ASSIGNMENT 1

1. **What is NumPy, and why is it so popular in Python?**
   NumPy is a powerful Python library for numerical computing. It provides support for multi-dimensional arrays, along with efficient mathematical operations, making it faster and more memory-efficient than traditional Python lists. It's widely used in data science, machine learning, and scientific computing.
2. **How does broadcasting work in NumPy?**
   Broadcasting is a technique that allows NumPy to perform element-wise operations on arrays of different shapes. Instead of manually reshaping arrays, NumPy automatically expands smaller arrays to match the dimensions of the larger one, reducing memory usage and improving performance.
3. **What is a Pandas DataFrame?**
   A Pandas DataFrame is a two-dimensional, table-like data structure, similar to an Excel spreadsheet. It allows you to store and manipulate data using labeled rows and columns, making data analysis and manipulation much easier.
4. **How does the groupby() method work in Pandas?**
   The groupby() method is used to group data based on specific column values. It allows you to apply functions like sum, mean, or count to each group separately, making it useful for summarizing large datasets.
5. **Why is Seaborn commonly used for statistical visualizations?**
   Seaborn simplifies the process of creating visually appealing and informative statistical charts. It is built on top of Matplotlib and provides easy-to-use functions for drawing histograms, scatter plots, heatmaps, and more, making data visualization much more intuitive.
6. **How are NumPy arrays different from Python lists?**
   NumPy arrays are more efficient than Python lists because they store data in a fixed type, allowing for faster computations and reduced memory consumption. Lists, on the other hand, store elements individually and can hold multiple data types, making them more flexible but slower.
7. **What is a heatmap, and when should you use it?**
   A heatmap is a graphical representation of data using colors to indicate values. It is often used to show correlations in a dataset, detect patterns, and visualize large amounts of numerical data in an intuitive way.
8. **What do "vectorized operations" mean in NumPy?**
   Vectorized operations refer to performing mathematical computations on entire arrays instead of looping through individual elements. This approach significantly improves performance by utilizing optimized low-level operations.
9. **How does Matplotlib differ from Plotly?**
   Matplotlib is a static plotting library, primarily used for simple and traditional charts. Plotly, on the other hand, is an interactive visualization library that allows zooming, panning, and hovering over data points. It’s great for web-based dashboards and exploratory data analysis.

10. **What is hierarchical indexing in Pandas, and why is it important?**
    Hierarchical indexing allows you to have multiple levels of indexing in a DataFrame, enabling better organization of complex datasets. It is especially useful for working with multi-dimensional data in a structured way.
11. **What does Seaborn's pairplot() function do?**
    The pairplot() function creates scatter plots for all numerical column combinations in a dataset. It helps in visualizing relationships and distributions between multiple variables at once.
12. **What is the purpose of the describe() function in Pandas?**
    The describe() function provides summary statistics like mean, median, min, max, and percentiles for numerical columns in a DataFrame, helping you quickly understand the distribution of your data.
13. **Why is handling missing data important in Pandas?**
    Missing data can lead to inaccurate analysis and predictions. Pandas provides tools to fill, drop, or interpolate missing values to ensure the dataset remains reliable and useful.
14. **What are the benefits of using Plotly for data visualization?**
    Plotly allows interactive and web-based visualizations, making data exploration more engaging. It supports a variety of chart types and integrates well with frameworks like Dash for building interactive dashboards.
15. **How does NumPy handle multi-dimensional arrays?**
    NumPy provides the ndarray object, which can store multi-dimensional data efficiently. It allows operations on entire arrays at once and offers reshaping, slicing, and indexing capabilities for easy manipulation.
16. **What is Bokeh's role in data visualization?**
    Bokeh is a Python library for creating interactive visualizations, particularly for web applications. It allows zooming, panning, and real-time data updates, making it ideal for dynamic dashboards.
17. **How do apply() and map() differ in Pandas?**

18. apply() is used to apply a function to each row or column in a DataFrame.

19. map() is used for element-wise transformations in a single Pandas Series.

20. **What are some advanced features of NumPy?**
    NumPy supports broadcasting, advanced indexing, fast Fourier transforms (FFT), linear algebra operations, and integration with C/C++ for performance improvements.
21. **How does Pandas simplify time series analysis?**
    Pandas provides specialized functions for handling date and time data, such as resampling, shifting, and rolling window calculations, making time series analysis more efficient.
22. **What is a pivot table in Pandas, and why is it useful?**
    A pivot table is a data summarization tool that allows you to reorganize and aggregate data in a flexible way. It is useful for grouping, sorting, and computing statistics dynamically.
23. **Why is NumPy's array slicing faster than Python's list slicing?**
    NumPy arrays store data in contiguous memory locations, allowing for direct memory access, whereas Python lists require multiple references, making them slower for large-scale operations.
24. **What are some common use cases for Seaborn?**
    Seaborn is commonly used for correlation analysis, distribution visualization, categorical plots, time series visualization, and regression plots.

CODE WITH OUTPUT

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, 3], [4, 5, 6]])
row_sums = np.sum(arr, axis=1)
print(row_sums)
```

```
PS D:\assignment> python -u "d:\assignment\NumPy.py"
[ 6 15]
```

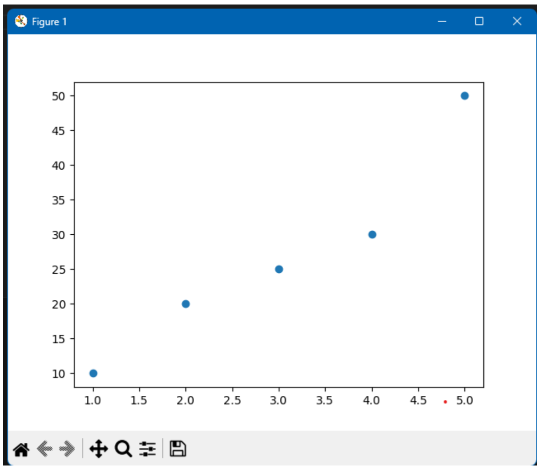
2. Write a Pandas script to find the mean of a specific column in a DataFrame.

```
import pandas as pd
df = pd.DataFrame({'A': [10, 20, 30, 40]})
mean_value = df['A'].mean()
print(mean_value)
```

```
PS D:\assignment> python -u "d:\assignment\NumPy.py"
25.0
```

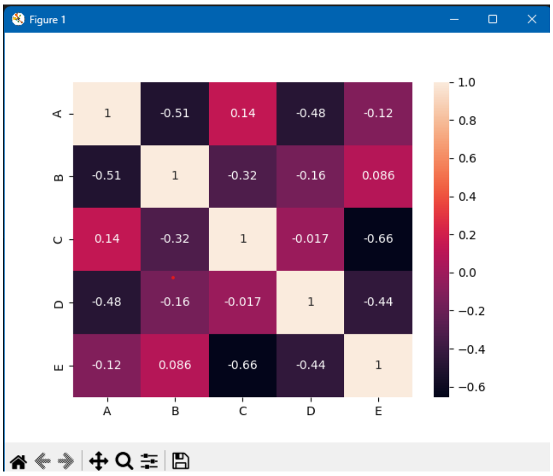
3. Create a scatter plot using Matplotlib.

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 50]
plt.scatter(x, y)
plt.show()
```



4. How do you calculate the correlation matrix using Seaborn and visualize it with a heatmap?

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.DataFrame(np.random.rand(5,5), columns=list('ABCDE'))
sns.heatmap(data.corr(), annot=True)
plt.show()
```



5. Generate a bar plot using Plotly.

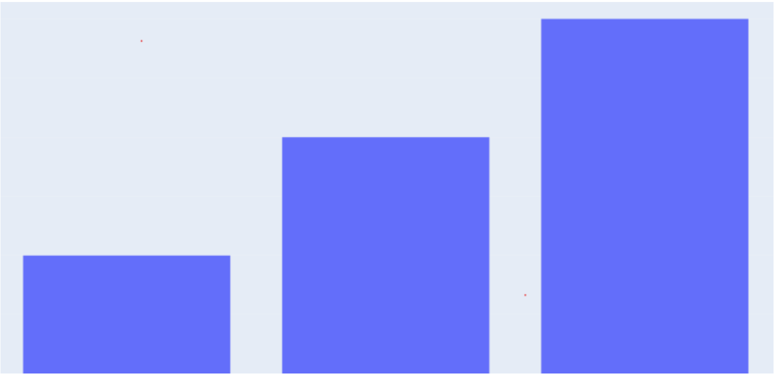
```
import plotly.express as px

import pandas as pd

data = pd.DataFrame({'Category': ['A', 'B', 'C'], 'Values': [10, 20, 30]})

fig = px.bar(data, x='Category', y='Values')

fig.show()
```



6. Create a DataFrame and add a new column based on an existing column.

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

7. Perform element-wise multiplication of two NumPy arrays.

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

8. Create a histogram using Seaborn to visualize a distribution.

```
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = np.random.randn(1000)
sns.histplot(data, kde=True)
plt.show()
```

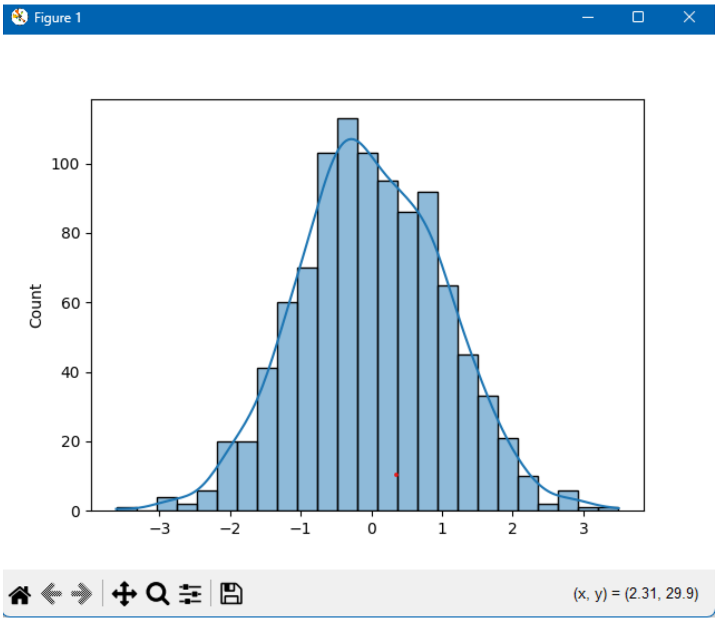


Figure 1
Count
0
20
40
60
80
100
−3
−2
−1
0
1
2
3
(x, y) = (2.31, 29.9)