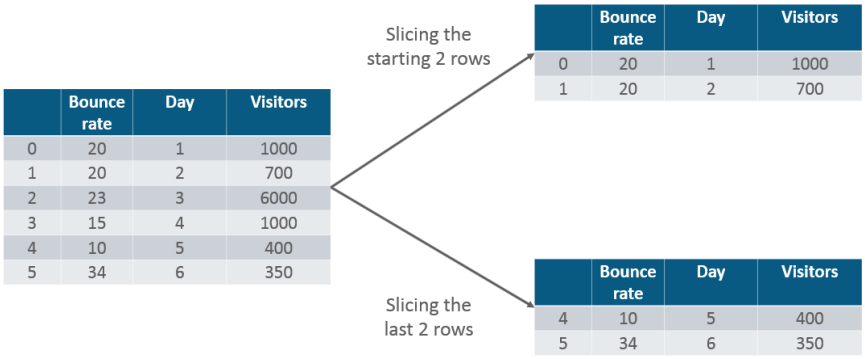
Lab Exercise 03

1. Slicing the Data Frame

Slicing the Data Frame as per the following output



	Bounce rate	Day	Visitors
0	20	1	1000
1	20	2	700
2	23	3	6000
3	15	4	1000
4	10	5	400
5	34	6	350

	Bounce rate	Day	Visitors
0	20	1	1000
1	20	2	700

	Bounce rate	Day	Visitors
4	10	5	400
5	34	6	350

2. Concatenation of two frames

a. **Concatenate following two dataframes**

```
df1 = pd.DataFrame({"HPI":[80,90,70,60],"Int_Rate":[2,1,2,3],
"IND_GDP":[50,45,45,67]}, index=[2001, 2002,2003,2004])

df2 = pd.DataFrame({"HPI":[80,90,70,60],"Int_Rate":[2,1,2,3],"IND_GDP":[50,45,45,67]},
index=[2005, 2006,2007,2008])
```

b. Concatenate following two dataframe using axis

```
df1 = pd.DataFrame({"HPI":[80,90,70,60],"Int_Rate":[2,1,2,3],
"IND_GDP":[50,45,45,67]}, index=[2001, 2002,2003,2004])

df2 = pd.DataFrame({"HPI":[80,90,70,60],"Int_Rate":[2,1,2,3],"IND_GDP":[50,45,45,67]},
index=[2005, 2006,2007,2008])
```

3. Change the index

a. **Display the following table index name look like output**

```
df= pd.DataFrame({"Day":[1,2,3,4], "Visitors":[200, 100,230,300],
"Bounce_Rate":[20,45,60,10]})
```

Output

Day	Bounce_Rate	Visitors
1	20	200
2	45	100
3	60	230
4	10	300

4. **Change the Column Headers from “Visitors” to “Users”**

```
df = pd.DataFrame({"Day":[1,2,3,4], "Visitors":[200, 100,230,300],
"Bounce_Rate":[20,45,60,10]})
```

5. **A data frame is made from the csv file and the data frame is sorted in ascending order of Names of Players.**

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
5	Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	NaN	12000000.0
6	Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2165160.0
8	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36.0	PG	22.0	6-4	220.0	Oklahoma State	3431040.0

6. Find the positions of numbers that are multiples of 3 from `ser`.

```
ser = pd.Series(np.random.randint(1, 10, 7))
```

7. How to extract items at given positions from a series?

From ser, extract the items at positions in list pos.

Input

```
ser = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))
pos = [0, 4, 8, 14, 20]
```

8. How to convert the first character of each element in a series to uppercase?
Change the first character of each word to upper case in each word of ser.

ser = pd.Series(['how', 'to', 'kick', 'ass?'])

9. How to compute difference of differences between consecutive numbers of a series?
Difference of differences between the consecutive numbers of ser.

Input

ser = pd.Series([1, 3, 6, 10, 15, 21, 27, 35])

Desired Output

[nan, 2.0, 3.0, 4.0, 5.0, 6.0, 6.0, 8.0]
[nan, nan, 1.0, 1.0, 1.0, 1.0, 0.0, 2.0]

10. How to get the day of month, week number, day of year and day of week from a series of date strings?

Get the day of month, week number, day of year and day of week from ser.

Input

ser = pd.Series(['01 Jan 2010', '02-02-2011', '20120303', '2013/04/04', '2014-05-05', '2015-06-06T12:20'])

Desired output

Date: [1, 2, 3, 4, 5, 6]
Week number: [53, 5, 9, 14, 19, 23]
Day num of year: [1, 33, 63, 94, 125, 157]
Day of week: ['Friday', 'Wednesday', 'Saturday', 'Thursday', 'Monday', 'Saturday']

11. How to filter words that contain at least 2 vowels from a series?

From ser, extract words that contain atleast 2 vowels.

Input

ser = pd.Series(['Apple', 'Orange', 'Plan', 'Python', 'Money'])
Desired Output

0 Apple
1 Orange
4 Money
dtype: object

12. How to filter valid emails from a series?

Extract the valid emails from the series emails. The regex pattern for valid emails is provided as reference.

Input

emails = pd.Series(['buying books at amazom.com', 'rameses@egypt.com', 'matt@t.co', 'narendra@modi.com'])

pattern ='[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,4}'

Desired Output

1 rameses@egypt.com

2 matt@t.co

3 narendra@modi.com

dtype: object

13. How to find all the local maxima (or peaks) in a numeric series?
Get the positions of peaks (values surrounded by smaller values on both sides) in ser.

Input

ser = pd.Series([2, 10, 3, 4, 9, 10, 2, 7, 3])
Desired output

array([1, 5, 7])

14. How to change the order of columns of a dataframe?

a) In df, interchange columns 'a' and 'c'.

b) Create a generic function to interchange two columns, without hardcoding column names.

c) Sort the columns in reverse alphabetical order, that is colume 'e' first through column 'a' last.

Input

df = pd.DataFrame(np.arange(20).reshape(-1, 5), columns=list('abcde'))

15. How to create a new column that contains the row number of nearest column by euclidean distance?

Create a new column such that, each row contains the row number of nearest row-record by euclidean distance

Input

df = pd.DataFrame(np.random.randint(1,100, 40).reshape(10, -1), columns=list('pqrs'), index=list('abcdefghij'))

df

```
#     p   q   r   s
# a  57  77  13  62
# b  68   5  92  24
# c  74  40  18  37
# d  80  17  39  60
# e  93  48  85  33
# f  69  55   8  11
# g  39  23  88  53
# h  63  28  25  61
# i  18   4  73   7
# j  79  12  45  34
```

Desired Output

```
df
#    p   q   r   s nearest_row   dist
# a  57  77  13  62           i  116.0
# b  68   5  92  24           a  114.0
# c  74  40  18  37           i   91.0
# d  80  17  39  60           i   89.0
# e  93  48  85  33           i   92.0
# f  69  55   8  11           g  100.0
# g  39  23  88  53           f  100.0
# h  63  28  25  61           i   88.0
# i  18   4  73   7           a  116.0
# j  79  12  45  34           a   81.0
```