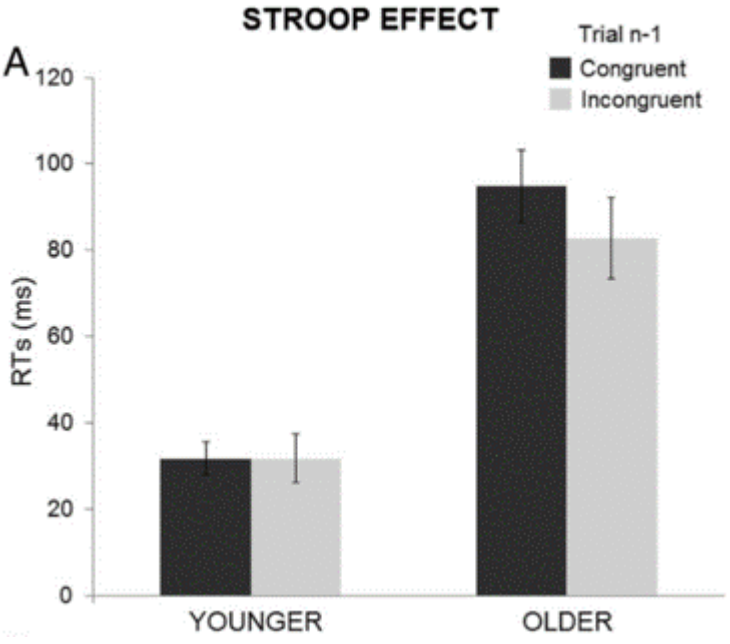
Evaluating Sensitivity of Different Measurements of a Cognitive Priming Task

John F. Adamek

12/5/2021

The Problem

- Response times in healthy young adults (i.e., college students) are very similar
- When people are not too different from one another, the limiting factor is trial variability rather than variability across participants (Rouder & Haaf, 2018).
- Intraindividual variability (IIV) refers to moment-to-moment variations in an individual's performance on a reaction time (RT) task
  - A more sensitive indicator of cognitive functioning compared to traditionally used mean RT (Brydges et al., 2021)



Question:

Analyzing whether IIV is a better way for calculating composite scores for cognitive tests (i.e., Stroop task) compared to traditional calculations in young adults where within subject variability is generally very small.

Process

Generate datasets with different samples sizes and then different variability in reaction time and calculate IVV and traditional scores:

1. Generate raw data for a typical Stroop task on congruent and incongruent conditions
2. Run 10,000 simulations with varying condition trials (20, 40, 60, 80, 100)
3. Repeat the same process with variability in response times (0, .1, .2, .3, .4)

Interindividual Variability – **Protocol**(Bielak & Anstey, 2019; Hultsch et al., 2000)

- Clean Data:
  - Remove Incorrect responses
  - Trim Outliers (Lower bound = .150(150ms), Upper bound = 3SDs)
- Calculation:
  - Remove any systematic within-subject sources of variance (i.e., practice effects):
    - Regress RT ~ trials, resulting residuals are independent of potential confounds from practice effects
  - Residuals converted to standardized T-scores (M = 50, SD = 10)
  - Each individual's SD was calculated as the SD of their T-scores. This is the IIV

Study Parameters

Study Parameters taken from the literature (Rouder & Haaf, 2021; Rouder et al., 2019)

- Trial var = 200ms
- Mean Effect size (congruent - incongruent) = 40ms
  - Congruent and Incongruent mean RT set at 60ms and 80ms
- Var of Effect size = 28ms

```
n <- 100
cond=30
# Correct Responses
corr_c <- 0.98
corr_i <- 0.94

## Congruent parameters
mean_cong <- 0.06
sd_cong <- 0.200

## Incongruent parameters
mean_inc <- 0.08
sd_inc <- 0.200
```

Functions

Generate Data Function:

- Default: 60 trials, 30 congruent + 30 incongruent
- Reaction time for each condition done with rnorm() using parameters above
- Correct responses done with sample() using average percent correct from parameters above
- Clean's data
- Runs t-test between conditions (congruent v. incongruent) based on the different measurements
- Default sample size n = 100

Simulation Function

```
# Simulation Function
generate_data <- function(n, mean_cong, sd_cong, mean_inc, sd_inc, cond){
  # Create a blank dataset
  data <- data.frame(ID = rep(1:n, each = cond*2))
  # Simulate trials
  data <- data %>%
    group_by(ID) %>%
    mutate(Trial = row_number(),
           Congruence = rep(c('cong','incong'),cond),
           Correct = ifelse(Congruence == 'incong',
                            sample(c(1,0), n*2, replace = T, prob = c(corr_i, 1-corr_i)),
                            sample(c(1,0), n*2, replace = T, prob = c(corr_c, 1-corr_c))),
           RT = ifelse(Congruence == 'incong', rnorm(length(Congruence == 'incong'), mean_inc, sd_inc),
                               rnorm(length(Congruence == 'cong'), mean_cong, sd_cong)))
  return(data)
}
```

IIV Function

```
iiv <- function(df){

  # Congruent Condition
  cong <- df %>%
    filter(Congruence == "cong", Correct == 1) %>%
    group_by(ID) %>%
    mutate(UL = mean(RT) + (sd(RT)*3)) %>%
    filter(RT > 0.150 & RT < UL)  # Trim Outliers
  # Calculate IIV
  cong$residual <- lm(RT ~ Trial, data = cong)$residuals
  cong$t_score <- 50 + (10 * scale(cong$residual)) # I WAS NOT ABLE TO FIGURE AN EASY WAY TO CALCULATE
  cong <- cong %>%
    group_by(ID) %>%
    summarize(IIV_Cong = sd(t_score, na.rm = T))


  # Incongruent Condition
```

```
  incong <- df %>%
    filter(Congruence == "incong", Correct == 1) %>%
    group_by(ID) %>%
    mutate(UL = mean(RT) + (sd(RT)*3)) %>%
    filter(RT > 0.150 & RT < UL)  # Trim Outliers
  # Calculate IIV
  incong$residual <- lm(RT ~ Trial, data = incong)$residuals
  incong$t_score <- 50 + (10 * scale(incong$residual)) # I WAS NOT ABLE TO FIGURE AN EASY WAY TO CALCUL
  incong <- incong %>%
    group_by(ID) %>%
    summarize(IIV_Inc = sd(t_score, na.rm = T))

  result <- cong %>%
    left_join(incong, by = 'ID')

  return(result)
}
```

Traditional Measures Function

```
traditional <- function(df){

  # Clean Data producing AVG RT for each condition
  Newdata1 <- df %>%
    filter(Correct == 1) %>%
    select(ID, Congruence, RT) %>%
    pivot_wider(names_from = Congruence,
                values_from = RT,
                values_fn = list(RT = ~ mean(., na.rm=TRUE))) %>%
    mutate(Interference_Index = incong - cong) %>%
    rename(AvgRT_Cong = cong,
           AvgRT_Incong = incong)
  # Clean Data producing AVG SD for each condition
  Newdata2 <- df %>%
    filter(Correct == 1) %>%
    select(ID, Congruence, RT) %>%
    pivot_wider(names_from = Congruence,
                values_from = RT,
                values_fn = list(RT = ~ sd(., na.rm=TRUE))) %>%
    rename(SDRT_Cong = cong,
           SDRT_Incong = incong)
  # Clean Data producing AVG SD Overall
  Newdata3 <- df %>%
    filter(Correct == 1) %>%
    group_by(ID) %>%
    summarize(AvgRT_Ovr = mean(RT, na.rm=TRUE),
              SDRT_Ovr = sd(RT, na.rm=TRUE))

  # Merge
  Final <- Newdata1 %>%
    left_join(Newdata2, by = 'ID') %>%
```

```
    left_join(Newdata3, by = 'ID')

  return(Final)

}
```

Simulation Function:

```
# Function
simtest <- function(sims, sd_cong, sd_inc, n = 100, cond = 30){
  nsims = sims
  df1 <- data.frame(p_mean = integer(),
                    p_iiv = integer(),
                    rt_diff = integer(),
                    ivv_diff = integer(),
                    lbrt = integer(),
                    ubrt = integer(),
                    lbiiv = integer(),
                    ubiiv = integer(),
                    mdrt = integer(),
                    mfiiv = integer())
  for(i in 1:nsims){
  # Generate Dataset THEN runs t-test
  t <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 30)
  #Calculate IIV
  iiv_data <- iiv(t)
  # Calculate Traditional
  trad_data <- traditional(t)
  # Merge
  final <- trad_data %>%
    left_join(iiv_data, by = "ID")
  # Run t-test
  trt <- t.test(final$AvgRT_Cong, final$AvgRT_Incong, paired = TRUE)$p.value
  tivv <- t.test(final$IIV_Cong, final$IIV_Inc, paired = TRUE)$p.value
  rt_diff <- final$AvgRT_Incong - final$AvgRT_Cong
  ivv_diff <- final$IIV_Inc- final$IIV_Cong
  ll <- t.test(final$AvgRT_Cong, final$AvgRT_Incong, paired = TRUE)$conf.int[1]
  ul <- t.test(final$AvgRT_Cong, final$AvgRT_Incong, paired = TRUE)$conf.int[2]
  l <- t.test(final$IIV_Cong, final$IIV_Inc, paired = TRUE)$conf.int[1]
  u <- t.test(final$IIV_Cong, final$IIV_Inc, paired = TRUE)$conf.int[2]
  mdrt <- t.test(final$AvgRT_Cong, final$AvgRT_Incong, paired = TRUE)$estimate
  mfiiv <- t.test(final$IIV_Cong, final$IIV_Inc, paired = TRUE)$estimate

  df1[nrow(df1) + 1,] <- c(trt, tivv, rt_diff, ivv_diff, ll, ul, l, u, mdrt, mfiiv)
  }

  return(df1)
}
```

Different Trials

Trials = 20

```
# Simulation's
sims = 10000
## Trials per condition
cond = 20

## Generate datesets
t1 <- simtest(sims, sd_cong, sd_inc, cond = cond)
```

Trials = 40

```
# Simulation's
sims = 10000
## Trials per condition
cond = 40

## Generate datesets
t2 <- simtest(sims, sd_cong, sd_inc, cond = cond)
```

Trials = 60

```
# Simulation's
sims = 10000
## Trials per condition
cond = 60

## Generate datesets
t3 <- simtest(sims, sd_cong, sd_inc, cond = cond)
```

Trials = 80

```
# Simulation's
sims = 10000
## Trials per condition
cond = 80

## Generate datesets
t4 <- simtest(sims, sd_cong, sd_inc, cond = cond)
```

Trials = 100

```
# Simulation's
sims = 10000
## Trials per condition
cond = 100

## Generate datesets
t5 <- simtest(sims, sd_cong, sd_inc, cond = cond)
```

Different trial variations

Variation adjustment = 0ms

```
# Simulation's
sims = 10000

## Amount to adjust sd
adj <- 0
sd_cong <- sd_cong + adj
sd_inc <- sd_inc + adj

## Generate datesets
v <- simtest(sims, sd_cong, sd_inc)
```

Variation adjustment = 100ms

```
# Simulation's
sims = 10000

## Amount to adjust sd
adj <- .1
sd_cong1 <- sd_cong + adj
sd_inc1 <- sd_inc + adj

## Generate datesets
v1 <- simtest(sims, sd_cong1, sd_inc1)
```

Variation adjustment = 200ms

```
# Simulation's
sims = 10000

## Amount to adjust sd
adj <- .2
sd_cong2 <- sd_cong + adj
sd_inc2 <- sd_inc + adj

## Generate datesets
v2 <- simtest(sims, sd_cong2, sd_inc2)
```

Variation adjustment = .300ms

```
# Simulation's
sims = 10000

## Amount to adjust sd
adj <- .3
sd_cong3 <- sd_cong + adj
sd_inc3 <- sd_inc + adj

## Generate datesets
v3 <- simtest(sims, sd_cong3, sd_inc3)
```

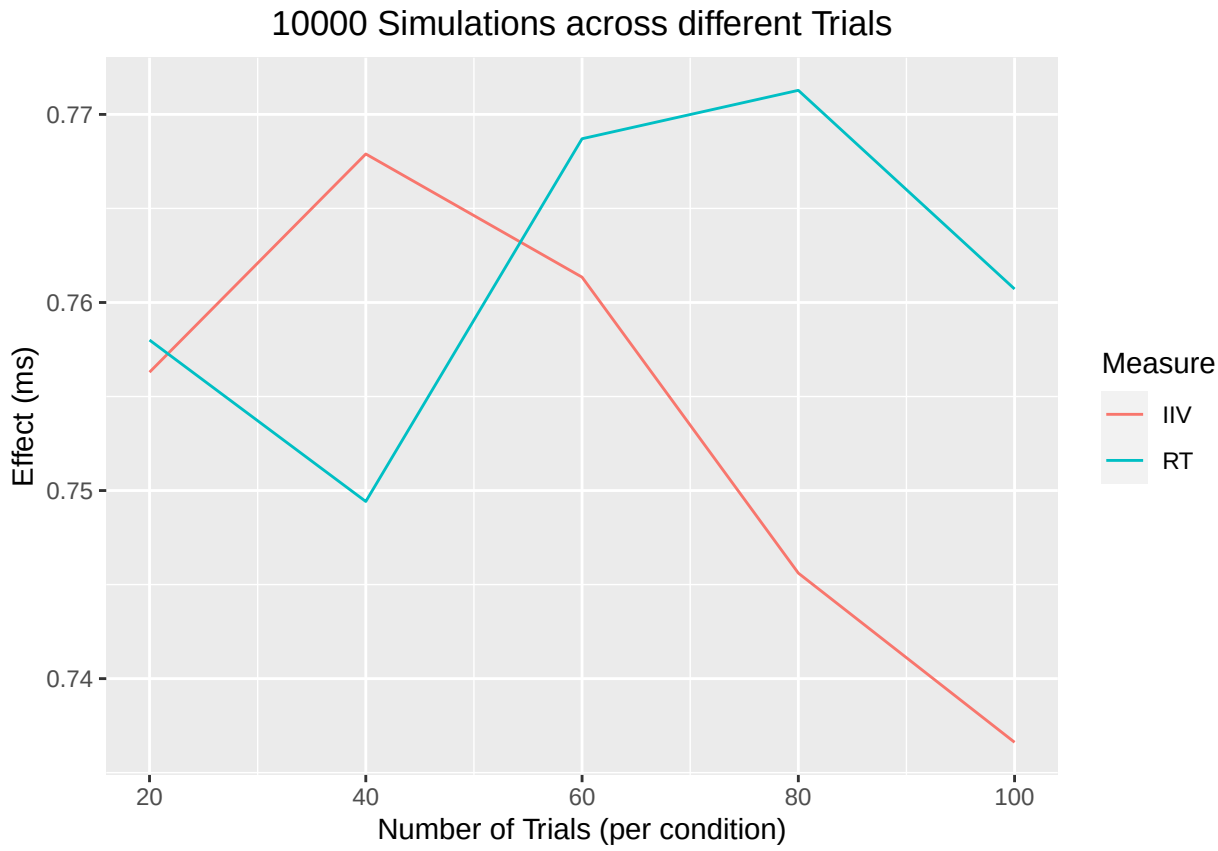
Variation adjustment = .400ms

```
# Simulation's
sims = 10000

## Amount to adjust sd
adj <- .4
sd_cong4 <- sd_cong + adj
sd_inc4 <- sd_inc + adj

## Generate datesets
v4 <- simtest(sims, sd_cong4, sd_inc4)
```

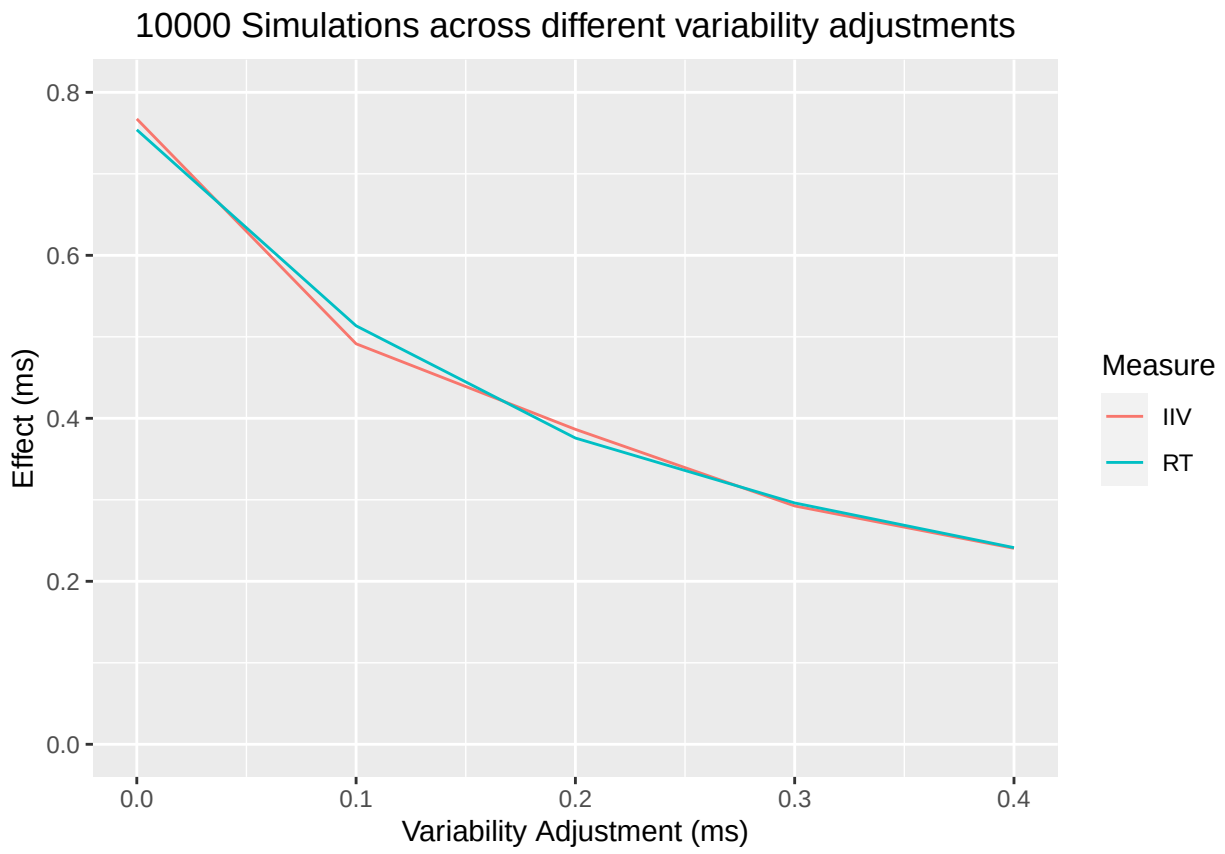
Plot

```
# I saved the environment when running the 10,000 simulations so that I didn't have to keep running it
load('C:/Users/HP/Box/Courses/Fall 2021/F21_PSYC593/FinalProject/projectdata.RData')
```

Trials - effect



Variability - effect



Bayesian Mixed Model (QUID)

With healthy young adults it is expect that they have an effect in the same direction (slower reaction times in the incongruent tasks and slower in congruent) which is considered a positive effect. I used the qualitative individual difference function which tests whether this is true or whether there is qualitative or quantitative variations in their effect (Haaf & Rouder, 2017).

Unconstrained Model: Captures qualitative individual difference

Positive-Effects Model: Captures quantitative individual difference

Common-Effects Model: Captures a lack of individual difference

Generate's dataset

```
# Trials
dat1 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 20)
dat2 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 40)
dat3 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 60)
dat4 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 80)
dat5 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 100)
# Makes condition column a factor so that it is properly ran through function
dat1$cond <- ifelse(dat1$Congruence=='cong',1,2)
dat2$cond <- ifelse(dat2$Congruence=='cong',1,2)
dat3$cond <- ifelse(dat3$Congruence=='cong',1,2)
dat4$cond <- ifelse(dat4$Congruence=='cong',1,2)
```

```
dat5$cond <- ifelse(dat5$Congruence=='cong',1,2)

# Variations
dat6 <- generate_data(n = 100, mean_cong, sd_cong, mean_inc, sd_inc, cond = 30)
dat7 <- generate_data(n = 100, mean_cong, sd_cong+.1, mean_inc, sd_inc+.1, cond = 30)
dat8 <- generate_data(n = 100, mean_cong, sd_cong+.2, mean_inc, sd_inc+.2, cond = 30)
dat9 <- generate_data(n = 100, mean_cong, sd_cong+.3, mean_inc, sd_inc+.3, cond = 30)
dat10 <- generate_data(n = 100, mean_cong, sd_cong+.4, mean_inc, sd_inc+.4, cond = 30)
# Makes condition column a factor so that it is properly ran through function
dat6$cond <- ifelse(dat6$Congruence=='cong',1,2)
dat7$cond <- ifelse(dat7$Congruence=='cong',1,2)
dat8$cond <- ifelse(dat8$Congruence=='cong',1,2)
dat9$cond <- ifelse(dat9$Congruence=='cong',1,2)
dat10$cond <- ifelse(dat10$Congruence=='cong',1,2)
```

Runs QUID Function and Bayes Factors

```
res1 <- quid(id = dat1$ID, condition = dat1$cond, rt = dat1$RT)
res2 <- quid(id = dat2$ID, condition = dat2$cond, rt = dat2$RT)
res3 <- quid(id = dat3$ID, condition = dat3$cond, rt = dat3$RT)
res4 <- quid(id = dat4$ID, condition = dat4$cond, rt = dat4$RT)
res5 <- quid(id = dat5$ID, condition = dat5$cond, rt = dat5$RT)

res6 <- quid(id = dat6$ID, condition = dat6$cond, rt = dat6$RT)
res7 <- quid(id = dat7$ID, condition = dat7$cond, rt = dat7$RT)
res8 <- quid(id = dat8$ID, condition = dat8$cond, rt = dat8$RT)
res9 <- quid(id = dat9$ID, condition = dat9$cond, rt = dat9$RT)
res10 <- quid(id = dat10$ID, condition = dat10$cond, rt = dat10$RT)
```

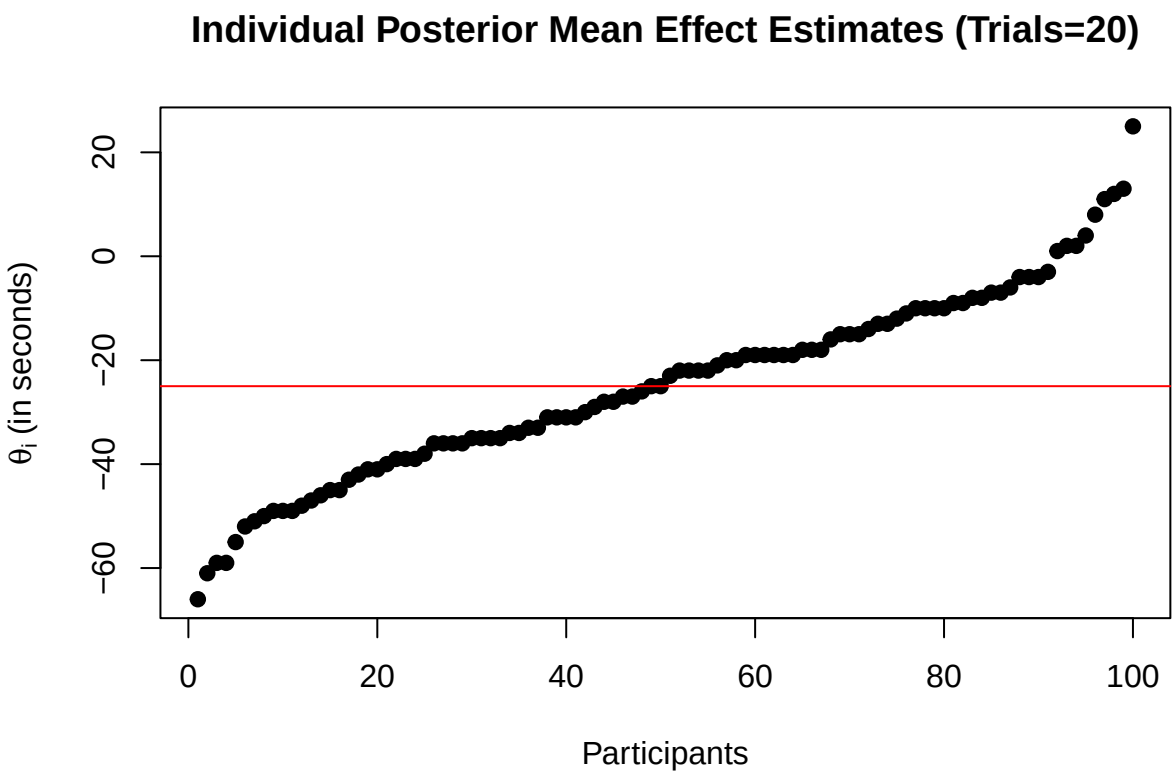
Bayes Factor

- Used to determine which model is preferred to inform plots

```
res1$bfs
res2$bfs
res3$bfs
res4$bfs
res5$bfs

res6$bfs
res7$bfs
res8$bfs
res9$bfs
res10$bfs
```

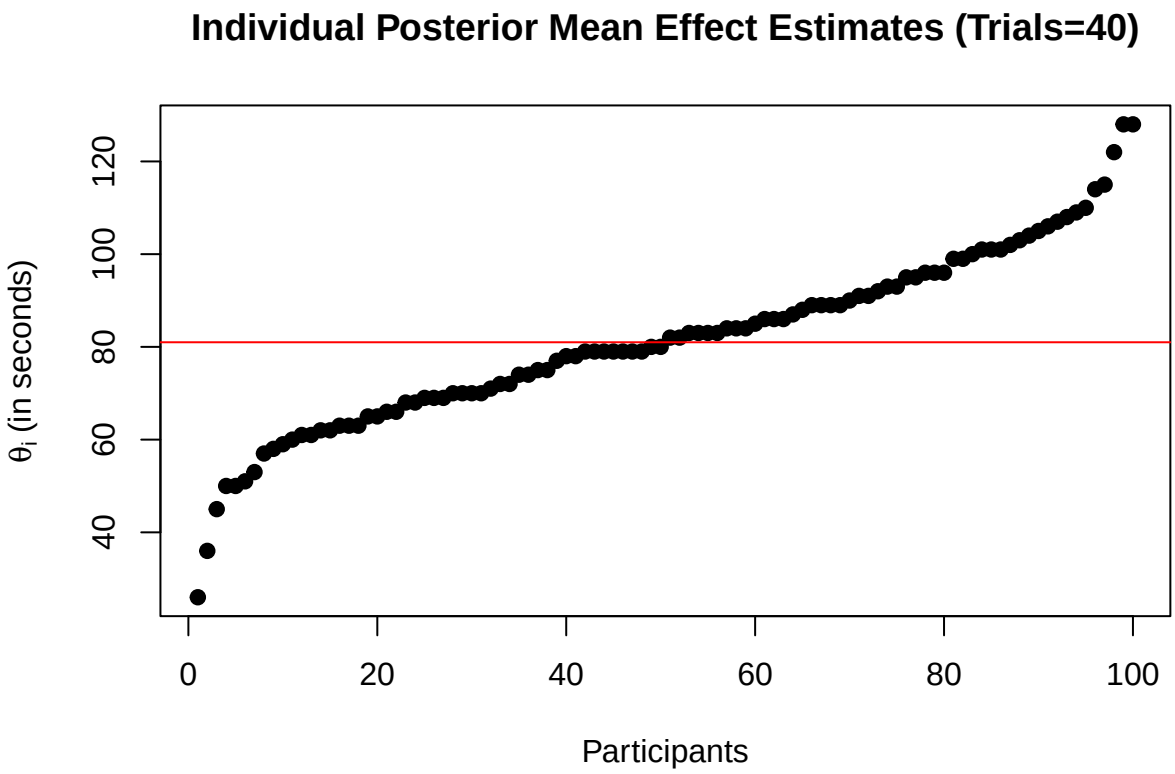
Individual Effects

Trials = 20



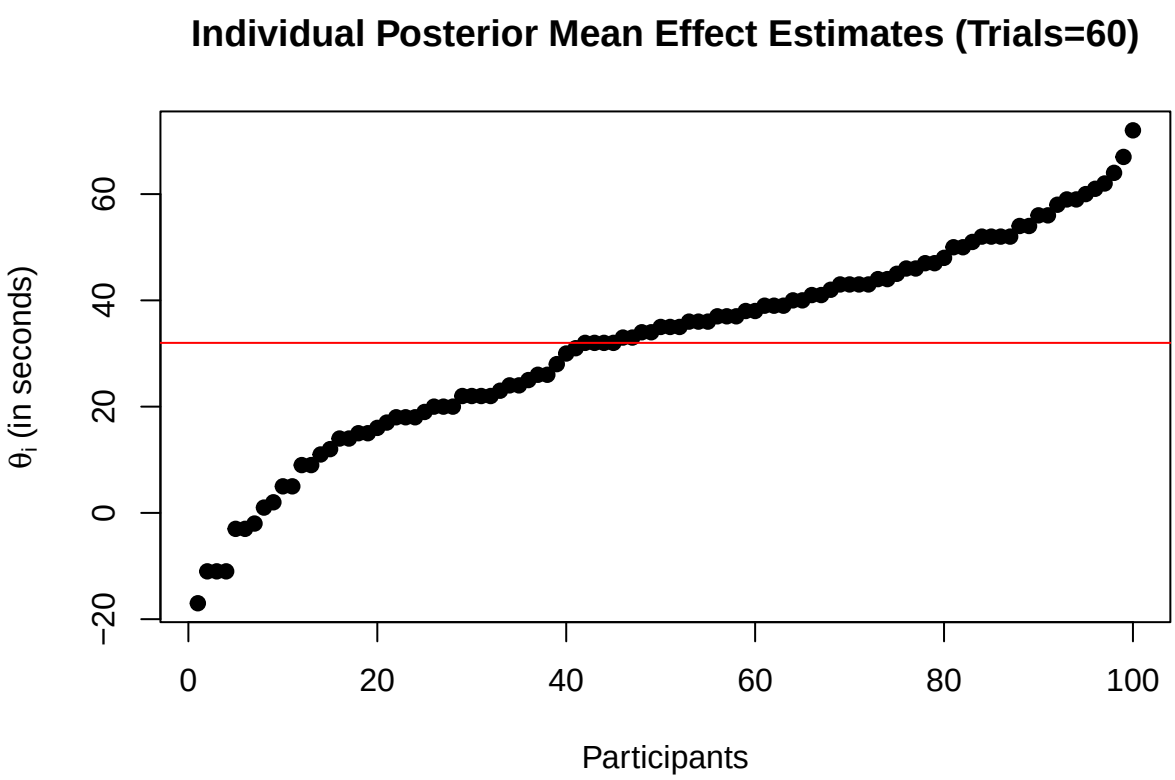
The estimated overall effect is -25 ms, and individual effects vary with a standard deviation of 18 ms around that mean. The unconstrained model is preferred with a Bayes factor of 32.52 indicating evidence that there are qualitative individual differences.

Trial = 40



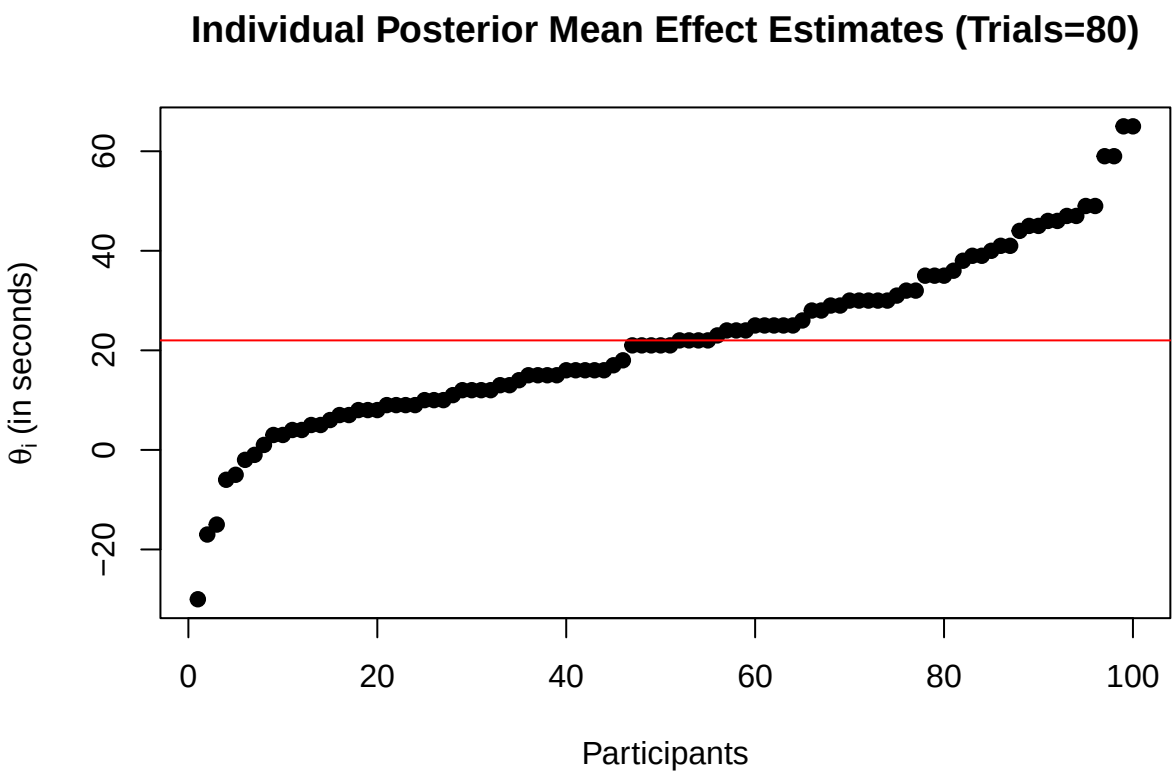
The estimated overall effect is 81 ms, and individual effects vary with a standard deviation of 19 ms around that mean. The common-effects model is preferred with a Bayes factor of 8.1571102×10^{8} indicating evidence for a lack of individual differences.

Trial = 60



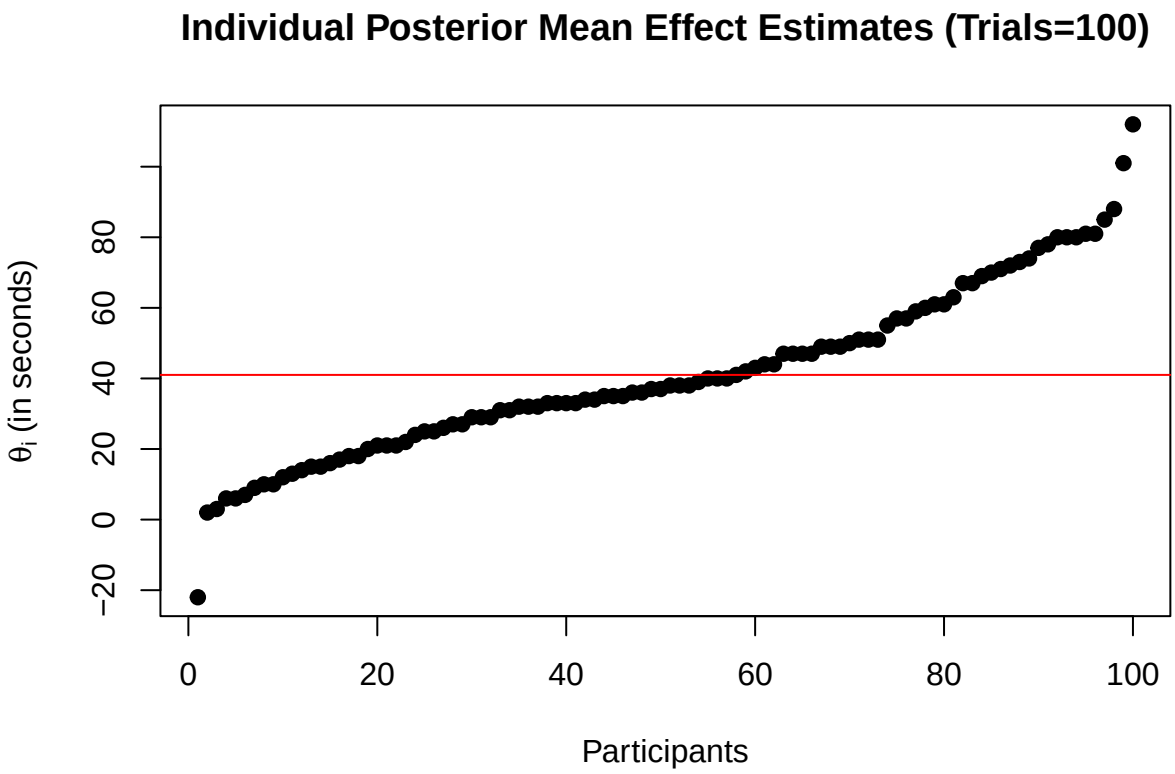
The estimated overall effect is 32 ms, and individual effects vary with a standard deviation of 19 ms around that mean. The unconstrained model is preferred with a Bayes factor of 16.93 indicating evidence that there are qualitative individual differences.

Trial = 80



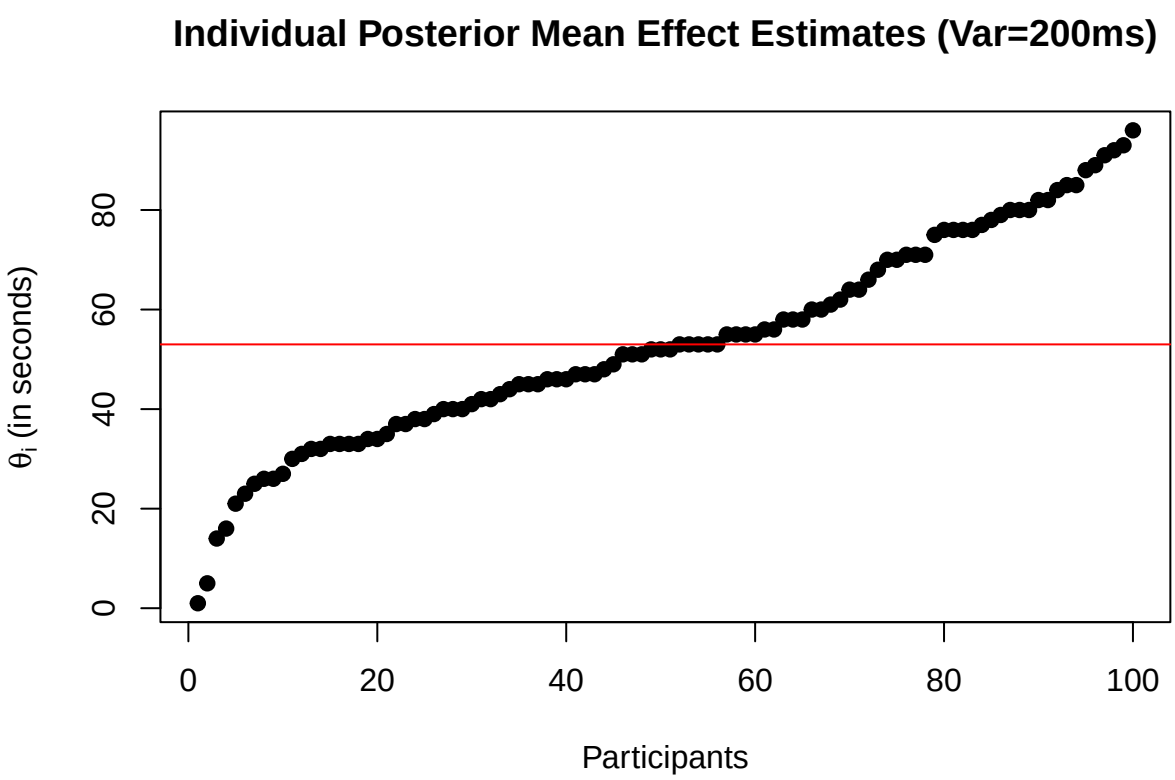
The estimated overall effect is 22 ms, and individual effects vary with a standard deviation of 17 ms around that mean. The unconstrained model is preferred with a Bayes factor of 1.8998147×10^{17} indicating evidence that there are qualitative individual differences.

Trial = 100



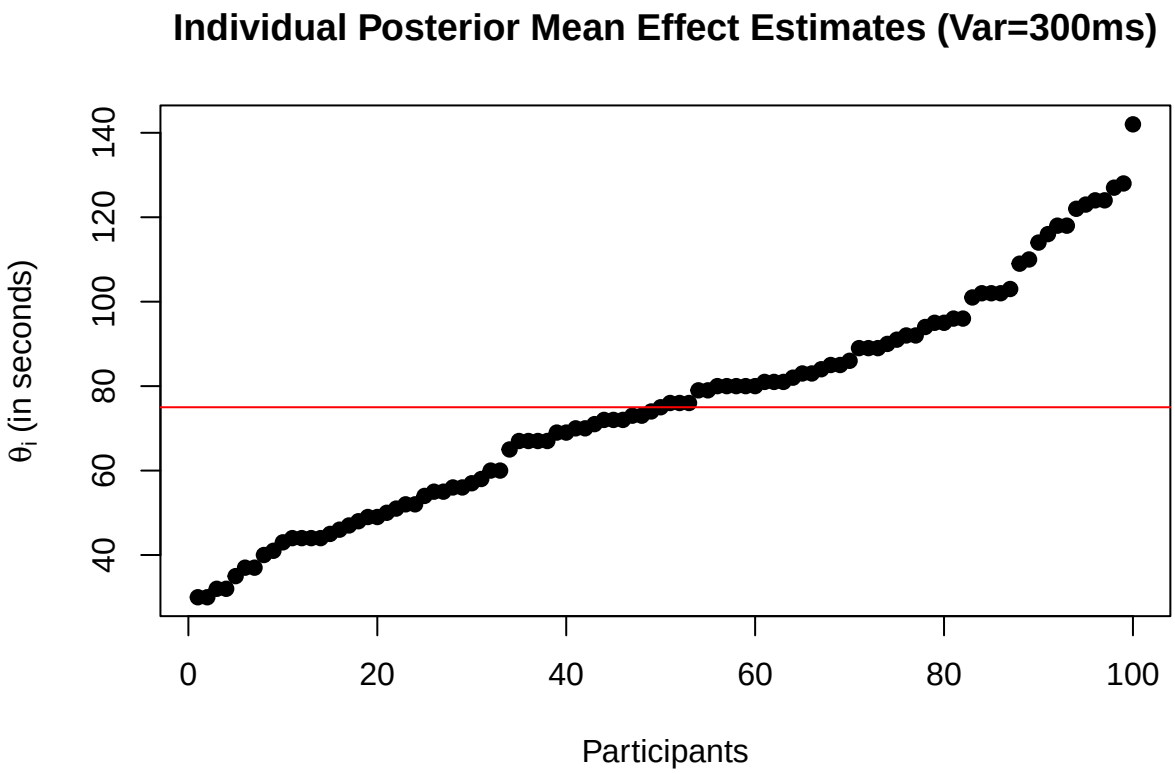
The estimated overall effect is 41 ms, and individual effects vary with a standard deviation of 24 ms around that mean. The common-effects model is preferred with a Bayes factor of 9.72 indicating evidence for a lack of individual differences.

Variation adjustment = 0ms



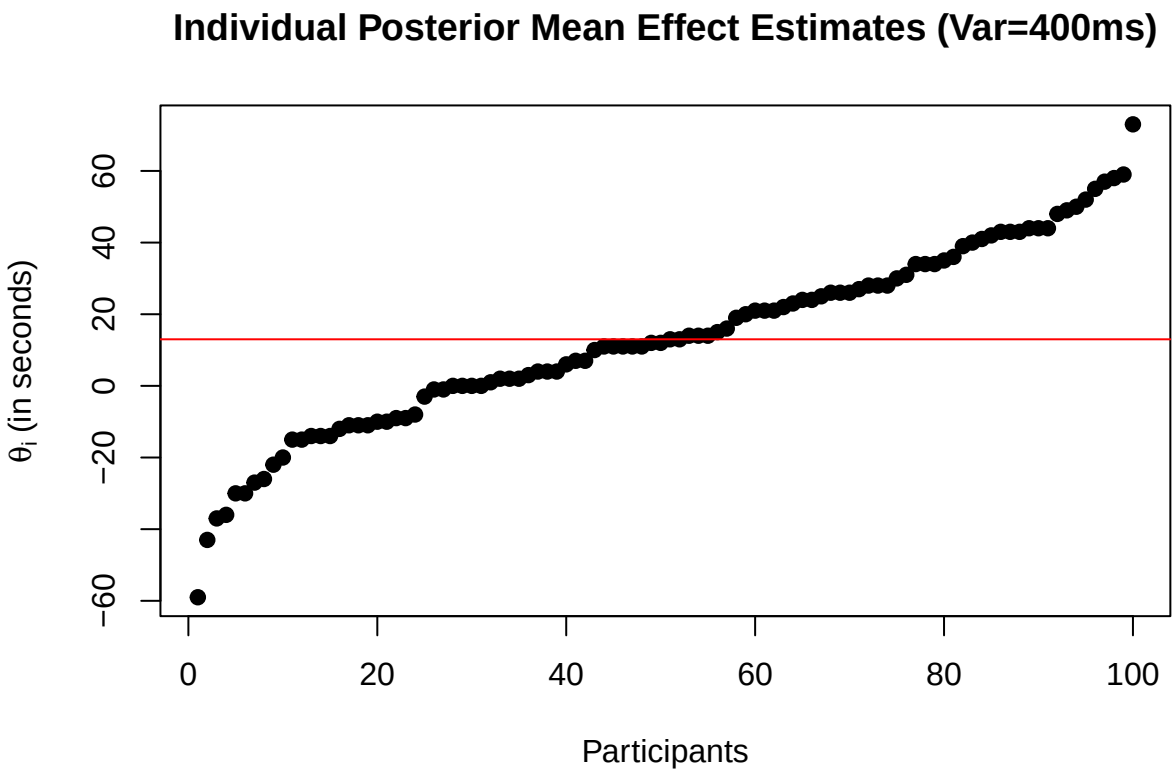
The estimated overall effect is 53 ms, and individual effects vary with a standard deviation of 21 ms around that mean. The unconstrained model is preferred with a Bayes factor of 8.06 indicating evidence that there are qualitative individual differences.

Variation adjustment = 100ms



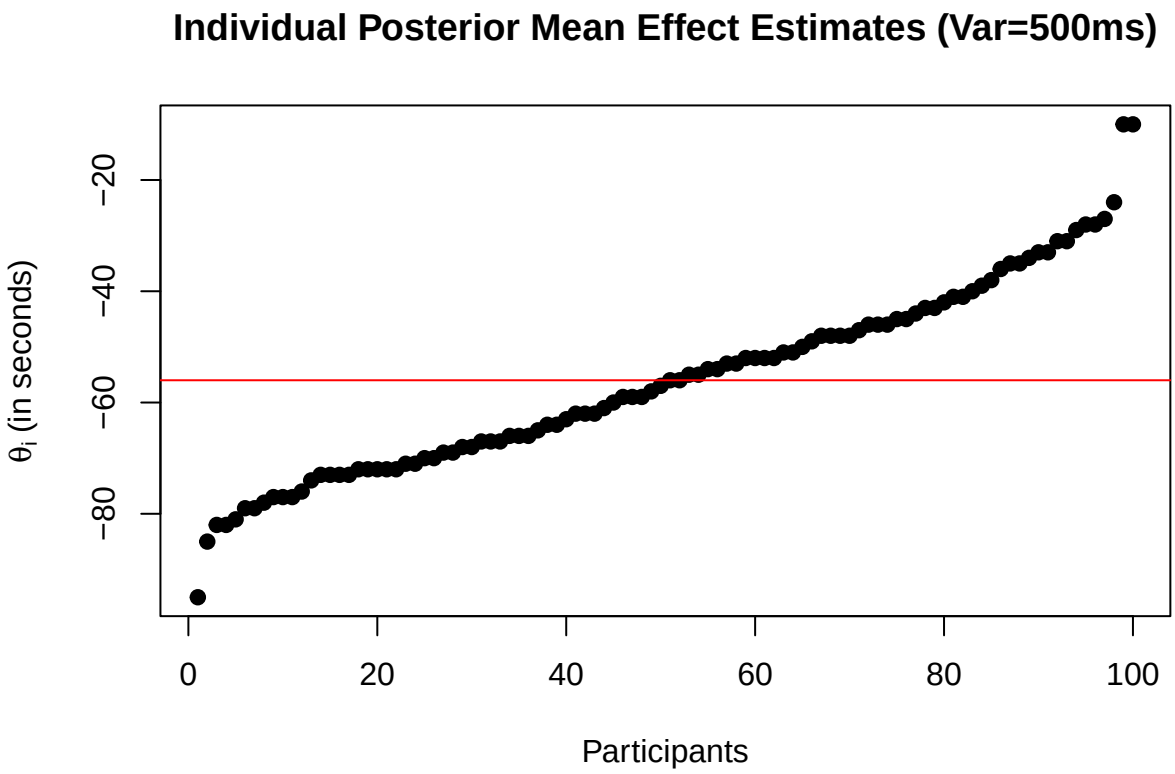
The estimated overall effect is 75 ms, and individual effects vary with a standard deviation of 26 ms around that mean. The common-effects model is preferred with a Bayes factor of 4.52 indicating evidence for a lack of individual differences.

Variation adjustment = 200ms



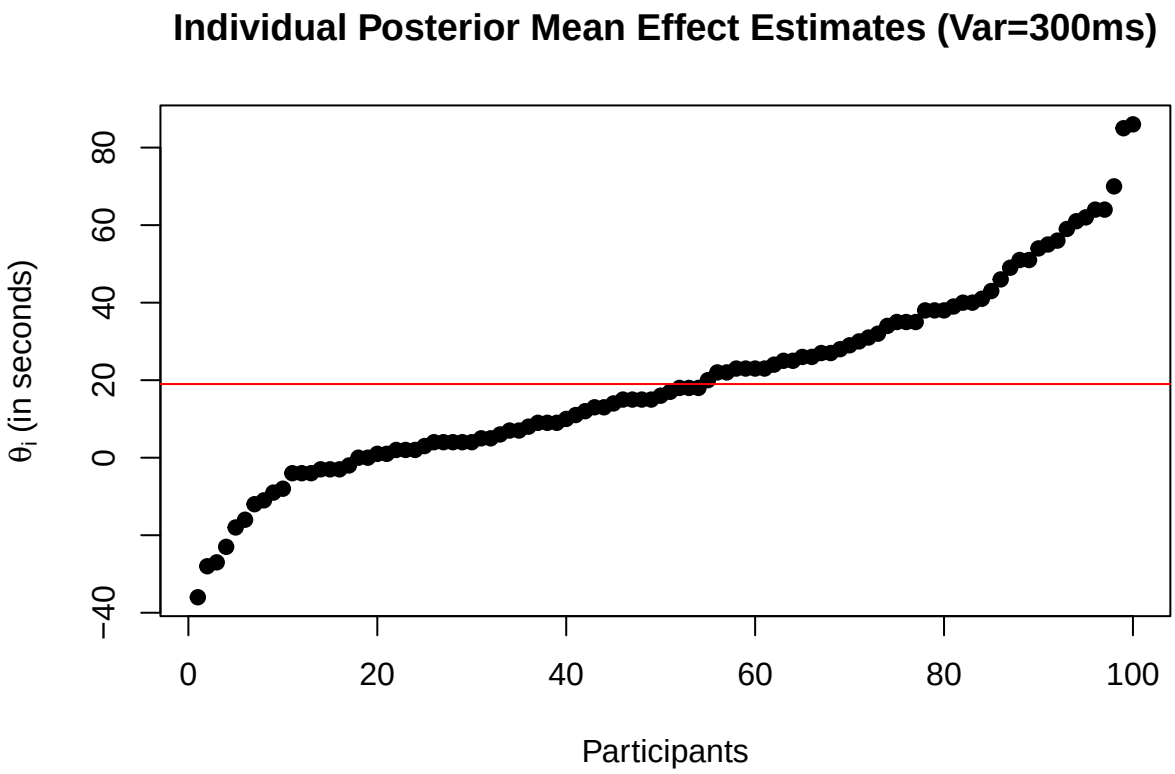
The estimated overall effect is 13 ms, and individual effects vary with a standard deviation of 25 ms around that mean. The unconstrained model is preferred with a Bayes factor of 7.04 indicating evidence that there are qualitative individual differences.

Variation adjustment = 300ms



The estimated overall effect is -56 ms, and individual effects vary with a standard deviation of 17 ms around that mean. The unconstrained model is preferred with a Bayes factor of 16.42 indicating evidence that there are qualitative individual differences.

Variation adjustment = 400ms



The estimated overall effect is 19 ms, and individual effects vary with a standard deviation of 24 ms around that mean. The common-effects model is preferred with a Bayes factor of 8.49 indicating evidence for a lack of individual differences.

Summary

It appears that there are no signficant differences between the two measurements when you vary the trial variation. There were differents between the two measures with varying sample sizes which were not expected. I would want to evaluate reasons for this finding if I were moving forward with such an analysis.

The qualitative individual differences did not show evidence for a positive-effect, which would be expected given the type of test (congruent, incongruent) and parameters. Interestingly as the variation of trial reaction time's increased, the overall mean effect (incongruentRT - congruentRT) decreased and in the final variation adjustment (500ms) the mean effect showed that overall individuals were quicker during the incongruent RT instead of congruent.

It's possibe that more factors are needed to properly compare IIV to meanRT. For instance, in this project I regressed RT on trials. The reason for this is to filter out practice effects with the expectation that reaction times are longer at the beginning and slower towards the end of the trial. The data was randomly generated and thus this pattern would not of occured. Additionally as part of IIV, it is typical to regress RT on trials, age, and the interaction of trials and age $RT \sim trials + age + trials * age$. In the future I would of started with the Bayes model comparison instead of ending with it to better inform my simulations. This is a comparision that I will continue to examing moving forward with lab data.