I] Descriptive Statistics

- Descriptive statistics is a means of describing features of a data set by generating summaries about data samples.
- A large number of methods collectively compute descriptive statistics and other related operations on DataFrame.
- These methods are:

S. No.	Function	Description
1	count()	Number of non-null observations
2	sum()	Sum of values
3	mean()	Mean of Values
4	median()	Median of Values
5	mode()	Mode of values
6	std()	Standard Deviation of the Values
7	min()	Minimum Value
8	max()	Maximum Value
9	abs()	Absolute Value
10	prod()	Product of Values
11	cumsum()	Cumulative Sum
12	cumprod()	Cumulative Product

Let us create a DataFrame

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
   'Lee','David','Gasper','Betina','Andres']),
   'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)

print(df)
```

```
      Name  Age  Rating
0      Tom   25    4.23
1    James   26    3.24
2    Ricky   25    3.98
3      Vin   23    2.56
4    Steve   30    3.20
5    Smith   29    4.60
6     Jack   23    3.80
7      Lee   34    3.78
8    David   40    2.98
9   Gasper   30    4.80
```

```
10  Betina   51    4.10
11  Andres   46    3.65
```

Let's apply descriptive statistical functions:

sum()

Returns the sum of the values for the requested axis. By default, axis is index (axis=0).

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
   'Lee','David','Gasper','Betina','Andres']),
   'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)
print(df.sum())
print(df)
```

```
Name      TomJamesRickyVinSteveSmithJackLeeDavidGasperBe...
Age                                                     382
Rating                                                44.92
dtype: object
      Name  Age  Rating
0      Tom   25    4.23
1    James   26    3.24
2    Ricky   25    3.98
3      Vin   23    2.56
4    Steve   30    3.20
5    Smith   29    4.60
6     Jack   23    3.80
7      Lee   34    3.78
8    David   40    2.98
9   Gasper   30    4.80
10  Betina   51    4.10
11  Andres   46    3.65
```

```
# sum of all salary stored in 'total'
df1 = df.sum(axis=1)
print(df1)
```

```
0     29.23
1     29.24
2     28.98
3     25.56
4     33.20
5     33.60
6     26.80
7     37.78
8     42.98
9     34.80
10    55.10
11    49.65
dtype: float64
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in Da
```

```
import pandas as pd
studentdetails = {
```

```
       "studentname":["Ram","Sam","Scott","Ann","John"],
       "mathematics" :[80,90,85,70,95],
       "science" :[85,95,80,90,75],
       "english" :[90,85,80,70,95]
             }
index_labels=['r1','r2','r3','r4','r5']

df = pd.DataFrame(studentdetails ,index=index_labels)
print(df)
```

```
   studentname  mathematics  science  english
r1         Ram           80       85       90
r2         Sam           90       95       85
r3       Scott           85       80       80
r4         Ann           70       90       70
r5        John           95       75       95
```

```
# Sum the rows of DataFrame
df['Sum'] = df.sum(axis=1)
print(df)

# If you have few columns to sum
df['Sum'] = df['mathematics'] + df['science'] + df['english']
print(df)
```

```
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  255
r2         Sam           90       95       85  270
r3       Scott           85       80       80  245
r4         Ann           70       90       70  230
r5        John           95       75       95  265
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  255
r2         Sam           90       95       85  270
r3       Scott           85       80       80  245
r4         Ann           70       90       70  230
r5        John           95       75       95  265
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in Da
```

```
# Create List of columns
col_list= ['studentname', 'mathematics', 'science']

# sum specific columns
df['Sum'] = df[col_list].sum(axis=1)
print(df)
```

```
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  165
r2         Sam           90       95       85  185
r3       Scott           85       80       80  165
r4         Ann           70       90       70  160
r5        John           95       75       95  170
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: FutureWarning: Dropping of nuisance columns in Da
  """
```

```
# Select 1 to 3 columns to sum
df['Sum']=df.iloc[:,1:3].sum(axis=1)
print(df)

# Select 1 and 2 columns to sum Using DataFrame.iloc[]
df['Sum']=df.iloc[:,[1,2]].sum(axis=1)
```

```
print(df)
```

```
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  165
r2         Sam           90       95       85  185
r3       Scott           85       80       80  165
r4         Ann           70       90       70  160
r5        John           95       75       95  170
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  165
r2         Sam           90       95       85  185
r3       Scott           85       80       80  165
r4         Ann           70       90       70  160
r5        John           95       75       95  170
```

```
# Using DataFrame.iloc[] to select 2 and 3 columns to sum
df['Sum']=df.iloc[:,[2,3]].sum(axis=1)
print(df)

# Sum columns Fee and Discount for row from r2 to r3
df['Sum'] = df.loc['r2':'r4',['mathematics','science']].sum(axis = 1)
print(df)
```

```
   studentname  mathematics  science  english  Sum
r1         Ram           80       85       90  175
r2         Sam           90       95       85  180
r3       Scott           85       80       80  160
r4         Ann           70       90       70  160
r5        John           95       75       95  170
   studentname  mathematics  science  english    Sum
r1         Ram           80       85       90    NaN
r2         Sam           90       95       85  185.0
r3       Scott           85       80       80  165.0
r4         Ann           70       90       70  160.0
r5        John           95       75       95    NaN
```

mean()

Calculates the mean or average value by using DataFrame/Series.mean() method.

```
#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
   'Lee','David','Gasper','Betina','Andres']),
   'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)
print(df)
```

```
      Name  Age  Rating
0      Tom   25    4.23
1    James   26    3.24
2    Ricky   25    3.98
3      Vin   23    2.56
4    Steve   30    3.20
5    Smith   29    4.60
6     Jack   23    3.80
7      Lee   34    3.78
8    David   40    2.98
9   Gasper   30    4.80
```

```
10  Betina   51    4.10
11  Andres   46    3.65
```

```
# Calculate the Mean of 'Age' column
mean = df['Age'].mean()

# Print mean
print(mean)
```

```
31.833333333333332
```

median()

Calculates the median value by using DataFrame/Series.median() method.

```
# Calculate Median of 'Age' column
median = df['Age'].median()

# Print median
print(median)
```

```
29.5
```

mode()

Calculates the mode or most frequent value by using DataFrame.mode() method.

```
# Calculate Mode of 'Age' column
mode = df['Age'].mode()

# Print mode
print(mode)
```

```
0    23
1    25
2    30
dtype: int64
```

count()

Calculates the count or frequency of non-null values by using DataFrame/Series.count() Method.

```
# Calculate Count of 'Name' column
count = df['Name'].count()

# Print count
print(count)
```

```
12
```

Standard Deviation Function: std()

Calculates the standard deviation of values by using DataFrame/Series.std() method.

```
# Calculate Standard Deviation
# of 'Fare' column
std = df['Rating'].std()

# Print standard deviation
print(std)
```

```
0.6616279280316959
```

max()

Calculates the maximum value using DataFrame/Series.max() method.

```
# Calculate Maximum value in 'Age' column
maxValue = df['Age'].max()

# Print maxValue
print(maxValue)
```

```
51
```

min()

Calculates the minimum value using DataFrame/Series.min() method.

```
# Calculate Minimum value in 'Fare' column
minValue = df['Age'].min()

# Print minValue
print(minValue)
```

```
23
```

II] Summarizing Data

describe()

> The describe() function computes a summary of statistics pertaining to the DataFrame columns.

A) Summary Statistic of the numeric columns:

describe() Function excludes character column and calculate summary statistics only for numeric columns

B) Summary Statistic of the character columns:

describe() Function with an argument named include along with value object i.e include='object' gives the summary statistics of the character columns.

C) Summary Statistic of all the columns

describe() Function with include='all' gives the summary statistics of all the columns.

Example 1:

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith','Jack',
   'Lee','David','Gasper','Betina','Andres']),
   'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])
}

#Create a DataFrame
df = pd.DataFrame(d)

print(df)
```

```
          Name  Age  Rating
     0      Tom   25    4.23
     1    James   26    3.24
     2    Ricky   25    3.98
     3      Vin   23    2.56
     4    Steve   30    3.20
     5    Smith   29    4.60
     6     Jack   23    3.80
     7      Lee   34    3.78
     8    David   40    2.98
     9   Gasper   30    4.80
     10  Betina   51    4.10
     11  Andres   46    3.65
```

```
# summary statistics

print(df.describe())
```

```
                Age     Rating
     count  12.000000  12.000000
     mean   31.833333   3.743333
     std     9.232682   0.661628
     min    23.000000   2.560000
     25%    25.000000   3.230000
     50%    29.500000   3.790000
     75%    35.500000   4.132500
     max    51.000000   4.800000
```

```
# summary statistics of character column

print(df.describe(include=['object']))
```

```
            Name
     count    12
     unique   12
     top     Tom
     freq      1
```

```
# summary statistics of all the column

print(df.describe(include='all'))
```

```
            Name        Age     Rating
     count    12  12.000000  12.000000
     unique   12        NaN        NaN
     top     Tom        NaN        NaN
     freq      1        NaN        NaN
     mean    NaN  31.833333   3.743333
     std     NaN   9.232682   0.661628
     min     NaN  23.000000   2.560000
     25%     NaN  25.000000   3.230000
     50%     NaN  29.500000   3.790000
     75%     NaN  35.500000   4.132500
     max     NaN  51.000000   4.800000
```

Example 2:

```
import pandas as pd
studentdetails = {
       "studentname":["Ram","Sam","Scott","Ann","John"],
       "mathematics" :[80,90,85,70,95],
       "science" :[85,95,80,90,75],
       "english" :[90,85,80,70,95]
             }
index_labels=['r1','r2','r3','r4','r5']
df = pd.DataFrame(studentdetails ,index=index_labels)
print(df)
```

```
   studentname  mathematics  science  english
r1         Ram           80       85       90
r2         Sam           90       95       85
r3       Scott           85       80       80
r4         Ann           70       90       70
r5        John           95       75       95
```

```
# summary statistics

print(df.describe())
```

```
       mathematics    science    english
count     5.000000   5.000000   5.000000
mean     84.000000  85.000000  84.000000
std       9.617692   7.905694   9.617692
min      70.000000  75.000000  70.000000
25%      80.000000  80.000000  80.000000
50%      85.000000  85.000000  85.000000
75%      90.000000  90.000000  90.000000
max      95.000000  95.000000  95.000000
```

```
# summary statistics of character column

print(df.describe(include=['object']))
```

```
       studentname
count            5
unique           5
top            Ram
freq             1
```

```
# summary statistics of all the column

print(df.describe(include='all'))
```

```
       studentname  mathematics    science    english
count            5     5.000000   5.000000   5.000000
unique           5          NaN        NaN        NaN
top            Ram          NaN        NaN        NaN
freq             1          NaN        NaN        NaN
mean           NaN    84.000000  85.000000  84.000000
std            NaN     9.617692   7.905694   9.617692
min            NaN    70.000000  75.000000  70.000000
25%            NaN    80.000000  80.000000  80.000000
50%            NaN    85.000000  85.000000  85.000000
75%            NaN    90.000000  90.000000  90.000000
max            NaN    95.000000  95.000000  95.000000
```

0s completed at 16:15