In [1]:

```
import pandas as pd
```

In [4]:

```
data = pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/ti
```

In [5]:

```
data.head(4)
```

Out[5]:

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

In [7]:

```
data.tail(4)
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	(
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	

In [9]:

```
# To know the data type of each column

data.dtypes
```

Out[9]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [10]:

```
# info() :- It gives a brief information about the dataset and an non-null content

data.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    int64  
 3   Name         891 non-null    object 
 4   Sex          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 [11]:

```
# Describe :- this describe() will give a little bit of statistical analysis of the data

data.describe()
```

Out[11]:

	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

In [6]:

```
data.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    int64  
 3   Name         891 non-null    object 
 4   Sex          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
```

1) Describe() will gives the statistical analysis for the numerical data only.

2) But in the above information categorical data is also there.

In [13]:

```
data['Fare']            # To access the single column
```

Out[13]:

```
0       7.2500
1      71.2833
2       7.9250
3      53.1000
4       8.0500
        ...   
886    13.0000
887    30.0000
888    23.4500
889    30.0000
890     7.7500
Name: Fare, Length: 891, dtype: float64
```

In [7]:

```
data.dtypes == "object"
```

Out[7]:

```
PassengerId    False
Survived       False
Pclass         False
Name            True
Sex             True
Age            False
SibSp          False
Parch          False
Ticket          True
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [4]:

```
# To get the column names of the categorical data follow as shown below...

data.dtypes[data.dtypes == "object"].index
```

Out[4]:

```
Index(['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked'], dtype='object')
```

In [5]:

```
data.dtypes               # 1)The left side column can also be considered as indexes.
                          # 2) The right side column is data.
```

Out[5]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [6]:

```
# To get the categorical data separately...as shown below

data[data.dtypes[data.dtypes == "object"].index]
```

Out[6]:

	Name	Sex	Ticket	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	A/5 21171	NaN	S
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	C85	C
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	NaN	S
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	C123	S
4	Allen, Mr. William Henry	male	373450	NaN	S
...	...	...	...	...	...
886	Montvila, Rev. Juozas	male	211536	NaN	S
887	Graham, Miss. Margaret Edith	female	112053	B42	S
888	Johnston, Miss. Catherine Helen "Carrie"	female	W./C. 6607	NaN	S
889	Behr, Mr. Karl Howell	male	111369	C148	C
890	Dooley, Mr. Patrick	male	370376	NaN	Q

891 rows × 5 columns

In [7]:

```
# Now, we are trying to describe the categorical data...  as shown below

data[data.dtypes[data.dtypes == "object"].index].describe()
```

Out[7]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	B96 B98	S
freq	1	577	7	4	644

In [12]:

```
data.head()
```

Out[12]:

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

In [8]:

```
# To get survived = 0

data[data['Survived'] == 0]
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0
...	...	...	...	...	...	...	...	...	...	
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

549 rows × 12 columns

In [9]:

```
data[(data['Survived']==0) & (data['Sex']=='male')]
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.86
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.07
...	...	...	...	...	...	...	...	...	...	
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.89
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.50
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.05
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

468 rows × 12 columns

In [10]:

```
data[(data['Survived']==0) & (data['Sex']=='male')].count()
```

Out[10]:

```
PassengerId    468
Survived       468
Pclass         468
Name           468
Sex            468
Age            360
SibSp          468
Parch          468
Ticket         468
Fare           468
Cabin           62
Embarked       468
dtype: int64
```

In [24]:

```
# To find the number of record

len(data[(data['Survived']==0) & (data['Sex']=='male')])
```

Out[24]:

```
468
```

In [25]:

```
len(data[(data['Survived']==0) & (data['Sex']=='female')])
```

Out[25]:

```
81
```

In [28]:

```
# To find the number of male and female in the dataset...

len(data[data['Sex']=='female'])
```

Out[28]:

```
314
```

In [29]:

```
len(data[data['Sex']=='male'])
```

Out[29]:

```
577
```

In [11]:

```
data['Sex'].value_counts()     # value_counts() :- This function works like 'group by- c
                               #        means it divides the sex column into two groups
                                #               'female' and it will gives the output.
```

Out[11]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [12]:

```
len(data[data['Sex'] == 'male'])
```

Out[12]:

```
577
```

In [33]:

```
len(data[data['Sex'] == 'female'])
```

Out[33]:

```
314
```

In [8]:

```
# how many male have survived?

len(data[(data['Survived'] == 1) & (data['Sex'] == 'male')])
```

Out[8]:

```
109
```

In [9]:

```
#how many female have survived?

len(data[(data['Survived'] == 1) & (data['Sex'] == 'female')])
```

Out[9]:

```
233
```

In [36]:

```
data.head()
```

Out[36]:

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

In [13]:

```
# To find the highest fare

max(data['Fare'])
```

Out[13]:

```
512.3292
```

In [14]:

```
data[data['Fare']== max(data['Fare']) ] ['Name']
```

Out[14]:

```
258                       Ward, Miss. Anna
679     Cardeza, Mr. Thomas Drake Martinez
737                 Lesurer, Mr. Gustave J
Name: Name, dtype: object
```

In [15]:

```
data[data['Fare']== max(data['Fare']) ]
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292

In [16]:

```
data[data['Fare']== min(data['Fare']) ]
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
179	180	0	3	Leonard, Mr. Lionel	male	36.0	0	0	LINE	0.0	
263	264	0	1	Harrison, Mr. William	male	40.0	0	0	112059	0.0	
271	272	1	3	Tornquist, Mr. William Henry	male	25.0	0	0	LINE	0.0	
277	278	0	2	Parkes, Mr. Francis "Frank"	male	NaN	0	0	239853	0.0	
302	303	0	3	Johnson, Mr. William Cahoone Jr	male	19.0	0	0	LINE	0.0	
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0	
466	467	0	2	Campbell, Mr. William	male	NaN	0	0	239853	0.0	
481	482	0	2	Frost, Mr. Anthony Wood "Archie"	male	NaN	0	0	239854	0.0	
597	598	0	3	Johnson, Mr. Alfred	male	49.0	0	0	LINE	0.0	
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0	
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0	
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0	
806	807	0	1	Andrews, Mr. Thomas Jr	male	39.0	0	0	112050	0.0	
815	816	0	1	Fry, Mr. Richard	male	NaN	0	0	112058	0.0	
822	823	0	1	Reuchlin, Jonkheer. John George	male	38.0	0	0	19972	0.0	

In [47]:

```
data.head()
```

Out[47]:

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

In [17]:

```
# Filter out the records where the column 'cabin' is not NaN

len(data[data['Cabin'].isnull() == False])
```

Out[17]:

```
204
```

In [18]:

```
data['Cabin'].isnull()    # isnull() will give true (or) False
                           # If we select the false we will get the not-NaN records as s
```

Out[18]:

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

In [19]:

```
# To add a new column in the data set...

data["new_col"] = "Naveen"
```

In [7]:

```
data
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

891 rows × 13 columns

In [20]:

```
# To create a duplicate column for the Name column with same data...

data["Name_new"] = data["Name"]
```

In [9]:

```
data
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

891 rows × 14 columns

In [21]:

```
# Addition of two columns and storing that addition values in a new column...as shown be
# From the above data set we are adding the 'Age' column and 'Pclass' column and storing
#       in the new column called 'Age_pclass'.

data["Age_pclass"] = data["Age"] + data["Pclass"]
```

In [11]:

```
data
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques	female	35.0	1	0	113803	53.1000	C123	

In [7]:

```
data.head()
```

Out[7]:

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

In [22]:

```
# To drop a column we have to give three parameters in drop() those are
# 1) column name -------> name of the column which we want to delete.
# 2) axis -------------> (i) if we want to delete column means axis value should be 1 (a
#                 (ii) if we want to delete the row means axis value should be 0 (axis = (
# 3) inplace--------> Inplace value should be 'true' to delete that particular column or
#                       permanently.
#------------------------------------------------------------------------------------------

# Another way to delete the column is by reassigning the value
data.drop("new_col", axis = 1, inplace =True)
```

In [9]:

```
data.head()
```

Out[9]:

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

1) Another way to delete the column permanently is by reassigning the value as shown below

```
    Example :- data = data.drop("new_col", axis = 1)
```

In [30]:

```
# To delete the row we will use the drop()  [in this case the default value of axis is 0

data = data.drop(0)
```

In [12]:

```
data
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

890 rows × 14 columns

In [31]:

```
# To fetch the rows there are two ways
#     These both ways are used to fetch the row level records.

# by using   1) iloc[] ----> This iloc will always takes the inbuit indexes.
#            2) loc[]-----> This will always takes the named indexes.----named indexes m
#                            visible to us on the screen.

data.iloc[0]
```

Out[31]:

```
PassengerId                                                    2
Survived                                                       1
Pclass                                                         1
Name           Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex                                                       female
Age                                                         38.0
SibSp                                                          1
Parch                                                          0
Ticket                                                  PC 17599
Fare                                                     71.2833
Cabin                                                        C85
Embarked                                                       C
Name_new       Cumings, Mrs. John Bradley (Florence Briggs Th...
Age_pclass                                                  39.0
Name: 1, dtype: object
```

In [33]:

```
data
```

Out[33]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

890 rows × 14 columns

In [36]:

```
# Like as shown now we can say that 'iloc' will take the inbuilt index and starts from t
# 'loc' will always shows the named indexes---means which are shown on the screen.

data.loc[1]
```

Out[36]:

```
PassengerId                                                    2
Survived                                                       1
Pclass                                                         1
Name           Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex                                                       female
Age                                                         38.0
SibSp                                                          1
Parch                                                          0
Ticket                                                  PC 17599
Fare                                                     71.2833
Cabin                                                        C85
Embarked                                                       C
Name_new       Cumings, Mrs. John Bradley (Florence Briggs Th...
Age_pclass                                                  39.0
Name: 1, dtype: object
```

In [37]:

```
# To fetch the multiple records at a time...

data.loc[2:7]
```

Out[37]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750

In [38]:

```
# Along with the limited records we can also select the columns which are required...

data.loc[101:105,['Name',"Cabin"]]
```

Out[38]:

	Name	Cabin
101	Petroff, Mr. Pastcho ("Pentcho")	NaN
102	White, Mr. Richard Frasar	D26
103	Johansson, Mr. Gustaf Joel	NaN
104	Gustafsson, Mr. Anders Vilhelm	NaN
105	Mionoff, Mr. Stoytcho	NaN

In [40]:

```
data.loc[3:4,["Fare","Cabin","Embarked"]]
```

Out[40]:

	Fare	Cabin	Embarked
3	53.10	C123	S
4	8.05	NaN	S

In [42]:

```
# To get the same above table with the 'iloc' means we have to give the index numbers in
#   column names....as shown below

data.iloc[2:4,[9,10,11]]
```

Out[42]:

	Fare	Cabin	Embarked
3	53.10	C123	S
4	8.05	NaN	S

In [47]:

```
# To get the complete row records based on only two columns (Passengerid, Pclass).

data.loc[: ,['PassengerId', 'Pclass']]
```

Out[47]:

```
     PassengerId  Pclass
  1            2       1
  2            3       3
  3            4       1
  4            5       3
  5            6       3
...          ...     ...
886          887       2
887          888       1
888          889       3
889          890       1
890          891       3
```

890 rows × 2 columns

In [48]:

```
#To get the same table by using'iloc' we will use the indexes instead of column names as

data.iloc[: ,[0,2]]
```

Out[48]:

```
     PassengerId  Pclass
  1            2       1
  2            3       3
  3            4       1
  4            5       3
  5            6       3
...          ...     ...
886          887       2
887          888       1
888          889       3
889          890       1
890          891       3
```

890 rows × 2 columns

In [50]:

```
# To get only particular records like 5th record & 9th record by using 'loc'

data.loc[[5,9], ['PassengerId','Survived','Pclass']]
```

Out[50]:

	PassengerId	Survived	Pclass
5	6	0	3
9	10	1	2

In [52]:

```
data.iloc[[4,8],[0,1,2]]
```

Out[52]:

	PassengerId	Survived	Pclass
5	6	0	3
9	10	1	2

In [54]:

```
# Fetch the record where the age is greater than 35.

data[data['Age']>35]
```

Out[54]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0000
...	...	...	...	...	...	...	...	...	...	...
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42.0	0	0	236852	13.0000
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47.0	0	0	345765	9.0000
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250

217 rows × 14 columns

In [60]:

```
# Taking the subset of data from the main dataset to perform some operations.

data1= data.iloc[1:5,[2,3,4,5,9,10]]
```

In [61]:

```
data1
```

Out[61]:

	Pclass	Name	Sex	Age	Fare	Cabin
2	3	Heikkinen, Miss. Laina	female	26.0	7.9250	NaN
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	53.1000	C123
4	3	Allen, Mr. William Henry	male	35.0	8.0500	NaN
5	3	Moran, Mr. James	male	NaN	8.4583	NaN

In [62]:

```
# To change the index numbers, if we want to set the name column a indexes...then

data1.set_index("Name")
```

Out[62]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [64]:

```
# To make these changes permanently, we have to reassingment to data1 (or) change the in

data1.set_index("Name",inplace = True)
```

In [65]:

```
data1
```

Out[65]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [66]:

```
# Drop the records where the value is 'NaN'...

data1.dropna()
```

Out[66]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1	C123

In [68]:

```
# Drop the columns which are having the 'NaN'...

data1.dropna(axis = 1)
```

Out[68]:

	Pclass	Sex	Fare
Name			
Heikkinen, Miss. Laina	3	female	7.9250
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	53.1000
Allen, Mr. William Henry	3	male	8.0500
Moran, Mr. James	3	male	8.4583

In [69]:

```
data1
```

Out[69]:

Name	Pclass	Sex	Age	Fare	Cabin
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [70]:

```
# New attribute called 'thresh' it will check upto the given number of non-NaN values, i
#  values are less than the given number means it will delete that particular column (or

data1.dropna(axis = 1, thresh=3)
```

Out[70]:

Name	Pclass	Sex	Age	Fare
Heikkinen, Miss. Laina	3	female	26.0	7.9250
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000
Allen, Mr. William Henry	3	male	35.0	8.0500
Moran, Mr. James	3	male	NaN	8.4583

In [71]:

```
data1.dropna(axis = 1, thresh=4)
```

Out[71]:

Name	Pclass	Sex	Fare
Heikkinen, Miss. Laina	3	female	7.9250
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	53.1000
Allen, Mr. William Henry	3	male	8.0500
Moran, Mr. James	3	male	8.4583

In [72]:

```
# Applying 'thresh' for the rows

data1.dropna(axis = 0, thresh=4)
```

Out[72]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.925	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.100	C123
Allen, Mr. William Henry	3	male	35.0	8.050	NaN

In [74]:

```
data1
```

Out[74]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [75]:

```
# From now we are going to see how to fill the 'Nan' values

data1.fillna("Naveen")
```

Out[75]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	Naveen
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	Naveen
Moran, Mr. James	3	male	Naveen	8.4583	Naveen

In [76]:

```
# if i want  to fill the 'NaN' with the average of age...

data1.fillna(data1['Age'].mean())
```

Out[76]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	32.0
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	32.0
Moran, Mr. James	3	male	32.0	8.4583	32.0

In [77]:

```
# I want to find the male & female  number of records separately...to do this we use gro

data1.groupby('Sex').count()
```

Out[77]:

	Pclass	Age	Fare	Cabin
Sex				
female	2	2	2	1
male	2	1	2	0

In [79]:

```
# If i want to know the Average age of female and male

data1.groupby('Sex').mean()["Age"]
```

Out[79]:

```
Sex
female    30.5
male      35.0
Name: Age, dtype: float64
```

In [80]:

```
# How much of revenue i am going to make from the male & female

data1.groupby('Sex').sum()['Fare']
```

Out[80]:

```
Sex
female    61.0250
male      16.5083
Name: Fare, dtype: float64
```

In [81]:

```
data1
```

Out[81]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [82]:

```
data2 = data.iloc[0:4, 0:5]
```

In [83]:

```
data2
```

Out[83]:

	PassengerId	Survived	Pclass	Name	Sex
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	3	1	3	Heikkinen, Miss. Laina	female
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	5	0	3	Allen, Mr. William Henry	male

In [84]:

```
# to merge horizontally

pd.concat([data1,data2])
```

Out[84]:

	Pclass	Sex	Age	Fare	Cabin	PassengerId	Survived	Name
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN	NaN	NaN	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123	NaN	NaN	NaN
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN	NaN	NaN	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN	NaN	NaN	NaN
1	1	female	NaN	NaN	NaN	2.0	1.0	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	3	female	NaN	NaN	NaN	3.0	1.0	Heikkinen, Miss. Laina
3	1	female	NaN	NaN	NaN	4.0	1.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	3	male	NaN	NaN	NaN	5.0	0.0	Allen, Mr. William Henry

In [85]:

```
# To merge vertically we have to give the axis = 1

pd.concat([data1,data2],axis =1)
```

Out[85]:

	Pclass	Sex	Age	Fare	Cabin	PassengerId	Survived	Pclass	Name
Heikkinen, Miss. Laina	3.0	female	26.0	7.9250	NaN	NaN	NaN	NaN	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1.0	female	35.0	53.1000	C123	NaN	NaN	NaN	NaN
Allen, Mr. William Henry	3.0	male	35.0	8.0500	NaN	NaN	NaN	NaN	NaN
Moran, Mr. James	3.0	male	NaN	8.4583	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	2.0	1.0	1.0	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	NaN	NaN	NaN	NaN	NaN	3.0	1.0	3.0	Heikkinen, Miss. Laina
3	NaN	NaN	NaN	NaN	NaN	4.0	1.0	1.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	NaN	NaN	NaN	NaN	NaN	5.0	0.0	3.0	Allen, Mr. William Henry

In [86]:

```
data1
```

Out[86]:

	Pclass	Sex	Age	Fare	Cabin
Name					
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN
Moran, Mr. James	3	male	NaN	8.4583	NaN

In [87]:

```
# If i want to take the Pclass column and divide that entire column with 3

data1['New'] = data1["Pclass"].apply(lambda x:x/3)
```

In [88]:

```
data1
```

Out[88]:

	Pclass	Sex	Age	Fare	Cabin	New
Name						
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN	1.000000
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123	0.333333
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN	1.000000
Moran, Mr. James	3	male	NaN	8.4583	NaN	1.000000

In [89]:

```
def fun(x):
    return x/3
```

In [90]:

```
# Now without using the lambda function, there is a another way

data1['New1'] = data1['Pclass'].apply(fun)
```

In [91]:

```
data1
```

Out[91]:

	Pclass	Sex	Age	Fare	Cabin	New	New1
Name							
Heikkinen, Miss. Laina	3	female	26.0	7.9250	NaN	1.000000	1.000000
Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	53.1000	C123	0.333333	0.333333
Allen, Mr. William Henry	3	male	35.0	8.0500	NaN	1.000000	1.000000
Moran, Mr. James	3	male	NaN	8.4583	NaN	1.000000	1.000000

In [92]:

```
data2
```

Out[92]:

	PassengerId	Survived	Pclass	Name	Sex
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	3	1	3	Heikkinen, Miss. Laina	female
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	5	0	3	Allen, Mr. William Henry	male

In [93]:

```
# To get the length of the name and store it in a separate column

data2['len_Name'] = data2['Name'].apply(len)
```

In [94]:

```
data2
```

Out[94]:

	PassengerId	Survived	Pclass	Name	Sex	len_Name
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	51
2	3	1	3	Heikkinen, Miss. Laina	female	22
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	44
4	5	0	3	Allen, Mr. William Henry	male	24

In [95]:

```
# where ever the Passengerid column having the value less than 3 then don't change the v
#    if the value is greater than 3 means give the logarithmic value..

import math
def cust1(x):
    if x < 3:
        return x
    else :
        return math.log10(x)
```

In [96]:

```
data2['Passengerid_filter'] = data2['PassengerId'].apply(cust1)
```

In [97]:

```
data2
```

Out[97]:

	PassengerId	Survived	Pclass	Name	Sex	len_Name	Passengerid_filter
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	51	2.000000
2	3	1	3	Heikkinen, Miss. Laina	female	22	0.477121
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	44	0.602060
4	5	0	3	Allen, Mr. William Henry	male	24	0.698970

In [106]:

```
# I want to replace the values in sex column as where ever there is female replace it wi
#  and in place male replace it with the '0'

def test1(x):
    if x == 'female' :
        return 1
    else :
        return 0
```

In [107]:

```
data2['Sex'] = data2['Sex'].apply(test1)
```

In [108]:

```
data2
```

Out[108]:

	PassengerId	Survived	Pclass	Name	Sex	len_Name	Passengerid_filter
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	51	2.000000
2	3	1	3	Heikkinen, Miss. Laina	1	22	0.477121
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	44	0.602060
4	5	0	3	Allen, Mr. William Henry	0	24	0.698970

In [109]:

```
data.head(10)
```

Out[109]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.86
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.07
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.13
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.07
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.70

If fare is 0-100 -->A 100-200 ---->B 200+ ---->C

In [110]:

```
def test3(x):
    if x <= 100 :
        return "A"
    elif x >100 and x<200:
        return "B"
    else :
        return "C"
```

In [111]:

```
data['Fare_group'] = data['Fare'].apply(test3)
```

In [112]:

```
data
```

Out[112]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.45
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

890 rows × 15 columns