DATA TOOLKIT - COMPLETE ASSIGNMENT SOLUTIONS

THEORY QUESTIONS

1) NumPy:
NumPy is a powerful Python library used for numerical computing.
It provides support for large multi-dimensional arrays and matrices
along with mathematical functions to operate on them efficiently.

2) Broadcasting:
Broadcasting allows NumPy to perform arithmetic operations on arrays
of different shapes automatically without using loops.

3) Pandas DataFrame:
A DataFrame is a two-dimensional labeled data structure in Pandas
consisting of rows and columns similar to a table in a database.

4) groupby():
The groupby() method splits data into groups, applies a function,
and then combines the results.

5) Why Seaborn:
Seaborn provides attractive statistical visualizations with
better default styles and built-in themes.

6) NumPy vs List:
NumPy arrays are faster, memory-efficient, and store homogeneous data.
Python lists are slower and can store mixed data types.

7) Heatmap:

A heatmap is a color-coded matrix used to visualize relationships between variables, especially correlation matrices.

8) Vectorized Operation:
Vectorization means applying operations to entire arrays at once without writing explicit loops.

9) Matplotlib vs Plotly:
Matplotlib is mainly static. Plotly provides interactive plots.

10) Hierarchical Indexing:
Allows multiple levels of indexing in rows or columns.

11) pairplot():
Creates pairwise plots for relationships between variables.

12) describe():
Returns summary statistics like mean, standard deviation, min, max.

13) Missing Data:
Handling missing data ensures accurate analysis results.

14) Benefits of Plotly:
Provides interactive, zoomable, web-based visualizations.

15) Multidimensional Arrays:
NumPy uses ndarray to handle multi-dimensional arrays efficiently.

16) Bokeh:
Bokeh is used to create interactive web-based visualizations.

17) apply() vs map():
apply() works on DataFrame rows/columns.

map() works only on Series.

18) Advanced NumPy:
Includes broadcasting, vectorization, linear algebra, random module.

19) Time Series in Pandas:
Supports datetime indexing, resampling, rolling operations.

20) Pivot Table:
Used to summarize and aggregate large datasets.

21) Faster Slicing:
NumPy slicing is faster due to contiguous memory allocation.

22) Seaborn Use Cases:
Used for heatmaps, pairplots, distribution plots, regression plots.

PRACTICAL QUESTIONS

1) 2D Array and Row Sum:

```
import numpy as np
arr = np.array([[1,2,3],[4,5,6]])
print(arr.sum(axis=1))
```

2) Mean of Column:

```
import pandas as pd
df = pd.DataFrame({'A':[10,20,30]})
print(df['A'].mean())
```

3) Scatter Plot:

```
import matplotlib.pyplot as plt
plt.scatter([1,2,3],[4,5,6])
plt.show()
```

4) Correlation Heatmap:

```
import seaborn as sns
df = pd.DataFrame({'A':[1,2,3],'B':[4,5,6]})
sns.heatmap(df.corr(), annot=True)
```

5) Bar Plot (Plotly):

```
import plotly.express as px
df = px.data.tips()
fig = px.bar(df, x='day', y='total_bill')
fig.show()
```

6) Add Column:

```
df['B'] = df['A'] * 2
```

7) Element-wise Multiplication:

```
arr1 = np.array([1,2,3])
arr2 = np.array([4,5,6])
print(arr1 * arr2)
```

8) Multiple Line Plot:

```
plt.plot([1,2,3],[4,5,6])
plt.plot([1,2,3],[6,5,4])
plt.show()
```

9) Filter Rows:

```
df[df['A'] > 15]
```

10) Histogram:

```
sns.histplot(df['A'])
```

11) Matrix Multiplication:

```
np.dot(arr1, arr2)
```

12) Load CSV:

```
df = pd.read_csv('file.csv')
print(df.head())
```

13) 3D Scatter Plot:

```
fig = px.scatter_3d(df, x='A', y='B', z='C')
fig.show()
```