Assignment 1 Housing in Brazil

July 12, 2022

```
[ ]: #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.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.

     df1 = pd.read_csv('data/brasil-real-estate-1.csv')
     df1.shape
     #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()
```

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

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

```
[ ]: #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.

     df1[["lat","lon"]]=df1['lat-lon'].str.split(',', expand=True).astype(float)

     df1.shape
```

[ ]:

```
#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.)

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

[ ]:

```
#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('$',"",regex=False)
    .str.replace(',','')
    .astype(float)
)
```

[ ]:

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

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

[ ]:

```
#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')
#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()
```

[ ]:

```
#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).round(2)
```

[ ]:

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

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

[ ]:

```
#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)
```

[ ]:

```
#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='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.

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

[ ]:

```
#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).

# 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)

# 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)

```

[ ]:

```
#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 of Home Sizes")

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

[ ]:

```
#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()
mean_price_by_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". Use pandas.

# Build bar chart, label axes, add title
mean_price_by_region.plot( kind="bar",
    xlabel="Region",
    ylabel="Mean Price [USD]",
    title="Mean House Price by Region"
)

# Don't change the code below
plt.savefig("images/1-5-15.png", dpi=150)
```

[ ]:

```
#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.head()
```

[ ]:

```
#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.

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

[ ]:

```
#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.

# Subset data
df_south_rgs = df[df_south['state']=='Rio Grande do Sul']

# 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)
```

[ ]:

```
#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}
south_states_corr = df_south['area_m2'].corr(df_south['price_usd'])

south_states_corr
```