

2PL Item Response Theory

Load the Packages

```
library(rstan)
```

Model

$$P_{kj} = \frac{\gamma_k * (e^{\alpha_j - \beta_k + \delta})}{1 + \gamma_k * (e^{\alpha_j - \beta_k + \delta})}$$

P_{kj} : probability of success by person j on item k

α_j : ability of person j

β_k : difficulty of item k

γ_k : discrimination parameter of item k

δ : mean person ability

Data

ltm library 안에 있는 Mobility 데이터 사용.

```
##data
set.seed(123)
library(ltm)
response <- Mobility[sample(1:nrow(Mobility), 20, replace = FALSE),]
head(response)

##list data
J = 20
K = 8

data1 <- list(
  J = J,
  K = K,
  N = J*K,
  jj = rep(1:J, times = K),
  kk = rep(1:K, each = J),
  y = as.numeric(unlist(response))
)
```

STAN Code

```

data {
  int<lower=1> J; // number of students
  int<lower=1> K; // number of questions
  int<lower=1> N; // number of observations
  int<lower=1, upper=J> jj[N]; // student for observation n
  int<lower=1, upper=K> kk[N]; // question for observation n
  int<lower=0, upper=1> y[N]; // correctness for observation n
}

parameters {
  real mu_beta; // mean question difficulty
  vector[J] alpha; // ability for student j - mean ability
  vector[K] beta; // difficulty for k-th item
  vector<lower=0>[K] gamma; // discrimination of k
  real<lower=0> sigma_beta; // scale of difficulties
  real<lower=0> sigma_gamma; // scale of log discrimination
}

model {
  alpha ~ std_normal(); // prior of alpha(ability)
  beta ~ normal(0, sigma_beta); // prior of beta(difficulty)
  gamma ~ lognormal(0, sigma_gamma); // prior of discrimination
  mu_beta ~ cauchy(0,5); // prior of mean question difficulty
  sigma_beta ~ cauchy(0,5); // prior of scale of difficulties
  sigma_gamma ~ cauchy(0,5); // prior of scale of log discrimination
  y ~ bernoulli_logit(gamma[kk] .* (alpha[jj] - (beta[kk] + mu_beta))); //
  likelihood
}

```

Fitting

```

##fitting
fit_2pl <- stan(file= '2pl_IRT.stan', data = data1)
fit_2pl

```

Trace Plot

```

traceplot(fit_2pl, pars = 'sigma_beta', inc_warmup = TRUE)
traceplot(fit_2pl, pars = 'sigma_gamma', inc_warmup = TRUE)
traceplot(fit_2pl, pars = 'mu_beta', inc_warmup = TRUE)

```

Estimated Parameters

```

plot(fit_2pl, pars = 'mu_beta', show_density=TRUE)
plot(fit_2pl, pars = 'sigma_beta', show_density=TRUE)
plot(fit_2pl, pars = 'sigma_gamma', show_density=TRUE)

```

