

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

import warnings
warnings.filterwarnings('ignore')

In [4]: # Loading the Dataset
titanic = pd.read_csv('Titanic-Dataset.csv')
titanic

Out[4]:
```

	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		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
...	...	...	...		...	...	...	...	...		...	...	...	...
886	887	0	2		Montvila, Rev. Juozas	male	27.0	0	0		211536	13.0000	NaN	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	NaN	1	2		W./C. 6607	23.4500	NaN	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	NaN	Q

891 rows × 12 columns

```
In [5]: # Reading first 5 rows
titanic.head()

Out[5]:
```

	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		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 [6]: # Reading last 5 rows
titanic.tail()

Out[6]:
```

	PassengerId	Survived	Pclass		Name	Sex	Age	SibSp	Parch		Ticket	Fare	Cabin	Embarked
886	887	0	2		Montvila, Rev. Juozas	male	27.0	0	0		211536	13.00	NaN	S
887	888	1	1		Graham, Miss. Margaret Edith	female	19.0	0	0		112053	30.00	B42	S
888	889	0	3		Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2		W./C. 6607	23.45	NaN	S
889	890	1	1		Behr, Mr. Karl Howell	male	26.0	0	0		111369	30.00	C148	C
890	891	0	3		Dodley, Mr. Patrick	male	32.0	0	0		370376	7.75	NaN	Q

```
In [7]: # Showing no. of rows and columns of dataset
titanic.shape

Out[7]:
(891, 12)
```

```
In [8]: # checking for columns
titanic.columns

Out[8]:
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
In [9]: # Checking for data types
titanic.dtypes

Out[9]:
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 [10]: # checking for duplicated values
titanic.duplicated().sum()

Out[10]:
0
```

```
In [11]: # checking for null values
nv = titanic.isna().sum().sort_values(ascending=False)
nv

Out[11]:
Cabin      687
Age         177
Embarked     2
dtype: int64
```

```
In [12]: # Checking what percentage column contain missing values
titanic.isnull().sum().sort_values(ascending=False)*100/len(titanic)

Out[12]:
Cabin      77.194377
Age        19.865320
Embarked    0.224467
PassengerId 0.000000
Survived    0.000000
Pclass      0.000000
Name        0.000000
Sex         0.000000
SibSp       0.000000
Parch       0.000000
Ticket      0.000000
Fare        0.000000
dtype: float64
```

```
In [13]: # Since Cabin column has more than 75 % null values ,So , we will drop this column
titanic.drop(columns = 'Cabin', axis = 1, inplace = True)
titanic.columns

Out[13]:
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Embarked'],
      dtype='object')
```

```
In [15]: # Filling Null Values in Age column with mean values of age column
titanic['Age'].fillna(titanic['Age'].mean(),inplace=True)

# filling null values in Embarked Column with mode values of embarked column
titanic['Embarked'].fillna(titanic['Embarked'].mode()[0],inplace=True)
```

```
In [16]: # checking for null values
titanic.isna().sum()

Out[16]:
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 [17]: # Finding no. of unique values in each column of dataset
titanic[['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
         'Parch', 'Ticket', 'Fare', 'Embarked']].nunique().sort_values()

Out[17]:
Survived      2
Sex            2
Pclass        3
Embarked      3
SibSp         7
Parch         7
Age           89
Fare         248
Ticket        681
PassengerId   891
Name          891
dtype: int64
```

```
In [20]: titanic['Survived'].unique()
titanic['Sex'].unique()

Out[20]:
array(['S', 'C', 'Q'], dtype=object)
```

```
In [21]: titanic['Pclass'].unique()
titanic['SibSp'].unique()
titanic['Parch'].unique()
titanic['Embarked'].unique()

Out[21]:
array(['S', 'C', 'Q'], dtype=object)
```

```
In [22]: #Dropping Some Unnecessary Columns
titanic.drop(columns=['PassengerId','Name','Ticket'],axis=1,inplace=True)
titanic.columns

Out[22]:
Index(['Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare',
       'Embarked'],
      dtype='object')
```

```
In [23]: # Showing information about the dataset
titanic.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  --
 0   Survived    891 non-null    int64
 1   Pclass      891 non-null    int64
 2   Sex         891 non-null    object
 3   Age         891 non-null    float64
 4   SibSp       891 non-null    int64
 5   Parch       891 non-null    int64
 6   Fare        891 non-null    float64
 7   Embarked    891 non-null    object
dtypes: float64(2), int64(4), object(2)
memory usage: 55.8+ KB
```

```
In [24]: # showing info. about numerical columns
titanic.describe()

Out[24]:
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
In [25]: # showing info. about categorical columns
titanic.describe(include='O')

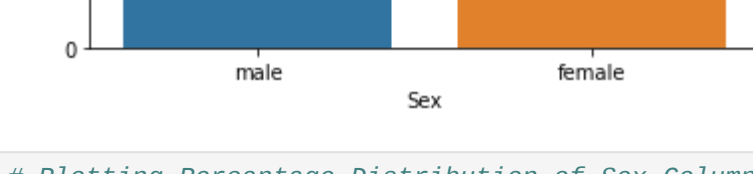
Out[25]:
```

	Sex	Embarked
count	891	891
unique	2	3
top	male	S
freq	577	646

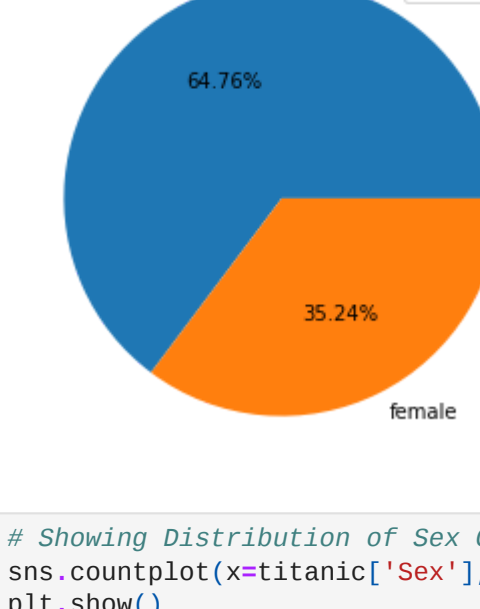
```
In [26]: #Data Visualization
d1 = titanic['Sex'].value_counts()
d1

Out[26]:
male      577
female    314
Name: Sex, dtype: int64
```

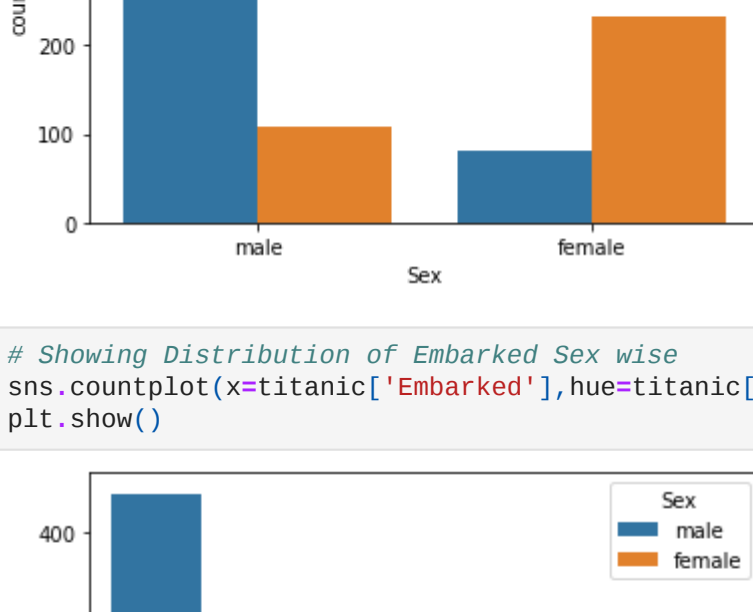
```
In [27]: # Plotting Count plot for sex column
sns.countplot(x=titanic['Sex'])
plt.show()
```



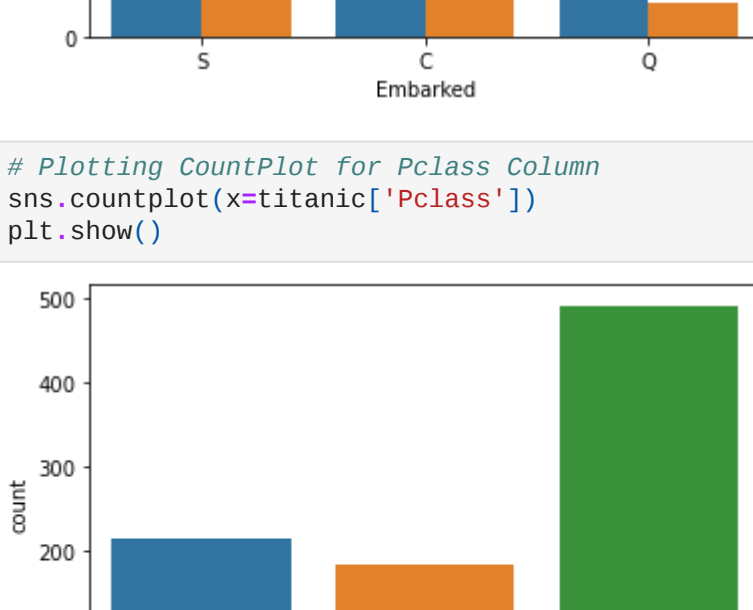
```
In [28]: # Plotting Percentage Distribution of Sex Column
plt.figure(figsize=(5,5))
plt.pie(d1.values,labels=d1.index,autopct='% .2f%%')
plt.legend()
plt.show()
```



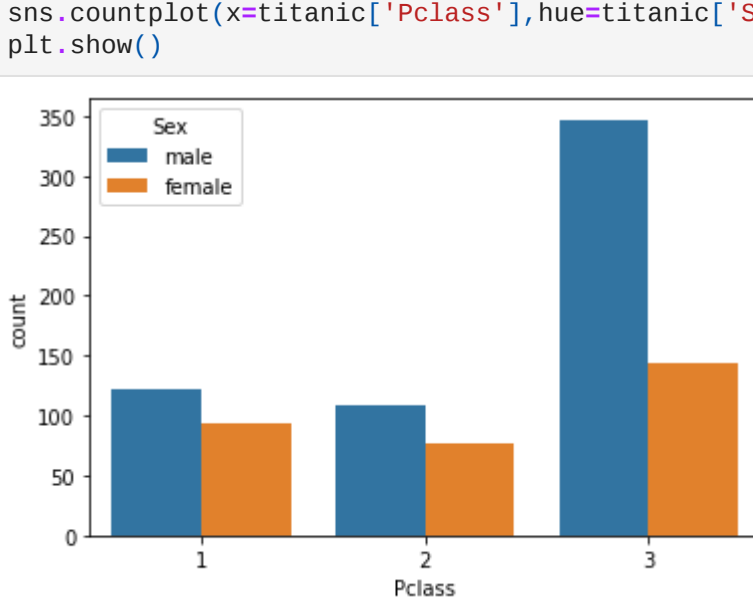
```
In [29]: # Showing Distribution of Sex Column Survived Wise
sns.countplot(x=titanic['Sex'],hue=titanic['Survived']) # In Sex (0 represents female and 1 represents male)
plt.show()
```



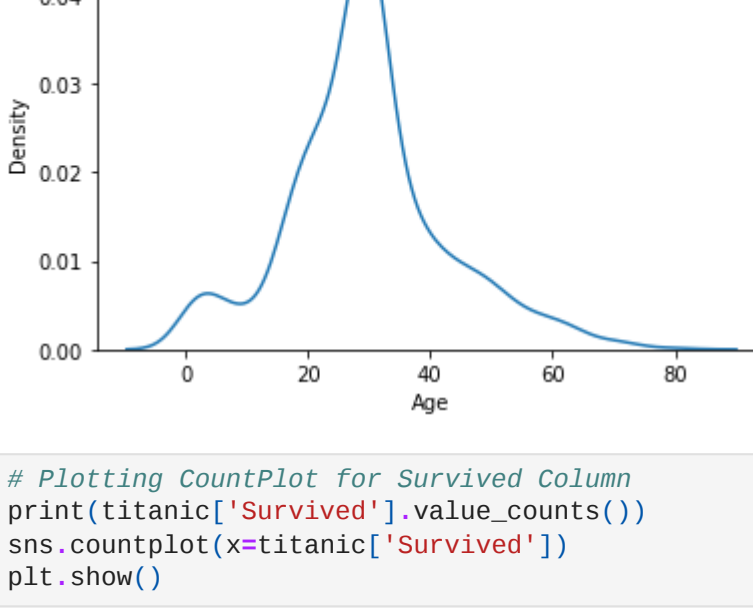
```
In [30]: # Showing Distribution of Embarked Sex wise
sns.countplot(x=titanic['Embarked'],hue=titanic['Sex'])
plt.show()
```



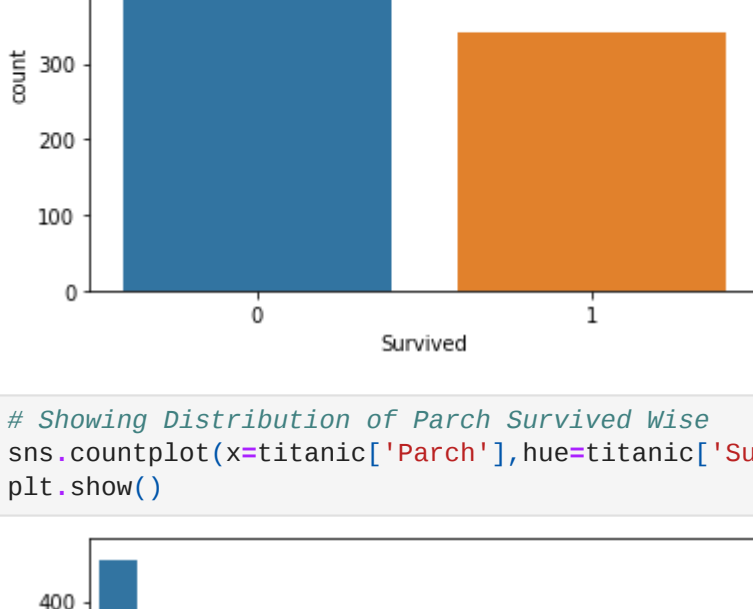
```
In [31]: # Plotting CountPlot for Pclass Column
sns.countplot(x=titanic['Pclass'])
plt.show()
```



```
In [32]: # Showing Distribution of Pclass Sex wise
sns.countplot(x=titanic['Pclass'],hue=titanic['Sex'])
plt.show()
```

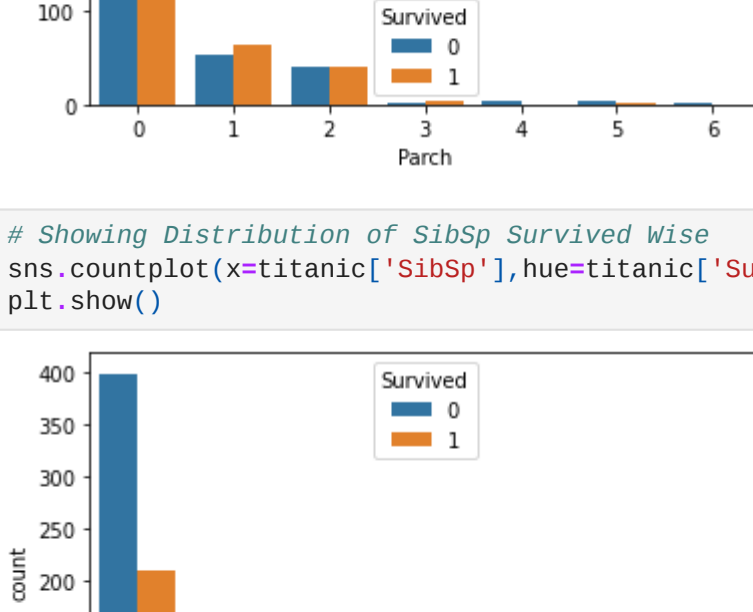


```
In [33]: # Age Distribution
sns.kdeplot(x=titanic['Age'])
plt.show()
```

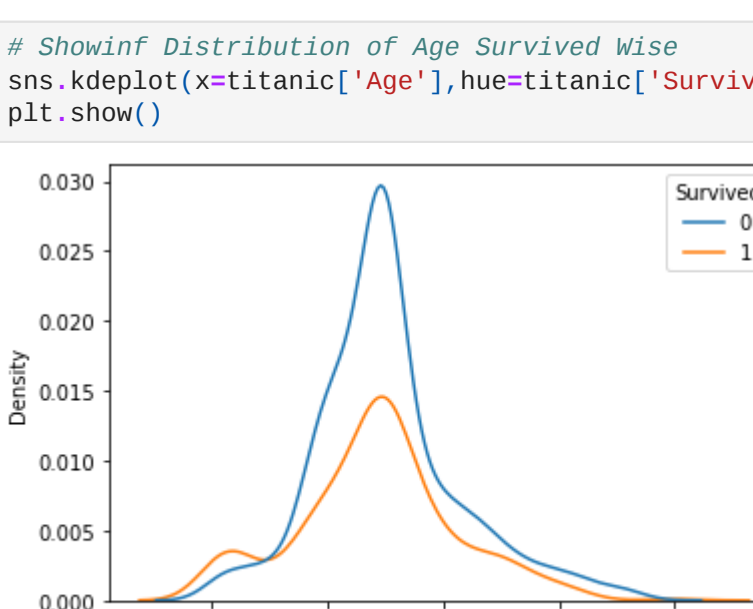


```
In [34]: # Plotting CountPlot for Survived Column
print(titanic['Survived'].value_counts())
sns.countplot(x=titanic['Survived'])
plt.show()
```

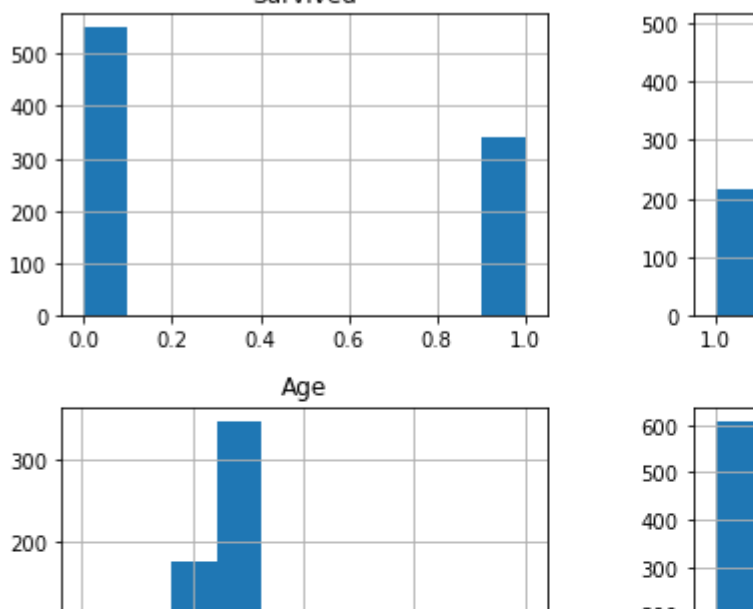
0 549  
1 342  
Name: Survived, dtype: int64



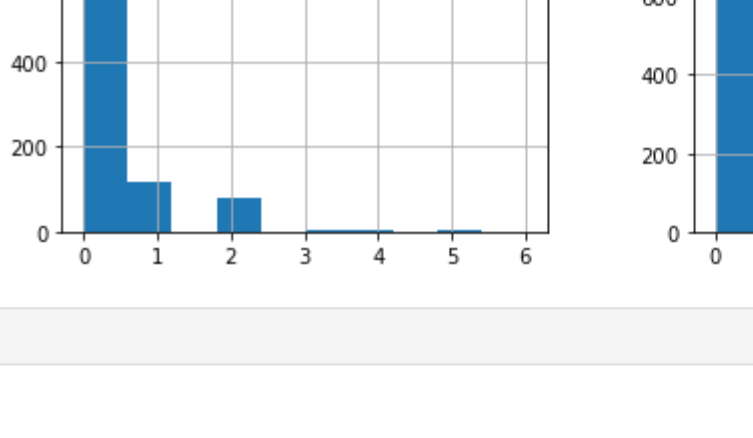
```
In [35]: # Showing Distribution of Parch Survived Wise
sns.countplot(x=titanic['Parch'],hue=titanic['Survived'])
plt.show()
```



```
In [36]: # Showing Distribution of SibSp Survived Wise
sns.countplot(x=titanic['SibSp'],hue=titanic['Survived'])
plt.show()
```



```
In [37]: # Showinf Distribution of Age Survived Wise
sns.kdeplot(x=titanic['Age'],hue=titanic['Survived'])
plt.show()
```



```
In [38]: # Plotting Histogram for Dataset
titanic.hist(figsize=(10,10))
plt.show()
```



```
In [ ]:
```