Q1 1.REad Data-We can read data in pandas data frame as read_csv(). Two most used data read formats are csv and excel. 2.Head and Tail:-To see the data frame we can use df.head(). Head returns the first rows, if no input is given it will always show above 5 rows. In contrast to see below rows, we can use df.tail(). 3.Shape, Size and Info:-Two most basic functions after reading data is to know the number of rows and columns, and to know the datatype of variables. We can use df.shape, it gives a total number of rows and then columns. df.size() returns the number of rows times number of columns in the data frame. We can also use df.info(), from that we get different information such as rows from RangeIndex, Data columns and then data type of each column. It also includes the information of non-null counts. 4.isna():-But, if one needs to get the total number of null values in a data, we can use df.isna() as below. Sum will give the total null values. If we want just one variable null values, we can also get it by giving the name of the variable as below. 5.Describe():-Then to understand basic statistics of variables we can use df.describe(). It will give you count, mean, standard deviation, and also 5 number summary.

In [1]:
```
#EX
import pandas as pd
```

In [2]:
```
df=pd.read_csv("penguins.csv")
```

In [3]:
```
df.head(2)
```

Out[3]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	ye
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	20
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	20

In [4]:
```
df.tail(2)
```

Out[4]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	y
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female	2

In [5]:
```
df.describe()
```

Out[5]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
count	342.000000	342.000000	342.000000	342.000000	344.000000
mean	43.921930	17.151170	200.915205	4201.754386	2008.029070
std	5.459584	1.974793	14.061714	801.954536	0.818356
min	32.100000	13.100000	172.000000	2700.000000	2007.000000
25%	39.225000	15.600000	190.000000	3550.000000	2007.000000
50%	44.450000	17.300000	197.000000	4050.000000	2008.000000
75%	48.500000	18.700000	213.000000	4750.000000	2009.000000
max	59.600000	21.500000	231.000000	6300.000000	2009.000000

In [7]:
```
df.isna().sum()
```

Out[7]:
```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
year                 0
dtype: int64
```

In [8]:
```
#Q2

data={'a':[1,2,3,4],
      'b':[4,5,6,7],
      'c':["sudh","krish","hitesh","navin"]
     }
df=pd.DataFrame(data,index=['a','b','c','d'])
```

In [9]:
```
df
```

Out[9]:

	a	b	c
a	1	4	sudh
b	2	5	krish
c	3	6	hitesh
d	4	7	navin

In [10]:
```
df.reindex(['b','c','d','a'])
```

Out[10]:

	a	b	c
b	2	5	krish
c	3	6	hitesh
d	4	7	navin
a	1	4	sudh

In [39]:
```
#Q3

data={'a':[1,2,3,4],
      'b':[4,5,6,7],
      'c':[10,20,30,0]
     }
df=pd.DataFrame(data,index=['a','b','c','d'])
```

In [40]:
```
df
```

Out[40]:

	a	b	c
a	1	4	10
b	2	5	20
c	3	6	30
d	4	7	0

In [41]:
```
def test(x):
    return x.sum()
df.apply(test,axis=0)
```

Out[41]:
```
a    10
b    22
c    60
dtype: int64
```

In [43]:
```
#Q4

data={'a':[1,2,3,4],
      'b':[4,5,6,7],
      'c':['sudh','krish','yash','raj']
     }
df=pd.DataFrame(data,index=['a','b','c','d'])
```

In [45]:
```
df['word']=df['c'].apply(len)
```

In [46]:
```
df['word']
```

Out[46]:
```
a    4
b    5
c    4
d    3
Name: word, dtype: int64
```

In [ ]:
```
#Q5
#size : Size and shape of a dataframe in pandas python: Size of a dataframe is the
#Shape of a dataframe gets the number of rows and number of columns of the datafram
```

In [ ]:
```
#Q6

#Pandas read_excel() function is used for reading the Excel files.
#Files can be imported using a URL link or can be directly imported from the disk.
```

In [56]:
```
#Q7

df = pd.DataFrame({'email':['kkk@gmail.com','aa@yahoo.com']})

df['domain'] = df['email'].str.split('@').str[1]
#faster solution if no NaNs values
#df['domain'] = [x.split('@')[1] for x in df['email']]
print (df)
```

```
           email     domain
0  kkk@gmail.com  gmail.com
1   aa@yahoo.com  yahoo.com
```

In [11]:
```
#Q8
df=pd.DataFrame({'A':[3,8,6,2,9],
                 'B':[5,2,9,3,1],
                 'C':[1,7,4,5,2]
                })
```

In [28]:
```
df=pd.DataFrame(df)
```

In [29]:
```
df.loc[1:5:3,['A','B','C']]
```

Out[29]:

	A	B	C
1	8	2	7
4	9	1	2

In [30]:
```
#Q9
df=pd.DataFrame({'A':[3,8,6,2,9],
                 'B':[5,2,9,3,1],
                 'C':[1,7,4,5,2]
                })
```

In [41]:
```
df.agg(['mean','median','std'])
```

Out[41]:

	A	B	C
mean	5.60000	4.000000	3.800000
median	6.00000	3.000000	4.000000
std	3.04959	3.162278	2.387467

In [22]:
```
#Q10

df=pd.DataFrame({'Sales':[23,34,45,25,20,23,34,56,89,90,65,25,85,
                          45,56,88,78,99,56]
                 })
```

In [32]:
```
pd.DataFrame(df)
```

Out[32]:

	Sales
0	23
1	34
2	45
3	25
4	20
5	23
6	34
7	56
8	89
9	90
10	65
11	25
12	85
13	45
14	56
15	88
16	78
17	99
18	56

In [33]:
```
df['Sales'].rolling(window=7).mean()
```

Out[33]:
```
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
5          NaN
6    29.142857
7    33.857143
8    41.714286
9    48.142857
10   53.857143
11   54.571429
12   63.428571
13   65.000000
14   65.000000
15   64.857143
16   63.142857
17   68.000000
18   72.428571
Name: Sales, dtype: float64
```

In [41]:
```
#Q11


import pandas as pd


df = pd.DataFrame({'my_dates':['2023-01-01','2023-01-02','2023-01-03','2023-01-04',
 '2023-01-05']})
df['my_dates'] = pd.to_datetime(df['my_dates'])

df['day_of_week'] = df['my_dates'].dt.day_name()
```

In [42]:
```
df
```

Out[42]:

	my_dates	day_of_week
0	2023-01-01	Sunday
1	2023-01-02	Monday
2	2023-01-03	Tuesday
3	2023-01-04	Wednesday
4	2023-01-05	Thursday

In [ ]:
```

```