파이썬프로그래밍
최종보고서

2017156034 전상민

OVERVIEW

1. 조사부문

i. **12 년, 16 년 미국 민주당, 공화당의 분포 및 추이**
ii. **1976 년~2016 년도의 정당 지지율 변화**
iii. **인구조사와 비교한 정당 지지율 비교**

2. 필요자료

- country_facts.csv, country_facts_dictionary.csv
  2016 년 미국 인구조사
- president-1976-2016.csv
  76~16 년 미국 대통령 선거 결과
- us-2016-primary-results.csv
  16 년 미국 대통령 선거 결과
- https://en.wikipedia.org/wiki/2012_United_States_presidential_election#Results_by_state
  2012 년 주별 투표 결과
- https://en.wikipedia.org/wiki/2016_United_States_presidential_election#Results_by_state
  2016 년 주별 투표 결과

3. 차례

i. **필요자료:** 필요 자료 구성을 보인다.
ii. **데이터 가공 계획:** 필요 자료들을 어떻게 가공할 것인지에 대해 설명한다.
iii. **코드 분석:** 데이터 가공 계획대로 코드를 구성한다.
iv. **결과 의미 분석:** 구성된 코드대로 나온 결과의 구성을 설명한다.
v. **결과 분석:** 얻은 결과가 의미하는 바가 무엇인지 찾아본다.

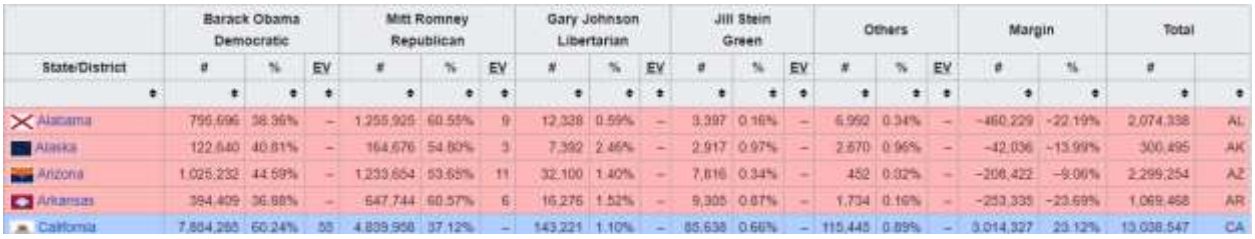
PROJECTS

1. 12 년, 16 년 미국 민주당, 공화당의 분포 및 추이

i. 필요자료

- https://en.wikipedia.org/wiki/2012_United_States_presidential_election#Results_by_state

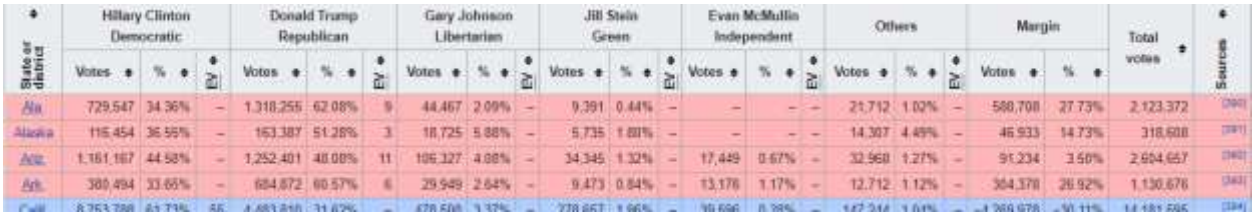
  2012 년 주별 투표 결과



State/District	Barack Obama Democratic			Mitt Romney Republican			Gary Johnson Libertarian			Jill Stein Green			Others			Margin		Total	
	#	%	EV	#	%	EV	#	%	EV	#	%	EV	#	%	EV	#	%	#	
Alabama	795,696	38.36%	–	1,255,925	60.55%	9	12,338	0.59%	–	3,397	0.16%	–	6,992	0.34%	–	−460,229	−22.19%	2,074,338	AL
Alaska	122,640	40.81%	–	164,676	54.80%	3	7,392	2.46%	–	2,917	0.97%	–	2,670	0.96%	–	−42,036	−13.99%	300,495	AK
Arizona	1,025,232	44.59%	–	1,233,654	53.68%	11	32,100	1.40%	–	7,816	0.34%	–	452	0.02%	–	−208,422	−9.06%	2,299,254	AZ
Arkansas	394,409	36.88%	–	647,744	60.57%	6	16,276	1.52%	–	9,305	0.87%	–	1,734	0.16%	–	−253,335	−23.69%	1,069,468	AR
California	7,854,285	60.24%	55	4,839,958	37.12%	–	143,221	1.10%	–	85,638	0.66%	–	115,445	0.89%	–	3,014,327	23.12%	13,038,547	CA

- https://en.wikipedia.org/wiki/2016_United_States_presidential_election#Results_by_state

  2016 년 주별 투표 결과



State or district	Hillary Clinton Democratic			Donald Trump Republican			Gary Johnson Libertarian			Jill Stein Green			Evan McMullin Independent			Others			Margin		Total votes	Source
	Votes	%	EV	Votes	%	EV	Votes	%	EV	Votes	%	EV	Votes	%	EV	Votes	%	EV	Votes	%		
Ala.	729,547	34.36%	–	1,318,255	62.08%	9	44,467	2.09%	–	9,391	0.44%	–	–	–	–	21,712	1.02%	–	588,708	27.73%	2,123,372	
Alaska	116,454	36.55%	–	163,387	51.28%	3	18,725	5.88%	–	5,735	1.80%	–	–	–	–	14,307	4.49%	–	46,933	14.73%	318,608	
Ariz.	1,161,167	44.58%	–	1,252,401	48.08%	11	106,327	4.08%	–	34,345	1.32%	–	17,449	0.67%	–	32,960	1.27%	–	91,234	3.50%	2,604,657	
Ark.	380,494	33.65%	–	684,872	60.57%	6	29,949	2.64%	–	9,473	0.84%	–	13,176	1.17%	–	12,712	1.12%	–	304,378	26.92%	1,130,676	
Calif.	8,753,788	61.73%	55	4,483,810	31.62%	–	478,500	3.37%	–	278,657	1.96%	–	30,696	0.22%	–	147,344	1.04%	–	4,269,978	30.11%	14,181,595	

ii. 데이터 가공 계획

'2012 년 주별 투표 결과'를 보면 민주당, 공화당 이외의 정당들도 고려한 통계임을 볼 수 있다. 본인은 가장 투표를 가장 많이 받은 1,2 위 정당인 민주당과 공화당만을 비교를 할 계획이다. 2016 년도 마찬가지이다.

1. **민주당%, 공화당% 비교**
   정당 투표수를 전체 투표수로 나누어 백분율을 구하고 민주당에서 공화당 백분율을 빼어, 공화당에 비해 민주당을 지지하는 인구가 더 많은 주, 그리고 그 반대의 주를 보인다.
2. **12 년~16 년 추이 비교**
   12 년 투표수와 16 년도 투표수를 빼어 추이를 분석하고, 가장 변화가 큰 주를 보인다.

iii. 코드 분석

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  #2012 United States presidential election votes Dataset
5  df = pd.read_html('https://en.wikipedia.org/wiki/2012_United_States_presidentia
6  res12=pd.DataFrame(df[21])
7  res12.index_col=0
8  res12.replace('-', 0, inplace=True)
9  for k in range(1,14,3):
10    res12.iloc[:,k] = pd.to_numeric(res12.iloc[:,k], errors='coerce')
11    res12.iloc[:,k+2] = pd.to_numeric(res12.iloc[:,k+2], errors='coerce')
12 res12.iloc[:,0]=res12.iloc[:,0].str.replace('★','')
13 res12.iloc[30,0]=res12.iloc[30,0].replace('[115]','')
14 res12.iloc[32,0]=res12.iloc[32,0].replace('[116]','')
15 res12.iloc[35,0]=res12.iloc[35,0].replace('[117]','')
16 res12.iloc[49,0]=res12.iloc[49,0].replace('[118]','')
17 res12.rename(columns={res12.columns[0][0]:res12.columns[0][1]}, inplace=True)
```

1. 먼저 12 년도 데이터 정제를 한다.
    - 인덱스를 지정
    - '-'으로 표현된 null 값 정제
    - String 데이터형을 int 형으로 형 변환
    - 특수문자 제거
    - 이름 재변경

```
24 res16 = pd.read_csv('res16.csv',header=[0,1])
25 res16=res16.drop(57)
26 res16=res16.drop(columns=['Sources'])
27
28 res16.replace('-', 0, inplace=True)
29 for k in range(1,17,3):
30    res16.iloc[:,k] = res16.iloc[:,k].str.replace(',', '')
31    res16.iloc[:,k] = pd.to_numeric(res16.iloc[:,k], errors='coerce')
32    res16.iloc[:,k+2] = pd.to_numeric(res16.iloc[:,k+2], errors='coerce')
33 res16.iloc[:,21] = res16.iloc[:,21].str.replace(',', '')
34 res16.iloc[:,21] = pd.to_numeric(res16.iloc[:,21])
```

2. 16 년도 테이블은 12 년도 테이블과 구조가 다르다.
    - '-'으로 표현된 null 값 정제
    - String 데이터형을 int 형으로 형 변환
    - 특수문자 제거

```
#Votes between Barack Obama and Mitt Romney
DR12 = pd.DataFrame(res12.iloc[:, [0,1,4,18]])
DR12.columns=DR12.columns.droplevel(1)

DR12['D%']=DR12.iloc[:,1]/DR12.iloc[:,3]
DR12['R%']=DR12.iloc[:,2]/DR12.iloc[:,3]
DR12['D-R%']=DR12.iloc[:,4]-DR12.iloc[:,5]
```

3. 12 년도의 민주당과 공화당 백분율과 백분율 차
    - 투표 수를 전체 투표수로 나누어 백분율을 구함
    - 구한 백분율을 빼어 백분율 차를 구함

```
#Votes between Donald Trump and Hilary Clinton
DR16 = pd.DataFrame(res16.iloc[:, [0,1,4,21]])
DR16.columns=DR16.columns.droplevel(1)
DR16.drop([20,21,30,31,32], inplace=True)
DR16.reset_index(drop=True, inplace=True)

DR16['D%']=DR16.iloc[:,1]/DR16.iloc[:,3]
DR16['R%']=DR16.iloc[:,2]/DR16.iloc[:,3]
DR16['D-R%']=DR16.iloc[:,4]-DR16.iloc[:,5]
```

4. 16 년도의 민주당과 공화당 백분율과 백분율 차
    - 구조가 달라 과정에 차이가 있다.

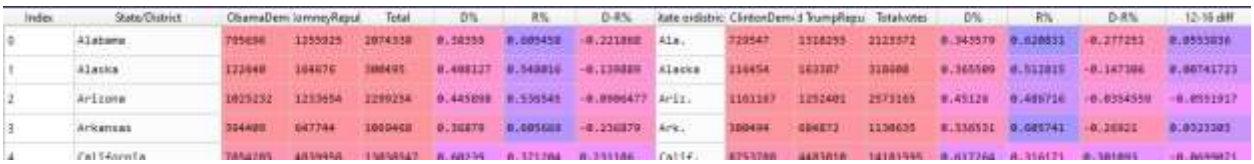
Nebr. †	284,494	33.70%	–
NE-1	100,132	35.46%	–
NE-2	131,030	44.92%	–
NE-3	53,332	19.73%	–

    - 세분화된 주별 투표 데이터가 있어 drop 하여 추가로 정제를 한다.
    - 그에 따라 index 도 리셋 해준다.

```
#joining DR12,DR16
DR=pd.concat([DR12,DR16],axis=1)
DR['12-16 diff']=DR.iloc[:,6]-DR.iloc[:,13]
```

5. 12 년도와 16 년도 테이블을 join 한다.
6. 그리고 두 년도의 차를 구한다.

- 병합된 테이블



```
59  #conditional bar graph coloring
60  colors_DR12 = []
61  for index, row in DR12.iterrows():
62      if row['D-R%']>0:
63          colors_DR12.append('b')
64      else:
65          colors_DR12.append('r')
66
67  colors_DR16 = []
68  for index, row in DR16.iterrows():
69      if row['D-R%'] > 0:
70          colors_DR16.append('b')
71      else:
72          colors_DR16.append('r')
```

7. 표현할 막대그래프의 색을 조건별로 입힌다.
    - 값이 양수이면 민주당을 나타내는 파란색, 음수이면 공화당을 나타내는 빨간색으로 설정한다.

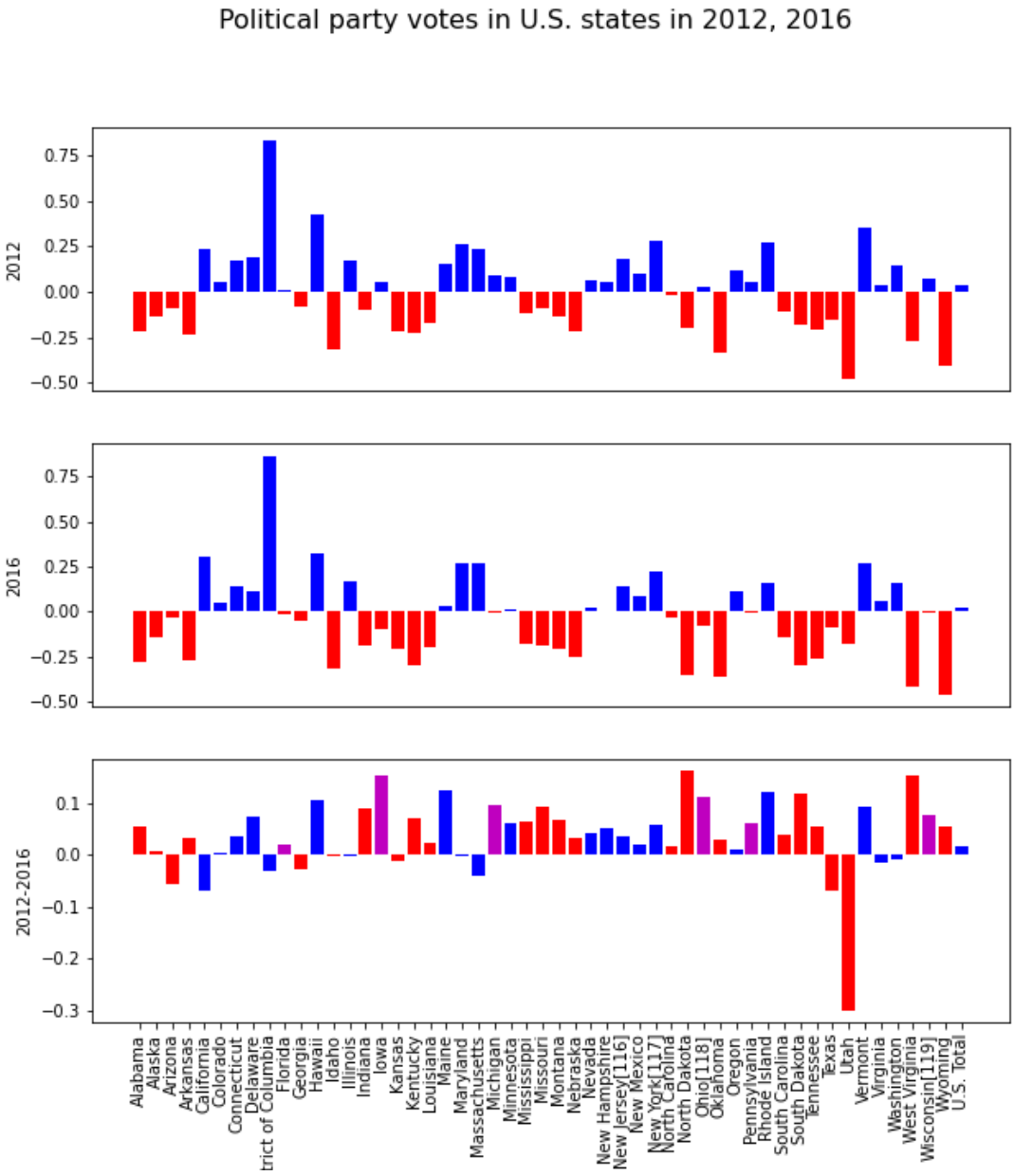
```
74  colors_DR=[]
75  for i in range(len(colors_DR12)):
76      if colors_DR12[i] == colors_DR16[i]:
77          colors_DR.append(colors_DR16[i])
78      else:
79          if colors_DR12[i] > colors_DR16[i]:
80              colors_DR.append('c')
81          else:
82              colors_DR.append('m')
```

    - 추이 데이터는 그 위에 덧붙여, 민주당에서 공화당으로 바뀌었으면 자홍색, 공화당에서 민주당으로 바뀌었으면 청록색으로 설정하였다.

```
84  #plt
85  plt.figure(figsize=(10,10))
86  plt.suptitle("Political party votes in U.S. states in 2012, 2016",
87               fontsize=16)
88
89  plt.subplot(3,1,1)
90  plt.xticks([])
91  plt.ylabel("2012")
92  plt.bar(DR12['State/District'],DR12['D-R%'], color=colors_DR12)
93
94  plt.subplot(3,1,2)
95  plt.xticks([])
96  plt.ylabel("2016")
97  plt.bar(DR12['State/District'],DR16['D-R%'], color=colors_DR16)
98
99  plt.subplot(3,1,3)
100 plt.xticks(rotation=90)
101 plt.ylabel("2012-2016")
102
103 plt.bar(DR['State/District'],DR['12-16 diff'], color=colors_DR)
104
105 #plt.savefig("1) res12,res16")
```

8. 막대그래프를 subplot 으로 나누어 그린다.
    - xlabel 을 회전하여 겹치지 않게 나타낸다.
    - 위 두 그래프는 xlabel 이 같으므로 나타내지 않게 하였다.

iv. 결과 의미 분석



- 위 두 막대그래프는 공화당, 민주당의 투표수 차의 백분율이다.
  - 파란 막대그래프는 그 주가 민주당에 더 많이 투표를 했다는 것이고, 반대로 빨간 막대그래프는 그 주가 공화당에 더 많이 투표를 했다는 것이다.
  - 그래프의 길이가 클수록 민주당과 공화당의 격차가 크다는 의미이다.
- 가장 밑 막대그래프는 2012 년도 결과에서 2016 년도 결과를 뺀 데이터이며, 막대그래프의 길이, 위치, 색에 따른 3 차원의 데이터를 보인다.
  - 파란 막대그래프는 두 해에서 모두 민주당의 투표수가 높음을 의미하고, 빨간 막대그래프는 두 해에서 모두 공화당의 투표수가 높음을 의미한다.

- 자홍색 막대그래프는 두 해 사이의 투표수의 우위가 민주당에서 공화당, 청록색 막대그래프는 두 해 사이의 투표수의 우위가 공화당에서 민주당으로 옮겨졌단 의미이다.
- 막대그래프가 위에 있으면 (양수 값을 가지면) 현재 정당의 지지율이 높아졌음을 의미하고,
  막대그래프가 아래에 있으면 (음수 값을 가지면) 반대정당의 지지율이 높아졌음을 의미한다.
- 막대그래프의 크기가 크면 추이가 크다는 의미이고,
  막대그래프의 크기가 작으면 추이가 작다는 의미이다.

Arizona 주를 예로 들어보자.

- Arizona 에서는 두 해 모두 빨간색으로 공화당을 더 지지함을 알 수 있다. 이는 추이 그래프에서 막대그래프가 빨간색인 것으로도 확인할 수 있다.
- 추이 그래프에서 음수 값을 가지는 것으로 보아, 2016 년에 민주당의 지지율이 높아졌음을 알 수 있다. 이는 2016 년 그래프가 2012 년 그래프의 값보다 더 줄어들었음에서도 확인할 수 있다.
- 추이 그래프에서의 막대 그래프의 크기가 0.1%보다 작은 것으로 보아 추이가 0.1%보다 작음을 알 수 있다.

v. 결과 분석

- 민주당의 지지율이 가장 높은 주는 두 해 모두 눈에 띄게 Washington, D.C.이다.
- 2012 년에 공화당의 지지율이 가장 높은 주는 Utah 이다.
- 2016 년에 공화당의 지지율이 가장 높은 주는 Wyoming 이다.

- 2012 년과 2016 년 사이의 추이가 가장 큰 주는 눈에 띄게 Utah 이다.
- 2012 년과 2016 년 사이의 추이가 가장 작은 주는

```
In [106]: DR[abs(DR['12-16 diff'])==min(abs(DR['12-16 diff']))]
Out[106]:
   State/District  Barack ObamaDemocratic  ...      D-R%  12-16 diff
12          Idaho                  212787  ... -0.317694    -0.00138

[1 rows x 15 columns]
```

0.00138%의 변화를 보인 Idaho 이다.

- 2012 년과 2016 년 사이, 현 정당의 지지율이 가장 높아진 주는 North Dakoda 이다.
- 2012 년과 2016 년 사이, 현 정당의 지지율이 가장 낮아진 주는 Utah 이다.
- 두 해 사이에 지지율이 높은 정당이 뒤바뀐 주들은

```
In [117]: for i in range(len(colors_DR12)):
     ...:     if colors_DR12[i] != colors_DR16[i]:
     ...:         print(DR12.iloc[i,0])
     ...:
Florida
Iowa
Michigan
Ohio[118]
Pennsylvania
Wisconsin[119]
```

6 주가 있다.

- 이 주들 중 민주당으로 바뀐 주들은

```
In [118]: for i in range(len(colors_DR12)):
     ...:     if colors_DR12[i] != colors_DR16[i]:
     ...:         if colors_DR12[i] > colors_DR16[i]:
     ...:             print(DR12.iloc[i,0])
     ...:

In [119]: for i in range(len(colors_DR12)):
```

없으며, 6 주 모두 공화당으로 바뀌었다.

```
In [119]: for i in range(len(colors_DR12)):
     ...:     if colors_DR12[i] != colors_DR16[i]:
     ...:         if colors_DR12[i] < colors_DR16[i]:
     ...:             print(DR12.iloc[i,0])
     ...:
Florida
Iowa
Michigan
Ohio[118]
Pennsylvania
Wisconsin[119]
```

- 그 중에서도 Iowa 가 가장 추이가 큰 것을 볼 수 있다.
- 미국 전체로는 민주당의 지지 백분율이 높으며 추이의 변화가 정당이 뒤집힐 정도로 크지 않음을 알 수 있다.

2. 1976 년~2016 년도의 정당 지지율 변화

i. 필요자료

- president-1976-2016.csv

  76~16 년 미국 대통령 선거 결과

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	year	state	state_po	state_fips	state_cen	state_ic	office	candidate	party	writein	candidate	totalvotes	version	notes
2	1976	Alabama	AL	1	63	41	US Preside	Carter, Jin	democrat	FALSE	659170	1182850	20171015	NA
3	1976	Alabama	AL	1	63	41	US Preside	Ford, Gera	republican	FALSE	504070	1182850	20171015	NA
4	1976	Alabama	AL	1	63	41	US Preside	Maddox, L	american	FALSE	9198	1182850	20171015	NA
5	1976	Alabama	AL	1	63	41	US Preside	Bubar, Be	prohibitio	FALSE	6669	1182850	20171015	NA
6	1976	Alabama	AL	1	63	41	US Preside	Hall, Gus	communis	FALSE	1954	1182850	20171015	NA
7	1976	Alabama	AL	1	63	41	US Preside	Macbride,	libertarian	FALSE	1481	1182850	20171015	NA
8	1976	Alabama	AL	1	63	41	US President			TRUE	308	1182850	20171015	NA
9	1976	Alaska	AK	2	94	81	US Preside	Ford, Gera	republican	FALSE	71555	123574	20171015	NA

ii. 데이터 가공 계획

이 csv 는 76 년도부터 16 년도까지, 각 주별로, 각 정당 투표수가 나타내어져 있다. 여기서 공화당과 민주당만을 추출해야 할 것이며, 년도와 주, 2 차원의 데이터를 multiindex 로 나타내야 할 것이다.

- **76 년 부터 16 년까지의 투표 결과**
  - 12 년 16 년보다 더 멀리 가 76 년도부터 16 년도까지의 흐름을 한 눈에 알아보기 쉽게 표현한다.
- **76 년 부터 16 년까지의 선형 회귀 그래프**
  - 얻은 그래프를 이용하여 선형 회귀 함수를 구하여, 정황의 흐름을 알아보기 쉽게 표현한다.

iii. 코드 분석

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  import seaborn as sns
4
5  df = pd.read_csv('president-1976-2016.csv')
6  df.drop(df.columns[[2,3,4,5,6,7,9,12,13]],axis=1,inplace=True)
```

1. 먼저 데이터 정제를 한다.
   (정제한 DataFrame)

Index	year	state	party	candidatevotes	totalvotes
0	1976	Alabama	democrat	659170	1182850
1	1976	Alabama	republican	504070	1182850
2	1976	Alabama	american independent party	9198	1182850
3	1976	Alabama	prohibition	6669	1182850
4	1976	Alabama	communist party use	1954	1182850
5	1976	Alabama	libertarian	1481	1182850
6	1976	Alabama	nan	308	1182850
7	1976	Alaska	republican	71555	123574
8	1976	Alaska	democrat	44058	123574

```
8  D=pd.DataFrame(df.loc[df['party'] == 'democrat'])
9  R=pd.DataFrame(df.loc[df['party'] == 'republican'])
```

2. party 에서 ‘democrat’과 ‘republican’만 따로 추출한다.

```
10 DR = pd.merge(D, R, how='outer', on=['year', 'state'])
11 DR.drop(DR.columns[[2,4,5]],axis=1,inplace=True)
12 DR.rename(columns={"candidatevotes_x": "Democrat",
13                    "candidatevotes_y": "Republican",
14                    "totalvotes_y":"total"},
15           inplace=True)
```

3. 추출한 DataFrame 을 병합하고 column 을 삭제 및 이름 재정의 등의 정제를 한다.

	state	year	Democrat	Republican	total
0	Alabama	1976	659170	504070	1182850
1	Alabama	1980	636730	654192	1341929
2	Alabama	1984	551899	872849	1441713

```
17  DR.sort_values(by=['state','year'],inplace=True)
18  DR.set_index(['state','year'],inplace=True)
```

4. 여기서 주별로 정렬할지 연도별로 정렬할지 정해야 한다. 현재 목표를 위해서는 주별로 먼저 정렬하고, 그 다음 연도별로 정렬한다.

```
20  DR.dropna(inplace=True)
21  DR=DR.astype(int)
```

5. 추가로 정제한다.
    - Null 값 처리
    - 숫자를 int 형으로 변환

```
23  DR = DR[~DR.index.duplicated(keep='first')]
```

6. 두 DataFrame 을 병합하다 보니 쓰레기 tuple 이 생성되어 이를 없앤다.

```
In [125]: all(DR.index.duplicated())
Out[125]: False
```

```
In [124]: DR.index.duplicated()
Out[124]:
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False,  True, False, False,
       False, False, False, False, False, False, False, False, False,
```

    - (이를 직접 확인하기 위한 코드)

232	Maryland	2016	1677928	943169	2781446
233	Maryland	2016	1677928	259	2781446
234	Maryland	2016	78	943169	2781446
235	Maryland	2016	78	259	2781446

```
25  DR['D%']=100*DR.iloc[:,0]/DR.iloc[:,2]
26  DR['R%']=100*DR.iloc[:,1]/DR.iloc[:,2]
27  DR['D-R%']=DR.iloc[:,3]-DR.iloc[:,4]
```

7. 투표수를 전체 투표수로 나누고 100 을 곱하여 백분율을 나타내고, 그 둘의 차를 구하였다.

	state	year	Democrat	Republican	total	D%	R%	D-R%
0	Alabama	1976	659170	504070	1182850	55.7273	42.6149	13.1124
1	Alabama	1980	636730	654192	1341929	47.4489	48.7501	-1.30126
2	Alabama	1984	551899	872849	1441713	38.2808	60.5425	-22.2617
3	Alabama	1988	549506	815576	1378476	39.8633	59.165	-19.3018
4	Alabama	1992	690080	804283	1688060	40.8801	47.6454	-6.76534
5	Alabama	1996	662165	769044	1534349	43.1561	50.1218	-6.96576
6	Alabama	2000	692611	941173	1666272	41.5665	56.4838	-14.9173

```
29  def lookup(states):
30      plt.figure()
31      plt.title(states)
32      sns.regplot(x=DR.loc[states].index,y=DR.loc[states]['D%']
33                  ).set(ylim=(0, 100), ylabel="")
34      sns.regplot(x=DR.loc[states].index,y=DR.loc[states]['R%']
35                  ).set(ylim=(0, 100), ylabel="")
```

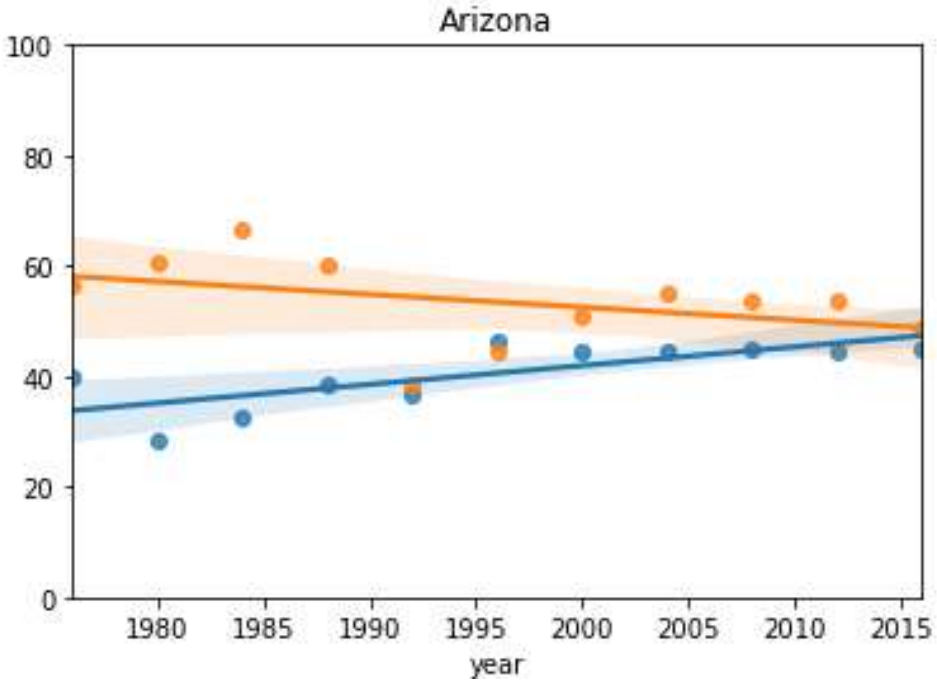
8. 이제 선형 회귀(linear regression) 그래프를 그리기 위하여 함수를 작성한다.
   - seaborn 을 이용하여 그래프를 그린다.
   - lookup 함수에 주 이름을 입력하면 그래프가 그려진다.
   - 그래프는 76 년도부터의 민주당 투표 백분율과 공화당 투표 백분율이 나오며 선형 회귀 그래프가 그려진다.
   - 그래프의 y 범위가 정적으로 0%부터 100%까지 모두 나타나게 하여 비교하기 용이하게 설정하였다.
   - 두 그래프가 겹쳐 보이게 되면 ylabel 값의 의미가 없어져 보이지 않게 하였다.

```
39  def lookupALL():
40      for st in statelist:
41          lookup(st)
42          plt.savefig("2) "+st)
```

9. 주 이름 리스트를 추출하여 lookup 함수에 입력할 수 있도록 하였다.
10. lookupALL() 함수를 구현하여 모든 주들의 그래프를 추출할 수 있도록 하였다.
    - 저장도 할 수 있게 구현하였다.

iv. 결과 의미 분석

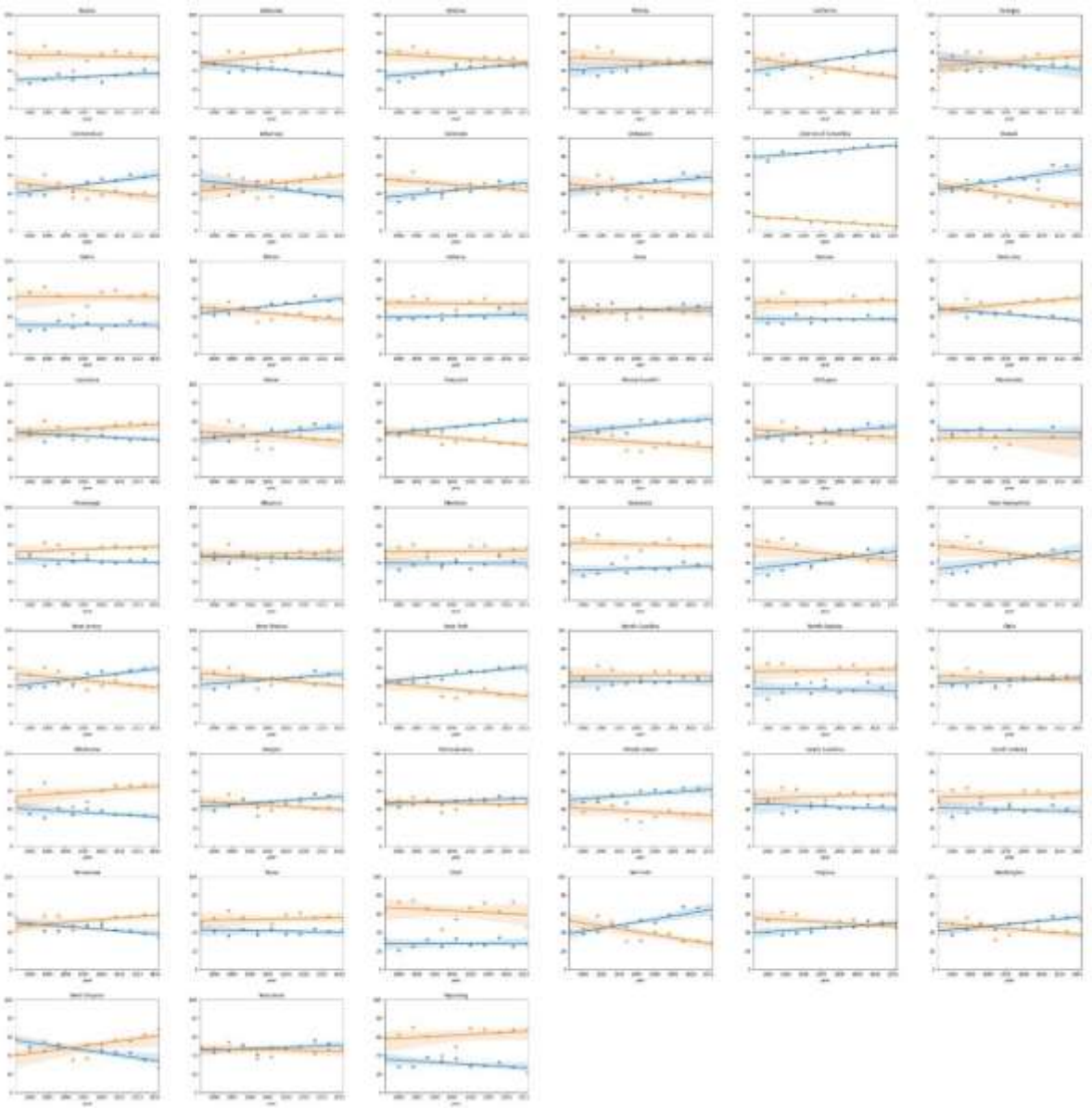
```
In [3]: lookup('Arizona')
```

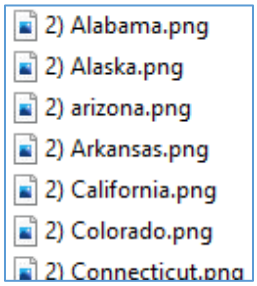


이번 역시 Arizona 주를 예시로 들어보겠다.

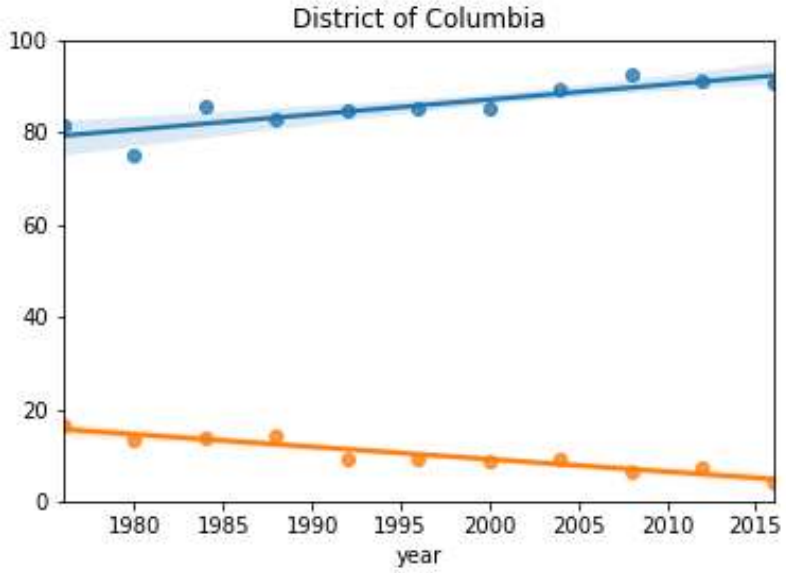
- x 좌표는 1976 년도부터 2016 년도까지 나타낸다.
- y 좌표는 0%부터 100%까지 나타낸다.
- 파란색 점들은 민주당 지지자 백분율이며 주황색 점들은 공화당 지지자 백분율이다.
- 파란색 실선 그래프는 민주당 지지자 백분율의 선형 회귀 그래프이며, 주황색 실선 그래프는 공화당 지지자 백분율의 선형 회귀 그래프이다.

v. 결과 분석

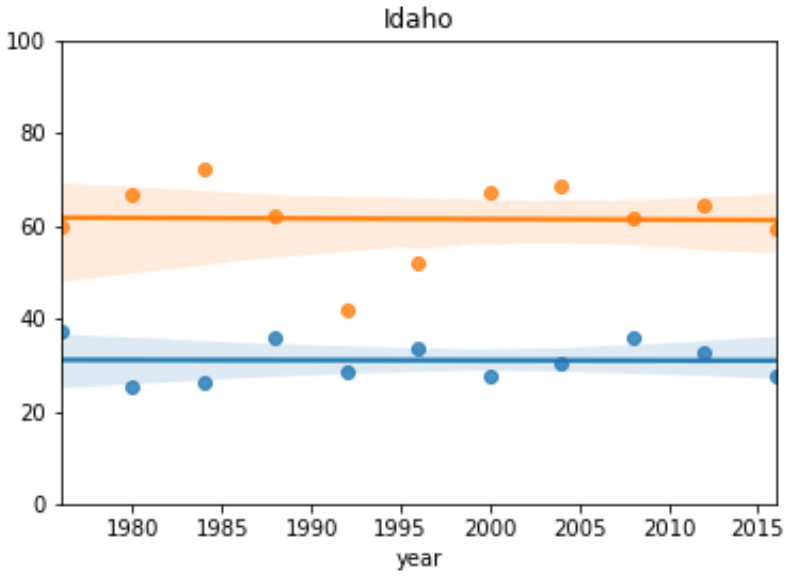




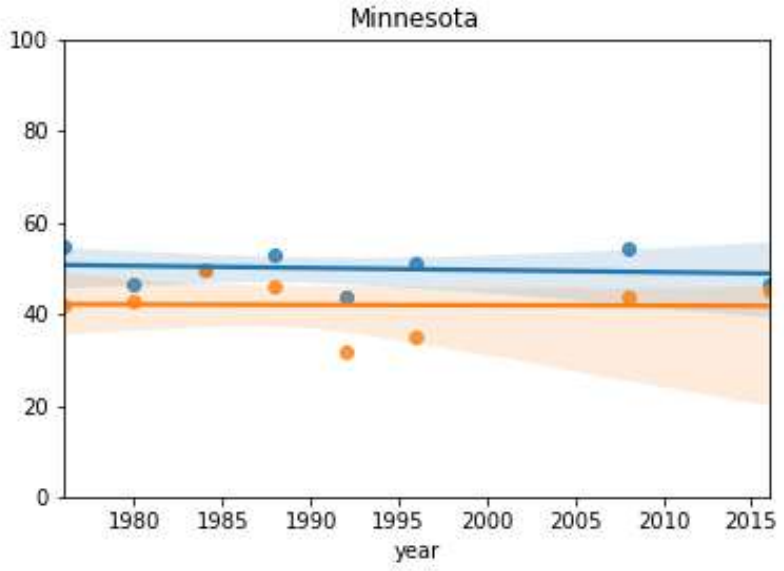
- 모든 51 개주의 그래프가 이런 식으로 출력되어 저장되었다.
- 대부분의 그래프가 평범하게 보이지만 눈에 띄는 그래프들도 몇 개 있다. 그 그래프들을 살펴보자.



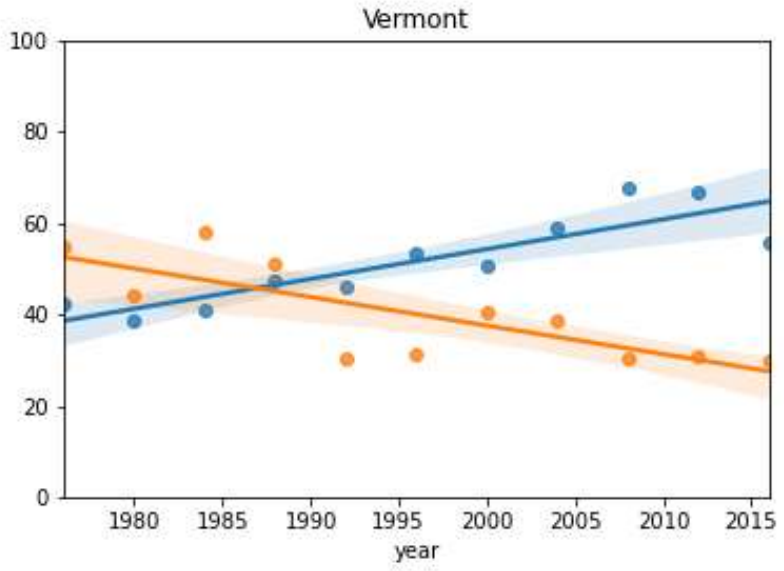
- District of Columbia, 워싱턴 DC 는민주당의 지지율이 공화당보다 충격적으로 높다.
- 그것도 모자라 76 년부터 변함없이 그를 유지해왔다.
- 그것 마저도 모자라 민주당의 지지율이 더욱 높아지고 있는 것을 확인할 수 있다.



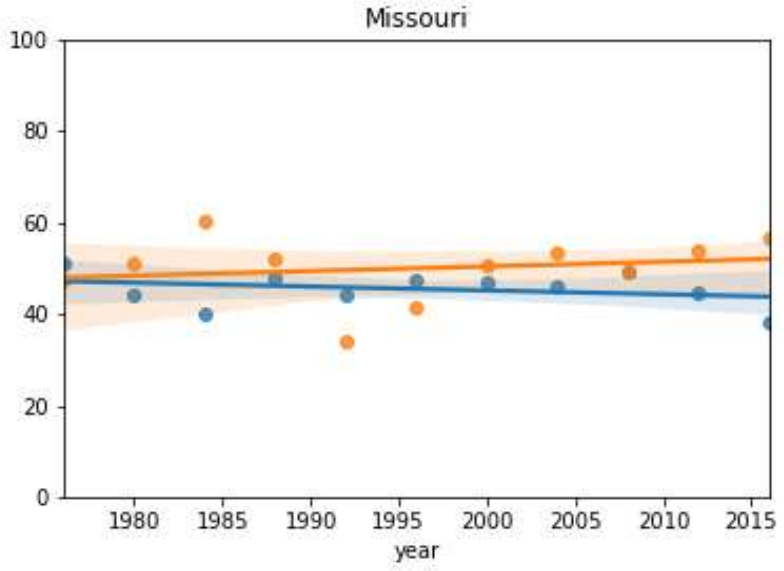
- Idaho 의 그래프는 우스울 정도로 완벽한 평행한 수준을 보인다. 이산 데이터임을 감안하면 더 그렇다.
- 이러한 평행을 보이는 그래프가 여러 개 있다. 그 중에서도 살펴보면,



- Minnesota 는 데이터가 결실되었다. 2000,2004 년도의 데이터가 없음을 볼 수 있다. 그러나 그래프는 평행을 보인다. 그 대신 그래프의 주변 범위가 넓은 것을 볼 수 있다.

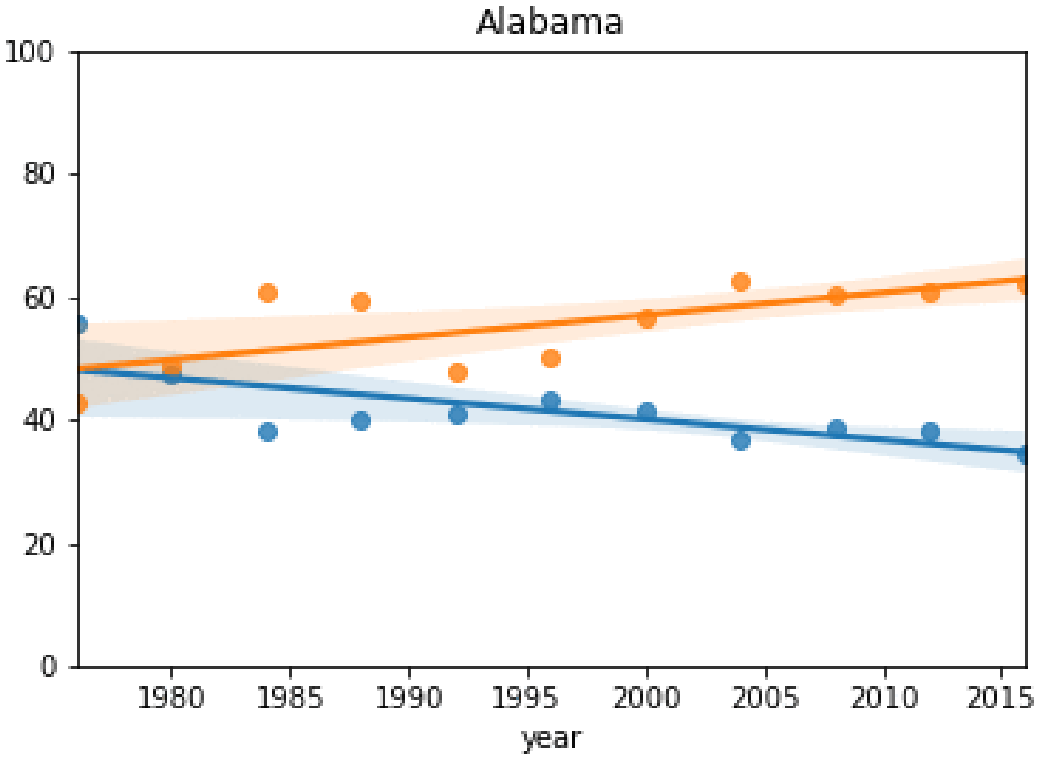


- Vermont 는 정당 지지율이 완전히 뒤집혔다. 이 모양을 보이는 그래프가 몇 개 있으나 이 주가 기울기가 가장 큰 것을 보아 변화가 가장 극심한 주라고 볼 수 있다.



- Missouri 는 내내 반반의 지지율을 유지하였다. 이러한 양상을 보이는 주들도 몇 개 있다.

- 모든 51 개의 그래프를 합쳐 하나의 gif 파일로 만들었다.
  (pdf 파일로 변환하면 보이지 않을 테니 첨부파일을 확인하여 주시길 바란다.)



3. 인구조사와 비교한 정당 지지율 비교

i. 필요자료

- country_facts.csv

  인구현황조사

fips	area_nam	state_abb	PST04521	PST04021	PST12021	POP01021	AGE13521	AGE29521	AGE77521	SEX25521	RHI12521	RHI2252
0	United States		3.19E+08	3.09E+08	3.3	3.09E+08	6.2	23.1	14.5	50.8	77.4	13
1000	Alabama		4849377	4780127	1.4	4779736	6.1	22.8	15.3	51.5	69.7	26
1001	Autauga C	AL	55395	54571	1.5	54571	6	25.2	13.8	51.4	77.9	18
1003	Baldwin C	AL	200111	182265	9.8	182265	5.6	22.2	18.7	51.2	87.1	9
1005	Barbour C	AL	26887	27457	-2.1	27457	5.7	21.2	16.5	46.6	50.2	47
1007	Bibb Coun	AL	22506	22919	-1.8	22915	5.3	21	14.8	45.9	76.3	22
1009	Blount Co	AL	57719	57322	0.7	57322	6.1	23.6	17	50.5	96	1
1011	Bullock Co	AL	10764	10915	-1.4	10914	6.3	21.4	14.9	45.3	26.9	70

  미국을 지역별로 나누는 fips code 마다 연령 별 인구, 인종, 수익, 출퇴근 시간 등 여러 데이터들을 제공한다.

  이 데이터들이 다루는 것들은 [v. 결과 분석]에서 살펴보겠다.

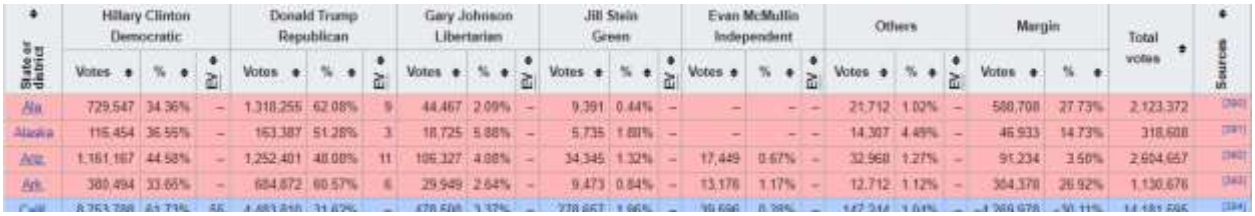
- country_facts_dictionary.csv

  인구현황조사 라벨 의미

column_name	description
PST045214	Population, 2014 estimate
PST040210	Population, 2010 (April 1) estimates base
PST120214	Population, percent change - April 1, 2010 to July 1, 2014
POP010210	Population, 2010
AGE135214	Persons under 5 years, percent, 2014
AGE295214	Persons under 18 years, percent, 2014
AGE775214	Persons 65 years and over, percent, 2014
SEX255214	Female persons, percent, 2014
RHI125214	White alone, percent, 2014
RHI225214	Black or African American alone, percent, 2014

- https://en.wikipedia.org/wiki/2016_United_States_presidential_election#Results_by_state

  2016 년 주별 투표 결과



ii. 데이터 가공 계획

투표결과와 인구조사와 어떠한 관계가 있을지 조사한다.

그를 위해서는 민주당 주와 공화당 주를 나누고 그들끼리 평균을 구한다.

그리고 평균들의 차를 구해 유난히 높거나 낮은 값들이 있는지 살펴보고 이들이 투표결과, 즉 지지율과 어떠한 관계가 있는지 살펴본다.

- **인구조사 데이터에 있는 분류에 따른 백분율로 나타낸 차이 분석**
    - 이 것을 계산하는 것이 여기에 쓰여 있는 것만큼 간단하지가 않다. 각 칼럼마다 데이터형과 의미가 다르기 때문인데, 왜 그런지 같이 설명하도록 한다.

iii. 코드 분석

```
1  import pandas as pd
2  import csv
3  import matplotlib.pyplot as plt
4
5  #read country_facts.csv
6  cDR = pd.read_csv('country_facts.csv')
7  cDR = cDR[cDR.isnull().any(axis=1)]
```

1. 이 데이터베이스는 이러한 구조로 되어있다.

fips	area_nam	state_abb	F
0	United States		
1000	Alabama		
1001	Autauga C	AL	
1003	Baldwin C	AL	
1005	Barbour C	AL	

   - fips 단위로 세분화되기 전 주별로 총합이 기재되어 있다. 전체 미국 tuple 과 더불어 이 tuple 들은 state_abb 칸이 공백이다.
   - 따라서 state_abb 칸이 공백인 데이터만 추출하면 주별로 데이터가 나올 것이다.

```
8  cDR.drop(cDR.columns[[0,2]],axis=1,inplace=True)
9  cDR0 = cDR.iloc[0]
10 cDR.drop(0, inplace=True)
11 cDR.reset_index(drop=True,inplace=True)
```

2. 가장 위 tuple 은 주 전체인 미국 전체의 대표 값이다. 이를 따로 떼어놓았고, 추가로 정제를 한다.

```
13 #read country_facts_dictionary.csv
14 dict={}
15 for line in csv.reader(open('country_facts_dictionary.csv')):
16     dict[line[0]]=line[1]
```

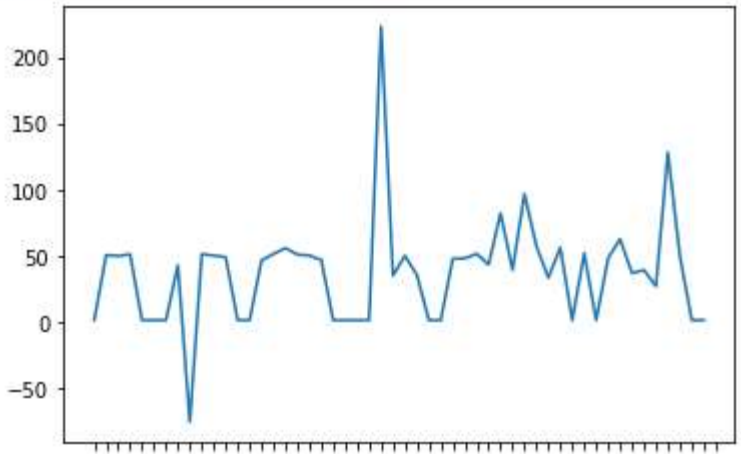
3. 데이터베이스의 인덱스 이름의 의미들을 알려줄 csv 파일의 내용들을 dictionary 형태로 불러온다.

Key	Type	Size	Value
AFN120207	str	1	Accommodation and food services sales, 2007 ($1,000)
AGE135214	str	1	Persons under 5 years, percent, 2014
AGE295214	str	1	Persons under 18 years, percent, 2014
AGE775214	str	1	Persons 65 years and over, percent, 2014
BPS030214	str	1	Building permits, 2014

4. 여기서 정제가 많이 필요하다.
   - 데이터를 백분율형으로 정제를 하려면 총합에서 전체를 빼야 한다.

- 하지만 모든 데이터가 그 공식이 통하는 것은 아니다!

```
In [72]: plt.plot(cDR[1:].sum()/cDR0[1:])
Out[72]: [<matplotlib.lines.Line2D at 0x1eec98de6d0>]
```



Index	0
area_name	United States
PST045214	318857056
PST040210	308758105
PST120214	3.3
POP010210	308745538
AGE135214	6.2
AGE295214	23.1
AGE775214	14.5
SEX255214	50.8

- 이 코드의 결과는 모든 값이 1 에 근접한 결과를 보여야 하지만 전혀 그렇지가 않다.
- 미국 전체 대표 값이 총합이 아니기 때문이다.

```
In [84]: for str in dict_np:
    ...:     print(cDRsD[str].mean()-cDRsR[str].mean())
    ...:
-0.5504761904761892
-1.9623809523809577
0.12476190476190219
0.088095238095238
1.0342857142856872
7.844285714285718
0.012142857142856567
11.33190476190477
-4.118571428571414
124172.38095238098
```

- 따로 구한 평균과 csv 에 있던 전체 미국 평균과 비교하면 0 에서 먼 값들이 있는 것을 확인할 수 있다.
- 왜 그런가 보았더니 총합 말고도 백분율 값, 평균 값들도 있기 때문에 각기 다른 접근방식이 필요하다.
- 대신 백분율 값과 평균 값은 계산하는 방법이 같기 때문에 같이 취급한다.

```
for key,value in dict.items():
    if "percent" in value:
        dict_p.append(key)
    else:
        dict_np.append(key)

dict_np.pop(0)
dict_np.remove('POP815213')
dict_np.remove('HSG445213')
```

5. 이 값들이 총합인지 백분율/평균인지 구분하는 방법이 필요하다.
    - 가장 눈에 띄는 방법은 dictionary 에 ‘percent’이라는 단어가 있는지 확인하는 방법이다.
    - 이를 위해서 dictionary 를 iterate 하면서 ‘percent’라는 단어가 있는지 확인하여 두 부류로 분류를 한다.
    - 그러자 민주당과 공화당의 총 합이 100 이어야하는데 그러지 못한 index 가 2 개 있었다.
    - 더 살펴보니, 단어 ‘percent’대신 ‘pct’나 ‘rate’라는 단어가 들어감을 확인하였다.
    - 이 둘을 따로 빼내어 다시 결과를 보았다.
    - 그럼에도 이상 현상을 보였다.
    - 다른 방법을 고안하였다.

```
dict_p=[]
dict_np=[]

DR_pcheck = cDR[1:].sum()/cDR0[1:]

for index,value in DR_pcheck.items():
    if 0.98 <= value <= 1.02:
        dict_p.append(index)
    else:
        dict_np.append(index)
dict_np.pop(-1)
```

6. 위 1 과 비교한 그래프에서 고안해내어, 아예 그 조건을 조건부로 설정했다.
    - 대푯값이 백분율/평균이었다면 결과가 1 이 나올 것이고,
      그렇지 않다면 합계를 의미하는 것이다.
    - 합계가 아닌 리스트에 가장 마지막에는 주 이름이 들어가 있어 더 편한 수작업으로 제외한다.
    - 그럼에도 이상 현상을 보였다. 이를 해결하려고 이틀을 훌쩍 보냈다.
    - 할 수 없이 여기에 덧붙여 수작업을 하였다.

```
#get DR16
res16 = pd.read_csv('res16.csv',header=[0,1])
res16=res16.drop(57)
res16=res16.drop(columns=['Sources'])

res16.replace('-', 0, inplace=True)
for k in range(1,17,3):
   res16.iloc[:,k] = res16.iloc[:,k].str.replace(',', '')
   res16.iloc[:,k] = pd.to_numeric(res16.iloc[:,k], errors='coerce')
   res16.iloc[:,k+2] = pd.to_numeric(res16.iloc[:,k+2], errors='coerce')
res16.iloc[:,21] = res16.iloc[:,21].str.replace(',', '')
res16.iloc[:,21] = pd.to_numeric(res16.iloc[:,21])

DR16 = pd.DataFrame(res16.iloc[:, [0,1,4,21]])
DR16.columns=DR16.columns.droplevel(1)
DR16.drop([20,21,30,31,32], inplace=True)
DR16.reset_index(drop=True, inplace=True)

DR16['D%']=DR16.iloc[:,1]/DR16.iloc[:,3]
DR16['R%']=DR16.iloc[:,2]/DR16.iloc[:,3]
DR16['D-R%']=DR16.iloc[:,4]-DR16.iloc[:,5]
```

7. 이 작업은 1 번 프로젝트에서 작업한 테이블과 비슷하다.
   - '-'으로 표현된 null 값 정제
   - String 데이터형을 int 형으로 형변환
   - 특수문자 제거
   - 세분화된 주별 투표 데이터가 있어 drop 하여 추가로 정제
   - 그에 따라 index 도 리셋

```
#cDR
cDR['DR']=DR16['D-R%']
cDR0['DR']=DR16.iloc[-1,-1]
```

8. 16 년도의 DataFrame 에 처음에 빼놓은 미국 전체 데이터를 병합한다.
   - 그와 더불어 7 번에서 구한 ('공화당%'-'민주당%') 데이터도 추가한다.

```
#separate D and R
cDRs_listR = []
cDRs_listD = []
for index, row in cDR.iterrows():
    if row['DR']>0:
        cDRs_listD.append(row)
    else:
        cDRs_listR.append(row)
cDRsD=pd.DataFrame(cDRs_listD)
cDRsR=pd.DataFrame(cDRs_listR)
```

9. 이제 주들을 민주당, 공화당으로 분류를 한다.

```
#categorizing data by D/R, p/np
cDRsDp=[]
cDRsDnp=[]
cDRsRp=[]
cDRsRnp=[]
```

10. 그리고 정당별로 분류한 데이터들을 대푯값 형식에 따라 분류를 해야 한다.
    - 따라서 4 가지로 분류가 될 것이다.
        - 민주당 – 데이터형: 확률/평균
        - 민주당 – 데이터형: 합계
        - 공화당 – 데이터형: 확률/평균
        - 공화당 – 데이터형: 합계

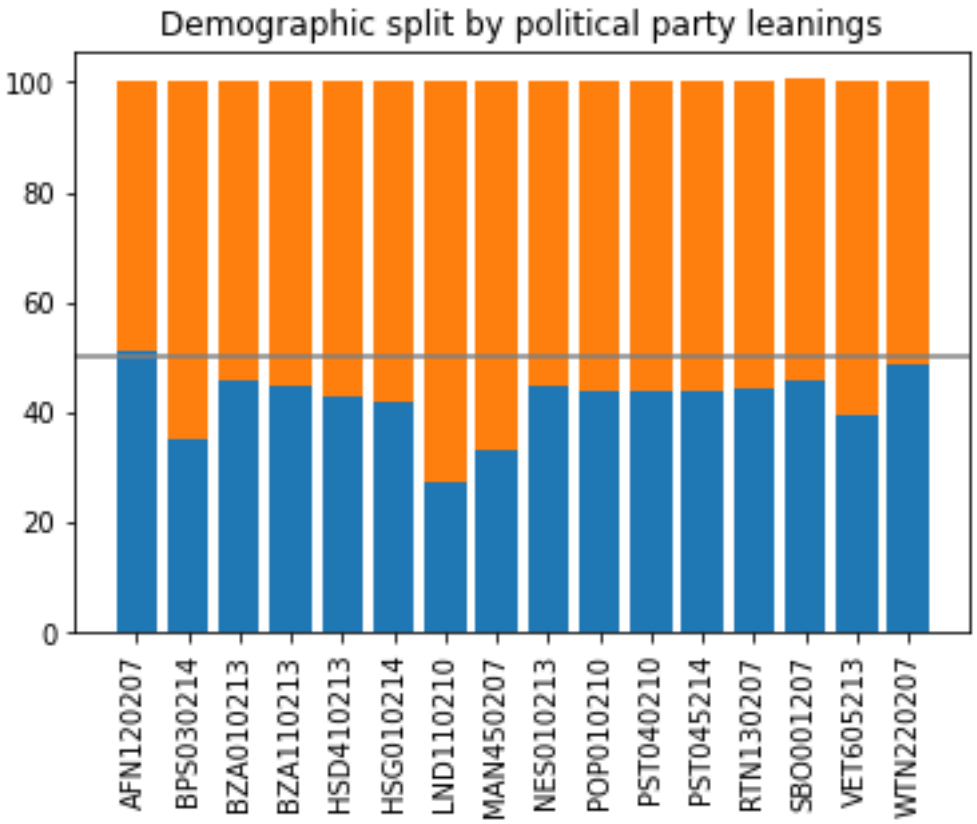
```
for str in dict_p:
    cDRsDp.append(100*cDRsD[str].sum()/cDR0[str])
    cDRsRp.append(100*cDRsR[str].sum()/cDR0[str])
for str in dict_np:
    cDRsDnp.append(100*cDRsD[str].sum()/(cDR0[str]*cDR0.size))
    cDRsRnp.append(100*cDRsR[str].sum()/(cDR0[str]*cDR0.size))
```

11. 데이터를 가공하는 법이 다르기 때문이라고 하였는데 어떻게 하는지 살펴보자.
    - 확률/평균인 경우 전체 대푯값은 평균의 값을 가진다.
      평균 확률/평균들의 총합을 구하고 이를 전체 평균으로 나누고 100 을 곱한다.
    - 총합인 경우는 조금 더 복잡하다.
      간단히 얘기하자면 부분평균을 총평균으로 합하는 과정이다.
      총합의 각 값들을 합하고, 이를 전체 값 평균의 개체 수만큼 곱하여 전체 평균을 내야한다.

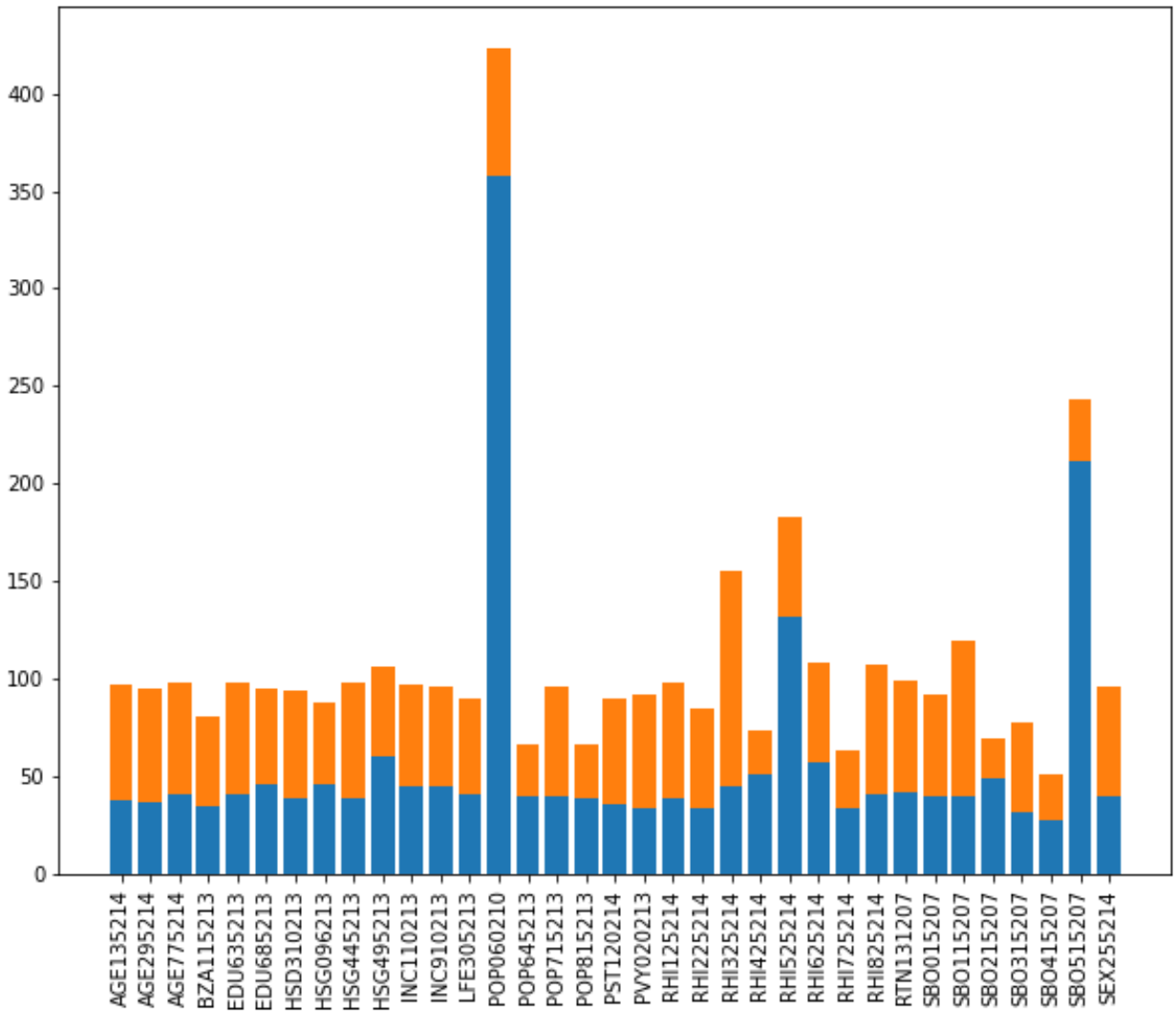
```
#plt
plt.figure(0)
plt.axhline(50,color='gray',linestyle="-")
plt.title('Demographic split by political party leanings')
plt.xticks(rotation=90)
plt.bar(dict_p,cDRsDp)
plt.bar(dict_p,cDRsRp,bottom=cDRsDp)

plt.figure(1,figsize=(10,8))
plt.xticks(rotation=90)
plt.bar(dict_np,cDRsDnp)
plt.bar(dict_np,cDRsRnp,bottom=cDRsDnp)
```

12. 그래프를 2 개를 그린다.
    - 막대그래프를 그려 민주당 막대그래프 위에 공화당 막대그래프를 올리는 stacked bar graph 형식으로 표현했다.
    - axhline 을 이용하여 50%에 점선을 그었다.

iv. 결과 의미 분석



- 확률/평균 데이터형의 그래프이다.
    - 민주당 막대그래프와 공화당 막대그래프가 쌓여 100%를 이루는 것을 볼 수 있다.

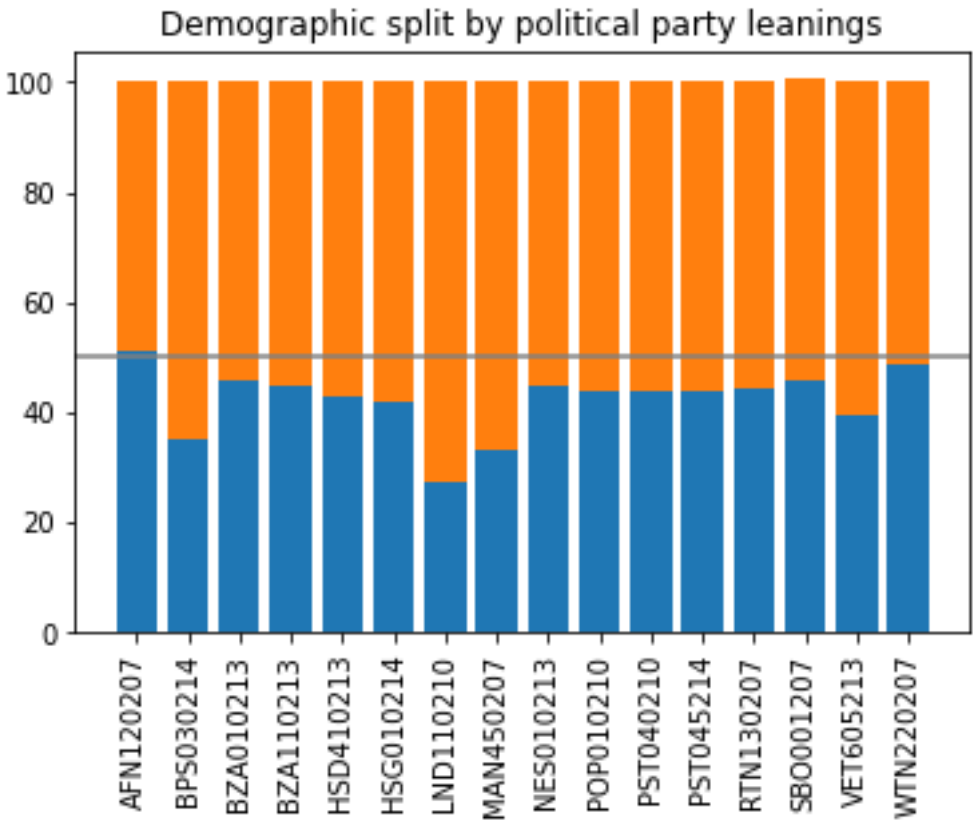


- 총합 데이터형의 그래프이다.
    - 두 막대그래프의 총합이 100 이 아님을 확인할 수 있다.
- 가장 눈에 뜨이는 부문이 있다.
  혹시 이것이 에러가 아닌가, 내가 무언가를 잘못했나 이틀간 밤을 세워가며 재검하고 코드를 변경하고 데이터들을 하나하나 점검하는 등, 애를 써보았다. 하지만 계속 이러한 양상을 보여 맞는 값으로 생각하겠다.

- 인구 현황조사에는 이러한 데이터들을 다룬다.
    - AFN: 숙박 및 식당 수익
    - AGE: 연령대
    - BPS: 건물 허가증
    - BZA: 농장을 제외한 사적 건물 및 인용
    - EDU: 학위 수준
    - HSD: 거주현황
    - HSG: 거주지 관련 (세입자, 다층주택 여부 등)
    - INC: 수입
    - LFE: 출퇴근 시간
    - LND: 땅 면적
    - MAN: 생산 출고량
    - NES: 비정규직 회사
    - POP: 인구
    - PST: 인구 백분율
    - PVY: 가난 인구
    - RHI: 인종
    - RTN: 소매 판매량
    - SBO: 회사 수
    - SEX: 여성 인구수
    - VET: 참전 용사 수
    - WTN: 도매 판매량

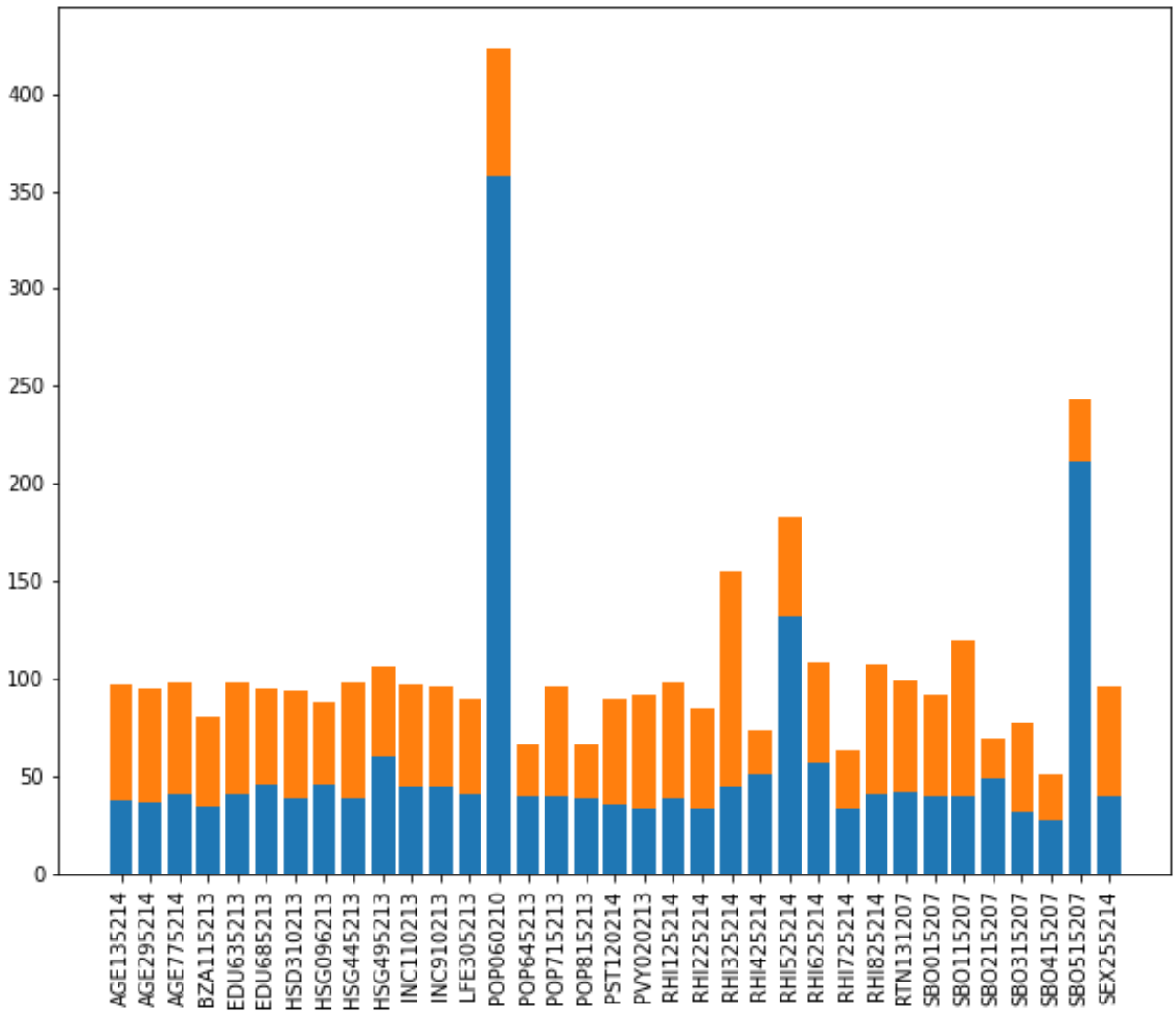
AGE135214	str	1	Persons under 5 years, percent, 2014
AGE295214	str	1	Persons under 18 years, percent, 2014
AGE775214	str	1	Persons 65 years and over, percent, 2014
BPS030214	str	1	Building permits, 2014
BZA010213	str	1	Private nonfarm establishments, 2013
BZA110213	str	1	Private nonfarm employment, 2013
BZA115213	str	1	Private nonfarm employment, percent change, 2012-2013
EDU635213	str	1	High school graduate or higher, percent of persons age 25+, 2009-2013
EDU685213	str	1	Bachelor's degree or higher, percent of persons age 25+, 2009-2013
HSD310213	str	1	Persons per household, 2009-2013
HSD410213	str	1	Households, 2009-2013
HSG010214	str	1	Housing units, 2014
HSG096213	str	1	Housing units in multi-unit structures, percent, 2009-2013

v. 결과 분석



(대푯값이 백분율인 부류)

- 이중에서는 민주당이 월등하는 것이 한 부류밖에 없다!
    - AFN120207: 숙박 및 식당 수익, 2007 ($1,000)
- 나머지 부류에서는 공화당이 더 앞선다. 그 중 가장 많이 앞서는 부류들을 살펴보자.
    - LND: 땅 면적
    - BPS: 건물 허가증
    - MAN: 생산 출고량



(대푯값이 절댓값인 부류)

- 딱 봐도 눈에 띄는 것이 4 부류가 보인다.
    - POP060210: 제곱마일당 인구 수, 2010
      민주당 Washington D.C.가 매우 압도적으로 높다.
    - RHI325214: 원주민 및 Alaska 원주민 수, 2014
      Alaska, New Mexico 에서 높은 값들을 보였다.
    - RHI525214: 하와이 및 태평양 섬 주민 수, 2014
      역시 Hawaii 가 가담을 했다.
    - SBO515207: 하와이 및 태평양 주민 보유 회사 수, 2007
      전과 같은 이유이다.
- 이렇게 조사한 결과만으로는 쉽게 무슨 말인지 알기가 힘들다.
- 다른 방법이 필요하다. 부류별로 세분화하여 분석해보자.

239.1
48.5
738.1
460.8
9856.5
350.6
168.4
211.8
19

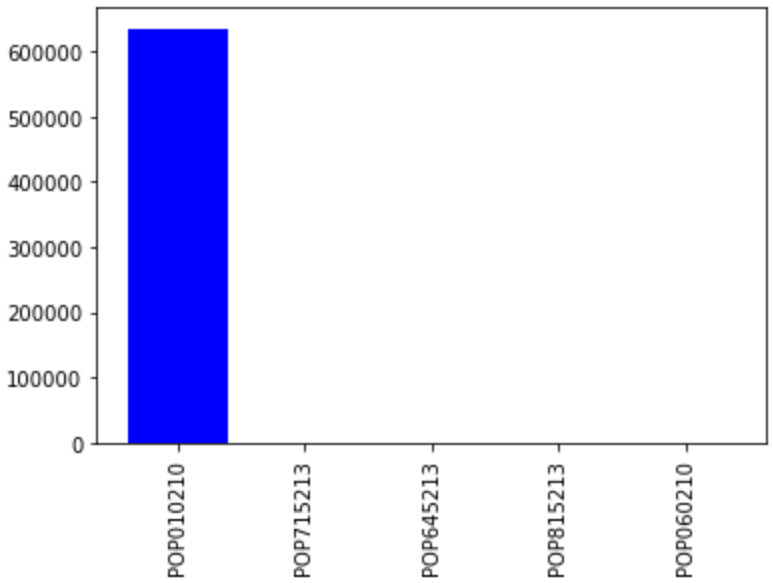
vi. 코드 분석 (v2)

```
119  colors=[]
120  def lookup(la):
121      colors.clear()
122      i=0
123      dict_temp=dict
124      plt.title(la, fontsize=14)
125      plt.xticks(rotation=90)
126      for str in dict:
127          if(la in str):
128              if(cDRsD[str].mean()>cDRsR[str].mean()):
129                  colors.append('b')
130              else:
131                  colors.append('r')
132      for str in dict_temp:
133          if(la in str):
134              plt.bar(str,cDRsD[str].mean()-cDRsR[str].mean(),
135                      color=colors[i],log=flag_log)
136              i+=1
```

- 분류별로만 그래프를 나타나게 하는 함수를 구현하였다
  - 프로젝트 1 번의 그래프와 비슷하게 색을 입히는 코드와
  - 그래프를 그린다. 그 그래프는 민주당의 평균과 공화당의 평균의 차를 구한다.

- 그렇게 그래프를 그리다 보면 읽기 어려운 그래프들도 생긴다.

```
In [92]: lookup('POP')
```

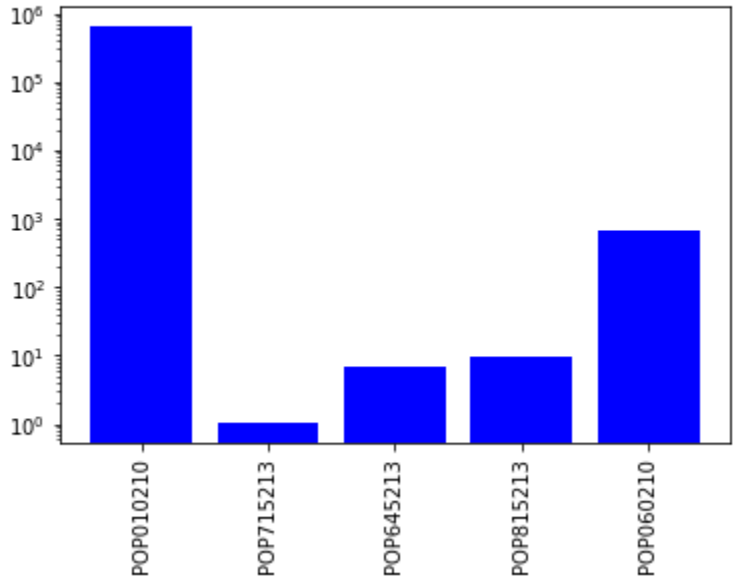


- 이를 위해서 y 축을 n 형태로 둘지 $\log n$ 형태로 둘지 정하는 스위치를 구현하였다.

```
114 flag_log=False
115 def log_switch():
116     global flag_log
117     flag_log = not flag_log
```

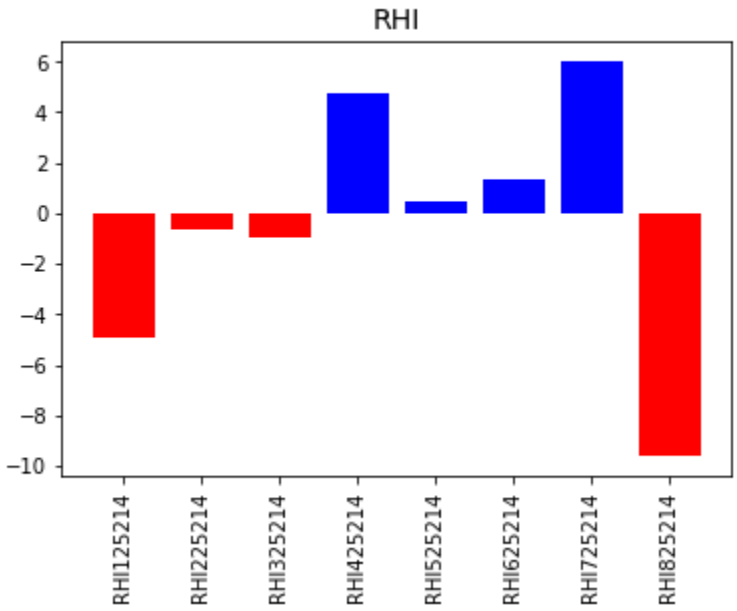
- $y = x$에서 $y = \log x$ 그래프로 전환할 시에는 이 함수만 불러주면 된다.

```
In [93]: log_switch()
    ...: lookup('POP')
```



vii. 결과 의미 분석 (v2)

```
In [100]: lookup('RHI')
```



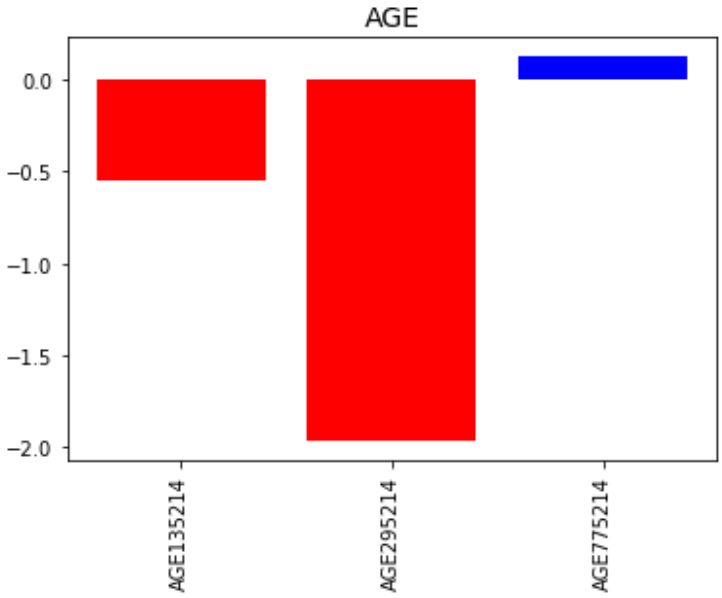
- RHI 로 검색을 하면 자료형의 대푯값을 기준으로 y 축의 값들이 정해진다.
- 분류된 그래프에서 민주당의 평균이 높으면, 다시 말해 양수의 값을 가지면 파란색, 반대이면 빨간색으로 표시하게 하였다.

viii. 결과 분석 (v2)

(여기서 얻은 결과들은 올바르게 나왔는지를 확인하기 위해 이 리포트 결과와 비교하였다.)

「Wide Gender Gap, Growing Educational Divide in Voters' Party Identification」, 『Pew Research Center (U.S. Politics & Policy)』, MARCH 20, 2018, https://www.pewresearch.org/politics/2018/03/20/1-trends-in-party-affiliation-among-demographic-groups/, DECEMBER 17, 2020.

- 먼저 연령, 성별, 인종, 학력, 회사 수를 보겠다.

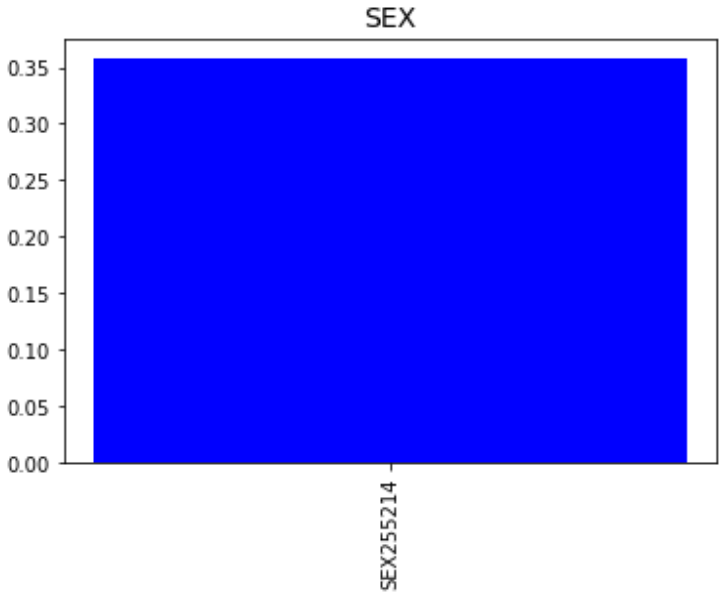


AGE135214: 연령 만 5 세 이하, 2014

AGE135214: 연령 만 18 세 이하, 2014

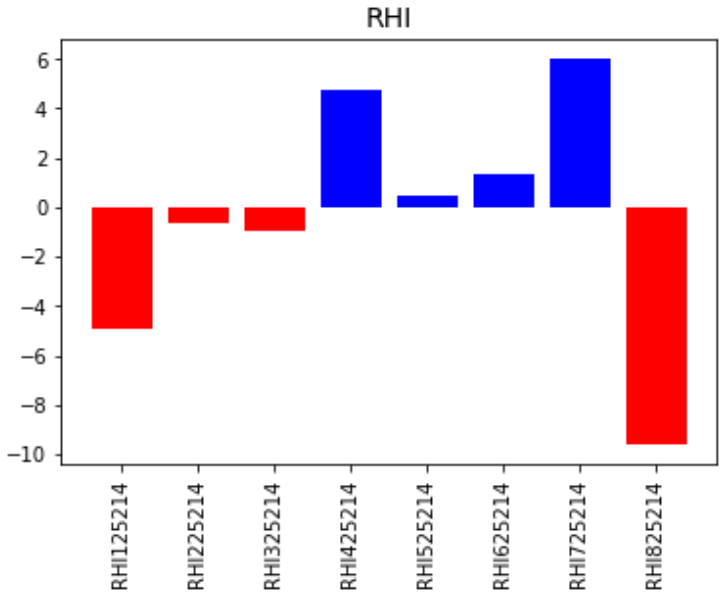
AGE135214: 연령 만 65 세 이상, 2014

- 만 18 세 이하의 인구의 비중이 높은 주들은 공화당을 추구하는 주가 더 많으며
- 만 65 세 이상의 인구의 비중이 높은 주들은 민주당을 추구하는 주가 더 많다.



SEX255214: 여성 수 백분율, 2014

- 여성 인구들이 높은 주들이 민주당의 투표율이 높은 것으로 보아,
- 여성 인구들은 민주당에 투표할 확률이 높다고 볼 수 있다.



RHI125214: 백인 백분율, 2014

RHI225214: 흑인, 아프리카계 미국인 백분율, 2014

RHI325214: 토착민 백분율, 2014

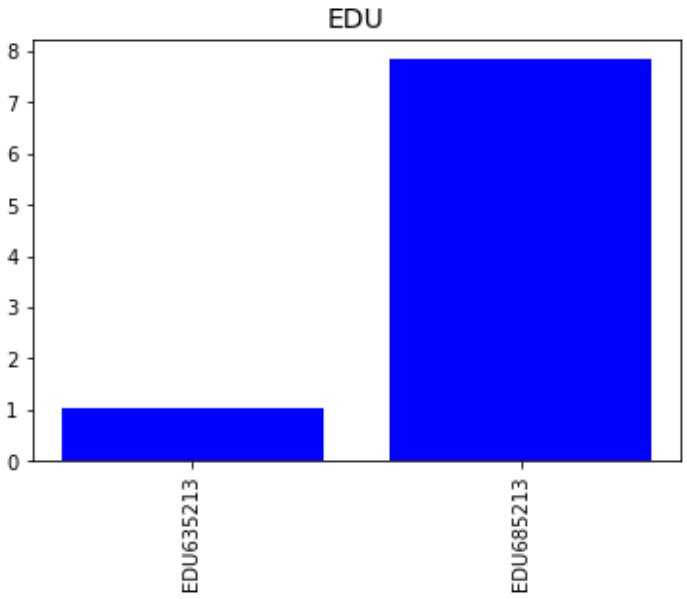
RHI425214: 아시아인 백분율, 2014

RHI525214: 하와이 및 다른 섬 주민 백분율, 2014

RHI625214: 혼혈인구 백분율, 2014

RHI725214: 히스패닉, 라티노 백분율, 2014 *RHI825214: 히스패닉, 라티노 제외 백인 백분율, 2014*

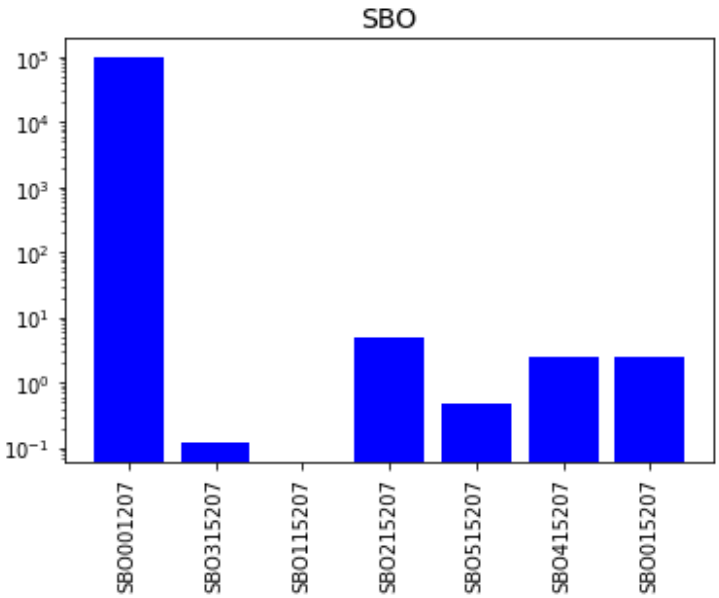
- 백인 인구들이 높은 주들은 공화당에 투표를 더 많이 하였다.
- 그와 반대로, 아시아인, 하와이 주민, 혼혈인과 히스패닉, 라티노 계열 인구들이 더 높은 주들에서는 민주당에 투표를 더 많이 하였다.
- 그들이 아프리카계 미국인이나 인도계 미국인들이 많은 주보다도 훨씬 높게 민주당의 지지율을 보인다.
- 히스패닉, 라틴계 인구들을 백인 투표에서 제외하자 공화당 쪽으로 더 치우친 것을 볼 수 있다.



EDU635213: 고등학력 이상, 나이 25+, 2009~2013

EDU635213: 박사학력 이상, 나이 25+, 2009~2013

- 학력이 높은 자들을 민주당에 투표를 더 많이 한다.
- 학력이 높아질수록 민주당에 투표율이 더 증가한다.



SBO001207: 총 회사 수, 2007

SBO015207: 여성 소유 회사 수, 2007

SBO115207: 토착민 소유 회사 수, 2007

SBO215207: 아시아인 소유 회사 수, 2007

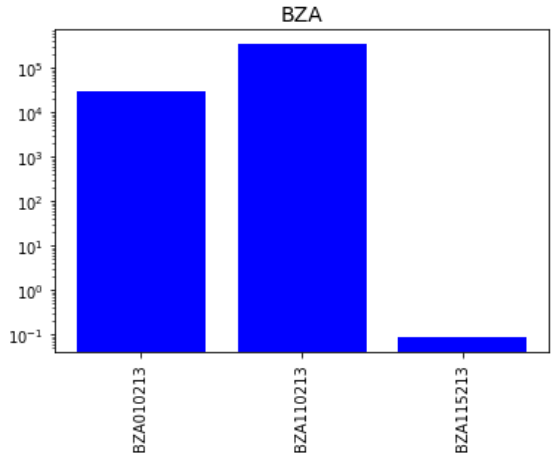
SBO315207: 흑인 소유 회사 수, 2007

SBO415207: 히스패닉 소유 회사 수, 2007

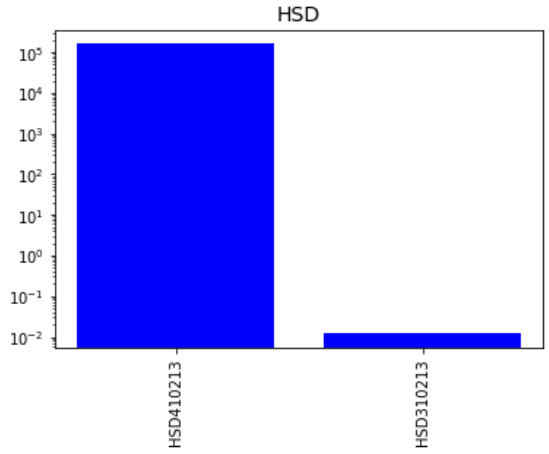
SBO515207: 하와이인 및 섬 주민 소유 회사 수, 2007

- 회사 수가 많은 주들은 적은 주들과 달리 민주당의 지지율이 높게 보인다.

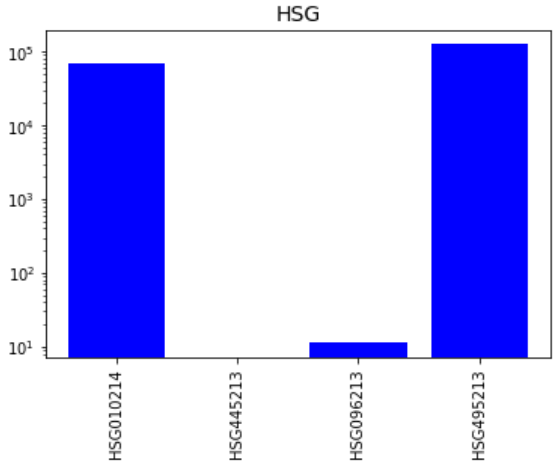
- 그 외 그래프들이다. 파란색이 많이 보인다.



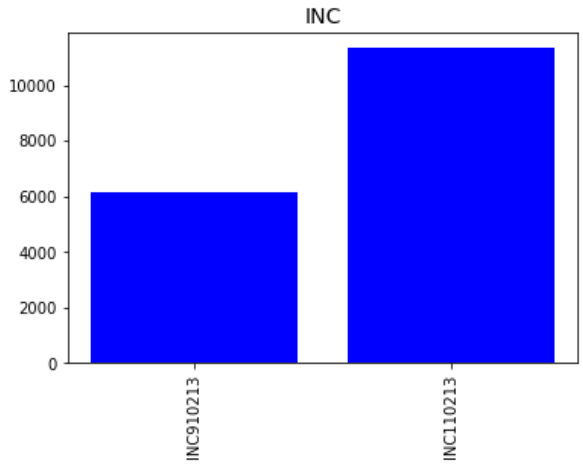
o BZA: 농장을 제외한 사적 건물 및 인용



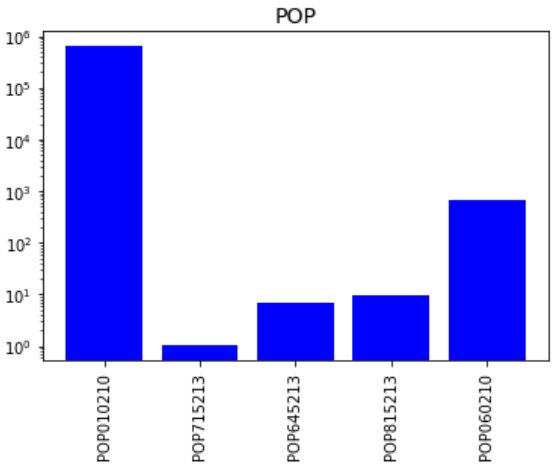
o HSD: 거주현황



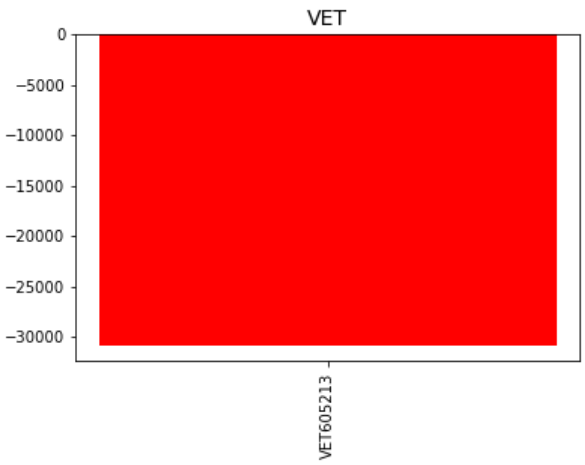
o HSG: 거주지 관련 (세입자, 다층주택 여부 등)



o INC: 수익



o POP: 인구 수



o VET: 참전용사 수

COMMENTS

- 프로젝트 관련 추가 comment

- 지금까지 배운 여러가지 기능들을 추가하고 싶었다. sklearn, tensorflow 등을 사용하고 싶었지만, 본인이 고른 부문에서는 크게 이용될 곳이 없었다. 기계학습을 해보았자 3 번 프로젝트에 사용한 인구조사에 밖에 적합한 곳이 없다. 투표수로 기계학습을 한다고 하여도 조사를 주별로 하였기 때문에 큰 효과를 보지 못할 것이라고 생각한다. 따라서 간단명료하고, 보기 쉬운 프로젝트 3 개를 간추려보니 기계학습에 관한 것은 하나도 뽑히지 못하였다. 아쉽게도 말이다.

- 중간보고서와 다른 계획 변경 요소들

- 20 년 결과를 리포트에 반영할 수 없었다! 결과를 얻은 시점에는 프로젝트가 이미 많이 진행된 시점일 뿐더러 이 리포트가 더 길어질 것이다. 이 프로젝트를 시작한 이유가 2020 년도 미국 선거를 기념으로 선정한 주제였는데 그 데이터가 포함되지 못하여 안타깝다.
  하지만 이 점은 미리 예견하였고 이미 중간보고서에서 명시를 하여 큰 타격은 없었다.
- 그 이외의 것들은 모두 성취한 것 같다.
  - 연도별 투표 결과 비교
  - 경향 이동 분석
  - 인구조사와의 관련성 분석
  - 76 년도부터의 정당 변화 관찰

- 느낀점

- 본인은 미국 주들이 어떠한 경향을 나타내는지 궁금했었다. 미국이 대체로 어떻다는 것만 알지 그 세세한 주들은 특성이 어떤지 잘 알지 못하였다. 따라서 이 기회로 좀 더 알아볼 겸 이 주제를 택한 것이었다.
- 여기서 알아낸 점은 워싱턴 DC 가 다른 주들과는 현저히 차이가 난다는 점을 시작으로, 아시아인들이 흑인들보다 민주당을 더 지지한다는 점, 미국은 전체적으로 민주당의 지지율이 높다는 점 등, 여러 새로운 통찰력을 얻은 기분이다.

- Python 을 프로젝트를 진행한 한달동안 쭉 이용하면서 C 언어나 Java 에 비해 느낌이 매우 다르다는 것을 시간이 지나갈수록 느끼고 있다.
- 사이트에서나 보았던 그래프들이 어떻게 만들어지는지도 경험하고 직접 만들어보게 되어 신기했던 점들도 있다.

파이썬프로그래밍 최종보고서

2017156034 전상민