Home-LLC-Assignment

Task: Utilize publicly available data on national factors impacting housing supply and demand in the US to build a model and study their effect on home prices.

Approach:

- Chosen variables for the study:
  - Unemployment Rate
  - Per Capita GDP
  - Construction Prices
  - CPI
  - Interest Rates
  - Housing Subsidies
  - Number of New Households Owned
- S&P CASE-SHILLER Index used as a proxy for home prices.
- Most data downloaded from https://fred.stlouisfed.org/.

Additional Variables:

- Earning Population Yearwise
- Net-immigration (Positive impact expected, but suitable data unavailable)
- Marriage Rate (Possible effect due to increased home buying after marriage, but data unavailable)
- Land Availability (Lower availability potentially contributing to rising prices, even with decreasing average house size, but relevant data unavailable)
- Number of Active Listings (Data prior to 2017 unavailable)
- Effects of Epidemics and Demographics
- Classification of Different Age Groups Present

Data Sources:

- CASE-SCHILLER Home Price Index: https://fred.stlouisfed.org/series/CSUSHPISA

- Interest Rates: https://fred.stlouisfed.org/series/FEDFUNDS
- Unemployment Rate: https://fred.stlouisfed.org/series/UNRATE
- Income: https://fred.stlouisfed.org/series/DSPIC96
- Per Capita GDP: https://fred.stlouisfed.org/series/A939RC0A052NBEA
- New House Holds Supplied: https://fred.stlouisfed.org/release?rid=97
- Construction Price Index: https://fred.stlouisfed.org/tags/series?t=construction%3Bprice+index
- Housing Subsidies (Federal): https://fred.stlouisfed.org/tags/series?t=subsidies

```
## Data Preparation and Model Selection
```

Data Preparation:

Data for all chosen variables was downloaded, preprocessed, and combined to create a single dataset. While combining the data, necessary interpolations were made due to different frequencies across the variables.

Model Selection:

Linear Regression was chosen as the primary modeling technique for several reasons:

- **High Correlation:** Preliminary analysis revealed a high degree of correlation between the target variable (home prices) and several independent variables, making linear regression a suitable candidate for capturing these linear relationships.
- **Interpretability:** Linear regression models are highly interpretable, allowing for easier understanding of how each variable affects home prices. This allows for better insights into the factors driving the housing market.
- **Efficiency:** Linear regression models are relatively simple and efficient to train, especially compared to more complex models like deep neural networks. This makes them a practical choice for initial analysis and exploration.

In [131…

```
#Importing Neccesary Libraries
import pandas as pd
import numpy as np
```

In [2]:

```
df_cs = pd.read_csv('CS_Price_index.csv')
```

In [3]:

```
df_cs['DATE'] = pd.to_datetime(df_cs["DATE"])
```

In [4]:
```
df_cs.reset_index(inplace= True)
df_cs.drop(columns = ['index'], inplace=True)
```

In [6]:
```
df_cs["Year"] = pd.DatetimeIndex(df_cs["DATE"]).year
df_cs["Month"] = pd.DatetimeIndex(df_cs["DATE"]).month
print(df_cs.shape)
df_cs.tail()
```

```
(238, 4)
```

Out[6]:

	DATE	CSUSHPISA	Year	Month
233	2023-05-01	302.566	2023	5
234	2023-06-01	304.593	2023	6
235	2023-07-01	306.767	2023	7
236	2023-08-01	309.155	2023	8
237	2023-09-01	311.175	2023	9

In [7]:
```
# Reading Unemployment Rate Data into a dataframe
df_unemp = pd.read_csv("UNRATE.csv")
print(df_unemp.shape)
df_unemp.tail()
```

```
(240, 2)
```

Out[7]:

	DATE	UNRATE
235	2023-07-01	3.5
236	2023-08-01	3.8
237	2023-09-01	3.8
238	2023-10-01	3.9
239	2023-11-01	3.7

In [8]:
```
df_unemp.drop([239,238], inplace = True)
```

In [9]:
```
df_unemp.tail()
```

Out[9]:

	DATE	UNRATE
233	2023-05-01	3.7
234	2023-06-01	3.6
235	2023-07-01	3.5
236	2023-08-01	3.8
237	2023-09-01	3.8

In [10]:
```
# Reading Per Capita GDP Data into a dataframe
df_pcgdp = pd.read_csv("GDP_per_capita.csv", names = ["DATE", "Per_Capita_GDP"], skiprows = 1)
print(df_pcgdp.shape)
df_pcgdp.tail()
```

```
(80, 2)
```

Out[10]:

	DATE	Per_Capita_GDP
75	2022-07-01	65462.0
76	2022-10-01	65783.0
77	2023-01-01	66078.0
78	2023-04-01	66341.0
79	2023-07-01	67083.0

The data is quarterly. We will impute for other months using linear interpolation after we create the final dataframe combining all the data.

In [11]:
```
df_Fed_rate = pd.read_csv("FEDFUNDS.csv")
print(df_Fed_rate.shape)
df_Fed_rate.tail()
```

```
(240, 2)
```

Out[11]:

	DATE	FEDFUNDS
235	2023-07-01	5.12
236	2023-08-01	5.33
237	2023-09-01	5.33
238	2023-10-01	5.33
239	2023-11-01	5.33

In [12]:
```
df_Fed_rate.drop([239,238], inplace = True)
```

In [13]:
```
df_Fed_rate.tail()
```

Out[13]:

	DATE	FEDFUNDS
233	2023-05-01	5.06
234	2023-06-01	5.08
235	2023-07-01	5.12
236	2023-08-01	5.33
237	2023-09-01	5.33

In [15]:
```
# Reading Construction Price Data into a dataframe
df_cons_price_index = pd.read_csv("Construction_price.csv", names = ["DATE", "Cons_Materials"], skiprows = 1)
print(df_cons_price_index.shape)
df_cons_price_index.tail()
```

```
(239, 2)
```

Out[15]:

	DATE	Cons_Materials
234	2023-06-01	337.336
235	2023-07-01	334.576
236	2023-08-01	333.980
237	2023-09-01	332.224
238	2023-10-01	329.690

In [16]:

```
df_cons_price_index.drop([238], inplace = True)
df_cons_price_index.tail()
```

Out[16]:

	DATE	Cons_Materials
233	2023-05-01	337.473
234	2023-06-01	337.336
235	2023-07-01	334.576
236	2023-08-01	333.980
237	2023-09-01	332.224

In [17]:

```
# Consumer Price Index
df_CPI = pd.read_csv("CPI.csv", names = ["DATE", "CPI"], skiprows = 1)
print(df_CPI.shape)
df_CPI.tail()
```

```
(239, 2)
```

Out[17]:

	DATE	CPI
234	2023-06-01	337.336
235	2023-07-01	334.576
236	2023-08-01	333.980
237	2023-09-01	332.224
238	2023-10-01	329.690

In [18]:
```
df_CPI.drop([238], inplace = True)
df_CPI.tail()
```

Out[18]:

	DATE	CPI
233	2023-05-01	337.473
234	2023-06-01	337.336
235	2023-07-01	334.576
236	2023-08-01	333.980
237	2023-09-01	332.224

In [20]:
```
# Monthly new house owned
df_house = pd.read_csv("COMPUTSA.csv", names = ["DATE", "Houses"], skiprows = 1)
print(df_house.shape)
df_house.tail()
```

```
(239, 2)
```

Out[20]:

	DATE	Houses
234	2023-06-01	1492.0
235	2023-07-01	1334.0
236	2023-08-01	1370.0
237	2023-09-01	1478.0
238	2023-10-01	1410.0

In [21]:
```
df_house.drop([238], inplace = True)
df_house.tail()
```

Out[21]:

	DATE	Houses
233	2023-05-01	1534.0
234	2023-06-01	1492.0
235	2023-07-01	1334.0
236	2023-08-01	1370.0
237	2023-09-01	1478.0

In [22]:

```
# Housing Subsidies
df_subsidy = pd.read_csv("Housing_subsidies.csv", names = ["DATE", "Subsidy"], skiprows = 1)
print(df_subsidy.shape)
df_subsidy.tail()
```

```
(20, 2)
```

Out[22]:

	DATE	Subsidy
15	2018-01-01	38.859
16	2019-01-01	40.185
17	2020-01-01	44.147
18	2021-01-01	45.299
19	2022-01-01	48.021

In [23]:

```
# Real Disposable Income

df_income = pd.read_csv("Income.csv", names = ["DATE", "Income"], skiprows = 1)
print(df_income.shape)
df_income.tail()
```

```
(239, 2)
```

Out[23]:

	DATE	Income
234	2023-06-01	16809.5
235	2023-07-01	16796.9
236	2023-08-01	16799.7
237	2023-09-01	16804.8
238	2023-10-01	16848.7

In [24]:

```
df_income.drop([238], inplace = True)
df_income.tail()
```

Out[24]:

	DATE	Income
233	2023-05-01	16818.5
234	2023-06-01	16809.5
235	2023-07-01	16796.9
236	2023-08-01	16799.7
237	2023-09-01	16804.8

In [27]:

```
# Concating dataframes having monthly data to create one dataframe
df = pd.DataFrame()
df_bymonth = [df_cs, df_unemp, df_Fed_rate, df_cons_price_index, df_CPI, df_house, df_income]
for df1 in df_bymonth:
    df1["DATE"] = pd.to_datetime(df1["DATE"])
    df1 = df1.set_index("DATE")
    df = pd.concat([df,df1], axis = 1)
print(df.shape)
df.head()
```

```
(238, 10)
```

Out[27]:

DATE	CSUSHPISA	Year	Month	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income
2003-12-01	140.179	2003	12	NaN	5.7	0.98	149.7	149.7	1716.0	11057.2
2004-01-01	141.646	2004	1	52179.0	5.7	1.00	150.0	150.0	1709.0	11051.2
2004-02-01	143.192	2004	2	NaN	5.6	1.01	153.4	153.4	1718.0	11071.0
2004-03-01	145.059	2004	3	NaN	5.8	1.00	156.5	156.5	1794.0	11115.6
2004-04-01	146.593	2004	4	52469.0	5.6	1.00	160.1	160.1	1938.0	11153.3

In [31]:

```
# Merging Housing Subsidaries
Housing_subsidy = [df_subsidy]
for df1 in Housing_subsidy:
    if "Year" not in df1.columns:
        df1["Year"] = pd.DatetimeIndex(df1["DATE"]).year
        df1.set_index("DATE", inplace = True)
        df = pd.merge(df, df1, how = "left", on = "Year")
    else:
        df1.set_index("DATE", inplace = True)
        df = pd.merge(df, df1, how = "left", on = "Year")
df["DATE"] = df_cs["DATE"]
df.set_index("DATE", inplace = True)
df.head()
```

Out[31]:

DATE	CSUSHPISA	Year	Month	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income	Subsidy
2003-12-01	140.179	2003	12	NaN	5.7	0.98	149.7	149.7	1716.0	11057.2	25.930
2004-01-01	141.646	2004	1	52179.0	5.7	1.00	150.0	150.0	1709.0	11051.2	27.201
2004-02-01	143.192	2004	2	NaN	5.6	1.01	153.4	153.4	1718.0	11071.0	27.201
2004-03-01	145.059	2004	3	NaN	5.8	1.00	156.5	156.5	1794.0	11115.6	27.201
2004-04-01	146.593	2004	4	52469.0	5.6	1.00	160.1	160.1	1938.0	11153.3	27.201

Done Building Data Frame Built

EDA

In [33]:
```
df.isnull().sum()
```

Out[33]:
```
CSUSHPISA          0
Year               0
Month              0
Per_Capita_GDP   159
UNRATE             0
FEDFUNDS           0
Cons_Materials     0
CPI                0
Houses             0
Income             0
Subsidy            9
dtype: int64
```

The "Per_Capita_GDP" column has missing values because the data was quarterly. The missing values in the other columns is due to unavailability of fresh data. We will first fill the missing values in the "Per_Capita_GDP" column using linear interpolation. We will drop the rows having missing values in the other columns.

In [34]:
```
# Filling missing values in the Per_Capita_GDP column using linear interpolation
df["Per_Capita_GDP"] = df["Per_Capita_GDP"].interpolate()
```

In [36]:
```
df.isnull().sum()
```

Out[36]:
```
CSUSHPISA         0
Year              0
Month             0
Per_Capita_GDP    1
UNRATE            0
FEDFUNDS          0
Cons_Materials    0
CPI               0
Houses            0
Income            0
Subsidy           9
dtype: int64
```

In [39]:
```
df.dropna(inplace = True)
```

In [40]:
```
df.isnull().sum()
```

Out[40]:
```
CSUSHPISA         0
Year              0
Month             0
Per_Capita_GDP    0
UNRATE            0
FEDFUNDS          0
Cons_Materials    0
CPI               0
Houses            0
Income            0
Subsidy           0
dtype: int64
```

In [41]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 228 entries, 2004-01-01 to 2022-12-01
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   CSUSHPISA       228 non-null    float64
 1   Year            228 non-null    int64  
 2   Month           228 non-null    int64  
 3   Per_Capita_GDP  228 non-null    float64
 4   UNRATE          228 non-null    float64
 5   FEDFUNDS        228 non-null    float64
 6   Cons_Materials  228 non-null    float64
 7   CPI             228 non-null    float64
 8   Houses          228 non-null    float64
 9   Income          228 non-null    float64
 10  Subsidy         228 non-null    float64
dtypes: float64(9), int64(2)
memory usage: 21.4 KB
```

In [42]:
```
df.shape
```

Out[42]:
```
(228, 11)
```

In [43]:
```
df.tail()
```

Out[43]:

DATE	CSUSHPISA	Year	Month	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income	Subsidy
2022-08-01	301.473	2022	8	65569.000000	3.7	2.33	342.753	342.753	1355.0	16161.4	48.021
2022-09-01	299.353	2022	9	65676.000000	3.5	2.56	336.464	336.464	1438.0	16184.9	48.021
2022-10-01	298.873	2022	10	65783.000000	3.7	3.08	333.796	333.796	1348.0	16223.5	48.021
2022-11-01	298.269	2022	11	65881.333333	3.6	3.78	330.369	330.369	1543.0	16229.6	48.021
2022-12-01	297.413	2022	12	65979.666667	3.5	4.10	326.449	326.449	1390.0	16265.1	48.021

This is the final Dataset, Let us save this as "final_df.csv"

In [44]:
```
df.to_csv("final_df.csv")
```

In [45]:
```
df = pd.read_csv("final_df.csv").set_index("DATE")
df.head()
```

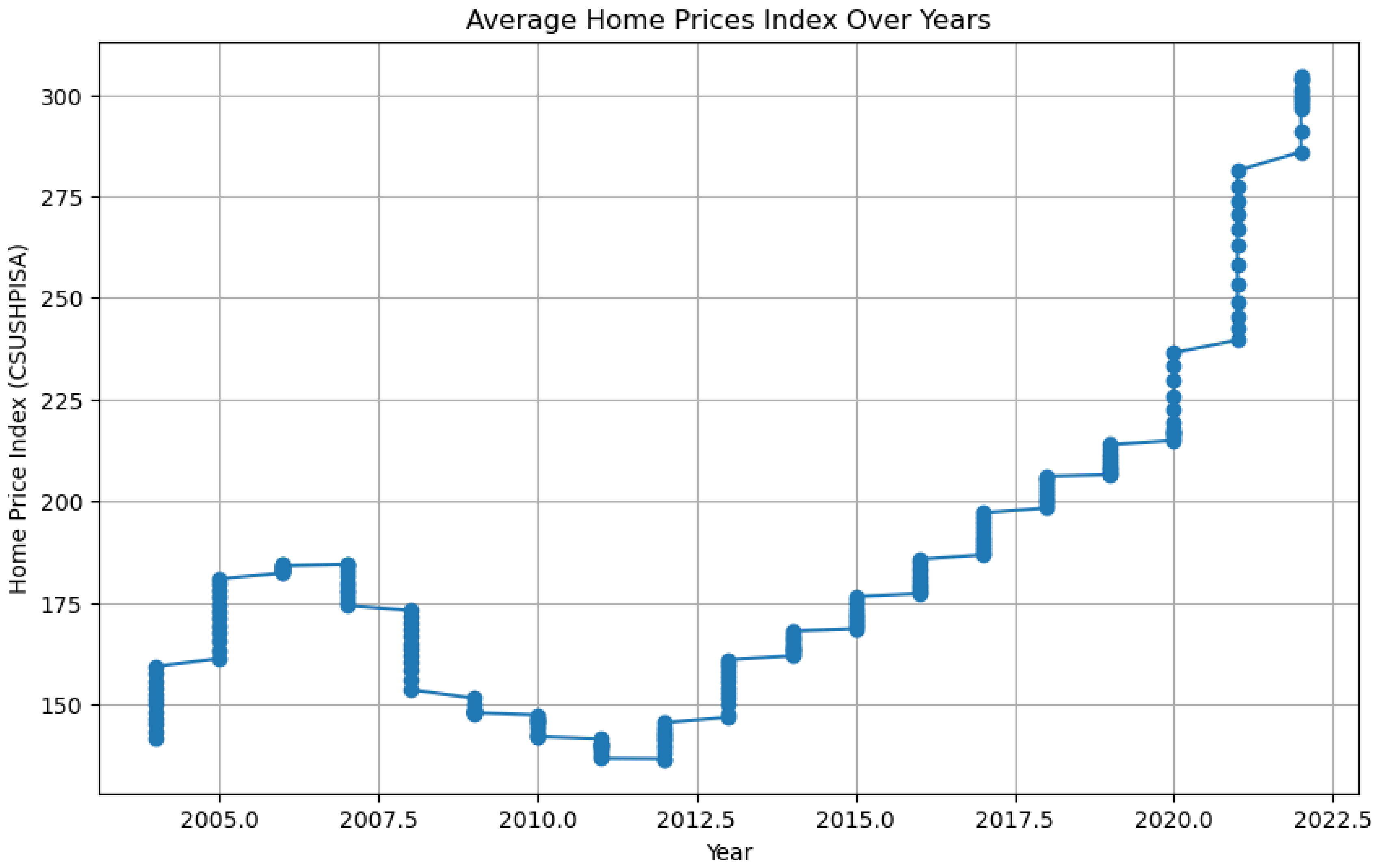
Out[45]:

DATE	CSUSHPISA	Year	Month	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income	Subsidy
2004-01-01	141.646	2004	1	52179.000000	5.7	1.00	150.0	150.0	1709.0	11051.2	27.201
2004-02-01	143.192	2004	2	52275.666667	5.6	1.01	153.4	153.4	1718.0	11071.0	27.201
2004-03-01	145.059	2004	3	52372.333333	5.8	1.00	156.5	156.5	1794.0	11115.6	27.201
2004-04-01	146.593	2004	4	52469.000000	5.6	1.00	160.1	160.1	1938.0	11153.3	27.201
2004-05-01	148.186	2004	5	52591.000000	5.6	1.00	162.7	162.7	1893.0	11208.9	27.201

In [46]:
```
import matplotlib.pyplot as plt
```

In [48]:
```
# Year-wise analysis
plt.figure(figsize=(10, 6))
plt.plot(df['Year'], df['CSUSHPISA'], marker='o', linestyle='-')
plt.xlabel('Year')
plt.ylabel('Home Price Index (CSUSHPISA)')
plt.title('Average Home Prices Index Over Years')
```

```
plt.grid(True)
plt.show()
```



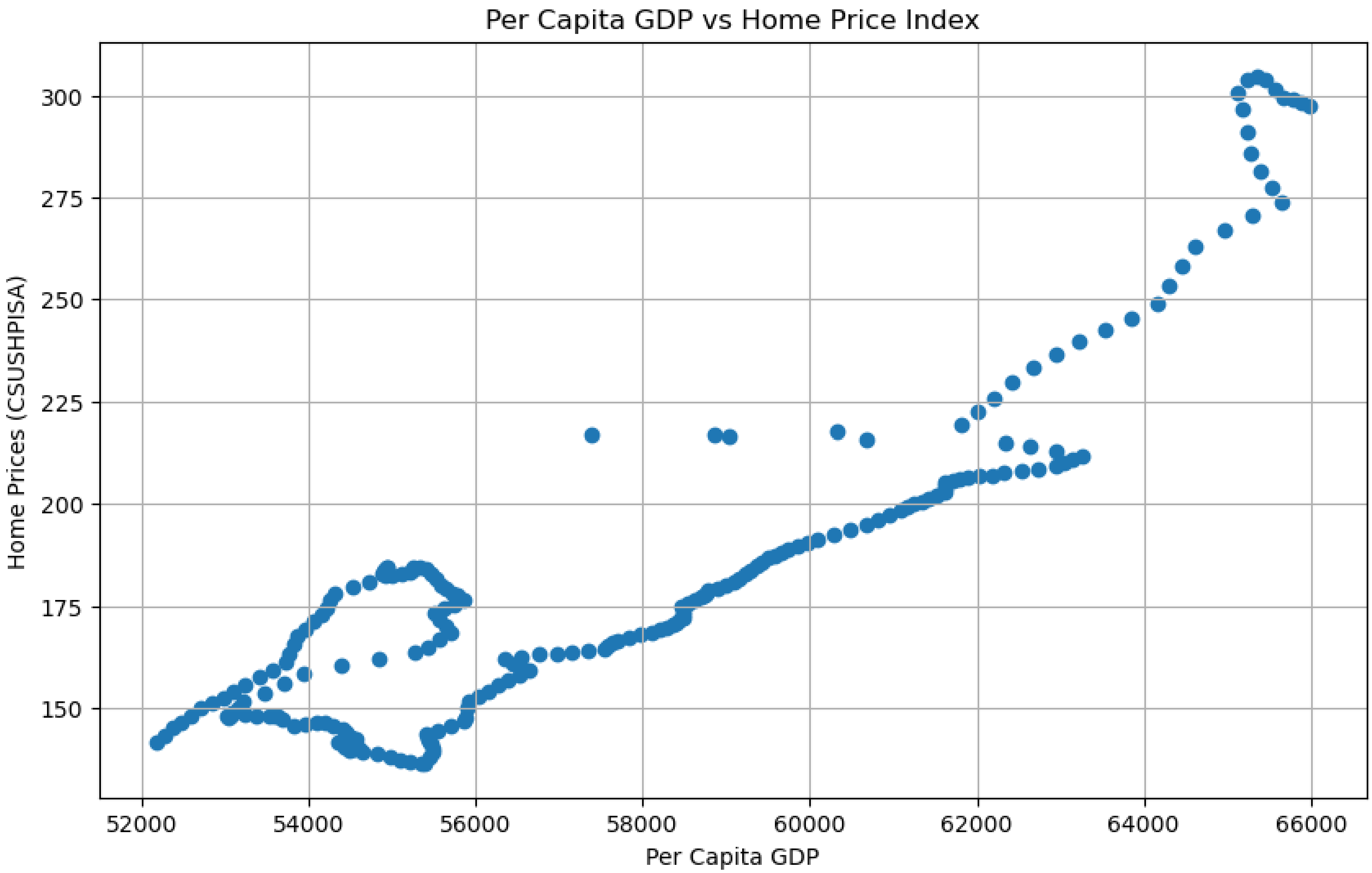
Key Insights:-

1. Home prices have increased significantly over the past 10 years. The average home price index in the graph has increased from 150 in 2005 to 300 in 2022.5, representing a 100% increase.

2. Home price growth has been relatively steady over the past 10 years. There have been no major spikes or declines in home prices during this period. This suggests that the housing market is relatively stable.
3. Home prices are expected to continue to increase in the future. The graph shows a clear upward trend in home prices over the past 10 years. This trend is likely to continue in the future, given the strong demand for housing and the limited supply of homes.

In [51]:

```
plt.figure(figsize=(10, 6))
plt.scatter(df['Per_Capita_GDP'], df['CSUSHPISA'])
plt.xlabel('Per Capita GDP')
plt.ylabel('Home Prices (CSUSHPISA)')
plt.title('Per Capita GDP vs Home Price Index')
plt.grid(True)
plt.show()
```



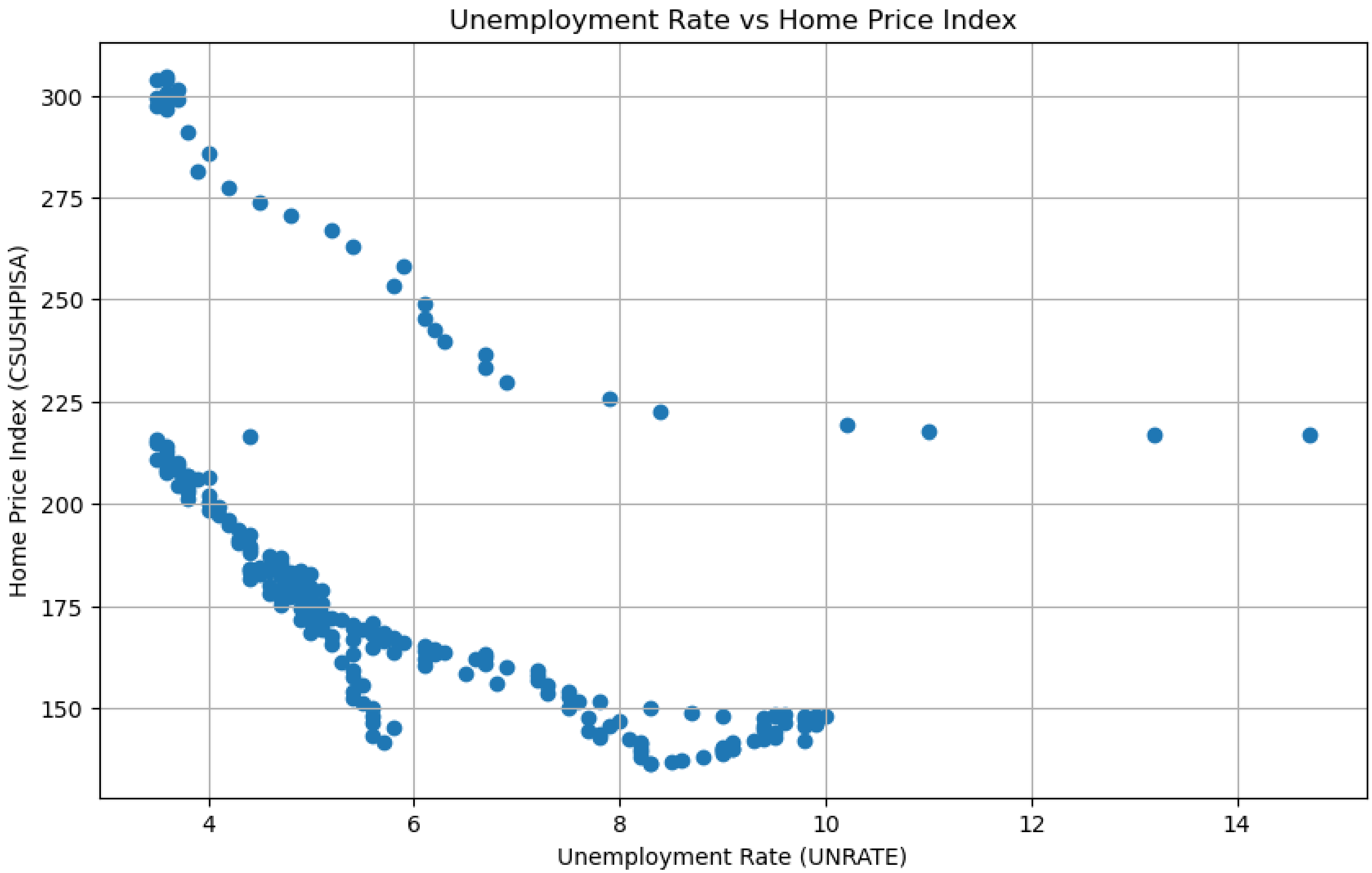
Major Insights:

1. As per capita GDP increases, the home price index increases. This suggests that there is a positive correlation between per capita GDP and the home price index. This is likely because as people become wealthier, they are able to afford to spend more on housing.
2. The rate of increase in the home price index is greater than the rate of increase in per capita GDP. This is evident from the steeper slope of the line in the graph. This suggests that the relationship between per capita GDP and the home price index is not linear. Instead, it is likely

that the home price index is increasing at an exponential rate relative to per capita GDP.

3. The relationship between per capita GDP and the home price index is stronger in recent years. This is evident from the fact that the line in the graph is becoming steeper in recent years. This is likely due to a number of factors, including low interest rates, strong economic growth, and a limited supply of housing.

In [55]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['UNRATE'], df['CSUSHPISA'])
plt.xlabel('Unemployment Rate (UNRATE)')
plt.ylabel('Home Price Index (CSUSHPISA)')
plt.title('Unemployment Rate vs Home Price Index')
plt.grid(True)
plt.show()
```



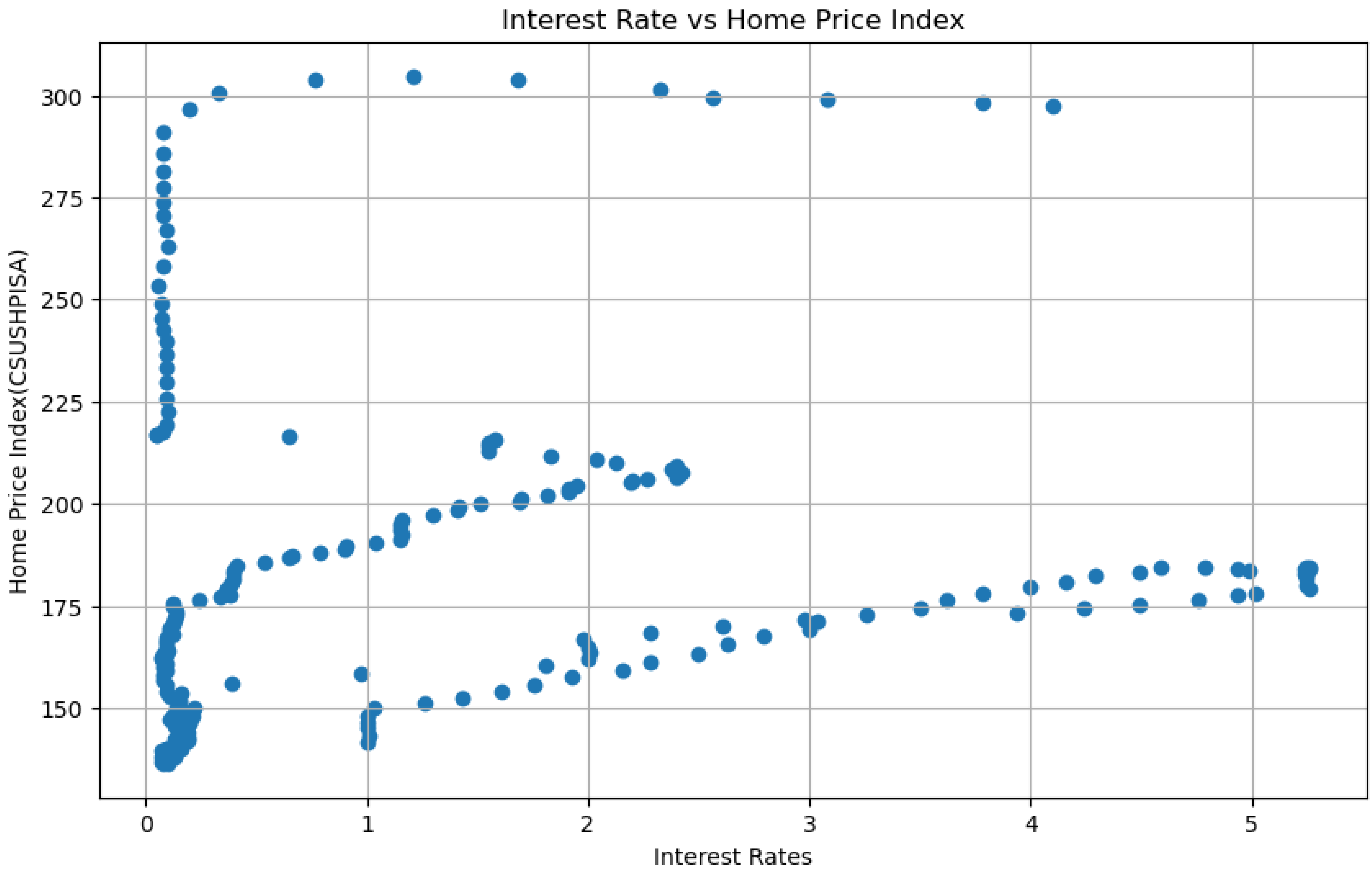
Key Insights

1. The unemployment rate and the home price index are negatively correlated. This means that as the unemployment rate increases, the home price index tends to decrease, and vice versa. This is likely because when more people are unemployed, there is less demand for housing, which can lead to lower home prices.

2. The correlation between the unemployment rate and the home price index is stronger in recent years. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between the unemployment rate and the home price index has become stronger over time.

3. The relationship between the unemployment rate and the home price index is not linear. This means that the decrease in home prices is not proportional to the increase in the unemployment rate. Instead, the relationship is likely more complex, with other factors such as interest rates and economic growth also playing a role.

In [54]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['FEDFUNDS'], df['CSUSHPISA'])
plt.xlabel('Interest Rates')
plt.ylabel('Home Price Index(CSUSHPISA)')
plt.title('Interest Rate vs Home Price Index')
plt.grid(True)
plt.show()
```

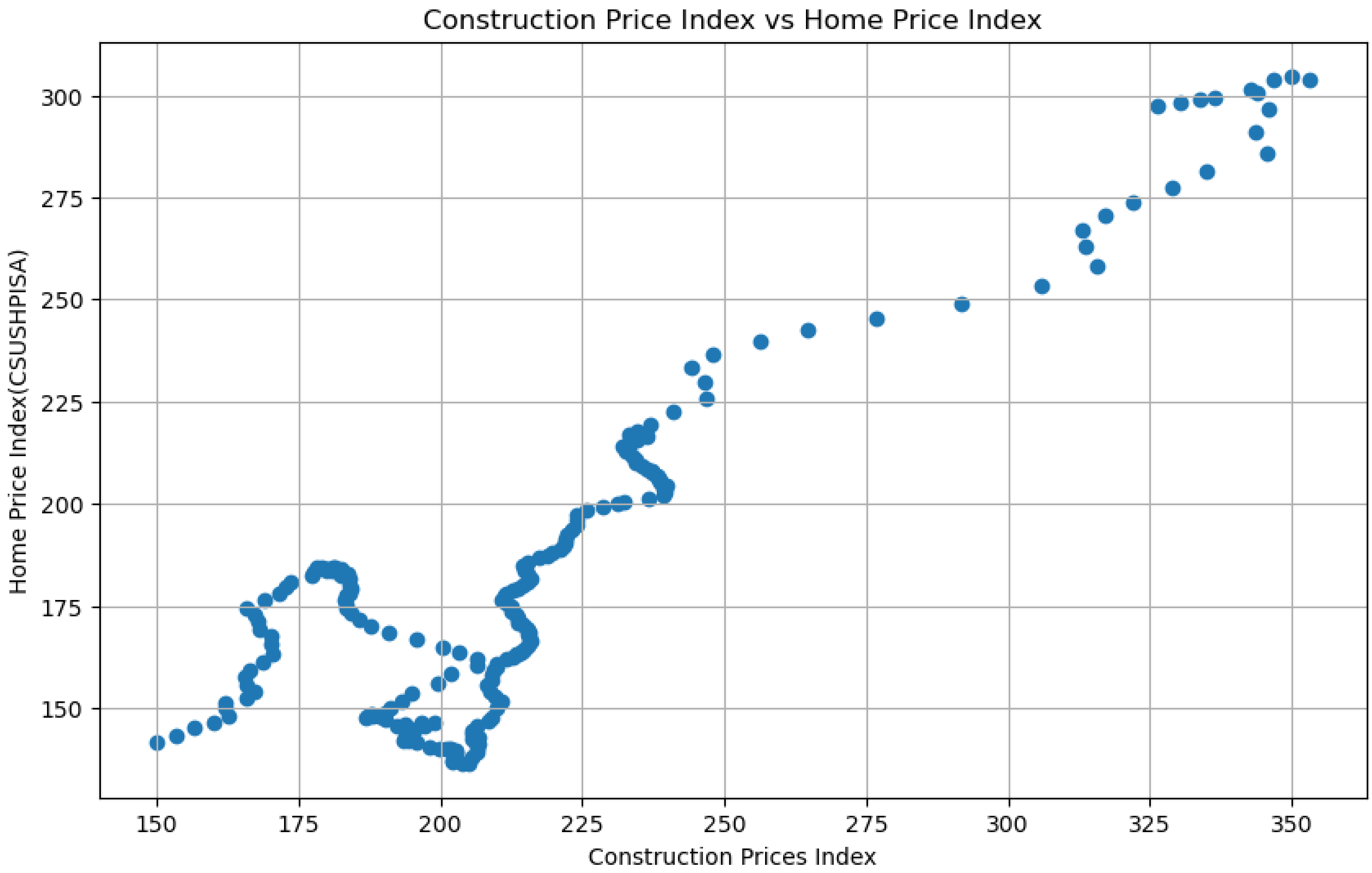


Key Insights : -

1. There is a negative correlation between interest rates and the home price index. This means that as interest rates increase, the home price index tends to decrease, and vice versa. This is because higher interest rates make it more expensive to borrow money to buy a home, which can lead to lower demand for housing and lower home prices.

2. The correlation between interest rates and the home price index is stronger in recent years. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between interest rates and the home price index has become stronger over time.

In [57]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['CPI'], df['CSUSHPISA'])
plt.xlabel('Construction Prices Index')
plt.ylabel('Home Price Index(CSUSHPISA)')
plt.title('Construction Price Index vs Home Price Index')
plt.grid(True)
plt.show()
```

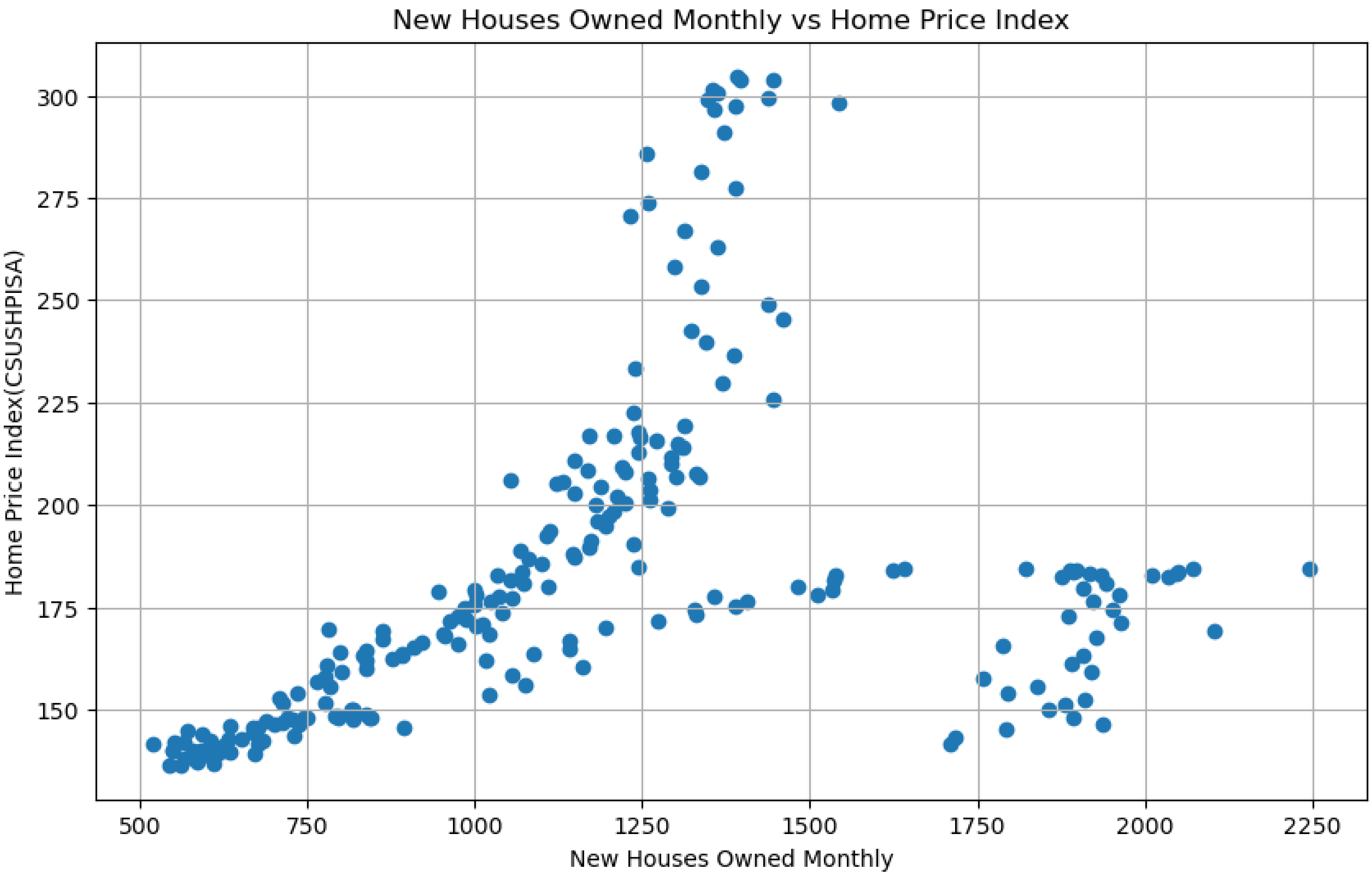


Key Insights:-

1. There is a positive correlation between the construction price index and the home price index. This means that as the construction price index increases, the home price index also tends to increase. This is likely because the cost of construction is a significant component of the home price index.

2. The correlation between the construction price index and the home price index is stronger in recent years. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between the construction price index and the home price index has become stronger over time.

In [61]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Houses'], df['CSUSHPISA'])
plt.xlabel('New Houses Owned Monthly')
plt.ylabel('Home Price Index(CSUSHPISA)')
plt.title('New Houses Owned Monthly vs Home Price Index')
plt.grid(True)
plt.show()
```

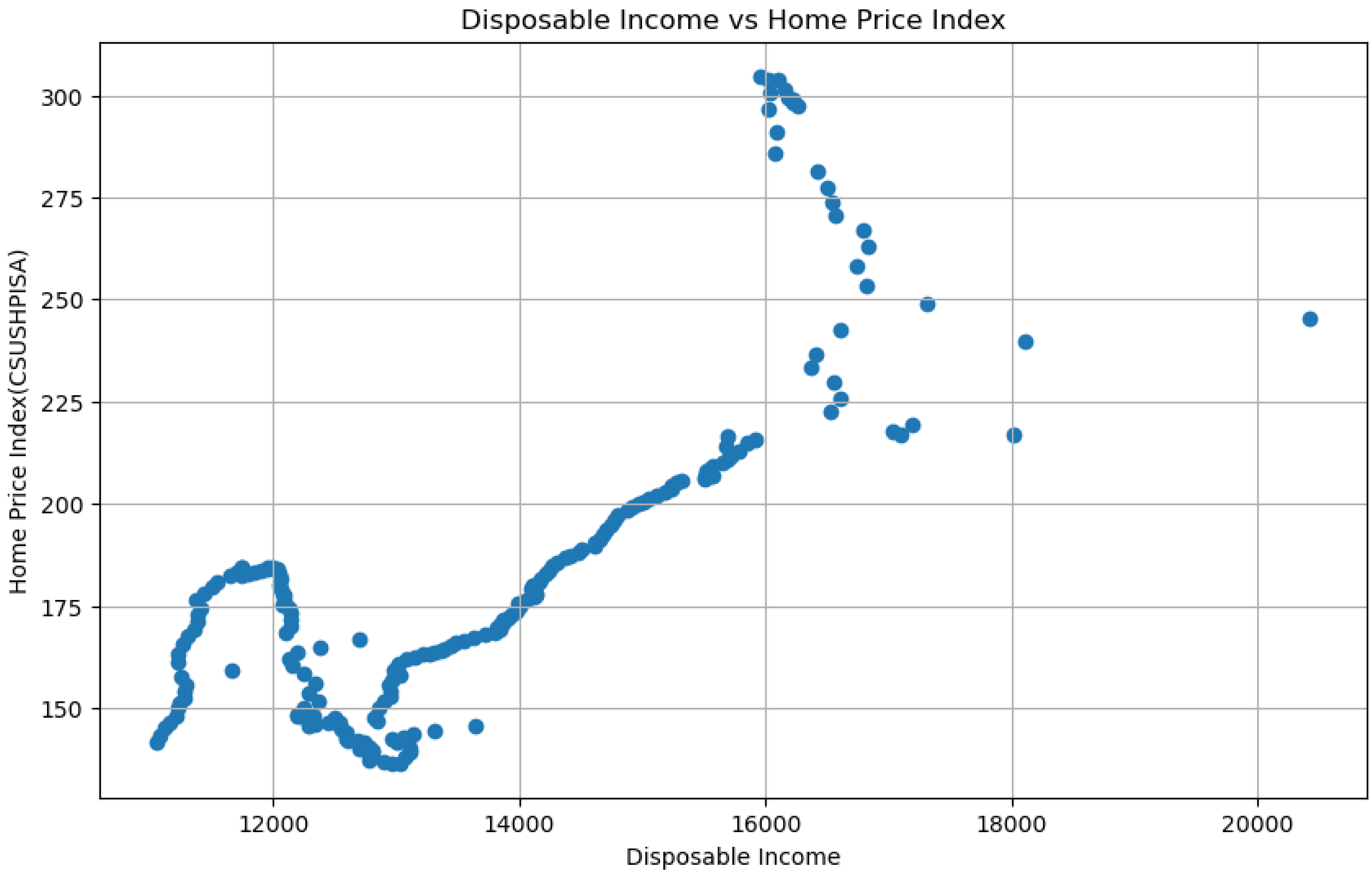


Key Insights:-

1. There is a positive correlation between the home price index and new houses owned monthly. This means that as the home price index increases, the number of new houses owned monthly also tends to increase. This is likely because people are more likely to buy a home when the home price index is increasing, as they believe that they will be able to sell their home for a profit in the future.

2. The correlation between the home price index and new houses owned monthly is stronger in recent years. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between the home price index and new houses owned monthly has become more pronounced over time.

In [59]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Income'], df['CSUSHPISA'])
plt.xlabel('Disposable Income')
plt.ylabel('Home Price Index(CSUSHPISA)')
plt.title('Disposable Income vs Home Price Index')
plt.grid(True)
plt.show()
```

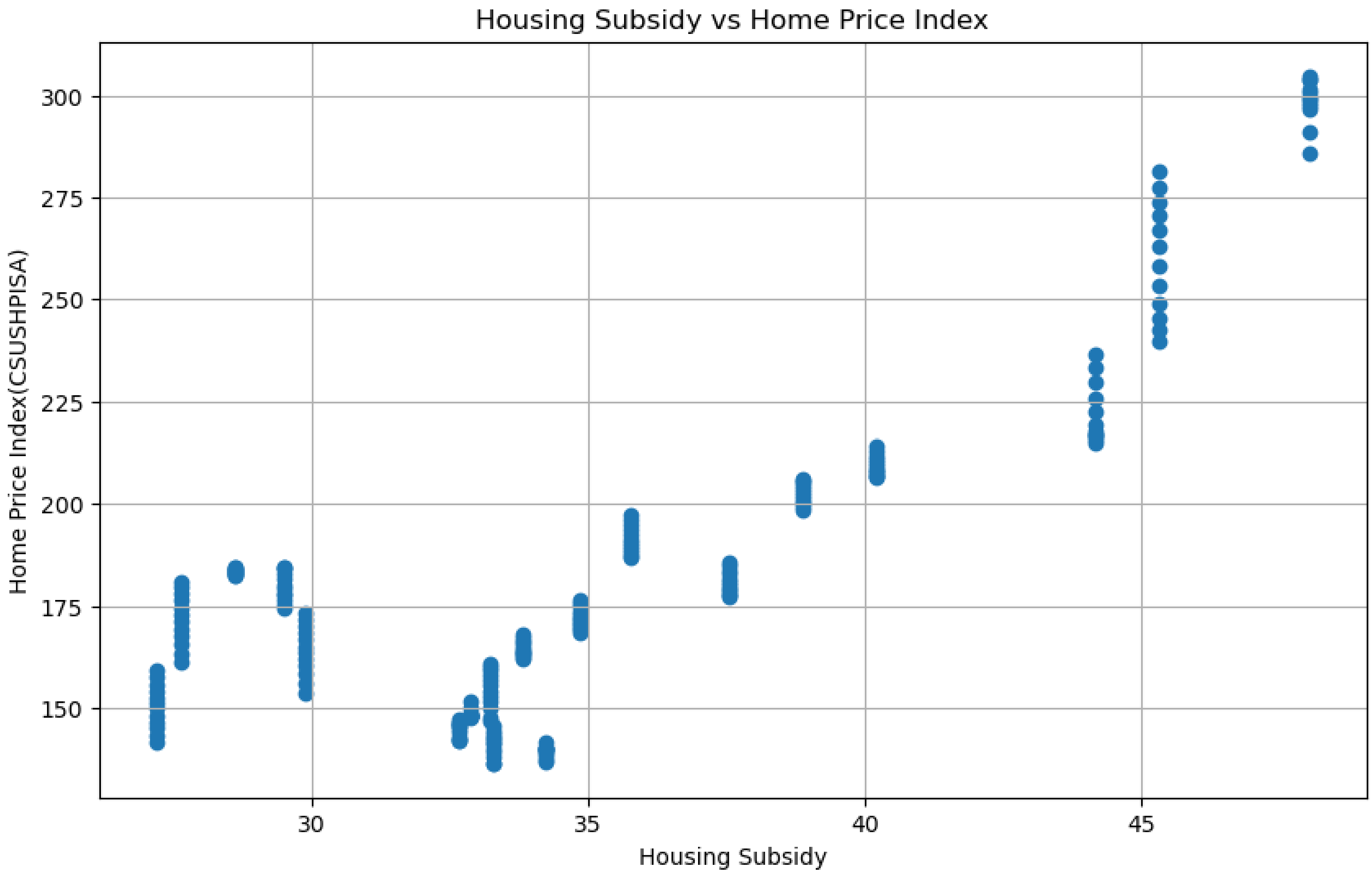


Key Insights:-

1. There is a positive correlation between the house price index and disposable income. This means that as disposable income increases, the house price index also tends to increase. This is likely because people with higher disposable income are more likely to be able to afford to buy a house.

2. The correlation between the house price index and disposable income has become stronger over time. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between the house price index and disposable income has become more pronounced over time.

In [60]:
```
plt.figure(figsize=(10, 6))
plt.scatter(df['Subsidy'], df['CSUSHPISA'])
plt.xlabel('Housing Subsidy')
plt.ylabel('Home Price Index(CSUSHPISA)')
plt.title('Housing Subsidy vs Home Price Index')
plt.grid(True)
plt.show()
```



Key Insights:-

1. There is a negative correlation between housing subsidy and house price index. This means that as housing subsidy increases, the house price index tends to decrease. This is likely because housing subsidies make it more affordable for people to buy homes, which can lead to increased demand and higher home prices. However, government subsidies can also reduce the supply of affordable housing, which can also lead to higher home prices.

2. The correlation between housing subsidy and house price index is stronger in recent years. This is evident from the fact that the points in the scatter plot are more tightly clustered together in recent years. This suggests that the relationship between housing subsidy and house price index has become more pronounced over time.

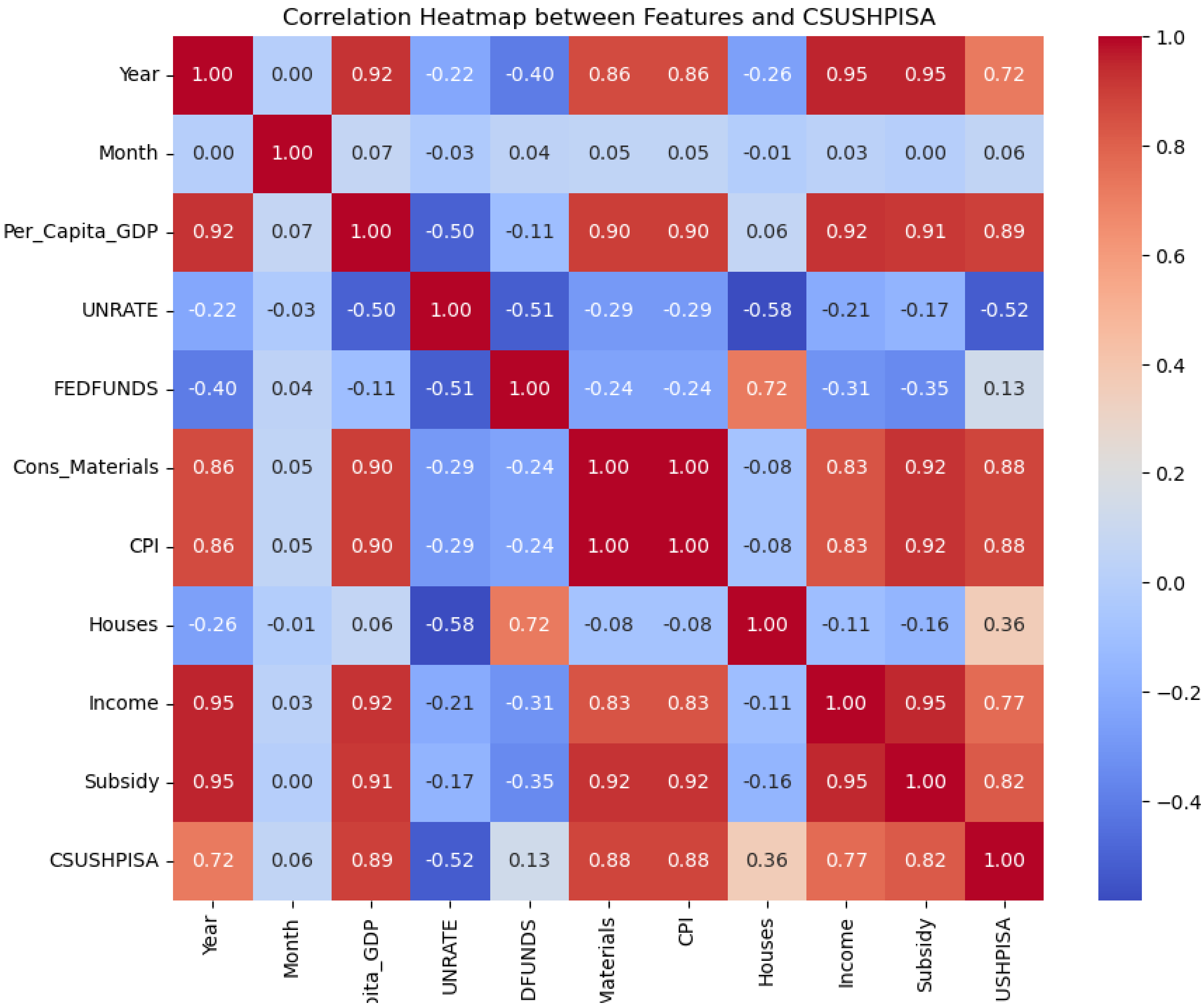
Correlation Analysis

In [62]:
```
import seaborn as sns
```

In [63]:
```
target_variable = df['CSUSHPISA']
features = df.drop(columns=['CSUSHPISA'])

correlation_matrix = pd.concat([features, target_variable], axis=1).corr()

# Plotting the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap between Features and CSUSHPISA')
plt.show()
```


Correlation Heatmap between Features and CSUSHPISA
Year
Month
Per_Capita_GDP
UNRATE
FEDFUNDS
Cons_Materials
CPI
Houses
Income
Subsidy
CSUSHPISA
1.00 0.00 0.92 -0.22 -0.40 0.86 0.86 -0.26 0.95 0.95 0.72
0.00 1.00 0.07 -0.03 0.04 0.05 0.05 -0.01 0.03 0.00 0.06
0.92 0.07 1.00 -0.50 -0.11 0.90 0.90 0.06 0.92 0.91 0.89
-0.22 -0.03 -0.50 1.00 -0.51 -0.29 -0.29 -0.58 -0.21 -0.17 -0.52
-0.40 0.04 -0.11 -0.51 1.00 -0.24 -0.24 0.72 -0.31 -0.35 0.13
0.86 0.05 0.90 -0.29 -0.24 1.00 1.00 -0.08 0.83 0.92 0.88
0.86 0.05 0.90 -0.29 -0.24 1.00 1.00 -0.08 0.83 0.92 0.88
-0.26 -0.01 0.06 -0.58 0.72 -0.08 -0.08 1.00 -0.11 -0.16 0.36
0.95 0.03 0.92 -0.21 -0.31 0.83 0.83 -0.11 1.00 0.95 0.77
0.95 0.00 0.91 -0.17 -0.35 0.92 0.92 -0.16 0.95 1.00 0.82
0.72 0.06 0.89 -0.52 0.13 0.88 0.88 0.36 0.77 0.82 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Analysis of Heatmap : -

1. There is a strong positive correlation between the the S&P Case-Schiller Home Price index (HPI) and most of the other variables. This means that as the HPI increases, the other variables also tend to increase. This is likely because the HPI is a good proxy for the overall health of the housing market, and when the housing market is strong, other economic variables, such as GDP growth and employment, are also likely to be strong.

2. The strongest positive correlations are between the HPI and per capita GDP, the construction price index, and the consumer price index (CPI). This suggests that these variables are particularly important drivers of the housing market.

3. There is a negative correlation between the HPI and the unemployment rate. This is because when the unemployment rate is high, fewer people are able to afford to buy homes, which can lead to lower demand and lower home prices.

4. The correlation between the HPI and interest rates is relatively weak. This suggests that interest rates are not a major driver of the housing market in the short term. However, in the long term, interest rates can have a significant impact on home prices by making it more or less expensive to borrow money to buy a home.

5. The correlation between the HPI and the number of houses sold is very strong. This suggests that the demand for housing is a major driver of the housing market.

Overall, the correlation heatmap suggests that the HPI is strongly correlated with a number of other economic variables, but the strongest correlations are with per capita GDP, the construction price index, and the CPI. The negative correlation between the HPI and the unemployment rate suggests that the unemployment rate is a factor that can dampen the housing market. The relatively weak correlation between the HPI and interest rates suggests that interest rates are not a major driver of the housing market in the short term, but they can have a significant impact on home prices in the long term. Finally, the very strong correlation between the HPI and the number of houses sold suggests that the demand for housing is a major driver of the housing market.

Model Building for Predicting S&P Case-Schiller Home Price index (HPI)

We don't need the month and year columns for our analysis. So, let's drop these colums.

In [64]:
```
# Dropping year and month columns
df.drop(columns = ["Year", "Month"], inplace = True)
```

In [65]:
```
df.corr()
```

Out[65]:

	CSUSHPISA	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income	Subsidy
CSUSHPISA	1.000000	0.889204	-0.524812	0.133636	0.881236	0.881236	0.360267	0.770447	0.815288
Per_Capita_GDP	0.889204	1.000000	-0.503517	-0.107949	0.896284	0.896284	0.062943	0.920462	0.911448
UNRATE	-0.524812	-0.503517	1.000000	-0.514132	-0.287496	-0.287496	-0.581244	-0.207413	-0.166992
FEDFUNDS	0.133636	-0.107949	-0.514132	1.000000	-0.244405	-0.244405	0.717158	-0.313565	-0.348070
Cons_Materials	0.881236	0.896284	-0.287496	-0.244405	1.000000	1.000000	-0.075495	0.834810	0.920305
CPI	0.881236	0.896284	-0.287496	-0.244405	1.000000	1.000000	-0.075495	0.834810	0.920305
Houses	0.360267	0.062943	-0.581244	0.717158	-0.075495	-0.075495	1.000000	-0.106112	-0.158639
Income	0.770447	0.920462	-0.207413	-0.313565	0.834810	0.834810	-0.106112	1.000000	0.947331
Subsidy	0.815288	0.911448	-0.166992	-0.348070	0.920305	0.920305	-0.158639	0.947331	1.000000

In [66]:
```
df.describe()
```

Out[66]:

	CSUSHPISA	Per_Capita_GDP	UNRATE	FEDFUNDS	Cons_Materials	CPI	Houses	Income	Subsidy
count	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000	228.000000
mean	183.127022	57706.026316	6.013158	1.310789	216.311632	216.311632	1176.714912	13630.801316	35.137842
std	40.848812	3821.383548	2.086817	1.618934	43.017225	43.017225	420.104982	1781.841800	5.807508
min	136.533000	52179.000000	3.500000	0.050000	150.000000	150.000000	520.000000	11051.200000	27.201000
25%	151.464750	54599.750000	4.500000	0.117500	189.275000	189.275000	830.250000	12212.725000	29.876000
50%	174.617000	56212.333333	5.350000	0.375000	209.200000	209.200000	1155.000000	13072.950000	33.806000
75%	200.139000	60716.416667	7.525000	2.062500	231.525000	231.525000	1374.750000	14982.900000	38.859000
max	304.724000	65979.666667	14.700000	5.260000	353.015000	353.015000	2245.000000	20422.600000	48.021000

In [68]:
```
# Droping Columns to reduce any issue of Multicolinearity
df.drop(columns=['Cons_Materials', 'Subsidy'], inplace=True)
```

In [69]:
```
# Separating the target variable and the independent variable
y = df.pop("CSUSHPISA")
X = df
```

In [77]:
```
#Importing Necessary Libraries
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
```

In [73]:
```
# Scaling
scalar = MinMaxScaler()
X = scalar.fit_transform(X)
```

In [74]:
```
# Splitting data into train and validation sets
X_train, X_valid, y_train, y_valid = train_test_split(X,y, test_size= 0.2, random_state= 42)
```

In [128…
```
# Linear Regression Model
modelLR = LinearRegression()
modelLR.fit(X_train, y_train)
pred = modelLR.predict(X_valid)
```

```
score = r2_score(pred, y_valid)
print("The r2_score for the validation set is: ", score)
```

```
The r2_score for the validation set is:  0.6766625795652645
```

In [ ]: