Pre-processing Titanic

November 3, 2021

[1]:
```
# Import necessary package
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

0.0.1 Step 1: Load the dataset

[2]:
```
# Load the dataset into pandas dataframe
df = pd.read_csv("E:\\MY LECTURES\\8.2021-09-03 DATA SCIENCE (KNU)\\3.
↪Programs\\dataset\\titanic.csv")
# Change this location based on the location of dataset in your machine
```

[3]:
```
df.head()
```

[3]:

	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

[4]:
```
df.shape
```

[4]: (891, 12)

0.0.2 Step 2: Apply EDA

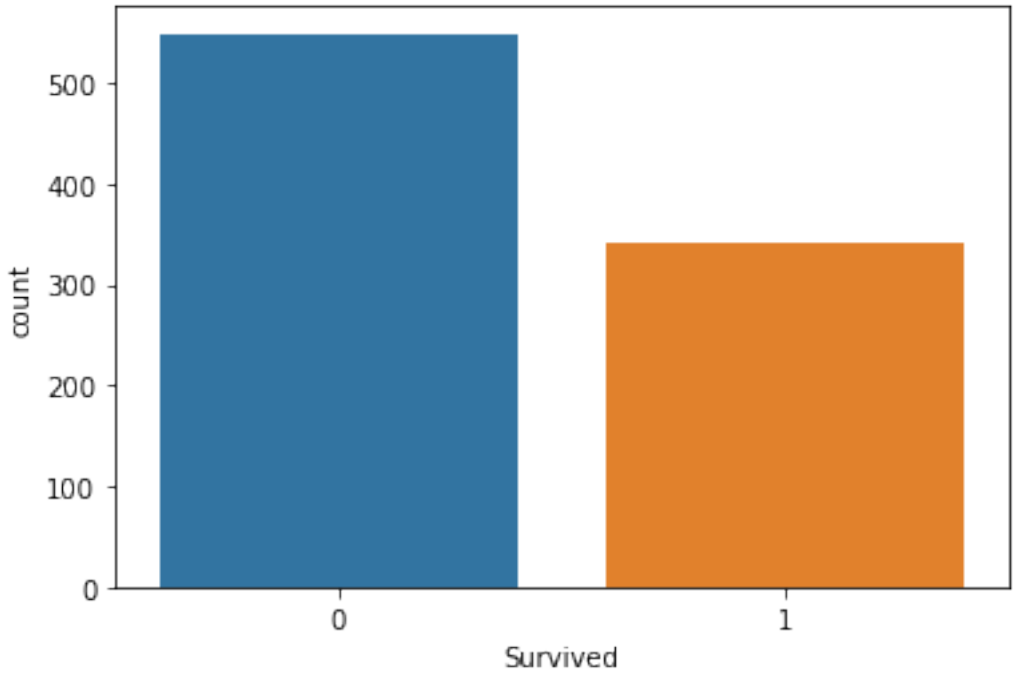
Column information in the dataset

[5]:
```
df.info()
```

```
<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
```

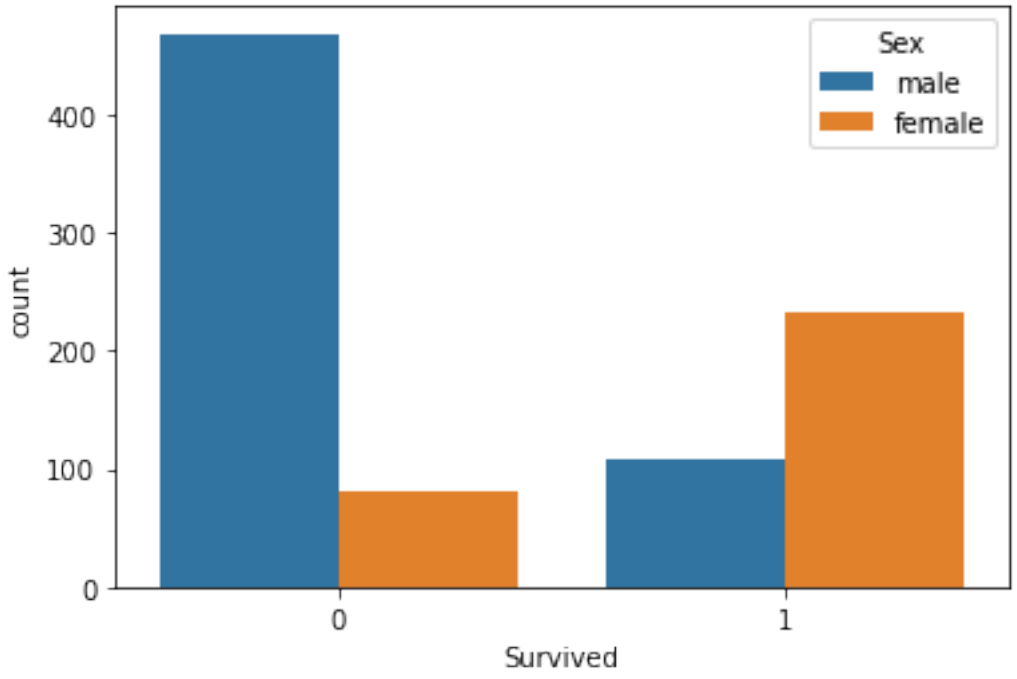
How many survived and lost life?

[6]:
```
# 0 indicates passenger did not survive and 1 indicates passenger survived
sns.countplot(x="Survived", data=df)
plt.show()
```



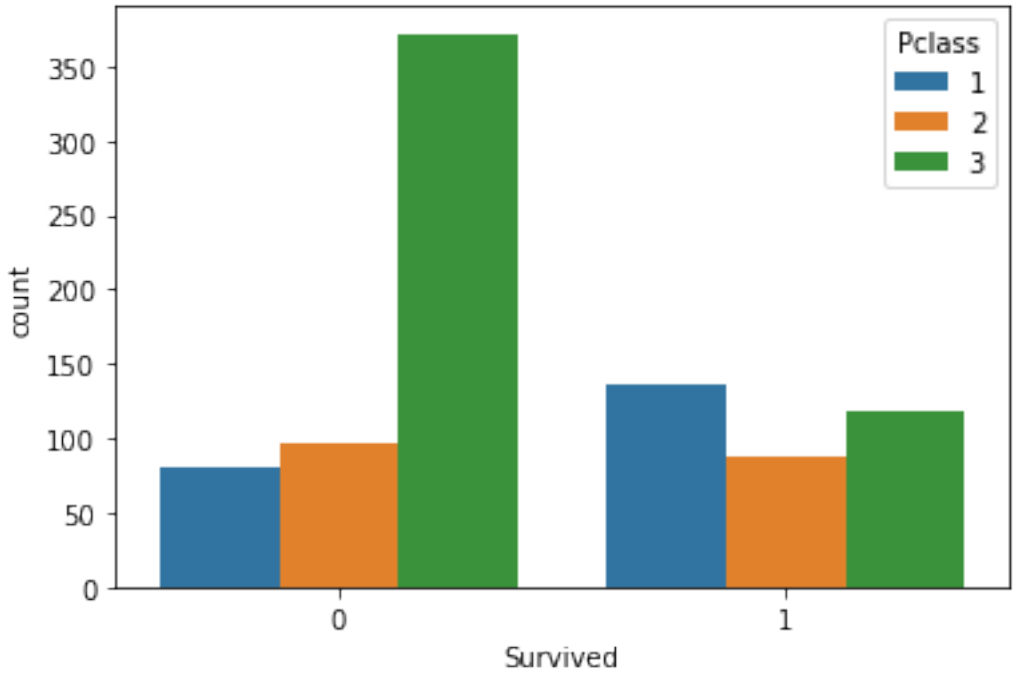
How many men and women survived and lost life?

[7]:
```
# Majority of male did not survive and majority of female survived
sns.countplot(x="Survived", hue="Sex", data=df)
plt.show()
```



How many survived and lost life based on passenger class?

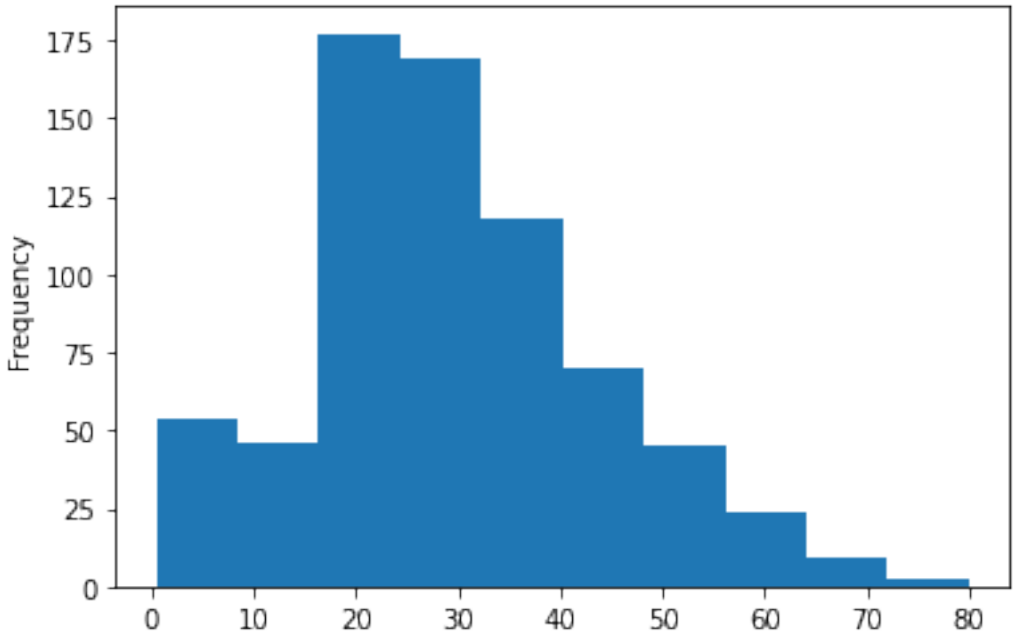
[8]:
```
# Majority of third class passengers did not survive
sns.countplot(x="Survived", hue="Pclass", data=df)
plt.show()
```



Age distribution of passengers travelled in Titanic

[9]:
```
# age 20 to 40 are the majority of the passengers
df['Age'].plot.hist()
```

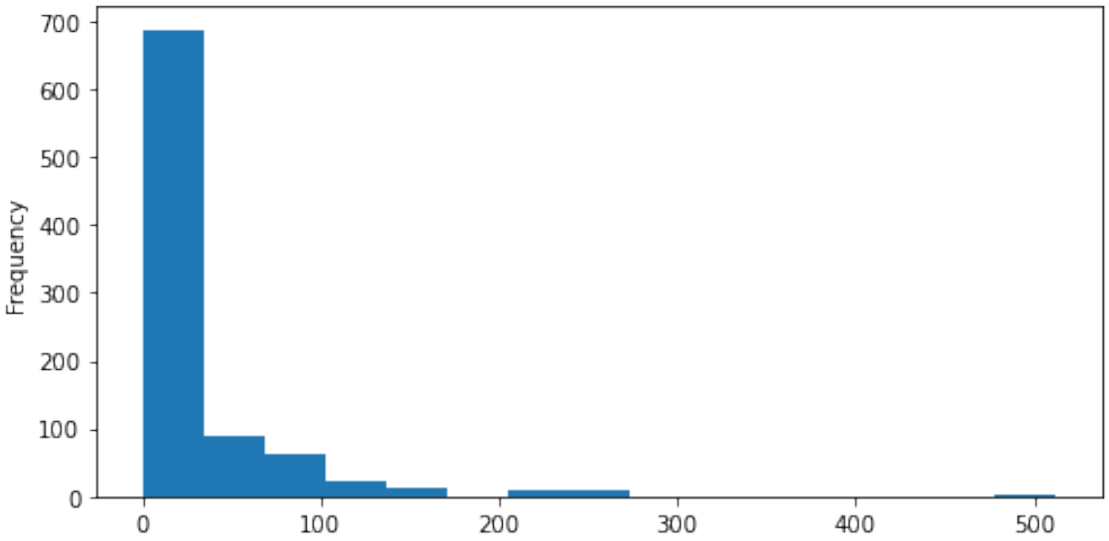
[9]: <AxesSubplot:ylabel='Frequency'>



Fare distribution

[10]:
```
df['Fare'].plot.hist(bins=15,figsize=(8,4))
```

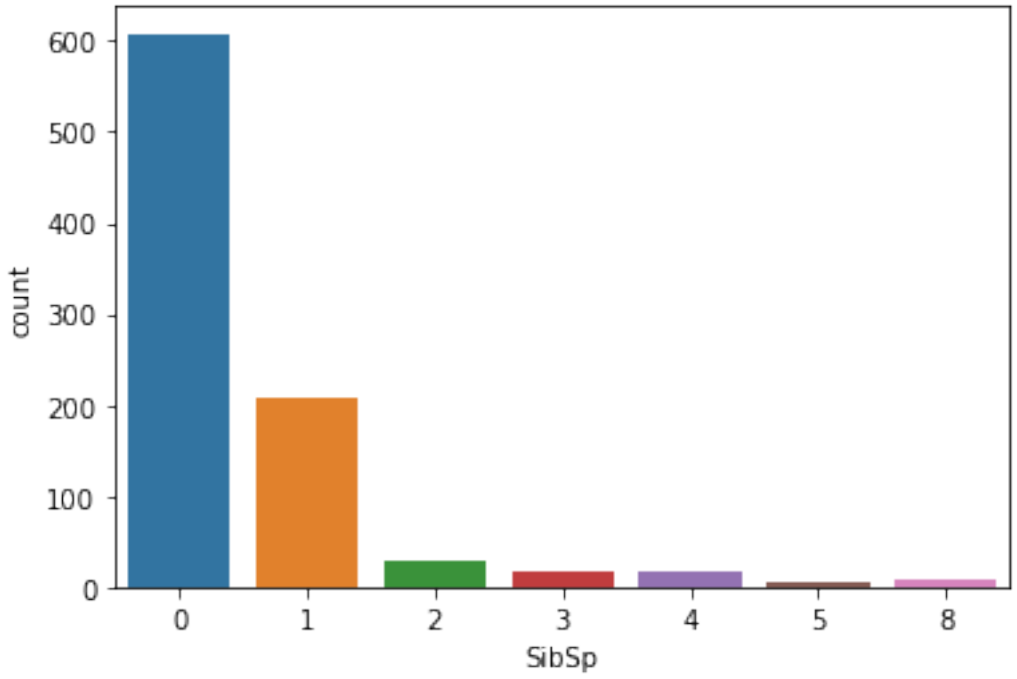
[10]: <AxesSubplot:ylabel='Frequency'>



Passengers number of siblings boarded on the ship

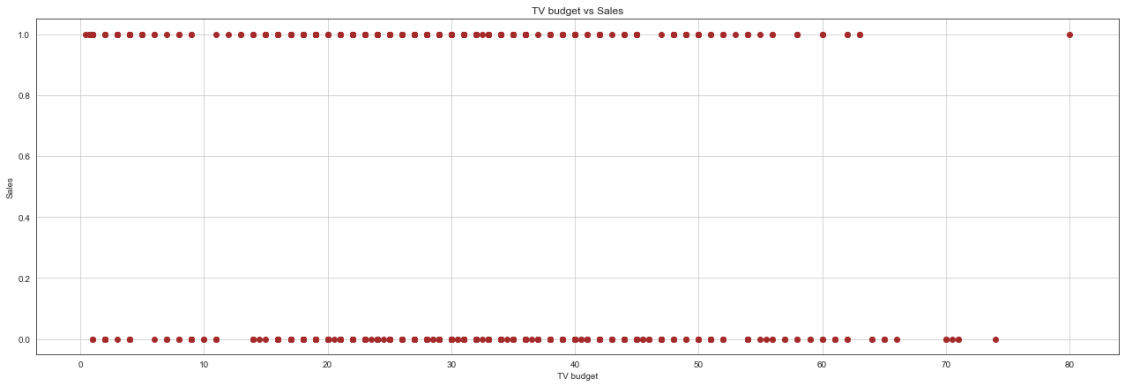
[11]:
```
# Majority are singles
sns.countplot(x="SibSp", data=df)
```

[11]: <AxesSubplot:xlabel='SibSp', ylabel='count'>



Scatter plot

[12]:
```
sns.set_style(style='white')
fig = plt.figure(figsize=(22,7))
plt.scatter(df["Age"],df["Survived"],color="brown")
plt.grid(b=None)
plt.xlabel("TV budget")
plt.ylabel("Sales")
plt.title("TV budget vs Sales")
plt.show()
```



0.0.3 Step 3. Pre-process and extract the features

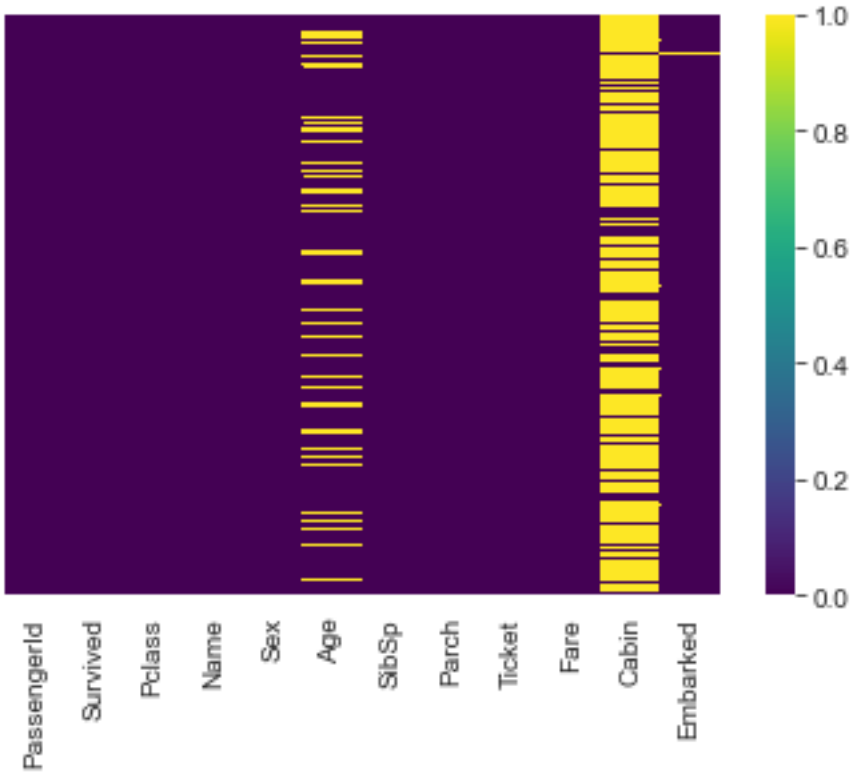
Cleaning null values

[13]:
```
df.isnull().sum()
```

[13]:
```
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
```

[14]:
```
# Heatmap displays the missing values (yellow color) in the respective column
sns.heatmap(df.isnull(), yticklabels=False, cmap="viridis" )
```

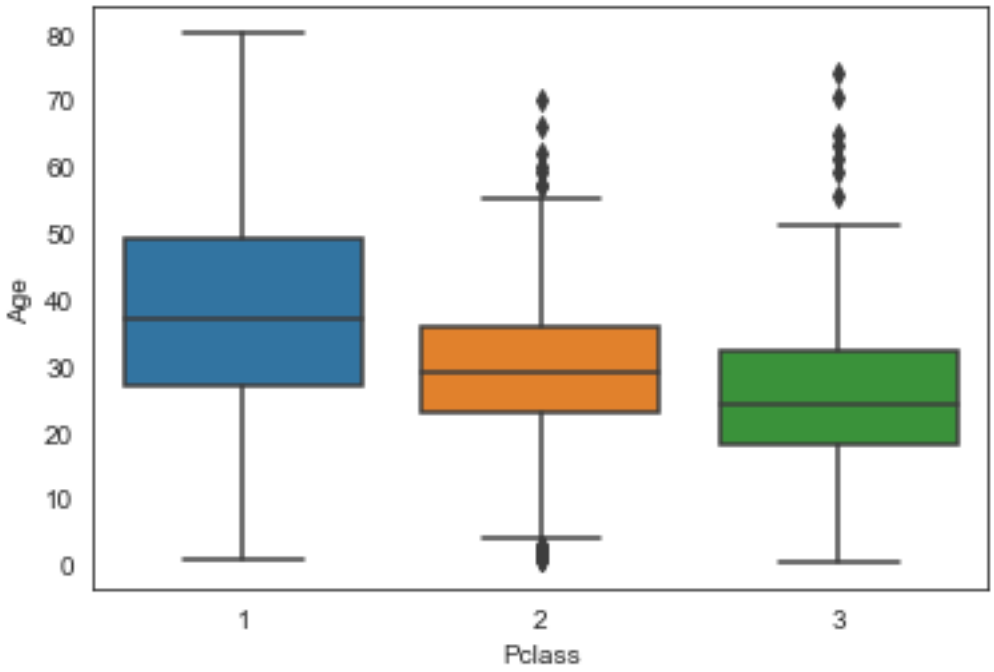
[14]:
```
<AxesSubplot:>
```



[15]:
```
# You can fill dummy values or mean or suitable value or drop the column/record␣
 ↪that contains missing (NaN) values
df.drop("Cabin",axis=1,inplace=True)
```

[16]:
```
# Value distribution in Age feature
sns.boxplot(x="Pclass", y="Age",data=df)
```

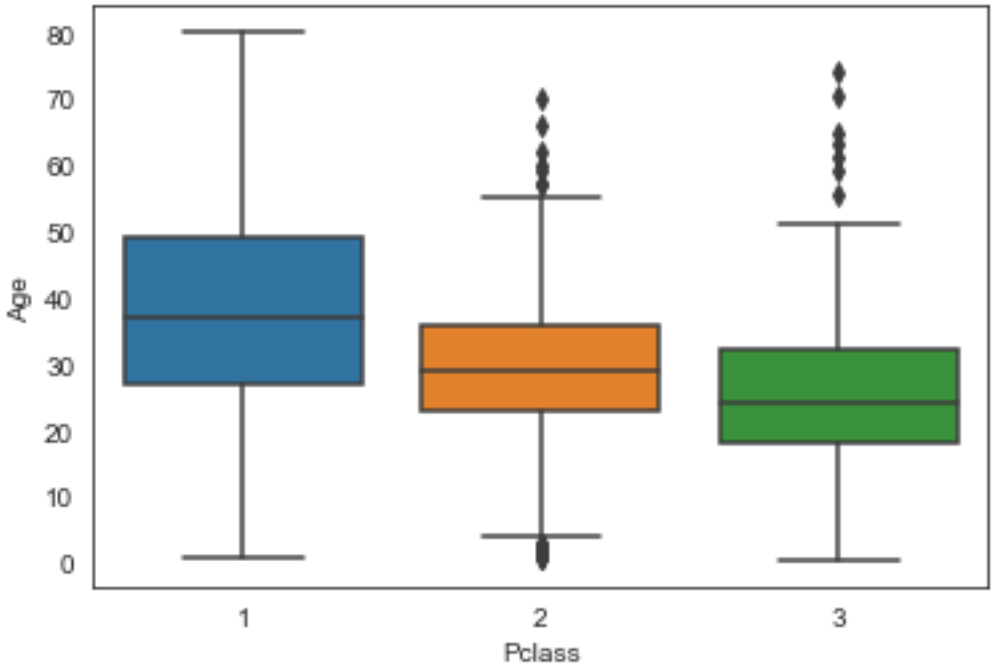
[16]: <AxesSubplot:xlabel='Pclass', ylabel='Age'>



[17]:
```
# dropping recrods that contain NaN values
df.dropna(inplace=True)
```

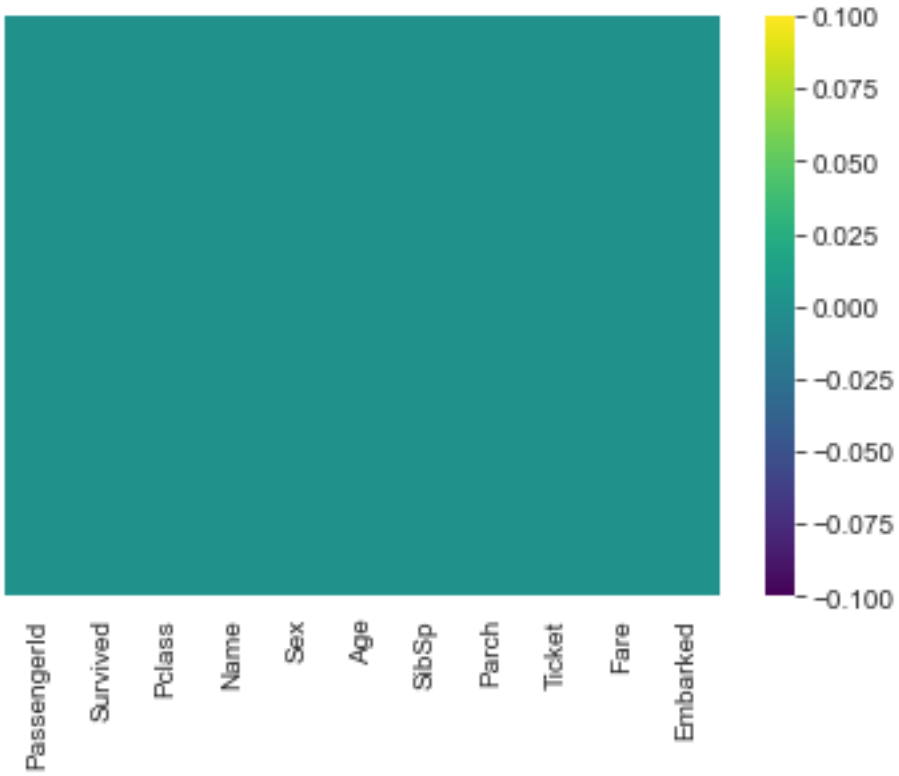
[18]:
```
# Value distribution in Age feature
sns.boxplot(x="Pclass", y="Age",data=df)
```

[18]: <AxesSubplot:xlabel='Pclass', ylabel='Age'>



[19]:
```
# Heatmap displays no missing (NaN) values
sns.heatmap(df.isnull(), yticklabels=False, cmap="viridis" )
```

[19]: <AxesSubplot:>



[20]:
```
df.isnull().sum()
```

[20]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

[21]:
```
df.shape
```

[21]:
```
(712, 11)
```