Misinformation Exposure and the False Belief that Trump Won the 2020 U.S. Presidential Election

Ross Dahlke

October 19, 2022

Data

```
main_data <- read_csv("data/main_data.csv") %>%
    mutate(untrustworthy_n_total = untrustworthy_n_pre + untrustworthy_n_test,
           untrustworthy_flag_total = if_else(untrustworthy_n_total > 0, 1, 0))
```

Top-line difference

```
weights::wtd.t.test(
  x = main_data %>% filter(trump_support_pre == 1) %>%  pull(won_election_trump),
  weight = main_data %>% filter(trump_support_pre == 1) %>% pull(weight))
```

```
## $test
## [1] "One Sample Weighted T-Test"
##
## $coefficients
##   t.value       df   p.value
##   19.4667 427.0000    0.0000
##
## $additional
##   Difference        Mean Alternative    Std. Err
##   0.47019208  0.47019208  0.00000000  0.02415366
```

Exposure Descriptives

```
t.test(main_data %>% filter(untrustworthy_n_test > 0) %>% pull(untrustworthy_n_test)) %>%
  broom::tidy()
```

```
## # A tibble: 1 x 8
##   estimate statistic     p.value parameter conf.low conf.high method     alter~1
##      <dbl>     <dbl>       <dbl>     <dbl>    <dbl>     <dbl> <chr>      <chr>
## 1     36.5      5.37 0.000000126       472     23.1      49.8 One Sampl~ two.si~
## # ... with abbreviated variable name 1: alternative
```

Gelbach Decomposition

Binary

```
save(binary_model1, file = "tables_and_figures/binary_model1")
save(binary_model2, file = "tables_and_figures/binary_model2")
save(binary_model3, file = "tables_and_figures/binary_model3")
save(binary_model4, file = "tables_and_figures/binary_model4")
save(binary_model5, file = "tables_and_figures/binary_model5")
save(binary_model6, file = "tables_and_figures/binary_model6")
save(binary_model7, file = "tables_and_figures/binary_model7")
save(binary_model8, file = "tables_and_figures/binary_model8")
save(binary_model9, file = "tables_and_figures/binary_model9")
save(binary_model10, file = "tables_and_figures/binary_model10")
```

Dosage

```
outcome <- "won_election_trump"
lm1 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test"), collapse = " + "),
        sep = " ~ "))
dosage_model1 <- lm(lm1, data = main_data, weights = main_data$weight)

lm2 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "trump_support_pre"), collapse = " + "),
        sep = " ~ "))
dosage_model2 <- lm(lm2, data = main_data, weights = main_data$weight)

lm3 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre"), collapse = " + "),
        sep = " ~ "))
dosage_model3 <- lm(lm3, data = main_data, weights = main_data$weight)

lm4 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ
        sep = " ~ "))
dosage_model4 <- lm(lm4, data = main_data, weights = main_data$weight)

lm5 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ
        sep = " ~ "))
dosage_model5 <- lm(lm5, data = main_data, weights = main_data$weight)

lm6 <- as.formula(
  paste(outcome,
```

```
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ4
        sep = " ~ "))
dosage_model6 <- lm(lm6, data = main_data, weights = main_data$weight)

lm7 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ4
        sep = " ~ "))
dosage_model7 <- lm(lm7, data = main_data, weights = main_data$weight)

lm8 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ4
        sep = " ~ "))
dosage_model8 <- lm(lm8, data = main_data, weights = main_data$weight)

lm9 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre","educ4_college_grad", "educ4_
        sep = " ~ "))
dosage_model9 <- lm(lm9, data = main_data, weights = main_data$weight)

lm10 <- as.formula(
  paste(outcome,
        paste(c("untrustworthy_n_test", "total_n_pre", "trump_support_pre", "educ4_college_grad", "educ4
        sep = " ~ "))
dosage_model10 <- lm(lm10, data = main_data, weights = main_data$weight)
```

```
save(dosage_model1, file = "tables_and_figures/dosage_model1")
save(dosage_model2, file = "tables_and_figures/dosage_model2")
save(dosage_model3, file = "tables_and_figures/dosage_model3")
save(dosage_model4, file = "tables_and_figures/dosage_model4")
save(dosage_model5, file = "tables_and_figures/dosage_model5")
save(dosage_model6, file = "tables_and_figures/dosage_model6")
save(dosage_model7, file = "tables_and_figures/dosage_model7")
save(dosage_model8, file = "tables_and_figures/dosage_model8")
save(dosage_model9, file = "tables_and_figures/dosage_model9")
save(dosage_model10, file = "tables_and_figures/dosage_model10")
```

Double ML

Binary

```
dml_data <- DoubleMLData$new(main_data %>% as.data.frame() %>% select(won_election_trump, untrustworthy_
                             y_col = "won_election_trump",
                             d_cols = "untrustworthy_flag_test",
                             x_cols = c("untrustworthy_flag_pre", "total_n_pre", "trump_support_pre", "e
```

```r
learner <- lrn("regr.ranger", num.trees = 500, mtry = floor(sqrt(12)), max.depth = 5, min.node.size = 2
ml_g <- learner$clone()
ml_m <- learner$clone()
```

```r
obj_dml_plr <- DoubleMLPLR$new(dml_data, ml_g = ml_g, ml_m = ml_m)
```

```r
obj_dml_plr$fit()
obj_dml_plr
```

```
## ================= DoubleMLPLR Object ==================
##
##
## ------------------ Data summary      ------------------
## Outcome variable: won_election_trump
## Treatment variable(s): untrustworthy_flag_test
## Covariates: untrustworthy_flag_pre, total_n_pre, trump_support_pre, educ4_college_grad, educ4_hs_or_l
## Instrument(s):
## No. Observations: 1194
##
## ------------------ Score & algorithm ------------------
## Score function: partialling out
## DML algorithm: dml2
##
## ------------------ Machine learner   ------------------
## ml_l: regr.ranger
## ml_m: regr.ranger
##
## ------------------ Resampling        ------------------
## No. folds: 5
## No. repeated sample splits: 1
## Apply cross-fitting: TRUE
##
## ------------------ Fit summary       ------------------
##  Estimates and significance testing of the effect of target variables
##                         Estimate. Std. Error t value Pr(>|t|)
## untrustworthy_flag_test   0.05889    0.02216   2.658  0.00787 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dosage

```r
dml_data <- DoubleMLData$new(main_data %>% as.data.frame() %>% select(won_election_trump, untrustworthy_
                             y_col = "won_election_trump",
                             d_cols = "untrustworthy_n_test",
                             x_cols = c("untrustworthy_flag_pre", "trump_support_pre", "college", "femal
```

```r
learner <- lrn("regr.ranger", num.trees = 500, mtry = floor(sqrt(12)), max.depth = 5, min.node.size = 2
ml_g <- learner$clone()
ml_m <- learner$clone()
```

```
obj_dml_plr <- DoubleMLPLR$new(dml_data, ml_g = ml_g, ml_m = ml_m)
```

```
obj_dml_plr$fit()
obj_dml_plr
```

```
## ================= DoubleMLPLR Object ==================
##
##
## ------------------ Data summary      ------------------
## Outcome variable: won_election_trump
## Treatment variable(s): untrustworthy_n_test
## Covariates: untrustworthy_flag_pre, trump_support_pre, college, female, non_white, knowledge, interes
## Instrument(s):
## No. Observations: 1194
##
## ------------------ Score & algorithm ------------------
## Score function: partialling out
## DML algorithm: dml2
##
## ------------------ Machine learner   ------------------
## ml_l: regr.ranger
## ml_m: regr.ranger
##
## ------------------ Resampling        ------------------
## No. folds: 5
## No. repeated sample splits: 1
## Apply cross-fitting: TRUE
##
## ------------------ Fit summary       ------------------
##  Estimates and significance testing of the effect of target variables
##                      Estimate. Std. Error t value    Pr(>|t|)
## untrustworthy_n_test 0.00035018 0.00006645    5.27 0.000000136 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Causal Forest

```
set.seed(1)
X_binary <- main_data[, c("untrustworthy_flag_pre", "trump_support_pre", "educ4_college_grad", "educ4_hs
Y_binary <- main_data$won_election_trump
W_binary <- main_data$untrustworthy_flag_test
```

```
tau.forest_binary <- causal_forest(X_binary, Y_binary, W_binary, num.trees = 4000)
tau.hat_binary <- predict(tau.forest_binary, X_binary, estimate.variance = TRUE)
sigma.hat_binary <- sqrt(tau.hat_binary$variance.estimates)
```

```
average_treatment_effect(tau.forest_binary, target.sample = "overlap")
```

```
##   estimate    std.err
## 0.04194356 0.01966475
```

```r
main_data_predictions_binary <- main_data %>%
  cbind(tau.hat_binary)
```

```r
average_treatment_effect(tau.forest_binary, target.sample = "overlap", subset = main_data$trump_support
```

```
##   estimate    std.err
## 0.12587578 0.05555805
```

```r
average_treatment_effect(tau.forest_binary, target.sample = "overlap", subset = main_data$trump_support
```

```
##    estimate     std.err
## -0.00177385  0.00884615
```

```r
test_calibration(tau.forest_binary)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value  Pr(>t)
## mean.forest.prediction          0.961393   0.464795  2.0684 0.01941 *
## differential.forest.prediction -0.021454   0.543922 -0.0394 0.51573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dosage

```r
X_dosage <- main_data[, c("untrustworthy_flag_pre", "trump_support_pre", "educ4_college_grad", "educ4_hs
Y_dosage <- main_data$won_election_trump
W_dosage <- main_data$untrustworthy_n_test
```

```r
tau.forest_dosage <- causal_forest(X_dosage, Y_dosage, W_dosage, num.trees = 4000)
tau.hat_dosage <- predict(tau.forest_dosage, X_dosage, estimate.variance = TRUE)
sigma.hat_dosage <- sqrt(tau.hat_dosage$variance.estimates)
```

```r
average_treatment_effect(tau.forest_dosage, target.sample = "overlap")
```

```
##      estimate       std.err
## 0.00033741188 0.00007802666
```

```r
main_data_predictions_dosage <- main_data %>%
  cbind(tau.hat_dosage)
```

```r
average_treatment_effect(tau.forest_dosage, target.sample = "overlap", subset = main_data$trump_support
```

```
##      estimate       std.err
## 0.00035346681 0.00009593705
```

```r
average_treatment_effect(tau.forest_dosage, target.sample = "overlap", subset = main_data$trump_support
```

```
##     estimate      std.err
## 0.0001008334 0.0001556445
```

```r
test_calibration(tau.forest_dosage)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value  Pr(>t)
## mean.forest.prediction          0.98300    0.45920  2.1407 0.01625 *
## differential.forest.prediction -0.32848    0.79272 -0.4144 0.66066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
main_data %>%
  cbind(tau.hat_dosage) %>%
  mutate(trump_support_factor = if_else(trump_support_pre == 0, "Liberal", "Conservative"),
         trump_support_factor = factor(trump_support_factor, levels = c("Liberal", "Conservative")),
         untrustworthy_n_test = untrustworthy_n_test + .1) %>%
  ggplot(aes(untrustworthy_n_test, predictions, color = trump_support_factor)) +
  geom_point(alpha = .2) +
  geom_smooth(method = "lm") +
  scale_color_manual(values = c("darkblue", "darkred")) +
  scale_x_log10(breaks = c(0, 1, 10, 100, 1000)) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Conditional Average Dosage Effect of Misinformation Exposure on
False Fraudulent Election Belief & # of Misinformation Exposures",
subtitle = "Each point represents one person",
x = "Log # of Misinformation Website Exposure",
y = "Participants' Conditional Average Dosage Effect
on the False Belief that the Election was Fraudulent",
color = "") +
  theme_bw()
```

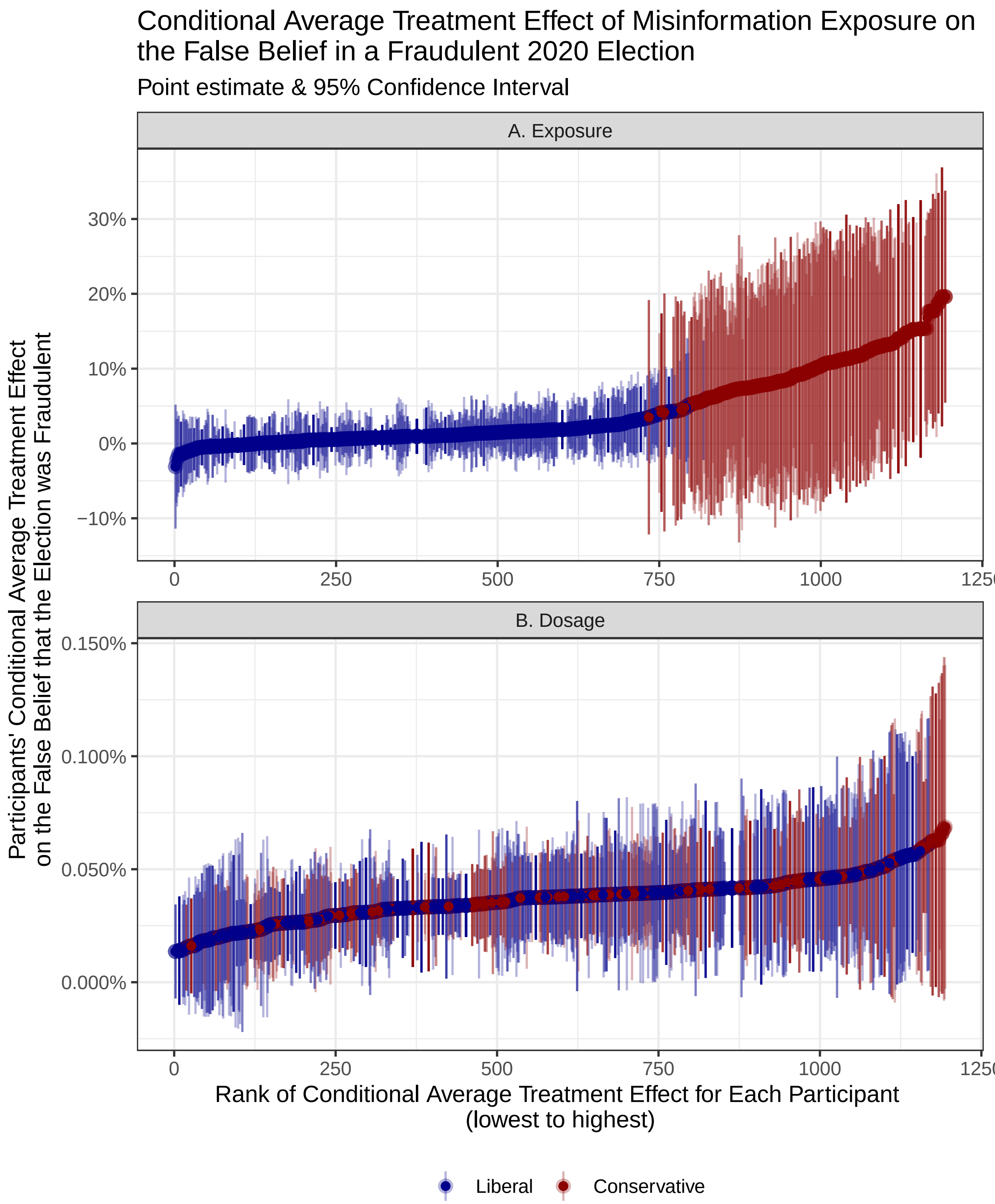


Figure 1: Plot of estimated conditional average differences and 95% confidence interval of misinformation exposure on the belief that Donald Trump won the 2020 U.S. Presidential Election for each individual in our sample. Participants are ordered along the x-axis in order from lowest estimated conditional difference to the highest. The y-axis is the estimated conditional average difference.

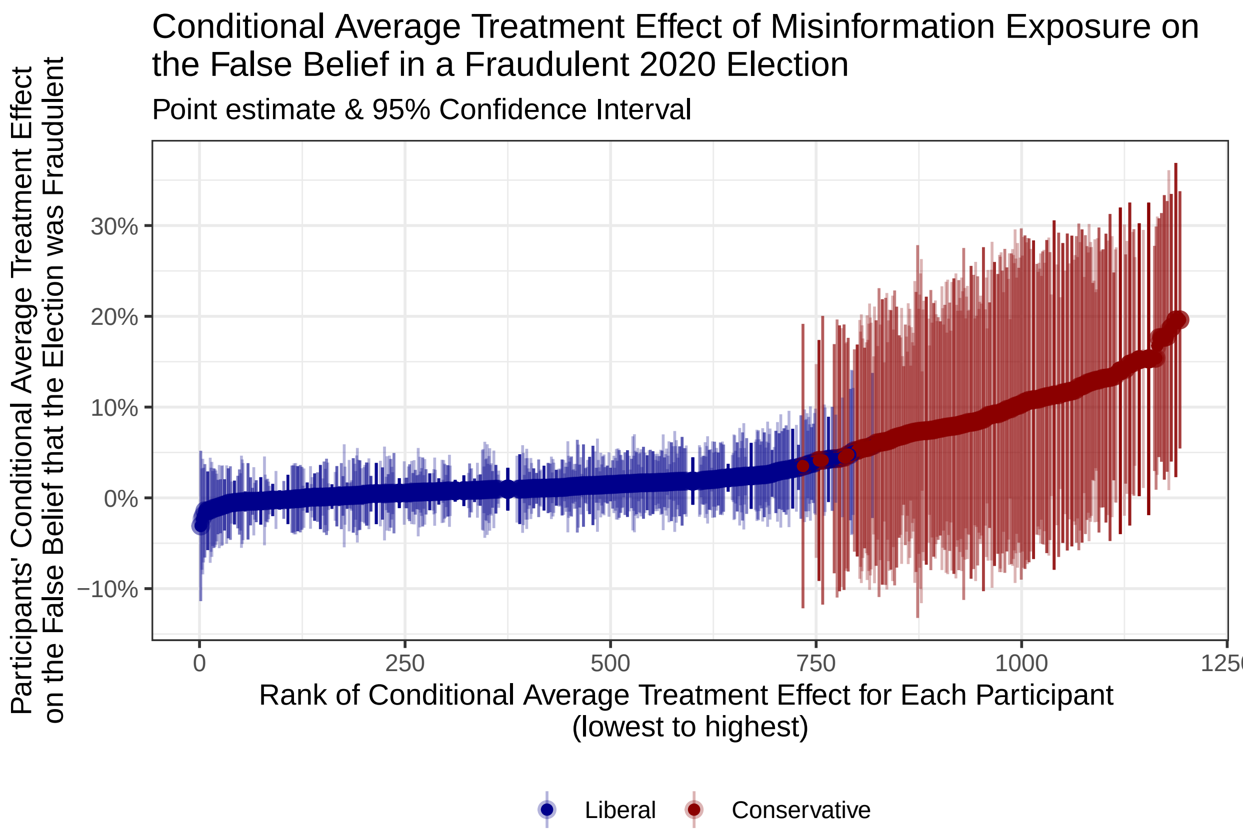
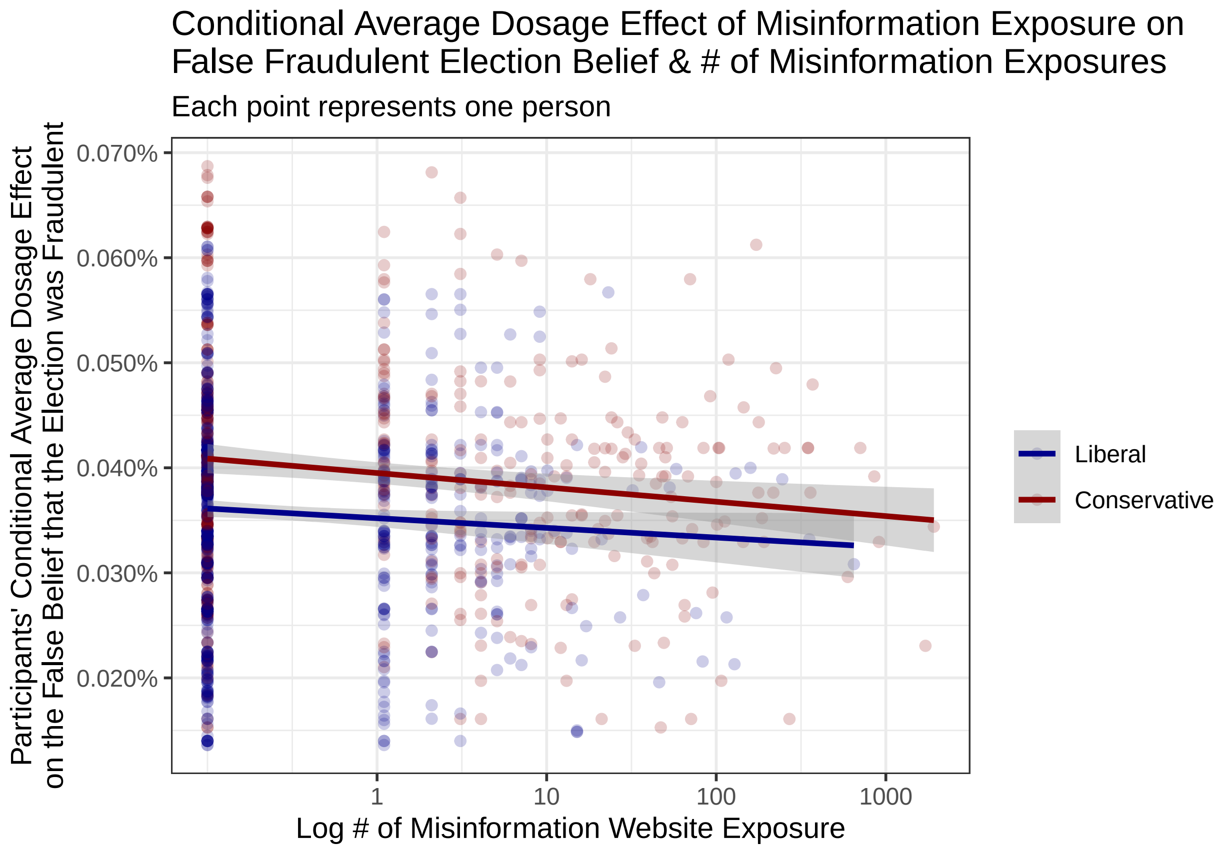


Figure 2: Plot of estimated conditional average differences and 95% confidence interval of misinformation exposure on the belief that Donald Trump won the 2020 U.S. Presidential Election for each individual in our sample. Participants are ordered along the x-axis in order from lowest estimated conditional difference to the highest. The y-axis is the estimated conditional average difference.



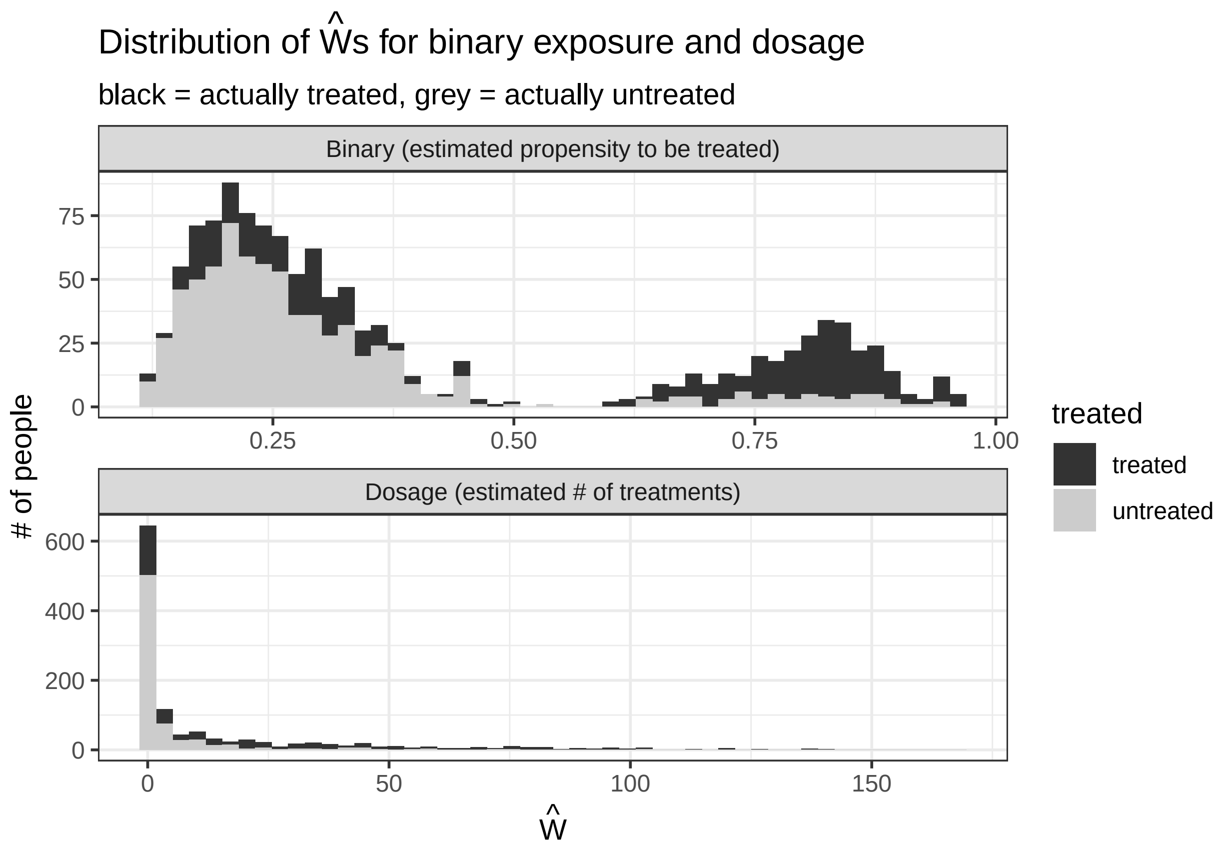

```
ggsave("tables_and_figures/dosage_graph.pdf")
```

```
main_data %>%
  cbind(tau.hat_dosage) %>%
  lm(predictions ~ log(untrustworthy_n_test + .1) * trump_support_pre, data = .) %>%
  stargazer::stargazer(header = FALSE, type = "latex", dep.var.labels.include = F, dep.var.caption = "C
```

```
main_data %>%
  cbind(w_hat_binary = tau.forest_binary$W.hat) %>%
  cbind(w_hat_dosage = tau.forest_dosage$W.hat) %>%
  pivot_longer(cols = c(w_hat_binary, w_hat_dosage)) %>%
  mutate(treated = if_else(untrustworthy_flag_test == 1, "treated", "untreated"),
         name = if_else(name == "w_hat_binary", "Binary (estimated propensity to be treated)", "Dosage
  ggplot(aes(value, fill = treated)) +
  geom_histogram(bins = 50) +
  scale_fill_grey() +
  theme_bw() +
  theme(legend.position = "right") +
  facet_wrap(.~name, scales = "free", ncol = 1) +
  labs(title = latex2exp::TeX(r'(Distribution of \hat{W}s for binary exposure and dosage)'),
       subtitle = "black = actually treated, grey = actually untreated",
       x = latex2exp::TeX(r'(\hat{W})'),
       y = "# of people")
```

Table 1:

	Conditional Average Dosage Effect
log(Number of Misinformation Website Exposures)	−0.00000
	(0.00000)
trump_support_pre	0.00004***
	(0.00001)
log(untrustworthy_n_test + 0.1):trump_support_pre	−0.00000
	(0.00000)
Constant	0.0004***
	(0.00000)
Observations	1,194
R^2	0.042
Adjusted R^2	0.040
Residual Std. Error	0.0001 (df = 1190)
F Statistic	17.425*** (df = 3; 1190)
Note:	*p<0.05; **p<0.01; ***p<0.001



```
ggsave("tables_and_figures/w_hat_distributions.pdf")
```

Oster Omitted Variable Bias

Binary

```
f01c <- won_election_trump ~ untrustworthy_flag_test + untrustworthy_flag_pre + total_n_pre + trump_supp

f01u <- won_election_trump ~ untrustworthy_flag_test

fit01c <- lm(f01c, data = main_data, weights = main_data$weight)

main_data$infit01c <- is.element(rownames(main_data), names(fit01c$residuals))

fit01u <- lm(f01u, data = main_data, subset = infit01c, weights = main_data$weight)

z <- oster(fit01u, fit01c, "untrustworthy_flag_test")
b13 <- oster(fit01u, fit01c, "untrustworthy_flag_test", rm = 1.3)$beta

round(c(z$input$beta_o, z$input$beta_tilde, z$beta, b13, z$rmax), 6)
```

```
## [1]  0.173302  0.057813 -0.148747  0.000965  0.494875
```

Dosage

```
f01c <- won_election_trump ~ untrustworthy_n_test + untrustworthy_flag_pre + total_n_pre + trump_support

f01u <- won_election_trump ~ untrustworthy_n_test

fit01c <- lm(f01c, data = main_data, weights = main_data$weight)

main_data$infit01c <- is.element(rownames(main_data), names(fit01c$residuals))

fit01u <- lm(f01u, data = main_data, subset = infit01c, weights = main_data$weight)

z <- oster(fit01u, fit01c, "untrustworthy_n_test")
b13 <- oster(fit01u, fit01c, "untrustworthy_n_test", rm = 1.3)$beta

round(c(z$input$beta_o, z$input$beta_tilde, z$beta, b13, z$rmax), 6)
```

```
## [1]  0.000809  0.000381 -0.000069  0.000239  0.683844
```

```
temp <- robomit::o_beta_boot(y = "won_election_trump",
                x = "untrustworthy_n_test",
                con = "untrustworthy_flag_pre + total_n_pre + trump_support_pre + educ4_college_grad + e
                data = main_data,
                R2max = 1.3*0.3798,
```

```
                type = "lm",
                w = "weight",
                delta = 1,
                sim = 1000,
                obs = 1000,
                rep = T)

mean(temp$`beta*`)
```

```
## [1] 0.002432631
```

```
temp <- robomit::o_beta_boot(y = "won_election_trump",
                x = "untrustworthy_flag_test",
                con = "untrustworthy_flag_pre + total_n_pre + trump_support_pre + educ4_college_grad + 
                data = main_data,
                R2max = 1.3*0.3821,
                type = "lm",
                w = "weight",
                delta = 1,
                sim = 1000,
                obs = 1000,
                rep = T,
                useed = 1)

median(temp$`beta*`)
```

```
## [1] 0.032089
```

```
temp <- robomit::o_beta_boot(y = "won_election_trump",
                x = "untrustworthy_n_test",
                con = "untrustworthy_flag_pre + total_n_pre + trump_support_pre + educ4_college_grad + 
                data = main_data,
                R2max = 1.3*0.3838,
                type = "lm",
                w = "weight",
                delta = 1,
                sim = 1000,
                obs = 1000,
                rep = T,
                useed = 1)

median(temp$`beta*`)
```

```
## [1] 0.000334
```

```r
weights::wtd.t.test(x = main_data$untrustworthy_n_total,
                    weight = main_data$weight)
```

```
## $test
## [1] "One Sample Weighted T-Test"
##
## $coefficients
##             t.value                  df             p.value
##    5.41157750431008 1193.00000000000000    0.00000007545303
##
## $additional
##  Difference        Mean Alternative    Std. Err
##   15.422216   15.422216    0.000000    2.849856
```

```r
weights::wtd.t.test(x = main_data %>% filter(untrustworthy_flag_total == 1) %>% pull(untrustworthy_n_to
                    weight = main_data %>% filter(untrustworthy_flag_total == 1) %>% pull(weight))
```

```
## $test
## [1] "One Sample Weighted T-Test"
##
## $coefficients
##            t.value                 df            p.value
##   5.78542153909954 534.00000000000000   0.00000001233252
##
## $additional
##  Difference        Mean Alternative    Std. Err
##   37.966261   37.966261    0.000000    6.562402
```

Articles mentioning "election"

```r
set.seed(1)
X_binary <- main_data[, c("untrustworthy_flag_pre", "trump_support_pre", "educ4_college_grad", "educ4_hs
Y_binary <- main_data$won_election_trump
W_binary <- main_data$election_untrustworthy_flag_test
```

```r
tau.forest_binary <- causal_forest(X_binary, Y_binary, W_binary, num.trees = 4000)
tau.hat_binary <- predict(tau.forest_binary, X_binary, estimate.variance = TRUE)
sigma.hat_binary <- sqrt(tau.hat_binary$variance.estimates)
```

```r
average_treatment_effect(tau.forest_binary, target.sample = "overlap")
```

```
##   estimate    std.err
## 0.03454481 0.03330964
```

```r
main_data_predictions_binary <- main_data %>%
  cbind(tau.hat_binary)
```

```
average_treatment_effect(tau.forest_binary, subset = main_data$trump_support == 1, target.sample = "ove
```

```
##   estimate    std.err
## 0.06959163 0.06517365
```

```
average_treatment_effect(tau.forest_binary, subset = main_data$trump_support == 0, target.sample = "ove
```

```
##     estimate      std.err
## -0.003126376  0.010726793
```

```
test_calibration(tau.forest_binary)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value Pr(>t)
## mean.forest.prediction          0.21874    1.18784  0.1842 0.4270
## differential.forest.prediction -3.29265    1.84044 -1.7891 0.9631
```

Dosage

```
set.seed(1)
X_dosage <- main_data[, c("untrustworthy_flag_pre", "trump_support_pre", "educ4_college_grad", "educ4_hs
Y_dosage <- main_data$won_election_trump
W_dosage <- main_data$election_untrustworthy_n_test
```

```
tau.forest_dosage <- causal_forest(X_dosage, Y_dosage, W_dosage, num.trees = 4000)
tau.hat_dosage <- predict(tau.forest_dosage, X_dosage, estimate.variance = TRUE)
sigma.hat_dosage <- sqrt(tau.hat_dosage$variance.estimates)
```

```
average_treatment_effect(tau.forest_dosage, target.sample = "overlap")
```

```
##     estimate      std.err
## 0.0009247837 0.0002402091
```

```
main_data_predictions_dosage <- main_data %>%
  cbind(tau.hat_dosage)
```

```
average_treatment_effect(tau.forest_dosage, subset = main_data$trump_support == 1, target.sample = "ove
```

```
##     estimate      std.err
## 0.0009895525 0.0005202569
```

```
average_treatment_effect(tau.forest_dosage, subset = main_data$trump_support == 0, target.sample = "ove
```

```
##     estimate      std.err
## 0.0002534368 0.0006564776
```

```r
test_calibration(tau.forest_dosage)
```

```
##
## Best linear fit using forest predictions (on held-out data)
## as well as the mean forest prediction as regressors, along
## with one-sided heteroskedasticity-robust (HC3) SEs:
##
##                                Estimate Std. Error t value  Pr(>t)
## mean.forest.prediction          1.33128    0.94000  1.4163 0.07848 .
## differential.forest.prediction -0.10308    3.44600 -0.0299 0.51193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

References