Answers to Question Set 6

Date: 27/04/2020   Name: D.Saravanan

Question 1:

Write a python program to generate 9 random integers in numpy. Convert it to a 3×3 matrix and then convert it to a dataframe.

Program:

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

rlist = []; [rlist.append(np.random.randint(0,9)) for n in range(9)]
print("9 random elemets: \n{}".format(rlist))

nlist = np.array([rlist[i:i+3] for i in range(0, len(rlist), 3)])
print("\n3x3 matrix format: \n{}".format(nlist))

df = pd.DataFrame(nlist, index = ['x', 'y', 'z'], columns = ['a', 'b', 'c'])
print("\nPandas DataFrame format: \n{}".format(df))
```

Output:

```
9 random elemets:
[3, 3, 7, 4, 2, 5, 5, 7, 4]

3x3 matrix format:
[[3 3 7]
 [4 2 5]
 [5 7 4]]

Pandas DataFrame format:
   a  b  c
x  3  3  7
y  4  2  5
z  5  7  4
```

Question 2:

Write a python program to generate a dataframe with the random values in column and calculate the mean and the standard deviation.

Program:

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

N = int(input("Enter the number of terms: "))
alist = []; blist = []; clist = []; dlist = []

for n in range(N):
    alist.append(np.random.randint(0,25))
    blist.append(np.random.randint(25,50))
    clist.append(np.random.randint(50,75))
    dlist.append(np.random.randint(75,100))

data = {'a':[n for n in alist], 'b':[n for n in blist], 'c':[n for n in clist], 'd':[n
    for n in dlist]}
df = pd.DataFrame(data); df['mean'] = df.mean(axis=1); df['stdev'] = df.std(axis=1)
print(df)
```

Output:

```
Enter the number of terms: 5
    a   b   c   d    mean      stdev
0  23  37  62  78  50.00  21.365861
1  10  25  70  92  49.25  33.116272
2  23  36  74  97  57.50  29.516944
3   5  36  56  96  48.25  33.017988
4  24  42  53  88  51.75  23.349251
```

Question 3:

Write a python program to plot the mean and the standard deviation from the dataframe of previous question using pandas inbuilt plot function and save the dataframe as JSON file format.

Program:

```
df[['mean', 'stdev']].plot(grid=True, kind='line', subplots=True)
plt.legend(loc='upper right'); plt.savefig('dframe.png')

df.to_json('data.json') # save dataframe in json format
```

Output: