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

In [2]:
```
dataset=pd.read_csv(r"C:\Users\HP\Downloads\tested.csv")
dataset
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.829
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.000
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.687
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.662
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.287
...	...	...	...	...	...	...	...	...	...	
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.050
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.900
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.250
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.050
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.358

418 rows × 12 columns

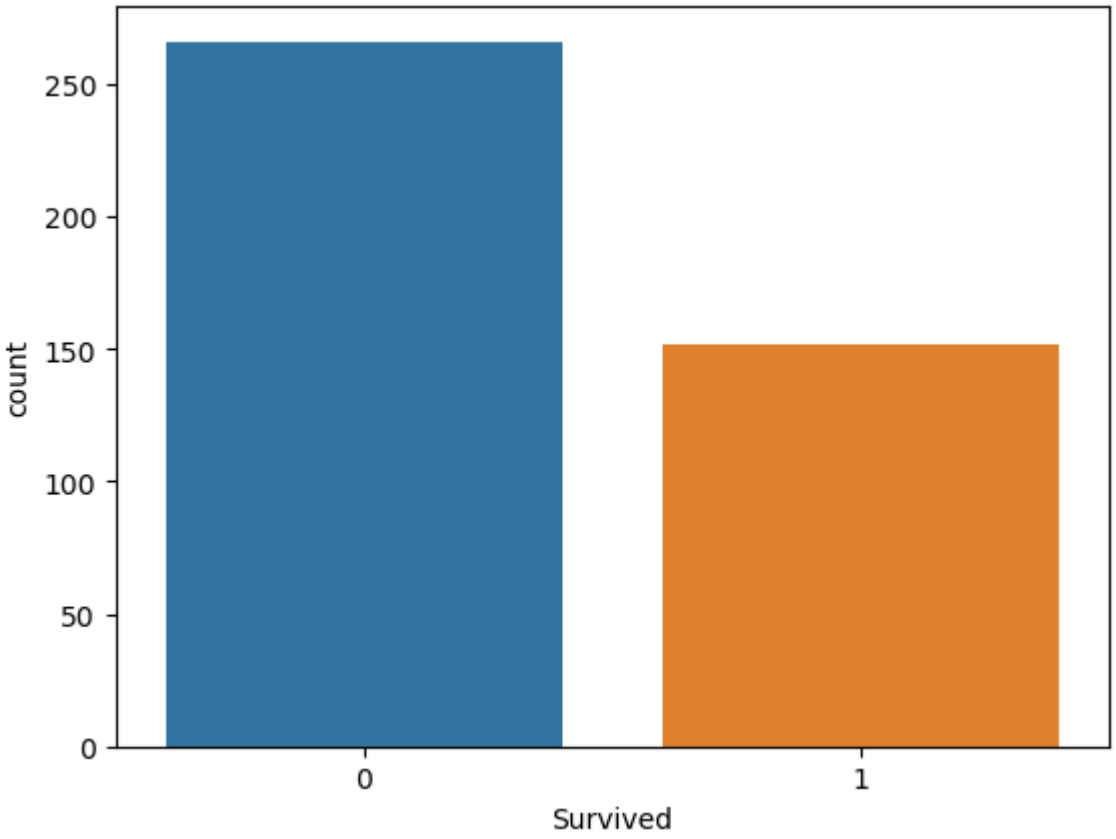
In [3]:
```
dataset.shape
```

Out[3]:
```
(418, 12)
```

Analysing data

In [4]:
```
sns.countplot(x="Survived",data=dataset)
```

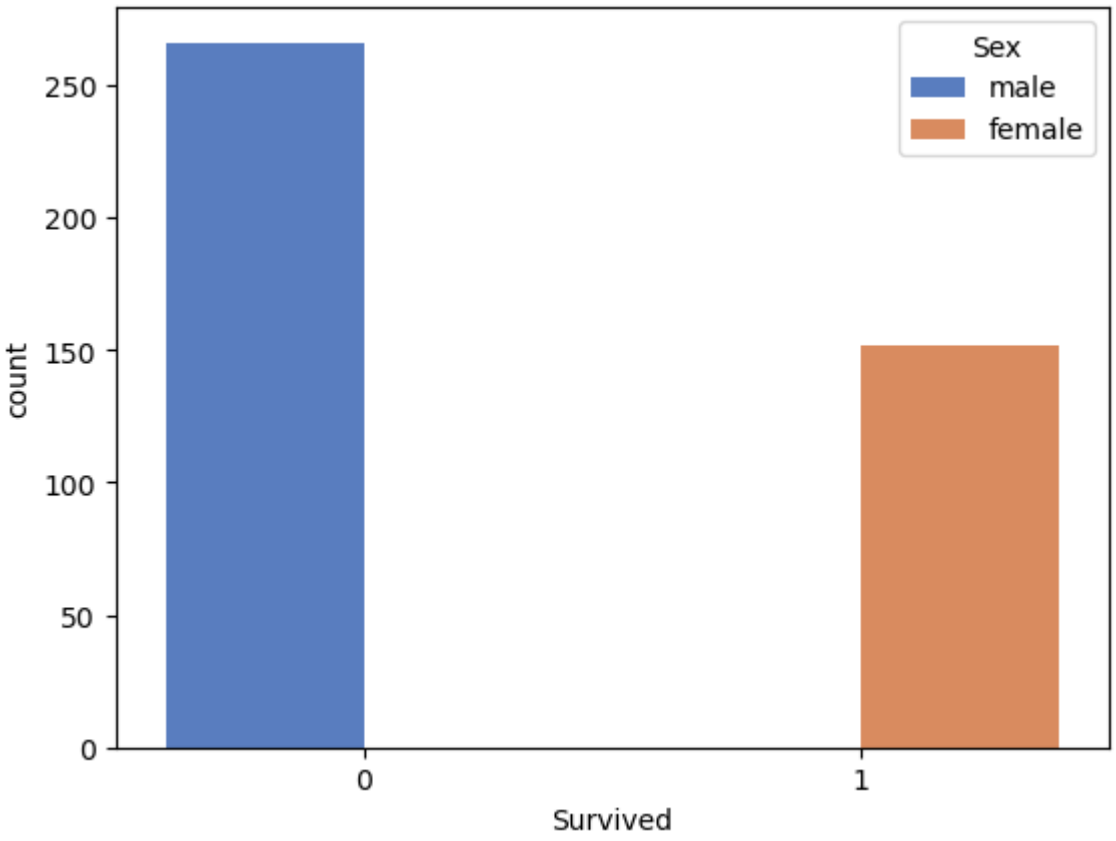
Out[4]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



Those who did not Survived (more than 500) are greater than those who survived (nearly 300)

In [5]:
```
sns.countplot(x="Survived",hue="Sex",data=dataset,palette="muted")
```

Out[5]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

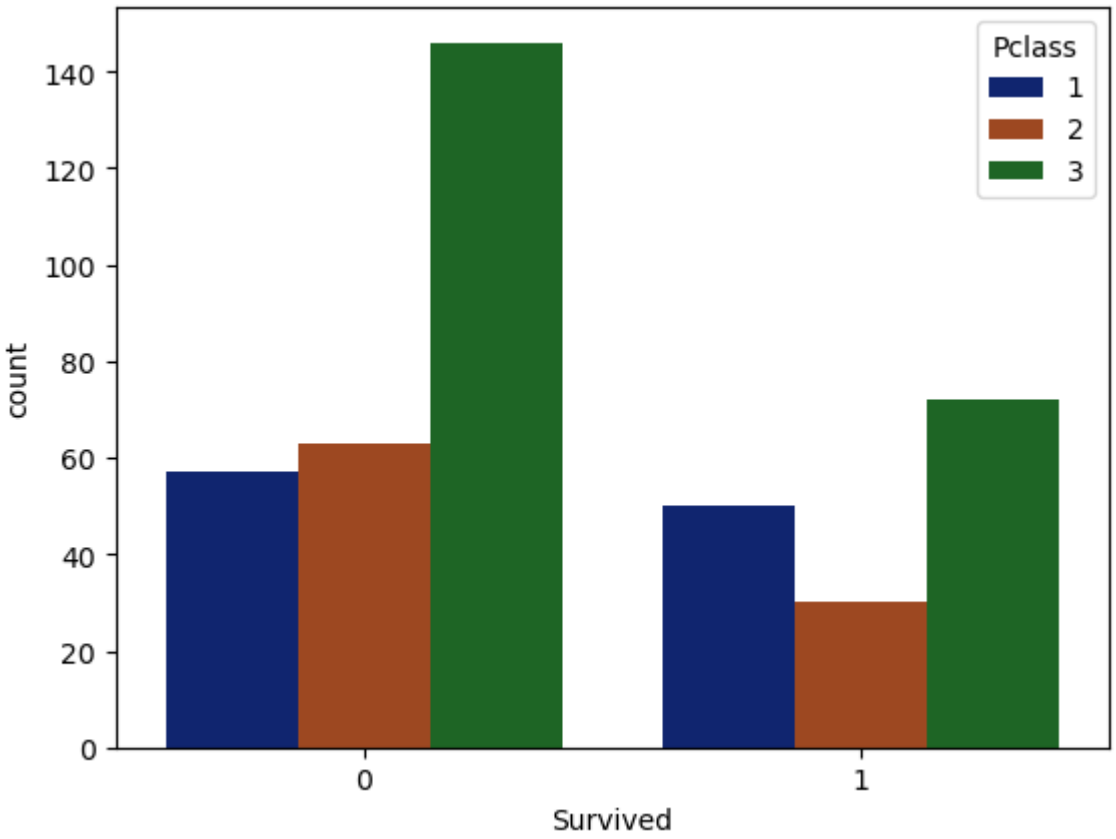


Analysis : o represent not survived and 1 is for survived

males are trice more likely to survive than females

In [6]:
```
sns.countplot(x="Survived",hue="Pclass",data=dataset,palette="dark")
```

Out[6]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```

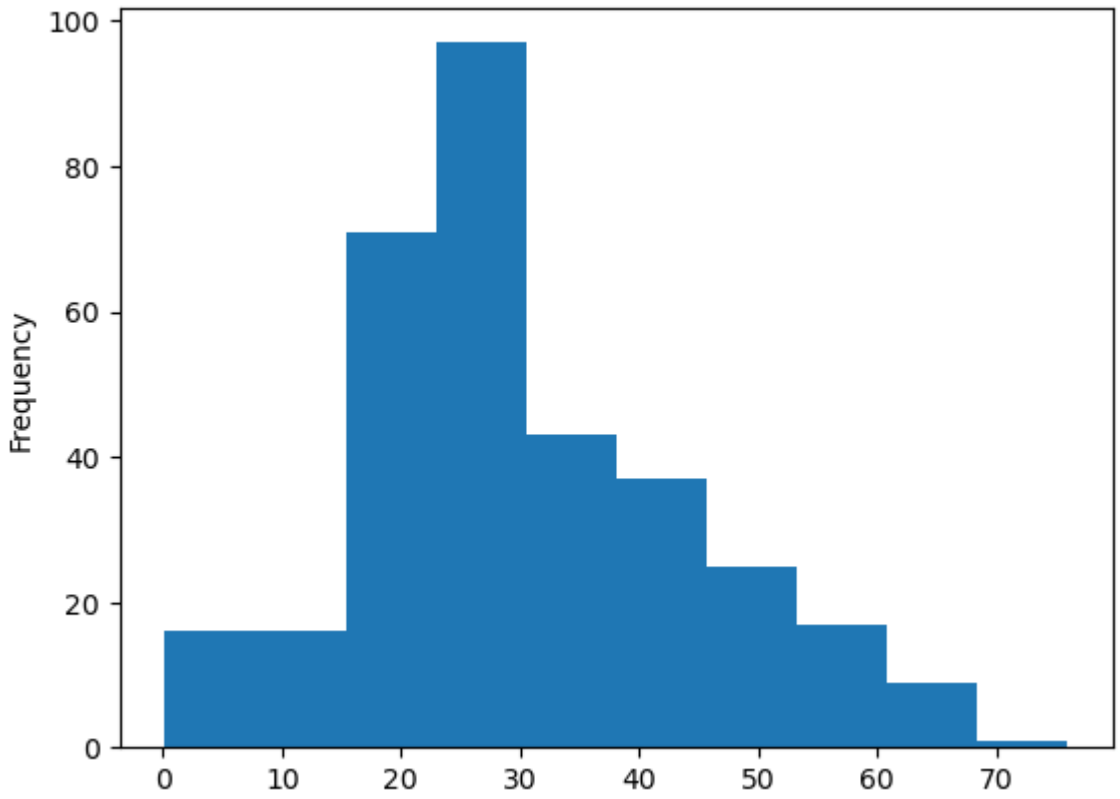


Analysis = the passenger who did not sarvived belong to the 3rd class

1st class passamger are more likely to survive

In [7]:
```
dataset["Age"].plot.hist()
```

Out[7]:
```
<AxesSubplot:ylabel='Frequency'>
```

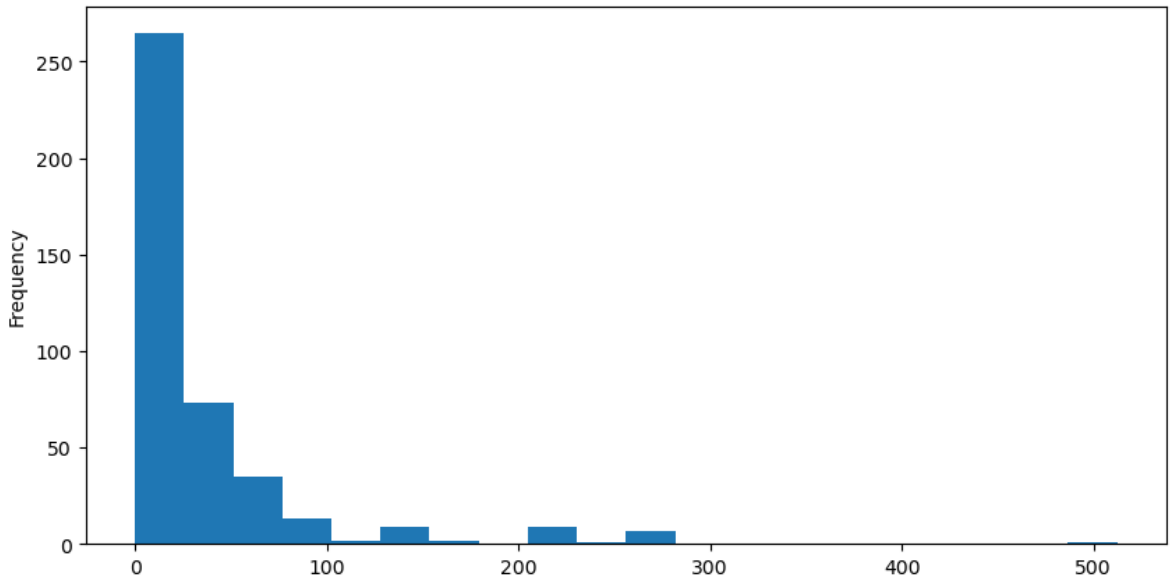


we notice that highest age group travelling are among the young age between 20-30

very few passenger in age group 60-70

In [8]:
```
dataset["Fare"].plot.hist(bins=20,figsize=(10,5))
```

Out[8]:
```
<AxesSubplot:ylabel='Frequency'>
```

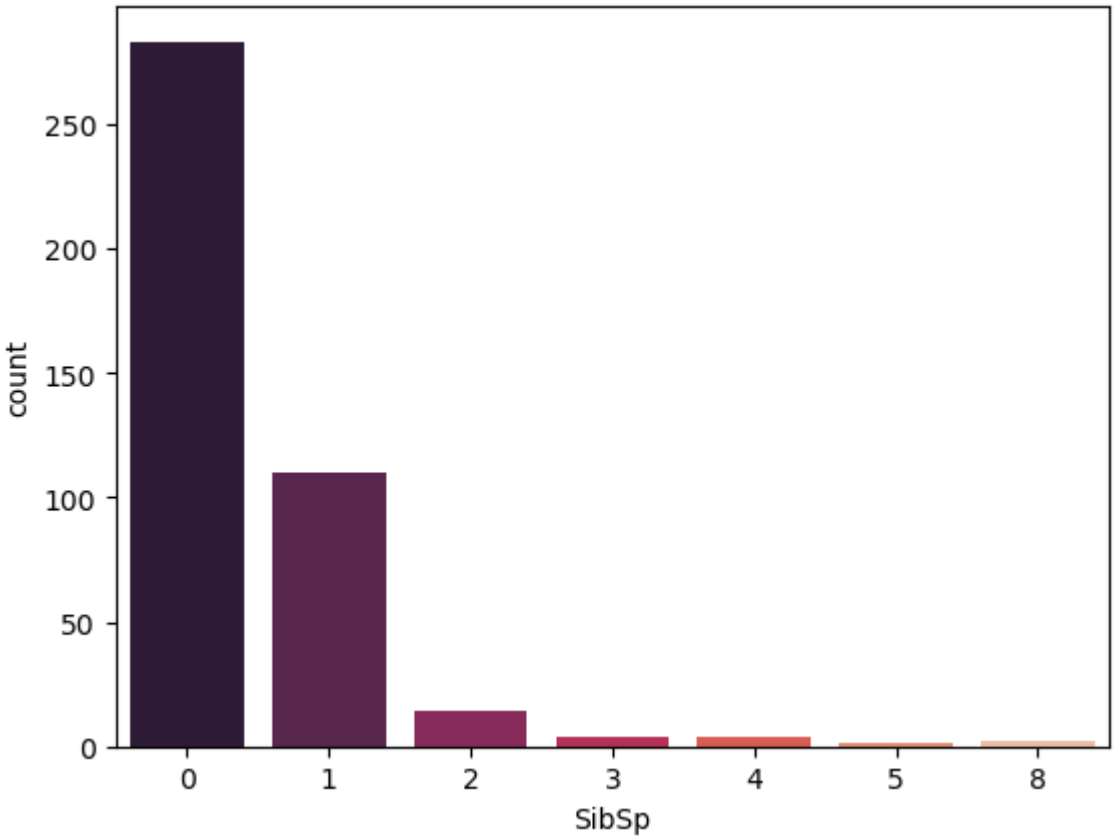


we observe that most of the tickets bought are under fare 100

and very for are on the higher side of fare i.e. 100-300

In [9]:
```
sns.countplot(x="SibSp",data=dataset,palette="rocket")
```

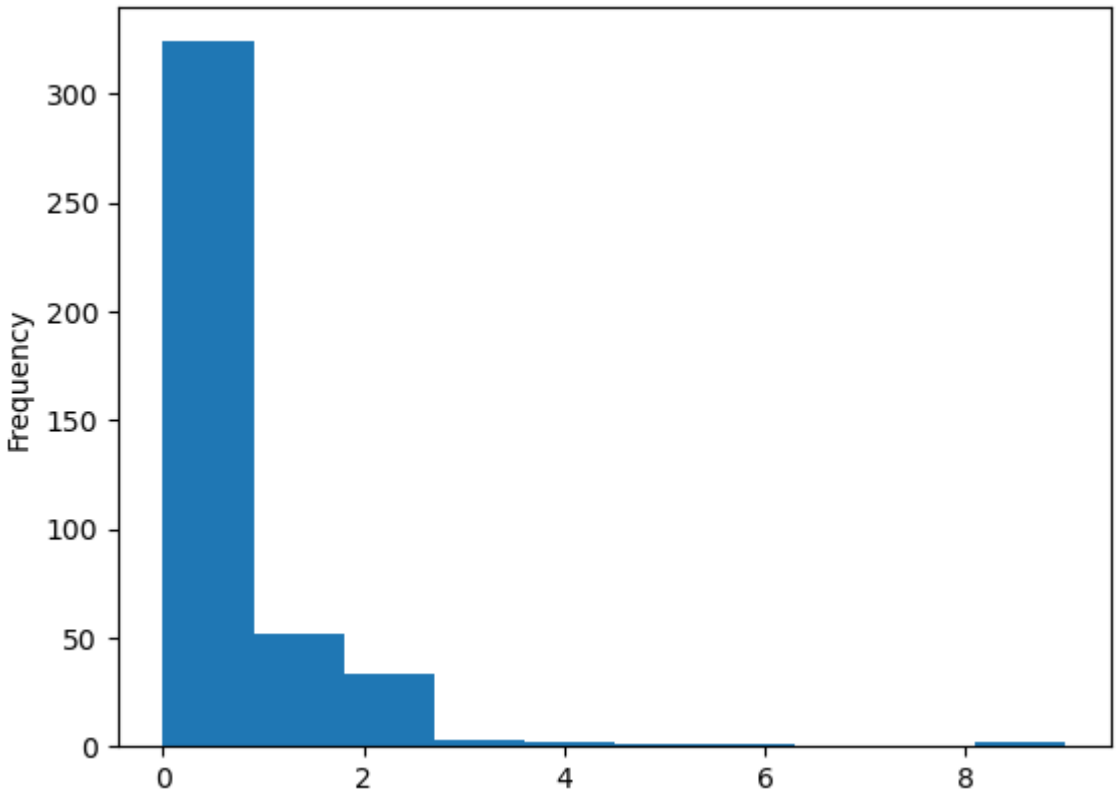
Out[9]:
```
<AxesSubplot:xlabel='SibSp', ylabel='count'>
```



We notice that most of the passanger do not have their sibling aboard

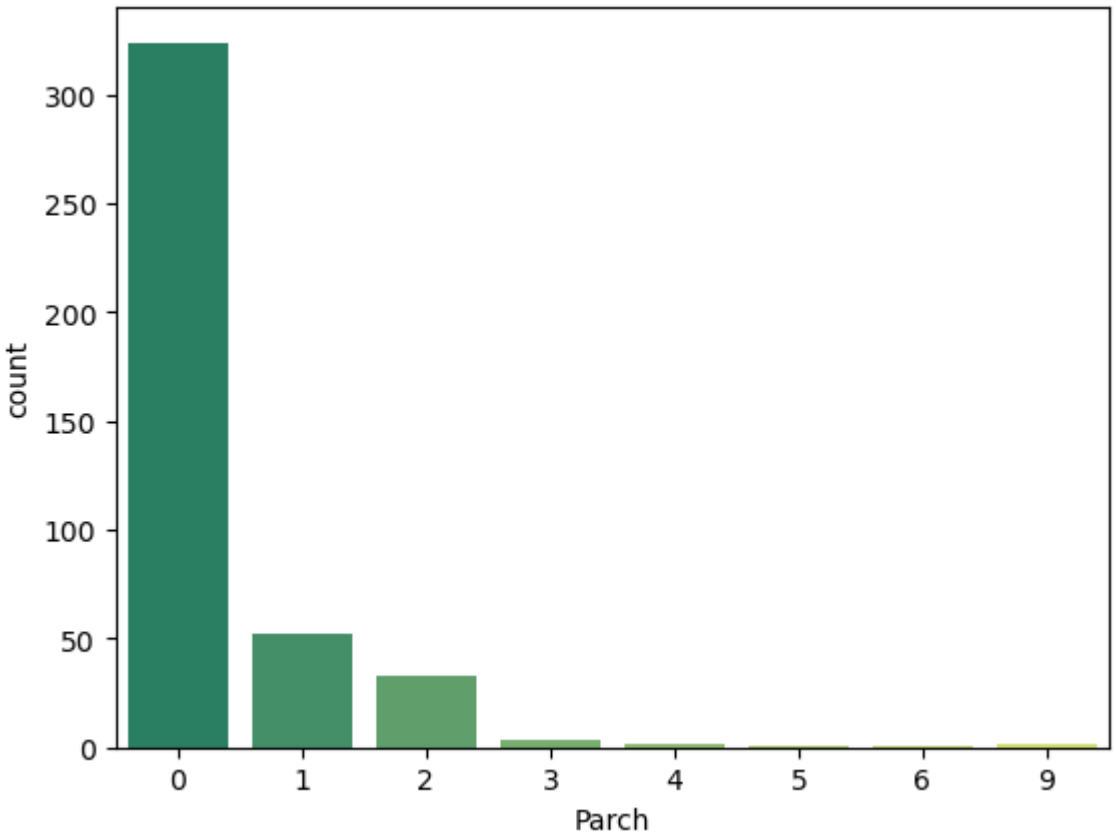
In [10]:
```
dataset["Parch"].plot.hist()
```

Out[10]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [11]:
```
sns.countplot(x="Parch",data=dataset,palette="summer")
```

Out[11]:
```
<AxesSubplot:xlabel='Parch', ylabel='count'>
```



the number of parent and siblingd who aboard the ship are less

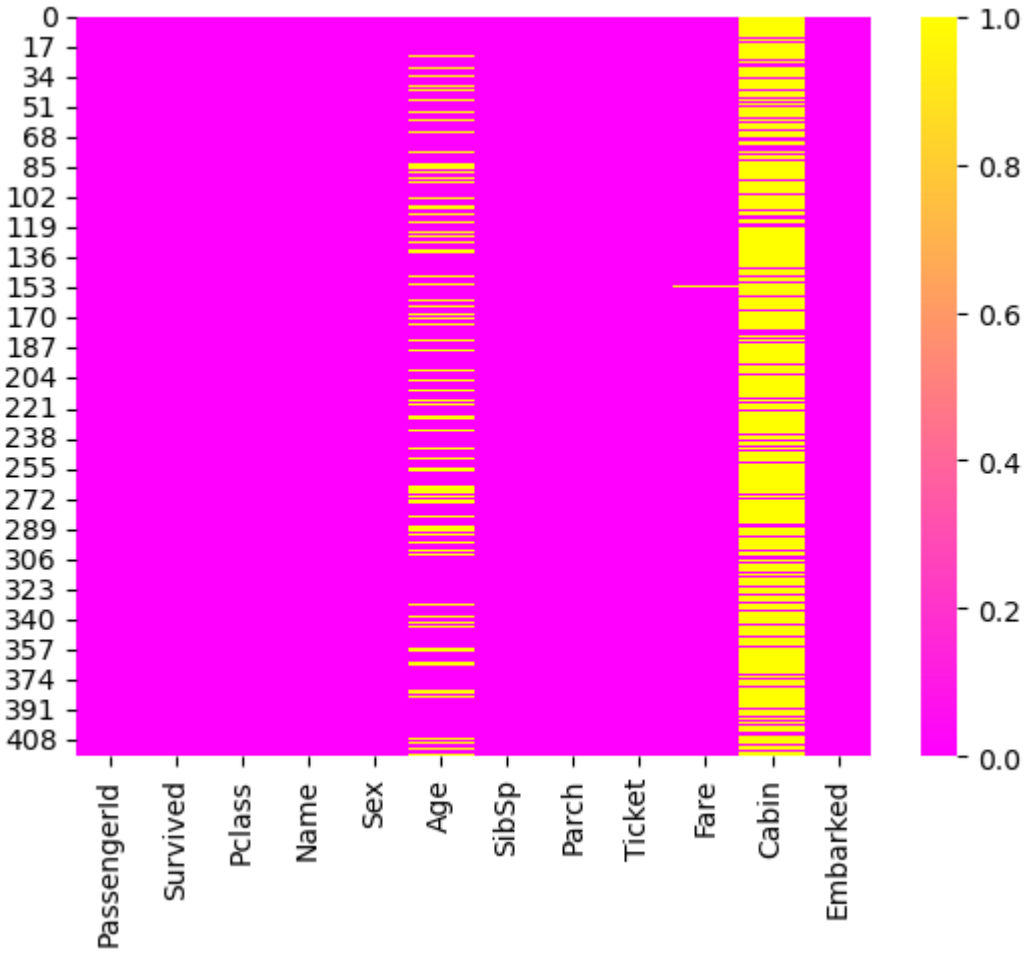
Data Wrangling

In [12]:
```
dataset.isnull().sum()
```

Out[12]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

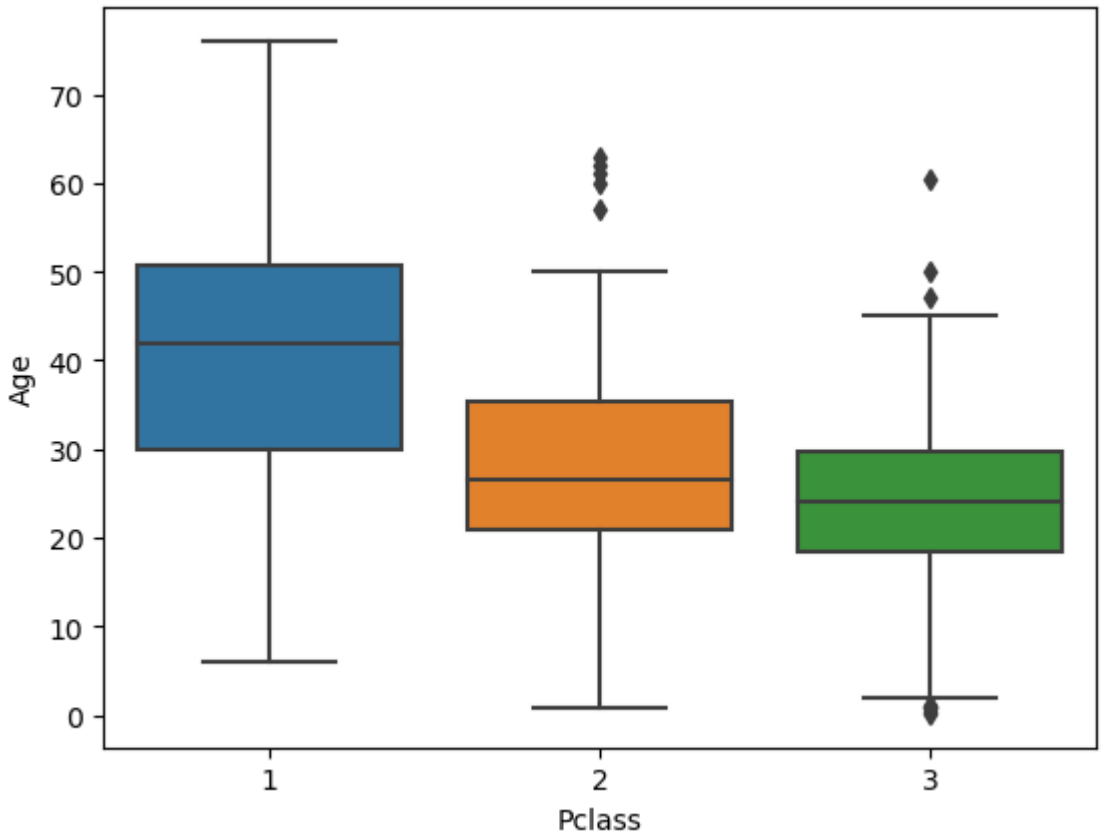
In [13]:
```
sns.heatmap(dataset.isnull(),cmap="spring")
```

Out[13]: <AxesSubplot:>



In [14]:
```
sns.boxplot(x="Pclass",y="Age",data=dataset)
```

Out[14]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```



In [15]:
```
dataset.head()
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Na
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	Na
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Na
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	Na
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	Na

In [16]:
```
dataset.drop('Cabin',axis=1,inplace=True)
```

In [17]:
```
dataset.head()
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emb
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	

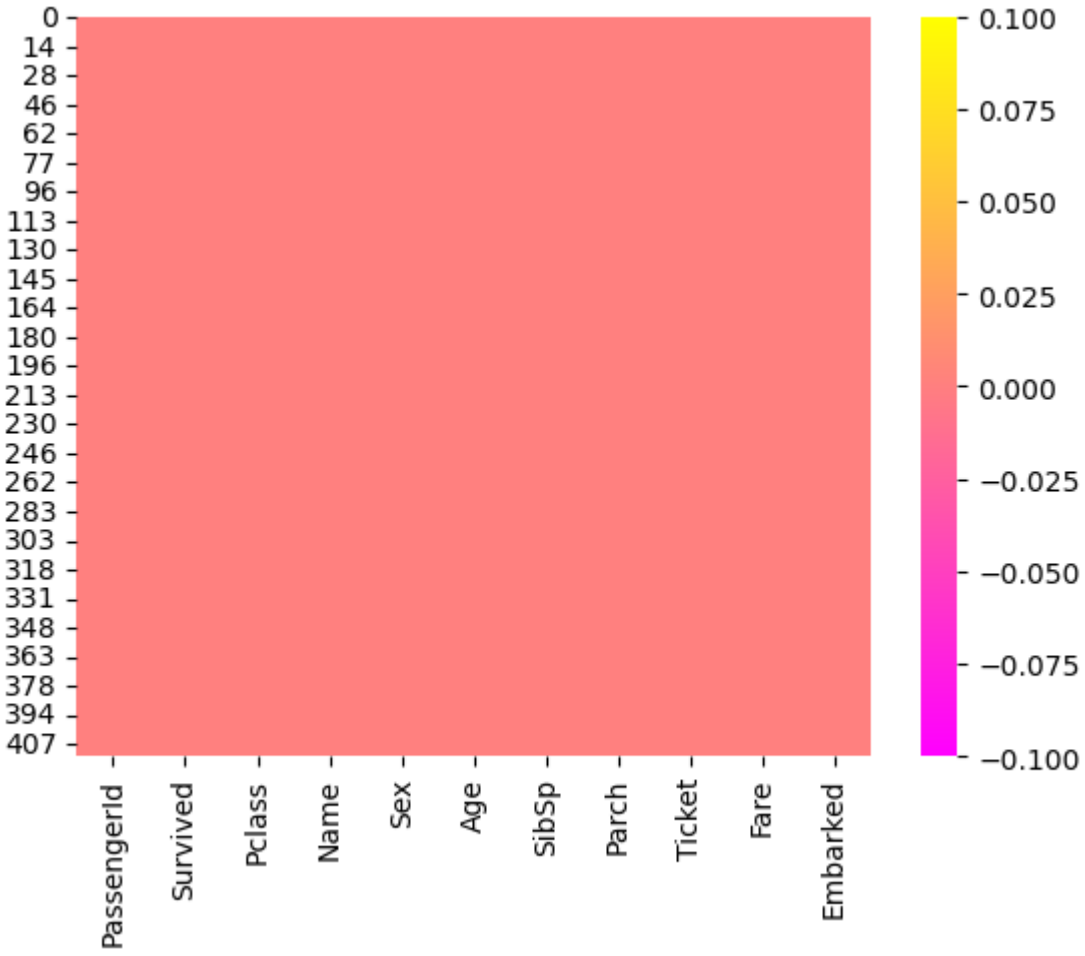
In [18]:
```
dataset.dropna(inplace=True)
```

In [19]:
```
dataset.isnull().sum()
```

Out[19]:
```
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 [20]:
```
sns.heatmap(dataset.isnull(),cmap="spring")
```

Out[20]: <AxesSubplot:>



In [21]:
```
dataset.isnull().sum()
```

Out[21]:
```
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 [22]:
```
dataset.head()
```

Out[22]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Emb
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	

In [23]:
```
from sklearn.preprocessing import LabelEncoder
l1=LabelEncoder()
```

In [24]:
```
dataset["Sex"]=l1.fit_transform(dataset["Sex"])
dataset["Embarked"]=l1.fit_transform(dataset["Embarked"])
```

In [25]:
```
dataset
```

Out[25]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	1	34.5	0	0	330911	7.829
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0	363272	7.000
2	894	0	2	Myles, Mr. Thomas Francis	1	62.0	0	0	240276	9.687
3	895	0	3	Wirz, Mr. Albert	1	27.0	0	0	315154	8.662
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1	3101298	12.287
...	...	...	...	...	...	...	...	...	...	..
409	1301	1	3	Peacock, Miss. Treasteall	0	3.0	1	1	SOTON/O.Q. 3101315	13.775
411	1303	1	1	Minahan, Mrs. William Edward (Lillian E Thorpe)	0	37.0	1	0	19928	90.000
412	1304	1	3	Henriksson, Miss. Jenny Lovisa	0	28.0	0	0	347086	7.775
414	1306	1	1	Oliva y Ocana, Dona. Fermina	0	39.0	0	0	PC 17758	108.900
415	1307	0	3	Saether, Mr. Simon Sivertsen	1	38.5	0	0	SOTON/O.Q. 3101262	7.250

331 rows × 11 columns

In [26]:
```
dataset.drop(["PassengerId","Name","Ticket","Pclass"],axis=1,inplace=True)
```

In [27]:
```
dataset.head()
```

Out[27]:

	Survived	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	1	34.5	0	0	7.8292	1
1	1	0	47.0	1	0	7.0000	2
2	0	1	62.0	0	0	9.6875	1
3	0	1	27.0	0	0	8.6625	2
4	1	0	22.0	1	1	12.2875	2

Train data

In [28]:
```
x=dataset.drop("Survived",axis=1)
y=dataset["Survived"]
```

In [29]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
```

In [30]:
```
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
lg.fit(x_train,y_train)
```

Out[30]:
```
LogisticRegression()
```

In [31]:
```
y_pred=lg.predict(x_test)
y_pred
```

Out[31]:
```
array([0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0,
       1], dtype=int64)
```

In [32]:
```
from sklearn.metrics import accuracy_score
ac=accuracy_score(y_test,y_pred)*100
print(ac)
```

```
100.0
```

We have the accuracy of 100% which is good and the model can predict the data accurately

In [ ]:
```

```