Assignment: Housing in Brazil 🇧🇷

In [1]:
```python
import wqet_grader

wqet_grader.init("Project 1 Assessment")
```

In this assignment, you'll work with a dataset of homes for sale in Brazil. Your goal is to determine if there are regional differences in the real estate market. Also, you will look at southern Brazil to see if there is a relationship between home size and price, similar to what you saw with housing in some states in Mexico.

Before you start: Import the libraries you'll use in this notebook: Matplotlib, pandas, and plotly. Be sure to import them under the aliases we've used in this project.

In [73]:
```python
# Import Matplotlib, pandas, and plotly
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
```

Prepare Data

In this assignment, you'll work with real estate data from Brazil. In the `data` directory for this project there are two CSV that you need to import and clean.

Import

Task 1.5.1: Import the CSV file `data/brasil-real-estate-1.csv` into the DataFrame `df1`.

In [8]:
```python
#Import CSV file to create dataframe
df1 = pd.read_csv('data/brasil-real-estate-1.csv')
#Inspect dataset with the 'info' and 'head' methods
print(df1.info())
print(df1.head())
```

```
<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
None
  property_type  place_with_parent_names     region                 lat-lon  \
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

   area_m2    price_usd
0    110.0  $187,230.85
1     65.0   $81,133.37
2    211.0  $154,465.45
3     99.0  $146,013.20
4     55.0  $101,416.71
```

In [9]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.1", df1)
```

Good work!
Score: 1

Before you move to the next task, take a moment to inspect `df1` using the `info` and `head` methods. What issues do you see in the data? What cleaning will you need to do before you can conduct your analysis?

Five features contain null values. The 'lat-lon' and 'price_usd' features contain string stored as objects. All the values in the 'price_usd' feature contain the dollar sign and commas. The null values need to be dropped. The object datatypes of 'lat-lon' and 'price_usd' features need to be converted to float datatype. All values in the 'price_usd' feature will need to be stripped off the dollar sign and commas.

Task 1.5.2: Drop all rows with `NaN` values from the DataFrame `df1`.

In [11]:
```python
#Drop all rows with null values
df1.dropna(inplace=True)
#Inspect dataframe to see new dataset
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11551 entries, 0 to 12833
Data columns (total 6 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   property_type           11551 non-null  object
 1   place_with_parent_names 11551 non-null  object
 2   region                  11551 non-null  object
 3   lat-lon                 11551 non-null  object
 4   area_m2                 11551 non-null  float64
 5   price_usd               11551 non-null  object
dtypes: float64(1), object(5)
memory usage: 631.7+ KB
```

In [12]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.2", df1)
```

Excellent! Keep going.
Score: 1

Task 1.5.3: Use the `"lat-lon"` column to create two separate columns in `df1`: `"lat"` and `"lon"`. Make sure that the data type for these new columns is `float`.

In [20]:
```python
#Split the 'lat-lon' column to create 'lat' and 'lon' columns
df1[['lat' , 'lon']] = (
    df1['lat-lon']
    .str.split(',' , expand=True)
    .astype(float)
)
#Print dataset to view new features
df1.head()
```

Out[20]:

	property_type	place_with_parent_names	region	lat-lon	area_m2	price_usd	lat	lon
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	110.0	$187,230.85	-9.644305	-35.708814
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	65.0	$81,133.37	-9.643093	-35.704840
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	211.0	$154,465.45	-9.622703	-35.729795
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	99.0	$146,013.20	-9.622837	-35.719556
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	55.0	$101,416.71	-9.654955	-35.700227

In [21]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.3", df1)
```

Yes! Your hard work is paying off.
Score: 2

Task 1.5.4: Use the `"place_with_parent_names"` column to create a `"state"` column for `df1`. (Note that the state name always appears after `"|Brasil|"` in each string.)

In [25]:
```python
#Create the 'state' column from the 'place_with_parent_names' column
df1['state'] = (
    df1['place_with_parent_names']
    .str.split('|' , expand=True)[2]
)
#Print dataset to view new feature
df1.head()
```

Out[25]:

	property_type	place_with_parent_names	region	lat-lon	area_m2	price_usd	lat	lon	state
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	110.0	$187,230.85	-9.644305	-35.708814	Alagoas
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	65.0	$81,133.37	-9.643093	-35.704840	Alagoas
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	211.0	$154,465.45	-9.622703	-35.729795	Alagoas
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	99.0	$146,013.20	-9.622837	-35.719556	Alagoas
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	55.0	$101,416.71	-9.654955	-35.700227	Alagoas

In [26]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.4", df1)
```

You = coding 🥷
Score: 1

Task 1.5.5: Transform the `"price_usd"` column of `df1` so that all values are floating-point numbers instead of strings.

In [32]:
```python
#Convert all values in the 'price_usd' column to float removing the dollar sign and commas
df1['price_usd'] = (df1['price_usd']
                    .str.replace('$',"")
                    .str.replace(',',"")
                    .astype(float)
)
#Inspect dataframe to view new dataset
df1.head()
```

Out[32]:

	property_type	place_with_parent_names	region	lat-lon	area_m2	price_usd	lat	lon	state
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	110.0	187230.85	-9.644305	-35.708814	Alagoas
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	65.0	81133.37	-9.643093	-35.704840	Alagoas
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	211.0	154465.45	-9.622703	-35.729795	Alagoas
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	99.0	146013.20	-9.622837	-35.719556	Alagoas
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	55.0	101416.71	-9.654955	-35.700227	Alagoas

In [33]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.5", df1)
```

That's the right answer. Keep it up!
Score: 1

Task 1.5.6: Drop the `"lat-lon"` and `"place_with_parent_names"` columns from `df1`.

In [36]:
```python
#Drop the 'lat-lon' and 'place_with_parent_names' columns
df1.drop(columns=['lat-lon' , 'place_with_parent_names'] , inplace=True)
#Inspect dataframe to see new dataset
df1.head()
```

Out[36]:

	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

In [37]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.6", df1)
```

You = coding 🥷
Score: 1.0

Good work! You're halfway through your data wrangling. Take a break: Get up from your machine and stretch. 😁

Task 1.5.7: Import the CSV file `brasil-real-estate-2.csv` into the DataFrame `df2`.

In [55]:
```python
#Import CSV file to create dataframe
df2 = pd.read_csv('data/brasil-real-estate-2.csv')
#Inspect dataset
print(df2.info())
print(df2.head())
```

```
<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
None
  property_type       state     region       lat        lon  area_m2  \
0     apartment  Pernambuco  Northeast -8.134204 -34.906326     72.0
1     apartment  Pernambuco  Northeast -8.126664 -34.903924    136.0
2     apartment  Pernambuco  Northeast -8.125550 -34.907601     75.0
3     apartment  Pernambuco  Northeast -8.120249 -34.895920    187.0
4     apartment  Pernambuco  Northeast -8.142666 -34.906906     80.0

   price_brl
0  414222.98
1  848408.53
2  299438.28
3  848408.53
4  464129.36
```

In [40]:
```python
wqet_grader.grade(
    "Project 1 Assessment", "Task 15.7", df2.sort_values("price_brl").head()
)
```

Good work!
Score: 1.0

Before you jump to the next task, take a look at `df2` using the `info` and `head` methods. What issues do you see in the data? How is it similar or different from `df1`?

Six features (columns) contain null values. The currency of the prices in the 'price_brl' column are in Brazilian Real. The second dataset is different from the first in that all numeric values are stored as float datatype. The values in the 'price_brl' column does not contain special characters. The 'state' column contains only one set string values.

Task 1.5.8: Use the `"price_brl"` column to create a new column named `"price_usd"`. (Keep in mind that, when this data was collected in 2015 and 2016, a US dollar cost 3.19 Brazilian reals.)

In [56]:
```python
#Create the 'price_usd' column from the 'price_brl' column
df2['price_usd'] = df2['price_brl'] / 3.19
#Print dataframe to view new dataset
df2.head()
```

Out[56]:

	property_type	state	region	lat	lon	area_m2	price_brl	price_usd
0	apartment	Pernambuco	Northeast	-8.134204	-34.906326	72.0	414222.98	129850.463950
1	apartment	Pernambuco	Northeast	-8.126664	-34.903924	136.0	848408.53	265958.786834
2	apartment	Pernambuco	Northeast	-8.125550	-34.907601	75.0	299438.28	93867.799373
3	apartment	Pernambuco	Northeast	-8.120249	-34.895920	187.0	848408.53	265958.786834
4	apartment	Pernambuco	Northeast	-8.142666	-34.906906	80.0	464129.36	145495.097179

In [43]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.8", df2)
```

Excellent! Keep going.
Score: 1

Task 1.5.9: Drop the `"price_brl"` column from `df2`, as well as any rows that have `NaN` values.

In [57]:
```python
#Drop the 'price_brl' column as well as null values
df2.drop(columns=['price_brl'],inplace=True)
df2.dropna(inplace=True)
#Inspect dataframe to view new dataset
df2.head()
```

Out[57]:

	property_type	state	region	lat	lon	area_m2	price_usd
0	apartment	Pernambuco	Northeast	-8.134204	-34.906326	72.0	129850.463950
1	apartment	Pernambuco	Northeast	-8.126664	-34.903924	136.0	265958.786834
2	apartment	Pernambuco	Northeast	-8.125550	-34.907601	75.0	93867.799373
3	apartment	Pernambuco	Northeast	-8.120249	-34.895920	187.0	265958.786834
4	apartment	Pernambuco	Northeast	-8.142666	-34.906906	80.0	145495.097179

In [58]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.9", df2)
```

Very impressive.
Score: 1.0

Task 1.5.10: Concatenate `df1` and `df2` to create a new DataFrame named `df`.

In [60]:
```python
#Concatenate df1 and df2 to create df
df = pd.concat([df1,df2])
#Print new dataframe
df.shape
```

Out[60]:
```
(22844, 7)
```

In [61]:
```python
wqet_grader.grade(
    "Project 1 Assessment", "Task 15.10", df.sort_values("price_usd").head()
)
```

Yes! Your hard work is paying off.
Score: 1.0

Keep it up! Your data is clean. Take another break. You've earned it. 🤓

Explore

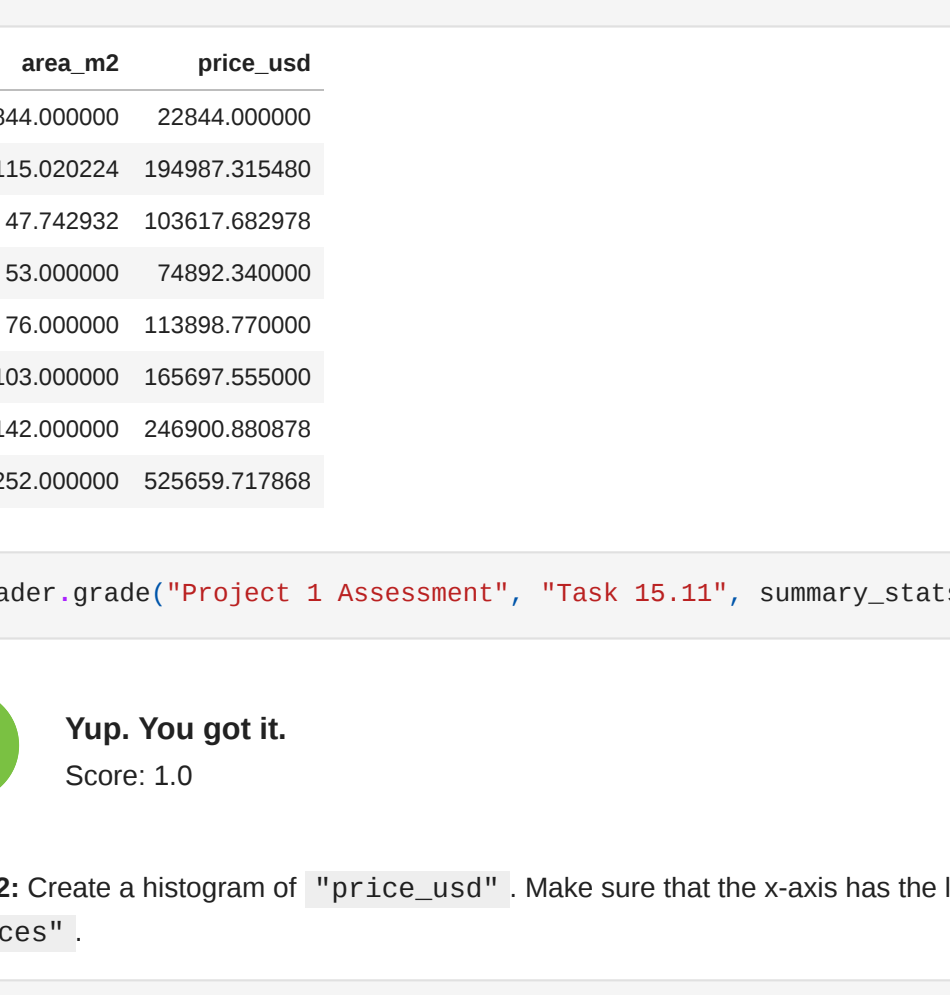
It's time to start exploring your data. In this section, you'll use your new data visualization skills to learn more about the regional differences in the Brazilian real estate market.

Complete the code below to create a `scatter_mapbox` showing the location of the properties in `df`.

In [62]:
```python
fig = px.scatter_mapbox(
    df,
    lat="lat",
    lon="lon",
    center={"lat": -14.2, "lon": -51.9},  # Map will be centered on Brazil
    width=600,
    height=600,
    hover_data=["price_usd"],  # Display price when hovering mouse over house
)

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

fig.show()
```



Task 1.5.11: Use the `describe` method to create a DataFrame `summary_stats` with the summary statistics for the `"area_m2"` and `"price_usd"` columns.

In [63]:
```python
#Create and print summary statistics for the 'area_m2' and 'price_usd' columns
summary_stats = df[['area_m2' , 'price_usd']].describe()
summary_stats
```

Out[63]:

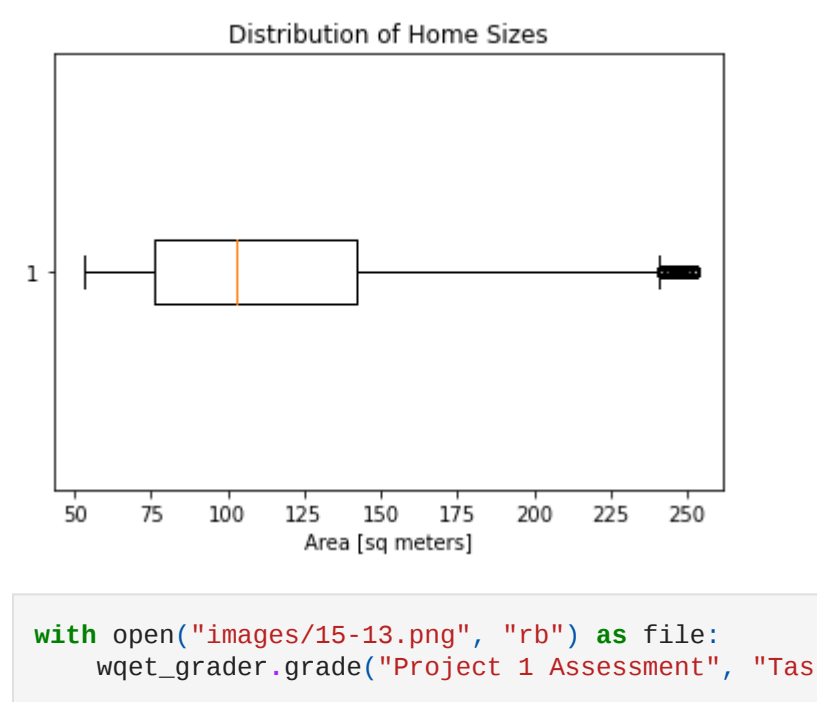
	area_m2	price_usd
count	22844.000000	22844.000000
mean	115.020224	194987.315480
std	47.742932	103617.682978
min	53.000000	74892.340000
25%	76.000000	113898.770000
50%	103.000000	165697.555000
75%	142.000000	246900.880878
max	252.000000	525659.717868

In [64]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.11", summary_stats)
```

Yup. You got it.
Score: 1.0

Task 1.5.12: Create a histogram of `"price_usd"`. Make sure that the x-axis has the label `"Price [USD]"`, the y-axis has the label `"Frequency"`, and the plot has the title `"Distribution of Home Prices"`.

In [74]:
```python
#Create a histogram for the 'price_usd' series
plt.hist(df['price_usd'])
plt.xlabel('Price [USD]')
plt.ylabel('Frequency')
plt.title('Distribution of Home Prices')
# Don't change the code below
plt.savefig("images/15-12.png", dpi=150)
```

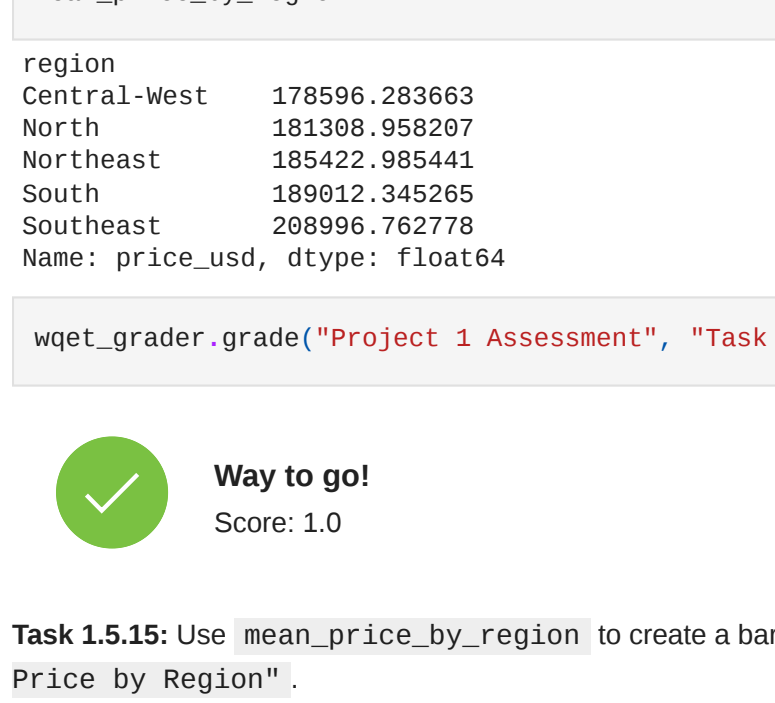


In [75]:
```python
with open("images/15-12.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 15.12", file)
```

Wow, you're making great progress.
Score: 1.0

Task 1.5.13: Create a horizontal boxplot of `"area_m2"`. Make sure that the x-axis has the label `"Area [sq meters]"` and the plot has the title `"Distribution of Home Sizes"`.

In [76]:
```python
#Create a horizontal boxplot of 'area_m2'
plt.boxplot(df['area_m2'] , vert=False)
plt.xlabel('Area [sq meters]')
plt.title('Distribution of Home Sizes')
# Don't change the code below
plt.savefig("images/15-13.png", dpi=150)
```



In [77]:
```python
with open("images/15-13.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 15.13", file)
```

You = coding 🥷
Score: 1.0

Task 1.5.14: Use the `groupby` method to create a Series named `mean_price_by_region` that shows the mean home price in each region in Brazil, sorted from smallest to largest.

In [82]:
```python
#Create a Series with GroupBy method
mean_price_by_region = (
                        df
                        .groupby('region')['price_usd'].mean()
                        .sort_values()
)

mean_price_by_region
```

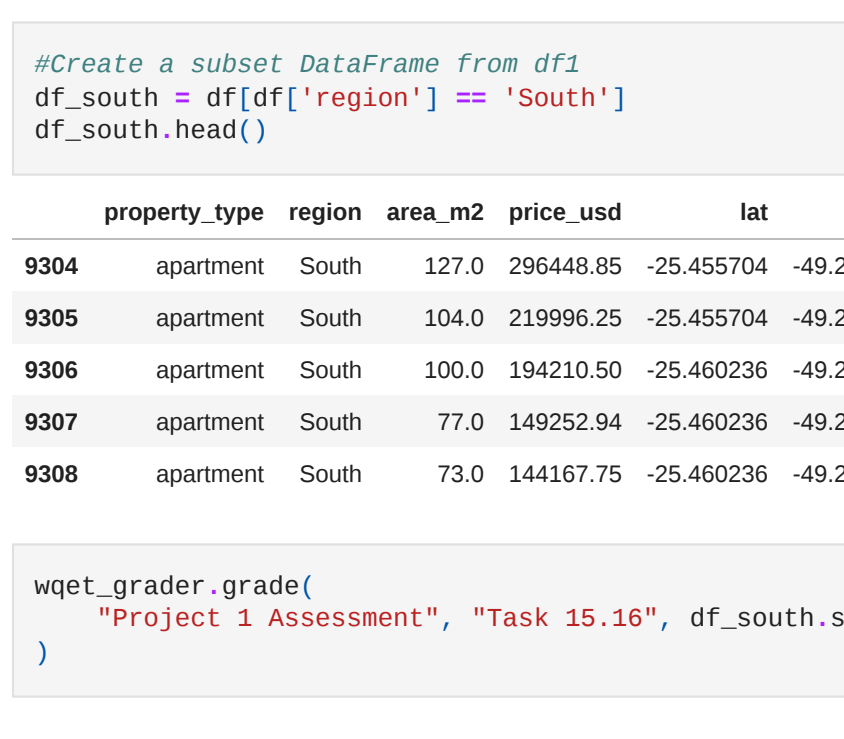
Out[82]:
```
region
Central-West    178596.283663
North           181308.958207
Northeast       185422.985441
South           189012.345265
Southeast       208996.762778
Name: price_usd, dtype: float64
```

In [83]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.14", mean_price_by_region)
```

Way to go!
Score: 1.0

Task 1.5.15: Use `mean_price_by_region` to create a bar chart. Make sure you label the x-axis as `"Region"` and the y-axis as `"Mean Price [USD]"`, and give the chart the title `"Mean Home Price by Region"`.

In [85]:
```python
#Create a bar chart of 'mean_price_by_region'
mean_price_by_region.plot(
    kind='bar',
    xlabel='Region',
    ylabel='Mean Price [USD]',
    title='Mean Home Price by Region'
);
# Don't change the code below
plt.savefig("images/15-15.png", dpi=150)
```



In [86]:
```python
with open("images/15-15.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 15.15", file)
```

Python master 😁
Score: 1.0

Keep it up! You're halfway through your data exploration. Take one last break and get ready for the final push. 🚀

You're now going to shift your focus to the southern region of Brazil, and look at the relationship between home size and price.

Task 1.5.16: Create a DataFrame `df_south` that contains all the homes from `df` that are in the `"South"` region.

In [89]:
```python
#Create a subset DataFrame from df1
df_south = df[df['region'] == 'South']
df_south.head()
```

Out[89]:

	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

In [90]:
```python
wqet_grader.grade(
    "Project 1 Assessment", "Task 15.16", df_south.sort_values("area_m2").head()
)
```

Good work!
Score: 1.0

Task 1.5.17: Use the `value_counts` method to create a Series `homes_by_state` that contains the number of properties in each state in `df_south`.

In [97]:
```python
#Create a Series that contains the number of properties in each state
homes_by_state = df_south['state'].value_counts()
homes_by_state
```

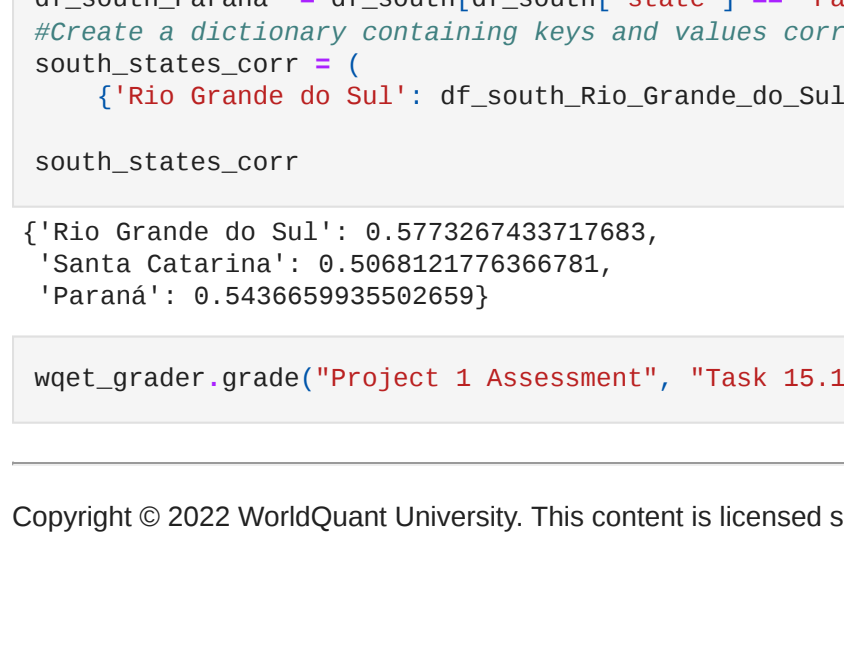
Out[97]:
```
Rio Grande do Sul    2643
Santa Catarina       2634
Paraná               2544
Name: state, dtype: int64
```

In [98]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.17", homes_by_state)
```

Correct.
Score: 1.0

Task 1.5.18: Create a scatter plot showing price vs. area for the state in `df_south` that has the largest number of properties. Be sure to label the x-axis `"Area [sq meters]"` and the y-axis `"Price [USD]"`; and use the title `"<name of state>: Price vs. Area"`.

In [119…
```python
#Subset `df` to include only observations from `Rio Grande do Sul`
df_Rio_Grande_do_Sul = df_south[df_south['state'] == 'Rio Grande do Sul']
#Create a scatter plot showing price vs. area for Rio Grande do Sul
plt.scatter(df_Rio_Grande_do_Sul['area_m2'] , df_Rio_Grande_do_Sul['price_usd'])
plt.xlabel('Area [sq meters]')
plt.ylabel('Price [USD]')
plt.title('Rio Grande do Sul: Price vs. Area')
# Don't change the code below
plt.savefig("images/15-18.png", dpi=150)
```



In [2]:
```python
with open("images/15-18.png", "rb") as file:
        wqet_grader.grade("Project 1 Assessment", "Task 15.18", file)
```

Task 1.5.19: Create a dictionary `south_states_corr`, where the keys are the names of the three states in the `"South"` region of Brazil, and their associated values are the correlation coefficient between `"area_m2"` and `"price_usd"` in that state.

As an example, here's a dictionary with the states and correlation coefficients for the Southeast region. Since you're looking at a different region, the states and coefficients will be different, but the structure of the dictionary will be the same.

```python
{'Espírito Santo': 0.6311332554173303,
 'Minas Gerais': 0.5830029036378931,
 'Rio de Janeiro': 0.4554077103515366,
 'São Paulo': 0.45882050624839366}
```

In [123…
```python
#Subset `df` to include only observations from `Rio Grande do Sul` , 'Santa Catarina' and 'Paraná'
df_south_Rio_Grande_do_Sul = df_south[df_south['state'] == 'Rio Grande do Sul']
df_south_Santa_Catarina  = df_south[df_south['state'] == 'Santa Catarina']
df_south_Paraná  = df_south[df_south['state'] == 'Paraná']
#Create a dictionary containing keys and values corresponding the correlation coefficient between 'area_m2' and 'price_usd' of states in df_south
south_states_corr = (
    {'Rio Grande do Sul': df_south_Rio_Grande_do_Sul['area_m2'].corr(df_south_Rio_Grande_do_Sul['price_usd']) , 'Santa Catarina': df_south_Santa_Catarina['area_m2

south_states_corr
```

Out[123…
```
{'Rio Grande do Sul': 0.5773267433717683,
 'Santa Catarina': 0.5068121776366781,
 'Paraná': 0.5436659935502659}
```

In [ ]:
```python
wqet_grader.grade("Project 1 Assessment", "Task 15.19", south_states_corr)
```