Pandas

1.Pandas Series (https://pandas.pydata.org/pandas-docs/stable/reference/series.html)

resource - https://www.youtube.com/watch?v=m7gxnZx2vT4 (https://www.youtube.com/watch?v=m7gxnZx2vT4)

1*4=4 points

Import pandas and numpy with their aliases

In [2]:

```
```

Create a variable a = pd.Series([100, 200, 300, 400])

In [3]:

```
```

Print a, and data type

In [4]:

```
```

```
0    100
1    200
2    300
3    400
dtype: int64
<class 'pandas.core.series.Series'>
```

Using indexing access the element 300 from the series a.

In [5]:

```
```

Out[5]:

```
300
```

What are the values of index for series a?

2*4 = 8 points

In [6]:

```
```

Out[6]:

```
RangeIndex(start=0, stop=4, step=1)
```

Change the index to ['c', 'a', 'b', 'd']

In [7]:

```
```

```
c    100
a    200
b    300
d    400
dtype: int64
```

Access the value in the series with index 'd'

In [8]:

```
```

Out[8]:

```
400
```

Sort the values wrt to the index and print it

In [9]:

```
```

```
a    200
b    300
c    100
d    400
dtype: int64
```

Create a new Pandas Series b having index as 'e', 'f', and 'g' and value 800,450,100 and print it

4*2=8 points

In [10]:

```
b=# code here
```

```
e    800
f    450
g    100
dtype: int64
```

Append b series at the end of a series

In [11]:

```
```

In [12]:

```
#print a again after appending b into it
```

```
a    200
b    300
c    100
d    400
e    800
f    450
g    100
dtype: int64
```

Sort the values in descending order of a and print the index of the sorted series

In [13]:

```
a=#code here
```

In [14]:

```
# print index of a
```

```
Index(['e', 'f', 'd', 'b', 'a', 'g', 'c'], dtype='object')
```

2. Pandas DataFrame (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html)

Reference:-https://www.youtube.com/watch?v=KB-19V-cSs4 (https://www.youtube.com/watch?v=KB-19V-cSs4)

8*2 = 16 points

Part 1

5 points

Create a pandas dataframe df from the series 'a' that we used in the last section, print the dataframe

In [15]:

```
df
```

```
     0
e  800
f  450
d  400
b  300
a  200
g  100
c  100
```

Point to ponder! Don't you think series a and dataframe df looks alike? But there must be some difference between them. Hey buddy think and write your thoughts below.

In [ ]:

```
# happy to know your comment
```

2 + 1*4 = 6 points

What is the shape of the datafarme

In [16]:

```

```

Out[16]:

```
(7, 1)
```

Hey! remember shape (7,1) implies dataframe has 7 rows and 1 column.

What is the index of the dataframe, is it same as the series 'a' ?

In [17]:

```

```

Out[17]:

```
Index(['e', 'f', 'd', 'b', 'a', 'g', 'c'], dtype='object')
```

print the head and tail of the dataframe.
Additional - (what does head and tali represent?)

In [18]:

```
#head
```

Out[18]:

	0
e	800
f	450
d	400
b	300
a	200

In [19]:

```
#tail
```

Out[19]:

	0
d	400
b	300
a	200
g	100
c	100

Rename the columns of the dataframe as 'points'

2 points

In [20]:

```

```

```
   points
e     800
f     450
d     400
b     300
a     200
g     100
c     100
```

Create another Series 'fruits', which contains random names of fruits from ['orange','mango','apple']. The series should contain 7 elements, randomly selected from ['orange','mango','apple']

5 points

In [21]:

```
#Create fruits array
```

Out[21]:

```
array(['orange', 'orange', 'mango', 'mango', 'mango', 'mango', 'mango'],
      dtype='<U6')
```

The above output might vary based randomness

In [22]:

```
#Create series fruits out of fruits array
```

```
0    orange
1    orange
2     mango
3     mango
4     mango
5     mango
6     mango
dtype: object
```

Change the index of fruits to the index of dataframe df

5 points

In [23]:

```

```

```
e    orange
f    orange
d     mango
b     mango
a     mango
g     mango
c     mango
dtype: object
```

Add this fruits series as a new column to the dataframe df with its column name as 'fruits'
print the head of the dataframe to verify

In [24]:

```
```

Out[24]:

	points	fruits
e	800	orange
f	450	orange
d	400	mango
b	300	mango
a	200	mango

Bonus optional question:
What if we had to changed the index of the dataframe?
Try to add any series to the dataframe with the origianl index (when index of series is not same as to the index of dataframe)

Use series as pd.Series([100,200,300,400,500,600,700]) and try to add it to df with column name Bonus_number.

Bonus marks: 5 points

In [ ]:

```
#bonus code here
```

Part 2

Reference:- https://www.youtube.com/watch?v=yPVQZZmheCg (https://www.youtube.com/watch?v=yPVQZZmheCg)

Create a dataframe d1 where the columns are 'city' : ['Chandigarh', 'Delhi', 'Kanpur', 'Chennai', 'Manali'] and 'Temperature' : [15, 22, 20, 26,-2]

2*5=10 points

In [25]:

```
```

Print d1

In [26]:

```
         city  Temperature
0  Chandigarh           15
1       Delhi           22
2      Kanpur           20
3     Chennai           26
4      Manali           -2
```

What is the shape of d1.

In [27]:

```
(5, 2)
```

Set city = d1['city']

In [28]:

print city
What is the type of city.

In [29]:

```
0    Chandigarh
1         Delhi
2        Kanpur
3       Chennai
4        Manali
Name: city, dtype: object
```

Out[29]:

```
pandas.core.series.Series
```

Create another dataframe d2 where the columns are
'city' - ['Bengalaru','Coimbatore','Srirangam','Pondicherry']
'Temperature' - [24,35,36,39]

3+2+5 = 10 points

In [30]:

```
          city  Temperature
0    Bengaluru           24
1   Coimbatore           35
2    Srirangam           36
3  Pondicherry           39
```

print the shape of this dataframe

In [31]:

Out[31]:

```
(4, 2)
```

merge the two dataframes together, save it in a new dataframe named 'd3'

In [33]:

Out[33]:

	city	Temperature
0	Chandigarh	15
1	Delhi	22
2	Kanpur	20
3	Chennai	26
4	Manali	-2
0	Bengaluru	24
1	Coimbatore	35
2	Srirangam	36
3	Pondicherry	39

fun fact at high level

1. .concat() simply stacks multiple DataFrame together either vertically, or stitches horizontally after aligning on index
2. .merge() first aligns two DataFrame' selected common column(s) or index, and then pick up the remaining columns from the aligned rows of each DataFrame

Do you know about join function? This interesting quest we give to you to find.

Select the part of the dataframe such that it contains cities where temp is less then or equal to 20
How many cities are there?

5+5=10 points

In [36]:

```
```

Out[36]:

	city	Temperature
0	Chandigarh	15
2	Kanpur	20
4	Manali	-2

Select the part of the dataframe such that it contains the cities where tempearature greater than or equal to 35

In [37]:

```
```

Out[37]:

	city	Temperature
1	Coimbatore	35
2	Srirangam	36
3	Pondicherry	39

3. Applying functions to columns and creating new columns

Reference: https://www.youtube.com/watch?v=L0wMml__Gow (https://www.youtube.com/watch?v=L0wMml__Gow)

20 points

We need to create another column in d3, which contains a boolean value for each city to indicate whether it's a union territory or not.

- HINT: Chandigarh, Pondicherry and Delhi are only 3 union territories here.

In [40]:

```
# write function here
```

In [41]:

```
#print d3
```

Out[41]:

	city	Temperature	is_ut
0	Chandigarh	15	True
1	Delhi	22	True
2	Kanpur	20	False
3	Chennai	26	False
4	Manali	-2	False
0	Bengaluru	24	False
1	Coimbatore	35	False
2	Srirangam	36	False
3	Pondicherry	39	True

The temperatures mentioned in 'Temperature' column are mentioned in Celsius, we need another column which contains the same in Fahrenheit.

HINT -

- Define a function c_to_f which takes input temp in celsius and returns a value with temperature in Fahrenheit.
- To check: c_to_f(10) should return 50.

In [46]:

```
# write function here
```

In [48]:

```
# check  function c_to_f(10)
```

Out[48]:

```
50.0
```

In [50]:

```
# apply function c_to_f to d3 to create a column 'temp_farenhiet'
```

Out[50]:

	city	Temperature	is_ut	temp_farenhiet
0	Chandigarh	15	True	59.0
1	Delhi	22	True	71.6
2	Kanpur	20	False	68.0
3	Chennai	26	False	78.8
4	Manali	-2	False	28.4
0	Bengaluru	24	False	75.2
1	Coimbatore	35	False	95.0
2	Srirangam	36	False	96.8
3	Pondicherry	39	True	102.2

4. Indexing and selecting rows in DataFrame

Reference:-https://www.youtube.com/watch?v=XKFRFDgA7ZY (https://www.youtube.com/watch?v=XKFRFDgA7ZY)

20 points

Select subset of the dataframe d3 such that it contains the cities which are union territories.

In [51]:

```

```

Out[51]:

	city	Temperature	is_ut	temp_farenhiet
0	Chandigarh	15	True	59.0
1	Delhi	22	True	71.6
3	Pondicherry	39	True	102.2

Select a subset of the dataframe d3 such that it contains the cities which only have temperature above 90 Farenhiet.

In [52]:

Out[52]:

	city	Temperature	is_ut	temp_farenhiet
1	Coimbatore	35	False	95.0
2	Srirangam	36	False	96.8
3	Pondicherry	39	True	102.2

Select only the first three rows of the dataframe d3.

In [57]:

Out[57]:

	city	Temperature	is_ut	temp_farenhiet
0	Chandigarh	15	True	59.0
1	Delhi	22	True	71.6
2	Kanpur	20	False	68.0

Select all the rows and last two columns in the dataframe.

In [60]:

Out[60]:

	is_ut	temp_farenhiet
0	True	59.0
1	True	71.6
2	False	68.0
3	False	78.8
4	False	28.4
0	False	75.2
1	False	95.0
2	False	96.8
3	True	102.2

5. Reading csv file and group by

Reference: https://www.youtube.com/watch?v=ENhGz1HkzvY (https://www.youtube.com/watch?v=ENhGz1HkzvY)
https://www.youtube.com/watch?v=nulx_VmV7dE (https://www.youtube.com/watch?v=nulx_VmV7dE)

2*2=4 points

Read the file 'weather_data.csv' and store it in a pandas dataframe 'df'

In [62]:

```
```

print the shape and head of the dataframe

In [63]:

```
```

```
(6, 4)
```

Out[63]:

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow
4	1/5/2017	32	4	Rain

Group the dataframe wrt to 'event' and save the resulting output in a variable named 'g'

5*3 = 15 points

In [75]:

```
```

Print g and type(g)

In [76]:

```
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7fd3580e09d0>
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

Iterate over all the temperatures in g and print them

In [77]:

```
('Rain',         day  temperature  windspeed event
0  1/1/2017           32          6  Rain
4  1/5/2017           32          4  Rain)
('Snow',         day  temperature  windspeed event
2  1/3/2017           28          2  Snow
3  1/4/2017           24          7  Snow)
('Sunny',         day  temperature  windspeed  event
1  1/2/2017           35          7  Sunny
5  1/6/2017           31          2  Sunny)
```

Use the describle method on g to get the complete description on it.

In [78]:

Out[78]:

	temperature												
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%
event													
Rain	2.0	32.0	0.000000	32.0	32.0	32.0	32.0	32.0	2.0	5.0	1.414214	4.0	4.5(
Snow	2.0	26.0	2.828427	24.0	25.0	26.0	27.0	28.0	2.0	4.5	3.535534	2.0	3.2
Sunny	2.0	33.0	2.828427	31.0	32.0	33.0	34.0	35.0	2.0	4.5	3.535534	2.0	3.2

Create a new column in the dataframe df, named 'hot_temp' which contains True if temp is above 30, else False

5 points

In [80]:

```
#code here to filter df with above condition
```

In [81]:

```
# print df
```

Out[81]:

	day	temperature	windspeed	event	hot_temp
0	1/1/2017	32	6	Rain	True
1	1/2/2017	35	7	Sunny	True
2	1/3/2017	28	2	Snow	False
3	1/4/2017	24	7	Snow	False
4	1/5/2017	32	4	Rain	True
5	1/6/2017	31	2	Sunny	True

Data Range

Reference:-https://www.youtube.com/watch?v=dRxCvSbMEto (https://www.youtube.com/watch?v=dRxCvSbMEto)

5*3 = 15 points

Create a pandas daterange where starting date is 1st of January,2020 and end date is 1st of April 2021, store it in a new variable named 'a'

In [2]:

```
a =
```

print a

In [3]:

```

```

Out[3]:

```
DatetimeIndex(['2020-01-01', '2020-01-02', '2020-01-03', '2020-01-04',
               '2020-01-05', '2020-01-06', '2020-01-07', '2020-01-08',
               '2020-01-09', '2020-01-10',
               ...
               '2021-03-23', '2021-03-24', '2021-03-25', '2021-03-26',
               '2021-03-27', '2021-03-28', '2021-03-29', '2021-03-30',
               '2021-03-31', '2021-04-01'],
              dtype='datetime64[ns]', length=457, freq='D')
```

What is the len of a?

In [4]:

```
```

Out[4]:

457

What is the type of a?

In [87]:

```
```

Out[87]:

```
pandas.core.indexes.datetimes.DatetimeIndex
```

Hey high five! You solved the 4th milestone challenge too! Pawri to banti hai ;)

Its Feedback Time!

We hope you've enjoyed this course so far. We're committed to help you use "AI for All" course to its full potential, so that you have a great learning experience. And that's why we need your help in form of a feedback here.

Please fill this feedback form https://zfrmz.in/MtRG5oWXBdesm6rmSM7N (https://zfrmz.in/MtRG5oWXBdesm6rmSM7N)