In [1]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:

```
import numpy as np, pandas as pd, matplotlib.pyplot as plt, seaborn
from datetime import datetime, timedelta
from fredapi import Fred
import quandl
```

In [3]:

```
def get_info(names):
    data = []
    for i in range(len(names)):
        data.append(fred.get_series(names[i]).to_frame().rename(col
        data[i] = data[i].groupby(data[i].index.year).mean().dropna
    return data
```

In [4]:

```
# https://github.com/mortada/fredapi
fred = Fred(api_key="a02df0a22c57860f5f7cf25edc70ffb3")
quandl.ApiConfig.api_key = "QZLZXdHDDPZna9Yw48NP"
```

Midwest - Ohio

```
Define the variables to be used in analysis:

X attributes:

 - *Monthly* Stocks
     - S&P 500 (MULTPL/SP500_REAL_PRICE_MONTH)
 - *Quarterly* GDP (GDP)
 - *Annual* Unemployment Rate (LAUST390000000000003A)
 - *Annual* House Ownership Ratio (OHHOWN)
 - *Annual* Resident Population (OHPOP)
 - *Annual* Median Income Rate (MEHOINUSOHA672N)
 - *Annual* Home Vacancy Rate (OHHVAC)

y attributes:

 - *Quarterly* Ohio State Housing Price Index (OHSTHPI)
```

Connect to APIs and create a dataframe with information on each dataset:

In [5]:

```
sp500 = quandl.get('MULTPL/SP500_REAL_PRICE_MONTH').rename(columns=
sp500 = sp500.groupby(sp500.index.year).mean().dropna()
names_oh = ['LAUST390000000000003A', "OHHOWN","OHPOP", "MEHOINUSOHA
oh_data_series = get_info(names_oh) + [sp500]
```

In [6]:

```
# quarterly housing price index
ohHPI = fred.get_series('OHSTHPI').to_frame()
ohHPI.index.name = "DATE"
ohHPI = ohHPI.rename(columns={0:"OHSTHPI"})
# convert to annual
ohHPI_annual = ohHPI.groupby(ohHPI.index.year).mean()
```

In [7]:

```
oh_annual = ohHPI_annual.copy()
for df in oh_data_series:
    oh_annual = oh_annual.merge(df, left_index=True, right_index=Tr
oh_annual.tail()
```

Out[7]:

	OHSTHPI	LAUST390000000000003A	OHHOWN	OHPOP	MEHOINU
2014	241.2700	5.8	67.3	11602.973	
2015	250.8625	4.9	66.4	11617.850	
2016	261.2600	5.0	66.1	11635.003	
2017	274.8250	5.0	66.0	11664.129	
2018	291.9225	4.6	67.3	11689.442	

In [ ]:

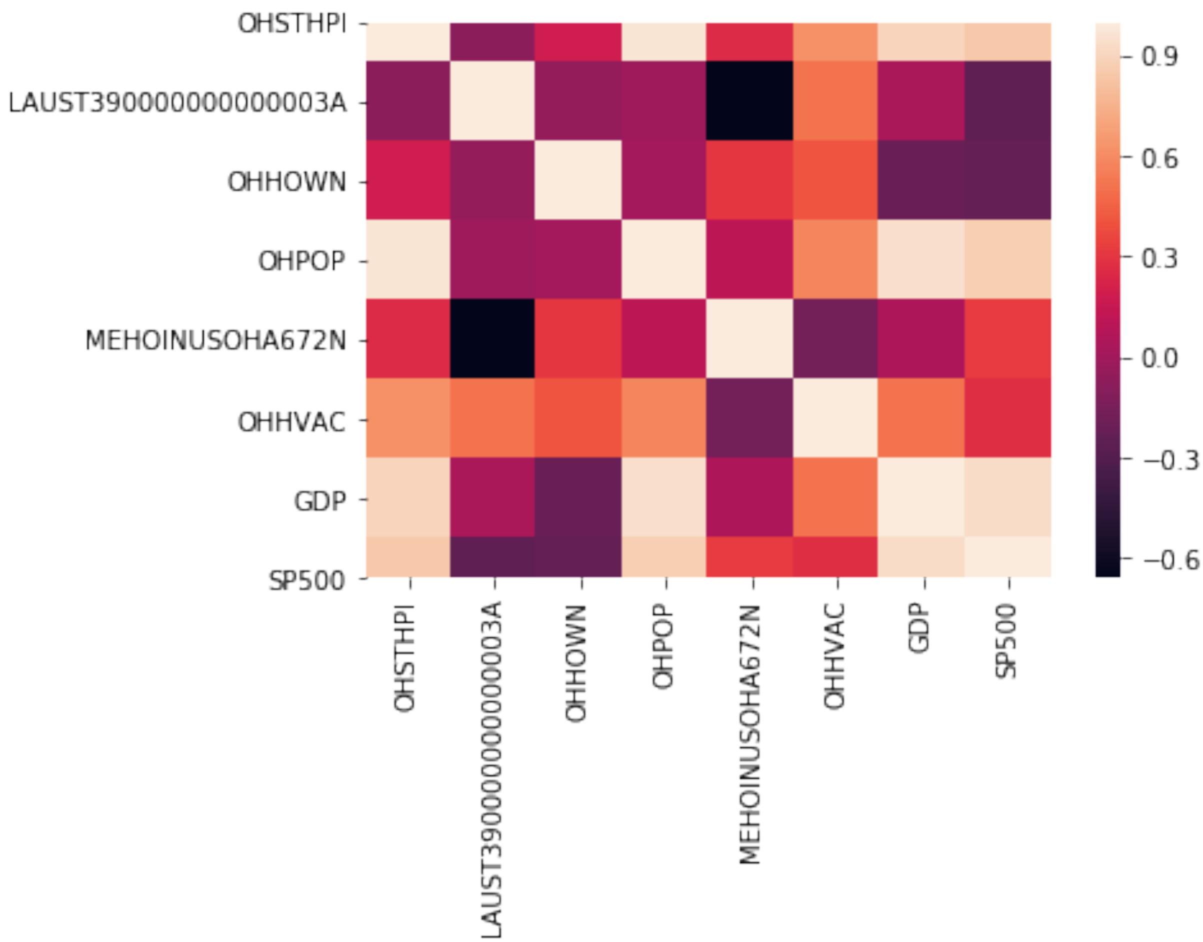
```
Analyze the correlation coefficient for each indicator we have spec
```

In [8]:

```
corr = oh_annual.corr().round(4)
sns.heatmap(data=corr)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12b786978>
```



In [9]:

```
corr
```

Out[9]:

	OHSTHPI	LAUST390000000000003A	OHHOWN	C
OHSTHPI	1.0000	-0.0742	0.1842	
LAUST390000000000003A	-0.0742	1.0000	-0.0379	-
OHHOWN	0.1842	-0.0379	1.0000	
OHPOP	0.9685	-0.0087	0.0100	
MEHOINUSOHA672N	0.2553	-0.6591	0.3062	
OHHVAC	0.6238	0.5090	0.4048	
GDP	0.9003	0.0432	-0.2085	
SP500	0.8491	-0.2486	-0.2276	

```
Create a model using linear regression to express the Housing
Price Index as dependent on the other datasets we have downloaded:
```

In [10]:

```
X = oh_annual.drop(columns=['OHSTHPI'], axis=1)
Y = oh_annual['OHSTHPI']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size
lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)
```

Out[10]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jo
bs=None, normalize=False)
```

In [11]:

```
# model evaluation for training set
y_train_predict = lin_model.predict(X_train)
rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
r2 = r2_score(Y_train, y_train_predict)

print("The model performance for training set")
print("--------------------------------------")
print('Root Mean Squared Error is {}'.format(rmse))
print('R-Squared score is {}'.format(r2))
print("\n")

# model evaluation for testing set
y_test_predict = lin_model.predict(X_test)
rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
r2 = r2_score(Y_test, y_test_predict)

print("The model performance for testing set")
print("--------------------------------------")
print('Root Mean Squared Error is {}'.format(rmse))
print('R-Squared score is {}'.format(r2))
```

```
The model performance for training set
--------------------------------------
Root Mean Squared Error is 5.797686770566625
R-Squared score is 0.9867195090934263


The model performance for testing set
--------------------------------------
Root Mean Squared Error is 5.249607745741014
R-Squared score is 0.987106155235977З
```

In [1]:

In [2]:

```

```

In [3]:

```

```

In [4]:

```

```

Midwest - Ohio

Define the variables to be used in analysis:

X attributes:

- *Monthly* Stocks
  - S&P 500 (MULTPL/SP500_REAL_PRICE_MONTH)
- *Quarterly* GDP (GDP)
- *Annual* Unemployment Rate (LAUST390000000000003A)
- *Annual* House Ownership Ratio (OHHOWN)
- *Annual* Resident Population (OHPOP)
- *Annual* Median Income Rate (MEHOINUSOHA672N)
- *Annual* Home Vacancy Rate (OHHVAC)

y attributes:

- *Quarterly* Ohio State Housing Price Index (OHSTHPI)

Connect to APIs and create a dataframe with information on each dataset:

In [5]:

```

```

In [6]:

```

```

In [7]:

```

```

Out[7]:

	OHSTHPI	LAUST390000000000003A	OHHOWN	OHPOP	MEHOINU
2014	241.2700	5.8	67.3	11602.973	
2015	250.8625	4.9	66.4	11617.850	
2016	261.2600	5.0	66.1	11635.003	
2017	274.8250	5.0	66.0	11664.129	
2018	291.9225	4.6	67.3	11689.442	

In [ ]:

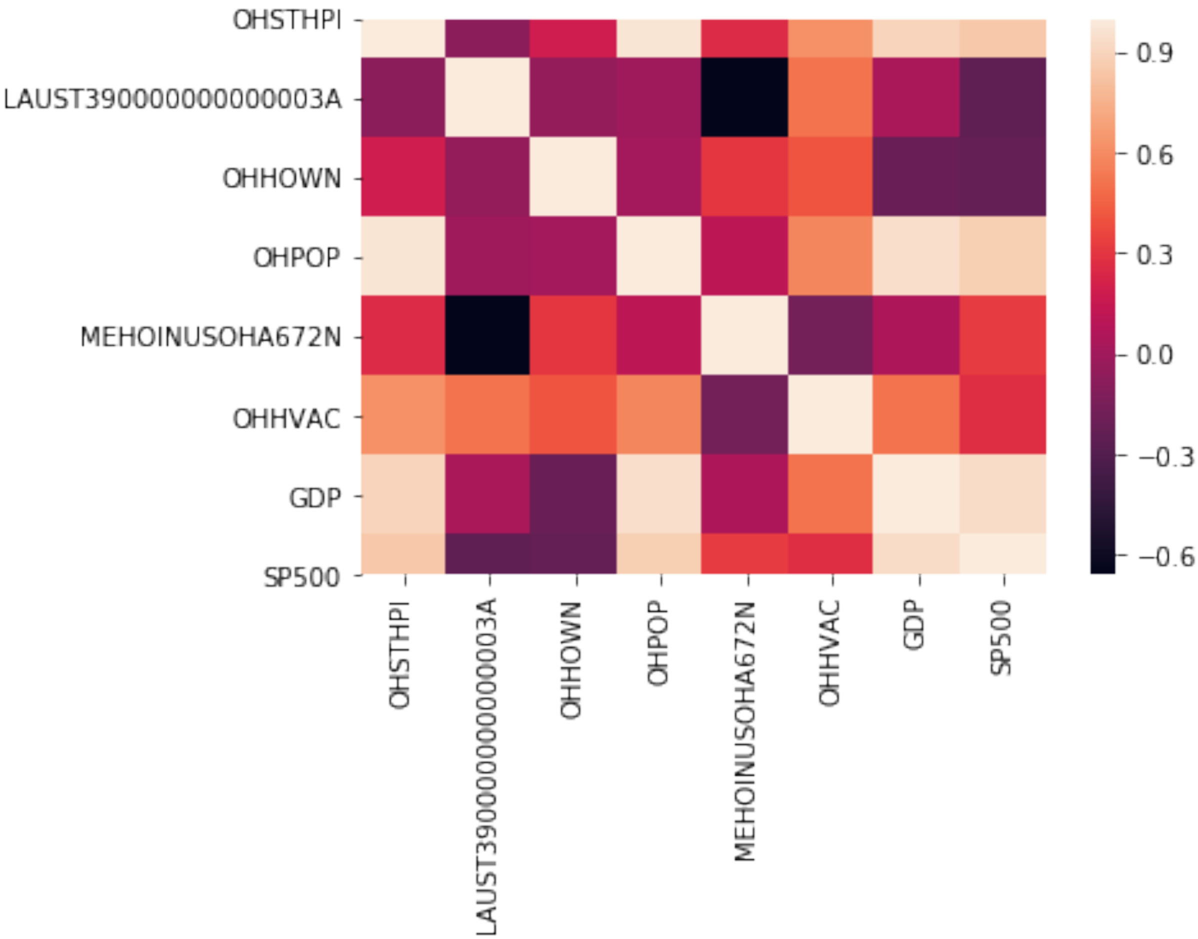
```

```

In [8]:

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x12b786978>



In [9]:

Out[9]:

	OHSTHPI	LAUST390000000000003A	OHHOWN	C
OHSTHPI	1.0000	-0.0742	0.1842	
LAUST390000000000003A	-0.0742	1.0000	-0.0379	-
OHHOWN	0.1842	-0.0379	1.0000	
OHPOP	0.9685	-0.0087	0.0100	
MEHOINUSOHA672N	0.2553	-0.6591	0.3062	
OHHVAC	0.6238	0.5090	0.4048	
GDP	0.9003	0.0432	-0.2085	
SP500	0.8491	-0.2486	-0.2276	

Create a model using linear regression to express the Housing Price Index as dependent on the other datasets we have downloaded:

In [10]:

Out[10]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jo
bs=None, normalize=False)
```

In [11]:

```
The model performance for training set
--------------------------------------
Root Mean Squared Error is 5.797686770566625
R-Squared score is 0.9867195090934263


The model performance for testing set
--------------------------------------
Root Mean Squared Error is 5.249607745741014
R-Squared score is 0.9871061552359773
```