MA678_homework_08

Yifu Dong

November 24, 2018

Getting to know stan

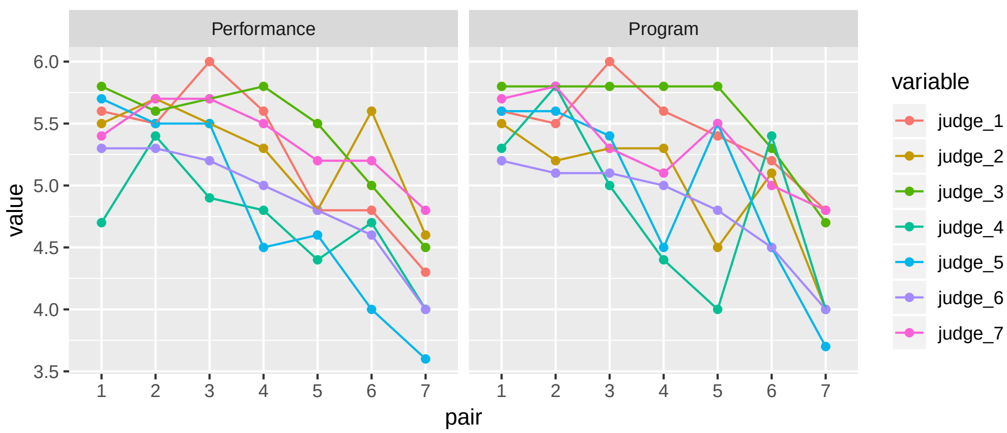
Read through the tutorial on Stan https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started

- 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.

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

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

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

```r
dataList.1 <- list(N=49, n_judges=7, n_pairs=7,  judge=as.integer(olympics_long$Judge), pair=as.integer

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=TRU

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

```
##
## SAMPLING FOR MODEL 'a6b319833942b0687339229360b24f51' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.5e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.177916 seconds (Warm-up)
## Chain 1:                0.268067 seconds (Sampling)
## Chain 1:                0.445983 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'a6b319833942b0687339229360b24f51' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.213805 seconds (Warm-up)
## Chain 2:                0.200507 seconds (Sampling)
## Chain 2:                0.414312 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'a6b319833942b0687339229360b24f51' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.3e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.207818 seconds (Warm-up)
## Chain 3:                0.143123 seconds (Sampling)
## Chain 3:                0.350941 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'a6b319833942b0687339229360b24f51' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.177056 seconds (Warm-up)
## Chain 4:                0.161866 seconds (Sampling)
## Chain 4:                0.338922 seconds (Total)
## Chain 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 $r_{ij1}, \ldots, r_{ij6}$ as person i's numerical ratings of person j on the dimensions of attractiveness, compatibility, and so forth. Please look at http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/Speed%20Dating%20Data%20Key.doc for details.

```
dating<-fread("http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/Speed%20Dating%20Data.csv
dating_pooled <- glm(match~attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o,data=dating,family=binomial)
dating_pooled <- glmer(match~gender + attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o+(1|iid)+(1|pid),data
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
```

```
## control$checkConv, : Model failed to converge with max|grad| = 0.67667 (tol
## = 0.001, component 1)
```

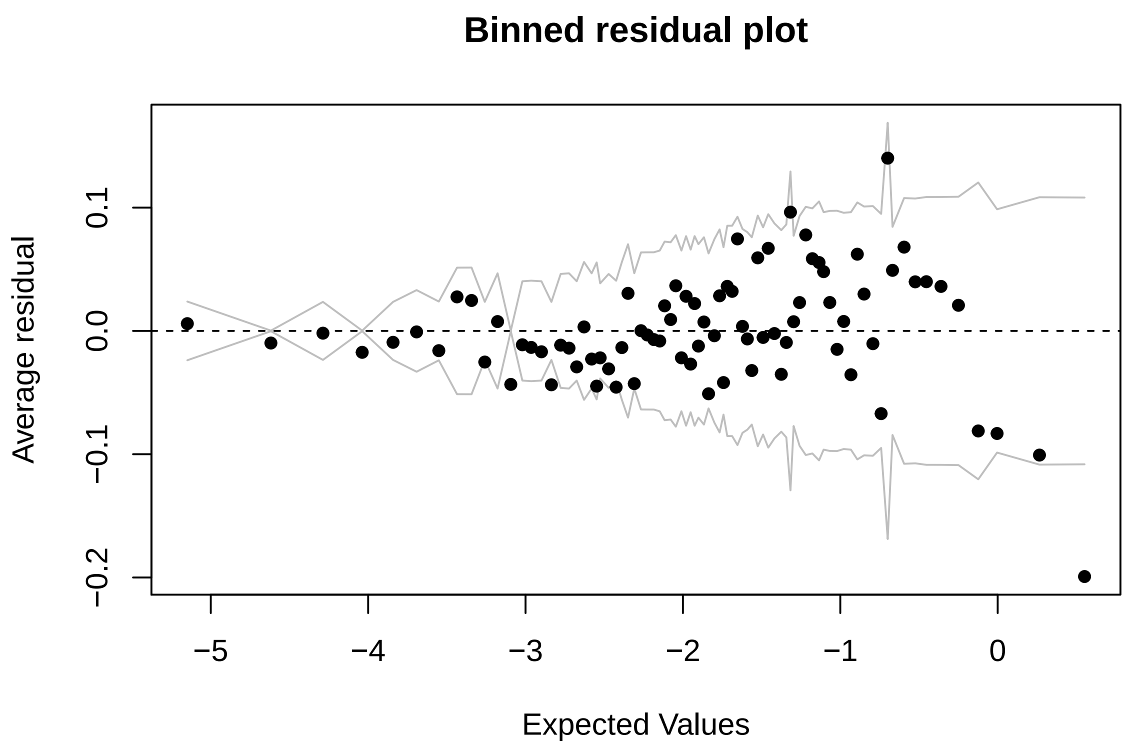
1. Fit a classical logistic regression predicting $Pr(y_{ij} = 1)$ given person i's 6 ratings of person j. Discuss the importance of attractiveness, compatibility, and so forth in this predictive model.

```r
m1_1 <- glm(match~attr_o +sinc_o +intel_o +fun_o +amb_o +shar_o,data=dating,family=binomial(link="logit
summary(m1_1)
```

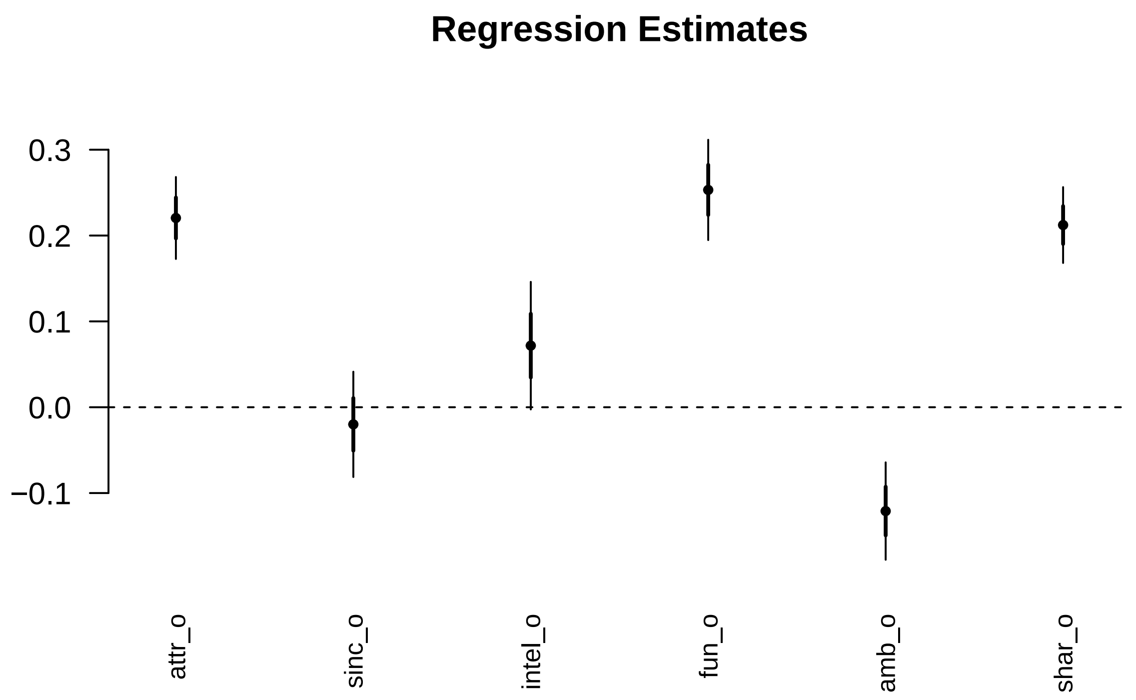
```
##
## Call:
## glm(formula = match ~ attr_o + sinc_o + intel_o + fun_o + amb_o +
##     shar_o, family = binomial(link = "logit"), data = dating)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5300  -0.6362  -0.4420  -0.2381   3.1808
##
## Coefficients:
##             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
```

Now let's see the goodness of fit. We found that:

```r
binnedplot(predict(m1_1),resid(m1_1,type = "response"))
```




```r
coefplot(m1_1,vertical=FALSE)
```



We found that our model actually fit not bad. Almost all residual plots are in the interval.

Also, the second plot shows the importance of each coefficients, where the farthest from y=0 represents the variable having biggest influence on the response.

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.

For this request, we will use multilevel regression to fit the model.

```r
m1_2 <- glmer(match~gender+scale(attr_o) +scale(sinc_o) +scale(intel_o) +scale(fun_o) +scale(amb_o) +sca
```

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

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 5
## negative eigenvalues
```

```r
summary(m1_2)
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from 
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## 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
##
##      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)   0.46047    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)   0.48474    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 5 negative eigenvalues
```

From the result, we know that the function is : $P(match = 1) = logit^{-1}(-2.13+0.15gender+0.46scale(attr)-0.02scale(sinc) + 0.11scale(intel) + 0.51scale(fun) - 0.23scale(amb) + 0.48scale(shar) + iid_i)$

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

Now we add another multilevel:

```
m1_3 <- glmer(match~gender+scale(attr_o) +scale(sinc_o) +scale(intel_o) +scale(fun_o) +scale(amb_o) +sca
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.263517
## (tol = 0.001, component 1)
```

```
summary(m1_3)
```

```
## 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
## -3.7847 -0.3824 -0.2194 -0.0917  9.1546
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  iid    (Intercept) 0.595    0.7713
##  pid    (Intercept) 1.262    1.1235
## Number of obs: 7031, groups:  iid, 551; pid, 537
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.53475    0.11733 -21.604  < 2e-16 ***
## gender          0.16773    0.14956   1.121   0.2621
## scale(attr_o)   0.63906    0.06376  10.023  < 2e-16 ***
## scale(sinc_o)   0.03499    0.06786   0.516   0.6061
## scale(intel_o)  0.17125    0.07360   2.327   0.0200 *
## scale(fun_o)    0.57661    0.07099   8.122 4.59e-16 ***
## scale(amb_o)   -0.16544    0.06466  -2.559   0.0105 *
## scale(shar_o)   0.58881    0.06158   9.561  < 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       -0.646
## scale(ttr_) -0.221  0.092
## scale(snc_) -0.049  0.036 -0.064
## scale(ntl_) -0.009 -0.044 -0.024  -0.438
## scale(fun_) -0.139  0.008 -0.220  -0.123   -0.098
## scale(amb_)  0.072 -0.070 -0.051   0.011   -0.334  -0.167
## scale(shr_) -0.138  0.004 -0.072  -0.057   -0.020  -0.234  -0.158
## convergence code: 0
## Model failed to converge with max|grad| = 0.263517 (tol = 0.001, component 1)
```

```r
iid <- data.frame(ranef(m1_3))[1:5,4]
pid <- data.frame(ranef(m1_3))[552:556,4]
iid
```

```
## [1]  0.452243548 -0.465241025 -0.797514626 -0.324989639 -0.005380029
```

```r
pid
```

```
## [1]  0.97842409  0.11652972 -1.79408020 -0.63282281  0.05274865
```

4. You will now fit some models that allow the coefficients for attractiveness, compatibility, and the other attributes to vary by person. Fit a no-pooling model: for each person i, fit a logistic regression to the data y_{ij} for the 10 persons j whom he or she rated, using as predictors the 6 ratings $r_{ij1}, \ldots, r_{ij6}$. (Hint: with 10 data points and 6 predictors, this model is difficult to fit. You will need to simplify it in some way to get reasonable fits.)

First, let's fit the no-pooling model:

```r
m1_4 <- glm(match~scale(attr_o) +scale(sinc_o) +scale(intel_o) +scale(fun_o) +scale(amb_o) +scale(shar_o
 + factor(iid)-1,data=dating)
```

It shows that for every single iid, there is a unique intercept for it. The AIC for this no-pooling model is 5607.8

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

```r
m1_5 <- glmer(match~(1+attr_o+sinc_o+intel_o+fun_o+amb_o+shar_o|iid) + attr_o + sinc_o + intel_o + fun_o
```

```
## 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 with 1
## 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.

```r
anova(m1_5,m1_1,m1_4)
```

```
## Data: dating
```

```
## Models:
## m1_1: match ~ attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o
## m1_5: match ~ (1 + attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o |
## m1_5:     iid) + attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o
## m1_4: match ~ scale(attr_o) + scale(sinc_o) + scale(intel_o) + scale(fun_o) +
## m1_4:     scale(amb_o) + scale(shar_o) + factor(iid) - 1
##       Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m1_1   7 5625.0 5673.0 -2805.5   5611.0
## m1_5  35 5576.8 5816.8 -2753.4   5506.8  104.23     28  1.034e-10 ***
## m1_4 558 5607.8 9434.6 -2245.9   4491.8 1014.93    523  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```