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"
```

South - South Carolina

Define the variables to be used in analysis:

X attributes:

- *Monthly* Stocks
  - S&P 500 (MULTPL/SP500_REAL_PRICE_MONTH)
- *Quarterly* Gross Domestic Product (GDP)
- *Annual* Unemployment Rate (LAUST450000000000003A)
- *Annual* House Ownership Ratio (SCHOWN)
- *Annual* Resident Population (SCPOP)
- *Annual* Median Income Rate (MEHOINUSSCA672N)
- *Annual* Home Vacancy Rate (SCHVAC)

y attributes:

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

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

In [5]:

```
sp500 = quandl.get('MULTPL/SP500_REAL_PRICE_MONTH').rename(columns=
names_sc = ['LAUST450000000000003A', "SCHOWN","SCPOP", "MEHOINUSSCA
sp500 = sp500.groupby(sp500.index.year).mean().dropna()
sc_data_series = get_info(names_sc)  + [sp500]
```

In [6]:

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

In [7]:

```
sc_annual = scHPI_annual.copy()
for df in sc_data_series:
    sc_annual = sc_annual.merge(df, left_index=True, right_index=Tr
sc_annual.tail()
```

Out[7]:

	SCSTHPI	LAUST450000000000003A	SCHOWN	SCPOP	MEHOINUS
2014	309.3825	6.5	72.9	4823.793	
2015	325.5225	6.0	67.1	4892.253	
2016	343.3150	5.0	68.9	4958.235	
2017	363.0900	4.3	72.8	5021.219	
2018	388.1250	3.4	72.0	5084.127	

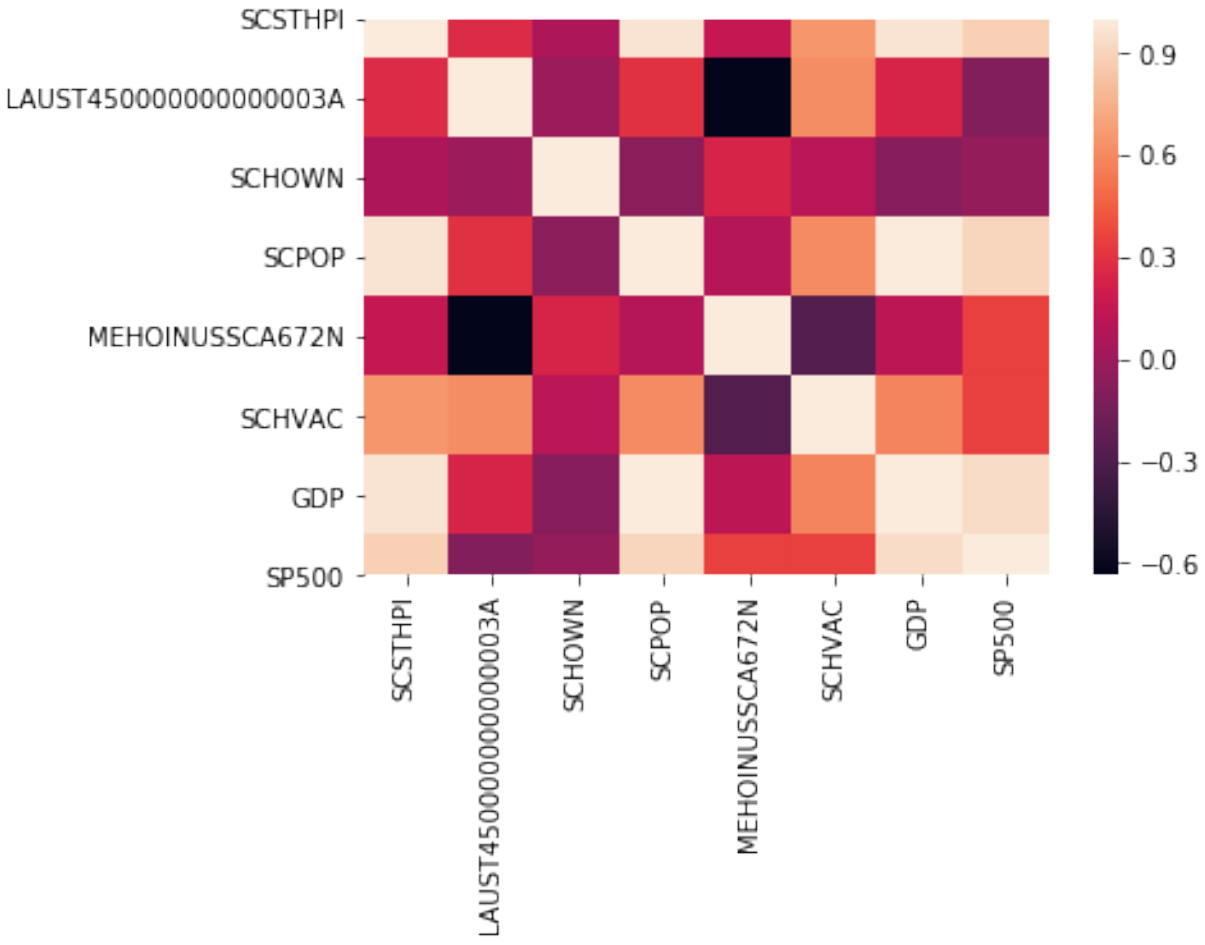
Analyze the correlation coefficient for each indicator we have specified:

In [8]:

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

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11eac4588>
```



In [9]:

```
corr
```

Out[9]:

	SCSTHPI	LAUST450000000000003A	SCHOWN	S
SCSTHPI	1.0000	0.2672	0.0640	0
LAUST450000000000003A	0.2672	1.0000	0.0004	0
SCHOWN	0.0640	0.0004	1.0000	-0
SCPOP	0.9651	0.2936	-0.0665	1
MEHOINUSSCA672N	0.1580	-0.6345	0.2389	0
SCHVAC	0.6438	0.6129	0.1073	0
GDP	0.9666	0.2355	-0.0766	0
SP500	0.8790	-0.0920	-0.0276	0

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

In [10]:

```
X = sc_annual.drop(columns=['SCSTHPI'], axis=1)
Y = sc_annual['SCSTHPI']
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 13.04729263105491
R-Squared score is 0.9701514692551751


The model performance for testing set
--------------------------------------
Root Mean Squared Error is 14.387754373625608
R-Squared score is 0.9519664612277512
```

In [1]:

In [2]:

In [3]:

In [4]:

South - South Carolina

Define the variables to be used in analysis:

X attributes:

- *Monthly* Stocks
  - S&P 500 (MULTPL/SP500_REAL_PRICE_MONTH)
- *Quarterly* Gross Domestic Product (GDP)
- *Annual* Unemployment Rate (LAUST450000000000003A)
- *Annual* House Ownership Ratio (SCHOWN)
- *Annual* Resident Population (SCPOP)
- *Annual* Median Income Rate (MEHOINUSSCA672N)
- *Annual* Home Vacancy Rate (SCHVAC)

y attributes:

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

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

In [5]:

In [6]:

In [7]:

Out[7]:

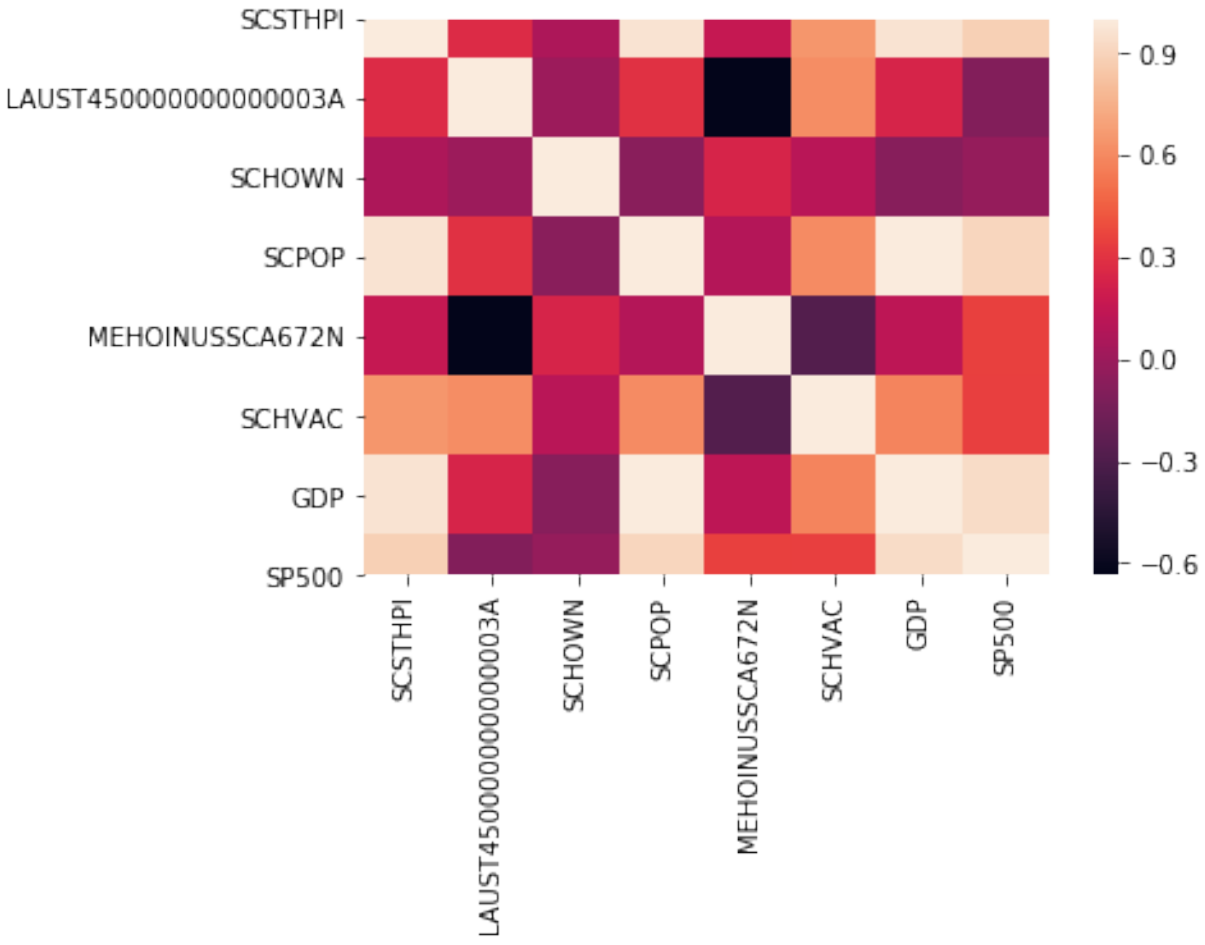
	SCSTHPI	LAUST450000000000003A	SCHOWN	SCPOP	MEHOINUS
2014	309.3825	6.5	72.9	4823.793	
2015	325.5225	6.0	67.1	4892.253	
2016	343.3150	5.0	68.9	4958.235	
2017	363.0900	4.3	72.8	5021.219	
2018	388.1250	3.4	72.0	5084.127	

Analyze the correlation coefficient for each indicator we have specified:

In [8]:

Out[8]:

<matplotlib.axes._subplots.AxesSubplot at 0x11eac4588>



In [9]:

Out[9]:

	SCSTHPI	LAUST450000000000003A	SCHOWN	S
SCSTHPI	1.0000	0.2672	0.0640	0
LAUST450000000000003A	0.2672	1.0000	0.0004	0
SCHOWN	0.0640	0.0004	1.0000	-0
SCPOP	0.9651	0.2936	-0.0665	1
MEHOINUSSCA672N	0.1580	-0.6345	0.2389	0
SCHVAC	0.6438	0.6129	0.1073	0
GDP	0.9666	0.2355	-0.0766	0
SP500	0.8790	-0.0920	-0.0276	0

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 13.04729263105491
R-Squared score is 0.9701514692551751


The model performance for testing set
--------------------------------------
Root Mean Squared Error is 14.387754373625608
R-Squared score is 0.9519664612277512
```