In [3]:
```
#importing pandas
import numpy as np
import pandas as pd
```

In [5]:
```
# reading the csv file

df = pd.read_csv("data.csv")
```

In [6]:
```
#seeing the dimension of the file

df.shape
```

Out[6]:
```
(891, 12)
```

In [7]:
```
#seeing the names of varibales in the dataset

df.columns
```

Out[7]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [8]:
```
#seeing the top 5 rows

df.head()
```

Out[8]:

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

In [9]:
```
df['Pclass'].dtypes
```

Out[9]:
```
dtype('int64')
```

In [10]:
```
df.dtypes
```

Out[10]:

```
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 [11]:

```
df.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 [12]:

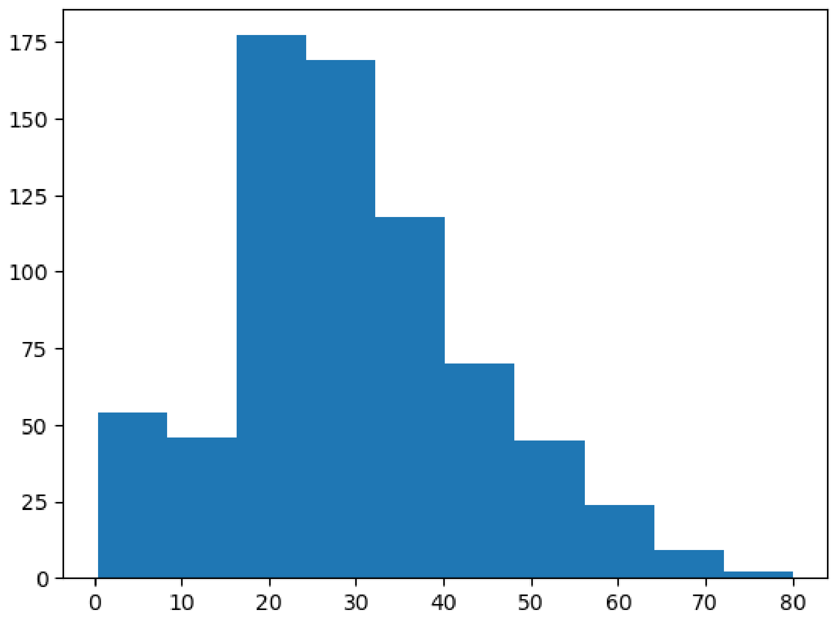
```
import matplotlib.pyplot as plt
```

In [13]:

```
plt.hist(df['Age'])
```

Out[13]:

```
(array([ 54.,  46., 177., 169., 118.,  70.,  45.,  24.,   9.,   2.]),
 array([ 0.42 ,  8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,
        64.084, 72.042, 80.   ]),
 <BarContainer object of 10 artists>)
```

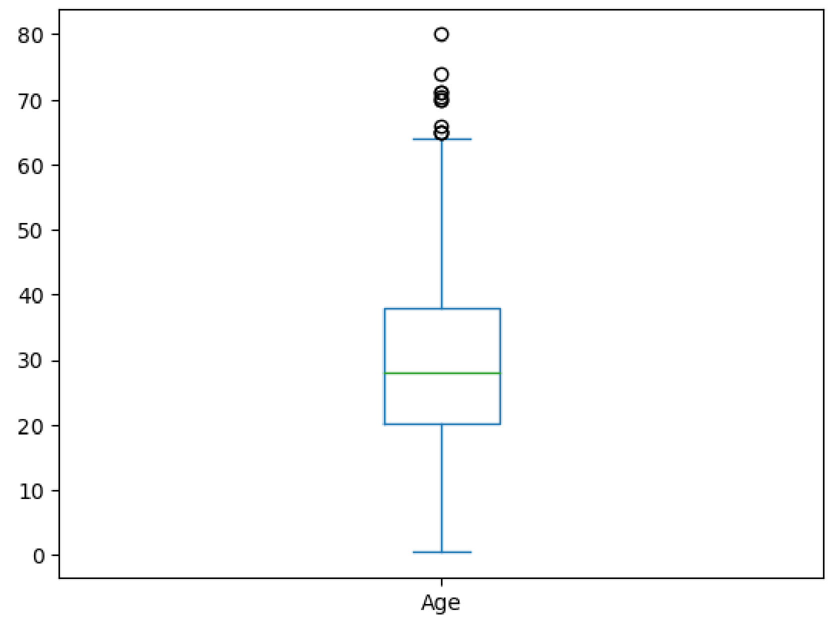


```
In [14]: df.dtypes
```

```
Out[14]: 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 [15]: # plotting a box plot
         df['Age'].plot.box()
```

```
Out[15]: <Axes: >
```

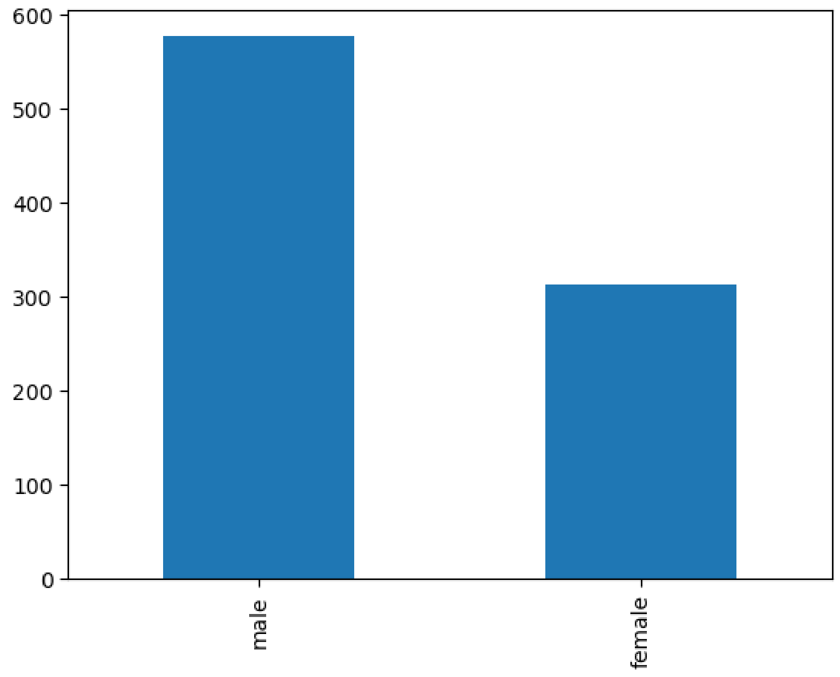


```
In [16]: df['Sex'].value_counts()
```

```
Out[16]: male      577
         female    314
         Name: Sex, dtype: int64
```

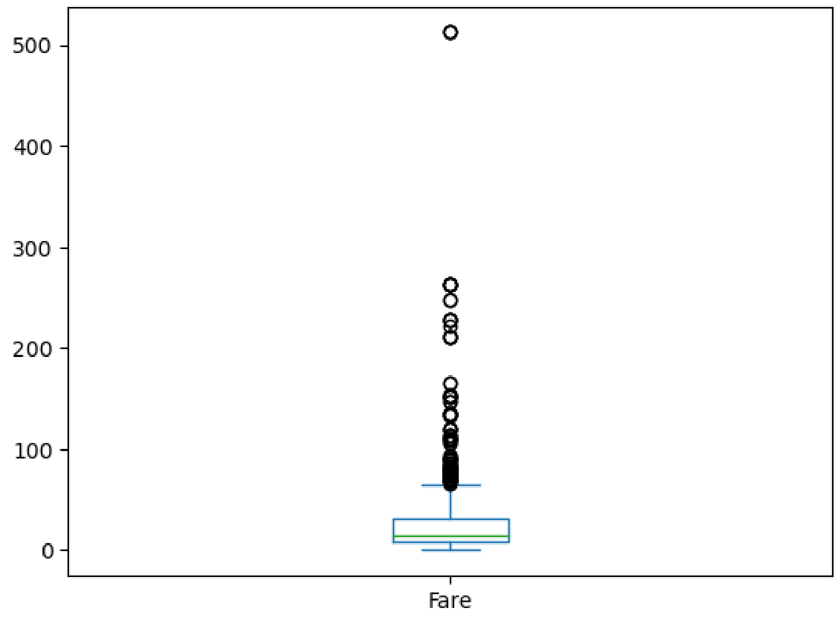
```
In [17]: df['Sex'].value_counts().plot.bar()
```

```
Out[17]: <Axes: >
```



In [18]:
```
df['Fare'].plot.box()
```

Out[18]:
```
<Axes: >
```

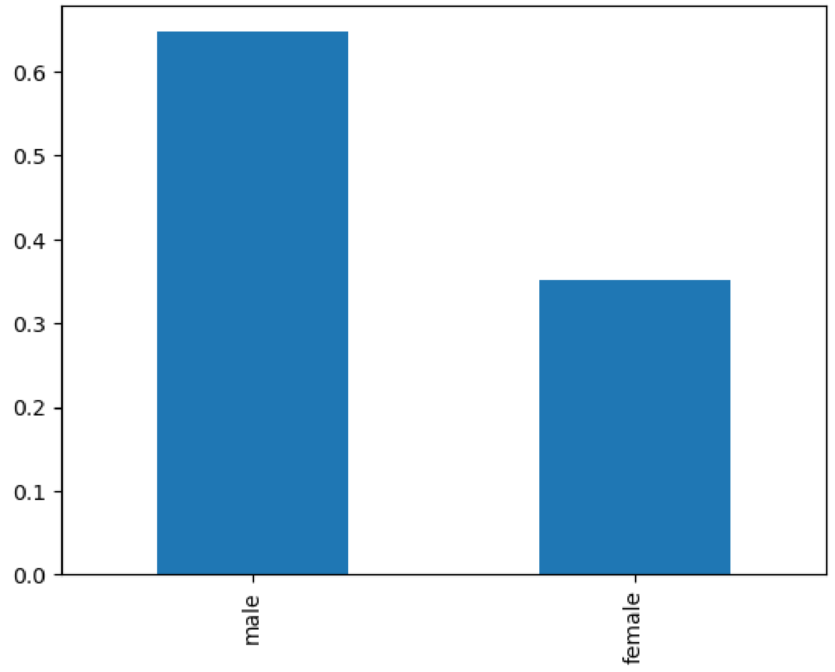


```
In [19]: df['Sex'].value_counts()/len(df['Sex'])
```

```
Out[19]: male      0.647587
         female    0.352413
         Name: Sex, dtype: float64
```

```
In [20]: (df['Sex'].value_counts()/len(df['Sex'])).plot.bar()
```

```
Out[20]: <Axes: >
```

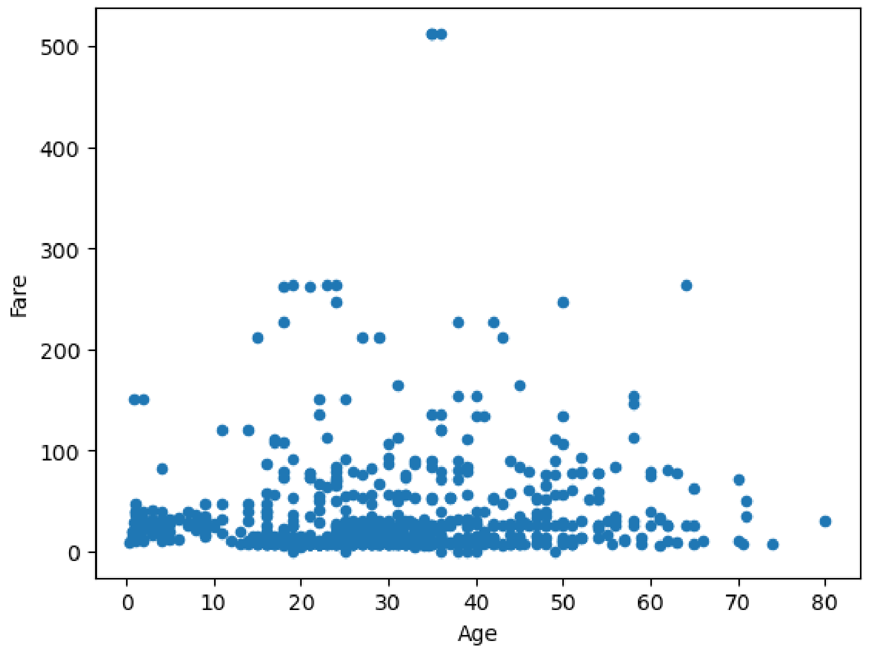


```
In [21]: df.dtypes
```

```
Out[21]: 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 [22]: df.plot.scatter('Age','Fare')
```

```
Out[22]: <Axes: xlabel='Age', ylabel='Fare'>
```



In [23]:
```
df.corr()
```

```
C:\Users\Red Devil\AppData\Local\Temp\ipykernel_3540\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a fut
ure version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

Out[23]:

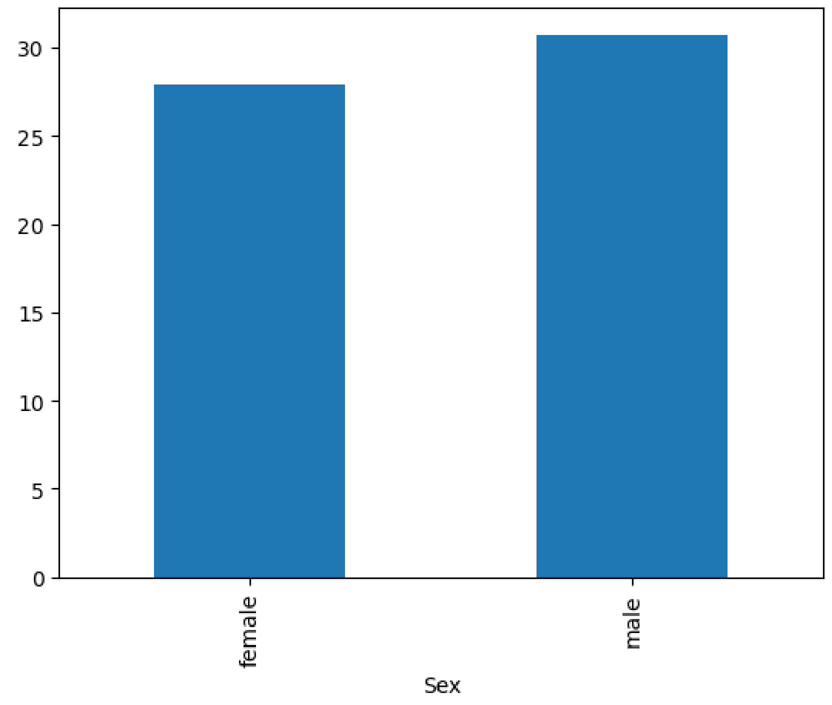
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

In [24]:
```
df['Age'].corr(df['Fare'])
```

Out[24]:
```
0.0960666917690389
```

In [25]:
```
df.groupby('Sex')['Age'].mean().plot.bar()
```

Out[25]:
```
<Axes: xlabel='Sex'>
```



In [26]:
```
import scipy.stats as stats
from scipy.stats import ttest_ind
```

In [27]:
```
males=df[df['Sex']=='male']
females=df[df['Sex']=='female']
```

In [28]:
```
ttest_ind(males['Age'],females['Age'],nan_policy='omit')
```

Out[28]:
```
Ttest_indResult(statistic=2.499206354920835, pvalue=0.012671296797013709)
```

Categorical - Categorical Bivariate Analysis

In [29]:
```
pd.crosstab(df['Sex'],df['Survived'])
```

Out[29]:

Survived	0	1
Sex		
female	81	233
male	468	109

In [30]:
```
from scipy.stats import chi2_contingency
```

In [31]:
```python
chi2_contingency(pd.crosstab(df['Sex'],df['Survived']))
```
Out[31]:
```
Chi2ContingencyResult(statistic=260.71702016732104, pvalue=1.1973570627755645e-58, dof=1, expected_freq=array([[193.47474747, 120.52525253],
       [355.52525253, 221.47474747]]))
```
In [32]:
```python
df.isnull()
```
Out[32]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

In [33]:
```python
df.isnull().sum()
```
Out[33]:
```
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 [34]:
```python
#dropping missing value in row
df.dropna().isnull().sum()
```

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

In [35]:
```
df.dropna(how='all').shape
```

Out[35]:
```
(891, 12)
```

In [36]:
```
df.dropna(axis=1)
```

Out[36]:

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

891 rows × 9 columns

In [37]:
```
df.dropna(axis=1).shape
```

Out[37]:
```
(891, 9)
```

In [38]:
```
df.dropna(axis=1,how='all').shape
```

Out[38]:
```
(891, 12)
```

In [39]:
```
df.fillna(0)
```

Out[39]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	0	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	0	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	0	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	0	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	0.0	1	2	W./C. 6607	23.4500	0	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	0	Q

891 rows × 12 columns

In [40]:
```
df['Age'].fillna(0)
```

Out[40]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     0.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

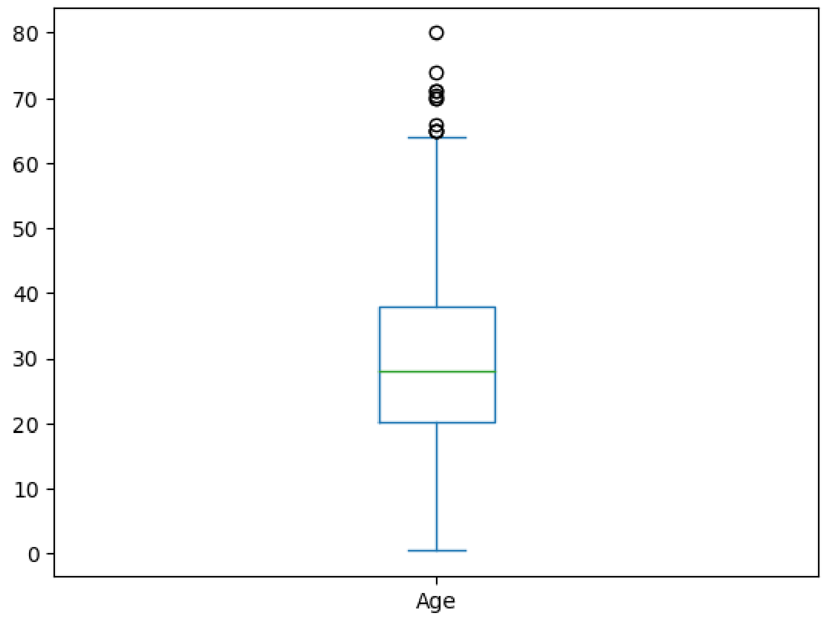
In [41]:
```
df['Age'].fillna(df['Age'].mean())
```

Out[41]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

univariate outlier detectioin

```
In [42]: df['Age'].plot.box()
```

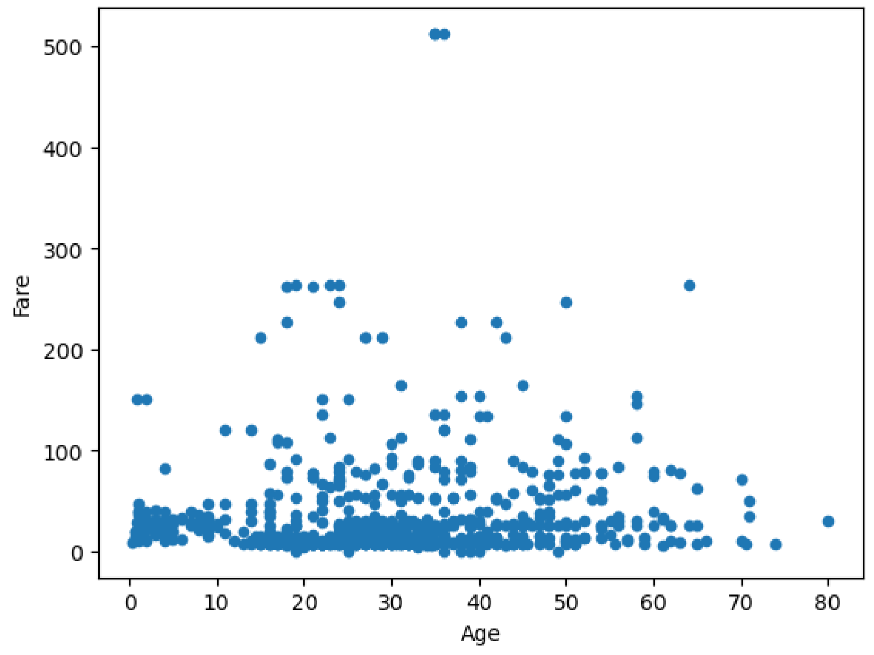
```
Out[42]: <Axes: >
```



Bivariate Outlier Detection

```
In [43]: df.plot.scatter('Age','Fare')
```

```
Out[43]: <Axes: xlabel='Age', ylabel='Fare'>
```

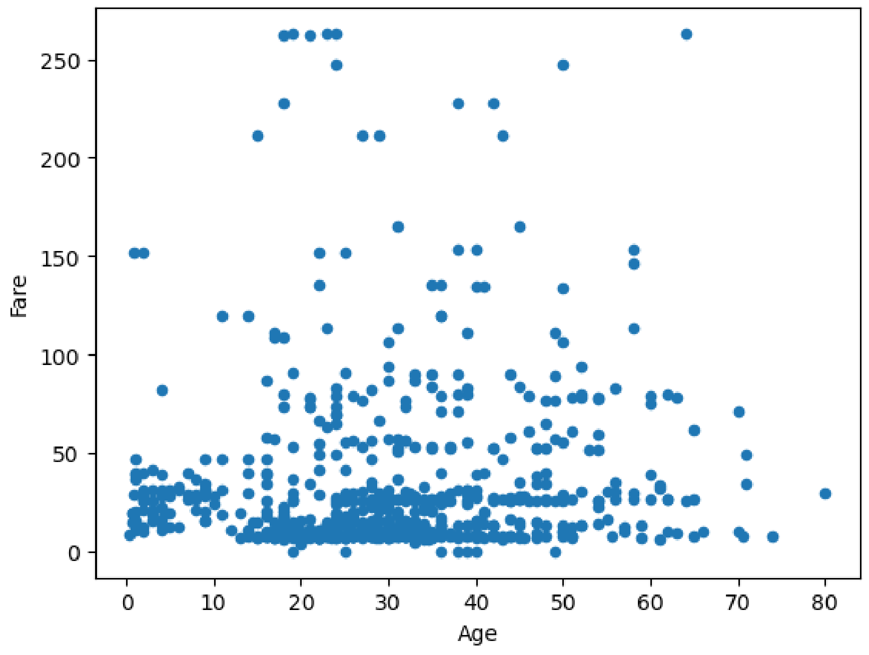


Removing outliers from the dataset

```
In [44]: df=df[df['Fare']<300]
```

```
In [45]: df.plot.scatter('Age','Fare')
```

```
Out[45]: <Axes: xlabel='Age', ylabel='Fare'>
```

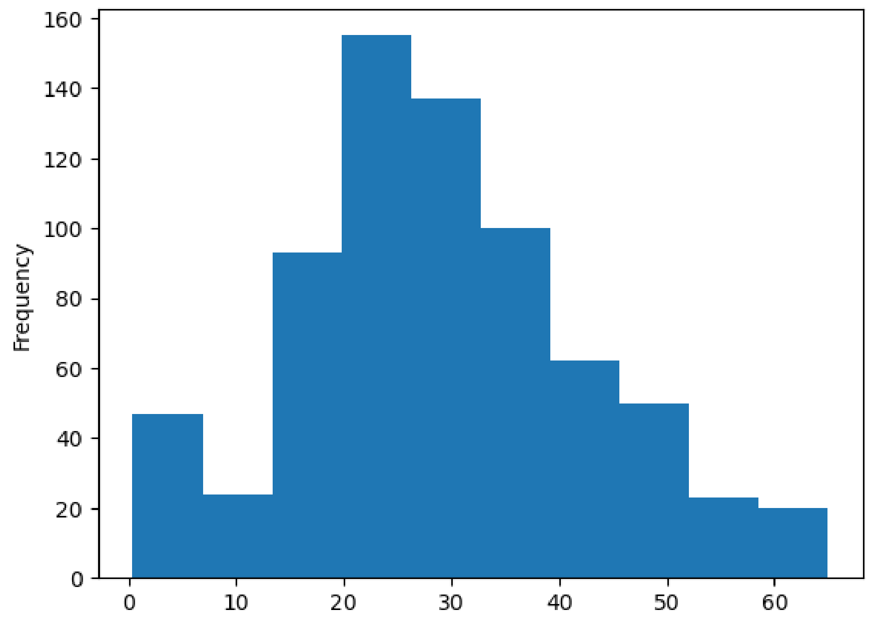


Replacing outliers in age with the mean age value

In [46]:
```
df.loc[df['Age']>65,'Age']=np.mean(df['Age'])
```

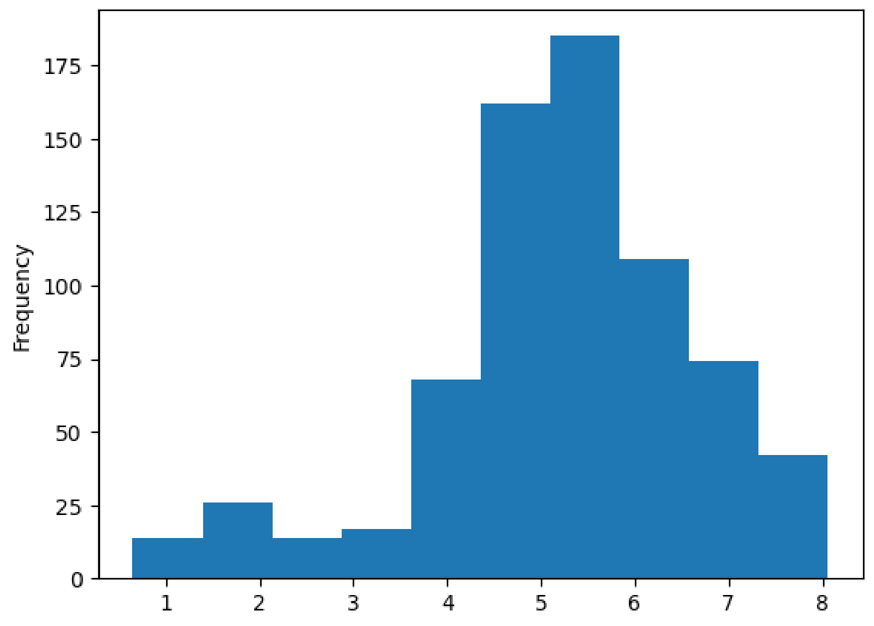
In [47]:
```
df['Age'].plot.hist()
```

Out[47]:
```
<Axes: ylabel='Frequency'>
```



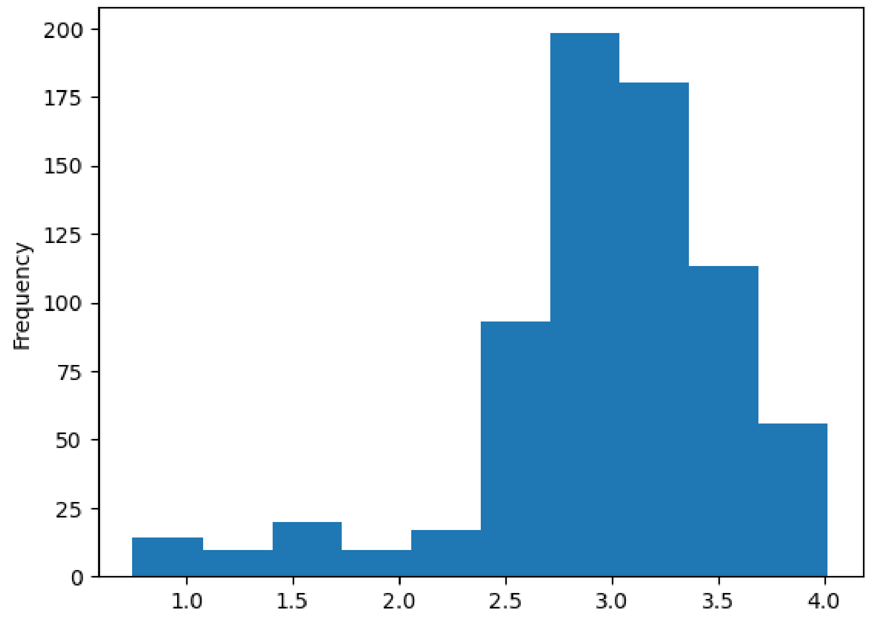
```
In [48]: np.power(df['Age'],1/2).plot.hist()
```

```
Out[48]: <Axes: ylabel='Frequency'>
```



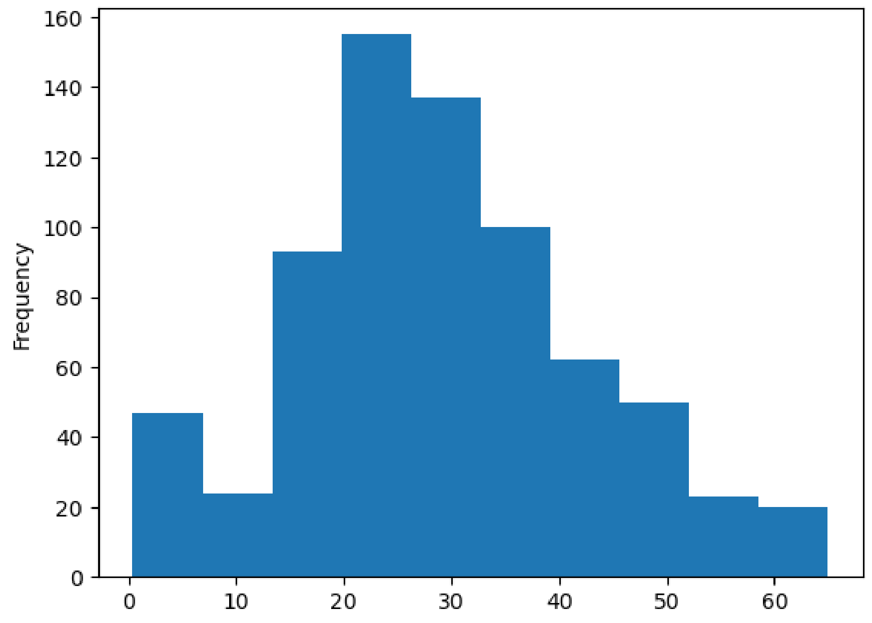
```
In [49]: np.power(df['Age'],1/3).plot.hist()
```

```
Out[49]: <Axes: ylabel='Frequency'>
```



```
In [50]: df['Age'].plot.hist()
```

```
Out[50]: <Axes: ylabel='Frequency'>
```



In [51]:
```
df.head()
```
Out[51]:

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

In [52]:
```
df.isnull().sum()
```
Out[52]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          686
Embarked         2
dtype: int64
```
In [ ]:
```
df.drop('Cabin',axis = 1,inplace= True)
```
In [55]:
```
df['Age'].fillna(df['Age'].mean(), inplace = True)
```
In [56]:
```
df
```

Out[56]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.00000	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00000	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.00000	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00000	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	male	35.00000	0	0	373450	8.0500	S
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.00000	0	0	211536	13.0000	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.00000	0	0	112053	30.0000	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.20404	1	2	W./C. 6607	23.4500	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.00000	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.00000	0	0	370376	7.7500	Q

888 rows × 11 columns

In [57]:
```
print(df['Embarked'].mode())
```
```
0    S
Name: Embarked, dtype: object
```

In [58]:
```
print(df['Embarked'].mode()[0])
```
```
S
```

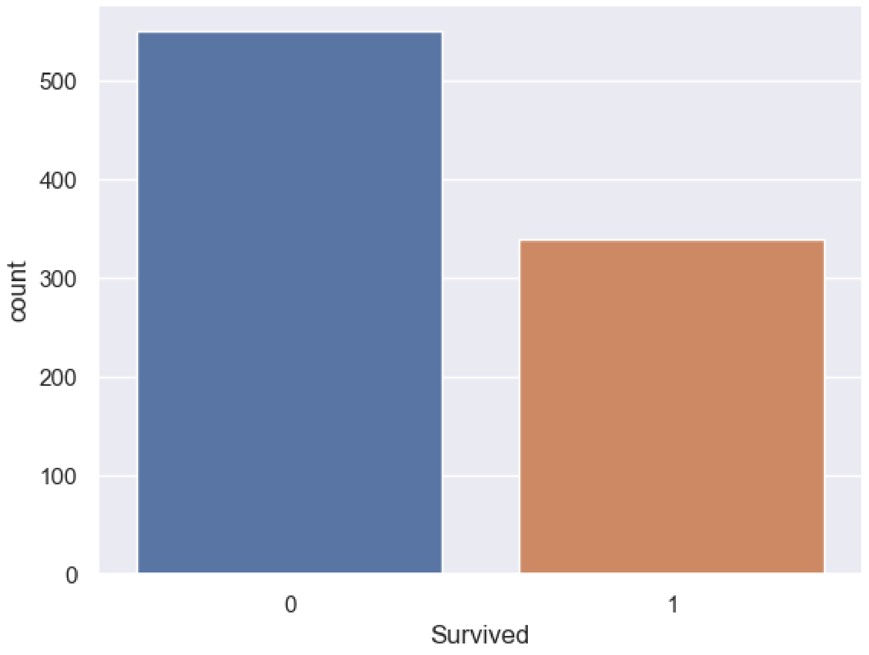
In [59]:
```
df['Embarked'].fillna(df['Embarked'].mode()[0],inplace = True)
```

In [60]:
```
df.isnull().sum()
```
Out[60]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

In [61]:
```
df.describe()
```

Out[61]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000
mean	445.618243	0.381757	2.313063	29.204040	0.524775	0.381757	30.582164
std	257.405474	0.486091	0.834007	12.384821	1.104186	0.806949	41.176366
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	222.750000	0.000000	2.000000	22.000000	0.000000	0.000000	7.895800
50%	445.500000	0.000000	3.000000	29.204040	0.000000	0.000000	14.454200
75%	667.250000	1.000000	3.000000	35.000000	1.000000	0.000000	30.771850
max	891.000000	1.000000	3.000000	65.000000	8.000000	6.000000	263.000000

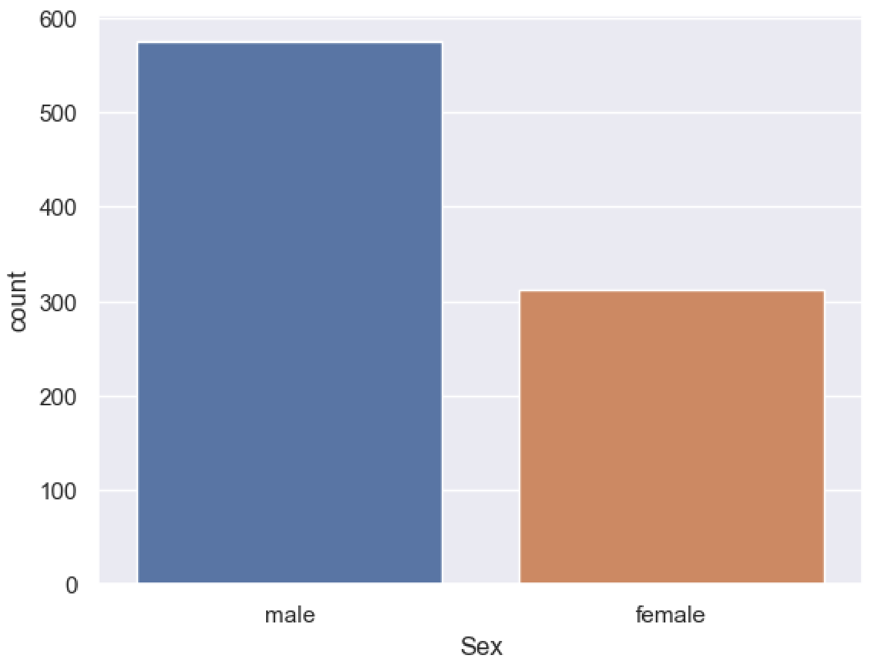
In [62]:
```
df['Survived'].value_counts()
```
Out[62]:
```
0    549
1    339
Name: Survived, dtype: int64
```
In [63]:
```
import seaborn as sns
```
In [64]:
```
sns.set()
```
In [73]:
```
sns.countplot(data = df,x=df['Survived'])
```
Out[73]:
```
<Axes: xlabel='Survived', ylabel='count'>
```


```
In [74]: df['Sex'].value_counts()
```

```
Out[74]: male      575
         female    313
         Name: Sex, dtype: int64
```

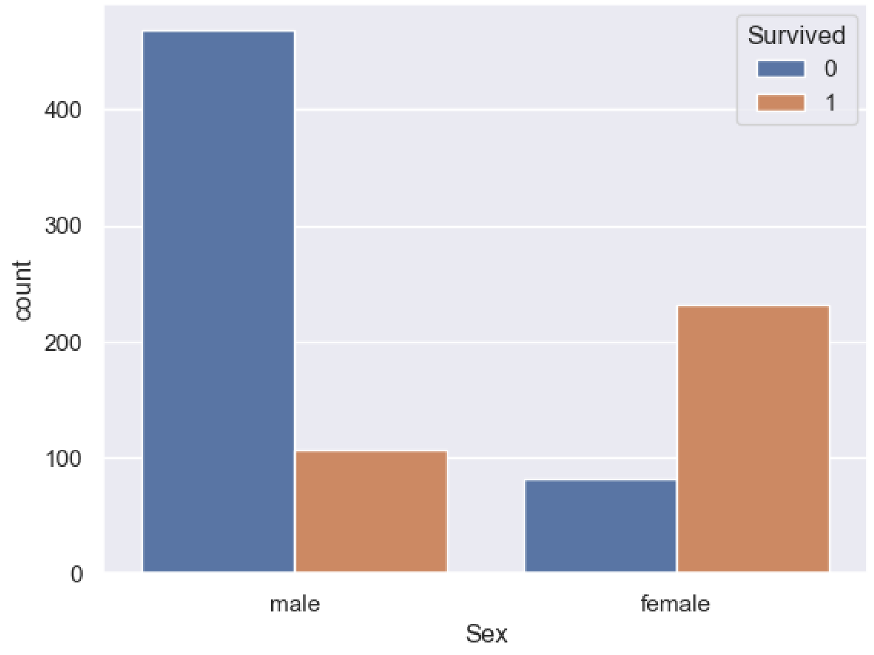
```
In [75]: sns.countplot(data = df, x= df['Sex'])
```

```
Out[75]: <Axes: xlabel='Sex', ylabel='count'>
```



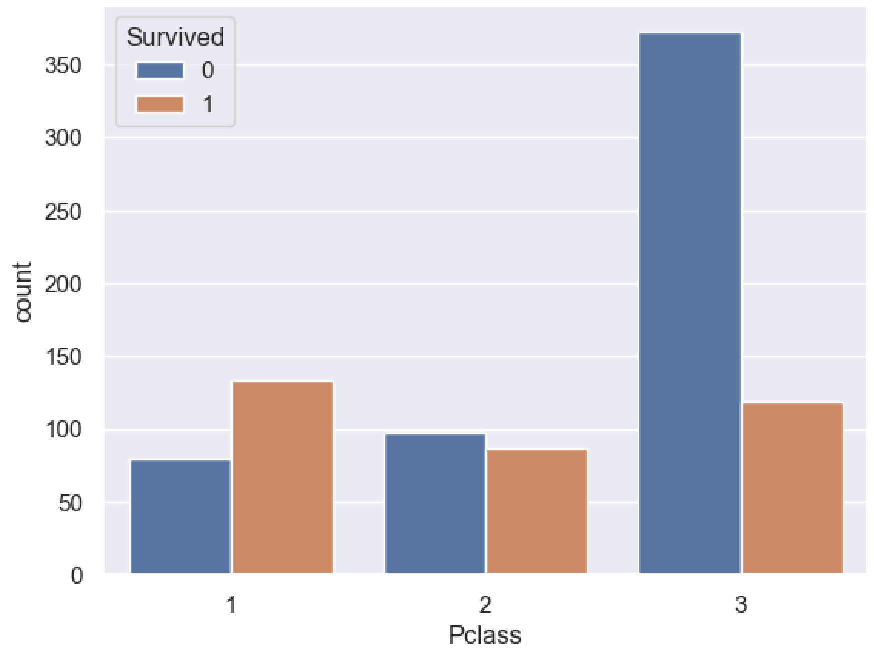
```
In [76]: sns.countplot(data = df , x = df['Sex'], hue = df['Survived'])
```

```
Out[76]: <Axes: xlabel='Sex', ylabel='count'>
```



```
In [77]: sns.countplot(data = df , x = df['Pclass'], hue = df['Survived'])
```

```
Out[77]: <Axes: xlabel='Pclass', ylabel='count'>
```



In [78]:
```python
#encoding the categorical columns
df['Embarked'].value_counts()
```

Out[78]:
```
S    646
C    165
Q     77
Name: Embarked, dtype: int64
```

In [80]:
```python
df.replace({'Sex':{'male':0,'female':1},'Embarked':{'S':0,'C':1,'Q':2}},inplace = True)
```

In [81]:
```python
df.head()
```

Out[81]:

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

In [82]:
```python
#separating Features And target
```

In [83]:
```python
x= df.drop(columns= ['Name','Ticket','PassengerId', 'Survived'],axis =1)
y= df['Survived']
```

In [84]:
```python
x
```

Out[84]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	22.00000	1	0	7.2500	0
1	1	1	38.00000	1	0	71.2833	1
2	3	1	26.00000	0	0	7.9250	0
3	1	1	35.00000	1	0	53.1000	0
4	3	0	35.00000	0	0	8.0500	0
...	...	...	...	...	...	...	...
886	2	0	27.00000	0	0	13.0000	0
887	1	1	19.00000	0	0	30.0000	0
888	3	1	29.20404	1	2	23.4500	0
889	1	0	26.00000	0	0	30.0000	1
890	3	0	32.00000	0	0	7.7500	2

888 rows × 7 columns

In [85]:
```python
y
```

```
Out[85]: 0      0
         1      1
         2      1
         3      1
         4      0
               ..
         886    0
         887    1
         888    0
         889    1
         890    0
         Name: Survived, Length: 888, dtype: int64
```

```
In [86]: from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import accuracy_score
```

```
In [87]: train_x,test_x,train_y,test_y = train_test_split(x,y, test_size=0.2,random_state= 2)
```

Model Training using Logistic Regression

```
In [90]: log = LogisticRegression()
```

```
In [91]: log.fit(train_x,train_y)
```

```
Out[91]: ▾ LogisticRegression
         LogisticRegression()
```

```
In [92]: train_x_pre = log.predict(train_x)
```

```
In [93]: train_x_pre
```

```
Out[93]: array([0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
                0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1,
                1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,
                1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
                0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0,
                0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0,
                0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1,
                0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
                1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0,
                0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
                0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0,
                0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
                0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
                1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1,
                0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1,
                1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
                1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0,
                0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
                0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
                1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0,
                1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
                1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0,
                0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1,
                0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1,
                0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,
                0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
                1, 1, 1, 0, 0, 1], dtype=int64)
```

```
In [94]: accu = accuracy_score(train_y,train_x_pre)
```

```
In [95]: accu
```

```
Out[95]: 0.8084507042253521
```

```
In [96]: print('Accuracy of test data :',accu)
```

```
Accuracy of test data : 0.8084507042253521
```

```
In [ ]:
```