

Untitled

Spencer Braun

11/8/2020

```
library(DOS2)
library(optmatch)
library(RITools)
library(rcbalance)

library(readstata13)
library(tidyverse)
library(haven)

source("utility.R")
```

Main Analysis

```
main <- read.csv("processed_data/main.csv")
# main %>% head()
# main %>% names()

# p2p
#transfersshare
#donationsshare
#gov
#y2006
#female
#exminister
#previous_vote
#media_mentions, quality
#years_served

analysis.cols <- main %>% select(
  p2p,
  current_vote,
  transfersshare,
  donationssshare,
  gov,
  y2006,
  female,
  exminister,
  previous_vote,
  media_mentions,
  quality,
  years_served
)
```

```

covariates <- c(
  "transfersshare",
  "donationssshare",
  "gov",
  "y2006",
  "female",
  "exminister",
  "previous_vote",
  "media_mentions",
  "quality",
  "years_served"
)

sapply(analysis.cols, function(x) sum(is.na(x)))

##           p2p    current_vote transfersshare donationssshare          gov
##           0           0           29           29           0
##        y2006        female    exminister    previous_vote media_mentions
##           0           0           0           0           2
##      quality    years_served
##           0           0

# for now, drop NAs

analysis <- analysis.cols %>%
  drop_na() %>%
  rename(zb = p2p) %>%
  rename(y = current_vote)

analysis.cols.large <- main %>% select(
  p2p,
  current_vote,
  transfersshare,
  donationssshare,
  gov,
  y2006,
  female,
  exminister,
  previous_vote,
  media_mentions,
  quality,
  years_served,
  # province, #prov if need numeric
  election,
  winner,
  pop_per_km2,
  immigrants,
  citizens,
  unemployment_rate,
  median_family_income
)

```

```
analysis.large <- analysis.cols.large %>%
  drop_na() %>%
  rename(zb = p2p) %>%
  rename(y = current_vote)
```

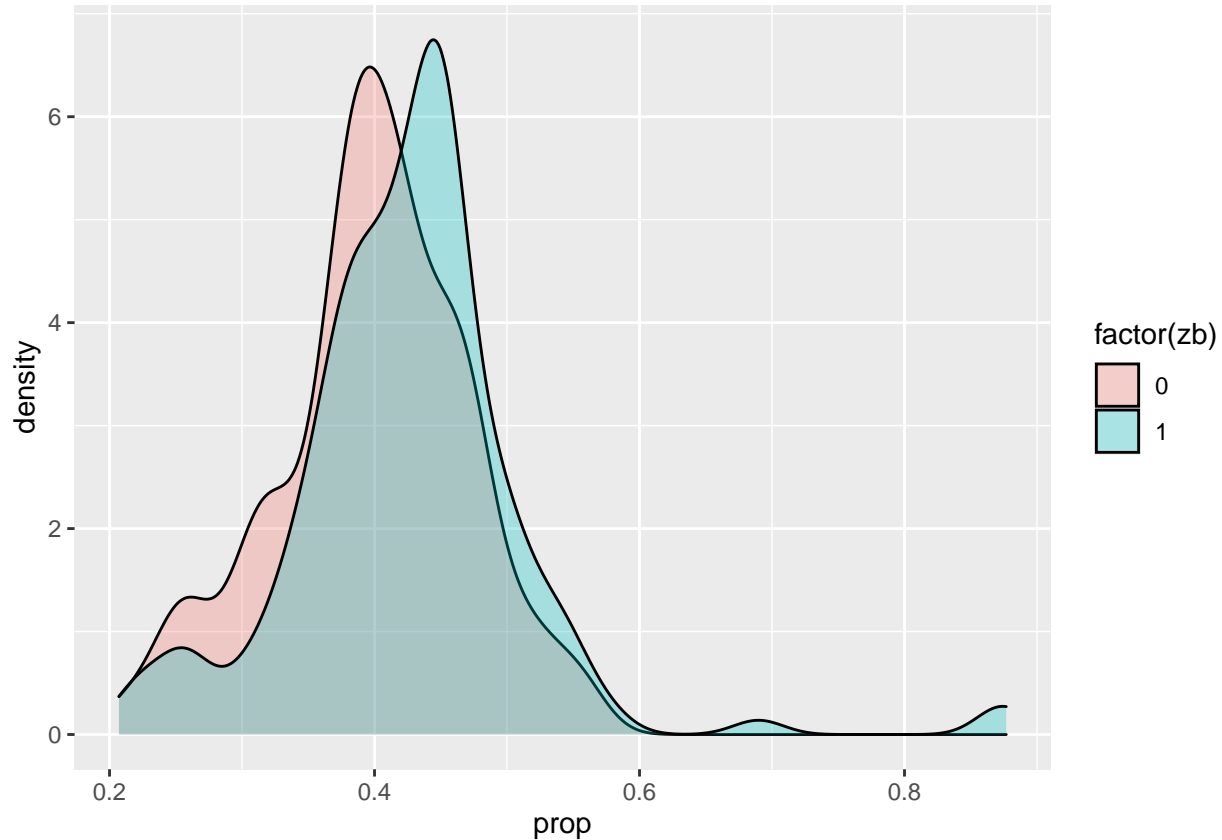
Note: matches stata output . gen SHARE=transfers+donations (29 missing values generated)

Propensity Scores

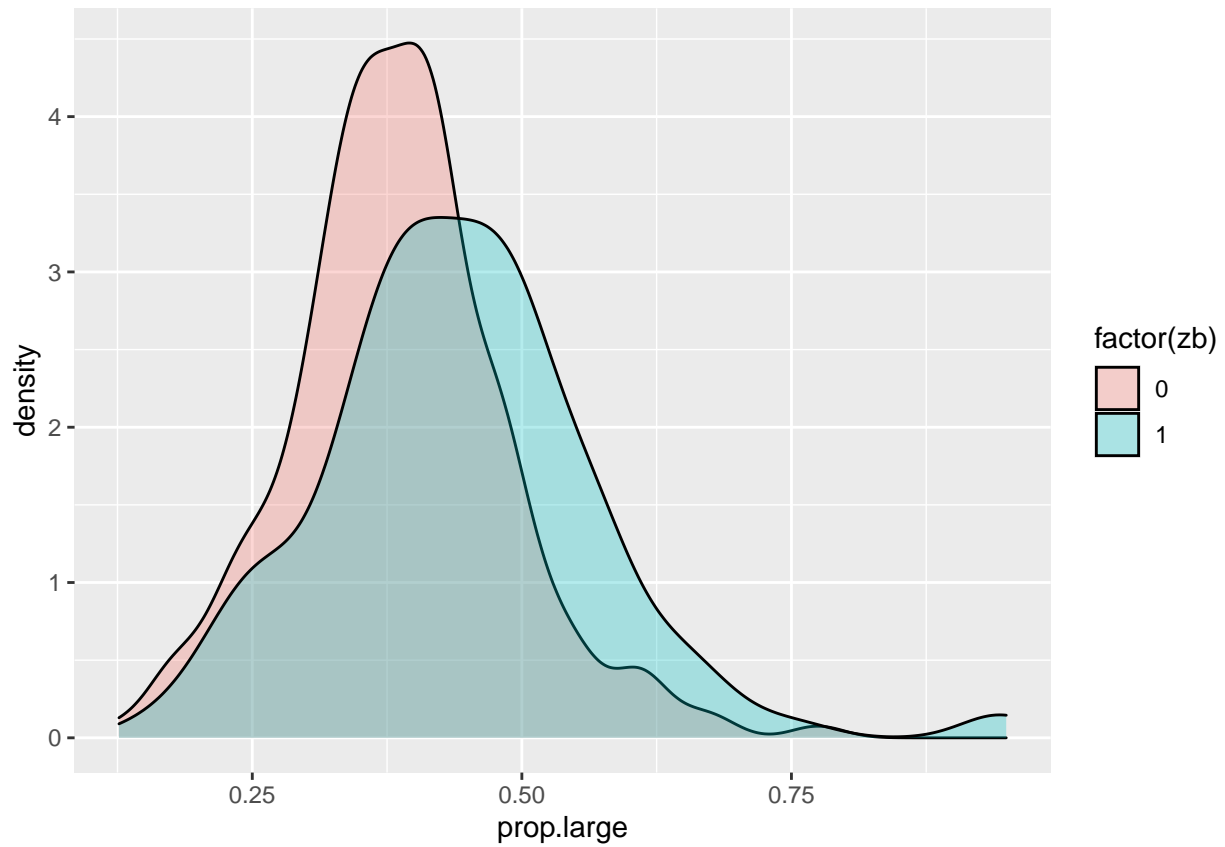
```
prop.scores <- glm(zb ~ . - y, family=binomial, data=analysis)
analysis$prop <- prop.scores$fitted.values
```

```
prop.scores.large <- glm(zb ~ . - y, family=binomial, data=analysis.large)
analysis.large$prop.large <- prop.scores.large$fitted.values
```

```
analysis %>% ggplot() +
  geom_density(aes(x=prop, fill=factor(zb)), alpha=0.3)
```

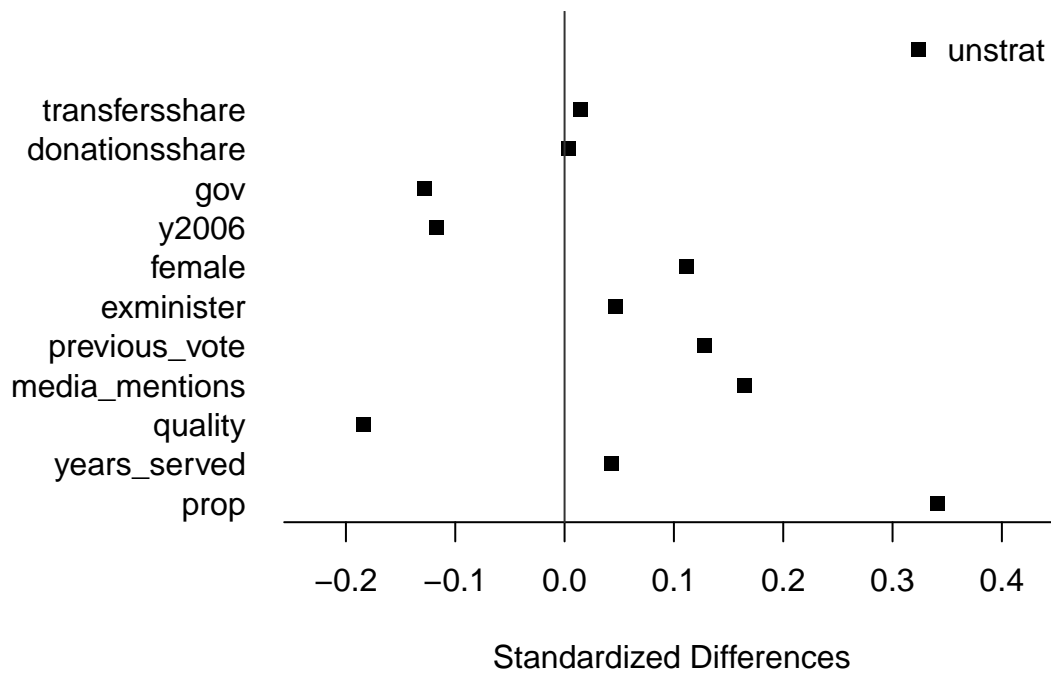


```
analysis.large %>% ggplot() +
  geom_density(aes(x=prop.large, fill=factor(zb)), alpha=0.3)
```

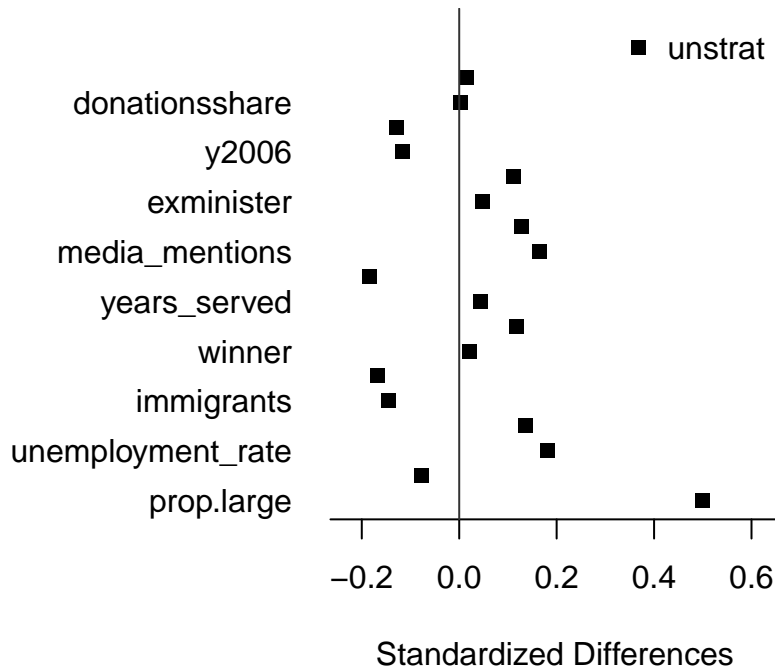


Matching

```
plot(xBalance(zb ~ . - 1 - y, data=analysis))
```

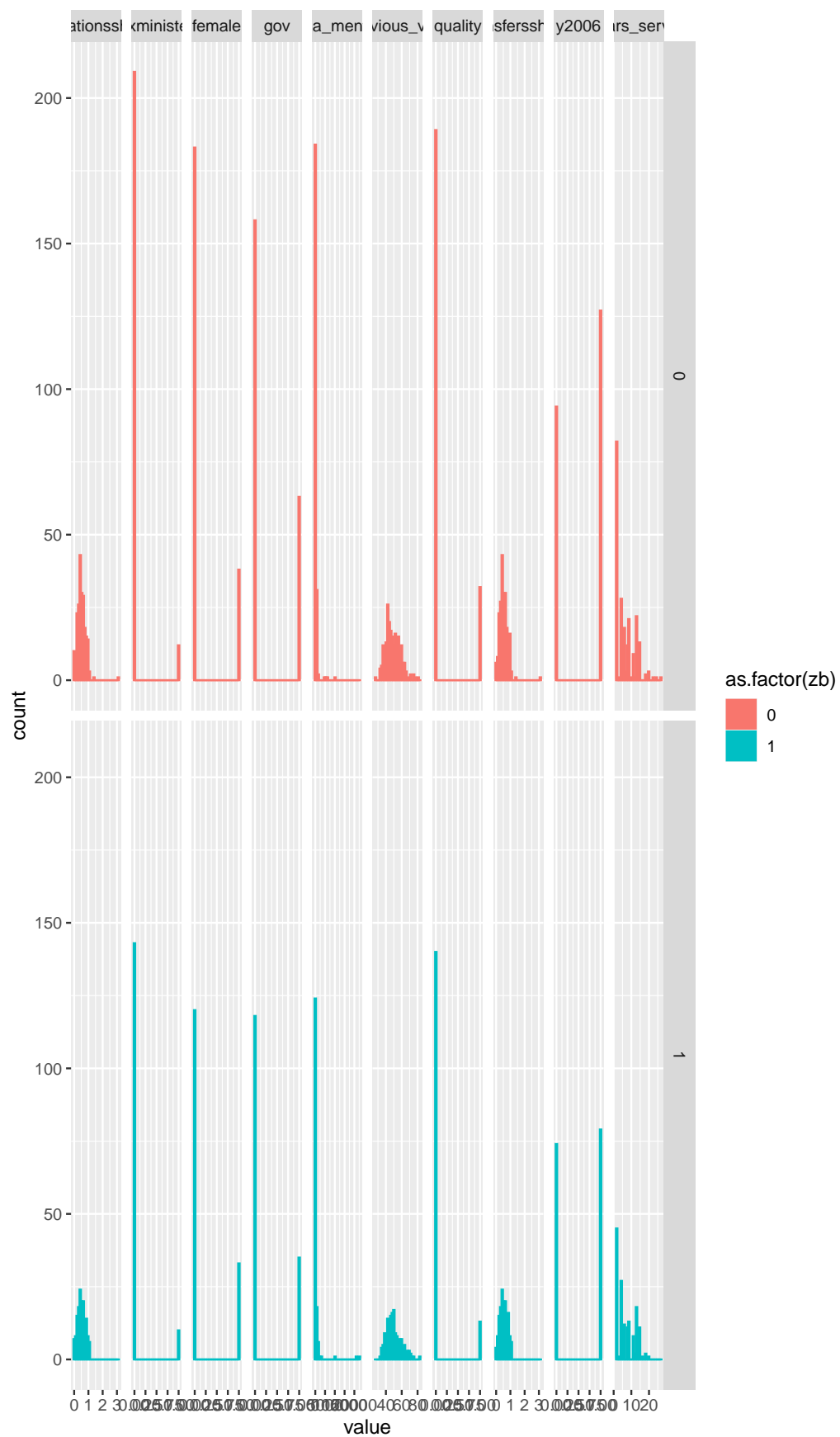


```
plot(xBalance(zb ~ . - 1 - y, data=analysis.large))
```

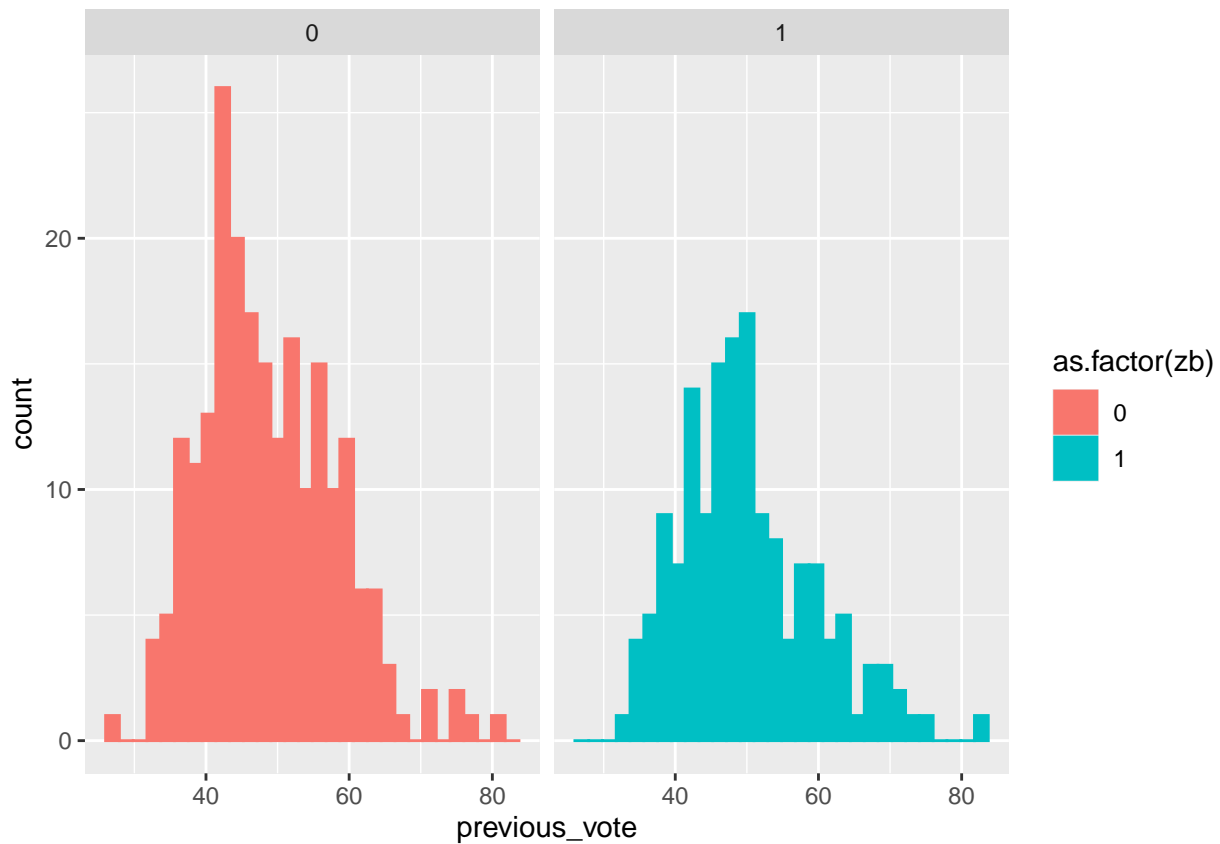


```
analysis %>%
  dplyr::select(zb, covariates) %>%
  pivot_longer(-zb, names_to="covariate", values_to="value") %>%
  ggplot(aes(x = value, color = as.factor(zb), fill = as.factor(zb))) +
  geom_histogram(bins=30) +
  facet_grid(cols=vars(covariate), rows=vars(zb), scales='free_x')
```

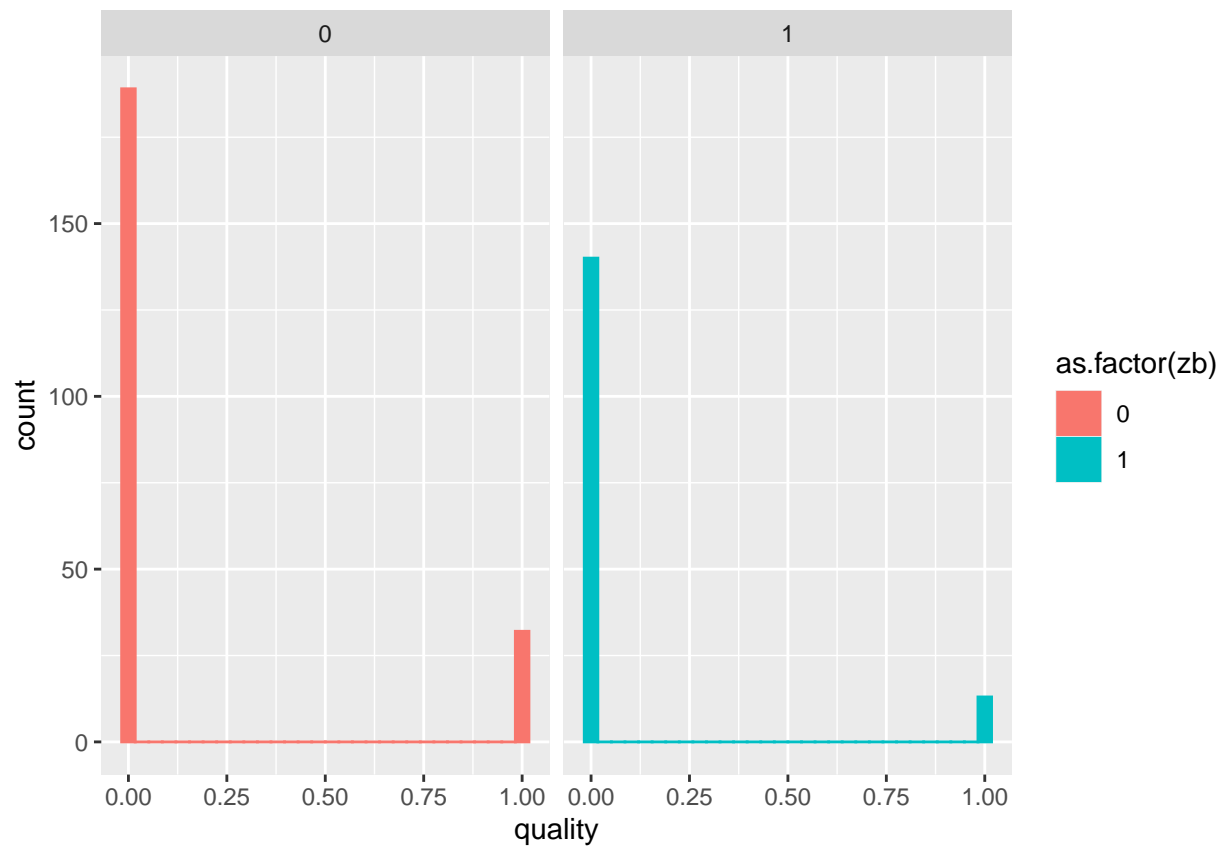
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(covariates)` instead of `covariates` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```



```
analysis.large %>%
  ggplot(aes(x = previous_vote, color = as.factor(zb), fill = as.factor(zb))) +
  geom_histogram(bins=30) +
  facet_wrap(~zb)
```



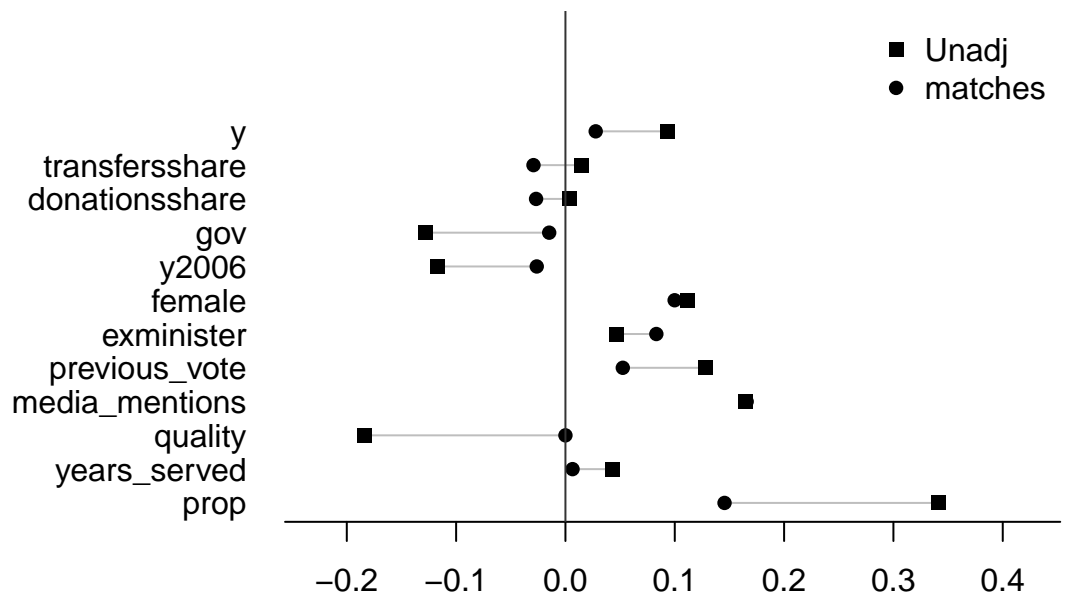
```
analysis %>%
  ggplot(aes(x = quality, color = as.factor(zb), fill = as.factor(zb))) +
  geom_histogram(bins=30) +
  facet_wrap(~zb)
```



```
z <- analysis$zb
X <- analysis %>% dplyr::select(-c(zb, prop, y))

distance <- smahal(z, X)
distance.cal <- addcaliper(distance, z=analysis$zb, p=analysis$prop, caliper=0.1)

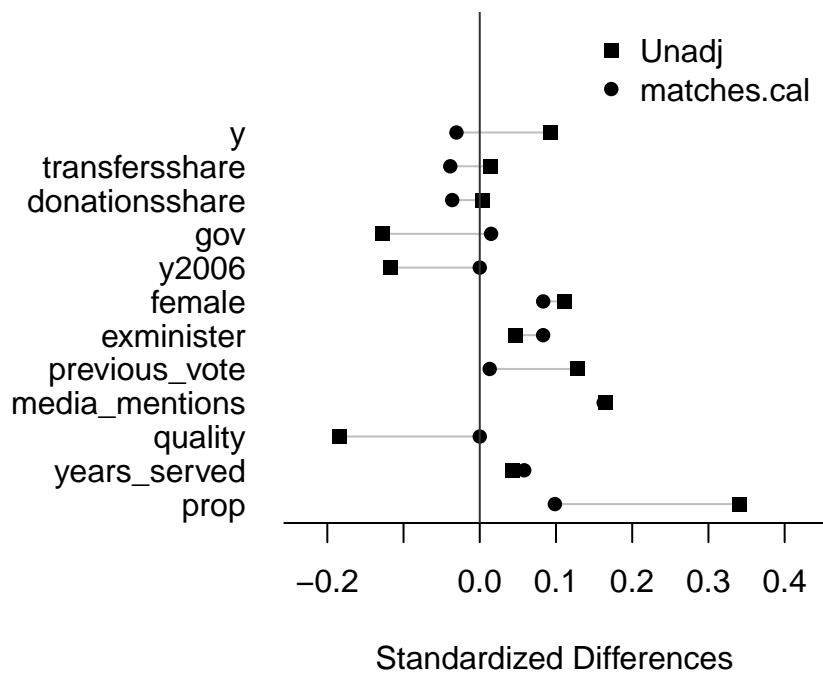
matches <- pairmatch(distance, data=analysis)
plot(xBalance(zb ~ . - 1 + strata(matches) , data=analysis))
```

1:1 Exact Matching

```
matches.cal <- pairmatch(distance.cal, data=analysis)
matches.df <- summarize.match(analysis, matches.cal)

plot(xBalance(zb ~ . + strata(matches.cal) - 1, data=analysis))
```



```
sum(is.na(matches))
```

```
## [1] 68
```

```
sum(!is.na(matches))
```

```
## [1] 306
```

```
sum(is.na(matches.cal))
```

```
## [1] 68
```

```
sum(!is.na(matches.cal))
```

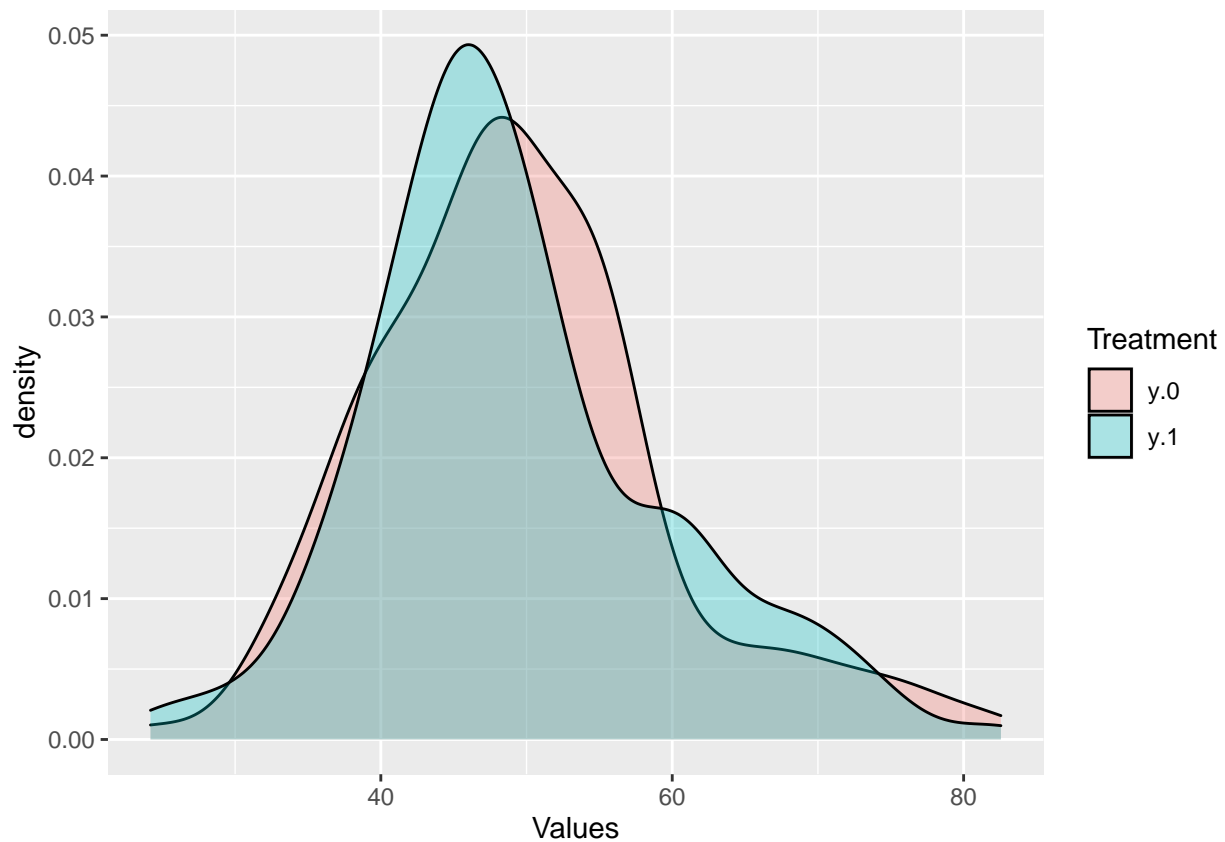
```
## [1] 306
```

1:3 Approximate Matching Not sure we have enough controls to do more than a 1:1 matching

FRT

In matched pairs, the vote share received looks nearly identical.

```
matches.df %>%  
  select(y.1, y.0) %>%  
  pivot_longer(c(y.1, y.0), names_to='Treatment', values_to='Values') %>%  
  ggplot() +  
  geom_density(aes(x=Values, fill=Treatment), alpha=0.3)
```



```
# DIM current assignment  
T.obs1 <- matches.df %>%  
  summarise(mean(y.1) - mean(y.0)) %>%  
  pull(.)
```

```
T.obs1 #T obs small and negative!
```

```
## [1] -0.3057513
```

```

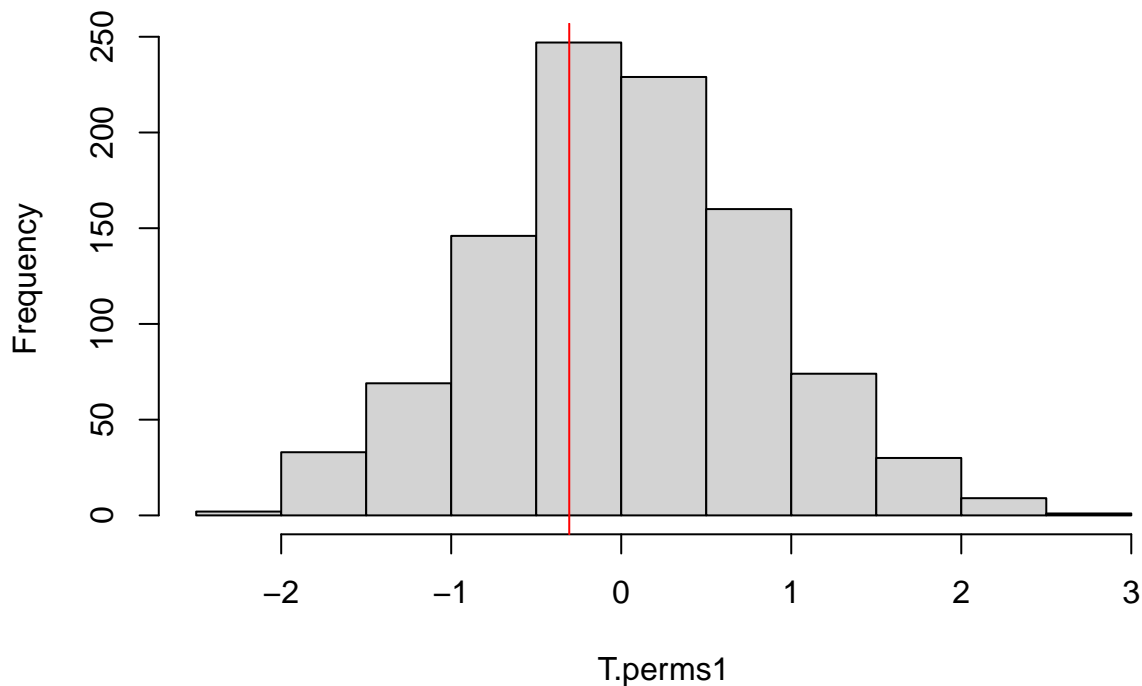
genPermute <- function(x, matches) {
  treated_unit <- sample(c(0,1), nrow(matches), replace=TRUE)
  matches %>%
    select(y.0, y.1) %>%
    mutate(treated=treated_unit) %>%
    mutate(treated_y = ifelse(treated == 1, y.1, y.0),
           control_y = ifelse(treated == 1, y.0, y.1)) %>%
    summarise(mean(treated_y) - mean(control_y)) %>%
    pull(.)
}

set.seed(123)
iters <- 1000
reps <- rep(NA, iters)

# Generate vector of test statistics under permutations
T.perms1 <- sapply(reps, function(x) genPermute(x, matches.df))
hist(T.perms1)
abline(v=T.obs1, col='red')

```

Histogram of T.perms1



```

pval <- (1/iters) * (sum(ifelse(T.perms1 >= T.obs1, 1, 0)) ) #calculate one sided p-value
pval

```

```
## [1] 0.652
```

```

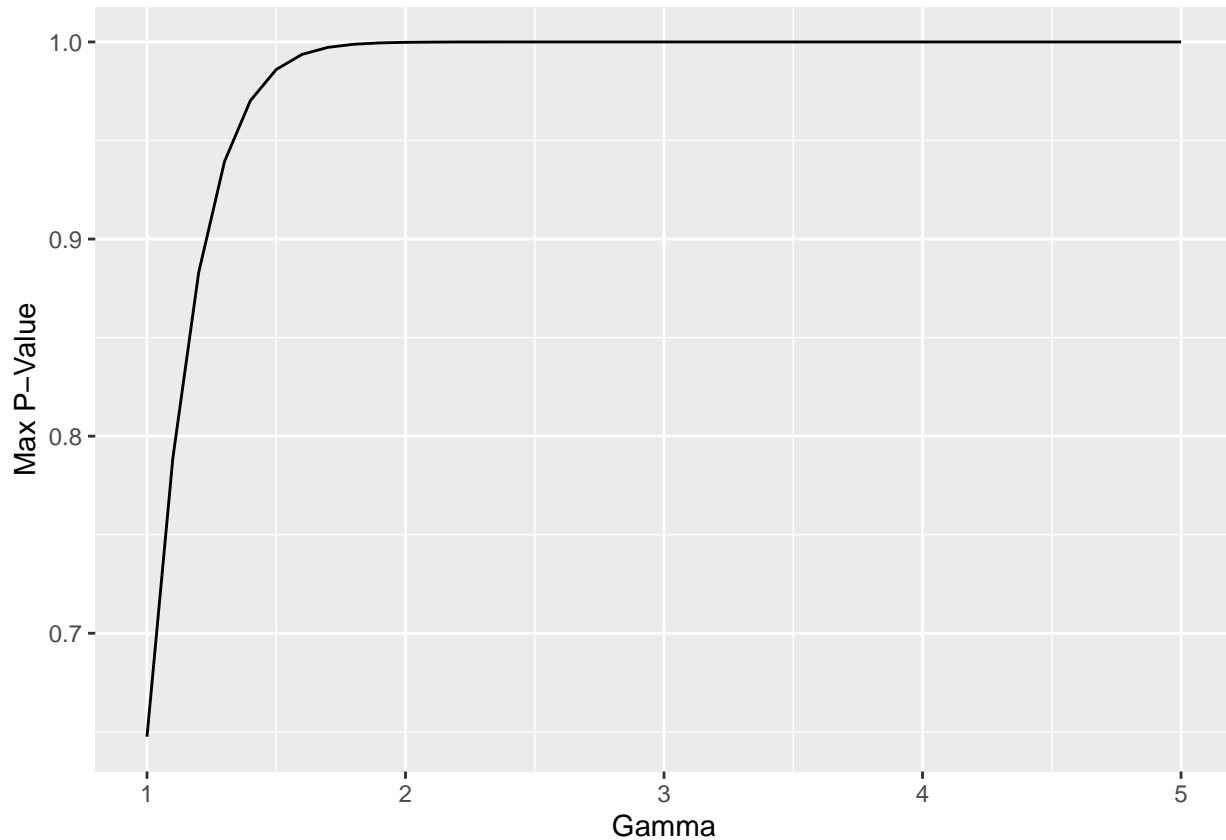
senm.data1 <- cast.senm(analysis, matches.cal)
sen.out <- sensitivitymult::senm(senm.data1$y, senm.data1$z, senm.data1$mset, gamma=1.2, trim = Inf, in

gamma.compute <- function(gamma) sensitivitymult::senm(senm.data1$y, senm.data1$z, senm.data1$mset, gam
grange = seq(1,5,by=0.1)

```

```
sensitivity <- sapply(grange, gamma.compute)

data.frame(gamma=grange, pvalue=sensitivity) %>%
  ggplot(aes(x=gamma, y=sensitivity)) +
  geom_line() +
  labs(x="Gamma", y="Max P-Value")
```



IPW Estimators

```
horvitz.thompson <- analysis %>%
  mutate(ZY1 = (zb * y)/prop) %>%
  mutate(ZY0 = ((1-zb) * y)/(1-prop)) %>%
  summarise(
    Units = n(),
    YBar1 = sum(ZY1) / sum(zb),
    YBar0 = sum(ZY0) / sum(1-zb),
    .groups='drop'
  ) %>%
  mutate(DIM=(YBar1 - YBar0)) %>%
  summarise(sum(DIM)) %>%
  pull(.)
```

```
horvitz.thompson
```

Horvitz Thompson

```
## [1] 37.20555
```

```
analysis.large %>%
  mutate(ZY1 = (zb * y)/prop.large) %>%
  mutate(ZY0 = ((1-zb) * y)/(1-prop.large)) %>%
  summarise(
    Units = n(),
    YBar1 = sum(ZY1) / sum(zb),
    YBar0 = sum(ZY0) / sum(1-zb),
    .groups='drop'
  ) %>%
  mutate(DIM=(YBar1 - YBar0)) %>%
  summarise(sum(DIM)) %>%
  pull(.)
```

```
## [1] 37.79287
```

```
hayek <- analysis %>%
  mutate(ZY1 = (zb * y)/prop) %>%
  mutate(ZY0 = ((1-zb) * y)/(1-prop)) %>%
  summarise(
    Units = n(),
    YBar1 = sum(ZY1) / sum(zb / prop),
    YBar0 = sum(ZY0) / sum((1-zb)/(1-prop)),
    .groups='drop'
  ) %>%
  mutate(DIM=(YBar1 - YBar0)) %>%
  summarise(sum(DIM)) %>%
  pull(.)
```

```
hayek
```

Hayek

```
## [1] -0.148164
```

Subclassification with Neymanian CI's

```
quant.vec <- quantile(analysis$prop, c(0.2, 0.4, 0.6, 0.8))
```

```
stratified <- analysis %>%
  mutate(stratum = case_when(
    prop < quant.vec[1] ~ 1,
    (quant.vec[1] <= prop) & (prop < quant.vec[2]) ~ 2,
    (quant.vec[2] <= prop) & (prop < quant.vec[3]) ~ 3,
    (quant.vec[3] <= prop) & (prop < quant.vec[4]) ~ 4,
    prop >= quant.vec[4] ~ 5
  ))
```

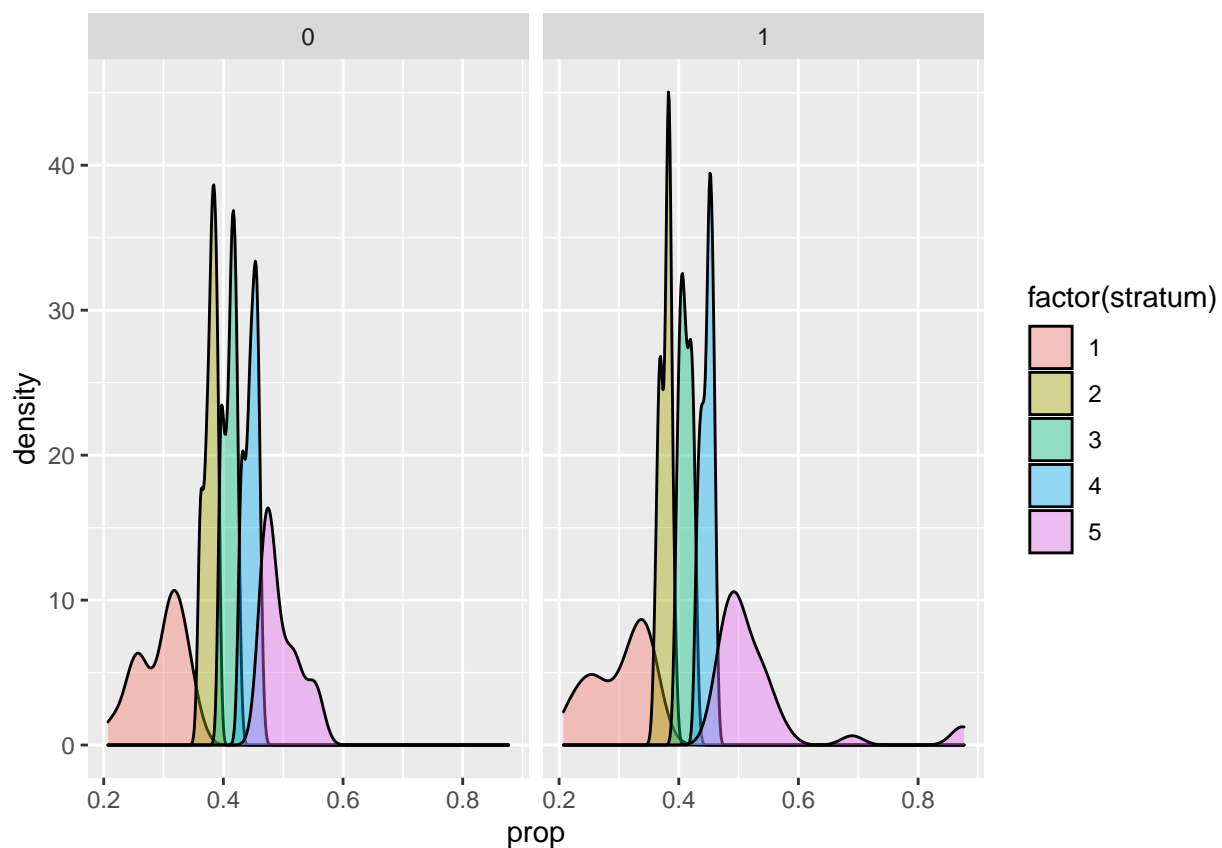
Number of units by stratum and treatment status

```
stratified %>%
  group_by(stratum) %>%
  summarise(
    treated = sum(zb),
    control = sum(1-zb),
```

```
.groups='drop'
)
```

```
## # A tibble: 5 x 3
##   stratum treated control
##   <dbl>   <int>   <dbl>
## 1     1     25     50
## 2     2     25     50
## 3     3     25     49
## 4     4     44     31
## 5     5     34     41
```

```
stratified %>%
  ggplot() +
  geom_density(aes(x=prop, fill=factor(stratum)), alpha=0.4) +
  facet_wrap(vars(zb))
```



```
tau_k <- stratified %>%
  mutate(ZY1 = zb * y) %>%
  mutate(ZY0 = (1-zb) * y) %>%
  group_by(stratum) %>%
  summarise(
    Units = n(),
    YBar1 = sum(ZY1) / sum(zb),
    YBar0 = sum(ZY0) / sum(1-zb),
    .groups='drop'
  ) %>%
  mutate(DIM=(YBar1 - YBar0) * Units/nrow(stratified)) %>%
```

```
summarise(sum(DIM)) %>%
pull(.)
```

tau_k

```
## [1] 0.1906014
```

```
tau_k.var <- stratified %>%
  group_by(stratum, zb) %>%
  summarise(
    N = n(),
    Var = var(y),
    .groups='drop'
  ) %>%
  mutate(weighted_V = Var / N) %>%
  group_by(stratum) %>%
  summarise(
    stratum_var = sum(weighted_V) * (sum(N) / nrow(stratified))^2,
    .groups='drop'
  ) %>%
  summarise(sum(stratum_var)) %>%
  pull(.)
```

tau_k.var

```
## [1] 1.036314
```

Compute a 95% confidence interval

```
normalCI <- function(tau, variance) {
  c(tau - sqrt(variance)*qnorm(.975), tau + sqrt(variance)*qnorm(.975))
}
```

```
normalCI(tau_k, tau_k.var)
```

```
## [1] -1.804632  2.185835
```

Null result