1.5. Housing in Brazil 🇧🇷

In [262…
```
# Import Matplotlib, pandas, and plotly
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
```

Task 1.5.1

In [263…
```
df1 = pd.read_csv("data/brasil-real-estate-1.csv")
```

In [264…
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12834 entries, 0 to 12833
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   property_type           12834 non-null  object 
 1   place_with_parent_names  12834 non-null  object 
 2   region                  12834 non-null  object 
 3   lat-lon                 11551 non-null  object 
 4   area_m2                 12834 non-null  float64
 5   price_usd               12834 non-null  object 
dtypes: float64(1), object(5)
memory usage: 601.7+ KB
```

In [265…
```
df1.dropna(inplace=True)
```

Task 1.5.2

In [266…
```
df1.head()
```

Out[266…

	property_type	place_with_parent_names	region	lat-lon	are
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	

In [267…
```
df1.describe()
```

Out[267…

	area_m2
count	11551.000000
mean	116.695264
std	48.186630
min	53.000000
25%	78.000000
50%	105.000000
75%	145.000000
max	252.000000

Task 1.5.3

In [268…
```
df1[["lat", "lon"]] = df1['lat-lon'].str.split(',',expand =True).astype(f
```

In [269…
```
df1.head(10)
```

Out[269…

	property_type	place_with_parent_names	region	lat-lon	ar
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	
5	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.614414,-35.735621	
6	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.584755,-35.662909	
7	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.658285,-35.703827	
9	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.66082,-35.702976	
10	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6637998,-35.7115455	

Task 1.5.4

In [270…
```
df1["state"] = df1['place_with_parent_names'].str.split('|',expand=True)[

df1.drop(columns=['place_with_parent_names','lat-lon'],inplace=True)
df1.head()
```

Out[270…

	property_type	region	area_m2	price_usd	lat	lon	state
0	apartment	Northeast	110.0	$187,230.85	-9.644305	-35.708814	Alagoas
1	apartment	Northeast	65.0	$81,133.37	-9.643093	-35.704840	Alagoas
2	house	Northeast	211.0	$154,465.45	-9.622703	-35.729795	Alagoas
3	apartment	Northeast	99.0	$146,013.20	-9.622837	-35.719556	Alagoas
4	apartment	Northeast	55.0	$101,416.71	-9.654955	-35.700227	Alagoas

Task 1.5.5

In [271…

```
df1["price_usd"] = df1["price_usd"].str.replace(r'[^0-9.]',"",regex=True)
df1.head()
```

Out[271…

	property_type	region	area_m2	price_usd	lat	lon	state
0	apartment	Northeast	110.0	187230.85	-9.644305	-35.708814	Alagoas
1	apartment	Northeast	65.0	81133.37	-9.643093	-35.704840	Alagoas
2	house	Northeast	211.0	154465.45	-9.622703	-35.729795	Alagoas
3	apartment	Northeast	99.0	146013.20	-9.622837	-35.719556	Alagoas
4	apartment	Northeast	55.0	101416.71	-9.654955	-35.700227	Alagoas

Task 1.5.6

In [ ]:

```
```

Task 1.5.7

In [272…

```
df2 = pd.read_csv("data/brasil-real-estate-2.csv")
df2.head()
```

Out[272…

	property_type	state	region	lat	lon	area_m2	price_
0	apartment	Pernambuco	Northeast	-8.134204	-34.906326	72.0	414222
1	apartment	Pernambuco	Northeast	-8.126664	-34.903924	136.0	848408
2	apartment	Pernambuco	Northeast	-8.125550	-34.907601	75.0	299438
3	apartment	Pernambuco	Northeast	-8.120249	-34.895920	187.0	848408
4	apartment	Pernambuco	Northeast	-8.142666	-34.906906	80.0	464129

In [273…

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12833 entries, 0 to 12832
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   property_type  12833 non-null  object 
 1   state          12833 non-null  object 
 2   region         12833 non-null  object 
 3   lat            12833 non-null  float64
 4   lon            12833 non-null  float64
 5   area_m2        11293 non-null  float64
 6   price_brl      12833 non-null  float64
dtypes: float64(4), object(3)
memory usage: 701.9+ KB
```

Task 1.5.8

In [274…
```
df2["price_usd"] = df2['price_brl']/3.19
```

Task 1.5.9

In [275…
```
df2.drop(columns='price_brl',inplace=True)
```

In [276…
```
df2.dropna(inplace=True)
```

Task 1.5.10

In [277…
```
df = pd.concat([df1,df2])
print("df shape:", df.shape)
```

```
df shape: (22844, 7)
```

In [278…
```
df.head()
```

Out[278…

	property_type	region	area_m2	price_usd	lat	lon	state
0	apartment	Northeast	110.0	187230.85	-9.644305	-35.708814	Alagoas
1	apartment	Northeast	65.0	81133.37	-9.643093	-35.704840	Alagoas
2	house	Northeast	211.0	154465.45	-9.622703	-35.729795	Alagoas
3	apartment	Northeast	99.0	146013.20	-9.622837	-35.719556	Alagoas
4	apartment	Northeast	55.0	101416.71	-9.654955	-35.700227	Alagoas

Explore

In [279…
```
fig = px.scatter_mapbox(
    df,
    lat=df['lat'],
    lon=df['lon'],
    center={"lat": -14.2, "lon": -51.9},  # Map will be centered on Brazi
    width=600,
    height=600,
    hover_data=["price_usd"],  # Display price when hovering mouse over h
)
```

```
fig.update_layout(mapbox_style="open-street-map")

fig.show()
```

Task 1.5.11

In [280…
```
summary_stats = ...
summary_stats
```

Out[280… `Ellipsis`

Slight Code Change

In the following task, you'll notice a small change in how plots are created compared to what you saw in the lessons. While the lessons use the global matplotlib method like `plt.plot(...)`, in this task, you are expected to use the object-oriented (OOP) API instead. This means creating your plots using `fig, ax = plt.subplots()` and then calling plotting methods on the `ax` object, such as `ax.plot(...)`, `ax.hist(...)`, or `ax.scatter(...)`.

If you're using pandas' or seaborn's built-in plotting methods (like `df.plot()` or `sns.lineplot()`), make sure to pass the `ax=ax` argument so that the plot is rendered on the correct axes.

This approach is considered best practice and will be used consistently across all graded tasks that involve matplotlib.

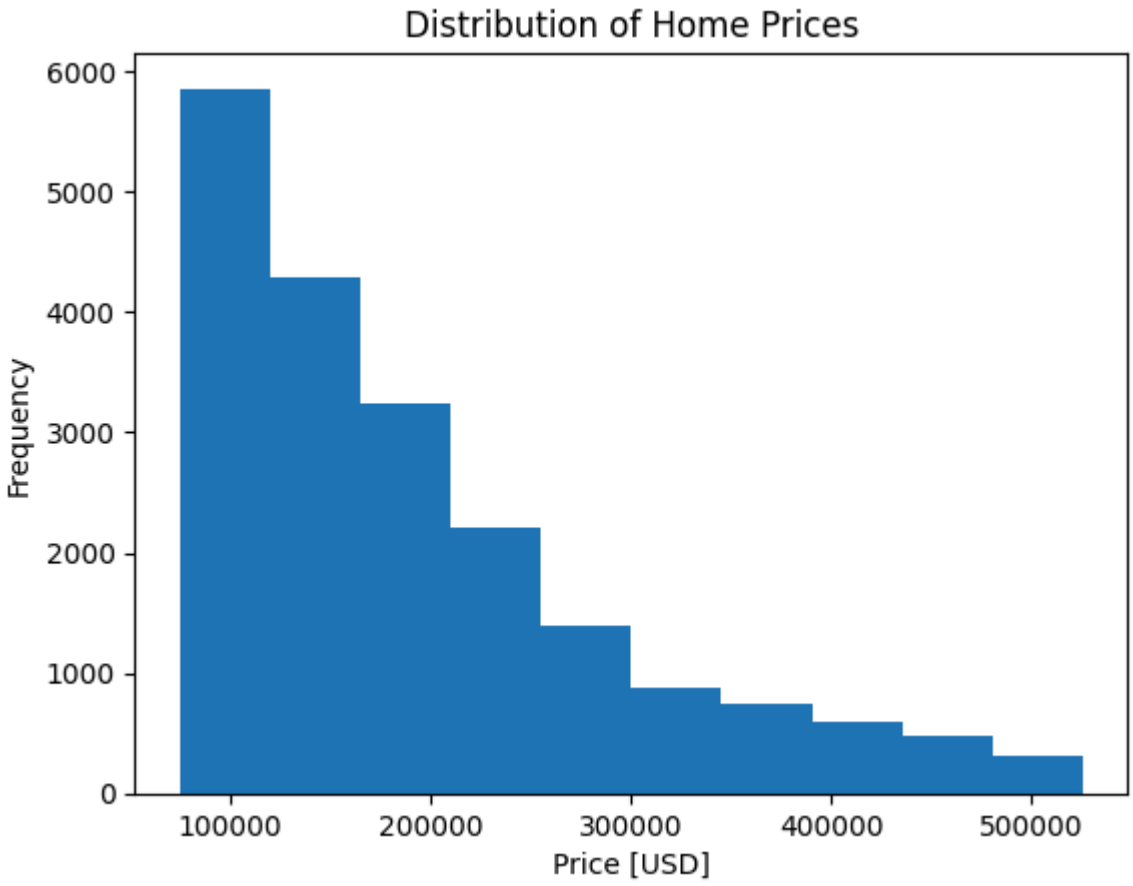
Task 1.5.12

In [281…
```
# Don't change the code below 👇
fig, ax = plt.subplots()

# Build histogram
ax.hist(df['price_usd'][:20000])

# Label axes
plt.xlabel('Price [USD]')
plt.ylabel('Frequency')


# Add title
plt.title('Distribution of Home Prices')
plt.show()
```



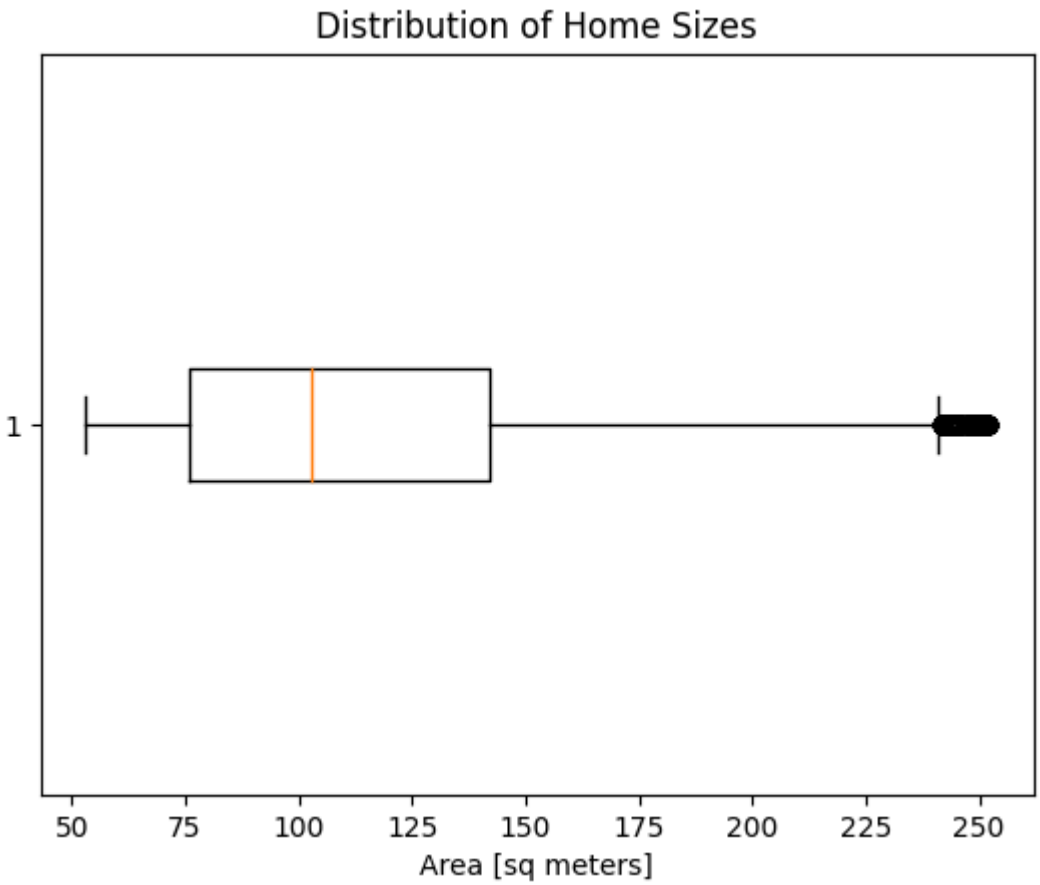
Task 1.5.13

In [282…
```
# Don't change the code below 👇
fig, ax = plt.subplots()
```

```
#Build box plot
ax.boxplot(df['area_m2'],vert=False)

# Label x-axis
plt.xlabel('Area [sq meters]')
plt.title('Distribution of Home Sizes')
# Add title
```

Out[282… `Text(0.5, 1.0, 'Distribution of Home Sizes')`



Task 1.5.14

In [283…
```
mean_price_by_region = df.groupby('region')['price_usd'].mean().sort_valu
mean_price_by_region
```

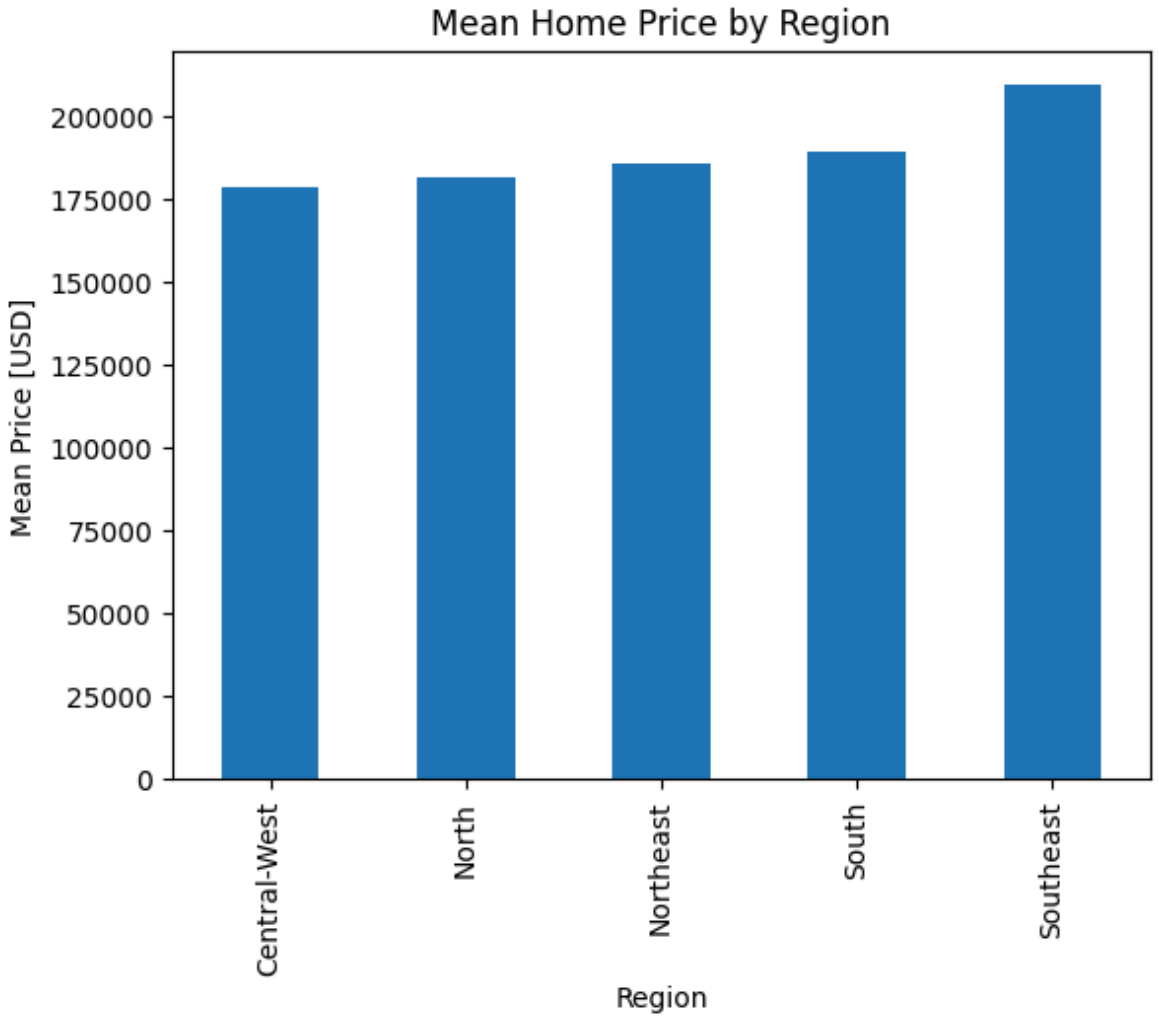
Out[283…
```
region
Central-West    178596.283663
North           181308.958207
Northeast       185422.985441
South           189012.345265
Southeast       208996.762778
Name: price_usd, dtype: float64
```

Task 1.5.15

In [284…
```
# Don't change the code below 👇
fig, ax = plt.subplots()

# Build bar chart, label axes, add title
mean_price_by_region.plot(kind='bar',xlabel='Region',ylabel='Mean Price [
```

Out[284… <Axes: title={'center': 'Mean Home Price by Region'}, xlabel='Region', ylabel='Mean Price [USD]'>



Task 1.5.16

In [285…
```
df_south = df[df['region']=='South']
df_south.head()
```

Out[285…

	property_type	region	area_m2	price_usd	lat	lon	state
9304	apartment	South	127.0	296448.85	-25.455704	-49.292918	Paraná
9305	apartment	South	104.0	219996.25	-25.455704	-49.292918	Paraná
9306	apartment	South	100.0	194210.50	-25.460236	-49.293812	Paraná
9307	apartment	South	77.0	149252.94	-25.460236	-49.293812	Paraná
9308	apartment	South	73.0	144167.75	-25.460236	-49.293812	Paraná

Task 1.5.17

In [286…
```
homes_by_state = df_south['state'].value_counts()
homes_by_state
```

Out[286…
```
Rio Grande do Sul    2643
Santa Catarina       2634
Paraná               2544
Name: state, dtype: int64
```

Task 1.5.18

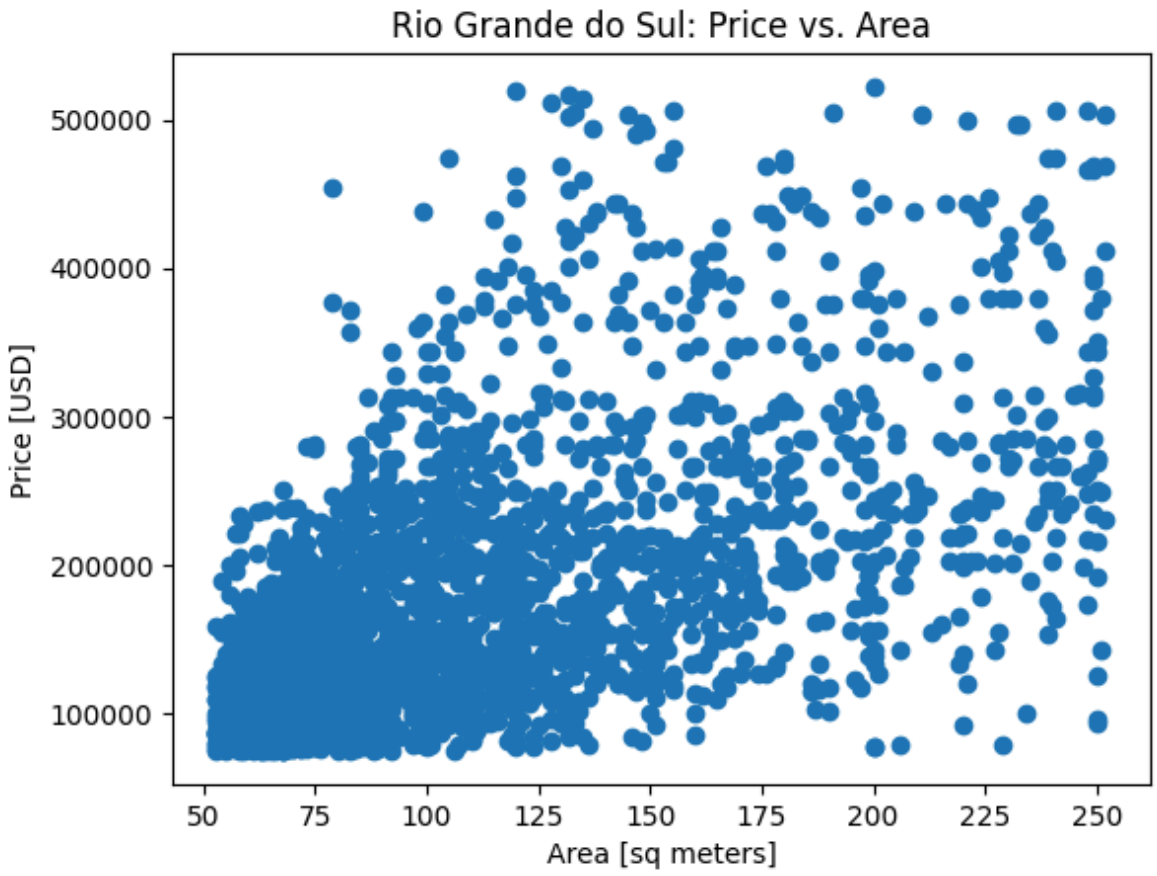
In [287…
```
# Subset data
df_south_rgs = df_south[df_south['state']=='Rio Grande do Sul']

# Don't change the code below 👇
fig, ax = plt.subplots()


# Build scatter plot
ax.scatter(df_south_rgs['area_m2'],df_south_rgs['price_usd'])

# Label axes
plt.xlabel("Area [sq meters]")
plt.ylabel("Price [USD]")
# Add title
plt.title("Rio Grande do Sul: Price vs. Area")
```

Out[287…
```
Text(0.5, 1.0, 'Rio Grande do Sul: Price vs. Area')
```



Task 1.5.19

In [288…
```
south_states_corr = {
    'Espírito Santo': 0.6311332554173303,
    'Minas Gerais': 0.5830029036378931,
    'Rio de Janeiro': 0.4554077103515366,
    'São Paulo': 0.45882050624839366
```

```
}

df_south = df[df['region'] == 'South']
south_states = df_south['state'].unique()

south_states_corr = {}
for state in south_states:
    df_state = df_south[df_south['state'] == state]
    corr = df_state['price_usd'].corr(df_state['area_m2'])
    south_states_corr[state] = corr

south_states_corr
```

Out[288…

```
{'Paraná': 0.5436659935502657,
 'Rio Grande do Sul': 0.5773267433717685,
 'Santa Catarina': 0.5068121776366781}
```

In [ ]: