TSTV-Obs_full.R

danny

2020-02-29

```
#*** Mochen Yang ***#
#*** Modified Original Script by Gordon Burtch ***
#*** Propensity Score Matching ***#

library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(MatchIt)
```

```
## Warning: package 'MatchIt' was built under R version 3.6.2
```

```
# import data
data = read.csv("TSTV-Obs-Dataset.csv")

# Data Exploration
# When does the treatment begin?
# This is recorded by the "after" variable
min(data$week)
```

```
## [1] 2220
```

```
max(data$week)
```

```
## [1] 2233
```

```r
min(data %>% filter(after==1) %>% select(week))
```

```
## [1] 2227
```

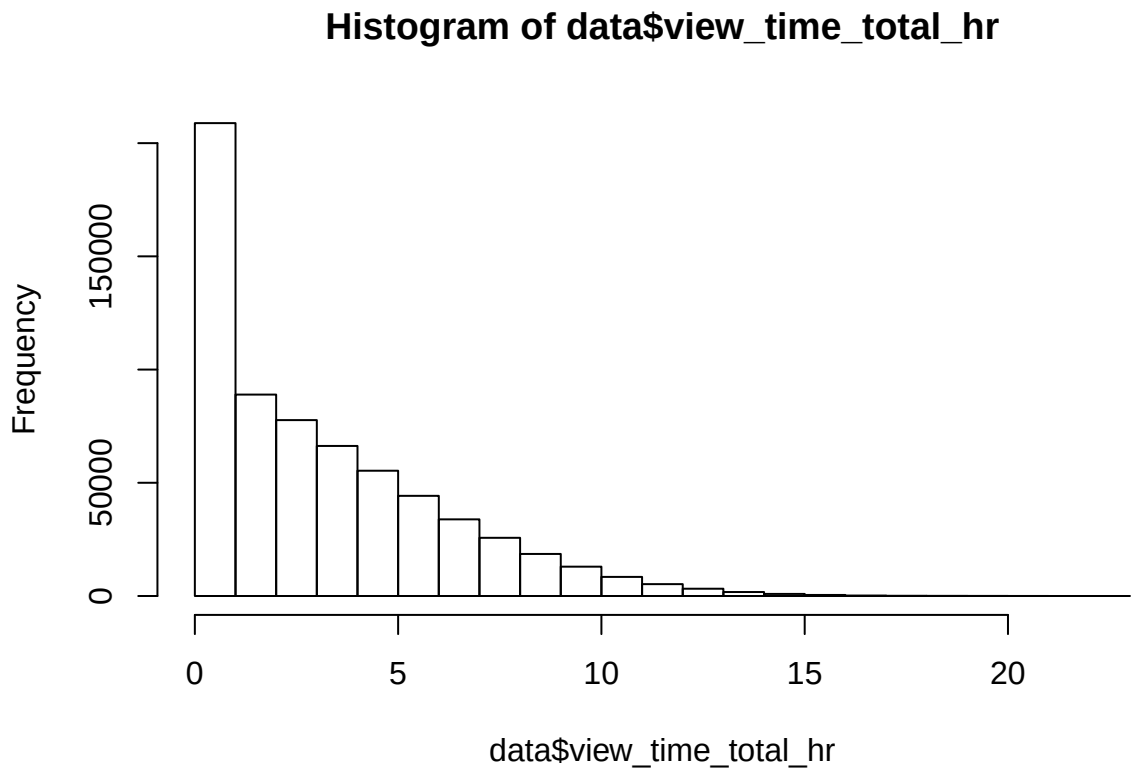
```r
# How many and what proportion of customers were treated with TSTV?
# This is recorded by the "premium" variable
data %>% filter(premium == 1) %>% select(id) %>% unique() %>% nrow()
```

```
## [1] 8348
```

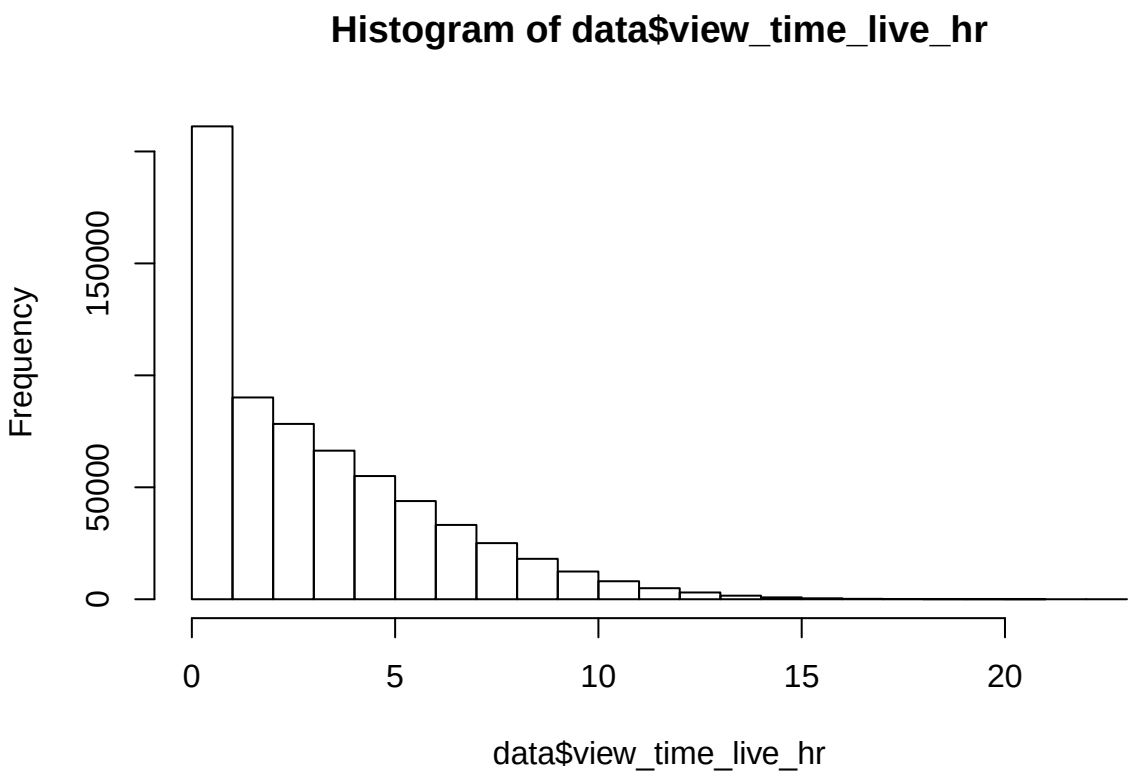
```r
data %>% filter(premium == 0) %>% select(id) %>% unique() %>% nrow()
```

```
## [1] 41686
```

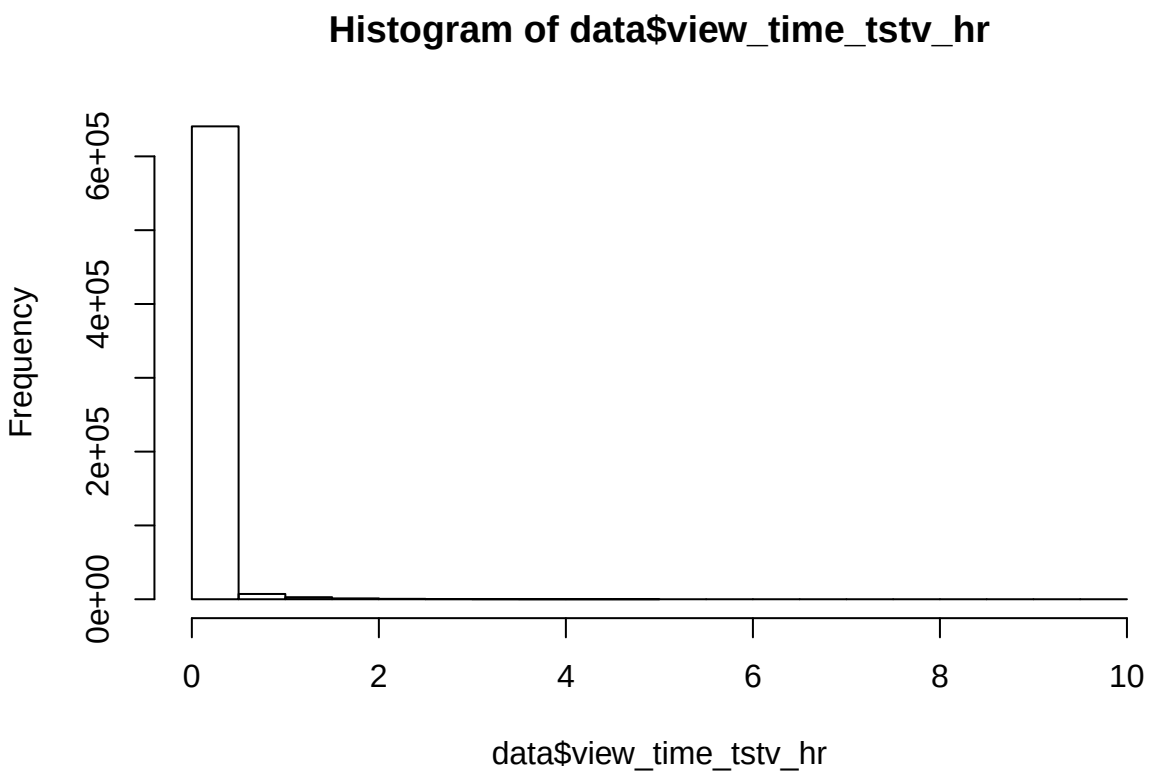
```r
#How are the viewership variables distributed?
hist(data$view_time_total_hr)
```



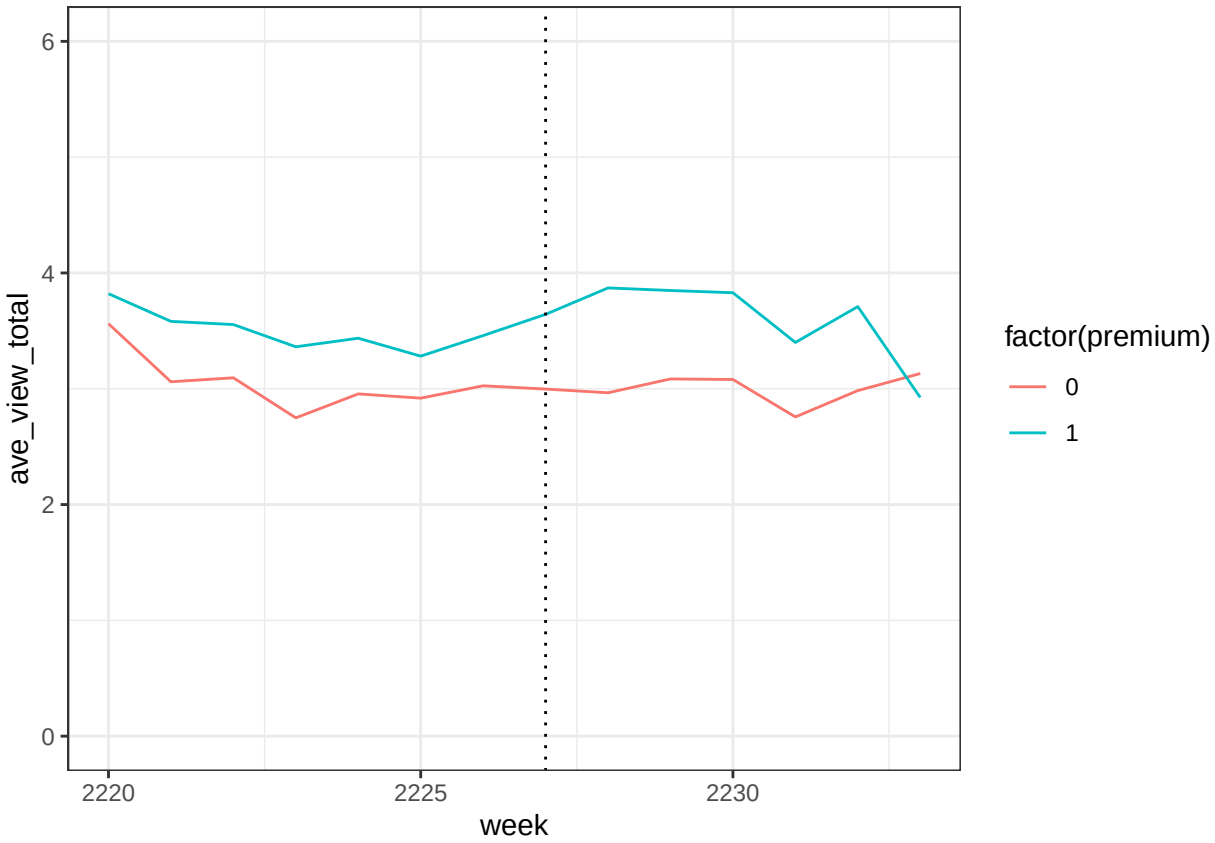

```r
hist(data$view_time_live_hr)
```



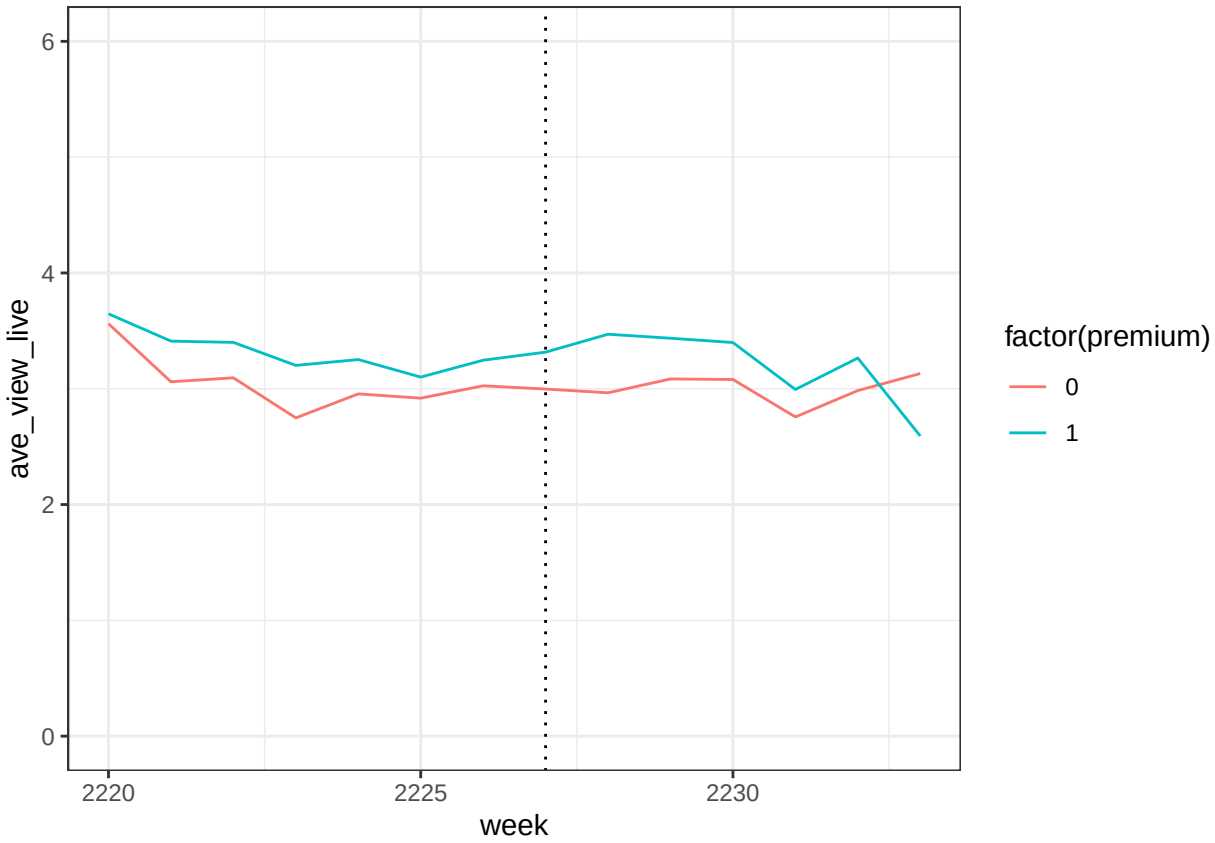

```
hist(data$view_time_tstv_hr)
```



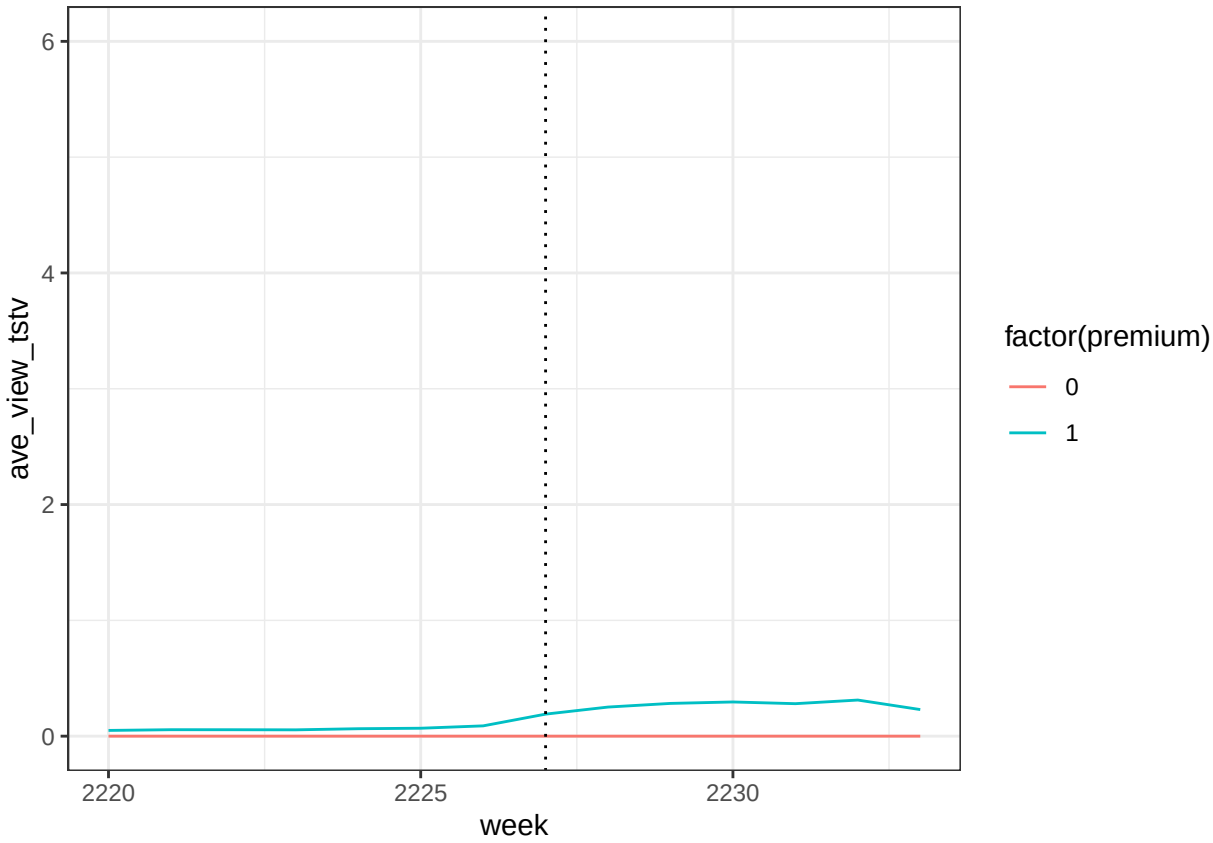

```
# Let's just look at what is going on with average viewership behavior for treated vs. untreated, in th
# Create aggregated viewership data by averaging across households in the same group
week_ave = data %>% group_by(week, premium) %>%
  summarise(ave_view_total = mean(view_time_total_hr),
            ave_view_live = mean(view_time_live_hr),
            ave_view_tstv = mean(view_time_tstv_hr)) %>% ungroup()

# plot for total TV time
ggplot(week_ave, aes(x = week, y = ave_view_total, color = factor(premium))) +
  geom_line() +
  geom_vline(xintercept = 2227, linetype='dotted') +
  ylim(0, 6) + xlim(2220,2233) +
  theme_bw()
```




```
# plot for live TV time
ggplot(week_ave, aes(x = week, y = ave_view_live, color = factor(premium))) +
  geom_line() +
  geom_vline(xintercept = 2227, linetype='dotted') +
  ylim(0, 6) + xlim(2220,2233) +
  theme_bw()
```



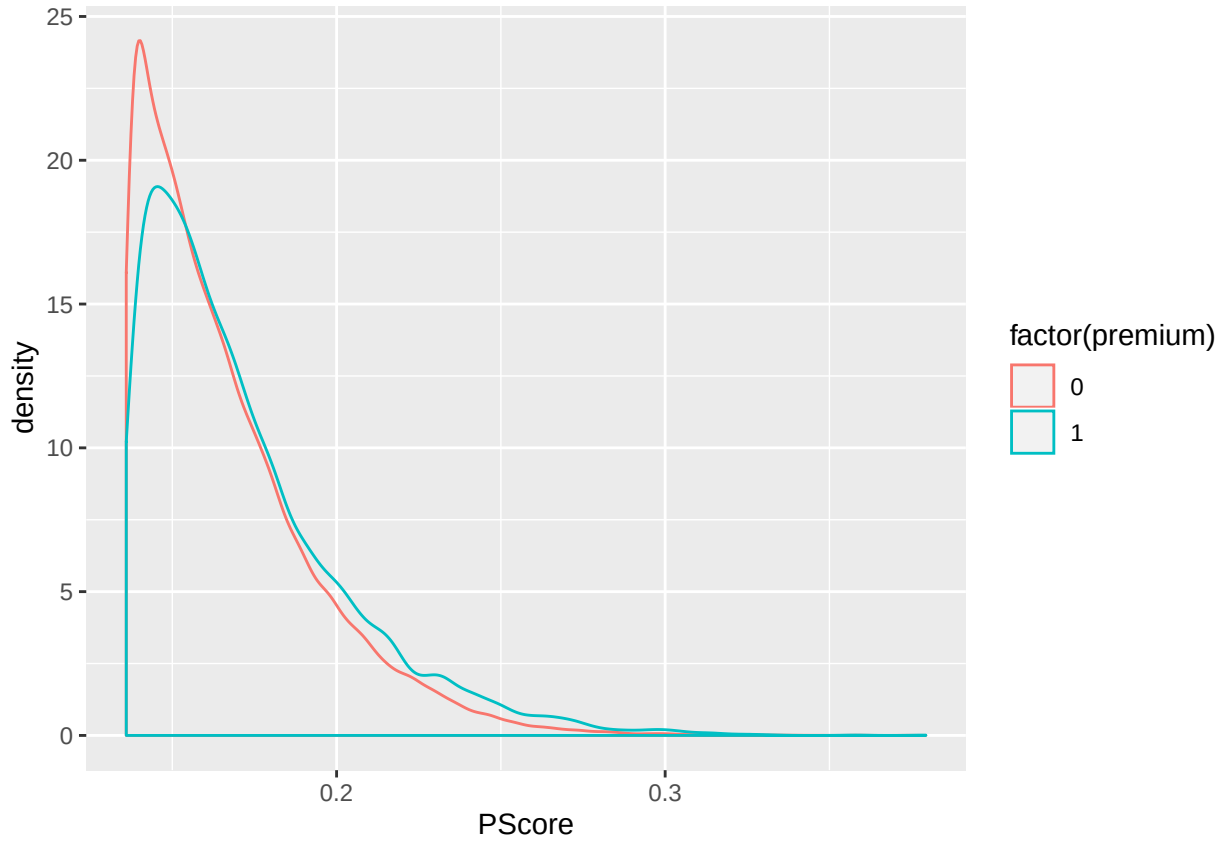

```
# plot for TSTV time
ggplot(week_ave, aes(x = week, y = ave_view_tstv, color = factor(premium))) +
  geom_line() +
  geom_vline(xintercept = 2227, linetype='dotted') +
  ylim(0, 6) + xlim(2220,2233) +
  theme_bw()
```




```
# Propensity Score Matching

#For this demonstration, we will use data from the pre-period for matching, then estimate the effect of

# create a dataset of before vs. after for convenience
data_summary = data %>% group_by(id, after) %>%
  summarise_all(mean) %>% ungroup()

# Check covariance balancing with t.test
data_pre = data_summary %>% filter(after == 0)
t.test(view_time_total_hr ~ premium, data = data_pre)
```

```
##
##  Welch Two Sample t-test
##
## data:  view_time_total_hr by premium
## t = -15.386, df = 11227, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5498987 -0.4256168
## sample estimates:
## mean in group 0 mean in group 1
##        2.975421        3.463179
```

```r
# Let's see what propensity scores distribution look like
PScore = glm(premium ~ view_time_total_hr, data = data_pre, family = "binomial")$fitted.values
data_pre$PScore = PScore
ggplot(data_pre, aes(x = PScore, color = factor(premium))) +
  geom_density()
```




```r
# Perform Matching
# Note: the matchit command may take a long time to run with large datasets
match_output <- matchit(premium ~ view_time_total_hr, data = data_pre, method = 'nearest', distance = "
summary(match_output)
```

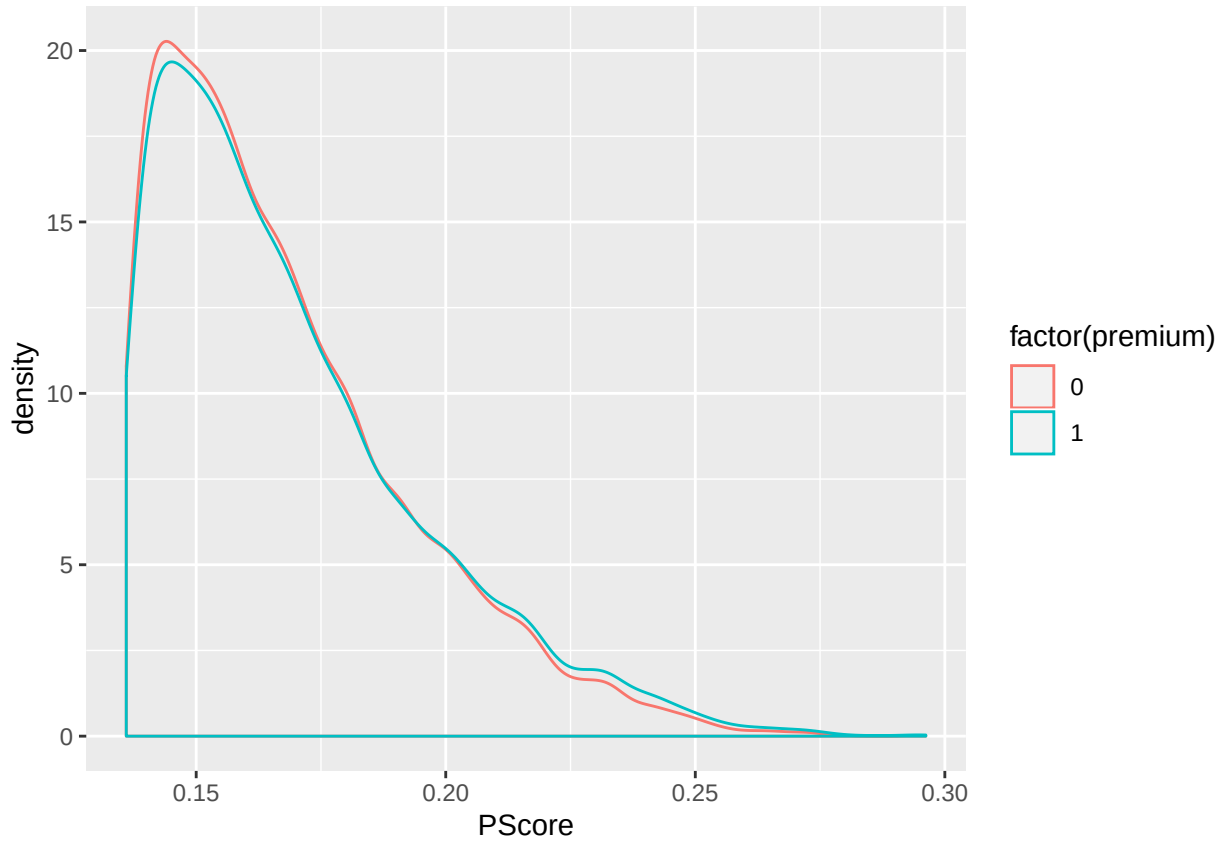
```
##
## Call:
## matchit(formula = premium ~ view_time_total_hr, data = data_pre,
##     method = "nearest", distance = "logit", caliper = 0.001,
##     replace = FALSE, ratio = 2)
##
## Summary of balance for all data:
##                     Means Treated Means Control SD Control Mean Diff
## distance                   0.1714        0.1659     0.0267    0.0055
## view_time_total_hr         3.4632        2.9754     2.4220    0.4878
##                     eQQ Med eQQ Mean eQQ Max
## distance             0.0044   0.0054  0.0249
## view_time_total_hr   0.4336   0.4873  1.6377
##
```

```
##
## Summary of balance for matched data:
##                    Means Treated Means Control SD Control Mean Diff
## distance                  0.1689        0.1689     0.0265     0e+00
## view_time_total_hr        3.2650        3.2648     2.4060     1e-04
##                    eQQ Med eQQ Mean eQQ Max
## distance            0.0006   0.0012  0.0154
## view_time_total_hr  0.0602   0.0985  1.0100
##
## Percent Balance Improvement:
##                    Mean Diff. eQQ Med eQQ Mean eQQ Max
## distance              99.9730 86.0078  78.8312 37.9170
## view_time_total_hr    99.9713 86.1144  79.7781 38.3297
##
## Sample sizes:
##           Control Treated
## All         41686    8348
## Matched     15969    8133
## Unmatched   25717     215
## Discarded       0       0
```

```
data_match = match.data(match_output)

# Evaluate covariance balance again, after matching
t.test(view_time_total_hr ~ premium, data = data_match)
```

```
##
##  Welch Two Sample t-test
##
## data:  view_time_total_hr by premium
## t = -3.0436, df = 15767, p-value = 0.002341
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.16175057 -0.03502594
## sample estimates:
## mean in group 0 mean in group 1
##        3.166565        3.264954
```

```
ggplot(data_match, aes(x = PScore, color = factor(premium))) +
  geom_density()
```




```
#Now let's estimate the treatment effect with vs. without matching.
data_post = data_summary %>% filter(after == 1)

model_unmatch = lm(log(view_time_total_hr+1)~ premium, data = data_post)
summary(model_unmatch)
```

```
##
## Call:
## lm(formula = log(view_time_total_hr + 1) ~ premium, data = data_post)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3441 -0.5183  0.0664  0.5199  1.7357
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.170892   0.003324  352.23   <2e-16 ***
## premium     0.173219   0.008025   21.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6662 on 48481 degrees of freedom
## Multiple R-squared:  0.009518,   Adjusted R-squared:  0.009498
## F-statistic: 465.9 on 1 and 48481 DF,  p-value: < 2.2e-16
```

```r
model_match = lm(log(view_time_total_hr+1)~ premium, data = data_post %>% filter(id %in% data_match$id))
summary(model_match)
```

```
##
## Call:
## lm(formula = log(view_time_total_hr + 1) ~ premium, data = data_post %>%
##     filter(id %in% data_match$id))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31890 -0.45564  0.08096  0.49555  1.49586
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.231906   0.005158 238.849   <2e-16 ***
## premium     0.086992   0.008783   9.904   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.64 on 23501 degrees of freedom
## Multiple R-squared:  0.004157,   Adjusted R-squared:  0.004114
## F-statistic: 98.09 on 1 and 23501 DF,  p-value: < 2.2e-16
```

```r
# What difference do you see, with and without matching?


# Sensitivity checks:
# 1. change caliper to 0.005
# 2. match with replacement
# 3. match 1 treated unit with 2 control units
```