DOT & Bay State Banner Voting Patterns
Final Project Report

Abdullah Albijadi, Audrey McMillion, Gordon Ng, Ji Zhang, Thachathum Amornkasemwong

Client Background

The Bay State Banner is an African American owned news weekly that reports on the political, economic, social and cultural issues that are of interest to African American and English speaking Latinos in Boston and throughout New England.

The Dorchester Reporter, founded in 1983, is a community newspaper which is published every Thursday. It serves the Dorchester and adjoining communities in Boston.

Project Description

The goal of this project is to understand how racial voting patterns in Boston changed over the past decade on the Ward Precinct level. The components to measure voting behavior include ballots cast, blanks cast, registration, eligible voters, and turnout as an auxiliary measure.

Tableau Workbooks

Using the link below, you can access all interactive Tableau workbooks created for this project:

- https://drive.google.com/drive/folders/1YwAT2eGOoU_x8w0DvmMddomr_7Jo3j8M?usp=sharing

Datasets

Using the link below, you can access all datasets used for this project:

- https://drive.google.com/drive/folders/1OEk5fTLiAlvpBDsPbXt1_7ZxpnVtsY6u?usp=sharing

Key Questions

Overarching Project Question: Are there any overarching racial voting patterns in Boston on the Ward-Precinct level, and how does this vary across election type over the past decade?

Key Questions:

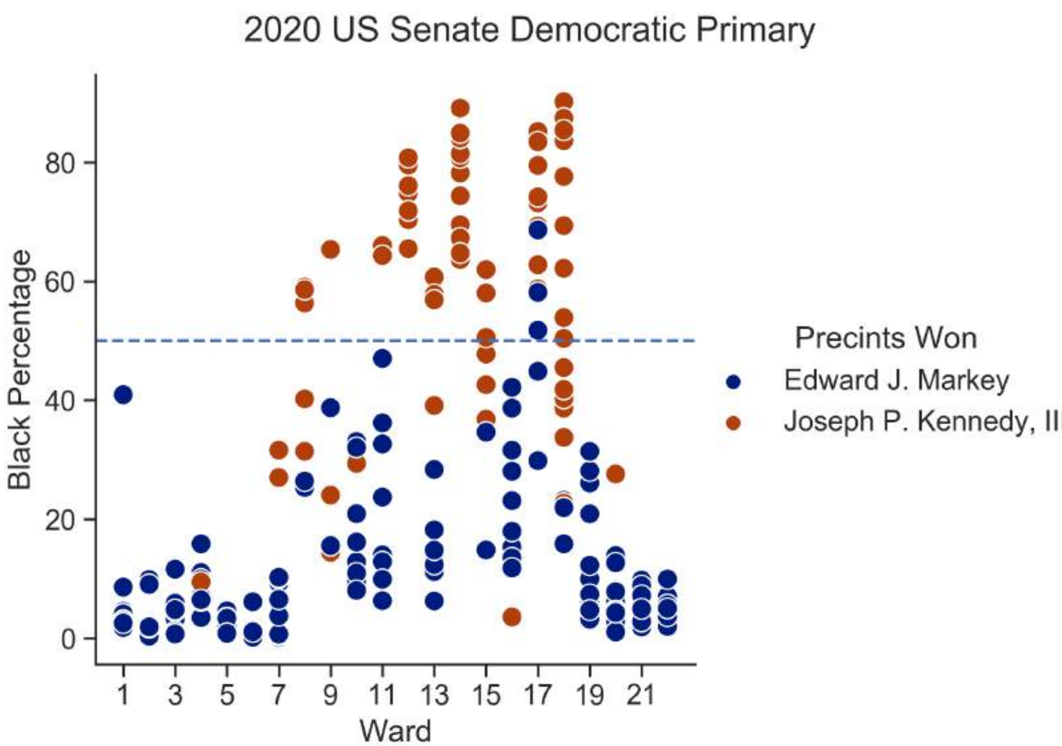
1. How do racial voting patterns differ by election type?
2. What are the key predictors in determining support for Black candidates for Boston?
3. How has voter turnout by precinct changed across city council election year?
4. What are changes in District 3 particularly?
5. How have the demographic changes in Boston affected local elections over the past decade?

Key question: How have voting patterns of Black populations in Boston changed by election type?

PROCESS: After collecting missing data, we performed some preliminary analysis focusing mainly on how voting patterns change with Black demographics.

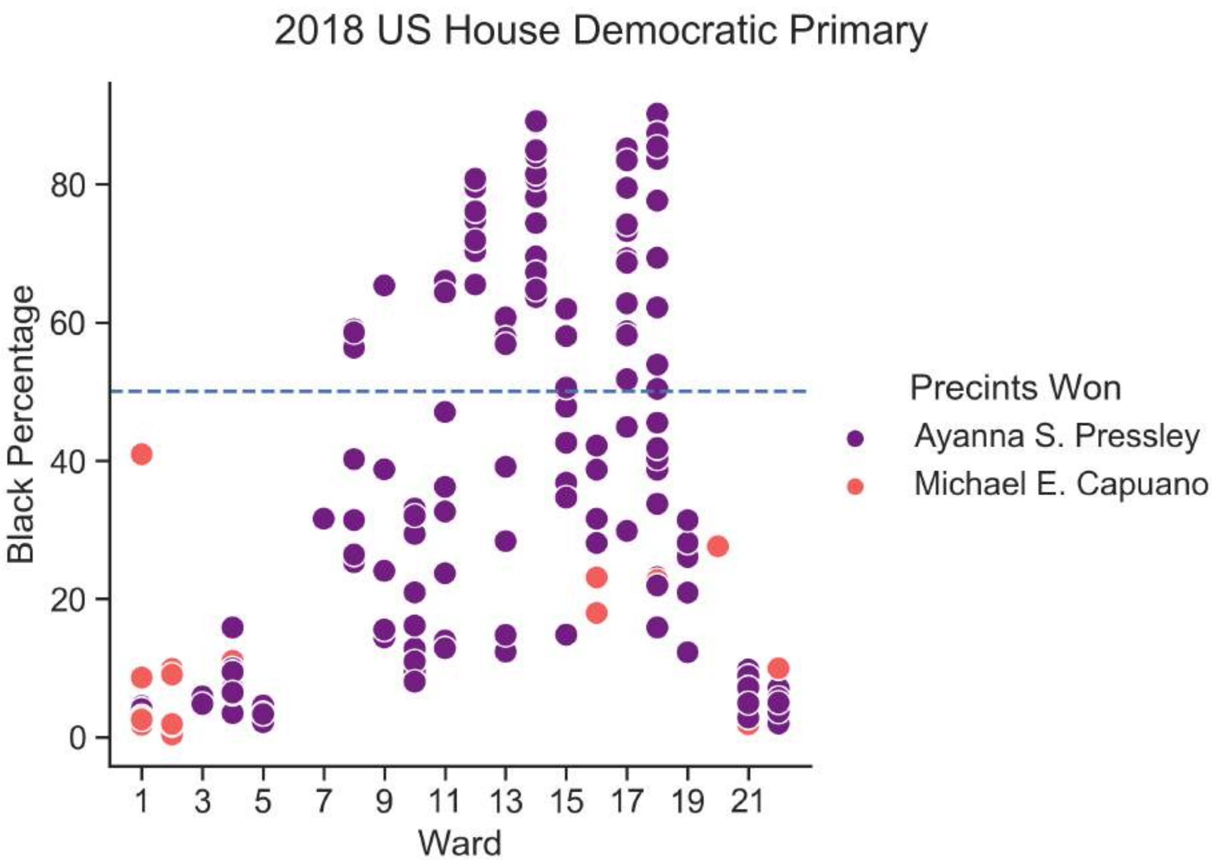
Part 1: Senate 2020 Demorcratic Primaries and 2018 US Democratic House Primaries

1. For the Senate 2020 Democratic Primaries, here is what we found:



Here we found a clear racial pattern. Across all Wards, Kennedy won most of the precincts with Black majority as we can notice above the horizontal line, while Markey dominated the precincts with Black minority, and specifically the ones with lowest Black percentage living there.

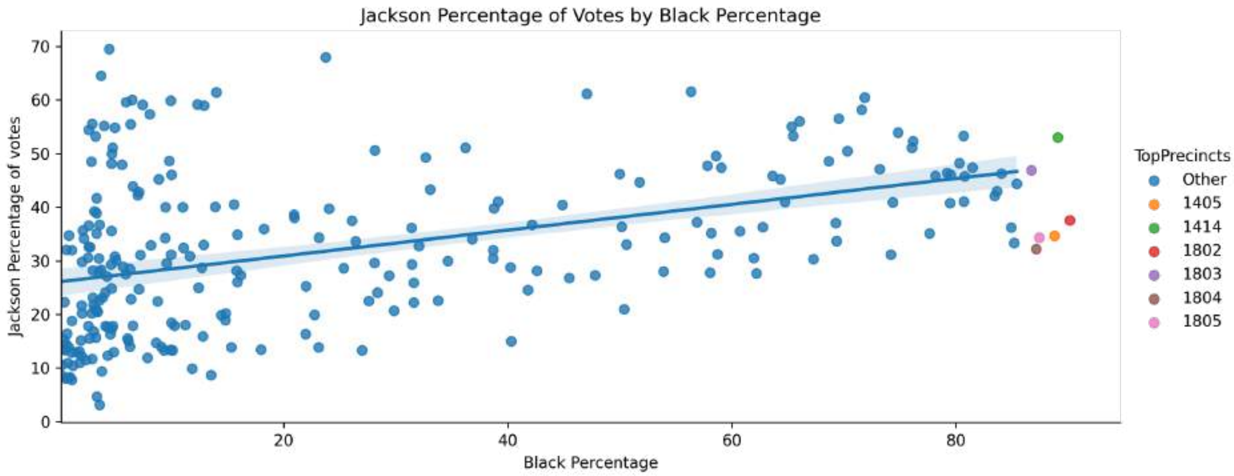
2. And for the 2018 US Democratic House Primaries:



Similarly, another racial pattern. Across all Wards, Pressley won all of the precincts with Black majority as we can notice above the horizontal line, while Capuano' wins were in the precincts with Black minority, with most of them being in lowest Black percentage precincts.

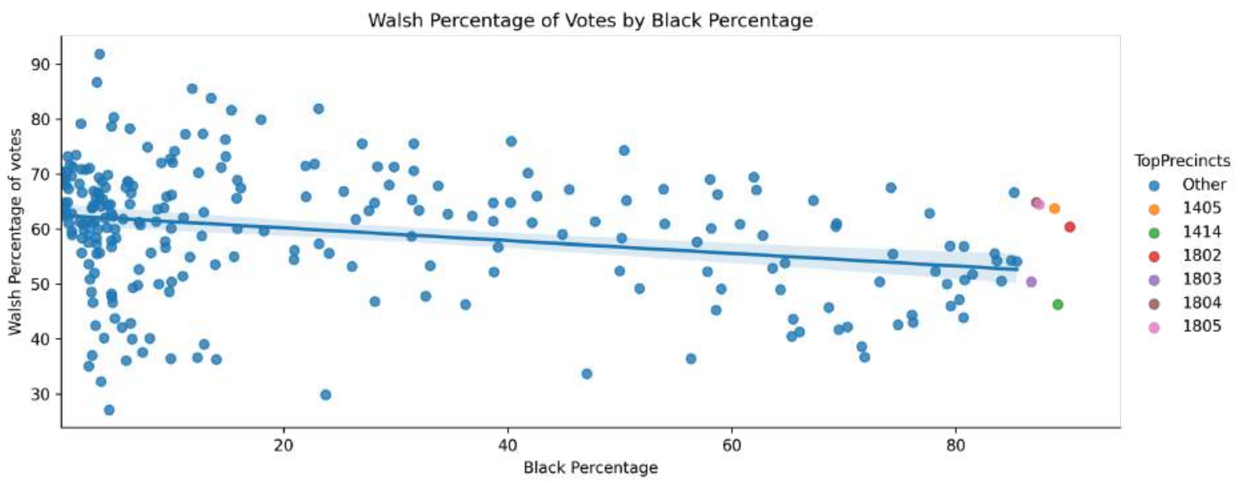
Part 2: Mayoral 2017 Preliminary Race and Mayoral 2017 General Turnout Race

1. For the Mayoral 2017 Preliminary Race, we obtained the following results:



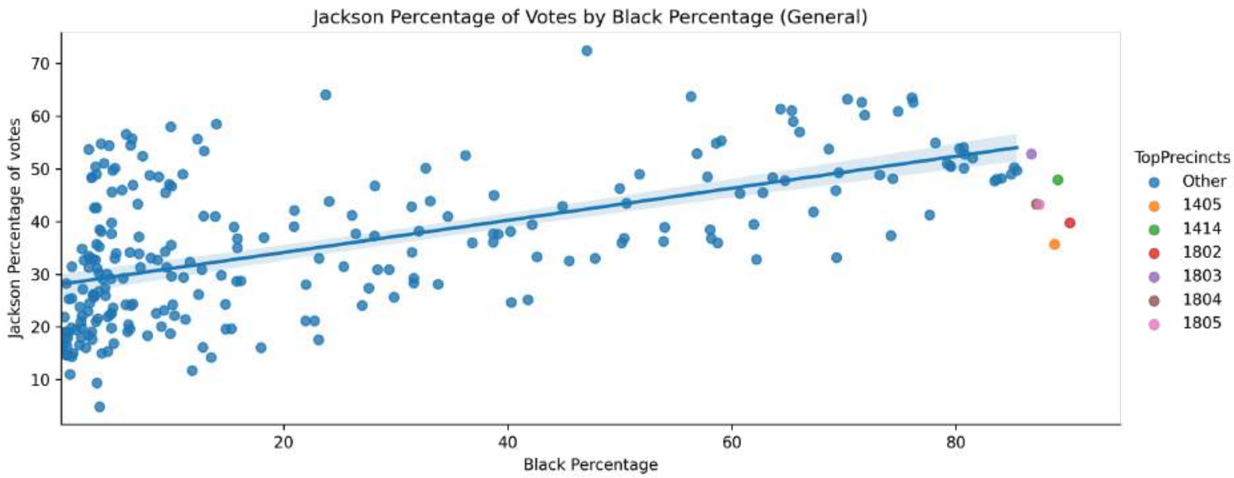
Here, we found that there is a slight positive correlation (correlation coefficient: 0.42) between a higher Black population and greater percentage of votes for Tito Jackson.

Similarly, we analyzed Marty Walsh's percentage of votes by Black Percentage:



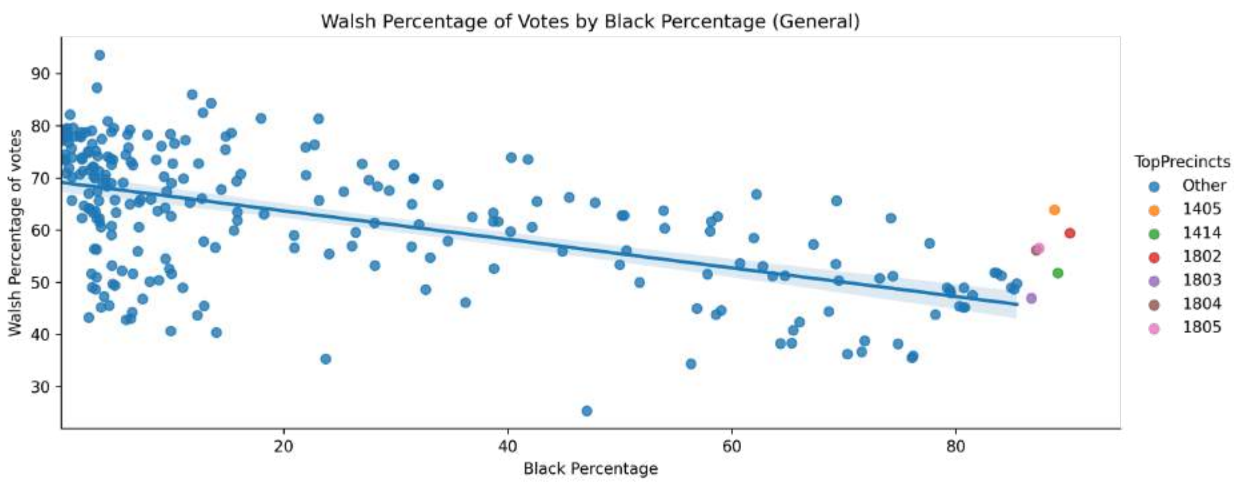
Finding that there is a slight negative correlation (coefficient: -0.25) between the two.

2. For the Mayoral 2017 General Turnout Race, we obtained the following results:



In the general race, there was a slightly stronger positive correlation between Jackson's percentage of votes and the Black population per precinct, with correlation coefficient of 0.58.

Similarly, analyzing Marty Walsh's percentage of votes by Black Percentage:

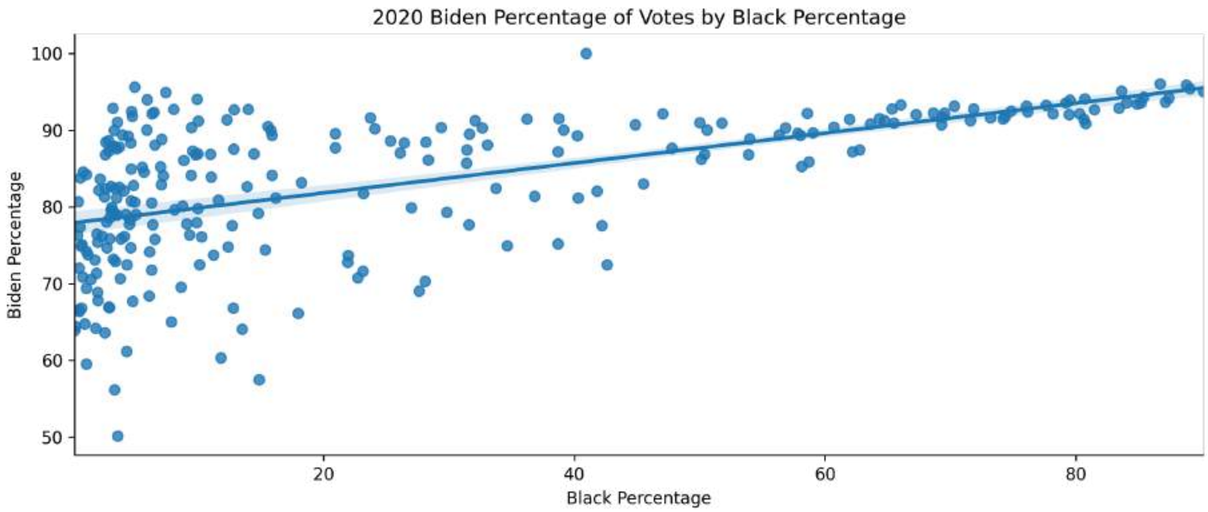


Here, we found a much stronger negative correlation of -0.56.

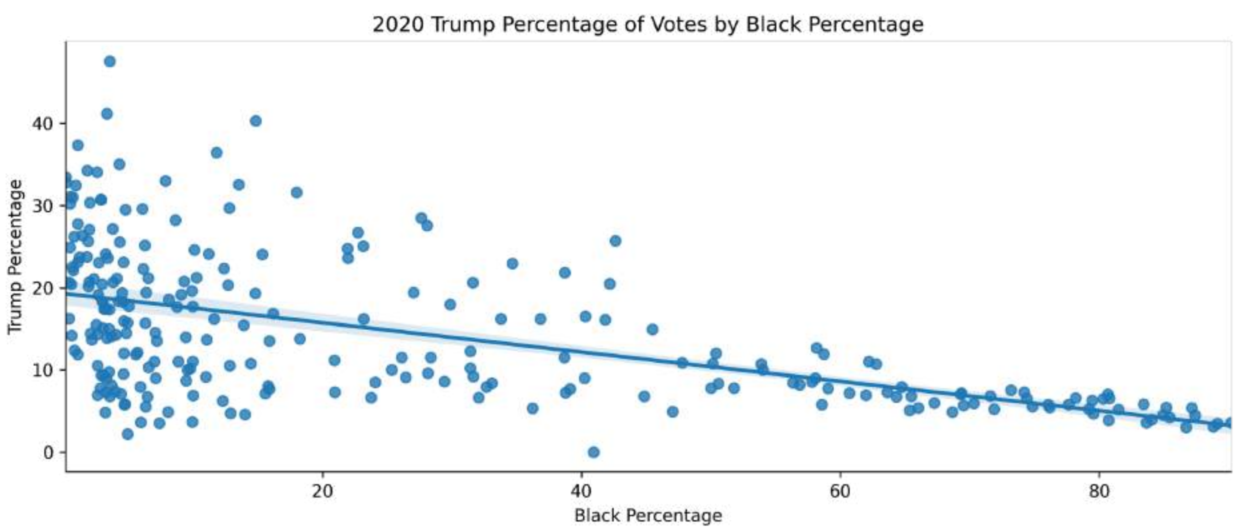
Part 3: Presidential Elections 2020 and 2016 Presidential Elections

1. For the Presidential Elections, we obtained the following results:

First, we analyzed the 2020 Trump/Biden race by Black Percentage per Boston precinct:



Here, we have a correlation coefficient of 0.58, indicating a positive correlation.



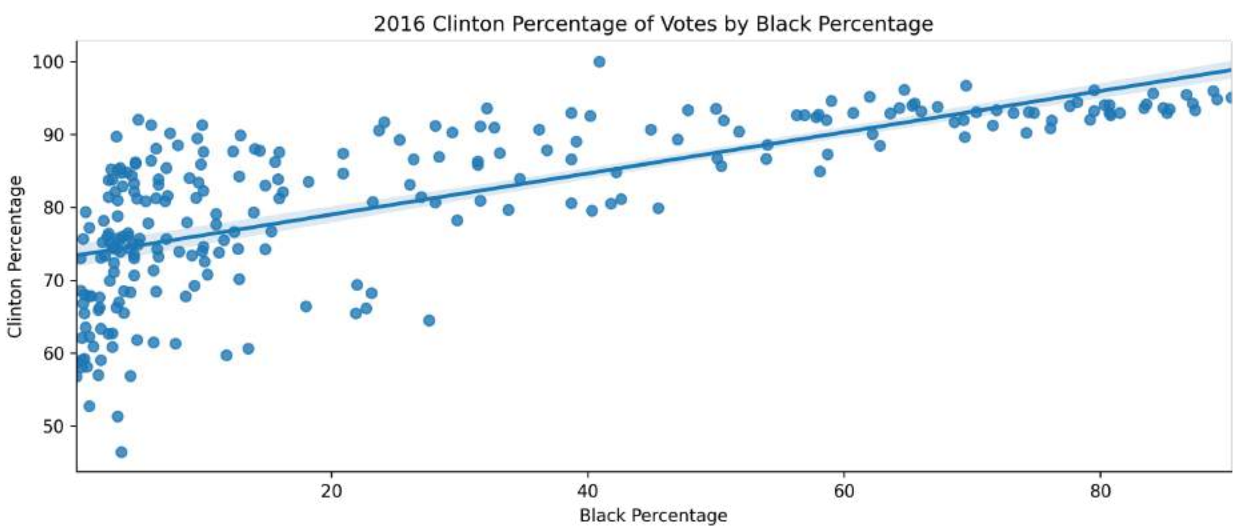
Here, we have a correlation coefficient of -0.55, indicating a negative correlation between Black Percentage per precinct and Trump's percentage of votes.

Sorting our precincts by By Black Population Percentage:

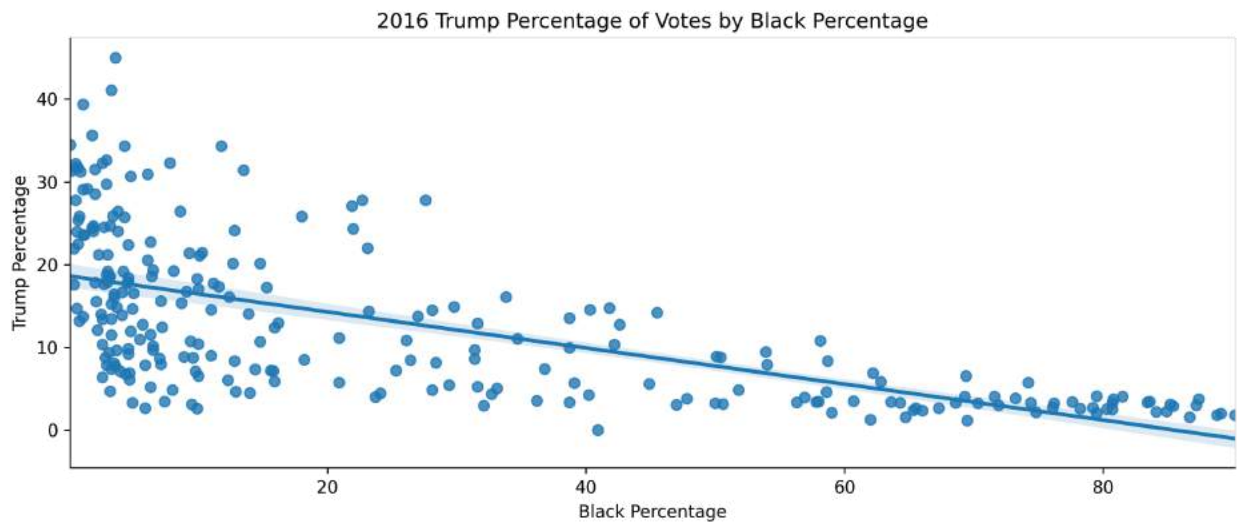
	City/Town	Ward	Pct	Joseph R. Biden, Jr.	Donald J. Trump	Jo Jorgensen	Howard Hawkins	All Others	No Preference	Blanks	...	Two or More Races/Ethnicities (alone)	Black Percentage	Native American Percentage	Asian Percentage	Ha
61	Boston	7	1.0	731	383	12	1	8.0	0.0	9.0	...	3.0	0.146735	0.000000	0.880411	
59	Boston	6	8.0	823	419	19	4	7.0	0.0	5.0	...	10.0	0.194175	0.064725	0.647249	
20	Boston	2	5.0	1,054	286	14	8	10.0	0.0	9.0	...	8.0	0.374532	0.000000	1.373283	
18	Boston	2	3.0	1,054	212	18	0	13.0	0.0	9.0	...	15.0	0.423729	0.181598	1.452785	
58	Boston	6	7.0	948	430	21	0	17.0	0.0	7.0	...	8.0	0.458716	0.065531	0.786370	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
173	Boston	18	4.0	1,161	67	5	2	1.0	0.0	3.0	...	40.0	87.154150	0.247036	0.444664	
174	Boston	18	5.0	1,203	57	5	5	2.0	0.0	5.0	...	37.0	87.415497	1.248050	0.208008	
125	Boston	14	5.0	702	23	2	1	2.0	0.0	2.0	...	28.0	88.811675	0.347464	0.764420	
134	Boston	14	14.0	773	28	4	1	2.0	0.0	2.0	...	29.0	89.111111	0.222222	0.370370	
171	Boston	18	2.0	984	37	1	7	2.0	0.0	4.0	...	43.0	90.180587	0.451467	0.959368	

253 rows × 37 columns

2. Next, we analyzed the 2016 Trump/Clinton race by Black Percentage per Boston precinct:



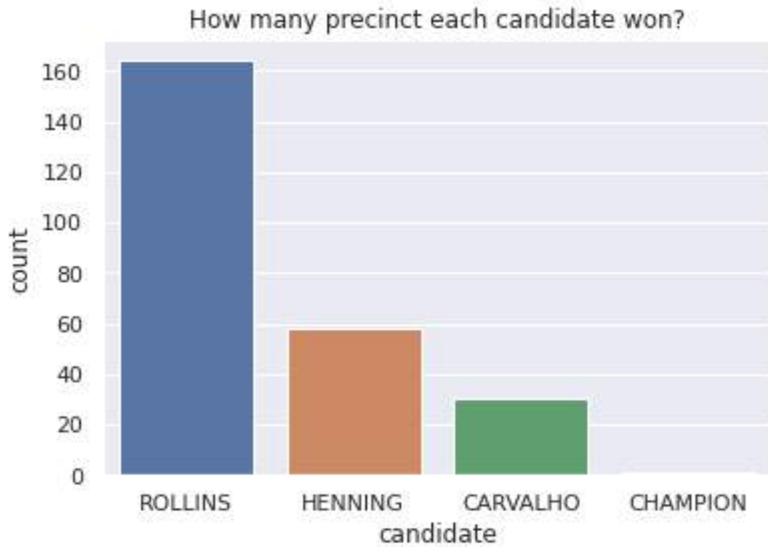
Here, we have a correlation coefficient of 0.71, indicating a strong positive correlation.

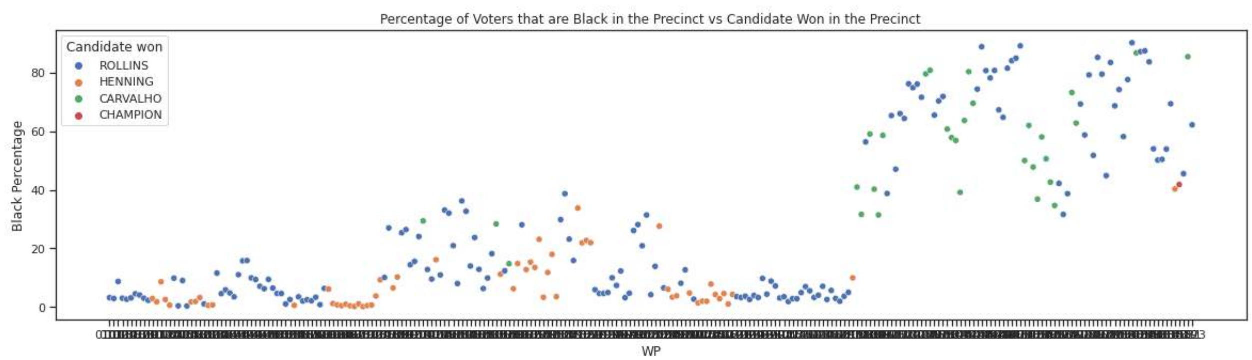


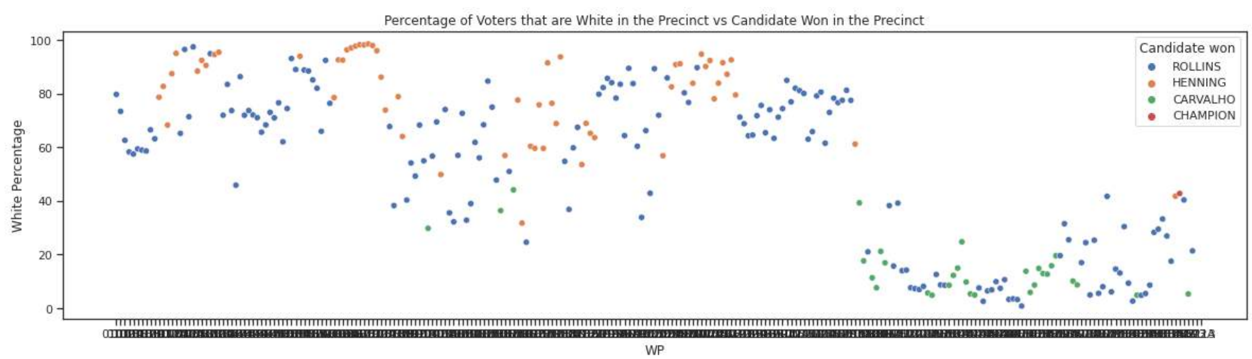
Here we have a correlation coefficient of -0.63, indicating a strong negative correlation.

Part 4: DA Race Turnout 2018

1. For the DA Race Turnout 2018, we obtained the following results:





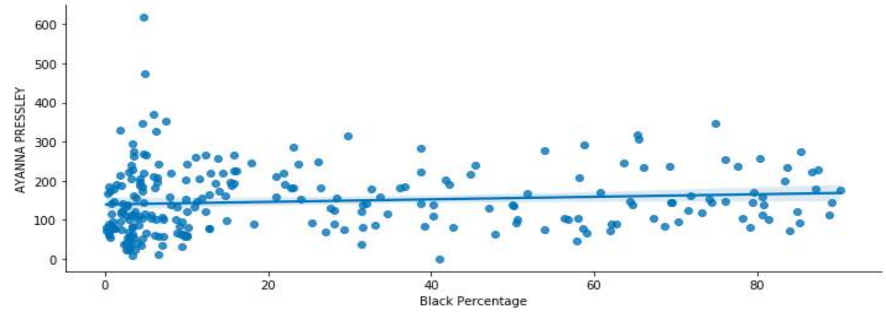


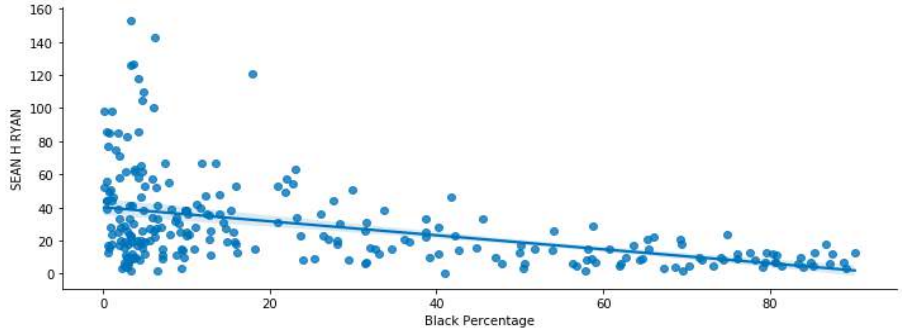
For precincts with a high percentage of voters that are black, the candidate, Rolling, seems to be favored more than other candidates. Henning, who also won a lot of precincts, doesn't seem to be favored by black people as most of his winning precincts are from precincts with a dense white population.

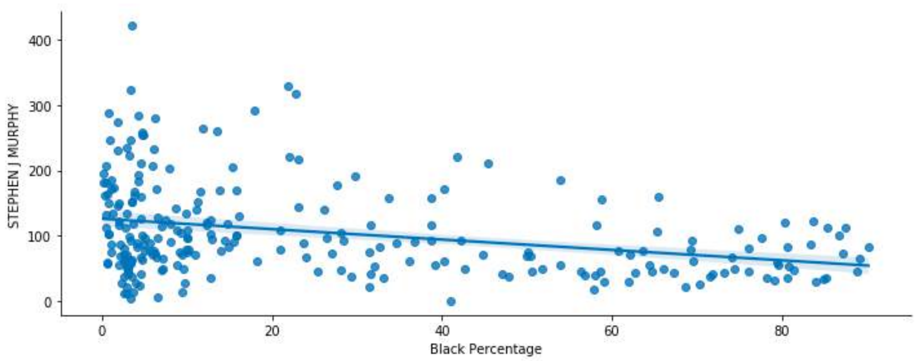
Part 5: City Council Result Analysis including 2011, 2013, 2015, 2017 and 2019

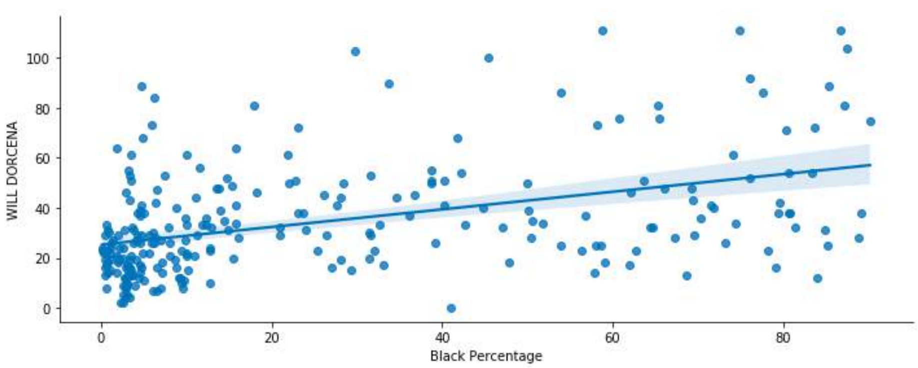
1. 2011 City Council Result

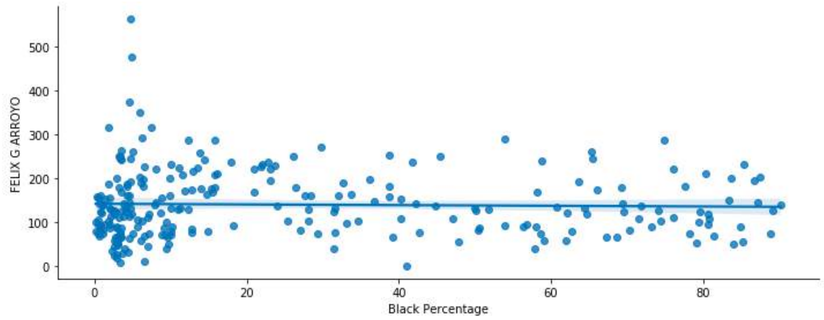
- Individual vote counts vs black percentage

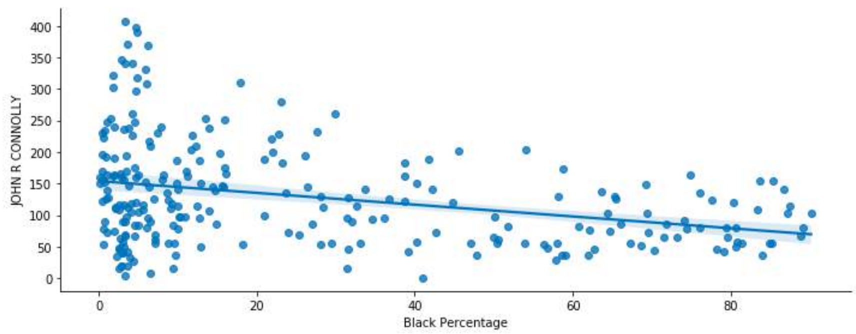


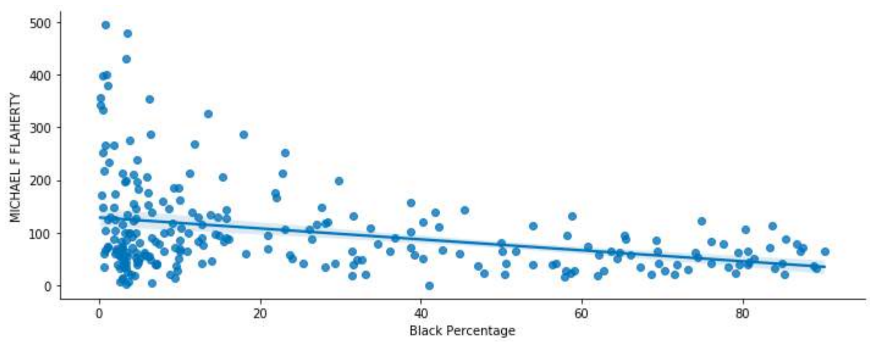






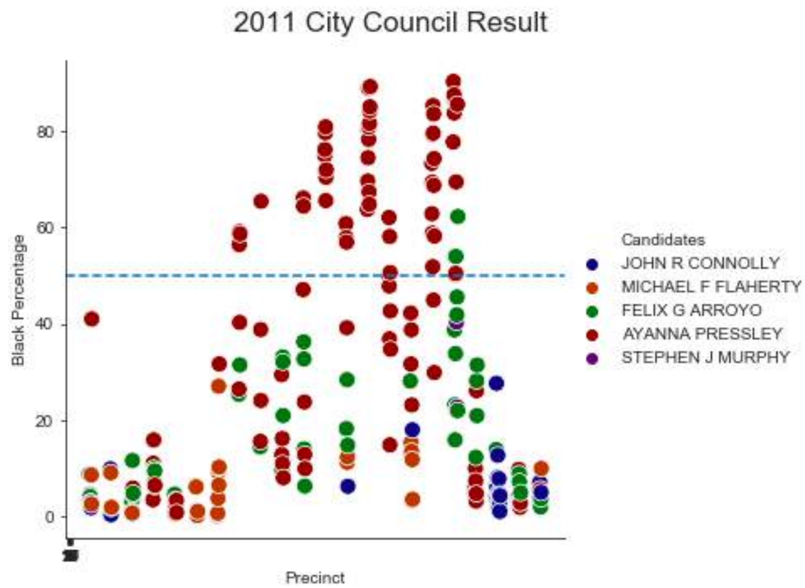






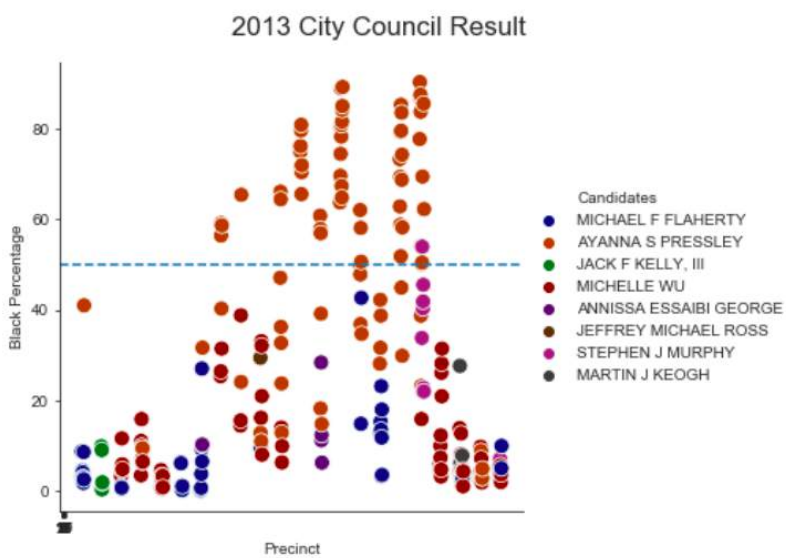
From the plot, we can see that Will Dorcena has the strongest positive correlation between black percentage and vote counts. Ayanna Pressley has a slight positive correlation. John R Connoly, Michael F Flaherty and Sean H Ryan all have negative correlation. According to Wikipedia, Ayanna Pressley is a black woman and Will Dorcena is a black man. The rest of the candidates are white men. Thus, the finding is not surprising and it confirms the hypothesis that black voters are more likely to vote black candidates.

- Winner in each precinct vs black percentage



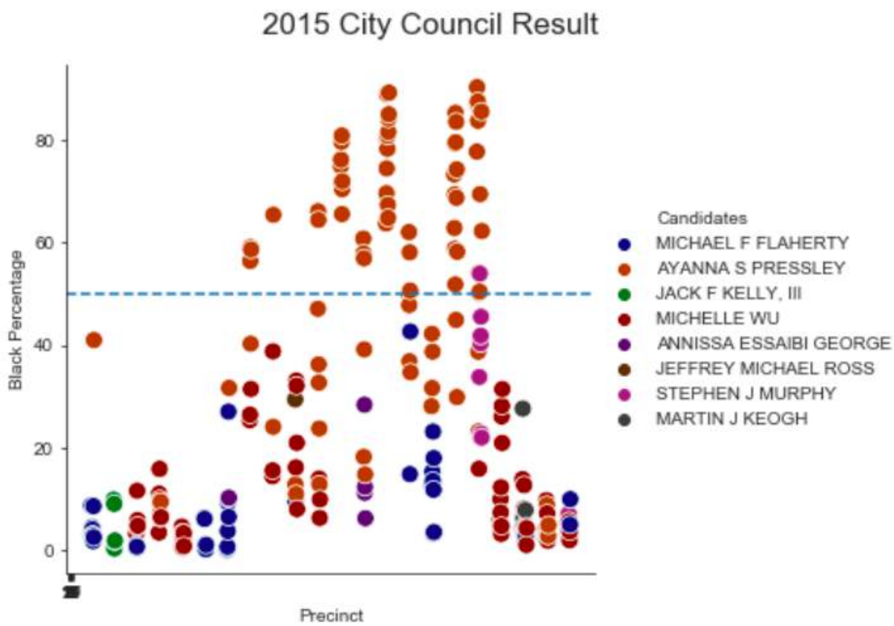
From the graph we can see that Ayanna Pressley is leading in precincts that have large black percentages. Felix, who has similar vote counts with Ayanna, his supporters are mostly in neighborhoods that have large white populations.

2. 2013 City Council Result General Analysis



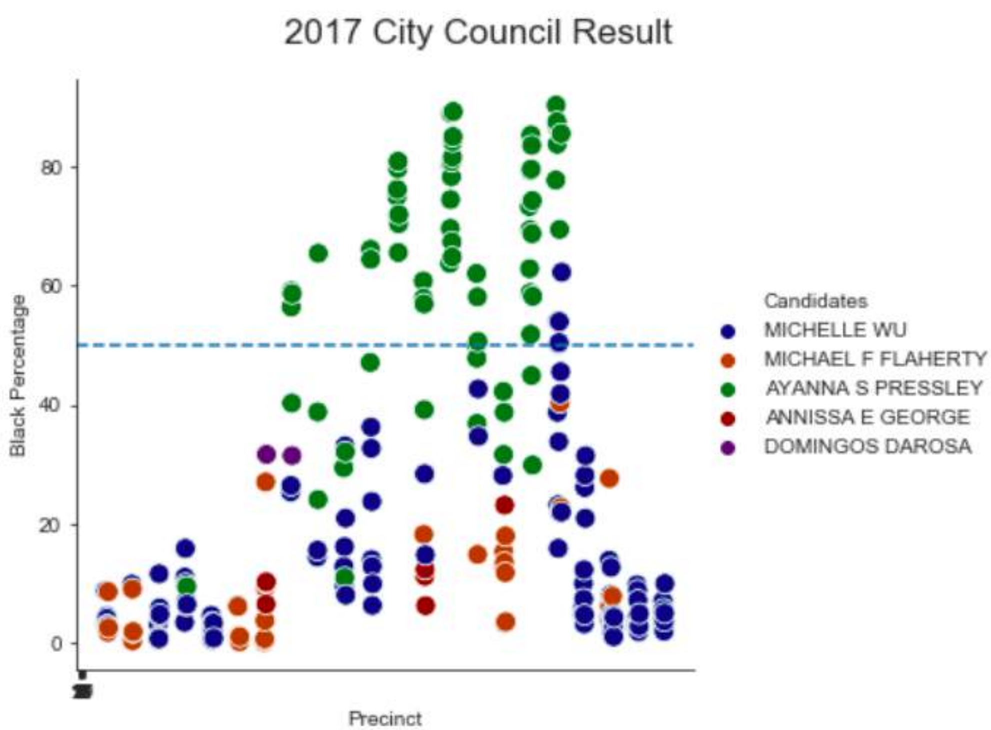
We can see that Ayanna Pressley, the only black candidate, is still leading in precincts that have large black percentages.

3. 2015 City Council Result General Analysis



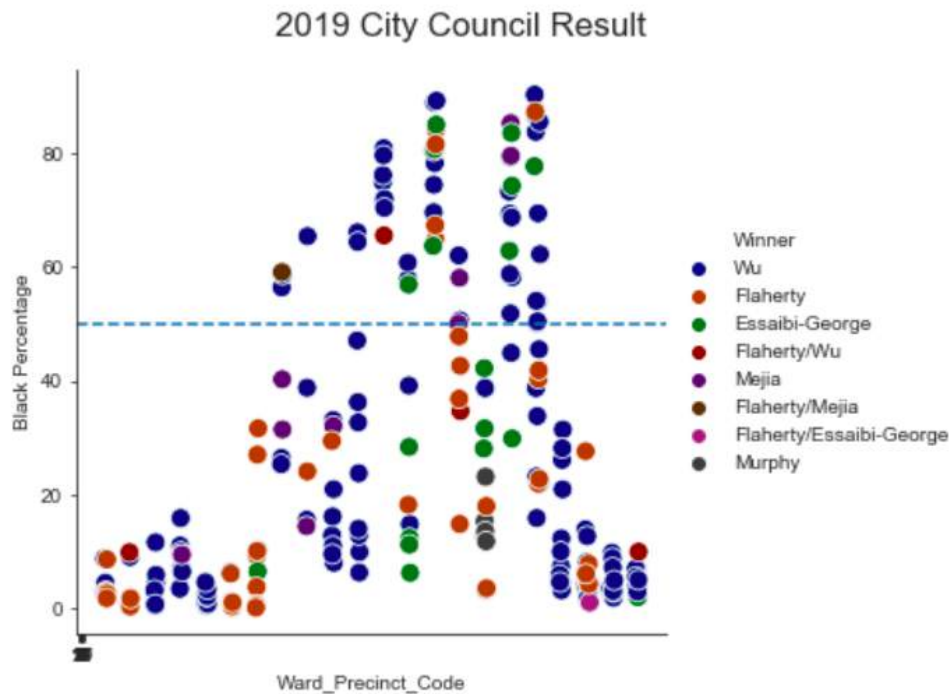
We can see that Ayanna Pressley, the only black candidate, is still leading in precincts that have large black percentages.

4. 2017 City Council Result General Analysis



We can see that Ayanna Pressley, the only black candidate, is still leading in precincts that have large black percentages.

5. 2019 City Council Result General Analysis

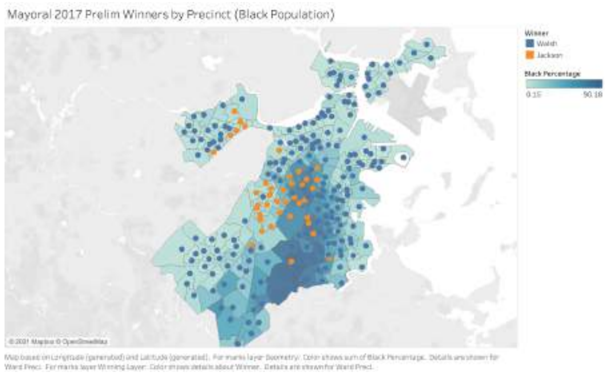


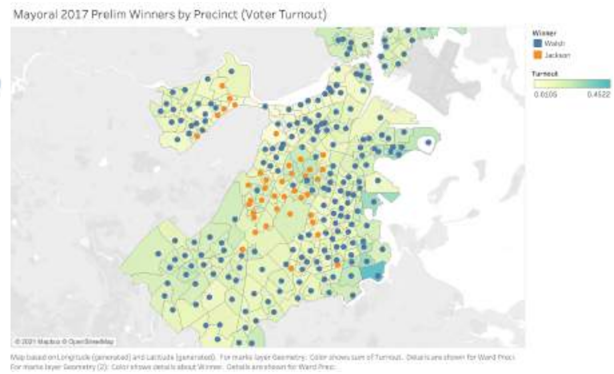
In 2019, there is no black candidate, so the winner in precincts that have a large black population is pretty random.

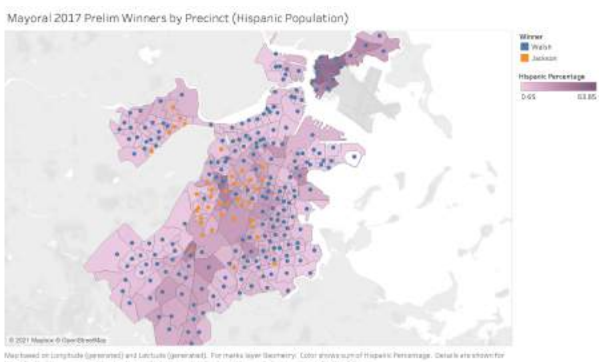
Visualizing on Boston Maps How Racial Voting Patterns Differ by Election Type

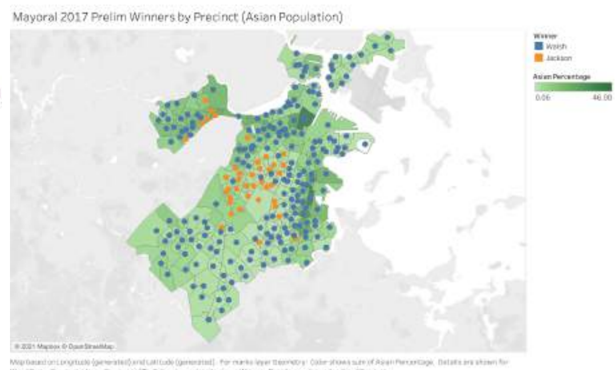
Mayoral 2017 Elections

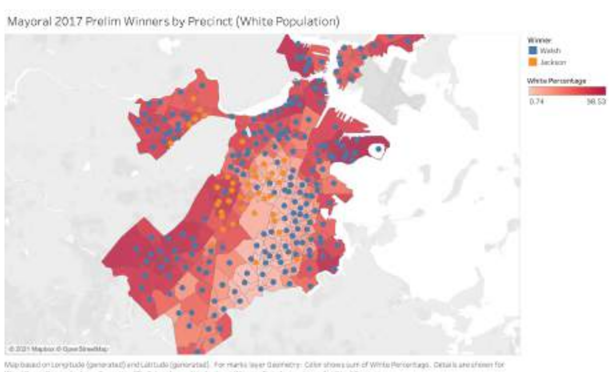
Preliminary Race:









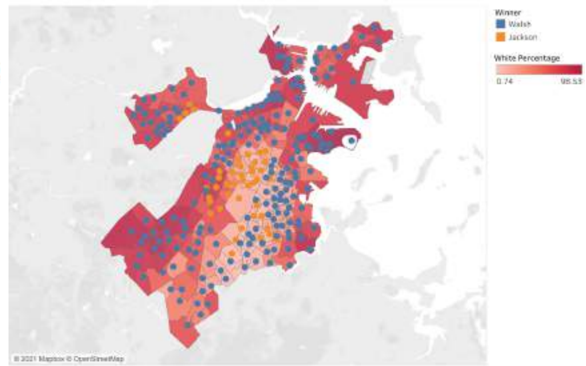


General Race:

Mayoral 2017 General Winners by Precinct (Black Population)

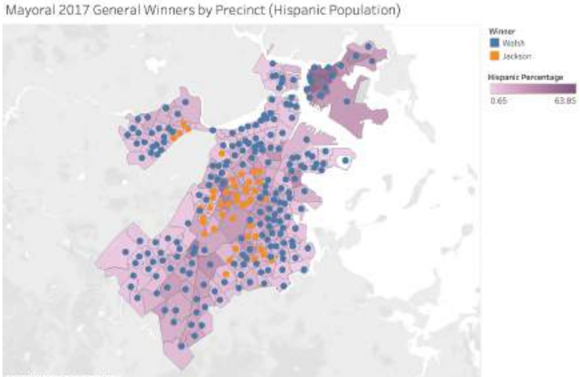
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Mayoral 2017 General Winners by Precinct (White Population)



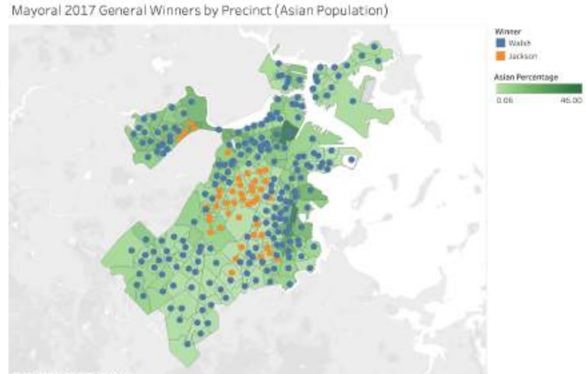
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Mayoral 2017 General Winners by Precinct (Hispanic Population)



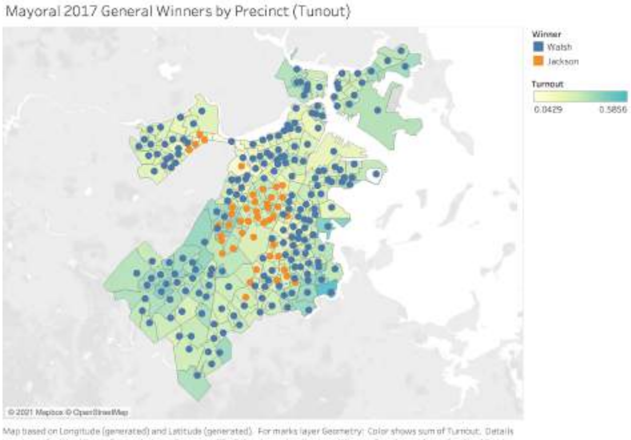
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Mayoral 2017 General Winners by Precinct (Asian Population)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Mayoral 2017 General Winners by Precinct (Tunout)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

2019 City Council Analysis

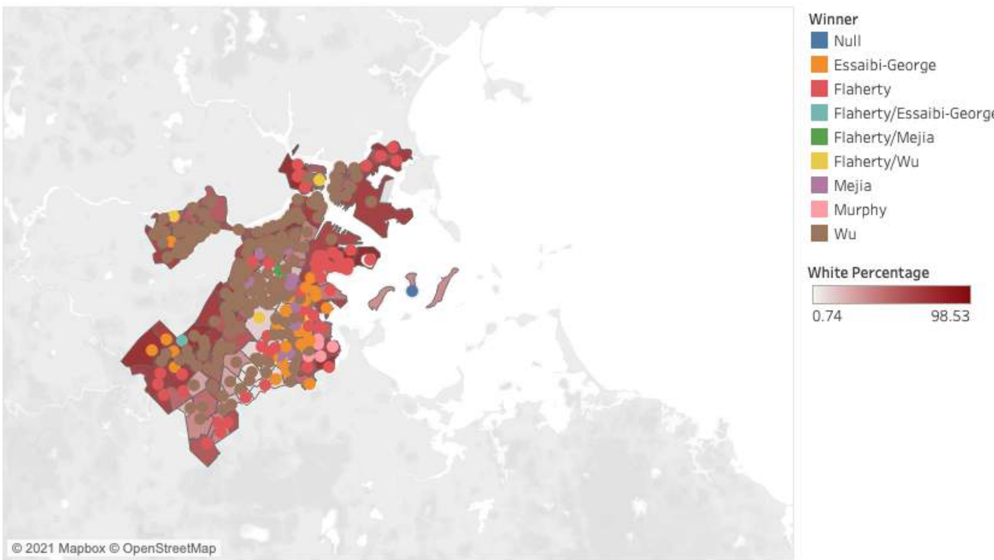
Winner and Black Percentage - 2019 cc

Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and Hispanic Percentage - 2019 cc

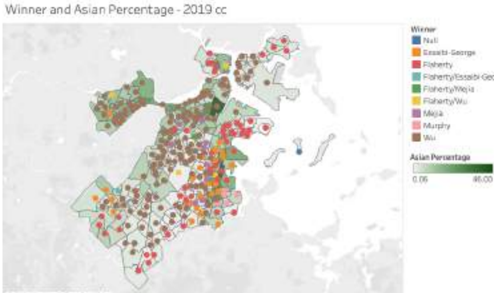
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and White Percentage - 2019 cc



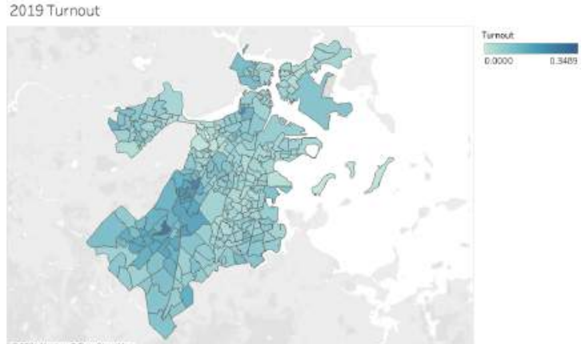
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and Asian Percentage - 2019 cc



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

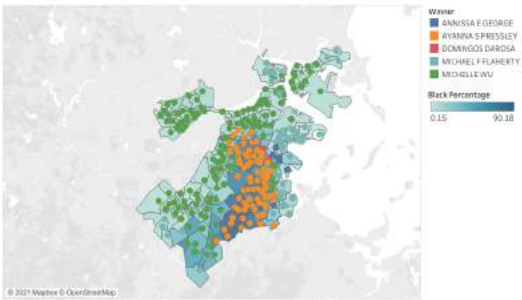
2019 Turnout



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Turnout. Details are shown for Ward Preci.

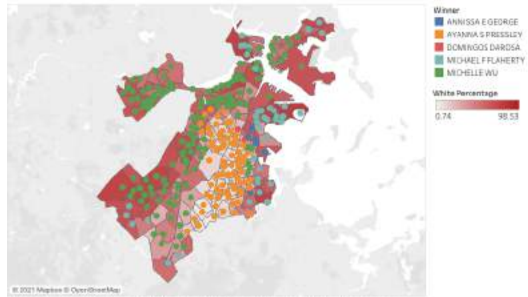
2017 City Council Analysis

Winner and Black Percentage - 2017 cc



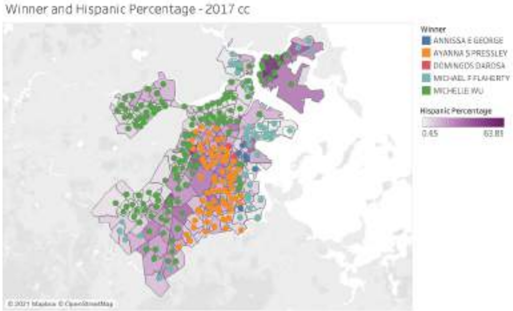
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and White Persentage - 2017 cc



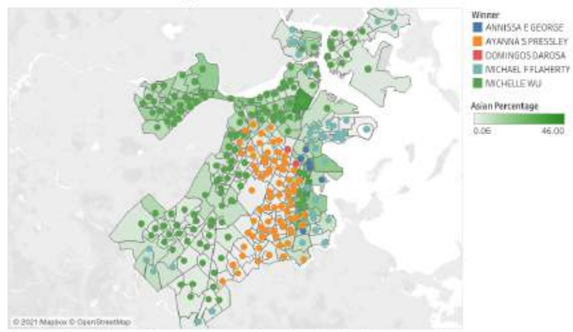
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and Hispanic Percentage - 2017 cc



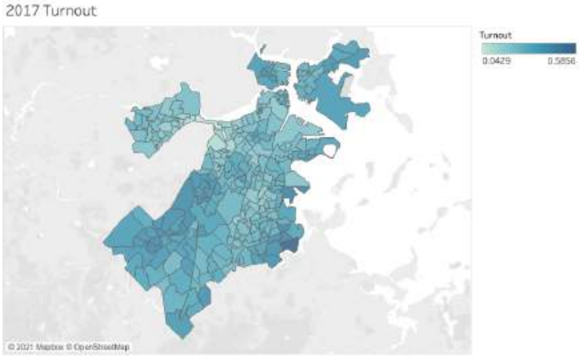
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

Winner and Asian Percentage - 2017 cc



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci.

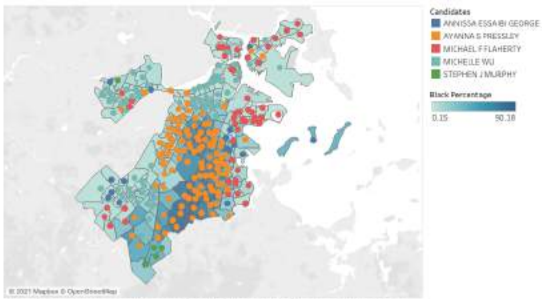
2017 Turnout



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Turnout. Details are shown for Ward Preci.

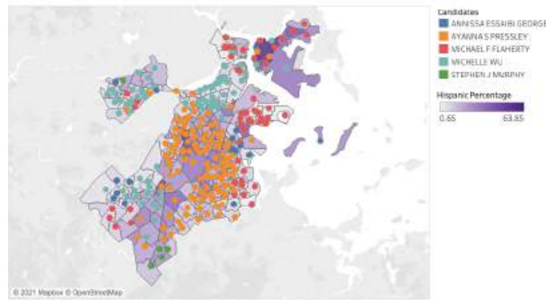
2015 City Council Analysis

Winner and Black Percentage - 2015 cc



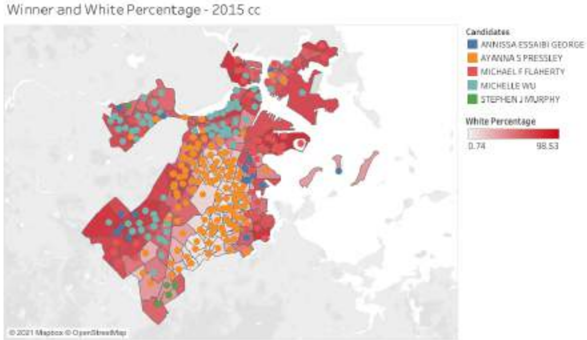
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Candidates. Details are shown for Ward Preci.

Winner and Hispanic Percentage - 2015 cc



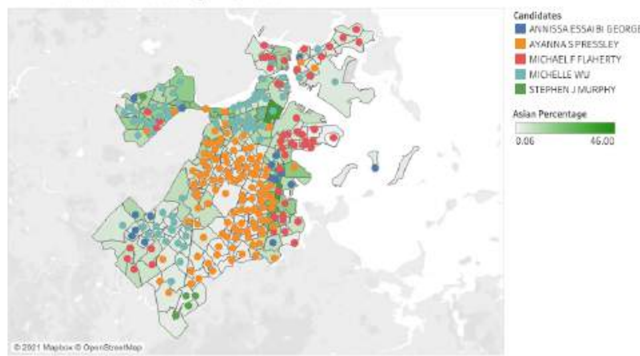
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Candidates. Details are shown for Ward Preci.

Winner and White Percentage - 2015 cc



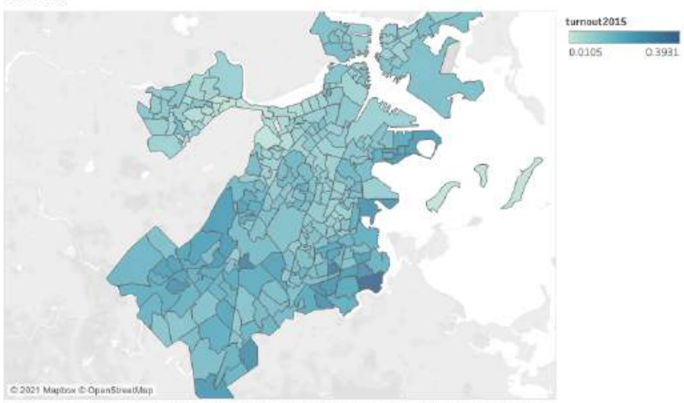
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Candidates. Details are shown for Ward Preci.

Winner and Asian Percenage - 2015 cc



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Candidates. Details are shown for Ward Preci.

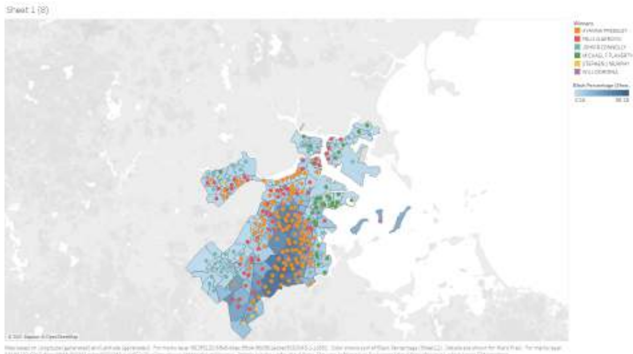
Sheet 5



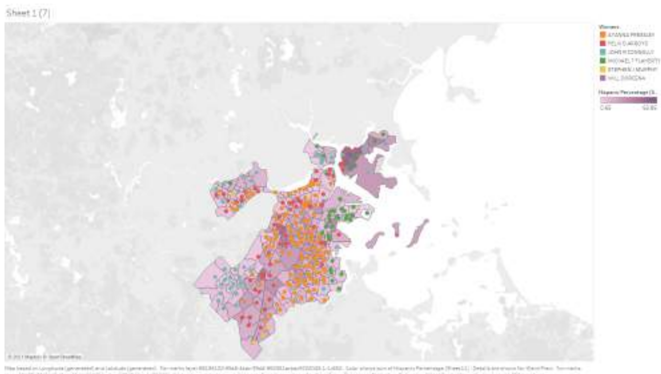
Map based on Longitude (generated) and Latitude (generated). Color shows sum of turnout2015. Details are shown for Ward Preci.

<u>2013 City Council Analysis</u>

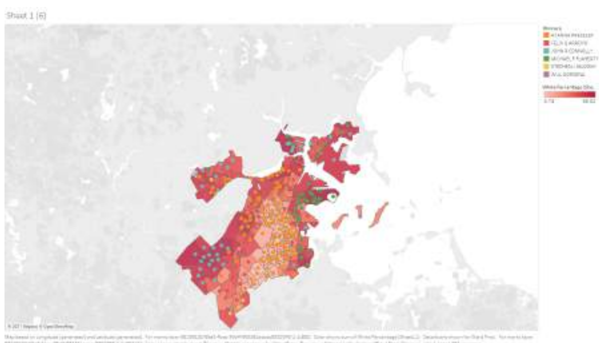
Black Percentage & Candidates



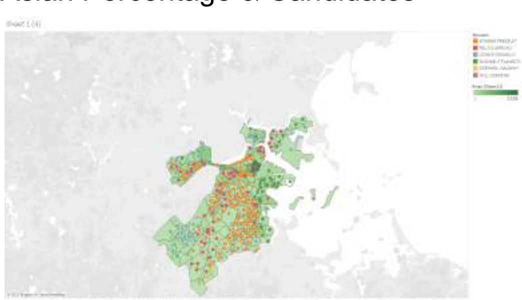
Hispanic Percentage & Candidates



White Percentage & Candidates

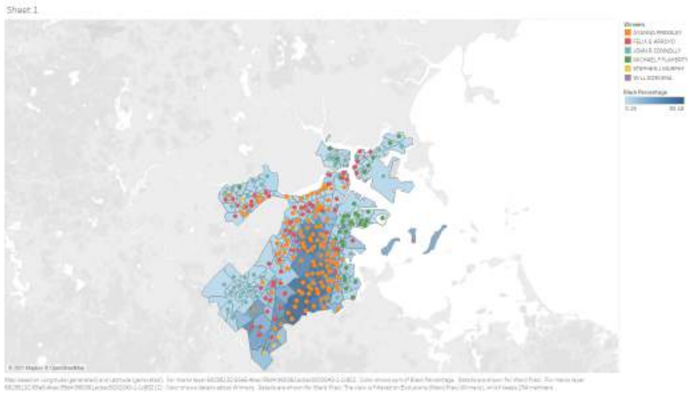


Asian Percentage & Candidates

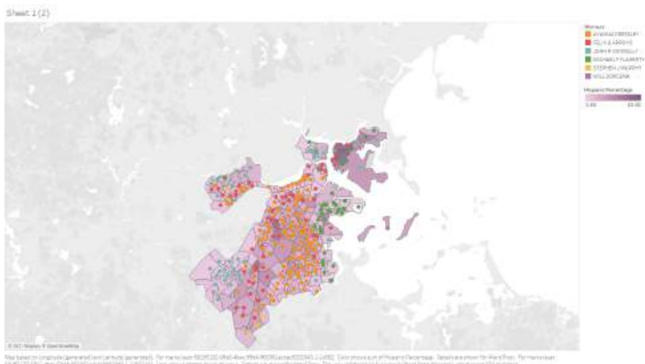


<u>2011 City Council Analysis</u>

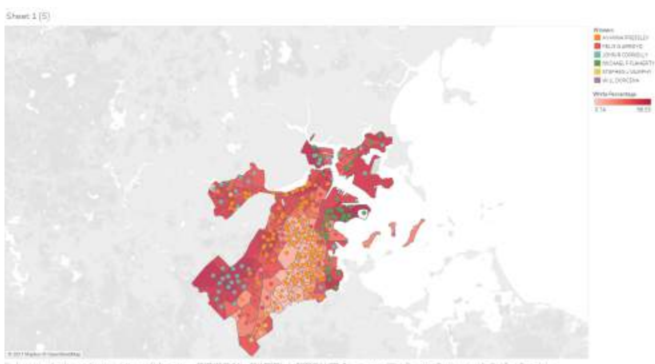
Black Percentage & Candidates



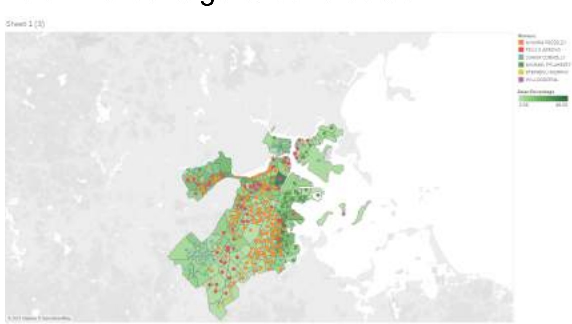
Hispanic Percentage & Candidates



White Percentage & Candidates

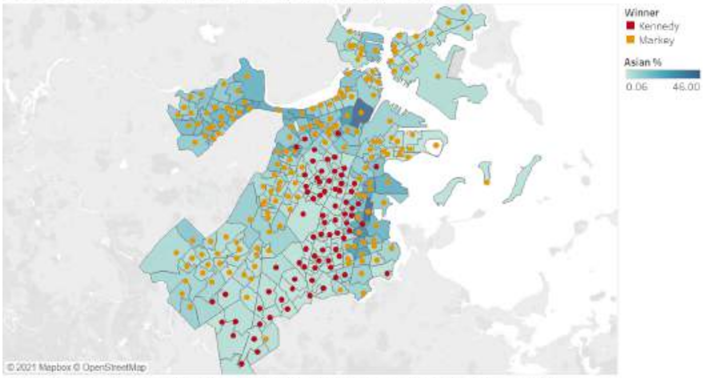


Asian Percentage & Candidates



<u>2020 US Senate Democratic Primary</u>

2020 US Senate Democratic Primary Winners By Precinct

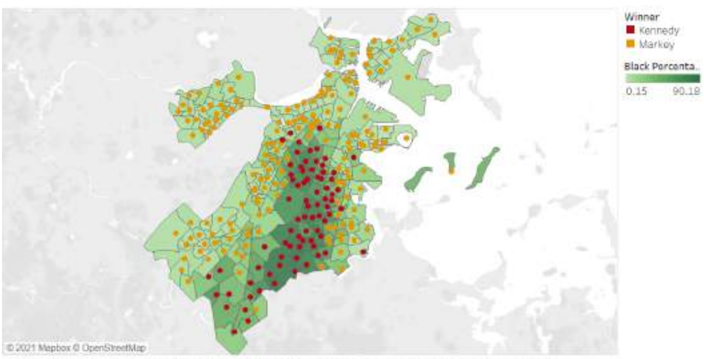


Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci. The view is filtered on Ward Preci, which keeps 255 of 255 members.

2020 US Senate Democratic Primary Winners By Precinct - (White Population)

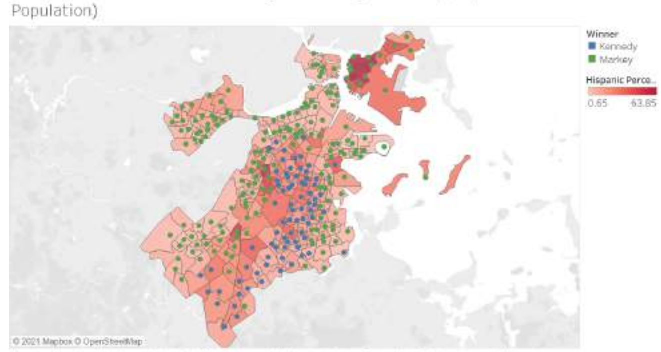
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci. The view is filtered on Winner, which keeps Kennedy and Markey.

2020 US Senate Democratic Primary Winners By Precinct - (Black Population)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci. The view is filtered on Winner, which keeps Kennedy and Markey.

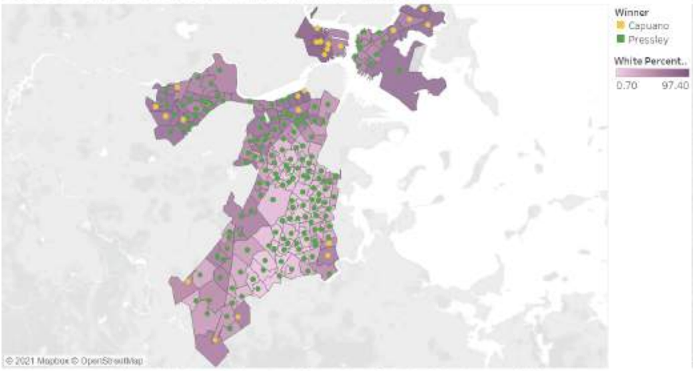
2020 US Senate Democratic Primary Winners By Precinct - (Hispanic Population)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Winner. Details are shown for Ward Preci. The view is filtered on Winner, which keeps Kennedy and Markey.

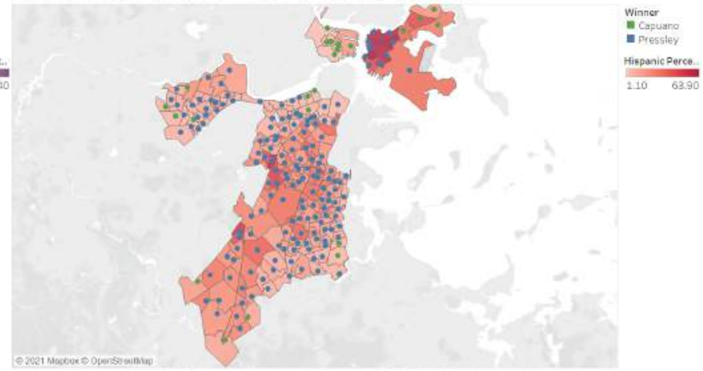
2018 US House Democratic Primary

2018 US House Democratic Primary - (White Population)



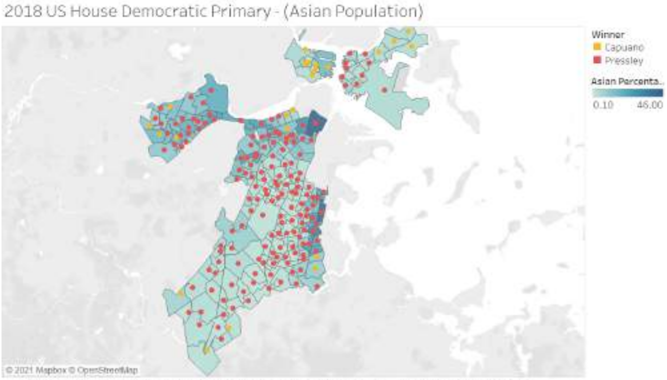
Map based on Longitude (generated) and Latitude (generated). For marks layer Pcnts.shp: Color shows sum of White Percentage. Details are shown for Ward Prec. For marks layer Pcnts.shp (2): Color shows details about Winner. Details are shown for Ward Prec. The view is filtered on Winner, which keeps Capuano and Pressley.

2018 US House Democratic Primary - (Hispanic Population)



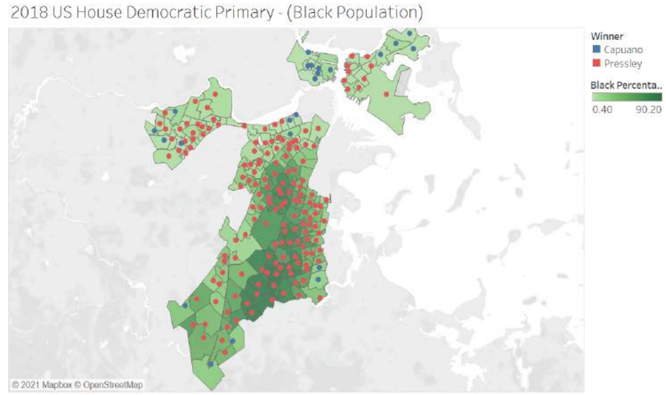
Map based on Longitude (generated) and Latitude (generated). For marks layer Pcnts.shp: Color shows sum of Hispanic Percentage. Details are shown for Ward Prec. For marks layer Pcnts.shp (2): Color shows details about Winner. Details are shown for Ward Prec. The view is filtered on Winner, which keeps Capuano and Pressley.

2018 US House Democratic Primary - (Asian Population)



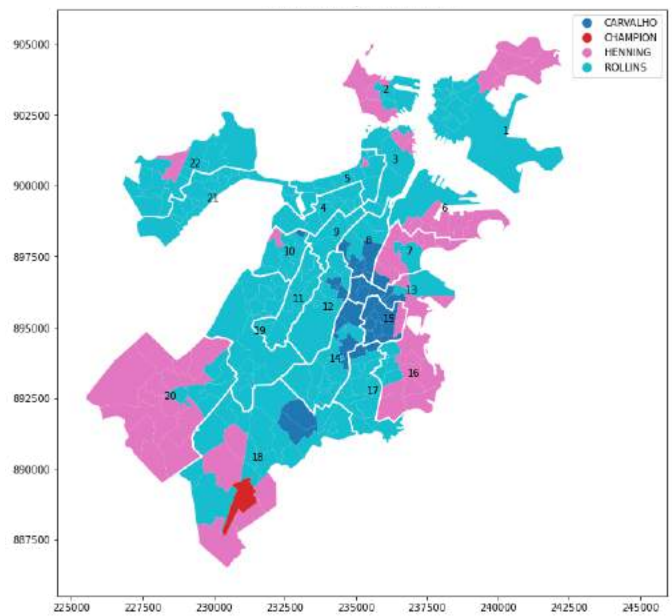
Map based on Longitude (generated) and Latitude (generated). For marks layer Pcnts.shp: Color shows sum of Asian Percentage. Details are shown for Ward Prec. For marks layer Pcnts.shp (2): Color shows details about Winner. Details are shown for Ward Prec. The view is filtered on Winner, which keeps Capuano and Pressley.

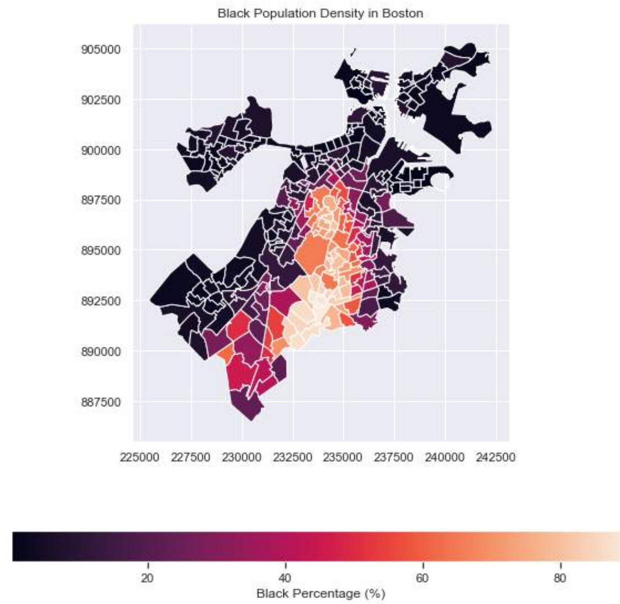
2018 US House Democratic Primary - (Black Population)

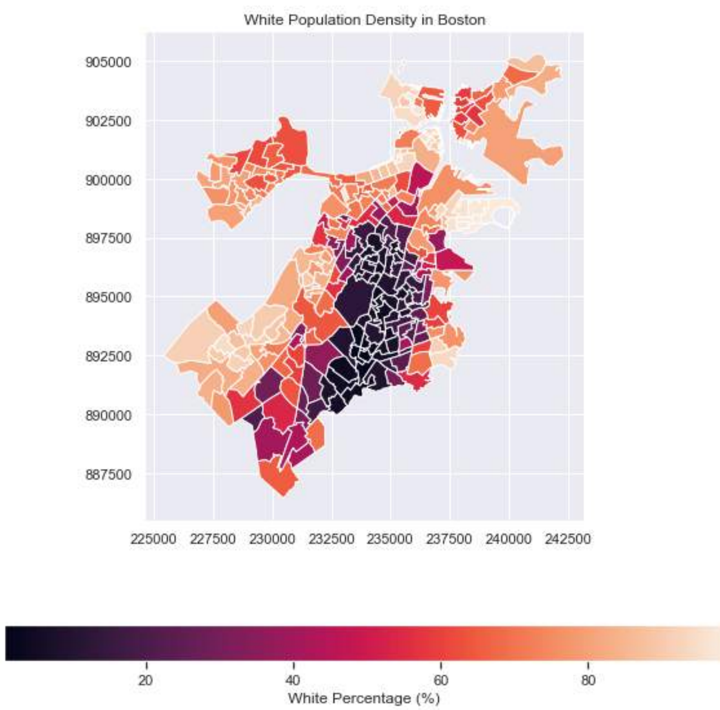


Map based on Longitude (generated) and Latitude (generated). For marks layer Pcnts.shp: Color shows sum of Black Percentage. Details are shown for Ward Prec. For marks layer Pcnts.shp (2): Color shows details about Winner. Details are shown for Ward Prec. The view is filtered on Winner, which keeps Capuano and Pressley.

DA Race 2018


CARVALHO
CHAMPION
HENNING
ROLLINS
905000
902500
900000
897500
895000
892500
890000
887500
225000
227500
230000
232500
235000
237500
240000
242500
245000


Black Population Density in Boston
905000
902500
900000
897500
895000
892500
890000
887500
225000 227500 230000 232500 235000 237500 240000 242500
20
40
60
80
Black Percentage (%)


White Population Density in Boston
905000
902500
900000
897500
895000
892500
890000
887500
225000 227500 230000 232500 235000 237500 240000 242500
20
40
60
80
White Percentage (%)

City Council Turnout Analysis

We examine the change in voter turnout over the various CC election years (2011, 2013, 2015, 2017, and 2019). We are answering an essential question: **How has voter turnout by precinct changed across city council election year**?

Part 1: “Volatility” of Voter Turnout

To begin, we examine the volatility of voter turnout in each election year. Here we are examining the voter turnout of each specific precinct, that is, the percentage of registered voters in a specific precinct who cast a ballot in the election.

First finding the mean:

```
Mean voter turnout 2011: 18.3 %
Mean voter turnout 2013: 38.5 %
Mean voter turnout 2015: 13.9 %
Mean voter turnout 2017: 28.3 %
Mean voter turnout 2019: 9.0 %
```

As we can see, 2013 has the highest average voter turnout.

Then the median:

```
Median voter turnout 2011: 17.0 %
Median voter turnout 2013: 37.3 %
Median voter turnout 2015: 13.5 %
Median voter turnout 2017: 27.8 %
Median voter turnout 2017: 7.6 %
```

Again, 2013 is the highest.

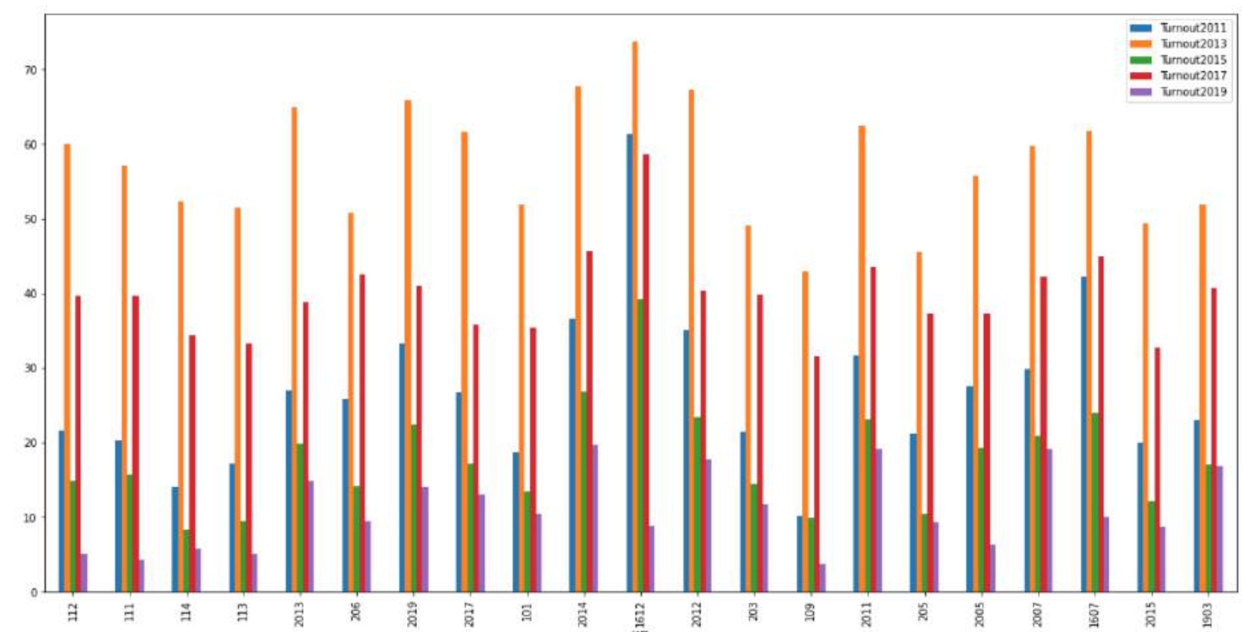
Finding that 2013 and 2017 had the highest voter turnout rates across all precincts due to the fact that they coincided with the Mayoral Elections of those same years.

Next, we determine the “Top 20 Precincts” which have the most volatile voter turnout. That is, have the greatest average change in voter turnout in that specific precinct based on election year. Listed in descending order:

WP	Turnout2011	Turnout2013	Turnout2015	Turnout2017	Turnout2019	AvgChange
112	21.5	60.1	14.8	39.7	5.1	35.8
111	20.3	57.0	15.7	39.7	4.3	34.4
114	14.0	52.4	8.3	34.4	5.7	34.3
113	17.1	51.5	9.5	33.2	5.1	32.1
2013	27.0	64.9	19.8	38.8	14.9	31.5
206	25.8	50.8	14.1	42.5	9.4	30.8
2019	33.2	65.9	22.4	40.9	14.0	30.4
2017	26.7	61.6	17.2	35.8	13.0	30.2
101	18.7	52.0	13.5	35.4	10.5	29.6
2014	36.5	67.8	26.9	45.7	19.7	29.2
1612	61.3	73.8	39.3	58.6	8.9	29.0
2012	35.1	67.3	23.4	40.4	17.7	29.0
203	21.4	49.1	14.4	39.8	11.7	29.0
109	10.2	43.0	9.9	31.5	3.8	28.8
2011	31.7	62.5	23.1	43.5	19.1	28.8
205	21.1	45.5	10.4	37.3	9.3	28.6
2005	27.5	55.8	19.3	37.2	6.3	28.4
2007	29.8	59.7	20.8	42.3	19.2	28.4
1607	42.2	61.8	24.0	44.9	10.0	28.3
2015	20.0	49.4	12.2	32.7	8.7	27.8
1903	23.0	51.9	17.0	40.7	16.9	27.8

Simply from looking at the raw data, we can see that Ward 1 (East Boston), sees a lot of volatility.

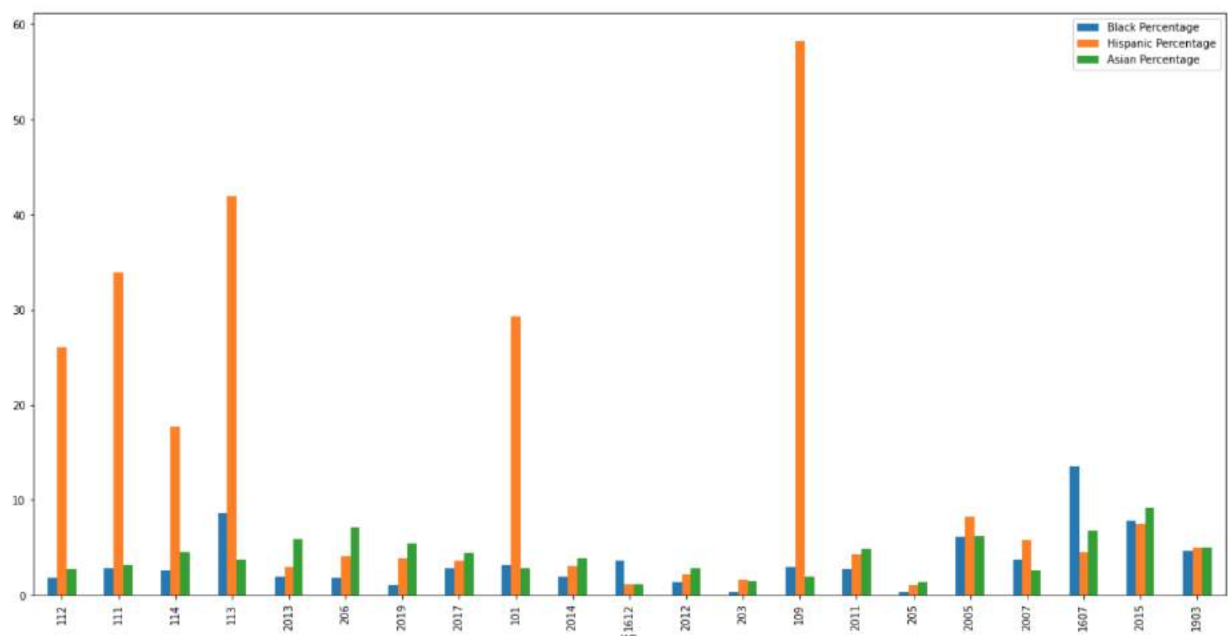
Visualizing this change across elections on a bar chart, we have:



We can clearly see that voter turnout is highly volatile across each of these precincts.
Listing the demographic breakdown of each of these precincts:

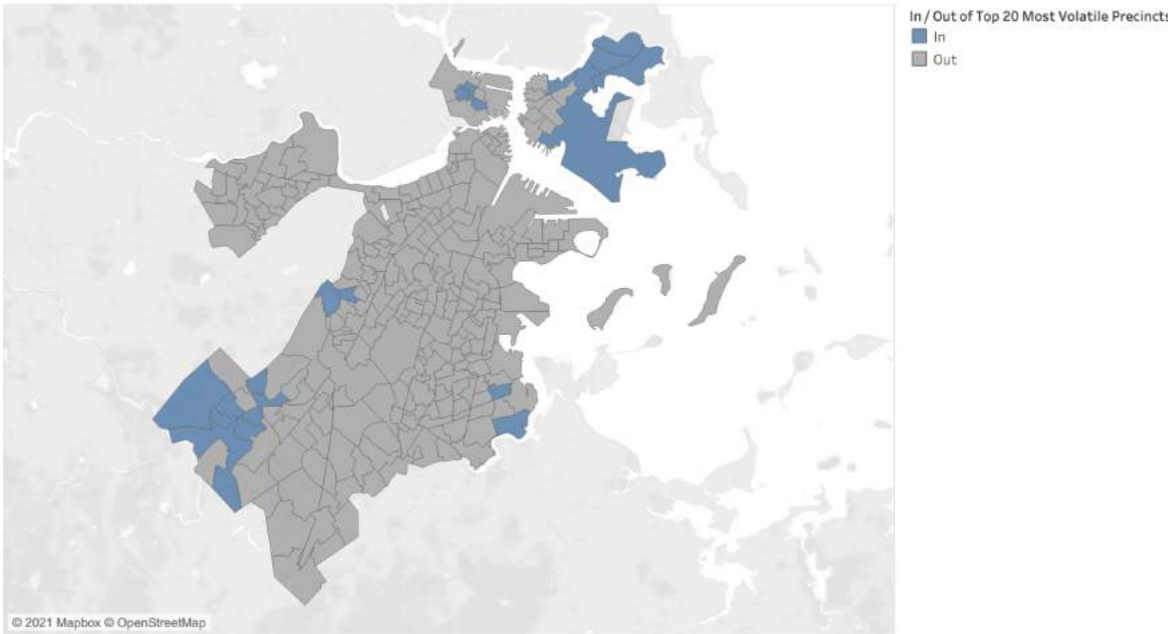
WP	Black Percentage	Hispanic Percentage	Asian Percentage
112	1.8	26.0	2.7
111	2.9	33.9	3.2
114	2.6	17.7	4.5
113	8.6	41.9	3.8
2013	2.0	3.0	5.9
206	1.8	4.1	7.1
2019	1.1	3.9	5.4
2017	2.9	3.6	4.4
101	3.2	29.3	2.9
2014	2.0	3.1	3.9
1612	3.6	1.2	1.2
2012	1.4	2.2	2.9
203	0.4	1.6	1.5
109	3.0	58.3	2.0
2011	2.7	4.3	4.9
205	0.4	1.1	1.4
2005	6.1	8.3	6.2
2007	3.8	5.8	2.6
1607	13.5	4.5	6.8
2015	7.8	7.5	9.1
1903	4.7	5.0	5.0

Now, we do a demographic breakdown of each of these precincts, finding the following:



And viewing these precincts on a Boston map, we again see the concentration of precincts in East Boston as well as a concentration in West Roxbury:

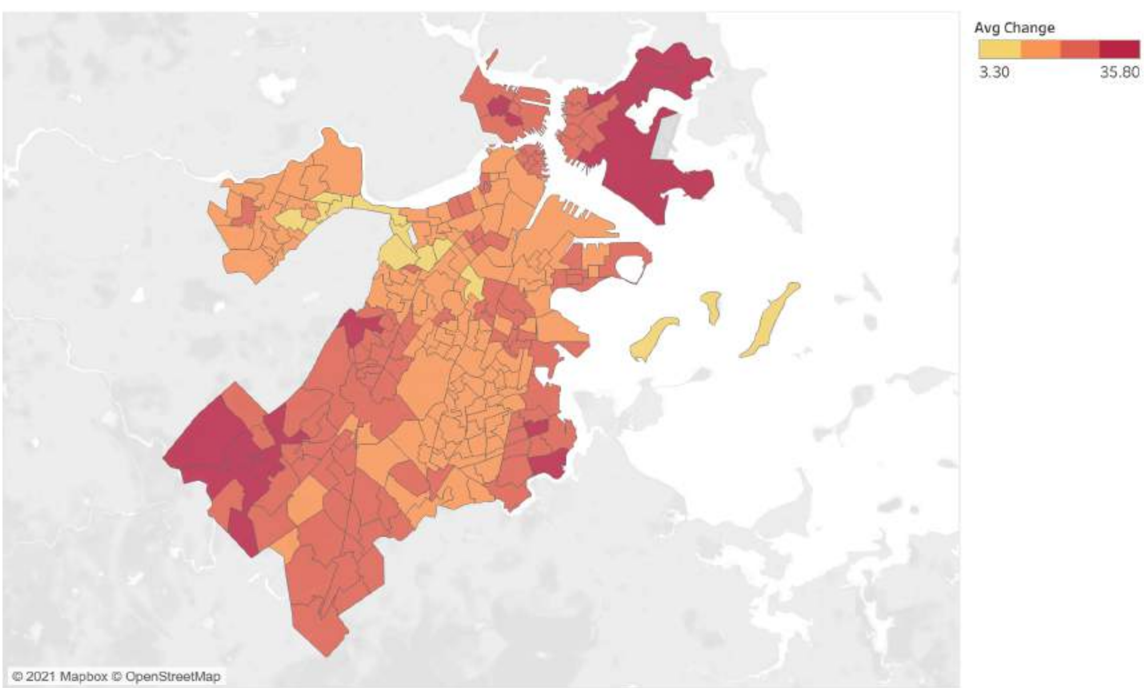
Top 20 Most Volatile Precincts in terms of Voter Turnout (2011-2019)



Map based on Longitude (generated) and Latitude (generated). Color shows details about In / Out of Top 20 Most Volatile Precincts. Details are shown for Ward Preci.

Additionally, we can use a heat map to visualize which districts experience the most volatility across city council elections:

Volatility Heat Map: Average Change in Voter Turnout in City Council Elections from 2011 to 2019



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Avg Change. Details are shown for Ward Preci.

Part 2: Changes in Share of Voter Turnout

We now analyze which precincts experienced the greatest average change in *share of voter turnout* over time. That is, we are analyzing what percentage of the total votes cast for this particular election is from each precinct, and then analyzing the precincts which experienced the greatest change.

Metric: For each precinct, we calculate the percent share of votes: [Number of Ballots Cast in Precinct X]/[Total Ballots Cast in CC Election YYYY] * 100.

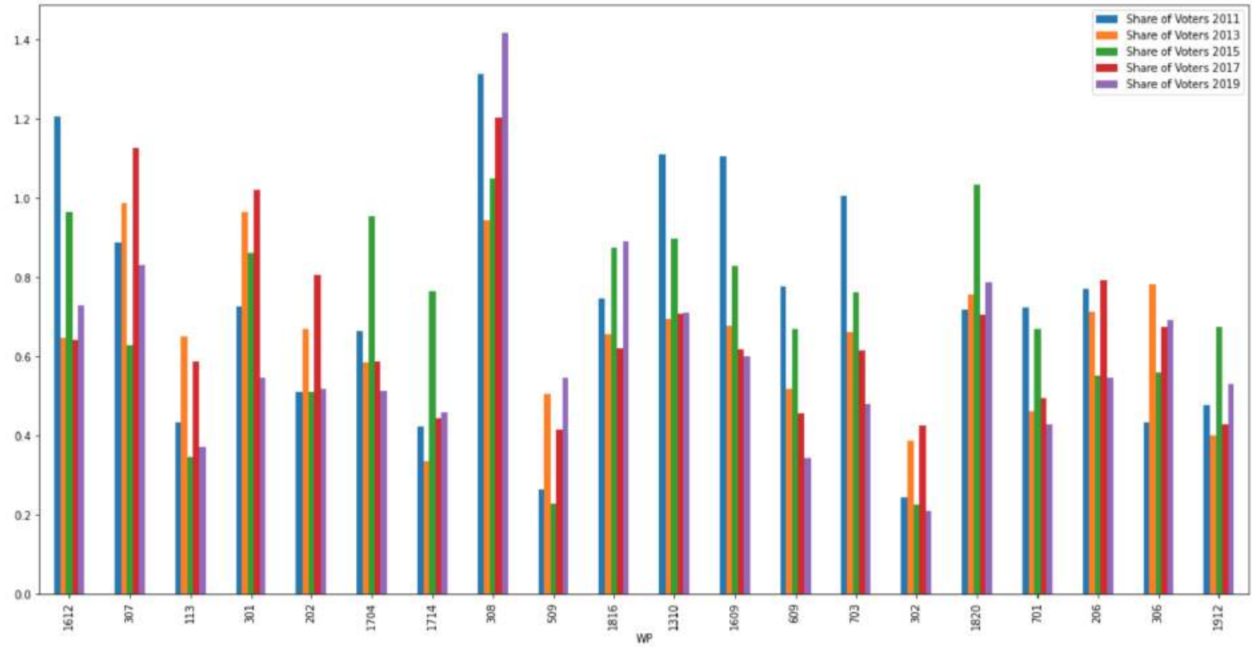
First, we find the top 20 precincts with the greatest average change in share of voters across each election. Note that these top 20 could be experiencing either significant negative or positive change.

Looking at the raw data:

WP	Share of Voters 2011	Share of Voters 2013	Share of Voters 2015	Share of Voters 2017	Share of Voters 2019	AvgChange
1612	1.20712	0.64750	0.96486	0.64263	0.72872	0.32133
307	0.88745	0.98822	0.62943	1.12690	0.82983	0.31352
113	0.43259	0.65104	0.34727	0.58646	0.37009	0.24444
301	0.72682	0.96418	0.86226	1.01918	0.54691	0.24212
202	0.51052	0.67012	0.51104	0.80558	0.51852	0.22507
1704	0.66479	0.58388	0.95499	0.58646	0.51404	0.22324
1714	0.42305	0.33577	0.76557	0.44468	0.45974	0.21326
308	1.31209	0.94298	1.04970	1.20239	1.41858	0.21118
509	0.26401	0.50613	0.22691	0.41430	0.54591	0.21008
1816	0.74749	0.65528	0.87409	0.62145	0.89110	0.20833
1310	1.11010	0.69274	0.89580	0.70799	0.70979	0.20251
1609	1.10533	0.67860	0.82871	0.61685	0.60071	0.20121
609	0.77612	0.51744	0.66889	0.45573	0.34369	0.18383
703	1.00673	0.66022	0.76163	0.61593	0.48116	0.18210
302	0.24333	0.38808	0.22494	0.42627	0.20870	0.18170
1820	0.71886	0.75707	1.03392	0.70523	0.78600	0.18113
701	0.72523	0.46230	0.66889	0.49440	0.42836	0.17751
206	0.77135	0.71395	0.55248	0.79177	0.54492	0.17625
306	0.43259	0.78181	0.56037	0.67485	0.69236	0.17566
1912	0.47871	0.40080	0.67481	0.42903	0.52848	0.17429
501	1.44886	1.14656	0.98064	1.16556	1.19892	0.17163

From the raw data, we can see that some of the top 20 precincts that experienced the greatest average change had a significant decrease in the share of voters (Example: WP 703), while others like WP 509 saw significant dips up and down across election year.

Visualizing, on a bar chart, the percent share of voters of each of these top 20 precincts:

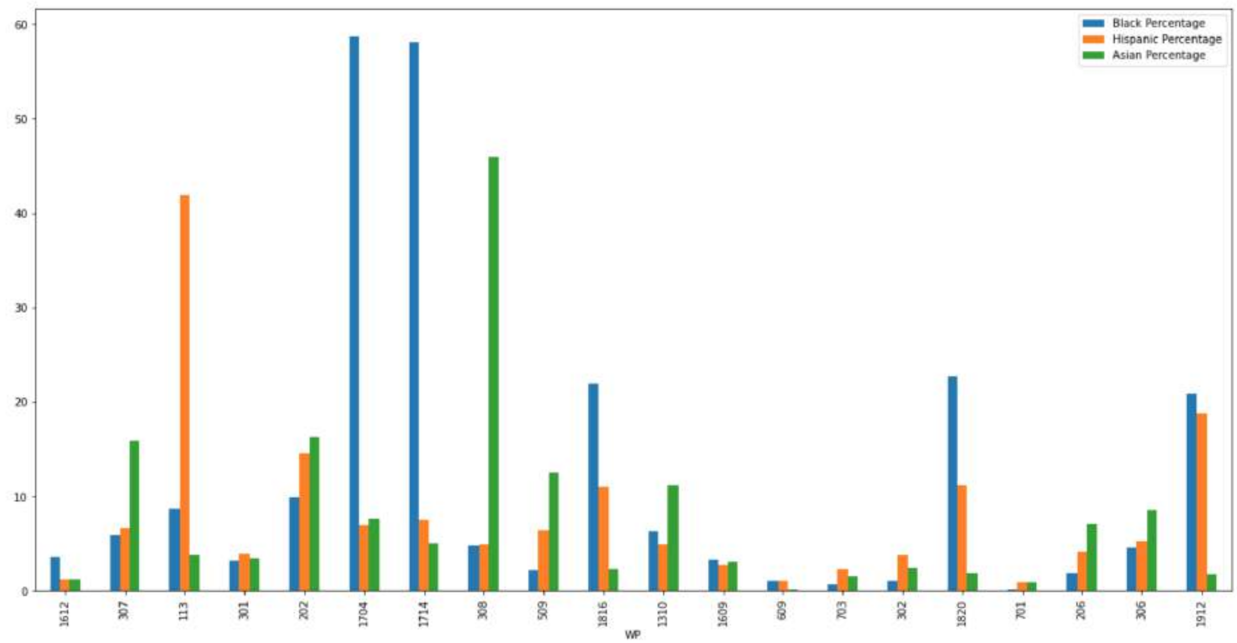


We can clearly see that the voter turnout has changed dramatically across each election in each of these precincts.

Listing the demographic breakdown of these precincts:

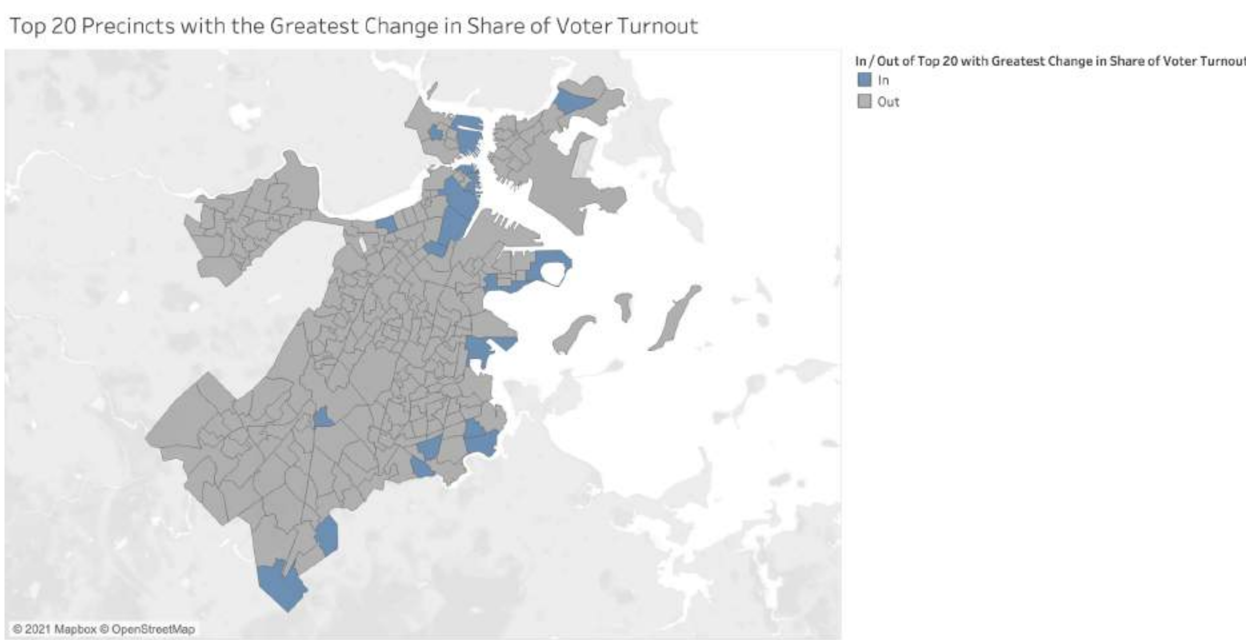
	WP	Black Percentage	Hispanic Percentage	Asian Percentage
154	1612	3.6	1.2	1.2
28	307	5.9	6.6	15.9
12	113	8.6	41.9	3.8
22	301	3.2	3.9	3.4
16	202	9.9	14.5	16.2
158	1704	58.7	6.9	7.6
168	1714	58.1	7.5	5.0
29	308	4.8	4.9	46.0
48	509	2.2	6.4	12.5
184	1816	21.9	11.0	2.3
119	1310	6.3	4.9	11.2
151	1609	3.3	2.7	3.1
59	609	1.1	1.1	0.1
62	703	0.7	2.3	1.5
23	302	1.1	3.8	2.4
188	1820	22.7	11.1	1.8
60	701	0.1	0.9	0.9
20	206	1.8	4.1	7.1
27	306	4.6	5.3	8.5
203	1912	20.9	18.7	1.7

Now, visualizing the demographic breakdown of each of these precincts:



Here, we see that most of the precincts with the greatest change in share of voter turnout have a very small POC population. However, four precincts stand out amongst them: 113, 1704, 1714, and 308, all of which are majority POC precincts.

Visualizing on a map the geographic distribution of each of these precincts:

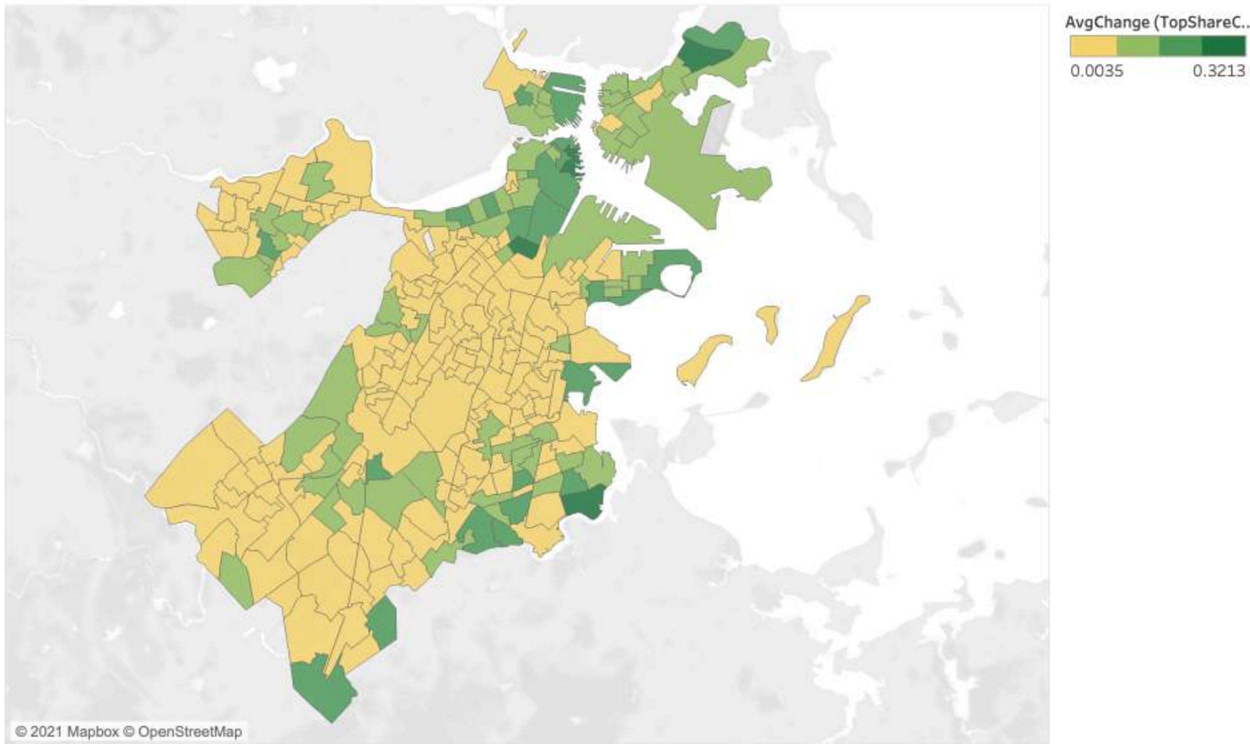


Map based on Longitude (generated) and Latitude (generated). Color shows details about In / Out of Top 20 with Greatest Change in Share of Voter Turnout. Details are shown for Ward Preci.

Here, we see that these precincts are a bit more scattered across Boston, but there is a slight concentration in an area of Dorchester and in Downtown Boston.

Viewing this information on a heat map:

Change in Share of Voters Heat Map



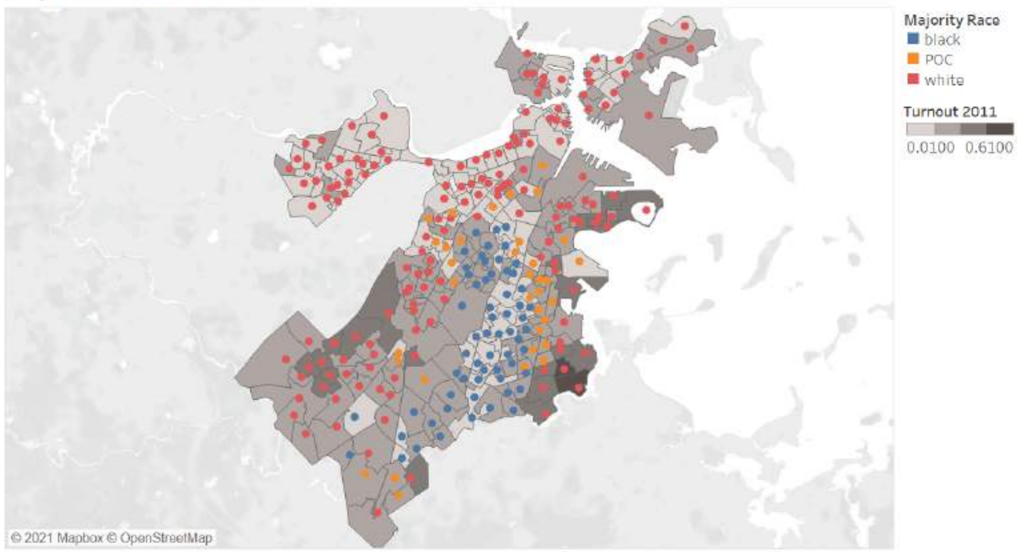
Map based on Longitude (generated) and Latitude (generated). Color shows sum of AvgChange (TopShareChange). Details are shown for Ward Preci.

Here, the greens indicated those districts which have experienced a measurable change in share of total voter turnout. As we can observe, a lot of the intense green is concentrated in the Dorchester neighborhoods, as well as near downtown Boston, Seaport, and East Boston.

Part 3: Visualizing the Change in Voter Turnout in Majority White/African American Precincts

Analyzing City Council turnout by precinct, based on the majority race of that precinct, we find that in 2011, white population had much higher turnouts:

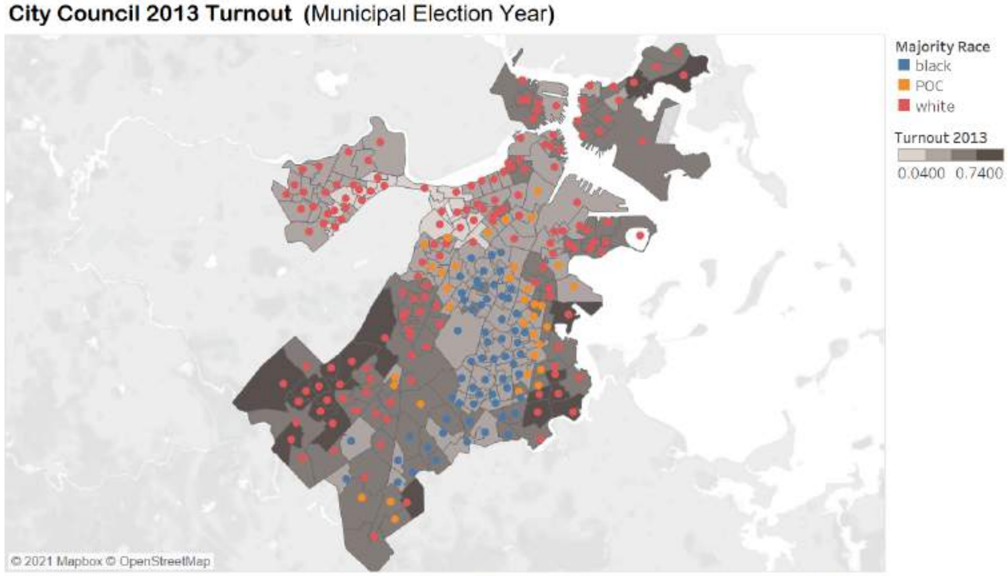
City Council 2011 Turnout



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2011. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

Similarly in 2013, but this time different precincts with majority of whites are having very high turnout rate:

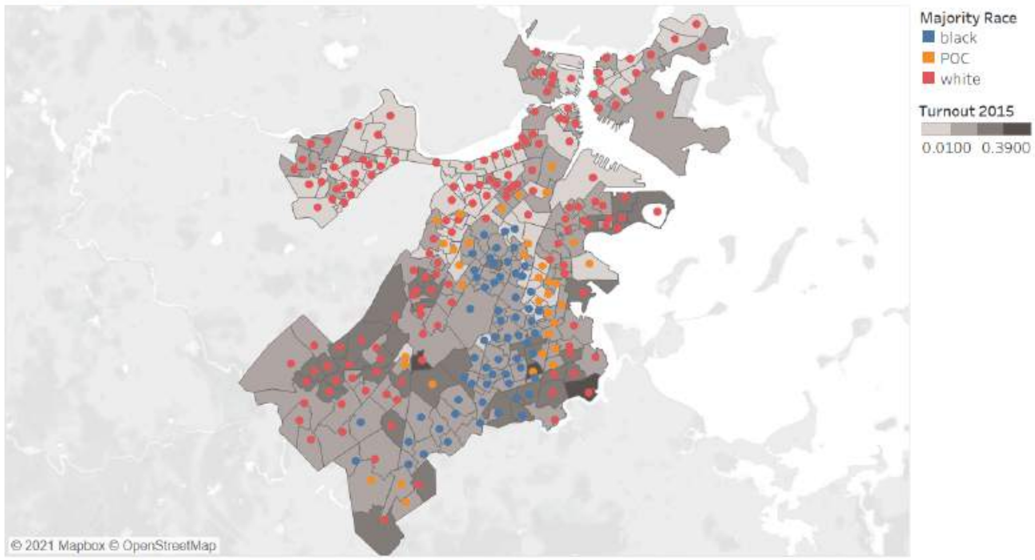
City Council 2013 Turnout (Municipal Election Year)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2013. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

In 2015 however, we saw some majority Black precincts show higher turnouts in relative terms, compared to white as you can see below:

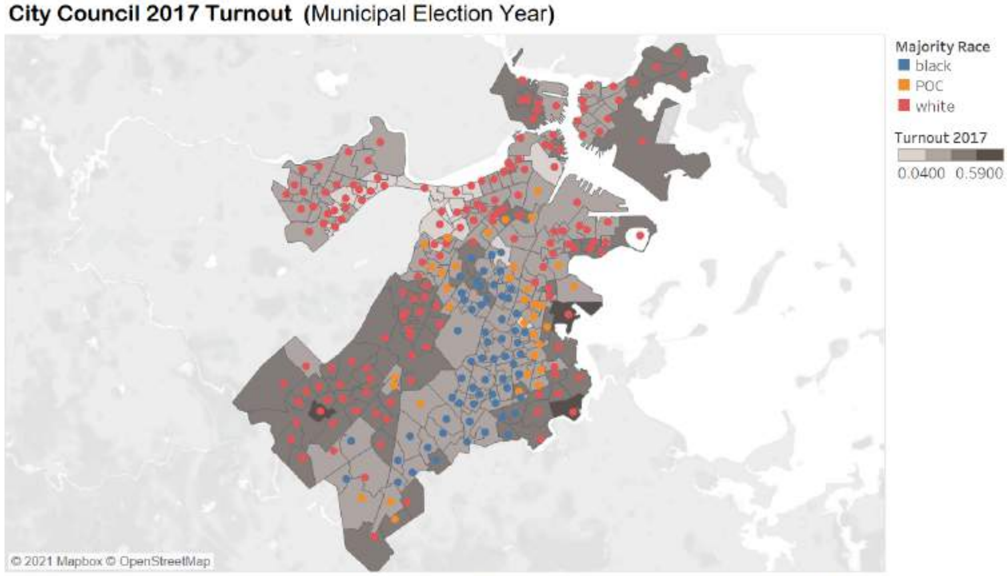
City Council 2015 Turnout



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2015. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

In 2017, we see different majority-Black precincts with higher turnouts compared to previous years:

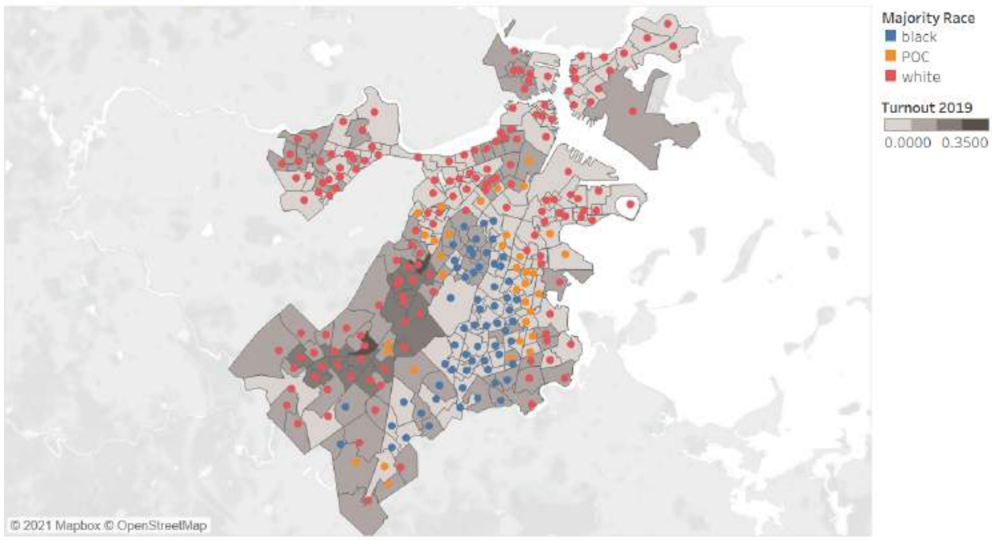
City Council 2017 Turnout (Municipal Election Year)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2017. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

Finally, in 2019 overall turnout has decreased significantly compared to previous years:

City Council 2019 Turnout



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2019. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

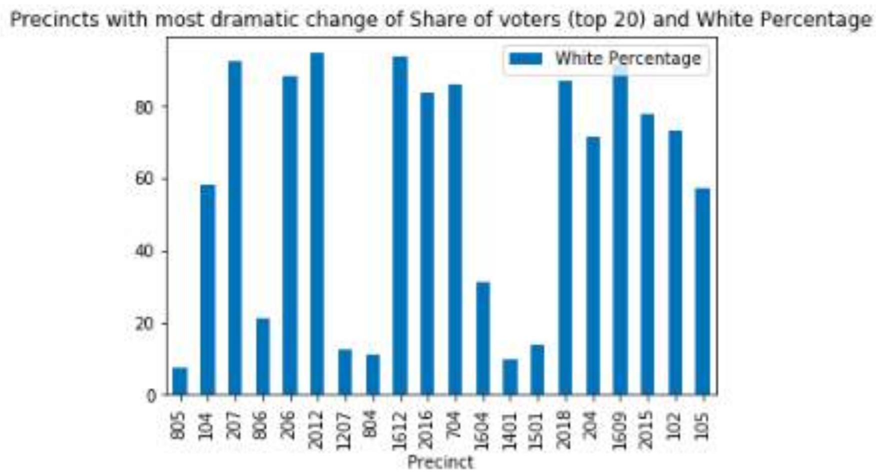
Overall, the turnouts fluctuate from year to year, and as we can see in 2011 the turnout rate ranged from 0.01-0.61, and in 2013 the range is 0.04-0.74, in 2015 the range is 0.01-0.39, in 2017 the range is 0.04-0.59, and finally in 2019 the range 0-0.35. So, clearly there is huge overall fluctuation in turnouts.

On relative terms, the turnout trend for majority Black precincts tends to increase across the years, while for majority white precincts is volatile, pretty much reflecting overall fluctuations.

Part 4: How has Michael F. Flaherty's performance changed across districts with large white populations?

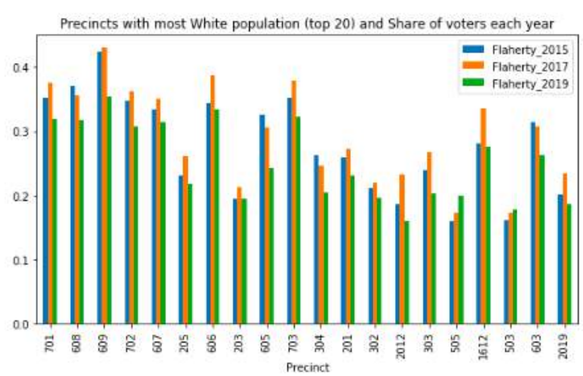
In this section, we are going to find some relations between Michael F. Flaherty's performance and districts with large white populations.

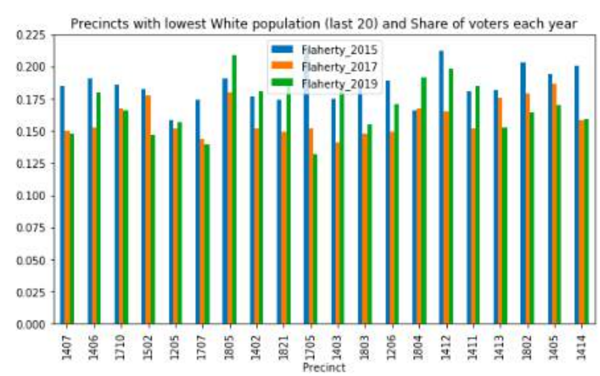
To begin, we made a graph of precincts with the most dramatic change of share of voters, and printed out the average turnout change for each year. (Flaherty is in the election from 2015 to 2019.)

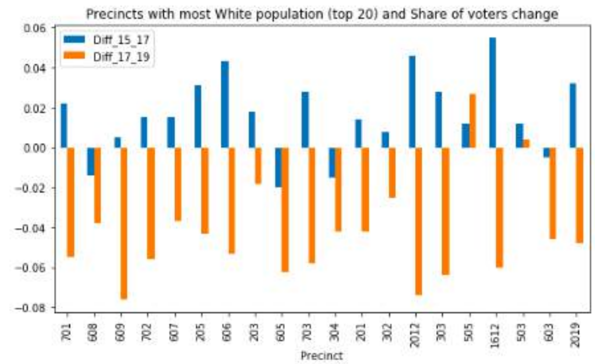


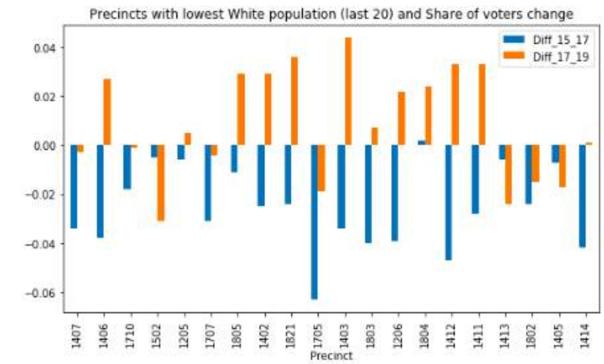

```
Average Share of voters Change from 2015 to 2017: -0.009761904761904765
Average Share of voters Change from 2017 to 2019: -0.02182142857142858
Average Share of voters Change from 2015 to 2019: -0.03157142857142857
```

From the graph above, we can see that 13 over 20 precincts with the greatest average change in *share of voter turnout* over time have a large white population (above 50%). The average share of voters decreases each year since they are all negative.

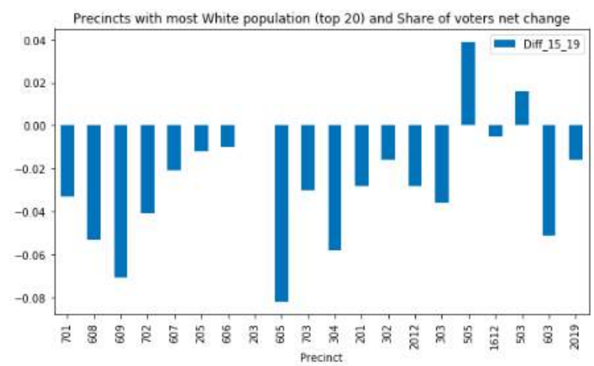


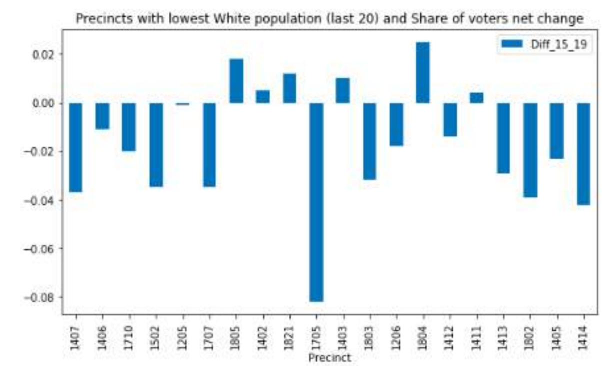






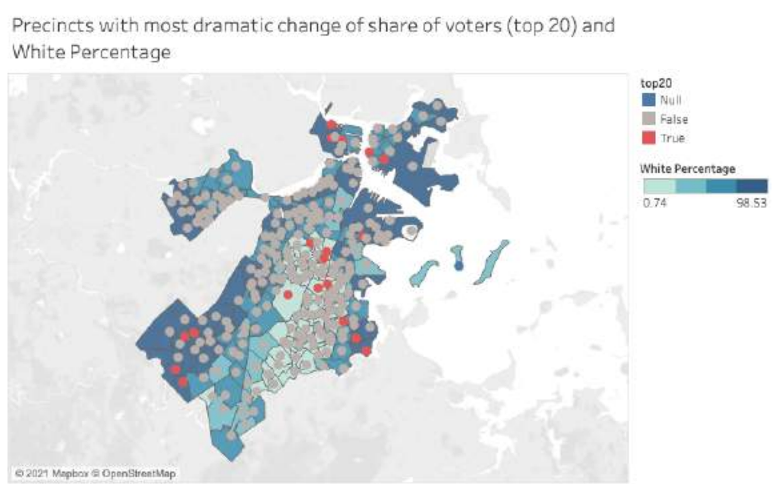
From the graphs above, we can see that the share of voter changes of 2015 to 2017 are mostly positive in precincts with most white population while are negative in precincts with lowest white population. We can know that, in precincts with most white populations, Flaherty was popular in 2017 but lost supporters in 2019.





From the graph of the share of voters net change, we can see that most changes are negative, which corresponds to what we found before: Flaherty lost supporters in 2019 compared to 2015.

Visualizing it on an actual map:

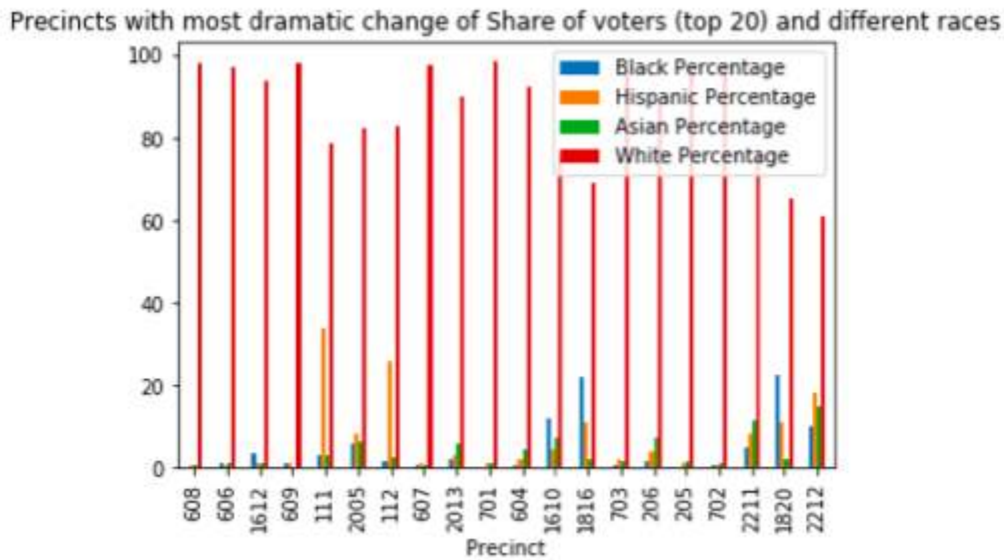


Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about top20. Details are shown for Ward Preci.

Part 5: How has Michelle Wu's performance changed over time?

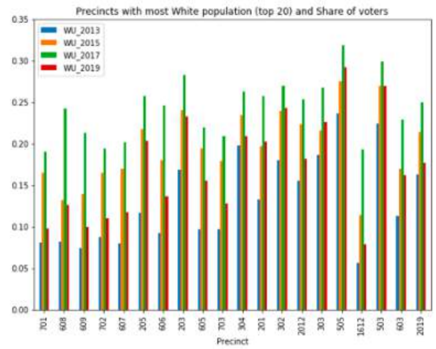
In this section, we will try to find the correlation between Michelle Wu's most dramatic change of share of voters and different racial demographics.

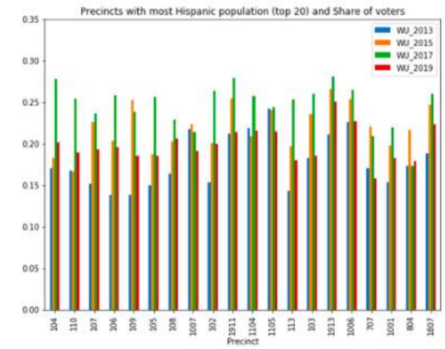
To begin our analysis, we made a graph of precincts with the most dramatic change of share of voters across different races. We also printed out the average turnout change for each year. (Wu is in the election from 2013 to 2019.)

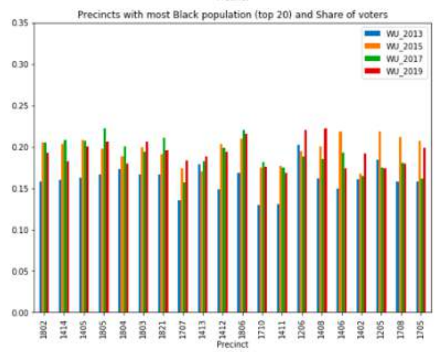


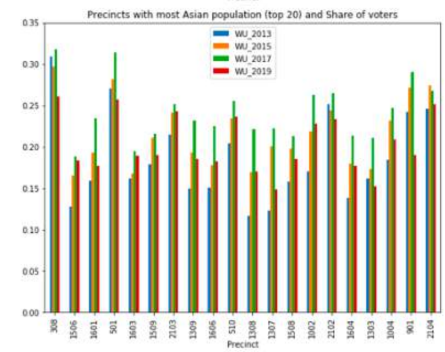

```
Average Turnout Change from 2013 to 2015: 0.03730830039525692
Average Turnout Change from 2015 to 2017: 0.017567460317460298
Average Turnout Change from 2017 to 2019: -0.035019841269841284
Average Turnout Change from 2013 to 2019: 0.02062698412698412
```

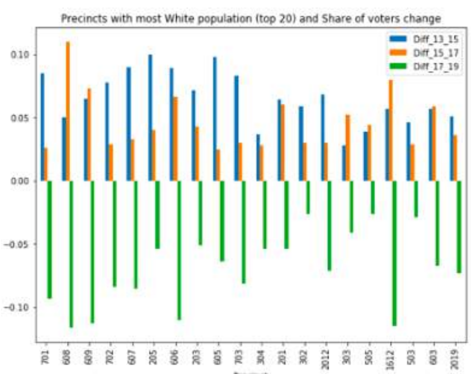
I found that, in general, she has more supporters in 2019 than in 2013. Only in 2019, she lost supporters compared to the last election (2017). In other years, she had a larger share of voters than last elections. All precincts with the most dramatic change of share of voters are with the most white population.

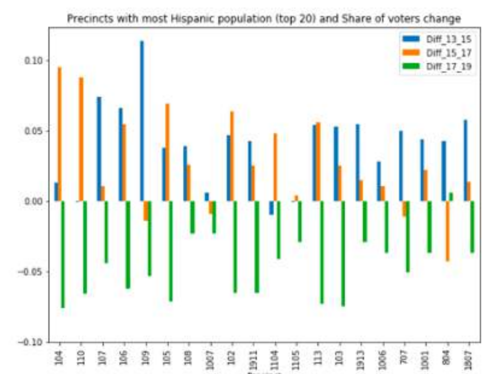


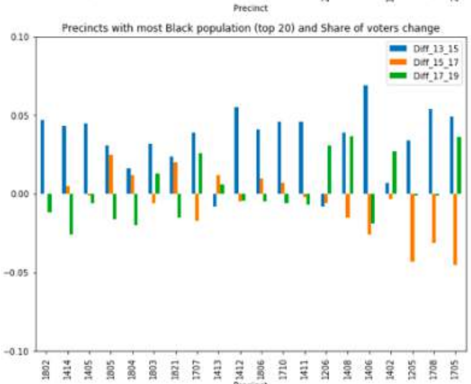


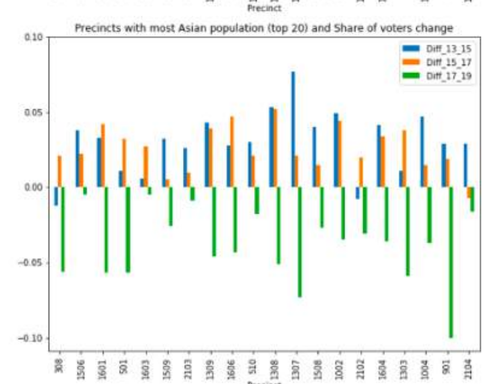




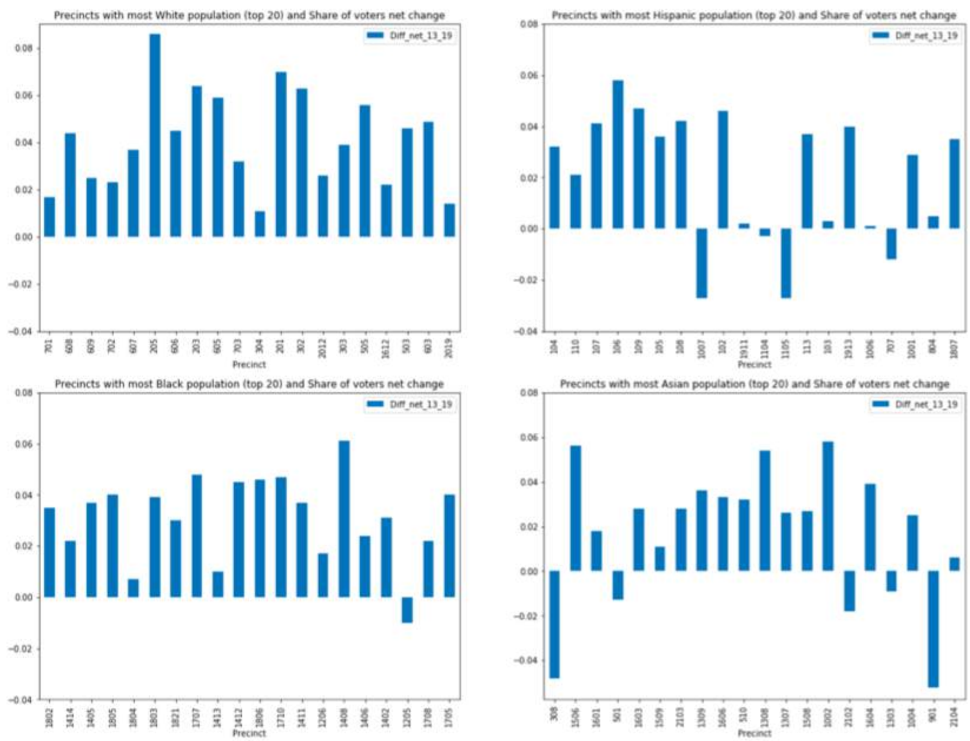








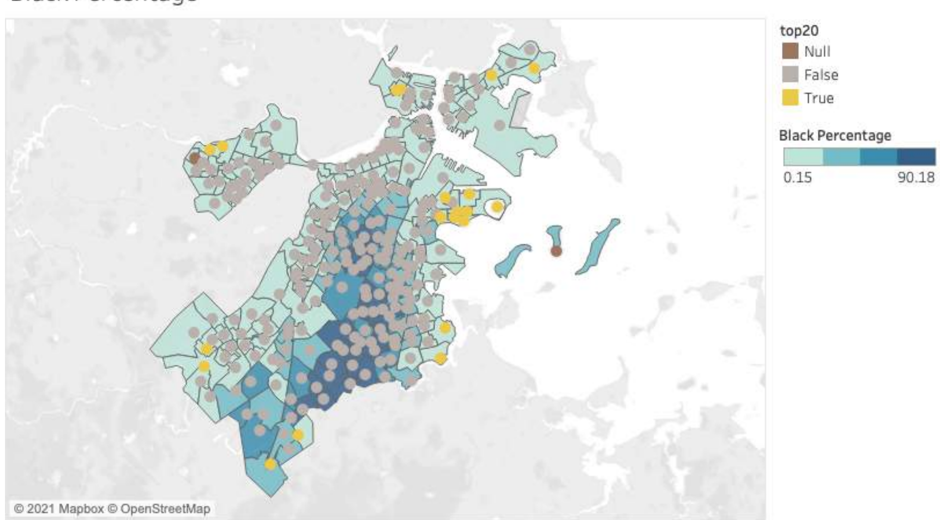
From the graphs, we can see that Wu has the most share of voters in 2017. In 2019 we can see a huge decrease of the share of voters in precincts with most white, Hispanic and Asian populations. The decrease is not obvious in precincts with most Black populations. We can even see some increase.



In general, Wu gained share of voters in 2019 compared to 2013, which corresponds to what we found before.

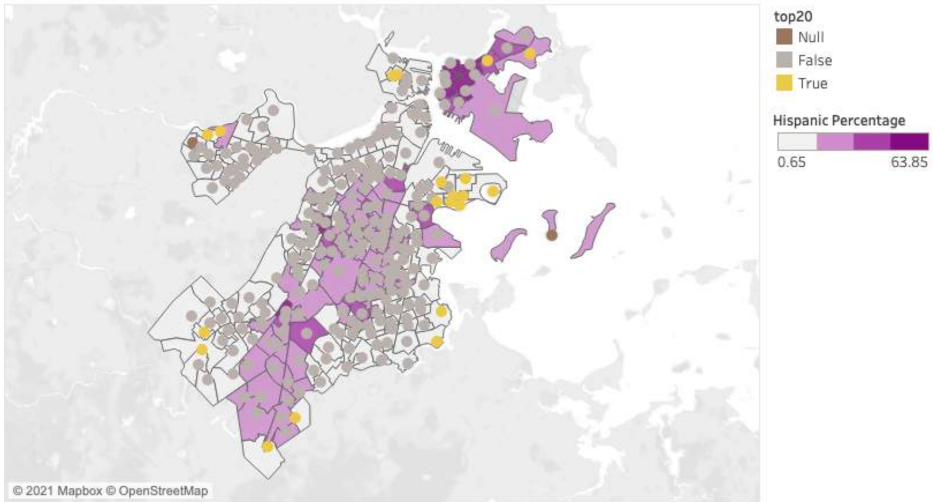
Visualizing it on an actual map:

Precincts with most dramatic change of share of voters (top 20) and Black Percentage



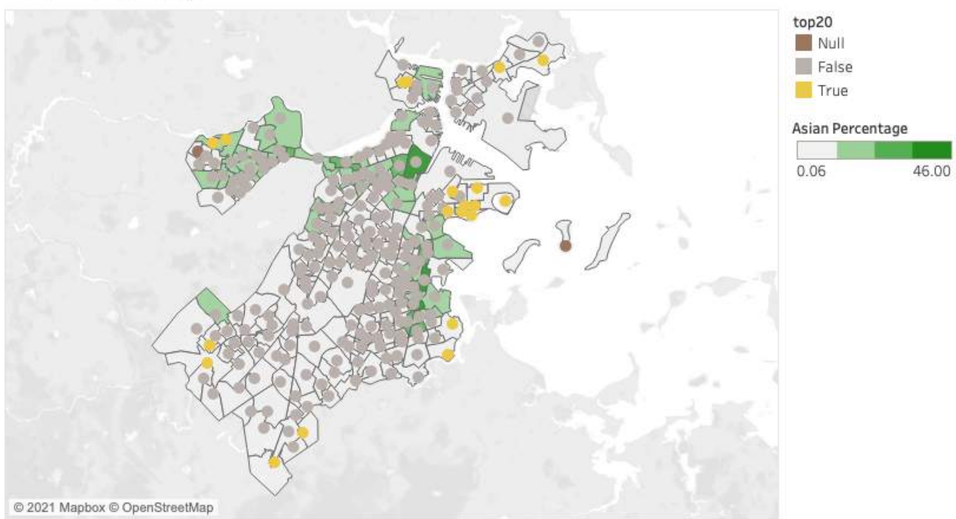
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Black Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about top20. Details are shown for Ward Preci.

Precincts with most dramatic change of share of voters (top 20) and Hispanic Percentage



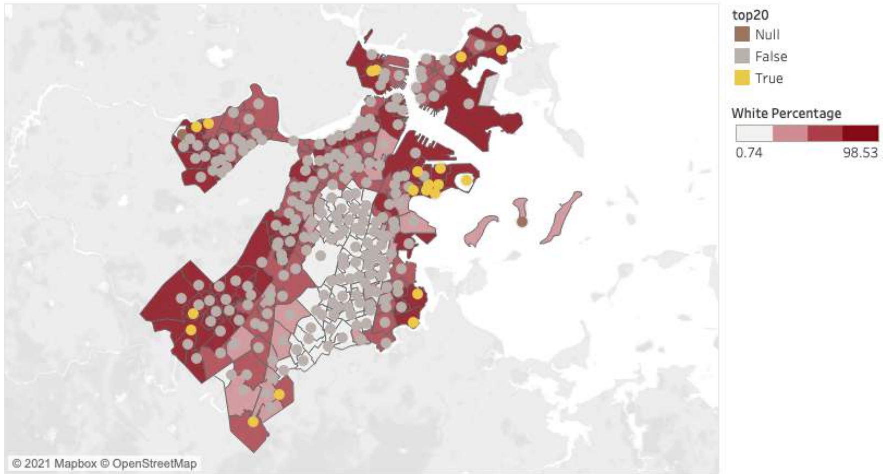
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Hispanic Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about top20. Details are shown for Ward Preci.

Precincts with most dramatic change of share of voters (top 20) and Asian Percentage



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Asian Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about top20. Details are shown for Ward Preci.

Precincts with most dramatic change of share of voters (top 20) and White Percentage



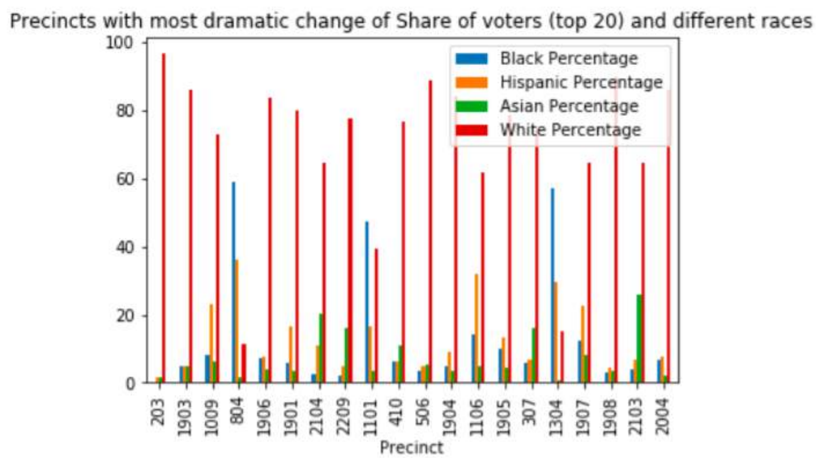
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of White Percentage. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about top20. Details are shown for Ward Preci.

Part 6: Annissa E. George and how her performance has changed over time.

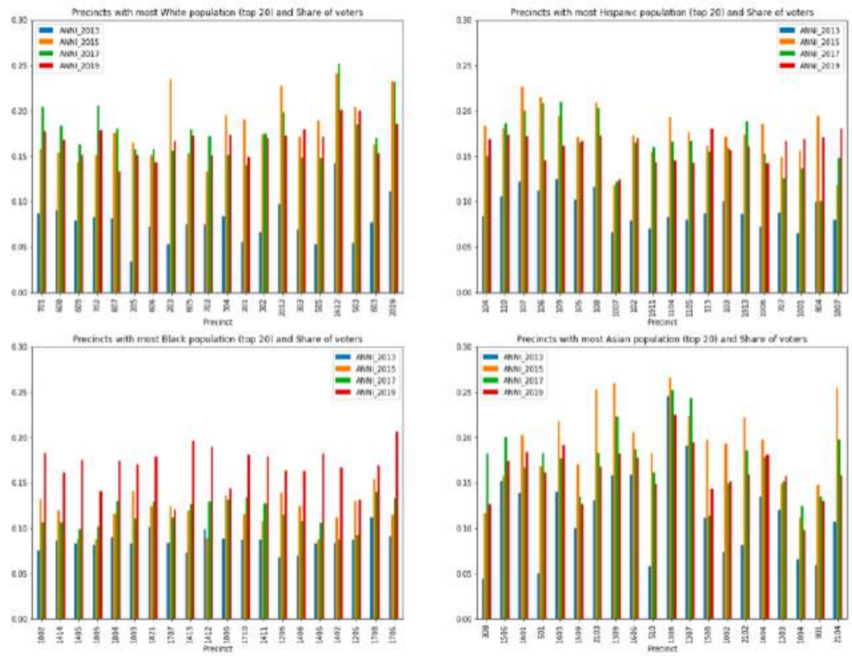
In this section, we will try to find the correlation between Annissa E. George's most dramatic change of share of voters and different racial demographics.

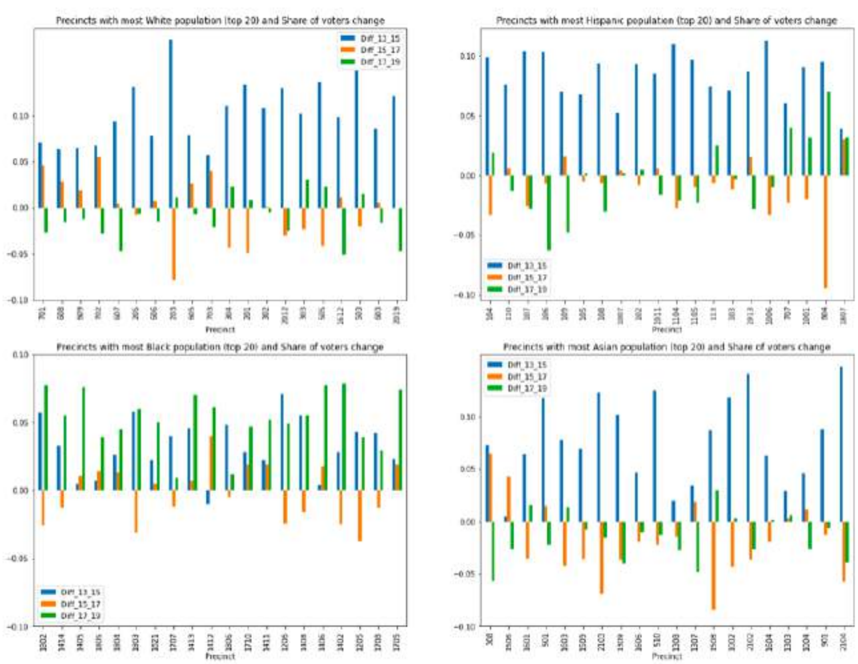
To begin our analysis, we made a graph of precincts with the most dramatic change of share of voters across different races. We also printed out the average turnout change for each year. (Annissa is in the election from 2013 to 2019.)

```
Average Turnout Change from 2013 to 2015: 0.089201581027668
Average Turnout Change from 2015 to 2017: -0.012269841269841262
Average Turnout Change from 2017 to 2019: 0.0026071428571428517
Average Turnout Change from 2013 to 2019: 0.07628571428571426
```



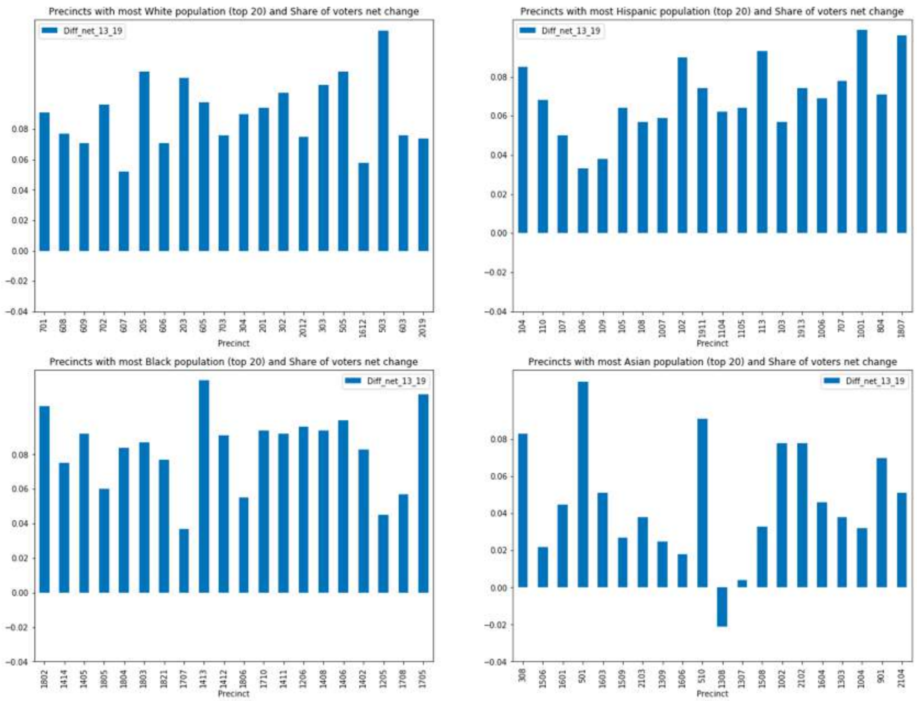
I found that, in general, she has more supporters in 2019 than in 2013. Only in 2017, she lost supporters compared to the last election (2015). In other years, she had a larger share of voters than last elections. 17 out of 20 precincts with the most dramatic change of share of voters are with the most white population. 3 precincts are with most Black Percentage.





From the graphs, we can see that Annissa has the most share of voters in 2019 in precincts with the most Black population. There is a huge increase in Annissa’s share of voters in 2015 than that in 2013. Her share of voters in precincts with most Black population didn’t decrease as much as that in other precincts in 2017 and 2019. In

2019, her share of voters increased in precincts with the most Black population but decreased in general in other precincts dominated by other race populations.



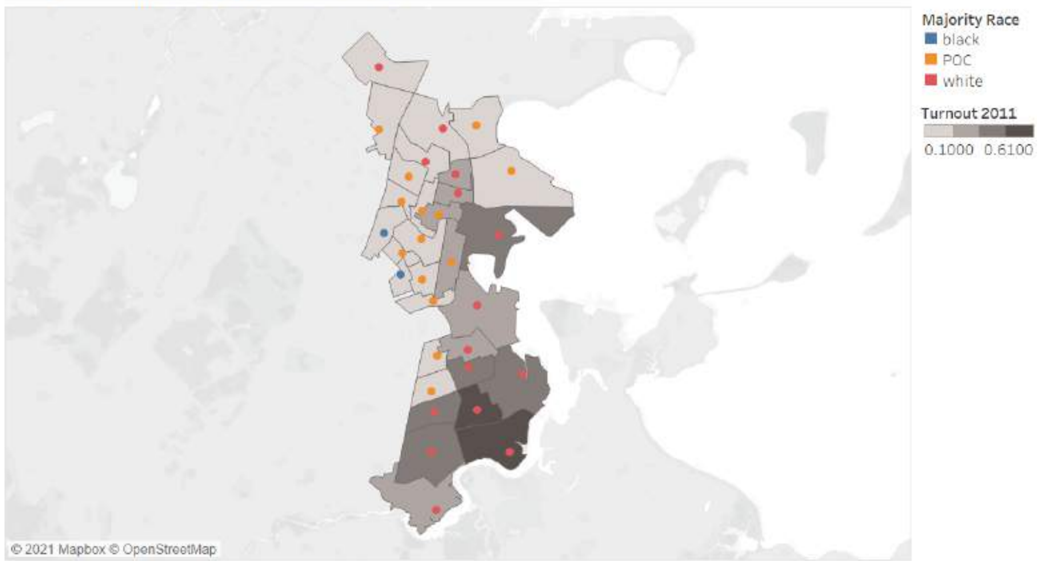
In general, Annissa gained share of voters in 2019 compared to 2013, which corresponds to what we found before.

Part 7: Changes in District 3 Specifically

We break down our findings to focus specifically on District 3.

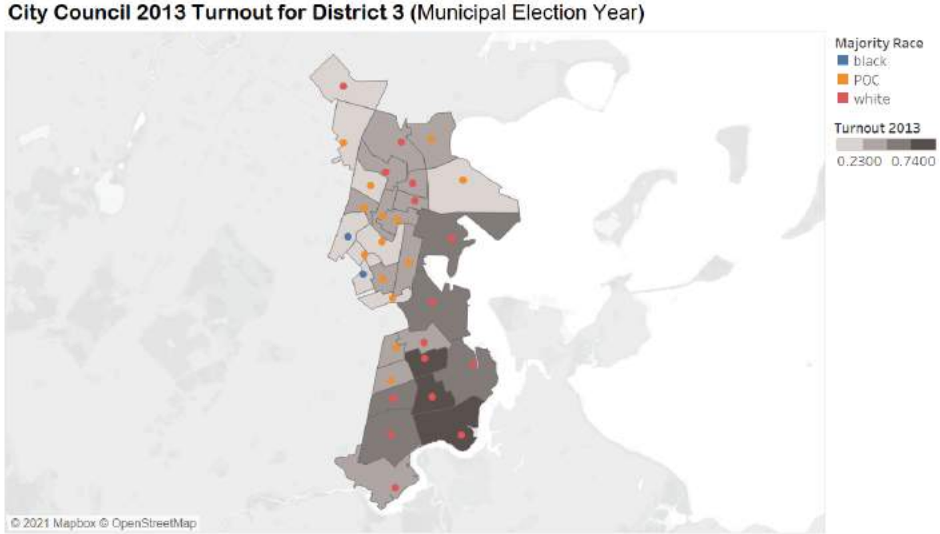
First, we look at all those areas, and compare turnout across the years in the following maps:

City Council 2011 Turnout for District 3



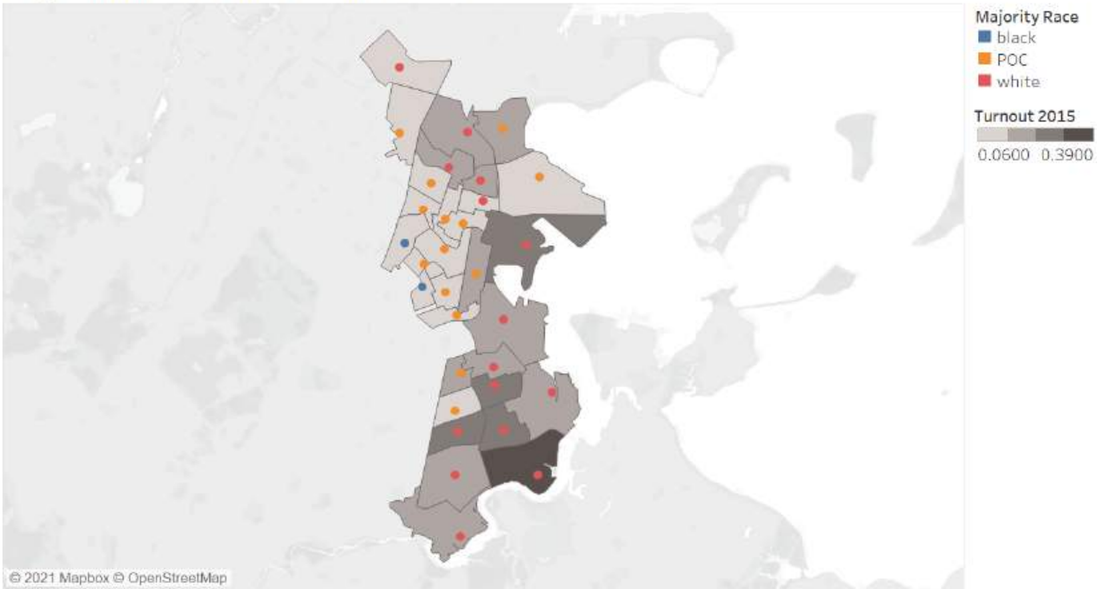
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2011. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

City Council 2013 Turnout for District 3 (Municipal Election Year)



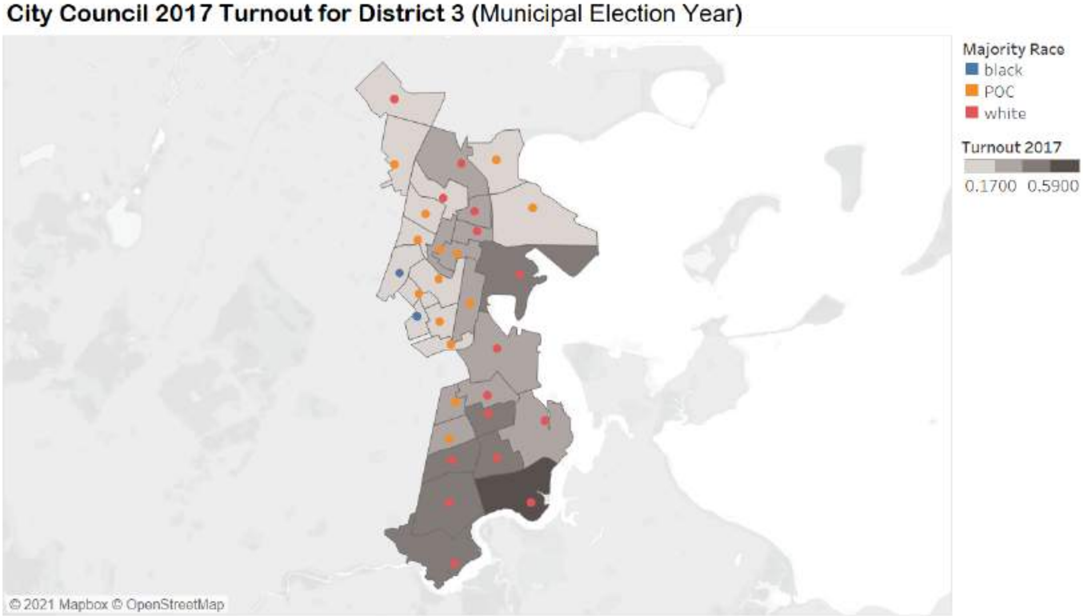
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2013. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

City Council 2015 Turnout for District 3



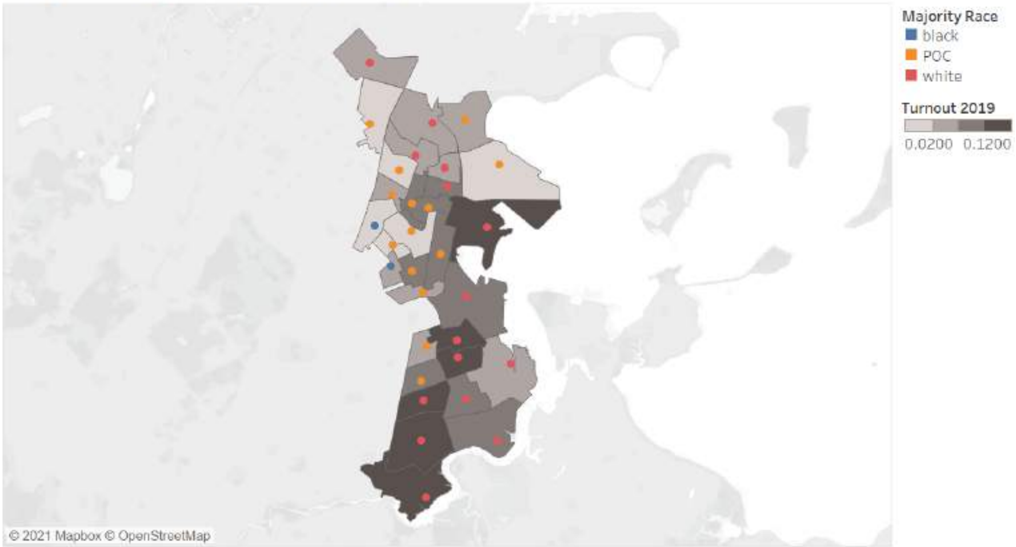
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2015. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

City Council 2017 Turnout for District 3 (Municipal Election Year)



Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2017. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

City Council 2019 Turnout for District 3



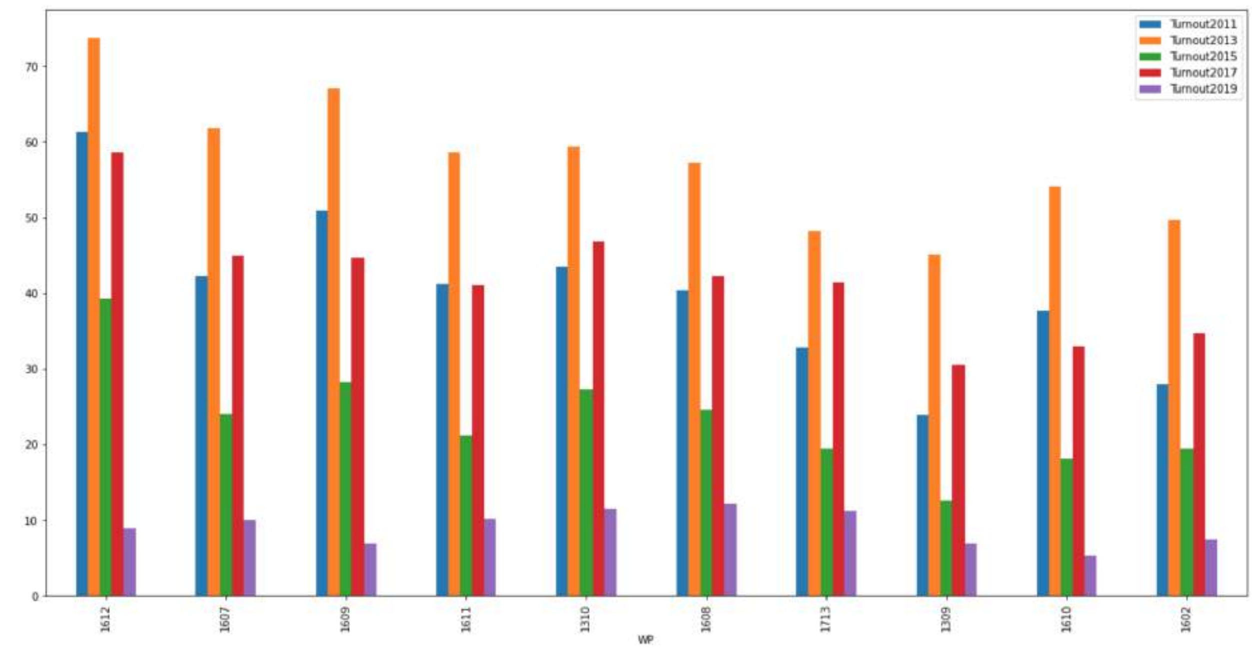
Map based on Longitude (generated) and Latitude (generated). For marks layer Geometry: Color shows sum of Turnout 2019. Details are shown for Ward Preci. For marks layer Geometry (2): Color shows details about Majority Race. Details are shown for Ward Preci.

We find that for different years, different precincts have different turnouts, but the constant factor is the southern areas usually have much higher turnouts compared to northern ones on the above maps. On the racial side of things, precincts with a majority Black population are recently improving their turnouts, as you can see on the left side of the maps.

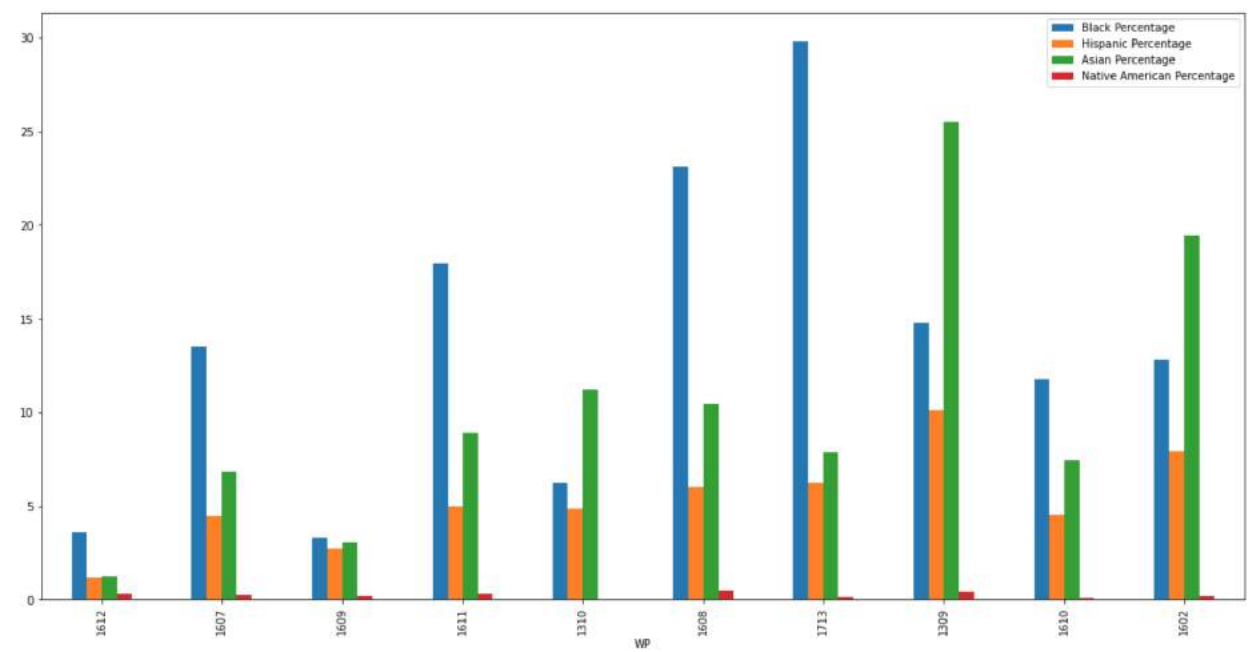
Second, we find the top 10 precincts with the greatest volatility across city council election year in District 3:

WP	Turnout2011	Turnout2013	Turnout2015	Turnout2017	Turnout2019	AvgChange
1612	61.3	73.8	39.3	58.6	8.9	29.0
1607	42.2	61.8	24.0	44.9	10.0	28.3
1609	51.0	67.0	28.2	44.7	6.9	27.3
1611	41.2	58.7	21.1	41.1	10.2	26.5
1310	43.5	59.4	27.3	46.8	11.5	25.7
1608	40.4	57.2	24.6	42.3	12.1	24.3
1713	32.8	48.2	19.5	41.4	11.1	24.1
1309	23.8	45.1	12.6	30.5	6.9	23.8
1610	37.7	54.1	18.2	33.0	5.3	23.7
1602	28.0	49.7	19.4	34.7	7.5	23.6

Visualizing the volatility:



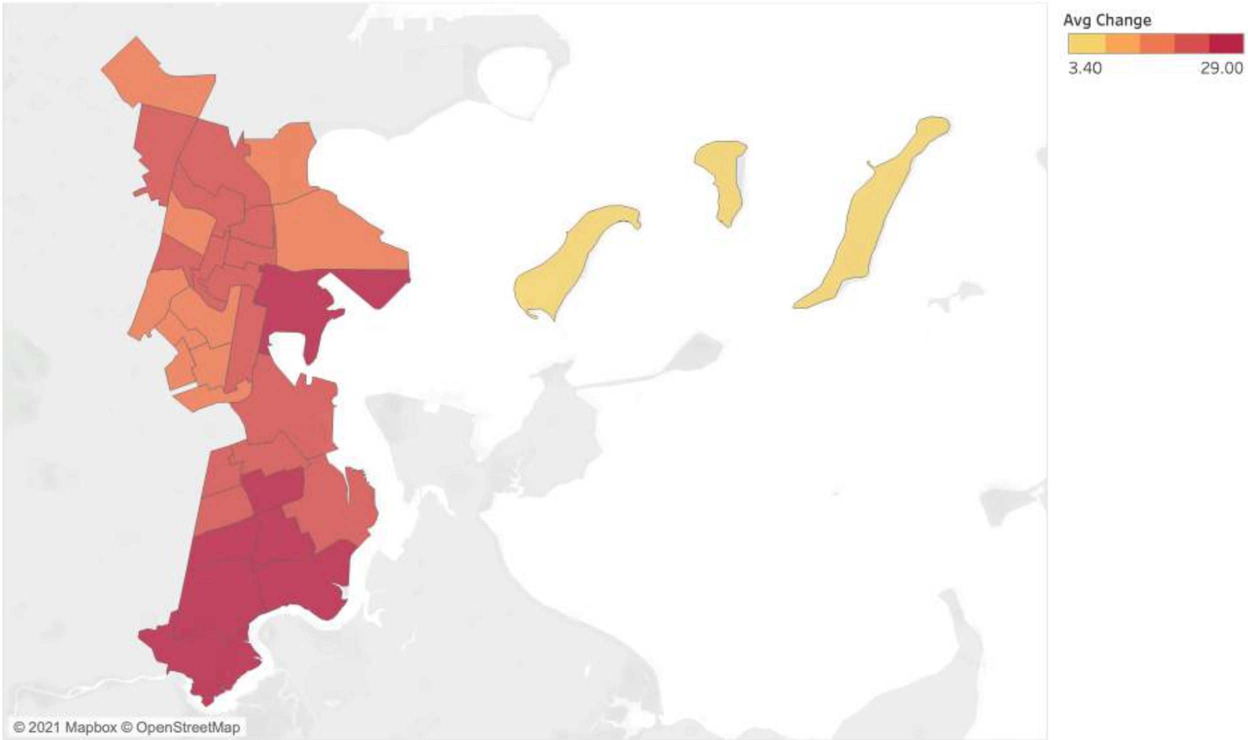
And finally, looking at the demographic breakdown of each of these top precincts:



As we can see, about 3 of these precincts have a sizable Black population (greater than 15%), while two have a sizable Asian population (greater than 15%). However, most have a majority white population, and the top most volatile district, 1612, has a very low POC population.

Visualizing this on a heat map of District 3:

Heat Map of Volatility of Precincts in District 3



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Avg Change. Details are shown for Ward Preci.

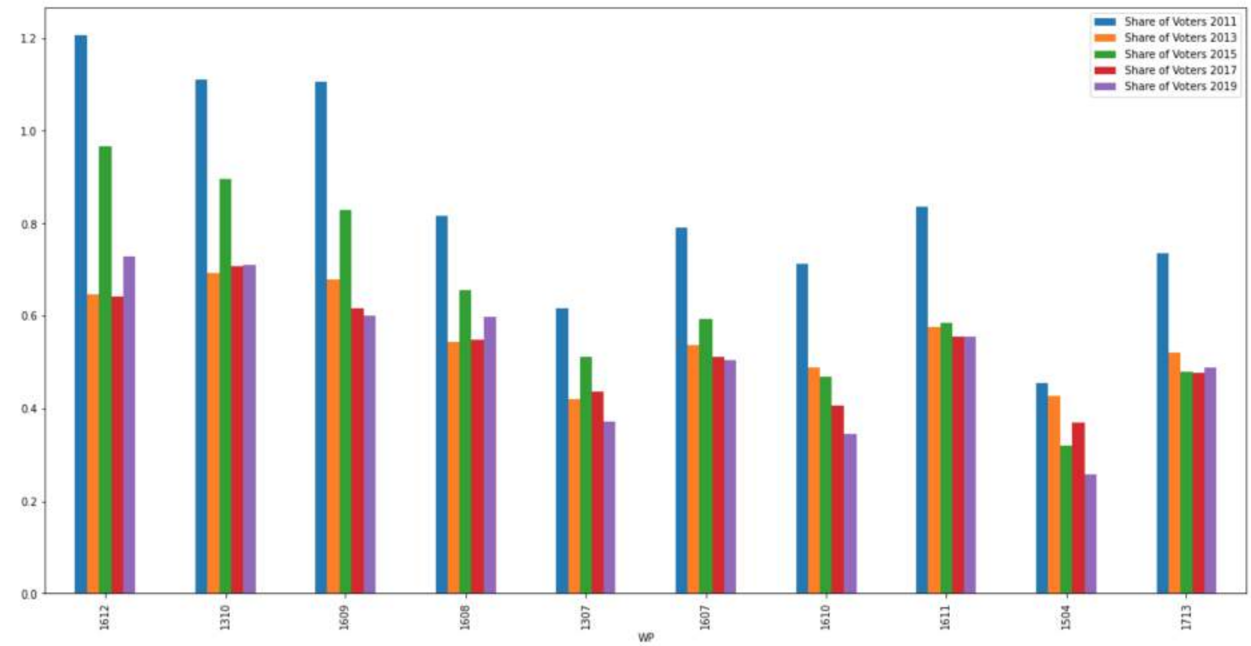
As we can see, those precincts with the greatest volatility across CC election years seem to be mostly concentrated in the southern part of District 3.

Now, for the same subset of District 3 wards, we determine the top 10 precincts with the greatest change in *share of voter turnout* from 2011 to 2019:

WP	Share of Voters 2011	Share of Voters 2013	Share of Voters 2015	Share of Voters 2017	Share of Voters 2019	AvgChange
1612	1.20712	0.64750	0.96486	0.64263	0.72872	0.32133
1310	1.11010	0.69274	0.89580	0.70799	0.70979	0.20251
1609	1.10533	0.67860	0.82871	0.61685	0.60071	0.20121
1608	0.81747	0.54430	0.65705	0.54872	0.59821	0.13594
1307	0.61549	0.42130	0.51104	0.43732	0.37108	0.10597
1607	0.79202	0.53581	0.59391	0.51189	0.50407	0.10104
1610	0.71091	0.48916	0.46960	0.40693	0.34618	0.09118
1611	0.83655	0.57611	0.58602	0.55516	0.55637	0.07561
1504	0.45645	0.42695	0.31965	0.36827	0.25652	0.07429
1713	0.73477	0.52168	0.47947	0.47691	0.48764	0.06715

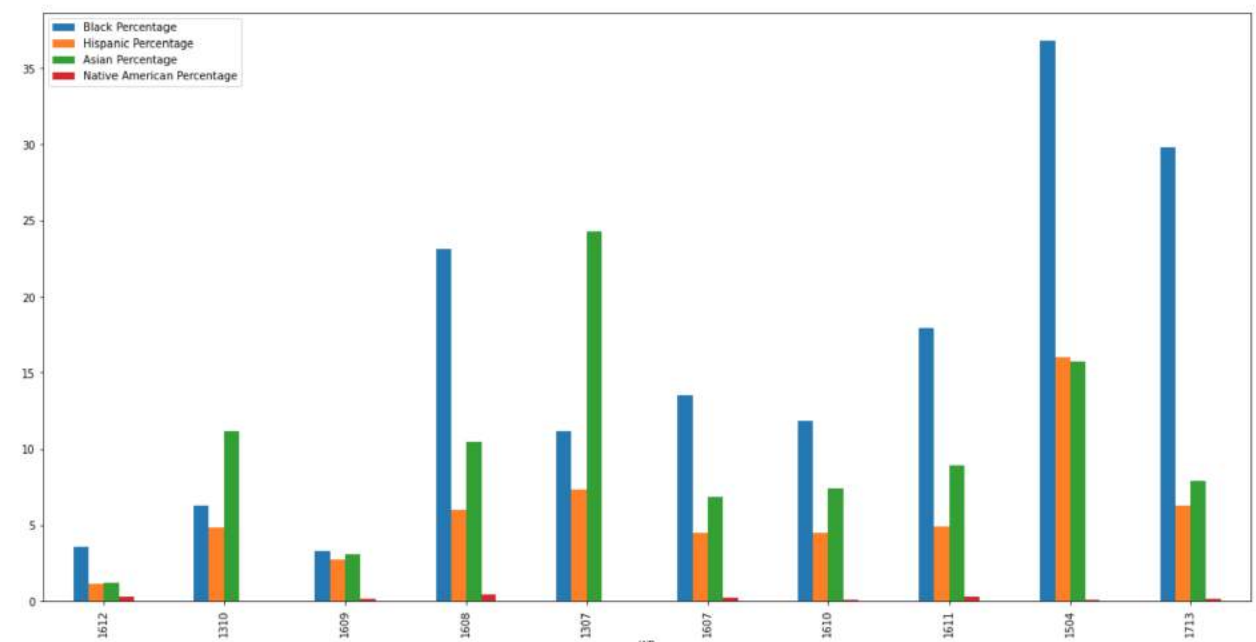
Again, here share of voters is [Ballots Cast in X Precinct]/[Total Ballots Cast in Boston] *100.

Visualizing how this share has changed over each year on a bar chart:



As we can see, the share of voters in Ward-Precinct 16-12 really fluctuates across the City Council election years, while Ward-Precinct 16-10's share of total votes cast has really dropped off since 2011. 16-11, on the other hand, stays steady across most city council elections except 2011.

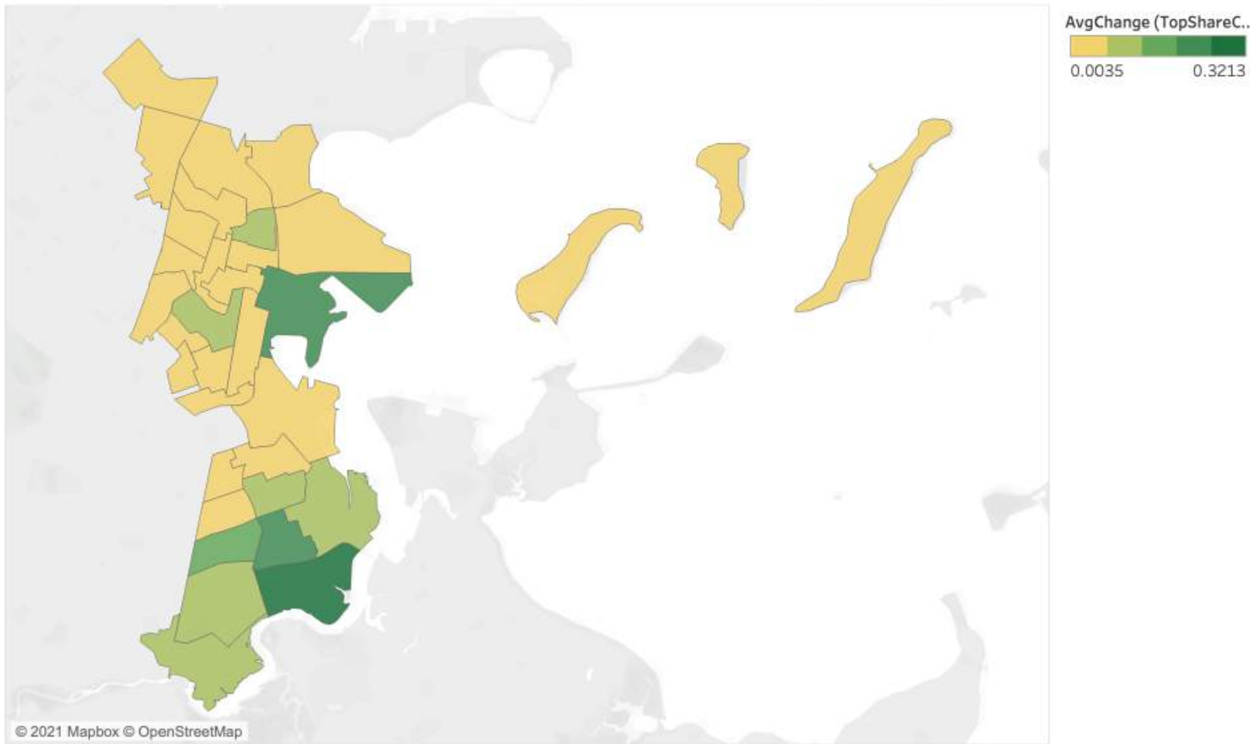
Now, for the demographic breakdown of each of these precincts:



The demographics breakdown here indicates no clear pattern: some precincts have a very small POC population while some, like 1608 and 1307 are multiracial.

Visualizing this change in share of voter turnout on a heat map:

Change in Share of Voter Turnout accross City Council Elections (District 3)

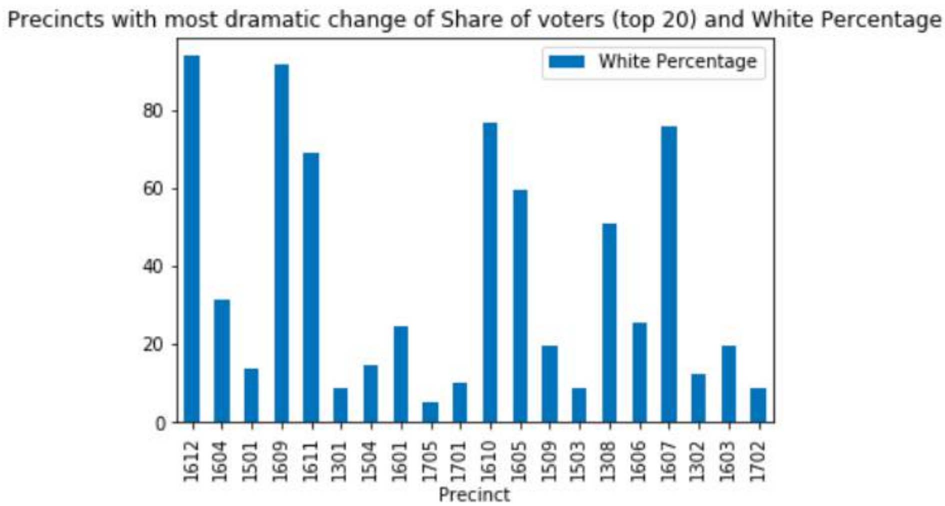


Map based on Longitude (generated) and Latitude (generated). Color shows sum of AvgChange (TopShareChangeD3). Details are shown for Ward Preci.

Similar to our previous findings, we find that those precincts with the greatest change in share of voter turnout tend to be concentrated in the southern part of District 3.

Now, we focus on how Michael F. Flaherty's performance changed in precincts in District 3 with large white populations.

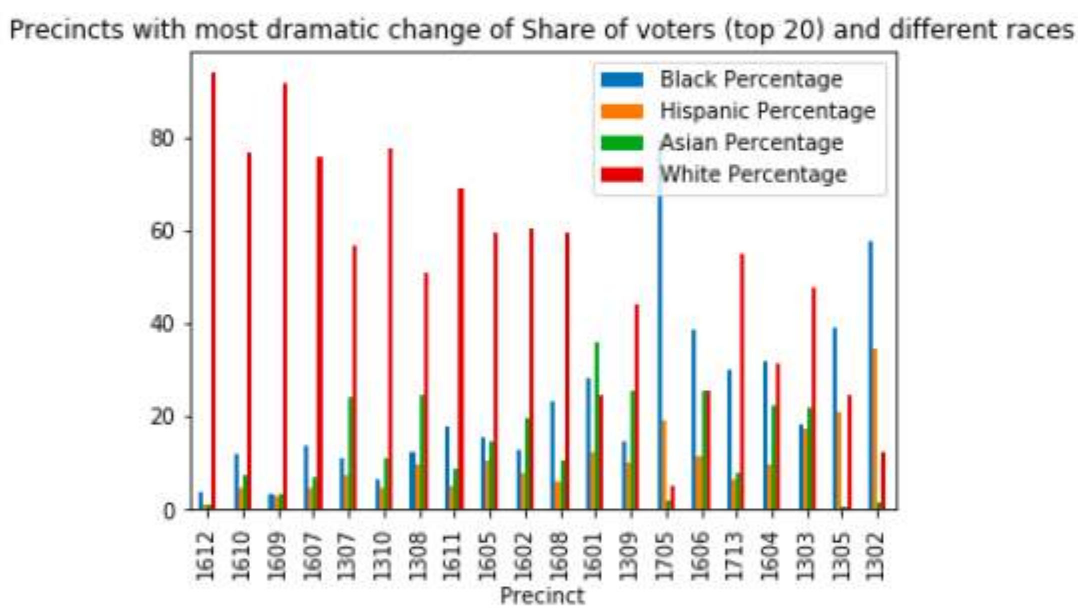
```
Average Turnout Change from 2015 to 2017: -0.020622222222222226
Average Turnout Change from 2017 to 2019: -0.017911111111111117
Average Turnout Change from 2015 to 2019: -0.03855555555555554
```



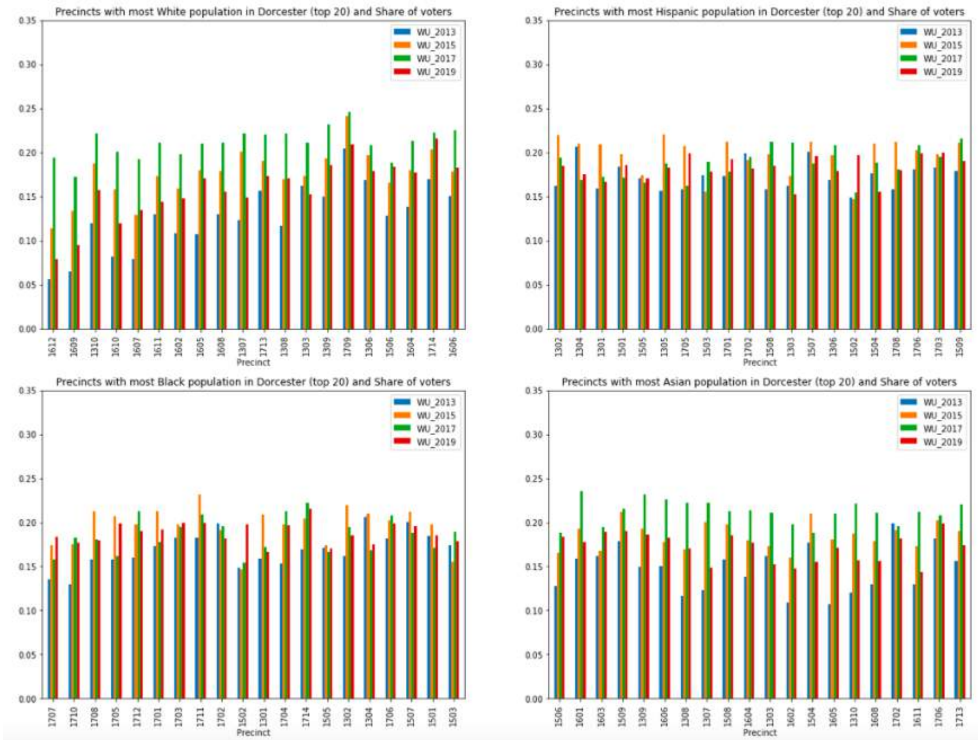
From the graph above, we can see that districts with the greatest average change in *share of voter turnout* over time do not have strong correlation with white percentage.

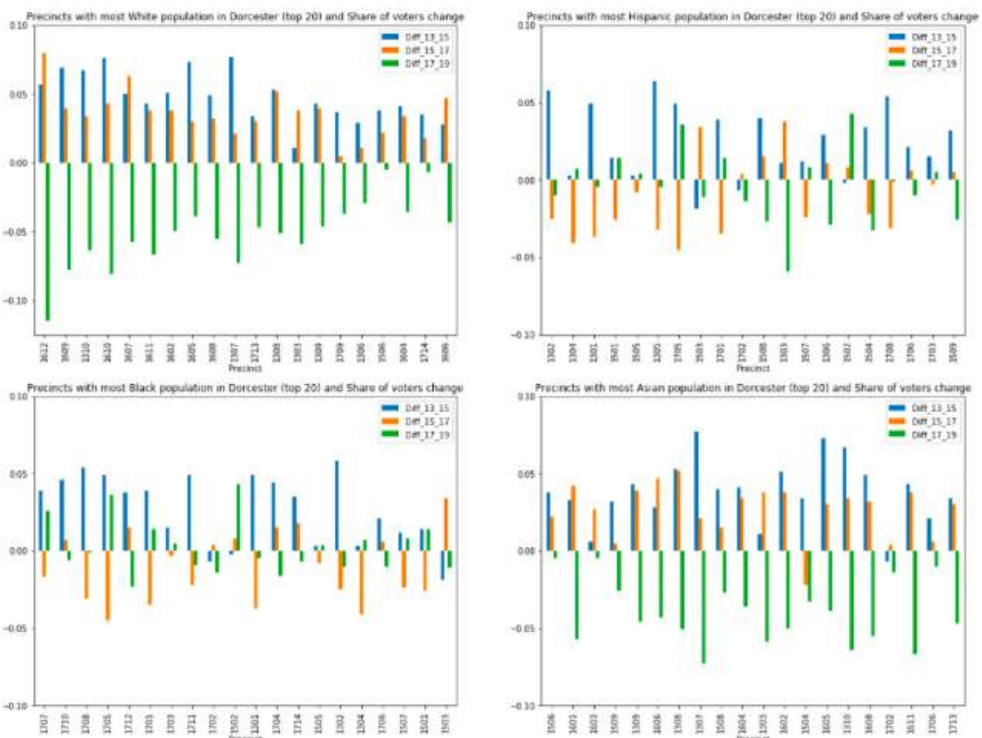
Lastly, we focus on how Michelle Wu's performance changed in district 3 over time.

```
Average Turnout Change from 2013 to 2015: 0.03722222222222222
Average Turnout Change from 2015 to 2017: 0.011644444444444442
Average Turnout Change from 2017 to 2019: -0.025355555555555544
Average Turnout Change from 2013 to 2019: 0.023577777777777788
```

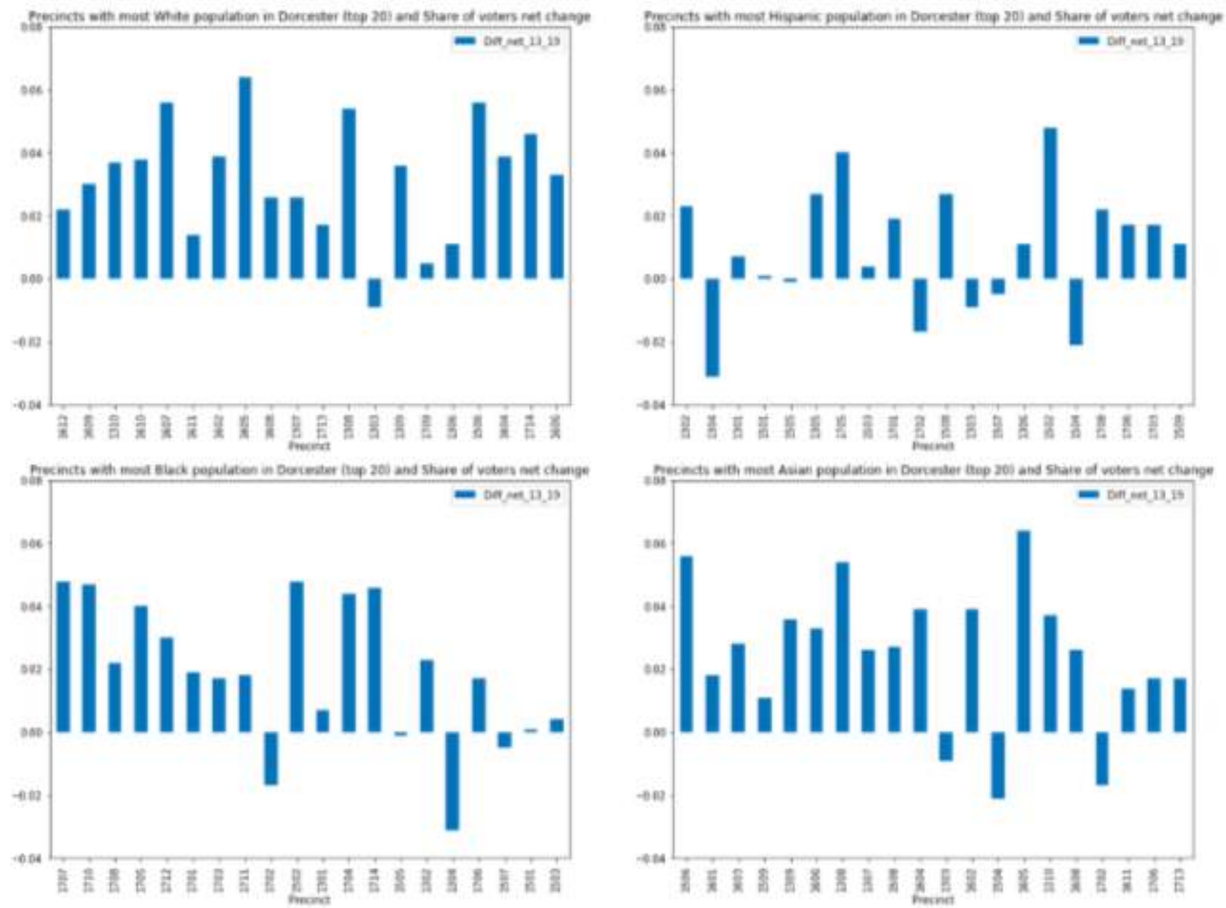


The turnout change trend in District 3 is the same as the overall trend. 16 out of 20 precincts with the most dramatic change of voter turnout are with the most white population. The rest 3 of the precincts with the most dramatic change of voter turnout are with the most Black population. One is with the most Asian population.





We can see that the decrease of share of voter turnout in 2019 is only obvious in precincts with most white and Asian populations. In precincts with most Hispanic and Black population, we can see a decrease of share of voter turnout in 2017, but the decrease does not appear in precincts with most white and Asian population.

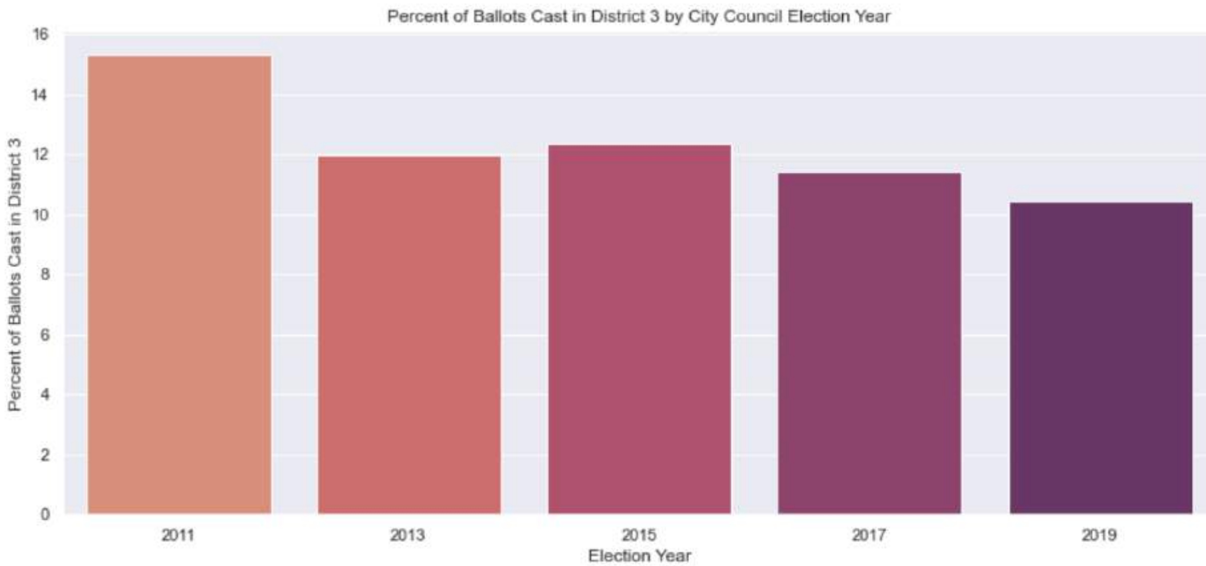


In general, Wu gained more supporters in 2019 compared to 2013, which is the same as the overall population.

Part 8: District 3 as a Share of City Vote

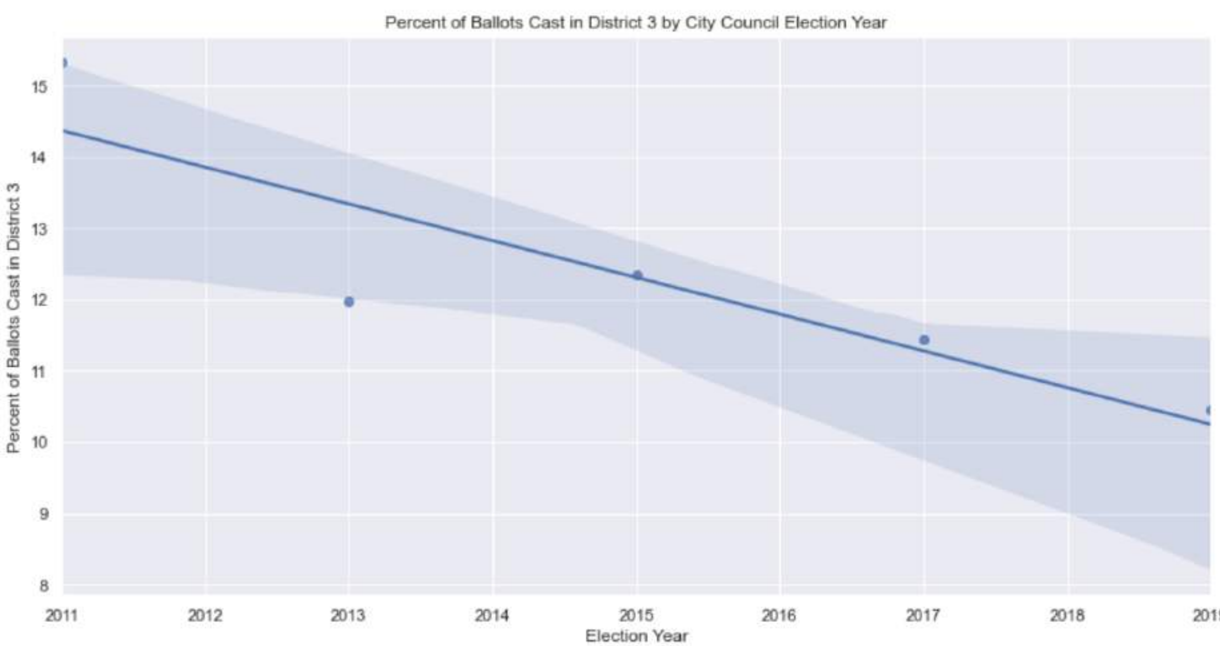
In this section we measure District 3 as a share of total city vote. That is, across city council election years 2011-2019, we measure the total number of ballots cast in District 3 / the total number of ballots cast in Boston as a percentage.

We first visualize our findings on a bar chart:



As we can see, the net change from 2011 to 2019 is almost a 5% difference.

Now, creating a linear regression plot:



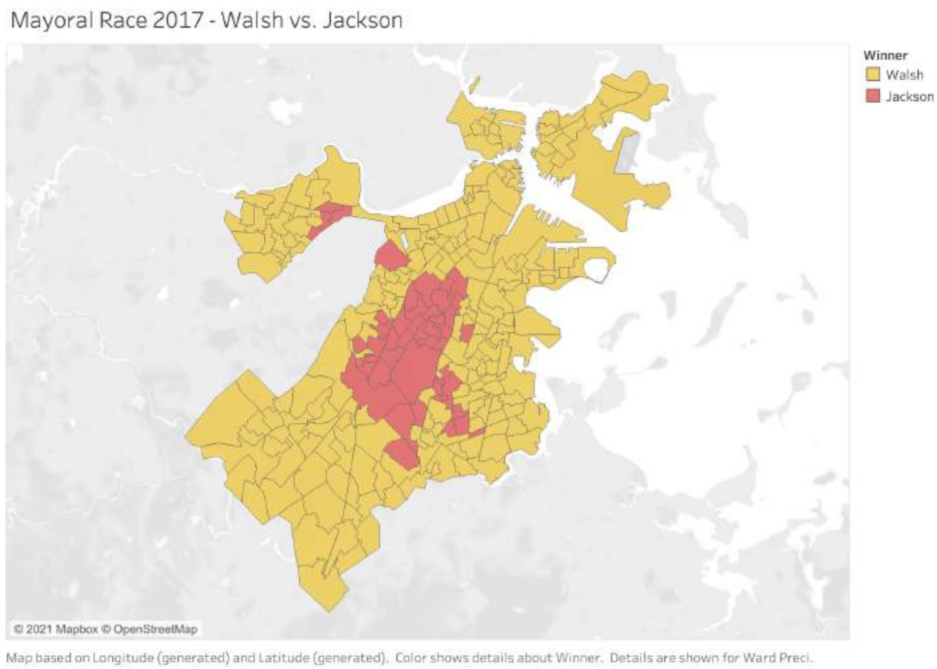
Here, we find that the share of votes in District 3 has been on a downward trend since 2011.

Key Question: What are the key predictors in determining support for Black candidates for Boston?

From our previous analysis, we have found that race plays a key role in determining support for Black candidates. We will now be examining how other factors, such as median income, educational level, and features of neighborhoods may affect support for Black candidates. We will be comparing the outcomes to these maps of three elections with African American frontrunners: 2017 Mayoral Race, 2018 DA Race, and the 2018 US House Democratic Primary.

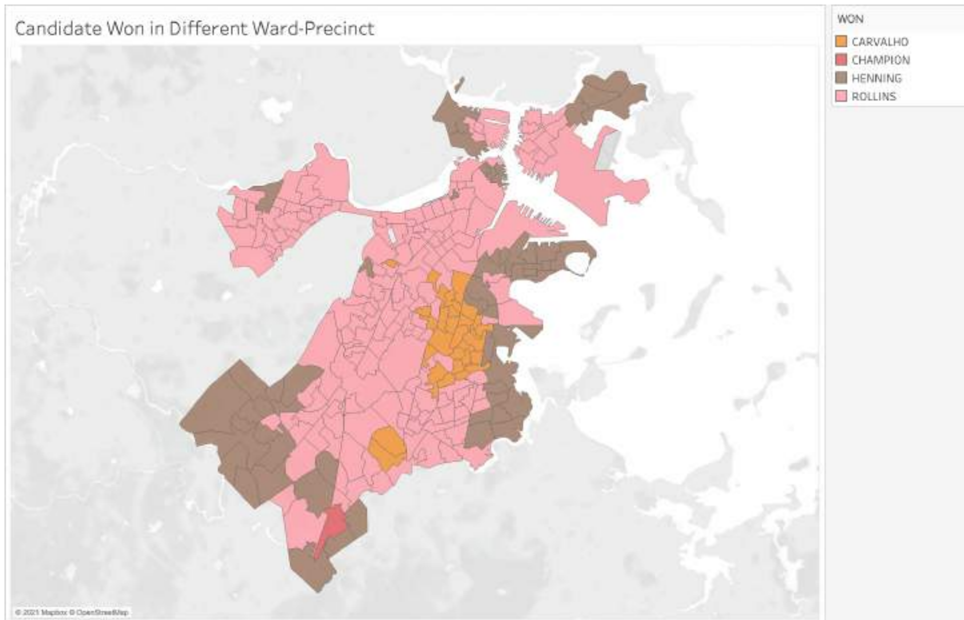
Part 1: Visualization of Key Races

<u>Outcome of the 2017 General Mayoral Race:</u>



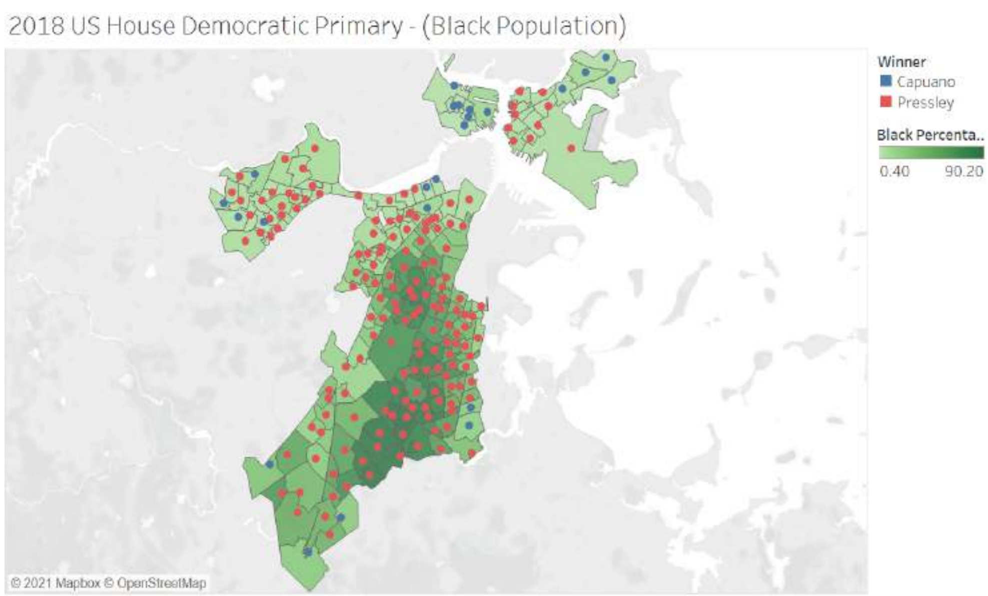
In this race, we will be focusing on Tito Jackson, the Black candidate for mayor in 2017.

Outcome of the 2018 DA Race:



Here, we will be focusing on the performance of the first Black woman DA in Boston, DA Rollins.

Outcome of the 2018 US House Democratic Primary:

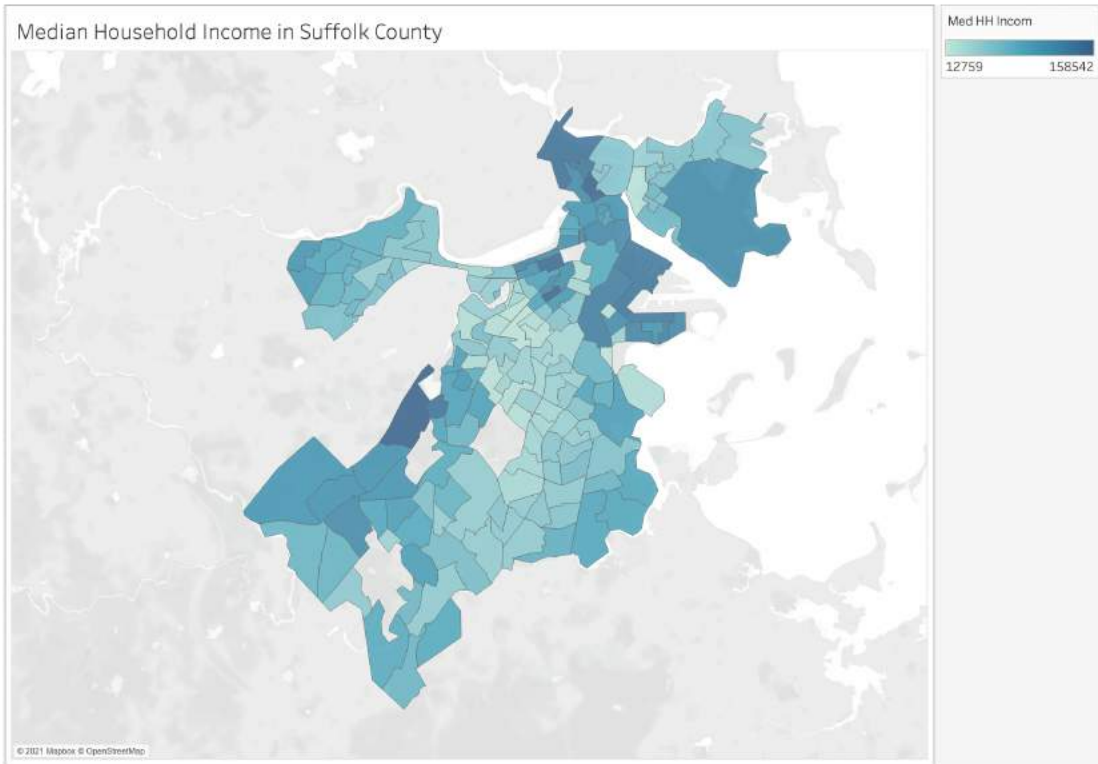


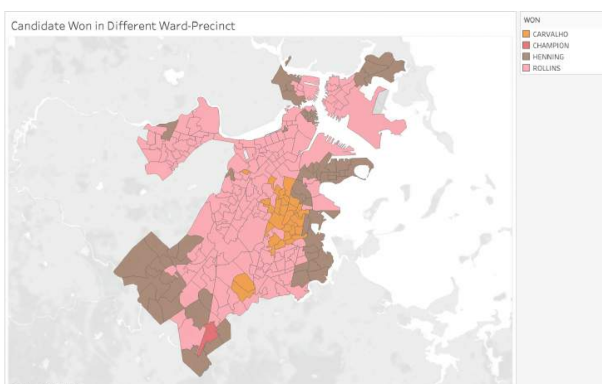
Map based on Longitude (generated) and Latitude (generated). For marks layer Pcnts.shp: Color shows sum of Black Percentage. Details are shown for Ward Prec. For marks layer Pcnts.shp (2): Color shows details about Winner. Details are shown for Ward Prec. The view is filtered on Winner, which keeps Capuano and Pressley.

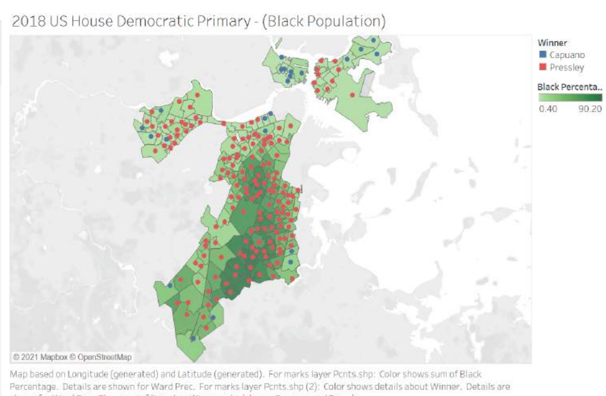
Here, we will primarily focus on Ayanna Pressley's performance.

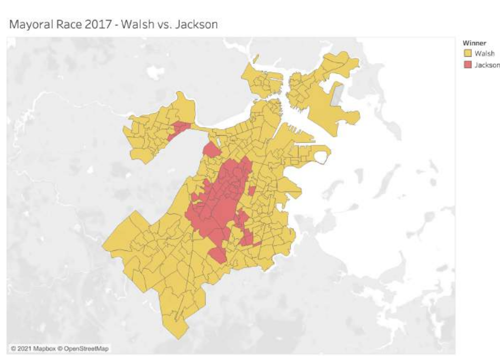
Part 2: Median Household Income

We first visualize the median household income across various Boston precincts:









Source:
https://data.census.gov/cedsci/table?t=Financial%20Characteristics&g=0500000US25025.140000&tid=ACSST5Y2019.S2503&moe=false&tp=false&hi

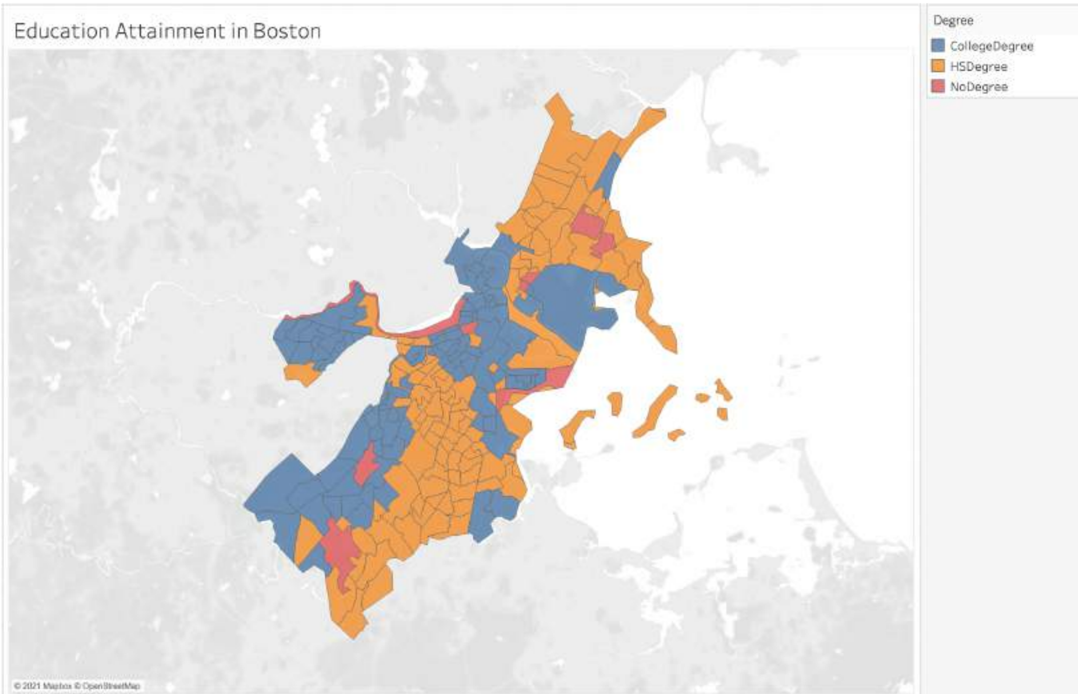
dePreview=true

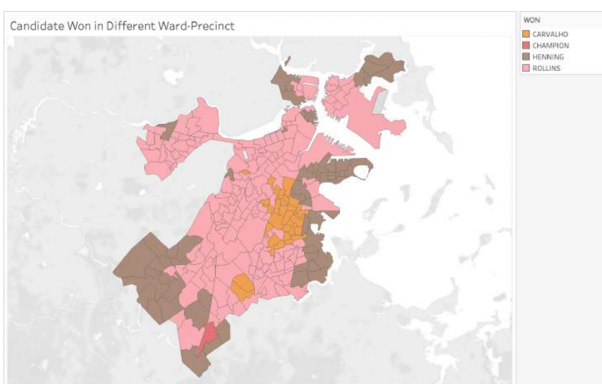
We compare this median income map with the maps of various elections in which a Black candidate was a frontrunner.

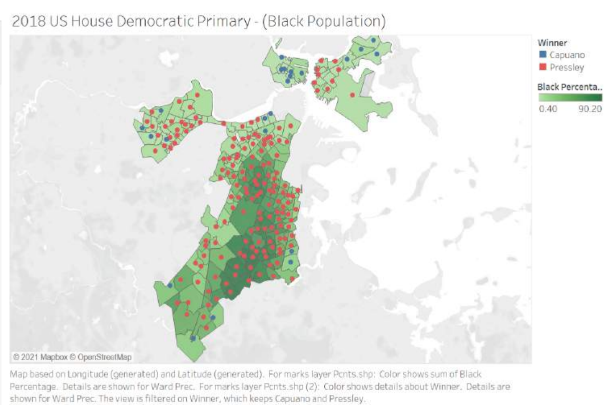
In both the 2017 Mayoral Races and the 2018 DA Race, there seems to be a slight correlation between the median income of a given precinct and a Black candidate winning said precinct. By contrast, however, this does not seem to be the case in the 2018 U.S. House Democratic Primary, as Ayanna Pressley won the majority of precincts regardless of median income.

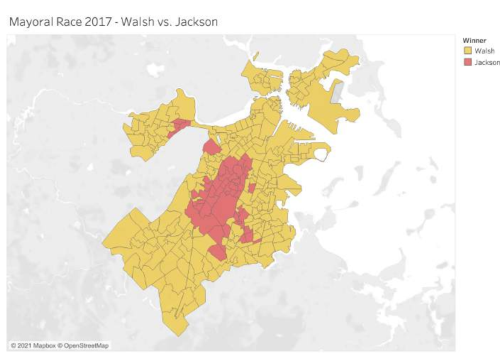
Part 3: Educational Attainment

2018 Educational Attainment









Source:
https://data.census.gov/cedsci/table?t=Education&g=0500000US25025.140000&tid=ACSST5Y2019.S1501&moe=false&tp=false&hidePreview=false

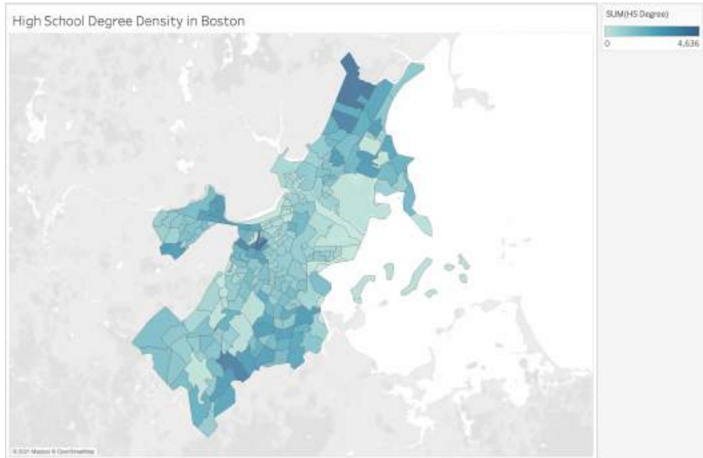
We now analyze any correlation between the average educational attainment of a given precinct and the performance of Black candidates in that precinct:

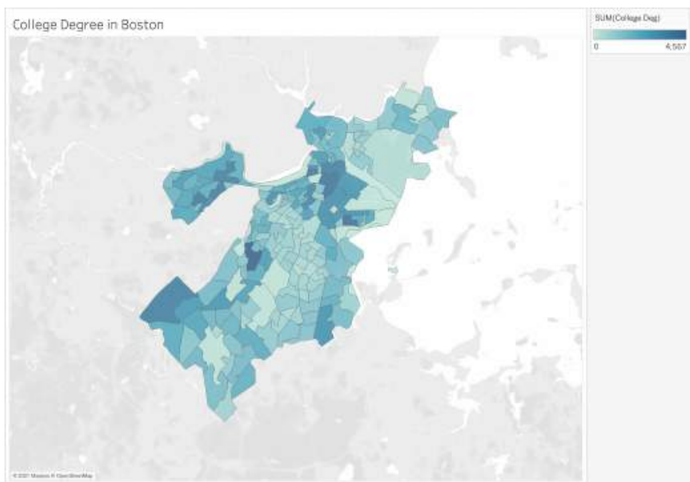
2017 Mayoral Race:
In most precincts where Tito Jackson won, the average educational attainment of the residents was a high school diploma. By contrast, Walsh won almost all districts in which the average educational attainment is a college degree.

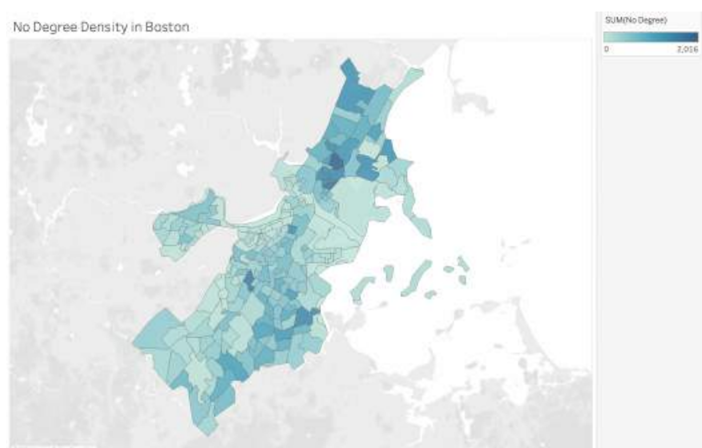
2018 DA Race:
From the data visualization of the educational attainment in Boston compared with the candidate won in each ward-precincts visualization, we could see that most of Henning's precinct that he won over have high density of college degree population; however, Rollin's encompass wide variety of educational attainment population; therefore, it is hard to find correlation between the two variables.

2018 US Senate Democratic Primary:
Again, there is no correlation between the precincts which voted for Pressley and educational attainment, as Pressely won the overwhelming majority of districts regardless of educational attainment.

Lastly we visualize the education levels across all Boston precincts as a gradient:







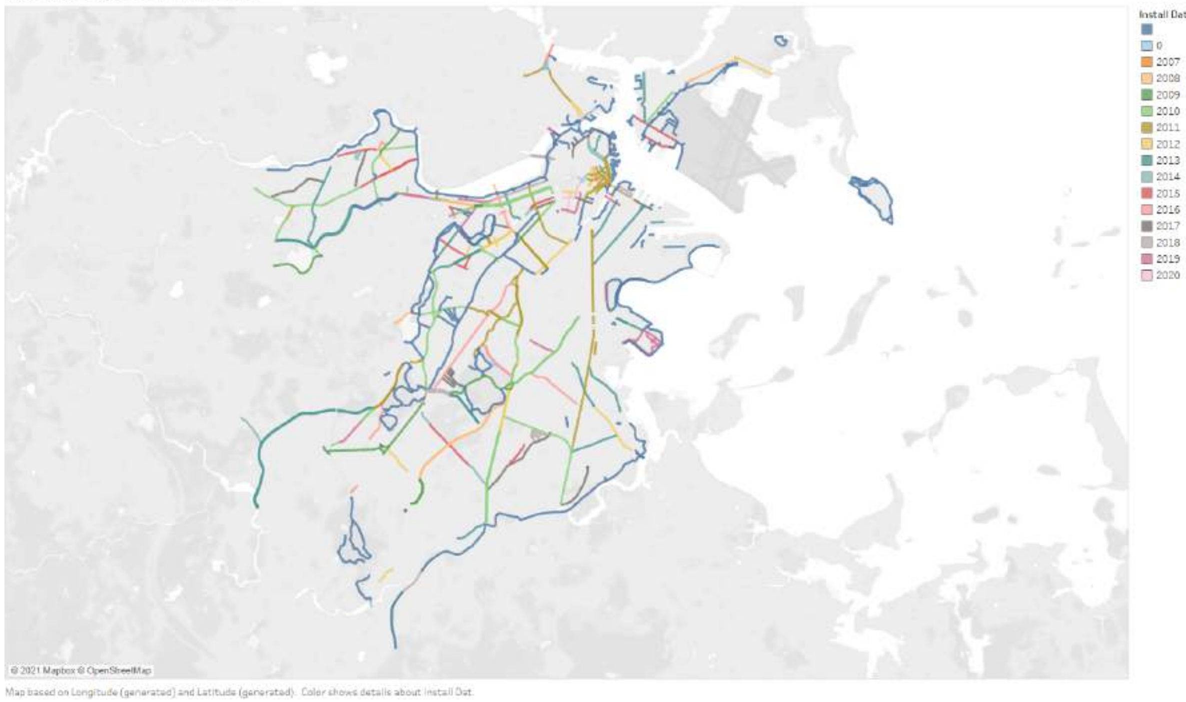
Does more houses affect voting participation for black candidates?

Housing Data

Map based on Longitude (generated) and Latitude (generated).

The Bottom of Boston's Building population is around 82% less populated. The Very tip of Boston's Building population is around 56% less populated. This is not a key predictor of black voting participation. There is not a correlation with the amount of houses for both 2017 General mayoral races where Johnson got most votes in the center of the map. And there is no correlation when Pressley received the most votes during the US Democratic party.

Bike Lanes Based On Install Date



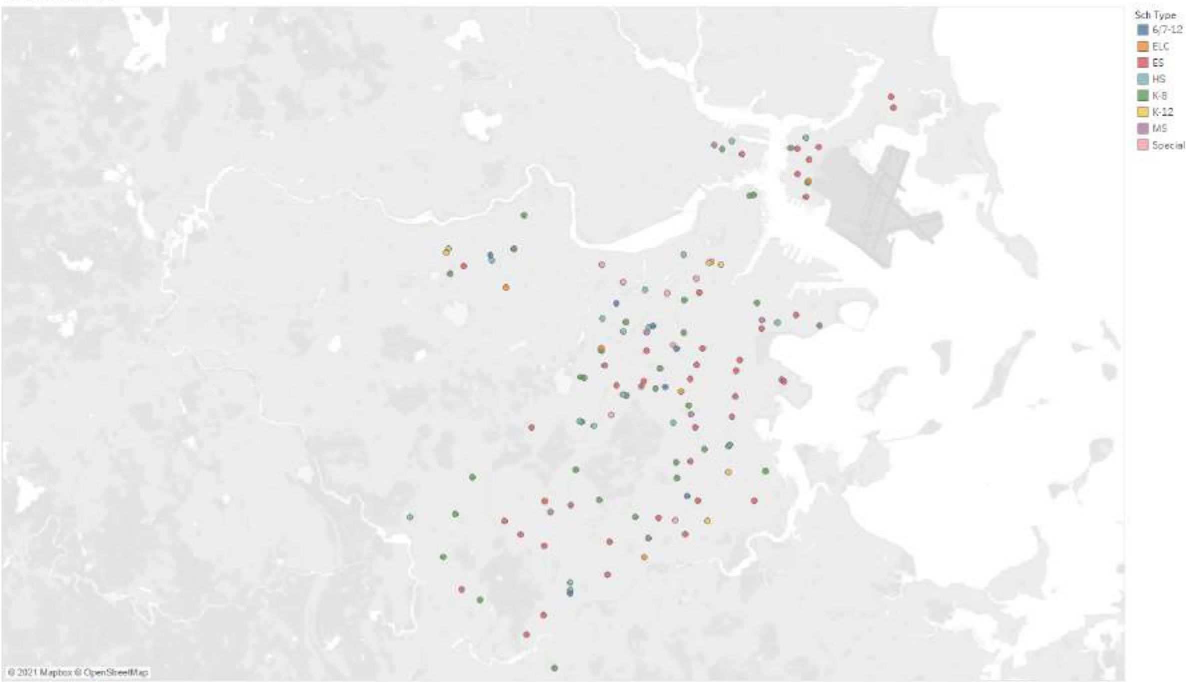
We do not believe bike lanes installation dates affect support for Black candidates. There is no clear correlation for bike lanes and the 2017 General Mayoral Race nor the US House Democratic Party.

Bike Lanes Based On Jurisdiction

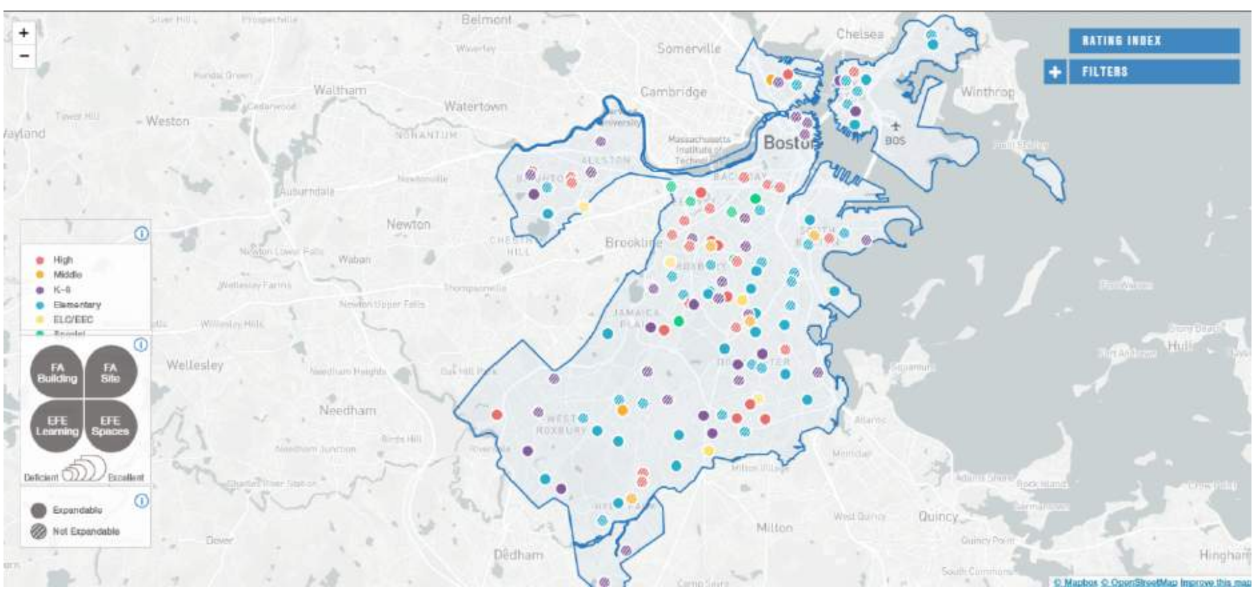
We do not believe bike lanes affect support for Black candidates. There is no clear correlation for bike lanes and the 2017 General Mayoral Race nor the US House Democratic Party. This shows that there is more transit on the left side of the map.

Does Public Schools Affect the Support for black candidates?

Public Schools



Map based on Longitude (generated) and Latitude (generated). Color shows details about Sch Type.



https://data.boston.gov/showcase/buildbps-dashboard

There is no clear correlation for the number of schools and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race. This shows that there are more schools in the center/top of the map.

Do Park Trees Affect the Support for black candidates?

Park Trees

Map based on Longitude (generated) and Latitude (generated). Color shows details about Type. The view is filtered on Type, which keeps PARK-TREE.

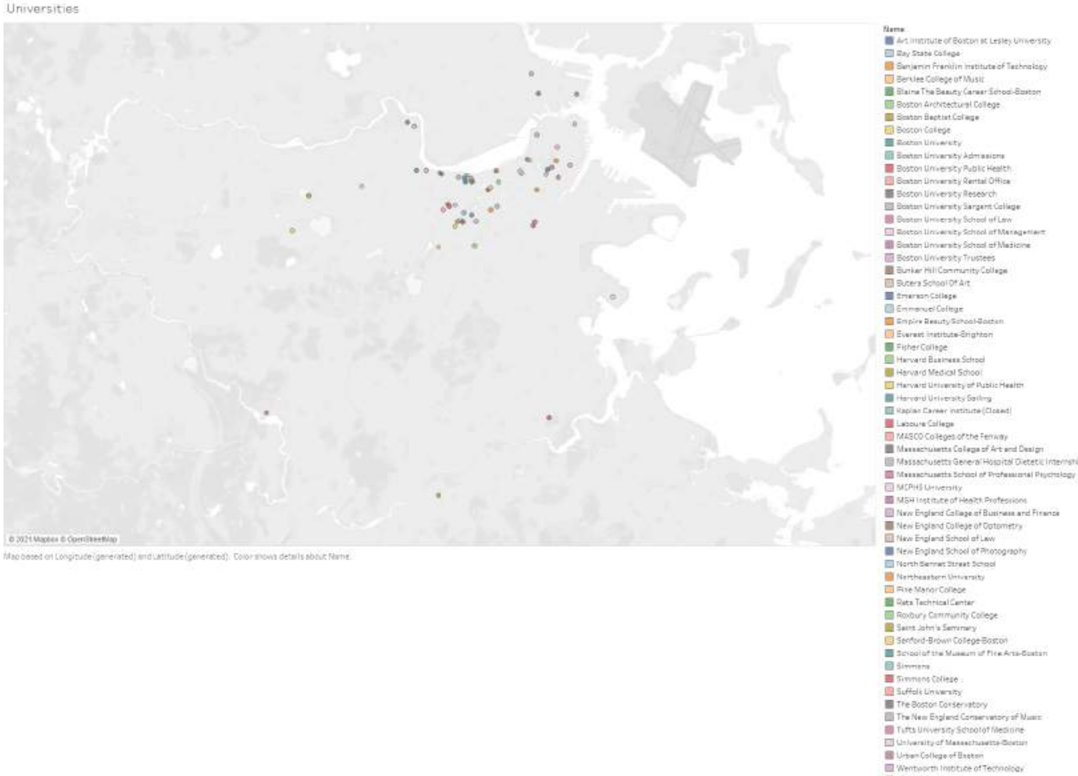
No clear correlation.There is no clear correlation for park trees and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race. This shows that there are a few spots that are empty/void of trees, which does not correlate to voting data.

Street Trees

Map based on Longitude (generated) and Latitude (generated). Color shows details about Type. The view is filtered on Type, which keeps STREET-TREE.

Street trees do not have a correlation. There is no clear correlation for street trees and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race. This shows that there are a few spots that are empty/void of trees, which does not correlate to voting data.

Universities



Map based on Longitude (generated) and Latitude (generated). Color shows details about Name.

Universities do not affect support for black candidates.

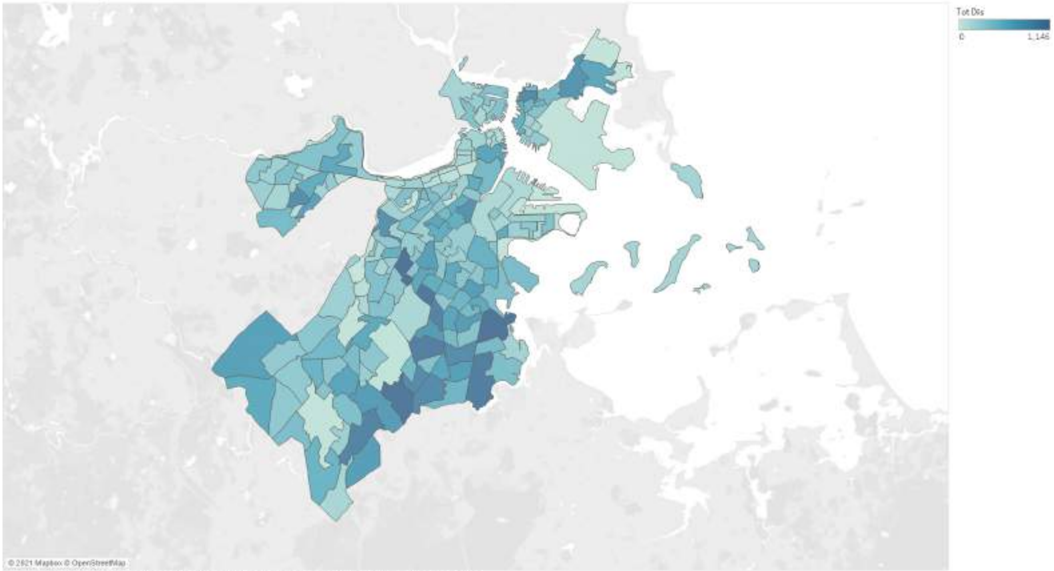
For the following Tableaus Social vulnerability during a climate event:
Population Definitions:
Older Adults:
Older adults (those over age 65) have physical vulnerabilities in a climate event; they suffer from higher rates of medical illness than the rest of the population and can have some functional limitations in an evacuation scenario, as well as when preparing for and recovering from a disaster. Furthermore, older adults are physically more vulnerable to the impacts of extreme heat. Beyond the physical risk, older adults are more likely to be socially isolated. Without an appropriate support network, an initially small risk could be exacerbated if an older adult is not able to get help.
Data source: 2008-2012 American Community Survey 5-year Estimates (ACS) data by census tract for population over 65 years of age.
Children:
Families with children require additional resources in a climate event. When school is cancelled, parents need alternative childcare options, which can mean missing work. Children are especially vulnerable to extreme heat and stress following a natural disaster.
Data source: 2010 American Community Survey 5-year Estimates (ACS) data by census tract for population under 5 years of age.
People of Color:
People of color make up a majority (53 percent) of Boston's population. People of color are more likely to fall into multiple vulnerable groups as
well. People of color statistically have lower levels of income and higher levels of poverty than the population at large. People of color, many of whom also have limited English proficiency, may not have ready access in their primary language to information about the dangers of extreme heat or about cooling center resources. This risk to extreme heat can be compounded by the fact that people of color often live in more densely populated urban areas that are at higher risk for heat exposure due to the urban heat island effect.
Data source: 2008-2012 American Community Survey 5-year Estimates (ACS) data by census tract: Black, Native American, Asian, Island, Other, Multi, Non-white Hispanics.
Limited English Proficiency:
Without adequate English skills, residents can miss crucial information on how to prepare
for hazards. Cultural practices for information sharing, for example, may focus on word-of-mouth communication. In a flood event, residents can also face challenges communicating with emergency response personnel. If residents are more socially
isolated, they may be less likely to hear about upcoming events. Finally, immigrants, especially ones who are undocumented, may be reluctant to use government services out of fear of deportation or general distrust of the government or emergency personnel.
Data Source: 2008-2012 American Community Survey 5-year Estimates (ACS) data by census tract, defined as speaks English only or speaks English "very well".
Low to no Income:
A lack of financial resources impacts a household's ability to prepare for a disaster event and to support friends and neighborhoods. For example, residents without televisions, computers, or data-driven mobile phones may face challenges getting news about hazards or recovery resources. Renters may have trouble finding and paying deposits for replacement housing if their residence is impacted by flooding. Homeowners may be less able to afford insurance that will cover flood damage. Having low or no income can create difficulty evacuating in a disaster event because of a higher reliance on public transportation. If unable to evacuate, residents may be more at risk without supplies to stay in their homes for an extended period of time. Low- and no-income residents can also be more vulnerable to hot weather if running air conditioning or fans puts utility costs out of reach.
Data source: 2008-2012 American Community Survey 5-year Estimates (ACS) data by census tract for low-to- no income populations. The data represents a calculated field that combines people who were 100% below the poverty level and those who were 100–149% of the poverty level.
People with Disabilities:
People with disabilities are among the most vulnerable in an emergency; they sustain disproportionate rates of illness, injury, and death in disaster events.46 People with disabilities can find it difficult to adequately prepare for a disaster event, including moving to a safer place. They are more likely to be left behind or abandoned during evacuations. Rescue and relief resources—like emergency transportation or shelters, for example— may not be universally accessible. Research has revealed a historic pattern of discrimination against people with disabilities in times of resource scarcity, like after a major storm and flood.
Data source: 2008-2012 American Community Survey 5-year Estimates (ACS) data by census tract for total civilian non-institutionalized population, including: hearing difficulty, vision difficulty, cognitive difficulty, ambulatory difficulty, self-care difficulty, and independent living difficulty.
Medical Illness:

Symptoms of existing medical illnesses are often exacerbated by hot temperatures. For example, heat can trigger asthma attacks or increase already high blood pressure due to the stress of high temperatures put on the body. Climate events can interrupt access to normal sources of healthcare and even life-sustaining medication. Special planning is required for people experiencing medical illness. For example, people dependent on dialysis will have different evacuation and care needs than other Boston residents in a climate event.

Data source: Medical illness is a proxy measure which is based on EASI data accessed through Simply Map. Health data at the local level in Massachusetts is not available beyond zip codes. EASI modeled the health statistics for the U.S. population based upon age, sex, and race probabilities using U.S. Census Bureau data. The probabilities are modeled against the census and current year and five year forecasts. Medical illness is the sum of asthma in children, asthma in adults, heart disease, emphysema, bronchitis, cancer, diabetes, kidney disease, and liver disease. A limitation is that these numbers may be over-counted as the result of people potentially having more than one medical illness. Therefore, the analysis may have greater numbers of people with medical illness within census tracts than actually present. Overall, the analysis was based on the relationship between social factors.

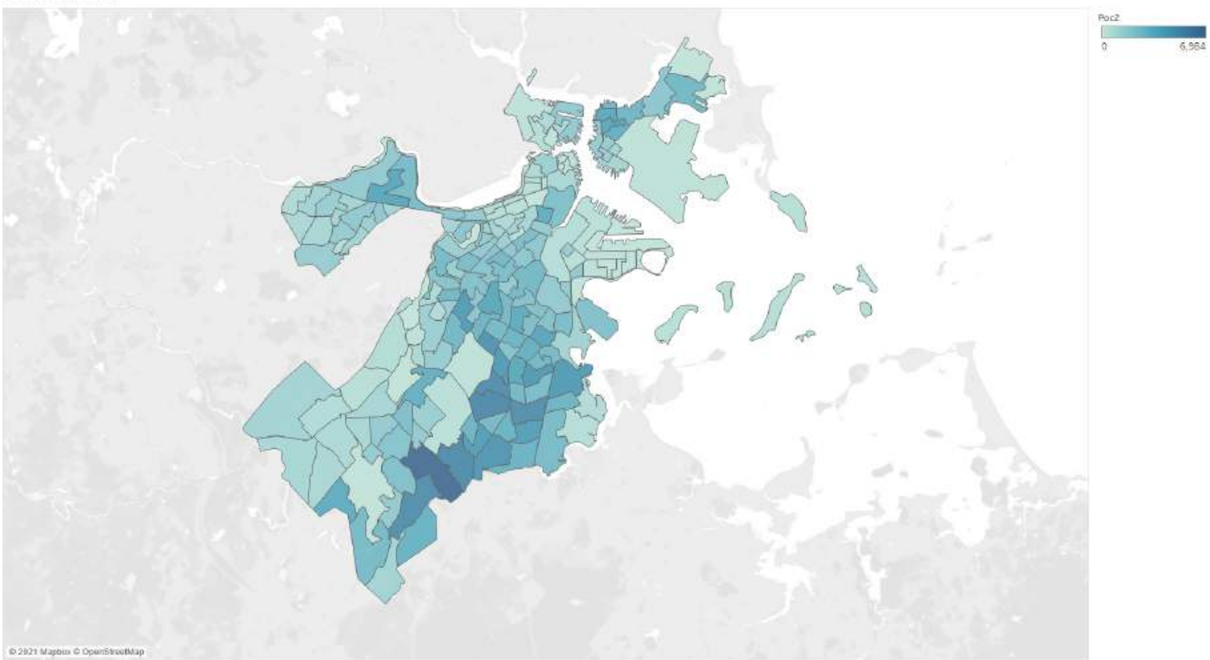
Attribute label: MedIllnes

Total Disability



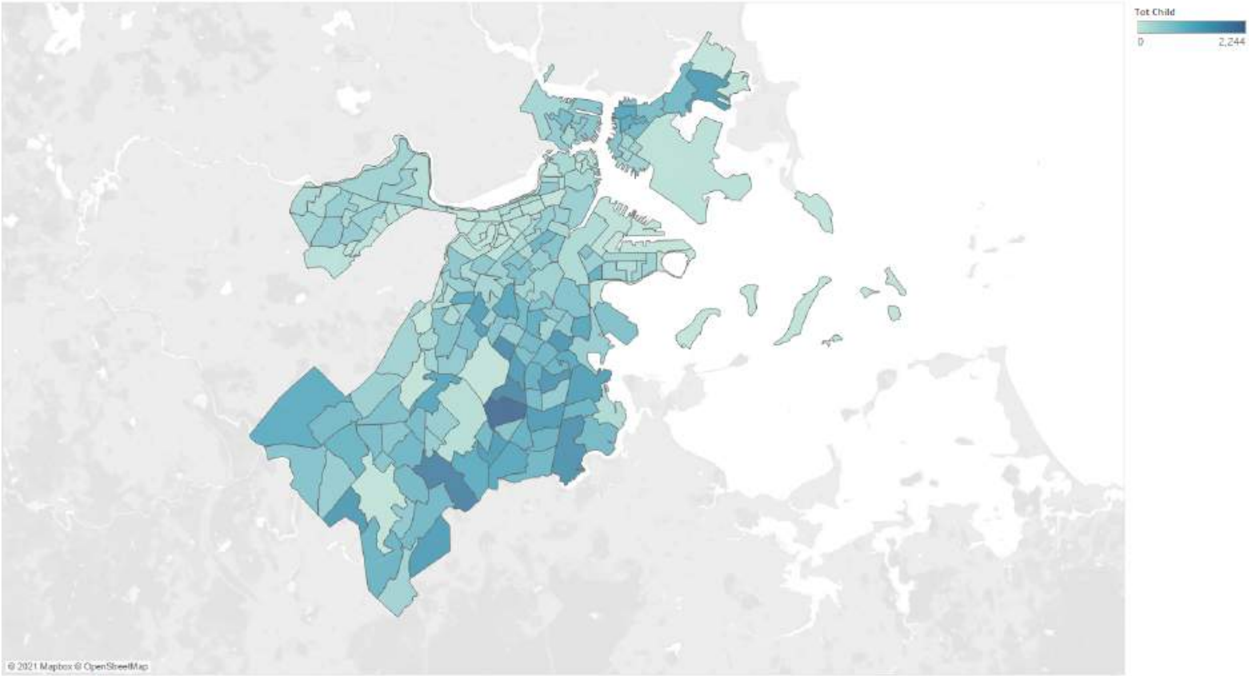
Map based on Longitude (generated) and Latitude (generated). Color shows sum of Tot Dis. Details are shown for Geoid10.

People of Color



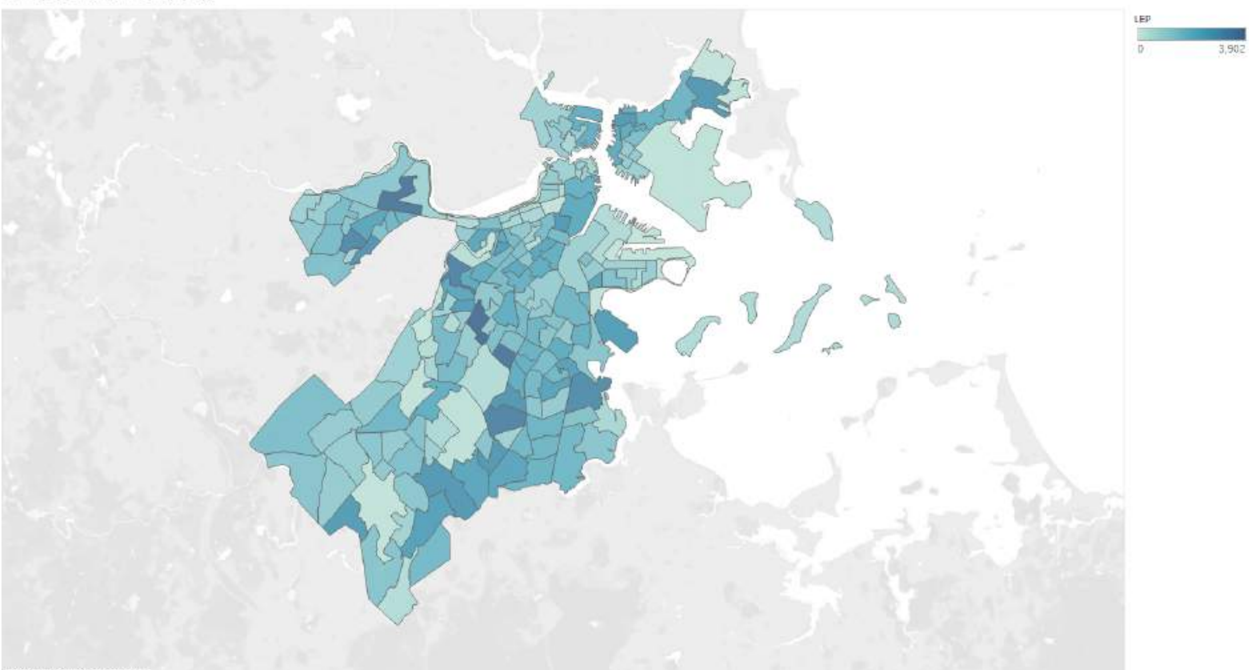
Map based on Longitude (generated) and Latitude (generated). Color shows sum of Poc2. Details are shown for Geoid10.

Children



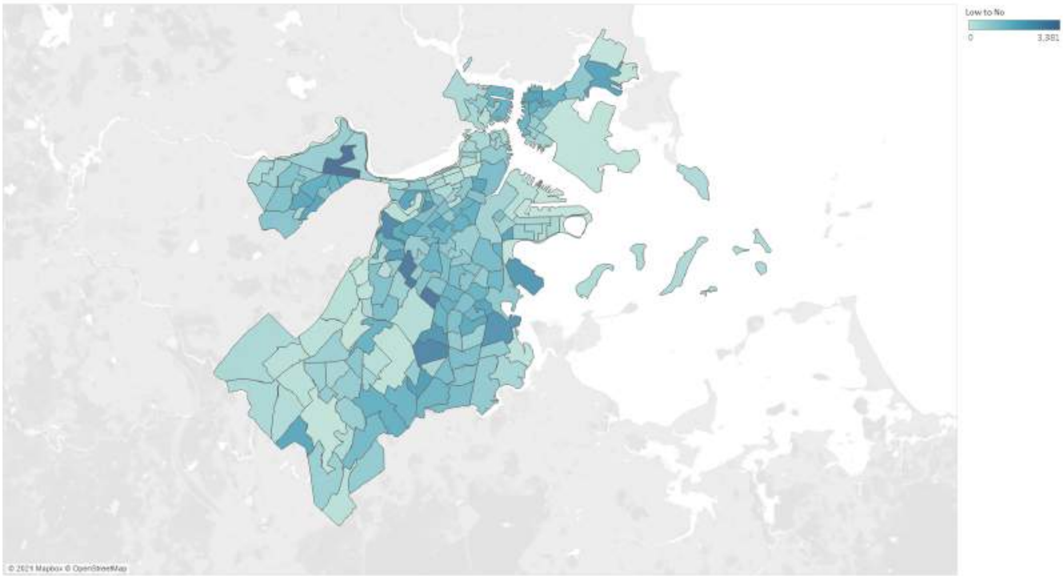
Map based on Longitude (generated) and Latitude (generated). Color shows sum of Tot Child. Details are shown for Geoid10.

Limited English Proficiency



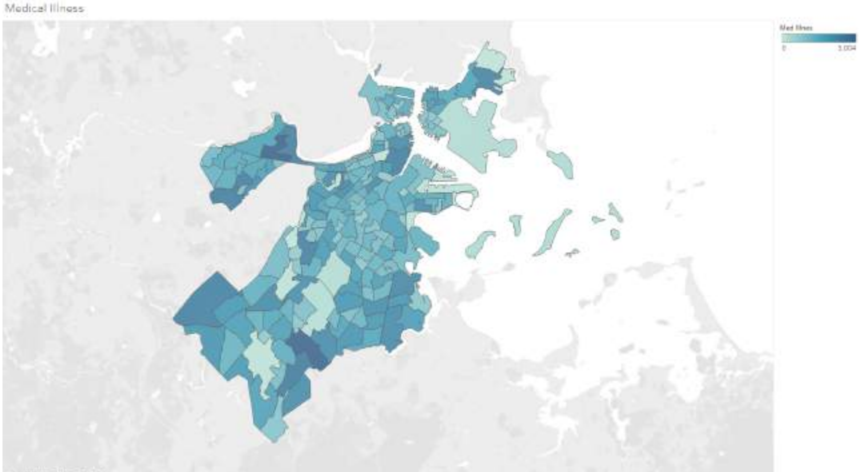
Map based on Longitude (generated) and Latitude (generated). Color shows sum of LEP. Details are shown for Geoid10.

Low to No Income



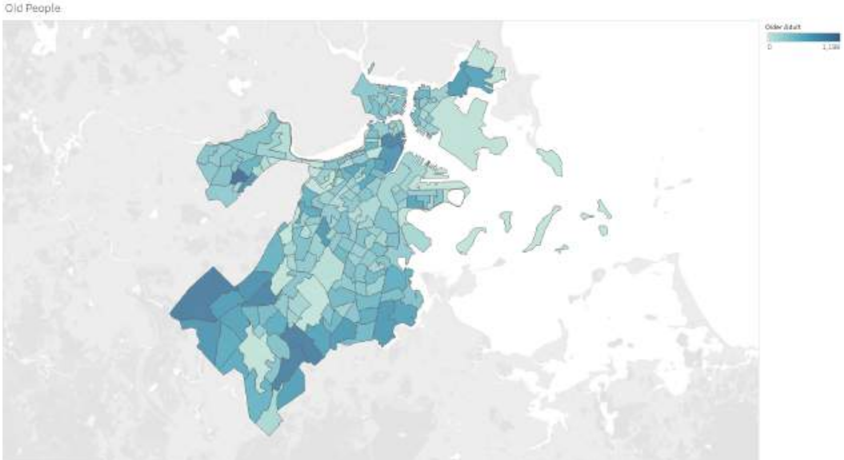
Map based on Longitude (generated) and Latitude (generated). Color shows sum of Low to No. Details are shown for Geoid10.

Medical Illness



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Med Illnes. Details are shown for Geoid10.

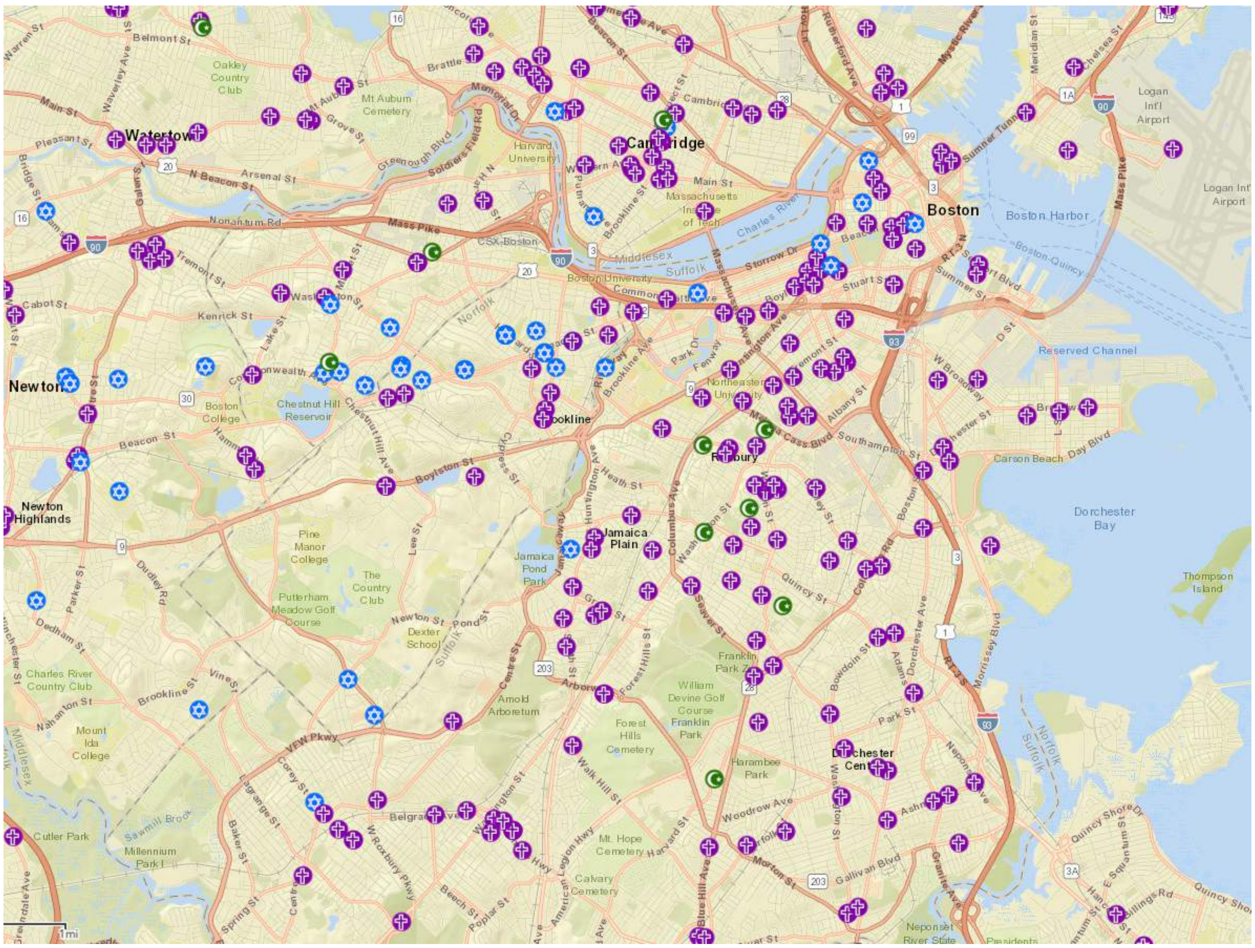
Old People



Map based on Longitude (generated) and Latitude (generated). Color shows sum of Older Adult. Details are shown for Geoid10.

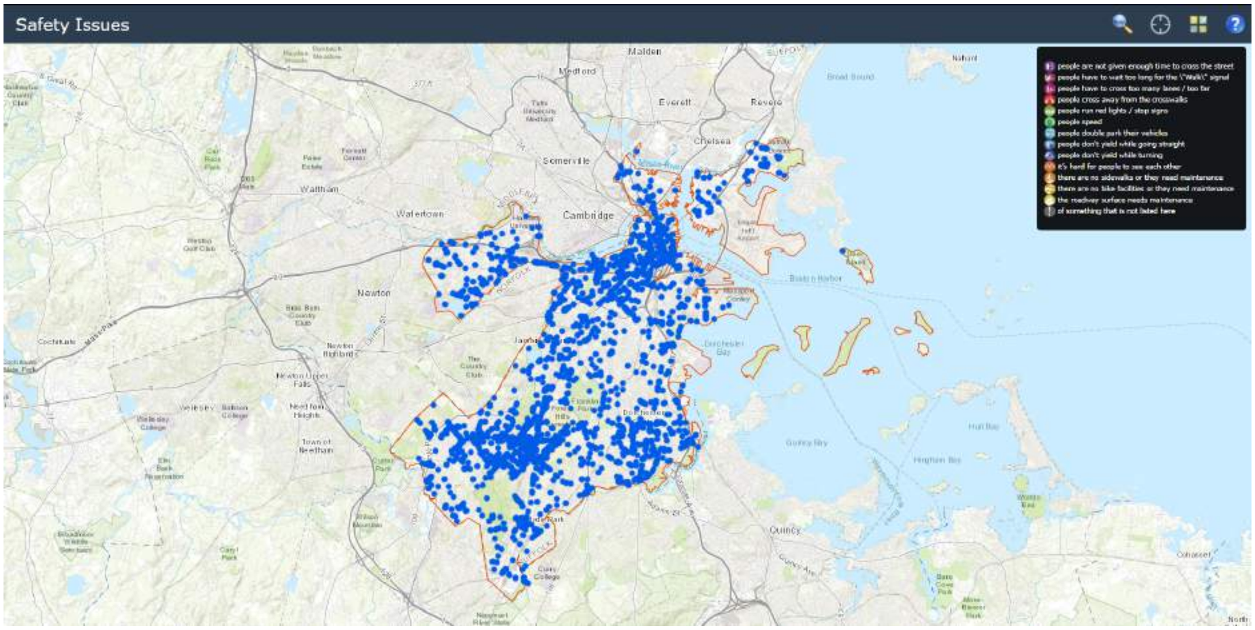
There is no clear correlation for Total disability/People of Color/Old People/ Medical Illness/Low To No Income/ English Proficiency and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race. What this shows is that Disabilities and Medical Illnesses are more apparent and are more prone to climate changes. There might be a slight correlation between people low to no income vulnerable to climate change voting for Jackson in the 2017 General Mayoral Race.
We see that the south end of boston is the most prone to climate, there's no clear correlation.

Boston Churches:



https://massgis.maps.arcgis.com/apps/webappviewer/index.html?id=9fc18ba7901546a1b342393b6486cf31
There is no clear correlation for Number of Churches and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race.
Link to More Geospatial data (Not Correlated):
https://docs.digital.mass.gov/dataset/massgis-data-layers

311 Reports Trash areas in Boston
https://jhaddadin.github.io/trashcity/garbagemap.html
There is no clear correlation for Trash Areas and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race.



http://app01.cityofboston.gov/VZSafety/

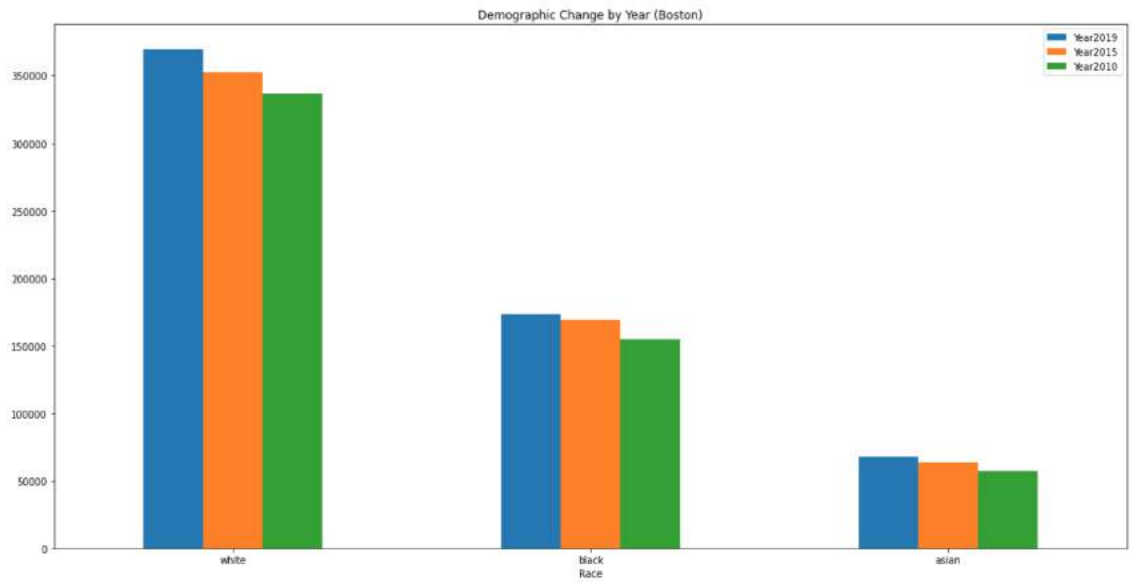
There is no clear correlation for Safety Issues and the 2017 General Mayoral Race nor the US House Democratic Party nor the 2018 DA race.

Key Question: How have the demographic changes in Boston affected local elections over the past decade?

Part 1: Examining the Demographic Changes in Boston over the Past Decade

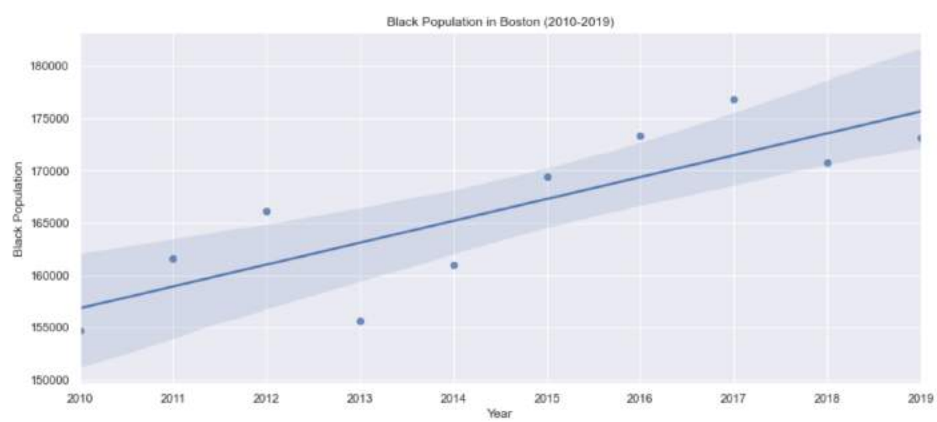
Using census data, we give an overview of how the demographics have changed in Boston in the past decade. Note that this analysis is not on the precinct level, since the data used is from in-between census years using the American Community Survey. Additionally, this data is only a rough estimate of the actual population for this same reason.

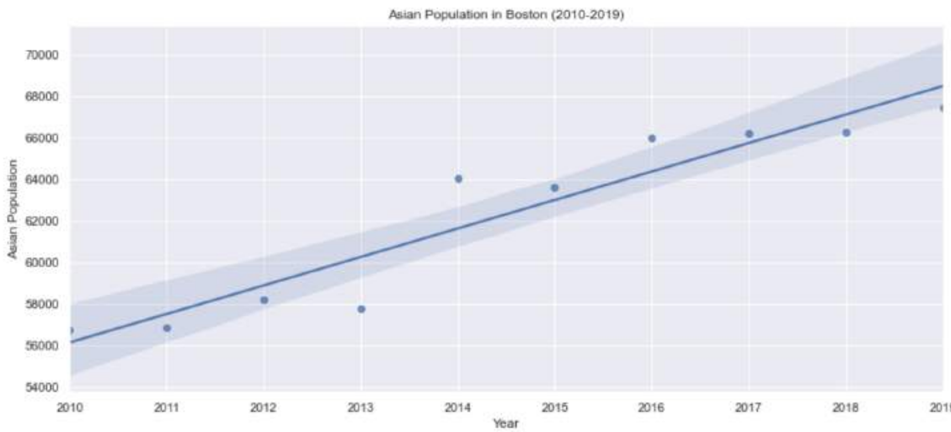
First, we examine the raw numbers of white, Black, and Asian populations for the years 2010, 2015, and 2019.



Here, we find that each of these demographics have grown in Boston throughout the past 10 years.

We also visualize the growth of Black and Asian populations using the following charts:



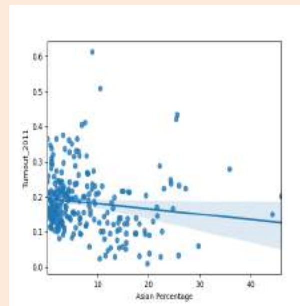
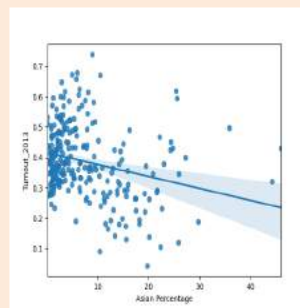
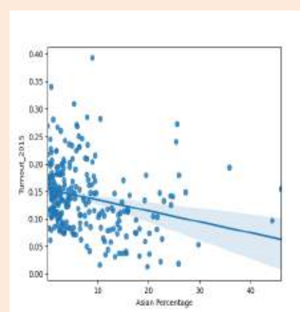
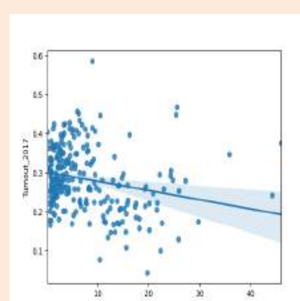


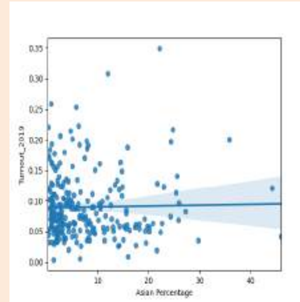
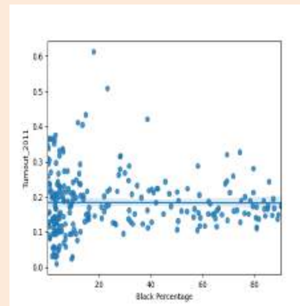
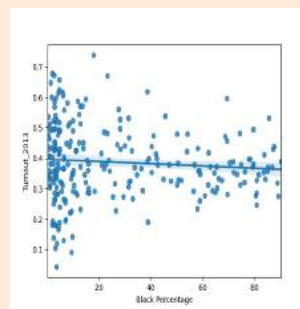
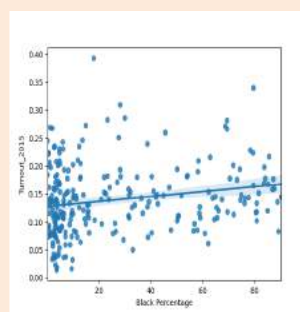
Clearly, both demographics are on an upward trend of growth. In the past 10 years, the Black population in Boston has grown by around 20,000 people, while the Asian population in Boston has grown by approximately 12,000 people.

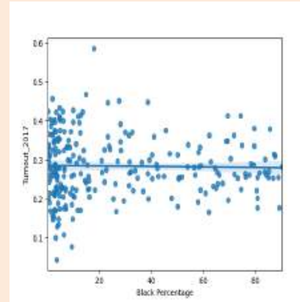
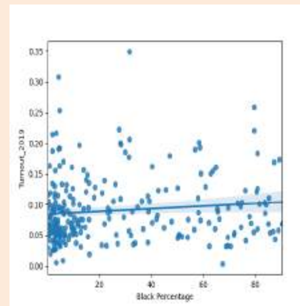
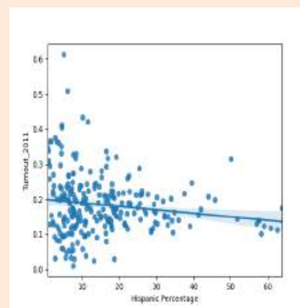
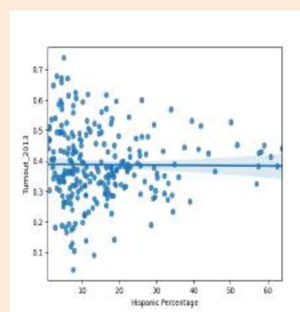
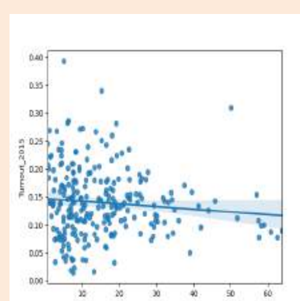
Now, we will examine the performance of Black candidates over key elections in the past decade. While we cannot determine causation, we will be able to examine if there is any correlation between the growth in Black population and the performance of these candidates over the years.

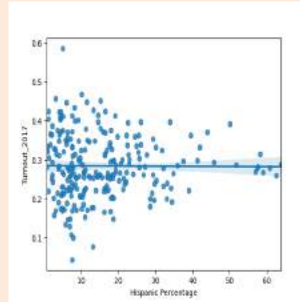
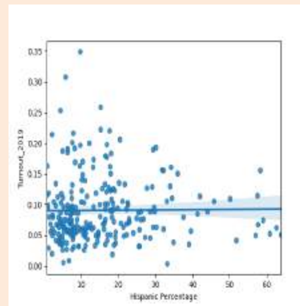
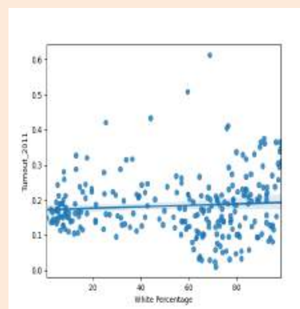
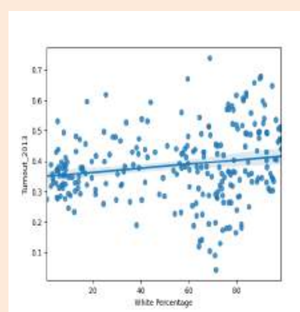
Part 2: Regression Analysis of City Council Voter Turnout Percentages by a given Race's Percentage in Each Precinct

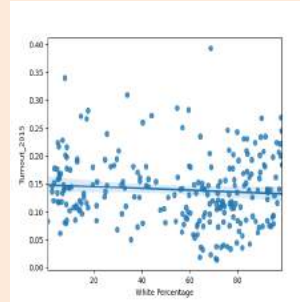
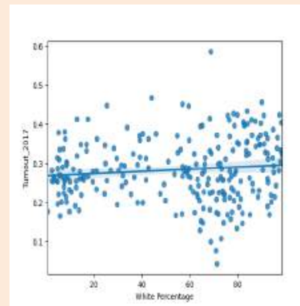
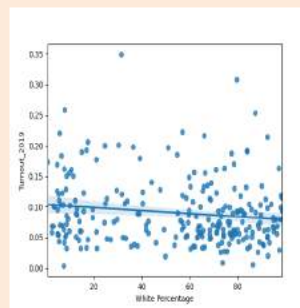
In this section, we examine whether or not there is any correlation between voter turnout and the percentage of a given race in each precinct. For this analysis, we examine the turnouts of the City Council elections of the years 2011 to 2019.

Year	Race/Ethnicity	Description	Graph
2011	Asian	We find a slight negative correlation between Asian percentage and percent Turnout for CC Election 2011, however, it is not significant.	
2013	Asian	We find a slight negative correlation between Asian percentage and percent Turnout for CC Election 2013.	
2015	Asian	We find a slight negative correlation between Asian percentage and percent Turnout for CC Election 2015, however, it is not significant.	
2017	Asian	We find a slight negative correlation between Asian percentage and percent Turnout for CC Election 2017, however, it is not significant.	

2019	Asian	We find no real correlation between Asian percentage and percent Turnout for CC Election 2019.	
2011	Black	We find no real correlation between Black percentage and percent Turnout for CC Election 2011.	
2013	Black	We find no real correlation between Black percentage and percent Turnout for CC Election 2013.	
2015	Black	We find a slight positive correlation between Black percentage and percent Turnout for CC Election 2015, however, it is not significant.	
2017	Black	We find no real correlation between Black percentage and percent Turnout for CC Election 2017.	

			
2019	Black	We find no real correlation between Black percentage and percent Turnout for CC Election 2019.	
2011	Hispanic	We find a slight negative correlation between Hispanic percentage and percent Turnout for CC Election 2011, however, it is not significant.	
2013	Hispanic	We find no real correlation between Hispanic percentage and percent Turnout for CC Election 2013.	
2015	Hispanic	We find no real correlation between Hispanic percentage and percent Turnout for CC Election 2015.	

2017	Hispanic	We find no real correlation between Hispanic percentage and percent Turnout for CC Election 2017.	
2019	Hispanic	We find no real correlation between Hispanic percentage and percent Turnout for CC Election 2019.	
2011	White	We find no real correlation between White percentage and percent Turnout for CC Election 2011.	
2013	White	We find a slight positive correlation between White percentage and percent Turnout for CC Election 2013, however, it is not significant.	
2015	White	We find no real correlation between White percentage and percent Turnout for CC Election 2015.	

			
2017	White	We find no real correlation between White percentage and percent Turnout for CC Election 2017.	
2019	White	We find no real correlation between White percentage and percent Turnout for CC Election 2019.	

Summary of results:

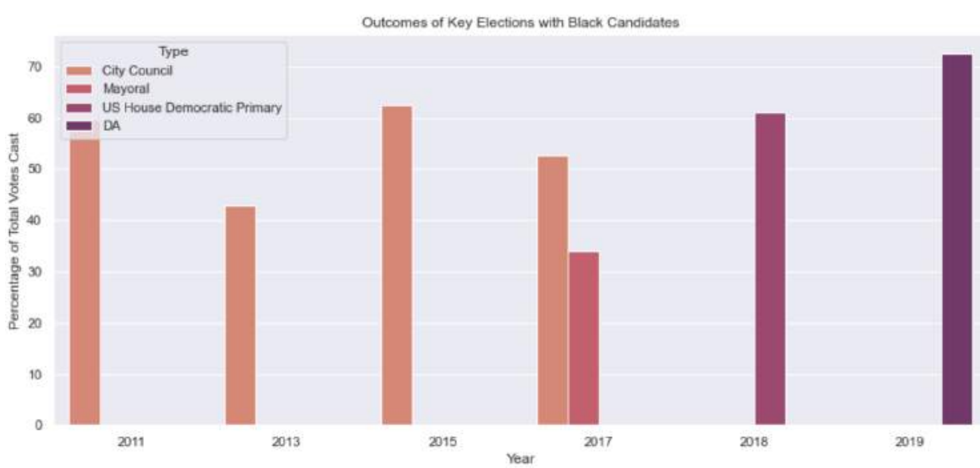
- For CC 2011 through CC 2017, we found that precincts with higher Asian populations tended to have lower voter turnout. However, in 2019, this changed, as we no longer found a negative correlation.
    - This coincides with the growing Asian population in Boston, but causation cannot be determined.
- We found no real relationship between precincts with high Black populations and voter turnout, except perhaps in 2015, when there was a slight positive correlation between the two.
- We found no correlation between precincts with high Hispanic populations and voter turnout.
- We found no correlation between precincts with high White populations and voter turnout.

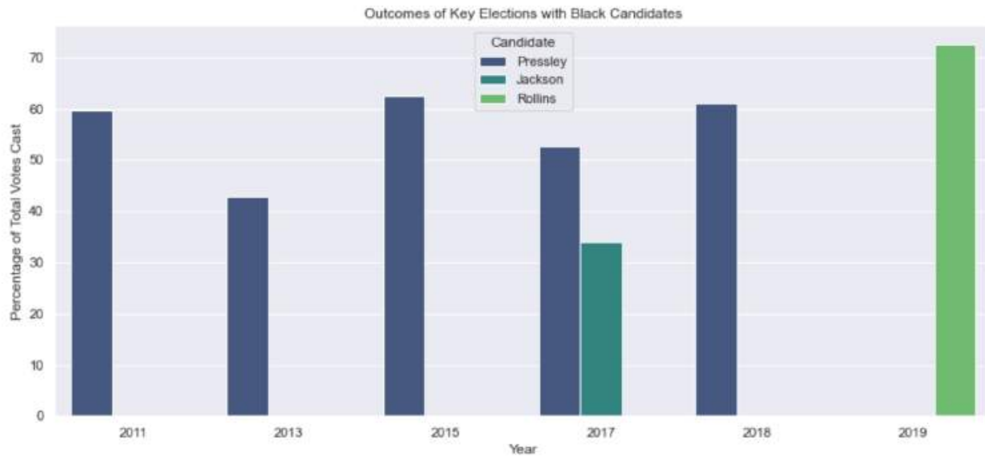
Part 3: The Performance of Key Black Candidates Over the Past Decade

In this section, we will be examining the following elections and Black candidates while keeping in mind our findings in Part 1 that the Black population in Boston has been growing over the past decade:

Year	Type	Candidate	Percentage
2011	City Council	Pressley	59.57
2013	City Council	Pressley	42.81
2015	City Council	Pressley	62.51
2017	City Council	Pressley	52.75
2017	Mayoral	Jackson	33.97
2018	US House Democratic Primary	Pressley	61.12
2019	DA	Rollins	72.60

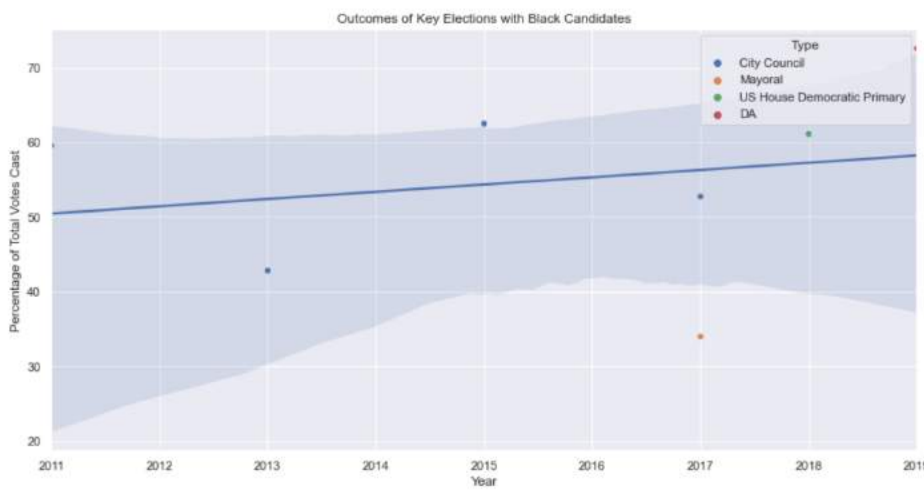
Visualizing their performance:





From these bar charts, we find no clear trend in the performance of Black candidates across each year.

However, we also fit a regression line to determine if there has been any trend:



As we can see, there has been a very slight positive trend in the performance of Black candidates over the past decade, however it is not significant. This begs the question of whether or not this has any relationship with the growing Black population in Boston, as we saw from our earlier analysis.

Key Questions and Summary of Results

Overarching Project Question: Are there any overarching racial voting patterns in Boston on the Ward-Precinct level, and how does this vary across election type over the past decade?

- We have found a variety of racial voting patterns on the Ward-Precinct level in Boston across election type during the past decade.
- Most of the answers to this question are incorporated into our key questions that we explore below.

Key Questions and Principal Findings:

1. How do racial voting patterns differ by election type?
    a. Generally, across election types, Black candidates tend to receive support from precincts with high Black populations.
    b. Turnout for city council elections which coincide mayoral elections is generally much higher than turnout for stand-alone city council elections.
    c. For certain elections, such as the 2018 U.S. House Democratic primary, the race of the candidate does not seem to be a primary factor in determining whether or not the candidate will win a specific precinct.

2. What are the key predictors in determining support for Black candidates for Boston?
    a. There is a strong positive correlation between the percentage of votes Black candidates receive and the higher the Black population in a specific precinct.
    b. There seems to be a slight correlation between certain Black candidates (Tito Jackson and DA Rollins) winning a given precinct and that precinct's median income and average educational attainment.

3. How has voter turnout by precinct changed across city council election year?
    a. Precincts which have the most volatile voter turnout are concentrated in East Boston and West Roxbury and are either majority Hispanic or majority white.
    b. Precincts which have experienced the greatest change in share of voter turnout are more dispersed ethnically, and are concentrated in areas of Dorchester and Downtown Boston.
    c. Michael F. Flaherty is in the election from 2015 to 2019. His share of voter turnout decreases each year. His share of voter turnout, in general, decreases from 2015 to 2019.
    d. 13 out of 20 precincts which have experienced the greatest change in share of Michael F. Flaherty's voter turnout have a large white population (above 50%). Flaherty's share of voter change's trends in precincts with most white populations of each year are opposite than those in precincts with lowest white populations.
    e. Michelle Wu is in the election from 2013 to 2019. Her share of voter turnout only decreased in 2019 and increased in general from 2013 to 2019.
    f. Precincts which have experienced the greatest change in share of Michelle Wu's voter turnout all have a large white population (above 50%). Wu has the most share of voters in 2017.

g. In 2019 we can see a huge decrease of the share of voters in precincts with most white, Hispanic and Asian populations. However, there is no decrease, and even some increases, in precincts with most Black populations.
h. For precincts with a majority Black population, there has been a consistent improvement in turnouts in relative terms compared to other precincts over the 2010-2019 decade.
i. In general, Annissa E. George has more supporters in 2019 than in 2013. Only in 2017, she lost supporters compared to the last election (2015). In other years, she had a larger share of voters than last election.
j. Annissa E. George has the most share of voters in 2019 in precincts with the most Black population. There is a huge increase in Annissa's share of voters in 2015 than that in 2013. Her share of voters in precincts with most Black population didn't decrease as much as that in other precincts in 2017 and 2019.
k. In 2019, Annissa E. George's share of voters increased in precincts with the most Black population but decreased in general in other precincts dominated by other race populations.

4. What are changes in District 3 particularly?
   a. Those precincts which experience the greatest volatility and change in share of voter turnout across City Council election year tend to be concentrated in the southern part of District 3. Additionally, these precincts tend to either be ethnically varied or have a very low POC population.
   b. The trends in Michael F. Flaherty's voter turnout for District 3 is the same as the general pattern. Precincts which have experienced the greatest change in share of Michael F. Flaherty's voter turnout do not have a strong correlation with a single race population.
   c. The trends in Michelle Wu's voter turnout for District 3 follow the same general pattern as Boston overall.
   d. Overall, the southern precincts in District 3 tend to have much higher turnouts than southern areas. And recently, precincts with high Black populations have seen higher turnout than earlier ones.
   e. District 3's share of votes in Boston has been trending downwards over the past decade.

5. How have the demographic changes in Boston affected local elections over the past decade?
   a. Over the past decade, we found that Black, White, and Asian populations in Boston have steadily grown in numbers.
   b. Over the City Council years, we were unable to find any significant correlation between precincts with high Black, Hispanic, and White populations and high voter turnouts. However, we did find that for CC years 2011-2017, there was a negative correlation between precincts with high Asian populations and voter turnout, but this changed in CC election year 2019. This coincides with the growing Asian population in Boston, however, causation cannot be determined.
   c. In examining the performance of Black candidates in key Boston elections, we found that, over the past decade, there has been a slight upward trend in their performance. This coincides with the growing Black population, however, causation cannot be determined.

APPENDIX

All code used for this project will follow.

house_and_senate_prelim_analysis

April 29, 2021

[8]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def clean_data(filename): # removes any lines with nan values
    df = pd.read_csv(filename)
    #replacing empty entries with nan
    df.replace(r'', np.nan)
    #cleaning data
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

df =␣
 ↪clean_data("2020_U_S_Senate_Democratic_Primary_including_precincts_with_demographics.
 ↪csv")
df['Ward'] = df.Ward.astype(int)
df
```

[8]:
```
    City/Town  Ward   Pct  Edward J. Markey  Joseph P. Kennedy, III  \
0      Boston     1   1.0             373.0                   170.0
1      Boston     1   2.0             244.0                    90.0
2      Boston     1   3.0             538.0                   338.0
3      Boston     1   4.0             200.0                   124.0
4      Boston     1   5.0             259.0                   121.0
..        …     …    …               …                      …
248    Boston    22   8.0             268.0                   122.0
249    Boston    22   9.0             367.0                   134.0
250    Boston    22  10.0             478.0                   168.0
251    Boston    22  11.0             287.0                   119.0
252    Boston    22  12.0             263.0                   145.0

     All Others  Blanks  Total Votes Cast   Total   White  …  \
0           1.0     2.0             546.0  1717.0  1369.0  …
1           3.0     5.0             342.0  2071.0  1520.0  …
2           1.0     4.0             881.0  3179.0  1989.0  …
3           0.0     1.0             325.0  1884.0  1097.0  …
```

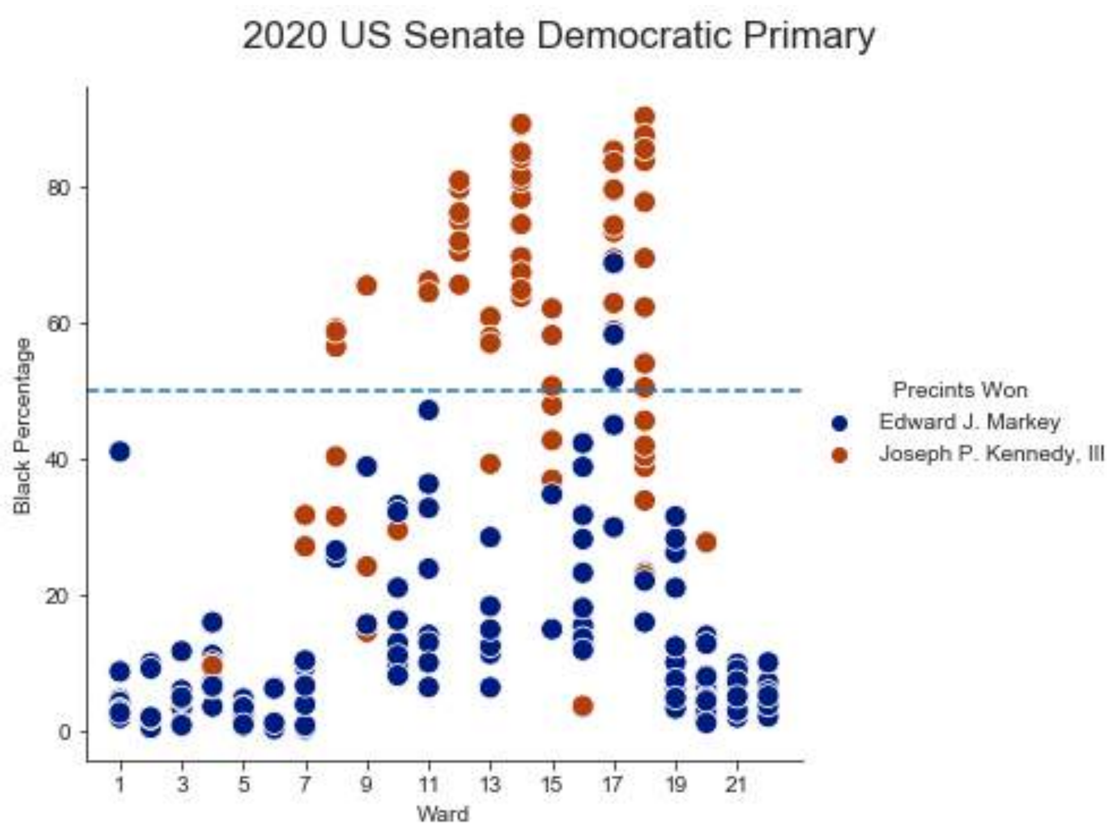
```
4          1.0     3.0            384.0  2888.0  1659.0  …
..         …      …            …      …      …  …
248        1.0     1.0            392.0  2065.0  1583.0  …
249        1.0    11.0            513.0  2018.0  1564.0  …
250        1.0     6.0            653.0  2442.0  1984.0  …
251        1.0     2.0            409.0  1315.0  1019.0  …
252        2.0     4.0            414.0  1646.0  1007.0  …

     Some Other Race/Ethnicity (alone)  Two or More Races/Ethnicities (alone)  \
0                                 27.0                                   27.0
1                                 53.0                                   33.0
2                                 57.0                                   89.0
3                                 59.0                                   24.0
4                                 93.0                                   40.0
..                                 …                                    …
248                                8.0                                   31.0
249                               11.0                                   52.0
250                               34.0                                   29.0
251                               12.0                                   28.0
252                               29.0                                   30.0

     Black Percentage  Native American Percentage  Asian Percentage  \
0                3.20                        0.41              2.85
1                2.95                        0.14              2.03
2                8.74                        0.31              7.17
3                3.03                        0.64              2.02
4                2.67                        0.76              2.63
..              …                         …               …
248              2.95                        0.05             17.29
249              1.98                        0.00             15.81
250              3.69                        0.04             10.24
251              5.02                        0.38             11.79
252              9.96                        0.49             14.70

     Native Hawaiian/Pacific Islander Percentage  Other Race Percentage  \
0                                           0.23                  10.08
1                                           0.00                  16.95
2                                           0.00                  15.54
3                                           0.05                  29.46
4                                           0.17                  31.02
..                                          …                     …
248                                         0.00                   1.26
249                                         0.10                   1.64
250                                         0.00                   2.99
251                                         0.00                   2.21
252                                         0.06                   9.60
```

```
     Two or more races Percentage  Hispanic Percentage  White Percentage
0                            3.49                29.30             79.73
1                            4.54                51.81             73.39
2                            5.66                41.33             62.57
3                            6.58                63.85             58.23
4                            5.30                57.62             57.44
..                            …                   …                …
248                          1.79                 5.23             76.66
249                          2.97                 4.91             77.50
250                          1.80                 8.48             81.24
251                          3.12                 8.29             77.49
252                          4.01                18.17             61.18

[253 rows x 33 columns]
```

[3]:
```
df['candidate'] = np.where(df['Edward J. Markey'] >= df['Joseph P. Kennedy,␣
↪III'], 'Edward J. Markey', 'Joseph P. Kennedy, III')
```

[4]:
```
sns.set_style("ticks")

g = sns.relplot(data=df, x="Ward", y="Black Percentage",␣
↪hue="candidate",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2020 US Senate Democratic Primary',fontsize=17)
g.legend.set_title("Precints Won")
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

[4]:
```
<matplotlib.lines.Line2D at 0x23ca04eec88>
```




```
[182]: g.savefig('2020_senate.png', dpi=300)
```

1 The Pressley & Capuano Dataset 2018

```
[5]: df2 = clean_data("2018_Pressley_Capuano_U_S_House_Democratic_with_demographics.
      ↪csv")
     df2['Ward'] = df2.Ward.astype(int)
     df2
```

```
[5]:     City/Town  Ward   Pct  Ayanna S. Pressley  Michael E. Capuano  All Others  \
     0      Boston     1   1.0               212.0               129.0         0.0
     1      Boston     1   2.0               121.0                64.0         1.0
     2      Boston     1   3.0               276.0               192.0         2.0
     3      Boston     1   4.0                73.0                67.0         1.0
     4      Boston     1   5.0               115.0               108.0         2.0
     ..        …     …     …                   …                   …           …
     182    Boston    22   8.0                82.0               145.0         0.0
     183    Boston    22   9.0               180.0               132.0         0.0
     184    Boston    22  10.0               214.0               163.0         0.0
     185    Boston    22  11.0               115.0               114.0         0.0
```

```
186    Boston    22  12.0               110.0               134.0         0.0

     Blanks  Total Votes Cast   Total   White  …  \
0      12.0             353.0  1717.0  1369.0  …
1       8.0             194.0  2071.0  1520.0  …
2      22.0             492.0  3179.0  1989.0  …
3       2.0             143.0  1884.0  1097.0  …
4       5.0             230.0  2888.0  1659.0  …
..      …               …      …       …       …
182     5.0             232.0  2065.0  1583.0  …
183     4.0             316.0  2018.0  1564.0  …
184     5.0             382.0  2442.0  1984.0  …
185     8.0             237.0  1315.0  1019.0  …
186     8.0             252.0  1646.0  1007.0  …

     Some Other Race/Ethnicity (alone)  Two or More Races/Ethnicities (alone)  \
0                                 27.0                                   27.0
1                                 53.0                                   33.0
2                                 57.0                                   89.0
3                                 59.0                                   24.0
4                                 93.0                                   40.0
..                                 …                                      …
182                                8.0                                   31.0
183                               11.0                                   52.0
184                               34.0                                   29.0
185                               12.0                                   28.0
186                               29.0                                   30.0

     Black Percentage  Native American Percentage  Asian Percentage  \
0                 3.2                         0.4               2.9
1                 2.9                         0.1               2.0
2                 8.7                         0.3               7.2
3                 3.0                         0.6               2.0
4                 2.7                         0.8               2.6
..                 …                          …                 …
182               3.0                         0.0              17.3
183               2.0                         0.0              15.8
184               3.7                         0.0              10.2
185               5.0                         0.4              11.8
186              10.0                         0.5              14.7

     Native Hawaiian/Pacific Islander Percentage  Other Race Percentage  \
0                                            0.2                   10.1
1                                            0.0                   16.9
2                                            0.0                   15.5
3                                            0.1                   29.5
4                                            0.2                   31.0
```

```
..                                                …                …
182                                             0.0              1.3
183                                             0.1              1.6
184                                             0.0              3.0
185                                             0.0              2.2
186                                             0.1              9.6

     Two or more races Percentage  Hispanic Percentage  White Percentage
0                             3.5                 29.3              79.7
1                             4.5                 51.8              73.4
2                             5.7                 41.3              62.6
3                             6.6                 63.9              58.2
4                             5.3                 57.6              57.4
..                            …                   …                 …
182                           1.8                  5.2              76.7
183                           3.0                  4.9              77.5
184                           1.8                  8.5              81.2
185                           3.1                  8.3              77.5
186                           4.0                 18.2              61.2

[187 rows x 33 columns]
```

[6]:
```python
df2['candidate'] = np.where(df2['Ayanna S. Pressley'] > df2['Michael E. Capuano'],
                            'Ayanna S. Pressley', 'Michael E. Capuano')
df2
```

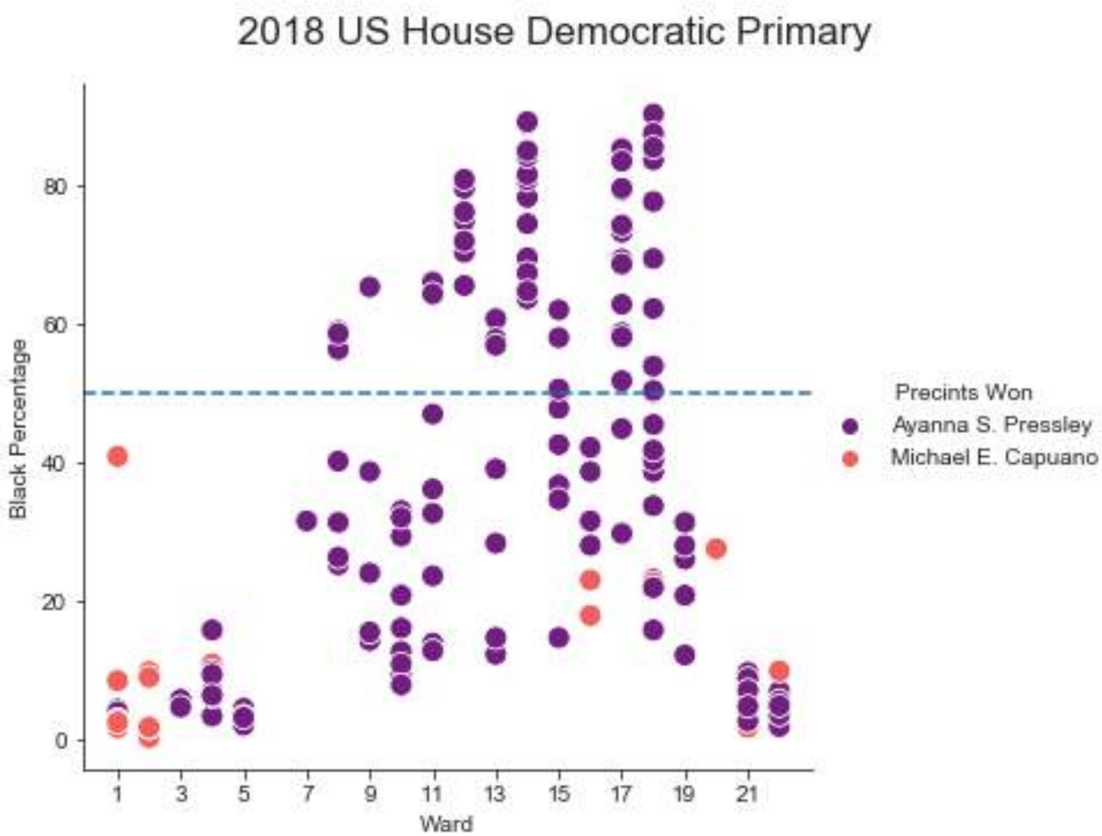
[6]:
```
    City/Town  Ward   Pct  Ayanna S. Pressley  Michael E. Capuano  All Others  \
0      Boston     1   1.0               212.0               129.0         0.0
1      Boston     1   2.0               121.0                64.0         1.0
2      Boston     1   3.0               276.0               192.0         2.0
3      Boston     1   4.0                73.0                67.0         1.0
4      Boston     1   5.0               115.0               108.0         2.0
..        …   …   …                 …                   …         …
182    Boston    22   8.0                82.0               145.0         0.0
183    Boston    22   9.0               180.0               132.0         0.0
184    Boston    22  10.0               214.0               163.0         0.0
185    Boston    22  11.0               115.0               114.0         0.0
186    Boston    22  12.0               110.0               134.0         0.0

     Blanks  Total Votes Cast   Total   White  …  \
0      12.0             353.0  1717.0  1369.0  …
1       8.0             194.0  2071.0  1520.0  …
2      22.0             492.0  3179.0  1989.0  …
3       2.0             143.0  1884.0  1097.0  …
4       5.0             230.0  2888.0  1659.0  …
..      …               …     …     …  …
```

```
182     5.0             232.0  2065.0  1583.0  …
183     4.0             316.0  2018.0  1564.0  …
184     5.0             382.0  2442.0  1984.0  …
185     8.0             237.0  1315.0  1019.0  …
186     8.0             252.0  1646.0  1007.0  …

     Two or More Races/Ethnicities (alone)  Black Percentage  \
0                                     27.0               3.2
1                                     33.0               2.9
2                                     89.0               8.7
3                                     24.0               3.0
4                                     40.0               2.7
..                                     …               …
182                                   31.0               3.0
183                                   52.0               2.0
184                                   29.0               3.7
185                                   28.0               5.0
186                                   30.0              10.0

     Native American Percentage  Asian Percentage  \
0                           0.4               2.9
1                           0.1               2.0
2                           0.3               7.2
3                           0.6               2.0
4                           0.8               2.6
..                          …               …
182                         0.0              17.3
183                         0.0              15.8
184                         0.0              10.2
185                         0.4              11.8
186                         0.5              14.7

     Native Hawaiian/Pacific Islander Percentage  Other Race Percentage  \
0                                            0.2                   10.1
1                                            0.0                   16.9
2                                            0.0                   15.5
3                                            0.1                   29.5
4                                            0.2                   31.0
..                                           …                   …
182                                          0.0                    1.3
183                                          0.1                    1.6
184                                          0.0                    3.0
185                                          0.0                    2.2
186                                          0.1                    9.6

     Two or more races Percentage  Hispanic Percentage  White Percentage  \
0                             3.5                 29.3              79.7
```

```
1                            4.5              51.8             73.4
2                            5.7              41.3             62.6
3                            6.6              63.9             58.2
4                            5.3              57.6             57.4
..                           …                …                …
182                          1.8               5.2             76.7
183                          3.0               4.9             77.5
184                          1.8               8.5             81.2
185                          3.1               8.3             77.5
186                          4.0              18.2             61.2

              candidate
0    Ayanna S. Pressley
1    Ayanna S. Pressley
2    Ayanna S. Pressley
3    Ayanna S. Pressley
4    Ayanna S. Pressley
..                  …
182  Michael E. Capuano
183  Ayanna S. Pressley
184  Ayanna S. Pressley
185  Ayanna S. Pressley
186  Michael E. Capuano

[187 rows x 34 columns]
```

[7]:
```
sns.set_style("ticks")

g2 = sns.relplot(data=df2, x="Ward", y="Black Percentage",␣
↪hue="candidate",palette="magma", kind='scatter', s=100)
g2.fig.subplots_adjust(top=.9) # adjust the Figure in g
g2.fig.suptitle('2018 US House Democratic Primary',fontsize=17)
g2.legend.set_title("Precints Won")
g2.set(xticks=np.arange(1,23,2))
g2.axes[0][0].axhline(50, ls='--')
#g.fig.set_figwidth(10)
#g.fig.set_figheight(5)
```

[7]: <matplotlib.lines.Line2D at 0x23ca1853308>



[184]:
```
g2.savefig('2018_house.png', dpi=300)
```

Mayoral2017Analysis

April 29, 2021

1 Mayoral Race 2017 Black Demographics Analysis

Using the Mayor_2017_Turnout_Race dataset

Cleaning data and analyzing Tito Jackson percentages by percent Black population per precinct

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def clean_data(filename): # removes any lines with nan values
    df = pd.read_csv(filename)
    #replacing empty entries with nan
    df.replace(r'', np.nan)
    #cleaning data
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

def find_top_precincts(df, col_name):
    BP_data = df[[col_name]].to_numpy() # creating numpy array of Black␣
 ↪percentages
    precinct_data = df[["Precinct"]].to_numpy() # creating numpy array of␣
 ↪Precinct labels
    desc_data = df.sort_values(by = col_name, ascending = False)
    desc_data = desc_data[[col_name]].to_numpy()
    labels = []
    for i in range(np.shape(BP_data)[0]):
        if float(BP_data[i]) > float(desc_data[6]):
            labels.append(int(precinct_data[i]))
        else:
            labels.append("Other")
    return labels

df = clean_data("Mayor2017PrelimTurnout.csv")
```

```
# votes for Tito Jackson by race
df['TopPrecincts'] = find_top_precincts(df, "Black Percentage")
f = sns.lmplot(x="Black Percentage", y="Jackson Percentage of votes", hue␣
↪='TopPrecincts', data=df, height = 4, aspect = 2.5);
plt.title("Jackson Percentage of Votes by Black Percentage")
f.savefig('2017_BPJackson.png', dpi=300)
```

Calculating the correlation coefficient between the Black Percentage column and the Jackson Percentage of votes column

[ ]:
```
BP_Jackson = df[["Black Percentage", "Jackson Percentage of votes"]]
correlation = BP_Jackson.corr(method='pearson')
print(correlation)
```

Correlation coefficient of 0.4292 indicates a slight positive correlation between the two variables.

Analyzing Martin Walsh percentages by percent Black population per precinct

[ ]:
```
g = sns.lmplot(x="Black Percentage", y="Walsh Percentage of votes", hue␣
↪='TopPrecincts', data=df, height = 4, aspect = 2.5);
plt.title("Walsh Percentage of Votes by Black Percentage")
g.savefig('2017_BPWalsh.png', dpi=300)
```

Calculating the correlation coefficient between the Black Percentage column and the Walsh Percentage of votes column

[ ]:
```
BP_Walsh = df[["Black Percentage", "Walsh Percentage of votes"]]
correlation2 = BP_Walsh.corr(method='pearson')
print(correlation2)
```

Correlation coefficient of -0.2527 indicates a slight negative correlation between the two variables.

2 Dorcester Mayoral Race Demographic Analysis

Dorcester wards include 15, 16, 17.

[ ]:
```
# processing data to only include Dorcester data
def Dorcester_precincts(filename):
    df = clean_data(filename)
    data_array = df.to_numpy()

    rows = []
    for i in range(np.shape(data_array)[0]):
        current_p = str(int(data_array[i, 16]))
        if len(current_p) == 4:
            if current_p[0:2] == '13' or current_p[0:2] == '14' or current_p[0:
↪2] == '15' or current_p[0:2] == '16' or current_p[0:2] == '17':
                rows.append(data_array[i].tolist())
```

```
    Dorcester_df = pd.DataFrame(np.asarray(rows), columns = df.columns)
    return Dorcester_df

df = Dorcester_precincts('Mayor2017PrelimTurnout.csv')
plt.figure(figsize=(15,8))
g = sns.barplot(x="Precinct", y="Black Percentage", data=df, palette = 'rocket')
g.set_xticklabels(g.get_xticklabels(), rotation=60);
g.set_title("Black Percentage by Precinct")
```

Jackson Percentage Plotted by Black Percentage

[ ]:
```
df['TopPrecincts'] = find_top_precincts(df, "Black Percentage")
sns.lmplot(x="Black Percentage", y="Jackson Percentage of votes", data=df,␣
 ↪height = 4, aspect = 2.5);
```

[ ]:
```
BP_Jackson = df[["Black Percentage", "Jackson Percentage of votes"]]
correlation = BP_Jackson.corr(method='pearson')
print(correlation)
```

Jackson Percentage Plotted by White Percentage

[ ]:
```
sns.lmplot(x="White Percentage", y="Jackson Percentage of votes", data = df,␣
 ↪height = 4, aspect = 2.5);
```

[ ]:
```
WP_Jackson = df[["White Percentage", "Jackson Percentage of votes"]]
correlation = WP_Jackson.corr(method='pearson')
print(correlation)
```

Walsh Percentage Plotted by Black Percentage

[ ]:
```
sns.lmplot(x="Black Percentage", y="Walsh Percentage of votes", data = df,␣
 ↪height = 4, aspect = 2.5);
```

[ ]:
```
BP_Walsh = df[["Black Percentage", "Walsh Percentage of votes"]]
correlation = BP_Walsh.corr(method='pearson')
print(correlation)
```

Walsh Percentage plotted by White Percentage

[ ]:
```
sns.lmplot(x="White Percentage", y="Walsh Percentage of votes", data = df,␣
 ↪height = 4, aspect = 2.5);
```

[ ]:
```
WP_Walsh = df[["White Percentage", "Walsh Percentage of votes"]]
correlation = WP_Walsh.corr(method='pearson')
print(correlation)
```

3 Mayoral 2017 General Analysis

Using the Mayor_2017_General_Turnout_Race dataset

[ ]:
```
df = clean_data("Mayor_2017_General_Turnout_Race.csv")
df['TopPrecincts'] = find_top_precincts(df, "Black Percentage")
f = sns.lmplot(x="Black Percentage", y="Jackson Percentage of votes", hue␣
 ↪='TopPrecincts', data=df, height = 4, aspect = 2.5);
plt.title("Jackson Percentage of Votes by Black Percentage (General)")
f.savefig('2017_BPJacksonGeneral.png', dpi=300)
```

[ ]:
```
BP_JacksonGeneral = df[["Black Percentage", "Jackson Percentage of votes"]]
correlation = BP_JacksonGeneral.corr(method='pearson')
print(correlation)
```

[ ]:
```
g = sns.lmplot(x="Black Percentage", y="Walsh Percentage of votes", hue␣
 ↪='TopPrecincts', data=df, height = 4, aspect = 2.5);
plt.title("Walsh Percentage of Votes by Black Percentage (General)")
g.savefig('2017_BPWalshGeneral.png', dpi=300)
```

[ ]:
```
BP_WalshGeneral = df[["Black Percentage", "Walsh Percentage of votes"]]
correlation2 = BP_WalshGeneral.corr(method='pearson')
print(correlation2)
```

[ ]:
```

```

PresidentialAnalysis

April 29, 2021

1 Presidential Races Analysis

1.1 2020 Trump/Biden Election

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from tabulate import tabulate

# processing data
def clean_data(filename): # removes any lines with nan values
    df = pd.read_csv(filename)
    #replacing empty entries with nan
    df.replace(r'', np.nan)
    #cleaning data
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

df =␣
 ↪clean_data("2020_President_General_Election_including_precincts_and_demographics.
 ↪csv")
f = sns.lmplot(x="Black Percentage", y="Biden Percentage", data=df, height = 4,␣
 ↪aspect = 2.5);
plt.title("2020 Biden Percentage of Votes by Black Percentage")
f.savefig('2020_BPBiden.png', dpi=300)
```

[ ]:
```
BP_Biden = df[["Black Percentage", "Biden Percentage"]]
correlation = BP_Biden.corr(method='pearson')
print(correlation)
```

[ ]:
```
from IPython.display import display, HTML
df = df.sort_values(by='Black Percentage', ascending=False)[["Ward", "Pct",␣
 ↪"Biden Percentage", "Trump Percentage", "Black Percentage"]]
display(HTML(df.to_html()))
```

```
[ ]: g = sns.lmplot(x="Black Percentage", y="Trump Percentage", data=df, height = 4,␣
      ↪aspect = 2.5);
     plt.title("2020 Trump Percentage of Votes by Black Percentage")
     g.savefig('2020_BPTrump.png', dpi=300)
```

```
[ ]: BP_Trump = df[["Black Percentage", "Trump Percentage"]]
     correlation2 = BP_Trump.corr(method='pearson')
     print(correlation2)
```

1.2 2016 Trump/Clinton Election

```
[ ]: df2 =␣
      ↪clean_data("2016_President_General_Election_including_precincts_and_demographics.
      ↪csv")
     m = sns.lmplot(x="Black Percentage", y="Clinton Percentage", data=df2, height =␣
      ↪4, aspect = 2.5);
     plt.title("2016 Clinton Percentage of Votes by Black Percentage")
     m.savefig('2016_BPClinton.png', dpi=300)
```

```
[ ]: BP_Clinton = df2[["Black Percentage", "Clinton Percentage"]]
     correlation3 = BP_Clinton.corr(method='pearson')
     print(correlation3)
```

```
[ ]: n = sns.lmplot(x="Black Percentage", y="Trump Percentage", data=df2, height =␣
      ↪4, aspect = 2.5);
     plt.title("2016 Trump Percentage of Votes by Black Percentage")
     n.savefig('2016_BPTrump.png', dpi=300)
```

```
[ ]: BP_Trump2016 = df2[["Black Percentage", "Trump Percentage"]]
     correlation4 = BP_Trump2016.corr(method='pearson')
     print(correlation4)
```

```
[ ]:
```

A General Analysis of Black Percentage and Vote Counts

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def clean_data(filename): # removes any lines with nan values
    df = pd.read_csv(filename)
    #replacing empty entries with nan
    df.replace(r'', np.nan)
    #cleaning data
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

# find the precinct winner
def precinct_winner_2011(df,c1,c2,c3,c4,c5):
    outcome = []
    c1_data = df[[c1]].to_numpy()
    c2_data = df[[c2]].to_numpy()
    c3_data = df[[c3]].to_numpy()
    c4_data = df[[c4]].to_numpy()
    c5_data = df[[c5]].to_numpy()
    for i in range(len(c1_data)):
        ltemp = [c1_data[i],c2_data[i],c3_data[i],c4_data[i],c5_data[i
]]
        index = ltemp.index(max(ltemp))
        if index == 0:
            outcome.append(c1)
        elif index == 1:
            outcome.append(c2)
        elif index == 2:
            outcome.append(c3)
        elif index == 3:
            outcome.append(c4)
        else:
            outcome.append(c5)
    return outcome

# find the precinct winner
def precinct_winner_2013(df,c1,c2,c3,c4,c5,c6,c7,c8):
    outcome = []
```

```
    c1_data = df[[c1]].to_numpy()
    c2_data = df[[c2]].to_numpy()
    c3_data = df[[c3]].to_numpy()
    c4_data = df[[c4]].to_numpy()
    c5_data = df[[c5]].to_numpy()
    c6_data = df[[c6]].to_numpy()
    c7_data = df[[c7]].to_numpy()
    c8_data = df[[c8]].to_numpy()
    for i in range(len(c1_data)):
        ltemp = [c1_data[i],c2_data[i],c3_data[i],c4_data[i],c5_data[i
],c6_data[i],c7_data[i],c8_data[i]]
        index = ltemp.index(max(ltemp))
        if index == 0:
            outcome.append(c1)
        elif index == 1:
            outcome.append(c2)
        elif index == 2:
            outcome.append(c3)
        elif index == 3:
            outcome.append(c4)
        elif index == 4:
            outcome.append(c5)
        elif index == 5:
            outcome.append(c6)
        elif index == 6:
            outcome.append(c7)
        else:
            outcome.append(c8)
    return outcome
```

Result Analysis on 2011 Dataset

In [ ]:
```
df_2011 = clean_data("2011_CityCouncil_Results_Race_Turnout.csv")
```

QUESTION1: The correlation of black percentage and candidate's vote count

plotting Black Percentage on the x-axis and each candidate on the y-axis for each precinct

In [ ]:
```
# votes for AYANNA PRESSLEY and Black Percentage
sns.lmplot(x="Black Percentage", y="AYANNA PRESSLEY", data=df_2011, he
ight = 4, aspect = 2.5);
```

In [ ]:
```
# votes for FELIX G ARROYO and Black Percentage
sns.lmplot(x="Black Percentage", y="FELIX G ARROYO", data=df_2011, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for JOHN R CONNOLLY and Black Percentage
sns.lmplot(x="Black Percentage", y="JOHN R CONNOLLY", data=df_2011, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for MICHAEL F FLAHERTY and Black Percentage
sns.lmplot(x="Black Percentage", y="MICHAEL F FLAHERTY", data=df_2011, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for STEPHEN J MURPHY and Black Percentage
sns.lmplot(x="Black Percentage", y="STEPHEN J MURPHY", data=df_2011, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for SEAN H RYAN and Black Percentage
sns.lmplot(x="Black Percentage", y="SEAN H RYAN", data=df_2011, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for WILL DORCENA and Black Percentage
sns.lmplot(x="Black Percentage", y="WILL DORCENA", data=df_2011, height = 4, aspect = 2.5);
```

Analysis:

From the plot, we can see that Will Dorcena has the strongest positive correlation between black percentage and vote counts. Ayanna Pressley has a slight positive correlation. JOHN R CONNOLLY, MICHAEL F FLAHERTY and SEAN H RYAN all have negative correlation. According to Wikipedia, Ayanna Pressley is first black woman elected to the Boston City Council. Will Dorcena is a black man and the rest of candidates are white men. Thus, the finding is not surprising and it comfirms the hypothesis that black voters are more likely to vote black candidates.

Question2 The Correlation of Black Percentage and Winner in Each Ward

plotting ward on the x-axis and black percentage on the y-axis and hue is the winner

In [ ]:
```
df_2011['Candidates'] = precinct_winner_2011(df_2011, "AYANNA PRESSLEY
","FELIX G ARROYO","JOHN R CONNOLLY", "MICHAEL F FLAHERTY","STEPHEN J
MURPHY")
sns.set_style("ticks")

g = sns.relplot(data=df_2011, x="Precinct", y="Black Percentage", hue=
"Candidates",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2011 City Council Result',fontsize=17)
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

Analysis

From the graph we can see that Ayanna Pressley is leading in precincts that has large black percentage.

Result Analysis on 2013 Dataset

In [ ]:
```
df_2013 = clean_data("2013_CityCouncil_Race_Turnout_Results.csv")
```

QUESTION1: The correlation of black percentage and candidate's vote count

plotting Black Percentage on the x-axis and each candidate on the y-axis for each precinct

In [ ]:
```
# votes for AYANNA S PRESSLEY and Black Percentage
sns.lmplot(x="Black Percentage", y="AYANNA S PRESSLEY", data=df_2013,
height = 4, aspect = 2.5);

# votes for MARTIN J KEOGH and Black Percentage
sns.lmplot(x="Black Percentage", y="MARTIN J KEOGH", data=df_2013, hei
ght = 4, aspect = 2.5);

# votes for JACK F KELLY, III and Black Percentage
sns.lmplot(x="Black Percentage", y="JACK F KELLY, III", data=df_2013,
height = 4, aspect = 2.5);

# votes for ANNISSA ESSAIBI GEORGE and Black Percentage
sns.lmplot(x="Black Percentage", y="ANNISSA ESSAIBI GEORGE", data=df_2
013, height = 4, aspect = 2.5);
```

In [ ]:
```
# votes for MICHAEL F FLAHERTY and Black Percentage
sns.lmplot(x="Black Percentage", y="MICHAEL F FLAHERTY", data=df_2013,
height = 4, aspect = 2.5);

# votes for MICHELLE WU and Black Percentage
sns.lmplot(x="Black Percentage", y="MICHELLE WU", data=df_2013, height
= 4, aspect = 2.5);

# votes for STEPHEN J MURPHY and Black Percentage
sns.lmplot(x="Black Percentage", y="STEPHEN J MURPHY", data=df_2013, h
eight = 4, aspect = 2.5);

# votes for JEFFREY MICHAEL ROSS and Black Percentage
sns.lmplot(x="Black Percentage", y="JEFFREY MICHAEL ROSS", data=df_201
3, height = 4, aspect = 2.5);
```

Question2 The Correlation of Black Percentage and Winner in Each Ward

plotting ward on the x-axis and black percentage on the y-axis and hue is the winner

In [ ]:
```
df_2013['Candidates'] = precinct_winner_2013(df_2013,"AYANNA S PRESSLE
Y","MARTIN J KEOGH","JACK F KELLY, III","ANNISSA ESSAIBI GEORGE","MICH
AEL F FLAHERTY","MICHELLE WU","STEPHEN J MURPHY","JEFFREY MICHAEL ROSS
")
sns.set_style("ticks")

g = sns.relplot(data=df_2013, x="Precinct", y="Black Percentage", hue=
"Candidates",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2013 City Council Result',fontsize=17)
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

Result Analysis on 2015 Dataset

In [ ]:
```
df_2015 = clean_data("2015_city_council.csv")
```

In [ ]:
```
df_2015['Candidates'] = precinct_winner_2011(df_2015,"AYANNA S PRESSLE
Y","ANNISSA ESSAIBI GEORGE","MICHAEL F FLAHERTY","MICHELLE WU","STEPHE
N J MURPHY")
sns.set_style("ticks")

g = sns.relplot(data=df_2015, x="Precinct", y="Black Percentage", hue=
"Candidates",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2015 City Council Result',fontsize=17)
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

In [ ]:
```
t=[]
for i in df_2015["Precinct"]:
    i = int(i)
    t.append(str(i).zfill(4))
df_2015['Precinct_formatted'] = t
print(df_2015["Precinct_formatted"])
df_2015.to_csv("data_2015.csv",index=False)
```

Result Analysis on 2017 Dataset

In [ ]:
```
df_2017 = clean_data("2017_CityCouncil_AtLarge_Turnout_Race.csv")
```

In [ ]:
```
df_2017['Winner'] = precinct_winner_2013(df_2017,"AYANNA S PRESSLEY","
ANNISSA E GEORGE","MICHAEL F FLAHERTY","MICHELLE WU","PAT PAYASO","DOM
INGOS DAROSA","ALTHEA GARRISON","WILLIAM A KING")
sns.set_style("ticks")

g = sns.relplot(data=df_2017, x="Precinct", y="Black Percentage", hue=
"Winner",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2017 City Council Result',fontsize=17)
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

In [ ]:
```
print(df_2017.columns)
t=[]
for i in df_2017["Precinct"]:
    i = int(i)
    t.append(str(i).zfill(4))
df_2017['Precinct_formatted'] = t
print(df_2017["Precinct_formatted"])
df_2017.to_csv("data_2017.csv",index=False)
```

Result Analysis on 2019 Dataset

In [ ]:
```
df_2019 = clean_data("2019 CC Race Turnout.csv")
```

In [ ]:
```
sns.set_style("ticks")

g = sns.relplot(data=df_2019, x="Ward_Precinct_Code", y="Black Percentage", hue="Winner",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('2019 City Council Result',fontsize=17)
g.set(xticks=np.arange(1,23,2))
g.axes[0][0].axhline(50, ls='--')
```

In [ ]:
```
t=[]
for i in df_2019["Precinct_y"]:
    i = int(i)
    t.append(str(i).zfill(4))
df_2019['Precinct_formatted'] = t
print(df_2019["Precinct_formatted"])
df_2019.to_csv("data_2019.csv",index=False)
```

In [ ]:

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Import & Clean Data ¶

In [3]:

```
raw_df = pd.read_csv("DA_Race_Turnout.csv")
```

In [4]:

```
raw_df
```

Out[4]:

	Unnamed: 0	Unnamed: 0.1	WP	CARVALHO	CHAMPION	HENNING	McAULIFFE	ROLLINS
0	0	0	0101	49	17	68	45	138
1	1	1	0102	46	12	27	18	66
2	2	2	0103	95	53	54	44	174
3	3	3	0104	34	13	21	11	45
4	4	4	0105	40	18	50	23	67
...	...	...	...	...	...	...	...	...
250	250	250	1819	81	146	142	40	182
251	251	251	1821	169	89	41	8	147
252	252	252	1823	66	56	35	38	93
253	253	253	0502A	36	24	29	24	55
254	254	254	2213	41	8	49	30	62

255 rows × 48 columns

Drop columns that are useless/unknown/unidentified

In [5]:

```
drop_columns = ['Unnamed: 0', 'Unnamed: 0.1', 'Unnamed: 0.1.1'] #unidentified/unknow
drop_columns += ['Ward', 'Precinct'] #WP == WARD and Precinct == WP
drop_columns += ['Total.1'] #Total == Total.1; infer that total means the # of peopl

drop_columns.append('TOTAL') #VOTES_CAST == TOTAL; therefore VOTES_CAST --> "TOTAL V
rename_columns={'VOTE CAST': 'VOTE_CAST', 'BALLOTS CAST': 'BALLOTS_CAST'} #rename fo
```

In [6]:

```
def clean_columns(df, drop_columns=[], rename_columns={}):
    new_df = df.drop(drop_columns, axis=1).dropna()
    return new_df.rename(columns=rename_columns)
```

In [7]:

```
clean_df = clean_columns(raw_df, drop_columns, rename_columns)
```

In [8]:

```
clean_df.head()
```

Out[8]:

	WP	CARVALHO	CHAMPION	HENNING	McAULIFFE	ROLLINS	OTHERS	VOTES_CAST	BL
0	0101	49	17	68	45	138	1	318	
1	0102	46	12	27	18	66	0	169	
2	0103	95	53	54	44	174	2	422	
3	0104	34	13	21	11	45	2	126	
4	0105	40	18	50	23	67	0	198	

5 rows × 41 columns

Analysis

of Precinct Won of each candidate

In [9]:

```
# num of wins of each participant
winners_df = clean_df[['WP', 'CARVALHO', 'CHAMPION', 'HENNING', 'McAULIFFE', 'ROLLIN
```

In [10]:

```
winners_df.head()
```

Out[10]:

	WP	CARVALHO	CHAMPION	HENNING	McAULIFFE	ROLLINS	OTHERS
0	0101	49	17	68	45	138	1
1	0102	46	12	27	18	66	0
2	0103	95	53	54	44	174	2
3	0104	34	13	21	11	45	2
4	0105	40	18	50	23	67	0

In [11]:

```
winners_df['WON'] = winners_df.drop('WP', axis=1).idxmax(axis=1)
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

In [12]:

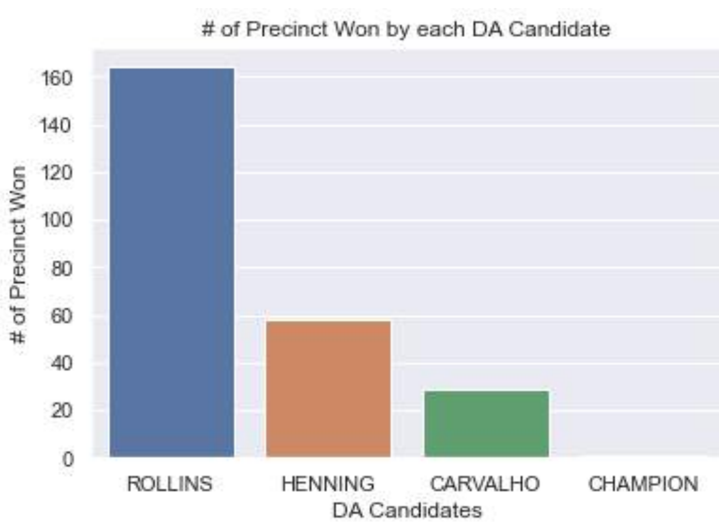
```
winners_df.head()
```

Out[12]:

	WP	CARVALHO	CHAMPION	HENNING	McAULIFFE	ROLLINS	OTHERS	WON
0	0101	49	17	68	45	138	1	ROLLINS
1	0102	46	12	27	18	66	0	ROLLINS
2	0103	95	53	54	44	174	2	ROLLINS
3	0104	34	13	21	11	45	2	ROLLINS
4	0105	40	18	50	23	67	0	ROLLINS

In [13]:

```
sns.set_theme(style="darkgrid")
ax = sns.countplot(x="WON", data=winners_df)
plt.ylabel('# of Precinct Won')
plt.xlabel('DA Candidates')
plt.title('# of Precinct Won by each DA Candidate')
plt.show()
```



We could see from this countplot that Rollins won the DA Race by a significant amount

Analyze by Race

Percentage of Black in each Precinct vs Candidate that won in that precinct

In [14]:

```
#get the columns we want to analyze this
black_percentage_df = clean_df[['WP','Black Percentage']]
black_percentage_df = pd.merge(black_percentage_df, winners_df[['WP', 'WON']])
```

In [15]:

```
black_percentage_df.head()
```

Out[15]:

	WP	Black Percentage	WON
0	0101	3.203262	ROLLINS
1	0102	2.945437	ROLLINS
2	0103	8.744888	ROLLINS
3	0104	3.025478	ROLLINS
4	0105	2.666205	ROLLINS

In [16]:

```
sns.set_theme(style="darkgrid")
fig, ax = plt.subplots(figsize=(20, 10))
sns.scatterplot(ax=ax, data=black_percentage_df, x='WP', y='Black Percentage', hue='
ax.legend(title='DA Candidates')
ax.axhline(50, ls='--')
plt.setp(ax.get_xticklabels(), visible=False)
plt.ylabel('Percentage of Black in a Precinct')
plt.xlabel('Precinct')
plt.title('% of Black in each Precinct vs Candidate that won in each Precinct')
plt.show()
```

Percentage of White in each Precinct vs Candidate that won in that precinct

In [17]:

```
#get the columns we want to analyze this
white_percentage_df = clean_df[['WP','White Percentage']]
white_percentage_df = pd.merge(white_percentage_df, winners_df[['WP', 'WON']])
```

In [18]:

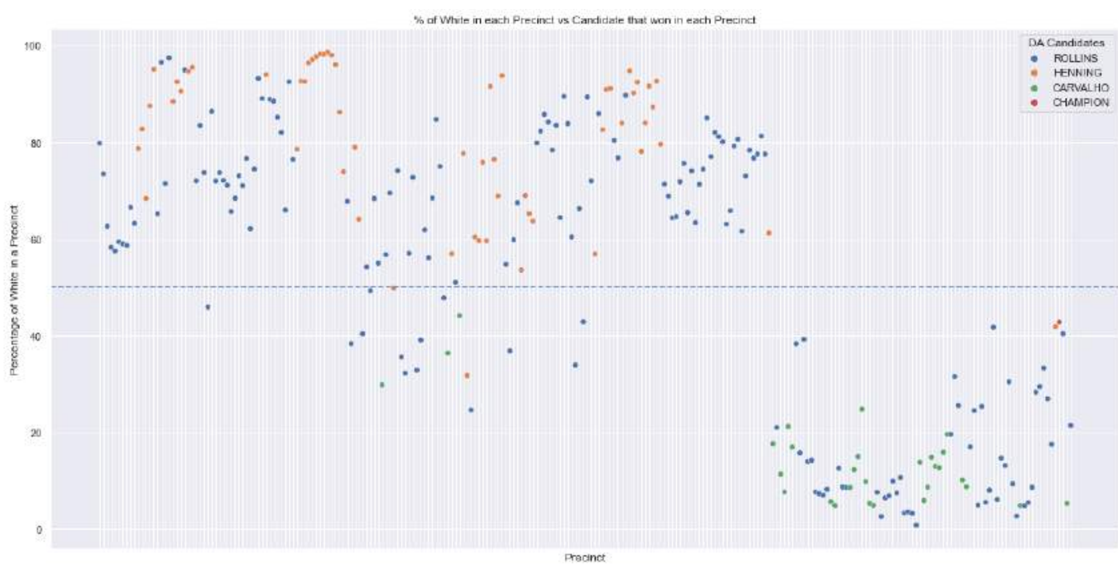
```
white_percentage_df.head()
```

Out[18]:

	WP	White Percentage	WON
0	0101	79.732091	ROLLINS
1	0102	73.394495	ROLLINS
2	0103	62.566845	ROLLINS
3	0104	58.227176	ROLLINS
4	0105	57.444598	ROLLINS

In [19]:

```
sns.set_theme(style="darkgrid")
fig, ax = plt.subplots(figsize=(20, 10))
sns.scatterplot(ax=ax, data=white_percentage_df, x='WP', y='White Percentage', hue='
ax.legend(title='DA Candidates')
ax.axhline(50, ls='--')
plt.setp(ax.get_xticklabels(), visible=False)
plt.ylabel('Percentage of White in a Precinct')
plt.xlabel('Precinct')
plt.title('% of White in each Precinct vs Candidate that won in each Precinct')
plt.show()
```



Written analysis

From the two graphs above, we could see that all of the precincts with black population over 50% support Rollings & Carvalho, while Henning, although second in the order of votes, mostly receive his votes from precincts with dense white population. However, these premilinary analysis doesn't provide enough evidences to support the hypothesis that black voters are more likely to vote for black candidates. This is because we don't have enough data of the voters to get an accurate measure. For example, if we were to set a horizontal line at 70-80% in the first graph, we could see that the precincts with dense black population mostly vote for Rollings & Carvalho, which would likely signifies that most black in those specific precincts support black candidates; however, that is very sample and therefore is not enough to support the hypothesis. Furthermore, this could not be say the same for any precinct with black percentage below 70-80%, since it is possible that the precincts with lower than 70% black population could have won because of other race other than black. Furthermore, we are using the race percentage of the population, not the race percantge of voters.

Visualizing data

Clean & Preprocess shapefile and csv

In [20]:

```
import geopandas as gpd
import shapely
from geopandas.tools import sjoin
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/geopandas/_compat.py:110: UserWarning: The Shapely GEOS versio
n (3.8.0-CAPI-1.13.1 ) is incompatible with the GEOS version PyGEOS wa
s compiled with (3.9.0-CAPI-1.16.2). Conversions between both will be
slow.
  shapely_geos_version, geos_capi_version_string
```

In [21]:

```
import requests
import json
import os
from dotenv import load_dotenv
load_dotenv()
```

Out[21]:

```
True
```

In [22]:

```
#Google API
def google_geocode_api(street_address: str, city: str) -> (float, float):
    api_key = os.getenv('GOOGLE_API_KEY')
    google_query = "https://maps.googleapis.com/maps/api/geocode/json?address="+stre
    Google_request = requests.get(google_query)
    Google_JSON = Google_request.json()

    Google_coordinates = Google_JSON['results'][0]['geometry']['location']
    lat = Google_coordinates['lat']
    lon = Google_coordinates['lng']

    return (lat, lon)
```

In [23]:

```
#MA ward precincts
#https://docs.digital.mass.gov/dataset/massgis-data-wards-and-precincts
#download and unzip it inside "data" folder

WP_gdf = gpd.read_file('data/WARDSPRECINCTS_POLY.shp')
```

In [24]:

```
BOSTON_geodf = WP_gdf[(WP_gdf['TOWN'] == 'BOSTON')]
BOSTON_geodf.head(3)
```

Out[24]:

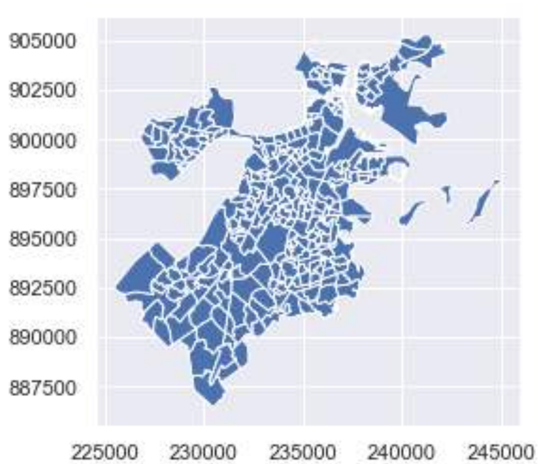
	WP_NAME	WARD	PRECINCT	DISTRICT	POP_2010	TOWN	TOWN_ID	AREA_SQMI	A
1773	Boston City Ward 19 Precinct 5	19	5	19-5	1783	BOSTON	35	0.101157	
1774	Boston City Ward 19 Precinct 3	19	3	19-3	1469	BOSTON	35	0.303572	
1775	Boston City Ward 14 Precinct 2	14	2	14-2	3798	BOSTON	35	0.186428	

In [25]:

```
BOSTON_geodf.plot()
```

Out[25]:

<AxesSubplot:>



In [26]:

```
#split WP into WARD and PRECINCT columns
clean_df.head()
```

Out[26]:

	WP	CARVALHO	CHAMPION	HENNING	McAULIFFE	ROLLINS	OTHERS	VOTES_CAST	BL
0	0101	49	17	68	45	138	1	318	
1	0102	46	12	27	18	66	0	169	
2	0103	95	53	54	44	174	2	422	
3	0104	34	13	21	11	45	2	126	
4	0105	40	18	50	23	67	0	198	

5 rows × 41 columns

In [27]:

```
def replace_WP(row):
    w = row[:2]
    p = row[2:]
    if(w[0] == '0'):
        w = w[1]
    if(p[0] == '0'):
        p = p[1]
    return w+'-'+p
```

In [28]:

```
def combine_df_geodf(df, geodf):
    new_df = df.copy()
    new_df['WP'] = df.apply(lambda row: replace_WP(str(row['WP'])), axis=1)
    return pd.merge(geodf, new_df,  how='left', left_on=['DISTRICT'], right_on = ['W
```

In [29]:

```
WP_merged_geodf = combine_df_geodf(clean_df, BOSTON_geodf)
```

In [30]:

```
WP_merged_geodf.shape
```

Out[30]:

```
(255, 54)
```

In [31]:

```
WP_merged_geodf.head()
```

Out[31]:

	WP_NAME	WARD	PRECINCT	DISTRICT	POP_2010	TOWN	TOWN_ID	AREA_SQMI	ARE
0	Boston City Ward 19 Precinct 5	19	5	19-5	1783	BOSTON	35	0.101157	6
1	Boston City Ward 19 Precinct 3	19	3	19-3	1469	BOSTON	35	0.303572	19
2	Boston City Ward 14 Precinct 2	14	2	14-2	3798	BOSTON	35	0.186428	11
3	Boston City Ward 14 Precinct 7	14	7	14-7	2002	BOSTON	35	0.085961	5
4	Boston City Ward 14 Precinct 9	14	9	14-9	2036	BOSTON	35	0.086615	5

5 rows × 54 columns

In [33]:

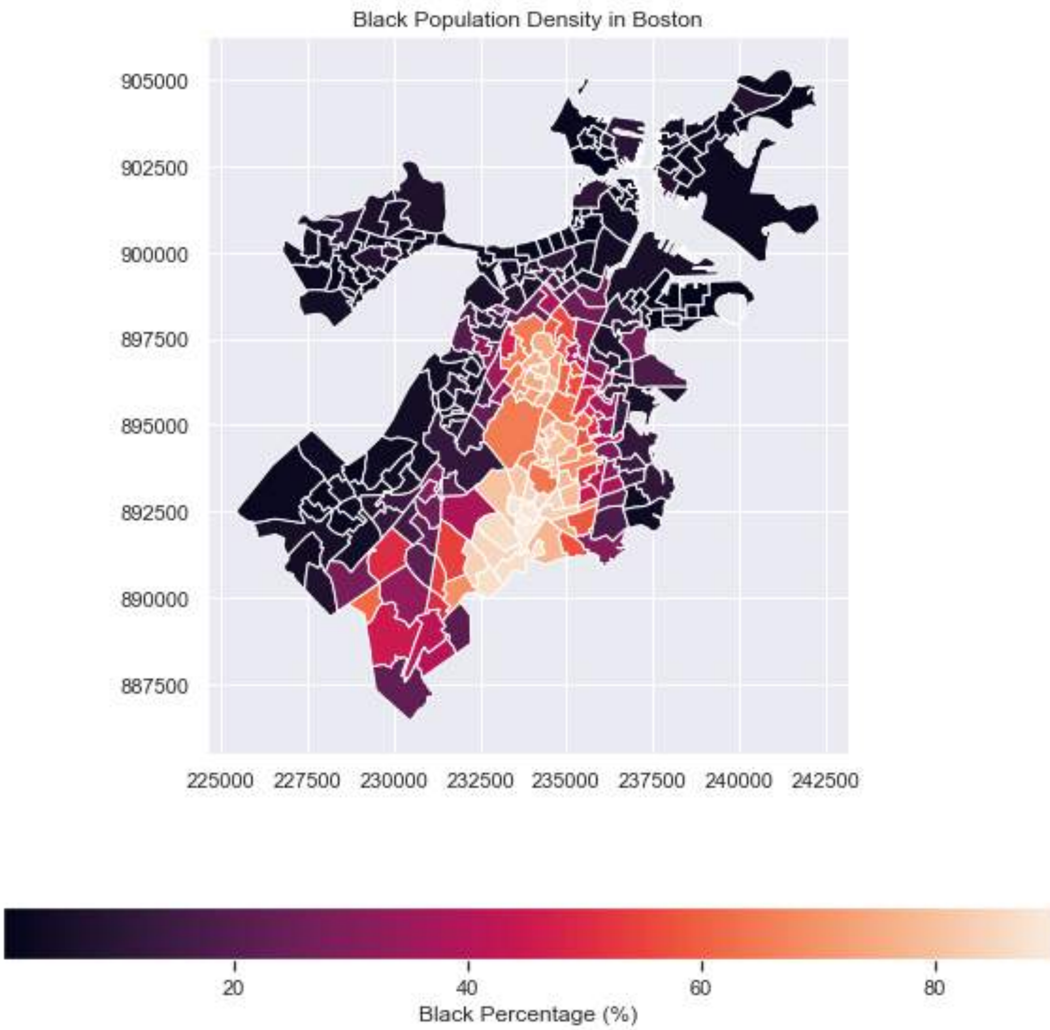
```
WP_merged_geodf.to_file("data/WP_merged_geodf.shp")
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/ipykernel_launcher.py:1: UserWarning: Column names longer than
10 characters will be truncated when saved to ESRI Shapefile.
  """Entry point for launching an IPython kernel.
```

Visualize Race Density Population in Boston

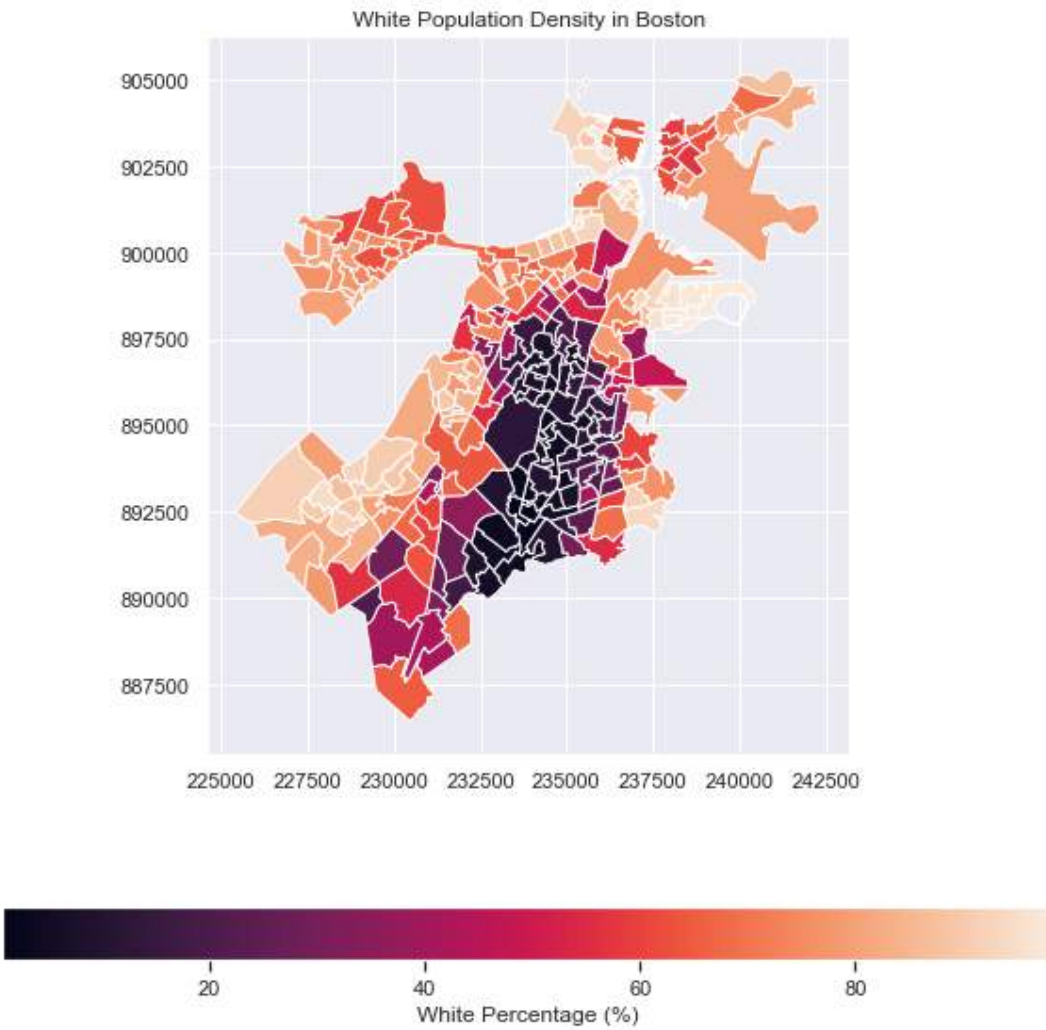
In [83]:

```
fig, ax_black = plt.subplots(figsize=(10,10))
WP_merged_geodf.plot(column='Black Percentage', ax=ax_black, legend=True, legend_kwc
plt.title('Black Population Density in Boston')
plt.show()
```



In [84]:

```
fig, ax = plt.subplots(figsize=(10,10))
WP_merged_geodf.plot(column='White Percentage', ax=ax, legend=True, legend_kwds={'la
plt.title('White Population Density in Boston')
plt.show()
```



In [36]:

```
WINNER_geodf = combine_df_geodf(winners_df, BOSTON_geodf)
WINNER_geodf.head(3)
```

Out[36]:

	WP_NAME	WARD	PRECINCT	DISTRICT	POP_2010	TOWN	TOWN_ID	AREA_SQMI	ARE/
0	Boston City Ward 19 Precinct 5	19	5	19-5	1783	BOSTON	35	0.101157	(
1	Boston City Ward 19 Precinct 3	19	3	19-3	1469	BOSTON	35	0.303572	1(
2	Boston City Ward 14 Precinct 2	14	2	14-2	3798	BOSTON	35	0.186428	1'

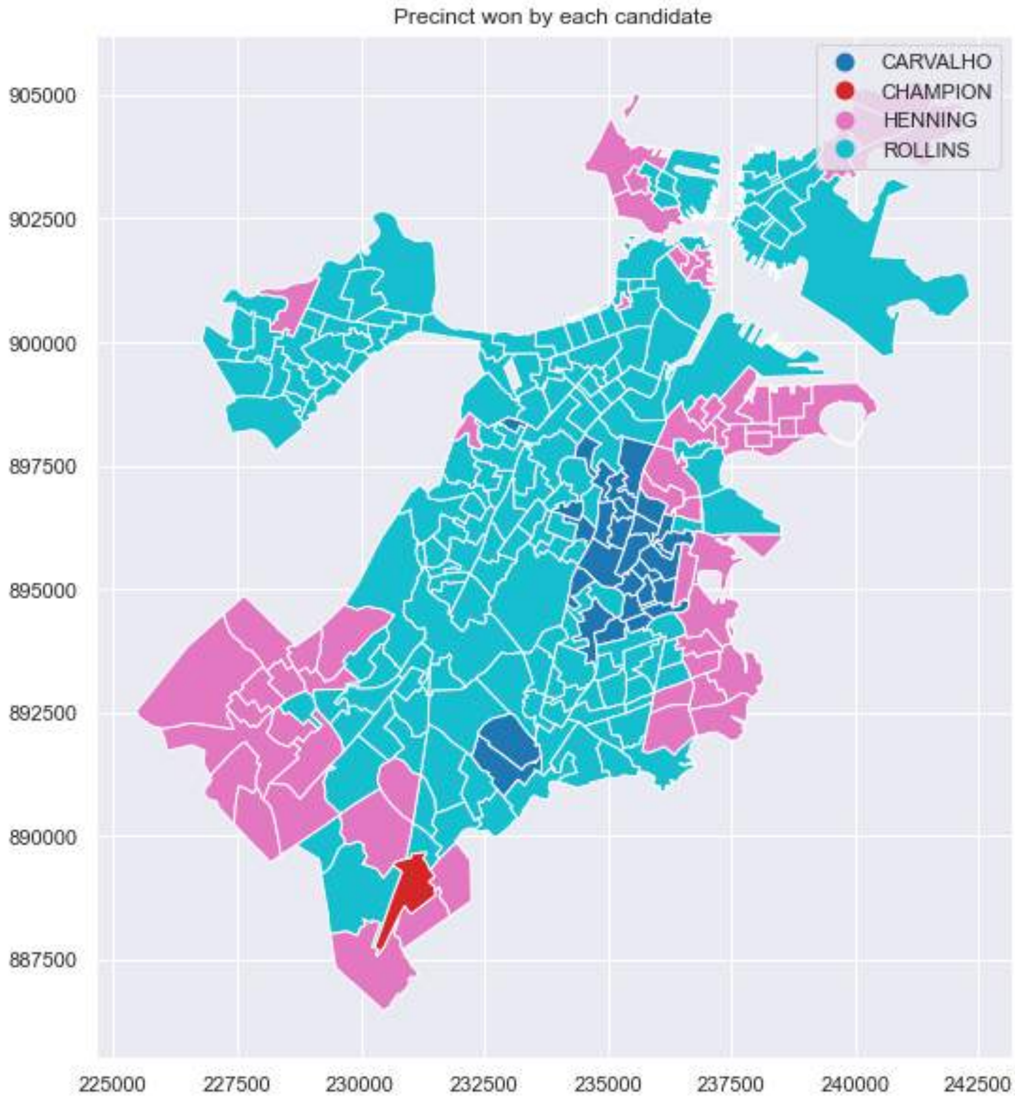
3 rows × 21 columns

In [37]:

```
WINNER_geodf.to_file('data/WINNER_geodf.shp')
```

In [39]:

```
fig, ax_winner = plt.subplots(figsize=(10,10))
WINNER_geodf.plot(column='WON', ax=ax_winner, legend=True)
plt.title('Precinct won by each candidate')
plt.show()
```



Visualize Race Density Population in Boston with Ward Labeled

In [69]:

```
dissolve_WARD = WINNER_geodf.copy().dissolve(by='WARD')
```

In [70]:

```
dissolve_WARD.shape
```

Out[70]:

```
(22, 20)
```

In [71]:

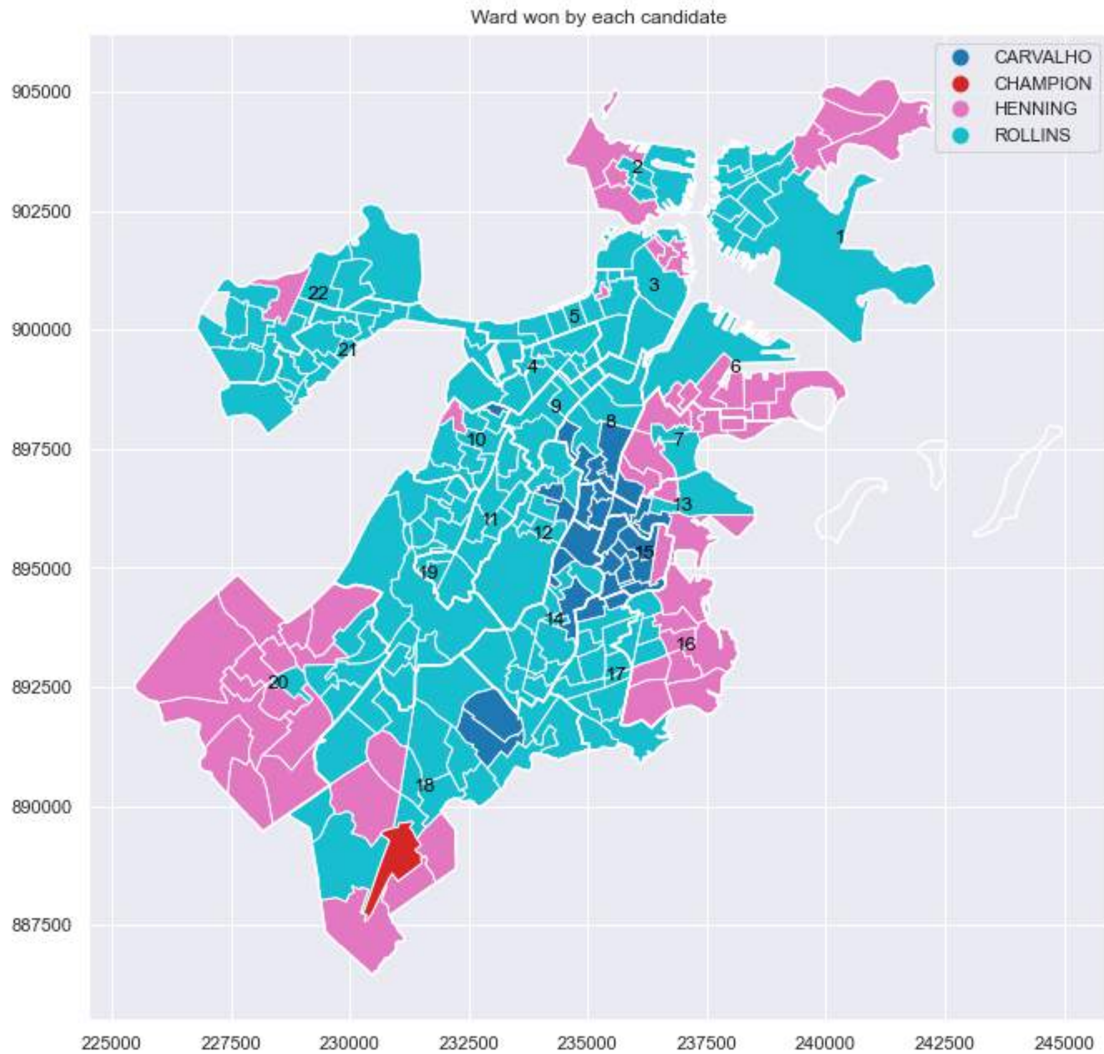
```
dissolve_WARD.head()
```

Out[71]:

	geometry	WP_NAME	PRECINCT	DISTRICT	POP_2010	TOWN	TOWN_ID	ARE
WARD								
1	MULTIPOLYGON (((237516.043 902204.085, 237499....	Boston City Ward 1 Precinct 8	8	1-8	3984	BOSTON	35	C
10	POLYGON ((232557.538 896625.021, 232545.346 89...	Boston City Ward 10 Precinct 4	4	10-4	3114	BOSTON	35	C
11	POLYGON ((231356.623 895019.240, 231358.330 89...	Boston City Ward 11 Precinct 10	10	11-10	1748	BOSTON	35	C
12	POLYGON ((233721.602 896678.361, 233681.429 89...	Boston City Ward 12 Precinct 7	7	12-7	4293	BOSTON	35	C
13	POLYGON ((234885.788 895910.172, 234692.606 89...	Boston City Ward 13 Precinct 10	10	13-10	2094	BOSTON	35	C

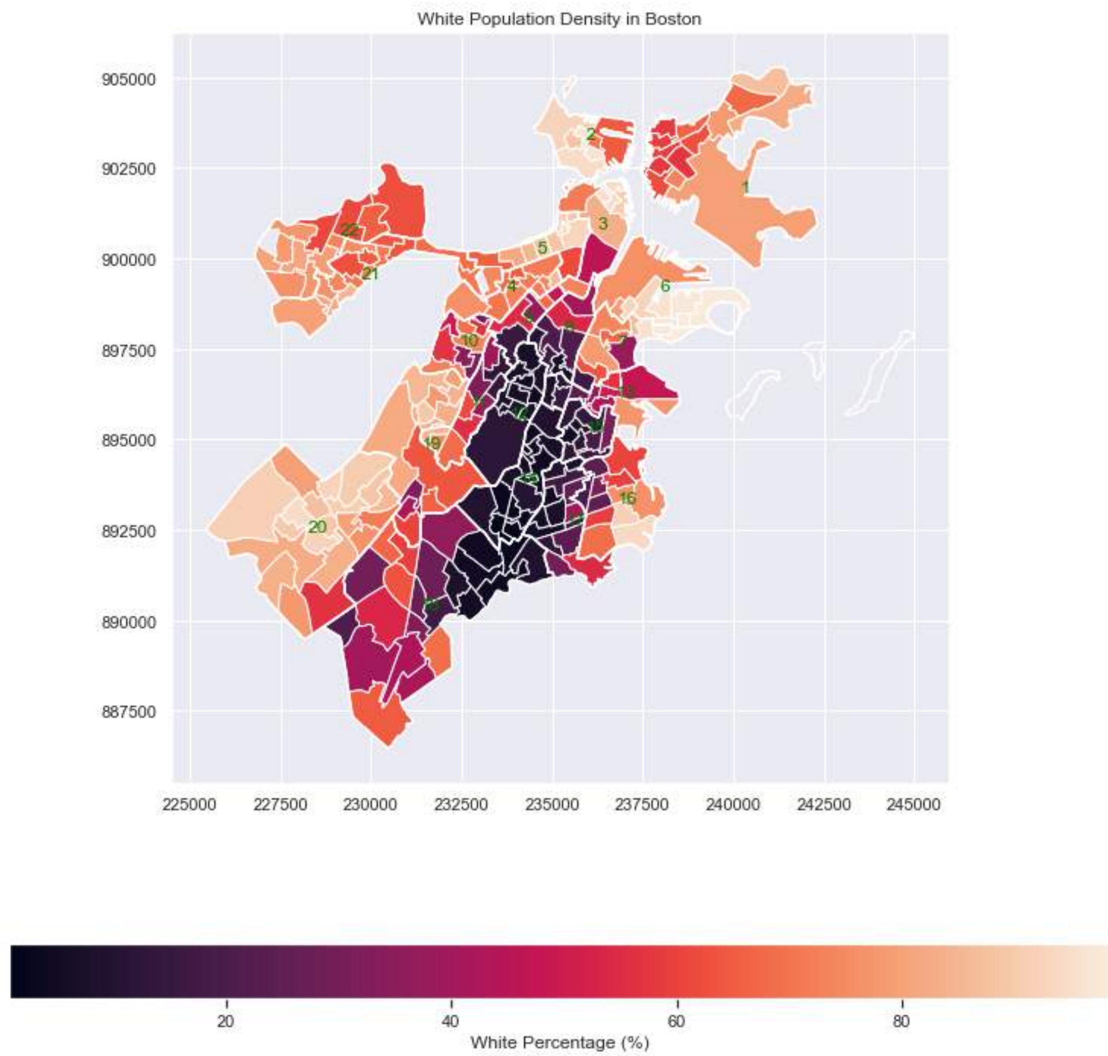
In [72]:

```
fig, ax_winner = plt.subplots(figsize=(11,11))
ax_boundary = dissolve_WARD.boundary.plot(ax=ax_winner, color='white')
WINNER_geodf.plot(column='WON', ax=ax_boundary, legend=True)
for district, (i, geo) in zip(dissolve_WARD.index ,dissolve_WARD.centroid.iteritems(
    ax_winner.annotate(district, xy=[geo.x, geo.y], color="black")
plt.title('Ward won by each candidate')
plt.show()
```



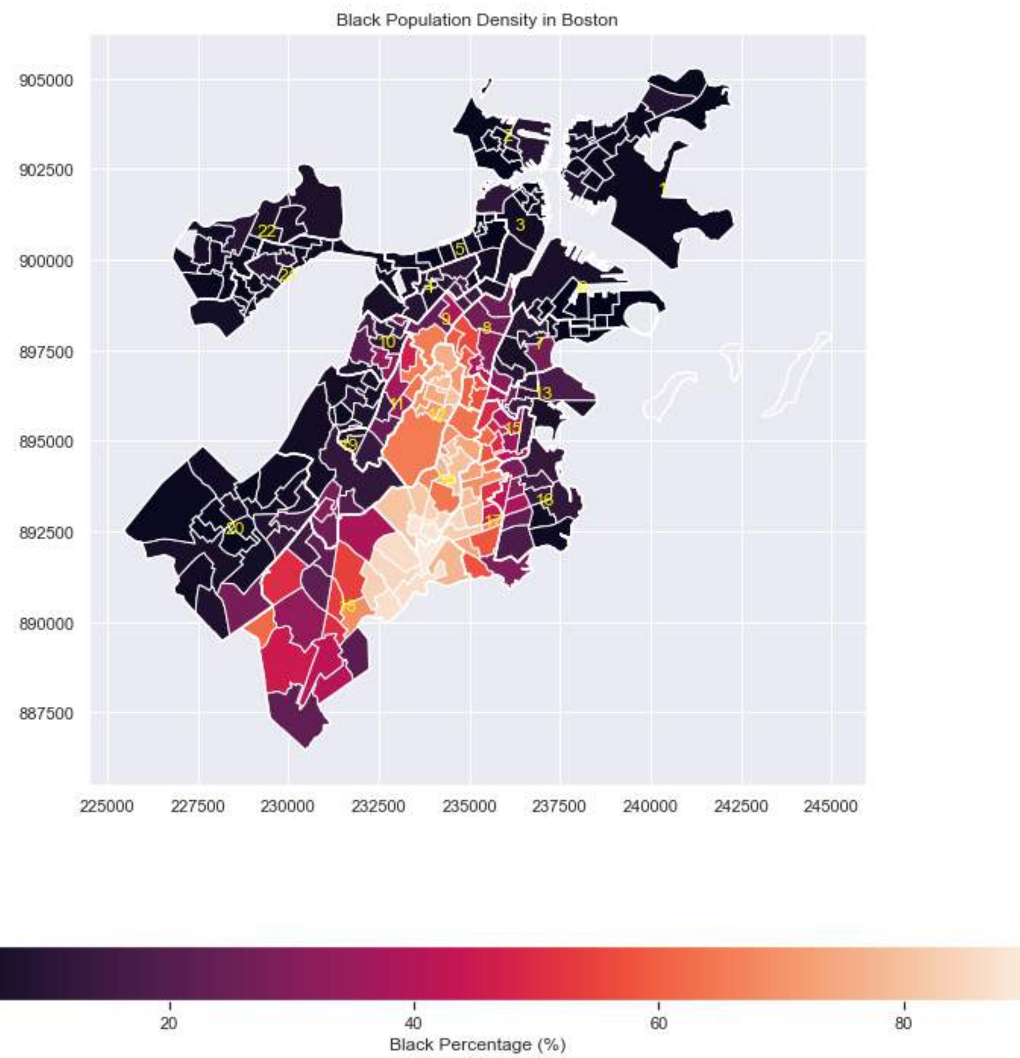
In [51]:

```
fig, ax_black = plt.subplots(figsize=(13,13))
ax_boundary = dissolve_WARD.boundary.plot(ax=ax_black, color='white')
WP_merged_geodf.plot(column='White Percentage', ax=ax_boundary, legend=True, legend_
for district, (i, geo) in zip(dissolve_WARD.index ,dissolve_WARD.centroid.iteritems(
    ax_boundary.annotate(district, xy=[geo.x, geo.y], color="green")
plt.title('White Population Density in Boston')
plt.show()
```



In [68]:

```
fig, ax_black = plt.subplots(figsize=(13,13))
ax_boundary = dissolve_WARD.boundary.plot(ax=ax_black, color='white')
WP_merged_geodf.plot(column='Black Percentage', ax=ax_boundary, legend=True, cmap='r
for district, (i, geo) in zip(dissolve_WARD.index ,dissolve_WARD.centroid.iteritems(
    ax_boundary.annotate(district, xy=[geo.x, geo.y], color="yellow")
plt.title('Black Population Density in Boston')
plt.show()
```



In [ ]:

VoterTurnoutCC

April 29, 2021

1 City Council Voter Turnout Analysis

We examine the change in voter turnout over the various CC election years. Here, we are answering an essential question: How has voter turnout by precinct changed across election year?

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

#loading datasets into dataframes
cc2011 = pd.read_csv("2011_CityCouncil_Results_Race_Turnout.csv")
cc2013 = pd.read_csv("2013_CityCouncil_Race_Turnout_Results.csv")
cc2015 = pd.read_csv("2015_city_council.csv")
cc2017 = pd.read_csv("2017_CityCouncil_AtLarge_Turnout_Race.csv")
cc2019 = pd.read_csv("2019 CC Race Turnout.csv")
```

[ ]:
```
#checking that these are all the same length
print("Shape of cc2011:", cc2011.shape)
print("Shape of cc2013:", cc2013.shape)
print("Shape of cc2015:", cc2015.shape)
print("Shape of cc2017:", cc2017.shape)
print("Shape of cc2019:", cc2019.shape)
```

[ ]:
```
cc2011 = cc2011.drop([253])
```

[ ]:
```
cc2013 = cc2013.drop([253])
cc2015 = cc2015.drop([253])
cc2017 = cc2017.drop([253])
```

We will also disregard WP 2213 from each of the datasets since this data is incomplete.

Adding voter turnout column to each of the datasets

[ ]:
```
#beginning with 2011, add turnout column
cc2011["Turnout2011"] = (cc2011["BALLOTS CAST"]/cc2011["Registered Voters"]␣
↪*100).round(1)
cc2011["Turnout2011"]
```

```
[ ]: # adding turnout column to 2013
cc2013["Turnout2013"] = (cc2013["BALLOTS CAST"]/cc2013["Registered Voters"]␣
 ↪*100).round(1)
cc2013["Turnout2013"]
```

```
[ ]: # adding turnout column to 2015
cc2015["Turnout2015"] = (cc2015["BALLOTS CAST"]/cc2015["Registered Voters"]␣
 ↪*100).round(1)
cc2015["Turnout2015"]
```

```
[ ]: # 2017 already has a turnout column for some reason
cc2017 = cc2017.rename(columns= {"Turnout":"Turnout2017"})
cc2017["Turnout2017"] = (cc2017["Turnout2017"] *100).round(1)
```

```
[ ]: cc2017["Turnout2017"]
```

```
[ ]: # 2019
cc2019 = cc2019.rename(columns= {"Turnout":"Turnout2019"})
cc2019
```

```
[ ]: cc2019["Turnout2019"] = (cc2019["Turnout2019"] *100).round(1)
cc2019["Turnout2019"]
```

```
[ ]: cc2019
```

Creating a DataFrame with each of the voter turnouts

```
[ ]: # creating a new dataframe with turnout data
temp1 = cc2011[["WP", "Turnout2011"]]
turnouts = temp1.join(cc2013[["Turnout2013"]]).join(cc2015[["Turnout2015"]]).
 ↪join(cc2017[["Turnout2017"]]).join(cc2019[["Turnout2019"]])
turnouts
```

Creating a DataFrame of percent of total vote for each year

1.0.1 Visualizing the voter turnout across election year for each precinct.

```
[ ]: # visualizing our datasets 50 WPs at a time
turnouts[:50].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
 ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

```
[ ]: turnouts[50:100].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
 ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

```
[ ]: turnouts[100:150].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
 ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

```
[ ]: turnouts[150:200].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
     ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

```
[ ]: turnouts[200:].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
     ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

The 2013 race seems to have the highest voter turnout. Why is this?

Calculating the mean voter turnout for each election year

```
[ ]: print("Mean voter turnout 2011:", turnouts["Turnout2011"].mean().round(1), "%")
     print("Mean voter turnout 2013:", turnouts["Turnout2013"].mean().round(1), "%")
     print("Mean voter turnout 2015:", turnouts["Turnout2015"].mean().round(1), "%")
     print("Mean voter turnout 2017:", turnouts["Turnout2017"].mean().round(1), "%")
     print("Mean voter turnout 2019:", turnouts["Turnout2019"].mean().round(1), "%")
```

As we can see, 2013 has the highest average voter turnout.

Calculating the median voter turnout for each election year

```
[ ]: print("Median voter turnout 2011:", turnouts["Turnout2011"].median().round(1),␣
     ↪"%")
     print("Median voter turnout 2013:", turnouts["Turnout2013"].median(), "%")
     print("Median voter turnout 2015:", turnouts["Turnout2015"].median(), "%")
     print("Median voter turnout 2017:", turnouts["Turnout2017"].median(), "%")
     print("Median voter turnout 2017:", turnouts["Turnout2019"].median(), "%")
```

Again, 2013 is the highest.

Calculating average change over time

```
[ ]: import math
     turnouts["Diff11_13"] = turnouts["Turnout2011"] - turnouts["Turnout2013"]
     turnouts["Diff11_13"] = turnouts["Diff11_13"].abs()
     turnouts["Diff13_15"] = turnouts["Turnout2013"] - turnouts["Turnout2015"]
     turnouts["Diff13_15"] = turnouts["Diff13_15"].abs()
     turnouts["Diff15_17"] = turnouts["Turnout2015"] - turnouts["Turnout2017"]
     turnouts["Diff15_17"] = turnouts["Diff15_17"].abs()
     turnouts["Diff17_19"] = turnouts["Turnout2017"] - turnouts["Turnout2019"]
     turnouts["Diff17_19"] = turnouts["Diff17_19"].abs()
     turnouts["SumChange"] = turnouts["Diff11_13"] + turnouts["Diff13_15"] +␣
     ↪turnouts["Diff15_17"] + turnouts["Diff17_19"]
     turnouts["AvgChange"] = (turnouts["SumChange"] / 4.0).round(1)
     turnouts
```

1.0.2 Finding the Top 20 Precincts with the greatest average change in voter turnout.

```
[ ]: #finding the top 20 precincts with the greatest average change
top_change = turnouts.sort_values(by=['AvgChange'], ascending=False)
top_change
```

```
[ ]: #visualizing these top 20
top_change[:21].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
 ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

1.0.3 Preliminary Analysis of these Top 20 Precincts

```
[ ]: top_change[:21].drop(columns=['Diff11_13', 'Diff13_15', 'Diff15_17',␣
 ↪'Diff17_19', 'SumChange'])
```

```
[ ]: top20 = cc2011.iloc[[11, 10, 13, 12, 217, 20, 223, 221, 0, 218, 154, 216, 17,␣
 ↪8, 215, 19, 209, 211, 149, 219, 194]]
```

```
[ ]: top20["Black Percentage"] = top20["Black Percentage"].round(1)
top20["Hispanic Percentage"] = top20["Hispanic Percentage"].round(1)
top20["Asian Percentage"] = top20["Asian Percentage"].round(1)
```

```
[ ]: top20[["WP", "Black Percentage", "Hispanic Percentage", "Asian Percentage"]]
```

Breakdown by racial demographics:

```
[ ]: top20.plot(x="WP", y=["Black Percentage", "Hispanic Percentage", "Asian␣
 ↪Percentage"], kind="bar", figsize = (20,10))
```

```
[ ]: top_change.to_csv("TurnoutVolatility.csv", index = False)
```

This chart seems to indicate that the most "volatile" precincts in terms of voter turnout are either not dense in minority populations or have a significant Hispanic population. As we can see, a total of 6 precincts with a significant Hispanic population are in the Top 20 of this list.

1.1 We will now look at which precincts had the greatest change in share of total votes

Here, we use cc2011, cc2013, cc2015, and cc2017.

```
[ ]: sum_reg_11 = cc2011["BALLOTS CAST"].sum()
reg_11_lst = [sum_reg_11] * 253
cc2011["Sum of Ballots Cast 2011"] = reg_11_lst
```

```
[ ]: cc2011["Share of Voters 2011"] = (cc2011["BALLOTS CAST"]/cc2011["Sum of Ballots␣
 ↪Cast 2011"] * 100).round(5)
```

```
[ ]: cc2011["Share of Voters 2011"]
```

```
[ ]: sum_reg_13 = cc2013["BALLOTS CAST"].sum()
     reg_13_lst = [sum_reg_13] * 253
     cc2013["Sum of Ballots Cast 2013"] = reg_13_lst
```

```
[ ]: cc2013["Share of Voters 2013"] = (cc2013["BALLOTS CAST"]/cc2013["Sum of Ballots␣
     ↪Cast 2013"] * 100).round(5)
```

```
[ ]: cc2013["Share of Voters 2013"]
```

```
[ ]: sum_reg_15 = cc2015["BALLOTS CAST"].sum()
     reg_15_lst = [sum_reg_15] * 253
     cc2015["Sum of Ballots Cast 2015"] = reg_15_lst
```

```
[ ]: cc2015["Share of Voters 2015"] = (cc2015["BALLOTS CAST"]/cc2015["Sum of Ballots␣
     ↪Cast 2015"] * 100).round(5)
```

```
[ ]: cc2015["Share of Voters 2015"]
```

```
[ ]: sum_reg_17 = cc2017["BALLOTS CAST"].sum()
     reg_17_lst = [sum_reg_17] * 253
     cc2017["Sum of Ballots Cast 2017"] = reg_17_lst
```

```
[ ]: cc2017["Share of Voters 2017"] = (cc2017["BALLOTS CAST"]/cc2017["Sum of Ballots␣
     ↪Cast 2017"] * 100).round(5)
```

```
[ ]: sum_reg_19 = cc2019["Total Votes Cast"].sum()
     reg_19_lst = [sum_reg_19] * 253
     cc2019["Sum of Ballots Cast 2019"] = reg_19_lst
```

```
[ ]: cc2019["Share of Voters 2019"] = (cc2019["Total Votes Cast"]/cc2019["Sum of␣
     ↪Ballots Cast 2019"] * 100).round(5)
     cc2019["Share of Voters 2019"]
```

1.1.1 Determining the top 20 precincts with the greatest change in share of turnout over time

```
[ ]: temp2 = cc2011[["WP"]]
     shareturnouts = temp2.join(cc2011[["Share of Voters 2011"]]).
     ↪join(cc2013[["Share of Voters 2013"]]).join(cc2015[["Share of Voters␣
     ↪2015"]]).join(cc2017[["Share of Voters 2017"]]).join(cc2019[["Share of␣
     ↪Voters 2019"]])
     shareturnouts
```

```
[ ]: import math
     shareturnouts["Diff11_13"] = shareturnouts["Share of Voters 2011"] -␣
     ↪shareturnouts["Share of Voters 2013"]
```

```
shareturnouts["Diff11_13"] = shareturnouts["Diff11_13"].abs()

shareturnouts["Diff13_15"] = shareturnouts["Share of Voters 2013"] -␣
↪shareturnouts["Share of Voters 2015"]
shareturnouts["Diff13_15"] = shareturnouts["Diff13_15"].abs()

shareturnouts["Diff15_17"] = shareturnouts["Share of Voters 2015"] -␣
↪shareturnouts["Share of Voters 2017"]
shareturnouts["Diff15_17"] = shareturnouts["Diff15_17"].abs()

shareturnouts["Diff17_19"] = shareturnouts["Share of Voters 2019"] -␣
↪shareturnouts["Share of Voters 2017"]
shareturnouts["Diff17_19"] = shareturnouts["Diff17_19"].abs()

shareturnouts["SumChange"] = shareturnouts["Diff11_13"] +␣
↪shareturnouts["Diff13_15"] + shareturnouts["Diff15_17"] +␣
↪shareturnouts["Diff17_19"]
shareturnouts["AvgChange"] = (shareturnouts["SumChange"] / 4.0).round(5)
shareturnouts
```

[ ]:
```
#finding the top 20 precincts with the greatest average change
top_share_change = shareturnouts.sort_values(by=['AvgChange'], ascending=False)
top_share_change[:20]
```

[ ]:
```
#visualizing these top 20
top_share_change[:20].plot(x="WP", y=["Share of Voters 2011", "Share of Voters␣
↪2013", "Share of Voters 2015", "Share of Voters 2017", "Share of Voters␣
↪2019"], kind="bar", figsize = (20,10))
```

[ ]:
```
top20_list = [1612, 307, 113, 301, 202, 1704, 1714, 308, 509, 1816, 1310, 1609,␣
↪609, 703, 302, 1820, 701, 206, 306, 1912]
ilocs = []
indexes = cc2011["WP"].astype(int).to_list()
for l in top20_list:
    ilocs.append(indexes.index(l))
print(ilocs)
```

[ ]:
```
sharetop20 = cc2011.iloc[[154, 28, 12, 22, 16, 158, 168, 29, 48, 184, 119, 151,␣
↪59, 62, 23, 188, 60, 20, 27, 203]]
```

[ ]:
```
sharetop20["Black Percentage"] = sharetop20["Black Percentage"].round(1)
sharetop20["Hispanic Percentage"] = sharetop20["Hispanic Percentage"].round(1)
sharetop20["Asian Percentage"] = sharetop20["Asian Percentage"].round(1)
```

[ ]:
```
sharetop20[["WP","Black Percentage", "Hispanic Percentage", "Asian Percentage"]]
```

Breakdown by racial demographics

[ ]:
```
sharetop20.plot(x="WP", y=["Black Percentage", "Hispanic Percentage", "Asian␣
↪Percentage"], kind="bar", figsize = (20,10))
```

Here, we see the demographic breakdown of those districts which experienced the greatest AVERAGE change in share of total percentage of votes for each city council election. Here, we see that precints with a significant Black population seem to experience a significant average change.

[ ]:
```
top_share_change[:21].drop(columns=['Diff11_13', 'Diff13_15', 'Diff15_17',␣
↪'Diff17_19', 'SumChange'])
```

[ ]:
```
top_share_change.to_csv("TopShareChange.csv")
```

1.2 Deeper look: Wards 13, 15, and 16

[ ]:
```
def Dorcester_precincts(df):
    columns = df.columns.to_list()
    index = columns.index("WP")

    data_array = df.to_numpy()

    d_precs_int = [802, 806, 708, 707, 710, 709, 710, 1305, 1306, 1307, 1308,␣
↪1309, 1303, 1310, 1501, 1503, 1504, 1506, 1507, 1508, 1509, 1602, 1604,␣
↪1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1713, 115]
    d_precs = [str(i) for i in d_precs_int]
    rows = []
    for i in range(np.shape(data_array)[0]):
        current_p = str(int(data_array[i, index]))
        if current_p in d_precs:
            rows.append(data_array[i].tolist())

    Dorcester_df = pd.DataFrame(np.asarray(rows), columns = df.columns)
    if str(df["WP"].dtype) == "float64":
        Dorcester_df["WP"] = Dorcester_df["WP"].astype('Int64')
    return Dorcester_df
```

[ ]:
```
# shortening our volatility dataframe
top_change_d3 = Dorcester_precincts(top_change)
top_change_d3
```

[ ]:
```
top_change_d3[:10].drop(columns=['Diff11_13', 'Diff13_15', 'Diff15_17',␣
↪'Diff17_19', 'SumChange'])
```

[ ]:
```
top_change_d3.to_csv("TopChangeD3.csv")
```

```
[ ]: top10_list = [1612, 1607, 1609, 1611, 1310, 1608, 1713, 1309, 1610, 1602]
     indexes = cc2011["WP"].astype(int).to_list()
     ilocs1 = []
     for l in top10_list:
         ilocs1.append(indexes.index(l))
     ilocs1
```

```
[ ]: #visualizing these top 10
     top_change_d3[:10].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015",␣
      ↪"Turnout2017", "Turnout2019"], kind="bar", figsize = (20,10))
```

```
[ ]: top10d3 = cc2011.iloc[[154, 149, 151, 153, 119, 150, 167, 118, 152, 144]]
```

```
[ ]: top10d3.plot(x="WP", y=["Black Percentage", "Hispanic Percentage", "Asian␣
      ↪Percentage", "Native American Percentage"], kind="bar", figsize = (20,10))
```

1.2.1 Now finding the precincts in District 3 with the top share in change of voter turnout

```
[ ]: top_share_change_d3 = Dorcester_precincts(top_share_change)
```

```
[ ]: top_share_change_d3.to_csv("TopShareChangeD3.csv")
```

```
[ ]: top_share_change_d3[:10].drop(columns=['Diff11_13', 'Diff13_15', 'Diff15_17',␣
      ↪'Diff17_19', 'SumChange'])
```

```
[ ]: #visualizing these top 10
     top_share_change_d3[:10].plot(x="WP", y=["Share of Voters 2011", "Share of␣
      ↪Voters 2013", "Share of Voters 2015", "Share of Voters 2017", "Share of␣
      ↪Voters 2019"], kind="bar", figsize = (20,10))
```

```
[ ]: top10share_list = [1612, 1310, 1609, 1608, 1307, 1607, 1610, 1611, 1504, 1713]
     ilocs2 = []
     for l in top10share_list:
         ilocs2.append(indexes.index(l))
     ilocs2
```

```
[ ]: sharetop10d3 = cc2011.iloc[[154, 119, 151, 150, 116, 149, 152, 153, 137, 167]]
```

```
[ ]: sharetop10d3.plot(x="WP", y=["Black Percentage", "Hispanic Percentage", "Asian␣
      ↪Percentage", "Native American Percentage"], kind="bar", figsize = (20,10))
```

1.3 Determining the Share of District 3's Votes Over Time

Recall our processed DataFrames are cc2011, cc2013, cc2015, cc2017, and cc2019.

```
[ ]: cc2011d3 = Dorcester_precincts(cc2011)
     cc2013d3 = Dorcester_precincts(cc2013)
     cc2015d3 = Dorcester_precincts(cc2015)
```

```
[ ]: #adjusting cc2017 to fit our function
     cc2017["WP"] = cc2017["WP"].astype('object')
     if "WP" in cc2017.columns.to_list():
         print("yes")
```

```
[ ]: cc2017d3 = Dorcester_precincts(cc2017)
     cc2019d3 = Dorcester_precincts(cc2019.rename(columns={"Ward_Precinct_Code":
     ↪"WP"}))
```

```
[ ]: #finding the total number of votes cast in each election year

     total2011 = cc2011['BALLOTS CAST'].sum()
     total2013 = cc2013['BALLOTS CAST'].sum()
     total2015 = cc2015['BALLOTS CAST'].sum()
     total2017 = cc2017['BALLOTS CAST'].sum()
     total2019 = cc2019['Total Votes Cast'].sum()

     print(total2011, total2013, total2015, total2017, total2019)
```

```
[ ]: cc2011d3['BALLOTS CAST']
```

```
[ ]: total2011d3 = cc2011d3['BALLOTS CAST'].astype('float').sum()
     total2013d3 = cc2013d3['BALLOTS CAST'].astype('float').sum()
     total2015d3 = cc2015d3['BALLOTS CAST'].astype('float').sum()
     total2017d3 = cc2017d3['BALLOTS CAST'].astype('float').sum()
     total2019d3 = cc2019d3['Total Votes Cast'].astype('float').sum()

     print(total2011d3, total2013d3, total2015d3, total2017d3, total2019d3)
```

```
[ ]: lst_total = [total2011, total2013, total2015, total2017, total2019]
     lst_d3 = [total2011d3, total2013d3, total2015d3, total2017d3, total2019d3]
     lst_frac = []

     for i in range(len(lst_total)):
         lst_frac.append(round(lst_d3[i]/lst_total[i]*100,2))

     lst_frac
```

```
[ ]: #create new dataframe containing all of this info
     years = [2011, 2013, 2015, 2017, 2019]
```

```
d3change = pd.DataFrame(data={'Election Years': years, 'Total Ballots Cast':␣
↪lst_total, "Ballots Cast in D3": lst_d3, "Share of Ballots Cast by D3":␣
↪lst_frac})

d3change
```

[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns; sns.set_theme(color_codes=True)
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.barplot(x="Election Years", y="Share of Ballots Cast by D3",␣
↪data=d3change, palette = "flare")
plt.xlabel("Election Year")
plt.ylabel("Percent of Ballots Cast in District 3")
plt.title("Percent of Ballots Cast in District 3 by City Council Election Year")
```

[ ]:
```
fig, ax = plt.subplots(figsize=(14, 7))
ax = sns.regplot(x="Election Years", y="Share of Ballots Cast by D3",␣
↪data=d3change)
plt.xlabel("Election Year")
plt.ylabel("Percent of Ballots Cast in District 3")
plt.title("Percent of Ballots Cast in District 3 by City Council Election Year")
```

[ ]:
```

```

Deliverable 3

Questions to Answer:

1. How white turnout attrition since 2015 in vote total - Flaherty

2a. Michelle Wu how has her performance changed over time

2b. Annissa E George how has her performance changed over time

3. Add a subsection to your analysis which breaks down your findings for District 3 (Wards 13, 15, 16)

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def read_data(filename, wp):
    df = pd.read_csv(filename)
    df[wp] = df[wp].astype('Int64')
    #df['wp_string'] = df[wp].apply(lambda x: str(x).zfill(4))
    return df

df_2011 = read_data("2011_CityCouncil_Results_Race_Turnout.csv", "WP")
df_2013 = read_data("2013_CityCouncil_Race_Turnout_Results.csv", "WP")
df_2015 = read_data("2015_city_council.csv","WP").drop([253])
df_2017 = read_data("2017_CityCouncil_AtLarge_Turnout_Race.csv", "Ward
_Precinct").drop([253])
df_2019 = read_data("2019 CC Race Turnout.csv","Precinct_y")
```

In [ ]:
```
df_2015['MICHAEL F FLAHERTY'] = df_2015['MICHAEL F FLAHERTY']/df_2015[
'VOTES CAST']
df_2017['MICHAEL F FLAHERTY'] = df_2017['MICHAEL F FLAHERTY']/df_2017[
'VOTES CAST']
df_2019['Flaherty'] = df_2019['Flaherty']/df_2019['Total Votes Cast']
```

In [ ]:
```
print(df_2015['MICHAEL F FLAHERTY'])
```

In [ ]:
```
# 1. How white turnout attrition since 2015 in vote total - MICHAEL F
FLAHERTY

#ward_precinct and white(alone)
#df_f = df_2015[['wp_string','WP','White Percentage' ,'MICHAEL F FLAHE
RTY']]
df_f = df_2015[['WP','White Percentage' ,'MICHAEL F FLAHERTY']]
df_f = df_f.rename(columns= {"WP":"Precinct"})
df_f = df_f.rename(columns= {"MICHAEL F FLAHERTY":"Flaherty_2015"})
df_f = df_f.join(df_2017[["MICHAEL F FLAHERTY"]])
df_f = df_f.rename(columns= {"MICHAEL F FLAHERTY":"Flaherty_2017"})
df_f = df_f.join(df_2019[["Flaherty"]])
df_f = df_f.rename(columns= {"Flaherty":"Flaherty_2019"})
#df_f['Diff_15_17'] = round((df_f['Flaherty_2017'] - df_f['Flaherty_20
15'])/df_f['Flaherty_2015'], 3)
#df_f['Diff_17_19'] = round((df_f['Flaherty_2019'] - df_f['Flaherty_20
17'])/df_f['Flaherty_2017'], 3)
#df_f['Diff_15_19'] = round((df_f['Flaherty_2019'] - df_f['Flaherty_20
15'])/df_f['Flaherty_2015'], 3)
#df_f['Diff_Ave'] = round((abs(df_f['Diff_15_17']) + abs(df_f['Diff_17
_19']))/2, 3)
```

In [ ]:
```
df_f['Diff_15_17'] = round((df_f['Flaherty_2017'] - df_f['Flaherty_201
5']), 3)
df_f['Diff_17_19'] = round((df_f['Flaherty_2019'] - df_f['Flaherty_201
7']), 3)
df_f['Diff_15_19'] = round((df_f['Flaherty_2019'] - df_f['Flaherty_201
5']), 3)
df_f['Diff_Ave'] = round((abs(df_f['Diff_15_17']) + abs(df_f['Diff_17_
19']))/2, 3)
```

In [ ]:
```
print("Average Share of voters Change from 2015 to 2017:",df_f['Diff_1
5_17'].mean())
print("Average Share of voters Change from 2017 to 2019:",df_f['Diff_1
7_19'].mean())
print("Average Share of voters Change from 2015 to 2019:",df_f['Diff_1
5_19'].mean())
```

In [ ]:
```
top_white_fla = df_f.sort_values(by = ["White Percentage"],ascending =
False)
```

In [ ]:
```
fig, ax = plt.subplots(ncols=2,figsize = (20,5))
top_white_fla[:20].plot(x = "Precinct", y = ['Flaherty_2015','Flaherty
_2017','Flaherty_2019'], kind = 'bar', title = "Precincts with most Wh
ite population (top 20) and Share of voters each year",ax=ax[0])
top_white_fla[-20:].plot(x = "Precinct", y = ['Flaherty_2015','Flahert
y_2017','Flaherty_2019'], kind = 'bar',title = "Precincts with lowest
White population (last 20) and Share of voters each year",ax=ax[1])
```

In [ ]:
```
fig, ax = plt.subplots(ncols=2,figsize = (20,5))
top_white_fla[:20].plot(x = "Precinct", y = ['Diff_15_17','Diff_17_19'
], kind = 'bar', title = "Precincts with most White population (top 20
) and Share of voters change", ax = ax[0])
top_white_fla[-20:].plot(x = "Precinct", y = ['Diff_15_17','Diff_17_19
'], kind = 'bar', title = "Precincts with lowest White population (las
t 20) and Share of voters change", ax = ax[1])
```

In [ ]:
```
fig, ax = plt.subplots(ncols=2,figsize = (20,5))
top_white_fla[:20].plot(x = "Precinct", y = ['Diff_15_19'], kind = 'ba
r', title = "Precincts with most White population (top 20) and Share o
f voters net change", ax = ax[0])

top_white_fla[-20:].plot(x = "Precinct", y = ['Diff_15_19'], kind = 'b
ar', title = "Precincts with lowest White population (last 20) and Sha
re of voters net change", ax = ax[1])
```

In [ ]:
```
top_change_fla = df_f.sort_values(by = ["Diff_Ave"],ascending = False)
top_change_fla[:20].plot(x = "Precinct", y = ['White Percentage'], kin
d = 'bar', title = "Precincts with most dramatic change of Share of vo
ters (top 20) and White Percentage")
# >0.048
```

In [ ]:
```
df_2013['MICHELLE WU'] = df_2013['MICHELLE WU']/df_2013['VOTES CAST']
df_2015['MICHELLE WU'] = df_2015['MICHELLE WU']/df_2015['VOTES CAST']
df_2017['MICHELLE WU'] = df_2017['MICHELLE WU']/df_2017['VOTES CAST']
df_2019['Wu'] = df_2019['Wu']/df_2019['Total Votes Cast']
```

In [ ]:
```
# 2. Michelle Wu how has her performance changed over time

# She started to run the city council starting from 2013

def turnout_change_race():
    #df_f = df_2013[['wp_string','WP','Black Percentage', 'Hispanic Pe
rcentage', 'Asian Percentage' , 'White Percentage','MICHELLE WU']]
    df_f = df_2013[['WP','Black Percentage', 'Hispanic Percentage', 'A
sian Percentage' , 'White Percentage','MICHELLE WU']]
    df_f = df_f.rename(columns= {"MICHELLE WU":"WU_2013"})
    df_f = df_f.rename(columns= {"WP":"Precinct"})

    df_f = df_f.join(df_2015[["MICHELLE WU"]])
    df_f = df_f.rename(columns= {"MICHELLE WU":"WU_2015"})

    df_f = df_f.join(df_2017[["MICHELLE WU"]])
    df_f = df_f.rename(columns= {"MICHELLE WU":"WU_2017"})

    df_f = df_f.join(df_2019[["Wu"]])
    df_f = df_f.rename(columns= {"Wu":"WU_2019"})
    return df_f


df_wu = turnout_change_race()

df_wu['Diff_13_15'] = round((df_wu['WU_2015'] - df_wu['WU_2013']), 3)
df_wu['Diff_15_17'] = round((df_wu['WU_2017'] - df_wu['WU_2015']), 3)
df_wu['Diff_17_19'] = round((df_wu['WU_2019'] - df_wu['WU_2017']), 3)
df_wu['Diff_Ave'] = round((abs(df_wu['Diff_13_15']) + abs(df_wu['Diff_
15_17'])+ abs(df_wu['Diff_17_19']))/3, 3)

df_wu['Diff_net_13_19'] = round((df_wu['WU_2019'] - df_wu['WU_2013']),
3)
```

In [ ]:
```
print("Average Turnout Change from 2013 to 2015:",df_wu['Diff_13_15'].
mean())
print("Average Turnout Change from 2015 to 2017:",df_wu['Diff_15_17'].
mean())
print("Average Turnout Change from 2017 to 2019:",df_wu['Diff_17_19'].
mean())
#print("Average Turnout Change from 2015 to 2017:",df_wu['Diff_Ave'].m
ean())

print("Average Turnout Change from 2013 to 2019:",df_wu['Diff_net_13_1
9'].mean())
```

In [ ]:
```
top_change_wu = df_wu.sort_values(by = ["Diff_Ave"],ascending = False)
top_change_wu[:20].plot(x = "Precinct", y = ['Black Percentage', 'Hispanic Percentage', 'Asian Percentage','White Percentage'], kind = 'bar', title = "Precincts with most dramatic change of Share of voters (top 20) and different races")
# >=0.063
```

In [ ]:
```
top_white_wu = df_wu.sort_values(by = ["White Percentage"],ascending = False)
top_black_wu = df_wu.sort_values(by = ["Black Percentage"],ascending = False)
top_his_wu = df_wu.sort_values(by = ['Hispanic Percentage'],ascending = False)
top_asian_wu = df_wu.sort_values(by = ['Asian Percentage'],ascending = False)

# visualizing least 20 (white)
#top_white_wu[-20:].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_2017','WU_2019'], kind = 'bar', title = "Precincts with lowest White population (last 20) and Turnout")
#top_white_wu[-20:].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1,2,3,4,5], title = "Precincts with lowest White population (last 20) and Turnout change ratio")
#top_white_wu[-20:].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1],title = "Precincts with lowest White population (last 20) and Turnout net change ratio")

# visualizing least 20 (black)
#top_black_wu[-20:].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_2017','WU_2019'], kind = 'bar',title = "Precincts with lowest Black population (last 20) and Turnout")
#top_black_wu[-20:].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1,2,3,4,5],title = "Precincts with lowest Black population (last 20) and Turnout change ratio")
#top_black_wu[-20:].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1],title = "Precincts with lowest Black population (last 20) and Turnout net change ratio")

# visualizing least 20 (Hispanic or Latino)
#top_his_wu[-20:].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_2017','WU_2019'], kind = 'bar',title = "Precincts with lowest Hispanic population (last 20) and turnout")
#top_his_wu[-20:].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1,2,3,4,5], title = "Precincts with lowest Hispanic population (last 20) and turnout change ratio")
#top_his_wu[-20:].plot(x = "Precinct", y = ['Diff_net_13_19'], kind =
```

```
'bar',yticks = [-1,-0.5,0,0.5,1],title = "Precincts with lowest Hispan
ic population (last 20) and Turnout net change ratio")

# visualizing least 20 (Asian)
#top_asian_wu[-20:].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_
2017','WU_2019'], kind = 'bar',title = "Precincts with lowest Asian po
pulation (last 20) and turnout")
#top_asian_wu[-20:].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17
','Diff_17_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1,2,3,4,5], titl
e = "Precincts with lowest Asian population (last 20) and Turnout chan
ge ratio")
#top_asian_wu[-20:].plot(x = "Precinct", y = ['Diff_net_13_19'], kind
= 'bar',yticks = [-1,-0.5,0,0.5,1],title = "Precincts with lowest Asia
n population (last 20) and Turnout net change ratio")
```

In [ ]:

```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_20
17','WU_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.
35], title = "Precincts with most White population (top 20) and Share
of voters", ax = ax[0,0])
top_black_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_20
17','WU_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.
35], title = "Precincts with most Black population (top 20) and Share
of voters",ax = ax[1,0])
top_his_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_2017
','WU_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,0.3
5],title = "Precincts with most Hispanic population (top 20) and Share
of voters",ax = ax[0,1])
top_asian_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_20
17','WU_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,0
.35],title = "Precincts with most Asian population (top 20) and Share
of voters",ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17',
'Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "
Precincts with most White population (top 20) and Share of voters chan
ge", ax = ax[0,0])
top_black_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17',
'Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1], title =
"Precincts with most Black population (top 20) and Share of voters cha
nge", ax =ax[1,0])
top_his_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','D
iff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "Pr
ecincts with most Hispanic population (top 20) and Share of voters cha
nge" , ax =ax[0,1])
top_asian_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17',
'Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "
Precincts with most Asian population (top 20) and Share of voters chan
ge", ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind =
'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts
with most White population (top 20) and Share of voters net change", a
x = ax[0,0])

top_black_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind =
'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts
with most Black population (top 20) and Share of voters net change", a
x =ax[1,0])

top_his_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'b
ar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts wi
th most Hispanic population (top 20) and Share of voters net change",a
x =ax[0,1])

top_asian_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind =
'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts
with most Asian population (top 20) and Share of voters net change", a
x =ax[1,1])
```

In [ ]:
```
df_2013['ANNISSA ESSAIBI GEORGE'] = df_2013['ANNISSA ESSAIBI GEORGE']/
df_2013['VOTES CAST']
df_2015['ANNISSA ESSAIBI GEORGE'] = df_2015['ANNISSA ESSAIBI GEORGE']/
df_2015['VOTES CAST']
df_2017['ANNISSA E GEORGE'] = df_2017['ANNISSA E GEORGE']/df_2017['VOT
ES CAST']
df_2019['Essaibi-George'] = df_2019['Essaibi-George']/df_2019['Total V
otes Cast']
```

In [ ]:
```
# 2b. Annissa E George how has her performance changed over time

# She started to run the city council starting from 2013

def turnout_change_race():
    #df_f = df_2013[['wp_string','WP','Black Percentage', 'Hispanic Pe
rcentage', 'Asian Percentage' , 'White Percentage','MICHELLE WU']]
    df_f = df_2013[['WP','Black Percentage', 'Hispanic Percentage', 'A
sian Percentage' , 'White Percentage','ANNISSA ESSAIBI GEORGE']]
    df_f = df_f.rename(columns= {"ANNISSA ESSAIBI GEORGE":"ANNI_2013"}
)
    df_f = df_f.rename(columns= {"WP":"Precinct"})

    df_f = df_f.join(df_2015[["ANNISSA ESSAIBI GEORGE"]])
    df_f = df_f.rename(columns= {"ANNISSA ESSAIBI GEORGE":"ANNI_2015"}
)

    df_f = df_f.join(df_2017[["ANNISSA E GEORGE"]])
    df_f = df_f.rename(columns= {"ANNISSA E GEORGE":"ANNI_2017"})

    df_f = df_f.join(df_2019[["Essaibi-George"]])
    df_f = df_f.rename(columns= {"Essaibi-George":"ANNI_2019"})
    return df_f


df_anni = turnout_change_race()

df_anni['Diff_13_15'] = round((df_anni['ANNI_2015'] - df_anni['ANNI_20
13']), 3)
df_anni['Diff_15_17'] = round((df_anni['ANNI_2017'] - df_anni['ANNI_20
15']), 3)
df_anni['Diff_17_19'] = round((df_anni['ANNI_2019'] - df_anni['ANNI_20
17']), 3)
df_anni['Diff_Ave'] = round((abs(df_anni['Diff_13_15']) + abs(df_anni[
'Diff_15_17'])+ abs(df_anni['Diff_17_19']))/3, 3)

df_anni['Diff_net_13_19'] = round((df_anni['ANNI_2019'] - df_anni['ANN
I_2013']), 3)
```

In [ ]:
```
print("Average Turnout Change from 2013 to 2015:",df_anni['Diff_13_15'
].mean())
print("Average Turnout Change from 2015 to 2017:",df_anni['Diff_15_17'
].mean())
print("Average Turnout Change from 2017 to 2019:",df_anni['Diff_17_19'
].mean())
#print("Average Turnout Change from 2015 to 2017:",df_wu['Diff_Ave'].m
ean())

print("Average Turnout Change from 2013 to 2019:",df_anni['Diff_net_13
_19'].mean())
```

In [ ]:
```
top_change_anni = df_anni.sort_values(by = ["Diff_Ave"],ascending = Fa
lse)
top_change_anni[:20].plot(x = "Precinct", y = ['Black Percentage', 'Hi
spanic Percentage', 'Asian Percentage','White Percentage'], kind = 'ba
r', title = "Precincts with most dramatic change of Share of voters (t
op 20) and different races")
# >=0.063
```

In [ ]:
```
top_white_anni = df_anni.sort_values(by = ["White Percentage"],ascendi
ng = False)
top_black_anni = df_anni.sort_values(by = ["Black Percentage"],ascendi
ng = False)
top_his_anni = df_anni.sort_values(by = ['Hispanic Percentage'],ascend
ing = False)
top_asian_anni = df_anni.sort_values(by = ['Asian Percentage'],ascendi
ng = False)
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_anni[:20].plot(x = "Precinct", y = ['ANNI_2013','ANNI_2015',
'ANNI_2017','ANNI_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0
.25,0.3], title = "Precincts with most White population (top 20) and S
hare of voters", ax = ax[0,0])
top_black_anni[:20].plot(x = "Precinct", y = ['ANNI_2013','ANNI_2015',
'ANNI_2017','ANNI_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0
.25,0.3], title = "Precincts with most Black population (top 20) and S
hare of voters",ax = ax[1,0])
top_his_anni[:20].plot(x = "Precinct", y = ['ANNI_2013','ANNI_2015','A
NNI_2017','ANNI_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,0.
25,0.3],title = "Precincts with most Hispanic population (top 20) and
Share of voters",ax = ax[0,1])
top_asian_anni[:20].plot(x = "Precinct", y = ['ANNI_2013','ANNI_2015',
'ANNI_2017','ANNI_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,
0.25,0.3],title = "Precincts with most Asian population (top 20) and S
hare of voters",ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_anni[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "Precincts with most White population (top 20) and Share of voters change", ax = ax[0,0])
top_black_anni[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1], title = "Precincts with most Black population (top 20) and Share of voters change", ax =ax[1,0])
top_his_anni[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "Precincts with most Hispanic population (top 20) and Share of voters change" , ax =ax[0,1])
top_asian_anni[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "Precincts with most Asian population (top 20) and Share of voters change", ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
top_white_anni[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts with most White population (top 20) and Share of voters net change", ax = ax[0,0])

top_black_anni[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts with most Black population (top 20) and Share of voters net change", ax =ax[1,0])

top_his_anni[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts with most Hispanic population (top 20) and Share of voters net change",ax =ax[0,1])

top_asian_anni[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind = 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts with most Asian population (top 20) and Share of voters net change", ax =ax[1,1])
```

In [ ]:
```
# 3. Add a subsection to your analysis which breaks down your findings
for District 3 (Wards 13, 15, 16)
def Dorcester_precincts(df):
    columns = df.columns.to_list()
    index = columns.index("Precinct")

    data_array = df.to_numpy()

    rows = []
    for i in range(np.shape(data_array)[0]):
        current_p = str(int(data_array[i, index]))
        if len(current_p) == 4:
            if current_p[0:2] == '13' or current_p[0:2] == '15' or cur
rent_p[0:2] == '16' or current_p[0:2] == '17':
                rows.append(data_array[i].tolist())

    Dorcester_df = pd.DataFrame(np.asarray(rows), columns = df.columns
)
    Dorcester_df["Precinct"] = Dorcester_df["Precinct"].astype('Int64'
)
    return Dorcester_df

d_df_wu = Dorcester_precincts(df_wu)
d_df_f = Dorcester_precincts(df_f)
```

In [ ]:
```
print(d_df_f.head())
```

In [ ]:
```
print("Average Turnout Change from 2015 to 2017:", d_df_f['Diff_15_17'
].mean())
print("Average Turnout Change from 2017 to 2019:", d_df_f['Diff_17_19'
].mean())

print("Average Turnout Change from 2015 to 2019:", d_df_f['Diff_15_19'
].mean())
```

In [ ]:
```
d_top_change_fla = d_df_f.sort_values(by = ["Diff_Ave"],ascending = Fa
lse)
d_top_change_fla[:20].plot(x = "Precinct", y = ['White Percentage'], k
ind = 'bar', title = "Precincts with most dramatic change of Share of
voters (top 20) and White Percentage")
```

In [ ]:
```
d_top_white_fla = d_df_f.sort_values(by = ["White Percentage"],ascendi
ng = False)
# visualizing top 20
d_top_white_fla[:20].plot(x = "Precinct", y = ['Flaherty_2015','Flaher
ty_2017','Flaherty_2019'], kind = 'bar', title = "Precincts with most
White population in Dorcester(top 20) and Share of voters each year")
d_top_white_fla[:20].plot(x = "Precinct", y = ['Diff_15_17','Diff_17_1
9'], kind = 'bar', title = "Precincts with most White population in Do
rcester(top 20) and Share of voters change")
d_top_white_fla[:20].plot(x = "Precinct", y = ['Diff_15_19'], kind = '
bar', title = "Precincts with most White population in Dorcester (top
20) and Share of voters net change")

# visualizing least 20 (white)
#d_top_white_wu[-20:].plot(x = "Precinct", y = ['WU_2013','WU_2015','W
U_2017','WU_2019'], kind = 'bar', title = "Precincts with lowest White
population in Dorcester (last 20) and Turnout")
#d_top_white_wu[-20:].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_
17','Diff_17_19'], kind = 'bar',yticks = [-1,-0.5,0,0.5,1,2,3,4,5], ti
tle = "Precincts with lowest White population in Dorcester (last 20) a
nd Turnout change ratio")
#d_top_white_wu[-20:].plot(x = "Precinct", y = ['Diff_net_13_19'], kin
d = 'bar',yticks = [-1,-0.5,0,0.5,1],title = "Precincts with lowest Wh
ite population in Dorcester (last 20) and Turnout net change ratio")
```

In [ ]:
```
print("Average Turnout Change from 2013 to 2015:", d_df_wu['Diff_13_15
'].mean())
print("Average Turnout Change from 2015 to 2017:", d_df_wu['Diff_15_17
'].mean())
print("Average Turnout Change from 2017 to 2019:", d_df_wu['Diff_17_19
'].mean())

print("Average Turnout Change from 2013 to 2019:", d_df_wu['Diff_net_1
3_19'].mean())
```

In [ ]:
```
d_top_change_wu = d_df_wu.sort_values(by = ["Diff_Ave"],ascending = Fa
lse)
d_top_change_wu[:20].plot(x = "Precinct", y = ['Black Percentage', 'Hi
spanic Percentage', 'Asian Percentage','White Percentage'], kind = 'ba
r', title = "Precincts with most dramatic change of Share of voters (t
op 20) and different races")
```

In [ ]:
```
d_top_white_wu = d_df_wu.sort_values(by = ["White Percentage"],ascendi
ng = False)
d_top_black_wu = d_df_wu.sort_values(by = ["Black Percentage"],ascendi
ng = False)
d_top_his_wu = d_df_wu.sort_values(by = ['Hispanic Percentage'],ascend
ing = False)
d_top_asian_wu = d_df_wu.sort_values(by = ['Asian Percentage'],ascendi
ng = False)
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
d_top_white_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_
2017','WU_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,
0.35], title = "Precincts with most White population in Dorcester (top
20) and Share of voters", ax = ax[0,0])
d_top_black_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_
2017','WU_2019'], kind = 'bar',yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,
0.35], title = "Precincts with most Black population in Dorcester (top
20) and Share of voters",ax = ax[1,0])
d_top_his_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_20
17','WU_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3,0
.35],title = "Precincts with most Hispanic population in Dorcester (to
p 20) and Share of voters",ax = ax[0,1])
d_top_asian_wu[:20].plot(x = "Precinct", y = ['WU_2013','WU_2015','WU_
2017','WU_2019'], kind = 'bar', yticks = [0,0.05,0.1,0.15,0.2,0.25,0.3
,0.35],title = "Precincts with most Asian population in Dorcester (top
20) and Share of voters",ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
d_top_white_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17
','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title =
"Precincts with most White population in Dorcester (top 20) and Share
of voters change", ax = ax[0,0])
d_top_black_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17
','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1], title
= "Precincts with most Black population in Dorcester (top 20) and Shar
e of voters change", ax =ax[1,0])
d_top_his_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17',
'Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title = "
Precincts with most Hispanic population in Dorcester (top 20) and Shar
e of voters change" , ax =ax[0,1])
d_top_asian_wu[:20].plot(x = "Precinct", y = ['Diff_13_15','Diff_15_17
','Diff_17_19'], kind = 'bar',yticks = [-0.1,-0.05,0,0.05,0.1],title =
"Precincts with most Asian population in Dorcester (top 20) and Share
of voters change", ax =ax[1,1])
```

In [ ]:
```
fig, ax = plt.subplots(nrows = 2,ncols=2,figsize = (20,15))
d_top_white_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind
= 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precinct
s with most White population in Dorcester (top 20) and Share of voters
net change", ax = ax[0,0])

d_top_black_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind
= 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precinct
s with most Black population in Dorcester (top 20) and Share of voters
net change", ax =ax[1,0])

d_top_his_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind =
'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precincts
with most Hispanic population in Dorcester (top 20) and Share of voter
s net change",ax =ax[0,1])

d_top_asian_wu[:20].plot(x = "Precinct", y = ['Diff_net_13_19'], kind
= 'bar',yticks = [-0.04,-0.02,0,0.02,0.04,0.06,0.08],title = "Precinct
s with most Asian population in Dorcester (top 20) and Share of voters
net change", ax =ax[1,1])
```

In [ ]:
```
# export all datasets
df_f['wp_string'] = df_f['Precinct'].apply(lambda x: str(x).zfill(4))
df_wu['wp_string'] = df_wu['Precinct'].apply(lambda x: str(x).zfill(4)
)
df_f.to_csv("fra.csv")
df_wu.to_csv("wu.csv")
```

In [ ]:
```
```

merging_CC_turnouts_2011_to_2019

April 29, 2021

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def clean_data(filename):
    df = pd.read_excel(filename,index=False)
    df.replace(r'', np.nan)
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

def Dorcester_precincts(df):
    columns = df.columns.to_list()
    index = columns.index("Ward")

    data_array = df.to_numpy()

    rows = []
    for i in range(np.shape(data_array)[0]):
        current_p = str(int(data_array[i, index]))
        if len(current_p) == 4:
            if current_p[0:2] == '13' or current_p[0:2] == '15' or current_p[0:
↪2] == '16' or current_p[0:2] == '17':
                rows.append(data_array[i].tolist())

    Dorcester_df = pd.DataFrame(np.asarray(rows), columns = df.columns)
    #Dorcester_df["Ward"] = Dorcester_df["Ward"].astype('Int64')
    return Dorcester_df

CC_all_yrs = clean_data("CC_turnout_all_years.xlsx")
df_2011 = clean_data("2011_CityCouncil_Results_Race_Turnout.xlsx")
df_2013 = clean_data("2013_CityCouncil_Race_Turnout_Results.xlsx")
df_2015 = clean_data("2015_city_council.xlsx")
df_2017 = clean_data("2017_CityCouncil_AtLarge_Turnout_Race.xlsx")
df_2019 = clean_data("2019_CityCouncil_Race Turnout.xlsx")
df_2011
```

```
[1]:      Unnamed: 0  WILL DORCENA  AYANNA PRESSLEY  FELIX G ARROYO  \
     0             0            16              113             111
     1             1             9               44              55
     2             2            40              133             155
     3             3             5               29              47
     4             4            12               54              67
     ..          ...           ...              ...             ...
     248         248            16               71              88
     249         249            18               75              86
     250         250            32              123             126
     251         251            11               66              74
     252         252            22               60              79

          JOHN R CONNOLLY  MICHAEL F FLAHERTY  STEPHEN J MURPHY  SEAN H RYAN  \
     0                117                 100                99           24
     1                 42                  70                52           10
     2                123                 101               109           34
     3                 40                  45                43            8
     4                 63                  69                63           11
     ..               ...                 ...               ...          ...
     248               90                  75                72           18
     249               75                  60                55           17
     250              112                  99               119           30
     251               84                  52                77           12
     252               79                  86                75           23

          WILLIAM B FEEGBEH Write-in  DESHON PORTER Write-in  …  \
     0                             0                       0  …
     1                             0                       0  …
     2                             0                       0  …
     3                             0                       0  …
     4                             0                       0  …
     ..                          ...                     ... ...
     248                           0                       0  …
     249                           0                       0  …
     250                           0                       0  …
     251                           0                       0  …
     252                           0                       0  …

          Asian Percentage  Native Hawaiian/Pacific Islander Percentage  \
     0            2.853815                                     0.232964
     1            2.028006                                     0.000000
     2            7.172067                                     0.000000
     3            2.016985                                     0.053079
     4            2.631579                                     0.173130
     ..                ...                                          ...
     248         17.288136                                     0.000000
```

```
249         15.807730                                    0.099108
250         10.237510                                    0.000000
251         11.787072                                    0.000000
252         14.702309                                    0.060753

     Other Race Percentage  Two or more races Percentage  Hispanic Percentage  \
0                10.075713                      3.494467            29.295282
1                16.948334                      4.538870            51.810719
2                15.539478                      5.662158            41.333753
3                29.458599                      6.581741            63.853503
4                31.024931                      5.297784            57.617729
..                     …                            …                  …
248               1.259080                      1.791768             5.230024
249               1.635282                      2.973241             4.905847
250               2.989353                      1.801802             8.476658
251               2.205323                      3.117871             8.288973
252               9.599028                      4.009721            18.165249

     White Percentage  White plurality (Y/N)  Black plurality (Y/N)  \
0           79.732091                      Y                      N
1           73.394495                      Y                      N
2           62.566845                      Y                      N
3           58.227176                      N                      N
4           57.444598                      N                      N
..                …                    …                    …
248         76.658596                      Y                      N
249         77.502478                      Y                      N
250         81.244881                      Y                      N
251         77.490494                      Y                      N
252         61.178615                      Y                      N

     Hispanic plurality (Y/N)  Asian plurality (Y/N)
0                           N                      N
1                           N                      N
2                           N                      N
3                           Y                      N
4                           Y                      N
..                        …                    …
248                         N                      N
249                         N                      N
250                         N                      N
251                         N                      N
252                         N                      N

[253 rows x 48 columns]
```

[2]:
```python
def get_turnout(df):
    df['Turnout'] = round(df['BALLOTS CAST']/df['Registered Voters'],2)
    return df

def processing(df):
    conditions = [df['Black Percentage'] >= 50,  df['White Percentage'] >= 50]
    choices = ['black', 'white']
    df['majority_race'] = np.select(conditions, choices, "POC")
    return df[["Ward", 'majority_race', 'Turnout']]
```

[3]:
```python
df_2011 = processing(get_turnout(df_2011))
df_2011 = df_2011.rename(columns={"Turnout": "Turnout_2011"})

df_2013 = processing(get_turnout(df_2013))
df_2013 = df_2013.rename(columns={"Turnout": "Turnout_2013"})

df_2015 = processing(get_turnout(df_2015))
df_2015 = df_2015.rename(columns={"Turnout": "Turnout_2015"})

df_2017 = df_2017.rename(columns={"Ward_Precinct": "Ward"})
df_2017 = processing(get_turnout(df_2017))
df_2017 = df_2017.rename(columns={"Turnout": "Turnout_2017"})

df_2019 = df_2019.drop('Ward', axis=1)
df_2019 = df_2019.rename(columns={"Ward_Precinct_Code": "Ward"})
df_2019 = processing(df_2019)
df_2019 = df_2019.rename(columns={"Turnout": "Turnout_2019"})
df_2019['Turnout_2019'] = round(df_2019['Turnout_2019'], 2)
```

[4]:
```python
df_2019['Ward'] = df_2019.Ward.astype(int)

df_2019
```

[4]:
```
      Ward majority_race  Turnout_2019
0      505         white          0.22
1      503         white          0.16
2      506         white          0.10
3      308           POC          0.10
4     2106         white          0.06
..     …             …            …
249    702         white          0.05
250    609         white          0.04
251    701         white          0.06
252   1609         white          0.07
253   1612         white          0.09

[252 rows x 3 columns]
```

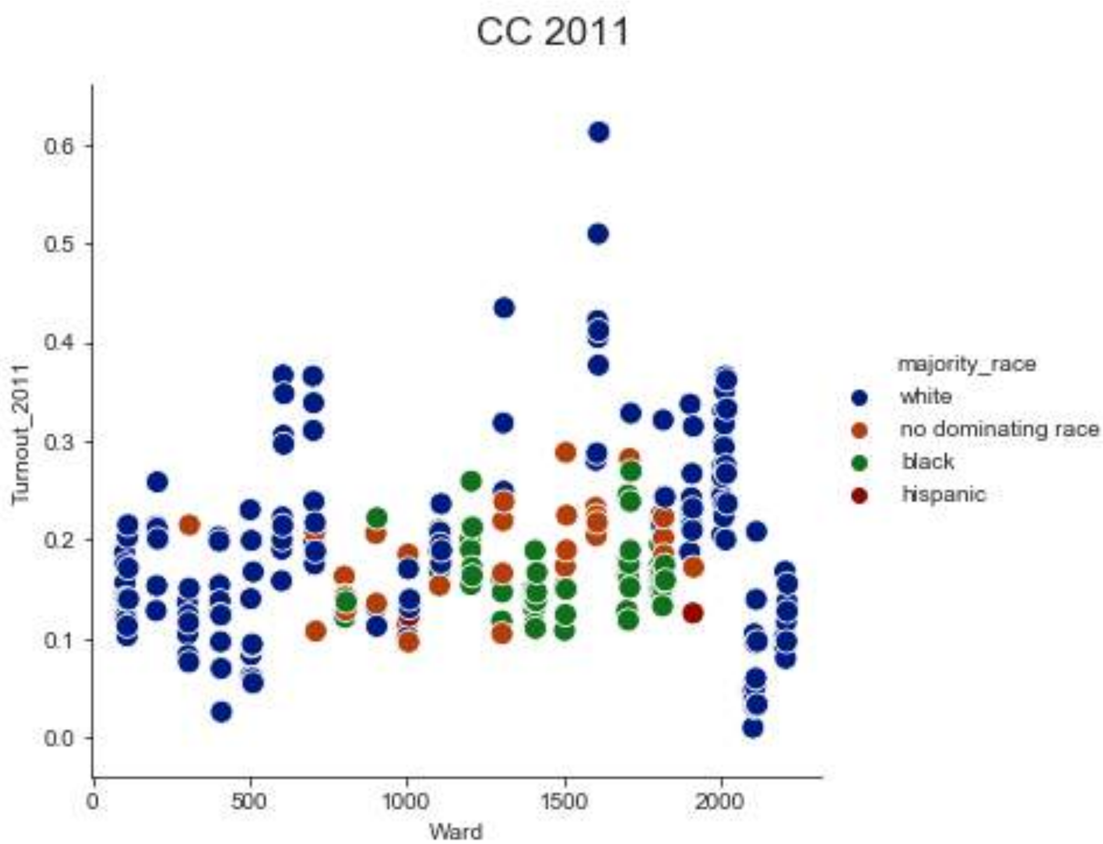
[5]:
```
df_final = df_2011.merge(df_2013,on=['Ward','majority_race']).
↪merge(df_2015,on=['Ward','majority_race']).
↪merge(df_2017,on=['Ward','majority_race']).
↪merge(df_2019,on=['Ward','majority_race'])
```

[6]:
```
df_final
```

[6]:
```
     Ward majority_race  Turnout_2011  Turnout_2013  Turnout_2015  \
0     101         white          0.19          0.52          0.14
1     102         white          0.15          0.45          0.11
2     103         white          0.16          0.44          0.13
3     104         white          0.18          0.44          0.09
4     105         white          0.14          0.43          0.08
..    …             …            …             …             …
247  2208         white          0.12          0.32          0.09
248  2209         white          0.10          0.27          0.08
249  2210         white          0.14          0.36          0.11
250  2211         white          0.13          0.34          0.11
251  2212         white          0.16          0.32          0.12

     Turnout_2017  Turnout_2019
0            0.35          0.10
1            0.29          0.04
2            0.30          0.07
3            0.29          0.04
4            0.28          0.03
..            …             …
247          0.23          0.07
248          0.21          0.08
249          0.27          0.10
250          0.25          0.12
251          0.24          0.09

[252 rows x 7 columns]
```

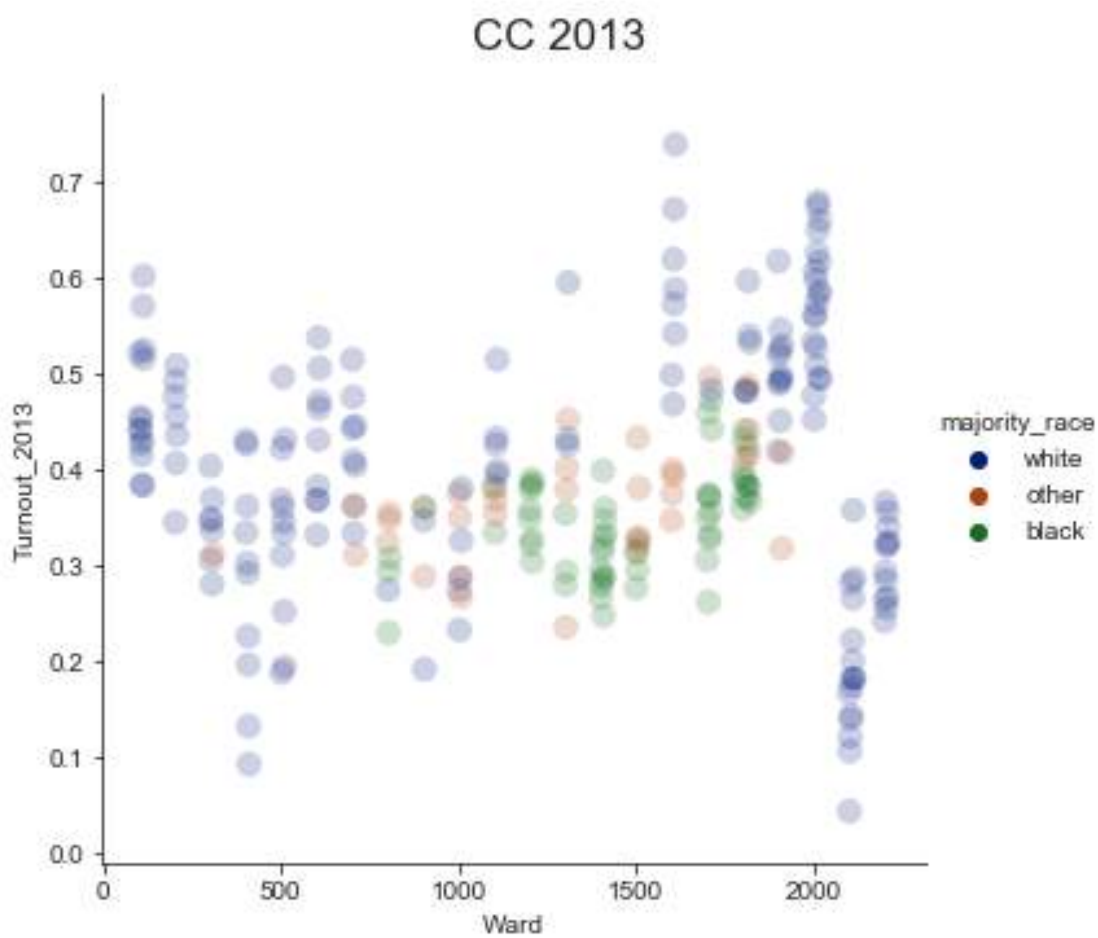
[17]:
```
sns.set_style("ticks")

g = sns.relplot(data=df_final, x="Ward", y="Turnout_2011",␣
↪hue="majority_race",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('CC 2011',fontsize=17)
#g.legend.set_title("Precints Won")
#g.set(xticks=np.arange(1,23,2))
#g.axes[0][0].axhline(50, ls='--')
```

[17]:
```
Text(0.5, 0.98, 'CC 2011')
```



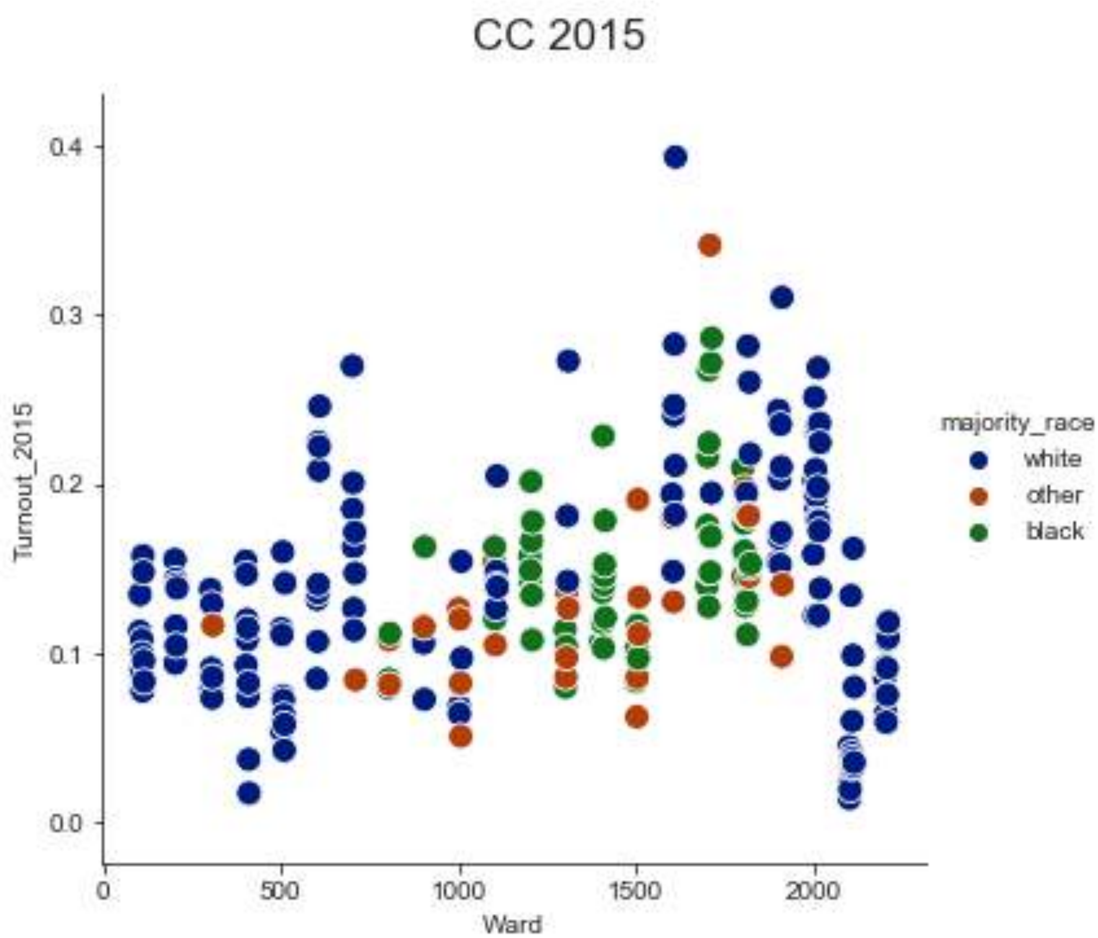

```
[44]: g = sns.relplot(data=df_final, x="Ward", y="Turnout_2013",␣
      ↪hue="majority_race",palette="dark", kind='scatter', s=100,alpha=.2)
      g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
      g.fig.suptitle('CC 2013',fontsize=17)
      #g.legend.set_title("Precints Won")
```

```
[44]: Text(0.5, 0.98, 'CC 2013')
```



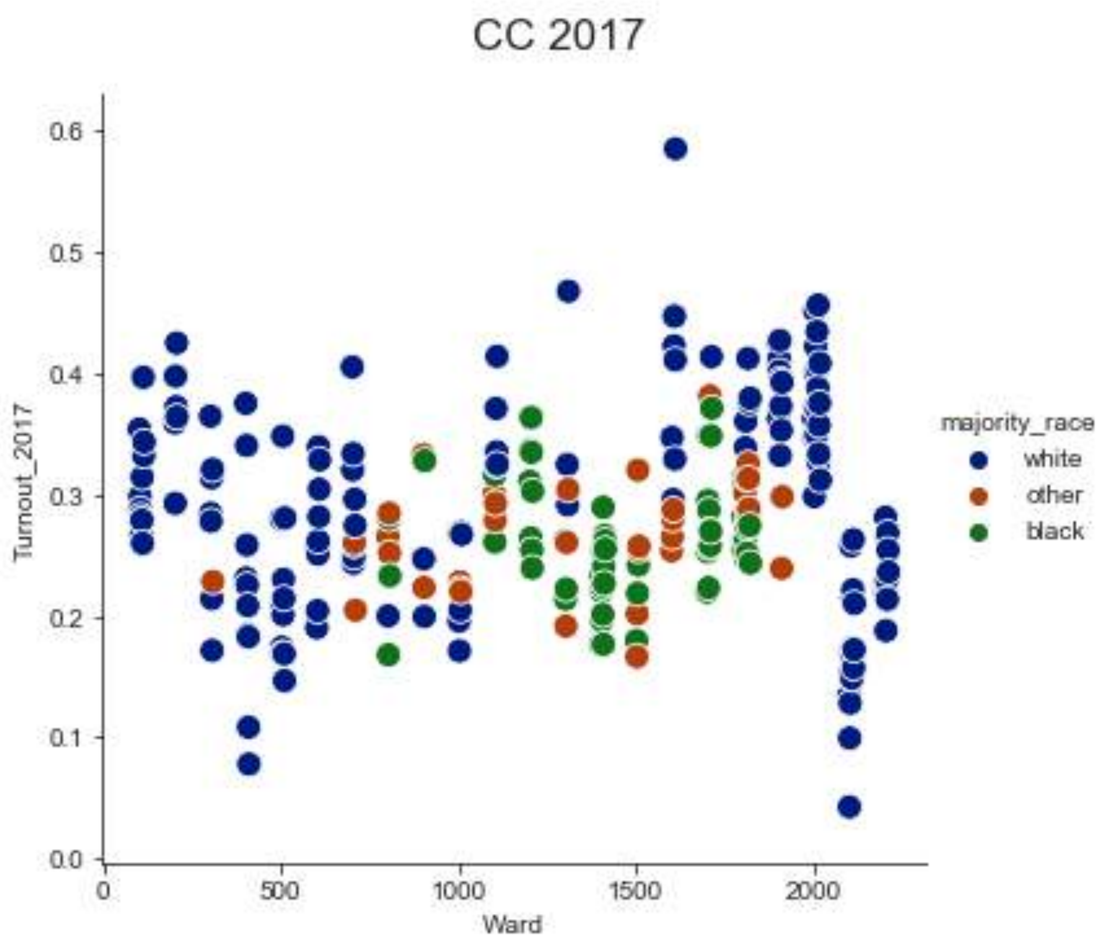

```
[30]: g = sns.relplot(data=df_final, x="Ward", y="Turnout_2015",␣
       ↪hue="majority_race",palette="dark", kind='scatter', s=100)
      g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
      g.fig.suptitle('CC 2015',fontsize=17)
```

```
[30]: Text(0.5, 0.98, 'CC 2015')
```



[31]:
```
g = sns.relplot(data=df_final, x="Ward", y="Turnout_2017",␣
 ↪hue="majority_race",palette="dark", kind='scatter', s=100)
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('CC 2017',fontsize=17)
```

[31]: Text(0.5, 0.98, 'CC 2017')



[ ]:

[7]:
```
df_final_long = pd.melt(df_final, id_vars=['Ward','majority_race'],␣
 ↪value_vars=['Turnout_2011',
                                                                        ␣
 ↪'Turnout_2013',
                                                                        ␣
 ↪'Turnout_2015',
                                                                        ␣
 ↪'Turnout_2017',
                                                                        ␣
 ↪'Turnout_2019'])
df_final_long
```

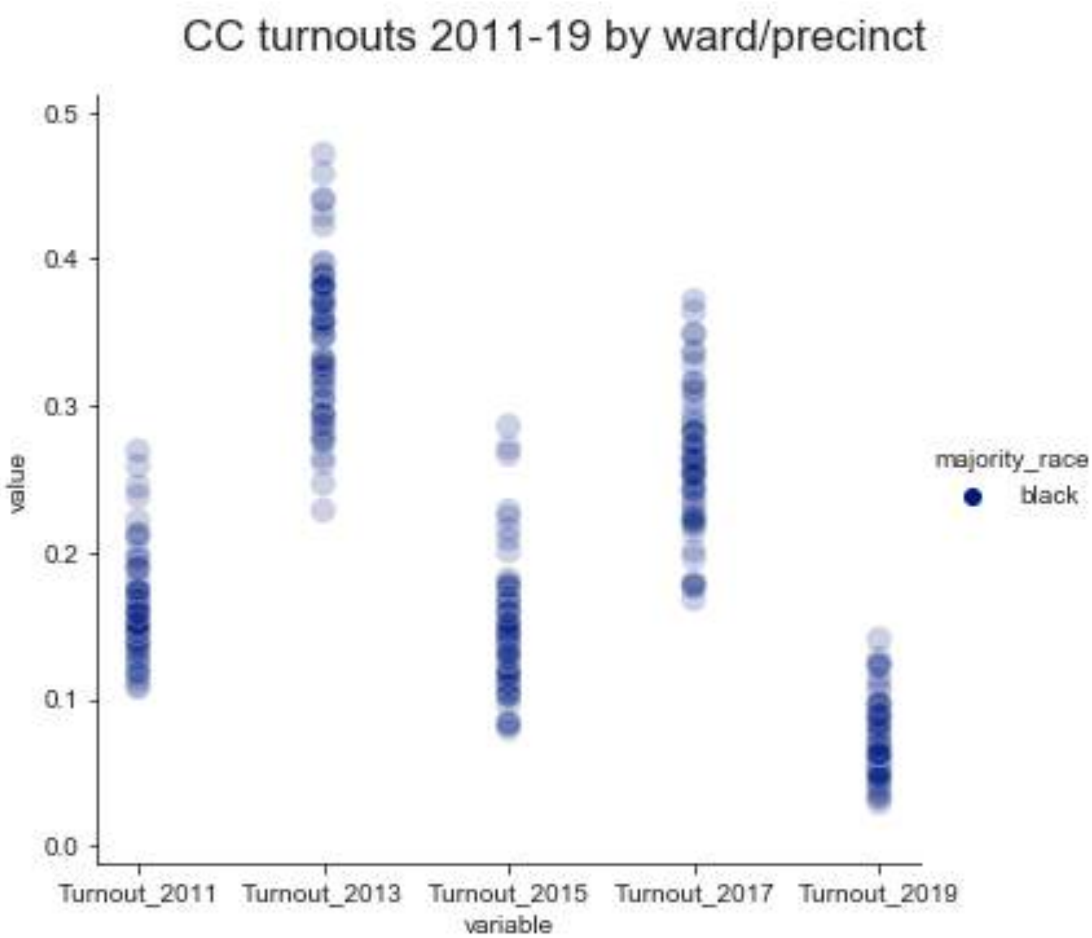
[7]:

	Ward	majority_race	variable	value
0	101	white	Turnout_2011	0.186715
1	102	white	Turnout_2011	0.145342

```
2       103          white  Turnout_2011  0.156431
3       104          white  Turnout_2011  0.175182
4       105          white  Turnout_2011  0.140314
…     …            …        …        …
1255  2208          white  Turnout_2019  0.070218
1256  2209          white  Turnout_2019  0.079782
1257  2210          white  Turnout_2019  0.099509
1258  2211          white  Turnout_2019  0.119392
1259  2212          white  Turnout_2019  0.094168

[1260 rows x 4 columns]
```

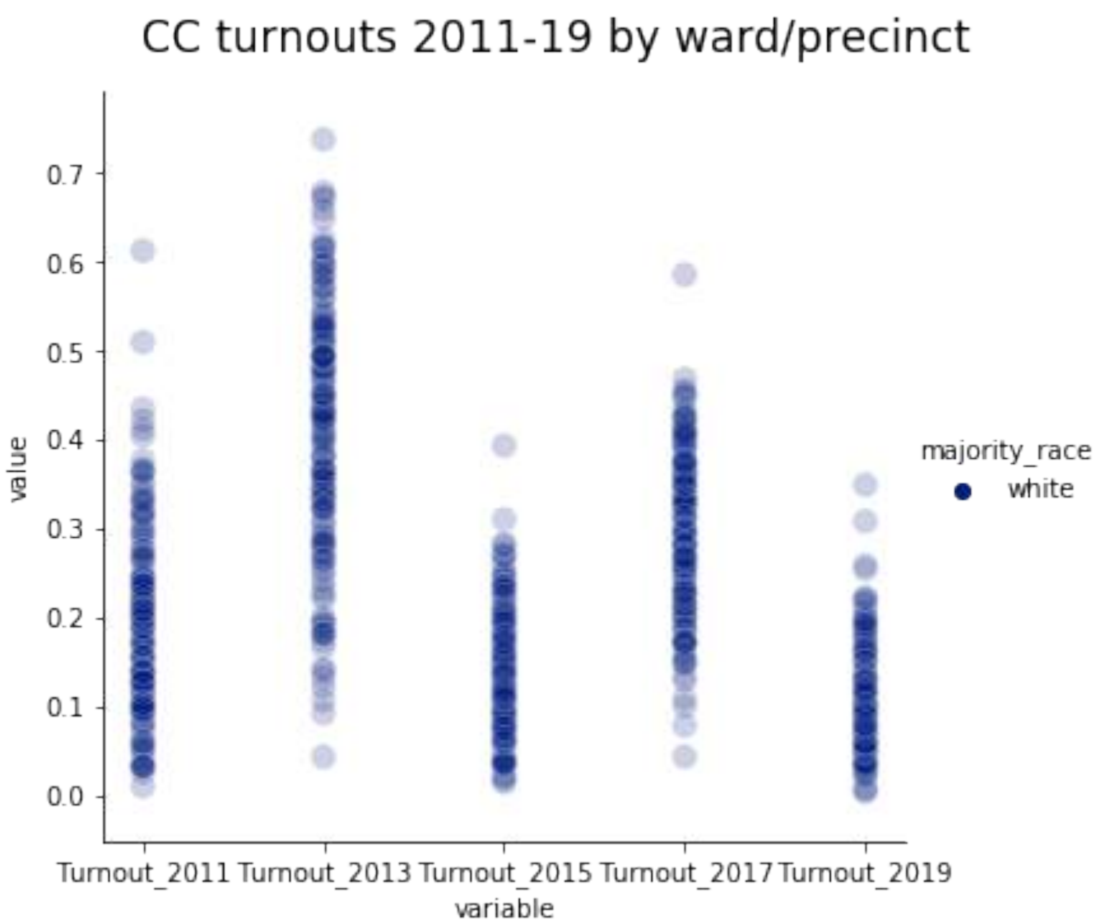
[46]:
```python
g = sns.relplot(data=df_final_long[df_final_long['majority_race']=='black'],␣
↪x="variable", y="value", hue="majority_race",palette="dark", kind='scatter',␣
↪s=100,alpha=.2 )
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('CC turnouts 2011-19 by ward/precinct',fontsize=17)
```

[46]:
```
Text(0.5, 0.98, 'CC turnouts 2011-19 by ward/precinct')
```



[8]:
```
g = sns.relplot(data=df_final_long[df_final_long['majority_race']=='white'],
↪x="variable", y="value", hue="majority_race",palette="dark", kind='scatter',
↪s=100,alpha=.2 )
g.fig.subplots_adjust(top=0.9) # adjust the Figure in g
g.fig.suptitle('CC turnouts 2011-19 by ward/precinct',fontsize=17)
```

[8]:
```
Text(0.5, 0.98, 'CC turnouts 2011-19 by ward/precinct')
```



```
# Coding
import matplotlib.pyplot as plt
import pandas as pd
eleven =
pd.DataFrame(pd.read_excel("2011_CityCouncil_Results_Race_Turnout.xlsx
"))
thirteen =
pd.DataFrame(pd.read_excel("2013_CityCouncil_Race_Turnout_Results.xlsx
"))
fifteen = pd.DataFrame(pd.read_excel("2015_city_council.xlsx"))
seventeen =
pd.DataFrame(pd.read_excel("2017_CityCouncil_AtLarge_Turnout_Race.xlsx
"))
nineteen = pd.DataFrame(pd.read_excel("2019_CityCouncil_Race
Turnout.xlsx"))
Turnout = pd.read_csv("CC_turnout_all_years.csv")

mydict = dict(sorted(zip(eleven['Black
Percentage'],Turnout['Turnout_2011'] )))
#mydict = dict(sorted(mydict.items, key=lambda item: item[1]))
lists = mydict.items()
print(lists)
x, y = zip(*lists)
plt.xlabel("Black Percentage")
plt.ylabel("Turnout In 2011")
plt.plot(x,y)
plt.show()
print(mydict)

plt.figure()
mydict = dict(sorted(zip(eleven['White
Percentage'],Turnout['Turnout_2011'] )))
lists = mydict.items()
x, y = zip(*lists)
plt.xlabel("White Percentage")
plt.ylabel("Turnout In 2011")
plt.plot(x,y)
plt.show()

plt.figure()
mydict = dict(sorted(zip(eleven['Hispanic
Percentage'],Turnout['Turnout_2011'] )))
lists = mydict.items()
x, y = zip(*lists)
plt.xlabel("Hispanic Percentage")
plt.ylabel("Turnout In 2011")
plt.plot(x,y)
plt.show()

# print(Turnout['Turnout_2011'])
# print(eleven['Black Percentage'])

# Coding
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
from pandas import DataFrame
eleven =
pd.DataFrame(pd.read_excel("2011_CityCouncil_Results_Race_Turnout.xlsx
"))
thirteen =
pd.DataFrame(pd.read_excel("2013_CityCouncil_Race_Turnout_Results.xlsx
"))
fifteen = pd.DataFrame(pd.read_excel("2015_city_council.xlsx"))
seventeen =
pd.DataFrame(pd.read_excel("2017_CityCouncil_AtLarge_Turnout_Race.xlsx
"))
nineteen = pd.DataFrame(pd.read_excel("2019_CityCouncil_Race
Turnout.xlsx"))
Turnout = pd.read_csv("CC_turnout_all_years.csv")


plt.figure()
x = eleven['Black Percentage']
y = Turnout['Turnout_2011']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Black
Percentage','Turnout_2011'])
one = data['Black Percentage']
two = data['Turnout_2011']
sns.regplot(x = one,y= two,data = data)
plt.savefig('BlackX_Turnout2011Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = eleven['White Percentage']
y = Turnout['Turnout_2011']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['White
Percentage','Turnout_2011'])
one = data['White Percentage']
two = data['Turnout_2011']
sns.regplot(x = one,y= two,data = data)
plt.savefig('WhiteX_Turnout2011Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = eleven['Hispanic Percentage']
y = Turnout['Turnout_2011']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Hispanic
Percentage','Turnout_2011'])
one = data['Hispanic Percentage']
two = data['Turnout_2011']
sns.regplot(x = one,y= two,data = data)
plt.savefig('HispanicX_Turnout2011Y')
plt.show(block = False)
```

```
plt.pause(2)
plt.close()


plt.figure()
x = eleven['Asian Percentage']
y = Turnout['Turnout_2011']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Asian
Percentage','Turnout_2011'])
one = data['Asian Percentage']
two = data['Turnout_2011']
sns.regplot(x = one,y= two,data = data)
plt.savefig('AsianX_Turnout2011Y')
plt.show(block = False)
plt.pause(2)
plt.close()


########################

plt.figure()
x = thirteen['Black Percentage']
y = Turnout['Turnout_2013']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Black
Percentage','Turnout_2013'])
one = data['Black Percentage']
two = data['Turnout_2013']
sns.regplot(x = one,y= two,data = data)
plt.savefig('BlackX_Turnout2013Y')
plt.show(block = False)
plt.pause(2)
plt.close()



plt.figure()
x = thirteen['White Percentage']
y = Turnout['Turnout_2013']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['White
Percentage','Turnout_2013'])
one = data['White Percentage']
two = data['Turnout_2013']
sns.regplot(x = one,y= two,data = data)
plt.savefig('WhiteX_Turnout2013Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = thirteen['Hispanic Percentage']
y = Turnout['Turnout_2013']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Hispanic
```

```
Percentage','Turnout_2013'])
one = data['Hispanic Percentage']
two = data['Turnout_2013']
sns.regplot(x = one,y= two,data = data)
plt.savefig('HispanicX_Turnout2013Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = thirteen['Asian Percentage']
y = Turnout['Turnout_2013']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Asian
Percentage','Turnout_2013'])
one = data['Asian Percentage']
two = data['Turnout_2013']
sns.regplot(x = one,y= two,data = data)
plt.savefig('AsianX_Turnout2013Y')
plt.show(block = False)
plt.pause(2)
plt.close()

########################

plt.figure()
x = fifteen['Black Percentage']
y = Turnout['Turnout_2015']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Black
Percentage','Turnout_2015'])
one = data['Black Percentage']
two = data['Turnout_2015']
sns.regplot(x = one,y= two,data = data)
plt.savefig('BlackX_Turnout2015Y')
plt.show(block = False)
plt.pause(2)
plt.close()



plt.figure()
x = fifteen['White Percentage']
y = Turnout['Turnout_2015']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['White
Percentage','Turnout_2015'])
one = data['White Percentage']
two = data['Turnout_2015']
sns.regplot(x = one,y= two,data = data)
plt.savefig('WhiteX_Turnout2015Y')
plt.show(block = False)
plt.pause(2)
plt.close()
```

```
plt.figure()
x = fifteen['Hispanic Percentage']
y = Turnout['Turnout_2015']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Hispanic
Percentage','Turnout_2015'])
one = data['Hispanic Percentage']
two = data['Turnout_2015']
sns.regplot(x = one,y= two,data = data)
plt.savefig('HispanicX_Turnout2015Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = fifteen['Asian Percentage']
y = Turnout['Turnout_2015']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Asian
Percentage','Turnout_2015'])
one = data['Asian Percentage']
two = data['Turnout_2015']
sns.regplot(x = one,y= two,data = data)
plt.savefig('AsianX_Turnout2015Y')
plt.show(block = False)
plt.pause(2)
plt.close()



########################

plt.figure()
x = seventeen['Black Percentage']
y = Turnout['Turnout_2017']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Black
Percentage','Turnout_2017'])
one = data['Black Percentage']
two = data['Turnout_2017']
sns.regplot(x = one,y= two,data = data)
plt.savefig('BlackX_Turnout2017Y')
plt.show(block = False)
plt.pause(2)
plt.close()



plt.figure()
x = seventeen['White Percentage']
y = Turnout['Turnout_2017']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['White
Percentage','Turnout_2017'])
one = data['White Percentage']
two = data['Turnout_2017']
```

```
sns.regplot(x = one,y= two,data = data)
plt.savefig('WhiteX_Turnout2017Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = seventeen['Hispanic Percentage']
y = Turnout['Turnout_2017']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Hispanic
Percentage','Turnout_2017'])
one = data['Hispanic Percentage']
two = data['Turnout_2017']
sns.regplot(x = one,y= two,data = data)
plt.savefig('HispanicX_Turnout2017Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = seventeen['Asian Percentage']
y = Turnout['Turnout_2017']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Asian
Percentage','Turnout_2017'])
one = data['Asian Percentage']
two = data['Turnout_2017']
sns.regplot(x = one,y= two,data = data)
plt.savefig('AsianX_Turnout2017Y')
plt.show(block = False)
plt.pause(2)
plt.close()

########################

plt.figure()
x = nineteen['Black Percentage']
y = Turnout['Turnout_2019']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Black
Percentage','Turnout_2019'])
one = data['Black Percentage']
two = data['Turnout_2019']
sns.regplot(x = one,y= two,data = data)
plt.savefig('BlackX_Turnout2019Y')
plt.show(block = False)
plt.pause(2)
plt.close()



plt.figure()
x = nineteen['White Percentage']
y = Turnout['Turnout_2019']
```

```
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['White
Percentage','Turnout_2019'])
one = data['White Percentage']
two = data['Turnout_2019']
sns.regplot(x = one,y= two,data = data)
plt.savefig('WhiteX_Turnout2019Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = nineteen['Hispanic Percentage']
y = Turnout['Turnout_2019']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Hispanic
Percentage','Turnout_2019'])
one = data['Hispanic Percentage']
two = data['Turnout_2019']
sns.regplot(x = one,y= two,data = data)
plt.savefig('HispanicX_Turnout2019Y')
plt.show(block = False)
plt.pause(2)
plt.close()


plt.figure()
x = nineteen['Asian Percentage']
y = Turnout['Turnout_2019']
mydict = dict(sorted(zip(x,y )))
data = pd.DataFrame(list(mydict.items()),columns = ['Asian
Percentage','Turnout_2019'])
one = data['Asian Percentage']
two = data['Turnout_2019']
sns.regplot(x = one,y= two,data = data)
plt.savefig('AsianX_Turnout2019Y')
plt.show(block = False)
plt.pause(2)
plt.close()

# For reading file name
import glob
print(glob.glob('.'))
myList=[]
myListWithoutSlash=[]
for file_name in glob.iglob('./Imag/*.*', recursive=True):
  print(file_name)
  myList.append(file_name)
  # using split()
  # Get String after substring occurrence
  res = file_name.split('/')
  for item in res:
      if "jpeg" in item:
        myListWithoutSlash.append(item)
      if "JPEG" in item:
        myListWithoutSlash.append(item)
```

```
      if "png" in item:
        myListWithoutSlash.append(item)
      if "PNG" in item:
        myListWithoutSlash.append(item)
      if "jpg" in item:
        myListWithoutSlash.append(item)
      if "JPG" in item:
        myListWithoutSlash.append(item)
# for python-docx package
# create words documents
from docx import Document
from docx.shared import Cm
document = Document()
table = document.add_table(rows = 1, cols = 2)
table.style = 'Table Grid'
hdr_cells = table.rows[0].cells
hdr_cells[0].text = 'ImageName'
hdr_cells[1].text = 'Image'
for i in range(len(myList)):
  print(i)
  print(myListWithoutSlash[i])
  row_cells = table.add_row().cells
  row_cells[0].text = myListWithoutSlash[i]
  p = row_cells[1].add_paragraph()
  r = p.add_run()
  r.add_picture(myList[i],width=Cm(4.0), height=Cm(4))
document.save('./demo.docx')


# # For reading file name
# import glob
# print(glob.glob('.'))
# myList=[]
# myListWithoutSlash=[]
# for file_name in glob.iglob('./*.*', recursive=True):
#   print(file_name)
#   myList.append(file_name)
#   # using split()
#   # Get String after substring occurrence
#   res = file_name.split('/')
#   for item in res:
#       if "jpeg" in item:
#         myListWithoutSlash.append(item)
#       if "JPEG" in item:
#         myListWithoutSlash.append(item)
#       if "png" in item:
#         myListWithoutSlash.append(item)
#       if "PNG" in item:
#         myListWithoutSlash.append(item)
#       if "jpg" in item:
#         myListWithoutSlash.append(item)
#       if "JPG" in item:
#         myListWithoutSlash.append(item)
# # for python-docx package
# # create words documents
# from docx import Document
# from docx.shared import Cm
```

```
# document = Document()
# table = document.add_table(rows = 1, cols = 4)
# table.style = 'Table Grid'
# hdr_cells = table.rows[0].cells
# hdr_cells[0].text = 'ProductName'
# hdr_cells[1].text = 'Id'
# hdr_cells[2].text = 'ImageName'
# hdr_cells[3].text = 'Image'
# for i in range(len(myList)):
#   print(i)
#   print(myListWithoutSlash[i])
#   row_cells = table.add_row().cells
#   row_cells[2].text = myListWithoutSlash[i]
#   p = row_cells[3].add_paragraph()
#   r = p.add_run()
#   r.add_picture(myList[i],width=Cm(4.0), height=Cm(4))
# document.save('./demo.docx')
```

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [11]:

```
import geopandas as gpd
import shapely
from geopandas.tools import sjoin
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/geopandas/_compat.py:110: UserWarning: The Shapely GEOS versio
n (3.8.0-CAPI-1.13.1 ) is incompatible with the GEOS version PyGEOS wa
s compiled with (3.9.0-CAPI-1.16.2). Conversions between both will be
slow.
  shapely_geos_version, geos_capi_version_string
```

In [7]:

```
#import raw csv
raw_df = pd.read_csv("data/household_income_2018.csv", header=1)
```

In [17]:

```
raw_df = raw_df.set_index('id')
```

In [12]:

```
raw_gdf = gpd.read_file('data/shape.shp')
```

In [16]:

```
raw_gdf = raw_gdf.set_index('AFFGEOID')
```

In [20]:

```
raw_df.shape
```

Out[20]:

```
(204, 553)
```

In [21]:

```
raw_gdf.shape
```

Out[21]:

```
(1475, 9)
```

In [173]:

```
household_income_units = raw_df.filter(like='Occupied housing units!!Occupied housir
```

In [174]:

```
household_income_units.shape
```

Out[174]:

```
(204, 12)
```

In [175]:

```
household_income_units.head()
```

Out[175]:

	Estimate!!Occupied housing units!!Occupied housing units!!HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2018 INFLATION-ADJUSTED DOLLARS)!!Less than $5,000	Estimate!!Occupied housing units!!Occupied housing units!!HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2018 INFLATION-ADJUSTED DOLLARS)!!5, 000*to*9,999	Estimate!!Occupied housing units!!Occupied housing units!!HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2018 INFLATION-ADJUSTED DOLLARS)!!10, 000*to*14,999
id			
1400000US25025000100	36	8	91
1400000US25025000201	24	41	0
1400000US25025000202	29	53	71
1400000US25025000301	0	11	53
1400000US25025000302	36	57	26

In [176]:

```
def clean_title(s):
    while('!!' in s):
        index = s.index('!!')
        s = s[index+2:]
    if('to' in s):
        index = s.index('to')
        s = s[:index] + ' to' + s[index+2:]
    return s.strip()
```

In [177]:

```
clean_title('Estimate!!Occupied housing units!!Occupied housing units!!HOUSEHOLD INC
```

Out[177]:

```
'5,000 to 9,999'
```

In [178]:

```
income_labels = ['<5000', '5k~10k', '10k~15k', '15k~20k', '20k~25k', '25k~35k', '35k
```

In [181]:

```
household_income_units = household_income_units.set_axis(income_labels, axis=1)
```

In [186]:

```
household_income_total = household_income_units.loc[household_income_units['medHHInc
```

In [182]:

```
household_income_gdf = pd.merge(raw_gdf, household_income_units, left_index=True, ri
```

In [187]:

```
household_income_total = pd.merge(raw_gdf, household_income_total, left_index=True,
```

In [188]:

```
household_income_total.head()
```

Out[188]:

	STATEFP	COUNTYFP	TRACTCE	GEOID	NAME	LSAD	ALAND
1400000US25025050500	25	025	050500	25025050500	505	CT	111868
1400000US25025060101	25	025	060101	25025060101	601.01	CT	295623
1400000US25025060400	25	025	060400	25025060400	604	CT	399279
1400000US25025061101	25	025	061101	25025061101	611.01	CT	165282
1400000US25025061200	25	025	061200	25025061200	612	CT	1835771

In [148]:

```
household_income_gdf.to_file('data/household_income_gdf.shp')
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/ipykernel_launcher.py:1: UserWarning: Column names longer than
10 characters will be truncated when saved to ESRI Shapefile.
  """Entry point for launching an IPython kernel.
```

In [189]:

```
household_income_total.to_file('data/house_income_total.shp')
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/ipykernel_launcher.py:1: UserWarning: Column names longer than
10 characters will be truncated when saved to ESRI Shapefile.
  """Entry point for launching an IPython kernel.
```

In [30]:

```
clean_gdf = pd.merge(raw_gdf, raw_df, left_index=True, right_index=True, how="inner"
```

In [33]:

```
clean_gdf.shape
```

Out[33]:

```
(204, 562)
```

In [192]:

```
raw_education_df = pd.read_csv("data/educational_attainment_2018.csv", header=1)
```

In [238]:

```
raw_education_df = raw_education_df.set_index('id')
```

In [239]:

```
raw_education_df.head()
```

Out[239]:

	Geographic Area Name	Estimate!!Total!!Population 18 to 24 years	Margin of Error!!Total MOE!!Population 18 to 24 years	Estimate!!Tota 18 to 24 yea high sch
id				
1400000US25025000100	Census Tract 1, Suffolk County, Massachusetts	570	120	
1400000US25025000201	Census Tract 2.01, Suffolk County, Massachusetts	516	159	
1400000US25025000202	Census Tract 2.02, Suffolk County, Massachusetts	494	165	
1400000US25025000301	Census Tract 3.01, Suffolk County, Massachusetts	437	148	
1400000US25025000302	Census Tract 3.02, Suffolk County, Massachusetts	353	119	

5 rows × 769 columns

In [252]:

```
educational_attainment_units = raw_education_df.filter(like='Estimate!!Total!!P', ax
```

In [253]:

```
educational_attainment_units.head()
```

Out[253]:

id	Estimate!!Total!!Population 18 to 24 years	Estimate!!Total!!Population 18 to 24 years!!Less than high school graduate	Estimate!!Total!!Popu 18 to 24 years school graduate (in equiva
1400000US25025000100	570	82	
1400000US25025000201	516	9	
1400000US25025000202	494	9	
1400000US25025000301	437	50	
1400000US25025000302	353	24	

In [254]:

```
drop_labels = ['Estimate!!Total!!Population 18 to 24 years', 'Estimate!!Total!!Popul
```

In [255]:

```
educational_attainment_clean = educational_attainment_units.drop(columns=drop_labels
```

In [258]:

```
print(educational_attainment_clean.shape)
educational_attainment_clean.head()
```

```
(204, 11)
```

Out[258]:

id	Estimate!!Total!!Population 18 to 24 years!!Less than high school graduate	Estimate!!Total!!Population 18 to 24 years!!High school graduate (includes equivalency)	Estimate!!Total!!Popu 18 to 24 years! college or asso (
1400000US25025000100	82	92	
1400000US25025000201	9	23	
1400000US25025000202	9	60	
1400000US25025000301	50	36	
1400000US25025000302	24	1	

In [274]:

```
under_hs = pd.concat([educational_attainment_clean.iloc[:, 0], educational_attainmer
under_hs.name = 'NoDegree'
under_hs
```

Out[274]:

```
id
1400000US25025000100    589
1400000US25025000201    161
1400000US25025000202    379
1400000US25025000301    255
1400000US25025000302    241
                       ...
1400000US25025981502      0
1400000US25025981600      0
1400000US25025981700      0
1400000US25025981800      0
1400000US25025990101      0
Name: NoDegree, Length: 204, dtype: int64
```

In [275]:

```
hs = pd.concat([educational_attainment_clean.iloc[:, 2:4], educational_attainment_cl
hs.name = 'HSDegree'
hs
```

Out[275]:

```
id
1400000US25025000100    1717
1400000US25025000201    1148
1400000US25025000202    1218
1400000US25025000301     970
1400000US25025000302     995
                        ...
1400000US25025981502       0
1400000US25025981600       0
1400000US25025981700       0
1400000US25025981800       0
1400000US25025990101      49
Name: HSDegree, Length: 204, dtype: int64
```

In [276]:

```
college = pd.concat([educational_attainment_clean.iloc[:, 3], educational_attainment
college.name = 'CollegeDegree'
college
```

Out[276]:

```
id
1400000US25025000100    2406
1400000US25025000201    2632
1400000US25025000202    1954
1400000US25025000301    1635
1400000US25025000302    1900
                        ...
1400000US25025981502       0
1400000US25025981600       0
1400000US25025981700       0
1400000US25025981800      21
1400000US25025990101       0
Name: CollegeDegree, Length: 204, dtype: int64
```

In [277]:

```
educational_attain_pdf = pd.concat([under_hs, hs, college], axis=1)
```

In [278]:

```
educational_attain_pdf.head()
```

Out[278]:

	NoDegree	HSDegree	CollegeDegree
id			
1400000US25025000100	589	1717	2406
1400000US25025000201	161	1148	2632
1400000US25025000202	379	1218	1954
1400000US25025000301	255	970	1635
1400000US25025000302	241	995	1900

In [282]:

```
educational_attain_pdf['Degree'] = educational_attain_pdf.idxmax(axis=1)
```

In [283]:

```
educational_attain_gdf = pd.merge(raw_gdf, educational_attain_pdf, left_index=True,
```

In [284]:

```
educational_attain_gdf.head()
```

Out[284]:

	STATEFP	COUNTYFP	TRACTCE	GEOID	NAME	LSAD	ALAND
1400000US25025050500	25	025	050500	25025050500	505	CT	111868
1400000US25025060101	25	025	060101	25025060101	601.01	CT	295623
1400000US25025060400	25	025	060400	25025060400	604	CT	399279
1400000US25025061101	25	025	061101	25025061101	611.01	CT	165282
1400000US25025061200	25	025	061200	25025061200	612	CT	1835771

In [286]:

```
educational_attain_gdf.to_file('data/educational_attain_gdf.shp')
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-p
ackages/ipykernel_launcher.py:1: UserWarning: Column names longer than
10 characters will be truncated when saved to ESRI Shapefile.
  """Entry point for launching an IPython kernel.
```

In [ ]:

```

```

add_POC_as_race

April 29, 2021

[1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# processing data
def clean_data(filename):
    df = pd.read_excel(filename,index=False)
    df.replace(r'', np.nan)
    cleaned_df = df[~pd.isnull(df).any(axis = 1)]
    return cleaned_df

def Dorcester_precincts(df):
    columns = df.columns.to_list()
    index = columns.index("Ward")

    data_array = df.to_numpy()

    d_precs_int = [802, 806, 708, 707, 710, 709, 710, 1305, 1306, 1307, 1308,␣
 ↪1309, 1303, 1310, 1501, 1503, 1504, 1506, 1507, 1508, 1509, 1602, 1604,␣
 ↪1605, 1606, 1607, 1608, 1609, 1610, 1611, 1612, 1713, 115]
    d_precs = [str(i) for i in d_precs_int]
    rows = []
    for i in range(np.shape(data_array)[0]):
        current_p = str(int(data_array[i, index]))
        if current_p in d_precs:
            rows.append(data_array[i].tolist())

    Dorcester_df = pd.DataFrame(np.asarray(rows), columns = df.columns)
    Dorcester_df["Ward"] = Dorcester_df["Ward"].astype('float')
    Dorcester_df["Ward"] = Dorcester_df["Ward"].astype('Int64')
    return Dorcester_df

CC_all_yrs = pd.read_excel("df_final.xlsx")

Dorcester_precincts(CC_all_yrs)
```

```
[1]:    Unnamed: 0  Ward majority_race Turnout_2011 Turnout_2013 Turnout_2015  \
    0           65   707           POC         0.21         0.36         0.15
    1           66   708         white         0.22         0.44         0.17
    2           67   709         white         0.19          0.4         0.15
    3           68   710           POC         0.11         0.31         0.08
    4           70   802         white         0.13         0.27         0.08
    5           74   806           POC         0.13         0.35         0.08
    6          111  1303           POC          0.1         0.23         0.09
    7          113  1305           POC         0.17         0.38          0.1
    8          114  1306           POC         0.22          0.4         0.14
    9          115  1307         white         0.32         0.43         0.18
    10         116  1308         white         0.25         0.42         0.14
    11         117  1309           POC         0.24         0.45         0.13
    12         118  1310         white         0.43         0.59         0.27
    13         133  1501         black         0.15         0.28         0.08
    14         135  1503           POC         0.15         0.31         0.06
    15         136  1504           POC         0.17         0.33         0.09
    16         138  1506           POC         0.29         0.43         0.19
    17         139  1507         black         0.15         0.32          0.1
    18         140  1508           POC         0.19         0.33         0.11
    19         141  1509           POC         0.22         0.38         0.13
    20         143  1602         white         0.28          0.5         0.19
    21         145  1604           POC         0.22          0.4         0.15
    22         146  1605         white         0.29         0.47         0.15
    23         147  1606           POC         0.22         0.39         0.13
    24         148  1607         white         0.42         0.62         0.24
    25         149  1608         white          0.4         0.57         0.25
    26         150  1609         white         0.51         0.67         0.28
    27         151  1610         white         0.38         0.54         0.18
    28         152  1611         white         0.41         0.59         0.21
    29         153  1612         white         0.61         0.74         0.39
    30         166  1713         white         0.33         0.48         0.19

        Turnout_2017 Turnout_2019
    0           0.26         0.05
    1            0.3         0.06
    2           0.27         0.05
    3            0.2         0.04
    4            0.2         0.05
    5           0.25         0.04
    6           0.19         0.02
    7           0.26         0.05
    8            0.3         0.07
    9           0.33         0.06
    10          0.29         0.07
    11           0.3         0.07
    12          0.47         0.12
```

13	0.18	0.03
14	0.17	0.03
15	0.2	0.03
16	0.32	0.08
17	0.24	0.05
18	0.25	0.05
19	0.26	0.07
20	0.35	0.07
21	0.28	0.05
22	0.3	0.11
23	0.29	0.07
24	0.45	0.1
25	0.42	0.12
26	0.45	0.07
27	0.33	0.05
28	0.41	0.1
29	0.59	0.09
30	0.41	0.11

[2]:
```
CC_all_yrs_D3 = Dorcester_precincts(CC_all_yrs)
```

[3]:
```
CC_all_yrs_D3.to_excel('CC_all_yrs_D3.xlsx', index=False)
```

[ ]:
```
```

[ ]:
```
df_final_long = pd.melt(df_final, id_vars=['Ward','majority_race'],␣
 ↪value_vars=['Turnout_2011',
                                                                              ␣
 ↪'Turnout_2013',
                                                                              ␣
 ↪'Turnout_2015',
                                                                              ␣
 ↪'Turnout_2017',
                                                                              ␣
 ↪'Turnout_2019'])
```

[ ]:
```
```

DemographicChanges

April 29, 2021

1 Changing Demographics and Black Candidates

[ ]:
```
import pandas as pd
import numpy as np

pop_df = pd.read_csv("RacebyYearBoston.csv")
```

[ ]:
```
pop_df
```

[ ]:
```
pop_df_transposed = pop_df.T
```

[ ]:
```
pop_df_transposed = pop_df_transposed.reset_index()
new_header = pop_df_transposed.iloc[0]
pop_df_transposed = pop_df_transposed[1:]
pop_df_transposed.columns = new_header
pop_df_transposed = pop_df_transposed.rename(columns={"Race":"year"})
```

[ ]:
```
pop_df_transposed['year'] = pop_df_transposed['year'].str[-4:]
pop_df_transposed = pop_df_transposed.apply(pd.to_numeric)
pop_df_transposed
```

[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.barplot(x="year", y="black", data=pop_df_transposed, palette="Blues_d")
plt.xlabel("Year")
plt.ylabel("Black Population")
plt.title("Black Population in Boston (2010-2019)")
```

[ ]:
```
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.barplot(x="year", y="asian", data=pop_df_transposed, palette="Reds_d")
plt.xlabel("Year")
plt.ylabel("Asian Population")
plt.title("Asian Population in Boston (2010-2019)")
```

[ ]:
```

```

```
ax = pop_df[["Race", "Year2019", "Year2015", "Year2010"]].iloc[[1, 2, 4]].plot.
↪bar(x='Race', rot=0, figsize = (20,10), title = "Demographic Change by Year␣
↪(Boston)")
```

```
[ ]: pop_df_transposed.plot.line(x='year', y ='black', figsize = (20,10), title =␣
↪"Change in Black Population by Year")
```

```
[ ]: pop_df_transposed= pop_df_transposed.apply(pd.to_numeric)
```

```
[ ]: import seaborn as sns; sns.set_theme(color_codes=True)
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.regplot(data=pop_df_transposed, x="year", y="black")
plt.xlabel("Year")
plt.ylabel("Black Population")
plt.title("Black Population in Boston (2010-2019)")
plt.show()
```

```
[ ]: import seaborn as sns; sns.set_theme(color_codes=True)
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.regplot(data=pop_df_transposed, x="year", y="asian")
plt.xlabel("Year")
plt.ylabel("Asian Population")
plt.title("Asian Population in Boston (2010-2019)")
plt.show()
```

1.1 A closer look at key races with Black Candidates

```
[ ]: import pandas as pd

d = {'Year': [2011, 2013, 2015, 2017, 2017, 2018, 2019], 'Type': ["City␣
↪Council", "City Council", "City Council", "City Council", "Mayoral", "US␣
↪House Democratic Primary", "DA"], 'Candidate': ['Pressley', 'Pressley',␣
↪'Pressley', 'Pressley', 'Jackson', 'Pressley', 'Rollins'], 'Percentage':␣
↪[round(37532/63009*100,2), round((60702/141782)*100, 2), round((31747/
↪50786)*100, 2), round((57435/108879)*100, 2), 33.97, round((40615/
↪66456)*100, 2), 72.60]}
df = pd.DataFrame(data=d)
df
```

```
[ ]: import matplotlib.pyplot as plt
import seaborn as sns; sns.set_theme(color_codes=True)
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.barplot(x="Year", y="Percentage", data=df, hue = "Type", palette =␣
↪"flare")
plt.xlabel("Year")
plt.ylabel("Percentage of Total Votes Cast")
```

```
plt.title("Outcomes of Key Elections with Black Candidates")
```

[ ]:
```
fig, ax = plt.subplots(figsize=(14, 6))
ax = sns.barplot(x="Year", y="Percentage", data=df, hue = "Candidate", palette␣
 ↪= "viridis")
plt.xlabel("Year")
plt.ylabel("Percentage of Total Votes Cast")
plt.title("Outcomes of Key Elections with Black Candidates")
```

[ ]:
```
fig, ax = plt.subplots(figsize=(14, 7))
ax = sns.scatterplot(data=df, x="Year", y="Percentage", hue = "Type")
ax = sns.regplot(data=df, x="Year", y="Percentage", scatter=False)
plt.xlabel("Year")
plt.ylabel("Percentage of Total Votes Cast")
plt.title("Outcomes of Key Elections with Black Candidates")
```

[ ]:
```
# finding line of best fit
def best_fit(X, Y):

    xbar = sum(X)/len(X)
    ybar = sum(Y)/len(Y)
    n = len(X) # or len(Y)

    numer = sum([xi*yi for xi,yi in zip(X, Y)]) - n * xbar * ybar
    denum = sum([xi**2 for xi in X]) - n * xbar**2

    b = numer / denum
    a = ybar - b * xbar

    print('best fit line:\ny = {:.2f} + {:.2f}x'.format(a, b))

    return a, b

best_fit(df["Year"].to_list(), df["Percentage"].to_list())
```

[ ]: