```
import pandas as pd
import numpy as np
```

```
data = {
    "Name": ["Steve", "Lia", "Vin", "Katie"],
    "Age": [32, 28, 45, 38],
    "Gender": ["Male", "Female", "Male", "Female"],
    "Review": [3.45, 4.60, 3.90, 2.78]
}

# Creating a DataFrame
df = pd.DataFrame(data, index=["r1", "r2", "r3", "r4"])

# Displaying the DataFrame
print(df)
```

```
     Name  Age  Gender  Review
r1  Steve   32    Male    3.45
r2    Lia   28  Female    4.60
r3    Vin   45    Male    3.90
r4  Katie   38  Female    2.78
```

```
df=df.rename(columns={"Review":"Rating"})
```

```
df
```

	Name	Age	Gender	Rating
r1	Steve	32	Male	3.45
r2	Lia	28	Female	4.60
r3	Vin	45	Male	3.90
r4	Katie	38	Female	2.78

```
df.rename(columns={"Rating":"Review"},inplace=True)#inplace = True -- means the changes will be eff
```

```
df
```

	Name	Age	Gender	Review
r1	Steve	32	Male	3.45
r2	Lia	28	Female	4.60
r3	Vin	45	Male	3.90
r4	Katie	38	Female	2.78

```
df=df.rename(index={"r1":"row1","r2":"row2"})
```

```
df
```

	Name	Age	Gender	Review
row1	Steve	32	Male	3.45
row2	Lia	28	Female	4.60
r3	Vin	45	Male	3.90
r4	Katie	38	Female	2.78

```
df["country"]=["Aus","US","Uk","sweden"]
```

```
df
```

```
     Name  Age  Gender  Review  country
r1  Steve   32    Male    3.45      Aus
r2    Lia   28  Female    4.60       US
r3    Vin   45    Male    3.90       Uk
r4  Katie   38  Female    2.78   sweden
```

```
df.insert(3,"city",["melbourne","dertroit","london","stolkhom"])
```

```
df
```

```
     Name  Age  Gender       city  Review  country
r1  Steve   32    Male  melbourne    3.45      Aus
r2    Lia   28  Female   dertroit    4.60       US
r3    Vin   45    Male     london    3.90       Uk
r4  Katie   38  Female   stolkhom    2.78   sweden
```

```
df["Age"]=[30,45,89,25]
```

```
df
```

```
       Name  Age  Gender       city  Review  country
row1  Steve   30    Male  melbourne    3.45      Aus
row2    Lia   45  Female   dertroit    4.60       US
  r3    Vin   89    Male     london    3.90       Uk
  r4  Katie   25  Female   stolkhom    2.78   sweden
```

```
df.replace({"city":"melbourne","Review":4.60},"Nil",inplace=True)
```

```
df
```

```
       Name  Age  Gender      city  Review  country
row1  Steve   30    Male       Nil    3.45      Aus
row2    Lia   45  Female  dertroit     Nil       US
  r3    Vin   89    Male    london     3.9       Uk
  r4  Katie   25  Female  stolkhom    2.78   sweden
```

```
#you can delete columns from your data frme usinf the .drop()method

df.drop(columns=["Gender","city"])
```

```
       Name  Age  Review  country
row1  Steve   30    3.45      Aus
row2    Lia   45     Nil       US
  r3    Vin   89     3.9       Uk
  r4  Katie   25    2.78   sweden
```

```
df
```

```
      Name  Age  Gender      city  Review  country
row1  Steve  30    Male       Nil    3.45      Aus
row2    Lia  45  Female  dertroit     Nil       US
  r3    Vin  89    Male    london     3.9       Uk
  r4  Katie  25  Female  stolkhom    2.78   sweden
```

```
df.drop(columns=["Gender","city"],inplace=True)
```

```
df
```

```
      Name  Age  Review  country
row1  Steve  30    3.45      Aus
row2    Lia  45     Nil       US
  r3    Vin  89     3.9       Uk
  r4  Katie  25    2.78   sweden
```

```
df.drop("r3")
df
```

```
      Name  Age  Review       city  country
row1  Steve  30    3.45  melbourne      Aus
row2    Lia  45     Nil   dertroit       US
  r3    Vin  89     3.9     london       Uk
  r4  Katie  25    2.78   stolkhom   sweden
```

```
df.insert(2,"Gender",["Male","Male","Female","Female"])
```

```
df
```

```
      Name  Age  Gender  Review       city  country
row1  Steve  30    Male    3.45  melbourne      Aus
row2    Lia  45    Male     Nil   dertroit       US
  r3    Vin  89  Female     3.9     london       Uk
  r4  Katie  25  Female    2.78   stolkhom   sweden
```

```
#you can delete multuole rows from your data frame by  passing  a list of row labels to your .drop
```

```
df=df.drop(["row1","r4"])
```

```
df
```

```
     Name  Age  Gender  Review      city  country
row2  Lia   45    Male     Nil  dertroit       US
  r3  Vin   89  Female     3.9  london         Uk
```

```
data = {
    "Name": ["Steve", "Lia", "Vin", "Katie"],
    "Age": [32, 28, 45, 38],
    "Gender": ["Male", "Female", "Male", "Female"],
    "Review": [3.45, 4.60, 3.90, 2.78]
}
```

```
# Creating a DataFrame
df = pd.DataFrame(data, index=["r1", "r2", "r3", "r4"])

# Displaying the DataFrame
print(df)
```

```
     Name  Age  Gender  Review
r1  Steve   32    Male    3.45
r2    Lia   28  Female    4.60
r3    Vin   45    Male    3.90
r4  Katie   38  Female    2.78
```

```
df
```

	Name	Age	Gender	city	Review	country
r1	Steve	32	Male	melbourne	3.45	Aus
r2	Lia	28	Female	dertroit	4.60	US
r3	Vin	45	Male	london	3.90	Uk
r4	Katie	38	Female	stolkhom	2.78	sweden

```
df=df[df["Review"]<=4]
df
```

	Name	Age	Gender	city	Review	country
r1	Steve	32	Male	melbourne	3.45	Aus
r3	Vin	45	Male	london	3.90	Uk
r4	Katie	38	Female	stolkhom	2.78	sweden

```
data = {
    "Name": ["Steve", "Lia", "Vin", "Katie"],
    "Age": [32, 28, 45, 38],
    "Gender": ["Male", "Female", "Male", "Female"],
    "Review": [3.45, 4.60, 3.90, 2.78]
}

# Creating a DataFrame
df = pd.DataFrame(data, index=["r1", "r2", "r3", "r4"])

# Displaying the DataFrame
print(df)
df["country"]=["Aus","US","Uk","sweden"]
df.insert(3,"city",["melbourne","dertroit","london","stolkhom"])
```

```
     Name  Age  Gender  Review
r1  Steve   32    Male    3.45
r2    Lia   28  Female    4.60
r3    Vin   45    Male    3.90
r4  Katie   38  Female    2.78
```

```
df
```

	Name	Age	Gender	city	Review	country
r1	Steve	32	Male	melbourne	3.45	Aus
r2	Lia	28	Female	dertroit	4.60	US
r3	Vin	45	Male	london	3.90	Uk
r4	Katie	38	Female	stolkhom	2.78	sweden

```
df.drop(df.index[0:2])
```

	Name	Age	Gender	city	Review	country
r3	Vin	45	Male	london	3.90	Uk
r4	Katie	38	Female	stolkhom	2.78	sweden

```
df=df.drop(df.index[0::2])#this slicing is also the stop value excluded.
```

```
df
```

	Name	Age	Gender	city	Review	country
r2	Lia	28	Female	dertroit	4.60	US
r4	Katie	38	Female	stolkhom	2.78	sweden

```
df_num = pd.DataFrame(np.random.rand(4,5),index=['R1','R2','R3','R4'],columns=['C1','C2','C3','C4'
```

```
df_num
```

	C1	C2	C3	C4	C5
R1	0.832560	0.569747	0.913261	0.717511	0.025963
R2	0.642766	0.843965	0.838349	0.191944	0.876494
R3	0.938047	0.778409	0.326198	0.424893	0.802749
R4	0.769201	0.705776	0.206219	0.002919	0.471366

```
df*3
```

	Name	Age	Gender	city	Review	country
r2	LiaLiaLia	84	FemaleFemaleFemale	dertroitdertroitdertroit	13.80	USUSUS
r4	KatieKatieKatie	114	FemaleFemaleFemale	stolkhomstolkhomstolkhom	8.34	swedenswedensweden

```
df_num*2  #it just gives a copy of the Data frame with changes ,the original DataFrame is not affe
```

	C1	C2	C3	C4	C5
R1	1.665120	1.139494	1.826522	1.435021	0.051925
R2	1.285532	1.687930	1.676697	0.383887	1.752989
R3	1.876094	1.556817	0.652396	0.849787	1.605498
R4	1.538402	1.411552	0.412438	0.005838	0.942732

```
df
```

	Name	Age	Gender	city	Review	country
r2	Lia	28	Female	dertroit	4.60	US
r4	Katie	38	Female	stolkhom	2.78	sweden

```
df_num
```

	C1	C2	C3	C4	C5
R1	0.832560	0.569747	0.913261	0.717511	0.025963
R2	0.642766	0.843965	0.838349	0.191944	0.876494
R3	0.938047	0.778409	0.326198	0.424893	0.802749
R4	0.769201	0.705776	0.206219	0.002919	0.471366

```
#Arithmetic operations between two data frames
```

```
#we have seen taht dataframes are allignes by theier index and column  labels.so it will perform y
```

```
df_num
```

	C1	C2	C3	C4	C5
R1	0.832560	0.569747	0.913261	0.717511	0.025963
R2	0.642766	0.843965	0.838349	0.191944	0.876494
R3	0.938047	0.778409	0.326198	0.424893	0.802749
R4	0.769201	0.705776	0.206219	0.002919	0.471366

```
df_num2=pd.DataFrame(np.random.rand(3,3),index=["R3","R4","R5"],columns=["C1","C2","C3"])
```

```
df_num2
```

	C1	C2	C3
R3	0.440232	0.016451	0.867797
R4	0.668665	0.481141	0.494440
R5	0.559501	0.769926	0.552914

```
df_num2+df_num
```

	C1	C2	C3	C4	C5
R1	NaN	NaN	NaN	NaN	NaN
R2	NaN	NaN	NaN	NaN	NaN
R3	1.378279	0.794859	1.193995	NaN	NaN
R4	1.437866	1.186918	0.700659	NaN	NaN
R5	NaN	NaN	NaN	NaN	NaN

```
df_num2-df_num
```

	C1	C2	C3	C4	C5
R1	NaN	NaN	NaN	NaN	NaN
R2	NaN	NaN	NaN	NaN	NaN
R3	-0.497815	-0.761958	0.541600	NaN	NaN
R4	-0.100536	-0.224635	0.288221	NaN	NaN
R5	NaN	NaN	NaN	NaN	NaN

```
df3=pd.DataFrame([["Python","High",2],["C++","Low",1],["Java","Low",3]],index=["R1","R2","R3"],col
```

```
df3
```

	C1	C2	C3
R1	Python	High	2
R2	C++	Low	1
R3	Java	Low	3

```
df_num+df3##integer cannot be added with string nor string cannot be concatenated with numeric val
```

```
#pandas -I/O Tools
#read_csv -- to read .csv files
#read_excel -- to read .xls,xlsx etc files
```

```
# In your rec_csv function we have ana argument termed  sep-- to mention the seperator
```

```
Start coding or generate with AI.
```

```
points_table=pd.read_csv("C:\\Users\\jobin jose\\OneDrive\\Desktop\\NTTF AI ML\\day7\\points_table
```

```
import pandas as pd
```

```
points_table
```

	season	rank	name	short_name	matchesplayed	matcheswon	matcheslost	noresult	matchpoints	nrr	for	against
0	2023	1	Gujarat Titans	GT	14	10	4	0	20	0.809	2450/268.1	2326/279.2
1	2023	2	Chennai Super Kings	CSK	14	8	5	1	17	0.652	2369/254.3	2232/257.5
2	2023	3	Lucknow Super Giants	LSG	14	8	5	1	17	0.284	2253/255.2	2216/259.3
3	2023	4	Mumbai Indians	MI	14	8	6	0	16	-0.044	2592/270.3	2620/272.1
4	2023	5	Rajasthan Royals	RR	14	7	7	0	14	0.148	2419/272.1	2389/273.2
...	...	...	...	...	...	...	...	...	...	...	...	...
131	2008	4	Delhi Capitals	DC	14	7	6	1	15	0.342	2001/233.2	2031/246.4
132	2008	5	Mumbai Indians	MI	14	7	7	0	14	0.570	2080/249.1	2096/269.3
133	2008	6	Kolkata Knight Riders	KKR	14	6	7	1	13	-0.147	1845/242.4	1718/221.4

```
language=pd.read_excel("C:\\Users\\jobin jose\\OneDrive\\Desktop\\NTTF AI ML\\day7\\Book1.xlsx",he
```

```
language
```

	0	1	2
0	1	Python	High
1	2	Java	low
2	3	c++	Expert
3	4	Excel	noob

```
language.columns=["Sl.No","Language","Level"]
```

```
language
```

	Sl.No	Language	Level
0	1	Python	High
1	2	Java	low
2	3	c++	Expert
3	4	Excel	noob

```
points_table.dtypes#quickly get all the data types of your DataFrame
```

```
season            int64
rank              int64
name             object
short_name       object
matchesplayed     int64
matcheswon        int64
matcheslost       int64
noresult          int64
matchpoints       int64
nrr             float64
for              object
against          object
dtype: object
```

```
dict_ot_types={"season":np.float32}
```

```
import numpy as np
```

```
points_table=pd.read_csv("C:\\Users\\jobin jose\\OneDrive\\Desktop\\NTTF AI ML\\day7\\points_table
```

```
points_table.dtypes
```

```
season          float32
rank              int64
name             object
short_name       object
matchesplayed     int64
matcheswon        int64
matcheslost       int64
noresult          int64
matchpoints       int64
nrr             float64
for              object
against          object
dtype: object
```

```
points_table_2=pd.read_csv("C:\\Users\\jobin jose\\OneDrive\\Desktop\\NTTF AI ML\\day7\\points_tab
```

```
points_table_2
```

	2023	2	Chennai Super Kings	CSK	14	8	5	1	17	0.652	2369/254.3	2232/257.5
0	2023	3	Lucknow Super Giants	LSG	14	8	5	1	17	0.284	2253/255.2	2216/259.3
1	2023	4	Mumbai Indians	MI	14	8	6	0	16	-0.044	2592/270.3	2620/272.1
2	2023	5	Rajasthan Royals	RR	14	7	7	0	14	0.148	2419/272.1	2389/273.2
3	2023	6	Royal Challengers Bangalore	RCB	14	7	7	0	14	0.135	2502/275.4	2435/272.2
4	2023	7	Kolkata Knight Riders	KKR	14	6	8	0	12	-0.239	2463/274.3	2432/264.0
...	...	...	...	...	...	...	...	...	...	...	...	...
129	2008	4	Delhi Capitals	DC	14	7	6	1	15	0.342	2001/233.2	2031/246.4
130	2008	5	Mumbai Indians	MI	14	7	7	0	14	0.570	2080/249.1	2096/269.3
131	2008	6	Kolkata Knight Riders	KKR	14	6	7	1	13	-0.147	1845/242.4	1718/221.4
132	2008	7	Royal Challengers Bangalore	RCB	14	4	10	0	8	-1.160	1983/272.4	2205/261.3
133	2008	8	Deccan Chargers	SRH	14	2	12	0	4	-0.467	2229/270.0	2307/264.3

134 rows × 12 columns

```
data_string = """Name:Gender: Age
Braund: male: 22
Cumings: female:38
Heikkinen: female: 26
Futrelle: female: 35"""
```

```
from io import StringIO
#Converting the above string data into a file-like object
obj = StringIO(data_string)
df = pd.read_csv(obj,delimiter=":")
df
```

	Name	Gender	Age
0	Braund	male	22
1	Cumings	female	38
2	Heikkinen	female	26
3	Futrelle	female	35

```
df.to_csv("Sample_name.csv")
```

```
table_df=pd.read_table("C:\\Users\\jobin jose\\OneDrive\\Desktop\\NTTF AI ML\\day7\\Sample_name.cs
```

```
print(table_df)
```

```
   Unnamed: 0       Name  Gender  Age
0           0     Braund    male   22
1           1    Cumings  female   38
2           2  Heikkinen  female   26
3           3   Futrelle  female   35
```

```
table_df.drop(columns="Unnamed: 0",inplace=True)
```

```
table_df
```

	Name	Gender	Age
0	Braund	male	22
1	Cumings	female	38
2	Heikkinen	female	26
3	Futrelle	female	35

```
data_json="""[
{"Name": "Braund", "Gender": "Male", "Age": 30},
{"Name": "Cumings", "Gender": "Female", "Age": 25},
{"Name": "Heikkinen", "Gender": "Female", "Age": 35}
]"""#the string must be in proper json format enclosed within []
```

```
obj=StringIO(data_json)#StringIO converts the string into a string object
df=pd.read_json(obj)#it reads the string object as a json object
```

```
df
```

	Name	Gender	Age
0	Braund	Male	30
1	Cumings	Female	25
2	Heikkinen	Female	35

```
hel="""[{
"Name": "Flash",
"Colour":"Black",
"Number":2,
"Boolean": false
}]"""
```

```
obj2=StringIO(hel)
df2=pd.read_json(obj2)
```

```
df2
```

```
    Name  Colour  Number  Boolean
0  Flash   Black       2    False
```

```
Start coding or generate with AI.
```