05/06/2025

s.loc[0] → value at index label 0

s.iloc[0] → value at index 0

→ s.loc[0:1] → returns 0 and 1

s.iloc[0:1] → returns only 0, exclusive of 1

(*) What if you sort the Series → s.sort_index()

→ They will return same results.

Data Visualization

→ Delivers data with efficiency, clarity and effectiveness

→ Can identify patterns
- correlations
- Trends over time
- Frequency

→ Analyze large datasets and have data driven decision management.

Matplotlib

→ import matplotlib.pyplot as plt
plt.plot(Convert)

→ Histograms: Measures frequency of data

→ Area/Barchart: Represent no. of observations for different categories

→ Piecharts: Represent percentage data by each category.

```
Data = {'Year': [1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010, 2020],
        'Exchange Rate': [65, 69, 71, 64, 62, 59, 72, 71, 75, 78, 81]}

df = pd.DataFrame(Data, columns = ['Year', 'Exchange Rate'])
```

→ df.plot(x='Year', y='Exchang Rate', kind='bar')

① bar
② area
③ Line
④ barh

→ plt.scatter(df['Year'], df['Exchange Rate'])
plt.show()

→ Data = {'Tasks': [100, 500, 300]}

df = pd.DataFrame(Data, columns = ['Tasks'], index = ['Pending', 'Complete', 'Ongoing'])

df.plot.pie(y='Tasks', figsize = (5, 5))

plt.show()

06/06/2025

→ pd.columnname.valueCounts() : Returns the count of values in Each category of the columns specified.
(or)
df['Gender'].valueCounts()

→ plt.bar(Churn_df['Gender'], Churn_df['Age'].max(), color='Cyan', edgecolor='Orange')
plt.title('Gender vs Age', fontdict={'fontsize':20, 'color':'Green'})
plt.show()

→ plt.scatter(Churn_df['Age'], Churn_df['tenure'])
plt.title('Scatter plot')
plt.xlabel('This is X Axis')
plt.ylabel('This is Y Axis')
plt.show()

Scatter plot

→ plt.hist(Churn_df['Tenure'], bins=30)
plt.xlabel('Tenure')
plt.ylabel('Probability')
plt.show()

histogram

→ labels_ex = 'JavaScript', 'Java', 'Python', 'R'
sizes = [15, 30, 45, 10]
explode_labels = (0, 0.5, 0, 0)
fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode_labels, labels=labels_ex, shadow=True, startingangle=90)
ax1.axis('equal')
plt.show()

07/06/2025

→ df['Tenure'].plot.box() → box plot

→ Boxplot has 5 important measures Q_1, Q_3, Minimum, Maximum, Median

Seaborn: it Tries to make easy things easy and hard things possible.

→ import seaborn as sns