In this file, we will introduce:

- NumPy Arrays
- Pandas Series
- Pandas DataFrame

Python Numpy

↳ *35 cells hidden*

Pandas Data Structures

↳ *62 cells hidden*

Excercies

NumPy Arrays

Question 1 Write a Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.

```
import numpy as np
l=[1,2,3]
a=np.array(l)
a
```

```
array([1, 2, 3])
```

Question 2

Write a NumPy program to create a 5x5 array with random values, then

- Find the minimum and maximum values.
- Find the indices of the minimum and maximum values.
- Extract the first two rows of the array and store them into a variable.

Hint: you can use the functions: min(), max(), argmax(), argmin().

```
import numpy as np
x = np.random.random((5,5))
print("Original Array:")
print(x)
xmin, xmax = x.min(), x.max()
print("Minimum and Maximum Values:")
print(xmin, xmax)
print("Maximum Values: ",np.argmax(x))
print("Minimum Values: ",np.argmin(x))

#extract the first two rows of the array and store them into a variable.

y= x[:2, :]
print("First 2 rows of the above array:")
print(y)
```

```
Original Array:
[[0.17548685 0.21916869 0.71419026 0.53954744 0.45258217]
 [0.80398697 0.51387988 0.52358944 0.78163764 0.69892034]
 [0.1804729  0.37871651 0.20979048 0.40967697 0.66594013]
 [0.13740716 0.97142537 0.97029685 0.23958181 0.17213444]
 [0.3369958  0.54290908 0.89531325 0.6325588  0.94053328]]
Minimum and Maximum Values:
0.13740715949346505 0.9714253678123024
Maximum Values:  16
Minimum Values:  15
```

Question 3

Write a NumPy program to find common values between two arrays (hint: find the intersection)

```
import numpy as np
array1 = np.array([0, 10, 20, 40, 60])
print("Array1: ",array1)
array2 = [10, 30, 40]
print("Array2: ",array2)
print("Common values between two arrays:")
print(np.intersect1d(array1, array2))
```

```
Array1:  [ 0 10 20 40 60]
Array2:  [10, 30, 40]
Common values between two arrays:
[10 40]
```

Question 4

Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10. Then append two new rows [40, 50, 60], [70, 80, 90] to the array.

```
import numpy as np
x =  np.arange(2, 11).reshape(3,3)
print(x)

#append two new rows [40, 50, 60], [70, 80, 90]to the array.

x = np.append(x, [[40, 50, 60], [70, 80, 90]], axis=0)
print("After append values to the end of the array:")
print(x)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
After append values to the end of the array:
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]
 [40 50 60]
```

Question 5

a NumPy program to create a 2d array with 1 on the border and 0 inside.

```
import numpy as np
x = np.ones((5,5))
print("Original array:")
print(x)
print("1 on the border and 0 inside in the array")
x[1:-1,1:-1] = 0
print(x)
```

```
Original array:
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
1 on the border and 0 inside in the array
[[1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 1.]
 [1. 0. 0. 0. 1.]
```

convert array to series then return back to **list**

```
import numpy as np
import pandas as pd
np_array = np.array([10, 20, 30, 40, 50])
print("NumPy array:")
print(np_array)
new_series = pd.Series(np_array)
print("Converted Pandas series:")
print(new_series)

print("Series to an array")
a = np.array(new_series.values.tolist())
print (a)
```

To find if elements of a2 are exiting in a1

```
import numpy as np
array1 = np.array([0, 10, 20, 40, 60])
print("Array1: ",array1)
array2 = [0, 40]
print("Array2: ",array2)
print("Compare each element of array1 and array2")
print(np.in1d(array1, array2))
```

```
Array1:  [ 0 10 20 40 60]
Array2:  [0, 40]
Compare each element of array1 and array2
[ True False False  True False]
```

Pandas Data Structure

Q1: Write a Pandas program to add, subtract, multiple and divide two Pandas Series.

Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9]

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 9])
ds = ds1 + ds2
print("Add two Series:")
print(ds)
print("Subtract two Series:")
ds = ds1 - ds2
print(ds)
print("Multiply two Series:")
ds = ds1 * ds2
print(ds)
print("Divide Series1 by Series2:")
ds = ds1 / ds2
print(ds)
```

Q2: Write a Pandas program to compare the elements of the two Pandas Series. Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9]

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])
print("Series1:")
print(ds1)
print("Series2:")
print(ds2)
print("Compare the elements of the said Series:")
print("Equals:")
l = (ds1 == ds2)
print(l)
print("Greater than:")
print(ds1 > ds2)
print("Less than:")
print(ds1 < ds2)
```

Q3: Write a Python program to convert a NumPy array to a Pandas series, then convert a the resulted Series into an array again.

Sample array np.array([10, 20, 30, 40, 50])

```
import numpy as np
import pandas as pd
np_array = np.array([10, 20, 30, 40, 50])
print("NumPy array:")
print(np_array)
new_series = pd.Series(np_array)
print("Converted Pandas series:")
print(new_series)

print("Series to a list")
l = np.array(new_series.values.tolist())
print (l)

print("Series to an array")
a = new_series.array
print (a)
```

Q4: Write a Pandas program to add some data to an existing Series.

Sample Series: [2, 4, 6, 8, 10]

```
import pandas as pd
s = pd.Series(['100', '200', 'python', '300.12', '400'])
print("Original Data Series:")
print(s)
print("\nData Series after adding some data:")
new_s = s.append(pd.Series(['500', 'php']))
print(new_s)
```

Q5: Write a Pandas program to create a subset of a given series based on value and condition.

Sample Series: [2, 4, 6, 8, 10]

Sample condition: the value > 5.

```
import pandas as pd
s = pd.Series([0, 1,2,3,4,5,6,7,8,9,10])
print("Original Data Series:")
print(s)
print("\nSubset of the above Data Series:")
n = 5
new_s = s[s > n]
print(new_s)
```

Q6: Write a Pandas program to create a DataFrame using the data given in the following:

a) calculate the sum of the examination attempts by the students.

b) change the score in row 'd' to 11.5.

c) return the row having the maximum score

d) append a new row 'k' to DataFrame with given values for each column.

e) delete the new row and return the original data frame.

f) change the name 'James' to 'Sam' in name column of the data frame.

g) insert a new column in existing DataFrame.

h) get list from DataFrame column headers.

i) get the datatypes of columns of a DataFrame

```
import pandas as pd
import numpy as np
exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michae
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("\nOriginal data frame:")
print(df)

#a) calculate the sum of the examination attempts by the students.

print("\nSum of the examination attempts by the students:")
print(df['attempts'].sum())


#b) change the score in row 'd' to 11.5.

print("\nChange the score in row 'd' to 11.5:")
df.loc['d', 'score'] = 11.5
print(df)


#c) return the row having the maximum score

#find the rwo with the max score
#find the Max in Score column
m= df.score.max()
#find the row
result = df.loc[df['score'] == m]
print("Index-2: Details")
print(result)


#d) append a new row 'k' to DataFrame
#with given values for each column.

print("\nAppend a new row:")
df.loc['k'] = ['Suresh', 1,  15.5,'yes']
print("Print all records after insert a new record:")
print(df)


#e) delete the new row and return the original dataframe.

print("\nDelete the new row and display the original  rows:")
df = df.drop('k')
print(df)


#f) change the name 'James' to 'Sam' in name column of the data frame.

print("\nChange the name 'James' to ‘Sam’:")
df['name'] = df['name'].replace('James', 'Sam')
print(df)

#g) insert a new column in existing DataFrame.

color = ['Red','Blue','Orange','Red','White','White','Blue','Green','Green','Red']
df['color'] = color
print("\nNew DataFrame after inserting the 'color' column")
print(df)

#h) get list from DataFrame column headers.

print("Data lables of the columns of the DataFrame:")
print(df.columns)


#i) get the datatypes of columns of a DataFrame

print("Data types of the columns of the DataFrame:")
print(df.dtypes)
```