

Power Simulations for Registered Report

Emmanuel Guizar Rosales

last rendered on: Aug 16, 2024

Summary

- *Sample-size determination* analysis suggests that a sample size of $N = 600$ would result in high statistical power (95%) to detect a smallest effect size of interest (SESOI) of 0.1143 (see Figure 4) for the effect of political affiliation on $\Delta\text{Duration}$. Taking into account potential participant exclusions, we aim for a final sample size of $N = 950$.
- *Effect-size sensitivity* analysis suggests that a final sample size of $N = 950$ enables the detection of a two-way interaction effect of political affiliation with extreme weather exposure of **0.1619** with 95% statistical power (see Figure 7).
- *Effect-size sensitivity* analysis suggests that a final sample size of $N = 950$ enables the detection of a three-way interaction effect of political affiliation with extreme weather exposure and subjective attribution of extreme weather events to climate change of **0.1890** with 95% statistical power (see Figure 11).

Purpose & Rationale

The primary goal of the present analyses is to determine the number of participants required to achieve 95% statistical power to detect a Smallest Effect Size Of Interest (SESOI) for a main effect of political affiliation (Democrat vs. Republican) on attentional information search behavior as assessed by $\Delta\text{Duration}$. That is, we primarily aim for conducting a power analysis for *sample-size determination*, also called *a priori* power analysis (Giner-Sorolla et al. 2024). To this end, we use power simulations with parameters informed by previous studies to assess the statistical power for different sample sizes. This will allow us to decide on a sample size that will provide high statistical power to detect a true and theoretically relevant main effect of political affiliation.

Given this sample size, our secondary goal is to then assess the statistical power to detect different effect sizes for (1) the two-way interaction between political affiliation and extreme weather exposure and (2) the three-way interaction between political affiliation, extreme weather exposure, and the subjective attribution of extreme weather events to climate change. That is, we conduct power analyses for assessing *effect-size sensitivity* for these two- and three-way interactions (Giner-Sorolla et al. 2024).

The present report is organized as follows:

1. We describe all important variables of the planned study and how we will assess them.
2. We derive a SESOI for the main effect of political affiliation on $\Delta\text{Duration}$.
3. We conduct power simulations for *sample-size determination analysis* based on this political affiliation main effect SESOI.
4. We conduct power simulations for *effect-size sensitivity analyses* for a two-way interaction effect between political affiliation and extreme weather exposure.
5. We conduct power simulations for *effect-size sensitivity analyses* for a three-way interaction effect between political affiliation, extreme weather exposure, and subjective attribution of extreme weather events to climate change.

Important Variables

$\Delta\text{Duration}$

Participants will complete 25 trials in a new variant of the Carbon Emission Task (CET, Berger and Wyss 2021) optimized for online process-tracing using MouselabWEB (Willemssen and Johnson 2019). An example trial of the task is displayed and explained in Figure 1.

In all analyses, the criterion (dependent variable) will be $\Delta\text{Duration}$ (`deltaDuration`), which is calculated in each trial as:

$$\Delta\text{Duration} = \frac{(t_{\text{Carbon}_A} + t_{\text{Carbon}_B}) - (t_{\text{Bonus}_A} + t_{\text{Bonus}_B})}{t_{\text{Carbon}_A} + t_{\text{Carbon}_B} + t_{\text{Bonus}_A} + t_{\text{Bonus}_B}}$$

with t representing the summed up dwell time on *Carbon/Bonus* boxes in Option *A/B*. $\Delta\text{Duration}$ varies between -1 and 1 with:

- a value of -1 indicating that the entire dwell time was spent on gathering Bonus information
- a value of 0 indicating that the dwell time was equally split between gathering Bonus and Carbon information

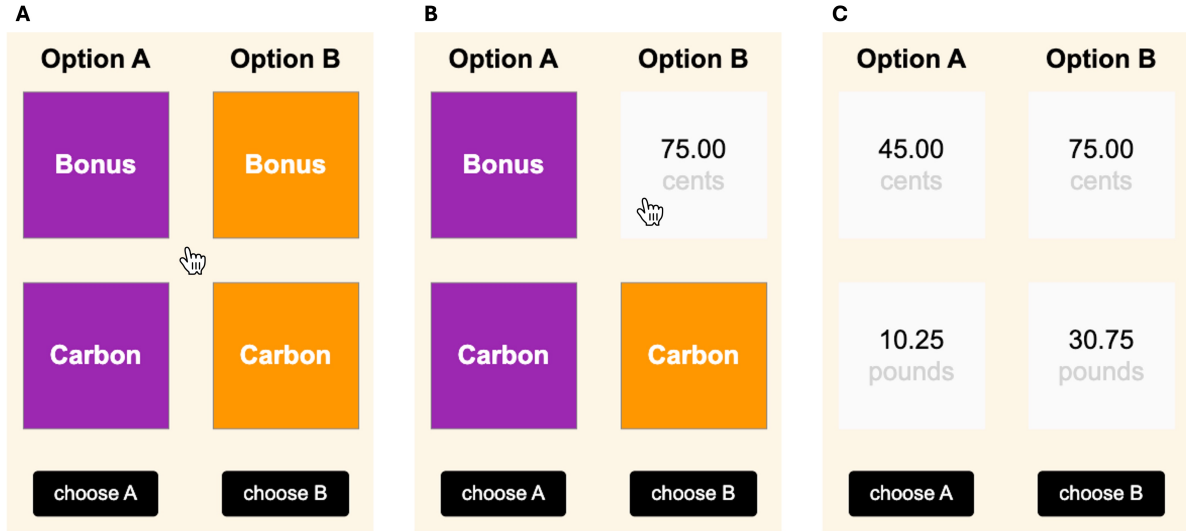


Figure 1: An example trial of the new variant of the CET. **(A)** Participants are presented with two options which are associated with different bonus payment and carbon emission consequences. **(B)** Participants inspect the exact consequences regarding each attribute of each option by hovering their mouse over the respective boxes. Moving the mouse outside of a box occludes the information again. Note that whether the bonus or carbon attributes are displayed in the top row is randomized across participants but held constant within participants. **(C)** By visiting all boxes, participants can acquire all available information in a trial (all boxes opened simultaneously for demonstration purposes only). Note that whether the option maximizing the bonus payment for participants is presented in the left or right row is randomized within participants.

- a value of 1 indication that the entire dwell time was spent on gathering Carbon information

Political Affiliation

We will assess political affiliation using the following question:

Generally speaking, do you think of yourself as a Democrat, Republican or Independent?

Participants will answer on the following 7-point Likert scale:

[1] Strong Republican, [2] Not strong Republican, [3] Independent, close to Republican, [4] Independent, [5] Independent, close to Democrat, [6] Not strong Democrat, [7] Strong Democrat

Additionally, if participants self-identify as [4] Independent, they will be asked:

You said that you think of yourself as an Independent politically. If you had to identify with one party of the two parties, which one would you choose?

Participants will answer on the following scale:

[1] Republican Party, [2] Democratic Party

Based on these questions, we classify participants as either Republican or Democrat as represented by the variable `polAff` that can take on the following values:

[rep] Republican Party, [dem] Democratic Party

Extreme Weather Exposure

For each participant, we will assess the number of extreme weather episodes that occurred in the participants' county of residence in the 30 days prior to study completion. We then create a variable `ewe` which equals to *TRUE* if at least one extreme weather episode occurred in the specified time interval, and *FALSE* otherwise.

Subjective Attribution

We will assess the degree to which participants attribute extreme weather events to climate change (see Ogunbode et al. 2019). Participants will rate their agreement to the following three questions:

1. *Extreme weather events are caused in part by climate change.*
2. *Extreme weather events are a sign that the impacts of climate change are happening now.*

3. *Extreme weather events show us what we can expect from climate change in the future.*

Participants will answer on the following 5-point Likert scale:

[1] *Strongly disagree*, [2] *Somewhat disagree*, [3] *Neither agree nor disagree*, [4] *Somewhat agree*, [5] *Strongly agree*

Subjective attribution of EWE to climate change will be operationalised as the mean agreement to these three statements. Note that Ogunbode et al. (2019), who used the same questions, response options, and aggregation, report a mean of 3.67 and a SD of 0.85 for this variable.

SESOI

As argued in the Registered Report, we hypothesize that (H_1) compared to Republicans, Democrats prioritize searching for and attending to carbon over bonus information during decision-making in the CET. That is, Democrats display higher $\Delta\text{Duration}$ values compared to Republicans.

While H_1 describes the direction of the expected effect, it does not specify the expected magnitude of the effect. This later point is clarified by asking the question what would be the smallest effect size that researchers would still consider theoretically relevant, i.e., what is the smallest effect size of interest (SESOI)? We argue for such a SESOI on theoretical grounds and based on previous MouselabWEB research.

In mouselabWEB studies, it is standard practice to filter out any information acquisition lasting less than 200 msec because such very short (spurious) acquisitions are very unlikely to be consciously processed (Willemsen and Johnson 2019). Therefore, we derive the SESOI based on the consideration that for an effect to be meaningful, the increase in $\Delta\text{Duration}$ from a Republican to a Democrat should be due to an increase of the time spent gathering carbon (relative to bonus) information of at least 200 msec. We need to translate these considerations into the metric of $\Delta\text{Duration}$.

Suppose that Republicans on average spend t_{Carbon} msec on gathering carbon information and t_{Bonus} msec on gathering bonus information in each trial, with a total time of gathering any information $t_{\text{Total}} = t_{\text{Carbon}} + t_{\text{Bonus}}$. Thus, for Republicans, this results in:

$$\Delta\text{Duration}_{\text{Rep}} = \frac{t_{\text{Carbon}} - t_{\text{Bonus}}}{t_{\text{Total}}}$$

Suppose that Democrats have the same t_{Total} as Republicans, but they spend 200 msec more on gathering carbon information and, correspondingly, 200 msec less on gathering bonus information. Thus, for Democrats, this results in:

$$\Delta Duration_{Dem} = \frac{(t_{Carbon} + 200) - (t_{Bonus} - 200)}{t_{Total}} = \frac{400 + t_{Carbon} - t_{Bonus}}{t_{Total}}$$

Therefore, the difference in $\Delta Duration$ for Democrats and Republicans is:

$$\begin{aligned} \Delta Duration_{Dem} - \Delta Duration_{Rep} &= \frac{400 + t_{Carbon} - t_{Bonus}}{t_{Total}} - \frac{t_{Carbon} - t_{Bonus}}{t_{Total}} \\ &= \frac{(400 + t_{Carbon} - t_{Bonus}) - (t_{Carbon} - t_{Bonus})}{t_{Total}} \\ &= \frac{400}{t_{Total}} \end{aligned}$$

Thus, we define our SESOI as:

$$SESOI_{polAff} = \frac{400 \text{ msec}}{t_{Total}}$$

As this definition shows, the SESOI depends on our expectation regarding t_{Total} . We form these expectations based on previous research. In our lab, we recently conducted another MouselabWEB study whose setup was very similar to the one described in the Registered Report (Studler et al., in preparation). In short, student participants completed a behavioral task in which they searched for and attended to information presented in a 2-by-2 decision matrix adapted from Reeck, Wall, and Johnson (2017). The average time participants spent on acquiring information in each trial (t_{Total}) was 3.3 seconds. In the planned study we will assess a representative US sample. That is, the average age of our sample will be higher than in a typical student sample. As processing speed is known to decline with age (Salthouse 2000), we assume a slightly higher total information acquisition time in our sample of $t_{Total} = 3500 \text{ msec}$. Therefore, we define our SESOI as:

$$SESOI_{polAff} = \frac{400 \text{ msec}}{3500 \text{ msec}} = 0.1143$$

To account for uncertainty in this estimation, we also provide sample-size determination analysis results based on $t_{Total} = 4000 \text{ msec}$ and $t_{Total} = 3000 \text{ msec}$. This results in a high and low estimate of SESOI:

$$\begin{aligned} SESOI_{polAff_{high}} &= \frac{400 \text{ msec}}{3000 \text{ msec}} = 0.1333 \\ SESOI_{polAff_{low}} &= \frac{400 \text{ msec}}{4000 \text{ msec}} = 0.1000 \end{aligned}$$

Main Effect

As argued in the Registered Report, we hypothesize:

(H₁) *Compared to Republicans, Democrats prioritize searching for and attending to carbon over bonus information during decision-making in the CET. That is, Democrats display higher Δ Duration values compared to Republicans.*

```
# Create a data frame with predicted true effects

# Smallest Effect Size Of Interest (SESOI)
SESOI <- 0.4/3.5

# Betas
beta_p <- SESOI

# Predicted "true" effects
polAff_trueEffects <- expand_grid(
  polAff = factor(c("rep", "dem"), levels = c("rep", "dem"))
) %>%
  add_contrast("polAff", contrast = "anova", colnames = "X_p") %>%
  mutate(
    trueDeltaDuration =
      0 + # intercept
      X_p * beta_p # main effect polAff
  )
```

Data Simulation Function

We first define a function that simulates data for the main effect of political affiliation on Δ Duration: FUN_sim. The function will simulate data according to the following model, expressed in lme4-lingo:

```
deltaDuration ~ polAff + (1|subj) + (1|trial)
```

The function FUN_sim takes, among others, the following important arguments:

- **n_subj: Number of subjects.** Changing this parameter allows us to assess statistical power for different sample sizes.
- **beta_0: Fixed intercept (grand mean).** We assume that the average participant (irrespective of political affiliation or any other predictor variable) spends about the same time on searching for and attending to carbon as to bonus information in the CET.

Therefore, we set `beta_0` to zero. The effects of predictor variables will be modeled as deviations from this grand mean.

- **beta_p: Fixed effect of political affiliation.** This value represents the average difference in $\Delta\text{Duration}$ between Democrats and Republicans (Democrats - Republicans). As discussed above, we set this value to 0.1143 by default, and we assess the effect of changing this variable on statistical power.
- **subj_0: By-subject random intercept SD.** We simulate that a participants' deviations from the grand mean for $\Delta\text{Duration}$ follows a normal distribution with a mean of 0 and a standard deviation of `subj_0` = 0.29. We base our default value on a study by Reeck, Wall, and Johnson (2017). They investigated whether variability in information search behavior is driven predominantly by differences in the features of a choice (i.e., in our case: the relative differences between carbon and bonus outcomes in options A and B) or by individual differences. To this end, they predicted information acquisition using a intercept-only model that included random intercepts for subjects and items. They estimated the random intercept of subjects to be 0.29.
- **trial_0: By-trial random intercept SD.** We simulate that a items' deviations from the grand mean for $\Delta\text{Duration}$ follows a normal distribution with a mean of 0 and a standard deviation of `trial_0` = 0.04. Again, we base our default value on the study by Reeck, Wall, and Johnson (2017), who estimated a random intercept for trials of 0.04.
- **sigma: Trial-level noise (error) SD.** We model the error SD to be of the same size as the sum of the random intercept SDs = $0.29 + 0.04 = 0.33$. We also report simulation results for an error SD that is twice the size of the random intercept SDs, i.e., 0.66.

The function `FUN_sim` is defined below:

```
# define data simulation function
FUN_sim <- function(
  n_subj      = 1000, # number of subjects
  n_subj_prop = c(.5, .5), # proportion of republican and democrat subjects
  n_trial     = 25, # number of trials
  beta_0      = 0, # intercept (grand mean) for deltaDuration
  beta_p      = 0.4/3.5, # effect of political affiliation on deltaDuration
  subj_0      = .29, # by-subject random intercept sd for dt carbon
  trial_0     = .04, # by-trial random intercept sd
  sigma       = 1*(.29+.04), # residual (error) sd

  truncNums   = TRUE, # should impossible deltaDuration values be truncated?
  setSeed     = NULL # seed number to achieve reproducible results. Set to NULL for
) {

  # set seed to achieve reproducible results for demonstration purposes
```



```

set.seed(setSeed)

# simulate data for dwell time on carbon information
dataSim <-
  # add random factor subject
  add_random(subj = n_subj) %>%
  # add random factor trial
  add_random(trial = n_trial) %>%
  # add between-subject factor political affiliation (with anova contrast)
  add_between("subj", polAff = c("rep", "dem"), .prob = n_subj_prop*n_subj, .shuffle = FALSE) %>%
  add_contrast("polAff", colnames = "X_p", contrast = "anova") %>%
  # add by-subject random intercept
  add_ranef("subj", S_0 = subj_0) %>%
  # add by-trial random intercept
  add_ranef("trial", T_0 = trial_0) %>%
  # add error term
  add_ranef(e_st = sigma) %>%
  # add response values
  mutate(
    # add together fixed and random effects for each effect
    B_0 = beta_0 + S_0 + T_0,
    B_p = beta_p,
    # calculate dv by adding each effect term multiplied by the relevant
    # effect-coded factors and adding the error term
    deltaDuration = B_0 + (B_p * X_p) + e_st
  )

# unset seed
set.seed(NULL)

# truncate impossible deltaDurations
if(truncNums) {
  dataSim <- dataSim %>%
    mutate(deltaDuration = if_else(deltaDuration < -1, -1,
      if_else(deltaDuration > 1, 1, deltaDuration)))
}

# run a linear mixed effects model and check summary
mod <- lmer(
  deltaDuration ~ polAff + (1 | subj) + (1 | trial),
  data = dataSim,
)

```

```

mod.sum <- summary(mod)

# get results in tidy format
mod.broom <- broom.mixed::tidy(mod)

return(list(
  dataSim = dataSim,
  modelLmer = mod,
  modelResults = mod.broom
))
}

```

We call the function once and extract the results of this single simulation:

```

# Call function
out <- FUN_sim(
  n_subj      = 1000, # number of subjects
  n_subj_prop = c(.5, .5), # proportion of republican and democrat subjects
  n_trial     = 25, # number of trials
  beta_0      = 0, # intercept (grand mean) for deltaDuration
  beta_p      = 0.4/3.5, # effect of political affiliation on deltaDuration
  subj_0      = .29, # by-subject random intercept sd for dt carbon
  trial_0     = .04, # by-trial random intercept sd
  sigma       = 1*(.29+.04), # residual (error) sd

  truncNums   = TRUE, # should impossible deltaDuration values be truncated?
  setSeed     = 1234 # seed number to achieve reproducible results. Set to NULL for
)

# Get results table
resultsTable <- out$modelResults %>%
  select(-c(std.error, statistic, df)) %>%
  mutate(across(where(is_double), ~ round(.x, 4))) %>%
  knitr::kable()
formulaUsedForFit <- paste(as.character(formula(out$modelLmer))[c(2,1,3)], collapse = " ")

# Create plot
p.demo.mainEffect <- out$dataSim %>%
  ggplot(aes(x = polAff, y = deltaDuration, color = polAff)) +
  geom_hline(yintercept = 0) +
  geom_violin(alpha = 0.3) +

```

```

geom_point(
  data = polAff_trueEffects,
  mapping = aes(x = polAff, y = trueDeltaDuration),
  shape = "circle open",
  size = 3.5,
  stroke = 2,
  color = "black"
) +
stat_summary(
  fun = mean,
  fun.min = \(x){mean(x) - sd(x)},
  fun.max = \(x){mean(x) + sd(x)},
  position = position_dodge(width = .9)
) +
ggrepel::geom_label_repel(
  data = polAff_trueEffects,
  mapping = aes(x = polAff, y = trueDeltaDuration, label = round(trueDeltaDuration, 4)),
  color = "black",
  box.padding = 1
) +
scale_color_manual(values = c("red", "dodgerblue")) +
scale_y_continuous(breaks = seq(-1, 1, .2)) +
labs(title = "Demo Output of One Simulation for Main Effect") +
theme_bw() +
theme(legend.position = "none")

```

Figure 2 visualizes the results of this single simulation and Table 1 summarises the statistical results of fitting the actual model used in data generation to the simulated data.

Table 1: Statistical results of one simulation created using FUN_sim. Data was fit using $\text{deltaDuration} \sim \text{polAff} + (1 \mid \text{subj}) + (1 \mid \text{trial})$.

effect	group	term	estimate	p.value
fixed	NA	(Intercept)	-0.0188	0.1105
fixed	NA	polAff.dem-rep	0.0896	0.0000
ran_pars	subj	sd__(Intercept)	0.2841	NA
ran_pars	trial	sd__(Intercept)	0.0363	NA
ran_pars	Residual	sd__Observation	0.3223	NA

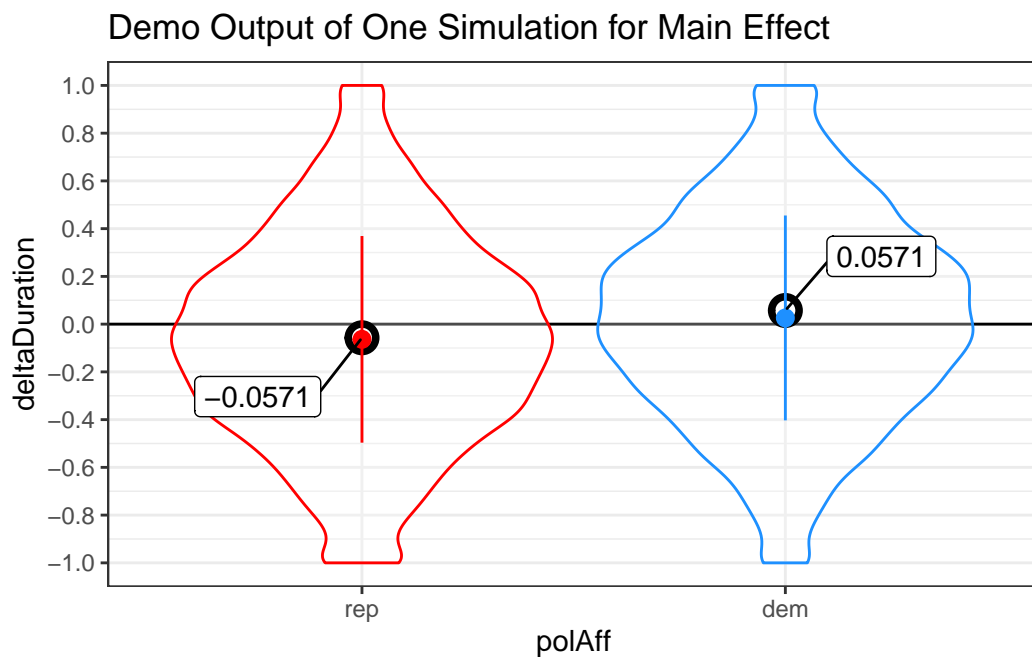


Figure 2: Visual representation of results of one simulation created using `FUN_sim`. Violin plots display the full distribution of the data. Points and surrounding lines indicate the mean \pm 1 SD. The black horizontal line displays the true sample mean and the black open circles indicate the true means for each cell.

Power Simulation

We now simulate multiple random samples drawn from the same synthetic population with a known true effect of political affiliation. For each random sample, we fit our statistical model to the data. The statistical power to detect a true effect of political affiliation is calculated as the proportion of significant effects out of the total number of simulations. We aim for a statistical power of 95%.

In the following code, the simulations are calculated. We do not recommend executing this code junk as it takes several hours to run.

```
FUN_sim_pwr <- function(sim, ...){
  out <- FUN_sim(...)
  modelResults <- out$modelResults %>%
    mutate(sim = sim) %>%
    relocate(sim)
  return(modelResults)
}

# How many simulations should be run?
n_sims <- 1000

# What are the breaks for number of subjects we would like to calculate power for?
breaks_subj <- seq(200, 1000, 200)

# What are the breaks for SESOI?
breaks_sesoi <- c(0.4/3, 0.4/3.5, 0.4/4)

# What are the breaks for different error SDs?
breaks_sigma <- c(1:2)*(.29+.04)

res_mainEffect <- tibble()
for (s in seq_along(breaks_sigma)) {

  res_sesoi <- tibble()
  for (sesoi in seq_along(breaks_sesoi)) {

    res_nSubj <- tibble()
    for (nSubj in seq_along(breaks_subj)) {

      # Give feedback regarding which model is simulated
      cat(paste0(
        "Simulation:\n",
```

```

    "  sigma = ", round(breaks_sigma[s], 4), "\n",
    "  sesoi = ", round(breaks_sesoi[sesoi], 4), "\n",
    "  nSubject = ", breaks_subj[nSubj], "\n"
  ))

  # Start timer
  cat(paste0("Start date time: ", lubridate::now(), "\n"))
  tic()

  # Loop over simulations
  pwr <- map_df(
    1:n_sims,
    FUN_sim_pwr,
    n_subj = breaks_subj[nSubj],
    beta_p = breaks_sesoi[sesoi],
    sigma = breaks_sigma[s]
  )

  # Stop timer and calculate elapsed time
  elapsed_time <- toc(quiet = TRUE)
  elapsed_seconds <- elapsed_time$toc - elapsed_time$tic
  elapsed_minutes <- elapsed_seconds / 60
  cat(paste0("End date time: ", lubridate::now(), "\n"))
  cat("Elapsed time: ", elapsed_minutes, " minutes\n\n")

  # Add number of subjects to pwr
  pwr <- pwr %>%
    mutate(
      nSubjects = breaks_subj[nSubj],
      sesoi = breaks_sesoi[sesoi],
      sigma = breaks_sigma[s]
    )

  # Add results to the results table
  res_nSubj <- res_nSubj %>%
    rbind(pwr)
}

# Add results to the results table
res_sesoi <- res_sesoi %>%
  rbind(res_nSubj)
}

```

```

# Add results to the results table
res_mainEffect <- res_mainEffect %>%
  rbind(res_sesoi)
}

res_mainEffect.summary <- res_mainEffect %>%
  filter(term == "polAff.dem-rep") %>%
  group_by(sigma, sesoi, nSubjects) %>%
  summarise(
    power = mean(p.value < 0.05),
    ci.lower = binom.confint(power*n_sims, n_sims, methods = "exact")$lower,
    ci.upper = binom.confint(power*n_sims, n_sims, methods = "exact")$upper,
    .groups = 'drop'
  ) %>%
  mutate(
    sigma_fact = factor(format(round(sigma, 4), nsmall = 4)),
    sigma_level = match(sigma_fact, levels(sigma_fact)),
    sesoi_fact = factor(format(round(sesoi, 4), nsmall = 4)),
    sesoi_level = match(sesoi_fact, levels(sesoi_fact))
  )

# Save results in a list object
time <- format(Sys.time(), "%Y%m%d_%H%M")
fileName <- paste0("res_mainEffect", "_", time, ".RDS")
saveRDS(
  list(
    res_mainEffect = res_mainEffect,
    res_mainEffect.summary = res_mainEffect.summary
  ),
  file = file.path("../powerSimulationsOutput", fileName)
)

```

We retrieve pre-run results:

```

# Load power simulation data
resList_mainEffect <- readRDS(file.path("../powerSimulationsOutput", "res_mainEffect_20240811"))
resList_mainEffect.summary <- resList_mainEffect$res_mainEffect.summary

# Extract power values for some specific effect sizes at N = 1000
chosenN <- 600
chosenSigma <- c("0.3300", "0.6600")

```

```

chosenSESOI <- c("0.1000", "0.1143")
powerValues <- resList_mainEffect.summary %>%
  filter(sigma_fact %in% chosenSigma) %>%
  filter(sesoi_fact %in% chosenSESOI) %>%
  filter(nSubjects %in% chosenN) %>%
  mutate(power_str = paste0(round(ci.lower*100, 2), "%")) %>%
  pull(power_str)

# Extract number of simulations
label_nSimulations <- resList_mainEffect$res_mainEffect$sim %>% n_distinct()

```

Figure 3 displays the distribution of estimated fixed effects across all simulations. The figure shows that the estimated fixed effects are close to the true ones provided as input in the data simulation function, validating that simulations worked as expected.

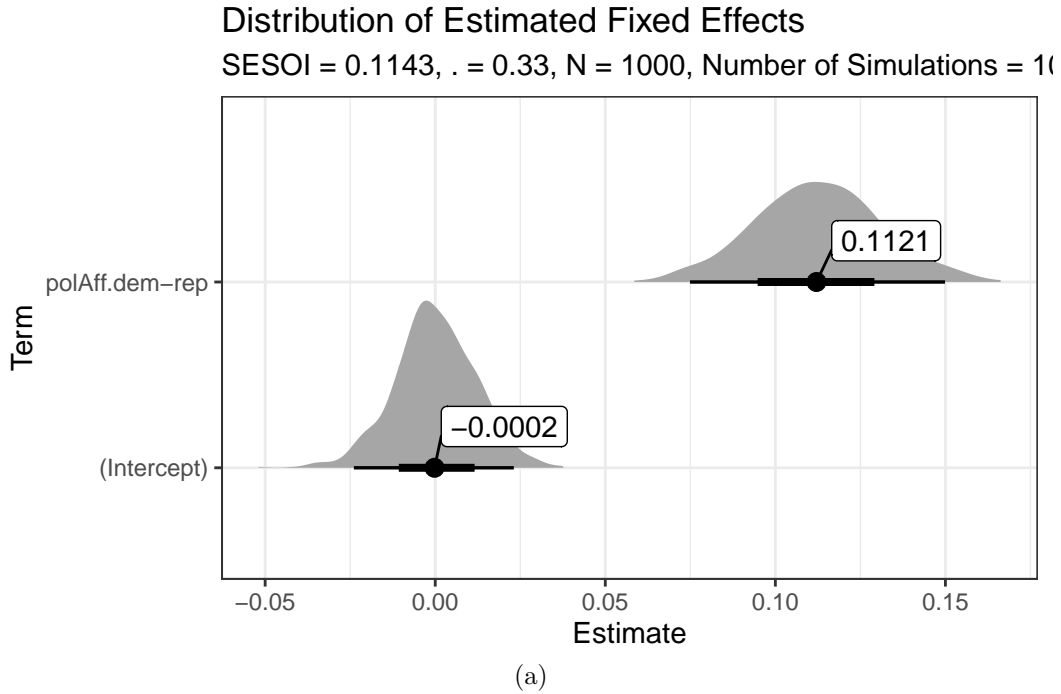


Figure 3: Distribution of estimated fixed effects resulting from 1000 simulations for the model $\text{deltaDuration} \sim \text{polAff} + (1 \mid \text{subj}) + (1 \mid \text{trial})$. Shaded area represent densities, annotated points indicate medians, and thick and thin lines represent 66% and 95% quantiles.

Figure 4 shows results of our sample-size determination analyses. We find that a sample size of 600 provides a statistical power of 95.4% (lower bound of 95%-CI) even under the most conservative assumptions (SESOI = 0.1000, Error SD = 0.6600).

We optimize the study design to detect a true SESOI for political affiliation. However, we are also interested in two-way and three-way interaction effects, which are known to require greater sample sizes to achieve the same statistical power as for main effects. Moreover, greater sample sizes are more likely to accurately represent target populations with respect to variables like exposure to extreme weather events and subjective attribution of extreme weather events to climate change. Therefore, we opt for a sample size of $N = 1000$ for further effect-size sensitivity analyses regarding interaction effects.

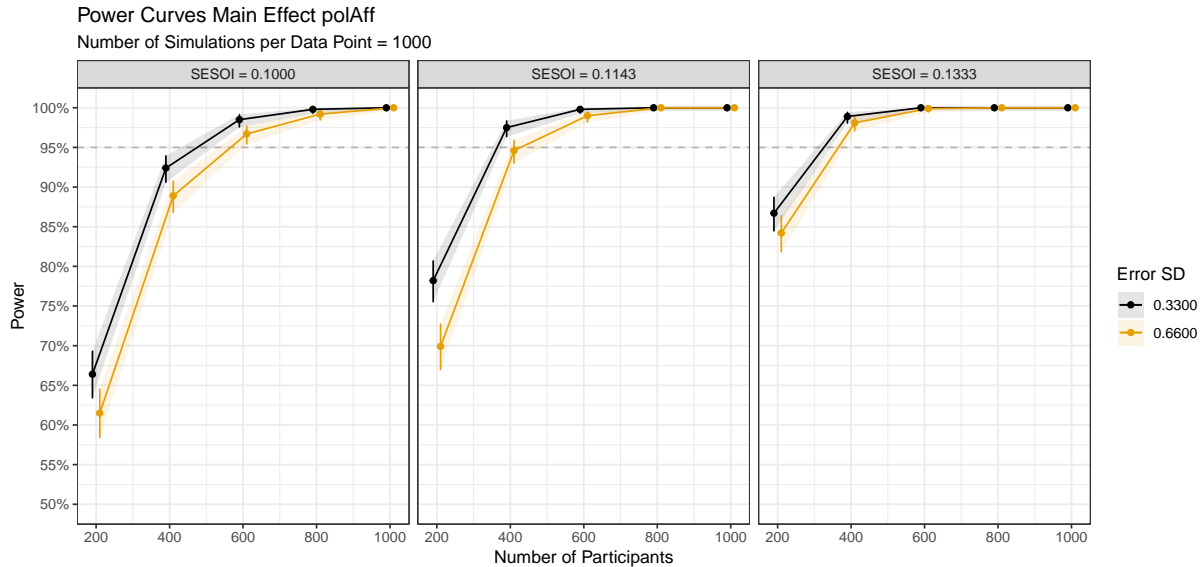


Figure 4: Power curves for the main effect of polAff. Points represent statistical power surrounded by a 95%-CI based on 1000 simulations with $\alpha = 0.05$.

Two-Way Interaction Effect

As argued in the Registered Report, we hypothesize:

(H₂): *The effect of political affiliation on Δ Duration is greater among individuals with prior exposure to extreme weather events compared to individuals without such exposure. That is, extreme weather exposure positively interacts with political affiliation in predicting Δ Duration.*

```
# Create a data frame with predicted true effects

# Smallest Effect Size Of Interest (SESOI)
SESOI <- 0.4/3.5

# Betas
```

```

beta_p <- SESOI
beta_e <- 0
beta_p_e_inx <- SESOI

# Predicted "true" effects
polAff_ewe_trueEffects <- expand_grid(
  polAff = factor(c("rep", "dem"), levels = c("rep", "dem")),
  ewe = factor(c(FALSE, TRUE), levels = c(FALSE, TRUE))
) %>%
  add_contrast("polAff", contrast = "anova", colnames = "X_p") %>%
  add_contrast("ewe", contrast = "anova", colnames = "X_e") %>%
  mutate(
    trueDeltaDuration =
      0 + # intercept
      X_p * beta_p + # main effect polAff
      X_e * beta_e + # main effect ewe
      X_p * X_e * beta_p_e_inx # interaction effect polAff*ewe
  )

```

SESOI for Two-Way Interaction

For deriving a SESOI for the two-way interaction of interest, similar considerations apply as in the case of the SESOI for the main effect of interest. In this section, we make these considerations explicit. We start by noticing that the complete fixed two-way interaction $\text{polAff} \times \text{ewe}$ is modeled as:

$$\Delta \text{Duration} = \beta_0 + \beta_1 \cdot \text{polAff} + \beta_2 \cdot \text{ewe} + \beta_3 \cdot (\text{polAff} \times \text{ewe})$$

By rearranging terms, one can show that the effect of polAff is given by:

$$\text{Effect}_{\text{polAff}} = \beta_1 + \beta_3 \cdot \text{ewe}$$

Now, let's calculate this effect for two individuals who differ in their levels of ewe . As ewe is a binary variable, we have two types of individuals: individuals with ($\text{ewe} = 1$) and without ($\text{ewe} = 0$) extreme weather exposure. For an individual without extreme weather exposure, the effect of polAff will be:

$$\text{Effect}_{\text{noEWE}} = \beta_1 + \beta_3 \cdot 0 = \beta_1$$

For an individual with extreme weather exposure, the effect of polAff will be:

$$Effect_{EWE} = \beta_1 + \beta_3 \cdot 1 = \beta_1 + \beta_3$$

The difference in the effect of polAff between these two individuals is given by:

$$Effect_{EWE} - Effect_{noEWE} = (\beta_1 + \beta_3) - \beta_1 = \beta_3$$

We consider this difference as theoretically relevant if it is at least of the same size as the SESOI for the effect of polAff:

$$SESOI_{polAff \times ewe} = Effect_{EWE} - Effect_{noEWE} = SESOI_{polAff} = 0.1143$$

Data Simulation Function

We next define a function that simulates data for the two-way interaction effect of political affiliation with extreme weather exposure on Δ Duration: `FUN_sim_2wayInt`. The function will simulate data according to the following model, expressed in lme4-lingo:

```
deltaDuration ~ polAff * ewe + (1|subj) + (1|trial)
```

The function `FUN_sim_2wayInt` takes, among others, the following important arguments (in addition to the arguments discussed for `FUN_sim`):

- **beta_p: Fixed main effect of political affiliation.** Compatible with H_1 , we keep this value at the SESOI of 0.1143. That is, we model that the average effect of political affiliation across all participants, irrespective of their extreme weather exposure, is 0.1143.
- **beta_e: Fixed main effect of extreme weather exposure.** As we are interested in the moderating role of extreme weather exposure, we set this main effect to zero. That is, we assume that the effect of extreme weather exposure is highly dependent on participants' political affiliation.
- **beta_p_e_inx: Fixed two-way interaction effect of political affiliation and extreme weather exposure.** We set this initial value to the SESOI derived above, but we investigate how changing this effect size impacts statistical power, as we are conducting effect-size sensitivity analyses for interaction effects.

The function `FUN_sim_2wayInt` is defined below:

```

# define data simulation function
FUN_sim_2wayInt <- function(
  n_subj          = 1000, # number of subjects
  n_subj_prop_p   = c(.5, .5), # proportion of republican and democrat subjects
  n_subj_prop_e   = c(.5, .5), # proportion of subjects without and with extreme weather exposure
  n_trial         = 25, # number of trials
  beta_0          = 0, # intercept (grand mean) for deltaDuration
  beta_p          = 0.4/3.5, # main effect of political affiliation (polAff)
  beta_e          = 0, # main effect of extreme weather exposure (ewe)
  beta_p_e_inx    = 0.4/3.5, # two-way interaction effect of polAff and ewe
  subj_0          = .29, # by-subject random intercept sd for dt carbon
  trial_0         = .04, # by-trial random intercept sd
  sigma           = 1*(.29+.04), # residual (error) sd

  truncNums       = TRUE, # should impossible numbers be truncated?
  setSeed         = NULL, # seed number to achieve reproducible results. Set to NULL for random
) {

  # set seed to achieve reproducible results for demonstration purposes
  set.seed(setSeed)

  # simulate data for dwell time on carbon information
  dataSim <-
    # add random factor subject
    add_random(subj = n_subj) %>%
    # add random factor trial
    add_random(trial = n_trial) %>%
    # add between-subject factor political affiliation (with anova contrast)
    add_between("subj", polAff = c("rep", "dem"), .prob = n_subj_prop_p*n_subj, .shuffle = TRUE) %>%
    add_contrast("polAff", colnames = "X_p", contrast = "anova") %>%
    # add between-subject factor extreme weather exposure (with anova contrast)
    add_between("subj", ewe = c(FALSE, TRUE), .prob = n_subj_prop_e*n_subj, .shuffle = TRUE) %>%
    add_contrast("ewe", colnames = "X_e", contrast = "anova") %>%
    # add by-subject random intercept
    add_ranef("subj", S_0 = subj_0) %>%
    # add by-trial random intercept
    add_ranef("trial", T_0 = trial_0) %>%
    # add error term
    add_ranef(e_st = sigma) %>%
    # add response values
    mutate(
      # add together fixed and random effects for each effect

```

```

    B_0 = beta_0 + S_0 + T_0,
    B_p = beta_p,
    B_e = beta_e,
    B_p_e_inx = beta_p_e_inx,
    # calculate dv by adding each effect term multiplied by the relevant
    # effect-coded factors and adding the error term
    deltaDuration =
      B_0 + e_st +
      (X_p * B_p) +
      (X_e * B_e) +
      (X_p * X_e * B_p_e_inx)
  )

# truncate impossible deltaDurations
if(truncNums) {
  dataSim <- dataSim %>%
    mutate(deltaDuration = if_else(deltaDuration < -1, -1,
      if_else(deltaDuration > 1, 1, deltaDuration)))
}

# run a linear mixed effects model and check summary
mod <- lmer(
  deltaDuration ~ polAff*ewe + (1 | subj) + (1 | trial),
  data = dataSim
)
mod.sum <- summary(mod)

# get results in tidy format
mod.broom <- broom.mixed::tidy(mod)

return(list(
  dataSim = dataSim,
  modelLmer = mod,
  modelResults = mod.broom
))
}

```

We call the function once and extract the results of this single simulation:

```

out <- FUN_sim_2wayInt(
  n_subj      =      1000, # number of subjects

```

```

n_subj_prop_p = c(.5, .5), # proportion of republican and democrat subjects
n_subj_prop_e = c(.5, .5), # proportion of subjects without and with extreme weather exposure
n_trial      =      25, # number of trials
beta_0       =      0, # intercept (grand mean) for deltaDuration
beta_p       =    0.4/3.5, # main effect of political affiliation (polAff)
beta_e       =      0, # main effect of extreme weather exposure (ewe)
beta_p_e_inx =    0.4/3.5, # two-way interaction effect of polAff and ewe
subj_0       =    .29, # by-subject random intercept sd for dt carbon
trial_0      =    .04, # by-trial random intercept sd
sigma        = 1*(.29+.04), # residual (error) sd

truncNums    =      TRUE, # should impossible numbers be truncated?
setSeed      =    1234 # seed number to achieve reproducible results. Set to NULL for random
)

# Get results table
resultsTable <- out$modelResults %>%
  select(-c(std.error, statistic, df)) %>%
  mutate(across(where(is_double), ~ round(.x, 4))) %>%
  knitr::kable()
formulaUsedForFit <- paste(as.character(formula(out$modelLmer))[c(2,1,3)], collapse = " ")

# Create plot
p.demo.2wayInt <- out$dataSim %>%
  ggplot(aes(x = ewe, y = deltaDuration, color = polAff)) +
  geom_hline(yintercept = 0) +
  geom_violin(alpha = 0.3) +
  geom_point(
    data = polAff_ewe_trueEffects,
    mapping = aes(x = ewe, y = trueDeltaDuration, fill = polAff),
    shape = "circle open",
    size = 3.5,
    stroke = 2,
    color = "black",
    position = position_dodge(width = .9),
    show.legend = FALSE
  ) +
  stat_summary(
    fun = mean,
    fun.min = \(x){mean(x) - sd(x)},
    fun.max = \(x){mean(x) + sd(x)},
    position = position_dodge(width = .9)
  )

```

```

) +
ggrepel::geom_label_repel(
  data = polAff_ewe_trueEffects,
  mapping = aes(x = ewe, y = trueDeltaDuration, fill = polAff, label = round(trueDeltaDuration, 2)),
  color = "black",
  box.padding = 1,
  position = position_dodge(width = .9),
  show.legend = FALSE
) +
scale_color_manual(values = c("red", "dodgerblue")) +
scale_fill_manual(values = c("white", "white")) +
scale_y_continuous(breaks = seq(-1, 1, .2)) +
labs(title = "Demo Output of One Simulation for Two-Way Interaction") +
theme_bw()

```

Figure 5 visualizes the results of this single simulation and Table 2 summarizes the statistical results of fitting the actual model used in data generation to the simulated data.

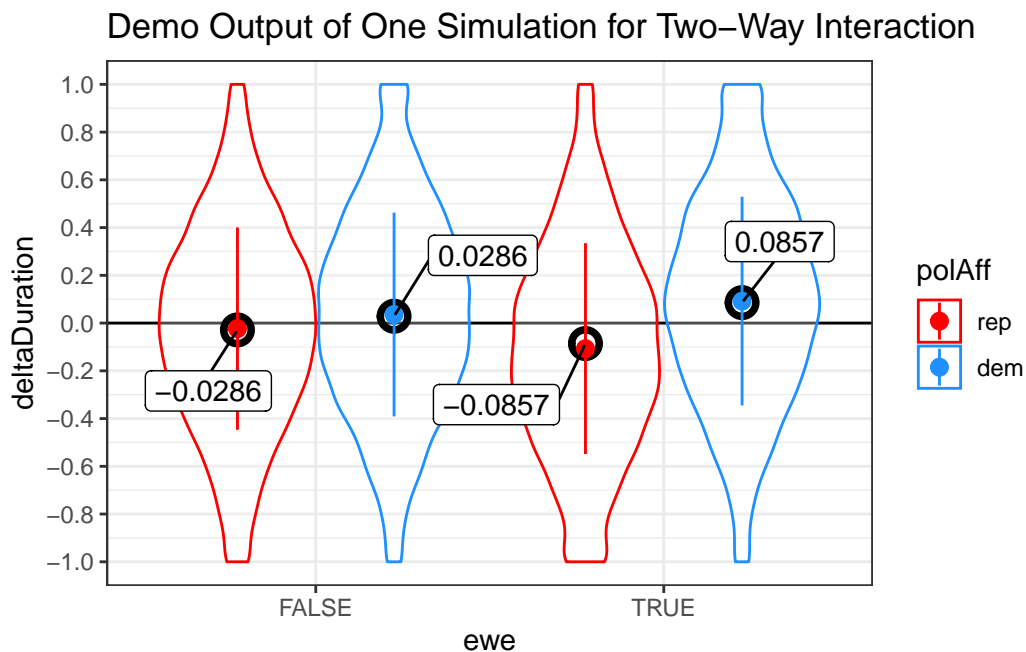


Figure 5: Visual representation of results of one simulation created using FUN_sim_2wayInt. Violin plots display the full distribution of the data. Points and surrounding lines indicate the mean \pm 1 SD. The black horizontal line displays the true sample mean and the black open circles indicate the true means for each cell.

Table 2: Statistical results of one simulation created using FUN_sim_2wayInt. Data was fit using $\text{deltaDuration} \sim \text{polAff} * \text{ewe} + (1 | \text{subj}) + (1 | \text{trial})$.

effect	group	term	estimate	p.value
fixed	NA	(Intercept)	-0.0006	0.9625
fixed	NA	polAff.dem-rep	0.1290	0.0000
fixed	NA	ewe.TRUE-FALSE	-0.0139	0.4516
fixed	NA	polAff.dem-rep:ewe.TRUE-FALSE	0.1398	0.0002
ran_pars	subj	sd__(Intercept)	0.2855	NA
ran_pars	trial	sd__(Intercept)	0.0382	NA
ran_pars	Residual	sd__Observation	0.3232	NA

Power Simulation

In the following code, the simulations are calculated. We do not recommend executing this code junk as it takes several hours to run.

```
FUN_sim_2wayInt_pwr <- function(sim, ...){
  out <- FUN_sim_2wayInt(...)
  modelResults <- out$modelResults %>%
    mutate(sim = sim) %>%
    relocate(sim)
  return(modelResults)
}

# How many simulations should be run?
n_sims <- 1000

# What are the breaks for number of subjects we would like to calculate power for?
breaks_subj <- c(900, 950, 1000)

# What are the breaks for SESOI?
breaks_sesoi <- (0.4/3.5)*seq(1, 2, .25)

# What are the breaks for different error SDs?
breaks_sigma <- c((.29+.04), 2*(.29+.04))

res_2wayInt <- tibble()
for (s in seq_along(breaks_sigma)) {

  res_sesoi <- tibble()
```



```

for (sesoi in seq_along(breaks_sesoi)) {

  res_nSubj <- tibble()
  for (nSubj in seq_along(breaks_subj)) {

    # Give feedback regarding which model is simulated
    cat(paste0(
      "Simulation:\n",
      "  sigma = ", round(breaks_sigma[s], 4), "\n",
      "  sesoi = ", round(breaks_sesoi[sesoi], 4), "\n",
      "  nSubject = ", breaks_subj[nSubj], "\n"
    ))

    # Start timer
    cat(paste0("Start date time: ", lubridate::now(), "\n"))
    tic()

    # Loop over simulations
    pwr <- map_df(
      1:n_sims,
      FUN_sim_2wayInt_pwr,
      n_subj = breaks_subj[nSubj],
      beta_p_e_inx = breaks_sesoi[sesoi],
      sigma = breaks_sigma[s]
    )

    # Stop timer and calculate elapsed time
    elapsed_time <- toc(quiet = TRUE)
    elapsed_seconds <- elapsed_time$toc - elapsed_time$tic
    elapsed_minutes <- elapsed_seconds / 60
    cat(paste0("End date time: ", lubridate::now(), "\n"))
    cat("Elapsed time: ", elapsed_minutes, " minutes\n\n")

    # Add number of subjects to pwr
    pwr <- pwr %>%
      mutate(
        nSubjects = breaks_subj[nSubj],
        sesoi = breaks_sesoi[sesoi],
        sigma = breaks_sigma[s]
      )

    # Add results to the results table

```

```

    res_nSubj <- res_nSubj %>%
      rbind(pwr)
  }

  # Add results to the results table
  res_sesoi <- res_sesoi %>%
    rbind(res_nSubj)
}

# Add results to the results table
res_2wayInt <- res_2wayInt %>%
  rbind(res_sesoi)
}

res_2wayInt.summary <- res_2wayInt %>%
  filter(term == "polAff.dem-rep:ewe.TRUE-FALSE") %>%
  group_by(sigma, sesoi, nSubjects) %>%
  summarise(
    power = mean(p.value < 0.05),
    ci.lower = binom.confint(power*n_sims, n_sims, methods = "exact")$lower,
    ci.upper = binom.confint(power*n_sims, n_sims, methods = "exact")$upper,
    .groups = 'drop'
  ) %>%
  mutate(
    sigma_fact = factor(format(round(sigma, 4), nsmall = 4)),
    sigma_level = match(sigma_fact, levels(sigma_fact)),
    sesoi_fact = factor(format(round(sesoi, 4), nsmall = 4)),
    sesoi_level = match(sesoi_fact, levels(sesoi_fact))
  )

# Save results in a list object
time <- format(Sys.time(), "%Y%m%d_%H%M")
fileName <- paste0("res_2wayInt", "_", time, ".RDS")
saveRDS(
  list(
    res_2wayInt = res_2wayInt,
    res_2wayInt.summary = res_2wayInt.summary
  ),
  file = file.path("../powerSimulationsOutput", fileName)
)

```

We retrieve pre-run results:

```
# Load power simulation data
resList_2wayInt <- readRDS(file.path("../powerSimulationsOutput", "res_2wayInt_20240814_1052"))
resList_2wayInt.summary <- resList_2wayInt$res_2wayInt.summary

# Extract power values for some specific assumptions
chosenN <- 950
chosenSigma <- c("0.3300", "0.6600")
chosenSESOI <- c("0.1429", "0.1714")
powerValues <- resList_2wayInt.summary %>%
  filter(sigma_fact %in% chosenSigma) %>%
  filter(sesoi_fact %in% chosenSESOI) %>%
  filter(nSubjects %in% chosenN) %>%
  mutate(power_str = paste0(round(power*100, 2), "%")) %>%
  pull(power_str)

# Extract number of simulations
label_nSimulations <- resList_2wayInt$res_2wayInt$sim %>% n_distinct()

# Repeat breaks_sesoi
breaks_sesoi <- (0.4/3.5)*seq(1, 2, .25)
```

Figure 6 displays the distribution of estimated fixed effects across all simulations. The figure shows that the estimated fixed effects are close to the true ones provided as input in the data simulation function, validating that simulations worked as expected.

```

# Define some filters
filter_nSubjects <- 1000
filter_sesoi <- unique(resList_2wayInt$res_2wayInt$sesoi)[1]
filter_sigma <- unique(resList_2wayInt$res_2wayInt$sigma)[1]

# Prepare data for plot
fixedEstimates <- resList_2wayInt$res_2wayInt %>%
  mutate(
    group = ifelse(is.na(group), "", group),
    group_term = str_remove(str_c(group, term, sep = "_"), "^_")
  ) %>%
  filter(effect == "fixed") %>%
  filter(nSubjects == filter_nSubjects) %>%
  filter(sesoi == filter_sesoi) %>%
  filter(sigma == filter_sigma)
fixedEstimates_medians <- fixedEstimates %>%
  group_by(group_term) %>%
  summarise(
    median = median(estimate, na.rm = TRUE),
    median_rounded = format(round(median, 4), nsmall = 4, scientific = FALSE),
    .groups = 'drop'
  )

# Create plot
p.checkSims.2wayInt <- fixedEstimates %>%
  ggplot(aes(x = estimate, y = group_term)) +
  ggdist::stat_halfeye() +
  ggrepel::geom_label_repel(
    data = fixedEstimates_medians,
    mapping = aes(x = median, y = group_term, label = median_rounded),
    box.padding = .5
  ) +
  labs(
    title = "Distribution of Estimated Fixed Effects",
    subtitle = paste0(
      "SESOI = ", round(filter_sesoi, 4), ", ",
      " = ", round(filter_sigma, 4), ", ",
      "N = ", filter_nSubjects, ", ",
      "Number of Simulations = ", label_nSimulations
    ),
    x = "Estimate",
    y = "Term"
  ) +
  theme_bw()

print(p.checkSims.2wayInt)

```

28

Distribution of Estimated Fixed Effects
SESOI = 0.1143, . = 0.33, N = 1000, Number of

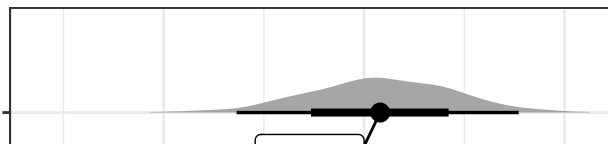


Figure 7 shows results of our effect-size sensitivity analyses. We plot statistical power (y-axis) for different effect sizes (x-axis), taking into account different assumptions for the error SD (color) and sample size (panel). Regarding the latter, we report results not only for the full sample size we aim for ($N = 1000$), but also for sample sizes taking into account different participant exclusion-rates due to exclusion criteria defined in the Registered Report.

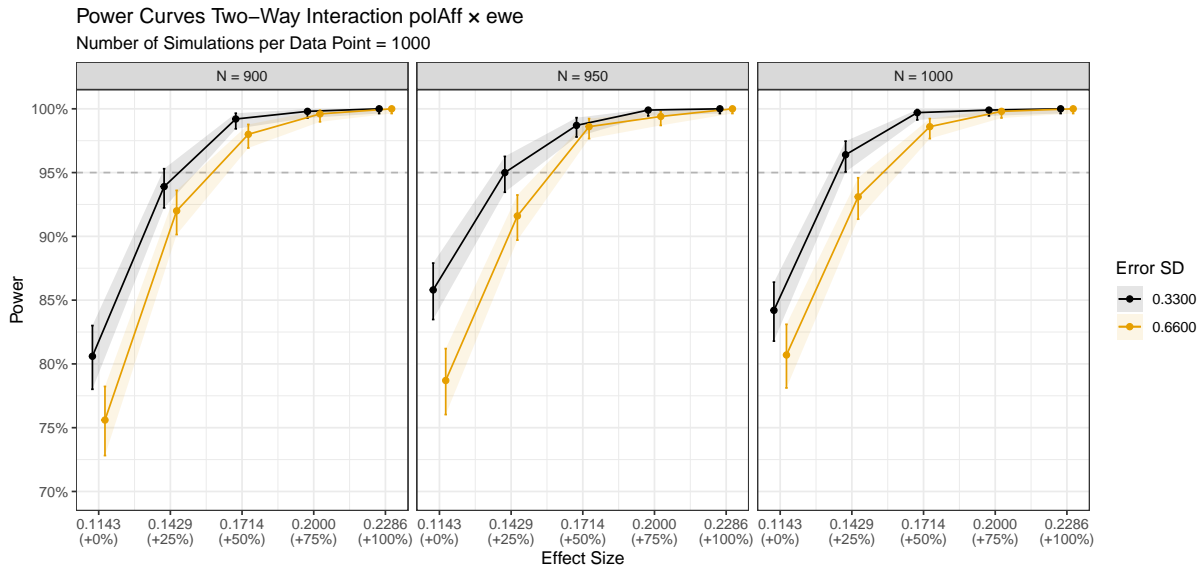


Figure 7: Power curves for the two-way interaction polAff \times ewe. Points represent simulated power surrounded by a 95%-CI based on 1000 simulations with $\alpha = 0.05$. Note that, in contrast to Figure 4, the x-axis represents different effect sizes, starting from the defined SESOI, while the panels represent different sample sizes, taking into account participant exclusion-rates of 10% ($N = 900$), 5% ($N = 950$), and 0% ($N = 1000$). Note that estimates are displayed with a slight shift along the x-axis to reduce overlap.

```
# Get power values of interest
chosenN <- 950
chosenSigma <- c("0.3300", "0.6600")
chosenSESOI <- c("0.1429", "0.1714")
powerValues <- resList_2wayInt.summary %>%
  filter(sigma_fact %in% chosenSigma) %>%
  filter(sesoi_fact %in% chosenSESOI) %>%
  filter(nSubjects %in% chosenN)

# Get lower CI for power values for more liberal and more conservative sigma assumptions
powerValues_sigmaLib <- powerValues %>%
  filter(sigma_fact == "0.3300")
```

```

powerValues_sigmaCons <- powerValues %>%
  filter(sigma_fact == "0.6600")

# Interpolate the effect sizes at which we achieve 95% power
eff_sigmaLib <- approx(powerValues_sigmaLib$ci.lower, powerValues_sigmaLib$sesoi, xout = 0.9)
eff_sigmaCons <- approx(powerValues_sigmaCons$ci.lower, powerValues_sigmaCons$sesoi, xout = 0.9)
# Round results for display in text
eff_sigmaLib_txt <- format(round(eff_sigmaLib, 4), nsmall = 4)
eff_sigmaCons_txt <- format(round(eff_sigmaCons, 4), nsmall = 4)

```

To assess the smallest effect size that can be detected with 95% statistical power, we inspect the lower bounds of the 95%-CI power estimates in Figure 7. Specifically, we focus on the power simulation results for $N = 950$, which takes into account a participant exclusion-rate of 5%. There, we interpolate between the two point estimates that lie just below and above the 95% power line, i.e., between the power estimates for effect sizes 0.1429 and 0.1714. Assuming an error SD of 0.3300, we achieve 95% statistical power to detect a two-way interaction effect of at least 0.1530. For a more conservative error SD of 0.6600, this smallest detectable effect size is only marginally higher (0.1619).

Three-Way Interaction Effect

As argued in the Registered Report, we hypothesize:

(H₃): *The moderating role of extreme weather exposure is greater among individuals who more strongly attribute extreme weather events to climate change. In other words, there is a positive three-way interaction between political affiliation, exposure to extreme weather events, and their attribution to climate change.*

```

# Smallest Effect Size Of Interest (SESOI)
SESOI <- 0.4/3.5

# Betas
beta_p <- SESOI
beta_e <- 0
beta_s <- 0
beta_p_e_inx <- SESOI
beta_p_s_inx <- 0
beta_e_s_inx <- 0
beta_p_e_s_inx <- SESOI

# Predicted "true" effects

```

```

polAff_ewe_subjAttr_trueEffects <- expand_grid(
  polAff = factor(c("rep", "dem"), levels = c("rep", "dem")),
  ewe = factor(c(FALSE, TRUE), levels = c(FALSE, TRUE)),
  subjAttr = factor(c("-SD", "M", "+SD"), levels = c("-SD", "M", "+SD"))
) %>%
  add_contrast("polAff", contrast = "anova", colnames = "X_p") %>%
  add_contrast("ewe", contrast = "anova", colnames = "X_e") %>%
  add_contrast("subjAttr", contrast = "anova", colnames = c("X_s_1", "X_s_2")) %>%
  mutate(
    trueDeltaDuration =
      0 + # intercept
      X_p * beta_p + # main effect polAff
      X_e * beta_e + # main effect ewe
      X_s_1 * beta_s + # main effect subjAttr, dummy variable for M
      X_s_2 * (2 * beta_s) + # main effect subjAttr, dummy variable for +SD
      X_p * X_e * beta_p_e_inx + # 2-way interaction polAff * ewe
      X_p * X_s_1 * beta_p_s_inx + # 2-way interaction polAff * subjAttr, dummy variable for M
      X_p * X_s_2 * (2 * beta_p_s_inx) + # 2-way interaction polAff * subjAttr, dummy variable for +SD
      X_e * X_s_1 * beta_e_s_inx + # 2-way interaction ewe * subjAttr, dummy variable for M
      X_e * X_s_2 * (2 * beta_e_s_inx) + # 2-way interaction ewe * subjAttr, dummy variable for +SD
      X_p * X_e * X_s_1 * beta_p_e_s_inx + # 3-way interaction polAff*ewe*subjAttr, dummy variable for M
      X_p * X_e * X_s_2 * (2 * beta_p_e_s_inx) # 3-way interaction polAff*ewe*subjAttr, dummy variable for +SD
  )

```

SESOI for Three-way Interaction

In the sections above, we derived the SESOI used for the main effect sample-size determination analysis and for the two-way interaction effect-size sensitivity analysis for the binary variables *polAff* (dem vs. rep) and *ewe* (TRUE vs. FALSE). *subjAttr*, however, is a continuous variable that ranges from 1 to 5. Fortunately, in finding a theoretically sound SESOI for the three-way interaction, the same considerations apply as for the main effect and two-way interaction before. We just need to translate these considerations into the continuous metric of *subjAttr*.

We start by noticing that the complete fixed three-way interaction $\text{polAff} \times \text{ewe} \times \text{subjAttr}$ is modeled as:

$$\begin{aligned}
 \Delta \text{Duration} = & \beta_0 + \\
 & \beta_1 \cdot \text{polAff} + \beta_2 \cdot \text{ewe} + \beta_3 \cdot \text{subjAttr} + \\
 & \beta_4 \cdot (\text{polAff} \times \text{ewe}) + \beta_5 \cdot (\text{polAff} \times \text{subjAttr}) + \beta_6 \cdot (\text{ewe} \times \text{subjAttr}) + \\
 & \beta_7 \cdot (\text{polAff} \times \text{ewe} \times \text{subjAttr})
 \end{aligned}$$

By rearranging terms, one can show that the two-way interaction $\text{polAff} \times \text{ewe}$ is given by:

$$\text{Two-way Interaction}_{\text{polAff} \times \text{ewe}} = \beta_4 + \beta_7 \cdot \text{subjAttr}$$

Now, let's calculate this two-way interaction for two individuals who differ in their level of subjective attribution of extreme weather events to climate change. First, an individual who has an average score on subjAttr will show the following two-way interaction effect, with μ_{subjAttr} being the sample average of the variable subjAttr :

$$\text{Effect}_{\text{Avg}} = \beta_4 + \beta_7 \cdot \mu_{\text{subjAttr}}$$

Second, we define an individual with a low score on subjAttr as one that shows a subjective attribution of one SD bellow the average. This individual will show the following two-way interaction effect, with σ_{subjAttr} being the SD of subjAttr :

$$\text{Effect}_{\text{Low}} = \beta_4 + \beta_7 \cdot (\mu_{\text{subjAttr}} - \sigma_{\text{subjAttr}})$$

The difference in the two-way interaction effect $\text{polAff} \times \text{ewe}$ between these two individuals is given by:

$$\begin{aligned} \text{Effect}_{\text{Avg}} - \text{Effect}_{\text{Low}} &= \\ \beta_4 + \beta_7 \cdot \mu_{\text{subjAttr}} - (\beta_4 + \beta_7 \cdot (\mu_{\text{subjAttr}} - \sigma_{\text{subjAttr}})) &= \\ \beta_7 \cdot [\mu_{\text{subjAttr}} - (\mu_{\text{subjAttr}} - \sigma_{\text{subjAttr}})] &= \\ \beta_7 \cdot \sigma_{\text{subjAttr}} \end{aligned}$$

As outlined above in Section Section , we assume that the SESOI for the two-way interaction $\text{polAff} \times \text{ewe}$ is 0.1143 for an average individual (with respect to subjAttr). If the same two-way interaction $\text{polAff} \times \text{ewe}$ shrinks to zero for an individual low in subjAttr , we would consider this interaction effect difference as theoretically relevant (see Figure 8). These assumptions translate to:

$$\text{Effect}_{\text{Avg}} - \text{Effect}_{\text{Low}} = 0.1143 = \beta_7 \cdot \sigma_{\text{subjAttr}}$$

Division by σ_{subjAttr} gives us the SESOI for the three-way interaction in the suitable metric of subjAttr :

$$\text{SESOI}_{\text{polAff} \times \text{ewe} \times \text{subjAttr}} = \frac{0.1143}{\sigma_{\text{subjAttr}}}$$

We will assess subjective attribution of extreme weather events to climate change using the same questions, response options, and aggregation as Ogunbode et al. (2019). These authors reported $\mu_{subjAttr} = 3.67$ and $\sigma_{subjAttr} = 0.85$, resulting in:

$$SESOI_{polAff \times ewe \times subjAttr} = \frac{0.1143}{0.85} = 0.1345$$

Visualisation for the SESOI of polAff × ewe × subjAttr

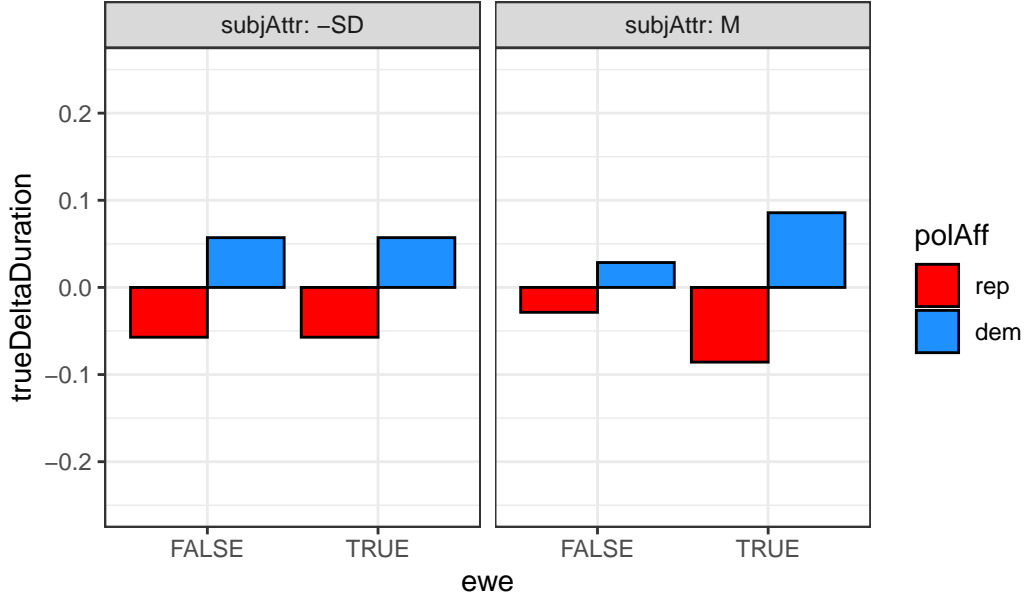


Figure 8: Assumed Δ Duration values for individuals scoring low (-SD) and average (M) on subjAttr. While individuals scoring average on subjAttr show a two-way interaction effect of polAff × ewe of 0.1143, this effect shrinks to zero for individuals scoring low on subjAttr. These individuals only show the predicted main effect of polAff (which is 0.1143).

Data Simulation Function

We finally define a function that simulates data for the three-way interaction effect of political affiliation with extreme weather exposure and attribution of extreme weather events to climate change on Δ Duration: `FUN_sim_3wayInt`. The function will simulate data according to the following model, expressed in lme4-lingo:

```
deltaDuration ~ polAff * ewe * subjAttr + (1|subj) + (1|trial)
```

The function `FUN_sim_3wayInt` takes, among others, the following important arguments (in addition to the arguments discussed for `FUN_sim_2wayInt`):

- **s_mean: Sample mean of subjective attribution.** Based on results reported by Ogunbode et al. (2019), we set this value to 3.67.
- **s_sd: Sample SD of subjective attribution.** Based on results reported by Ogunbode et al. (2019), we set this value to 0.85.
- **beta_s: Fixed main effect of subjective attribution.** As we are interested in the moderating role of subjective attribution of extreme weather events to climate change, we set this main effect to zero.
- **beta_p_s_inx Fixed two-way interaction effect of political affiliation and subjective attribution.** In order to accurately model a three-way interaction, one needs to include all two-way interactions in statistical models. Therefore, we include this two-way interaction, but we assume it to be zero.
- **beta_e_s_inx Fixed two-way interaction effect of extreme weather exposure and subjective attribution.** As before, we include this interaction for accurately modelling the three-way interaction of interest, but we assume this two-way interaction to be zero.
- **beta_p_e_s_inx Fixed three-way interaction effect of political affiliation, extreme weather exposure, and subjective attribution.** We set this initial value to the SESOI derived above, i.e. 0.1345. We investigate how changing this effect size impacts statistical power, as we are conducting effect-size sensitivity analyses for interaction effects.

The function is defined below:

```
# define data simulation function
FUN_sim_3wayInt <- function(
  n_subj          = 1000, # number of subjects
  n_subj_prop_p   = c(.5, .5), # proportion of republican and democrat subjects
  n_subj_prop_e   = c(.5, .5), # proportion of subjects without and with extreme weather exposure
  n_subj_prop_s   = c(.5, .5), # proportion of subjects with low and high subjective attribution
  n_trial         = 25, # number of trials
  s_mean          = 3.67, # mean of subjAttr, see Ogunbode et al. (2019)
  s_sd            = 0.85, # sd of subjAttr, see Ogunbode et al. (2019)
  beta_0          = 0, # intercept (grand mean) for deltaDuration
  beta_p          = 0.4/3.5, # main effect of political affiliation (polAff)
  beta_e          = 0, # main effect of extreme weather exposure (ewe)
  beta_s          = 0, # main effect of subjective attribution of ewe to climate change
  beta_p_e_inx    = 0.4/3.5, # two-way interaction effect of polAff and ewe
  beta_p_s_inx    = 0, # two-way interaction effect of polAff and subjAttr
  beta_e_s_inx    = 0, # two-way interaction effect of ewe and subjAttr
  beta_p_e_s_inx  = (0.4/3.5)/0.85, # three-way interaction effect of polAff, ewe, and subjAttr)
```

```

subj_0      = .29, # by-subject random intercept sd for dt carbon
trial_0     = .04, # by-trial random intercept sd
sigma       = 1*(.29+.04), # residual (error) sd

truncNums   = TRUE, # should impossible numbers be truncated?
setSeed     = NULL  # seed number to achieve reproducible results. Set to
) {

# set seed to achieve reproducible results for demonstration purposes
set.seed(setSeed)

# simulate data for dwell time on carbon information
dataSim <-
  # add random factor subject
  add_random(subj = n_subj) %>%
  # add random factor trial
  add_random(trial = n_trial) %>%
  # add between-subject factor political affiliation (with anova contrast)
  add_between("subj", polAff = c("rep", "dem"), .prob = n_subj_prop_p*n_subj, .shuffle = TRUE) %>%
  add_contrast("polAff", colnames = "X_p", contrast = "anova") %>%
  # add between-subject factor extreme weather exposure (with anova contrast)
  add_between("subj", ewe = c(FALSE, TRUE), .prob = n_subj_prop_e*n_subj, .shuffle = TRUE) %>%
  add_contrast("ewe", colnames = "X_e", contrast = "anova") %>%
  # add between-subject variable subjective attribution of EWE to climate change
  mutate(
    subjAttr = rep(rnorm(n = n_subj, mean = s_mean, sd = s_sd), each = n_trial),
    subjAttr_c = scale(subjAttr, center = TRUE, scale = FALSE)[,1]
  ) %>%
  # add by-subject random intercept
  add_ranef("subj", S_0 = subj_0) %>%
  # add by-trial random intercept
  add_ranef("trial", T_0 = trial_0) %>%
  # add error term
  add_ranef(e_st = sigma) %>%
  # add response values
  mutate(
    # add together fixed and random effects for each effect
    B_0 = beta_0 + S_0 + T_0,
    B_p = beta_p,
    B_e = beta_e,
    B_s = beta_s,
    B_p_e_inx = beta_p_e_inx,

```

```

    B_p_s_inx = beta_p_s_inx,
    B_e_s_inx = beta_e_s_inx,
    B_p_e_s_inx = beta_p_e_s_inx,
    # calculate dv by adding each effect term multiplied by the relevant
    # effect-coded factors and adding the error term
    deltaDuration =
      B_0 + e_st +
      (X_p * B_p) +
      (X_e * B_e) +
      (subjAttr_c * B_s) +
      (X_p * X_e * B_p_e_inx) +
      (X_p * subjAttr_c * B_p_s_inx) +
      (X_e * subjAttr_c * B_e_s_inx) +
      (X_p * X_e * subjAttr_c * B_p_e_s_inx)
  )

# unset seed
set.seed(NULL)

# truncate impossible deltaDurations
if(truncNums) {
  dataSim <- dataSim %>%
    mutate(deltaDuration = if_else(deltaDuration < -1, -1,
      if_else(deltaDuration > 1, 1, deltaDuration)))
}

# run a linear mixed effects model and check summary
mod <- lmer(
  deltaDuration ~ polAff*ewe*subjAttr_c + (1 | subj) + (1 | trial),
  data = dataSim
)
mod.sum <- summary(mod)

# get results in tidy format
mod.broom <- broom.mixed::tidy(mod)

return(list(
  dataSim = dataSim,
  modelLmer = mod,
  modelResults = mod.broom
))

```

```
}
```

We call the function once and extract the results of this single simulation:

```
# Note to myself: Consider setting beta_p = 0.4/3.5, beta_p_e_inx = 0.4/3.5,
# and beta_p_e_s_inx to 2 * 0.4/3.5/(2*.85).
# This way, the interaction effect polAff:ewe is 0.4/3.5 for individuals
# with an average subjAttr (subjAttr_c = 0). For individuals with
# subjAttr = mean - SD, polAff:ewe is 0. For individuals with subjAttr = mean + SD,
# polAff:ewe is 2 * 0.4/3.5.

out <- FUN_sim_3wayInt(
  n_subj          = 1000, # number of subjects
  n_subj_prop_p   = c(.5, .5), # proportion of republican and democrat subjects
  n_subj_prop_e   = c(.5, .5), # proportion of subjects without and with extreme weather
  n_subj_prop_s   = c(.5, .5), # proportion of subjects with low and high subjective attribution
  n_trial         = 25, # number of trials
  s_mean          = 3.67, # mean of subjAttr, see Ogunbode et al. (2019)
  s_sd            = 0.85, # sd of subjAttr, see Ogunbode et al. (2019)
  beta_0          = 0, # intercept (grand mean) for deltaDuration
  beta_p          = 0.4/3.5, # main effect of political affiliation (polAff)
  beta_e          = 0, # main effect of extreme weather exposure (ewe)
  beta_s          = 0, # main effect of subjective attribution of ewe to climate change
  beta_p_e_inx    = 0.4/3.5, # two-way interaction effect of polAff and ewe
  beta_p_s_inx    = 0, # two-way interaction effect of polAff and subjAttr
  beta_e_s_inx    = 0, # two-way interaction effect of ewe and subjAttr
  beta_p_e_s_inx  = (0.4/3.5)/0.85, # three-way interaction effect of polAff, ewe, and subjAttr
  subj_0          = .29, # by-subject random intercept sd for dt carbon
  trial_0         = .04, # by-trial random intercept sd
  sigma           = 1*(.29+.04), # residual (error) sd

  truncNums       = TRUE, # should impossible numbers be truncated?
  setSeed         = 123 # seed number to achieve reproducible results. Set to 123

)

# Get results table
resultsTable <- out$modelResults %>%
  select(-c(std.error, statistic, df)) %>%
  mutate(across(where(is_double), ~ round(.x, 4))) %>%
  knitr::kable()
formulaUsedForFit <- paste(as.character(formula(out$modelLmer))[c(2,1,3)], collapse = " ")
```

```

# Create predictions plot

# refit model to display subjAttr levels in original metric (not mean centered)
m <- lmer(
  deltaDuration ~ polAff*ewe*subjAttr + (1 | subj) + (1 | trial),
  data = out$dataSim
)
# define spotlights for spotlight analysis
spotlights <- c(3.67 - 0.85, 3.67, 3.67 + 0.85)

# create plot showing predictions
p.demo.3wayInt.pred <- predict_response(m, terms = c("ewe", "polAff", "subjAttr[spotlights]"))
p.demo.3wayInt <- p.demo.3wayInt.pred %>%
  plot(colors = c("red", "dodgerblue")) +
  coord_cartesian(ylim = c(-.25, .25)) +
  theme_bw()

```

Figure 9 visualizes predictions based on this single simulation and Table 3 summarizes the statistical results of fitting the actual model used in data generation to the simulated data.

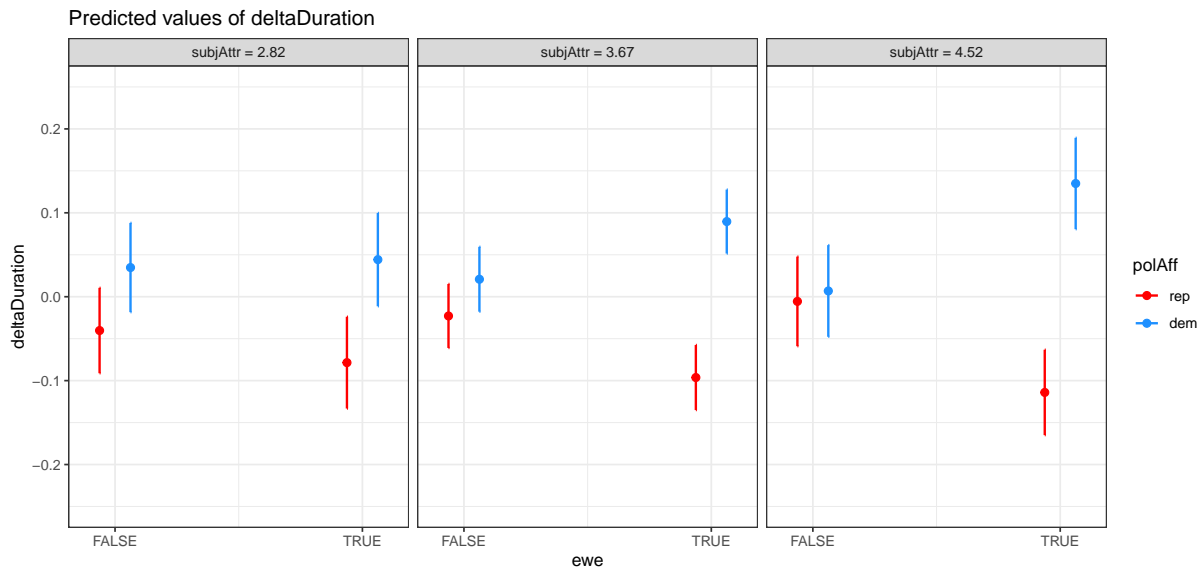


Figure 9: Visual representation of results of one simulation created using FUN_sim_3wayInt. Points indicate the predicted means surrounded by 95% Confidence Intervals. Panels indicate predictions for different values of subjAttr (Mean - SD, Mean, and Mean + SD).

Table 3: Statistical results of one simulation created using `FUN_sim_3wayInt`. Data was fit using $\text{deltaDuration} \sim \text{polAff} * \text{ewe} * \text{subjAttr_c} + (1 | \text{subj}) + (1 | \text{trial})$. Note that `subjAttr` is mean centered to ease interpretation of lower-level interactions and main effects.

effect	group	term	estimate	p.value
fixed	NA	(Intercept)	-0.0022	0.8478
fixed	NA	polAff.dem-rep	0.1149	0.0000
fixed	NA	ewe.TRUE-FALSE	-0.0023	0.9031
fixed	NA	subjAttr_c	0.0091	0.4141
fixed	NA	polAff.dem-rep:ewe.TRUE-FALSE	0.1426	0.0001
fixed	NA	polAff.dem-rep:subjAttr_c	0.0187	0.4025
fixed	NA	ewe.TRUE-FALSE:subjAttr_c	0.0142	0.5239
fixed	NA	polAff.dem-rep:ewe.TRUE-FALSE:subjAttr_c	0.1111	0.0130
ran_pars	subj	sd__(Intercept)	0.2849	NA
ran_pars	trial	sd__(Intercept)	0.0323	NA
ran_pars	Residual	sd__Observation	0.3240	NA

Power Simulation

In the following code, the simulations are calculated. We do not recommend executing this code junk as it takes several hours to run.

```
FUN_sim_3wayInt_pwr <- function(sim, ...){
  out <- FUN_sim_3wayInt(...)
  modelResults <- out$modelResults %>%
    mutate(sim = sim) %>%
    relocate(sim)
  return(modelResults)
}

# How many simulations should be run?
n_sims <- 1000

# What are the breaks for number of subjects we would like to calculate power for?
breaks_subj <- c(900, 950, 1000)

# What are the breaks for SESOI?
breaks_sesoi <- (0.4/3.5)/0.85 * seq(1, 2, .25)
```

```

# What are the breaks for different error SDs?
breaks_sigma <- c((.29+.04), 2*(.29+.04))

res_3wayInt <- tibble()
for (s in seq_along(breaks_sigma)) {

  res_sesoi <- tibble()
  for (sesoi in seq_along(breaks_sesoi)) {

    res_nSubj <- tibble()
    for (nSubj in seq_along(breaks_subj)) {

      # Give feedback regarding which model is simulated
      cat(paste0(
        "Simulation:\n",
        "  sigma = ", round(breaks_sigma[s], 4), "\n",
        "  sesoi = ", round(breaks_sesoi[sesoi], 4), "\n",
        "  nSubject = ", breaks_subj[nSubj], "\n"
      ))

      # Start timer
      cat(paste0("Start date time: ", lubridate::now(), "\n"))
      tic()

      # Loop over simulations
      pwr <- map_df(
        1:n_sims,
        FUN_sim_3wayInt_pwr,
        n_subj = breaks_subj[nSubj],
        beta_p_e_s_inx = breaks_sesoi[sesoi],
        sigma = breaks_sigma[s]
      )

      # Stop timer and calculate elapsed time
      elapsed_time <- toc(quiet = TRUE)
      elapsed_seconds <- elapsed_time$toc - elapsed_time$tic
      elapsed_minutes <- elapsed_seconds / 60
      cat(paste0("End date time: ", lubridate::now(), "\n"))
      cat("Elapsed time: ", elapsed_minutes, " minutes\n\n")

      # Add number of subjects to pwr
      pwr <- pwr %>%

```



```

      mutate(
        nSubjects = breaks_subj[nSubj],
        sesoi = breaks_sesoi[sesoi],
        sigma = breaks_sigma[s]
      )

      # Add results to the results table
      res_nSubj <- res_nSubj %>%
        rbind(pwr)
    }

    # Add results to the results table
    res_sesoi <- res_sesoi %>%
      rbind(res_nSubj)
  }

  # Add results to the results table
  res_3wayInt <- res_3wayInt %>%
    rbind(res_sesoi)
}

res_3wayInt.summary <- res_3wayInt %>%
  filter(term == "polAff.dem-rep:ewe.TRUE-FALSE:subjAttr_c") %>%
  group_by(sigma, sesoi, nSubjects) %>%
  summarise(
    power = mean(p.value < 0.05),
    ci.lower = binom.confint(power*n_sims, n_sims, methods = "exact")$lower,
    ci.upper = binom.confint(power*n_sims, n_sims, methods = "exact")$upper,
    .groups = 'drop'
  ) %>%
  mutate(
    sigma_fact = factor(format(round(sigma, 4), nsmall = 4)),
    sigma_level = match(sigma_fact, levels(sigma_fact)),
    sesoi_fact = factor(format(round(sesoi, 4), nsmall = 4)),
    sesoi_level = match(sesoi_fact, levels(sesoi_fact))
  )

# Save results in a list object
time <- format(Sys.time(), "%Y%m%d_%H%M")
fileName <- paste0("res_3wayInt", "_", time, ".RDS")
saveRDS(

```

```
list(
  res_3wayInt = res_3wayInt,
  res_3wayInt.summary = res_3wayInt.summary
),
file = file.path("../powerSimulationsOutput", fileName)
)
```

We retrieve pre-run results:

```
# Load power simulation data
resList_3wayInt <- readRDS(file.path("../powerSimulationsOutput", "res_3wayInt_20240814_1612
resList_3wayInt.summary <- resList_3wayInt$res_3wayInt.summary

# Extract power values for some specific effect sizes at N = 1000
powerValues <- resList_3wayInt.summary %>%
  filter(sigma_fact == "0.3300") %>%
  filter(sesoi_fact == "0.1345") %>%
  filter(nSubjects == 950) %>%
  mutate(power_str = paste0(round(power*100, 2), "%")) %>%
  pull(power_str)

# Extract number of simulations
label_nSimulations <- resList_3wayInt$res_3wayInt$sim %>% n_distinct()

# Repeat breaks_sesoi
breaks_sesoi <- (0.4/3.5)/(0.85) * seq(1, 2, .25)
```

Figure 10 displays the distribution of estimated fixed effects across all simulations. The figure shows that the estimated fixed effects are close to the true ones provided as input in the data simulation function, validating that simulations worked as expected.

Figure 11 shows results of our effect-size sensitivity analyses. We plot statistical power (y-axis) for different effect sizes (x-axis), taking into account different assumptions for the error SD (color) and sample size (panel). Regarding the latter, we report results not only for the full sample size we aim for ($N = 1000$), but also for sample sizes taking into account different participant exclusion-rates due to exclusion criteria defined in the Registered Report.

```
# Get power values of interest
chosenN <- 950
chosenSigma <- c("0.3300", "0.6600")
chosenSESOI <- c("0.1681", "0.2017")
powerValues <- resList_3wayInt.summary %>%
```

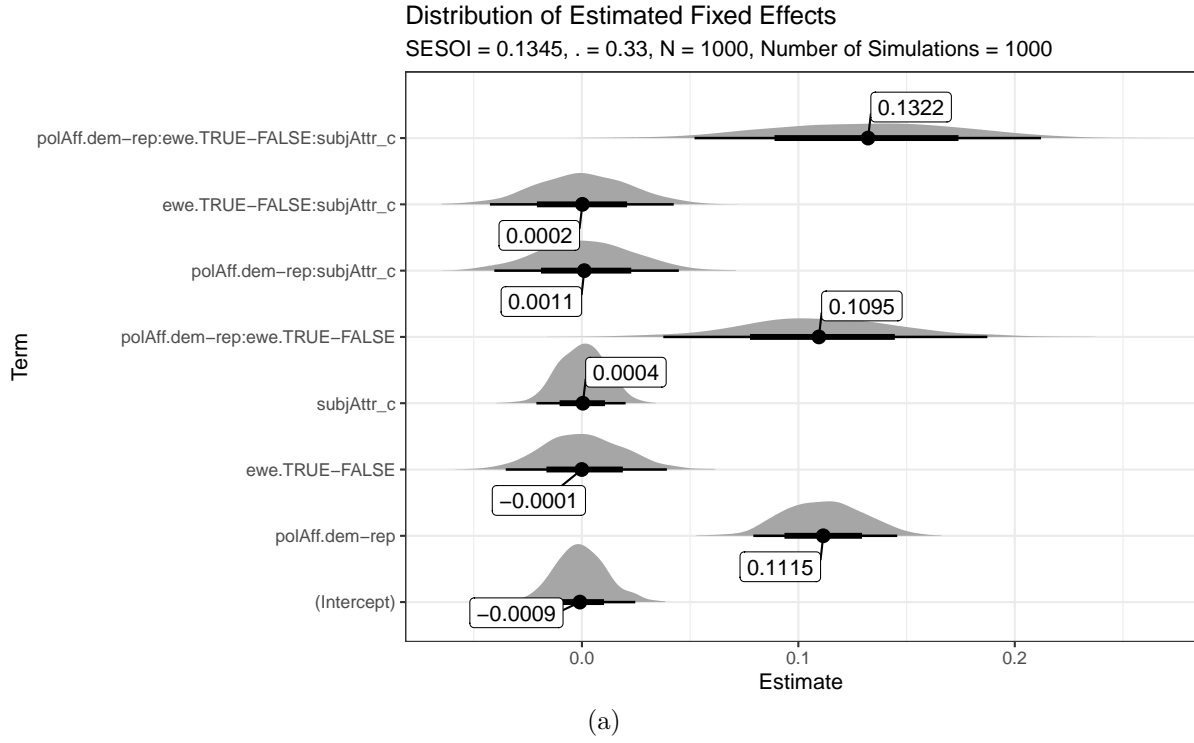


Figure 10: Distribution of estimated fixed effects resulting from 1000 simulations for the model $\text{deltaDuration} \sim \text{polAff} * \text{ewe} * \text{subjAttr_c} + (1 | \text{subj}) + (1 | \text{trial})$. Shaded area represent densities, annotated points indicate medians, and thick and thin lines represent 66% and 95% quantiles.

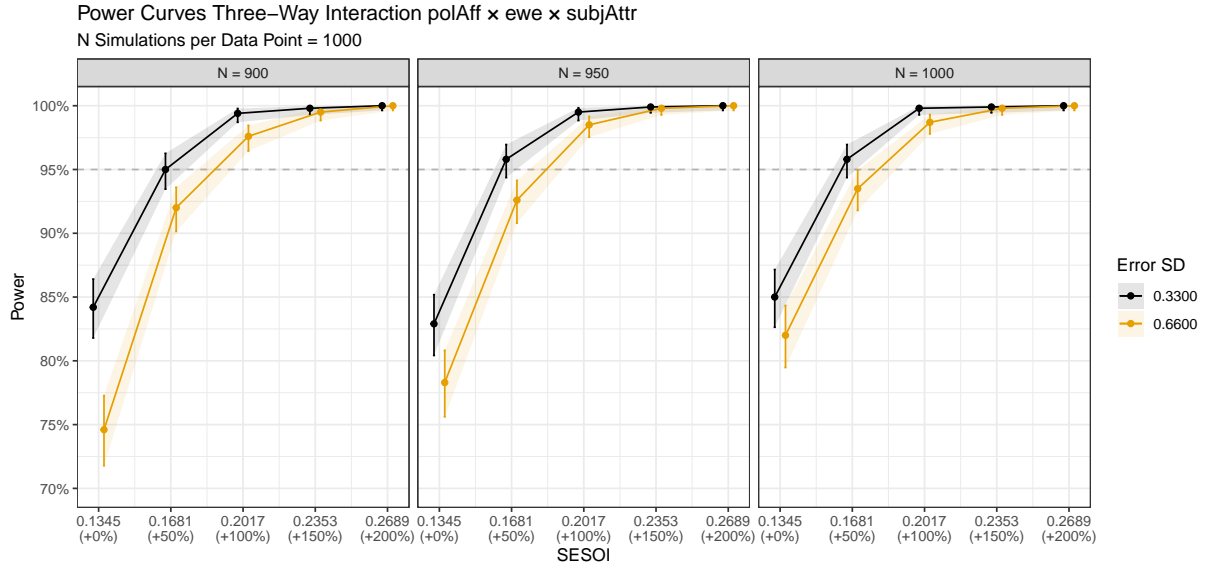


Figure 11: Power curves for the three-way interaction polAff \times ewe \times subjAttr. Points represent simulated power surrounded by a 95%-CI based on 1000 simulations with $\alpha = 0.05$. Note that, in contrast to Figure 4, the x-axis represents different effect sizes, starting from the defined SESOI, while the panels represent different sample sizes, taking into account participant exclusion-rates of 10% ($N = 900$), 5% ($N = 950$), and 0% ($N = 1000$). Note that estimates are displayed with a slight shift along the x-axis to reduce overlap.

```

filter(sigma_fact %in% chosenSigma) %>%
filter(sesoi_fact %in% chosenSES0I) %>%
filter(nSubjects %in% chosenN)

# Get lower CI for power values for more liberal and more conservative sigma assumptions
powerValues_sigmaLib <- powerValues %>%
  filter(sigma_fact == "0.3300")
powerValues_sigmaCons <- powerValues %>%
  filter(sigma_fact == "0.6600")

# Interpolate the effect sizes at which we achieve 95% power
eff_sigmaLib <- approx(powerValues_sigmaLib$ci.lower, powerValues_sigmaLib$sesoi, xout = 0.95)
eff_sigmaCons <- approx(powerValues_sigmaCons$ci.lower, powerValues_sigmaCons$sesoi, xout = 0.95)
# Round results for display in text
eff_sigmaLib_txt_3way <- format(round(eff_sigmaLib, 4), nsmall = 4)
eff_sigmaCons_txt_3way <- format(round(eff_sigmaCons, 4), nsmall = 4)

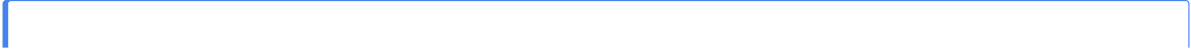
```

To assess the smallest effect size that can be detected with 95% statistical power, we inspect the lower bounds of the 95%-CI power estimates in Figure 11. Specifically, we focus on the power simulation results for $N = 950$, which takes into account a participant exclusion rate of 5%. There, we interpolate between the two point estimates that lie just below and above the 95% power line, i.e., between the power estimates for effect sizes 0.1681 and 0.2017. Assuming an error SD of 0.3300, we achieve 95% statistical power to detect a three-way interaction effect of at least 0.1728. For a more conservative error SD of 0.6600, this smallest detectable effect size is only marginally higher (0.1890).

Conclusion

Using power simulations with mixed-effects models for sample-size determination and effect-size sensitivity analyses, we show that with a final sample size of $N = 950$:

1. We will achieve $> 95\%$ statistical power to detect a main effect of political affiliation.
2. We will be able to detect a two-way interaction effect of political affiliation with extreme weather exposure of at least 0.1619 with 95% statistical power.
3. We will be able to detect a three-way interaction effect of political affiliation with extreme weather exposure and subjective attribution of extreme weather events to climate change of at least 0.1890 with 95% statistical power.



i Expand for Session Info

```
- Session info -----
setting value
version R version 4.4.0 (2024-04-24)
os      macOS Sonoma 14.4.1
system  aarch64, darwin20
ui      X11
language (EN)
collate en_US.UTF-8
ctype   en_US.UTF-8
tz      Europe/Zurich
date    2024-08-16
pandoc  3.1.11 @ /usr/local/bin/ (via rmarkdown)
quarto  1.5.45 @ /usr/local/bin/quarto
```

```
- Packages -----
package      * version  date (UTC) lib source
binom        * 1.1-1.1  2022-05-02 [1] CRAN (R 4.4.0)
dplyr        * 1.1.4    2023-11-17 [1] CRAN (R 4.4.0)
faux         * 1.2.1    2023-04-20 [1] CRAN (R 4.4.0)
forcats      * 1.0.0    2023-01-29 [1] CRAN (R 4.4.0)
ggdist       * 3.3.2    2024-03-05 [1] CRAN (R 4.4.0)
ggeffects    * 1.7.0    2024-06-20 [1] CRAN (R 4.4.0)
ggplot2      * 3.5.1    2024-04-23 [1] CRAN (R 4.4.0)
ggpubr       * 0.6.0    2023-02-10 [1] CRAN (R 4.4.0)
ggrepel      * 0.9.5    2024-01-10 [1] CRAN (R 4.4.0)
ggthemes     * 5.1.0    2024-02-10 [1] CRAN (R 4.4.0)
lme4         * 1.1-35.5 2024-07-03 [1] CRAN (R 4.4.0)
lmerTest     * 3.1-3     2020-10-23 [1] CRAN (R 4.4.0)
lubridate    * 1.9.3    2023-09-27 [1] CRAN (R 4.4.0)
Matrix       * 1.7-0     2024-03-22 [1] CRAN (R 4.4.0)
purrr        * 1.0.2    2023-08-10 [1] CRAN (R 4.4.0)
readr        * 2.1.5    2024-01-10 [1] CRAN (R 4.4.0)
sessioninfo  * 1.2.2    2021-12-06 [1] CRAN (R 4.4.0)
stringr      * 1.5.1    2023-11-14 [1] CRAN (R 4.4.0)
tibble       * 3.2.1    2023-03-20 [1] CRAN (R 4.4.0)
tictoc       * 1.2.1    2024-03-18 [1] CRAN (R 4.4.0)
tidyr        * 1.3.1    2024-01-24 [1] CRAN (R 4.4.0)
tidyverse    * 2.0.0    2023-02-22 [1] CRAN (R 4.4.0)
```

[1] /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library

- Berger, Sebastian, and Annika M Wyss. 2021. "Measuring Pro-Environmental Behavior Using the Carbon Emission Task." *Journal of Environmental Psychology* 75: 101613. <https://doi.org/https://doi.org/10.1016/j.jenvp.2021.101613>.
- Giner-Sorolla, Roger, Amanda K. Montoya, Alan Reifman, Tom Carpenter, Neil A. Lewis, Christopher L. Aberson, Dries H. Bostyn, et al. 2024. "Power to Detect What? Considerations for Planning and Evaluating Sample Size." *Personality and Social Psychology Review*, February, 10888683241228328. <https://doi.org/10.1177/10888683241228328>.
- Ogunbode, Charles A., Christina Demski, Stuart B. Capstick, and Robert G. Sposato. 2019. "Attribution Matters: Revisiting the Link Between Extreme Weather Experience and Climate Change Mitigation Responses." *Global Environmental Change* 54 (January): 31–39. <https://doi.org/10.1016/j.gloenvcha.2018.11.005>.
- Reeck, Crystal, Daniel Wall, and Eric J. Johnson. 2017. "Search Predicts and Changes Patience in Intertemporal Choice." *Proceedings of the National Academy of Sciences* 114 (45): 11890–95. <https://doi.org/10.1073/pnas.1707040114>.
- Salthouse, Timothy A. 2000. "Aging and Measures of Processing Speed." *Biological Psychology* 54 (1): 35–54. [https://doi.org/10.1016/S0301-0511\(00\)00052-1](https://doi.org/10.1016/S0301-0511(00)00052-1).
- Willemsen, Martijn C., and Eric J. Johnson. 2019. "(Re)visiting the Decision Factory: Ob-sweriving Cognition with MouselabWEB." In, edited by Michael Schulte-Mecklenbeck, Anton Kuehberger, and Joseph G. Johnson, 2nd ed., 76–95. New York: Routledge. <https://doi.org/10.4324/9781315160559>.