Consolidated Argument

Null Result Penalty Replication

Ryan McWay Emily Kurtz

2025-03-18

Direct Replication

The direct replication was sucessful. But the paper seems almost too good to be true. The point of the paper is the null results are penalized for publication. Yet all the results, even the appendix results, have huge statistically significant effects.

This is strange get the sample. They survey economists and ask them if they would publish a paper. This is measured on a sliding scale of 0 to 100. They provide each person with four of five vignettes. The authors take the vigenttes from real studies that are statistically significant and published. They keep the standard errors the same, but randomize if they shift the coefficient left in the distribution such that the effect is now statistically insignficant.

The get a sample of 480 respondents who complete four vigenettes for 1920 observations. On top of that they cross-randomize 6 other attributes of the vigenettes. Aspects such as gender, prestige, etc. could effect if the finding is publishable beyond statistical significance. This produces 48 treatment assignments using a factorial design. In practice, the authors have 40 observations per treatment assignment to identify off of – 10 respondents. Despite these small clusters, the standard errors are tiny. This makes us suspiciuos.

As part of the reproduction, we identify Table 3 and Figure 2 has presenting the main effects. Table 3 is of primary interest as it estimates the null result effect on the primary outcome of interest and the secondary outcomes. Figure 2 estimates the interaction effect of the null effect with the cross-randomized characteristics of the vigenettes. Below we represent a reproduction of the main estimate – Column 1 of Table 3. In addition, we are able to reproduce all the results from the replication packet provided by the authors using the original Stata code.

```
# Column 1
col1 <- plm(publish ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
```

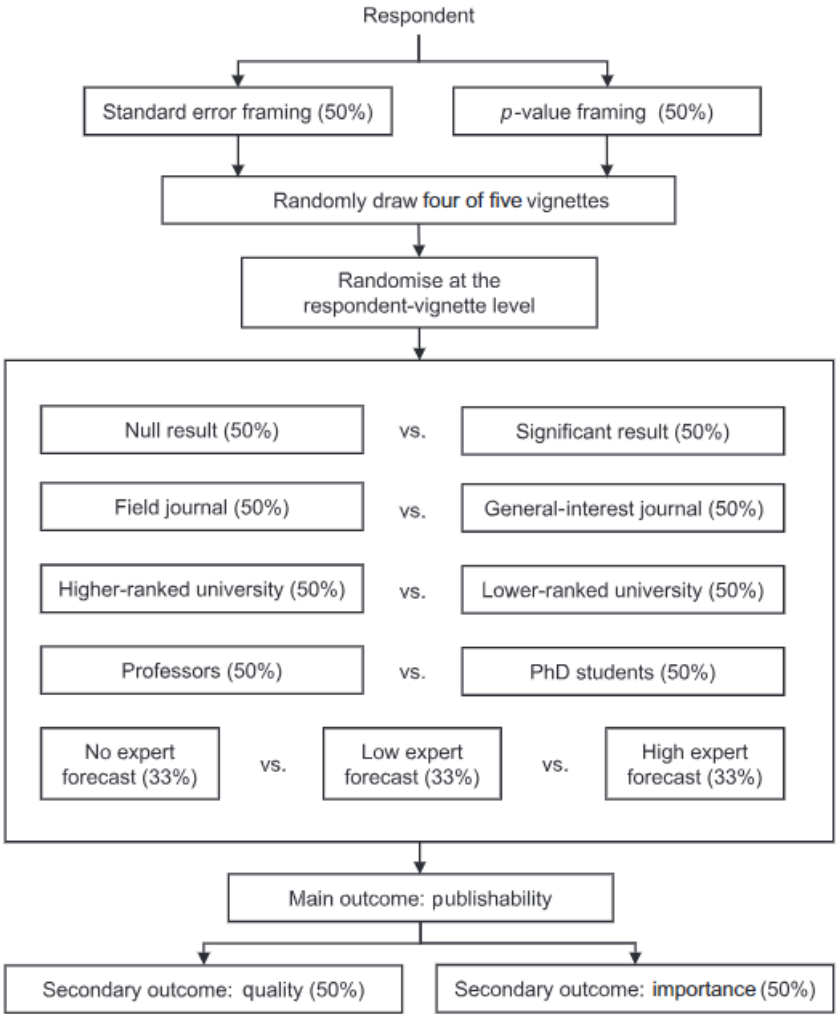


Figure 1: Factorial Design

```
                model = "within")
col1_se <- sqrt(diag(vcovHC(col1, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
df_control <- subset(df, df$low == 0)
col1_mean <- round(mean(df_control$publish), 3) # Subset for control
# Present
stargazer(col1,
        type = "text",
        keep = c(1),
        covariate.labels = c("Null result treatment"),
        se = list(col1_se),
        keep.stat = c("n", "adj.rsq"),
        model.numbers = TRUE,
        digits = 3,
        add.lines = list(c("Mean Dep. Var.", col1_mean)
        ))
```

```
================================================
                        Dependent variable:
                    ---------------------------
                               publish
------------------------------------------------
Null result treatment        -14.054***
                               (1.099)

------------------------------------------------
Mean Dep. Var.                 57.193
Observations                    1,920
Adjusted R2                    -0.070
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

We examine this in a couple of ways in particular.

1. Z-curve is off the chart in comparision to other RCT publications in economics
2. Variation in the dependent variable is strange
3. Contrasted to secondary outcomes, appears more strange
4. Sample composition (Maybe as robustness check)

The motivation for these robustness checks are to stress test the results in examining if there is potential data manipulation that ensures statistical significance. Our current results suggest

that the data is unlikely to have been generated from real world data. The recommendation of this replication is that Chopra et al. (2023) should be replicated using new data with an independent team of researchers.

Z-curve

NOTE: Worth discussing

The results of this study are very robust. The very large t-statistics are driving the results. For an RCT, we would not expect the results to be so robust. In fact, this article appears to be a gold mine for statistical significance. If we compare this to the distribution of RCT z-scores found in economics by Brodeur et al. (2020), we note that the z-scores in this study are far on the right tail (https://www.aeaweb.org/articles?id=10.1257/aer.20190687). In expecation, we should rarely find that the treatment effects are so statistically significant. Further, the authors note the ex-post power calculation that suggests that this study is underpowered to detect a measurable effect. And yet, each robustness check confirms a large and statistically significant effect.

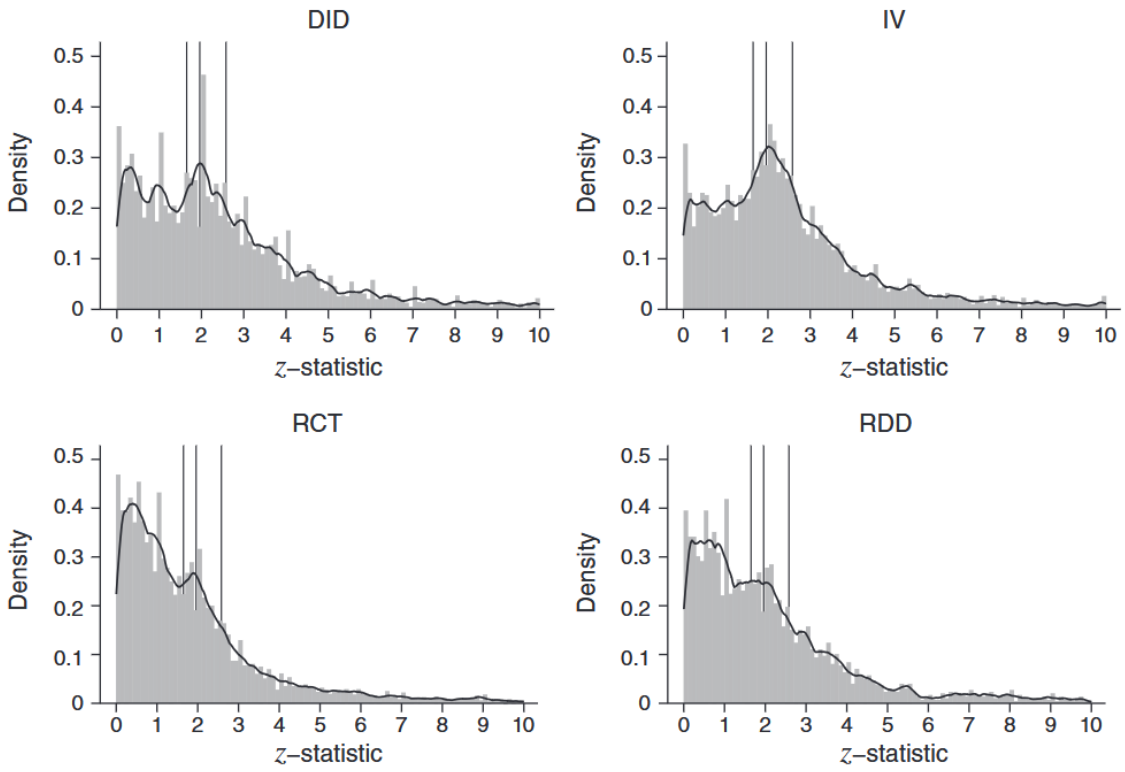


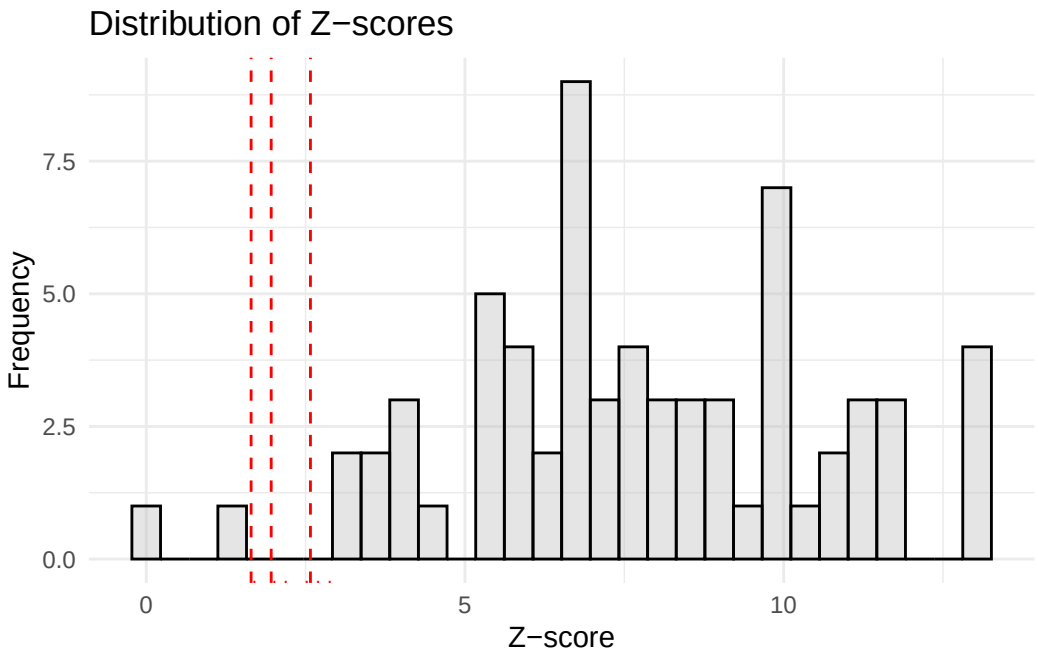
Figure 2: RCT Z-curve in Economics

To compare the statitical significance of this study to RCTs reported in top economics journals, we display two z-curves. The first is the z-curve for reported results in the main study and the online appendix (delimited as two curves). And the second is a z-curve with the results from our replication robustness checks. To do this, I take the ratio of the point estimate (beta) to the variation in the estimate (standard) error to approximate the t-statistic. I will focus specifically on the pre-specified primary outcome of interest: publishability.

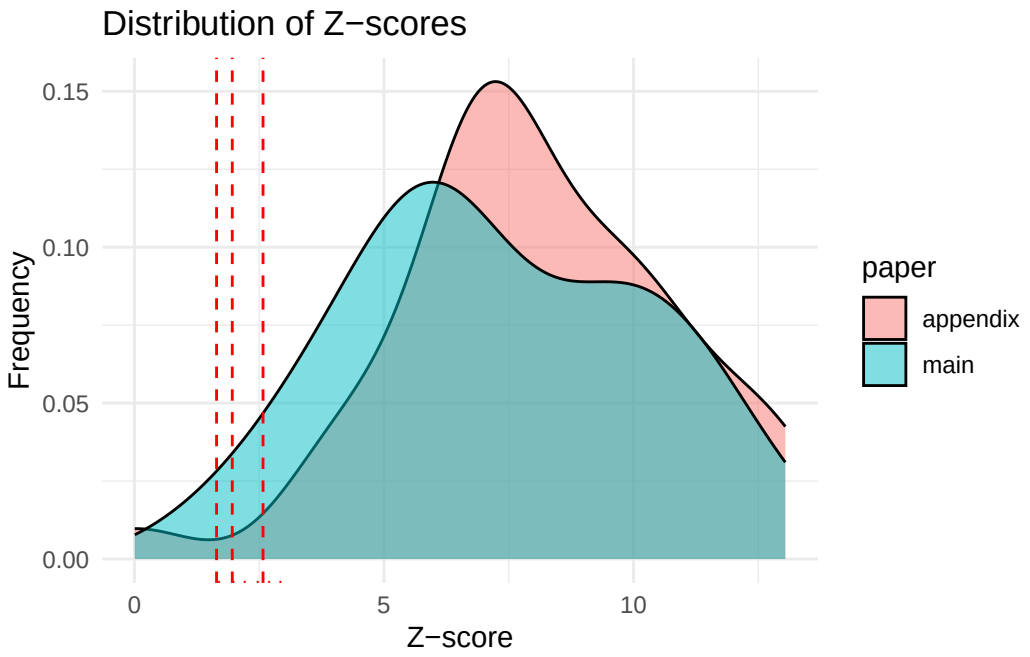
```
# Create main paper list of values
df_main_values <- data.frame(
    table = c(3,3,4,4,4,4,4,
            NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,
            NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,
            5, 5),
    figure = c(NA,NA,NA,NA,NA,NA,NA,
            3,3,3,3,3,3,3,3,3,3,
            2,2,2,2,2,2,2,2,2,2,
            NA, NA),
    panel = c("a", "b", "a", "b", "c", "d", "e",
            "labor", "public", "development", "political", "finance", "experimental", "beh
            "male", "female", "phd", "prof", "editor", "noeditor", "highcite", "lowcite",
            "a", "b"),
    beta = c(-14.058, -14.474, -11.239, -14.571, -14.945, -14.320, -11.96,
            -14.875, -13.368, -20.716, -12.202, -14.601, -12.595, -11.191, -6.617, -13.961
            -13.947, -15.128, -14.082, -14.199, -15.610, -14.278, -15.714, -14.065,  -14.7
            -19.755, -18.134),
    se = c(1.09, 1.224, 1.913, 1.465, 1.491, 1.48, 1.736,
            2.666, 3.182, 3.165, 2.295, 4.006, 4.297, 3.374, 4.635, 2.621, 3.018,
            1.241, 2.495, 2.311, 1.287, 2.707, 1.280, 1.747, 1.892, 1.866, 1.431,
            2.269, 2.605),
    obs = c(1920, 1920, 1920, 1920, 1920, 1920, 1920,
            352, 216, 300, 280, 176, 104, 152, 112, 236, 236,
            1488, 420, 456, 1412, 268, 1508, 656, 656, 628, 1220,
            475, 475)
)
df_main_values["paper"] <- "main"

# Create appendix list of values
df_appendix_values <- data.frame(
    table = c(NA, NA, NA, NA, NA,
            "A1", "A1", "A1", "A1", "A1", "A1", "A1", "A1", "A1", "A1",
            "A2",
            "A3", "A3", "A3", "A3",
```

```
        "A5",
        "A6", "A6", "A6", "A6",
        "A7", "A7", "A7", "A7", "A7",
        "A8",
        "A9", "A9", "A9",
        "A12", "A12", "A12", "A12"),
figure = c("A2", "A2", "A2", "A2", "A2",
        NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
        NA,
        NA, NA, NA, NA,
        NA,
        NA, NA, NA, NA,
        NA, NA, NA, NA, NA,
        NA,
        NA, NA, NA,
        NA, NA, NA, NA),
panel = c("vignette1", "vignette2", "vignette3", "vignette4", "vignette5",
        "a1", "a2", "a3", "a4", "a5", "b1", "b2", "b3", "b4", "b5",
        "1",
        "a", "b", "c", "d",
        "1",
        "a", "b", "c", "d",
        "a", "b", "c", "d", "e",
        "1",
        "a", "b", "c",
        "a", "b", "c", "d"),
beta = c(-15.249, -17.783, -11.268, -11.789, -14.295,
        -11.754, -11.702, -12.009, -11.960, -11.161, -11.924, -11.828, -12.305, -12.19
        -15.735,
        -14.058, -14.131, -14.474, -14.628,
        -16.242,
        -14.058, -11.486, -15.336, -13.417,
        -11.239, -14.571, -14.945, -14.32, -11.96,
        -11.072,
        -13.202, -13.206, -18.067,
        -14.685, -15.518, -16.654, -16.986),
se = c(2.289, 2.231, 2.154, 2.266, 2.244,
        1.783, 1.777, 1.745, 1736, 3.063, 1.585, 1.533, 1.491, 1.432, 2.681,
        2.232,
        1.09, 1.286, 1.224, 1.471,
        2.133,
```

```
            1.09, 1.569, 2.27, 1.897,
            1.913, 1.465, 1.491, 1.48, 1.736,
            2.681,
            1.253, 1.457, 2.117,
            1.126, 1.32, 1.575, 1.853),
    obs = c(387, 377, 385, 389, 382,
            1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920,
            502,
            480, 480, 480, 480,
            480,
            480, 480, 284, 348,
            1920, 1920, 1920, 1920, 1920,
            1920,
            1920, 988, 566,
            1788, 1360, 884, 640)
)
df_appendix_values["paper"] <- "appendix"

# Append datasets denoting origin of values
df_values = setDT(rbind(df_main_values, df_appendix_values))

# Create z-scores from values
df_values = df_values[, zscore := abs(beta/se)]
# Create z-curve density plot with both density and frequency on y-axis
ggplot(df_values, aes(x = zscore)) +
    geom_histogram(fill = "gray", color = "black", alpha = 0.4) +
    geom_vline(xintercept = c(1.645, 1.96, 2.576), linetype = "dashed", color = "red") +
    annotate("text", x = 1.7, y = 0, label = "*", vjust = 2, color = "red") +
    annotate("text", x = 2.1, y = 0, label = "**", vjust = 2, color = "red") +
    annotate("text", x = 2.7, y = 0, label = "***", vjust = 2, color = "red") +
    labs(title = "Distribution of Z-scores",
        x = "Z-score",
        y = "Frequency") +
    theme_minimal()
```




```r
# Density plot
ggplot(df_values, aes(x = zscore)) +
    # geom_density(alpha = 0.5, fill = "gray") +
    geom_density(alpha = 0.5, aes(fill = paper)) +
    geom_vline(,xintercept = c(1.645, 1.96, 2.576), linetype = "dashed", color = "red") +
    annotate("text", x = 1.7, y = 0, label = "*", vjust = 2, color = "red") +
    annotate("text", x = 2.1, y = 0, label = "**", vjust = 2, color = "red") +
    annotate("text", x = 2.7, y = 0, label = "***", vjust = 2, color = "red") +
    labs(title = "Distribution of Z-scores",
        x = "Z-score",
        y = "Frequency") +
    theme_minimal()
```



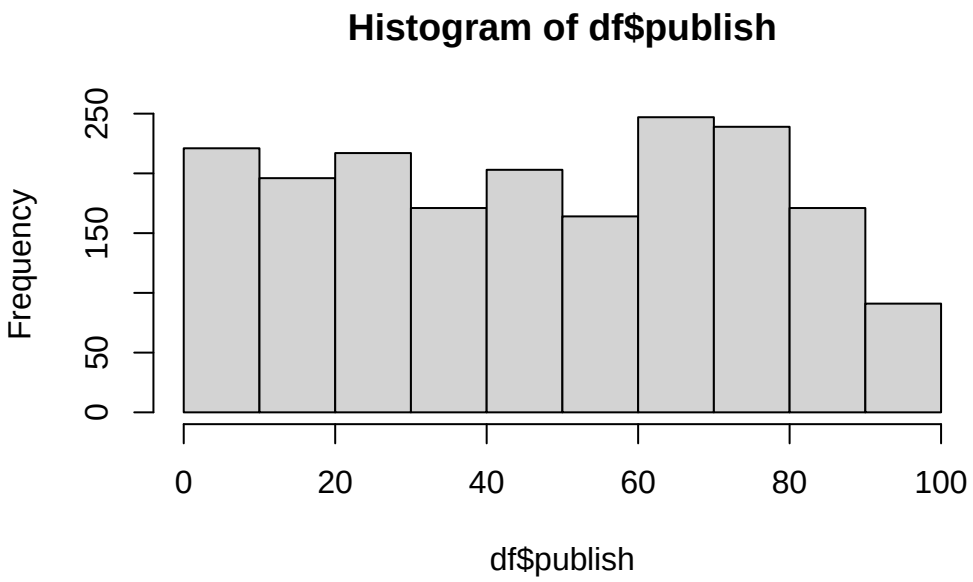

```
# Create list of values
```

Variation in the Publishability

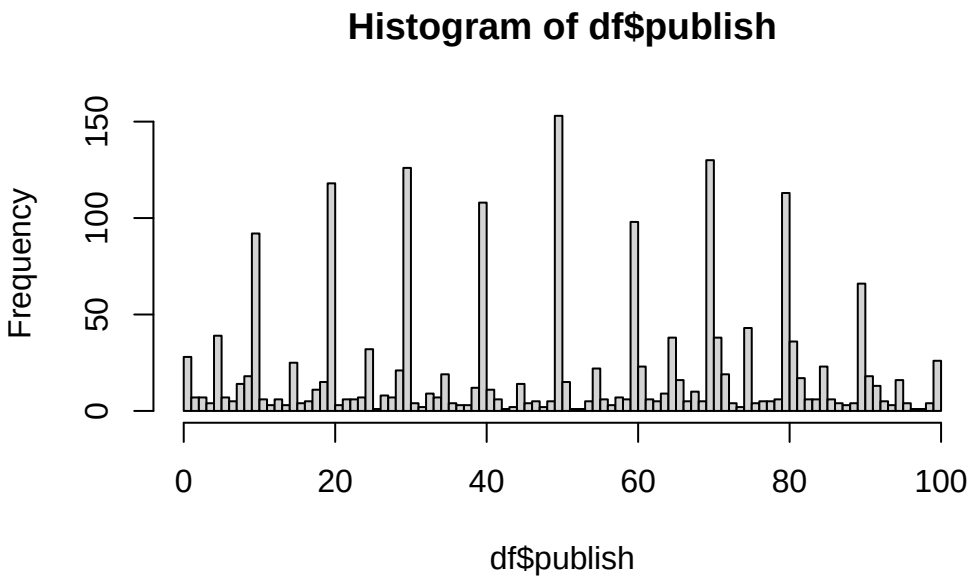
The first thing that we note is the distribution of the primary outcome of interest – publishability.

The first thing that is strange is that the outcome measure appears to be uniformly distributed. That is a bit odd. Without binning, we also see that there is some grouping around divisors of 5 along the sliding scale used by respondents.

```
# Suspiciously uniform
hist(df$publish, breaks = 10)
```

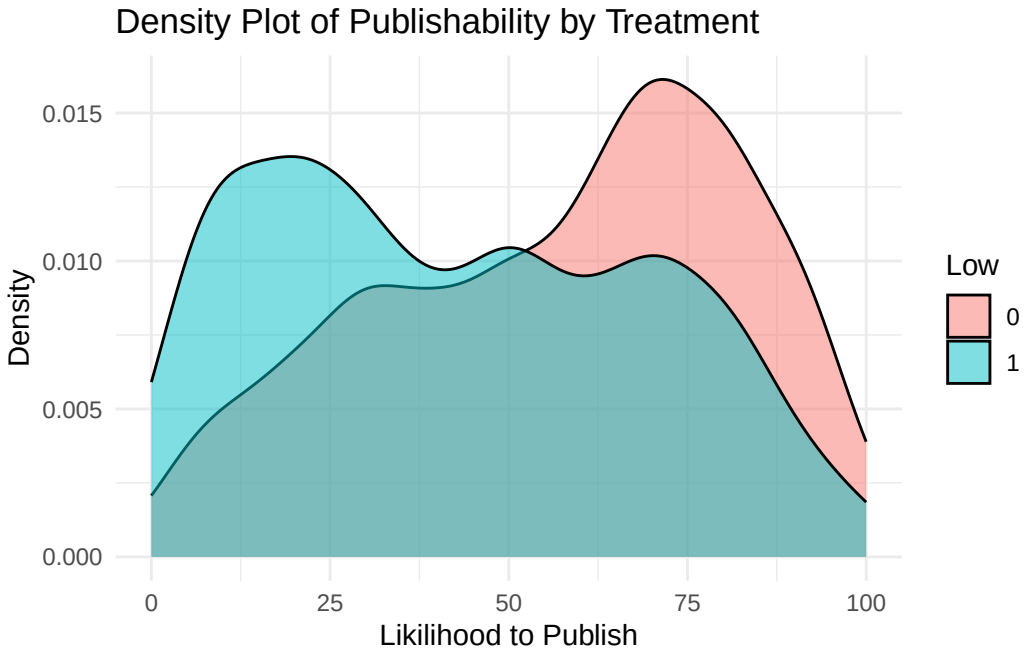



```r
# Bunching around specific values
hist(df$publish, breaks = 100) # Lots of grouping on individual values
```



We notice something strange when we examine the distribution of the outcome measure when highlighting treatment assignment. Notably, the control and treatment distributions look like mirrors of one another.

```
# Publish by treatment
ggplot(df, aes(x = publish, fill = factor(low))) +
    geom_density(alpha = 0.5) +
    labs(title = "Density Plot of Publishability by Treatment",
         x = "Likilihood to Publish",
         y = "Density",
         fill = "Low") +
    theme_minimal()
```



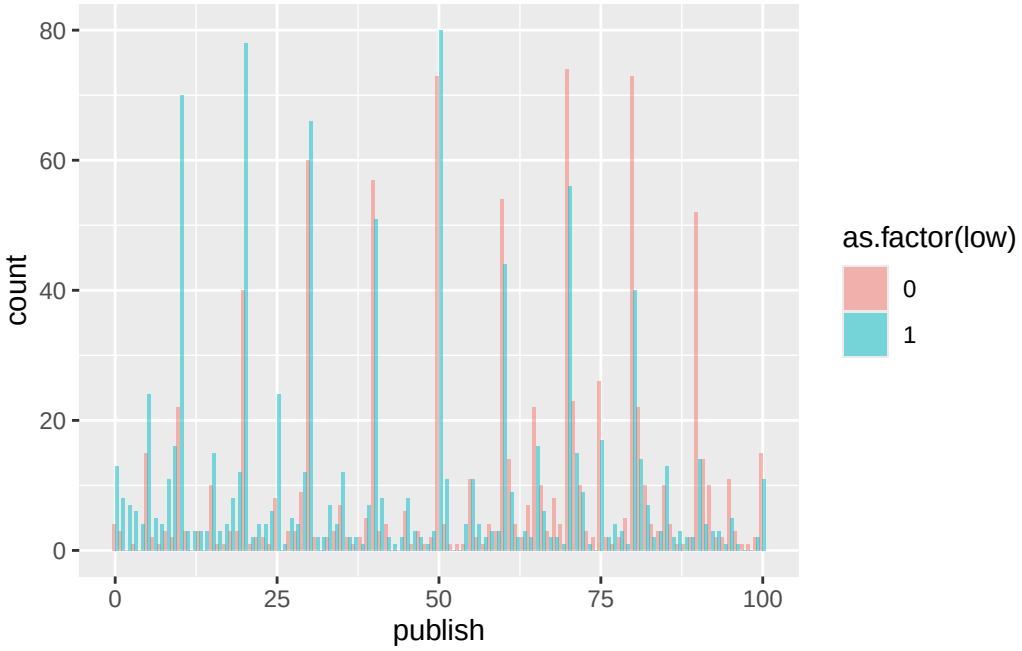
We suspect that treatment and control are the same distribution but systemtric about the middle of the range (50). In context, this is meaninful as 50 can be interperted as the threshold between publishing and not publishing the article. When we flip the control group distribution by the forumla $[publish|t_i = 0] = 100 - publish$ we find that treatment and control have the same distribution. This suggests that the data could have been generated from a random distribution rather than real data. In particular, this appears to be a Beta distribution. Using the following formula for the probability distribution function, you could reproduce the underlying data, split the sample in half, and flip the 'control' group about the range to create a reflection. With this reflection, we could produce the results from the Chopra et al. paper

without collecting any data.

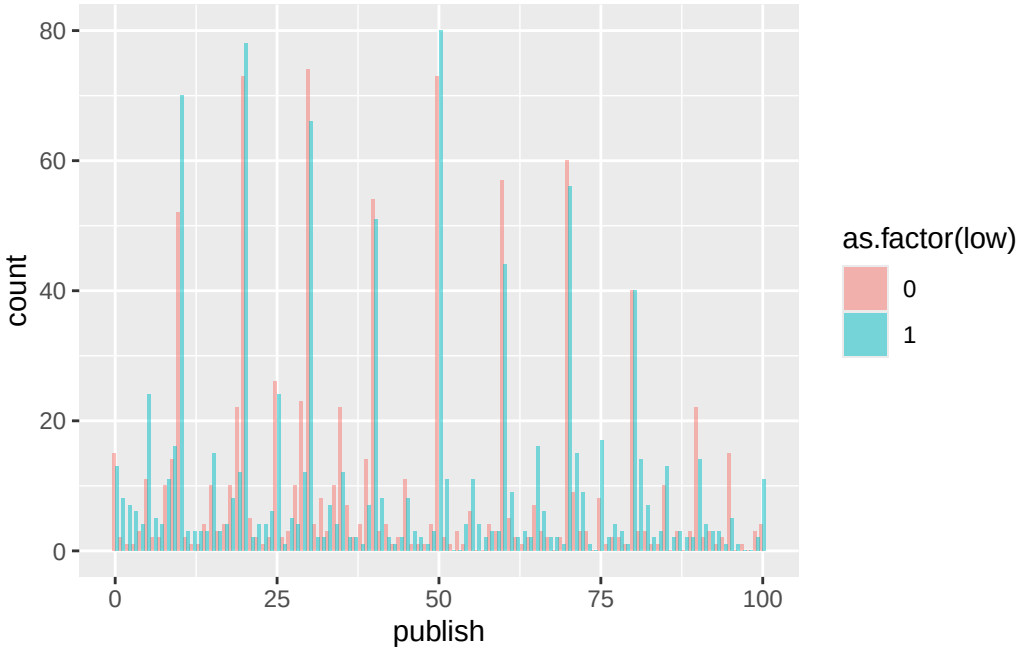
> The PDF for the beta distribution, for $0 \leq x \leq 1$, uses the shape parameters $\alpha, \beta > 0$ to create a power function of some variable x. The denominator is normalization to ensure total probability of 1.

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1} du}$$

```
# Histogram overlaying control onto treatment
df2 <- data.table::copy(df)
df2[, publish := ifelse(low == 1, publish, 100 - publish)]

# Regular historgram by treatment
ggplot(df, aes(publish, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Histogram after flip the scale (e.g., are the symettric about the average (50))
ggplot(df2, aes(publish, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```



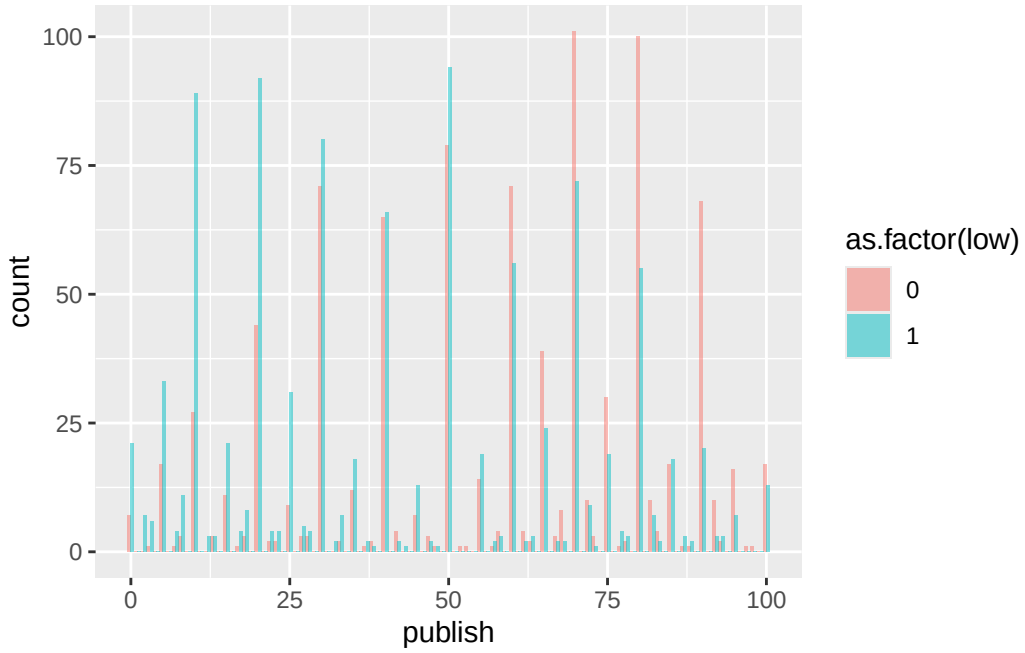

```
# This looks awfully symettric...
```

One thing that the authors could have done to make the data appear more 'realistic' is to 'jitter' the data in the distrbution and apply a heurisitic for how participants would select values. Suppose that we expect people to tend to select items that are multiple of 5s or 10s. Then I could just create this Beta distribution as a discrete function with intervals of fives. For the formula above, instead of $\int$ you could replace it with $\sum_i^n f(5i)$ to make this discrete distribution. Because this would be too neat, the authors may add some noise. Specifically, values that are not multiples of 5, as well as adding values near multiples of 5 to show human errors.

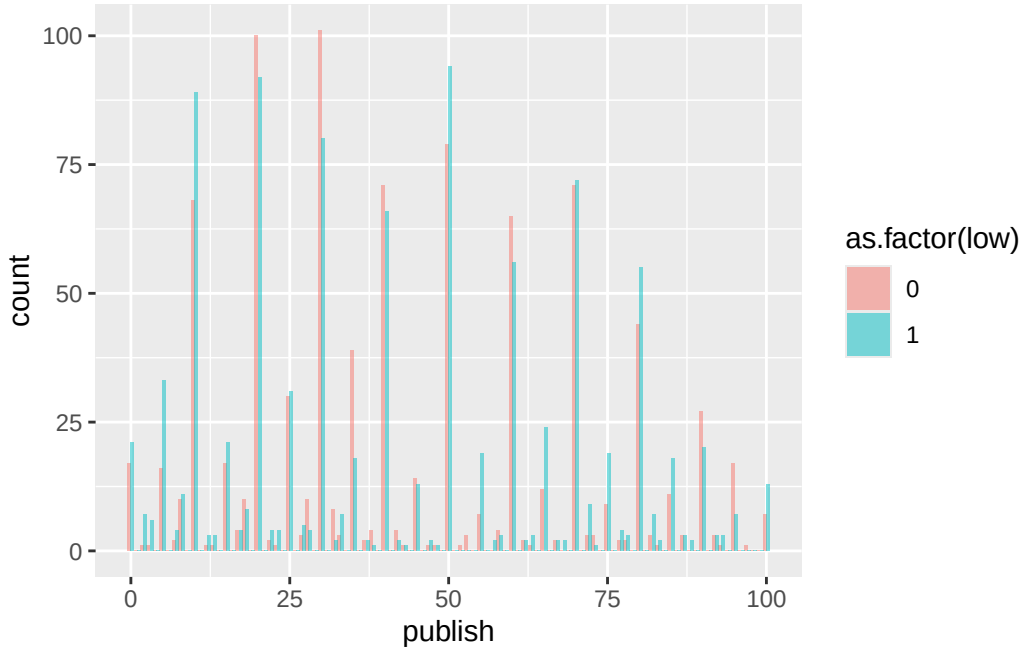
We account for this in our descriptive of the distribution by recoding values near divisors of 5 to the nearest divisor. As a bandwidth, we recode values that are 1 value away. For example, if you have a uniform distrbution from 5 to 10 you would expect the observations: $5, 6, 7, 8, 9, 10$. Using our bandwidth to recode we will now have the observations $5, 5, 7, 8, 10, 10$. In a uniform distribution, that means that rather than a $2/6$ chance of selection for divisors of 5 there is now a $4/6$ chance of divisors of 5. The increased likelihood should apply similiarly to the Beta distribution.

```
# Recode values within bandwidth
df = df[, publish := fifelse(publish %% 5 == 0, publish,
                fifelse(publish %% 5 == 1, publish - 1,
```

```
                fifelse(publish %% 5 == 4, publish + 1, publish)))]

# Same for overlay data set
df2 = df2[, publish := fifelse(publish %% 5 == 0, publish,
                fifelse(publish %% 5 == 1, publish - 1,
                fifelse(publish %% 5 == 4, publish + 1, publish)))]

# Replot
ggplot(df, aes(publish, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1) # As presented
```




```
ggplot(df2, aes(publish, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1) # With overlay
```



Content to add here Emily: - Details on distribution from other slider bars. In particular if we could get some that are from other studies predicting things on a slider from 0 to 100. - The empircal tests: kolmogorov Smirnov test

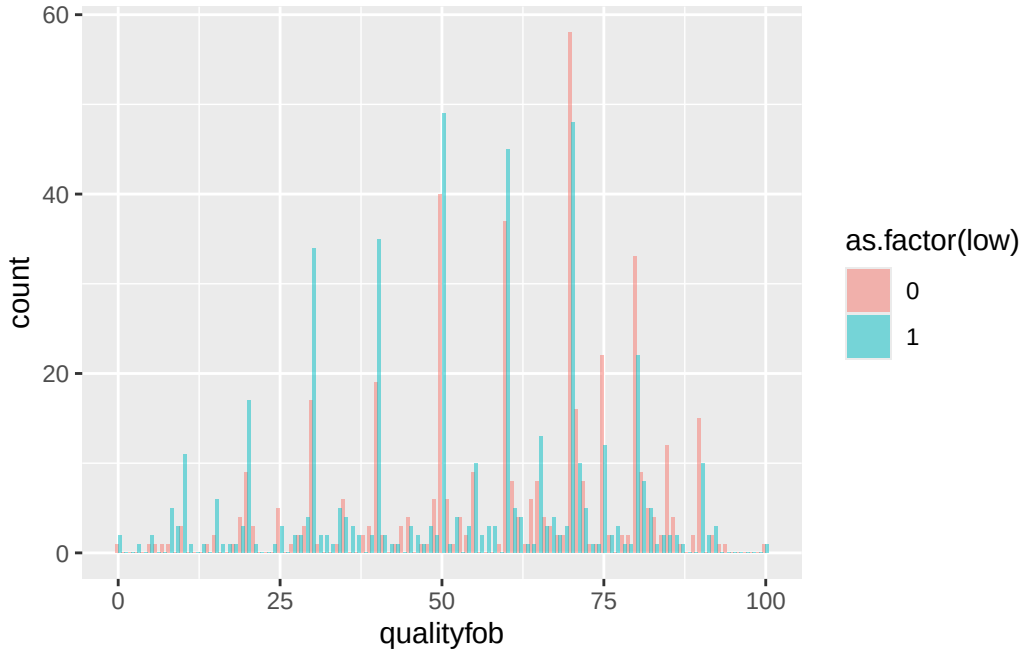
Variation in Secondary Outcomes

NOTE: Worth discussing in contrast to variation in primary outcome

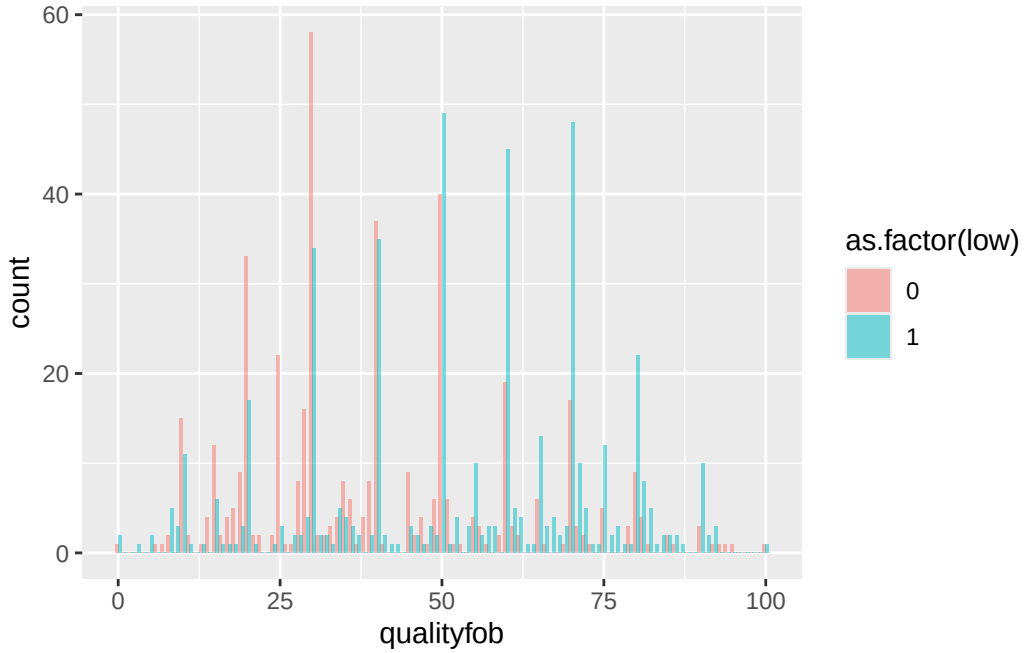
In brief, we do not find signs of potential manipulation for the the secondary outcomes like we do for the primary outcome for publication. We examine them through similar replication of Table 3 results and exploring histograms for the secondary outcomes. These histograms are presenting the opposite relationship that was found for the primary outcome. There is considerable overlap in the original distribution – a more reasonable generated data set. Note that z-scores are what are estimated in the paper. So I present those histograms and then show esimates of Table 3 for the secondary outcomes before and after the z-score moditification.

- First order quality

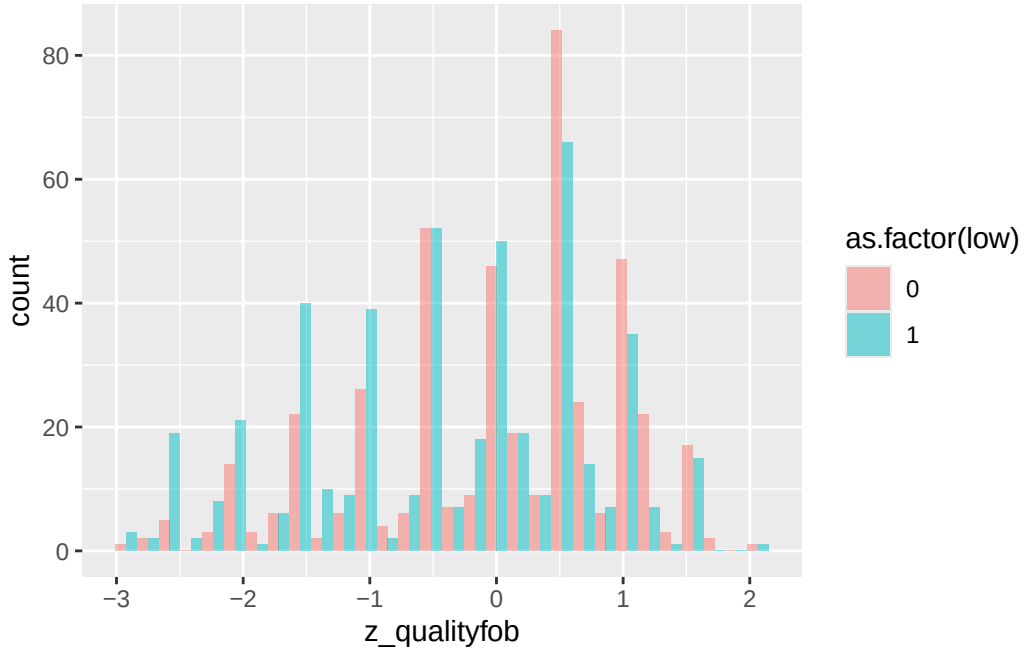
```
# Original
ggplot(df, aes(qualityfob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Fliped
df2[, qualityfob := ifelse(low == 1, qualityfob, 100 - qualityfob)]
ggplot(df2, aes(qualityfob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```

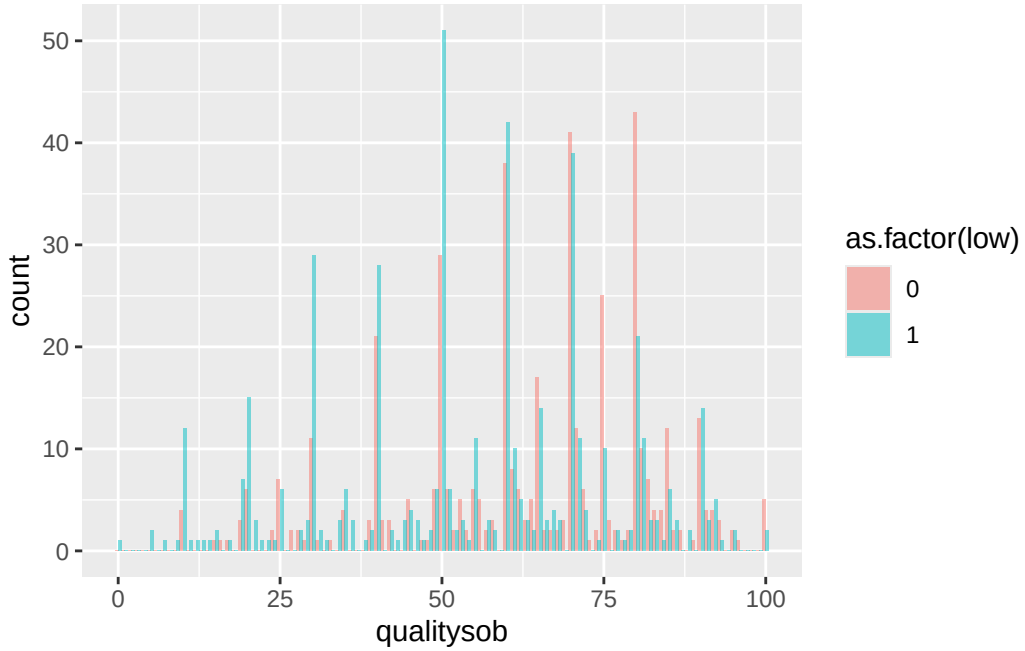



```
# Z-score
ggplot(df, aes(z_qualityfob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge')
```

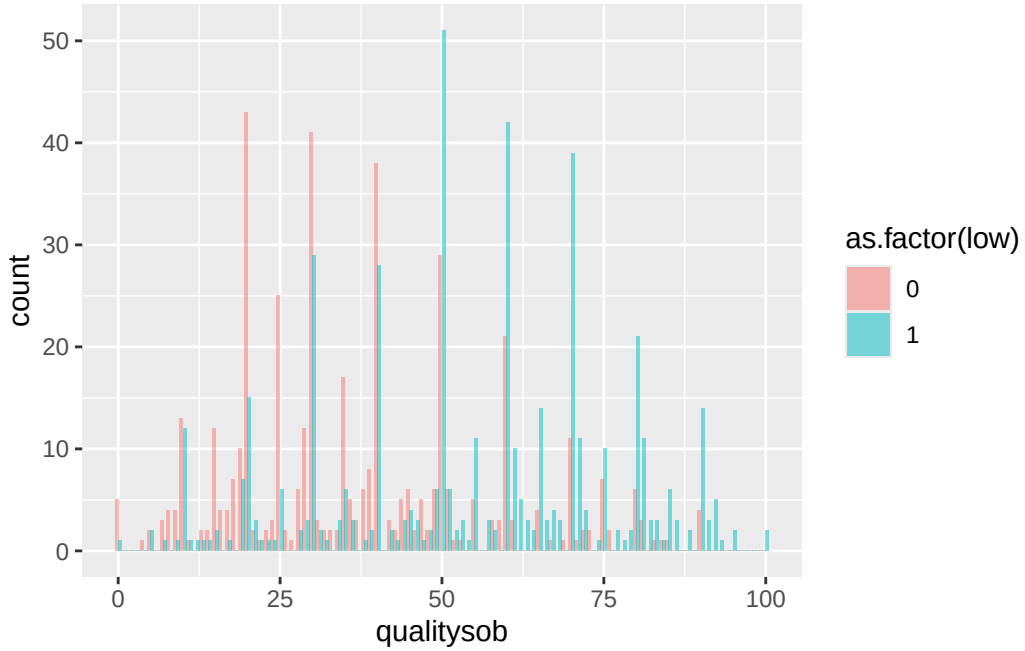


- Second order quality

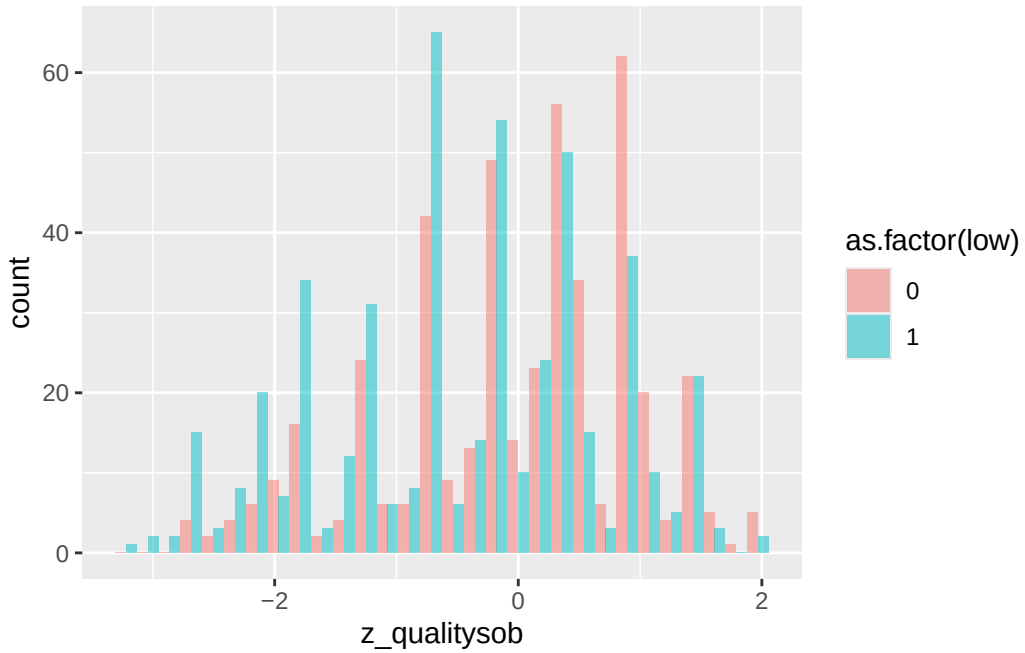
```
# Original
ggplot(df, aes(qualitysob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Fliped
df2[, qualitysob := ifelse(low == 1, qualitysob, 100 - qualitysob)]
ggplot(df2, aes(qualitysob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```

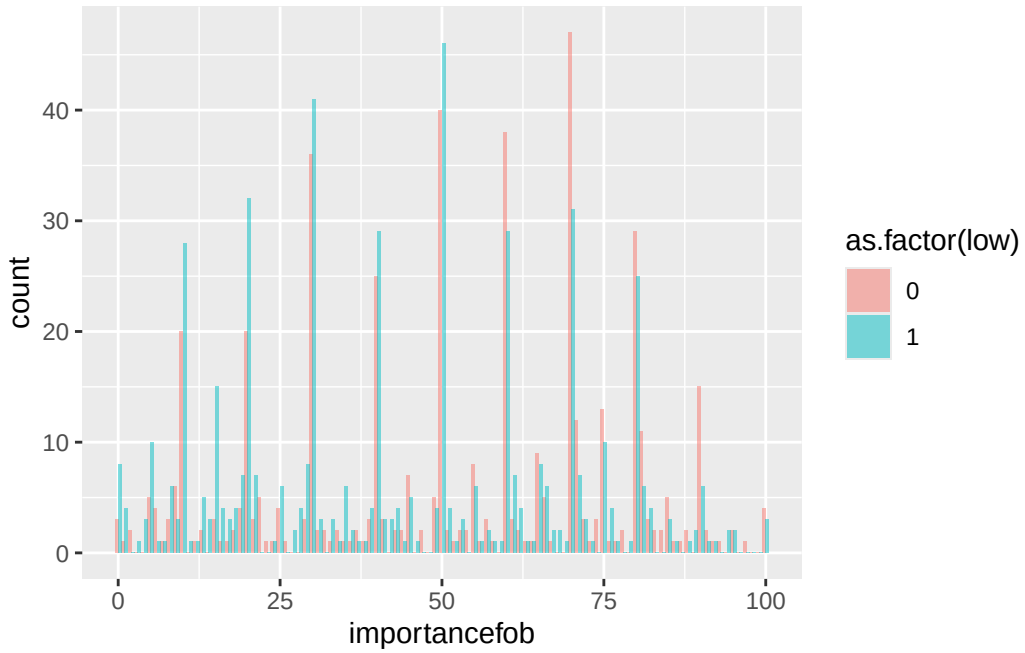



```
# Z-score
ggplot(df, aes(z_qualitysob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge')
```

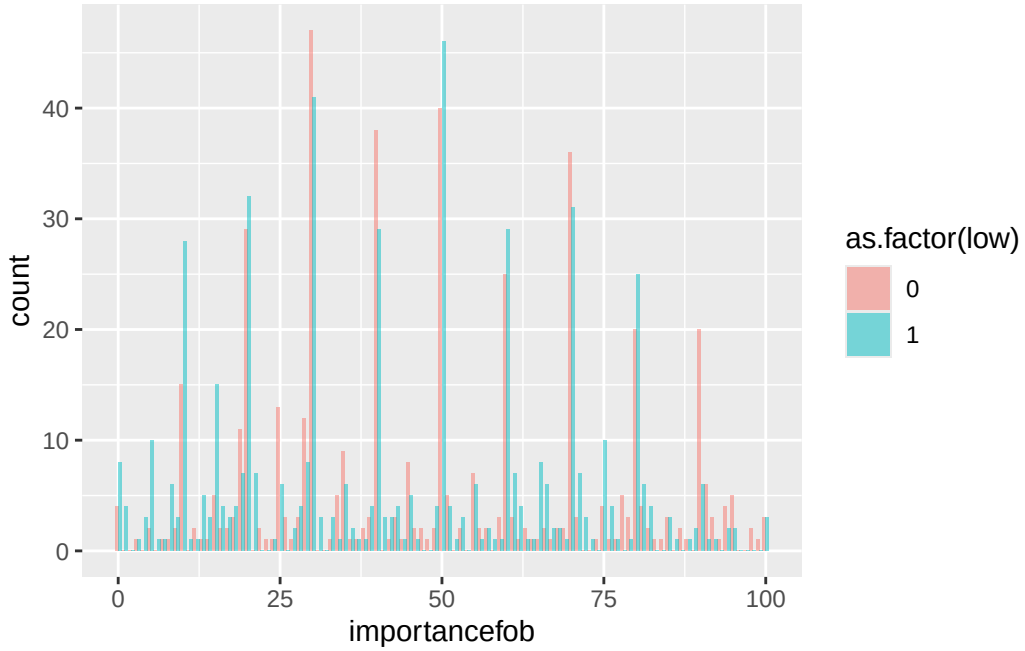


- First order importance

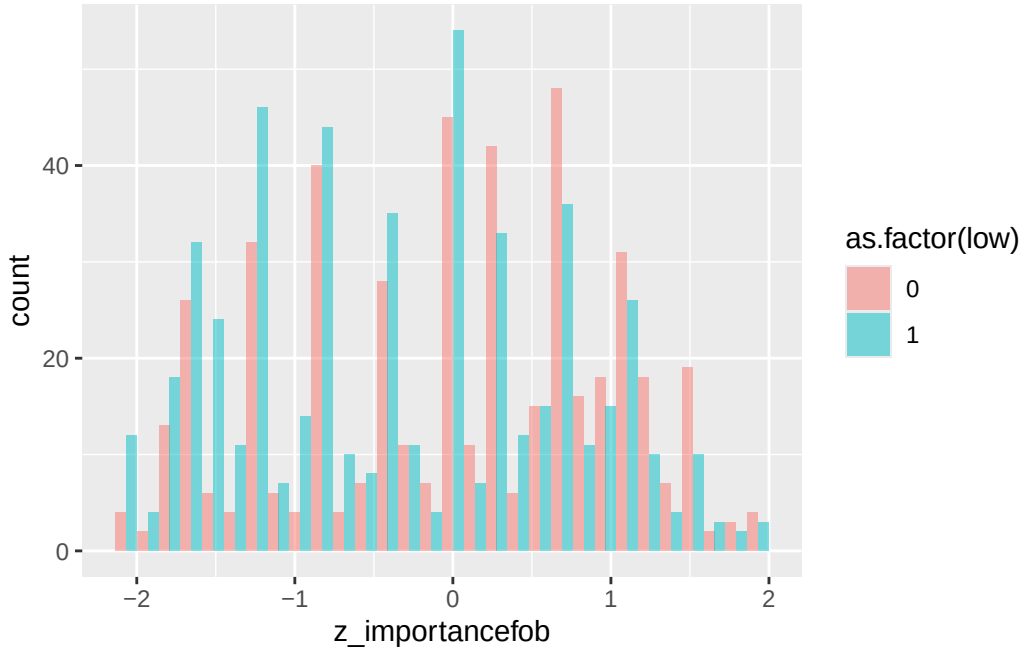
```
# Original
ggplot(df, aes(importancefob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Fliped
df2[, importancefob := ifelse(low == 1, importancefob, 100 - importancefob)]
ggplot(df2, aes(importancefob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```

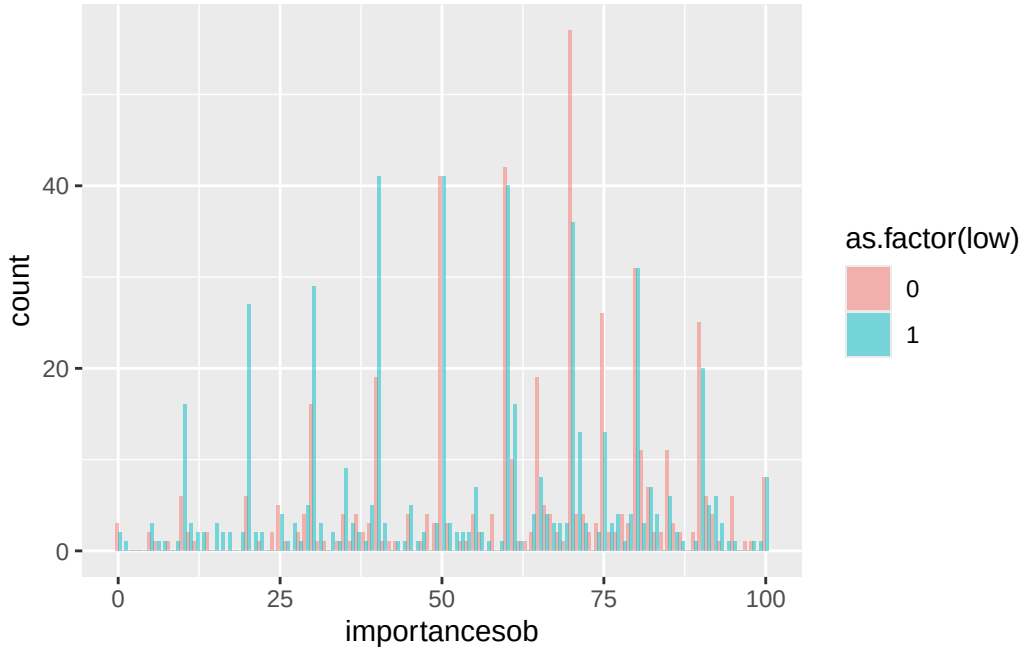



```
# Z-score
ggplot(df, aes(z_importancefob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge')
```

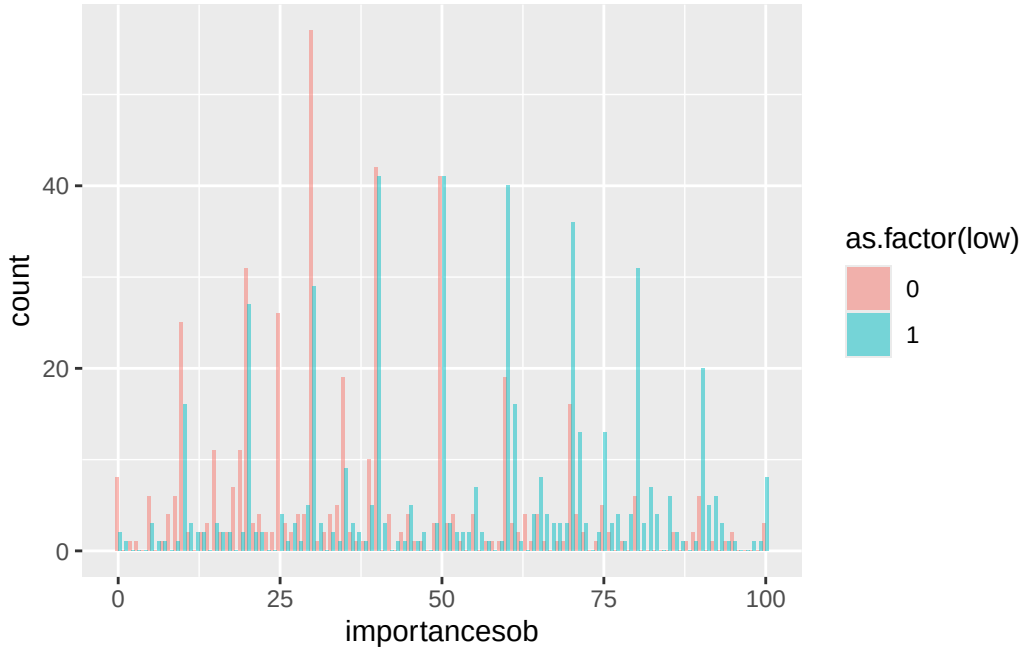


- Second order importance

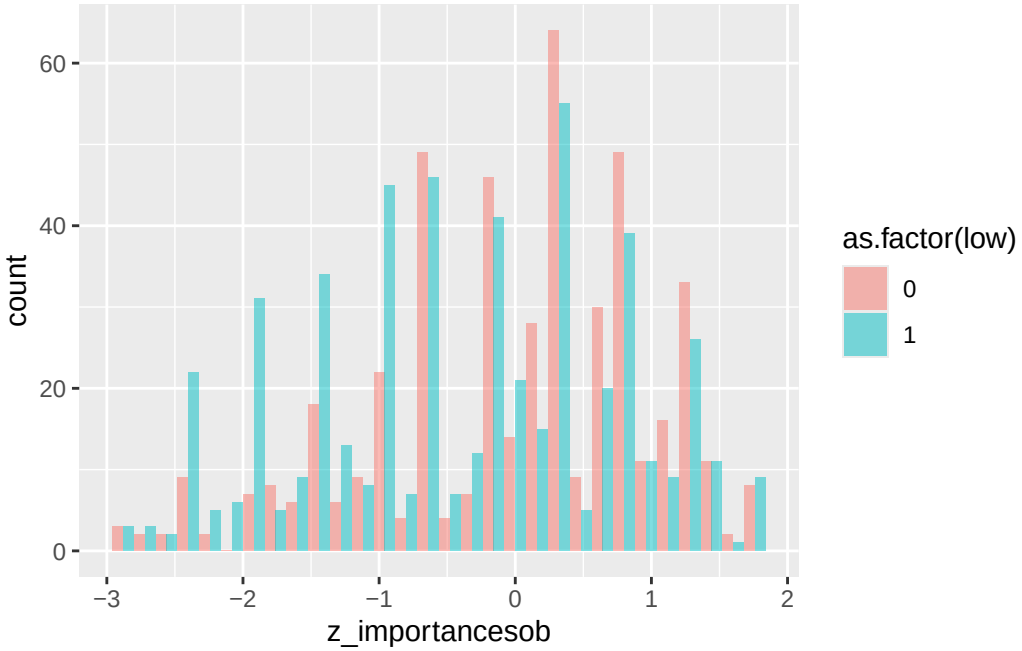
```
# Original
ggplot(df, aes(importancesob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Fliped
df2[, importancesob := ifelse(low == 1, importancesob, 100 - importancesob)]
ggplot(df2, aes(importancesob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge', binwidth = 1)
```




```
# Z-score
ggplot(df, aes(z_importancesob, fill = as.factor(low))) +
    geom_histogram(alpha = 0.5, position = 'dodge')
```



This is the replication of Table 3 with the original z-score values. The values match very closely. But note that these are measured in standard deviations. A 0.3 or 0.4 standard deviation change is a massive effect size.

```
# Column 2
col2 <- plm(z_qualityfob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col2_se <- sqrt(diag(vcovHC(col2, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
# Column 3
col3 <- plm(z_qualitysob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col3_se <- sqrt(diag(vcovHC(col3, type = "HC1", cluser = "id")))
# Column 4
col4 <- plm(z_importancefob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
```

```r
col4_se <- sqrt(diag(vcovHC(col4, type = "HC1", cluser = "id")))
# Column 5
col5 <- plm(z_importancesob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col5_se <- sqrt(diag(vcovHC(col5, type = "HC1", cluser = "id")))

# Means
df_control <- subset(df, df$low == 0) # Subset for control
col2_mean <- round(mean(df_control$z_qualityfob, na.rm = TRUE), 3)
col3_mean <- round(mean(df_control$z_qualitysob, na.rm = TRUE), 3)
col4_mean <- round(mean(df_control$z_importancefob, na.rm = TRUE), 3)
col5_mean <- round(mean(df_control$z_importancesob, na.rm = TRUE), 3)

# Present
stargazer(col2, col3, col4, col5,
        type = "text",
        keep = c(1),
        covariate.labels = c("Null result treatment"),
        se = list(col2_se, col3_se, col4_se, col5_se),
        keep.stat = c("n", "adj.rsq"),
        model.numbers = TRUE,
        digits = 3,
        add.lines = list(c("Mean Dep. Var.", col2_mean, col3_mean, col4_mean, col5_mean)
        ))
```

```
================================================================================
                                        Dependent variable:
                        ---------------------------------------------------------
                        z_qualityfob z_qualitysob z_importancefob z_importancesob
                            (1)          (2)            (3)             (4)
--------------------------------------------------------------------------------
Null result treatment   -0.353***    -0.446***      -0.353***       -0.436***
                         (0.066)      (0.062)        (0.057)         (0.057)

--------------------------------------------------------------------------------
Mean Dep. Var.               0            0              0               0
Observations               920          920          1,000           1,000
Adjusted R2               -0.268       -0.206         -0.259          -0.227
================================================================================
```

```
Note:                                                *p<0.1; **p<0.05; ***p<0.01
```

Again, I replicate Table 3 but now with the original percentage point distribution (same units as Column 1 for primary outcome of interest). Again, the effect sizes are very statistically significant and large. But relative to the control group's dependent variable means, these effects are only shifting towards 50/50 decisions on measures of importance or quality for the paper. This is not flipping the decision as we see for the measure of publishability.

```
# Column 2
col2 <- plm(qualityfob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col2_se <- sqrt(diag(vcovHC(col2, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
# Column 3
col3 <- plm(qualitysob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col3_se <- sqrt(diag(vcovHC(col3, type = "HC1", cluser = "id")))
# Column 4
col4 <- plm(importancefob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col4_se <- sqrt(diag(vcovHC(col4, type = "HC1", cluser = "id")))
# Column 5
col5 <- plm(importancesob ~ low + exlow + exhigh + field + phd + unilow + pval,
                data = df,
                index = c("id", "vignette"),
                model = "within")
col5_se <- sqrt(diag(vcovHC(col5, type = "HC1", cluser = "id")))

# Means
df_control <- subset(df, df$low == 0) # Subset for control
col2_mean <- round(mean(df_control$qualityfob, na.rm = TRUE), 3)
col3_mean <- round(mean(df_control$qualitysob, na.rm = TRUE), 3)
col4_mean <- round(mean(df_control$importancefob, na.rm = TRUE), 3)
col5_mean <- round(mean(df_control$importancesob, na.rm = TRUE), 3)

# Present
```

```r
stargazer(col2, col3, col4, col5,
        type = "text",
        keep = c(1),
        covariate.labels = c("Null result treatment"),
        se = list(col2_se, col3_se, col4_se, col5_se),
        keep.stat = c("n", "adj.rsq"),
        model.numbers = TRUE,
        digits = 3,
        add.lines = list(c("Mean Dep. Var.", col2_mean, col3_mean, col4_mean, col5_mean)
        ))
```

```
=================================================================
                                Dependent variable:
                    ---------------------------------------------
                    qualityfob qualitysob importancefob importancesob
                       (1)        (2)          (3)           (4)
-----------------------------------------------------------------
Null result treatment -7.072***  -8.595***    -8.815***     -9.390***
                       (1.318)    (1.200)      (1.421)       (1.228)

-----------------------------------------------------------------
Mean Dep. Var.          60.165     63.074       51.468        62.382
Observations             920        920         1,000         1,000
Adjusted R2             -0.268     -0.206       -0.259        -0.227
=================================================================
Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

Sample Composition

NOTE: Potentially worth discussing

- Ryan
- Two things: include the two sample selections they edit and…
- Remove observations that may not have salience of treatment (short and long duration or vigenette observations as well as 'finished == 0' observations)
- Re-estimate Table 3 effects

Salience of Treatment

NOTE: Not worth discussing. They have appendix table on this...

The treatment for the null result treatment and the cross-randomized vigenette characteristics are presented through paragraphs the reviewer reads. These are short paragraphs. But some respondents take very little time or a very long time to respond to each vigenette. Therefore, we can use the time duration for each vigenette as a measure of salience that the respondent is (1) paying attention and (2) absorbing the treatment. For example, we show what the vigenettes look like to the participants (from the online appendix).

First, I explore for outliers. I start by looking at the tails of the distribution. I note that there is a very long right tail in time. This suggests that some people open the survey, leave it in the background, and then come back to it. This is times at the vigenette level, not overall. So this ideally is not a measure of people not closing out of the survey. I examing times over 10,000 seconds (166 minutes). This is 5% of the sample. That is reasonable as 1 in 20 folks are getting distracted. But if I lower this to 2000 seconds (33 minutes), the proportion is 31% of the sample. This suggests that a large portion of the sample is take a very considerably long time to make a decision on the short paragraph above. This is not necessarily bad, it is just a bit suprising. On the other hand, I examine folks for whom they may not be examining the information closely. These are folks who read the paragraph perhaps too quickly, and by consequence are not recieving a salient treatment. About 20% of respondents are completing the vigenette section in under a minute. With seasoned eyes, perhaps that is reasonable. But it does suggest the respondents are just glossing over the information rather than reading carefully. If I restrict this to only 30 seconds, only 2.5% of the sample respondents are replying very quickly. What I find suspicious is how exact these values are for the lower measures.

If we explore the distribution (cutting off the long right tail at 2000 seconds – removing 30% of the sample), we can see the influence of treatment as a salient effect and order effects of the vigenettes as the respodent learns (gets quicker) or gets bored (gets slower). The average is 177 seconds (3 minutes) to read the vigenette and respond. The treatment groups have considerable overlap. And the order effects show there is some learning to become quicker over time.

```
# Number of Observations in each tail of the distribution
time <- 10000 # 166.67 minutes
num_above_threshold <- sum(df$pagetime > time) # n above threshold
total_observations <- nrow(df) # n
percent_above_threshold <- (num_above_threshold / total_observations) * 100 # percent
percent_above_threshold # 5%
```

```
[1] 0.05208333
```

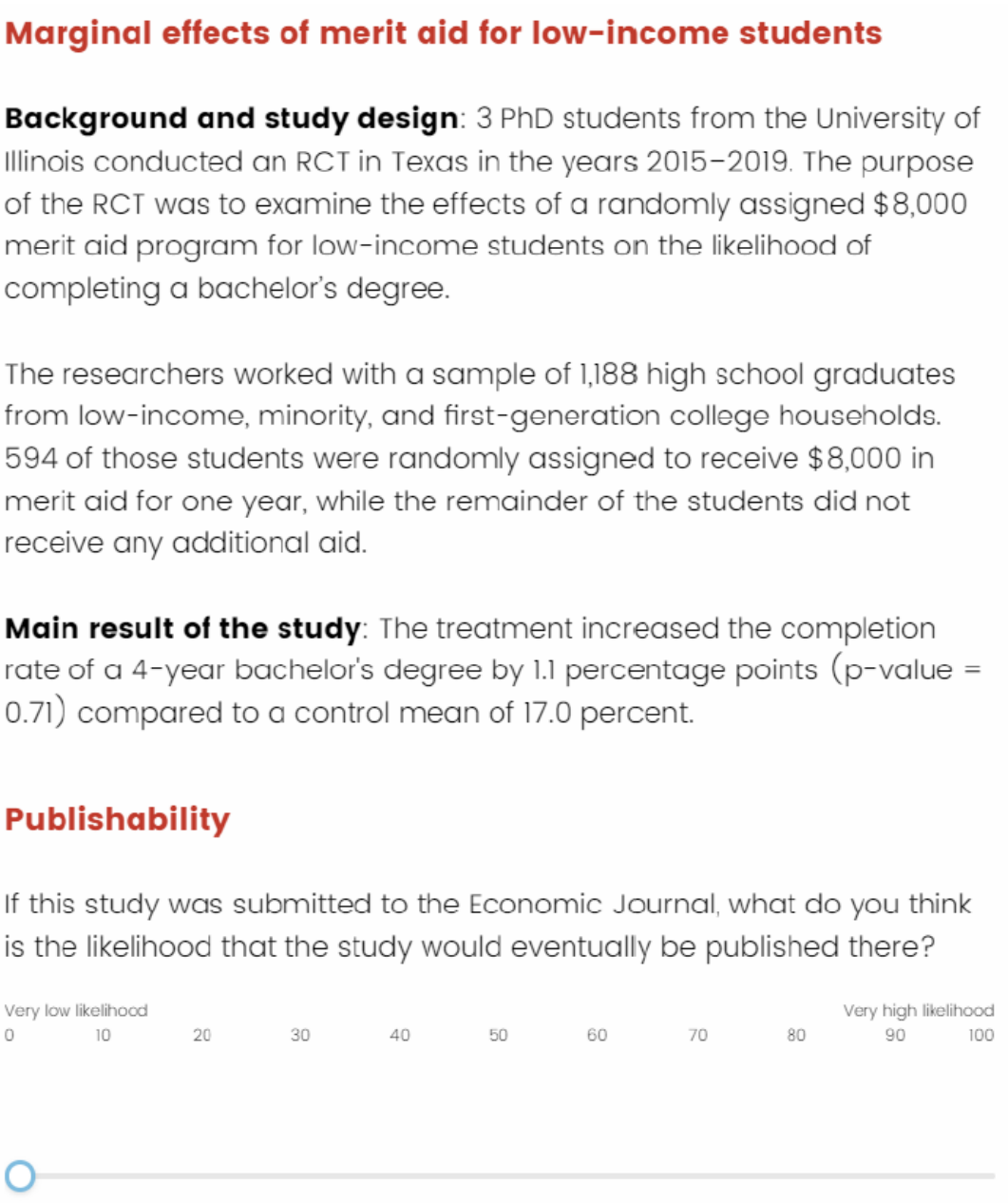


Figure 3: Vigenette Example

```
print(paste0("Precentage of Vigenette Times Above ", time, " Seconds: ", format(percent_ab
```

```
[1] "Precentage of Vigenette Times Above 10000 Seconds: 0.0521"
```

```
time <- 2000 # 33 minutes
num_above_threshold <- sum(df$pagetime > time) # n above threshold
total_observations <- nrow(df) # n
percent_above_threshold <- (num_above_threshold / total_observations) * 100 # percent
percent_above_threshold # 31%
```

```
[1] 0.3125
```

```
print(paste0("Precentage of Vigenette Times Above ", time, " Seconds: ", format(percent_ab
```

```
[1] "Precentage of Vigenette Times Above 2000 Seconds: 0.312"
```

```
time <- 60 # 1 minute
num_above_threshold <- sum(df$pagetime < time) # n above threshold
total_observations <- nrow(df) # n
percent_above_threshold <- (num_above_threshold / total_observations) * 100 # percent
percent_above_threshold # 20%
```

```
[1] 20
```

```
print(paste0("Precentage of Vigenette Times Below ", time, " Seconds: ", format(percent_ab
```

```
[1] "Precentage of Vigenette Times Below 60 Seconds: 20"
```

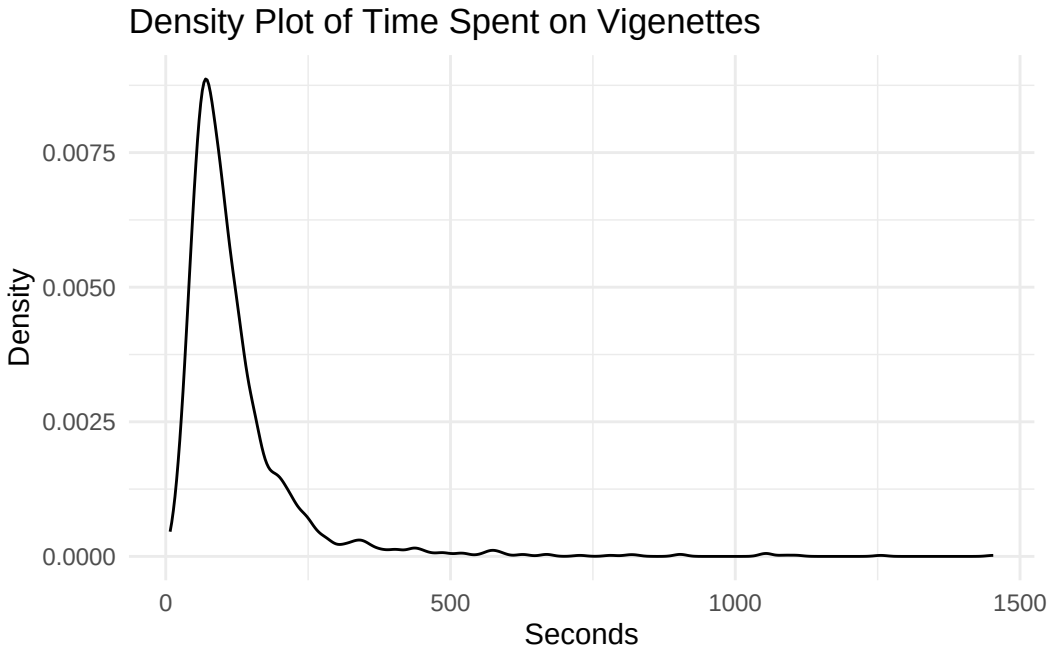
```
time <- 30 # 1/2 minute
num_above_threshold <- sum(df$pagetime < time) # n above threshold
total_observations <- nrow(df) # n
percent_above_threshold <- (num_above_threshold / total_observations) * 100 # percent
percent_above_threshold # 2.5%
```

```
[1] 2.5
```

```
print(paste0("Precentage of Vigenette Times Below ", time, " Seconds: ", format(percent_ab
```

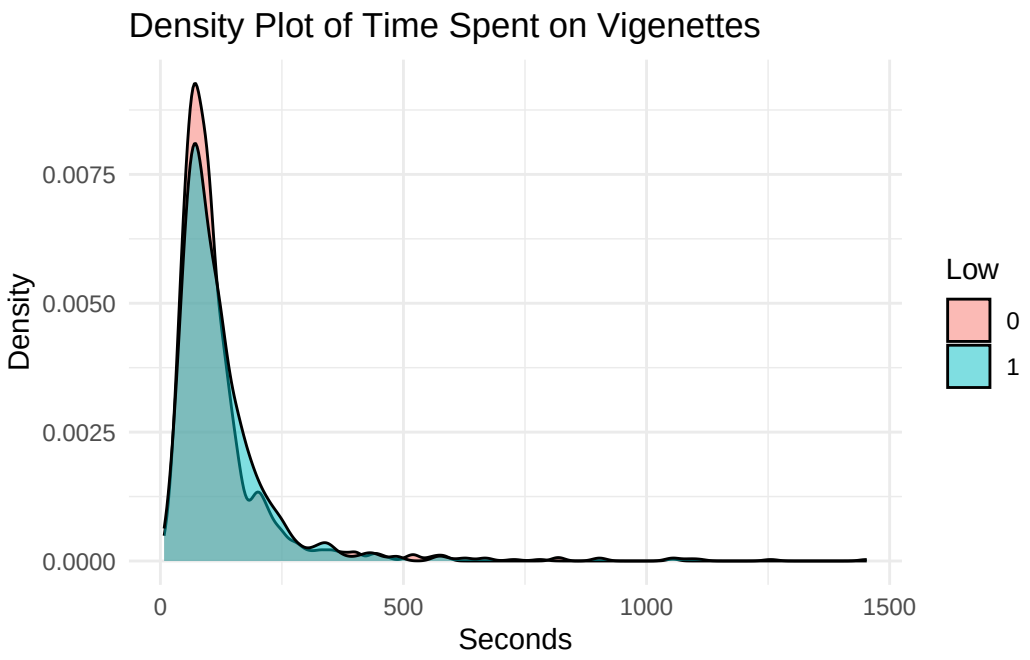
```
[1] "Precentage of Vigenette Times Below 30 Seconds: 2.5"
```

```
# Overall Distribution
ggplot(subset(df, df$pagetime < 2000), aes(x = pagetime)) +
    geom_density(alpha = 0.5) +
    labs(title = "Density Plot of Time Spent on Vigenettes",
         x = "Seconds",
         y = "Density",
         fill = "Low") +
    theme_minimal()
```

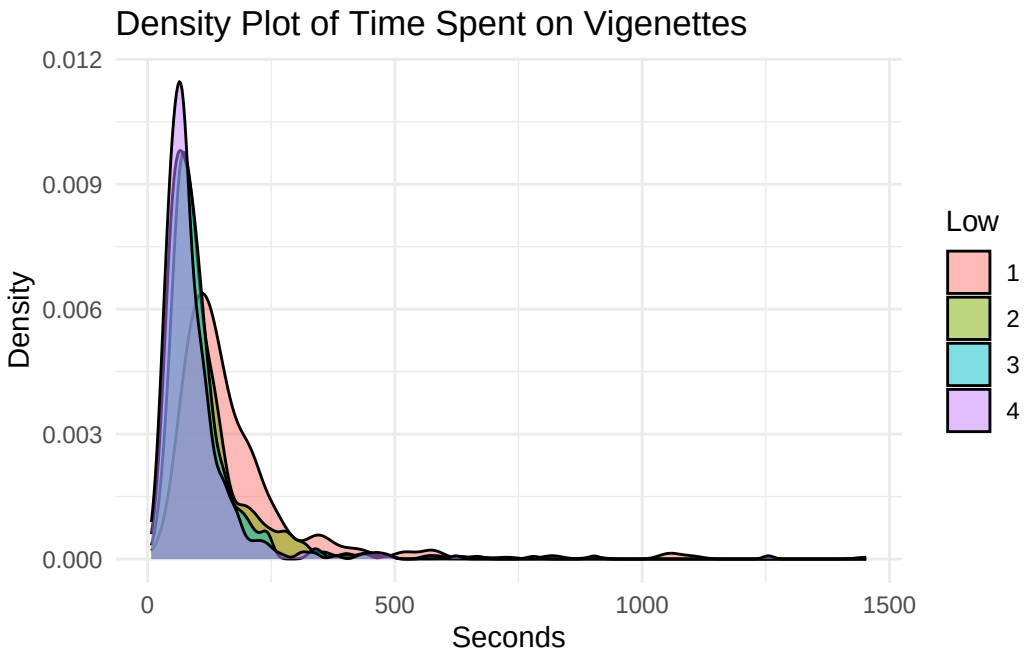



```
# By Treatment Status
ggplot(subset(df, df$pagetime < 2000), aes(x = pagetime, fill = factor(low))) +
    geom_density(alpha = 0.5) +
    labs(title = "Density Plot of Time Spent on Vigenettes",
         x = "Seconds",
         y = "Density",
         fill = "Low") +
```

```
    theme_minimal()
```




```
# Order Effects
ggplot(subset(df, df$pagetime < 2000), aes(x = pagetime, fill = factor(order))) +
    geom_density(alpha = 0.5) +
    labs(title = "Density Plot of Time Spent on Vigenettes",
         x = "Seconds",
         y = "Density",
         fill = "Low") +
    theme_minimal()
```



When I include an interaction effect of the vigenette order with the null effect treatment, the treatment effect for the null result becomes larger. This is in line with the learning effect we observe in the histograms. There is no effect of order or the interaction effect though, which is a bit suprising.

```
# Column 1
df = df[, low_order := low*order]
col1 <- plm(publish ~ low + order + low_order + exlow + exhigh + field + phd + unilow + pv
                data = df,
                index = c("id", "vignette"),
                model = "within")
col1_se <- sqrt(diag(vcovHC(col1, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
df_control <- subset(df, df$low == 0)
col1_mean <- round(mean(df_control$publish), 3) # Subset for control
# Present
stargazer(col1,
        type = "text",
        keep = c(1, 2, 3),
        covariate.labels = c("Null result treatment", "Vigenette Order", "Interaction"),
        se = list(col1_se),
        keep.stat = c("n", "adj.rsq"),
```

```
          model.numbers = TRUE,
          digits = 3,
          add.lines = list(c("Mean Dep. Var.", col1_mean)
          ))
```

```
================================================
                          Dependent variable:
                      ---------------------------
                                   publish
------------------------------------------------
Null result treatment          -15.235***
                                 (2.495)

Vigenette Order                  -0.223
                                 (0.582)

Interaction                       0.501
                                 (0.895)

------------------------------------------------
Mean Dep. Var.                   57.128
Observations                      1,920
Adjusted R2                      -0.071
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

Finally we consider how the result would change if we windsorize the tails of the sample. Specifically for the observations that we believe are recieving a salient treatment. These are those answering within less than 30 seconds and those answering after 10,000 seconds. This represents 2.5% and 5% of the sample, respectively. Additionally, I use a data driven winsorizing approach replacing outliers in the bottom and top 5% of the distribution with the most extreme retained values at the 5% and 95% quantiles of the original distribution for publishability. Accounting for these outliers does not change the magnitude of the effect size.

```
# Manual Trim
df3 = data.table::copy(df)
df3 = df3[pagetime > 30,]
df3 = df3[pagetime < 10000,]
# Re-estimate
```

```r
col1 <- plm(publish ~ low + + exlow + exhigh + field + phd + unilow + pval,
                data = df3,
                index = c("id", "vignette"),
                model = "within")
col1_se <- sqrt(diag(vcovHC(col1, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
df_control <- subset(df3, df3$low == 0)
col1_mean <- round(mean(df_control$publish), 3) # Subset for control

# Automatic winsorizing
library(DescTools)
df4 = data.table::copy(df)
df4 = df4[, publish := Winsorize(publish, val = quantile(publish, probs = c(0.05, 0.95), n
# Re-estimate
col2 <- plm(publish ~ low + + exlow + exhigh + field + phd + unilow + pval,
                data = df4,
                index = c("id", "vignette"),
                model = "within")
col2_se <- sqrt(diag(vcovHC(col1, type = "HC1", cluser = "id")))
    # TODO: Issue with adding clustering
df_control <- subset(df4, df4$low == 0)
col2_mean <- round(mean(df_control$publish), 3) # Subset for control

# Present
stargazer(col1, col2,
        type = "text",
        keep = c(1),
        covariate.labels = c("Null result treatment"),
        se = list(col1_se, col2_se),
        keep.stat = c("n", "adj.rsq"),
        model.numbers = TRUE,
        digits = 3,
        add.lines = list(c("Mean Dep. Var.", col1_mean, col2_mean)
        ))
```

```
================================================
                        Dependent variable:
                    ---------------------------
                              publish
                         (1)            (2)
------------------------------------------------
```

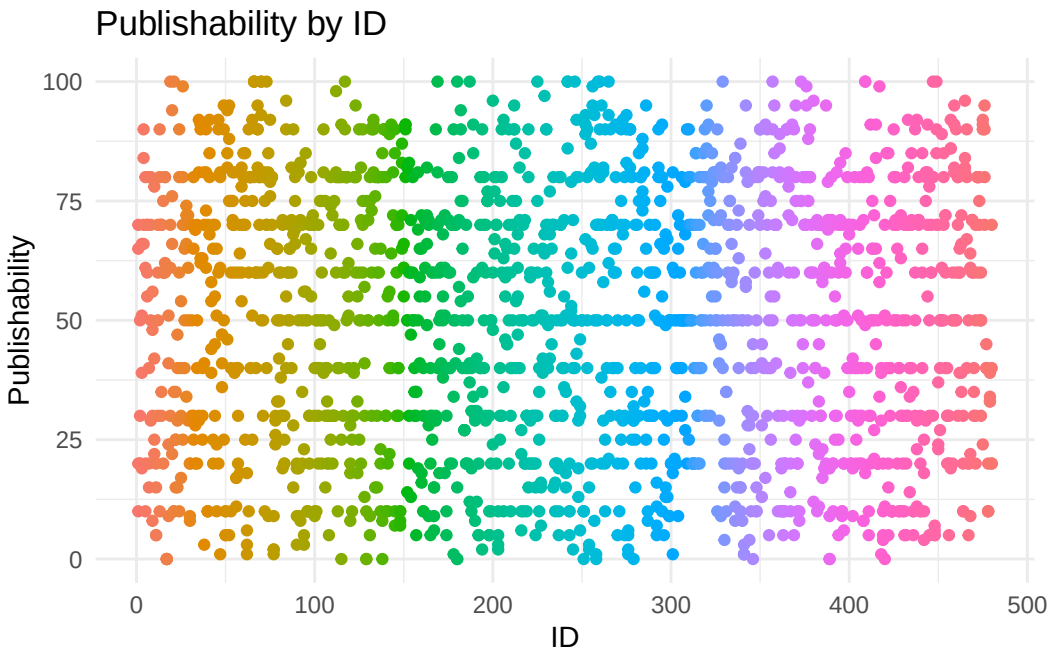
```
Null result treatment   -14.210***    -13.656***
                         (1.117)       (1.117)

-------------------------------------------------
Mean Dep. Var.            57.191        56.908
Observations              1,871          1,920
Adjusted R2               -0.073        -0.074
=================================================
Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Data Patterns

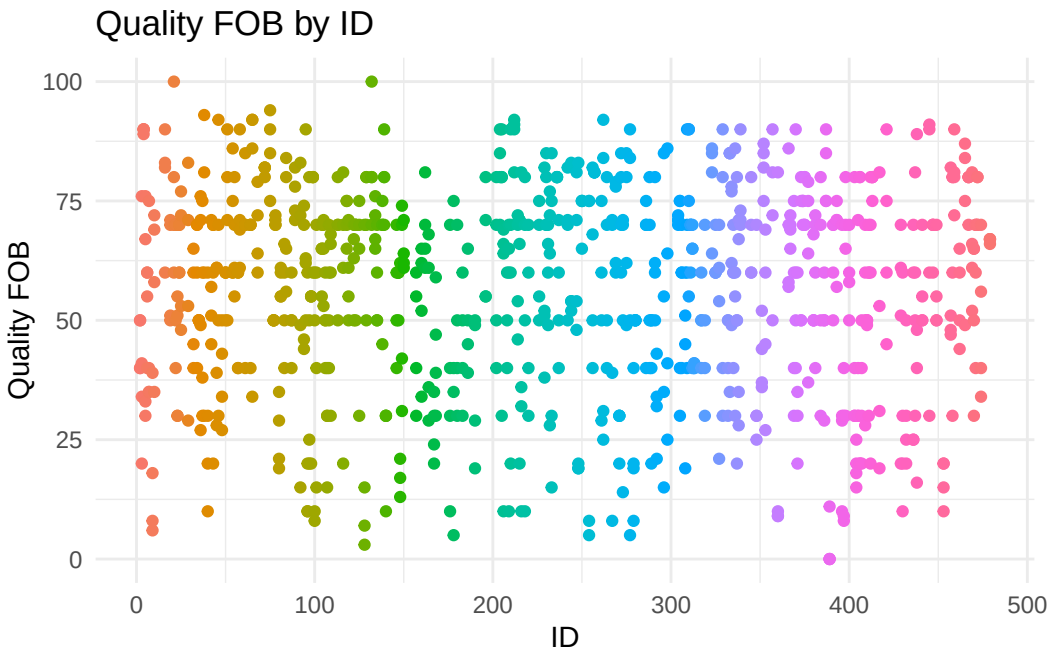
NOTE: Not worth publishing about

Plot frequency of observation values for outcomes by appearence in unsorted data. What we would be looking for a horizontal streaks. This means the the same value for the outcome measure (y axis) is being repeated. This could be evidence of fabricating the data. I am not seeing a lot of evidence that this is occurring.

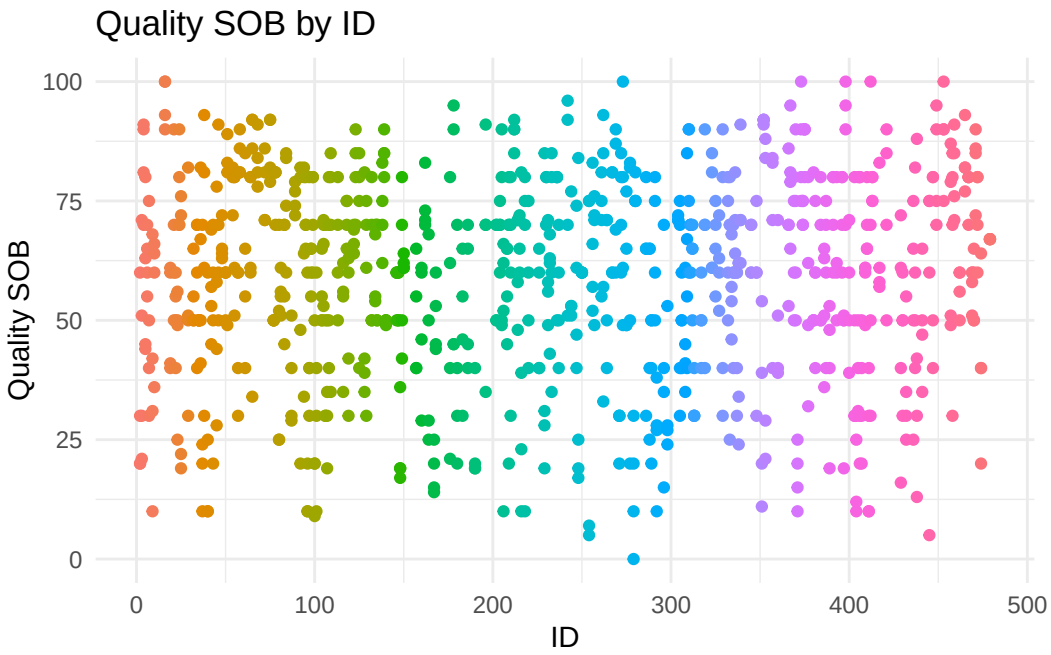
```r
# Load data
df <- setDT(import(path_data))

# Publish
ggplot(df, aes(x = id, y = publish, color = factor(id))) +
    geom_point() +
    labs(title = "Publishability by ID",
         x = "ID",
         y = "Publishability",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



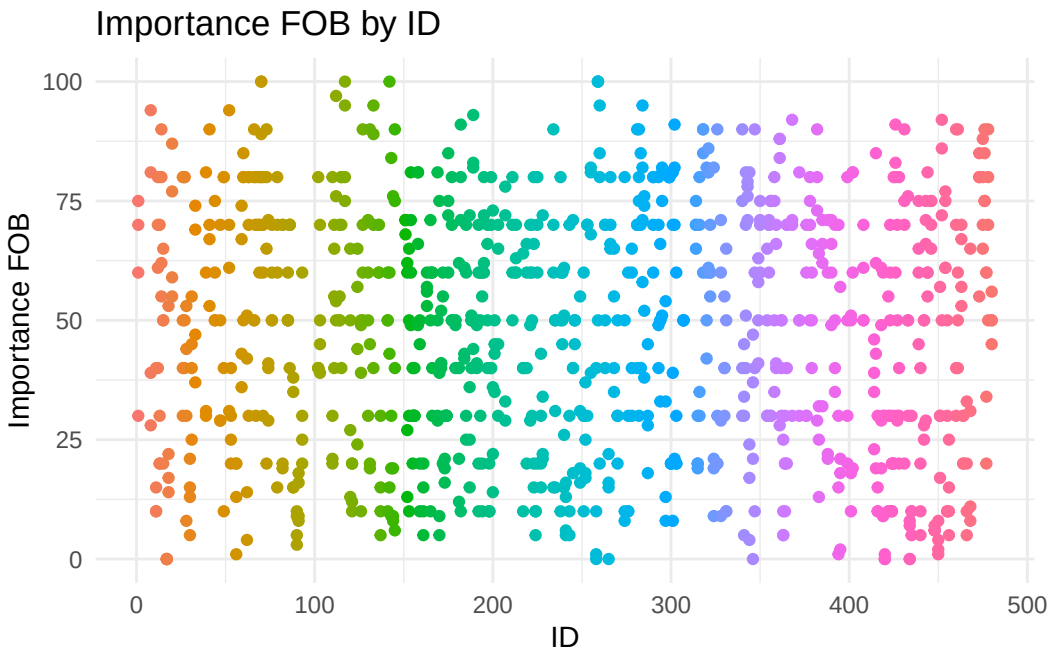

```
# Quality FOB
ggplot(df, aes(x = id, y = qualityfob, color = factor(id))) +
    geom_point() +
    labs(title = "Quality FOB by ID",
         x = "ID",
         y = "Quality FOB",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



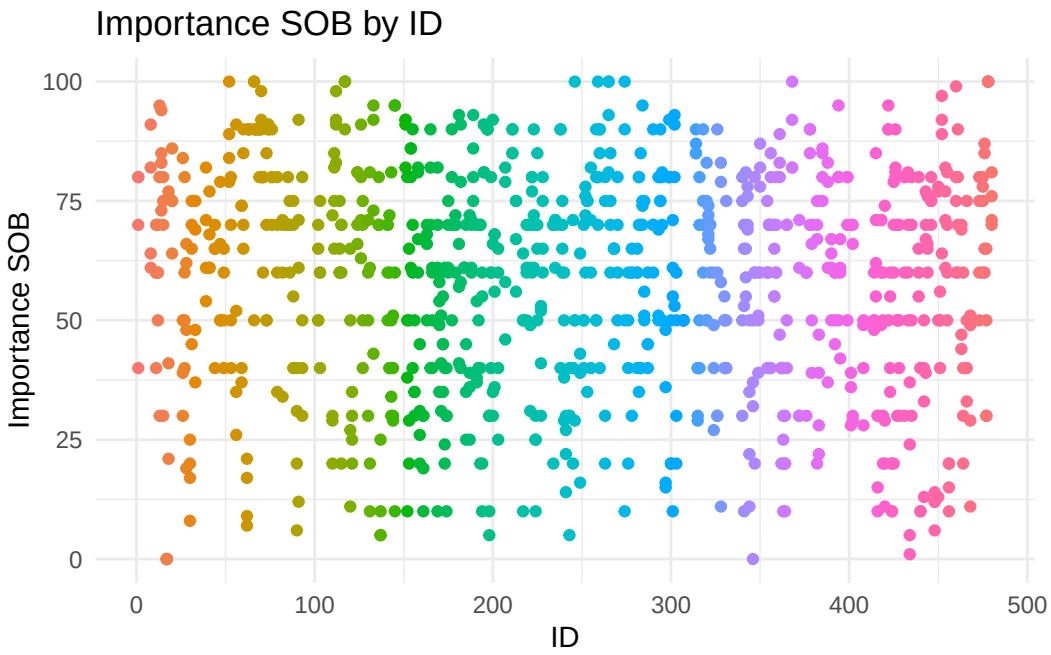

```
# Quality SOB
ggplot(df, aes(x = id, y = qualitysob, color = factor(id))) +
    geom_point() +
    labs(title = "Quality SOB by ID",
         x = "ID",
         y = "Quality SOB",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```




```r
# Importance FOB
ggplot(df, aes(x = id, y = importancefob, color = factor(id))) +
    geom_point() +
    labs(title = "Importance FOB by ID",
         x = "ID",
         y = "Importance FOB",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```

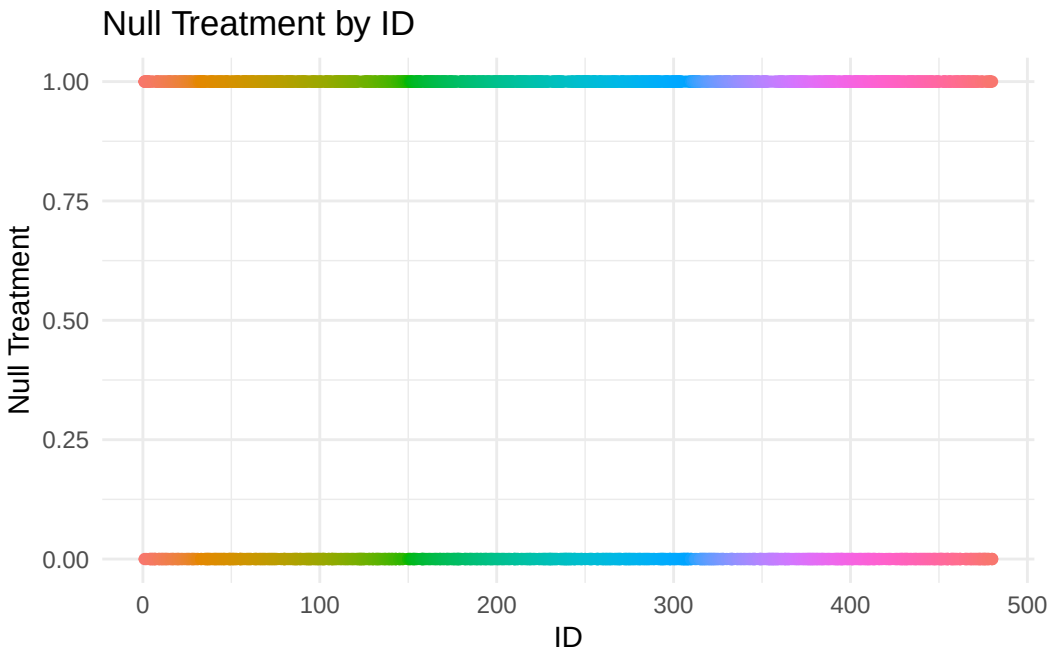



```
# Importance SOB
ggplot(df, aes(x = id, y = importancesob, color = factor(id))) +
    geom_point() +
    labs(title = "Importance SOB by ID",
         x = "ID",
         y = "Importance SOB",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```

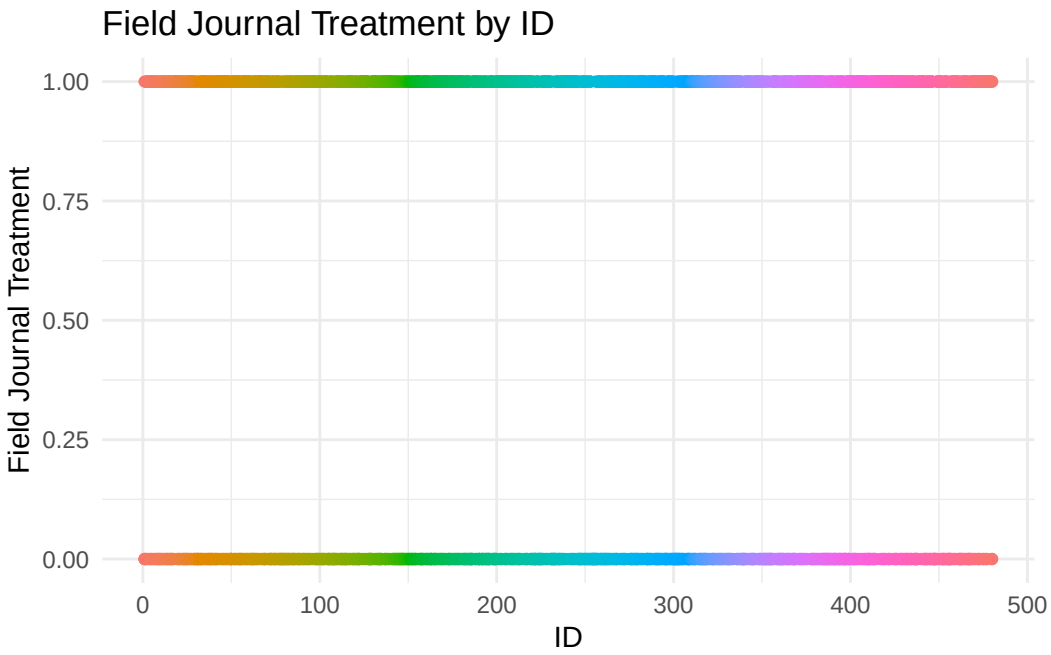


I also check to make sure that random assign to the many treatment groups appears randomly ordered in the data. If there are gaps in the lines between categories (e.g., either you are zero or one), then this is evidence of a streak of treatment assignments in the data. While streaks may happen randomly, we would not expect several long streaks in the data. From what I plot below, we do not observe that.

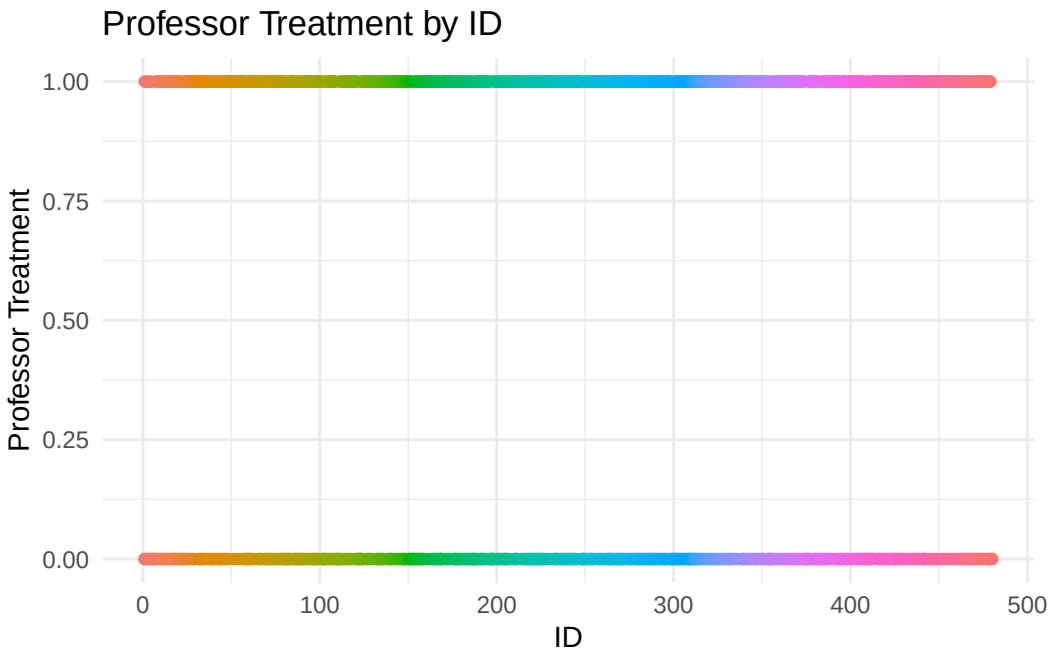
```
# Null Treatment
ggplot(df, aes(x = id, y = low, color = factor(id))) +
    geom_point() +
    labs(title = "Null Treatment by ID",
         x = "ID",
         y = "Null Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



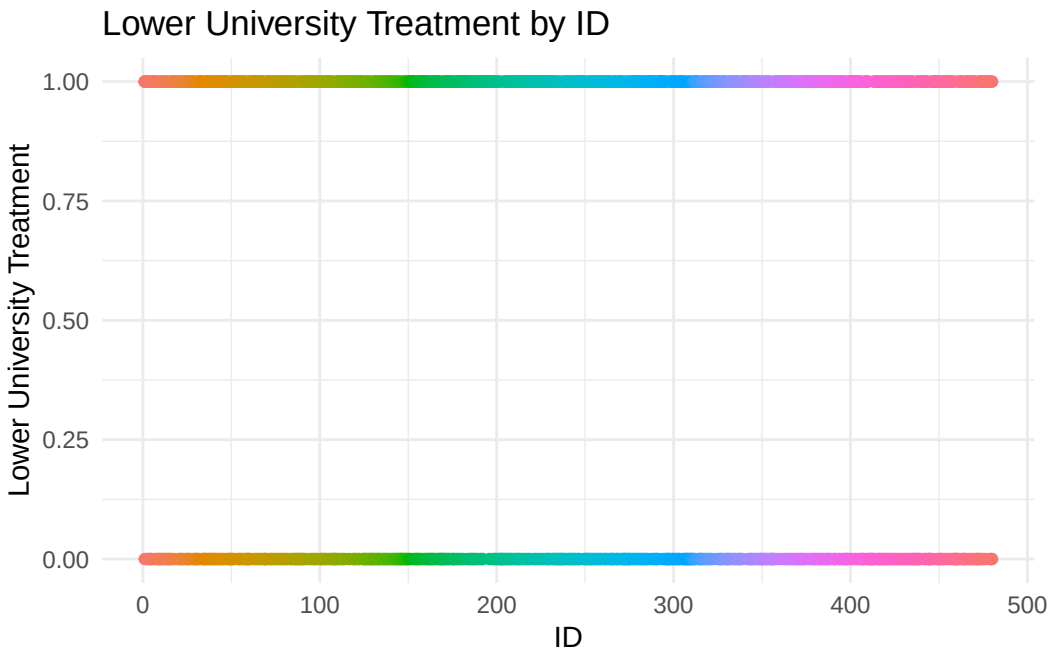

```r
# Field Journal Treatment
ggplot(df, aes(x = id, y = field, color = factor(id))) +
    geom_point() +
    labs(title = "Field Journal Treatment by ID",
         x = "ID",
         y = "Field Journal Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



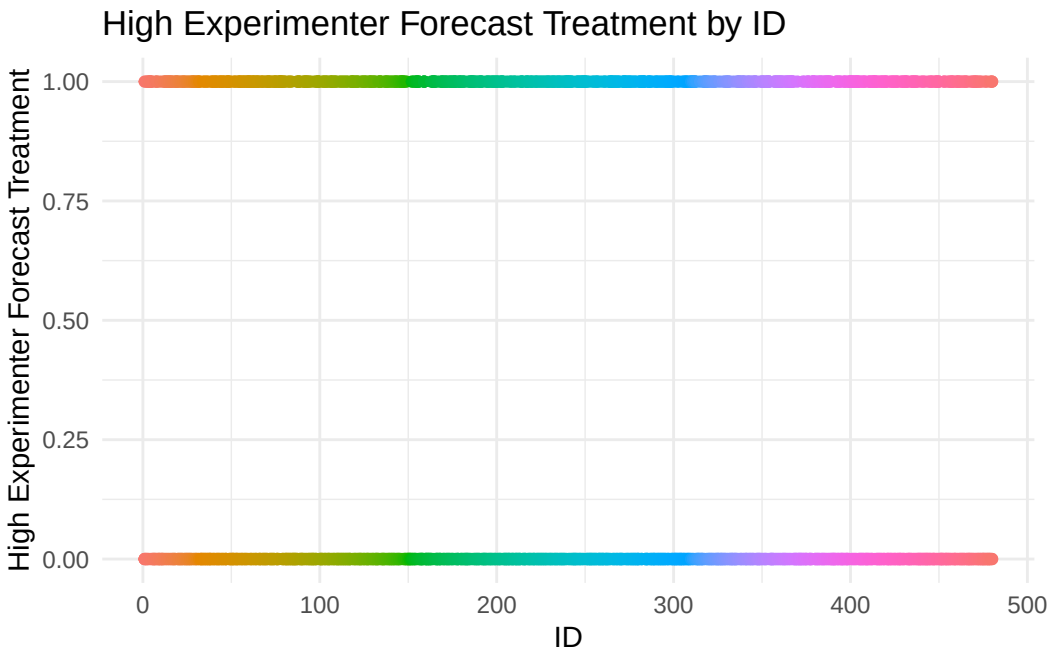

```
# Professor Treatment
ggplot(df, aes(x = id, y = professor, color = factor(id))) +
    geom_point() +
    labs(title = "Professor Treatment by ID",
         x = "ID",
         y = "Professor Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



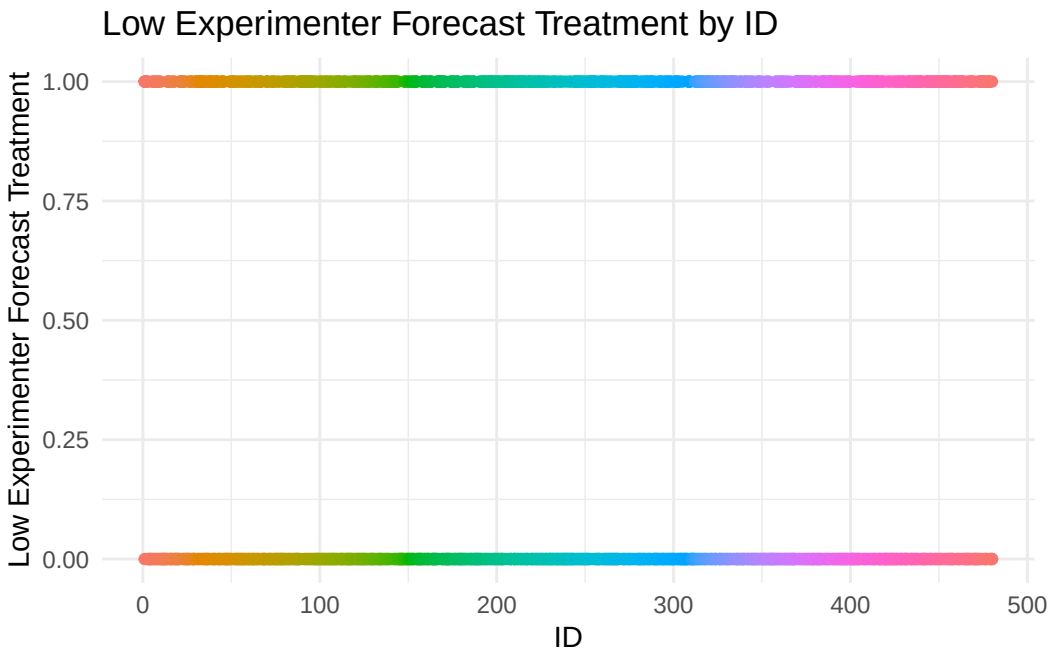

```r
# Lower University Treatment
ggplot(df, aes(x = id, y = unilow, color = factor(id))) +
    geom_point() +
    labs(title = "Lower University Treatment by ID",
         x = "ID",
         y = "Lower University Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```




```r
# High Experimenter Forecast Treatment
ggplot(df, aes(x = id, y = exhigh, color = factor(id))) +
    geom_point() +
    labs(title = "High Experimenter Forecast Treatment by ID",
         x = "ID",
         y = "High Experimenter Forecast Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```




```
# Low Experimenter Forecast Treatment
ggplot(df, aes(x = id, y = exlow, color = factor(id))) +
    geom_point() +
    labs(title = "Low Experimenter Forecast Treatment by ID",
         x = "ID",
         y = "Low Experimenter Forecast Treatment",
         color = "ID") +
    theme_minimal() +
    theme(legend.position = "none")
```



Quantile Regressions

NOTE: Not worth discussing

Quantile regressions to see variation across deciles. We should expect treatment effects to vary. If not, this may be a sign of fabrication. The quantile regressions below suggest that this is not the case.

```
# Quantile Regression:
    # Tau is quantile: Repeat for 0.1 to 0.9.
taus <- seq(from = .1, to = .9, by = 0.1) # Range of quantiles
quant_all  <- rq(publish ~ low + exlow + exhigh + field + phd + unilow + pval + id + vigne
                 tau = taus,
                 data = df)

print(quant_all$coef)
```

```
                tau= 0.1     tau= 0.2      tau= 0.3    tau= 0.4      tau= 0.5
(Intercept)  21.348115299  34.00591716  41.585728444  55.4291725  65.244009450
low         -10.757206208 -16.02366864 -17.734390486 -21.0434783 -20.944313196
exlow        -1.043237251   0.43195266   2.537165510   1.3941094  -1.055686804
```

```
exhigh        1.042128603   0.58579882   2.384539148   1.5315568  -2.173135336
field         8.573170732  11.31952663  15.092170466  16.9523142  15.204184948
phd          -3.177383592  -3.53846154  -4.019821606  -4.5820477  -4.132635842
unilow       -2.618625277  -4.12426036  -4.036669970  -5.7854137  -6.296658792
pval         -2.311529933  -4.44378698  -7.978196234  -9.0995792  -8.391495106
id           -0.005543237  -0.00591716  -0.004955401  -0.0112202  -0.009449882
vignette      0.378048780   0.29585799   0.173439049   0.1346424   0.209247384
                tau= 0.6      tau= 0.7      tau= 0.8     tau= 0.9
(Intercept)  71.37862233  74.260091469  83.091891892 91.528519247
low         -18.85216152 -13.543646848 -11.137297297 -8.721270020
exlow        -1.20978622  -0.716245775  -3.062702703 -2.286035403
exhigh       -1.18546318  -1.584410420  -3.713513514 -2.406012925
field        14.64351544  12.794591370  11.102702703  8.451531329
phd          -5.21263658  -2.650825214  -3.993513514 -5.009834223
unilow       -5.06156770  -2.730562736  -4.016216216 -2.075864007
pval         -6.25472684  -5.585603500  -4.873513514 -3.522337735
id           -0.01168646  -0.004573474  -0.004324324 -0.003371734
vignette      0.20931116  -0.062239014   0.445405405 -0.324810340
```

```r
print(quant_all$coef[2,])
```

```
 tau= 0.1  tau= 0.2  tau= 0.3  tau= 0.4  tau= 0.5  tau= 0.6  tau= 0.7  tau= 0.8
-10.75721 -16.02367 -17.73439 -21.04348 -20.94431 -18.85216 -13.54365 -11.13730
 tau= 0.9
 -8.72127
```

```r
# NOTE: So there is variation over the quantiles... good sign

# q01_se <- sqrt(diag(vcovHC(q_05, type = "HC1", cluser = "id")))
#     # TODO: Issue with quantile regression with FE
#     # TODO: Do for other values

# # Present
# stargazer(col1,
#         type = "text",
#         keep = c(1),
#         covariate.labels = c("Null result treatment"),
#         se = list(col1_se),
#         keep.stat = c("n", "adj.rsq"),
#         model.numbers = TRUE,
#         digits = 3,
```

```
#         add.lines = list(c("Mean Dep. Var.", col1_mean)
#         ))
# # Summarize the results
# summary(quantile_reg)
```

```
# TODO: Need to recover the SE to create CI for the plots.


# Plot the quantile regressions
# plot_models(ols, quant_reg_med, quant_reg_first, quant_reg_last,
#             show.values = TRUE,
#             m.labels = c("OLS", "Median", "10th percentile",
#                          "95th percentile",
#                          legend.title = "Model")
#             )
```

(Unlikely to Complete) Propensity Score Matching

NOTE: Not worth discussing

- Ryan/Derek…
- Create Propensity scores with logit
- Do matching
- Estimated effect on matched pairs

(Minor) Median is Static

NOTE: Not worth discussing

- No matter the sample difference in each randomized group, it is almost always 50.