13-11-2024

Training Day – 41

November 13, Wednesday

- ***Topic:* Visualizing Cleaned Data**
  - Created histograms and scatter plots for cleaned datasets.
  - Example: Visualized the distribution of sales data.

Visualizing cleaned data is an essential step in the data analysis process. Once you've processed and cleaned your data (by removing outliers, handling missing values, normalizing, etc.), visualizations help uncover patterns, trends, and insights. Here are key visualization techniques to consider for cleaned data:

1. Histograms

- **Use:** To visualize the distribution of numerical variables.
- **Why:** Helps identify skewness, normality, and outliers in the data.
- **Tools:** Matplotlib, Seaborn, Plotly.

2. Box Plots

- **Use:** To summarize the distribution of a variable and show outliers.
- **Why:** Provides a five-number summary (minimum, Q1, median, Q3, maximum) and identifies anomalies.
- **Tools:** Matplotlib, Seaborn.

3. Bar Charts

- **Use:** To compare categorical variables.
- **Why:** Visualizes the frequency or proportion of categories.
- **Tools:** Matplotlib, Seaborn, Plotly.

4. Scatter Plots

- **Use:** To visualize relationships between two continuous variables.
- **Why:** Helps identify correlations or trends.
- **Tools:** Matplotlib, Seaborn, Plotly.

5. Pair Plots

- **Use:** To visualize relationships between multiple continuous variables.
- **Why:** Helps identify patterns or trends between variables in a multi-dimensional dataset.

6. Line Charts

- **Use:** To show trends over time.
- **Why:** Ideal for time-series data to observe changes over time.
- **Tools:** Matplotlib, Plotly.

7. Pie Charts

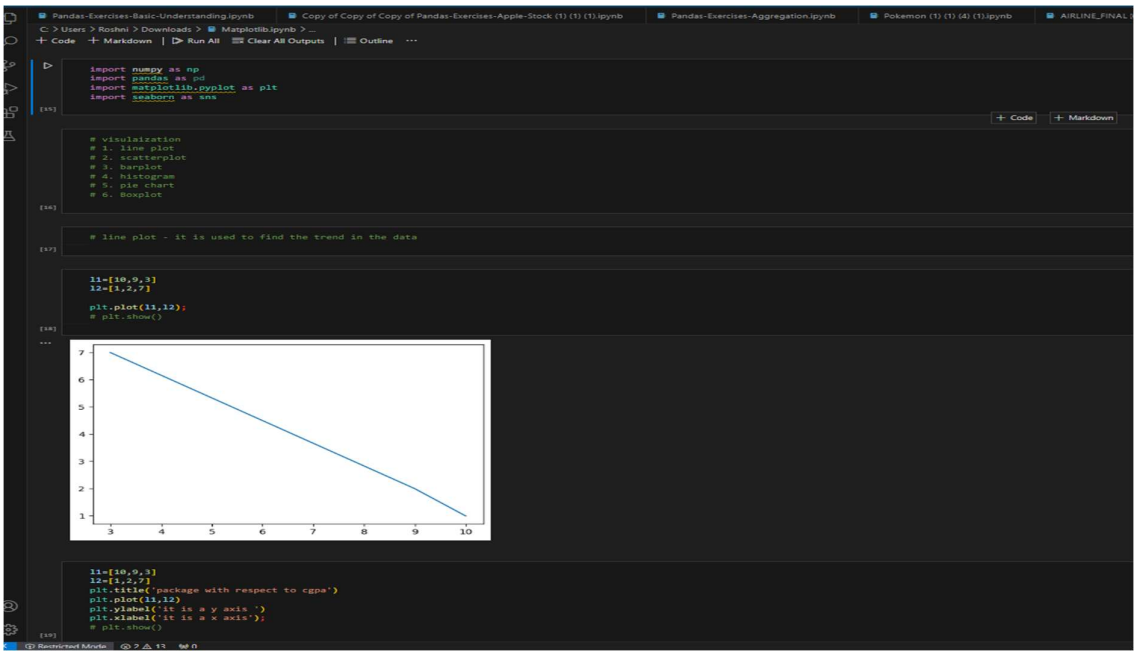
- **Use:** To represent proportions of categorical variables.
- **Why:** Helps quickly understand the composition of a variable.
- **Tools:** Matplotlib.

8. Violin Plots

- **Use:** To show the distribution of a variable across different categories.
- **Why:** Combines box plot and density plot to provide a deeper understanding of the data.
- **Tools:** Seaborn.

Best Practices for Visualizing Cleaned Data:

- **Clarity:** Ensure that visuals are easy to understand, avoiding clutter.
- **Consistency:** Use consistent scales, colors, and labels to ensure comparisons are meaningful.
- **Appropriate charts:** Choose the chart that best represents the data, and avoid using charts that distort the story.



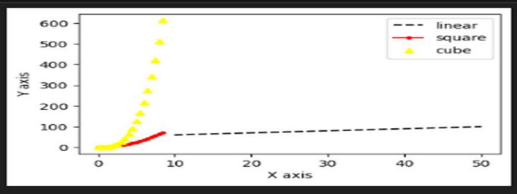
Two or more plots in figure for comparions

```
x=[10,20,30,40,50]
y=[60,70,80,90,100]
plt.figure(figsize=(5,3))
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.plot(x,y,'--',color='black',label='linear');
x2=np.arange(0,9,0.5)


plt.plot(x2,x2**2,'.-',color='red',label='square');

plt.plot(x2,x2**3,'^',color='yellow',label='cube');
plt.legend()

plt.show()
```




```
a=np.array([1,2,3,4,5,6])
b=np.array([8,2,4,6,10,12])

plt.subplot(1,3,1)
plt.title('linear graph')
plt.plot(a,b)

plt.subplot(1,3,2)
plt.title('exponential')
plt.plot(a,b**2)


plt.subplot(1,3,3)

plt.plot(a,b)


plt.show()
```

```
plt.boxplot(new_df_cap['tip']);
```

```
df=sns.load_dataset('tips')
df
```

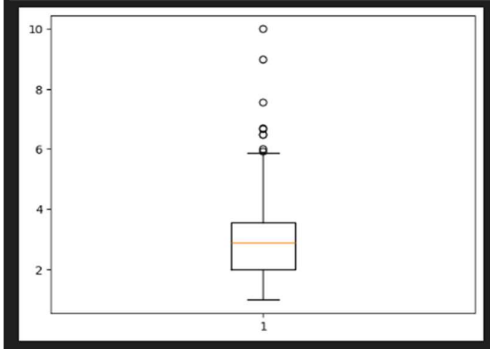
```
df['tip'].mean()
```

```
2.99827868852459
```

```
df['tip'].median()
```

```
2.9
```

```
plt.boxplot(df['tip']);
```




```
# upper_limit=q3+1.5*(IQR)
# lower_limit=q1-1.5*(IQR)
```