CS504 Final Report

Background and general description of the project:

Our BU Spark! partner is Mass Alliance, which is a coalition for political and advocacy organizations that work together to build a politically aware Massachusetts. Our partner wants us to create visualized maps which shows the level of progressiveness of different Senate and House voting districts. Although the Senate and the House used different ways to draw the borders and set up their voting districts, we find that they are all made up by a smaller unit, which is called precinct. If we can calculate a political score for each precinct, then we can simply combine them together into different Senate and House districts and visualize them.

Therefore, our mission is to collect and organize voter data on Massachusetts precincts, score them based on the past election results, combine the results of precincts into House and Senate Districts, and map districts to creatively present data for this organization.

This project has four steps:

1. Scrap all candidates of President, Governor, Lieutenant Governor, Senate, Treasurer, and Auditor elections from 2002 – 2018 and score them on the level of progressiveness.
2. Calculate the score of each Mass precinct for each position and for each year based on their voting cast to each candidate.
3. Calculate the average score of each precinct for each position from 2002-2018, and calculate a final score for each precinct by using the weighted average of different positions. (President 25%, Governor 25%, Lieutenant Governor 10%, Senate 20%, Treasurer 10%, and Auditor 10%)
4. Visualize the results and generate political preference map by combining precinct into House and Senate voting districts.
5. Further, improve the result by using other ways to find regions with a high level of progressiveness.

Data sets we use:

- Election dataset from PD43 http://electionstats.state.ma.us/
- Uber dataset from Uber Movement https://movement.uber.com/cities?lang=en-US
- CDC health dataset https://chronicdata.cdc.gov/500-Cities/500-Cities-Local-Data-for-Better-Health-2018-relea/6vp6-wxuq
- Unemployment dataset http://datamechanics.io/data/ZHU_town_unemployment.json

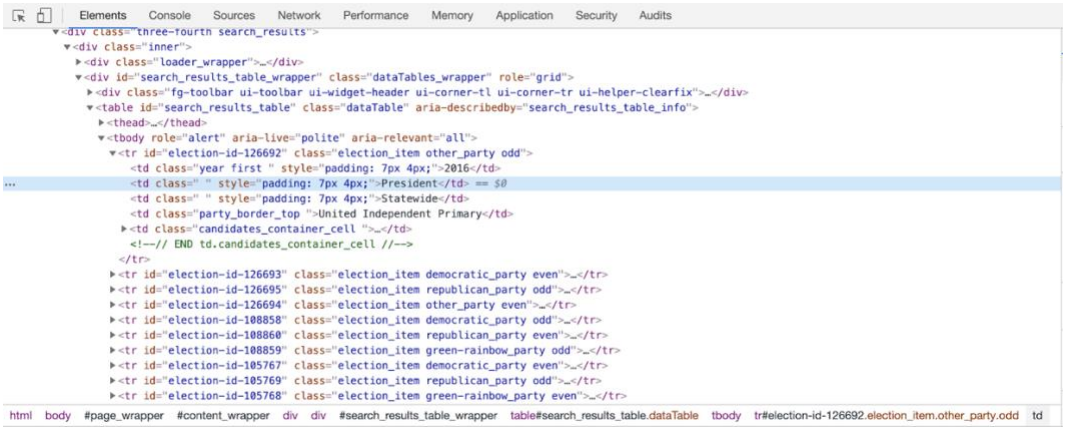
Tools we use:

Python, Pandas, Requests, Beautiful Soup, MongoDB, ECharts, Leafletjs.

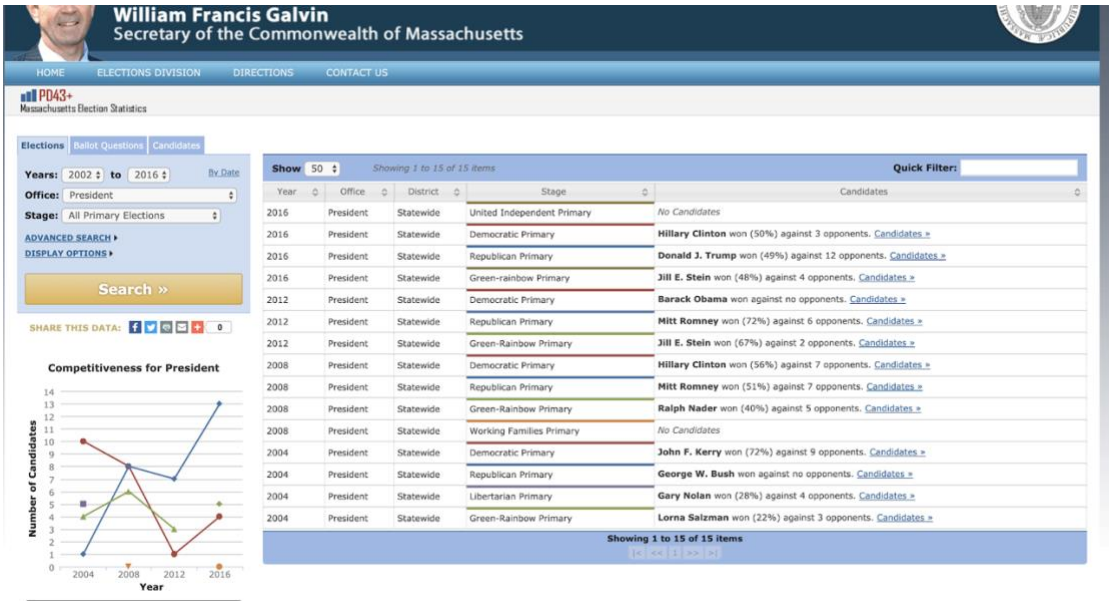
Process:
1.Retrieve data

1.1 Election dataset

Our main dataset is election dataset, which is online at PD43, so we define a web crawler to request all the information from PD43 by mainly using Pandas, Requests and Beautiful Soup. In request_web_information.ipynb, we firstly define 2 lists, one is for the stage of the election, the other is for office id, each office id represents a position. By iterating over the 2 lists, we can get all the url that we need to request. We use the Requests library to request url. Once we successfully request the url, we use Beautiful Soup library to parse response content. By using Google developer tools, we can have a clear understanding of the structure of the web.



In fact, the response content has been converted to a tree structure by Beautiful Soup. Our goal is to find useful information under the different tag. By using .find_all() provided by Beautiful Soup, we can easily get access to the content of different tag at a different level.



Since the information is displayed in table form on the website, it's convenient for us to use Pandas DataFrame to store the data. Finally, we can convert data to .csv file for further use.

1.2 Uber dataset and CDC health dataset

Uber dataset and CDC health dataset can be downloaded directly from their website. They provide .csv file directly.

2.Data processing

2.1 Calculate progressive to conservative score for different districts

In the previous part, we get all the election information that we need. We send them to our partner and they give a score range from 1 to 5 to each candidate.1 represents conservative and 5 represents progressive. Our goal is to calculate the score for each district.

President Election					
Year	Name	Party	Type of Election	Elected	Level of Progressive/Conservative
2004	Lorna Salzman	Green-Rainbow	Primary	Yes	5
2004	David Cobb	Green-Rainbow	Primary		5
2004	Paul Glover	Green-Rainbow	Primary		5
2004	Kent Mesplay	Green-Rainbow	Primary		5
2004	Gary Nolan	Libertarian	Primary	Yes	3
2004	Aaron Russo	Libertarian	Primary		3
2004	Michael Badnarik	Libertarian	Primary		3
2004	Jeffrey Diket	Libertarian	Primary		3
2004	Ruben Perez	Libertarian	Primary		3
2004	George W. Bush	Republican	Primary	Yes	1
2004	John F. Kerry	Democratic	Primary	Yes	3
2004	John Edwards	Democratic	Primary		4
2004	Dennis J. Kucinich	Democratic	Primary		5
2004	Howard Dean	Democratic	Primary		4
2004	Al Sharpton	Democratic	Primary		4
2004	Joseph Lieberman	Democratic	Primary		2
2004	Wesley K. Clark	Democratic	Primary		4
2004	Richard E. Gephardt	Democratic	Primary		4
2004	Carol Moseley Braun	Democratic	Primary		5

City/Town	Ward	Pct	A. Joseph DeNucci	John James Xenakis	Kamal Jain	All Others	Blanks	Total Votes C
Abington	-	1	1,005	166	76	2	186	1,435
Abington	-	2	919	149	73	0	195	1,336
Abington	-	3	1,050	186	66	1	205	1,508
Abington	-	4	1,112	182	74	1	221	1,590
Acton	-	1	768	157	153	3	191	1,272
Acton	-	2	890	186	157	4	253	1,490
Acton	-	3	1,082	185	208	0	273	1,748
Acton	-	4	906	181	140	1	289	1,517
Acton	-	5	965	149	141	4	236	1,495
Acton	-	6	903	158	142	0	224	1,427
Acushnet	-	1	833	141	50	0	129	1,153
Acushnet	-	2	908	173	71	0	178	1,330
Acushnet	-	3	907	158	54	0	153	1,272
Adams	-	1	364	35	14	1	43	457
Adams	-	2	416	50	16	0	48	530
Adams	-	3	446	53	21	1	49	570
Adams	-	4	391	53	16	0	45	505
Adams	-	5	474	61	17	0	67	619
Agawam	-	1	646	157	49	0	181	1,033
Agawam	-	2	775	201	76	0	273	1,325
Agawam	-	3	737	165	71	0	230	1,203
Agawam	-	4	725	184	54	0	223	1,186
Agawam	-	5	865	226	74	0	265	1,430
Agawam	-	6	693	226	62	0	220	1,201
Agawam	-	7	647	195	80	0	235	1,157
Agawam	-	8	745	229	72	0	215	1,261
Alford	-	1	144	28	9	0	37	218

In this part, we mainly use 2 datasets. One tells us the progressive score for each candidate in different stages and different years, the other tells us the district vote number for each candidate in different years, stages and positions. For simplicity, we only consider the vote for democratic and republican since these two parties get the vast majority of votes. And here is our formula to calculate the district's score:
(Democratic vote number * Democratic candidate score + Republican vote number * Republican candidate score) / (Democratic vote number + Republican vote number)
In fact, this is a weighted average. In calculate_score.ipynb, first of all, we read all .csv file into DataFrame. Then we use for loop to iterate over each district DataFrame to get the candidate's name, which is the column name of the DataFrame. Next, we search the name in the candidate score DataFrame to get these candidates' score. Finally, we can use .apply() to calculate the score.

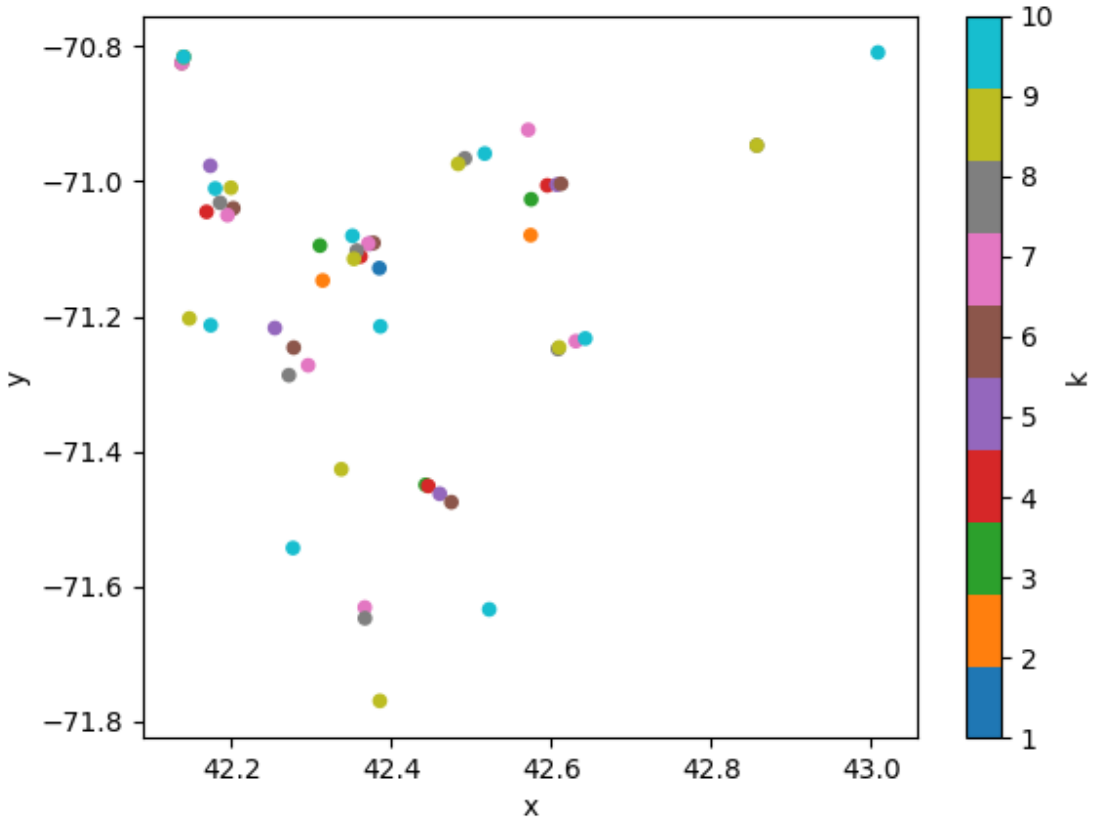
2.2 Load Uber data and DCD data into MongoDB and implement transformations

In health_uber_input.py, we load Uber dataset and DCD dataset into MongoDB for further use. The Uber dataset is available online at Github. The DCD dataset is available online at datamechanics.io. By requesting the corresponding url, we can the data in JSON format and then insert them into MongoDB's different collection (chuci_yfch_yuwan_zhurh.uber and chuci_yfch_yuwan_zhurh.health) for further use.
In health_uber_output.py, we implement some transformations over Uber dataset and DCD dataset to satisfy Project 1's requirement. we firstly aggregate chuci_yfch_yuwan_zhurh.health, use CityName as the key and find each CityName's max population, secondly we aggregate chuci_yfch_yuwan_zhurh.uber, use CITY as

the key and find each CITY's mean travel time. Finally, we join two datasets together to see each city's population and mean travel time. Since mongodb don't support traditional join, so I use python to do the join work. We want to figure out if there is a relation between a city's population and mean travel time. Unfortunately, due to the limit of data and city, so far we don't find any obvious relation between them.

2.3 K-mean over Uber dataset

In uber_kmean.py, to satisfy with the request of implementing a constraint satisfaction or optimization technique in Project 2, we decide to implement k-mean over our uber dataset to see if we can find the main area that people most frequently go to. Since we have the data of uber travel destination name, we can use these destination name to get the destination's (latitude, longitude) by requesting Google Map API. But we cannot put this part in our script since each time's request will cost money and we cannot afford to request it again and again. So we directly load the data with (latitude, longitude) into the MongoDB. With the data with (latitude, longitude), we can implement k-mean to see how we can divide the destination into several groups of areas with various k value. And below is what we get. We also visualize the data by drawing a scatter plot to display each group of the center point, in order to get an intuitive insight into the data. A more complex visualization will be described in the Data Visualization part.



```
----------
[(42.38546496730506, -71.12863810683987)]
----------
[(42.5746589155779, -71.07980441758797), (42.3146950505639, -71.14690484398496)]
----------
[(42.31121337436976, -71.09528653550414), (42.44333992121211, -71.44959410202021), (42.57529918397436, -71.02671954551286)]
----------
[(42.169470184403664, -71.04530526697249), (42.59543071796874, -71.00610609609379), (42.446716758163255, -71.4513407632653), (42.361892087878765, -71.11132122954547)]
----------
[(42.60730443589743, -71.0051062700855), (42.25486849690721, -71.21735040103093), (42.37309257853105, -71.09280874858757), (42.461068398876385, -71.46288324382023), (42.17416442702703, -70.97706929594598)]
----------
[(42.37818882882351, -71.09134381647058), (42.47554124303795, -71.47527352278482), (42.20285491123595, -71.04008528988766), (42.27843163195875, -71.24636044329897), (42.612700637499984, -71.00374728303574), (42.13846459285715, -70.82476433571428)]
----------
[(42.571619914102556, -70.92379243461538), (42.631146427160495, -71.23680633580247), (42.37176313569321, -71.09241236489676), (42.36726228333332, -71.6311335722222), (42.29645366888888, -71.27257872222222), (42.19566879032258, -71.05021448387097), (42.13846459285715, -70.82476433571428)]
----------
[(42.14086371538462, -70.81585447692308), (42.85738442941175, -70.94694721176472), (42.35736260818181, -71.1028092339394), (42.186307666666664, -71.03184168266667), (42.272418099999975, -71.28727151375), (42.36747322424242, -71.64716999393939), (42.60885174444444, -71.24850984320987), (42.49243466764707, -70.96615140882354)]
----------
[(42.14086371538462, -70.81585447692308), (42.48405079652175, -70.97430214), (42.14796313846154, -71.20316334102564), (42.61027231625, -71.24606772875), (42.19997726249999, -71.0096019953125), (42.35376671213017, -71.11494113076921), (42.337959226530614, -71.42651033469384), (42.3861401625, -71.76956714375), (42.85738442941175, -70.94694721176472)]
----------
[(42.14086371538462, -70.81585447692308), (42.27751828787879, -71.54317463333331), (42.174681802325594, -71.21326520232557), (42.64293513108108, -71.232818477027), (42.35186879653977, -71.0808227276817), (43.00885675714286, -70.80925485714286), (42.52286514375, -71.63403499375), (42.38680966448597, -71.21494413831775), (42.180293767796606, -71.01070631864408), (42.51725726555555, -70.95896451333333)]
```

2.4 Statistic analysis over Uber dataset and Election dataset

In uber_statics.py, to satisfy with the request of implementing a statistical analysis in Project 2, we choose to compute correlation coefficient and p-value over Uber dataset and Election dataset to see if there is a correlation between them or not. We use both the functions that are defined in the slide and the scipy library to calculate the data. First of all, we analyze the relation between uber's distance and uber's travel time. In fact, even we don't calculate the correlation coefficient, with our intuition, the two variables should be positively related. The correlation coefficient is in line with our expectation, 0.8402 means that two variables are highly positively related. And the p-value is almost 0. When we do a hypothesis test, our H0 should be: uber's distance and uber's travel time are not related, our H1 should be: uber's distance and uber's travel time are related, we could set the significance level alpha = 0.05, since 0 < 0.05, we should reject H0, accept H1, so uber's distance and uber's travel time are related.
Now let's see some more interesting data sets. Let's try to find out if there is a relation between GOP's vote percent and the unemployment rate. This time I don't think we can judge it by our intuition so we will rely on the correlation coefficient. We choose 3 days to calculate the correlation coefficient and the value is between -0.145 ~ -0.081. It seems that these two variables are very weakly negative related.

```
Correlation coefficient of uber's distance and uber's travel time:
0.8401998346528659
p-value:
0.0
by using the library,we can get:
(0.8401998346528659, 6.208458756448018e-196)
----
/Users/chuci/github/cs504/course-2019-spr-proj/chuci_yfch_yuwan_zhurh/uber_statics.py:77: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  data_unemploy = data_unemploy[data_unemploy.sort_values('City').duplicated()].sort_values('City')
Correlation coefficient of GOP's vote percent and Dec-17's unemployment rate
-0.08135569863859243
p-value:
0.127
by using the library,we can get:
(-0.08135569863859253, 0.12928870745778923)
----
Correlation coefficient of GOP's vote percent and Dec-18's unemployment rate
-0.1447985150831711
p-value:
0.0105
by using the library,we can get:
(-0.14479851508317104, 0.006735440488431301)
----
Correlation coefficient of GOP's vote percent and Nov-18's unemployment rate
-0.10302328563011572
p-value:
0.0555
by using the library,we can get:
(-0.10302328563011559, 0.05449910126049234)
```

3.Data Visualization

3.1 Election Map

We finally produce two maps which are Senate map and House map. These two maps show progressiveness of voting districts of both Senate and House. We initially had the information of progressiveness score for each candidate and the detailed voting data for six different positions from 2002 to 2018 given by our Spark Partner Mass Alliance. We first calculated the progressiveness score of each precinct and then we integrated the data for every year and every position. After doing this, we average progressiveness scores for each precinct. The detailed rule to integrate different positions are President (25%), Governor (25%), Lieutenant Governor (10%), Senate (20%), Treasurer (10%), and Auditor (10%). The way to integrate all the voting years is just a simple mathematical average. Finally, we used these scores to build our map by using ArcGIS.

3.2 Election Chart

For this part, we used ECharts to help us to visualize our data. We firstly have the data of average scores for every precinct in each year and every position. By extracting data from csv file and constructing JSON String and doing some simple calculations, we successfully adjusted the format of JSON String and send these strings to ECharts. Besides, we also needed to select a template provided by ECharts. We selected the template chart with can help us better demonstrate our information and then we had to change some code of the template to make it exactly match our data. In the final chart, we have a line that contains six buttons which represent the six positions. By clicking on any one of those, the chart will be changed to the corresponding position. The x-axis represents different years and the y-axis represents the progressiveness scores. Each node in the graph means a precinct. To make the chart more clear, we only selected part of all the precincts.

3.3 Uber Map

In order for us to clearly see the “popular” region using the Uber destination data set, we choose to use the Leaflet to visualize the main area using the k-mean algorithm with different K. We first show the Boston map, then label all the mean center on the map, each k using a different color. By clicking each point in the map, the map will tell you the point's corresponding k. In this way, we can clearly see the most popular region in Boston.

Summary:

Overall, we built Senate Map and House Map to show progressiveness of voting districts of both Senate and House. We also built a Uber Map to show population distribution in Massachusetts. Besides, we demonstrated the score distribution of each of the six positions in different years with an interactive chart.

Based on the visualized Senate and House maps, we found that city areas where a high amount of population lived tend to have a higher progressiveness score (blue area in the maps), which means people living in cities tend to be more progressive (such as Boston, Cambridge, Springfield). By contrast, people living in rural areas tend to be more conservative. However, since people in Massachusetts mostly live in cities, we believe that is why the state of Massachusetts is considered more progressive compared with other states in the US. In addition, by comparing the Uber map to our political preference map, we found a positive correlation between popular regions on uber map and progressive regions on political maps. This observation proved our hypothesis that city areas tend to be more progressive.