HandlingTime_z ~ stim_cat * Reward_z + (1 | Participant_ID)

Three reward normalisations are tested as `Reward_z`:

- `Reward_pct_max_theo_z` — reward as % of theoretical maximum, globally z-scored
- `Reward_rel_baseline_z` — reward relative to task baseline, globally z-scored
- `Reward_minmax_emp_z` — empirical min-max normalised reward, globally z-scored

```r
library(tidyverse)
library(brms)
library(bayesplot)
library(loo)
library(patchwork)

options(mc.cores = parallel::detectCores())
rstan::rstan_options(auto_write = TRUE)

EPSILON <- 1e-6
```

```r
df_post <- read.csv("/Users/debarpita/Desktop/arjun/trial_wise_dataset_post.csv")
df_pre  <- read.csv("/Users/debarpita/Desktop/arjun/trial_wise_dataset_pre.csv")

df <- bind_rows(df_post, df_pre)
```

```r
# Game parameters
df <- df %>%
  mutate(
    rew_base = ifelse(stim_cat == "pre", 91, 181),
    rew_hi   = ifelse(stim_cat == "pre",  9,  19),
    rew_dec  = ifelse(stim_cat == "pre", 10,  20)
  )

# Trial index within each patch (needed for R_min_theo)
df <- df %>%
  group_by(Participant_ID, stim_cat, env, patch_id) %>%
  mutate(trial_in_patch = row_number()) %>%
  ungroup()

# Theoretical bounds
df <- df %>%
  mutate(
    R_max_theo = rew_base + rew_hi,
    R_min_theo = pmax(rew_base - rew_dec * trial_in_patch, 0)
  )

# Empirical bounds within Participant × stim_cat × env
df <- df %>%
  group_by(Participant_ID, stim_cat, env) %>%
  mutate(
    R_min_empirical = min(Reward, na.rm = TRUE),
    R_max_empirical = max(Reward, na.rm = TRUE)
  ) %>%
```

```r
  ungroup()

# Normalised reward variants
df <- df %>%
  group_by(Participant_ID, stim_cat, env) %>%
  mutate(
    Reward_pct_max_theo = Reward / R_max_theo,
    Reward_rel_baseline = (Reward - rew_base) / (rew_hi + 1e-6),
    Reward_minmax_emp   = (Reward - R_min_empirical) /
                          (R_max_empirical - R_min_empirical + 1e-6)
  ) %>%
  ungroup()

# Global z-scores
df <- df %>%
  mutate(
    Reward_pct_max_theo_z = scale(Reward_pct_max_theo)[, 1],
    Reward_rel_baseline_z = scale(Reward_rel_baseline)[, 1],
    Reward_minmax_emp_z   = scale(Reward_minmax_emp)[, 1]
  )
```

```r
df <- df %>%
  mutate(HandlingTime_eps = HandlingTime + EPSILON)

# Step 2: drop zero-reward trials
df_ht <- df %>%
  filter(Reward != 0, !is.na(HandlingTime), !is.na(Reward))

# Step 3: IQR-based outlier removal within each stim_cat
iqr_bounds <- df_ht %>%
  group_by(stim_cat) %>%
  summarise(
    Q1  = quantile(HandlingTime_eps, 0.25, na.rm = TRUE),
    Q3  = quantile(HandlingTime_eps, 0.75, na.rm = TRUE),
    IQR = IQR(HandlingTime_eps, na.rm = TRUE),
    lower = Q1 - 1.5 * IQR,
    upper = Q3 + 1.5 * IQR,
    .groups = "drop"
  )

cat("IQR bounds per stim_cat:\n")
```

```
## IQR bounds per stim_cat:
```

```r
print(iqr_bounds)
```

```
## # A tibble: 2 x 6
##   stim_cat    Q1    Q3   IQR  lower upper
##   <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1 post      2.39  3.10 0.713  1.32   4.17
## 2 pre       2.54  4.34 1.80  -0.164  7.04
```

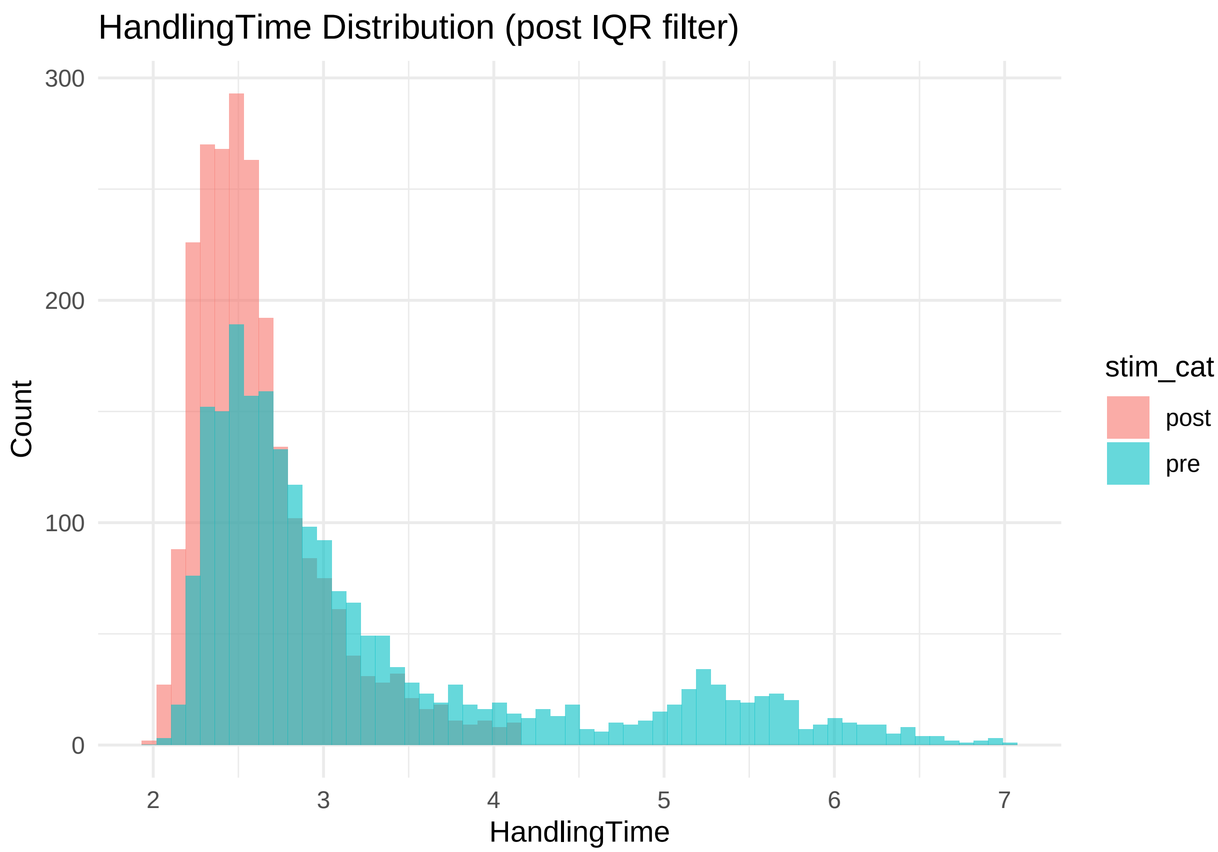
```r
df_ht <- df_ht %>%
  left_join(iqr_bounds %>% select(stim_cat, lower, upper), by = "stim_cat") %>%
  filter(HandlingTime_eps >= lower, HandlingTime_eps <= upper) %>%
  select(-lower, -upper)

cat(sprintf("\nObservations after IQR filter: %d\n", nrow(df_ht)))
```

```
##
## Observations after IQR filter: 4505
```

```r
# Step 4: z-score HandlingTime
df_ht <- df_ht %>%
  mutate(
    HandlingTime_z        = scale(HandlingTime_eps)[, 1],
    trait_anxiety_score_z = scale(trait_anxiety_score)[, 1]
  )
```

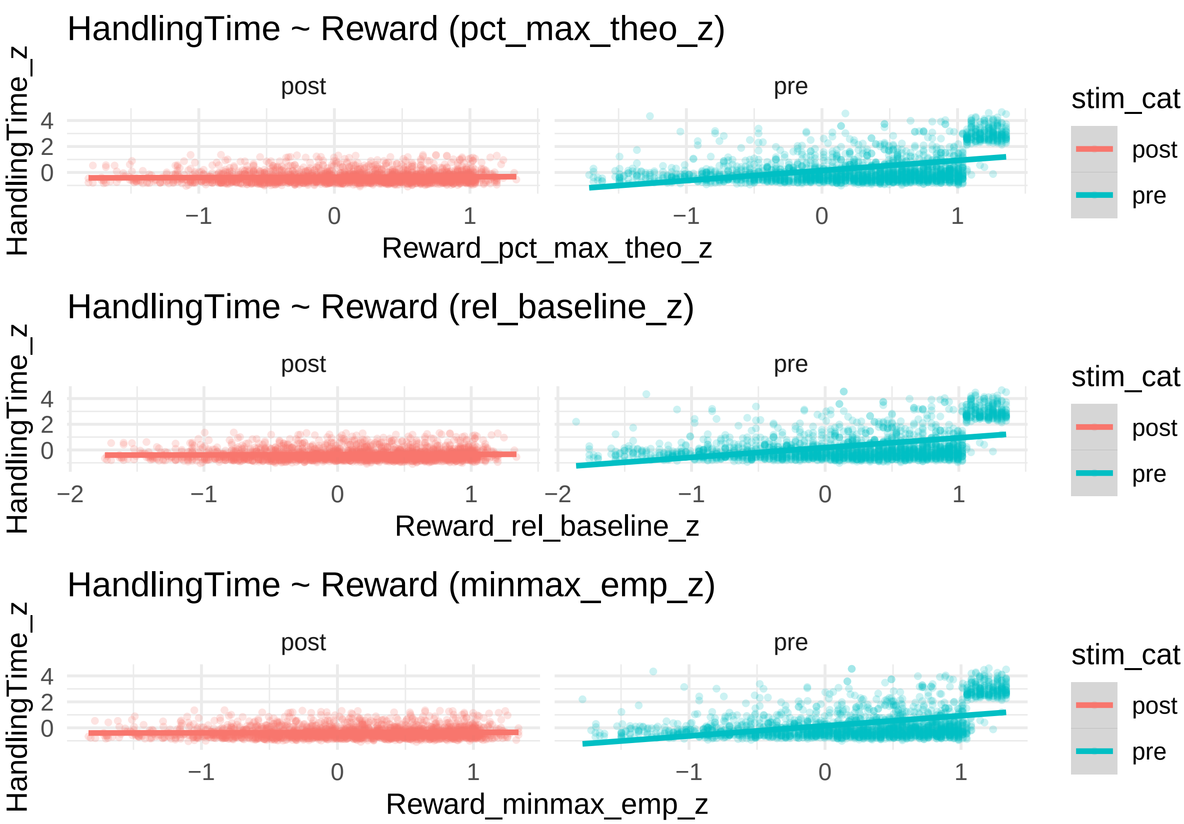
```r
p1 <- ggplot(df_ht, aes(x = HandlingTime_eps, fill = stim_cat)) +
  geom_histogram(bins = 60, alpha = 0.6, position = "identity") +
  theme_minimal() +
  labs(title = "HandlingTime Distribution (post IQR filter)",
       x = "HandlingTime", y = "Count")

p2 <- ggplot(df_ht %>% sample_n(min(4000, nrow(df_ht))),
             aes(x = Reward_pct_max_theo_z, y = HandlingTime_z,
                 colour = stim_cat)) +
  geom_point(alpha = 0.2, size = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  facet_wrap(~stim_cat) +
  theme_minimal() +
  labs(title = "HandlingTime ~ Reward (pct_max_theo_z)",
       x = "Reward_pct_max_theo_z", y = "HandlingTime_z")

p3 <- ggplot(df_ht %>% sample_n(min(4000, nrow(df_ht))),
             aes(x = Reward_rel_baseline_z, y = HandlingTime_z,
                 colour = stim_cat)) +
  geom_point(alpha = 0.2, size = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  facet_wrap(~stim_cat) +
  theme_minimal() +
  labs(title = "HandlingTime ~ Reward (rel_baseline_z)",
       x = "Reward_rel_baseline_z", y = "HandlingTime_z")

p4 <- ggplot(df_ht %>% sample_n(min(4000, nrow(df_ht))),
             aes(x = Reward_minmax_emp_z, y = HandlingTime_z,
                 colour = stim_cat)) +
  geom_point(alpha = 0.2, size = 0.7) +
  geom_smooth(method = "lm", se = TRUE) +
  facet_wrap(~stim_cat) +
  theme_minimal() +
  labs(title = "HandlingTime ~ Reward (minmax_emp_z)",
       x = "Reward_minmax_emp_z", y = "HandlingTime_z")
```

```
p1
```



HandlingTime Distribution (post IQR filter)
300
200
100
0
Count
2
3
4
5
6
7
HandlingTime
stim_cat
post
pre


```
p2 / p3 / p4
```



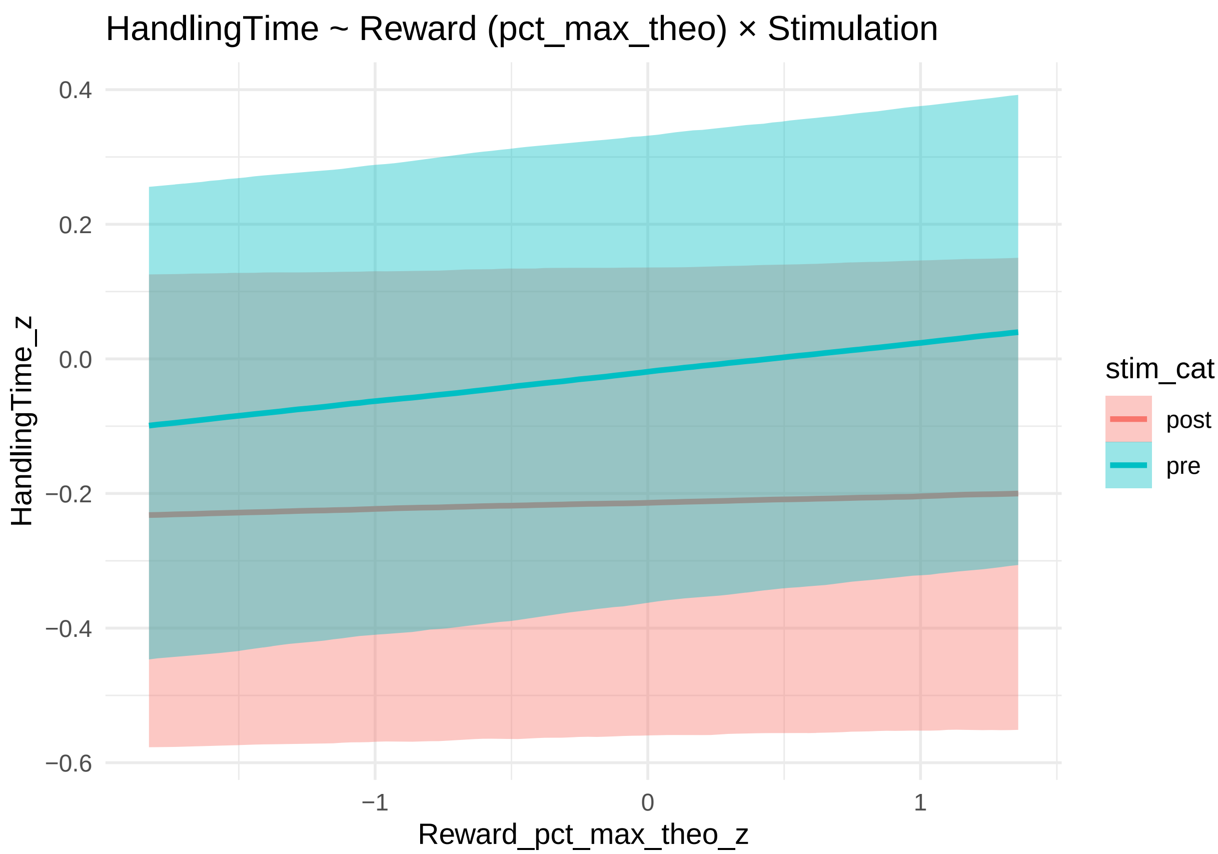
Model A — Reward: % of Theoretical Maximum

```
modelA <- brm(
  HandlingTime_z ~ stim_cat * Reward_pct_max_theo_z + (1 | Participant_ID),
  data    = df_ht,
  family  = student(),
  chains  = 4,
  iter    = 3000,
  warmup  = 1000,
  cores   = 4,
  seed    = 123,
  file    = "modelA_ht_pct_max_theo"
)

summary(modelA)
```

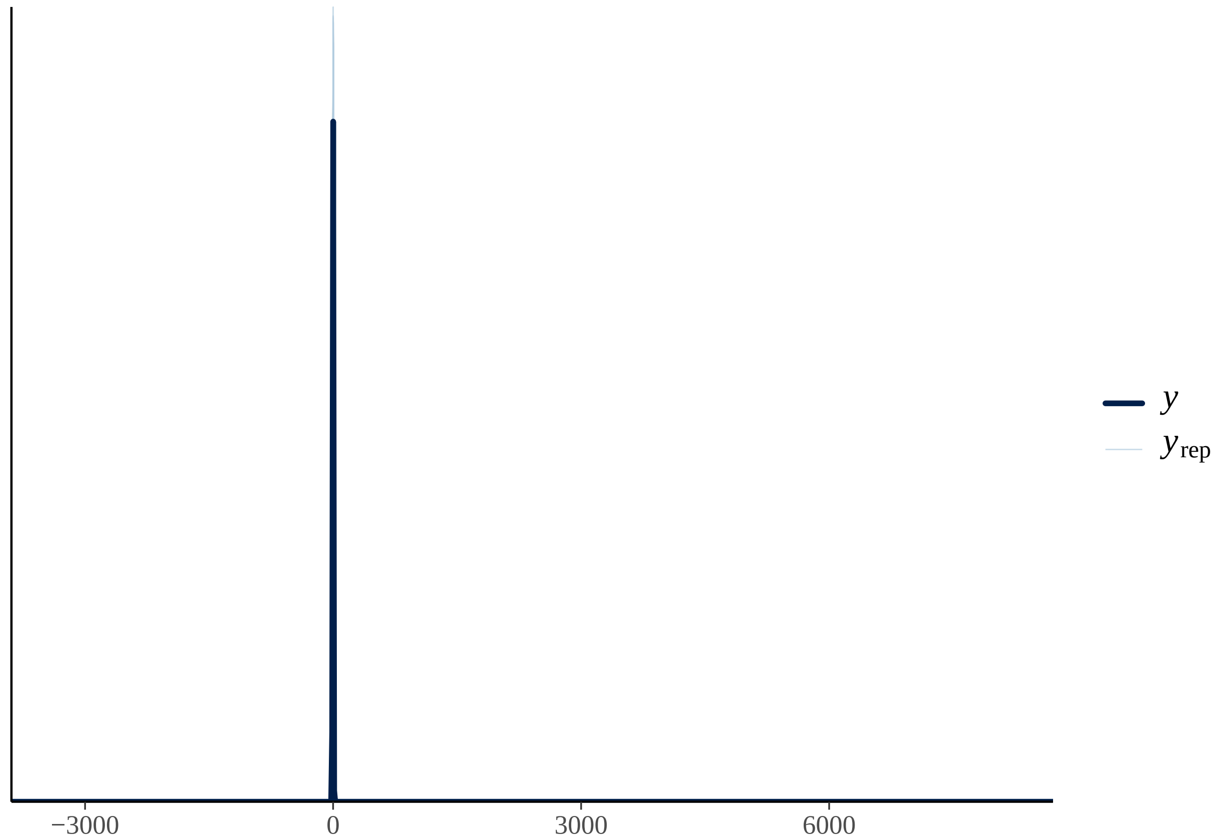
```
##  Family: student
##   Links: mu = identity
## Formula: HandlingTime_z ~ stim_cat * Reward_pct_max_theo_z + (1 | Participant_ID)
##    Data: df_ht (Number of observations: 4505)
##   Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup draws = 8000
##
## Multilevel Hyperparameters:
## ~Participant_ID (Number of levels: 21)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept)     0.82      0.14     0.61     1.15 1.00     1624     2620
##
## Regression Coefficients:
##                                    Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                             -0.21      0.18    -0.56     0.14 1.00
## stim_catpre                            0.20      0.01     0.18     0.22 1.00
## Reward_pct_max_theo_z                  0.01      0.01    -0.01     0.03 1.00
## stim_catpre:Reward_pct_max_theo_z      0.03      0.02     0.00     0.06 1.00
##                                    Bulk_ESS Tail_ESS
## Intercept                               887     1216
## stim_catpre                            5240     4617
## Reward_pct_max_theo_z                  4656     4741
## stim_catpre:Reward_pct_max_theo_z      4286     4068
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.24      0.01     0.22     0.25 1.00     4465     3940
## nu        1.29      0.04     1.21     1.36 1.00     4370     4000
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
ce <- conditional_effects(
  modelA,
  effects = "Reward_pct_max_theo_z:stim_cat"
)

p <- plot(ce, plot = FALSE)[[1]]

p +
  labs(
    title = "HandlingTime ~ Reward (pct_max_theo) × Stimulation",
    x = "Reward_pct_max_theo_z",
    y = "HandlingTime_z"
  ) +
  theme_minimal()
```




```
pp_check(modelA, ndraws = 100)
```



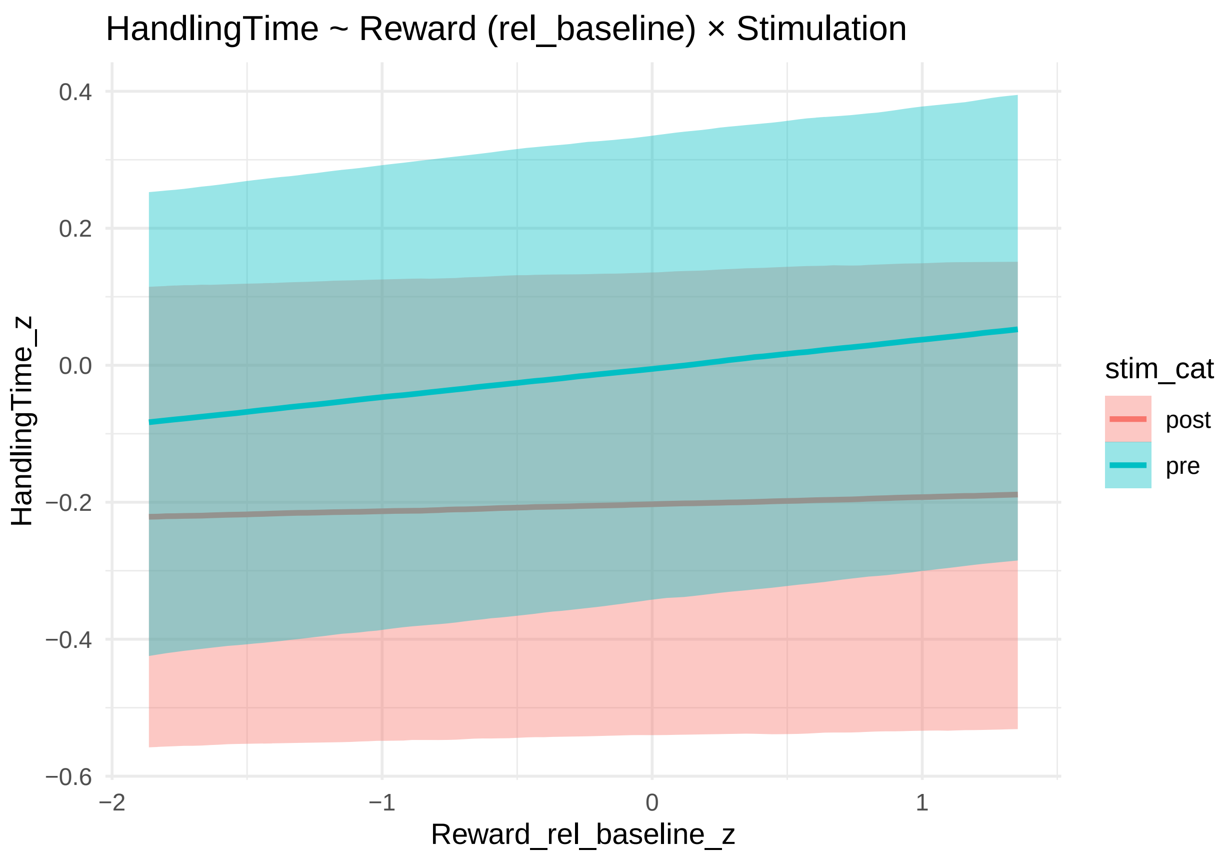
Model — Reward: Relative to Baseline

```
modelB <- brm(
  HandlingTime_z ~ stim_cat * Reward_rel_baseline_z + (1 | Participant_ID),
  data    = df_ht,
  family  = student(),
  chains  = 4,
  iter    = 3000,
  warmup  = 1000,
  cores   = 4,
  seed    = 123,
  file    = "modelB_ht_rel_baseline"
)

summary(modelB)
```

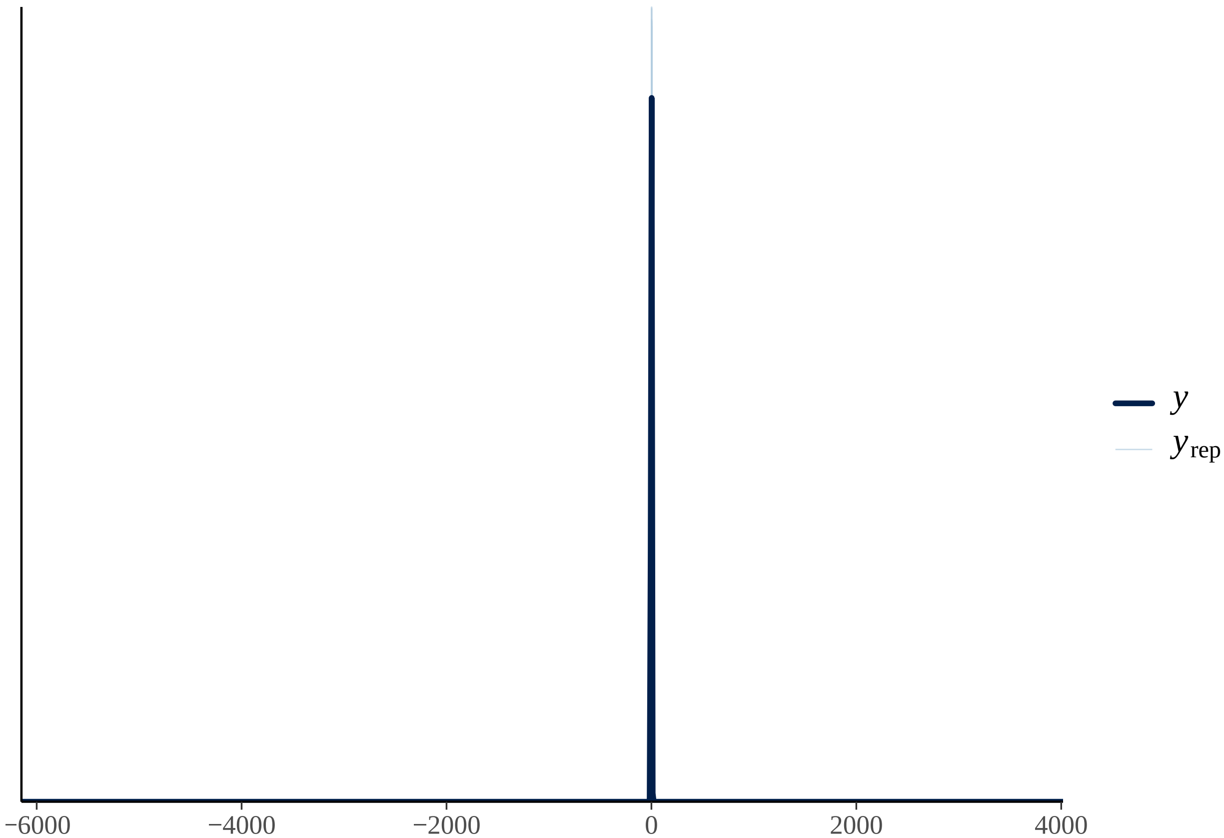
```
##  Family: student
##   Links: mu = identity
## Formula: HandlingTime_z ~ stim_cat * Reward_rel_baseline_z + (1 | Participant_ID)
##    Data: df_ht (Number of observations: 4505)
##   Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup draws = 8000
##
## Multilevel Hyperparameters:
## ~Participant_ID (Number of levels: 21)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept)     0.82      0.14     0.60     1.14 1.00     1349     2110
##
## Regression Coefficients:
##                                    Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                             -0.20      0.17    -0.54     0.14 1.01
## stim_catpre                            0.20      0.01     0.18     0.22 1.00
## Reward_rel_baseline_z                  0.01      0.01    -0.01     0.03 1.00
## stim_catpre:Reward_rel_baseline_z      0.03      0.02     0.00     0.06 1.00
##                                    Bulk_ESS Tail_ESS
## Intercept                               706     1339
## stim_catpre                            5306     4713
## Reward_rel_baseline_z                  4312     3579
## stim_catpre:Reward_rel_baseline_z      4119     3834
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.24      0.01     0.22     0.25 1.00     4075     4057
## nu        1.29      0.04     1.21     1.36 1.00     3967     3601
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
ce <- conditional_effects(
  modelB,
  effects = "Reward_rel_baseline_z:stim_cat"
)

p <- plot(ce, plot = FALSE)[[1]]

p +
  labs(
    title = "HandlingTime ~ Reward (rel_baseline) × Stimulation",
    x = "Reward_rel_baseline_z",
    y = "HandlingTime_z"
  ) +
  theme_minimal()
```




```r
pp_check(modelB, ndraws = 100)
```



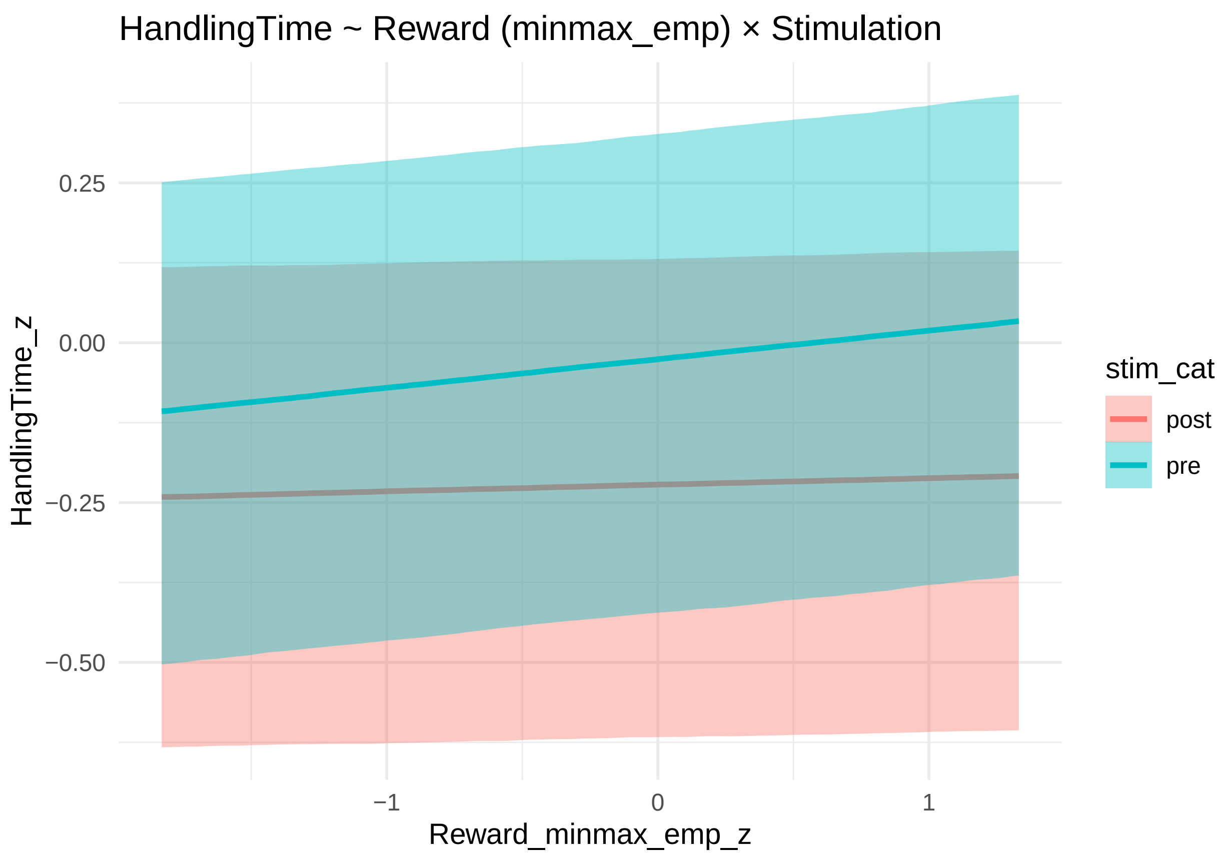
Model— Reward: Empirical Min-Max

```
modelC <- brm(
  HandlingTime_z ~ stim_cat * Reward_minmax_emp_z + (1 | Participant_ID),
  data    = df_ht,
  family  = student(),
  chains  = 4,
  iter    = 3000,
  warmup  = 1000,
  cores   = 4,
  seed    = 123,
  file    = "modelC_ht_minmax_emp"
)

summary(modelC)
```

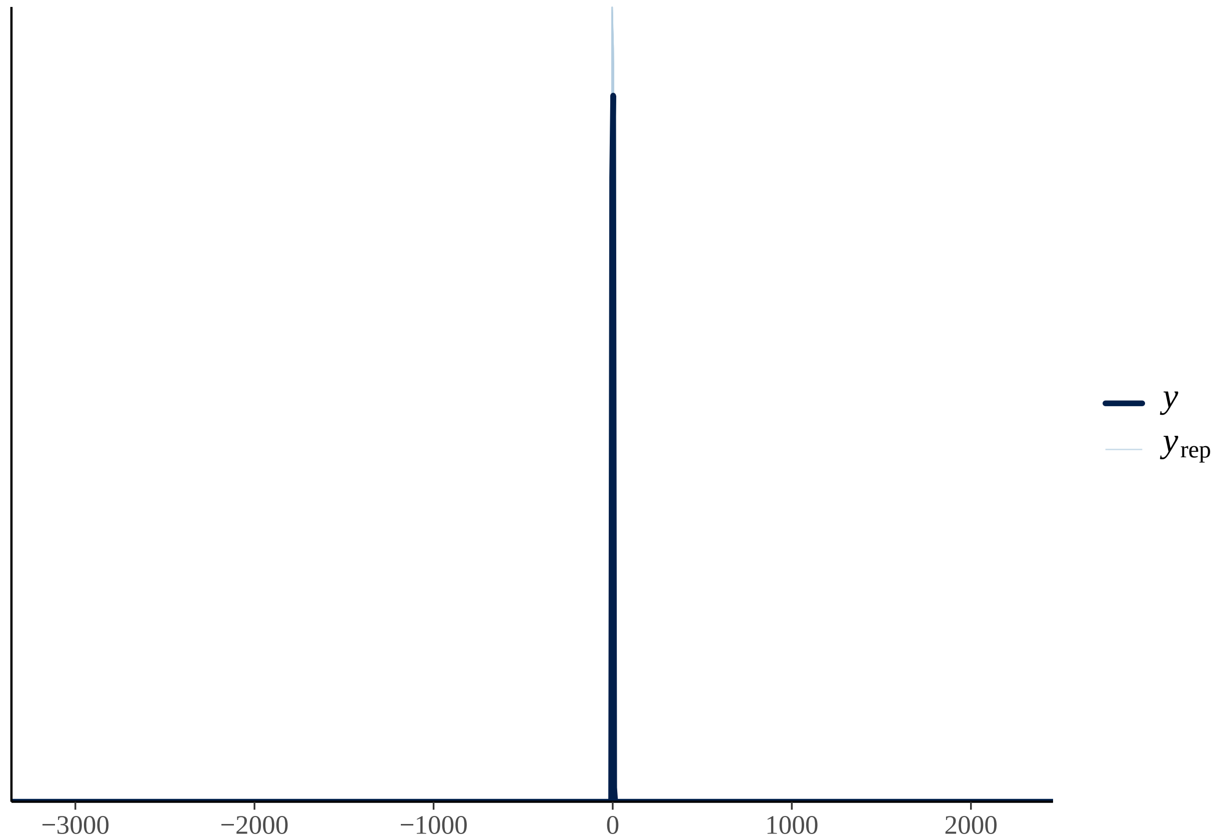
```
##  Family: student
##   Links: mu = identity
## Formula: HandlingTime_z ~ stim_cat * Reward_minmax_emp_z + (1 | Participant_ID)
##    Data: df_ht (Number of observations: 4505)
##   Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup draws = 8000
##
## Multilevel Hyperparameters:
## ~Participant_ID (Number of levels: 21)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept)     0.83      0.14     0.60     1.16 1.00     1237     1973
##
## Regression Coefficients:
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                          -0.23      0.19    -0.62     0.13 1.01
## stim_catpre                         0.20      0.01     0.18     0.22 1.00
## Reward_minmax_emp_z                 0.01      0.01    -0.01     0.03 1.00
## stim_catpre:Reward_minmax_emp_z     0.03      0.02     0.00     0.06 1.00
##                                 Bulk_ESS Tail_ESS
## Intercept                            686      937
## stim_catpre                         5468     4197
## Reward_minmax_emp_z                 4820     4834
## stim_catpre:Reward_minmax_emp_z     4727     4664
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.24      0.01     0.22     0.25 1.00     4620     4811
## nu        1.29      0.04     1.21     1.36 1.00     4546     4065
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
ce <- conditional_effects(
  modelC,
  effects = "Reward_minmax_emp_z:stim_cat"
)

p <- plot(ce, plot = FALSE)[[1]]

p +
  labs(
    title = "HandlingTime ~ Reward (minmax_emp) × Stimulation",
    x = "Reward_minmax_emp_z",
    y = "HandlingTime_z"
  ) +
  theme_minimal()
```




```r
pp_check(modelC, ndraws = 100)
```




```r
looA <- loo(modelA)
looB <- loo(modelB)
looC <- loo(modelC)

loo_compare(looA, looB, looC)
```

```
##        elpd_diff se_diff
## modelC  0.0       0.0
## modelA -0.2       0.1
## modelB -0.3       0.2
```

The model with the highest ELPD is preferred. Large differences (> 4 SE) indicate meaningfully different fit.

```r
cat(rep("=", 72), "\n")
```

```
## = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

```r
cat("HANDLING TIME ~ STIMULATION × REWARD - SUMMARY\n")
```

```
## HANDLING TIME ~ STIMULATION × REWARD - SUMMARY
```

```r
cat(rep("=", 72), "\n\n")
```

```
## = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

```r
cat("IQR filter: trials outside Q1 - 1.5*IQR / Q3 + 1.5*IQR removed per stim_cat\n")
```

```
## IQR filter: trials outside Q1 - 1.5*IQR / Q3 + 1.5*IQR removed per stim_cat
```

```r
cat("Family: Student-t (robust to outliers)\n\n")
```

```
## Family: Student-t (robust to outliers)
```

```r
models_list <- list(
  "A (pct_max_theo)"  = modelA,
  "B (rel_baseline)"  = modelB,
  "C (minmax_emp)"    = modelC
)

int_lookup <- c(
  "A (pct_max_theo)" = "stim_catpre:Reward_pct_max_theo_z",
  "B (rel_baseline)" = "stim_catpre:Reward_rel_baseline_z",
  "C (minmax_emp)"   = "stim_catpre:Reward_minmax_emp_z"
)

rew_lookup <- c(
  "A (pct_max_theo)" = "Reward_pct_max_theo_z",
  "B (rel_baseline)" = "Reward_rel_baseline_z",
  "C (minmax_emp)"   = "Reward_minmax_emp_z"
)

for (nm in names(models_list)) {
  fx <- fixef(models_list[[nm]])

  # Main reward effect
  r_row <- rew_lookup[[nm]]
  r_est <- fx[r_row, "Estimate"]
  r_lo  <- fx[r_row, "Q2.5"]
  r_hi  <- fx[r_row, "Q97.5"]

  # Interaction
  i_row <- int_lookup[[nm]]
  i_est <- fx[i_row, "Estimate"]
  i_lo  <- fx[i_row, "Q2.5"]
  i_hi  <- fx[i_row, "Q97.5"]

  cat(sprintf("Model %s\n", nm))
  cat(sprintf("  Main reward beta  = %6.3f [%6.3f, %6.3f]", r_est, r_lo, r_hi))
  if (r_lo > 0)        cat("  → Higher reward = LONGER handling time\n")
  else if (r_hi < 0)  cat("  → Higher reward = SHORTER handling time\n")
  else                 cat("  → No credible main reward effect\n")
```

```r
  cat(sprintf("  Interaction beta  = %6.3f [%6.3f, %6.3f]", i_est, i_lo, i_hi))
  if (i_lo > 0) {
    cat("  → Pre-stim: reward-dwell relationship STRONGER (more sensitivity)\n")
  } else if (i_hi < 0) {
    cat("  → Pre-stim: reward-dwell relationship WEAKER (less sensitivity)\n")
  } else {
    cat("  → No credible stimulation moderation of reward-handling link\n")
  }
  cat("\n")
}
```

```
## Model A (pct_max_theo)
##   Main reward beta  =  0.010 [-0.009,  0.029]  → No credible main reward effect
##   Interaction beta  =  0.033 [ 0.002,  0.064]  → Pre-stim: reward-dwell relationship STRONGER (more s
##
## Model B (rel_baseline)
##   Main reward beta  =  0.010 [-0.009,  0.030]  → No credible main reward effect
##   Interaction beta  =  0.032 [ 0.002,  0.062]  → Pre-stim: reward-dwell relationship STRONGER (more s
##
## Model C (minmax_emp)
##   Main reward beta  =  0.010 [-0.009,  0.029]  → No credible main reward effect
##   Interaction beta  =  0.034 [ 0.004,  0.064]  → Pre-stim: reward-dwell relationship STRONGER (more s
```

```r
cat(rep("=", 72), "\n")
```

```
## = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```