23rd Feb Assignment

March 1, 2023

1 Assignment 22

Q1. Create a Pandas Series that contains the following data: 4, 8, 15, 16, 23, and 42. Then, print the series.

Ans.The code for pandas series that contains the following data: 4, 8, 15, 16, 23, and 42.

[3]:
```
import pandas as pd
```

[4]:
```
s=pd.Series([4,8,15,16,23,42])
```

[5]:
```
print(s)
```

```
0     4
1     8
2    15
3    16
4    23
5    42
dtype: int64
```

Q2. Create a variable of list type containing 10 elements in it, and apply pandas.Series function on the variable print it.

[7]:
```
my_list=[10,20,30,40,50,60,70,80,90,100]
```

[8]:
```
my_series=pd.Series(my_list)
```

[9]:
```
print(my_series)
```

```
0     10
1     20
2     30
3     40
4     50
5     60
6     70
7     80
```

```
8      90
9     100
dtype: int64
```

Q3. Create a Pandas DataFrame that contains the following data:

Name	Age	Gender
Alice	25	Female
Bob	30	Male
Clair	27	Female

[10]:
```
data={'Name':['Alice','Bob','Clair'],
      'Age':[25,30,27],
     'Gender':['Female','Male','Female']}
```

[12]:
```
df=pd.DataFrame(data)
```

[15]:
```
print(df)
```

```
    Name  Age  Gender
0  Alice   25  Female
1    Bob   30    Male
2  Clair   27  Female
```

Q4. What is 'DataFrame' in pandas and how is it different from pandas.series? Explain with an example.

Ans.In Pandas, a DataFrame is a two-dimensional labeled data structure that is used to store and manipulate tabular data. It is similar to a spreadsheet or SQL table, where the data is organized in rows and columns. Each column in a DataFrame can have a different data type, such as numeric, string, or boolean, and can have a label or name associated with it.

A Pandas Series, on the other hand, is a one-dimensional labeled array that can hold data of any type, such as integers, strings, or floats. A Series can be thought of as a single column of a DataFrame, where the values are indexed by a unique label.

[16]:
```
# Creating a DataFrame
my_dataframe = pd.DataFrame({'Name': ['Alice', 'Bob', 'Claire'], 'Age': [25,␣
 ↪30, 27], 'Gender': ['Female', 'Male', 'Female']})

# Printing the DataFrame
print(my_dataframe)
```

```
     Name  Age  Gender
0   Alice   25  Female
```

```
1      Bob   30    Male
2   Claire   27  Female
```

[17]:

```
# Creating a Series
my_series = pd.Series([10, 20, 30, 40, 50])

# Printing the Series
print(my_series)
```

```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

Q5. What are some common functions you can use to manipulate data in a Pandas DataFrame? Can you give an example of when you might use one of these functions?

Ans.The some common function we can use to manipulate data in Pandas are

[18]:

```
# Creating a DataFrame
data = {'Name': ['Alice', 'Bob', 'Claire'], 'Age': [25, 30, 27], 'Gender':␣
 ↪['Female', 'Male', 'Female']}
df = pd.DataFrame(data)
```

1. head() and tail()

[23]:

```
df.head(1) #print 1 row from beginning
```

[23]:

```
    Name  Age  Gender
0  Alice   25  Female
```

[24]:

```
df.tail(1) #print 1 row from end
```

[24]:

```
     Name  Age  Gender
2  Claire   27  Female
```

2.describe()->This function is used to view some basic statistical information about a DataFrame, such as mean, standard deviation, minimum, and maximum values.

[25]:

```
df.describe()
```

[25]:

```
             Age
count   3.000000
mean   27.333333
std     2.516611
min    25.000000
25%    26.000000
```

```
50%     27.000000
75%     28.500000
max     30.000000
```

3.sort_values() - This function is used to sort a DataFrame by one or more columns.

```
[26]: df.sort_values(by='Age')
```

```
[26]:      Name  Age  Gender
      0   Alice   25  Female
      2  Claire   27  Female
      1     Bob   30    Male
```

Q6. Which of the following is mutable in nature Series, DataFrame, Panel?

Ans.In Pandas, both Series and DataFrame are mutable in nature, which means their contents can be modified after they are created. However, Panel is no longer recommended for use as it has been deprecated since version 0.25.0 and replaced with MultiIndex or xarray.

Q7. Create a DataFrame using multiple Series. Explain with an example.

```
[27]: name=pd.Series(['Alice','Bob','Chianx'])
      age=pd.Series([22,34,12])
      gender=pd.Series(['Female','Male','Female'])

      data={'Name':name,'Age':age,'Gender':gender}
      dataframe=pd.DataFrame(data)

      print(dataframe)
```

```
     Name  Age  Gender
0   Alice   22  Female
1     Bob   34    Male
2  Chianx   12  Female
```

In above example, we created three Series for name, age, and gender, each containing three elements. Then, we passed these Series as a dictionary to the pd.DataFrame() constructor to create a DataFrame with the same data. Finally, we printed the DataFrame to verify that it was created correctly.