

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
%matplotlib inline

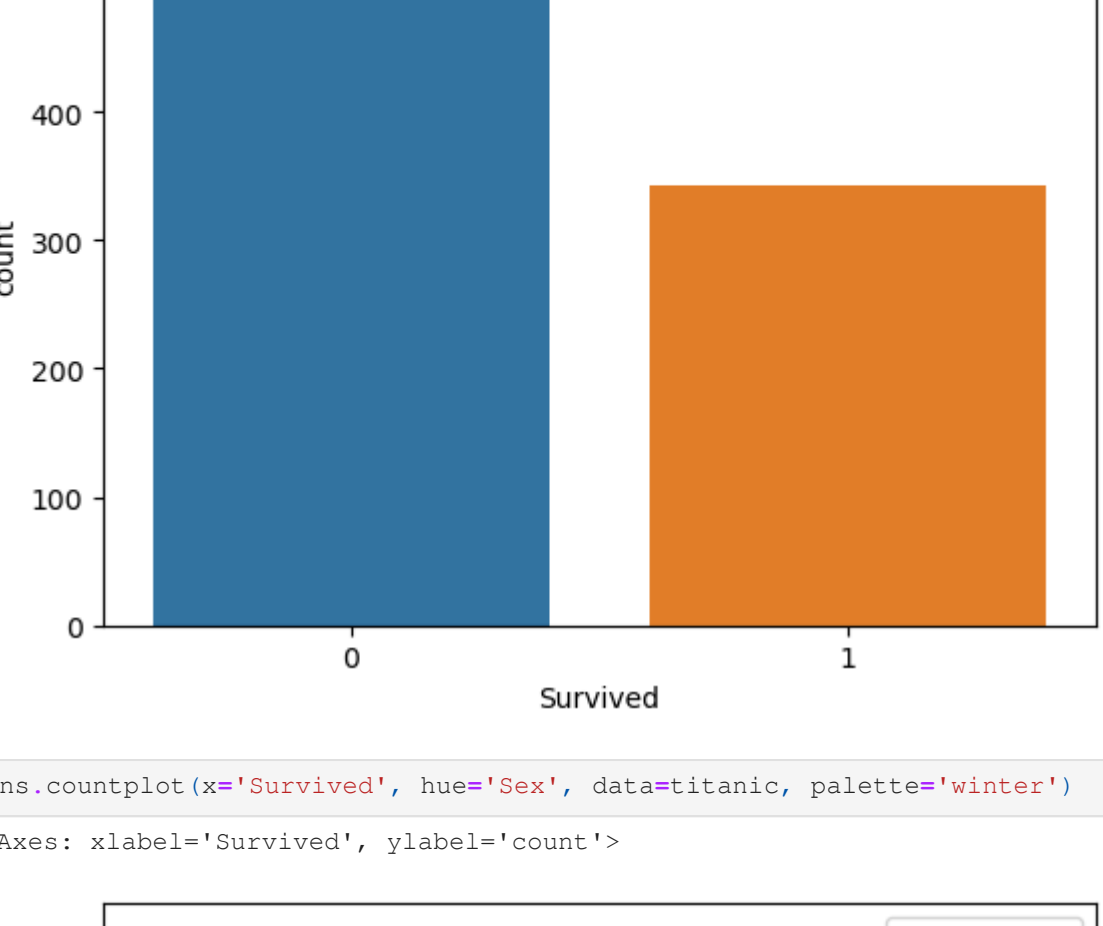
UsageError: Line magic function `%` not found.

In [3]: titanic=pd.read_excel(r"C:\Users\Tanushree\Downloads\Titanic.xlsx")
titanic.head()
```

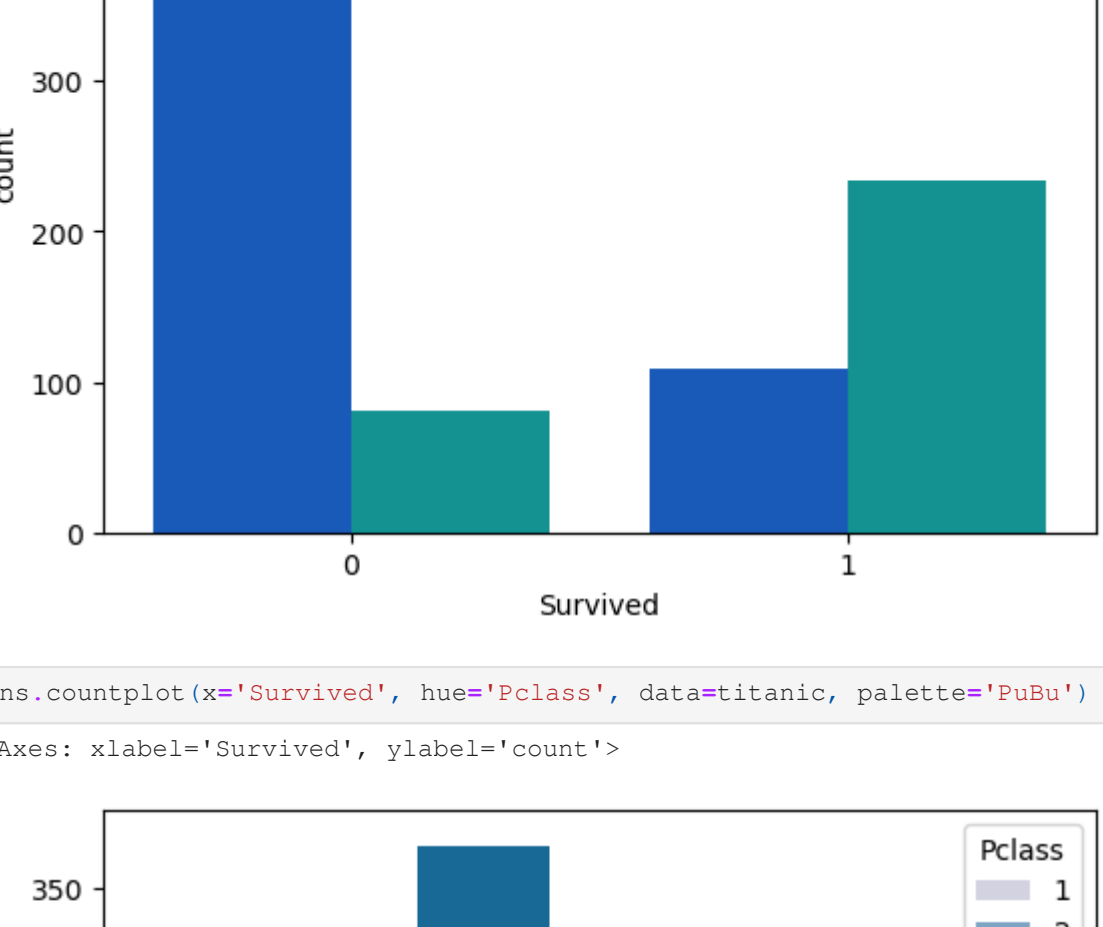
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikinen, Miss. Laina	female	26.0	0	0	STON/O2 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 [5]: titanic.shape
Out[5]: (891, 12)
```

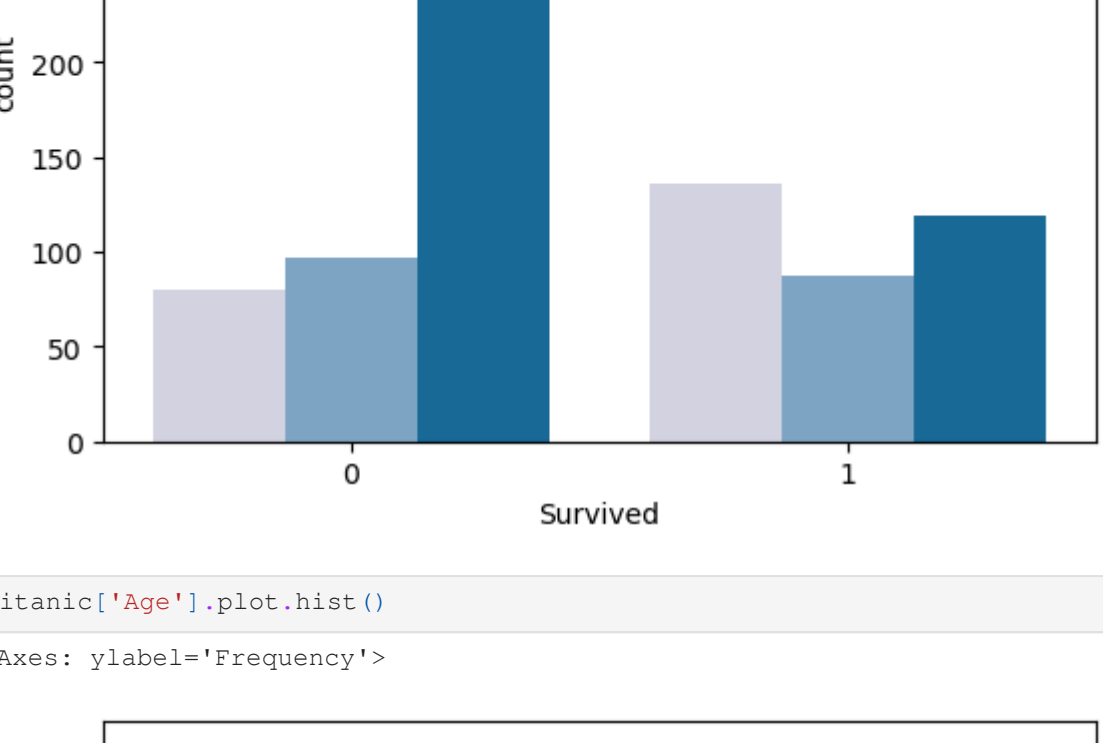
```
In [6]: sns.countplot(x='Survived', data=titanic)
Out[6]: <Axes: xlabel='Survived', ylabel='count'>
```



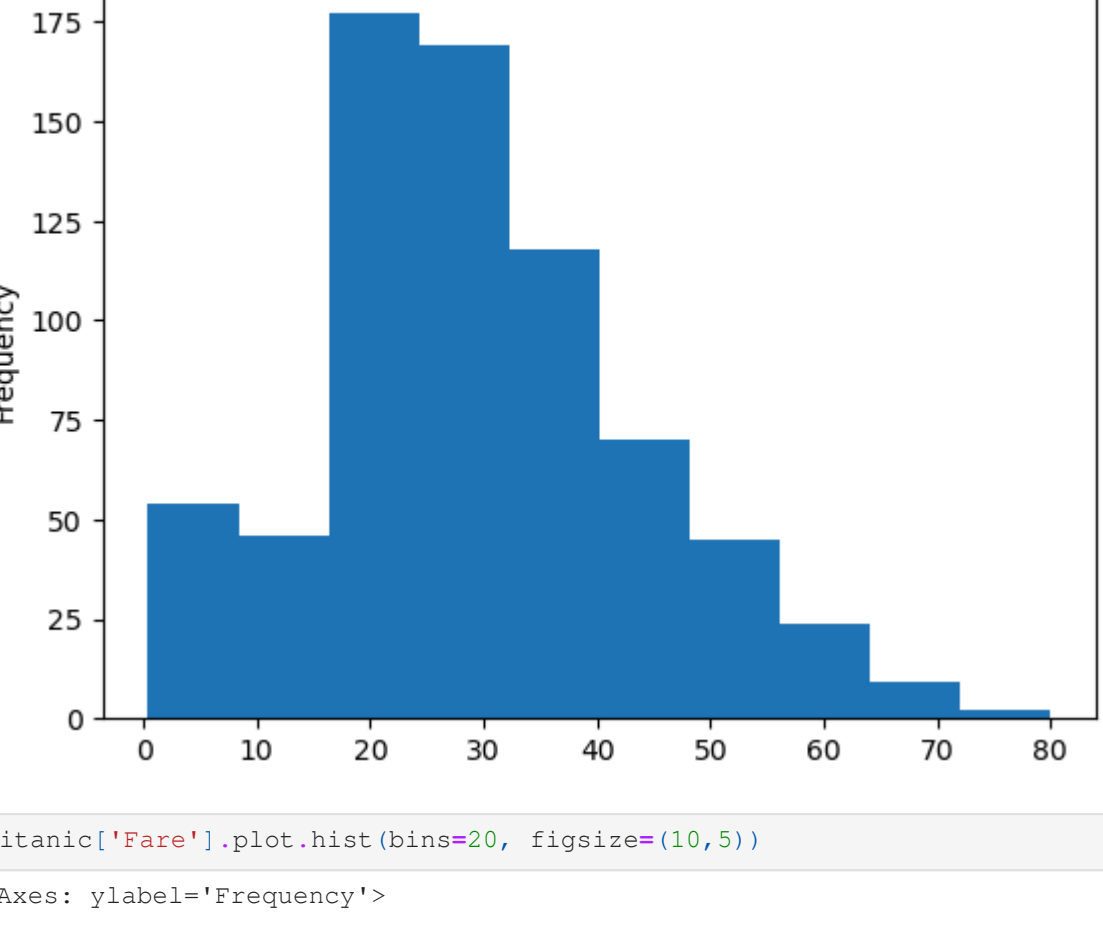
```
In [7]: sns.countplot(x='Survived', hue='Sex', data=titanic, palette='winter')
Out[7]: <Axes: xlabel='Survived', ylabel='count'>
```



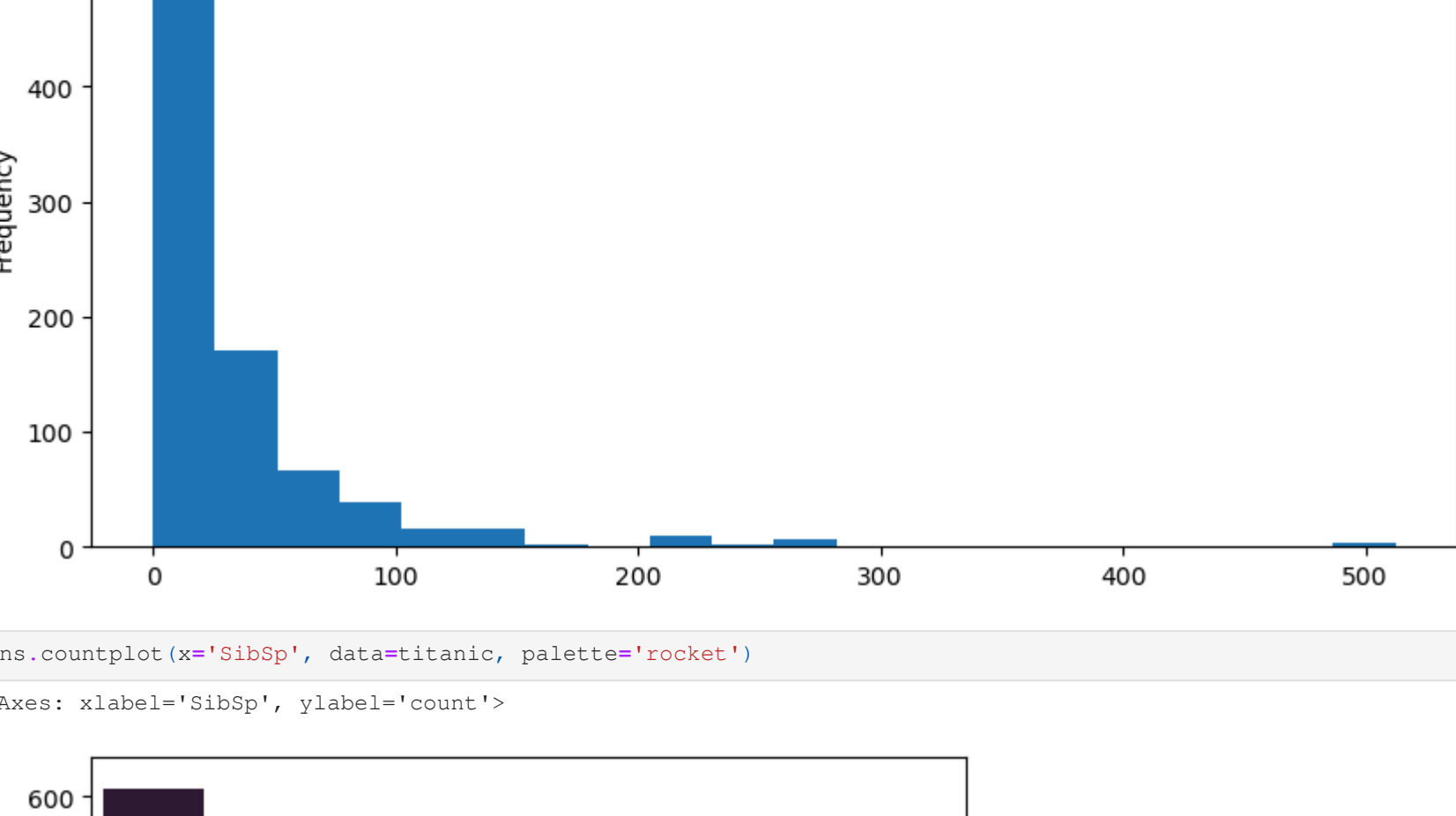
```
In [8]: sns.countplot(x='Survived', hue='Pclass', data=titanic, palette='PuBu')
Out[8]: <Axes: xlabel='Survived', ylabel='count'>
```



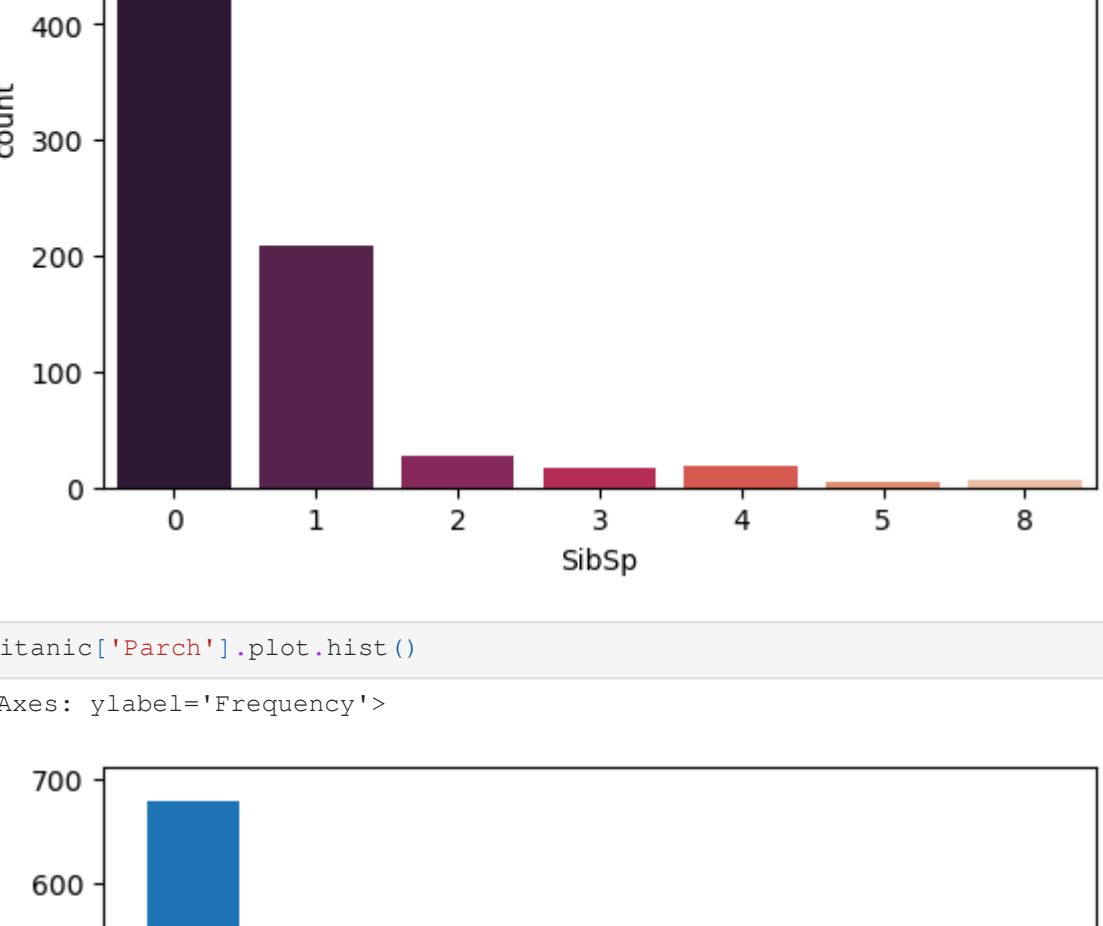
```
In [9]: titanic['Age'].plot.hist()
Out[9]: <Axes: ylabel='Frequency'>
```




```
In [11]: titanic['Fare'].plot.hist(bins=20, figsize=(10,5))
Out[11]: <Axes: ylabel='Frequency'>
```



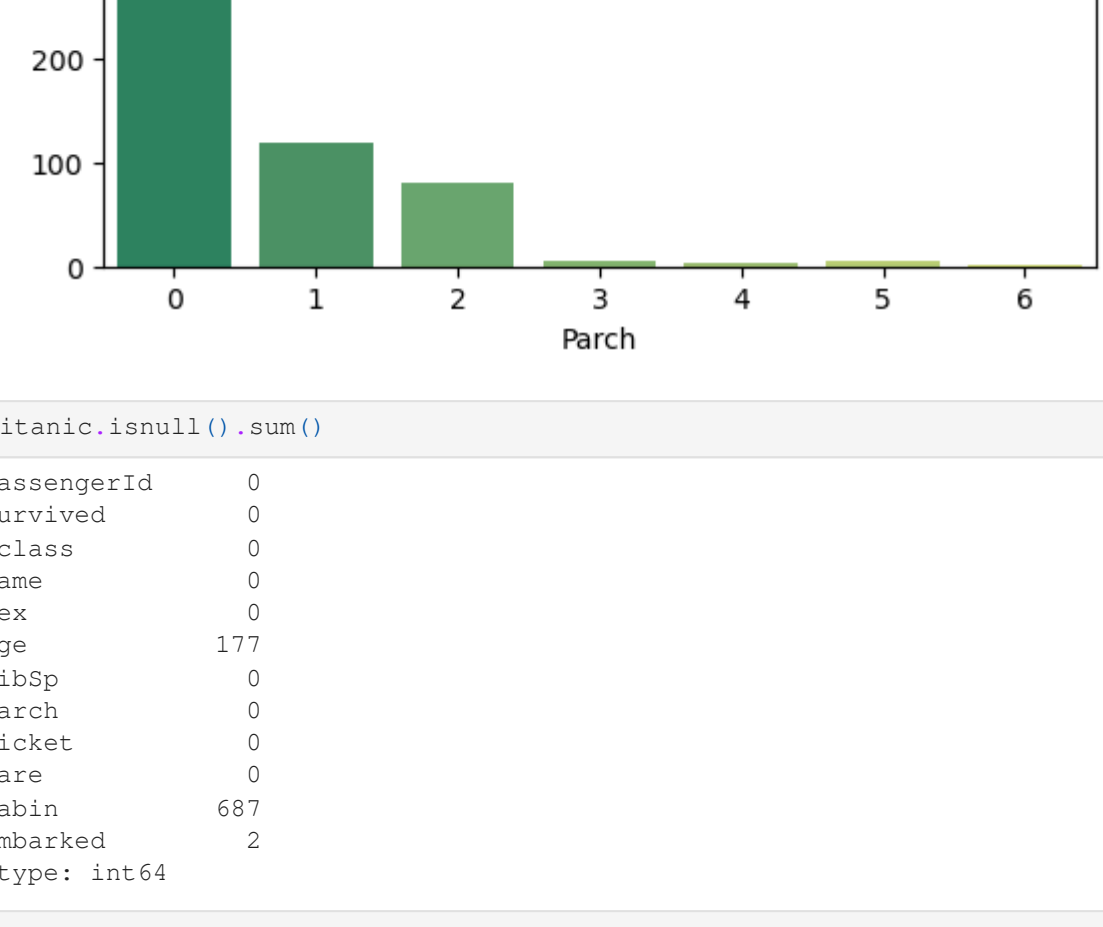
```
In [13]: sns.countplot(x='SibSp', data=titanic, palette='rocket')
Out[13]: <Axes: xlabel='SibSp', ylabel='count'>
```



```
In [14]: titanic['Parch'].plot.hist()
Out[14]: <Axes: ylabel='Frequency'>
```

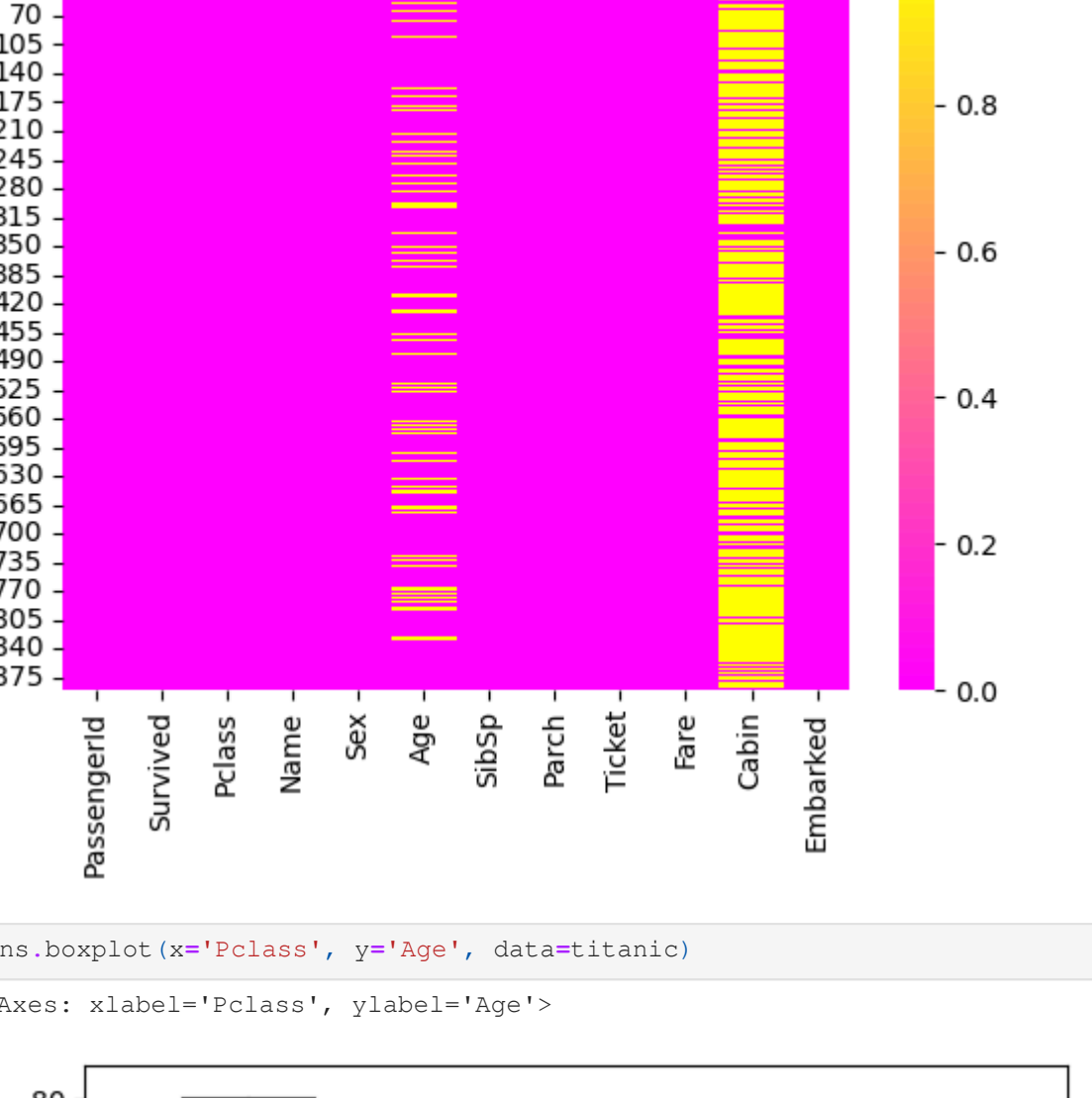


```
In [15]: sns.countplot(x='Parch', data=titanic, palette='summer')
Out[15]: <Axes: xlabel='Parch', ylabel='count'>
```

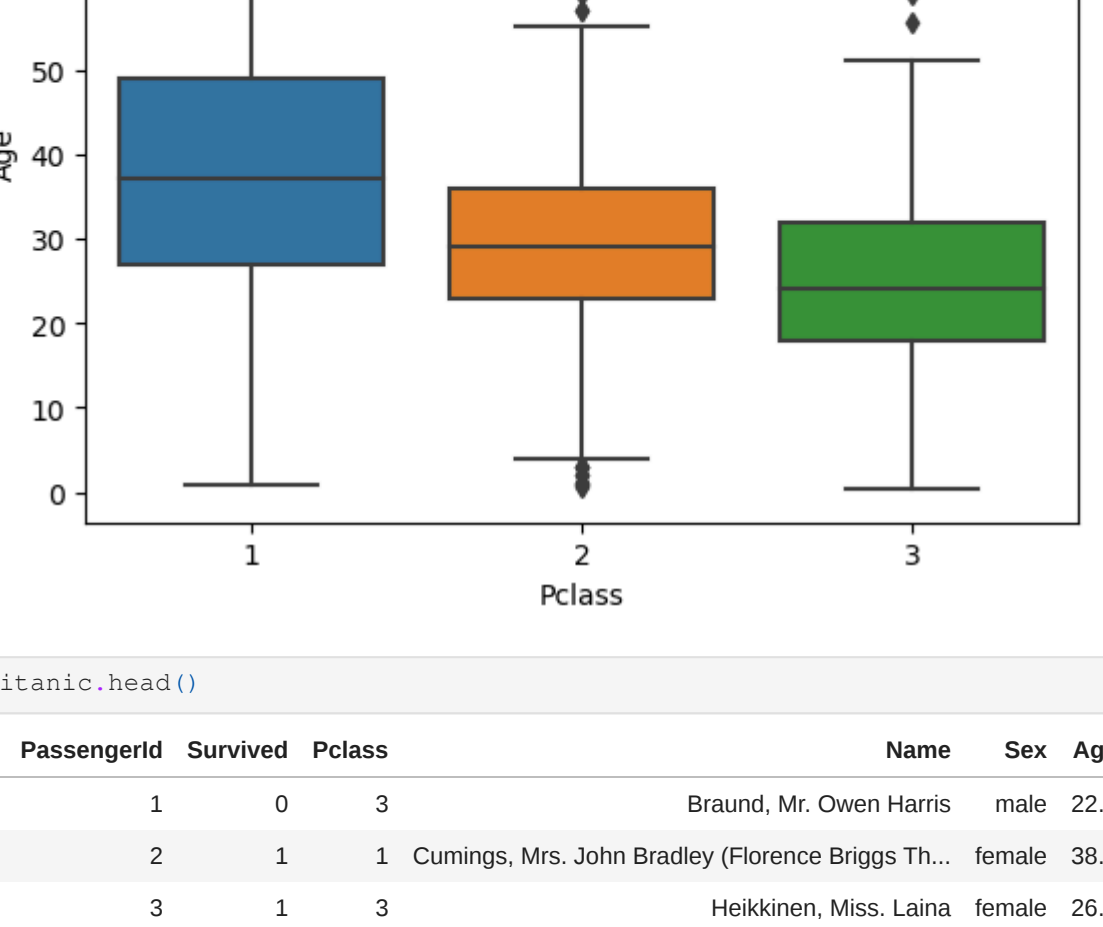


```
In [16]: titanic.isnull().sum()
Out[16]: PassengerId    0
Survived         0
Pclass           0
Name             0
Sex              0
Age             177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
In [17]: sns.heatmap(titanic.isnull(), cmap='spring')
Out[17]: <Axes: >
```



```
In [18]: sns.boxplot(x='Pclass', y='Age', data=titanic)
Out[18]: <Axes: xlabel='Pclass', ylabel='Age'>
```



```
In [19]: titanic.head()
```

	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	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikinen, 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 [20]: titanic.drop('Cabin', axis=1, inplace=True)
```

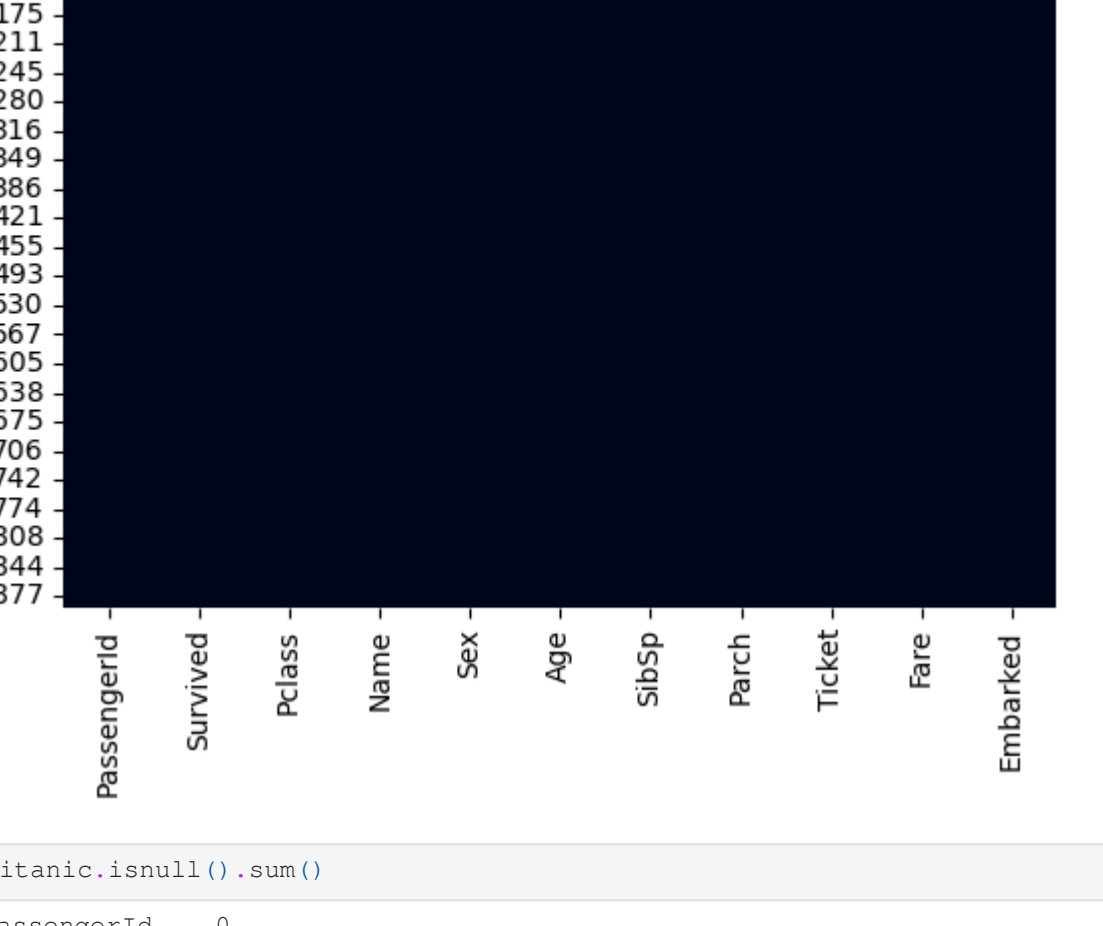
```
In [21]: titanic.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikinen, Miss. Laina	female	26.0	0	0	STON/O2 3101282	7.9250	S

```
In [22]: titanic.dropna(inplace=True)
```

```
In [23]: sns.heatmap(titanic.isnull(), cbar=False)
```

```
Out[23]: <Axes: >
```



```
In [24]: titanic.isnull().sum()
Out[24]: PassengerId    0
Survived         0
Pclass           0
Name             0
Sex              0
Age             177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         0
dtype: int64
```

```
In [25]: titanic.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C

```
In [26]: pd.get_dummies(titanic['Sex']).head()
```

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1

```
In [27]: sex=pd.get_dummies(titanic['Sex'], drop_first=True)
sex.head(3)
```

	male
0	1
1	0
2	0

```
In [29]: embarked=pd.get_dummies(titanic['Embarked'])
```

```
In [30]: embarked.head(3)
```

	C	Q	S
0	0	0	1
1	1	0	0
2	0	0	1

```
In [31]: Pol=pd.get_dummies(titanic['Pclass'], drop_first=True)
Pol.head(3)
```

	2	3
0	0	1
1	0	0
2	0	1

```
In [33]: titanic=pd.concat([titanic, sex, embarked, Pol], axis=1)
```

```
In [34]: titanic.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	male	C	Q	S	2	3
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S	1	0	0	1	0	0
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C	0	1	0	0	0	0
2	3	1	3	Heikinen, Miss. Laina	female	26.0	0	0	STON/O2 3101282	7.9250	S	0	0	0	1	0	1

```
In [35]: titanic.drop(['Name', 'PassengerId', 'Pclass', 'Ticket', 'Sex', 'Embarked'], axis=1, inplace=True)
```

```
In [36]: titanic.head(3)
```

	Survived	Age	SibSp	Parch	Fare	male	C	Q	S	2	3
0	0	22.0	1	0	7.2500	1	0	0	1	0	1
1	1	38.0	1	0	71.2833	0	1	0	0	0	0
2	1	26.0	0	0	7.9250	0	0	0	1	0	1

```
In [37]: x=titanic.drop('Survived', axis=1)
y=titanic['Survived']
```

```
In [38]: from sklearn.model_selection import train_test_split
```

```
In [39]: x_train, x_test, y_train, y_test= train_test_split(x,y, test_size=0.33, random_state=4)
```

```
In [40]: from sklearn.linear_model import LogisticRegression
lm=LogisticRegression()
```

```
In [42]: x_train.columns = x_train.columns.astype(str)
x_test.columns = x_test.columns.astype(str)
lm.fit(x_train, y_train)
```

```
C:\Users\Tanushree\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
0.0001 > tol; check your results!
```

```
Out[42]: LogisticRegression()
LogisticRegression()
```

```
In [43]: prediction=lm.predict(x_test)
```

```
In [44]: from sklearn.metrics import classification_report
```

```
In [45]: from sklearn.metrics import confusion_matrix
```

```
In [46]: from sklearn.metrics import confusion_matrix
```

```
In [47]: confusion_matrix(y_test, prediction)
```

```
Out[47]: array([[126, 19],
       [ 30, 60]], dtype=int64)
```

```
In [48]: from sklearn.metrics import accuracy_score
```

```
In [49]: accuracy_score(y_test, prediction)
```

```
Out[49]: 0.7914893617021277
```