Joint Returns as an Influence on Dependents in Massachusetts Zip Code Areas

Aditi Dass and Navraj Narula

Abstract

According to the Internal Revenue Service (IRS), a dependent is a either a "qualifying child" or a "qualifying relative," or rather a person who depends on someone else and perhaps their earnings for survival. Beyond the general definition specified by the IRS, this could mean a person's retired parents or someone with a mental or physical disability. For our project, we have chosen to examine the number of dependents in congruence with the number of marriages per zip code area in the state of Massachusetts. Our assumption is that since married couples tend to signify more children, a discrepancy would signify an aging community. We would like to see how far this notion extends when considering different types of communities. An ageing population is generally not good for the economy in terms of costs, but more children generally result in greater future stability.

Introduction

Our primary dataset was retrieved from the Internal Revenue Service (IRS). Made available for public use, it contains a collection of zip codes from the state of Massachusetts during 2013. Our data, though comprised of strings, is completely numeric. Out of over 250,000 zip codes in total, we have chosen to focus on a subset of 3,866 entries. These rows hold 112 attributes in total, with some values appearing as null. Several of these attributes include:

- Adjusted gross income
- Number of personal exemptions
- Number of returns
- Number of Dependents
- Wages and Salaries
- Unemployment compensation
- Interest Received

Though Massachusetts only has roughly 670 zip codes according to our dataset, the actual number of entries (as mentioned previously) exceed this value due to the divides set forth by creating separate row entries for five income classes: $1 under $25000, $25000 under $50000, $50000 under $75000, $75000 under $100000, $100000 under $200000, $200000 or more. Another row includes the numeric total for all these cases. In conclusion, each zip code has seven separate entries.

We are examining on these three goals for our project:

1. Determine whether or not a correlation exists between the number of dependents and the number of joint returns.
2. Identify zip codes of densely populated areas (cities, towns, etc.) vs. sparsely populated areas (rural, suburban, etc).
3. Discover whether certain communities produce a higher number of dependents or number of joint returns than others and whether a community's ratio of dependents and joint returns makes it economically stable or not.

Our secondary dataset was retrieved from ProximityOne. It contains a collection of all zip codes in the United States in general along with attributes including:

- Land area (square meters)
- Water area (square meters)
- Internal Point (latitude/longitude)
- Population

In total, there exists 33,120 entries and 9 attributes. We are utilizing this dataset to assist us in analyzing our second and third goals mentioned above. We are only considering row entries 131 to 668, which is the subset pertaining to zip codes in Massachusetts. Specifically, these zip codes are 01001 to 02791.

We will go into greater detail in the next section (i.e. techniques) in regards to how we perform the analysis and utilize both datasets.

Techniques

Our first goal was to determine whether or not a correlation exists between the number of dependents and the following three attributes: number of single returns, number of joint returns, and total number of returns inclusive of both groups. To achieve this task, we first created a separate data frame using the pandas library that only included the attributes we care about. We then accumulated each entry into a list corresponding to each attribute. They are: number of dependents, number of single returns, number of joint returns, and total number of returns. We then used the matplotlib library to display the results in three graphs which spoke the correlation (or lack of it) among: number of dependents vs. number of single returns, number of dependents vs. number of joint returns, and number of dependents vs. total number or returns. We then used then registered our regression results for each graph by utilizing the statsmodel library.

Our second goal was to identify communities of densely populated areas (cities, towns, etc.) vs. sparsely populated areas (rural, suburban, etc). For this task, we took into consideration Proximity One's dataset on zip codes, restricting ourself to lines 131 to 668 which correspond to zip codes in the Massachusetts area. We first created an array of longitudinal and latitudinal values to cluster. In order to produce an accurate representation of the multiple clusters of neighbouring nodes, we decided to utilise Affinity Propagation. Affinity Propagation was chosen over the more commonly used K-Means because the latter would have required us to decide on a number of clusters while the former calculated the optimal number for us by identifying the exemplary points.

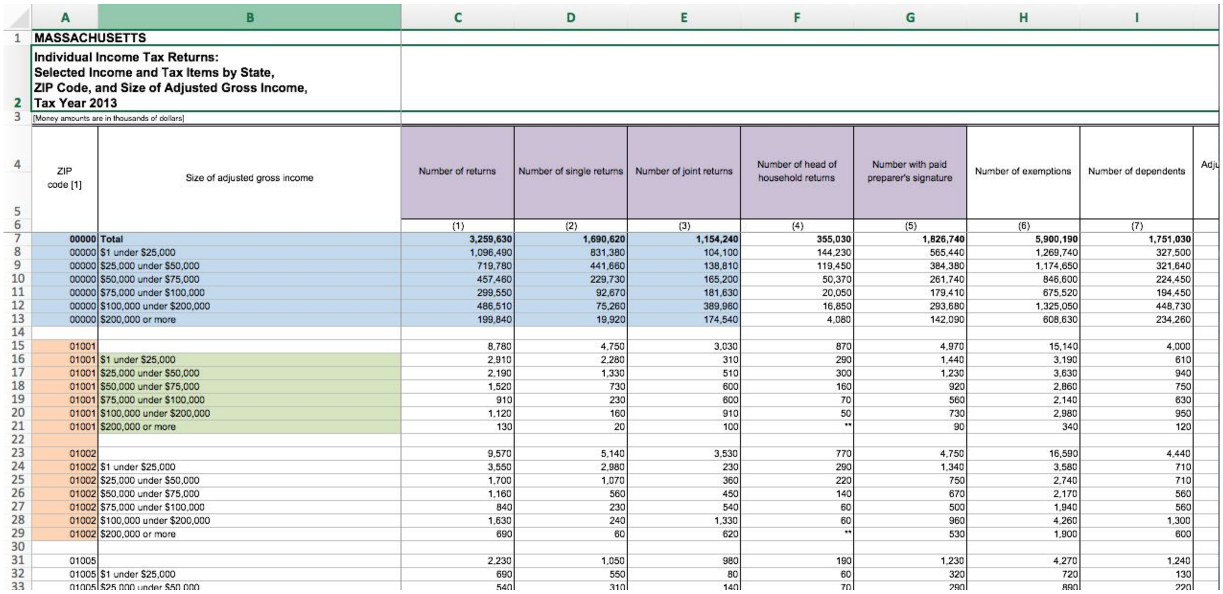
Our third goal was to discover whether certain communities produce a higher number of dependents or number of joint returns than others and whether a community's ratio of dependents and joint returns makes it economically stable or not. Here we also took into consideration the number of single returns and the total number of returns in general. We first read in both datasets using the pandas library. Following that, a default dictionary was initialized to contain attributes pertaining to each zip code. An example entry is below, with the key being the zip code itself:

```
1026: {'dependents': 200,
        'income': 27217,
        'joint': 250,
        'returns': 530,
        'single': 210}
```

The following information was received from our primary dataset strictly concerning zip codes in the Massachusetts area. Utilizing our secondary dataset, in which we only worked on a subset pertaining to Massachusetts zip codes, we then aggregated values into two separate arrays in order to work with clusters. We chose to access the longitude and latitude variables pertaining to each zip code in order to inform our cluster predictions. Our most effective clustering method turned out to be affinity propagation, which we will elaborate further on in our discussion section. For each cluster of zip codes, we then found the total number of dependents, income, joint returns, single returns, and returns pertaining to each location. This process was made easy due to our initial thought to initialize a dictionary containing these totals as values for each key that described our attribute. We also collected the average number of dependents, income, joint returns, single returns, and overall returns of all zip codes in the Massachusetts area. Having all our information stored allowed us to perform regression analysis and graph scatter plots in order to achieve our third goal. In summary, we analyzed: the total number of dependents per return vs. the total number of joint returns per return, the total number of dependents per return vs. the total number of single returns per return, and the total number of dependents per return vs. the total income per return.

Results for all goals will be elaborated on in the next section.

Datasets and Experiments



MASSACHUSETTS

Individual Income Tax Returns:
Selected Income and Tax Items by State,
ZIP Code, and Size of Adjusted Gross Income,
Tax Year 2013

[Money amounts are in thousands of dollars]

ZIP code [1]	Size of adjusted gross income	Number of returns	Number of single returns	Number of joint returns	Number of head of household returns	Number with paid preparer's signature	Number of exemptions	Number of dependents	Adju
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	
00000	**Total**	**3,259,630**	**1,690,620**	**1,154,240**	**355,030**	**1,826,740**	**5,900,190**	**1,751,030**	
00000	$1 under $25,000	1,096,490	831,380	104,100	144,230	565,440	1,269,740	327,500	
00000	$25,000 under $50,000	719,780	441,660	138,810	119,450	384,380	1,174,650	321,640	
00000	$50,000 under $75,000	457,460	229,730	165,200	50,370	261,740	846,600	224,450	
00000	$75,000 under $100,000	299,550	92,670	181,630	20,050	179,410	675,520	194,450	
00000	$100,000 under $200,000	486,510	75,260	389,960	16,850	293,680	1,325,050	448,730	
00000	$200,000 or more	199,840	19,920	174,540	4,080	142,090	608,630	234,260	
01001		8,780	4,750	3,030	870	4,970	15,140	4,000	
01001	$1 under $25,000	2,910	2,280	310	290	1,440	3,190	610	
01001	$25,000 under $50,000	2,190	1,330	510	300	1,230	3,630	940	
01001	$50,000 under $75,000	1,520	730	600	160	920	2,860	750	
01001	$75,000 under $100,000	910	230	600	70	560	2,140	630	
01001	$100,000 under $200,000	1,120	160	910	50	730	2,980	950	
01001	$200,000 or more	130	20	100	**	90	340	120	
01002		9,570	5,140	3,530	770	4,750	16,590	4,440	
01002	$1 under $25,000	3,550	2,980	230	290	1,340	3,580	710	
01002	$25,000 under $50,000	1,700	1,070	360	220	750	2,740	710	
01002	$50,000 under $75,000	1,160	560	450	140	670	2,170	560	
01002	$75,000 under $100,000	840	230	540	60	500	1,940	560	
01002	$100,000 under $200,000	1,630	240	1,330	60	960	4,260	1,300	
01002	$200,000 or more	690	60	620	**	530	1,900	600	
01005		2,230	1,050	980	190	1,230	4,270	1,240	
01005	$1 under $25,000	690	550	80	60	320	720	130	
01005	$25,000 under $50,000	540	310	140	70	290	890	220	

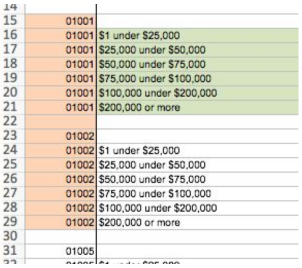
The above is a screenshot of a section of our **dataset from IRS.**

Rows: 3870

Columns: 112 columns

110 quantitative columns

2 qualitative columns



01001	
01001	$1 under $25,000
01001	$25,000 under $50,000
01001	$50,000 under $75,000
01001	$75,000 under $100,000
01001	$100,000 under $200,000
01001	$200,000 or more
01002	
01002	$1 under $25,000
01002	$25,000 under $50,000
01002	$50,000 under $75,000
01002	$75,000 under $100,000
01002	$100,000 under $200,000
01002	$200,000 or more
01005	

The orange box represents the zip code numbers and the green box shows the division of data for each zip code area:

$1 under $25,000

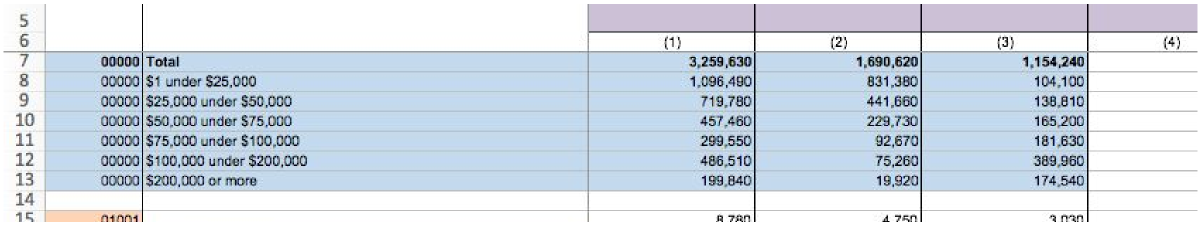
$25,000 under $50,000

$50,000 under $75,000

$75,000 under $100,000

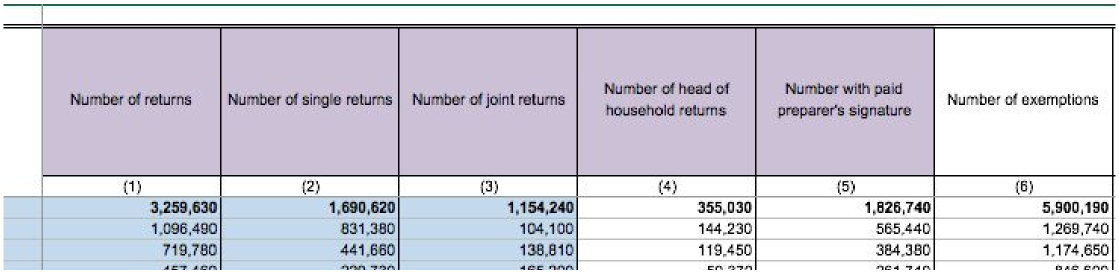
$100,000 under $200,000

$200,000 or more



5						
6			(1)	(2)	(3)	(4)
7	00000	**Total**	**3,259,630**	**1,690,620**	**1,154,240**	
8	00000	$1 under $25,000	1,096,490	831,380	104,100	
9	00000	$25,000 under $50,000	719,780	441,660	138,810	
10	00000	$50,000 under $75,000	457,460	229,730	165,200	
11	00000	$75,000 under $100,000	299,550	92,670	181,630	
12	00000	$100,000 under $200,000	486,510	75,260	389,960	
13	00000	$200,000 or more	199,840	19,920	174,540	
14						
15	01001		8,780	4,750	3,030	

The blue box refers to the total values of all zip code areas across the six income divisions.



	Number of returns	Number of single returns	Number of joint returns	Number of head of household returns	Number with paid preparer's signature	Number of exemptions
	(1)	(2)	(3)	(4)	(5)	(6)
	3,259,630	**1,690,620**	**1,154,240**	**355,030**	**1,826,740**	**5,900,190**
	1,096,490	831,380	104,100	144,230	565,440	1,269,740
	719,780	441,660	138,810	119,450	384,380	1,174,650

The purple box and right onwards are the numeric parameters of the data, of which there are 112.

Though our data was obtained from an online source in the form a spreadsheet, it was necessary to clean the dataset itself in order to process more information easily. As previously mentioned, there exists values pertaining to each attribute (unemployment compensation, number of exemptions, etc.) per each zip code; however, these values themselves are replicated each time as across the six different income divisions. In addition to this, the total value is mentioned. Therefore, there are actually seven numerical values pertaining to each attribute that we are focusing on.

For our project, we are only concerning ourselves with the total value per attribute (i.e. summing over separate values for each of the six income divisions). To exclude rows we did not want to consider, we iterated through the dataframe that we loaded in and only considered rows that were a multiple of 8 (this includes the empty row after information pertaining to each zip code as displayed in our examples above), starting with the tenth row since previous entries contained miscellaneous information. Here is an example from our code:

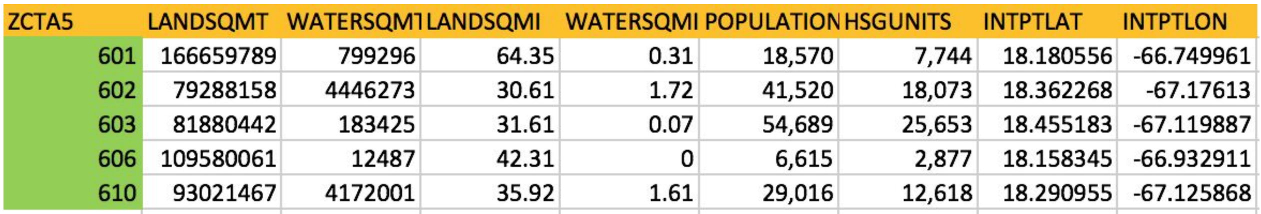
```
arr = []

for i in df2.index:
  mod = (i - 10)%8
  if mod == 0:
     arr.append(df2.iloc[i])
```

In our analysis, we also made sure to not include that contain null values. We excluded these rows once again by iterating through the aforementioned dataframe and columns which contained a “**” value as opposed to an actual numeric number. Here is an example from our code:

```
joint_returns = []
single_returns = []
for i in df.index:
    jr = df['Number of joint returns']
    sr = df['Number of single returns']
for j in jr:
    if j != "**":
        joint_returns.append(float(j))
    #print result[0:10]
for k in sr:
    if k != "**":
        single_returns.append(float(k))
    #print result[0:10]
```

Our second dataset obtained from **ProximityOne is displayed visually below:**



ZCTA5	LANDSQMT	WATERSQMT	LANDSQMI	WATERSQMI	POPULATION	HSGUNITS	INTPTLAT	INTPTLON
601	166659789	799296	64.35	0.31	18,570	7,744	18.180556	-66.749961
602	79288158	4446273	30.61	1.72	41,520	18,073	18.362268	-67.17613
603	81880442	183425	31.61	0.07	54,689	25,653	18.455183	-67.119887
606	109580061	12487	42.31	0	6,615	2,877	18.158345	-66.932911
610	93021467	4172001	35.92	1.61	29,016	12,618	18.290955	-67.125868

Rows: 33,120

Columns: 9

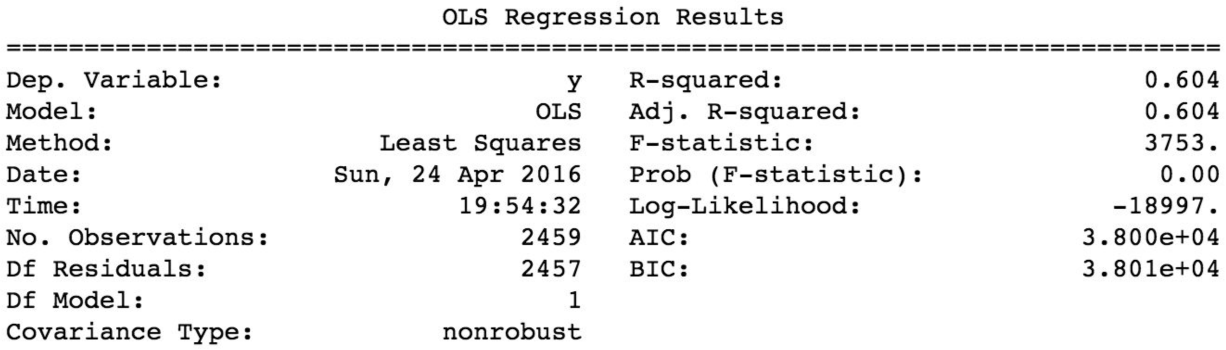
The orange color indicates the attributes and the green color indicates the zip codes. As mentioned previously in our introduction, we are focusing on lines 131 to 668 which indicate Massachusetts zip codes.

For **our first goal**, we compared three regressions:

- total number of dependents vs. joint returns (Linear Regression 1)
- total number of dependents vs. single returns (Linear Regression 2)
- total number of dependents vs. total number of returns (Linear Regression 3)

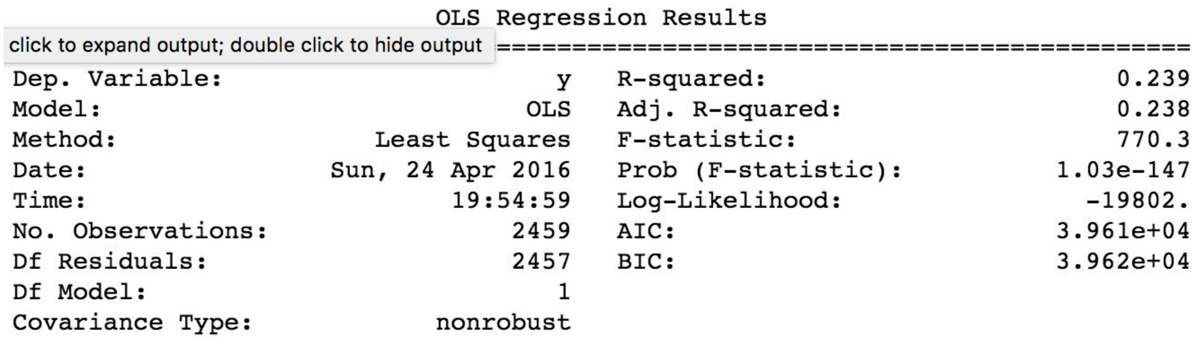
It's interesting to note that the R-squared value for Linear Regression 1 and Linear Regression 3 do not vary much. The value for Linear Regression 1 is 0.599 and the value for Linear Regression 3 is 0.604. A higher correlation exists among these tasks compared to Linear Regression 2, which revealed an R-squared value of 0.239. Therefore, we can conclude that Linear Regression 1 and Linear Regression 3 are correlated, while Linear Regression 2 is not.

Linear Regression 1



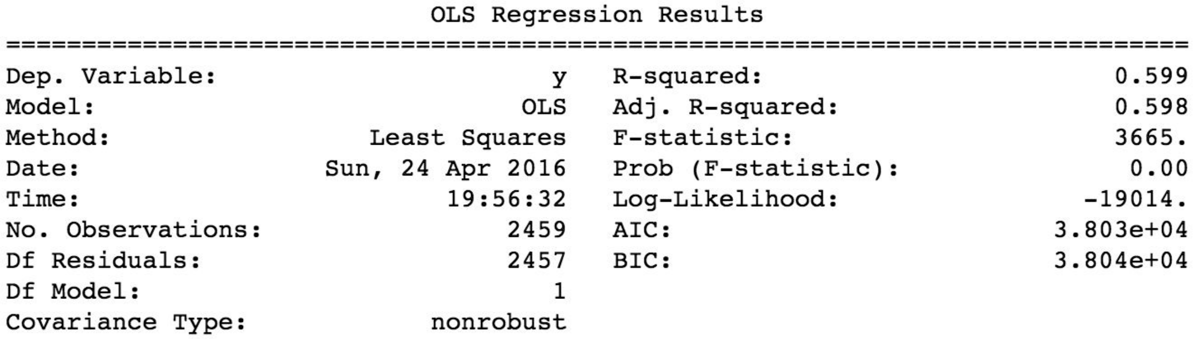
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.604
Model:                            OLS   Adj. R-squared:                  0.604
Method:                 Least Squares   F-statistic:                     3753.
Date:                Sun, 24 Apr 2016   Prob (F-statistic):               0.00
Time:                        19:54:32   Log-Likelihood:                -18997.
No. Observations:                2459   AIC:                         3.800e+04
Df Residuals:                    2457   BIC:                         3.801e+04
Df Model:                           1
Covariance Type:            nonrobust
```

Linear Regression 2



```
                            OLS Regression Results
click to expand output; double click to hide output ===========================
Dep. Variable:                      y   R-squared:                       0.239
Model:                            OLS   Adj. R-squared:                  0.238
Method:                 Least Squares   F-statistic:                     770.3
Date:                Sun, 24 Apr 2016   Prob (F-statistic):          1.03e-147
Time:                        19:54:59   Log-Likelihood:                -19802.
No. Observations:                2459   AIC:                         3.961e+04
Df Residuals:                    2457   BIC:                         3.962e+04
Df Model:                           1
Covariance Type:            nonrobust
```

Linear Regression 3



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.599
Model:                            OLS   Adj. R-squared:                  0.598
Method:                 Least Squares   F-statistic:                     3665.
Date:                Sun, 24 Apr 2016   Prob (F-statistic):               0.00
Time:                        19:56:32   Log-Likelihood:                -19014.
No. Observations:                2459   AIC:                         3.803e+04
Df Residuals:                    2457   BIC:                         3.804e+04
Df Model:                           1
Covariance Type:            nonrobust
```

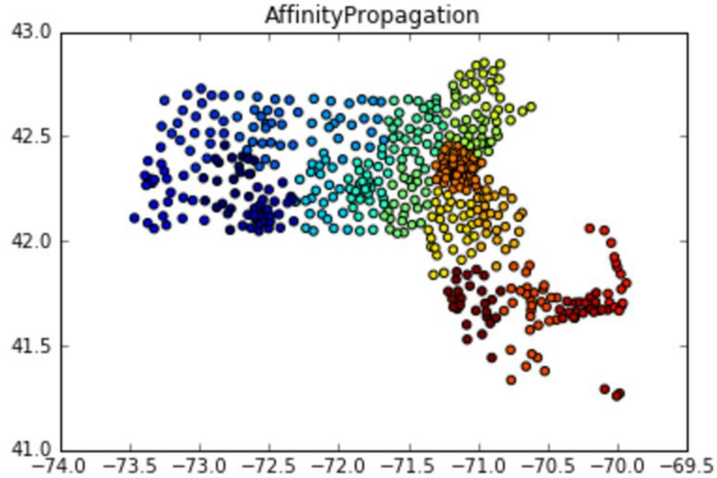
Our second goal was to identify communities of densely populated areas (cities, towns, etc.) vs. sparsely populated areas (rural, suburban, etc). As zip codes were assigned based on the number of people living in

Affinity Propagation was utilised to cluster the data according to the density of zip codes in an area.

Basic Code for Clustering

```
y_pred = sklearn.cluster.AffinityPropagation().fit_predict(clusters)
plt.scatter(x, y, c=y_pred)
```

Below, is the final clustering method we chose:



Below are some of the other clustering techniques we experimented with

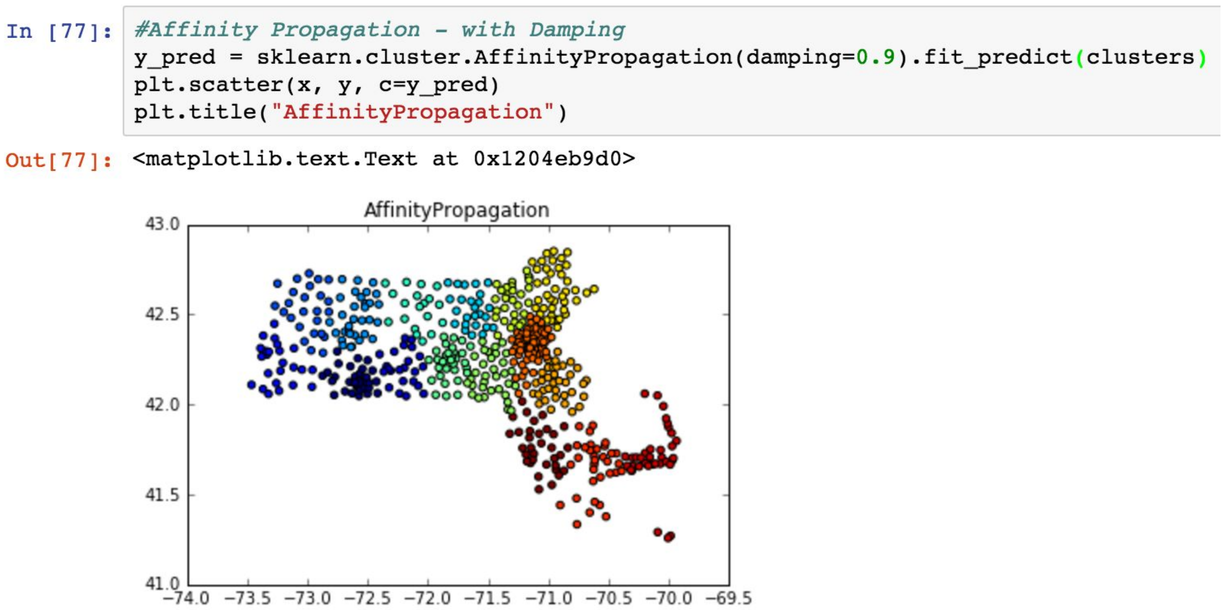
We researched some attributes of Affinity Propagation in order to see if there was a more accurate representation of the differently populated areas in Boston.

To define accuracy of clustering techniques, we measured the distribution of population densities in each cluster using standard deviation. We wanted the zip code areas in each cluster to have similar population densities, thus the lower the standard deviation of the population densities in each defined cluster, the better. Using this method we compared the average standard deviations the different techniques produced and chose those with the lower outcome.

Damping

Damping is commonly needed in over-relaxation methods (eg: loopy belief propagation) where it prevents the availability and responsibility updates from overshooting the solution and leading to oscillations. Since affinity propagation converged, the exact damping level should not have had a significant effect on the resulting net similarity. However, we decided to experiment with different levels of damping to test if this was the case.

Below, is the Affinity Propagation with a damping of 0.9



```
In [77]: #Affinity Propagation - with Damping
         y_pred = sklearn.cluster.AffinityPropagation(damping=0.9).fit_predict(clusters)
         plt.scatter(x, y, c=y_pred)
         plt.title("AffinityPropagation")

Out[77]: <matplotlib.text.Text at 0x1204eb9d0>
```

The resulting technique produced 16 clusters that performed worse. Observationally, the more densely populated areas are more defined in the original than in the graph above. Analytically, when we compared population densities of the new clusters to the original, the average standard deviation was 1.6 times the original - indicating that above clusters were less accurate. Due to the unwanted influence upon the system that has the effect of restricting its oscillations, it was decided that less damping was more relevant to producing the type of clusters wanted and the lowest value (also the default value) of 0.5 was chosen.

Convergence Iterations

Iterations below 2 made the clustering less accurate. Those equal to 2 and above did not make a difference. As the added run time was insignificant, we decided to go with the default number of 15.

Max Iterations

Iterations below 80 made the clustering less accurate. Those above 80 did not make a difference. As the added run time was insignificant, we decided to go with the default number of 200.

Preference

Points with larger values of preferences are more likely to be chosen as exemplars. The number of exemplars - which would indicate the number of clusters - is influenced by the input preferences value. The default of this attribute is for the algorithm to choose the median of the input similarities - which has produced the best results so far (despite our efforts to map the middle of Massachusetts urban and rural areas).

Affinity

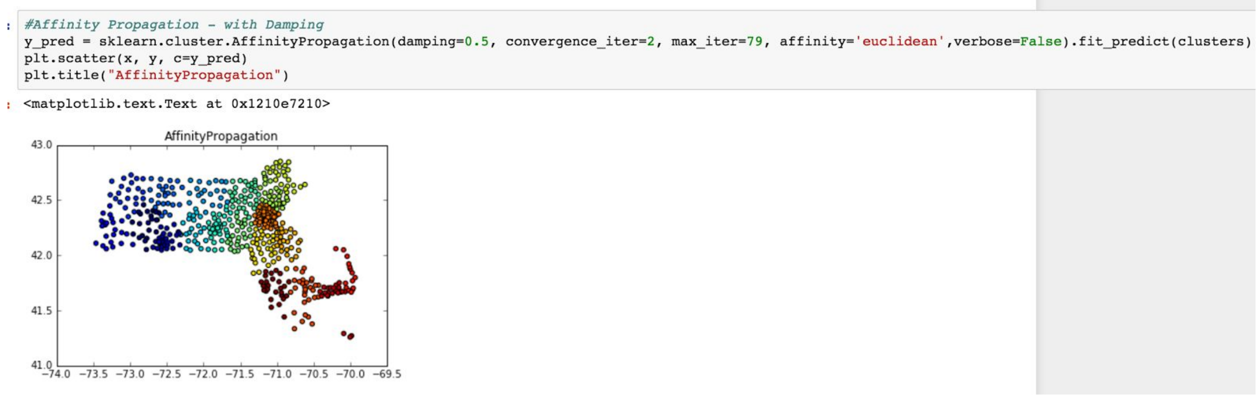
Since parameters of "Latitude" and "Longitude" are used, **Euclidean** was the best option and also the default option.

Verbose
Did not make a significant difference when set to True other than let us know that it "converged after 79 iterations".

All tests with Affinity Propagation are available in our code - Question_2.ipynb

Overall, the default options produced the best representation of clusters as filling in some attributes according to the above findings gave us the exact same graph:

```
y_pred = sklearn.cluster.AffinityPropagation(damping=0.5, convergence_iter=2, max_iter=79,
affinity='euclidean',verbose=False).fit_predict(clusters)
plt.scatter(x, y, c=y_pred)
```



```
#Affinity Propagation - with Damping
y_pred = sklearn.cluster.AffinityPropagation(damping=0.5, convergence_iter=2, max_iter=79, affinity='euclidean',verbose=False).fit_predict(clusters)
plt.scatter(x, y, c=y_pred)
plt.title("AffinityPropagation")
```

```
<matplotlib.text.Text at 0x1210e7210>
```

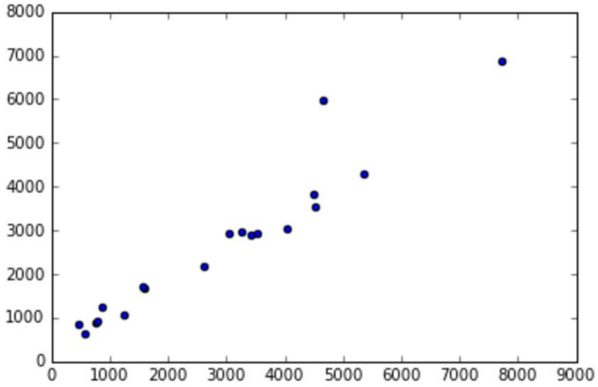
Our third goal was to discover whether certain communities produce a higher number of dependents or number of joint returns than others and whether a community's ratio of dependents and joint returns makes it economically stable or not.

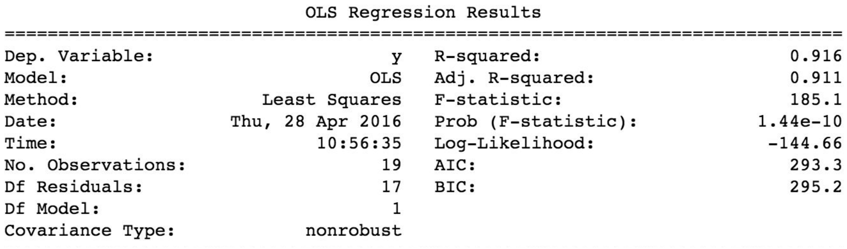
Statistical Analysis of our results:
For dependents:
Mean: 2870.13173884
StdDev: 1924.61205187
Median: 3039.56521739

For joint returns:
Mean: 2653.46550744
StdDev: 1689.61767721
Median: 2892.5

When we tried to plot the dependents of each community against the joint returns, we got an exceptionally correlated result:

```
<matplotlib.collections.PathCollection at 0x11ce40f10>
```



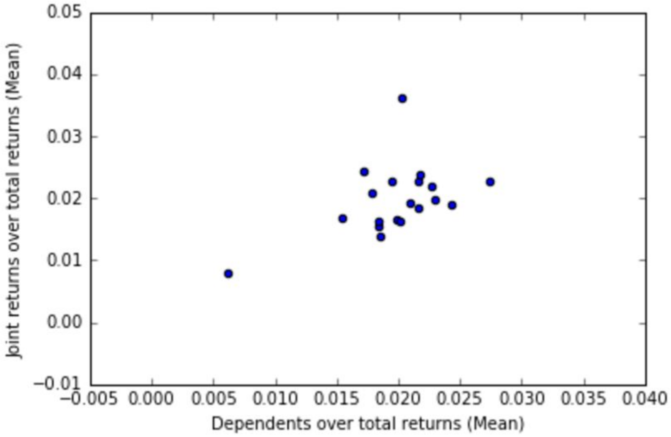


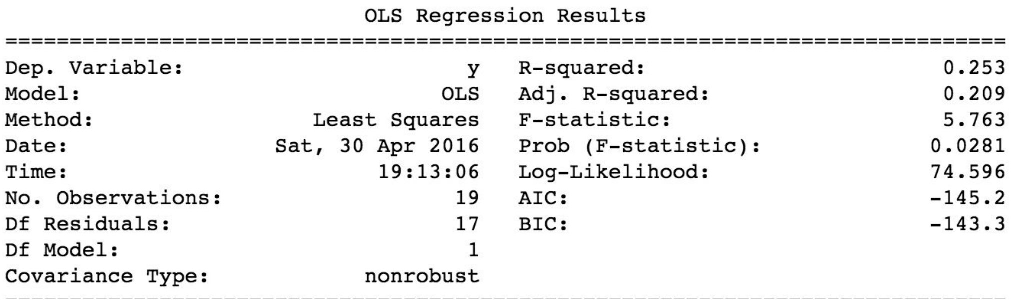
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.916
Model:                            OLS   Adj. R-squared:                  0.911
Method:                 Least Squares   F-statistic:                     185.1
Date:                Thu, 28 Apr 2016   Prob (F-statistic):           1.44e-10
Time:                        10:56:35   Log-Likelihood:                -144.66
No. Observations:                  19   AIC:                             293.3
Df Residuals:                      17   BIC:                             295.2
Df Model:                           1
Covariance Type:            nonrobust
```

Correlation Coefficient = 0.916

We speculated that the unusually high correlation might be a byproduct of the influence of the population size on each community. As a result, we attempted to eliminate this factor by dividing each point of both parameters by their respective populations. Then we plotted:

- Joint Returns over Total Returns (Mean)
- Dependents over Total Returns (Mean)





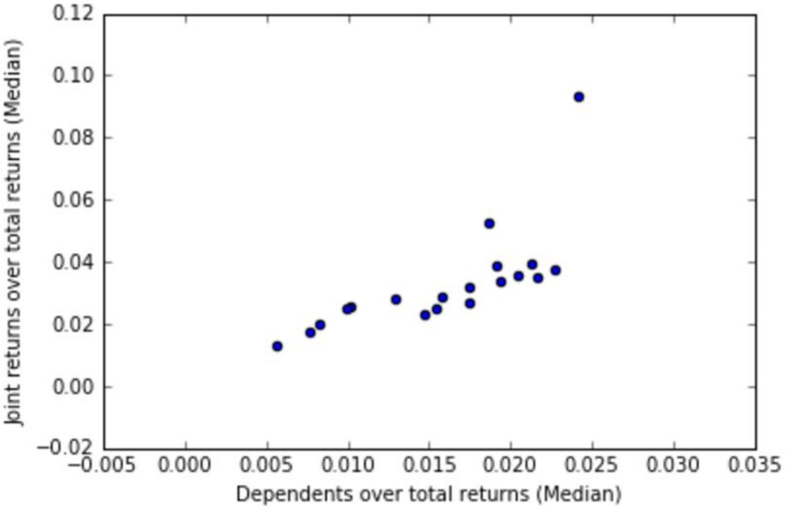
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.253
Model:                            OLS   Adj. R-squared:                  0.209
Method:                 Least Squares   F-statistic:                     5.763
Date:                Sat, 30 Apr 2016   Prob (F-statistic):             0.0281
Time:                        19:13:06   Log-Likelihood:                 74.596
No. Observations:                  19   AIC:                            -145.2
Df Residuals:                      17   BIC:                            -143.3
Df Model:                           1
Covariance Type:            nonrobust
```

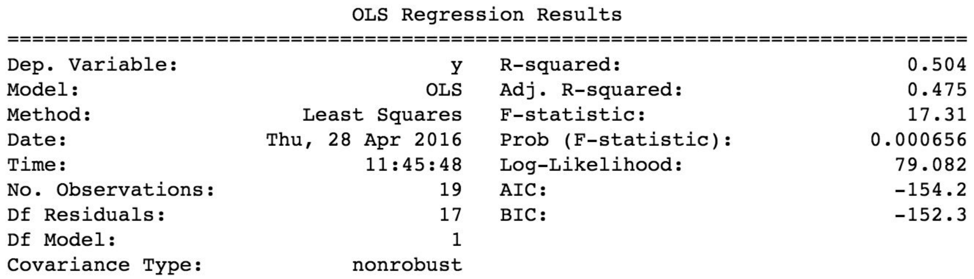
Correlation Coefficient = 0.253

Slight but insignificant correlation.

To test all avenues, we attempted to plot Median as well and found that is resulted in a higher correlation coefficient. We attributed this to median being more accurate in the presence of outliers

```
<matplotlib.text.Text at 0x11e2bab10>
```





```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.504
Model:                            OLS   Adj. R-squared:                  0.475
Method:                 Least Squares   F-statistic:                     17.31
Date:                Thu, 28 Apr 2016   Prob (F-statistic):           0.000656
Time:                        11:45:48   Log-Likelihood:                 79.082
No. Observations:                  19   AIC:                            -154.2
Df Residuals:                      17   BIC:                            -152.3
Df Model:                           1
Covariance Type:            nonrobust
```

A higher correlation Coefficient of 0.504 which indicates that the relationship between joint returns and dependents expanded to the identified communities.

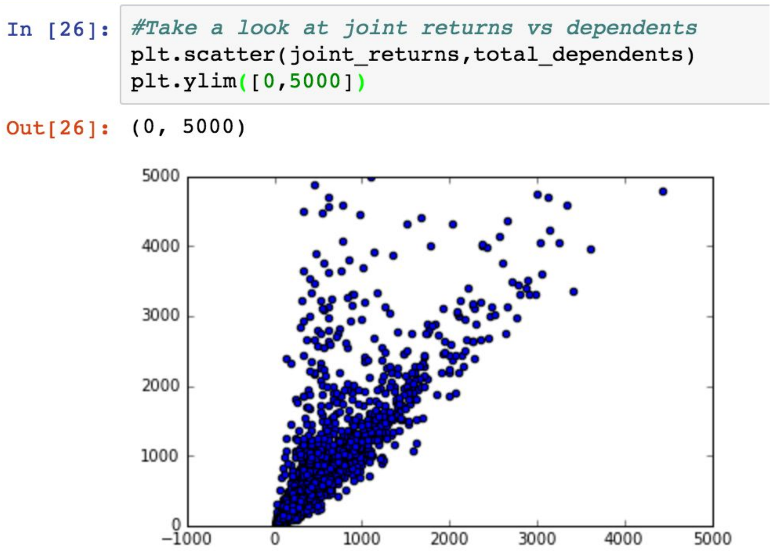
Communities with a low number of joint returns (signifying a low number of married couples) and a high number of dependents correlate with aging areas in Massachusetts. To identify potentially economically unstable communities, we set a benchmark of the mean number of dependents and the mean number of joint returns per community. Then defined economically unstable communities as those that fell below the mean number of joint returns but above the mean number of dependents. The results are discussed in the section below.

Results and Discussions

Our first goal was to determine whether or not a correlation exists between the number of dependents and the number of joint returns.

We expected there to be a correlation between dependents and joint returns as joint tax returns are typically filed by married couples. We assumed that married couples were more likely to have children (aka dependents) and thus a zip code with more joint returns filed should have more dependents as well.

By plotting the number of joint returns vs. the number of dependents, we got the plot below:



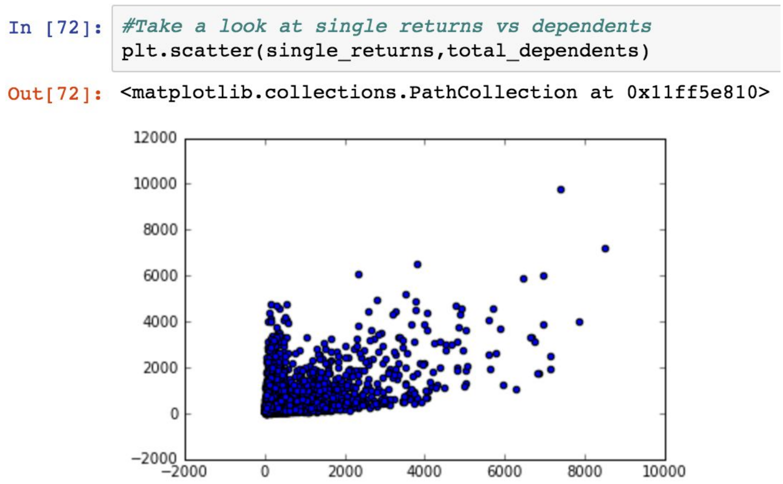
```
In [26]: #Take a look at joint returns vs dependents
         plt.scatter(joint_returns,total_dependents)
         plt.ylim([0,5000])

Out[26]: (0, 5000)
```

The graph shows a slight correlation between these two parameters and this is supported by the OLS Regression Results (available in our Question_1.ipynb file) that indicate an R square value of 0.604.

This means that there is a significant relationship between the aforementioned parameters which indicates that married couples contribute to the number of dependents in an area. However, it is possible that the correlation might simply exist because a greater number of people would result in a greater number of dependents and that both joint returns and dependents are correlated because they are caused by the same factor but are not necessarily related to each other.

To determine whether this was the case or not, we decided to look at the other related factors to see if there was a similar trend. First we plotted the number of single returns vs the number of dependents:



```
In [72]: #Take a look at single returns vs dependents
         plt.scatter(single_returns,total_dependents)
```

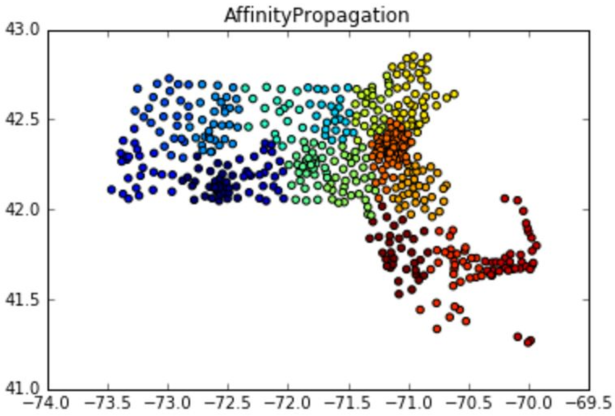
```
Out[72]: <matplotlib.collections.PathCollection at 0x11ff5e810>
```

While there is a slight upward trend, the graph is more spread out than the previous one indicating a weaker correlation if any. This is supported by the OLS Regression Results (available in our Question_1.ipynb file) that indicate an R square value of 0.239.

The above graph illustrates that a higher number of taxes filed does not necessarily produce a higher number of dependents. As a result, the perceived connection between joint returns and dependents is valid.

To elaborate further, it should be noted that there is a small section between 0 and ~1000 of the x-axis where there is a steep coagulated rise in the number of dependents. When looking at only this part of the graph, it would indicate that areas with a low number of single tax returns tend to have a wide range of dependents. For areas with low single tax returns but a high number of dependents, it is assumed that the high number of dependents is the result of a greater proportion of married couples living in the area.

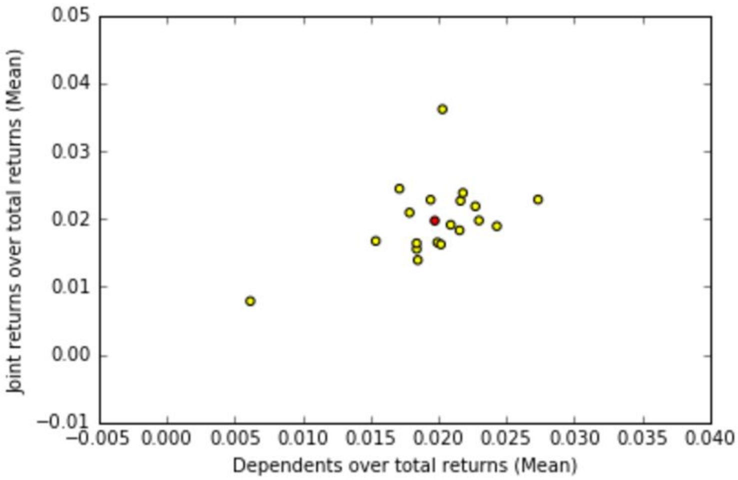
Our second goal was to identify communities of densely populated areas (cities, towns, etc.) vs. sparsely populated areas (rural, suburban, etc). As zip codes were assigned based on the number of people living in the area and were not based on the land area square meter.



The clustering method of Affinity Propagation - as discussed in our techniques section - was used to generate the above graph as well as the clusters we subsequently used in pursuing our third goal.

Our third goal was to discover whether certain communities produce a higher number of dependents or number of joint returns than others and whether a community's ratio of dependents and joint returns makes it economically stable or not.

We defined economically unstable communities as those that fell below the mean number of joint returns but above the mean number of dependents. This benchmark is represented by the Red Dot in the following graph.



This resulted in 5 communities (lower right from red dot) being identified which were all urban areas and generally were next to large water bodies such as lakes and even the Massachusetts Bay.

- Urban area comprising of Belchertown and Agawam
- Urban area between Worcester and Providence (not inclusive)
- Urban area between Framingham and Brockton (not inclusive)
- Urban area south of Boston, Massachusetts Bay Area
- Urban area North-west of Providence and New Bedford (not inclusive)

The community made up of areas in Cape Cod Bay proved to be an outlier with a very high number of dependents and joint returns. This is because of the disproportionate number of retirement homes and beach houses along the Cape.

Conclusions

- There is a significant correlation between the number of dependents and the number of joint returns.
- This correlation extends to different types of communities to a lesser but significant extent.
- Communities with a low number of joint returns and a high number of dependents correlate with aging areas in Massachusetts.
- There are only 5 such communities in Massachusetts, all in urban areas (which may also be representative of retirement communities)
- MA cities are still safe from the effects of an ageing demographic

Sources

MA Zip Code Dataset (2013):
https://www.irs.gov/uac/SOI-Tax-Stats-Individual-Income-Tax-Statistics-2013-ZIP-Code-Data-(SOI)
US Zip Code Dataset (2010): http://proximityone.com/cen2010_zcta_dp.htm