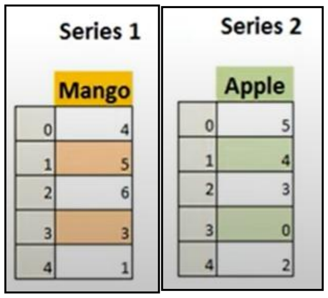
# ( INTRODUCTION )

- "Panel Data" or "Python Data Analysis"
- Used for analyzing,cleaning,exploring and manipulation of data
- It can work with - CSV,Text,Json,Zip,etc Files.



Series 1

|   | Mango |
|---|---|
| 0 | 4 |
| 1 | 5 |
| 2 | 6 |
| 3 | 3 |
| 4 | 1 |

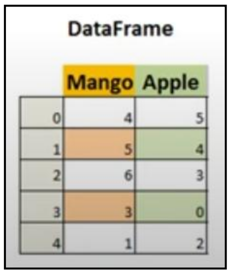
Series 2

|   | Apple |
|---|---|
| 0 | 5 |
| 1 | 4 |
| 2 | 3 |
| 3 | 0 |
| 4 | 2 |

## Types of Data Structures :-

**1. Series :** 1D Labeled arrays pd.Series(data).

**2. Dataframes :** 2D data structures of colums just like tables.



DataFrame

|   | Mango | Apple |
|---|---|---|
| 0 | 4 | 5 |
| 1 | 5 | 4 |
| 2 | 6 | 3 |
| 3 | 3 | 0 |
| 4 | 1 | 2 |

**3. Panel :** 3D container of data.

## For installation of pandas : -

Write command in cmd : **pip install pandas**

For importing pandas in python : **import pandas as pd**

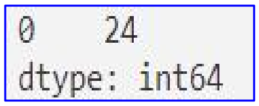
## Importance of Pandas :-

**1.** Pandas allows us to analze big data and make connclusions based on statical theories.
**2.** Pandas can clean messy data set ,make them reliable and relevant.
**3.** Eassily handling of missing data (NaN) in floating as well as non-floating.
**4.** Size Mutability : Can insert and delete data from dataframe and higher dimensional objects.
**5.** Provide Data set merging and joining.

# ( DATA STRUCTURES )

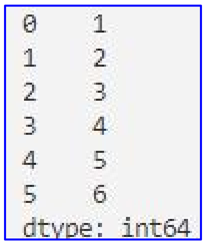
**Series( ) :** It is defined as the 1Darray capable of storing various types of data.

```
import pandas as pd
A = pd.Series(24)
print(A)
```



```
0    24
dtype: int64
```

```
import pandas as pd
X = [1,2,3,4,5,6]
A = pd.Series(X)
print(A)
```

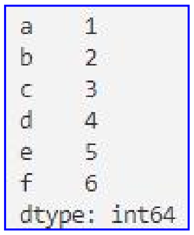


```
0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
```

```
print(type(A))    #    <class 'pandas.core.series.Series'>
print(A[1])        #    2
```

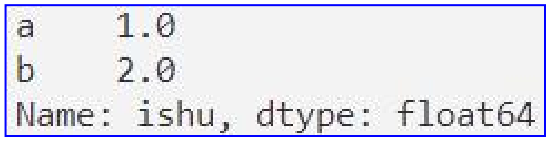
## <u>For Changing the index number</u>

```
import pandas as pd
X = [1,2,3,4,5,6]
A = pd.Series(X,index=['a','b','c','d','e','f'])
print(A)
print( A['a'])
```



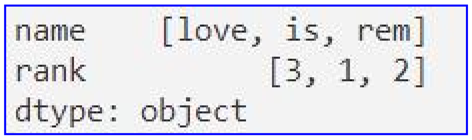
```
a    1
b    2
c    3
d    4
e    5
f    6
dtype: int64
```

```
import pandas as pd
X = [1,2]
A =
pd.Series(X,index=['a','b'],dtype="float",name="ishu")
print(A)
```



```
a    1.0
b    2.0
Name: ishu, dtype: float64
```

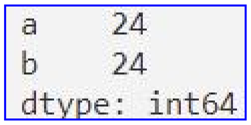
```
import pandas as pd
Dic = {"name":["love","is","rem"],"rank":[3,1,2]}
A = pd.Series(Dic)
print(A)
```



```
name    [love, is, rem]
rank          [3, 1, 2]
dtype: object
```

*#    name or rank can have more values .Here, the size of the of the list does not matter.*

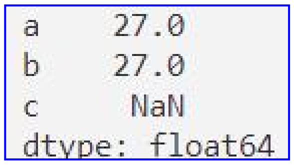
```
import pandas as pd
A = pd.Series(24,index=['a','b'])
print(A)
```



```
a    24
b    24
dtype: int64
```

*#   We can use direct operation without the use of broadcasting rule.*

```
import pandas as pd
A = pd.Series(24,index=['a','b'])
B = pd.Series(3,index=['a','b','c'])
print(A+B)
```



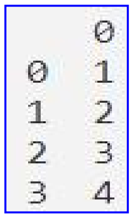
```
a    27.0
b    27.0
c     NaN
dtype: float64
```

*# NaN : Not a Number*
*# Here you can see it is working with the missing data.*

## ( DataFrame )

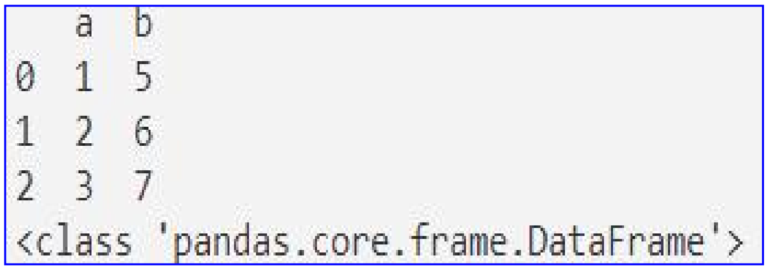
2D Data Structure is the **DataFrame. (***Eiher List or Dictionary***)**

```
import pandas as pd
A = [1,2,3,4]
Var = pd.DataFrame(A)
print(Var)
print(type(Var))      #    <class 'pandas.core.frame.DataFrame'>
```



```
   0
0  1
1  2
2  3
3  4
```

```
import pandas as pd
D = {"a":[1,2,3],"b":[5,6,7]}
Var = pd.DataFrame(D)
print(Var)
print(type(Var))
```

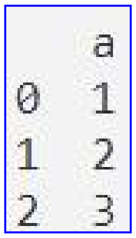


```
   a  b
0  1  5
1  2  6
2  3  7
<class 'pandas.core.frame.DataFrame'>
```

*#   Data must be of same length. In above you cannot put more than 2 element in a list.*
*#   **Values passed in a dictionary must be in list or tuple***

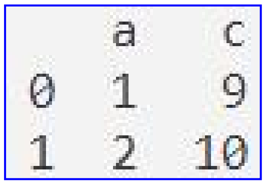
**<u>To get a particular column data . Eg:- We need the column named as "a" .</u>**

```
import pandas as pd
D = {"a":[1,2,3],"b":[5,6,7]}
Var = pd.DataFrame(D,columns=["a"])
print(Var)
```



```
   a
0  1
1  2
2  3
```

```
import pandas as pd
D = {"a":[1,2],"b":[5,6],"c":[9,10]}
Var = pd.DataFrame(D,columns=["a","c"])
print(Var)
```



```
   a   c
0  1   9
1  2  10
```

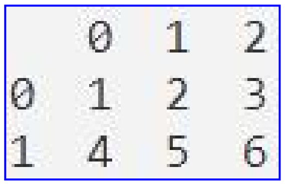
**<u>To get the value from a DataFrame .</u>**

```
import pandas as pd
D = {"a":[1,2],"b":[5,6],"c":[9,10]}
Var = pd.DataFrame(D,columns=["a","c"]) # It will take only "a" and "c" data.
print(Var["a"][0])         # 1
```

*Note: we have to first specify the column then indexing of that data .*

**<u>Convert a nested list into a DataFrame .</u>**

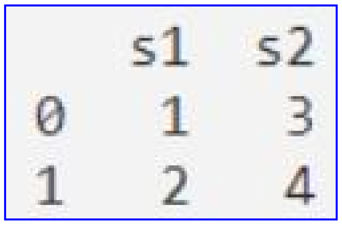
```
import pandas as pd
Ls = [[1,2,3],[4,5,6]]
Var = pd.DataFrame(Ls)
print(Var)
```



```
   0  1  2
0  1  2  3
1  4  5  6
```

**<u>To convert series of data into DataFrame .</u>**

```
import pandas as pd
Sr = {"s1":pd.Series([1,2]),"s2":pd.Series([3,4])}
Var = pd.DataFrame(Sr)
print(Var)
```



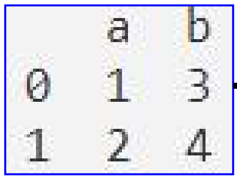
```
   s1  s2
0   1   3
1   2   4
```

Ishu_Agrawal_ CA

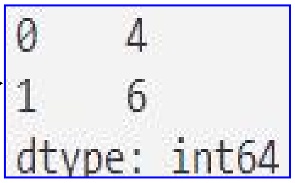
# ( ARITHMETIC OPERATIONS )

**Operations on columns.**

```python
import pandas as pd
Var = pd.DataFrame({"a":[1,2],"b":[3,4]})
print(Var["a"]+Var["b"])
```

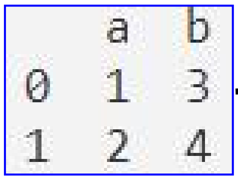


```
   a  b
0  1  3
1  2  4
```

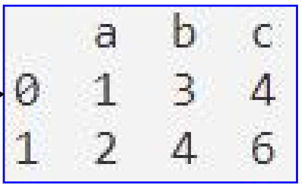


```
0    4
1    6
dtype: int64
```

```python
import pandas as pd
Var = pd.DataFrame({"a":[1,2],"b":[3,4]})
Var["c"] = Var["a"]+Var["b"]
print(Var)
```

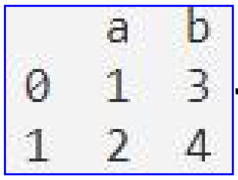


```
   a  b
0  1  3
1  2  4
```

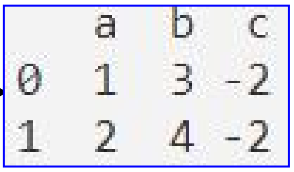


```
   a  b  c
0  1  3  4
1  2  4  6
```

```python
import pandas as pd
Var = pd.DataFrame({"a":[1,2],"b":[3,4]})
Var["c"] = Var["a"]-Var["b"]
print(Var)
```

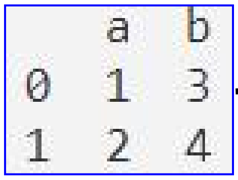


```
   a  b
0  1  3
1  2  4
```

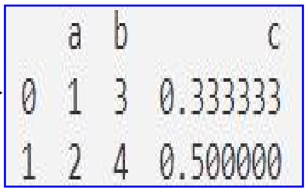


```
   a  b  c
0  1  3 -2
1  2  4 -2
```

```python
import pandas as pd
Var = pd.DataFrame({"a":[1,2],"b":[3,4]})
Var["c"] = Var["a"]/Var["b"]
print(Var)
```

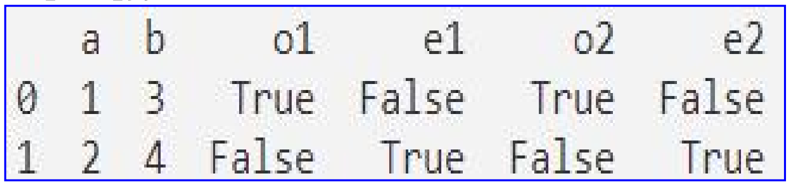


```
   a  b
0  1  3
1  2  4
```



```
   a  b         c
0  1  3  0.333333
1  2  4  0.500000
```

```python
import pandas as pd
Var = pd.DataFrame({"a":[1,2],"b":[3,4]})
Var["o1"] = Var["a"]%2!=0
Var["e1"] = Var["a"]%2==0
Var["o2"] = Var["b"]%2!=0
Var["e2"] = Var["b"]%2==0
print(Var)
```



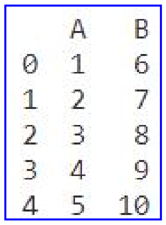
```
   a  b     o1     e1     o2     e2
0  1  3   True  False   True  False
1  2  4  False   True  False   True
```

Ishu_Agrawal_ CA

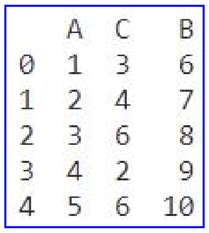
## ( DELETE / INSERT )

### INSERT

```
import pandas as pd
Var = pd.DataFrame( { "A" : [1,2,3,4,5], "B" : [6,7,8,9,10] } )
# Object.insert( index_of_place, define_name, data_to_insert )
Var.insert( 1 , "C", [3,4,6,2,6] )
# Remember : Length of data must be equal
print( Var )
```



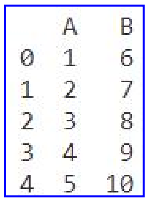
```
   A   B
0  1   6
1  2   7
2  3   8
3  4   9
4  5  10
```



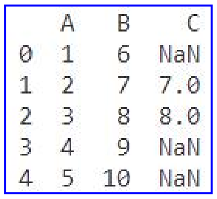
```
   A  C   B
0  1  3   6
1  2  4   7
2  3  6   8
3  4  2   9
4  5  6  10
```

### For copying the limited amount of data from a column

```
import pandas as pd
Var = pd.DataFrame({ "A" : [1,2,3,4,5],"B" : [6,7,8,9,10] })
Var["C"] = Var["B"][1:3]
print( Var )
```



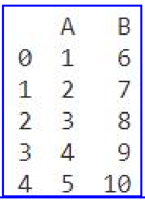
```
   A   B
0  1   6
1  2   7
2  3   8
3  4   9
4  5  10
```



```
   A   B    C
0  1   6  NaN
1  2   7  7.0
2  3   8  8.0
3  4   9  NaN
4  5  10  NaN
```

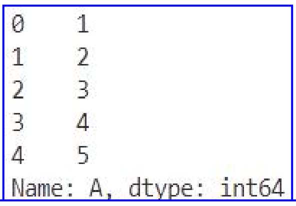
### DELETE    *(pop,del)*

```
import pandas as pd
Var = pd.DataFrame({ "A" : [1,2,3,4,5],"B" : [6,7,8,9,10] })
Var1=Var.pop("A")
print(Var1)
print(Var)
```



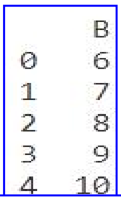
```
   A   B
0  1   6
1  2   7
2  3   8
3  4   9
4  5  10
```

Var( Before )



```
0    1
1    2
2    3
3    4
4    5
Name: A, dtype: int64
```

Var1



```
    B
0   6
1   7
2   8
3   9
4  10
```

Var ( After )

***Similarly :*** If you want to use del and want to remove column A . So, Use

del Var["A"]

***Excel File :*** *Excel File have Binary Coded data .*
***CSV File :*** *CSV File have comma separated values as plain text .*

Ishu_Agrawal_ CA

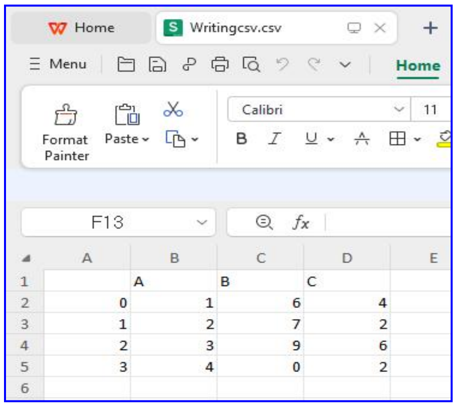
# ( WRITING CSV FILE )

CSV format is a plain text format in which values are separated by commas.

## How to write CSV File

*Before writing a CSV file first create dataframe.*

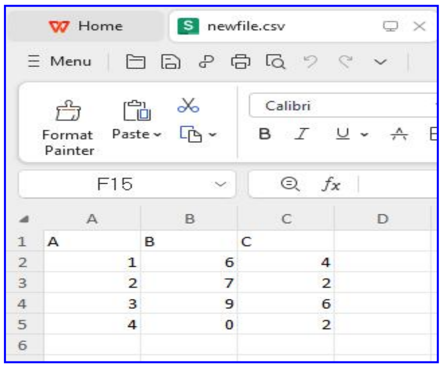
```
import pandas as pd
dic={"A":[1,2,3,4],"B":[6,7,9,0],"C":[4,2,6,2]}
d=pd.DataFrame(dic)
d.to_csv("Writingcsv.csv")
```



|  | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 |  | A | B | C |  |
| 2 | 0 | 1 | 6 | 4 |  |
| 3 | 1 | 2 | 7 | 2 |  |
| 4 | 2 | 3 | 9 | 6 |  |
| 5 | 3 | 4 | 0 | 2 |  |
| 6 |  |  |  |  |  |

*# Here, you can see indexes are also present in the file so to remove this we can set index to False.*

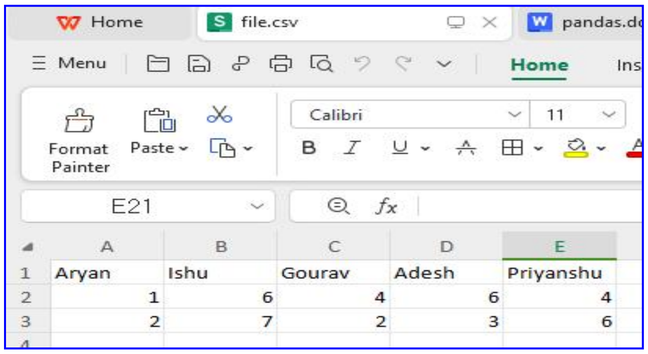
```
import pandas as pd
dic={"A":[1,2,3,4],"B":[6,7,9,0],"C":[4,2,6,2]}
d=pd.DataFrame(dic)
d.to_csv("newfile.csv",index=False)
```



|  | A | B | C | D |
|---|---|---|---|---|
| 1 | A | B | C |  |
| 2 | 1 | 6 | 4 |  |
| 3 | 2 | 7 | 2 |  |
| 4 | 3 | 9 | 6 |  |
| 5 | 4 | 0 | 2 |  |
| 6 |  |  |  |  |

## For changing the header

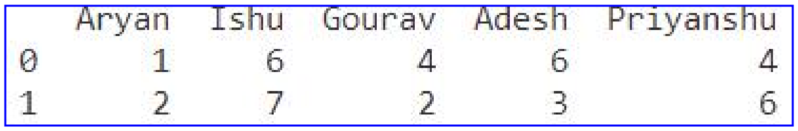
```
import pandas as pd
dic={"A":[1,2],"B":[6,7],"C":[4,2],"D":[6,3],"E":[4,6]}
d=pd.DataFrame(dic)
d.to_csv("file.csv",index=False,header=["Aryan","Ishu","Gourav","Adesh","Priyanshu"])
```



|  | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Aryan | Ishu | Gourav | Adesh | Priyanshu |
| 2 | 1 | 6 | 4 | 6 | 4 |
| 3 | 2 | 7 | 2 | 3 | 6 |

# ( READING CSV FILE )

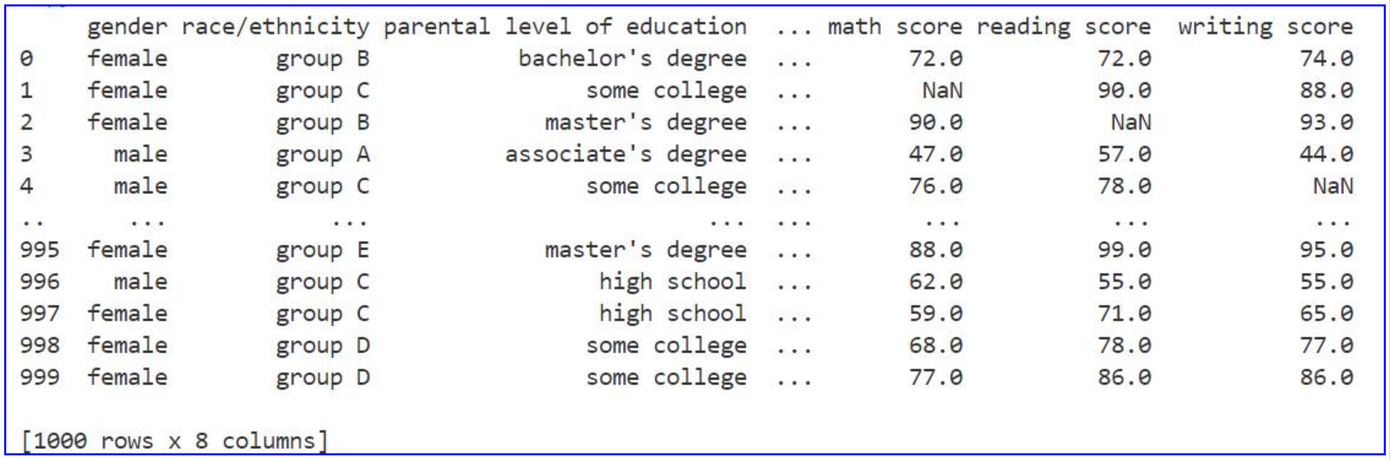
```
import pandas as pd
csfile=pd.read_csv("file.csv")
print(csfile)
```



```
   Aryan  Ishu  Gourav  Adesh  Priyanshu
0      1     6       4      6          4
1      2     7       2      3          6
```

**Download link : StudensPerformance.csv file**

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv")
print(rdcsv)
```

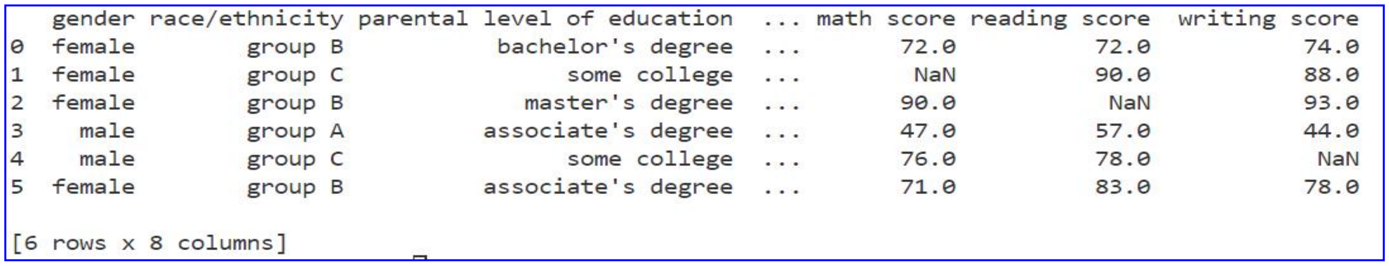


```
     gender race/ethnicity parental level of education  ... math score reading score  writing score
0    female        group B           bachelor's degree  ...       72.0          72.0           74.0
1    female        group C                some college  ...        NaN          90.0           88.0
2    female        group B             master's degree  ...       90.0           NaN           93.0
3      male        group A          associate's degree  ...       47.0          57.0           44.0
4      male        group C                some college  ...       76.0          78.0            NaN
..      ...            ...                         ...  ...        ...           ...            ...
995  female        group E             master's degree  ...       88.0          99.0           95.0
996    male        group C                 high school  ...       62.0          55.0           55.0
997  female        group C                 high school  ...       59.0          71.0           65.0
998  female        group D                some college  ...       68.0          78.0           77.0
999  female        group D                some college  ...       77.0          86.0           86.0

[1000 rows x 8 columns]
```

*# see here it only give fist and last five record only.*

**<u>To get the number of rows from starting</u>**

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=6)
print(rdcsv)
```

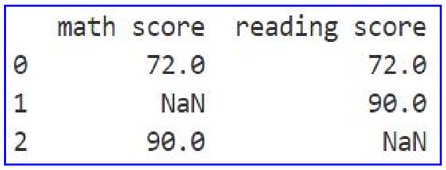


```
   gender race/ethnicity parental level of education  ... math score reading score  writing score
0  female        group B           bachelor's degree  ...       72.0          72.0           74.0
1  female        group C                some college  ...        NaN          90.0           88.0
2  female        group B             master's degree  ...       90.0           NaN           93.0
3    male        group A          associate's degree  ...       47.0          57.0           44.0
4    male        group C                some college  ...       76.0          78.0            NaN
5  female        group B          associate's degree  ...       71.0          83.0           78.0

[6 rows x 8 columns]
```

## To get columns

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",usecols=["math score","reading score"],nrows=3)
print(rdcsv)
```

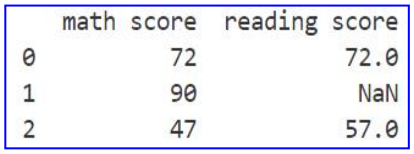
***Note :*** *we can also use index instead of writing name of particular column.*
*Eg: For this we can use* **usecols = [5,6]**



```
   math score  reading score
0        72.0           72.0
1         NaN           90.0
2        90.0            NaN
```

## For skipping the Row/Rows

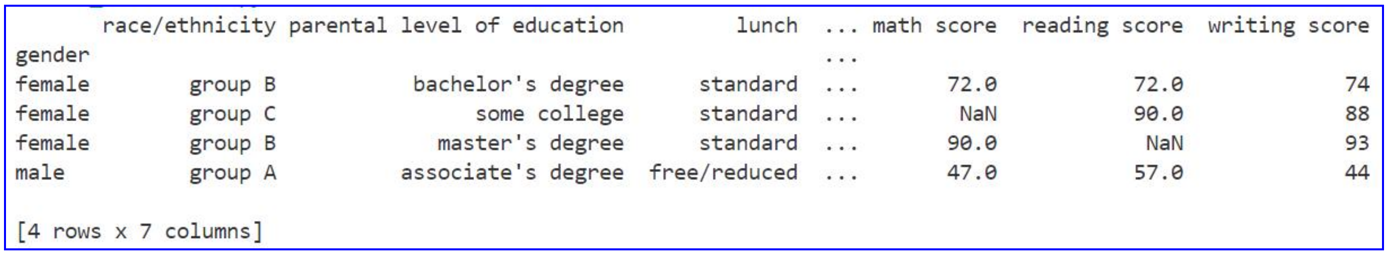
```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",usecols=[5,6],nrows=3,skiprows=[2])
print(rdcsv)
```



```
   math score  reading score
0          72           72.0
1          90            NaN
2          47           57.0
```

## To make a particular column as index

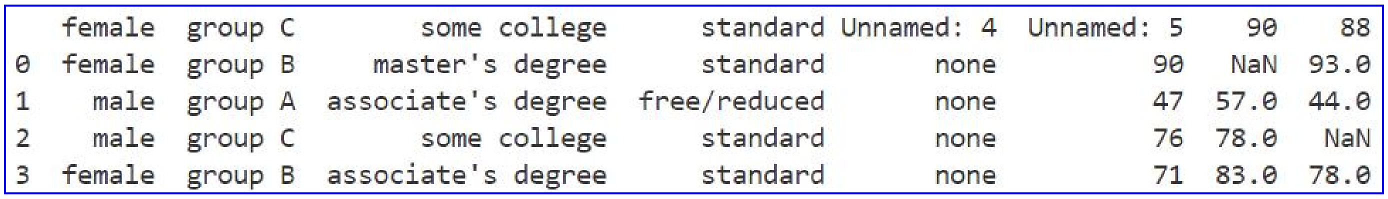
```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",index_col="gender",nrows=4)
print(rdcsv)
```



```
       race/ethnicity parental level of education         lunch  ... math score  reading score  writing score
gender                                                           ...
female        group B           bachelor's degree      standard  ...       72.0           72.0             74
female        group C                some college      standard  ...        NaN           90.0             88
female        group B             master's degree      standard  ...       90.0            NaN             93
male          group A          associate's degree  free/reduced  ...       47.0           57.0             44

[4 rows x 7 columns]
```

## To make row as a Header

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4,header=2)
print(rdcsv)
```

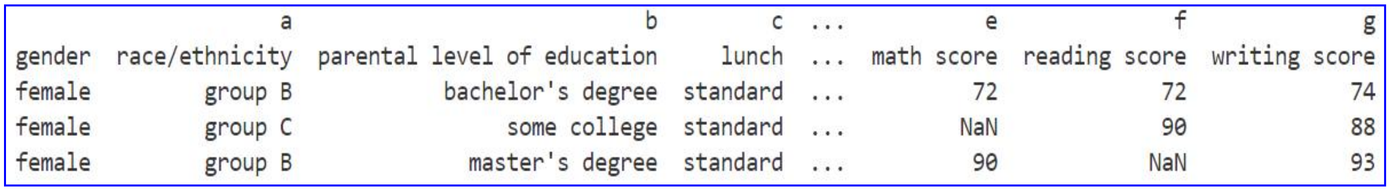


```
   female  group C         some college      standard Unnamed: 4  Unnamed: 5    90    88
0  female  group B      master's degree      standard       none          90   NaN  93.0
1    male  group A   associate's degree  free/reduced       none          47  57.0  44.0
2    male  group C         some college      standard       none          76  78.0   NaN
3  female  group B   associate's degree      standard       none          71  83.0  78.0
```

Ishu_Agrawal_ CA

## To give name to header

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4,names=["a","b","c","d","e","f","g"])
print(rdcsv)
```

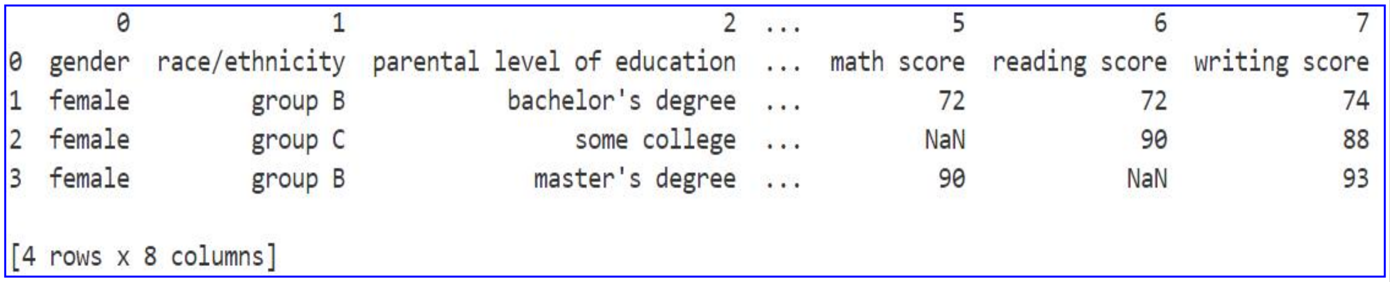


```
                       a                            b         c  ...           e               f              g
gender  race/ethnicity  parental level of education     lunch  ...  math score  reading score  writing score
female         group B            bachelor's degree  standard  ...          72              72             74
female         group C                 some college  standard  ...         NaN              90             88
female         group B              master's degree  standard  ...          90             NaN             93
```

**Note:** *The names of columns must be unique.*

## To remove header

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4,header=None)
print(rdcsv)
```



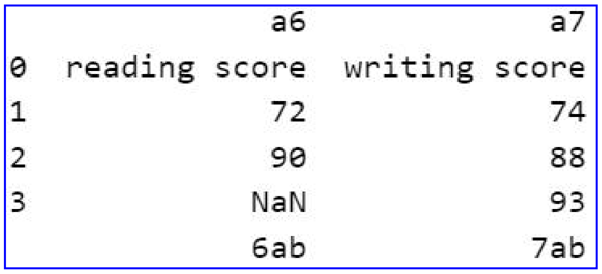
```
        0               1                            2  ...           5              6              7
0  gender  race/ethnicity  parental level of education  ...  math score  reading score  writing score
1  female         group B            bachelor's degree  ...          72             72             74
2  female         group C                 some college  ...         NaN             90             88
3  female         group B              master's degree  ...          90            NaN             93

[4 rows x 8 columns]
```

**Note :** ***When we remove header it will be replace by indexes of that particular column***

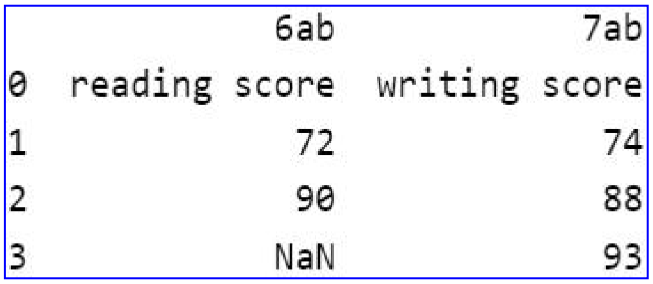
***You can change the type of a column by using*** ***dtype= {"column_name" : "data_type"}***

***For adding something in adder before use*** ***rdcsv.add_prefix("a").***

***For adding something in adder before use*** ***rdcsv.add_suffix("ab").***



```
              a6              a7
0  reading score  writing score
1             72              74
2             90              88
3            NaN              93
             6ab             7ab
```



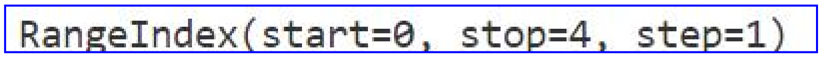
```
             6ab            7ab
0  reading score  writing score
1             72             74
2             90             88
3            NaN             93
```

***rdcsv.add_prefix("a")***　　　　***rdcsv.add_suffix("ab")***

# ( PANDAS FUCTION )

## For printing index details : index

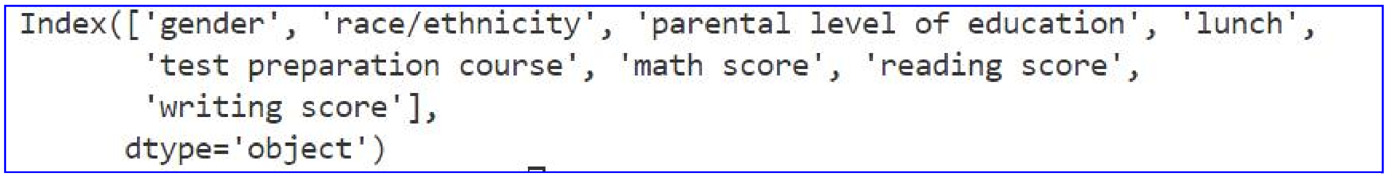
```
import pandas as pd
rdcsv = pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.index)
```



```
RangeIndex(start=0, stop=4, step=1)
```

## For printing columns name : columns

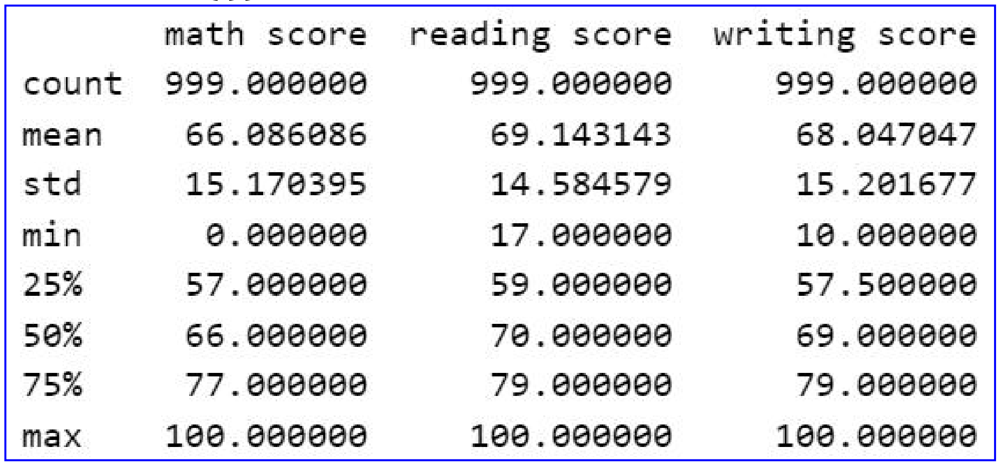
```
import pandas as pd
rdcsv = pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.columns)
```



```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

## To find all the detail about data : describe()

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv")
print(rdcsv.describe( ))
```



```
       math score  reading score  writing score
count  999.000000     999.000000     999.000000
mean    66.086086      69.143143      68.047047
std     15.170395      14.584579      15.201677
min      0.000000      17.000000      10.000000
25%     57.000000      59.000000      57.500000
50%     66.000000      70.000000      69.000000
75%     77.000000      79.000000      79.000000
max    100.000000     100.000000     100.000000
```

***# If we use "nrows" then it will data according to those rows.***

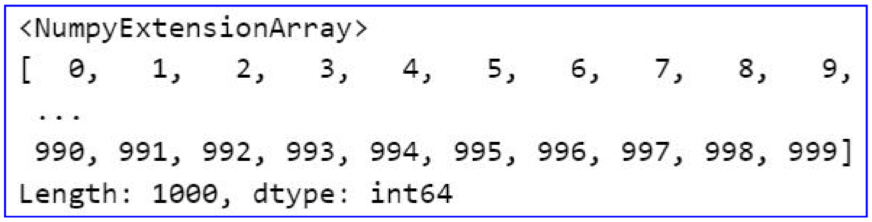
Ishu_Agrawal_ CA

## For getting records from data

```
print(rdcsv.head( ))            # give first 5 rows
print(rdcsv.head(2))      # give first 2 rows
print(rdcsv.tail( ))            # give last 5 rows
print(rdcsv.tail(3))      # give last 3 rows
print(rdcsv[2:5])          # give rows from index 2 to 4
```

## To print Indexes as array

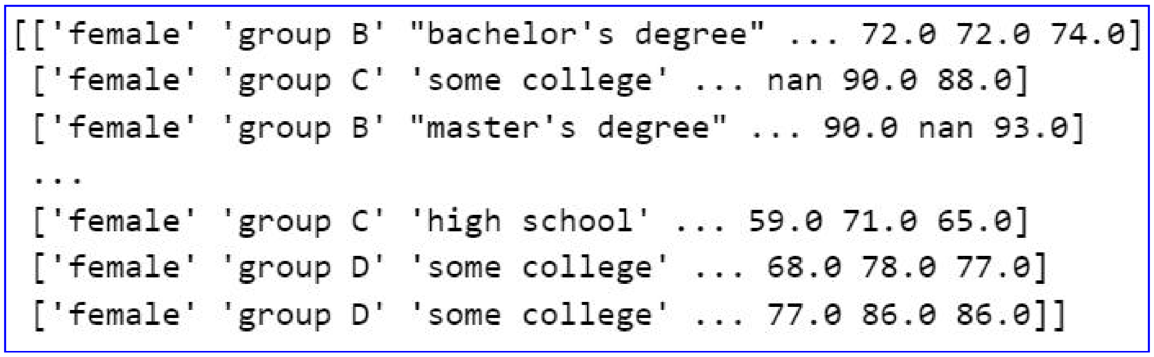
```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv")
print(rdcsv.index.array)
```



```
<NumpyExtensionArray>
[  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,
 ...
 990, 991, 992, 993, 994, 995, 996, 997, 998, 999]
Length: 1000, dtype: int64
```

## To covert All recors into array

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv")
print(rdcsv.to_numpy())
```

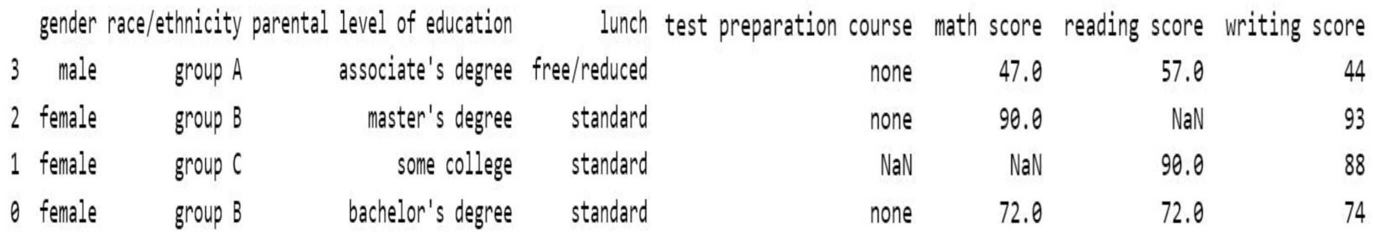


```
[['female' 'group B' "bachelor's degree" ... 72.0 72.0 74.0]
 ['female' 'group C' 'some college' ... nan 90.0 88.0]
 ['female' 'group B' "master's degree" ... 90.0 nan 93.0]
 ...
 ['female' 'group C' 'high school' ... 59.0 71.0 65.0]
 ['female' 'group D' 'some college' ... 68.0 78.0 77.0]
 ['female' 'group D' 'some college' ... 77.0 86.0 86.0]]
```

## To sort data in ascending order

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.sort_index(axis=0,ascending=False))
```

axis = 0 : according to row.



```
  gender race/ethnicity parental level of education        lunch test preparation course  math score  reading score  writing score
3   male        group A          associate's degree  free/reduced                     none        47.0           57.0             44
2 female        group B             master's degree      standard                     none        90.0            NaN             93
1 female        group C                some college      standard                      NaN         NaN           90.0             88
0 female        group B           bachelor's degree      standard                     none        72.0           72.0             74
```

axis = 1 : according to column.

**Q.) Change the gender from female to make of the first record.**

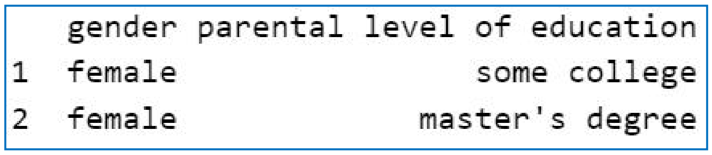
Ans) rdcsv[“gender”][0]=”male”

# This particular statement will work and also though error but this is the wrong way to change the particular thing in in a record.

**Correct way :** rdcsv.loc[0,"gender"]="male"

- **For getting multiple column data using loc .**

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.loc[[1,2],["gender","parental level of education"]])
```



```
   gender parental level of education
1  female                some college
2  female             master's degree
```

- **For getting all the rows :**

rdcsv.loc[ : ,["gender","parental level of education"]]

- **For getting all the rows and columns :**

rdcsv.loc[ : , : ]

## Use of iloc : To get a particular data

# in this we pass index number only

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.iloc[0,0])          # female
```

- **For multiple rows and columns using iloc**

```
rdcsv.iloc[[0,1],[0,2]])
```

## Use Of drop : For skipping rows and columns

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.drop("gender",axis=1))
```

**# if axis is 1 then we can skip columns otherwise rows for 0**

- **For skipping multiple column**

```
rdcsv.drop(["gender","lunch"],axis=1)
```

- **For skipping a row**

```
rdcsv.drop(1,axis=0)
```

- **For skipping multiple rows**

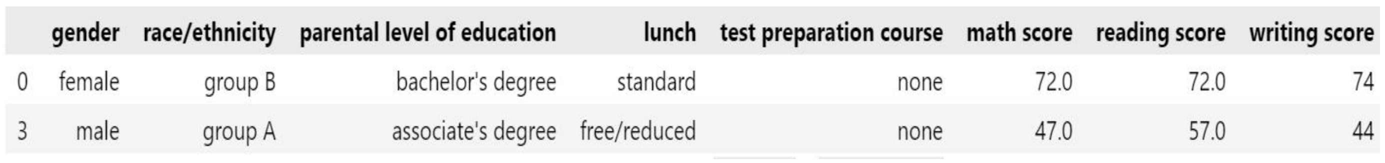
```
rdcsv.drop([0,2,3],axis=0)
```

# Last one will give only one row as record because we have select nrows=4 , if you try to drop 4 or greater than 4 index data then it will through error.

# ( HANDLING MISSING VALUES )

**Use of dropna( ) :** It will remove the whole record if any NaN value or empty spaces found in that particular record.

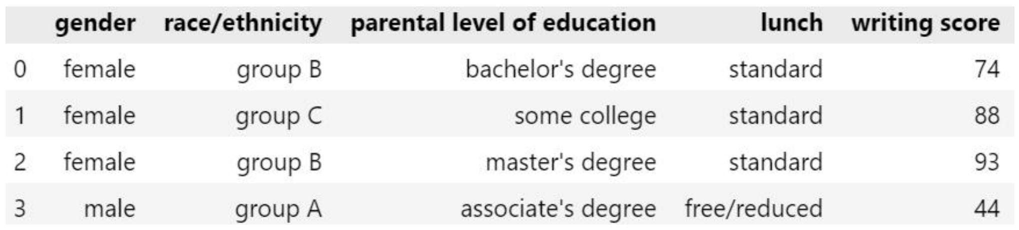
```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.dropna(    ))
```



| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

*# In this drop is done along rows .*

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.dropna(axis=1))
```



| | gender | race/ethnicity | parental level of education | lunch | writing score |
|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | 74 |
| 1 | female | group C | some college | standard | 88 |
| 2 | female | group B | master's degree | standard | 93 |
| 3 | male | group A | associate's degree | free/reduced | 44 |

*# In this drop is done along columns .*

## Use of how parameter in drop

- **For removing those rows which have any NaN value**

  rdcsv.dropna(axis=0,how="any")

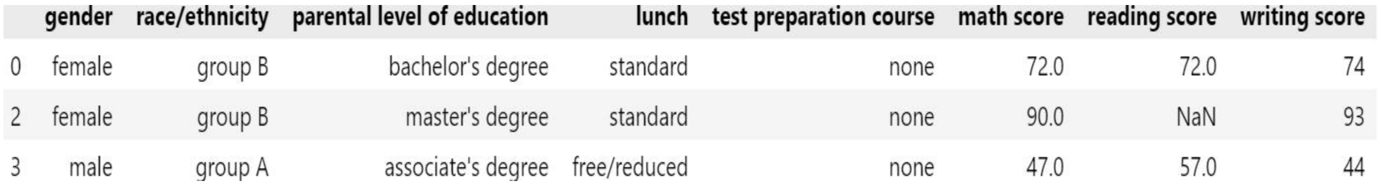
- **For removing those rows which have all NaN value not those which have some record**

  rdcsv.dropna(axis=0,how="all")

## Use of subset parameter in drop :

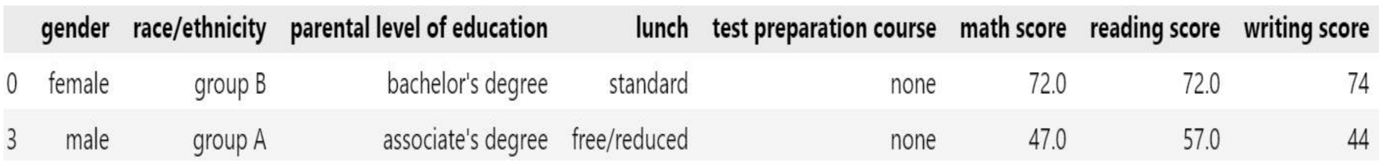
It will remove the NaN record of a particular columns.

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.dropna(axis=0,subset=["math score"]))
```



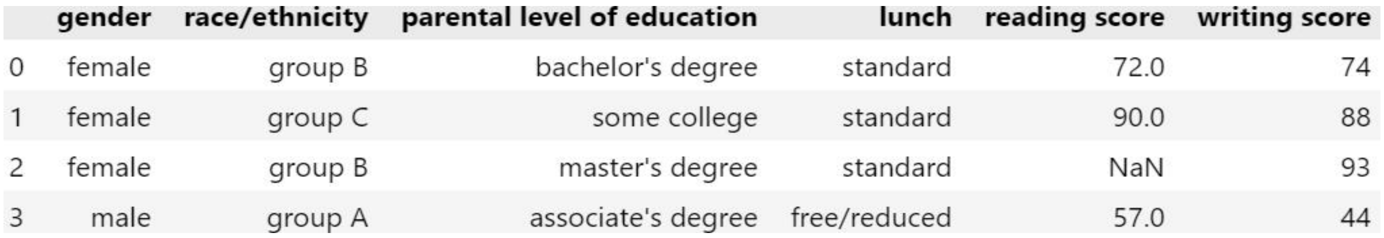
| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 2 | female | group B | master's degree | standard | none | 90.0 | NaN | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.dropna(axis=0,subset=["math score","reading score"])
```



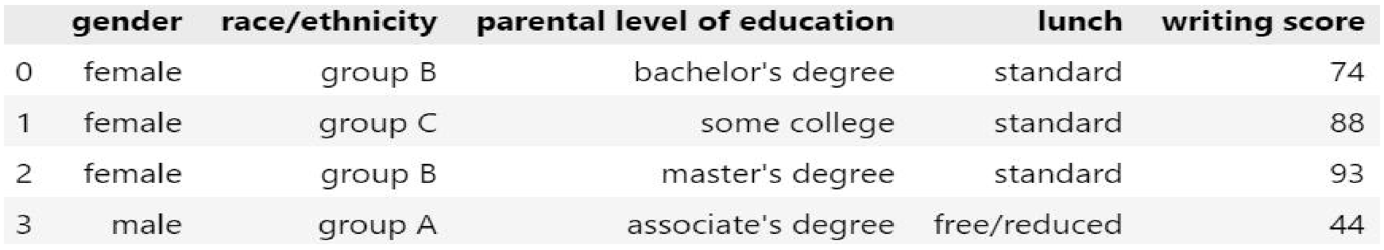
| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.dropna(axis=1,subset=1)
```



| | gender | race/ethnicity | parental level of education | lunch | reading score | writing score |
|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | 72.0 | 74 |
| 1 | female | group C | some college | standard | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | NaN | 93 |
| 3 | male | group A | associate's degree | free/reduced | 57.0 | 44 |

```
rdcsv.dropna(axis=1,subset=[1,2])
```



| | gender | race/ethnicity | parental level of education | lunch | writing score |
|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | 74 |
| 1 | female | group C | some college | standard | 88 |
| 2 | female | group B | master's degree | standard | 93 |
| 3 | male | group A | associate's degree | free/reduced | 44 |

Ishu_Agrawal_ CA

- **Use of inplace in dropna:** To remove null values and convert into a new database.

  Eg: rdscv.dropna(inplace=True)

- **Use of thresh in dropna :** We can specify no. Of NaN values rows to remove.

  Eg: rdscv.dropna(thresh=2)

## Use of fillna( )

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.fillna("ishu"))
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | ishu | ishu | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | ishu | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

- **<u>For filling specific values in columns</u>**

```
import pandas as pd
rdcsv=pd.read_csv("StudentsPerformance.csv",nrows=4)
print(rdcsv.fillna({"math score":45,"reading score":69}))
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | NaN | 45.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 69.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

- **<u>For filling the data according to forward/backward data</u>**

```
rdcsv.fillna(method="ffill")    # for forward filling
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | none | 72.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 90.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.fillna(method="bfill")    # for backward filling
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | none | 90.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 57.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.fillna(method="ffill",axis=1)
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | none | 72.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 90.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.fillna(method="bfill",axis=1)
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | 90.0 | 90.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 93 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.fillna(method="bfill",axis=0)
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | none | 90.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 57.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

```
rdcsv.fillna(method="ffill",axis=0)
```

| | gender | race/ethnicity | parental level of education | lunch | test preparation course | math score | reading score | writing score |
|---|---|---|---|---|---|---|---|---|
| 0 | female | group B | bachelor's degree | standard | none | 72.0 | 72.0 | 74 |
| 1 | female | group C | some college | standard | none | 72.0 | 90.0 | 88 |
| 2 | female | group B | master's degree | standard | none | 90.0 | 90.0 | 93 |
| 3 | male | group A | associate's degree | free/reduced | none | 47.0 | 57.0 | 44 |

Ishu_Agrawal_ CA

# CLASS NOTES

```
import pandas as pd
a=pd.read_csv("student.csv")
print(a.info()) # gives the information of that data with its column name
print(a.head()) # give first five record by default
print(a.head(7)) # give first seven record
print(a.tail()) # give last 5 records by default
print(a.tail(7)) # give last seven record
print(pd.options.display.max_rows)# give how many records will be visible
pd.options.display.max_rows=13 # defining we can only display 10 records as specified
print(a.head().isnull()) # give the result in which it will replace NaN to True
print(a.loc[0]) # give the zeroth index column data
print(a.loc[[0,2]]) # give the zeroth and second index column data
print(a.loc[1:11]) # give the rows of data from 1 to 11 included
print(a['math score'].loc[[0,1]]) # give only math score on 0 and 1 indexes records
print(a[['gender','math score',]].loc[[0,1]]) # give gender and math score of 0 and 1
indexes records
b=a.fillna(99) # It will fill Nan value to 99 in which variable data is returned
print(b.head()) # Note: it will not affect the original data in a
# a.fillna(99,inplace=True) # It will affect the original data
# print(a.head())
print("msmean",a["math score"].mean(),type(a["math score"].mean())) # gives mean of math
score as float
print("msmedian",a["math score"].median(),type(a["math score"].median())) # gives median
of math score as float
print("msmode",a["math score"].mode(),type(a["math score"].mode())) # gives mode of math
score as series
a["math score"].fillna(a["math score"].mean(),inplace=True) # it will replace Nan value
to corresponding statical method
print(a.head())
a["math score"].fillna(a["math score"].mode()[0],inplace=True) # we have to replace
zeroth position in order to change mode of the data
print(a.head())
print(a.describe()) # gives all statical record
a.dropna(inplace=True) # drop all Nan valued record
print(a.head())

c=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]])
print(c)
print(c.agg("cumsum",axis=1)) # give cumsum value row-wise
print(c.agg("cumsum",axis=0)) # give cumsum value column-wise

print(c.agg("sum",axis=1))
print(c.agg("sum",axis=0))
```