# Bayesian longitudinal item response with an application to educational assessments

Lisa Wilson

June 12, 2020

**Introduction**

Item response models are used to analyze responses from surveys, assessments, and other settings where respondents answer a set number of questions using a discrete set of answers. For simplicity, the models discussed for this project assume that all questions have only two possible answers. In general, item response models aim to measure and analyze the underlying abilities of individual respondents and the difficulty of each question. Following a 2019 paper by Anna Scharl and Timo Gnabs in *The Quantitative Methods for Psychology*, Bayesian longitudinal analysis can be applied to educational assessment data to estimate latent abilities of respondents and question difficulties.

**Item Response Models**

One fundamental item response model is the Rasch model, introduced in 1960, which represents the probability of individual $i$ answering question $j$ "correctly" as

$$P(Y_{ij} = 1|\theta_i, \beta_j) = logit^{-1}(\theta_i - \beta_j)$$

where $\theta_i$ is the latent ability of individual $i$ and $\beta_j$ is the difficulty of item $j$.

The Rasch model assumes that each question is equally difficult for all individuals. To relax this assumption, Birnbaum developed the two-parameter logistic (2PL) model in 1960, given below:

$$P(Y_{ij} = 1|\theta_i, \alpha_j, \beta_j) = logit^{-1}(\alpha_j\theta_i - \beta_j)$$

where $\alpha_j$ measures how much item $j$ differentiates between individuals of high and low abilities.

Longitudinal extensions of these item response models are straightforward, allowing for the analysis of assessments given over multiple time points. Under the longitudinal Rasch extension, $\theta_i$ becomes a vector of latent abilities for individual $i$ over all time points $t$:

$$P(Y_{ij} = 1|\theta_i, \beta_j) = logit^{-1}(\sum_t \theta_{it} - \beta_j)$$

Similarly, the $\alpha_i$ are also vectors with length equal to the number of time points over which assessments were given in the longitudinal two-parameter logistic extension:

$$P(Y_{ij} = 1|\theta_i, \alpha_j, \beta_j) = logit^{-1}(\sum_t \alpha_{jt}\theta_{it} - \beta_j)$$

Increased computational efficiency has made Bayesian approaches to item response analysis more appealing in recent years. Following guidance given in Scharl & Gnambs (2019), the following disctributions are commonly used as weakly informative priors for the parameters $\alpha$, $\beta$, and $\theta$:

| Parameter | Distribution |
|---|---|
| $\alpha$ | lognormal or truncated normal |
| $\beta$ | normal |
| $\theta$ | multivariate normal (bivariate if 2 time points) |

### Pre-Kindergarten Assessment Application

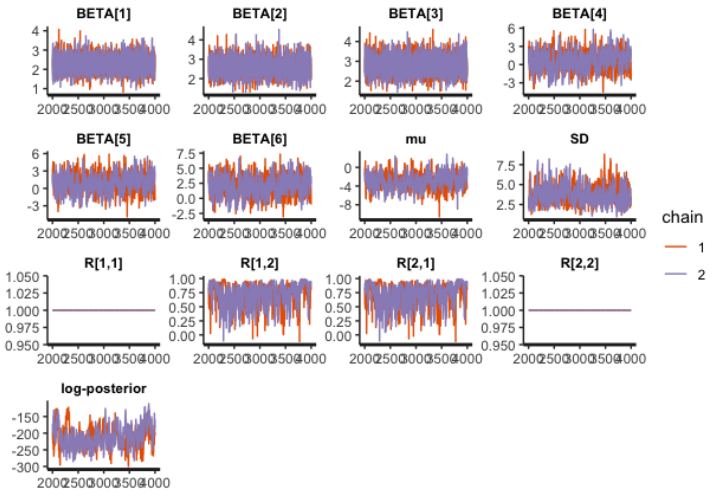
To demonstrate this longitudinal Bayesian analysis, educational assessment data was found from the National Center for Early Development and Learning. During the 2001-2002 school year, the Multi-State Study of Pre-Kindergarten was carried out in California, Georgia, Illinois, Kentucky, New York, and Ohio. Pre-kindergarten teachers from sampled publicly-funded programs were asked to rate their students' verbal, writing, and math skills using a questionnaire that was administered in both the fall and spring. Responses for these skill questions were "Not yet," "Beginning," "In progress," "Intermediate," and "Proficient." To keep the responses binary, as appropriate for the models described above, responses were recoded as either "proficient" or "not proficient" before analysis. To create a smaller dataset, the data used for the following analysis includes students from California who were assessed in both the fall and spring rounds of questionnaires and includes three verbal ability questions about whether the student:

- uses complex sentence structures
- understands and interprets a story read to them
- predicts what will happen next in stories.

Because this data includes results of assessments given at two time points, the longitudinal item response models given above are appropriate for analyzing how individual latent abilities and question difficulties changed between the fall and spring. Stan and R code for this analysis was adapted from code provided in the supplement to Scharl & Gnambs (2019).

### Longitudinal Rasch Model

Assessing diagnostics for the longitudinal Rasch model, the traceplots for model parameters suggest that the chains mixed well and indicate convergence.

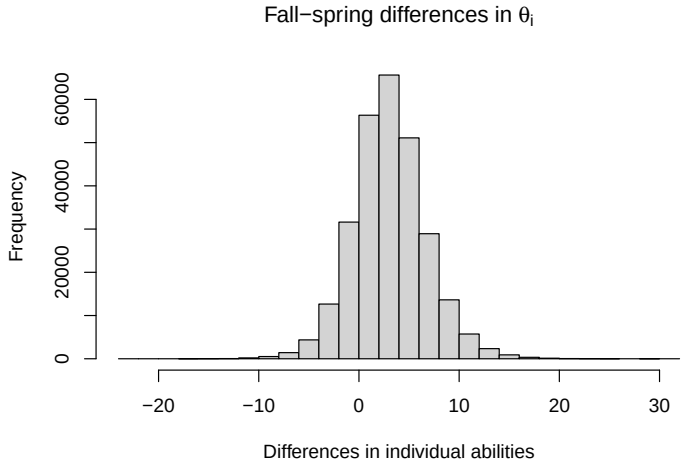



Additionally, the $\hat{R}$ values for all parameters are below 1.1, also indicating convergence.

Posterior parameter means are given in the table below. $\beta_1, \beta_2$, and $\beta_3$ represent item difficulties for the three verbal skills questions in the fall, and $\beta_4, \beta_5$, and $\beta_6$ represent item difficulties for the same questions on the spring questionnaire. The values are similar for the fall $\beta$ parameter means, indicating that all three questions were of similar difficulty in the fall. For the spring $\beta$ parameter means, the values for the first two questions have dropped to less than 1, indicating that those questions were easier for students in the spring. While the mean for $\beta_6$ is also lower than for $\beta_3$, suggesting the third question may have been less difficult in the spring than in the fall, it remains higher than the posterior means for $\beta_4$ and $\beta_5$, indicating that the third question was more difficult than the first two in the spring. As a note, the posterior standard deviation for the spring $\beta$ are higher than for the fall $\beta$, meaning that the differences between item difficulties in the spring may not be as pronounced as suggested by the point estimates.

| | Posterior mean | Posterior standard deviation |
|---|---|---|
| $\beta_1$ | 2.265 | 0.4092 |
| $\beta_2$ | 2.593 | 0.4353 |
| $\beta_3$ | 2.795 | 0.4608 |
| $\beta_4$ | 0.6328 | 1.532 |
| $\beta_5$ | 0.8992 | 1.526 |
| $\beta_6$ | 2.071 | 1.572 |

The posterior distribution of differences in $\theta_i$ can also be analyzed to assess whether the underlying abilities of students changed between the fall and spring.



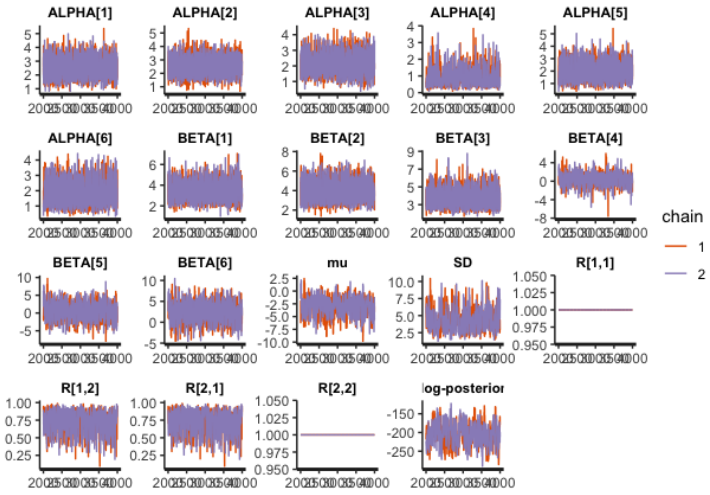

While the 95% Bayesian credible interval given below for $\theta_{i1} - \theta_{i2}$ contains 0, meaning it was plausible that there was no improvement, the posterior probability that $\theta_{i1} - \theta_{i2}$ was greater than 0 is 0.8158. There is a high probability, therefore, that individual students experienced improvement in their underlying verbal ability skills between the fall and spring of the school year.

| 2.5% | 50% | 97.5% |
|---|---|---|
| -3.91 | 2.922 | 10.69 |

**Longitudinal 2PL Model**

Similar to the diagnostics for the longitudinal Rasch model, the traceplots for the longitudinal 2PL model

parameters suggest that the chains mixed well and indicate convergence. The $\hat{R}$ values for all parameters are also below 1.1, indicating convergence.




The similar posterior means for $\alpha_1, \alpha_2$, and $\alpha_3$ indicate that the three questions did a similar job of differentiating between low and high ability students on the fall questionnaire. By contrast, the relatively lower posterior mean of $\alpha_4$ compared to $\alpha_5$ and $\alpha_6$ suggest that the first question on the spring questionnaire was less effective at distinguishing between students of different abilities than the other two.

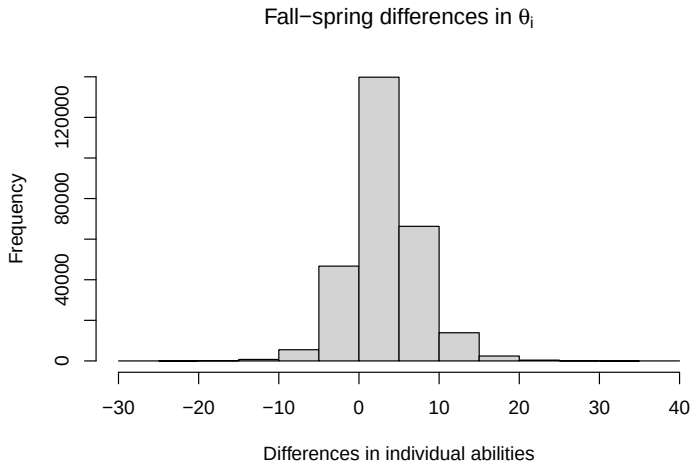
| | Posterior mean | Posterior standard deviation |
|---|---|---|
| $\alpha_1$ | 2.519 | 0.6388 |
| $\alpha_2$ | 2.539 | 0.6098 |
| $\alpha_3$ | 2.094 | 0.6029 |
| $\alpha_4$ | 0.8457 | 0.4343 |
| $\alpha_5$ | 1.958 | 0.6776 |
| $\alpha_6$ | 1.813 | 0.6751 |

The interpretation of the posterior means of the $\beta$ parameters is very similar to that of the longitudinal Rasch model, where the three questions seem to be of similar difficulty in the fall while the third was more difficult than the first two in the spring. While the difference in posterior standard deviation between the fall and spring is less pronounced than in the Rasch analysis, the larger posterior standard deviation for the spring parameters, particularly for the second question, again raises issues about the magnitude of differences in the item difficulty parameters.

| | Posterior mean | Posterior standard deviation |
|---|---|---|
| $\beta_1$ | 3.364 | 0.8026 |
| $\beta_2$ | 3.948 | 0.8952 |
| $\beta_3$ | 3.791 | 0.865 |
| $\beta_4$ | 0.3695 | 1.127 |
| $\beta_5$ | 0.4051 | 2.109 |
| $\beta_6$ | 2.083 | 1.994 |

The posterior distribution of $\theta_{i1} - \theta_{i2}$ indicates very similar results to the Rasch analysis, with again a high probability that students' underlying verbal abilities improved between the fall and spring: $P(\theta_{i1} - \theta_{i2} >$

0) = 0.8076. The histogram and 95% credible interval for $\theta_{i1} - \theta_{i2}$ are given below.




| 2.5% | 50% | 97.5% |
|---|---|---|
| -4.816 | 3.026 | 12.57 |

## Conclusion

Applying a Bayesian analysis to both the Rasch and 2PL longitudinal item response models can be done efficiently in Stan. Overall, the results of the two models agree closely for the pre-kindergarten verbal ability data. Both the longitudinal Rasch and 2PL models indicate that the items on the spring assessment were less difficult than in the fall. While all three items were of similar difficulty in the fall, item 3 on predicting story plot seemed to be more difficult in the spring compared to item 1 on using complex sentences and item 2 on interpreting stories. Results from both models also show evidence that for the majority of students, there was improvement in latent verbal abilities between fall and spring.

## References

1. Anna Scharl & Timo Gnambs, "Longitudinal item response modeling and posterior predictive checking in R and Stan," The Quantitative Methods for Psychology 15 (2019): 75-95. [Link at the end of the paper provides a download of the authors' code appendix.]

2. "Pre-Kindergarten in Eleven States: NCEDL's Multi-State Study of Pre-Kindergarten and Study of State-Wide Early Education Programs (SWEEP) (ICPSR 34877)." Child & Family Data Archive. October 2, 2013. Accessed May 13, 2020. https://www.childandfamilydataarchive.org/cfda/archives/cfda/studies/34877/summary.

## Code appendix

```
library(tidyverse)
library(rstan)
library(edstan)
```

```
# https://www.childandfamilydataarchive.org/cfda/archives/cfda/studies/34877/summary
load(file = "~/Documents/ST 559/ICPSR_34877/DS0002/34877-0002-Data.rda")
dat <- da34877.0002
str(dat)
head(dat)

# 1 is "child uses complex sentences",
# 3 is "child easily & quickly names all upper and lowercase letters of alphabet"
cols <- c("ICPSR_STUDY_ID", "CHPARTP", "STATE",
          "CSLANGPF1", "CSLANGPF2", "CSLANGPF3", "CSLANGPF4", "CSLANGPF5", "CSLANGPF6",
          "CSLANGPF7", "CSLANGPF8", "CSLANGPF9",
          "CSLANGPS1", "CSLANGPS2", "CSLANGPS3", "CSLANGPS4", "CSLANGPS5", "CSLANGPS6",
          "CSLANGPS7", "CSLANGPS8", "CSLANGPS9")
complex <- dat[,cols]
summary(complex)
summary(complex$STATE)

# probably convert to binary: 0 if 1-3, 1 if 4-5
# could filter to just one state if want fewer observations

complex <- complex %>%
    filter(STATE %in% c("(01) California", "(02) Illinois", "(03) Georgia",
                        "(04) Kentucky", "(05) New York", "(06) Ohio"),
           CHPARTP == "(1) Fall & Spring") %>%
    mutate(STATE = fct_drop(STATE)) %>%
    mutate(STATE = fct_recode(STATE, California = "(01) California", Illinois = "(02) Illinois",
                              Georgia = "(03) Georgia",
                Kentucky = "(04) Kentucky", NewYork = "(05) New York", Ohio = "(06) Ohio")) %>%
    mutate(CSLANGPF1 = fct_collapse(CSLANGPF1, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                  prof = "(5) Proficient"),
           CSLANGPS1 = fct_collapse(CSLANGPS1, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPF2 = fct_collapse(CSLANGPF2, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPS2 = fct_collapse(CSLANGPS2, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPF3 = fct_collapse(CSLANGPF3, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPS3 = fct_collapse(CSLANGPS3, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPF4 = fct_collapse(CSLANGPF4, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPS4 = fct_collapse(CSLANGPS4, notprof = c("(1) Not Yet", "(2) Beginning",
                                                           "(3) In Progress", "(4) Intermediate"),
                                    prof = "(5) Proficient"),
           CSLANGPF5 = fct_collapse(CSLANGPF5, notprof = c("(1) Not Yet", "(2) Beginning",
```

```
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPS5 = fct_collapse(CSLANGPS5, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPF6 = fct_collapse(CSLANGPF6, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPS6 = fct_collapse(CSLANGPS6, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPF7 = fct_collapse(CSLANGPF7, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPS7 = fct_collapse(CSLANGPS7, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPF8 = fct_collapse(CSLANGPF8, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPS8 = fct_collapse(CSLANGPS8, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPF9 = fct_collapse(CSLANGPF9, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient"),
         CSLANGPS9 = fct_collapse(CSLANGPS9, notprof = c("(1) Not Yet", "(2) Beginning",
                                                    "(3) In Progress", "(4) Intermediate"),
                                  prof = "(5) Proficient")) %>%
    select(ICPSR_STUDY_ID, STATE, CSLANGPF1, CSLANGPS1, CSLANGPF2, CSLANGPS2, CSLANGPF3, CSLANGPS3,
           CSLANGPF4, CSLANGPS4, CSLANGPF5, CSLANGPS5, CSLANGPF6, CSLANGPS6, CSLANGPF7, CSLANGPS7,
           CSLANGPF8, CSLANGPS8, CSLANGPF9, CSLANGPS9) %>%
    mutate(CSLANGPF1_bin = ifelse(CSLANGPF1 == "notprof", 0, 1),
           CSLANGPF2_bin = ifelse(CSLANGPF2 == "notprof", 0, 1),
           CSLANGPF3_bin = ifelse(CSLANGPF3 == "notprof", 0, 1),
           CSLANGPF4_bin = ifelse(CSLANGPF4 == "notprof", 0, 1),
           CSLANGPF5_bin = ifelse(CSLANGPF5 == "notprof", 0, 1),
           CSLANGPF6_bin = ifelse(CSLANGPF6 == "notprof", 0, 1),
           CSLANGPF7_bin = ifelse(CSLANGPF7 == "notprof", 0, 1),
           CSLANGPF8_bin = ifelse(CSLANGPF8 == "notprof", 0, 1),
           CSLANGPF9_bin = ifelse(CSLANGPF9 == "notprof", 0, 1),
           CSLANGPS1_bin = ifelse(CSLANGPS1 == "notprof", 0, 1),
           CSLANGPS2_bin = ifelse(CSLANGPS2 == "notprof", 0, 1),
           CSLANGPS3_bin = ifelse(CSLANGPS3 == "notprof", 0, 1),
           CSLANGPS4_bin = ifelse(CSLANGPS4 == "notprof", 0, 1),
           CSLANGPS5_bin = ifelse(CSLANGPS5 == "notprof", 0, 1),
           CSLANGPS6_bin = ifelse(CSLANGPS6 == "notprof", 0, 1),
           CSLANGPS7_bin = ifelse(CSLANGPS7 == "notprof", 0, 1),
           CSLANGPS8_bin = ifelse(CSLANGPS8 == "notprof", 0, 1),
           CSLANGPS9_bin = ifelse(CSLANGPS9 == "notprof", 0, 1),) %>%
    drop_na()

# adapted from 0_prepare_data.R
```

```r
# data for estimation in Bayesian estimation with Stan
complex_stan <- list(
    I = nrow(complex), # number of students
    J = c(9, 9), # number of items; can be scalar if same test multiple times
    Y = complex[, 21:38], # response data
    T = 2 # number of time points
)
# save raw data (Stan compatible)
# save(data, file = "data/data.RData")

# smaller dataset: just California, just "verbal" items
complex_CA <- complex %>%
    filter(STATE == "California") %>%
    select(ICPSR_STUDY_ID, CSLANGPF1_bin, CSLANGPF2_bin, CSLANGPF5_bin,
           CSLANGPS1_bin, CSLANGPS2_bin, CSLANGPS5_bin)
complex_stan_CA <- list(
    I = nrow(complex_CA),
    J = c(3, 3),
    Y = complex_CA[, -1],
    T = 2
)

# adapted from 1_estimate_models.R
# set parameters to be stored in final stanfit object
params <- list(lrasch = c("BETA", "mu", "SD", "R", "theta", "y_rep", "log_lik"),
               l2pl=c("ALPHA", "BETA", "mu", "SD", "R", "theta", "y_rep",
                      "log_lik"),
               l3pl=c("ALPHA", "BETA", "GAMMA", "mu", "SD", "R", "theta",
                      "y_rep", "log_lik"))

# data structure to store individual likelihoods in
log_lik <- list()

# might need more iteractions based on warning messages
# worked best so far with 4 chains, 4000 iter, 2000 warmup
# --> but not consistently!
# try with more cores?
stanfit1 <- stan(file = "~/Documents/ST 559/lrasch2.stan",
                 data = complex_stan_CA,
                 chains = 2, #2
                 iter = 4000, # 4000
                 warmup = 2000, # 2000
                 # thin = 4,
                 pars = params[[1]],
                 control = list(adapt_delta = 0.9)
)

# rasch
# convergence diagnostics
trace1 <- traceplot(stanfit1, pars = c('y_rep', 'theta', 'log_lik'),
                    include = FALSE)
rhat1 <- bayesplot::rhat(stanfit1, pars = params[[1]][1:(length(params[[1]])-3)])
scp1 <- edstan::stan_columns_plot(stanfit1, pars = params[[1]])
```

```r
scp1 + theme_bw()

# EAP estimators for further analysis and summary tables
# why just third chain? probably because only had 2 chaings --> want all chains
post_means1 <- get_posterior_mean(stanfit1,
                                  pars = params[[1]][1:(length(params[[1]])-3)])[, 3, drop=FALSE]
# get_posterior_mean(stanfit, pars = params[[1]][1:(length(params[[1]])-3)])
write.csv(post_means1, "post_means1.csv")

# extract and save data
y_rep1 <- extract(stanfit1, pars = "log_lik", include = FALSE)

beta1 <- cbind(colMeans(y_rep1$BETA), apply(y_rep1$BETA, 2, sd))
write.csv(beta1, "beta1.csv")

log_lik[[1]] <- extract(stanfit, pars = "log_lik")

hist(y_rep1$theta[,,1][1,])
hist(y_rep1$theta[,,1][2,])

# change in individual abilities?
hist(y_rep1$theta[,,1][1,] - y_rep1$theta[,,2][1,])
summary(y_rep1$theta[,,1][1,] - y_rep1$theta[,,2][1,])
summary(y_rep1$theta[,,1][2,] - y_rep1$theta[,,2][2,])
summary(colMeans(y_rep1$theta[,,1]) - colMeans(y_rep1$theta[,,2]))

hist(y_rep1$theta[,,1] - y_rep1$theta[,,2]) ## use this
pander(quantile(y_rep1$theta[,,1] - y_rep1$theta[,,2], probs = c(0.025, 0.5, 0.975))) ## use this
write.csv(y_rep1$theta, "y_rep1_theta.csv")

# difference in item difficult between fall and spring?
summary(y_rep1$BETA[,4:6] - y_rep1$BETA[,1:3])
quantile(y_rep1$BETA[,4] - y_rep1$BETA[,1], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,5] - y_rep1$BETA[,2], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,6] - y_rep1$BETA[,3], probs = c(0.025, 0.5, 0.975))

quantile(y_rep1$BETA[,1], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,2], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,3], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,4], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,5], probs = c(0.025, 0.5, 0.975))
quantile(y_rep1$BETA[,6], probs = c(0.025, 0.5, 0.975))

print_irt_stan(stanfit1)

# get CIs (should cut off thetas, though)
print(stanfit1, probs=c(0.025,.5,.975))

# l2pl
stanfit2 <- stan(file = "~/Documents/ST 559/l2pl_no_crossloadings2.stan",
                 data = complex_stan_CA,
                 chains = 2, #2
                 iter = 4000, # 4000
```

```
                warmup = 2000, # 2000
                # thin = 4,
                pars = params[[2]],
                control = list(adapt_delta = 0.9)
)

# convergence diagnostics
trace2 <- traceplot(stanfit2, pars = c('y_rep', 'theta', 'log_lik'),
                    include = FALSE)
rhat2 <- bayesplot::rhat(stanfit2, pars = params[[2]][1:(length(params[[2]])-3)])
scp2 <- edstan::stan_columns_plot(stanfit2, pars = params[[2]])
scp2 + theme_bw()

# EAP estimators for further analysis and summary tables
# why just third chain? probably because only had 2 chaings --> want all chains
post_means2 <- get_posterior_mean(stanfit2,
                                  pars = params[[2]][1:(length(params[[2]])-3)])[, 3, drop=FALSE]
# get_posterior_mean(stanfit, pars = params[[1]][1:(length(params[[1]])-3)])
write.csv(post_means2, "post_means2.csv")

# alpha4 low --> look at distributions of prof/notprof to see if there's a connection

# extract and save data
y_rep2 <- extract(stanfit2, pars = "log_lik", include = FALSE)
write.csv(y_rep2$theta, "y_rep2_theta.csv")

beta2 <- cbind(colMeans(y_rep2$BETA), apply(y_rep2$BETA, 2, sd))
write.csv(beta2, "beta2.csv")
alpha2 <- cbind(colMeans(y_rep2$ALPHA), apply(y_rep2$ALPHA, 2, sd))
write.csv(alpha2, "alpha2.csv")

log_lik[[2]] <- extract(stanfit2, pars = "log_lik")
```

```
y_rep1_theta <- read.csv("y_rep1_theta.csv")
hist(as.matrix(y_rep1_theta[,2:70]) - as.matrix(y_rep1_theta[,71:139]),
     main = expression(paste("Fall-spring differences in ", theta[i])),
     xlab = "Differences in individual abilities")
     # cex.main = 2, cex.lab = 1.5, cex.axis = 1)

prob1 <- mean(as.matrix(y_rep1_theta[,2:70]) - as.matrix(y_rep1_theta[,71:139]) > 0)

y_rep2_theta <- read.csv("y_rep2_theta.csv")

prob2 <- mean(as.matrix(y_rep2_theta[,2:70]) - as.matrix(y_rep2_theta[,71:139]) > 0)

hist(as.matrix(y_rep2_theta[,2:70]) - as.matrix(y_rep2_theta[,71:139]),
     main = expression(paste("Fall-spring differences in ", theta[i])),
     xlab = "Differences in individual abilities")
     #cex.main = 2, cex.lab = 1.5, cex.axis = 1)
```

```
# lrasch2.stan
# adapted from Scharl & Gnambs
// longitudinal Rasch model for 2 time points (NEPS sample)
```

```
functions {
    real corr_mat_pdf_log(matrix R, real k) {
        real log_dens;
        log_dens = ((k * (k - 1) / 2) - 1) * log_determinant(R) + (-((k + 1) / 2)) *
          sum(log(diagonal(R)));
        return log_dens;
    }
}
data {
    int<lower=1> I; // number of persons
    int<lower=1> T; // number of time points
    int<lower=1> J[T]; // number of items per time point
    int<lower=0, upper=1> Y[I, sum(J)]; // binary item response data
}
parameters {
    matrix[I, T] theta; // latent ability
    vector[sum(J)] beta; // item difficulty
    real mu; // prior mean of latent ability (time point 2)
    corr_matrix[T] R; // correlation matrix of latent ability
    real<lower=0> SD; // std. deviation of latent ability (time point 2)
}
transformed parameters {
    vector[T] mutheta;
    vector[T] S;
    cov_matrix[T] sigmatheta;
    vector[sum(J)] BETA; // item difficulty

    mutheta[1] = 0;
    mutheta[2] = mu;
    S[1] = 1;
    S[2] = SD;
    sigmatheta = diag_matrix(S) * R * diag_matrix(S);
    BETA = beta;
}
model {
    mu ~ normal(1, 3);
    R ~ corr_mat_pdf(T);
    SD ~ normal(1, 3) T[0, ];

    for (i in 1:I) {
        theta[i, ] ~ multi_normal(mutheta, sigmatheta);
    }
    beta ~ normal(0, 3);

    for (j in 1:(sum(J))) {
        if (j > J[1]) {
            Y[, j] ~ bernoulli_logit(theta[, 1] + theta[, 2] - BETA[j]);
        } else {
            Y[, j] ~ bernoulli_logit(theta[, 1] - BETA[j]);
        }
    }
}
generated quantities {
```

```
    int y_rep[I, sum(J)];
    real log_lik[I, sum(J)];

    for (i in 1:I) {
        for (j in 1:sum(J)) {
            if (j > J[1]) {
                y_rep[i, j] = bernoulli_logit_rng(theta[i, 1] + theta[i, 2] - BETA[j]);
            } else {
                y_rep[i, j] = bernoulli_logit_rng(theta[i, 1] - BETA[j]);}}}
    for (i in 1:I) {
        for (j in 1:sum(J)) {
            if (j > J[1]) {
                log_lik[i, j] = bernoulli_logit_lpmf(Y[i, j] | theta[i, 1] + theta[i, 2] - BETA[j]);
           } else {
               log_lik[i, j] = bernoulli_logit_lpmf(Y[i, j] | theta[i, 1] - BETA[j]);
           }}}}
```

```
# l2pl_no_crossloadings2.stan
# adapted from Scharl & Gnambs
// longitudinal 2PL model for 2 time points without cross-loadings (NEPS sample)
functions {
    real corr_mat_pdf_log(matrix R, real k) {
        real log_dens;
        log_dens = ((k * (k - 1) / 2) - 1) * log_determinant(R) +
          (-((k + 1) / 2)) * sum(log(diagonal(R)));
        return log_dens;
    }
}
data {
    int<lower=1> I; // number of persons
    int<lower=1> T; // number of time points
    int<lower=1> J[T]; // number of items per time point
    int<lower=0, upper=1> Y[I, sum(J)]; // binary item response data
}
parameters {
    matrix[I, T] theta; // latent ability
    vector<lower=0>[sum(J)] alpha; // item discrimination
    vector[sum(J)] beta; // item difficulty
    real mu; // prior mean of latent ability (time point 2)
    corr_matrix[T] R; // correlation matrix of latent ability
    real<lower=0> SD; // std. deviation of latent ability (time point 2)
}
transformed parameters {
    vector[T] mutheta;
    vector[T] S;
    cov_matrix[T] sigmatheta;
    vector[sum(J)] BETA; // item difficulty
    vector<lower=0>[sum(J)] ALPHA; // item discrimination

    mutheta[1] = 0;
    mutheta[2] = mu;
    S[1] = 1;
    S[2] = SD;
```

```
    sigmatheta = diag_matrix(S) * R * diag_matrix(S);
    ALPHA = alpha;
    BETA = beta;
}
model {
    mu ~ normal(1, 3);
    R ~ corr_mat_pdf(T);
    SD ~ normal(1, 3) T[0, ];

    for (i in 1:I) {
        theta[i, ] ~ multi_normal(mutheta, sigmatheta);
    }
    beta ~ normal(0, 3);
    for (j in 1:(sum(J))) alpha[j] ~ normal(1, 1) T[0, ];

    for (j in 1:sum(J)) {
        if (j > J[1]) {
            Y[, j] ~ bernoulli_logit(ALPHA[j] * theta[, 2] - BETA[j]);
        } else {
            Y[, j] ~ bernoulli_logit(ALPHA[j] * theta[, 1] - BETA[j]);
        }
    }
}
generated quantities {
    int y_rep[I, sum(J)];
    real log_lik[I, sum(J)];

    for (i in 1:I) {
        for (j in 1:sum(J)) {
            if (j > J[1]) {
                y_rep[i, j] = bernoulli_logit_rng(ALPHA[j] * theta[i, 2] - BETA[j]);
            } else {
                y_rep[i, j] = bernoulli_logit_rng(ALPHA[j] * theta[i, 1] - BETA[j]);}}}
    for (i in 1:I) {
        for (j in 1:sum(J)) {
            if (j > J[1]) {
                log_lik[i, j] = bernoulli_logit_lpmf(Y[i, j] | ALPHA[j] * theta[i, 2] - BETA[j]);
           } else {
               log_lik[i, j] = bernoulli_logit_lpmf(Y[i, j] | ALPHA[j] * theta[i, 1] - BETA[j]);
           }}}}
```