Screen Time Usage Experiment

Sebastian Vences

March 19, 2023

Background

This project aims to assess changes in college students' screen time behavior before and after receiving an email blast highlighting the potential dangers of excessive social media usage. The primary outcome is change in screen time.

I. Data Manipulation and Preparation

Load Libraries

```
library(estimatr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.2     v purrr   1.0.0
## v tibble  3.2.1     v dplyr   1.1.3
## v tidyr   1.2.1     v stringr 1.4.1
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Load Data

After loading our data into dataframe `raw_data`, I assigned a copy to `df` so that any necessary manipulation is done to this copy instead of overriding the raw data itself. I then used the `head()` function to explore the data and noted the long column names and variable types.

```
raw_data <- read_csv("https://www.dropbox.com/s/qoj5vqyfos0lbvl/fake_data_section_F.csv?dl=1")
```

```
## Rows: 348 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): treatment, Give an estimated time for how many hours of social medi...
## dbl (6): time before treatment, time after treatment, How much do you think ...
## lgl (1): section
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- raw_data

head(df)
```

```
## # A tibble: 6 x 9
```

```
##   section treatment `time before treatment` `time after treatment`
##   <lgl>   <chr>                      <dbl>                  <dbl>
## 1 FALSE   control                       14                     13
## 2 FALSE   beliefs                       15                     10
## 3 FALSE   knowledge                     18                     15
## 4 FALSE   knowledge                     17                      9
## 5 FALSE   knowledge                     11                     12
## 6 FALSE   control                       12                     15
## # i 5 more variables:
## #   `How much do you think your social media use is driven by the fear of missing out?` <dbl>,
## #   `Do you correlate screen-time with productivity?` <dbl>,
## #   `Do you feel that after going through the treatment and monitoring screen-time, there was an eff
## #   `Adolescents would be at risk for mental health struggles if they use social media for more than
## #   `Give an estimated time for how many hours of social media use per day will lead to individuals l
```

Cleaning the Data

I renamed the dataframe variables to shorter strings by assigning a character vector to `colnames(df)`. Using the `mutate()` function, I created variable `diff_time` that takes the difference between screen time after and before treatment. A negative `diff_time` implies a decrease in screen time after treatment. This is our primary-outcome variable. I adjusted the original `treat` variable to a character variable for clarity. I also created variable `fomo_level` that assignes "low" or "high" to an individual based on how their response to the `fomo` question relates to the mean response – this will be analyzed in Section III: Secondary Outcomes.

```
colnames(df) <- c(
  "section",
  "treat",
  "time_before",
  "time_after",
  "fomo",
  "correlate_prod",
  "effect_prod",
  "adolescents",
  "risk_health"
)

df <- df %>% # create a time difference variable -- primary outcome
  mutate(
    diff_time = time_after - time_before,
    treat = if_else(treat == "control", "control", "treatment"),
    fomo_level = if_else(fomo <= mean(fomo), "low", "high")
  )

head(df) # ensure successful variable creation
```

```
## # A tibble: 6 x 11
##   section treat     time_before time_after  fomo correlate_prod effect_prod
##   <lgl>   <chr>           <dbl>      <dbl> <dbl>          <dbl>       <dbl>
## 1 FALSE   control            14         13     3              1           4
## 2 FALSE   treatment          15         10     3              0           3
## 3 FALSE   treatment          18         15     3              0           4
## 4 FALSE   treatment          17          9     3              1           3
## 5 FALSE   treatment          11         12     1              0           3
## 6 FALSE   control            12         15     4              1           4
## # i 4 more variables: adolescents <dbl>, risk_health <chr>, diff_time <dbl>,
```

```
## #   fomo_level <chr>
```

Codebook

- **treat**:
  <chr> group ("control" or "treatment")
  Whether each subject placed in a control or treatment group.
- **time_before**:
  <dbl> time (hours)
  Screen time before treatment.
- **time_after**:
  <dbl> time (hours)
  Screen time after treatment.
- **diff_time**:
  <dbl> time (hours)
  Difference between screen time after treatment and screen time before treatment.
- **fomo**:
  <dbl> scale (1 = "not at all", 5 = "completely")
  How much do you think your social media use is driven by the fear of missing out?
- **fomo_level**:
  <chr> group ("low" or "high") Responses to "fomo" question less than or greater than the mean of "fomo" responses.
- **correlate_prod**:
  <dbl> binary (1 = yes, 0 = no)
  Do you correlate screen-time with productivity?
- **effect_prod**:
  <dbl> scale (1 = no effect, 5 = significant effect)
  Do you feel that after going through the treatment and monitoring screen-time, there was an effect on your productivity?
- **adolescents**:
  <dbl> binary (1 = yes, 0 = no)
  Adolescents would be at risk for mental health struggles if they use social media for more than 3 hours per day.
- **risk_health**:
  <chr> time range (hours)
  Give an estimated time for how many hours of social media use per day will lead to individuals being at risk for mental health struggles.

II. Primary Outcome

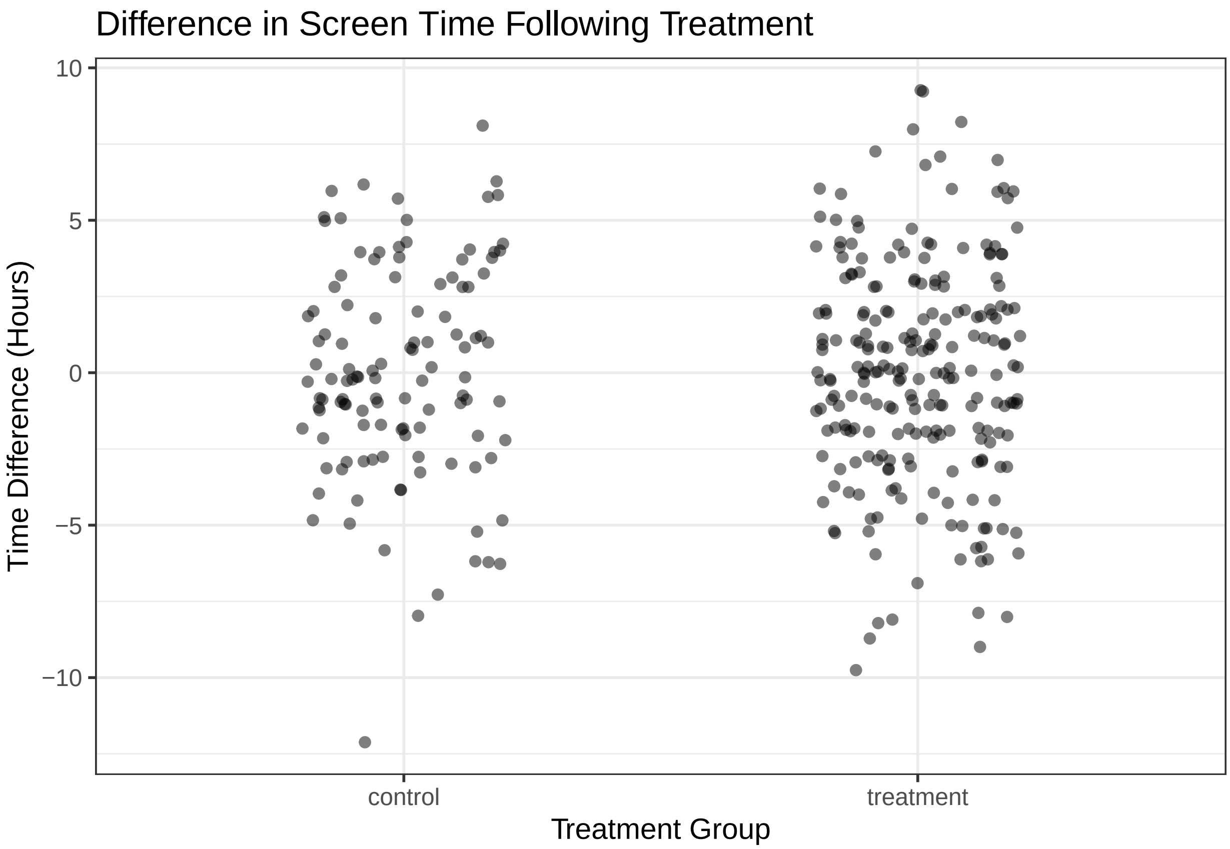
Our goal is determining the difference in screen times before and after we sent our treatment email.

Average Treatment Effect

Using the `ggplot()` function, I created a jitter plot to visualize our treatment effect against the individual's treatment group.

```r
df %>%
  ggplot(aes(x = treat, y = diff_time)) + # plot primary outcome vs. treatment group
  geom_jitter(alpha = 0.5, height = 0.3, width = 0.2) + # adjust jitter amount
  labs(x = "Treatment Group", y = "Time Difference (Hours)",
```

```
    title = "Difference in Screen Time Following Treatment") + # set plot labels
  theme_bw()
```



Applying `difference_in_means()`, I calculated our treatment's average effect and assigned the resulting dataframe object to `treat_effect`. In the future, I will access the difference-in-means estimate using the `$` operator, as in `treat_effect$estimate`.

```
treat_effect <- difference_in_means(
  diff_time ~ treat, # compare our primary outcome to treatment group
  condition1 = "control", # ensure our control and treatment groups properly assigned
  condition2 = "treatment",
  data = df
) %>% tidy

treat_effect
```

```
##              term    estimate std.error  statistic   p.value    conf.low
## 1 treattreatment -0.05172414 0.4011454 -0.1289411 0.8975138 -0.8419965
##    conf.high       df   outcome
## 1 0.7385482 236.6354 diff_time
```

> The average treatment effect is -0.0517241. According to our analysis, our treatment email reduced screen time by 0.05 hours. Is this value statistically different from a 0 hour difference in screen times?

Sharp Null Thought Experiment

Before running the sharp null though experiment, I calculated the standard deviation for our control group's primary outcome and assigned that value to `sd`. I also returned our treatment assignment's count, applying these values to my `sharp_null()` function.

```
sd <- sd(filter(df, treat == "control")$diff_time) # control group's standard deviation

df %>% count(treat)
```

```
## # A tibble: 2 x 2
##   treat         n
##   <chr>     <int>
## 1 control     116
## 2 treatment   232
```

In our sharp null thought experiment, we are assuming there is no individual treatment effect. I created function `sharp_null()` to assign potential outcomes for a control and treatment group, randomize treatment assignments, apply the switching equation to reveal actual outcomes, and finally return a difference-in-means estimate.

```
sharp_null <-
  function() {
    df %>%
      mutate(
        pot_outcome_control = rnorm(n(), sd = sd),
        pot_outcome_treatment = pot_outcome_control, # assume no individual treat effect
        Z = sample(rep(c(0, 1), times = c(116, 232)), size = n()), # randomize assignment
        Y = if_else(Z == 0, pot_outcome_control, pot_outcome_treatment)
      ) %>%
      difference_in_means(Y ~ Z, data = .) %>% # calculate the average treatment effect
      tidy
  }
```

After setting my generation seed, I reran the `sharp_null()` function 1000 times and saved each to `sampling_distribution_sharp_null`.

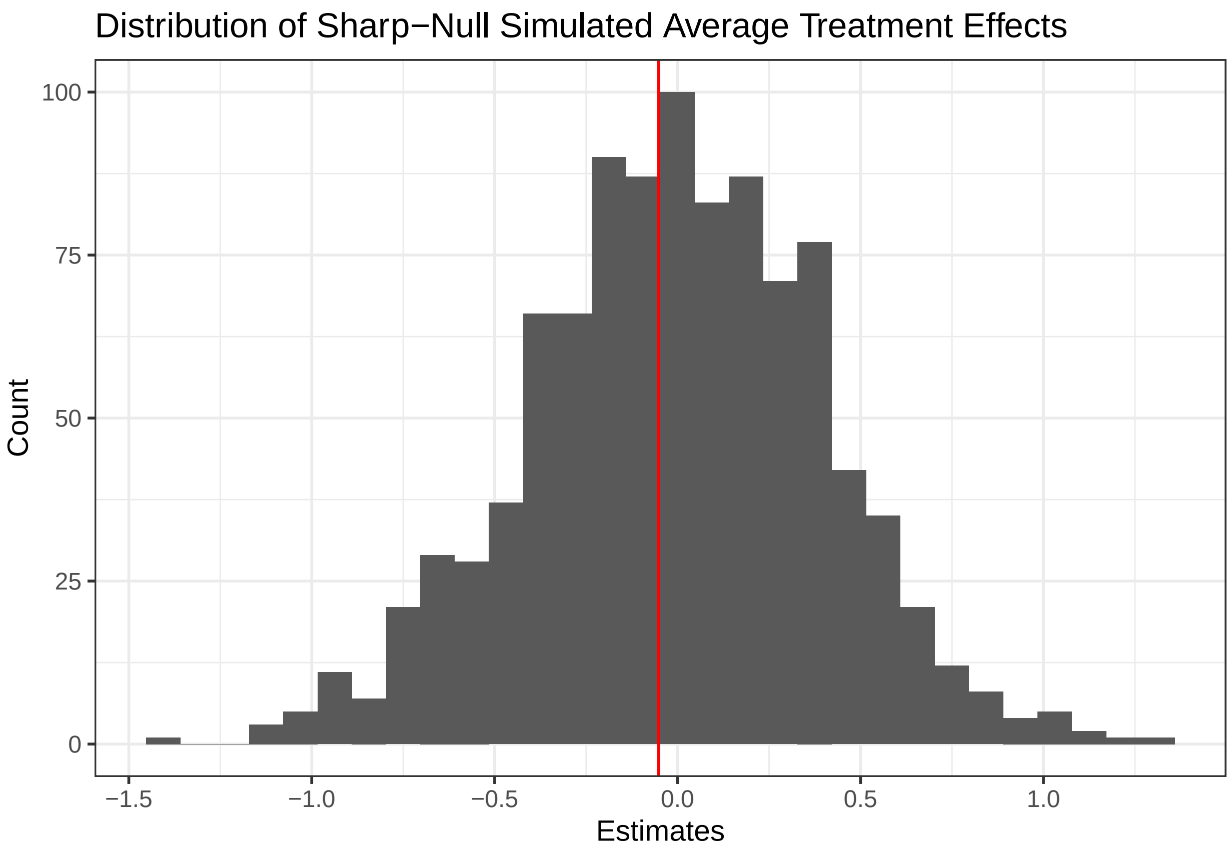
```
set.seed(1234)

sampling_distribution_sharp_null <-
  rerun(1000, sharp_null()) %>% # rerun the sharp_null function 1000 times
  bind_rows # saves each run
```

```
## Warning: `rerun()` was deprecated in purrr 1.0.0.
## i Please use `map()` instead.
##   # Previously
##   rerun(1000, sharp_null())
##
##   # Now
##   map(1:1000, ~ sharp_null())
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

I created a histogram of the `sampling_distribution_sharp_null` simulated average treatment effects and marked our calculated average treatment effect, noting the quantity of treatment effects to the right of that vertical line. It appears over about half of the thought experiment's average treatment effect estimates are to the right of that line.

```r
sampling_distribution_sharp_null %>%
  ggplot(aes(x = estimate)) + # plot the simulation's average treatment effects
    geom_histogram() +
    geom_vline(data = treat_effect, aes(xintercept = estimate), col = "red") + # our dim
    labs(x = "Estimates", y = "Count", # set plot labels
         title = "Distribution of Sharp-Null Simulated Average Treatment Effects")+
    theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I summarized `sampling_distribution_sharp_null` by finding the proportion of average treatment effects in our sharp null thought experiment that are greater than our calculated estimate.

```r
sampling_distribution_sharp_null %>%
  summarize(
    p_value = mean(
      sampling_distribution_sharp_null$estimate > treat_effect$estimate
    ) # count of simulated estimates > than calculated estimate divided by total count
  )
```

```
##   p_value
## 1   0.553
```

> 55.3% percent of the average treatment effects simulated assuming there was no individual treatment effect are greater than our calculated estimate. This provides us with a p-value of 0.553, critically larger than the 0.05 threshold that would represent a statistically significant result. Therefore, we cannot reject the sharp null hypothesis and our calculated average treatment effect of -0.0517241 is not statistically significant.

Statistical Power Analysis

The power thought experiement asks: what is the probability of finding evidence in favor of an average treatment effect of size -0.0517241? I created function `statistical_power()` to assign potential outcomes for a control and treatment group, randomize treatment assignments, apply the switching equation to reveal actual outcomes, and finally return a difference-in-means estimate.

```
statistical_power <-
  function() {
    df %>%
      mutate(
        pot_outcome_control = rnorm(n(), sd = sd),
        pot_outcome_treatment = pot_outcome_control + treat_effect$estimate, # using dim estimate
        Z = sample(rep(c(0, 1), times = c(116, 232), size = n())), # randomize assignment
        Y = if_else(Z == 0, pot_outcome_control, pot_outcome_treatment)
      ) %>%
      difference_in_means(Y ~ Z, data = .) %>% # calculate average treatment effect
      tidy
  }
```

After setting my generation seed, I reran the `statistical_power()` function 1000 times and saved each to `sampling_distribution_power`.

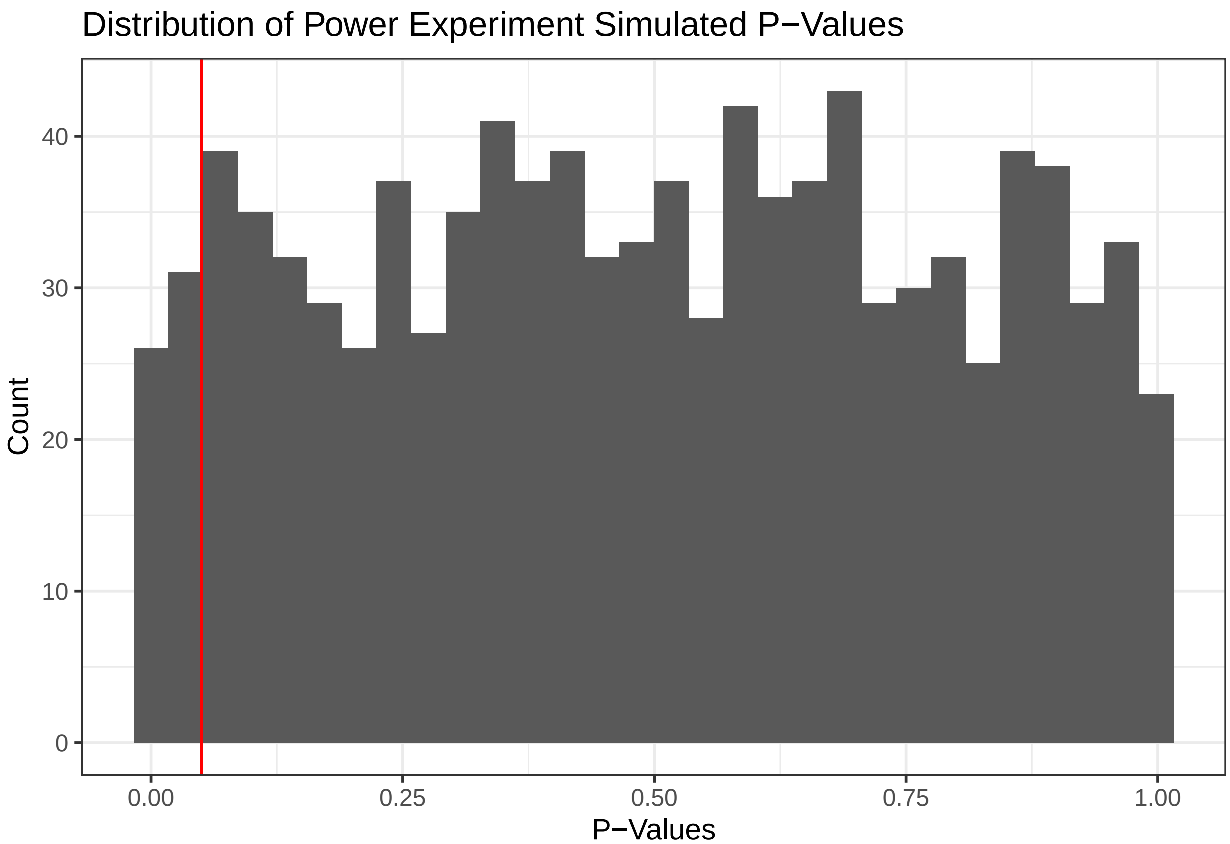
```
set.seed(1234)

sampling_distribution_power <-
  rerun(1000, statistical_power()) %>% # rerun the statistical_power function 1000 times
  bind_rows # saves each run
```

```
## Warning: `rerun()` was deprecated in purrr 1.0.0.
## i Please use `map()` instead.
##   # Previously
##   rerun(1000, statistical_power())
##
##   # Now
##   map(1:1000, ~ statistical_power())
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

I created a histogram of the `sampling_distribution_powerl` simulated p-values and marked our statistical significance level of 0.05, noting the quantity of p-values to the left of that vertical line. It appears an overwhelming majority of our simulated p-values are to the right of that line and greater than 0.05.

```
sampling_distribution_power %>%
  ggplot(aes(x = p.value)) + # plot the simulation's p-values
  geom_histogram() +
  geom_vline(xintercept = 0.05, col = "red") + # p-value of 0.05
  labs(x = "P-Values", y = "Count", # set plot labels
       title = "Distribution of Power Experiment Simulated P-Values") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



I summarized `sampling_distribution_power` by finding the proportion of p-values in our power thought experiment that are greater than the statistical significance level of 0.05.

```
sampling_distribution_power %>%
  summarize(
    statistical_power = mean(p.value < 0.05) # count of p-values < 0.05 divided by total count
  )
```

```
##   statistical_power
## 1             0.056
```

> Analyzing our data, there is only a 5.6% chance of finding evidence in favor of our small average treatment effect size. Our calculated statistical power is 0.056, critically lower than the 80% threshold that would make our experiment credible.

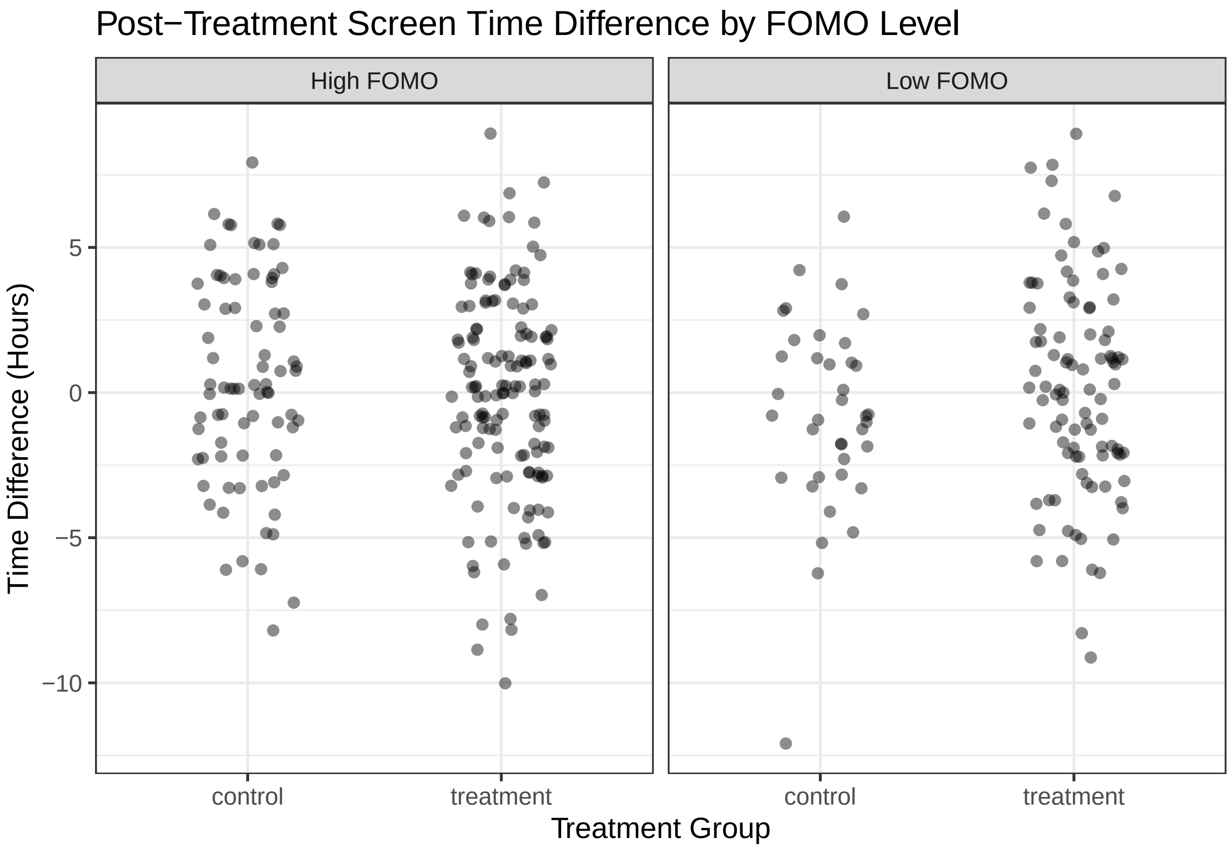
III. Secondary Outcomes

1. Norms: Fear of Missing Out

Did fear of missing out prevent individuals from cutting back on screen time after treatment? To investigate this question, I created another screen time difference vs. treatment group jitter plot, and I used the `facet_wrap()` function to split the visualization between those who feel strongly about their screen time usage being driven by the fear of missing out versus those who believe otherwise. At first glance, it appears those with high FOMO reduced their screen time, while those with low FOMO increased their screen time after treatment.

```
facet_labs <- c("high" = "High FOMO", "low" = "Low FOMO") # labels for plot facet

df %>%
  ggplot(aes(x = treat, y = diff_time)) + # plot primary outcome vs. treatment group
```

```
    geom_jitter(alpha = 0.45, height = 0.3, width = 0.2) + # adjusting jitter amount
    facet_wrap(~fomo_level, labeller = labeller(fomo_level = facet_labs)) + # fomo_level facet
    labs(x = "Treatment Group", y = "Time Difference (Hours)", # setting plot labels
         title = "Post-Treatment Screen Time Difference by FOMO Level") +
    theme_bw()
```



To confirm the visualization's results, I ran two difference-in-means estimates to calculate the average treatment effect of our experiment and subset the data to include only those with a "low" `fomo_level`, then only those with a "high" `fomo_level`.

```
difference_in_means(
    diff_time ~ treat, # compare our primary outcome to treatment group
    condition1 = "control", # ensure our control and treatment groups properly assigned
    condition2 = "treatment",
    subset = fomo_level == "low", # subset our analysis to those with "low" fomo_level
    data = df
)
```

```
## Design:  Standard
##                  Estimate Std. Error  t value  Pr(>|t|)   CI Lower CI Upper
## treattreatment 0.8526316  0.6569777 1.297809 0.1983002 -0.4559722 2.161235
##                      DF
## treattreatment 75.57369
```

```
difference_in_means(
    diff_time ~ treat, # compare our primary outcome to treatment group
    condition1 = "control", # ensure our control and treatment groups properly assigned
    condition2 = "treatment",
```

```
    subset = fomo_level == "high", # subset data to those with "high" fomo_level
    data = df
)
```

```
## Design:  Standard
##                  Estimate Std. Error   t value Pr(>|t|)  CI Lower CI Upper
## treattreatment -0.5215235  0.5022141 -1.038448 0.300621 -1.513313 0.470266
##                      DF
## treattreatment 160.7226
```

> The average treatment effect on those with a "low" `fomo_level` was 0.8526316, confirming the slight increase visualized in the jitter plot. The average treatment effect on those with a "high" `fomo_level` was -0.5215235, confirming the slight decrease visualized in the jitter plot. It appears fear of missing out may have had a small effect in the opposite direction than anticipated. We would expect those with a "low" `fomo_level` to have an estimated average treatment effect below zero, not above. However, following the previous sharp null and statistical power thought experiments, these results are not statistically significant.

2. Beliefs: Link Between Screen Time and Productivity

Did the belief that calculating and monitoring screen time would not have beneficial outcomes prevent individuals from cutting back on screen time after treatment? To investigate this question, I created another screen time difference vs. treatment group jitter plot, and I used the `facet_wrap()` function to split the visualization between those who believed lower screen time is correlated with increased productivity versus those who don't accept that claim. I also mapped responses to the `effect_prod` variable to color. At first glance, it appears those who believed there is no correlation reduced their screen time, while those who believe there is a correlation slightly increased their screen time despite accepting that correlation.

```
facet_labs <- c(`0` = "No Correlation", `1` = "Correlation") # labels for plot facet

df %>%
  ggplot(aes(x = treat, y = diff_time, col = effect_prod)) + # map color to effect_prod
    geom_jitter(alpha = 0.45, height = 0.3, width = 0.2) + # adjust jitter amount
    facet_wrap(~correlate_prod, labeller = labeller(correlate_prod = facet_labs)) + # correlate_prod fa
    labs(x = "Treatment Group", y = "Time Difference (Hours)", # set plot labels
         title = "Screen Time Difference by Correlation with Productivity") +
    theme_bw()
```

To confirm the visualization's results, I ran two difference-in-means estimates to calculate the average treatment effect of our experiment and subset the data to include only those who did not believe in the correlation (responded "0" to `correlate_prod`), then only those who did believe in the correlation (responded "1" to `correlate_prod`).

```
difference_in_means(
  diff_time ~ treat, # compare our primary outcome to treatment group
  condition1 = "control", # ensure our control and treatment groups properly assigned
  condition2 = "treatment",
  subset = correlate_prod == 0, # subset data to those who don't correlate productivity with screen tim
  data = df
)
```

```
## Design:  Standard
##                  Estimate Std. Error   t value  Pr(>|t|)  CI Lower CI Upper
## treattreatment -0.6251526  0.5944427 -1.051662 0.2951796 -1.802743 0.552438
##                      DF
## treattreatment 113.9614
```

```
difference_in_means(
  diff_time ~ treat, # compare our primary outcome to treatment group
  condition1 = "control", # ensure our control and treatment groups properly assigned
  condition2 = "treatment",
  subset = correlate_prod == 1, # subset data to those who don't correlate productivity with screen tim
  data = df
)
```

```
## Design:  Standard
##                 Estimate Std. Error  t value Pr(>|t|)   CI Lower CI Upper
```

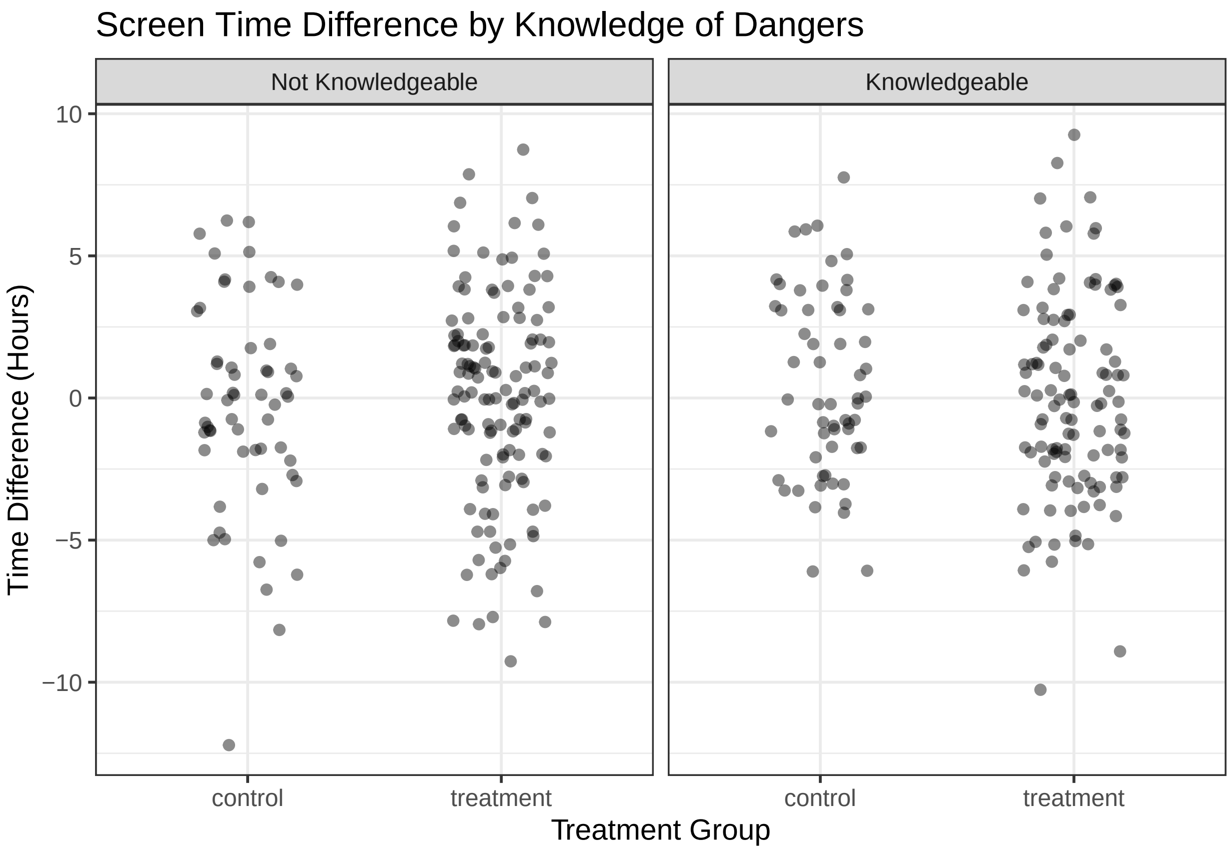
```
## treattreatment 0.4978261  0.5394629 0.922818 0.357946 -0.5702267 1.565879
##                      DF
## treattreatment 120.5129
```

The average treatment effect on those who did not believe in the correlation was -0.6251526, confirming the slight decrease visualized in the jitter plot. The average treatment effect on those who did believe in the correlation was 0.4978261, confirming the slight increase visualized in the jitter plot. It appears this correlation belief may have had a small effect in the opposite direction than anticipated. We would expect those who believe a correlation exists between lower screen time and increased productivity to have an average treatment effect below zero, not above. However, following theprevious sharp null and statistical power thought experiments, these results are not statistically significant.

3. Lack of Knowledge: Dangers of Social Media Use / Screen Time

Did the lack of knowledge of the of the dangers of social media use or excessive screen time prevent individuals from cutting back on screen time after treatment? To investigate this question, I created another screen time difference vs. treatment group jitter plot, and I used the `facet_wrap()` function to split the visualization between those who knew that adolescents are at risk of mental health struggles if they use social media for more than 3 hours a day versus those who didn't. At first glance, it is difficult to make suggestions about the data. Those who were not knowledgeable appeared to slightly increase their screen time, while those who were knowledgeable appeared largely unmoved or slightly decreased their screen time.

```r
facet_labs <- c(`0` = "Not Knowledgeable", `1` = "Knowledgeable") # labels for plot facet

df %>%
  ggplot(aes(x = treat, y = diff_time)) + # plot primary outcome vs. treatment group
    geom_jitter(alpha = 0.45, height = 0.3, width = 0.2) + # adjust jitter amount
    facet_wrap(~adolescents, labeller = labeller(adolescents = facet_labs)) + # adolescents facet
    labs(x = "Treatment Group", y = "Time Difference (Hours)", # set plot labels
         title = "Screen Time Difference by Knowledge of Dangers") +
    theme_bw()
```



I ran two difference-in-means estimates to calculate the average treatment effect of our experiment and subset the data to include only those who were not knowledgeable about the dangers of social media use or excessive screen time (responded "0" to `adolescents`), then only those who were knowledgeable about these dangers (responded "1" to `adolescents`).

```
difference_in_means(
  diff_time ~ treat, # compare our primary outcome to treatment group
  condition1 = "control", # ensure our control and treatment groups properly assigned
  condition2 = "treatment",
  subset = adolescents == 0, # subset data to those who are not knowledgeable about dangers
  data = df
)
```

```
## Design:  Standard
##                 Estimate Std. Error   t value  Pr(>|t|)   CI Lower CI Upper
## treattreatment 0.3954394  0.5876664 0.6728977 0.5024195 -0.7691403 1.560019
##                      DF
## treattreatment 110.3175
```

```
difference_in_means(
  diff_time ~ treat, # compare our primary outcome to treatment group
  condition1 = "control", # ensure our control and treatment groups properly assigned
  condition2 = "treatment",
  subset = adolescents == 1, # subset data to those who are knowledgeable about dangers
  data = df
)
```

```
## Design:  Standard
##                   Estimate Std. Error    t value  Pr(>|t|)  CI Lower  CI Upper
```

```
## treattreatment -0.5051086  0.5451633 -0.9265271 0.3559427 -1.583966 0.5737493
##                       DF
## treattreatment 126.0506
```

The average treatment effect on those who were not knowledgeable was 0.3954394. The average treatment effect on those were knowledgeable was -0.5051086. It appears lack of knowledge may have had a small effect in the predictable direction: those who were not knowledgeable having an average treatment effect closer to or above 0, while those who were knowledgeable having an average treatment effect below 0. However, again, following the previous sharp null and statistical power thought experiments, these results are not statistically significant.