```
#Q1
from scipy.spatial import distance
A=(1,2,3,4,5,6)
B=(7,8,9,10,11,12)
A,B
minkowski_distance = distance.minkowski(A, B, p=3)
print('Minkowski Distance b/w', A, 'and', B, 'is: ', minkowski_distance)

#Q2
import numpy as np
a = np.arange(9).reshape((3,3))
print("Original flattened array:")
print(a)
print("Weighted average along the specified axis of the above flattened array:")
print(np.average(a, axis=1, weights=[1./4, 2./4, 2./4]))

#Q3
import numpy as np
x = np.array([0, 1, 3])
y = np.array([2, 4, 5])
print("\nOriginal array1:")
print(x)
print("\nOriginal array2:")
print(y)
print("\nCross-correlation of the said arrays:\n",np.cov(x, y))

import io
import pandas as pd
df = pd.read_csv("titanic.csv")
df

df.describe()

df.mean()

df.isnull()

import io
import pandas as pd
df = pd.read_csv("nursery-growers-and-greenhouse.csv")
df

print("Given Dataframe is :\n",df)

# bydefault splitting is done on the basis of single space.
print("\nSplitting 'Name' column into two different columns
:\n",df.City.str.split(expand=True))

from numpy import mean
df.groupby(["City"]).mean()
```

```
#Q6 Arithmetic mean
import pandas as pd
import numpy as np
data={'Name':pd.Series(['Hikigaya','Hisoka','Roy','Joseph','Gojo','Yato','Spike','K
akashi','Saitama','Rem']),'Python':pd.Series([80,82,60,72,88,87,61,89,81,96]),'Elec
tronics':pd.Series([72,92,77,89,73,91,75,92,93,76]),'Statistics':pd.Series([78,77,6
8,62,88,76,63,98,93,67]),'FDS':([90,94,80,73,95,93,67,97,65,89]),'TCS':([88,68,69,6
5,58,84,72,91,75,89])}
df=pd.DataFrame(data)
df

result=pd.DataFrame.mean(df)
print('Arithmetic Mean:',result)
df

#Row wise geometric mean
import scipy
from scipy import stats
df['Geometric Mean']=stats.gmean(df.iloc[:, 1:3], axis=1)
df

#column wise Geometric mean
Python=scipy.stats.gmean(df.loc[:, "Python"])
print("Geometric mean of Python=",Python)
Electronics=scipy.stats.gmean(df.loc[:, "Electronics"])
print("Geometric mean of Electronics=",Electronics)
Statistics=scipy.stats.gmean(df.loc[:, "Statistics"])
print("Geometric mean of Statistics=",Statistics)
FDS=scipy.stats.gmean(df.loc[:, "FDS"])
print("Geometric mean of FDS=",FDS)
TCS=scipy.stats.gmean(df.loc[:, "TCS"])
print("Geometric mean of TCS=",TCS)

#Row wise Harmonic mean
import scipy
from scipy import stats
scipy.stats.hmean(df.iloc[:,1:3],axis=1)

#column wise Harmonic mean
Python=scipy.stats.hmean(df.loc[:,"Python"])
print("Harmonic mean of Python =",Python)
Electronics=scipy.stats.hmean(df.loc[:,"Electronics"])
print("Harmonic mean of Electronics =",Electronics)
Statistics=scipy.stats.hmean(df.loc[:,"Statistics"])
print("Harmonic mean of SDtatistics =",Statistics)
FDS=scipy.stats.hmean(df.loc[:,"FDS"])
print("Harmonic mean of FDS =",FDS)
TCS=scipy.stats.hmean(df.loc[:,"TCS"])
print("Harmonic mean of TCS =",TCS)
```

```
#Q7 Pandas Profiling
#pip install
https://github.com/pandas-profiling/pandas-profiling/archive/master.zip

import pandas as pd
df = pd.read_csv("titanic.csv")
print(df.shape)
df.head()

import numpy as np
import seaborn as sns

import numpy as np
import pandas as pd
from pandas_profiling import ProfileReport
df = pd.DataFrame(np.random.rand(100, 5), columns=["a", "b", "c", "d", "e"])

profile=ProfileReport(df, title="Pandas Profiling Report")

profile=ProfileReport(df, title="Pandas Profiling Report", explorative=True)

profile.to_widgets()

profile.to_notebook_iframe()

profile.to_file("your_report.html")

# as a string
json_data = profile.to_json()

#as a file
profile.to_file("your_report.json")

profile = ProfileReport(df, minimal=True)
profile.to_file(output_file="REPORT.html")

#! ls - this is for linux
get_ipython().system(' dir')
#above one is for windows

#! cat REPORT.html - this is for linux
get_ipython().system(' type REPORT.html')
#above one is for windows
```