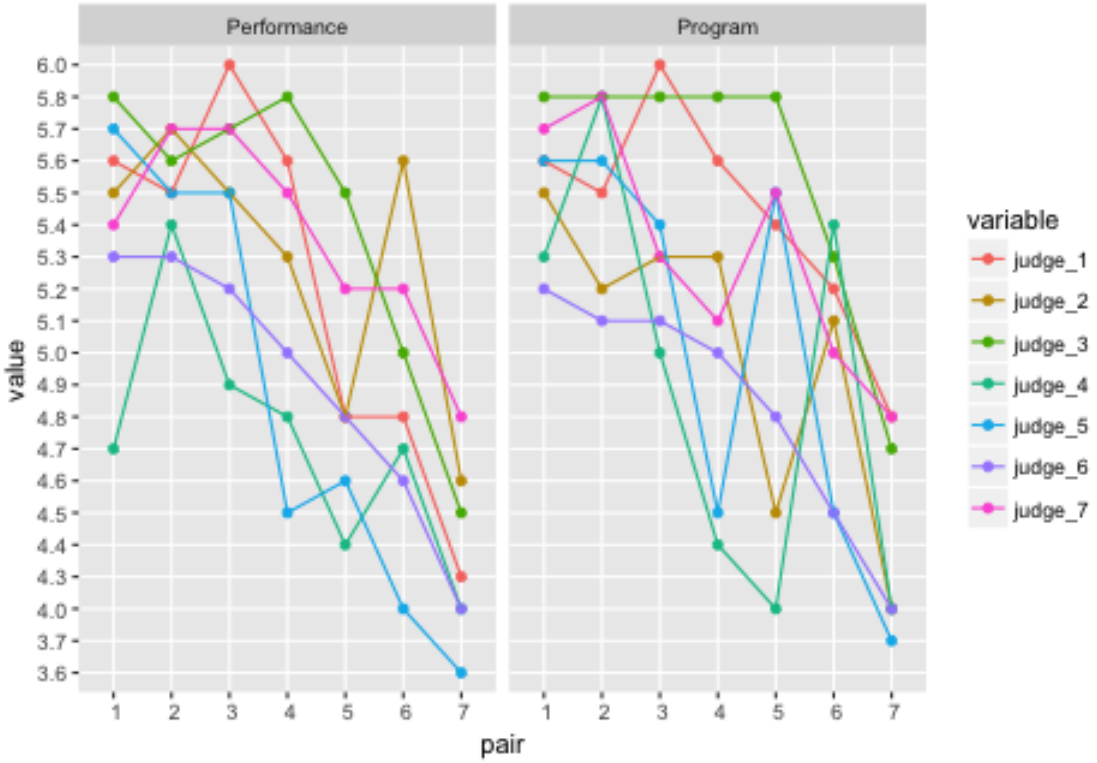


Figure skate in the 1932 Winter Olympics

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).

```
ggplot(melt(olympics1932,id.vars=c("pair","criterion")))+geom_point()+aes(x=pair,y=value,group=variable
```




```
olympic_array <- array(NA,c(7,7,2))
olympic_array[,,1]<-as.double(unlist(olympics1932[seq(1,14,by=2),3:9]))
olympic_array[,,2]<-as.double(unlist(olympics1932[seq(2,14,by=2),3:9]))
olympic_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 49×4 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.

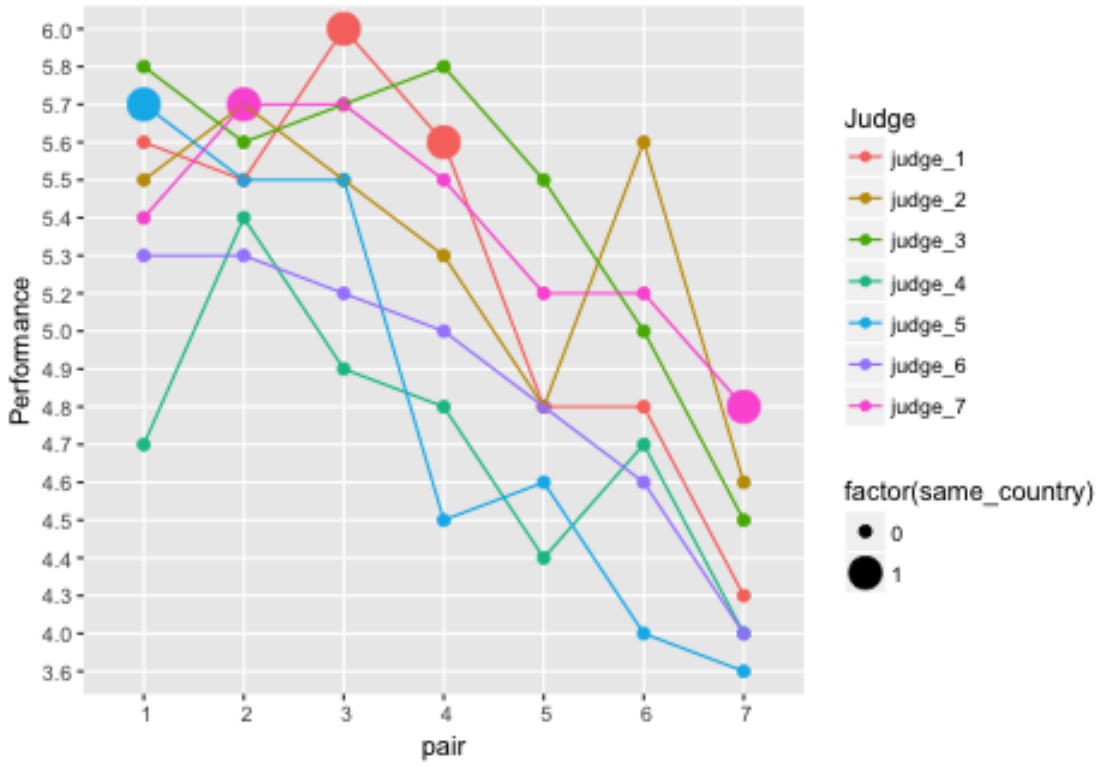
```
olong <- dcast(melt(olympics1932,id.vars=c("pair","criterion")),pair+variable~criterion)
setnames(olong,c( "variable"),c("Judge"))
olympics_long <- olong
head(olympics_long)
```

```
##   pair   Judge Performance Program
## 1    1 judge_1         5.6     5.6
## 2    1 judge_2         5.5     5.5
## 3    1 judge_3         5.8     5.8
## 4    1 judge_4         4.7     5.3
## 5    1 judge_5         5.7     5.6
## 6    1 judge_6         5.3     5.2
```

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.

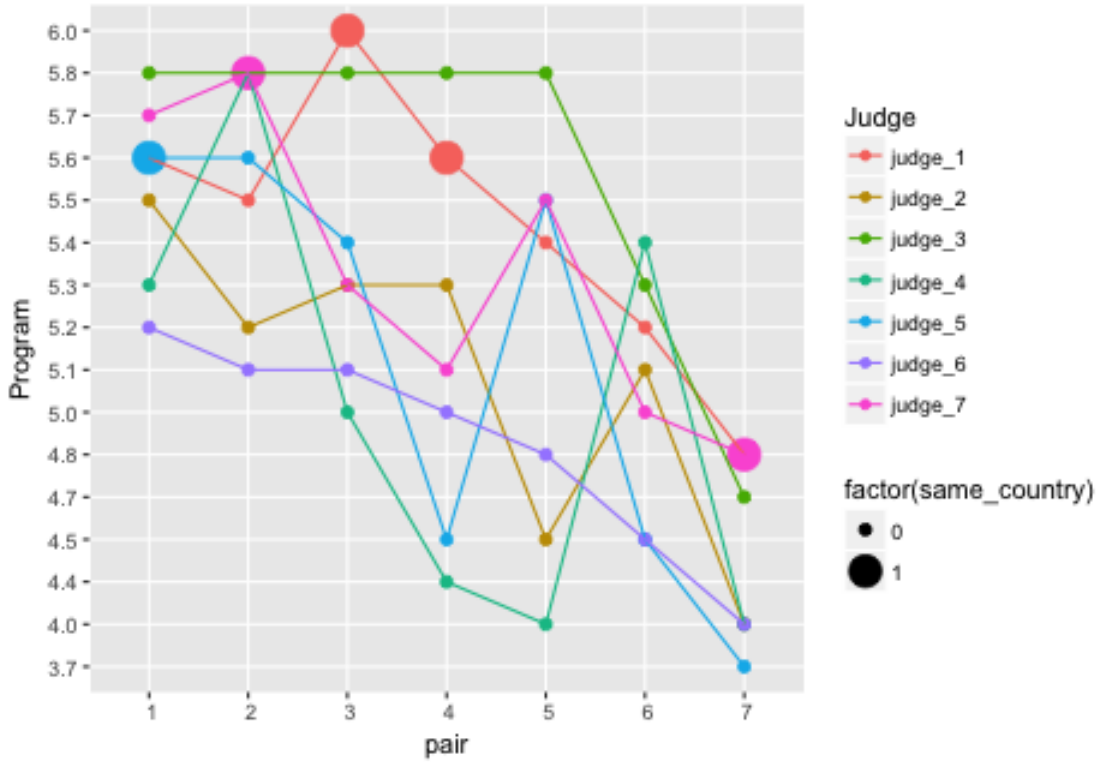
```
file_name<-"http://www.stat.columbia.edu/~gelman/arm/examples/olympics/olympics1932.txt"
pair_country<-str_trim(read.csv(file_name,skip=3,
                                nrows = 7,
                                header=FALSE,stringsAsFactors=FALSE)$V3)
judge_country<-str_trim(read.csv(file_name,skip=12,
                                 nrows = 7,
                                 header=FALSE,stringsAsFactors=FALSE)$V2)
names(pair_country)<-1:7
names(judge_country)<-paste("judge",1:7,sep="_")
olympics_long$same_country<-1*(pair_country[as.integer(olympics_long$pair)]==
                                 judge_country[as.character(olympics_long$Judge)])

ggplot(olympics_long)+geom_point(aes(size=factor(same_country)))+aes(x=pair,y=Performance,group=Judge,c
```

```
## Warning: Using size for a discrete variable is not advised.
```




```
ggplot(olympics_long)+geom_point(aes(size=factor(same_country)))+aes(x=pair,y=Program,group=Judge,color=
```

```
## Warning: Using size for a discrete variable is not advised.
```



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

$$
\begin{aligned}
y_i &\sim N(\mu + \gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2), \text{ for } i = 1, \ldots, n && (1)\\
\gamma_j &\sim N(0, \sigma_\gamma^2) j = 1, \ldots, 7 && (2)\\
\delta_k &\sim N(0, \sigma_\delta^2) k = 1, \ldots, 7 && (3)
\end{aligned}
$$

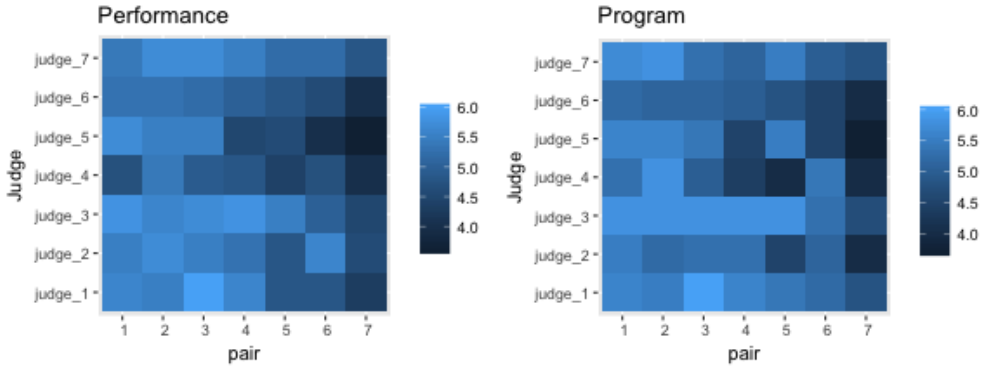
```
olympics_long$Performance<-as.double(olympics_long$Performance)
olympics_long$Program<-as.double(olympics_long$Program)
fit_program<-lmer(Program~1+(1|pair) +  (1|Judge),olympics_long)
```

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

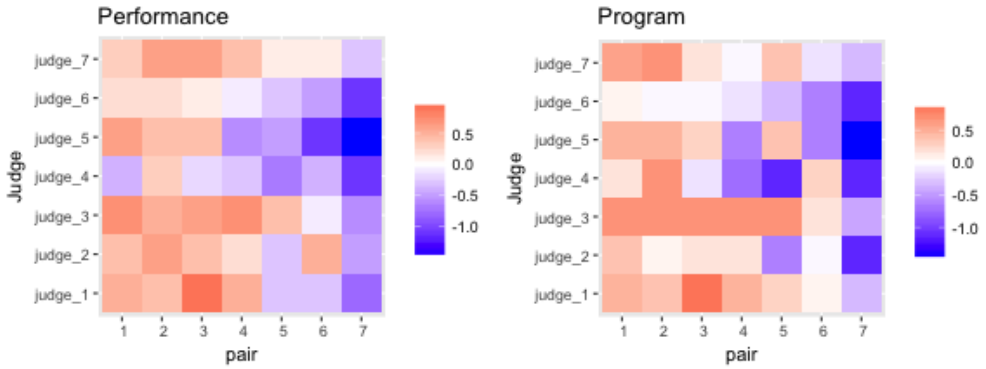
```
fit_performance<-lmer(Performance~1+(1|pair) +  (1|Judge),olympics_long)
```

6. Display your results for both outcomes graphically.

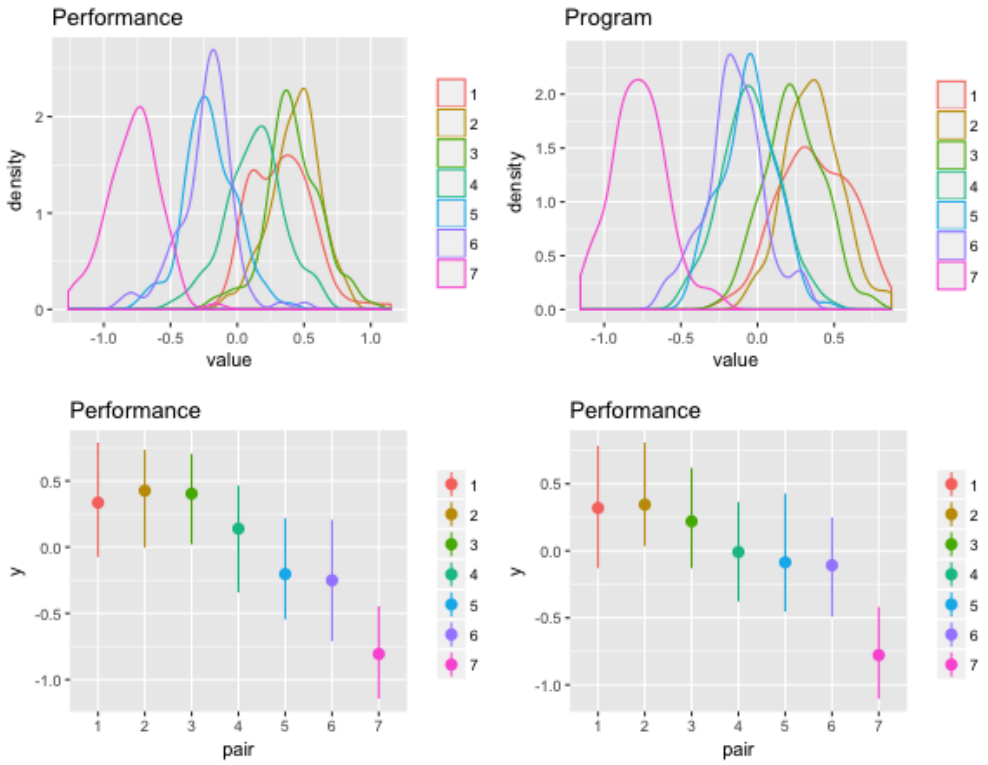
Comparing the scores



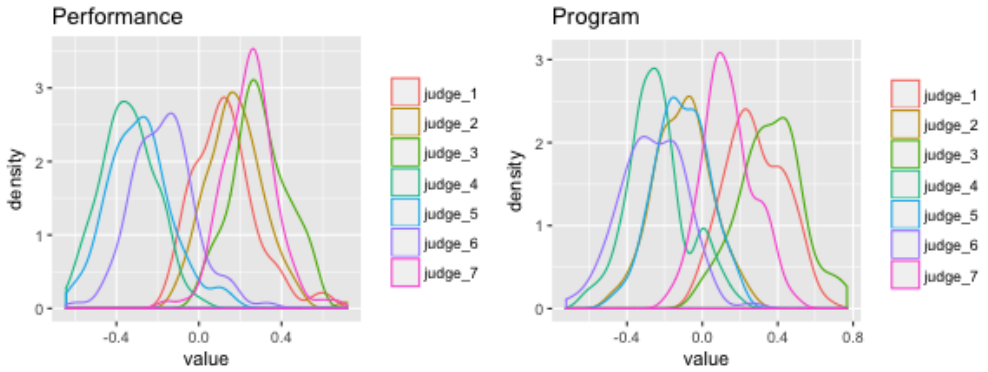
After subtracting the means.



Posterior distribution of the scores for the pairs.



Posterior distribution of the judge scores.



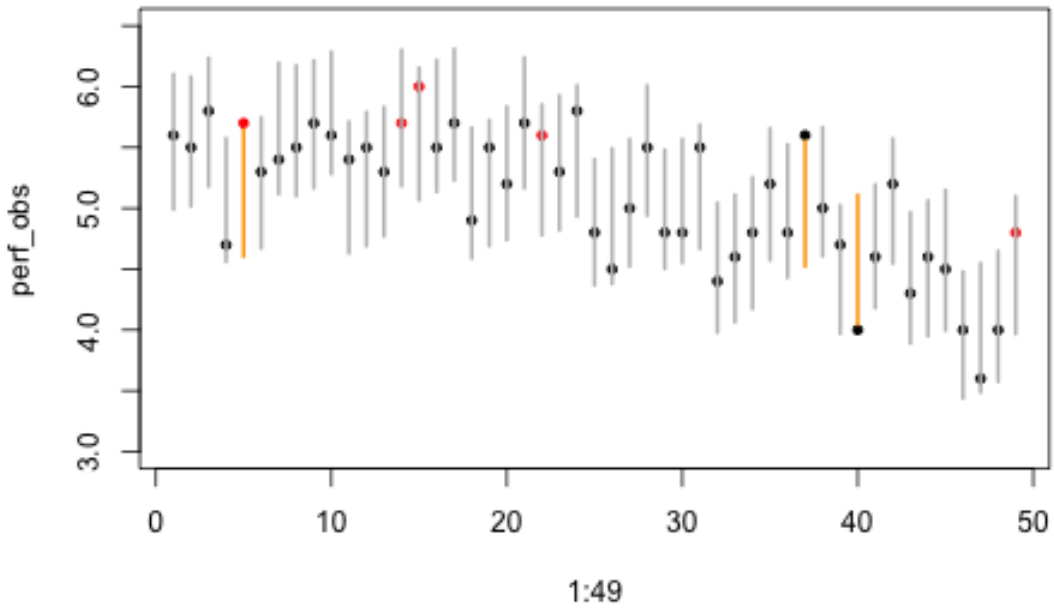
Using Stan

```
stan_fit_program<-stan_lmer(Program~1+(1|pair) +  (1|Judge),olympics_long)
saveRDS(stan_fit_program,"stan_fit_program.rds")
stan_fit_program<-readRDS("stan_fit_program.rds")

stan_fit_performance<-stan_lmer(Performance~1+(1|pair) +
                                  (1|Judge),olympics_long)
saveRDS(stan_fit_performance,"stan_fit_performance.rds")
stan_fit_performance<-readRDS("stan_fit_performance.rds")

 ggplot(melt(as.array(stan_fit_program)[,1,2:8]))+
   geom_density()+aes(x=value,color=parameters)
ggplot(melt(as.array(stan_fit_performance)[,1,2:8]))+
  geom_density()+aes(x=value,color=parameters)
grid.arrange(
```

```
 ggplot(melt(as.array(stan_fit_program)[,1,9:15]))+
   geom_density()+aes(x=value,color=parameters),
ggplot(melt(as.array(stan_fit_performance)[,1,9:15]))+
  geom_density()+aes(x=value,color=parameters),ncol=1)
```

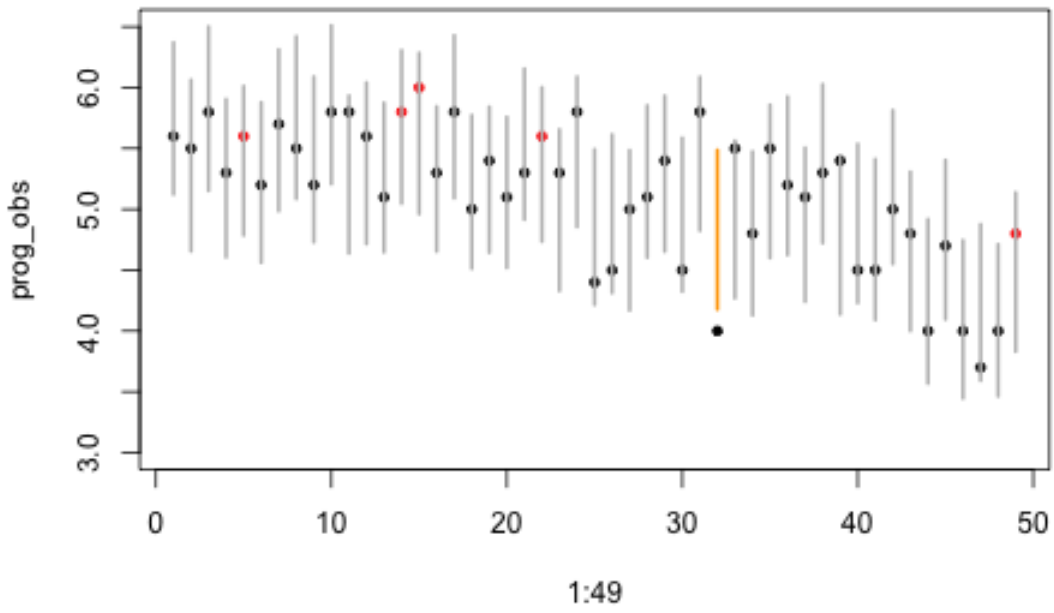
7. (extra) Use posterior predictive checks to investigate model fit in (4) and (5).

```
perf_obs <- olympics_long$Performance
judge_id <- as.integer(olympics_long$Judge)
s_y<-sigma.hat(fit_performance)$sigma$data
s_p<-sigma.hat(fit_performance)$sigma$pair
mu_p <- ranef(fit_performance)$pair
mu_j <- ranef(fit_performance)$Judge
mu_sum<-fixef(fit_performance)+mu_p[olympics_long$pair,]+mu_j[judge_id,]
ytilde<- matrix(NA,1000,49)
for(i in 1:1000){
  ytilde[i,]<- rnorm(49,mean=mu_sum,sd=s_y)
}

plot(1:49,perf_obs,ylim=c(3,6.5),pch=20,col=olympics_long$same_country+1)
pint<-apply(ytilde,2,quantile,c(0.025,0.975))
miss<-(perf_obs<pint[1,]|perf_obs>pint[2,])
for(i in 1:49){
  lines(c(i,i),pint[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
}
```




```
prog_obs <- olympics_long$Program
judge_id <- as.integer(olympics_long$Judge)
s_y      <- sigma.hat(fit_program)$sigma$data
```

```r
s_p      <- sigma.hat(fit_program)$sigma$pair
mu_p     <- ranef(fit_program)$pair
mu_j     <- ranef(fit_program)$Judge
mu_sum   <- fixef(fit_program)+mu_p[olympics_long$pair,]+mu_j[judge_id,]
ytilde   <- matrix(NA,1000,49)
for(i in 1:1000){
  ytilde[i,] <- rnorm(49,mean=mu_sum,sd=s_y)
}

plot(1:49,prog_obs,ylim=c(3,6.5),pch=20,col=olympics_long$same_country+1)
pint<-apply(ytilde,2,quantile,c(0.025,0.975))
miss<-(prog_obs<pint[1,]|prog_obs>pint[2,])
for(i in 1:49){
  lines(c(i,i),pint[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
}
```

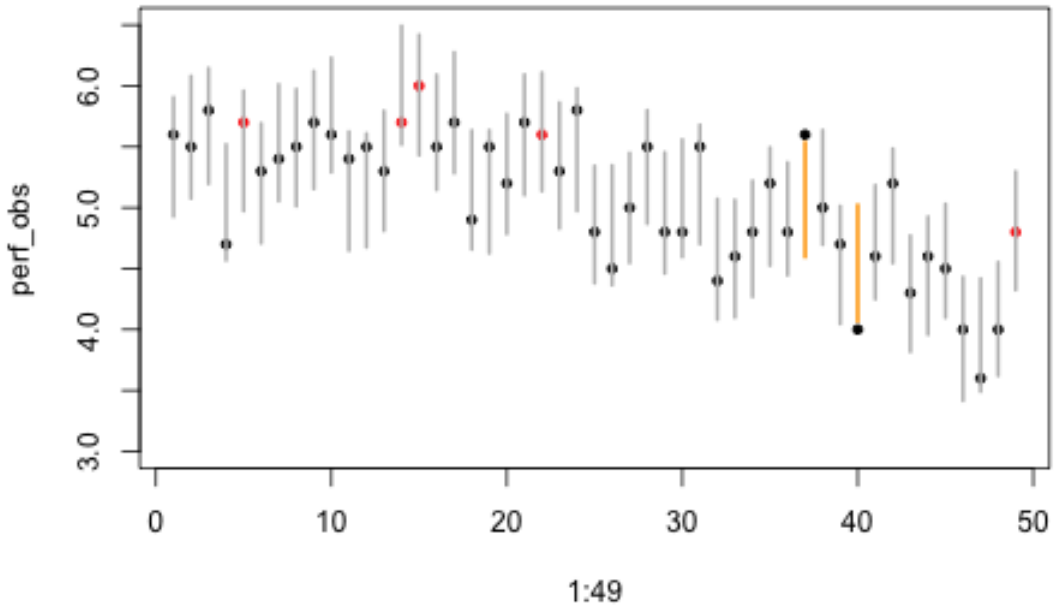



```r
#
# gperobs<-ggplot(olympics_long)+geom_density()+aes(x=Performance,color=factor(pair))
# gpersim<-ggplot(melt(ytilde))+geom_density()+aes(x=value,color=factor(Var2))
# grid.arrange(gperobs,gpersim,ncol=2)
#
# perf_obs <- olympics_long$Performance
# s_y<-sigma.hat(fit_program)$sigma$data
# s_p<-sigma.hat(fit_program)$sigma$pair
# mu_p <- coef(fit_program)$pair
# ytilde_pro<- matrix(NA,1000,7)
# for(i in 1:1000){
#   ytilde_pro[i,]<- rnorm(7,mean=mu_p[1:7,],sd=s_y)
# }
```

```
# gproobs<-ggplot(olympics_long)+geom_density()+aes(x=Program,color=factor(pair))
# gprosim<-ggplot(melt(ytilde_pro))+geom_density()+aes(x=value,color=factor(Var2))
# grid.arrange(gproobs,gprosim,ncol=2)
#
# ppdprog<-data.frame(olympics_long,post=t(posterior_predict(stan_fit_program)))
# pint <- apply(posterior_predict(stan_fit_program),2,quantile,c(0.025,0.975))
# plot(c(1,49), c(3,8),type="n",ylab="Program",xlab="pair:judge")
# for(i in 1:49) lines(c(i,i),pint[,i],lwd=2,col="gray")
# points(1:49,olympics_long$Program,col="red",pch=20)
#
# olympics_long[t(pint)[,1]>olympics_long$Program|t(pint)[,2]<olympics_long$Program,]
# rint <- apply(posterior_predict(stan_fit_performance),2,quantile,c(0.025,0.975))
# plot(c(1,49), c(3,8),type="n",ylab="Performance",xlab="pair:judge")
# for(i in 1:49) lines(c(i,i),rint[,i],lwd=2,col="gray")
# points(1:49,olympics_long$Performance,col="red",pch=20)
#
# plot(ppdprog$Program,ppdprog$post.1,type="n"); abline(0,1);
# text(ppdprog$Program,ppdprog$post.1,paste(ppdprog$pair,ppdprog$Judge))
```

Explicity modeling the same country effect

```
fit_program_2    <- lmer(Program~1+same_country+(1|pair) +  (1|Judge),olympics_long)

fit_performance_2<- lmer(Performance~1+same_country+(1|pair) +  (1|Judge),olympics_long)

perf_obs <- olympics_long$Performance
judge_id <- as.integer(olympics_long$Judge)
s_y<-sigma.hat(fit_performance_2)$sigma$data
s_p<-sigma.hat(fit_performance_2)$sigma$pair
mu_p <- ranef(fit_performance_2)$pair
mu_j <- ranef(fit_performance_2)$Judge
mu_sum<-fixef(fit_performance_2)[1]+
        fixef(fit_performance_2)[2]*olympics_long$same_country+
        mu_p[olympics_long$pair,]+mu_j[judge_id,]
ytilde<- matrix(NA,1000,49)
for(i in 1:1000){
  ytilde[i,]<- rnorm(49,mean=mu_sum,sd=s_y)
}

plot(1:49,perf_obs,ylim=c(3,6.5),pch=20,col=olympics_long$same_country+1)
pint<-apply(ytilde,2,quantile,c(0.025,0.975))
miss<-(perf_obs<pint[1,]|perf_obs>pint[2,])
for(i in 1:49){
  lines(c(i,i),pint[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
}
```

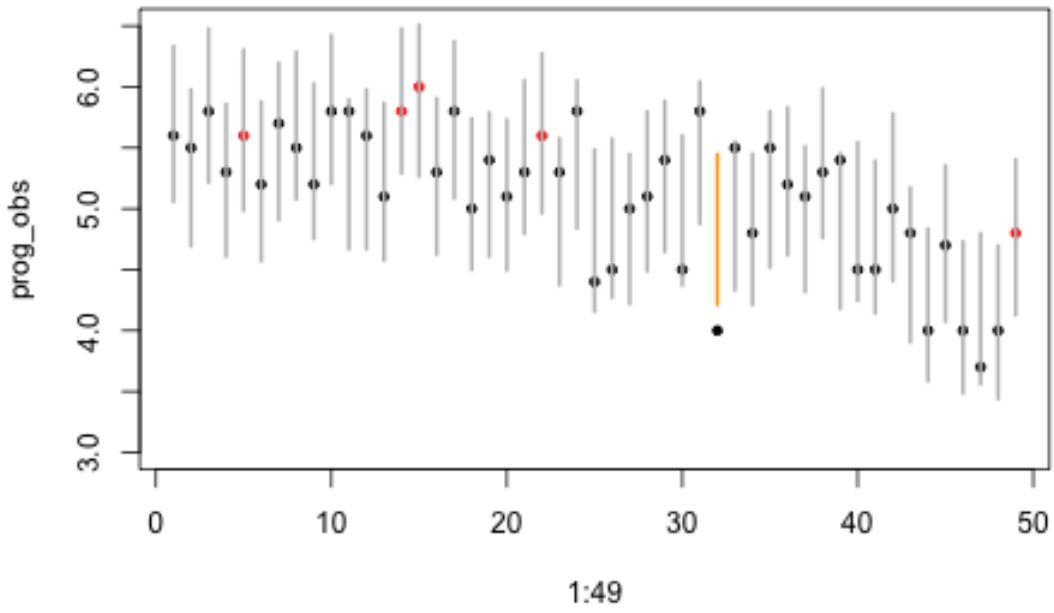



```
prog_obs <- olympics_long$Program
judge_id <- as.integer(olympics_long$Judge)
mu_p[olympics_long$pair,]
```

```
##  [1]  0.3067703  0.3067703  0.3067703  0.3067703  0.3067703  0.3067703
##  [7]  0.3067703  0.4025758  0.4025758  0.4025758  0.4025758  0.4025758
## [13]  0.4025758  0.4025758  0.3752028  0.3752028  0.3752028  0.3752028
## [19]  0.3752028  0.3752028  0.3752028  0.1014727  0.1014727  0.1014727
## [25]  0.1014727  0.1014727  0.1014727  0.1014727 -0.1715626 -0.1715626
## [31] -0.1715626 -0.1715626 -0.1715626 -0.1715626 -0.1715626 -0.1989356
## [37] -0.1989356 -0.1989356 -0.1989356 -0.1989356 -0.1989356 -0.1989356
## [43] -0.8155233 -0.8155233 -0.8155233 -0.8155233 -0.8155233 -0.8155233
## [49] -0.8155233
```

```
s_y<-sigma.hat(fit_program_2)$sigma$data
s_p<-sigma.hat(fit_program_2)$sigma$pair
mu_p <- ranef(fit_program_2)$pair
mu_j <- ranef(fit_program_2)$Judge
mu_sum<-fixef(fit_program_2)[1]+fixef(fit_program_2)[2]*olympics_long$same_country+mu_p[olympics_long$pa
ytilde<- matrix(NA,1000,49)
for(i in 1:1000){
  ytilde[i,]<- rnorm(49,mean=mu_sum,sd=s_y)
}

plot(1:49,prog_obs,ylim=c(3,6.5),pch=20,col=olympics_long$same_country+1)
pint<-apply(ytilde,2,quantile,c(0.025,0.975))
miss<-(prog_obs<pint[1,]|prog_obs>pint[2,])
for(i in 1:49){
  lines(c(i,i),pint[,i],lwd=2,col=ifelse(miss[i],"orange","gray"))
```

```
}
```



Different ways to write the model:

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

- Allowing regression coefficients to vary across groups

$$y_i = \alpha_{j[i]} + \mathbf{X}_i\boldsymbol{\beta} + e_i \quad (4)$$
$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2) \quad (5)$$

- Combining separate local regressions -

$$y_i \sim N(\alpha_j + \boldsymbol{\beta}x_i, \sigma_y^2) \quad (6)$$
$$\alpha_j \sim N(\gamma_0 + \gamma_1 u_j, \sigma_\alpha^2) \quad (7)$$

- Modeling the coefficients of a large regression model

$$y_i \sim N(\mathbf{X}_i\boldsymbol{\beta}, \sigma_y^2) \quad (8)$$
$$\beta_j \sim N(0, \sigma_\alpha^2) \quad (9)$$

- Regression with multiple error terms

$$y_i \sim N(\mathbf{X}\boldsymbol{\beta} + \eta_{j[i]}, \sigma_y^2) \quad (10)$$
$$\eta_j \sim N(0, \sigma_\alpha^2) \quad (11)$$

- Large regression with correlated errors

$$y_i = \mathbf{X}_i\boldsymbol{\beta} + e_i \quad (12)$$
$$\epsilon_i \sim N(0, \Sigma) \quad (13)$$

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).

Observation is evaluation $i = 1, \ldots, 300$ there are 100 applicants $j = 1, \ldots 100$ and 10 committee members $k = 1, \ldots 10$.

$$y_i \sim N(\mu + \gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2), \text{ for } i = 1, \ldots, 300, j = 1, 2, 3 \quad (14)$$
$$\gamma_j \sim N(\mu_j, \sigma_\gamma^2), j = 1, \ldots, 100 \quad (15)$$
$$\delta_k \sim N(\mu_k, \sigma_\delta^2), k = 1, \ldots, 10 \quad (16)$$

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.

We add extra parameter τ_k^2 to vary by k.

$$y_i \sim N(\mu + \gamma_{j[i]} + \delta_{k[i]}, \sigma_y^2), \text{ for } i = 1, \ldots, 300, j = 1, 2, 3 \quad (17)$$
$$\gamma_j \sim N(\mu_j, \sigma_\gamma^2), j = 1, \ldots, 100 \quad (18)$$
$$\delta_k \sim N(\mu_k, \tau_k^2), k = 1, \ldots, 10 \quad (19)$$
$$\tau_k^2 \sim Cauchy(0, 10) \quad (20)$$