Mini-Project 2

Sam Ellinwood, Haley McAtee & Sam Schneider

October 31, 2017

Loading the data

Loading packages

```r
library(tidyverse)
library(scales)
```

Create our own dataset from the FEC dataset

```r
#Merge candidates and contributions
cand_cont <- candidates %>%
  inner_join(contributions, by = "cand_id")

#Rename fec_id to cand_id
names(house_elections) <- c("cand_id", "state", "district", "incumbent", "candidate_name", "party", "pri

#Merge house_elections and my_candidates
my_candidates <- cand_cont %>%
  inner_join(house_elections, by = "cand_id")

#Select specific columns out of the my_candidates dataset
my_candidates <- my_candidates %>%
  select(1:6, transaction_amt, general_votes, ge_winner) %>%
  filter(transaction_amt > 0, general_votes > 0, cand_party_affiliation == "DEM" | cand_party_affiliati
```

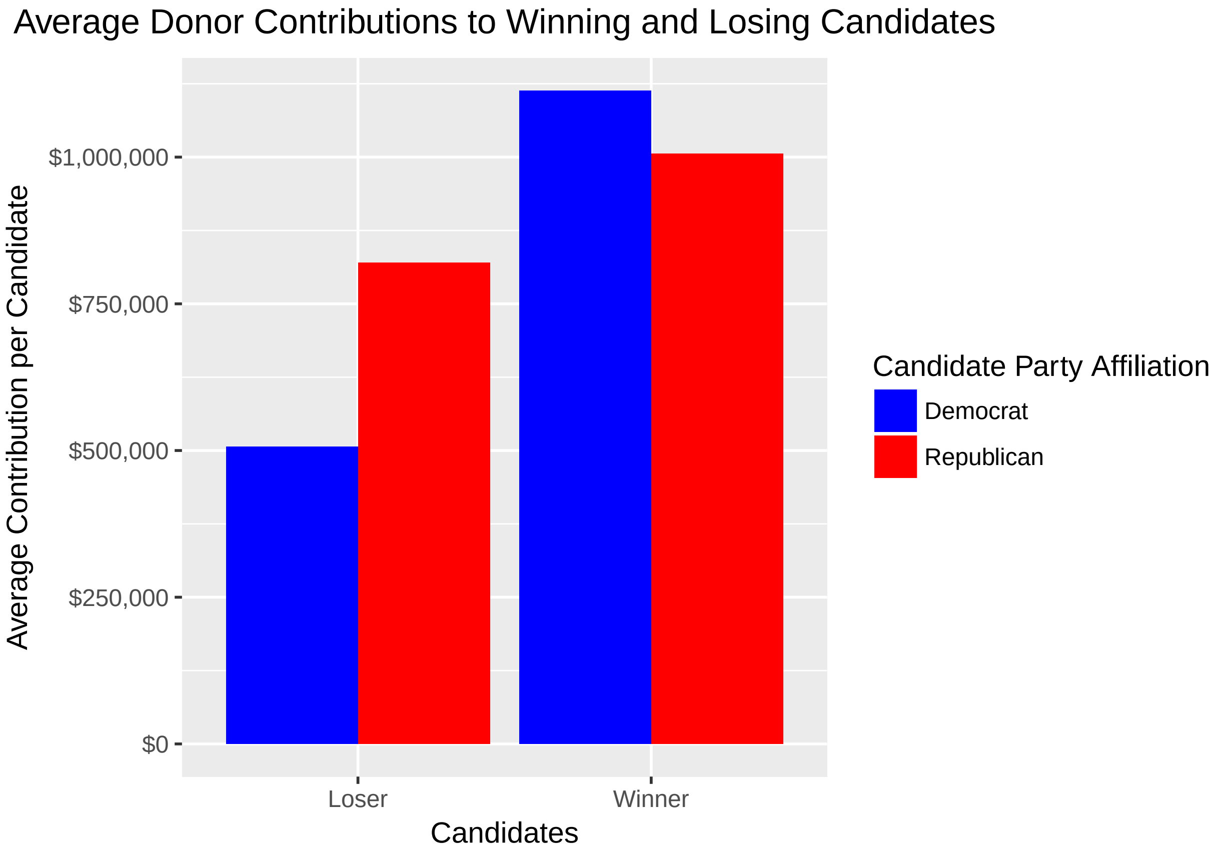
```r
#Storing republican winner and loser averages in rep_winner and rep_loser
rep_wl <- my_candidates %>%
    filter(cand_party_affiliation == "REP") %>%
    group_by(ge_winner) %>%
    summarize("total_for_winners" = sum(transaction_amt),
              "num_winners" = n_distinct(cand_id),
               "winner_avg" = total_for_winners/num_winners)

rep_winner <- rep_wl$winner_avg[[2]]
rep_loser <- rep_wl$winner_avg[[1]]

#Storing democarat winner and loser averages in dem_winner and dem_loser
dem_wl <- my_candidates %>%
    filter(cand_party_affiliation == "DEM") %>%
    group_by(ge_winner) %>%
    summarize("total_for_winners" = sum(transaction_amt),
              "num_winners" = n_distinct(cand_id),
               "winner_avg" = total_for_winners/num_winners)
```

```
dem_winner <- dem_wl$winner_avg[[2]]
dem_loser <- dem_wl$winner_avg[[1]]
```

```
#Add a winner_avg variable to my_candidates
my_candidates <- my_candidates %>%
  mutate("winner_avg" = ifelse(ge_winner == "W" & cand_party_affiliation == "DEM", dem_winner,
                        ifelse(ge_winner == "N" & cand_party_affiliation == "DEM", dem_loser,
                        ifelse(ge_winner == "W" & cand_party_affiliation == "REP", rep_winner,
                        rep_loser))))
```

Creating the data graphic

```
#Creating our plot
ggplot(my_candidates, aes(x = ge_winner,
                          y = winner_avg,
                          fill = cand_party_affiliation)) +
  scale_fill_manual(values = c("blue", "red"), labels = c("Democrat", "Republican")) +
  stat_summary(fun.y = "mean", geom = "bar", position = position_dodge()) +
  scale_y_continuous(breaks = c(0, 250000, 500000, 750000, 1000000), labels = dollar) +
  scale_x_discrete(breaks=c("N", "W"),labels=c("Loser", "Winner")) +
  ggtitle("Average Donor Contributions to Winning and Losing Candidates") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(y = "Average Contribution per Candidate",
       x = "Candidates",
       fill = "Candidate Party Affiliation")
```



Function to create rep/dem winner & loser variables

```r
#Attempting to write a function that creates the rep/dem winner and loser variables
get_avg <- function(party_arg, party_winner_arg, party_loser_arg) {
  party_winners <- my_candidates %>%
    filter(cand_party_affiliation == party_arg) %>%
    group_by(ge_winner) %>%
    summarize("total_for_winners" = sum(transaction_amt),
              "num_winners" = n_distinct(cand_id),
               "winner_avg" = total_for_winners/num_winners)

party_winner_arg <- party_winners$winner_avg[[2]]
party_loser_arg <- party_winners$winner_avg[[1]]

print(party_winner_arg)
print(party_loser_arg)
}

get_avg(party_arg = "REP", party_winner_arg = "rep_winner", party_loser_arg = "rep_loser")
```

```
## [1] 1006103
## [1] 820044.8
```