In [7]:

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

In [29]:

```
data = pd.read_csv('Desktop/train456.csv')
```

In [30]:

```
data.head()
```

Out[30]:

| | Id | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 60 | RL | 65.0 | 8450 | Pave | NaN | Reg | Lvl | AllPub | ... |
| **1** | 2 | 20 | RL | 80.0 | 9600 | Pave | NaN | Reg | Lvl | AllPub | ... |
| **2** | 3 | 60 | RL | 68.0 | 11250 | Pave | NaN | IR1 | Lvl | AllPub | ... |
| **3** | 4 | 70 | RL | 60.0 | 9550 | Pave | NaN | IR1 | Lvl | AllPub | ... |
| **4** | 5 | 60 | RL | 84.0 | 14260 | Pave | NaN | IR1 | Lvl | AllPub | ... |

5 rows × 81 columns

In [31]:

```
features = data[['GrLivArea', 'BedroomAbvGr', 'FullBath']]
target = data['SalePrice']
```

In [32]:

```
X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.2, random_state=42)
```

In [33]:

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

Out[33]:

```
LinearRegression
LinearRegression()
```

In [34]:

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

In [35]:

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

In [36]:

```
r2 = r2_score(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')
print(f'R² Score: {r2:.2f}')
Mean Squared Error: 2806426667.25
R² Score: 0.63
```

In [37]:

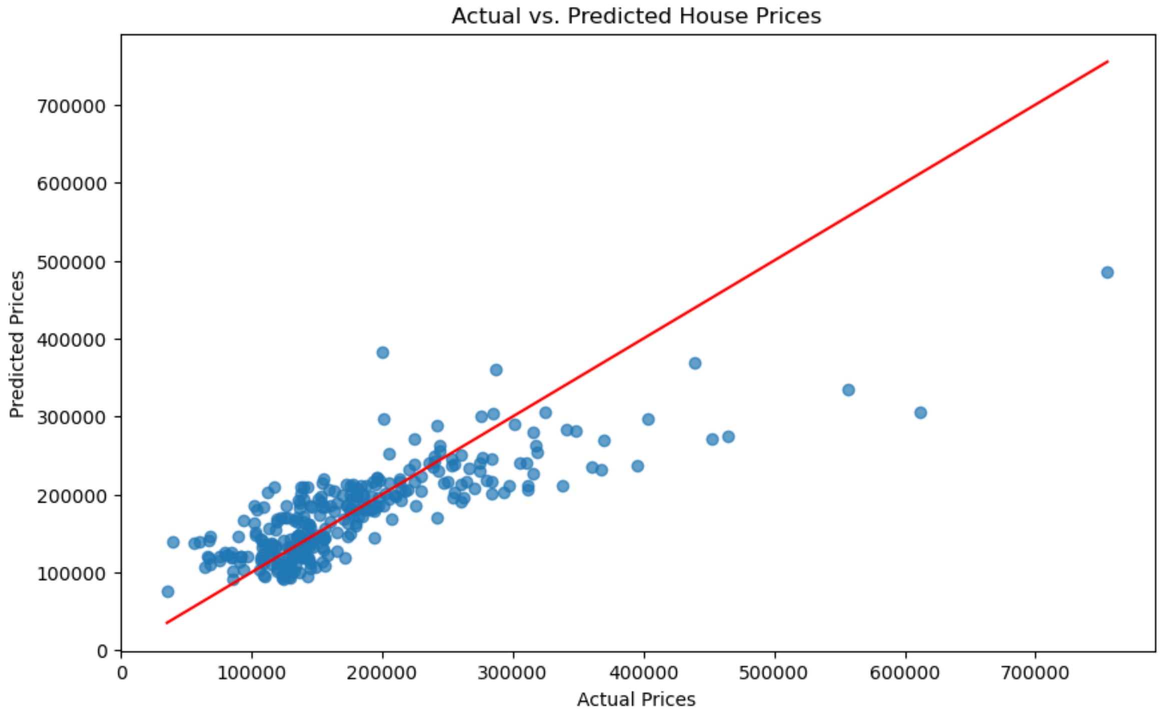
```
#example
```

```
new_house = np.array([[2000, 3, 2]])  # 2000 square feet, 3 bedrooms, 2
bathrooms
predicted_price = model.predict(new_house)

print(f'Predicted Price: ${predicted_price[0]:,.2f}')
Predicted Price: $240,377.51
/opt/anaconda3/lib/python3.11/site-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but LinearRegression was
fitted with feature names
  warnings.warn(
```

In [38]:

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.7)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs. Predicted House Prices')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)],
color='red')
plt.show()
```



In [ ]:

In [ ]: