Importing pandas

```
import pandas as pd
```

Importing Zoo.csv and .head() will return 5 row from the top by default

```
dataset=pd.read_csv("/content/zoo.csv")
print("18012021034_ParmarSahilkhan")
dataset.head()
```

```
18012021034_ParmarSahilkhan
```

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320

.tail()will return 5 row from the bottom

```
print("18012021034_ParmarSahilkhan")
dataset.tail()
```

```
18012021034_ParmarSahilkhan
```

	animal	uniq_id	water_need
17	lion	1018	500
[illegible]	[illegible]	[illegible]	390
[illegible]	[illegible]	[illegible]	410
20	kangaroo	1021	430
21	kangaroo	1022	410

Saving… ×

Creating The Dataframe

```
dataframe1 = pd.DataFrame({'animal': ['elephant', 'tiger','lion','zebra','kangaroo'],
'type': ['Veg','Non-veg','Non-veg','veg','vegan']})
```

```
print("18012021034_ParmarSahilkhan")
dataframe1.head()
```

```
18012021034_ParmarSahilkhan
     animal     type
0  elephant      Veg
1     tiger  Non-veg
2      lion  Non-veg
3     zebra      veg
```

merge is used to merge the data frame

```
dataframe1 = dataframe1.merge(dataset)
```

```
print("18012021034_ParmarSahilkhan")
dataframe1.head()
```

```
18012021034_ParmarSahilkhan
     animal     type  uniq_id  water_need
0  elephant      Veg     1001         500
1  elephant      Veg     1002         600
2  elephant      Veg     1003         550
3     tiger  Non-veg     1004         300
4     tiger  Non-veg     1005         320
```

```
print("18012021034_ParmarSahilkhan")
dataframe1.tail()
```

```
18012021034_ParmarSahilkhan
       animal   type  uniq_id  water_need
                                      100
                                       80
19  kangaroo   vegan     1020         410
20  kangaroo   vegan     1021         430
21  kangaroo   vegan     1022         410
```

Saving… ×

```
print("18012021034_ParmarSahilkhan")
dataframe1.head(10)
```

```
18012021034_ParmarSahilkhan
```

	animal	type	uniq_id	water_need
0	elephant	Veg	1001	500
1	elephant	Veg	1002	600
2	elephant	Veg	1003	550
3	tiger	Non-veg	1004	300
4	tiger	Non-veg	1005	320
5	tiger	Non-veg	1006	330
7	tiger	Non-veg	1008	310

for reading excel file we need to use read excel method

```
dataset1=pd.read_excel('/content/Covid cases in India.xlsx')
```

```
dataset1=pd.read_excel('/content/Covid cases in India.xlsx',sheet_name='Sheet2')
```

```
print("18012021034_ParmarSahilkhan")
dataset1.info()
```

```
18012021034_ParmarSahilkhan
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 13 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   S. No.                                      16 non-null     object 
 1   Name of State / UT                          15 non-null     object 
 2   Total Confirmed cases (Indian National)     16 non-null     float64
 3   Total Confirmed cases ( Foreign National )  16 non-null     float64
 4   Cured/                                      17 non-null     object 
 5   Death                                       16 non-null     float64
 6   Unnamed: 6                                  0 non-null      float64
 7   S. No..1                                    20 non-null     float64
                                                 20 non-null     object 
                                          dian National).1       20 non-null     float64
     Total Confirmed cases ( Foreign National ).1 20 non-null     float64
 11  Cured/.1                                    21 non-null     object 
 12  Death.1                                     20 non-null     float64
dtypes: float64(8), object(5)
memory usage: 2.3+ KB
```

Saving... ×

```
print("18012021034_ParmarSahilkhan")
dataset1.head()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured/	Death	Unnamed: 6	S. No..1	Name State UT
0	NaN	NaN	NaN	NaN	Discharged	NaN	NaN	NaN	N

```
print("18012021034_ParmarSahilkhan")
dataset1.tail()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured/	Death	Unnamed: 6	S. No..1	Name o State UT.
16	Total number of confirmed cases in India	NaN	104.0	22.0	13	3.0	NaN	16.0	Unio Territory c Jammu an Kashm
17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	17.0	Unio Territory c Ladak
18	NaN	NaN	NaN	NaN	NaN	NaN	NaN	18.0	Utta Prades
19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	19.0	Uttarakhan

```
print("18012021034_ParmarSahilkhan")
```

Saving… ✕

```
_
```

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured/	Death	Unnamed: 6	S. No..1	Name of State / UT.1	C Nat
3	3	Haryana	1.0	14.0	0	0.0	NaN	3.0	Delhi	
4	4	Karnataka	8.0	0.0	0	1.0	NaN	4.0	Gujarat	
5	5	Kerala	22.0	2.0	3	0.0	NaN	5.0	Haryana	

```
dataset1=pd.read_excel('/content/Covid cases in India.xlsx',sheet_name='Sheet2',index_col=
```

```
print("18012021034_ParmarSahilkhan")
dataset1
```

```
dataset1
```

Saving… ×

S. No.	Name of State / UT	Confirmed cases (Indian National)	cases (Foreign National)	Cured/	Death	Unnamed: 6	S. No..1

columns will return a list, which is containing the name of all columns

```
print("18012021034_ParmarSahilkhan")
columns = dataset1.columns
columns
```

```
18012021034_ParmarSahilkhan
Index(['S. No.', 'Name of State / UT',
       'Total Confirmed cases (Indian National)',
       'Total Confirmed cases ( Foreign National )', 'Cured/', 'Death',
       'Unnamed: 6', 'S. No..1', 'Name of State / UT.1',
       'Total Confirmed cases (Indian National).1',
       'Total Confirmed cases ( Foreign National ).1', 'Cured/.1', 'Death.1'],
      dtype='object')
```

8 8 Punjab 1.0 0.0 0 0.0 NaN 8.0

we are doing the slicing of columns,just like string

```
newds = dataset1[dataset1.columns[3:6]]
```

```
print("18012021034_ParmarSahilkhan")
newds
```

Saving…

```
18012021034_ParmarSahilkhan
```

	Total Confirmed cases (Foreign National)	Cured/	Death
0	NaN	Discharged	NaN
1	0.0	0	0.0
2	0.0	2	1.0
3	14.0	0	0.0

```
df = pd.read_excel('/content/Covid cases in India.xlsx')
```

```
print("18012021034_ParmarSahilkhan")
df.head(6)
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	TotalConfirmedCases	Active	Recovered	Deaths
0	1.0	Maharashtra	20228	15649	3800	779
1	2.0	Gujarat	7797	5234	2091	472
2	3.0	Delhi	6542	4454	2020	68
3	4.0	Tamil Nadu	6535	4667	1824	44
4	5.0	Rajasthan	3741	1458	2176	107
5	6.0	Madhya Pradesh	3457	1766	1480	211

```
print("18012021034_ParmarSahilkhan")
new_df = df[df.columns[2:]]
new_df.head(3)
```

```
18012021034_ParmarSahilkhan
```

	TotalConfirmedCases	Active	Recovered	Deaths
0	20228	15649	3800	779
		4	2091	472
2	6542	4454	2020	68

Saving... ×

```
print("18012021034_ParmarSahilkhan")
new_df = df[0:10]
new_df
```

```
18012021034_ParmarSahilkhan
   S. No.  Name of State / UT  TotalConfirmedCases  Active  Recovered  Deaths
0     1.0         Maharashtra                20228   15649       3800     779
1     2.0             Gujarat                 7797    5234       2091     472
2     3.0               Delhi                 6542    4454       2020      68
3     4.0          Tamil Nadu                 6535    4667       1824      44
```

```
print("18012021034_ParmarSahilkhan")
newcol = ['Active','Deaths','Recovered']
new_df=df[newcol]
new_df.head(7)
```

```
18012021034_ParmarSahilkhan
   Active  Deaths  Recovered
0   15649     779       3800
1    5234     472       2091
2    4454      68       2020
3    4667      44       1824
4    1458     107       2176
5    1766     211       1480
6    1800      74       1499
```

.iloc meanse we can access the data frame using the name of the column

```
print("18012021034_ParmarSahilkhan")
df.loc[(df['Active']<100) & (df['Active']>3), ['Deaths','Active','Name of State / UT']]
```

Saving… ×

```
                              ate / UT
15      4     17              Kerala
18      3     75           Jharkhand
20      1     20         Uttarakhand
21      1     26               Assam
22      0     16        Chhattisgarh
23      3     11    Himachal Pradesh
24      0     24              Ladakh
```

here .iloc meanse we can access the dataframe using the indexing of the dataframe

```
print("18012021034_ParmarSahilkhan")
```

```
df.iloc[(df['Active']>1000).values & (df['Deaths']>100).values, [0,1,2]]
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	TotalConfirmedCases
0	1.0	Maharashtra	20228
1	2.0	Gujarat	7797
4	5.0	Rajasthan	3741
5	6.0	Madhya Pradesh	3457
8	9.0	West Bengal	1786
33	NaN	Total	62916

```
print("18012021034_ParmarSahilkhan")
df.iloc[(df['Active']>1000).values & (df['Deaths']>100).values]
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	TotalConfirmedCases	Active	Recovered	Deaths
0	1.0	Maharashtra	20228	15649	3800	779
1	2.0	Gujarat	7797	5234	2091	472
4	5.0	Rajasthan	3741	1458	2176	107
5	6.0	Madhya Pradesh	3457	1766	1480	211
8	9.0	West Bengal	1786	1243	372	171
33	NaN	Total	62916	41495	19315	2102

.nlergest will return largest value from the give column

```
print("18012021034_ParmarSahilkhan")
df.nlargest(9,'Recovered')
```

Saving… ×

	S. No.	Name of State / UT	TotalConfirmedCases	Active	Recovered	Deaths
33	NaN	Total	62916	41495	19315	2102
0	1.0	Maharashtra	20228	15649	3800	779
4	5.0	Rajasthan	3741	1458	2176	107
1	2.0	Gujarat	7797	5234	2091	472
2	3.0	Delhi	6542	4454	2020	68
3	4.0	Tamil Nadu	6535	4667	1824	44
6	7.0	Uttar Pradesh	3373	1800	1499	74
5	6.0	Madhya Pradesh	3457	1766	1480	211
7	8.0	Andhra Pradesh	1930	999	887	44

```
print("18012021034_ParmarSahilkhan")
mylist=list(df.nsmallest(5,'Recovered')['Name of State / UT'])
mylist
```

```
18012021034_ParmarSahilkhan
['Dadra and Nagar Haveli and Daman and Diu',
 'Mizoram',
 'Arunachal Pradesh',
 'Tripura',
 'Manipur']
```

.Frac is usefule to generate the samples from dataset

```
print("18012021034_ParmarSahilkhan")
df.sample(frac=.5)
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	TotalConfirmedCases	Active	Recovered	Deaths
6	7.0	Uttar Pradesh	3373	1800	1499	74
4	5.0	Rajasthan	3741	1458	2176	107
5	6.0	Madhya Pradesh	3457	1766	1480	211
20	21.0	Uttarakhand	67	20	46	1
13	14.0	Haryana	675	376	290	9
9	10.0	Punjab	1762	1574	157	31
19	20.0	Tripura	135	133	2	0
7	8.0	Andhra Pradesh	1930	999	887	44
11	12.0	Jammu and Kashmir	836	459	368	9
[illegible]	[illegible]	Dadra and Nagar Haveli and Daman and Diu	1	1	0	0
27	28.0	Puducherry	10	2	8	0
25	26.0	Andaman and Nicobar Islands	33	0	33	0
24	25.0	Ladakh	42	24	18	0
26	27.0	Meghalaya	13	2	10	1
18	19.0	Jharkhand	156	75	78	3

Saving… ×

```
print("18012021034_ParmarSahilkhan")
df.info()
```

```
18012021034_ParmarSahilkhan
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   S. No.               33 non-null     float64
 1   Name of State / UT   34 non-null     object 
 2   TotalConfirmedCases  34 non-null     int64  
 3   Active               34 non-null     int64  
 4   Recovered            34 non-null     int64  
 5   Deaths               34 non-null     int64  
dtypes: float64(1), int64(4), object(1)
memory usage: 1.7+ KB
```

.describe is used to describe the statiticaL valuve of the data frame

```
print("18012021034_ParmarSahilkhan")
df.describe()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	TotalConfirmedCases	Active	Recovered	Deaths
count	33.00000	34.000000	34.000000	34.000000	34.000000
mean	17.00000	3700.941176	2440.882353	1136.176471	123.647059
std	9.66954	11145.782912	7482.373233	3333.979190	382.590760
min	1.00000	1.000000	0.000000	0.000000	0.000000
25%	9.00000	44.500000	12.250000	19.500000	0.000000
50%	17.00000	429.000000	212.000000	117.500000	3.500000
75%	25.00000	1894.000000	1404.250000	853.000000	44.000000
max	33.00000	62916.000000	41495.000000	19315.000000	2102.000000

```
print("18012021034_ParmarSahilkhan")
df['totalcase'] = df['Active']+df['Deaths']+df['Recovered']
df.head()
```

Saving… ×

	S. No.	Name of State / UT	TotalConfirmedCases	Active	Recovered	Deaths	totalcase
0	1.0	Maharashtra	20228	15649	3800	779	20228
1	2.0	Gujarat	7797	5234	2091	472	7797
2	3.0	Delhi	6542	4454	2020	68	6542
3	4.0	Tamil Nadu	6535	4667	1824	44	6535
4	5.0	Rajasthan	3741	1458	2176	107	3741

.drop is used to drop particular column

```
print("18012021034_ParmarSahilkhan")
df.drop(labels='TotalConfirmedCases', axis=1, inplace=True )
df.tail()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Active	Recovered	Deaths	totalcase
29	30.0	Manipur	0	2	0	2
30	31.0	Mizoram	0	1	0	1
31	32.0	Arunachal Pradesh	0	1	0	1
32	33.0	Dadra and Nagar Haveli and Daman and Diu	1	0	0	1

```
print("18012021034_ParmarSahilkhan")
import numpy as np
new_df=df[:]
df.head()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Active	Recovered	Deaths	totalcase
0	1.0	Maharashtra	15649	3800	779	20228
1	2.0	Gujarat	5234	2091	472	7797
2	3.0	Delhi	4454	2020	68	6542
3	4.0	Tamil Nadu	4667	1824	44	6535
4	5.0	Rajasthan	1458	2176	107	3741

.isnull() will return the column whic contains the null value and sum will return the totla no of missing value from each column

```
print("18012021034_ParmarSahilkhan")
new_df.isnull().sum()
```

Saving... ×

```
Name of State / UT    0
Active                0
Recovered             0
Deaths                0
totalcase             0
dtype: int64
```

.fillna is used to fill null values

```
print("18012021034_ParmarSahilkhan")
ndf=new_df.fillna(value= new_df.mean())
ndf.head()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Active	Recovered	Deaths	totalcase
0	1.0	Maharashtra	15649	3800	779	20228
1	2.0	Gujarat	5234	2091	472	7797
2	3.0	Delhi	4454	2020	68	6542
3	4.0	Tamil Nadu	4667	1824	44	6535

```
print("18012021034_ParmarSahilkhan")
ndf=new_df.fillna(value= new_df.median())
ndf.head()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Active	Recovered	Deaths	totalcase
0	1.0	Maharashtra	15649	3800	779	20228
1	2.0	Gujarat	5234	2091	472	7797
2	3.0	Delhi	4454	2020	68	6542
3	4.0	Tamil Nadu	4667	1824	44	6535
4	5.0	Rajasthan	1458	2176	107	3741

```
print("18012021034_ParmarSahilkhan")
ndf['Active']=ndf['Active'].astype(int)
ndf.head()
```

```
18012021034_ParmarSahilkhan
```

	S. No.	Name of State / UT	Active	Recovered	Deaths	totalcase
0	1.0	Maharashtra	15649	3800	779	20228
1	2.0	Gujarat	5234	2091	472	7797
2	3.0	Delhi	4454	2020	68	6542
[illegible]	[illegible]	[illegible]	4667	1824	44	6535
4	5.0	Rajasthan	1458	2176	107	3741

Saving… ×

```
import seaborn as sns
import matplotlib.pyplot as plt
```

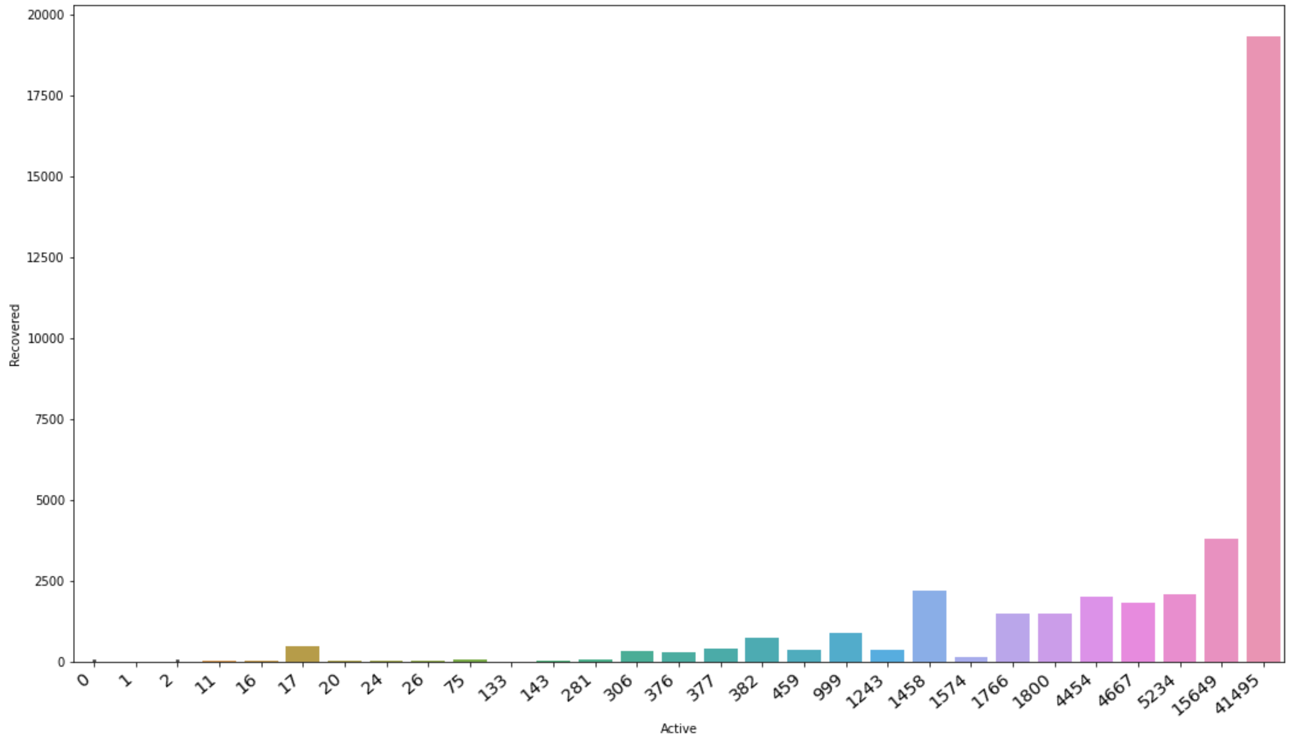
```
print("18012021034_ParmarSahilkhan")
df.reset_index(drop=True , inplace=True )
df
```

1	2.0	Gujarat	5234	2091	472	7797
2	3.0	Delhi	4454	2020	68	6542
3	4.0	Tamil Nadu	4667	1824	44	6535
4	5.0	Rajasthan	1458	2176	107	3741
5	6.0	Madhya Pradesh	1766	1480	211	3457
6	7.0	Uttar Pradesh	1800	1499	74	3373
7	8.0	Andhra Pradesh	999	887	44	1930
8	9.0	West Bengal	1243	372	171	1786
9	10.0	Punjab	1574	157	31	1762
10	11.0	Telangana	382	751	30	1163
11	12.0	Jammu and Kashmir	459	368	9	836
12	13.0	Karnataka	377	386	30	793
13	14.0	Haryana	376	290	9	675
14	15.0	Bihar	306	318	5	629
15	16.0	Kerala	17	485	4	506
16	17.0	Odisha	281	68	3	352
17	18.0	Chandigarh	143	24	2	169
18	19.0	Jharkhand	75	78	3	156
19	20.0	Tripura	133	2	0	135
20	21.0	Uttarakhand	20	46	1	67
21	22.0	Assam	26	35	1	62
22	23.0	Chhattisgarh	16	43	0	59
23	24.0	Himachal Pradesh	11	35	3	49
		Ladakh	24	18	0	42
25	26.0	Andaman and Nicobar Islands	0	33	0	33
26	27.0	Meghalaya	2	10	1	13
27	28.0	Puducherry	2	8	0	10
28	29.0	Goa	0	7	0	7
29	30.0	Manipur	0	2	0	2
30	31.0	Mizoram	0	1	0	1
31	32.0	Arunachal Pradesh	0	1	0	1

Saving... ×

```
print("18012021034_ParmarSahilkhan")
plt.figure(figsize=(18,10))
sns.barplot(x='Active',y='Recovered',data=df)
plt.xticks(rotation=40, horizontalalignment='right', fontweight='light', fontsize='x-large
plt.show()
```

```
18012021034_ParmarSahilkhan
```



Saving...