In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import pandas_profiling
```

In [2]:

```
df1=pd.read_csv(r"C:\Users\HP\Downloads\titanic_train.csv")
```

In [3]:
```
1 df1
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
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	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 12 columns

In [4]:
```
1 df2=pd.read_csv(r"C:\Users\HP\Downloads\test.csv")
```

In [5]:
```
1 df2
```

Out[5]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	

418 rows × 11 columns

In [6]:
```
1 df1.head(10) # Initial rows & columns
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	Na
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E4
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	Na
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	Na
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	Na

In [7]:
```
1 df1.tail(10) # last rows & columns
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.8958
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.5000
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

In [8]:
```
1 df1.shape # Total No. of rows =418 & Columns=11
```

Out[8]: (891, 12)

In [9]:
```
df1.describe() #Give the numarical details in the dataset
```

Out[9]:

	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 [10]:
```
#describe fucntion give the mathematical insights of the numarical details o
#1.df1 is a training dataset which 891 or 40% of the actual number of people
#2.Survived is a catogorical feature with 0 & 1 values
#3.38% of the sample survived rate represents the actual survival rate at 32
#4.
```

In [11]:
```
df1.info() # Gives the data types,Columns & Non Null Count.
```

```
<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 [12]:
```
#We can use pandas profile to see the overall summary of dataset in a nutshe
from pandas_profiling import ProfileReport
```

In [13]:
```
titanic_profile=ProfileReport(df1,title="Titanic Profile Report")
```

In [14]:
```
1 titanic_profile
```

Summarize dataset: 100% 47/47 [00:08<00:00, 5.67it/s, Completed]

Generate report structure: 100% 1/1 [00:11<00:00, 11.16s/it]

Render HTML: 100% 1/1 [00:02<00:00, 2.11s/it]

Overview

Dataset statistics

Number of variables	12
Number of observations	891
Missing cells	866
Missing cells (%)	8.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	83.7 KiB
Average record size in memory	96.1 B

Variable types

Numeric	5
Categorical	7

Alerts

`Name` has a high cardinality: 891 distinct values	**High cardinality**
`Ticket` has a high cardinality: 681 distinct values	**High cardinality**
`Cabin` has a high cardinality: 147 distinct values	**High cardinality**

Out[14]:

In [15]:
```
1 type(titanic_profile)
```

Out[15]: pandas_profiling.profile_report.ProfileReport

In [16]:
```
1 titanic_profile.to_file(output_file="Titanic_profile ")
```

```
C:\Users\HP\anaconda3\lib\site-packages\pandas_profiling\profile_report.py:309:
UserWarning: Extension  not supported. For now we assume .html was intended. To
remove this warning, please use .html or .json.
  warnings.warn(
```

Export report to file: 100% 1/1 [00:00<00:00, 2.29it/s]

Data Preprocessing

In [17]:
```
1 #Total number of null values
2 miss=df1.isnull().sum()
```

In [18]:
```
1 miss
```

Out[18]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [19]:
```
1  len(df1)*100
```

Out[19]:
```
89100
```

In [20]:
```
1 #percentage of the missing data from the table(ratio of Null value with the
2 m_data=miss/len(df1)*100
```

In [21]:
```
m_data
```

Out[21]:
```
PassengerId     0.000000
Survived        0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            19.865320
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.000000
Cabin          77.104377
Embarked        0.224467
dtype: float64
```

In [22]:
```
miss=pd.concat([miss,m_data],axis=1,keys=["Total","%"])
print(miss)
```

```
             Total          %
PassengerId      0   0.000000
Survived         0   0.000000
Pclass           0   0.000000
Name             0   0.000000
Sex              0   0.000000
Age            177  19.865320
SibSp            0   0.000000
Parch            0   0.000000
Ticket           0   0.000000
Fare             0   0.000000
Cabin          687  77.104377
Embarked         2   0.224467
```

In [23]:
```
#filling the missing values of age with the median of age
med_age=df1.Age.median()
med_age
df1.Age.fillna(med_age,inplace=True)
```

In [24]:
```
#No null values of age
df1.isnull().sum()
```

Out[24]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [25]:
```
em_mode=df1.Embarked
```

In [26]:
```
em_mode
```

Out[26]:
```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
888    S
889    C
890    Q
Name: Embarked, Length: 891, dtype: object
```

In [27]:
```
df1.Embarked=df1.Embarked.fillna(df1['Embarked'].mode()[0])
```

In [28]:
```
df1.Embarked.isnull().sum()
```

Out[28]:
```
0
```

In [29]:
```
df1.isnull().sum(),len(df1)
```

Out[29]:
```
(PassengerId      0
 Survived         0
 Pclass           0
 Name             0
 Sex              0
 Age              0
 SibSp            0
 Parch            0
 Ticket           0
 Fare             0
 Cabin          687
 Embarked         0
 dtype: int64,
 891)
```

In [30]:
```
#in above data, cabin feature can be dropped as it do not contain most of th
```

In [31]:
```
#dropping cabin column
df1.drop("Cabin",axis=1,inplace=True)
```

In [32]:
```
#passenger Ticket column can also drop as it do not contribute to the surviv
df1.drop("Ticket",axis=1,inplace=True)
```

In [33]:
```
#passenger Id column can also drop as it do not contribute to the survival
df1.drop("PassengerId",axis=1,inplace=True)
```

In [34]:
```
df1.isnull().sum()
```

Out[34]:
```
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

Create new Fields

In [35]:
```
#craete new age bands to improve prediction insights
#create new feature called family based on parch & Sibsp
#create a fare range feature if it helps analysis
```

In [36]:
```
df1.Age.isnull().sum()
```

Out[36]: 0

In [37]:
```
df1.loc[df1["Age"]<=1,"Age Bands"]="Infant"
df1.loc[(df1["Age"]>1) & (df1["Age"]<12),"Age Bands"]="Children"
df1.loc[df1["Age"]>12,"Age Bands"]="Adult"
```

In [38]:
```
df1
```

Out[38]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Adult
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Adult
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Adult
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Adult
...	...	...	...	...	...	...	...	...	...	...
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	S	Adult
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	S	Adult
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	23.4500	S	Adult
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C	Adult
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	Q	Adult

891 rows × 10 columns

In [39]:
```
#Creating ticket fare band to check the dependency of ticket fare rate to su
df1.loc[(df1["Fare"]>=0) &(df1["Fare"]<=10),"Fare band"]=1
df1.loc[(df1["Fare"]>10) &(df1["Fare"]<=15),"Fare band"]=2
df1.loc[(df1["Fare"]>15) &(df1["Fare"]<=35),"Fare band"]=3
df1.loc[df1["Fare"]>35,"Fare band"]=4
```

In [40]:
```
1 df1.head()
```

Out[40]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Adult	1.0
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult	4.0
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Adult	1.0
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Adult	4.0
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Adult	1.0

In [41]:
```
1 k=df1.Embarked.mode()
```

In [42]:
```
1 k
```

Out[42]:
```
0    S
Name: Embarked, dtype: object
```

In [43]:
```
#here we can extract names from Name
df1["Title"]=df1.Name.str.extract('([A-Za-z]+)\.',expand=False)
#Now we can use cross tab and make the name as roes and sex in columns
pd.crosstab(df1['Name'],df1['Sex'])
```

Out[43]:

Sex	female	male
Name		
Abbing, Mr. Anthony	0	1
Abbott, Mr. Rossmore Edward	0	1
Abbott, Mrs. Stanton (Rosa Hunt)	1	0
Abelson, Mr. Samuel	0	1
Abelson, Mrs. Samuel (Hannah Wizosky)	1	0
...	...	...
de Mulder, Mr. Theodore	0	1
de Pelsmaeker, Mr. Alfons	0	1
del Carlo, Mr. Sebastiano	0	1
van Billiard, Mr. Austin Blyler	0	1
van Melkebeke, Mr. Philemon	0	1

891 rows × 2 columns

In [44]:
```
df1["Title"]=df1["Title"].replace(['Lady','Countess','capt','Col','Don','Dr'
```

In [45]:
```
df1["Title"]
```

Out[45]:
```
0        Mr
1       Mrs
2      Miss
3       Mrs
4        Mr
       ...
886    Rare
887    Miss
888    Miss
889      Mr
890      Mr
Name: Title, Length: 891, dtype: object
```

In [46]:
```
#Moidify the sur name as per our requirement to analyse
df1["Title"]=df1["Title"].replace('Mlle',"Miss")
df1["Title"]=df1["Title"].replace('Ms',"Miss")
df1["Title"]=df1["Title"].replace('Mme',"Mrs")
#take the mean of the title of people, which will give an insight of people
df1[["Title","Survived"]].groupby("Title").mean()
```

Out[46]:

Title	Survived
Master	0.575000
Miss	0.702703
Mr	0.156673
Mrs	0.793651
Rare	0.347826

In [47]:
```
df1[["Title","Survived"]].groupby("Title").mean()
```

Out[47]:

Title	Survived
Master	0.575000
Miss	0.702703
Mr	0.156673
Mrs	0.793651
Rare	0.347826

In [48]:
```
#Give numberring of people from different catogary, if there is a need to fe
title_map={"Mr":1,"Mrs":2,"Mrs":3,"Master":4,"Rare":5}
df1["Title"]=df1["Title"].map(title_map)
df1["Title"]=df1["Title"].fillna(0)
```

In [49]:
```
1 df1
```

Out[49]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Adult	1.0
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult	4.0
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Adult	1.0
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Adult	4.0
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Adult	1.0
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	S	Adult	2.0
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	S	Adult	3.0
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	23.4500	S	Adult	3.0
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C	Adult	3.0
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	Q	Adult	1.0

891 rows × 12 columns

#Decision

In [50]:
```
#Converting sex feature to a new feature called Gender, where male is 0 and
```

In [51]:
```
gend={"male":0,"female":1}
df1["Gender"]=df1["Sex"].map(gend)
```

In [52]:
```
df1
```

Out[52]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Adult	1.0
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult	4.0
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Adult	1.0
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Adult	4.0
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Adult	1.0
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	S	Adult	2.0
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	S	Adult	3.0
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	28.0	1	2	23.4500	S	Adult	3.0
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C	Adult	3.0
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	Q	Adult	1.0

891 rows × 13 columns

In [53]:
```
#we can drop name,as name is not a relavent field to survive.
df1.drop("Name",axis=1)
```

Out[53]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band	Title	Gende
0	0	3	male	22.0	1	0	7.2500	S	Adult	1.0	1.0	
1	1	1	female	38.0	1	0	71.2833	C	Adult	4.0	3.0	
2	1	3	female	26.0	0	0	7.9250	S	Adult	1.0	0.0	
3	1	1	female	35.0	1	0	53.1000	S	Adult	4.0	3.0	
4	0	3	male	35.0	0	0	8.0500	S	Adult	1.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Adult	2.0	5.0	
887	1	1	female	19.0	0	0	30.0000	S	Adult	3.0	0.0	
888	0	3	female	28.0	1	2	23.4500	S	Adult	3.0	0.0	
889	1	1	male	26.0	0	0	30.0000	C	Adult	3.0	1.0	
890	0	3	male	32.0	0	0	7.7500	Q	Adult	1.0	1.0	

891 rows × 12 columns

In [54]:
```
df1["Embarked"]= df1["Embarked"].map({"S":0,"C":1,"Q":2})
```

In [55]:
```
1 df1.head()
```

Out[55]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band	Titl
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	0	Adult	1.0	1.
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	1	Adult	4.0	3.
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	0	Adult	1.0	0.
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	0	Adult	4.0	3.
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	0	Adult	1.0	1.

In [56]:
```
1 df1.drop("Name",axis=1)
```

Out[56]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band	Title	Gende
0	0	3	male	22.0	1	0	7.2500	0	Adult	1.0	1.0	
1	1	1	female	38.0	1	0	71.2833	1	Adult	4.0	3.0	
2	1	3	female	26.0	0	0	7.9250	0	Adult	1.0	0.0	
3	1	1	female	35.0	1	0	53.1000	0	Adult	4.0	3.0	
4	0	3	male	35.0	0	0	8.0500	0	Adult	1.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	0	Adult	2.0	5.0	
887	1	1	female	19.0	0	0	30.0000	0	Adult	3.0	0.0	
888	0	3	female	28.0	1	2	23.4500	0	Adult	3.0	0.0	
889	1	1	male	26.0	0	0	30.0000	1	Adult	3.0	1.0	
890	0	3	male	32.0	0	0	7.7500	2	Adult	1.0	1.0	

891 rows × 12 columns

In [57]:
```
#drop the catogorical columns
df1.drop("Sex",axis=1,inplace=True)
```

In [58]:
```
#drop the catogorical columns
df1.drop("Name",axis=1,inplace=True)
```

In [59]:
```
df1
```

Out[59]:

	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band	Title	Gender
0	0	3	22.0	1	0	7.2500	0	Adult	1.0	1.0	0
1	1	1	38.0	1	0	71.2833	1	Adult	4.0	3.0	1
2	1	3	26.0	0	0	7.9250	0	Adult	1.0	0.0	1
3	1	1	35.0	1	0	53.1000	0	Adult	4.0	3.0	1
4	0	3	35.0	0	0	8.0500	0	Adult	1.0	1.0	0
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	13.0000	0	Adult	2.0	5.0	0
887	1	1	19.0	0	0	30.0000	0	Adult	3.0	0.0	1
888	0	3	28.0	1	2	23.4500	0	Adult	3.0	0.0	1
889	1	1	26.0	0	0	30.0000	1	Adult	3.0	1.0	0
890	0	3	32.0	0	0	7.7500	2	Adult	1.0	1.0	0

891 rows × 11 columns

Post Pandas Profiling

In [60]:

```
#Now we can see the final profile of the data with pandas profiling
df1.profile_report()
```

Summarize dataset: 100% 36/36 [00:05<00:00, 6.72it/s, Completed]

Generate report structure: 100% 1/1 [12:54<00:00, 774.44s/it]

Render HTML: 100% 1/1 [00:01<00:00, 1.71s/it]

Most occurring characters

Value	Count	Frequency (%)
3	491	55.1%
1	216	24.2%
2	184	20.7%

Most occurring categories

Value	Count	Frequency (%)
Decimal Number	891	100.0%

Most frequent character per category

Decimal Number

Value	Count	Frequency (%)
3	491	55.1%
1	216	24.2%
2	184	20.7%

Out[60]:

In [61]:
```
1 df1
```

Out[61]:

	Survived	Pclass	Age	SibSp	Parch	Fare	Embarked	Age Bands	Fare band	Title	Gender
0	0	3	22.0	1	0	7.2500	0	Adult	1.0	1.0	0
1	1	1	38.0	1	0	71.2833	1	Adult	4.0	3.0	1
2	1	3	26.0	0	0	7.9250	0	Adult	1.0	0.0	1
3	1	1	35.0	1	0	53.1000	0	Adult	4.0	3.0	1
4	0	3	35.0	0	0	8.0500	0	Adult	1.0	1.0	0
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	13.0000	0	Adult	2.0	5.0	0
887	1	1	19.0	0	0	30.0000	0	Adult	3.0	0.0	1
888	0	3	28.0	1	2	23.4500	0	Adult	3.0	0.0	1
889	1	1	26.0	0	0	30.0000	1	Adult	3.0	1.0	0
890	0	3	32.0	0	0	7.7500	2	Adult	1.0	1.0	0

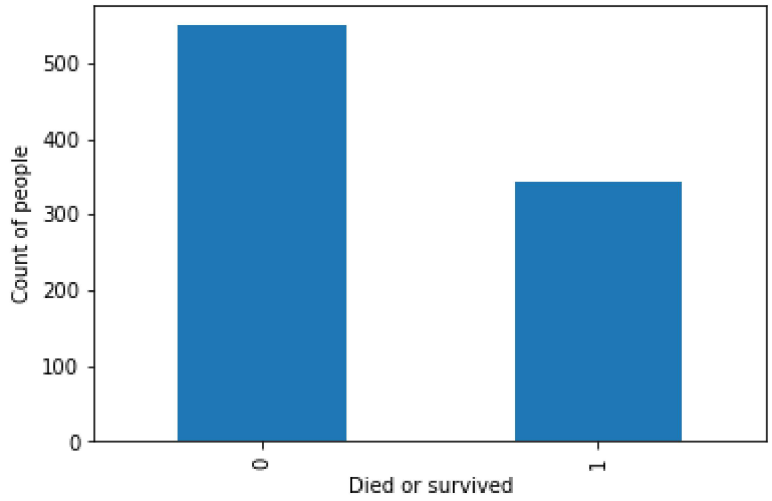
891 rows × 11 columns

Data Visualisation

In [62]:
```
1 #Visualise the Total Count of survival and Victims
2 sc=df1.groupby(["Survived"])["Survived"].count()
```

In [63]:
```
plt=sc.plot(kind="bar")
plt.set_xlabel("Died or survived")
plt.set_ylabel("Count of people")
```

Out[63]:
```
Text(0, 0.5, 'Count of people')
```



INSIGHTS

#Only 342 survives our of 842 #Majority of the people died,which means there is less chance of survival

In [64]:
```
#Plot the gender with more survival rate adults
```

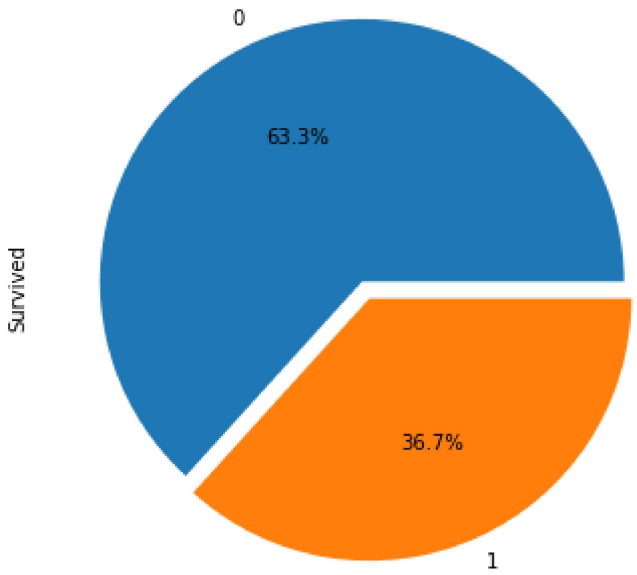
In [65]:
```
gen_pl=df1.groupby(["Survived","Age Bands"]).count()['Gender']
```

In [66]:
```
gen_pl
```

Out[66]:
```
Survived  Age Bands
0         Adult        520
          Children      27
          Infant         2
1         Adult        302
          Children      27
          Infant        12
Name: Gender, dtype: int64
```

In [67]:
```
#Survided adult by gender
df1[df1["Age Bands"]=="Adult"].Survived.groupby(df1.Survived).count().plot(k
plt.axis('equal')
```

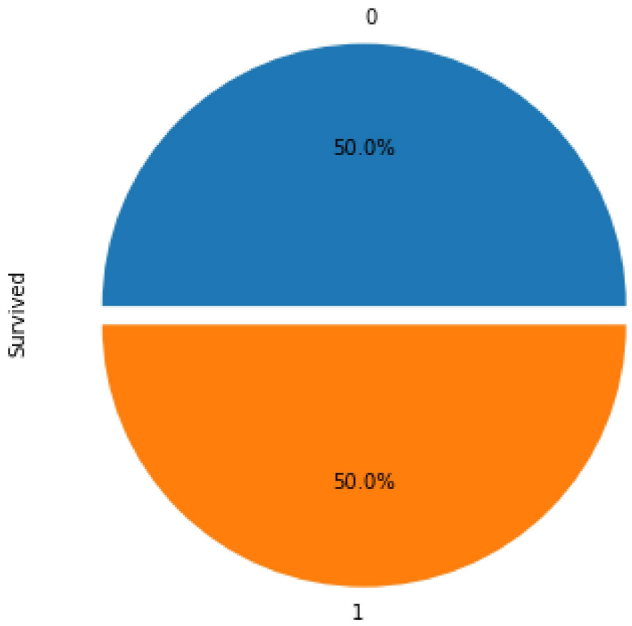
Out[67]: (-0.325, 1.325, 0.0, 576.45)



In [68]:

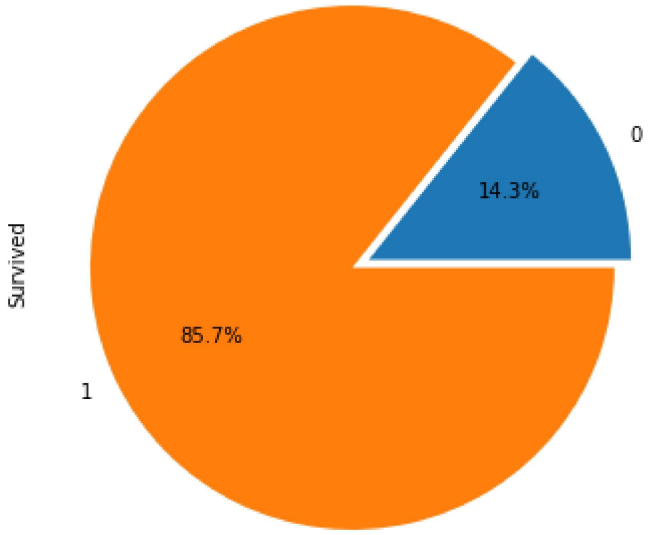
```
#Survided Children by gender
df1[df1["Age Bands"]=="Children"].Survived.groupby(df1.Survived).count().plo
plt.axis('equal')
```

Out[68]: (-0.325, 1.325, 0.0, 576.45)



In [69]:
```
#Survided infants by gender
df1[df1["Age Bands"]=="Infant"].Survived.groupby(df1.Survived).count().plot(
```

Out[69]: <AxesSubplot:ylabel='Survived'>



Insights

Majority Passengers were Adults

Almost half of the total number of children survived.

Most of the Adults failed to Survive

More than 85 percent of Infant Survived

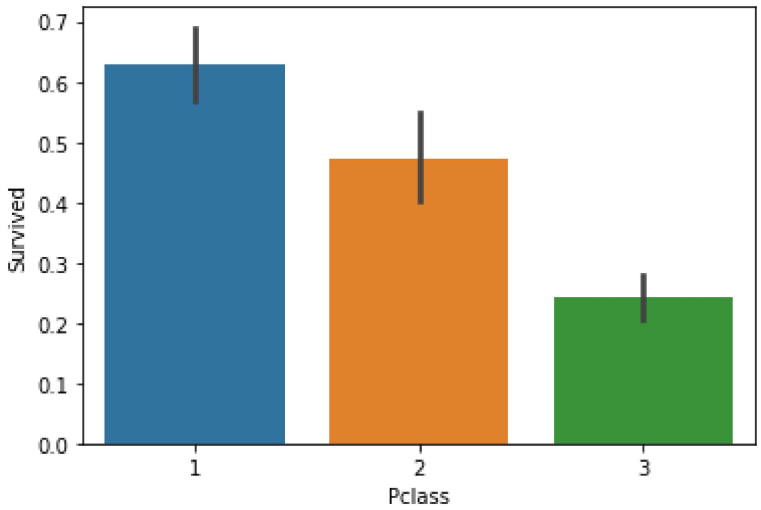
In [70]:
```
#Did Economy Class had an impact on survival rate?
```

In [71]:
```
1 df1.groupby(['Survived','Pclass'])["Survived"].sum()
```

Out[71]:
```
Survived  Pclass
0         1           0
          2           0
          3           0
1         1         136
          2          87
          3         119
Name: Survived, dtype: int64
```

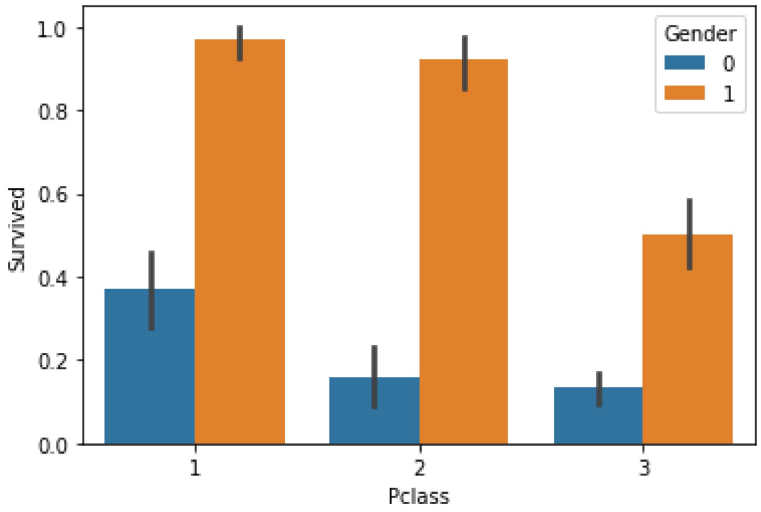
In [72]:
```
1 sns.barplot(x="Pclass",y="Survived",data=df1)
```

Out[72]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



In [73]:
```
1 #Male & Female survival rate comparison on different passenger class
2 sns.barplot(x="Pclass",y="Survived",hue="Gender",data=df1)
```

Out[73]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



Insights

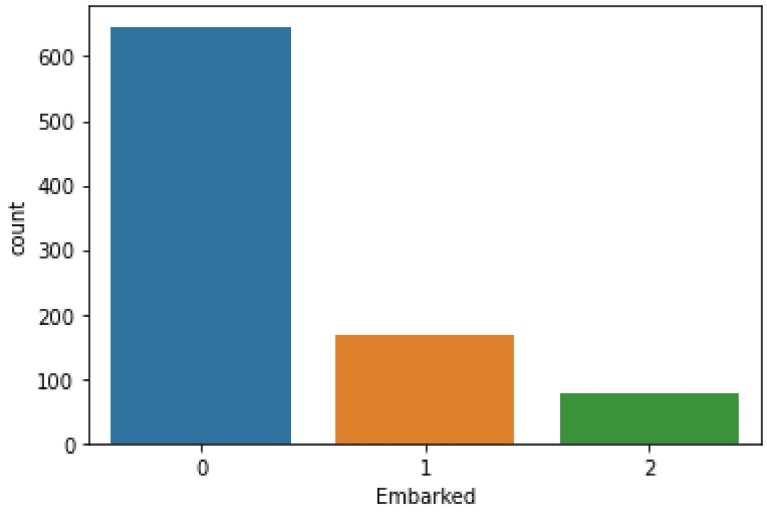
#Most of the passengers travelled in third class only. But 24% of them survived #More passengers in 1st class survived and female are given priority. #Passengers travelled in 1st class has more chances of survival than 2nd and third class.

In [74]:
```
#What is the survival possibilty based on the embarkement of passengers
df1.groupby(["Survived","Embarked"])["Survived"].count()
```

Out[74]:
```
Survived  Embarked
0         0           427
          1            75
          2            47
1         0           219
          1            93
          2            30
Name: Survived, dtype: int64
```

In [75]:
```
sns.countplot(x="Embarked",data=df1)
```

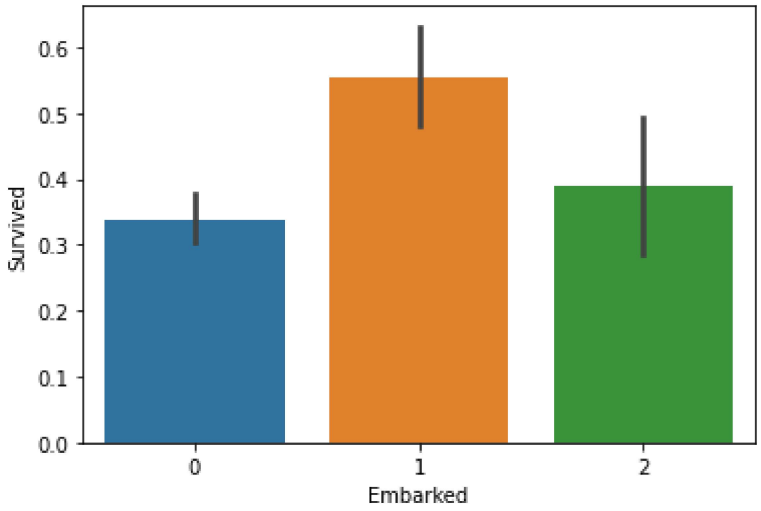
Out[75]:
```
<AxesSubplot:xlabel='Embarked', ylabel='count'>
```



INSIGHT #The passengers who embarked 0 has survived more than 1 and 2 embarkment

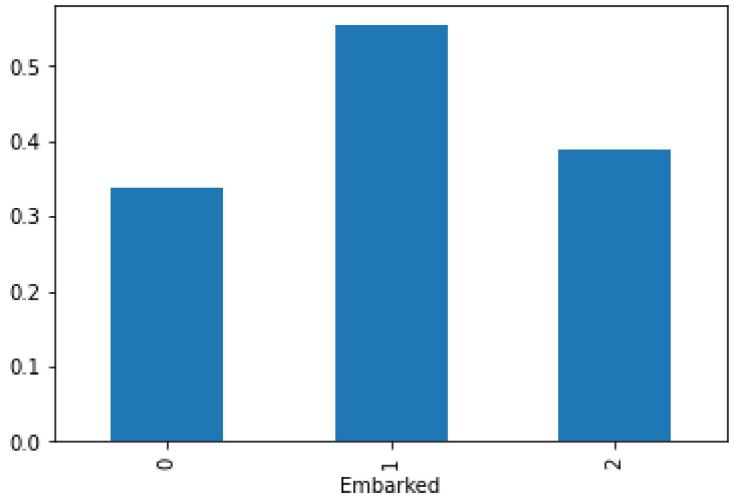
In [76]:
```
3# Most number of boarded by embarkment location
sns.barplot(x="Embarked",y="Survived",data=df1)
```

Out[76]: <AxesSubplot:xlabel='Embarked', ylabel='Survived'>



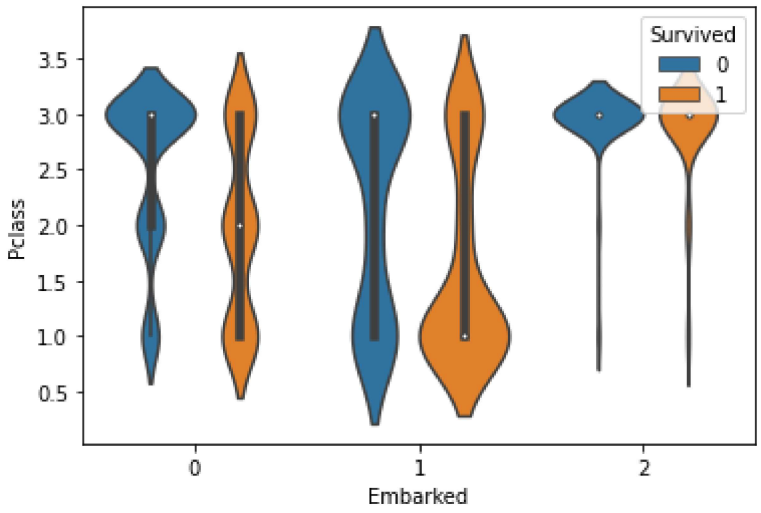
In [108]:
```
df1[["Embarked","Survived"]].groupby("Embarked").mean().Survived.plot(kind='
plt.set_xlabel("Embarked")
plt.set_ylabel("Survival Probability")
```

Out[108]: Text(17.200000000000003, 0.5, 'Survival Probability')



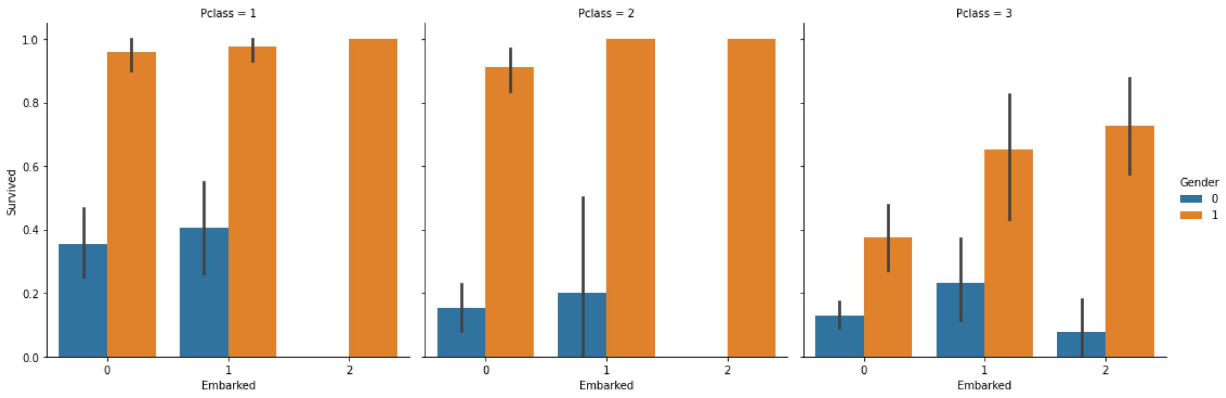
In [112]:
```
1 #To compare the relative survival rate and decesed ratde based on the embark
2 sns.violinplot(x="Embarked",y="Pclass",hue="Survived",data=df1,Split=True)
```

Out[112]: <AxesSubplot:xlabel='Embarked', ylabel='Pclass'>



In [128]:
```
1 sns.catplot(x="Embarked",y="Survived",hue="Gender",data=df1,kind="bar",col="
```

Out[128]: <seaborn.axisgrid.FacetGrid at 0x2145d13ee20>

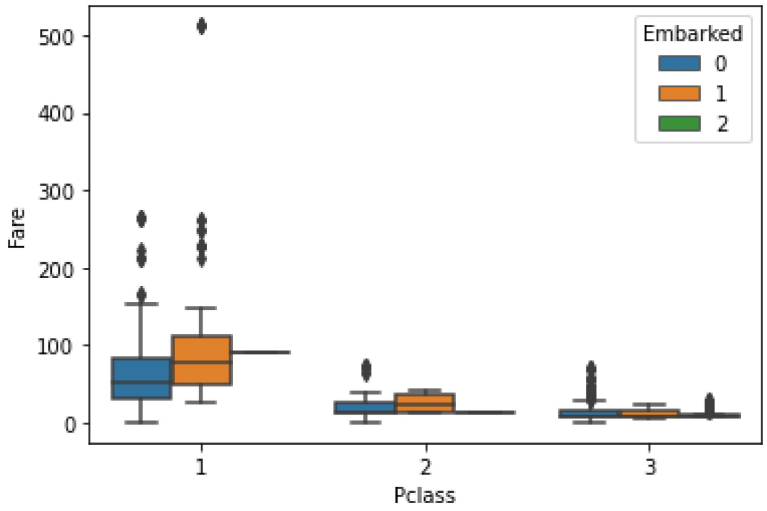


INSIGHTS

-Most passengers from Port C survived -Males passengers survived most -Females survived less in all the 3 classes, comparartively

In [138]:
```
#Average fare by passenger class & embark location
sns.boxplot(x="Pclass", y="Fare",data=df1, hue="Embarked")
```

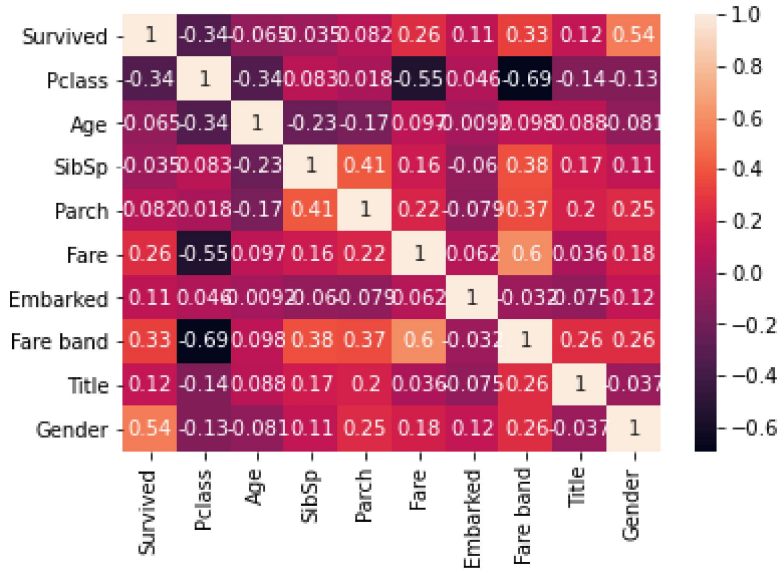
Out[138]: <AxesSubplot:xlabel='Pclass', ylabel='Fare'>



INSIGHT -The mean average fair of class 1 is higher,compared to class 2 and class 3

In [157]:
```
# Which Feature had more impact on survival rate
sns.heatmap(df1.corr(),annot=True)
```

Out[157]: <AxesSubplot:>



INSIGHTS

-Gender is one of the main factor determine the survival -All the Features are not necesary to predict the survival -More Features Create complexity -Fare has Positive Corelation

Conclusion:

Most of the passengers died. 85% of the infants Survived. Majority of the Males Died.

In [ ]:
```
1
```