In [3]:

```
import os
os.getcwd()
```

Out[3]:

```
'/home/harshit/Desktop/Edelweiss'
```

In [2]:

```
import pandas as pd
```

In [3]:

```
df=pd.read_csv("titanic.csv") #load the dataset from given path
```

In [7]:

```
df.sample(5) #n number of random samples from data. Default is 1
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
526	527	1	2	Ridsdale, Miss. Lucy	female	50.0	0	0	W./C. 14258	10.5000	NaN	S
814	815	0	3	Tomlin, Mr. Ernest Portage	male	30.5	0	0	364499	8.0500	NaN	S
154	155	0	3	Olsen, Mr. Ole Martin	male	NaN	0	0	Fa 265302	7.3125	NaN	S
514	515	0	3	Coleff, Mr. Satio	male	24.0	0	0	349209	7.4958	NaN	S
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89.1042	C92	C

In [8]:

```
df.head() #rows from top. Default is 5 rows
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [10]:

```
df.tail(3) #rows from end of data set. 5 is default
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [11]:

```
df.info()  #information of columns and data type of each
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Gender       891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [12]:

```
df.columns #column names
```

Out[12]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Gender', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [13]:

```
df.shape #rows followed by column number
```

Out[13]:

```
(891, 12)
```

In [ ]:

```
#Exploratory Data Analysis (EDA)
```

In [14]:

```
df.index #current indexing logic
```

Out[14]:

```
RangeIndex(start=0, stop=891, step=1)
```

In [15]:

```
df #head+tail
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [16]:

```
df[     ['Name' ,'Gender' ,'Age']        ]    #list of column names to see
```

Out[16]:

	Name	Gender	Age
0	Braund, Mr. Owen Harris	male	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	Allen, Mr. William Henry	male	35.0
...	...	...	...
886	Montvila, Rev. Juozas	male	27.0
887	Graham, Miss. Margaret Edith	female	19.0
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN
889	Behr, Mr. Karl Howell	male	26.0
890	Dooley, Mr. Patrick	male	32.0

891 rows × 3 columns

In [17]:

```
df[      ['Name']        ]
```

Out[17]:

	Name
0	Braund, Mr. Owen Harris
1	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	Heikkinen, Miss. Laina
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	Allen, Mr. William Henry
...	...
886	Montvila, Rev. Juozas
887	Graham, Miss. Margaret Edith
888	Johnston, Miss. Catherine Helen "Carrie"
889	Behr, Mr. Karl Howell
890	Dooley, Mr. Patrick

891 rows × 1 columns

In [19]:

```
df[      ['Age','Gender']        ]
```

Out[19]:

	Age	Gender
0	22.0	male
1	38.0	female
2	26.0	female
3	35.0	female
4	35.0	male
...	...	...
886	27.0	male
887	19.0	female
888	NaN	female
889	26.0	male
890	32.0	male

891 rows × 2 columns

In [20]:

```
df.loc[ [0,10,23,45,67] ] #pass indices for row numbers to see
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	8.0500	NaN	S
67	68	0	3	Crease, Mr. Ernest James	male	19.0	0	0	S.P. 3464	8.1583	NaN	S

In [21]:

```
df.loc[ [0,10,23,45,67] ] [['Name','Gender']] #chain commands together as per needs
```

Out[21]:

	Name	Gender
0	Braund, Mr. Owen Harris	male
10	Sandstrom, Miss. Marguerite Rut	female
23	Sloper, Mr. William Thompson	male
45	Rogers, Mr. William John	male
67	Crease, Mr. Ernest James	male

In [23]:

```
df.loc[ [0,10,23,45,67] ] [['Name','Gender']].sample()
```

Out[23]:

	Name	Gender
10	Sandstrom, Miss. Marguerite Rut	female

In [24]:

```
df[['Name','Gender']].loc[[0,10,23,45,67]]
```

Out[24]:

	Name	Gender
0	Braund, Mr. Owen Harris	male
10	Sandstrom, Miss. Marguerite Rut	female
23	Sloper, Mr. William Thompson	male
45	Rogers, Mr. William John	male
67	Crease, Mr. Ernest James	male

In [33]:

```
df[['Age']].head(4).sample()
```

Out[33]:

	Age
0	22.0

In [38]:

```
#records of passengers whose age is less than 25


condition=(df['Age']<25)

condition
```

Out[38]:

```
0       True
1      False
2      False
3      False
4      False
       ...  
886    False
887     True
888    False
889    False
890    False
Name: Age, Length: 891, dtype: bool
```

In [40]:

```
temp=df[condition]
```

In [43]:

```
#records of passengers with age between 20 and 30

condition=(  df['Age'].between(20,31)     ) #range
df[condition]
```

Out[43]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	A/5. 2151	8.0500	NaN	S
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	1	0	345763	18.0000	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167	NaN	S
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.5000	NaN	S
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

264 rows × 12 columns

In [46]:

```
#age to be less than 30 AND gender to be male

condition=(df['Age']<30)&(df['Gender']=='male')
df[condition][['Age','Name']]
```

Out[46]:

	Age	Name
0	22.0	Braund, Mr. Owen Harris
7	2.0	Palsson, Master. Gosta Leonard
12	20.0	Saundercock, Mr. William Henry
16	2.0	Rice, Master. Eugene
23	28.0	Sloper, Mr. William Thompson
...	...	...
877	19.0	Petroff, Mr. Nedelio
883	28.0	Banfield, Mr. Frederick James
884	25.0	Sutehall, Mr. Henry Jr
886	27.0	Montvila, Rev. Juozas
889	26.0	Behr, Mr. Karl Howell

237 rows × 2 columns

In [50]:

```
#multiple conditions together

condition=(df['Age']<30)&(df['Gender']=='male')
df[~condition][['Age','Name','Gender']]
```

Out[50]:

	Age	Name	Gender
1	38.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	26.0	Heikkinen, Miss. Laina	female
3	35.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	35.0	Allen, Mr. William Henry	male
5	NaN	Moran, Mr. James	male
...	...	...	...
882	22.0	Dahlberg, Miss. Gerda Ulrika	female
885	39.0	Rice, Mrs. William (Margaret Norton)	female
887	19.0	Graham, Miss. Margaret Edith	female
888	NaN	Johnston, Miss. Catherine Helen "Carrie"	female
890	32.0	Dooley, Mr. Patrick	male

654 rows × 3 columns

In [47]:

```
#negate the condition by putting ~ before condition

condition=(df['Age']<30)&(df['Gender']=='male')
df[~condition][['Age','Name']]
```

Out[47]:

	Age	Name
1	38.0	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	26.0	Heikkinen, Miss. Laina
3	35.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	35.0	Allen, Mr. William Henry
5	NaN	Moran, Mr. James
...	...	...
882	22.0	Dahlberg, Miss. Gerda Ulrika
885	39.0	Rice, Mrs. William (Margaret Norton)
887	19.0	Graham, Miss. Margaret Edith
888	NaN	Johnston, Miss. Catherine Helen "Carrie"
890	32.0	Dooley, Mr. Patrick

654 rows × 2 columns

In [51]:

```
df[  (df['Age']<30) ]
```

Out[51]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
...	...	...	...	...	...	...	...	...	...	...	...	...
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.5000	NaN	S
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

384 rows × 12 columns

In [52]:

```
df.describe() #show all statistical data
```

Out[52]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

500---->Max----->100 percentage

475(maximum value in a range)---->100--->percentile

50(minimum value)--------->0 percentile

In [53]:

```
#example of calculating quantiles
l1=[1,2,3,45,67,18,20,100,112]
import statistics
statistics.quantiles(l1,n=5,method='inclusive')
```

Out[53]:

```
[2.6, 18.4, 40.0, 80.2]
```

In [56]:

```
df['Age'].describe()
```

Out[56]:

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

In [58]:

```
#mean


df[['Age']].mean()
```

Out[58]:

```
Age    29.699118
dtype: float64
```

In [59]:

```
df[['Age']].std() #standard deviation
```

Out[59]:

```
Age    14.526497
dtype: float64
```

In [60]:

```
df[['Age']].min() #minimum value
```

Out[60]:

```
Age    0.42
dtype: float64
```

In [61]:

```
df[['Age']].max() #maximum value
```

Out[61]:

```
Age    80.0
dtype: float64
```

In [62]:

```
df[['Age','Fare']].max() #two columns together
```

Out[62]:

```
Age      80.0000
Fare    512.3292
dtype: float64
```

In [63]:

```
df[['Age','Fare']].median() #median value. Also same as 50 percentile
```

Out[63]:

```
Age     28.0000
Fare    14.4542
dtype: float64
```

In [67]:

```
df[['Age','Fare']].quantile(0.35) #qualtile with providing the point. 0 - 1 range. 1 means 100 percentile
```

Out[67]:

```
Age     24.0
Fare     9.0
Name: 0.35, dtype: float64
```

In [68]:

```
df[['Age']].mode() #most frequently seen value in age column
```

Out[68]:

	Age
0	24.0

In [69]:

```
def percent90(x):
    return 0.9*x


df['Age'].apply(percent90)
```

Out[69]:

```
0      19.8
1      34.2
2      23.4
3      31.5
4      31.5
       ...
886    24.3
887    17.1
888     NaN
889    23.4
890    28.8
Name: Age, Length: 891, dtype: float64
```

In [66]:

```
#Show name and age of female passengers from Pclass 1 who are 30 or above in age
condition=(df['Age']>=30)&(df['Gender']=='female')&(df['Pclass']==1)
df[condition]
```

Out[66]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	35.64165
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	26.55000
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S	13.27500
52	53	1	1	Harper, Mrs. Henry Sleeper (Myna Haxtun)	female	49.0	1	0	PC 17572	76.7292	D33	C	38.36460
61	62	1	1	Icard, Miss. Amelie	female	38.0	0	0	113572	80.0000	B28	NaN	40.00000
177	178	0	1	Isham, Miss. Ann Elizabeth	female	50.0	0	0	PC 17595	28.7125	C49	C	14.35625
194	195	1	1	Brown, Mrs. James Joseph (Margaret Tobin)	female	44.0	0	0	PC 17610	27.7208	B4	C	13.86040
195	196	1	1	Lurette, Miss. Elise	female	58.0	0	0	PC 17569	146.5208	B80	C	73.26040
215	216	1	1	Newell, Miss. Madeleine	female	31.0	1	0	35273	113.2750	D36	C	56.63750
218	219	1	1	Bazzani, Miss. Albina	female	32.0	0	0	11813	76.2917	D15	C	38.14585
230	231	1	1	Harris, Mrs. Henry Birkhardt (Irene Wallach)	female	35.0	1	0	36973	83.4750	C83	S	41.73750
257	258	1	1	Cherry, Miss. Gladys	female	30.0	0	0	110152	86.5000	B77	S	43.25000
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C	256.16460
268	269	1	1	Graham, Mrs. William Thompson (Edith Junkins)	female	58.0	0	1	PC 17582	153.4625	C125	S	76.73125
269	270	1	1	Bissette, Miss. Amelia	female	35.0	0	0	PC 17760	135.6333	C99	S	67.81665
275	276	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0	1	0	13502	77.9583	D7	S	38.97915
299	300	1	1	Baxter, Mrs. James (Helene DeLaudeniere	female	50.0	0	1	PC 17558	247.5208	B58 B60	C	123.76040

				Chaput)									
309	310	1	1	Francatelli, Miss. Laura Mabel	female	30.0	0	0	PC 17485	56.9292	E36	C	28.46460
318	319	1	1	Wick, Miss. Mary Natalie	female	31.0	0	2	36928	164.8667	C7	S	82.43335
319	320	1	1	Spedden, Mrs. Frederic Oakley (Margaretta Corn...	female	40.0	1	1	16966	134.5000	E34	C	67.25000
325	326	1	1	Young, Miss. Marie Grice	female	36.0	0	0	PC 17760	135.6333	C32	C	67.81665
337	338	1	1	Burns, Miss. Elizabeth Margaret	female	41.0	0	0	16966	134.5000	E40	C	67.25000
366	367	1	1	Warren, Mrs. Frank Manley (Anna Sophia Atkinson)	female	60.0	1	0	110813	75.2500	D37	C	37.62500
380	381	1	1	Bidois, Miss. Rosalie	female	42.0	0	0	PC 17757	227.5250	NaN	C	113.76250
383	384	1	1	Holverson, Mrs. Alexander Oskar (Mary Aline To...	female	35.0	1	0	113789	52.0000	NaN	S	26.00000
412	413	1	1	Minahan, Miss. Daisy E	female	33.0	1	0	19928	90.0000	C78	Q	45.00000
486	487	1	1	Hoyt, Mrs. Frederick Maxfield (Jane Anne Forby)	female	35.0	1	0	19943	90.0000	C93	S	45.00000
496	497	1	1	Eustis, Miss. Elizabeth Mussey	female	54.0	1	0	36947	78.2667	D20	C	39.13335
513	514	1	1	Rothschild, Mrs. Martin (Elizabeth L. Barrett)	female	54.0	1	0	PC 17603	59.4000	NaN	C	29.70000
520	521	1	1	Perreault, Miss. Anne	female	30.0	0	0	12749	93.5000	B73	S	46.75000
523	524	1	1	Hippach, Mrs. Louis Albert (Ida Sophia Fischer)	female	44.0	0	1	111361	57.9792	B18	C	28.98960
537	538	1	1	LeRoy, Miss. Bertha	female	30.0	0	0	PC 17761	106.4250	NaN	C	53.21250
540	541	1	1	Crosby, Miss. Harriet R	female	36.0	0	2	WE/P 5735	71.0000	B22	S	35.50000
556	557	1	1	Duff Gordon, Lady. (Lucille Christiana Sutherl...	female	48.0	1	0	11755	39.6000	A16	C	19.80000
558	559	1	1	Taussig, Mrs. Emil (Tillie Mandelbaum)	female	39.0	1	1	110413	79.6500	E67	S	39.82500
571	572	1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53.0	2	0	11769	51.4792	C101	S	25.73960
577	578	1	1	Silvey, Mrs. William Baird (Alice Munger)	female	39.0	1	0	13507	55.9000	E44	S	27.95000
581	582	1	1	Thayer, Mrs. John Borland (Marian Longstreth M...	female	39.0	1	1	17421	110.8833	C68	C	55.44165
591	592	1	1	Stephenson, Mrs. Walter Bertram (Martha Eustis)	female	52.0	1	0	36947	78.2667	D20	C	39.13335
609	610	1	1	Shutes, Miss. Elizabeth W	female	40.0	0	0	PC 17582	153.4625	C125	S	76.73125
716	717	1	1	Endres, Miss. Caroline Louise	female	38.0	0	0	PC 17757	227.5250	C45	C	113.76250
759	760	1	1	Rothes, the Countess. of (Lucy Noel Martha Dye...	female	33.0	0	0	110152	86.5000	B77	S	43.25000
763	764	1	1	Carter, Mrs. William Ernest (Lucile Polk)	female	36.0	1	2	113760	120.0000	B96 B98	S	60.00000
765	766	1	1	Hogeboom, Mrs. John C (Anna Andrews)	female	51.0	1	0	13502	77.9583	D11	S	38.97915
779	780	1	1	Robert, Mrs. Edward Scott (Elisabeth Walton Mc...	female	43.0	0	1	24160	211.3375	B3	S	105.66875
796	797	1	1	Leader, Dr. Alice (Farnham)	female	49.0	0	0	17465	25.9292	D17	S	12.96460
809	810	1	1	Chambers, Mrs. Norman Campbell (Bertha Griggs)	female	33.0	1	0	113806	53.1000	E8	S	26.55000
820	821	1	1	Hays, Mrs. Charles Melville (Clara Jennings Gr...	female	52.0	1	1	12749	93.5000	B69	S	46.75000
				Stone, Mrs. G									

829	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80.0000	B28	NaN	40.00000
835	836	1	1	Compton, Miss. Sara Rebecca	female	39.0	1	1	PC 17756	83.1583	E49	C	41.57915
842	843	1	1	Serepeca, Miss. Augusta	female	30.0	0	0	113798	31.0000	NaN	C	15.50000
856	857	1	1	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45.0	1	1	36928	164.8667	NaN	S	82.43335
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48.0	0	0	17466	25.9292	D17	S	12.96460
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S	26.27710
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C	41.57915

In [67]:

```
#find the average fare of passengers from Pclass 2 whose embarked value is 'S'
df[    (df['Pclass']==2)&(df['Embarked']=='S')       ]['Fare'].mean()
```

Out[67]:

```
20.327439024390245
```

In [68]:

```
#show first 5 rows which have fare above 500 and pclass 2
df [(df['Fare']>500)&(df['Pclass']==2)].head(5)
```

Out[68]:

PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount

In [73]:

```
pd.read_excel('/home/harshit/Data_excel.xls') #reading from excel file
```

Out[73]:

	STA	Date	Precip	WindGustSpd	MaxTemp	MinTemp	MeanTemp	Snowfall	PoorWeather	YR	...	FB	FTI	ITH	PGT
0	10001	1942-07-01	1.016	NaN	25.555556	22.222222	23.888889	0.0	NaN	42	...	NaN	NaN	NaN	NaN
1	10001	1942-07-02	0	NaN	28.888889	21.666667	25.555556	0.0	NaN	42	...	NaN	NaN	NaN	NaN
2	10001	1942-07-03	2.54	NaN	26.111111	22.222222	24.444444	0.0	NaN	42	...	NaN	NaN	NaN	NaN
3	10001	1942-07-04	2.54	NaN	26.666667	22.222222	24.444444	0.0	NaN	42	...	NaN	NaN	NaN	NaN
4	10001	1942-07-05	0	NaN	26.666667	21.666667	24.444444	0.0	NaN	42	...	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65530	31701	1944-09-02	22.098	NaN	27.777778	19.444444	23.333333	0.0	NaN	44	...	NaN	NaN	NaN	NaN
65531	31701	1944-09-03	14.732	NaN	27.222222	18.333333	22.777778	0.0	NaN	44	...	NaN	NaN	NaN	NaN
65532	31701	1944-09-04	4.064	NaN	26.666667	18.888889	22.777778	0.0	NaN	44	...	NaN	NaN	NaN	NaN
65533	31701	1944-09-05	T	NaN	27.222222	17.777778	22.222222	0.0	NaN	44	...	NaN	NaN	NaN	NaN
65534	31701	1944-09-06	T	NaN	27.777778	19.444444	23.333333	0.0	NaN	44	...	NaN	NaN	NaN	NaN

65535 rows × 31 columns

In [70]:

```
#reading from a link
pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv')
```

Out[70]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [82]:

```
#reading fixed number of rows and setting a desired index column

pd.read_csv('titanic.csv',nrows=100,index_col='PassengerId')
```

Out[82]:

	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId											
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8.0500	NaN	S
97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A5	C
98	1	1	Greenfield, Mr. William Bertram	male	23.0	0	1	PC 17759	63.3583	D10 D12	C
99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.0	0	1	231919	23.0000	NaN	S
100	0	2	Kantor, Mr. Sinai	male	34.0	1	0	244367	26.0000	NaN	S

100 rows × 11 columns

In [85]:

```
#categorical values
df['Gender'].unique()
```

Out[85]:

```
array(['male', 'female'], dtype=object)
```

In [87]:

```
temp=df.groupby( ['Gender'] ) #group data by a column
```

In [88]:

```
temp.groups #show group formed
```

Out[88]:

```
{'female': Int64Index([  1,   2,   3,   8,   9,  10,  11,  14,  15,  18,
             ...
             866, 871, 874, 875, 879, 880, 882, 885, 887, 888],
            dtype='int64', length=314),
 'male': Int64Index([  0,   4,   5,   6,   7,  12,  13,  16,  17,  20,
             ...
             873, 876, 877, 878, 881, 883, 884, 886, 889, 890],
            dtype='int64', length=577)}
```

In [89]:

```
temp.get_group('female') #fetch a group by passing label names
```

Out[89]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C
...	...	...	...	...	...	...	...	...	...	...	...	...
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	0	1	230433	26.0000	NaN	S
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

314 rows × 12 columns

In [90]:

```
temp=df.groupby(['Gender','Pclass']) #group by multiple columns

temp.groups
```

Out[90]:

```
{('female',
  1): Int64Index([  1,   3,  11,  31,  52,  61,  88, 136, 151, 166, 177, 194, 195,
             215, 218, 230, 256, 257, 258, 268, 269, 275, 290, 291, 297, 299,
             306, 307, 309, 310, 311, 318, 319, 325, 329, 334, 337, 341, 356,
             366, 369, 375, 380, 383, 393, 412, 435, 457, 486, 496, 498, 504,
             513, 520, 523, 537, 539, 540, 556, 558, 571, 577, 581, 585, 591,
             609, 627, 641, 669, 689, 700, 708, 710, 716, 730, 742, 759, 763,
             765, 779, 781, 796, 809, 820, 829, 835, 842, 849, 853, 856, 862,
             871, 879, 887],
            dtype='int64'),
 ('female',
  2): Int64Index([  9,  15,  41,  43,  53,  56,  58,  66,  84,  98, 123, 133, 161,
             190, 199, 211, 237, 247, 259, 272, 303, 312, 316, 322, 323, 327,
             345, 346, 357, 387, 389, 399, 416, 417, 426, 427, 432, 437, 440,
             443, 446, 458, 472, 473, 506, 516, 518, 526, 530, 535, 546, 576,
             580, 596, 600, 608, 615, 618, 635, 651, 670, 706, 717, 720, 726,
             747, 750, 754, 772, 774, 801, 854, 865, 866, 874, 880],
            dtype='int64'),
 ('female',
  3): Int64Index([  2,   8,  10,  14,  18,  19,  22,  24,  25,  28,
             ...
             823, 830, 852, 855, 858, 863, 875, 882, 885, 888],
            dtype='int64', length=144),
 ('male',
  1): Int64Index([  6,  23,  27,  30,  34,  35,  54,  55,  62,  64,
             ...
             793, 802, 806, 815, 822, 839, 857, 867, 872, 889],
            dtype='int64', length=122),
 ('male',
  2): Int64Index([ 17,  20,  21,  33,  70,  72,  78,  99, 117, 120,
             ...
             812, 817, 827, 831, 841, 848, 861, 864, 883, 886],
            dtype='int64', length=108),
 ('male',
  3): Int64Index([  0,   4,   5,   7,  12,  13,  16,  26,  29,  36,
             ...
             868, 869, 870, 873, 876, 877, 878, 881, 884, 890],
            dtype='int64', length=347)}
```

In [95]:

```
temp.get_group(('female',3))[['Age']].head(10).mean() #can chain commands as usual
```

Out[95]:

```
Age    20.375
dtype: float64
```

In [103]:

```
#read a specified sheet from excel file
dfex1=pd.read_excel('TitanicSheets.xlsx',sheet_name='P1')
dfex1
```

Out[103]:

	Unnamed: 0	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
1	3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
2	6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
3	11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
4	23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...	...
211	871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
212	872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
213	879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
214	887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
215	889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

216 rows × 13 columns

In [105]:

```
#drop a column. axis 1 is for column. 0 is for rows
dfex1.drop(['Unnamed: 0'],axis=1,inplace=True)
```

In [106]:

```
dfex1
```

Out[106]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
1	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
2	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
3	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
4	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
211	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
212	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
213	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
214	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
215	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

216 rows × 12 columns

In [113]:

```
df['Age'].sort_values() #sort a column values
```

Out[113]:

```
803    0.42
755    0.67
644    0.75
469    0.75
78     0.83
       ...
859     NaN
863     NaN
868     NaN
878     NaN
888     NaN
Name: Age, Length: 891, dtype: float64
```

In [115]:

```
df.sort_values(by='Age',ascending=False) #sort entire data frame by a specified column
```

Out[115]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	A23	S
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	NaN	S
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	NaN	C
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A5	C
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	NaN	Q
...	...	...	...	...	...	...	...	...	...	...	...	...
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	C
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

891 rows × 12 columns

In [116]:

```
df.sort_values(by='Fare',ascending=False) #arrange is descending order
```

Out[116]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
...	...	...	...	...	...	...	...	...	...	...	...	...
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0000	NaN	S
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0000	NaN	S
822	823	0	1	Reuchlin, Jonkheer. John George	male	38.0	0	0	19972	0.0000	NaN	S
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0000	NaN	S
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0000	NaN	S

891 rows × 12 columns

In [119]:

```
#specify Fare in ascending, Age in descending.
df.sort_values(by=['Fare','Age'],ascending=[False,True]).head(3)
```

Out[119]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C

In [120]:

```
temp=pd.read_csv('salary.csv')
```

In [122]:

```
#Second column will be used in case of tie in first
temp.sort_values(by=['Salary'])
```

Out[122]:

	Age	Name	Salary
1	19	XYZ	89000
0	24	ABC	140000
2	35	LMNOP	140000

In [124]:

```
temp.sort_values(by=['Salary','Age'],ascending=[True,False])
```

Out[124]:

	Age	Name	Salary
1	19	XYZ	89000
2	35	LMNOP	140000
0	24	ABC	140000

In [127]:

```
df.loc[0,'Name']='XYZ' #set a specified cell value
```

In [ ]:

```
original=df.copy() #create a proper replica
```

In [132]:

```
df.loc[0,'Pclass']="XYZ"
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    object 
 3   Name         891 non-null    object 
 4   Gender       891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(4), object(6)
memory usage: 83.7+ KB
```

In [130]:

```
df.loc[3,'Gender']='F'
df
```

Out[130]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	XYZ	F	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	F	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [131]:

```
def changes(x):
    if x=='male':
        return 'M'
    if x=='female':
        return 'F'

df['Gender']=df['Gender'].apply(changes)
```

Out[131]:

```
0      None
1         F
2         F
3      None
4         M
       ...
886       M
887       F
888       F
889       M
890       M
Name: Gender, Length: 891, dtype: object
```

In [136]:

```
#set values in rows / columns matching a specified filter condition

condition=(df['Pclass']==3)

df.loc[(df['Pclass']==3),'Pclass']='three'
```

In [137]:

```
df
```

Out[137]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	XYZ	XYZ	F	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	three	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	F	35.0	1	0	113803	53.1000	C123	S
4	5	0	three	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	three	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	three	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [140]:

```
def magic(x):
    return 0.5*x

df['discount']=df['Fare'].apply(magic)
```

In [141]:

```
df
```

Out[141]:

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
0	1	0	XYZ	XYZ	F	22.0	1	0	A/5 21171	7.2500	NaN	S	3.62500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	35.64165
2	3	1	three	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	3.96250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	F	35.0	1	0	113803	53.1000	C123	S	26.55000
4	5	0	three	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	4.02500
...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	6.50000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	15.00000
888	889	0	three	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	11.72500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	15.00000
890	891	0	three	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	3.87500

891 rows × 13 columns

In [143]:

```
def name_changer(x):
    return x.upper() #string to upper case

df.insert(3,'UPPER NAMES',df['Name'].apply(name_changer))
```

In [144]:

```
df
```

Out[144]:

	PassengerId	Survived	Pclass	UPPER NAMES	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
0	1	0	XYZ	XYZ	XYZ	F	22.0	1	0	A/5 21171	7.2500	NaN	S	3.62500
1	2	1	1	CUMINGS, MRS. JOHN BRADLEY (FLORENCE BRIGGS TH...	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	35.64165
2	3	1	three	HEIKKINEN, MISS. LAINA	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	3.96250
3	4	1	1	FUTRELLE, MRS. JACQUES HEATH (LILY MAY PEEL)	Futrelle, Mrs. Jacques Heath (Lily May Peel)	F	35.0	1	0	113803	53.1000	C123	S	26.55000
4	5	0	three	ALLEN, MR. WILLIAM HENRY	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	4.02500
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	MONTVILA, REV. JUOZAS	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S	6.50000
887	888	1	1	GRAHAM, MISS. MARGARET EDITH	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S	15.00000
888	889	0	three	JOHNSTON, MISS. CATHERINE HELEN "CARRIE"	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	11.72500
889	890	1	1	BEHR, MR. KARL HOWELL	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C	15.00000
890	891	0	three	DOOLEY, MR. PATRICK	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q	3.87500

891 rows × 14 columns

In [145]:

```
ans=df.drop('UPPER NAMES',axis=1,inplace=True) #not correct approach. Nothing will be present inside ans
```

In [154]:

```
col1=df.pop('discount')
```

In [155]:

```
col1
```

Out[155]:

```
0       3.62500
1      35.64165
2       3.96250
3      26.55000
4       4.02500
         ...   
886     6.50000
887    15.00000
888    11.72500
889    15.00000
890     3.87500
Name: discount, Length: 891, dtype: float64
```

In [156]:

```
df.head(3)[['PassengerId','Survived','Pclass']]
```

Out[156]:

	PassengerId	Survived	Pclass
0	1	0	XYZ
1	2	1	1
2	3	1	three

In [157]:

```
df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]
```

Out[157]:

	PassengerId	Survived	Pclass
67	68	0	three
96	97	0	1
106	107	1	three

In [20]:

```
import timeit


start=timeit.default_timer()

# df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]

df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]
df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]
df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]
df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]
df.loc[   [67,96,106]  ,  ['PassengerId','Survived','Pclass']      ]

end=timeit.default_timer()

print(end-start)
```

```
0.008157037998898886
```

In [21]:

```
import timeit


start=timeit.default_timer()

df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]
df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]
df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]
df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]
df.loc[   [67,96,106] ] [['PassengerId','Survived','Pclass'] ]


end=timeit.default_timer()

print(end-start)
```

```
0.012378177998471074
```

In [23]:

```
def magic(x):
    return 0.5*x

df['discount']=df['Fare'].apply(magic)
```

In [ ]:

```
df.to_csv('modified.csv')
```

In [ ]:

```
df.to_csv('modified.csv',index=False])
```

In [27]:

```
df.to_csv('modified.csv',index=False,columns=['Name','Age','discount'])
```

In [38]:

```
# df.to_csv('modified.csv',df['Age']>30,columns=['Name','Age','discount']) #rows?????
```

In [30]:

```
df.to_csv('modified.txt',columns=['Name','Age','discount'],sep="\t") #rows?????
```

Now we will work with EXCEL FILES

In [32]:

```
df.to_excel('analysis.xls')
```

In [33]:

```
df.to_excel('analysis.xls',index=False)
```

In [ ]:

```
df.to_excel('analysis.xls',index=False,columns=['Name','Age'])
```

In [34]:

```
df.to_excel('analysis.xls',index=False,columns=['Name','Age'],sheet_name='DEMO SHEET')
```

In [41]:

```
df.to_excel('analysis.xls',index=False,columns=['Name','Age'],\
            sheet_name='DEMO SHEET',\
            header=False)
```

In [35]:

```
#WORKING WITH MULTIPLE SHEETS IN SAME FILE



writer=pd.ExcelWriter('analysis.xls')

df.to_excel(writer,index=False,columns=['Name','Age'],sheet_name='DEMO SHEET')
df.to_excel(writer,index=False,columns=['Fare','Pclass'],sheet_name='SECOND SHEET')

writer.save()
```

In [45]:

```
#set date as index column . parse to datetimeindex format for time-series, resampling operation
ydf=pd.read_csv('/home/harshit/Downloads/YESBANK.csv',index_col='Date',parse_dates=True)
ydf
```

Out[45]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-16	334.000000	338.500000	328.000000	333.899994	319.873657	470267.0
2018-01-17	336.000000	343.750000	331.250000	342.500000	328.112457	653618.0
2018-01-18	350.000000	356.500000	333.100006	340.250000	325.956970	2419109.0
2018-01-19	348.000000	352.000000	339.250000	348.299988	333.668793	1659646.0
2018-01-22	349.000000	358.000000	349.000000	355.250000	340.326874	663569.0
...	...	...	...	...	...	...
2020-01-09	47.150002	48.450001	46.299999	47.299999	47.299999	6835915.0
2020-01-10	47.599998	48.349998	43.900002	44.799999	44.799999	15918973.0
2020-01-13	43.400002	44.000000	41.200001	42.099998	42.099998	10763969.0
2020-01-14	41.750000	41.750000	36.549999	38.549999	38.549999	18250917.0
2020-01-15	38.549999	41.099998	36.650002	39.799999	39.799999	19876620.0

489 rows × 6 columns

In [46]:

```
ydf.sample()
```

Out[46]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2019-12-26	51.0	51.599998	48.25	48.700001	48.700001	11357799.0

In [47]:

```
ydf.head()
```

Out[47]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-16	334.0	338.50	328.000000	333.899994	319.873657	470267.0
2018-01-17	336.0	343.75	331.250000	342.500000	328.112457	653618.0
2018-01-18	350.0	356.50	333.100006	340.250000	325.956970	2419109.0
2018-01-19	348.0	352.00	339.250000	348.299988	333.668793	1659646.0
2018-01-22	349.0	358.00	349.000000	355.250000	340.326874	663569.0

In [48]:

```
ydf.tail()
```

Out[48]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-01-09	47.150002	48.450001	46.299999	47.299999	47.299999	6835915.0
2020-01-10	47.599998	48.349998	43.900002	44.799999	44.799999	15918973.0
2020-01-13	43.400002	44.000000	41.200001	42.099998	42.099998	10763969.0
2020-01-14	41.750000	41.750000	36.549999	38.549999	38.549999	18250917.0
2020-01-15	38.549999	41.099998	36.650002	39.799999	39.799999	19876620.0

In [49]:

```
ydf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 489 entries, 2018-01-16 to 2020-01-15
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Open       488 non-null    float64
 1   High       488 non-null    float64
 2   Low        488 non-null    float64
 3   Close      488 non-null    float64
 4   Adj Close  488 non-null    float64
 5   Volume     488 non-null    float64
dtypes: float64(6)
memory usage: 26.7 KB
```

In [50]:

```
ydf.index
```

Out[50]:

```
DatetimeIndex(['2018-01-16', '2018-01-17', '2018-01-18', '2018-01-19',
               '2018-01-22', '2018-01-23', '2018-01-24', '2018-01-25',
               '2018-01-29', '2018-01-30',
               ...
               '2020-01-02', '2020-01-03', '2020-01-06', '2020-01-07',
               '2020-01-08', '2020-01-09', '2020-01-10', '2020-01-13',
               '2020-01-14', '2020-01-15'],
              dtype='datetime64[ns]', name='Date', length=489, freq=None)
```

In [53]:

```
ydf.loc['2018-01'] #locate data for a month
```

Out[53]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-16	334.000000	338.500000	328.000000	333.899994	319.873657	470267.0
2018-01-17	336.000000	343.750000	331.250000	342.500000	328.112457	653618.0
2018-01-18	350.000000	356.500000	333.100006	340.250000	325.956970	2419109.0
2018-01-19	348.000000	352.000000	339.250000	348.299988	333.668793	1659646.0
2018-01-22	349.000000	358.000000	349.000000	355.250000	340.326874	663569.0
2018-01-23	358.000000	360.500000	352.200012	358.850006	343.775604	659804.0
2018-01-24	359.000000	366.000000	355.500000	364.799988	349.475677	773702.0
2018-01-25	365.000000	365.000000	356.500000	363.500000	348.230286	892824.0
2018-01-29	360.000000	363.750000	355.600006	357.549988	342.530212	346356.0
2018-01-30	355.549988	360.850006	352.000000	353.450012	338.602448	3995380.0
2018-01-31	354.000000	356.700012	350.600006	354.450012	339.560455	588375.0

In [55]:

```
ydf.loc['2018-01':'2018-04',  ['High','Low']   ] #a range of months can be given with columns specified
```

Out[55]:

	High	Low
Date		
2018-01-16	338.500000	328.000000
2018-01-17	343.750000	331.250000
2018-01-18	356.500000	333.100006
2018-01-19	352.000000	339.250000
2018-01-22	358.000000	349.000000
...	...	...
2018-04-24	328.649994	313.049988
2018-04-25	328.350006	317.799988
2018-04-26	361.200012	323.700012
2018-04-27	368.750000	345.450012
2018-04-30	367.000000	349.500000

70 rows × 2 columns

In [56]:

```
ydf.loc['2018':'2019',  ['High','Low']   ]
```

Out[56]:

Date	High	Low
2018-01-16	338.500000	328.000000
2018-01-17	343.750000	331.250000
2018-01-18	356.500000	333.100006
2018-01-19	352.000000	339.250000
2018-01-22	358.000000	349.000000
...	...	...
2019-12-24	52.000000	50.049999
2019-12-26	51.599998	48.250000
2019-12-27	49.799999	47.700001
2019-12-30	48.950001	46.700001
2019-12-31	48.049999	46.349998

478 rows × 2 columns

In [58]:

```
#resample on monthly basis by aggregating by mean
ydf.resample('M').mean()
```

Out[58]:

Date	Open	High	Low	Close	Adj Close	Volume
2018-01-31	351.686363	356.504547	345.727275	352.072726	337.283039	1.192968e+06
2018-02-28	328.421053	333.505262	321.950001	327.439472	313.684572	7.938088e+05
2018-03-31	308.789475	313.086840	302.857897	307.563159	294.643211	1.012892e+06
2018-04-30	318.004761	324.290478	313.283332	318.947622	305.549444	1.369852e+06
2018-05-31	345.063636	349.731818	339.627269	343.743185	329.303404	4.985085e+05
2018-06-30	336.416670	339.597621	332.302383	335.552383	326.209441	6.876987e+05
2018-07-31	369.254543	374.409091	363.595455	369.061360	359.196929	6.939900e+05
2018-08-31	378.240481	382.980953	371.514287	376.373811	366.313921	8.888597e+05
2018-09-30	298.116668	302.250002	282.369444	288.986114	281.261965	4.330388e+06
2018-10-31	214.819047	224.028572	206.592858	214.480953	208.748212	4.504789e+06
2018-11-30	201.707500	206.147498	195.022499	200.200000	194.848964	6.249109e+06
2018-12-31	178.344999	182.005000	173.815000	177.987501	173.230172	5.050041e+06
2019-01-31	197.269565	202.384782	191.178262	195.823912	190.589844	5.298535e+06
2019-02-28	201.776316	207.402631	196.710527	202.763157	197.343615	5.538068e+06
2019-03-31	244.702779	249.063887	241.744444	246.038890	239.462652	3.042696e+06
2019-04-30	260.286843	263.421053	251.671051	254.807895	247.997274	3.623291e+06
2019-05-31	154.336363	158.129544	149.145455	152.688637	148.607509	5.513620e+06
2019-06-30	124.673684	127.686843	119.789475	122.992105	122.886842	6.574336e+06
2019-07-31	94.208695	97.521740	90.371739	93.467391	93.467391	9.584150e+06
2019-08-31	75.417500	77.410000	70.595000	73.172500	73.172500	1.348925e+07
2019-09-30	60.271053	62.242105	57.239473	59.165789	59.165789	1.594390e+07
2019-10-31	46.371053	49.955263	43.534211	47.134211	47.134211	2.839263e+07
2019-11-30	67.650000	70.052500	65.339999	67.132500	67.132500	2.177805e+07
2019-12-31	52.407143	54.116667	49.871428	51.452381	51.452381	1.837017e+07
2020-01-31	45.259092	46.236363	43.522728	44.536363	44.536363	9.578793e+06

In [60]:

```
ydf.loc['2018-01'].resample('W').mean() #locating data for a range and then resampling
```

Out[60]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-21	342.000000	347.687500	332.900002	341.237496	326.902969	1.300660e+06
2018-01-28	357.750000	362.375000	353.300003	360.599998	345.452110	7.474748e+05
2018-02-04	356.516663	360.433339	352.733337	355.150004	340.231038	1.643370e+06

In [63]:

```
ydf.resample('Y').sum()
```

Out[63]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-12-31	70854.800108	72030.300063	69302.200072	70537.500150	68182.888091	5.340901e+08
2019-12-31	31640.150010	32448.999933	30584.200025	31378.550020	30815.636235	2.750862e+09
2020-12-31	497.850008	508.599998	478.750005	489.899992	489.899992	1.053667e+08

In [65]:

```
#aggregate each column with separate functions

ydf.resample('M').agg({'Open':'sum', 'Low':'mean', 'Volume':'max'  })
```

Out[65]:

	Open	Low	Volume
Date			
2018-01-31	3868.549988	345.727275	3995380.0
2018-02-28	6240.000000	321.950001	2826458.0
2018-03-31	5867.000030	302.857897	2468765.0
2018-04-30	6678.099976	313.283332	3867851.0
2018-05-31	7591.399994	339.627269	855300.0
2018-06-30	7064.750060	332.302383	5312720.0
2018-07-31	8123.599946	363.595455	1839314.0
2018-08-31	7943.050108	371.514287	4974124.0
2018-09-30	5366.100021	282.369444	20785936.0
2018-10-31	4511.199997	206.592858	8444783.0
2018-11-30	4034.150009	195.022499	25791984.0
2018-12-31	3566.899979	173.815000	10816982.0
2019-01-31	4537.199997	191.178262	14527540.0
2019-02-28	3833.750000	196.710527	19498115.0
2019-03-31	4404.650024	241.744444	7038240.0
2019-04-30	4945.450012	251.671051	20646532.0
2019-05-31	3395.399994	149.145455	9989831.0
2019-06-30	2368.799988	119.789475	14285099.0
2019-07-31	2166.799987	90.371739	15911076.0
2019-08-31	1508.349995	70.595000	24065313.0
2019-09-30	1145.150010	57.239473	33140827.0
2019-10-31	881.050009	43.534211	62620476.0
2019-11-30	1353.000000	65.339999	45824409.0
2019-12-31	1100.549994	49.871428	43998321.0
2020-01-31	497.850008	43.522728	19876620.0

In [ ]:

```
#SUM resampled data for months of JANUARY FEBRUARY MARCH 2019
```

In [70]:

```
df.set_index('Name',inplace=True) #set a index manually
df
```

Out[70]:

	PassengerId	Survived	Pclass	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
Name												
Braund, Mr. Owen Harris	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S	3.62500
Cumings, Mrs. John Bradley (Florence Briggs Thayer)	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	C	35.64165
Heikkinen, Miss. Laina	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	3.96250
Futrelle, Mrs. Jacques Heath (Lily May Peel)	4	1	1	female	35.0	1	0	113803	53.1000	C123	S	26.55000
Allen, Mr. William Henry	5	0	3	male	35.0	0	0	373450	8.0500	NaN	S	4.02500
...	...	...	...	...	...	...	...	...	...	...	...	...
Montvila, Rev. Juozas	887	0	2	male	27.0	0	0	211536	13.0000	NaN	S	6.50000
Graham, Miss. Margaret Edith	888	1	1	female	19.0	0	0	112053	30.0000	B42	S	15.00000
Johnston, Miss. Catherine Helen "Carrie"	889	0	3	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	11.72500
Behr, Mr. Karl Howell	890	1	1	male	26.0	0	0	111369	30.0000	C148	C	15.00000
Dooley, Mr. Patrick	891	0	3	male	32.0	0	0	370376	7.7500	NaN	Q	3.87500

891 rows × 12 columns

In [73]:

```
df.reset_index(inplace=True) #reset to a previous index
df
```

Out[73]:

	Name	PassengerId	Survived	Pclass	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
0	Braund, Mr. Owen Harris	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S	3.62500
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	C	35.64165
2	Heikkinen, Miss. Laina	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	3.96250
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	4	1	1	female	35.0	1	0	113803	53.1000	C123	S	26.55000
4	Allen, Mr. William Henry	5	0	3	male	35.0	0	0	373450	8.0500	NaN	S	4.02500
...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	Montvila, Rev. Juozas	887	0	2	male	27.0	0	0	211536	13.0000	NaN	S	6.50000
887	Graham, Miss. Margaret Edith	888	1	1	female	19.0	0	0	112053	30.0000	B42	S	15.00000
888	Johnston, Miss. Catherine Helen "Carrie"	889	0	3	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	11.72500
889	Behr, Mr. Karl Howell	890	1	1	male	26.0	0	0	111369	30.0000	C148	C	15.00000
890	Dooley, Mr. Patrick	891	0	3	male	32.0	0	0	370376	7.7500	NaN	Q	3.87500

891 rows × 13 columns

In [74]:

```
#check for a value in a range of options in a filter

values=[22.0,26.0,32.0,19.0]
condition=(df['Age'].isin(values))

df[ condition  ]
```

Out[74]:

	Name	PassengerId	Survived	Pclass	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
0	Braund, Mr. Owen Harris	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S	3.62500
2	Heikkinen, Miss. Laina	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	3.96250
27	Fortune, Mr. Charles Alexander	28	0	1	male	19.0	3	2	19950	263.0000	C23 C25 C27	S	131.50000
44	Devaney, Miss. Margaret Delia	45	1	3	female	19.0	0	0	330958	7.8792	NaN	Q	3.93960
60	Sirayanian, Mr. Orsen	61	0	3	male	22.0	0	0	2669	7.2292	NaN	C	3.61460
...	...	...	...	...	...	...	...	...	...	...	...	...	...
877	Petroff, Mr. Nedelio	878	0	3	male	19.0	0	0	349212	7.8958	NaN	S	3.94790
882	Dahlberg, Miss. Gerda Ulrika	883	0	3	female	22.0	0	0	7552	10.5167	NaN	S	5.25835
887	Graham, Miss. Margaret Edith	888	1	1	female	19.0	0	0	112053	30.0000	B42	S	15.00000
889	Behr, Mr. Karl Howell	890	1	1	male	26.0	0	0	111369	30.0000	C148	C	15.00000
890	Dooley, Mr. Patrick	891	0	3	male	32.0	0	0	370376	7.7500	NaN	Q	3.87500

88 rows × 13 columns

In [75]:

```
age=[22.0,26.0,32.0,19.0]
pclass=[1,2]
condition=(df['Age'].isin(values))  &  (df['Pclass'].isin(pclass))

df[ condition  ]
```

Out[75]:

	Name	PassengerId	Survived	Pclass	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	discount
27	Fortune, Mr. Charles Alexander	28	0	1	male	19.0	3	2	19950	263.0000	C23 C25 C27	S	131.50000
70	Jenkin, Mr. Stephen Curnow	71	0	2	male	32.0	0	0	C.A. 33111	10.5000	NaN	S	5.25000
136	Newsom, Miss. Helen Monypeny	137	1	1	female	19.0	0	2	11752	26.2833	D47	S	13.14165
145	Nicholls, Mr. Joseph Charles	146	0	2	male	19.0	1	1	C.A. 33112	36.7500	NaN	S	18.37500
151	Pears, Mrs. Thomas (Edith Wearne)	152	1	1	female	22.0	1	0	113776	66.6000	C2	S	33.30000
190	Pinsky, Mrs. (Rosa)	191	1	2	female	32.0	0	0	234604	13.0000	NaN	S	6.50000
191	Carbines, Mr. William	192	0	2	male	19.0	0	0	28424	13.0000	NaN	S	6.50000
218	Bazzani, Miss. Albina	219	1	1	female	32.0	0	0	11813	76.2917	D15	C	38.14585
226	Mellors, Mr. William John	227	1	2	male	19.0	0	0	SW/PP 751	10.5000	NaN	S	5.25000
238	Pengelly, Mr. Frederick William	239	0	2	male	19.0	0	0	28665	10.5000	NaN	S	5.25000
290	Barber, Miss. Ellen "Nellie"	291	1	1	female	26.0	0	0	19877	78.8500	NaN	S	39.42500
291	Bishop, Mrs. Dickinson H (Helen Walton)	292	1	1	female	19.0	1	0	11967	91.0792	B49	C	45.53960
312	Lahtinen, Mrs. William (Anna Sylfven)	313	0	2	female	26.0	1	1	250651	26.0000	NaN	S	13.00000
323	Caldwell, Mrs. Albert Francis (Sylvia Mae Harb...	324	1	2	female	22.0	1	1	248738	29.0000	NaN	S	14.50000
356	Bowerman, Miss. Elsie Edith	357	1	1	female	22.0	0	1	113505	55.0000	E33	S	27.50000
373	Ringhini, Mr. Sante	374	0	1	male	22.0	0	0	PC 17760	135.6333	NaN	C	67.81665
427	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	428	1	2	female	19.0	0	0	250655	26.0000	NaN	S	13.00000
539	Frolicher, Miss. Hedwig Margaritha	540	1	1	female	22.0	0	2	13568	49.5000	B39	C	24.75000
543	Beane, Mr. Edward	544	1	2	male	32.0	1	0	2908	26.0000	NaN	S	13.00000
546	Beane, Mrs. Edward (Ethel Clarke)	547	1	2	female	19.0	1	0	2908	26.0000	NaN	S	13.00000
608	Laroche, Mrs. Joseph (Juliette Marie Louise La...	609	1	2	female	22.0	1	2	SC/Paris 2123	41.5792	NaN	C	20.78960
619	Gavey, Mr. Lawrence	620	0	2	male	26.0	0	0	31028	10.5000	NaN	S	5.25000
632	Stahelin-Maeglin, Dr. Max	633	1	1	male	32.0	0	0	13214	30.5000	B50	C	15.25000
665	Hickman, Mr. Lewis	666	0	2	male	32.0	2	0	S.O.C. 14879	73.5000	NaN	S	36.75000
708	Cleaver, Miss. Alice	709	1	1	female	22.0	0	0	113781	151.5500	NaN	S	75.77500
748	Marvin, Mr. Daniel Warner	749	0	1	male	19.0	1	0	113773	53.1000	D30	S	26.55000
887	Graham, Miss. Margaret Edith	888	1	1	female	19.0	0	0	112053	30.0000	B42	S	15.00000
889	Behr, Mr. Karl Howell	890	1	1	male	26.0	0	0	111369	30.0000	C148	C	15.00000

In [ ]:

```
"""
Load 300 rows from titanic data set
-Group the data on Pclass,Embarked and Gender column
-Drop data of all female passengers in Pclass 3 who embarked from 'S'
-Show the statistical data for all passengers who travelled from Pclass 2
-Calculate the total fare collected from male passengers in Pclass 3
-Show the Names of passengers who paid a fare of above 100 if they embarked from C
-Create a column next to Name column which shows the fare paid by them if 18% service tax was applied on it
"""
```