In [35]:
```
import pandas as pd

# Define the file path to your dataset
file_path = 'house_file.csv'
```

In [37]:
```
sp_data = pd.read_csv(file_path, delimiter='\t')
```

In [41]:
```
print(sp_data.columns)
```

```
Index(['DATE,CSUSHPISA'], dtype='object')
```

In [42]:
```
sp_data = pd.read_csv(file_path, delimiter=',', parse_dates=['DATE'], dayfirst=True)
```

In [43]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [44]:
```
sp_data.set_index('DATE', inplace=True)
```

In [45]:
```
key_factors_data = pd.read_csv('economic_factors.csv', parse_dates=['Date'])
```

In [46]:
```
merged_data = sp_data.join(key_factors_data.set_index('Date'), how='inner')
```

In [47]:
```
if not merged_data.empty:
    print("merged_data is not empty")
else:
    print("merged_data is empty")
```

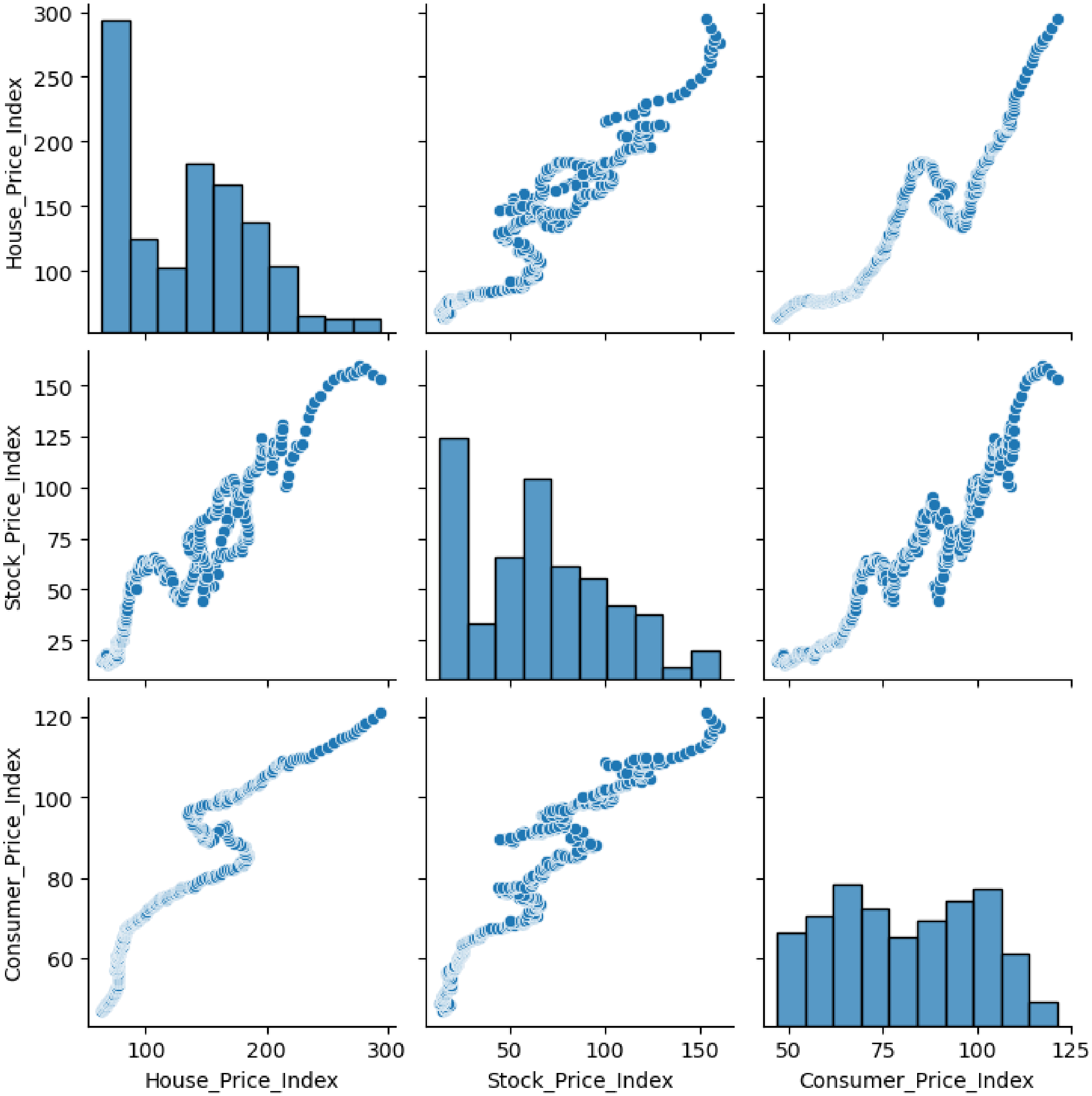
```
merged_data is not empty
```

In [48]:
```
missing_values = merged_data.isnull().sum()
```

In [49]:
```
print("Missing Values:\n", missing_values)
```

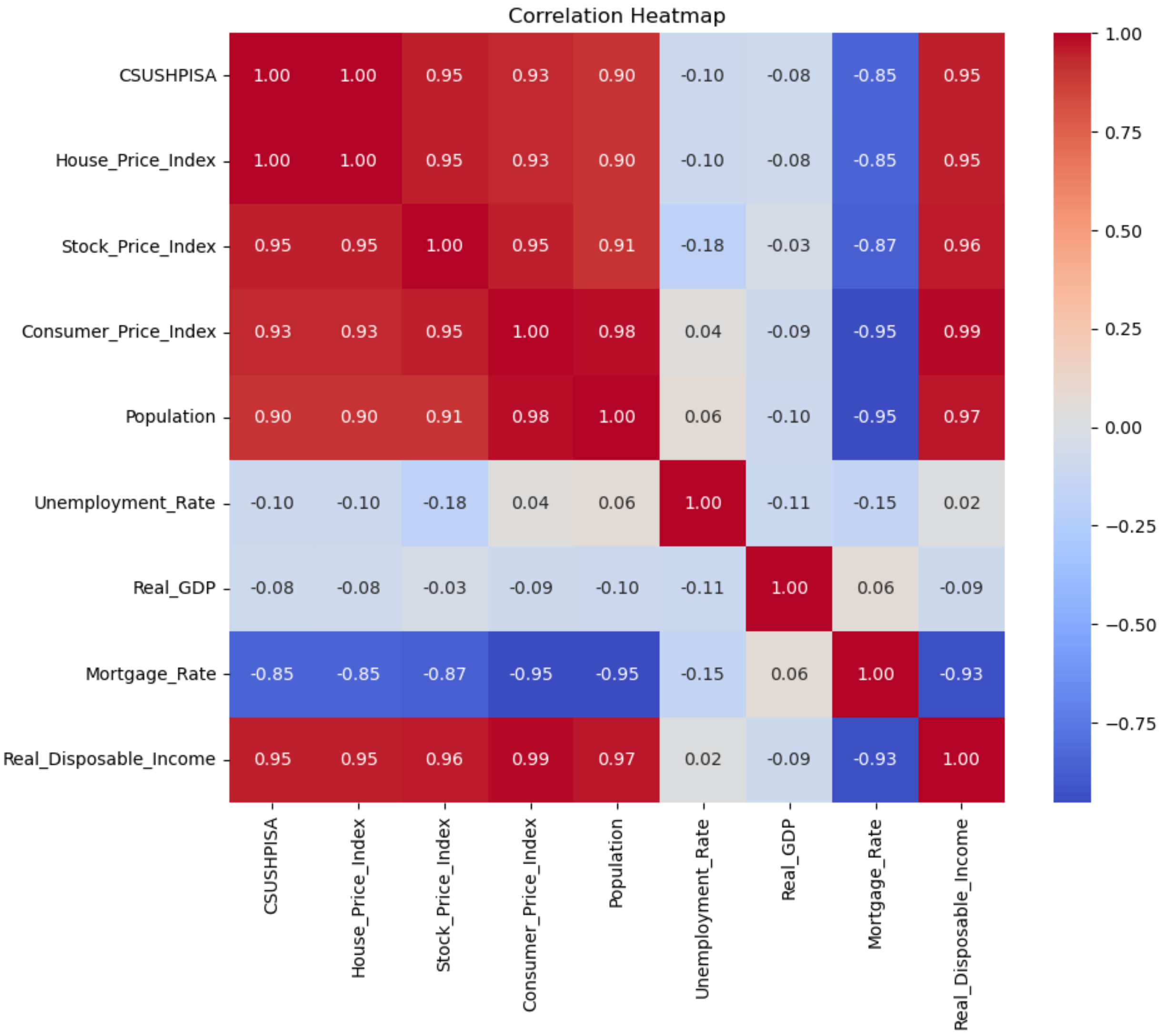
```
Missing Values:
 CSUSHPISA                 0
House_Price_Index         0
Stock_Price_Index         0
Consumer_Price_Index      0
Population                0
Unemployment_Rate         0
Real_GDP                  0
Mortgage_Rate             0
Real_Disposable_Income    0
dtype: int64
```

In [50]:
```
sns.pairplot(data=merged_data, vars=['House_Price_Index', 'Stock_Price_Index', 'Consum
plt.show()
```



In [51]:

```
correlation_matrix = merged_data.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



	CSUSHPISA	House_Price_Index	Stock_Price_Index	Consumer_Price_Index	Population	Unemployment_Rate	Real_GDP	Mortgage_Rate	Real_Disposable_Income
CSUSHPISA	1.00	1.00	0.95	0.93	0.90	-0.10	-0.08	-0.85	0.95
House_Price_Index	1.00	1.00	0.95	0.93	0.90	-0.10	-0.08	-0.85	0.95
Stock_Price_Index	0.95	0.95	1.00	0.95	0.91	-0.18	-0.03	-0.87	0.96
Consumer_Price_Index	0.93	0.93	0.95	1.00	0.98	0.04	-0.09	-0.95	0.99
Population	0.90	0.90	0.91	0.98	1.00	0.06	-0.10	-0.95	0.97
Unemployment_Rate	-0.10	-0.10	-0.18	0.04	0.06	1.00	-0.11	-0.15	0.02
Real_GDP	-0.08	-0.08	-0.03	-0.09	-0.10	-0.11	1.00	0.06	-0.09
Mortgage_Rate	-0.85	-0.85	-0.87	-0.95	-0.95	-0.15	0.06	1.00	-0.93
Real_Disposable_Income	0.95	0.95	0.96	0.99	0.97	0.02	-0.09	-0.93	1.00

In [52]:

```
pip install pandas scikit-learn
```

```
Requirement already satisfied: pandas in c:\users\akanksha\anaconda3\lib\site-package
s (1.5.3)
Requirement already satisfied: scikit-learn in c:\users\akanksha\anaconda3\lib\site-p
ackages (1.2.1)
Requirement already satisfied: numpy>=1.21.0 in c:\users\akanksha\anaconda3\lib\site-
packages (from pandas) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\akanksha\anaconda3
\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\akanksha\anaconda3\lib\site-p
ackages (from pandas) (2022.7)
Requirement already satisfied: joblib>=1.1.1 in c:\users\akanksha\anaconda3\lib\site-
packages (from scikit-learn) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\akanksha\anaconda3\lib\site-p
ackages (from scikit-learn) (1.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\akanksha\anaconda3\li
b\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\akanksha\anaconda3\lib\site-packa
ges (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [53]:

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

In [54]:

```
features = ['Stock_Price_Index', 'Consumer_Price_Index', 'Population', 'Unemployment_R
target = 'House_Price_Index'
```

In [55]:

```
X = merged_data[features]
y = merged_data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [56]:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[56]:

```
▾ LinearRegression
LinearRegression()
```

In [57]:

```
y_pred = model.predict(X_test)
```

In [58]:

```
r_squared = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
```

In [59]:

```
print(f'R-squared: {r_squared:.2f}')
print(f'Mean Squared Error (MSE): {mse:.2f}')
```

```
R-squared: 0.94
Mean Squared Error (MSE): 192.70
```

In [60]:

```
coefficients = pd.DataFrame({'Feature': features, 'Coefficient': model.coef_})
intercept = model.intercept_
```

In [61]:

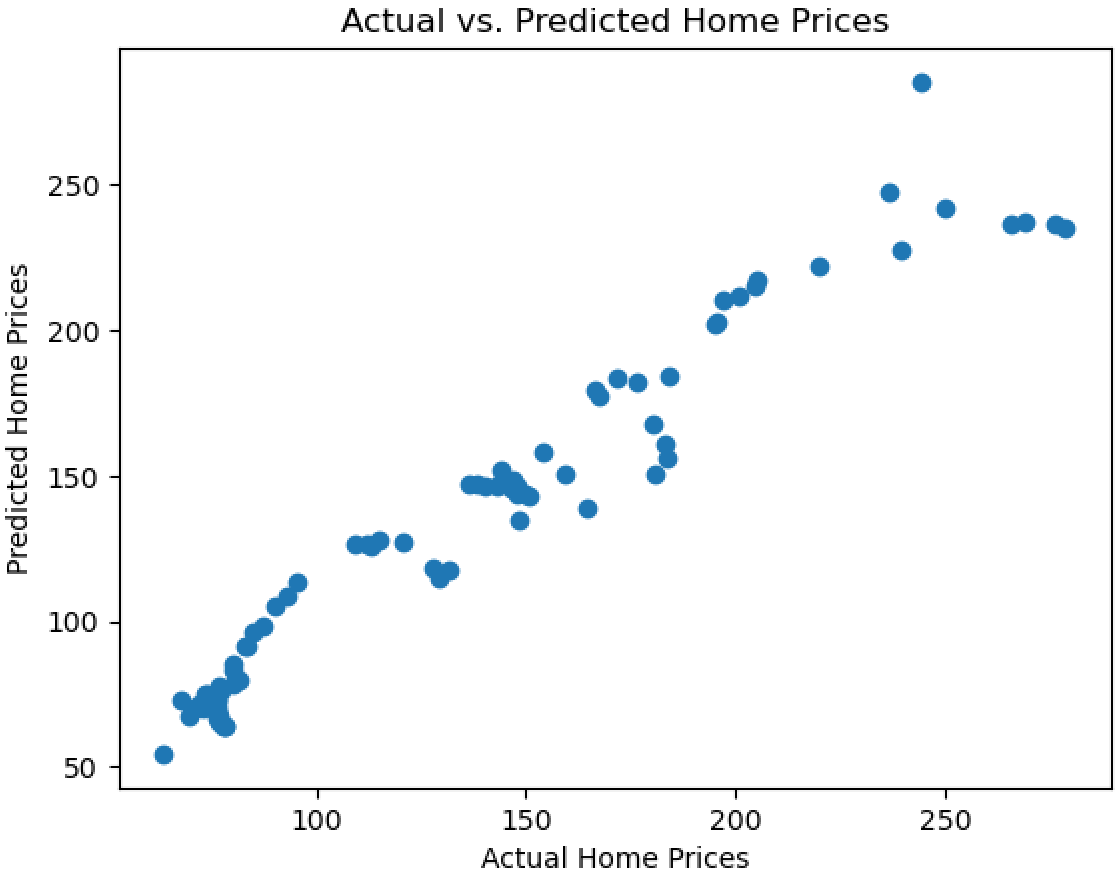
```
print('Intercept:', intercept)
print(coefficients)
```

```
Intercept: -184.3415799770794
                  Feature   Coefficient
0       Stock_Price_Index  4.802764e-01
1    Consumer_Price_Index -1.489407e-01
2              Population  5.071001e-07
3       Unemployment_Rate -4.851168e-01
4                Real_GDP  1.375784e-02
5           Mortgage_Rate  7.548667e+00
6  Real_Disposable_Income  1.528093e-02
```

In [62]:

```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.xlabel('Actual Home Prices')
plt.ylabel('Predicted Home Prices')
plt.title('Actual vs. Predicted Home Prices')
plt.show()
```


Actual vs. Predicted Home Prices
Predicted Home Prices
50
100
150
200
250
Actual Home Prices
100
150
200
250

In [ ]: