

Chapter 1

R code for estimating Rasch model and log-normal estimates from IAT (SC-IAT) data.

The Rasch and log-normal estimates were obtained by means of `lme4` package (J, J) in R. The R code used for estimating the models and for extracting the parameters estimates is illustrated in this Appendix. This code can be copied and pasted in an R script, and it can be executed without changes as long as the data set on which the models are applied has the following characteristics:

- `subject`: Column containing the respondents IDs (can be numeric, a factor, or a string, as long as it is unique for each respondent).
- `condition`: Column containing the labels for the two associative conditions of the

IAT (factor with two levels such as `mappingA` and `mappingB`).

- `stimuli`: column containing the labels identifying each stimulus (e.g., `good`, `bad`, `DarkChocolate1`, `MilkChocolate2`).
- `latency`: Column containing the latency of the IAT responses. Latency can be expressed in seconds or milliseconds (in both studies of this paper, we used seconds). In case the IAT included a built-in correction for the error responses, the raw response times should be used instead of the corrected ones.
- `correct`: Column containing the accuracy of the IAT responses, where 0 is the incorrect response and 1 is the correct response.

The data set must be in a long format. This means that the response of each respondent on each stimulus in each associative condition must be on a separate row, and the total number of observations (and rows) for each each subject must correspond to the total number of critical trials in the two associative conditions. For instance, in the two studies reported in this paper, participants were presented with 60 trials in each associative condition, so that we had 120 trials for each respondent, and consequently 120 rows for each participant.

In both accuracy and log-time responses, the fixed intercept was set at 0, so that the estimates for the effect of the IAT associative conditions can be interpreted as the expected log-odds of the probability of a correct response in each condition or the expected average log-response time in each condition, respectively.

For both accuracy and log-time responses, in Model 1 (Table ??) the estimates of the stimuli are centered at 0 (argument `(1 | stimuli)`), while in Model 2 (Table ??) respondents

estimates are centered at 0 (argument `(1|subject)`).

Accuracy models specification

The code for the specification of the accuracy models is illustrated. The name of the data set in the argument `data` has to be changed accordingly.

Model 1: Between–stimuli variability specified as random intercepts (i.e., `(1|stimuli)`). Within–subjects and between–conditions variability specified as the random slopes of the respondents in the conditions (i.e., `(0 + condition|subject)`).

```
library(lme4) # upload the package for the estimation of the models

a1 <- glmer(correct ~ 0 + condition + (1|stimuli) +
            (0 + condition|subject), data = your_data,
            family = "binomial")

summary(a1) # summary of the results
```

Model 2: Between–subjects variability specified as random intercepts (i.e., `(1|subject)`). Within–stimuli and between–conditions variability specified as the random slopes of the stimuli in the conditions (i.e., `(0 + condition|stimuli)`).

```
a2 <- glmer(correct ~ 0 + condition + (1|subject) +
            (0 + condition|stimuli), data = your_data,
            family = "binomial")

summary(a2) # summary of the results
```

Model 3: Between–subjects variability specified as random intercepts (i.e., `(1|subjects)`). Between–stimuli variability specified as random intercepts (i.e., `(1|stimuli)`).

```

a3 <- glmer(correct ~ 0 + condition + (1|stimuli) + (1|subject),
data = your_data,

family = "binomial")

summary(a3) # summary of the results

```

Once the three models have been estimated, they can be compared with each other. Model 1 (a1) and Model 2 (a2) have the same degrees of freedom:

```
anova(a1, a2, a3)
```

Accuracy models: Rasch model parameters Grounding on the results of the model comparison, the best fitting model can be selected for extracting the Rasch model parameters estimates.

Model 1 results in condition-specific respondents parameters and overall stimuli parameters. Respondents' condition-specific ability parameters can be extracted as follows:

```

cond_ability<- coef(a1)$subject[, -1] # drop the first column
# (fixed intercepts set at 0)
# rownames are the subjects' IDs

```

Stimuli easiness parameters can be extracted and stored in a data frame as well:

```

easiness <- data.frame(

stimuli = rownames(coef(a1)$stimuli),

easiness = coef(a1)$stimuli[, 1] # select only the

# random estimates intercept

)

```

Model 2 results in condition-specific stimuli parameters and overall respondents' parameters. Stimuli condition-specific parameters can be extracted as follows:

```
easiness_cond <- coef(a2)$stimuli[, -1] # drop the first column
# (fixed intercept set at 0)
# rownames are stimuli labels
```

Respondents overall ability parameters can be extracted and stored in a dataframe:

```
ability <- data.frame(
  subject = rownames(coef(a2)$subject),
  ability = coef(a2)$subject[, 1] # select only the random
  # intercept estimates
)
```

Model 3 results in overall respondents' parameters and overall stimuli parameters. Respondents overall ability parameters can be extracted and stored in a data frame:

```
ability <- data.frame(
  subject = rownames(coef(a3)$subject),
  ability = coef(a3)$subject[, -1]
)
```

Stimuli overall easiness parameters can be extracted and stored as well:

```
easiness <- data.frame(
  stimuli = rownames(coef(a3)$stimuli),
  easiness = coef(a3)$stimuli[, -1]
)
```

Log-time models specification

The code for the estimation of the log-normal models is the same as the one used for the Rasch models. The changes concern the name of the specific function to use (from `glmer()` to `lmer()`) and the dependent variable (from `correct` to `log(latency)`). For this reason, we report the code for the estimation of only Model 1.

```
t1 <- lmer(log(seconds) ~ 0 + condition + (1|stimuli) +
  (0 + condition|subject),
  data = your_data,
  REML = FALSE) # Maximum Likelihood estimation

summary(t1) # summary of the results
```

For log-time models comparison, the same code as the one for accuracy models comparison can be used by changing the names of the models from `a` to `t`.

Log-time models: Log-normal model parameters We report the code for extracting the log-normal model parameters for log-time Model 1, assuming it was the best fitting model according to models comparison. The same code used for extracting the parameters for the accuracy models can be used for extracting the parameters of the log-normal models. The changes regard the name of the objects containing the models, from `a` to `t`, and the names of the new objects created for the parameters (e.g., from `easiness` to `intensity`).

Respondents' condition-specific parameters:

```
cond_speed <- coef(a1)$subject[, -1] # drop the first column
# (fixed intercepts set at 0)
# rownames are the subjects' IDs
```

Stimuli overall time intensity parameters:

```
intensity <- data.frame(  
  stimuli = rownames(coef(t1)$stimuli),  
  easiness = coef(t1)$stimuli[, 1] # select only the  
  # random intercept estimates  
)
```