PSYC575 Course Project

Qianhui (Vicky) Ni

2020-11-18

1. Load packages

```
library(tidyverse)
library(psych)
library(lme4)
library(broom.mixed)
library(brms)
library(modelsummary)
library(haven)  # for importing SPSS/SAS/Stata data
library(lmerTest)  # for testing coefficients
library(MuMIn)  # for R^2
library(lattice)  # for dotplot (working with lme4)
library(ggplot2)
library(sjPlot)  # for plotting effects
library(psych)
library(emmeans)
library(tidybayes)
# Add the following so that the LOO will be included in the msummary table
glance_custom.brmsfit <- function(x) {
  broom.mixed::glance(x, looic = TRUE)
}
theme_set(theme_bw())  # Theme; just my personal preference
```

2. Import Data

```
setwd("/Users/qianhuini/Desktop/USC/Study/2020 Fall/575_Multilevel Modeling/575_project")
ToM_dat <- read.csv("ELdataset.csv")
head(ToM_dat)
```

```
##   Subject RT_ID_AD_p RT_ID_AU_p RT_ID_AD_a RT_ID_AU_a RT_LO_AD_p RT_LO_AU_p
## 1       1       1208       1197       2105        933       1554       3255
## 2       1       1017       1160       1063       1541       1192       1265
## 3       1       1050       1243       1076       1236       1044       1324
## 4       1        914       1024       1159        904       1319       1454
## 5       1        924       1133        821        890       1586       1328
## 6       1       1539       1615       1096       1492       1073       1677
##   RT_LO_AD_a RT_LO_AU_a ERR_ID_AD_p ERR_ID_AU_p ERR_ID_AD_a ERR_ID_AU_a
## 1       1909       1389           1           1           1           1
## 2       2205       1810           1           1           1           1
## 3       1134       1240           1           1           1           1
## 4       3375        979           1           1           1           1
## 5       1076        996           1           1           1           1
```

```
## 6       1366       1044           1           1           1           0
##   ERR_LO_AD_p ERR_LO_AU_p ERR_LO_AD_a ERR_LO_AU_a
## 1           0           1           1           1
## 2           1           1           1           1
## 3           1           1           1           1
## 4           1           1           1           1
## 5           1           1           1           1
## 6           1           1           1           1
```

For data storage needs, the name of each column is slightly modified. The "+" and "-" in the original dataset are changed to "p" for presence and "a ' for absence.

3. Descriptive statistics

```
ds1 <- describe(ToM_dat)
ds1
```

```
##             vars   n    mean      sd median trimmed    mad min   max range
## Subject        1 400   20.50   11.56   20.5   20.50  14.83   1    40    39
## RT_ID_AD_p     2 400 1628.57 1465.20 1306.0 1411.72 467.76 520 21609 21089
## RT_ID_AU_p     3 400 1327.99 1026.97 1076.0 1144.92 382.51 503 11002 10499
## RT_ID_AD_a     4 400 1857.20 1955.10 1441.0 1583.37 634.55 754 29117 28363
## RT_ID_AU_a     5 400 1762.41 1411.78 1434.5 1547.01 606.38 690 17651 16961
## RT_LO_AD_p     6 400 1264.81 1223.46  981.0 1078.58 404.75 424 19892 19468
## RT_LO_AU_p     7 400 1682.86 2124.93 1353.0 1474.86 631.59 506 39483 38977
## RT_LO_AD_a     8 400 1698.23 1050.51 1422.0 1554.30 647.90 554 13687 13133
## RT_LO_AU_a     9 400 1533.20  914.10 1318.0 1413.49 475.91 603 13335 12732
## ERR_ID_AD_p   10 399    0.92    0.28    1.0    1.00   0.00   0     1     1
## ERR_ID_AU_p   11 400    0.92    0.26    1.0    1.00   0.00   0     1     1
## ERR_ID_AD_a   12 400    0.89    0.31    1.0    0.99   0.00   0     1     1
## ERR_ID_AU_a   13 400    0.87    0.34    1.0    0.96   0.00   0     1     1
## ERR_LO_AD_p   14 400    0.96    0.21    1.0    1.00   0.00   0     1     1
## ERR_LO_AU_p   15 400    0.94    0.23    1.0    1.00   0.00   0     1     1
## ERR_LO_AD_a   16 400    0.86    0.34    1.0    0.95   0.00   0     1     1
## ERR_LO_AU_a   17 400    0.95    0.21    1.0    1.00   0.00   0     1     1
##              skew kurtosis     se
## Subject      0.00    -1.21   0.58
## RT_ID_AD_p   8.40    95.72  73.26
## RT_ID_AU_p   5.65    42.06  51.35
## RT_ID_AD_a   9.09   107.86  97.75
## RT_ID_AU_a   6.95    65.56  70.59
## RT_LO_AD_p  10.05   139.05  61.17
## RT_LO_AU_p  14.70   251.12 106.25
## RT_LO_AD_a   4.97    45.66  52.53
## RT_LO_AU_a   6.80    74.73  45.70
## ERR_ID_AD_p -3.02     7.13   0.01
## ERR_ID_AU_p -3.22     8.36   0.01
## ERR_ID_AD_a -2.48     4.18   0.02
## ERR_ID_AU_a -2.19     2.81   0.02
## ERR_LO_AD_p -4.37    17.17   0.01
## ERR_LO_AU_p -3.89    13.16   0.01
## ERR_LO_AD_a -2.10     2.41   0.02
## ERR_LO_AU_a -4.24    16.01   0.01
```

4. Data Preprocessing

The original dataset is in wide-format and with `rt` and `err` together, so I break it into two subsets and then transform them to long-format. Log transformation is also performed for the variable `rt`. There are two predictions: `Task` and `Condition`, so I will separate them too during the transformation. Each participant experienced all 8 situations, and these situations differ in their task type and condition type.

```
# Seperate into two subsets
ToM_rt_wide <- ToM_dat[,1:9]
ToM_err_wide <- ToM_dat[,c(1,10:17)]
# Covert to long-format dataset
ToM_rt <- ToM_rt_wide %>%
  pivot_longer(
    cols = RT_ID_AD_p:RT_LO_AU_a,
    names_to = "Condition",
    names_prefix = "rt",
    values_to = "rt",)
ToM_rt$Task <- substring(ToM_rt$Condition, 4,5)
ToM_rt$Condition <- sub('^......','',ToM_rt$Condition)

ToM_err <- ToM_err_wide %>%
  pivot_longer(
    cols = ERR_ID_AD_p:ERR_LO_AU_a,
    names_to = "Condition",
    names_prefix = "err",
    values_to = "err",)
ToM_err$Task <- substring(ToM_err$Condition, 5,6)
ToM_err$Condition <- sub('^.......','',ToM_err$Condition)

# Log transformation for rt
ToM_rt$lg_rt <- log(ToM_rt$rt)
ToM_rt <- ToM_rt[,c(1,4,2,3,5)]
head(ToM_rt)
```

```
## # A tibble: 6 x 5
##   Subject Task  Condition    rt lg_rt
##     <int> <chr> <chr>     <int> <dbl>
## 1       1 ID    AD_p       1208  7.10
## 2       1 ID    AU_p       1197  7.09
## 3       1 ID    AD_a       2105  7.65
## 4       1 ID    AU_a        933  6.84
## 5       1 LO    AD_p       1554  7.35
## 6       1 LO    AU_p       3255  8.09
```

```
# Add one colume about situation
ToM_rt$Situation <- paste(ToM_rt$Task, ToM_rt$Condition)
ToM_rt$Situation[ToM_rt$Situation=="ID AD_p"] <- "1"
ToM_rt$Situation[ToM_rt$Situation=="ID AU_p"] <- "2"
ToM_rt$Situation[ToM_rt$Situation=="ID AD_a"] <- "3"
ToM_rt$Situation[ToM_rt$Situation=="ID AU_a"] <- "4"
ToM_rt$Situation[ToM_rt$Situation=="LO AD_p"] <- "5"
ToM_rt$Situation[ToM_rt$Situation=="LO AU_p"] <- "6"
ToM_rt$Situation[ToM_rt$Situation=="LO AD_a"] <- "7"
ToM_rt$Situation[ToM_rt$Situation=="LO AU_a"] <- "8"

ToM_acc <- ToM_err
```

```r
ToM_acc$Situation <- paste(ToM_acc$Task, ToM_acc$Condition)
ToM_acc$Situation[ToM_acc$Situation=="ID AD_p"] <- "1"
ToM_acc$Situation[ToM_acc$Situation=="ID AU_p"] <- "2"
ToM_acc$Situation[ToM_acc$Situation=="ID AD_a"] <- "3"
ToM_acc$Situation[ToM_acc$Situation=="ID AU_a"] <- "4"
ToM_acc$Situation[ToM_acc$Situation=="LO AD_p"] <- "5"
ToM_acc$Situation[ToM_acc$Situation=="LO AU_p"] <- "6"
ToM_acc$Situation[ToM_acc$Situation=="LO AD_a"] <- "7"
ToM_acc$Situation[ToM_acc$Situation=="LO AU_a"] <- "8"
```

4.1 Check speed-accuracy tradeoffs

```r
cor.test(ToM_rt$rt, ToM_err$err)
```

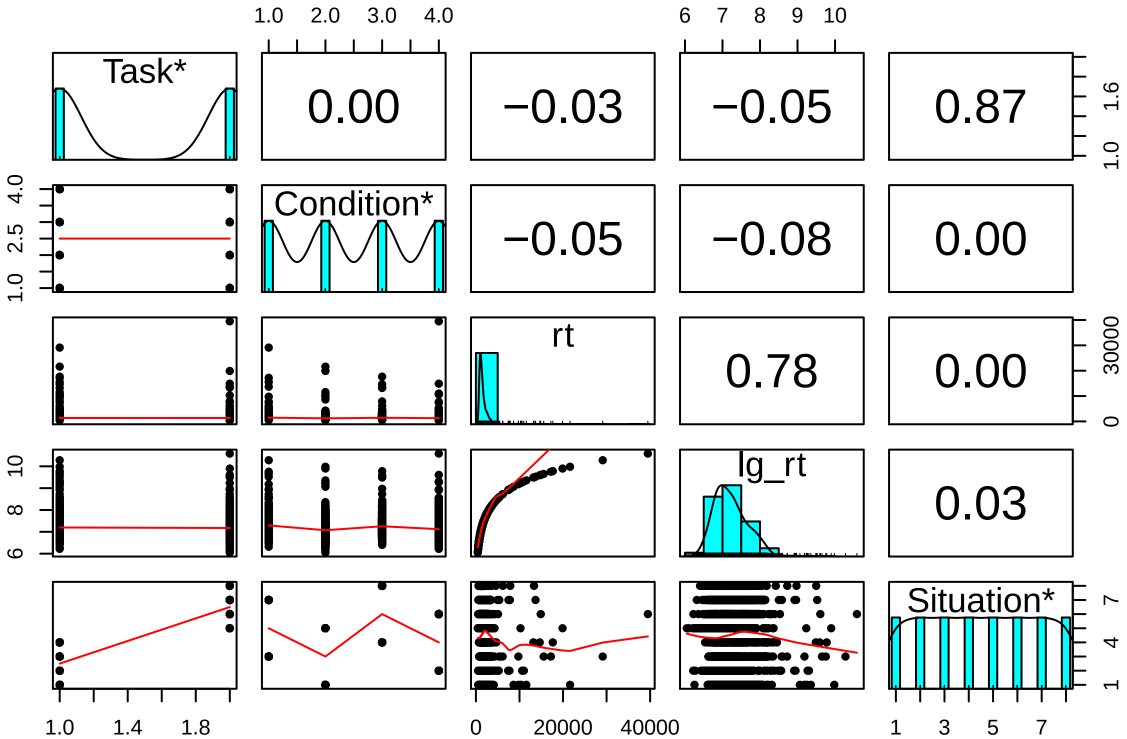
```
##
##  Pearson's product-moment correlation
##
## data:  ToM_rt$rt and ToM_err$err
## t = 1.2496, df = 3197, p-value = 0.2115
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.01256934  0.05670763
## sample estimates:
##        cor
## 0.02209567
```

5. Analysis of Reaction Time

There are two outcome variables: `rt` and `err`. They will be analyzed separately.

5.1 Data Exploration

```r
pairs.panels(ToM_rt[ , -1],
             ellipses = FALSE)
```



5.2 Unconditional model with random intercepts

I first run an unconditional model with random intercepts of `Situation` and `Condition`.

Repeated measure (within-cell) level (Lv1):

$$\text{lg rt}_{i(j,k)} = \beta_{0(j,k)} + e_{ijk}$$

Between-cell leve l(Lv2):

$$\beta_{0(j,k)} = \gamma_{00} + u_{0j} + v_{0k}$$

```
m0_rt <- lmer(lg_rt ~ (1 | Subject) + (1 | Situation), data = ToM_rt)
vc_m0_rt <- as.data.frame(VarCorr(m0_rt))

# Proportion of variance at the within-cell level
icc_e_rt <- vc_m0_rt$vcov[3] / sum(vc_m0_rt$vcov)

# ICC/Deff (Subject; cluster size = 8)
icc_subj_rt <- vc_m0_rt$vcov[1] / sum(vc_m0_rt$vcov)
c("ICC(subj_rt)" = icc_subj_rt,
  "Deff(subj_rt)" = 1 + ((8-1) * icc_subj_rt))
```

```
##  ICC(subj_rt) Deff(subj_rt)
##     0.2337834     2.6364836
```

```
# ICC/Deff (Situation; cluster size = 40)
icc_situation_rt <- vc_m0_rt$vcov[2] / sum(vc_m0_rt$vcov)
c("ICC(Situation_rt)" = icc_situation_rt, "Deff(Situation_rt)" = 1 + (40-1) * icc_situation_rt)
```

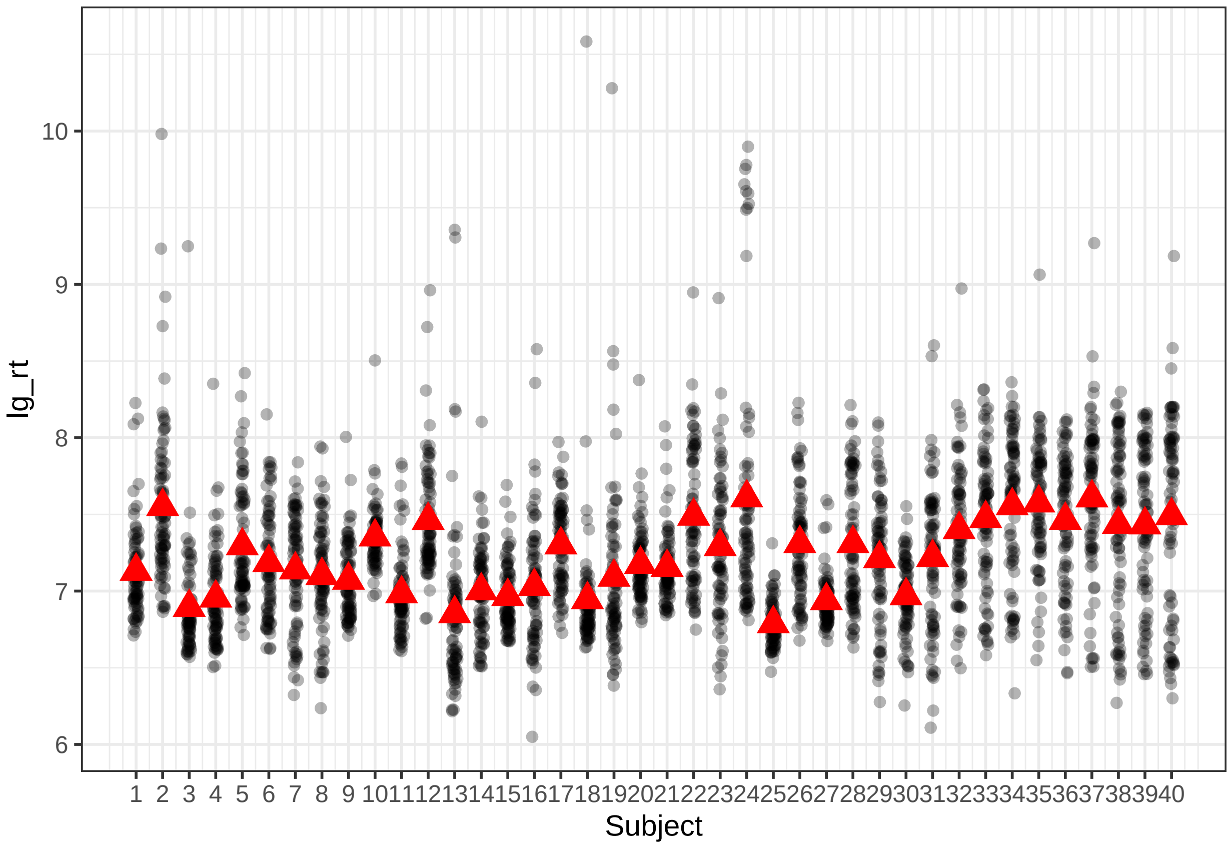
```
##  ICC(Situation_rt) Deff(Situation_rt)
##         0.07710844         4.00722920
```

```r
c("ICC(Subject_rt + Situation_rt)" = sum(vc_m0_rt$vcov[1:2]) / sum(vc_m0_rt$vcov))
```

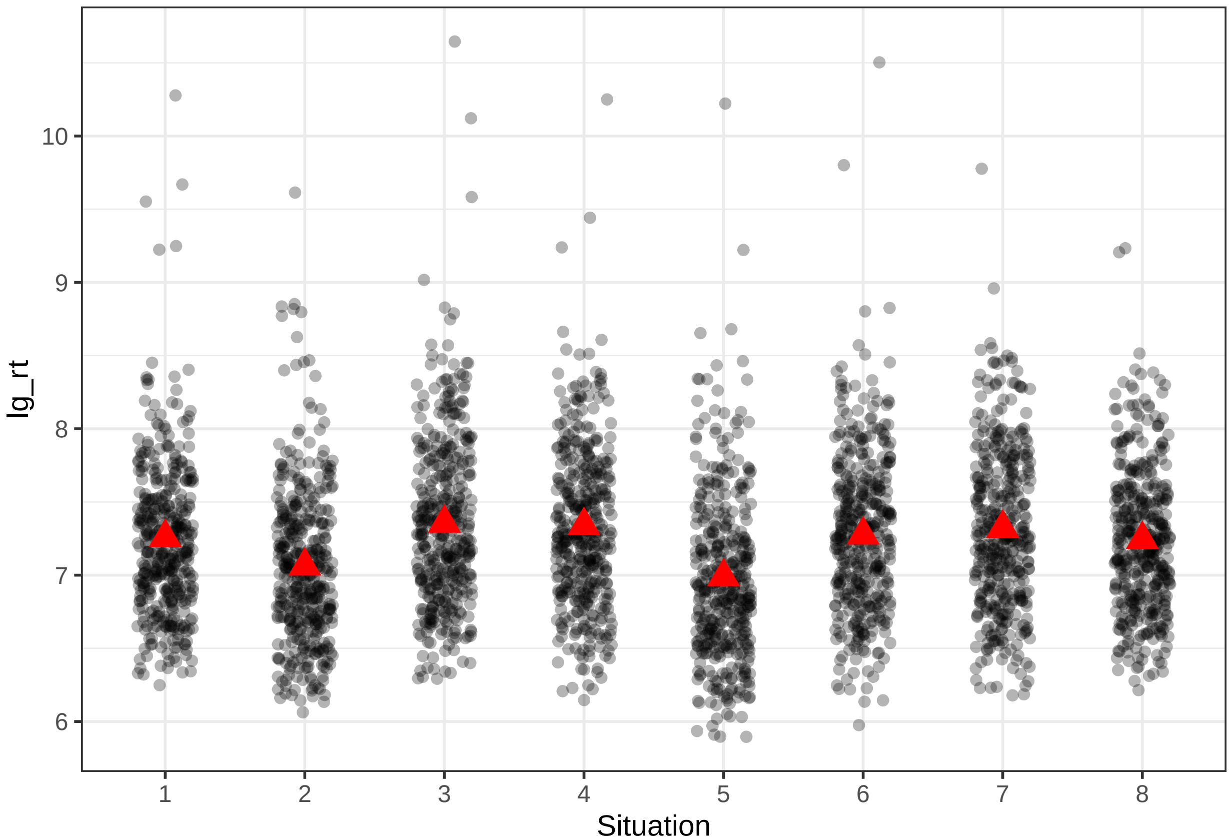
```
## ICC(Subject_rt + Situation_rt)
##                      0.3108918
```

The results show that the ICC of `Subject` is .234 with the design effect of 2.636. This means that we can expect a correlation between two randomly drawn units from the same subject. This is quite reasonable because the same person will have a certain reaction pattern and a range of reaction times. The design effect is large, so it's necessary to use multilevel modeling. Next, the ICC of `Situation` is .077 with the design effect of 4.007. In addition, the ICC with subject and situation together is 0.311 so if we have the same participant experiencing the same situation multiple times, the responses will also be correlated.
The variations across subjects, and across situations are plotted as below.

```r
# Variation across persons
sub_ids <- unique(ToM_rt$Subject)
(p_set <- ToM_rt %>%
    filter(Subject %in% sub_ids) %>%
    ggplot(aes(x = Subject, y = lg_rt)) +
    geom_jitter(height = 0, width = 0.1, alpha = 0.3) +
    scale_x_continuous(breaks = sub_ids, labels = sub_ids) +
    # Add subject means
    stat_summary(
      fun = "mean",
      geom = "point",
      col = "red",
      shape = 17,
      # use triangles
      size = 4)
)
```




```
# Variation across persons
sub_si <- unique(ToM_rt$Situation)
(p_set <- ToM_rt %>%
    filter(Situation %in% sub_si) %>%
    ggplot(aes(x = Situation, y = lg_rt)) +
    geom_jitter(height = 0.5, width = 0.2, alpha = 0.3) +
    scale_x_discrete(breaks = sub_si, labels = sub_si) +
    # Add subject means
    stat_summary(
      fun = "mean",
      geom = "point",
      col = "red",
      shape = 17,
      # use triangles
      size = 4)
)
```



5.3 Recode the levels of each predictor

Because I am mainly interested in the interaction of LO & ID and AD+ & AU+, I now make Identity task the reference group by making the variable a factor with Identity task as the first category. For `Condition`, AD+ is the the first category.

```
ToM_rt <- ToM_rt %>%
  mutate(Task = factor(Task, levels = c("ID", "LO")))
ToM_rt <- ToM_rt %>%
  mutate(Condition = factor(Condition, levels = c("AD_p", "AU_p","AD_a","AU_a")))
```

5.4 Modeling for `rt`

5.4.1 Model Equations

Repeated-Measure level (Lv 1):

$$\text{lg rt}_{i(j,k)} = \beta_{0(j,k)} + e_{ijk}$$

Between-cell (Subject x Situation) level:

$$\beta_{0(j,k)} = \gamma_{00} + \beta_{1j}\text{Task}_{ik} + \beta_{2j}\text{Condition}_{ik} + \beta_{3j}\text{Task}_{ik} \times \text{Condition}_{ik} + u_{0j} + v_{0k}$$

Subject level:

$$\beta_{1j} = \gamma_{10} + u_{1j}\beta_{2j} = \gamma_{20} + u_{2j}\beta_{3j} = \gamma_{30} + u_{3j}$$

Combined equations

$$
\begin{aligned}
\lg\ \mathrm{rt}_{i(j,k)} = \gamma_{00} \\
& + \gamma_{10}\mathrm{Task}_{ik} + \gamma_{20}\mathrm{Condition}_{ik} + \gamma_{30}\mathrm{Condition}_{ik} \times \mathrm{Task}_{ik} + \\
& + u_{0j} + u_{1j}\mathrm{Task}_{ik} + u_{2j}\mathrm{Condition}_{ik} + u_{3j}\mathrm{Task}_{ik} \times \mathrm{Condition}_{ik} \\
& + v_{0k} + e_{ijk}
\end{aligned}
$$

5.4.2 Fit a Model

To make sure that random slops are necessary, I tested them one by one.

```
# First, no random slops
m_test_no <- lmer(lg_rt ~ Condition * Task + (1 | Subject) + (1 | Situation), data = ToM_rt)
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -4.5e-06
```

```
# Then test random slopes one by one
# Random slopes of Task-Condition interaction across subjects
m_test_1 <- lmer(lg_rt ~ Condition*Task + (Condition:Task | Subject) + (1 | Situation), data = ToM_rt)
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Warning: Model failed to converge with 7 negative eigenvalues: -4.7e-05 -2.0e-04
## -5.6e-04 -4.3e-03 -9.9e-03 -1.4e-01 -1.6e+02
```

```
ranova(m_test_1)
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -4.5e-06
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Warning: Model failed to converge with 4 negative eigenvalues: -4.0e-05 -2.0e-03
## -2.5e-03 -3.2e-03
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## lg_rt ~ Condition + Task + (Condition:Task | Subject) + (1 |
##     Situation) + Condition:Task
##                                              npar  logLik    AIC    LRT Df
## <none>                                         55 -1296.6 2703.2
## Condition:Task in (Condition:Task | Subject)   11 -1656.5 3335.0 719.79 44
## (1 | Situation)                                54 -1296.6 2701.2   0.00  1
##                                              Pr(>Chisq)
## <none>
## Condition:Task in (Condition:Task | Subject)     <2e-16 ***
## (1 | Situation)                                       1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Random slopes of Task (situation-level) across subjects
m_test_2 <- lmer(lg_rt ~ Condition*Task + (Task | Subject) + (1 | Situation), data = ToM_rt)
```

```
## 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
```

```r
ranova(m_test_2)
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -4.5e-06
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## lg_rt ~ Condition + Task + (Task | Subject) + (1 | Situation) +
##     Condition:Task
##                          npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                     13 -1531.0 3088.1
## Task in (Task | Subject)   11 -1656.5 3335.0 250.87  2     <2e-16 ***
## (1 | Situation)            12 -1531.0 3086.1   0.00  1     0.9999
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Random slopes of Condition (situation-level) across subjects
m_test_3 <- lmer(lg_rt ~ Condition*Task + (Condition | Subject) + (1 | Situation), data = ToM_rt)
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Warning: Model failed to converge with 2 negative eigenvalues: -1.8e-06 -2.2e-01
```

```r
ranova(m_test_3)
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -4.5e-06
```

```
## boundary (singular) fit: see ?isSingular
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -9.4e-01
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## lg_rt ~ Condition + Task + (Condition | Subject) + (1 | Situation) +
##     Condition:Task
##                                    npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                               20 -1616.0 3271.9
## Condition in (Condition | Subject)   11 -1656.5 3335.0 81.042  9  1.003e-13 ***
## (1 | Situation)                      19 -1616.0 3269.9  0.000  1          1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Judgement:
The random slopes of of Task-Condition interaction across subjects, of Task across subjects, and of COndition across subjects are all significant. So they will be included in the final model.

Here I fit a Bayesian multilevel model to estimate the effect of `Task` on ‘`rt`.
The multilevel models were fitted using the brms package (Bürkner, 2017) in R, which performs Markdov Chain Monte Carlo approximation with the No U-Turn Sampler to approximate the posterior distributions of the model parameters. For each model, 4 chains are used, each with 2,000 iterations (1,000 warmup). The default priors from brms were used, which include uniform non-informative priors on the fixed-effect parameters and weakly informative Student-t priors on the standard deviations of the random effects. For all model, Rhat < 1.01 (Vehtari et al., 2020), indicating convergence of the chains to a stationary posterior distributions. The posterior distributions of the model parameters are summarized using the posterior means and the 95% equal-tailed credible intervals.
Interaction between these two predictors and varing slopes are also included. Because of counterbalancing, there is no need for cluster-mean centering.

```r
m1_rt <- brm(lg_rt ~ Task + Condition + Task * Condition +
               (Task + Condition + Task:Condition | Subject),
             data = ToM_rt,
             control = list(adapt_delta = .9),
             cores = 2)
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```r
summary(m1_rt)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: lg_rt ~ Task + Condition + Task * Condition + (Task + Condition + Task:Condition | Subject)
##    Data: ToM_rt (Number of observations: 3200)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~Subject (Number of levels: 40)
##                                                 Estimate Est.Error l-95% CI
## sd(Intercept)                                       0.28      0.03     0.22
## sd(TaskLO)                                          0.42      0.05     0.33
## sd(ConditionAU_p)                                   0.32      0.04     0.25
## sd(ConditionAD_a)                                   0.20      0.03     0.14
## sd(ConditionAU_a)                                   0.20      0.03     0.14
## sd(TaskLO:ConditionAU_p)                            0.56      0.06     0.45
## sd(TaskLO:ConditionAD_a)                            0.33      0.04     0.25
## sd(TaskLO:ConditionAU_a)                            0.31      0.04     0.23
## cor(Intercept,TaskLO)                              -0.48      0.12    -0.69
## cor(Intercept,ConditionAU_p)                       -0.44      0.13    -0.67
## cor(TaskLO,ConditionAU_p)                           0.39      0.14     0.09
## cor(Intercept,ConditionAD_a)                        0.12      0.17    -0.21
## cor(TaskLO,ConditionAD_a)                           0.05      0.17    -0.30
## cor(ConditionAU_p,ConditionAD_a)                    0.09      0.17    -0.25
## cor(Intercept,ConditionAU_a)                       -0.01      0.17    -0.35
## cor(TaskLO,ConditionAU_a)                          -0.02      0.17    -0.35
## cor(ConditionAU_p,ConditionAU_a)                    0.07      0.17    -0.27
## cor(ConditionAD_a,ConditionAU_a)                    0.82      0.09     0.60
## cor(Intercept,TaskLO:ConditionAU_p)                 0.42      0.13     0.14
## cor(TaskLO,TaskLO:ConditionAU_p)                   -0.60      0.10    -0.77
## cor(ConditionAU_p,TaskLO:ConditionAU_p)            -0.83      0.06    -0.92
## cor(ConditionAD_a,TaskLO:ConditionAU_p)             0.15      0.16    -0.17
## cor(ConditionAU_a,TaskLO:ConditionAU_p)             0.14      0.16    -0.19
## cor(Intercept,TaskLO:ConditionAD_a)                 0.32      0.15    -0.01
## cor(TaskLO,TaskLO:ConditionAD_a)                   -0.65      0.11    -0.82
## cor(ConditionAU_p,TaskLO:ConditionAD_a)            -0.55      0.13    -0.78
## cor(ConditionAD_a,TaskLO:ConditionAD_a)            -0.46      0.14    -0.70
## cor(ConditionAU_a,TaskLO:ConditionAD_a)            -0.42      0.16    -0.70
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAD_a)      0.65      0.10     0.43
## cor(Intercept,TaskLO:ConditionAU_a)                 0.34      0.15     0.02
## cor(TaskLO,TaskLO:ConditionAU_a)                   -0.64      0.11    -0.81
## cor(ConditionAU_p,TaskLO:ConditionAU_a)            -0.46      0.15    -0.72
## cor(ConditionAD_a,TaskLO:ConditionAU_a)            -0.28      0.17    -0.59
```

```
## cor(ConditionAU_a,TaskLO:ConditionAU_a)          -0.43      0.15    -0.68
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAU_a)     0.64      0.11     0.39
## cor(TaskLO:ConditionAD_a,TaskLO:ConditionAU_a)     0.88      0.07     0.71
##                                                 u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)                                       0.36 1.00     1642     2384
## sd(TaskLO)                                          0.52 1.00     1224     2228
## sd(ConditionAU_p)                                   0.41 1.00     1119     2024
## sd(ConditionAD_a)                                   0.27 1.00     1820     2562
## sd(ConditionAU_a)                                   0.27 1.00     1720     2719
## sd(TaskLO:ConditionAU_p)                            0.69 1.00     1073     1950
## sd(TaskLO:ConditionAD_a)                            0.42 1.00     1487     2653
## sd(TaskLO:ConditionAU_a)                            0.40 1.00     1409     2234
## cor(Intercept,TaskLO)                              -0.22 1.00     1204     1911
## cor(Intercept,ConditionAU_p)                       -0.15 1.00     1497     2370
## cor(TaskLO,ConditionAU_p)                           0.64 1.00     1383     2085
## cor(Intercept,ConditionAD_a)                        0.44 1.00     2081     2679
## cor(TaskLO,ConditionAD_a)                           0.35 1.00     2172     2503
## cor(ConditionAU_p,ConditionAD_a)                    0.42 1.00     2005     2618
## cor(Intercept,ConditionAU_a)                        0.33 1.00     2260     2705
## cor(TaskLO,ConditionAU_a)                           0.31 1.00     2032     2613
## cor(ConditionAU_p,ConditionAU_a)                    0.39 1.00     2022     3084
## cor(ConditionAD_a,ConditionAU_a)                    0.95 1.00     1982     2859
## cor(Intercept,TaskLO:ConditionAU_p)                 0.65 1.00     1446     2110
## cor(TaskLO,TaskLO:ConditionAU_p)                   -0.37 1.00     1542     2148
## cor(ConditionAU_p,TaskLO:ConditionAU_p)            -0.69 1.00     1557     2520
## cor(ConditionAD_a,TaskLO:ConditionAU_p)             0.45 1.00     2122     2770
## cor(ConditionAU_a,TaskLO:ConditionAU_p)             0.45 1.00     2329     2665
## cor(Intercept,TaskLO:ConditionAD_a)                 0.59 1.00     1703     2306
## cor(TaskLO,TaskLO:ConditionAD_a)                   -0.41 1.00     2104     2809
## cor(ConditionAU_p,TaskLO:ConditionAD_a)            -0.27 1.00     2026     2653
## cor(ConditionAD_a,TaskLO:ConditionAD_a)            -0.14 1.00     2281     2820
## cor(ConditionAU_a,TaskLO:ConditionAD_a)            -0.09 1.00     2255     3165
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAD_a)      0.83 1.00     2565     3162
## cor(Intercept,TaskLO:ConditionAU_a)                 0.60 1.00     1781     2420
## cor(TaskLO,TaskLO:ConditionAU_a)                   -0.39 1.00     2063     2891
## cor(ConditionAU_p,TaskLO:ConditionAU_a)            -0.15 1.00     2061     2843
## cor(ConditionAD_a,TaskLO:ConditionAU_a)             0.07 1.00     1724     2464
## cor(ConditionAU_a,TaskLO:ConditionAU_a)            -0.10 1.00     2032     2569
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAU_a)      0.82 1.00     2515     2763
## cor(TaskLO:ConditionAD_a,TaskLO:ConditionAU_a)      0.97 1.00     2452     2781
##
## Population-Level Effects:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                7.26      0.05     7.16     7.35 1.00      860
## TaskLO                  -0.27      0.07    -0.41    -0.13 1.00      763
## ConditionAU_p           -0.20      0.06    -0.31    -0.08 1.00     1078
## ConditionAD_a            0.10      0.04     0.02     0.17 1.00     1599
## ConditionAU_a            0.08      0.04     0.00     0.16 1.00     1796
## TaskLO:ConditionAU_p     0.48      0.10     0.29     0.67 1.00     1006
## TaskLO:ConditionAD_a     0.24      0.06     0.12     0.36 1.00     1150
## TaskLO:ConditionAU_a     0.18      0.06     0.06     0.30 1.00     1248
##                      Tail_ESS
## Intercept                1500
## TaskLO                   1328
```

```
## ConditionAU_p            1872
## ConditionAD_a            2316
## ConditionAU_a            2655
## TaskLO:ConditionAU_p     1622
## TaskLO:ConditionAD_a     2005
## TaskLO:ConditionAU_a     2115
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.34      0.00     0.33     0.35 1.00     5501     2891
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
msummary(m1_rt, statistic = "conf.int", statistic_vertical = FALSE)
```

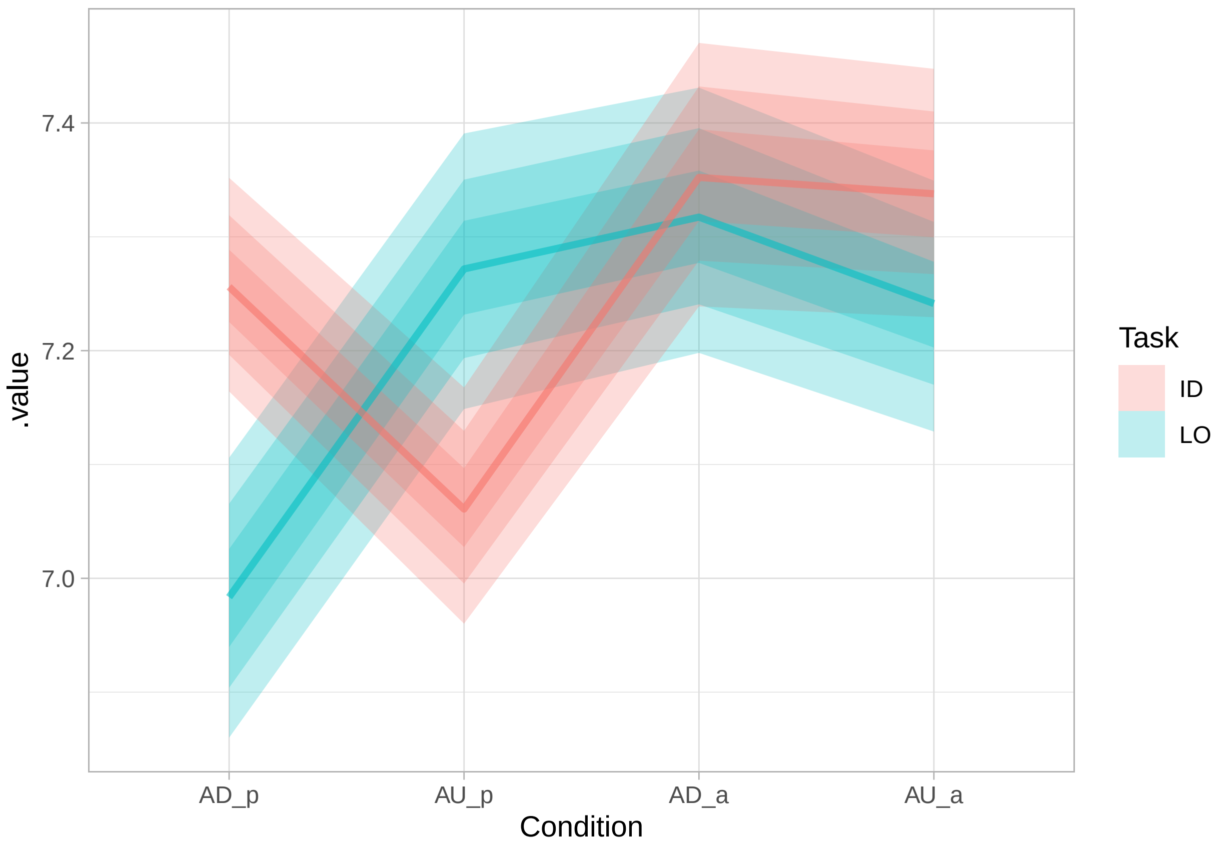
```
## Warning in tidy.brmsfit(model, conf.int = TRUE, conf.level = conf_level, : some
## parameter names contain underscores: term naming may be unreliable!
```

```
## Warning: Found 5 observations with a pareto_k > 0.7 in model 'x'. It is
## recommended to set 'moment_match = TRUE' in order to perform moment matching for
## problematic observations.
```

	Model 1
(Intercept)	7.257 [7.164, 7.352]
TaskLO	-0.274 [-0.413, -0.134]
ConditionAU_p	-0.195 [-0.306, -0.077]
ConditionAD_a	0.096 [0.019, 0.173]
ConditionAU_a	0.081 [0.003, 0.155]
TaskLO × ConditionAU_p	0.483 [0.293, 0.669]
TaskLO × ConditionAD_a	0.238 [0.115, 0.361]
TaskLO × ConditionAU_a	0.177 [0.059, 0.295]
sd___(Intercept)	0.278 [0.221, 0.357]
sd___TaskLO	0.417 [0.334, 0.520]
sd___ConditionAU_p	0.322 [0.253, 0.406]
sd___ConditionAD_a	0.200 [0.139, 0.272]
sd___ConditionAU_a	0.196 [0.136, 0.266]
sd___TaskLO × ConditionAU_p	0.556 [0.448, 0.689]
sd___TaskLO × ConditionAD_a	0.328 [0.247, 0.419]
sd___TaskLO × ConditionAU_a	0.309 [0.233, 0.400]
cor___(Intercept).TaskLO	-0.481 [-0.687, -0.219]
cor___(Intercept).ConditionAU_p	-0.443 [-0.666, -0.148]
cor___TaskLO.ConditionAU_p	0.395 [0.090, 0.639]
cor___(Intercept).ConditionAD_a	0.118 [-0.210, 0.442]
cor___TaskLO.ConditionAD_a	0.046 [-0.296, 0.353]
cor___ConditionAU_p.ConditionAD_a	0.094 [-0.250, 0.416]
cor___(Intercept).ConditionAU_a	-0.009 [-0.350, 0.326]
cor___TaskLO.ConditionAU_a	-0.016 [-0.347, 0.308]
cor___ConditionAU_p.ConditionAU_a	0.074 [-0.275, 0.391]
cor___ConditionAD_a.ConditionAU_a	0.820 [0.598, 0.952]
cor___(Intercept).TaskLO × ConditionAU_p	0.422 [0.136, 0.649]
cor___TaskLO.TaskLO × ConditionAU_p	-0.601 [-0.774, -0.369]
cor___ConditionAU_p.TaskLO × ConditionAU_p	-0.830 [-0.922, -0.688]
cor___ConditionAD_a.TaskLO × ConditionAU_p	0.149 [-0.175, 0.452]
cor___ConditionAU_a.TaskLO × ConditionAU_p	0.139 [-0.191, 0.446]
cor___(Intercept).TaskLO × ConditionAD_a	0.315 [-0.008, 0.587]
cor___TaskLO.TaskLO × ConditionAD_a	-0.651 [-0.823, -0.407]
cor___ConditionAU_p.TaskLO × ConditionAD_a	-0.553 [-0.779, -0.267]
cor___ConditionAD_a.TaskLO × ConditionAD_a	-0.457 [-0.700, -0.144]
cor___ConditionAU_a.TaskLO × ConditionAD_a	-0.419 [-0.696, -0.092]
cor___TaskLO × ConditionAU_p.TaskLO × ConditionAD_a	0.654 [0.425, 0.828]
cor___(Intercept).TaskLO × ConditionAU_a	0.340 [0.023, 0.603]
cor___TaskLO.TaskLO × ConditionAU_a	-0.642 [-0.814, -0.391]
cor___ConditionAU_p.TaskLO × ConditionAU_a	-0.462 [-0.717, -0.155]
cor___ConditionAD_a.TaskLO × ConditionAU_a	-0.280 [-0.590, 0.073]
cor___ConditionAU_a.TaskLO × ConditionAU_a	-0.430 [-0.683, -0.100]
cor___TaskLO × ConditionAU_p.TaskLO × ConditionAU_a	0.638 [0.386, 0.824]
cor___TaskLO × ConditionAD_a.TaskLO × ConditionAU_a	0.879 [0.713, 0.969]
sd___Observation	0.338 [0.329, 0.346]
Num.Obs.	3200
algorithm	sampling
elpd_loo	-1185.316
looic	2370.631
p_loo	221.820
pss	4000.000

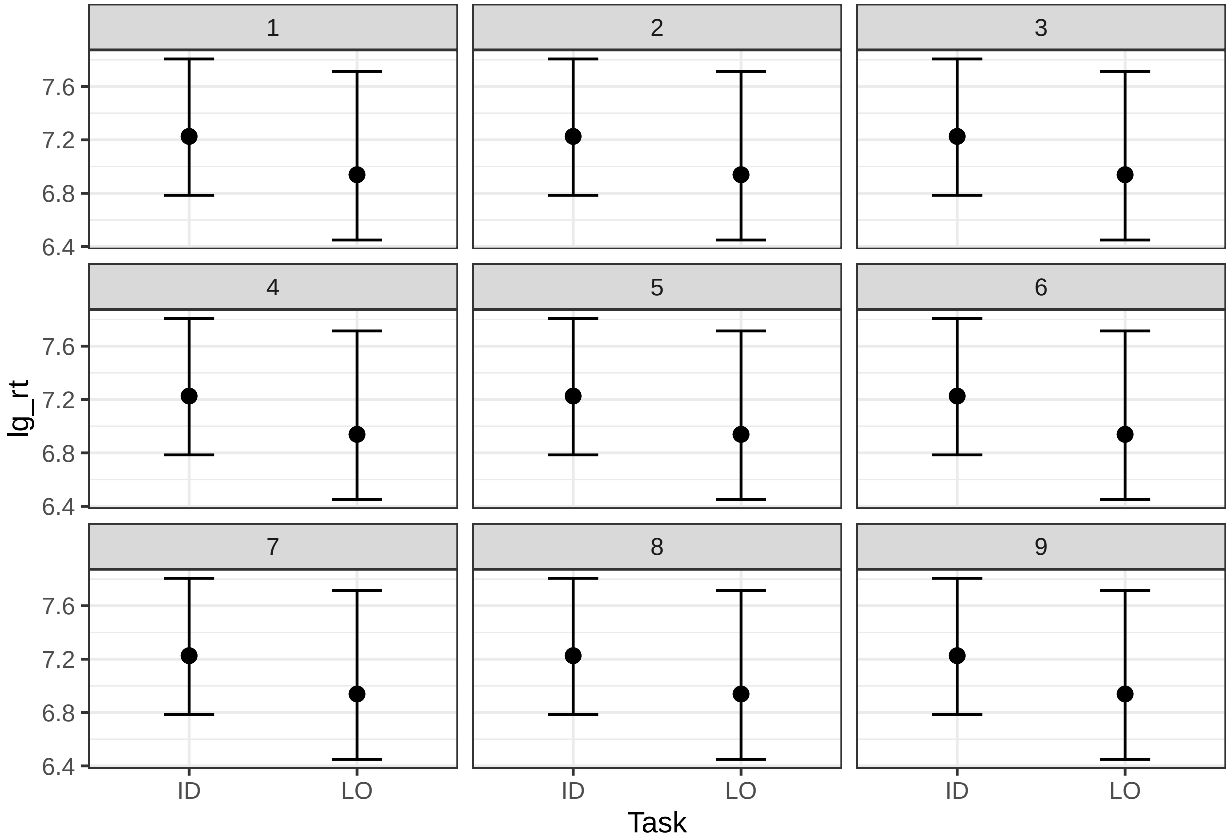
5.5 Plotting

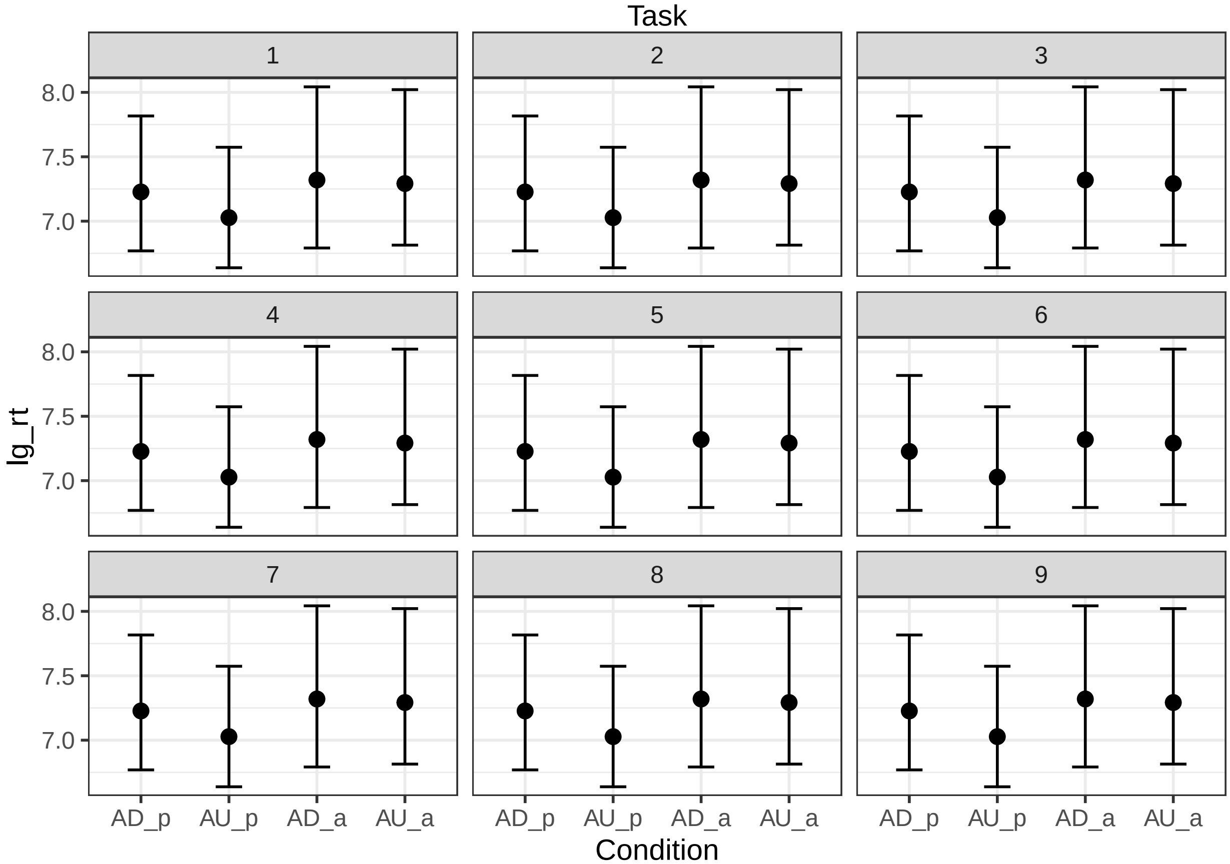
```
m1_rt %>%
  emmeans( ~ Condition | Task) %>%
  gather_emmeans_draws() %>%
  ggplot(aes(x = Condition, y = .value, fill = Task, color = Task)) +
  stat_lineribbon(alpha = 1/4) +
  theme_light()
```

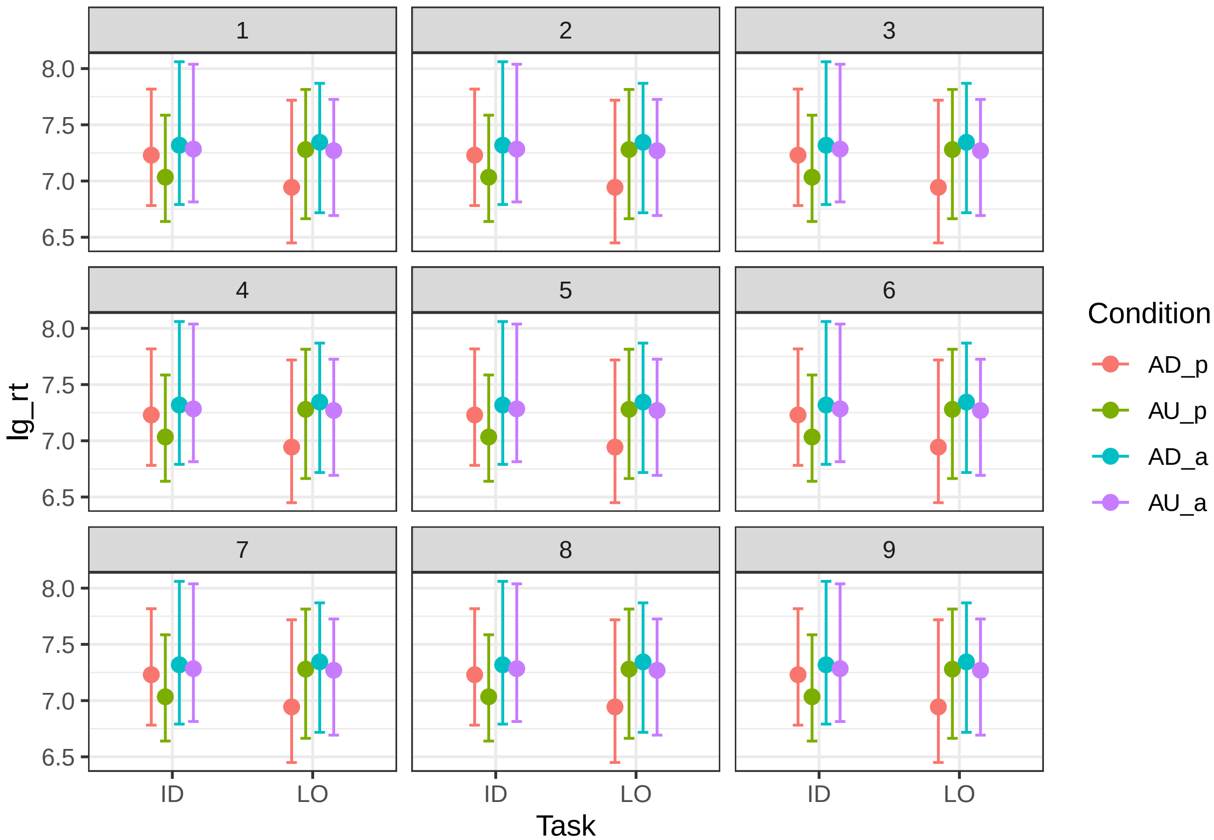



```
rand_subj <- sample(unique(ToM_dat$Subject), size = 9)
conditional_effects(m1_rt, type = "pred", re_formula = NULL,
                    conditions = tibble(subj = rand_subj))
```

```
## Warning: The following variables in 'conditions' are not part of the model:
## 'subj'
```

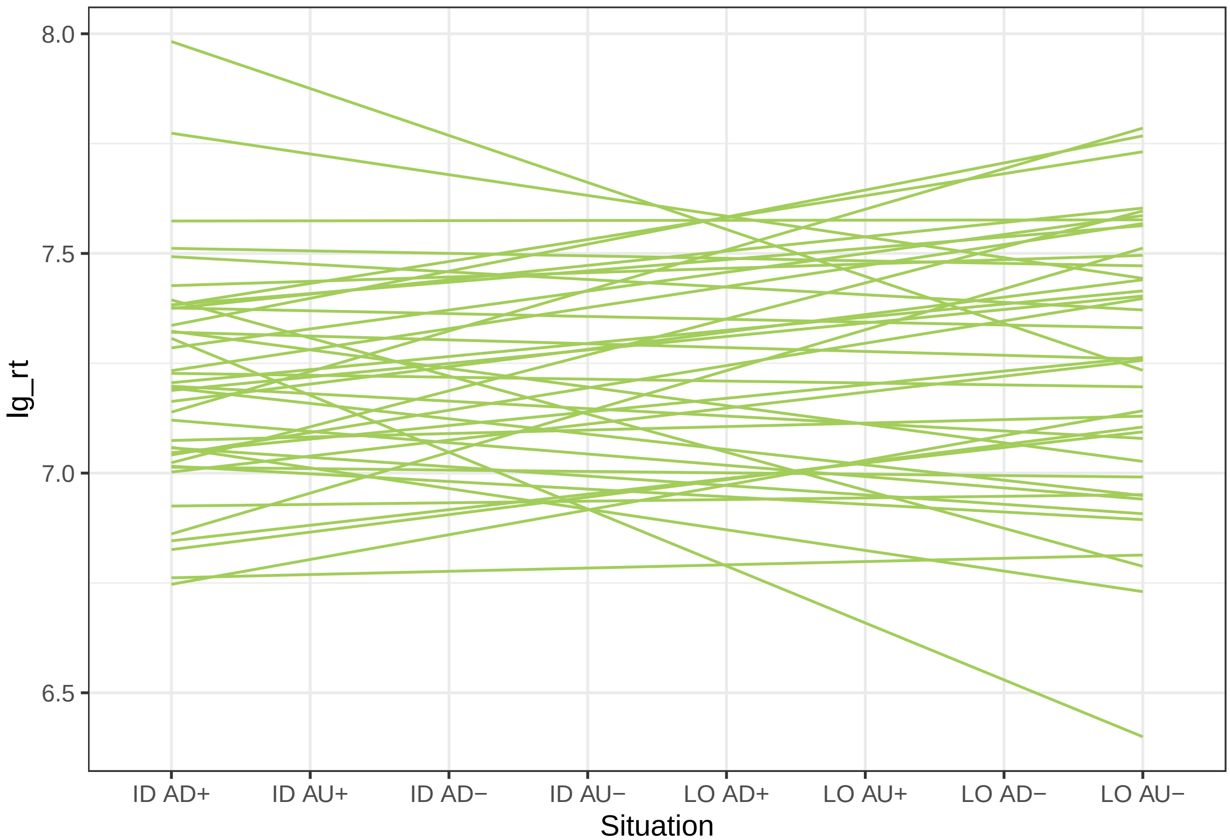

1
2
3
4
5
6
7
8
9
lg_rt
7.6
7.2
6.8
6.4
ID
LO
Task


1
2
3
4
5
6
7
8
9
lg_rt
8.0
7.5
7.0
AD_p
AU_p
AD_a
AU_a
Condition




```
m1_rt %>%
  augment(data = ToM_rt) %>%
  ggplot(aes(x = Situation, y = lg_rt, group = Subject)) +
  geom_smooth(method = "lm", se = FALSE, size = 0.5,color="darkolivegreen3")+
  scale_x_discrete(labels=c("1" = "ID AD+", "2" = "ID AU+","3" = "ID AD-","4" = "ID AU-",
                            "5" = "LO AD+", "6" = "LO AU+","7" = "LO AD-","8" = "LO AU-"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



5.6 Interpretation

TaskLO represents the difference between Location and Identity Task in Condition AD_p (because it is the reference group).In Condition AD_p, the difference between Location and Identity Task is significant. RTs in Location Task are shorter than in Identity Task whenparticipants experienced Condition AD_p. As for condition, AU_p has the shortest reaction time in Identity Task. If we look at the interaction, the coefficient for [TaskLO x ConditionAU_p] represents how much the differnce between Location Task and Identity Task differs between Condition AU_p and AD_p. The CI doesn't contain 0 so the interaction we are most interested in, between COndition AD+ AU+ and Task is significant. Furthermore, as we mentioned, the random slopes here are also significant. If we take a sample of 9 participants, we can see that there are many individual differences in their reaction times. Overall, the results of reaction time can support the hypothesis that When the actor falsely believed that a desired object was in the box, participants would be faster in Location Task than in Identity Task, while when the actor falsely believed that an undesired object was in the box, participants would be faster in Location Task than in Identity Task. This reveals the identity limits in the efficient mind-reading system.

6. Analysis of Accuracy

6.1 Data Preprocessing

For each situation, each participant will be measured 10 times. In a single trial, if they judge correctly, they will get 1, if not, they will get 0. Thus, there are two methods to deal with accuracy data. The first one is use a number calculated by n/10 to represent their accuracy rate for each cell. In this case, each participant has a score ranging from 0 to 1 for each situation. The second way is to use logistic MLM. Considering that the data is binary, and most people reacted correct in 8-9 trials out of 10 trials, I will use logistic MLM to analyze accuracy.

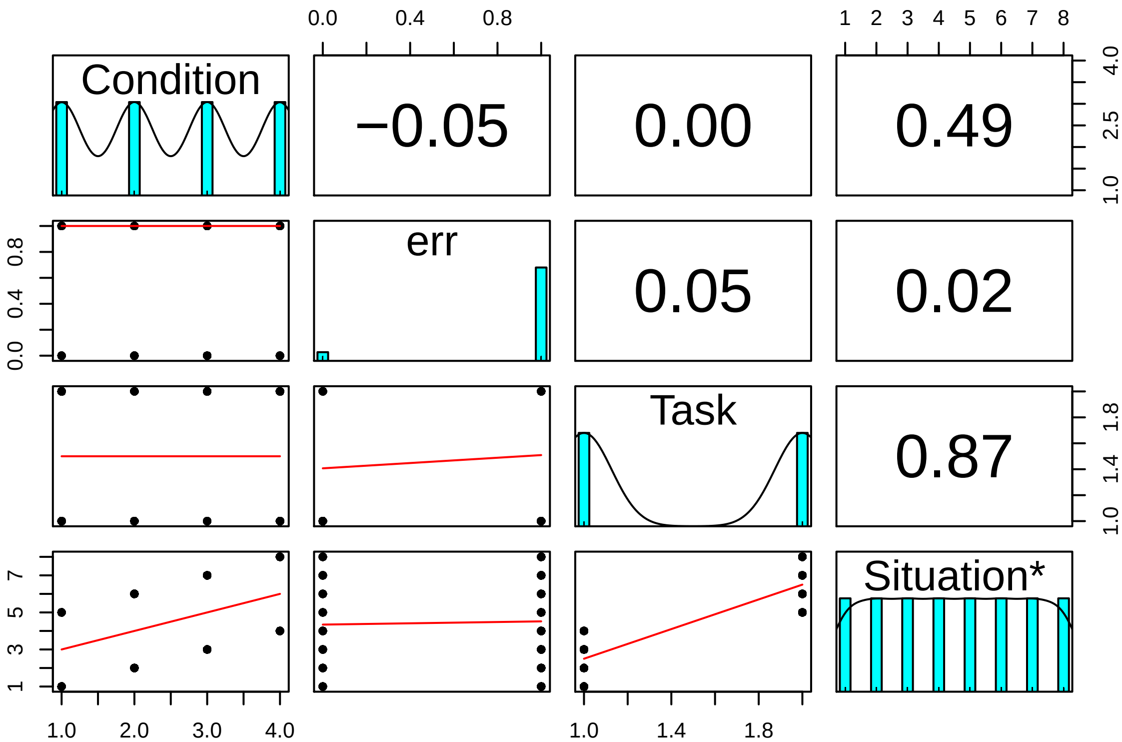
```r
head(ToM_acc)
```

```
## # A tibble: 6 x 5
##   Subject Condition   err Task  Situation
##     <int> <chr>     <int> <chr> <chr>
## 1       1 AD_p          1 ID    1
## 2       1 AU_p          1 ID    2
## 3       1 AD_a          1 ID    3
## 4       1 AU_a          1 ID    4
## 5       1 AD_p          0 LO    5
## 6       1 AU_p          1 LO    6
```

```r
# Recode the levels of each predictor
ToM_acc <- ToM_acc %>%
  mutate(Task = factor(Task, levels = c("ID", "LO")))
ToM_acc <- ToM_acc %>%
  mutate(Condition = factor(Condition, levels = c("AD_p", "AU_p","AD_a","AU_a")))
```

6.2 Data Exploration

```r
pairs.panels(ToM_acc[ , -1],
             ellipses = FALSE)
```



6.3 Unconditional model with random intercepts

First, I run an unconditional model with random intercepts of both `Subject` and `Situation`.

```r
m0_acc <- brm(err ~ (1 | Subject) + (1 | Situation), data = ToM_acc,
              family = bernoulli("logit"),
               seed = 31420)
```

```
## Warning: Rows containing NAs were excluded from the model.

## Compiling Stan program...

## Start sampling

##
## SAMPLING FOR MODEL '06e3ec92606669828964f4838e94a585' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000384 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 3.84 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: 6.20268 seconds (Warm-up)
## Chain 1:                6.26689 seconds (Sampling)
## Chain 1:                12.4696 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '06e3ec92606669828964f4838e94a585' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000254 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 2.54 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: 6.08898 seconds (Warm-up)
## Chain 2:                4.68629 seconds (Sampling)
## Chain 2:                10.7753 seconds (Total)
```

```
## Chain 2:
##
## SAMPLING FOR MODEL '06e3ec92606669828964f4838e94a585' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000246 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.46 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: 6.10831 seconds (Warm-up)
## Chain 3:                5.02521 seconds (Sampling)
## Chain 3:                11.1335 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '06e3ec92606669828964f4838e94a585' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000331 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 3.31 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: 5.80935 seconds (Warm-up)
## Chain 4:                6.095 seconds (Sampling)
## Chain 4:                11.9043 seconds (Total)
## Chain 4:
```

```r
# Calculate ICC
post_tau <- posterior_samples(m0_acc, pars = c("sd"))
```

```r
# ICC for Subject
icc_samples_sub <- post_tau$sd_Subject__Intercept^2 /
  (post_tau$sd_Subject__Intercept^2 + pi^2 / 3)
posterior_summary(icc_samples_sub)
```

```
##        Estimate  Est.Error       Q2.5     Q97.5
## [1,] 0.08453863 0.03123707 0.03678951 0.1591807
```

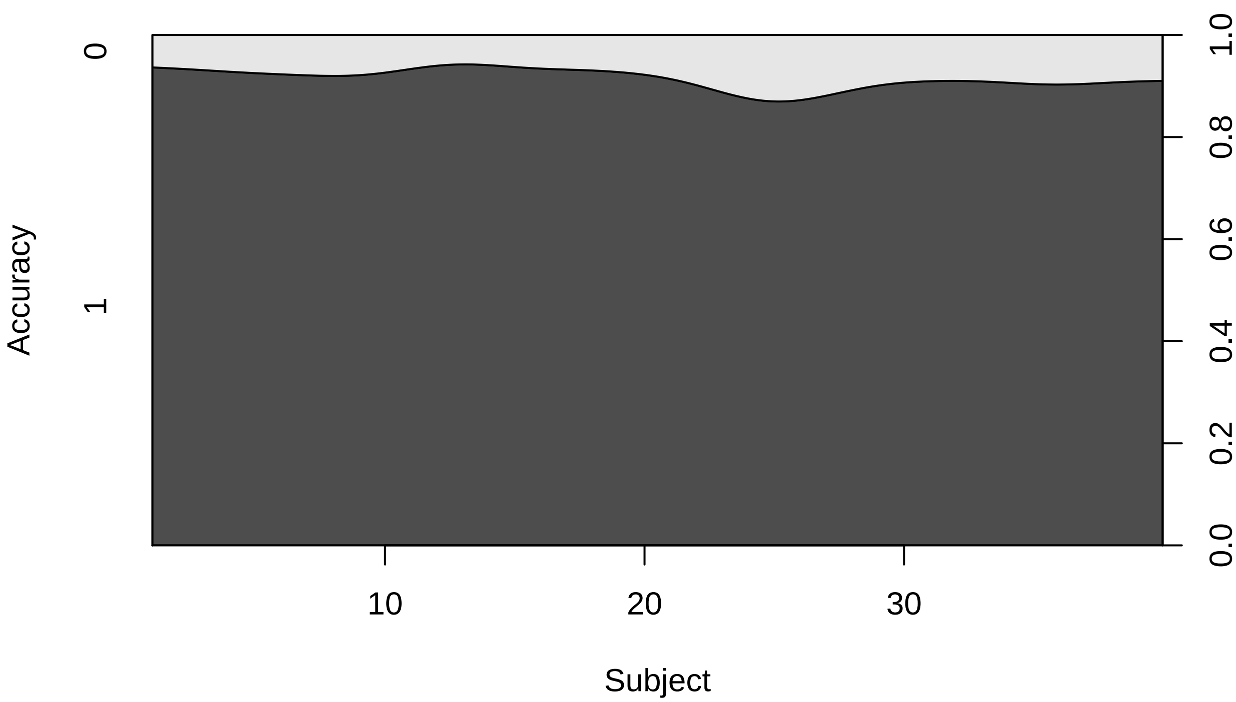
```r
# Design effect
Deff_sub <- 1 + ((8-1) * 0.08453863)
# ICC for Situation
icc_samples_si <- post_tau$sd_Situation__Intercept^2 /
  (post_tau$sd_Situation__Intercept^2 + pi^2 / 3)
posterior_summary(icc_samples_si)
```

```
##        Estimate  Est.Error       Q2.5     Q97.5
## [1,] 0.09521494 0.06924856 0.02362641 0.2790373
```

```r
Deff_si <- 1 + ((40-1) * 0.09521494)
```

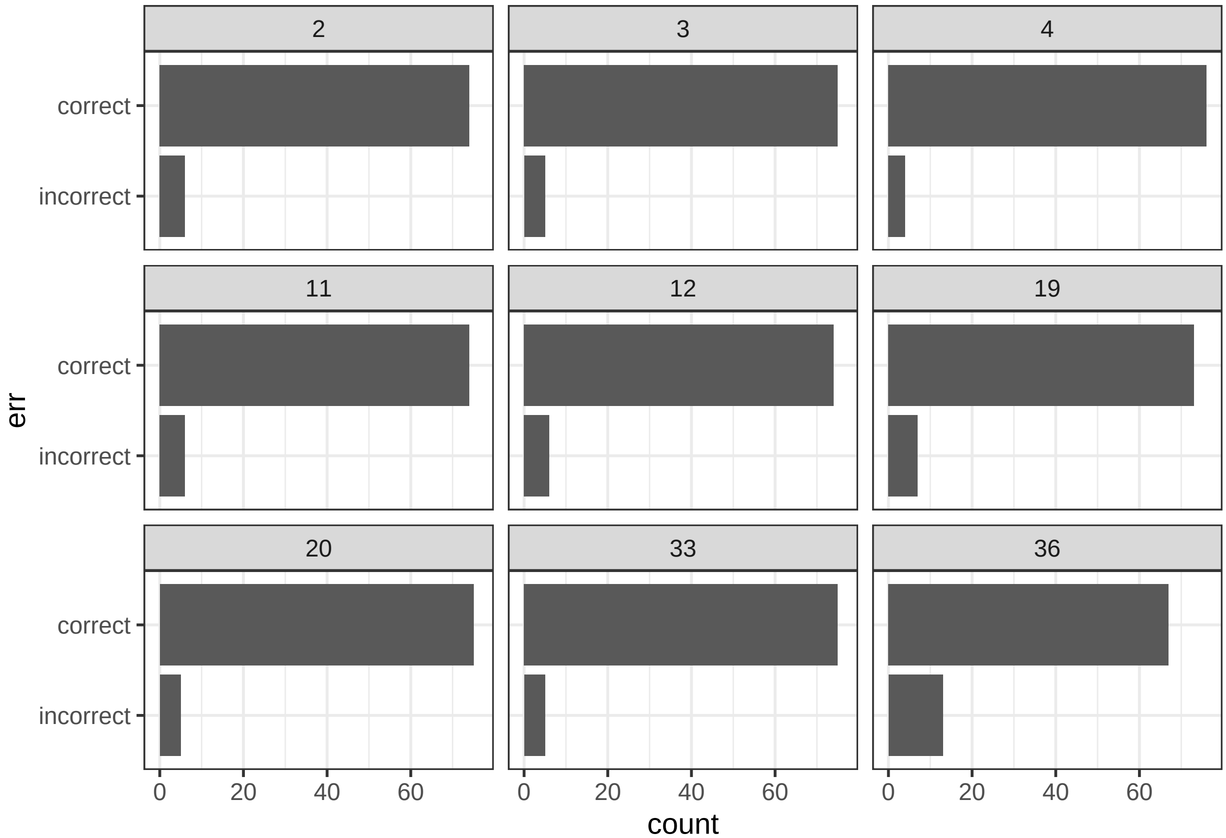
The results show that the ICC of `Subject` is .085 with the design effect of 1.592. There is a very week correlation between two randomly drawn units from the same subject. The ICC of `Situation` is .095 with the design effect of 4.713. This time, the correlation between two randomly drawn units from the same subject is very week. The distribution of accuracy and a example subset are plotted as below.

```r
cdplot(factor(ToM_acc$err) ~ ToM_acc$Subject, xlab = "Subject", ylab = "Accuracy")
```




```r
set.seed(31420)
# Randomly select some subjects
random_subjects <- sample(ToM_acc$Subject, size = 9)
ToM_acc %>%
  filter(Subject %in% random_subjects) %>%
  mutate(err = factor(err, labels = c("incorrect", "correct"))) %>%
  ggplot(aes(x = err)) +
```

```r
  geom_bar() +
  facet_wrap( ~ Subject, ncol = 3) +
  coord_flip()
```



6.5 Modeling for `acc`

6.5.1 Model Equations

Repeated-Measure level (Lv 1):

$$
\begin{aligned}
\text{acc}_{ijk} &\sim \text{Bernoulli}(\mu_{ijk}) \\
\eta_{ij} &= \text{logit}(\mu_{ijk}) \\
\eta_{ijk} &= \beta_{0j}
\end{aligned}
$$

Lv 2:

$$\beta_{0(j,k)} = \gamma_{00} + \beta_{1j}\text{Task}_{ik} + \beta_{2j}\text{Condition}_{ik} + \beta_{3j}\text{Task}_{ik} \times \text{Condition}_{ik} + u_{0j} + v_{0k}$$

Condition level (Lv 2a) random slopes

$$
\begin{aligned}
\beta_{1j} &= \gamma_{10} + +u_{1j} \\
\beta_{2j} &= \gamma_{20} + u_{2j} \\
\beta_{3j} &= \gamma_{30} + u_{3j}
\end{aligned}
$$

6.5.2 Fit a Model

Now I fit another Bayesian multilevel model here. Note that this is a logistic model. Random slopes will also be included.

```r
m1_acc <-
  brm(err ~ Task + Condition + Task*Condition + (Task + Condition + Task*Condition | Subject),
      data = ToM_acc,
      family = bernoulli("logit"),
      seed = 112314)
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '1637acbdd3bcd411308c71837c6bc245' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.002095 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 20.95 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: 59.0472 seconds (Warm-up)
## Chain 1:                69.5871 seconds (Sampling)
## Chain 1:                128.634 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '1637acbdd3bcd411308c71837c6bc245' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001086 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 10.86 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: 59.0548 seconds (Warm-up)
## Chain 2:                34.7598 seconds (Sampling)
## Chain 2:                93.8146 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '1637acbdd3bcd411308c71837c6bc245' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.00108 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10.8 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: 60.3581 seconds (Warm-up)
## Chain 3:                67.4498 seconds (Sampling)
## Chain 3:                127.808 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '1637acbdd3bcd411308c71837c6bc245' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001269 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 12.69 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: 57.6705 seconds (Warm-up)
```

```
## Chain 4:                34.0198 seconds (Sampling)
## Chain 4:                91.6903 seconds (Total)
## Chain 4:
```

```r
summary(m1_acc)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: err ~ Task + Condition + Task * Condition + (Task + Condition + Task * Condition | Subject)
##    Data: ToM_acc (Number of observations: 3199)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~Subject (Number of levels: 40)
##                                                 Estimate Est.Error l-95% CI
## sd(Intercept)                                       0.22      0.15     0.01
## sd(TaskLO)                                          0.28      0.21     0.01
## sd(ConditionAU_p)                                   0.54      0.32     0.03
## sd(ConditionAD_a)                                   1.29      0.27     0.84
## sd(ConditionAU_a)                                   0.68      0.29     0.09
## sd(TaskLO:ConditionAU_p)                            1.47      0.60     0.35
## sd(TaskLO:ConditionAD_a)                            0.64      0.42     0.03
## sd(TaskLO:ConditionAU_a)                            1.34      0.66     0.14
## cor(Intercept,TaskLO)                               0.02      0.32    -0.60
## cor(Intercept,ConditionAU_p)                       -0.01      0.33    -0.63
## cor(TaskLO,ConditionAU_p)                           0.09      0.33    -0.58
## cor(Intercept,ConditionAD_a)                       -0.05      0.32    -0.65
## cor(TaskLO,ConditionAD_a)                          -0.04      0.32    -0.60
## cor(ConditionAU_p,ConditionAD_a)                    0.12      0.29    -0.47
## cor(Intercept,ConditionAU_a)                       -0.06      0.33    -0.66
## cor(TaskLO,ConditionAU_a)                           0.05      0.33    -0.59
## cor(ConditionAU_p,ConditionAU_a)                   -0.04      0.31    -0.62
## cor(ConditionAD_a,ConditionAU_a)                   -0.20      0.27    -0.69
## cor(Intercept,TaskLO:ConditionAU_p)                 0.01      0.32    -0.61
## cor(TaskLO,TaskLO:ConditionAU_p)                   -0.06      0.32    -0.65
## cor(ConditionAU_p,TaskLO:ConditionAU_p)            -0.05      0.31    -0.61
## cor(ConditionAD_a,TaskLO:ConditionAU_p)            -0.24      0.26    -0.71
## cor(ConditionAU_a,TaskLO:ConditionAU_p)             0.01      0.30    -0.56
## cor(Intercept,TaskLO:ConditionAD_a)                 0.08      0.33    -0.57
## cor(TaskLO,TaskLO:ConditionAD_a)                   -0.04      0.34    -0.66
## cor(ConditionAU_p,TaskLO:ConditionAD_a)             0.12      0.33    -0.54
## cor(ConditionAD_a,TaskLO:ConditionAD_a)            -0.09      0.30    -0.65
## cor(ConditionAU_a,TaskLO:ConditionAD_a)             0.12      0.32    -0.53
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAD_a)     -0.02      0.31    -0.61
## cor(Intercept,TaskLO:ConditionAU_a)                -0.03      0.32    -0.63
## cor(TaskLO,TaskLO:ConditionAU_a)                   -0.15      0.34    -0.73
## cor(ConditionAU_p,TaskLO:ConditionAU_a)            -0.02      0.32    -0.62
## cor(ConditionAD_a,TaskLO:ConditionAU_a)             0.06      0.28    -0.50
## cor(ConditionAU_a,TaskLO:ConditionAU_a)            -0.04      0.30    -0.60
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAU_a)     -0.01      0.30    -0.61
## cor(TaskLO:ConditionAD_a,TaskLO:ConditionAU_a)     -0.14      0.32    -0.70
##                                                 u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)                                       0.55 1.00     1277     1450
## sd(TaskLO)                                          0.77 1.00     1393     1592
```

```
## sd(ConditionAU_p)                                     1.23 1.00     1077     1511
## sd(ConditionAD_a)                                     1.86 1.00     1803     2454
## sd(ConditionAU_a)                                     1.26 1.01      837      906
## sd(TaskLO:ConditionAU_p)                              2.79 1.01      986      577
## sd(TaskLO:ConditionAD_a)                              1.59 1.00      913     1735
## sd(TaskLO:ConditionAU_a)                              2.73 1.00      918      803
## cor(Intercept,TaskLO)                                 0.62 1.00     3836     2701
## cor(Intercept,ConditionAU_p)                          0.62 1.00     2824     2821
## cor(TaskLO,ConditionAU_p)                             0.68 1.00     2132     2604
## cor(Intercept,ConditionAD_a)                          0.59 1.00      740     1363
## cor(TaskLO,ConditionAD_a)                             0.61 1.01      509     1038
## cor(ConditionAU_p,ConditionAD_a)                      0.65 1.01      805     1456
## cor(Intercept,ConditionAU_a)                          0.59 1.00     1521     2414
## cor(TaskLO,ConditionAU_a)                             0.65 1.01     1450     2163
## cor(ConditionAU_p,ConditionAU_a)                      0.56 1.00     2064     2989
## cor(ConditionAD_a,ConditionAU_a)                      0.36 1.00     3247     2844
## cor(Intercept,TaskLO:ConditionAU_p)                   0.61 1.00     1614     2197
## cor(TaskLO,TaskLO:ConditionAU_p)                      0.59 1.00     1124     2431
## cor(ConditionAU_p,TaskLO:ConditionAU_p)               0.56 1.01     1464     2269
## cor(ConditionAD_a,TaskLO:ConditionAU_p)               0.30 1.00     2687     3067
## cor(ConditionAU_a,TaskLO:ConditionAU_p)               0.59 1.00     1985     2696
## cor(Intercept,TaskLO:ConditionAD_a)                   0.67 1.00     2654     2590
## cor(TaskLO,TaskLO:ConditionAD_a)                      0.62 1.00     2436     2621
## cor(ConditionAU_p,TaskLO:ConditionAD_a)               0.72 1.00     2222     2436
## cor(ConditionAD_a,TaskLO:ConditionAD_a)               0.52 1.00     3415     3282
## cor(ConditionAU_a,TaskLO:ConditionAD_a)               0.69 1.00     2507     2810
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAD_a)        0.58 1.00     2998     3336
## cor(Intercept,TaskLO:ConditionAU_a)                   0.60 1.00     1513     2542
## cor(TaskLO,TaskLO:ConditionAU_a)                      0.55 1.00     1541     2269
## cor(ConditionAU_p,TaskLO:ConditionAU_a)               0.58 1.00     2129     2788
## cor(ConditionAD_a,TaskLO:ConditionAU_a)               0.60 1.00     3576     3079
## cor(ConditionAU_a,TaskLO:ConditionAU_a)               0.55 1.00     2174     2297
## cor(TaskLO:ConditionAU_p,TaskLO:ConditionAU_a)        0.57 1.00     2531     2729
## cor(TaskLO:ConditionAD_a,TaskLO:ConditionAU_a)        0.52 1.00     1938     2810
##
## Population-Level Effects:
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                2.45      0.19     2.09     2.82 1.00     2889
## TaskLO                   0.71      0.31     0.11     1.34 1.00     2175
## ConditionAU_p            0.24      0.31    -0.35     0.86 1.00     2795
## ConditionAD_a            0.17      0.36    -0.50     0.88 1.00     2124
## ConditionAU_a           -0.37      0.28    -0.91     0.23 1.00     2563
## TaskLO:ConditionAU_p     0.37      0.67    -0.78     1.81 1.00     1891
## TaskLO:ConditionAD_a    -0.95      0.44    -1.82    -0.06 1.00     2267
## TaskLO:ConditionAU_a     1.05      0.66    -0.08     2.50 1.00     1623
##                      Tail_ESS
## Intercept                3365
## TaskLO                   2498
## ConditionAU_p            2851
## ConditionAD_a            2739
## ConditionAU_a            3032
## TaskLO:ConditionAU_p     2087
## TaskLO:ConditionAD_a     2666
## TaskLO:ConditionAU_a     2007
```

```
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
msummary(m1_acc, statistic = "conf.int", statistic_vertical = FALSE)
```

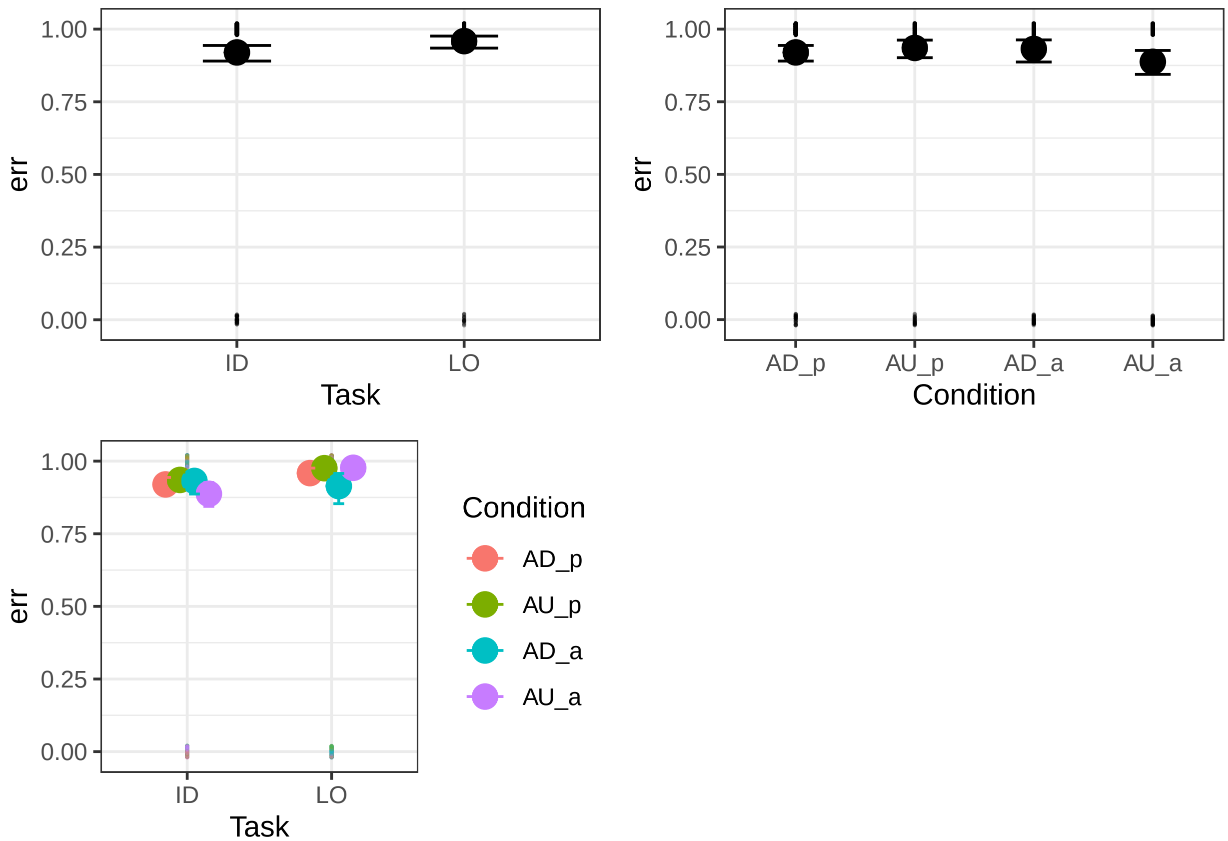
```
## Warning in tidy.brmsfit(model, conf.int = TRUE, conf.level = conf_level, : some
## parameter names contain underscores: term naming may be unreliable!
```

```
## Warning: Found 1 observations with a pareto_k > 0.7 in model 'x'. It is
## recommended to set 'moment_match = TRUE' in order to perform moment matching for
## problematic observations.
```

	Model 1
(Intercept)	2.447 [2.091, 2.821]
TaskLO	0.706 [0.109, 1.344]
ConditionAU_p	0.235 [-0.353, 0.864]
ConditionAD_a	0.173 [-0.500, 0.881]
ConditionAU_a	-0.371 [-0.907, 0.229]
TaskLO × ConditionAU_p	0.369 [-0.782, 1.807]
TaskLO × ConditionAD_a	-0.947 [-1.824, -0.057]
TaskLO × ConditionAU_a	1.050 [-0.083, 2.502]
sd___(Intercept)	0.218 [0.008, 0.552]
sd___TaskLO	0.278 [0.009, 0.767]
sd___ConditionAU_p	0.544 [0.031, 1.231]
sd___ConditionAD_a	1.292 [0.840, 1.865]
sd___ConditionAU_a	0.680 [0.094, 1.260]
sd___TaskLO × ConditionAU_p	1.467 [0.349, 2.795]
sd___TaskLO × ConditionAD_a	0.644 [0.035, 1.594]
sd___TaskLO × ConditionAU_a	1.336 [0.136, 2.735]
cor___(Intercept).TaskLO	0.025 [-0.600, 0.617]
cor___(Intercept).ConditionAU_p	-0.013 [-0.628, 0.625]
cor___TaskLO.ConditionAU_p	0.087 [-0.581, 0.682]
cor___(Intercept).ConditionAD_a	-0.045 [-0.649, 0.586]
cor___TaskLO.ConditionAD_a	-0.041 [-0.604, 0.605]
cor___ConditionAU_p.ConditionAD_a	0.116 [-0.473, 0.652]
cor___(Intercept).ConditionAU_a	-0.056 [-0.661, 0.586]
cor___TaskLO.ConditionAU_a	0.048 [-0.589, 0.650]
cor___ConditionAU_p.ConditionAU_a	-0.044 [-0.625, 0.559]
cor___ConditionAD_a.ConditionAU_a	-0.204 [-0.691, 0.363]
cor___(Intercept).TaskLO × ConditionAU_p	0.008 [-0.611, 0.608]
cor___TaskLO.TaskLO × ConditionAU_p	-0.060 [-0.645, 0.590]
cor___ConditionAU_p.TaskLO × ConditionAU_p	-0.048 [-0.615, 0.561]
cor___ConditionAD_a.TaskLO × ConditionAU_p	-0.240 [-0.715, 0.301]
cor___ConditionAU_a.TaskLO × ConditionAU_p	0.014 [-0.560, 0.588]
cor___(Intercept).TaskLO × ConditionAD_a	0.078 [-0.573, 0.670]
cor___TaskLO.TaskLO × ConditionAD_a	-0.045 [-0.660, 0.622]
cor___ConditionAU_p.TaskLO × ConditionAD_a	0.120 [-0.540, 0.723]
cor___ConditionAD_a.TaskLO × ConditionAD_a	-0.088 [-0.648, 0.516]
cor___ConditionAU_a.TaskLO × ConditionAD_a	0.115 [-0.532, 0.691]
cor___TaskLO × ConditionAU_p.TaskLO × ConditionAD_a	-0.022 [-0.605, 0.581]
cor___(Intercept).TaskLO × ConditionAU_a	-0.026 [-0.632, 0.596]
cor___TaskLO.TaskLO × ConditionAU_a	-0.149 [-0.729, 0.548]
cor___ConditionAU_p.TaskLO × ConditionAU_a	-0.024 [-0.623, 0.584]
cor___ConditionAD_a.TaskLO × ConditionAU_a	0.063 [-0.502, 0.595]
cor___ConditionAU_a.TaskLO × ConditionAU_a	-0.036 [-0.605, 0.554]
cor___TaskLO × ConditionAU_p.TaskLO × ConditionAU_a	-0.015 [-0.608, 0.571]
cor___TaskLO × ConditionAD_a.TaskLO × ConditionAU_a	-0.144 [-0.699, 0.518]
Num.Obs.	3199
algorithm	sampling
elpd_loo	-859.596
looic	1719.193
p_loo	92.006
pss	4000.000

6.6 Plotting

```
m1_plots <- plot(
  conditional_effects(
    m1_acc
  ),
  points = TRUE,
  point_args = c(height = 0.02, alpha = 0.3, size = 0.1),
  plot = FALSE
)
gridExtra::grid.arrange(grobs = m1_plots, ncol = 2)
```



6.7 Interpretation

For accuracy, participants still got more correct responses in Location Task than in Identity Task when they are in Condition AD+. As for condition, there is no significant difference between `ConditionAD_p` and other three conditions in Identity Task. In addition, we cannot find significant interactions between Task and Condition in accuracy. But we still see random slops of `Task` and `Condition` across `Subject`. In a nutshell, results of accuracy cannot support the hypotheses.