# 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

```
my_sim_data <- function(
    n_subj      = 1280,   # number of subjects
    n_simple  =  12,   # number of complex stimuli
    n_complex =  12,   # number of complex stimuli
    n_small_fam = 8,   #small familiarization time
    n_medium_fam = 8,  #medium familiarization time
    n_high_fam = 8,    #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
```

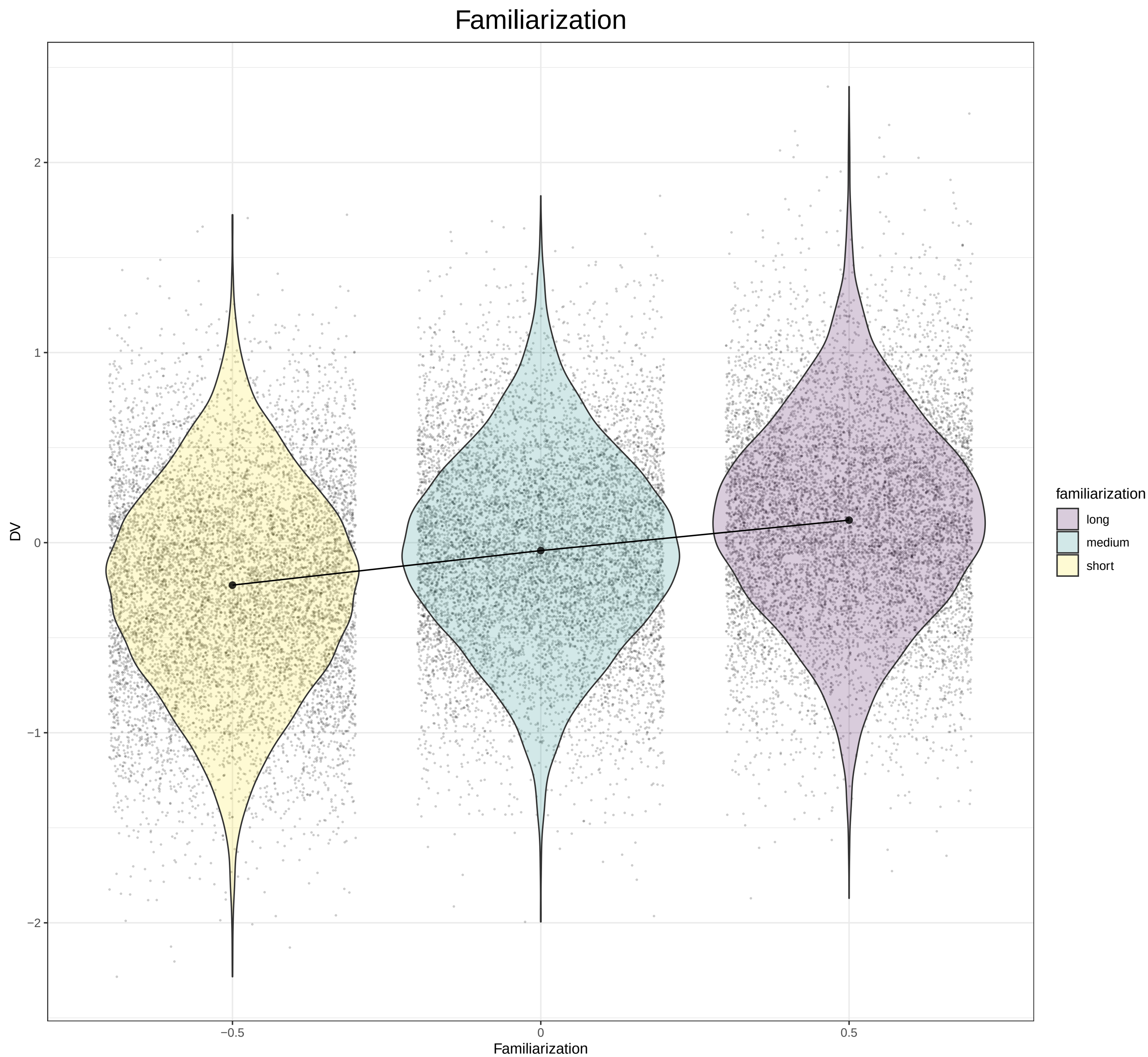
```r
  item_rho = 0.2, # correlations between by-item random effects


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

# simulate a sample of items
items <- data.frame(
  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")))

# 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)
```

```r
  )


  # cross subject and item IDs
  temp <- crossing(subjects, items)  %>%
    left_join(lab_subj_dict, by = "subj_id") %>%
    left_join(labs, by = "lab_id")

  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:

```r
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
```



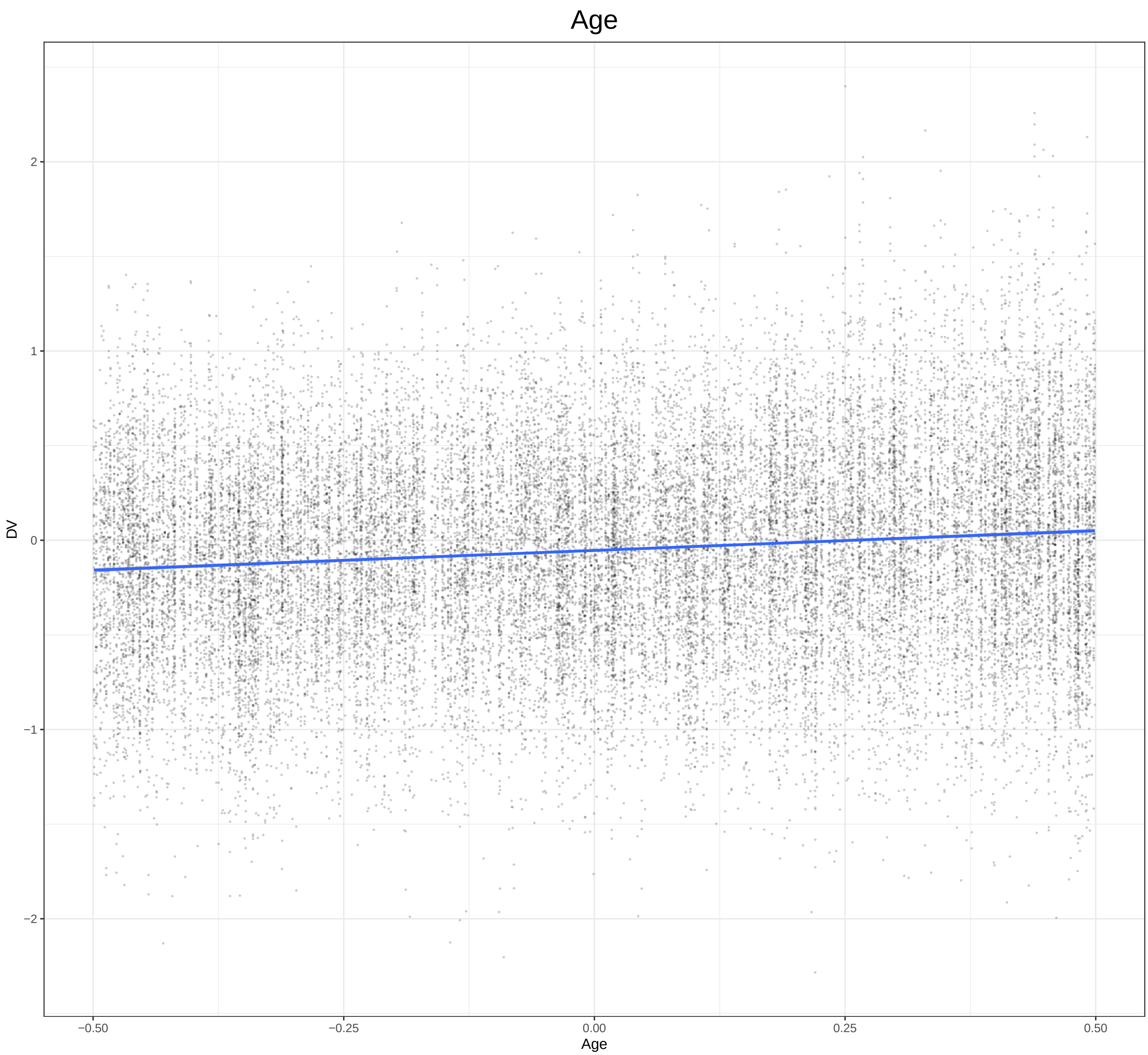

## 2.2 Complexity:

```
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
```

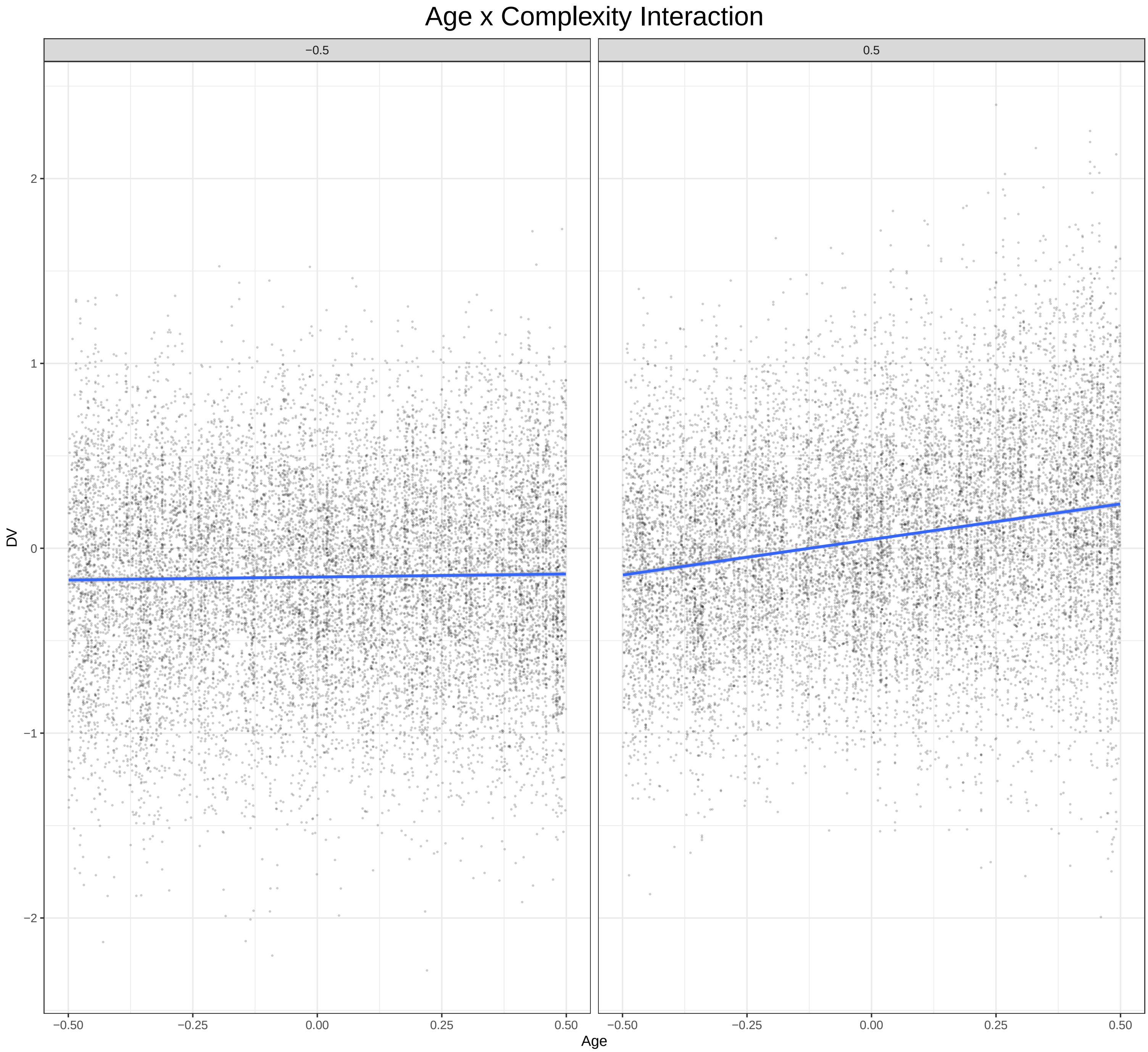



## 2.4 Age*Familiarization:

```
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
```



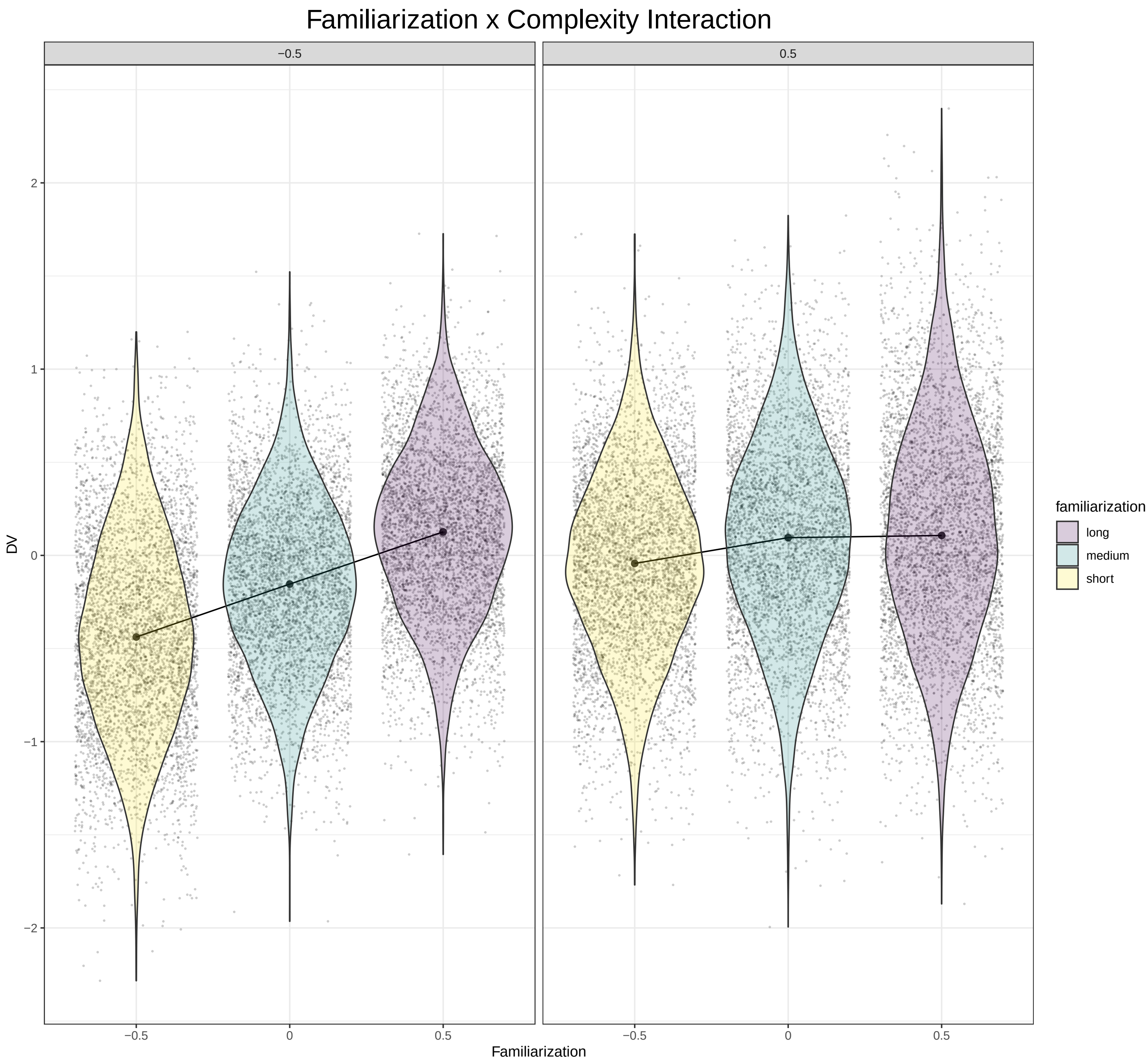

## 2.6 Familiarization*Complexity:

```
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))
```

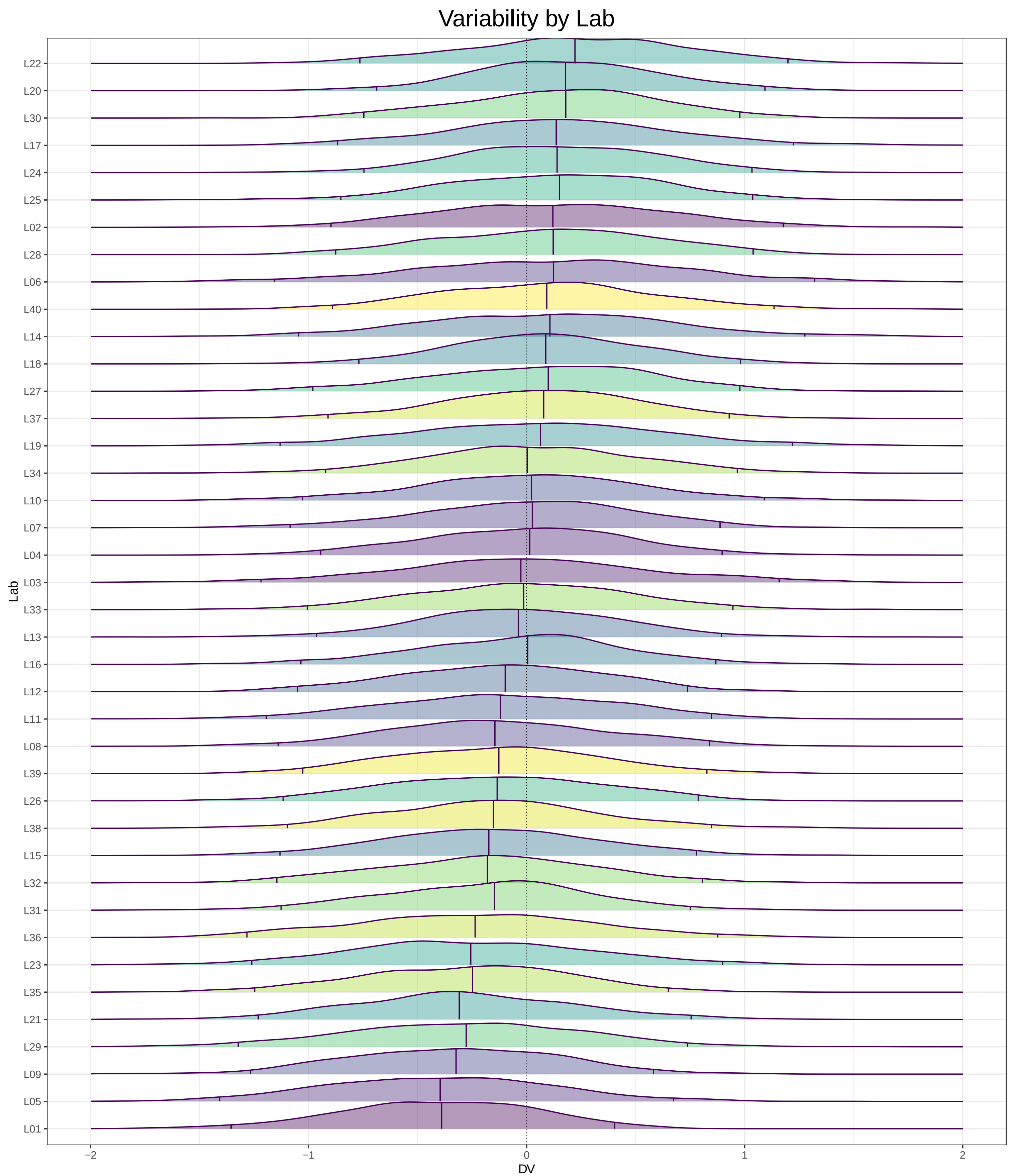
```
## `summarise()` has grouped output by 'X_f'. You can override using the `.groups`
## argument.
```

```
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",
```

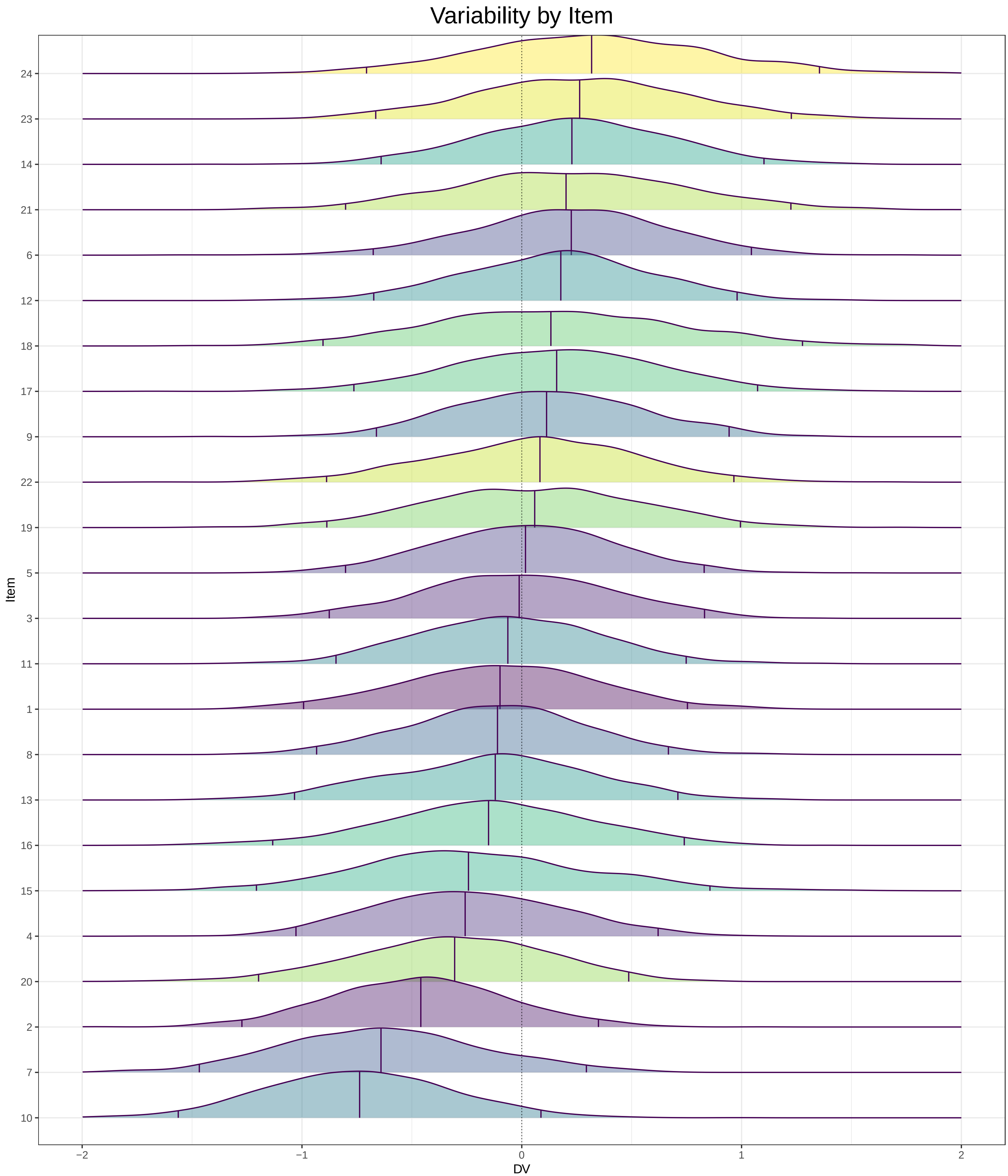
```
    alpha = 0.2, size = 0.2) + geom_point(aes(y = med_DV, x = as.factor(X_f)),
    alpha = 0.8, size = 2, data = dat_f_c_interaction) + geom_line(aes(y = med_DV,
    x = as.factor(X_f), group = 1), data = dat_f_c_interaction) +
    geom_violin(aes(y = DV, x = X_f, fill = familiarization),
        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 <- 350

# 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
```

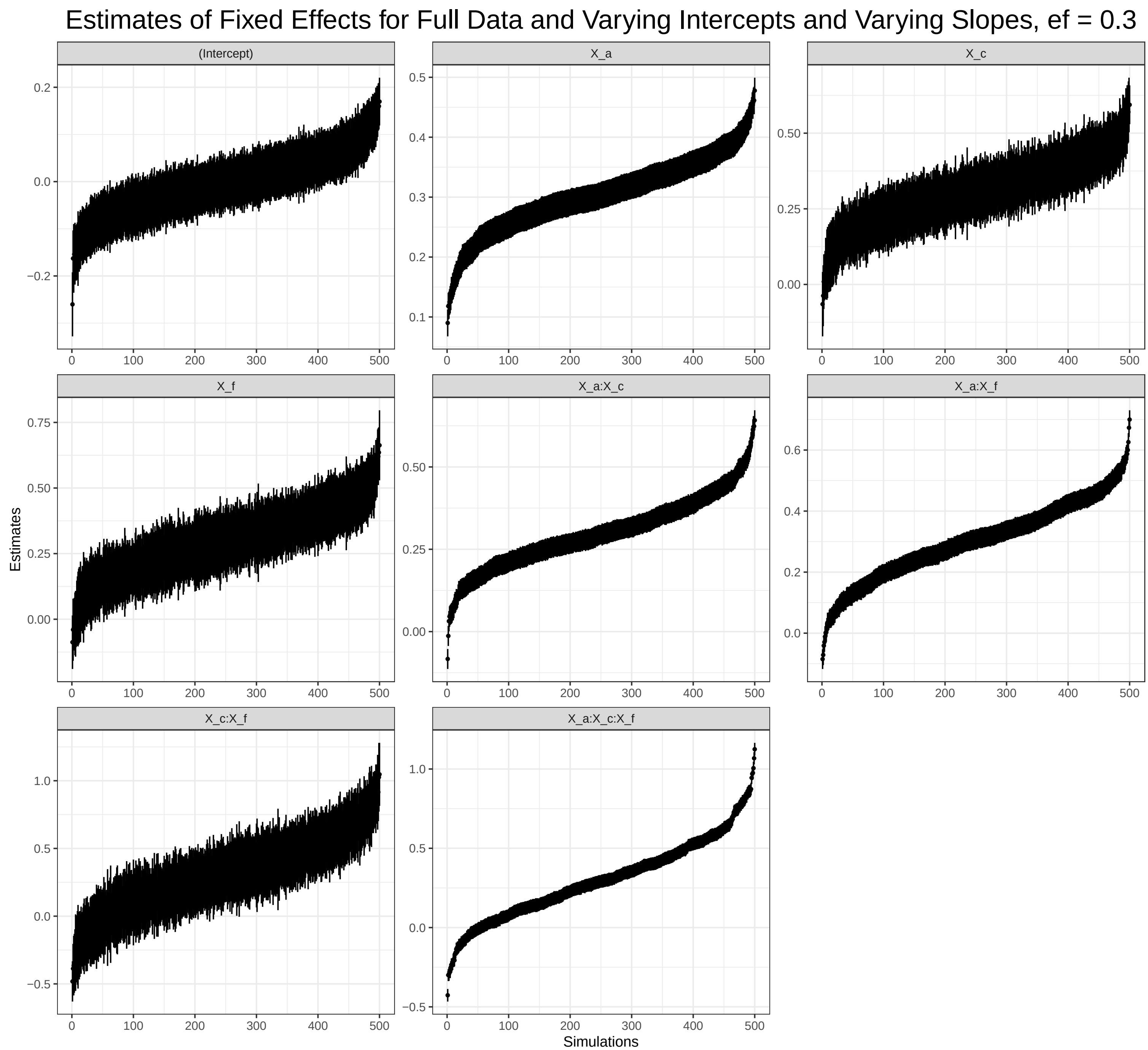
## 3.3 Effect Size = 0.3

```
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
```

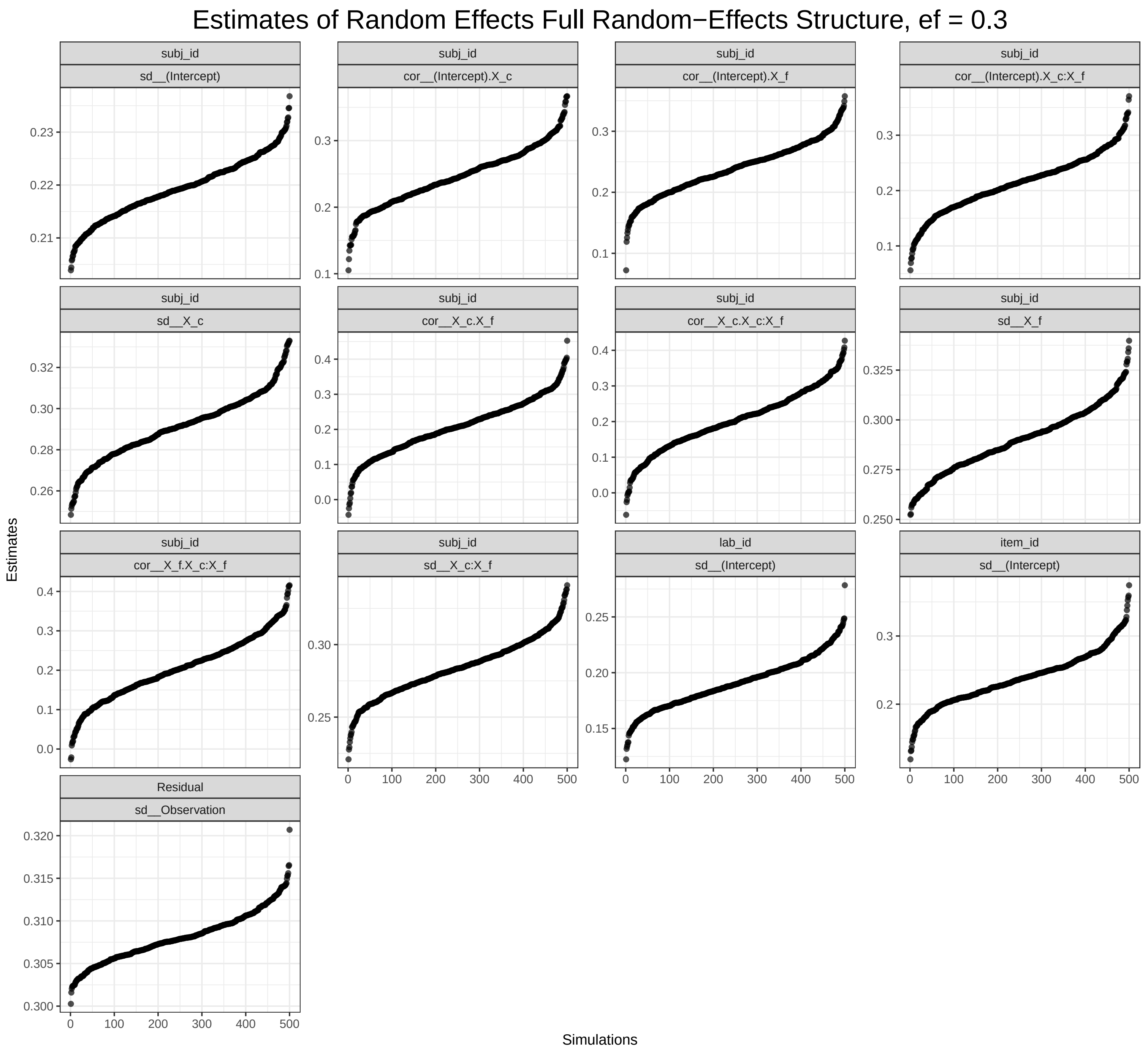
#### 3.3.1 Visualise Estimates for Fixed Effects:

```
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

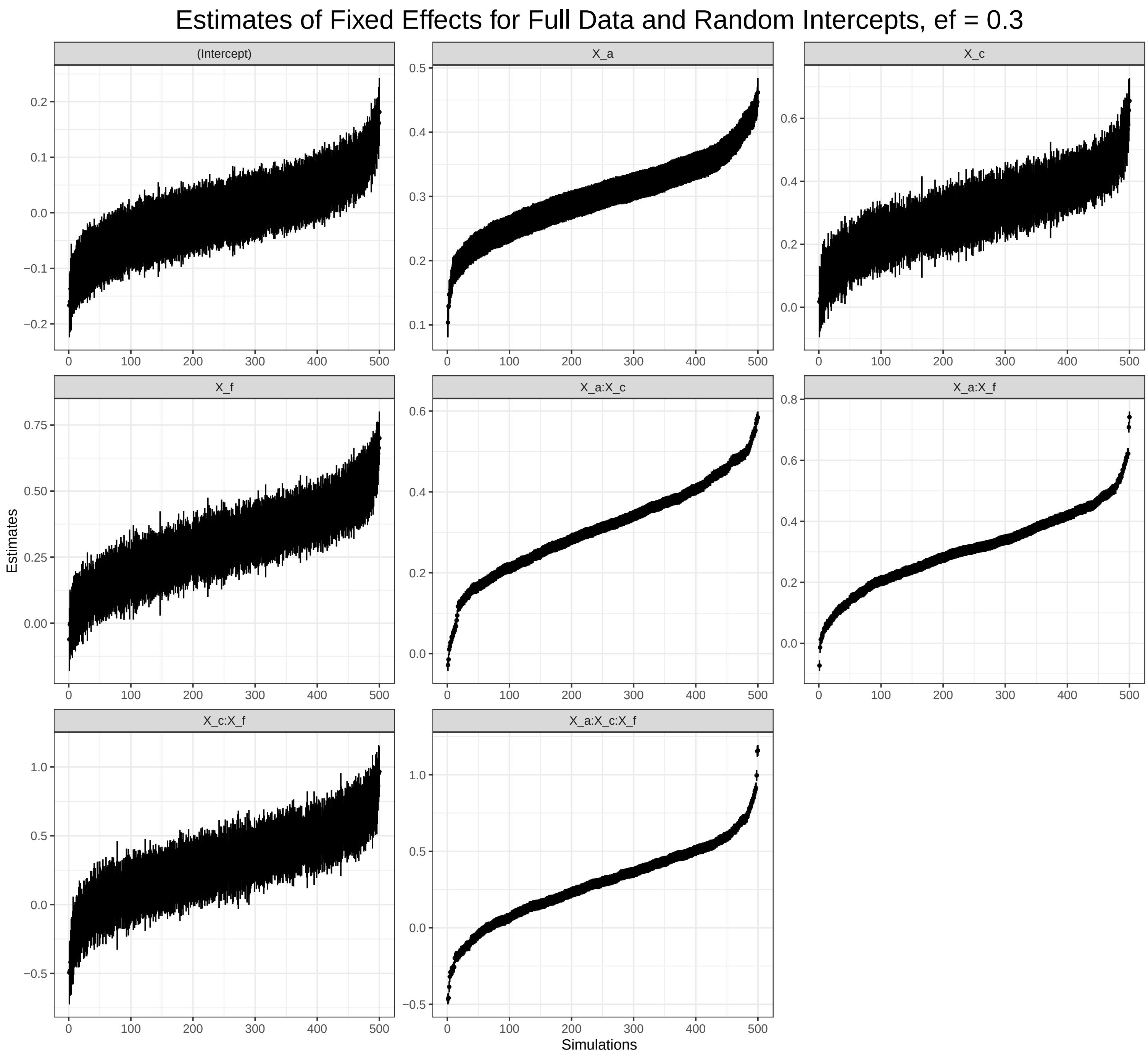
```
filename_full_int_0.4 = "run_sims_full_int_0.4.csv"
start_time <- Sys.time()
sims <- purrr::map_df(1:reps, ~run_sims(filename_full = filename_full_int_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

## 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
```

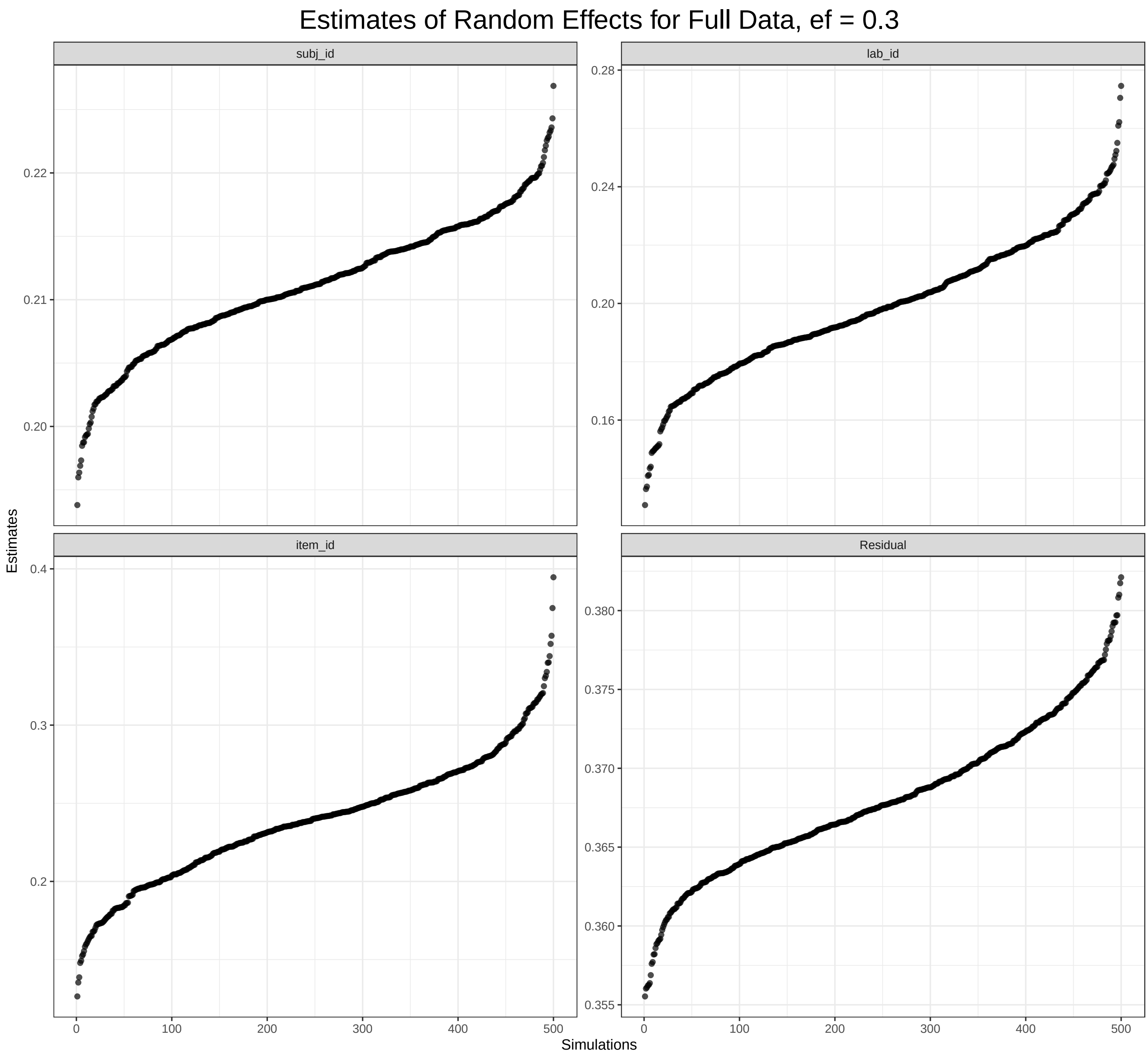
#### 4.3.1 Visualise Estimates for Fixed Effects:

```
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

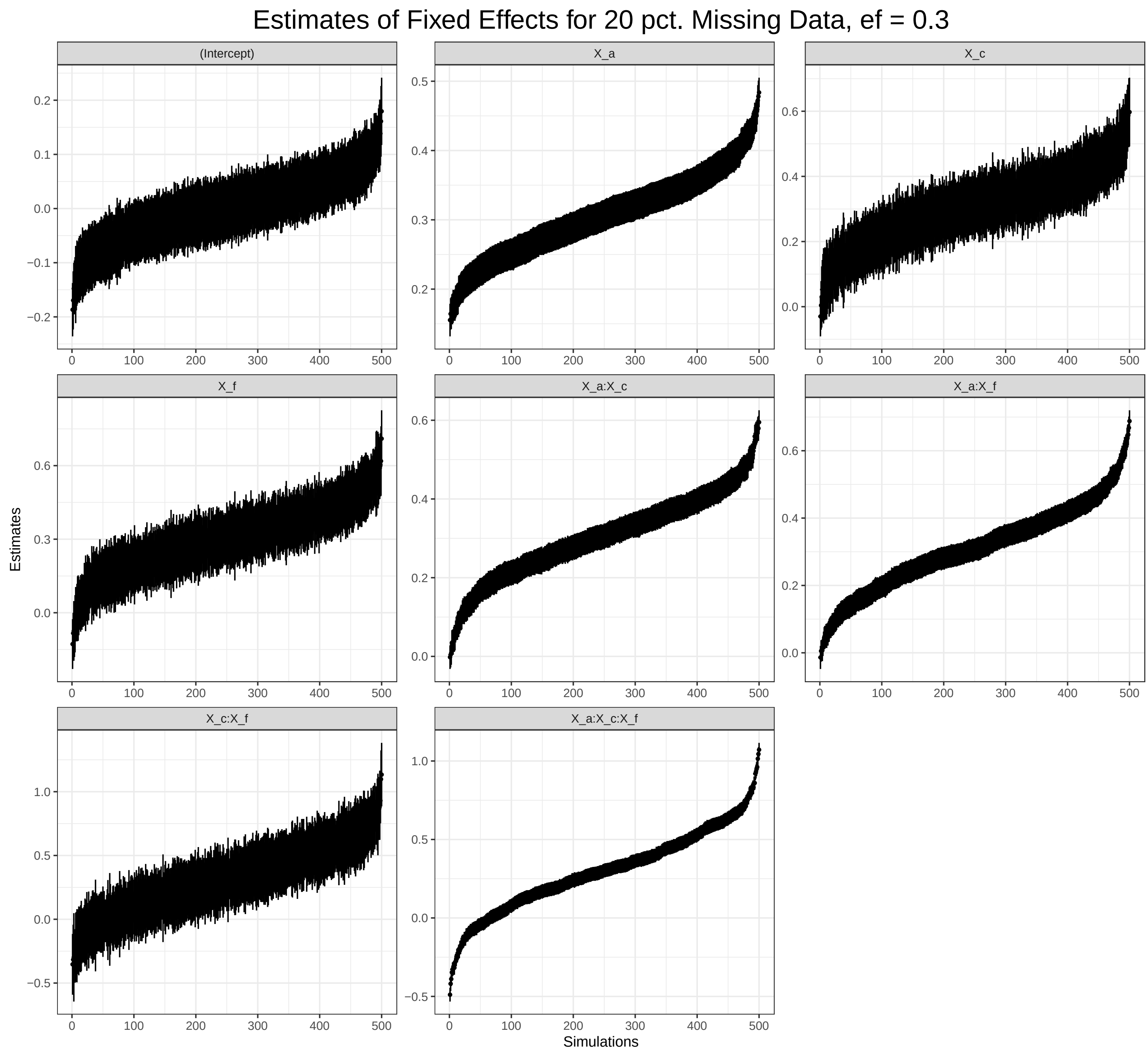
```
filename_20_missing_0.4 = "run_sims_20_missing_0.4.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_20_missing_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

### 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
```

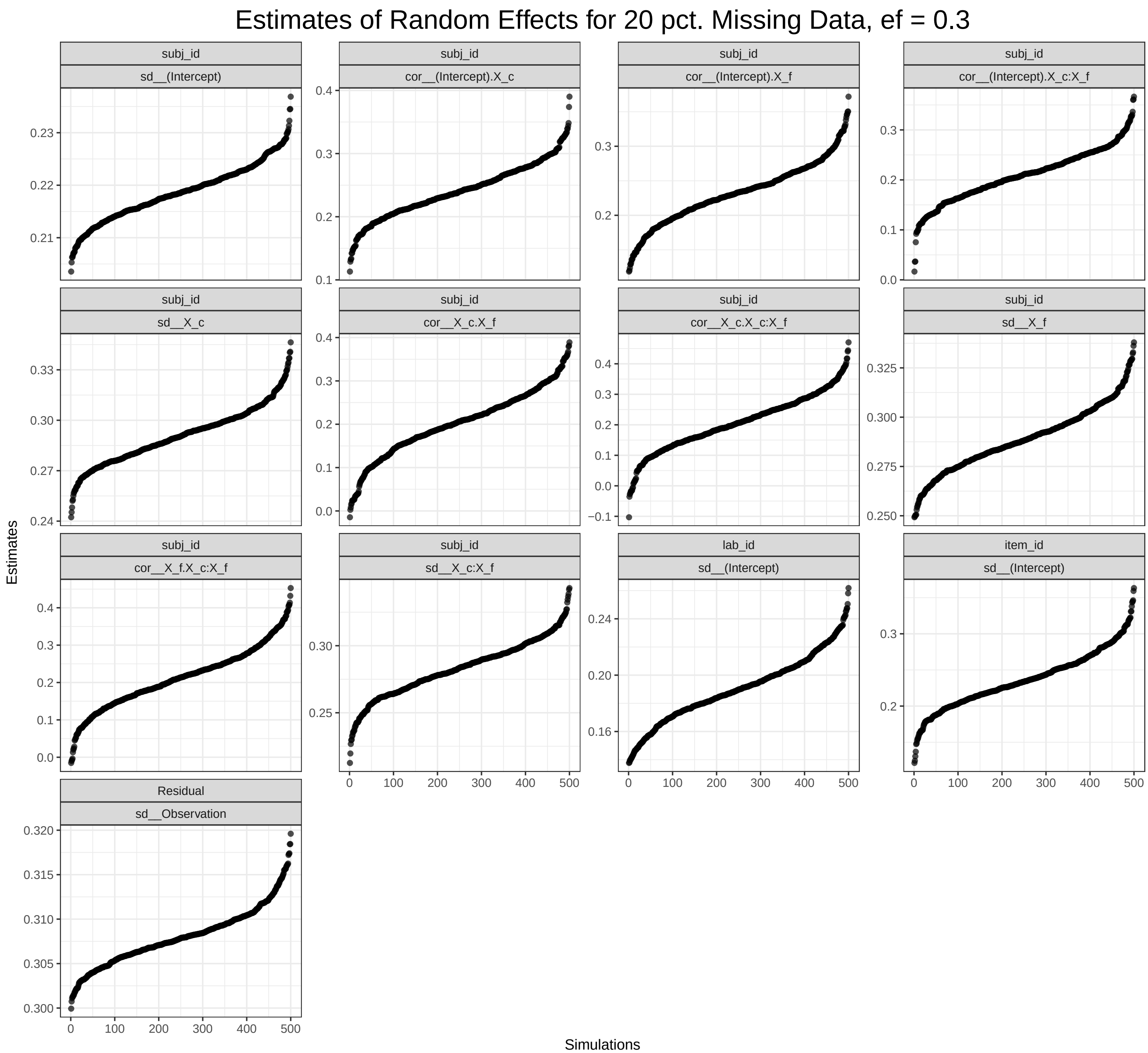
#### 5.3.1 Visualise Estimates for Fixed Effects:

```
# 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

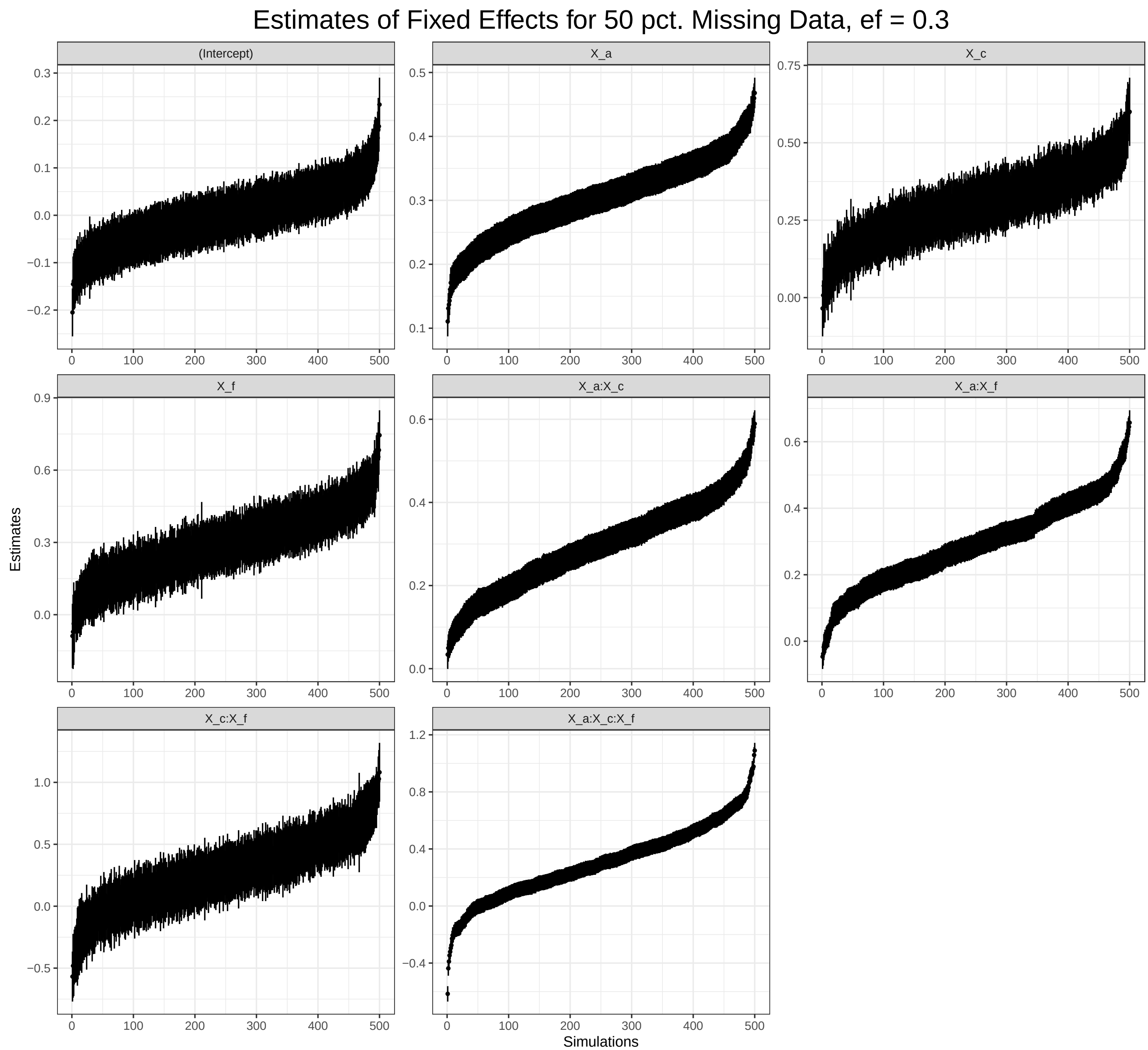
```
filename_50_missing_0.4 = "run_sims_50_missing_0.4.csv"
start_time <- Sys.time()
sims_missing <- purrr::map_df(1:reps, ~run_sims_missing(filename_missing = filename_50_missing_0.4,
    ef = 0.4))
end_time <- Sys.time()
end_time - start_time
```

## 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
```

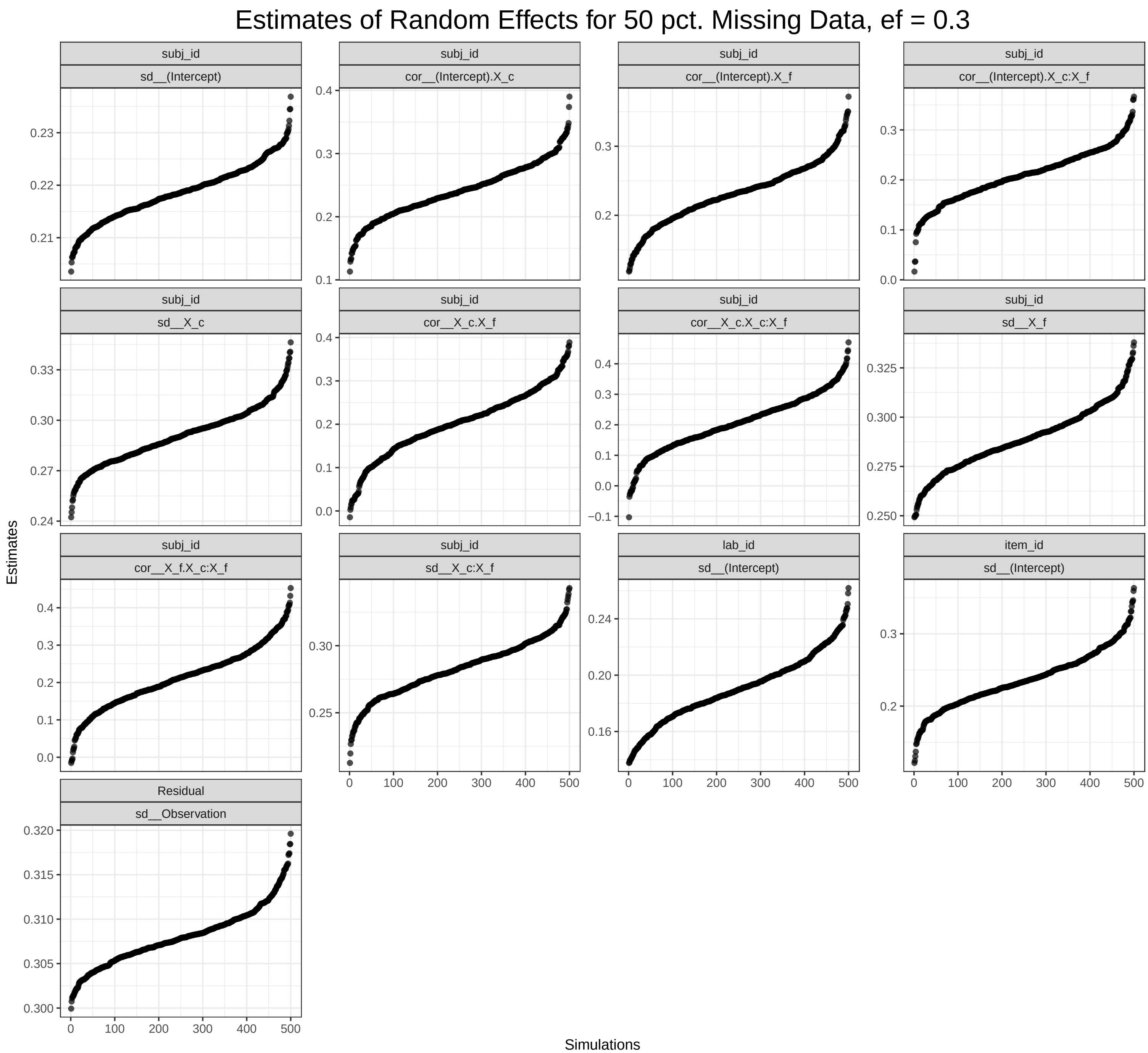
### 6.3.1 Visualise Estimates for Fixed Effects:

```
# 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

```r
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

```r
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) | 0.050 | 0.056 | 0.070 | 0.056 | 0.026 |
| X_a | 0.844 | 0.990 | 1.000 | 1.000 | 1.000 |
| X_c | 0.170 | 0.474 | 0.838 | 0.966 | 0.992 |
| X_f | 0.118 | 0.360 | 0.634 | 0.864 | 0.968 |
| X_a:X_c | 0.690 | 0.894 | 0.986 | 1.000 | 1.000 |
| X_a:X_f | 0.748 | 0.864 | 0.966 | 0.994 | 1.000 |
| X_c:X_f | 0.072 | 0.124 | 0.244 | 0.350 | 0.486 |
| X_a:X_c:X_f | 0.770 | 0.808 | 0.856 | 0.938 | 0.956 |

## 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 | 0.056 | 0.050 | 0.080 | 0.058 |
| X_a | 0.822 | 0.994 | 1.000 | 1.000 | 1.000 |
| X_c | 0.162 | 0.482 | 0.810 | 0.998 | 0.966 |
| X_f | 0.136 | 0.352 | 0.644 | 0.970 | 0.884 |
| X_a:X_c | 0.868 | 0.958 | 0.990 | 1.000 | 0.998 |
| X_a:X_f | 0.848 | 0.932 | 0.990 | 1.000 | 0.998 |
| X_c:X_f | 0.092 | 0.136 | 0.220 | 0.564 | 0.348 |
| X_a:X_c:X_f | 0.798 | 0.832 | 0.882 | 0.964 | 0.934 |

## 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) | 0.040 | 0.054 | 0.042 | 0.064 | 0.052 |
| X_a | 0.832 | 0.996 | 1.000 | 1.000 | 1.000 |
| X_c | 0.162 | 0.476 | 0.806 | 0.950 | 0.998 |
| X_f | 0.122 | 0.348 | 0.658 | 0.842 | 0.972 |
| X_a:X_c | 0.686 | 0.928 | 0.982 | 1.000 | 1.000 |
| X_a:X_f | 0.674 | 0.872 | 0.974 | 0.996 | 1.000 |
| X_c:X_f | 0.062 | 0.132 | 0.266 | 0.376 | 0.498 |
| X_a:X_c:X_f | 0.762 | 0.800 | 0.862 | 0.926 | 0.950 |

### 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) | 0.036 | 0.058 | 0.054 | 0.044 | 0.046 |
| X__a | 0.822 | 0.986 | 1.000 | 1.000 | 1.000 |
| X__c | 0.168 | 0.540 | 0.840 | 0.970 | 0.996 |
| X__f | 0.120 | 0.366 | 0.658 | 0.836 | 0.964 |
| X__a:X__c | 0.666 | 0.902 | 0.992 | 0.998 | 1.000 |
| X__a:X__f | 0.680 | 0.826 | 0.966 | 0.992 | 1.000 |
| X__c:X__f | 0.066 | 0.174 | 0.232 | 0.372 | 0.460 |
| X__a:X__c:X__f | 0.690 | 0.754 | 0.840 | 0.896 | 0.942 |