015-assignment

May 23, 2022

Assignment: Housing in Brazil

[1]:
```
import wqet_grader

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

```
<IPython.core.display.HTML object>
```

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.

```
<b>Note:</b> There are are 19 graded tasks in this assignment, but you only need to complete 18
```

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.

[2]:
```
# Import Matplotlib, pandas, and plotly
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
```

1 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.

1.1 Import

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

[3]:
```
df1 = pd.read_csv("data/brasil-real-estate-1.csv")
df1.head()
```

[3]:

	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
```

[4]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.1", df1)
```

```
<IPython.core.display.HTML object>
```

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?

[5]:
```
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
```

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

[6]:
```
df1.dropna(inplace=True)
df1.head()
```

[6]:
```
  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
```

[7]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.2", df1)
```

```
<IPython.core.display.HTML object>
```

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`.

[8]:
```
df1["lat-lon"].head()
```

[8]:
```
0    -9.6443051,-35.7088142
1      -9.6430934,-35.70484
2    -9.6227033,-35.7297953
3      -9.622837,-35.719556
4      -9.654955,-35.700227
Name: lat-lon, dtype: object
```

[9]:
```
df1["lat-lon"].str.split(",")
```

[9]:
```
0        [-9.6443051, -35.7088142]
1          [-9.6430934, -35.70484]
2        [-9.6227033, -35.7297953]
3          [-9.622837, -35.719556]
4          [-9.654955, -35.700227]
                   …
12828      [-8.044497, -34.909519]
12829      [-8.056418, -34.909309]
12830     [-8.1373477, -34.909181]
12831     [-8.1136717, -34.896252]
12833     [-8.0578381, -34.882897]
Name: lat-lon, Length: 11551, dtype: object
```

[10]:
```
df1["lat-lon"].str.split(",", expand=True)
```

[10]:
```
                0           1
0      -9.6443051  -35.7088142
1      -9.6430934    -35.70484
2      -9.6227033  -35.7297953
3       -9.622837   -35.719556
4       -9.654955   -35.700227
…            …           …
12828   -8.044497   -34.909519
12829   -8.056418   -34.909309
12830  -8.1373477   -34.909181
12831  -8.1136717   -34.896252
12833  -8.0578381   -34.882897
```

```
[11551 rows x 2 columns]
```

[11]:
```
df1[["lat","lon"]] =df1["lat-lon"].str.split(",", expand=True).astype(float)
df1.head()
```

[11]:
```
  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       lat        lon
0    110.0  $187,230.85 -9.644305 -35.708814
1     65.0   $81,133.37 -9.643093 -35.704840
2    211.0  $154,465.45 -9.622703 -35.729795
3     99.0  $146,013.20 -9.622837 -35.719556
4     55.0  $101,416.71 -9.654955 -35.700227
```

[12]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.3", df1)
```

```
<IPython.core.display.HTML object>
```

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.)

[13]:
```
df1["place_with_parent_names"].head()
```

[13]:
```
0    |Brasil|Alagoas|Maceió|
1    |Brasil|Alagoas|Maceió|
2    |Brasil|Alagoas|Maceió|
3    |Brasil|Alagoas|Maceió|
4    |Brasil|Alagoas|Maceió|
Name: place_with_parent_names, dtype: object
```

[14]:
```
df1["place_with_parent_names"].str.split("|", expand=True).head()
```

[14]:
```
   0       1        2       3 4     5
0    Brasil  Alagoas  Maceió    None
1    Brasil  Alagoas  Maceió    None
2    Brasil  Alagoas  Maceió    None
3    Brasil  Alagoas  Maceió    None
4    Brasil  Alagoas  Maceió    None
```

[15]:
```
df1["place_with_parent_names"].str.split("|", expand=True)[2].head()
```

[15]:
```
0    Alagoas
1    Alagoas
2    Alagoas
3    Alagoas
4    Alagoas
Name: 2, dtype: object
```

[16]:
```
df1["state"]= df1["place_with_parent_names"].str.split("|", expand=True)[2]
```

[17]:
```
df1.head()
```

[17]:

	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

[18]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.4", df1)
```

```
<IPython.core.display.HTML object>
```

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

[21]:
```
#df1["price_usd"].str.replace("$","",regex=False).head()
```

[21]:
```
0    187,230.85
1     81,133.37
2    154,465.45
3    146,013.20
4    101,416.71
Name: price_usd, dtype: object
```

[22]:
```
#df1["price_usd"].str.replace("$","",regex=False).str.replace(",","").head()
```

[22]:
```
0    187230.85
1     81133.37
2    154465.45
3    146013.20
4    101416.71
```

```
Name: price_usd, dtype: object
```

[19]:
```
df1["price_usd"].str.replace("$","",regex=False)
```

[19]:
```
0        187,230.85
1         81,133.37
2        154,465.45
3        146,013.20
4        101,416.71
            …
12828    134,182.11
12829    174,748.79
12830    115,459.02
12831    137,302.62
12833    168,507.77
Name: price_usd, Length: 11551, dtype: object
```

[20]:
```
df1["price_usd"]=df1["price_usd"].str.replace(",","",regex=False)
```

[21]:
```
df1["price_usd"]
```

[21]:
```
0        $187230.85
1         $81133.37
2        $154465.45
3        $146013.20
4        $101416.71
            …
12828    $134182.11
12829    $174748.79
12830    $115459.02
12831    $137302.62
12833    $168507.77
Name: price_usd, Length: 11551, dtype: object
```

[22]:
```
df1["price_usd"]=df1["price_usd"].str.replace("$","",regex=False)
```

[23]:
```
df1["price_usd"]
```

[23]:
```
0        187230.85
1         81133.37
2        154465.45
3        146013.20
4        101416.71
            …
12828    134182.11
12829    174748.79
12830    115459.02
```

```
12831    137302.62
12833    168507.77
Name: price_usd, Length: 11551, dtype: object
```

[24]:
```
df1["price_usd"]=df1["price_usd"].astype(float)
```

[25]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11551 entries, 0 to 12833
Data columns (total 9 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  float64
 6   lat                     11551 non-null  float64
 7   lon                     11551 non-null  float64
 8   state                   11551 non-null  object
dtypes: float64(4), object(5)
memory usage: 902.4+ KB
```

[22]:
```
#df1["price_usd"].str.replace("$","",regex=False).str.replace(",","").
↪astype(float).head()
```

[26]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.5", df1)
```

```
<IPython.core.display.HTML object>
```

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

[27]:
```
df1.drop(columns=["place_with_parent_names", "lat-lon",],inplace=True)
df1.head()
```

[27]:

	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

[28]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.6", df1)
```

```
<IPython.core.display.HTML object>
```

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

[29]:
```
df2 = pd.read_csv("data/brasil-real-estate-2.csv")
df2.head()
```

[29]:

	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

[30]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.7", df2)
```

```
<IPython.core.display.HTML object>
```

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`?

[31]:
```
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
```

[32]:
```
df2.head()
```

[32]:

	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
```

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.)

[40]:
```
#(df2["price_brl"] / 3.19).head()
```

[40]:
```
0    129850.463950
1    265958.786834
2     93867.799373
3    265958.786834
4    145495.097179
Name: price_brl, dtype: float64
```

[26]:
```
#(df2["price_brl"] / 3.19).round(2).head()
```

[26]:
```
0    129850.46
1    265958.79
2     93867.80
3    265958.79
4    145495.10
Name: price_brl, dtype: float64
```

[33]:
```
df2["price_usd"] = df2["price_brl"] / 3.19

df2.head()
```

[33]:
```
  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      price_usd
0  414222.98  129850.463950
1  848408.53  265958.786834
2  299438.28   93867.799373
3  848408.53  265958.786834
4  464129.36  145495.097179
```

[46]:
```
#df2.drop(columns=["price_brl"], inplace=True)

#df2.head()
```

[46]:
```
  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_usd
0  129850.463950
1  265958.786834
2   93867.799373
3  265958.786834
4  145495.097179
```

[47]:
```
#df2.dropna(inplace=True,axis=0)
#df2.head()
```

[47]:
```
  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_usd
0  129850.463950
1  265958.786834
2   93867.799373
3  265958.786834
4  145495.097179
```

[34]:
```
df2.shape
```

[34]:
```
(12833, 8)
```

[35]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.8", df2)
```

```
<IPython.core.display.HTML object>
```

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

[36]:
```
df2.dropna(inplace=True,axis=0)
df2.head()
```

[36]:
```
  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      price_usd
0  414222.98  129850.463950
1  848408.53  265958.786834
2  299438.28   93867.799373
3  848408.53  265958.786834
4  464129.36  145495.097179
```

[37]:
```
df2.drop(columns=["price_brl"],inplace=True)
```

[38]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.9", df2)
```

```
<IPython.core.display.HTML object>
```

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

[39]:
```
df = pd.concat([df1,df2,])
print("df shape:", df.shape)
```

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

[40]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.10", df)
```

```
<IPython.core.display.HTML object>
```

```
<p><b>Frequent Question:</b> I can't pass this question, and I don't know what I've done wrong
<p><b>Tip:</b> In this assignment, you're working with data that's similar -
but not identical - the data used in the lessons. That means that you might need to make adjust
```

1.2 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`.

[41]:
```
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.

[42]:
```
summary_stats = df[["area_m2","price_usd"]]
summary_stats= summary_stats.describe()
```

[43]:
```
summary_stats.shape
```

[43]:
```
(8, 2)
```

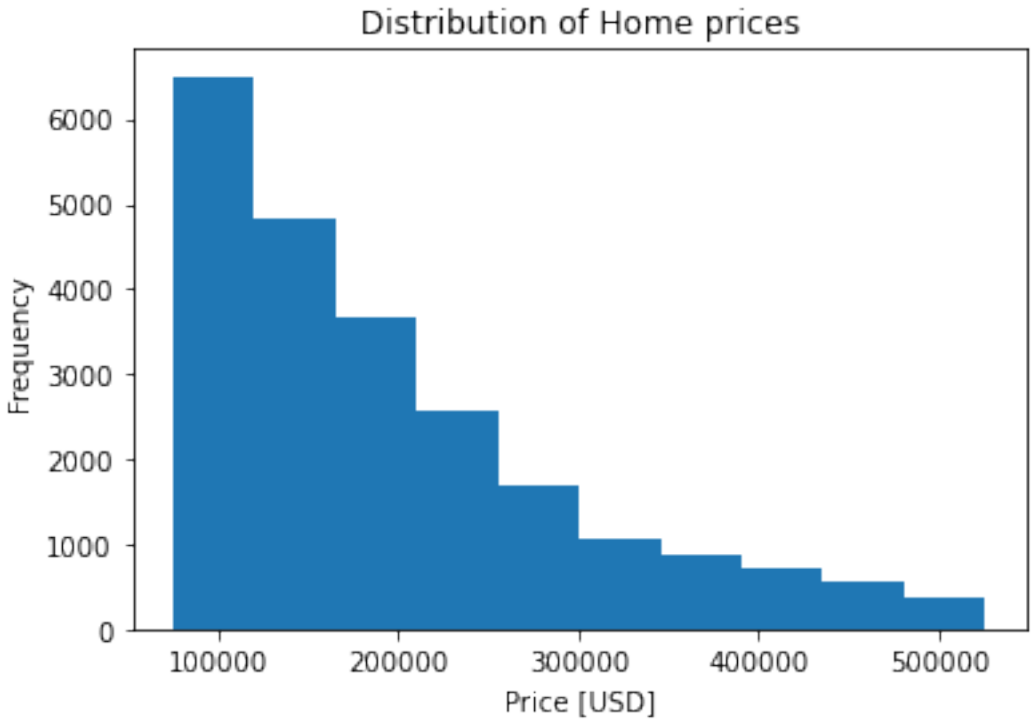
[44]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.11", summary_stats)
```

```
<IPython.core.display.HTML object>
```

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"`.

[45]:
```
plt.hist(df["price_usd"])
plt.xlabel("Price [USD]") # x axis label
plt.ylabel("Frequency") #vy axis label
plt.title("Distribution of Home prices");
# Don't change the code below
```

```
plt.savefig("images/1-5-12.png", dpi=150)
```

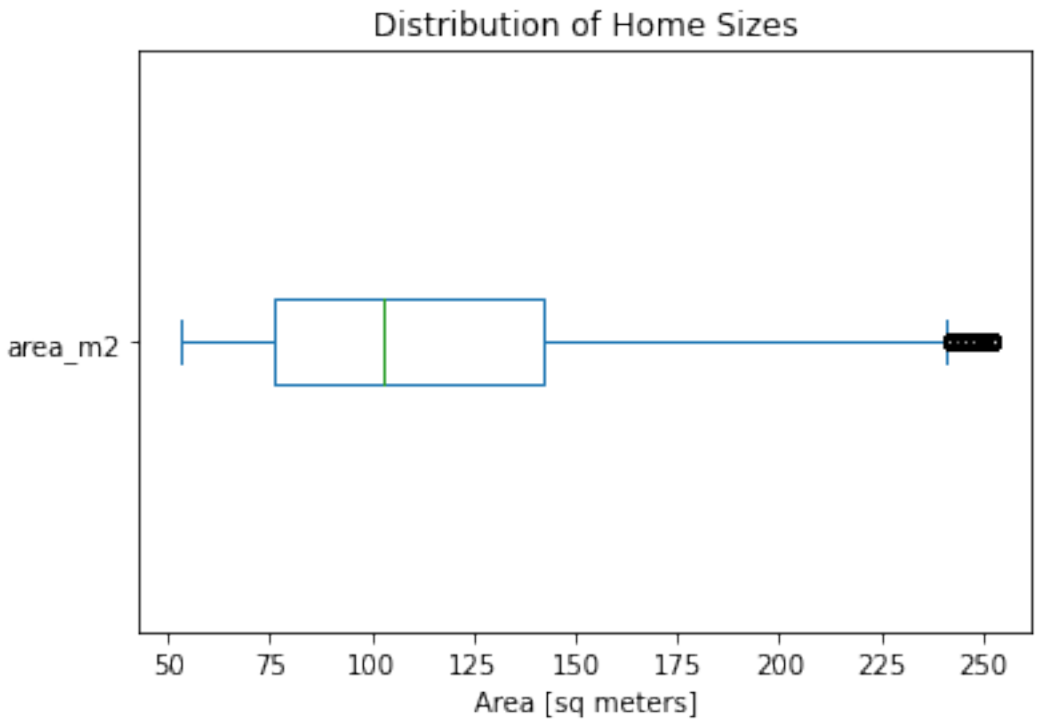


[46]:
```
with open("images/1-5-12.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.12", file)
```

```
<IPython.core.display.HTML object>
```

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"`.

[47]:
```
df["area_m2"].plot(kind="box",vert=False)
plt.xlabel("Area [sq meters]") # x axis label
plt.title("Distribution of Home Sizes");
# Don't change the code below
plt.savefig("images/1-5-13.png", dpi=150)
```



[48]:
```
with open("images/1-5-13.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.13", file)
```

```
<IPython.core.display.HTML object>
```

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.

[49]:
```
df
```

[49]:

	property_type	region	area_m2	price_usd	lat	lon	\
0	apartment	Northeast	110.0	187230.850000	-9.644305	-35.708814	
1	apartment	Northeast	65.0	81133.370000	-9.643093	-35.704840	
2	house	Northeast	211.0	154465.450000	-9.622703	-35.729795	
3	apartment	Northeast	99.0	146013.200000	-9.622837	-35.719556	
4	apartment	Northeast	55.0	101416.710000	-9.654955	-35.700227	
…	…	…	…	…	…	…	
12827	house	Southeast	180.0	131414.921630	-23.595098	-46.796448	
12828	house	Southeast	250.0	134543.852665	-23.587495	-46.559401	
12829	apartment	Southeast	55.0	79121.880878	-23.522029	-46.189290	
12830	apartment	Southeast	57.0	100125.655172	-23.526443	-46.529182	
12832	apartment	North	70.0	90738.874608	-10.249091	-48.324286	

	state

```
0          Alagoas
1          Alagoas
2          Alagoas
3          Alagoas
4          Alagoas
…            …
12827    São Paulo
12828    São Paulo
12829    São Paulo
12830    São Paulo
12832    Tocantins

[22844 rows x 7 columns]
```

[50]:
```
mean_price_by_region = df.groupby("region")["price_usd"].mean()
mean_price_by_region
```

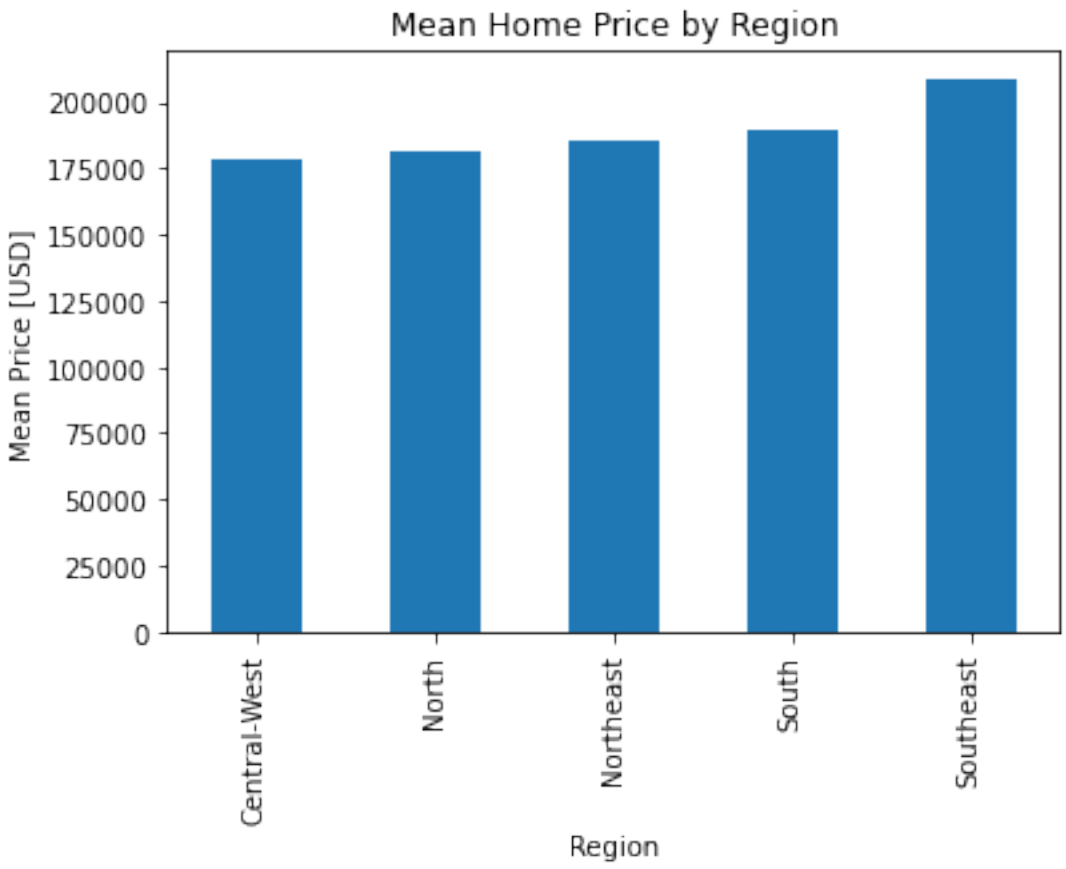
[50]:
```
region
Central-West    178596.283663
North           181308.958207
Northeast       185422.985441
South           189012.345265
Southeast       208996.762778
Name: price_usd, dtype: float64
```

[51]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.14", mean_price_by_region)
```

```
<IPython.core.display.HTML object>
```

"Region"**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"`.

[52]:
```
mean_price_by_region.plot(kind="bar")
plt.xlabel("Region")
plt.ylabel("Mean Price [USD]")
plt.title("Mean Home Price by Region")
# Don't change the code below
plt.savefig("images/1-5-15.png", dpi=150)
```



[53]:
```
with open("images/1-5-15.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.15", file)
```

```
<IPython.core.display.HTML object>
```

```
<b>Keep it up!</b> You're halfway through your data exploration. Take one last break and get re
```

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.

[70]:
```
df_south =df [df["region"]=="South"]
df_south.head()
```

[70]:

	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
```

[71]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.16", df_south)
```

```
<IPython.core.display.HTML object>
```

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`.

[72]:
```
homes_by_state = df_south["state"].value_counts()
homes_by_state
```

[72]:
```
Rio Grande do Sul    2643
Santa Catarina       2634
Paraná               2544
Name: state, dtype: int64
```

[73]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.17", homes_by_state)
```

```
<IPython.core.display.HTML object>
```

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"`.

[ ]:
```

```

```
<p><b>Tip:</b> You should replace <code>&lt;name of state&gt;</code> with the name of the state
```

[74]:
```
df_south
```

[74]:

	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

```
9308          Paraná
…               …
9741  Santa Catarina
9742  Santa Catarina
9744  Santa Catarina
9745  Santa Catarina
9747  Santa Catarina

[7821 rows x 7 columns]
```

[75]:
```
df_south =df_south[df_south["state"]=="Rio Grande do Sul"]
```

[76]:
```
df_south
```

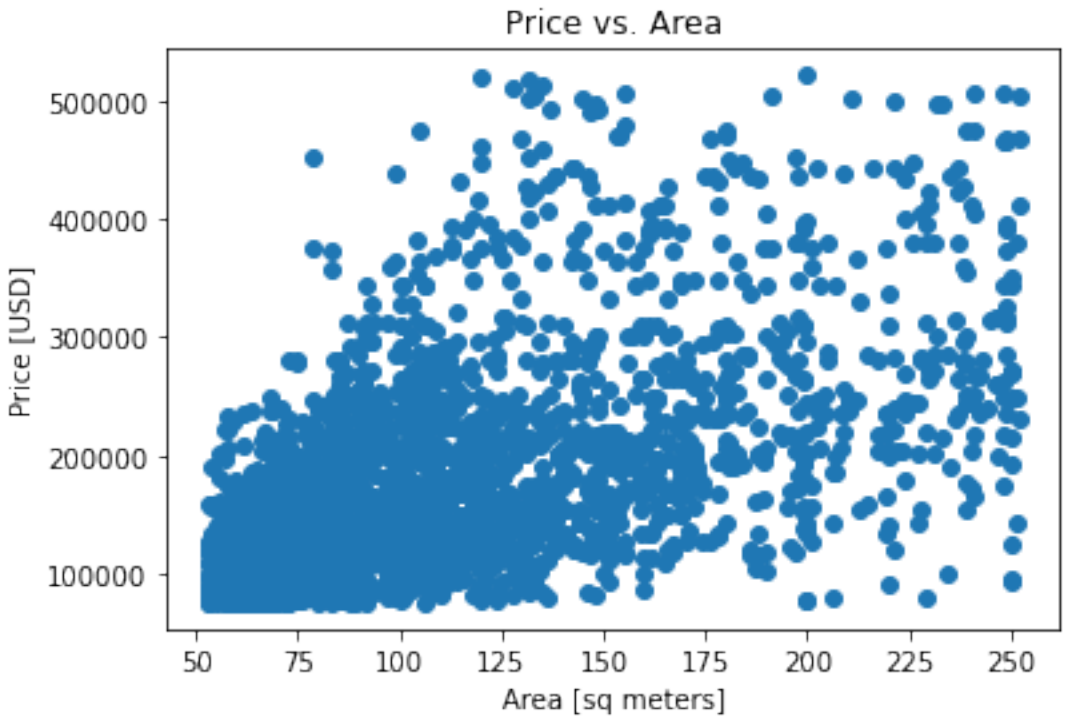
[76]:
```
     property_type region  area_m2      price_usd        lat        lon  \
743          house  South    188.0  115770.288401 -30.027105 -51.130470
745      apartment  South     65.0  123430.141066 -30.039816 -51.223164
746      apartment  South    142.0  185145.222571 -29.696850 -53.858382
748      apartment  South    151.0  256571.996865 -30.033820 -51.198596
750      apartment  South     68.0   75957.012539 -30.034061 -51.135494
…              …    …       …             …         …         …
3738     apartment  South    180.0  142102.918495 -29.692444 -53.813514
3739     apartment  South    172.0  199889.115987 -29.973013 -51.124569
3740     apartment  South    200.0  201483.893417 -29.162448 -51.517110
3741     apartment  South     89.0  136089.648903 -30.019669 -51.200359
3742     apartment  South     90.0  165833.119122 -30.047504 -51.204636

                  state
743   Rio Grande do Sul
745   Rio Grande do Sul
746   Rio Grande do Sul
748   Rio Grande do Sul
750   Rio Grande do Sul
…                   …
3738  Rio Grande do Sul
3739  Rio Grande do Sul
3740  Rio Grande do Sul
3741  Rio Grande do Sul
3742  Rio Grande do Sul

[2643 rows x 7 columns]
```

[77]:
```
plt.scatter(y=df_south["price_usd"],x=df_south["area_m2"])
plt.xlabel("Area [sq meters]")
plt.ylabel("Price [USD]")
plt.title("Price vs. Area")
# Don't change the code below
```

```
plt.savefig("images/1-5-18.png", dpi=150)
```



[78]:
```
with open("images/1-5-18.png", "rb") as file:
    wqet_grader.grade("Project 1 Assessment", "Task 1.5.18", file)
```

```
<IPython.core.display.HTML object>
```

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.

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

[79]:
```
south_states_corr = {'Espírito Santo': 0.6311332554173303,
 'Minas Gerais': 0.5830029036378931,
 'Rio de Janeiro': 0.4554077103515366,
 'São Paulo': 0.45882050624839366}
```

```
south_states_corr
```

[79]:
```
{'Espírito Santo': 0.6311332554173303,
 'Minas Gerais': 0.5830029036378931,
 'Rio de Janeiro': 0.4554077103515366,
 'São Paulo': 0.45882050624839366}
```

[80]:
```
wqet_grader.grade("Project 1 Assessment", "Task 1.5.19", south_states_corr)
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
Input In [80], in <cell line: 1>()
----> 1␣
 ↪wqet_grader.grade("Project 1 Assessment", "Task 1.5.19", south_states_corr)

File /opt/conda/lib/python3.9/site-packages/wqet_grader/__init__.py:180, in␣
 ↪grade(assessment_id, question_id, submission)
    175 def grade(assessment_id, question_id, submission):
    176   submission_object = {
    177     'type': 'simple',
    178     'argument': [submission]
    179   }
--> 180   return␣
 ↪show_score(grade_submission(assessment_id, question_id, submission_object))

File /opt/conda/lib/python3.9/site-packages/wqet_grader/transport.py:145, in␣
 ↪grade_submission(assessment_id, question_id, submission_object)
    143     raise Exception('Grader raised error: {}'.format(error['message']))
    144   else:
--> 145     raise Exception('Could not grade submission: {}'.
 ↪format(error['message']))
    146 result = envelope['data']['result']
    148 # Used only in testing

Exception: Could not grade submission: Could not verify access to this␣
 ↪assessment: Received error from WQET submission API: You have already passed␣
 ↪this course!
```