AIM

Programs to handle data using pandas

Programming Code:

```
import numpy as np

import pandas as pd

s = pd.Series([1, 3, 5, 6, 8])
print(s)
```

OUTPUT:

```
0    1
1    3
2    5
3    6
4    8
dtype: int64
```

Programming Code:

```
dict = {"country": ["Brazil", "Russia", "India", "China", "South
Africa"],
       "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing",
"Pretoria"],
       "area": [8.516, 17.10, 3.286, 9.597, 1.221],
       "population": [200.4, 143.5, 1252, 1357, 52.98] }
b = pd.DataFrame(dict)
print(b)
```

OUTPUT:

```
        country    capital    area  population
0        Brazil   Brasilia   8.516      200.40
1        Russia     Moscow  17.100      143.50
2         India  New Dehli   3.286     1252.00
3         China    Beijing   9.597     1357.00
4  South Africa   Pretoria   1.221       52.98
```

Programming Code:

```
b.index = ["BR", "RU", "IN", "CH", "SA"]
print(b)
```

OUTPUT:

```
          country    capital    area  population
BR         Brazil   Brasilia   8.516      200.40
RU         Russia     Moscow  17.100      143.50
IN          India  New Dehli   3.286     1252.00
CH          China    Beijing   9.597     1357.00
SA   South Africa   Pretoria   1.221       52.98
```

Dataset used: cars1.csv

Programming Code:

```
import pandas as pd
cars = pd.read_csv('cars1.csv')
print(cars)
```

OUTPUT:

	Car	Model	Volume	Weight	CO2
0	Toyoty	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105
5	VW	Up!	1000	929	105
6	Skoda	Fabia	1400	1109	90
7	Mercedes	A-Class	1500	1365	92
8	Ford	Fiesta	1500	1112	98
9	Audi	A1	1600	1150	99
10	Hyundai	I20	1100	980	99
11	Suzuki	Swift	1300	990	101
12	Ford	Fiesta	1000	1112	99
13	Honda	Civic	1600	1252	94
14	Hundai	I30	1600	1326	97
15	Opel	Astra	1600	1330	97
16	BMW	1	1600	1365	99
17	Mazda	3	2200	1280	104
18	Skoda	Rapid	1600	1119	104
19	Ford	Focus	2000	1328	105
20	Ford	Mondeo	1600	1584	94
21	Opel	Insignia	2000	1428	99
22	Mercedes	C-Class	2100	1365	99
23	Skoda	Octavia	1600	1415	99
24	Volvo	S60	2000	1415	99
25	Mercedes	CLA	1500	1465	102
26	Audi	A4	2000	1490	104
27	Audi	A6	2000	1725	114
28	Volvo	V70	1600	1523	109
29	BMW	5	2000	1705	114
30	Mercedes	E-Class	2100	1605	115
31	Volvo	XC70	2000	1746	117
32	Ford	B-Max	1600	1235	104
33	BMW	216	1600	1390	108
34	Opel	Zafira	1600	1405	109
35	Mercedes	SLK	2500	1395	120

Programming Code:

```
# Print out first 4 observations
print(cars[0:4])

# Print out fifth and sixth observation
print(cars[4:6])
```

OUTPUT:

```
          Car       Model  Volume  Weight  CO2
0      Toyoty        Aygo    1000     790   99
1  Mitsubishi  Space Star    1200    1160   95
2       Skoda      Citigo    1000     929   95
3        Fiat         500     900     865   90
    Car   Model  Volume  Weight  CO2
4  Mini  Cooper    1500    1140  105
5    VW     Up!    1000     929  105
```

Programming Code:

```
import pandas as pd
cars = pd.read_csv('cars1.csv', index_col = 0) #first column is t
aen as index column


print(cars.iloc[2])
```

OUTPUT:

```
Model     Citigo
Volume      1000
Weight       929
CO2           95
Name: Skoda, dtype: object
```

Programming Code:

```
#Slicing dataframe
import pandas as pd

df = pd.DataFrame([['Jay','M',18],['Jennifer','F',17],
                   ['Preity','F',19],['Neil','M',17]],
                  columns = ['Name','Gender','Age'])
print(df)
df1 = df.iloc[2:,:]
df2 = df.iloc[:2,:]
print(df1)
print(df2)
```

OUTPUT:

```
      Name Gender  Age
0      Jay      M   18
1  Jennifer      F   17
2    Preity      F   19
3      Neil      M   17
     Name Gender  Age
2  Preity      F   19
3    Neil      M   17
      Name Gender  Age
0      Jay      M   18
1  Jennifer      F   17
```

Programming Code:

```
import pandas as pd
import numpy as np

#Create a series with 4 random numbers
s = pd.Series(np.random.randn(4))
print(s)

print ("The actual data series is:")
print( s.values)
```

OUTPUT:

```
0   -1.138968
1   -1.097746
2    0.109717
3    1.159537
dtype: float64
The actual data series is:
[-1.13896826 -1.09774589  0.10971687  1.15953676]
```

Programming Code:

```
print (s.head(2))
```

OUTPUT:

```
0   -1.138968
1   -1.097746
dtype: float64
```

Programming Code:

```
print(s.tail(3))
```

OUTPUT:

```
1   -1.097746
2    0.109717
3    1.159537
dtype: float64
```

Programming Code:

```
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith
','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,23]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

# Create a DataFrame
df = pd.DataFrame(d)
print(df)
print ("The transpose of the data series is:")
print(df.T)
```

OUTPUT:

```
     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
The transpose of the data series is:
          0      1      2     3      4      5     6
Name     Tom  James  Ricky   Vin  Steve  Smith  Jack
Age       25     26     25    23     30     29    23
Rating  4.23   3.24   3.98  2.56    3.2    4.6   3.8
```

Programming Code:

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith
','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,23]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

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

```
print(df)
print ("Row axis labels and column axis labels are:")
print (df.axes)
```

OUTPUT:

```
    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
Row axis labels and column axis labels are:
[RangeIndex(start=0, stop=7, step=1), Index(['Name', 'Age', 'Rating'], dtype='object')]
```

Programming Code:

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith
','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,23]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(d)
print ("The data types of each column are:")
print (df.dtypes)
```

OUTPUT:

```
The data types of each column are:
Name       object
Age         int64
Rating    float64
dtype: object
```

Programming Code:

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith
','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,23]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(d)
print ("Is the object empty?")
print (df.empty)
```

OUTPUT:

```
Is the object empty?
False
```

Programming Code:

```
import pandas as pd
import numpy as np

#Create a Dictionary of series
d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smith
','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,23]),  'Rating':pd.Series([
4.23,3.24,3.98,2.56,3.20,4.6,3.8])
   }

#Create a DataFrame
df = pd.DataFrame(d)
print ("Our object is:")
print (df)
print ("The dimension of the object is:")
print (df.ndim)
```

OUTPUT:

```
Our object is:
    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
The dimension of the object is:
2
```

Programming Code:

```
 d = {'Name':pd.Series(['Tom','James','Ricky','Vin','Steve','Smit
h','Jack']),
   'Age':pd.Series([25,26,25,23,30,29,30]),
   'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8])}

#Create a DataFrame
df = pd.DataFrame(d)
print(df)
print ("Our object is:")
print ("The shape of the object is:")
print (df.shape)
```

OUTPUT:

```
    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   30    3.80
Our object is:
The shape of the object is:
(7, 3)
```

Programming code:

```
print (df.size)

print (df.values)
```

OUTPUT:

```
21


  [['Tom' 25 4.23]
   ['James' 26 3.24]
   ['Ricky' 25 3.98]
   ['Vin' 23 2.56]
   ['Steve' 30 3.2]
   ['Smith' 29 4.6]
   ['Jack' 30 3.8]]
```

Programming code:

```
df.isnull().sum()
```

OUTPUT:

```
Name      0
Age       0
Rating    0
dtype: int64
```

Programming code:

```
df = pd.DataFrame(np.arange(12).reshape(3, 4),

                  columns=['A', 'B', 'C', 'D'])
print(df)
```

OUTPUT:

```
   A  B   C   D
0  0  1   2   3
1  4  5   6   7
2  8  9  10  11
```