In [36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv('Real_Estate.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Transaction date	House age	Distance to the nearest MRT station	Number of convenience stores	Latitude	Longitude	House price of unit area
0	2012-09-02 16:42:30.519336	13.3	4082.0150	8	25.007059	121.561694	6.488673
1	2012-09-04 22:52:29.919544	35.5	274.0144	2	25.012148	121.546990	24.970725
2	2012-09-05 01:10:52.349449	1.1	1978.6710	10	25.003850	121.528336	26.694267
3	2012-09-05 13:26:01.189083	22.2	1055.0670	5	24.962887	121.482178	38.091638
4	2012-09-06 08:29:47.910523	8.5	967.4000	6	25.011037	121.479946	21.654710

In [4]:
```
df.tail()
```

Out[4]:

	Transaction date	House age	Distance to the nearest MRT station	Number of convenience stores	Latitude	Longitude	House price of unit area
409	2013-07-25 15:30:36.565239	18.3	170.12890	6	24.981186	121.486798	29.096310
410	2013-07-26 17:16:34.019780	11.9	323.69120	2	24.950070	121.483918	33.871347
411	2013-07-28 21:47:23.339050	0.0	451.64190	8	24.963901	121.543387	25.255105
412	2013-07-29 13:33:29.405317	35.9	292.99780	5	24.997863	121.558286	25.285620
413	2013-08-01 09:49:41.506402	12.0	90.45606	6	24.952904	121.526395	37.580554

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 7 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   Transaction date                        414 non-null    object 
 1   House age                               414 non-null    float64
 2   Distance to the nearest MRT station     414 non-null    float64
 3   Number of convenience stores            414 non-null    int64  
 4   Latitude                                414 non-null    float64
 5   Longitude                               414 non-null    float64
 6   House price of unit area                414 non-null    float64
dtypes: float64(5), int64(1), object(1)
memory usage: 22.8+ KB
```

In [6]:
```
df.describe()
```

Out[6]:

	House age	Distance to the nearest MRT station	Number of convenience stores	Latitude	Longitude	House price of unit area
count	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000
mean	18.405072	1064.468233	4.265700	24.973605	121.520268	29.102149
std	11.757670	1196.749385	2.880498	0.024178	0.026989	15.750935
min	0.000000	23.382840	0.000000	24.932075	121.473888	0.000000
25%	9.900000	289.324800	2.000000	24.952422	121.496866	18.422493
50%	16.450000	506.114400	5.000000	24.974353	121.520912	30.394070
75%	30.375000	1454.279000	6.750000	24.994947	121.544676	40.615184
max	42.700000	6306.153000	10.000000	25.014578	121.565321	65.571716

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Transaction date                        0
House age                               0
Distance to the nearest MRT station     0
Number of convenience stores            0
Latitude                                0
Longitude                               0
House price of unit area                0
dtype: int64
```

In [9]:
```
df.columns
```

Out[9]:
```
Index(['Transaction date', 'House age', 'Distance to the nearest MRT station',
       'Number of convenience stores', 'Latitude', 'Longitude',
       'House price of unit area'],
      dtype='object')
```

In [10]:
```
df.dtypes
```

Out[10]:
```
Transaction date                         object
House age                               float64
Distance to the nearest MRT station     float64
Number of convenience stores              int64
Latitude                                float64
Longitude                               float64
House price of unit area                float64
dtype: object
```

In [12]:
```
df.duplicated().sum()
```

Out[12]:
```
0
```

In [13]:
```
df.nunique()
```

Out[13]:
```
Transaction date                        414
House age                               178
Distance to the nearest MRT station     183
Number of convenience stores             11
Latitude                                414
Longitude                               414
House price of unit area                384
dtype: int64
```

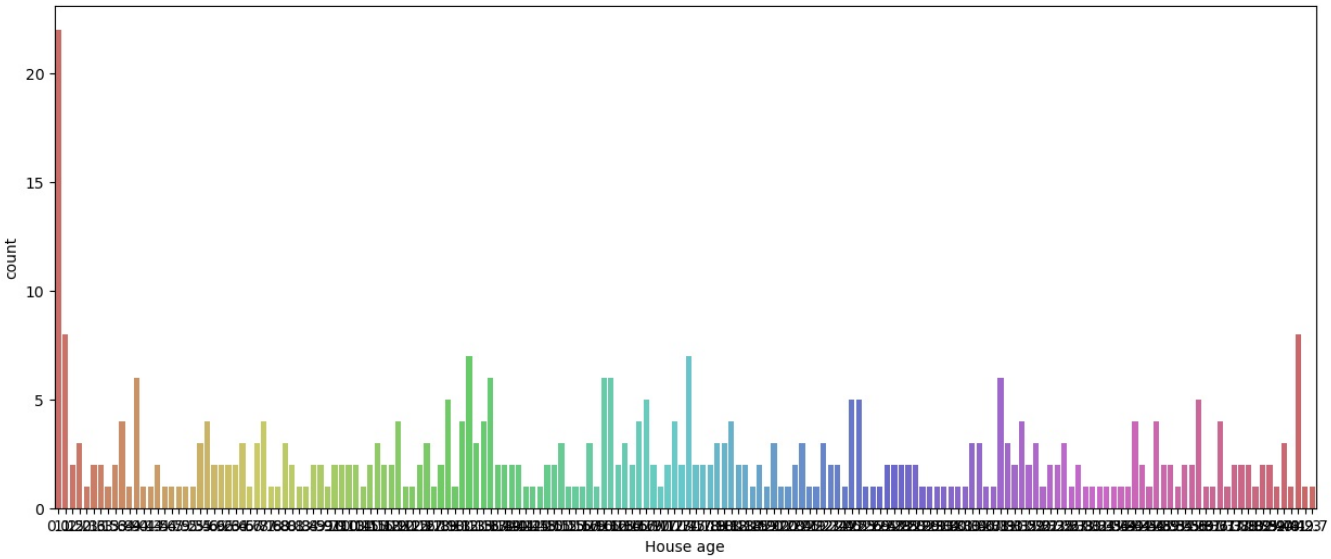
In [16]:
```
df['House age'].unique()
```

Out[16]:
```
array([13.3, 35.5,  1.1, 22.2,  8.5, 38.5, 15.2, 24. , 13. , 13.2,  6.5,
       27.5, 12.9,  8. ,  4.9,  3.1, 10.3, 13.7,  8.4, 31.3, 17.3, 14.1,
        1.5, 16.1, 34.4, 18.2, 11.9,  5.4,  9.7, 32.8, 31. , 34.8,  9.9,
        5.9, 37.1, 26.8, 30.4,  4. , 25.3, 16.4, 26.4, 19.2, 30.3, 13.5,
        8.1, 32.3, 38. , 24.2,  5.6, 14.2,  3.8, 38.2, 15.6,  0. , 18.9,
       16.6,  7.6, 12. ,  6.6, 11.6, 25.9, 29.4, 17.5, 32.6, 16.3, 39.2,
       18.4, 33. , 17.7, 35.9, 16.2, 35.4, 37.7, 40.9, 34.9, 32. , 12.7,
        4.5, 33.6, 39.8,  6.8, 18.3, 17. , 15.1, 34.5, 37.8,  5.3,  4.3,
       14. , 28.2, 35.3, 32.1, 15.7,  6.4, 32.7, 20.6, 25.6, 12.6, 20.4,
       33.4, 30. , 10.4, 16.9, 23. , 32.5, 17.4, 30.9, 35.8, 30.6, 33.9,
       15.9, 15. , 17.8, 15.5, 18.1, 11.5, 13.9,  8.9, 17.9, 36.6, 26.9,
       18. , 31.9, 34. , 17.2,  4.1,  3.5,  7.1, 12.8, 12.2, 14.8, 41.3,
        3.9,  2.6,  4.7, 10. , 28. , 34.6, 12.5, 39.7, 28.6, 14.4, 36.1,
        7.8,  2. ,  8.3, 37.3, 30.8, 19.1,  5.2, 21.7, 20.9, 33.2, 13.8,
       29.3, 31.5, 16.5,  3.6, 13.6, 13.1,  6.3, 17.1, 42.7,  2.1, 18.5,
        6.2, 21.2, 20.3, 30.1, 20. , 11.8, 16. , 29.1, 30.7, 40.1, 10.5,
        4.6, 33.5])
```

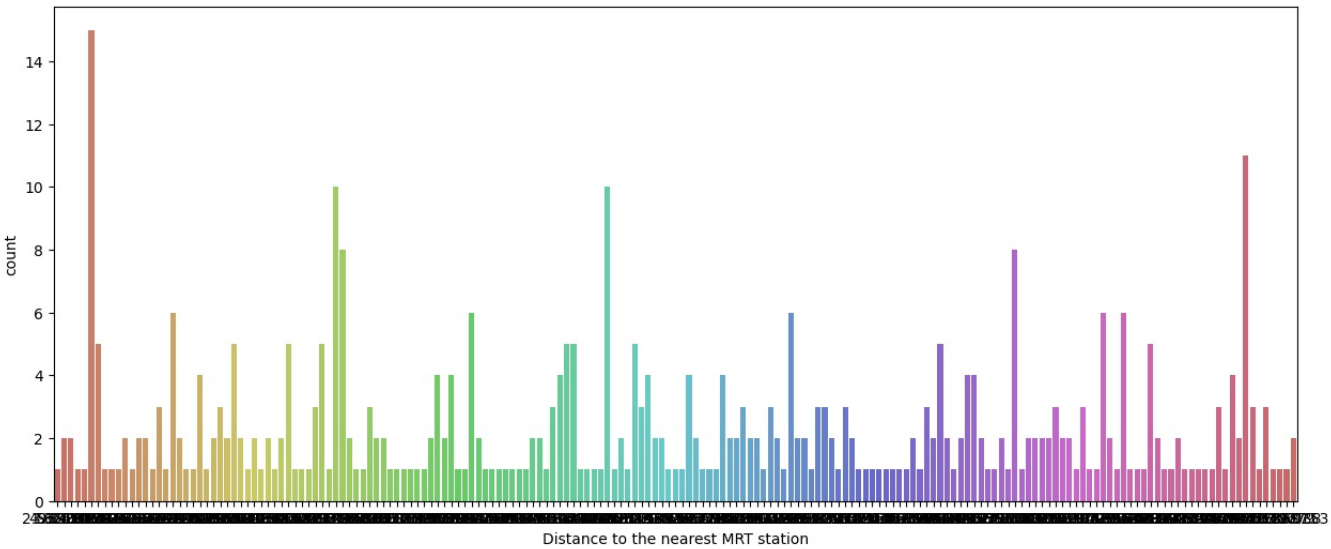
In [17]:
```
df['House age'].value_counts()
```

Out[17]:
```
House age
0.0     22
1.1      8
40.9     8
13.3     7
17.5     7
        ..
36.6     1
26.4     1
30.3     1
15.5     1
33.5     1
Name: count, Length: 178, dtype: int64
```

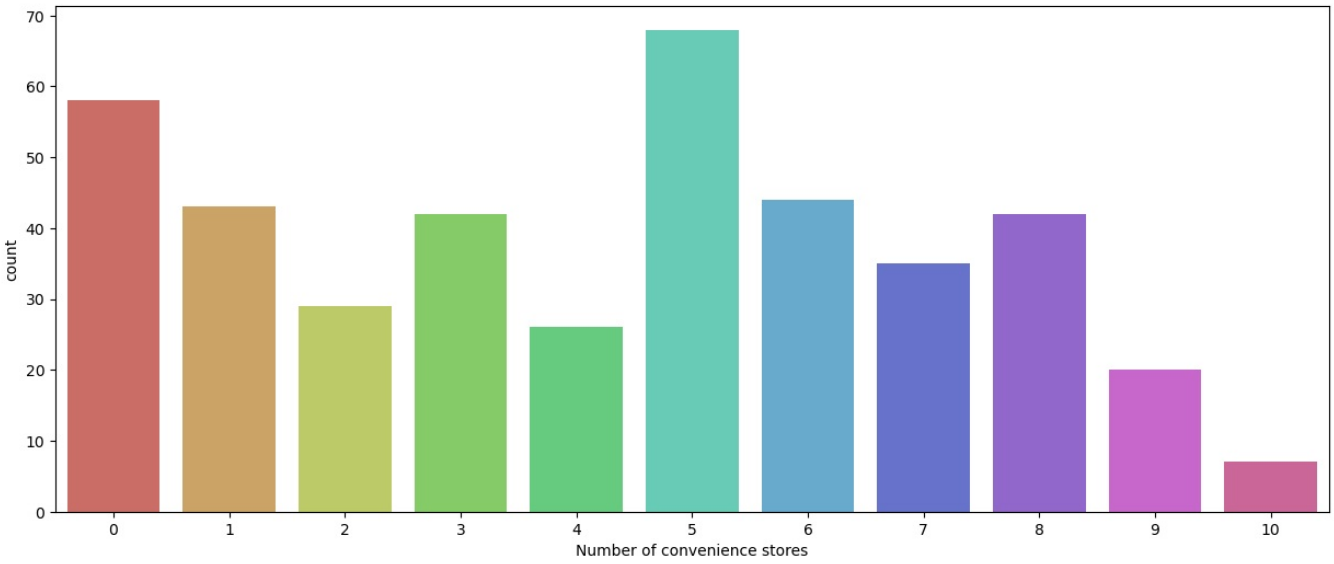
In [23]:
```
plt.figure(figsize=(15, 6))
sns.countplot(x='House age', data=df, palette='hls')
plt.show()
```



In [24]:
```
plt.figure(figsize=(15,6))
sns.countplot(x='Distance to the nearest MRT station',data=df,palette='hls')
plt.show()
```



In [25]:
```
plt.figure(figsize=(15, 6))
sns.countplot(x='Number of convenience stores', data=df, palette='hls')
plt.show()
```



In [26]:
```
print(df.isnull().sum())
```

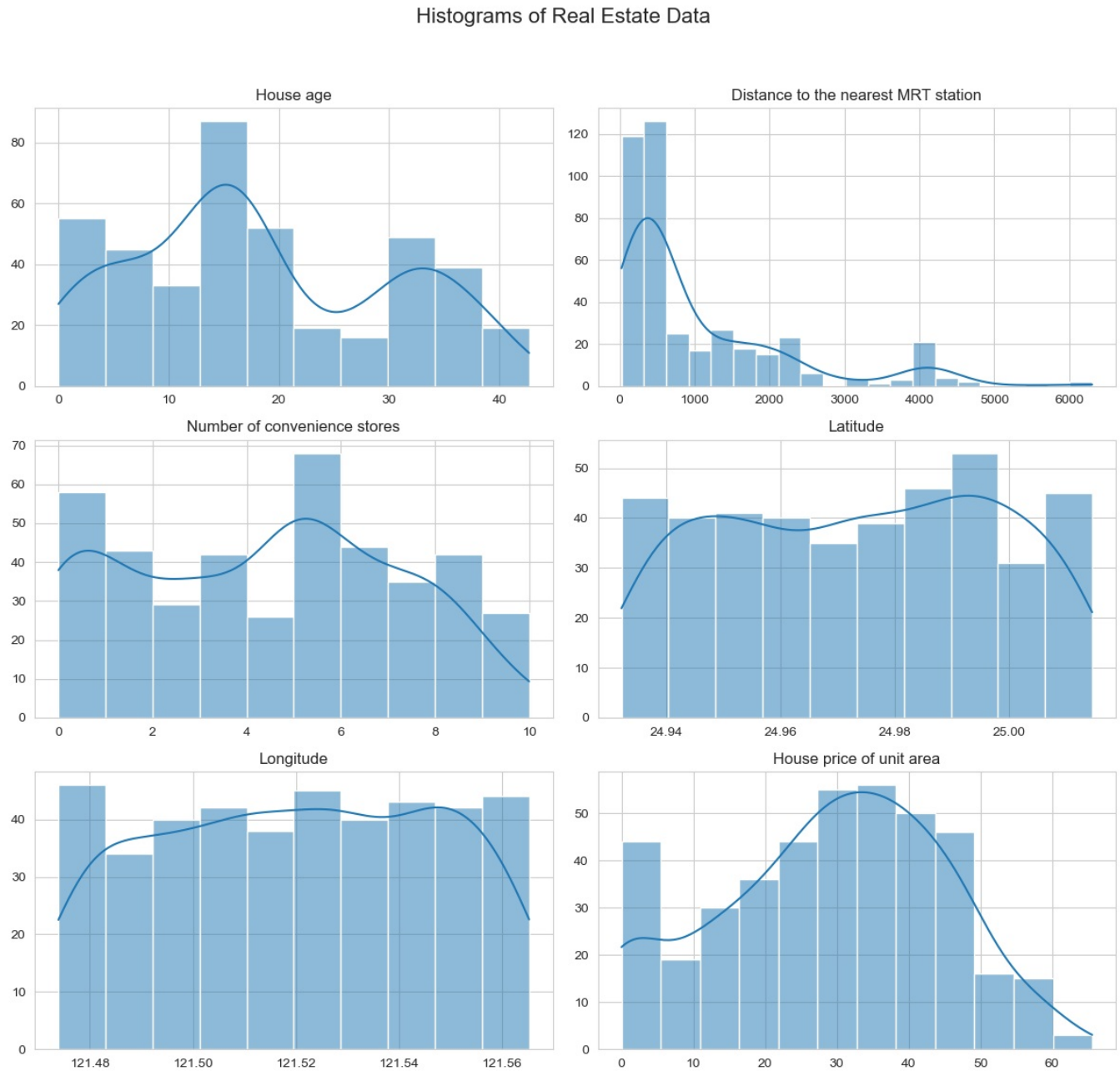
```
Transaction date                        0
House age                               0
Distance to the nearest MRT station     0
Number of convenience stores            0
Latitude                                0
Longitude                               0
House price of unit area                0
dtype: int64
```

In [29]:
```
descriptive_stats = df.describe()

print(descriptive_stats)
```

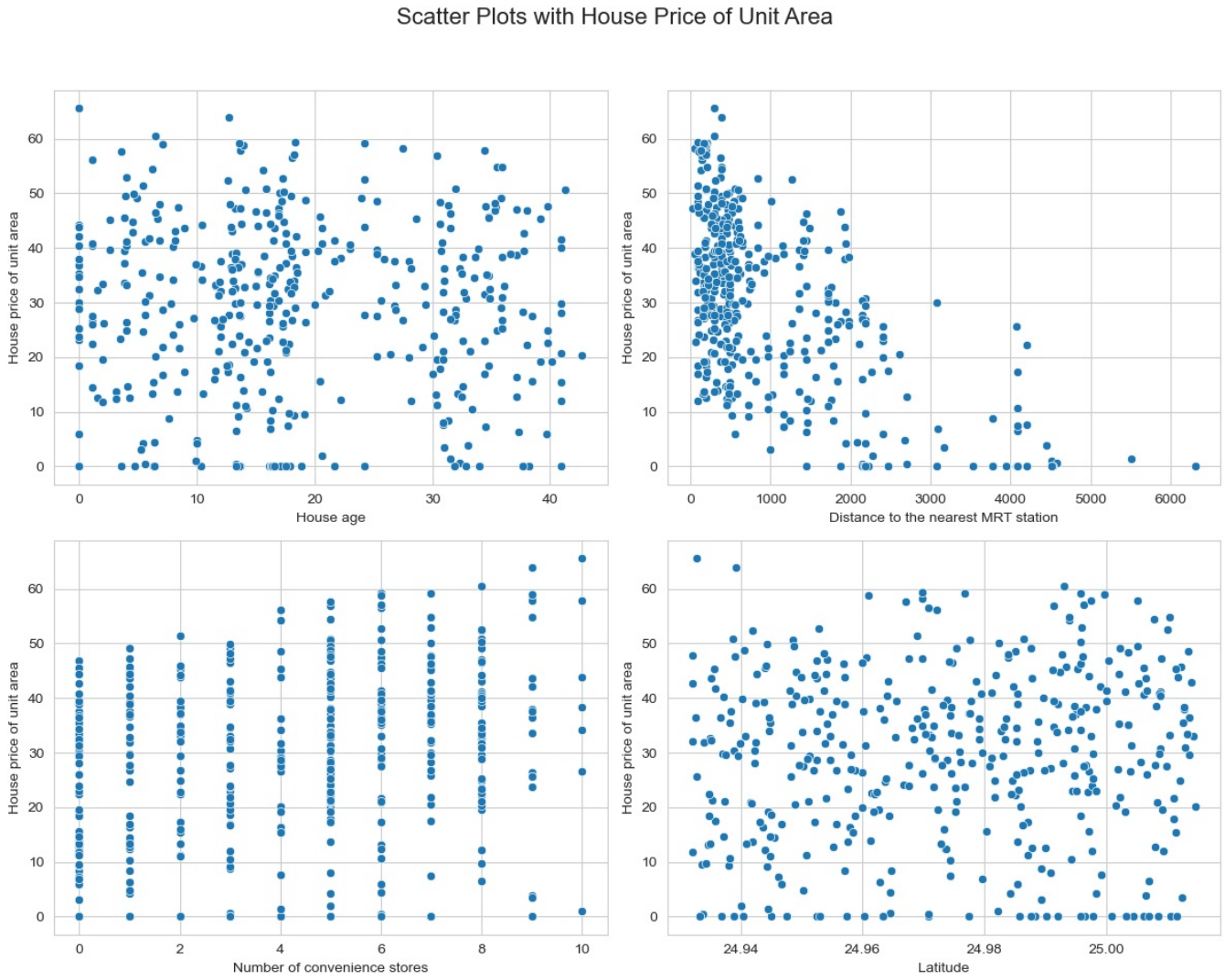
```
        House age  Distance to the nearest MRT station
count  414.000000                          414.000000  \
mean    18.405072                         1064.468233
std     11.757670                         1196.749385
min      0.000000                           23.382840
25%      9.900000                          289.324800
50%     16.450000                          506.114400
75%     30.375000                         1454.279000
max     42.700000                         6306.153000

       Number of convenience stores    Latitude   Longitude
count                    414.000000  414.000000  414.000000  \
mean                       4.265700   24.973605  121.520268
std                        2.880498    0.024178    0.026989
min                        0.000000   24.932075  121.473888
25%                        2.000000   24.952422  121.496866
50%                        5.000000   24.974353  121.520912
75%                        6.750000   24.994947  121.544676
max                       10.000000   25.014578  121.565321

       House price of unit area
count                414.000000
mean                  29.102149
std                   15.750935
min                    0.000000
25%                   18.422493
50%                   30.394070
75%                   40.615184
max                   65.571716
```

In [31]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set_style("whitegrid")

# Create histograms for the numerical columns
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(12, 12))
fig.suptitle('Histograms of Real Estate Data', fontsize=16)

cols = ['House age', 'Distance to the nearest MRT station', 'Number of convenience stores',
        'Latitude', 'Longitude', 'House price of unit area']

for i, col in enumerate(cols):
    sns.histplot(df[col], kde=True, ax=axes[i//2, i%2])
    axes[i//2, i%2].set_title(col)
    axes[i//2, i%2].set_xlabel('')
    axes[i//2, i%2].set_ylabel('')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

```
plt.show()
```



In [34]:
```
# Scatter plots to observe the relationship with house price
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))
fig.suptitle('Scatter Plots with House Price of Unit Area', fontsize=16)

# Scatter plot for each variable against the house price
sns.scatterplot(df, x='House age', y='House price of unit area', ax=axes[0, 0])
sns.scatterplot(df, x='Distance to the nearest MRT station', y='House price of unit area', ax=axes[0, 1])
sns.scatterplot(df, x='Number of convenience stores', y='House price of unit area', ax=axes[1, 0])
sns.scatterplot(df, x='Latitude', y='House price of unit area', ax=axes[1, 1])

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



The scatter plots revealed interesting relationships between various factors and house prices:

House Age vs. House Price: There doesn't seem to be a strong linear relationship between house age and price. However, it appears that very new and very old houses might have higher prices.

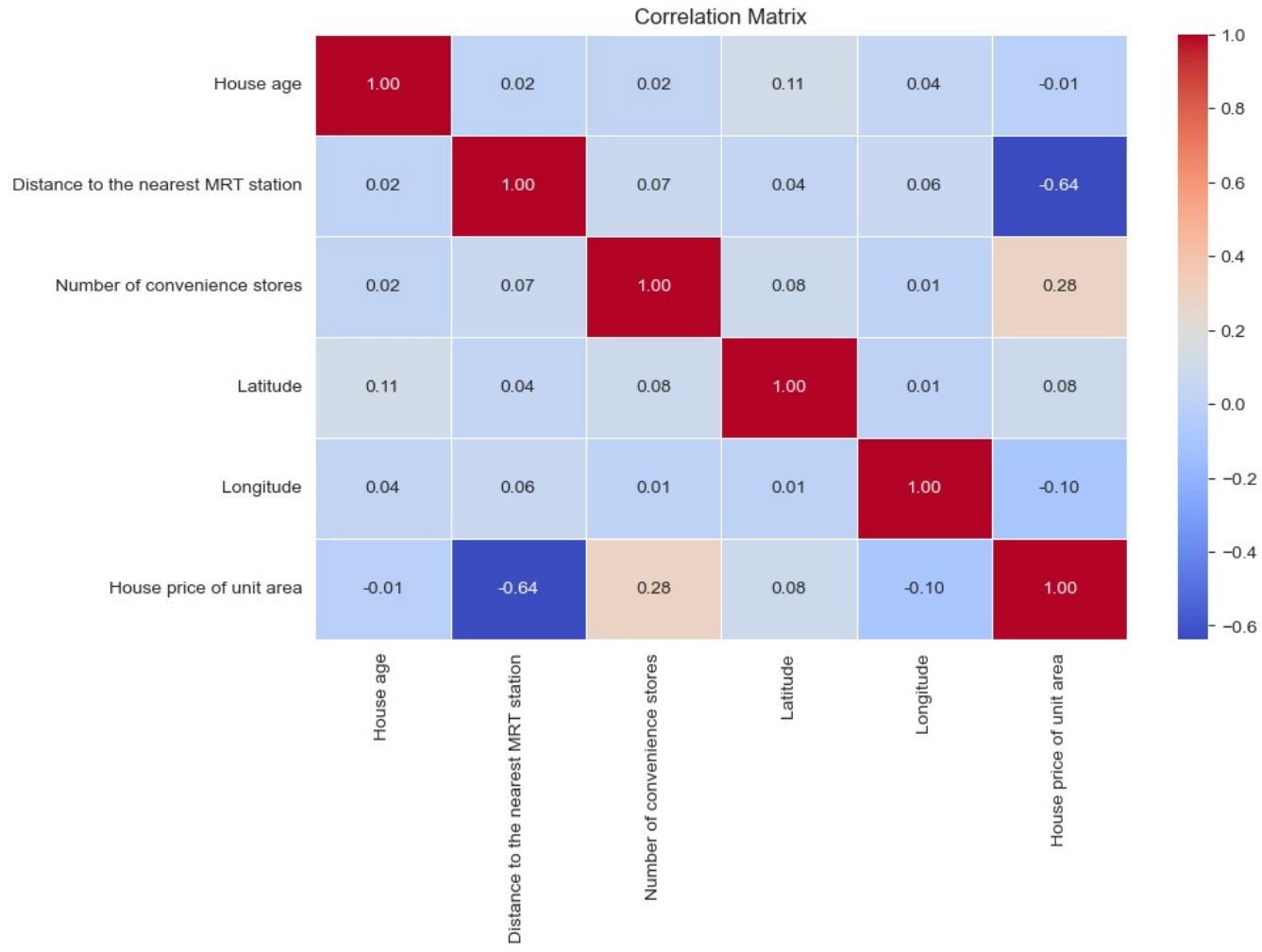
Distance to the Nearest MRT Station vs. House Price: There is a clear trend showing that as the distance to the nearest MRT station increases, the house price tends to decrease. It suggests a strong negative relationship between these two variables.

Number of Convenience Stores vs. House Price: There seems to be a positive relationship between the number of convenience stores and house prices. Houses with more convenience stores in the vicinity tend to have higher prices.

Latitude vs. House Price: While not a strong linear relationship, there seems to be a pattern where certain latitudes correspond to higher or lower house prices. It could be indicative of specific neighbourhoods being more desirable.

In [47]:
```
df = df.drop('Transaction date', axis = 1)
```

In [48]:
```
correlation_matrix = df.corr()

# Plotting the correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix')
plt.show()

print(correlation_matrix)
```




```
                                House age
House age                        1.000000  \
Distance to the nearest MRT station  0.021596
Number of convenience stores         0.021973
Latitude                             0.114345
Longitude                            0.036449
House price of unit area            -0.012284

                                Distance to the nearest MRT station
House age                                                  0.021596  \
Distance to the nearest MRT station                        1.000000
Number of convenience stores                               0.069015
Latitude                                                   0.038954
Longitude                                                  0.064229
House price of unit area                                  -0.636579

                                Number of convenience stores  Latitude
House age                                           0.021973  0.114345  \
Distance to the nearest MRT station                 0.069015  0.038954
Number of convenience stores                        1.000000  0.082725
Latitude                                            0.082725  1.000000
Longitude                                           0.013156  0.007754
House price of unit area                            0.280763  0.081008

                                Longitude  House price of unit area
House age                        0.036449                 -0.012284
Distance to the nearest MRT station  0.064229                 -0.636579
Number of convenience stores         0.013156                  0.280763
Latitude                             0.007754                  0.081008
Longitude                            1.000000                 -0.098626
House price of unit area            -0.098626                  1.000000
```

In [49]:

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Selecting features and target variable
features = ['Distance to the nearest MRT station', 'Number of convenience stores', 'Latitude', 'Longitude']
target = 'House price of unit area'

X = df[features]
y = df[target]

# Splitting the dataset 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)

# Model initialization
```

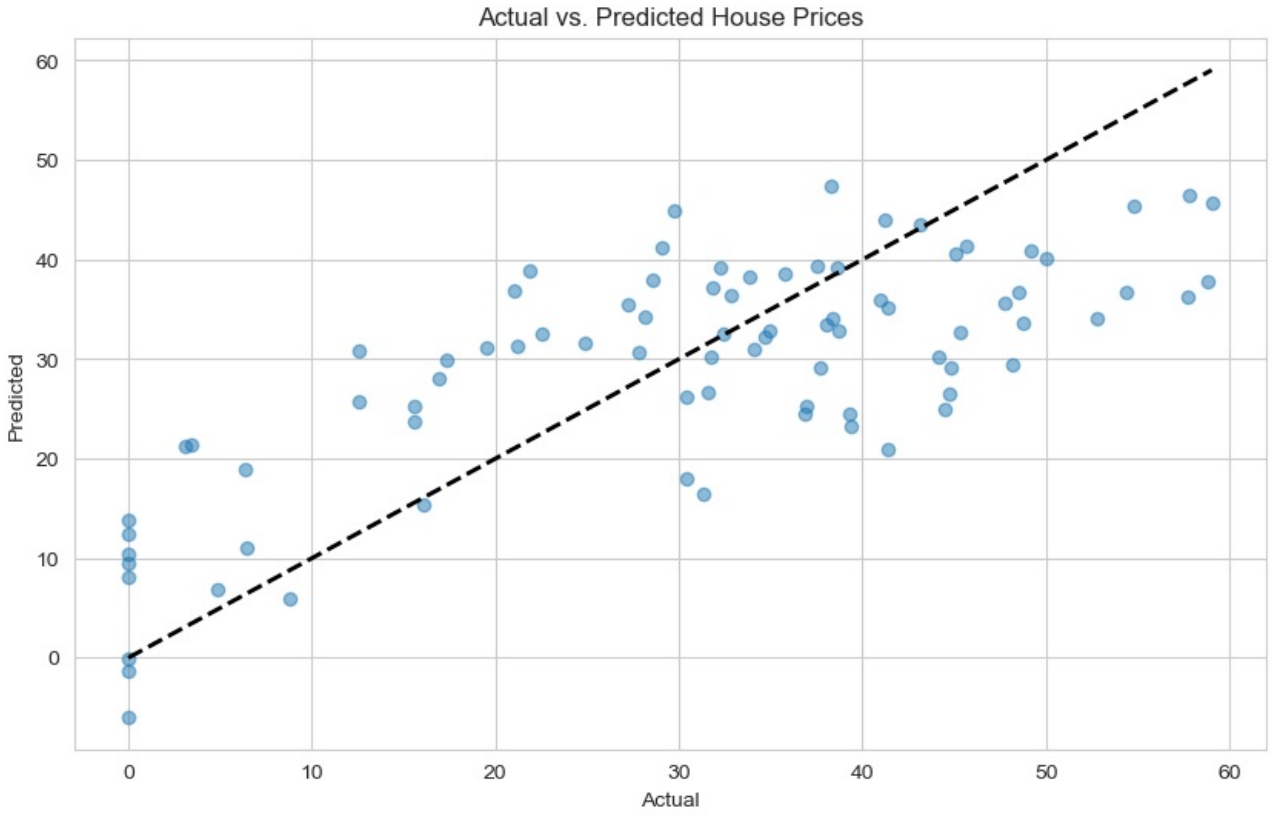
```
model = LinearRegression()

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

Out[49]:

```
▾ LinearRegression
LinearRegression()
```

In [39]:

```
# Making predictions using the linear regression model
y_pred_lr = model.predict(X_test)

# Visualization: Actual vs. Predicted values
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_lr, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs. Predicted House Prices')
plt.show()
```



In [40]:

```
plt.figure(figsize=(15,6))
sns.pairplot(data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```

```
<Figure size 1500x600 with 0 Axes>
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js