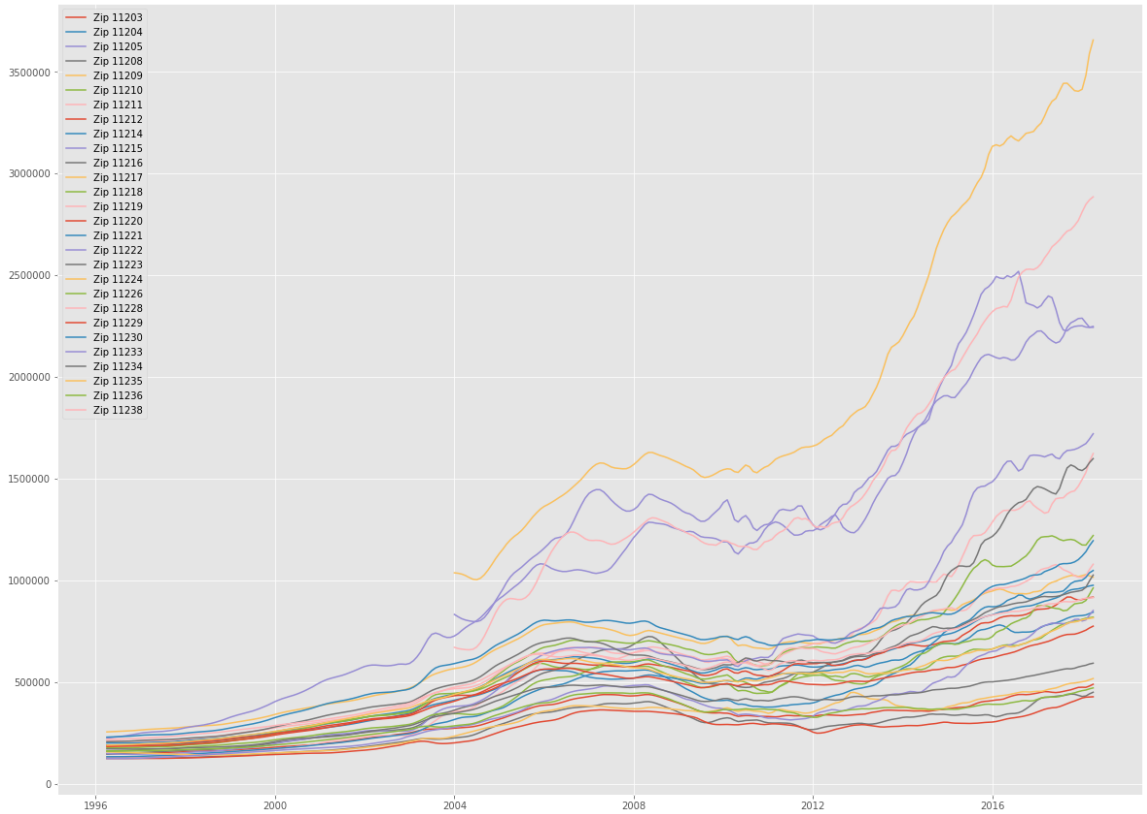
Time Series Mod 4 Project Blog Post

This project consisted of being given a time series dataset with a relatively simple question: what are the top 5 zipcodes to invest in?

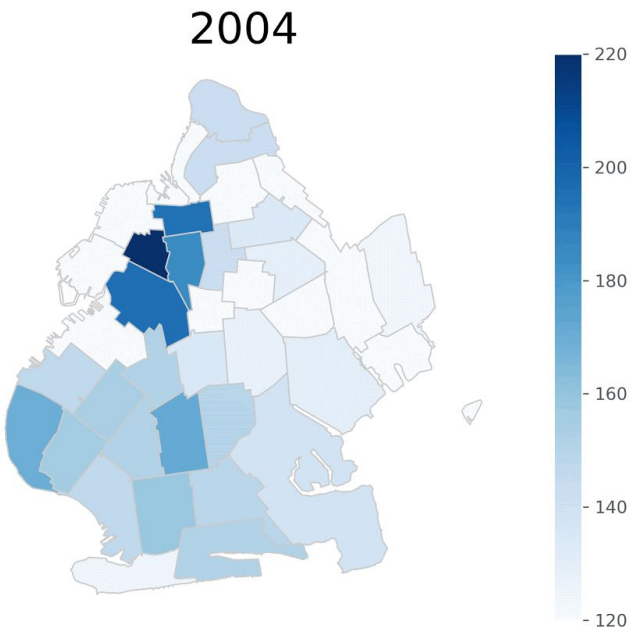
At first glance, the dataset is messy. It is situated so the time is listed as columns, and there seems to be more information than necessary.

	RegionID	RegionName	City	State	Metro	CountyName	SizeRank	1996-04	1996-05	1996-06	...
0	84654	60657	Chicago	IL	Chicago	Cook	1	334200.0	335400.0	336500.0	...
1	90668	75070	McKinney	TX	Dallas-Fort Worth	Collin	2	235700.0	236900.0	236700.0	...
2	91982	77494	Katy	TX	Houston	Harris	3	210400.0	212200.0	212200.0	...
3	84616	60614	Chicago	IL	Chicago	Cook	4	498100.0	500900.0	503100.0	...
4	93144	79936	El Paso	TX	El Paso	El Paso	5	77300.0	77300.0	77300.0	...
5	91733	77084	Houston	TX	Houston	Harris	6	95000.0	95200.0	95400.0	...
6	61807	10467	New York	NY	New York	Bronx	7	152900.0	152700.0	152600.0	...
7	84640	60640	Chicago	IL	Chicago	Cook	8	216500.0	216700.0	216900.0	...

I imported the data, but immediately decided to slim it down to just be Brooklyn(bias!).
Next, I wanted to reshape it from Wide to Long format. For this, I used the melt function, changed the time columns (now rows) to datetime and set them as the index.
My initial graph of the data shows a large skew of the zipcodes as seen below:



After seeing the lack of data for some zip-codes pre 2004, I decided to make the dataset from 2004 onwards. thought it would be fun to explore a visual representation of the data beyond that of the graph above. I decided to create a gif of the data relative to itself over time. For this, I merged public GIS data with my dataset, grouped by year, and then graphed it for each year. To create the gif I used a packaged called imageio, which allowed me to use each year map as a frame to create a gif.



After visually looking at the data in a few ways, I came up with the method of calling an ARIMA model on each zipcode, and using the predictions of that model to look at property value over time trends. I figured, since I would be looking zip-code by zip-code, that writing functions for all of the operations I was going to be performing would be cleaner and simpler in the long run. I decided to write the following functions:

- **sampling:** this samples from the dataset by a provided zipcode. This only returns the median $ values and keeps the time index. We also resample at the Month to avoid errors, (even though the data is already monthly)
- **pdqz:** this is a PDQ parameter function. It uses the AICAIC in order to determine the best p, d, and q values. It takes in a dataframe of time and values. It returns p, which has the attributes, pdq, pdqd, and aic. The aic value is the second minimum of all aic values tested. Passing the right PDQs accounts for stationarity in the model in the future.
- **Arimamodel:** this performs an ARIMA model on the dataset. This function takes in the dataframe, and the results of the AIC pdqs, and applies them to an arima model. It returns the output of the model.
- **MSEr:** Mean Squared Error. This function takes in the dataframe, the results of the ARIMA, and the date at which you want the ARIMA to start. It returns the mean squared error of the ARIMA predictions, vs the actual data.
- **Future_value:** This function takes in the dataframe, the results of the ARIMA, and the number of steps into the future. It returns the mean value at the most recent step, the upper bound, and the lower bound of the confidence interval
- **Plotting_forecasts:** This function takes in the dataframe, the results of the ARIMA, the amount of time in the future . The start date of your predictions, the x-axis label, and the y-axis label. It returns a graph with the observed and the forecasted results

I tested each function, and after some tweaking, the output of the ARIMA model was normal, with acceptable P values. From there I was able to build a for loop that would run through each of my brooklyn zip codes, create an ARIMA model, and store the results of the predictions in a dataframe. **My first time around on this, I made the mistake of running every zipcode, not just brooklyn ones, and 45 minutes later of it still running I decided to make it a smaller dataset!**

My Results Dataframe was structured with the following parameters:

- Zipcode
- MSE
- Mean value at 24 steps (2 years)
- Value at 5 Years ago
- Value at 10 Years ago
- Lower bound of confidence value at 24 steps
- Upper bound of confidence value at 24 steps
- 10 year Growth (mean value - 10 year value)
- Conservative 10 year growth (lower bound - 10 year value)
- Risky 10 year growth (upper bound - 10 year value)
- 5 year Growth (mean value - 5 year value)
- Conservative 5 year growth (lower bound - 5 year value)
- Risky 5 year growth (upper bound - 5 year value)

My method for choosing the top five zipcodes to invest in was to look at all of the growth parameters for both five and ten year growth. There was immediately a distinct pattern: for both 5 and 10 year growth, the best zipcodes to invest in remained the same, across conservative and risky growth. Overall, consistently 11217, 11238, 11211, 11216, and 11222 as the top 5 zipcodes to invest in. These correspond with Boreum Hill, Prospect Park, Williamsburg (my home!) Bed-Stuy, and Greenpoint. This is consistent with my local knowledge of the areas, as they're known to be places with an increasing younger demographic looking for places with more square footage for the $. Seeing these trends in these places would be great if you were a home-owner or land-lord (or lived in a rent-controlled apartment), but isn't so great for regular renters like me! There are a few other players, (11223, 11215, and 11230) which could also be considered, however this would be more dependent on investment preferences.