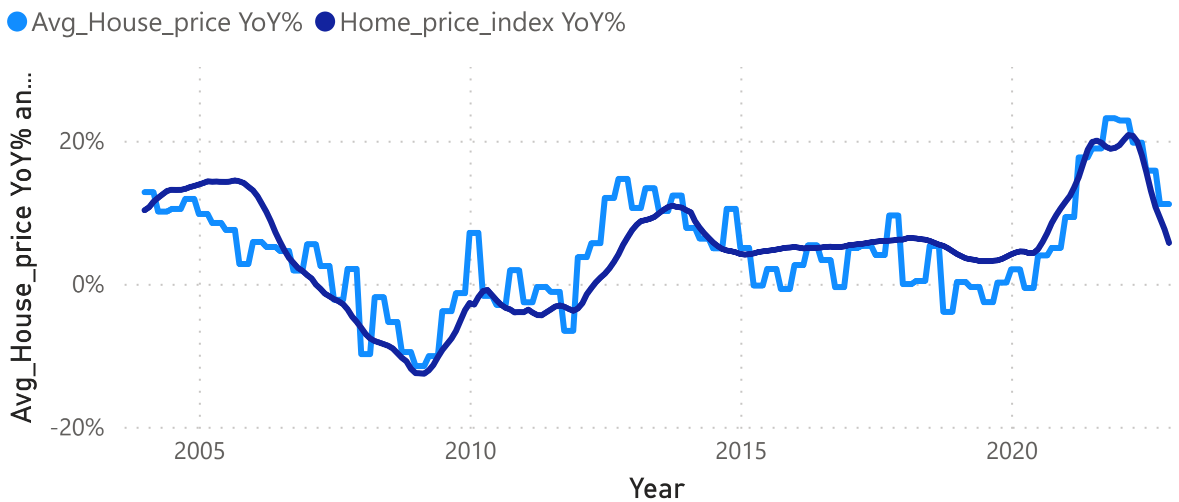
Ye CTRL+click here to follow link

2003 2022

Year-on-Year Change Line Chart

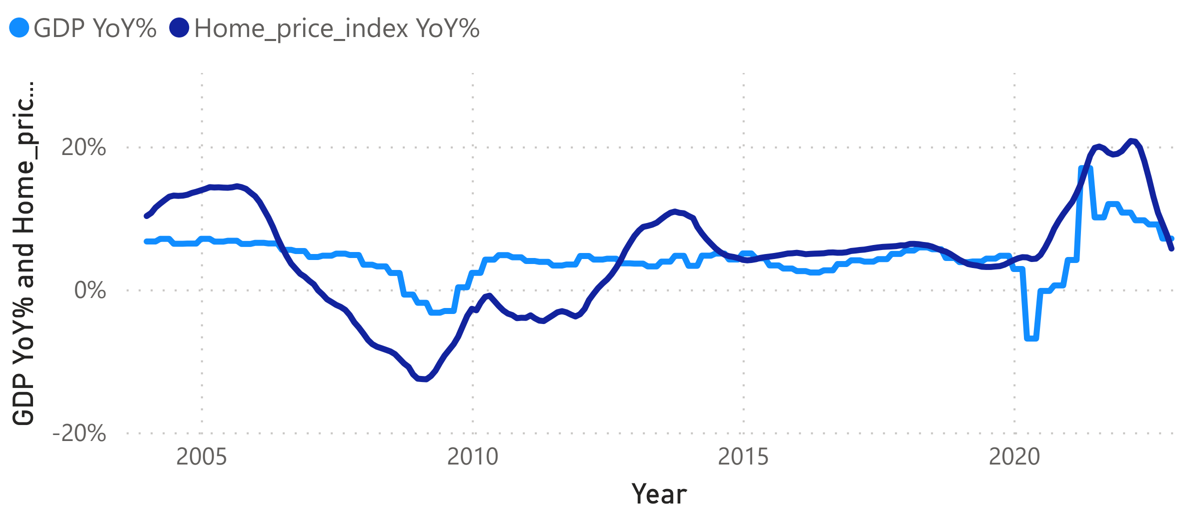
Avg_House_price YoY% and Home_price_index YoY% by Year and Month



Observations:

- There is a positive relationship between the average house price and the home price index.
- The strong correlation in yearly changes underscores the reliability of using year-on-year shifts in average house prices as a robust indicator for understanding and predicting corresponding movements in the home price index.

GDP YoY% and Home_price_index YoY% by Year and Month



Observations:

- There is a positive relationship between GDP and the house price index.
- when GDP experiences a slowdown, it tends to have a negative effect on the home price index. Reduced economic activity can lead to lower demand for housing, impacting prices and overall market dynamics

<u>**Yearly Line Chart**</u>

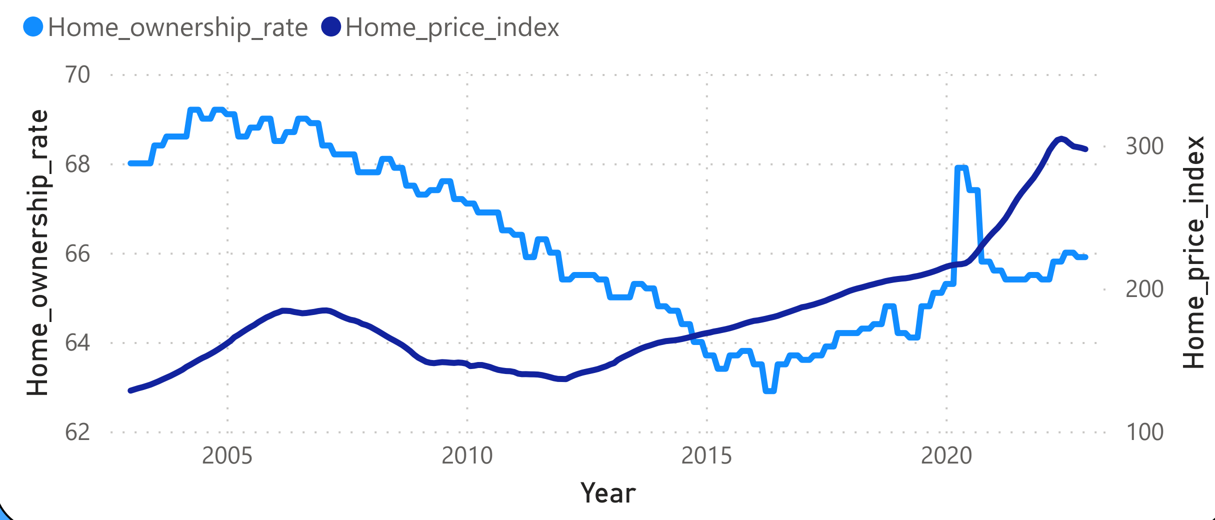
2003 2022

<u>**Year-on-Year Change Line Chart**</u>

Observations:

- Homeownership rate and home price index show a negative relationship.
- On a year-to-year basis, the correlation between homeownership rate and home price index isn't strong.
- significant shifts in homeownership rates can notably influence the home price index, often resulting in a negative impact.

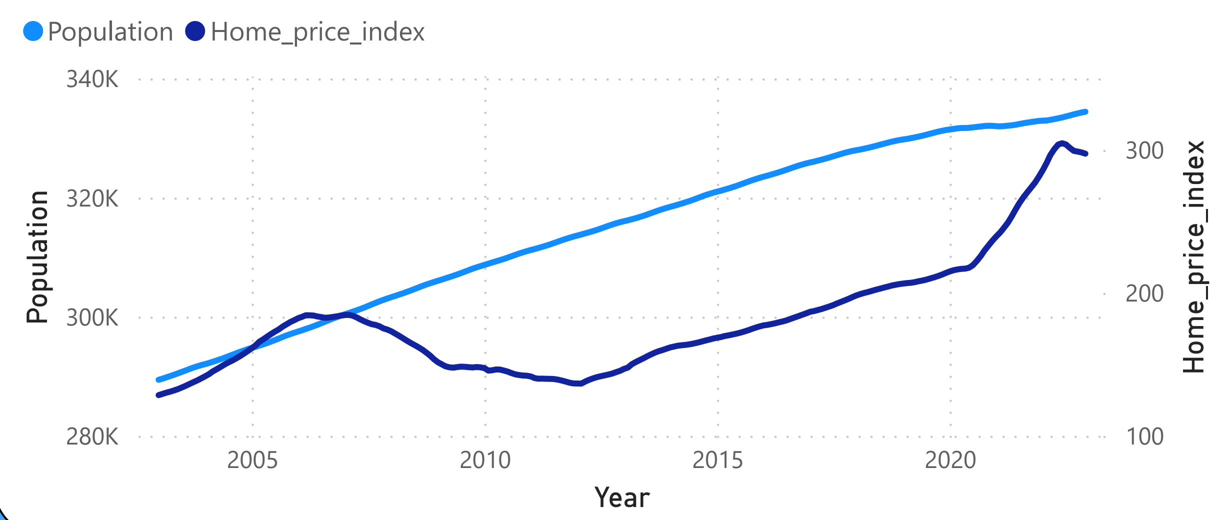
Home_ownership_rate and Home_price_index by Year and Month



Observations:

- Population and home prices demonstrate a positive relationship.
- While population growth is associated with higher home prices, it's noted that changes in population alone don't significantly impact the home price index.
- Over time, population growth may slow down, yet home prices continue to rise, suggesting other factors contribute more significantly to the upward trend in home prices.

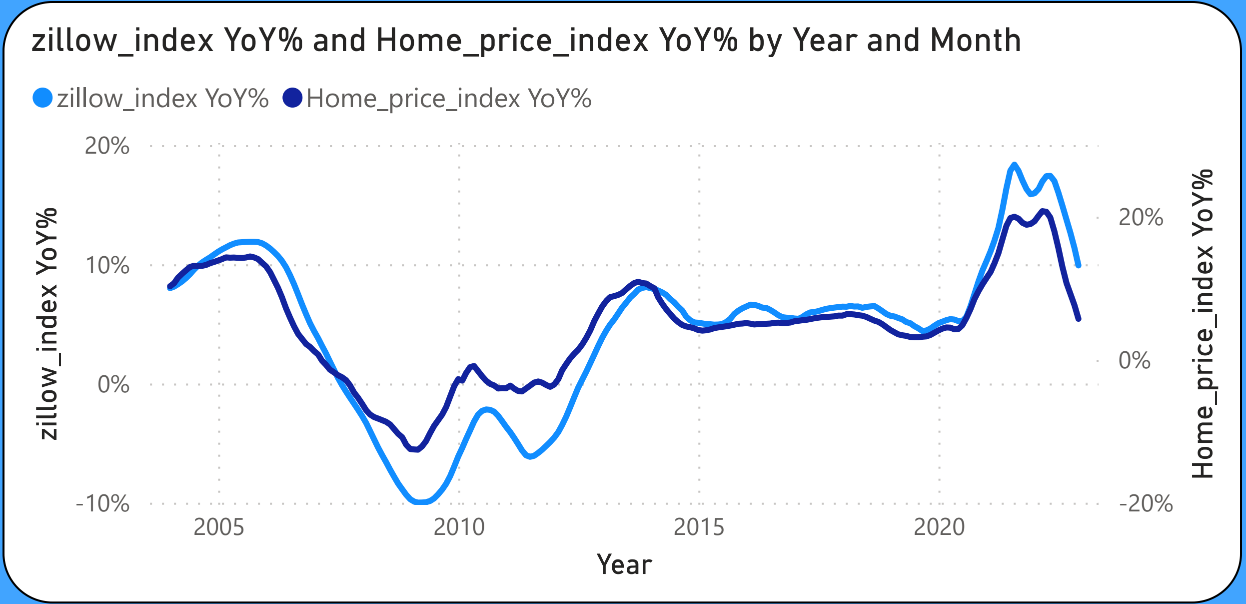
Population and Home_price_index by Year and Month



Yearly Line Chart

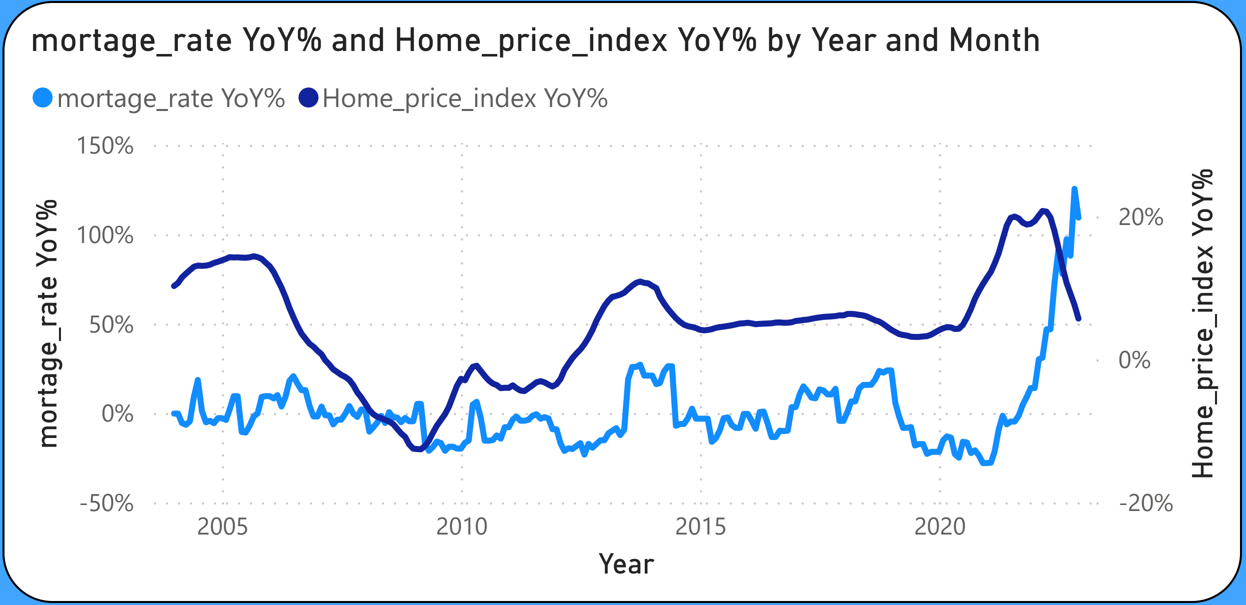
2003 2022

Year-on-Year Change Line Chart



Observations:

- The Zillow index and home price index are essentially the same.
- They highly correlate, making the Zillow index a reliable predictor of the home price index.



Observations:

- Mortgage rates and the home price index exhibit a clear negative correlation.
- in 2022, as mortgage rates increased, the home price index decreased, and inversely, in 2019 to 2020, a decrease in mortgage rates coincided with an increase in the home price index.

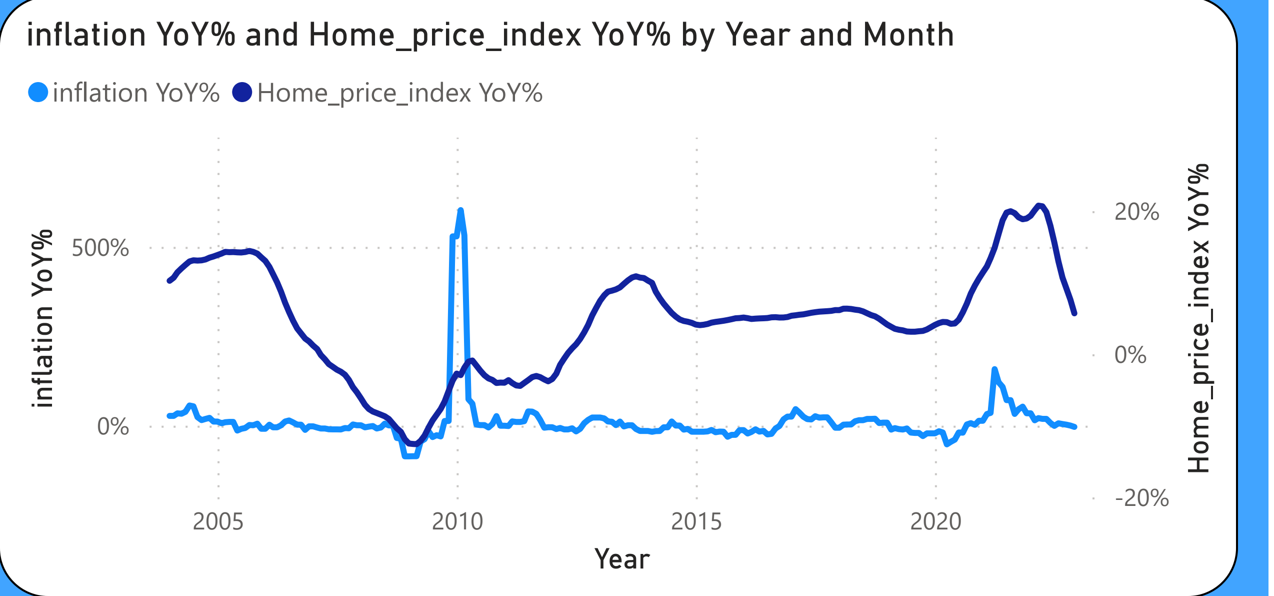
Yearly Line Chart

2003 2022

Year-on-Year Change Line Chart

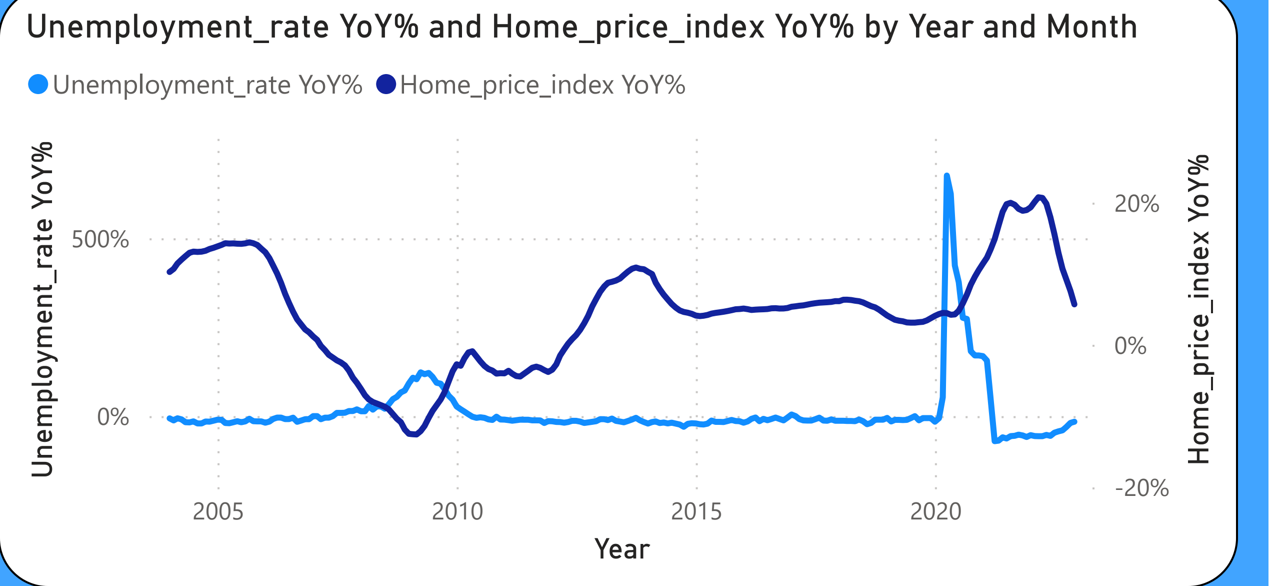
Observations:

- The relationship between inflation and the home price index is not strongly correlated, indicating a limited direct influence.
- High inflation takes time to show its effect on home prices and impacting home prices over time.
- A delay exists between sustained inflation and its effect on home prices



Observations:

- Inverse relationship between the unemployment rate and the home price index. As unemployment increases, the home price index tends to decrease, and vice versa.
- The home price index is notably sensitive to minor fluctuations in the unemployment rate

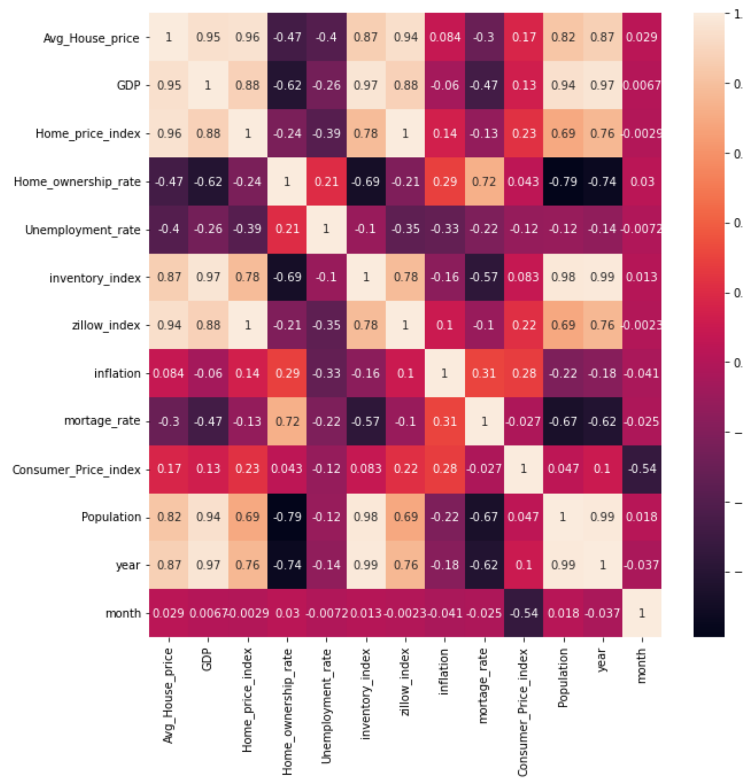


Data Frame Dimensions And Data Type Information

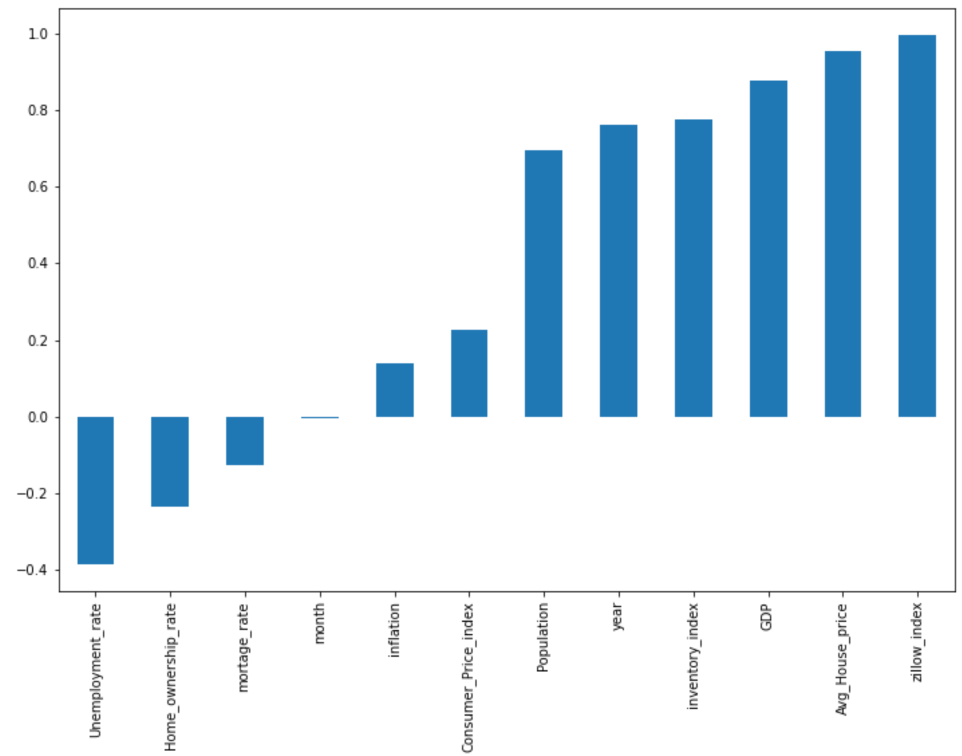
In [51]:
```
print('Dimentions',usa_df.shape)
usa_df.info()
```

```
Dimentions (246, 14)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   date                  246 non-null    datetime64[ns]
 1   Avg_House_price       246 non-null    float64
 2   GDP                   246 non-null    float64
 3   Home_price_index      246 non-null    float64
 4   Home_ownership_rate   246 non-null    float64
 5   Unemployment_rate     246 non-null    float64
 6   inventory_index       246 non-null    float64
 7   zillow_index          246 non-null    float64
 8   inflation             246 non-null    float64
 9   mortage_rate          246 non-null    float64
 10  Consumer_Price_index  246 non-null    float64
 11  Population            246 non-null    float64
 12  year                  246 non-null    int64
 13  month                 246 non-null    int64
dtypes: datetime64[ns](1), float64(11), int64(2)
memory usage: 27.0 KB
```

In [27]:
```
plt.figure(figsize=(10,10))
sns.heatmap(usa_df.corr(),annot=True)
plt.show()
```



In [28]:
```
plt.figure(figsize=(12,8))
usa_df.corr()["Home_price_index"].sort_values().drop(["Home_price_index"]).plot(kind = "bar")
plt.show()
```



	vif	Features
10	5582.905071	year
9	3664.023055	Population
1	951.899838	GDP
4	330.537769	inventory_index
5	91.096169	zillow_index
0	42.709297	Avg_House_price
2	21.347750	Home_ownership_rate
11	16.590669	month
3	11.829162	Unemployment_rate
7	8.043525	mortage_rate
6	1.994516	inflation
8	1.812942	Consumer_Price_index

Observation:

- features like Population, GDP, and zillow_index with strong positive correlations to Home_price_index.
- Unemployment_rate, Home_ownership_rate, and mortgage_rate for their negative correlations with Home_price_index.
- inflation and Consumer_Price_index for subtle positive correlations with Home_price_index.
- year and Population have the highest VIF values, indicating they might have strong multicollinearity with other variables in the model.
- GDP and inventory_index also have high VIF values, suggesting they might be correlated with other predictors.
- Avg_House_price, Home_ownership_rate, Unemployment_rate, and mortage_rate have moderate VIF values.
- inflation and Consumer_Price_index have the lowest VIF values, indicating they have the least multicollinearity with other variables.

Code for Model building

```python
results = []
# Loop through each model
for model_name, model in regression_models.items():
    # Cross-validation scores
    cv_scores = cross_val_score(model, X_train, y_train, cv=5)

    # Fit the model
    model.fit(X_train, y_train)

    # Make predictions for train data
    y_pred_train = model.predict(X_train)


    # Evaluate the model
    mae_train = mean_absolute_error(y_train,y_pred_train)
    mse_train = mean_squared_error(y_train, y_pred_train)
    r2_train = r2_score(y_train, y_pred_train)

    # Make predictions for test data
    y_pred_test = model.predict(X_test)

    # Evaluate the model
    mae_test = mean_absolute_error(y_test,y_pred_test)
    mse_test = mean_squared_error(y_test, y_pred_test)
    r2_test = r2_score(y_test, y_pred_test)

    results.append({
        "Model": model_name,
        "MSE Train": mse_train,
        "R2 Score Train": r2_train,
        "MAE Train": mae_train,
        "MSE Test": mse_test,
        "R2 Score Test": r2_test,
        "MAE Test": mae_test,
        "CV_Score": np.mean(cv_scores)
    })

# Display the results
results_df = pd.DataFrame(results)
print(results_df)
```

Model Used

In [33]:
```python
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
```

Metrics Score Comparison

In [36]:
```python
results_df
```

Out[36]:

	Model	MSE Train	R2 Score Train	MAE Train	MSE Test	R2 Score Test	MAE Test	CV_Score
0	Linear Regression	5.128700	0.997523	1.716588	5.909237	0.996322	1.879435	0.997005
1	Ridge Regression	6.203073	0.997004	1.860886	7.714368	0.995198	2.071351	0.996324
2	Lasso Regression	10.070173	0.995136	2.443181	7.947971	0.995053	2.272569	0.994648
3	Elastic Net	90.249051	0.956412	7.794727	92.262246	0.942571	7.800826	0.954221
4	Decision Tree Regressor	0.000000	1.000000	0.000000	4.558036	0.997163	1.612000	0.995332
5	Random Forest Regressor	0.414679	0.999800	0.476993	1.930133	0.998799	1.076937	0.997476
6	KNN Model	18.215202	0.991202	3.380989	19.647883	0.987770	3.438648	0.977243

Hyperparameter tuning

```
from sklearn.model_selection import GridSearchCV

# Defining the parameter grid
param_grid = {
    'n_estimators': [100, 200, 250, 300],
    'max_depth': [5, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]}

# Instantiate the grid search model
grid_search = GridSearchCV(estimator=random_forest_regressor_model, param_grid=param_grid,
                           cv=3, n_jobs=-1, verbose=2)

# Fit the grid search to the data
grid_search.fit(X_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_

print(f"Best parameters: {best_params}")
```

```
Fitting 3 folds for each of 288 candidates, totalling 864 fits
Best parameters: {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
```

Final Model Score

```
In [42]: results

Out[42]: {'Model': 'Final model',
          'MSE Train': 0.4599491097765046,
          'R2 Score Train': 0.9997778539995211,
          'MAE Train': 0.5008132653060944,
          'MSE Test': 19.64788289759997,
          'R2 Score Test': 0.9877699942734672,
          'MAE Test': 3.438647999999995}
```

Final Model Link

Source:

Average house Price: https://fred.stlouisfed.org/series/USSTHPI
Consumer Price index: https://fred.stlouisfed.org/series/STICKCPIM157SFRBATL
GDP: https://fred.stlouisfed.org/series/GDP
Home Ownership Rate: https://fred.stlouisfed.org/series/RSAHORUSQ156S
Home Price Index:https://fred.stlouisfed.org/series/CSUSHPISA
Inflation Breakeven 5 yrs: https://fred.stlouisfed.org/series/T5YIE
Inventory index: https://fred.stlouisfed.org/series/ETOTALUSQ176N
Mortgage Rate: https://fred.stlouisfed.org/series/MORTGAGE30US
Population: https://fred.stlouisfed.org/series/POPTHM
Unemployment Rate: https://fred.stlouisfed.org/series/UNRATE
Zillow home price index: https://fred.stlouisfed.org/series/VAUCSFRCONDOSMSAMID