Name: Dheephiga A M

Worldline Temp ID: WDGET2024067

Assessment - 2

Numpy

In [20]:

```
import numpy as np
import pandas as pd
```

In [21]:

```
array1 = np.random.randint(1,101, size=(20))
```

In [22]:

```
#reshaping the array
matrix1 = array1.reshape(4,5)
matrix1
```

Out[22]:

```
array([[ 25,  31,  24,  28,  79],
       [ 92,  86,  62,  77,  88],
       [ 57,   8,   4,  29, 100],
       [ 19,  88,  56,  74,  12]])
```

In [23]:

```
#mean, median, standard deviation of the matrix
print(f'mean of matrix: {matrix1.mean()}')
print(f'median of matrix: {np.median(matrix1)}')
print(f'mean of matrix: {matrix1.std()}')
```

```
mean of matrix: 51.95
median of matrix: 56.5
mean of matrix: 31.320081417518697
```

In [24]:

```
#3x3 identity matrix
identityMatrix = np.eye(3)
identityMatrix
```

Out[24]:

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

In [25]:

```
#evenly spaced numbers
array3 = np.linspace(0,5,10)
array3
```

Out[25]:

```
array([0.        , 0.55555556, 1.11111111, 1.66666667, 2.22222222,
       2.77777778, 3.33333333, 3.88888889, 4.44444444, 5.        ])
```

In [26]:

```
#replacing values in an array
array4 = np.arange(0,10)
array4 = np.where(array4 % 2 == 0, -1, array4)
array4
```

Out[26]:

```
array([-1,  1, -1,  3, -1,  5, -1,  7, -1,  9])
```

In [27]:

```
#dot product of 2 matrices
mat1 = np.array([[1,2],
                [3,4],
                [5,6]])
# print(np.shape(mat1))
mat2 = np.array([[5,4,2],
                 [3,1,1]])
# print(np.shape(mat2))
print(f'Dot product of the matrices:\n{np.dot(mat1,mat2)}')
```

```
Dot product of the matrices:
[[11  6  4]
 [27 16 10]
 [43 26 16]]
```

In [28]:

```
#finding the indices of the min and max values in the array
max_value = np.max(array1)
max_index = np.argmax(array1)

min_value = np.min(array1)
min_index = np.argmin(array1)

print(f'Maximum value is {max_value}\nIndex is {max_index}')
print(f'Minimum value is {min_value}\nIndex is {min_index}')
```

```
Maximum value is 100
Index is 14
Minimum value is 4
Index is 12
```

In [29]:

```
#using filter in an array
array5 = np.arange(1,51)
array6 = array5[array5>30]
array6
```

Out[29]:

```
array([31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
       48, 49, 50])
```

In [30]:

```
#broadcasting
array7 = np.array([[3,5,2],
                   [1,2,4],
                   [3,6,1]])
operation = array7 + 5
print(f'Before broadcasting\n{array7}\n\nAfter broadcasting\n{operation}')
```

```
Before broadcasting
[[3 5 2]
 [1 2 4]
 [3 6 1]]

After broadcasting
[[ 8 10  7]
 [ 6  7  9]
 [ 8 11  6]]
 [ 8 11  6]]
```

Pandas

In [31]:

```
data = {'Name': ['A', 'B', 'C','D','E','F','G','H','I','J'],
        'Age': [23,42,33,52,25, 30, 35,29,25,38],
        'City': ['Santa Clara','Santa Clara', 'Boston', 'Los Angeles', 'Boston','Chicago
', 'Boston', 'Ohio','Brooklyn','Brooklyn']}
df = pd.DataFrame(data)
df.to_csv('data.csv',index=False)
```

In [32]:

```
df_csv = pd.read_csv('data.csv',index_col=None)

#adding a new row
newRow = {'Name': 'K', 'Age': 32, 'City': 'Santa Clara'}
df_csv = df_csv._append(newRow, ignore_index=True)
df_csv
```

Out[32]:

	Name	Age	City
0	A	23	Santa Clara
1	B	42	Santa Clara
2	C	33	Boston
3	D	52	Los Angeles
4	E	25	Boston
5	F	30	Chicago
6	G	35	Boston
7	H	29	Ohio
8	I	25	Brooklyn
9	J	38	Brooklyn
10	K	32	Santa Clara

In [33]:

```
#displaying first 5 rows
df_csv.head()
len(df_csv)
```

Out[33]:

```
11
```

In [34]:

```
df_csv['Salary'] = [70000,50000, 60000, 70000, 80000,50000, 60000, 70000,70000,50000, 60
000]
df_csv
```

Out[34]:

	Name	Age	City	Salary
0	A	23	Santa Clara	70000
1	B	42	Santa Clara	50000
2	C	33	Boston	60000
3	D	52	Los Angeles	70000

	Name	Age	City	Salary
4	E	25	Boston	80000
5	F	30	Chicago	50000
6	G	35	Boston	60000
7	H	29	Ohio	70000
8	I	25	Brooklyn	70000
9	J	38	Brooklyn	50000
10	K	32	Santa Clara	60000

In [35]:

```
#selecting only name and salary columns
df_csv[['Name','Salary']]
```

Out[35]:

	Name	Salary
0	A	70000
1	B	50000
2	C	60000
3	D	70000
4	E	80000
5	F	50000
6	G	60000
7	H	70000
8	I	70000
9	J	50000
10	K	60000

In [36]:

```
#sorting by salary
df_csv.sort_values(by='Salary',ascending=False)
```

Out[36]:

	Name	Age	City	Salary
4	E	25	Boston	80000
0	A	23	Santa Clara	70000
3	D	52	Los Angeles	70000
7	H	29	Ohio	70000
8	I	25	Brooklyn	70000
2	C	33	Boston	60000
6	G	35	Boston	60000
10	K	32	Santa Clara	60000
1	B	42	Santa Clara	50000
5	F	30	Chicago	50000
9	J	38	Brooklyn	50000

In [37]:

```
#mean and standard deviation of age column
print(f'Mean age is {df_csv['Age'].mean()}\nStandard Deviation of Age is {df_csv['Age'].s
td()}')
```

```
Mean age is 33.09090909090909
Standard Deviation of Age is 8.514159329664267
```

In [43]:

```
#Group the DataFrame by the 'City' column and calculate the mean salary for each city
df_csv.groupby('City')['Salary'].mean()
```

Out[43]:

```
City
Boston          66666.666667
Brooklyn        60000.000000
Chicago         50000.000000
Los Angeles     70000.000000
Ohio            70000.000000
Santa Clara     60000.000000
Name: Salary, dtype: float64
```

In [44]:

```
df_csv['Years'] = [2,5,7,8,9,3,10,6,5,1,2]
```

In [47]:

```
df_csv.rename(columns={'Name': 'Full Name','Years':'Years of Experience','City':'Locatio
n'})
```

Out[47]:

	Full Name	Age	Location	Salary	Years of Experience
0	A	23	Santa Clara	70000	2
1	B	42	Santa Clara	50000	5
2	C	33	Boston	60000	7
3	D	52	Los Angeles	70000	8
4	E	25	Boston	80000	9
5	F	30	Chicago	50000	3
6	G	35	Boston	60000	10
7	H	29	Ohio	70000	6
8	I	25	Brooklyn	70000	5
9	J	38	Brooklyn	50000	1
10	K	32	Santa Clara	60000	2

In [48]:

```
df_csv.to_csv('modified_data.csv')
```