In [26]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression

# Read the CSV file
my_data = pd.read_csv("C:/Users/user/Desktop/My learning/ClinSoft/Real est
```

In [27]:
```
my_data
```

Out[27]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1
...	...	...	...	...	...	...	...	...
409	410	2013.000	13.7	4082.01500	0	24.94155	121.50381	15.4
410	411	2012.667	5.6	90.45606	9	24.97433	121.54310	50.0
411	412	2013.250	18.8	390.96960	7	24.97923	121.53986	40.6
412	413	2013.000	8.1	104.81010	5	24.96674	121.54067	52.5
413	414	2013.500	6.5	90.45606	9	24.97433	121.54310	63.9

414 rows × 8 columns

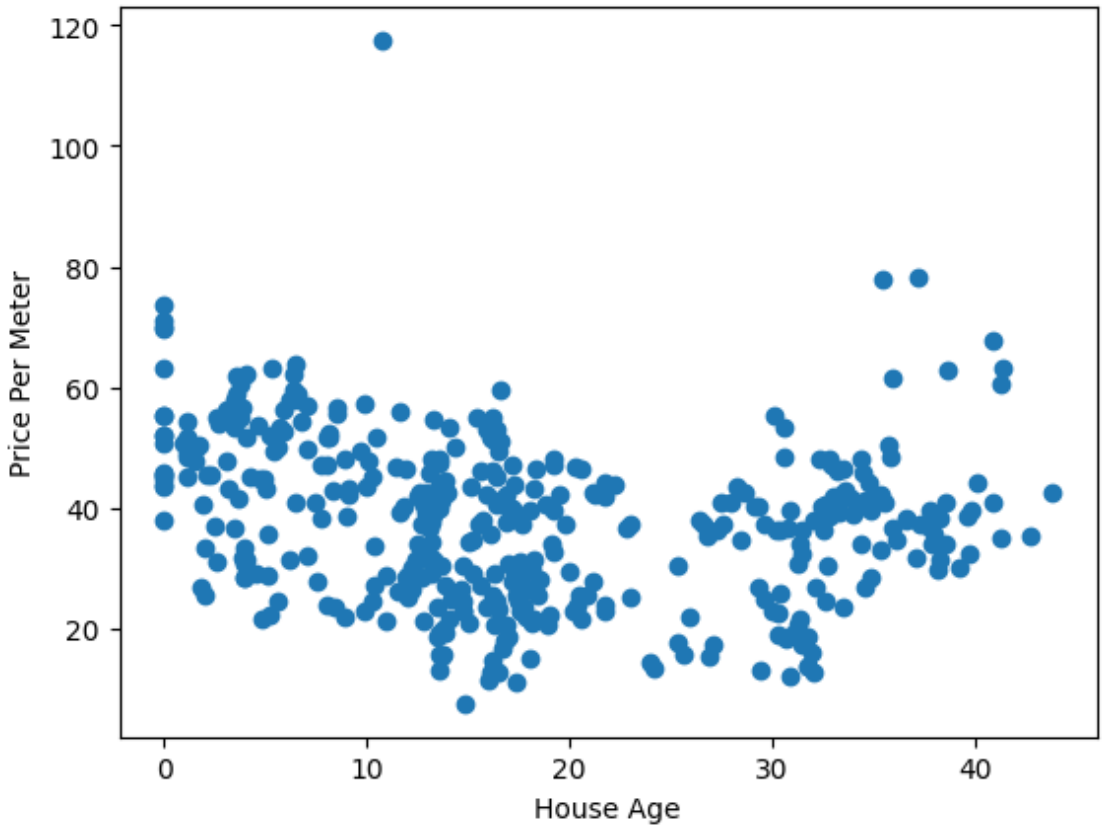
In [28]:

```
# Rename columns
my_data.columns = ["ID", "Deal_Date", "House_Age", "Station_Distance", "Ne

# Display column names
print(my_data.columns)

# Display summary of the dataset
print(my_data.describe())
```

```
Index(['ID', 'Deal_Date', 'House_Age', 'Station_Distance', 'Nearby_Store
s',
       'Latitude', 'Longitude', 'Price_Per_Meter'],
      dtype='object')
                ID    Deal_Date   House_Age  Station_Distance  Nearby_Stor
es  \
count  414.000000   414.000000  414.000000        414.000000     414.0000
00
mean   207.500000  2013.148971   17.712560       1083.885689       4.0942
03
std    119.655756     0.281967   11.392485       1262.109595       2.9455
62
min      1.000000  2012.667000    0.000000         23.382840       0.0000
00
25%    104.250000  2012.917000    9.025000        289.324800       1.0000
00
50%    207.500000  2013.167000   16.100000        492.231300       4.0000
00
75%    310.750000  2013.417000   28.150000       1454.279000       6.0000
00
max    414.000000  2013.583000   43.800000       6488.021000      10.0000
00

         Latitude   Longitude  Price_Per_Meter
count  414.000000  414.000000       414.000000
mean    24.969030  121.533361        37.980193
std      0.012410    0.015347        13.606488
min     24.932070  121.473530         7.600000
25%     24.963000  121.528085        27.700000
50%     24.971100  121.538630        38.450000
75%     24.977455  121.543305        46.600000
max     25.014590  121.566270       117.500000
```

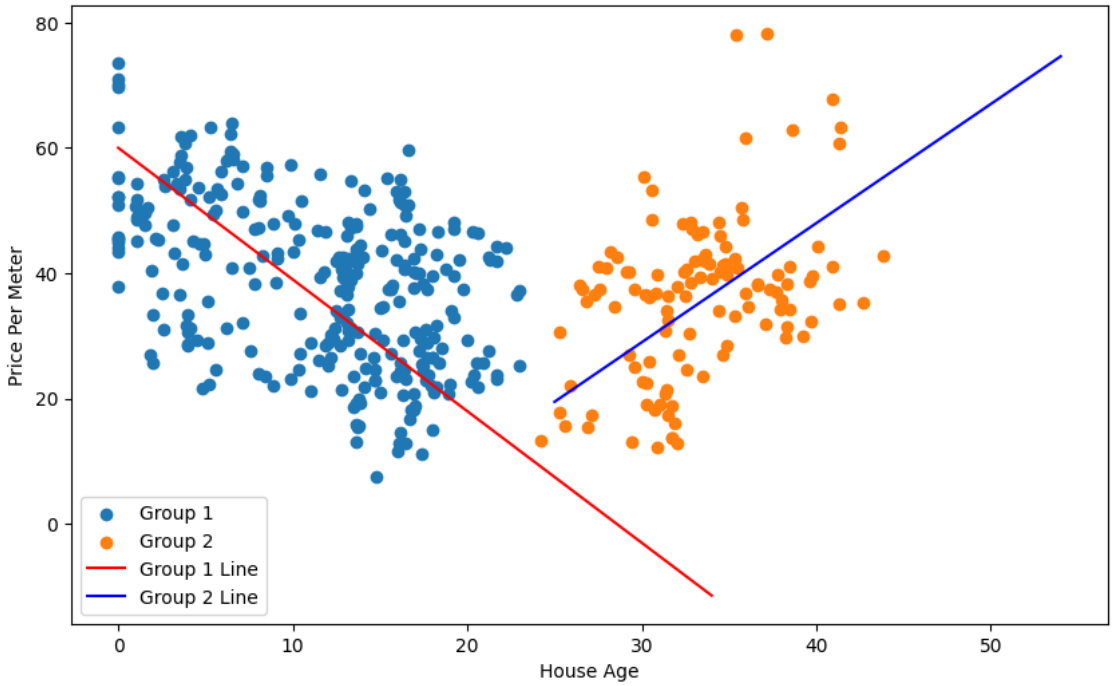
In [29]:

```
# Scatter plot
plt.scatter(my_data["House_Age"], my_data["Price_Per_Meter"])
plt.xlabel("House Age")
plt.ylabel("Price Per Meter")
plt.show()
```



In [30]:

```
# Filtering and plotting
age_price_filter1 = (my_data["House_Age"] < 24) & (my_data["Price_Per_Mete
age_price_filter2 = (my_data["House_Age"] > 24) & (my_data["Price_Per_Mete

plt.figure(figsize=(10, 6))
plt.scatter(my_data.loc[age_price_filter1, "House_Age"], my_data.loc[age_p
plt.scatter(my_data.loc[age_price_filter2, "House_Age"], my_data.loc[age_p
plt.plot(np.arange(0, 35), 60 - 2.1 * np.arange(0, 35), label="Group 1 Lin
plt.plot(np.arange(25, 55), -28 + 1.9 * np.arange(25, 55), label="Group 2
plt.xlabel("House Age")
plt.ylabel("Price Per Meter")
plt.legend()
plt.show()
```



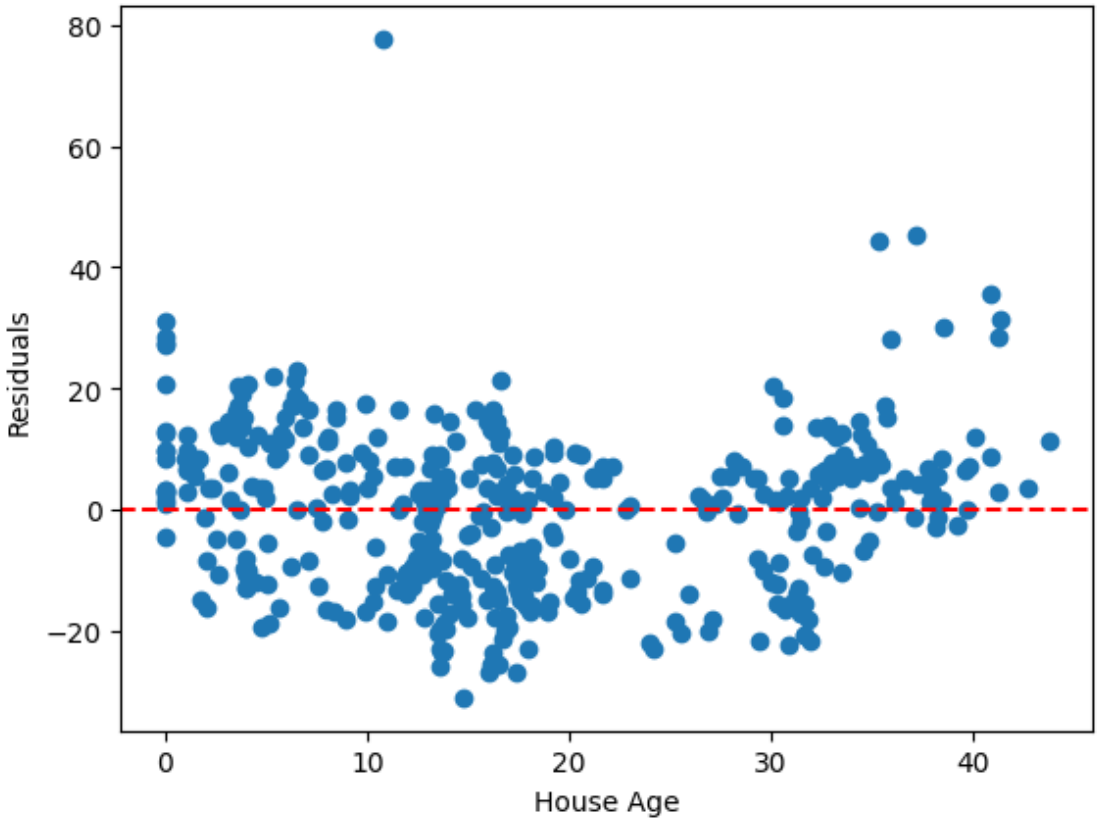
In [31]:

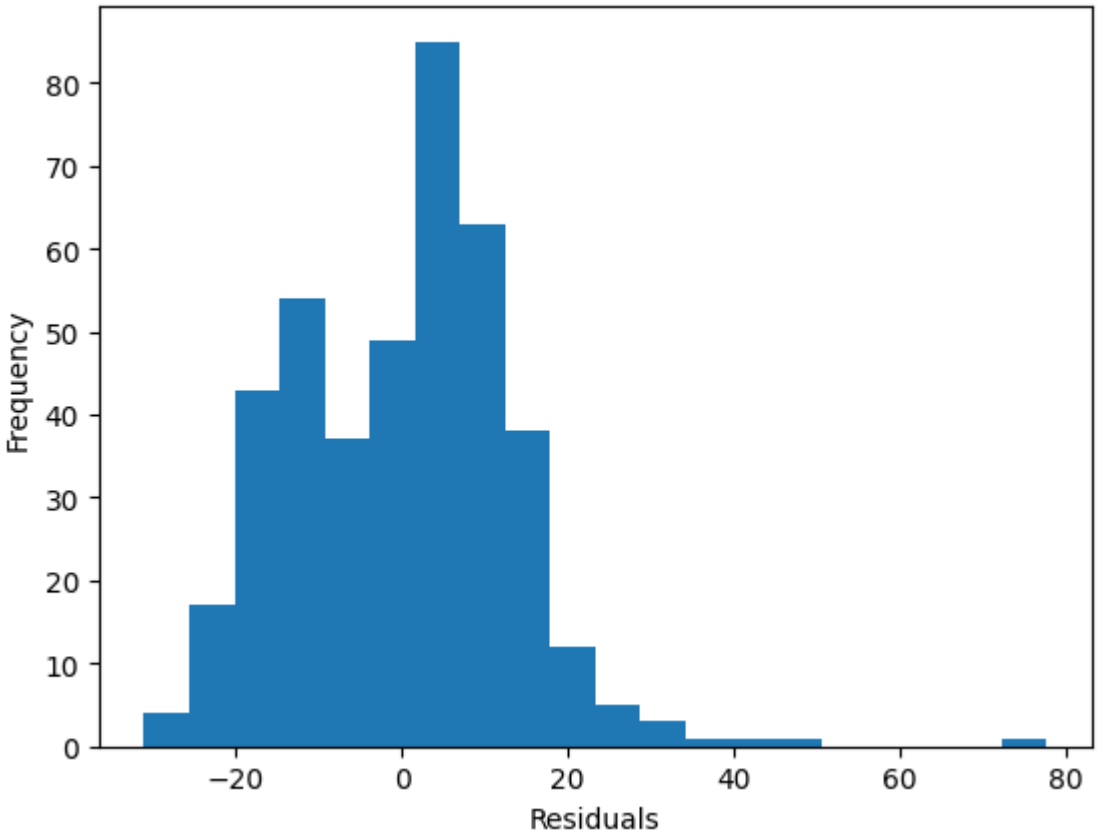
```
# Create a copy of the data
my_data2 = my_data.copy()

# Fit linear regression model
model1 = LinearRegression()
X = my_data2[["House_Age"]]
y = my_data2["Price_Per_Meter"]
model1.fit(X, y)

# Model summary
print("Intercept:", model1.intercept_)
print("Coefficient:", model1.coef_)
```

```
Intercept: 42.4346970462629
Coefficient: [-0.25148842]
```

In [47]:

```
residuals = y - model1.predict(X)

# Plot model residuals
plt.scatter(X, residuals)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel("House Age")
plt.ylabel("Residuals")
plt.show()

# Histogram of residuals
plt.hist(residuals, bins=20)
plt.xlabel("Residuals")
plt.ylabel("Frequency")
plt.show()
```





In [ ]:

```
# Predictions and visualization
my_data2["pred"] = model1.predict(X)
plt.scatter(my_data2["House_Age"], my_data2["Price_Per_Meter"], color='b')
plt.scatter(my_data2["House_Age"], my_data2["pred"], color='orangered')
plt.xlabel("House Age")
plt.ylabel("Price Per Meter")
plt.show()
```

In [33]:

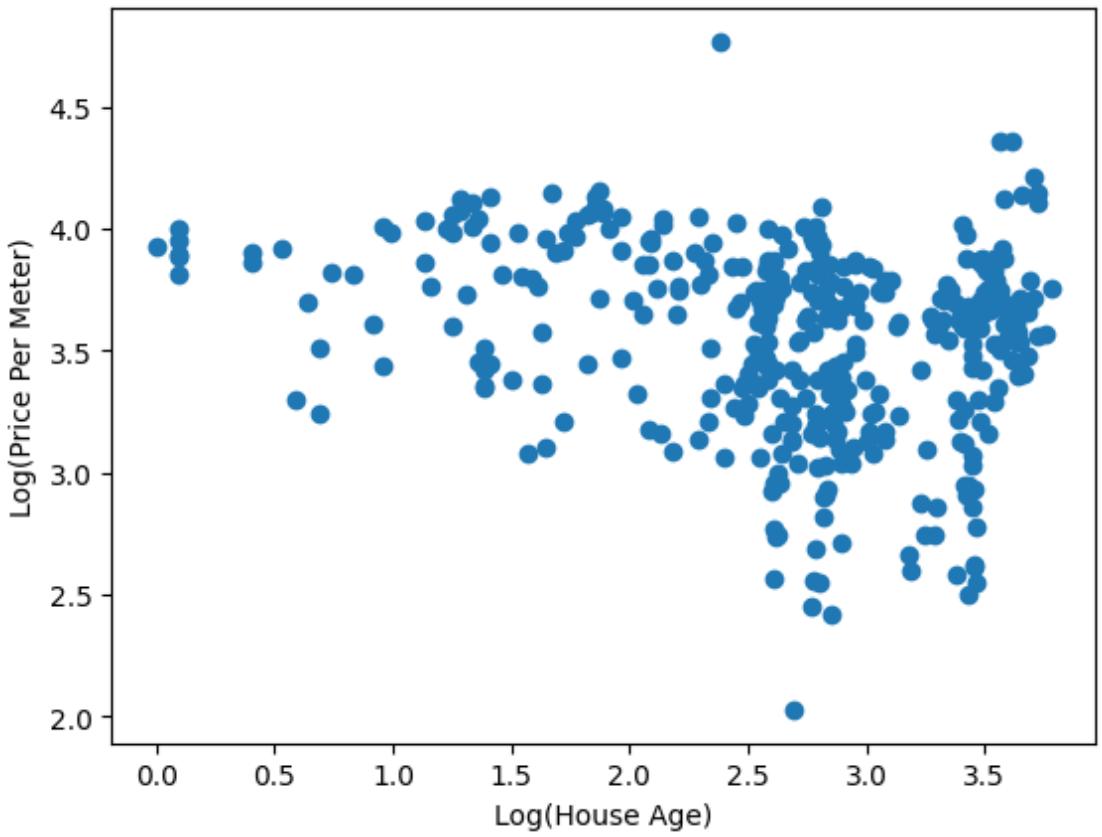
```
# Fit linear regression models with different subsets of data
model2 = LinearRegression()
X2 = my_data2.loc[30:200, ["Price_Per_Meter"]]
y2 = my_data2.loc[30:200, "House_Age"]
model2.fit(X2, y2)

# Summary of models
print("Model 1:")
print("Intercept:", model1.intercept_)
print("Coefficient:", model1.coef_)
print("Model 2:")
print("Intercept:", model2.intercept_)
print("Coefficient:", model2.coef_)
```

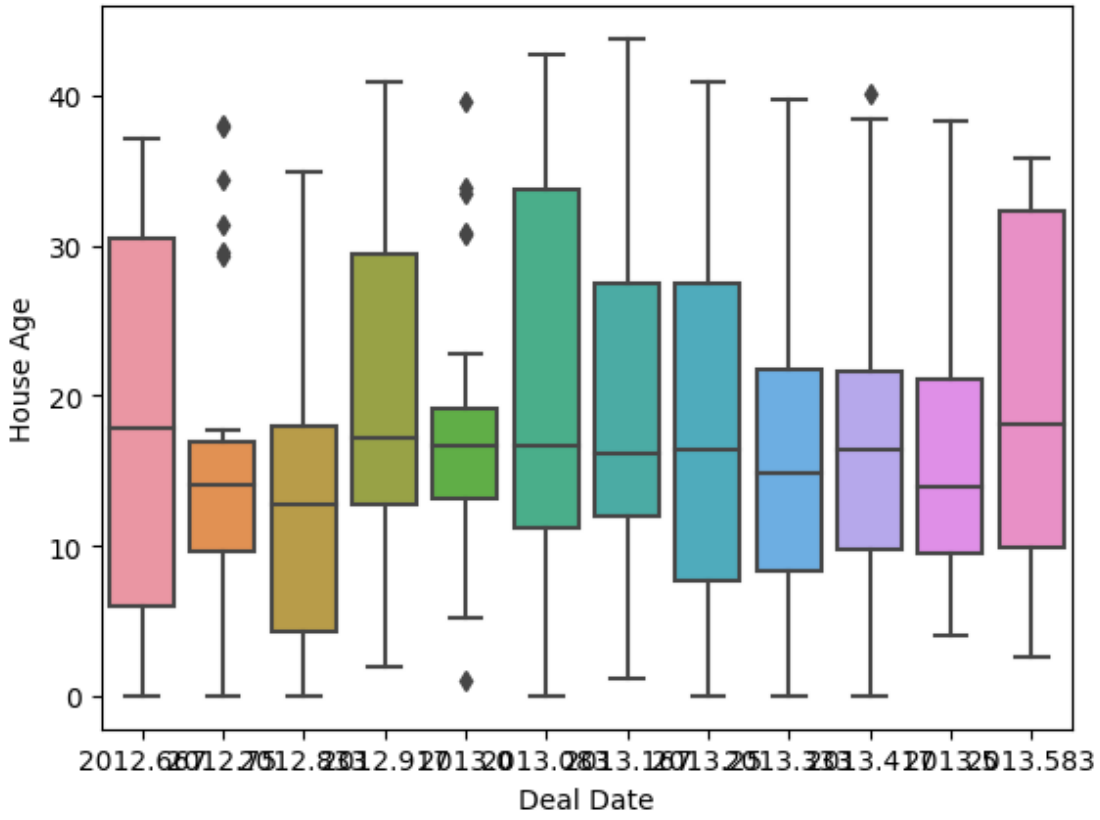
```
Model 1:
Intercept: 42.4346970462629
Coefficient: [-0.25148842]
Model 2:
Intercept: 26.68014364873038
Coefficient: [-0.22659476]
```

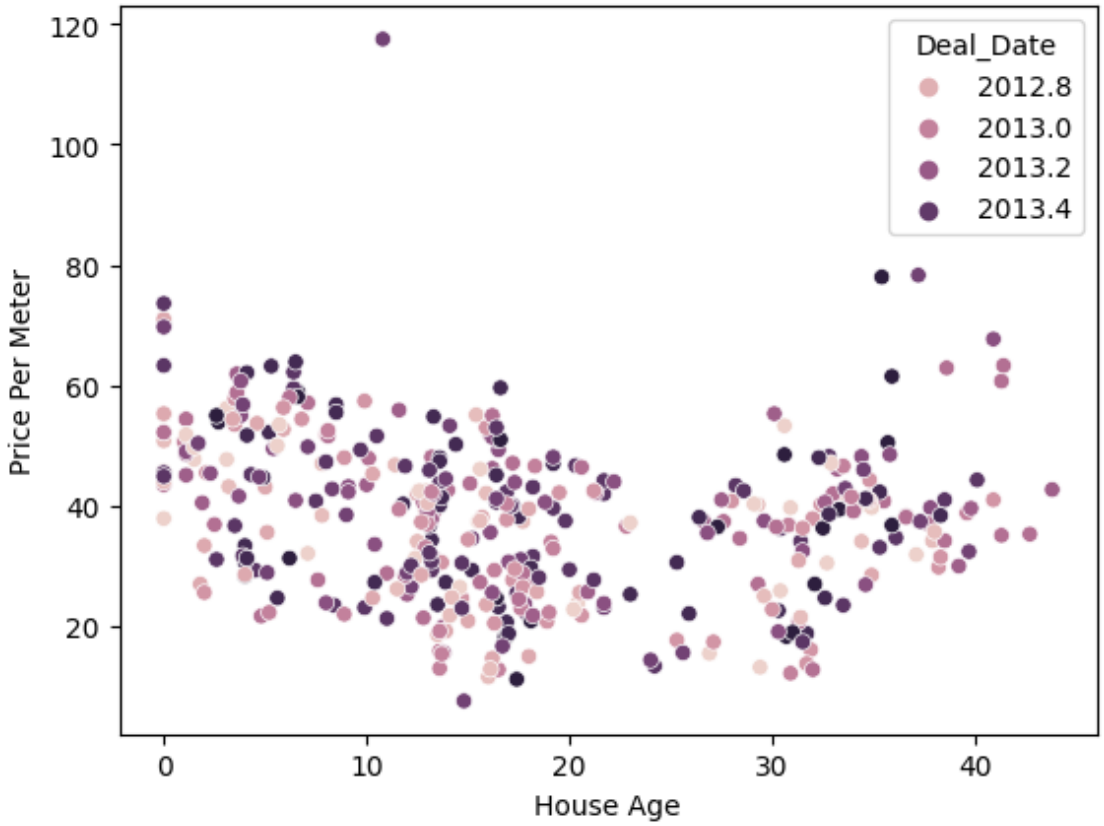
In [34]:

```
# Log-transformed scatter plot
plt.scatter(np.log(my_data2["House_Age"]), np.log(my_data2["Price_Per_Mete
plt.xlabel("Log(House Age)")
plt.ylabel("Log(Price Per Meter)")
plt.show()

# Box plot
sns.boxplot(x="Deal_Date", y="House_Age", data=my_data2)
plt.xlabel("Deal Date")
plt.ylabel("House Age")
plt.show()

# Scatter plot with color
sns.scatterplot(x="House_Age", y="Price_Per_Meter", hue="Deal_Date", data=
plt.xlabel("House Age")
plt.ylabel("Price Per Meter")
plt.show()
```

```
C:\Users\user\Anaconda3\lib\site-packages\pandas\core\arraylike.py:396: R
untimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```




House Age
Deal Date
0
10
20
30
40


Price Per Meter
House Age
Deal_Date
2012.8
2013.0
2013.2
2013.4
20
40
60
80
100
120
0
10
20
30
40

In [35]:

```
# Categorize data
my_data3 = my_data.copy()
my_data3["old_bin"] = np.where(my_data3["House_Age"] > 24, 1, 0)

# Linear regression with additional binary variable
model_1 = LinearRegression()
X3 = my_data3[["House_Age", "old_bin"]]
y3 = my_data3["Price_Per_Meter"]
model_1.fit(X3, y3)

# Model summary
print("Intercept:", model_1.intercept_)
print("Coefficients:", model_1.coef_)
```

```
Intercept: 46.305427206206396
Coefficients: [-0.66389086 12.04800713]
```

In [41]:

```
df = my_data.copy()
df
```

Out[41]:

	ID	Deal_Date	House_Age	Station_Distance	Nearby_Stores	Latitude	Longitude	Pr
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	
...	...	...	...	...	...	...	...	
409	410	2013.000	13.7	4082.01500	0	24.94155	121.50381	
410	411	2012.667	5.6	90.45606	9	24.97433	121.54310	
411	412	2013.250	18.8	390.96960	7	24.97923	121.53986	
412	413	2013.000	8.1	104.81010	5	24.96674	121.54067	
413	414	2013.500	6.5	90.45606	9	24.97433	121.54310	

414 rows × 11 columns

In [44]:

```
# Create new features in the DataFrame
df["old_bin"] = np.where(df["House_Age"] > 24, 1, 0)
df["age_new"] = np.where(df["House_Age"] > 24, 0, df["House_Age"])
df["age_old"] = np.where(df["House_Age"] > 24, df["House_Age"], 0)
```

In [48]:

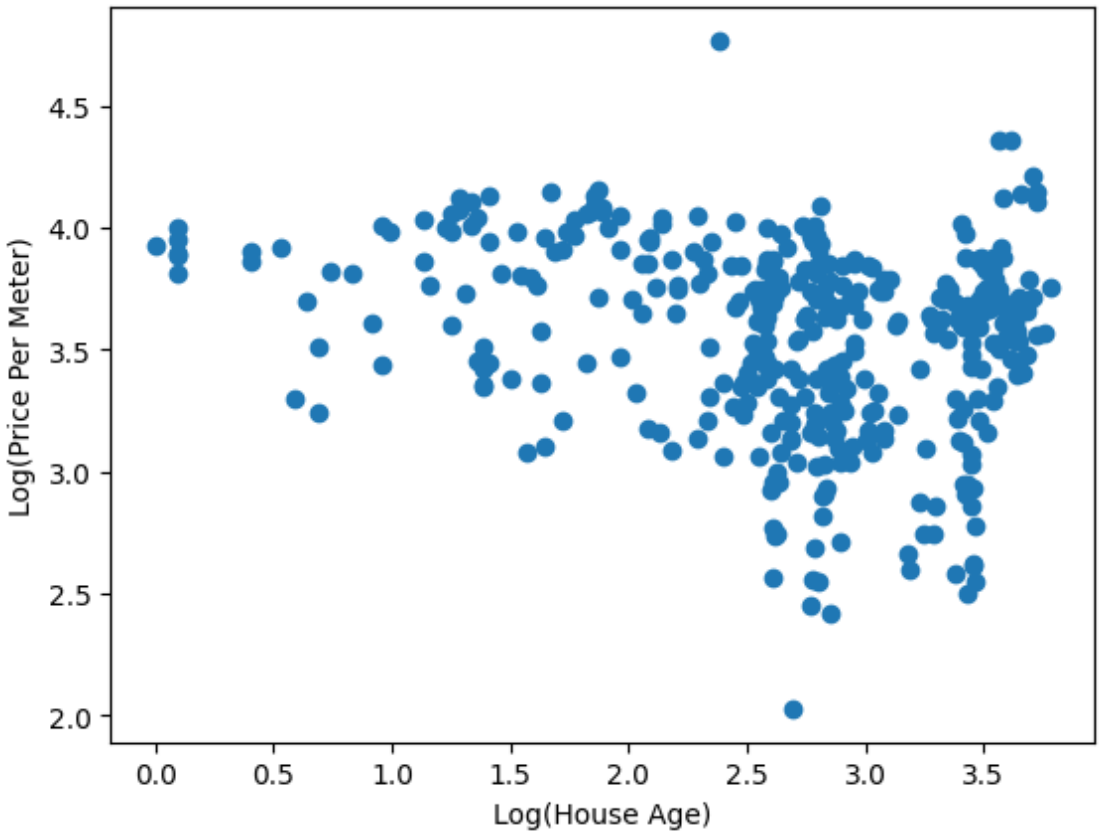
```
# Remove row 271
df_n = df.drop(index=271)

# Fit linear regression model
model_1 = LinearRegression()
X = df_n[['age_new', 'old_bin', 'age_old']]
y = df_n['Price_Per_Meter']
model_1.fit(X, y)

# Model summary
print("Intercept:", model_1.intercept_)
print("Coefficients:", model_1.coef_)
```

```
Intercept: 50.308070330098744
Coefficients: [ -1.00978668 -56.82095753   1.28628008]
```

In [49]:

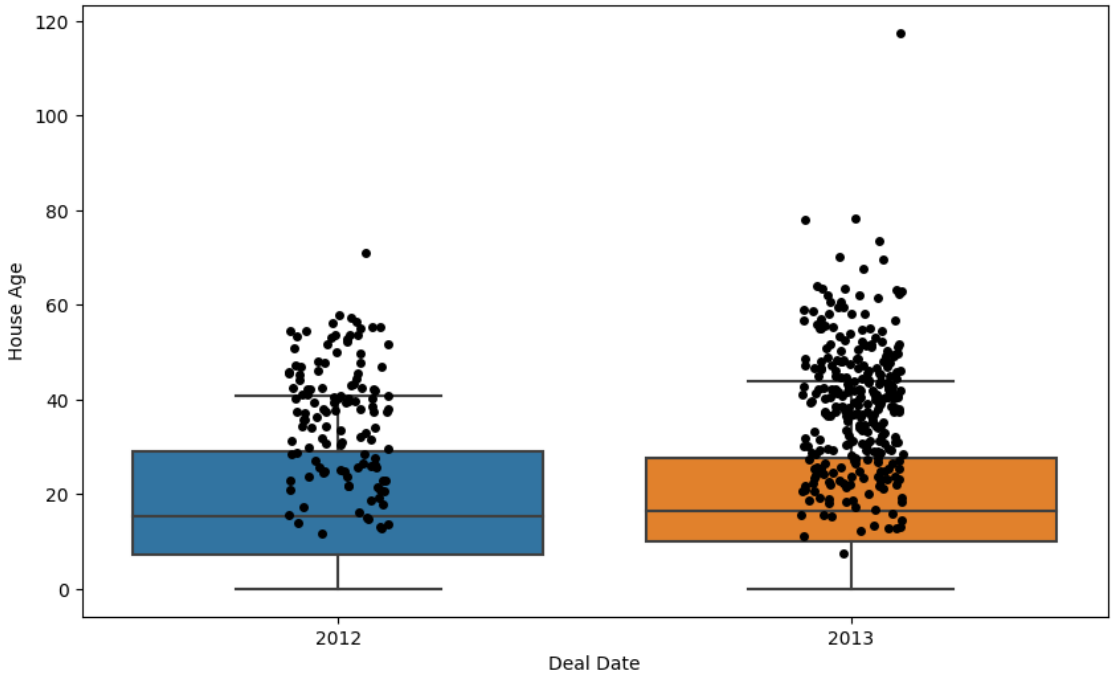
```
# Add predicted values to the DataFrame
df_n['pred'] = model_1.predict(X)

# Log-transformed scatter plot
plt.scatter(np.log(df_n['House_Age']), np.log(df_n['Price_Per_Meter']))
plt.xlabel("Log(House Age)")
plt.ylabel("Log(Price Per Meter)")
plt.show()

# Convert Deal_Date to integer
df_n['Deal_Date'] = df_n['Deal_Date'].astype(int)
```

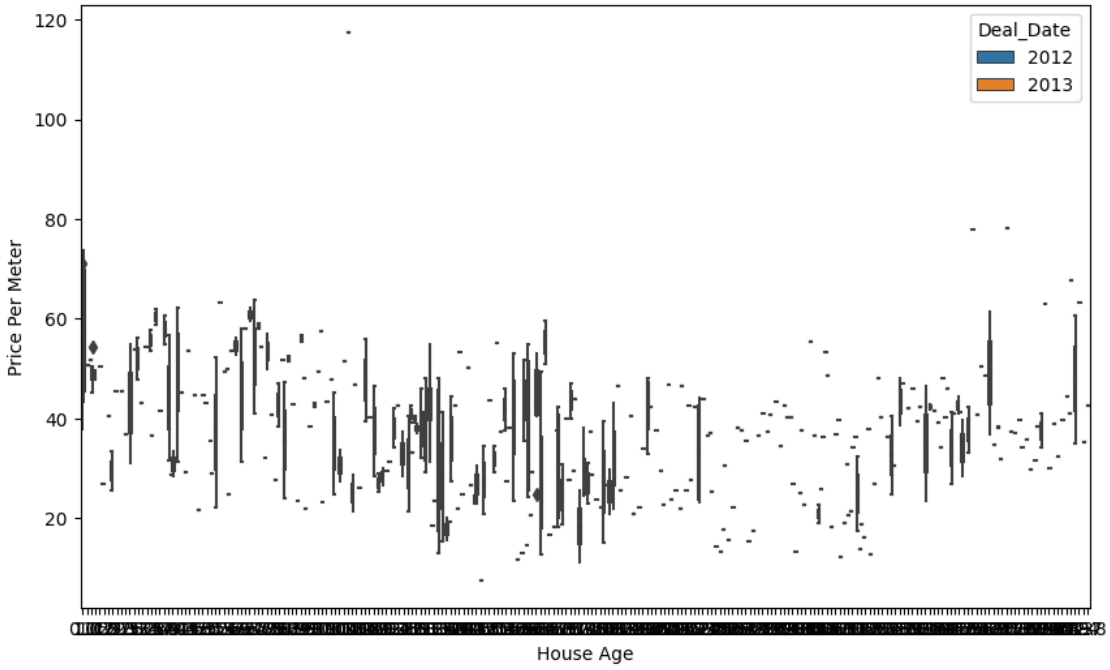
```
C:\Users\user\Anaconda3\lib\site-packages\pandas\core\arraylike.py:396: R
untimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

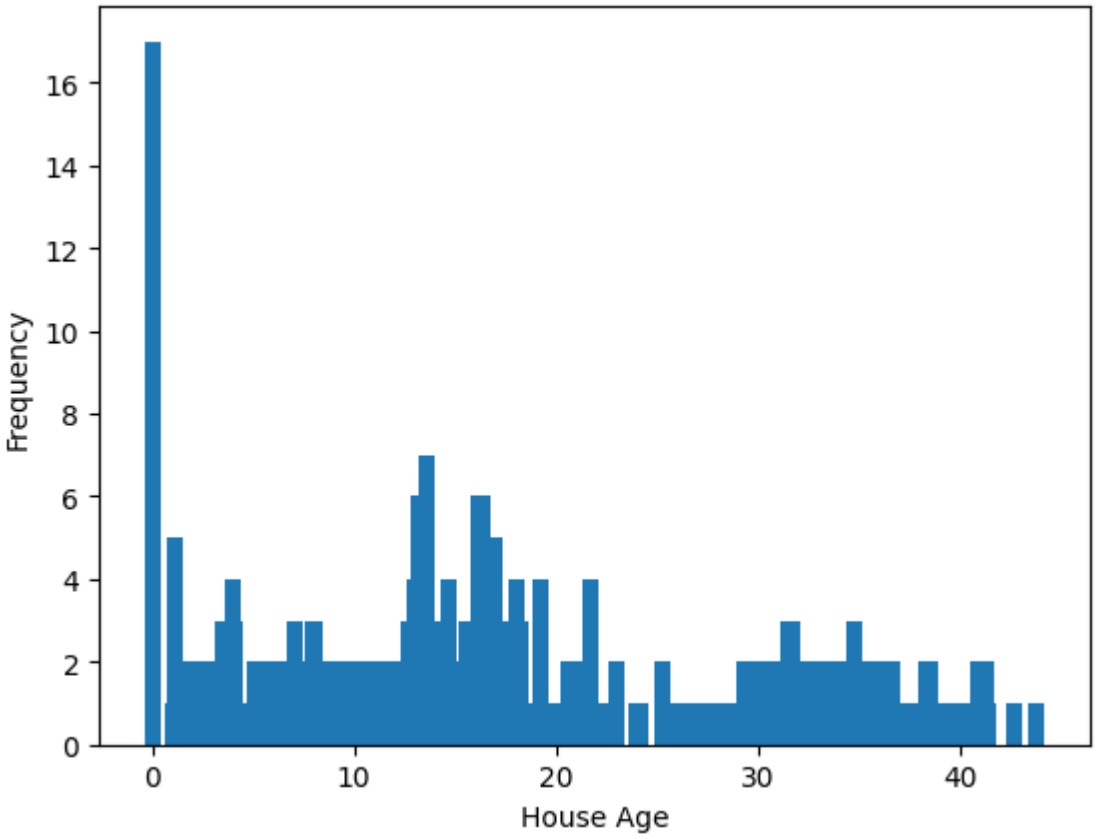


In [50]:

```
# Box plot with boxplot and jitter
plt.figure(figsize=(10, 6))
sns.boxplot(x='Deal_Date', y='House_Age', data=df_n)
sns.stripplot(x='Deal_Date', y='Price_Per_Meter', data=df_n, jitter=True,
plt.xlabel("Deal Date")
plt.ylabel("House Age")
plt.show()

# Box plot using Seaborn
plt.figure(figsize=(10, 6))
sns.boxplot(x='House_Age', y='Price_Per_Meter', hue='Deal_Date', data=df_n
plt.xlabel("House Age")
plt.ylabel("Price Per Meter")
plt.show()

# Bar plot
plt.bar(df['House_Age'].value_counts().index, df['House_Age'].value_counts
plt.xlabel("House Age")
plt.ylabel("Frequency")
plt.show()

# Create new DataFrame
my_data3 = df.copy()

# Create new column
my_data3['old_bin'] = np.where(my_data3['House_Age'] > 24, 1, 0)

# Fit linear regression model
model_1 = LinearRegression()
X3 = my_data3[['House_Age', 'old_bin']]
y3 = my_data3['Price_Per_Meter']
model_1.fit(X3, y3)

# Model summary
print("Intercept:", model_1.intercept_)
print("Coefficients:", model_1.coef_)
```


120
100
80
60
40
20
0
House Age
2012
2013
Deal Date


Deal_Date
2012
2013
120
100
80
60
40
20
Price Per Meter
House Age




```
Intercept: 46.305427206206396
Coefficients: [-0.66389086 12.04800713]
```

In [ ]: