

lab7

January 27, 2025

```
[2]: import numpy as np
import pandas as pd
print(np.__version__)
print(pd.__version__)
```

2.2.0

2.2.3

```
[3]: s = pd.Series([1, 2, 3, 4], index=['a', 'b', 'c', 'd'])
print(s)
```

a 1

b 2

c 3

d 4

dtype: int64

```
[4]: arr = np.array([[1, 2], [3, 4]])
print("Shape:", arr.shape)
print("Dimensions:", arr.ndim)
print("Data Type:", arr.dtype)
```

Shape: (2, 2)

Dimensions: 2

Data Type: int64

```
[5]: arr1 = np.array([1, 2, 3, -1])
arr2 = np.array([4, 5, 6, 9])
print(arr1 + arr2)
print(arr1 * arr2)
```

[5 7 9 8]

[4 10 18 -9]

```
[6]: arr = np.array([1, 2, 3, 4])
print(arr)
```

[1 2 3 4]

```
[7]: zeros = np.zeros((2, 3), dtype=int)
      print(zeros)
      print(zeros.dtype)
```

```
[[0 0 0]
 [0 0 0]]
int64
```

```
[8]: ones = np.ones((3, 2),dtype= int)
      print(ones)
```

```
[[1 1]
 [1 1]
 [1 1]]
```

```
[9]: linspace = np.linspace(0, 10, 2)
      print(linspace)
```

```
[ 0. 10.]
```

```
[10]: arrange = np.arange(0, 12, 3)
       print(arrange)
```

```
[0 3 6 9]
```

```
[11]: arr = np.array([[1, 2], [3, 4], [5, 6]],
                      [[2, 3], [3, 2], [5, 1]],
                      [[9, 1],[5, 9],[1, 0]])
      print("shape",arr.shape)
      print("Data Type:",arr.dtype)
      print("Size:",arr.size)
      print("Dimensions:",arr.ndim)
```

```
shape (3, 3, 2)
Data Type: int64
Size: 18
Dimensions: 3
```

```
[12]: arr = np.array([19,23,43,40])
      print(arr[2])
```

```
43
```

```
[13]: arr = np.array([1, 2, 3, 4, 5, 6, 12])
      print(arr[0:7])
```

```
[ 1  2  3  4  5  6 12]
```

```
[14]: arr = np.array([10, 20, 30, 40, 50, 12])  
      print(arr[[0, 2, 5]])
```

```
[10 30 12]
```

```
[15]: arr = np.array([1, 2, 3, 4, 5])  
      print(arr[arr > 3])
```

```
[4 5]
```

```
[16]: arr = np.arange(1, 7)  
      reshaped = arr.reshape(2, 3)  
      print(reshaped)#only reshape not manipulate
```

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

```
[17]: flattened = reshaped.ravel()  
      print(flattened)
```

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

```
[18]: transposed = reshaped.T  
      print(transposed)
```

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

```
[19]: arr1 = np.array([1, 2])  
      arr2 = np.array([3, 4])  
      print(np.vstack((arr1, arr2)))
```

```
[[1 2]  
 [3 4]]
```

```
[20]: transposed = reshaped.T  
      print(transposed)
```

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

```
[21]: arr1 = np.array([1, 2])  
      arr2 = np.array([3, 4])  
      print(np.vstack((arr1, arr2)))
```

```
[[1 2]  
 [3 4]]
```

```
[22]: print(np.hstack((arr1, arr2)))
```

```
[1 2 3 4]
```

```
[23]: arr = np.array([1, 2, 3, 4, 5, 6])
      print(np.split(arr, 3))
```

```
[array([1, 2]), array([3, 4]), array([5, 6])]
```

```
[24]: arr = np.array([1, 2, 3])
      print(arr + 2)
```

```
[3 4 5]
```

```
[25]: arr = np.array([1, 2, 3, 4])
      print("Sum:", arr.sum())
      print("Mean:", arr.mean())
      print("Std Dev:", arr.std())
```

```
Sum: 10
```

```
Mean: 2.5
```

```
Std Dev: 1.118033988749895
```

```
[26]: a = np.array([[1, 2], [3, 4]])
      b = np.array([[5, 6], [7, 8]])
      print(np.dot(a, b))
```

```
[[19 22]
 [43 50]]
```

```
[27]: from numpy.linalg import det, eig
      print("Determinant:", det(a))
      print("Eigenvalues:", eig(a))
```

```
Determinant: -2.0000000000000004
```

```
Eigenvalues: EigResult(eigenvalues=array([-0.37228132,  5.37228132]),
eigenvectors=array([[ -0.82456484, -0.41597356],
 [ 0.56576746, -0.90937671]]))
```

```
[28]: a = np.array([[1, 2], [3, 4]])
      b = np.array([1, 0])
      print(a + b)
```

```
[[2 2]
 [4 4]]
```

```
[29]: arr = np.arange(1_000_000)
      %timeit arr + 2
```

912 s \pm 87.7 s per loop (mean \pm std. dev. of 7 runs, 1,000 loops each)

```
[30]: import pandas as pd
s = pd.Series([10, 20, 30], index=["a", "b", "c"])
print(s)
```

```
a    10
b    20
c    30
dtype: int64
```

```
[31]: data = {"Name": ["Alice", "Bob"], "Age": [25, 30]}
df = pd.DataFrame(data)
print(df)
```

```
      Name  Age
0  Alice   25
1   Bob   30
```

```
[32]: data = [["Alice", 25], ["Bob", 30]]
df = pd.DataFrame(data, columns=["Name", "Age"])
print(df)
```

```
      Name  Age
0  Alice   25
1   Bob   30
```

```
[33]: arr = np.array([[1, 2], [3, 4]])
df = pd.DataFrame(arr, columns=["A", "B"])
print(df)
```

```
   A  B
0  1  2
1  3  4
```

```
[34]: df = pd.read_csv("phone_usage.csv")
df = pd.read_excel("Obesity_Dataset.xlsx")
```

```
[ ]: print(df.head(2))
print(df.tail(2))
```

```
[ ]: print(df.info())
print(df.describe())
```

```
[ ]: print(df.loc[0])
print(df.loc[:, "Sex"])
```

```
[38]: filtered = df[df["Height"]>190]
      filtered
```

```
[38]:      Sex  Age  Height  Overweight_Obese_Family  Consumption_of_Fast_Food  \
424     1   25    191                        2                1
680     1   39    191                        1                1
1485    1   39    193                        1                1

      Frequency_of_Consuming_Vegetables  Number_of_Main_Meals_Daily  \
424                                2                2
680                                1                3
1485                               1                3

      Food_Intake_Between_Meals  Smoking  Liquid_Intake_Daily  \
424                        1         1                1
680                        4         1                1
1485                       1         1                1

      Calculation_of_Calorie_Intake  Physical_Excercise  \
424                        1                2
680                        2                3
1485                       2                3

      Schedule_Dedicated_to_Technology  Type_of_Transportation_Used  Class
424                                3                4        3
680                                1                1        4
1485                               1                1        4
```

```
[39]: data = {
      ("A", "X"): [1, 2],
      ("A", "Y"): [3, 4],
      ("B", "X"): [5, 6]
    }
df = pd.DataFrame(data)
print(df)
```

```
   A    B
   X  Y  X
0  1  3  5
1  2  4  6
```

```
[ ]: df = pd.DataFrame({"Group": ["A", "A", "B"], "Value": [1, 2, 3]})
agg_result = df.groupby("Group").agg({"Value": ["mean", "sum"]})
print(agg_result)
```

```
[ ]: df = pd.DataFrame({"A": [1, 2, 3], "B": [4, 5, 6]})
df["C"] = df.eval("A + B")
```

```
print(df)
```

```
[1]: import numpy as np  
      np.random.seed(42)
```

```
[2]: pip freeze > requirements.txt
```

Note: you may need to restart the kernel to use updated packages.

```
[ ]: df.to_csv("processed_data.csv", index=False)
```

```
[ ]: df.to_parquet("data.parquet")
```

```
[ ]: df = pd.read_csv("large_data.csv", usecols=["col1", "col2"], nrows=1_000_000)
```

```
[ ]: df["new_col"] = df["col"] * 2
```