ANALYSIS OF FACTORS AFFECTING APARTMENT PRICES

FLOW

Purpose

Implementation

Conclusion and limitations

Recommendations

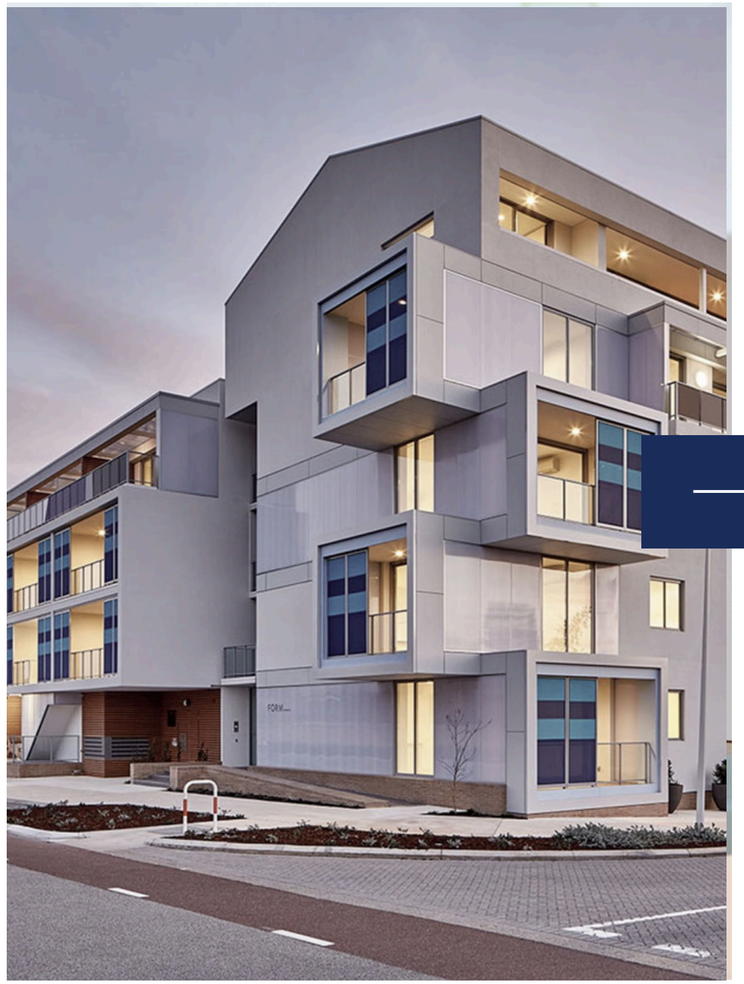
REASON FOR THE TOPIC

- **Relevance in the Real Estate Market**
- **Post-COVID Apartment Prices (After 2020)**
- **Factors that have huge impact on apartment price.**
- **Researching and forecasting apartment prices**

- **What are the global trends in apartment prices post-2020 ?**
- **Which three factors most significantly impact apartment prices (USD)?**
- **What are the key factors influencing apartment prices (USD), and what are their relationships?**

Introduction.

The 'World's Real Estate Data (147k)' dataset contains details of apartment properties worldwide. Including attributes like country, location, apartment floors, building total floors, apartment rooms, total area, and price across different countries. This comprehensive dataset provides valuable insights into the global real estate market. We will now delve into the data to extract insights, perform exploratory data analysis (EDA), draw meaningful conclusions, and train a model to evaluate the impact of the variables in the dataset.



Dataset

	title	country	location	building_construction_year	building_total_floors	apartment_floor	apartment_rooms	apartment_bedrooms	apartment_bathrooms	apartment_total_area	apartment_living_area	price_in_USD	
0	2 room apartment 120 m² in Mediterranean Regio...	Turkey	Mediterranean Region, Turkey	NaN	5.0	1.0	3.0	2.0	2.0	120 m²	110 m²	315209.0	https://realting.com/uploa
1	4 room villa 500 m² in Kalkan, Turkey	Turkey	Kalkan, Mediterranean Region, Kas, Turkey	2021.0	2.0	NaN	NaN	NaN	NaN	500 m²	480 m²	1108667.0	https://realting.com/uploa
2	1 room apartment 65 m² in Antalya, Turkey	Turkey	Mediterranean Region, Antalya, Turkey	NaN	5.0	2.0	2.0	1.0	1.0	65 m²	60 m²	173211.0	https://realting.com/uploa
3	1 room apartment in Pattaya, Thailand	Thailand	Chon Buri Province, Pattaya, Thailand	2020.0	15.0	5.0	2.0	1.0	1.0	NaN	40 m²	99900.0	https://realting.com/uploa
4	2 room apartment in Pattaya, Thailand	Thailand	Chon Buri Province, Pattaya, Thailand	2026.0	8.0	3.0	3.0	2.0	1.0	NaN	36 m²	67000.0	https://realting.com/uploa
...	...	...	...	...	...	...	...	...	...	...	...	...	
147531	5 room apartment 310 m² in Gazipasa, Turkey	Turkey	Mediterranean Region, Gazipasa, Turkey	NaN	NaN	NaN	NaN	5.0	NaN	310 m²	NaN	597810.0	https://realting.com/uploa
147532	4 room apartment 192 m² in Marmara Region, Turkey	Turkey	Marmara Region, Turkey	2023.0	5.0	NaN	5.0	4.0	2.0	192 m²	151 m²	637195.0	https://realting.com/uploa
147533	2 room apartment in Marmara Region, Turkey	Turkey	Marmara Region, Turkey	NaN	NaN	NaN	3.0	2.0	2.0	NaN	84 m²	477146.0	https://realting.com/uploa
147534	Apartment in Akarca, Turkey	Turkey	Akarca, Central Anatolia Region, Turkey	2023.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	819163.0	https://realting.com/uploa
147535	4 room apartment 140 m² in, Turkey	Turkey	Turkey	NaN	2.0	NaN	5.0	4.0	NaN	140 m²	NaN	939164.0	https://realting.com/uploa

147536 rows × 14 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 147536 entries, 0 to 147535
Data columns (total 14 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   title                      147536 non-null  object 
 1   country                    147406 non-null  object 
 2   location                   147405 non-null  object 
 3   building_construction_year 64719 non-null   float64
 4   building_total_floors      68224 non-null   float64
 5   apartment_floor            54592 non-null   float64
 6   apartment_rooms            74178 non-null   float64
 7   apartment_bedrooms         36982 non-null   float64
 8   apartment_bathrooms        55973 non-null   float64
 9   apartment_total_area       141796 non-null  object 
 10  apartment_living_area      27712 non-null   object 
 11  price_in_USD               144961 non-null  float64
 12  image                      147536 non-null  object 
 13  url                        147536 non-null  object 
dtypes: float64(7), object(7)
memory usage: 15.8+ MB
```

```
df.describe()
```

	building_construction_year	building_total_floors	apartment_floor	apartment_rooms	apartment_bedrooms	apartment_bathrooms	price_in_USD
count	64719.000000	68224.000000	54592.000000	74178.000000	36982.000000	55973.000000	1.449610e+05
mean	1996.921754	8.575692	5.791709	2.572097	2.289222	1.364229	4.121722e+05
std	157.527635	8.356781	5.541368	1.319545	18.276913	0.745019	8.420984e+05
min	1.000000	-1.000000	-2.000000	-1.000000	-1.000000	1.000000	0.000000e+00
25%	2004.000000	2.000000	2.000000	2.000000	1.000000	1.000000	1.054200e+05
50%	2021.000000	5.000000	4.000000	2.000000	2.000000	1.000000	1.902120e+05
75%	2024.000000	14.000000	8.000000	3.000000	3.000000	2.000000	3.989300e+05
max	2316.000000	124.000000	202.000000	124.000000	2009.000000	43.000000	3.060283e+07

The variables require cleaning and transformation

- **(1) building_construction_year**
- **(2) building_total_floors and apartment_floor**
- **(3) apartment_total_area**

Preprocessing

```
[160] #logic between the apartment_floor and the building_total_floors
      condition = df['apartment_floor'] <= df['building_total_floors']
      df = df[condition]
```

```
[161] df = df.dropna(how='any',axis=0)
      #dropping any rows that has Nan (no value) innit.
```

```
[162] #dropping Unnecessary, excessive, nominal variables in dataset
      df.drop(['location'],axis=1,inplace=True)
      df.drop(['title'],axis=1,inplace=True)
      df.drop(['image'],axis=1,inplace=True)
      df.drop(['url'],axis=1,inplace=True)
```

```
[163] #extract the numeric value only from area_variables.
      df['apartment_total_area'] = df['apartment_total_area'].apply(lambda x: int(str(x).rstrip(" m²").replace(' ','')))
      df['apartment_living_area'] = df['apartment_living_area'].apply(lambda x: int(str(x).rstrip(" m²").replace(' ','')))
```

```
[164] #looking for duplicated values in the dataset.
      df.isna().sum()
```

	0
country	0
building_construction_year	0
building_total_floors	0
apartment_floor	0
apartment_rooms	0
apartment_bedrooms	0
apartment_bathrooms	0
apartment_total_area	0
apartment_living_area	0
price_in_USD	0

```
'''As you can see above, the values of apartment_total_area and apartment_living_area
almost the same. Not trying to have VIF in the model.'''
df.drop(['apartment_living_area'],axis=1,inplace=True)
df.drop(['apartment_rooms'],axis=1,inplace=True)
```

```
'''make a function to visualize the datapoint though boxplot, and counts how many
outliers are there in the dataset.'''
def visualize_outliers(df, price_in_USD):
  plt.figure(figsize=(10, 6))  # adjusting figuresize of the plots.
  sns.boxplot(x=df[price_in_USD])
  plt.xlabel(price_in_USD)
  plt.show()
# calculating Interquartile range
  Q1 = df[price_in_USD].quantile(0.25)
  Q3 = df[price_in_USD].quantile(0.75)
  IQR = Q3 - Q1
# Identifying lower and upper bounds.
  upper_bound = Q3 + 1.5 * IQR
  lower_bound = Q1 - 1.5 * IQR
  outliers = df[(df[price_in_USD] < lower_bound) | (df[price_in_USD] > upper_bound)][price_in_USD]
  if not outliers.empty:
    print("\nObservations need to be considered:")
    print(outliers)
    print(f"\nNumber of outliers: {len(outliers)}")
  else:
    print("\nNo outliers are in this variable")
```

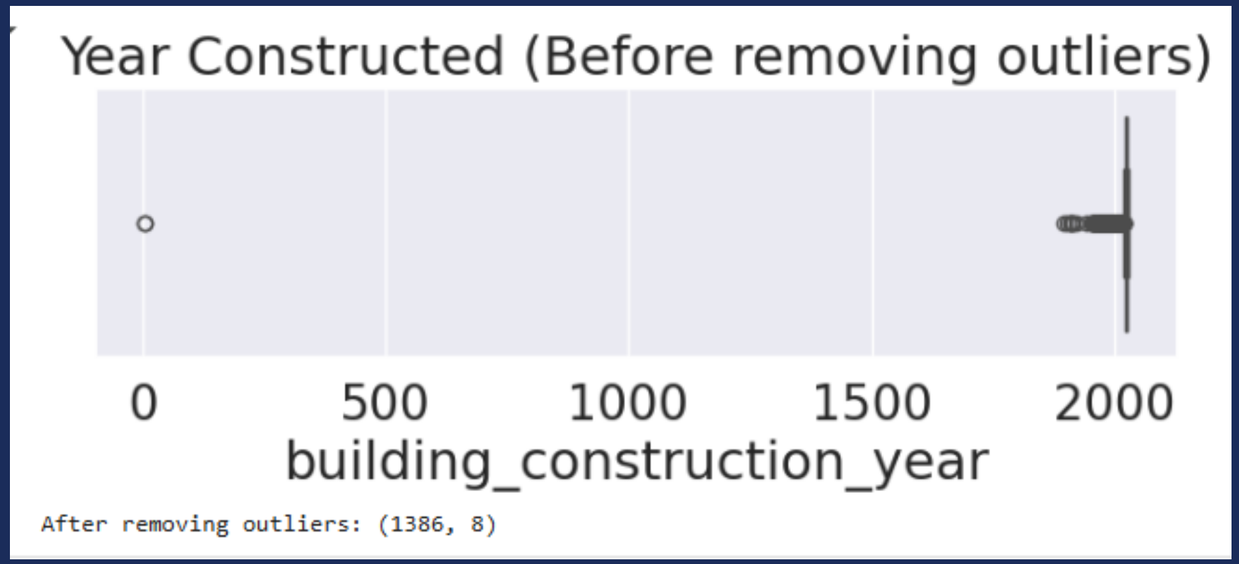
```
try:
  visualize_outliers(df, 'price_in_USD')  # execute.
except NameError:
    print("Error: Dataframe is not defined")
```

- **Extract numerical values from the apartment_total_area and apartment_living_area, removing the units (m²).**
- **Keep only valuable rows and clean duplicate and missing values from the dataset.**
- **Drop the nominal columns from the dataset.**

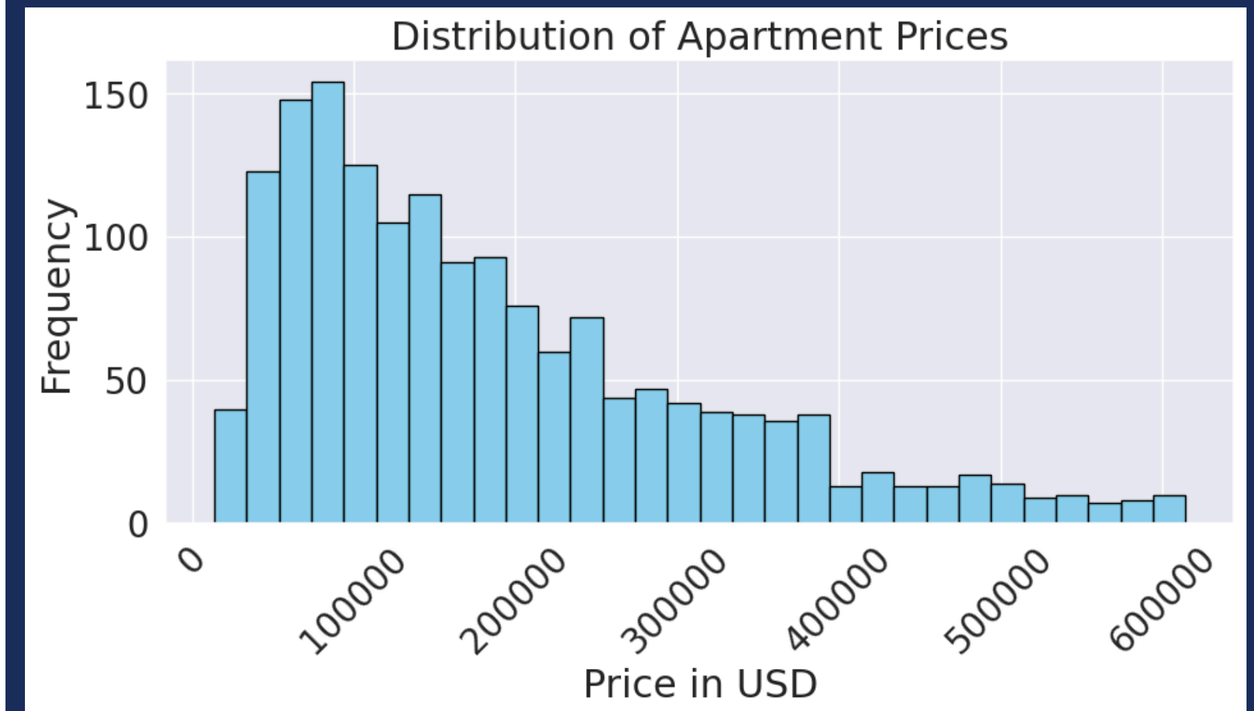
Removing outliers using Boxplot


```
Observations need to be considered:
3937       686947.0
4065       696721.0
4074       820631.0
4090       733677.0
4276      1190186.0
             ...
143768     880924.0
145528     664044.0
147415     650000.0
147422     730000.0
147447     850000.0
Name: price_in_USD, Length: 123, dtype: float64

Number of outliers: 123
```




```
After removing outliers: (1386, 8)
```



After removing outliers.

```
df = df_clean
df.shape
```

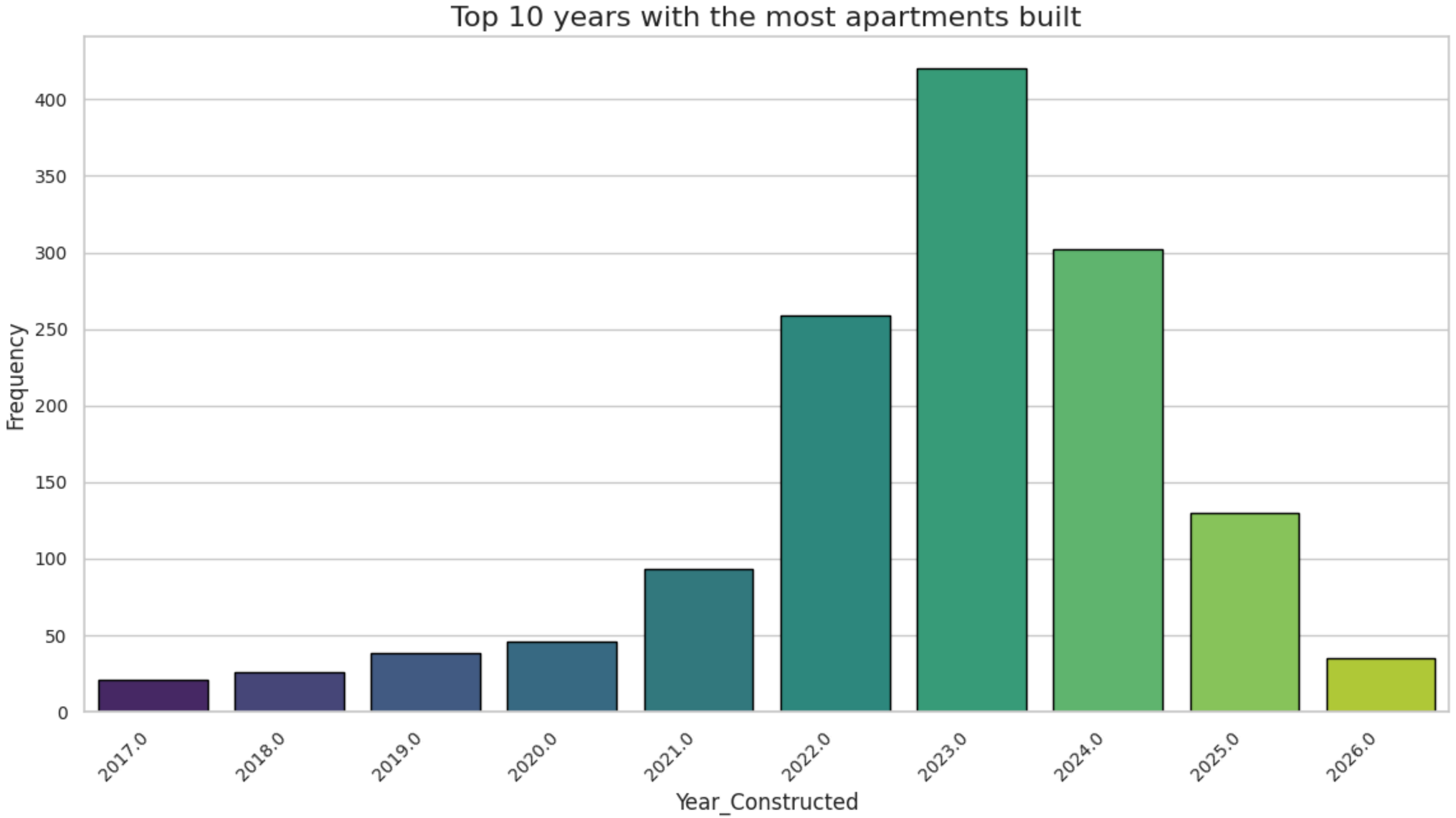
```
(1329, 8)
```

```
df.describe()
```

	building_construction_year	building_total_floors	apartment_floor	apartment_bedrooms	apartment_bathrooms	apartment_total_area	price_in_USD
count	1329.000000	1329.000000	1329.000000	1329.000000	1329.000000	1329.000000	1329.000000
mean	2022.818661	12.249059	6.109857	1.670429	1.355154	84.254327	184644.555305
std	1.789632	11.181638	7.239552	0.776949	0.544943	37.214030	127110.136827
min	2017.000000	1.000000	1.000000	1.000000	1.000000	26.000000	15985.000000
25%	2022.000000	4.000000	1.000000	1.000000	1.000000	56.000000	85120.000000
50%	2023.000000	8.000000	3.000000	1.000000	1.000000	76.000000	149674.000000
75%	2024.000000	16.000000	8.000000	2.000000	2.000000	105.000000	250000.000000
max	2027.000000	65.000000	51.000000	5.000000	4.000000	195.000000	614347.000000

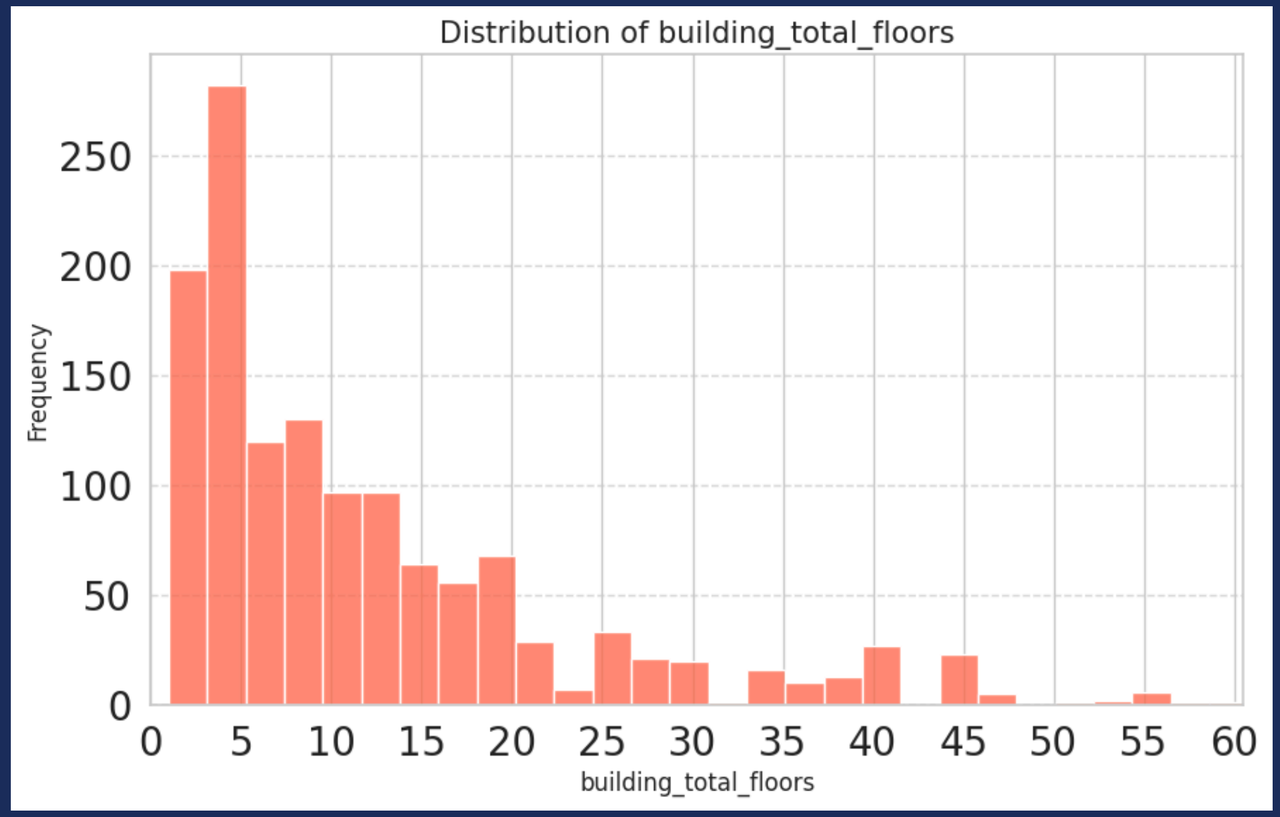
⟶ After the processing phase, we will proceed with univariate analysis to examine the distribution, the dispersion and central tendency of each variable.

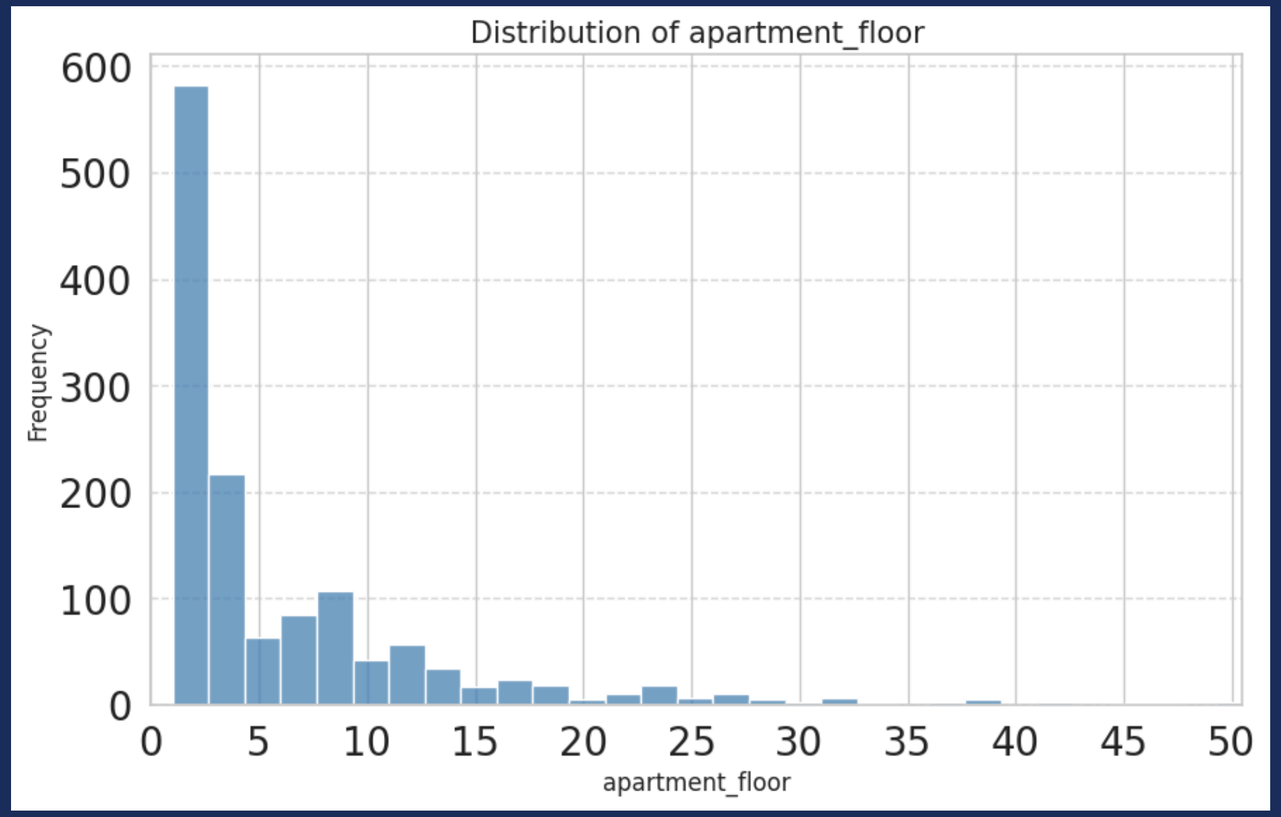
Univariate Analysis



- 2023 saw the highest number of apartments built, with over 400 units — far exceeding other years.
- There is a clear upward trend from 2017 to 2023, indicating a rapid growth in apartment construction over time.

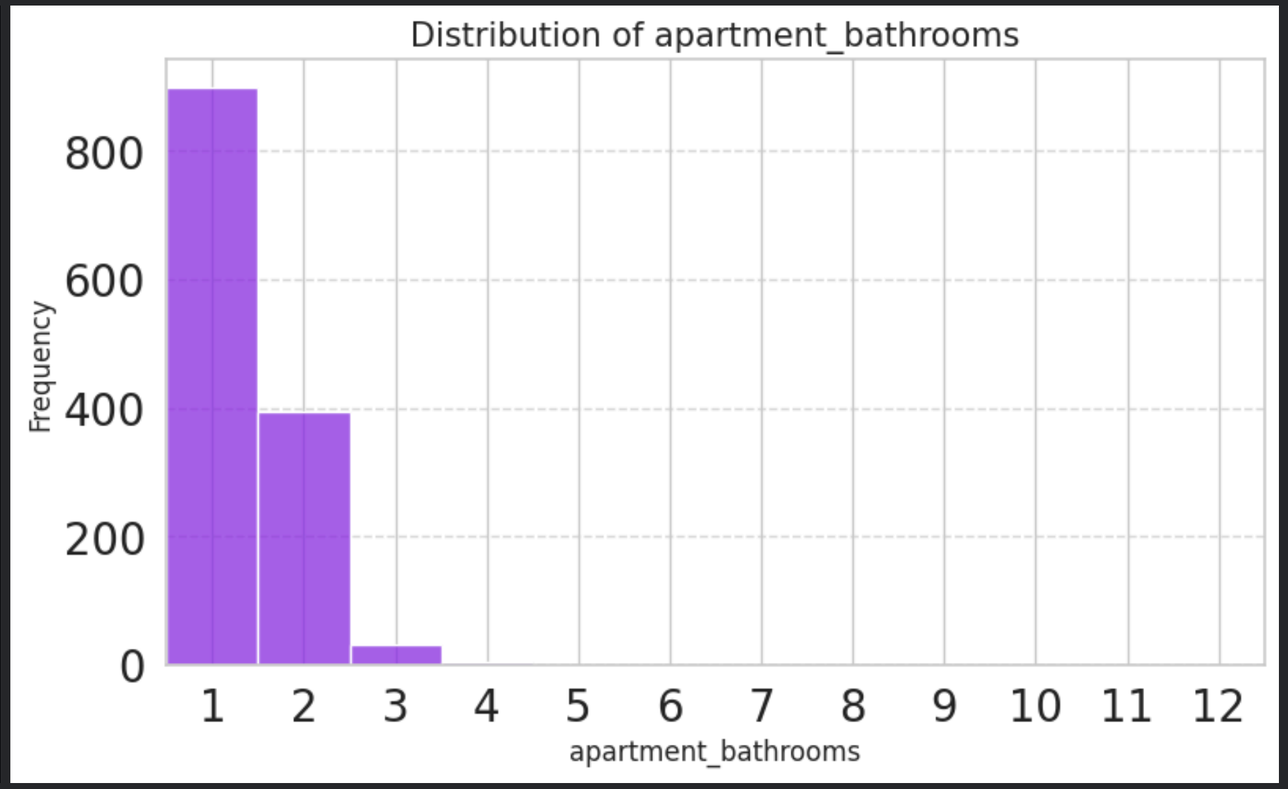
Distribution of building_total_floors and apartment_floor.





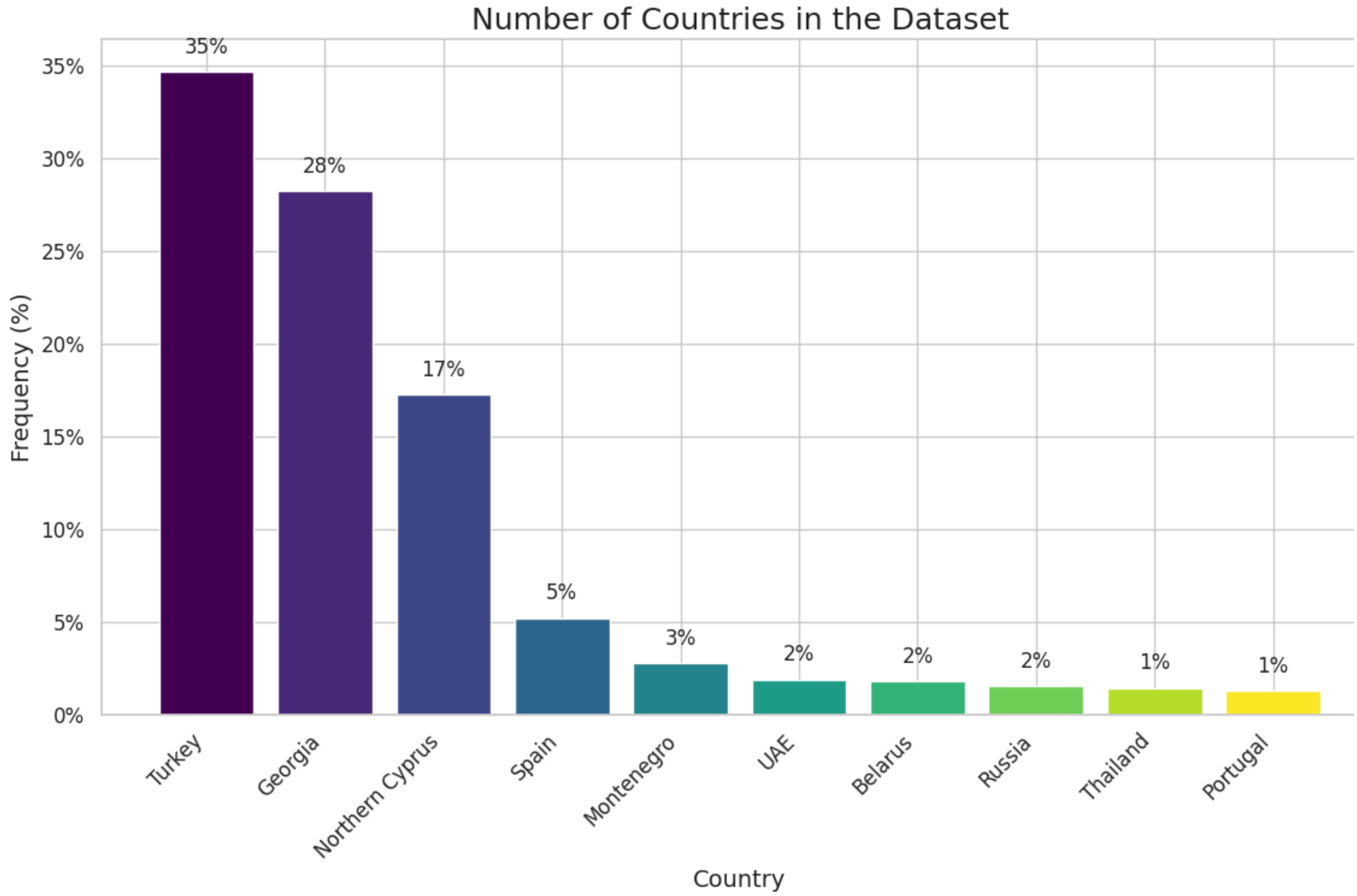
- As you can see here, both variables show right-skewed distributions. Most buildings have under 10 floors, peaking at 5, indicating low-rise buildings are more common. Similarly, most apartments are on lower floors, especially floors 1–5, with very few above the 20th floor.

Distribution of apartment_bedrooms and apartment_bathrooms.



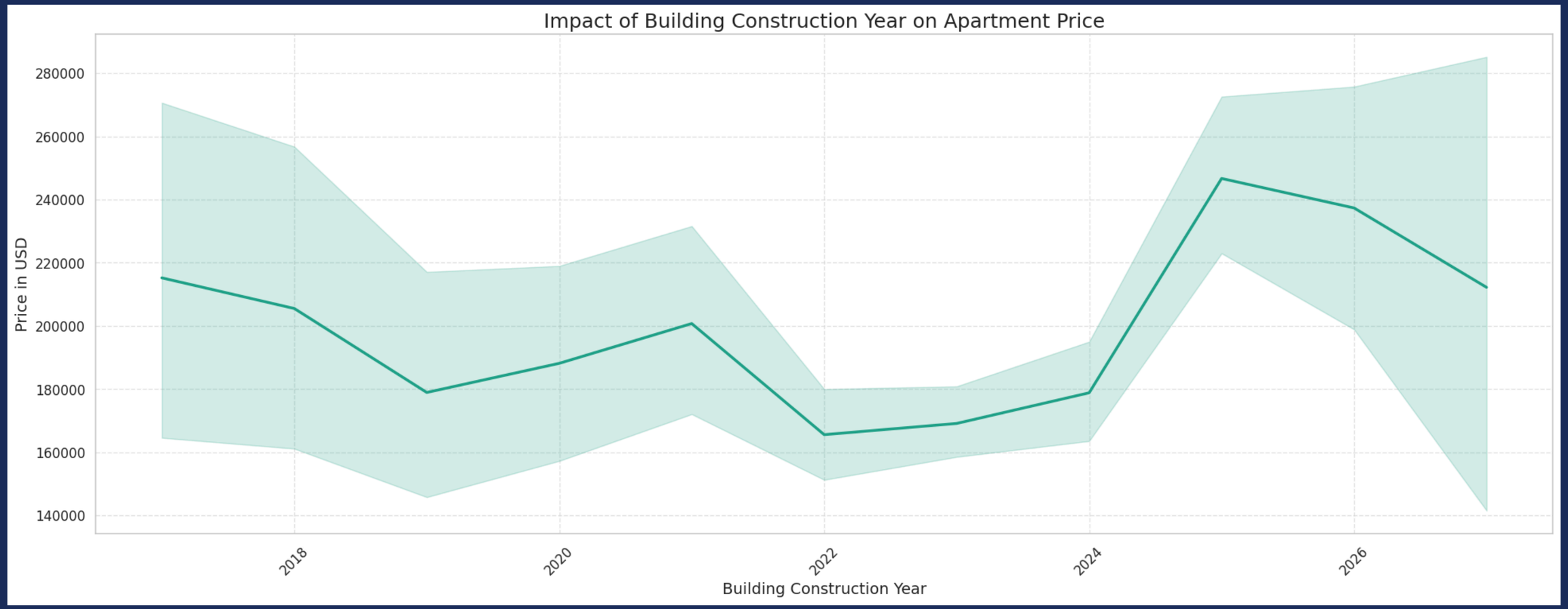
- Most apartments recorded in the dataset have 1 or 2 bedrooms, with 1-bedroom units being the most common. Similarly, the majority of apartments have 1 bathroom, followed by 2 bathrooms. Units with more than 3 bedrooms or bathrooms are very rare.

Countries in the dataset.



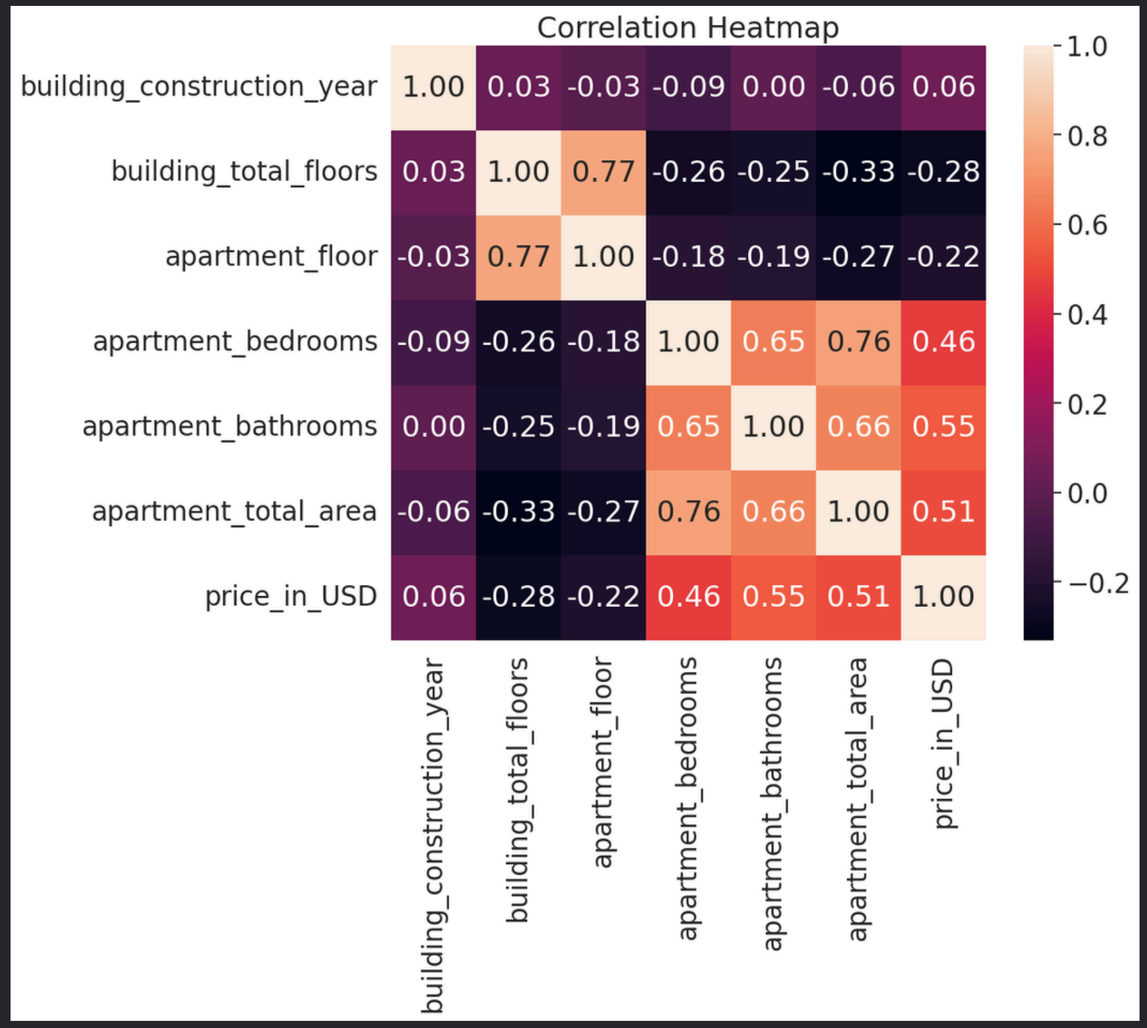
- **Turkey (35%), Georgia (28%), and Northern Cyprus (17%) make up the majority of the dataset, with all other countries representing small portions of 5% and below.**

Multivariate Analysis.



- The line chart above shows that apartment prices vary strongly depending on the building's construction year, with a general upward trend observed in recent years—particularly for buildings constructed around 2025—indicating that newer constructions may come with higher prices in the market.

Correlation between variables



- The strongest correlation is observed between apartment_bedrooms and apartment_total_area (0.76), followed by apartment_bathrooms and apartment_total_area (0.66), and apartment_bedrooms and apartment_bathrooms (0.65). In contrast, building_construction_year shows very weak correlations with other variables, with value close to zero.

Relationship between apartment Price and size.

- This scatter plot shows visually the relationship between apartment size (in square meters) and listing price (in USD). It also suggests a positive correlation, indicating that larger apartments tend to command higher prices. However, the spread of the data points also reveals considerable price variation for apartments of similar sizes, likely reflecting factors beyond size alone. But in short, it's acceptable to say that there's positive linear relationship between 2 variables.

Pairwise relationships of apartment variables

- This pair plot shows the relationships between apartment features. Histograms on the diagonal display individual variable distributions, while scatter plots reveal potential correlations. The "price_in_USD" variable seems to be correlated with "apartment_total_area", and possibly with "apartment_floor".

Remained data after EDA

	building_construction_year	building_total_floors	apartment_floor	apartment_bedrooms	apartment_bathrooms	apartment_total_area	price_in_USD
count	1329.00	1329.00	1329.00	1329.00	1329.00	1329.00	1329.00
mean	2022.82	12.25	6.11	1.67	1.36	84.25	184644.56
std	1.79	11.18	7.24	0.78	0.54	37.21	127110.14
min	2017.00	1.00	1.00	1.00	1.00	26.00	15985.00
25%	2022.00	4.00	1.00	1.00	1.00	56.00	85120.00
50%	2023.00	8.00	3.00	1.00	1.00	76.00	149674.00
75%	2024.00	16.00	8.00	2.00	2.00	105.00	250000.00
max	2027.00	65.00	51.00	5.00	4.00	195.00	614347.00

```
df
```

	country	building_construction_year	building_total_floors	apartment_floor	apartment_bedrooms	apartment_bathrooms	apartment_total_area	price_in_USD
15	Czech Republic	2023.0	4.0	2.0	2.0	1.0	54	314990.0
146	Turkey	2020.0	4.0	4.0	2.0	1.0	93	248071.0
156	Uzbekistan	2025.0	11.0	8.0	1.0	1.0	76	90021.0
424	Turkey	2023.0	3.0	3.0	5.0	2.0	160	344556.0
491	Montenegro	2023.0	8.0	5.0	1.0	1.0	69	336245.0
...	...	...	...	...	...	...	...	...
146346	Indonesia	2024.0	4.0	2.0	1.0	1.0	101	490000.0
146360	Indonesia	2024.0	4.0	2.0	1.0	1.0	81	260000.0
147397	Indonesia	2024.0	4.0	4.0	1.0	1.0	60	140000.0
147408	Indonesia	2025.0	4.0	4.0	2.0	2.0	60	121800.0
147411	Indonesia	2024.0	4.0	4.0	2.0	2.0	88	180000.0

1329 rows × 8 columns

- The table provides a summary of key statistics for each feature, along with a sample of the data entries, giving a clear picture of the data's structure and range.

One-hot encoding

```
df = pd.get_dummies(df, columns=['country'], prefix='country', drop_first=True)
df
```

```
df.info()
'''losing Australia in order to avoid multicollinearity
in nominal variables "Country"'''
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1329 entries, 15 to 147411
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   building_construction_year  1329 non-null   float64
 1   building_total_floors       1329 non-null   float64
 2   apartment_floor             1329 non-null   float64
 3   apartment_bedrooms          1329 non-null   float64
 4   apartment_bathrooms         1329 non-null   float64
 5   apartment_total_area        1329 non-null   int64
 6   price_in_USD                1329 non-null   float64
 7   country_Belarus             1329 non-null   bool
 8   country_Cyprus              1329 non-null   bool
 9   country_Czech Republic      1329 non-null   bool
 10  country_Georgia             1329 non-null   bool
 11  country_Greece              1329 non-null   bool
 12  country_Indonesia           1329 non-null   bool
 13  country_Italy               1329 non-null   bool
 14  country_Latvia              1329 non-null   bool
 15  country_Montenegro          1329 non-null   bool
 16  country_Northern Cyprus     1329 non-null   bool
 17  country_Poland              1329 non-null   bool
 18  country_Portugal            1329 non-null   bool
 19  country_Russia              1329 non-null   bool
 20  country_Spain               1329 non-null   bool
 21  country_Thailand            1329 non-null   bool
 22  country_Turkey              1329 non-null   bool
 23  country_UAE                 1329 non-null   bool
 24  country_Uzbekistan          1329 non-null   bool
dtypes: bool(18), float64(6), int64(1)
memory usage: 106.4 KB
```

```
Number of unique country in the dataset. 19

List of countries and count

country
Turkey              461
Georgia             375
Northern Cyprus     229
Spain                69
Montenegro           37
UAE                  25
Belarus              24
Russia               20
Thailand             19
Portugal             17
Cyprus               15
Indonesia            12
Uzbekistan           11
Poland                5
Latvia                3
Greece                3
Italy                 2
Czech Republic        1
Australia             1
Name: count, dtype: int64
```

- As you can see, I am performing one-hot encoding on the 'country' features using the pd.get_dummies() function in Python. The code creates numerical dummy variables to represent each country, with the drop_first=True argument being used to handle multicollinearity (for n attribute in the categorical variables, we have n - 1 dummy)

Multivariate Regression Analysis.

```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
import statsmodels.api as sm

X = df.drop("price_in_USD", axis=1)
y = df["price_in_USD"]
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=0)

sc = MinMaxScaler()
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.transform(X_test)

#Adding_a_constant_column_to_the_feature_matrix(X)_to_work_with_statsmodels.
X_train_const = sm.add_constant(X_train_scaled)

model = sm.OLS(y_train, X_train_const)

results = model.fit()

print(results.summary())
```

OLS Regression Results

Dep. Variable:	price_in_USD	R-squared:	0.549
Model:	OLS	Adj. R-squared:	0.538
Method:	Least Squares	F-statistic:	47.95
Date:	Fri, 18 Apr 2025	Prob (F-statistic):	3.07e-139
Time:	03:18:16	Log-Likelihood:	-11881.
No. Observations:	930	AIC:	2.381e+04
Df Residuals:	906	BIC:	2.393e+04
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.531e+05	1.45e+04	10.593	0.000	1.25e+05	1.81e+05
x1	1.965e+04	1.79e+04	1.101	0.271	-1.54e+04	5.47e+04
x2	-7.098e+04	2.77e+04	-2.561	0.011	-1.25e+05	-1.66e+04
x3	8.526e+04	3.15e+04	2.708	0.007	2.35e+04	1.47e+05
x4	89.1777	2.63e+04	0.003	0.997	-5.16e+04	5.18e+04
x5	1.429e+05	2.32e+04	6.158	0.000	9.74e+04	1.88e+05
x6	2.048e+05	2.42e+04	8.480	0.000	1.57e+05	2.52e+05
x7	-1.121e+05	2.6e+04	-4.312	0.000	-1.63e+05	-6.11e+04
x8	-8.598e+04	2.81e+04	-3.058	0.002	-1.41e+05	-3.08e+04
x9	1.177e+05	8.28e+04	1.423	0.155	-4.47e+04	2.8e+05
x10	-1.05e+05	1.21e+04	-8.655	0.000	-1.29e+05	-8.12e+04
x11	1.616e+05	8.32e+04	1.944	0.052	-1552.927	3.25e+05
x12	4.254e+04	3.11e+04	1.369	0.171	-1.85e+04	1.04e+05
x13	-1.899e+04	8.27e+04	-0.230	0.818	-1.81e+05	1.43e+05
x14	-1.028e+04	5.94e+04	-0.173	0.863	-1.27e+05	1.06e+05
x15	-7411.7460	1.9e+04	-0.390	0.697	-4.47e+04	2.99e+04
x16	-4.623e+04	1.26e+04	-3.683	0.000	-7.09e+04	-2.16e+04
x17	6.299e+04	4.85e+04	1.298	0.195	-3.23e+04	1.58e+05
x18	1.359e+05	2.5e+04	5.433	0.000	8.68e+04	1.85e+05
x19	1.721e+05	2.4e+04	7.157	0.000	1.25e+05	2.19e+05
x20	3.865e+04	1.59e+04	2.425	0.016	7370.539	6.99e+04
x21	-3.018e+04	2.42e+04	-1.248	0.212	-7.76e+04	1.73e+04
x22	-8.918e+04	1.12e+04	-7.950	0.000	-1.11e+05	-6.72e+04
x23	6.038e+04	2.31e+04	2.609	0.009	1.5e+04	1.06e+05
x24	-1.335e+05	3.29e+04	-4.062	0.000	-1.98e+05	-6.9e+04

Omnibus:	220.057	Durbin-Watson:	1.932
Prob(Omnibus):	0.000	Jarque-Bera (JB):	559.435
Skew:	1.229	Prob(JB):	3.31e-122
Kurtosis:	5.897	Cond. No.	8.43e+15

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.37e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

- The initial OLS regression model explains about 54.9% of the variance in price_in_USD. While several predictors are statistically significant ($p < 0.05$), the high condition number and strong multicollinearity suggest potential instability. Removing insignificant variables is recommended to enhance the model's reliability and interpretability

With building_total_floors and apartment_bedrooms removed.

```
<class 'pandas.core.frame.DataFrame'>
Index: 1329 entries, 15 to 147411
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   building_total_floors     1329 non-null   float64
 1   apartment_floor           1329 non-null   float64
 2   apartment_bathrooms       1329 non-null   float64
 3   apartment_total_area      1329 non-null   int64
 4   price_in_USD              1329 non-null   float64
 5   country_Belarus           1329 non-null   bool
 6   country_Cyprus            1329 non-null   bool
 7   country_Czech Republic    1329 non-null   bool
 8   country_Georgia           1329 non-null   bool
 9   country_Greece            1329 non-null   bool
 10  country_Indonesia         1329 non-null   bool
 11  country_Italy             1329 non-null   bool
 12  country_Latvia            1329 non-null   bool
 13  country_Montenegro        1329 non-null   bool
 14  country_Northern Cyprus   1329 non-null   bool
 15  country_Poland            1329 non-null   bool
 16  country_Portugal          1329 non-null   bool
 17  country_Russia            1329 non-null   bool
 18  country_Spain             1329 non-null   bool
 19  country_Thailand          1329 non-null   bool
 20  country_Turkey            1329 non-null   bool
 21  country_UAE               1329 non-null   bool
 22  country_Uzbekistan        1329 non-null   bool
dtypes: bool(18), float64(4), int64(1)
memory usage: 85.7 KB
```

OLS Regression Results

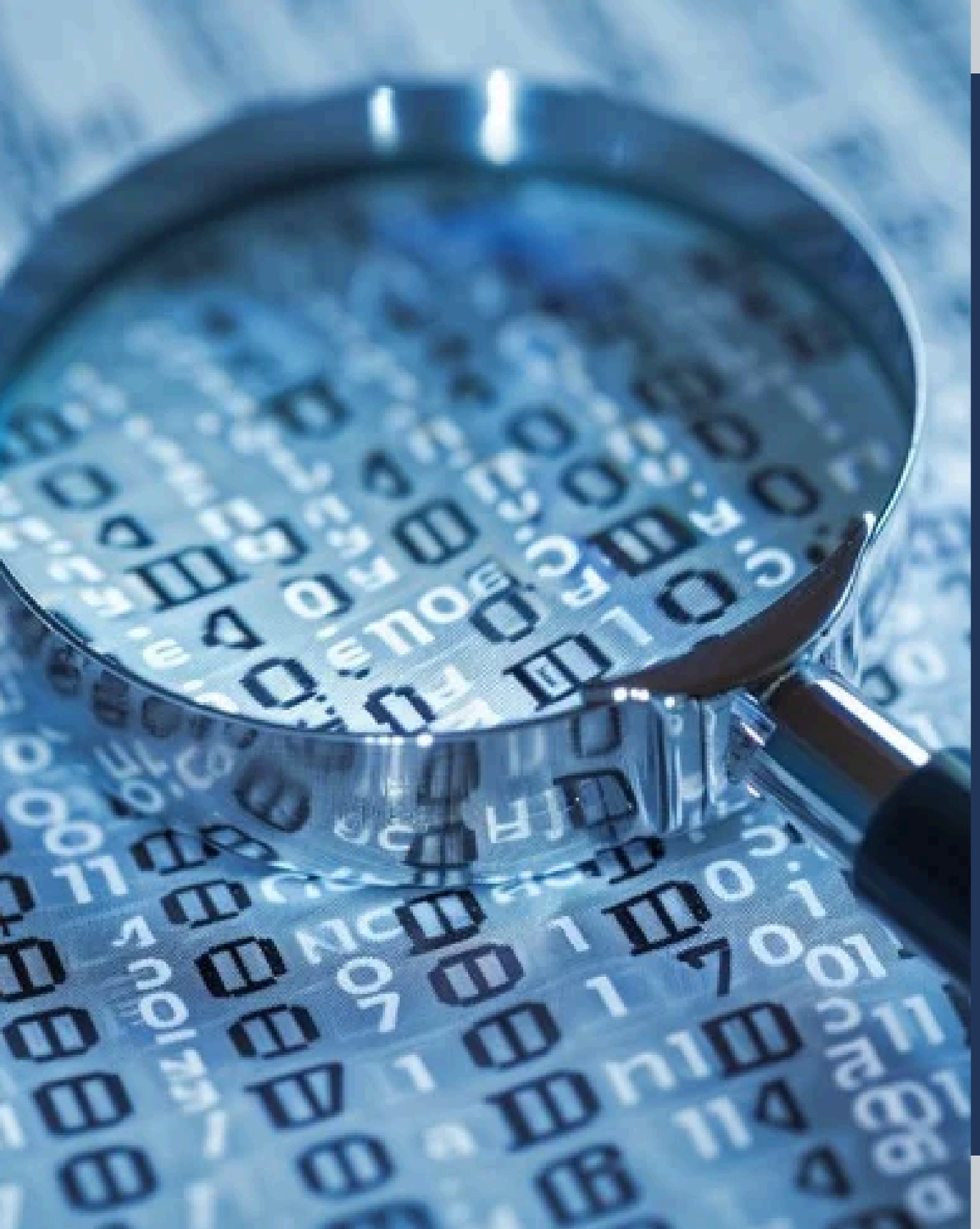
Dep. Variable:	price_in_USD	R-squared:	0.540
Model:	OLS	Adj. R-squared:	0.531
Method:	Least Squares	F-statistic:	55.58
Date:	Fri, 18 Apr 2025	Prob (F-statistic):	2.60e-158
Time:	03:18:16	Log-Likelihood:	-13591.
No. Observations:	1063	AIC:	2.723e+04
Df Residuals:	1040	BIC:	2.734e+04
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.031e+05	8.8e+04	4.583	0.000	2.31e+05	5.76e+05
x1	-5.404e+04	2.6e+04	-2.081	0.038	-1.05e+05	-3072.156
x2	7.5e+04	2.95e+04	2.546	0.011	1.72e+04	1.33e+05
x3	1.488e+05	2.16e+04	6.895	0.000	1.06e+05	1.91e+05
x4	1.957e+05	1.78e+04	11.016	0.000	1.61e+05	2.31e+05
x5	-3.535e+05	9.04e+04	-3.911	0.000	-5.31e+05	-1.76e+05
x6	-3.209e+05	9.11e+04	-3.524	0.000	-5e+05	-1.42e+05
x7	-1.195e+05	1.24e+05	-0.965	0.335	-3.62e+05	1.23e+05
x8	-3.445e+05	8.77e+04	-3.930	0.000	-5.17e+05	-1.72e+05
x9	-7.781e+04	1.24e+05	-0.628	0.530	-3.21e+05	1.65e+05
x10	-1.849e+05	9.19e+04	-2.013	0.044	-3.65e+05	-4636.014
x11	-3.036e+05	1.07e+05	-2.829	0.005	-5.14e+05	-9.31e+04
x12	-2.573e+05	1.07e+05	-2.403	0.016	-4.67e+05	-4.72e+04
x13	-2.432e+05	8.91e+04	-2.730	0.006	-4.18e+05	-6.84e+04
x14	-2.848e+05	8.78e+04	-3.245	0.001	-4.57e+05	-1.13e+05
x15	-1.781e+05	1.01e+05	-1.761	0.078	-3.76e+05	2.03e+04
x16	-1.011e+05	9.08e+04	-1.113	0.266	-2.79e+05	7.71e+04
x17	-6.623e+04	9.03e+04	-0.733	0.464	-2.43e+05	1.11e+05
x18	-1.969e+05	8.82e+04	-2.232	0.026	-3.7e+05	-2.38e+04
x19	-2.724e+05	9.03e+04	-3.016	0.003	-4.5e+05	-9.52e+04
x20	-3.245e+05	8.76e+04	-3.705	0.000	-4.96e+05	-1.53e+05
x21	-1.797e+05	8.95e+04	-2.006	0.045	-3.55e+05	-3962.476
x22	-3.719e+05	9.23e+04	-4.028	0.000	-5.53e+05	-1.91e+05

Omnibus:	262.518	Durbin-Watson:	1.921
Prob(Omnibus):	0.000	Jarque-Bera (JB):	703.461
Skew:	1.268	Prob(JB):	1.76e-153
Kurtosis:	6.074	Cond. No.	171.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

- After using backward elimination, removing building_total_floors (0.2) and apartment_bedrooms (0.997), the model is statistically significant (F-test p-value is very small) and explains about 53.1% of the price_in_USD variance. This is a reasonable good result, especially considering the complex qualitative nature of the real estate data



CONCLUSION

R-squared: 0.540, Adjusted R-squared: 0.531

- We can say that approximately 54% of the variance in apartment prices can be explained by the independent variables included in the model and this is a good reasonable fit.

F-statistic: 55.58, p-value: < 0.0001

- The model is statistically significant overall

Statistically insignificant coefficients for Poland, Portugal, and Russia suggest prices in these markets aren't reliably distinguishable from Australia's, likely due to data variability.

Top 3 variables affecting apartment prices:

- apartment_total_area
- building_total_floors
- apartment_bathrooms

About the countries dummy

Country	Coefficient	p-value	Implication
Belarus	-353,500	<0.001	Significantly cheaper than Australia
Cyprus	-320,900	<0.001	Significantly cheaper
Georgia	-344,500	<0.001	Significantly cheaper
Indonesia	-184,900	0.044	Cheaper, moderately significant
Italy	-303,600	0.005	Cheaper
Latvia	-257,300	0.016	Cheaper
Montenegro	-243,200	0.006	Cheaper
Northern Cyprus	-284,800	0.001	Cheaper

Country	Coefficient	p-value	Implication
Poland	**-178,100**	**0.078**	**Marginally insignificant**
Portugal	-101,100	0.266	Not significant
Russia	-66,230	0.464	Not significant
Spain	-196,900	0.026	Significant difference
Thailand	-272,400	0.003	Significant difference
Turkey	-324,500	<0.001	Large difference
UAE	-179,700	0.045	Slightly significant
Uzbekistan	-371,900	<0.001	Very large and significant difference

- **Countries such as Poland, Portugal, and Russia show coefficients that are statistically insignificant. This suggests prices in these markets might not be meaningfully different from Australia — or more likely, there's too much variation in the data to conclude confidently.**

Recommendations.

Data-driven decisions that can be informed based on the multiple linear regression model.

FOR IMPROVING MODELFIT

- We can execute log-transforming for the dependent variable “price_in_USD” to deal with skewed distribution and improve residual normality.
- Check for multicollinearity and remove or combine countries with insignificant coefficients.

FOR BUSINESS STRATEGY

Country Impact:

- Apartments in Australia are, on average, the most expensive across the dataset.
- Countries like Uzbekistan, Turkey, and Georgia offer significantly cheaper apartment prices, which may indicate attractive investment opportunities for budget-sensitive buyers.

Property Features:

- Apartment size and bathroom count are strong predictors of price — as expected.
- Higher-floor apartments are seen as more valuable, while taller buildings overall are associated with lower unit prices — possibly due to being in more densely packed areas or lower per-floor value.

⟶ The multivariate regression model shows that apartment prices are mainly driven by **total area** and **number of bathrooms**, which should guide property design and marketing. While higher-floor units add value, too many building floors may lower prices. Countries like **Turkey**, **Georgia**, and **Uzbekistan** have significantly lower prices than **Australia**, suggesting investment opportunities in affordable markets. Pricing should be localized, and investors should focus on undervalued locations with strong features.

THANK YOU