data-toolkit

July 24, 2025

THEORITICAL -

#1) What is NumPy, and why is it widely used in Python ? - NumPy stands for 'Numerical Python' and is a fundamental library in Python for scientific computing. NumPy is ideal for handling large amounts of homogeneous data, offering significant improvements in speed and memomry efficiency.

#2) How does broadcasting work in NumPy ? - NumPy broadcasting is a powerful feature that allows for implicit element- wise operations between arrays of different shapes and sizes. Broadcasting automatically adjusts the shape of smaller arrays to make them compatible with larger arrays, eliminating the need for explicit looping or copying of data. The general rule for broadcasting is that the dimensions of the arrays must be compatible, meaning that either the dimensions are equal or one of them is 1.

#3) What is a Pandas DataFrame ? - Pandas DataFrame is a two-dimensional table, similar to a spreadsheet or a SQL table. It consists of rows and columns and each column is a series. DataFrames are the primary data structure for most pandas operations.

#4) Explain the use of the groupby() method in Pandas. - Window grouping involves applying operations within specific groups of data rather than the entire dataset. This is achieved using Pandas groupby() method along with window operations.

#5) Why Seaborn preferred for statistical visualizations ? - Seaborn is preferred for statistical visualizations as it provides a high- level interface that reduces the complexity of creating statistical graphics, making it accessible even to those who may not have deep expertise in programming.

#6) What are the differences between NumPy arrays and Python lists ? - NumPy arrays are homogeneous and uses less memory whereas Python lists are heterogeneous and uses more memory.

#7) What is a heatmap and when should it be used ? - Heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. It is used to see the correlation between columns of a dataset where we can use a darker color for columns havinf a high correlation.

#8) What does the term "Vectorized operation" mean in NumPy ? - Vectorized operations in NumPy refer to performing opeartions on entire arrays without the need for explicit Python loops.

#9) How does Matplotlib differ from Plotly ? - Matplotlib is best for static, publication- quality plots with full customization whereas Plotly is ideal for interactive, web-friendly visualizations and dashboard.

#10) What is the significance of hierarchical indexing in Pandas ? - The significance of hierarchical indexing in Pandas is: a) Efficient handling of multi-dimensional data in 1D/2D structures. b)

Better organization and grouping of complex datasets. c)Flexible selection, slicing and filtering using multiple index levels.

#11) What is the role of Seaborn's pairplot() function ? - The role of Seaborn's pairplot() function to plot pairwise relationships between variables within a data set, making it easier to visualize and understand large data sets.

#12) What is the purpose of the describe() function in Pandas ? - The purpose of the describe() function in Pandas is to generate descriptive statistics that summarize the central tendency, dispersion, and shape of dataset's distribution.

#13) Why is handling missing data important in Pandas ? - Handling missing data in Pandas is important because data cleaning and preprocessing are crucial steps in text data analysis. It also helps in removing duplicates and normalizing text.

#14) What are the benefits of using Plotly for data visualization ? - The benefits of using Plotly for data visualization as it allows users to create interactive and visually appealing data visualizations.

#15) How does NumPy handle multidimensional arrays ? - NumPy handles multidimensional arrays with powerful indexing, shape manipulation and broadcasting capabilities.

#16) What is the role of Bokeh in data visualization ? - The role of Bokeh in data visualization is to create interactive , web-friendly data visualizations. It allows uses to build dynamic plots that support zooming, panning and tooltips and can be easily embedded in web apps or dashboards.

#17) Explain the difference between apply() and map() in Pandas. - The apply() method in Pandas allows to apply a function along the axis of a DataFrame or series wheres as map() method in Pandas is used to substitute each value in a series with another value.

#18) What are the advanced features of NumPy ? - The advanced features of NumPy are broadcasting, vectorization, advanced indexing and slicing.

#19) How does Pandas simplify time series analysis ? - Pandas simplifies time series analysis through its specialized data structures and functions designed to handle time-indexed data efficiently.

#20) What is the role of a pivot table in Pandas? - The role of a pivot table in Pandas is to help in quickly summarization, exploration and analyzation of data by aggregating and organizing it in a easy to read format.

#21) Why is NumPy's array slicing faster than Python's list slicing ? - NumPy's array slicing faster than Python's list slicing because it creates views instead of copies, avoiding extra memory and Python loops and takes advantage of efficient, low-level operations.

#22) What are the common use cases for Seaborn ? - The common use cases for Seaborn are : * Bar charts to compare groups. * Line charts to see trends over time. * Scatter plots to find relationship between numbers.

PRACTICAL-

[ ]:
```
#1) How do you create a 2D NumPy array and calculate the sum of each row.

import numpy as np
```

```
arr= np.array([[1,2,4],[6,8,9],[4,5,3]])
row_sum = np.sum(arr, axis=1)

print(arr)
print(row_sum)
```

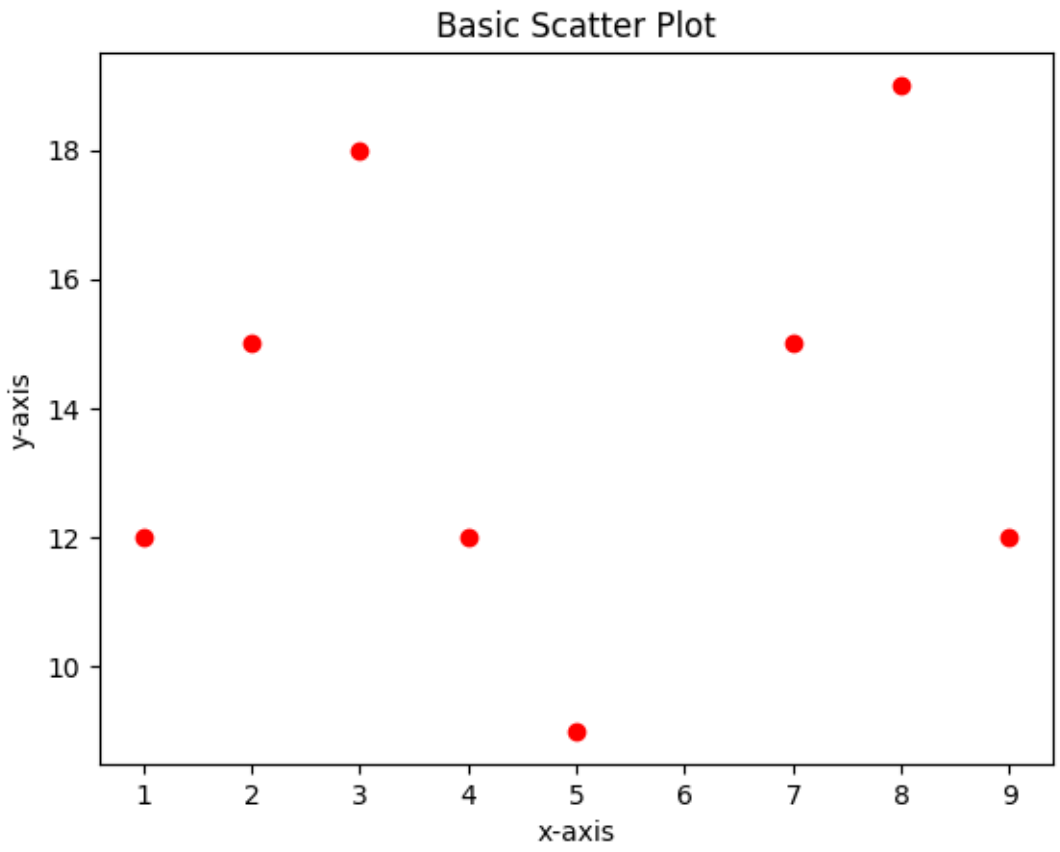
```
[[1 2 4]
 [6 8 9]
 [4 5 3]]
[ 7 23 12]
```

[ ]:
```
#2) Write a Pandas script to the mean of a specific column in a DataFrame.

import pandas as pd

d={"Name":['A','B','C','D','E','F'],"Age":[7,9,11,5,13,2]}
df= pd.DataFrame(d)
print(df)
df["Age"].mean()
```

```
  Name  Age
0    A    7
1    B    9
2    C   11
3    D    5
4    E   13
5    F    2
```

[ ]:
```
np.float64(7.833333333333333)
```

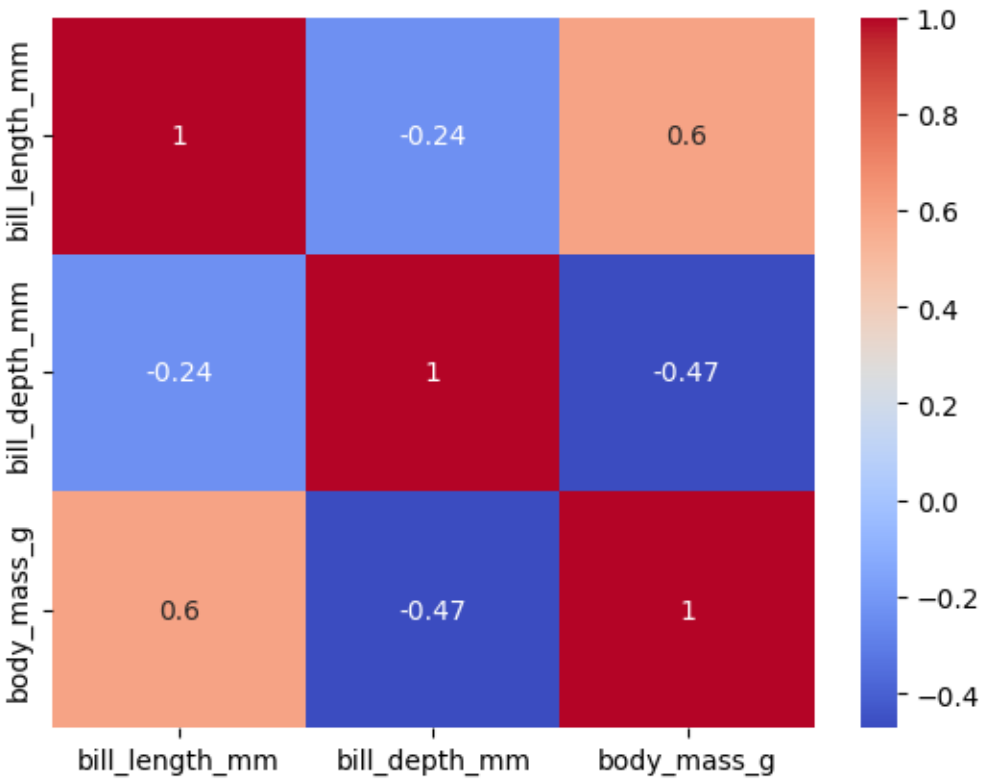
[ ]:
```
#3) Create a scatter plot using Matplotlib.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

x=[1,2,3,4,5,7,8,9]
y=[12,15,18,12,9,15,19,12]

plt.scatter(x, y, color="red",marker="o")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Basic Scatter Plot")
plt.show()
```



[ ]:
```
#4) How do you calculate the correlation matrix using Seaborn and visualze it␣
 ↪with a heatmap?

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")

df=sns.load_dataset("penguins")
df1= df[['bill_length_mm', 'bill_depth_mm', 'body_mass_g']]
df1.corr()
print(df1.corr())
sns.heatmap(df1.corr(),cmap="coolwarm",annot=True)
plt.show()
```

```
                bill_length_mm  bill_depth_mm  body_mass_g
bill_length_mm        1.000000      -0.235053     0.595110
```

```
bill_depth_mm       -0.235053       1.000000    -0.471916
body_mass_g          0.595110      -0.471916     1.000000
```



[ ]:
```
#5) Generate a bar plot using Plotly.

import plotly.graph_objects as go
import plotly.express as px

fig=px.bar(x=['A','B','C','D','E'], y=[4,10,5,6,9])
fig.show()
```

[ ]:
```
#6) Create a DataFrame and add a new column based on an existing column.

import pandas as pd

data= {"Name":['Ravi','Sam','Alina','Zaid'],"Age":[23,17,26,30],"Gender":
 ↪['M','M','F','M']}
df=pd.DataFrame(data)

df["Eligible for vote"] = df["Age"].apply (lambda age: 'not to give vote' if␣
 ↪age<18 else 'to give vote' )
print(df)
```

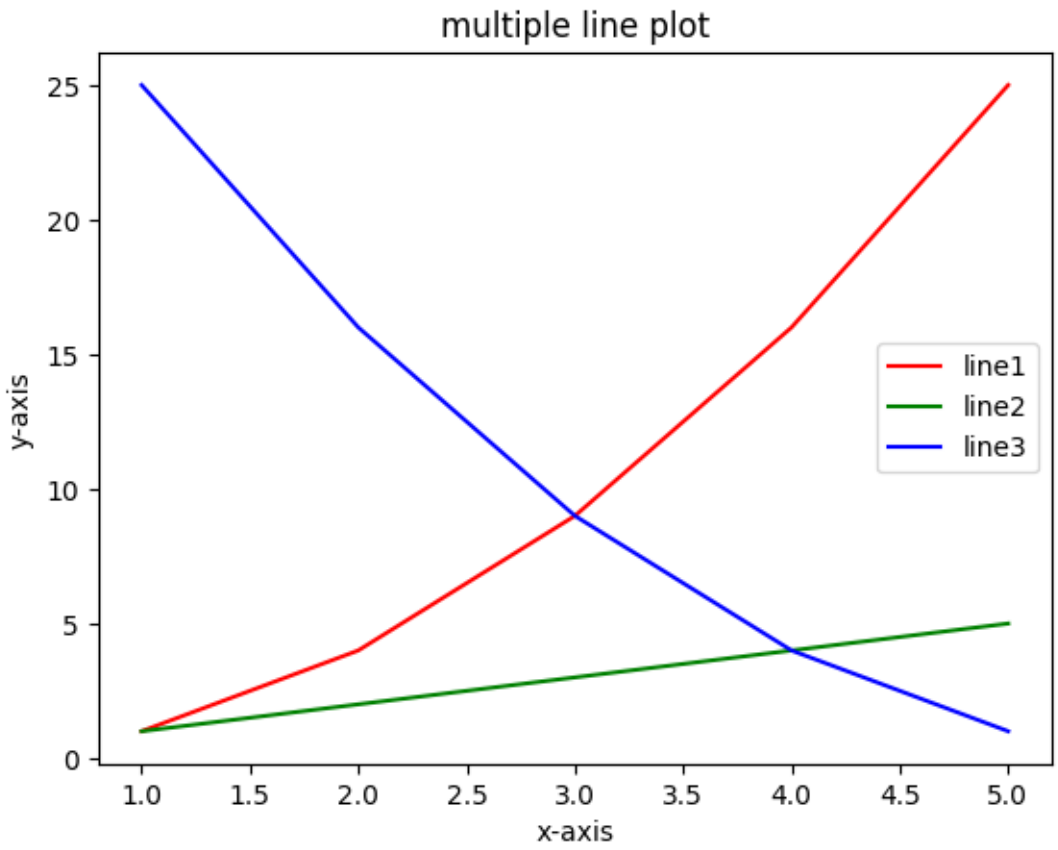
```
    Name  Age Gender Eligible for vote
0   Ravi   23      M      to give vote
1    Sam   17      M  not to give vote
2  Alina   26      F      to give vote
3   Zaid   30      M      to give vote
```

[ ]:
```
#7) Write a program to perform element-wise multiplication of two NumPy arrays.

import numpy as np

arr1= np.array([[6,8,9],[4,5,2]])
print(arr1)
arr2= np.array([[4,5,2],[11,13,3]])
print(arr2)

arr1*arr2
```

```
[[6 8 9]
 [4 5 2]]
[[ 4  5  2]
 [11 13  3]]
```

[ ]:
```
array([[24, 40, 18],
       [44, 65,  6]])
```

[ ]:
```
#8) Create a line plot with multiple lines using Matplotlib.

import pandas as pd
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

x= [1,2,3,4,5]
y1=[1,4,9,16,25]
y2=[1,2,3,4,5]
y3=[25,16,9,4,1]

plt.plot(x,y1, label ="line1", color ="red")
plt.plot(x,y2, label="line2",color="green")
plt.plot(x,y3, label ="line3", color="blue")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("multiple line plot")
plt.legend()
plt.show()
```



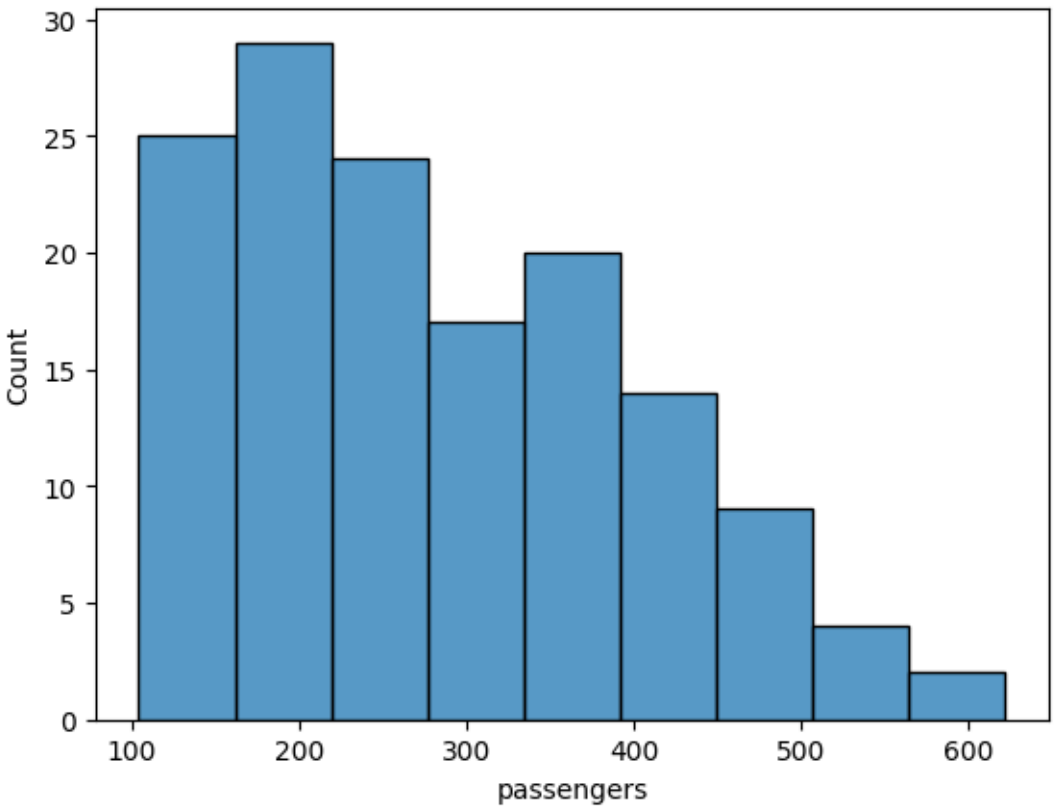
[3]:
```
#9) Generate a Pandas DataFrame and filter rows where a column value is greater than a threshold.

import pandas as pd

data = {"Name":['A','B','C','D','E','F'],"Marks":[98,87,91,71,83,78]}
df= pd.DataFrame(data)
threshold=75
filtered_marks=df[df["Marks"]>threshold]
print(filtered_marks)
```

```
  Name  Marks
0    A     98
1    B     87
2    C     91
4    E     83
5    F     78
```

[ ]:
```
#10) Create a histogram using seaborn to visualize a distribution.
```

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")

df=sns.load_dataset("flights")
print(df)
sns.histplot(df["passengers"])
plt.show()
```

```
     year month  passengers
0    1949   Jan         112
1    1949   Feb         118
2    1949   Mar         132
3    1949   Apr         129
4    1949   May         121
..    …   …           …
139  1960   Aug         606
140  1960   Sep         508
141  1960   Oct         461
142  1960   Nov         390
143  1960   Dec         432

[144 rows x 3 columns]
```



[ ]:
```
#11) Perform matrix multiplication using Numpy.

import numpy as np

arr1= np.array([[4,7,9],[2,1,5],[1,8,6]])
arr2 =([[5,8,9],[4,2,1],[6,5,5]])
np.dot(arr1,arr2)
```

[ ]:
```
array([[102,  91,  88],
       [ 44,  43,  44],
       [ 73,  54,  47]])
```

[ ]:
```
#12) Use Pandas to load a csv file and its first 5 rows.

import pandas as pd

df=pd.read_csv("/content/players.csv")
df.head()
```

[ ]:
```
   Rk         Player Pos Age   Tm   G  GS    MP   FG  FGA  …   FT%  ORB  DRB  \
0   1     Quincy Acy  PF  24  NYK  68  22  1287  152  331  …  .784   79  222
```

```
1  2   Jordan Adams  SG  20  MEM  30   0   248   35   86  …  .609    9   19
2  3   Steven Adams   C  21  OKC  70  67  1771  217  399  …  .502  199  324
3  4    Jeff Adrien  PF  28  MIN  17   0   215   19   44  …  .579   23   54
4  5  Arron Afflalo  SG  29  TOT  78  72  2502  375  884  …  .843   27  220

   TRB  AST STL BLK  TOV   PF   PTS
0  301   68  27  22   60  147   398
1   28   16  16   7   14   24    94
2  523   66  38  86   99  222   537
3   77   15   4   9    9   30    60
4  247  129  41   7  116  167  1035

[5 rows x 30 columns]
```

[ ]:
```
#13) Create a 3D scatter plot using Plotly.

import plotly.graph_objects as go
import plotly.express as px


fig= px.scatter_3d(x=[1,2,3,4,5], y=[3,8,5,2,6], z=[1,2,7,3,5])
fig.show()
```