Task-1 Exploratory Data Analysis (EDA) in Titanic Dataset

Task Description: 1.Perform exploratory data analysis on Titanic dataset. 2.Data manipulation and visualization. 3.Summary statistics, visualizations, and insights from the dataset.

1.Perform exploratory data analysis on Titanic dataset

In [19]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
#Load the Titanic dataset
titanic_df = pd.read_csv(r'C:\Users\sathi\Downloads\Titanic.csv')
titanic_df
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 12 columns

In [44]:
```
def summary(titanic_df):
    df=pd.DataFrame(index=titanic_df.columns)
    df['dtypes']=titanic_df.dtypes
    df['count']=titanic_df.count()
    df['#unique']=titanic_df.nunique()
    df['missing']=titanic_df.isna().sum()
    df['missing%']=titanic_df.isna().sum()/len(titanic_df)*100
    df = pd.concat([df,(titanic_df.describe().T.drop('count',axis=1))],axis=1)
    return df
```

In [45]:
```
summary(titanic_df).style.background_gradient(cmap='YlGnBu')
```

Out[45]:

	dtypes	count	#unique	missing	missing%	mean	std	min	25%	50%	75%	max
PassengerId	int64	418	418	0	0.000000	1100.500000	120.810458	892.000000	996.250000	1100.500000	1204.750000	1309.000000
Survived	int64	418	2	0	0.000000	0.363636	0.481622	0.000000	0.000000	0.000000	1.000000	1.000000
Pclass	int64	418	3	0	0.000000	2.265550	0.841838	1.000000	1.000000	3.000000	3.000000	3.000000
Name	object	418	418	0	0.000000	nan	nan	nan	nan	nan	nan	nan
Sex	object	418	2	0	0.000000	nan	nan	nan	nan	nan	nan	nan
Age	float64	418	80	0	0.000000	30.272590	12.634534	0.170000	23.000000	30.272590	35.750000	76.000000
SibSp	int64	418	7	0	0.000000	0.447368	0.896760	0.000000	0.000000	0.000000	1.000000	8.000000
Parch	int64	418	8	0	0.000000	0.392344	0.981429	0.000000	0.000000	0.000000	0.000000	9.000000
Ticket	object	418	363	0	0.000000	nan	nan	nan	nan	nan	nan	nan
Fare	float64	417	169	1	0.239234	35.627188	55.907576	0.000000	7.895800	14.454200	31.500000	512.329200
Cabin	object	418	77	0	0.000000	nan	nan	nan	nan	nan	nan	nan
Embarked	object	418	3	0	0.000000	nan	nan	nan	nan	nan	nan	nan

In [28]:
```
# Display the first few rows of the dataset
titanic_df.head()
```

Out[28]:

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

In [73]:
```
# Display the last few rows of the dataset
titanic_df.tail()
```

Out[73]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	0	3	Spector, Mr. Woolf	male	30.27259	0	0	A.5. 3236	8.0500	Unknown	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.00000	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.50000	0	0	SOTON/O.Q. 3101262	7.2500	Unknown	S
416	1308	0	3	Ware, Mr. Frederick	male	30.27259	0	0	359309	8.0500	Unknown	S
417	1309	0	3	Peter, Master. Michael J	male	30.27259	1	1	2668	22.3583	Unknown	C

In [21]:
```
#Check the data types of each column
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
 2   Pclass       418 non-null    int64
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [22]:
```
#Check for missing values
titanic_df.isnull().sum()
```

Out[22]:
```
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 [23]:
```
#Impute missing values with the mean
titanic_df['Age'].fillna(titanic_df['Age'].mean(), inplace=True)
```

In [24]:
```
#Handling missing values in 'Embarked' column by imputing with mode
titanic_df['Embarked'].fillna(titanic_df['Embarked'].mode()[0], inplace=True)
```

In [25]:
```
#Handling missing values in 'Cabin' column by creating a new category 'Unknown'
titanic_df['Cabin'].fillna('Unknown', inplace=True)
```

In [26]:
```
#Check if there are any missing values left
titanic_df.isnull().sum()
```

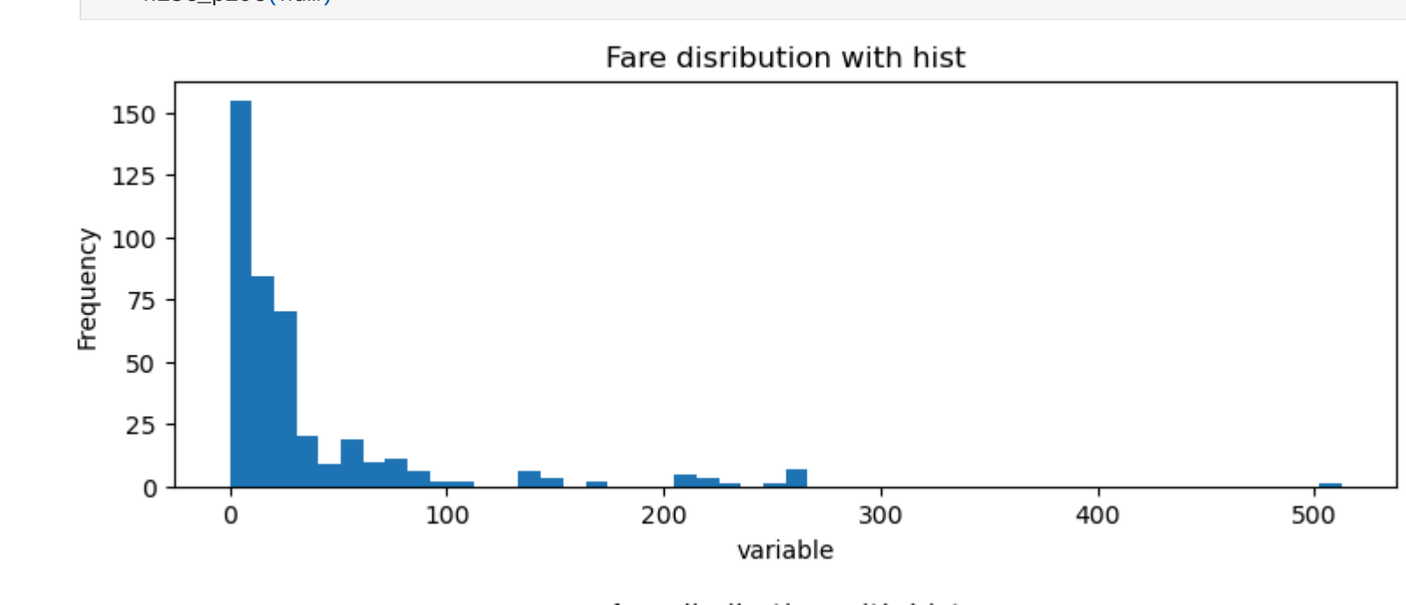
Out[26]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           1
Cabin          0
Embarked       0
dtype: int64
```

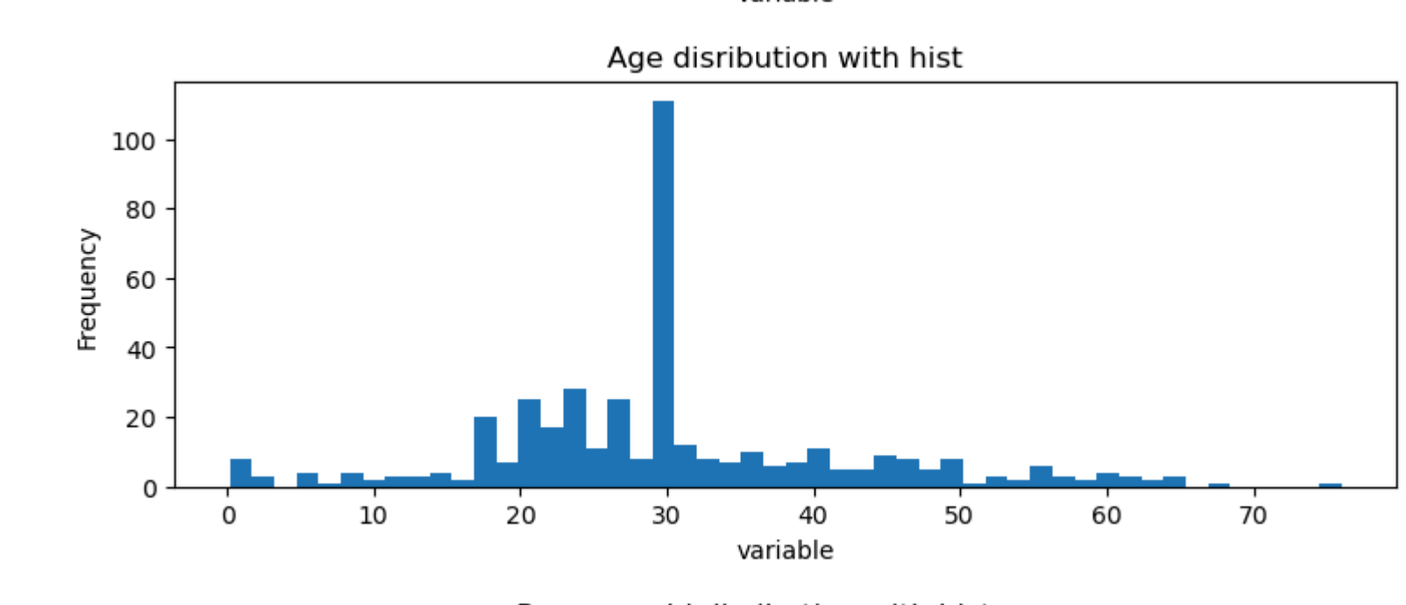
In [28]:
```
#Save encoded data
titanic_df.to_csv("titanic.csv", index=False)
```

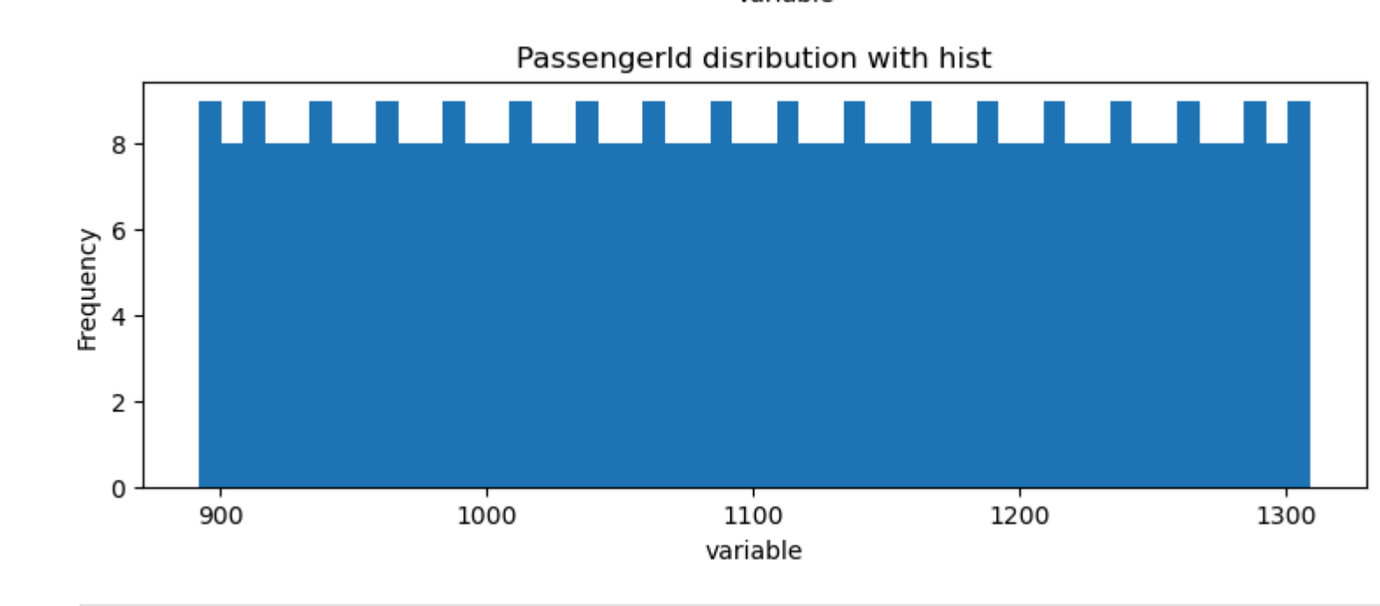
2.Data manipulation and visualization.

In [74]:
```
def hist_plot(variable):
    plt.figure(figsize=(9,3))
    plt.hist(titanic_df[variable], bins = 50)
    plt.xlabel("variable")
    plt.ylabel("Frequency")
    plt.title("{} disribution with hist".format(variable))
    plt.show()
```

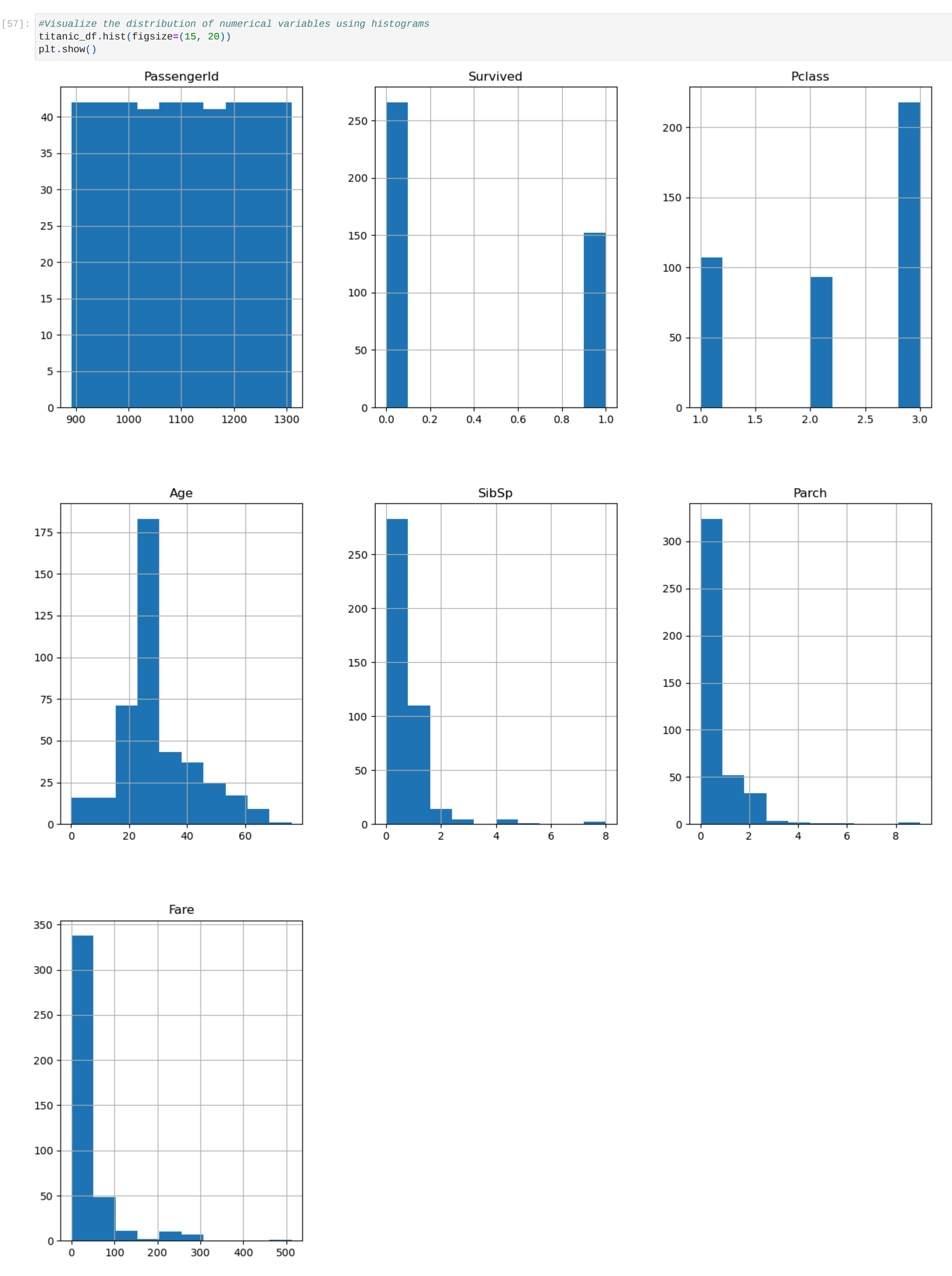
In [77]:
```
numericVar = ['Fare', 'Age', 'PassengerId']
for num in numericVar:
    hist_plot(num)
```



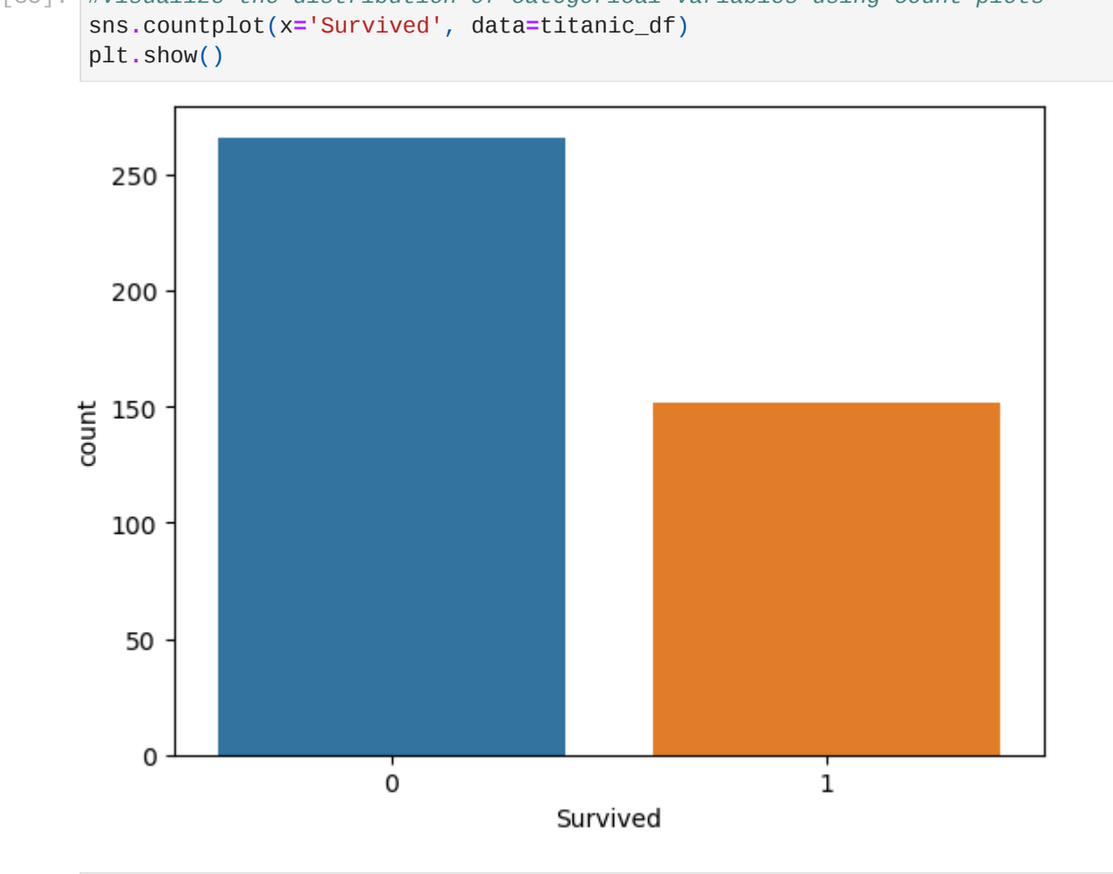




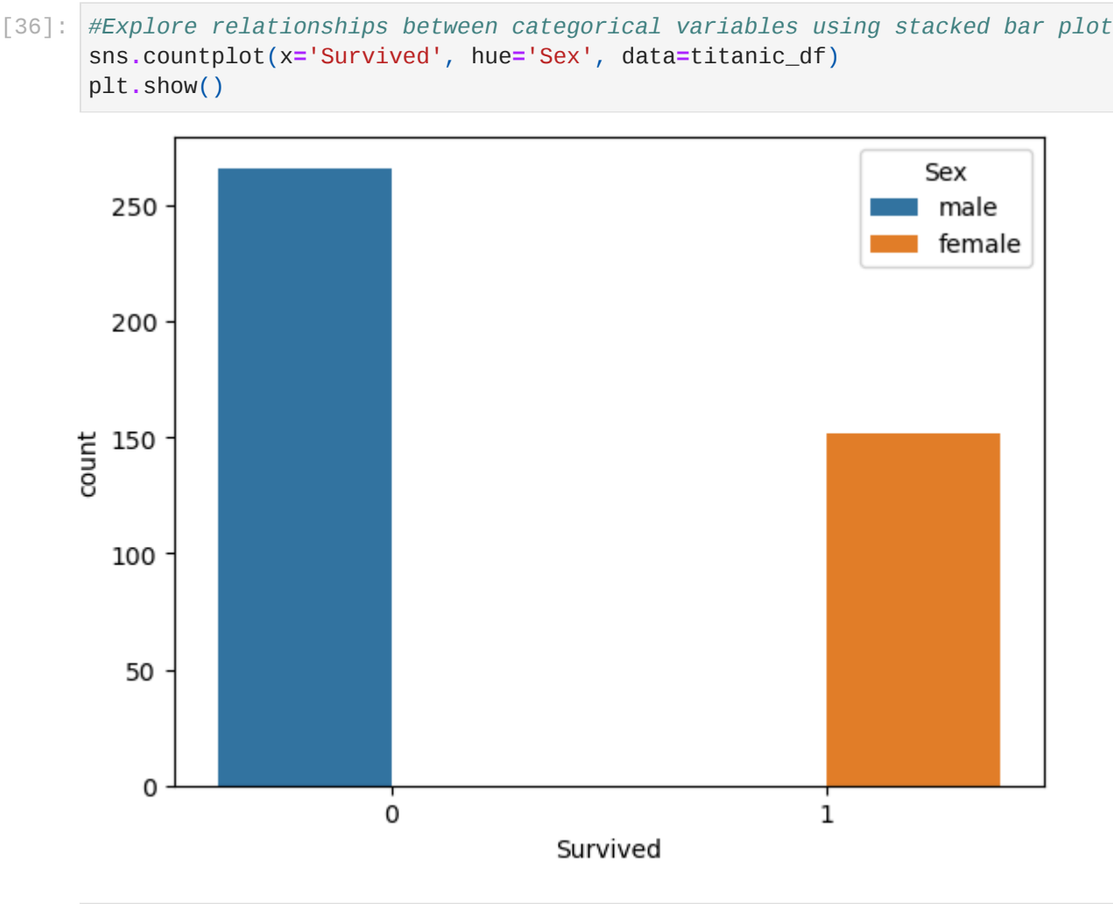
In [57]:
```
#Visualize the distribution of numerical variables using histograms
titanic_df.hist(figsize=(15, 20))
plt.show()
```



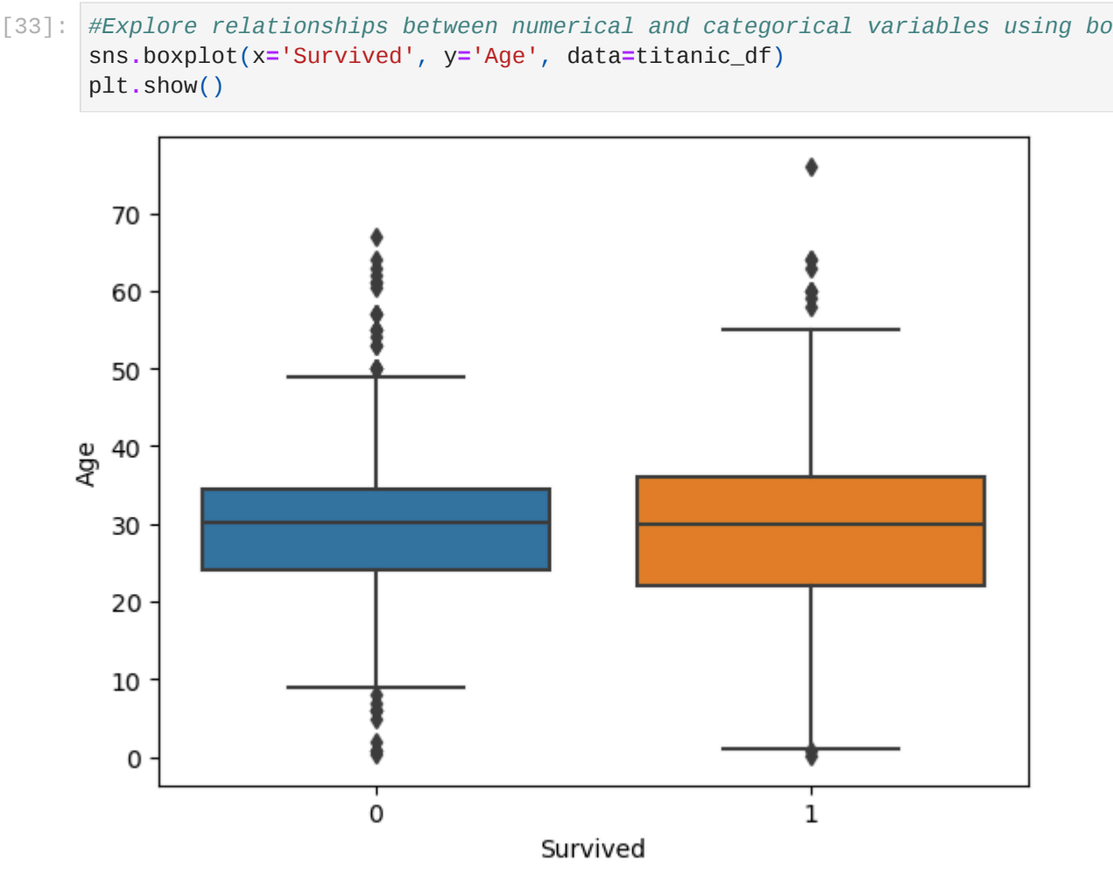
In [35]:
```
#Visualize the distribution of categorical variables using count plots
sns.countplot(x='Survived', data=titanic_df)
plt.show()
```



In [36]:
```
#Explore relationships between categorical variables using stacked bar plots
sns.countplot(x='Survived', hue='Sex', data=titanic_df)
plt.show()
```

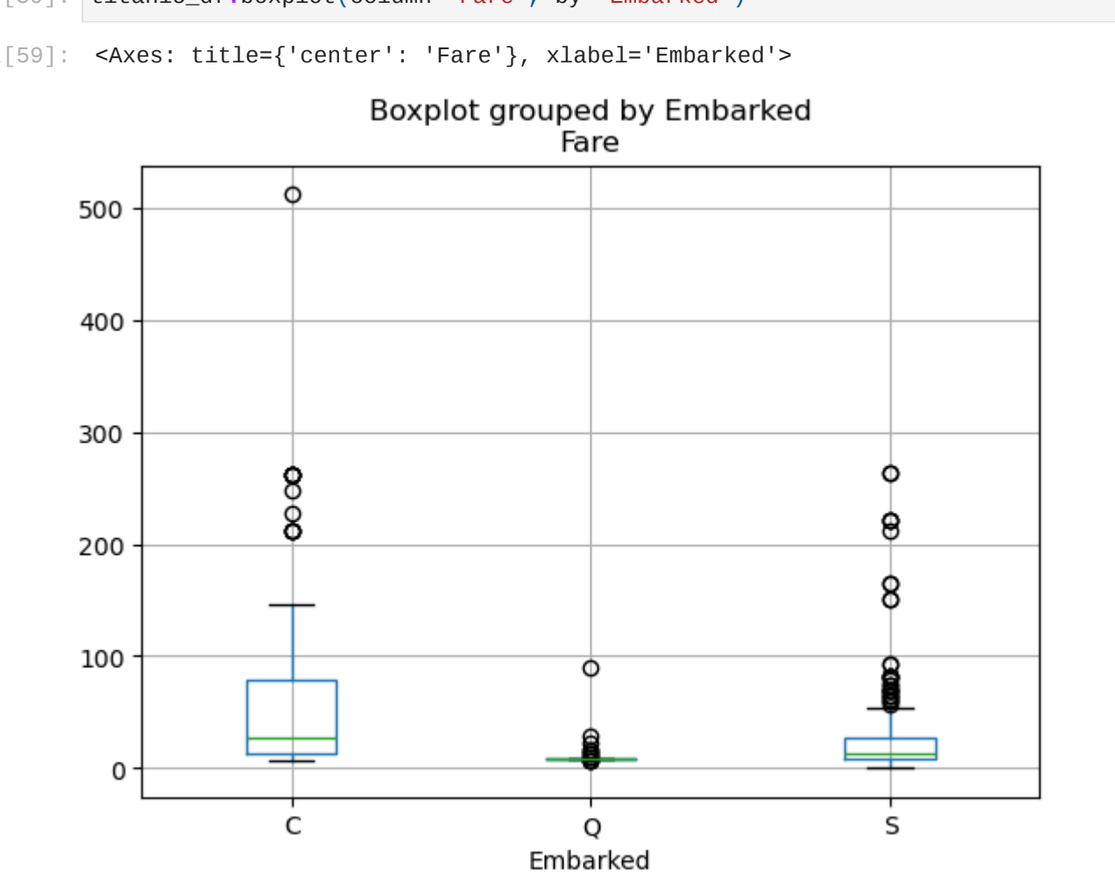


In [33]:
```
#Explore relationships between numerical and categorical variables using box plots
sns.boxplot(x='Survived', y='Age', data=titanic_df)
plt.show()
```



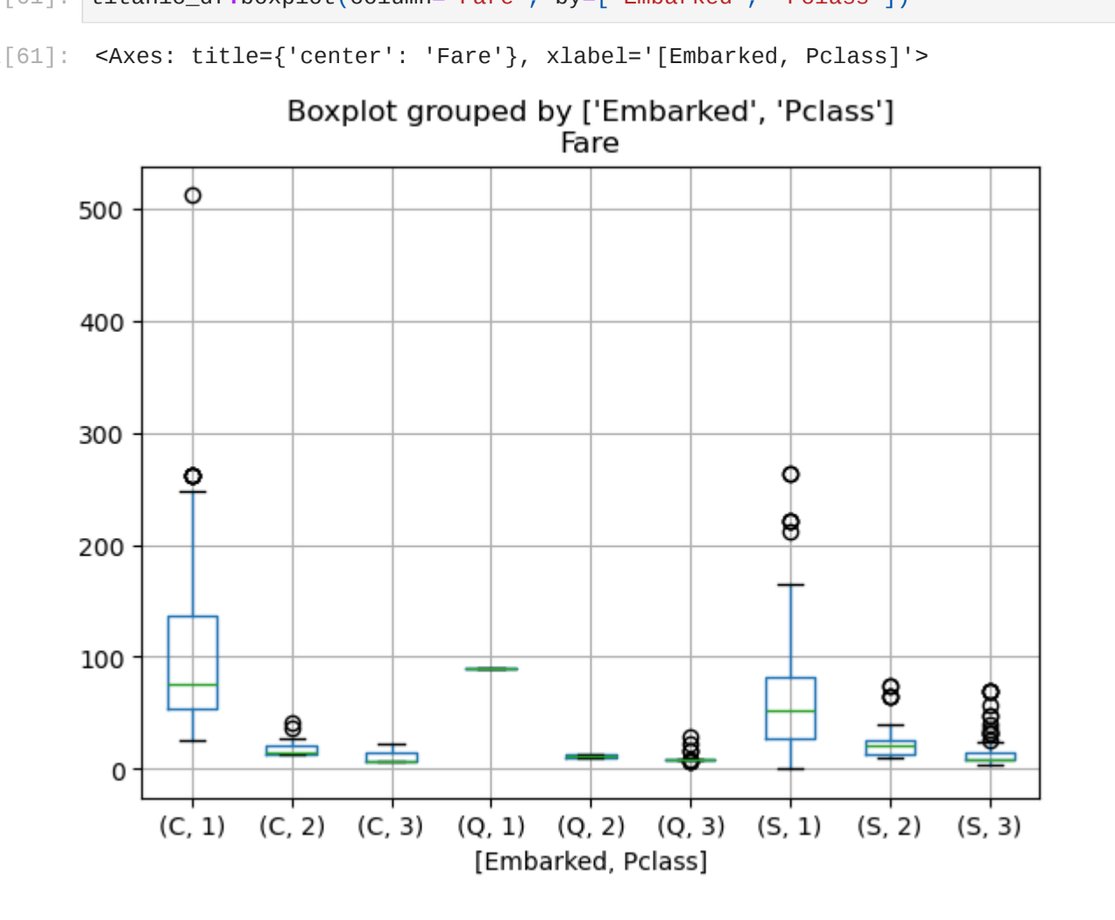
In [59]:
```
titanic_df.boxplot(column='Fare', by='Embarked')
```

Out[59]: `<Axes: title={'center': 'Fare'}, xlabel='Embarked'>`



In [61]:
```
titanic_df.boxplot(column='Fare', by=['Embarked', 'Pclass'])
```

Out[61]: `<Axes: title={'center': 'Fare'}, xlabel='[Embarked, Pclass]'>`



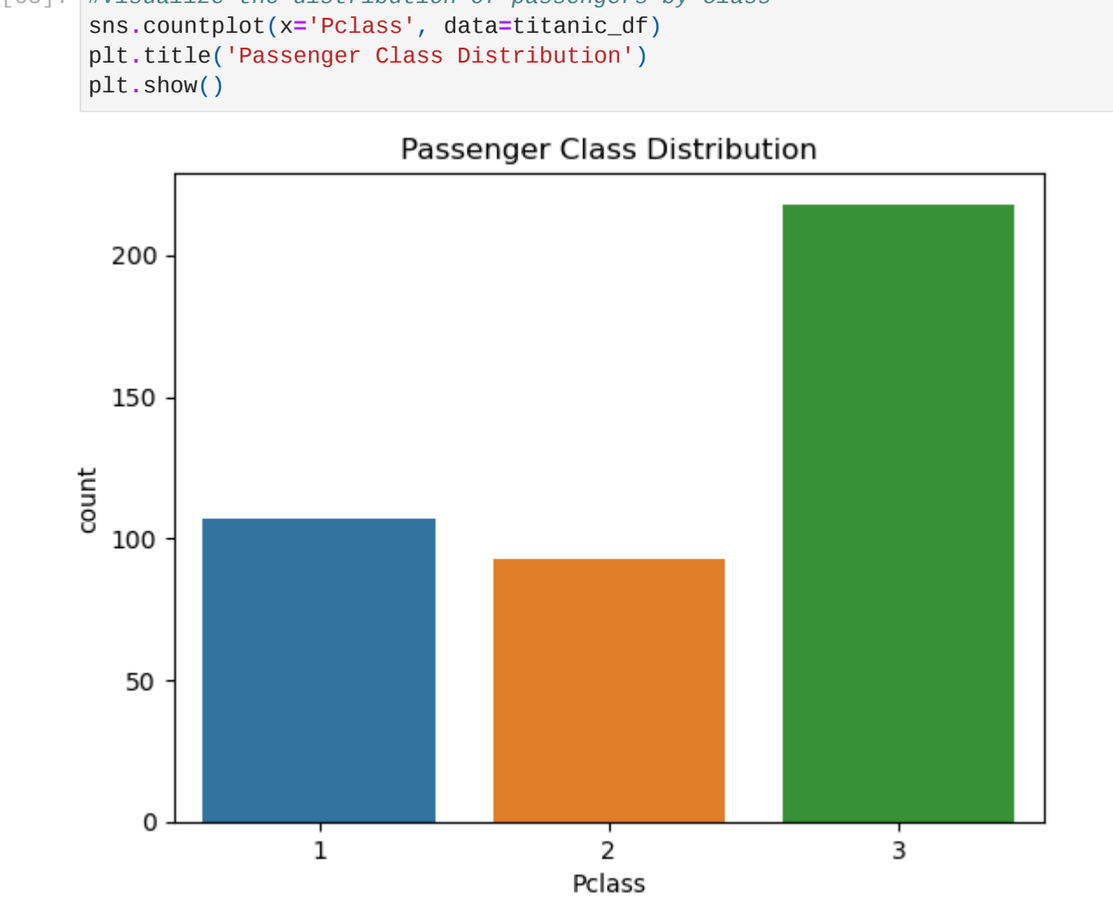
3.Summary statistics, visualizations, and insights from the dataset.

In [62]:
```
#Summary statistics of numerical variables
titanic_df.describe()
```

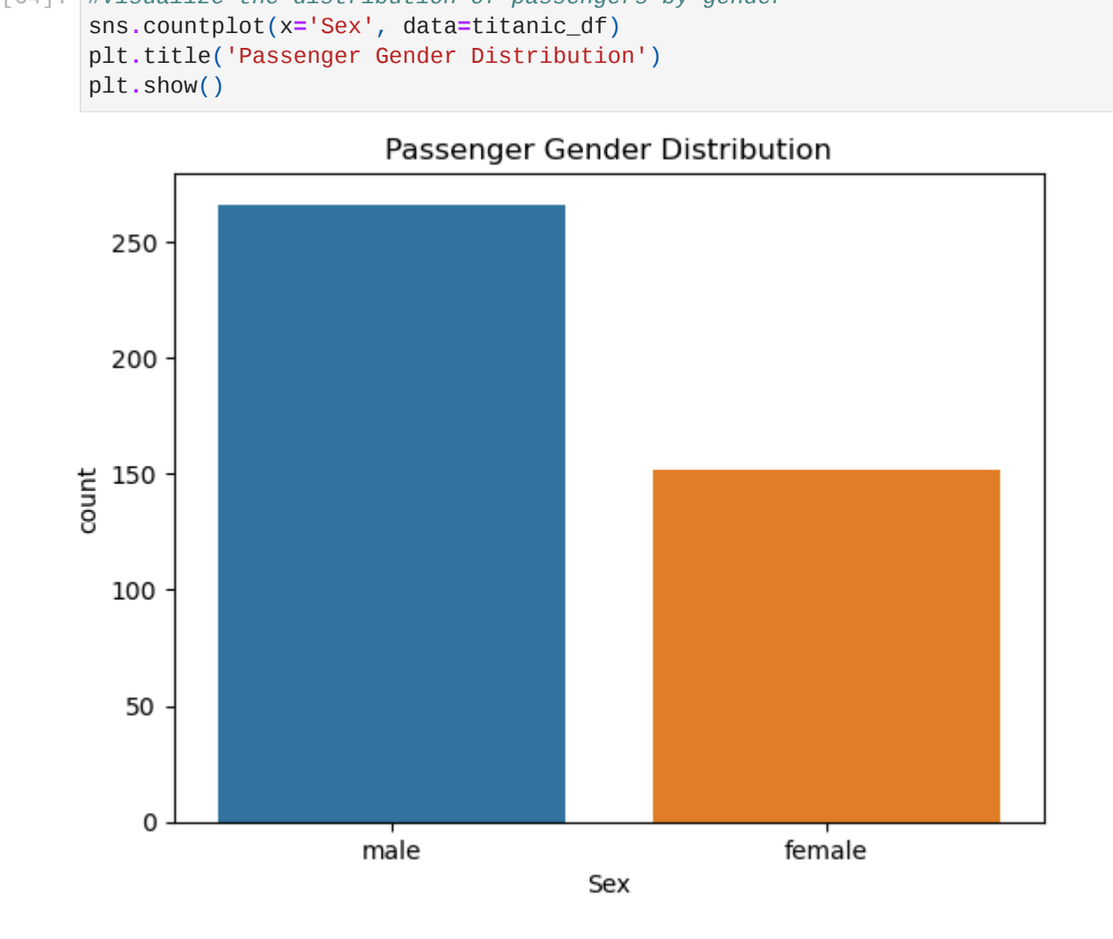
Out[62]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	12.634534	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	23.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	30.272590	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	35.750000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

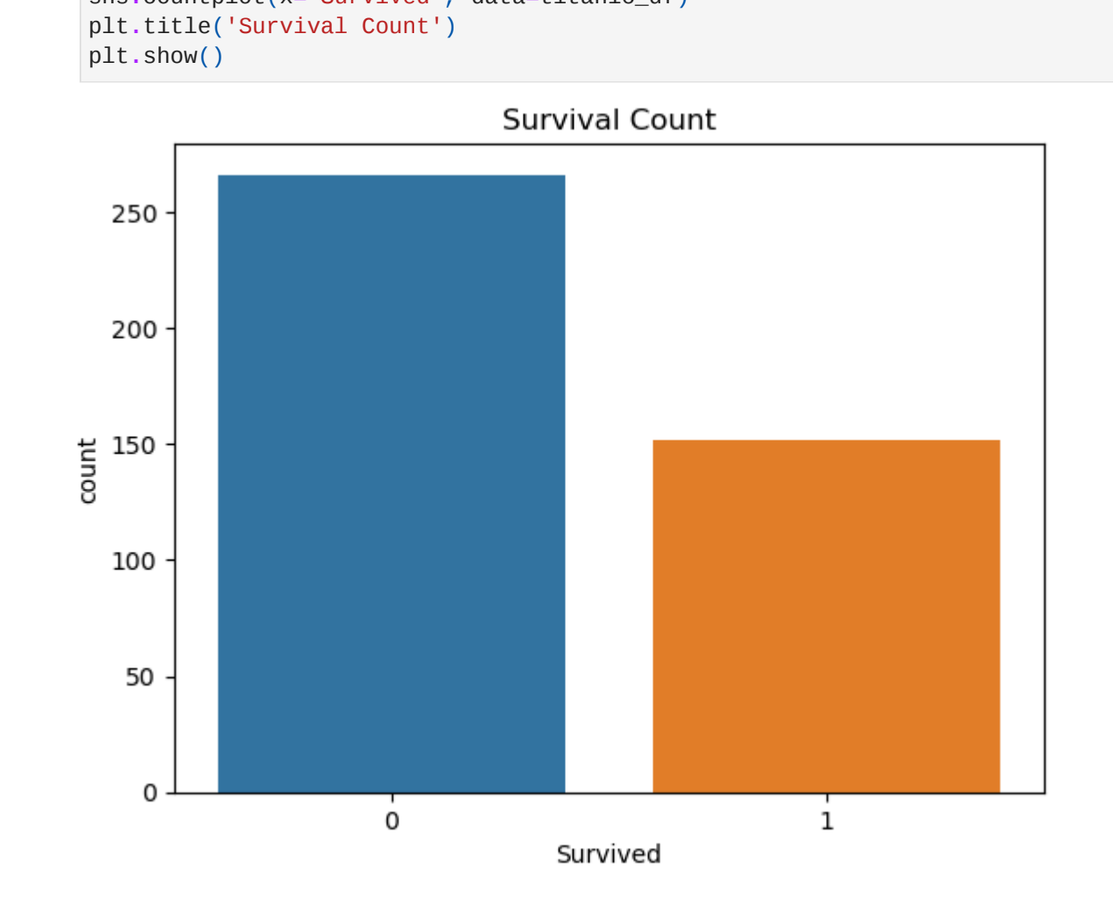
In [63]:
```
#Visualize the distribution of passengers by class
sns.countplot(x='Pclass', data=titanic_df)
plt.title('Passenger Class Distribution')
plt.show()
```



In [64]:
```
#Visualize the distribution of passengers by gender
sns.countplot(x='Sex', data=titanic_df)
plt.title('Passenger Gender Distribution')
plt.show()
```



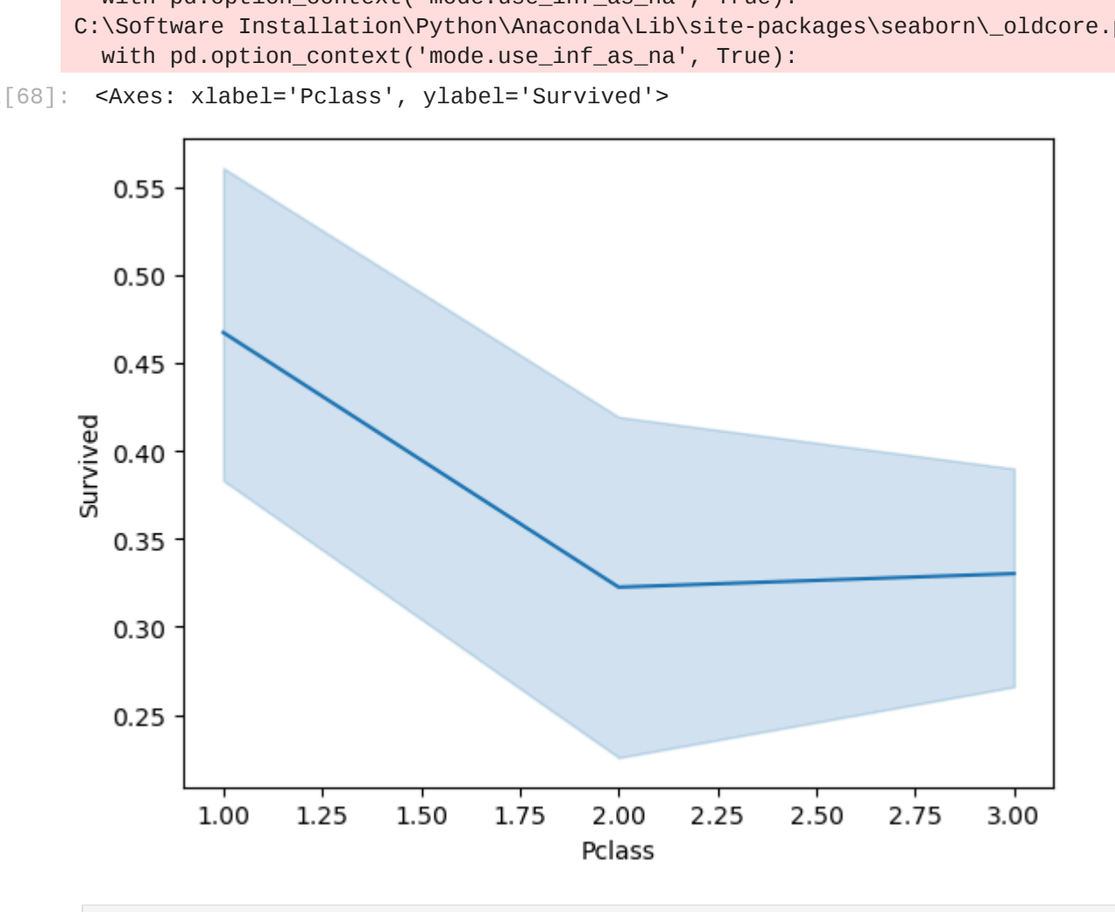
In [65]:
```
#Visualize the distribution of passengers by survival
sns.countplot(x='Survived', data=titanic_df)
plt.title('Survival Count')
plt.show()
```



In [68]:
```
sns.lineplot(data=titanic_df,x='Pclass',y='Survived')
```

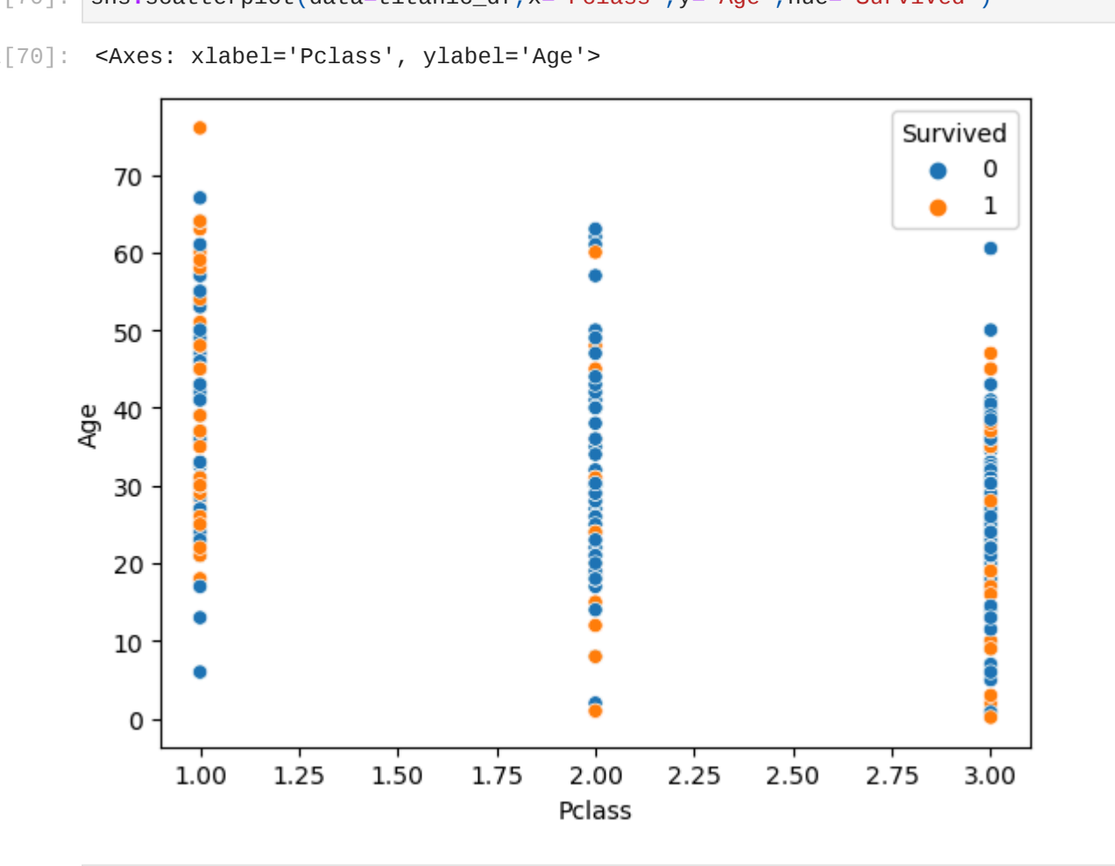
```
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[68]: `<Axes: xlabel='Pclass', ylabel='Survived'>`



In [70]:
```
sns.scatterplot(data=titanic_df,x='Pclass',y='Age',hue='Survived')
```

Out[70]: `<Axes: xlabel='Pclass', ylabel='Age'>`

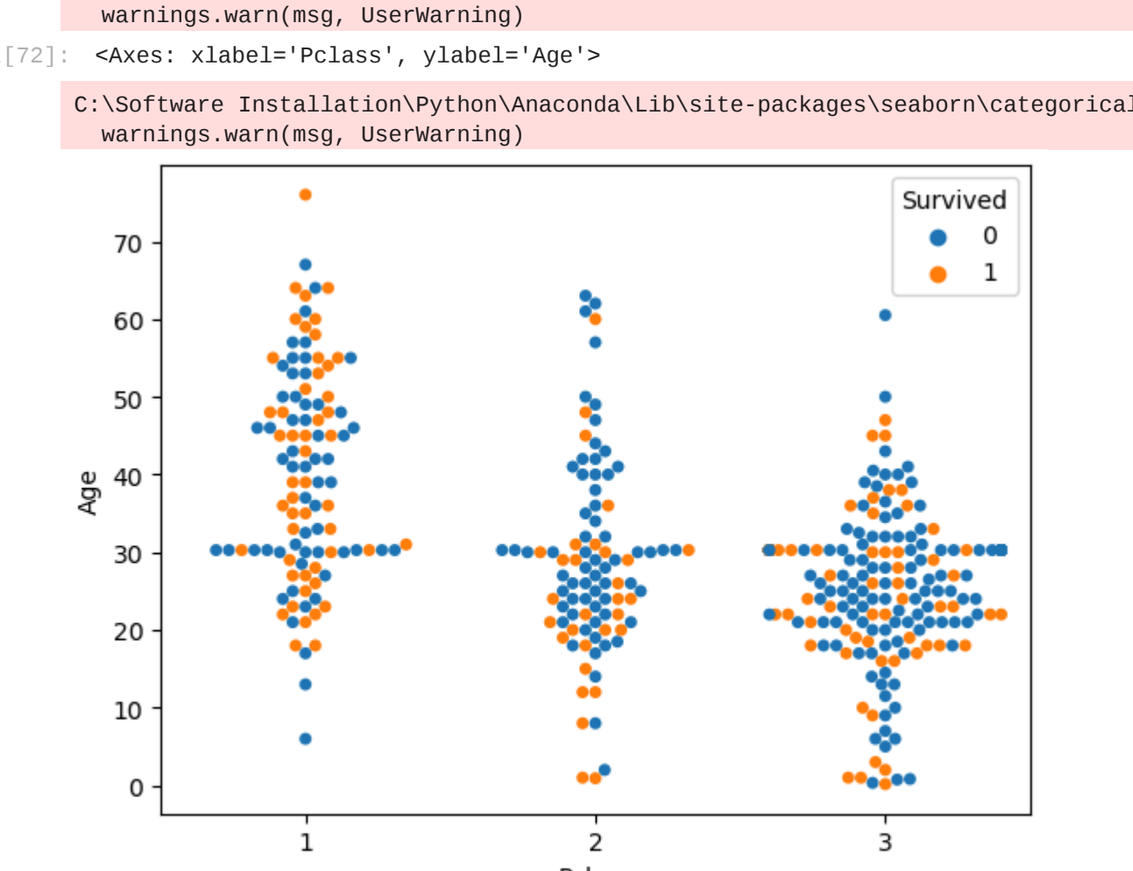


In [72]:
```
sns.swarmplot(data=titanic_df,x='Pclass',y='Age',hue='Survived')
```

```
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 23.9% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[72]: `<Axes: xlabel='Pclass', ylabel='Age'>`

```
C:\Software Installation\Python\Anaconda\Lib\site-packages\seaborn\categorical.py:3544: UserWarning: 28.4% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



Insights

1.Titanic dataset has 418 rows and 12 columns.

2.Dataset consists of Passengerid,Survival,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin and Embarked.

3.missing value from the data set is 1 from 'Fare'.

4.More people from Age of 20-30 travelled in Titanic.

5.Third class passengers are more in Titanic.

6.Male passengers are more compared to Female passengers.

7.Only 'One' Female Survived in Titanic.

In [ ]: