**Pandas Library**

**Q1. Import pandas under the alias pd**

**CODE :**

import pandas as pd

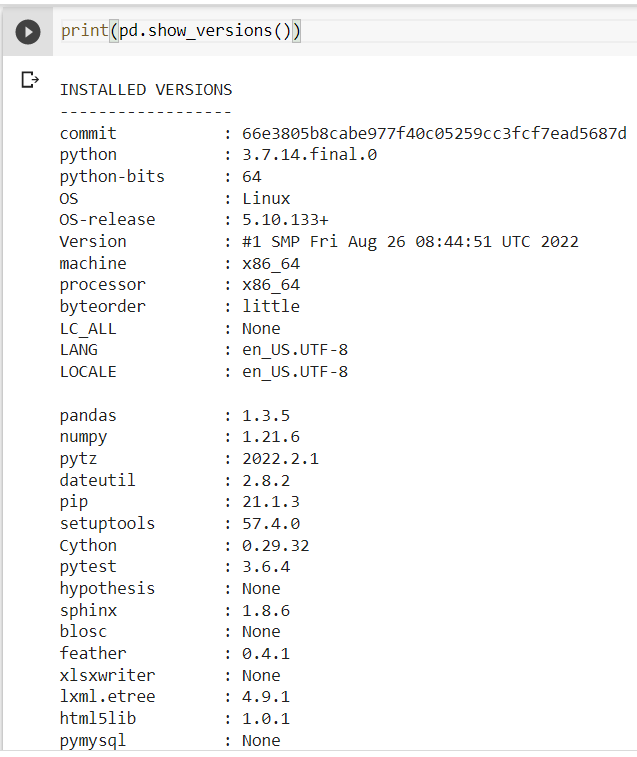
**Q2. Print the version of pandas**

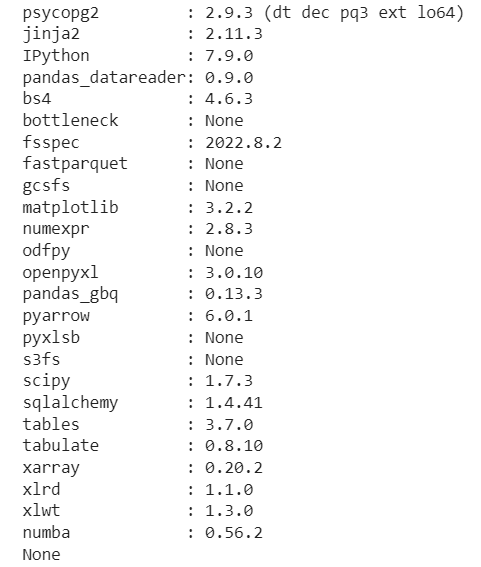
**Q3. Print out all the version information of the libraries that are required by the panda's library**

**CODE :**

print(pd.show\_versions())

**OUTPUT :**





**Q4. Create a DataFrame df from this dictionary data which has the index labels**

**CODE :**

import numpy as np

data = {'animal': ['cat', 'cat', 'snake', 'dog', 'dog', 'cat', 'snake', 'cat', 'dog', 'dog'],

'age': [2.5, 3, 0.5, np.nan, 5, 2, 4.5, np.nan, 7, 3],

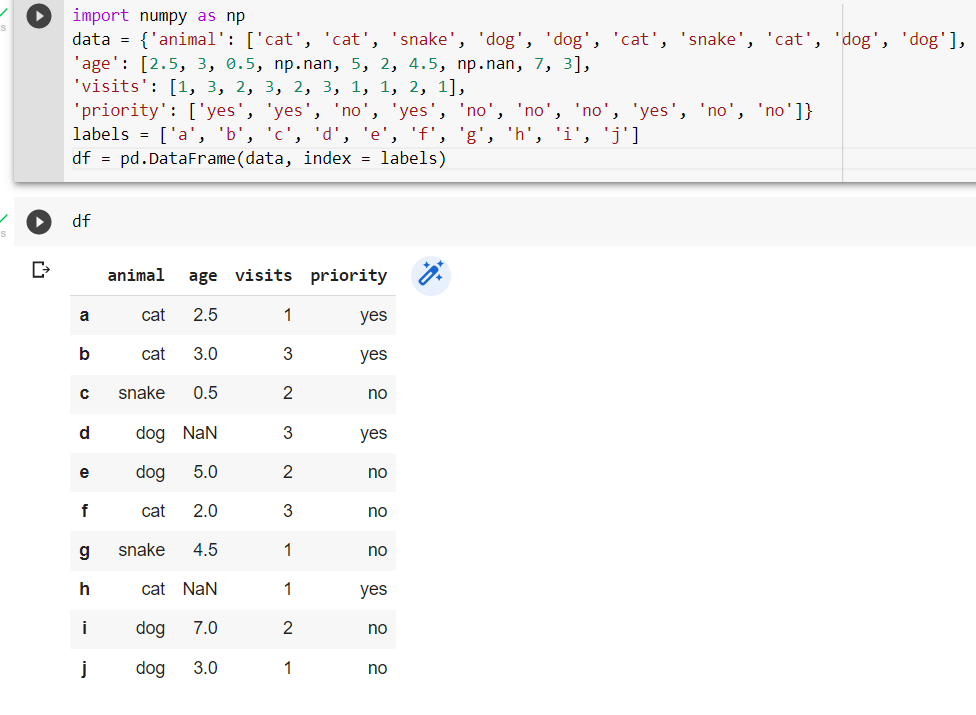
'visits': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

'priority': ['yes', 'yes', 'no', 'yes', 'no', 'no', 'no', 'yes', 'no', 'no']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(data, index = labels)

**OUTPUT :**

****

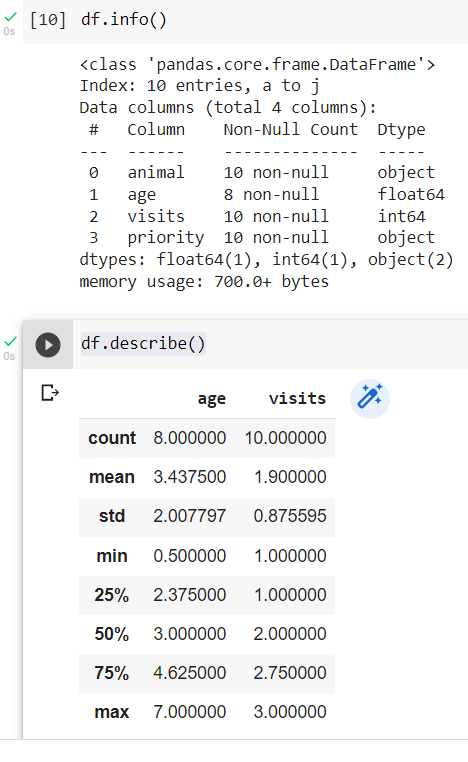
**Q5. Display a summary of the basic information about this DataFrame and its data**

**CODE :**

df.info()

df.describe()

**OUTPUT :**

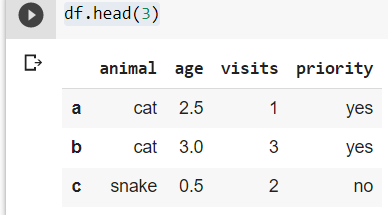
****

**Q6. Return the first 3 rows of the DataFrame df.**

**CODE :**

df.head(3)

**OUTPUT :**

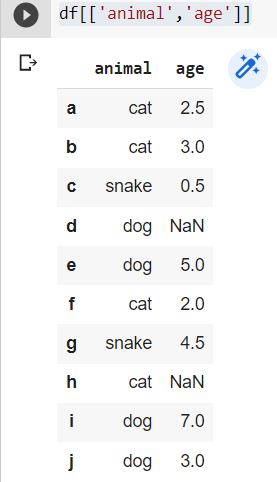
****

**Q7.Select just the 'animal' and 'age' columns from the DataFrame df**

**CODE :**

df[['animal','age']]

**OUTPUT :**

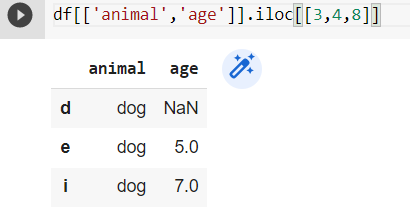
****

**Q8. Select the data in rows [3, 4, 8] and in columns ['animal', 'age'].**

**CODE :**

df[['animal','age']].iloc[[3,4,8]]

**OUTPUT :**

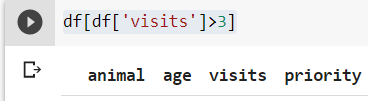
****

**Q9. Select only the rows where the number of visits is greater than 3.**

**CODE :**

df[df['visits']>3]

**OUTPUT :**

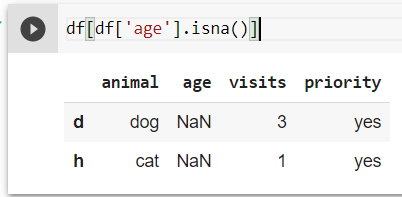
****

**Q10. Select the rows where the age is missing, i.e. it is NaN**

**CODE :**

df[df['age'].isna()]

**OUTPUT :**

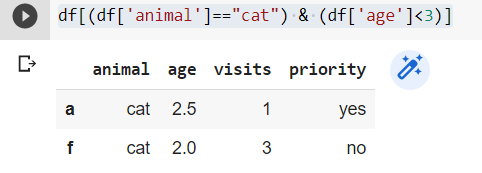
****

**Q11. Select the rows where the animal is a cat and the age is less than 3.**

**CODE :**

df[(df['animal']=="cat") & (df['age']<3)]

**OUTPUT :**

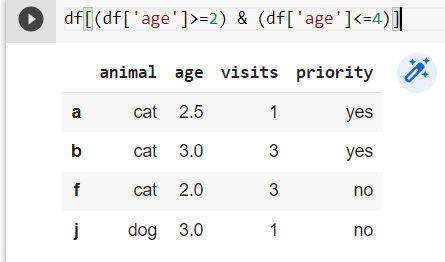
****

**Q12. Select the rows the age is between 2 and 4 (inclusive)**

**CODE :**

df[(df['age']>=2) & (df['age']<=4)]

**OUTPUT:**

****

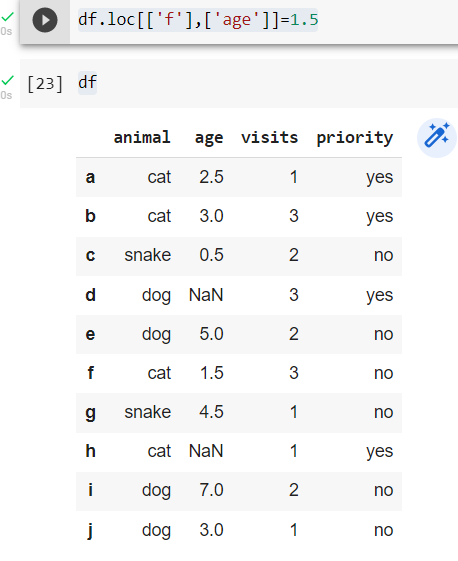
**Q13. Change the age in row 'f' to 1.5**

**CODE ;**

df.loc[['f'],['age']]=1.5

df

**OUTPUT :**

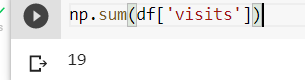
****

**Q14. Calculate the sum of all visits in df (i.e. find the total number of visits)**

**CODE :**

np.sum(df['visits'])

**OUTPUT :**

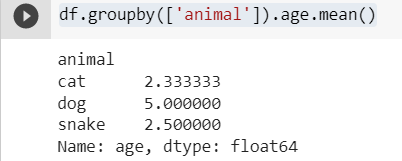
****

**Q15. Calculate the mean age for each different animal in df.**

**CODE :**

df.groupby(['animal']).age.mean()

**OUTPUT :**

****

**Q16. Append a new row 'k' to df with your choice of values for each column.**

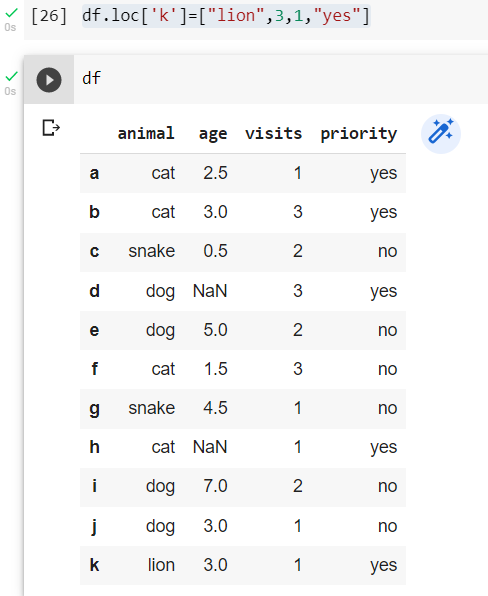
**#Then delete that row to return the original DataFrame**

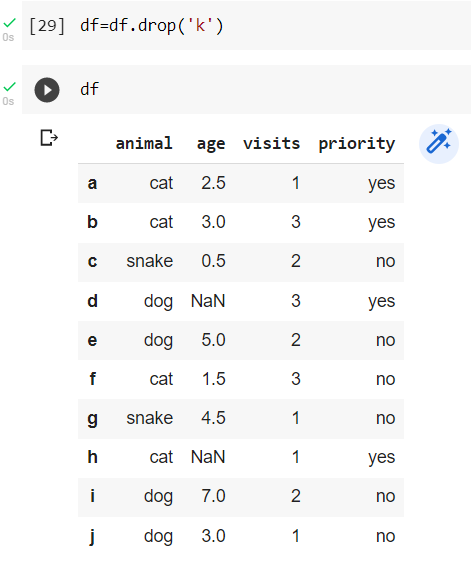
**CODE :**

df.loc['k']=["lion",3,1,"yes"]

df

**OUTPUT :**

****

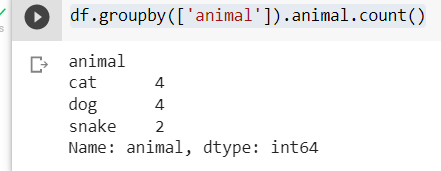
****

**Q17. Count the number of each type of animal in df.**

**CODE :**

df.groupby(['animal']).animal.count()

**OUTPUT :**

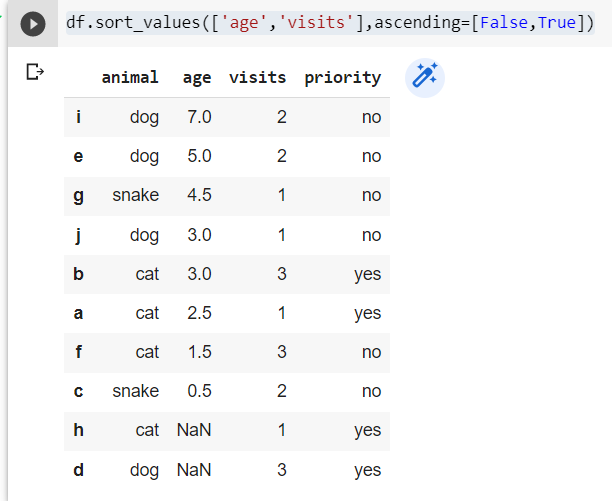
****

**Q18. Sort df first by the values in the 'age' in decending order, then by the value in the 'visits' column in ascending order**

**CODE :**

df.sort\_values(['age','visits'],ascending=[False,True])

**OUTPUT :**

****

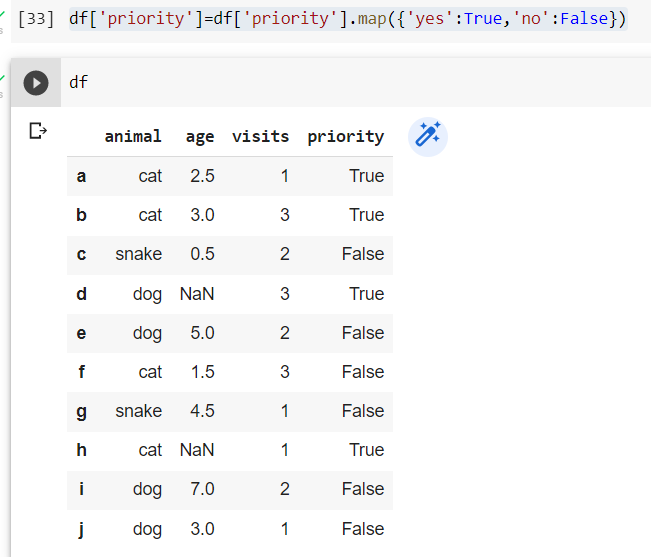
**Q19. The 'priority' column contains the values 'yes' and 'no'. Replace this column with a column of boolean values: 'yes' should be True and 'no' should be False**

**CODE :**

df['priority']=df['priority'].map({'yes':True,'no':False})

df

**OUTPUT :**

****

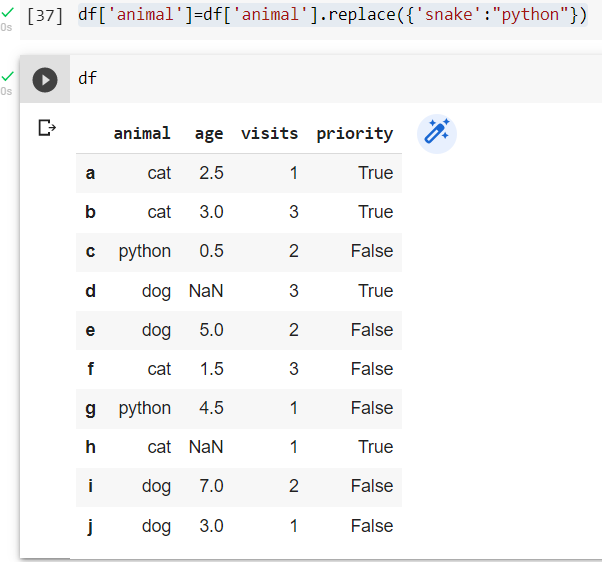
**Q20. In the 'animal' column, change the 'snake' entries to 'python'.**

**CODE :**

df['animal']=df['animal'].replace({'snake':"python"})

df

**OUTPU ;**

****

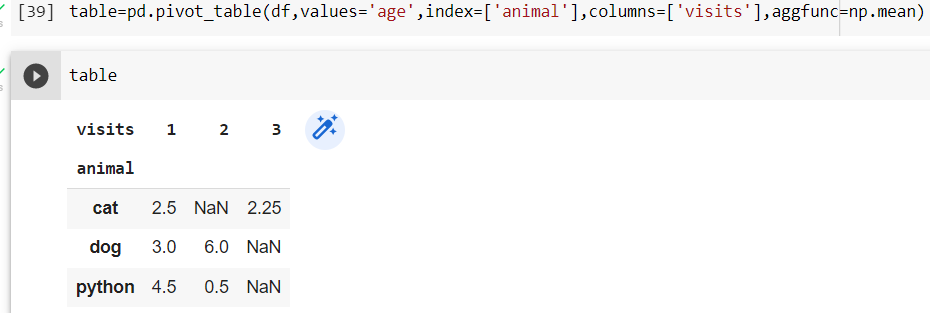
**Q21.For each animal type and each number of visits, find the mean age. In other words, each row is an animal, each column is a number of visits and the values are the mean ages (hint: use a pivot table).**

**CODE :**

table=pd.pivot\_table(df,values='age',index=['animal'],columns=['visits'],aggfunc=np.mean)

table

**OUTPUT :**

****

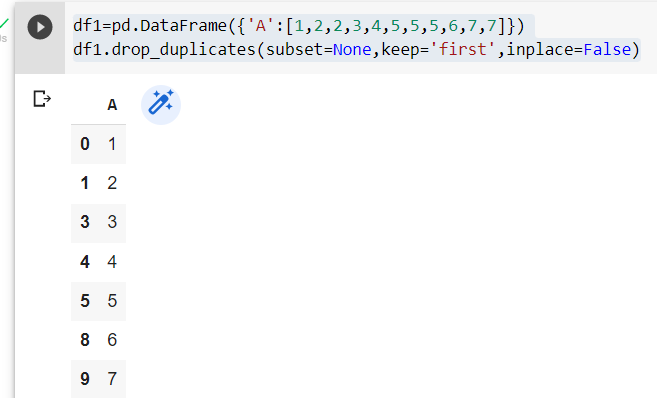
**Q22. You have a Data Frame df with a column 'A' of integers. #How do you filter out rows which contain the same integer as the row immediately above?**

**CODE :**

df1=pd.DataFrame({'A':[1,2,2,3,4,5,5,5,6,7,7]})

df1.drop\_duplicates(subset=None,keep='first',inplace=False)

**OUTPUT ;**

**[**

**Q23. Given a DataFrame of numeric values, say df = pd.DataFrame(np.random.random(size=(5, 3))) # a 5x3 frame of float values how do you subtract the row mean from each element in the row?**

**CODE :**

**OUTPUT :**

**Q24. Suppose you have Data Frame with 10 columns of real numbers.**

**Which column of #numbers has the smallest sum? Return that column's label**

**Q25. How do you count how many unique rows a Data Frame has (i.e. ignore all rows that are**

**#duplicates)? As input, use a Data Frame of zeros and ones with 10 rows and 3 columns**

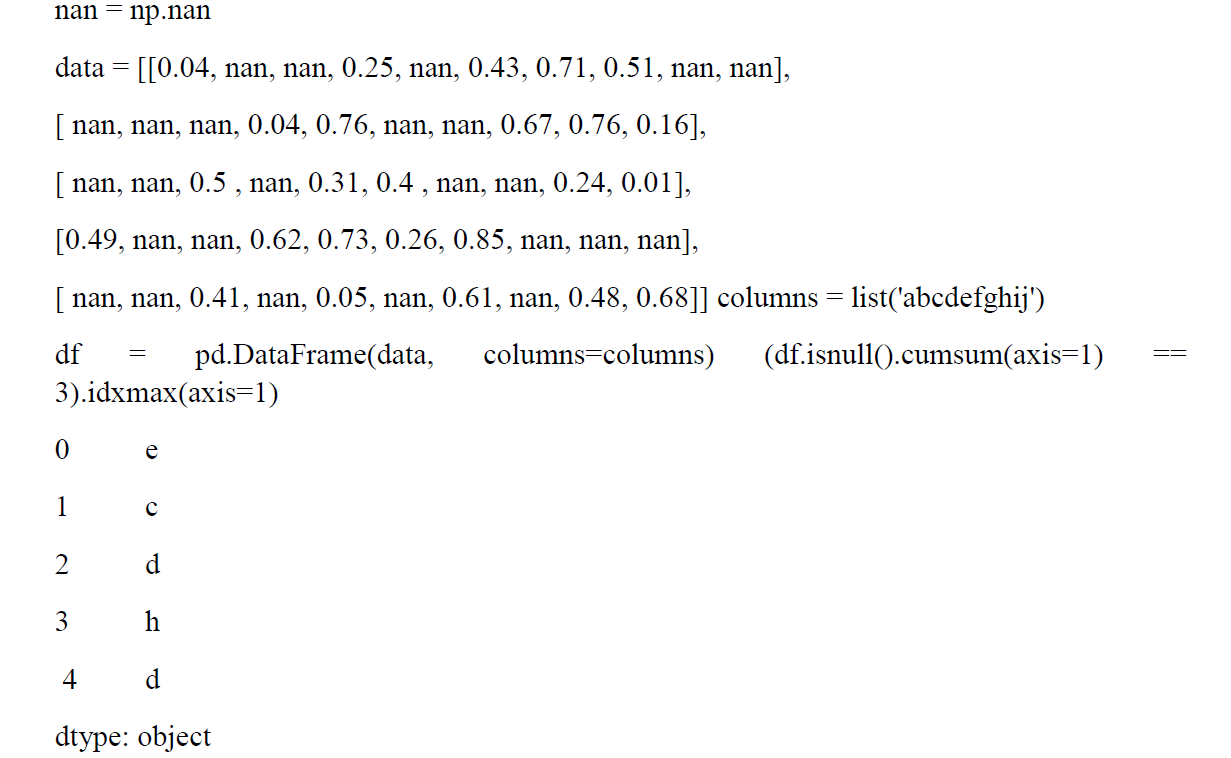
**Q26. In the cell below, you have a Data Frame df that consists of 10 columns of floating-point**

**#numbers. Exactly 5 entries in each row are NaN values.**

**#For each row of the Data Frame, find the column which contains the third NaN value.**

**#You should return a Series of column labels: e, c, d, h, d**

**Example:**

****

**Q27.A Data Frame has a column of groups 'grps' and column of integer values 'vals':**

**#For each group, find the sum of the three greatest values.**

**Q28. The data Frame df constructed below has two integer columns 'A' and 'B'. The values in 'A' are between 1 and 100(inclusive).**

**#for each group of 10 consecutive integers in 'A' (i.e.'(0,101] (1,201],...)calculate the sum of corresponding values in column'B'.**