**1. How to import pandas and check the version?**

**2. How to create a series from a list, numpy array and dict?**

Create a pandas series from each of the items below: a list, numpy and a dictionary

Input

import numpy as np

mylist = list('abcedfghijklmnopqrstuvwxyz')

myarr = np.arange(26)

mydict = dict(zip(mylist, myarr))

**3. How to convert the index of a series into a column of a dataframe?**

Difficulty Level: L1

Convert the series ser into a dataframe with its index as another column on the dataframe.

Input

mylist = list('abcedfghijklmnopqrstuvwxyz')

myarr = np.arange(26)

mydict = dict(zip(mylist, myarr))

ser = pd.Series(mydict)

**4. How to combine many series to form a dataframe?**

Difficulty Level: L1

Combine ser1 and ser2 to form a dataframe.

Input

import numpy as np

ser1 = pd.Series(list('abcedfghijklmnopqrstuvwxyz'))

ser2 = pd.Series(np.arange(26))

**5. How to assign name to the series’ index?**

Difficulty Level: L1

Give a name to the series ser calling it ‘alphabets’.

Input

ser = pd.Series(list('abcedfghijklmnopqrstuvwxyz'))

**6. How to get the items of series A not present in series B?**

Difficulty Level: L2

From ser1 remove items present in ser2.

ser1 = pd.Series([1, 2, 3, 4, 5])

ser2 = pd.Series([4, 5, 6, 7, 8])

**7. How to get the items not common to both series A and series B?**

Difficulty Level: L2

Get all items of ser1 and ser2 not common to both.

Input

ser1 = pd.Series([1, 2, 3, 4, 5])

ser2 = pd.Series([4, 5, 6, 7, 8])

**8. How to get the minimum, 25th percentile, median, 75th, and max of a numeric series?**

Difficuty Level: L2

Compute the minimum, 25th percentile, median, 75th, and maximum of ser.

Input

ser = pd.Series(np.random.normal(10, 5, 25))

**9. How to get frequency counts of unique items of a series?**

Difficulty Level: L1

Calculte the frequency counts of each unique value ser.

Input

ser = pd.Series(np.take(list('abcdefgh'), np.random.randint(8, size=30)))

**10. How to keep only top 2 most frequent values as it is and replace everything else as ‘Other’?**

Difficulty Level: L2

From ser, keep the top 2 most frequent items as it is and replace everything else as ‘Other’.

Input

np.random.RandomState(100)

ser = pd.Series(np.random.randint(1, 5, [12]))

**11. How to bin a numeric series to 10 groups of equal size?**

Difficulty Level: L2

Bin the series ser into 10 equal deciles and replace the values with the bin name.

Input

ser = pd.Series(np.random.random(20))

Desired Output

# First 5 items

0 7th

1 9th

2 7th

3 3rd

4 8th

dtype: category

Categories (10, object): [1st < 2nd < 3rd < 4th ... 7th < 8th < 9th < 10th]

**12. How to convert a numpy array to a dataframe of given shape? (L1)**

Difficulty Level: L1

Reshape the series ser into a dataframe with 7 rows and 5 columns

Input

ser = pd.Series(np.random.randint(1, 10, 35))

**13. How to find the positions of numbers that are multiples of 3 from a series?**

Difficulty Level: L2

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

Input

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

**14. How to extract items at given positions from a series**

Difficulty Level: L1

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

Input

ser = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))

pos = [0, 4, 8, 14, 20]

**15. How to stack two series vertically and horizontally ?**

Difficulty Level: L1

Stack ser1 and ser2 vertically and horizontally (to form a dataframe).

Input

ser1 = pd.Series(range(5))

ser2 = pd.Series(list('abcde'))

**16. How to get the positions of items of series A in another series B?**

Difficulty Level: L2

Get the positions of items of ser2 in ser1 as a list.

Input

ser1 = pd.Series([10, 9, 6, 5, 3, 1, 12, 8, 13])

ser2 = pd.Series([1, 3, 10, 13])

**17. How to compute the mean squared error on a truth and predicted series?**

Difficulty Level: L2

Compute the mean squared error of truth and pred series.

Input

truth = pd.Series(range(10))

pred = pd.Series(range(10)) + np.random.random(10)

**18. How to convert the first character of each element in a series to uppercase?**

Difficulty Level: L2

Change the first character of each word to upper case in each word of ser.

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

**19. How to calculate the number of characters in each word in a series?**

Difficulty Level: L2

Input

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

**20. How to compute difference of differences between consequtive numbers of a series?**

Difficulty Level: L1

Difference of differences between the consequtive 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]

**21. How to convert a series of date-strings to a timeseries?**

Difficiulty Level: L2

Input

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

Desired Output

0 2010-01-01 00:00:00

1 2011-02-02 00:00:00

2 2012-03-03 00:00:00

3 2013-04-04 00:00:00

4 2014-05-05 00:00:00

5 2015-06-06 12:20:00

dtype: datetime64[ns]

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

Difficiulty Level: L2

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']

**23. How to convert year-month string to dates corresponding to the 4th day of the month?**

Difficiulty Level: L2

Change ser to dates that start with 4th of the respective months.

Input

ser = pd.Series(['Jan 2010', 'Feb 2011', 'Mar 2012'])

Desired Output

0 2010-01-04

1 2011-02-04

2 2012-03-04

dtype: datetime64[ns]

**24. How to filter words that contain atleast 2 vowels from a series?**

Difficiulty Level: L3

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

**25. How to filter valid emails from a series?**

Difficiulty Level: L3

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](http://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

**26. How to get the mean of a series grouped by another series?**

Difficiulty Level: L2

Compute the mean of weights of each fruit.

Input

fruit = pd.Series(np.random.choice(['apple', 'banana', 'carrot'], 10))

weights = pd.Series(np.linspace(1, 10, 10))

print(weight.tolist())

print(fruit.tolist())

#> [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]

#> ['banana', 'carrot', 'apple', 'carrot', 'carrot', 'apple', 'banana', 'carrot', 'apple', 'carrot']

Desired output

# values can change due to randomness

apple 6.0

banana 4.0

carrot 5.8

dtype: float64

**27. How to compute the euclidean distance between two series?**

Difficiulty Level: L2

Compute the [euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance) between series (points) p and q, without using a packaged formula.

Input

p = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

q = pd.Series([10, 9, 8, 7, 6, 5, 4, 3, 2, 1])

Desired Output

18.165

**28. How to find all the local maxima (or peaks) in a numeric series?**

Difficiulty Level: L3

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])

**29. How to replace missing spaces in a string with the least frequent character?**

Replace the spaces in my\_str with the least frequent character.

Difficiulty Level: L2

Input

my\_str = 'dbc deb abed gade'

Desired Output

'dbccdebcabedcgade' # least frequent is 'c'

**30. How to create a TimeSeries starting ‘2000-01-01’ and 10 weekends (saturdays) after that having random numbers as values?**

Difficiulty Level: L2

Desired output

# values can be random

2000-01-01 4

2000-01-08 1

2000-01-15 8

2000-01-22 4

2000-01-29 4

2000-02-05 2

2000-02-12 4

2000-02-19 9

2000-02-26 6

2000-03-04 6

**31. How to fill an intermittent time series so all missing dates show up with values of previous non-missing date?**

Difficiulty Level: L2

ser has missing dates and values. Make all missing dates appear and fill up with value from previous date.

Input

ser = pd.Series([1,10,3,np.nan], index=pd.to\_datetime(['2000-01-01', '2000-01-03', '2000-01-06', '2000-01-08']))

print(ser)

#> 2000-01-01 1.0

#> 2000-01-03 10.0

#> 2000-01-06 3.0

#> 2000-01-08 NaN

#> dtype: float64

Desired Output

2000-01-01 1.0

2000-01-02 1.0

2000-01-03 10.0

2000-01-04 10.0

2000-01-05 10.0

2000-01-06 3.0

2000-01-07 3.0

2000-01-08 NaN

**32. How to compute the autocorrelations of a numeric series?**

Difficiulty Level: L3

Compute autocorrelations for the first 10 lags of ser. Find out which lag has the largest correlation.

Input

ser = pd.Series(np.arange(20) + np.random.normal(1, 10, 20))

Desired output

# values will change due to randomness

[0.29999999999999999, -0.11, -0.17000000000000001, 0.46000000000000002, 0.28000000000000003, -0.040000000000000001, -0.37, 0.41999999999999998, 0.47999999999999998, 0.17999999999999999]

Lag having highest correlation: 9

**33. How to import only every nth row from a csv file to create a dataframe?**

Difficiulty Level: L2

Import every 50th row of [BostonHousing dataset](https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv) as a dataframe.

**34. How to change column values when importing csv to a dataframe?**

Difficulty Level: L2

Import the [boston housing dataset](https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv), but while importing change the 'medv' (median house value) column so that values < 25 becomes ‘Low’ and > 25 becomes ‘High’.

**35. How to create a dataframe with rows as strides from a given series?**

Difficiulty Level: L3

Input

L = pd.Series(range(15))

Desired Output

array([[ 0, 1, 2, 3],

[ 2, 3, 4, 5],

[ 4, 5, 6, 7],

[ 6, 7, 8, 9],

[ 8, 9, 10, 11],

[10, 11, 12, 13]])

**36. How to import only specified columns from a csv file?**

Difficulty Level: L1

Import ‘crim’ and ‘medv’ columns of the [BostonHousing dataset](https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv) as a dataframe.

**37. How to get the n*rows, n*columns, datatype, summary stats of each column of a dataframe? Also get the array and list equivalent.**

Difficulty Level: L2

Get the number of rows, columns, datatype and summary statistics of each column of the [Cars93](https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv)dataset. Also get the numpy array and list equivalent of the dataframe.

**38. How to extract the row and column number of a particular cell with given criterion?**

Difficulty Level: L1

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

Which manufacturer, model and type has the highest Price? What is the row and column number of the cell with the highest Price value?

**39. How to rename a specific columns in a dataframe?**

Difficulty Level: L2

Rename the column Type as CarType in df and replace the ‘.’ in column names with ‘\_’.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

print(df.columns)

#> Index(['Manufacturer', 'Model', 'Type', 'Min.Price', 'Price', 'Max.Price',

#> 'MPG.city', 'MPG.highway', 'AirBags', 'DriveTrain', 'Cylinders',

#> 'EngineSize', 'Horsepower', 'RPM', 'Rev.per.mile', 'Man.trans.avail',

#> 'Fuel.tank.capacity', 'Passengers', 'Length', 'Wheelbase', 'Width',

#> 'Turn.circle', 'Rear.seat.room', 'Luggage.room', 'Weight', 'Origin',

#> 'Make'],

#> dtype='object')

Desired Solution

print(df.columns)

#> Index(['Manufacturer', 'Model', 'CarType', 'Min\_Price', 'Price', 'Max\_Price',

#> 'MPG\_city', 'MPG\_highway', 'AirBags', 'DriveTrain', 'Cylinders',

#> 'EngineSize', 'Horsepower', 'RPM', 'Rev\_per\_mile', 'Man\_trans\_avail',

#> 'Fuel\_tank\_capacity', 'Passengers', 'Length', 'Wheelbase', 'Width',

#> 'Turn\_circle', 'Rear\_seat\_room', 'Luggage\_room', 'Weight', 'Origin',

#> 'Make'],

#> dtype='object')

**40. How to check if a dataframe has any missing values?**

Difficulty Level: L1

Check if df has any missing values.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

**41. How to count the number of missing values in each column?**

Difficulty Level: L2

Count the number of missing values in each column of df. Which column has the maximum number of missing values?

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

**42. How to replace missing values of multiple numeric columns with the mean?**

Difficulty Level: L2

Replace missing values in Min.Price and Max.Price columns with their respective mean.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

**43. How to use apply function on existing columns with global variables as additional arguments?**

Difficulty Level: L3

In df, use apply method to replace the missing values in Min.Price with the column’s mean and those in Max.Price with the column’s median.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

[Use Hint from StackOverflow](https://stackoverflow.com/questions/32437435/passing-additional-arguments-to-python-pandas-dataframe-apply)

**44. How to select a specific column from a dataframe as a dataframe instead of a series?**

Difficulty Level: L2

Get the first column (a) in df as a dataframe (rather than as a Series).

Input

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

**45. How to change the order of columns of a dataframe?**

Difficulty Level: L3

Actually 3 questions.

1. In df, interchange columns 'a' and 'c'.
2. Create a generic function to interchange two columns, without hardcoding column names.
3. 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'))

**46. How to set the number of rows and columns displayed in the output?**

Difficulty Level: L2

Change the pamdas display settings on printing the dataframe df it shows a maximum of 10 rows and 10 columns.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

**47. How to format or suppress scientific notations in a pandas dataframe?**

Difficulty Level: L2

Suppress scientific notations like ‘e-03’ in df and print upto 4 numbers after decimal.

Input

df = pd.DataFrame(np.random.random(4)\*\*10, columns=['random'])

df

#> random

#> 0 3.474280e-03

#> 1 3.951517e-05

#> 2 7.469702e-02

#> 3 5.541282e-28

Desired Output

#> random

#> 0 0.0035

#> 1 0.0000

#> 2 0.0747

#> 3 0.0000

**48. How to format all the values in a dataframe as percentages?**

Difficulty Level: L2

Format the values in column 'random' of df as percentages.

Input

df = pd.DataFrame(np.random.random(4), columns=['random'])

df

#> random

#> 0 .689723

#> 1 .957224

#> 2 .159157

#> 3 .21082

Desired Output

#> random

#> 0 68.97%

#> 1 95.72%

#> 2 15.91%

#> 3 2.10%

**49. How to filter every nth row in a dataframe?**

Difficulty Level: L1

From df, filter the 'Manufacturer', 'Model' and 'Type' for every 20th row starting from 1st (row 0).

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>')

**50. How to create a primary key index by combining relevant columns?**

Difficulty Level: L2

In df, Replace NaNs with ‘missing’ in columns 'Manufacturer', 'Model' and 'Type' and create a index as a combination of these three columns and check if the index is a primary key.

Input

df = pd.read\_csv('<https://raw.githubusercontent.com/selva86/datasets/master/Cars93_miss.csv>', usecols=[0,1,2,3,5])

Desired Output

Manufacturer Model Type Min.Price Max.Price

Acura\_Integra\_Small Acura Integra Small 12.9 18.8

missing\_Legend\_Midsize missing Legend Midsize 29.2 38.7

Audi\_90\_Compact Audi 90 Compact 25.9 32.3

Audi\_100\_Midsize Audi 100 Midsize NaN 44.6

BMW\_535i\_Midsize BMW 535i Midsize NaN NaN

**51. How to get the row number of the nth largest value in a column?**

Difficulty Level: L2

Find the row position of the 5th largest value of column 'a' in df.

Input

df = pd.DataFrame(np.random.randint(1, 30, 30).reshape(10,-1), columns=list('abc'))

**52. How to find the position of the nth largest value greater than a given value?**

Difficulty Level: L2

In ser, find the position of the 2nd largest value greater than the mean.

Input

ser = pd.Series(np.random.randint(1, 100, 15))

**53. How to get the last n rows of a dataframe with row sum > 100?**

Difficulty Level: L2

Get the last two rows of df whose row sum is greater than 100.

df = pd.DataFrame(np.random.randint(10, 40, 60).reshape(-1, 4))

**54. How to find and cap outliers from a series or dataframe column?**

Difficulty Level: L2

Replace all values of ser in the lower 5%ile and greater than 95%ile with respective 5th and 95th %ile value.

Input

ser = pd.Series(np.logspace(-2, 2, 30))

**55. How to reshape a dataframe to the largest possible square after removing the negative values?**

Difficulty Level: L3

Reshape df to the largest possible square with negative values removed. Drop the smallest values if need be. The order of the positive numbers in the result should remain the same as the original.

Input

df = pd.DataFrame(np.random.randint(-20, 50, 100).reshape(10,-1))

**56. How to swap two rows of a dataframe?**

Difficulty Level: L2

Swap rows 1 and 2 in df.

Input

df = pd.DataFrame(np.arange(25).reshape(5, -1))

**57. How to reverse the rows of a dataframe?**

Difficulty Level: L2

Reverse all the rows of dataframe df.

Input

df = pd.DataFrame(np.arange(25).reshape(5, -1))

**58. How to create one-hot encodings of a categorical variable (dummy variables)?**

Difficulty Level: L2

Get one-hot encodings for column 'a' in the dataframe df and append it as columns.

Input

df = pd.DataFrame(np.arange(25).reshape(5,-1), columns=list('abcde'))

a b c d e

0 0 1 2 3 4

1 5 6 7 8 9

2 10 11 12 13 14

3 15 16 17 18 19

4 20 21 22 23 24

Output

0 5 10 15 20 b c d e

0 1 0 0 0 0 1 2 3 4

1 0 1 0 0 0 6 7 8 9

2 0 0 1 0 0 11 12 13 14

3 0 0 0 1 0 16 17 18 19

4 0 0 0 0 1 21 22 23 24

**59. Which column contains the highest number of row-wise maximum values?**

Difficulty Level: L2

Obtain the column name with the highest number of row-wise maximum’s in df.

df = pd.DataFrame(np.random.randint(1,100, 40).reshape(10, -1))

**60. 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.

Difficulty Level: L3

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

**61. How to know the maximum possible correlation value of each column against other columns?**

Difficulty Level: L2

Compute maximum possible absolute correlation value of each column against other columns in df.

Input

df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1), columns=list('pqrstuvwxy'), index=list('abcdefgh'))

**62. How to create a column containing the minimum by maximum of each row?**

Difficulty Level: L2

Compute the minimum-by-maximum for every row of df.

df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1))

**63. How to create a column that contains the penultimate value in each row?**

Difficulty Level: L2

Create a new column 'penultimate' which has the second largest value of each row of df.

Input

df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1))

**64. How to normalize all columns in a dataframe?**

Difficulty Level: L2

1. Normalize all columns of df by subtracting the column mean and divide by standard deviation.
2. Range all columns of df such that the minimum value in each column is 0 and max is 1.

Don’t use external packages like sklearn.

Input

df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1))

**65. How to compute the correlation of each row with the suceeding row?**

Difficulty Level: L2

Compute the correlation of each row of df with its succeeding row.

Input

df = pd.DataFrame(np.random.randint(1,100, 80).reshape(8, -1))

**66. How to replace both the diagonals of dataframe with 0?**

Difficulty Level: L2

Replace both values in both diagonals of df with 0.

Input

df = pd.DataFrame(np.random.randint(1,100, 100).reshape(10, -1))

df

# 0 1 2 3 4 5 6 7 8 9

# 0 11 46 26 44 11 62 18 70 68 26

# 1 87 71 52 50 81 43 83 39 3 59

# 2 47 76 93 77 73 2 2 16 14 26

# 3 64 18 74 22 16 37 60 8 66 39

# 4 10 18 39 98 25 8 32 6 3 29

# 5 29 91 27 86 23 84 28 31 97 10

# 6 37 71 70 65 4 72 82 89 12 97

# 7 65 22 97 75 17 10 43 78 12 77

# 8 47 57 96 55 17 83 61 85 26 86

# 9 76 80 28 45 77 12 67 80 7 63

Desired output

# 0 1 2 3 4 5 6 7 8 9

# 0 0 46 26 44 11 62 18 70 68 0

# 1 87 0 52 50 81 43 83 39 0 59

# 2 47 76 0 77 73 2 2 0 14 26

# 3 64 18 74 0 16 37 0 8 66 39

# 4 10 18 39 98 0 0 32 6 3 29

# 5 29 91 27 86 0 0 28 31 97 10

# 6 37 71 70 0 4 72 0 89 12 97

# 7 65 22 0 75 17 10 43 0 12 77

# 8 47 0 96 55 17 83 61 85 0 86

# 9 0 80 28 45 77 12 67 80 7 0

**67. How to get the particular group of a groupby dataframe by key?**

Difficulty Level: L2

This is a question related to understanding of grouped dataframe. From df\_grouped, get the group belonging to 'apple' as a dataframe.

Input

df = pd.DataFrame({'col1': ['apple', 'banana', 'orange'] \* 3,

'col2': np.random.rand(9),

'col3': np.random.randint(0, 15, 9)})

df\_grouped = df.groupby(['col1'])

# Input

df = pd.DataFrame({'col1': ['apple', 'banana', 'orange'] \* 3,

'col2': np.random.rand(9),

'col3': np.random.randint(0, 15, 9)})

df\_grouped = df.groupby(['col1'])

# Solution 1

df\_grouped.get\_group('apple')

# Solution 2

for i, dff in df\_grouped:

if i == 'apple':

print(dff)

col1 col2 col3

0 apple 0.673434 7

3 apple 0.182348 14

6 apple 0.050457 3

[/expand]

**68. How to get the n’th largest value of a column when grouped by another column?**

Difficulty Level: L2

In df, find the second largest value of 'taste' for 'banana'

Input

df = pd.DataFrame({'fruit': ['apple', 'banana', 'orange'] \* 3,

'rating': np.random.rand(9),

'price': np.random.randint(0, 15, 9)})

**69. How to compute grouped mean on pandas dataframe and keep the grouped column as another column (not index)?**

Difficulty Level: L1

In df, Compute the mean price of every fruit, while keeping the fruit as another column instead of an index.

Input

df = pd.DataFrame({'fruit': ['apple', 'banana', 'orange'] \* 3,

'rating': np.random.rand(9),

'price': np.random.randint(0, 15, 9)})

**70. How to join two dataframes by 2 columns so they have only the common rows?**

Difficulty Level: L2

Join dataframes df1 and df2 by ‘fruit-pazham’ and ‘weight-kilo’.

Input

df1 = pd.DataFrame({'fruit': ['apple', 'banana', 'orange'] \* 3,

'weight': ['high', 'medium', 'low'] \* 3,

'price': np.random.randint(0, 15, 9)})

df2 = pd.DataFrame({'pazham': ['apple', 'orange', 'pine'] \* 2,

'kilo': ['high', 'low'] \* 3,

'price': np.random.randint(0, 15, 6)})

**72. How to get the positions where values of two columns match?**

Difficulty Level: L2

**73. How to create lags and leads of a column in a dataframe?**

Difficulty Level: L2

Create two new columns in df, one of which is a lag1 (shift column a down by 1 row) of column ‘a’ and the other is a lead1 (shift column b up by 1 row).

Input

df = pd.DataFrame(np.random.randint(1, 100, 20).reshape(-1, 4), columns = list('abcd'))

a b c d

0 66 34 76 47

1 20 86 10 81

2 75 73 51 28

3 1 1 9 83

4 30 47 67 4

Desired Output

a b c d a\_lag1 b\_lead1

0 66 34 76 47 NaN 86.0

1 20 86 10 81 66.0 73.0

2 75 73 51 28 20.0 1.0

3 1 1 9 83 75.0 47.0

4 30 47 67 4 1.0 NaN

**74. How to get the frequency of unique values in the entire dataframe?**

Difficulty Level: L2

Get the frequency of unique values in the entire dataframe df.

Input

df = pd.DataFrame(np.random.randint(1, 10, 20).reshape(-1, 4), columns = list('abcd'))

**75. How to split a text column into two separate columns?**

Difficulty Level: L2

Split the string column in df to form a dataframe with 3 columns as shown.

Input

df = pd.DataFrame(["STD, City State",

"33, Kolkata West Bengal",

"44, Chennai Tamil Nadu",

"40, Hyderabad Telengana",

"80, Bangalore Karnataka"], columns=['row'])

print(df)

#> row

#> 0 STD, City\tState

#> 1 33, Kolkata\tWest Bengal

#> 2 44, Chennai\tTamil Nadu

#> 3 40, Hyderabad\tTelengana

#> 4 80, Bangalore\tKarnataka

Desired Output

0 STD City State

1 33 Kolkata West Bengal

2 44 Chennai Tamil Nadu

3 40 Hyderabad Telengana

4 80 Bangalore Karnataka