Identifying Metropolitan Areas in the South East similar to Silicon Valley

Timothy Brannon

March 18, 2019

I. Introduction

i.i Background:

An upstart information technology company has grown rapidly and is looking to move into an office space that will serve as its Headquarters. The company is based in the South East United States and would like to remain in the South East region. The company would like to identify metropolitan areas in North Carolina, Virginia, Georgia, South Carolina and Eastern Tennessee as possible location candidates. As they are an upstart company, they would like to convey a modern and trendy business environment, as a possible draw for potential candidates for hire. The ideal location would be comparable to and offer similar talent pools, economic demographics and surrounding amenities as San Jose, California which is home to the largest tech hub in the United States and is known as Silicon Valley.

i.ii Data Problem:

Which metropolitan areas in North Carolina, Virginia, Georgia, South Carolina and Eastern Tennessee are most comparable to San Jose California? This will be answered by compiling and analyzing demographic and venue data for potential cities within the given region.

i.ii Relevance/Interest:

The Silicon Valley area is a desirable hub for new tech companies and talent. The ability for a tech company to identify a comparable area in their general geographic location may provide a variety of potential cost savings. Others looking to identify cities to move to may also benefit from being able to identify comparable cities from a potential list.

II. Data Strategy

ii.i Data Concept

Potential Headquarter Cities were identified by reviewing a list of moderate to large cities within the target are (South Eastern, United States). For each identified city, compile various demographic data describing the city, collect venue information to describe the type of venues located in the city. Comparable demographic and venue information will be compile for the Test City of San Jose, California. Demographic data will be reviewed and analyzed for Machine Learning feature selection. The feature selected compiled data and venue information will then be preprocessed to perform Machine Learning K-means clustering to potentially group prospective headquarter cities with the Test City.

> Note: Initial review of Silicon Valley showed several cities being associated with this area. Two additional Silicon Valley cities were identified and included in the data to see if they are clustered with the Test City, or any other prospective cities as a control. The two additional cities are Palo Alto and Santa Clara, California.

ii.ii Data Sources:

Population Data: Wikipedia maintains general city data for most cities within the United States. Standard information presented for most cities include recent national census population, recent population estimations, population density and area density information. This information is presented in a standard table forms an example of this data can be seen at https://en.wikipedia.org/wiki/San_Jose,_California.

Population Density Data: World Population Review maintains general United States City data table for 200 large cities within the United States. Standard information presented for these large cities includes recent national census population, recent population estimations, percent change and population density. This information is presented in a standard table at http://worldpopulationreview.com/us-cities/.

Demographic Data: AreaVibes.com maintains large database of demographic information for cites and neighborhoods in the United States. Each city has several pages on AreaVibes that include table information regarding crime, income, employment, education statistics. These tables are presented as tabs for each city an example can be viewed at https://www.areavibes.com/chattanooga-tn/crime/.

Venue Data: the FourSquare API will be queried for the venue data returning the top 100 venues in relation to each cities Latitude and Longitude values. Detailed document of the FourSquare API and the search feature can be found at https://developer.foursquare.com/docs/api

ii.iii Data Compilation:

Webscraping for Source Information:

Each Cities Wikipedia.com page is scraped for the main data table containing general city information, referred to as the "V-Card". Each Cities AreaVibes.com details table page is scraped for that page's table data, multiple tables/pages are scraped for each city. Each source data object for each City is stored in a nested dictionary where the primary key object is the City Name, and each sub-dictionary has an identical set of keys that correspond to source data objects. The WorldPopulationReview.com's US Cities page is scraped for its Cities table data and stored as a data-frame object. The dictionary structure used to store source information objects allow for a uniform storage and access mechanism for demographic data, allowing for the production of City Demographic data, as well as, allowing for future access. Furthermore, this dictionary can be written to a separate json file which would provide all source information used for any clustering session. See Figure 1. for a detailed description of the nested dictionary keys. See Figure 1.A for a detailed description of sub-dictionary keys.

Figure 1: Dictionary Keys

Dictionary Key(s): ['San Jose', 'Palo Alto', 'Santa Clara', 'Raleigh', 'Charlotte', 'Atlanta', 'Charleston', 'Winston Salem', 'Augusta', 'Columbia', 'Durham', 'Richmond', 'Norfolk', 'Alexandria', 'Florence', 'Savannah', 'Asheville', 'Roanoke', 'Knoxville', 'Greensboro', 'Cary', 'Athens', 'Albany', 'Chattanooga']

Dictionary Value(s): [Sub Dictionary]

Figure 1.1: Sub-dictionary value items

Sub Dictionary Structure (for each Dictionary Key)

Keys:	Values:
Html:	Stored Wikipedia Request response - for record
Soup:	Beautiful Soup Parsed Html from Stored Response - for record
Geo:	Latitude/Longitude from Soup Parsed Html - provides data values (Latitude; Longitude)
Pop:	Wikipedia Population Table from Soup Parsed Html presented as a DataFrame - provides data values (Estimated Population; Census Population 2010)
Crime:	AreaVibes Crime Table get response stored as a DataFrame - provides data values (Crime/100k ppl; Violent Crime/100k; Property Crime/100k)
Housing:	AreaVibes Housing Table get response stored as a DataFrame (Median Home Price; Median Rent Ask)
Income:	AreaVibes Income Table get response stored as a DataFrame (Income per capita; Median household income; Unemployment rate; Poverty level)

Storing Source Information after Scraping:

Demographic data is compiled for each city in the same fashion. Each Cities nested dictionary is accessed for source html data and processed as followed. **"Longitude"** and **"Latitude"** is presented in the **"Geo"** class object for each Wikipedia page. The Geo class object is identified within the parsed html data stored as the corresponding value for each cities [Soup] dictionary key. The returned value from the **"Geo"** class object consists of both longitude and latitude values presented as a text string separated with a semi-colon (;) this text string is stored in the nested dictionary as the corresponding value to each cities **[Geo]** key. Population data for each city is presented in a population table stored in the value for each cities **[Soup]** key, with a table class **"toccolours"**. The population table is then stored as the value that corresponds to each cities dictionary **[Pop]** entry "**Crime per 100k"**, **"Violent Crime per 100k"** and **"Property Crime per 100k"** values are presented at a Cities AreaVibes website within the Crime section for each city. For each city the crime table is read from the website directly into a pandas data-frame and stored as a dictionary value that matches the **[Crime]** key. Dictionary values for each cities **[Housing]** and **[Income]** keys are accessed and stored in the same fashion. They are read into a pandas data-frame directly from the AreaVibes website. Tables read in from AreaVibes for Crime, Housing and Income have their headers adjust to normalize data access. The tables contain 3 columns that present data, these columns are renamed [City, State, Nat] respectively.

ii.iv Cleaning Compiled Data:

Clean data for each city is stored into a nested list for ease of creating a panda's data-frame. Each list is created with the city name as the first entry and appended with the identified data values described previously.

Longitude and Latitude is created by accessing each cities **[Geo]** dictionary entry. The returned data consists of a string containing Longitude and Latitude values separated by white space and a semi-colon. This string is sliced by finding the index location of whitespace and the semi-colon, and omitting all trailing values for "**Latitude"**, and all receding values for "**Latitude**", these values are appended to the nested list.

The **[Pop]** dictionary entry is accessed for each city, and returns a table that contains "**Current Estimate"** and "**2010 Census"** values as strings along with previous census population totals, this table is used to populate those values. Current estimate and 2010 census is accessed by indexing the table, identifying the

row index that matches the estimate and 2010 census title strings, and returning the adjacent population value from the next column within the same row index, these values are appended to the nested list.

The "**Population Density**" variable is created by accessing the World Population review Cities table, where each cities name corresponds to a row header, city name is used to index rows, and access column data returning a string value for "**Population Density"**

The **[Crime]** dictionary is accessed for each city values for "Crime per 100k", **"Violent Crime per 100k"** and **"Property Crime per 100k"** are each identified by selecting only the table row and table column that corresponds to row headers for each variable stored as strings. **[Housing]** and **[Income]** dictionary entries are then accessed for each city where their respective values are identified and appended to the nested list in the same fashion. Values are presented by matching specific row headers that correspond to the desired data values.

This uniform method of storing and accessing data resulted in a very clean data set, where there were only 7 missing variables these variables were all for Population Density. Value were missing for the following cities Palo Alto, Santa Clara, Florence, Asheville, Roanoke, Knoxville, Athens and Albany. These values were supplied by doing an individual search for population density for each city, notating the source address and inserting the value in the corresponding data-frame cell for each city. See Figure 2 for an example of the compiled demographic dataset.

Figure 2: Demographic Dataset

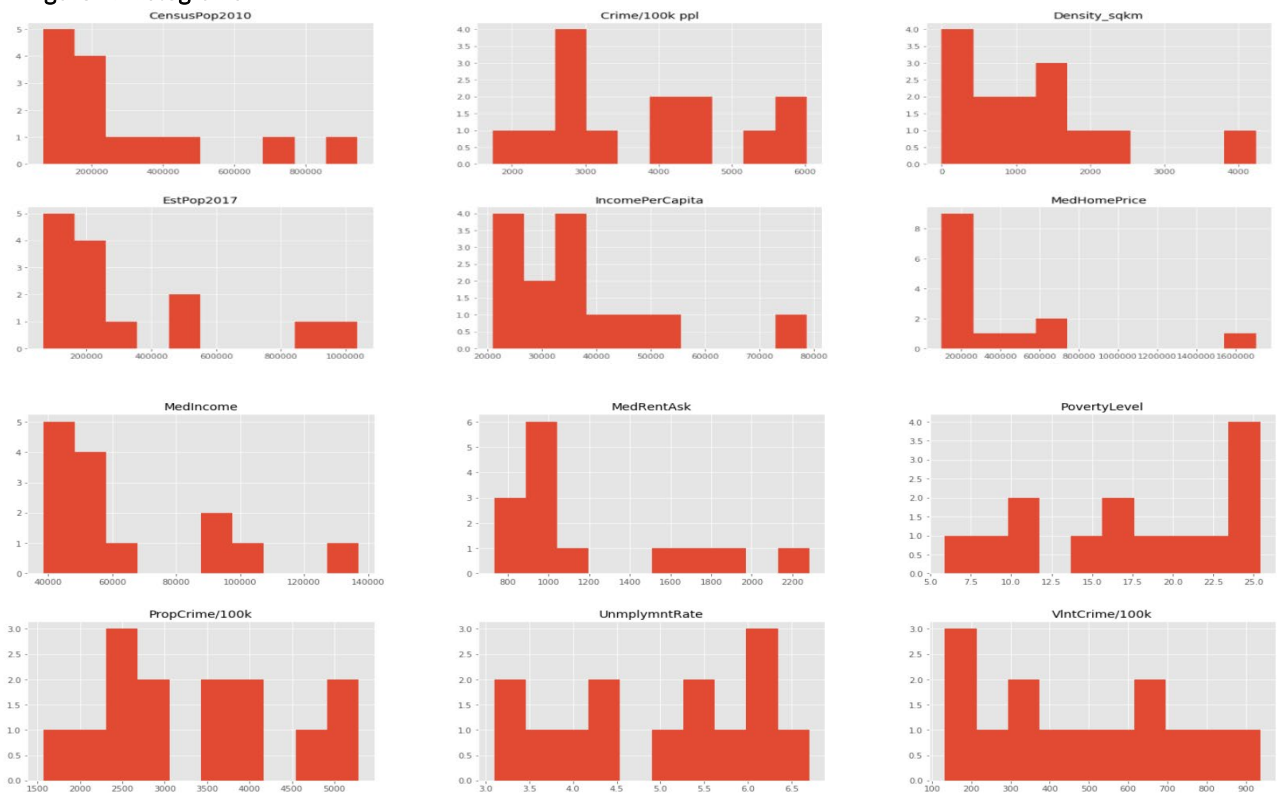
	City	Latitude	Longitude	EstPop2017	CensusPop2010	Density_sqkm	Crime/100k ppl	VIntCrime/100k	PropCrime/100k	IncomePerCapita	MedIncome	UnmplymntRate	PovertyLevel
0	San Jose	37.333	-121.900	1046964	945942	2248	2844	404	2441	37845	90303	5.0	10.9
1	Palo Alto	37.429	-122.138	67024	64403	964	2777	133	2644	78721	137043	3.1	5.9
2	Santa Clara	37.354	-121.969	125948	116468	2443	3005	168	2837	44843	102533	3.6	8.2
3	Raleigh	35.767	-78.633	479332	403892	1293	2657	333	2325	33682	58641	4.3	14.9
4	Charlotte	35.227	-80.8430	859035	731424	1121	4479	663	3815	33050	55599	6.1	15.8
5	Atlanta	33.755	-84.3900	486290	420003	1450	5712	936	4776	38686	49398	6.7	24.0
6	Charleston	32.783	-79.9333	138036	120083	496	2581	284	2297	35948	57603	4.0	16.3
7	Winston Salem	36.102	-80.2604	244605	229617	720	4179	523	3656	25852	40898	6.2	24.3
8	Augusta	33.467	-81.967	197166	195844	253	5555	616	4938	20876	38458	6.3	25.3
9	Columbia	34.000	-81.0347	133114	129272	387	6020	734	5286	25990	42875	5.4	22.9
10	Durham	35.988	-78.9072	267743	228330	978	4619	792	3826	30645	52115	4.5	18.5
11	Richmond	37.533	-77.467	223170	204214	1507	3198	332	2866	29011	41187	5.9	25.4
12	Norfolk	36.917	-76.200	244703	242803	1754	4305	556	3749	25450	45268	5.5	21.0
13	Alexandria	38.804	-77.0472	160035	139966	4233	1734	166	1568	55534	89200	3.1	9.8
14	Florence	34.183	-79.7741	38317	37056	660	8362	1267	7095	28130	44989	5.7	17.6
15	Savannah	32.017	-81.117	146444	136286	556	4169	463	3706	21192	37108	7.5	25.4
16	Asheville	35.580	-82.5558	91902	83393	779	5326	612	4715	28933	44946	3.9	16.2
17	Roanoke	37.270	-79.9416	99837	97032	776	4894	402	4492	23611	39201	4.7	22.2
18	Knoxville	35.972	-83.9422	187347	178874	741	6338	894	5445	23086	34556	4.7	26.5
19	Greensboro	36.080	-79.8194	290222	269666	878	4288	725	3562	26809	42802	4.9	19.9
20	Cary	35.791	-78.7811	165904	135234	1164	1139	64	1075	43925	94617	2.9	5.9
21	Athens	33.950	-83.383	125691	115452	329	3562	416	3146	20094	32959	5.2	35.5
22	Albany	31.582	-84.1655	73801	77434	513	6752	1130	5622	18887	31263	10.6	33.2
23	Chattanooga	35.045	-85.2672	177571	167674	490	7052	1066	5986	25659	41278	5.3	21.1

ii.v Analyzing Demographic data for Feature Selection:

Visualizing Demographic Data Distribution:

Demographic data is visualized with matplotlib histogram plots to review data distribution. In general most plots show a fairly uniform distribution of demographic data. The areas that do not confirm to a uniform or normal distribution may be a result of having a limited potential data set. For example constraining prospective cities to the south east does limit the number of large cities that could be included in the data set which may contribute to the left skewness of the Population data. Histogram plots for demographic data can be seen in Figure 2. No data features were identified to be removed as a result of distribution review.

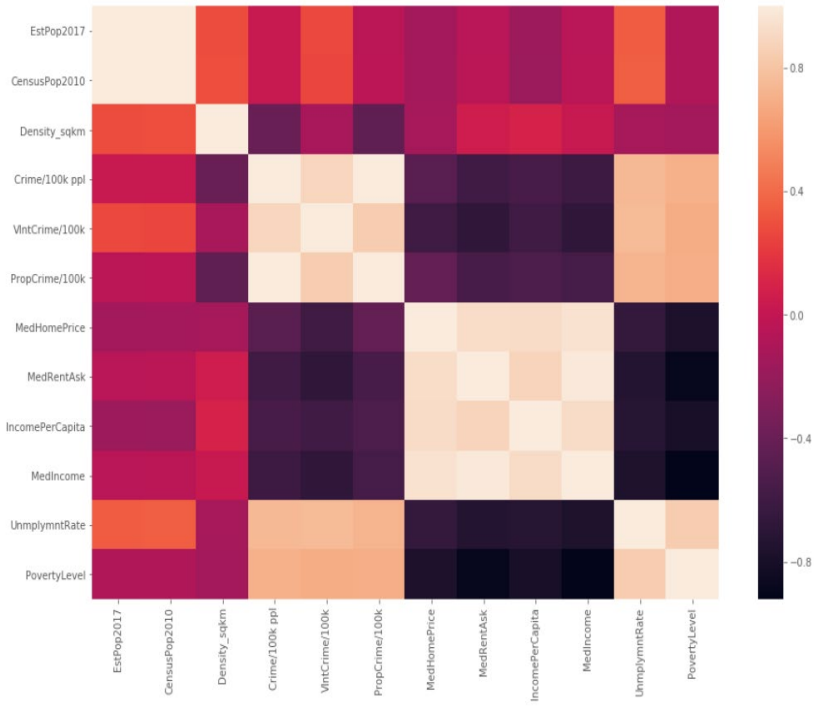
Figure 2: Histograms



Visualizing Demographic Data Correlation (Heat-map):

Correlation analysis is performed on the demographic data set to visualize potential relationships between compiled variables. Each variable is plotted on a heat-map with darker blocks representing high negative correlation between variables. Light color blocks represent high positive correlation between variables. See Figure 3. for the heat-map representation of correlation.
A data-frame is also displayed representing the correlation values displayed in the presented heat-map. See Figure 3.1 for specific variable correlation coefficients.

Figure 3: Heat-map



Some compiled data variables display a strong Correlation. For example Estimate Population, and 2010 Census Population has A strong positive correlation. This is depicted as a light-colored block on the heat-map, and with a correlation coefficient of .997. Crime per 100k people also display a high positive correlation with the two breakdown variables of the total included. (Violent Crime per 100k and Property Crime per 100k) This correlation is also indicated by 2 light-colored blocks on the heat-map, and with correlation coefficients of .8945 and 9984.respectivley.

As a result of the correlation between Estimate Population and 2010 Population these two values were replaced with a growth rate value calculated by subtracting 2010 Census Population from the Estimate

Population dividing that value by the Estimate Population. (Estimate Population -2010 Population/ Estimate Population)

Figure 3.1: Correlation Coefficients

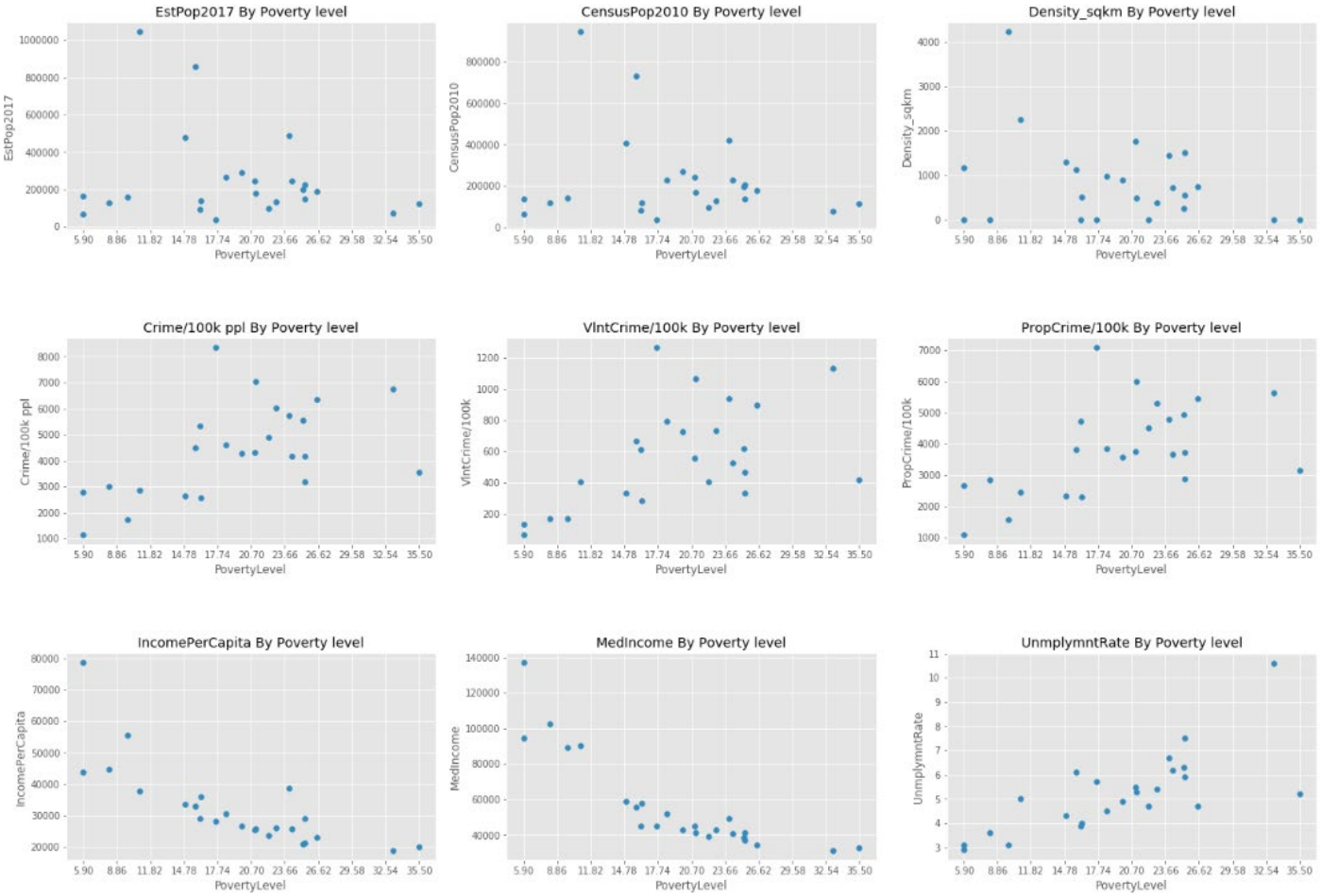
	EstPop2017	CensusPop2010	Density_sqkm	Crime/100k ppl	VIntCrime/100k	PropCrime/100k	MedHomePrice	MedRentAsk	IncomePerCapita	MedIncome	UnmplymntRate	PovertyLevel
EstPop2017	1.000000	0.997838	0.286064	0.019074	0.269935	-0.038316	-0.147008	-0.049215	-0.172990	-0.043771	0.345003	-0.091664
CensusPop2010	0.997838	1.000000	0.287468	0.023927	0.263989	-0.031141	-0.140360	-0.041190	-0.184449	-0.042365	0.354128	-0.082992
Density_sqkm	0.286064	0.287468	1.000000	-0.396520	-0.122776	-0.447099	-0.132145	0.056673	0.096624	0.021245	-0.130833	-0.141981
Crime/100k ppl	0.019074	0.023927	-0.396520	1.000000	0.894528	0.994837	-0.475914	-0.592055	-0.559216	-0.606560	0.751857	0.712996
VIntCrime/100k	0.269935	0.263989	-0.122776	0.894528	1.000000	0.844547	-0.605400	-0.676352	-0.592609	-0.674584	0.758224	0.688070
PropCrime/100k	-0.038316	-0.031141	-0.447099	0.994837	0.844547	1.000000	-0.432656	-0.555670	-0.535397	-0.573463	0.728581	0.697927
MedHomePrice	-0.147008	-0.140360	-0.132145	-0.475914	-0.605400	-0.432656	1.000000	0.928567	0.920007	0.948044	-0.655517	-0.775132
MedRentAsk	-0.049215	-0.041190	0.056673	-0.592055	-0.676352	-0.555670	0.928567	1.000000	0.884806	0.985551	-0.734852	-0.889889
IncomePerCapita	-0.172990	-0.184449	0.096624	-0.559216	-0.592609	-0.535397	0.920007	0.884806	1.000000	0.920182	-0.723956	-0.799814
MedIncome	-0.043771	-0.042365	0.021245	-0.606560	-0.674584	-0.573463	0.948044	0.985551	0.920182	1.000000	-0.770145	-0.921144
UnmplymntRate	0.345003	0.354128	-0.130833	0.751857	0.758224	0.728581	-0.655517	-0.734852	-0.723956	-0.770145	1.000000	0.849090
PovertyLevel	-0.091664	-0.082992	-0.141981	0.712996	0.688070	0.697927	-0.775132	-0.889889	-0.799814	-0.921144	0.849090	1.000000

Similarly, the correlation between Crime per 100k in relation to Violent Crime per 100k and property crime per 100k resulted in the decision to omit the Crime per 100k variable. It is dropped from the final Feature Selected data-frame.

Visualizing Demographic Data Correlation with Scatterplots:

Scatterplots are also used to display correlation between the standardized demographic statistics (Poverty Level) to further visualize relationships. Scatterplots show some loose linear relationships though given the size of data set these observations did not influence additional feature selection. Scatterplot depictions are shown in Figure 4.

Figure 4: Poverty Level Scatter Plots



Final Feature Selection:

After analyzing demographic data, the final feature selection resulted in creating a growth rate variable and dropping two highly correlated population statics, as well as, dropping the Crim per 100k variable due to its high correlation with two other crime variables that who's sum total equals Crime per 100k. This resulted in a final feature selected data set which consist of the following variables. [Density_sqkm; VIntCrime/100k; PropCrime/100k; IncomePerCapita; MedIncome; UnmplymntRate; PovertyLevel; GrowthRate]

Demographic Feature Preprocessing:

In order to prepare feature selected data for machine learning the minimum/maximum standard scaler is imported from matplotlib and used to scale all selected features within the demographic data set. This process assigns a value of 1 to the highest column value and scales each corresponding value with the 0 being equal to 0. This scaled data is now in a state of pre-processing where it can be merged with summarized FourSquare Venue data and passed to the K-means algorithm for clustering. Scaled demographic data can be seen in Figure 5.

Figure 5

	City	Density_sqkm	VIntCrime/100k	PropCrime/100k	IncomePerCapita	MedIncome	UnmplymntRate	PovertyLevel	GrowthRate
0	San Jose	0.501256	0.282627	0.226910	0.316843	0.558140	0.272727	0.168919	0.622477
1	Palo Alto	0.178643	0.057357	0.260631	1.000000	1.000000	0.025974	0.000000	0.377339
2	Santa Clara	0.550251	0.086451	0.292691	0.433800	0.673757	0.090909	0.077703	0.531824
3	Raleigh	0.261307	0.223608	0.207641	0.247267	0.258820	0.181818	0.304054	0.882610
4	Charlotte	0.218090	0.497922	0.455150	0.236705	0.230062	0.415584	0.334459	0.844872
5	Atlanta	0.300754	0.724855	0.614784	0.330899	0.171441	0.493506	0.611486	0.792585
6	Charleston	0.061055	0.182876	0.202990	0.285139	0.249007	0.142857	0.351351	0.765881
7	Winston Salem	0.117337	0.381546	0.428738	0.116405	0.091085	0.428571	0.621622	0.472040
8	Augusta	0.000000	0.458853	0.641694	0.033242	0.068019	0.441558	0.655405	0.238931
9	Columbia	0.033668	0.556941	0.699502	0.118712	0.109775	0.324675	0.574324	0.333583
10	Durham	0.182161	0.605154	0.456977	0.196510	0.197126	0.207792	0.425676	0.839118
11	Richmond	0.315075	0.222776	0.297508	0.169201	0.093817	0.389610	0.658784	0.573134
12	Norfolk	0.377136	0.408978	0.444186	0.109687	0.132397	0.337662	0.510135	0.243457
13	Alexandria	1.000000	0.084788	0.081894	0.612478	0.547712	0.025974	0.131757	0.745989
14	Florence	0.102261	1.000000	1.000000	0.154477	0.129760	0.363636	0.395270	0.350872
15	Savannah	0.076131	0.331671	0.437043	0.038523	0.055256	0.597403	0.658784	0.506600
16	Asheville	0.132161	0.455528	0.604651	0.167898	0.129353	0.129870	0.347973	0.605805
17	Roanoke	0.131407	0.280964	0.567608	0.078952	0.075043	0.233766	0.550676	0.330308
18	Knoxville	0.122613	0.689942	0.725914	0.070177	0.031131	0.233766	0.695946	0.403486
19	Greensboro	0.157035	0.549460	0.413123	0.132400	0.109085	0.259740	0.472973	0.512854
20	Cary	0.228894	0.000000	0.000000	0.418458	0.598922	0.000000	0.000000	1.000000
21	Athens	0.019095	0.292602	0.344020	0.020172	0.016033	0.298701	1.000000	0.558277
22	Albany	0.065327	0.886118	0.755316	0.000000	0.000000	1.000000	0.922297	0.000000
23	Chattanooga	0.059548	0.832918	0.815781	0.113180	0.094678	0.311688	0.513514	0.448379

Accessing Four Square Venue Data via API:

FourSquare.com maintains a large-scale crowdsourced database of local venue information which can be accessed and searched with a developer API. The FourSquare API was queried for each city by passing the cities' latitude and longitude values along with response limit and range definitions. The API returns a JSON object for each cities' API call, these responses are stored in the "Response" variable as a nested diction item and are accessed to build a list of all returned Venue Names, Latitude/Longitude values and the venue category type. A sample of API response information can be seen in Figure 6.

Figure 6: Venue Response Sample (data-frame shape = 2400 x 7)

	City	City Latitude	City Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	San Jose	37.333	-121.900	SAP Center at San Jose	37.332600	-121.901387	Hockey Arena
1	San Jose	37.333	-121.900	Henry's Hi-Life	37.335405	-121.898080	BBQ Joint
2	San Jose	37.333	-121.900	Whole Foods Market	37.332086	-121.904623	Grocery Store
3	San Jose	37.333	-121.900	Poor House Bistro	37.329650	-121.900431	Cajun / Creole Restaurant
4	San Jose	37.333	-121.900	San Pedro Square Market	37.336480	-121.894403	Food Court

Pre-Processing FourSquare Data:

The normalized method of the JSON response and storing that information in a nested dictionary format allows for the response venue data to be recursively populated into a panda's data-frame. This new Venues data-frame has a shape of 2400 rows and 7 columns. Each row corresponds to a specific venue for a specific city. Due to Limits imposed within the FourSquare API each city has 100 rows representing top venues. This is shown by displaying a summary of the Venues data-frame. Figure 7. Venues data is then passed to a data-frame using the onehot encoding method. This method creates a column for each venue category and assigns a value of one to cells that match the category, all other cells are marked with zeros. This allows for all rws to be grouped by City, while averaging the sum values of each category as a result on grouping by City. This produces a summarized and normalized venue category value for each category for each city. An example of this summarized Venue data set can be seen in Figure 8. The summarized dataset presents a data-frame shape that is 24 rows by 278 columns of venue categories. The summarized data-frame is then able to be merged with the feature selected demographic data set.

Figure 7

City	City Latitude	City Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
Albany	100	100	100	100	100	100
Alexandria	100	100	100	100	100	100
Asheville	100	100	100	100	100	100
Athens	100	100	100	100	100	100
Atlanta	100	100	100	100	100	100

Figure 8

```
onehot shape: (2400, 278)
venNorm shape: (24, 278)
```

	City	Accessories Store	American Restaurant	Animal Shelter	Antique Shop	Aquarium	Arcade	Argentinian Restaurant	Art Gallery	Art Museum	Arts & Crafts Store	Arts & Entertainment	Asian Restaurant	Athletics & Sports	Automotive Shop	BBQ Joint	Bagel Shop	Bakery	Bar	Baseball Field	Baseball Stadium	Basketbal Stadiun
0	Albany	0.0	0.07	0.0	0.0	0.01	0.02	0.00	0.00	0.01	0.00	0.0	0.0	0.00	0.0	0.02	0.00	0.00	0.01	0.0	0.00	0.0(
1	Alexandria	0.0	0.05	0.0	0.0	0.00	0.00	0.00	0.01	0.00	0.01	0.0	0.0	0.00	0.0	0.01	0.00	0.02	0.00	0.0	0.00	0.0(
2	Asheville	0.0	0.03	0.0	0.0	0.00	0.00	0.00	0.02	0.00	0.00	0.0	0.0	0.00	0.0	0.02	0.00	0.03	0.04	0.0	0.01	0.0(
3	Athens	0.0	0.02	0.0	0.0	0.00	0.01	0.01	0.00	0.01	0.00	0.0	0.0	0.00	0.0	0.02	0.02	0.02	0.13	0.0	0.00	0.0(
4	Atlanta	0.0	0.04	0.0	0.0	0.01	0.01	0.00	0.00	0.00	0.00	0.0	0.0	0.01	0.0	0.00	0.00	0.01	0.03	0.0	0.00	0.0

Merging Data Sets:

The final preprocessing step prior to engaging in K-means clustering is to merge the feature selected demographic data set, with the FourSquare API venue information. This step is done by creating a new data-frame that merges doth data sets along the "City" column. This is stored as **"citymerged"** merged variable.

III. Machine Learning and Analysis:

iii.i K-Means Clustering:

Argument Parameters:

With all data compiled and scaled the K-Means algorithm is used to perform an unsupervised machine learning clustering algorithm. This algorithm will take in all provided data and attempt to match like objects based on the input data. This process will label like objects allowing for cities comparable to San Jose California to be grouped along with it.

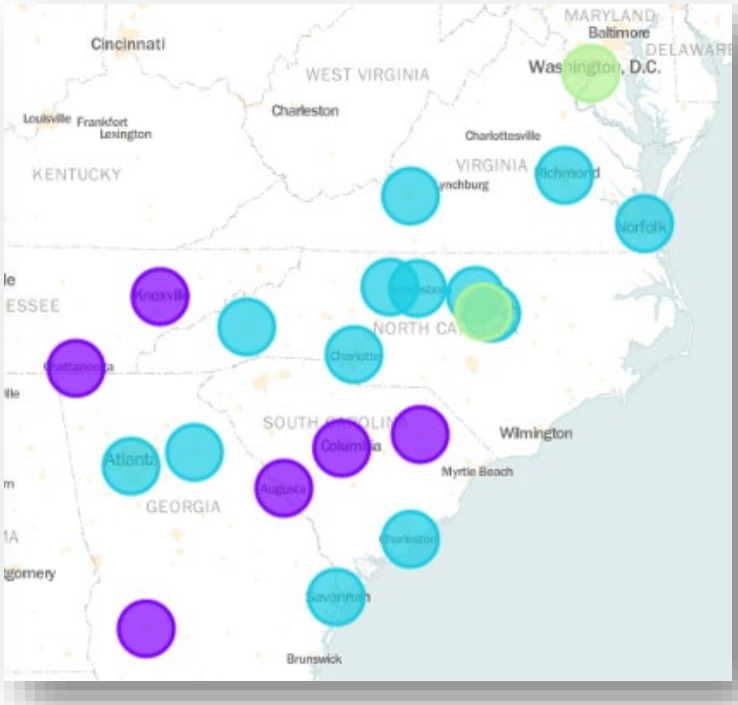
The algorithm takes several augments that may impact the returned cluster labels, with the main argument parameters being cluster size and random state. Clyster size defines the total number of groupings available for the K-means algorithm to lace objects. Random state determines the initial number generation for center points for each identified group. Given the limited number of total city objects being compared the cluster count was set to equal 3 allowing for three distinct groupings. The random state value was set to 0.

Unsupervised label assignment:

With the K-means algorithm parameters set the **'citymerged'** data set is passed to the K-means algorithm for clustering. The clustering algorithm assign each row object a number of the group with which comparable objects belong. Labels are generated for the Test City, Control Cities, and Prospective Cities. These labels are added back to the **'citymerged'** data –frame and plotted using the Folium map library. This can be seen in Figure. 9 & Figure 9.1.

Figure 9

Figure 9.1



Clustering Analysis & Recommendations:

The target and control cities were indeed added to the same cluster group given data features and machine learning parameters, despite varying differences in growth rate, density, violent crime, property crime, unemployment rate and poverty level amongst them. There were two additional cities that were included in the target city grouping, Cary, NC and Alexandria VA given the feature set and K-means settings. These cities share common top venue categories like Coffee Shops, Ice Cream Shops, Restaurants, Parks and Plaza's.

Discussion & Considerations:

Though Cary, NC and Alexandria VA were identified as recommended prospective cities. Additional demographic data points, other metrics, scale procedures and K-means setting decision may affect the output results of the clustering the same sample set. Additionally, increasing the prospective cities options may also produce differing cluster results. While these varying criteria are to extensive to list, additional controls were included to bolster the recommendation. Controls include ensuring fulsome data set with no missing values, including two control cities, as well as, additional iterations of the K-means algorithm at 2 and 4 possible cluster groupings. Additional iterations also grouped Cary, NC and Alexandria, VA in the Target grouping.