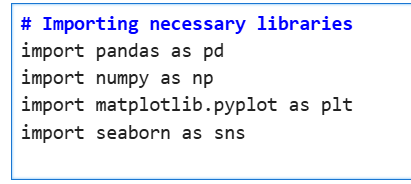
**Case Study-2**

Dataset: **mudah-apartment-kl-selangor**

Name: **E.R Harish**

**Output:**

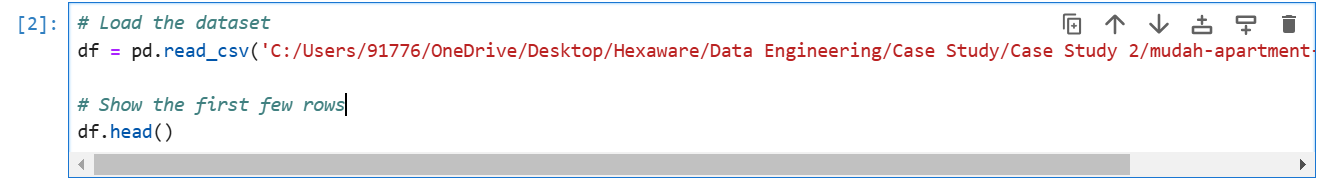
1. **Cell 1:**

****

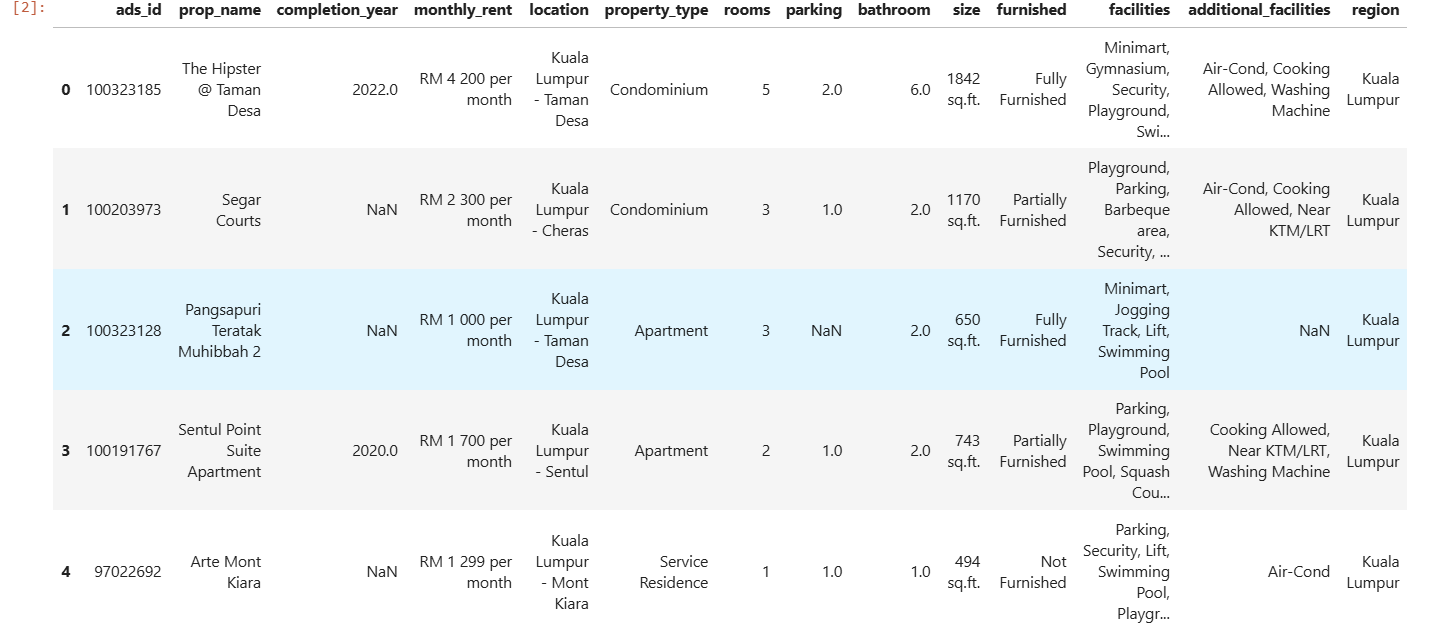
**Explanation:** This imports the following essential libraries for data analysis and visualization:

* **pandas (as pd)**: Used for data manipulation and analysis, especially working with tabular data.
* **numpy (as np)**: Provides support for numerical operations and arrays.
* **matplotlib.pyplot (as plt)**: A plotting library for creating static, animated, and interactive visualizations.
* **seaborn (as sns)**: Built on matplotlib, used for creating more advanced and attractive statistical plots.

1. **Cell 2:**

****

**Output:**

****

**Explanation:**

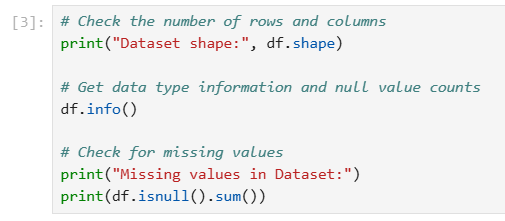
1. **Load the dataset:**

* **pd.read\_csv('path\_to\_file'):** Reads a CSV file located at the specified path into a pandas DataFrame named df.
* The file contains data related to **apartments in KL Selangor.**

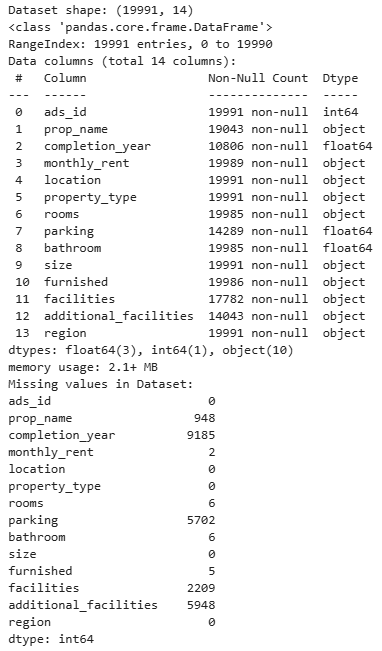
1. **Display the first few rows:**

* **df.head():** Displays the first 5 rows of the DataFrame by default. This helps to get a quick overview of the dataset's structure and content.

1. **Cell 3:**

****

**Output:**

****

**Explanation:**

1. **Check the shape of the dataset:**

* **df.shape:** Returns the number of rows and columns in the dataset.

1. **Get data type information and memory usage:**

* **df.info():** Displays column names, data types, non-null counts, and memory usage.

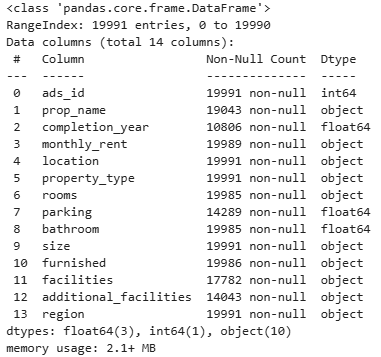
1. **Check for missing values:**

* **df.isnull().sum():** Computes the total number of missing (NaN) values in each column.

1. **Cell 4:**

****

**Output:**

****

**Explanation:**

1. **Column names and types:**

* Lists all columns along with their respective data types (e.g., int64, float64, object).

1. **Non-null counts:**

* Shows the number of non-missing (non-null) values in each column.
* Useful for identifying columns with missing data.

1. **Memory usage:**

* Displays the amount of memory being used by the dataset.

1. **Cell 5:**

****

**Explanation:**

1. **df.dropna(inplace=True):**

* **dropna():** Drops all rows that contain at least one missing (NaN) value.
* **inplace=True:** Modifies the original DataFrame directly instead of returning a new one.

1. **Cell 6:**

****

**Output:**

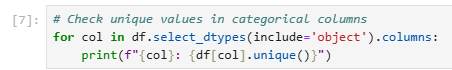
****

**Explanation:**

1. **df.describe():**

* Provides a statistical summary of numerical columns in the DataFrame.

1. **Cell 7:**

****

**Output:**

****

**Explanation:**

1. **Select categorical columns:**

* **df.select\_dtypes(include='object'):** Filters columns with data type object (typically used for categorical or string data).

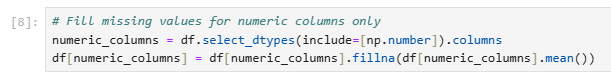
1. **Iterate over the categorical columns:**

* Loops through each column in the filtered DataFrame.

1. **Print unique values:**

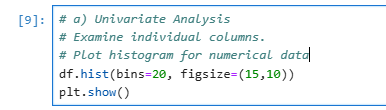
* **df[col].unique():** Retrieves and prints all unique values in the current column.

1. **Cell 8:**

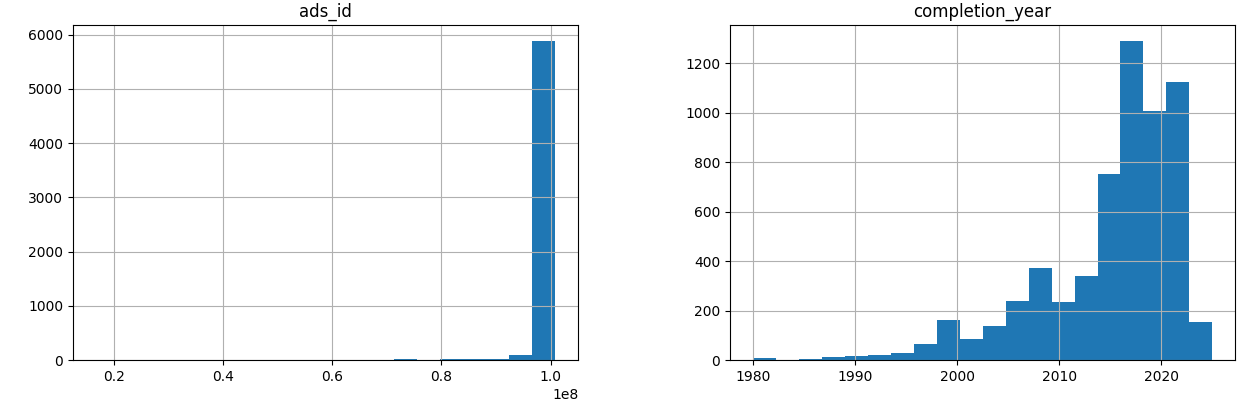
****

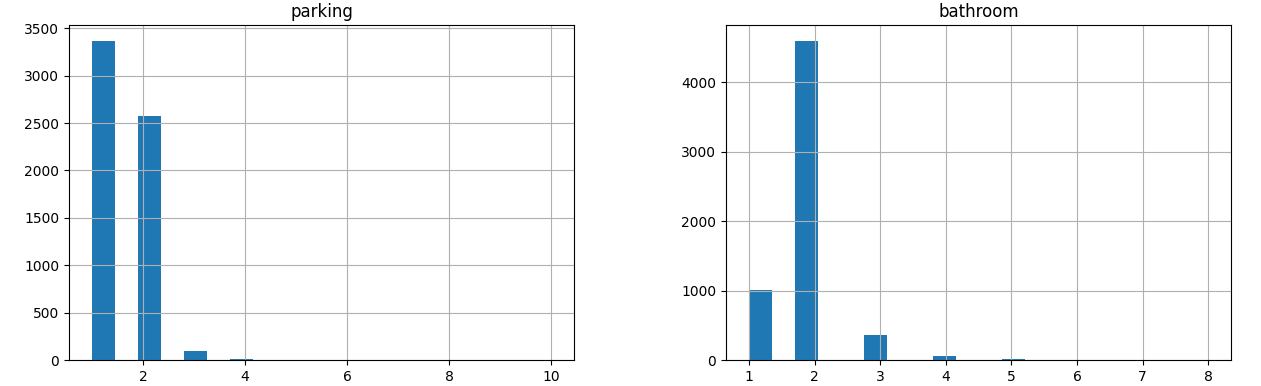
**Explanation:**

1. **df.select\_dtypes(include=[np.number]).columns:** Selects columns with numeric data types (integers or floats) in the DataFrame.
2. **df[numeric\_columns]:** Selects the numeric columns from the DataFrame.
3. **.fillna(df[numeric\_columns].mean()):** Fills missing values (NaN) in these numeric columns with the mean of the respective column.
4. **Cell 9:**

****

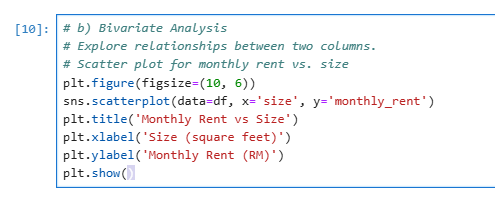
**Output:**

****

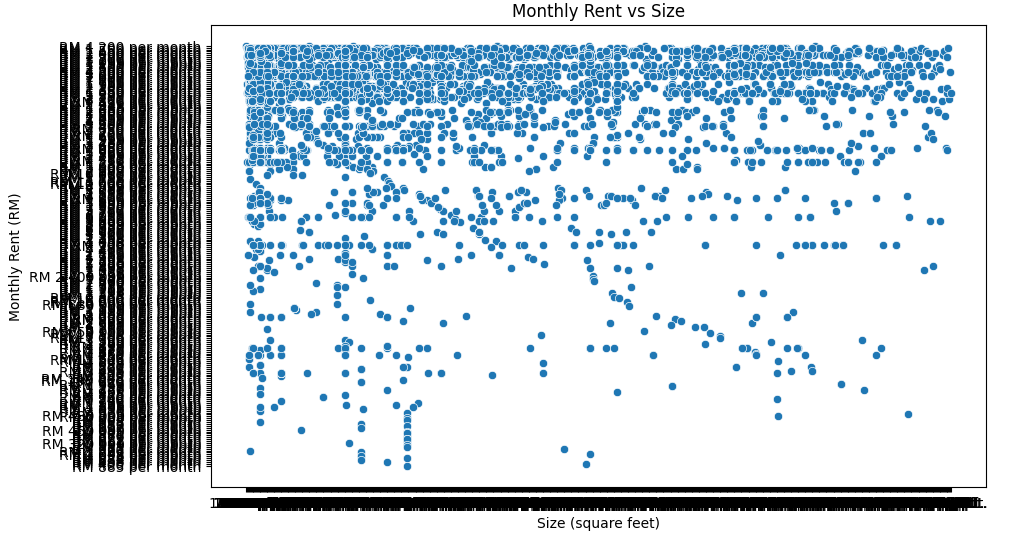
****

**Explanation:**

1. **df.hist(bins=20, figsize=(15,10)):** This creates histograms for each numeric column in the DataFrame. The bins=20 parameter specifies that the data will be divided into 20 bins for each histogram. The figsize=(15,10) adjusts the overall size of the plot for better visualization.
2. **plt.show():** Displays the histograms.
3. **Cell 10:**

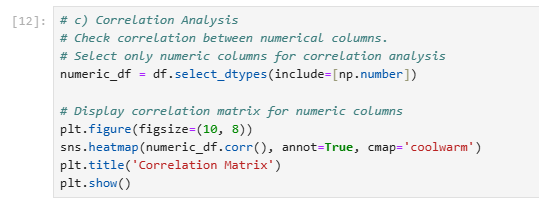
****

**Output:**

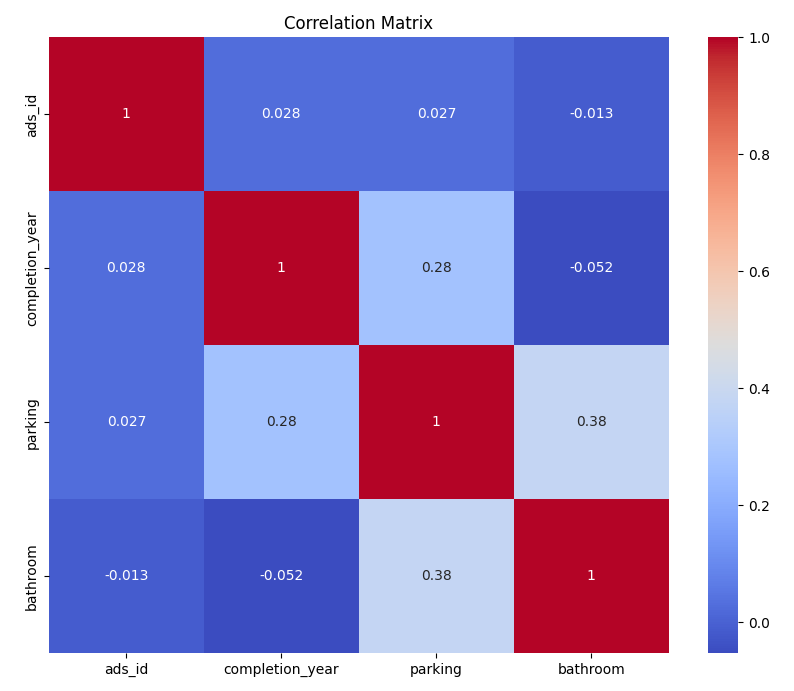
****

**Explanation:**

1. **plt.figure(figsize=(10, 6)):** Sets the figure size for the plot.
2. **sns.scatterplot(data=df, x='size', y='monthly\_rent'):** Creates a scatter plot with size on the x-axis and monthly\_rent on the y-axis using Seaborn's scatterplot function. It visualizes how changes in size are related to changes in monthly rent.
3. **plt.title('Monthly Rent vs Size'):** Adds a title to the plot.
4. **plt.xlabel('Size (square feet)'):** Labels the x-axis as "Size (square feet)".
5. **plt.ylabel('Monthly Rent (RM)'): L**abels the y-axis as "Monthly Rent (RM)".
6. **plt.show():** Displays the plot.
7. **Cell 11:**

****

**Output:**

****

**Explanation:**

1. **numeric\_df = df.select\_dtypes(include=[np.number]):** This line selects only the numeric columns from the DataFrame (df) for correlation analysis.
2. **numeric\_df.corr():** Computes the correlation matrix, which shows the pairwise correlation between the numeric columns. The values range from -1 (perfect negative correlation) to 1 (perfect positive correlation).
3. **plt.figure(figsize=(10, 8)):** Adjusts the figure size of the plot.
4. **sns.heatmap(numeric\_df.corr(), annot=True, cmap='coolwarm'):** Creates a heatmap of the correlation matrix using Seaborn's heatmap function. The annot=True argument displays the correlation values in the heatmap cells, and cmap='coolwarm' specifies the color palette for the heatmap (blue for negative correlations and red for positive correlations).
5. **plt.title('Correlation Matrix'):** Adds a title to the heatmap**.**
6. **plt.show():** Displays the heatmap.
7. **Summary and Conclusion:**

* **Summary of Findings:**

Throughout this exploratory data analysis (EDA), several key insights emerged regarding the apartment rental market in Malaysia, specifically in Kuala Lumpur and Selangor. Here are the main observations:

* Rental Prices and Apartment Size: A positive correlation between apartment size (square footage) and monthly rental prices was observed, meaning larger apartments generally command higher rent. However, the scatter plot also showed outliers, which could indicate high-demand areas or luxury apartments with higher-than-average rents for a given size.
* Location Impact: The data showed that location significantly impacts rental prices. Apartments in premium locations in Kuala Lumpur, such as the city center, tend to have higher rental rates than those in surrounding areas. This suggests that location, alongside apartment size, is a major factor in determining rental prices.
* Property Type and Amenities: Properties with more amenities or additional facilities, such as parking, security, and furnished options, showed a tendency for higher rental prices. This is in line with expectations, as additional features enhance property value and attractiveness.
* Room and Bathroom Count: A moderate correlation was observed between the number of rooms and bathrooms with rental prices, likely because larger apartments with more rooms are more suitable for families or groups, leading to higher demand.
* **Conclusion:**

This analysis provided insights into the factors influencing apartment rental prices in Malaysia, highlighting the importance of size, location, and amenities. By understanding these factors, renters and real estate investors can make more informed decisions, and property managers can better tailor their offerings to meet market demand. The EDA also emphasized the importance of data cleaning, as it affects the quality and reliability of the analysis.

1. **Interesting Insights and Graphs**

* Scatter Plot of Monthly Rent vs. Size: This graph demonstrated the positive relationship between apartment size and rental price, allowing for easy identification of outliers.
* Heatmap of Correlation Matrix: The correlation matrix visually displayed the relationships between numerical variables, highlighting which features are most associated with higher rental prices.
* Bar Plot of Average Rent by Location: This plot provided a clear view of how average rental prices vary across different regions, with central areas showing higher averages.

1. **Ideas for Future Work**

* **Incorporate Additional Datasets:** To deepen the analysis, we could incorporate additional data, such as:
* **Economic indicators** (e.g., average income in different regions) to study affordability.
* **Public transportation proximity** to explore its impact on rental prices.
* **Crime rates by neighborhood** to assess the influence of safety on rental values.
* **Time Series Analysis:** If rental data were available across different years or months, a time series analysis could reveal rental price trends over time, showing how prices have evolved and predicting future trends.
* **Sentiment Analysis on Property Descriptions:** Analyzing the sentiment or keyword frequencies in property descriptions could uncover additional factors influencing rental decisions.
* **Machine Learning Models:** Using this cleaned dataset, machine learning models such as linear regression or decision trees could predict rental prices based on property features, providing valuable insights for property valuation.

1. **Resources:**

Here are some resources that were helpful during this analysis:

* Seaborn Documentation: For creating statistical visualizations and heatmaps: Seaborn Documentation
* Pandas Documentation: To explore and manipulate data: Pandas Documentation
* NumPy Documentation: For handling numerical operations efficiently: NumPy Documentation
* Matplotlib Documentation: For plotting and data visualization: Matplotlib Documentation