ManyBabies 5: The Hunter and Ames Model of Infant Looking Preference

Supplementary Materials: Data Simulation and Power Analysis

ManyBabies Analysis Team

Contents

1 Data Simulation **2**

2 Visualisation of Simulated Data **4**

2.1 Familiarization: 4

2.2 Complexity: 5

2.3 Age: 6

2.4 Age*Familiarization: 7

2.5 Age*Complexity: 8

2.6 Familiarization*Complexity: 9

2.7 Variability by Lab 11

2.8 Variability by Item 12

3 Power Calculation with Full Data and Varying Intercepts and Varying Slopes **13**

3.1 Effect Size = 0.5 13

3.2 Effect Size = 0.4 13

3.3 Effect Size = 0.3 13

3.3.1 Visualise Estimates for Fixed Effects: 14

3.3.2 Visualise Estimates for Random Effects: 15

3.4 Effect Size = 0.2 16

3.5 Effect Size = 0.1 16

4 Power Calculation with Full Data and Varying Intercepts **17**

4.1 Effect Size = 0.5 17

4.2 Effect Size = 0.4 17

4.3 Effect Size = 0.3 17

4.3.1 Visualise Estimates for Fixed Effects: 18

4.3.2 Visualise Estimates for Random Effects: 19

4.4 Effect Size = 0.2 20

4.5 Effect Size = 0.1 20

5 Power Calculation with 20 pct. Missing Data and Varying Intercepts and Varying Slopes **21**
5.1 Effect Size = 0.5 . . . 21
5.2 Effect Size = 0.4 . . . 21
5.3 Effect Size = 0.3 . . . 22
5.3.1 Visualise Estimates for Fixed Effects: . . . 22
5.3.2 Visualise Estimates for Random Effects: . . . 23
5.4 Effect Size = 0.2 . . . 24
5.5 Effect Size = 0.1 . . . 24

6 Power Calculation with 50 pct. Missing Data and Varying Intercepts and Varying Slopes **25**
6.1 Effect Size = 0.5 . . . 25
6.2 Effect Size = 0.4 . . . 25
6.3 Effect Size = 0.3 . . . 26
6.3.1 Visualise Estimates for Fixed Effects: . . . 26
6.3.2 Visualise Estimates for Random Effects: . . . 27
6.4 Effect Size = 0.2 . . . 28
6.5 Effect Size = 0.1 . . . 28

7 Overview of Power Simulation Results **29**
7.1 Summary Statistics for Power Calculation with Full Data and Varying Intercepts and Varying Slopes: . . . 29
7.2 Summary Statistics for Power Calculation with Full Data and Varying Intercepts: . . . 29
7.3 Summary Statistics for Power Calculation with 20 pct. Missing Data and Varying Intercepts and Varying Slopes: . . . 29
7.4 Summary Statistics for Power Calculation with 50 pct. Missing Data and Varying Intercepts and Varying Slopes: . . . 30

8 Overview of Bias Results **30**

9 Increasing Variance with Trial Number **31**
9.1 Extracting estimates of increasing variance from the adult pilot data . . . 31
9.2 Visualisation of adult data in proportions . . . 32
9.3 Visualisation of adult data in z-score . . . 33
9.4 Modification of simulation function to include variance increase . . . 34
9.5 Visualise the variance increase . . . 37
9.6 Model building and bias assessment . . . 38
9.7 Test how many of the models violate homoskedasticity . . . 42
9.8 Overview of how many of the models exhibit homoskedasticity . . . 42
9.9 Bayesian robust location-scale regression model . . . 42
9.9.1 Prior-Posterior Update Checks . . . 43

10 Data missing not completely at random **44**
10.1 Simulation of missing data according to increasing infant age . . . 44
10.1.1 Simulation of models . . . 45
10.1.2 Visualise Estimates for Fixed Effects: . . . 46
10.1.3 Visualise Estimates for Random Effects: . . . 47
10.2 Simulation of missing data according to increasing trial number . . . 48

10.2.1 Simulation of models . . . 49
10.2.2 Visualise Estimates for Fixed Effects: . . . 50
10.2.3 Visualise Estimates for Random Effects: . . . 51
10.3 Simulation of missing data according to increasing familiarisation . . . 52
10.3.1 Simulation of models . . . 53
10.3.2 Visualise Estimates for Fixed Effects: . . . 54

11 Summary Statistics for Power Calculation with Data missing not completely at random 56

12 Logistic Regression Models 57

1 Data Simulation

```
my_sim_data <- function(
    n_subj      = 1280,   # number of subjects
    n_simple  =  6,   # number of simple stimuli
    n_complex =  6,   # number of complex stimuli
    n_small_fam = 4,   #small familiarization time
    n_medium_fam = 4,  #medium familiarization time
    n_high_fam = 4,    #high familiarization time
    n_lab = 40,

    beta_0  =  0, # intercept; i.e., the grand mean
    beta_c  =  0.3, # main effect for complexity
    beta_f = 0.3, # main effect for familiarization time
    beta_a = 0.3, # main effect for age

    beta_ca = 0.3,
    beta_af = 0.3,
    beta_cf = 0.3,

    beta_cfa = 0.3, #main effect for interaction between complexity and familiarization.

    subject_0   = 0.2, # by-subject random intercept sd

    subject_c   =  0.2, # by-subject slope complexity sd
    subject_f = 0.2, # by-subject slope familiarization sd
    subject_a = 0.2, # by-subject slope age sd

    subject_ca = 0.2,# by-subject slope for interaction between age and complexity sd
    subject_af = 0.2, # by-subject slope for interaction between age and familiarization sd
    subject_cf = 0.2, # by-subject slope complexity*familiarization sd

    subject_cfa = 0.2, # by-subject slope for interaction between age, complexity and familiarization sd

    subj_rho     =  .2, # correlations between by-subject random effects

    lab_0 = 0.2, # by-lab random intercept sd

    lab_c = 0.2, # by-lab slope complexity sd
    lab_f = 0.2, # by-lab slope familiarization sd
    lab_a = 0.2, # by-lab slope age sd

    lab_ca = 0.2, # by-lab slope for interaction between age and complexity sd
    lab_af = 0.2, # by-lab slope for interaction between age and familiarization sd
    lab_cf = 0.2, # by-lab random slope complexity*familiarization sd

    lab_cfa = 0.2, # by-lab slope for interaction between age, complexity and familiarization sd

    lab_rho = 0.2, # correlations between by-lab random effects

    item_0 =  0.2, # by-item random intercept sd

    item_c = 0.2, # by-item slope complexity sd
    item_f = 0.2, # by-item slope familiarization sd
    item_a = 0.2, # by-item slope age sd

    item_ca = 0.2, # by-item slope for interaction between age and complexity sd
    item_af = 0.2, # by-item slope for interaction between age and familiarization sd
    item_cf = 0.2, # by-item random slope complexity*familiarization sd

    item_cfa = 0.2, # by-item slope for interaction between age, complexity and familiarization sd
```

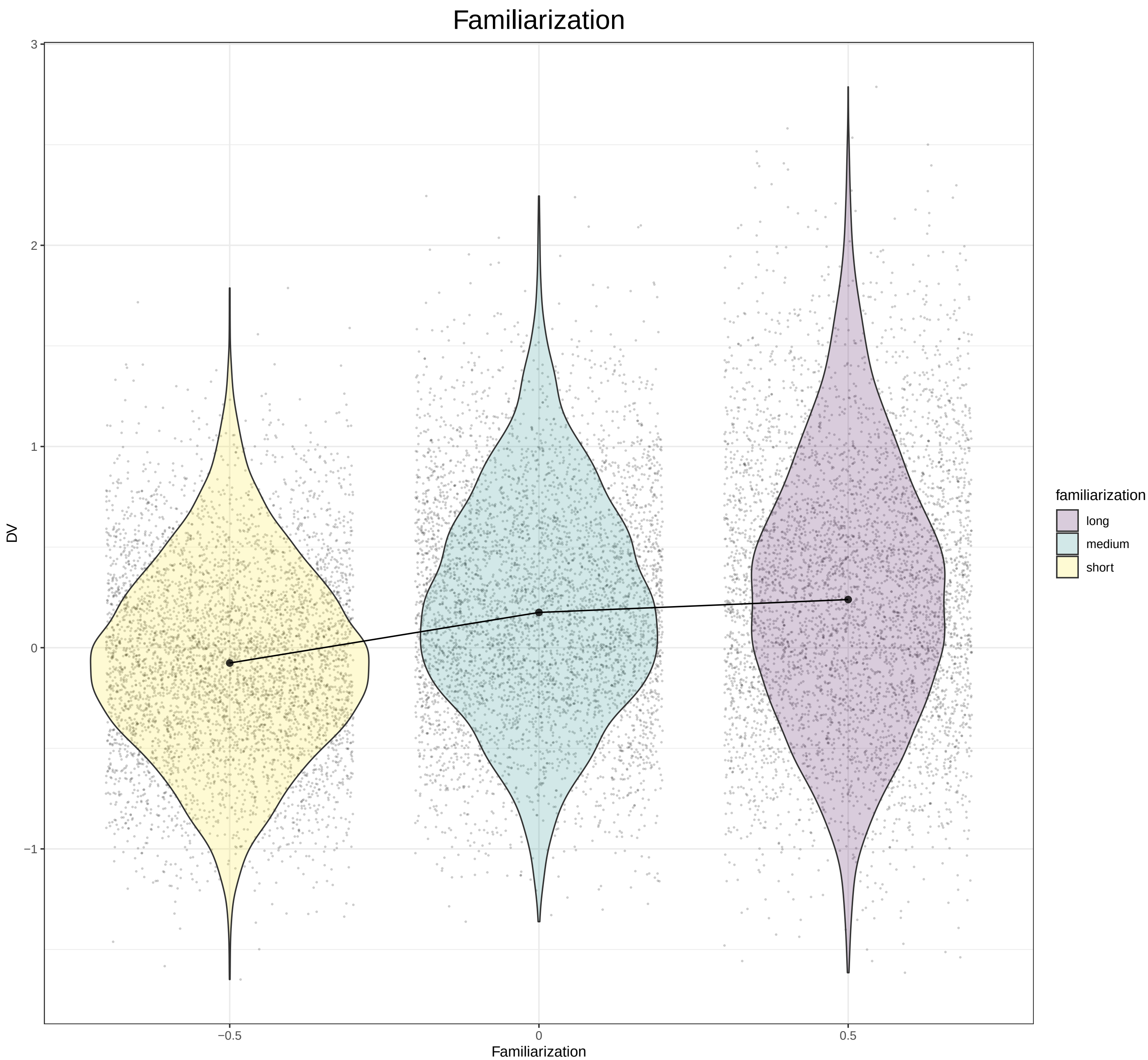
```r
  item_rho = 0.2, # correlations between by-item random effects


  sigma = 0.3 # residual (error) sd
  ) { # residual (standard deviation)

# simulate a sample of items
items <- data.frame(
  category = rep(c("simple", "complex"), c(n_simple, n_complex)),
  X_c = rep(c(-0.5, 0.5), c(n_simple, n_complex)),
  familiarization = rep(c("short", "medium", "long"), (n_simple + n_complex)/3),
  X_f = rep(c(-0.5, 0, 0.5), (n_simple + n_complex)/3),
  faux::rnorm_multi(
  n = n_simple + n_complex, mu = 0, sd = c(item_0,
                              item_c,
                              item_f,
                              item_a,
                              item_ca,
                              item_af,
                              item_cf,
                              item_cfa), r = item_rho,
  varnames = c("I_0", "I_c","I_f","I_a",
             "I_ca","I_af", "I_cf",
             "I_cfa"))
  ) %>%
  mutate(item_id = faux::make_id(nrow(.), "I"))

# simulate a sample of subjects
subjects <-
  faux::rnorm_multi(
  n = n_subj, mu = 0, sd = c(subject_0,
                              subject_c,
                              subject_f,
                              subject_a,
                              subject_ca,
                              subject_af,
                              subject_cf,
                              subject_cfa), r = subj_rho,
  varnames = c("S_0", "S_c","S_f","S_a",
             "S_ca","S_af", "S_cf",
             "S_cfa")
) %>%
  mutate(subj_id = faux::make_id(nrow(.), "S")) %>%
  mutate(X_a = runif(n_subj, min = -0.5, max = 0.5))
#add subject age measure, sample from distribution from -0.5 to 0.5. #subjects$subj_id <- 1:n_subj

labs <- faux::rnorm_multi(
  n = n_lab, mu = 0, sd = c(lab_0, lab_c, lab_f, lab_a,
                             lab_ca, lab_af, lab_cf,
                             lab_cfa), r = lab_rho,
  varnames = c("L_0", "L_c","L_f","L_a",
             "L_ca","L_af", "L_cf",
             "L_cfa")
) %>%
  mutate(lab_id = faux::make_id(nrow(.), "L"))

#create lab and subj nesting structure
#Number of subjects must be a multiple of number of labs
lab_multiplier = n_subj/n_lab
lab_subj_dict <- data.frame(
  subj_id = subjects$subj_id,
```

```
    lab_id = rep(labs$lab_id,lab_multiplier)
  )

  # cross subject and item IDs
  temp <- crossing(subjects, items)  %>%
    left_join(lab_subj_dict, by = "subj_id") %>%
    left_join(labs, by = "lab_id") %>%
    group_by(subj_id, item_id) %>%
    mutate(item_id = sample(item_id)) %>%
    ungroup() %>%
    mutate(trial_num = rep(seq(n_simple + n_complex), n_subj))

  temp <- temp %>%
    mutate(
      B_0  = beta_0 + S_0 + L_0 + I_0,

      B_c  = beta_c + S_c + L_c + I_c,
      B_f  = beta_f + S_f + L_f + I_f,
      B_a = beta_a + S_a + L_a + I_a,

      B_ca = beta_ca + S_ca + L_ca + I_ca,
      B_af = beta_af + S_af + L_af + I_af,
      B_cf = beta_cf + S_cf + L_cf + I_cf,

      B_cfa = beta_cfa + S_cfa + L_cfa + I_cfa,

      e_si = rnorm(nrow(temp), mean = 0, sd = sigma),

      DV = B_0 +
        (B_a * X_a) + (B_c * X_c) + (B_f * X_f) +
        (B_cf * X_c * X_f) + (B_af * X_a * X_f) + (B_ca * X_c * X_a) +
        (B_cfa * X_c * X_f * X_a) + e_si
    )
}

dat_sim <- my_sim_data()
```

2 Visualisation of Simulated Data

2.1 Familiarization:

```
dat_sim_plot_familiarization <- dat_sim %>%
    group_by(X_f) %>%
    dplyr::summarise(med_DV = median(DV))

plot_familiarization <- dat_sim %>%
    mutate(X_f = as.factor(X_f)) %>%
    ggplot() + geom_point(aes(y = DV, x = X_f), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_violin(aes(y = DV, x = X_f,
    fill = familiarization), alpha = 0.2) + geom_line(aes(y = med_DV,
    x = as.factor(X_f), group = 1), data = dat_sim_plot_familiarization) +
    geom_point(aes(y = med_DV, x = as.factor(X_f)), alpha = 0.8,
        size = 2, data = dat_sim_plot_familiarization) + scale_fill_manual(values = viridis(n = 3)) +
    ggtitle("Familiarization") + xlab("Familiarization") + theme_bw()

plot_familiarization <- plot_familiarization + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_familiarization
```



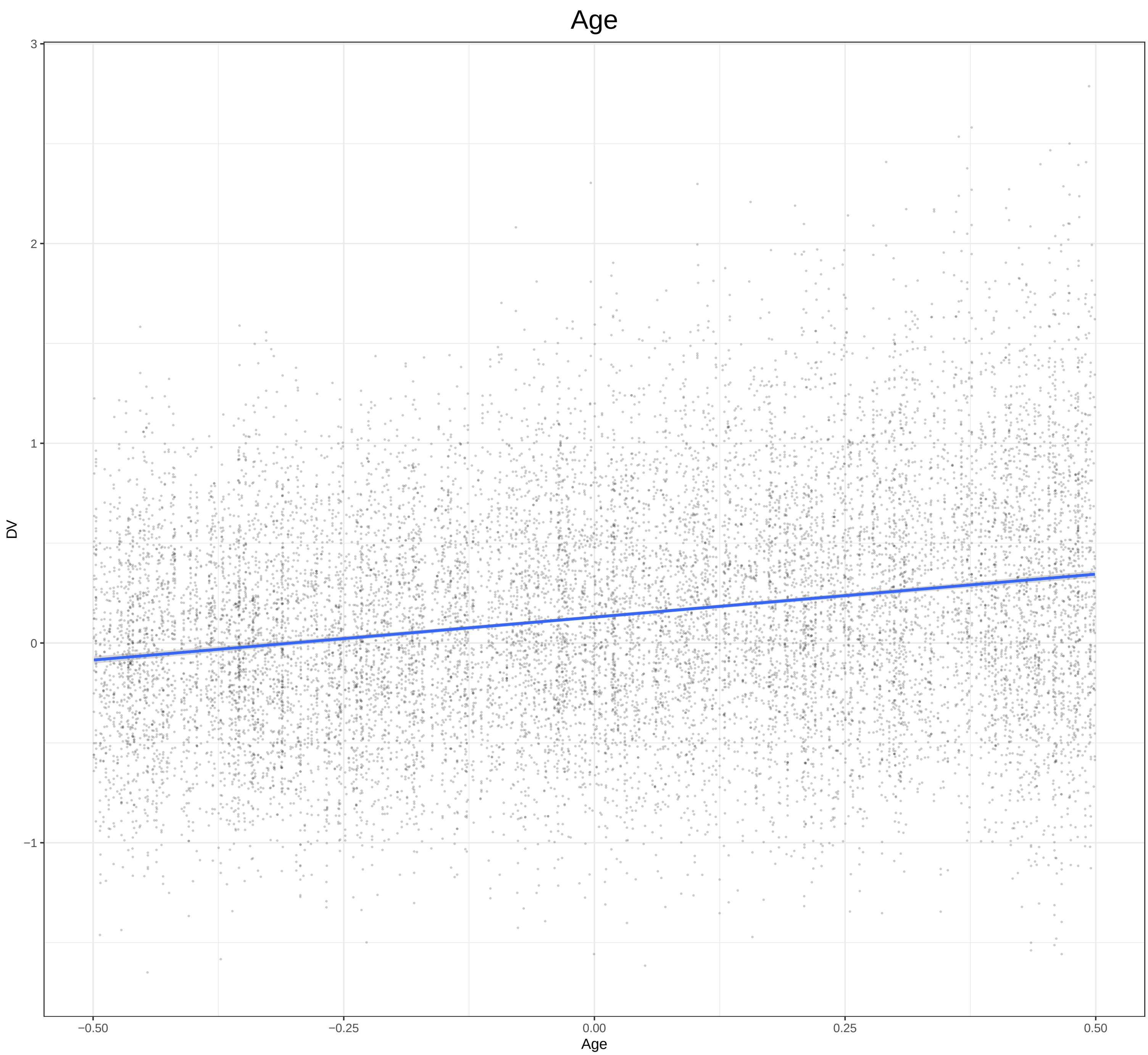
2.2 Complexity:

```
dat_sim_plot_complexity <- dat_sim %>%
    group_by(X_c) %>%
    dplyr::summarise(med_DV = median(DV))

plot_complexity <- dat_sim %>%
    mutate(X_c = as.factor(X_c)) %>%
    ggplot() + geom_point(aes(y = DV, x = X_c), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_violin(aes(y = DV, x = X_c,
    fill = category), alpha = 0.2) + geom_line(aes(y = med_DV,
    x = as.factor(X_c), group = 1), data = dat_sim_plot_complexity) +
    geom_point(aes(y = med_DV, x = as.factor(X_c)), alpha = 0.8,
        size = 2, data = dat_sim_plot_complexity) + scale_fill_manual(values = viridis(n = 2)) +
    ggtitle("Stimulus Complexity") + xlab("Stimulus Complexity") +
    theme_bw()
```

```
plot_complexity <- plot_complexity + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_complexity
```

2.3 Age:

```
plot_age <- dat_sim %>%
    ggplot() + geom_point(aes(y = DV, x = X_a), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_smooth(method = "lm", se = TRUE,
    formula = y ~ x, aes(y = DV, x = X_a)) + ggtitle("Age") +
    xlab("Age") + theme_bw()

plot_age <- plot_age + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_age
```

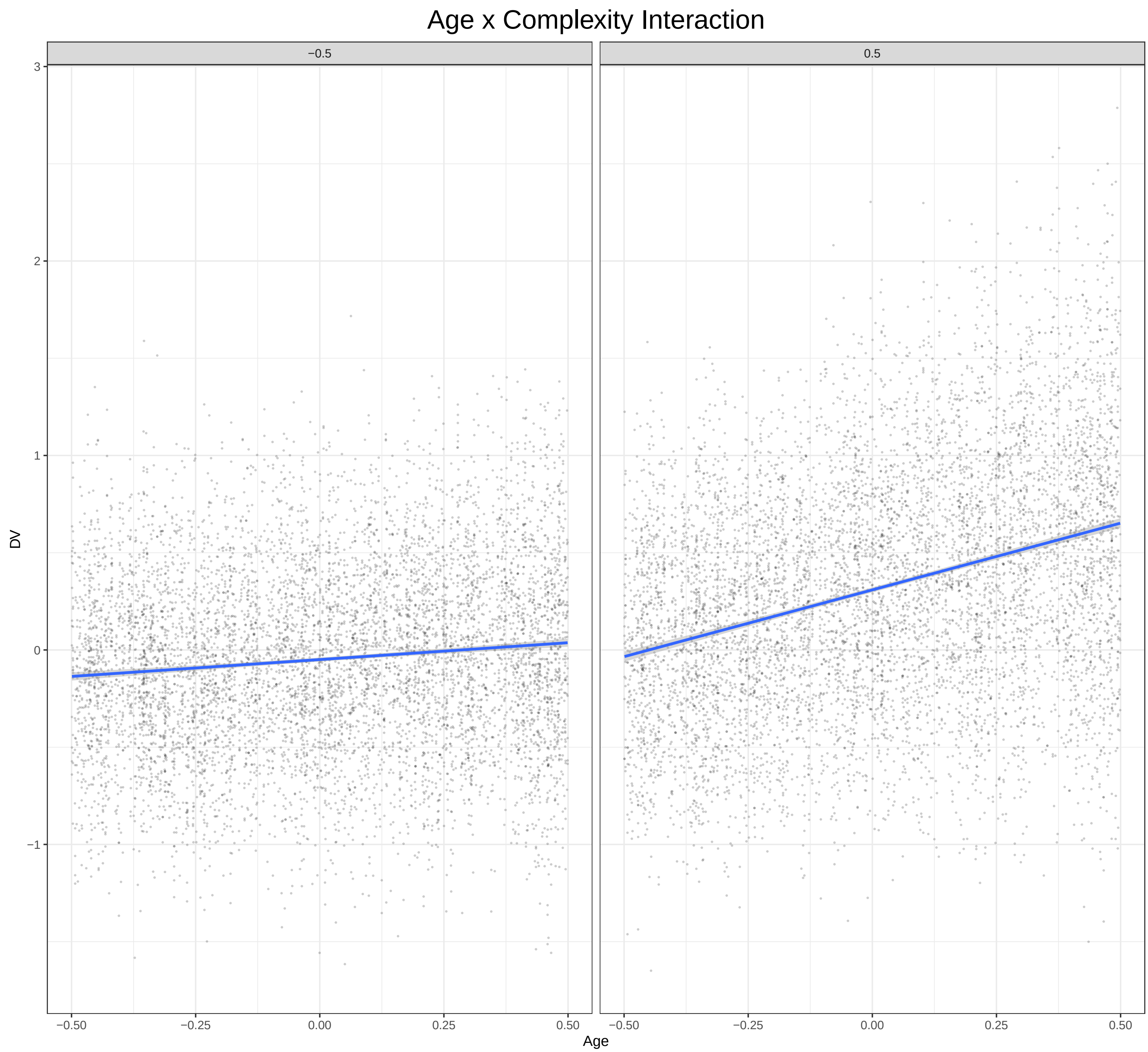


2.4 Age*Familiarization:

```
plot_age_familiarization <- dat_sim %>%
    ggplot() + geom_point(aes(y = DV, x = X_a), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_smooth(method = "lm", formula = y ~
    x, se = TRUE, aes(y = DV, x = X_a)) + facet_wrap(~X_f) +
    ggtitle("Age x Familiarization Interaction") + xlab("Age") +
    theme_bw()
plot_age_familiarization <- plot_age_familiarization + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_age_familiarization
```

2.5 Age*Complexity:

```
plot_age_complexity <- dat_sim %>%
    ggplot() + geom_point(aes(y = DV, x = X_a), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_smooth(method = "lm", formula = y ~
    x, se = TRUE, aes(y = DV, x = X_a)) + facet_wrap(~X_c) +
    ggtitle("Age x Complexity Interaction") + xlab("Age") + theme_bw()

plot_age_complexity <- plot_age_complexity + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_age_complexity
```



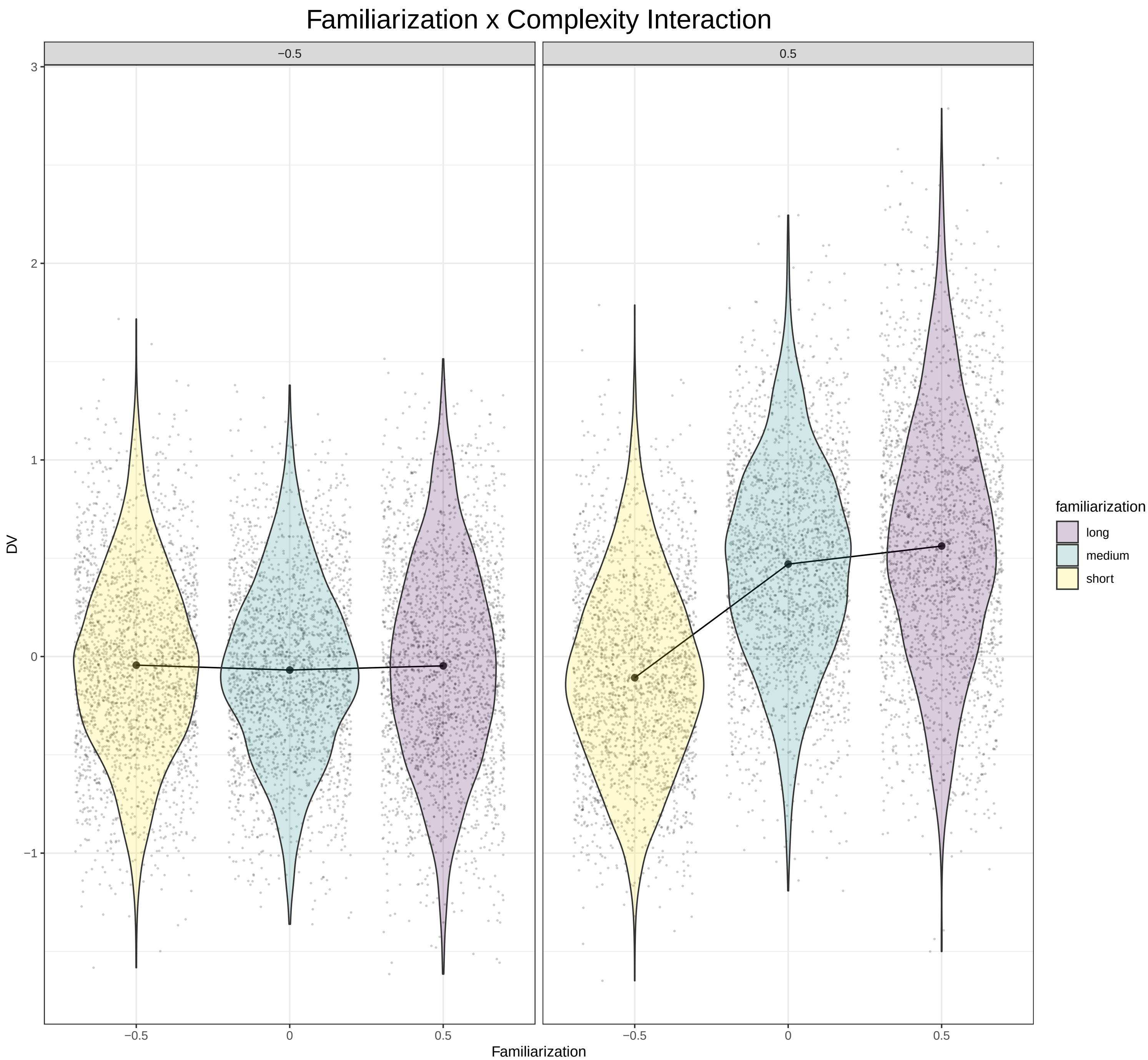
2.6 Familiarization*Complexity:

```
dat_f_c_interaction <- dat_sim %>%
    mutate(X_c = as.factor(X_c)) %>%
    mutate(X_f = as.factor(X_f)) %>%
    group_by(X_f, X_c) %>%
    dplyr::summarise(med_DV = median(DV))
```

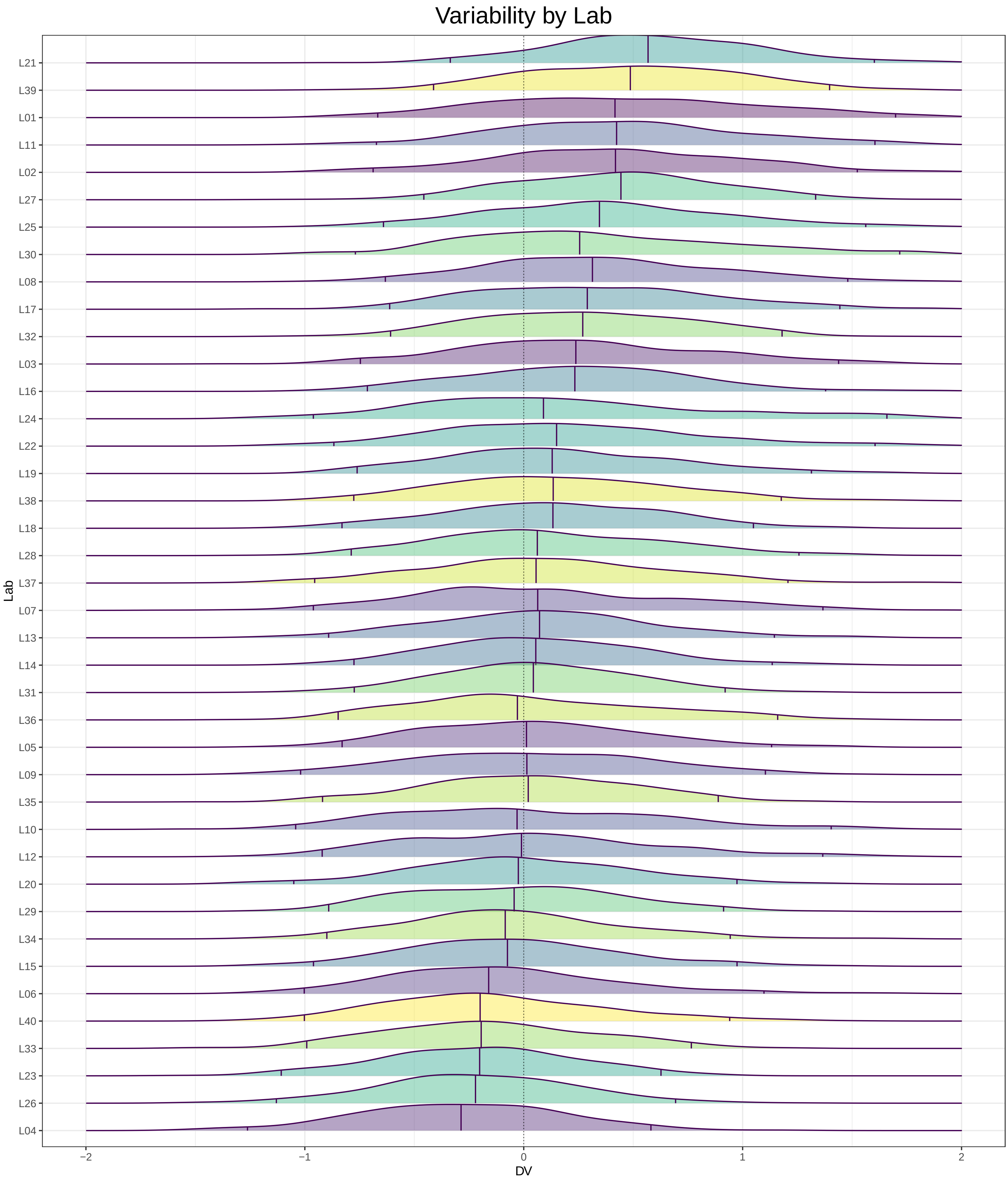
```
## `summarise()` has grouped output by 'X_f'. You can override using the `.groups`
## argument.
```

```
plot_familiarization_complexity <- dat_sim %>%
    mutate(X_c = as.factor(X_c)) %>%
    mutate(X_f = as.factor(X_f)) %>%
    ggplot() + geom_point(aes(y = DV, x = X_f), position = "jitter",
```

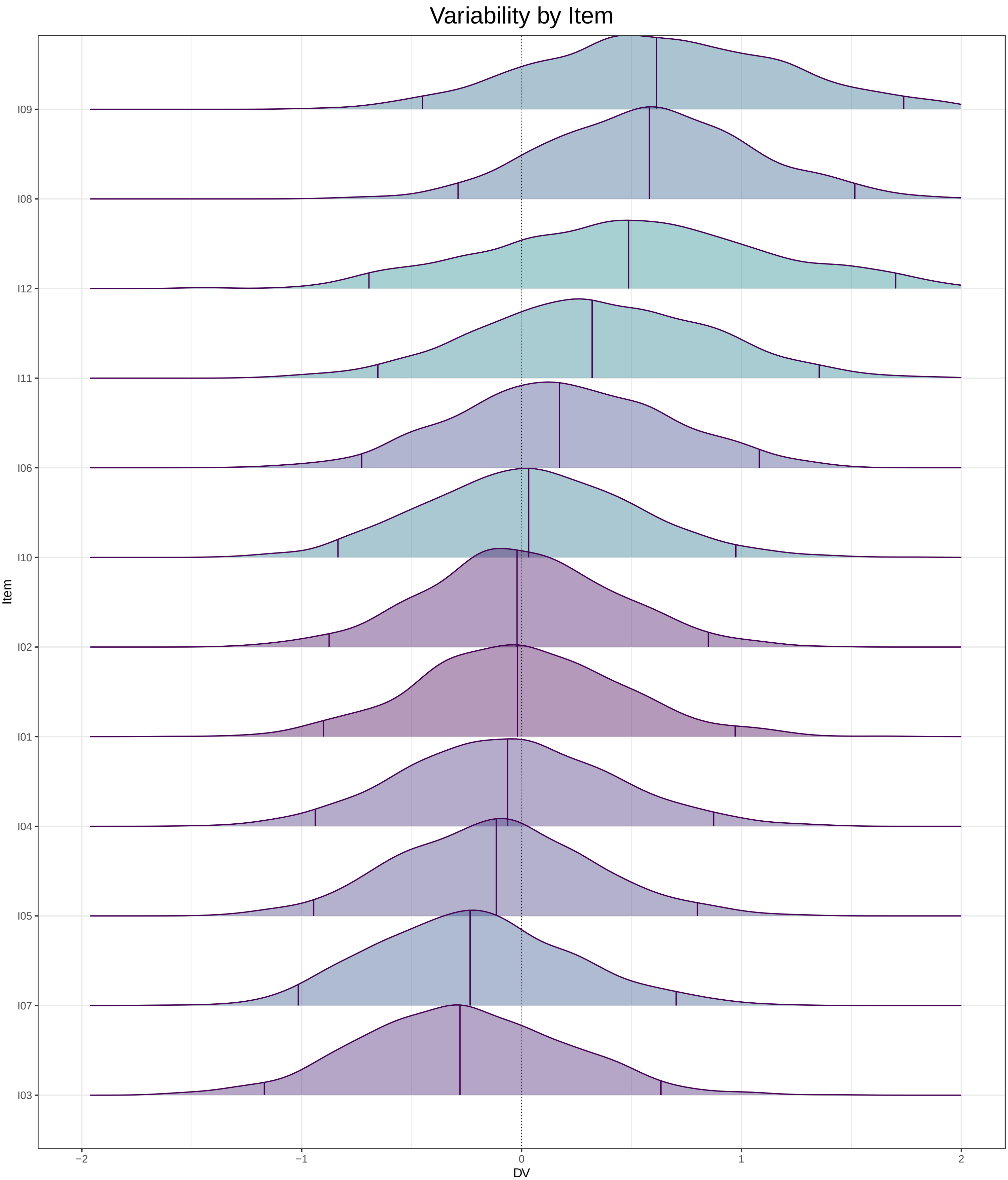
```
    alpha = 0.2, size = 0.2) + geom_point(aes(y = med_DV, x = as.factor(X_f)),
    alpha = 0.8, size = 2, data = dat_f_c_interaction) + geom_line(aes(y = med_DV,
    x = as.factor(X_f), group = 1), data = dat_f_c_interaction) +
    geom_violin(aes(y = DV, x = X_f, fill = familiarization),
        alpha = 0.2) + scale_fill_manual(values = viridis(n = 3)) +
    facet_wrap(~X_c) + ggtitle("Familiarization x Complexity Interaction") +
    xlab("Familiarization") + theme_bw()
plot_familiarization_complexity <- plot_familiarization_complexity +
    theme(plot.title = element_text(hjust = 0.5, size = 20))
plot_familiarization_complexity
```



2.7 Variability by Lab



2.8 Variability by Item



3 Power Calculation with Full Data and Varying Intercepts and Varying Slopes

3.1 Effect Size = 0.5

```
# Number of simulations:
reps <- 500

# Simulation function:
run_sims <- function(filename_full, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,

                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,

                         beta_cfa = ef)

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 + X_c * X_f | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=dat_sim)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_full)
  write_csv(sim_results, filename_full, append = append)

  # return the tidy table
  sim_results
}

filename_full_0.5 = 'run_sims_full_0.5.csv'
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.5, ef = 0.5))
end_time <- Sys.time()
end_time - start_time
```

3.2 Effect Size = 0.4

```
filename_full_0.4 = "run_sims_full_0.4.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

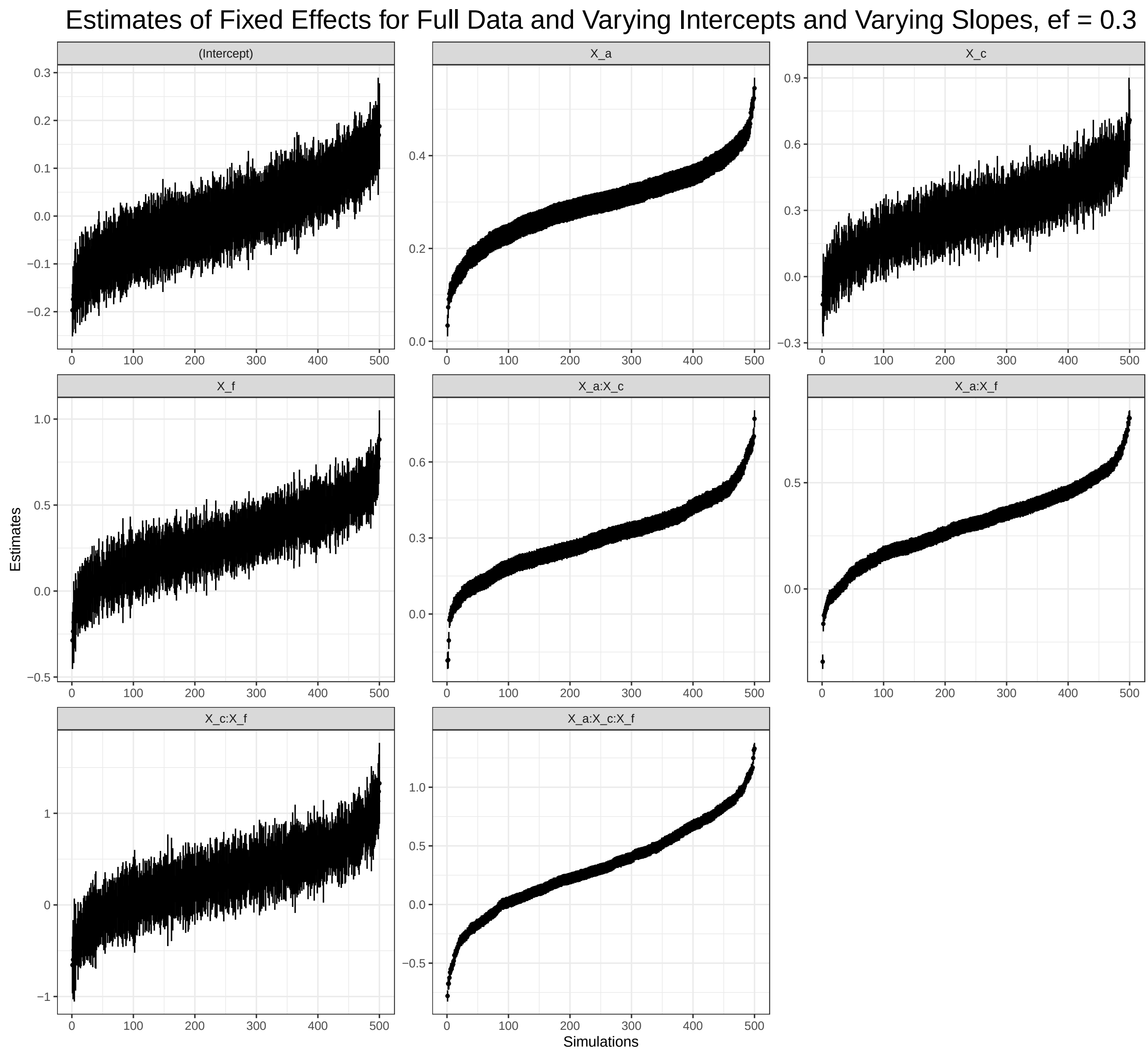
3.3 Effect Size = 0.3

```
filename_full_0.3 = "run_sims_full_0.3.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.3,
```

```
    ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

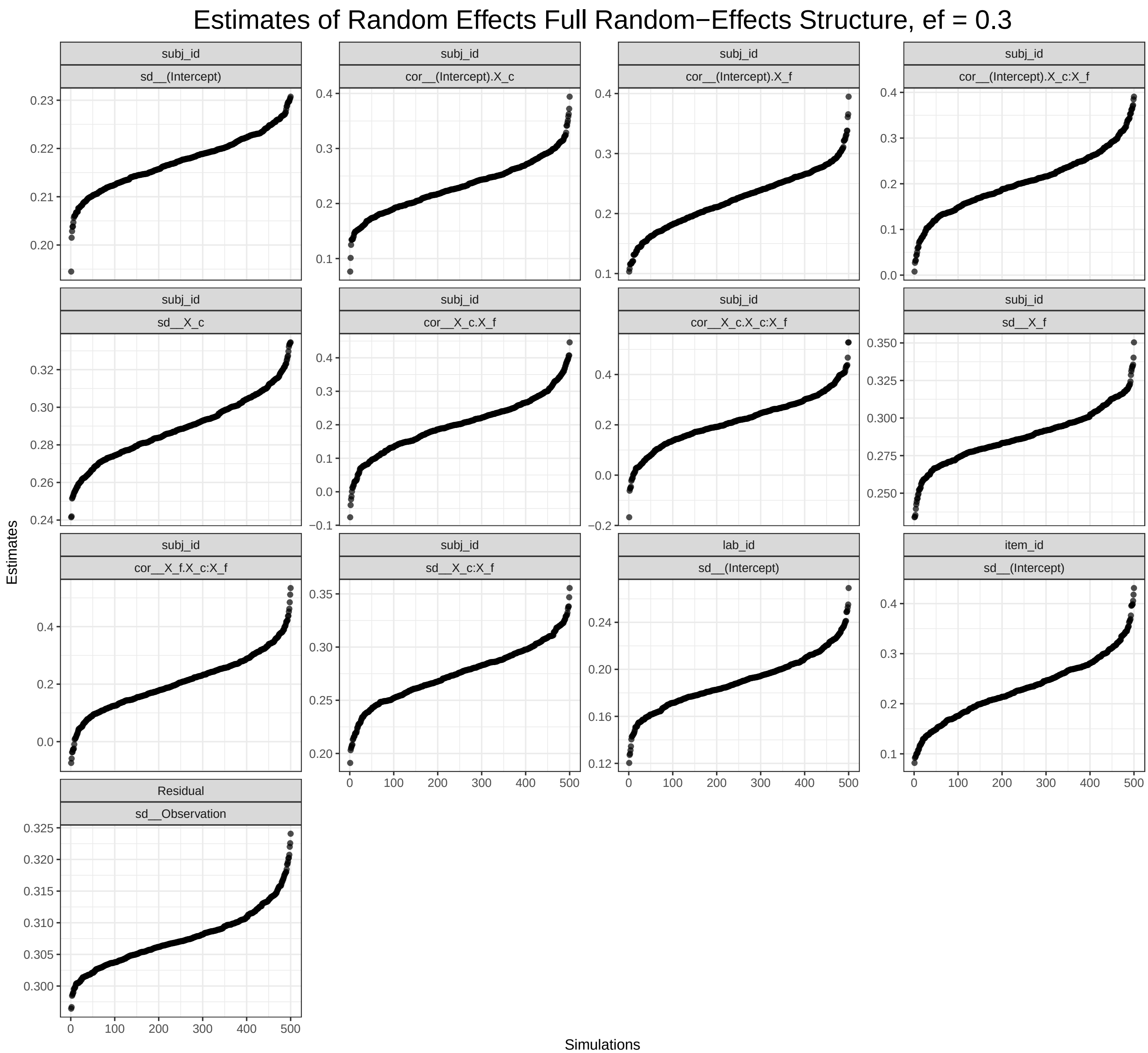
3.3.1 Visualise Estimates for Fixed Effects:

```
sims_full_0.3 <- read_csv(filename_full_0.3, col_types = cols(group = col_factor(ordered = TRUE),
    term = col_factor(ordered = TRUE)))

fixed_full_plot <- sims_full_0.3 %>%
    filter(effect == "fixed") %>%
    ungroup() %>%
    arrange(term, estimate) %>%
    mutate(row = rep(seq(1:reps), 8)) %>%
    ggplot(aes(x = row, y = estimate, ymin = estimate - std.error,
        ymax = estimate + std.error)) + facet_wrap(~term, scales = "free") +
    geom_pointrange(fatten = 1/2) + ylab("Estimates") + xlab("Simulations") +
    ggtitle("Estimates of Fixed Effects for Full Data and Varying Intercepts and Varying Slopes, ef = 0.3") +
    theme_bw()

fixed_full_plot <- fixed_full_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
fixed_full_plot
```



3.3.2 Visualise Estimates for Random Effects:

```
ran_full_plot <- sims_full_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 13)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects Full Random-Effects Structure,
    theme_bw()
ran_full_plot <- ran_full_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_full_plot
```



3.4 Effect Size = 0.2

```
filename_full_0.2 = "run_sims_full_0.2.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.2,
    ef = 0.2))
end_time <- Sys.time()
end_time - start_time
```

3.5 Effect Size = 0.1

```
filename_full_0.1 = "run_sims_full_0.1.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.1,
```

```
    ef = 0.1))
end_time <- Sys.time()
end_time - start_time
```

4 Power Calculation with Full Data and Varying Intercepts

4.1 Effect Size = 0.5

```
# Simulation function:
run_sims <- function(filename_full, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,

                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,

                         beta_cfa = ef)

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=dat_sim)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_full)
  write_csv(sim_results, filename_full, append = append)

  # return the tidy table
  sim_results
}

filename_full_int_0.5 = 'run_sims_full_int_0.5.csv'
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.5, ef = 0.5))
end_time <- Sys.time()
end_time - start_time
```

4.2 Effect Size = 0.4

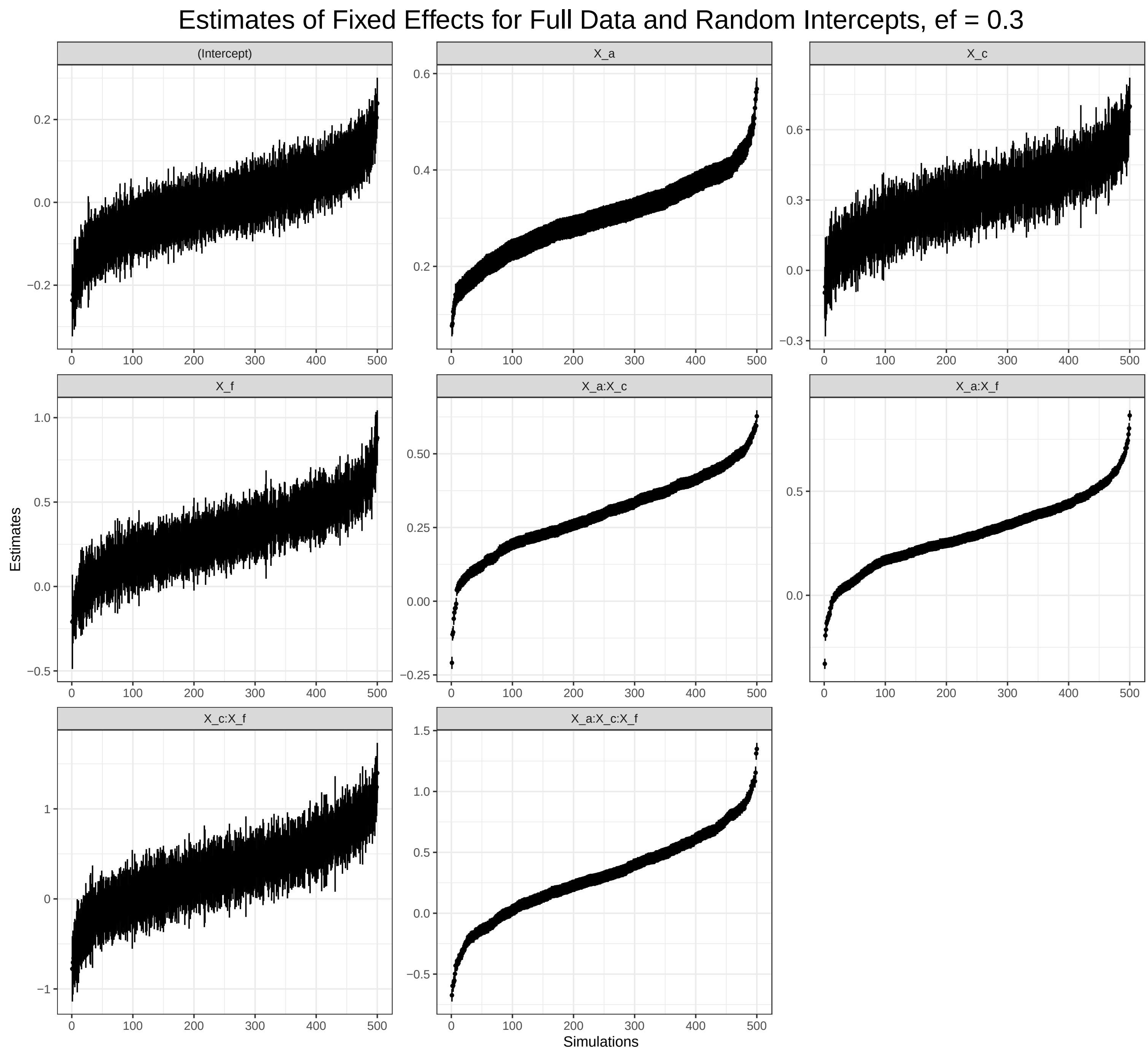
```
filename_full_int_0.4 = "run_sims_full_int_0.4.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

4.3 Effect Size = 0.3

```
filename_full_int_0.3 = "run_sims_full_int_0.3.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.3,
    ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

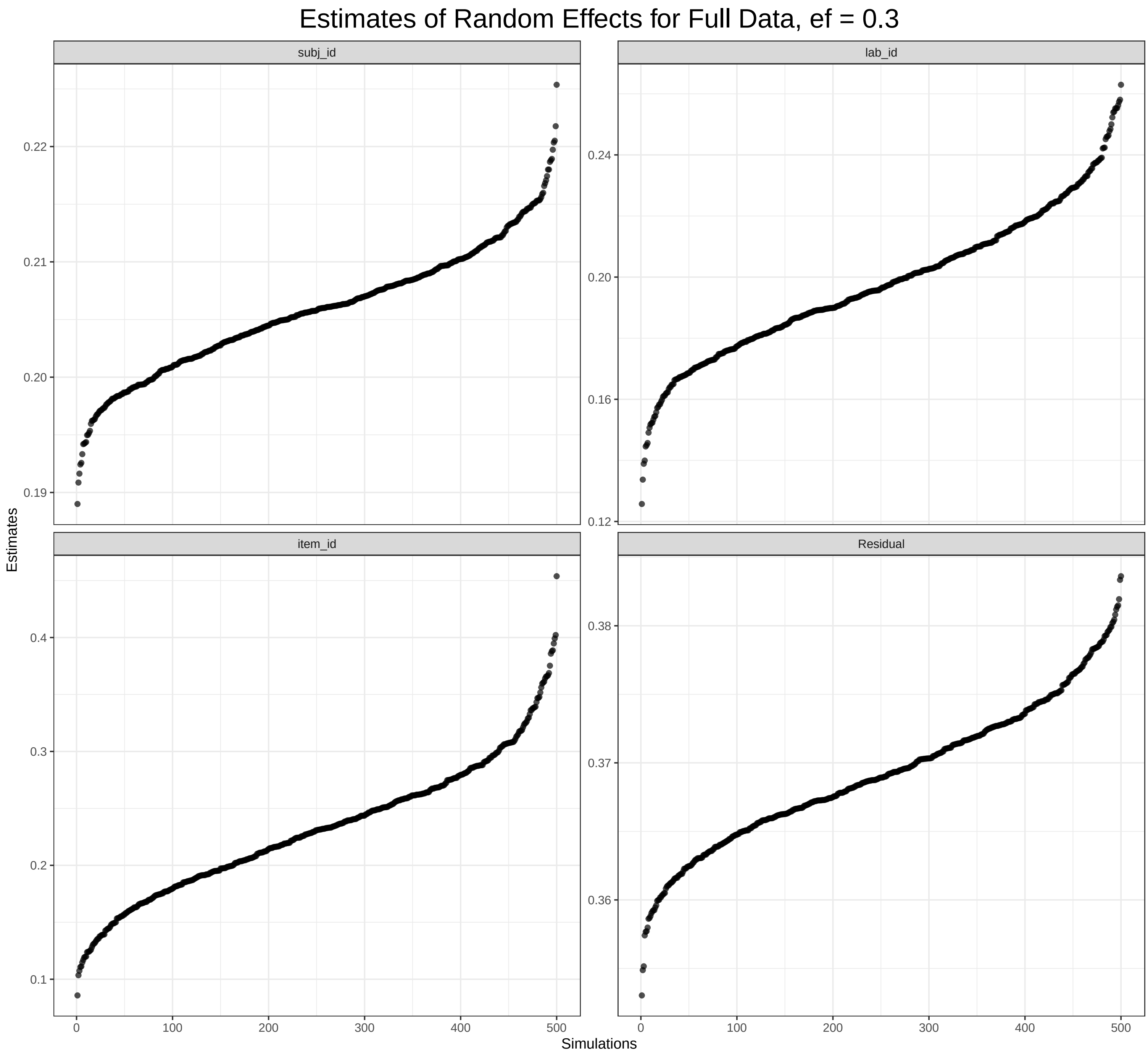
4.3.1 Visualise Estimates for Fixed Effects:

```
sims_full_int_0.3 <- read_csv(filename_full_int_0.3, col_types = cols(group = col_factor(ordered = TRUE),
    term = col_factor(ordered = TRUE)))

fixed_full_int_plot <- sims_full_int_0.3 %>%
    filter(effect == "fixed") %>%
    ungroup() %>%
    arrange(term, estimate) %>%
    mutate(row = rep(seq(1:reps), 8)) %>%
    ggplot(aes(x = row, y = estimate, ymin = estimate - std.error,
        ymax = estimate + std.error)) + facet_wrap(~term, scales = "free") +
    geom_pointrange(fatten = 1/2) + ylab("Estimates") + xlab("Simulations") +
    ggtitle("Estimates of Fixed Effects for Full Data and Random Intercepts, ef = 0.3") +
    theme_bw()

fixed_full_int_plot <- fixed_full_int_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
fixed_full_int_plot
```



4.3.2 Visualise Estimates for Random Effects:

```
ran_full_int_plot <- sims_full_int_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, estimate) %>%
    mutate(row = rep(seq(1:reps), 4)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group, scales = "free_y") + theme_bw() + ylab("Estimates") +
    xlab("Simulations") + ggtitle("Estimates of Random Effects for Full Data, ef = 0.3") +
    theme_bw()
ran_full_int_plot <- ran_full_int_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_full_int_plot
```



4.4 Effect Size = 0.2

```
filename_full_int_0.2 = "run_sims_full_int_0.2.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.2,
    ef = 0.2))
end_time <- Sys.time()
end_time - start_time
```

4.5 Effect Size = 0.1

```
filename_full_int_0.1 = "run_sims_full_int_0.1.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.1,
```

```
    ef = 0.1))
end_time <- Sys.time()
end_time - start_time
```

5 Power Calculation with 20 pct. Missing Data and Varying Intercepts and Varying Slopes

5.1 Effect Size = 0.5

```
run_sims_missing <- function(filename_missing, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                       beta_f = ef,
                       beta_a = ef,

                       beta_ca = ef,
                       beta_af = ef,
                       beta_cf = ef,

                       beta_cfa = ef)

  missing_samples <- dat_sim %>%
    mutate(nas = rbinom(nrow(dat_sim), 1, 1 - .20)) %>%
    mutate(DV = ifelse(nas == 1, DV, NA)) %>%
    drop_na()

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                  (1 + X_c * X_f | subj_id) +
                  (1 | lab_id) +
                  (1 | item_id),
                data=missing_samples)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_missing)
  write_csv(sim_results, filename_missing, append = append)

  # return the tidy table
  sim_results
}

filename_20_missing_0.5 = 'run_sims_20_missing_0.5.csv'
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.5, ef = 0.5))
end_time <- Sys.time()
end_time - start_time
```

5.2 Effect Size = 0.4

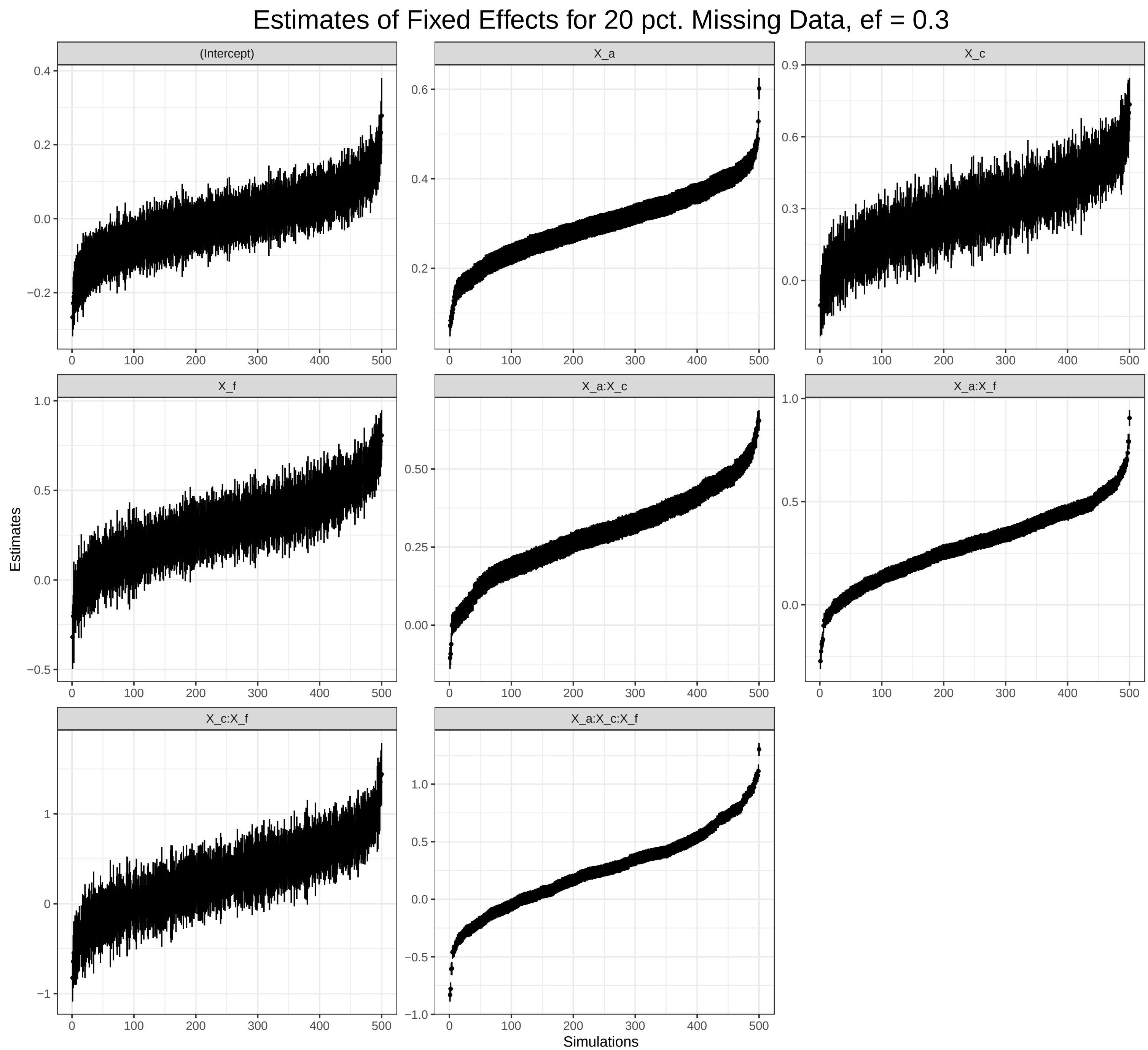
```
filename_20_missing_0.4 = "run_sims_20_missing_0.4.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

5.3 Effect Size = 0.3

```
filename_20_missing_0.3 = "run_sims_20_missing_0.3.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.3,
    ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

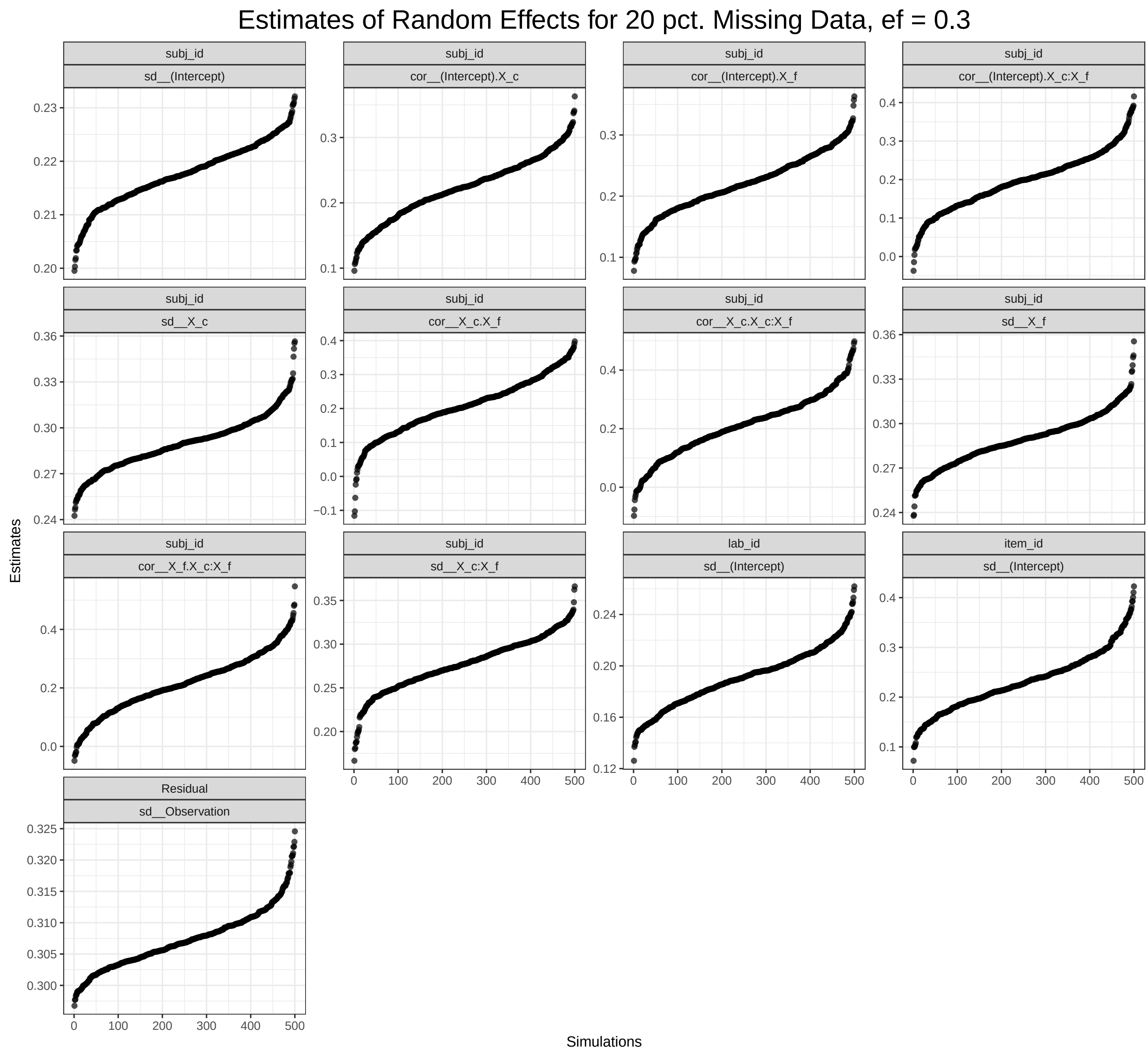
5.3.1 Visualise Estimates for Fixed Effects:

```
# read saved simulation data
sims_20_missing_0.3 <- read_csv(filename_20_missing_0.3, col_types = cols(
  # makes sure plots display in this order
  group = col_factor(ordered = TRUE),
  term = col_factor(ordered = TRUE)
))

fixed_missing_plot <- sims_20_missing_0.3 %>%
  filter(effect == "fixed") %>%
  ungroup() %>%
  arrange(term, estimate) %>%
  mutate(row = rep(seq(1:reps), 8)) %>%
  ggplot(aes(x = row, y = estimate, ymin = estimate-std.error, ymax = estimate+std.error)) +
  facet_wrap(~term, scales = "free") +
  geom_pointrange(fatten = 1/2) +
  ylab("Estimates") +
  xlab("Simulations") +
  ggtitle('Estimates of Fixed Effects for 20 pct. Missing Data, ef = 0.3') +
  theme_bw()

fixed_missing_plot <- fixed_missing_plot + theme(plot.title = element_text(hjust = 0.5, size=20))
fixed_missing_plot
```



5.3.2 Visualise Estimates for Random Effects:

```
ran_missing_plot <- sims_20_missing_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 13)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects for 20 pct. Missing Data, ef =
    theme_bw()
ran_missing_plot <- ran_missing_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_missing_plot
```



5.4 Effect Size = 0.2

```
filename_20_missing_0.2 = "run_sims_20_missing_0.2.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.2,
    ef = 0.2))
end_time <- Sys.time()
end_time - start_time
```

5.5 Effect Size = 0.1

```
filename_20_missing_0.1 = "run_sims_20_missing_0.1.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.1,
```

```
    ef = 0.1))
end_time <- Sys.time()
end_time - start_time
```

6 Power Calculation with 50 pct. Missing Data and Varying Intercepts and Varying Slopes

6.1 Effect Size = 0.5

```
run_sims_missing <- function(filename_missing, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,

                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,

                         beta_cfa = ef)

  missing_samples <- dat_sim %>%
    mutate(nas = rbinom(nrow(dat_sim), 1, 1 - .50)) %>%
    mutate(DV = ifelse(nas == 1, DV, NA)) %>%
    drop_na()

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 + X_c * X_f | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=missing_samples)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_missing)
  write_csv(sim_results, filename_missing, append = append)

  # return the tidy table
  sim_results
}

filename_50_missing_0.5 = 'run_sims_50_missing_0.5.csv'
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.5, ef = 0.5))
end_time <- Sys.time()
end_time - start_time
```

6.2 Effect Size = 0.4

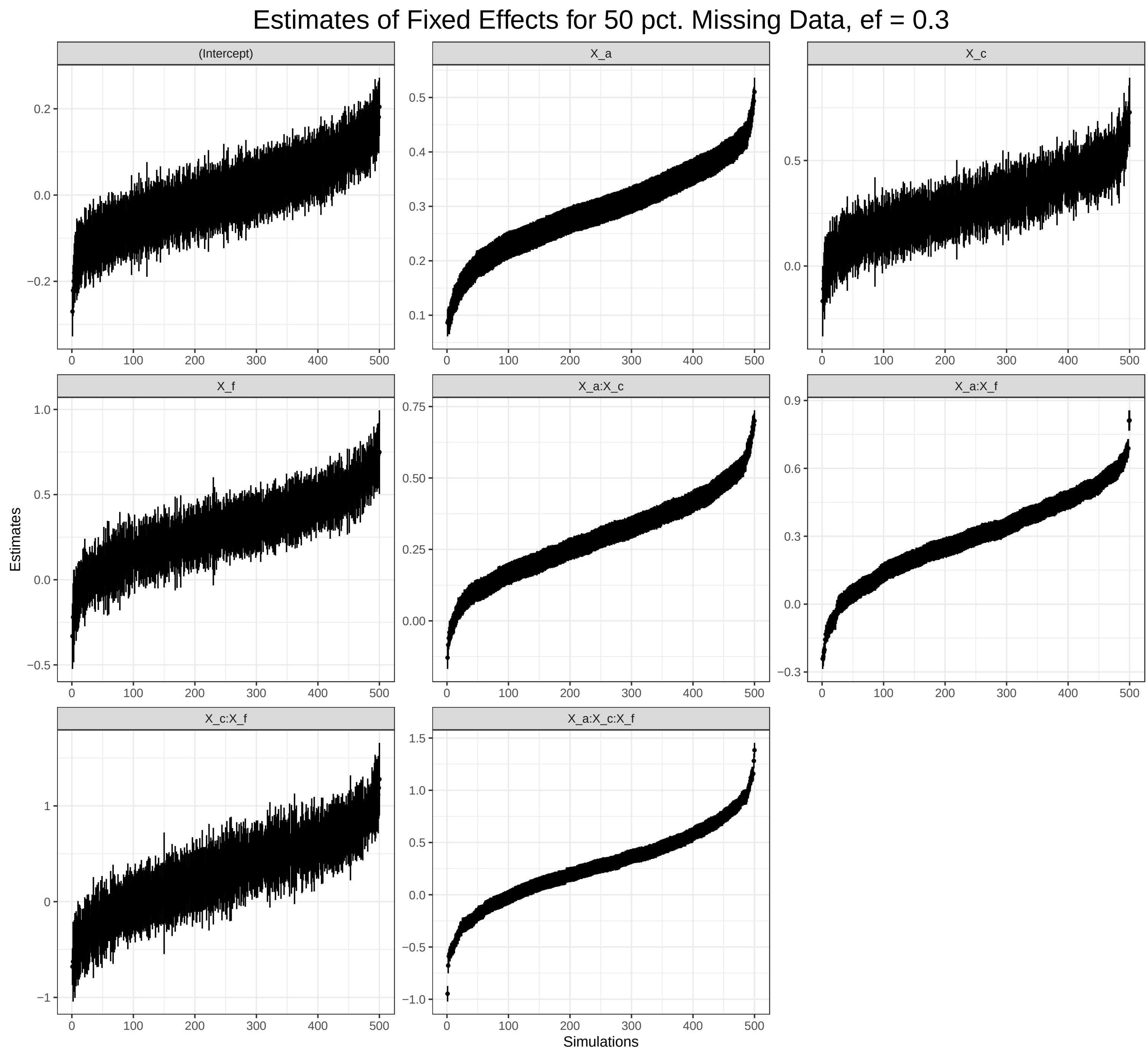
```
filename_50_missing_0.4 = "run_sims_50_missing_0.4.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

6.3 Effect Size = 0.3

```
filename_50_missing_0.3 = "run_sims_50_missing_0.3.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.3,
    ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

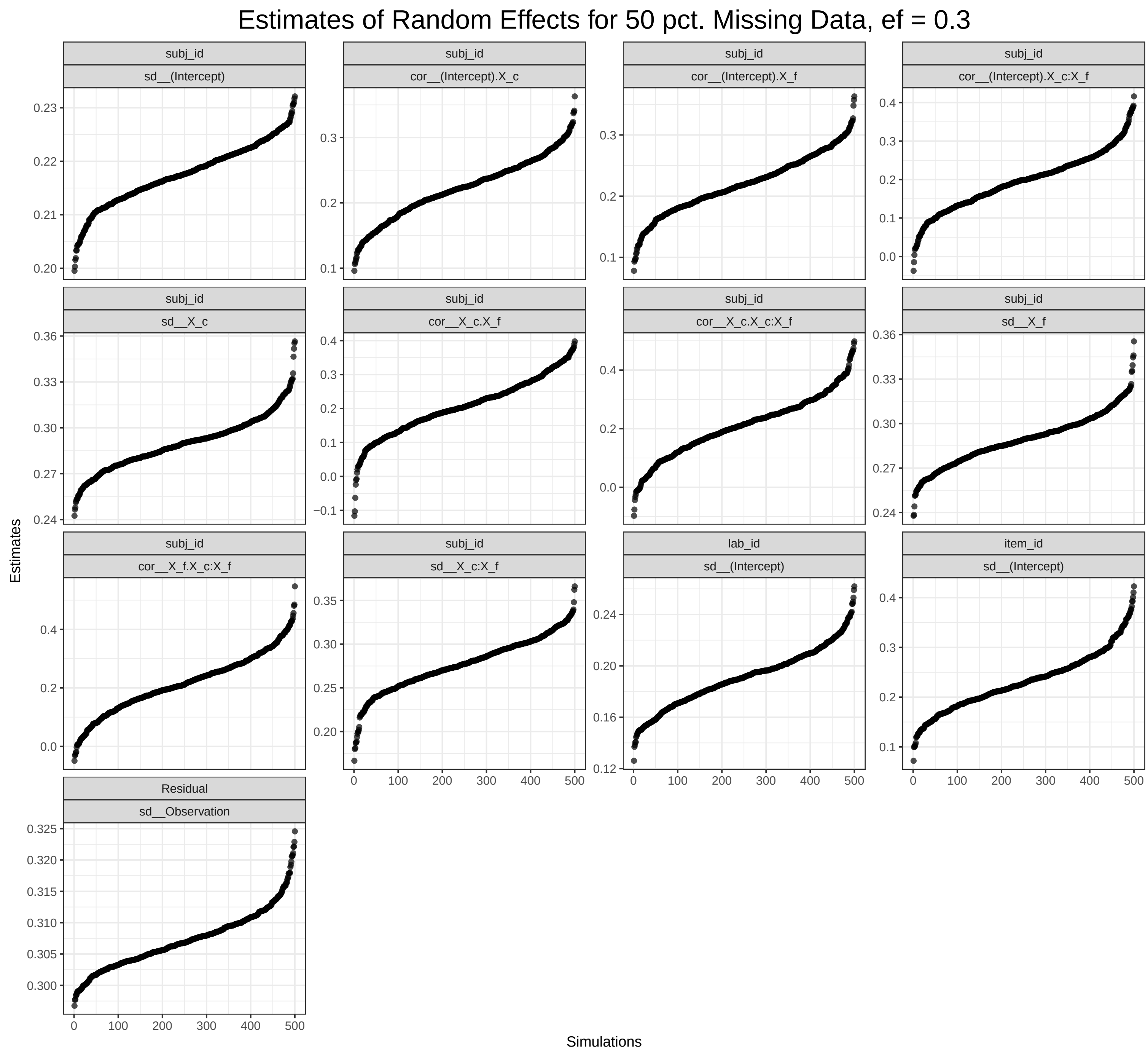
6.3.1 Visualise Estimates for Fixed Effects:

```
# read saved simulation data
sims_50_missing_0.3 <- read_csv(filename_50_missing_0.3, col_types = cols(
  # makes sure plots display in this order
  group = col_factor(ordered = TRUE),
  term = col_factor(ordered = TRUE)
))

fixed_missing_plot <- sims_50_missing_0.3 %>%
  filter(effect == "fixed") %>%
  ungroup() %>%
  arrange(term, estimate) %>%
  mutate(row = rep(seq(1:reps), 8)) %>%
  ggplot(aes(x = row, y = estimate, ymin = estimate-std.error, ymax = estimate+std.error)) +
  facet_wrap(~term, scales = "free") +
  geom_pointrange(fatten = 1/2) +
  ylab("Estimates") +
  xlab("Simulations") +
  ggtitle('Estimates of Fixed Effects for 50 pct. Missing Data, ef = 0.3') +
  theme_bw()

fixed_missing_plot <- fixed_missing_plot + theme(plot.title = element_text(hjust = 0.5, size=20))
fixed_missing_plot
```



6.3.2 Visualise Estimates for Random Effects:

```
ran_missing_plot <- sims_20_missing_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 13)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects for 50 pct. Missing Data, ef =
    theme_bw()
ran_missing_plot <- ran_missing_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_missing_plot
```



6.4 Effect Size = 0.2

```
filename_50_missing_0.2 = "run_sims_50_missing_0.2.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.2,
    ef = 0.2))
end_time <- Sys.time()
end_time - start_time
```

6.5 Effect Size = 0.1

```
filename_50_missing_0.1 = "run_sims_50_missing_0.1.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.1,
```

```
    ef = 0.1))
end_time <- Sys.time()
end_time - start_time
```

7 Overview of Power Simulation Results

7.1 Summary Statistics for Power Calculation with Full Data and Varying Intercepts and Varying Slopes:

Table 1: Power for Simulations with Full Data and Varying Intercepts and Varying Slopes

term	power, ef = 0.1	power, ef = 0.2	power, ef = 0.3	power, ef = 0.4	power, ef = 0.5
(Intercept)	0.046	0.056	0.062	0.054	0.054
X_a	0.778	0.968	0.998	1.000	1.000
X_c	0.094	0.254	0.458	0.718	0.902
X_f	0.106	0.206	0.330	0.546	0.696
X_a:X_c	0.736	0.868	0.960	0.994	0.998
X_a:X_f	0.762	0.808	0.920	0.970	0.988
X_c:X_f	0.066	0.068	0.128	0.208	0.268
X_a:X_c:X_f	0.778	0.822	0.860	0.884	0.910

7.2 Summary Statistics for Power Calculation with Full Data and Varying Intercepts:

Table 2: Power for Simulations with Full Data and Varying Intercepts

term	power, ef = 0.1	power, ef = 0.2	power, ef = 0.3	power, ef = 0.4	power, ef = 0.5
(Intercept)	0.050	0.042	0.052	0.052	0.052
X_a	0.776	0.984	1.000	1.000	1.000
X_c	0.108	0.258	0.504	0.892	0.712
X_f	0.108	0.190	0.340	0.718	0.556
X_a:X_c	0.822	0.908	0.990	1.000	0.998
X_a:X_f	0.816	0.882	0.942	0.996	0.980
X_c:X_f	0.078	0.080	0.140	0.260	0.202
X_a:X_c:X_f	0.786	0.828	0.862	0.906	0.890

7.3 Summary Statistics for Power Calculation with 20 pct. Missing Data and Varying Intercepts and Varying Slopes:

Table 3: Power for Simulations with 20 pct. Missing Data and Varying Intercepts and Slopes

term	power, ef = 0.1	power, ef = 0.2	power, ef = 0.3	power, ef = 0.4	power, ef = 0.5
(Intercept)	0.058	0.074	0.052	0.058	0.066
X_a	0.762	0.972	1.000	1.000	1.000
X_c	0.096	0.246	0.454	0.720	0.868
X_f	0.096	0.224	0.344	0.524	0.682
X_a:X_c	0.742	0.850	0.944	0.990	1.000
X_a:X_f	0.732	0.816	0.892	0.954	0.992
X_c:X_f	0.082	0.100	0.134	0.180	0.262
X_a:X_c:X_f	0.756	0.782	0.804	0.864	0.898

7.4 Summary Statistics for Power Calculation with 50 pct. Missing Data and Varying Intercepts and Varying Slopes:

Table 4: Power for Simulations with 50 pct. Missing Data and Varying Intercepts and Slopes

term	power, ef = 0.1	power, ef = 0.2	power, ef = 0.3	power, ef = 0.4	power, ef = 0.5
(Intercept)	0.046	0.054	0.054	0.060	0.060
X_a	0.738	0.970	1.000	1.000	1.000
X_c	0.078	0.226	0.456	0.710	0.852
X_f	0.080	0.182	0.354	0.530	0.682
X_a:X_c	0.644	0.816	0.944	0.980	1.000
X_a:X_f	0.678	0.824	0.890	0.960	0.992
X_c:X_f	0.074	0.102	0.138	0.198	0.244
X_a:X_c:X_f	0.702	0.774	0.774	0.798	0.868

8 Overview of Bias Results

Table 5: Bias for Simulations with Full Data and Varying Intercepts and Varying Slopes

term	bias, ef = 0.1	bias, ef = 0.2	bias, ef = 0.3	bias, ef = 0.4	bias, ef = 0.5
(Intercept)	-0.001	0.000	0.006	-0.002	-0.003
X_a	0.012	0.006	0.001	0.004	0.000
X_c	-0.009	0.008	0.005	0.012	-0.004
X_f	-0.002	0.003	0.020	0.004	-0.001
X_a:X_c	0.001	0.001	0.003	-0.015	0.001
X_a:X_f	-0.008	0.006	-0.009	-0.014	-0.009
X_c:X_f	-0.008	-0.002	-0.010	0.010	-0.016
X_a:X_c:X_f	0.013	-0.019	-0.004	0.013	0.012

Table 6: Bias for Simulations with Full Data and Varying Intercepts

term	bias, ef = 0.1	bias, ef = 0.2	bias, ef = 0.3	bias, ef = 0.4	bias, ef = 0.5
(Intercept)	-0.003	-0.006	0.002	0.000	-0.008
X_a	-0.001	-0.003	-0.003	-0.099	0.089
X_c	-0.011	0.008	-0.016	-0.097	0.105
X_f	-0.003	-0.002	0.001	-0.103	0.092
X_a:X_c	-0.003	0.014	0.003	-0.114	0.106
X_a:X_f	-0.008	0.002	0.010	-0.106	0.082
X_c:X_f	-0.029	-0.032	-0.006	-0.101	0.065
X_a:X_c:X_f	-0.016	-0.012	-0.002	-0.113	0.087

Table 7: Bias for Simulations with 20 pct. Missing Data and Varying Intercepts and Slopes

term	bias, ef = 0.1	bias, ef = 0.2	bias, ef = 0.3	bias, ef = 0.4	bias, ef = 0.5
(Intercept)	-0.004	0.002	0.002	-0.004	0.000
X_a	0.004	-0.001	0.000	-0.005	-0.009
X_c	0.010	0.009	0.006	-0.005	0.000
X_f	0.014	-0.014	-0.009	0.010	-0.001
X_a:X_c	-0.007	-0.009	0.004	-0.006	-0.004
X_a:X_f	-0.001	-0.002	0.000	0.006	-0.006
X_c:X_f	-0.017	-0.002	-0.009	0.013	-0.009
X_a:X_c:X_f	0.030	-0.016	0.046	0.013	0.025

Table 8: Bias for Simulations with 50 pct. Missing Data and Varying Intercepts and Slopes

term	bias, ef = 0.1	bias, ef = 0.2	bias, ef = 0.3	bias, ef = 0.4	bias, ef = 0.5
(Intercept)	-0.001	0.003	0.001	0.002	0.004
X_a	0.002	0.006	0.008	0.001	0.001
X_c	-0.007	0.002	0.010	0.002	0.004
X_f	-0.011	-0.016	-0.005	0.004	0.012
X_a:X_c	0.011	0.003	0.006	0.005	-0.014
X_a:X_f	0.004	-0.013	-0.001	-0.001	0.002
X_c:X_f	0.001	-0.044	-0.024	-0.001	-0.011
X_a:X_c:X_f	0.036	-0.018	0.022	0.010	0.039

9 Increasing Variance with Trial Number

9.1 Extracting estimates of increasing variance from the adult pilot data

```
# download data from adult pilot study:
adult_data <- read.csv("02_processed_lt_data.csv")

stimulus_info_df <- adult_data %>%
    filter(phase == "pref") %>%
    mutate(familiar_item = case_when(left_stimulus == "familiar" ~
        stimulus_processed_left, right_stimulus == "familiar" ~
        stimulus_processed_right), novel_item = case_when(left_stimulus ==
        "novel" ~ stimulus_processed_left, right_stimulus ==
        "novel" ~ stimulus_processed_right)) %>%
    select(subject, block_number, familiar_item, novel_item) %>%
    distinct(subject, block_number, .keep_all = TRUE)

lt_comparison_df <- adult_data %>%
    filter(phase == "pref") %>%
    group_by(subject, exposure_time, block_number, complexity,
        gaze_location_type) %>%
    summarise(sum_dwell_time = sum(dwell_time)) %>%
    pivot_wider(names_from = gaze_location_type, values_from = sum_dwell_time) %>%
    ungroup() %>%
    left_join(stimulus_info_df, by = c("subject", "block_number")) %>%
    rename(familiarization_time = exposure_time, stimulus_complexity = complexity,
        trial_number = block_number, familiar_looking_time = familiar,
        novel_looking_time = novel, participant = subject) %>%
    # currently excluding all the empty trials
filter(!is.na(familiar_looking_time) & !is.na(novel_looking_time))
```

```
## `summarise()` has grouped output by 'subject', 'exposure_time', 'block_number',
## 'complexity'. You can override using the `.groups` argument.
```

```
lt_comparison_df <- lt_comparison_df %>%
    mutate(familiarization_time_scaled_centered = log10(familiarization_time) -
        log10(2000), contrast_coded_complexity = ifelse(stimulus_complexity ==
        "complex", 0.5, -0.5))

proportion_df <- adult_data %>%
    filter(phase == "pref") %>%
    group_by(block_number, exposure_time, complexity, gaze_location_type,
        subject) %>%
    summarise(sum_dwell_time = sum(dwell_time, na.rm = TRUE)) %>%
    pivot_wider(names_from = gaze_location_type, values_from = sum_dwell_time) %>%
    mutate(familiar = ifelse(is.na(familiar), 0, familiar), novel = ifelse(is.na(novel),
```

```r
        0, novel)) %>%
    mutate(novelty_looking_proportion = novel/(familiar + novel))
```

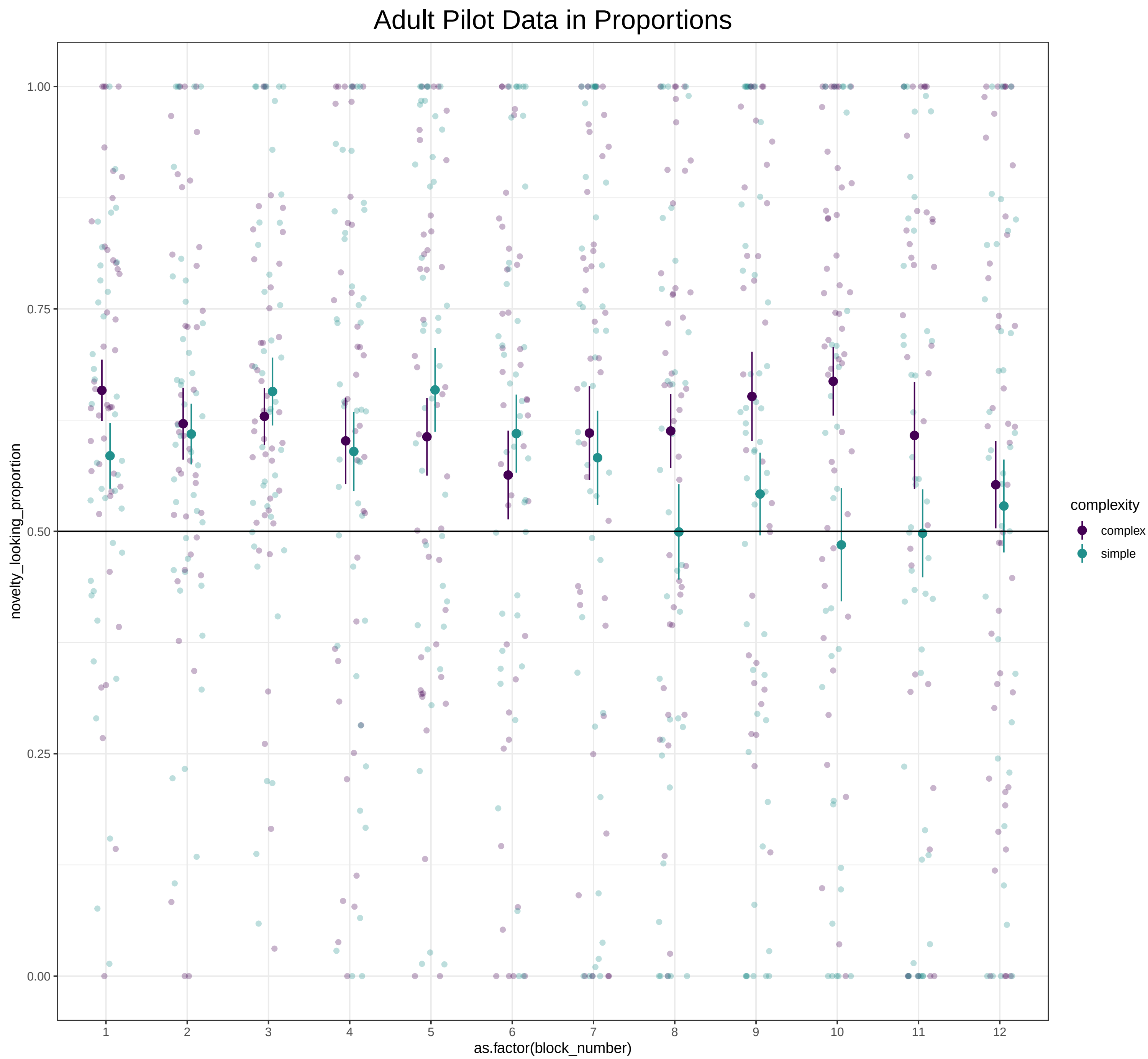
```
## `summarise()` has grouped output by 'block_number', 'exposure_time',
## 'complexity', 'gaze_location_type'. You can override using the `.groups`
## argument.
```

```r
dwell_proportion_mean <- mean(proportion_df$novelty_looking_proportion,
    na.rm = T)
dwell_proportion_sd <- sd(proportion_df$novelty_looking_proportion,
    na.rm = T)

adult_data_z <- proportion_df %>%
    mutate(dwell_time_z = (novelty_looking_proportion - dwell_proportion_mean)/dwell_proportion_sd)

data_on_increase_in_sd <- adult_data_z %>%
    group_by(block_number) %>%
    summarise(n = n(), dwell_time_median = median(dwell_time_z,
        na.rm = T), sd_block = sd(dwell_time_z, na.rm = T))

increase_sd_value <- data_on_increase_in_sd$sd_block[12] - data_on_increase_in_sd$sd_block[1]
increase_sd_value
```

```
## [1] 0.3098641
```

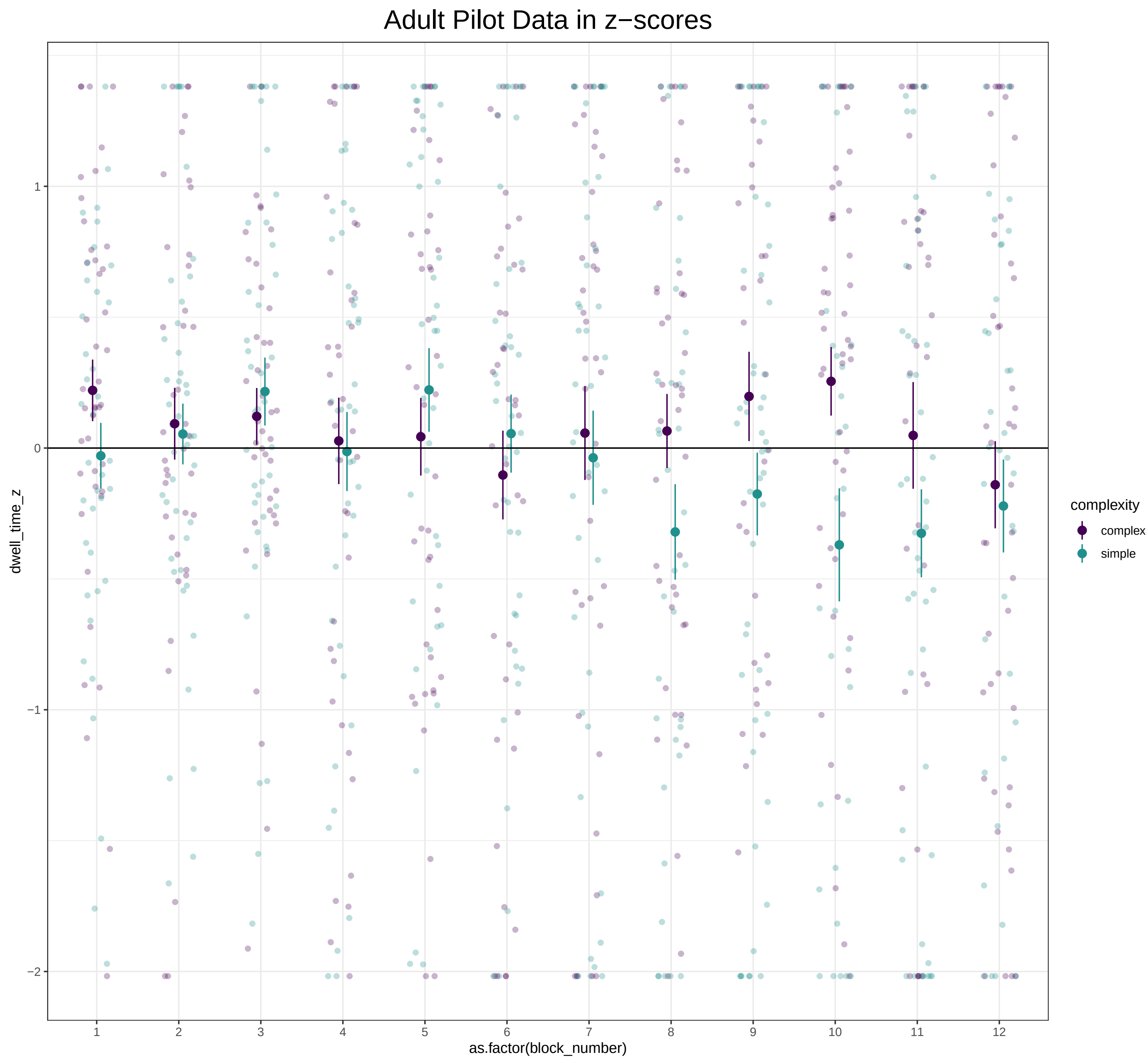
9.2 Visualisation of adult data in proportions

```r
adult_proportion_plot <- proportion_df %>%
    ggplot(aes(x = as.factor(block_number), y = novelty_looking_proportion,
        color = complexity)) + geom_jitter(alpha = 0.3, width = 0.2) +
    stat_summary(aes(x = as.factor(block_number), y = novelty_looking_proportion,
        color = complexity), position = position_dodge(width = 0.2)) +
    geom_hline(yintercept = 0.5) + scale_color_manual(values = viridis(n = 3)) +
    ggtitle("Adult Pilot Data in Proportions") + theme_bw()
adult_proportion_plot <- adult_proportion_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
adult_proportion_plot
```



9.3 Visualisation of adult data in z-score

```
adult_data_z_plot <- adult_data_z %>%
    ggplot(aes(x = as.factor(block_number), y = dwell_time_z,
        color = complexity)) + geom_jitter(alpha = 0.3, width = 0.2) +
    stat_summary(aes(x = as.factor(block_number), y = dwell_time_z,
        color = complexity), position = position_dodge(width = 0.2)) +
    geom_hline(yintercept = 0) + scale_color_manual(values = viridis(n = 3)) +
    ggtitle("Adult Pilot Data in z-scores") + theme_bw()

adult_data_z_plot <- adult_data_z_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
adult_data_z_plot
```



9.4 Modification of simulation function to include variance increase

```
my_sim_data_variance <- function(
    increase_sd = increase_sd_value, #Add variance to increase with trial number

    n_subj      = 1280,   # number of subjects
    n_simple  =  6,    # number of complex stimuli
    n_complex =  6,    # number of complex stimuli
    n_small_fam = 4,    #small familiarization time
    n_medium_fam = 4,   #medium familiarization time
    n_high_fam = 4,     #high familiarization time
    n_lab = 40,

    beta_0  =  0, # intercept; i.e., the grand mean
    beta_c  =  0.3, # main effect for complexity
    beta_f = 0.3, # main effect for familiarization time
```

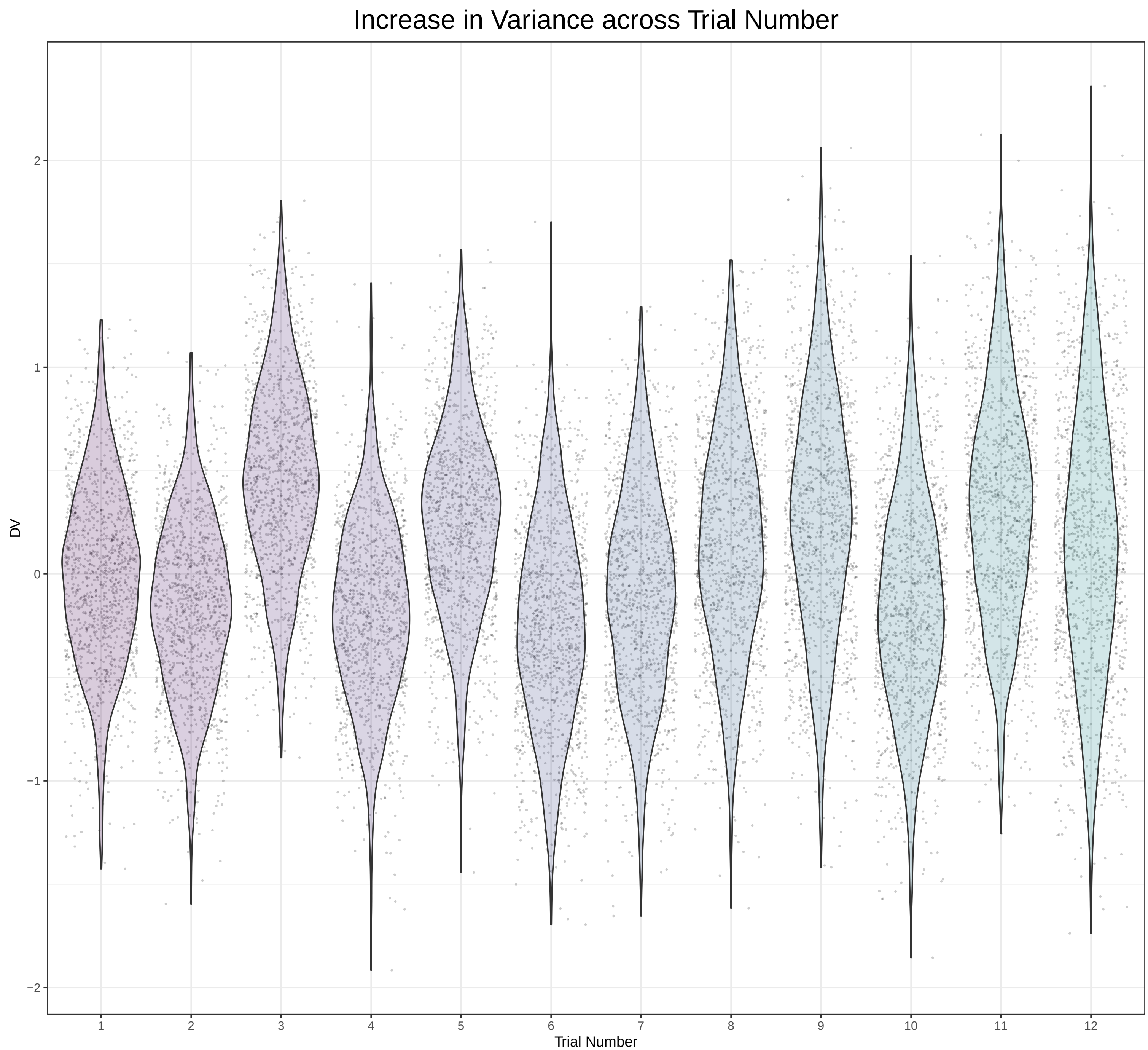
```
    beta_a = 0.3, # main effect for age

    beta_ca = 0.3,
    beta_af = 0.3,
    beta_cf = 0.3,

    beta_cfa = 0.3, #main effect for interaction between complexity and familiarization.

    subject_0   = 0.2, # by-subject random intercept sd

    subject_c   =  0.2, # by-subject slope complexity sd
    subject_f = 0.2, # by-subject slope familiarization sd
    subject_a = 0.2, # by-subject slope age sd

    subject_ca = 0.2,# by-subject slope for interaction between age and complexity sd
    subject_af = 0.2, # by-subject slope for interaction between age and familiarization sd
    subject_cf = 0.2, # by-subject slope complexity*familiarization sd

    subject_cfa = 0.2, # by-subject slope for interaction between age, complexity and familiarization sd

    subj_rho     =  .2, # correlations between by-subject random effects

    lab_0 = 0.2, # by-lab random intercept sd

    lab_c = 0.2, # by-lab slope complexity sd
    lab_f = 0.2, # by-lab slope familiarization sd
    lab_a = 0.2, # by-lab slope age sd

    lab_ca = 0.2, # by-lab slope for interaction between age and complexity sd
    lab_af = 0.2, # by-lab slope for interaction between age and familiarization sd
    lab_cf = 0.2, # by-lab random slope complexity*familiarization sd

    lab_cfa = 0.2, # by-lab slope for interaction between age, complexity and familiarization sd

    lab_rho = 0.2, # correlations between by-lab random effects

    item_0 =  0.2, # by-item random intercept sd

    item_c = 0.2, # by-item slope complexity sd
    item_f = 0.2, # by-item slope familiarization sd
    item_a = 0.2, # by-item slope age sd

    item_ca = 0.2, # by-item slope for interaction between age and complexity sd
    item_af = 0.2, # by-item slope for interaction between age and familiarization sd
    item_cf = 0.2, # by-item random slope complexity*familiarization sd

    item_cfa = 0.2, # by-item slope for interaction between age, complexity and familiarization sd

    item_rho = 0.2, # correlations between by-item random effects


    sigma = 0.3 # residual (error) sd
    ) { # residual (standard deviation)

  # simulate a sample of items
  items <- data.frame(
    item_id = seq_len(n_simple + n_complex),
    category = rep(c("simple", "complex"), c(n_simple, n_complex)),
    X_c = rep(c(-0.5, 0.5), c(n_simple, n_complex)),
    familiarization = rep(c("short", "medium", "long"), (n_simple + n_complex)/3),
    X_f = rep(c(-0.5, 0, 0.5), (n_simple + n_complex)/3),
    faux::rnorm_multi(
```

```
  n = n_simple + n_complex, mu = 0, sd = c(item_0,
                              item_c,
                              item_f,
                              item_a,
                              item_ca,
                              item_af,
                              item_cf,
                              item_cfa), r = item_rho,
  varnames = c("I_0", "I_c","I_f","I_a",
               "I_ca","I_af", "I_cf",
               "I_cfa"))
  ) %>%
  mutate(item_id = faux::make_id(nrow(.), "I"))

# simulate a sample of subjects
subjects <-
  faux::rnorm_multi(
  n = n_subj, mu = 0, sd = c(subject_0,
                              subject_c,
                              subject_f,
                              subject_a,
                              subject_ca,
                              subject_af,
                              subject_cf,
                              subject_cfa), r = subj_rho,
  varnames = c("S_0", "S_c","S_f","S_a",
               "S_ca","S_af", "S_cf",
               "S_cfa")
) %>%
  mutate(subj_id = faux::make_id(nrow(.), "S")) %>%
  mutate(X_a = runif(n_subj, min = -0.5, max = 0.5))
#add subject age measure, sample from distribution from -0.5 to 0.5. #subjects$subj_id <- 1:n_subj

labs <- faux::rnorm_multi(
  n = n_lab, mu = 0, sd = c(lab_0, lab_c, lab_f, lab_a,
                             lab_ca, lab_af, lab_cf,
                             lab_cfa), r = lab_rho,
  varnames = c("L_0", "L_c","L_f","L_a",
               "L_ca","L_af", "L_cf",
               "L_cfa")
) %>%
  mutate(lab_id = faux::make_id(nrow(.), "L"))

#create lab and subj nesting structure
#Number of subjects must be a multiple of number of labs
lab_multiplier = n_subj/n_lab
lab_subj_dict <- data.frame(
  subj_id = subjects$subj_id,
  lab_id = rep(labs$lab_id,lab_multiplier)
)

# cross subject and item IDs
temp <- crossing(subjects, items)  %>%
  left_join(lab_subj_dict, by = "subj_id") %>%
  left_join(labs, by = "lab_id") %>%
  group_by(subj_id, item_id) %>% mutate(item_id = sample(item_id)) %>%
  ungroup() %>%
  mutate(trial_num = rep(seq(n_simple + n_complex), n_subj))

temp <- temp %>%
  mutate(trial_i_var = ifelse(trial_num > 0, rnorm(nrow(temp), 0, trial_num*(increase_sd / 12)), 0))
```

```r
  temp %>%
    mutate(
      B_0  = beta_0 + S_0 + L_0 + I_0,

      B_c  = beta_c + S_c + L_c + I_c,
      B_f  = beta_f + S_f + L_f + I_f,
      B_a = beta_a + S_a + L_a + I_a,

      B_ca = beta_ca + S_ca + L_ca + I_ca,
      B_af = beta_af + S_af + L_af + I_af,
      B_cf = beta_cf + S_cf + L_cf + I_cf,

      B_cfa = beta_cfa + S_cfa + L_cfa + I_cfa,

      e_si = rnorm(nrow(temp), mean = 0, sd = sigma) + trial_i_var,

      DV = B_0 +
        (B_a * X_a) + (B_c * X_c) + (B_f * X_f) +
        (B_cf * X_c * X_f) + (B_af * X_a * X_f) + (B_ca * X_c * X_a) +
        (B_cfa * X_c * X_f * X_a) + e_si
    )
}

dat_sim <- my_sim_data_variance()
```

9.5 Visualise the variance increase

```r
# The increase in variance is over 24 trials in this
# ManyBabies5 case:
dat_sim <- my_sim_data_variance(increase_sd = increase_sd_value)

plot_trial_num <- dat_sim %>%
    mutate(trial_num = as.factor(trial_num)) %>%
    ggplot() + geom_point(aes(y = DV, x = trial_num), position = "jitter",
    alpha = 0.2, size = 0.2) + geom_violin(aes(y = DV, x = trial_num,
    fill = trial_num), alpha = 0.2, show.legend = FALSE) + scale_fill_manual(values = viridis(n = 24)) +
    ggtitle("Increase in Variance across Trial Number") + xlab("Trial Number") +
    theme_bw()

plot_trial_num <- plot_trial_num + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
plot_trial_num
```



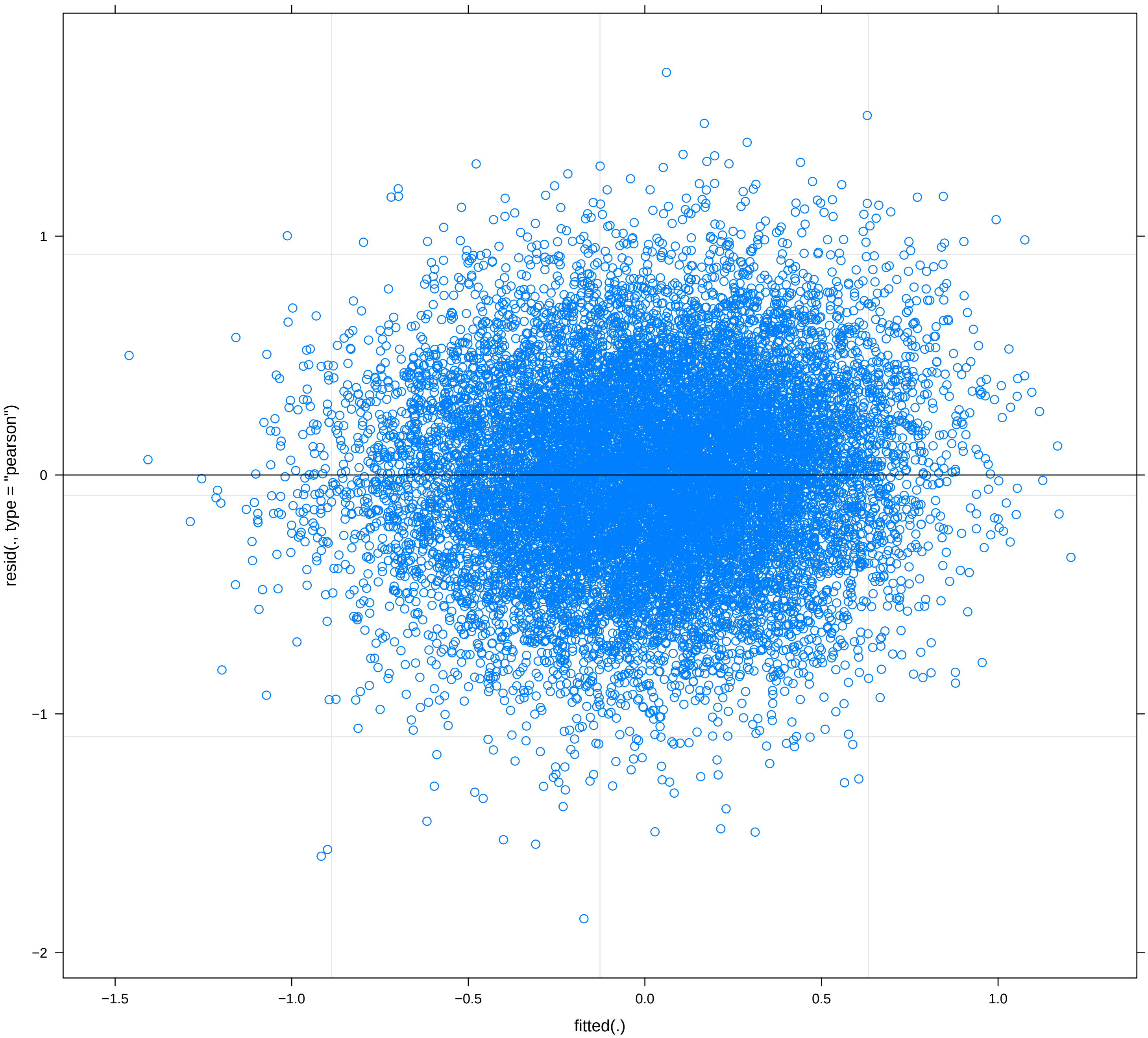
9.6 Model building and bias assessment

```
dat_sim <- my_sim_data_variance(increase_sd = increase_sd_value)

mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f + (1 | subj_id) + (1 |
    lab_id) + (1 | item_id), data = dat_sim)

summary(mod_sim)
```

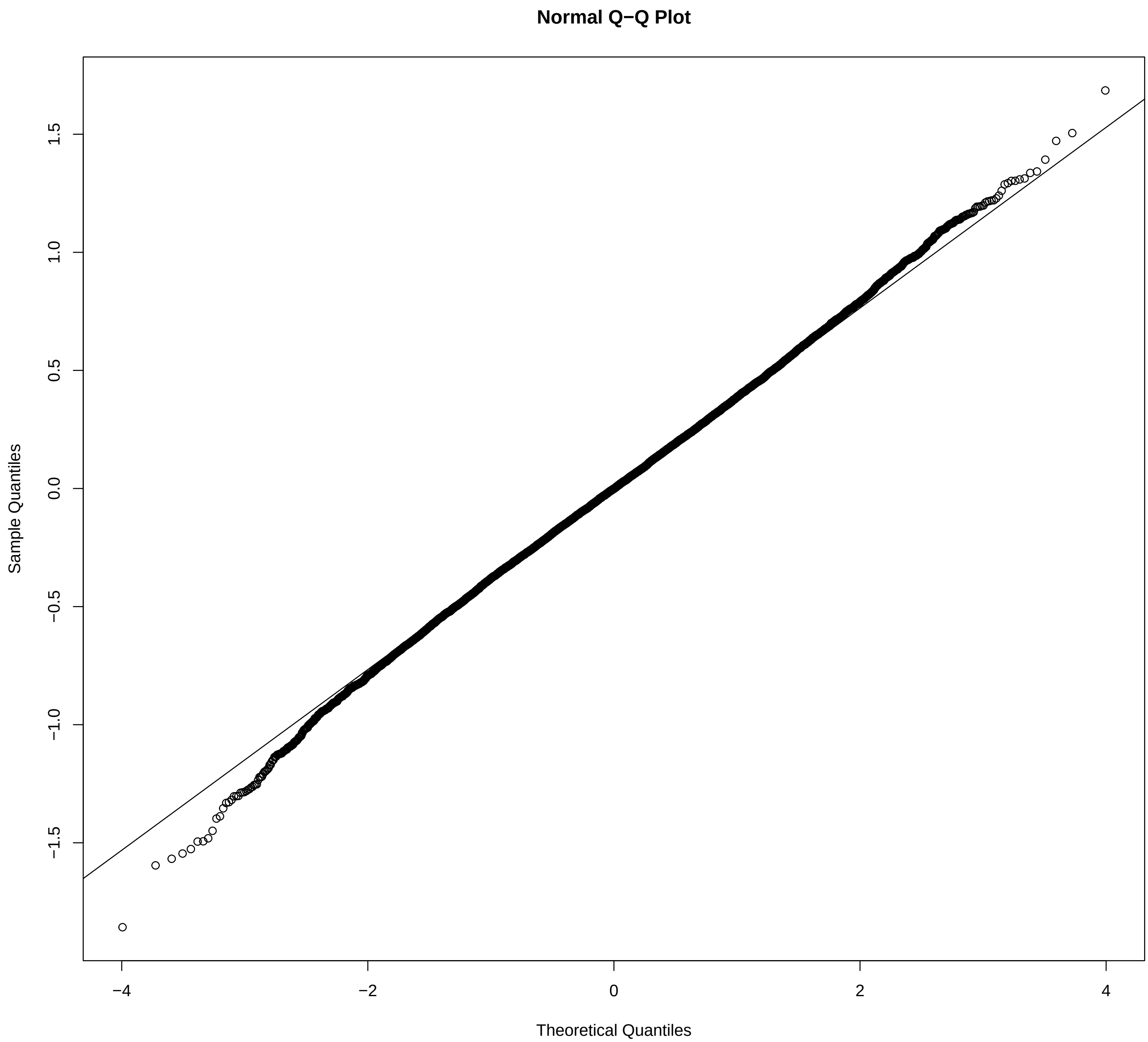
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: DV ~ 1 + X_a * X_c * X_f + (1 | subj_id) + (1 | lab_id) + (1 |
##     item_id)
##    Data: dat_sim
##
```

```
## REML criterion at convergence: 17955.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5706 -0.6382 -0.0016  0.6318  4.1470
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj_id  (Intercept) 0.04247  0.2061
##  lab_id   (Intercept) 0.05442  0.2333
##  item_id  (Intercept) 0.03959  0.1990
##  Residual             0.16517  0.4064
## Number of obs: 15360, groups:  subj_id, 1280; lab_id, 40; item_id, 12
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept) 1.452e-02  6.859e-02 1.558e+01   0.212    0.835
## X_a         1.661e-01  2.341e-02 1.242e+03   7.095 2.17e-12 ***
## X_c         8.363e-02  1.151e-01 8.002e+00   0.727    0.488
## X_f         2.491e-01  1.409e-01 8.002e+00   1.767    0.115
## X_a:X_c     9.519e-02  2.280e-02 1.407e+04   4.176 2.99e-05 ***
## X_a:X_f     3.546e-02  2.792e-02 1.407e+04   1.270    0.204
## X_c:X_f     1.605e-01  2.819e-01 8.002e+00   0.569    0.585
## X_a:X_c:X_f 7.225e-02  5.584e-02 1.407e+04   1.294    0.196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) X_a   X_c   X_f   X_:X_c X_:X_f X_c:X_
## X_a         0.000
## X_c         0.000  0.000
## X_f         0.000  0.000 0.000
## X_a:X_c     0.000  0.000 0.000 0.000
## X_a:X_f     0.000  0.000 0.000 0.000 0.000
## X_c:X_f     0.000  0.000 0.000 0.000 0.000  0.000
## X_a:X_c:X_f 0.000  0.000 0.000 0.000 0.000  0.000  0.000
```

```
plot(mod_sim)
```




```
qqnorm(resid(mod_sim))
qqline(resid(mod_sim))
```




```
check_normality(mod_sim)
```

```
## OK: residuals appear as normally distributed (p = 0.258).
```

```
check_outliers(mod_sim)
```

```
## OK: No outliers detected.
## - Based on the following method and threshold: cook (0.92).
## - For variable: (Whole model)
```

```
check_collinearity(mod_sim)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
```

```
##         Term  VIF  VIF 95% CI Increased SE Tolerance Tolerance 95% CI
##          X_a 1.00 [1.00,    ]         1.00      1.00     [    , 1.00]
##          X_c 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
##          X_f 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
##      X_a:X_c 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
##      X_a:X_f 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
##      X_c:X_f 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
##  X_a:X_c:X_f 1.00 [1.00, Inf]         1.00      1.00     [0.00, 1.00]
```

```
check_heteroscedasticity(mod_sim)
```

```
## OK: Error variance appears to be homoscedastic (p = 0.804).
```

9.7 Test how many of the models violate homoskedasticity

```
reps <- 200

run_sims_heteroskedasticity <- function(filename_heteroskedasticity) {

    dat_sim <- my_sim_data_variance(increase_sd = increase_sd_value)

    mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f + (1 | subj_id) +
        (1 | lab_id) + (1 | item_id), data = dat_sim)

    heteroskedasticity_results <- as_tibble(check_heteroscedasticity(mod_sim)[1])

    # append the results to a file
    append <- file.exists(filename_heteroskedasticity)
    write_csv(heteroskedasticity_results, filename_heteroskedasticity,
        append = append)

    # return the tidy table
    heteroskedasticity_results
}

filename_heteroskedasticity = "filename_heteroskedasticity.csv"
start_time <- Sys.time()
heteroskedasticity_results_data <- purrr::map_df(1:reps, ~run_sims_heteroskedasticity(filename_heteroskedasticity
end_time <- Sys.time()
end_time - start_time
```

9.8 Overview of how many of the models exhibit homoskedasticity

```
heteroskedasticity_results <- read.csv(filename_heteroskedasticity)

homoskedastic_models <- heteroskedasticity_results %>%
    filter(value > 0.05)

print(paste0(nrow(homoskedastic_models), "/", nrow(heteroskedasticity_results),
    " of the models exhibit homoskedasticity"))
```

```
## [1] "195/200 of the models exhibit homoskedasticity"
```

9.9 Bayesian robust location-scale regression model

```
model_formula <- bf(DV ~ 1 + X_a * X_c * X_f +
                      (1 | subj_id) + (1 | lab_id) + (1 | item_id),
                    sigma ~ 1 + trial_num + (1 | subj_id) + (1 | lab_id) + (1 | item_id))

#get_prior(model_formula, data = dat_sim, family = student)

priors1 <- c(prior(normal(0, 0.5), class = Intercept),
             prior(normal(0, 0.5), class = b),
             prior(normal(0, 0.5), class = b, dpar = sigma),
             prior(normal(0.25, 0.3), class = sd),
             prior(gamma(2, 0.1), class = nu)
             )

heteroskedasticity_fit <-
  brm(data = dat_sim,
           family = student,
           model_formula,
           prior = priors1,
           sample_prior = "yes",
           iter = 4000,
           warmup = 500,
           #backend = "cmdstanr",
           #threads = threading(2),
           file = "heteroskedasticity_fit",
           cores = 64,
           chains = 2,
           save_pars = save_pars(all = TRUE))

pp_check(heteroskedasticity_fit, ndraws = 100)

heteroskedasticity_fit
```

9.9.1 Prior-Posterior Update Checks

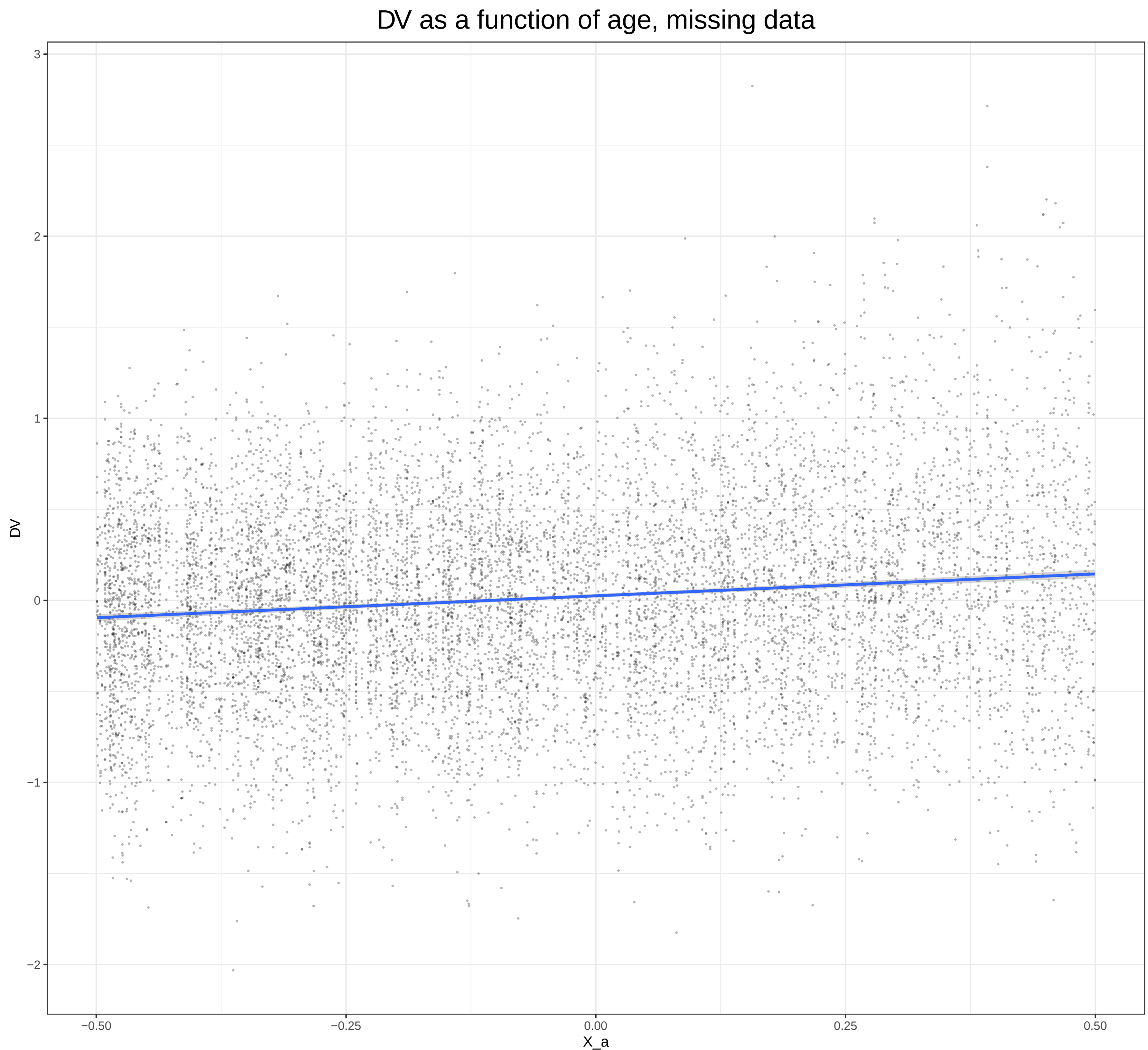
```
# Sample the parameters of interest:
Posterior_m1 <- as_draws_df(heteroskedasticity_fit)

# Plot the prior-posterior update plot for the intercept:
ggplot(Posterior_m1) + geom_density(aes(prior_Intercept), fill = "steelblue",
    color = "black", alpha = 0.6) + geom_density(aes(b_Intercept),
    fill = "#FC4E07", color = "black", alpha = 0.6) + theme_classic()

ggplot(Posterior_m1) + geom_density(aes(prior_b_sigma), fill = "steelblue",
    color = "black", alpha = 0.6) + geom_density(aes(b_sigma_trial_num),
    fill = "#FC4E07", color = "black", alpha = 0.6) + theme_classic()

ggplot(Posterior_m1) + geom_density(aes(prior_sd_item_id), fill = "steelblue",
    color = "black", alpha = 0.6) + geom_density(aes(sd_subj_id__Intercept),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(sd_lab_id__Intercept),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(sd_item_id__Intercept),
    fill = "#FC4E07", color = "black", alpha = 0.6) + theme_classic()

ggplot(Posterior_m1) + geom_density(aes(prior_b), fill = "steelblue",
    color = "black", alpha = 0.6) + geom_density(aes(b_X_a),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(b_X_c),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(b_X_f),
    fill = "#FC4E07", color = "black", alpha = 0.6) + theme_classic()

ggplot(Posterior_m1) + geom_density(aes(prior_b), fill = "steelblue",
```

```
    color = "black", alpha = 0.6) + geom_density(aes(`b_X_a:X_c`),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(`b_X_a:X_f`),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(`b_X_c:X_f`),
    fill = "#FC4E07", color = "black", alpha = 0.6) + geom_density(aes(`b_X_a:X_c:X_f`),
    fill = "#FC4E07", color = "black", alpha = 0.6) + theme_classic()

ggplot(Posterior_m1) + geom_density(aes(prior_nu), fill = "steelblue",
    color = "black", alpha = 0.6) + geom_density(aes(nu), fill = "#FC4E07",
    color = "black", alpha = 0.6) + theme_classic()
```

10 Data missing not completely at random

10.1 Simulation of missing data according to increasing infant age

```
# back to assumptions of equal variance across trials:
dat_sim <- my_sim_data()

# Proportion of missing data increases with age:
missing_samples_age <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(X_a > -0.5, (X_a +
        0.5) * 0.5, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

missing_age_plot <- missing_samples_age %>%
    drop_na() %>%
    ggplot() + ggtitle("DV as a function of age, missing data") +
    geom_point(aes(x = X_a, y = DV), alpha = 0.3, size = 0.2,
        position = "jitter") + geom_smooth(aes(x = X_a, y = DV),
    method = "lm", se = TRUE, formula = y ~ x) + theme_bw()

missing_age_plot <- missing_age_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
missing_age_plot
```



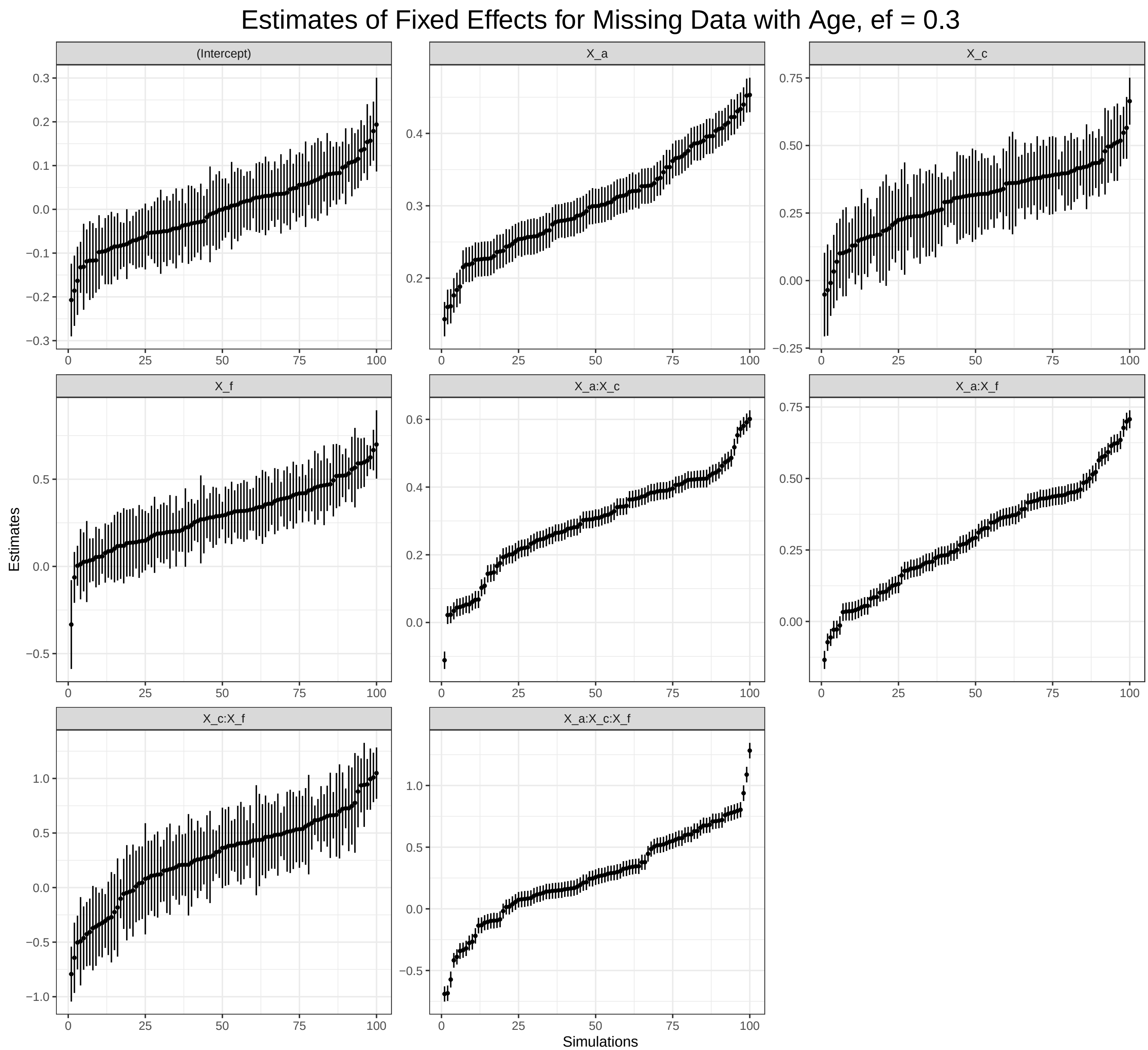
10.1.1 Simulation of models

```
# Number of simulations:
reps <- 100

# Simulation function:
run_sims <- function(filename_full, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,

                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,
```

```
                         beta_cfa = ef)

  missing_samples_age <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(X_a > -0.5, (X_a+0.5)*0.50, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=missing_samples_age)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_full)
  write_csv(sim_results, filename_full, append = append)

  # return the tidy table
  sim_results
}

filename_full_0.3_missing_age = 'sims/run_sims_0.3_age_missing.csv'
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.3_missing_age, ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

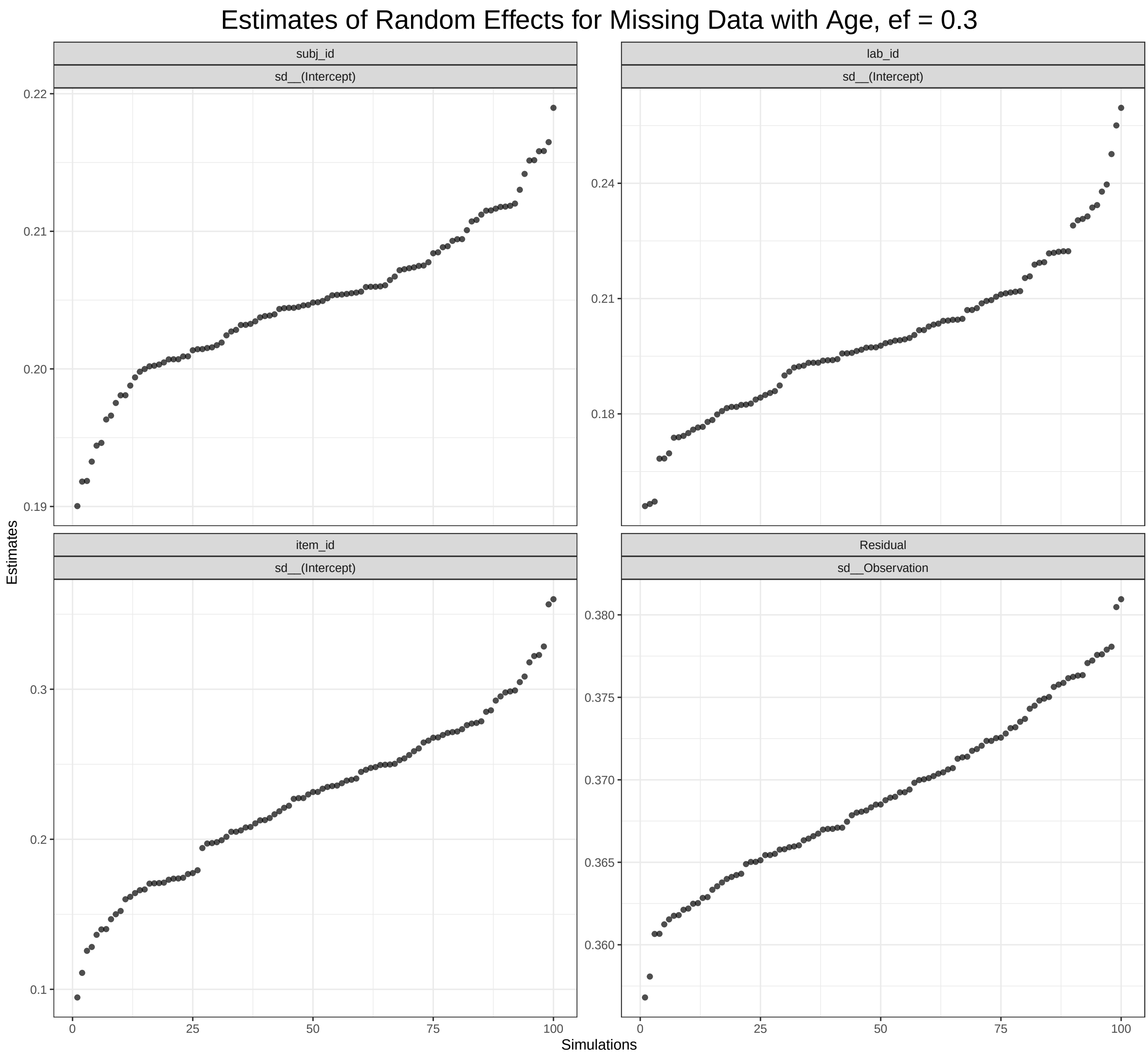
10.1.2 Visualise Estimates for Fixed Effects:

```
# read saved simulation data
sims_50_missing_age_0.3 <- read_csv(filename_full_0.3_missing_age, col_types = cols(
  # makes sure plots display in this order
  group = col_factor(ordered = TRUE),
  term = col_factor(ordered = TRUE)
))

reps <- 100

fixed_missing_age_plot <- sims_50_missing_age_0.3 %>%
  filter(effect == "fixed") %>%
  ungroup() %>%
  arrange(term, estimate) %>%
  mutate(row = rep(seq(1:reps), 8)) %>%
  ggplot(aes(x = row, y = estimate, ymin = estimate-std.error, ymax = estimate+std.error)) +
  facet_wrap(~term, scales = "free") +
  geom_pointrange(fatten = 1/2) +
  ylab("Estimates") +
  xlab("Simulations") +
  ggtitle('Estimates of Fixed Effects for Missing Data with Age, ef = 0.3') +
  theme_bw()

fixed_missing_age_plot <- fixed_missing_age_plot + theme(plot.title = element_text(hjust = 0.5, size=20))
fixed_missing_age_plot
```



10.1.3 Visualise Estimates for Random Effects:

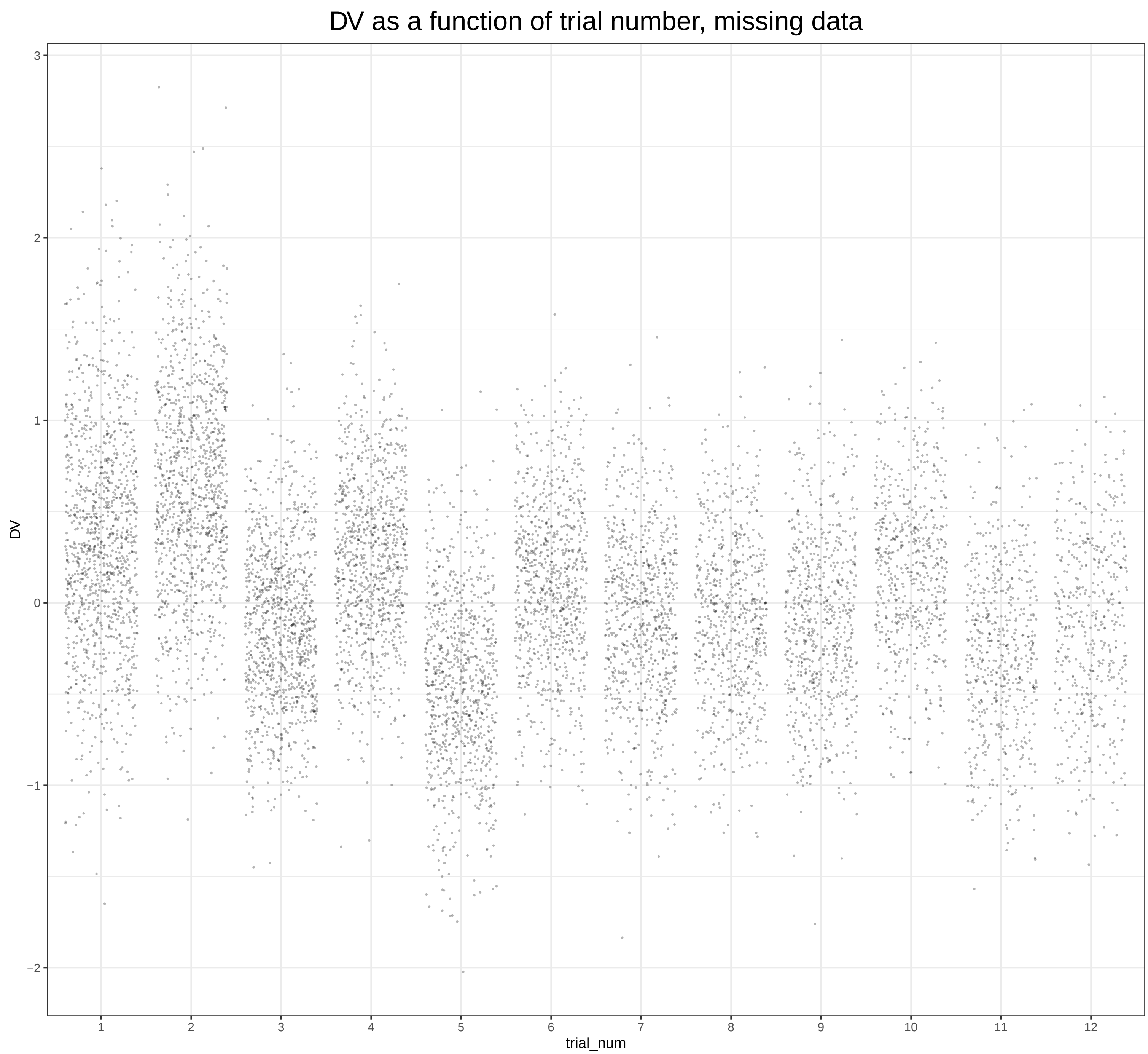
```
ran_missing_age_plot <- sims_50_missing_age_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 4)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects for Missing Data with Age, ef
    theme_bw()
ran_missing_age_plot <- ran_missing_age_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_missing_age_plot
```



10.2 Simulation of missing data according to increasing trial number

```
# Proportion of missing data increases with trial number:
missing_samples_trial <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(trial_num > 0,
        trial_num * 0.05, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

missing_trialnum_plot <- missing_samples_trial %>%
    mutate(trial_num = as.factor(trial_num)) %>%
    drop_na() %>%
    ggplot() + geom_point(aes(x = trial_num, y = DV), alpha = 0.3,
    size = 0.2, position = "jitter") + geom_smooth(aes(x = trial_num,
    y = DV), method = "lm", se = TRUE, formula = y ~ x) + ggtitle("DV as a function of trial number, missing data
    theme_bw()
```

```
missing_trialnum_plot <- missing_trialnum_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
missing_trialnum_plot
```



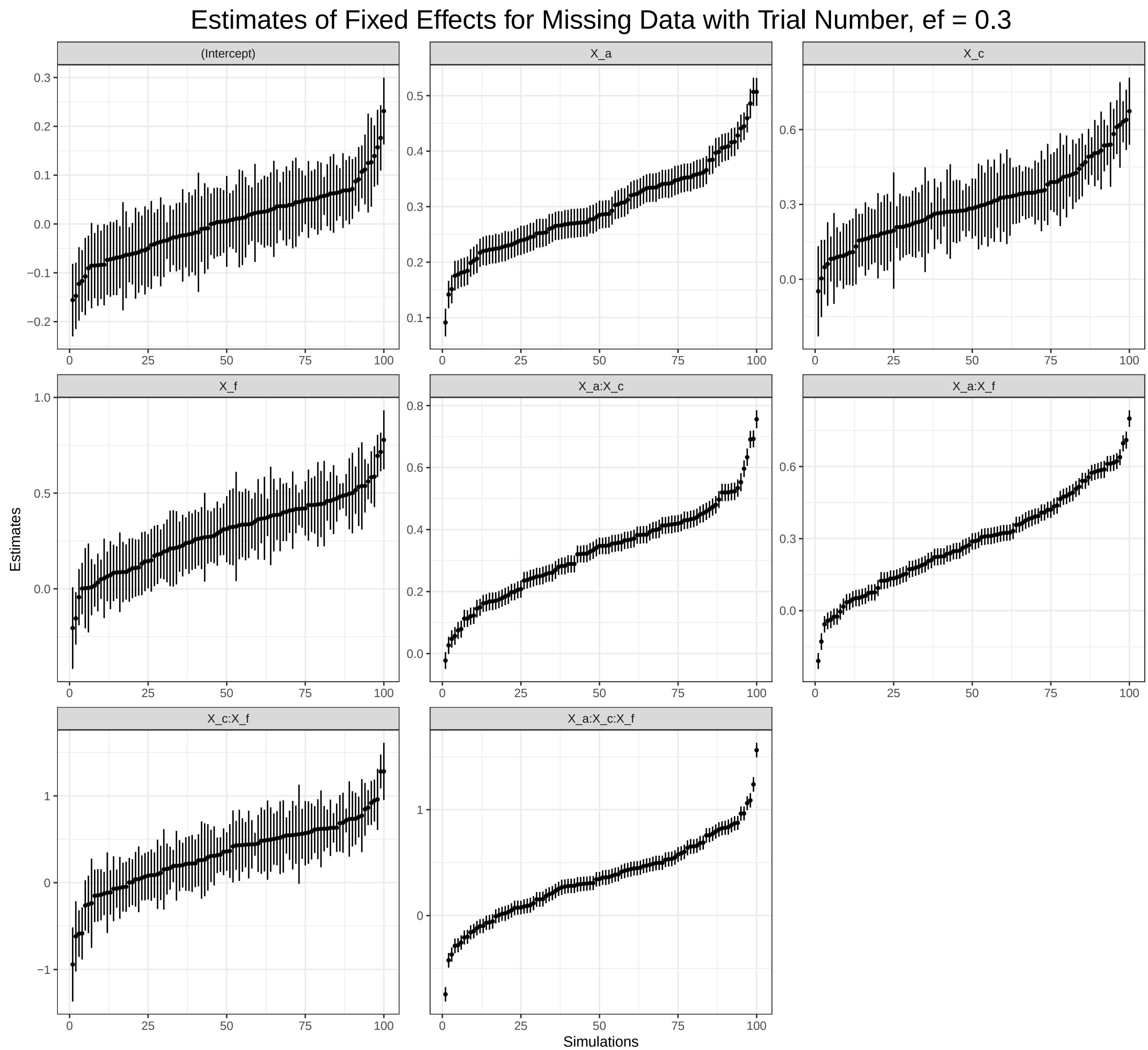
10.2.1 Simulation of models

```
# Number of simulations:
reps <- 100

# Simulation function:
run_sims <- function(filename_full, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,
```

```
                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,

                         beta_cfa = ef)

  missing_samples_trial <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(trial_num > 0, trial_num*0.05, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=missing_samples_trial)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_full)
  write_csv(sim_results, filename_full, append = append)

  # return the tidy table
  sim_results
}

filename_full_0.3_missing_trial = 'sims/run_sims_0.3_trial_missing.csv'
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.3_missing_trial, ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

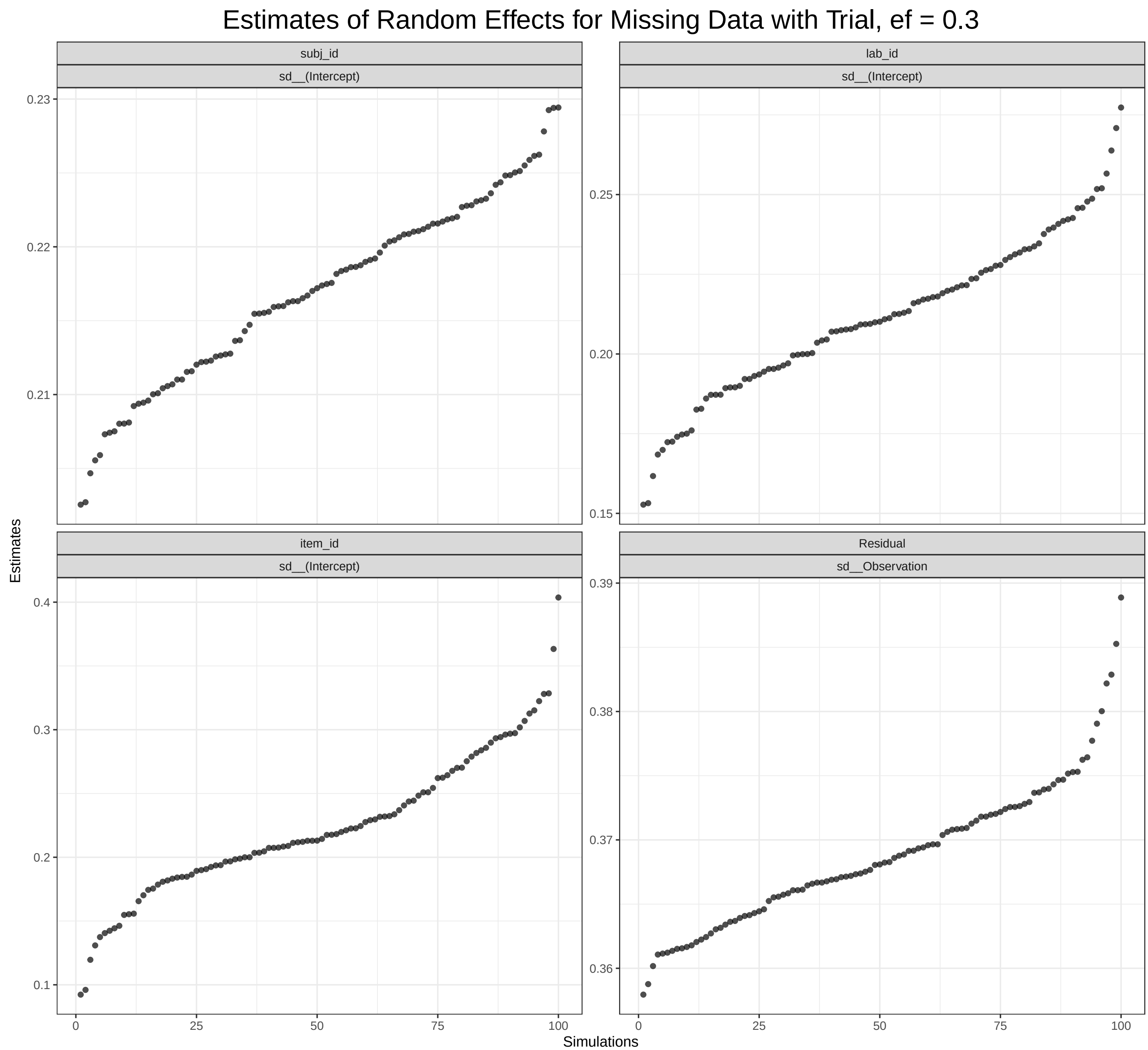
10.2.2 Visualise Estimates for Fixed Effects:

```
# read saved simulation data
sims_50_missing_trial_0.3 <- read_csv(filename_full_0.3_missing_trial, col_types = cols(
  # makes sure plots display in this order
  group = col_factor(ordered = TRUE),
  term = col_factor(ordered = TRUE)
))

fixed_missing_trial_plot <- sims_50_missing_trial_0.3 %>%
  filter(effect == "fixed") %>%
  ungroup() %>%
  arrange(term, estimate) %>%
  mutate(row = rep(seq(1:reps), 8)) %>%
  ggplot(aes(x = row, y = estimate, ymin = estimate-std.error, ymax = estimate+std.error)) +
  facet_wrap(~term, scales = "free") +
  geom_pointrange(fatten = 1/2) +
  ylab("Estimates") +
  xlab("Simulations") +
  ggtitle('Estimates of Fixed Effects for Missing Data with Trial Number, ef = 0.3') +
  theme_bw()

fixed_missing_trial_plot <- fixed_missing_trial_plot + theme(plot.title = element_text(hjust = 0.5, size=20))
fixed_missing_trial_plot
```



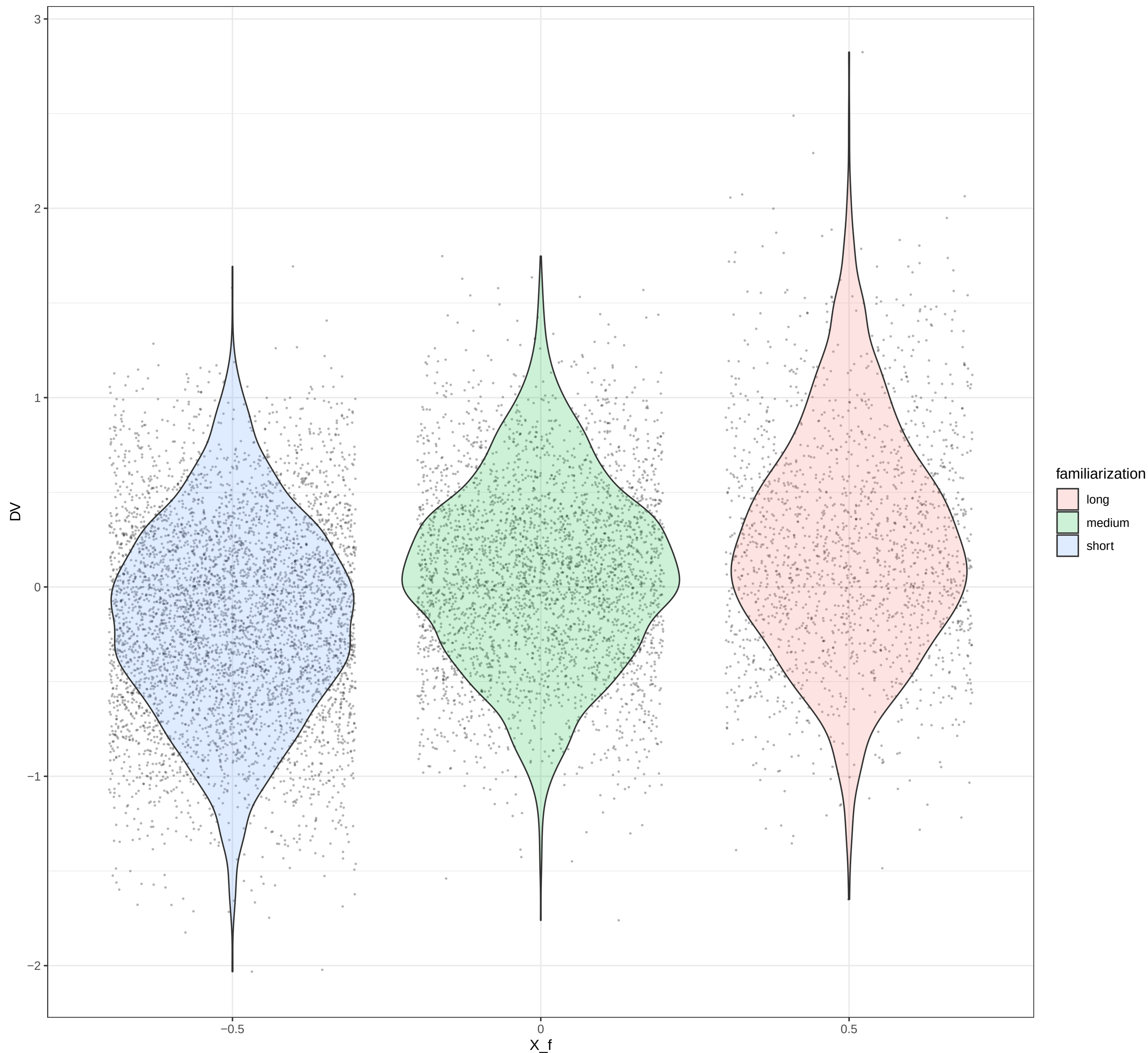
10.2.3 Visualise Estimates for Random Effects:

```
ran_missing_trial_plot <- sims_50_missing_trial_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 4)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects for Missing Data with Trial, e
    theme_bw()
ran_missing_trial_plot <- ran_missing_trial_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_missing_trial_plot
```



10.3 Simulation of missing data according to increasing familiarisation

```
# Proportion of missing data increases with
# familiarisation:
missing_samples_fam <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(X_f > -0.5, (X_f +
        0.5) * 0.6, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

missing_samples_fam %>%
    mutate(X_f = as.factor(X_f)) %>%
    drop_na() %>%
    ggplot() + geom_point(aes(x = X_f, y = DV), alpha = 0.3,
    size = 0.2, position = "jitter") + geom_violin(aes(y = DV,
    x = X_f, fill = familiarization), alpha = 0.2) + theme_bw()
```



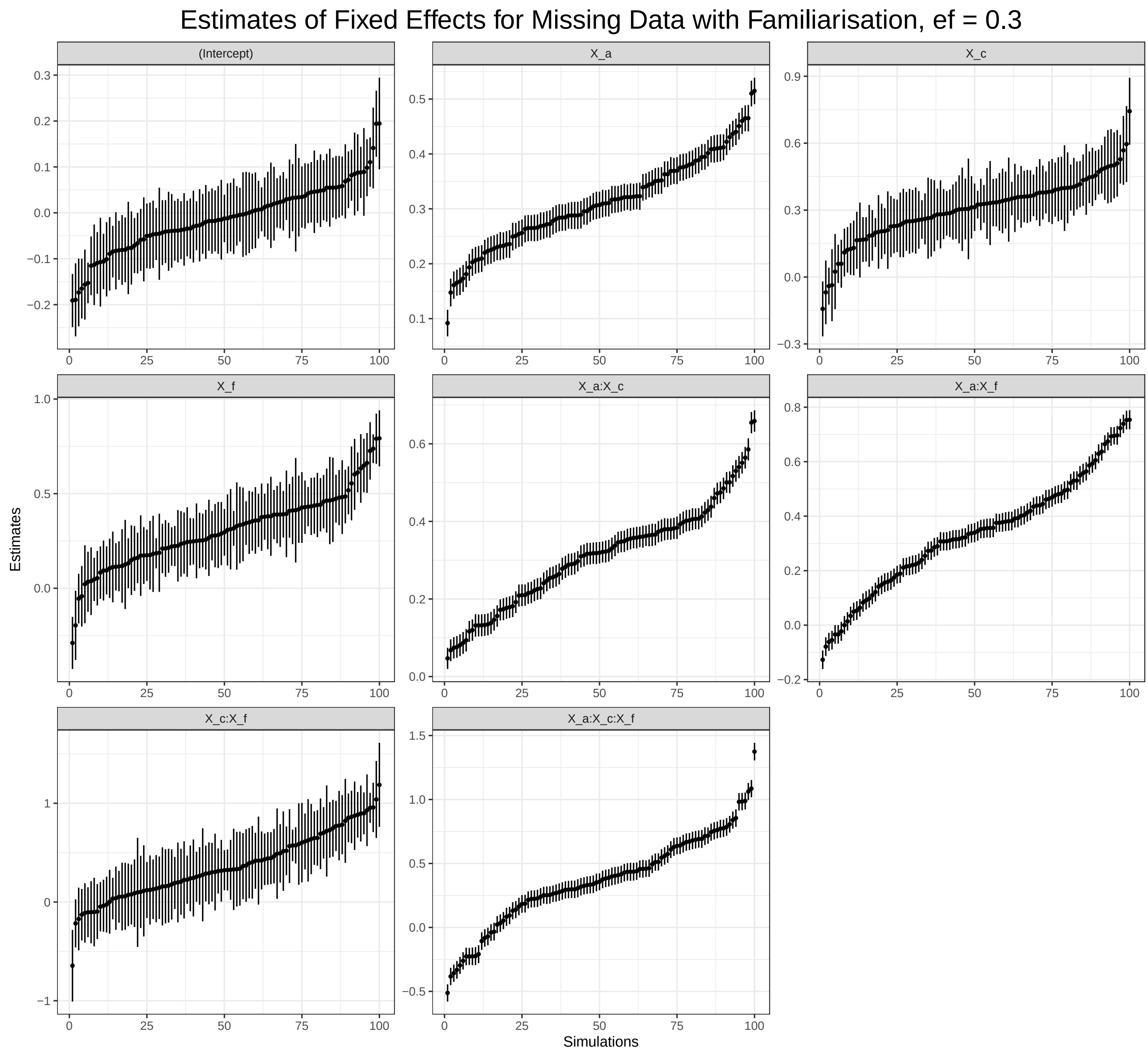
10.3.1 Simulation of models

```
# Number of simulations:
reps <- 100

# Simulation function:
run_sims <- function(filename_full, ef) {

  dat_sim <- my_sim_data(beta_c  =  ef,
                         beta_f = ef,
                         beta_a = ef,

                         beta_ca = ef,
                         beta_af = ef,
                         beta_cf = ef,
```

```
                         beta_cfa = ef)

  missing_samples_fam <- dat_sim %>%
    mutate(nas = rbinom(n(), 1, 0.95 - ifelse(X_f > -0.5, (X_f+0.5)*0.6, 0))) %>%
    mutate(DV = ifelse(nas == 1, DV, NA))

  mod_sim <- lmer(DV ~ 1 + X_a * X_c * X_f +
                    (1 | subj_id) +
                    (1 | lab_id) +
                    (1 | item_id),
                  data=missing_samples_fam)

  sim_results <- broom.mixed::tidy(mod_sim)

  # append the results to a file
  append <- file.exists(filename_full)
  write_csv(sim_results, filename_full, append = append)

  # return the tidy table
  sim_results
}

filename_full_0.3_missing_fam = 'sims/run_sims_0.3_fam_missing.csv'
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_0.3_missing_fam, ef = 0.3))
end_time <- Sys.time()
end_time - start_time
```

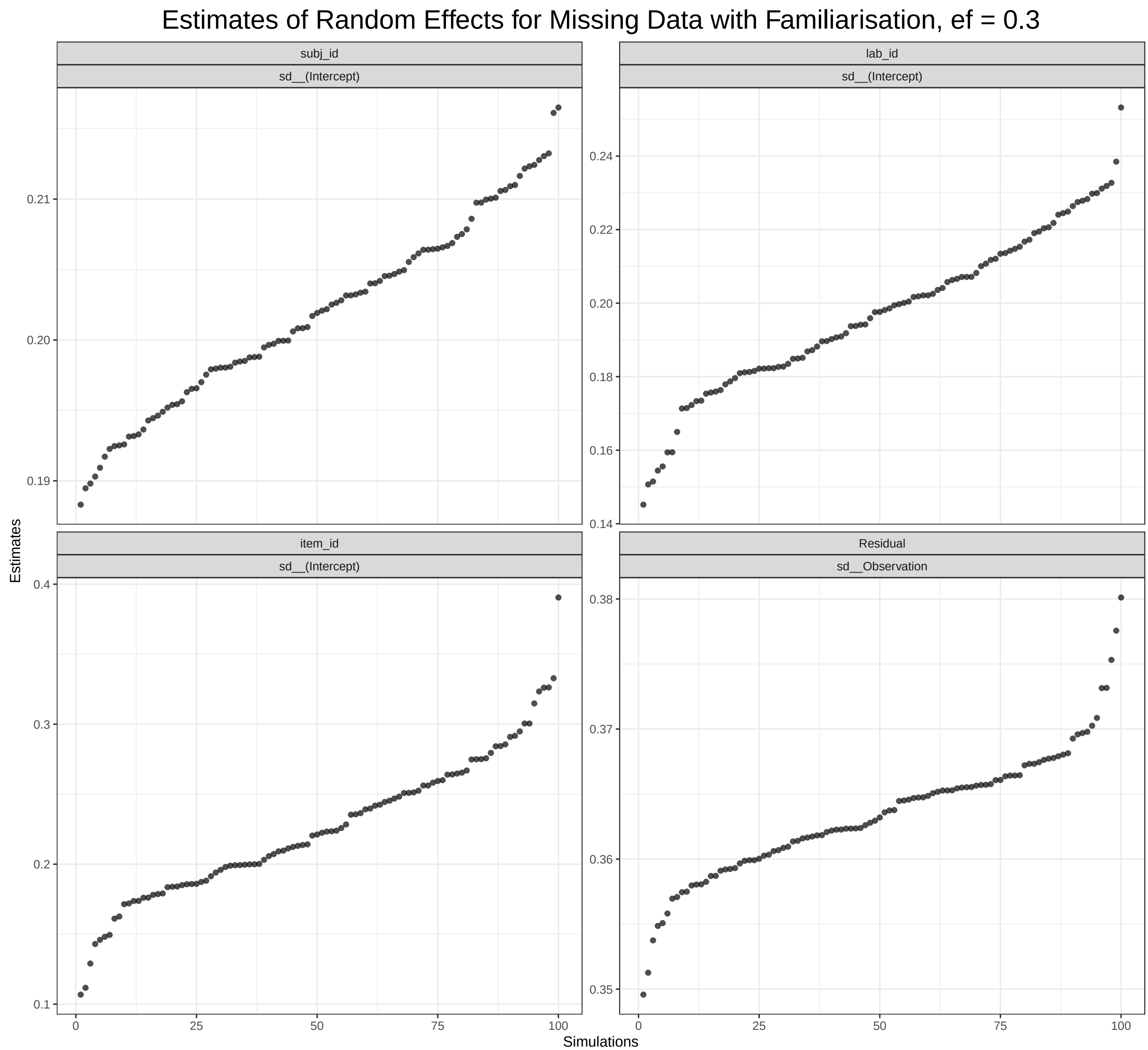
10.3.2 Visualise Estimates for Fixed Effects:

```
# read saved simulation data
sims_50_missing_fam_0.3 <- read_csv(filename_full_0.3_missing_fam, col_types = cols(
  # makes sure plots display in this order
  group = col_factor(ordered = TRUE),
  term = col_factor(ordered = TRUE)
))

fixed_missing_fam_plot <- sims_50_missing_fam_0.3 %>%
  filter(effect == "fixed") %>%
  ungroup() %>%
  arrange(term, estimate) %>%
  mutate(row = rep(seq(1:reps), 8)) %>%
  ggplot(aes(x = row, y = estimate, ymin = estimate-std.error, ymax = estimate+std.error)) +
  facet_wrap(~term, scales = "free") +
  geom_pointrange(fatten = 1/2) +
  ylab("Estimates") +
  xlab("Simulations") +
  ggtitle('Estimates of Fixed Effects for Missing Data with Familiarisation, ef = 0.3') +
  theme_bw()

fixed_missing_fam_plot <- fixed_missing_fam_plot + theme(plot.title = element_text(hjust = 0.5, size=20))
fixed_missing_fam_plot
```



Visualise Estimates for Random Effects:

```
ran_missing_fam_plot <- sims_50_missing_fam_0.3 %>%
    filter(effect == "ran_pars") %>%
    ungroup() %>%
    arrange(group, term, estimate) %>%
    mutate(row = rep(seq(1:reps), 4)) %>%
    ggplot(aes(x = row, y = estimate)) + geom_point(alpha = 0.7) +
    facet_wrap(~group + term, scales = "free_y") + theme_bw() +
    ylab("Estimates") + xlab("Simulations") + ggtitle("Estimates of Random Effects for Missing Data with Familiar
    theme_bw()
ran_missing_fam_plot <- ran_missing_fam_plot + theme(plot.title = element_text(hjust = 0.5,
    size = 20))
ran_missing_fam_plot
```



11 Summary Statistics for Power Calculation with Data missing not completely at random

Table 9: Power for Simulations with Non-Random Missing Data and Varying Intercepts

term	missing with age	missing with trial	missing with fam
(Intercept)	0.07	0.03	0.06
X__a	1.00	1.00	1.00
X__c	0.50	0.48	0.55
X__f	0.35	0.37	0.35
X__a:X__c	0.94	0.97	0.99
X__a:X__f	0.87	0.86	0.90
X__c:X__f	0.17	0.16	0.20
X__a:X__c:X__f	0.81	0.81	0.89

Table 10: Bias for Simulations with Non-Random Missing Data and Varying Intercepts

term	age bias, ef = 0.3	trial bias, ef = 0.3	fam bias, ef = 0.3
(Intercept)	-0.001	-0.007	0.012
X__a	0.000	0.014	-0.008
X__c	-0.019	0.014	-0.019
X__f	0.007	-0.016	-0.002
X__a:X__c	-0.009	-0.047	-0.020
X__a:X__f	-0.002	0.011	-0.042
X__c:X__f	-0.065	-0.062	-0.023
X__a:X__c:X__f	0.039	-0.052	-0.066

12 Logistic Regression Models

```
m_missing_age <- glmer(nas ~ X_f + X_a + X_c + trial_num + (1 |
    subj_id) + (1 | lab_id) + (1 | item_id), data = missing_samples_age,
    family = binomial)

summary(m_missing_age)

m_missing_fam <- glmer(nas ~ X_f + X_a + X_c + trial_num + (1 |
    subj_id) + (1 | lab_id) + (1 | item_id), data = missing_samples_fam,
    family = binomial)

summary(m_missing_fam)

m_missing_trial <- glmer(nas ~ X_f + X_a + X_c + trial_num +
    (1 | subj_id) + (1 | lab_id) + (1 | item_id), data = missing_samples_trial,
    family = binomial)

summary(m_missing_trial)
```