Lab 1: Basic Data Understanding

Questions:

1. Data Understanding:

Task: Print the data types of each column and use descriptive statistics to understand the data.

Questions:

- Identify and justify the appropriateness of the data types for each attribute. Suggest changes if necessary.

```
data.dtypes

[10]:

ID              int64
Size(sqft)    float64
Bedrooms      float64
Badhrooms     float64
Location       object
House_Type     object
Year_Built    float64
Date_Sold      object
Price         float64
dtype: object
```

I suggest we drop all missing values, convert the 'Bedrooms', 'Badhrooms', and 'Year_Built' columns to an int data type. Also, convert the 'Date_Sold' column to a datetime data type.

- What does the statistical summary tell you about potential issues with data quality, such as range problems, missing values, or format inconsistencies?

The statistical summary tells us that there are problems with each attribute. Some are negative values and some have numbers that use natural logarithms.

- If there is a typo issue, fix it.

Changed 'Badhrooms' to 'Bathrooms'.

2. Identifying and Handling Missing Values:

Task: Identify missing values in the dataset and propose methods to handle them.

Questions:

- What patterns of missing data did you observe in the dataset?

```
data.isna().sum()

[6]:

ID              0
Size(sqft)     12
Bedrooms        8
Bathrooms      20
Location       10
House_Type     15
Year_Built     11
Date_Sold      10
Price           5
dtype: int64
```

In order to see if deleting any missing values would greatly change the shape of the data set, I gathered the total count of missing values for each attribute.

- How will you handle the missing values? Justify your approach.

```
data.dropna(inplace=True)
```

I decided it would be okay to drop the missing values because looking at the minimum number of missing values (20 for 'Bathrooms') and subtracting from the length of the data set (110), we can guess that making this change won't affect our statistics significantly.

3. Detecting and Correcting Invalid Entries:

Task: Identify and correct invalid entries in the dataset (e.g., negative values in columns where only positive values are appropriate, unrealistic dates, or other logical inconsistencies). Also, If there is a typo issue, fix it.

Questions:

- What invalid entries did you find in the dataset? Provide examples.

```
data.describe()
```

[3]:

	ID	Size(sqft)	Bedrooms	Badhrooms	Year_Built	Price
count	110.000000	98.000000	102.000000	90.000000	99.000000	1.050000e+02
mean	55.500000	17912.581633	14.813725	12.677778	11554.797980	1.424262e+06
std	31.898276	72463.968277	65.929114	50.381874	42931.458381	1.239105e+07
min	1.000000	-4529.000000	-3.000000	-3.000000	-1991.000000	-4.494570e+07
25%	28.250000	1415.500000	2.000000	1.000000	1918.500000	2.913350e+05
50%	55.500000	2544.500000	3.000000	2.000000	1949.000000	4.535310e+05
75%	82.750000	3803.500000	5.000000	3.000000	1990.000000	7.485310e+05
max	110.000000	497400.000000	500.000000	300.000000	200800.000000	7.481430e+07

Looking at the minimum value for each column, other than ‘ID’, I saw that there were negative values. For the maximum values under the ‘Size(sqft)’ column, I saw that there was an extremely large value. In the ‘Price’ column, I also saw that there were some extreme values not relative to the square feet of a property.

- Explain the steps you took to correct these invalid entries, and justify your methods.

```
data.head()
data_pos_nums = data[
(data['Size(sqft)'] > 0) &\
(data['Bedrooms'] > 0) &\
(data['Year_Built'] > 0) &\
(data['Price'] > 0) &\
(data['Price'] < 1000000) &\
(data['Bathrooms'] > 0)].copy()
```

[36]:

```
data_pos_nums
data_cleaned = data_pos_nums[
(data_pos_nums['Bathrooms'] < 4) &\
(data_pos_nums['Size(sqft)'] < 10000)].reset_index(drop=True)

data_cleaned_2 = data_cleaned.drop(
    index=([8,13,15,24]), axis=1).reset_index(drop=True).copy()
```

I hard coded conditional statements to filter for the cases in each quantitative column that were greater than zero. I filtered for all princes less than $1,000,000. Filtered for bathrooms that were less than four. Then I manually deleted cases that had a quantitative value in the ‘House_Type’ column.

4. Addressing Duplicate Records:

Task: Identify and remove duplicate records from the dataset.

Questions:

- How did you identify duplicate records in the dataset?

```
data_cleaned_2.drop_duplicates()
data_cleaned_2.index
```

```
[49]:

RangeIndex(start=0, stop=23, step=1)
```

I used `data_cleaned_2.index` to get the length of the data set. I used `.drop_duplicates` and then `data_cleaned.index` to compare lengths.

- What criteria did you use to decide which duplicates to remove, if any? Justify your approach.

I did not find any duplicates due to the lengths being the same after the method was used.

5. Data Range Issues:

Task: Identify and address any data range issues (e.g., values outside expected ranges, negative sizes, or dates in the future).

Questions:

- What range issues did you find in the dataset? Provide specific examples.

  Identified in question three.

- How did you address these range issues? Explain and justify your approach.

  Addressed in question three.

6. Format Inconsistencies:

Task: Identify and correct any format inconsistencies in the dataset (e.g., inconsistent date formats, units, or text formats).

Questions:

- What format inconsistencies did you find in the dataset? Provide examples.

```
data_cleaned_2.dtypes
[81]:

ID              int64
Size(sqft)    float64
Bedrooms      float64
Bathrooms     float64
Location       object
House_Type     object
Year_Built    float64
Date_Sold      object
Price         float64
dtype: object
```

In question one, I saw the data types for each column and realized they were defaults. For example: int64, float64, and objects.

- Explain how you standardized these formats and why it is important to do so.

```
data_cleaned_2 = data_cleaned_2.astype(
    {
        'ID':'int8',
        'Size(sqft)':'int8',
        'Bedrooms':'int8',
        'Bathrooms':'int8',
        'Location':'string',
        'House_Type':'string',
        'Year_Built':'int16',
        'Date_Sold':'datetime64[ns]',
        'Price':'float32'}
    , copy=True, errors='raise')
```

I converted datatypes to the minimum types, doing this would improve memory performance and make loading our data faster.

7. Misclassified Data:

Task: Detect and correct any misclassified data within the dataset (e.g., numeric data in text fields or vice versa).

Questions:

- What misclassified data did you identify in the dataset?

  This was identified in question three.

- How did you correct these misclassifications, and why did you choose these methods?

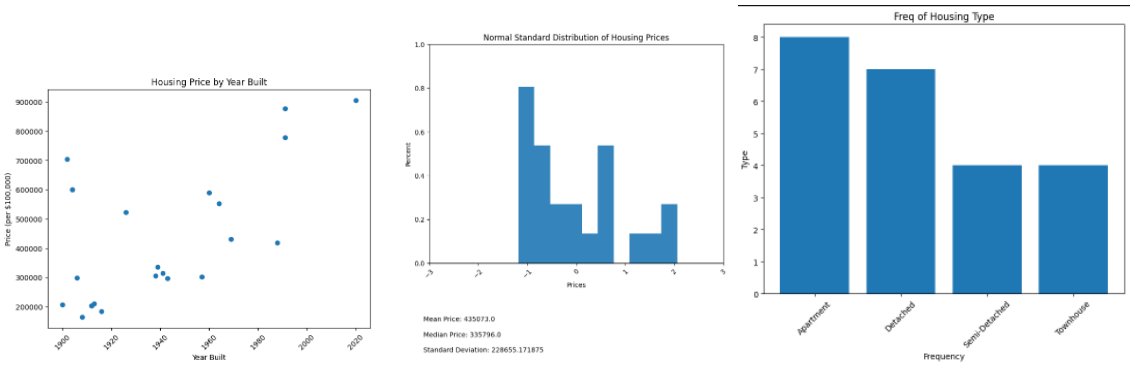
  This was addressed in question three.

8. Data Visualization:

Task: Use visualizations to identify patterns, inconsistencies, or outliers in the dataset.

Questions:

- Which visualizations did you use to explore the data, and what insights did they provide?



I used a scatter plot, histograms, and a bar chart.

- How did these visualizations help in identifying inconsistencies, outliers, or other data quality issues?

The scatter plot let me see the direction of housing prices dependent on the year it was made. I used a Normal Standard Distribution for the quantitative variables, which revealed that housing prices are positively skewed (which is a good thing for buyers). The bar chart was useful for seeing the count of housing types, which can help us understand how many are in the area.