Laboratory Exercise – 2

IST 718 Big Data Analytics

Dan Caley

Table of Contents

Disclaimer

The bonus section was completed which included not just the Median house price mapped across the United States but also included Census population and household median income both at the County and State level.

To see all the analysis please look the Jupyter Notebook.

Obtain

There were 4 datasets used in performing this analysis:

1. Zillow Static Data set found at https://files.zillowstatic.com/research/public/Zip/Zip_Zhvi_SingleFamilyResidence.csv
2. Zip Code Tabulation Area (ZCTA) Household Median Income and Population level data from the Census Bureau.
3. County Household Median Income and Population level data from the Census Bureau.
4. State Household Median Income and Population level data from the Census Bureau.
5. State code data mapping to plot on a geographic map.

Zillow Data

To obtain the Zillow data set the Pandas read csv function was used by inserting the above url.

```
In [3]: zillow = pd.read_csv("https://files.zillowstatic.com/research/public/Zip/Zip_Zhvi_SingleFamilyResidence.csv")
        zillow.head()
```

Out[3]:

	RegionID	SizeRank	RegionName	RegionType	StateName	State	City	Metro	CountyName	1996-01-31	...	2019-06-30	2019-07-31	2019-08-31	201
0	61639	0	10025	Zip	NY	NY	New York	New York-Newark-Jersey City	New York County	NaN	...	1413747.0	1405862.0	1402547.0	1390
1	84654	1	60657	Zip	IL	IL	Chicago	Chicago-Naperville-Elgin	Cook County	364892.0	...	974693.0	975616.0	975734.0	975
2	61637	2	10023	Zip	NY	NY	New York	New York-Newark-Jersey City	New York County	NaN	...	1528603.0	1514894.0	1502233.0	1492

Census Data

To obtain the census data, the following code lines were used:

1. Ping the census bureau api for Household Median Income and Population by year
2. Append the data to a dataframe
3. Loop through 2011 – 2020, this is everything that the census bureau has.
4. Rename the headers to Median Income and Population
5. Create a Pandas Dataframe

In [8]:
```
acs.printtable(acs.censustable('acs5', 2009, 'B19013'))
acs.printtable(acs.censustable('acs5', 2009, 'B01003'))
```

```
Variable     | Table                            | Label                                                    | Type
-----------------------------------------------------------------------------------------------------------------
B19013_001E  | MEDIAN HOUSEHOLD INCOME IN THE | !! Estimate Median household income in the past 12 month | int
-----------------------------------------------------------------------------------------------------------------
Variable     | Table                            | Label                                                    | Type
-----------------------------------------------------------------------------------------------------------------
B01003_001E  | TOTAL POPULATION                 | !! Estimate Total                                        | int
-----------------------------------------------------------------------------------------------------------------
```

Downloading American Census Data by Zipcode

In [9]:
```
acs_data = acs.download('acs5', 2011, acs.censusgeo([('zip code tabulation area', '*')]),
                                   ['B19013_001E', 'B01003_001E'])
```

In [10]:
```
census_year = list(range(2011,2020))
census_pull = pd.DataFrame()

for y in census_year:
  acs_data = acs.download('acs5', y, acs.censusgeo([('zip code tabulation area', '*')]),
                                   ['B19013_001E', 'B01003_001E'])
  cbd = pd.DataFrame(acs_data)
  cbd['year'] = y
  census_pull = census_pull.append(cbd)
```

In [14]:
```
acs.exportcsv('census_data.csv', census_data)
census_data = pd.read_csv('census_data.csv')
census_data.head()
```

Out[14]:

	state	zip code tabulation area	NAME	median_income	population	year
0	72	601	ZCTA5 00601	13318.0	18533	2011
1	72	602	ZCTA5 00602	14947.0	41930	2011
2	72	603	ZCTA5 00603	14437.0	54475	2011
3	72	606	ZCTA5 00606	11155.0	6386	2011
4	72	610	ZCTA5 00610	16367.0	29111	2011

This data frame shows the House Hold Median Income by year. We can clearly see that there is some data to be cleaned with every min is -66666, representing 0. Due to this the average gets thrown off. The median Household income can be seen at the 50%. By 2019 this increased by $8k. If accurate this track with inflation. Meaning inflation on average is 2%, over the course of 10 years that comes out to roughly a little above 20%

	median_income							
	count	mean	std	min	25%	50%	75%	max
year								
2011	33120.0	-1.852902e+07	1.097405e+08	-666666666.0	36595.00	46354.0	59471.00	250001.0
2012	33120.0	-1.744154e+07	1.065713e+08	-666666666.0	36875.00	46775.0	59821.00	250001.0
2013	33120.0	-2.279601e+07	1.212914e+08	-666666666.0	36944.00	46926.5	59938.50	250001.0
2014	33120.0	-2.233244e+07	1.200994e+08	-666666666.0	37285.50	47529.0	60625.00	250001.0
2015	32157.0	-2.254584e+07	1.206529e+08	-666666666.0	38095.00	48333.0	61339.00	250001.0
2016	33120.0	-4.036731e+07	1.591133e+08	-666666666.0	37857.00	48929.0	62188.00	250001.0
2017	33120.0	-4.350572e+07	1.647643e+08	-666666666.0	39005.00	50635.5	64560.75	250001.0
2018	33085.0	-4.415424e+07	1.659040e+08	-666666666.0	40595.00	52500.0	66910.00	250001.0
2019	33120.0	-4.621912e+07	1.694563e+08	-666666666.0	41899.25	54250.0	69583.00	250001.0

	population							
	count	mean	std	min	25%	50%	75%	max
year								
2011	33120.0	9369.842512	13669.672456	0.0	716.00	2792.0	12838.00	114941.0
2012	33120.0	9445.567693	13807.690410	0.0	720.75	2786.0	12952.00	115538.0
2013	33120.0	9516.959994	13939.177211	0.0	721.00	2801.5	13000.00	114734.0
2014	33120.0	9593.274607	14090.093299	0.0	717.00	2805.5	13066.00	115013.0
2015	33120.0	9664.375151	14237.949376	0.0	718.75	2808.0	13139.25	114982.0
2016	33120.0	9724.409300	14358.657599	0.0	718.00	2807.5	13177.75	115104.0
2017	33120.0	9796.435085	14510.547644	0.0	707.00	2804.0	13290.25	119204.0
2018	33120.0	9851.278865	14614.856872	0.0	705.00	2803.5	13378.50	122814.0
2019	33120.0	9903.343961	14714.043400	0.0	705.75	2801.0	13475.25	128294.0

This process was repeated for both the County and State level.

The mapping for State Name and State abbreviation for to map the geographic visualizations were manually inserted.

```
In [24]: code = {'Alabama': 'AL',
                 'Alaska': 'AK',
                 'Arizona': 'AZ',
                 'Arkansas': 'AR',
                 'California': 'CA',
                 'Colorado': 'CO',
                 'Connecticut': 'CT',
```

Scrub

Zillow Data

For the Zillow Data the following was needed to be scrubbed:

1. Zip code needed to have leading 0's. Meaning a Zip Code is 5 digits with 0 at the front in some instances
2. The dataframe was melted to have the dates as rows instead of columns
3. The date name and the values were then called date_zestimate and zestimate.
4. NaN were dropped from the dataset completely. Due to having so much data losing about 25% wasn't a huge hit like normal datasets.

Before

```
In [26]: zillow.head()
Out[26]:
```

	RegionID	SizeRank	RegionName	RegionType	StateName	State	City	Metro	CountyName	1996-01-31	...	2019-06-30	2019-07-31	2019-08-31	201
0	61639	0	10025	Zip	NY	NY	New York	New York-Newark-Jersey City	New York County	NaN	...	1413747.0	1405862.0	1402547.0	1390
1	84654	1	60657	Zip	IL	IL	Chicago	Chicago-Naperville-Elgin	Cook County	364892.0	...	974693.0	975616.0	975734.0	975
2	61637	2	10023	Zip	NY	NY	New York	New York-Newark-Jersey City	New York County	NaN	...	1528603.0	1514894.0	1502233.0	1492
3	91982	3	77494	Zip	TX	TX	Katy	Houston-The Woodlands-Sugar Land	Harris County	200475.0	...	335536.0	335878.0	335940.0	336
4	84616	4	60614	Zip	IL	IL	Chicago	Chicago-Naperville-Elgin	Cook County	546663.0	...	1207765.0	1208853.0	1208481.0	1206

5 rows × 300 columns

After

In [27]:
```
idvars = ['RegionID', 'SizeRank', 'RegionName', 'RegionType', 'StateName','State', 'City', 'Metro', 'CountyName']
zillow = zillow.melt(id_vars= idvars, var_name = 'date_zestimate', value_name='zestimate')

zillow = zillow.rename(columns = {'RegionName': 'Zipcode'})
zillow = zillow.drop(columns = ['RegionType','StateName'])
```

In [28]:
```
zillow
```

Out[28]:

	RegionID	SizeRank	Zipcode	State	City	Metro	CountyName	date_zestimate	zestimate
0	61639	0	10025	NY	New York	New York-Newark-Jersey City	New York County	1996-01-31	NaN
1	84654	1	60657	IL	Chicago	Chicago-Naperville-Elgin	Cook County	1996-01-31	364892.0
2	61637	2	10023	NY	New York	New York-Newark-Jersey City	New York County	1996-01-31	NaN
3	91982	3	77494	TX	Katy	Houston-The Woodlands-Sugar Land	Harris County	1996-01-31	200475.0
4	84616	4	60614	IL	Chicago	Chicago-Naperville-Elgin	Cook County	1996-01-31	546663.0
...	...	...	...	...	...	...	...	...	...
8865019	58111	35187	802	UT	Charlotte Amalie	NaN	Kane County	2020-03-31	132127.0
8865020	58115	35187	820	LA	Choudrant	Ruston	Lincoln Parish	2020-03-31	100708.0
8865021	58117	35187	822	LA	Choudrant	Ruston	Lincoln Parish	2020-03-31	181195.0
8865022	58121	35187	831	AL	Logan	Cullman	Cullman County	2020-03-31	75464.0
8865023	58125	35187	851	CO	Granby	NaN	Grand County	2020-03-31	456250.0

8865024 rows × 9 columns

Dropped Data Results:

In [47]:
```
zillow_clean = zillow_census.dropna(subset=['zestimate'])
print(1- len(zillow_clean) / len(zillow_census))
len(zillow_clean)
```

```
0.24057676455999066
```

Out[47]: 6662900

Census Data

For the census data the following was needed to scrubbed:

1. The ZCTA data was pretty clean after obtaining the data.
2. The county data needed to be reformatted to create FIPS codes which is just the state code concatenated with the county code.

In [22]:
```
census_county['state'] = census_county['state'].apply(lambda x: '{0:0>2}'.format(x))
census_county['county'] = census_county['county'].apply(lambda x: '{0:0>3}'.format(x))
```

In [23]:
```
census_county["FIPS"] = census_county["state"].astype(str) + census_county["county"].astype(st
census_county.head()
```

Out[23]:

	state	county	median_income	population	year	county_name	state_name	FIPS
0	37	043	36711.0	10506	2011	Clay County	North Carolina	37043
1	37	051	44861.0	316478	2011	Cumberland County	North Carolina	37051
2	37	081	46288.0	483081	2011	Guilford County	North Carolina	37081
3	37	099	36826.0	39574	2011	Jackson County	North Carolina	37099
4	37	139	45298.0	40511	2011	Pasquotank County	North Carolina	37139

Merging Zillow and Census Data

Merged the Zillow data with the census data. This is joining Census onto Zillow. The Zip Code in Zillow is a copy right of the postal service and the ZCTA is a copyright of the census bureau. These two Zip Codes are different, so this is not a perfect match. There is a mapping file but for this analysis combining on the Zillow Zip Code will be sufficient due to only 2% of the Zip Codes were unable to be mapped. This is opportunity to improve the precision of the analysis.

In [40]:
```
zillow_census = pd.merge(zillow, census_data,
                         how="left",
                         left_on = ["Zipcode","year"],
                         right_on = ["Zipcode","year"])

zillow = None
census_data = None
```

In [41]:
```
zillow_census.head()
```

Out[41]:

	RegionID	SizeRank	Zipcode	State	City	Metro	CountyName	date_zestimate	zestimate	year	month	state	median_income	population
0	61639	0	10025	NY	New York	New York-Newark-Jersey City	New York County	1996-01-31	NaN	1996	1	NaN	NaN	NaN
1	84654	1	60657	IL	Chicago	Chicago-Naperville-Elgin	Cook County	1996-01-31	364892.0	1996	1	NaN	NaN	NaN
2	61637	2	10023	NY	New York	New York-Newark-Jersey City	New York County	1996-01-31	NaN	1996	1	NaN	NaN	NaN
3	91982	3	77494	TX	Katy	Houston-The Woodlands-Sugar Land	Harris County	1996-01-31	200475.0	1996	1	NaN	NaN	NaN
4	84616	4	60614	IL	Chicago	Chicago-Naperville-Elgin	Cook County	1996-01-31	546663.0	1996	1	NaN	NaN	NaN

This is telling me that 2% of the zipcodes can't be mapped to the zillow dataset.

In addition I only have 2011 - 2020 worth of estimates

We are going to drop the 2% later in the model building.

An example of a Zip Code 85203

In [48]:
```
zillow_clean['zestimate_growth'] = (zillow_clean["zestimate"] / zillow_clean.groupby(['Zipcode'])['zestimate'].shift
zillow_clean['zestimate_annualized'] = (1 + zillow_clean.zestimate_growth)**12 - 1
zillow_clean[zillow_clean["Zipcode"]=="85203"].tail()
```

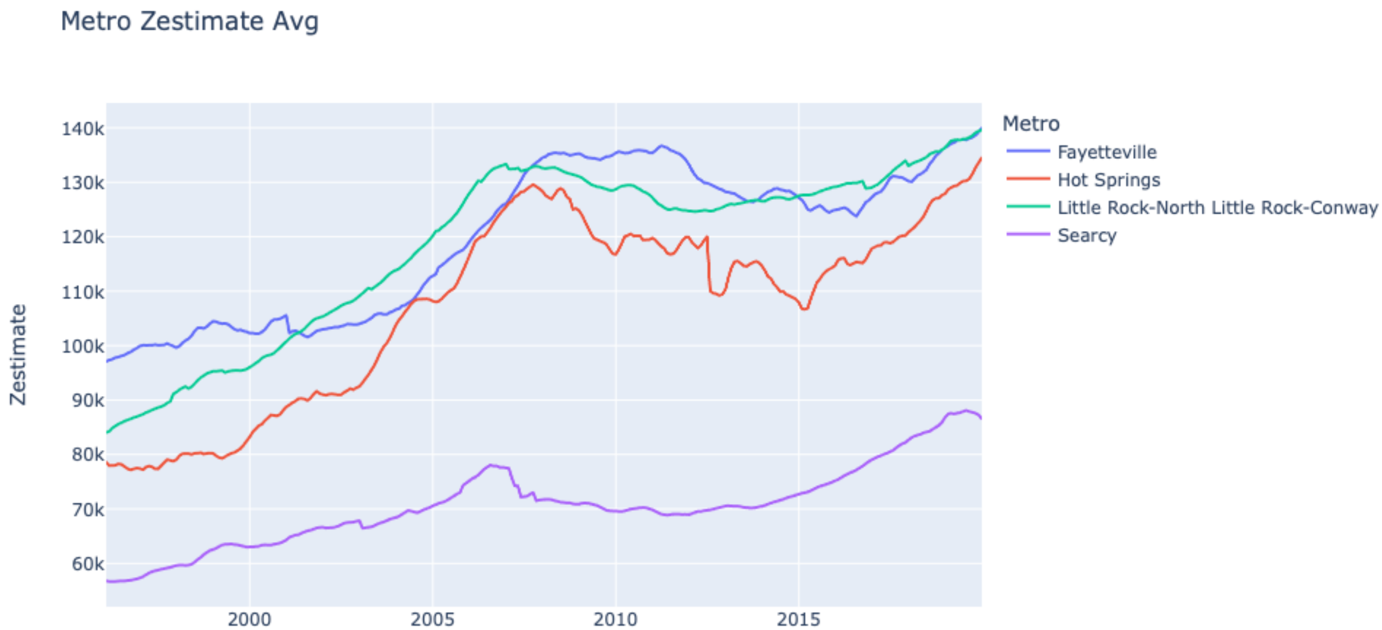
Out[48]:

	RegionID	SizeRank	Zipcode	State	City	Metro	CountyName	date_zestimate	zestimate	year	month	state	median_income	population	ze
8624435	94798	3150	85203	AZ	Mesa	Phoenix-Mesa-Scottsdale	Maricopa County	2019-08-31	274241.0	2019	8	4.0	54919.0	39797.0	
8654899	94798	3150	85203	AZ	Mesa	Phoenix-Mesa-Scottsdale	Maricopa County	2019-09-30	276154.0	2019	9	4.0	54919.0	39797.0	
8685363	94798	3150	85203	AZ	Mesa	Phoenix-Mesa-Scottsdale	Maricopa County	2019-10-31	278431.0	2019	10	4.0	54919.0	39797.0	
8715827	94798	3150	85203	AZ	Mesa	Phoenix-Mesa-Scottsdale	Maricopa County	2019-11-30	280874.0	2019	11	4.0	54919.0	39797.0	
8746291	94798	3150	85203	AZ	Mesa	Phoenix-Mesa-Scottsdale	Maricopa County	2019-12-31	283464.0	2019	12	4.0	54919.0	39797.0	

In [49]:
```
zillow_census = None
```

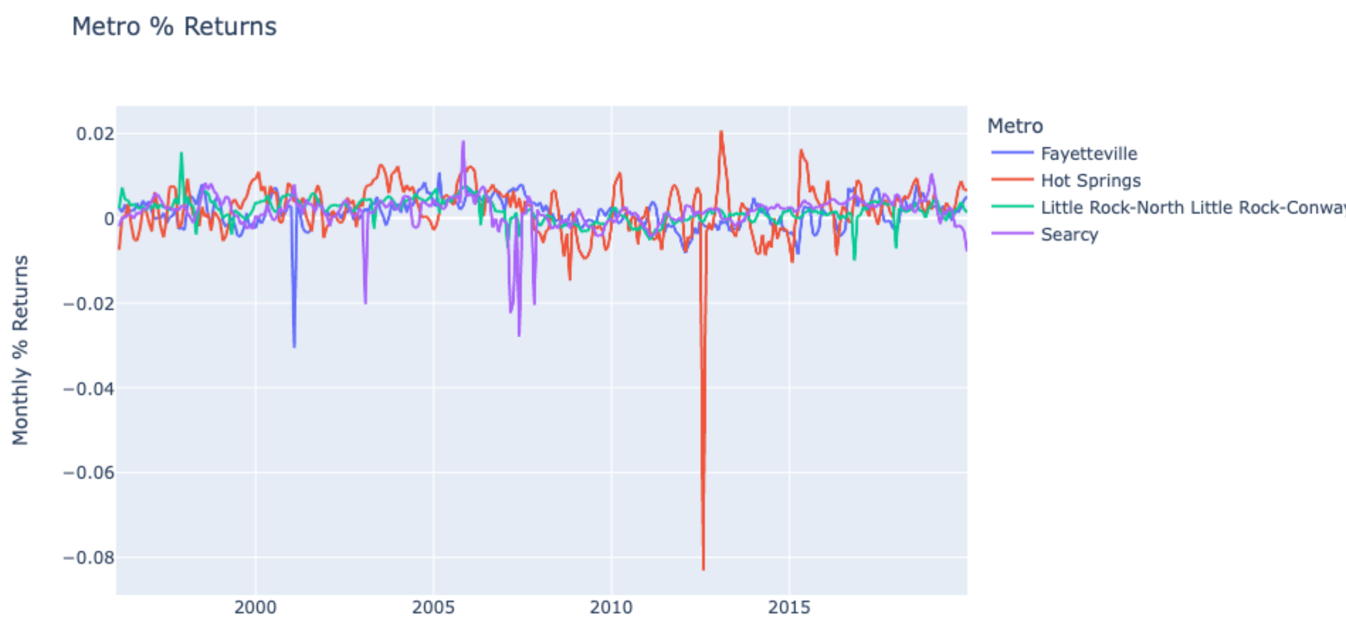
Explore

Arkansas Metro Time Series Plot



The graph above shows the 4 Metro areas Average Zestimate. In terms of growth Hot Springs appears to be the best with Little Rock being second best. This is hard to tell looking pearly out this graph. Also there is a lot of risk or volatility in all the Metro's besides Searcy. Using some finance techniques lets look at the overall Return, Risk, and Return over Risk also known as Sharpe Ratio.

Arkansas Metro Percentage Return (1997 – 2019)



This graph is a good visualization in volatility. Hot springs from 1996 until current has more volatility than the group with Searcy having the second most. Confirming our suspicions. Although Searcy appears to have done better after 2008.

Overall, from 1996 until the end of 2019 Overall Returns were as followed:

```
0.44 Return - Fayettevilee
0.71 Return - Hot Springs
0.66 Return - Little Rock
0.53 Return - Searcy
118.75 Sharpe Ratio - Fayettevilee
96.64 Sharpe Ratio - Hot Springs
238.46 Sharpe Ratio - Little Rock
125.32 Sharpe Ratio - Searcy
```

The Higher the Sharpe Ratio is better. Meaning that for every Return an investor received they took on smaller risk compared to other investments. Hot Springs return overall was 71% but an investor had to take on seen in the below chart compared to Little Rock where very risk was needed.

```
0.0037 Fayettevilee
0.0074 Hot Springs
0.0028 Little Rock
0.0042 Searcy
```

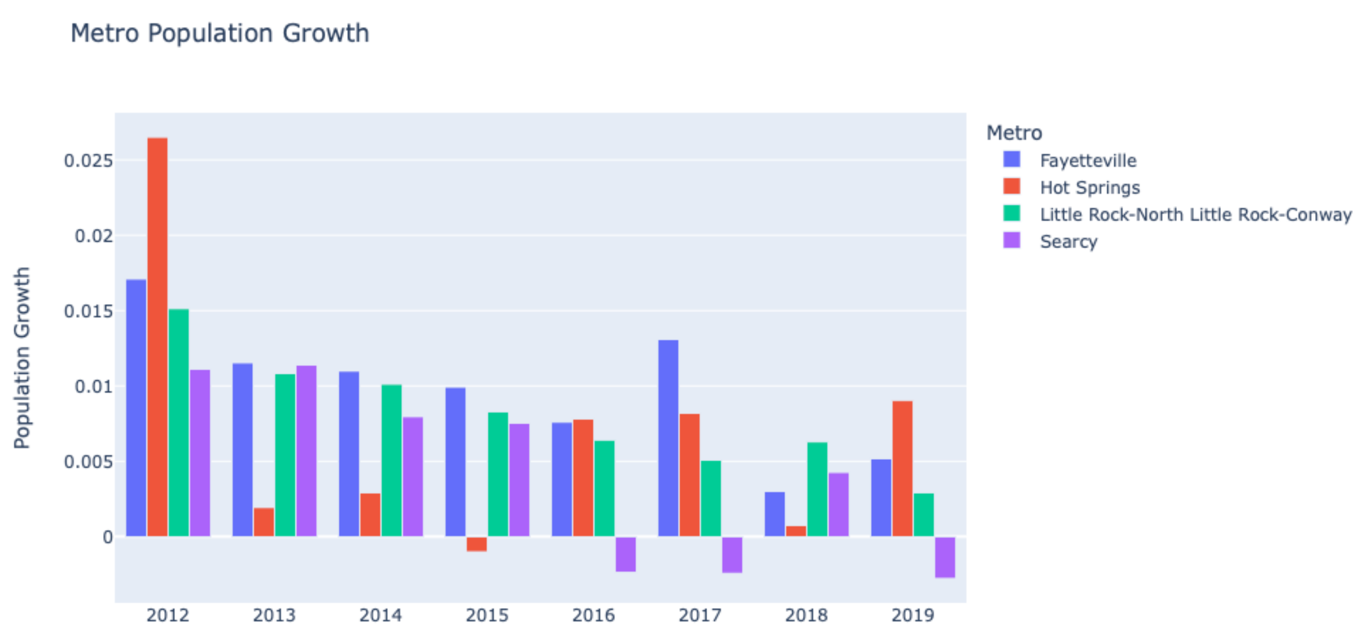
Arkansas Metro Results (2010 – 2019)

```
0.0033 std – Fayettevilee
0.0095 std – Hot Springs
0.0022 std – Little Rock
0.0025 std – Searcy
0.03 Return – Fayettevilee
0.14 Return – Hot Springs
0.08 Return – Little Rock
0.25 Return – Searcy
9.53 Sharpe Ratio – Fayettevilee
15.03 Sharpe Ratio – Hot Springs
37.04 Sharpe Ratio – Little Rock
97.68 Sharpe Ratio – Searcy
```

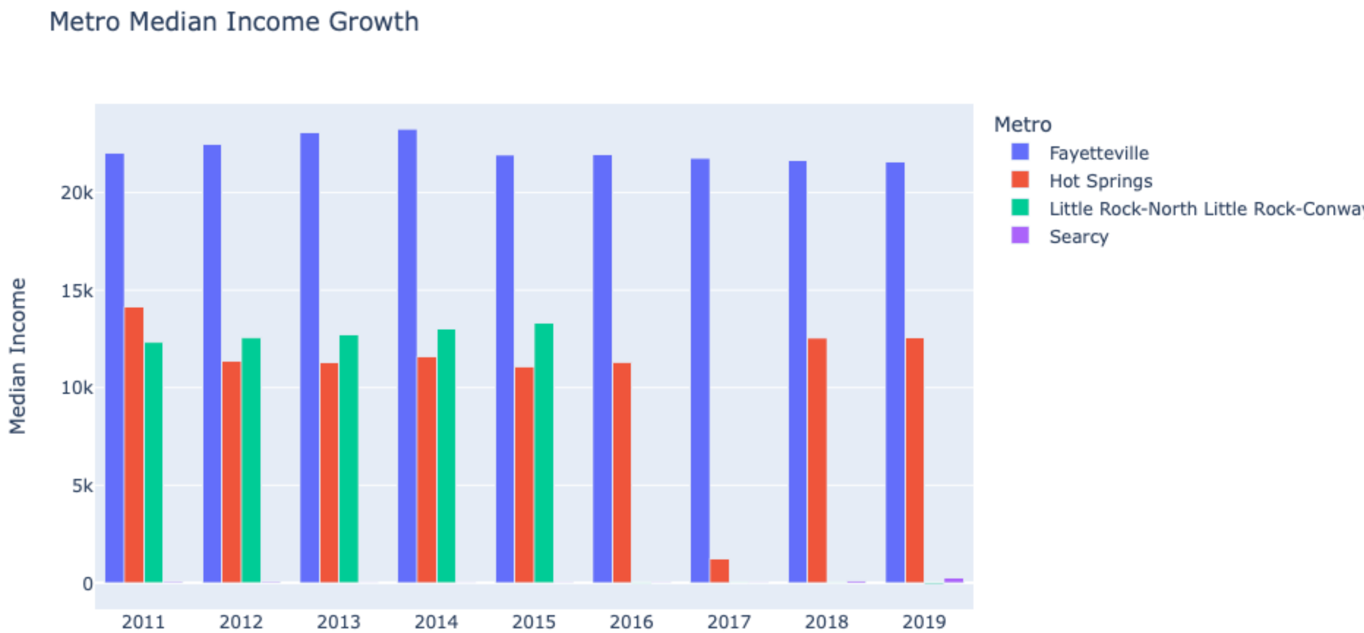
Before Hot Spring overall was the better choice from a historical perspective with the Return and Risk balanced very well. Looking to just this past decade Searcy has better return and a much higher Sharpe Ratio. The Riskiest area is actually Hot Springs now with Little over much smaller risk. Looking at more recent data will be important in the modeling stage of the analysis.

Looking at 2010 until 2019 the results are different. For the remaining of the analysis.

Arkansas Metro Population and Household Median Income (2010 – 2019)

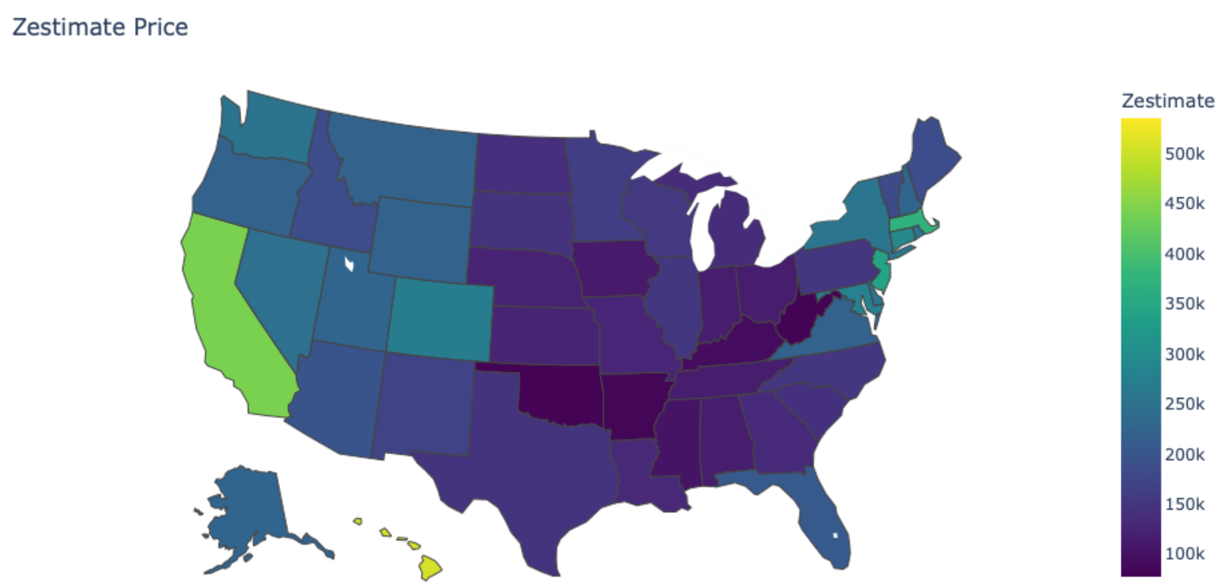


The population represented hear is just the zip codes that Zillow provided. This might not be a full representation of the population growth. There appears to be a disconnect with Searcy and Zillow data. Meaning that housing prices are going up, yet population increased the first 4 years and then is on the decline. Either this population is now becoming homeowners or houses are disappearing from the market. Census data is not always accurate and with the 2020 census data coming next year Searcy might see an adjusted population growth.

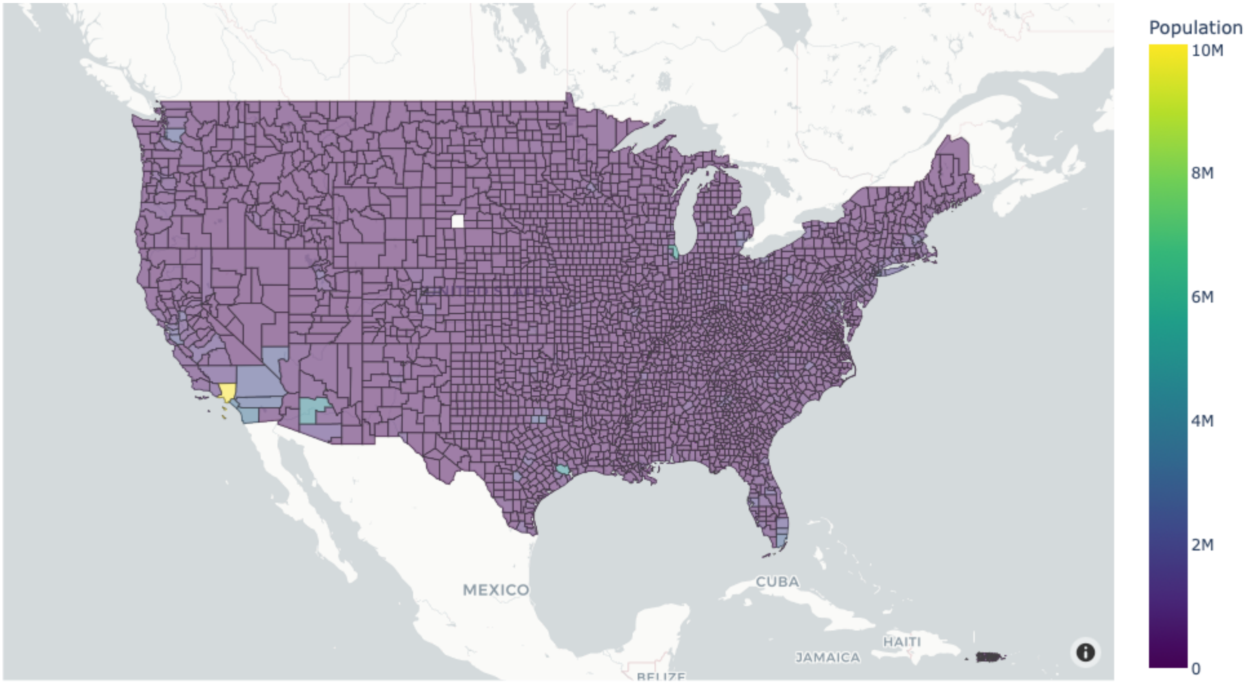


Confirming the same data issue with Searcy shows that household median income isn't increases although housing prices and population are. A big disconnect. The other graphs track with steady growth.

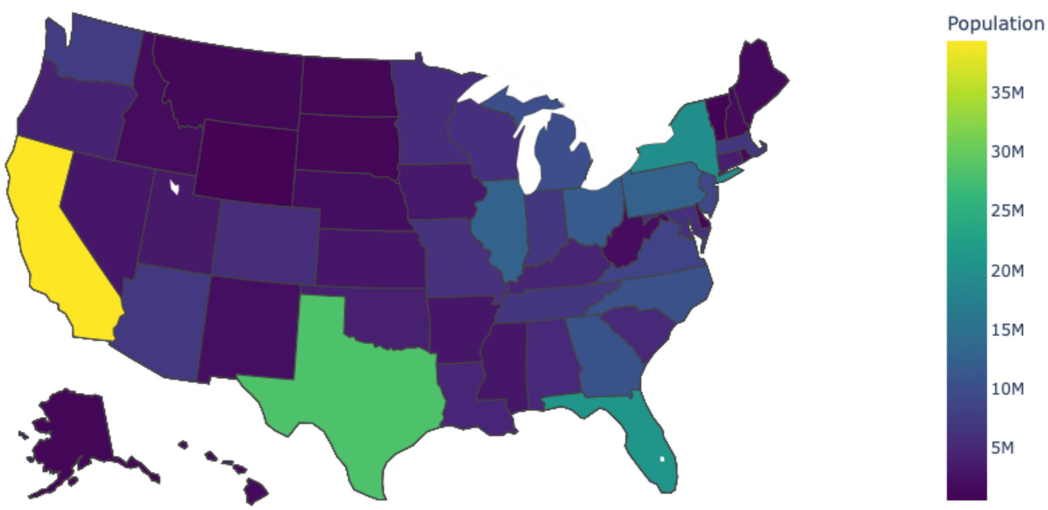
Bonus Geographic Visualization –Median Housing Price



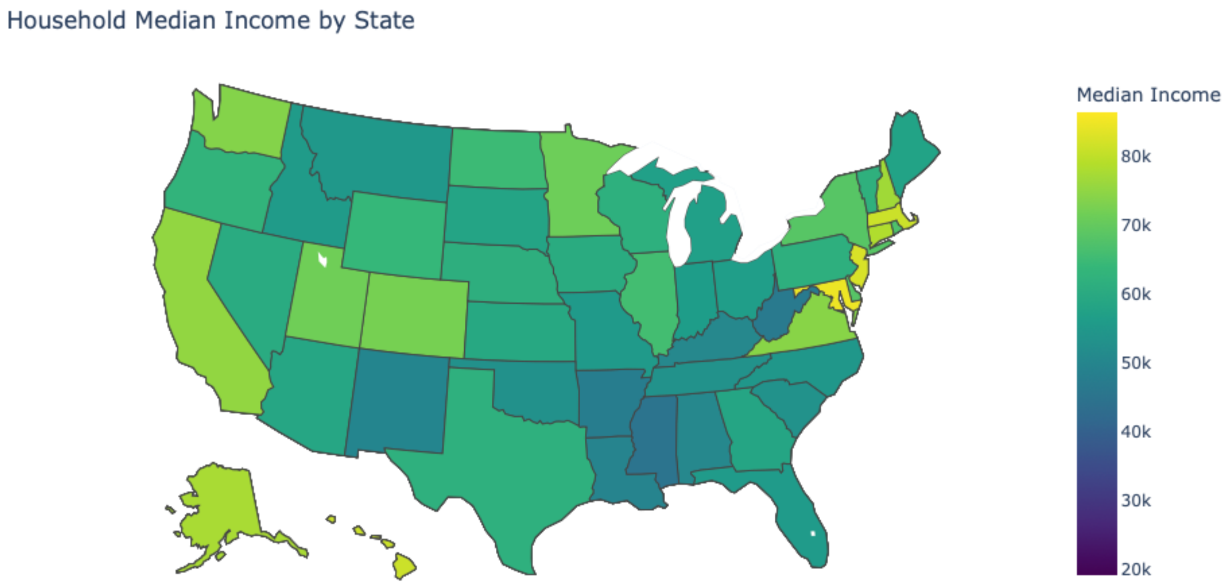
Bonus Geographic Visualization – Population



Population by State



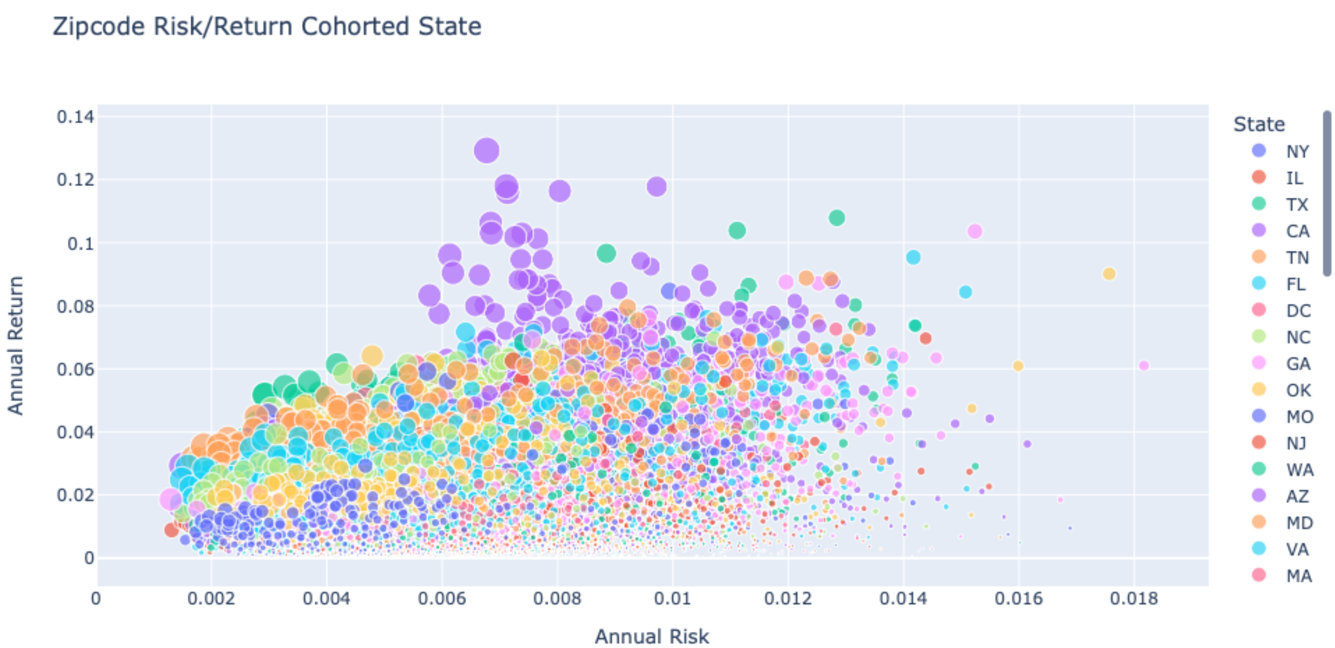
Bonus Geographic Visualization - Household Median Income



Modeling

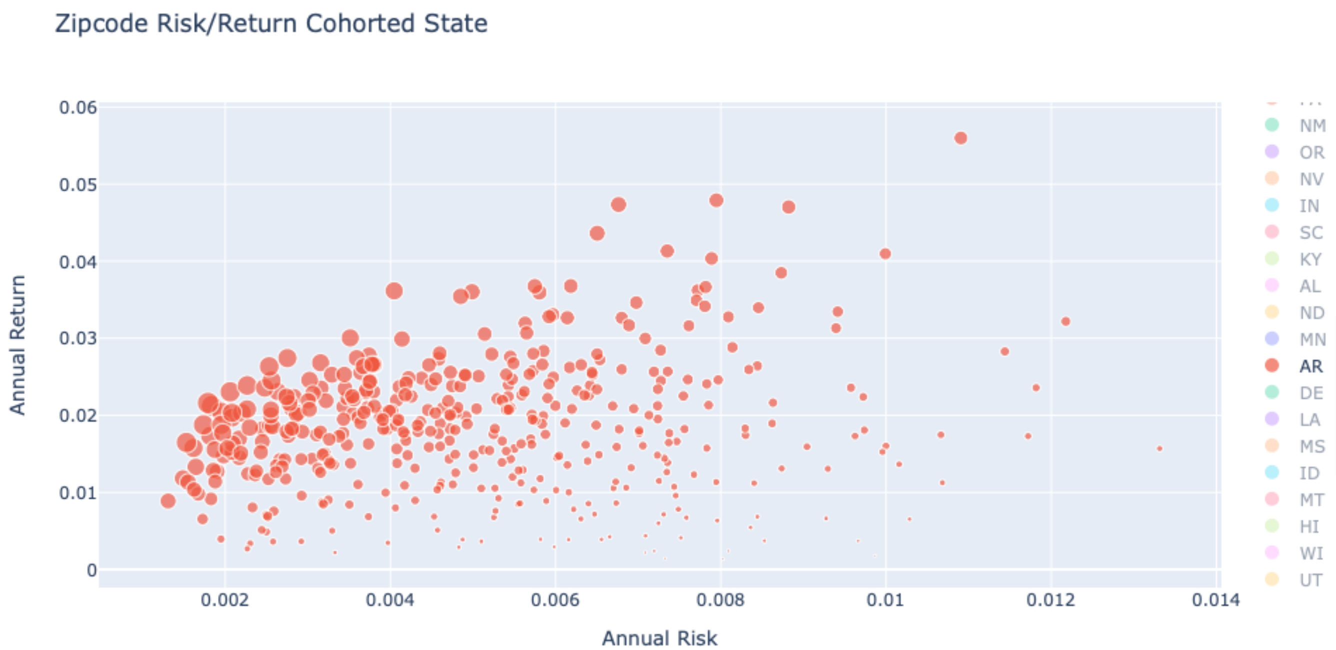
Historic Risk and Return - USA

When looking to perform the modeling historic returns should be considered by Zip code. The chart below shows all Zip Codes color coded by State. The objective in the analysis is too have the most return with lower risk. Meaning If a zip code achieves 12% return and a Standard Deviation (Risk) of 2% and another zip code achieves 12% return with a Risk percentage of 1% then taking the ladder zip code is the most optimal solution. This can also be described as a Sharpe Ratio where return is divided by risk. The higher the Sharpe Ratio means a more balance Return over Risk solution.



This graph shows every state from 2010 – 2019. The size is represented the Sharpe Ratio. As can be seen that California has a higher return with lower risk than many of the zip codes across the nation. Let's focus though on Arkansas.

Historic Risk and Return - Arkansas

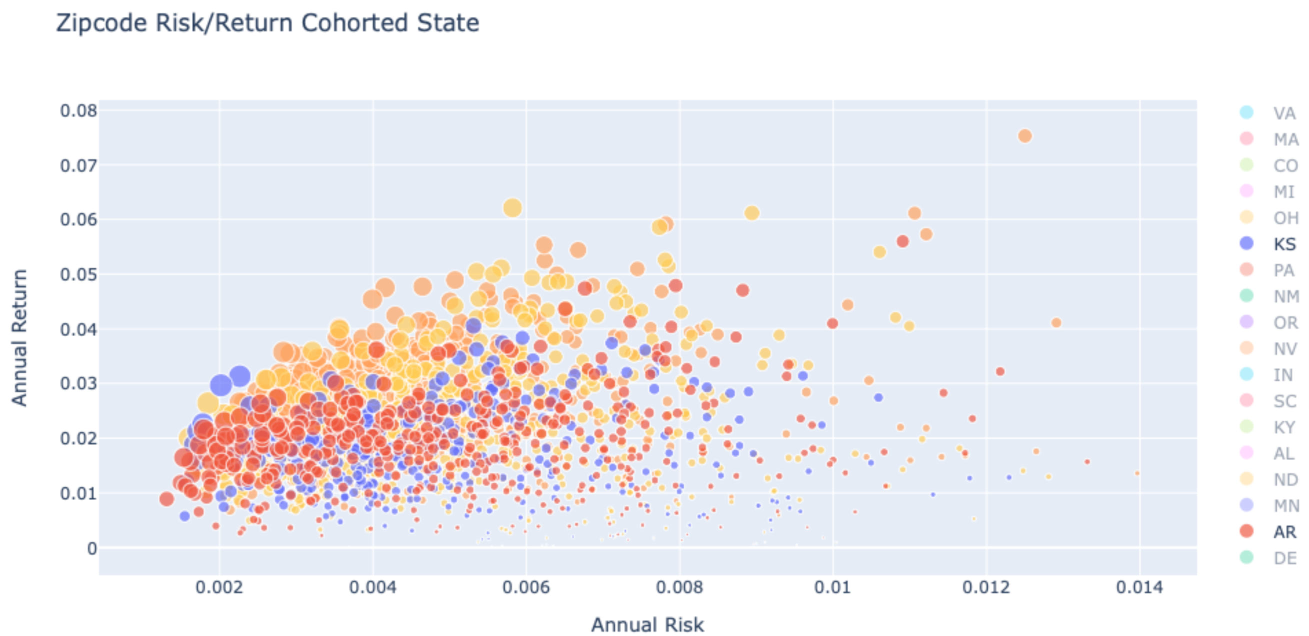


Historically Speaking zip code 72447 has the highest return at 5.5% with some of the highest risk. Depending on the risk tolerance of an investor will depend if investing in this zip makes sense.

The next 3 highest zip codes have similar returns at 5% but have different risk levels:

- 71740
- 72675
- 72645

The better of the 3 and of even the highest would be 72645 achieving 4.7% return with much lower risk than the highest.



KS = Blue
AR = Red
OK = Yellow
TN = Orange

When looking at Neighboring states, excluding Texas There is better returns over risk then Arkansas. Meaning any different color bubble than red and is above red means that there is more return for the same amount risk.

Forecasting Arkansas Return - Scrubing

When forecasting for Arkansas in 2020 and finding the 3 highest Zip codes Facebook Prophet was used in modeling these results. Before running the prophet additional scrubbing had to be done:

1. Filtered for just Zestimates from 2010 – 2019.
2. Only looked at Zipcodes with historic returns greater than 3% over the past 4 years.
3. Why looking over 3% is because inflation on average is 2% and investors minimum need to be compensated for taking on housing risk by a factor of 1%.
4. Then looked at just Arkansas which comes out to be 420 data points.
5. Changed names so that the Prophet would ingest the data.
6. Date_zestimate changed to ds and zestimate to y
7. Looped the data through prophet by zipcode and appened to an empty dataframe including the zip code as a column.
8. Found the annualized returns predicted, annual risk predicted, and the sharpe ratio predicted.

9. Please see the below code for these steps.

Looking at just a state

In [90]:
```
prophet_input =  zillow_clean[["Zipcode","date_zestimate","zestimate"]]
prophet_input = prophet_input[(prophet_input["date_zestimate"]>="2010-01-31") &
                              prophet_input["Zipcode"].isin(fronteir_filter) &
                              prophet_input["Zipcode"].isin(state_filter)
                              ]

prophet_input = prophet_input.rename(columns={'date_zestimate': 'ds',
                                              'zestimate' : 'y'})

len(prophet_input["Zipcode"].unique())
```

Out[90]: 420

In [91]:
```
prophet_input.head()
```

Out[91]:

	Zipcode	ds	y
5118283	71913	2010-01-31	137404.0
5118615	72034	2010-01-31	156265.0
5118820	72701	2010-01-31	163882.0
5118865	72764	2010-01-31	126111.0
5119182	72401	2010-01-31	104320.0

In [98]:
```
zipcode_list = prophet_input["Zipcode"].unique()
prophet_state_staging = pd.DataFrame()

for z in zipcode_list:
#for z in zipcode_list[:5]:

  prophet_zip = prophet_input[prophet_input["Zipcode"]==z]
  prophet_zip = prophet_zip[["ds","y"]].reset_index()
  forecast = run_prophet(prophet_zip)
  forecast["Zipcode"] = z
  prophet_state_staging = prophet_state_staging.append(forecast)
```

```
INFO:fbprophet:n_changepoints greater than number of observations. Using 24.
INFO:fbprophet:n_changepoints greater than number of observations. Using 23.
INFO:fbprophet:n_changepoints greater than number of observations. Using 23.
INFO:fbprophet:n_changepoints greater than number of observations. Using 20.
INFO:fbprophet:n_changepoints greater than number of observations. Using 20.
INFO:fbprophet:n_changepoints greater than number of observations. Using 18.
INFO:fbprophet:n_changepoints greater than number of observations. Using 17.
```

In [ ]:
```
prophet_state = prophet_state_staging
```

	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	multiplicat
0	2020-01-01	281164.549597	280052.789384	282239.433515	281164.549597	281164.549597	0.0	0.0	0.0	
1	2020-02-01	282541.223462	281315.715841	283606.205996	282520.580022	282541.223462	0.0	0.0	0.0	
2	2020-03-01	283829.079659	282497.956149	285154.693978	283460.668011	284050.937447	0.0	0.0	0.0	
3	2020-04-01	285205.753525	283548.256462	286654.651844	284301.996779	285861.273787	0.0	0.0	0.0	
4	2020-05-01	286538.018556	284488.383967	288235.842715	284887.231278	287765.569688	0.0	0.0	0.0	
5	2020-06-01	287914.692421	285240.945100	289953.373526	285407.169436	289754.472816	0.0	0.0	0.0	
6	2020-07-01	289246.957452	285882.474423	291757.021184	285978.341023	291879.720117	0.0	0.0	0.0	
7	2020-08-01	290623.631318	286317.166392	294263.472236	286546.221033	294111.783835	0.0	0.0	0.0	
8	2020-09-01	292000.305183	286760.059617	296426.644971	286969.188651	296321.188914	0.0	0.0	0.0	
9	2020-10-01	293332.570214	287071.378488	298707.073353	287293.743728	298770.585618	0.0	0.0	0.0	
10	2020-11-01	294709.244080	287527.296198	301228.205888	287516.505038	301213.037647	0.0	0.0	0.0	
11	2020-12-01	296041.509111	287745.459285	303752.382758	287864.496141	303595.335319	0.0	0.0	0.0	

In [108]:
```
prophet_state.head()
```

Out[108]:

	Zipcode	annualized_predicted	annual_risk_predicted	sharpe_ratio_predicted	State	City	Metro	CountyName
0	00727	0.045572	0.001158	39.353874	AR	Walnut Ridge	NaN	Lawrence County
1	00907	0.100686	0.003528	28.536166	AR	Widener	Forrest City	Saint Francis County
2	05030	0.005875	0.000133	44.296648	AR	Hoxie	NaN	Lawrence County
3	71601	0.039274	0.000969	40.547210	AR	Pine Bluff	Pine Bluff	Jefferson County
4	71602	0.036127	0.000879	41.107490	AR	White Hall	Pine Bluff	Jefferson County

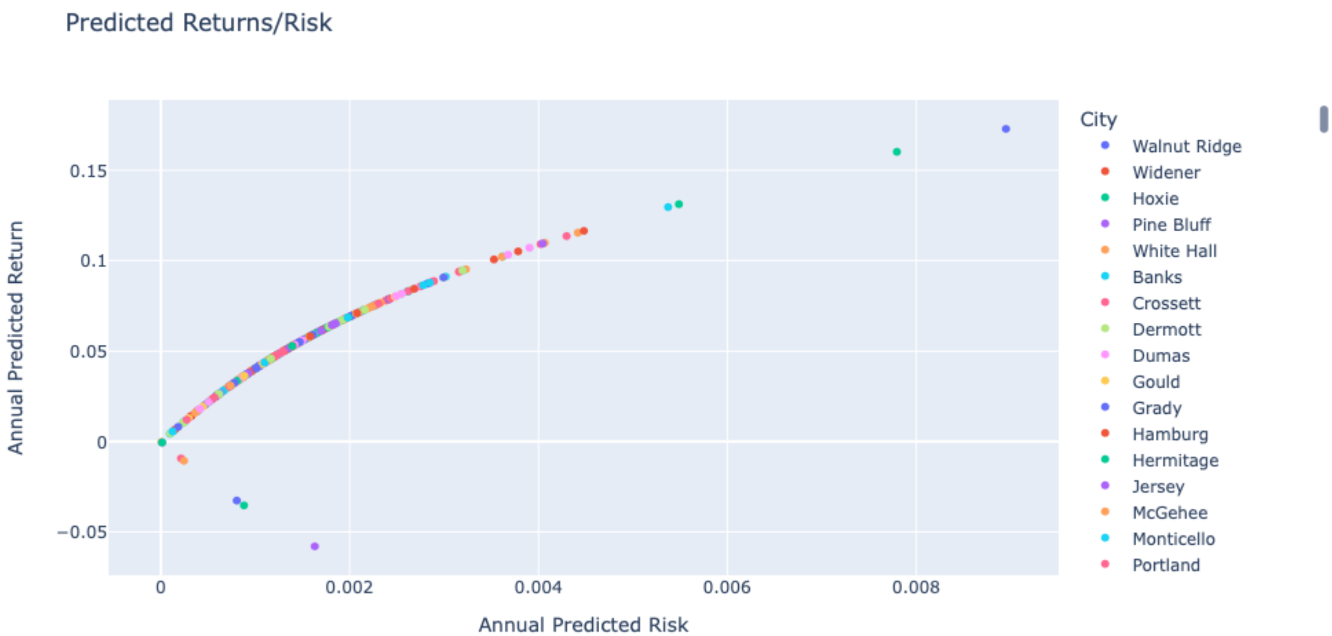
Forecasting Arkansas Return - Results

The below graphs shows that the 3 highest return for 2020:

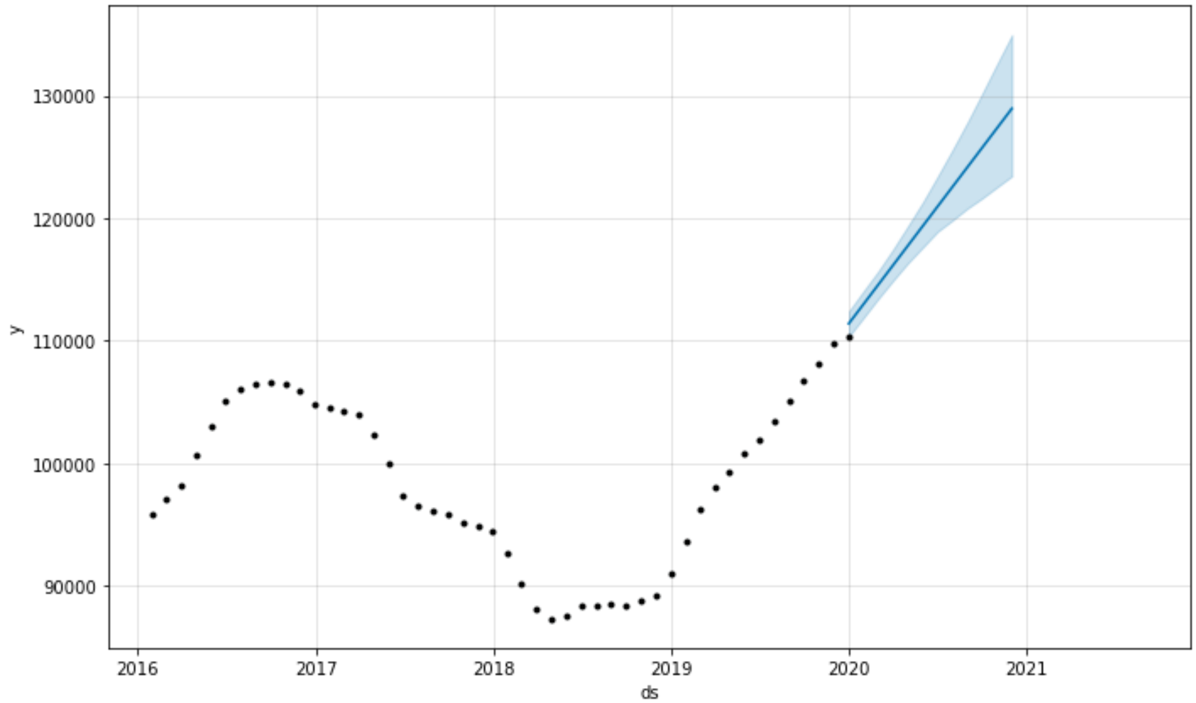
1. 72630 – Diamond City with a return of 17% and risk of 0.9%
2. 72431 – Grubbs with a return of 16% and risk of 0.8%
3. 71935 – Caddo Gap with a return of 13% and risk of 0.5%

Of the 3 that are the highest in graph 3 – 5 the predictive power illustrates the error bands in this forecast. Zip code 71935 appears to have much better predictive power of the 3 where the other's do not.

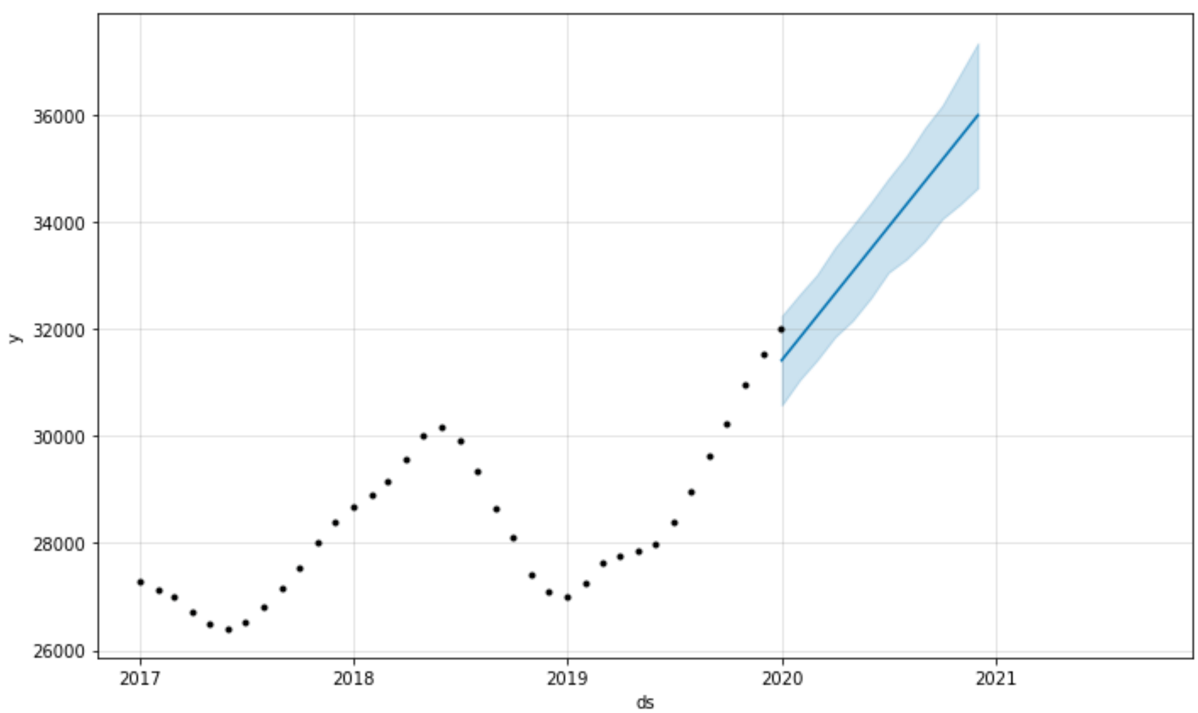
	Zipcode	annualized_predicted	annual_risk_predicted	sharpe_ratio_predicted	State	City	Metro	CountyName
0	71935	0.131250	0.005487	23.922168	AR	Caddo Gap	NaN	Montgomery County
1	72431	0.160193	0.007795	20.550340	AR	Grubbs	NaN	Jackson County
2	72630	0.172900	0.008950	19.317935	AR	Diamond City	Harrison	Boone County



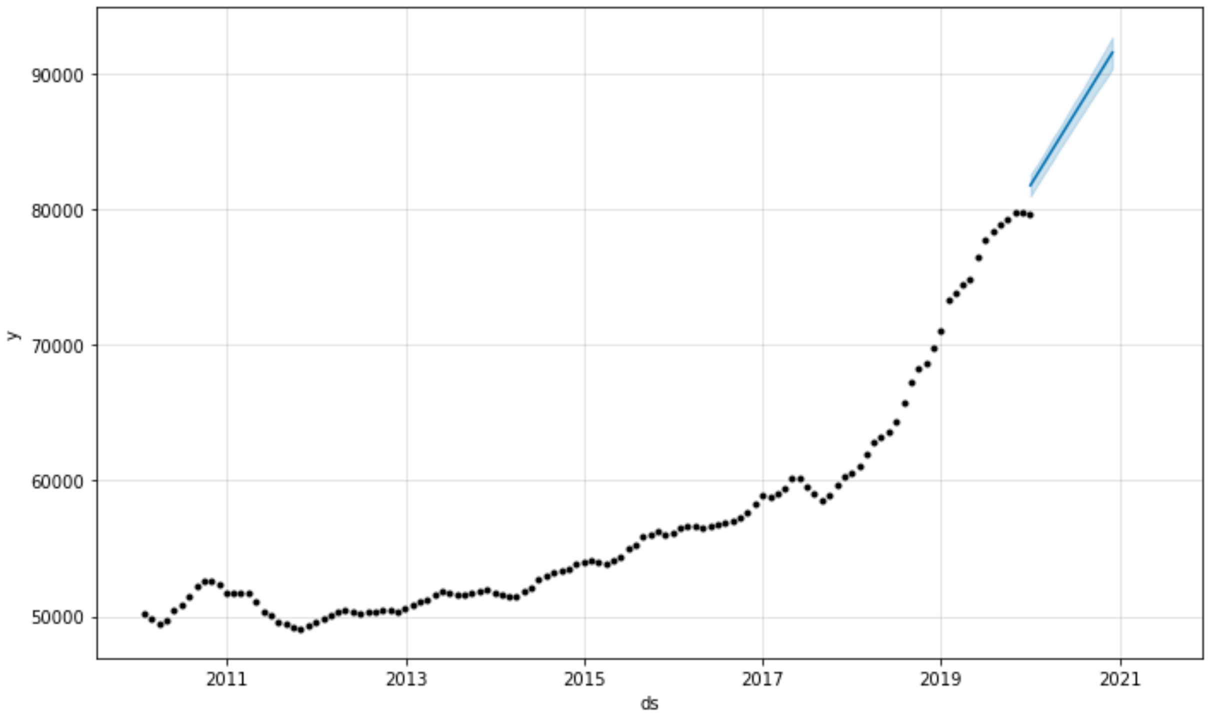
Zip code: 72630



Zip code: 72431



Zip code: 71935



	Zipcode	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	yhat
11	71935	2020-12-01	91567.496325	90426.706941	92689.232897	90690.328156	92372.641497	91567.496325
11	72630	2020-12-01	128944.672435	123551.605313	134661.304506	123813.694430	134569.823105	128944.672435
11	72431	2020-12-01	36003.709169	34743.758524	37330.122973	35024.207577	37010.344439	36003.709169