Usage Guidelines

This lesson is part of the **DS Lab core curriculum**. For that reason, this notebook can only be used on your WQU virtual machine.

This means:

- ⊘ No downloading this notebook.
- ⊘ No re-sharing of this notebook with friends or colleagues.
- ⊘ No downloading the embedded videos in this notebook.
- ⊘ No re-sharing embedded videos with friends or colleagues.
- ⊘ No adding this notebook to public or private repositories.
- ⊘ No uploading this notebook (or screenshots of it) to other websites, including websites for study resources.

1.5. Housing in Brazil BR

[1]:
```
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.

Note: There are are 19 graded tasks in this assignment, but you only need to complete 18. Once you've successfully completed 18 tasks, you'll be automatically enrolled in the next project, and this assignment will be closed. This means that you might not be allowed to complete the last task. So if you get an error saying that you've already complete the course, that's good news! Move to project 2.

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.

[3]:
```
# Import Matplotlib, pandas, and plotly
import matplotlib.pyplot as plt
import pandas as pd
```

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.

```
# 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, one-by-one.

Import

First, you are going to import and clean the data in `data/brasil-real-estate-1.csv`.

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

```
df1 = pd.read_csv("data/brasil-real-estate-1.csv")
df1.head()
```

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.1", df1)
```

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?

```
df1.info()
df1.head(10)
```

```
<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
```

	property_type	place_with_parent_names	region	lat-lon	area_m2	price_usd
0	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6443051,-35.7088142	110.0	$187,230.85
1	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.6430934,-35.70484	65.0	$81,133.37
2	house	|Brasil|Alagoas|Maceió|	Northeast	-9.6227033,-35.7297953	211.0	$154,465.45
3	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.622837,-35.719556	99.0	$146,013.20
4	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.654955,-35.700227	55.0	$101,416.71
5	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.614414,-35.735621	56.0	$75,727.07
6	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.584755,-35.662909	68.0	$110,916.18
7	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.658285,-35.703827	187.0	$249,641.14
8	apartment	|Brasil|Alagoas|Maceió|	Northeast	NaN	65.0	$100,792.61
9	apartment	|Brasil|Alagoas|Maceió|	Northeast	-9.66082,-35.702976	90.0	$115,459.02

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

```
df1.dropna(inplace=True)
df1.info()
df1.head()
```

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

```
df1.dropna(inplace=True)
df1.info()
df1.head()
```

```
<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
```

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.2", df1)
```

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`.

1]:
```
df1[["lat" , "lon"]] = df1["lat-lon"].str.split("," , expand=True).astype(float)
df1.head()
```

1]:

	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

2]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.3", df1)
```

Yes! Great problem solving.
Score: 1

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.)

4]:
```
df1["state"] = df1["place_with_parent_names"].str.split("|", expand=True)[2]
df1.head()
```

4]:

	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

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.4", df1)
```

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

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

```
df1["price_usd"] = df1["price_usd"].str.replace("$" , "").str.replace("," , "").astype(float)
df1.head()
```

```
/tmp/ipykernel_1098/1858691637.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addit
ion, single character regular expressions will *not* be treated as literal strings when regex=True.
  df1["price_usd"] = df1["price_usd"].str.replace("$" , "").str.replace("," , "").astype(float)
```

	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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.5", df1)
```

Score: 1

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

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.6", df1)
```

Awesome work.
Score: 1

Now that you have cleaned `data/brasil-real-estate-1.csv` and created `df1`, you are going to import and clean the data from the second file, `brasil-real-estate-2.csv`.

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

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.7", df2)
```

Excellent work.
Score: 1

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`?

```
df2.info()
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
```

```
df2.info()
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
```

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

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.)

```
df2["price_usd"] = df2["price_brl"] / 3.19
```

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.8", df2)
```

Very impressive.

Score: 1

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

```
df2.dropna(inplace=True)
df2.drop(columns=["price_brl"], inplace=True)
df2.head()
```

	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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.9", df2)
```

Excellent! Keep going.
Score: 1

OK! Now that you've cleaned the data from both CSV files and created `df1` and `df2`, it's time to combine them into a single DataFrame.

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

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

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

```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.10", df)
```

Yes! Great problem solving.
Score: 1

Explore

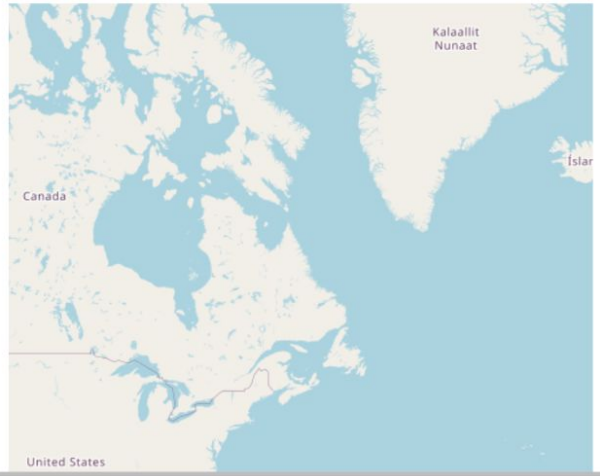
It's time to start exploring your data. In this section, you'll use your new data visualization skills to learn more about the regional differences in the Brazilian real estate market.

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

]:
```
fig = px.scatter_mapbox(
    df,
    lat = df["lat"],
    lon = df["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.

]:
```
summary_stats = df[["area_m2" , "price_usd"]].describe()
summary_stats
```

]:

	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

]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.11", summary_stats)
```

You're making this look easy. 😉
Score: 1

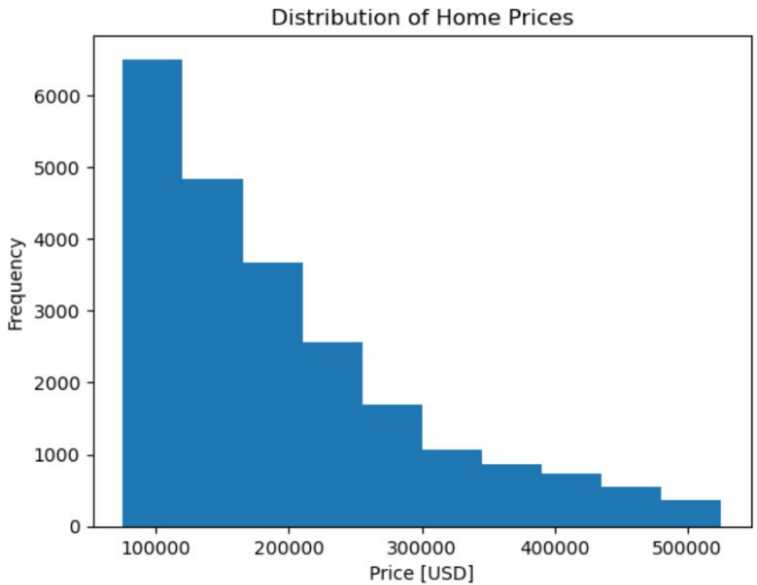
Task 1.5.12: Create a histogram of `"price_usd"`. Make sure that the x-axis has the label `"Price [USD]"`, the y-axis has the label `"Frequency"`, and the plot has the title `"Distribution of Home Prices"`. Use Matplotlib (`plt`).

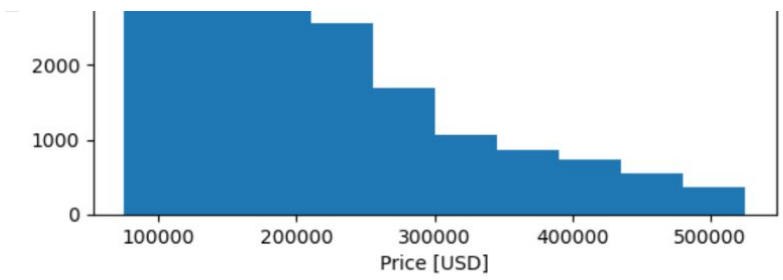
8]:
```
# Build histogram
plt.hist(df["price_usd"])


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

# Add title
plt.title("Distribution of Home Prices")

# Don't change the code below 👇
plt.savefig("images/1-5-12.png", dpi=150)
```






```
]: with open("images/1-5-12.png", "rb") as file:
       wqet_grader.grade("Project 1 Assessment", "Task 1.5.12", file)
```

Excellent work.

Score: 1

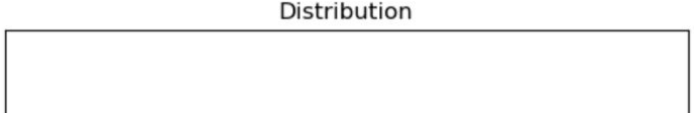
Task 1.5.13: Create a horizontal boxplot of `"area_m2"`. Make sure that the x-axis has the label `"Area [sq meters]"` and the plot has the title `"Distribution of Home Sizes"`. Use Matplotlib (`plt`).

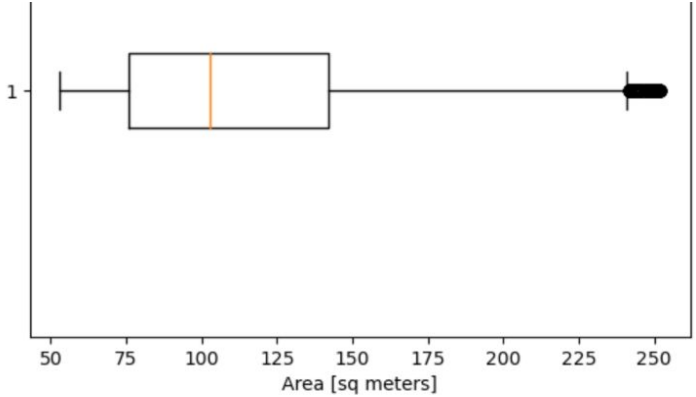
```
]: # Build box plot
   plt.boxplot(df["area_m2"], vert=False)


   # Label x-axis
   plt.xlabel("Area [sq meters]")

   # Add title
   plt.title("Distribution")

   # Don't change the code below 👇
   plt.savefig("images/1-5-13.png", dpi=150)
```






```
with open("images/1-5-13.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.13", file)
```

You = coding 🤦‍♀️
Score: 1

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.

```
mean_price_by_region = df.groupby("region")["price_usd"].mean().sort_values(ascending=True)

mean_price_by_region.shape
```

```
(5,)
```

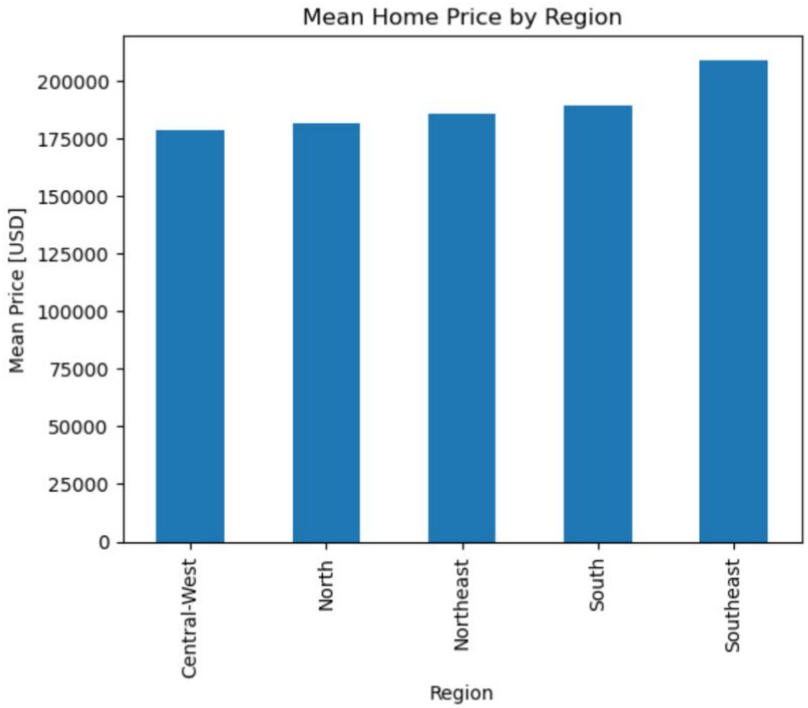
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.14", mean_price_by_region)
```

Wow, you're making great progress.
Score: 1

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"`. Use pandas.

]:
```
# Build bar chart, label axes, add title
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/1-5-15.png", dpi=150)
```



]:
```
with open("images/1-5-15.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.15", file)
```

```
with open("images/1-5-15.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.15", file)
```

Awesome work.
Score: 1

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.

```
df_south = df[df["region"] == "South"]
df_south
```

	property_type	region	area_m2	price_usd	lat	lon	state
9304	apartment	South	127.0	296448.850000	-25.455704	-49.292918	Paraná
9305	apartment	South	104.0	219996.250000	-25.455704	-49.292918	Paraná
9306	apartment	South	100.0	194210.500000	-25.460236	-49.293812	Paraná
9307	apartment	South	77.0	149252.940000	-25.460236	-49.293812	Paraná
9308	apartment	South	73.0	144167.750000	-25.460236	-49.293812	Paraná
...	...	...	...	...	...	...	...
9741	apartment	South	117.0	309763.761755	-26.966631	-48.636383	Santa Catarina
9742	house	South	110.0	88616.510972	-26.754795	-48.729183	Santa Catarina
9744	house	South	165.0	110770.645768	-27.454047	-48.411582	Santa Catarina
9745	apartment	South	65.0	86045.485893	-26.997210	-48.633877	Santa Catarina
9747	apartment	South	79.0	238122.915361	-27.594744	-48.541233	Santa Catarina

7821 rows × 7 columns

7821 rows × 7 columns

```
5]:
wqet_grader.grade("Project 1 Assessment", "Task 1.5.16", df_south)
```

Yes! Great problem solving.
Score: 1

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`.

```
7]:
homes_by_state = df_south["state"].value_counts()
homes_by_state
```

```
7]:
Rio Grande do Sul    2643
Santa Catarina       2634
Paraná               2544
Name: state, dtype: int64
```

```
3]:
wqet_grader.grade("Project 1 Assessment", "Task 1.5.17", homes_by_state)
```

Way to go!
Score: 1

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"`. Use Matplotlib (`plt`).

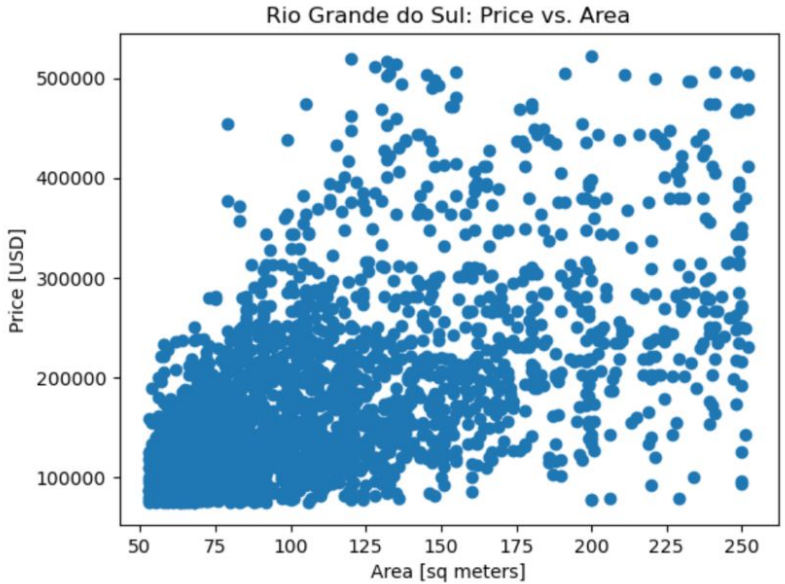
Tip: You should replace `<name of state>` with the name of the state that has the largest number of properties.

```
9]:
# Subset data
df_south_rgs = df_south[df_south['state'] == homes_by_state.idxmax()]


# Build scatter plot
plt.scatter(x=df_south_rgs["area_m2"] , y=df_south_rgs["price_usd"])
```

```
# Build scatter plot
plt.scatter(x=df_south_rgs["area_m2"] , y=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")

# Don't change the code below 👇
plt.savefig("images/1-5-18.png", dpi=150)
```



0]:
```
with open("images/1-5-18.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.18", file)
```

Score: 1