```
#Importing Required Libraries
import pandas as pd


#Reding excel file
df = pd.read_csv('result.csv')
df
```

	Name	Role	Team	Cost
0	R_Sharma	Batsman	Mumbai Indians	9.5
1	S_Yadab	Batsman	Mumbai Indians	10.0
2	R_Gaikwad	Batsman	Chennai Super Kings	9.0
3	Ishan Kishan	Wkt-Kepper	Mumbai Indians	8.5
4	M_ali	All-Rounder	Chennai Super Kings	8.5
5	J_Bumrah	Bowler	Mumbai Indians	9.0
6	M_Theekshna	Bowler	Chennai Super Kings	8.0
7	D_Chahar	Bowler	Chennai Super Kings	8.5
8	D_Conway	Batsman	Chennai Super Kings	8.5
9	T_David	All-Rounder	Mumbai Indians	8.0
10	C_Green	All-Rounder	Mumbai Indians	9.5

```
#Grouping by role og player
RoleGroup = df.groupby('Role')
#first()  mtehod is used to print first entry from each group
RoleGroup.first()
```

Role	Name	Team	Cost
All-Rounder	M_ali	Chennai Super Kings	8.5
Batsman	R_Sharma	Mumbai Indians	9.5
Bowler	J_Bumrah	Mumbai Indians	9.0
Wkt-Kepper	Ishan Kishan	Mumbai Indians	8.5

```
#count() method is used to count total number of groups
RoleGroup.count()
```

Role	Name	Team	Cost
All-Rounder	3	3	3
Batsman	4	4	4
Bowler	3	3	3
Wkt-Kepper	1	1	1

```
#Splitting Data intomultiple groups
Role_Filter = df['Role'] =='Batsman'
df[Role_Filter]
```

	Name	Role	Team	Cost
0	R_Sharma	Batsman	Mumbai Indians	9.5
1	S_Yadab	Batsman	Mumbai Indians	10.0
2	R_Gaikwad	Batsman	Chennai Super Kings	9.0
8	D_Conway	Batsman	Chennai Super Kings	8.5

```
Role_Filter = df['Role'] =='Bowler'
df[Role_Filter]
```

	Name	Role	Team	Cost
5	J_Bumrah	Bowler	Mumbai Indians	9.0
6	M_Theekshna	Bowler	Chennai Super Kings	8.0
7	D_Chahar	Bowler	Chennai Super Kings	8.5

```
Role_Filter = df['Role'] =='All-Rounder'
df[Role_Filter]
```

	Name	Role	Team	Cost
4	M_ali	All-Rounder	Chennai Super Kings	8.5
9	T_David	All-Rounder	Mumbai Indians	8.0
10	C_Green	All-Rounder	Mumbai Indians	9.5

```
    #Splitting the data and running an aggregation function
Role_Filter = df['Role']=='Batsman'
BatsmanCost = df[Role_Filter]['Cost']
print(BatsmanCost)
print(BatsmanCost.sum())
```

```
0     9.5
1    10.0
2     9.0
8     8.5
Name: Cost, dtype: float64
37.0
```

```
Role_Filter = df['Role']=='Batsman'
BatsmanCost = df[Role_Filter]['Cost'].sum()

Role_Filter = df['Role']=='Bowler'
BowlerCost = df[Role_Filter]['Cost'].sum()

Role_Filter = df['Role']=='All-Rounder'
All_RounderCost = df[Role_Filter]['Cost'].sum()

Role_Filter = df['Role']=='Wkt-Kepper'
Wkt_KepperCost = df[Role_Filter]['Cost'].sum()

print(BatsmanCost , BowlerCost, All_RounderCost, Wkt_KepperCost )
```

```
37.0 25.5 26.0 8.5
```

```
RoleGroup = df.groupby('Role')['Cost'].sum().sort_values(ascending=False)
RoleGroup
```

```
Role
Batsman        37.0
All-Rounder    26.0
Bowler         25.5
Wkt-Kepper      8.5
Name: Cost, dtype: float64
```

```
RoleTeamGroup = df.groupby(['Role', 'Team'])
RoleTeamGroup.first()
```

		Name	Cost
Role	Team		
All-Rounder	Chennai Super Kings	M_ali	8.5
	Mumbai Indians	T_David	8.0
Batsman	Chennai Super Kings	R_Gaikwad	9.0
	Mumbai Indians	R_Sharma	9.5
Bowler	Chennai Super Kings	M_Theekshna	8.0
	Mumbai Indians	J_Bumrah	9.0
Wkt-Kepper	Mumbai Indians	Ishan Kishan	8.5

```
d = pd.read_excel('cricket2.xlsx')
d.head()
```

	id	Sixes	Fours
0	1	3	5
1	2	4	4
2	3	1	3
3	4	2	2
4	5	1	1

```
d.corr()
```

	id	Sixes	Fours
id	1.000000	-0.782624	-0.885061
Sixes	-0.782624	1.000000	0.766358
Fours	-0.885061	0.766358	1.000000

```
#Showing the corelation between  two columns
d['Sixes'].corr(d['Fours'])
```

```
0.7663582481705323
```

```
print(d['Sixes'].count())
```

```
11
```

```
d.describe()
```

	id	Sixes	Fours
count	11.000000	11.000000	11.000000
mean	6.000000	1.272727	1.818182
std	3.316625	1.348400	1.601136
min	1.000000	0.000000	0.000000
25%	3.500000	0.000000	1.000000
50%	6.000000	1.000000	1.000000
75%	8.500000	2.000000	2.500000
max	11.000000	4.000000	5.000000

```
dr = pd.read_excel('Fees_Data.xlsx')
dr
```

	Sr. No	EN No.	Name of the student	Branch	Total Fees	Fees Paid	Date of payment	Mode of Payment	Time	Status
0	1	EN23204195	SHINDE GAURAV PRAKASH	CIVIL	10056.0	10056	27/7/23	UPI	2.45 pm	Yes
1	2	EN23146043	Chaudhari Prasad	IT	NaN	NaN	2023-04-08	UPI	11.40	Yes

```
#Droping repeated valuese from Branch column
DropDuplicates = dr['Branch'].drop_duplicates()
print(DropDuplicates)
```

```
0          CIVIL
1             IT
2            NaN
3       Computer
9     Electrical
11         Civil
24    Mechanical
38            EE
Name: Branch, dtype: object
```

5) *empty Property* :- used to determine if DataFrame is empty or not

68	69	EN23117836	Yadav Divya Dadabhai	Electrical	53778.0	40000	17/8/2015	UPI		Yes

```
#Checking if the Fees_Data.xlsx is empty or not
print(dr.empty)
```

```
False
```

```
#Reading Excel File
Data = pd.read_excel('Fees_Data.xlsx')
Data.head()
```

	Sr. No	EN	Name of the student	Branch	Total Fees	Fees Paid	Date of payment	Mode of Payment	Time	Status
0	1	EN23204195	SHINDE GAURAV PRAKASH	CIVIL	10056.0	10056	27/7/23	UPI	2.45 pm	Yes
1	2	EN23146043	Chaudhari Prasad mahesh	IT	NaN	NaN	2023-04-08 00:00:00	UPI	11.40 am	Yes
2	3	EN23135942	Borse Gunwant Ashok	NaN	53778.0	28778	2023-04-08 00:00:00	Credit Card	11.50 am	No
3	4	EN23135942	Borse Gunwant Ashok	Computer	NaN	25000	2023-04-08 00:00:00	Credit Card	12:00:00	No
4	5	EN23119584	Hire Tejas Ravindra	Computer	53778.0	53778	2023-04-08 00:00:00	UPI	12:10:00	Yes

```
#applying filter of give file
filtered = Data.filter(items=['Sr. No','EN', 'Branch', 'Time'])
print(filtered)
```

```
    Sr. No          EN      Branch      Time
0        1  EN23204195       CIVIL   2.45 pm
1        2  EN23146043          IT  11.40 am
2        3  EN23135942         NaN  11.50 am
3        4  EN23135942    Computer  12:00:00
4        5  EN23119584    Computer  12:10:00
..     ...         ...         ...       ...
66      67  EN23273933       Civil  17:00:00
67      68  EN23200937       Civil  11.58 am
68      69  EN23117836  Electrical  11.40 am
69      70  EN23247685       Civil  11.46 am
70      71  EN23237346       Civil  12.00 pm

[71 rows x 4 columns]
```

7) *equals() method* : - Compare two DataFrames to determine if they are equal or not

```
#Copying above file and storing to NewData
NewData = Data.copy()
NewData.head()
```

	Sr. No	EN	Name of the student	Branch	Total Fees	Fees Paid	Date of payment	Mode of Payment	Time	Status
0	1	EN23204195	SHINDE GAURAV PRAKASH	CIVIL	10056.0	10056	27/7/23	UPI	2.45 pm	Yes
1	2	EN23146043	Chaudhari Prasad mahesh	IT	NaN	NaN	2023-04-08 00:00:00	UPI	11.40 am	Yes
2	3	EN23135942	Borse Gunwant Ashok	NaN	53778.0	28778	2023-04-08 00:00:00	Credit Card	11.50 am	No
3	4	EN23135942	Borse Gunwant Ashok	Computer	NaN	25000	2023-04-08 00:00:00	Credit Card	12:00:00	No
4	5	EN23110584	Hire Tejas Ravindra	Computer	53778.0	53778	2023-04-08	UPI	12:10:00	Yes

```
#Checking if  Data and NewData are equal or not
print(Data.equals(NewData))
```

```
True
```

```
Data1 = pd.read_excel('Book1.xlsx')
Data1.head()
```

	Sr.	EnNo.	Name of Student	CA-I (10marks)	CA-II (10marks)	Midterm (20marks)	UT1	OEA	Midsem split
0	1	2054491246001	AAKANKSHA ANIL SALUNKE	8	8	18	6	8	6
1	2	2054491246002	ABHINASH KAILASH JOSHI	8	9	18	7	8	4
2	3	2054491246003	ADITYA JITENDRA MALI	6	8	14	6	8	4
3	4	2054491246004	AISHWARYA AVINASH PATIL	9	9	19	8	8	5
4	5	2054491246005	AKSHAY AVINASH PATIL	3	3	9	8	8	5

```
#Cheking if Data And Data1are equal or not
print(Data.equals(Data1))
```

```
False
```