Real Estate Property Investments

Invest with sound, objective data driven recommendations

Syracuse Applied Data Science, IST-707 Data Analytics

Ryan Timbrook (RTIMBROO)
DATE: 9/8/2019 ASSIGNMENT: Final Project

1. Introduction

A real estate transaction can be an emotional time for everyone. The complexities between buyers and sellers are the result of different experiences and expectations. Success in today's market is guided by knowledge, communication, and partnership.

Buyers are waiting later in life to purchase their first home. They have very specific expecations on what they are looking for, and willing to take the time to get exactly what they want. To be successful, buyers will turn to experienced professionals to guide them through the buying process and to sift through the voluminous of data.

Sellers past experiences have been rooted in market conditions significantly different than we aree seeing today. Many are resisting the realities of the market and are slow to react to the valuable feedback the data provides. To be successful, sellers will need to utilize skilled professionals to interpret the specifics of today's market and take swift action to adjust for changing trends.

1.1 Problem Statement:

- How to predict a low risk / high yield return on property investment in a volatile market.
- Where and when to buy and sell that maximizes investment profits.
- Forecast future growth and decline of a region in order to guide investors with optimized, data driven, recommendations.

```
=================================================================================
```

1.2 About the Data

Base Real Estate data provided by: [Zillow (files.zillowstatic.com/research/public/Zip/Zip_Zhvi_SingleFamilyResidence.csv)]
Base Federal Reserve data provided by: kaggle (https://www.kaggle.com/federalreserve/interest-rates
 Base Economic data sets provided by: [datahub.io](https://datahub.io/core/cpi-us![image.png]
(attachment:image.png)

Zillow Data: Timeseries Real Estate data by ZipCode U.S.
Zillow Home Value Index (ZHVI): A smoothed, seassonally adjusted measure of the median estimated home value across a given region and housing type. It is a dollar-denominated alternative to repeat-sales indices.

- Zip_Zhvi_SingleFamilyResidence.csv
- Zip_Zhvi_AllHomes.csv
- Zip_MedianRentalPricePerSqft_Sfr.csv
- Zip_MedianRentalPrice_AllHomes.csv
- Zip_MedianListingPrice_AllHomes.csv

Datahub.io: U.S., National Yearly Economic Reports

- interest_rates.csv
  - Inflation, GDP deflator (annual %) and Inflation, consumer prices (annual %) for most countries in the world when it has been measured. Data The data comes from The World Bank (CPI), The World Bank (GDP) and is collected from 1973 to 2014. There are some values missing from data
- inflation-consumer.csv
- inflation-gdp.csv
- education_budget_data.csv
  - United States of America Education budget to GDP analysis Data Data comes from Office of Management and Budget, President's Budget from white house official
- population.csv
  - Population figures for countries, regions (e.g. Asia) and the world. Data comes originally from World Bank and has been converted into standard CSV
- investor_flow_funds_monthly.csv
  - Monthly net new cash flow by US investors into various mutual fund investment classes (equities, bonds etc). Statistics come from the Investment Company Institute (ICI)
- housing_price_cities.csv
  - Case-Shiller Index of US residential house prices. Data comes from S&P Case-Shiller data and includes both the national index and the indices for 20 metropolitan regions. The indices are created using a repeat-sales methodology.
- household-income.csv
  - Upper limits of annual incomes for each fifth and lower limit of income for top 5 percent of all housholds from 1967 to last year Data This dataset is acquired from U.S. Census Bureau, Current Population Survey, Annual Social and Economic Supplements.
- employment.csv
  - US Employment and Unemployment rates since 1940. Official title: *Employment status of the civilian noninstitutional population, 1940 to date* from USA Bureau of Labor Statistics. Data Numbers are in thousands. US Employment and Unemployment rates since 1940 From the USA Bureau of Labor

- cpi.csv
  - Consumer Price Index for All Urban Consumers (CPI-U) from U.S. Department Of Labor Bureau of Labor Statistics. This is a monthly time series from January 1913. Values are U.S. city averages for all items and 1982-84=100.
- cash-surp-def_csv.csv
  - Repository of the data package of the Cash Surplus or Deficit, in percentage of GDP, from 1990 to 2013. Data Data comes originally from World Bank!
- bonds_yields_10y.csv
  - 10 year nominal yields on US government bonds from the Federal Reserve. The 10 year government bond yield is considered a standard indicator of long-term interest rates.
- gdp_quarter.csv
- gdp_year.csv
  - Gross Domestic Product (GDP) of the United States (US) both nominal and real on an annual and quarterly basis. Annual data is provided since 1930 and quarterly data since 1947. Both total GDP (levels) and annualized percentage change in GDP are provided.

Dataset Info: Economic

- The Time series data range our modeling and analysis was centered on was from **1997 through 2018**. All of the Realestate datasets achieved this desired range, however some of the Economic datasets did not. To achieve paraty and have a fuller dataset for baseline testing, time series future forecast methods were applied. More will be described in section 2 on Time Series forecasting.
- GDP Yearly: Forecasted for 2016, 2017, 2018 values
- Inflation: Forecasted for 2017, 2018 values
- Interest Rates: Forecasted for 2016, 2017, 2018
- Note: Kaggel Federal Reserve datasets proved to be useless, full of gaps and limited time series data to provide value. Economic data was pulled from the above mentioned sources and munged together to form a more useable data set.

Dataset Info: Real Estate

- This data is our base datasets and provides the core insights into preditable housing market trends given prior knowledge of price performance coupled with economic fluctuations. Timeseries prediction models are created for each type of housing dataset mentioned above by ZipCode and it's monthly price value from 1997 to 2018. For this initial analysis, ZipCode's were focused to the U.S. State of Washington. This represents 351 unique zipcodes that were modeled with a five year future price prediction. These zipcodes then were combined with the economic features above, in order to create a dataset that could be used in identifing and or predicting events that could have a positive or negative impact on housing prices given a unique zipcode.

```
All the files are downloaded
```

1.3 Obtain the data

- Using the base data available from [Zillow (files.zillowstatic.com/research/public/Zip/Zip_Zhvi_SingleFamilyResidence.csv)](files.zillowstatic.com/research/public/Zip/Zip_Zhvi_SingleFamilyResidence.csv)

Zillow Home Value Index (ZHVI): A smoothed, seassonally adjusted measure of the median estimated home value across a given region and housing type. It is a dollar-denominated [alternative to repeat-sales indices (https://wp.zillowstatic.com/3/ZHVI-InfoSheet-04ed2b.pdf)](https://wp.zillowstatic.com/3/ZHVI-InfoSheet-04ed2b.pdf).

- OBTAIN Interest Rates data from Kaggel
  - Using the dataset provided by the kaggel [Federal Reserve Interest Rates (https://www.kaggle.com/federalreserve/interest-rates/downloads/interest-rates.zip/1)](https://www.kaggle.com/federalreserve/interest-rates/downloads/interest-rates.zip/1)
- Obtain Economic Data from datahub.io `(https://datahub.io/core/cpi-us![image.png](attachment:image.png))`

Out[46]:

	date	level-current	level-chained	change-current	change-chained
0	1930	92.2	966.7	-16.0	-6.4
1	1931	77.4	904.8	-23.1	-12.9
2	1932	59.5	788.2	-4.0	-1.3
3	1933	57.2	778.3	16.9	10.8
4	1934	66.8	862.2	11.1	8.9

Out[45]:

	date	level-current	level-chained	change-current	change-chained
0	1947-04-01	246.3	1932.3	6.4	-0.4
1	1947-07-01	250.1	1930.3	17.3	6.4
2	1947-10-01	260.3	1960.7	9.3	6.0
3	1948-01-01	266.2	1989.5	10.5	6.7
4	1948-04-01	272.9	2021.9	10.0	2.3

Out[76]:

	Year	Month	Day	Federal Funds Target Rate	Federal Funds Upper Target	Federal Funds Lower Target	Effective Federal Funds Rate	Real GDP (Percent Change)	Unemployment Rate	Inflation Rate
339	1982	9	27	10.25	NaN	NaN	NaN	NaN	NaN	NaN
340	1982	10	1	10.00	NaN	NaN	9.71	0.4	10.4	5.9
341	1982	10	7	9.50	NaN	NaN	NaN	NaN	NaN	NaN
342	1982	11	1	9.50	NaN	NaN	9.20	NaN	10.8	5.3
343	1982	11	19	9.00	NaN	NaN	NaN	NaN	NaN	NaN

Out[43]:

	Country	Country Code	Year	Inflation
10559	United States	USA	1961	1.350154
10560	United States	USA	1962	1.244635
10561	United States	USA	1963	1.088386
10562	United States	USA	1964	1.503940
10563	United States	USA	1965	1.919826

Out[42]:

	YEAR	BUDGET_ON_EDUCATION	GDP	RATIO
0	1976	9314.0	1877587.0	0.496
1	1977	10568.0	2085951.0	0.507
2	1978	11625.0	2356571.0	0.493
3	1979	13996.0	2632143.0	0.532
4	1980	15209.0	2862505.0	0.531

Out[41]:

	Country Name	Country Code	Year	Value
14288	United States	USA	1960	180671000.0
14289	United States	USA	1961	183691000.0
14290	United States	USA	1962	186538000.0
14291	United States	USA	1963	189242000.0
14292	United States	USA	1964	191889000.0

Out[37]:

	Date	Total Equity	Domestic Equity	World Equity	Hybrid	Total Bond	Taxable Bond	Municipal Bond	Total
0	2007-01-31	27364	5723	21641	5321	15287	12453	2834	47972
1	2007-02-28	25306	8411	16895	5164	15064	11926	3137	45533
2	2007-03-31	6551	-486	7037	3764	15782	12925	2857	26097
3	2007-04-30	16063	-163	16225	4384	13701	12346	1355	34148
4	2007-05-31	-2876	-14176	11300	4318	20813	17215	3598	22256

Out[39]:

	Date	AZ-Phoenix	CA-Los Angeles	CA-San Diego	CA-San Francisco	CO-Denver	DC-Washington	FL-Miami	FL-Tampa	GA-Atlanta	...
0	1987-01-01	NaN	59.33	54.67	46.61	50.20	64.11	68.50	77.33	NaN	...
1	1987-02-01	NaN	59.65	54.89	46.87	49.96	64.77	68.76	77.93	NaN	...
2	1987-03-01	NaN	59.99	55.16	47.32	50.15	65.71	69.23	77.76	NaN	...
3	1987-04-01	NaN	60.81	55.85	47.69	50.55	66.40	69.20	77.56	NaN	...
4	1987-05-01	NaN	61.67	56.35	48.31	50.63	67.27	69.46	77.85	NaN	...

5 rows × 24 columns

Out[35]:

	Year	Number (thousands)	Lowest	Second	Third	Fourth	Top 5 percent
0	2016	126224	24518	46581	76479	123621.0	230095
1	2015	125819	23591	45020	74498	121060.0	221900
2	2014	124587	22213	42688	70699	116355.0	214100
3	2013	123931	22134	43251	70830	116186.0	216208
4	2013	122952	22029	42358	69039	111631.0	206587

Out[34]:

	year	population	labor_force	population_percent	employed_total	employed_percent	agricult
0	1941	99900	55910	56.0	50350	50.4	
1	1942	98640	56410	57.2	53750	54.5	
2	1943	94640	55540	58.7	54470	57.6	
3	1944	93220	54630	58.6	53960	57.9	
4	1945	94090	53860	57.2	52820	56.1	

Out[31]:

	Date	Index	Inflation
0	1913-01-01	9.8	NaN
1	1913-02-01	9.8	0.00
2	1913-03-01	9.8	0.00
3	1913-04-01	9.8	0.00
4	1913-05-01	9.7	-1.02

Out[32]:

	Country Name	Country Code	Year	Value
0	Afghanistan	AFG	2006	-2.027860
1	Afghanistan	AFG	2007	-1.731230
2	Afghanistan	AFG	2008	-2.314250
3	Afghanistan	AFG	2009	0.281700
4	Afghanistan	AFG	2010	1.495567

Out[33]:

	Date	Rate
0	1953-04-02	2.83
1	1953-05-02	3.05
2	1953-06-02	3.11
3	1953-07-02	2.93
4	1953-08-02	2.95

1.4 Data Exploration - SCRUB - CLEAN - Transform

Clean and perform initial transformations steps of the data

REAL ESTATE DATATSETS - ZILLOW

- Rename 'Region Name' Column to ZipCode
- Convert ZipCode field to string
- Remove columns of non-interest:
  - 'RegionID','SizeRank','City','Metro','CountyName'
  - '1996-04','1996-05','1996-06','1996-07','1996-08','1996-09','1996-10','1996-11','1996-12'
  - '2019-01','2019-02', '2019-03', '2019-04', '2019-05', '2019-06', '2019-07','2019-08','2019-09'
- Fill NaN with median value

Zillow Single Family Residence DataFrame Head:

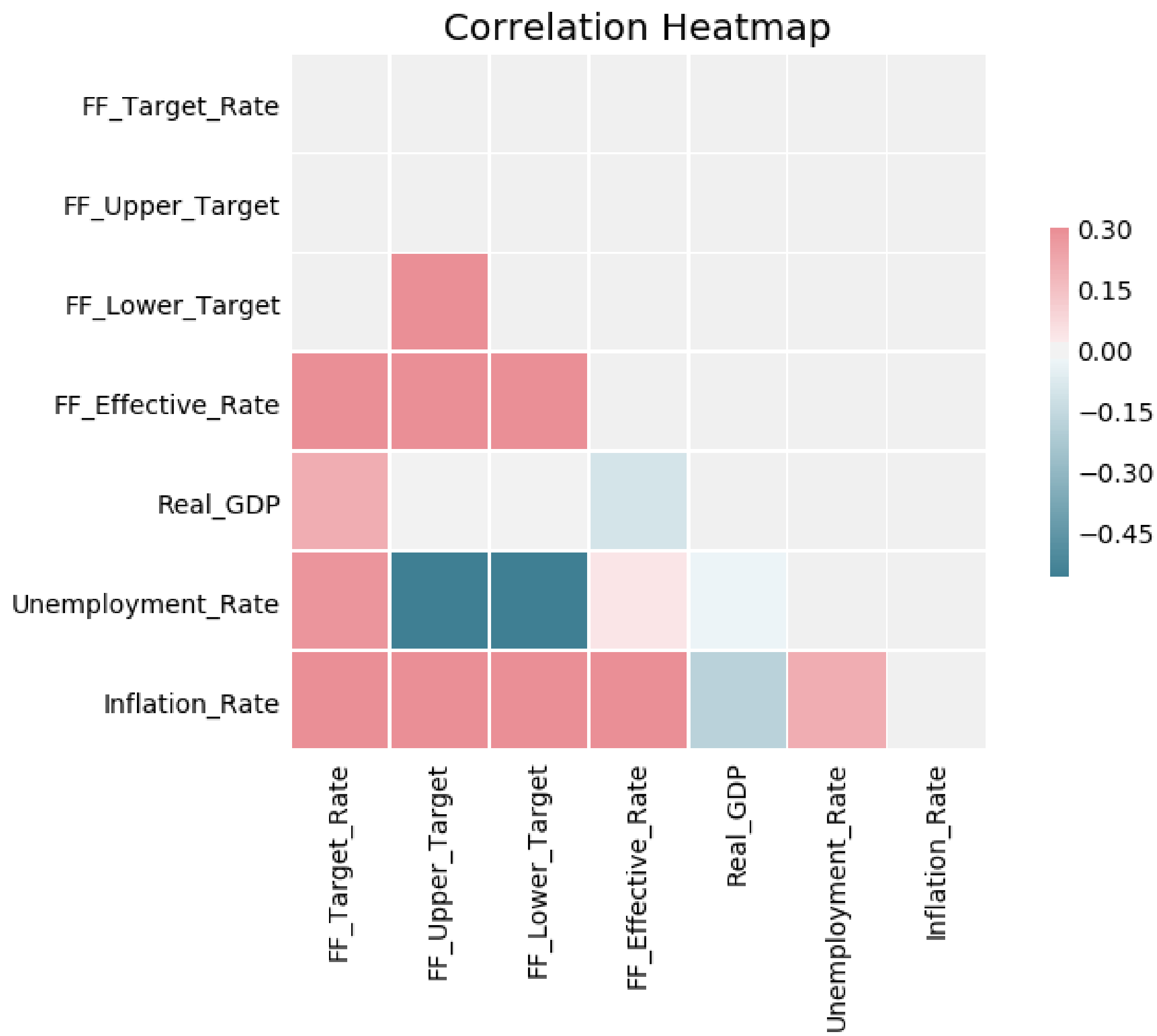
Out[20]:

	ZipCode	State	2018-01	2018-02	2018-03	2018-04	2018-05	2018-06	2018-07	20
68	98052	WA	899700.0	909000.0	909900.0	908600.0	913100.0	916700.0	913900.0	911
137	98012	WA	575800.0	585100.0	594200.0	602400.0	608500.0	612100.0	614100.0	613
159	99301	WA	219800.0	220300.0	219600.0	219500.0	220900.0	223200.0	225600.0	227
171	98103	WA	854600.0	861300.0	862800.0	862200.0	862800.0	860400.0	853800.0	847
301	98682	WA	298900.0	300600.0	302000.0	303100.0	305600.0	308200.0	309700.0	310

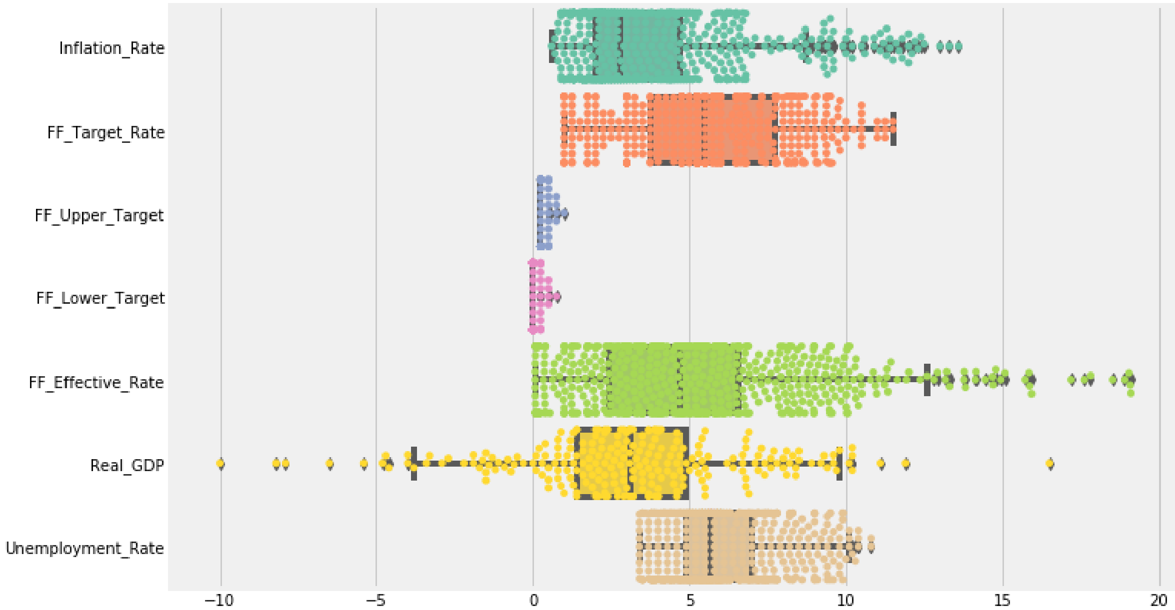
INTEREST RATE DATASET - KAGGEL

- Datasets:
  - Interest Rate:
    - Rename column names to make it easier to work with
    - View the new column names in a correlation heatmap

<Figure size 576x360 with 0 Axes>



A look at the datasets distributions of elements to determin best methods for cleaning the data



ECONOMIC DATASETS - DATAHUB.IO

- Datasets:
  - Interest Rate:
    - keep Year, Month, Federal Funds Target Rate
  - Inflation Consumer:
    - filter on Country = 'United States', keep Year, Inflation - drop the rest
  - GDP Year:
    - Change column names

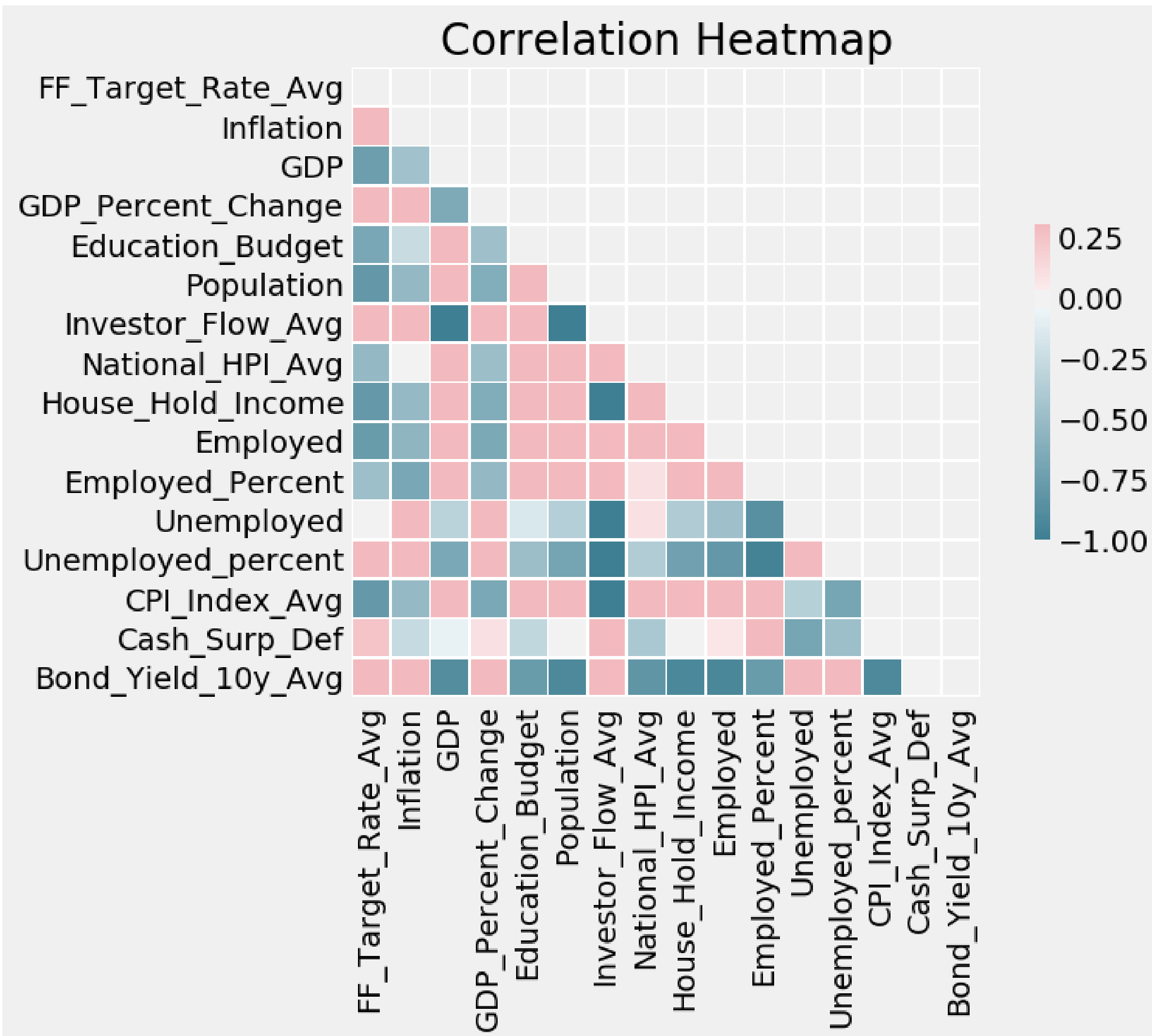
This process continuous for the remainder of the datasets. See accompaning notebook for details.

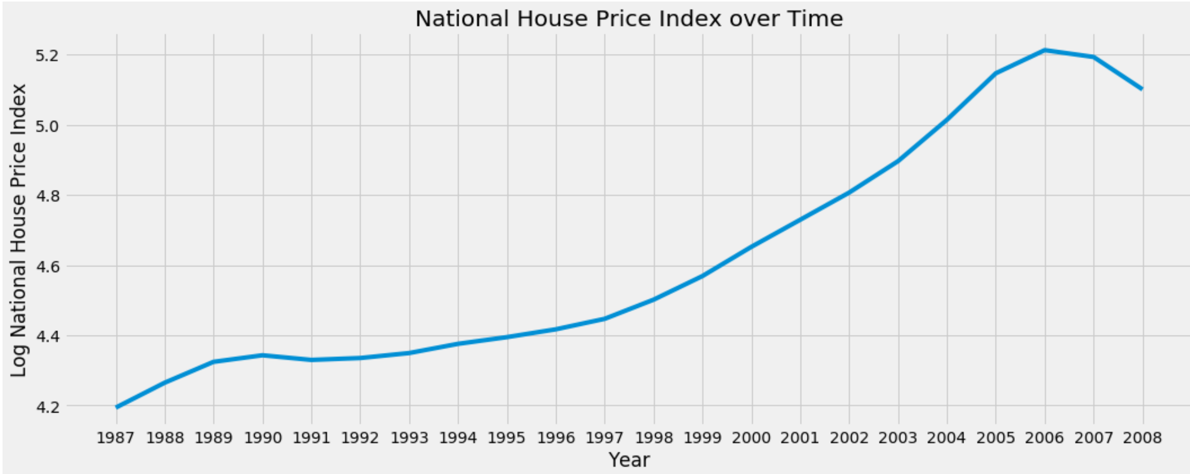
Merged Dataframe of Economic features aggregated from their individual source files

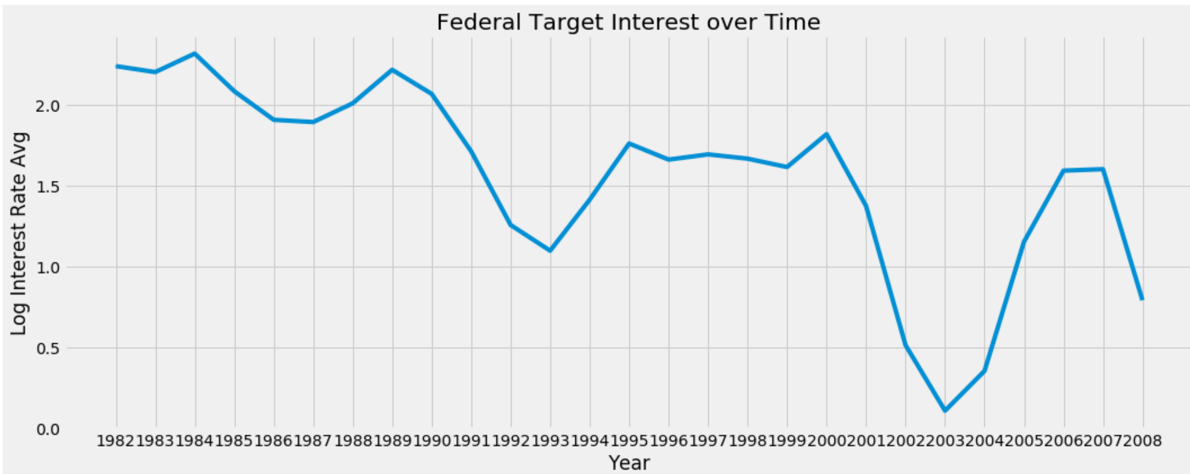
Out[85]:

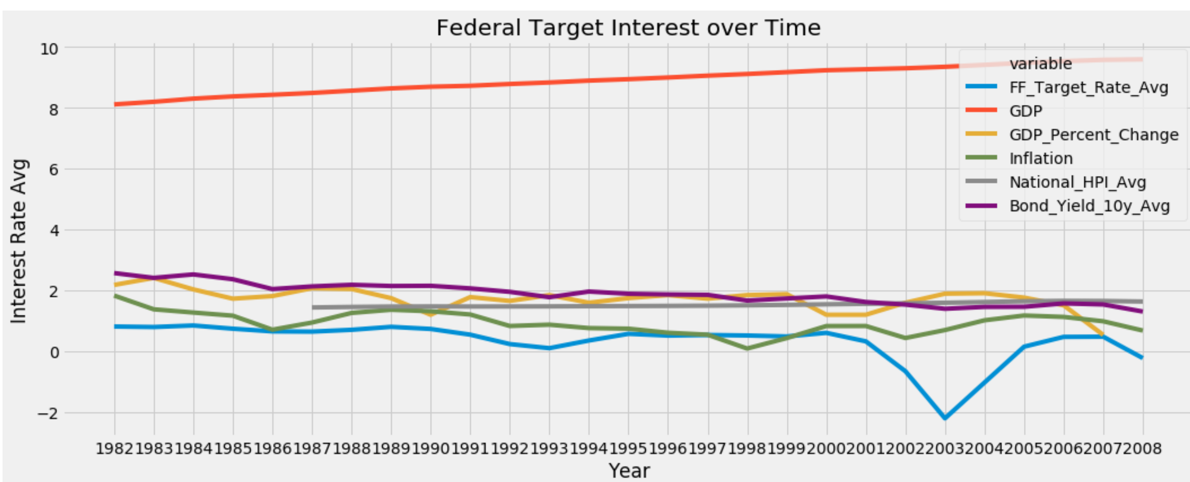
	Year	FF_Target_Rate_Avg	Inflation	GDP	GDP_Percent_Change	Education_Budget	Popu
0	1982	9.392857	6.203740	3345.0	8.8	15374.0	231664
1	1983	9.053125	3.948367	3638.1	11.1	15267.0	233792
2	1984	10.150000	3.548237	4040.7	7.6	15336.0	235825
3	1985	8.044643	3.199612	4346.7	5.6	18952.0	237924
4	1986	6.740132	2.017624	4590.2	6.1	17750.0	240133

Correlation Heatmap of the new Economic Dataset's features

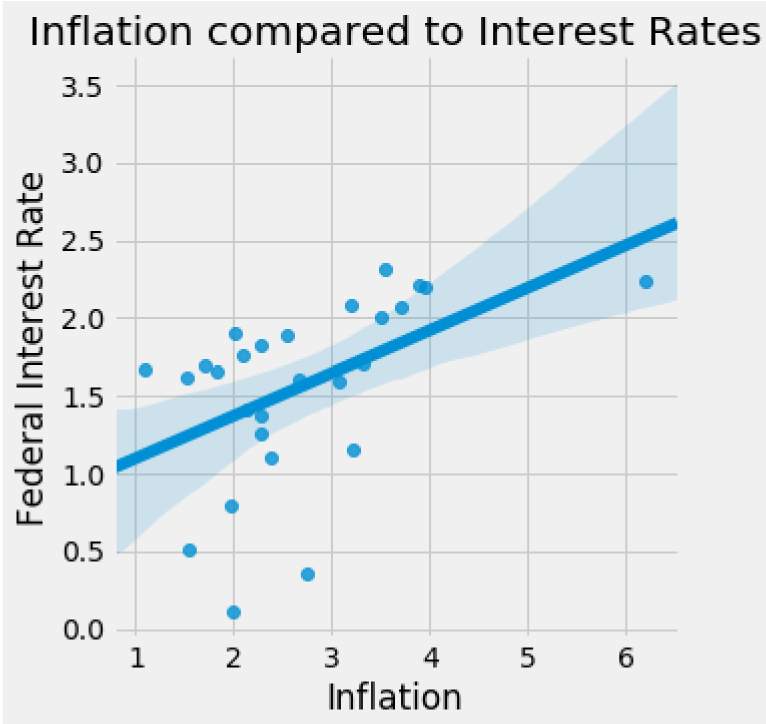



National House Price Index over Time
Log National House Price Index
Year


Federal Target Interest over Time
Log Interest Rate Avg
Year

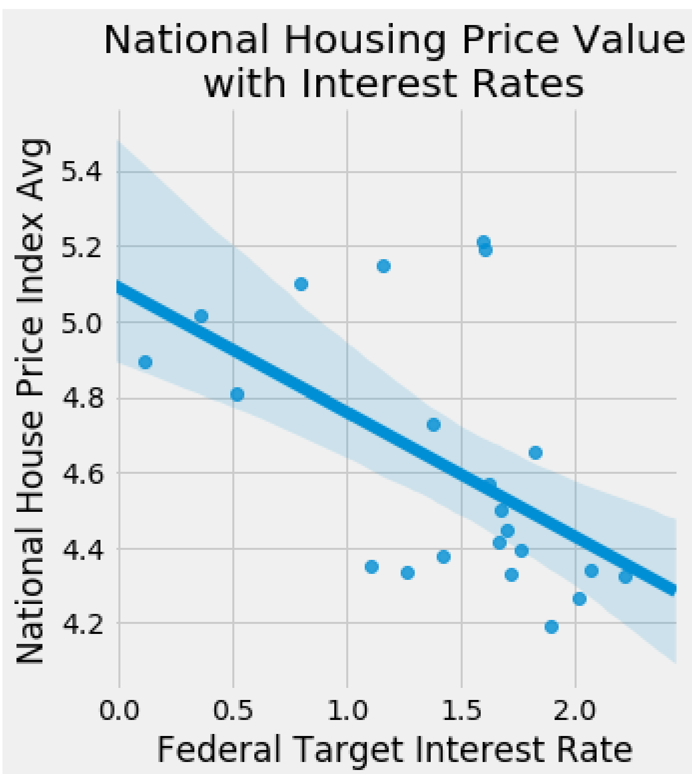

Federal Target Interest over Time
Interest Rate Avg
Year
variable
FF_Target_Rate_Avg
GDP
GDP_Percent_Change
Inflation
National_HPI_Avg
Bond_Yield_10y_Avg

<Figure size 1152x432 with 0 Axes>

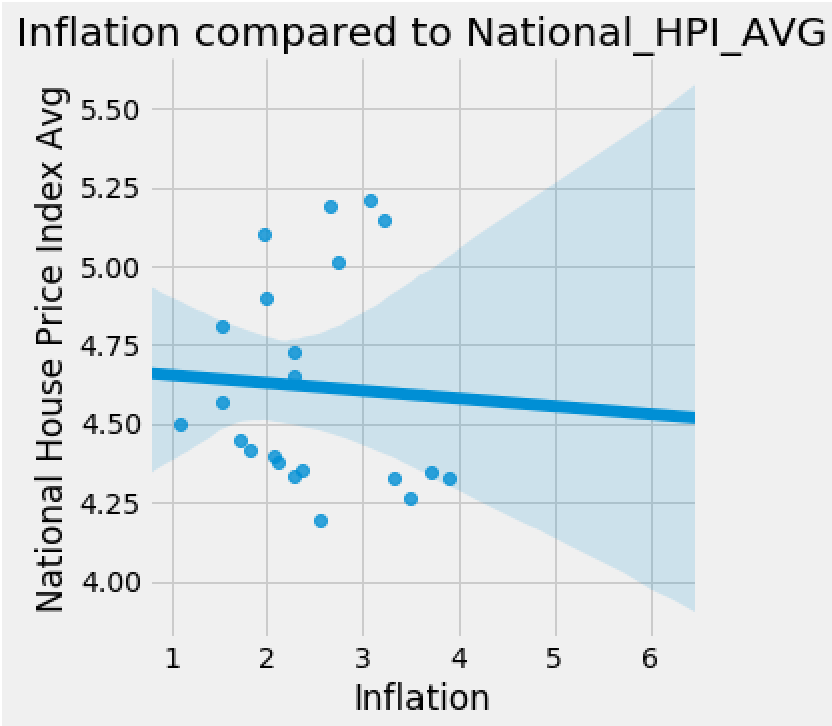


Impacts of Economic Factors on National Housing Price Index Avg

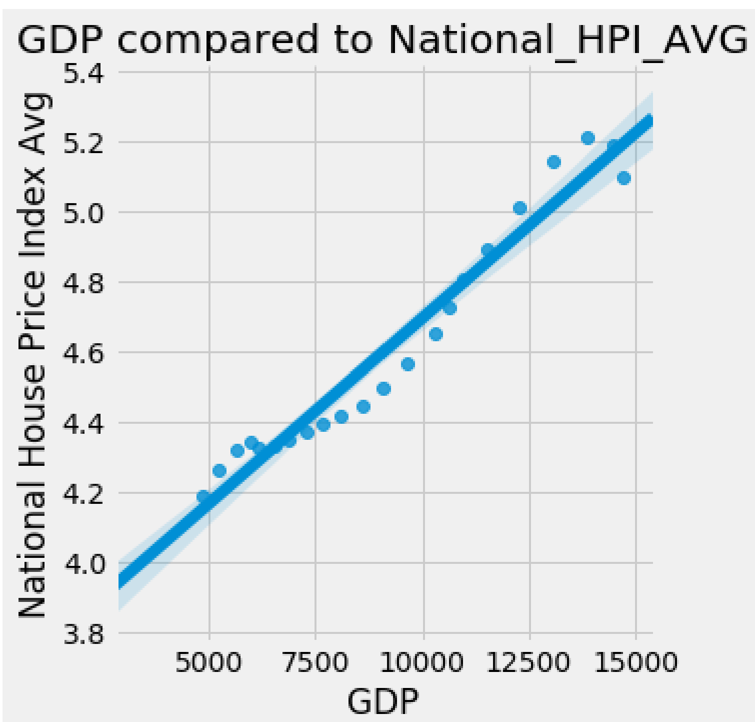
<Figure size 1152x432 with 0 Axes>



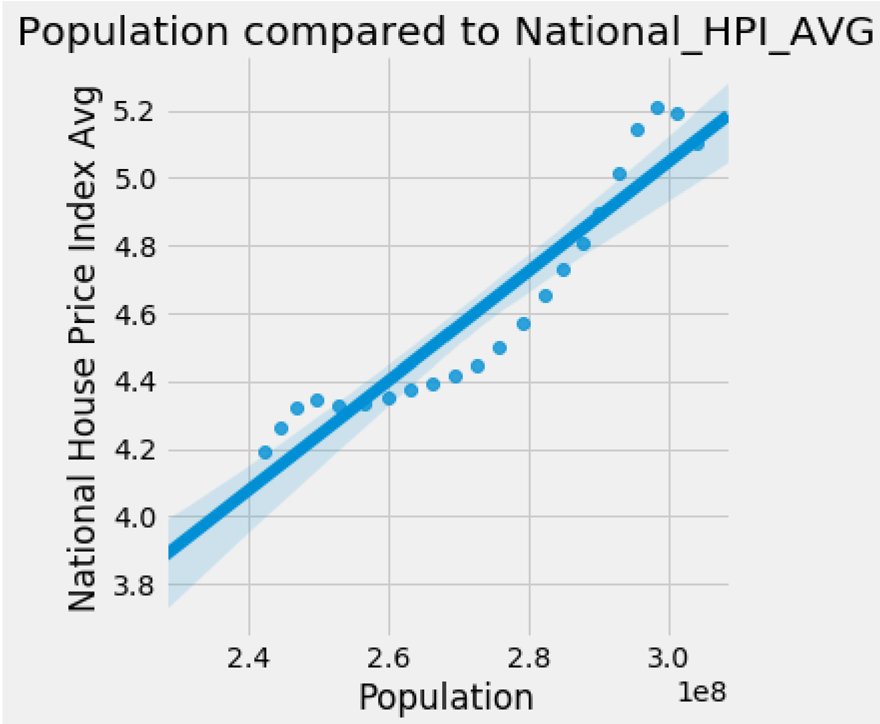
<Figure size 1152x432 with 0 Axes>



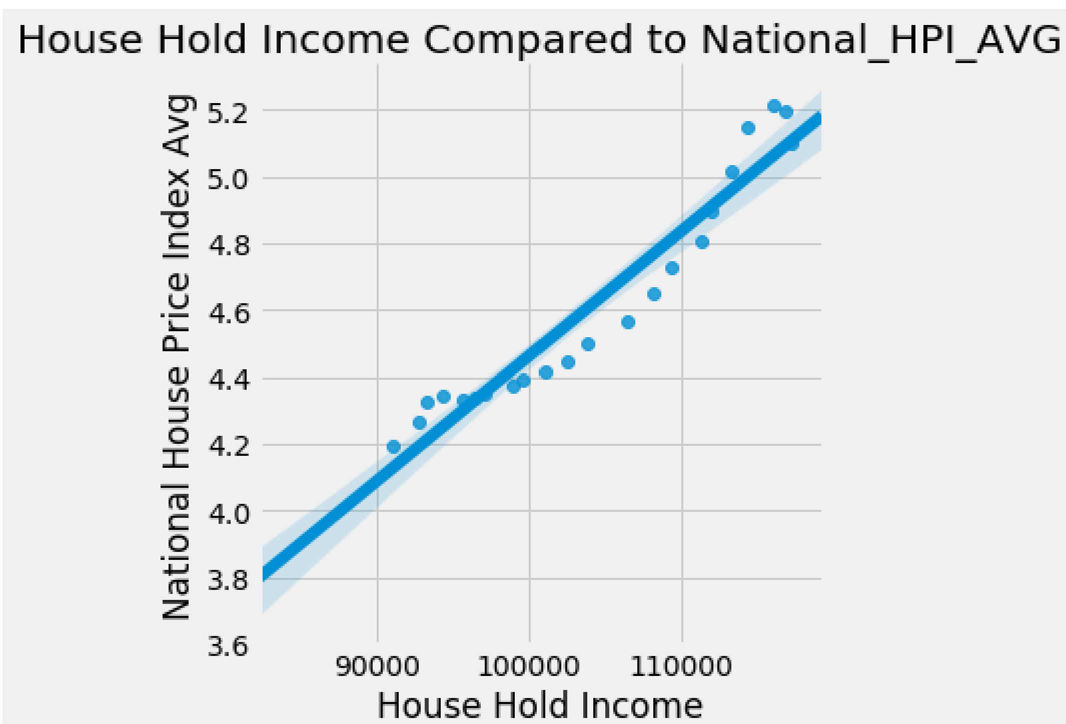
<Figure size 1152x432 with 0 Axes>



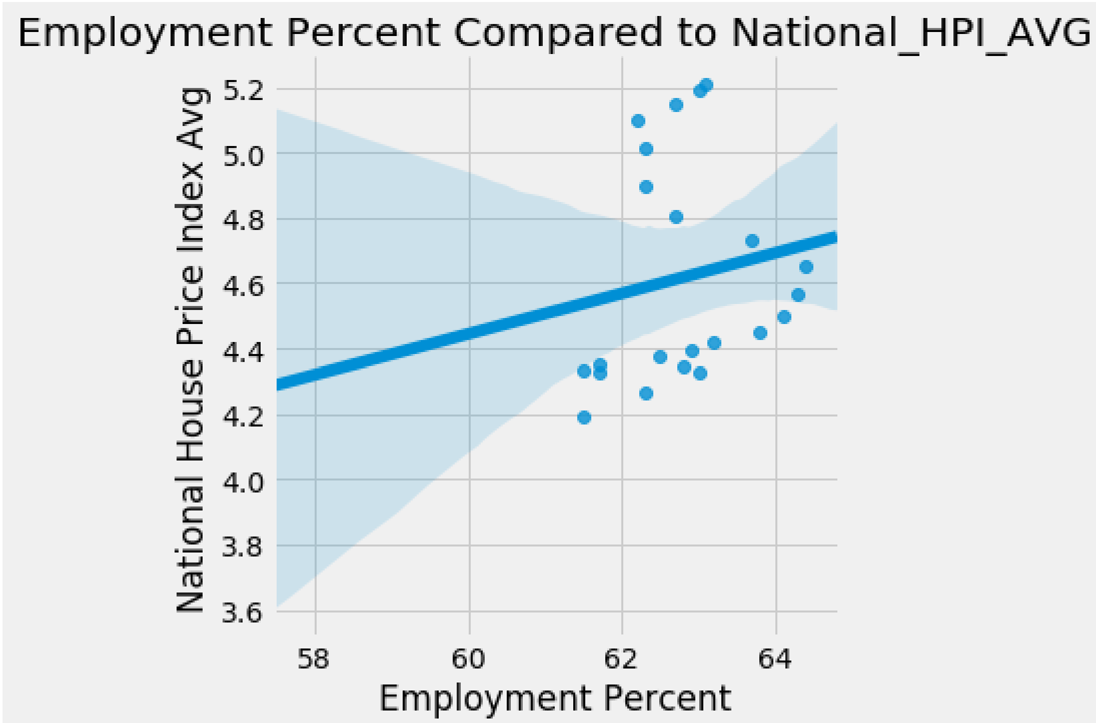
```
<Figure size 864x432 with 0 Axes>
```



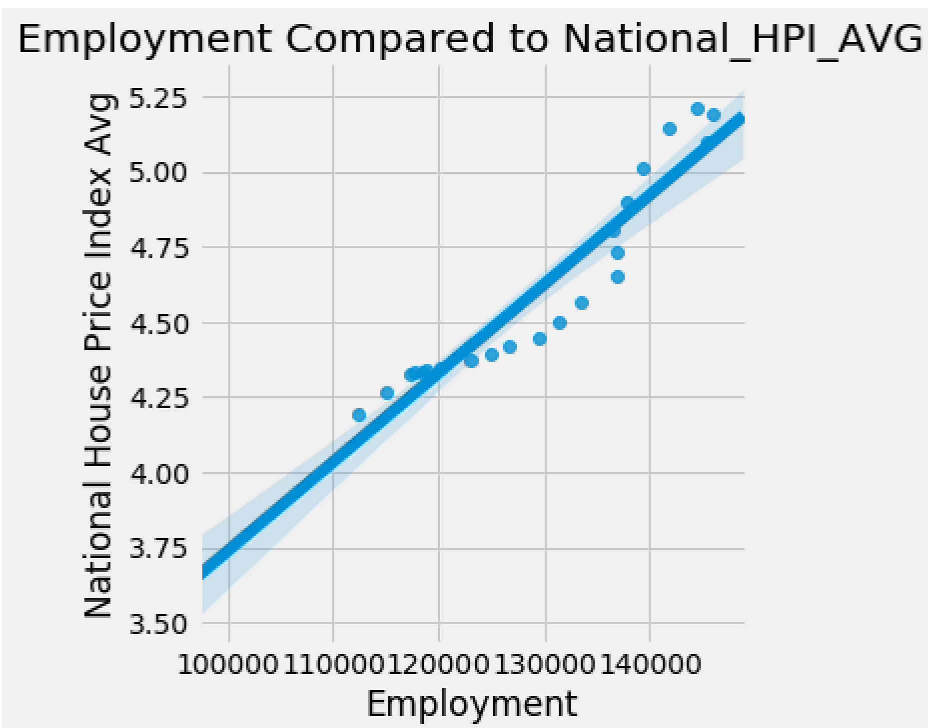

```
<Figure size 864x432 with 0 Axes>
```



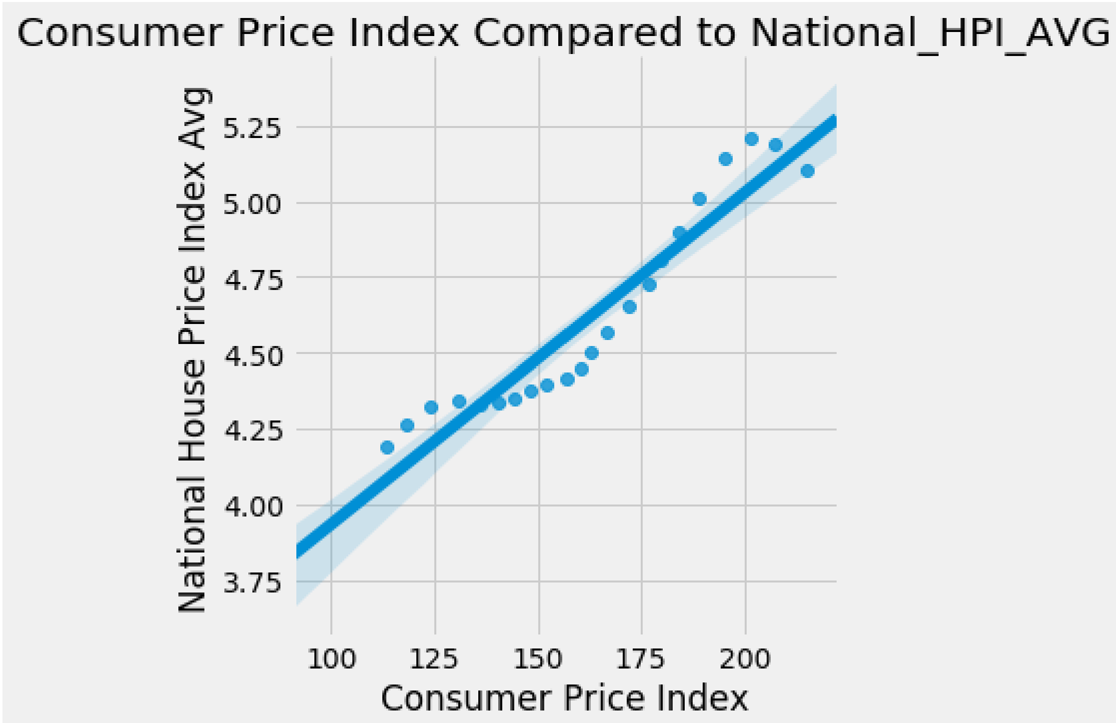
<Figure size 1152x432 with 0 Axes>



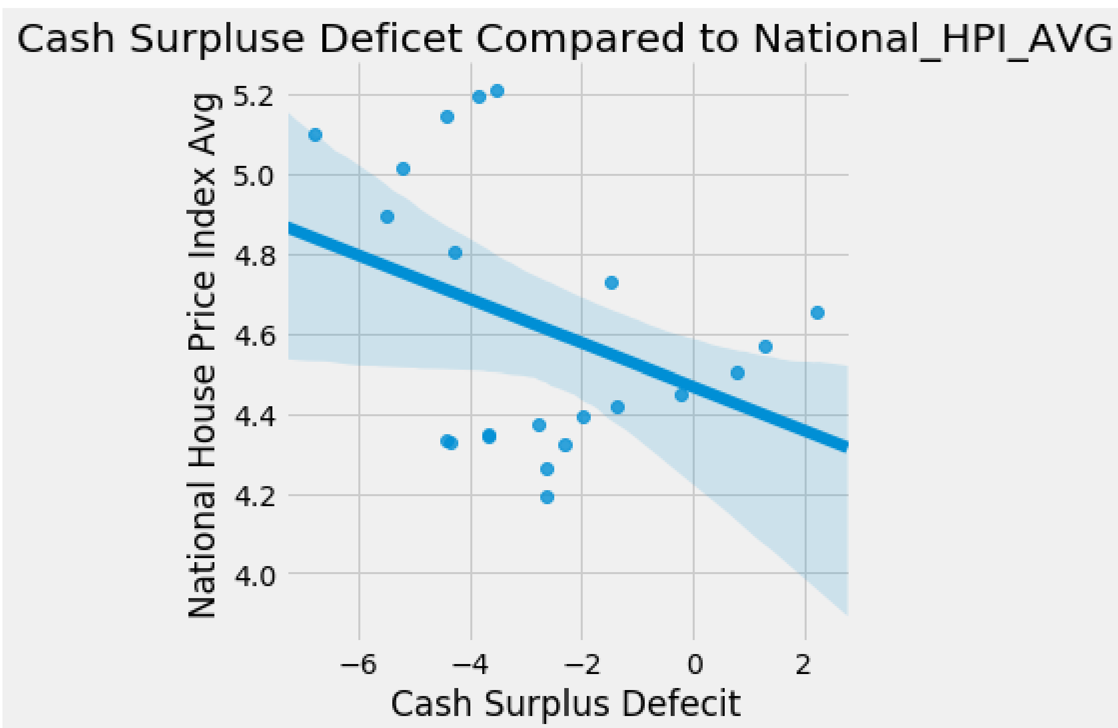
<Figure size 1152x432 with 0 Axes>



```
<Figure size 1152x432 with 0 Axes>
```




```
<Figure size 1152x432 with 0 Axes>
```



2. Time Series Analysis

Time series analysis on real estate median average price by zipcode

- Single Family Home Value
- Rental Price psf
- Listing Price Create future prediction models for all WA State zipcodes historical monthly housing price values.

2.1 Analysis

Transform Data

Transform Real Estate data for time series analysis

2.2 Exploration

```
INFO:file_logger:Single_Family_Residence shape: (252, 351)
INFO:file_logger:All_Homes shape: (252, 353)
INFO:file_logger:RentalPrice_PSF shape: (95, 66)
INFO:file_logger:RentalPrice_All_Homes shape: (95, 82)
INFO:file_logger:ListingPrice_All_Homes shape: (96, 261)
INFO:file_logger:Single_Family_Residence shape: (12, 351)
INFO:file_logger:All_Homes shape: (12, 353)
INFO:file_logger:RentalPrice_PSF shape: (12, 66)
INFO:file_logger:RentalPrice_All_Homes shape: (12, 82)
INFO:file_logger:ListingPrice_All_Homes shape: (12, 261)
INFO:file_logger:Single_Family_Residence shape: (264, 351)
INFO:file_logger:All_Homes shape: (264, 353)
INFO:file_logger:RentalPrice_PSF shape: (107, 66)
INFO:file_logger:RentalPrice_All_Homes shape: (107, 82)
INFO:file_logger:ListingPrice_All_Homes shape: (108, 261)
```

2.3 Model

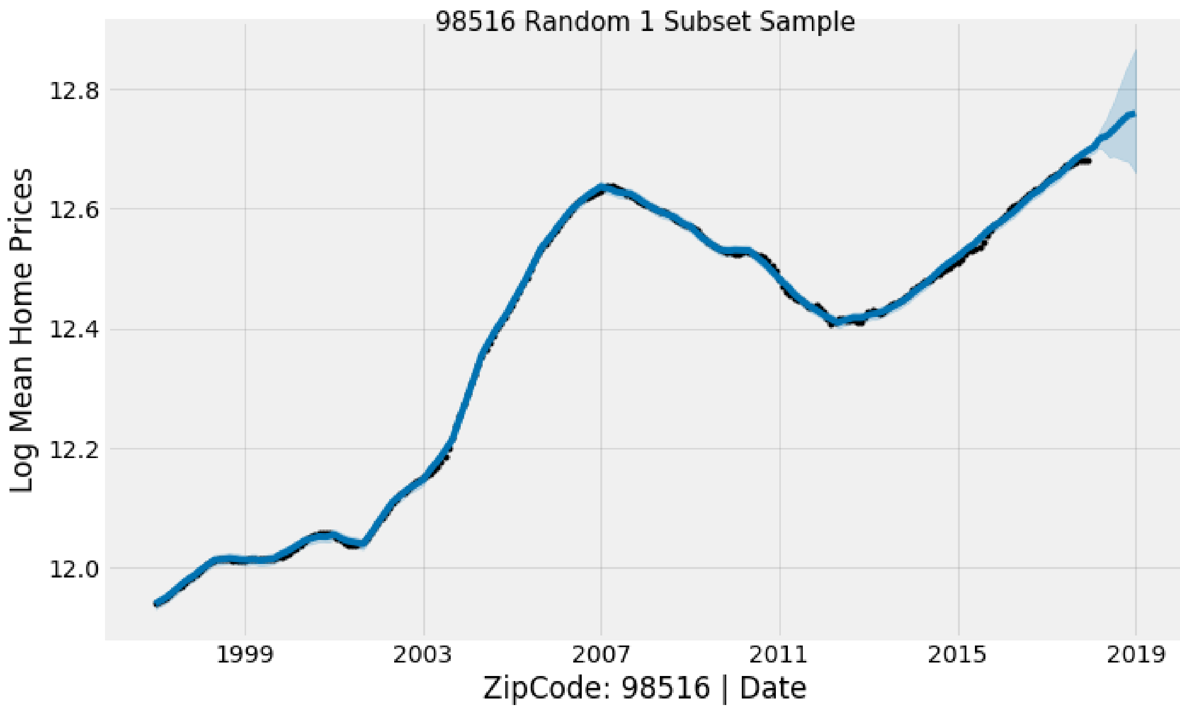
Note: All timeseries models were ran prior on google colab and saved as pickle files for continued downstream application

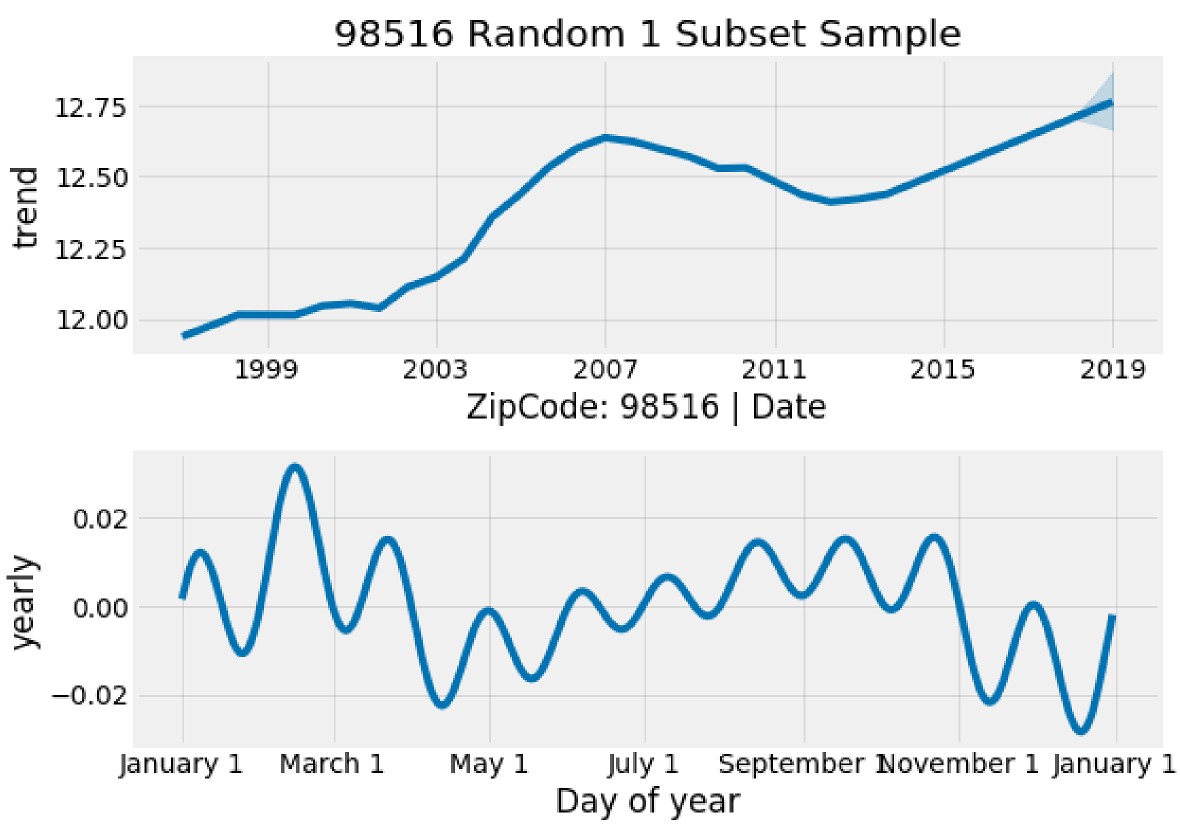
2.4 Results

Training Data Sets

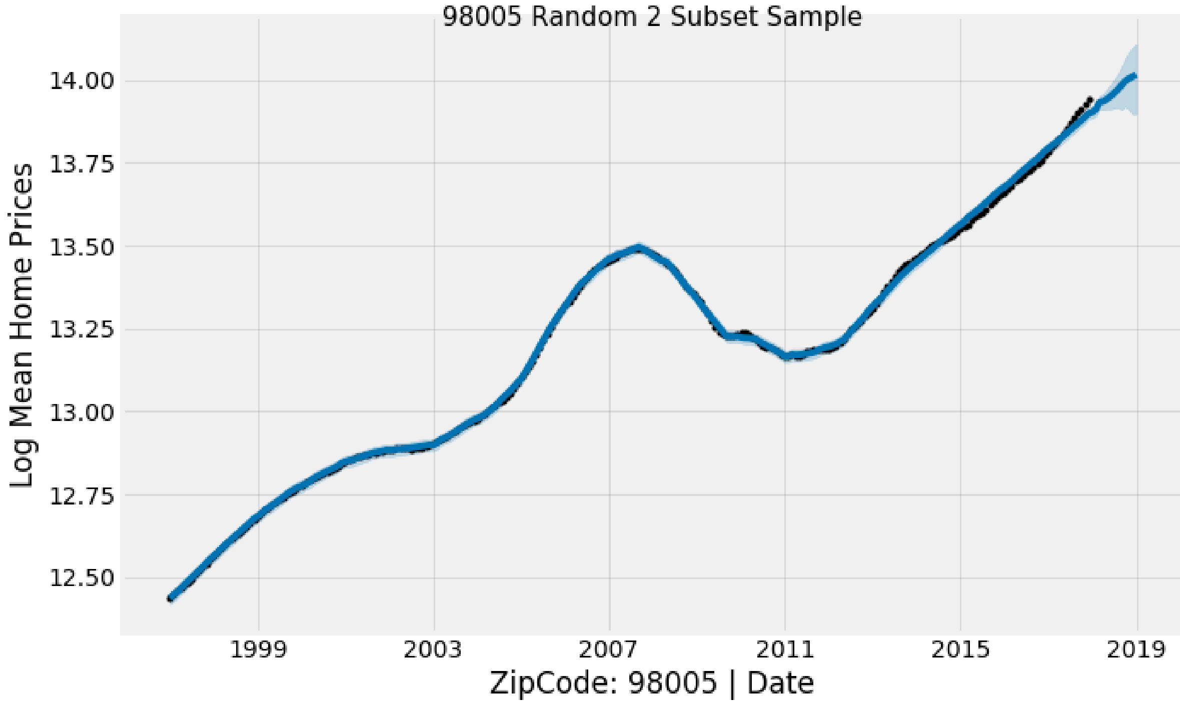
Price Trend from 1997 through 2017 - With a 12 month future prediction...

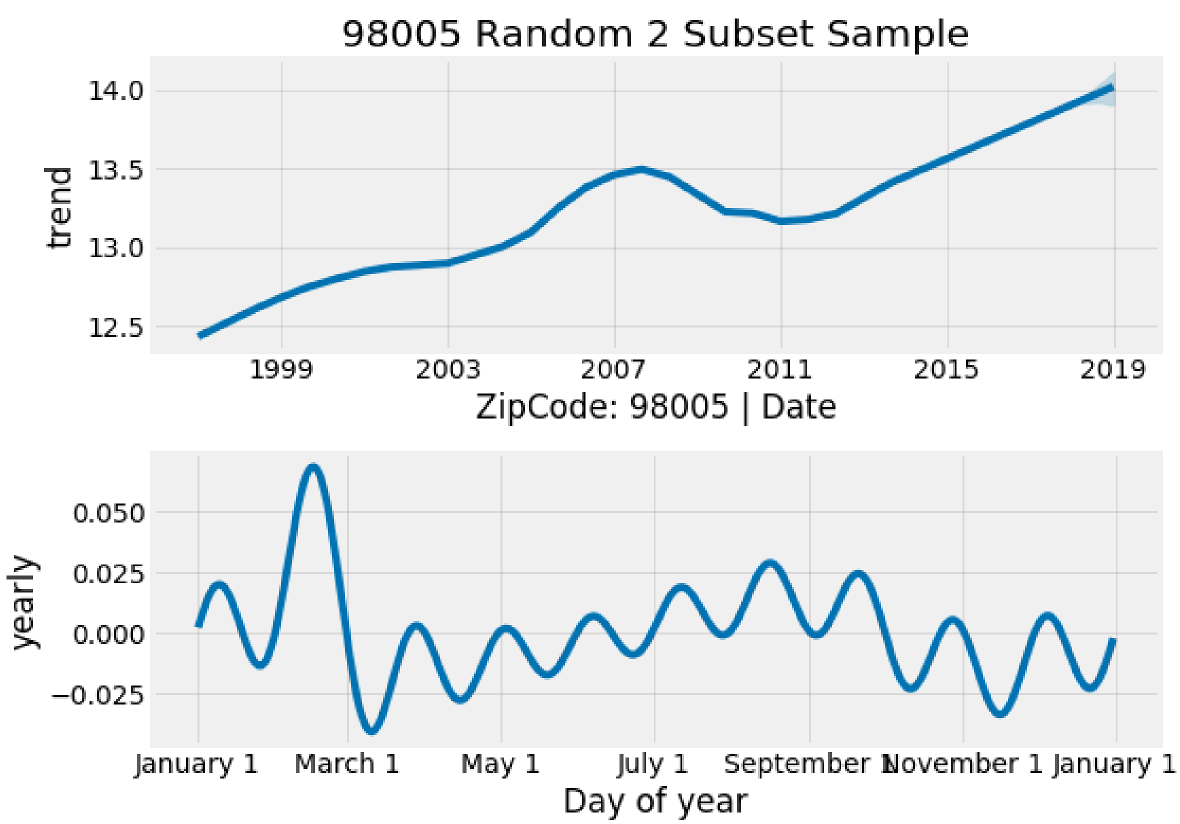
<Figure size 1440x1080 with 0 Axes>



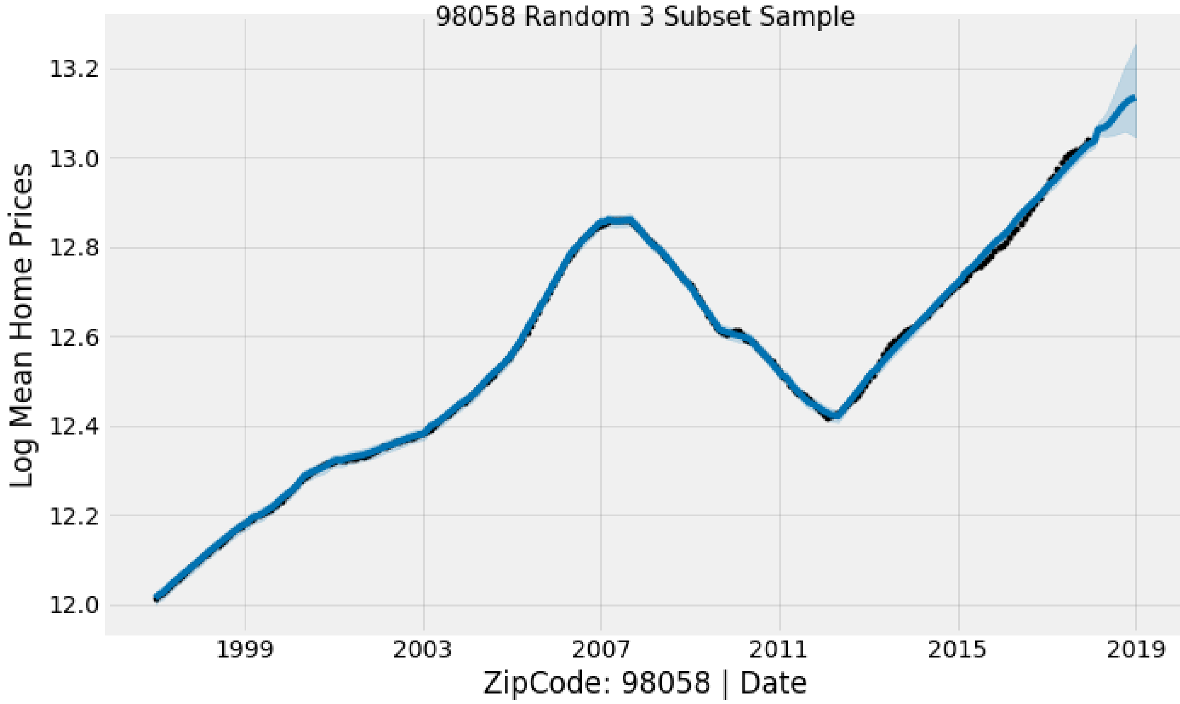


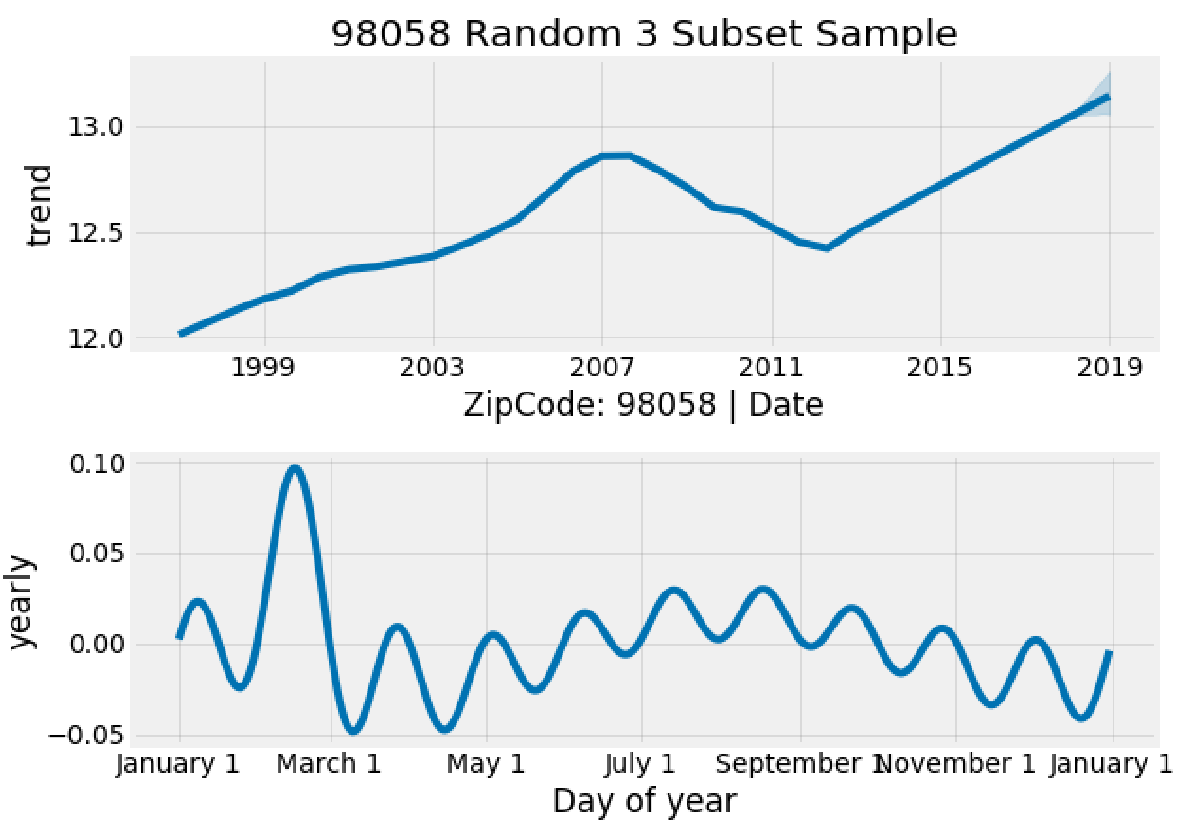
<Figure size 1440x1080 with 0 Axes>



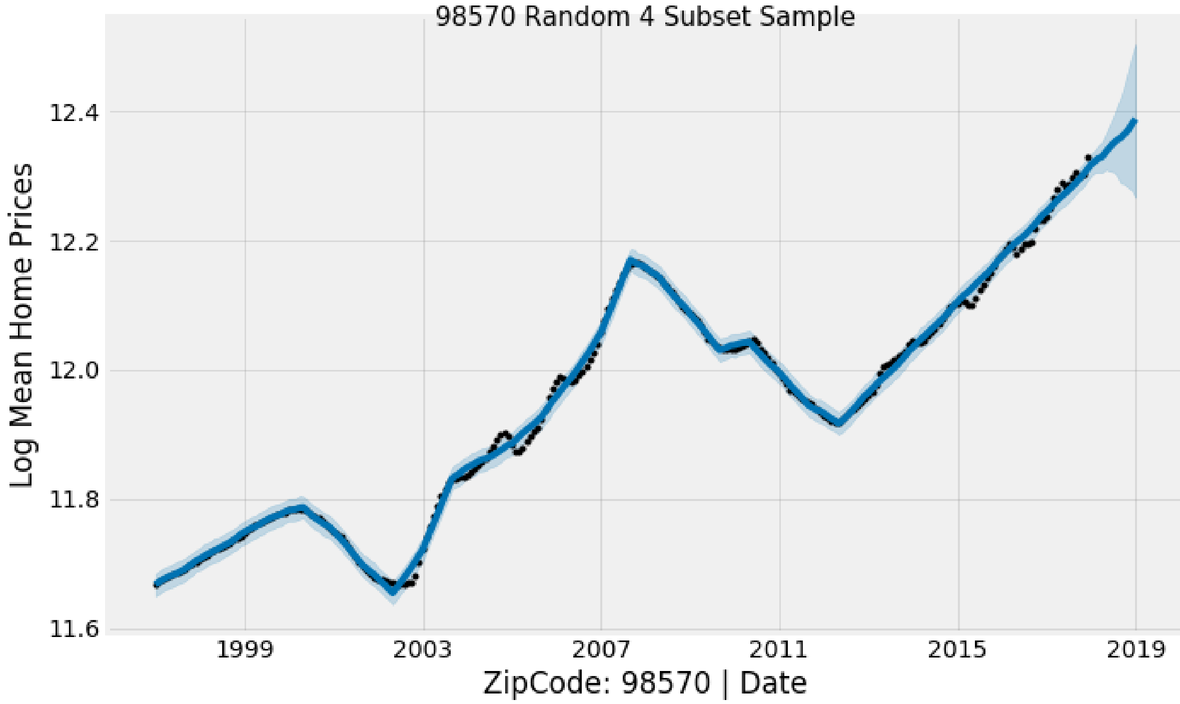


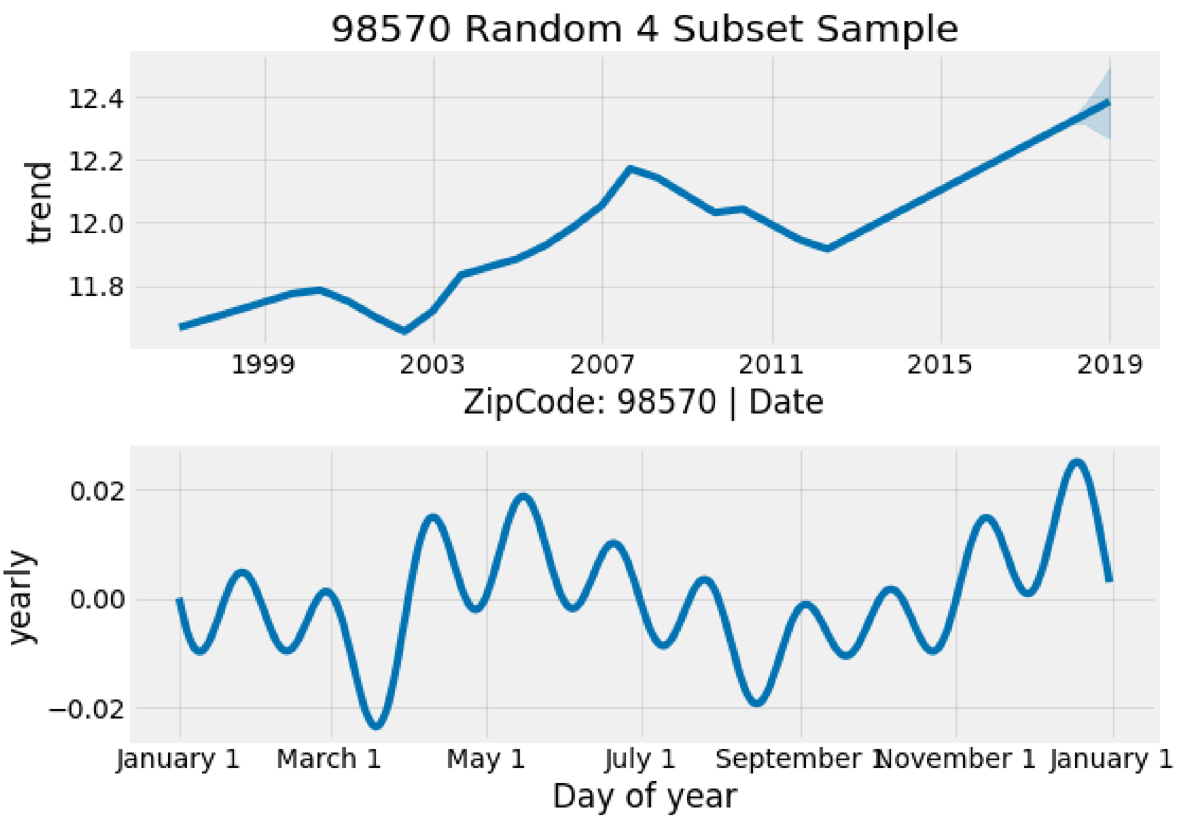

```
&lt;Figure size 1440x1080 with 0 Axes&gt;
```



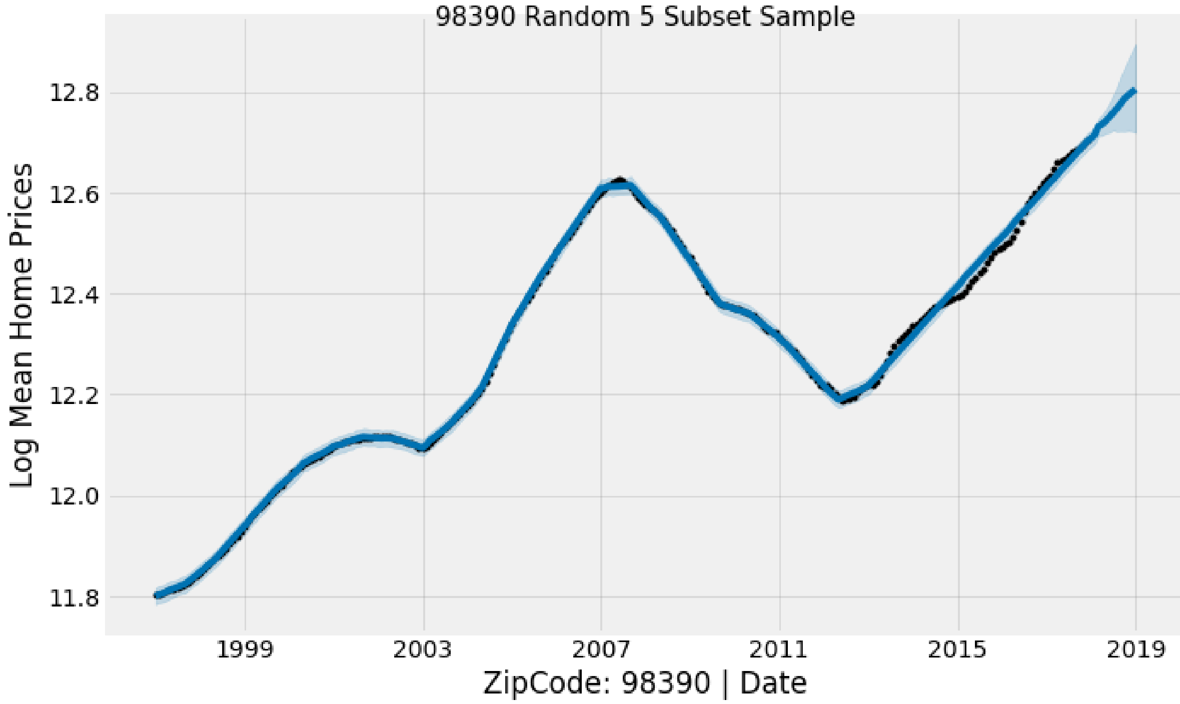


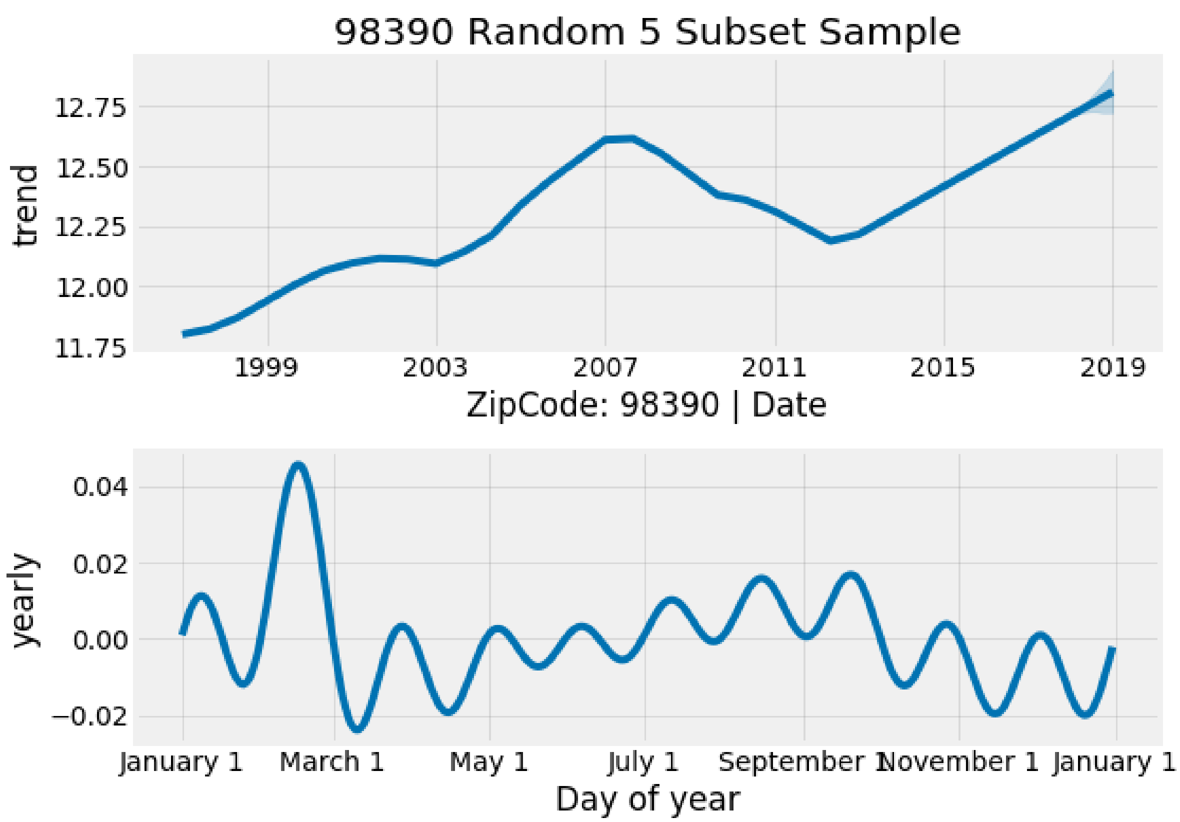
<Figure size 1440x1080 with 0 Axes>



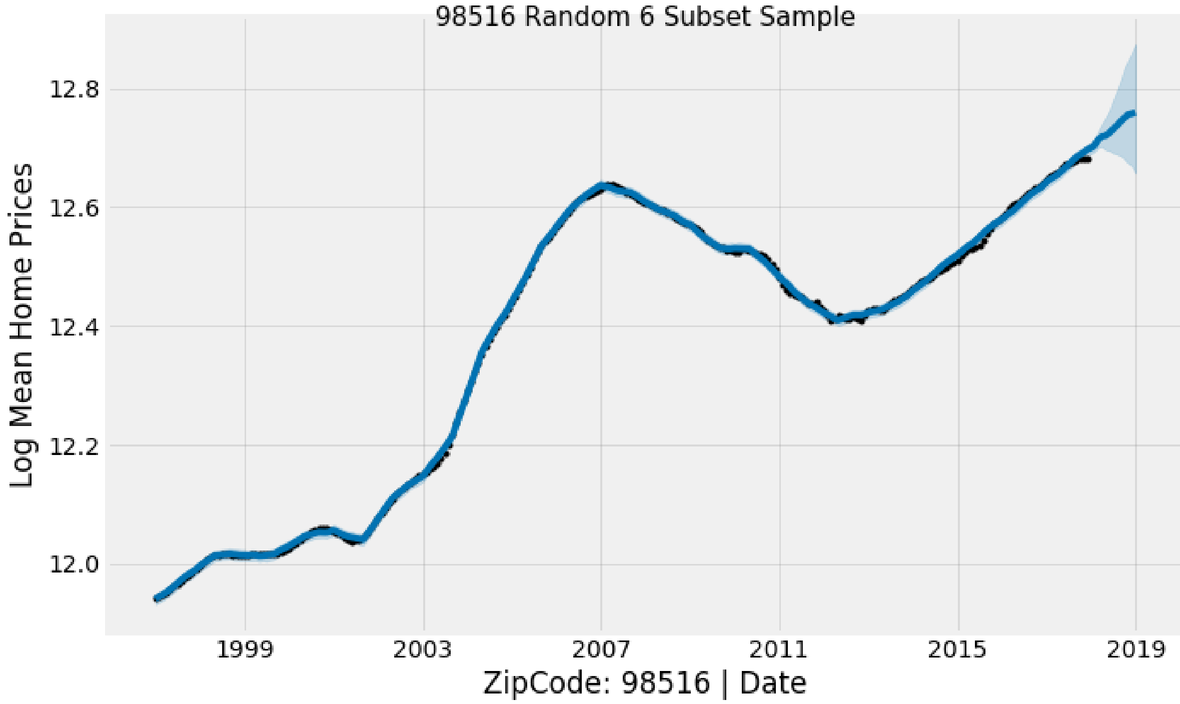


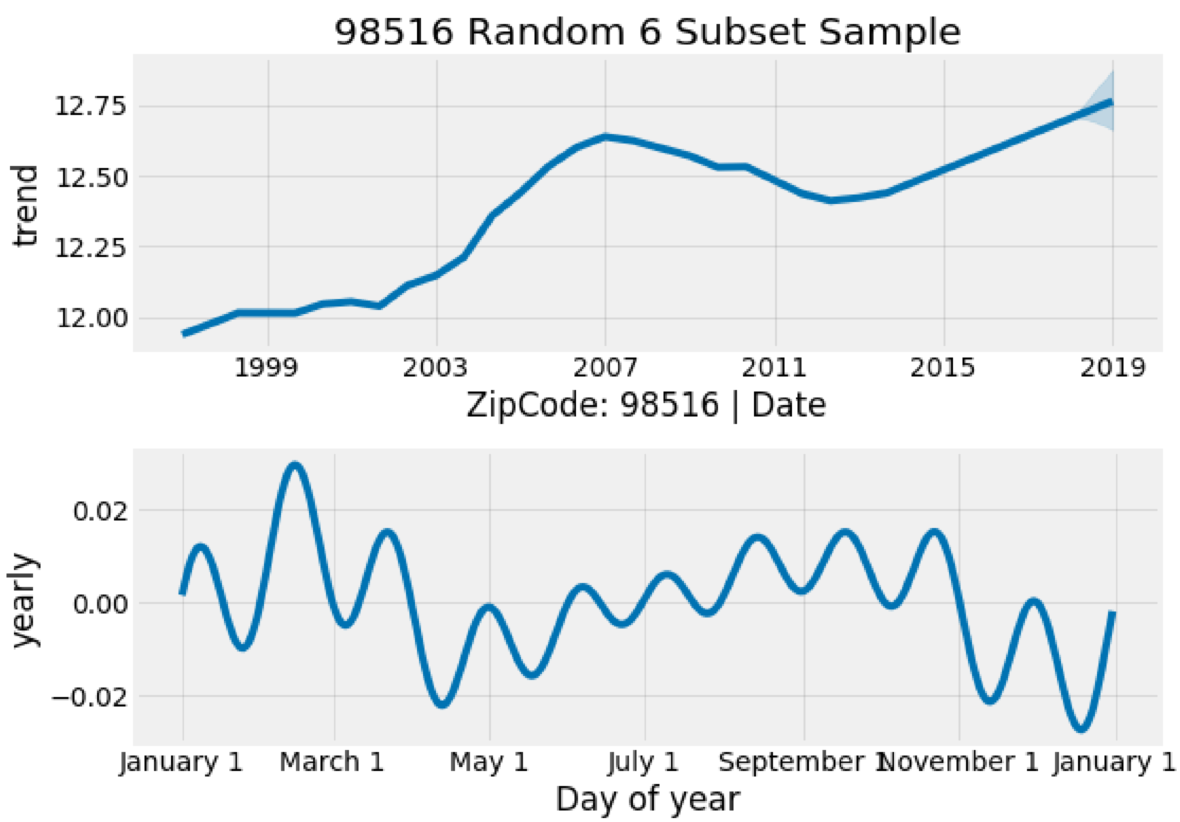

```
<Figure size 1440x1080 with 0 Axes>
```



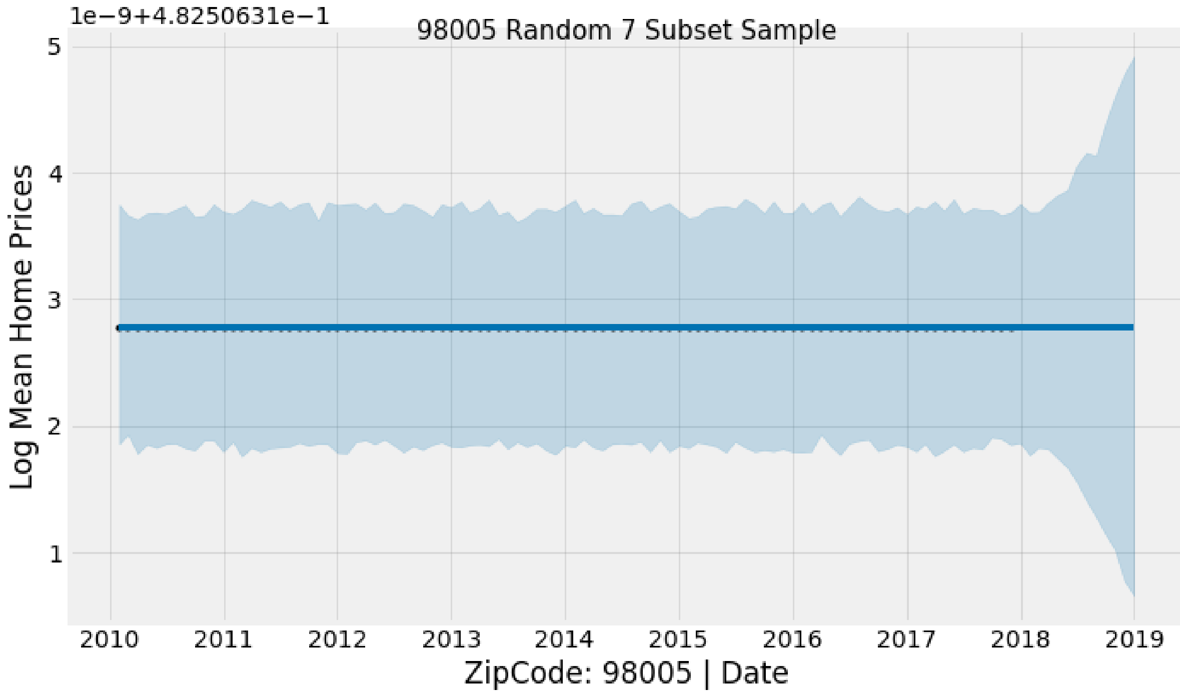


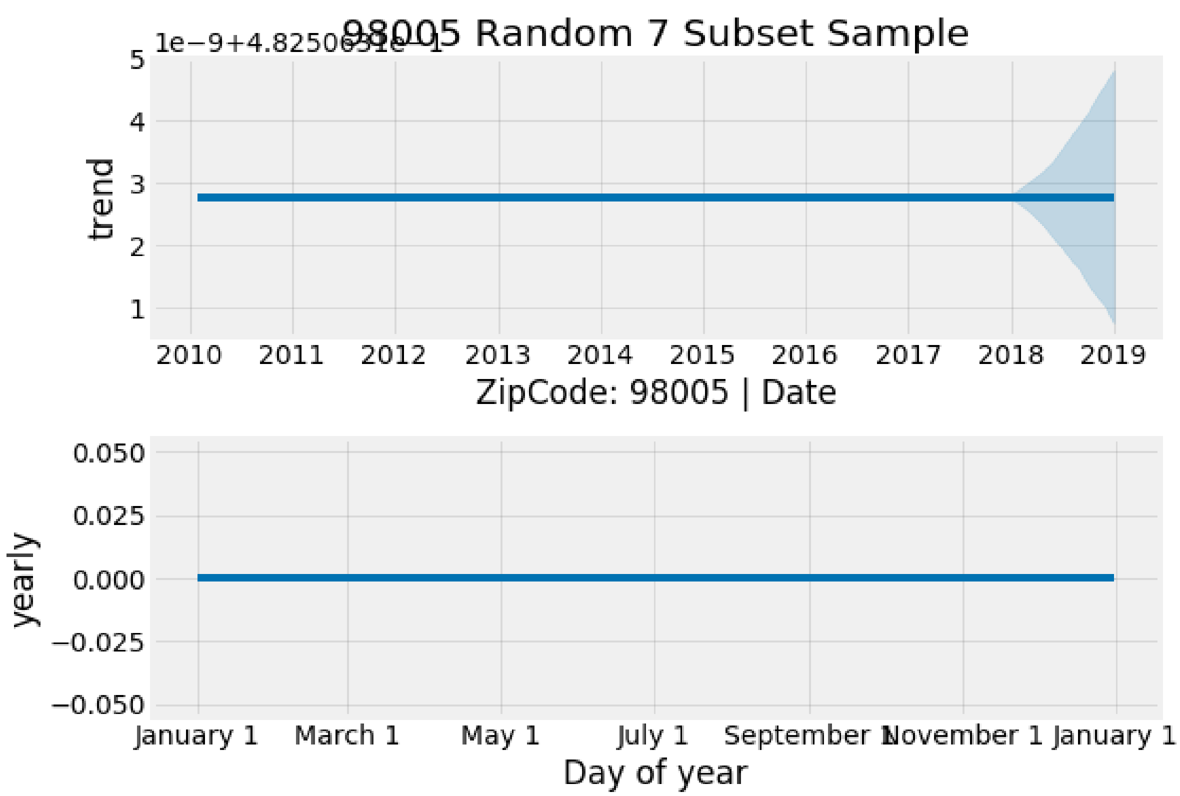

```
<Figure size 1440x1080 with 0 Axes>
```



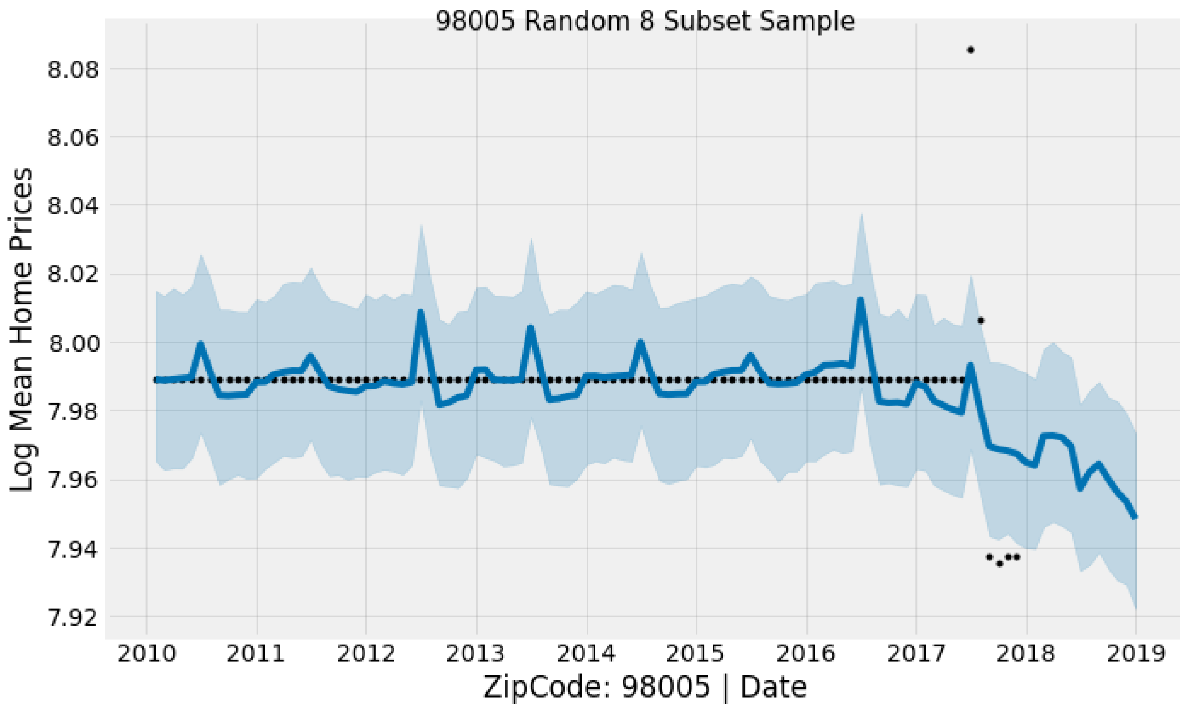


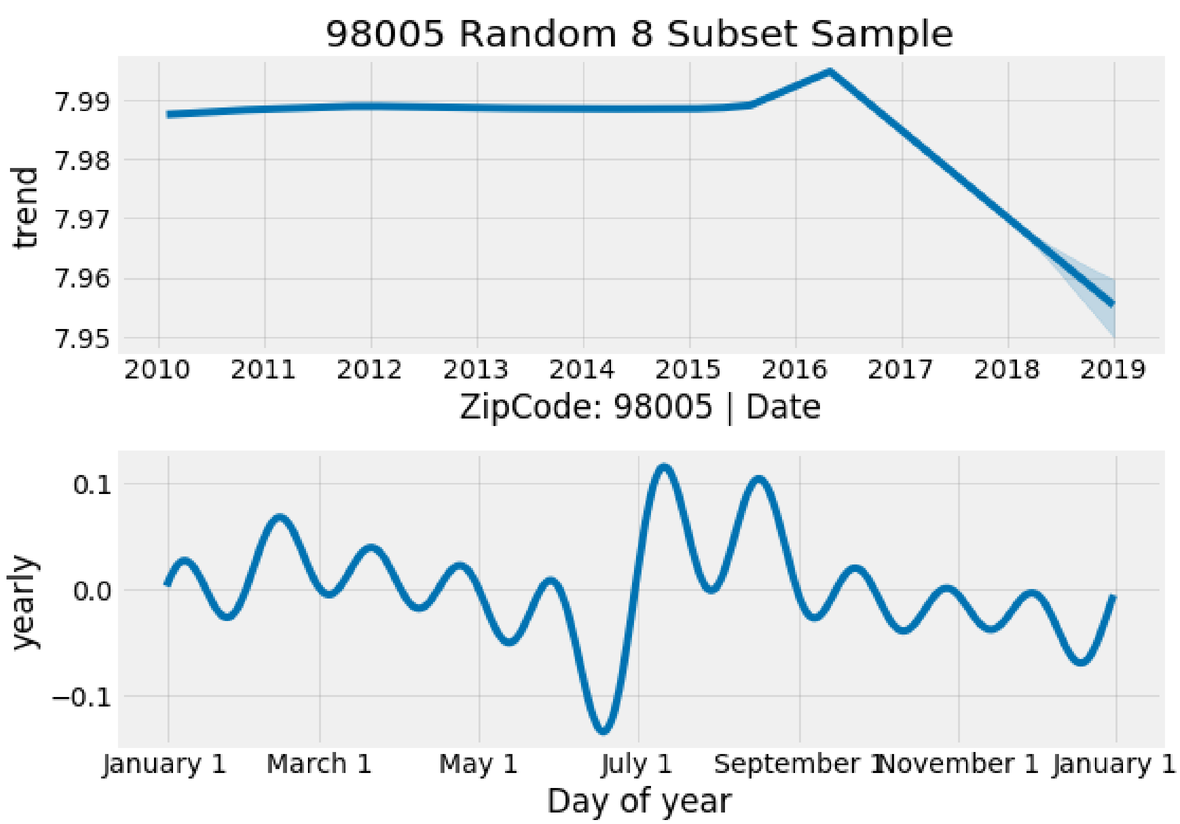

```
<Figure size 1440x1080 with 0 Axes>
```



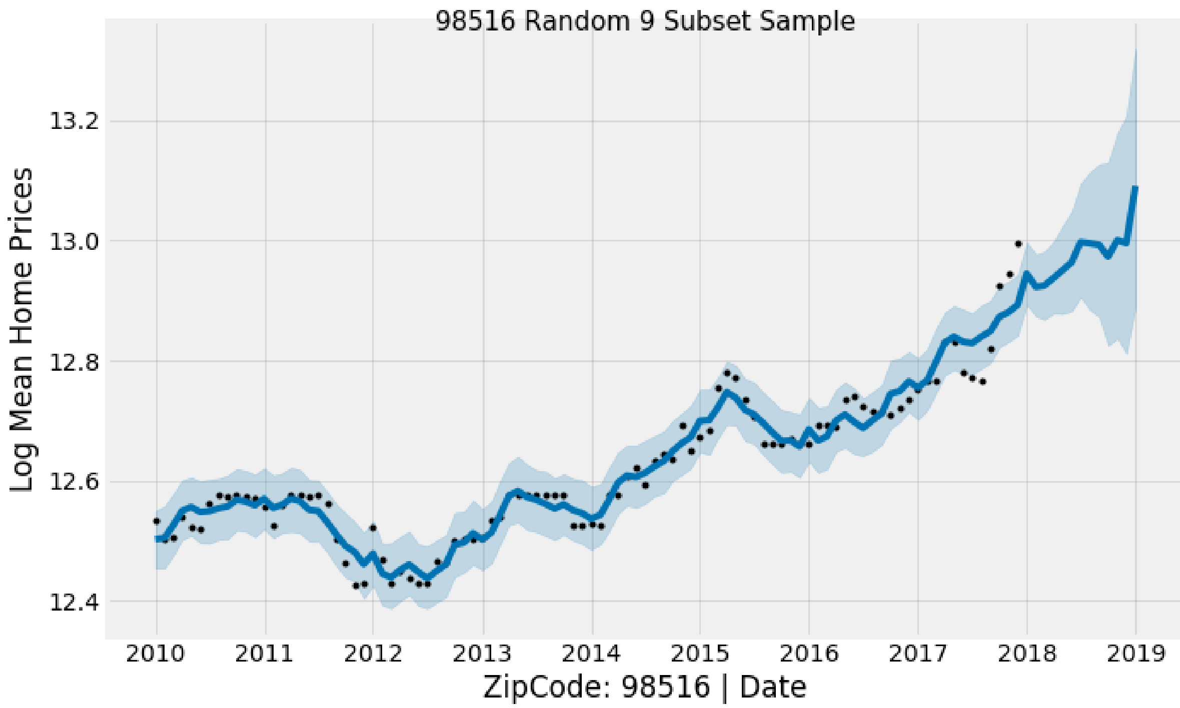


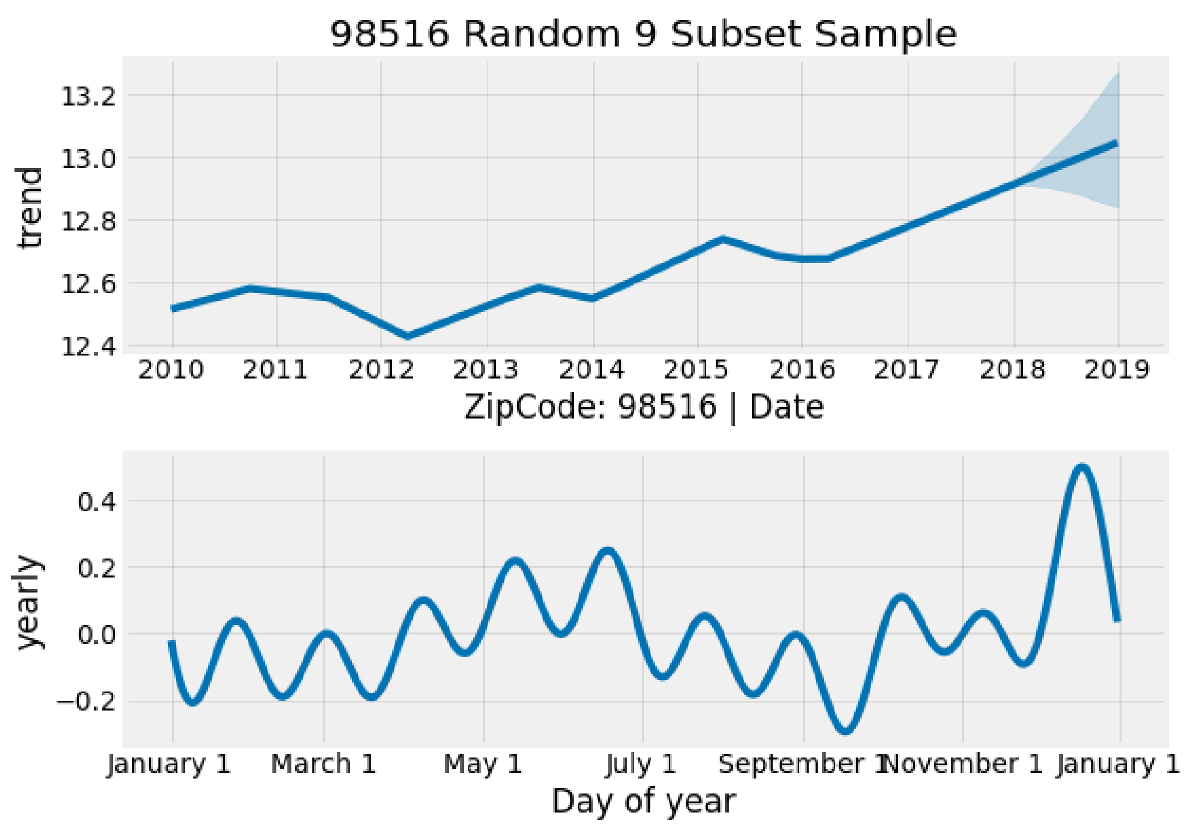
<Figure size 1440x1080 with 0 Axes>





<Figure size 1440x1080 with 0 Axes>

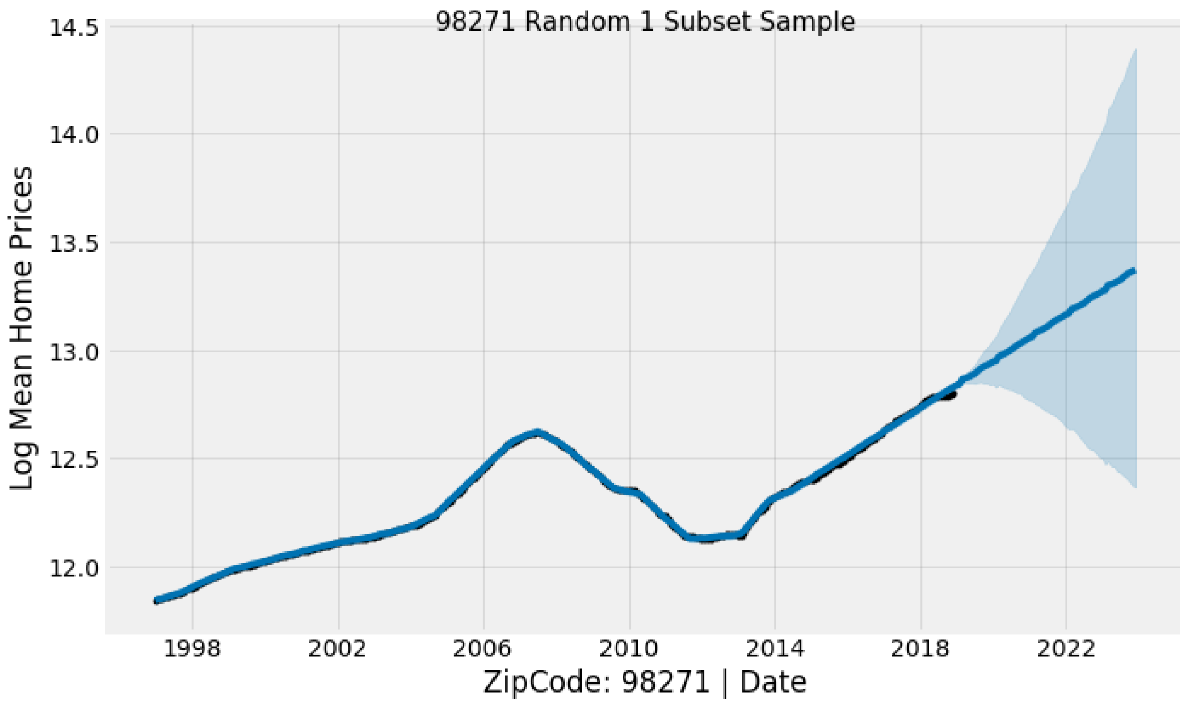


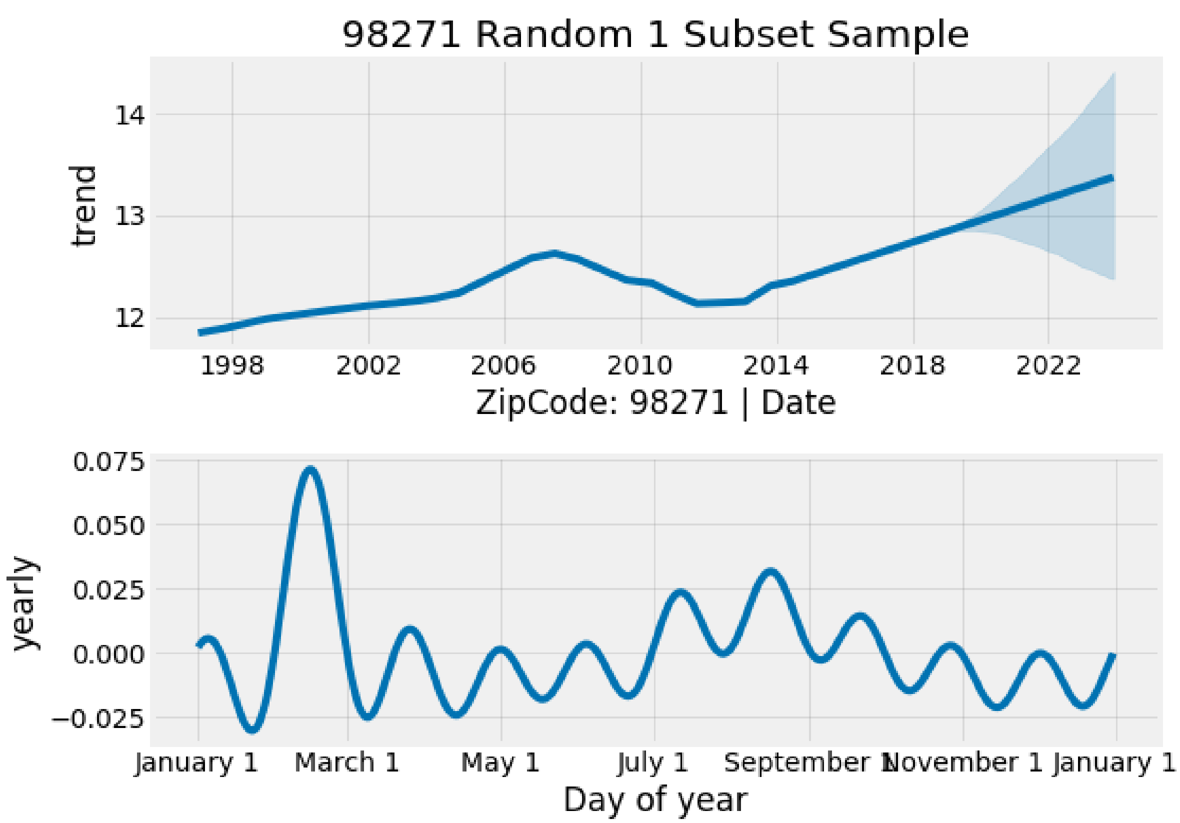


Future Prediction Trends

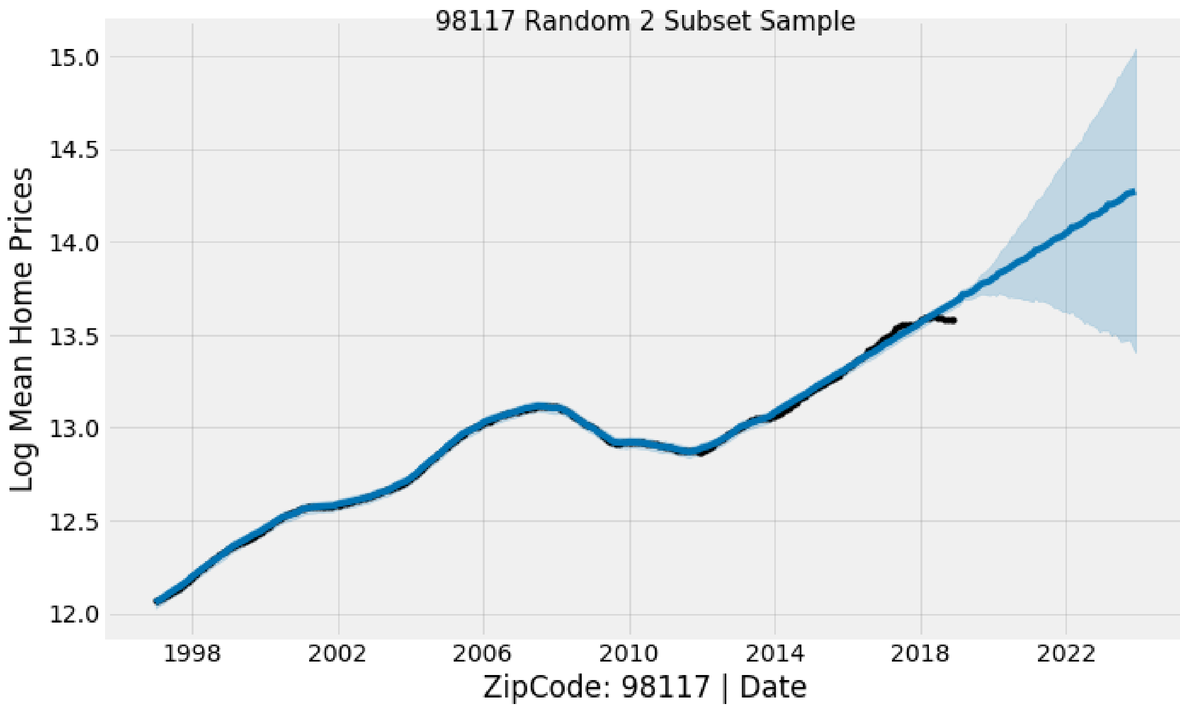
Price Trend from 1997 through 2018 - With a 5 year future prediction...

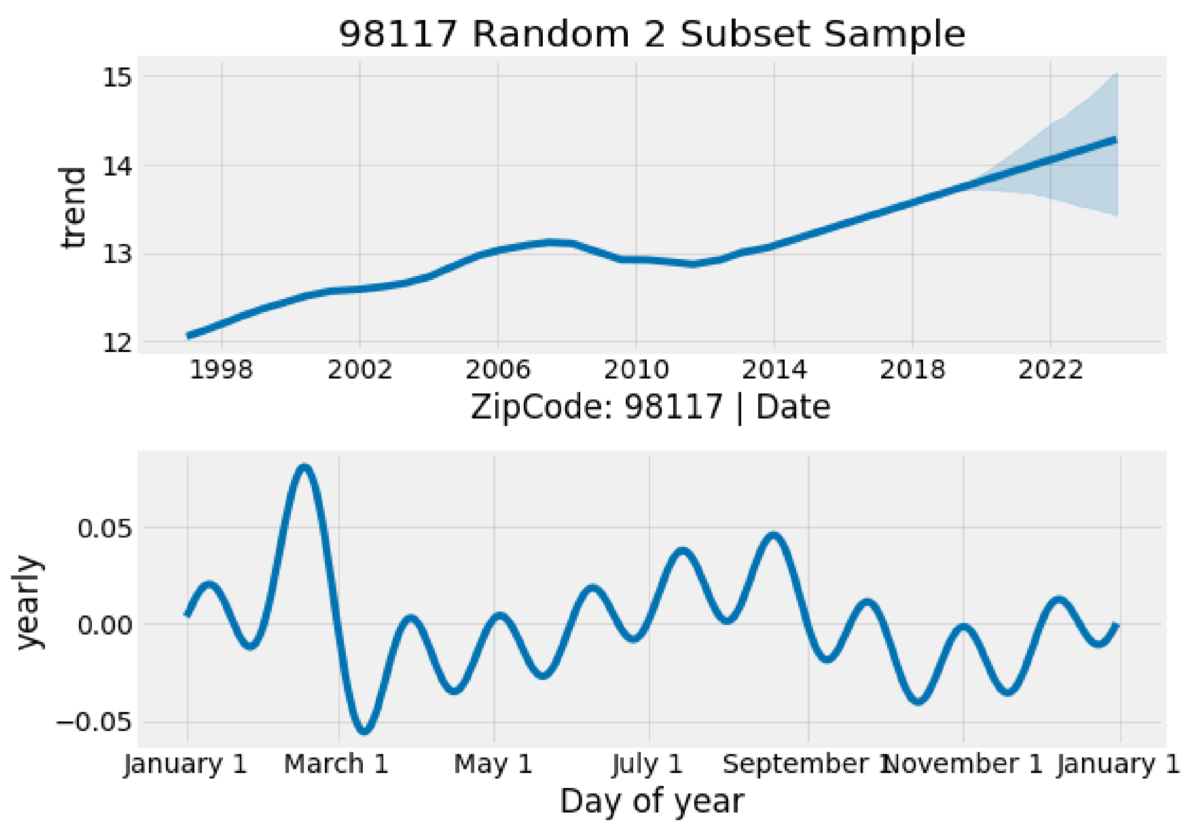
```
<Figure size 1440x1080 with 0 Axes>
```



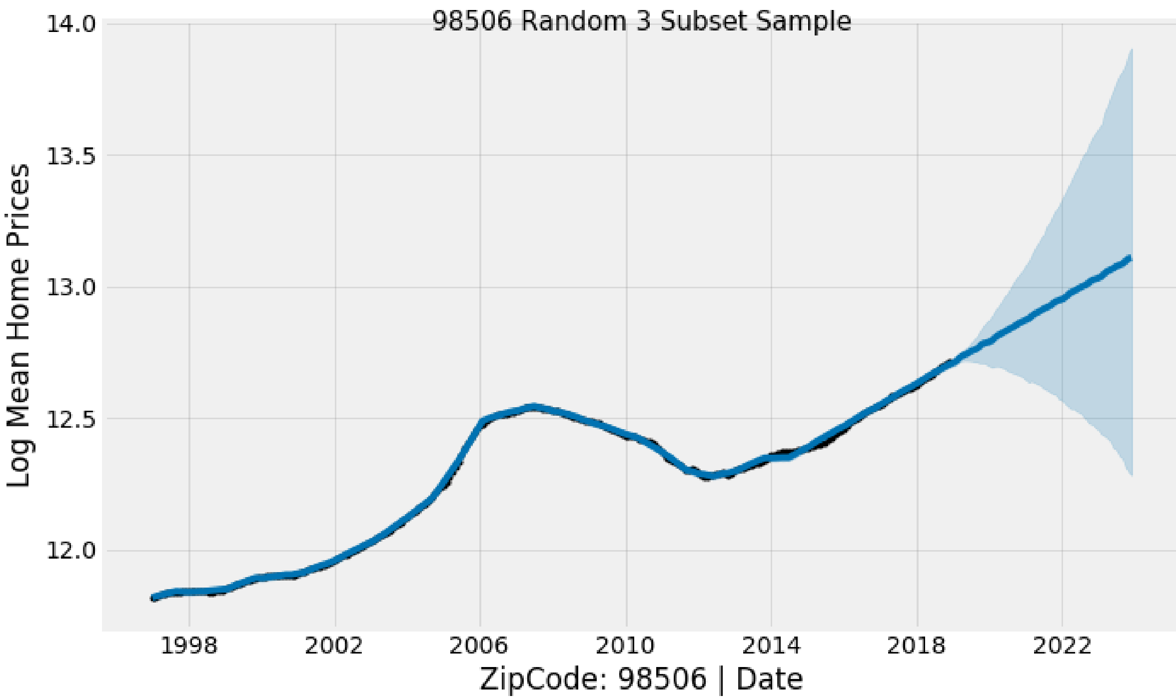


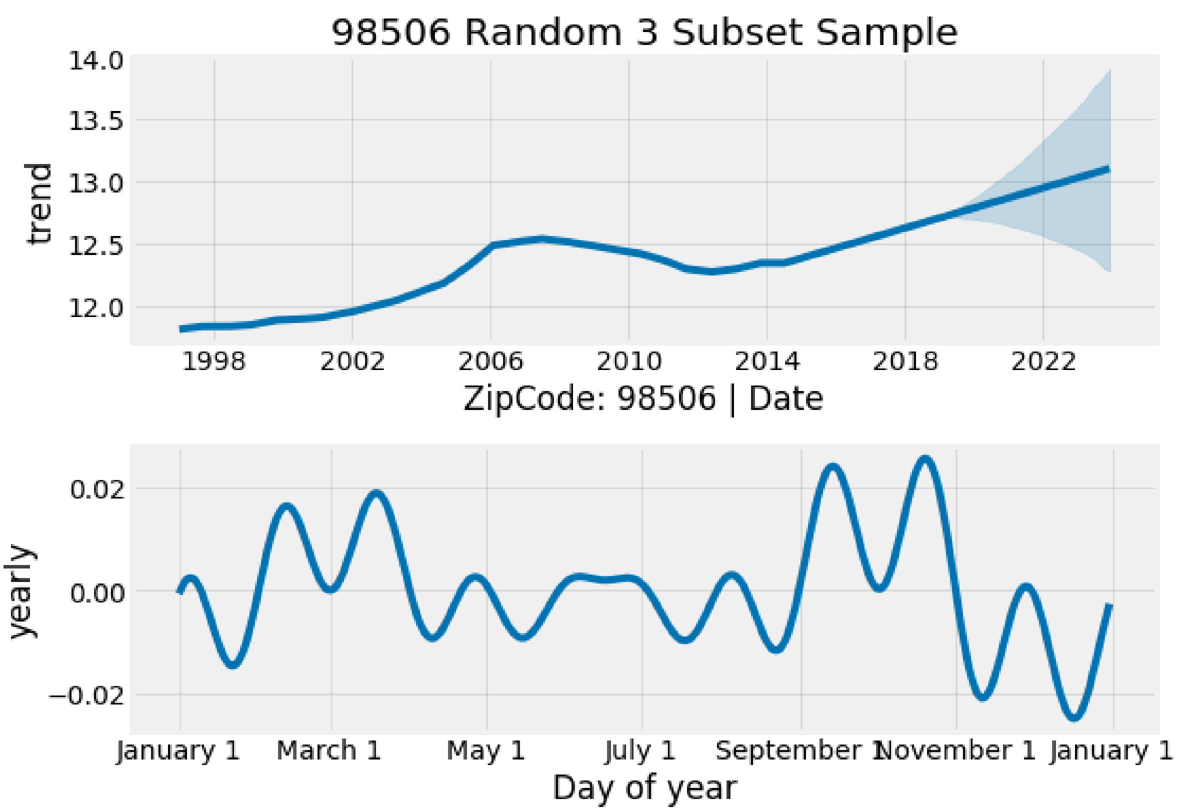

```
<Figure size 1440x1080 with 0 Axes>
```



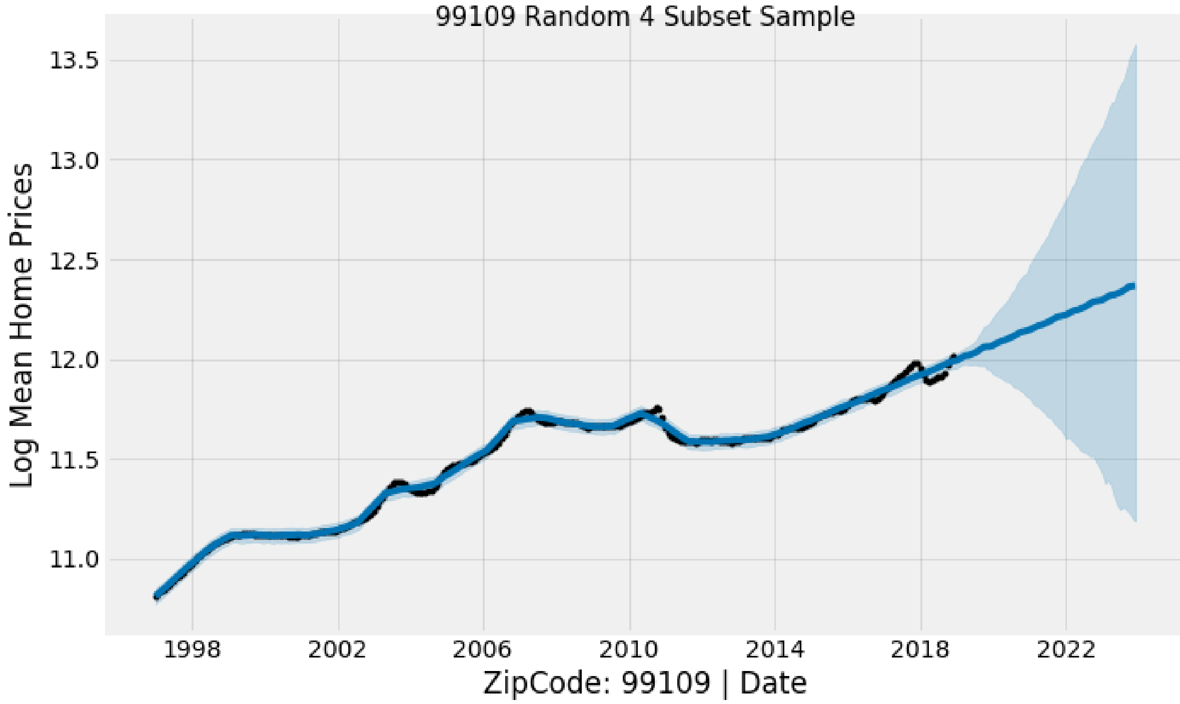


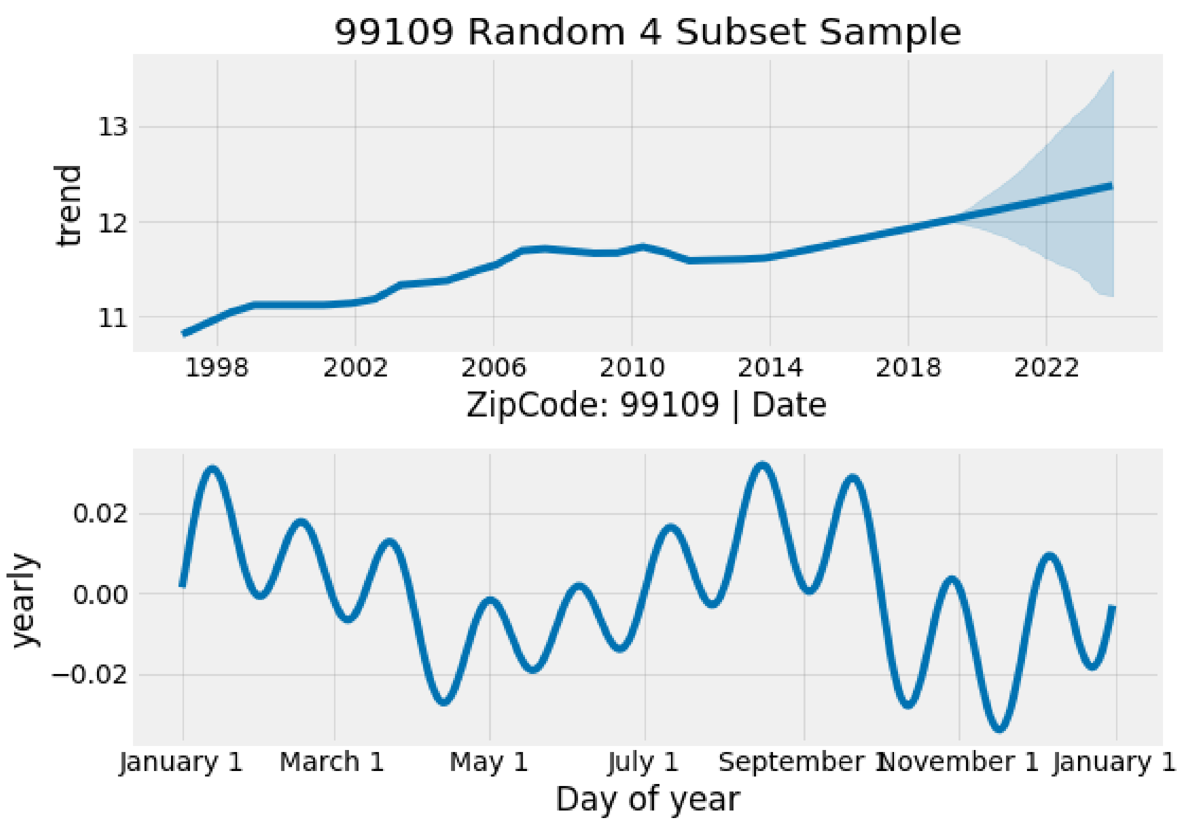
<Figure size 1440x1080 with 0 Axes>



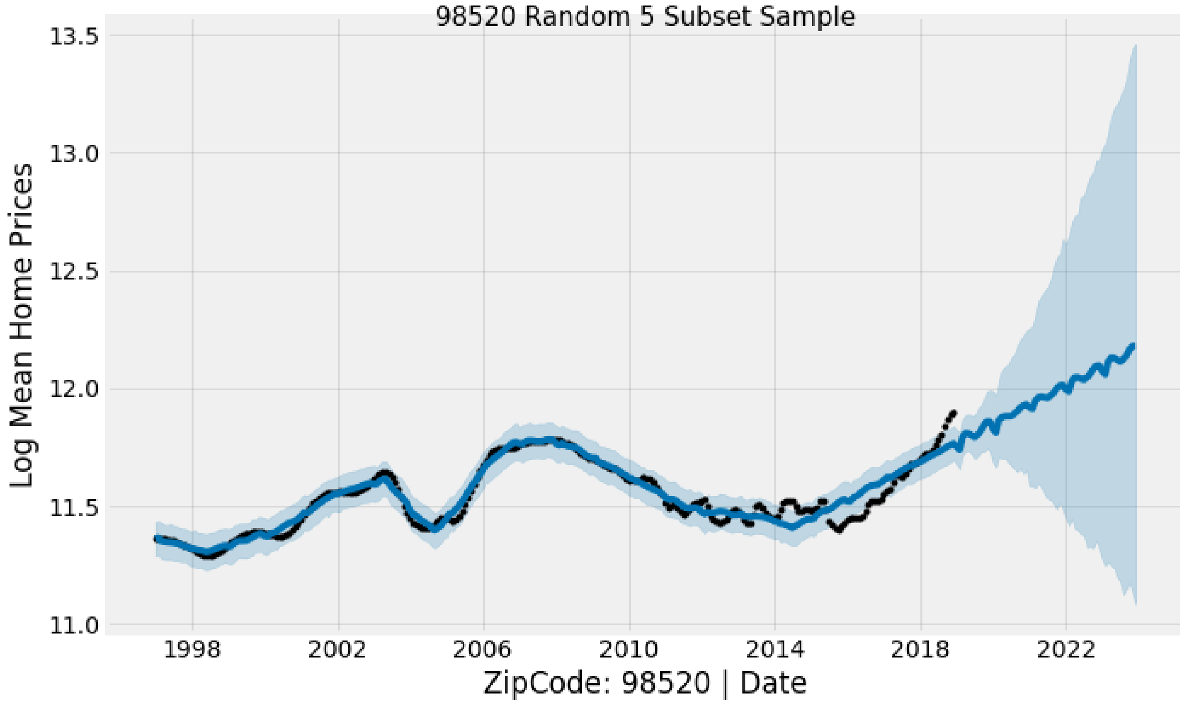


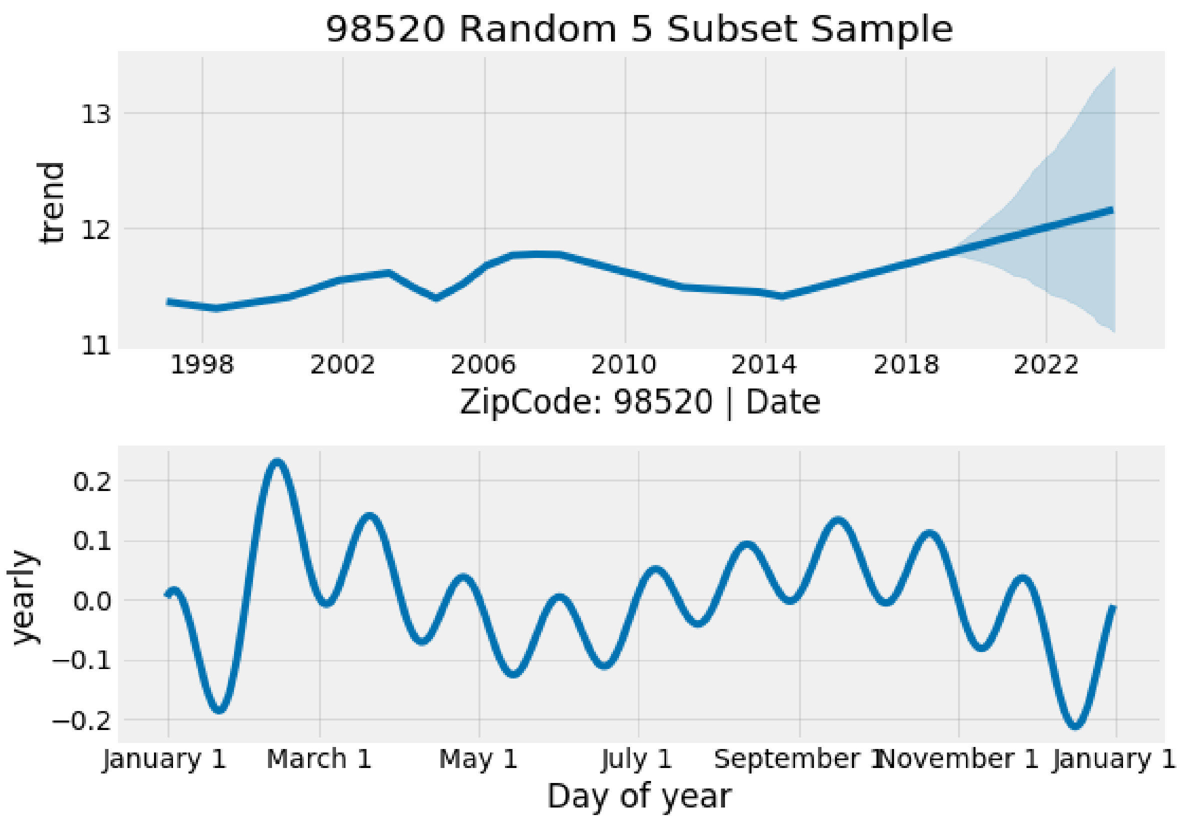

```
<Figure size 1440x1080 with 0 Axes>
```



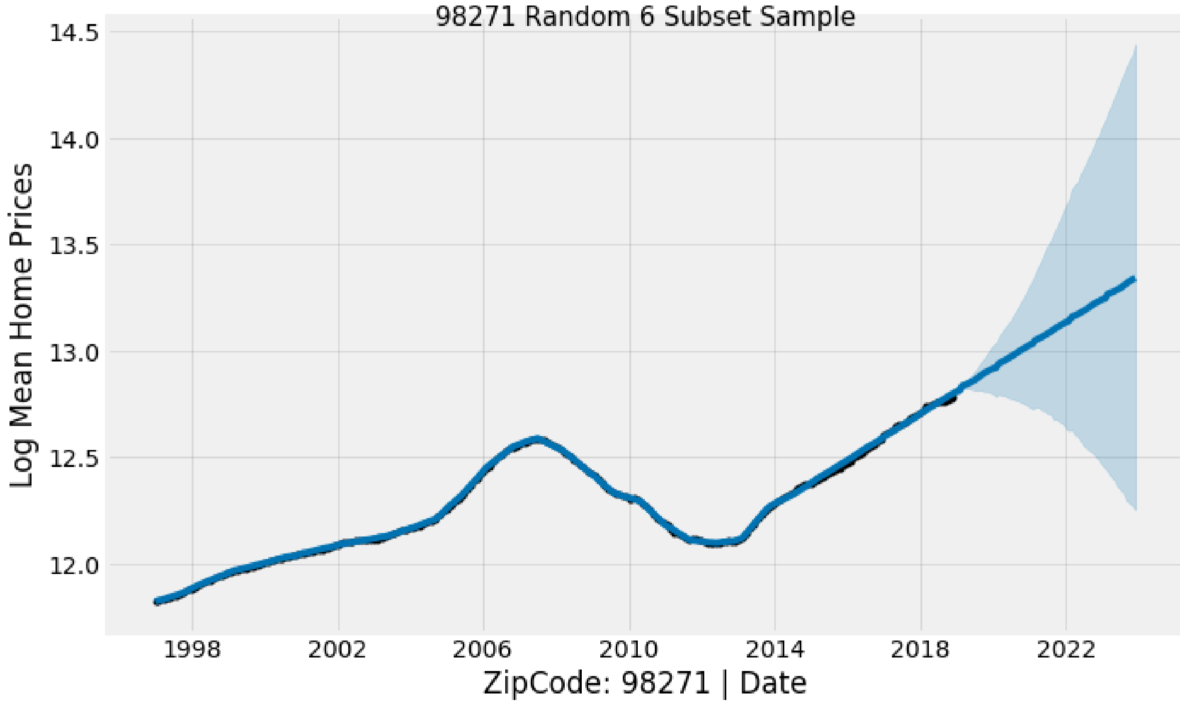


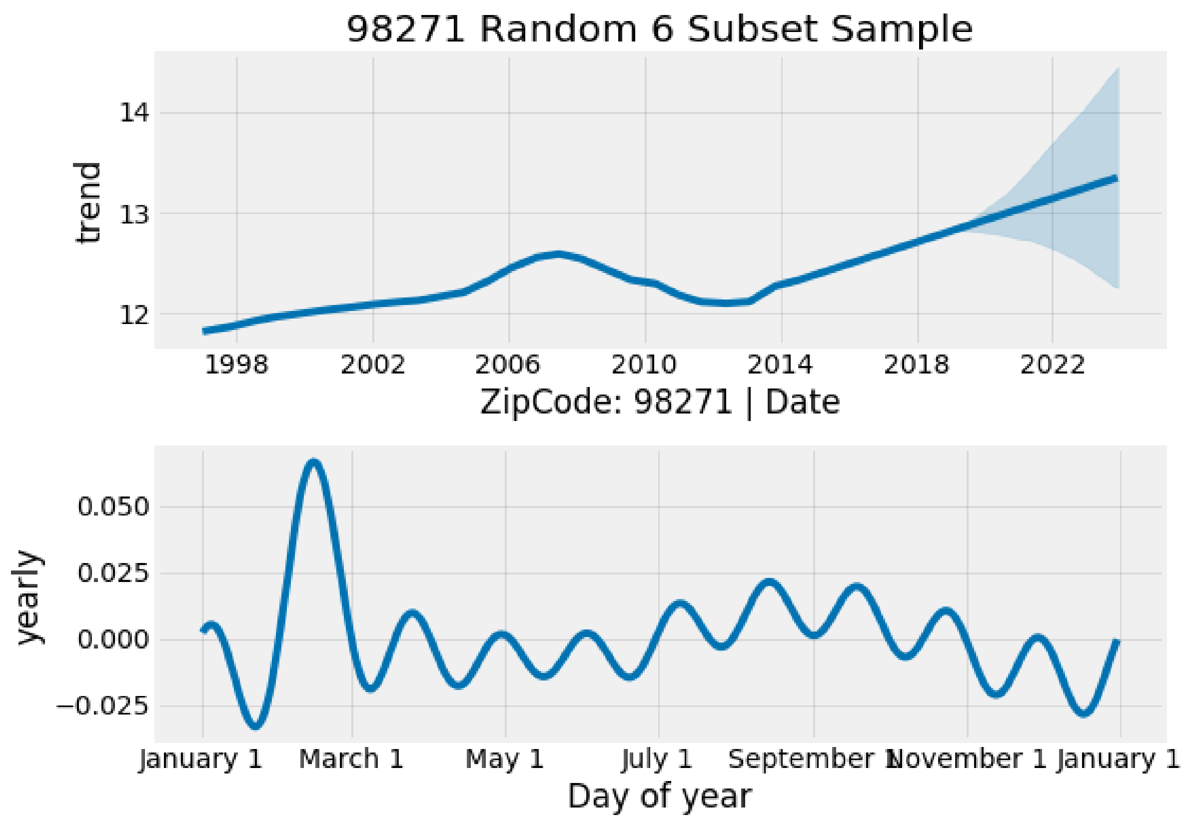
<Figure size 1440x1080 with 0 Axes>



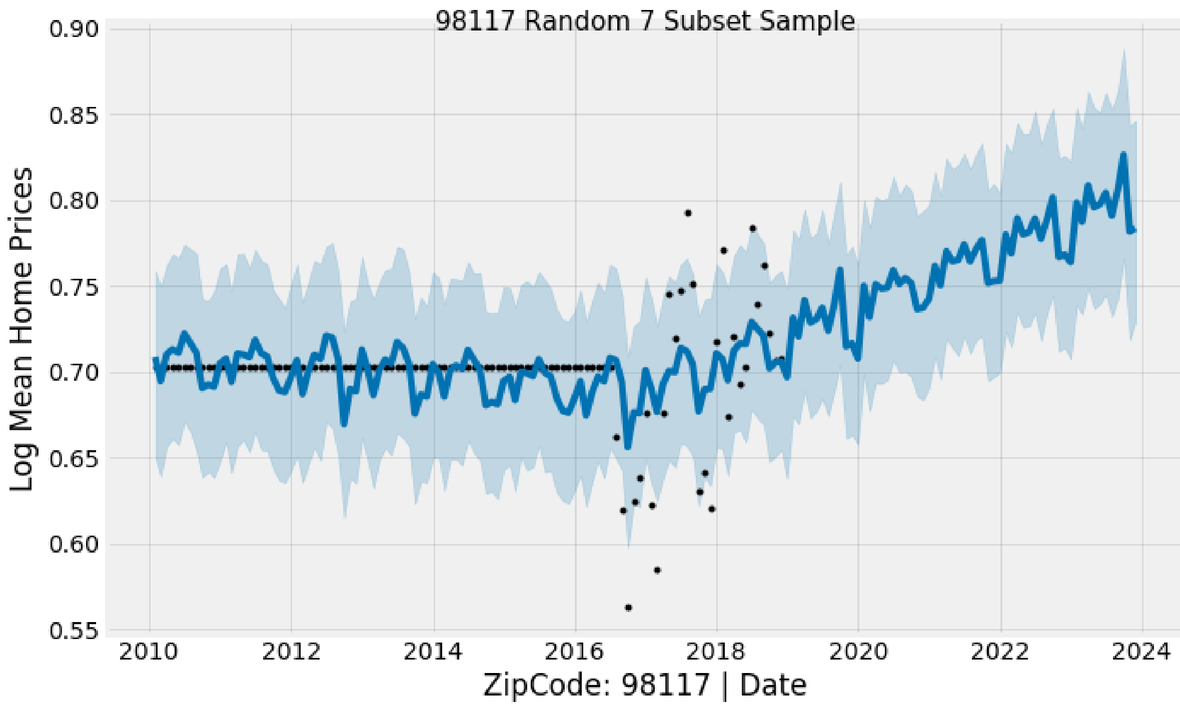


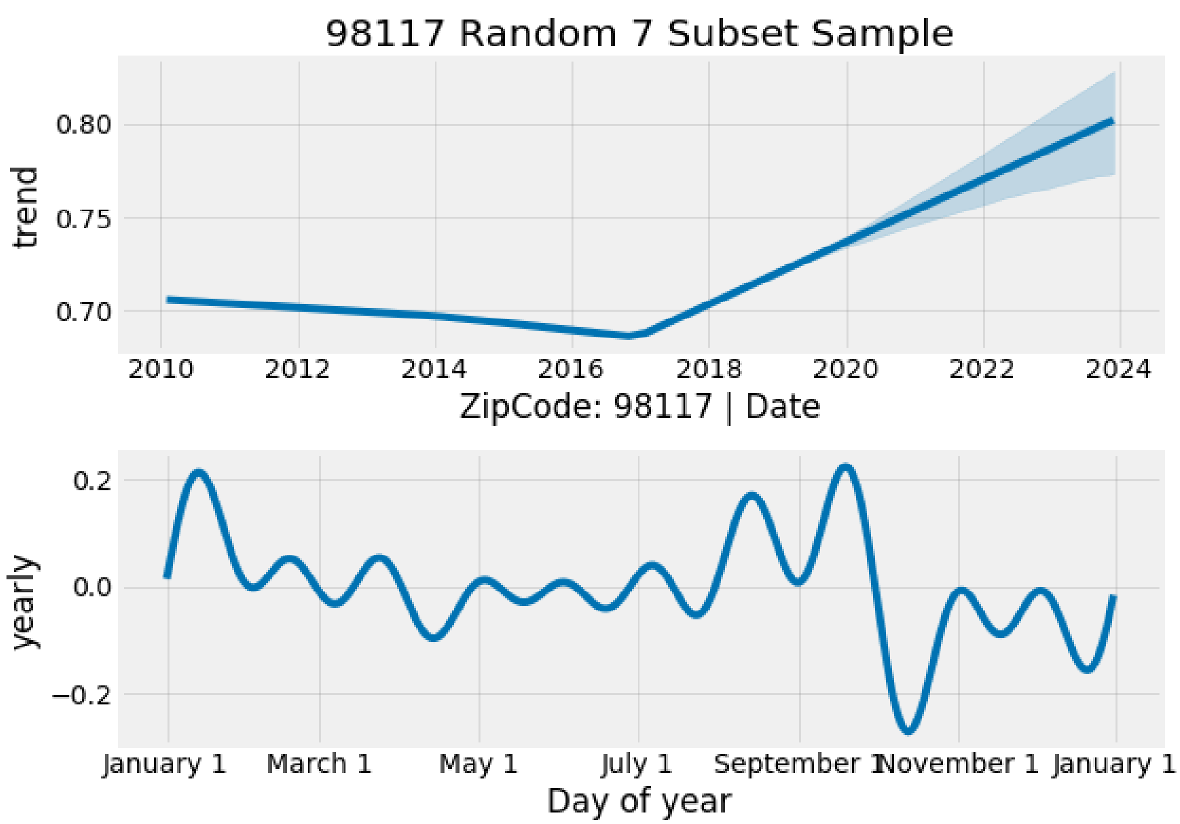
<Figure size 1440x1080 with 0 Axes>



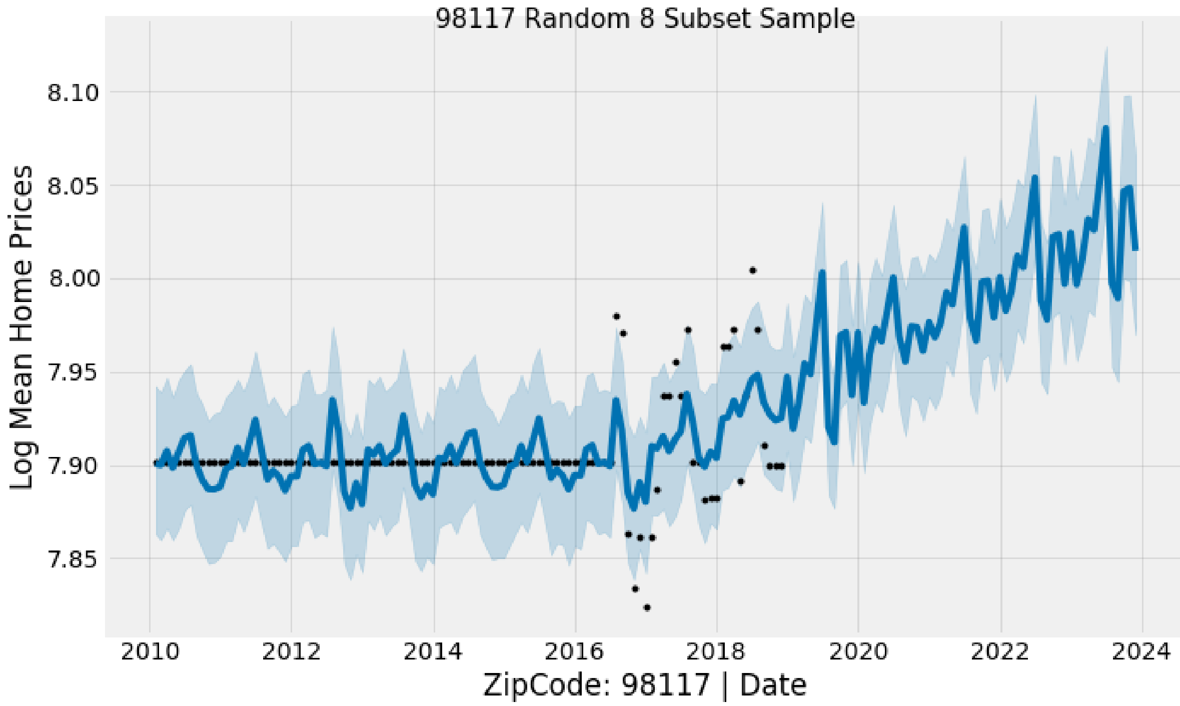


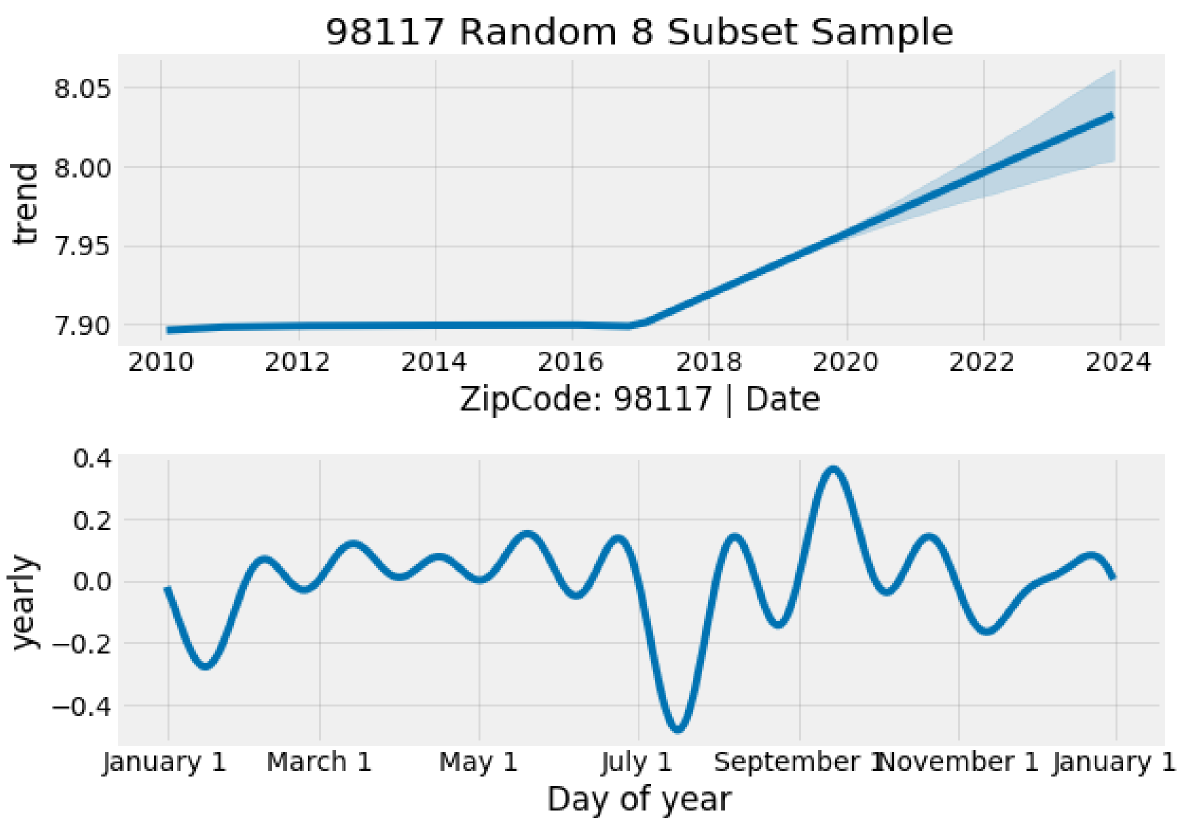

```
<Figure size 1440x1080 with 0 Axes>
```



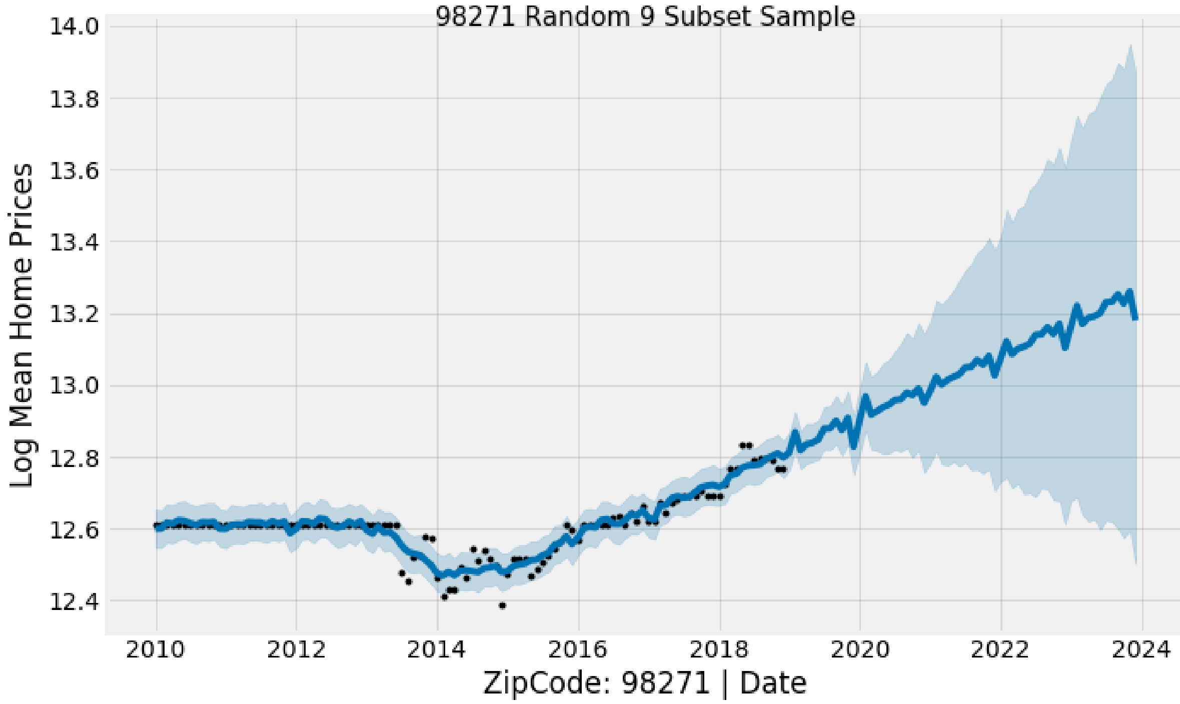


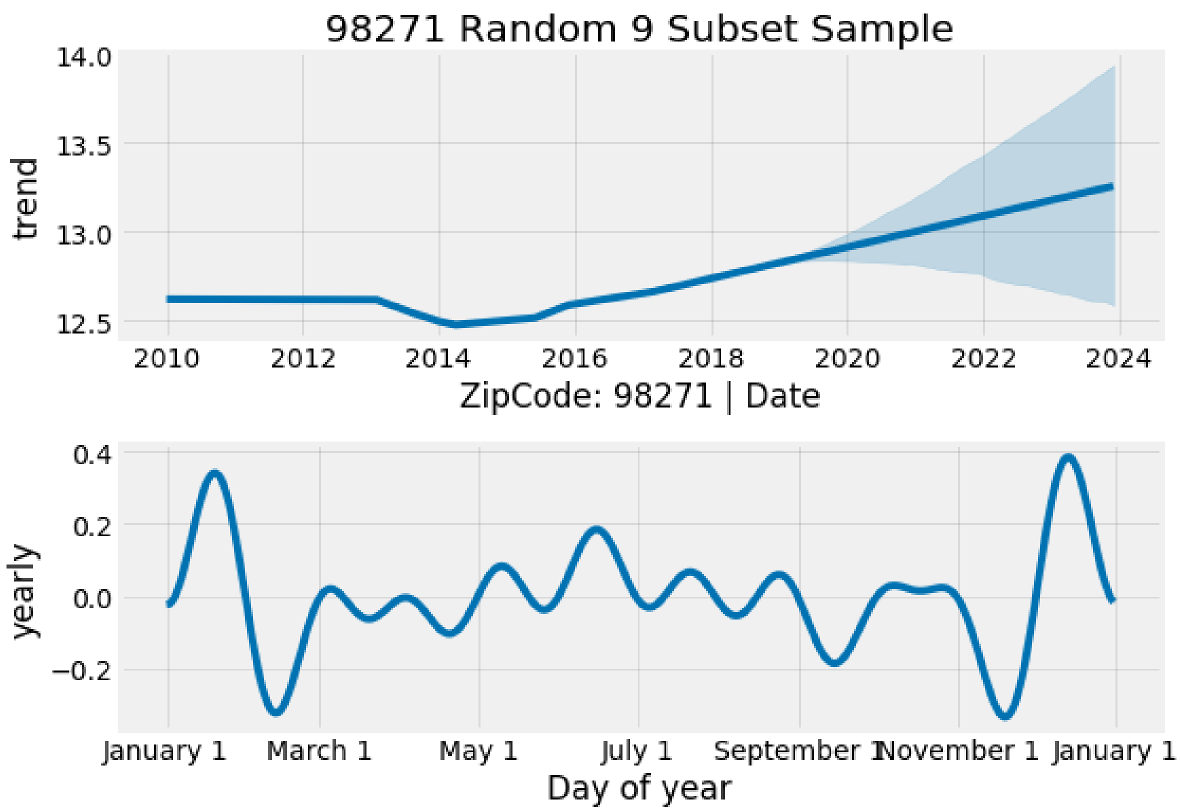

```
<Figure size 1440x1080 with 0 Axes>
```






```
<Figure size 1440x1080 with 0 Axes>
```


98271 Random 9 Subset Sample
Log Mean Home Prices
14.0
13.8
13.6
13.4
13.2
13.0
12.8
12.6
12.4
2010
2012
2014
2016
2018
2020
2022
2024
ZipCode: 98271 | Date


98271 Random 9 Subset Sample
trend
14.0
13.5
13.0
12.5
2010
2012
2014
2016
2018
2020
2022
2024
ZipCode: 98271 | Date
yearly
0.4
0.2
0.0
−0.2
January 1
March 1
May 1
July 1
September 1
November 1
January 1
Day of year

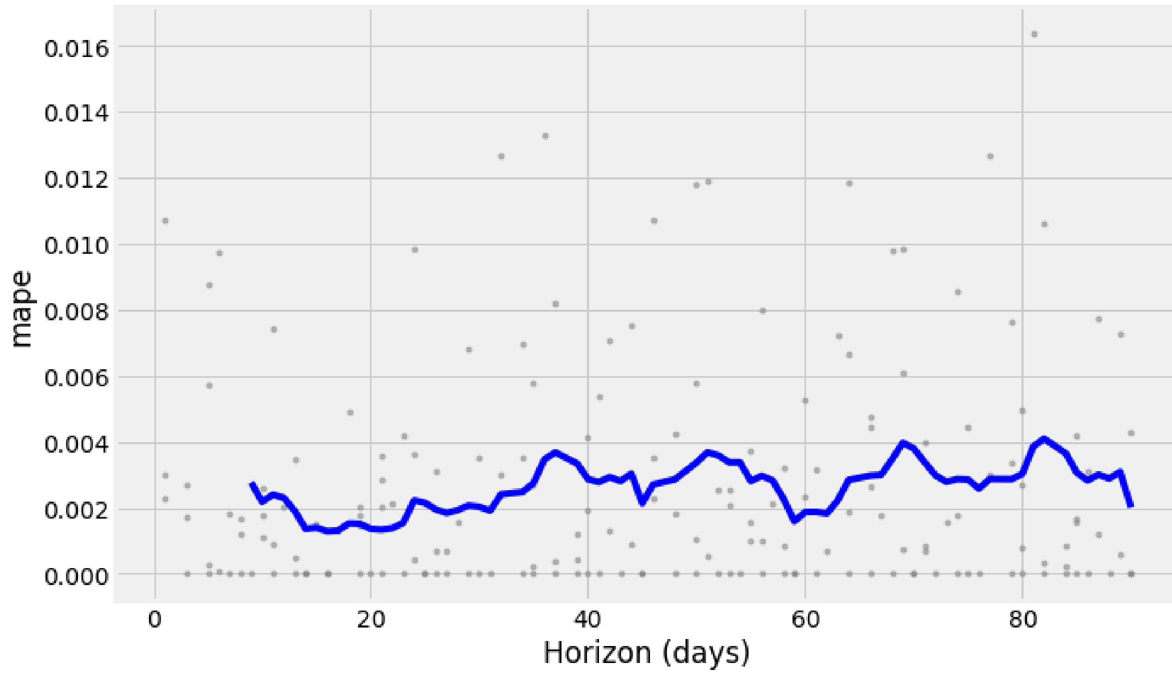
```
INFO:fbprophet:Making 63 forecasts with cutoffs between 2011-01-12 00:00:00 a
nd 2018-09-02 00:00:00
INFO:fbprophet:n_changepoints greater than number of observations.Using 9.
INFO:fbprophet:n_changepoints greater than number of observations.Using 10.
INFO:fbprophet:n_changepoints greater than number of observations.Using 11.
INFO:fbprophet:n_changepoints greater than number of observations.Using 12.
INFO:fbprophet:n_changepoints greater than number of observations.Using 14.
INFO:fbprophet:n_changepoints greater than number of observations.Using 15.
INFO:fbprophet:n_changepoints greater than number of observations.Using 16.
INFO:fbprophet:n_changepoints greater than number of observations.Using 17.
INFO:fbprophet:n_changepoints greater than number of observations.Using 19.
INFO:fbprophet:n_changepoints greater than number of observations.Using 19.
INFO:fbprophet:n_changepoints greater than number of observations.Using 21.
INFO:fbprophet:n_changepoints greater than number of observations.Using 22.
INFO:fbprophet:n_changepoints greater than number of observations.Using 23.
INFO:fbprophet:n_changepoints greater than number of observations.Using 24.
```

Timeseries Models Performance Metrics - 90 days Horizon

Out[136]:

	horizon	mse	rmse	mae	mape	coverage
0	9 days	0.003058	0.055295	0.034677	0.002765	0.722222
1	10 days	0.002039	0.045157	0.027378	0.002181	0.777778
2	11 days	0.002459	0.049593	0.030191	0.002397	0.777778
3	12 days	0.002257	0.047510	0.029036	0.002303	0.805556
4	13 days	0.001648	0.040594	0.024076	0.001906	0.888889

Timeseries Models Cross Validation Metric Mape - 90 day Horizon**



3. Clustering

- K-means - unsupervised
- Mean-Shift - unsupervised

Description: Run k-means for three choices for k and choose the best.
A loop of 10 iterations were ran of the zipecode models generated from the Timeseries process ran above. Based on the output of the Elbow technique K=4 was the best chosen choose.

Intent: Try and use unsupervised learing techniques to classify Timeseries models produced by prophet. Which are the best forecasters?

- Try and group into 3 classes using unsupervised learning
- Focus on single familey homes

3.1 K-means Clustering

Python package: scikit-learn v0.21.3 [sklearn.cluster.KMeans (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans)](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans)

Description: ...

3.1.1 Analysis

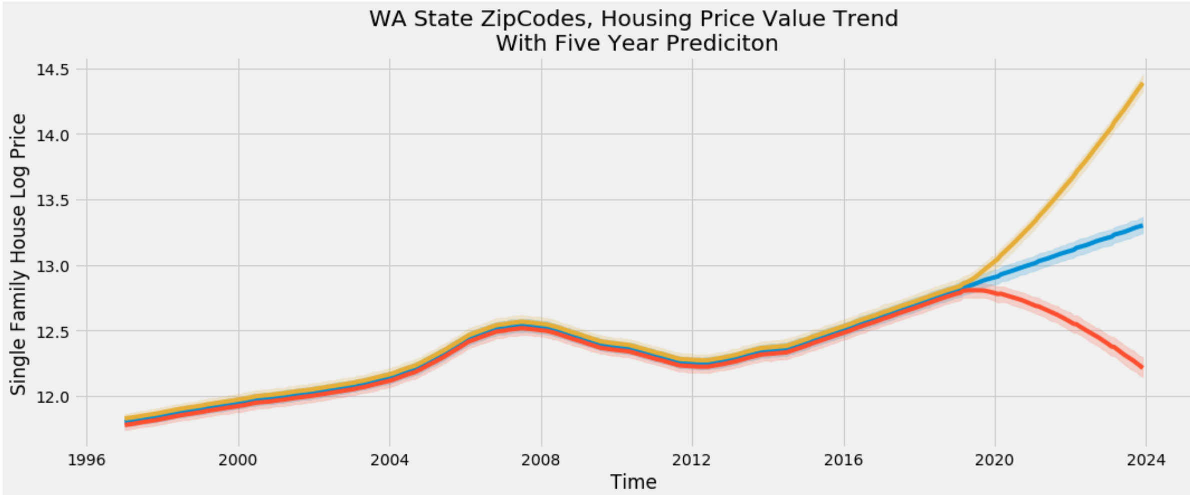
3.1.2 Exploration

Get all of the zip code forecast prediction models that generated in section 2 from disc, and prep for kmeans

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 113724 entries, 0 to 323
Data columns (total 17 columns):
ds                            113724 non-null datetime64[ns]
trend                         113724 non-null float64
yhat_lower                    113724 non-null float64
yhat_upper                    113724 non-null float64
trend_lower                   113724 non-null float64
trend_upper                   113724 non-null float64
additive_terms                113724 non-null float64
additive_terms_lower          113724 non-null float64
additive_terms_upper          113724 non-null float64
yearly                        113724 non-null float64
yearly_lower                  113724 non-null float64
yearly_upper                  113724 non-null float64
multiplicative_terms          113724 non-null float64
multiplicative_terms_lower    113724 non-null float64
multiplicative_terms_upper    113724 non-null float64
yhat                          113724 non-null float64
ZipCode                       113724 non-null object
dtypes: datetime64[ns](1), float64(15), object(1)
memory usage: 15.6+ MB
```

Out[145]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additiv
319	2023-07-31	13.155514	11.837246	14.674990	11.853864	14.682746	-0.007654	
320	2023-08-31	13.161885	11.835639	14.753381	11.823744	14.737809	-0.003536	
321	2023-09-30	13.168051	11.815777	14.783577	11.801436	14.793408	0.007843	
322	2023-10-31	13.174423	11.787066	14.854134	11.784665	14.850861	0.012167	
323	2023-11-30	13.180589	11.754081	14.922188	11.741391	14.906460	0.006648	



Clean the forecast dataset for clustering

- limite the features for clustering - and the observations to just the predition time (5 years) + one year observed
- remove additive terms and multiplicative terms as well as the datetimestamp
- save series objects for later re joining

Out[150]:

	ds	ZipCode	yhat	yhat_lower	yhat_upper	trend	trend_lower	trend_upper
252	2018-01-01	98052	13.675667	13.655640	13.696859	13.673974	13.673974	13.673974
253	2018-02-01	98052	13.685759	13.666272	13.705942	13.683988	13.683988	13.683988
254	2018-03-01	98052	13.695788	13.676042	13.716392	13.693034	13.693034	13.693034
255	2018-04-01	98052	13.705396	13.683485	13.724888	13.703048	13.703048	13.703048
256	2018-05-01	98052	13.715385	13.695982	13.736280	13.712740	13.712740	13.712740

Pull in generated datasets for modeling...

- Economic Date - economic_forecast_date_norm.csv --- economic factors per date
- Real Esate Date - sfr_price_zip.csv --- single family homes price value per zipcode and date

```
INFO:file_logger:economic_norm_date shape: (261, 11)
INFO:file_logger:realestate_sfr_prices shape: (264, 352)
```

Out[152]:

	Date	98052	98012	99301	98103	98682	98115	98122	98133	992(
0	1997-01-01	229300.0	199500.0	88700.0	183800.0	131100.0	191700.0	175800.0	155400.0	117000
1	1997-02-01	231400.0	200700.0	88600.0	185500.0	131400.0	193500.0	177300.0	156300.0	117300
2	1997-03-01	233500.0	202000.0	88400.0	187200.0	131500.0	195200.0	178700.0	157100.0	117700
3	1997-04-01	235600.0	203300.0	88000.0	189100.0	131400.0	197000.0	180500.0	158100.0	118100
4	1997-05-01	237800.0	204600.0	87500.0	191200.0	131100.0	198800.0	182400.0	159100.0	118400

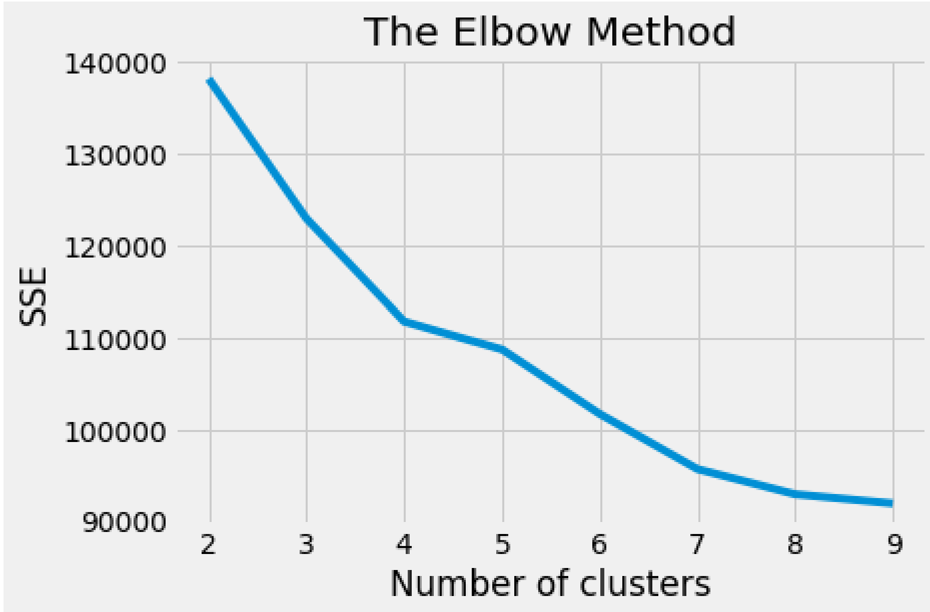
5 rows × 352 columns

Out[153]:

	Date	CPI_Index_Avg_f	Interest_Rate_f	Housing_Price_Index_f	Bond_Yeild_10y_f	Inflation_f
0	1997-01-01	157.959370	4.814434	83.076214	6.055381	2.812443
1	1997-02-01	158.404761	4.731693	83.392929	6.015761	2.812443
2	1997-03-01	158.947674	4.633084	82.836534	6.073349	2.812443
3	1997-04-01	159.387626	4.378458	83.893792	6.079290	2.812443
4	1997-05-01	159.765440	4.652420	85.006970	6.126489	2.812443

3.1.3 Model - KMeans

- Run multiple k means to determin optimal k size for final model creation
  - 8 iterations were ran, where k 4 was the most optimal

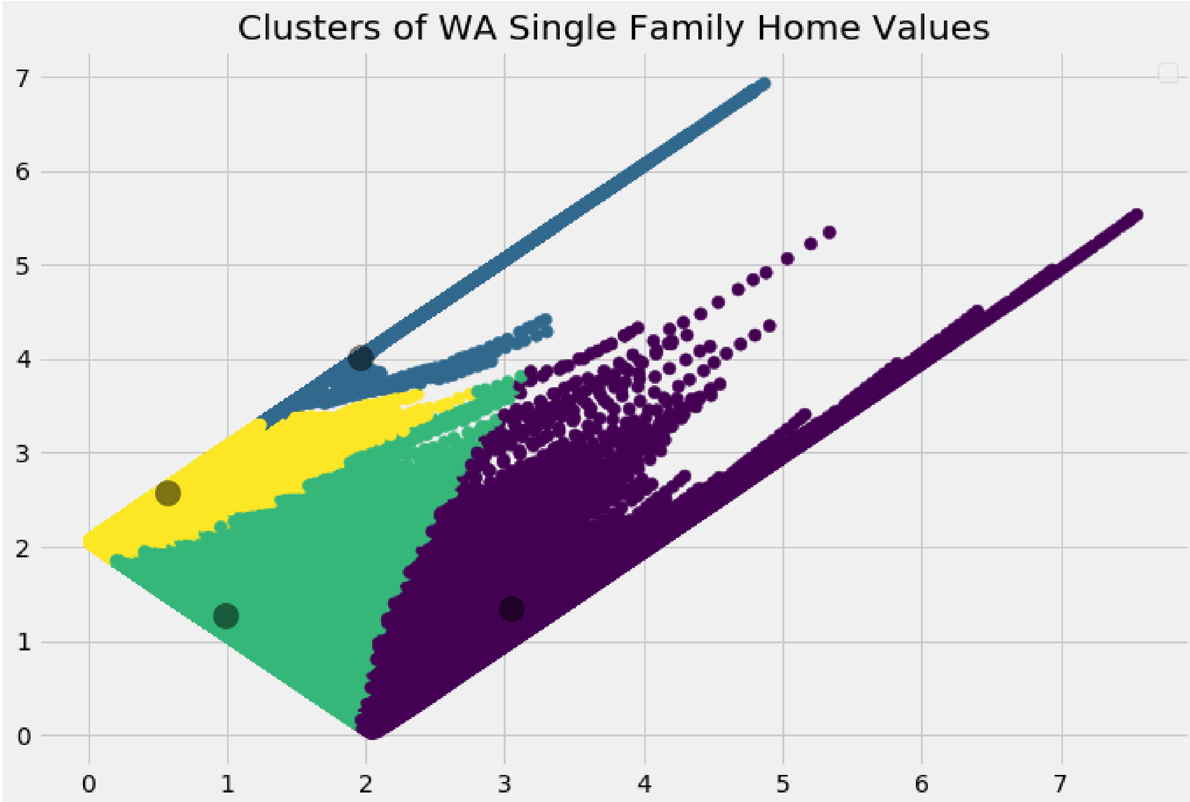


Build KMeans based on ideal cluster state found by Elbow method - 4

3.1.4 Results

Rusulting Cluster Classification at K equal 4

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```



4. Decision Tree

- Decision Tree - supervised
  - Include three different trees and their visualizations

Python package: [scikit-learn sklearn.tree.DecisionTreeClassifier (https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html)](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html)

*Build a decision tree model.

4.1 Analysis

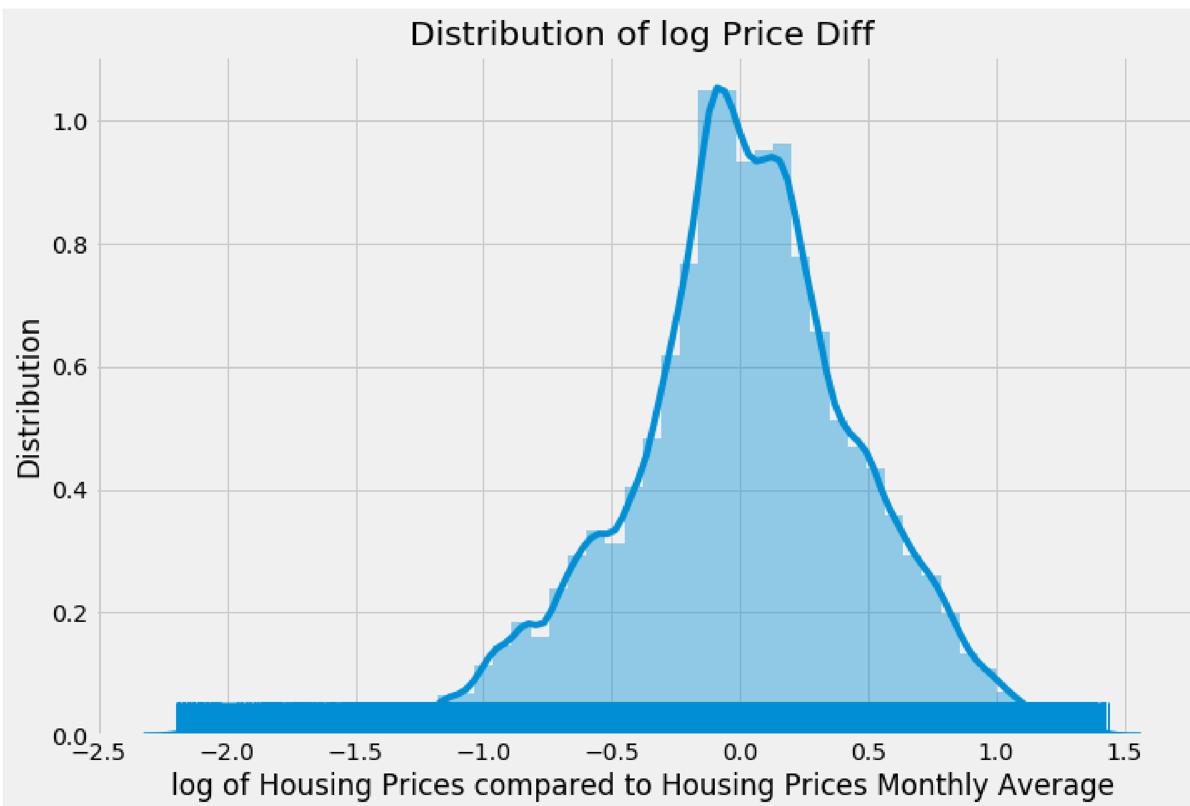
- Transformation of the data's necessary to merge the datasets together after processed through prophet.
- Look over the distribution of key features
- Set price thresholds for supervised learning classification
- Price_Point_Class is a generated feature for supervised classification. Details are shown below

transform this data set to be in the shape: columns are zip codes, yhat is the value as prices, date

Out[160]:

ZipCode	Date	98001	98002	98003	98004	98005	98006	98007	
0	1997-01-01	11.873703	11.731146	11.854656	12.930965	12.434717	12.474191	12.147111	12.
1	1997-02-01	11.880200	11.735203	11.860957	12.941784	12.446799	12.486282	12.159336	12.
2	1997-03-01	11.884774	11.735696	11.865059	12.947822	12.452669	12.491204	12.164220	12.
3	1997-04-01	11.891682	11.740681	11.871427	12.957818	12.464837	12.503393	12.176328	12.
4	1997-05-01	11.898385	11.745872	11.877825	12.967831	12.476572	12.515002	12.187934	12.

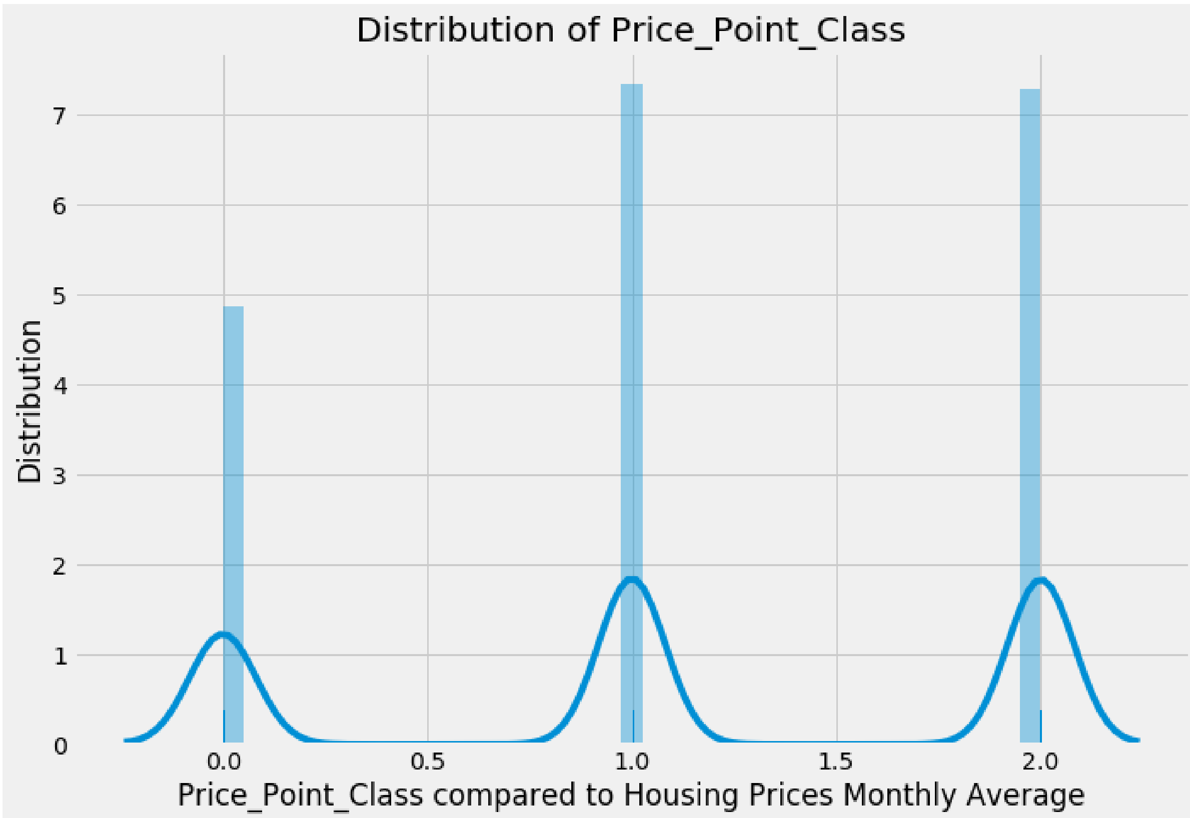
5 rows × 352 columns




```
INFO:file_logger:threshold_low [-0.25401387500000006] | threshold_neutral [0.
010042959999999823] | threshold_high [0.29538362500000037] | threshold_high_x
[0.36922953125000046] |            threshold_high_mid [0.14769181250000019]
```

Out[171]:

	Date	ZipCode	log_Price	log_Price_Monthly_Avg	log_Price_diff	Price_Point_Class
0	1997-01-01	98052	12.342486	11.804569	-0.537917	0
1	1997-02-01	98052	12.352806	11.808970	-0.543837	0
2	1997-03-01	98052	12.357526	11.811044	-0.546482	0
3	1997-04-01	98052	12.367903	11.815443	-0.552460	0
4	1997-05-01	98052	12.378131	11.819842	-0.558289	0



Out[179]: 351

Final Merged Dataset - Real Estate Combined with Economic Data Features

Time range - 1997 - 2017 (that was the cleanest that could be achieved at this time...
*Train classifiers on Feature 'Price_Point_Class'

- 0: means observation's price value is < 25% of the State Price Average
- 1: means observations fall within the normal (average) range of the State Price Average
- 2: means observations falls above the 75% range of the State Price Average

--Determin if classifiers can identify future home value classes based on prior date, location and economic features that have the most impact on both postive and negative price value swings...

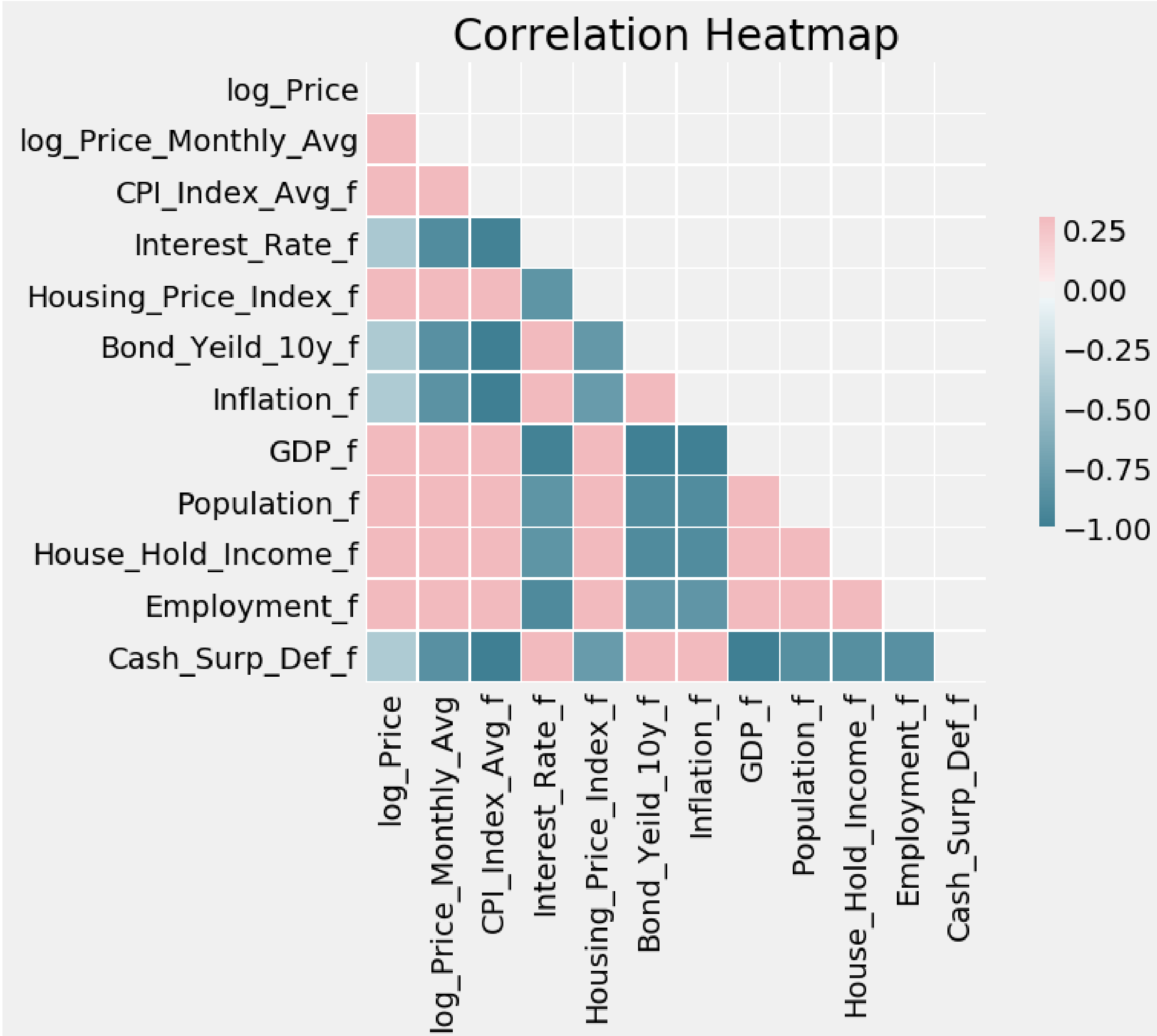
- Dataset Shape: (88452, 16)

Out[186]:

	Date	ZipCode	log_Price	log_Price_Monthly_Avg	log_Price_diff	Price_Point_Class	CPI_Inde
0	1997-01-01	98052	12.342486	11.804569	-0.537918	0	157
1	1997-02-01	98052	12.352806	11.808970	-0.543837	0	158
2	1997-03-01	98052	12.357526	11.811044	-0.546482	0	158
3	1997-04-01	98052	12.367903	11.815443	-0.552460	0	159
4	1997-05-01	98052	12.378131	11.819842	-0.558289	0	159

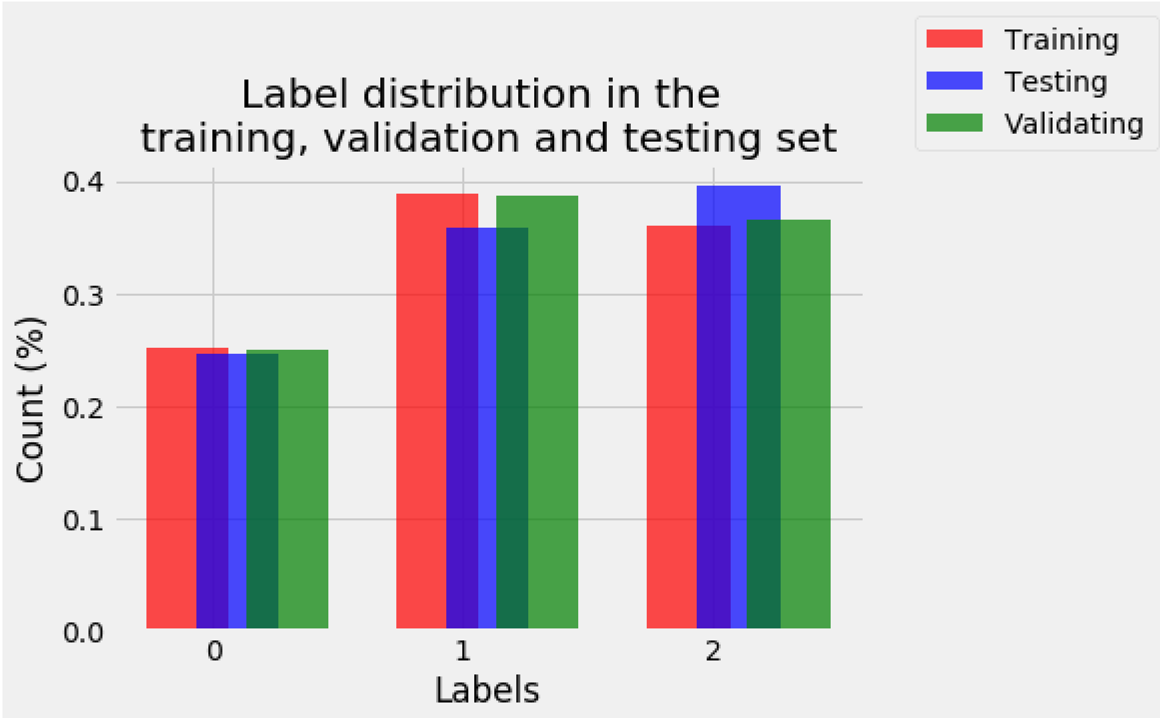
4.2 Exploration

```
<Figure size 576x432 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

Look for imbalance in the sample observations for the target class



4.3 Model - DecisionTree Classifier

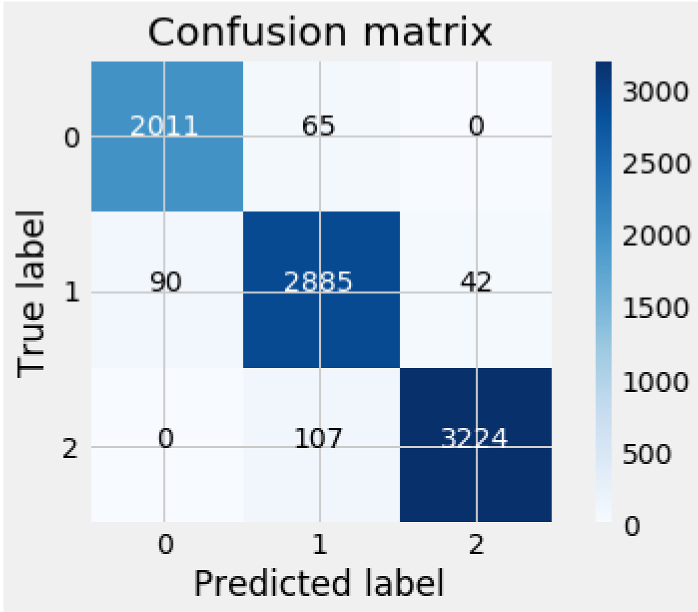
- max_depth: None (default)
- min_samples_split: 2
- randome_state: 42

```
INFO:file_logger:DecisionTreeClassifier Model Build Time: [0.355584300001282
8]

INFO:file_logger:DecisionTreeClassifier Model Fit Score: [0.9914423584407751]
INFO:file_logger:DecisionTreeClassifier Model Fit Score Time: [0.039111999998
567626]

INFO:file_logger:DecisionTreeClassifier Predict Time: [0.003885100000843522]
```

4.4 Results



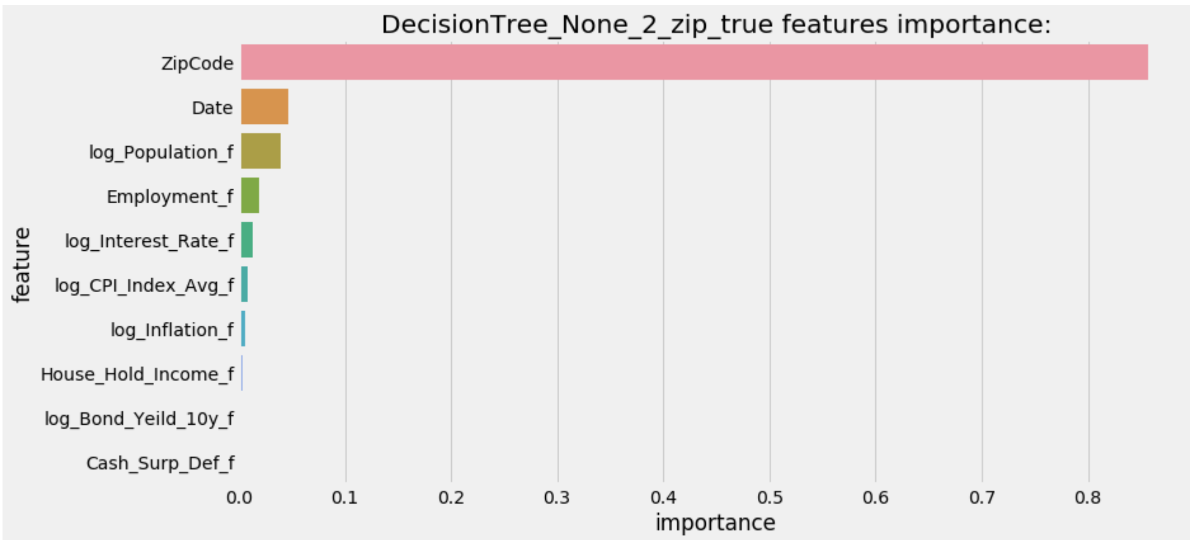

```
              precision    recall  f1-score   support

      Class0       0.96      0.97      0.96      2076
      Class1       0.94      0.96      0.95      3017
      Class2       0.99      0.97      0.98      3331

   micro avg       0.96      0.96      0.96      8424
   macro avg       0.96      0.96      0.96      8424
weighted avg       0.96      0.96      0.96      8424
```

```
INFO:file_logger:                  feature  importance
1                ZipCode    0.856557
0                   Date    0.046343
7        log_Population_f    0.040113
9            Employment_f    0.019456
3      log_Interest_Rate_f    0.012984
2      log_CPI_Index_Avg_f    0.008199
5         log_Inflation_f    0.006408
8      House_Hold_Income_f    0.003751
4      log_Bond_Yeild_10y_f    0.002397
10          Cash_Surp_Def_f    0.001922
```

```
<Figure size 432x288 with 0 Axes>
```



4.5 Random Forest Classifier

Python Package: scikit-learn v0.21.3 [sklearn.ensemble.RandomForestClassifier (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

- n_estimators: 100
- max_depth: None (default)
- min_samples_plit=2

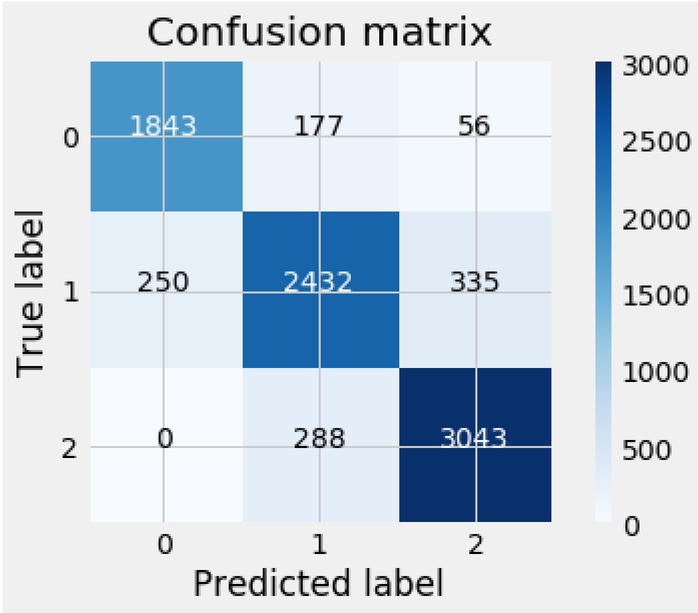
```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    6.8s finished
INFO:file_logger:Random Forest Classification Model Build Time: [7.0466736999
98792]
```

Out[210]:
```
{'bootstrap': True,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 1,
 'warm_start': False}
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.8s finished
INFO:file_logger:Random Forest Base Classification Model Fit Score: [0.643708
0600165875]
INFO:file_logger:Random Forest Base Classification Model Fit Score Time: [0.9
830512999997154]
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent worke
rs.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    0.0s finished
INFO:file_logger:Random Forest Base Classification Predict Time: [0.127537899
9998793]
```

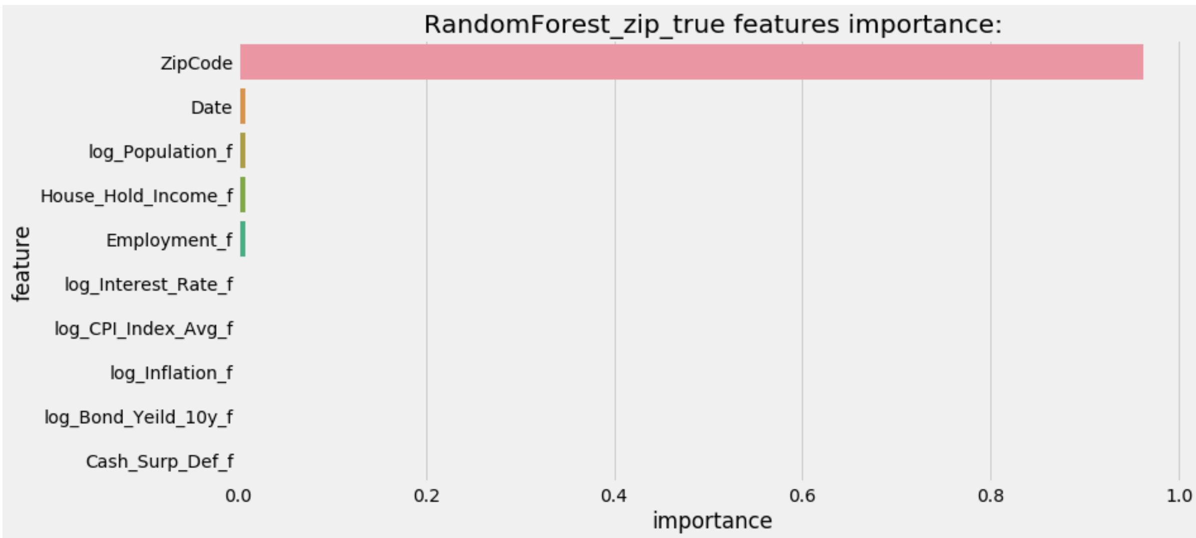
4.5.2 Randome Forest Results



	precision	recall	f1-score	support
Class0	0.88	0.89	0.88	2076
Class1	0.84	0.81	0.82	3017
Class2	0.89	0.91	0.90	3331
micro avg	0.87	0.87	0.87	8424
macro avg	0.87	0.87	0.87	8424
weighted avg	0.87	0.87	0.87	8424

```
INFO:file_logger:                  feature  importance
1                ZipCode    0.961918
0                   Date    0.008418
7       log_Population_f    0.008186
8     House_Hold_Income_f   0.007558
9           Employment_f    0.007557
3     log_Interest_Rate_f   0.002031
2     log_CPI_Index_Avg_f   0.001115
5         log_Inflation_f   0.001005
4    log_Bond_Yeild_10y_f   0.000770
10        Cash_Surp_Def_f   0.000725

<Figure size 432x288 with 0 Axes>
```



5. Naive Bayes

Python Package: SciKit-Learn [Gaussian Naive Bayes (https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes)](https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes)
Build a naïve Bayes model. Tune the parameters, such as the discretization options, to compare results.

5.1 Analysis - Naive Bayes

5.2 Exploration - Naive Bayes

5.3 Model - Naive Bayes

- priors: None (default)

```
INFO:file_logger:GNB Model Build Time: [0.08762639999986277]
```

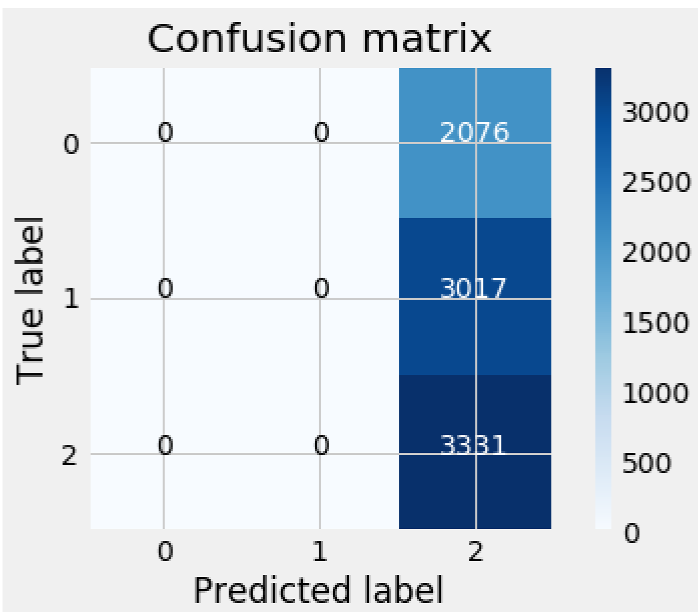
```
INFO:file_logger:GNB Fit Score: [0.38724270527030086]
INFO:file_logger:GNB Score Time: [0.049470199999632314]
```

```
INFO:file_logger:GNB Model Predict Time: [0.007201399999757996]
```

```
Wall time: 8.98 ms
```

5.4 Results

```
INFO:file_logger:Percent Accurately Labeled: [-5092.0]
```



	precision	recall	f1-score	support
Class0	0.00	0.00	0.00	2076
Class1	0.00	0.00	0.00	3017
Class2	0.40	1.00	0.57	3331
micro avg	0.40	0.40	0.40	8424
macro avg	0.13	0.33	0.19	8424
weighted avg	0.16	0.40	0.22	8424

6. Support Vector Classification - SVMs

Python Package: scikit-learn v0.21.3 [sklearn.svm.SVC (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC)](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC)

6.1 Analysis

6.2 Exploration

6.3 Model - SVM

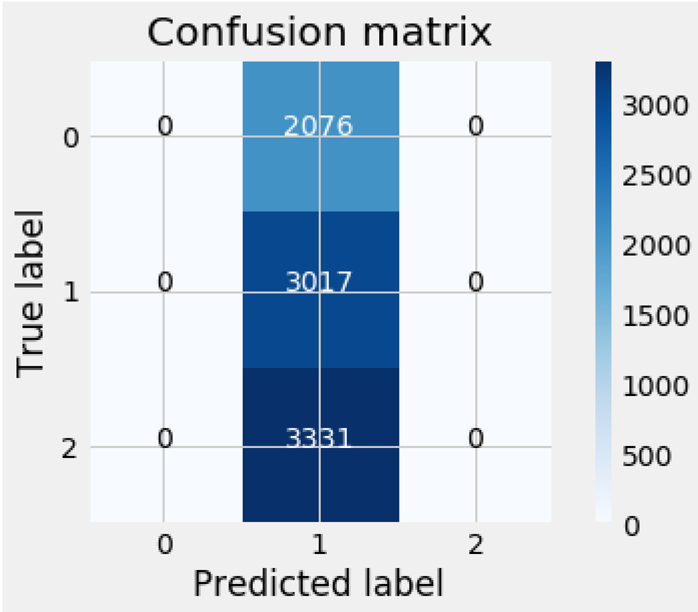
- Three rounds with different kernel's being evaluated
  - 1st: rbf
    - Results:
      - Class1 - Best f1-score of .53 `![image.png](attachment:image.png)`
  - 2nd: poly
    - Results:
      - Class2 - Best f1-score of .57 `![image.png](attachment:image.png)`
  - 3rd: sigmoid
    - Results:
      - Class1 - Best f1-score of .53 `![image.png](attachment:image.png)`

```
[LibSVM]

INFO:file_logger:SupportVectorClassifier Model SupportVector_sigmoid_zip_fals
e Build Time: [89.3771115999989]

INFO:file_logger:Support Vector Classification Model SupportVector_sigmoid_zi
p_false Fit Score: [0.3864133303174244]
INFO:file_logger:Support Vector Classification Model SupportVector_sigmoid_zi
p_false Fit Score Time: [25.591142700001]

INFO:file_logger:Support Vector Classification SupportVector_sigmoid_zip_fals
e Predict Time: [7.811279999999897]
```




```
              precision    recall  f1-score   support

      Class0       0.00      0.00      0.00      2076
      Class1       0.36      1.00      0.53      3017
      Class2       0.00      0.00      0.00      3331

   micro avg       0.36      0.36      0.36      8424
   macro avg       0.12      0.33      0.18      8424
weighted avg       0.13      0.36      0.19      8424
```

6.4 Results

Out[252]:
```
{'ModelName': ['DecisionTree_None_2_zip_true',
  'RandomForest_zip_true',
  'NaiveBayes_zip_false',
  'NaiveBayes_zip_true',
  'SupportVector_rbf_zip_false',
  'SupportVector_poly_zip_false',
  'SupportVector_sigmoid_zip_false'],
 'TestAccuracyScore': [0.9914423584407751,
  0.6437080600165875,
  0.38724270527030086,
  0.38724270527030086,
  0.5570383774410013,
  0.36454799065068233,
  0.36454799065068233,
  0.3864133303174244],
 'PredictAccuracyScore': [-5092.0],
 'FitTime': [0.3555843000012828,
  7.046673699998792,
  0.08719380000002275,
  0.08762639999986277,
  284.62899259999904,
  0.10550850000072387,
  89.3771115999989],
 'ScoreTime': [0.04137609999997949,
  0.9830512999997154,
  0.0912466999998287,
  0.049470199999632314,
  95.96742810000069,
  0.045794699999532895,
  0.04900169999928039,
  25.591142700001],
 'PredictTime': [0.003885100000843522,
  0.1275378999998793,
  0.007201399999757996,
  29.593625200001043,
  0.009665300000051502,
  7.811279999999897]}
```

8. Final Results & Conclusion

Real estate housing market trends are impacted by many factors that require deep data mining techniques and domain experts to pull the right data together and engineer it in meaningful ways to gain insights into this industry. Data proved to be the most challenging component of this research. There is a lack of quality datasets that are easily found which inhibits possible discoveries.

Certainly economic indicators are present that signal swings in price trends... Further research on comprehensive, state level economics is needed to expand on the datasets used in this study, which were at the national level. Most likely it's this that caused the inconsistencies with the models performance. The Real estate data being focused on was at the state level, whereas the economic data was at the national yearly average. This abstraction could have been a leading cause.