Performing EDA on TITANIC Dataset.

Problem statement.

"Can you perform EDA and predict whether a passenger will survive the Titanic disaster based on various features such as age, gender, passenger class, family size, fare, and embarkation port?"

In [1]:
```
# import python libraries.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [108]:
```
#firstly import dataset.
df = pd.read_csv('train.csv')
df
```

Out[108]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
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	N
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	N
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	N
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	N

891 rows × 12 columns

In [109]:
```
df.tail()
```

Out[109]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	I
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	

In [110]:
```
df['Name'].count()
```

Out[110]:
```
891
```

Why do EDA?

```
Model building
Analysis and reporting
Validate assumptions
Handling missing values
feature engineering
detecting outliers
```

Perform Univariate Analysis

AGE

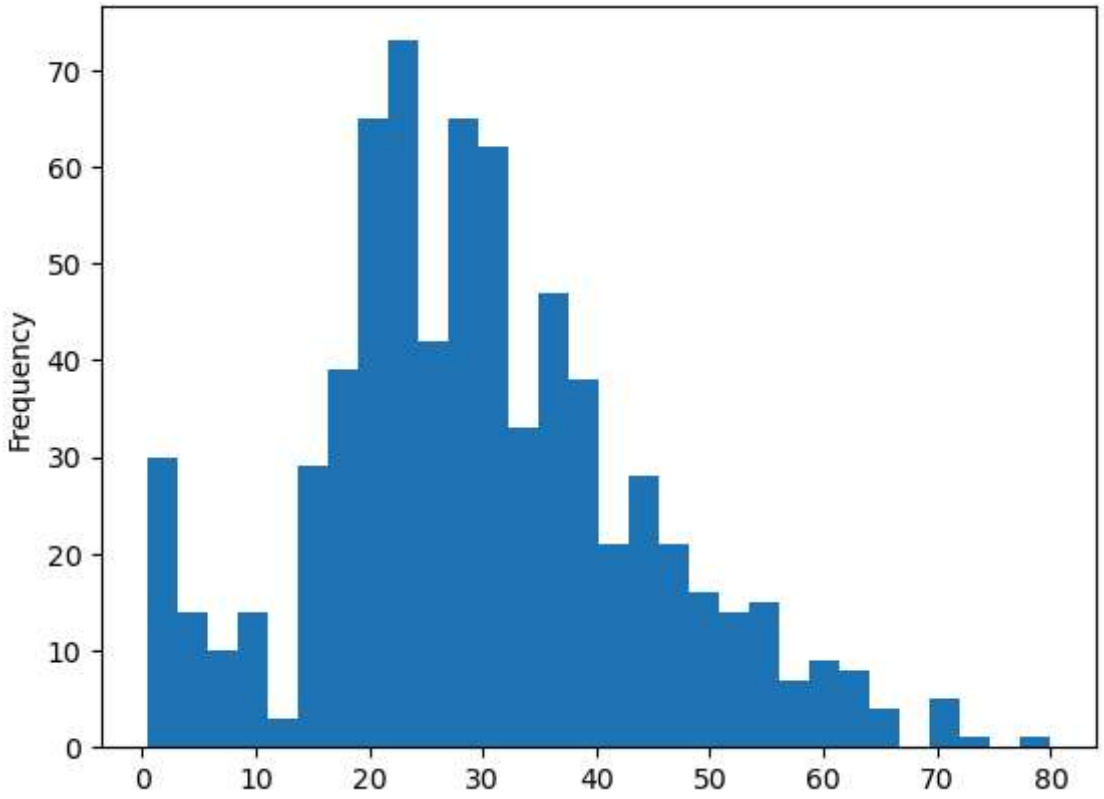
```
conclusions:

1.Age is normally(almost) distributed.
2.20% of the values are missing.
3.There are some outliers.
```

In [111]:
```
# age column, here we gonna to perform some Descriptive statatics. (mean, mead

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

Out[111]:
```
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
```

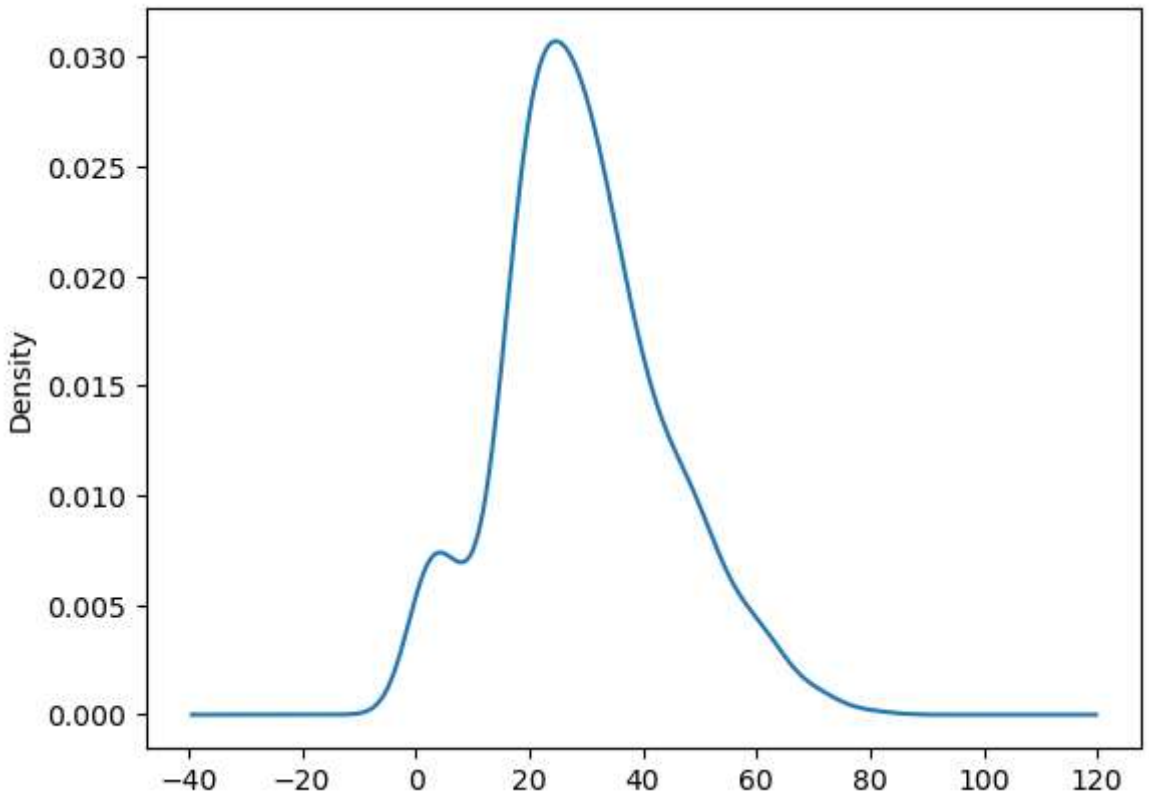
In [112]:
```
df['Age'].plot(kind='hist', bins=30)
```

Out[112]:
```
<Axes: ylabel='Frequency'>
```



In [113]:
```
df['Age'].plot(kind='kde') #kde graph-------> its give the distribution of dat
```

Out[113]:
```
<Axes: ylabel='Density'>
```

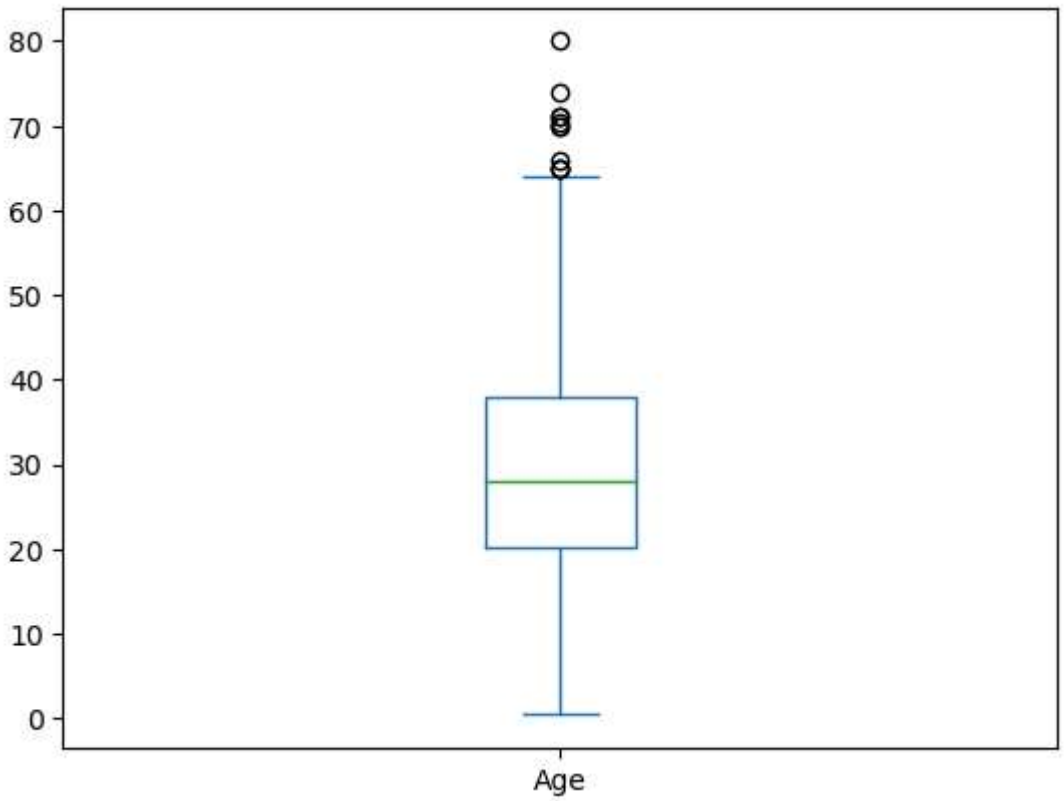


In [114]:
```
df["Age"].skew() #its very close to the the zero. no more -ve skews and no mor
```

Out[114]:
```
0.38910778230082704
```

In [10]:
```
df['Age'].plot(kind='box') #its is very imortants
```

Out[10]: <Axes: >



In [115]:
```
df[df['Age']>65]
```

Out[115]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A. 24579	10.5000	Nal
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	Nal
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	Nal
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	A2
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0	C.A. 24580	10.5000	Nal
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735	71.0000	B2
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	Nal

In [116]:
```
df['Age'].isnull().sum() #its give the missing value.
```

Out[116]: 177

In [117]:
```
len(df['Age']) #total no of passanger in age column
```

Out[117]: 891

In [118]:
```
df['Age'].isnull().sum()/len(df['Age'])*100
```

Out[118]: 19.865319865319865

In [15]:
```
# here around 19% missing value.
```

Fare

```
conclusions

1.The data is highly(positively) skewed.
2.Fare col actually contains the group fare and not the individual fare(This
migth be and issue).
```

```
3.We need to create a new col called individual fare.
```

In [16]:
```
df['Fare']
```

Out[16]:
```
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 [17]:
```
df['Fare'].describe()
```

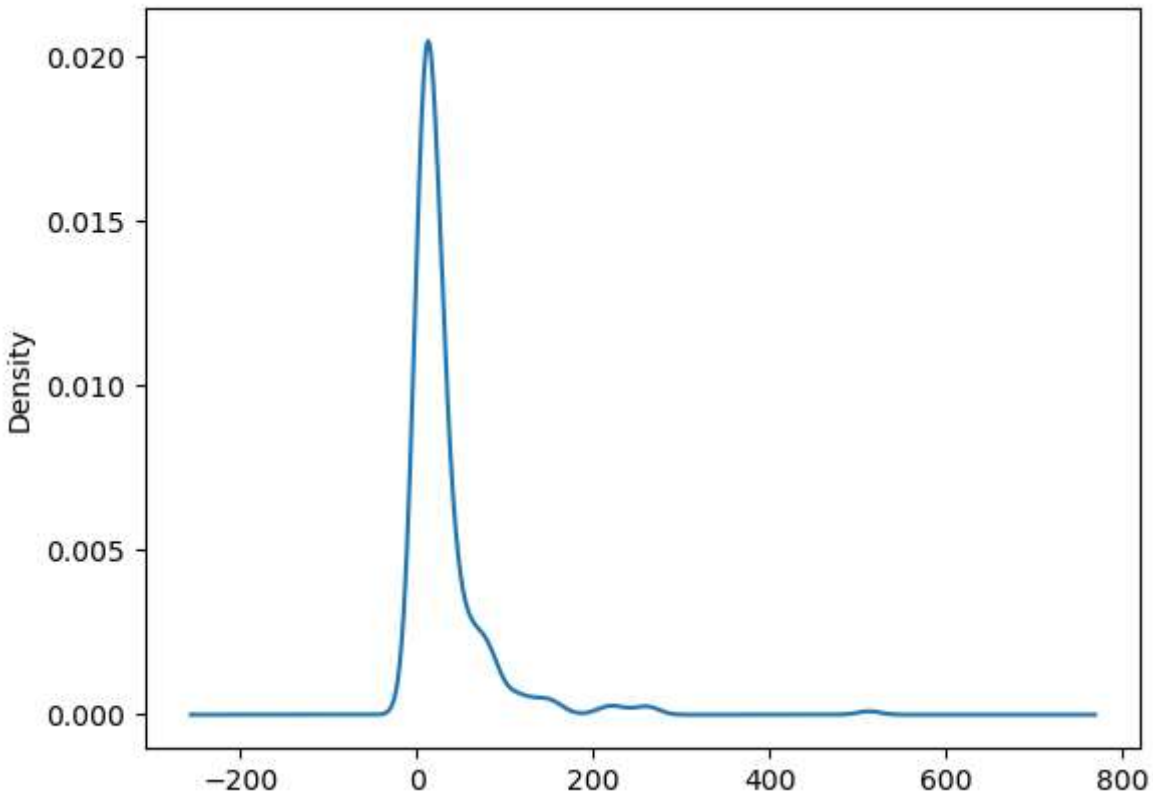
Out[17]:
```
count    891.000000
mean      32.204208
std       49.693429
min        0.000000
25%        7.910400
50%       14.454200
75%       31.000000
max      512.329200
Name: Fare, dtype: float64
```

In [18]:
```
df['Fare'].skew() #highly positive skews
```

Out[18]:
```
4.787316519674893
```

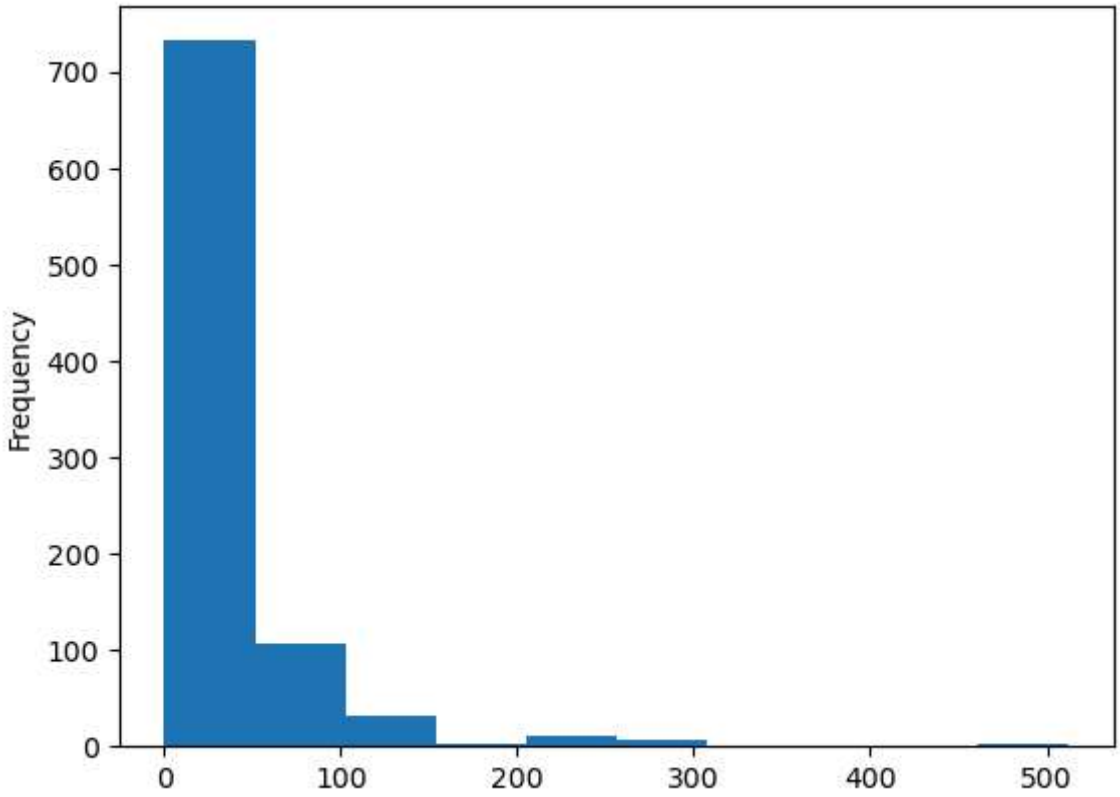
In [19]:
```
df['Fare'].plot(kind='kde')
```

Out[19]: <Axes: ylabel='Density'>



In [20]:
```
df['Fare'].plot(kind='hist')
```

Out[20]:
```
<Axes: ylabel='Frequency'>
```



In [21]:
```
df['Fare'].isnull().sum()
```

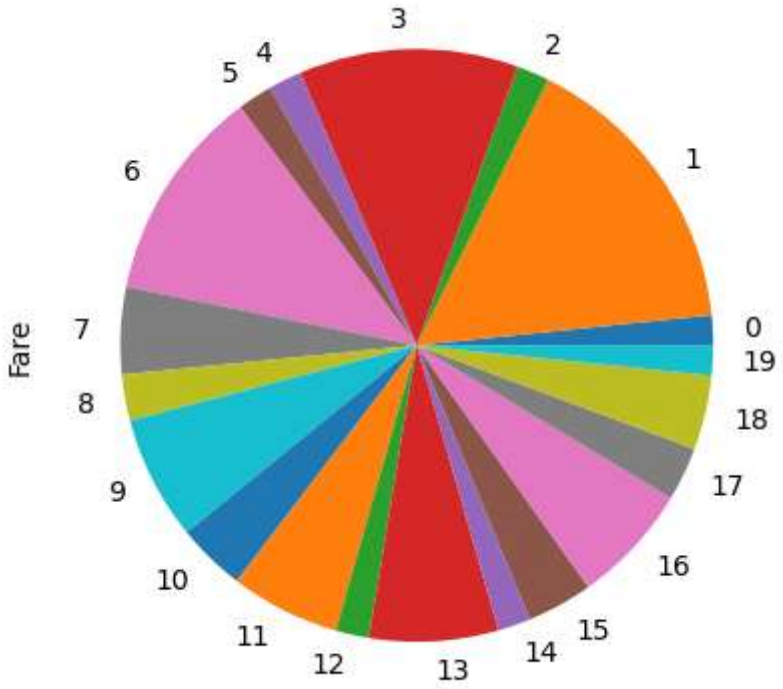
Out[21]:
```
0
```

In [22]:
```
len(df['Fare'])
```

Out[22]:
```
891
```

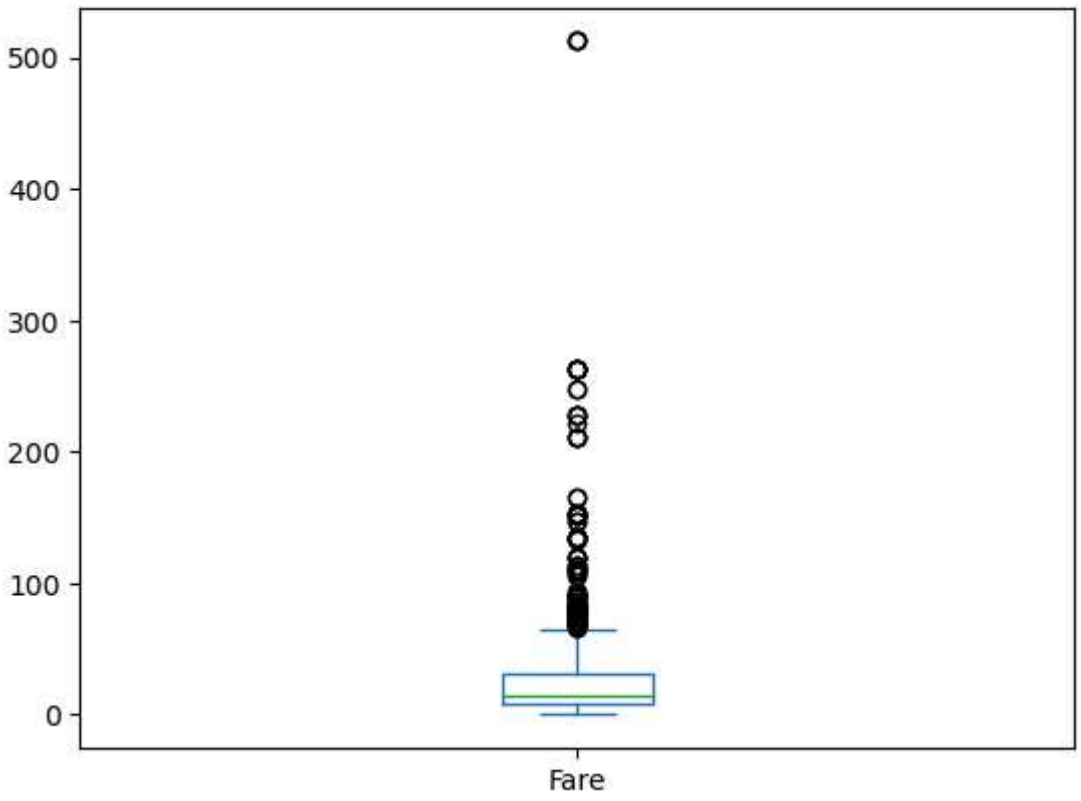
In [23]:
```
df['Fare'].head(20).plot(kind='pie')
```

Out[23]: <Axes: ylabel='Fare'>



In [24]:
```
df['Fare'].plot(kind='box')
```

Out[24]: <Axes: >



In [25]:
```
df[df['Fare']>265]
```

Out[25]:

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

In [26]:
```
len(df[df['Fare']>265])
```

Out[26]:
```
3
```

In [119]:
```
df['Fare'].isnull().sum()
```

Out[119]:
```
0
```

Univariate Analysis on Categorical columns

servived

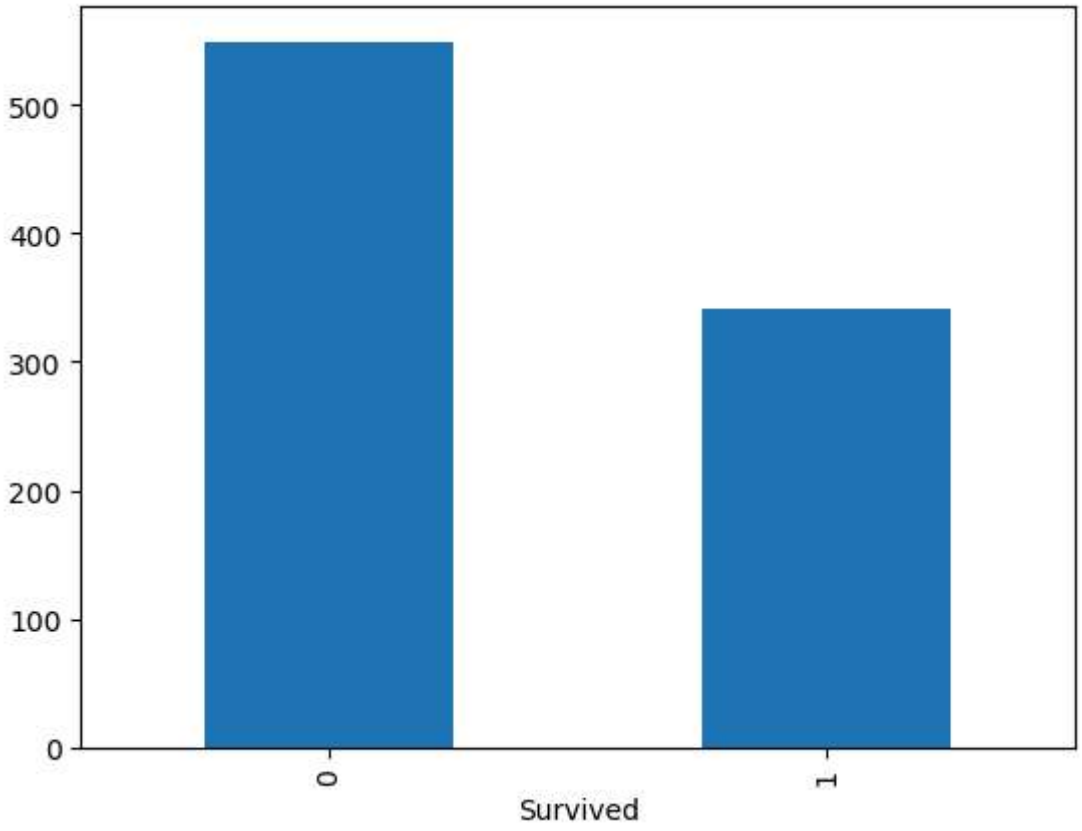
```
conclusions

1.Parch and SibSp cols can be merged to form a new col call family_size.
2.Create a new col called is_alone.
```

In [27]:
```
df['Survived'].value_counts()
```

Out[27]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

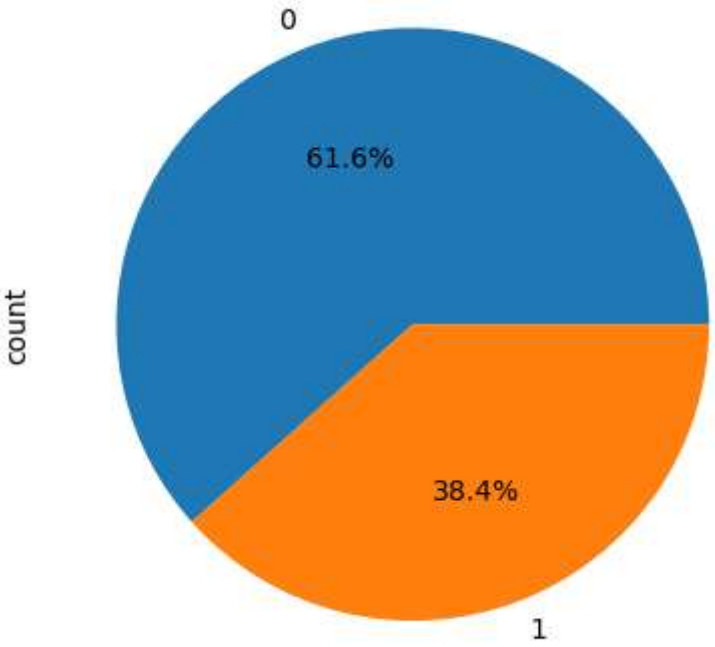
In [28]:
```
df['Survived'].value_counts().plot(kind='bar')
```

Out[28]:
```
<Axes: xlabel='Survived'>
```



In [29]:
```
df['Survived'].value_counts().plot(kind='pie',autopct="%0.1f%%")
```

Out[29]: <Axes: ylabel='count'>

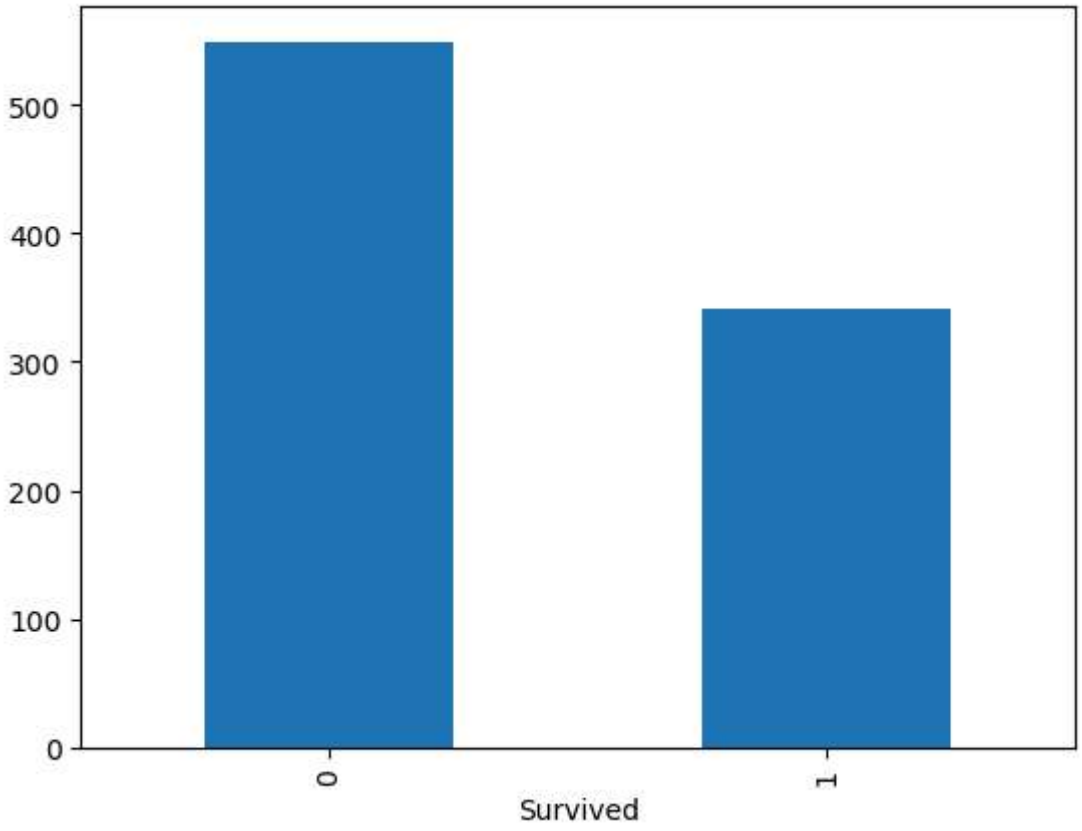


In [31]:
```
df['Survived'].value_counts()
```

Out[31]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

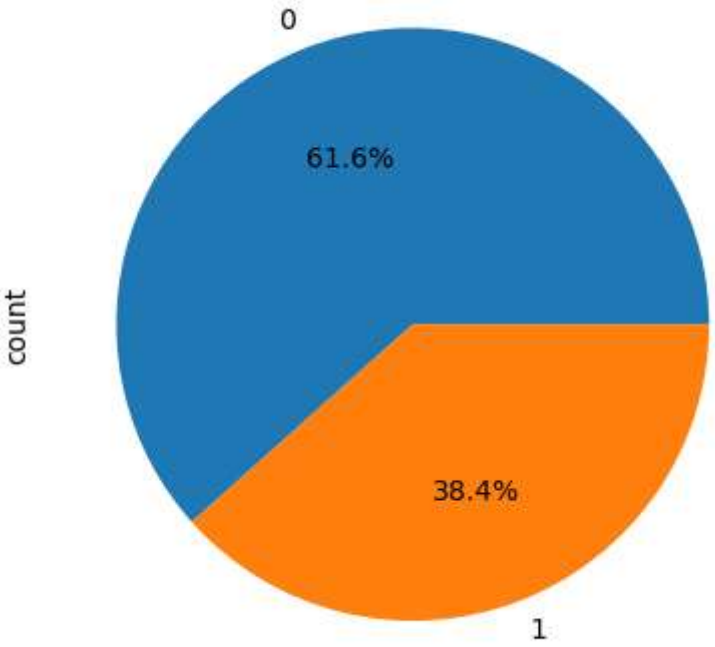
In [32]:
```
df['Survived'].value_counts().plot(kind='bar')
```

Out[32]:
```
<Axes: xlabel='Survived'>
```



In [33]:
```
df['Survived'].value_counts().plot(kind='pie',autopct='%0.1f%%') #important
```

Out[33]: <Axes: ylabel='count'>

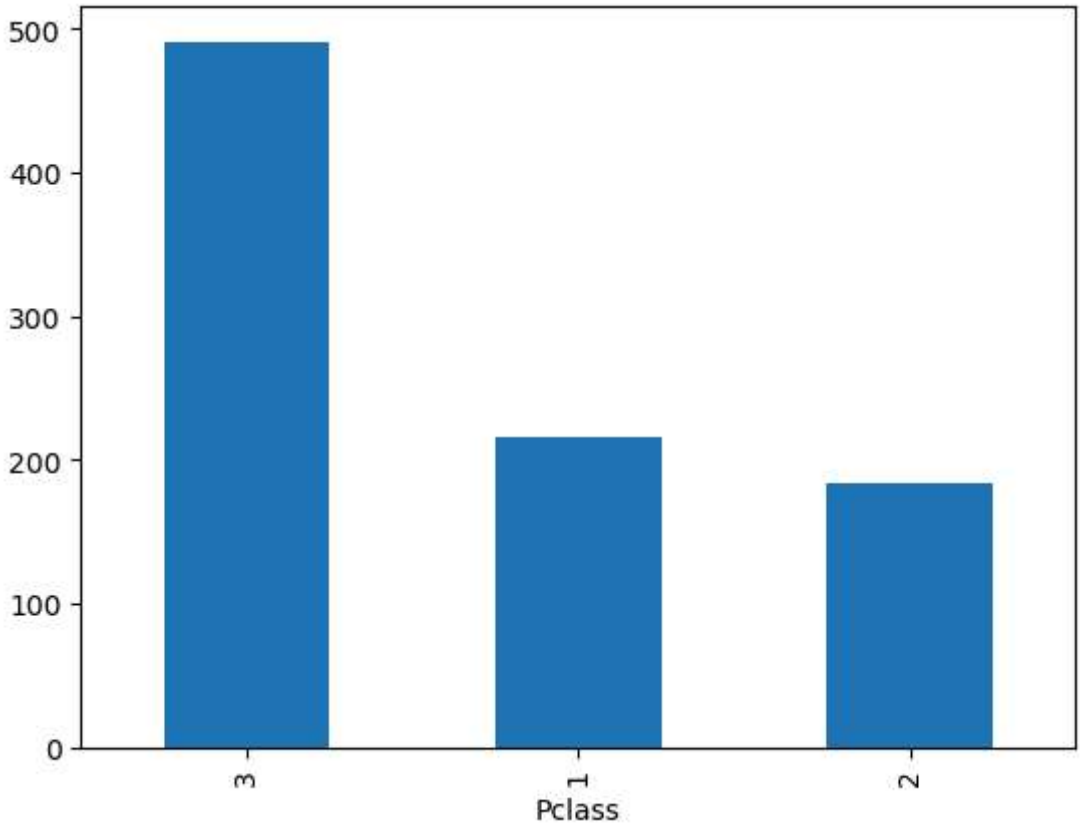


In [34]:
```
df["Pclass"].value_counts()
```

Out[34]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

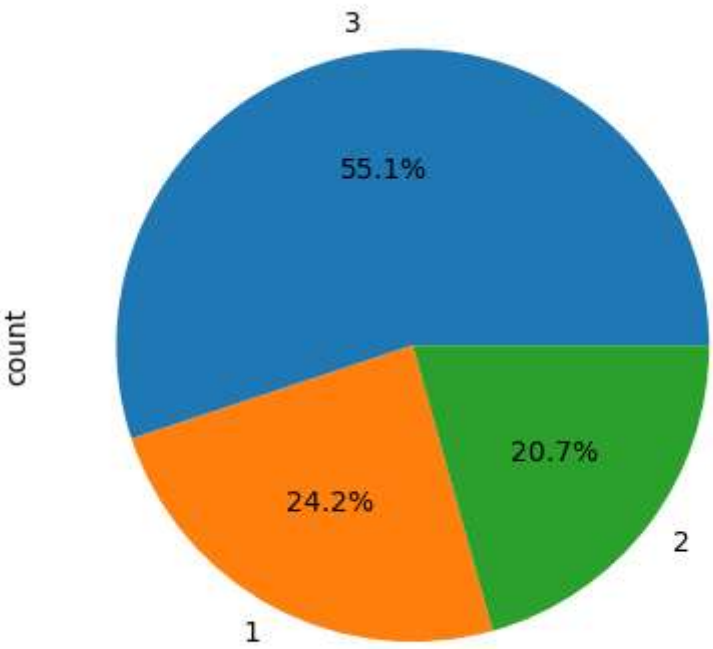
In [35]:
```
df["Pclass"].value_counts().plot(kind='bar')
```

Out[35]:
```
<Axes: xlabel='Pclass'>
```



In [36]:
```
df['Pclass'].value_counts().plot(kind='pie',autopct='%0.1f%%') #important
```

Out[36]: <Axes: ylabel='count'>

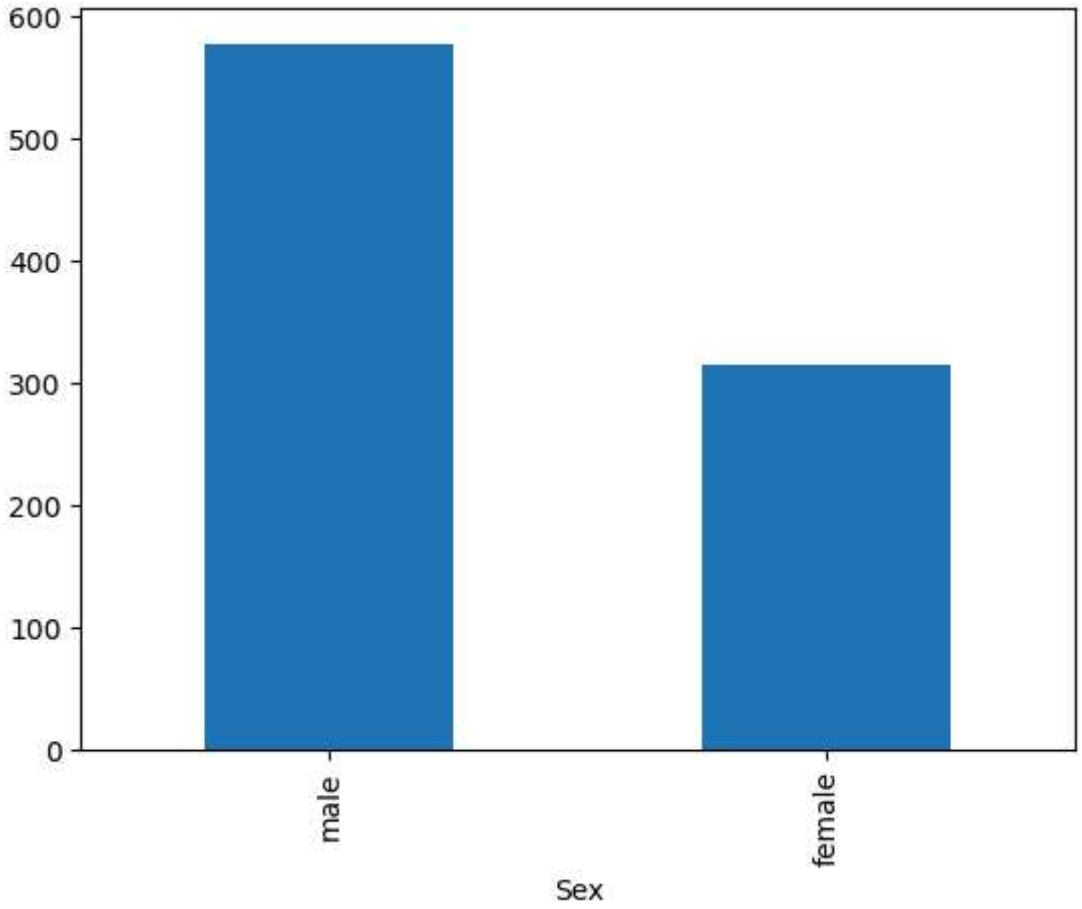


In [37]:
```
df["Sex"].value_counts()
```

Out[37]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

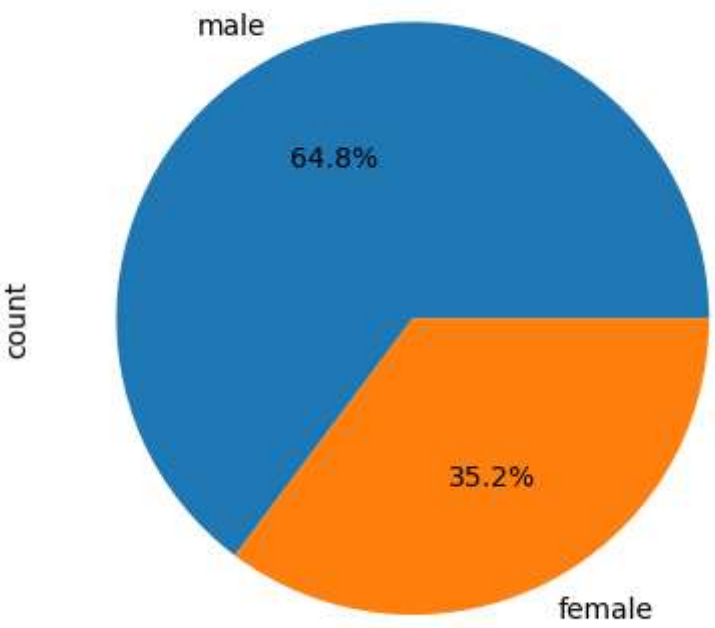
In [38]:
```
df["Sex"].value_counts().plot(kind='bar')
```

Out[38]: <Axes: xlabel='Sex'>



In [39]:
```
df['Sex'].value_counts().plot(kind='pie',autopct='%0.1f%%') #important
```

Out[39]:
```
<Axes: ylabel='count'>
```

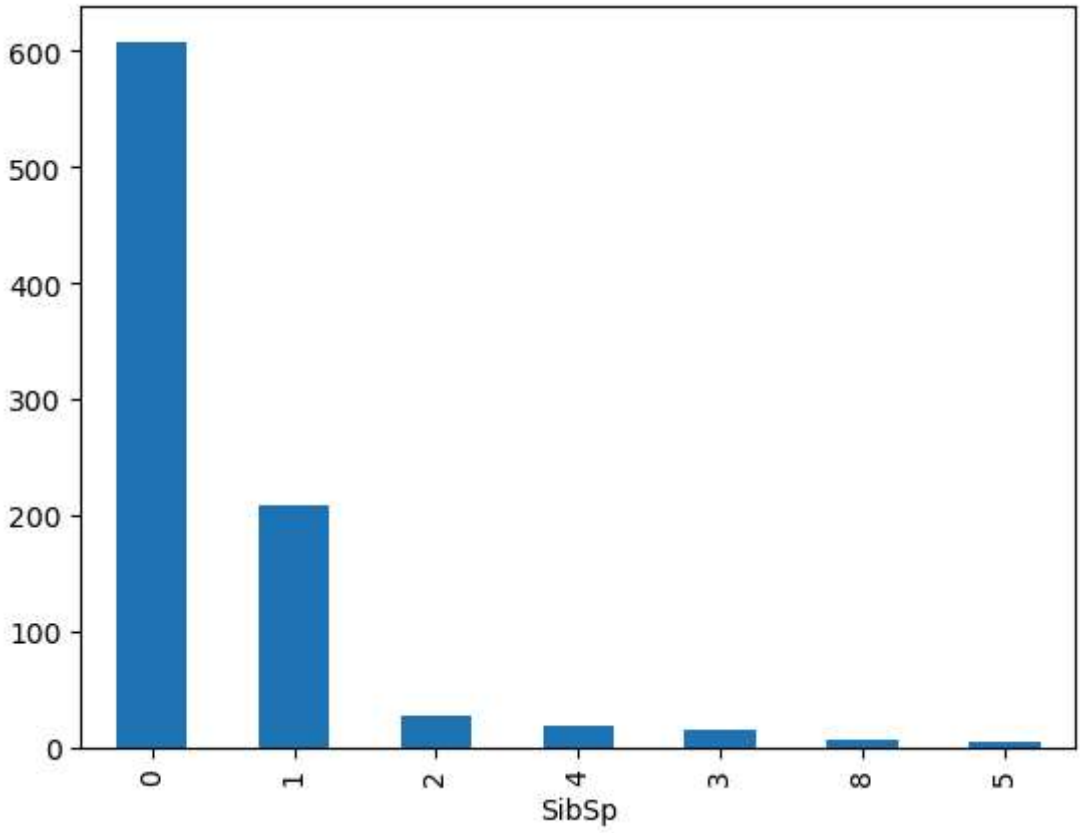


In [40]:
```
df['SibSp'].value_counts()
```

Out[40]:
```
SibSp
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: count, dtype: int64
```

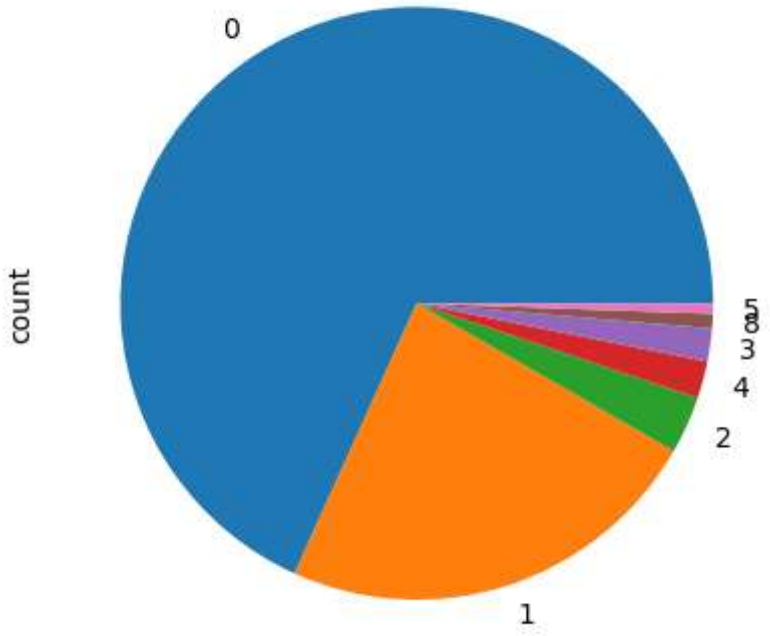
In [41]:
```
df['SibSp'].value_counts().plot(kind='bar')
```

Out[41]:
```
<Axes: xlabel='SibSp'>
```



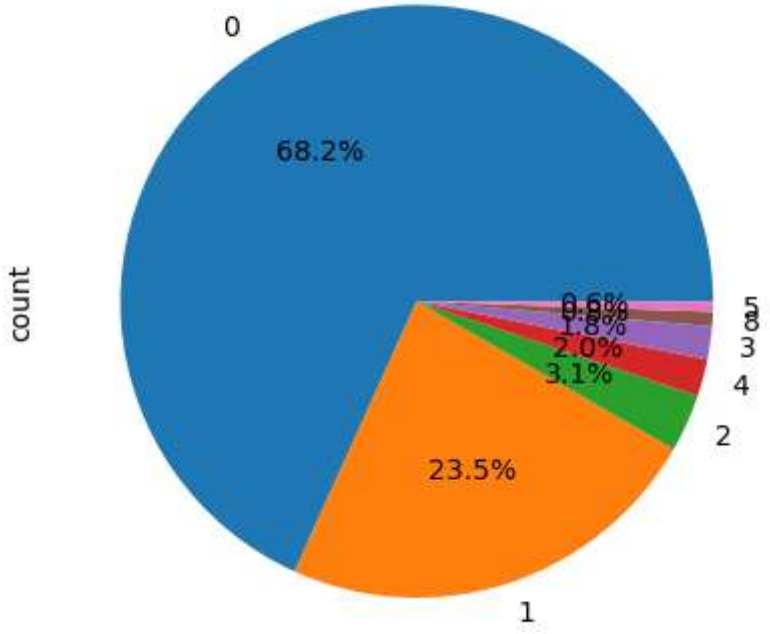
In [42]:
```
df['SibSp'].value_counts().plot(kind='pie')
```

Out[42]: <Axes: ylabel='count'>



In [43]:
```
df['SibSp'].value_counts().plot(kind='pie',autopct='%0.1f%%') #important
```

Out[43]: <Axes: ylabel='count'>

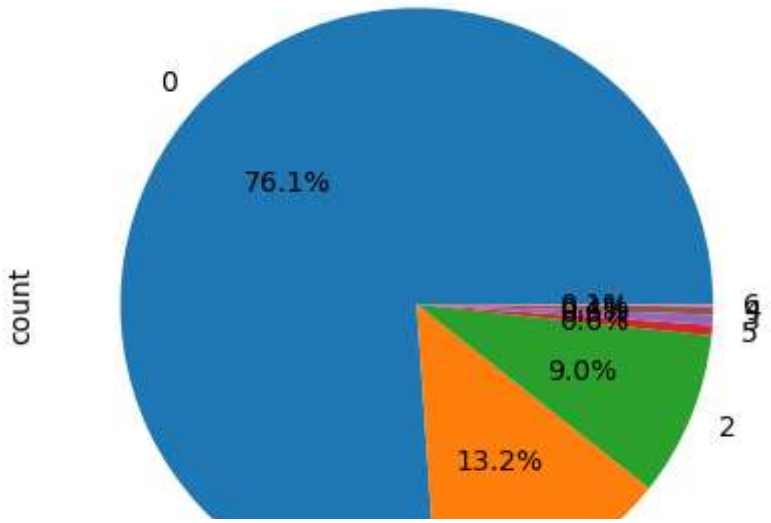


In [44]:
```
df['Parch'].value_counts()
```

Out[44]:
```
Parch
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: count, dtype: int64
```

In [45]:
```
df['Parch'].value_counbts().plot(kind='pie',autopct='%0.1f%%') #important
```

Out[45]:
```
<Axes: ylabel='count'>
```



In [120]:
```
df['Sex'].isnull().sum()
```

Out[120]:
```
0
```

In [46]:
```
# name, tickts , and cabin these all are mixed column.
```

In [47]:
```
df
```

Out[47]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
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	N
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	N
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	N
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	N

891 rows × 12 columns

Bivariate Analysis

Type *Markdown* and LaTeX: α^2

In [48]:
```
# select two column.
```

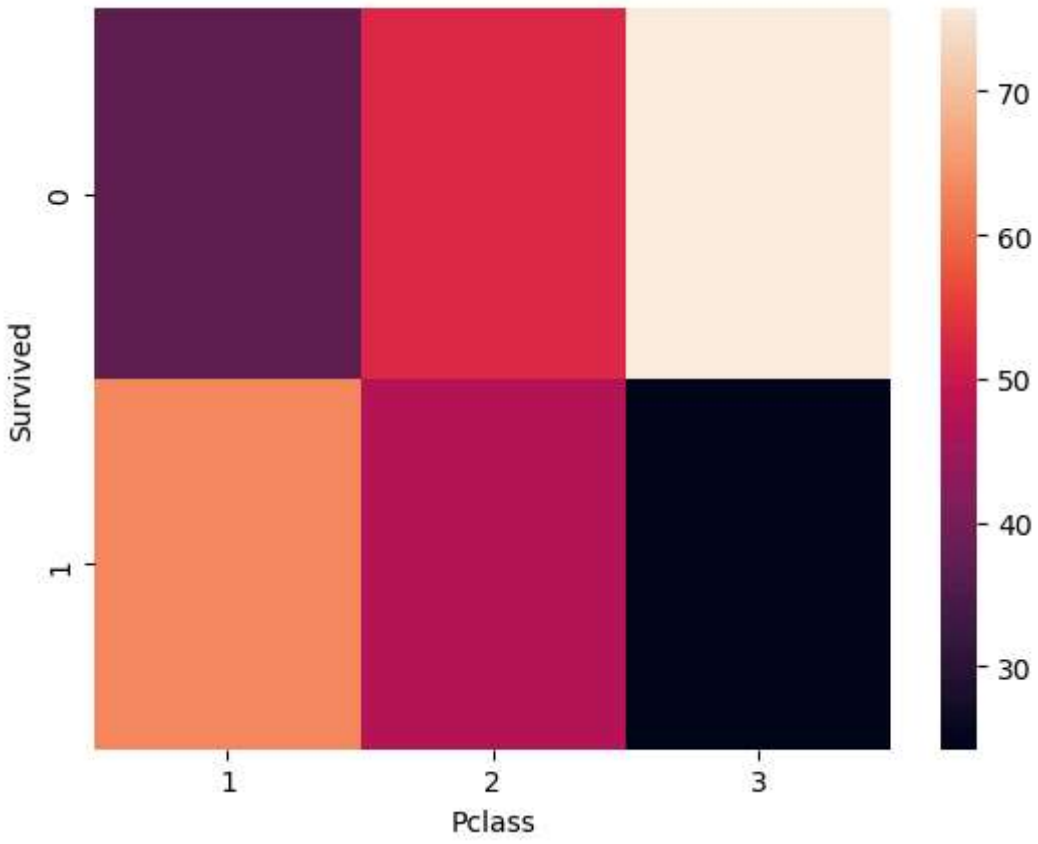
In [49]:
```
pd.crosstab(df['Survived'],df['Pclass'], normalize='columns')*100 # 0----> die
```

Out[49]:

Pclass	1	2	3
Survived			
0	37.037037	52.717391	75.763747
1	62.962963	47.282609	24.236253

In [50]:
```
sns.heatmap(pd.crosstab(df['Survived'],df['Pclass'], normalize='columns')*100)
```

Out[50]:
```
<Axes: xlabel='Pclass', ylabel='Survived'>
```



In [51]:
```
# travelling in pclass 3 is more dengeourus as compare to others
```

In [52]:
```
pd.crosstab(df['Survived'],df['Sex'], normalize='columns')*100
```

Out[52]:

Sex	female	male
Survived		
0	25.796178	81.109185
1	74.203822	18.890815

In [53]:
```
pd.crosstab(df['Survived'],df['Embarked'],normalize='columns')*100
```

Out[53]:

Embarked	C	Q	S
Survived			
0	44.642857	61.038961	66.304348
1	55.357143	38.961039	33.695652

In [54]:
```
# here we have two assumption
# 1. female is more servival as compare to the male passangers
# 2.the passanger pickup the titenic from Cherbourg is higher chance to serviv
```

In [55]:
```
pd.crosstab(df['Sex'],df['Embarked'],normalize='columns')*100
```

Out[55]:

Embarked	C	Q	S
Sex			
female	43.452381	46.753247	31.521739
male	56.547619	53.246753	68.478261

In [56]:
```
pd.crosstab(df['Pclass'],df['Embarked'],normalize='columns')*100 #  Cherbourg
# ameer log chade hai.
```

Out[56]:

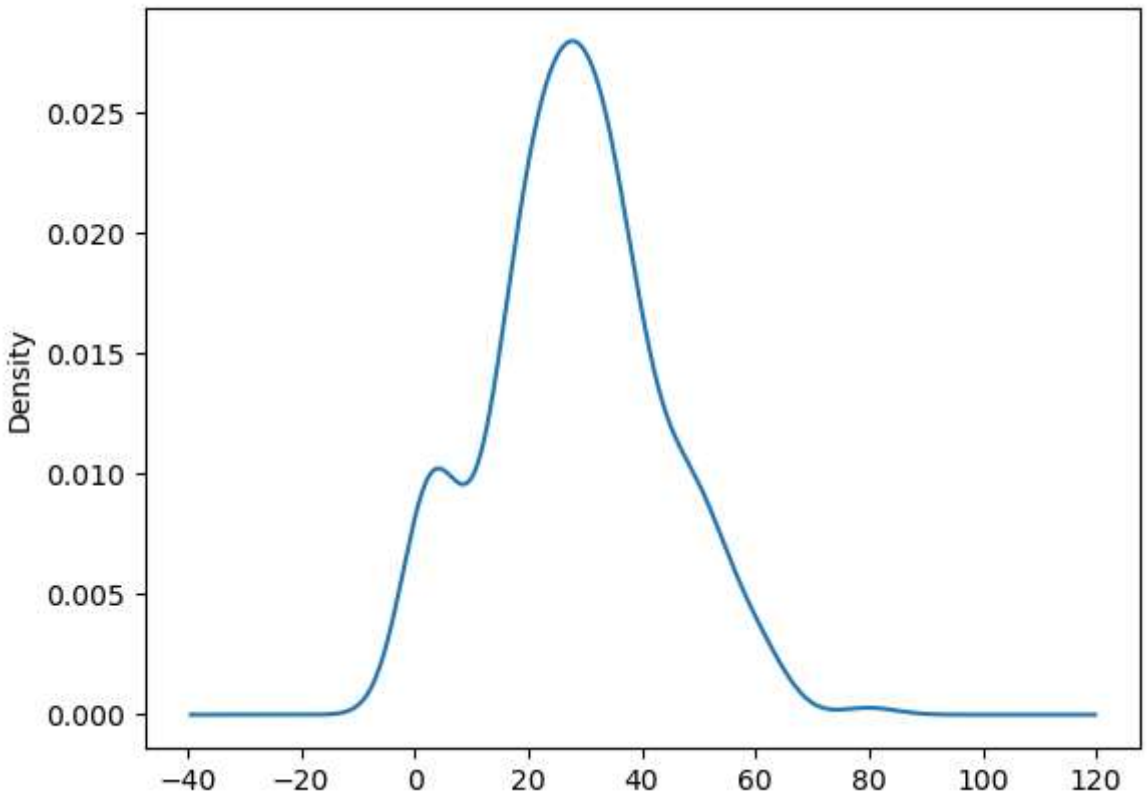
Embarked	C	Q	S
Pclass			
1	50.595238	2.597403	19.720497
2	10.119048	3.896104	25.465839
3	39.285714	93.506494	54.813665

In [ ]:
```

```

In [57]:
```
# servived vs age
```

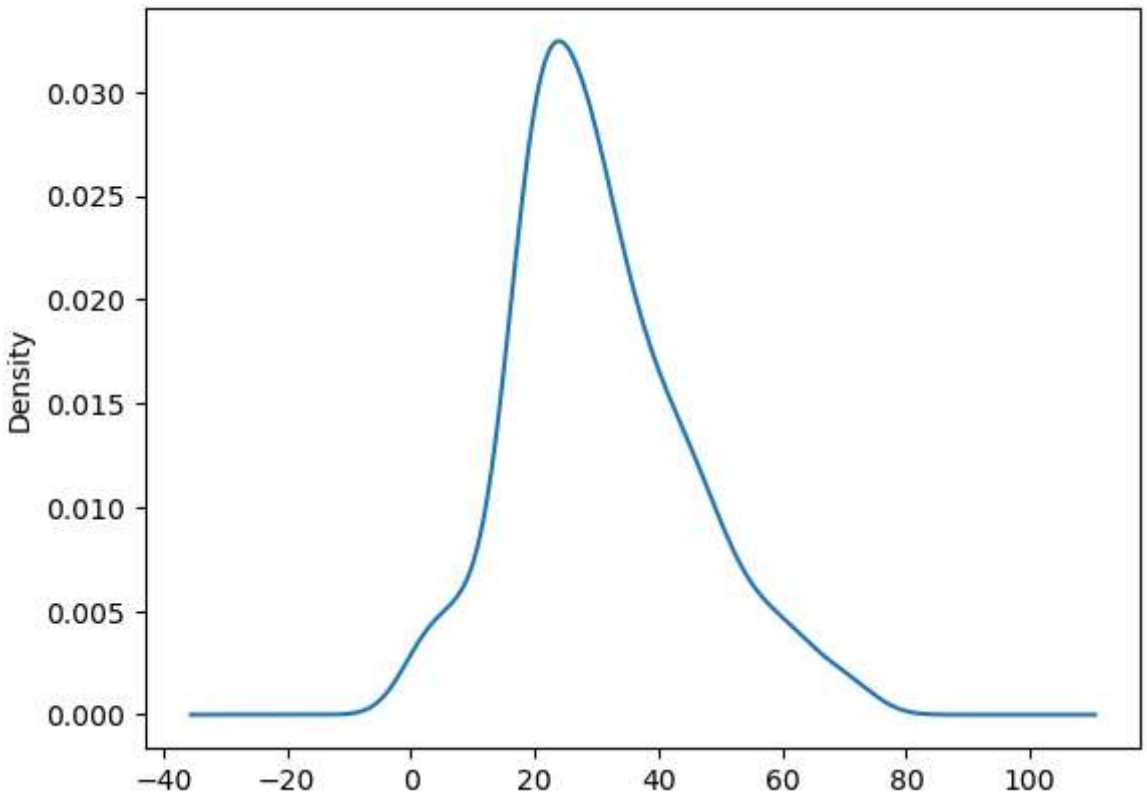
In [58]:
```
df[df['Survived']==1]['Age'].plot(kind='kde')
```

Out[58]:
```
<Axes: ylabel='Density'>
```

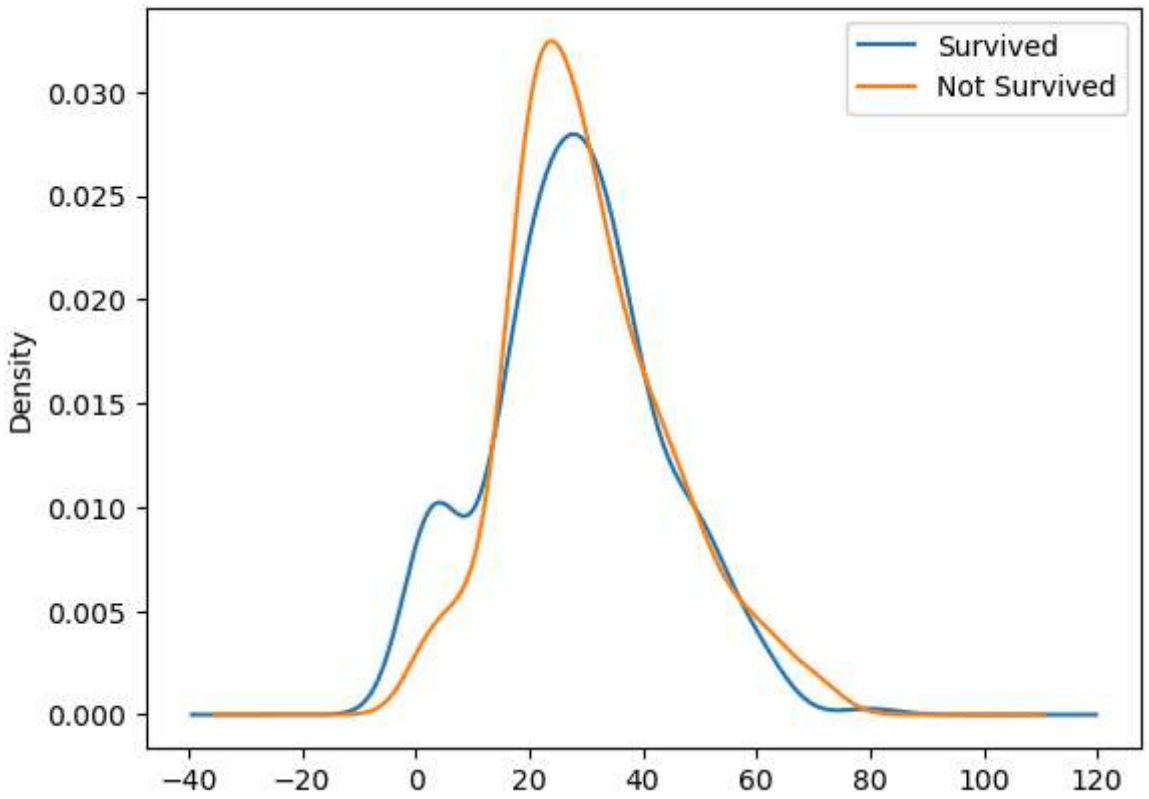


In [59]:
```
df[df['Survived'] == 0]['Age'].plot(kind='kde',label='Not Survived')
```

Out[59]: <Axes: ylabel='Density'>



In [60]:
```
df[df['Survived'] == 1]['Age'].plot(kind='kde',label='Survived')
df[df['Survived'] == 0]['Age'].plot(kind='kde',label='Not Survived')

plt.legend()
plt.show()
```



In [61]:
```
df[df['Pclass'] == 1]['Age'].mean()
```

Out[61]:
```
38.233440860215055
```

In [62]:
```
df
```

Out[62]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
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	N
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	N
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	N
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	N

891 rows × 12 columns

Feature Engineering on 'Fare' column

In [63]:
```
df['SibSp'].value_counts()
```

Out[63]:
```
SibSp
0    608
1    209
2     28
4     18
3     16
8      7
5      5
Name: count, dtype: int64
```

In [64]:
```
df[df['SibSp']==8]
```

Out[64]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
159	160	0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN	
180	181	0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN	
201	202	0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN	
324	325	0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN	
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN	
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN	
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN	

In [65]:
```
# there is one more dataset.
df1=pd.read_csv('test.csv')
```

In [66]:
```
df1
```

Out[66]:

	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 [67]:
```
df = pd.concat([df,df1])
```

In [68]:
```
df
```

Out[68]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583

1309 rows × 12 columns

In [69]:
```
df[df['Ticket'] == 'CA. 2343']
```

Out[69]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
159	160	0.0	3	Sage, Master. Thomas Henry	male	NaN	8	2	CA. 2343	69.55	NaN	
180	181	0.0	3	Sage, Miss. Constance Gladys	female	NaN	8	2	CA. 2343	69.55	NaN	
201	202	0.0	3	Sage, Mr. Frederick	male	NaN	8	2	CA. 2343	69.55	NaN	
324	325	0.0	3	Sage, Mr. George John Jr	male	NaN	8	2	CA. 2343	69.55	NaN	
792	793	0.0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.55	NaN	
846	847	0.0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.55	NaN	
863	864	0.0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.55	NaN	
188	1080	NaN	3	Sage, Miss. Ada	female	NaN	8	2	CA. 2343	69.55	NaN	
342	1234	NaN	3	Sage, Mr. John George	male	NaN	1	9	CA. 2343	69.55	NaN	
360	1252	NaN	3	Sage, Master. William Henry	male	14.5	8	2	CA. 2343	69.55	NaN	
365	1257	NaN	3	Sage, Mrs. John (Annie Bullen)	female	NaN	1	9	CA. 2343	69.55	NaN	

In [70]:
```
# we can check the members by ticket number.
df[df['Ticket'] == 'CA 2144']
```

Out[70]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
59	60	0.0	3	Goodwin, Master. William Frederick	male	11.0	5	2	CA 2144	46.9	NaN	
71	72	0.0	3	Goodwin, Miss. Lillian Amy	female	16.0	5	2	CA 2144	46.9	NaN	
386	387	0.0	3	Goodwin, Master. Sidney Leonard	male	1.0	5	2	CA 2144	46.9	NaN	
480	481	0.0	3	Goodwin, Master. Harold Victor	male	9.0	5	2	CA 2144	46.9	NaN	
678	679	0.0	3	Goodwin, Mrs. Frederick (Augusta Tyler)	female	43.0	1	6	CA 2144	46.9	NaN	
683	684	0.0	3	Goodwin, Mr. Charles Edward	male	14.0	5	2	CA 2144	46.9	NaN	
139	1031	NaN	3	Goodwin, Mr. Charles Frederick	male	40.0	1	6	CA 2144	46.9	NaN	
140	1032	NaN	3	Goodwin, Miss. Jessie Allis	female	10.0	5	2	CA 2144	46.9	NaN	

In [71]:
```
df["Ticket"].value_counts()
```

Out[71]:
```
Ticket
CA. 2343        11
CA 2144          8
1601             8
PC 17608         7
S.O.C. 14879     7
                ..
113792           1
36209            1
323592           1
315089           1
359309           1
Name: count, Length: 929, dtype: int64
```

In [72]:
```
df[df['Ticket'] == 'PC 17608']
```

Out[72]:

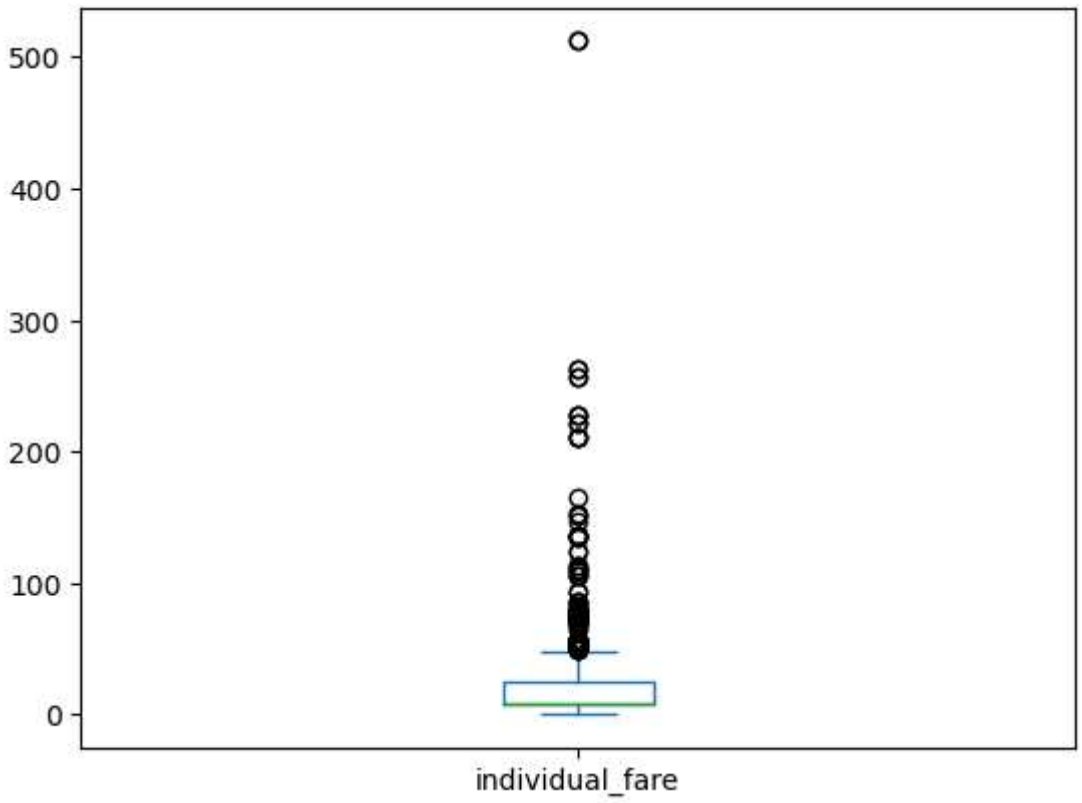
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
311	312	1.0	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.375	B5 B5 B6 B6
742	743	1.0	1	Ryerson, Miss. Susan Parker "Suzette"	female	21.0	2	2	PC 17608	262.375	B5 B5 B6 B6
24	916	NaN	1	Ryerson, Mrs. Arthur Larned (Emily Maria Borie)	female	48.0	1	3	PC 17608	262.375	B5 B5 B6 B6
59	951	NaN	1	Chaudanson, Miss. Victorine	female	36.0	0	0	PC 17608	262.375	B6
64	956	NaN	1	Ryerson, Master. John Borie	male	13.0	2	2	PC 17608	262.375	B5 B5 B6 B6
142	1034	NaN	1	Ryerson, Mr. Arthur Larned	male	61.0	1	3	PC 17608	262.375	B5 B5 B6 B6
375	1267	NaN	1	Bowen, Miss. Grace Scott	female	45.0	0	0	PC 17608	262.375	Na

Finding Indivisual column

In [73]:
```
df['individual_fare'] = df['Fare']/(df['SibSp'] + df['Parch'] + 1) #try to fin
```

In [74]:
```
df['individual_fare'].plot(kind='box')
```

Out[74]: <Axes: >



In [75]:
```
df[['individual_fare','Fare']].describe()
```

Out[75]:

	individual_fare	Fare
count	1308.000000	1308.000000
mean	20.518215	33.295479
std	35.774337	51.758668
min	0.000000	0.000000
25%	7.452767	7.895800
50%	8.512483	14.454200
75%	24.237500	31.275000
max	512.329200	512.329200

In [76]:
```
# now we have individual fare.
df['Fare']
```

Out[76]:
```
0        7.2500
1       71.2833
2        7.9250
3       53.1000
4        8.0500
          ...
413      8.0500
414    108.9000
415      7.2500
416      8.0500
417     22.3583
Name: Fare, Length: 1309, dtype: float64
```

In [77]:
```
df
```

Out[77]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583

1309 rows × 13 columns

In [78]:
```
# creating a new column name of "family size"
```

In [79]:
```
df['family_size'] = df['SibSp'] + df['Parch'] + 1
```

In [80]:
```
# family_type
# 1 -> alone
# 2-4 -> small
# >5 -> large
# ----------------------------

def transform_family_size(num):

  if num == 1:
    return 'alone'
  elif num>1 and num <5:
    return "small"
  else:
    return "large"
```

In [81]:
```
df['family_size'].apply(transform_family_size)
```

Out[81]:
```
0      small
1      small
2      alone
3      small
4      alone
       ...
413    alone
414    alone
415    alone
416    alone
417    small
Name: family_size, Length: 1309, dtype: object
```

In [82]:
```
df['family_size']=df['family_size'].apply(transform_family_size)
```

In [83]:
```
df
```

Out[83]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583

1309 rows × 14 columns

Creation of new column which is predict the your servival chances.

In [84]:
```
pd.crosstab(df['Survived'],df['family_size'],normalize="columns")*100
```

Out[84]:

family_size	alone	large	small
Survived			
0.0	69.646182	83.870968	42.123288
1.0	30.353818	16.129032	57.876712

Creation of new column of surname

In [85]:
```
df['Name'].str.split(',').str.get(0)
```

Out[85]:
```
0             Braund
1            Cumings
2          Heikkinen
3           Futrelle
4              Allen
           ...
413          Spector
414    Oliva y Ocana
415          Saether
416             Ware
417            Peter
Name: Name, Length: 1309, dtype: object
```

In [86]:
```
df
```

Out[86]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583

1309 rows × 14 columns

In [87]:
```
df["Name"]
```

Out[87]:
```
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
                             ...
413                                     Spector, Mr. Woolf
414                           Oliva y Ocana, Dona. Fermina
415                           Saether, Mr. Simon Sivertsen
416                                    Ware, Mr. Frederick
417                               Peter, Master. Michael J
Name: Name, Length: 1309, dtype: object
```

Extrect the title of name

In [88]:
```
df['title'] = df['Name'].str.split(',').str.get(1).str.strip().str.split(' ').
```

In [89]:
```
temp_df = df[df['title'].isin(['Mr.','Miss.','Mrs.','Master.','ootherr'])]
```

In [90]:
```
pd.crosstab(temp_df['Survived'],temp_df['title'],normalize='columns')*100
```

Out[90]:

title	Master.	Miss.	Mr.	Mrs.
Survived				
0.0	42.5	30.21978	84.332689	20.8
1.0	57.5	69.78022	15.667311	79.2

In [91]:
```
df['title'] = df['title'].str.replace('Rev.','other')
df['title'] = df['title'].str.replace('Dr.','other')
df['title'] = df['title'].str.replace('Col.','other')
df['title'] = df['title'].str.replace('Major.','other')
df['title'] = df['title'].str.replace('Capt.','other')
df['title'] = df['title'].str.replace('the','other')
df['title'] = df['title'].str.replace('Jonkheer.','other')
# ,'Dr.','Col.','Major.','Don.','Capt.','the','Jonkheer.']
```

Analyse the servival rate by cabin

In [92]:
```
df['Cabin'].isnull().sum()/len(df['Cabin'])
```

Out[92]:
```
0.774637127578304
```

In [93]:
```
df['Cabin'].fillna('M',inplace=True)
```

In [94]:
```
df['Cabin'].value_counts()
```

Out[94]:
```
Cabin
M                  1014
C23 C25 C27           6
B57 B59 B63 B66       5
G6                    5
F33                   4
                    ...
A14                   1
E63                   1
E12                   1
E38                   1
C105                  1
Name: count, Length: 187, dtype: int64
```

In [95]:
```
df['deck'] = df['Cabin'].str[0]
```

In [96]:
```
df
```

Out[96]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
413	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500
414	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000
415	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500
416	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500
417	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583

1309 rows × 16 columns

In [97]:
```
df['deck'].value_counts()
```

Out[97]:
```
deck
M    1014
C      94
B      65
D      46
E      41
A      22
F      21
G       5
T       1
Name: count, dtype: int64
```

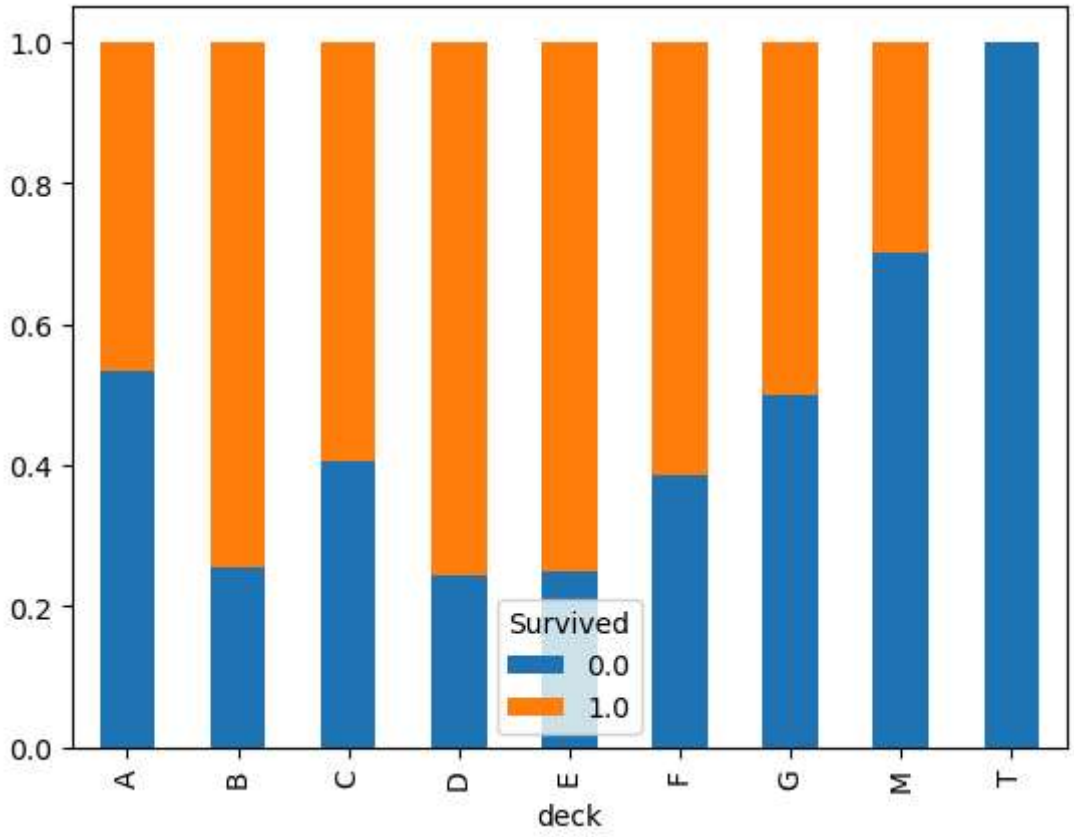
In [98]:
```
pd.crosstab(df['deck'],df['Pclass'])
```

Out[98]:

Pclass	1	2	3
deck			
A	22	0	0
B	65	0	0
C	94	0	0
D	40	6	0
E	34	4	3
F	0	13	8
G	0	0	5
M	67	254	693
T	1	0	0

In [99]:
```
pd.crosstab(df['deck'],df['Survived'],normalize='index').plot(kind='bar',stack
```

Out[99]: <Axes: xlabel='deck'>



In [100]:
```
df.corr
```

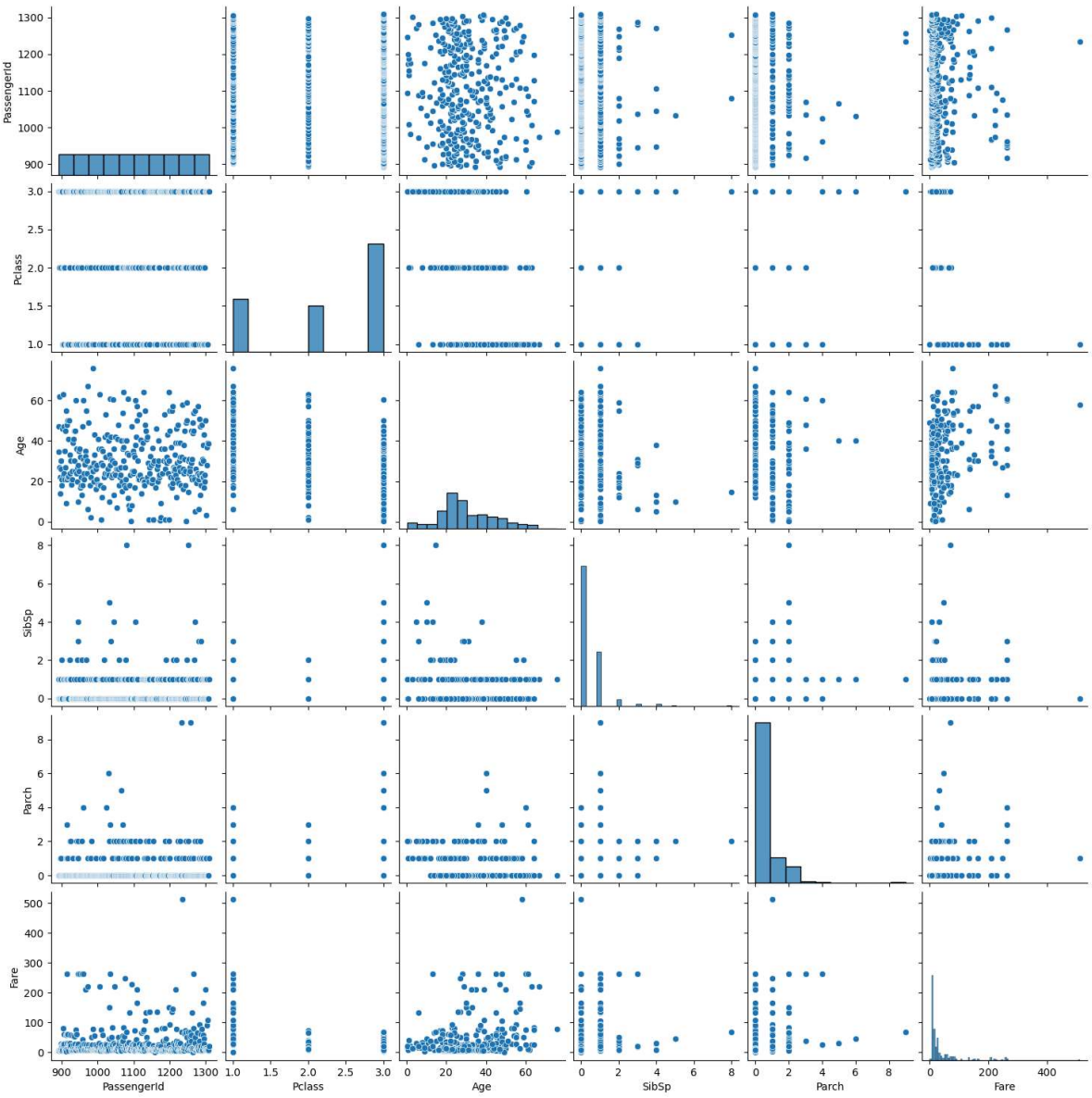
Out[100]:
```
<bound method DataFrame.corr of      PassengerId  Survived  Pclass  \
0              1       0.0       3
1              2       1.0       1
2              3       1.0       3
3              4       1.0       1
4              5       0.0       3
..           ...       ...     ...
413         1305       NaN       3
414         1306       NaN       1
415         1307       NaN       3
416         1308       NaN       3
417         1309       NaN       3

                                                 Name     Sex   Age  SibS
p  \
0                             Braund, Mr. Owen Harris    male  22.0
1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
```

Multivarient analysis

In [101]:
```
sns.pairplot(df1)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarni
ng: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[101]: <seaborn.axisgrid.PairGrid at 0x28cc55cced0>



In [102]:
```
df1
```

Out[102]:

	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

Conclusion:

```
1.About 33% survived the Titanic disaster.
2.Females were given higher priority in the rescue operation than male.so
females are more likely to survive.
3.Those who paid for first class are more likely to survive. Like The first
class people were given higher priority than the second class than the third
class.
4.Embarked - Those who embarked at 'C' have a higher chance at survival.
5.The features such as Age, Siblings Onboard and Parents onboard didn't have
major influence on the survival probability.
```

```
6.A better way for filling the missing values for Age column is explained by
extracting info from Name column.
```

Thank you