Appendix: Project Implementation Code

March 4, 2019

Contents

Dataset: Reading in and Tidying Data **1**
- Load libraries and data file . . . 1
- Subset with columns of interest; correct data types . . . 2
- Feature Engineering . . . 3

Measures: Understanding X, Y and Possible Confounders Present in Data **4**
- Inspection of variables . . . 4

Summary Stats **6**
- Our X & Y of interest: Duration predicting success . . . 6
- Evaluating Potential Confounders: Time, Category, Country, and Goal . . . 9

Propensity Score Matching **15**
- Balance in terms of propensity score, pre matching . . . 15
- Matching process . . . 17
- Balance in terms of propensity score, pre matching . . . 18

Fixed Effects Regression **20**
- Success on Treatment . . . 21
- Success on Campaign Duration . . . 21
- Pledged to Goal Ratio on Treatment . . . 22
- Pledged to Goal Ratio on Campaign Duration . . . 23
- Final FE Regression Output and Results . . . 25

Dataset: Reading in and Tidying Data

Load libraries and data file

```
library(dplyr)
library(magrittr)
library(ggplot2)
library(lubridate)
library(Hmisc)
library(MatchIt)
library(lfe)
library(stargazer)
library(MESS)
library(data.table)
library(tableone)

# Desktop directory
file_dir <- 'C:/Users/Donny/Google Drive/MSBA/Spring 2019/MSBA 6440/MSBA 6440 Data/'
# Laptop directory
file_dir <- 'G:/My Drive/MSBA/Spring 2019/MSBA 6440/MSBA 6440 Data/'
```

```r
# load data frame
ks_df <- read.csv(paste(file_dir, 'ks-projects-201801.csv', sep = ''))
```

Subset with columns of interest; correct data types

Here, we select the variables of interest and correct their data types.

- `ID`: A unique identifier for each campaign. Will be useful in later cleaning steps if we want to look at specific records
- `name`: Another unique identifier
- `category`: The subcategory within a larger main category that the campaign belongs to. There are 159 distinct categories here; we have enough data to not worry about testing power, but we will likely use the `main_category` for interpetation's sake.
- `main_category`: There are 15 main categories; we will use this for our fixed effects regression, as there are several observations within each level of category.
- `deadline` and `launched`: we will use these to calculate the duration of each campaign. Note that `launched` is provided with a timestamp as well; we do not know the timestamp for the `deadline`, so we are assuming it ends at midnight on the `deadline` date. We put `launched` in terms of date because, if our assumption about `deadline` is true, Kickstarter will treat a campaign that starts at 11:55 PM on March 1st and ends on March 2nd (at midnight) as a 1 day campaign.
- `goal` and `pledged`: The `goal` is the amount the kickstarter campaign manager hopes to raise, and `pledged` is the amount ultimately raised during the duration of the campaign. We will use the ratio of `pledged` amount to `goal` amount as an alternative success measure, where we are looking for what might cause an increase in this ratio. We also will use `goal` in our propensity matching, as we want balance in the treatment and control groups in terms of goal amount. For example, we don't want a lot of $1 campaigns that tend to have a higher success rate, regardless of campaign duration, biasing our results.
- `state`: this is a categorical label indicating the current success/failure/other status of the given campaign. We know what `success` and `failure` mean, and we need to determine if we can be sure about what the other labels here mean. We will explore this in the **Inspection of variables** section.
- `country`: The country of origin for the campaign. We also want to match away any potential bias here. Some countries may have an imbalance in terms of campaign duration, so we want to ensure that we have balanced groups within each country group.

```r
ks_df %<>%
  # select columns of interest
  select(c(
    'ID', 'name', 'category', 'main_category',
    'deadline', 'goal', 'launched',
    'pledged', 'state', 'country'
  )) %>%
  # correct the data types inferred from read.csv
  mutate(
    # campaign ID
    ID = as.factor(ID),
    # campaign name
    name = as.character(name),
    # sub category name
    category = as.factor(category),
    # main category name
    main_category = as.factor(main_category),
```

```
    # campaign deadline
    deadline = ymd(deadline),
    # campaign launch date
    launched = as.Date(ymd_hms(launched)),
    # campaign goal amount
    goal = as.numeric(goal),
    # campaign final pledged amount
    pledged = as.numeric(pledged),
    # campaign result / current status
    state = as.factor(state),
    # campaign country of origin
    country = as.factor(country)
  )
```

Feature Engineering

- `duration.days`: this is our continuous treatment, which we will later use to derive the binary treatment effect. as mentioned above, we will calculate duration days as we assume Kickstarter would define it. We also convert the data type to the common numeric type
- `pledged.goal.ratio`: our alternative target variable, where a ratio greater than `1` should imply success and less than `1` implies failure. We assume that there are campaigns that try to raise more than their goal, considering how many campaigns raised $1,000's of dollars with a goal of $1 or $10 dollars.
- `launch.year` and `launch.month`: these are our time derived groups. each year and each month is extracted from the date the campaign was `launched`. We will use these as time based fixed effects in our model, so static differences across periods of time do not confound our results
- `treat`: our binary treatment variable. It is derived from the continuous `duration.days` variable, where campaigns with `duration.days` <= 30 days is considered treated.
- `success`: our binary target variable. Simply a transformation of the `state` variable, where we keep it numeric instead of categorical. This allows us to build a linear probability model.

```
ks_df %<>%
  mutate(
    # length of campaign in days
    duration.days = as.numeric(deadline - launched),
    # the percent pledged to goal; success is equal or greater than 1
    pledged.goal.ratio = (pledged / goal),
    # year of launch, as a group level indicator
    launch.year = as.factor(year(launched)),
    # month of launch, as a group level indicator
    launch.month = as.factor(month(launched)),
    # binary treatment, in terms of campaign length
    treat = as.factor(ifelse(duration.days <= 30, 1, 0)),
    # successful campaigns are a 1, everything else is a 0
    success = as.numeric(ifelse(state == 'successful', 1, 0))
  )

# check for missing values; should be 0
sapply(ks_df, function(x) sum(is.na(x)))
```

```
##                  ID                name           category
##                   0                   0                  0
##       main_category            deadline                goal
```

```
##                 0                 0                 0
##          launched           pledged             state
##                 0                 0                 0
##           country     duration.days pledged.goal.ratio
##                 0                 0                 0
##       launch.year      launch.month             treat
##                 0                 0                 0
##           success
##                 0
```

Measures: Understanding X, Y and Possible Confounders Present in Data

Inspection of variables

We have 378,661 records, so we will have a high powered regression model if we don't prune too many records. Our econometric design should accomodate smaller record sizes for future use, so we will use `main_category` (15 levels) instead of `category` (159 levels) in our final model.

We also see an imbalance between the number of records in the test and control groups. We will address this with propensity matching in terms of `goal`, which we cannot address with fixed effects, and `country`, which has some small groups that may be matched away and could reduce our fixed effects regression dimensionality.

We also need to address the labels within `state` that aren't `successful` or `failed`. We find that `failed` make up the majority of non-`successful` records, and we don't see a clear enough pattern in `canceled` to try and infer if a given `canceled` record is a success or failure. We will drop any record that isn't a `successful` or `failed` campaign, and we will remove the 6 `failed` records with `pledged.goal.ratio` values over 1 and `successful` records with `pledged.goal.ratio` values under 1 (neither makes sense intuitively)

```
# 378661 records
length(unique(ks_df$ID))
```

```
## [1] 378661
```

```
# main categories has 15 levels
# we won't sacrifice nearly as much power if we model with main_category
length(unique(ks_df$category))
```

```
## [1] 159
```

```
length(unique(ks_df$main_category))
```

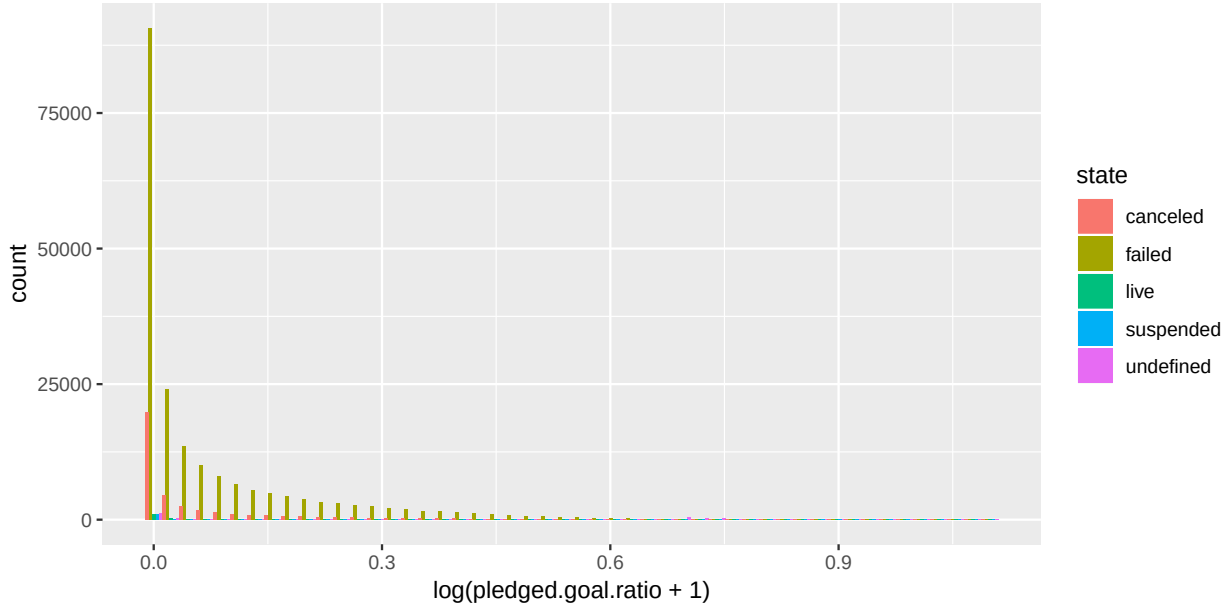
```
## [1] 15
```

```
# Do we only care about success vs failure? what about cancelations?
describe(ks_df$state)
```

```
## ks_df$state
##        n  missing distinct
##   378661        0        6
##
## Value        canceled     failed       live successful  suspended
## Frequency       38779     197719       2799     133956       1846
## Proportion      0.102      0.522      0.007      0.354      0.005
##
## Value       undefined
## Frequency        3562
```

```
## Proportion      0.009
```

```
# check each level in terms of pledged/goal ratio
# lots of skew, extremely high values in very skinny right tail
# look at records with ratio less than 2
ks_df %>%
  filter(
    (pledged.goal.ratio <= 2) &
      !(state == 'successful')
  ) %>%
  ggplot(aes(x = log(pledged.goal.ratio + 1), fill = state)) +
  geom_histogram(bins = 50, position = 'dodge')
```




```
# most of the records are failed
# canceled follows a similar trend

# Validate 'canceled' and 'failed records
# failed records
ks_df %>%
  filter(
   (state == 'successful') &
     (pledged.goal.ratio < 1)
  ) %>%
  nrow()
```

```
## [1] 5
```

```
# Validate 'canceled' and 'failed records
# failed records
ks_df %>%
  filter(
   (state == 'failed') &
     (pledged.goal.ratio >= 1)
  ) %>%
  nrow()
```

```
## [1] 6
```

```
# 6 total records with ratios above 1
# need to remove these

# cancelled records
ks_df %>%
  filter(
   (state == 'canceled') &
     (pledged.goal.ratio >= 1)
  ) %>%
  nrow()
```

```
## [1] 698
```

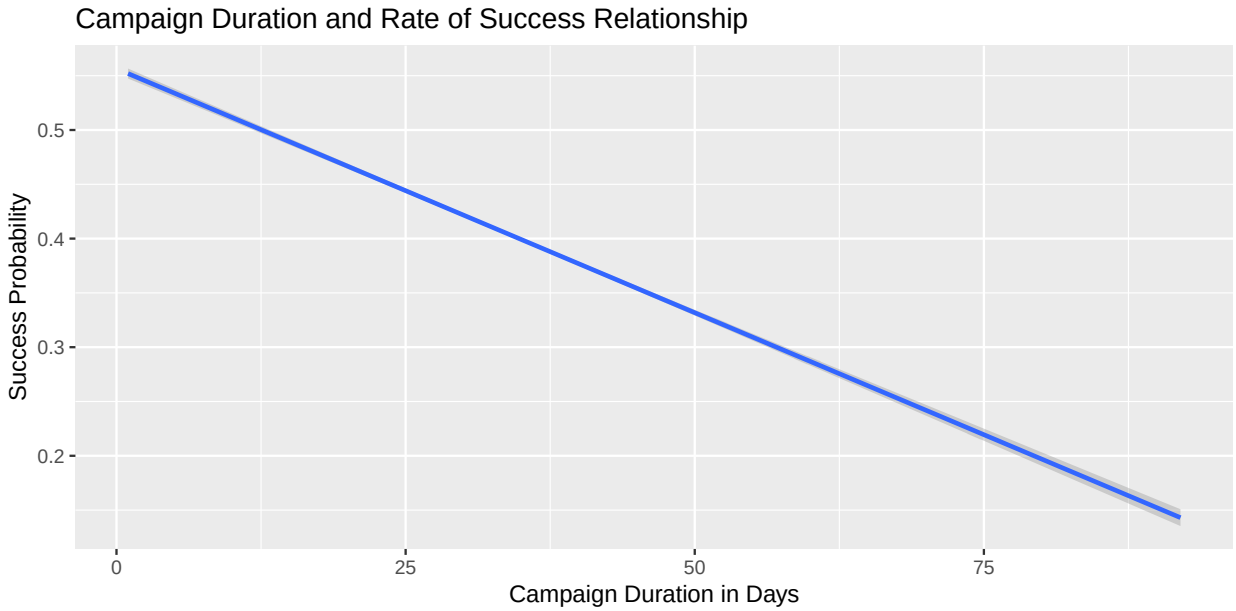
```
# 698 records
# canceled does not seem to imply failed
# nor can we infer it from context

# drop anything where state is not successful or failed
# and drop failed racords where pledged exceed goal
ks_df %<>%
  filter(
    (state %in% c('successful', 'failed'))
    & !((state == 'failed') & (pledged.goal.ratio >= 1))
    & !((state == 'successful') & (pledged.goal.ratio < 1))
  )
```

Summary Stats

Our X & Y of interest: Duration predicting success

We evaluated our X and Y relationship, where X is the treatment (duration in days <= 30) and Y is the success of the project. We observe a negative linear relationship between success and duration in days, and we note a mild correlation of `-0.11` with a p value near `0`.

```
# Duration days as a continuous treatment variable X
ks_df %>%
  ggplot(aes(y = success, x = duration.days)) +
  geom_smooth(method = 'lm') +
  labs(
    colour = '',
    x = 'Campaign Duration in Days',
    y = 'Success Probability',
    title = 'Campaign Duration and Rate of Success Relationship'
  )
```




```
# we see strong evidence of a linear relationship
# Correlation coefficient of -0.12 with strong p values
summary(lm(success ~ duration.days, data = ks_df))
```

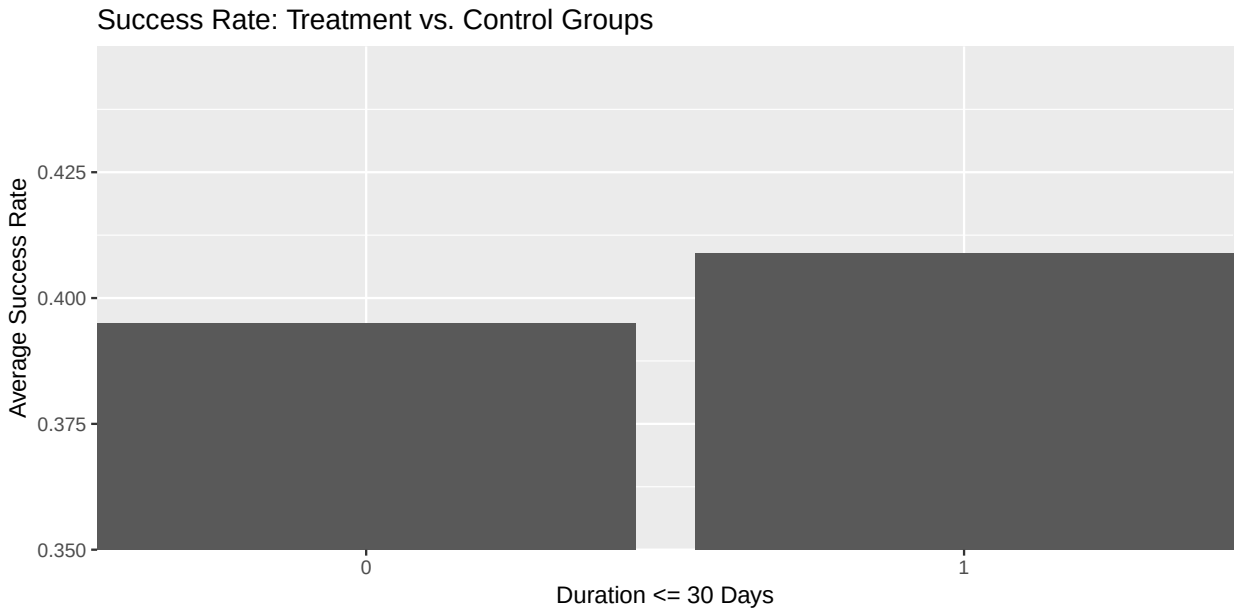
```
##
## Call:
## lm(formula = success ~ duration.days, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5520 -0.4217 -0.3453  0.5784  0.8570
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.565e-01  2.413e-03  230.58   <2e-16 ***
## duration.days -4.494e-03  6.656e-05  -67.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4873 on 331662 degrees of freedom
## Multiple R-squared:  0.01356,    Adjusted R-squared:  0.01355
## F-statistic:  4558 on 1 and 331662 DF,  p-value: < 2.2e-16
```

```
rcorr(ks_df$success, ks_df$duration.days)
```

```
##       x     y
## x  1.00 -0.12
## y -0.12  1.00
##
## n= 331664
##
##
## P
##   x  y
## x     0
```

```
## y  0
```

```
# with binary treatment effect
ks_df %>%
  group_by(treat) %>%
  summarise(average.success = mean(success)) %>%
  ggplot(aes(x = treat, y = average.success)) +
  geom_col() +
  coord_cartesian(ylim = c(.35, .45), expand = FALSE) +
  labs(
    colour = '',
    x = 'Duration <= 30 Days',
    y = 'Average Success Rate',
    title = 'Success Rate: Treatment vs. Control Groups'
  )
```




```
summary(lm(success ~ treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4089 -0.4089 -0.3950  0.5911  0.6050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.395043   0.001415 279.194  < 2e-16 ***
## treat1      0.013855   0.001772   7.818 5.37e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4906 on 331662 degrees of freedom
## Multiple R-squared:  0.0001843,  Adjusted R-squared:  0.0001813
```

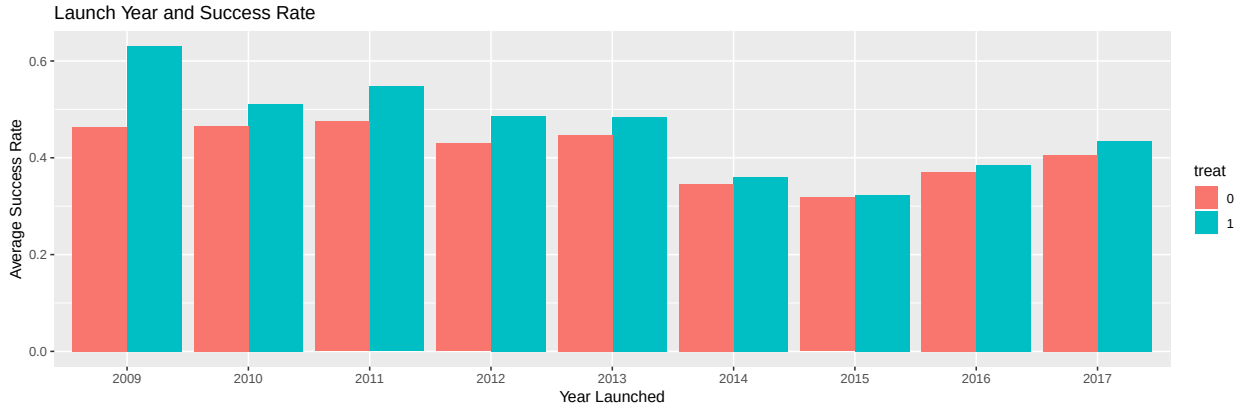
```
## F-statistic: 61.13 on 1 and 331662 DF,  p-value: 5.367e-15
```

Evaluating Potential Confounders: Time, Category, Country, and Goal

We see some significant effects from each of these potential confounders on `success` rate. If they were not counfounding, we sould expect the estimated $\hat{\beta}$ values to equal 0 (p-values higher than `0.05`). In other words, we would espect that regressing `success` on any of these X values would show no evidence of a relationship. We also see that there is a mild correlation coefficient with a strong p value with respect to `goal` and `success`. There is enough evidence to necessitate handling these as confounders

Success on Time

```
# Year as a potential confounder
ks_df %>%
  group_by(launch.year, treat) %>%
  summarise(mean.success = mean(success)) %>%
  ggplot() +
  geom_col(aes(x = launch.year, y = mean.success, fill = treat), position = 'dodge') +
  labs(
    colour = '',
    x = 'Year Launched',
    y = 'Average Success Rate',
    title = 'Launch Year and Success Rate'
  )
```

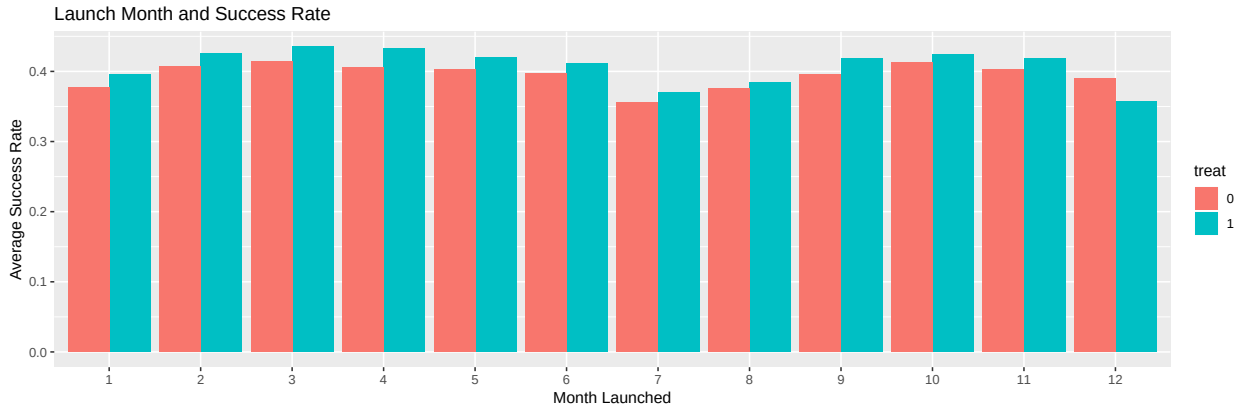



```
summary(lm(success ~ launch.year + treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ launch.year + treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5221 -0.4060 -0.3306  0.5525  0.6975
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.486522   0.014173  34.327  < 2e-16 ***
## launch.year2010 -0.015209   0.015019  -1.013  0.31123
## launch.year2011  0.007486   0.014521   0.516  0.60618
## launch.year2012 -0.039064   0.014409  -2.711  0.00671 **
```

```
## launch.year2013 -0.033017   0.014401  -2.293  0.02186 *
## launch.year2014 -0.149726   0.014340 -10.441  < 2e-16 ***
## launch.year2015 -0.184003   0.014326 -12.844  < 2e-16 ***
## launch.year2016 -0.124488   0.014367  -8.665  < 2e-16 ***
## launch.year2017 -0.080512   0.014391  -5.594 2.21e-08 ***
## treat1           0.028079   0.001790  15.686  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4866 on 331654 degrees of freedom
## Multiple R-squared:  0.01676,    Adjusted R-squared:  0.01673
## F-statistic:   628 on 9 and 331654 DF,  p-value: < 2.2e-16
```

```r
# Month as a potential confounder
ks_df %>%
  group_by(launch.month, treat) %>%
  summarise(mean.success = mean(success)) %>%
  ggplot() +
  geom_col(aes(x = launch.month, y = mean.success, fill = treat), position = 'dodge') +
  labs(
    colour = '',
    x = 'Month Launched',
    y = 'Average Success Rate',
    title = 'Launch Month and Success Rate'
  )
```



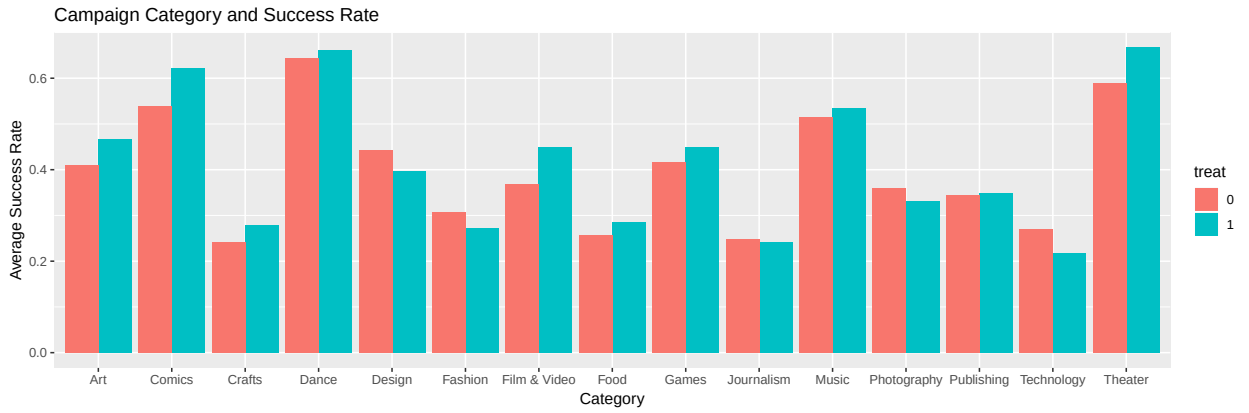

```r
summary(lm(success ~ launch.month + treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ launch.month + treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4326 -0.4140 -0.3765  0.5820  0.6445
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.379477   0.003324 114.155  < 2e-16 ***
## launch.month2   0.030065   0.004358   6.899 5.24e-12 ***
## launch.month3   0.038894   0.004222   9.211  < 2e-16 ***
```

```
## launch.month4   0.034522   0.004273   8.080 6.49e-16 ***
## launch.month5   0.024786   0.004252   5.829 5.59e-09 ***
## launch.month6   0.017589   0.004259   4.130 3.63e-05 ***
## launch.month7  -0.024029   0.004166  -5.767 8.06e-09 ***
## launch.month8  -0.007918   0.004279  -1.851 0.064234 .
## launch.month9   0.021077   0.004314   4.885 1.03e-06 ***
## launch.month10  0.031996   0.004246   7.536 4.84e-14 ***
## launch.month11  0.024275   0.004275   5.678 1.36e-08 ***
## launch.month12 -0.017258   0.004782  -3.609 0.000307 ***
## treat1          0.014263   0.001772   8.047 8.48e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4902 on 331651 degrees of freedom
## Multiple R-squared:  0.001904,   Adjusted R-squared:  0.001868
## F-statistic: 52.72 on 12 and 331651 DF,  p-value: < 2.2e-16
```

Success on Category

```
# main category as a potential confounder
ks_df %>%
  group_by(main_category, treat) %>%
  summarise(mean.success = mean(success)) %>%
  ggplot() +
  geom_col(aes(x = main_category, y = mean.success, fill = treat), position = 'dodge') +
  labs(
    colour = '',
    x = 'Category',
    y = 'Average Success Rate',
    title = 'Campaign Category and Success Rate'
  )
```



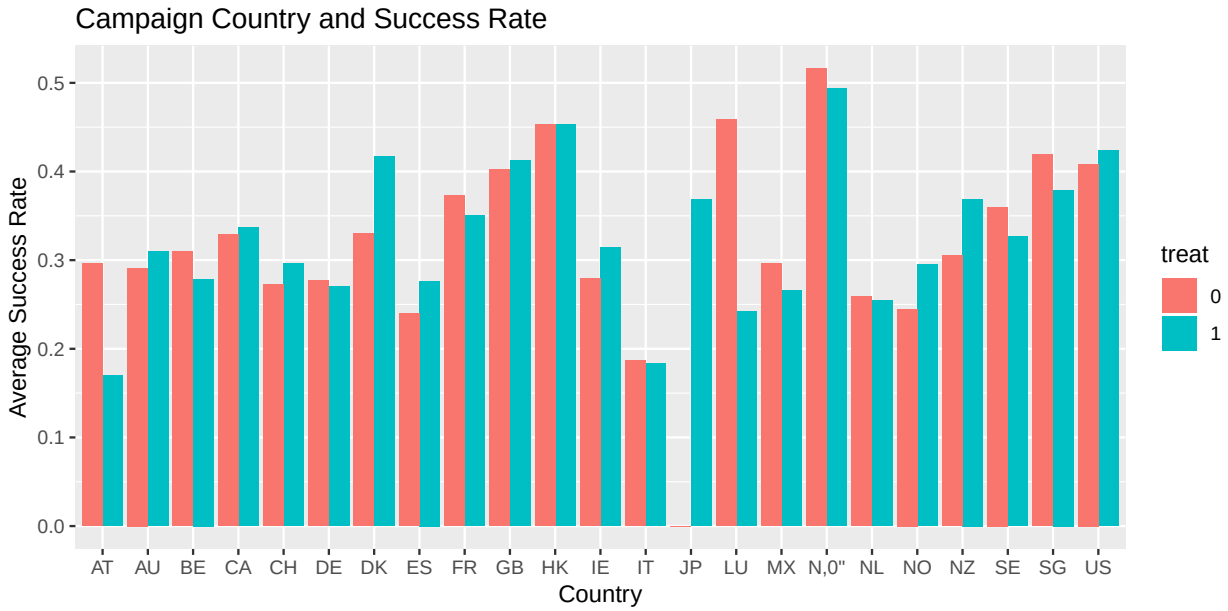

```
summary(lm(success ~ main_category + treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ main_category + treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6621 -0.4247 -0.2686  0.5542  0.7748
```

```
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.434435   0.003224 134.763  < 2e-16 ***
## main_categoryComics         0.143798   0.005673  25.346  < 2e-16 ***
## main_categoryCrafts        -0.179959   0.006190 -29.072  < 2e-16 ***
## main_categoryDance          0.206863   0.008555  24.182  < 2e-16 ***
## main_categoryDesign        -0.030569   0.004247  -7.198 6.11e-13 ***
## main_categoryFashion       -0.165869   0.004533 -36.588  < 2e-16 ***
## main_categoryFilm & Video -0.029179   0.003610  -8.083 6.32e-16 ***
## main_categoryFood          -0.172397   0.004399 -39.186  < 2e-16 ***
## main_categoryGames         -0.009504   0.004123  -2.305   0.0211 *
## main_categoryJournalism    -0.204435   0.008016 -25.504  < 2e-16 ***
## main_categoryMusic          0.079856   0.003738  21.363  < 2e-16 ***
## main_categoryPhotography   -0.107134   0.005713 -18.754  < 2e-16 ***
## main_categoryPublishing    -0.101253   0.003927 -25.781  < 2e-16 ***
## main_categoryTechnology    -0.209265   0.004178 -50.092  < 2e-16 ***
## main_categoryTheater        0.190383   0.005600  33.994  < 2e-16 ***
## treat1                      0.020842   0.001737  11.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 331648 degrees of freedom
## Multiple R-squared:  0.04702,    Adjusted R-squared:  0.04697
## F-statistic:  1091 on 15 and 331648 DF,  p-value: < 2.2e-16
```

Success on Country

```
# country as a confounder
ks_df %>%
  group_by(country, treat) %>%
  summarise(mean.success = mean(success)) %>%
  ggplot() +
  geom_col(aes(x = country, y = mean.success, fill = treat), position = 'dodge') +
  labs(
    colour = '',
    x = 'Country',
    y = 'Average Success Rate',
    title = 'Campaign Country and Success Rate'
  )
```



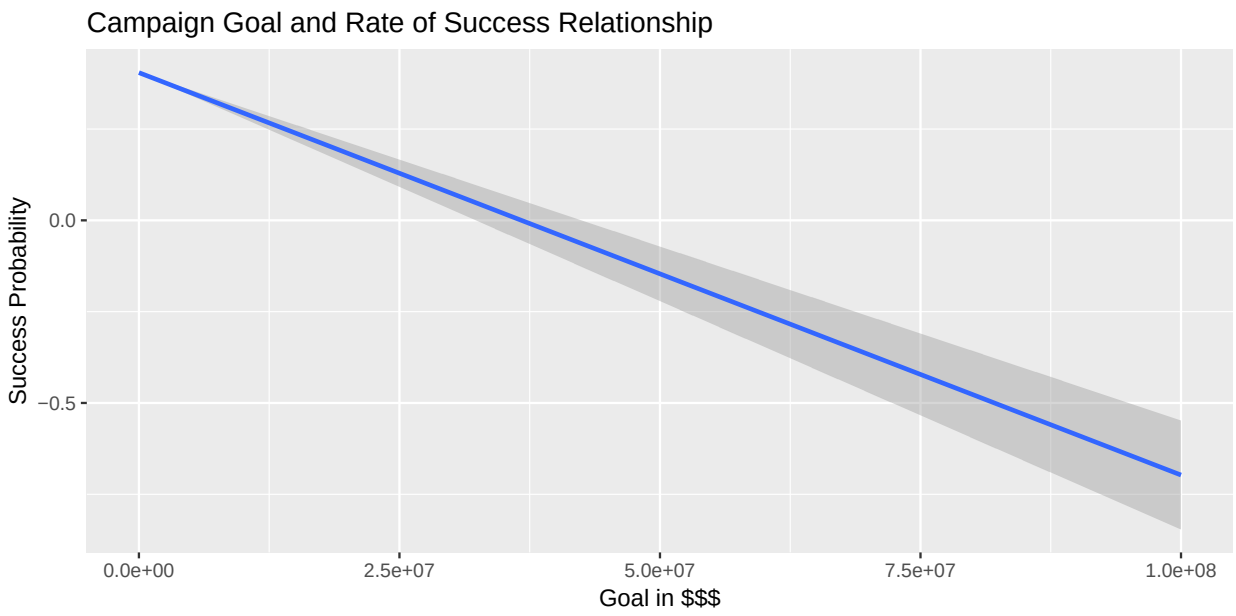

```
summary(lm(success ~ country + treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ country + treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5041 -0.4234 -0.3997  0.5766  0.8222
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21238    0.02223   9.554  < 2e-16 ***
## countryAU    0.08192    0.02301   3.561  0.00037 ***
## countryBE    0.06971    0.03083   2.261  0.02373 *
## countryCA    0.11301    0.02264   4.993 5.96e-07 ***
## countryCH    0.06620    0.02932   2.258  0.02396 *
## countryDE    0.05133    0.02372   2.164  0.03047 *
## countryDK    0.16709    0.02741   6.096 1.09e-09 ***
## countryES    0.04165    0.02491   1.672  0.09459 .
## countryFR    0.13998    0.02425   5.773 7.80e-09 ***
## countryGB    0.18728    0.02239   8.365  < 2e-16 ***
## countryHK    0.23219    0.03153   7.363 1.80e-13 ***
## countryIE    0.08136    0.02904   2.802  0.00508 **
## countryIT   -0.03453    0.02437  -1.417  0.15657
## countryJP    0.08055    0.10436   0.772  0.44021
## countryLU    0.11295    0.06847   1.650  0.09903 .
## countryMX    0.06112    0.02574   2.374  0.01757 *
## countryN,0"  0.27787    0.04040   6.879 6.04e-12 ***
## countryNL    0.03422    0.02434   1.406  0.15972
## countryNO    0.05680    0.03007   1.889  0.05886 .
## countryNZ    0.12921    0.02609   4.952 7.35e-07 ***
## countrySE    0.11546    0.02552   4.523 6.09e-06 ***
## countrySG    0.17037    0.03193   5.335 9.56e-08 ***
```

```
## countryUS    0.19714    0.02223   8.870  < 2e-16 ***
## treat1       0.01383    0.00177   7.812 5.63e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.489 on 331640 degrees of freedom
## Multiple R-squared:  0.006842,   Adjusted R-squared:  0.006773
## F-statistic: 99.34 on 23 and 331640 DF,  p-value: < 2.2e-16
```

Success on Goal

```
# goal as a confounder
ks_df %>%
  ggplot(aes(y = success, x = goal)) +
  geom_smooth(method = 'lm') +
  labs(
    colour = '',
    x = 'Goal in $$$',
    y = 'Success Probability',
    title = 'Campaign Goal and Rate of Success Relationship'
  )
```




```
# Strong evidence of a linear relationship
# correlation is 0.03, strong p-value
summary(lm(success ~ goal + treat, data = ks_df))
```

```
##
## Call:
## lm(formula = success ~ goal + treat, data = ks_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4093 -0.4091 -0.3956  0.5908  0.6973
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.958e-01  1.415e-03 279.614  < 2e-16 ***
## goal        -1.093e-08  7.619e-10 -14.346  < 2e-16 ***
## treat1       1.348e-02  1.772e-03   7.607 2.82e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4905 on 331661 degrees of freedom
## Multiple R-squared:  0.0008043,  Adjusted R-squared:  0.0007983
## F-statistic: 133.5 on 2 and 331661 DF,  p-value: < 2.2e-16
```

```
rcorr(ks_df$success, ks_df$goal)
```

```
##       x     y
## x  1.00 -0.03
## y -0.03  1.00
##
## n= 331664
##
##
## P
##   x  y
## x     0
## y  0
```

Propensity Score Matching

```
# data are imbalanced in terms of treatment
# we will use matching to address this
table(ks_df$treat)
```

```
##
##      0      1
## 120235 211429
```

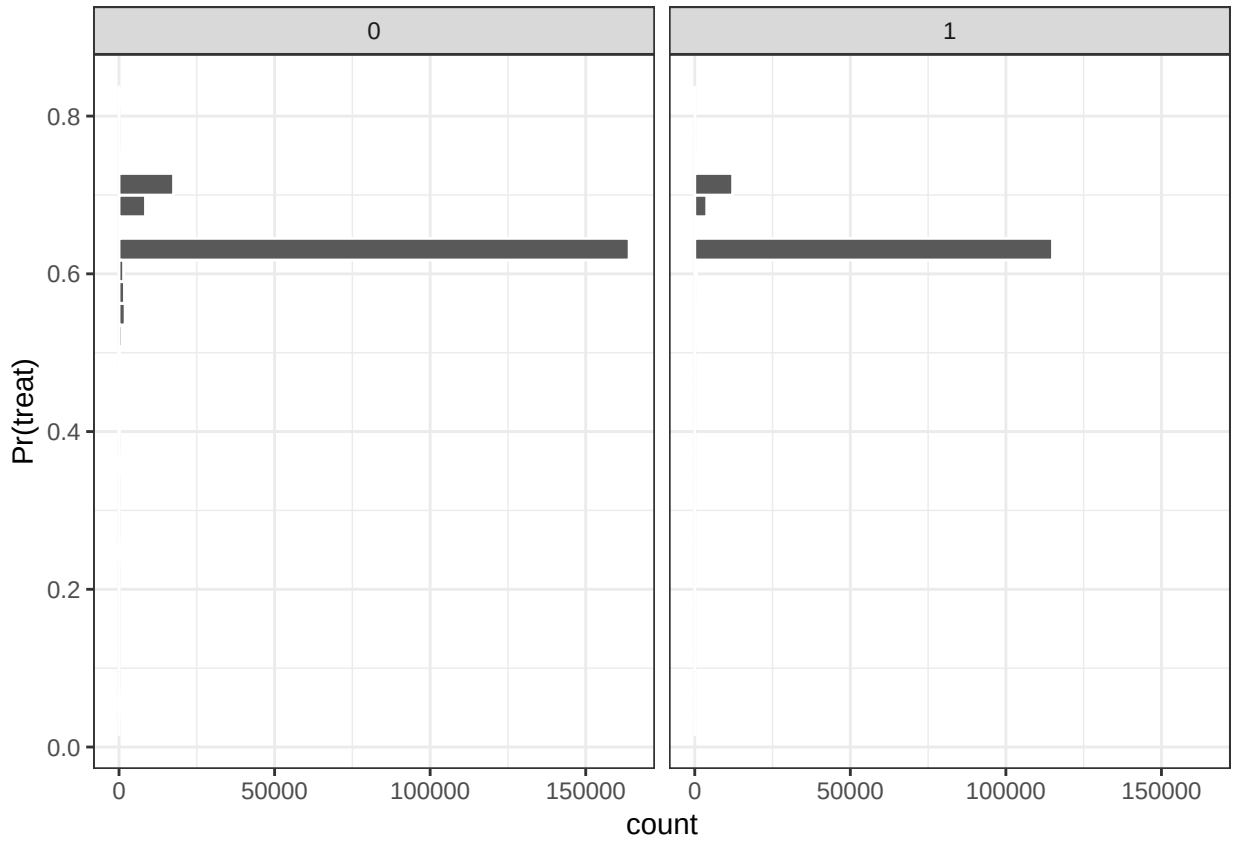
We will address the imbalance between the treatment group and control group with propensity score matching. We will match in terms of `country`, one of our categorical confounders, and `goal`, a continuous confounder that cannot be handled with fixed effects.

Balance in terms of propensity score, pre matching

There is a significant degree of imbalance in regards to both `goal` and `country`. We address this imbalance with propensity score matching. The desired outcome is a dataset that is 'as-good-as' reandom with regards to the two confounders.

```
# generate propensity scores for each record
ks_df$PS <- glm(
  treat ~ goal + country,
  data = ks_df,
  family = "binomial"
)$fitted.values

# visualize propensity score balance
ggplot(ks_df, aes(x = PS)) +
```

```
  geom_histogram(color = "white") +
  facet_wrap(~success) +
  xlab("Pr(treat)") + theme_bw() + coord_flip()
```




```
# balance test
print(
  CreateTableOne(
    vars = c("goal", "country"),
    strata = "treat",
    test = TRUE,
    data = ks_df
  ), smd = TRUE
)
```

```
##                     Stratified by treat
##                      0                     1                     p      test
##   n                    120235                211429
##   goal (mean (sd)) 66296.20 (1464770.75) 31716.83 (860201.39) <0.001
##   country (%)                                                  <0.001
##      AT                  196 ( 0.2)            289 ( 0.1)
##      AU                 2064 ( 1.7)           4552 ( 2.2)
##      BE                  200 ( 0.2)            323 ( 0.2)
##      CA                 4494 ( 3.7)           7876 ( 3.7)
##      CH                  264 ( 0.2)            388 ( 0.2)
##      DE                 1254 ( 1.0)           2181 ( 1.0)
##      DK                  303 ( 0.3)            623 ( 0.3)
```

```
##      ES                 701 ( 0.6)          1172 ( 0.6)
##      FR                1069 ( 0.9)          1451 ( 0.7)
##      GB                8080 ( 6.7)         21374 (10.1)
##      HK                 192 ( 0.2)           285 ( 0.1)
##      IE                 222 ( 0.2)           461 ( 0.2)
##      IT                1091 ( 0.9)          1278 ( 0.6)
##      JP                   4 ( 0.0)            19 ( 0.0)
##      LU                  24 ( 0.0)            33 ( 0.0)
##      MX                 681 ( 0.6)           730 ( 0.3)
##      N,0"                62 ( 0.1)           148 ( 0.1)
##      NL                 787 ( 0.7)          1624 ( 0.8)
##      NO                 196 ( 0.2)           386 ( 0.2)
##      NZ                 347 ( 0.3)           927 ( 0.4)
##      SE                 475 ( 0.4)          1034 ( 0.5)
##      SG                 148 ( 0.1)           306 ( 0.1)
##      US               97381 (81.0)        163969 (77.6)
##                    Stratified by treat
##                     SMD
##   n
##   goal (mean (sd))  0.029
##   country (%)       0.143
##      AT
##      AU
##      BE
##      CA
##      CH
##      DE
##      DK
##      ES
##      FR
##      GB
##      HK
##      IE
##      IT
##      JP
##      LU
##      MX
##      N,0"
##      NL
##      NO
##      NZ
##      SE
##      SG
##      US
```

Matching process

Due to processing time of matching (over 2 hours), we will knit the final version of the appendix after loading the matched data, rather than doing it during the knitting process.

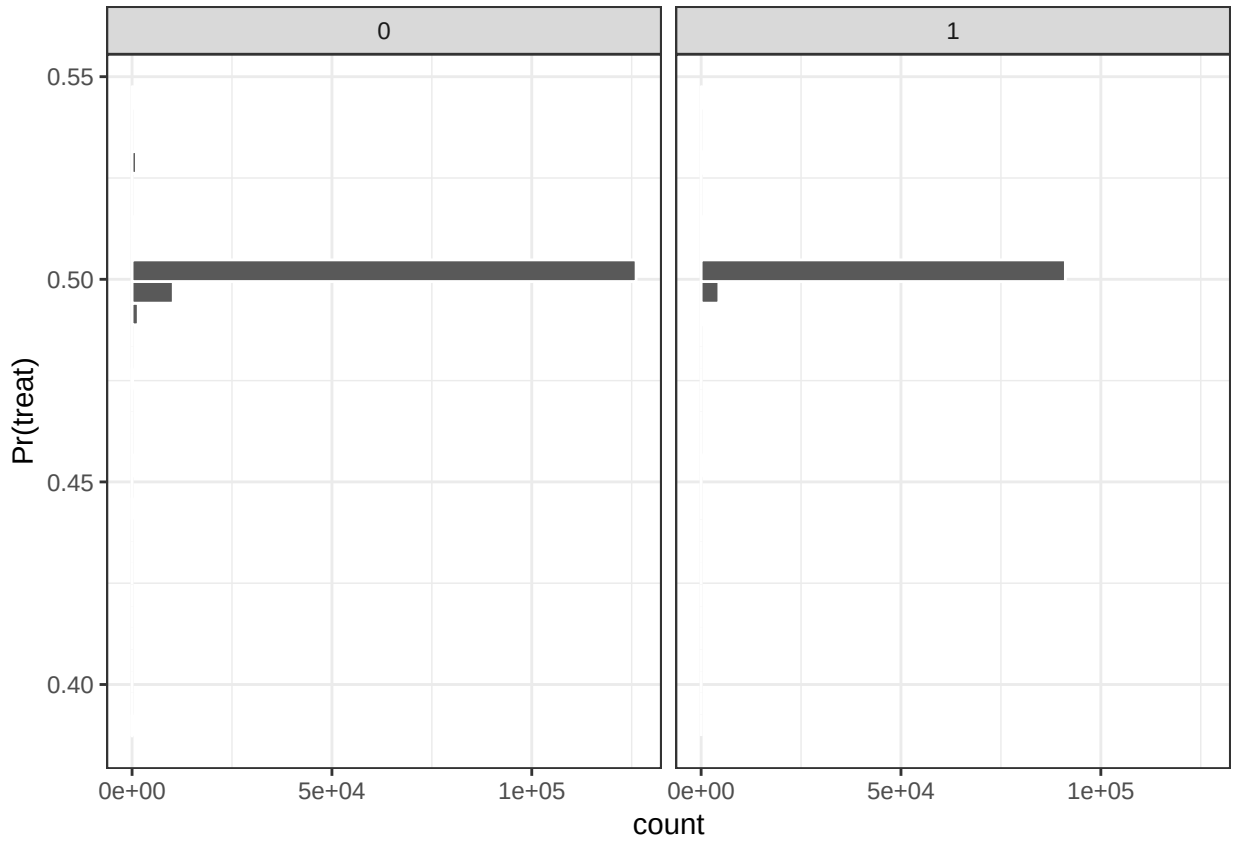
```
# we use a checkpoint file to avoid running matchit
##################
# m.out = matchit(
#   treat ~ goal + country,
#   data = ks_df,
```

```
#   method = 'nearest',
#   distance = "logit",
#   caliper = 0.001,
#   replace = FALSE
# )
#
# # prune the original data
# ks_df.matched <- ks_df[
#   ks_df$ID %in% data.table(match.data(m.out))$ID,
# ]
#
# checkpoint
# write.csv(ks_df.matched, paste(file_dir, 'KickStarterMatched.csv', sep=''))
###################

# load checkpoint
ks_df.matched <- read.csv(paste(file_dir, 'KickStarterMatched.csv', sep = ''))
ks_df.matched <- ks_df[ks_df$ID %in% ks_df.matched$ID,]
```

Balance in terms of propensity score, pre matching

Our balance is not perfect, but it is much better than before. Moreover, our significance tests fail to reject the null hypotheses, that the groups are balanced.

```
# generate propensity scores for each record (if loading checkpoint)
ks_df.matched$PS <- glm(
  treat ~ goal + country,
  data = ks_df.matched,
  family = "binomial"
)$fitted.values

# visualize propensity score balance
ggplot(ks_df.matched, aes(x = PS)) +
  geom_histogram(color = "white") +
  facet_wrap(~success) +
  xlab("Pr(treat)") + theme_bw() + coord_flip()
```




```
# balance test
print(
  CreateTableOne(
    vars = c("goal", "country"),
    strata = "treat",
    test = TRUE,
    data = ks_df.matched
  ), smd = TRUE
)
```

```
##                     Stratified by treat
##                      0                     1                     p      test
##   n                    119082                119082
##   goal (mean (sd)) 41329.28 (906361.01) 37871.01 (899791.48)  0.350
##   country (%)                                                  0.896
##      AT                  176 ( 0.1)            179 ( 0.2)
##      AU                 2007 ( 1.7)           1973 ( 1.7)
##      BE                  177 ( 0.1)            173 ( 0.1)
##      CA                 4418 ( 3.7)           4405 ( 3.7)
##      CH                  248 ( 0.2)            241 ( 0.2)
##      DE                 1220 ( 1.0)           1220 ( 1.0)
##      DK                  270 ( 0.2)            325 ( 0.3)
##      ES                  687 ( 0.6)            773 ( 0.6)
##      FR                 1025 ( 0.9)           1027 ( 0.9)
##      GB                 8033 ( 6.7)           8031 ( 6.7)
##      HK                  168 ( 0.1)            174 ( 0.1)
##      IE                  204 ( 0.2)            209 ( 0.2)
```

```
##      IT                  989 ( 0.8)           990 ( 0.8)
##      JP                    3 ( 0.0)             3 ( 0.0)
##      LU                   20 ( 0.0)            21 ( 0.0)
##      MX                  543 ( 0.5)           546 ( 0.5)
##      N,0"                 60 ( 0.1)            60 ( 0.1)
##      NL                  780 ( 0.7)           753 ( 0.6)
##      NO                  175 ( 0.1)           176 ( 0.1)
##      NZ                  329 ( 0.3)           329 ( 0.3)
##      SE                  444 ( 0.4)           469 ( 0.4)
##      SG                  148 ( 0.1)           124 ( 0.1)
##      US                96958 (81.4)         96881 (81.4)
##                    Stratified by treat
##                     SMD
##   n
##   goal (mean (sd))  0.004
##   country (%)       0.015
##      AT
##      AU
##      BE
##      CA
##      CH
##      DE
##      DK
##      ES
##      FR
##      GB
##      HK
##      IE
##      IT
##      JP
##      LU
##      MX
##      N,0"
##      NL
##      NO
##      NZ
##      SE
##      SG
##      US
```

```r
# overall balance
table(ks_df.matched$treat)
```

```
##
##      0      1
## 119082 119082
```

Fixed Effects Regression

Our final step in our appraoch to detecting the causal effect of our treatment on success rate is fised effects regression. We are eliminating any bias from static confounders with fixed effects regression, and we will fix `main_category` and our time variables, `launch.year` and `launch.month`.

Success on Treatment

Our entire econometric design has led to this model.

```
# fixed effects regression: success on treatment
FE_success_treat <- felm(
  data = ks_df.matched,
  success ~ treat | (main_category + launch.year + launch.month)
)
```

Success on Campaign Duration

Campaign managers may also be interested in the marginal effect of campaign duration days on success. We must note here that we are no longer using a binary treatment variable; `duration.days` is now our treatment. As such, we can no longer use our matched data, since that was balanced in terms of our binary treatment variable.

Since we are no longer matching, we can also try to include a control for `goal` and fix for `country`. When we do, we see no change in the $\hat{\beta}$ estimate for `duration.days`. We can assume that `goal` and `country` are not confounding the effect of `duration.days` on `success`.

```
# fixed effects regression: success on duration
FE_success_duration <- felm(
  data = ks_df,
  success ~ duration.days | (main_category + launch.year + launch.month)
)

# fixed effects regression: success on duration
FE_success_duration_goal <- felm(
  data = ks_df,
  success ~ duration.days + goal | (main_category + launch.year + launch.month)
)

FE_success_duration_FEcountry <- felm(
  data = ks_df,
  success ~ duration.days | (main_category + launch.year + launch.month + country)
)

stargazer(
  FE_success_duration,
  title = "Duration Effect on KS Success",
  type = "text",
  column.labels = "Success on Duration"
)
```

```
##
## Duration Effect on KS Success
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                success
##                          Success on Duration
## ------------------------------------------------
## duration.days                -0.005***
##                              (0.0001)
##
```

```
## ------------------------------------------------
## Observations                    331,664
## R2                               0.073
## Adjusted R2                      0.073
## Residual Std. Error     0.472 (df = 331629)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
stargazer(
  FE_success_duration_goal,
  FE_success_duration_FEcountry,
  title = "Duration on KS Success",
  type = "text",
  column.labels = c(
    "Success and Goal",
    "Success, FE Country"
  )
)
```

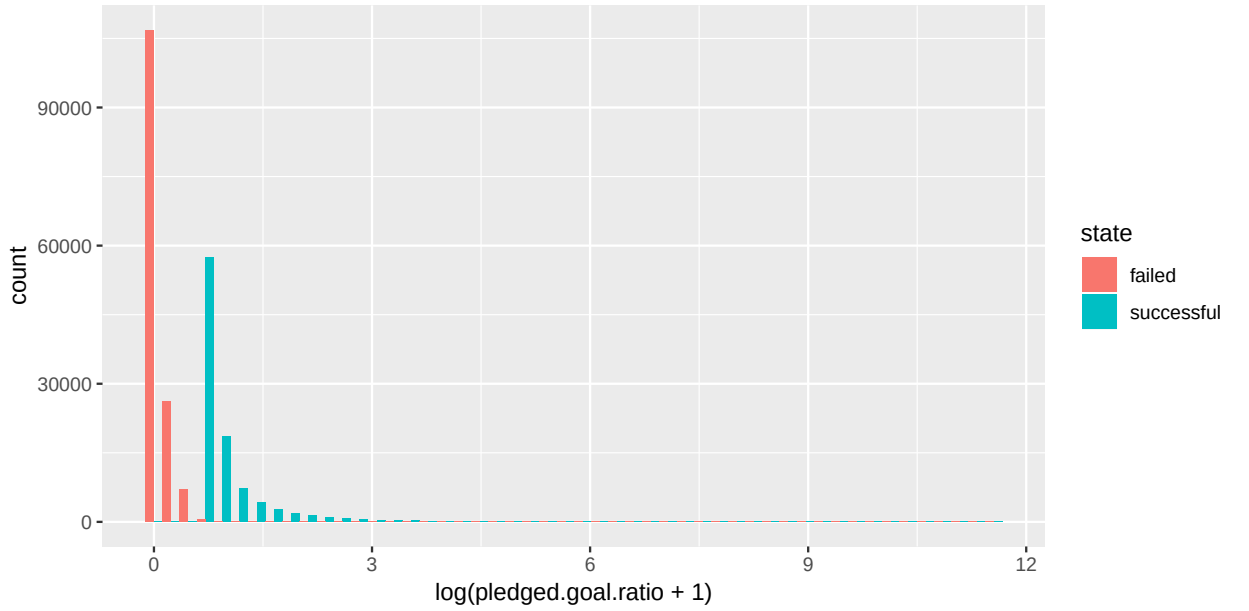
```
##
## Duration on KS Success
## ============================================================
##                                Dependent variable:
##                     ---------------------------------------
##                                       success
##                        Success and Goal   Success, FE Country
##                               (1)                 (2)
## ------------------------------------------------------------
## duration.days              -0.005***           -0.005***
##                            (0.0001)            (0.0001)
##
## goal                       -0.000***
##                             (0.000)
##
## ------------------------------------------------------------
## Observations                331,664             331,664
## R2                           0.073               0.075
## Adjusted R2                  0.073               0.075
## Residual Std. Error 0.472 (df = 331628) 0.472 (df = 331607)
## ============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
```

Pledged to Goal Ratio on Treatment

We may also be interested in the ratio of total `pledged` amount to `goal` amount. In some cases, campaigns are set up with `$1` goals, and the idea of the campaign is to open endedly raise as much money as possible. In these cases, it may not be enough to raise `$1` and simply succeed at meeting the goal. We can model it with the `pledged.goal.ratio` as the outcome variable. Note that we take the `log` transformation of `pledged.goal.ratio` due to its skewness; even after taking that transformation, the variable still exhibits high skewness. We have enough data to assume that the model $\hat{\beta}$ estimates will approach their true means in spite of the skewness.

```
# visualize skewness
ks_df.matched %>%
  ggplot(aes(x = log(pledged.goal.ratio + 1), fill = state)) +
```

```
  geom_histogram(bins = 50, position = 'dodge')
```




```
# fixed effects regression: pledged.goal.ratio on treatment
FE_PtoG_success <- felm(
  data = ks_df.matched,
  log(pledged.goal.ratio + 1) ~ treat | (main_category + launch.year + launch.month)
)
```

Pledged to Goal Ratio on Campaign Duration

Considering the two possible treatment and outcome variables, it makes sense to check the fourth and final possible model. Our design was meant to address success rate, and we used matching with a binary treatment. This model takes advantage of neither of those benefits. As with our other continuous treatment model, we will not be able to use our matched data. Also, we should make sure neither `goal` nor `country` is confounding the effect of `duration.days` on `pledged.goal.ratio`. Once again, we do not see strong evidence that `goal` or `country` is confounding `duration.days` on `pledged.goal.ratio`.

```
# fixed effects regression: pledged.goal.ratio on duration
FE_PtoG_duration <- felm(
  data = ks_df,
  log(pledged.goal.ratio + 1) ~ duration.days |
    (main_category + launch.year + launch.month)
)

# fixed effects regression: success on duration
FE_PtoG_duration_goal <- felm(
  data = ks_df,
  log(pledged.goal.ratio + 1) ~ duration.days + goal |
    (main_category + launch.year + launch.month)
)

FE_PtoG_duration_FEcountry <- felm(
  data = ks_df,
  log(pledged.goal.ratio + 1) ~ duration.days |
```

```
    (main_category + launch.year + launch.month + country)
)

stargazer(
  FE_PtoG_duration,
  title = "Duration on PLedge/Goal Ratio",
  type = "text",
  column.labels = "PLedge/Goal"
)
```

```
##
## Duration on PLedge/Goal Ratio
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                     log(pledged.goal.ratio + 1)
##                             PLedge/Goal
## ------------------------------------------------
## duration.days                -0.005***
##                              (0.0001)
##
## ------------------------------------------------
## Observations                  331,664
## R2                             0.061
## Adjusted R2                    0.061
## Residual Std. Error     0.554 (df = 331629)
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
stargazer(
  FE_PtoG_duration_goal,
  FE_PtoG_duration_FEcountry,
  title = "Duration on PLedge/Goal Ratio",
  type = "text",
  column.labels = c(
    "Pledge/Goal and Goal",
    "Pledge/Goal, FE Country"
  )
)
```

```
##
## Duration on PLedge/Goal Ratio
## ============================================================
##                                 Dependent variable:
##                     ---------------------------------------
##                             log(pledged.goal.ratio + 1)
##                     Pledge/Goal and Goal Pledge/Goal, FE Country
##                             (1)                   (2)
## ------------------------------------------------------------
## duration.days            -0.005***             -0.005***
##                           (0.0001)              (0.0001)
##
## goal                     -0.000***
##                           (0.000)
##
```

```
## ---------------------------------------------------------------
## Observations              331,664                 331,664
## R2                         0.061                   0.064
## Adjusted R2                0.061                   0.063
## Residual Std. Error 0.554 (df = 331628)    0.553 (df = 331607)
## ===============================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

Final FE Regression Output and Results

Our final results show our best attempts at detecting the causal effect. Because we used fixed effects regression, our treatment effect is always in terms of the mean. We interpret the four regressions like so:

- *Success on Treatment*: Setting a campaign duration to 30 days or less increases the probability of success by 4%
- *Success on Duration*: Increasing campaign duration by 1 day decreases the chance of success by 0.5%
- *Pledged/Goal on Treatment*: Setting a campaign duration to 30 days or less increases the Pledged/Goal ratio by 4%
- *Pledged/Goal on Duration*: Increasing campaign duration by 1 day decreases the Pledged/Goal ratio by 0.5%

```r
stargazer(
  FE_success_treat,
  FE_success_duration,
  title = "Fixed Effects: Success",
  type = "text",
  column.labels = c(
    "Success on Treatment",
    "Success on Duration"
  )
)
```

```
##
## Fixed Effects: Success
## ============================================================
##                               Dependent variable:
##                     -----------------------------------------
##                                     success
##                     Success on Treatment Success on Duration
##                             (1)                   (2)
## ------------------------------------------------------------
## treat1                     0.039***
##                            (0.002)
##
## duration.days                                    -0.005***
##                                                  (0.0001)
##
## ------------------------------------------------------------
## Observations              238,164                 331,664
## R2                         0.053                   0.073
## Adjusted R2                0.053                   0.073
## Residual Std. Error 0.478 (df = 238129)  0.472 (df = 331629)
## ============================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

```r
stargazer(
  FE_PtoG_success,
  FE_PtoG_duration,
  title = "Fixed Effects: Pledged/Goal Ratio",
  type = "text",
  column.labels = c(
    "Pledged/Goal on Treatment",
    "Pledged/Goal on Duration"
  )
)
```

```
##
## Fixed Effects: Pledged/Goal Ratio
## ================================================================================
##                                        Dependent variable:
##                          ---------------------------------------------------
##                                    log(pledged.goal.ratio + 1)
##                          Pledged/Goal on Treatment Pledged/Goal on Duration
##                                    (1)                       (2)
## --------------------------------------------------------------------------------
## treat1                            0.050***
##                                    (0.002)
##
## duration.days                                                -0.005***
##                                                               (0.0001)
##
## --------------------------------------------------------------------------------
## Observations                      238,164                    331,664
## R2                                 0.046                      0.061
## Adjusted R2                        0.046                      0.061
## Residual Std. Error    0.588 (df = 238129)        0.554 (df = 331629)
## ================================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```