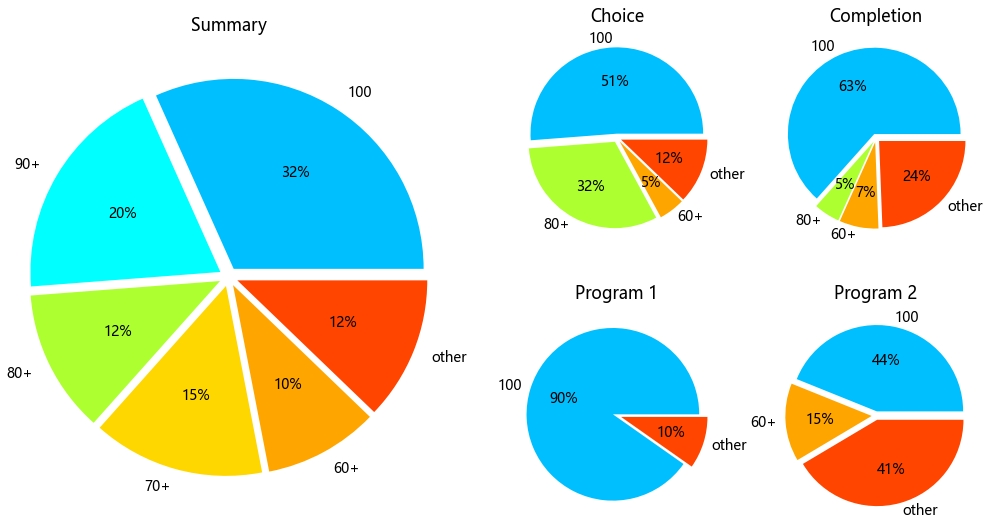
****

**Single-Choice Questions**

1. How do you extract a subarray from a 2D NumPy array arr consisting of rows 1 to 3 (inclusive) and columns 2 to 4 (inclusive)?

|  |
| --- |
| import pandas as pd  arr = np.array([  [10, 20, 30, 40, 50],  [60, 70, 80, 90, 100],  [110, 120, 130, 140, 150],  [160, 170, 180, 190, 200]  ]) *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |

A) arr[1:3, 2:4]

B) arr[1:4, 2:5]

C) arr[1:4, 2:4]

D) arr[1:3, 2:5]

2． In pandas, how do you increment the values in the column 'Age' by 1?

|  |
| --- |
| import pandas as pd  df = pd.DataFrame({  'Name': ['Alice', 'Bob', 'Charlie', 'David'],  'Age': [25, 30, 35, 40] }) *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |

A) df[Age] += 1  
B) df['Age'] = df['Age'] + 1  
C) df['Age'] = df['Age'].apply(lambda += 1)  
D) df['Age'] = df[Age] + 1

3. How do you find the maximum value in a NumPy array?

A) np.max()

B) np.argmax()

C) np.argmaxmax()

D) np.max\_index()

**Multiple-Choice Questions**

1. How to create a 3×3 numpy array of all True’s?

A) np.full((3, 3), True, dtype=bool)

B) np.ones((3,3), dtype=bool)

C) np.full((3, 3), dtype=bool)

D) np.ones((3,3))

Answer: AB

2. How can you save a NumPy array **arr** to a CSV file named output.csv?

A) np.savetxt('output.csv', arr)

B) arr.to\_csv('output.csv')

C) np.savetxt('output.csv', arr, delimiter=',')

D) pd.to\_csv(arr, 'output.csv')

Answer: C / AC

**Completion**

|  |  |
| --- | --- |
| Code | import numpy as np  a = np.arange(16) *# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* print(a.reshape(4, -1)) |
| Output | [[ 0 2 2 4]  [ 4 6 6 8]  [ 8 10 10 12]  [12 14 14 16]] |

Answer: a[1::2] += 1

**Programming**

1. Create the matrix as shown below. You can concatenate multiple matrices to obtain the result using np.concatenate, or create a large matrix and set values using slicing.

[[1 1 1 3 3]

[2 2 2 3 3]

[2 2 2 3 3]

[2 2 2 3 3]

[2 2 2 3 3]]

|  |
| --- |
| import numpy as np  a = np.full((5, 5), 2) a[0] = 1 a[:, 3:] = 3 print(a) |
| import numpy as np  a = np.full((1, 3), 1) b = np.full((4, 3), 2) c = np.full((5, 2), 3) print(np.concatenate((np.concatenate((a, b), axis=0), c), axis=1)) |

1. Merge the information of two sets of Titanic passengers and calculate the survival rate (0.6296) of first-class passengers (Pclass=1). You can filter out first-class passengers using a boolean vector, or sort and then slice to obtain the same result.

|  |
| --- |
| import pandas as pd  df = pd.merge(  pd.read\_csv("Titanic1.csv"),  pd.read\_csv("Titanic2.csv"),  on="Name" ) mask = df["Pclass"] == 1 print(df["Survived"][mask].mean()) |
| import pandas as pd  df = pd.merge(  pd.read\_csv("Titanic1.csv"),  pd.read\_csv("Titanic2.csv"),  on="Name" ) df = df.sort\_values("Pclass", ignore\_index=True) for i in range(len(df)):  if df.loc[i, "Pclass"] == 2:  print(df.loc[:i - 1, "Survived"].mean())  break |