```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Set random seed for reproducibility
np.random.seed(42)

# Number of samples
num_samples = 500

# Generate random synthetic features
CRIM = np.random.rand(num_samples) * 10  # Crime rate
RM = np.random.randint(4, 9, num_samples)  # Number of rooms
AGE = np.random.randint(10, 100, num_samples)  # Age of the house
TAX = np.random.randint(200, 600, num_samples)  # Property tax
DIS = np.random.rand(num_samples) * 12  # Distance to employment centers

# Generate house prices using a simple linear function with noise
MEDV = 50000 + (CRIM * -2000) + (RM * 15000) + (AGE * -100) + (TAX * -50) + (DIS * -3000) + np.random.randn(num_samples) * 10000

# Create a DataFrame
df = pd.DataFrame({'CRIM': CRIM, 'RM': RM, 'AGE': AGE, 'TAX': TAX, 'DIS': DIS, 'MEDV': MEDV})

# Display first few rows
df.head()
```

	CRIM	RM	AGE	TAX	DIS	MEDV
0	3.745401	4	20	271	4.851825	82787.609529
1	9.507143	4	11	476	5.138395	55724.768667
2	7.319939	4	76	473	8.611173	51371.330564
3	5.986585	8	96	519	8.309234	91867.705570
4	1.560186	7	21	537	11.895072	68936.726934

```
# Define independent (X) and dependent (y) variables
X = df.drop(columns=['MEDV'])  # Features
y = df['MEDV']  # Target variable (House Prices)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Initialize the model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Predict house prices
y_pred = model.predict(X_test)
```

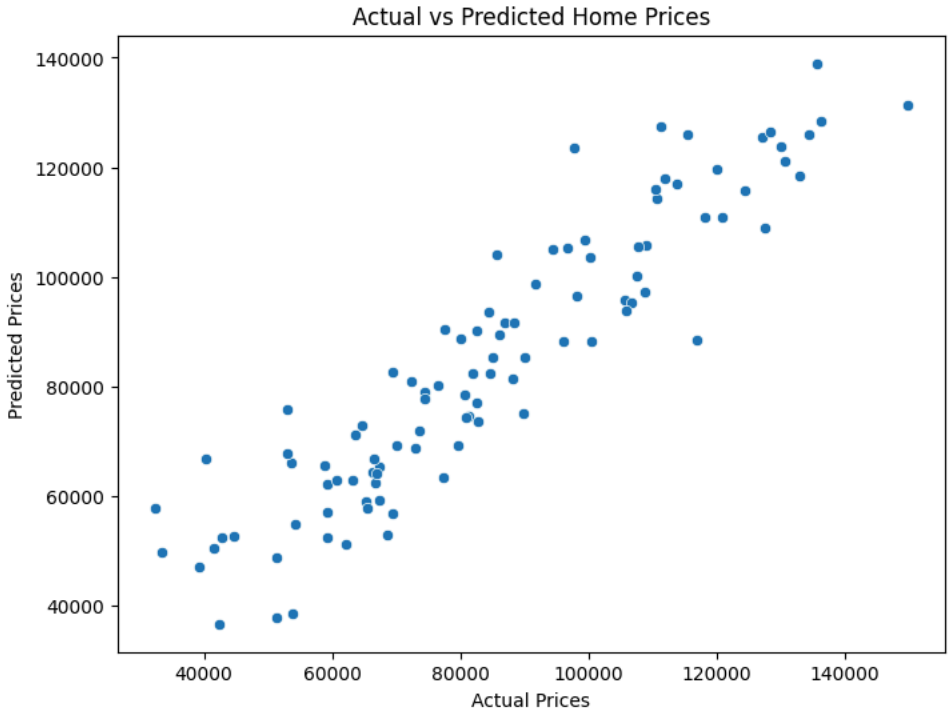
```
# Calculate errors
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5  # Square root of MSE
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"R² Score: {r2}")
```

```
Mean Squared Error (MSE): 105236437.24476792
Root Mean Squared Error (RMSE): 10258.481234801178
```

```
R² Score: 0.8557422784317762
```

```
# Scatter plot of actual vs predicted values
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Home Prices")
plt.show()
```




```
df.to_csv("synthetic_boston_housing.csv", index=False)
print("Dataset saved as synthetic_boston_housing.csv")
```

```
Dataset saved as synthetic_boston_housing.csv
```