MA678_homework_08

Jinfei XUE

Getting to know stan

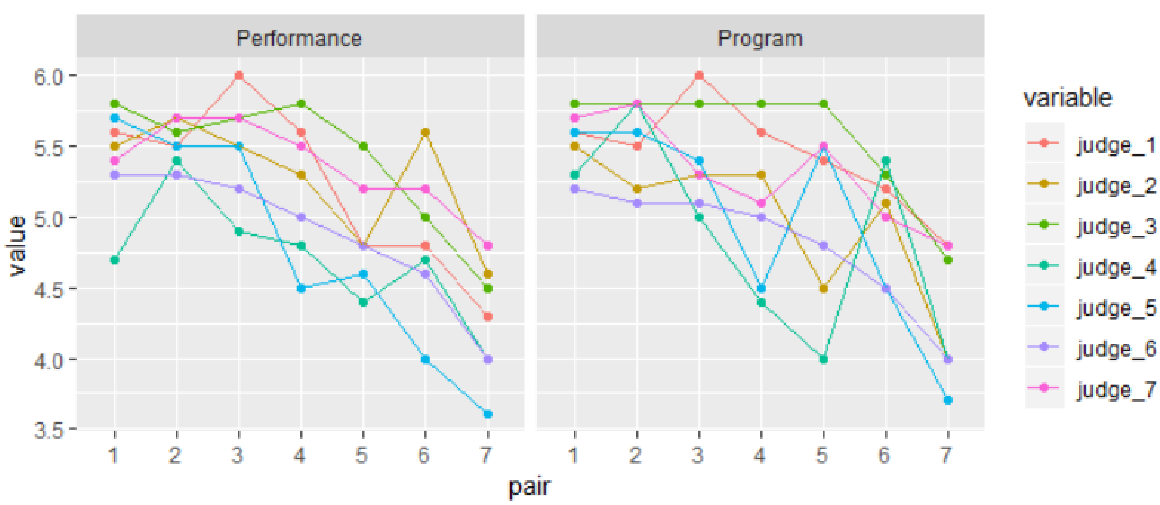
Read through the tutorial on Stan https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

- Explore Stan website and Stan reference manual and try to connect them with Gelman and Hill 16 - 17.

Data analysis

Using stan:

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




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

use stan to fit a non-nested multilevel model (varying across skaters and judges) for the technical merit ratings.

https://github.com/stan-dev/example-models/blob/master/ARM/Ch.17/17.3_flight_simulator.stan
https://github.com/stan-dev/example-models/blob/master/ARM/Ch.17/17.3_non-nested_models.R

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

dataList.1 <- list(N=49, n_judges=7, n_pairs=7,  judge=as.integer(olymp
ics_long$Judge), pair=as.integer(olympics_long$pair), y=olympics_long$P
rogram)

skating_stan<-"
data {
  int<lower=0> N;
  int<lower=0> n_judges;
  int<lower=0> n_pairs;
  int<lower=0,upper=n_judges> judge[N];
  int<lower=0,upper=n_pairs> pair[N];
  vector[N] y;
}
parameters {
  real<lower=0> sigma;
  real<lower=0> sigma_gamma;
  real<lower=0> sigma_delta;
  vector[n_judges] gamma;
  vector[n_pairs] delta;
  real mu;
}
model {
  vector[N] y_hat;

  sigma ~ uniform(0, 100);
  sigma_gamma ~ uniform(0, 100);
  sigma_delta ~ uniform(0, 100);

  mu ~ normal(0, 100);

  gamma ~ normal(0, sigma_gamma);
  delta ~ normal(0, sigma_delta);

  for (i in 1:N)
    y_hat[i] = mu + gamma[judge[i]] + delta[pair[i]];
  y ~ normal(y_hat, sigma);
}
"
```

pilots <- read.table
("http://www.stat.columbia.edu/~gelman/arm/examples/pilots/pilots.dat", header=TRUE)

flight_simulator.sf1 <- stan(model_code=skating_stan, data=dataList.1, iter=2000, chains=4)

Multilevel logistic regression

The folder `speed.dating` contains data from an experiment on a few hundred students that randomly assigned each participant to 10 short dates with participants of the opposite sex (Fisman et al., 2006). For each date, each person recorded several subjective numerical ratings of the other person (attractiveness, compatibility, and some other characteristics) and also wrote down whether he or she would like to meet the other person again. Label $y_{ij} = 1$ if person i is interested in seeing person j again 0 otherwise. And $r_{ij1}, \dots, r_{ij6}$ as person i's numerical ratings of person j on the dimensions of attractiveness, compatibility, and so forth. Please look at
http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/Speed%20Dating%20Data%20Key.doc for details.

```
dating<-fread("http://www.stat.columbia.edu/~gelman/arm/examples/speed.
dating/Speed%20Dating%20Data.csv")
```

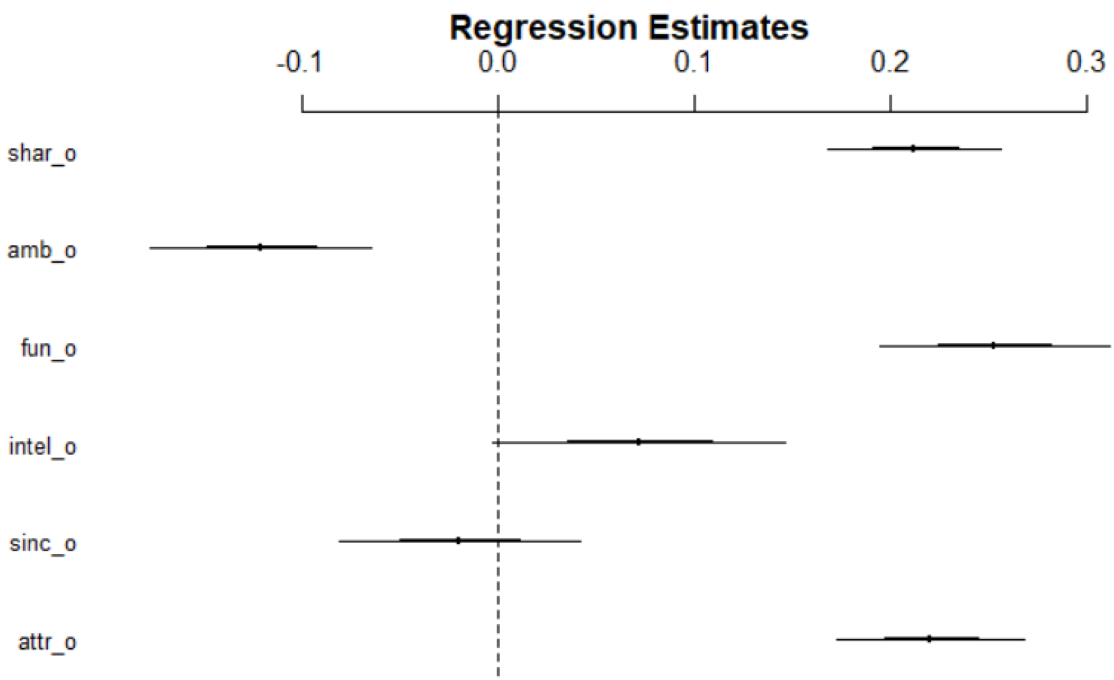
1. Fit a classical logistic regression predicting $Pr(y_{ij} = 1)$ given person i's 6 ratings of person j. Discuss the importance of attractiveness, compatibility, and so forth in this predictive model.

```
#classical logistic model without mixed effcects
#with attr_o: attractiveness, sinc_o: sincerity, intel_o: intelligence,
 fun_o: fun, amb_o: ambition, shar_o: shared interest/hobbies
log_1 <- glm(match~attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o,data=d
ating,family=binomial)
summary(log_1)

##
## Call:
## glm(formula = match ~ attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o, family = binomial, data = dating)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5300  -0.6362  -0.4420  -0.2381   3.1808
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.62091    0.21859 -25.714  < 2e-16 ***
## attr_o       0.22047    0.02388   9.233  < 2e-16 ***
## sinc_o      -0.01996    0.03067  -0.651   0.5152
## intel_o      0.07176    0.03716   1.931   0.0535 .
```

```
## fun_o        0.25315    0.02922   8.665  < 2e-16 ***
## amb_o       -0.12099    0.02838  -4.264 2.01e-05 ***
## shar_o       0.21225    0.02209   9.608  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6466.6  on 7030  degrees of freedom
## Residual deviance: 5611.0  on 7024  degrees of freedom
##   (1347 observations deleted due to missingness)
## AIC: 5625
##
## Number of Fisher Scoring iterations: 5
```

```
coefplot(log_1)
```



The result of this model can be written like:

$$P(match = 1) = logit^{-1}(-5.6 + 0.22attr - 0.02sinc + 0.07intel + 0.25fun - 0.12amb + 0.21shar)$$

.

(If we apply the “divided by 4” method to explain coefficients)

Attractiveness: $0.22/4 = 0.055$, one point higher in attractiveness will lead to 5.5% higher willingness of another date.

Sincerity: $-0.02/4 = -0.005$, one point higher in sincerity will lead to 0.5% lower willingness of another date, which is contrary to our expectation.

Intelligence: $0.07/4 = 0.0175$, one point higher in intelligence will lead to 1.75% higher willingness of another date.

Fun: $0.25/4 = 0.0625$, one point higher in humor will lead to 6.25% higher willingness of another date.

Ambition: $-0.12/4 = -0.03$, one point higher in ambition will lead to 3% lower willingness of another date.

Shared interest: $0.21/4 = 0.0525$, one point higher in shared interest will lead to 5.25% higher willingness of another date.

According to these results, the most important to determine a person's willingness to date again is humor and attractiveness.

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

```
log_mix <- lmer(match~gender+scale(attr_o) +scale(sinc_o) +scale(intel_
o) +scale(fun_o) +scale(amb_o) +scale(shar_o)+(1|iid),data=dating,famil
y=binomial(link="logit"))
summary(log_mix)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## match ~ gender + scale(attr_o) + scale(sinc_o) + scale(intel_o) +
##     scale(fun_o) + scale(amb_o) + scale(shar_o) + (1 | iid)
##    Data: dating
## Control:
## structure(list(optimizer = c("bobyqa", "Nelder_Mead"), calc.derivs =
 TRUE,
##     use.last.params = FALSE, restart_edge = FALSE, boundary.tol = 1e
-05,
##     tolPwrss = 1e-07, compDev = TRUE, nAGQ0initStep = TRUE, checkCon
trol = list(
##         check.nobs.vs.rankZ = "ignore", check.nobs.vs.nlev = "stop",

##         check.nlev.gtreq.5 = "ignore", check.nlev.gtr.1 = "stop",
##         check.nobs.vs.nRE = "stop", check.rankX = "message+drop.cols
",
##         check.scaleX = "warning", check.formula.LHS = "stop",
##         check.response.not.const = "stop"), checkConv = list(
##         check.conv.grad = list(action = "warning", tol = 0.001,
##             relTol = NULL), check.conv.singular = list(action = "ign
ore",
```

```
##              tol = 1e-04), check.conv.hess = list(action = "warning",

##              tol = 1e-06)), optCtrl = list()), class = c("glmerContro
l",
## "merControl"))
##
##      AIC      BIC   logLik deviance df.resid
##   5543.2   5605.0  -2762.6   5525.2     7022
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7458 -0.4453 -0.2877 -0.1454 10.3764
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  iid    (Intercept) 0.4294   0.6553
## Number of obs: 7031, groups:  iid, 551
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.13226    0.07079 -30.122  < 2e-16 ***
## gender          0.15452    0.09322   1.658   0.0974 .
## scale(attr_o)   0.46048    0.05203   8.850  < 2e-16 ***
## scale(sinc_o)  -0.02474    0.05728  -0.432   0.6658
## scale(intel_o)  0.10874    0.06203   1.753   0.0796 .
## scale(fun_o)    0.51341    0.06192   8.291  < 2e-16 ***
## scale(amb_o)   -0.23570    0.05468  -4.311 1.63e-05 ***
## scale(shar_o)   0.48473    0.05045   9.609  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) gender scl(t_) scl(sn_) scl(n_) scl(f_) scl(m_)
## gender       -0.672
## scale(ttr_) -0.202  0.109
## scale(snc_) -0.022  0.048 -0.123
## scale(ntl_)  0.026 -0.055 -0.039  -0.466
## scale(fun_) -0.156  0.015 -0.246  -0.150   -0.132
## scale(amb_)  0.143 -0.092 -0.062  -0.014   -0.370  -0.187
## scale(shr_) -0.135  0.009 -0.100  -0.054   -0.005  -0.268  -0.203
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 6 negative eigenv
alues

#ranef(log_mix)
```

$$P(match = 1)$$
$$= logit^{-1}(-2.13 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc)$$
$$+ 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar) + iid_i)$$

Among fixed effects: Gender: a male dating partner will receive approximately $0.16/4 = 0.0375$ 4% higher score from the person who give rates.

Attractiveness: $0.46/4 = 0.115$, one point higher than average attractiveness rate will lead to 11.5% higher willingness of another date.

Sincerity: $-0.02/4 = -0.005$, one point higher than average sincerity rate will lead to 0.5% lower willingness of another date, which is contrary to our expectation.

Intelligence: $0.11/4 = 0.0275$, one point higher than average intelligence rate will lead to 2.75% higher willingness of another date.

Fun: $0.51/4 = 0.1275$, one point higher than average humor rate will lead to 12.75% higher willingness of another date.

Ambition: $-0.23/4 = -0.0575$, one point higher than average ambition rate will lead to 5.75% lower willingness of another date.

Shared interest: $0.48/4 = 0.12$, one point higher than average shared interest rate will lead to 12% higher willingness of another date.

For random effects: If we take the first 5 people for example: 1 0.493948779

2 -0.181380981

3 -0.467790677

4 -0.107449058

5 0.124412503

The model with random effects for these five people can be written as follow:

$$P(match = 1) = logit^{-1}(-1.64 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.31 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.59 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.24 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.01 + 0.15gender + 0.46scale(attr) - 0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar))$$

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

```
log_mix2 <- glmer(match~gender+scale(attr_o) +scale(sinc_o) +scale(inte
l_o) +scale(fun_o) +scale(amb_o) +scale(shar_o)+(1|iid)+(1|pid),data=da
ting,family=binomial)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.217
492
## (tol = 0.001, component 1)

summary(log_mix2)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## match ~ gender + scale(attr_o) + scale(sinc_o) + scale(intel_o) +
##     scale(fun_o) + scale(amb_o) + scale(shar_o) + (1 | iid) +
##     (1 | pid)
##    Data: dating
##
##      AIC      BIC   logLik deviance df.resid
##   5257.6   5326.1  -2618.8   5237.6     7021
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7839 -0.3829 -0.2197 -0.0918  9.1741
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  iid    (Intercept) 0.594    0.7707
##  pid    (Intercept) 1.254    1.1199
## Number of obs: 7031, groups:  iid, 551; pid, 537
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.53701    0.11724 -21.639  < 2e-16 ***
## gender          0.17624    0.14929   1.180   0.2378
## scale(attr_o)   0.63703    0.06370  10.000  < 2e-16 ***
## scale(sinc_o)   0.03504    0.06782   0.517   0.6054
## scale(intel_o)  0.17216    0.07357   2.340   0.0193 *
## scale(fun_o)    0.57822    0.07097   8.147 3.73e-16 ***
## scale(amb_o)   -0.16857    0.06463  -2.608   0.0091 **
## scale(shar_o)   0.58936    0.06155   9.575  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) gender scl(t_) scl(sn_) scl(n_) scl(f_) scl(m_)
```

```
## gender      -0.647
## scale(ttr_) -0.221  0.094
## scale(snc_) -0.049  0.037 -0.065
## scale(ntl_) -0.009 -0.044 -0.024  -0.438
## scale(fun_) -0.140  0.009 -0.220  -0.123   -0.098
## scale(amb_)  0.072 -0.070 -0.051   0.011   -0.334  -0.168
## scale(shr_) -0.139  0.005 -0.072  -0.057   -0.020  -0.234  -0.159
## convergence code: 0
## Model failed to converge with max|grad| = 0.217492 (tol = 0.001, com
ponent 1)
```

```r
a_iid <- data.frame(ranef(log_mix2))[1:5,4]
a_pid <- data.frame(ranef(log_mix2))[552:556,4]
a_iid
```

```
## [1]  0.452970133 -0.463750802 -0.795488077 -0.323614171 -0.004300494
```

```r
a_pid
```

```
## [1]  0.97137439  0.11231247 -1.78993378 -0.63479896  0.04949994
```

This model has added another varying intercept, if we still take the first five people as example, now their models will become (taking the changed fixed effects into consideration):

$$P(match = 1) = logit^{-1}(-1.11 + 0.16gender + 0.64scale(attr)0.035scale(sinc) + 0.17scale(intel) + 0.58scale(fun) - 0.16scale(amb) + 0.59scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.88 + 0.16gender + 0.64scale(attr)0.035scale(sinc) + 0.17scale(intel) + 0.58scale(fun) - 0.16scale(amb) + 0.59scale(shar))$$

$$P(match = 1) = logit^{-1}(-5.12 + 0.16gender + 0.64scale(attr)0.035scale(sinc) + 0.17scale(intel) + 0.58scale(fun) - 0.16scale(amb) + 0.59scale(shar))$$

$$P(match = 1) = logit^{-1}(-3.48 + 0.16gender + 0.64scale(attr)0.035scale(sinc) + 0.17scale(intel) + 0.58scale(fun) - 0.16scale(amb) + 0.59scale(shar))$$

$$P(match = 1) = logit^{-1}(-2.49 + 0.16gender + 0.64scale(attr)0.035scale(sinc) + 0.17scale(intel) + 0.58scale(fun) - 0.16scale(amb) + 0.59scale(shar))$$

4. You will now fit some models that allow the coefficients for attractiveness, compatibility, and the other attributes to vary by person. Fit a no-pooling model: for each person i, fit a logistic regression to the data y_{ij} for the 10 persons j whom he or she rated, using as predictors the 6 ratings $r_{ij1}, \dots, r_{ij6}$.

(Hint: with 10 data points and 6 predictors, this model is difficult to fit. You will need to simplify it in some way to get reasonable fits.)

```
#No pooling model for each person i
no_pooling <- glm(match~attr_o + sinc_o + intel_o + fun_o + amb_o + sha
r_o + factor(iid)-1,data=dating)
```

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

```
#Vary characteristics by person
vary_slope <- glmer(match~(1+attr_o+sinc_o+intel_o+fun_o+amb_o+shar_o|i
id) + attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o,data=dating,fa
mily=binomial)

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

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

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

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian wi
th 3
## negative eigenvalues
```

6. Compare the inferences from the multilevel model in (5) to the no-pooling model in (4) and the complete-pooling model from part (1) of the previous exercise.

```
anova(vary_slope,log_1,no_pooling)

## Data: dating
## Models:
## log_1: match ~ attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o
## vary_slope: match ~ (1 + attr_o + sinc_o + intel_o + fun_o + amb_o +
 shar_o |
## vary_slope:     iid) + attr_o + sinc_o + intel_o + fun_o + amb_o + s
har_o
## no_pooling: match ~ attr_o + sinc_o + intel_o + fun_o + amb_o + shar
_o +
## no_pooling:     factor(iid) - 1
##              Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chis
q)
## log_1         7 5625.0 5673.0 -2805.5   5611.0


## vary_slope   35 5576.8 5816.8 -2753.4   5506.8  104.2     28  1.047e-
10 ***
```

```
## no_pooling 558 5607.8 9434.6 -2245.9   4491.8 1015.0    523  < 2.2e-
16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the anova test, we can see that the deviance and the AIC of no pooling model are the lowest, which is weird.