**Class Notes**

Class: XII  
Date: 02-May-2020  

Subject: Informatics Practices  
Topic: 2. Python Pandas (Contd....)

→ All notes of this chapter to be written in notes copy.

### Functions count() and sum()

Example: A program to demonstrate the use of sum and count function and other descriptive statistics function which are already covered in this chapter.

```python
import pandas as pd  
import numpy as np  
  
#Create a Dictionary of series  
d = {'Name':pd.Series(['Sachin','Dhoni','Virat','Rohit','Shikhar']),  
     'Age':pd.Series([26,25,25,24,31]),  
     'Score':pd.Series([87,67,89,55,47])}  
  
#Create a DataFrame  
df = pd.DataFrame(d)  
print("Dataframe contents")  
print(df)  
print(df.count())  
print("count age",df['Age'].count())  
print("sum of score",df['Score'].sum())  
print("minimum age",df['Age'].min())  
print("maximum score",df['Score'].max())  
print("mean age",df['Age'].mean())  
print("mode of age",df['Age'].mode())  
print("median of score",df['Score'].median())
```

**OUTPUT**

```
Dataframe contents
    Name  Age  Score
  0 Sachin   26     87
  1 Dhoni   25     67
  2 Virat   25     89
  3 Rohit   24     55
  4 Shikhar 31     47
```

Name  5
Age  5
Score  5
dtype: int64
count age Age  5
dtype: int64
sum of score Score  345
dtype: int64

NB.: This sheet is prepared from home.
minimum age Age 24
dtype: int64
maximum score Score 89
dtype: int64
mean age Age 26.2
dtype: float64
mode of age Age
0 25
median of score Score 67.0
dtype: float64

SORTING

Example 1: Sort the dataframe in python pandas by index in ascending order:

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

d = {'Name':pd.Series(['Sachin','Dhoni','Virat','Rohit','Shikhar']),
     'Age':pd.Series([26,25,25,24,31]),
     'Score':pd.Series([87,67,89,55,47])}

#Create a DataFrame
df = pd.DataFrame(d)
df1=df.reindex([1,4,3,2,0])

print("Dataframe contents without sorting")
print (df)
df1=df.sort_index()
print("Dataframe contents after sorting")
print(df1)
```

OUTPUT

Dataframe contents without sorting

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dhoni</td>
<td>25</td>
<td>67</td>
</tr>
<tr>
<td>Shikhar</td>
<td>31</td>
<td>47</td>
</tr>
<tr>
<td>Rohit</td>
<td>24</td>
<td>55</td>
</tr>
<tr>
<td>Virat</td>
<td>25</td>
<td>89</td>
</tr>
<tr>
<td>Sachin</td>
<td>26</td>
<td>87</td>
</tr>
</tbody>
</table>

Dataframe contents after sorting

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sachin</td>
<td>26</td>
<td>87</td>
</tr>
<tr>
<td>Dhoni</td>
<td>25</td>
<td>67</td>
</tr>
<tr>
<td>Virat</td>
<td>25</td>
<td>89</td>
</tr>
<tr>
<td>Rohit</td>
<td>24</td>
<td>55</td>
</tr>
<tr>
<td>Shikhar</td>
<td>31</td>
<td>47</td>
</tr>
</tbody>
</table>

NB.: This sheet is prepared from home.
Example 2: Sorting pandas dataframe by index in descending order.

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

# Create a Dictionary of series
d = {'Name':pd.Series(['Sachin', 'Dhoni', 'Virat', 'Rohit', 'Shikhar']),
    'Age':pd.Series([26, 25, 25, 24, 31]),
    'Score':pd.Series([87, 67, 89, 55, 47])}

# Create a DataFrame
df = pd.DataFrame(d)
df = df.reindex([1, 4, 3, 2, 0])
print("Dataframe contents without sorting")
print(df)

df1=df.sort_index(ascending=0)
print("Dataframe contents after sorting")
print(df1)
```

**OUTPUT**

Dataframe contents without sorting

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dhoni</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>Shikhar</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>Rohit</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>Virat</td>
<td>25</td>
</tr>
<tr>
<td>0</td>
<td>Sachin</td>
<td>26</td>
</tr>
</tbody>
</table>

Dataframe contents after sorting

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Shikhar</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>Rohit</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>Virat</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>Dhoni</td>
<td>25</td>
</tr>
<tr>
<td>0</td>
<td>Sachin</td>
<td>26</td>
</tr>
</tbody>
</table>

**VACATION HOME WORK**

Do the following questions in your practical Note-Book.

1. Write a Pandas program to convert a Panda module Series to Python list and it’s type

   **Solution:**
   ```python
   import pandas as pd
   ds= pd.Series([2, 4, 6, 8, 10])
   print("Pandas Series and type")
   print(ds)
   print(type(ds))
   print("Convert Pandas Series to Python list")
   print(ds.tolist())
   print(type(ds.tolist()))
   ```

2. Write a Pandas program to compare the elements of the two Pandas Series??

   **Solution:**
   ```python
   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:")
print(ds1 == ds2)
print("Greater than:")
print(ds1 > ds2)
print("Less than:")
print(ds1 < ds2)

3 Write a Python program to convert a dictionary to a Pandas series. Sample Series:
Dictionary:
{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
Converted series:
a 100
b 200
c 300
d 400
e 800
dtype: int64
Solution:
import pandas as pd
d1 = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
print("Dictionary:")
print(d1)
s1 = pd.Series(d1)
print("Converted series:")
print(s1)

4 Write a Pandas program to add, subtract, multiple and divide two Pandas Series
Solution:
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])

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)

NB.: This sheet is prepared from home.
5. Write a program to sort the element of Series S1 into S2
   **Solution:**
   ```python
   import pandas as pd
   s1= pd.Series(['100', '200', 'python', '300.12', '400'])
   print("Series before sorting:")
   print(s1)
   s2= pd.Series(s1).sort_values()
   print("Series After sorting:")
   print(s2)
   ```

6. Write a Pandas program to select the rows where the height is not known, i.e. is NaN.
   ```python
   name: ['Asha', 'Radha', 'Kamal', 'Divy', 'Anjali'],
   height: [5.5, 5, np.nan, 5.9, np.nan],
   age: [11, 23, 22, 33, 22]
   ```
   **Solution:**
   ```python
   import pandas as pd
   import numpy as np
   pers_data = {'name': ['Asha', 'Radha', 'Kamal', 'Divy', 'Anjali'],
                'height': [5.5, 5, np.nan, 5.9, np.nan],
                'age': [11, 23, 22, 33, 22]}
   labels = ['a', 'b', 'c', 'd', 'e']
   df = pd.DataFrame(pers_data , index=labels)
   print("Persons whose height not known:")
   print(df[df['height'].isnull()])
   ```

7. Write a Pandas program to select the name of persons whose height is between 5 to 5.5 (both values inclusive)
   ```python
   name: ['Asha', 'Radha', 'Kamal', 'Divy', 'Anjali'],
   height: [5.5, 5, np.nan, 5.9, np.nan],
   age: [11, 23, 22, 33, 22]
   ```
   **Solution:**
   ```python
   import pandas as pd
   import numpy as np
   pers_data = {'name': ['Asha', 'Radha', 'Kamal', 'Divy', 'Anjali'],
                'height': [5.5, 5, np.nan, 5.9, np.nan],
                'age': [11, 23, 22, 33, 22]}
   labels = ['a', 'b', 'c', 'd', 'e']
   df = pd.DataFrame(pers_data , index=labels)
   print("Persons whose height is between 5 and 5.5")
   print(df[(df['height']>= 5 )& (df['height']<= 5.5)])
   ```

8. Write a panda program to read marks detail of Manasvi and Calculate sum of all marks.
   **Solution:**
   ```python
   import pandas as pd
   import numpy as np
   data = {'Manasvi': ['Physics', 'Chemistry', 'English', 'Maths', 'Computer Sc'],
           'marks': [89,99,97,99,98]}
   df = pd.DataFrame(data )
   ```
9 Write a Pandas program to sort the data frame first by 'Designation' in Ascending order, then by 'Name' in Descending order.

**Solution:**
import pandas as pd
data1 = {'Name':['Akshat', 'Sameer', 'Pihu', 'Neha'],
'Age':[28,34,29,42], 'Designation':['Accountant', 'Clerk', 'Clerk', 'Manager']}
df1 = pd.DataFrame(data1)
print (df1.sort_values(by=['Designation','Name'], ascending=[True,False]))

10 Give python code to generate the following output based given dataframe.

**Dataframe:**

<table>
<thead>
<tr>
<th>rollno</th>
<th>name</th>
<th>physics</th>
<th>chem</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Pat</td>
<td>90</td>
<td>75</td>
</tr>
<tr>
<td>101</td>
<td>Sid</td>
<td>40</td>
<td>80</td>
</tr>
<tr>
<td>103</td>
<td>Tom</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>102</td>
<td>Kim</td>
<td>95</td>
<td>85</td>
</tr>
<tr>
<td>104</td>
<td>Ray</td>
<td>65</td>
<td>60</td>
</tr>
</tbody>
</table>

**Desired Output:**
minimum is: 40
maximum is: 95
average is: 68.0
median is: 65.0
mode is:
0  40
1  50
2  65
3  90
4  95
dtype: int64

**Solution:**
import pandas as pd
d1 = {'rollno':[101,101,103,102,104], 'name': ['Pat','Sid','Tom','Kim','Ray'],
'physics':[90,40,50,95,65],'chem':[75,80,60,85,60] }
df = pd.DataFrame(d1)
print(df)
print('Demonstrating statistical functions min(), max(), mean(), median(), mode() for the subject Physics')
print('minimum is:',df['physics'].min())
print('maximum is:',df['physics'].max())
print('average is:',df['physics'].mean())
print('median is:',df['physics'].median())
print('mode is:')

-- X --

NB.: This sheet is prepared from home.