Exploratory Data Analysis

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import plotly.express as px
from IPython.display import VimeoVideo
```

In [2]:
```
VimeoVideo("656355010", h="3cc6a34eba", width=600)
```

Out[2]:

After importing, the next step in many data science projects is exploratory data analysis (EDA), where you get a feel for your data by summarizing its main characteristics using descriptive statistics and data visualization. A good way to plan your EDA is by looking each column and asking yourself questions what it says about your dataset.

Import Data

In [3]:
```
VimeoVideo("656354357", h="8d99bdbfcd", width=600)
```

Out[3]:

Task 1.3.1: Read the CSV file that you created in the last notebook (`"../small-data/mexico-real-estate-clean.csv"`) into a DataFrame named `df` . Be sure to check that all your columns are the correct data type before you go to the next task.

- What's a DataFrame?
- What's a CSV file?
- Read a CSV file into a DataFrame using pandas.

In [4]:
```
df = pd.read_csv("data/Bikas-1st-concatenate-dataset.csv")
```

In [5]:
```
df.shape
```

Out[5]:
```
(1736, 6)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1736 entries, 0 to 1735
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   property_type  1736 non-null   object 
 1   state          1736 non-null   object 
 2   lat            1736 non-null   float64
 3   lon            1736 non-null   float64
 4   area_m2        1736 non-null   float64
 5   price_usd      1736 non-null   float64
dtypes: float64(4), object(2)
memory usage: 81.5+ KB
```

In [7]:
```
df.head()
```

Out[7]:

	property_type	state	lat	lon	area_m2	price_usd

	property_type	state	lat	lon	area_m2	price_usd
0	house	Estado de México	19.560181	-99.233528	150.0	67965.56
1	house	Nuevo León	25.688436	-100.198807	186.0	63223.78
2	apartment	Guerrero	16.767704	-99.764383	82.0	84298.37
3	apartment	Guerrero	16.829782	-99.911012	150.0	94308.80
4	house	Yucatán	21.052583	-89.538639	205.0	105191.37

While there are only two `dtypes` in our DataFrame (`object` and `float64`), there are three categories of data: location, categorical, and numeric. Each of these require a different kind of exploration in our analysis.

Location Data: `"lat"` and `"lon"`

They say that the most important thing in real estate is location, and we can see where where in Mexico our houses are located by using the `"lat"` and `"lon"` columns. Since latitude and longitude are based on a coordinate system, a good way to visualize them is to create a scatter plot on top of a map. A great tool for this is the `scatter_mapbox` from the plotly library.

In [8]:
```
VimeoVideo("656353826", h="236e9c5d43", width=600)
```

Out[8]:

Task 1.3.2: Add `"lat"` and `"lon"` to the code below, and run the code. You'll see a map that's centered on Mexico City, and you can use the "Zoom Out" button in the upper-right corner of the map so that you can see the whole country.

- What's location data?
- What's a scatter plot?

In [62]:
```
# visualization
```

```
fig = px.scatter_mapbox(
    df,  # Our DataFrame
    lat="lat",
    lon="lon",
    #color = ['lat', 'lon', 'price_usd'],
    center={"lat": 19.43, "lon": -99.13},  # Map will be centered on Mexico City
    width=800,  # Width of map
    height=800,  # Height of map
    hover_data=["price_usd"],  # Display price when hovering mouse over house

)

fig.update_layout(mapbox_style="open-street-map") #looks like a google map

fig.show()
```

Looking at this map, are the houses in our dataset distributed evenly throughout the country, or are there states or regions that are more prevalent? Can you guess where Mexico's biggest cities are based on this distribution?

Categorical Data: `"state"`

Even though we can get a good idea of which states are most common in our dataset from looking at a map, we can also get the exact count by using the `"state"` column.

In [25]:
```
VimeoVideo("656353463", h="ee8bffd02b", width=600)
```

Out[25]:

Task 1.3.3: Use the `value_counts` method on the `"state"` column to determine the 10 most prevalent states in our dataset.

- What's categorical data?
- What's a Series?
- Aggregate data in a Series using `value_counts` in pandas.

In [26]:
```
df["state"].nunique() #number of unique state
```

Out[26]:
```
30
```

In [27]:
```
df["state"].unique()
```

Out[27]:
```
array(['Estado de México', 'Nuevo León', 'Guerrero', 'Yucatán',
```

```
       'Querétaro', 'Morelos', 'Chiapas', 'Tabasco', 'Distrito Federal',
       'Nayarit', 'Puebla', 'Veracruz de Ignacio de la Llave', 'Sinaloa',
       'Tamaulipas', 'Jalisco', 'San Luis Potosí', 'Baja California',
       'Hidalgo', 'Quintana Roo', 'Sonora', 'Chihuahua',
       'Baja California Sur', 'Zacatecas', 'Aguascalientes', 'Guanajuato',
       'Durango', 'Tlaxcala', 'Colima', 'Oaxaca', 'Campeche'],
      dtype=object)
```

In [29]:
```
df["state"].value_counts()
```

Out[29]:
```
Distrito Federal                  303
Estado de México                  179
Yucatán                           171
Morelos                           160
Querétaro                         128
Veracruz de Ignacio de la Llave   117
Puebla                             95
Nuevo León                         83
Jalisco                            60
San Luis Potosí                    55
Chiapas                            55
Guerrero                           49
Tamaulipas                         48
Quintana Roo                       38
Baja California                    29
Sinaloa                            26
Chihuahua                          20
Tabasco                            20
Hidalgo                            17
Baja California Sur                15
Sonora                             12
Guanajuato                         12
Aguascalientes                     10
Nayarit                             9
Durango                             7
Tlaxcala                            6
Colima                              5
Campeche                            3
Zacatecas                           2
Oaxaca                              2
Name: state, dtype: int64
```

In [30]:
```
df["state"].value_counts().head(10)
```

Out[30]:
```
Distrito Federal                  303
Estado de México                  179
Yucatán                           171
Morelos                           160
Querétaro                         128
Veracruz de Ignacio de la Llave   117
Puebla                             95
Nuevo León                         83
Jalisco                            60
San Luis Potosí                    55
Name: state, dtype: int64
```

Numerical Data: "area_m2" and

"price_usd"

We have a sense for where the houses in our dataset are located, but how much do they cost? How big are they? The best way to answer those questions is looking at descriptive statistics.

In [31]:
```
VimeoVideo("656353149", h="2d5b273746", width=600)
```

Out[31]:

Task 1.3.4: Use the `describe` method to print the mean, standard deviation, and quartiles for the `"area_m2"` and `"price_usd"` columns.

- What's numerical data?
- What's a mean?
- What's a standard deviation?
- What are quartiles?
- Print the summary statistics for a DataFrame using pandas.

In [32]:
```
df[["area_m2" , "price_usd"]].describe()
```

Out[32]:

	area_m2	price_usd
count	1736.000000	1736.000000
mean	170.261521	115331.980800
std	80.594539	65426.173793
min	60.000000	33157.894737
25%	101.750000	65789.473684
50%	156.000000	99262.132105
75%	220.000000	150846.665000
max	385.000000	326733.660000

Let's start by looking at `"area_m2"`. It's interesting that the mean is larger than the median (another name for the 50% quartile). Both of these statistics are supposed to give an idea of the "typical" value for the column, so why is there a difference of almost 15 m^2 between them? To answer this question, we need to see how house sizes are distributed in our dataset. Let's look at two ways to visualize the distribution: a histogram and a boxplot.

In [33]:
```
VimeoVideo("656352616", h="6075fbacb5", width=600)
```

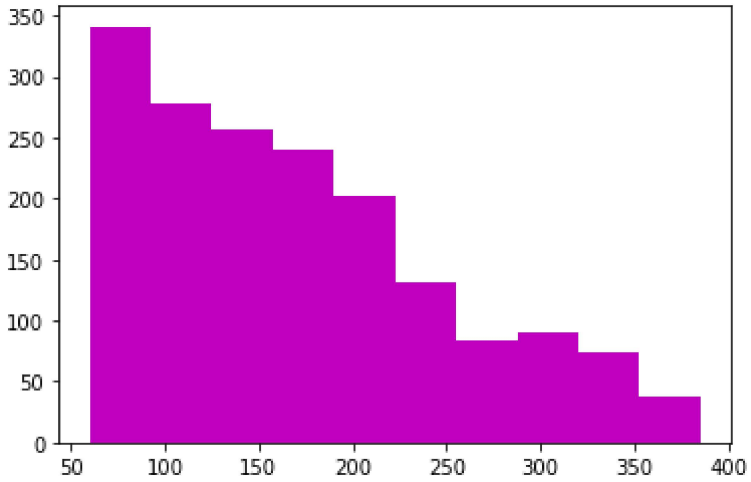
Out[33]:

Task 1.3.5: Create a histogram of `"area_m2"`. Make sure that the x-axis has the label `"Area [sq meters]"`, the y-axis has the label `"Frequency"`, and the plot has the title `"Distribution of Home Sizes"`.

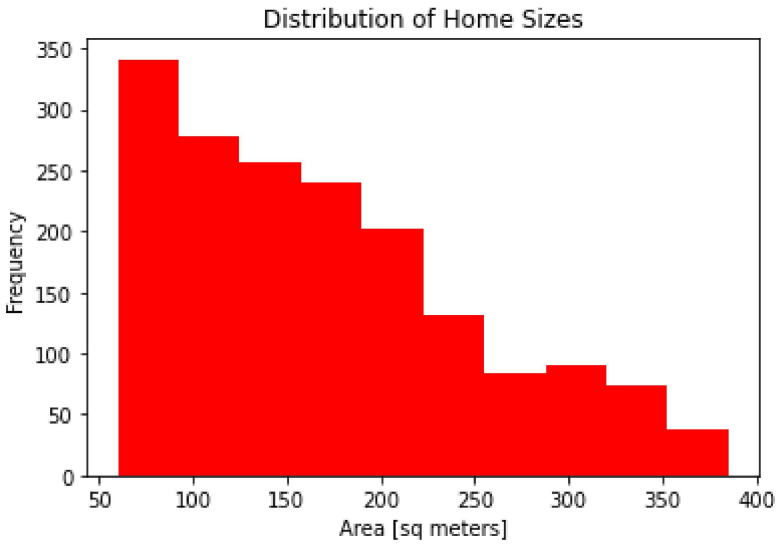
- What's a histogram?
- Create a histogram using Matplotlib.

In [37]:
```
plt.hist(df["area_m2"], color = 'm')
```

Out[37]:
```
(array([341., 278., 256., 240., 203., 132.,  84.,  91.,  73.,  38.]),
 array([ 60. ,  92.5, 125. , 157.5, 190. , 222.5, 255. , 287.5, 320. ,
        352.5, 385. ]),
 <BarContainer object of 10 artists>)
```



In [38]:
```
plt.hist(df["area_m2"], color = 'r')
plt.xlabel("Area [sq meters]")
plt.ylabel("Frequency")
plt.title("Distribution of Home Sizes");
```



Looking at our histogram, we can see that `"area_m2"` skews left. In other words, there are more houses at the lower end of the distribution (50–200m^2) than at the higher end (250–400m^2). That explains the difference between the mean and the median.

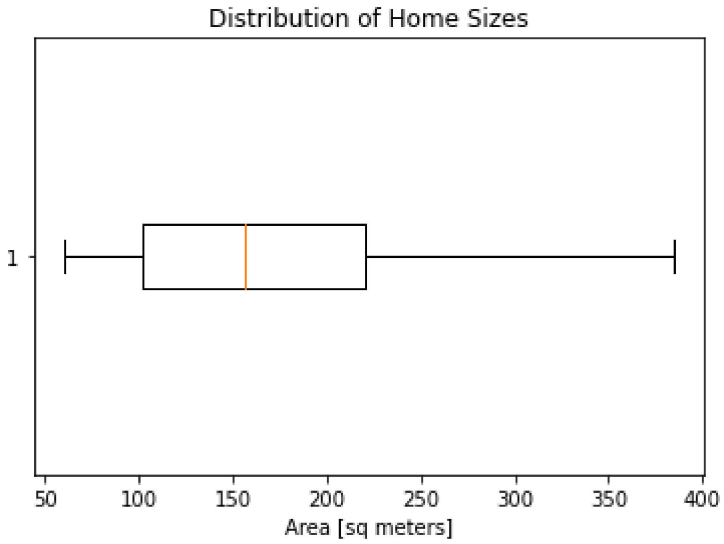
In [39]:
```
VimeoVideo("656352166", h="5531b6e160", width=600)
```

Out[39]:

Task 1.3.6: 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"`. How is the distribution and its left skew represented differently here than in your histogram?

- What's a boxplot?
- What's a skewed distribution?
- Create a boxplot using Matplotlib.

In [45]:
```
plt.boxplot(df["area_m2"], vert = False)
plt.xlabel("Area [sq meters]")
plt.title("Distribution of Home Sizes");
```



Does `"price_usd"` have the same distribution as `"price_per_m2"`? Let's use the same two visualization tools to find out.

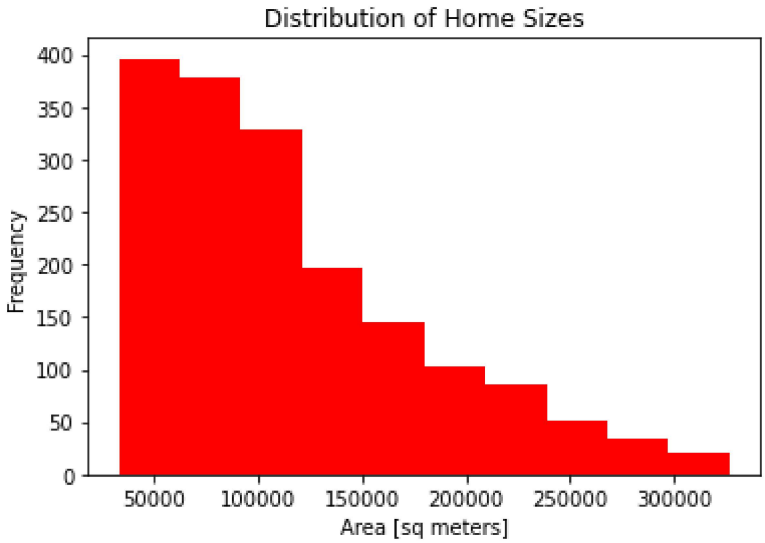
In [46]:
```
VimeoVideo("656351977", h="a0868bd01e", width=600)
```

Out[46]:

Task 1.3.7: 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"`.

- What's a histogram?
- Create a histogram using Matplotlib.

In [50]:
```
plt.hist(df["price_usd"], color = 'r')
plt.xlabel("Area [sq meters]")
plt.ylabel("Frequency")
plt.title("Distribution of Home Sizes");
```



Looks like `"price_usd"` is even more skewed than `"area_m2"`. What does this bigger skew look like in a boxplot?

In [51]:
```
VimeoVideo("656351234", h="44ca8af7ac", width=600)
```

Out[51]:

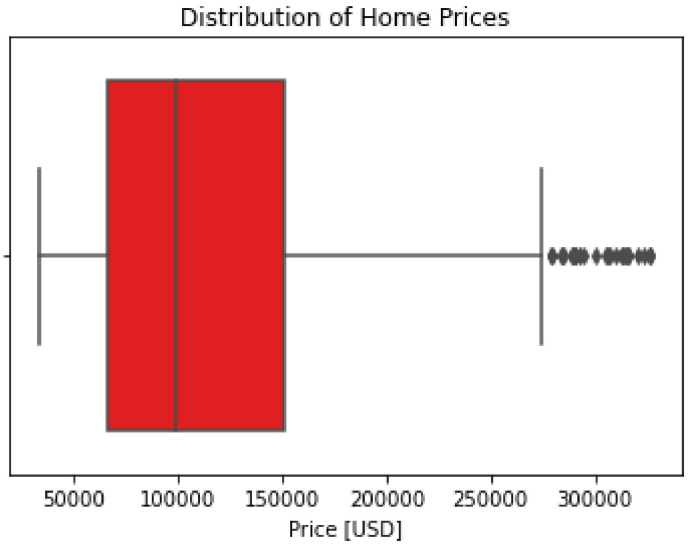
Task 1.3.8: Create a horizontal boxplot of `"price_usd"`. Make sure that the x-axis has the label `"Price [USD]"` and the plot has the title `"Distribution of Home Prices"`.

- What's a boxplot?
- What's an outlier?
- Create a boxplot using Matplotlib.

In [59]:
```
import seaborn as sns
sns.boxplot(df["price_usd"], color = 'r')
plt.xlabel("Price [USD]")
plt.title("Distribution of Home Prices");
```

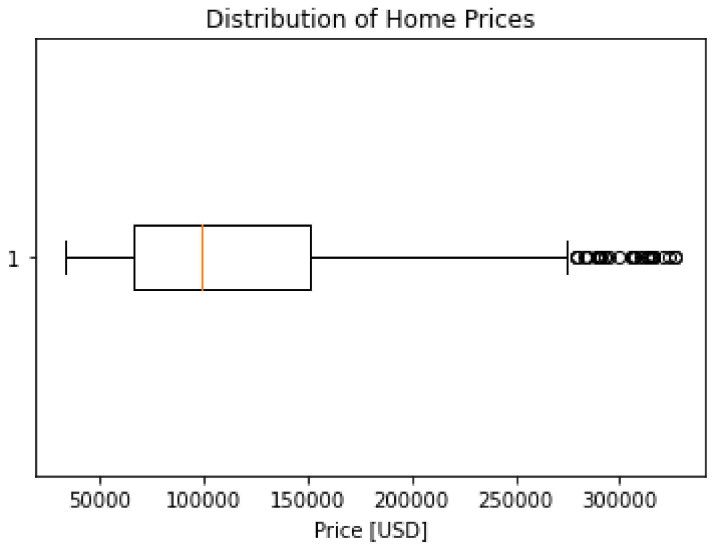
```
/opt/conda/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid posit
ional argument will be `data`, and passing other arguments without an explicit keyword w
ill result in an error or misinterpretation.
```



In [61]:
```
plt.boxplot(df["price_usd"], vert= False)
```

```
plt.xlabel("Price [USD]")
plt.title("Distribution of Home Prices");
```



Excellent job! Now that you have a sense of for the dataset, let's move to the next notebook and start answering some research questions about the relationship between house size, price, and location.
