

DSc-prac

March 7, 2024

```
[1]: import pandas as pd
import numpy as np
from IPython.display import HTML, Markdown, Latex

def display_df(tp_df=None, index=False):
    tp_df = tp_df if tp_df is not None else df
    display(Markdown(tp_df.to_markdown(index=index)))
```

1 Q1) Create a pandas series from a dictionary of values and an ndarray

```
[2]: import pandas as pd
import numpy as np

data = {
    'Name': np.array([ 'Ram', 'Manish', 'Sahil', 'Amrit', 'Mark' ]),
    'Age': np.array([ 16, 18, 15, 20, 19 ]),
    'Gender': np.array([ 'M', 'M', 'M', 'F', 'M' ])
}
df = pd.DataFrame(data)
display_df(df)
```

Name	Age	Gender
Ram	16	M
Manish	18	M
Sahil	15	M
Amrit	20	F
Mark	19	M

2 Q2) Create a Series and print all the elements that are above 75th percentile

```
[3]: import pandas as pd

data = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print('Original series:')
display_df(data)

percentile_75 = data.quantile(0.75)
above_75      = data[ data > percentile_75 ].to_list()
print('\nElements above 75th percentile:')
print(above_75)
```

Original series:

—
0
—
1
2
3
4
5
6
7
8
9
10
—

Elements above 75th percentile:
[8, 9, 10]

3 Q3) Perform sorting on Series data and DataFrames

```
[4]: import pandas as pd

# DataFrame
data = {
    'name':    ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
    'year':    [2012, 2012, 2013, 2014, 2014],
    'reports': [4, 24, 31, 2, 3]
}
df      = pd.DataFrame(data)
sorted_df = df.sort_values(by='reports', ascending=False)
print('Sorted dataframe (based on reports) in descending order:')
display_df(sorted_df)
```

```
# Series
s = pd.Series([3, 1, 2, 3, 4])
s.sort_values(inplace=True)
print('Sorted series in ascending order:')
display_df(s)
```

Sorted dataframe (based on reports) in descending order:

name	year	reports
Tina	2013	31
Molly	2012	24
Jason	2012	4
Amy	2014	3
Jake	2014	2

Sorted series in ascending order:

```
-
0
-
1
2
3
3
4
-
```

4 Q4) Write a program to implement pivot() and pivot-table() on a DataFrame

```
[5]: import pandas as pd

data = {
    'Day':      ['Monday', 'Monday', 'Tuesday', 'Tuesday', 'Wednesday',
↪ 'Wednesday'],
    'City':     ['Delhi', 'Mumbai', 'Delhi', 'Mumbai', 'Delhi', 'Mumbai'],
    'Temperature': [32, 34, 33, 35, 34, 36],
}
df = pd.DataFrame(data)
print('Dataframe for pivot(): ')
display_df(df)

pivot_df = df.pivot(index='Day', columns='City', values='Temperature')
print('\npivot(): ')
display_df(pivot_df, index=True)
```

```

data = {
    'Day':          ['Monday', 'Monday', 'Monday', 'Tuesday', 'Tuesday', 'Tuesday'],
    'City':         ['Delhi', 'Delhi', 'Mumbai', 'Delhi', 'Mumbai', 'Mumbai'],
    'Temperature': [32, 33, 36, 33, 36, 37],
}
df = pd.DataFrame(data)
print('\nDataframe for pivot_table(): ')
display_df(df)

pivot_table_df = df.pivot_table(index='Day', columns='City',
    values='Temperature', aggfunc='count')
print('\npivot_table(): ')
display_df(pivot_table_df, index=True)

```

Dataframe for pivot():

Day	City	Temperature
Monday	Delhi	32
Monday	Mumbai	34
Tuesday	Delhi	33
Tuesday	Mumbai	35
Wednesday	Delhi	34
Wednesday	Mumbai	36

pivot():

Day	Delhi	Mumbai
Monday	32	34
Tuesday	33	35
Wednesday	34	36

Dataframe for pivot_table():

Day	City	Temperature
Monday	Delhi	32
Monday	Delhi	33
Monday	Mumbai	36
Tuesday	Delhi	33
Tuesday	Mumbai	36
Tuesday	Mumbai	37

pivot_table():

Day	Delhi	Mumbai
Monday	2	1
Tuesday	1	2

- 5 Q5) Write a program to find mean absolute deviation on a DataFrame
- 6 Q6) Two Series object, Population stores the details of four metro cities of India and another object AvgIncome stores the total average income reported in four years in these cities. Calculate income per capita for each of these metro cities
- 7 Q7) Create a DataFrame based on E-Commerce data and generate mean, mode, and median

```
[6]: import pandas as pd

data = {
    'Order_ID':    ['101', '102', '103', '104', '105'],
    'Price':       [50, 20, 40, 50, 45],
    'Quantity':    [2, 3, 1, 2, 1]
}
df = pd.DataFrame(data)
print('Original dataframe:')
display_df(df)

mean_df    = df.mean(numeric_only=True)
mode_df    = df.mode(numeric_only=True)
median_df  = df.median(numeric_only=True)

print('\nMeans: ')
display_df(mean_df, index=True)
print('\nModes: ')
display_df(mode_df, index=True)
print('\nMedians: ')
display_df(median_df, index=True)
```

Original dataframe:

Order_ID	Price	Quantity
101	50	2
102	20	3
103	40	1

Order_ID	Price	Quantity
104	50	2
105	45	1

Means :

	0
Price	41
Quantity	1.8

Modes :

	Price	Quantity
0	50	1
1	nan	2

Medians :

	0
Price	45
Quantity	2

- 8 Q8) Create a DataFrame based on employee data and generate quartile and variance
- 9 Q9) Write a program to implement skewness on random data
- 10 Q10) Create a dataframe on any data and compute statistical function of kurtosis
- 11 Q11) Series objects temp1, temp2, temp3, and temp 4 stores the temperature of days of week 1, week 2, week 3, week 4. Write a script to:
 - 1) Print average temperature per week
 - 2) Print average temperature of entire month

12 Q12) Write a program to read a csv file and create its dataframe

```
[7]: import pandas as pd

filename = 'data.csv'
print(f'Contents of {filename}:')
with open(filename) as f:
    print(f.read())

df = pd.read_csv(filename)
print(f'\nDataframe:')
display_df(df)
```

Contents of data.csv:

Name, Age, Gender
Ram, 16, M
Manish, 18, M
Sahil, 15, M
Amrit, 20, F
Mark, 19, M

Dataframe:

Name	Age	Gender
Ram	16	M
Manish	18	M
Sahil	15	M
Amrit	20	F
Mark	19	M

13 Q13) Consider the dataframe QtrSales where each row contains the item category, item name and expenditure and group the rows by category, and print the average expenditure per category

```
[8]: import pandas as pd

QtrSales = pd.DataFrame({
    'category':    ['Electronics', 'Electronics', 'Fashion', 'Fashion', 'Fashion', 'Fashion'],
    'item_name':   ['Laptop', 'Headphones', 'T-Shirt', 'Jeans', 'Smartphone', 'Shoes'],
    'expenditure': [1200, 100, 31, 50, 800, 60]
})
```

```

})
print('Original dataframe:')
display_df(QtrSales)

grouped = QtrSales.groupby(by='category')['expenditure']
mean_df = grouped.mean()
print('\nAverage expenditure per category:')
display_df(mean_df, index=True)

```

Original dataframe:

category	item_name	expenditure
Electronics	Laptop	1200
Electronics	Headphones	100
Fashion	T-Shirt	31
Fashion	Jeans	50
Electronics	Smartphone	800
Fashion	Shoes	60

Average expenditure per category:

category	expenditure
Electronics	700
Fashion	47

14 Q14) Create a DataFrame having age, name, weight of five students. Write a program to display only the weight of first and fourth rows

```

[9]: import pandas as pd

data = {
    'name':  ['John', 'Emma', 'Michael', 'Sophia', 'William'],
    'age':   [20, 21, 22, 20, 23],
    'weight': [70, 65, 75, 68, 72]
}
df = pd.DataFrame(data)
print('Original dataframe:')
display_df(df, index=True)

weight_df = df.loc[ [0,3] , ['weight'] ]
print('\nWeight of the first and fourth rows:')
display_df(weight_df, index=True)

```


Original dataframe:

	name	age	weight
0	John	20	70
1	Emma	21	65
2	Michael	22	75
3	Sophia	20	68
4	William	23	72

Weight of the first and fourth rows:

	weight
0	70
3	68

15 Q15) Write a program to create a DataFrame to store weight, age and name of three people. Print the dataframe and its transpose

```
[10]: import pandas as pd

data = {
    'name':    ['John', 'Emma', 'Michael', 'Sophia', 'William'],
    'age':     [20, 21, 22, 20, 23],
    'weight':  [70, 65, 75, 68, 72]
}
df = pd.DataFrame(data)
print('Original dataframe:')
display_df(df, index=True)

print('\nTranspose of the dataframe:')
display_df(df.T, index=True)
```

Original dataframe:

	name	age	weight
0	John	20	70
1	Emma	21	65
2	Michael	22	75
3	Sophia	20	68
4	William	23	72

Transpose of the dataframe:

	0	1	2	3	4
name	John	Emma	Michael	Sophia	William
age	20	21	22	20	23
weight	70	65	75	68	72