

Ethan Feldman Coding Sample

Ethan Feldman

1/6/2022

#Introduction This is a sample of some of the code I am using to analyze and visualize data for my undergraduate thesis. The thesis examines the impacts of a campaign finance law in Connecticut, the Citizen's Election Program (CEP). The data comes from the National Institute for Money in Politics, which in turn scraped the data from various state data sources, meaning all data used is public.

```
#load in packages
library(tidyverse)
```

```
## — Attaching packages — tidyverse 1.3.1 —
```

```
## ✓ ggplot2 3.3.5      ✓ purrr   0.3.4
## ✓ tibble  3.1.4      ✓ dplyr   1.0.7
## ✓ tidyr   1.1.3      ✓ stringr 1.4.0
## ✓ readr   2.0.1      ✓ forcats 0.5.1
```

```
## — Conflicts — tidyverse_conflicts() —
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggalt)
```

```
## Registered S3 methods overwritten by 'ggalt':
##   method                from
##   grid.draw.absoluteGrob ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
#read in data and select columns of interest
setwd("/Users/ethanfeldman/OneDrive - Emory University/Thesis")
data<-read.csv("Thesis_Data.csv")

data<-data%>%
  select(Candidate.id,Candidate,Election_Status,General_Party,Election_Jurisdiction,
         Election_Year.token,
         Election_Year,Incumbency_Status,Contributor.id,Contributor,Type_of_Contributor,
         Amount,Date,Type_of_Transaction,In.State,Office_Sought)%>%
  filter(Amount>0)
```

```
#create a new variable that classifies each donation as either small/large individual or other
data<-data%>%
  mutate(class=case_when(
    Type_of_Contributor=="Individual"&Amount<101 & Election_Year<2017~"Small.Ind",
    Type_of_Contributor=="Individual"&Amount>100 & Election_Year<2017~"Large.Ind",
    Type_of_Contributor=="Individual"&Amount<251 & Election_Year==2018~"Small.Ind",
    Type_of_Contributor=="Individual"&Amount>250 & Election_Year==2018~"Large.Ind",
    Type_of_Contributor=="Individual"&Amount<271 & Election_Year==2020~"Small.Ind",
    Type_of_Contributor=="Individual"&Amount>270 & Election_Year==2020~"Large.Ind",
    Type_of_Contributor=="Other"&Contributor=="UNITEMIZED DONATIONS"~"Small.Ind",
    TRUE~"Other")
  )%>%
  mutate(Chamber=str_extract(string = Office_Sought, pattern = ".?* "))
```

```
#change candidate data type
data$Candidate<-as.factor(data$Candidate)
```

#Plot 1 In this section, I plot how the CEP has changed the composition of donations in Connecticut

```
#Beginning of plot 1.
#Create dataframe with the amount each candidate raised broken down by type.
#Cut out everything before 2004 because no complete RI data before that. Cut out grants from the CEP.
Candidate.Money.Perc<-data%>%
  filter(Election_Year>=2004,Contributor!="PUBLIC FUND")%>%
  group_by(Election_Year,Candidate,class,Election_Jurisdiction)%>%
  summarise(Total=sum(Amount))%>%
  pivot_wider(names_from=class,values_from=Total)%>%
  replace_na(list(Small.Ind=0,Large.Ind=0,Other=0))%>%
  mutate(perc.Large=100*(Large.Ind/(Large.Ind+Small.Ind+Other)),
         perc.Small=100*(Small.Ind/(Large.Ind+Small.Ind+Other)),
         perc.Other=100*(Other/(Large.Ind+Small.Ind+Other)))
```

```
## `summarise()` has grouped output by 'Election_Year', 'Candidate', 'class'. You can override using the `.groups` argument.
```

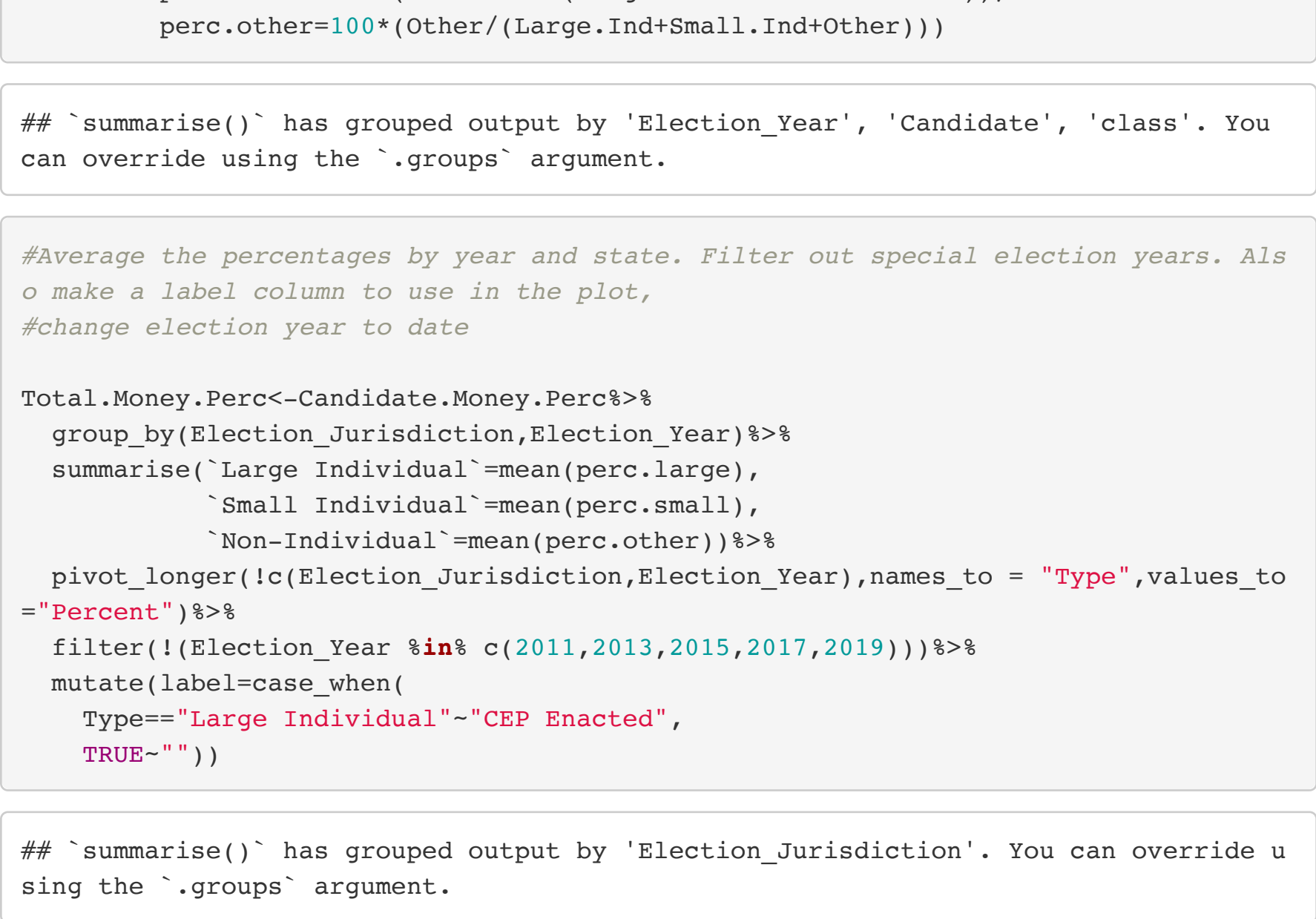
```
#Average the percentages by year and state. Filter out special election years. Also make a label column to use in the plot,
#change election year to date
```

```
Total.Money.Perc<-Candidate.Money.Perc%>%
  group_by(Election_Jurisdiction,Election_Year)%>%
  summarise(`Large Individual`=mean(perc.Large),
            `Small Individual`=mean(perc.Small),
            `Non-Individual`=mean(perc.Other))%>%
  pivot_longer(!c(Election_Jurisdiction,Election_Year),names_to = "Type",values_to = "Percent")%>%
  filter(!(Election_Year %in% c(2011,2013,2015,2017,2019)))%>%
  mutate(label=case_when(
    Type=="Large Individual"~"CEP Enacted",
    TRUE~""))
```

```
## `summarise()` has grouped output by 'Election_Jurisdiction'. You can override using the `.groups` argument.
```

```
Total.Money.Perc$Election_Year<-ymd(paste0((Total.Money.Perc$Election_Year),"0101"))
```

```
#Plot average donor composition by state, year
Total.Money.Perc%>%
  ggplot(aes(Election_Year,Percent,color=Election_Jurisdiction))+
  geom_point()+
  geom_line()+
  facet_wrap(~Type)+
  geom_vline(xintercept=ymd(20070101))+
  labs(x="Election Year",y="Average Percent of Total Donations",color="State",title="Average Percentage of Total Donations by Type and State Over Time")+
  geom_text(aes(x = ymd(20060601), y = 80, label = label),color="black",size=3,angle=90)+
  theme_bw()+
  theme(axis.text.x = element_text(size=6),plot.title = element_text(size=12))
```



#Plot 2 In this section, I create a dumbbell plot showing how much money candidates raised from small dollar donations in 2006 and 2008.

```
#Beginning of plot 2
#get the amount from small donors for each candidate in CT in 2006 and 2008
Small.Donors.2006<-data%>%
  filter(Election_Year==2006,Election_Jurisdiction=="CT")%>%
  mutate(Limit=case_when(
    Chamber=="HOUSE " ~5000,
    Chamber=="SENATE " ~15000,
    TRUE~0,
  ))%>%
  group_by(Election_Year,Candidate,class,Chamber,Limit)%>%
  summarise(Total=sum(Amount))%>%
  filter(class=="Small.Ind")
```

```
## `summarise()` has grouped output by 'Election_Year', 'Candidate', 'class', 'Chamber'. You can override using the `.groups` argument.
```

```
Small.Donors.2008<-data%>%
  filter(Election_Year==2008,Election_Jurisdiction=="CT")%>%
  mutate(Limit=case_when(
    Chamber=="HOUSE " ~5000,
    Chamber=="SENATE " ~15000,
    TRUE~0,
  ))%>%
  group_by(Election_Year,Candidate,class,Chamber,Limit)%>%
  summarise(Total=sum(Amount))%>%
  filter(class=="Small.Ind")
```

```
## `summarise()` has grouped output by 'Election_Year', 'Candidate', 'class', 'Chamber'. You can override using the `.groups` argument.
```

```
#Join the two datasets so we have the candidates who ran in both so we can compare them
Dumbbell<-Small.Donors.2006%>%
  inner_join(Small.Donors.2008,by=c("Candidate","class"))
```

```
#Create plot
ggplot()+
  geom_dumbbell(data=Dumbbell,aes(x = Total.x, xend = Total.y,
                                  y=fct_reorder(Candidate>Total.x), size=.3,colour_x = "blue", colour_xend = "red",size_x=2, size_xend = 2) +
  facet_wrap(~Chamber,x,scales="free")+
  geom_vline(data=Dumbbell,aes(xintercept =Limit.x),size=1))+
  labs(x="Amount Raised From Small Donors",y="Candidate",title="Amount Candidates Raise from Small Donors, 2006 vs 2008")+
  theme_bw()+
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank())+
  geom_text(data=filter(Dumbbell, Candidate=="HARRIS, JONATHAN A"),
            aes(x=Total.x, y=Candidate, label="2006"),size=3,hjust=-.2,color="blue") +
  geom_text(data=filter(Dumbbell, Candidate=="HARRIS, JONATHAN A"),
            aes(x=Total.y, y=Candidate, label="2008"),size=3,hjust=1.2,color="red")
  )
```

Amount Candidates Raise from Small Donors, 2006 vs 2008

