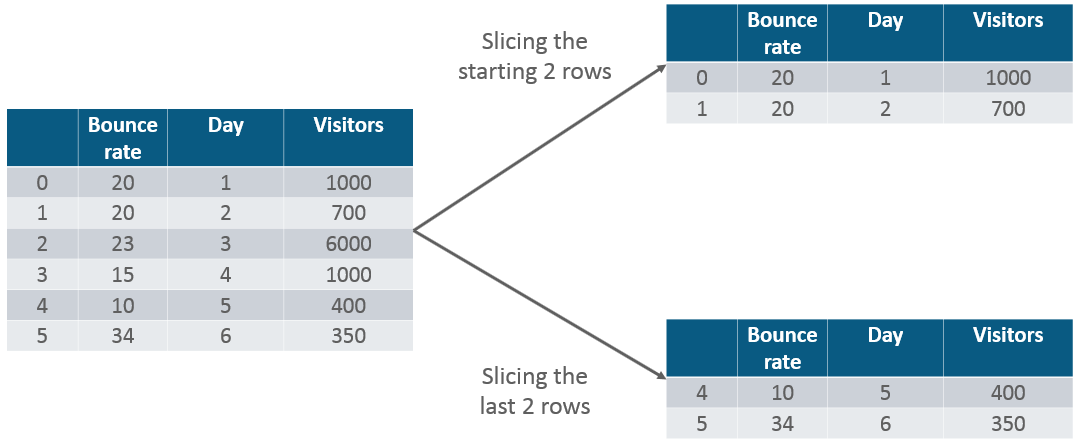
**Lab Exercise 03**

1. **Slicing the Data Frame**

Slicing the Data Frame as per the following output



### ****Concatenation of two frames****

### ****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])

### 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])

### ****Change the index****

### ****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

Bounce\_Rate  Visitors

Day

1      20           200

2      45           100

3      60           230

4      10           300

### ****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]})

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



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

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

1. 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]

1. 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?'])

1. 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]

1. 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']

1. 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

1. 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

1. 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])

1. How to change the order of columns of a dataframe?
2. In df, interchange columns 'a' and 'c'.
3. Create a generic function to interchange two columns, without hardcoding column names.
4. 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'))

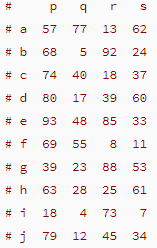
1. How to create a new column that contains the row number of nearest column by euclidean distance?

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



Desired Output

