A model data analysis: Ego depletion

MK

December 13, 2019

Contents

1 Ego depletion

Ego depletion is widely-debated topic in the psychological literature. It refers to the claim that people have less self-control in one domain after having to control themselves in another domain of their lives. Thus, one might expect that a person who performs a task that requires self-control (such as the suppression of thoughts, emotions, behavioral impulses, and habits) may not be able to exercise as much self-control in a subsequent task as a person who had not been required to engage in self-control. For example, people who had resisted tempting food, subsequently ate a larger amount of palatable (but unhealthy) food. Because it has been difficult to replicate studies of ego depletion, a group of psychologists in 23 labs independently conducted experiments using an agreed-upon standardized experimental method.

Method of the multilab replication of the ego-depletion effect

Data Collection

Participants were randomly allocated to experimental (ego-depletion) or control (no depletion) groups.

Procedure

They were told that the experiment was a study of speeded recognition of words and numbers. It consisted of two tasks: the *"*`e`*" task*, and the *MSIT*, both of which we will describe in a moment. Participants first practiced 20 trials from each tasks. Then they did the "`e`" task, after which they rated how effortful, fatiguing, difficult, and frustrating it had been. They then did the second task.

Materials

The "`e`*" task.* There were two conditions: *depletion* and *no depletion*, in which participants responded to each word as it appeared on the screen. In the depletion condition, they pressed a button when the word contained the letter "`e`," and refrained from pressing if the "`e`" was next to or one letter away from a vowel (an example might be the word `done`). In the no-depletion version they pressed a button whenever the word contained "`e`" (for example, `done`). The main session consisted of 150 trials and lasted 7.5 minutes.

Multi-source interference task (MSIT). The MSIT requires response inhibition. The stimuli were sets of three digits (the numerals `1`, `2`, `3`, or `0`). Participants placed their right index, middle, and ring fingers on three keys on the keyboard. On each trial, one of the digits (the *target*; `1`, `2`, or `3`) differed from the two identical *distractors*. They pressed the key corresponding to the target digit. On *congruent* trials, the target matched its position on the keyboard, the distractors were `0`s, and the size of the target differed from the distractors (for example: `0`2`0`). On *incongruent* trials, the target did not match its position, the distractors were potential targets,

and the different-size digit was not always the target (for example: 233). The session consisted of 200 trials (100 congruent and 100 incongruent), which took about 10 minutes.

The data herein were collected by the lab of Mark J. Brandt, Tilburg University, the Netherlands (OSF: https://osf.io/x3y9b/).

2 Import and tidy

Contents

2.1 e task

2.1.1 Read in

```
mkable <- function(data) {
  knitr::kable(data, booktabs = TRUE, digits = 2) %>%
    kable_styling(position = "center")
}
mkables <- function(data) {
  knitr::kable(data, booktabs = TRUE, digits = 2) %>%
    kable_styling(position = "center", latex_options = "scale_down")
}
```

```
letE <-
  read.delim("~/Dropbox/StatsBook/Book5/EgoDepletion/LetterE_EData.txt",
    fileEncoding = "UTF-16"
  )
# letE should be in package
glimpse(letE)

Observations: 24,750
Variables: 35
$ ExperimentName          <fct> LetterETaskt_150_3_easy_NL, LetterETaskt_15...
$ Subject                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Session                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Clock.Information       <fct> <?xml version=1.0?>\n<Clock xmlns:dt=urn:sc...
$ DataFile.Basename       <fct> LetterETaskt_150_3_easy_NL-1-1, LetterETask...
$ Display.RefreshRate     <dbl> 60.001, 60.001, 60.001, 60.001, 60.001, 60....
$ ExperimentVersion       <fct> 1.0.0.63, 1.0.0.63, 1.0.0.63, 1.0.0.63, 1.0...
$ Group                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ RandomSeed              <int> 985805828, 985805828, 985805828, 985805828,...
$ RuntimeVersion          <fct> 2.0.10.353, 2.0.10.353, 2.0.10.353, 2.0.10....
$ RuntimeVersionExpected  <fct> 2.0.10.242, 2.0.10.242, 2.0.10.242, 2.0.10....
$ SessionDate             <fct> 03-30-2015, 03-30-2015, 03-30-2015, 03-30-2...
$ SessionStartDateTimeUtc <fct> 30-3-2015 7:07:29, 30-3-2015 7:07:29, 30-3-...
$ SessionTime             <fct> 09:07:29, 09:07:29, 09:07:29, 09:07:29, 09:...
$ StudioVersion           <fct> 2.0.10.147, 2.0.10.147, 2.0.10.147, 2.0.10....
$ Block                   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ Procedure               <fct> ShowStim, ShowStim, ShowStim, ShowStim, Sho...
$ Running                 <fct> WordList, WordList, WordList, WordList, Wor...
$ target                  <int> 1, 1, 1, 1, NA, 1, 1, NA, 1, 1, 1, 1, 1, 1,...
$ TrialNum                <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ TrialType               <int> 3, 3, 2, 3, 1, 3, 2, 1, 3, 3, 3, 2, 2, 3, 1...
```

```
$ TrialTypeC              <fct> Lonely, Lonely, Companioned, Lonely, NoE, L...
$ Word                    <fct> scheppen, gebruikt, omstandigheid, zuster, ...
$ Word.ACC                <int> 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0...
$ Word.CRESP              <int> 1, 1, 1, 1, NA, 1, 1, NA, 1, 1, 1, 1, 1, 1,...
$ Word.DurationError      <int> -7, -7, -7, -7, -7, -7, -7, -7, -7, -7, -7,...
$ Word.OnsetDelay         <int> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7...
$ Word.OnsetTime          <int> 13449, 16449, 19449, 22449, 25449, 28449, 3...
$ Word.OnsetToOnsetTime   <int> 1500, 1500, 1500, 1500, 1500, 1500, 1500, 1...
$ Word.RESP               <int> 1, 1, 1, 1, NA, 1, 1, NA, 1, 1, 1, 1, 1, 1,...
$ Word.RT                 <int> 625, 425, 729, 537, 0, 457, 417, 0, 336, 38...
$ Word.RTTime             <int> 14074, 16874, 20178, 22986, 0, 28906, 31866...
$ WordList                <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ WordList.Cycle          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ WordList.Sample         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
```

Note: UTF-16 allows us to read a wide range of characters, such as used in text not coded in Latin-script alphabets, and also Greek, Cyrillic, Coptic, Armenian, Hebrew, Arabic, Syriac (roughly equivalent to Aramaic), Thaana (Maldives), and N'Ko (Manding languages). We need it here, because the data were collected in a Dutch-language setting.

2.1.2 Tidy

```
eTask <- as_tibble(letE) %>%
  replace_na(list(Word.RESP = 0, Word.CRESP = 0))
eTask <- eTask %>%
  mutate(
    taskDifficulty = as_factor(
      case_when(
        ExperimentName == "LetterETaskt_150_3_hard_NL" ~ "Hard",
        ExperimentName == "LetterETaskt_150_3_easy_NL" ~ "Easy"
      )
    ),
    goNoGo = as_factor(
      case_when(
        taskDifficulty == "Easy" & TrialType %in% c(2, 3) ~ "Go",
        taskDifficulty == "Hard" & TrialType %in% c(3) ~ "Go",
        taskDifficulty == "Easy" & TrialType %in% c(1) ~ "NoGo",
        taskDifficulty == "Hard" & TrialType %in% c(1, 2) ~ "NoGo"
      )
    ),
    displayType = as_factor(
      case_when(
        TrialTypeC == "Lonely" ~ "NoFlankers",
        TrialTypeC == "Companioned" ~ "Flankers",
        TrialTypeC == "Lonely" ~ "NoE"
      )
    ),
    sigDet = as_factor(
      case_when(
        Word.RESP == 1 & Word.CRESP == 1 ~ "Hit",
        Word.RESP == 0 & Word.CRESP == 1 ~ "Miss",
        Word.RESP == 1 & Word.CRESP == 0 ~ "FA",
        Word.RESP == 0 & Word.CRESP == 0 ~ "CR"
```

```
      )
    ),
    signalNoise = as_factor(
      case_when(
        Word.CRESP == 1 ~ "Signal",
        Word.CRESP == 0 ~ "Noise"
      )
    ),
    subject = as_factor(Subject),
    reactionTime = Word.RT,
    trialNumber = TrialNum
  ) %>%
  select(
    c(
      taskDifficulty,
      subject,
      goNoGo,
      displayType,
      sigDet,
      signalNoise,
      reactionTime
    )
  )
glimpse(eTask)

Observations: 24,750
Variables: 7
$ taskDifficulty <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy...
$ subject        <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ goNoGo         <fct> Go, Go, Go, Go, NoGo, Go, Go, NoGo, Go, Go, Go, Go, ...
$ displayType    <fct> NoFlankers, NoFlankers, Flankers, NoFlankers, NA, No...
$ sigDet         <fct> Hit, Hit, Hit, Hit, CR, Hit, Hit, CR, Hit, Hit, Hit,...
$ signalNoise    <fct> Signal, Signal, Signal, Signal, Noise, Signal, Signa...
$ reactionTime   <int> 625, 425, 729, 537, 0, 457, 417, 0, 336, 384, 344, 5...
```

2.2 MSIT task

2.2.1 Read in

```
msit <-
  read.delim(
    file = "~/Dropbox/StatsBook/Book5/EgoDepletion/MSIT200_EData.txt",
    fileEncoding = "UTF-16"
  )
glimpse(msit)

Observations: 33,000
Variables: 41
$ ExperimentName          <fct> MAS_MSIT_event_200_3_NL, MAS_MSIT_event_200...
$ Subject                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Session                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ Clock.Information       <fct> <?xml version=1.0?>\n<Clock xmlns:dt=urn:sc...
$ DataFile.Basename       <fct> MAS_MSIT_event_200_3_NL-1-1, MAS_MSIT_event...
$ Display.RefreshRate     <dbl> 60.001, 60.001, 60.001, 60.001, 60.001, 60....
```

```
$ ExperimentVersion       <fct> 1.0.0.43, 1.0.0.43, 1.0.0.43, 1.0.0.43, 1.0...
$ Group                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ RandomSeed              <int> -1687224442, -1687224442, -1687224442, -168...
$ RuntimeVersion          <fct> 2.0.10.353, 2.0.10.353, 2.0.10.353, 2.0.10....
$ RuntimeVersionExpected  <fct> 2.0.10.242, 2.0.10.242, 2.0.10.242, 2.0.10....
$ SessionDate             <fct> 03-30-2015, 03-30-2015, 03-30-2015, 03-30-2...
$ SessionStartDateTimeUtc <fct> 30-3-2015 7:16:57, 30-3-2015 7:16:57, 30-3-...
$ SessionTime             <fct> 09:16:57, 09:16:57, 09:16:57, 09:16:57, 09:...
$ StudioVersion           <fct> 2.0.10.147, 2.0.10.147, 2.0.10.147, 2.0.10....
$ Block                   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ content                 <int> 100, 3, 311, 212, 322, 331, 3, 100, 3, 211,...
$ Digits.ACC              <int> 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1...
$ Digits.CRESP            <int> 1, 3, 3, 1, 3, 1, 3, 1, 3, 2, 2, 2, 2, 3, 1...
$ Digits.DurationError    <int> -6, -6, -6, -6, -6, -6, -6, -6, -6, -6, -6,...
$ Digits.OnsetDelay       <int> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6...
$ Digits.OnsetTime        <int> 24021, 27021, 30021, 33021, 36021, 39021, 4...
$ Digits.OnsetToOnsetTime <int> 500, 500, 500, 500, 500, 500, 500, 500, 500...
$ Digits.RESP             <int> 1, 3, 1, 2, 1, 3, 3, 1, 3, 1, 3, 1, 2, 3, 1...
$ Digits.RT               <int> 530, 538, 426, 962, 882, 625, 489, 529, 553...
$ Digits.RTTime           <int> 24551, 27559, 30447, 33983, 36903, 39646, 4...
$ Left                    <int> 1, 0, 3, 2, 3, 3, 0, 1, 0, 2, 1, 2, 0, 0, 1...
$ LeftSize                <int> 28, 20, 28, 28, 20, 20, 20, 28, 20, 20, 20,...
$ Middle                  <int> 0, 0, 1, 1, 2, 3, 0, 0, 0, 1, 1, 3, 2, 0, 0...
$ MiddleSize              <int> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,...
$ Procedure               <fct> ShowStim, ShowStim, ShowStim, ShowStim, Sho...
$ Right                   <int> 0, 3, 1, 2, 2, 1, 3, 0, 3, 1, 2, 3, 0, 3, 0...
$ RightSize               <int> 20, 28, 20, 20, 28, 28, 28, 20, 28, 28, 28,...
$ Running                 <fct> StimList, StimList, StimList, StimList, Sti...
$ StimList                <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ StimList.Cycle          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ StimList.Sample         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ target                  <int> 1, 3, 3, 1, 3, 1, 3, 1, 3, 2, 2, 2, 2, 3, 1...
$ TrialNum                <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ TrialType               <int> 1, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1...
$ TrialTypeC              <fct> C, C, I, I, I, I, C, C, C, I, I, I, C, C, C...
```

2.2.2 Tidy

```
mTask <- as_tibble(msit) %>%
  transmute(
    response = Digits.RESP,
    stimulus = Digits.CRESP,
    sigDet = as_factor(
      case_when(
        stimulus == response ~ "Hit",
        abs(stimulus - response) == 1 ~ "FA1",
        abs(stimulus - response) == 2 ~ "FA2"
      )
    ),
    reactionTime = Digits.RT,
    subject = factor(Subject),
    trialType = as_factor(
      case_when(
        TrialTypeC == "C" ~ "Congruent",
```

```
        TrialTypeC == "I" ~ "Incongruent"
      )
    )
  )
glimpse(mTask)

Observations: 33,000
Variables: 6
$ response     <int> 1, 3, 1, 2, 1, 3, 3, 1, 3, 1, 3, 1, 2, 3, 1, 1, 1, 2, ...
$ stimulus     <int> 1, 3, 3, 1, 3, 1, 3, 1, 3, 2, 2, 2, 2, 3, 1, 3, 2, 1, ...
$ sigDet       <fct> Hit, Hit, FA2, FA1, FA2, FA2, Hit, Hit, Hit, FA1, FA1,...
$ reactionTime <int> 530, 538, 426, 962, 882, 625, 489, 529, 553, 721, 425,...
$ subject      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ trialType    <fct> Congruent, Congruent, Incongruent, Incongruent, Incong...
```

2.3 Subject info and ratings

2.3.1 Read in

```
subs <- read_csv(file = "~/Dropbox/StatsBook/Book5/EgoDepletion/SubjectStatus.csv")
glimpse(subs)

Observations: 165
Variables: 11
$ SubjectID  <dbl> 1, 2, 3, 5, 6, 8, 9, 10, 11, 14, 16, 17, 21, 22, 23, 25,...
$ Task       <chr> "E", "E", "E", "E", "E", "E", "E", "E", "E", "E", "E", "...
$ condition  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
$ gender     <dbl> 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,...
$ age        <dbl> 19, 19, 19, 19, 18, 22, 18, 19, 23, 19, 19, 18, 18, 19, ...
$ lang       <dbl> 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,...
$ langother  <chr> NA, NA, NA, "Duits", NA, NA, NA, NA, NA, NA, NA, NA, NA,...
$ effort     <dbl> 2, 5, 5, 2, 3, 2, 2, 4, 2, 2, 3, 2, 6, 3, 3, 3, 3, 5, 2,...
$ difficult  <dbl> 1, 3, 1, 3, 2, 2, 3, 3, 4, 2, 1, 3, 2, 3, 3, 1, 1, 2, 2,...
$ tired      <dbl> 6, 4, 2, 5, 3, 3, 5, 2, 2, 6, 4, 4, 6, 5, 2, 1, 4, 4, 4,...
$ frustrated <dbl> 1, 2, 1, 1, 1, 2, 1, 1, 3, 4, 2, 2, 4, 2, 2, 1, 1, 1, 4,...
```

2.3.2 Tidy

```
subjectInfo <- subs %>%
  select(-c(langother, condition)) %>%
  rename(
    subject = SubjectID,
    effortRating = effort,
    difficultyRating = difficult,
    fatigueRating = tired,
    firstLanguage = lang,
    frustrationRating = frustrated
  ) %>%
  mutate(
    subject = factor(subject),
    taskDifficulty = factor(
```

```
      case_when(
        Task == "E" ~ "Easy",
        Task == "H" ~ "Hard"
      )
    ),
    gender = factor(
      case_when(
        gender == 1 ~ "Male",
        gender == 2 ~ "Female"
      )
    ),
  ) %>%
  select(
    c(
      taskDifficulty,
      gender,
      subject,
      effortRating,
      difficultyRating,
      frustrationRating,
      fatigueRating
    )
  )
glimpse(subjectInfo)

Observations: 165
Variables: 7
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ gender            <fct> Female, Male, Female, Female, Female, Female, Mal...
$ subject           <fct> 1, 2, 3, 5, 6, 8, 9, 10, 11, 14, 16, 17, 21, 22, ...
$ effortRating      <dbl> 2, 5, 5, 2, 3, 2, 2, 4, 2, 2, 3, 2, 6, 3, 3, 3, 3...
$ difficultyRating  <dbl> 1, 3, 1, 3, 2, 2, 3, 3, 4, 2, 1, 3, 2, 3, 3, 1, 1...
$ frustrationRating <dbl> 1, 2, 1, 1, 1, 2, 1, 1, 3, 4, 2, 2, 4, 2, 2, 1, 1...
$ fatigueRating     <dbl> 6, 4, 2, 5, 3, 3, 5, 2, 2, 6, 4, 4, 6, 5, 2, 1, 4...
```

3 Preliminary description

Contents

3.1 eTask data

3.1.1 Disregarding subjects

RT

Transform RT? Now we would like to get a sense of the mean RTs in these conditions. This cannot be done reasonably unless we assume that these are distributed symmetrically (and preferably normally). But the distribution of RTs is meaningful *only relative to a model.* So we model these data with a linear model. (By doing so, we are — incorrectly — treating the data as if they came from one subject.)

```
eTaskH.FA <- eTask %>%
  filter(sigDet != "CR")
lmETaskH.FA <- lm(reactionTime ~ taskDifficulty * sigDet, data = eTaskH.FA)
```

Rather than examining the distribution of the residuals from `lmETaskHit`, we look for the value of λ (the power of the transformation) that will maximize the MLE of the model.

```
library(car)
summary(powerTransform(lmETaskH.FA))

bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1   -0.0595       -0.06      -0.0797      -0.0393

Likelihood ratio test that transformation parameter is equal to 0
 (log transformation)
                           LRT df       pval
LR test, lambda = (0) 31.94887  1 1.5828e-08

Likelihood ratio test that no transformation is needed
                          LRT df       pval
LR test, lambda = (1) 7376.75  1 < 2.22e-16
```

So, should we transform `reactionTime`? Yes. Since $\lambda = -0.06 \approx 0$, and in the ladder of powers $\lambda = 0$ means $y \to \log y$. We can also show this graphically (Figure 1):

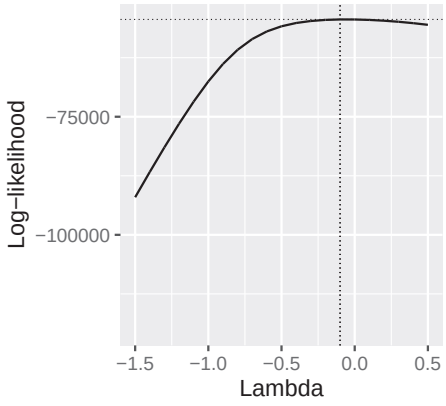

Figure 1: Plot profile log-likelihoods for the parameter of the Box-Cox power family for `lmETaskHit`.

```
library(lindia)
gg_boxcox(lmETaskH.FA)
```

So we transform `reactionTime`, and rerun the `lm()`:

```
eTaskH.FA <- eTaskH.FA %>%
  mutate(reactionTimeLog = log(reactionTime))
lmeTaskH.FA.Log <- update(object = lmETaskH.FA, formula. = reactionTimeLog ~ .)
```

(Note that we use `update()`.)

```
AIC(lmETaskH.FA, lmeTaskH.FA.Log) %>%
  as_tibble() %>%
  kable(digits = 0, booktabs = TRUE) %>%
  kable_styling(position = "center")
```

df	AIC
5	210791
5	6901

We made a point *not* to look at the linear models before determining the difference in AIC, and finding that `lmeTaskHitLog` is overwhelmingly better than `lmETaskHit`. We do not have to do this again when we run LME models on these data. (Nevertheless, it's always prudent to double-check by using `powerTransform()` on our final model, and confirming that $\lambda \approx 1$, which means that the fit of the model cannot be improved by transforming the response variable.)

Summarize Detection & RT

```
eTaskSum <- eTask %>%
  group_by(taskDifficulty, signalNoise, sigDet) %>%
  mutate(reactionTimeLog = log(reactionTime)) %>%
  summarize(meanRTlog = mean(reactionTimeLog), n = n()) %>%
  mutate(prop = n / sum(n), geomMean.eTask.RT = exp(meanRTlog)) %>%
  filter(sigDet != "CR") %>%
  select(-c(n, meanRTlog)) %>%
  select(taskDifficulty, sigDet, prop, geomMean.eTask.RT)
eTaskSum %>%
  kable(digits = c(0, 0, 0, 3, 0), booktabs = TRUE) %>%
  kable_styling(position = "center")
```

signalNoise	taskDifficulty	sigDet	prop	geomMean.eTask.RT
Signal	Easy	Hit	1.000	542
Noise	Easy	FA	0.064	450
Signal	Hard	Hit	1.000	1080
Noise	Hard	FA	0.138	1072

Clearly none of the subjects made errors when they responded to a word that contained `e`, but made more than twice as many inhibition errors in the `Hard` condition (13.8%) than in the `Easy` condition (6.4%).

Now we do this by subject:

```
eTaskBySub <- eTask %>%
  group_by(taskDifficulty, subject, signalNoise, sigDet) %>%
  mutate(reactionTimeLog = log(reactionTime)) %>%
  summarize(meanRTlog = mean(reactionTimeLog), n = n()) %>%
  mutate(prop = n / sum(n), geomMean.eTask.RT = exp(meanRTlog)) %>%
  filter(sigDet != "CR") %>%
  select(-c(n, meanRTlog)) %>%
  select(taskDifficulty, subject, sigDet, prop, geomMean.eTask.RT)
glimpse(eTaskBySub)

Observations: 308
Variables: 6
Groups: taskDifficulty, subject, signalNoise [308]
$ signalNoise       <fct> Signal, Noise, Signal, Signal, Noise, Signal, Noi...
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ subject           <fct> 1, 1, 2, 3, 3, 5, 5, 6, 8, 8, 9, 9, 10, 10, 11, 1...
$ sigDet            <fct> Hit, FA, Hit, Hit, FA, Hit, FA, Hit, Hit, FA, Hit...
$ prop              <dbl> 1.00000000, 0.03225806, 1.00000000, 1.00000000, 0...
$ geomMean.eTask.RT <dbl> 520.2102, 488.0000, 572.2292, 514.5156, 398.8283,...
```

3.1.2 Join tibbles

Now let's join the `eTaskBySub`, and `subjectInfo` tibbles by a common column, `subject`. We do this using the function `full_join()`

```
eTaskSub <- subjectInfo %>%
  full_join(eTaskBySub)
dim(eTaskSub)

[1] 308  11

eTaskSub <- eTaskSub %>%
  drop_na()
dim(eTaskSub)

[1] 302  11

glimpse(eTaskSub)

Observations: 302
Variables: 11
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ gender            <fct> Female, Female, Male, Female, Female, Female, Fem...
$ subject           <fct> 1, 1, 2, 3, 3, 5, 5, 6, 8, 8, 9, 9, 10, 10, 11, 1...
$ effortRating      <dbl> 2, 2, 5, 5, 5, 2, 2, 3, 2, 2, 2, 2, 4, 4, 2, 2, 2...
$ difficultyRating  <dbl> 1, 1, 3, 1, 1, 3, 3, 2, 2, 2, 3, 3, 3, 3, 4, 4, 2...
$ frustrationRating <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 3, 3, 4...
$ fatigueRating     <dbl> 6, 6, 4, 2, 2, 5, 5, 3, 3, 3, 5, 5, 2, 2, 2, 2, 6...
$ signalNoise       <fct> Signal, Noise, Signal, Signal, Noise, Signal, Noi...
$ sigDet            <fct> Hit, FA, Hit, Hit, FA, Hit, FA, Hit, Hit, FA, Hit...
$ prop              <dbl> 1.00000000, 0.03225806, 1.00000000, 1.00000000, 0...
$ geomMean.eTask.RT <dbl> 520.2102, 488.0000, 572.2292, 514.5156, 398.8283,...
```

3.1.3 A quick look at the e task data

Effect of task difficulty on ratings (Figure 2)

```
eTaskSub <- eTaskSub %>%
  mutate(rating = effortRating + difficultyRating + frustrationRating)
ggplot(data = eTaskSub, aes(
  x = taskDifficulty,
  y = rating,
  group = taskDifficulty
)) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
  labs(x = "task difficulty", y = "composite rating")
```

Effect of task difficulty on false-alarm rate (Figure 3)

```
eTaskSubFA <- eTaskSub %>%
  filter(sigDet == "FA")
ggplot(
  data = eTaskSubFA,
  aes(
    x = taskDifficulty,
    y = prop,
    group = taskDifficulty
  )
```

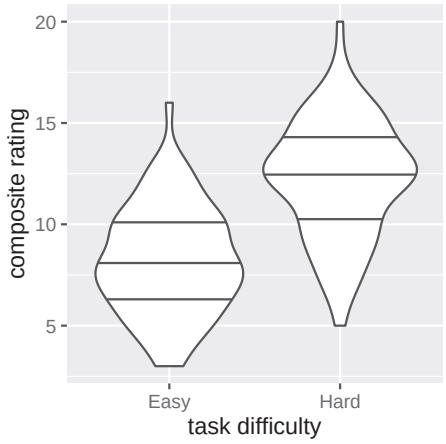

Figure 2: Violin plots of the effort + difficulty + frustration ratings, as a function of task difficulty.

```
) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
  labs(x = "task difficulty", y = "false-alarm rate")
```

Effect of task difficulty on RT (Figure 4)

```
ggplot(
  data = eTaskSub,
  aes(
    x = taskDifficulty,
    y = geomMean.eTask.RT,
    color = sigDet
  )
) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
  labs(x = "task difficulty", y = "RT (geometric mean)") +
  theme(
    legend.position = c(0.05, 0.95),
    legend.justification = c(0.05, 0.95)
  )
```

3.2 MSIT task data

3.2.1 Disregarding subjects

RT Transform RT?

```
lmMTask <- lm(reactionTime ~ sigDet * trialType, data = mTask)
```

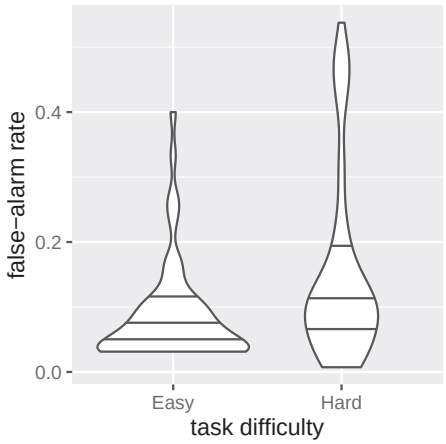

Figure 3: Violin plots of the false-alarm rate, as a function of task difficulty.

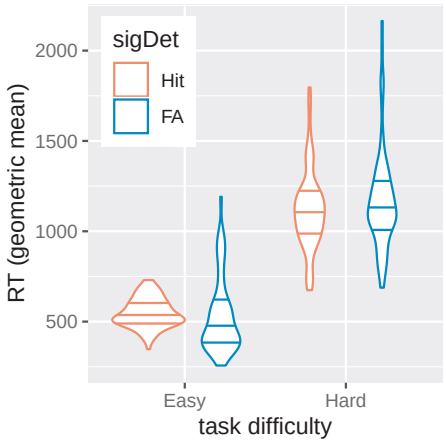

Figure 4: Violin plots of geometric mean RT, as a function of task difficulty and signal detection.

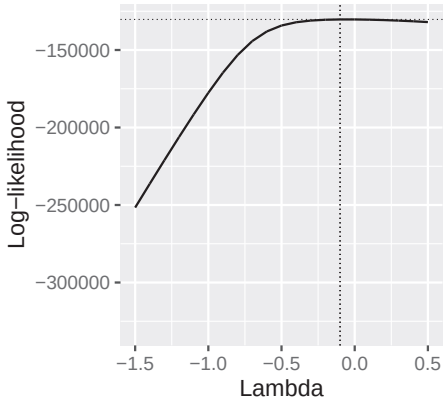

Figure 5: Plot profile log-likelihoods for the parameter of the Box-Cox power family for `lmMTask`.

```
lambda.mTask <- powerTransform(lmMTask) %>%
  print()

Estimated transformation parameter
         Y1
-0.05513577
```

We should transform `reactionTime` because $\lambda = -0.06 \approx 0$. We also show this graphically in Figure 5:

```
gg_boxcox(lmMTask)
```

Transform RT, and compare models.

```
mTask <- mTask %>%
  mutate(reactionTimeLog = log(reactionTime))
lmMTaskLog <- lm(reactionTimeLog ~ sigDet * trialType, data = mTask)
AIC(lmMTask, lmMTaskLog) %>%
  as_tibble() %>%
  kable(digits = 0, booktabs = TRUE) %>%
  kable_styling(position = "center")
```

df	AIC
7	449639
7	13896

Here too, there is no doubt that taking the log is the right thing to do.

Measures of accuracy Measures of association for ordinal factors (`stimulus`, `response`) in a two-way table:

```
mTaskTab <- with(mTask, table(response, stimulus))
library(questionr)
cv <- cramer.v(mTaskTab)
library(DescTools)
gkg <- GoodmanKruskalGamma(mTaskTab)
gkt <- GoodmanKruskalTau(mTaskTab)
l <- Lambda(mTaskTab)
tribble(
  ~measure, ~value,
  "Cramer's V", cv,
  "Goodman-Kruskal Gamma", gkg,
  "Goodman-Kruskal Tau", gkt,
  "Lambda", l
) %>%
  kable(digits = 2, booktabs = TRUE) %>%
  kable_styling(position = "center")
```

measure	value
Cramer's V	0.82
Goodman-Kruskal Gamma	0.92
Goodman-Kruskal Tau	0.67
Lambda	0.81

These measures represent how well the subjects responded to each stimulus with the corresponding correct response.

Using *Lambda* (as an example) we summarize accuracy as a function of trial type and signal detection category:

```
library(questionr)
mTaskC <- mTask %>%
  filter(trialType == "Congruent") %>%
  select(stimulus, response) %>%
  table() %>%
  cramer.v() %>%
  print()

[1] 0.9823959

mTaskI <- mTask %>%
  filter(trialType == "Incongruent") %>%
  select(stimulus, response) %>%
  table() %>%
  cramer.v() %>%
  print()

[1] 0.6559378
```

```
mTaskCramerS <- mTask %>%
  group_by(trialType, subject) %>%
  nest() %>%
  mutate(cv = map(data, ~
  select(.x, c(stimulus, response)) %>%
    table() %>%
    cramer.v() %>%
    as.list() %>%
    as_tibble(.name_repair = ~ c("Cramer_V")))) %>%
  select(-data) %>%
  unnest(cv) %>%
  ungroup()
glimpse(mTaskCramerS)

Observations: 330
Variables: 3
$ subject   <fct> 1, 1, 10, 10, 100, 100, 101, 101, 102, 102, 103, 103, 104...
$ trialType <fct> Congruent, Incongruent, Congruent, Incongruent, Congruent...
$ Cramer_V  <dbl> 0.98373875, 0.38283932, 1.00000000, 0.72519159, 1.0000000...
```

Effect of trial type on accuracy (Figure 6)

```
mTaskCramerS <- mTaskCramerS %>%
  spread(key = trialType, value = Cramer_V) %>%
  mutate(diff = Congruent - Incongruent)
mTaskDiff <- tibble(bySubject = mTaskCramerS$diff)
ggplot(data = mTaskDiff, aes(
  x = "",
  y = bySubject
)) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75)) +
  labs(x = "", y = "difference in Cramer's V")
```

The median difference between Cramer's V for congruent and incongruent trials is 0.18.

3.2.2 Join tibbles

```
eTaskmTaskSub <- eTaskSub %>%
  full_join(mTaskCramerS)
dim(eTaskmTaskSub)

[1] 305  15

eTaskSub <- eTaskSub %>%
  drop_na()
dim(eTaskmTaskSub)

[1] 305  15

glimpse(eTaskmTaskSub)
```

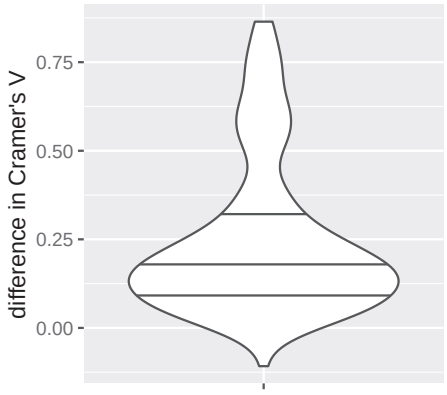

Figure 6: Violin plots of the difference between Cramer's V for congruent and incongruent trials.

```
Observations: 305
Variables: 15
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ gender            <fct> Female, Female, Male, Female, Female, Female, Fem...
$ subject           <fct> 1, 1, 2, 3, 3, 5, 5, 6, 8, 8, 9, 9, 10, 10, 11, 1...
$ effortRating      <dbl> 2, 2, 5, 5, 5, 2, 2, 3, 2, 2, 2, 2, 4, 4, 2, 2, 2...
$ difficultyRating  <dbl> 1, 1, 3, 1, 1, 3, 3, 2, 2, 2, 3, 3, 3, 3, 4, 4, 2...
$ frustrationRating <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 3, 3, 4...
$ fatigueRating     <dbl> 6, 6, 4, 2, 2, 5, 5, 3, 3, 3, 5, 5, 2, 2, 2, 2, 6...
$ signalNoise       <fct> Signal, Noise, Signal, Signal, Noise, Signal, Noi...
$ sigDet            <fct> Hit, FA, Hit, Hit, FA, Hit, FA, Hit, Hit, FA, Hit...
$ prop              <dbl> 1.00000000, 0.03225806, 1.00000000, 1.00000000, 0...
$ geomMean.eTask.RT <dbl> 520.2102, 488.0000, 572.2292, 514.5156, 398.8283,...
$ rating            <dbl> 4, 4, 10, 7, 7, 6, 6, 6, 6, 6, 6, 6, 8, 8, 9, 9, ...
$ Congruent         <dbl> 0.9837388, 0.9837388, 0.9856697, 1.0000000, 1.000...
$ Incongruent       <dbl> 0.3828393, 0.3828393, 0.8083788, 0.7994993, 0.799...
$ diff              <dbl> 0.6008994, 0.6008994, 0.1772909, 0.2005007, 0.200...

eTaskmTaskSubT <- eTaskmTaskSub %>%
  gather(congruence, CramerV, Congruent:Incongruent, factor_key = TRUE)
glimpse(eTaskmTaskSubT)

Observations: 610
Variables: 15
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ gender            <fct> Female, Female, Male, Female, Female, Female, Fem...
$ subject           <fct> 1, 1, 2, 3, 3, 5, 5, 6, 8, 8, 9, 9, 10, 10, 11, 1...
$ effortRating      <dbl> 2, 2, 5, 5, 5, 2, 2, 3, 2, 2, 2, 2, 4, 4, 2, 2, 2...
$ difficultyRating  <dbl> 1, 1, 3, 1, 1, 3, 3, 2, 2, 2, 3, 3, 3, 3, 4, 4, 2...
$ frustrationRating <dbl> 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 3, 3, 4...
$ fatigueRating     <dbl> 6, 6, 4, 2, 2, 5, 5, 3, 3, 3, 5, 5, 2, 2, 2, 2, 6...
$ signalNoise       <fct> Signal, Noise, Signal, Signal, Noise, Signal, Noi...
$ sigDet            <fct> Hit, FA, Hit, Hit, FA, Hit, FA, Hit, Hit, FA, Hit...
```

```
$ prop              <dbl> 1.00000000, 0.03225806, 1.00000000, 1.00000000, 0...
$ geomMean.eTask.RT <dbl> 520.2102, 488.0000, 572.2292, 514.5156, 398.8283,...
$ rating            <dbl> 4, 4, 10, 7, 7, 6, 6, 6, 6, 6, 6, 8, 8, 9, 9, ...
$ diff              <dbl> 0.6008994, 0.6008994, 0.1772909, 0.2005007, 0.200...
$ congruence        <fct> Congruent, Congruent, Congruent, Congruent, Congr...
$ CramerV           <dbl> 0.9837388, 0.9837388, 0.9856697, 1.0000000, 1.000...
```

```
eTaskmTaskSubT %>%
  count(taskDifficulty) %>%
  mutate(prop = prop.table(n)) %>%
  mkable()
```

```
Warning:  Factor ‘taskDifficulty‘ contains implicit NA, consider using ‘forcats::fct_explicit_na‘
```

taskDifficulty	n	prop
Easy	278	0.46
Hard	326	0.53
NA	6	0.01

```
eTaskmTaskSubT <- eTaskmTaskSubT %>%
  drop_na()
eTaskmTaskSubT %>%
  count(taskDifficulty) %>%
  mutate(prop = prop.table(n)) %>%
  mkable()
```

taskDifficulty	n	prop
Easy	278	0.46
Hard	326	0.54

```
ggplot(data = eTaskmTaskSubT, aes(
  x = taskDifficulty,
  y = CramerV, color = congruence
)) +
  geom_boxplot() +
  labs(x = "task difficulty", y = "Cramer's V")
```

```
ggplot(data = eTaskmTaskSubT, aes(
  x = rating,
  y = CramerV, color = congruence
)) +
  geom_point() +
  labs(x = "rating", y = "Cramer's V")
```

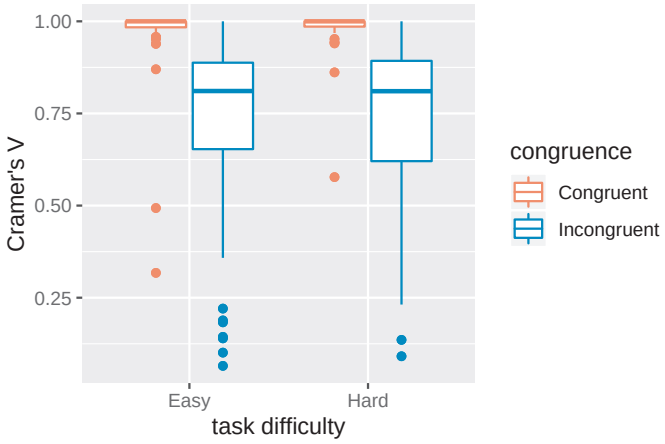

Figure 7: Boxplots of Cramer's V as a function of e task difficulty and congruence in the MSIT task.

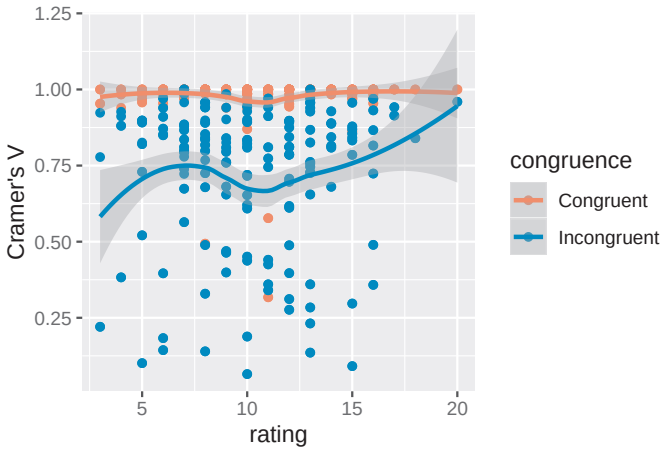

Figure 8: Cramer's V as a function of e rating and congruence in the MSIT task.

```
all <- eTaskmTaskSub %>%
  full_join(mTask)
```

```
Warning: Column ‘sigDet‘ joining factors with different levels, coercing to character
vector
```

```
dim(all)
```

```
[1] 33143    20
```

```
all <- all %>%
  drop_na()
dim(all)
```

```
[1] 28219    20
```

```
glimpse(all)
```

```
Observations: 28,219
Variables: 20
$ taskDifficulty    <fct> Easy, Easy, Easy, Easy, Easy, Easy, Easy, Easy, E...
$ gender            <fct> Female, Female, Female, Female, Female, Female, F...
$ subject           <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ effortRating      <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2...
$ difficultyRating  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ frustrationRating <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ fatigueRating     <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6...
$ signalNoise       <fct> Signal, Signal, Signal, Signal, Signal, Signal, S...
$ sigDet            <chr> "Hit", "Hit", "Hit", "Hit", "Hit", "Hit", "Hit", ...
$ prop              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ geomMean.eTask.RT <dbl> 520.2102, 520.2102, 520.2102, 520.2102, 520.2102,...
$ rating            <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4...
$ Congruent         <dbl> 0.9837388, 0.9837388, 0.9837388, 0.9837388, 0.983...
$ Incongruent       <dbl> 0.3828393, 0.3828393, 0.3828393, 0.3828393, 0.382...
$ diff              <dbl> 0.6008994, 0.6008994, 0.6008994, 0.6008994, 0.600...
$ response          <int> 1, 3, 3, 1, 3, 2, 3, 1, 1, 2, 3, 1, 2, 2, 1, 3, 1...
$ stimulus          <int> 1, 3, 3, 1, 3, 2, 3, 1, 1, 2, 3, 1, 2, 2, 1, 3, 1...
$ reactionTime      <int> 530, 538, 489, 529, 553, 593, 457, 417, 449, 762,...
$ trialType         <fct> Congruent, Congruent, Congruent, Congruent, Congr...
$ reactionTimeLog   <dbl> 6.272877, 6.287859, 6.192362, 6.270988, 6.315358,...
```

```
ggplot(data = all, aes(
  x = taskDifficulty,
  y = reactionTimeLog, color = trialType
)) +
  geom_violin() +
  labs(x = "e task difficult", y = "m task log(RT)")
```

3.3 Linear mixed-effects model

```
lmeMTaskLog <-
  lmer(reactionTimeLog ~ sigDet * trialType +
    (1 | subject), data = mTask)
```

Figure 9: Log(RT) as a function of e task difficulty and congruence in the MSIT task.

More complex random effects

```
lmeMTaskLog2 <-
  lmer(reactionTimeLog ~ sigDet * trialType +
    (sigDet * trialType | subject),
  data = mTask
  )
```

```
Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  Model
failed to converge with max|grad| = 0.117414 (tol = 0.002, component 1)
```

```
AIC(lmeMTaskLog, lmeMTaskLog2) %>%
  as_tibble() %>%
  kable(digits = 0, booktabs = TRUE) %>%
  kable_styling(position = "center")
```

df	AIC
8	131
28	-1814

```
tidy(lmeMTaskLog2, effects = "ran_pars") %>%
  mkables()
```

effect	group	term	estimate
ran_pars	subject	sd__(Intercept)	0.19
ran_pars	subject	sd__sigDetFA2	0.13
ran_pars	subject	sd__sigDetFA1	0.33
ran_pars	subject	sd__trialTypeIncongruent	0.13
ran_pars	subject	sd__sigDetFA2:trialTypeIncongruent	0.27
ran_pars	subject	sd__sigDetFA1:trialTypeIncongruent	0.33
ran_pars	subject	cor__(Intercept).sigDetFA2	0.40
ran_pars	subject	cor__(Intercept).sigDetFA1	0.18
ran_pars	subject	cor__(Intercept).trialTypeIncongruent	-0.16
ran_pars	subject	cor__(Intercept).sigDetFA2:trialTypeIncongruent	-0.12
ran_pars	subject	cor__(Intercept).sigDetFA1:trialTypeIncongruent	-0.14
ran_pars	subject	cor__sigDetFA2.sigDetFA1	0.05
ran_pars	subject	cor__sigDetFA2.trialTypeIncongruent	0.52
ran_pars	subject	cor__sigDetFA2.sigDetFA2:trialTypeIncongruent	-0.19
ran_pars	subject	cor__sigDetFA2.sigDetFA1:trialTypeIncongruent	0.08
ran_pars	subject	cor__sigDetFA1.trialTypeIncongruent	-0.14
ran_pars	subject	cor__sigDetFA1.sigDetFA2:trialTypeIncongruent	0.14
ran_pars	subject	cor__sigDetFA1.sigDetFA1:trialTypeIncongruent	-0.93
ran_pars	subject	cor__trialTypeIncongruent.sigDetFA2:trialTypeIncongruent	-0.61
ran_pars	subject	cor__trialTypeIncongruent.sigDetFA1:trialTypeIncongruent	-0.02
ran_pars	subject	cor__sigDetFA2:trialTypeIncongruent.sigDetFA1:trialTypeIncongruent	-0.05
ran_pars	Residual	sd__Observation	0.23

```
library(brms)
library(dotwhisker)
tidy(lmeMTaskLog2, effects = "fixed") %>%
  dwplot(vline = geom_vline(
    xintercept = 0,
    colour = "grey50",
    linetype = 2
  ))
```

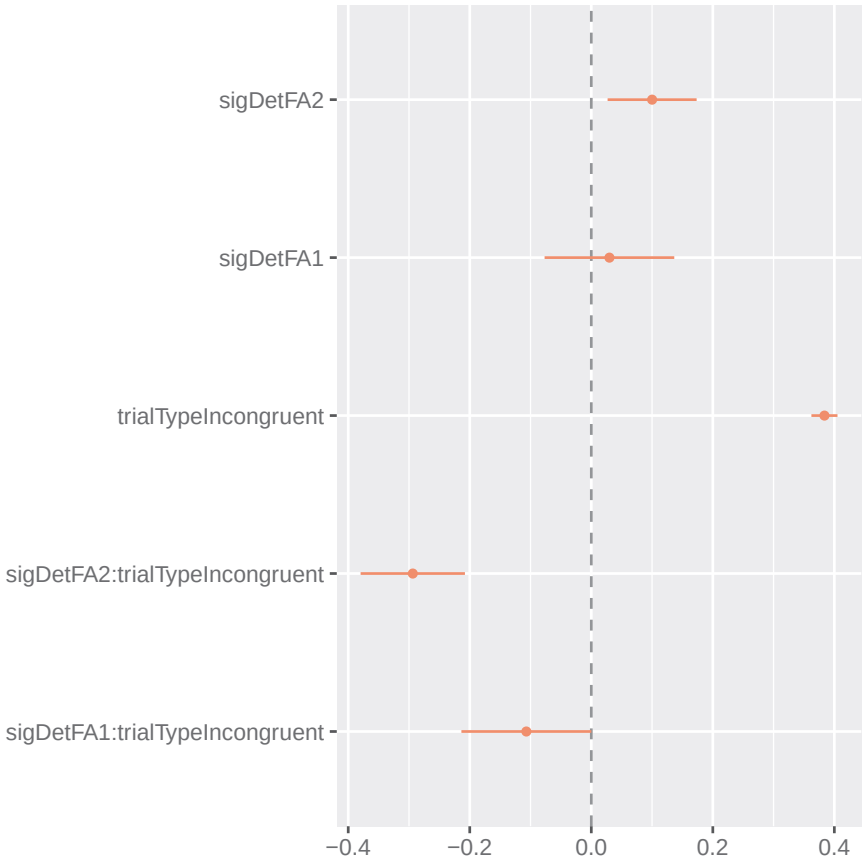

Figure 10: Dot-and-whisker plot of fixed effects of `lmeMTaskLog2` model .

R session information

```
sessionInfo()

R version 3.6.1 (2019-07-05)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Catalina 10.15.2

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] C

attached base packages:
[1] splines   parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] DescTools_0.99.31     questionr_0.7.0       janitor_1.2.0
 [4] xtable_1.8-4          WWGbook_1.0.1         texreg_1.36.23
 [7] styler_1.2.0          stargazer_5.2.2       R.utils_2.9.2
[10] R.oo_1.23.0           R.methodsS3_1.7.1     qqplotr_0.0.3
[13] prob_1.0-1            fAsianOptions_3042.82 fOptions_3042.86
[16] fBasics_3042.89       timeSeries_3042.102   timeDate_3043.102
[19] combinat_0.0-8        printr_0.1            predictmeans_1.0.1
[22] mgcv_1.8-31           nlme_3.1-142          merTools_0.5.0
[25] arm_1.10-1            MASS_7.3-51.4         lme4_1.1-21
[28] Matrix_1.2-18         lintr_2.0.0           lindia_0.9
[31] latex2exp_0.4.0       languageR_1.5.0       kableExtra_1.1.0
[34] haven_2.2.0           gridExtra_2.3         ggfortify_0.4.8
[37] formattable_0.2.0.1   foreign_0.8-72        foreach_1.4.7
[40] extrafontdb_1.0       extrafont_0.17        effects_4.1-4
[43] dotwhisker_0.5.0      car_3.0-5             carData_3.0-3
[46] brms_2.10.0           Rcpp_1.0.3            broom.mixed_0.2.4
[49] yardstick_0.0.4       rsample_0.0.5         recipes_0.1.7
[52] parsnip_0.0.4         infer_0.5.1           dials_0.0.4
[55] scales_1.1.0          broom_0.5.2           tidymodels_0.0.3
[58] forcats_0.4.0         stringr_1.4.0         dplyr_0.8.3
[61] purrr_0.3.3           readr_1.3.1           tidyr_1.0.0
[64] tibble_2.1.3          ggplot2_3.2.1         tidyverse_1.3.0
[67] knitr_1.26

loaded via a namespace (and not attached):
  [1] SnowballC_0.6.0      coda_0.19-3          dygraphs_1.1.1.6
  [4] data.table_1.12.6    rpart_4.1-15         inline_0.3.15
  [7] generics_0.0.2       callr_3.3.2          future_1.15.1
 [10] tokenizers_0.2.1     webshot_0.5.2        xml2_1.2.2
 [13] lubridate_1.7.4      httpuv_1.5.2         StanHeaders_2.19.0
 [16] assertthat_0.2.1     gower_0.2.1          xfun_0.11
 [19] hms_0.5.2            bayesplot_1.7.1      evaluate_0.14
 [22] promises_1.1.0       fansi_0.4.0          DEoptimR_1.0-8
 [25] dbplyr_1.4.2         readxl_1.3.1         igraph_1.2.4.2
 [28] DBI_1.0.0            htmlwidgets_1.5.1    stats4_3.6.1
 [31] ellipsis_0.3.0       crosstalk_1.0.0      backports_1.1.5
```

```
 [34] survey_3.36          markdown_1.1         vctrs_0.2.0
 [37] remotes_2.1.0        abind_1.4-5          withr_2.1.2
 [40] robustbase_0.93-5    xts_0.11-2           prettyunits_1.0.2
 [43] cyclocomp_1.1.0      lazyeval_0.2.2       crayon_1.3.4
 [46] labeling_0.3         pkgconfig_2.0.3      blme_1.0-4
 [49] nnet_7.3-12          rlang_0.4.2          spatial_7.3-11
 [52] globals_0.12.5       lifecycle_0.1.0      miniUI_0.1.1.1
 [55] colourpicker_1.0     rex_1.1.2            modelr_0.1.5
 [58] tidytext_0.2.2       cellranger_1.1.0     rprojroot_1.3-2
 [61] matrixStats_0.55.0   loo_2.1.0            boot_1.3-23
 [64] zoo_1.8-6            reprex_0.3.0         base64enc_0.1-3
 [67] ggridges_0.5.1       processx_3.4.1       viridisLite_0.3.0
 [70] pROC_1.15.3          shinystan_2.5.0      magrittr_1.5
 [73] plyr_1.8.4           threejs_0.3.1        compiler_3.6.1
 [76] rstantools_2.0.0     cli_1.1.0            DiceDesign_1.8-1
 [79] listenv_0.8.0        janeaustenr_0.1.5    ps_1.3.0
 [82] TMB_1.7.15           Brobdingnag_1.2-6    tidyselect_0.2.5
 [85] stringi_1.4.3        highr_0.8            mitools_2.4
 [88] bridgesampling_0.7-2 grid_3.6.1           tidypredict_0.4.3
 [91] tools_3.6.1          rio_0.5.16           rstudioapi_0.10
 [94] prodlim_2019.11.13   farver_2.0.1         digest_0.6.23
 [97] shiny_1.4.0          lava_1.6.6           later_1.0.0
[100] httr_1.4.1           rsconnect_0.8.15     ggstance_0.3.3
[103] colorspace_1.4-1     rvest_0.3.5          fs_1.3.1
[106] expm_0.999-4         shinythemes_1.1.2    rstanarm_2.19.2
[109] jsonlite_1.6         nloptr_1.2.1         rstan_2.19.2
[112] zeallot_0.1.0        ipred_0.9-9          R6_2.4.1
[115] pillar_1.4.2         htmltools_0.4.0      mime_0.7
[118] glue_1.3.1           fastmap_1.0.1        minqa_1.2.4
[121] DT_0.10              class_7.3-15         codetools_0.2-16
[124] utf8_1.1.4           pkgbuild_1.0.6       mvtnorm_1.0-11
[127] furrr_0.1.0          lattice_0.20-38      numDeriv_2016.8-1.1
[130] pbkrtest_0.4-7       curl_4.3             gtools_3.8.1
[133] tidyposterior_0.0.2  zip_2.0.4            shinyjs_1.0
[136] openxlsx_4.1.4       Rttf2pt1_1.3.7       survival_3.1-8
[139] rmarkdown_1.18       desc_1.2.0           munsell_0.5.0
[142] iterators_1.0.12     reshape2_1.4.3       gtable_0.3.0
```

```
Sys.time()
```

```
[1] "2019-12-13 20:54:49 EST"
```