MeToo Movement and Perceptions of Sexual Assault: Annotated analysis code

Anna E. Jaffe, Ian Cero, David DiLillo

Contents

Introduction

R Version:

x86_64-redhat-linux-gnu, x86_64, linux-gnu, x86_64, linux-gnu, , 3, 4.1, 2017, 06, 30, 72865, R, R version 3.4.1 (2017-06-30), Single Candle

Run date: 2019-11-17 00:19:24

Initial Seed

```
set.seed(3107941)
```

What follows is annotated code for *The #MeToo Movement and Perceptions of Sexual Assault: College Students' Recognition of Sexual Assault Experiences Over Time*. Its purpose is to allow readers to reproduce our analyses as exactly as possible.[1] You can read more about `.Rmd` files at *rmarkdown.rstudio.com*.

This file contains all of the data import and model estimation code and output.

Note, our code was originally written in modules to keep each `.Rmd` document a manageable size. However, for portability, all of those documents have been knitted together in one large document here. To produce this document, the original `.Rmd` files (list below), were all run in the following order:

Section 1: Importation and model estimation

1. `data_wrangling.Rmd`
2. `analysis_functions.Rmd`
3. `multiple_imputation.Rmd`
4. `models_and_priors.Rmd`
5. `model_comparison.Rmd`
6. `mcmc_chain_convergence.Rmd`

Section 2: Processing and summarizing results

1. `aws_results_summary.Rmd`

[1] Note there are always limits to reproducibility. For example, Monte Carlo simulation is used in some of our analyses. Depending on the random seed and randomization strategy you use, you may get different results for those analyses. However, the differences should be small.

The output (text, code, and results) of these files is given in their corresponding sections below. None of the code blocks used to produce our results in this document are hidden. The code you see is all of the code that was run, from start to finish.

Data Wrangling

Import participant data

Data were originally stored in SPSS format, so were imported with the `memisc` package. The resulting dataframe included n = 2,616 cases.

```
library(tidyverse)

spss_dataset <- memisc::spss.system.file('data/MeTooData_updatedAug2019.sav')

participant_df <- spss_dataset %>%
  memisc::as.data.set() %>%
  as.data.frame()

nrow(participant_df)
```

```
## [1] 2616
```

Coding date-related variables

We continued by scoring date-related variables. Note, there are odd parameters for the `as.Date()` function that have been commented out (lines 37-38). These were intentional, as there was a previous problem converting SPSS dates to POSIXct format. They appear to have been fixed by the new version of `memisc`, so we commented those lines out and simply use the `ymd_hsm()` function from `lubridate`. However, we leave the old lines available in the code block in case attempts to reproduce our work require them for some unforseen reason.

We also denoted a dichotomous pre-post MeToo timing variable (i.e., before/after 2017-10-15).

NOTE: In order to get the numerical optimization algorithms to behave appropriately, we produced a `week_num` variable by taking the number of days between a participant's study day and the minimum study day, then dividing by 7. This in turn reduces the magnitude of the resulting `week_num_SQ` variable. Our previous testing has shown that without this modification, MCMCpack is not able to compute marginal likelihoods effectively. Note that because we are dividing days by 7, a participants `week_num` variable can be a decimal (e.g., `week_num` = 13/7 = 1.85).

```
library(lubridate)

participant_df <- participant_df %>%
  mutate(
      # start_date = as.Date(startdate/86400, origin = '1582-10-14'),
      # end_date = as.Date(enddate/86400, origin = '1582-10-14'),
      start_date = ymd_hms(startdate),
      end_date = ymd_hms(enddate),
      hours_to_complete = difftime(end_date, start_date, units = 'hours'),
      start_day = as.Date(floor_date(start_date, unit = 'days')),
```

```
      study_week = as.Date(floor_date(start_day, unit = 'week')),
      # day_num = as.numeric(start_day - min(start_day)) # submission 1
      day_num = as.numeric(start_day - as.Date('2017-10-15')),# rr1
      week_num = day_num / 7,
      post_metoo = ifelse(start_day >= '2017-10-15', 1, 0))
```

Coding demographic variables

Once date-related variables are scored, we continued with demographics. Race is coded such that anyone who endorses being Latino, American Indian, Asian, Black, Pacific Islander, or "Other Race" is coded as "racial_minority = 1"; people who endorse being non-Hispanic White are coded 0; and people who are missing on all race questions are coded NA.

LGBTQ status is coded 1 (yes) for anyone who endorsed same sex partners, same sex attraction, or non cis-gender identity. It is coded 0 for anyone who answered no to all of those questions. People with missing scores for all of those questions are coded NA.

Gender is coded 0 = female, 1 = male (as non-cis-gender participants are already accounted for in the LGBTQ group).

```
participant_df$latino = as.numeric(participant_df$latino == 'Yes')
race_vars <- c('white', 'latino', 'amin', 'asian', 'black', 'pacisl', 'othrace')
missing_all_race <- apply(
  participant_df[, race_vars], 1, function(row) all(is.na(row)))
is_racial_minority <- apply(
  participant_df[, race_vars[2:7]], 1, function(row) any(row == 1, na.rm = T))

participant_df <- participant_df %>%
  mutate(
    has_race = !missing_all_race,
    racial_minority = as.numeric(has_race & is_racial_minority),
    male = gender == 'Male',
    male_partners = preference %in% c('Men', 'Both'),
    female_partners = preference %in% c('Women', 'Both'),
    non_hetero_identification = orientation != 'Heterosexual / Straight',
    lgbtq = (male & male_partners) | (!male & female_partners) |
      non_hetero_identification | (!gender %in% c('Male', 'Female')))
```

Coding sexual experience variables

Then, we scored substantive variables related to partipants' perception of their sexual experiences. Some of these variables were already coded in SPSS, but were renamed for convenience.

```
participant_df <- participant_df %>%
  mutate(
    vic_screener = msesyn,
    vic_worst_is_sa = as.numeric(mses_sa.1),
    vic_worst_prior_sex = msesperp1prior.1 %in% c('Yes', 'Unsure'),
    vic_worst_verbal = (as.numeric(mses_tactic1.1) + as.numeric(mses_tactic2.1) +
      as.numeric(mses_tactic3.1) + as.numeric(mses_tactic6.1))/4,
    vic_worst_incap = (as.numeric(mses_tactic4.1) + as.numeric(mses_tactic5.1))/2,
    vic_worst_force = (as.numeric(mses_tactic7.1) + as.numeric(mses_tactic8.1) +
```

```
      as.numeric(mses_tactic9.1))/3
    )
```

Interactions and variable reduction

Finally, we computed quadratic and interaction terms, saving the result to a new dataframe with only the variables we need. Also, participants who took longer than 24 hours to complete their survey were excluded.

```
df <- participant_df %>%
  filter(hours_to_complete <= 24) %>%
  mutate(
    week_num_SQ = week_num^2,
    post_metoo_week = post_metoo*week_num,
    post_metoo_week_SQ = post_metoo*week_num_SQ) %>%
  dplyr::select(
    day_num, post_metoo, study_week, week_num, week_num_SQ,
    post_metoo_week, post_metoo_week_SQ,
    male, lgbtq, racial_minority,
    vic_screener, vic_worst_is_sa, vic_worst_prior_sex, vic_worst_verbal,
    vic_worst_incap, vic_worst_force)

nrow(df)
```

```
## [1] 2566
```

Demographics

Total participants in the analysis

```
n <- participant_df %>%
  filter(hours_to_complete <= 24) %>%
  nrow()

n
```

```
## [1] 2566
```

Gender distribution

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  group_by(gender) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## Warning: Factor `gender` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
## # A tibble: 8 x 3
##   gender                     n     p
##   <fct>                  <int> <dbl>
## 1 Female                  1845 0.719
## 2 Male                     699 0.272
## 3 Transgender woman          3 0.001
## 4 Transgender man            3 0.001
## 5 Gender queer               8 0.003
## 6 Other - Please specify     2 0.001
## 7 Decline to state           3 0.001
## 8 <NA>                       3 0.001
```

```
levels(participant_df$gender_txt) #determine text response for "other"
```

```
## [1] ""             "genderfluid"  "Velociraptor"
```

Racial and ethnic distribution

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  dplyr::select(race_vars) %>%
  colSums(na.rm = T) %>%
  magrittr::divide_by(n) %>%
  round(3)
```

```
##   white  latino    amin   asian   black  pacisl othrace
##   0.825   0.093   0.019   0.099   0.054   0.003   0.040
```

```
table(participant_df$white)
```

```
##
##    1
## 2159
```

```
table(participant_df$latino)
```

```
##
##    0    1
## 2368  244
```

```
table(participant_df$amin)
```

```
##
##  1
## 49
```

```
table(participant_df$asian)
```

```
##
##   1
## 257
```

```r
table(participant_df$black)
```

```
##
##   1
## 141
```

```r
table(participant_df$pacisl)
```

```
##
## 1
## 8
```

```r
table(participant_df$othrace)
```

```
##
##   1
## 107
```

Racial/ethnic minority variable from above (FALSE = non-Hispanic White)

```r
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  group_by(racial_minority) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 2 x 3
##   racial_minority     n     p
##             <dbl> <int> <dbl>
## 1               0  1900  0.74
## 2               1   666  0.26
```

LGBTQ distribution

```r
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  group_by(orientation) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## Warning: Factor `orientation` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
## # A tibble: 6 x 3
##   orientation                 n     p
##   <fct>                   <int> <dbl>
## 1 Heterosexual / Straight  2368 0.923
```

```
## 2 Lesbian / Gay              40 0.016
## 3 Bisexual                  124 0.048
## 4 Something Else             12 0.005
## 5 Don't know                 19 0.007
## 6 <NA>                        3 0.001
```

LGBTQ variable from above (FALSE = heterosexual *and* cisgender)

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  group_by(lgbtq) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 2 x 3
##   lgbtq     n     p
##   <lgl> <int> <dbl>
## 1 FALSE  2331 0.908
## 2 TRUE    235 0.092
```

Age distribution

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  summarise(
    mean = mean(age, na.rm = T),
    sd = sd(age, na.rm = T),
    min = min(age, na.rm = T),
    max = max(age, na.rm = T),
    n = n(),
    n_missing = sum(is.na(age))) %>%
  round(2)
```

```
##    mean  sd min max    n n_missing
## 1 20.47 2.5  19  55 2566         3
```

Descriptives for study variables

Sexual assault screener

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes' | vic_screener == 'No') %>% # only consider those with data on screener
  group_by(vic_screener) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 2 x 3
##   vic_screener     n     p
```

```
##   <fct>        <int> <dbl>
## 1 No            1904 0.762
## 2 Yes            596 0.238
```

Sexual assault screener by gender

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes' | vic_screener == 'No') %>% # only consider those with data on screener
  group_by(gender, vic_screener) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(gender) %>% #other gender is specified as "genderfluid" in gender_txt
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 12 x 4
## # Groups:   gender [7]
##    gender                 vic_screener     n     p
##    <fct>                  <fct>        <int> <dbl>
##  1 Female                 No            1266 0.703
##  2 Female                 Yes            534 0.297
##  3 Male                   No             626 0.918
##  4 Male                   Yes             56 0.082
##  5 Transgender woman      No               3 1
##  6 Transgender man        No               2 0.667
##  7 Transgender man        Yes              1 0.333
##  8 Gender queer           No               4 0.5
##  9 Gender queer           Yes              4 0.5
## 10 Other - Please specify No               1 0.5
## 11 Other - Please specify Yes              1 0.5
## 12 Decline to state       No               2 1
```

Sexual assault screener by LGBTQ

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes' | vic_screener == 'No') %>% # only consider those with data on screener
  group_by(lgbtq, vic_screener) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(lgbtq) %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 4 x 4
## # Groups:   lgbtq [2]
##   lgbtq vic_screener     n     p
##   <lgl> <fct>        <int> <dbl>
## 1 FALSE No            1757 0.773
## 2 FALSE Yes            517 0.227
## 3 TRUE  No             147 0.65
## 4 TRUE  Yes             79 0.35
```

Sexual assault screener by Race/Ethnicity

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes' | vic_screener == 'No') %>% # only consider those with data on screener
  group_by(racial_minority, vic_screener) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(racial_minority) %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 4 x 4
## # Groups:   racial_minority [2]
##   racial_minority vic_screener     n     p
##             <dbl> <fct>        <int> <dbl>
## 1               0 No            1388 0.748
## 2               0 Yes            468 0.252
## 3               1 No             516 0.801
## 4               1 Yes            128 0.199
```

Sexual assault characteristics (only asked of participants with positive screens)

- Verbal coercion

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes') %>% # only given to people with positive screens
  summarise(
    mean = mean(vic_worst_verbal, na.rm = T),
    sd = sd(vic_worst_verbal, na.rm = T),
    min = min(vic_worst_verbal, na.rm = T),
    max = max(vic_worst_verbal, na.rm = T),
    n = n(),
    n_missing = sum(is.na(vic_worst_verbal))) %>%
  round(2)
```

```
##   mean   sd min max   n n_missing
## 1 1.78 0.76   1   5 596        25
```

- Incapacitation due to substances

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes') %>% # only given to people with positive screens
  summarise(
    mean = mean(vic_worst_incap, na.rm = T),
    sd = sd(vic_worst_incap, na.rm = T),
    min = min(vic_worst_incap, na.rm = T),
    max = max(vic_worst_incap, na.rm = T),
    n = n(),
    n_missing = sum(is.na(vic_worst_incap))) %>%
  round(2)
```

```
##   mean   sd min max   n n_missing
## 1 1.62 0.94   1   5 596        24
```

- Threatened or actual force

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes') %>% # only given to people with positive screens
  summarise(
    mean = mean(vic_worst_force, na.rm = T),
    sd = sd(vic_worst_force, na.rm = T),
    min = min(vic_worst_force, na.rm = T),
    max = max(vic_worst_force, na.rm = T),
    n = n(),
    n_missing = sum(is.na(vic_worst_force))) %>%
  round(2)
```

```
##   mean   sd min max   n n_missing
## 1 1.33 0.67   1   5 596        24
```

- History of consent with perpetrator

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes') %>% # only given to people with positive screens
  group_by(vic_worst_prior_sex) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 2 x 3
##   vic_worst_prior_sex     n     p
##   <lgl>               <int> <dbl>
## 1 FALSE                 338 0.567
## 2 TRUE                  258 0.433
```

- Sexual assault acknowledgment

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes') %>% # only given to people with positive screens
  summarise(
    mean = mean(vic_worst_is_sa, na.rm = T),
    sd = sd(vic_worst_is_sa, na.rm = T),
    min = min(vic_worst_is_sa, na.rm = T),
    max = max(vic_worst_is_sa, na.rm = T),
    n = n(),
    n_missing = sum(is.na(vic_worst_is_sa))) %>%
  round(2)
```

```
##   mean  sd min max   n n_missing
## 1  3.8 2.2   1   7 596        19
```

How many people were there before and after #MeToo?

```
table(df$post_metoo)
```

```
##
##    0    1
## 1068 1498
```

```
1498/(1068+1498)
```

```
## [1] 0.583788
```

Semester-by-semester

- Sexual assault prevalence

```
participant_df %>%
  filter(hours_to_complete <= 24) %>%
  filter(vic_screener == 'Yes' | vic_screener == 'No') %>% # only consider those with data on screener
  group_by(semester, vic_screener) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  group_by(semester) %>%
  mutate(p = round(n/sum(n), 3))
```

```
## # A tibble: 12 x 4
## # Groups:   semester [6]
##    semester    vic_screener     n     p
##    <fct>       <fct>        <int> <dbl>
##  1 Fall 2016   No             348 0.777
##  2 Fall 2016   Yes            100 0.223
##  3 Spring 2017 No             268 0.766
##  4 Spring 2017 Yes             82 0.234
##  5 Fall 2017   No             434 0.768
##  6 Fall 2017   Yes            131 0.232
##  7 Spring 2018 No             359 0.765
##  8 Spring 2018 Yes            110 0.235
##  9 Fall 2018   No             201 0.753
## 10 Fall 2018   Yes             66 0.247
## 11 Spring 2019 No             294 0.733
## 12 Spring 2019 Yes            107 0.267
```

- Mean acknowledgment

```
participant_df %>%
  group_by(semester) %>%
  summarise(
    n = n(),
    mean_ack = mean(vic_worst_is_sa, na.rm = T))
```

```
## # A tibble: 6 x 3
##   semester        n mean_ack
##   <fct>       <int>    <dbl>
## 1 Fall 2016     462     3.35
## 2 Spring 2017   372     3.92
## 3 Fall 2017     587     3.55
## 4 Spring 2018   483     3.95
## 5 Fall 2018     287     4.29
## 6 Spring 2019   425     4.05
```

Analysis Convenience Functions

The following functions are used to streamline analysis of the Bayesian models in the accompanying paper.

Missing data imputation

A function for imputing missing data to passing to parallel processing functions.

```
impute_mice <- function(seed, data, m = 25){
  mice(
    data = data,
    m = m,
    print = F,
    seed = seed)
}
```

Unpack mids (multiply imputed complete datasets) objects and reformat as a list that can be passed to later model functions.

```
unpack_parallel_mids <- function(par_mids_list){
  par_mids_list %>%
    map(complete, action = 'long') %>%
    bind_rows(.id = 'core') %>%
    mutate(
      run = group_indices(., core, .imp), # Compute unique imputation datasets
      vic_screener = as.numeric(vic_screener) - 1) %>% # MCMCpack wants numerics
    split(.$run)
}
```

Priors

Calculate the precision parameter (tau - the inverse of the variance) for the normally distributed priors.

```
calc_tau <- function(prior_sd){
  1/(prior_sd**2)
}
```

Count number of priors needed for a given model formula (2 + the number of +'s)

```r
n_priors <- function(model_formula){
  pluses <- max(str_count(model_formula, '\\+'))

  if(pluses == 0){
    return(1)
  } else {
    return(pluses + 2)
  }
}
```

MCMC logistic regression functions

Run a Bayesian logistic regression, given data, models, and a prior.

```r
run_logistic_mcmc <- function(df, model, prior_sd, mcmc = 1e4, seed, ...) {
  if(missing(seed)){
    seed <- as.integer(1e7*runif(1))
  }
  MCMClogit(
    formula = model,
    data = df,
    b0 = rep(0, n_priors(model)),
    B0 = calc_tau(prior_sd[1:n_priors(model)])*diag(n_priors(model)),
    mcmc = mcmc,
    marginal.likelihood = 'Laplace',
    seed = seed,
    ...)
}
```

Parallelize Bayesian logistic regressions across imputed data sets, then draw random samples from their posteriors.

```r
library(MCMCpack)
```

```
## Loading required package: coda
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2019 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
library(snow)

parallel_logistic_mcmc <- function(df_list, model, prior_sd, mcmc = 1e4, ...){
  parallel::mclapply(
      X = df_list,
      FUN = run_logistic_mcmc,
      model = model,
      prior_sd = prior_sd,
      mcmc = mcmc,
      ...) %>%
    map(as.data.frame) %>%
    bind_rows() %>%
    sample_n(size = mcmc, replace = T) %>%
    as.mcmc()
}

parallel_logistic_mcmc2 <- function(df_list, model, prior_sd, mcmc = 1e4, ...){

  cl <- makeCluster(spec = parallel::detectCores(), type='SOCK')

  clusterEvalQ(cl, library(MCMCpack))
  clusterExport(cl, c('n_priors', 'str_count', 'calc_tau', 'as.mcmc'))

  result <- snow::clusterApply(
      cl = cl,
      x = df_list,
      fun = run_logistic_mcmc,
      model = model,
      prior_sd = prior_sd,
      mcmc = mcmc,
      ...) %>%
    map(as.data.frame) %>%
    bind_rows() %>%
    sample_n(size = mcmc, replace = T) %>%
    as.mcmc()

  stopCluster(cl)

  return(result)
}
```

MCMC linear regression functions

Run a Bayesian linear regression, given data, models, and a prior.

```
run_linear_mcmc <- function(df, model, prior_sd, mcmc = 1e4, seed, ...) {
  if(missing(seed)){
    seed <- as.integer(1e7*runif(1))
  }

  MCMCregress(
    formula = model,
    data = df,
    b0 = rep(0, n_priors(model)),
    B0 = calc_tau(prior_sd[1:n_priors(model)])*diag(n_priors(model)),
    mcmc = mcmc,
    marginal.likelihood = 'Chib95',
    seed = seed,
    ...)
}
```

Parallelize Bayesian linear regressions across imputed data sets, then draw random samples from their posteriors.

```
parallel_linear_mcmc <- function(df_list, model, prior_sd, mcmc = 1e4, ...){
  parallel::mclapply(
      X = df_list,
      FUN = run_linear_mcmc,
      model = model,
      prior_sd = prior_sd,
      mcmc = mcmc,
      ...) %>%
    map(as.data.frame) %>%
    bind_rows() %>%
    sample_n(size = mcmc, replace = T) %>%
    as.mcmc()
}

parallel_linear_mcmc2 <- function(df_list, model, prior_sd, mcmc = 1e4, ...){

  cl <- makeCluster(spec = parallel::detectCores(), type='SOCK')

  clusterEvalQ(cl, library(MCMCpack))
  clusterExport(cl, c('n_priors', 'str_count', 'calc_tau', 'as.mcmc'))

  result <- snow::clusterApply(
      cl = cl,
      x = df_list,
      fun = run_linear_mcmc,
      model = model,
      prior_sd = prior_sd,
      mcmc = mcmc,
      ...) %>%
    map(as.data.frame) %>%
    bind_rows() %>%
    sample_n(size = mcmc, replace = T) %>%
    as.mcmc()
```

```
  stopCluster(cl)

  return(result)
}
```

Multiple Imputation

Missing data were handled using multiple imputation, implemented by the `mice` package. To ensure `mice` handled the dataset correctly, we first recast all categorical variables as factors.

Note, two imputed datasets were created: one for all participants and one specifically for participants who screened positive for history of sexual assault. This was done because some analyses only involved the subset of participants who screened positive for a history of sexual assault.

```
library(mice)

categorical_vars <- c( 'post_metoo', 'male', 'lgbtq', 'racial_minority',
                       'vic_screener', 'vic_worst_prior_sex')

df[, categorical_vars] <- purrr::map(df[, categorical_vars], as.factor)

imp_seeds <- map_int(1:parallel::detectCores(), ~ as.integer(runif(1)*1e8))
sets_per_core <- ceiling(100/parallel::detectCores())

full_imp <- parallel::mclapply(
    X = imp_seeds,
    FUN = impute_mice,
    data = df,
    m = sets_per_core) %>%
  unpack_parallel_mids()

vic_imp <- parallel::mclapply(
    X = imp_seeds,
    FUN = impute_mice,
    data = filter(df, vic_screener == 'Yes'),
    m = sets_per_core) %>%
  unpack_parallel_mids()

paste(sets_per_core, 'imputed datasets per core')
```

```
## [1] "7 imputed datasets per core"
```

```
paste(parallel::detectCores(), 'cores detected')
```

```
## [1] "16 cores detected"
```

Models and Priors

These are the models and priors used throughout the rest of the analysis. Note that in the paper, we clarify that the priors are all normally distributed, with mean = 0 and SD = 10. The attentive reader will thus observe that we didn't need to code all of our priors individually, if they are all the same. However, we did this so that if the code needed to be changed later, it would be easy to modify.

Constant prevalence hypothesis

H1: There are no changes in behavioral reports of sexual assault over time.

We begin by establishing the models to test.

```
h1_models <- list(
  intercept_only = formula(vic_screener ~ 1),
  demos = formula(vic_screener ~ male + lgbtq + racial_minority),
  demos_and_time = formula(vic_screener ~ male + lgbtq + racial_minority + week_num),
  demos_and_time_SQ = formula(
    vic_screener ~ male + lgbtq + racial_minority + week_num + week_num_SQ),
  post_metoo = formula(
    vic_screener ~ male + lgbtq + racial_minority + week_num + week_num_SQ +
      post_metoo),
  post_metoo_time = formula(
    vic_screener ~ male + lgbtq + racial_minority + week_num +
      week_num_SQ + post_metoo + post_metoo_week),
  post_metoo_time_SQ = formula(
    vic_screener ~ male + lgbtq + racial_minority + week_num + week_num_SQ +
    post_metoo + post_metoo_week + post_metoo_week_SQ))
```

Then identify uninformative priors, with SDs = 10.

```
h1_priors <- c(
  intercept = 10,
  male = 10,
  lgbtq = 10,
  racial_minority = 10,
  week_num = 10,
  week_num_SQ = 10,
  post_metoo = 10,
  post_metoo_week = 10,
  post_metoo_week_SQ = 10)
```

Increasing acknowledgment hypothesis

H2: A reliable increase will be observed in victims' acknowledgment of their experiences as 'sexual assault.'

```
h2_models <- list(
  intercept_only = formula(vic_worst_is_sa ~ 1),
  demos = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority),
  demos_and_context =  formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force),
  time = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force +
      week_num),
```

```
  time_SQ = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force +
      week_num + week_num_SQ),
  post_metoo = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force +
      week_num + week_num_SQ +
      post_metoo),
  post_metoo_time = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force +
      week_num + week_num_SQ +
      post_metoo + post_metoo_week),
  post_metoo_time_SQ = formula(vic_worst_is_sa ~  male + lgbtq + racial_minority +
      vic_worst_prior_sex + vic_worst_verbal + vic_worst_incap +
      vic_worst_force +
      week_num + week_num_SQ +
      post_metoo + post_metoo_week + post_metoo_week_SQ))
```

We use the same priors as above.

```
h2_priors <- c(
  intercept = 10,
  male = 10,
  lgbtq = 10,
  racial_minority = 10,
  vic_worst_prior_sex = 10,
  vic_worst_verbal = 10,
  vic_worst_incap = 10,
  vic_worst_force = 10,
  week_num = 10,
  week_num_SQ = 10,
  post_metoo = 10,
  post_metoo_week = 10,
  post_metoo_week_SQ = 10)
```

Model Estimation and Comparison

Hypothesis 1 - Constant Prevalence

Bayes factors are calculated with complete data because MCMCpack requires this.

```
library(MCMCpack)

h1_comp_post <- map(
  .x = h1_models,
  .f = ~ run_logistic_mcmc(
    df = df %>%
      mutate(vic_screener = as.numeric(vic_screener) - 1),
```

```
    model = .x,
    prior_sd = h1_priors))
```

Estimate the posterior distribution of complete data (`comp_post`), then compute Bayes factors comparing each subsequent model.

```
h1_bayes_factors <- h1_comp_post %>%
  map_dbl(attr, which = 'logmarglike') %>%
  data.frame(model = names(.), logmarglike = ., row.names = 1:length(.)) %>%
  mutate(
    model = as.character(model),
    BF = exp(logmarglike - lag(logmarglike))) %>%
  map_if(is.numeric, formatC, digits = 5, format = 'f', big.mark = ',') %>%
  as.data.frame()

h1_bayes_factors
```

```
##                model  logmarglike
## 1     intercept_only -1,378.43983
## 2              demos -1,306.38387
## 3     demos_and_time -1,313.92861
## 4  demos_and_time_SQ -1,326.41575
## 5         post_metoo -1,328.89719
## 6    post_metoo_time -1,336.21497
## 7 post_metoo_time_SQ -1,346.57343
##                                                BF
## 1                                              NA
## 2 19,656,419,195,564,889,022,319,527,198,720.00000
## 3                                         0.00053
## 4                                         0.00000
## 5                                         0.08362
## 6                                         0.00066
## 7                                         0.00003
```

Estimate the posterior distributions implied by each imputed dataset, then splice them. Perform this process for each model.

```
library(tidyverse)
library(MCMCpack)

h1_posteriors <- map(
  .x = h1_models,
  .f = ~ parallel_logistic_mcmc(
    df_list = full_imp,
    model = .x,
    prior_sd = h1_priors,
    mcmc = 6*1e4,
    thin = 2))

# Save to CSV
h1_posteriors %>%
  map(as.data.frame) %>%
```

```r
  bind_rows(.id = 'model') %>%
  gather(param, estimate, -model) %>%
  filter(complete.cases(.)) %>%
  write.csv(file = paste0('h1_posteriors_', Sys.Date(), '.csv'), row.names = F)
```

Summaries of the posteriors

```r
map(h1_posteriors, summary)
```

```
## $intercept_only
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##     -1.172e+00      6.064e-03      2.476e-05      2.476e-05
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## -1.185 -1.176 -1.174 -1.170 -1.159
##
##
## $demos
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean      SD  Naive SE Time-series SE
## (Intercept)      -0.8698 0.05949 0.0002429      0.0002429
## maleTRUE         -1.5570 0.15006 0.0006126      0.0006126
## lgbtqTRUE         0.6382 0.15487 0.0006323      0.0006323
## racial_minority1 -0.2963 0.11602 0.0004737      0.0004737
##
## 2. Quantiles for each variable:
##
##                     2.5%     25%     50%     75%    97.5%
## (Intercept)      -0.9873 -0.9096 -0.8694 -0.8298 -0.75330
## maleTRUE         -1.8573 -1.6571 -1.5553 -1.4537 -1.26933
## lgbtqTRUE         0.3309  0.5355  0.6389  0.7425  0.94003
## racial_minority1 -0.5258 -0.3744 -0.2954 -0.2181 -0.07174
##
##
```

```
## $demos_and_time
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean       SD  Naive SE Time-series SE
## (Intercept)      -0.886736 0.060148 2.456e-04      2.441e-04
## maleTRUE         -1.561754 0.148697 6.071e-04      6.071e-04
## lgbtqTRUE         0.636865 0.154759 6.318e-04      6.318e-04
## racial_minority1 -0.304527 0.115571 4.718e-04      4.718e-04
## week_num          0.001993 0.001149 4.692e-06      4.763e-06
##
## 2. Quantiles for each variable:
##
##                        2.5%       25%       50%       75%     97.5%
## (Intercept)      -1.0054211 -0.927050 -0.886553 -0.845901 -0.769915
## maleTRUE         -1.8600301 -1.660887 -1.558806 -1.460145 -1.276943
## lgbtqTRUE         0.3297768  0.532910  0.637834  0.741181  0.936027
## racial_minority1 -0.5341765 -0.382223 -0.303737 -0.226100 -0.080856
## week_num         -0.0002546  0.001216  0.001997  0.002765  0.004245
##
##
## $demos_and_time_SQ
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                         Mean        SD  Naive SE Time-series SE
## (Intercept)      -9.019e-01 7.743e-02 3.161e-04      3.183e-04
## maleTRUE         -1.564e+00 1.496e-01 6.107e-04      6.107e-04
## lgbtqTRUE         6.378e-01 1.548e-01 6.320e-04      6.320e-04
## racial_minority1 -3.048e-01 1.160e-01 4.736e-04      4.736e-04
## week_num          1.758e-03 1.357e-03 5.541e-06      5.541e-06
## week_num_SQ       9.147e-06 2.987e-05 1.219e-07      1.219e-07
##
## 2. Quantiles for each variable:
##
##                         2.5%        25%        50%        75%      97.5%
## (Intercept)      -1.054e+00 -9.541e-01 -9.017e-01 -8.495e-01 -7.519e-01
## maleTRUE         -1.865e+00 -1.663e+00 -1.561e+00 -1.461e+00 -1.276e+00
## lgbtqTRUE         3.312e-01  5.338e-01  6.380e-01  7.424e-01  9.396e-01
## racial_minority1 -5.335e-01 -3.832e-01 -3.044e-01 -2.263e-01 -7.855e-02
## week_num         -8.882e-04  8.456e-04  1.754e-03  2.670e-03  4.439e-03
## week_num_SQ      -4.919e-05 -1.102e-05  9.297e-06  2.931e-05  6.735e-05
```

```
##
##
## $post_metoo
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean        SD  Naive SE Time-series SE
## (Intercept)      -0.7025251 1.380e-01 5.632e-04      5.571e-04
## maleTRUE         -1.5589399 1.494e-01 6.098e-04      6.098e-04
## lgbtqTRUE         0.6367577 1.547e-01 6.315e-04      6.315e-04
## racial_minority1 -0.3067300 1.164e-01 4.754e-04      4.754e-04
## week_num          0.0055338 2.562e-03 1.046e-05      1.039e-05
## week_num_SQ      -0.0000139 3.286e-05 1.342e-07      1.340e-07
## post_metoo1      -0.3329885 1.910e-01 7.799e-04      7.584e-04
##
## 2. Quantiles for each variable:
##
##                        2.5%        25%        50%        75%      97.5%
## (Intercept)      -9.758e-01 -7.949e-01 -7.023e-01 -6.086e-01 -4.336e-01
## maleTRUE         -1.856e+00 -1.659e+00 -1.556e+00 -1.457e+00 -1.273e+00
## lgbtqTRUE         3.294e-01  5.332e-01  6.376e-01  7.417e-01  9.371e-01
## racial_minority1 -5.375e-01 -3.848e-01 -3.061e-01 -2.274e-01 -8.095e-02
## week_num          5.208e-04  3.791e-03  5.530e-03  7.253e-03  1.057e-02
## week_num_SQ      -7.858e-05 -3.599e-05 -1.407e-05  8.474e-06  5.021e-05
## post_metoo1      -7.078e-01 -4.624e-01 -3.317e-01 -2.044e-01  4.069e-02
##
##
## $post_metoo_time
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean        SD  Naive SE Time-series SE
## (Intercept)      -6.922e-01 0.1569546 6.408e-04      6.408e-04
## maleTRUE         -1.562e+00 0.1502724 6.135e-04      6.135e-04
## lgbtqTRUE         6.363e-01 0.1547698 6.318e-04      6.318e-04
## racial_minority1 -3.073e-01 0.1161208 4.741e-04      4.741e-04
## week_num          6.416e-03 0.0076420 3.120e-05      3.120e-05
## week_num_SQ      -2.053e-07 0.0001171 4.781e-07      4.781e-07
## post_metoo1      -3.299e-01 0.1943278 7.933e-04      7.933e-04
## post_metoo_week  -2.032e-03 0.0167645 6.844e-05      6.844e-05
##
## 2. Quantiles for each variable:
```

```
##
##                          2.5%        25%        50%        75%    97.5%
## (Intercept)      -1.0040027 -7.973e-01 -6.918e-01 -5.864e-01 -0.38485
## maleTRUE         -1.8635482 -1.662e+00 -1.561e+00 -1.459e+00 -1.27532
## lgbtqTRUE         0.3316921  5.327e-01  6.357e-01  7.418e-01  0.93651
## racial_minority1 -0.5337385 -3.853e-01 -3.069e-01 -2.284e-01 -0.08207
## week_num         -0.0085259  1.257e-03  6.462e-03  1.156e-02  0.02148
## week_num_SQ      -0.0002298 -7.869e-05  2.343e-07  7.826e-05  0.00023
## post_metoo1      -0.7119947 -4.597e-01 -3.293e-01 -1.989e-01  0.05012
## post_metoo_week  -0.0350430 -1.322e-02 -2.114e-03  9.196e-03  0.03077
##
##
## $post_metoo_time_SQ
##
## Iterations = 1:60000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 60000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                           Mean        SD  Naive SE Time-series SE
## (Intercept)        -0.7274405 0.1860037 7.594e-04      7.594e-04
## maleTRUE           -1.5615981 0.1500449 6.126e-04      6.088e-04
## lgbtqTRUE           0.6350938 0.1553620 6.343e-04      6.343e-04
## racial_minority1   -0.3079653 0.1167400 4.766e-04      4.766e-04
## week_num            0.0016845 0.0151256 6.175e-05      6.175e-05
## week_num_SQ        -0.0000857 0.0002623 1.071e-06      1.071e-06
## post_metoo1        -0.2738007 0.2498288 1.020e-03      1.012e-03
## post_metoo_week     0.0009877 0.0187768 7.666e-05      7.666e-05
## post_metoo_week_SQ  0.0001061 0.0002935 1.198e-06      1.198e-06
##
## 2. Quantiles for each variable:
##
##                          2.5%        25%        50%        75%      97.5%
## (Intercept)        -1.0982963 -8.523e-01 -0.7258182 -6.008e-01 -0.3681649
## maleTRUE           -1.8615089 -1.662e+00 -1.5597026 -1.458e+00 -1.2751076
## lgbtqTRUE           0.3298350  5.309e-01  0.6358265  7.395e-01  0.9398068
## racial_minority1   -0.5377022 -3.869e-01 -0.3076681 -2.286e-01 -0.0802777
## week_num           -0.0283175 -8.451e-03  0.0017351  1.191e-02  0.0309030
## week_num_SQ        -0.0006032 -2.609e-04 -0.0000840  9.032e-05  0.0004255
## post_metoo1        -0.7624513 -4.424e-01 -0.2733351 -1.067e-01  0.2144674
## post_metoo_week    -0.0352580 -1.176e-02  0.0009742  1.350e-02  0.0382627
## post_metoo_week_SQ -0.0004649 -9.285e-05  0.0001060  3.027e-04  0.0006873
```

```
h1_param_means <- map(h1_comp_post, colMeans) %>%
  map(data.frame) %>%
  map(~ set_names(.x, 'mean_est')) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  mutate(original_order = row_number())

h1_hpdi_97 <- map(h1_comp_post, HPDinterval, prob = .97) %>%
```

```r
  map(data.frame) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  dplyr::rename(lower_97 = lower, upper_97 = upper)

h1_hpdi_95 <- map(h1_comp_post, HPDinterval, prob = .95) %>%
  map(data.frame) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  dplyr::rename(lower_95 = lower, upper_95 = upper)

h1_params <- h1_param_means %>%
  left_join(h1_hpdi_97, by = c('model', 'param')) %>%
  left_join(h1_hpdi_95, by = c('model', 'param'))

h1_params
```

```
##                  model      mean_est              param original_order
## 1       intercept_only -1.161472e+00        (Intercept)              1
## 2                demos -8.605255e-01        (Intercept)              2
## 3                demos -1.557825e+00           maleTRUE              3
## 4                demos  6.510994e-01          lgbtqTRUE              4
## 5                demos -2.889889e-01    racial_minority1              5
## 6       demos_and_time -8.751223e-01        (Intercept)              6
## 7       demos_and_time -1.570328e+00           maleTRUE              7
## 8       demos_and_time  6.545828e-01          lgbtqTRUE              8
## 9       demos_and_time -3.080189e-01    racial_minority1              9
## 10      demos_and_time  2.063230e-03           week_num             10
## 11   demos_and_time_SQ -8.913287e-01        (Intercept)             11
## 12   demos_and_time_SQ -1.565685e+00           maleTRUE             12
## 13   demos_and_time_SQ  6.421851e-01          lgbtqTRUE             13
## 14   demos_and_time_SQ -3.005697e-01    racial_minority1             14
## 15   demos_and_time_SQ  1.799801e-03           week_num             15
## 16   demos_and_time_SQ  9.387570e-06        week_num_SQ             16
## 17          post_metoo -6.850868e-01        (Intercept)             17
## 18          post_metoo -1.555198e+00           maleTRUE             18
## 19          post_metoo  6.430341e-01          lgbtqTRUE             19
## 20          post_metoo -3.067355e-01    racial_minority1             20
## 21          post_metoo  5.582406e-03           week_num             21
## 22          post_metoo -1.400851e-05        week_num_SQ             22
## 23          post_metoo -3.422851e-01        post_metoo1             23
## 24     post_metoo_time -6.703347e-01        (Intercept)             24
## 25     post_metoo_time -1.566750e+00           maleTRUE             25
## 26     post_metoo_time  6.389077e-01          lgbtqTRUE             26
## 27     post_metoo_time -3.135895e-01    racial_minority1             27
## 28     post_metoo_time  6.746423e-03           week_num             28
## 29     post_metoo_time  3.708741e-09        week_num_SQ             29
## 30     post_metoo_time -3.383958e-01        post_metoo1             30
## 31     post_metoo_time -2.442531e-03    post_metoo_week             31
## 32  post_metoo_time_SQ -7.212757e-01        (Intercept)             32
## 33  post_metoo_time_SQ -1.563486e+00           maleTRUE             33
## 34  post_metoo_time_SQ  6.589805e-01          lgbtqTRUE             34
## 35  post_metoo_time_SQ -3.113713e-01    racial_minority1             35
```

```
## 36 post_metoo_time_SQ  1.697278e-03          week_num             36
## 37 post_metoo_time_SQ -8.519496e-05       week_num_SQ             37
## 38 post_metoo_time_SQ -2.696796e-01       post_metoo1             38
## 39 post_metoo_time_SQ  1.040243e-03   post_metoo_week             39
## 40 post_metoo_time_SQ  1.029846e-04 post_metoo_week_SQ            40
##         lower_97      upper_97      lower_95      upper_95
## 1  -1.161472e+00 -1.161472e+00 -1.161472e+00 -1.161472e+00
## 2  -9.823657e-01 -7.240646e-01 -9.729835e-01 -7.419398e-01
## 3  -1.887389e+00 -1.226621e+00 -1.858027e+00 -1.257161e+00
## 4   3.099024e-01  9.758841e-01  3.672192e-01  9.621820e-01
## 5  -5.367332e-01 -1.967686e-02 -5.315866e-01 -6.160924e-02
## 6  -1.008214e+00 -7.518152e-01 -9.918802e-01 -7.608503e-01
## 7  -1.894559e+00 -1.268462e+00 -1.842552e+00 -1.283482e+00
## 8   3.403698e-01  1.005535e+00  3.403698e-01  9.351123e-01
## 9  -5.663265e-01 -6.099349e-02 -5.362572e-01 -8.379391e-02
## 10 -1.897394e-04  4.707469e-03 -4.215562e-06  4.320673e-03
## 11 -1.064814e+00 -7.269613e-01 -1.044271e+00 -7.365631e-01
## 12 -1.902394e+00 -1.266427e+00 -1.867637e+00 -1.300179e+00
## 13  3.391574e-01  1.002241e+00  3.391574e-01  9.406148e-01
## 14 -5.650169e-01 -6.584165e-02 -5.247370e-01 -8.040466e-02
## 15 -9.939577e-04  4.811566e-03 -9.919030e-04  4.253359e-03
## 16 -5.797015e-05  7.371878e-05 -4.608183e-05  7.200555e-05
## 17 -9.859085e-01 -3.888784e-01 -9.415601e-01 -4.008513e-01
## 18 -1.880643e+00 -1.248171e+00 -1.860183e+00 -1.291462e+00
## 19  2.949259e-01  9.655723e-01  3.212118e-01  9.231800e-01
## 20 -5.892825e-01 -6.431702e-02 -5.325179e-01 -6.639377e-02
## 21  9.456724e-05  1.069464e-02  7.844936e-04  1.060286e-02
## 22 -7.979668e-05  5.949311e-05 -7.403035e-05  5.136987e-05
## 23 -7.255369e-01  8.747438e-02 -7.048203e-01  2.619543e-02
## 24 -1.030885e+00 -3.094063e-01 -1.022198e+00 -3.631725e-01
## 25 -1.882985e+00 -1.221086e+00 -1.856603e+00 -1.264509e+00
## 26  2.951389e-01  9.614669e-01  3.469794e-01  9.600234e-01
## 27 -5.596408e-01 -5.596697e-02 -5.467531e-01 -9.114183e-02
## 28 -1.174790e-02  2.338161e-02 -1.112278e-02  2.095019e-02
## 29 -2.727531e-04  2.644408e-04 -2.546684e-04  2.265615e-04
## 30 -7.619915e-01  1.072312e-01 -7.508691e-01  3.697182e-02
## 31 -3.929453e-02  3.718791e-02 -3.241094e-02  3.718791e-02
## 32 -1.122691e+00 -2.876781e-01 -1.093877e+00 -3.407429e-01
## 33 -1.873238e+00 -1.213594e+00 -1.861249e+00 -1.278610e+00
## 34  3.326345e-01  9.909227e-01  3.319415e-01  9.371740e-01
## 35 -5.434988e-01 -3.242927e-02 -5.498439e-01 -8.975239e-02
## 36 -3.018353e-02  3.635220e-02 -2.710022e-02  3.424042e-02
## 37 -6.441850e-04  5.223892e-04 -6.352473e-04  4.470427e-04
## 38 -8.506460e-01  2.715678e-01 -8.577533e-01  1.765016e-01
## 39 -4.521653e-02  3.950017e-02 -4.002423e-02  3.516989e-02
## 40 -5.819643e-04  6.927084e-04 -4.501488e-04  6.833069e-04
```

```
write.csv(h1_params, paste0('h1_params_', Sys.Date(), '.csv'), row.names = F)
```

Hypothesis 2 - Increasing Acknow edgment

Bayes factors are calculated with complete data because MCMCpack requires this.

```
h2_comp_post <- map(
  .x = h2_models,
  .f = ~ run_linear_mcmc(
    df = df,
    model = .x,
    prior_sd = h2_priors))
```

Estimate the posterior distribution of complete data (`comp_post`), then compute Bayes factors comparing each subsequent model.

```
h2_bayes_factors <- h2_comp_post %>%
  map_dbl(attr, which = 'logmarglike') %>%
  data.frame(model = names(.), logmarglike = ., row.names = 1:length(.)) %>%
  mutate(
    model = as.character(model),
    BF = exp(logmarglike - lag(logmarglike))) %>%
  map_if(is.numeric, formatC, digits = 5, format = 'f', big.mark = ',') %>%
  as.data.frame()

h2_bayes_factors
```

```
##                model  logmarglike
## 1     intercept_only -1,288.69005
## 2              demos -1,288.65833
## 3  demos_and_context -1,210.58796
## 4               time -1,215.35232
## 5            time_SQ -1,227.55852
## 6         post_metoo -1,231.02514
## 7    post_metoo_time -1,236.78897
## 8 post_metoo_time_SQ -1,245.38177
##                                                    BF
## 1                                                  NA
## 2                                             1.03222
## 3 8,045,103,140,013,963,422,936,160,087,834,624.00000
## 4                                             0.00853
## 5                                             0.00000
## 6                                             0.03122
## 7                                             0.00314
## 8                                             0.00019
```

Estimate the posterior distributions implied by each imputed dataset, then splice them. Perform this process for each model.

```
set.seed(3019301)

h2_posteriors <- map(
  .x = h2_models,
  .f = ~ parallel_linear_mcmc2(
    df_list = vic_imp,
    model = .x,
    prior_sd = h2_priors,
    mcmc = 1e4))
```

```r
# Save to CSV
h2_posteriors %>%
  map(as.data.frame) %>%
  bind_rows(.id = 'model') %>%
  gather(param, estimate, -model) %>%
  filter(complete.cases(.)) %>%
  write.csv(file = paste0('h2_posteriors_', Sys.Date(), '.csv'), row.names = F)
```

Summaries of the posteriors

```r
map(h2_posteriors, summary)
```

```
## $intercept_only
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean     SD Naive SE Time-series SE
## (Intercept) 3.810 0.0920 0.000920       0.000920
## sigma2      4.884 0.2843 0.002843       0.002843
##
## 2. Quantiles for each variable:
##
##              2.5%   25%   50%   75% 97.5%
## (Intercept) 3.631 3.748 3.810 3.871 3.992
## sigma2      4.366 4.690 4.868 5.068 5.476
##
##
## $demos
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                      Mean     SD Naive SE Time-series SE
## (Intercept)       3.73367 0.1119 0.001119       0.001119
## maleTRUE         -0.73898 0.3226 0.003226       0.003226
## lgbtqTRUE         1.11969 0.2672 0.002672       0.002672
## racial_minority1 -0.02818 0.2196 0.002196       0.002196
## sigma2            4.72537 0.2797 0.002797       0.002797
##
## 2. Quantiles for each variable:
##
```

```
##                    2.5%     25%      50%     75%   97.5%
## (Intercept)       3.5141  3.6573  3.73347  3.8094  3.9518
## maleTRUE         -1.3698 -0.9570 -0.73847 -0.5239 -0.1186
## lgbtqTRUE         0.5993  0.9424  1.11589  1.3011  1.6389
## racial_minority1 -0.4521 -0.1811 -0.03054  0.1235  0.3973
## sigma2            4.2176  4.5291  4.71413  4.9111  5.2981
##
##
## $demos_and_context
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                            Mean      SD  Naive SE Time-series SE
## (Intercept)              1.2182 0.29127 0.0029127      0.0029606
## maleTRUE                -0.6511 0.28096 0.0028096      0.0028096
## lgbtqTRUE                0.9348 0.23363 0.0023363      0.0023363
## racial_minority1        -0.1499 0.19329 0.0019329      0.0020017
## vic_worst_prior_sexTRUE -0.6714 0.16757 0.0016757      0.0016757
## vic_worst_verbal         0.2473 0.11560 0.0011560      0.0012255
## vic_worst_incap          0.5306 0.08886 0.0008886      0.0008886
## vic_worst_force          1.1621 0.12636 0.0012636      0.0013073
## sigma2                   3.5703 0.21176 0.0021176      0.0021176
##
## 2. Quantiles for each variable:
##
##                            2.5%     25%     50%      75%   97.5%
## (Intercept)              0.6502  1.0200  1.2140  1.41743  1.7885
## maleTRUE                -1.2015 -0.8417 -0.6469 -0.45927 -0.1047
## lgbtqTRUE                0.4770  0.7761  0.9345  1.08773  1.3956
## racial_minority1        -0.5302 -0.2799 -0.1498 -0.02163  0.2252
## vic_worst_prior_sexTRUE -1.0026 -0.7826 -0.6726 -0.55971 -0.3459
## vic_worst_verbal         0.0190  0.1720  0.2472  0.32416  0.4745
## vic_worst_incap          0.3577  0.4697  0.5312  0.59114  0.7034
## vic_worst_force          0.9185  1.0740  1.1618  1.24897  1.4111
## sigma2                   3.1850  3.4248  3.5624  3.70528  4.0073
##
##
## $time
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                              Mean       SD  Naive SE Time-series SE
```

```
## (Intercept)             1.185304 0.290484 2.905e-03      2.905e-03
## maleTRUE               -0.668908 0.276581 2.766e-03      2.766e-03
## lgbtqTRUE               0.901318 0.233456 2.335e-03      2.335e-03
## racial_minority1       -0.178364 0.192854 1.929e-03      1.887e-03
## vic_worst_prior_sexTRUE -0.681519 0.165185 1.652e-03      1.652e-03
## vic_worst_verbal        0.234801 0.115111 1.151e-03      1.172e-03
## vic_worst_incap         0.535375 0.087179 8.718e-04      8.718e-04
## vic_worst_force         1.167349 0.127045 1.270e-03      1.270e-03
## week_num                0.005179 0.001853 1.853e-05      1.853e-05
## sigma2                  3.528974 0.210354 2.104e-03      2.104e-03
##
## 2. Quantiles for each variable:
##
##                             2.5%       25%       50%       75%     97.5%
## (Intercept)             0.621593  0.986956  1.185379  1.384512  1.748759
## maleTRUE               -1.217587 -0.854935 -0.668950 -0.481599 -0.131860
## lgbtqTRUE               0.444952  0.740138  0.900976  1.060580  1.356137
## racial_minority1       -0.559032 -0.312014 -0.176412 -0.046850  0.199216
## vic_worst_prior_sexTRUE -1.008533 -0.792325 -0.682452 -0.570487 -0.362859
## vic_worst_verbal        0.008912  0.156759  0.236692  0.313168  0.458661
## vic_worst_incap         0.362322  0.476971  0.536518  0.592584  0.707294
## vic_worst_force         0.916142  1.081446  1.167293  1.253914  1.415760
## week_num                0.001560  0.003892  0.005172  0.006442  0.008831
## sigma2                  3.142547  3.382056  3.519674  3.665954  3.961080
##
##
## $time_SQ
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                              Mean        SD  Naive SE Time-series SE
## (Intercept)             1.1737414 3.031e-01 3.031e-03      3.031e-03
## maleTRUE               -0.6693122 2.767e-01 2.767e-03      2.767e-03
## lgbtqTRUE               0.9030929 2.345e-01 2.345e-03      2.345e-03
## racial_minority1       -0.1843849 1.937e-01 1.937e-03      1.937e-03
## vic_worst_prior_sexTRUE -0.6815818 1.677e-01 1.677e-03      1.677e-03
## vic_worst_verbal        0.2329343 1.152e-01 1.152e-03      1.152e-03
## vic_worst_incap         0.5342882 8.775e-02 8.775e-04      8.932e-04
## vic_worst_force         1.1669589 1.265e-01 1.265e-03      1.265e-03
## week_num                0.0048456 2.241e-03 2.241e-05      2.241e-05
## week_num_SQ             0.0000116 4.861e-05 4.861e-07      4.861e-07
## sigma2                  3.5365455 2.091e-01 2.091e-03      2.091e-03
##
## 2. Quantiles for each variable:
##
##                              2.5%        25%        50%        75%
## (Intercept)             5.806e-01  9.705e-01  1.179e+00  1.373e+00
## maleTRUE               -1.206e+00 -8.573e-01 -6.696e-01 -4.834e-01
```

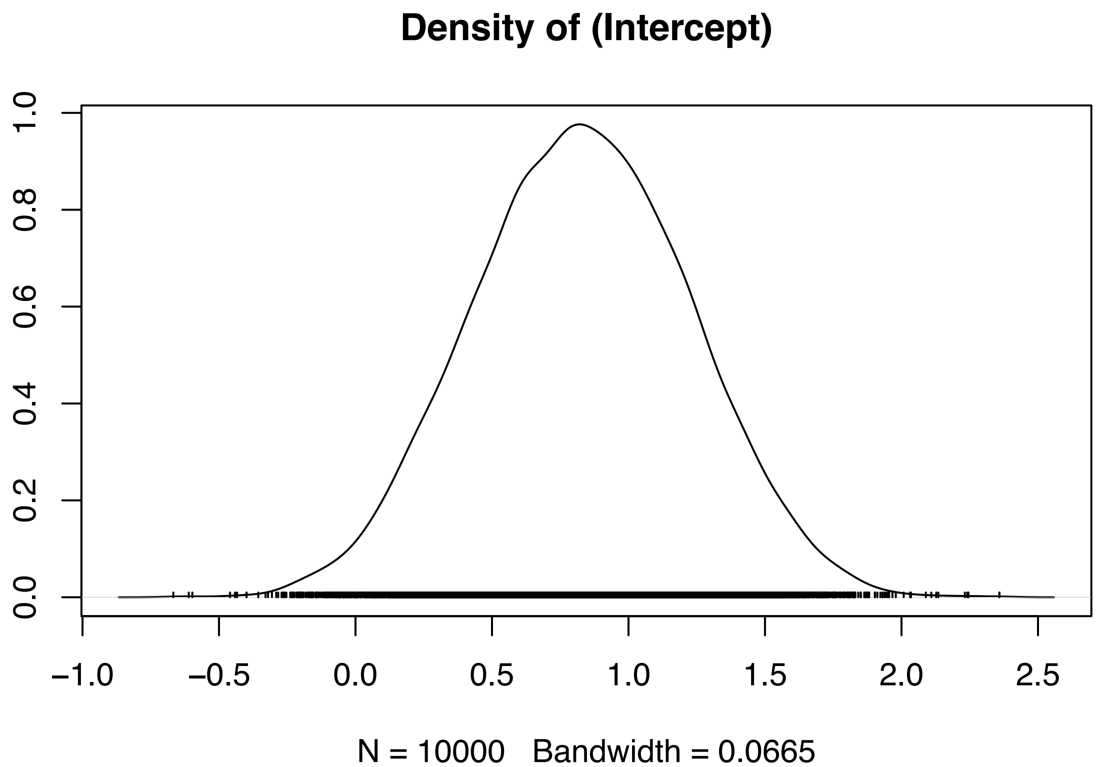
```
## lgbtqTRUE                4.478e-01  7.458e-01  9.016e-01  1.062e+00
## racial_minority1        -5.718e-01 -3.130e-01 -1.821e-01 -5.146e-02
## vic_worst_prior_sexTRUE -1.008e+00 -7.945e-01 -6.810e-01 -5.675e-01
## vic_worst_verbal         6.135e-03  1.572e-01  2.322e-01  3.098e-01
## vic_worst_incap          3.620e-01  4.757e-01  5.343e-01  5.930e-01
## vic_worst_force          9.184e-01  1.082e+00  1.167e+00  1.252e+00
## week_num                 3.928e-04  3.326e-03  4.851e-03  6.364e-03
## week_num_SQ             -8.441e-05 -2.106e-05  1.117e-05  4.411e-05
## sigma2                   3.150e+00  3.389e+00  3.531e+00  3.673e+00
##                              97.5%
## (Intercept)              1.7713119
## maleTRUE                -0.1292070
## lgbtqTRUE                1.3662450
## racial_minority1         0.1853507
## vic_worst_prior_sexTRUE -0.3543508
## vic_worst_verbal         0.4582967
## vic_worst_incap          0.7063463
## vic_worst_force          1.4108434
## week_num                 0.0092250
## week_num_SQ              0.0001075
## sigma2                   3.9683379
##
##
## $post_metoo
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                               Mean        SD  Naive SE Time-series SE
## (Intercept)              1.1436994 3.429e-01 3.429e-03      3.607e-03
## maleTRUE                -0.6689025 2.794e-01 2.794e-03      2.794e-03
## lgbtqTRUE                0.9025535 2.340e-01 2.340e-03      2.340e-03
## racial_minority1        -0.1816868 1.919e-01 1.919e-03      1.919e-03
## vic_worst_prior_sexTRUE -0.6836100 1.666e-01 1.666e-03      1.666e-03
## vic_worst_verbal         0.2355147 1.149e-01 1.149e-03      1.149e-03
## vic_worst_incap          0.5356594 8.860e-02 8.860e-04      8.860e-04
## vic_worst_force          1.1668511 1.268e-01 1.268e-03      1.268e-03
## week_num                 0.0044443 4.189e-03 4.189e-05      4.122e-05
## week_num_SQ              0.0000148 5.356e-05 5.356e-07      5.514e-07
## post_metoo1              0.0386708 3.115e-01 3.115e-03      3.041e-03
## sigma2                   3.5414844 2.110e-01 2.110e-03      2.110e-03
##
## 2. Quantiles for each variable:
##
##                               2.5%        25%        50%        75%
## (Intercept)              4.665e-01  9.145e-01  1.141e+00  1.372e+00
## maleTRUE                -1.204e+00 -8.566e-01 -6.731e-01 -4.827e-01
## lgbtqTRUE                4.521e-01  7.426e-01  9.030e-01  1.061e+00
## racial_minority1        -5.595e-01 -3.117e-01 -1.813e-01 -5.130e-02
```

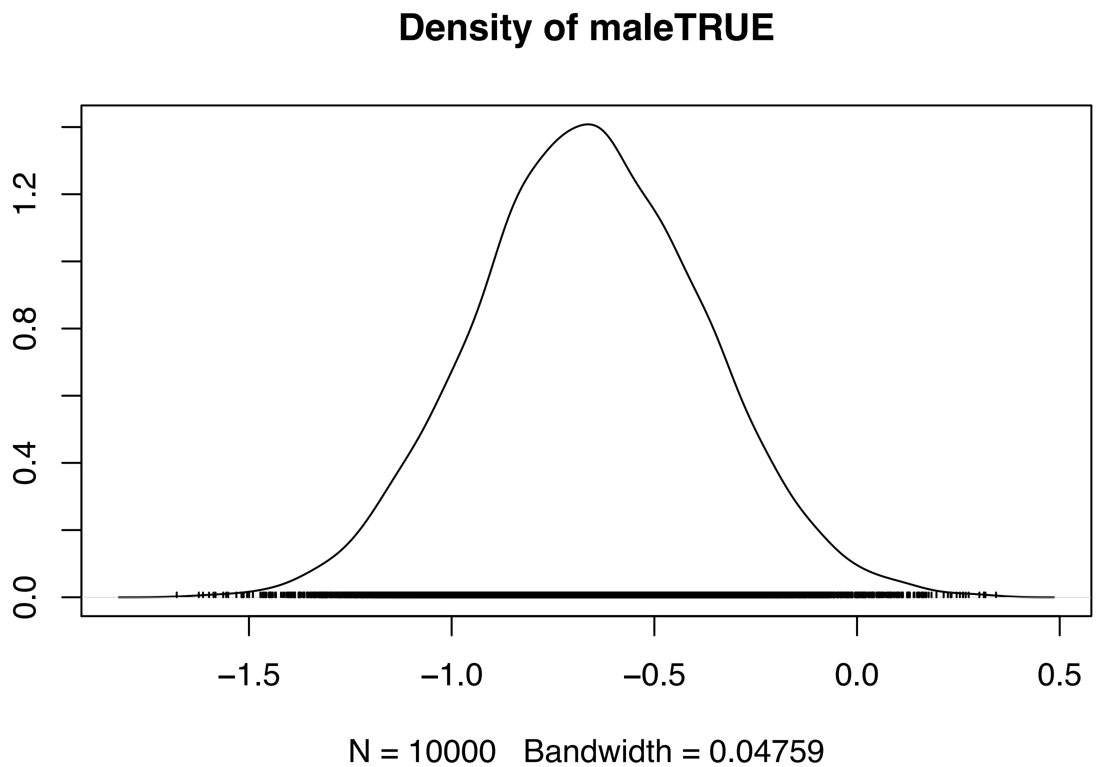
```
## vic_worst_prior_sexTRUE -1.004e+00 -7.973e-01 -6.856e-01 -5.752e-01
## vic_worst_verbal         7.489e-03  1.579e-01  2.356e-01  3.126e-01
## vic_worst_incap          3.617e-01  4.760e-01  5.358e-01  5.946e-01
## vic_worst_force          9.177e-01  1.080e+00  1.168e+00  1.253e+00
## week_num                -3.772e-03  1.694e-03  4.490e-03  7.250e-03
## week_num_SQ             -8.928e-05 -2.101e-05  1.474e-05  5.082e-05
## post_metoo1             -5.733e-01 -1.683e-01  3.853e-02  2.438e-01
## sigma2                   3.148e+00  3.396e+00  3.532e+00  3.680e+00
##                              97.5%
## (Intercept)              1.8113707
## maleTRUE                -0.1119072
## lgbtqTRUE                1.3584174
## racial_minority1         0.1884960
## vic_worst_prior_sexTRUE -0.3484704
## vic_worst_verbal         0.4648910
## vic_worst_incap          0.7098373
## vic_worst_force          1.4138863
## week_num                 0.0126118
## week_num_SQ              0.0001207
## post_metoo1              0.6541857
## sigma2                   3.9775325
##
##
## $post_metoo_time
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                               Mean        SD  Naive SE Time-series SE
## (Intercept)              1.090e+00 0.3624674 3.625e-03      3.468e-03
## maleTRUE                -6.525e-01 0.2837599 2.838e-03      2.838e-03
## lgbtqTRUE                8.956e-01 0.2340318 2.340e-03      2.340e-03
## racial_minority1        -1.750e-01 0.1911202 1.911e-03      1.838e-03
## vic_worst_prior_sexTRUE -6.789e-01 0.1655703 1.656e-03      1.671e-03
## vic_worst_verbal         2.346e-01 0.1148955 1.149e-03      1.149e-03
## vic_worst_incap          5.311e-01 0.0891224 8.912e-04      8.881e-04
## vic_worst_force          1.170e+00 0.1284959 1.285e-03      1.260e-03
## week_num                -7.865e-04 0.0125081 1.251e-04      1.219e-04
## week_num_SQ             -6.695e-05 0.0001971 1.971e-06      1.971e-06
## post_metoo1              7.988e-03 0.3242601 3.243e-03      3.102e-03
## post_metoo_week          1.207e-02 0.0279267 2.793e-04      2.741e-04
## sigma2                   3.550e+00 0.2101298 2.101e-03      2.101e-03
##
## 2. Quantiles for each variable:
##
##                               2.5%        25%        50%        75%
## (Intercept)              0.3716147  0.8440158  1.0926286  1.335e+00
## maleTRUE                -1.2044304 -0.8432324 -0.6530988 -4.594e-01
## lgbtqTRUE                0.4408227  0.7402224  0.8951325  1.051e+00
```

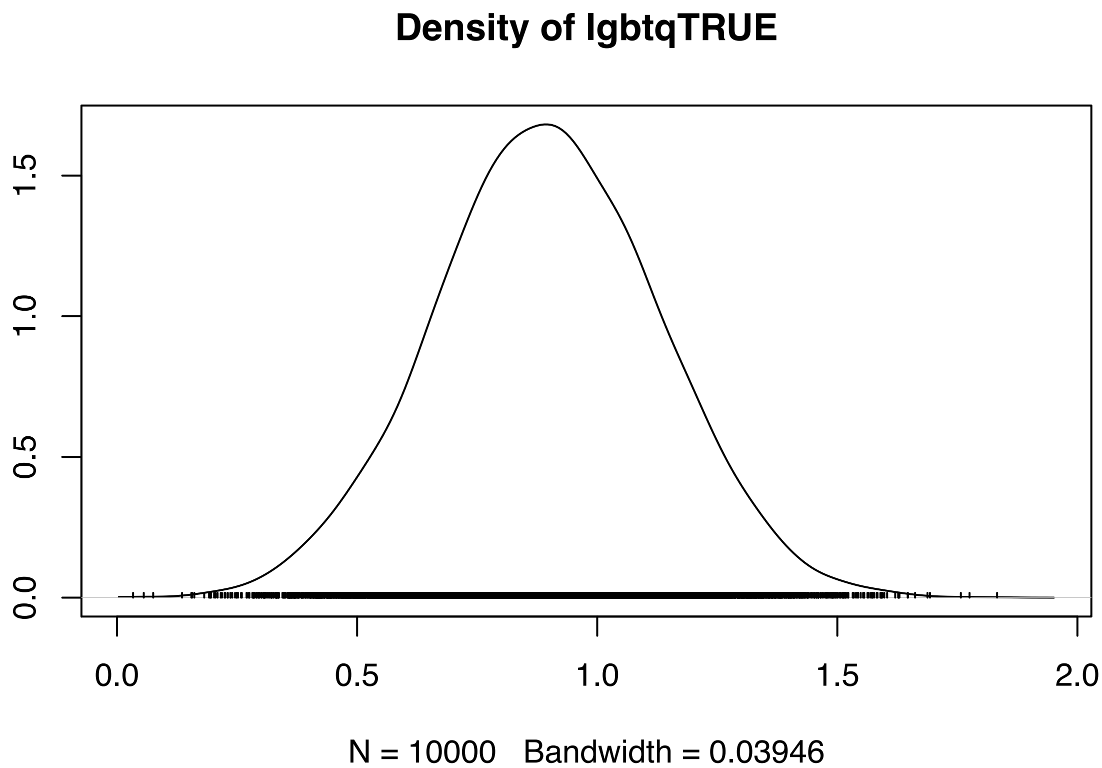
```
## racial_minority1        -0.5517202 -0.3026533 -0.1735122 -4.649e-02
## vic_worst_prior_sexTRUE -1.0026116 -0.7897890 -0.6800948 -5.641e-01
## vic_worst_verbal         0.0122558  0.1556611  0.2340521  3.126e-01
## vic_worst_incap          0.3544190  0.4727184  0.5320265  5.911e-01
## vic_worst_force          0.9212393  1.0834088  1.1701317  1.256e+00
## week_num                -0.0254703 -0.0091709 -0.0006940  7.607e-03
## week_num_SQ             -0.0004579 -0.0001994 -0.0000649  6.555e-05
## post_metoo1             -0.6377534 -0.2075016  0.0104806  2.244e-01
## post_metoo_week         -0.0426597 -0.0066637  0.0117947  3.089e-02
## sigma2                   3.1547815  3.4052843  3.5421237  3.687e+00
##                              97.5%
## (Intercept)              1.8009966
## maleTRUE                -0.1003684
## lgbtqTRUE                1.3692757
## racial_minority1         0.2020823
## vic_worst_prior_sexTRUE -0.3594371
## vic_worst_verbal         0.4574037
## vic_worst_incap          0.7031699
## vic_worst_force          1.4229761
## week_num                 0.0237959
## week_num_SQ              0.0003156
## post_metoo1              0.6437823
## post_metoo_week          0.0669757
## sigma2                   3.9810691
##
##
## $post_metoo_time_SQ
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                               Mean        SD  Naive SE Time-series SE
## (Intercept)              0.8292088 0.3958148 3.958e-03      3.958e-03
## maleTRUE                -0.6605150 0.2833032 2.833e-03      2.833e-03
## lgbtqTRUE                0.8946653 0.2348987 2.349e-03      2.349e-03
## racial_minority1        -0.1833450 0.1928774 1.929e-03      1.839e-03
## vic_worst_prior_sexTRUE -0.6836718 0.1673858 1.674e-03      1.650e-03
## vic_worst_verbal         0.2440506 0.1142443 1.142e-03      1.101e-03
## vic_worst_incap          0.5333099 0.0883061 8.831e-04      8.831e-04
## vic_worst_force          1.1705309 0.1259183 1.259e-03      1.259e-03
## week_num                -0.0362118 0.0250081 2.501e-04      2.553e-04
## week_num_SQ             -0.0007244 0.0004464 4.464e-06      4.544e-06
## post_metoo1              0.4034064 0.4007564 4.008e-03      4.090e-03
## post_metoo_week          0.0347544 0.0313294 3.133e-04      3.126e-04
## post_metoo_week_SQ       0.0008119 0.0004906 4.906e-06      4.997e-06
## sigma2                   3.5386304 0.2092159 2.092e-03      2.092e-03
##
## 2. Quantiles for each variable:
##
```

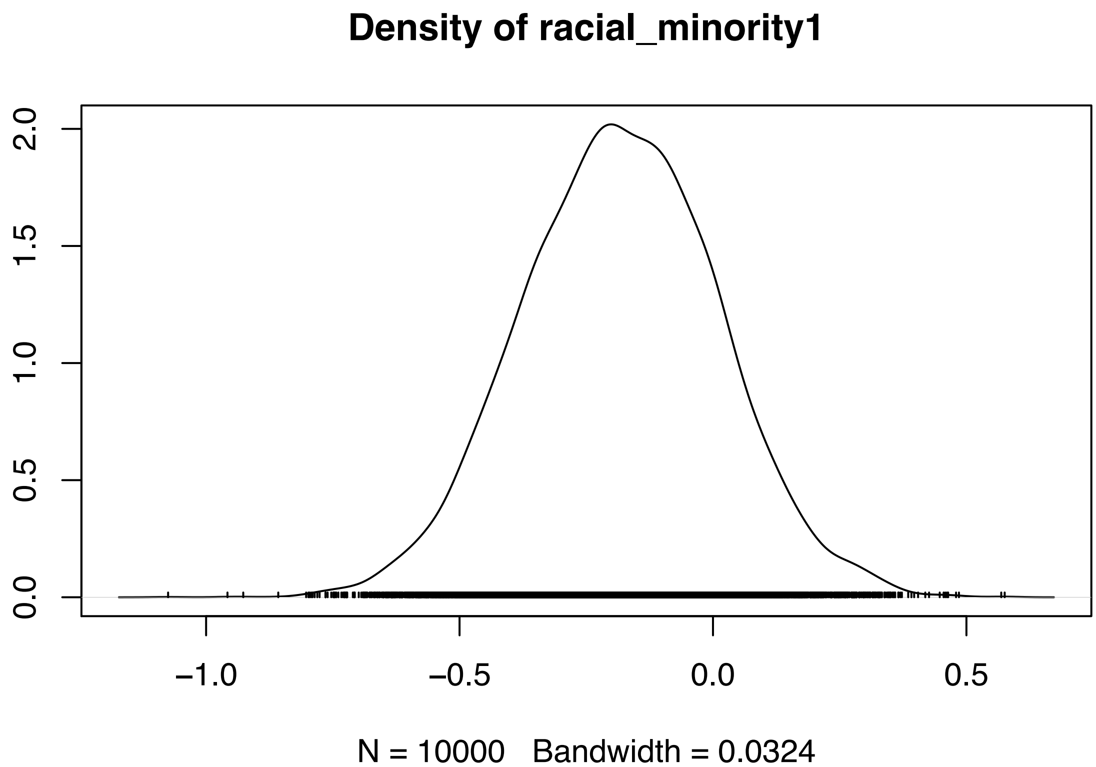
```
##                                2.5%        25%         50%         75%
## (Intercept)               0.0677862  0.559701  0.8273073  1.1021928
## maleTRUE                 -1.2105549 -0.849437 -0.6636819 -0.4686059
## lgbtqTRUE                 0.4285330  0.737927  0.8936220  1.0536681
## racial_minority1         -0.5624145 -0.315432 -0.1833386 -0.0522772
## vic_worst_prior_sexTRUE -1.0109447 -0.796816 -0.6816811 -0.5712171
## vic_worst_verbal          0.0189556  0.168546  0.2434890  0.3204187
## vic_worst_incap           0.3598458  0.473716  0.5330752  0.5931340
## vic_worst_force           0.9301468  1.085198  1.1700653  1.2564103
## week_num                 -0.0847719 -0.053118 -0.0364014 -0.0194960
## week_num_SQ              -0.0015824 -0.001023 -0.0007305 -0.0004334
## post_metoo1              -0.3900161  0.135788  0.4090188  0.6765989
## post_metoo_week          -0.0267459  0.013879  0.0348062  0.0563137
## post_metoo_week_SQ       -0.0001491  0.000486  0.0008107  0.0011393
## sigma2                    3.1514004  3.396580  3.5300145  3.6730515
##                               97.5%
## (Intercept)               1.5955248
## maleTRUE                 -0.1011489
## lgbtqTRUE                 1.3537085
## racial_minority1          0.1959153
## vic_worst_prior_sexTRUE -0.3596687
## vic_worst_verbal          0.4644550
## vic_worst_incap           0.7066714
## vic_worst_force           1.4191519
## week_num                  0.0129870
## week_num_SQ               0.0001507
## post_metoo1               1.1789668
## post_metoo_week           0.0960883
## post_metoo_week_SQ        0.0017904
## sigma2                    3.9739682
```

```
coda::densplot(h2_posteriors$post_metoo_time_SQ)
```

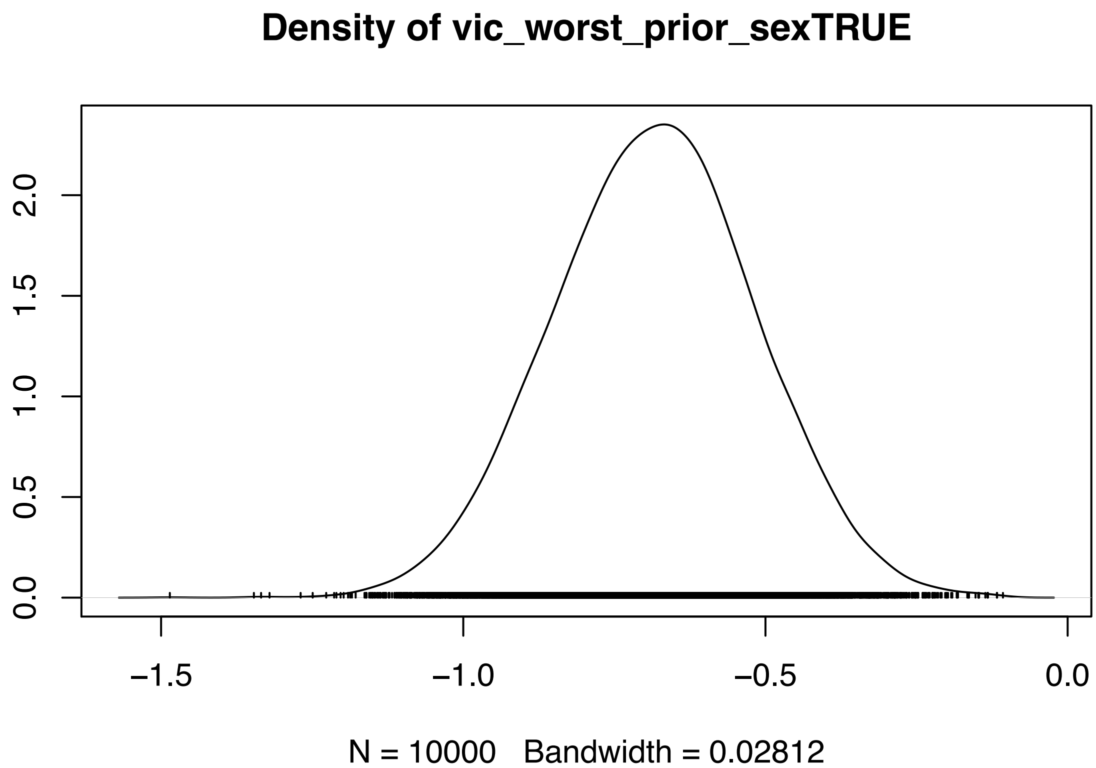

Density of (Intercept)
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
N = 10000   Bandwidth = 0.0665

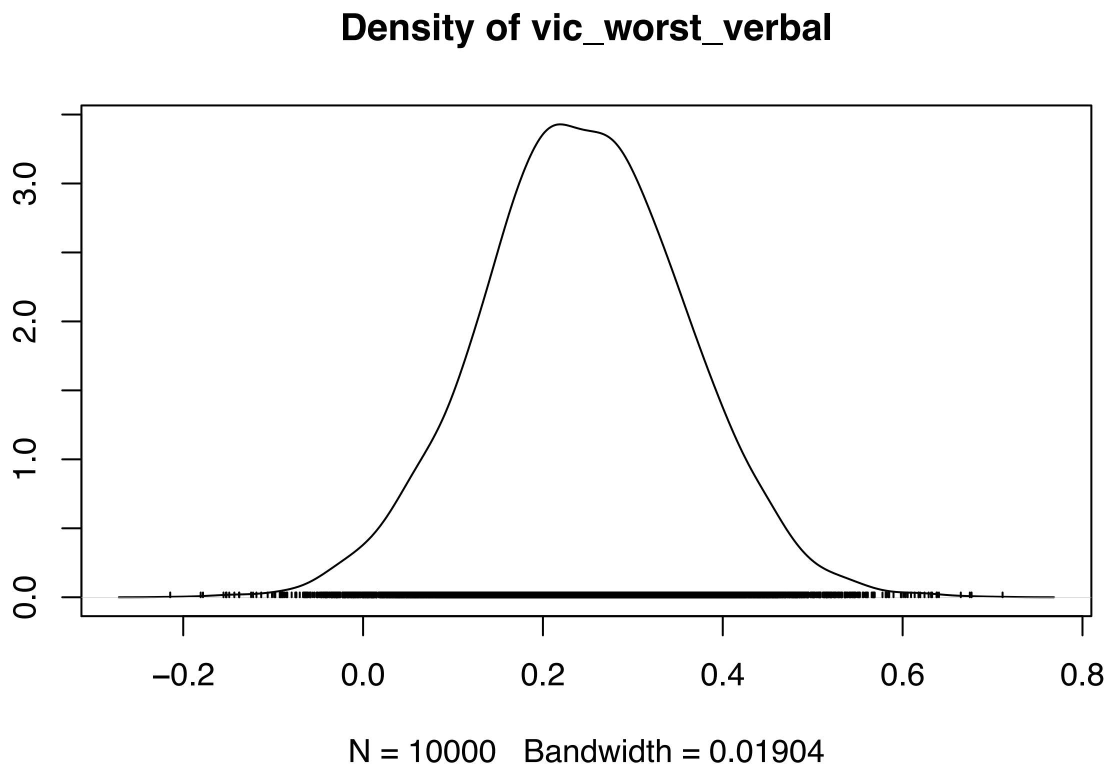

Density of maleTRUE
0.0
0.4
0.8
1.2
−1.5
−1.0
−0.5
0.0
0.5
N = 10000   Bandwidth = 0.04759

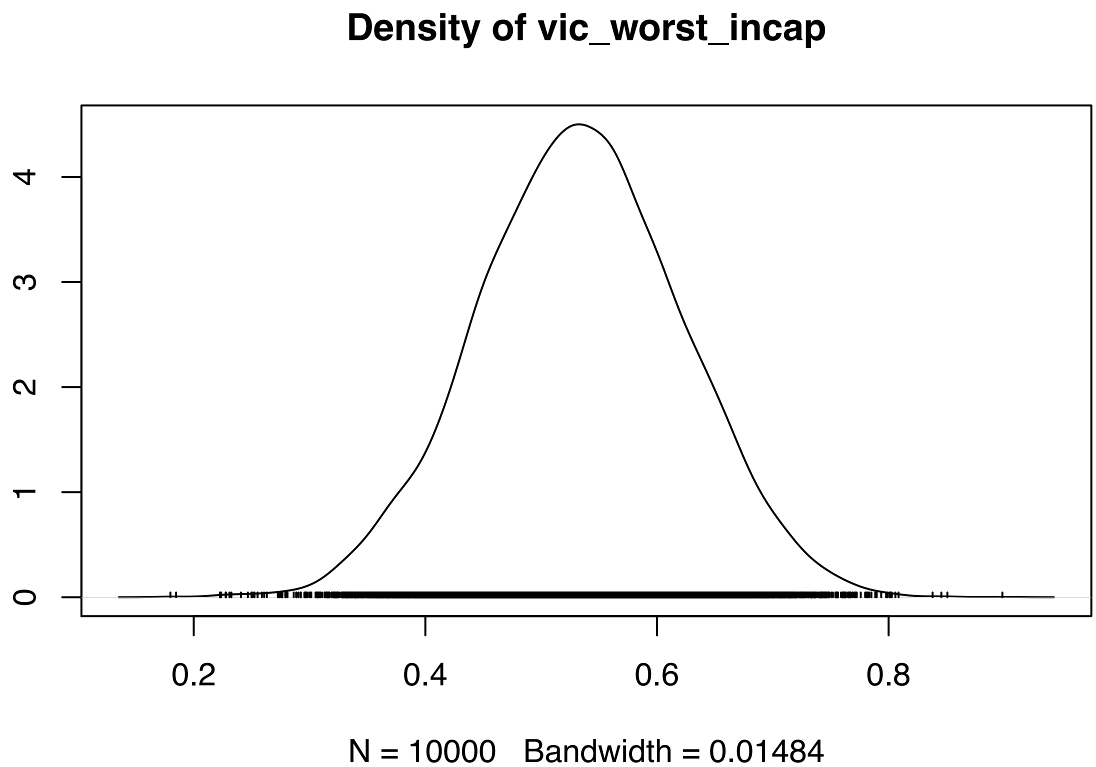

Density of lgbtqTRUE
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
N = 10000   Bandwidth = 0.03946

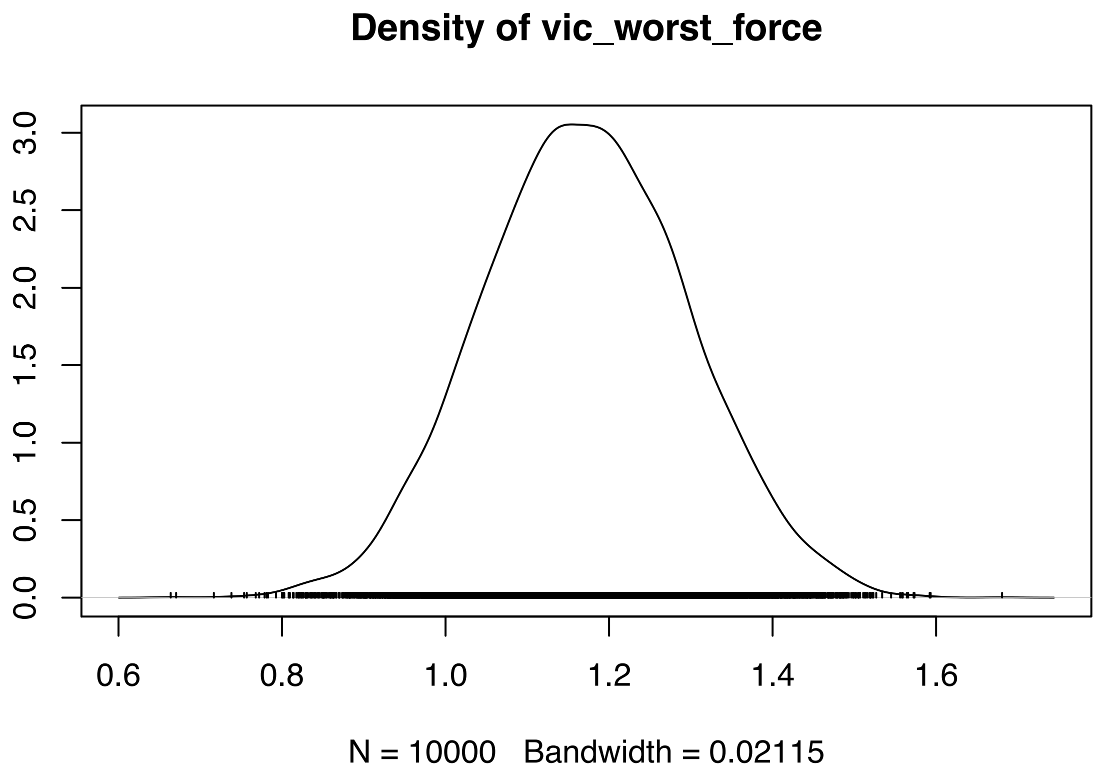


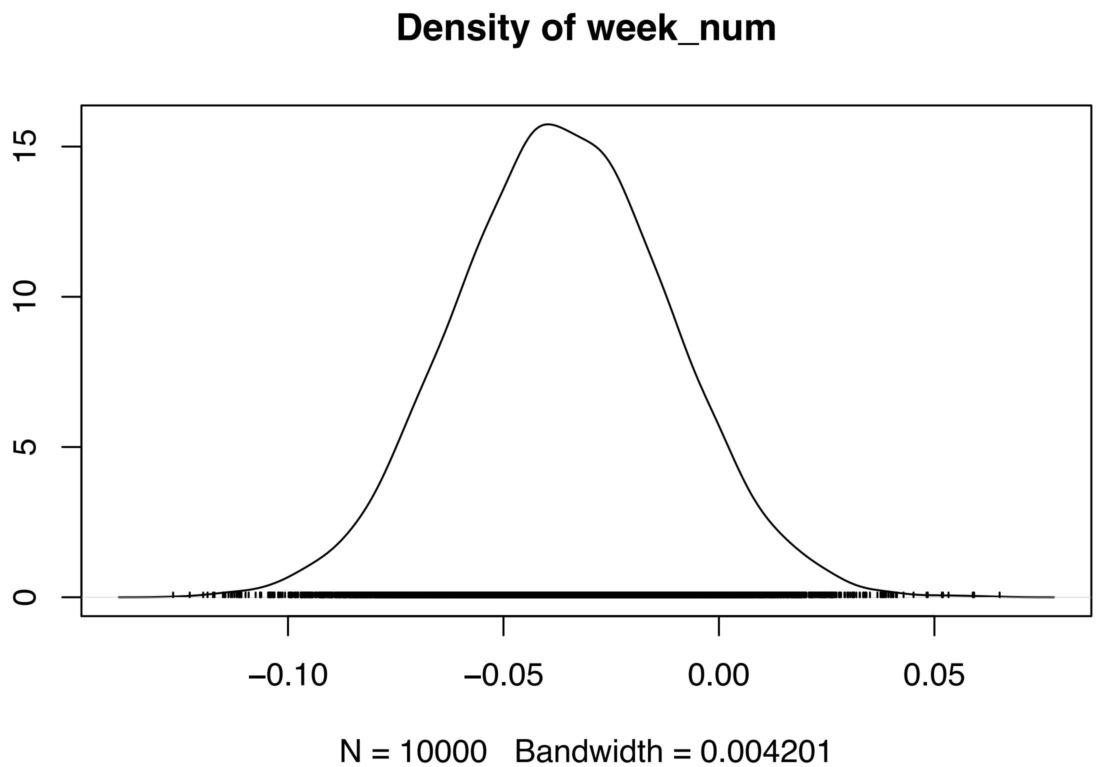
Density of racial_minority1
0.0
0.5
1.0
1.5
2.0
−1.0
−0.5
0.0
0.5
N = 10000   Bandwidth = 0.0324

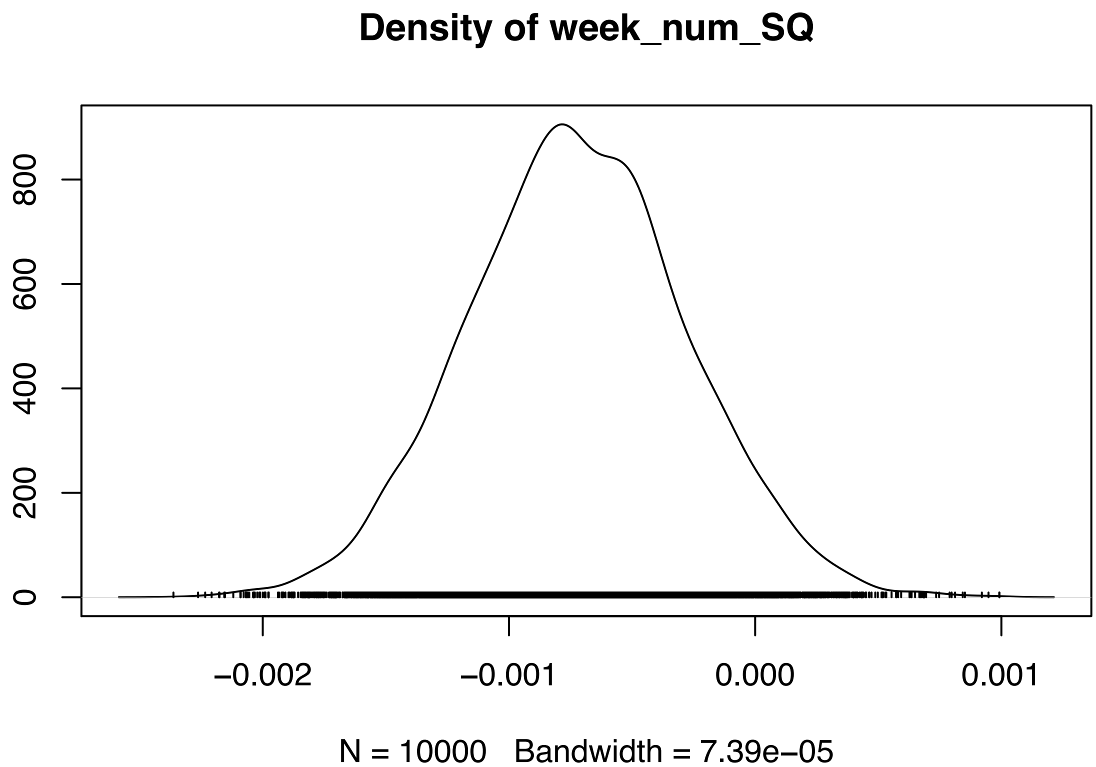

Density of vic_worst_prior_sexTRUE
0.0
0.5
1.0
1.5
2.0
−1.5
−1.0
−0.5
0.0
N = 10000   Bandwidth = 0.02812


Density of vic_worst_verbal
0.0
1.0
2.0
3.0
−0.2
0.0
0.2
0.4
0.6
0.8
N = 10000   Bandwidth = 0.01904

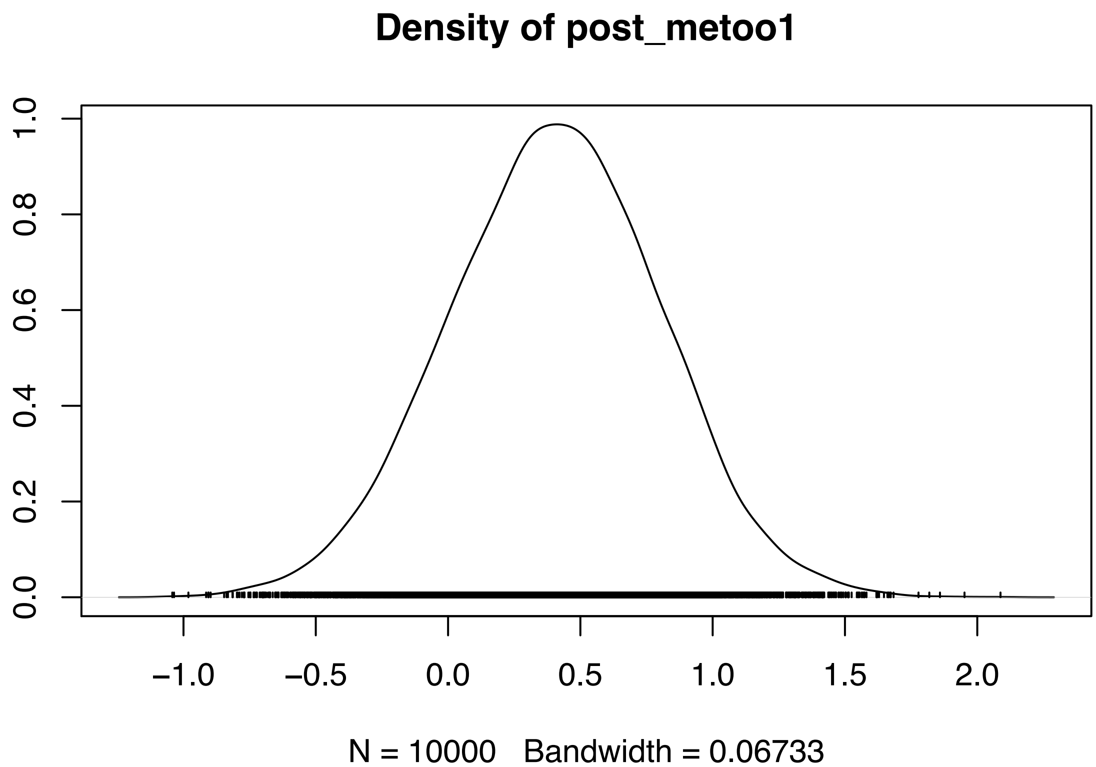

Density of vic_worst_incap
0
1
2
3
4
0.2
0.4
0.6
0.8
N = 10000   Bandwidth = 0.01484

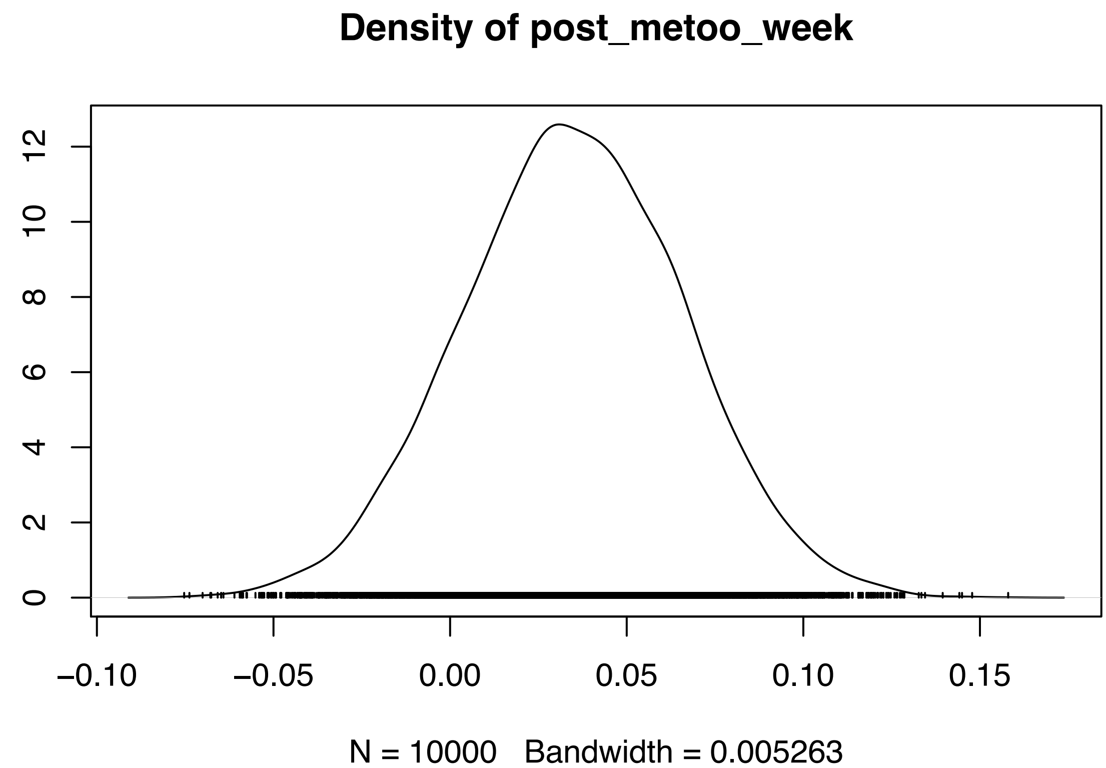

Density of vic_worst_force
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.6
0.8
1.0
1.2
1.4
1.6
N = 10000 Bandwidth = 0.02115

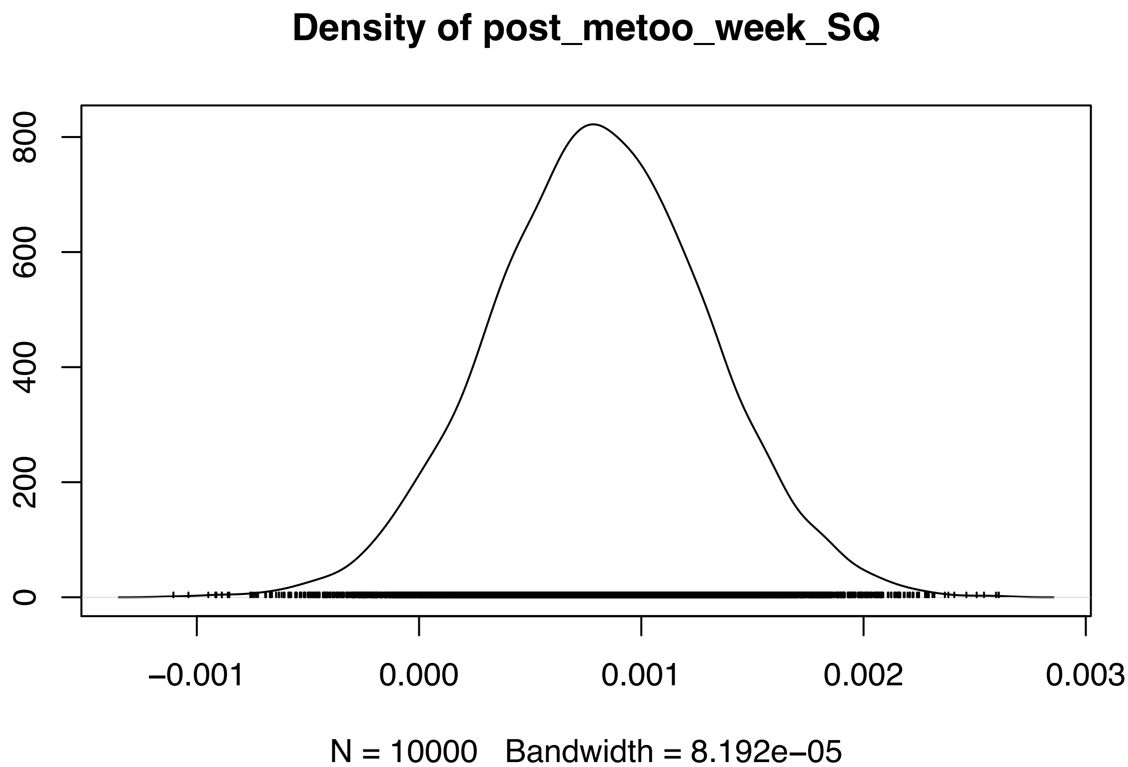

Density of week_num
15
10
5
0
−0.10
−0.05
0.00
0.05
N = 10000   Bandwidth = 0.004201

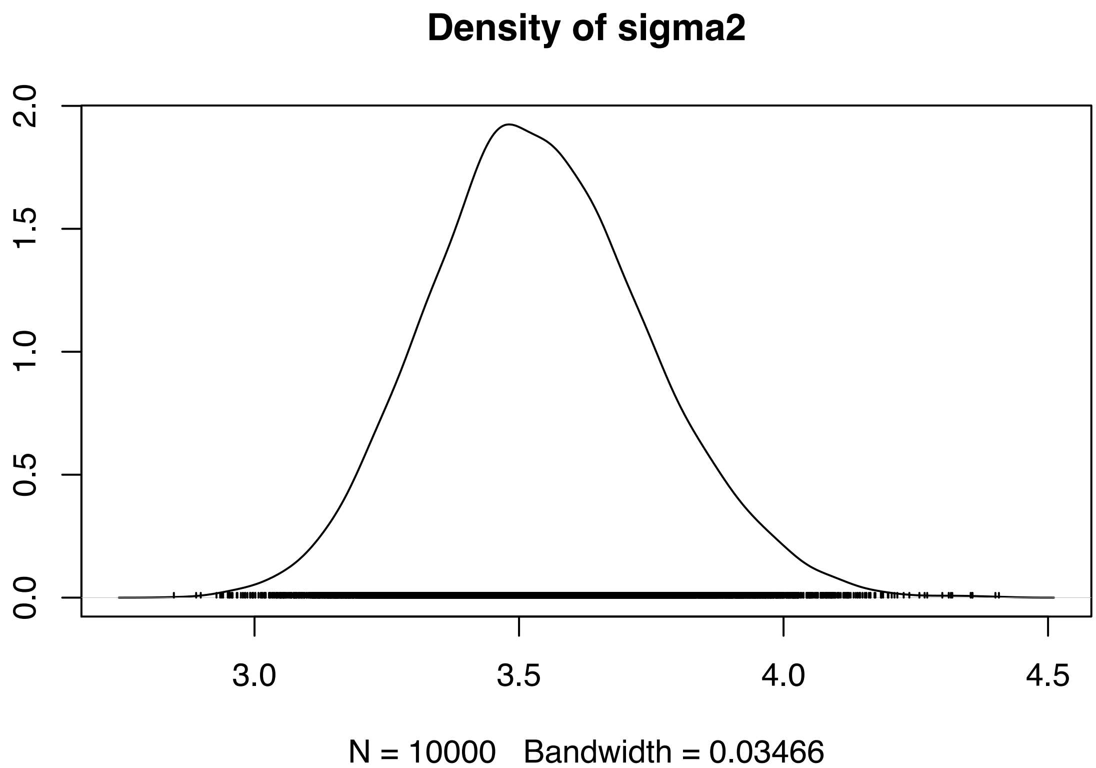


Density of week_num_SQ
800
600
400
200
0
−0.002
−0.001
0.000
0.001
N = 10000 Bandwidth = 7.39e−05


Density of post_metoo1
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
N = 10000   Bandwidth = 0.06733


Density of post_metoo_week
0
2
4
6
8
10
12
−0.10
−0.05
0.00
0.05
0.10
0.15
N = 10000 Bandwidth = 0.005263


Density of post_metoo_week_SQ
800
600
400
200
0
−0.001
0.000
0.001
0.002
0.003
N = 10000   Bandwidth = 8.192e−05




```
h2_param_means <- map(h2_comp_post, colMeans) %>%
  map(data.frame) %>%
  map(~ set_names(.x, 'mean_est')) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  mutate(original_order = row_number())

h2_param_sd <- map(h2_comp_post, colMeans) %>%
  map(data.frame) %>%
  map(~ set_names(.x, 'mean_est')) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  mutate(original_order = row_number())

h2_hpdi_97 <- map(h2_comp_post, HPDinterval, prob = .97) %>%
  map(data.frame) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  dplyr::rename(lower_97 = lower, upper_97 = upper)

h2_hpdi_95 <- map(h2_comp_post, HPDinterval, prob = .95) %>%
  map(data.frame) %>%
  map(~ mutate(.data = .x, param = rownames(.x))) %>%
  bind_rows(.id = 'model') %>%
  dplyr::rename(lower_95 = lower, upper_95 = upper)
```

```r
h2_params <- h2_param_means %>%
  left_join(h2_hpdi_97, by = c('model', 'param')) %>%
  left_join(h2_hpdi_95, by = c('model', 'param'))

h2_params
```

```
##                 model      mean_est                 param original_order
## 1      intercept_only  3.793366e+00           (Intercept)              1
## 2      intercept_only  4.877615e+00                sigma2              2
## 3               demos  3.724704e+00           (Intercept)              3
## 4               demos -7.753399e-01              maleTRUE              4
## 5               demos  1.118866e+00             lgbtqTRUE              5
## 6               demos -3.221098e-02       racial_minority1              6
## 7               demos  4.720490e+00                sigma2              7
## 8   demos_and_context  1.197940e+00           (Intercept)              8
## 9   demos_and_context -5.783731e-01              maleTRUE              9
## 10  demos_and_context  9.080412e-01             lgbtqTRUE             10
## 11  demos_and_context -1.556264e-01       racial_minority1             11
## 12  demos_and_context -6.402269e-01 vic_worst_prior_sexTRUE             12
## 13  demos_and_context  2.442476e-01       vic_worst_verbal             13
## 14  demos_and_context  5.322545e-01        vic_worst_incap             14
## 15  demos_and_context  1.167414e+00        vic_worst_force             15
## 16  demos_and_context  3.578458e+00                sigma2             16
## 17               time  1.163183e+00           (Intercept)             17
## 18               time -5.934384e-01              maleTRUE             18
## 19               time  8.718091e-01             lgbtqTRUE             19
## 20               time -1.817410e-01       racial_minority1             20
## 21               time -6.495053e-01 vic_worst_prior_sexTRUE             21
## 22               time  2.325767e-01       vic_worst_verbal             22
## 23               time  5.330431e-01        vic_worst_incap             23
## 24               time  1.174795e+00        vic_worst_force             24
## 25               time  5.172195e-03              week_num             25
## 26               time  3.541068e+00                sigma2             26
## 27            time_SQ  1.145318e+00           (Intercept)             27
## 28            time_SQ -5.961344e-01              maleTRUE             28
## 29            time_SQ  8.665201e-01             lgbtqTRUE             29
## 30            time_SQ -1.808430e-01       racial_minority1             30
## 31            time_SQ -6.492758e-01 vic_worst_prior_sexTRUE             31
## 32            time_SQ  2.329441e-01       vic_worst_verbal             32
## 33            time_SQ  5.358449e-01        vic_worst_incap             33
## 34            time_SQ  1.173081e+00        vic_worst_force             34
## 35            time_SQ  4.934240e-03              week_num             35
## 36            time_SQ  9.530423e-06           week_num_SQ             36
## 37            time_SQ  3.545892e+00                sigma2             37
## 38        post_metoo  1.147504e+00           (Intercept)             38
## 39        post_metoo -5.919566e-01              maleTRUE             39
## 40        post_metoo  8.686691e-01             lgbtqTRUE             40
## 41        post_metoo -1.824841e-01       racial_minority1             41
## 42        post_metoo -6.466003e-01 vic_worst_prior_sexTRUE             42
## 43        post_metoo  2.342714e-01       vic_worst_verbal             43
## 44        post_metoo  5.347992e-01        vic_worst_incap             44
## 45        post_metoo  1.171170e+00        vic_worst_force             45
## 46        post_metoo  4.948438e-03              week_num             46
```

```
## 47         post_metoo  8.790726e-06            week_num_SQ                47
## 48         post_metoo -1.883065e-03            post_metoo1                48
## 49         post_metoo  3.549096e+00                 sigma2                49
## 50    post_metoo_time  1.098153e+00            (Intercept)                50
## 51    post_metoo_time -5.746679e-01               maleTRUE                51
## 52    post_metoo_time  8.637244e-01              lgbtqTRUE                52
## 53    post_metoo_time -1.756329e-01       racial_minority1                53
## 54    post_metoo_time -6.463240e-01 vic_worst_prior_sexTRUE               54
## 55    post_metoo_time  2.318144e-01        vic_worst_verbal               55
## 56    post_metoo_time  5.302168e-01         vic_worst_incap               56
## 57    post_metoo_time  1.174840e+00         vic_worst_force               57
## 58    post_metoo_time -3.516417e-04               week_num                58
## 59    post_metoo_time -7.506711e-05            week_num_SQ                59
## 60    post_metoo_time -4.449188e-02            post_metoo1                60
## 61    post_metoo_time  1.256406e-02         post_metoo_week               61
## 62    post_metoo_time  3.553901e+00                 sigma2                62
## 63 post_metoo_time_SQ  8.321770e-01            (Intercept)                63
## 64 post_metoo_time_SQ -5.749980e-01               maleTRUE                64
## 65 post_metoo_time_SQ  8.618804e-01              lgbtqTRUE                65
## 66 post_metoo_time_SQ -1.815823e-01       racial_minority1                66
## 67 post_metoo_time_SQ -6.461953e-01 vic_worst_prior_sexTRUE               67
## 68 post_metoo_time_SQ  2.420592e-01        vic_worst_verbal               68
## 69 post_metoo_time_SQ  5.342429e-01         vic_worst_incap               69
## 70 post_metoo_time_SQ  1.177701e+00         vic_worst_force               70
## 71 post_metoo_time_SQ -3.549490e-02               week_num                71
## 72 post_metoo_time_SQ -7.276398e-04            week_num_SQ                72
## 73 post_metoo_time_SQ  3.389893e-01            post_metoo1                73
## 74 post_metoo_time_SQ  3.548293e-02         post_metoo_week               74
## 75 post_metoo_time_SQ  8.005857e-04      post_metoo_week_SQ               75
## 76 post_metoo_time_SQ  3.540552e+00                 sigma2                76
##         lower_97      upper_97      lower_95      upper_95
## 1   3.595397e+00  3.9923419929  3.609373e+00  3.9692691116
## 2   4.283655e+00  5.5250454563  4.320346e+00  5.4363445390
## 3   3.490107e+00  3.9694850016  3.508518e+00  3.9421184058
## 4  -1.469636e+00 -0.0815043853 -1.401183e+00 -0.1585566287
## 5   5.449931e-01  1.7031045017  5.795262e-01  1.6309328932
## 6  -5.150259e-01  0.4380258363 -4.559557e-01  0.4035402401
## 7   4.121814e+00  5.3353132132  4.174995e+00  5.2614568712
## 8   5.664221e-01  1.8015709091  6.228238e-01  1.7551876696
## 9  -1.228159e+00  0.0273697908 -1.157091e+00 -0.0288148735
## 10  4.015918e-01  1.4336852031  4.523162e-01  1.3756080356
## 11 -5.577394e-01  0.2680996567 -5.270874e-01  0.2204142483
## 12 -9.910790e-01 -0.2650852725 -9.743571e-01 -0.3230476440
## 13  2.507211e-03  0.4913711906  3.075708e-02  0.4750007915
## 14  3.461168e-01  0.7271641225  3.529341e-01  0.6978362449
## 15  8.947502e-01  1.4451671377  9.150060e-01  1.4116946985
## 16  3.114359e+00  4.0320463082  3.181807e+00  4.0106290433
## 17  5.265396e-01  1.8045676997  5.961834e-01  1.7464867162
## 18 -1.208659e+00  0.0173793405 -1.145612e+00 -0.0400976448
## 19  3.664116e-01  1.3922758599  4.033380e-01  1.3272063541
## 20 -5.897806e-01  0.2314314566 -5.470690e-01  0.1925110990
## 21 -9.900894e-01 -0.2672117237 -9.675444e-01 -0.3199687114
## 22 -2.480403e-02  0.4735678972  4.226998e-03  0.4534086952
## 23  3.406722e-01  0.7305106329  3.514120e-01  0.7023472250
```

```
## 24  9.047468e-01  1.4486851762  9.242989e-01  1.4180741842
## 25  1.128145e-03  0.0092431131  1.448875e-03  0.0088051190
## 26  3.095546e+00  4.0122961542  3.131023e+00  3.9628731471
## 27  4.577362e-01  1.7602012678  5.659763e-01  1.7404562907
## 28 -1.189769e+00  0.0443355477 -1.118275e+00 -0.0059373588
## 29  3.416386e-01  1.3510593312  4.413339e-01  1.3423427598
## 30 -6.099564e-01  0.2128147095 -5.361516e-01  0.2095838615
## 31 -1.015465e+00 -0.2915509857 -9.678972e-01 -0.3123163078
## 32 -1.413289e-02  0.4858169895  1.438567e-02  0.4580901631
## 33  3.504726e-01  0.7351151273  3.663336e-01  0.7126602814
## 34  9.015375e-01  1.4505413457  9.246369e-01  1.4264650477
## 35  1.526988e-04  0.0098714358  6.477746e-04  0.0094347038
## 36 -9.741973e-05  0.0001158848 -8.796369e-05  0.0001041760
## 37  3.115256e+00  4.0482772082  3.123060e+00  3.9661856640
## 38  4.272145e-01  1.8889352147  4.787724e-01  1.7847078655
## 39 -1.203420e+00  0.0358071909 -1.123182e+00 -0.0060633287
## 40  3.461999e-01  1.3837353700  4.028743e-01  1.3263394417
## 41 -6.110770e-01  0.2354334799 -5.704930e-01  0.1873784586
## 42 -9.939175e-01 -0.2799219787 -9.792464e-01 -0.3342702559
## 43 -2.192261e-02  0.4815704406 -5.121179e-03  0.4515528519
## 44  3.453593e-01  0.7256130489  3.679052e-01  0.7131005301
## 45  9.062909e-01  1.4534120925  9.286032e-01  1.4212133539
## 46 -3.952112e-03  0.0140854048 -3.450538e-03  0.0129706425
## 47 -1.058441e-04  0.0001234489 -9.822715e-05  0.0001113750
## 48 -6.774438e-01  0.6674761808 -5.980444e-01  0.6108886145
## 49  3.105310e+00  4.0091568066  3.159567e+00  3.9754926900
## 50  2.812141e-01  1.8825840000  3.903436e-01  1.8256135675
## 51 -1.197199e+00  0.0580890929 -1.154370e+00 -0.0069186357
## 52  3.747055e-01  1.3900243447  3.834387e-01  1.3010919996
## 53 -5.875943e-01  0.2454246983 -5.583857e-01  0.2036097077
## 54 -1.021670e+00 -0.2897862180 -9.717779e-01 -0.3117497348
## 55 -2.005138e-02  0.4878068406  1.103638e-02  0.4686111053
## 56  3.450681e-01  0.7285504677  3.578463e-01  0.7093604811
## 57  8.964305e-01  1.4418058112  9.302642e-01  1.4264280205
## 58 -2.739358e-02  0.0264897289 -2.397863e-02  0.0249543729
## 59 -5.021014e-04  0.0003500008 -4.577510e-04  0.0003110959
## 60 -7.404560e-01  0.6408607868 -6.598600e-01  0.5865775845
## 61 -4.618784e-02  0.0746373337 -4.256796e-02  0.0671202577
## 62  3.102398e+00  4.0485957993  3.142366e+00  3.9849524977
## 63 -1.106045e-02  1.7042608063  3.601376e-02  1.5968292791
## 64 -1.185816e+00  0.0588546682 -1.127500e+00  0.0029290324
## 65  3.362588e-01  1.3668408351  3.936576e-01  1.3238052455
## 66 -5.915063e-01  0.2426707897 -5.471268e-01  0.2086451714
## 67 -1.011331e+00 -0.2936467910 -9.641094e-01 -0.3192015770
## 68 -3.845585e-03  0.4935195324  2.412471e-02  0.4717972068
## 69  3.444411e-01  0.7243488050  3.691934e-01  0.7137884304
## 70  9.086130e-01  1.4598977149  9.220604e-01  1.4242891129
## 71 -8.790990e-02  0.0194771468 -8.659981e-02  0.0108990988
## 72 -1.658780e-03  0.0002612401 -1.605542e-03  0.0001342228
## 73 -5.197759e-01  1.1952091994 -4.494992e-01  1.1171694700
## 74 -3.429720e-02  0.1015448446 -2.545126e-02  0.0977908598
## 75 -2.352638e-04  0.0019041890 -1.588900e-04  0.0017658348
## 76  3.114227e+00  4.0303965893  3.136825e+00  3.9698779246
```

```r
write.csv(h2_params, paste0('h2_params_', Sys.Date(), '.csv'), row.names = F)
```

Lastly, we write the table to a .csv file for later.

```r
h2_model_table <- h2_bayes_factors

h2_model_table
```

```
##                model  logmarglike
## 1     intercept_only -1,288.69005
## 2              demos -1,288.65833
## 3  demos_and_context -1,210.58796
## 4               time -1,215.35232
## 5            time_SQ -1,227.55852
## 6         post_metoo -1,231.02514
## 7    post_metoo_time -1,236.78897
## 8 post_metoo_time_SQ -1,245.38177
##                                                   BF
## 1                                                 NA
## 2                                            1.03222
## 3 8,045,103,140,013,963,422,936,160,087,834,624.00000
## 4                                            0.00853
## 5                                            0.00000
## 6                                            0.03122
## 7                                            0.00314
## 8                                            0.00019
```

```r
write.csv(
  x = h2_model_table,
  file = paste0('h2_model_table', Sys.Date(), '.csv'),
  row.names = F)
```

Model MCMC Chain Convergence

Model convergence is assessed using the first multiply imputed dataset for each hypothesis, using standard techniques from the **coda** package.

H1 Convergence

```r
Sys.time()
```

```
## [1] "2019-11-17 01:49:17 UTC"
```

```r
library(MCMCpack)

test_df <- full_imp[[1]]

test_chains <- map(
  .x = h1_models,
  .f = ~ rerun(4, run_logistic_mcmc(
    df = test_df,
    model = .x,
    prior_sd = h1_priors,
    mcmc = 6*1e4,
    thin = 2)) %>%
    as.mcmc.list()) %>%
  set_names(names(h1_models))

Sys.time()
```

```
## [1] "2019-11-17 01:55:29 UTC"
```

```r
# coda::traceplot(test_chains$post_metoo_time_SQ)
```

```r
1 - coda::rejectionRate(test_chains$post_metoo_time_SQ)
```

```
##        (Intercept)           maleTRUE          lgbtqTRUE
##          0.2466332          0.2466332          0.2466332
##    racial_minority1           week_num        week_num_SQ
##          0.2466332          0.2466332          0.2466332
##        post_metoo1    post_metoo_week post_metoo_week_SQ
##          0.2466332          0.2466332          0.2466332
```

```r
coda::gelman.diag(test_chains$post_metoo_time_SQ)
```

```
## Potential scale reduction factors:
##
##                    Point est. Upper C.I.
## (Intercept)                 1       1.00
## maleTRUE                    1       1.01
## lgbtqTRUE                   1       1.00
## racial_minority1            1       1.00
## week_num                    1       1.00
## week_num_SQ                 1       1.00
## post_metoo1                 1       1.01
## post_metoo_week             1       1.00
## post_metoo_week_SQ          1       1.00
##
## Multivariate psrf
##
## 1
```

H2 Convergence

```r
test_df <- vic_imp[[1]]

test_chains <- map(
  .x = h2_models,
  .f = ~ rerun(4, run_linear_mcmc(
    df = test_df,
    model = .x,
    prior_sd = h2_priors,
    mcmc = 1e5,
    thin = 1)) %>%
    as.mcmc.list()) %>%
  set_names(names(h2_models))
```

```r
# coda::traceplot(test_chains$post_metoo_time_SQ)
```

Note, `coda::rejectionRate` is likely producing errors here. See helpfile, where it describes doing a naive test of two successive iterations of the sampler (using `==`). This likely produces rounding errors, causing the rejection rate to look like 0. This happened on simulated data too, so it is not just our data that is producing this odd output. However, all other diagnostics for our chains look healthy, so we proceed to estimating the rest of these models.

```r
map(test_chains[2:length(test_chains)], ~ 1 - coda::rejectionRate(.x))
```

```
## $demos
##      (Intercept)         maleTRUE        lgbtqTRUE racial_minority1
##                1                1                1                1
##           sigma2
##                1
##
## $demos_and_context
##             (Intercept)                maleTRUE               lgbtqTRUE
##                       1                       1                       1
##        racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                       1                       1                       1
##         vic_worst_incap         vic_worst_force                  sigma2
##                       1                       1                       1
##
## $time
##             (Intercept)                maleTRUE               lgbtqTRUE
##                       1                       1                       1
##        racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                       1                       1                       1
##         vic_worst_incap         vic_worst_force                week_num
##                       1                       1                       1
##                  sigma2
##                       1
##
## $time_SQ
##             (Intercept)                maleTRUE               lgbtqTRUE
##                       1                       1                       1
```

```
##       racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                      1                       1                       1
##         vic_worst_incap         vic_worst_force                week_num
##                      1                       1                       1
##             week_num_SQ                  sigma2
##                      1                       1
##
## $post_metoo
##             (Intercept)                maleTRUE               lgbtqTRUE
##                      1                       1                       1
##       racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                      1                       1                       1
##         vic_worst_incap         vic_worst_force                week_num
##                      1                       1                       1
##             week_num_SQ             post_metoo1                  sigma2
##                      1                       1                       1
##
## $post_metoo_time
##             (Intercept)                maleTRUE               lgbtqTRUE
##                      1                       1                       1
##       racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                      1                       1                       1
##         vic_worst_incap         vic_worst_force                week_num
##                      1                       1                       1
##             week_num_SQ             post_metoo1         post_metoo_week
##                      1                       1                       1
##                  sigma2
##                      1
##
## $post_metoo_time_SQ
##             (Intercept)                maleTRUE               lgbtqTRUE
##                      1                       1                       1
##       racial_minority1 vic_worst_prior_sexTRUE        vic_worst_verbal
##                      1                       1                       1
##         vic_worst_incap         vic_worst_force                week_num
##                      1                       1                       1
##             week_num_SQ             post_metoo1         post_metoo_week
##                      1                       1                       1
##      post_metoo_week_SQ                  sigma2
##                      1                       1
```

```
map(test_chains[2:length(test_chains)], coda::gelman.diag)
```

```
## $demos
## Potential scale reduction factors:
##
##                  Point est. Upper C.I.
## (Intercept)               1          1
## maleTRUE                  1          1
## lgbtqTRUE                 1          1
## racial_minority1          1          1
## sigma2                    1          1
##
## Multivariate psrf
```

```
##
## 1
##
## $demos_and_context
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                      1          1
## maleTRUE                         1          1
## lgbtqTRUE                        1          1
## racial_minority1                 1          1
## vic_worst_prior_sexTRUE          1          1
## vic_worst_verbal                 1          1
## vic_worst_incap                  1          1
## vic_worst_force                  1          1
## sigma2                           1          1
##
## Multivariate psrf
##
## 1
##
## $time
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                      1          1
## maleTRUE                         1          1
## lgbtqTRUE                        1          1
## racial_minority1                 1          1
## vic_worst_prior_sexTRUE          1          1
## vic_worst_verbal                 1          1
## vic_worst_incap                  1          1
## vic_worst_force                  1          1
## week_num                         1          1
## sigma2                           1          1
##
## Multivariate psrf
##
## 1
##
## $time_SQ
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                      1          1
## maleTRUE                         1          1
## lgbtqTRUE                        1          1
## racial_minority1                 1          1
## vic_worst_prior_sexTRUE          1          1
## vic_worst_verbal                 1          1
## vic_worst_incap                  1          1
## vic_worst_force                  1          1
## week_num                         1          1
## week_num_SQ                      1          1
```

```
## sigma2                        1          1
##
## Multivariate psrf
##
## 1
##
## $post_metoo
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                     1          1
## maleTRUE                        1          1
## lgbtqTRUE                       1          1
## racial_minority1                1          1
## vic_worst_prior_sexTRUE         1          1
## vic_worst_verbal                1          1
## vic_worst_incap                 1          1
## vic_worst_force                 1          1
## week_num                        1          1
## week_num_SQ                     1          1
## post_metoo1                     1          1
## sigma2                          1          1
##
## Multivariate psrf
##
## 1
##
## $post_metoo_time
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                     1          1
## maleTRUE                        1          1
## lgbtqTRUE                       1          1
## racial_minority1                1          1
## vic_worst_prior_sexTRUE         1          1
## vic_worst_verbal                1          1
## vic_worst_incap                 1          1
## vic_worst_force                 1          1
## week_num                        1          1
## week_num_SQ                     1          1
## post_metoo1                     1          1
## post_metoo_week                 1          1
## sigma2                          1          1
##
## Multivariate psrf
##
## 1
##
## $post_metoo_time_SQ
## Potential scale reduction factors:
##
##                         Point est. Upper C.I.
## (Intercept)                     1          1
```

```
## maleTRUE                          1          1
## lgbtqTRUE                         1          1
## racial_minority1                  1          1
## vic_worst_prior_sexTRUE           1          1
## vic_worst_verbal                  1          1
## vic_worst_incap                   1          1
## vic_worst_force                   1          1
## week_num                          1          1
## week_num_SQ                       1          1
## post_metoo1                       1          1
## post_metoo_week                   1          1
## post_metoo_week_SQ                1          1
## sigma2                            1          1
##
## Multivariate psrf
##
## 1
```

MeToo Movement and Perceptions of Sexual Assault: Section 2

Contents

Introduction

R Version:

x86_64-pc-linux-gnu, x86_64, linux-gnu, x86_64, linux-gnu, , 3, 6.3, 2020, 02, 29, 77875, R, R version 3.6.3 (2020-02-29), Holding the Windsock

Run date: 2020-05-20 10:15:17

What follows is the second section of annotated code for *The #MeToo Movement and Perceptions of Sexual Assault: College Students' Recognition of Sexual Assault Experiences Over Time.* This includes all processing and summarization of results, *after* models had been estimated.

Setup - load participant dataframe

```
knitr::knit('data_wrangling.Rmd', output = tempfile())
```

```
##
##
## processing file: data_wrangling.Rmd

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## output file: /tmp/RtmpaWsX4S/file30e97968af34
```

Figure - change over time

Create a dataframe for plotting participant probability of screening positive (*p_vic*) and mean sexual assault acknowledgment (*mean_sa*).

```
library(tidyverse)

fig_df <- df %>%
  group_by(study_week) %>%
  summarise(
    post_metoo = unique(post_metoo),
    n = n(),
    p_vic = sum(vic_screener == 'Yes', na.rm = T)/n,
    mean_sa = mean(vic_worst_is_sa, na.rm = T)) %>%
  mutate(post_metoo = ifelse(study_week >= '2017-10-16', 'Yes', 'No'))

fig_df
```

```
## # A tibble: 90 x 5
##    study_week post_metoo     n  p_vic mean_sa
##    <date>     <chr>      <int>  <dbl>   <dbl>
##  1 2016-09-04 No            32 0.0312    6
##  2 2016-09-11 No            18 0.333     4
##  3 2016-09-18 No            14 0.214     5.33
##  4 2016-09-25 No            33 0.242     2.88
##  5 2016-10-02 No            37 0.108     1.75
##  6 2016-10-09 No            23 0.174     2
##  7 2016-10-16 No            14 0.214     3.67
##  8 2016-10-23 No            17 0.294     5.8
##  9 2016-10-30 No            19 0.368     3.71
## 10 2016-11-06 No            26 0.269     4.33
## # ... with 80 more rows
```

Plot screener and acknowledgment over time.

```
post_metoo_plot_df <- participant_df %>%
  group_by(post_metoo) %>%
  summarise(
    begin = ymd(as.Date(min(study_week))),
    end = ymd(as.Date(max(study_week))),
    mid = begin + (end - begin)/2,
    p_vic = mean(vic_screener == 'Yes', na.rm = T),
    mean_sa = mean(vic_worst_is_sa, na.rm = T),
    n = n()) %>%
  gather(point_type, p, -post_metoo, -begin, -end, -mid, -n) %>%
  mutate(
```

```
    post_metoo = ifelse(post_metoo == 0, 'No', 'Yes'),
    point_type = factor(
      x = point_type,
      levels = c( 'p_vic', 'mean_sa'),
      labels = c(
        'Sexual Assault Prevalence', 'Mean Acknowledgment')))

post_metoo_plot_df
```

```
## # A tibble: 4 x 7
##   post_metoo begin      end        mid            n point_type             p
##   <chr>      <date>     <date>     <date>     <int> <fct>              <dbl>
## 1 No         2016-09-04 2017-10-08 2017-03-23  1086 Sexual Assault P~ 0.241
## 2 Yes        2017-10-15 2019-04-21 2018-07-18  1530 Sexual Assault P~ 0.239
## 3 No         2016-09-04 2017-10-08 2017-03-23  1086 Mean Acknowledgm~ 3.55
## 4 Yes        2017-10-15 2019-04-21 2018-07-18  1530 Mean Acknowledgm~ 4.00
```

```
p <- fig_df %>%
  filter(is.na(n) | n >= 5) %>%
  gather(point_type, p, -study_week, -n, -post_metoo) %>%
  mutate(
    n = ifelse(point_type == 'sa', 10, n),
    point_type = factor(
      x = point_type,
      levels = c( 'p_vic', 'mean_sa'),
      labels = c(
        'Sexual Assault Prevalence', 'Mean Acknowledgment'))) %>%
  ggplot(aes(study_week, p)) +
  theme_light() +
  scale_color_manual(values = c('grey60', 'grey20')) +
  facet_grid(point_type ~ ., scales = 'free') +
  geom_vline(xintercept = ymd('2017-10-15'), color = 'grey80', size = 2, linetype = 1) +
    # geom_line() +
  # geom_line(data = indiv_fig_df, color = 'grey20', linetype = 1, size = 1.5) +
    geom_point(aes(color = factor(post_metoo), size = n), alpha = .50) +
    # geom_smooth(span = .54, color = 'black', weights = n) +
  geom_segment(
    data = post_metoo_plot_df,
    mapping = aes(
      x = begin,
      xend = end,
      y = p,
      yend = p,
      color = factor(post_metoo)),
      linetype = 2
    ) +
  # geom_segment(
  #   data = sem_plot_df,
  #   mapping = aes(
  #     x = begin,
  #     xend = end,
  #     y = p,
  #     yend = p)) +
```

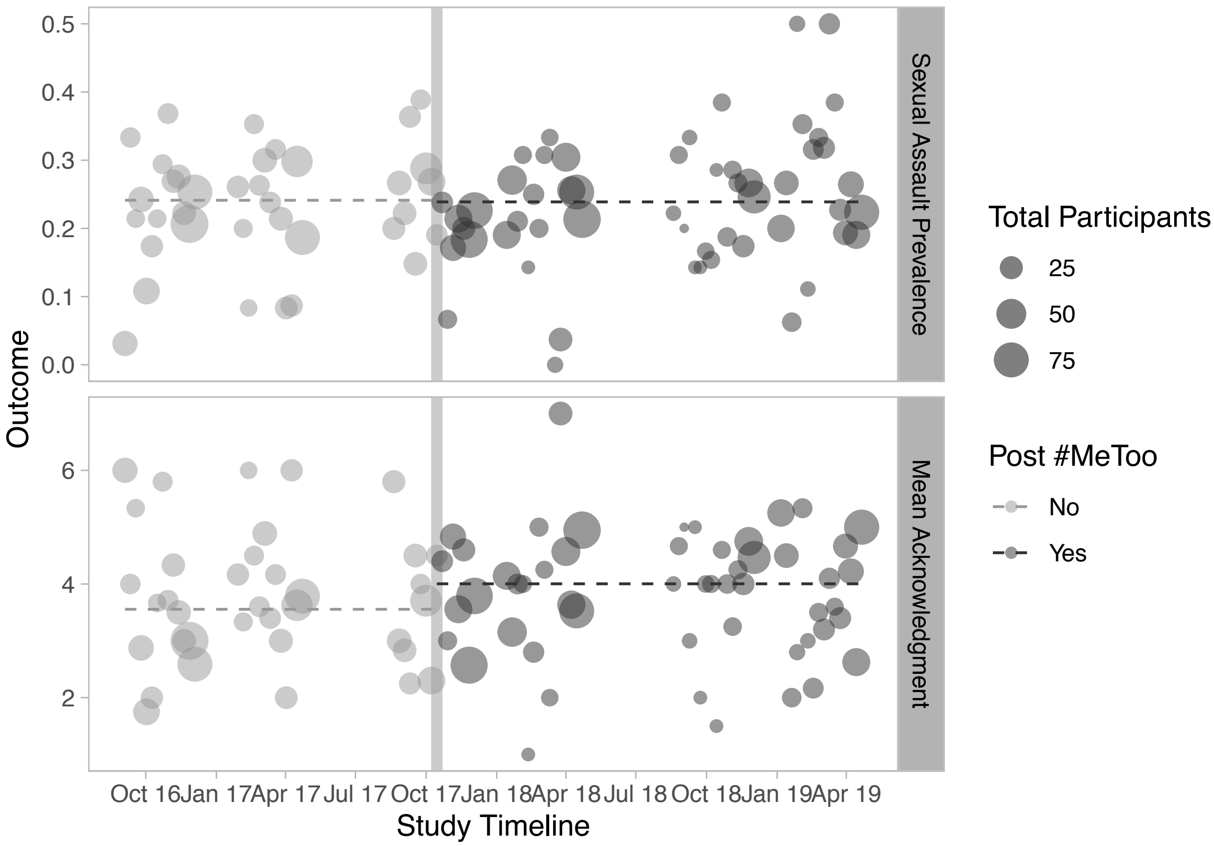
```r
  # geom_point(data = sem_plot_df, mapping = aes(x = mid), size = 3) +
  # geom_line(data = sem_plot_df, mapping = aes(x = mid)) +
  scale_x_date(
    date_breaks = '3 months', date_labels = '%b %y', date_minor_breaks = '1 month') +
  labs(
    y = 'Outcome',
    x = 'Study Timeline') +
  guides(
    color = guide_legend(title = 'Post #MeToo'),
    size = guide_legend(title = 'Total Participants')) +
  theme(
    strip.text = element_text(color = 'black'),
    panel.border = element_rect(color = 'grey'),
    strip.background = element_rect(color  = 'grey'),
    panel.grid = element_blank())

ggsave(
  filename = paste0('sa_endorsement_plot_', Sys.Date(), '.png'),
  plot = p,
  height = 6,
  width = 8,
  dpi = 800)
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```r
p
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Dummy coding table

Produce a dummy-coding table so readers can see how time and the MeToo interaction was coded.

```
coding_example_table <- df %>%
  group_by(study_week) %>%
  summarise(
    week_num = min(week_num),
    week_num_SQ = min(week_num_SQ),
    post_metoo = unique(post_metoo),
    post_metoo_week = min(post_metoo_week),
    post_metoo_week_SQ = min(post_metoo_week_SQ)) %>%
  filter(week_num >= 55 & week_num <= 60)

coding_example_table
```

```
## # A tibble: 5 x 6
##   study_week week_num week_num_SQ post_metoo post_metoo_week
##   <date>        <dbl>       <dbl>      <dbl>           <dbl>
## 1 2018-11-04       55        3025          1              55
## 2 2018-11-11       56        3136          1              56
## 3 2018-11-18       57        3249          1              57
## 4 2018-11-25       58        3364          1              58
## 5 2018-12-02       59        3481          1              59
## # ... with 1 more variable: post_metoo_week_SQ <dbl>
```

```
write.csv(coding_example_table, 'coding_example_table.csv', row.names = F)
```

Tables for posterior distribution model parameters

Import posterior distributions from model fitting in Part 1.

```
latest_h1_posterior <- max(list.files()[grepl('h1_posteriors', list.files())])
latest_h2_posterior <- max(list.files()[grepl('h2_posteriors', list.files())])

h1_post <- readr::read_csv(latest_h1_posterior)
```

```
## Parsed with column specification:
## cols(
##   model = col_character(),
##   param = col_character(),
##   estimate = col_double()
## )
```

```
h2_post <- readr::read_csv(latest_h2_posterior)
```

```
## Parsed with column specification:
## cols(
##   model = col_character(),
##   param = col_character(),
##   estimate = col_double()
## )
```

Create tables of parameter estimates and uncertainty from posterior distributions for H1 and H2

H1 parameter table

```
library(tidyverse)
library(MCMCpack)
```

```
## Loading required package: coda
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2020 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

## ##
## ## Support provided by the U.S. National Science Foundation

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
h1_param_table <- h1_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  group_by(param) %>%
  summarise(
    mean = mean(estimate),
    sd = sd(estimate),
    lower_hpdi_95 = HPDinterval(as.mcmc(estimate))[[1]],
    upper_hpdi_95 = HPDinterval(as.mcmc(estimate))[[2]],
    mean_OR = exp(mean(estimate))) %>%
  map_if(is.numeric, round, digits = 3) %>%
  as.data.frame()

h1_param_table
```

```
##                param   mean    sd lower_hpdi_95 upper_hpdi_95 mean_OR
## 1        (Intercept) -0.727 0.186        -1.097        -0.367   0.483
## 2          lgbtqTRUE  0.635 0.155         0.329         0.938   1.887
## 3           maleTRUE -1.562 0.150        -1.857        -1.272   0.210
## 4     post_metoo_week  0.001 0.019        -0.035         0.039   1.001
## 5 post_metoo_week_SQ  0.000 0.000         0.000         0.001   1.000
## 6         post_metoo1 -0.274 0.250        -0.771         0.205   0.760
## 7    racial_minority1 -0.308 0.117        -0.535        -0.078   0.735
## 8           week_num  0.002 0.015        -0.027         0.032   1.002
## 9        week_num_SQ  0.000 0.000        -0.001         0.000   1.000
```

```
write.csv(
  x = h1_param_table,
  file = paste0('h1_param_table_', Sys.Date(), '.csv'),
  row.names = F)
```

H2 parameter table

```
h2_param_table <- h2_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  group_by(param) %>%
  summarise(
    mean = mean(estimate),
    sd = sd(estimate),
    lower_hpdi_95 = HPDinterval(as.mcmc(estimate))[[1]],
    upper_hpdi_95 = HPDinterval(as.mcmc(estimate))[[2]]) %>%
  map_if(is.numeric, round, digits = 3) %>%
  as.data.frame()

h2_param_table
```

```
##                      param   mean    sd lower_hpdi_95 upper_hpdi_95
## 1              (Intercept)  0.829 0.396         0.070         1.597
## 2                lgbtqTRUE  0.895 0.235         0.411         1.333
## 3                 maleTRUE -0.661 0.283        -1.222        -0.115
## 4           post_metoo_week  0.035 0.031        -0.028         0.095
## 5        post_metoo_week_SQ  0.001 0.000         0.000         0.002
## 6              post_metoo1  0.403 0.401        -0.393         1.175
## 7          racial_minority1 -0.183 0.193        -0.570         0.185
## 8                   sigma2  3.539 0.209         3.138         3.957
## 9           vic_worst_force  1.171 0.126         0.925         1.413
## 10          vic_worst_incap  0.533 0.088         0.361         0.707
## 11 vic_worst_prior_sexTRUE -0.684 0.167        -1.005        -0.356
## 12         vic_worst_verbal  0.244 0.114         0.018         0.464
## 13                 week_num -0.036 0.025        -0.085         0.013
## 14              week_num_SQ -0.001 0.000        -0.002         0.000
```

```
write.csv(
  x = h2_param_table,
  file = paste0('h2_param_table_', Sys.Date(), '.csv'),
  row.names = F)
```

MeToo Movement and Perceptions of Sexual Assault: Section 3

Below are the additional analyses that occurred after primary model fitting. Among them are estimates of sample size and the computation of the #MeToo effect (post hoc interaction analyses).

Setup - load participant dataframe

For cleanliness, we reload the data and model formulas from earlier in the analysis.

```
options(knitr.duplicate.label = 'allow')

knitr::knit('data_wrangling.Rmd', output = tempfile())
knitr::knit('models_and_priors.Rmd', output = tempfile())
```

Effective sample size (ESS)

To evaluate the strength of our priors, we used Effective Sample Size methods from Morita et al. (2008), implemented in the BayesESS package.

For the constant prevalence hypothesis, the ESS was...

```
library(tidyverse)

set.seed(5647891)

reg_priors <- rerun(9, c('norm', 0, 10))

BayesESS::ess(
  model = 'logistic',
  prior = reg_priors,
  ncov = 10,
  m = 5,
  nsim = 1000,
  fast = T,
  svec1 = rep(1, 12),
  svec2 = rep(1, 12))
```

```
##
## ESS was calculated for a logistic regression model
##
## ESSoverall: Overall ESS for the whole vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8
## ESSsubvector1: ESS for the first sub-vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8
## ESSsubvector1: ESS for the second sub-vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8

## $ESSoverall
## [1] 0.8213573
##
## $ESSsubvec1
```

```
## [1] 0.8213573
##
## $ESSsubvec2
## [1] 0.8213573
```

For the increasing acknowledgment hypothesis, the ESS was...

```
set.seed(5647891)

reg_priors <- rerun(11, c('norm', 0, 10))
reg_priors[[length(reg_priors) + 1]] <- c('gamma', 1, 1)

BayesESS::ess(
  model = 'linreg',
  prior = reg_priors,
  ncov = 10,
  m = 5,
  nsim = 1000,
  fast = T,
  svec1 = rep(1, 12),
  svec2 = rep(1, 12))
```

```
##
## ESS was calculated for a linear regression model
##
## ESSoverall: Overall ESS for the whole vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8
## ESSsubvector1: ESS for the first sub-vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8
## ESSsubvector1: ESS for the second sub-vector (theta1,theta2,theta3,theta4,theta5,theta6,theta7,theta8

## $ESSoverall
## [1] 0.4292039
##
## $ESSsubvec1
## [1] 0.4292039
##
## $ESSsubvec2
## [1] 0.4292039
```

Region of practical equivalence (ROPE)

For the H1 ROPE, we established a practical metric to evaluate changes in prevalence: the advantage or disadvantage of being in the sample's state, then moving to any other state at random. We calculated this by taking the average absolute difference between the study sample state and every other state.

State-level data were acquired from Table 3.9 of this report (https://www.cdc.gov/violenceprevention/pdf/NISVS-StateReportBook.pdf) and reflect the lifetime prevalence of contact sexual violence in **women.**

```
h1_rope <- readr::read_csv('rope_materials/NIP&SV_survey_2012.csv') %>%
  mutate(
    NE_change =
      abs(sex_vio_lifetime_prev -
            sex_vio_lifetime_prev[which(state.name == 'Nebraska')])) %>%
```

```
  filter(State != 'Nebraska') %>%
  dplyr::select(NE_change) %>%
  unlist() %>%
  mean() %>%
  magrittr::divide_by(100) %>%
  round(2)

h1_rope
```

```
## [1] 0.03
```

To clarify, this means any change - up or down - in the prevalence of reported sexual assault that is within $\pm 3\%$ of the pre-#MeToo prevalance of sexual assault would be regarded as "negligibly different from no change." This is because any change within that range is statistically equivalent to simply choosing another state in which to live at random. This is admittedly a rough metric and future research will be needed to come up with a more comprehensive set of benchmarks; however, it is used here for its simplicity and clarity, in lieu of another approach.

For the H2 ROPE, which relies on linear regression, consensus benchmarks for effect sizes are already well established. For that reason, we choose a ROPE of $d = \pm .10$, which would be a negligible effect size according to Cohen (1988).

```
d <- .10

h2_rope <- d*sd(df$vic_worst_is_sa, na.rm = T)
h2_rope
```

```
## [1] 0.2204741
```

Quadratic effect probes

The effect of the #MeToo movement involves three terms in each model: a dichotomous term, a linear term, and a quadratic term. Because none of these terms represents the overall #MeToo effect in isolation, we have incorporated them all into a single metric that accounts for all of their influence simultaneously. To accomplish this, we followed Aiken and West (1991) and calculated the derivative of our regression equation (with respect to the #MeToo dummy variable).

This is simplest in the case of the linear regression, which requires only a basic application of the power rule. Specifically, if

$$y = b_0 + b_1(t) + b_2(\text{MeToo}) + b_3(\text{MeToo})(t) + b_4(t^2) + b_5(\text{MeToo})(t^2)$$

Where, t is *time*, then

$$\frac{dy}{d\text{MeToo}} = b_2 + b_3 t + b_5 t^2$$

This is slightly more complicated in the logistic regression from H1, where the log-link function must also be taken into account. In that case, a combination of the chain and power rule thus yields,

$$\frac{dy}{d\text{MeToo}} = p(1-p)(b_2 + b_3 t + b_5 t^2)$$

where p is any sensible value for baseline prevalence of sexual assault. In this case, we have chosen the mean sexual assault prevalence across all pre-#MeToo weeks in the dataset $\approx 24\%$.

```
pre_metoo_vic_prob <- df %>%
  filter(post_metoo == 0) %>%
  dplyr::select(vic_screener) %>%
  unlist(use.names = F) %>%
  map_lgl(~ .x == 'Yes') %>%
  mean(na.rm = T)

pre_metoo_vic_prob
```

```
## [1] 0.2382325
```

Implementation of post-hoc #MeToo probes

To test this, we first import the posterior posterior distributions from both of our models.

```
latest_h1_posterior <- max(list.files()[grepl('h1_posteriors', list.files())])
latest_h2_posterior <- max(list.files()[grepl('h2_posteriors', list.files())])

# h1_post <- readr::read_csv('h1_posteriors_2019-03-20.csv')
# h2_post <- readr::read_csv('h2_posteriors_2019-03-20.csv')

h1_post <- readr::read_csv(latest_h1_posterior)
```

```
## Parsed with column specification:
## cols(
##   model = col_character(),
##   param = col_character(),
##   estimate = col_double()
## )
```

```
h2_post <- readr::read_csv(latest_h2_posterior)
```

```
## Parsed with column specification:
## cols(
##   model = col_character(),
##   param = col_character(),
##   estimate = col_double()
## )
```

H1

Beginning with H1, we probe the hypotheses at each of the 0-80 weeks after the MeToo movement starts by extracting the parameters of interest from our posterior distribution and multiplying by our probe points

```
h1_param_matrix <- h1_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  mutate(i = rep(1:(6*1e4), times = 9)) %>%
```

```r
  spread(param, estimate) %>%
  dplyr::select(post_metoo1, post_metoo_week, post_metoo_week_SQ) %>%
  as.matrix()

# study included 80 weeks after #metoo
probe_points <- cbind(1, 0:80, (0:80)^2)
h1_probes <- probe_points %*% t(h1_param_matrix)

# 60k samples for each of the 28 weeks
dim(h1_probes)
```

```
## [1]    81 60000
```

We then reformat the results into a dataframe.

```r
h1_probes <- h1_probes %>%
  as.data.frame() %>%
  set_names(1:length(.)) %>%
  mutate(week_num = row_number()) %>%
  gather(mcmc_samp, probe_est, -week_num) %>%
  mutate(mcmc_samp = as.numeric(mcmc_samp)) %>%
  arrange(week_num, mcmc_samp)

h1_probe_results <- h1_probes %>%
  mutate(
    # See equations above for clarification
    prev_est = pre_metoo_vic_prob*(1 - pre_metoo_vic_prob)*probe_est) %>%
  group_by(week_num) %>%
  summarise(
    mean_prev_est = mean(prev_est),
    below_rope = mean(prev_est < -h1_rope),
    within_rope = mean(-h1_rope < prev_est & prev_est < h1_rope),
    above_rope = mean(h1_rope < prev_est)) %>%
  round(2) %>%
  ungroup() %>%
  filter(week_num %% 10 == 0)

h1_probe_results
```

```
## # A tibble: 8 x 5
##   week_num mean_prev_est below_rope within_rope above_rope
##      <dbl>         <dbl>      <dbl>       <dbl>      <dbl>
## 1       10         -0.05       0.61       0.290       0.1
## 2       20         -0.04       0.54       0.22        0.24
## 3       30         -0.03       0.5        0.16        0.34
## 4       40         -0.01       0.47       0.12        0.41
## 5       50          0.01       0.45       0.09        0.46
## 6       60          0.03       0.44       0.07        0.5
## 7       70          0.05       0.43       0.05        0.52
## 8       80          0.08       0.42       0.04        0.54
```

```r
write.csv(h1_probe_results, paste0('h1_probe_results_', Sys.Date(), '.csv'), row.names = F)
```

Results show the posterior distribution is always overlapping - but not contained within - the ROPE. Thus, we can only conclude that we are uncertain about the population effect.

H2

The same procedure is repeated for H2.

```r
h2_post <- readr::read_csv(latest_h2_posterior)
```

```
## Parsed with column specification:
## cols(
##   model = col_character(),
##   param = col_character(),
##   estimate = col_double()
## )
```

```r
h2_param_matrix <- h2_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  mutate(i = rep(1:1e4, times = 14)) %>%
  spread(param, estimate) %>%
  dplyr::select(post_metoo1, post_metoo_week, post_metoo_week_SQ) %>%
  as.matrix()

wks <- 1:80
probe_points <- cbind(1, wks, wks^2)

h2_probes <- probe_points %*% t(h2_param_matrix)
```

```r
h2_probes <- h2_probes %>%
  as.data.frame() %>%
  set_names(1:length(.)) %>%
  mutate(week_num = row_number()) %>%
  gather(mcmc_samp, probe_est, -week_num) %>%
  mutate(mcmc_samp = as.numeric(mcmc_samp)) %>%
  arrange(week_num, mcmc_samp)
```

```r
h2_probe_results <- h2_probes %>%
  group_by(week_num) %>%
  summarise(
    below_rope = mean(probe_est < -h2_rope),
    within_rope = mean(-h2_rope < probe_est & probe_est < h2_rope),
    above_rope = mean(h2_rope < probe_est)) %>%
  round(2) %>%
  ungroup() %>%
  filter(week_num %% 10 == 0)

h2_probe_results
```

```
## # A tibble: 8 x 4
##   week_num below_rope within_rope above_rope
##      <dbl>      <dbl>       <dbl>      <dbl>
## 1       10       0.03        0.11       0.86
## 2       20       0.03        0.06       0.91
## 3       30       0.04        0.04       0.92
## 4       40       0.04        0.02       0.93
## 5       50       0.04        0.02       0.94
## 6       60       0.04        0.01       0.94
## 7       70       0.04        0.01       0.95
## 8       80       0.04        0.01       0.95
```

```
write.csv(h2_probe_results, paste0('h2_probe_results_', Sys.Date(), '.csv'), row.names = F)
```

Power

The sample was collected as part of another study, so N was fixed before analyses could even begin. To that end, we conducted a sensitivity analysis to see what kinds of measurement precision and hypothesis testing power were available for a sample with this pattern of data and covariate relationships.

This process involved first taking the posterior distributions for both hypotheses and assuming that that their means and covariance matrices were approximately "correct." Then, we modified the means and standard deviations for the parameters involved in the MeToo effect displayed above.

```
knitr::knit('data_wrangling.Rmd', output = tempfile())
knitr::knit('analysis_functions.Rmd', output = tempfile())
knitr::knit('models_and_priors.Rmd', output = tempfile())
```

Note, to ensure power estimation accounted for missing data, we used a multiply-imputed dataset for our simulations. This will better mimic actual power in our analyses than would using the observed dataset with listwise deletion.

```
library(mice)

categorical_vars <- c( 'post_metoo', 'male', 'lgbtq', 'racial_minority',
                       'vic_screener', 'vic_worst_prior_sex')

df[, categorical_vars] <- purrr::map(df[, categorical_vars], as.factor)

imp_seeds <- map_int(1:parallel::detectCores(), ~ as.integer(runif(1)*1e8))
sets_per_core <- 2

full_imp <- parallel::mclapply(
    X = imp_seeds,
    FUN = impute_mice,
    data = df,
    m = sets_per_core) %>%
  unpack_parallel_mids()

vic_imp <- parallel::mclapply(
    X = imp_seeds,
    FUN = impute_mice,
```

```
    data = filter(df, vic_screener == 'Yes'),
    m = sets_per_core) %>%
  unpack_parallel_mids()

paste(sets_per_core, 'imputed datasets per core')
```

```
## [1] "2 imputed datasets per core"
```

```
paste(parallel::detectCores(), 'cores detected')
```

```
## [1] "4 cores detected"
```

H1

For H1, we initially reload the posterior distributions for all model parameters from the main analysis.

```
h1_full_params <- h1_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  group_by(param) %>%
  mutate(i = row_number()) %>%
  # filter(i <= 1000) %>% XXX shrink posterior sample size when testing
  spread(param, estimate) %>%
  dplyr::select(-model, -i) %>%
  as.matrix()

head(h1_full_params)
```

```
##      (Intercept) lgbtqTRUE  maleTRUE post_metoo_week post_metoo_week_SQ
## [1,]  -0.8886429 0.7021671 -1.699400     0.026482121      -5.981125e-05
## [2,]  -0.5937209 0.8038032 -1.717845    -0.026519924      -1.187580e-04
## [3,]  -0.5015628 0.3757730 -1.635688     0.012636844       3.116562e-04
## [4,]  -0.6471048 0.4610756 -1.664365     0.008383404       4.118130e-04
## [5,]  -0.9455278 0.9011552 -1.560569     0.006029041       5.471285e-04
## [6,]  -0.5702671 0.7200323 -1.691096     0.016829588       5.242529e-04
##      post_metoo1 racial_minority1     week_num   week_num_SQ
## [1,]  -0.5739410       -0.2485226 -0.003971182 -0.0001507965
## [2,]  -0.2015509       -0.1793750  0.019833539  0.0002276976
## [3,]  -0.5931556       -0.3809182 -0.003389214 -0.0003712934
## [4,]  -0.1409018       -0.3297326 -0.008010630 -0.0004108269
## [5,]   0.1094466       -0.2327070 -0.014911860 -0.0004113637
## [6,]  -0.3998894       -0.4711051 -0.011570891 -0.0005208271
```

We then isolated the variables involved in the power analysis and create a population-level effect for them, which will later be inserted into our simulated datasets for power analysis. In the case of H1, we created a population-level effect for the post-hoc effect of #MeToo that was centered at 0 and always contained within the ROPE, 6 months after #MeToo.

```
power_vars <- c('post_metoo1', 'post_metoo_week', 'post_metoo_week_SQ')
param_means <- colMeans(h1_full_params)
param_sds <- apply(h1_full_params, 2, sd)
```

```r
param_cor <- cor(h1_full_params)

 # This assumes no suddent shift, followed by no gradual, no accelerating growth
param_means[power_vars] <- c(0, 0, 0)

# values chosen by guessing and checking in next code block
# to produce 95% variance within H1 ROPE
param_sds[power_vars] <- c(.001, .0001, .00001)


param_cov <- diag(param_sds) %*% param_cor %*% diag(param_sds)

set.seed(76846235)

param_samples <- mvtnorm::rmvnorm(
  n = 10,
  mean = param_means,
  sigma = param_cov)

sim_metoo_effect <- param_samples[, power_vars] %*% c(1, 24, 24^2)
mean(pre_metoo_vic_prob*(1 - pre_metoo_vic_prob)*sim_metoo_effect)
```

```
## [1] -0.0004252763
```

```r
1.96*sd(sim_metoo_effect)
```

```
## [1] 0.01211935
```

```r
round(mean(sim_metoo_effect > - h1_rope & sim_metoo_effect < h1_rope), 3)
```

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

With the population-level effect distribution created, we inject samples of this effect into each of 100 datasets, then analyze them using our main analysis model.

```r
design_names <- colnames(param_samples) %>%
  gsub('TRUE|1|\\)|\\(', '', .)

design_matrix <- full_imp$`1` %>%
  sample_n(size = nrow(.)*100, replace = T) %>%
  map_df(as.numeric) %>%
  cbind(Intercept = 1, .) %>%
  as.matrix()

sim_logit <- design_matrix[, design_names] %*% t(param_samples)
sim_p <- exp(sim_logit)/(1 + exp(sim_logit))
sim_y <- runif(length(sim_p)) < sim_p


sim_y <- sim_y %>%
  as.data.frame() %>%
```

```
  set_names(paste0('', 1:ncol(.))) %>%
  mutate(id = row_number()) %>%
  gather(samp, y, -id)

sim_df <- full_imp$`1` %>%
  mutate(id = row_number()) %>%
  right_join(sim_y, by = 'id') %>%
  mutate(vic_screener = y)
```

We then analyze them without the main analysis model...

```
library(HDInterval)

Sys.time()
```

```
## [1] "2020-05-23 10:41:03 EDT"
```

```
h1_power_models <- sim_df %>%
  split(.$samp) %>%
  map(
    .f = ~ MCMClogit(
      formula = h1_models$post_metoo_time_SQ,
      b0 = 0,
      B0 = 1/10,
      data = .x))

Sys.time()
```

```
## [1] "2020-05-23 10:41:33 EDT"
```

...and compile the results from those analyses.

```
h1_power_posteriors <- h1_power_models %>%
  map(
    .f = ~ c(
      pre_metoo_vic_prob*(1-pre_metoo_vic_prob)*.x[, power_vars] %*%
        c(1, 24, 24^2)))

h1_power_results <- h1_power_posteriors %>%
  map(
    .f = ~ data.frame(
      mean_est = mean(.x),
      perc_within_rope = mean(h1_rope < .x & .x < h1_rope),
      perc_within_pm10 = mean(-.10 < .x & .x < .10),
      perc_within_pm20 = mean(-.20 < .x & .x < .20),
      perc_within_pm30 = mean(-.30 < .x & .x < .30),
      lhdi = hdi(.x)[['lower']],
      uhdi = hdi(.x)[['upper']])) %>%
  bind_rows(.id = 'samp') %>%
  summarise(
    mean_est = mean(mean_est),
```

```r
    power_for_rope = mean(perc_within_rope >= .95),
    power_for_pm10 = mean(perc_within_pm10 >= .95),
    power_for_pm20 = mean(perc_within_pm20 >= .95),
    power_for_pm30 = mean(perc_within_pm30 >= .95),
    mean_hpdi_width = mean(uhdi - lhdi))

h1_power_results
```

```
##     mean_est power_for_rope power_for_pm10 power_for_pm20 power_for_pm30
## 1 0.01381373              0              0              0            0.2
##   mean_hpdi_width
## 1       0.5839684
```

H2

For H2, we repeat an identical procedure as in H1.

```r
h2_full_params <- h2_post %>%
  filter(model == 'post_metoo_time_SQ') %>%
  mutate(i = rep(1:1e4, times = 14)) %>%
  spread(param, estimate) %>%
  dplyr::select(-model, -i) %>%
  as.matrix()

power_vars <- c('post_metoo1', 'post_metoo_week', 'post_metoo_week_SQ')
param_means <- colMeans(h2_full_params)
param_sds <- apply(h2_full_params, 2, sd)
param_cor <- cor(h2_full_params)

# values that produce an effect of about .30 by week 24 post metoo
# This assumes no suddent shift, followed by gradual, accelerating growth
param_means[power_vars] <- c(0, .04, .0001)

# values chosen by guessing and checking in next code block
# to produce variance for post-metoo effect of about
# +1-point on the outcome scale
param_sds[power_vars] <- c(.01, .02, .0001)


param_cov <- diag(param_sds) %*% param_cor %*% diag(param_sds)
```

```r
set.seed(156498416)

param_samples <- mvtnorm::rmvnorm(
  n = 100,
  mean = param_means,
  sigma = param_cov)

sim_metoo_effect <- param_samples[, power_vars] %*% c(1, 24, 24^2)
mean(sim_metoo_effect)
```

```
## [1] 0.9759749
```

```r
sd(sim_metoo_effect)
```

```
## [1] 0.5252076
```

```r
round(mean(sim_metoo_effect > .10), 3)
```

```
## [1] 0.96
```

```r
sim_error <- param_samples[, 'sigma2']
param_samples <- param_samples[, colnames(param_samples) != 'sigma2']

design_names <- colnames(param_samples) %>%
  gsub('TRUE|1|\\)|\\(', '', .)

design_matrix <- vic_imp$`1` %>%
  map_df(as.numeric) %>%
  cbind(Intercept = 1, .) %>%
  as.matrix()

sim_y <- design_matrix[, design_names] %*% t(param_samples) + sim_error

sim_y <- sim_y %>%
  as.data.frame() %>%
  set_names(paste0('', 1:ncol(.))) %>%
  mutate(id = row_number()) %>%
  gather(samp, y, -id)

sim_df <- vic_imp$`1` %>%
  mutate(id = row_number()) %>%
  right_join(sim_y, by = 'id') %>%
  mutate(vic_worst_is_sa = y)
```

Results are again estimated using the analysis model and then compiled to produce a power estimate.

```r
library(HDInterval)

h2_power_sims <- sim_df %>%
  split(.$samp) %>%
  map(
    .f = ~ MCMCregress(
      formula = h2_models$post_metoo_time_SQ,
      b0 = 0,
      B0 = 1/10,
      data = .x)) %>%
  map(~ c(.x[, power_vars] %*% c(1, 24, 24^2))) %>%
  map(
    .f = ~ data.frame(
      mean_est = mean(.x),
      perc_above_rope = mean(.x > h2_rope),
      lhdi = hdi(.x)[['lower']],
      uhdi = hdi(.x)[['upper']])) %>%
```

```r
  bind_rows(.id = 'samp') %>%
  summarise(
    power = mean(perc_above_rope > .95),
    mean_hdi_width = mean(uhdi - lhdi),
    sd_hd_width = sd(uhdi - lhdi))

h2_power_sims
```

```
##   power mean_hdi_width sd_hd_width
## 1  0.85      0.4896485 0.001734127
```

END OF DOCUMENT