**Experiment 2**

**Shashwat Tripathi**

**D15A Batch C  
RollNo: 64**

**AIM : Perform following data visualization and exploration on your selected dataset.**

**1. Create bar graph, contingency table using any 2 features.**

**2. Plot Scatter plot, box plot, Heatmap using seaborn.**

**3. Create histogram and normalized Histogram.**

**4. Describe what this graph and table indicates.**

**5. Handle outlier using box plot and Interquartile range.**

**THEORY :**

**Matplotlib:**

Matplotlib is a fundamental and highly versatile data visualization library in the Python ecosystem. Its architecture revolves around the concept of Figures and Axes, where a Figure represents the overall window or page, and Axes are individual subplots or charts within that window. The library offers a comprehensive set of plotting functions for creating various types of visualizations, ranging from simple line plots and scatter plots to more complex bar plots and histograms. Matplotlib provides users with extensive customization options, allowing precise control over plot aesthetics, including colors, linestyles, markers, and more. It supports LaTeX-style mathematical expressions for labels and titles, adding to its flexibility. Additionally, Matplotlib supports various backends, enabling users to choose different output formats according to their specific needs.

**Architecture:**

Matplotlib has a hierarchical structure where a Figure contains one or more Axes (subplots). The Figure is the overall window or page where visualizations are drawn, and Axes are the individual plots or charts within that window.

**Plotting Functions:**

Matplotlib provides a variety of functions for creating different types of plots. The most common are plt.plot() for line plots, plt.scatter() for scatter plots, plt.bar() for bar plots, and so on. Users can customize plots by setting parameters such as colors, labels, titles, and legends.

**Customization and Styling:**

Matplotlib offers extensive customization options to control the appearance of plots. Users can modify colors, linestyles, markers, grid lines, and more. Additionally, Matplotlib supports LaTeX-style mathematical expressions for labels and titles.

**Backends:**

Matplotlib supports various backends, allowing users to choose different output formats (e.g., display in a Jupyter notebook, save as an image file, etc.). The default backend is usually sufficient for most use cases, but users can switch to others for specific requirements.

**SEABORN:**

Built on top of Matplotlib, Seaborn is a statistical data visualization library designed to simplify the creation of attractive and informative statistical graphics. While Matplotlib provides a robust foundation for general-purpose plotting, Seaborn specializes in statistical plotting, making it particularly useful for data analysts exploring complex relationships within datasets. Seaborn introduces higher-level abstractions that enhance the aesthetics of plots, offering default color palettes and themes that contribute to visually appealing visualizations. Its integration with Pandas DataFrames streamlines the process of working with structured data, allowing users to pass DataFrame columns directly to Seaborn functions. Seaborn is known for its capability to create complex visualizations, including pair plots for exploring pairwise relationships, heatmaps for visualizing matrix-like data, and cluster maps for hierarchical clustering. The library's emphasis on statistical visualization and its seamless integration with Matplotlib make it a valuable tool for researchers and analysts seeking to communicate insights effectively through data visualizations.

**1. Built on Matplotlib:**

Seaborn is a statistical data visualization library in Python that is built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the process of creating complex visualizations and enhances the aesthetics.

**2. Statistical Plots:**

Seaborn specializes in statistical plotting and excels in visualizing complex relationships in datasets. It includes functions for creating informative visualizations such as scatter plots with regression lines, box plots with additional information, and violin plots.

**3. Aesthetics and Themes:**

Seaborn enhances the visual appeal of plots with its default color palettes and themes. It simplifies the process of making plots aesthetically pleasing by providing preset themes and color palettes. Users can also customize these settings according to their preferences.

**4. Integration with Pandas:**

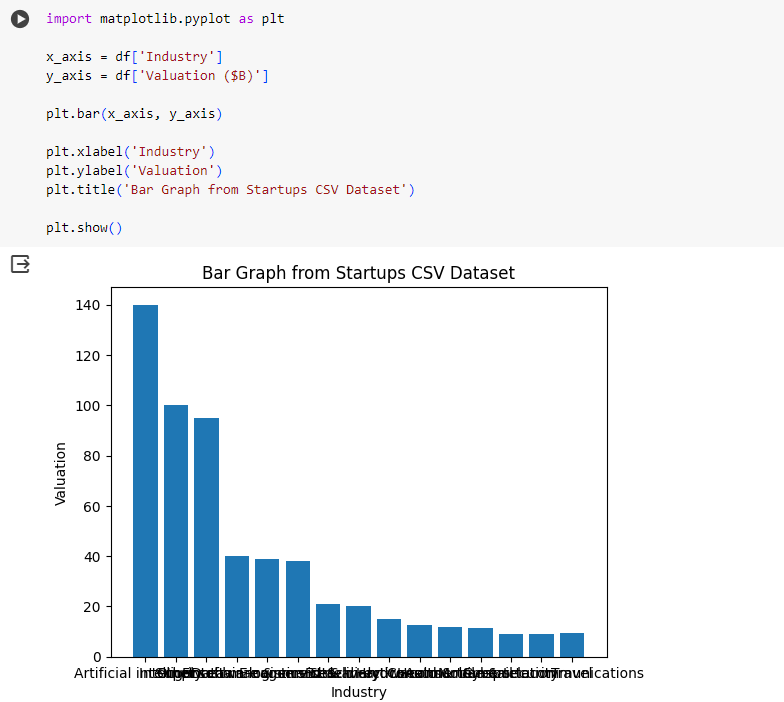
Seaborn seamlessly integrates with Pandas DataFrames, making it easy to work with structured data. Users can directly pass DataFrame columns to Seaborn functions, and Seaborn will handle the plotting details, reducing the need for manual data manipulation.

**5. Complex Visualizations:**

Seaborn is particularly powerful for creating complex visualizations, such as pair plots for exploring pairwise relationships in a dataset, heatmaps for visualizing matrix-like data, and cluster maps for hierarchical clustering.

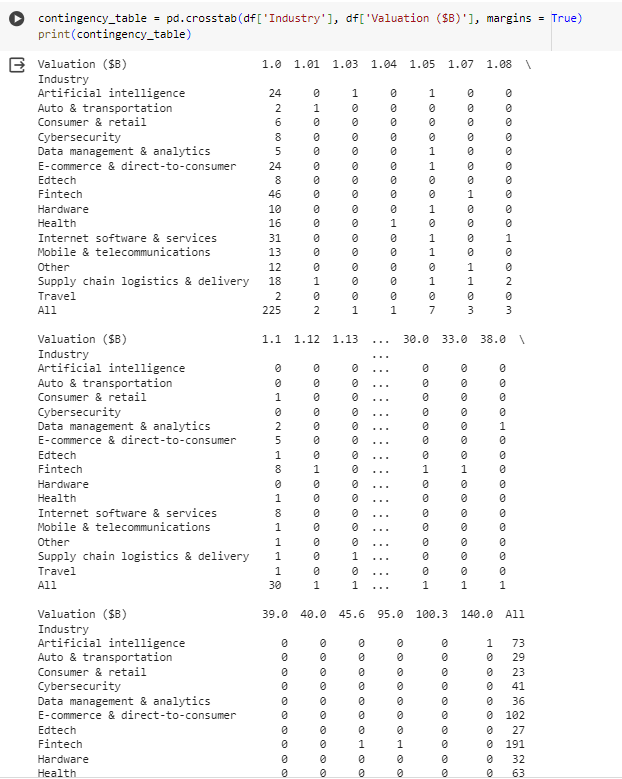
1. Bar Graph:

A bar graph is a visual representation of categorical data where each category is represented by a bar. You can use the plt.bar function in matplotlib or sns.barplot in seaborn to create a bar graph.



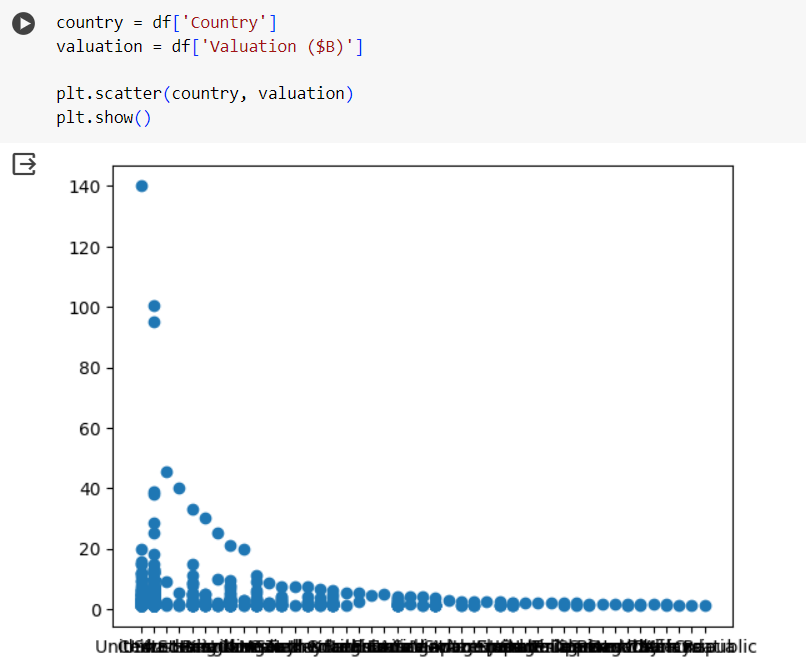
Contingency Table:

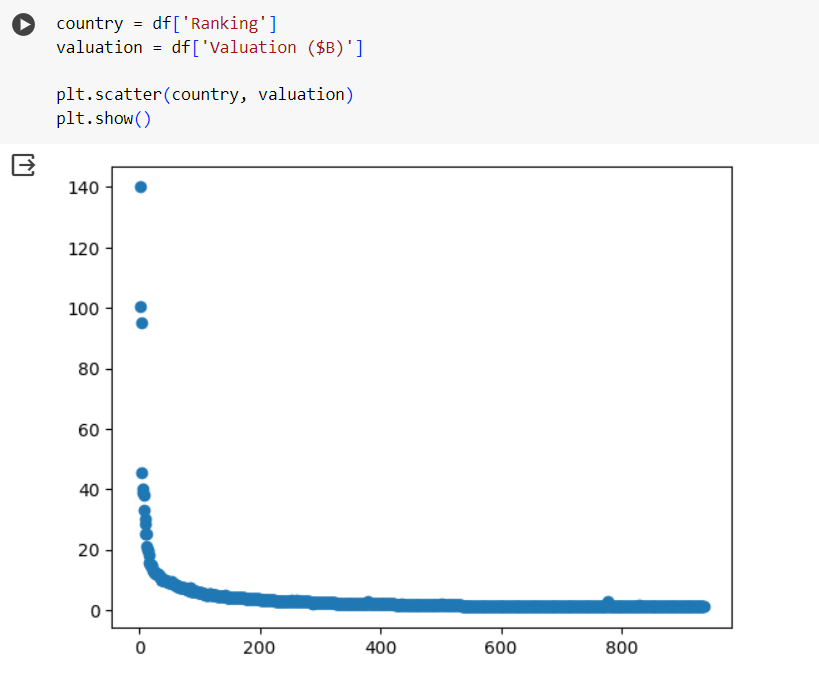
A contingency table is a table that shows the frequency distribution of two categorical variables. You can use pd.crosstab in pandas to create a contingency table.



Scatter Plot:

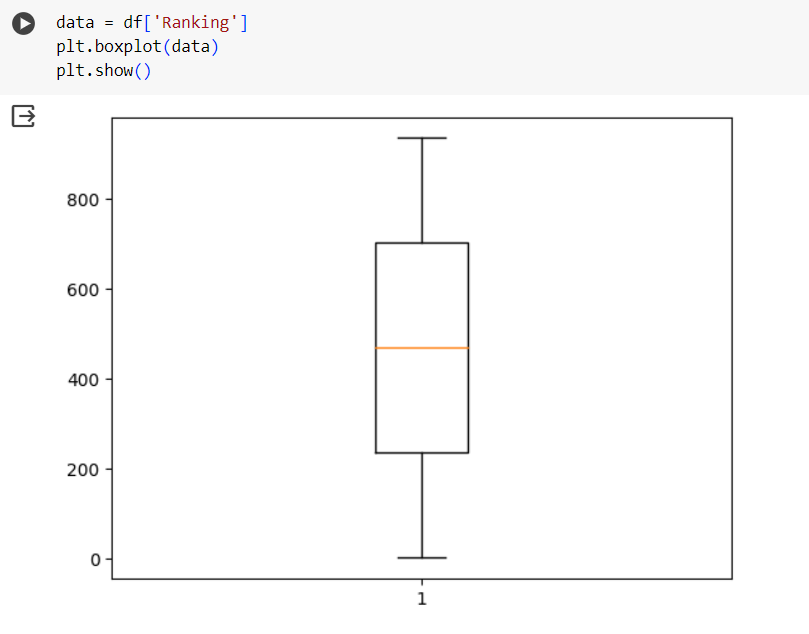
A scatter plot is a graphical representation of the relationship between two continuous variables. Each point on the plot represents an observation, with one variable on the x-axis and another on the y-axis. Scatter plots are useful for visualizing patterns, trends, and correlations between variables.

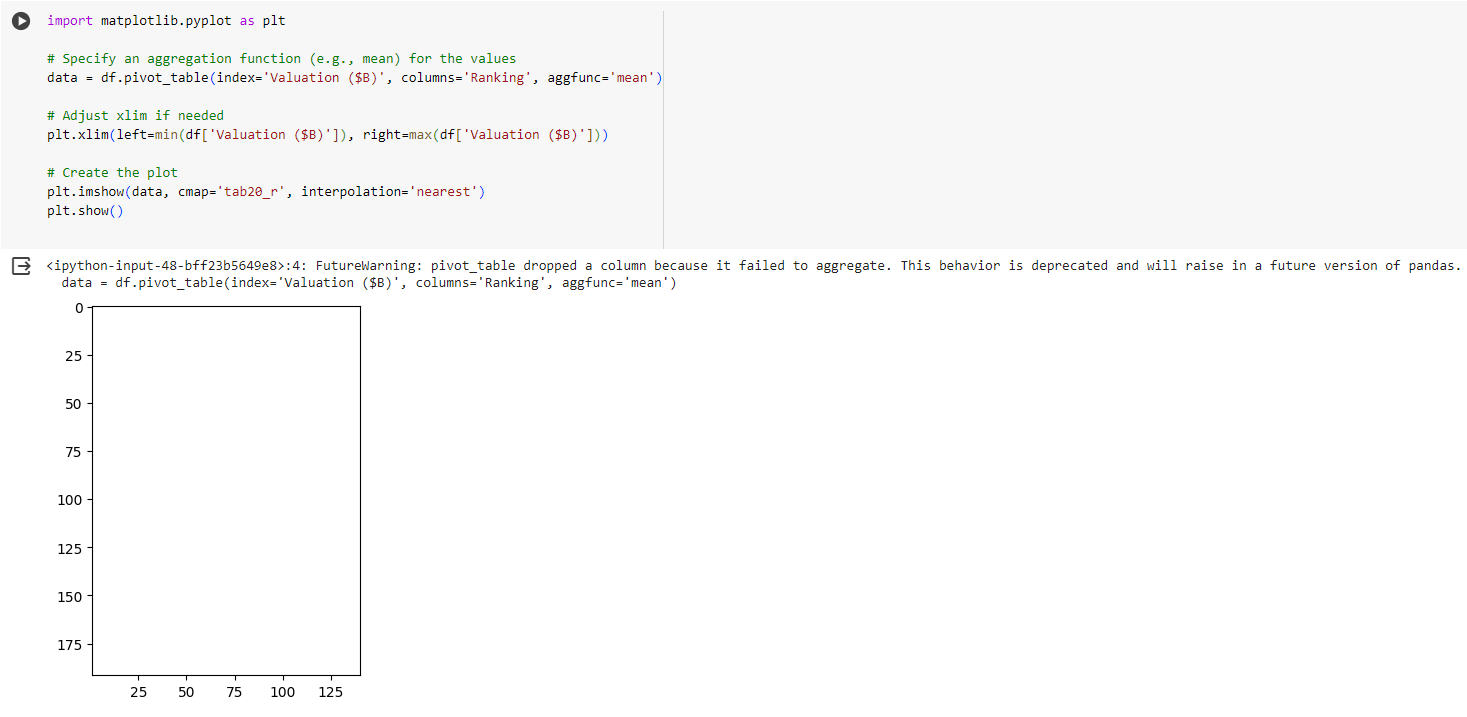




Box Plot:

A box plot (box-and-whisker plot) displays the distribution of a continuous variable. It provides a summary of the central tendency, spread, and identifies potential outliers. The box represents the interquartile range (IQR), while the whiskers extend to the minimum and maximum values within a certain range.



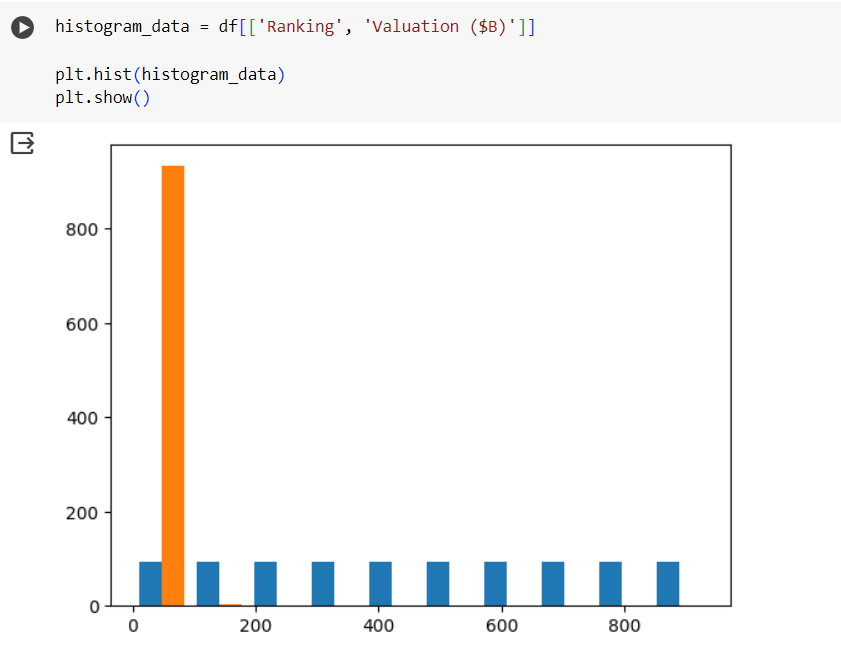


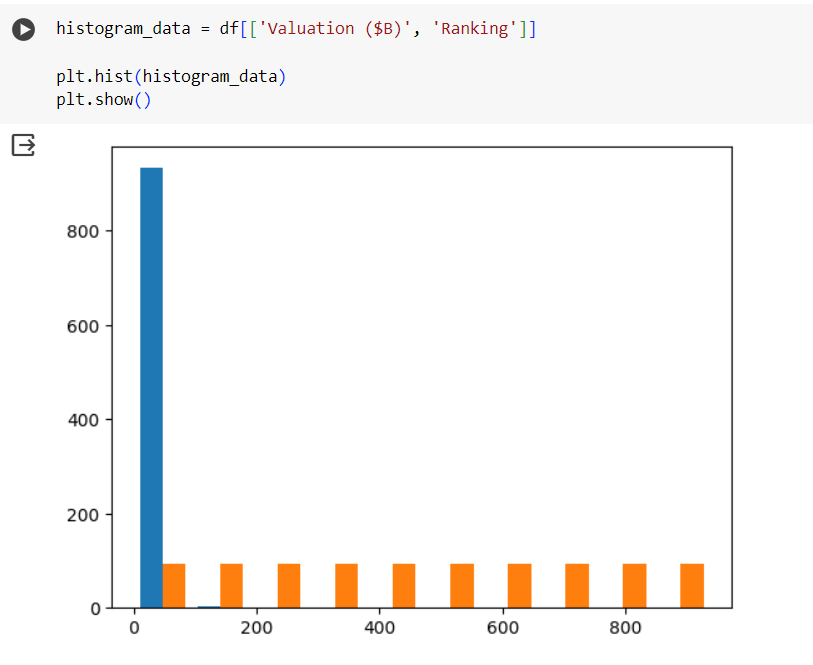
Histogram:

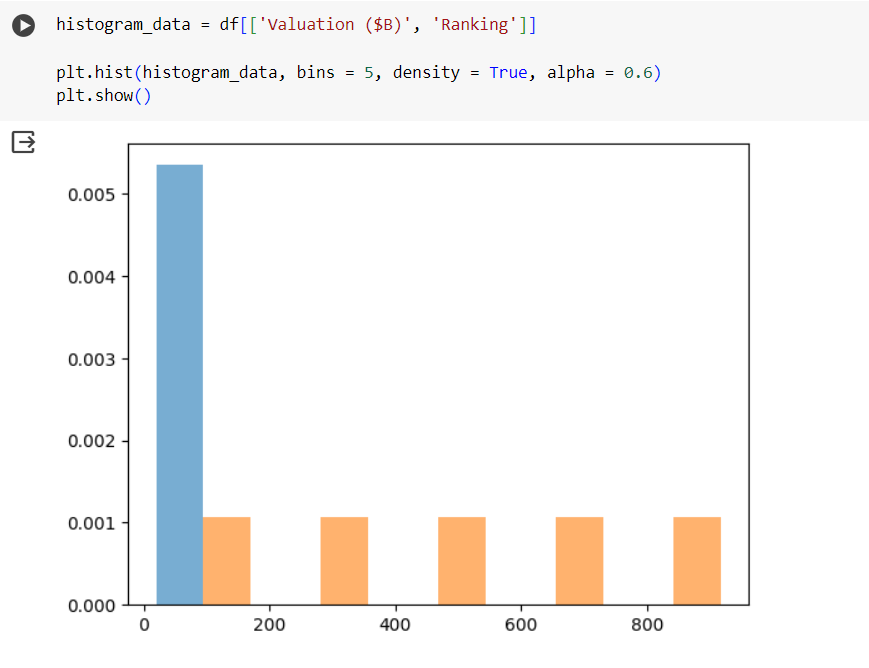
A histogram is a graphical representation of the distribution of a continuous variable. It divides the range of values into bins and shows the frequency or count of observations within each bin. Histograms provide insights into the shape and spread of the data distribution.

Normalized Histogram:

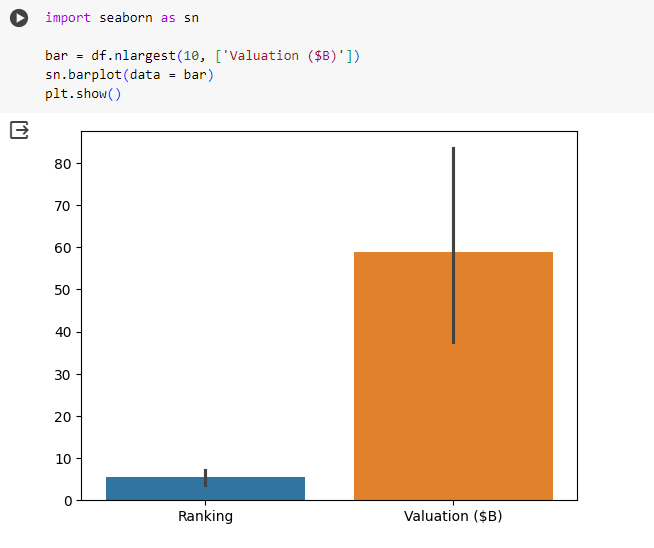
A normalized histogram represents the relative frequencies or proportions of observations in each bin, providing a normalized view of the distribution. It is useful for comparing distributions of variables with different scales or sample sizes.

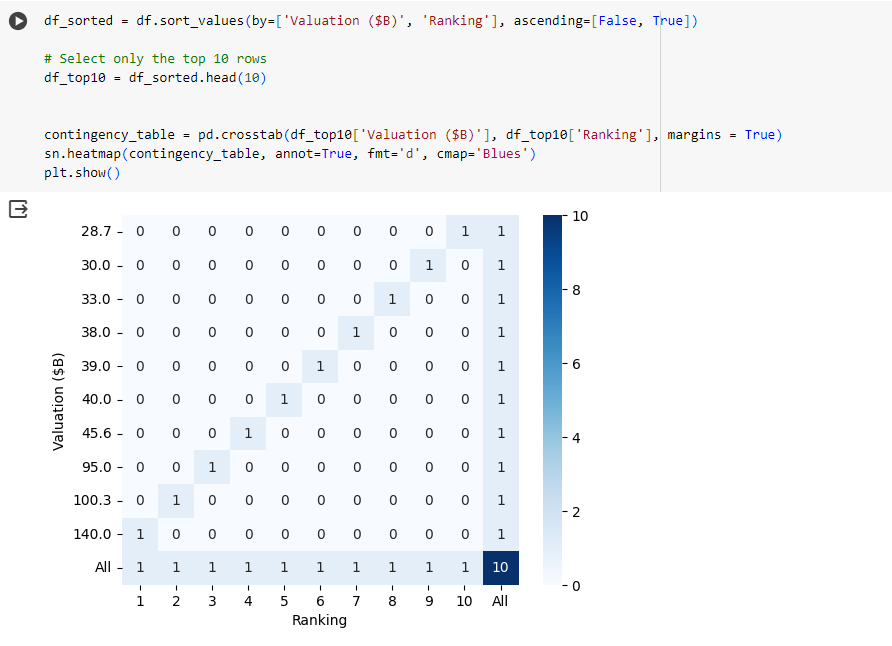


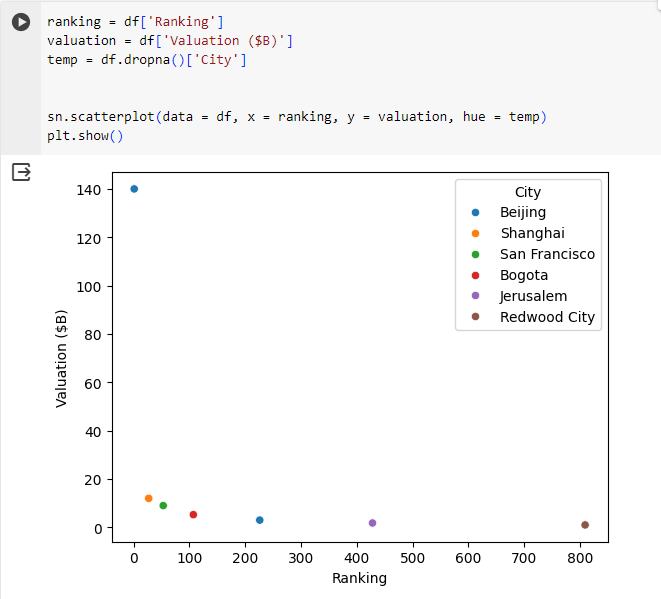


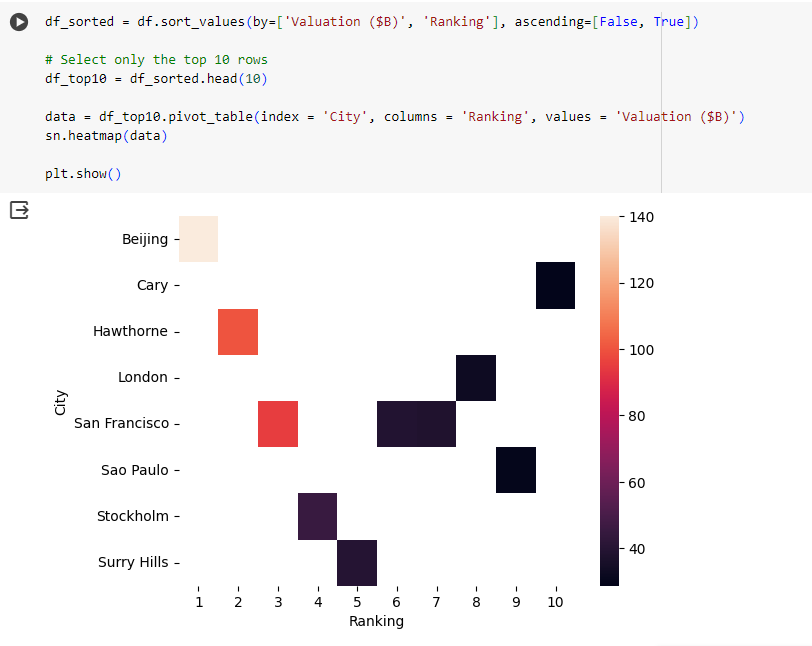


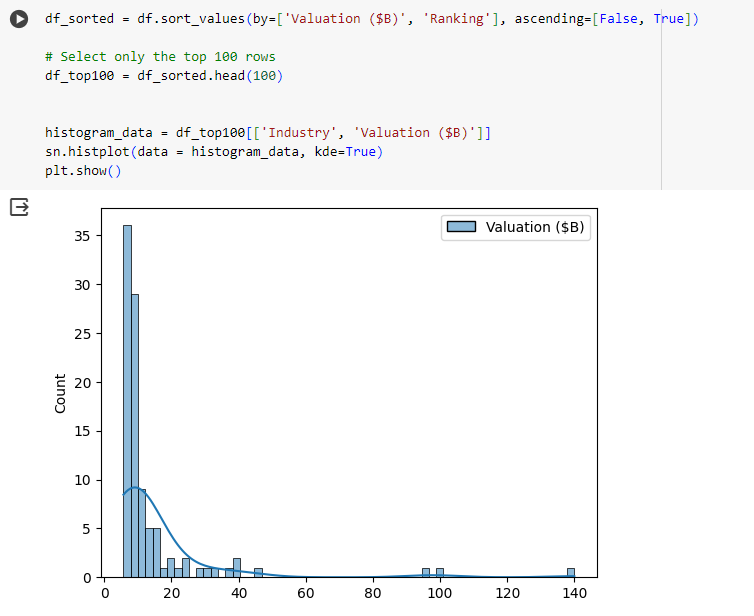
**USING SEABORN**

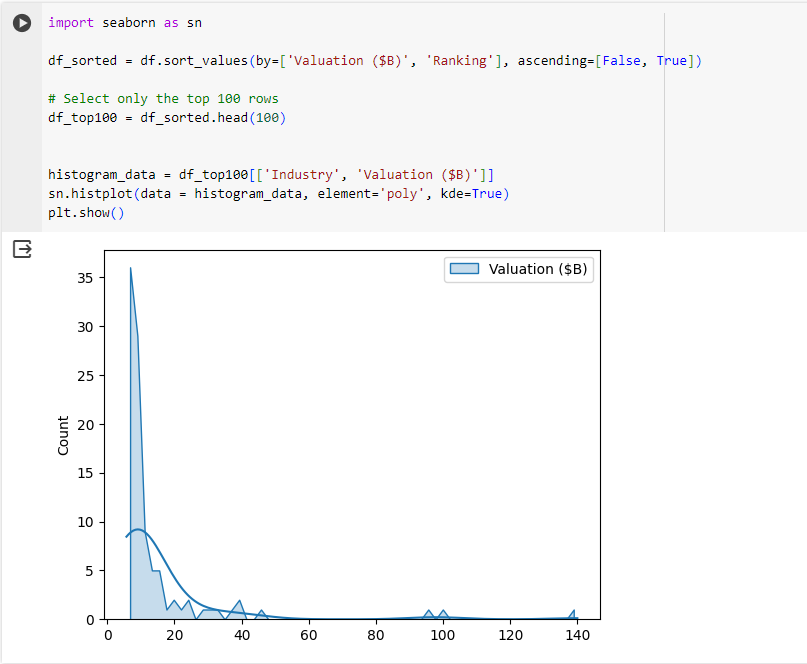




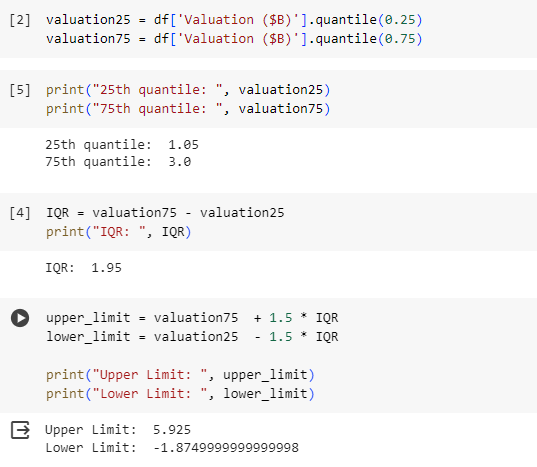


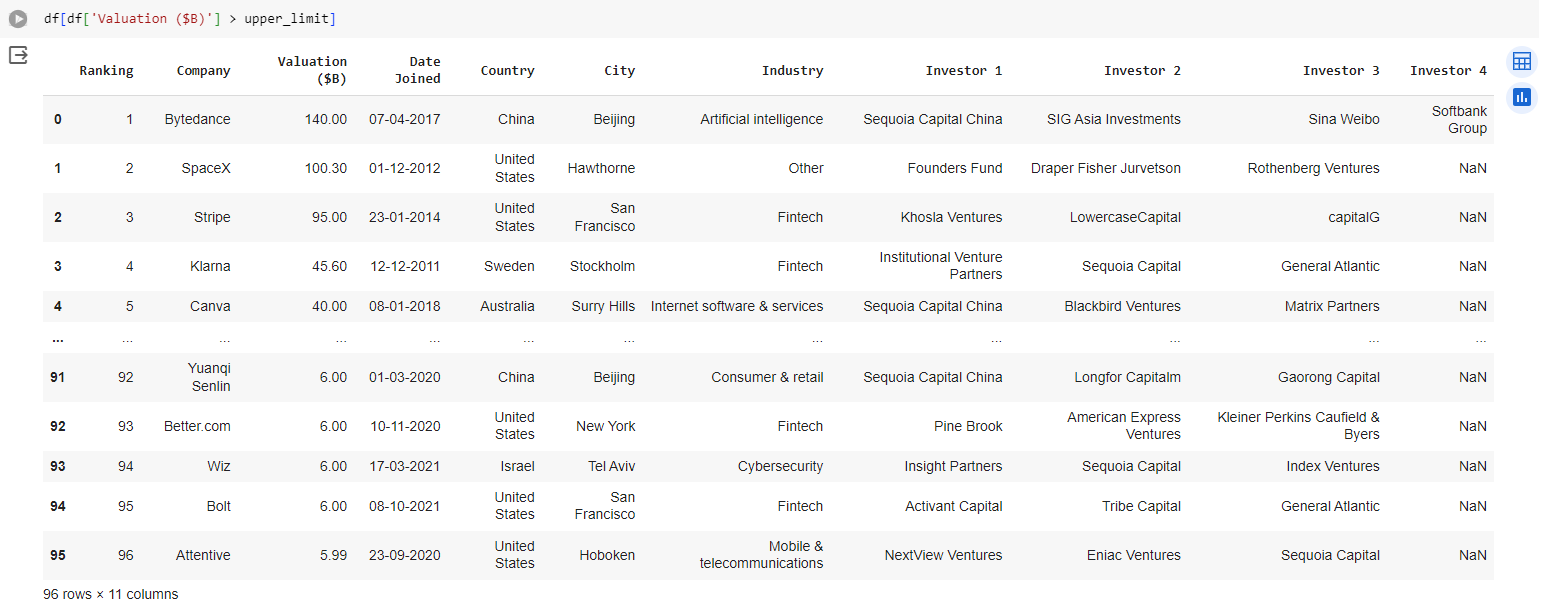


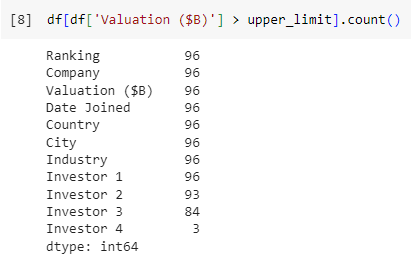


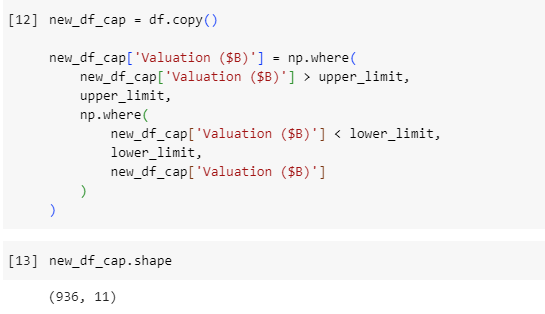


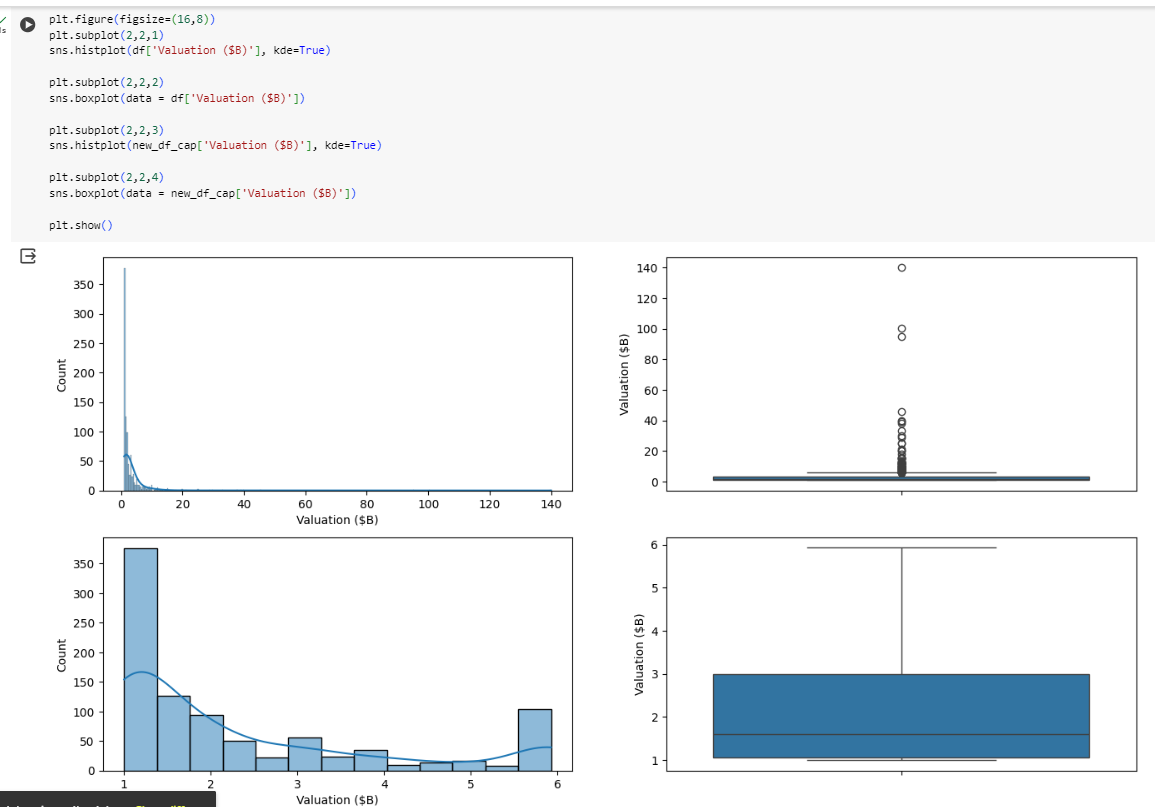
**Handle outlier using box plot and Interquartile range.**











**CONCLUSION :** By applying these visualization and exploration techniques, data analysts can gain valuable insights into the characteristics and relationships within a dataset, facilitating better decision-making and understanding of the underlying patterns.