

## Import Modules

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

## Loading the dataset

```
In [2]: train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
train.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [3]: `train.describe()`

Out[3]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
<b>count</b>	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
<b>mean</b>	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
<b>std</b>	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
<b>min</b>	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
<b>25%</b>	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
<b>50%</b>	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
<b>75%</b>	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
<b>max</b>	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

## statistical info

In [4]: `train.info()`

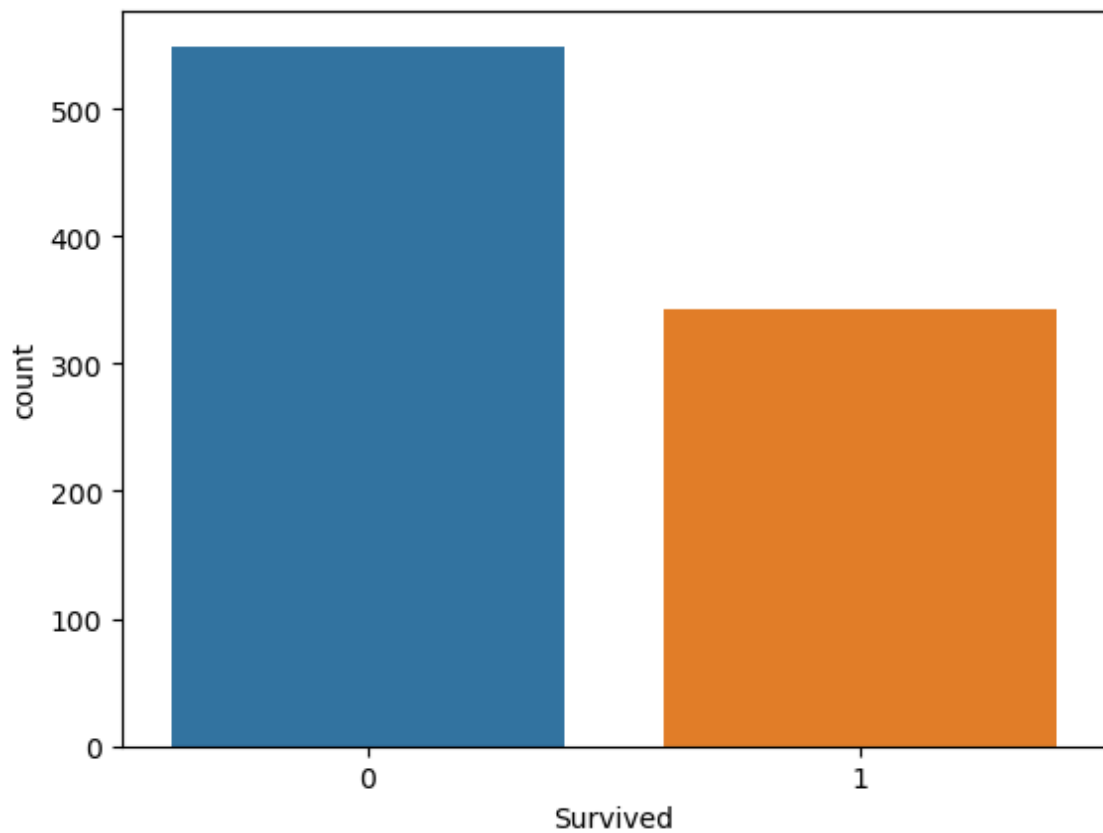
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   PassengerId   891 non-null    int64  
 1   Survived      891 non-null    int64  
 2   Pclass        891 non-null    int64  
 3   Name          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
```

## datatype info

## Exploratory Data Analysis

```
In [5]: sns.countplot(train['Survived'])
```

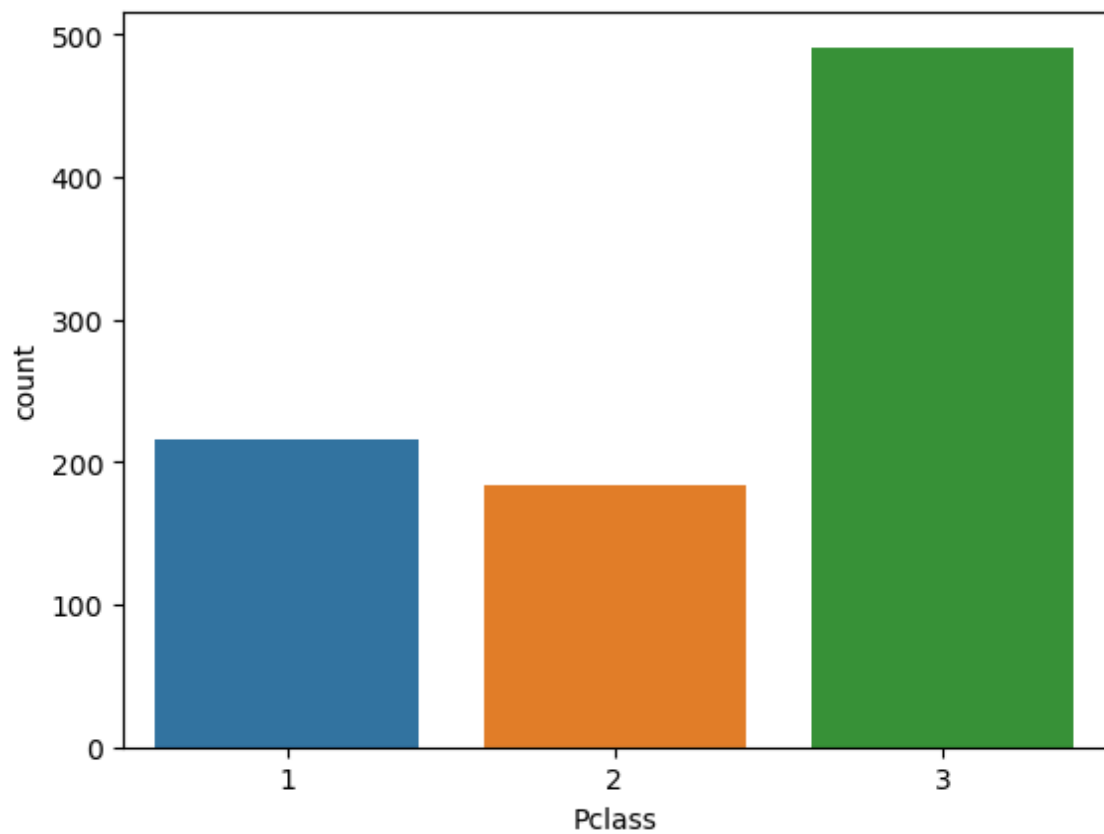
```
Out[5]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```



## categorical attributes

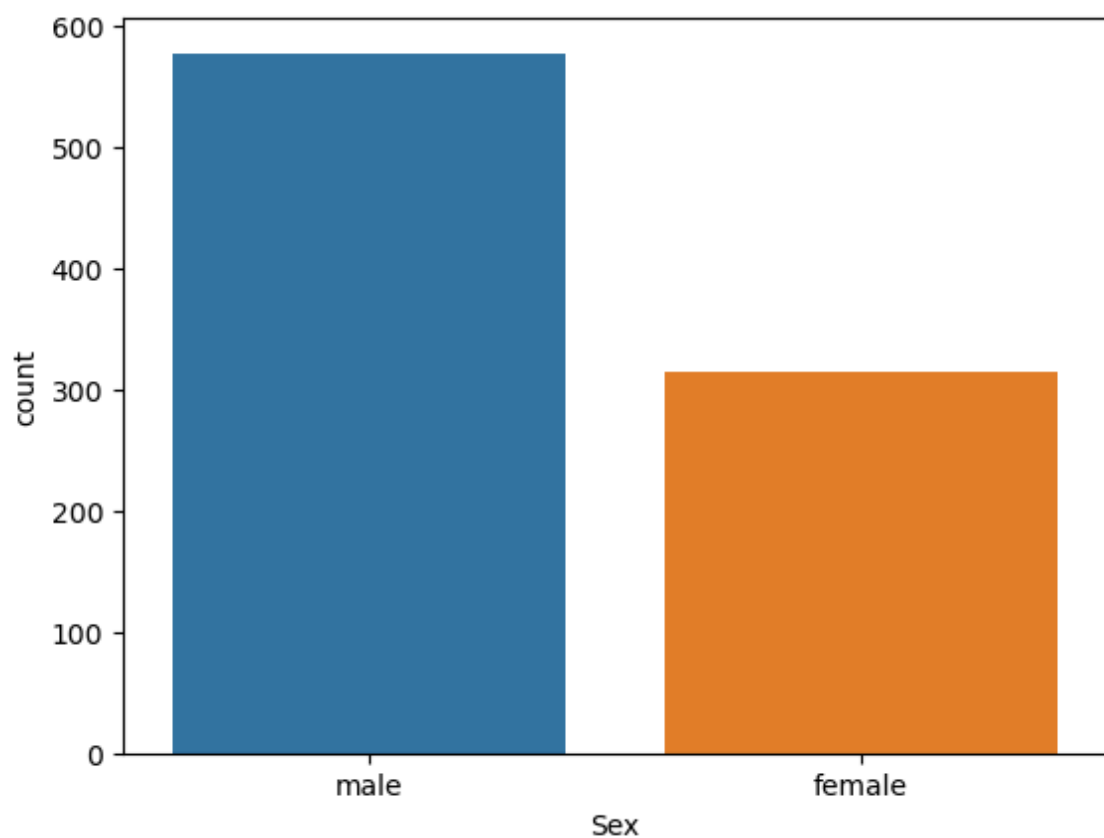
```
In [7]: sns.countplot(train['Pclass'])
```

```
Out[7]: <AxesSubplot:xlabel='Pclass', ylabel='count'>
```



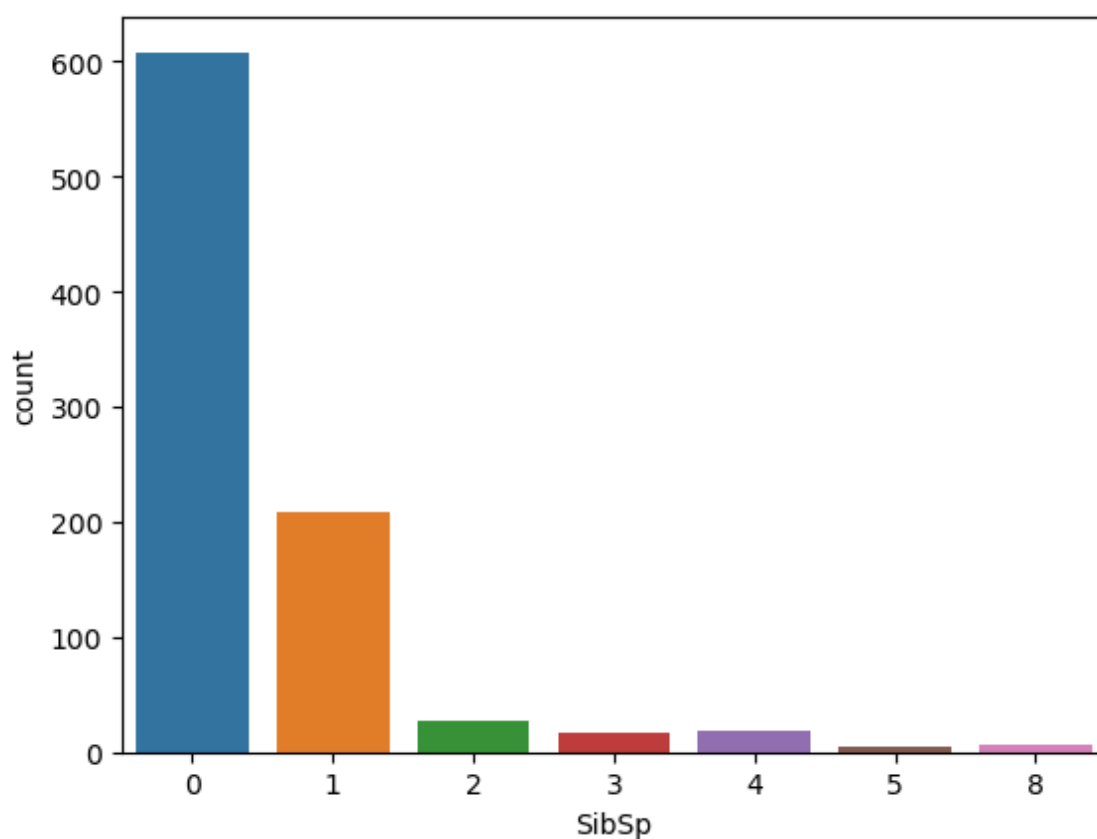
```
In [8]: sns.countplot(train['Sex'])
```

```
Out[8]: <AxesSubplot:xlabel='Sex', ylabel='count'>
```



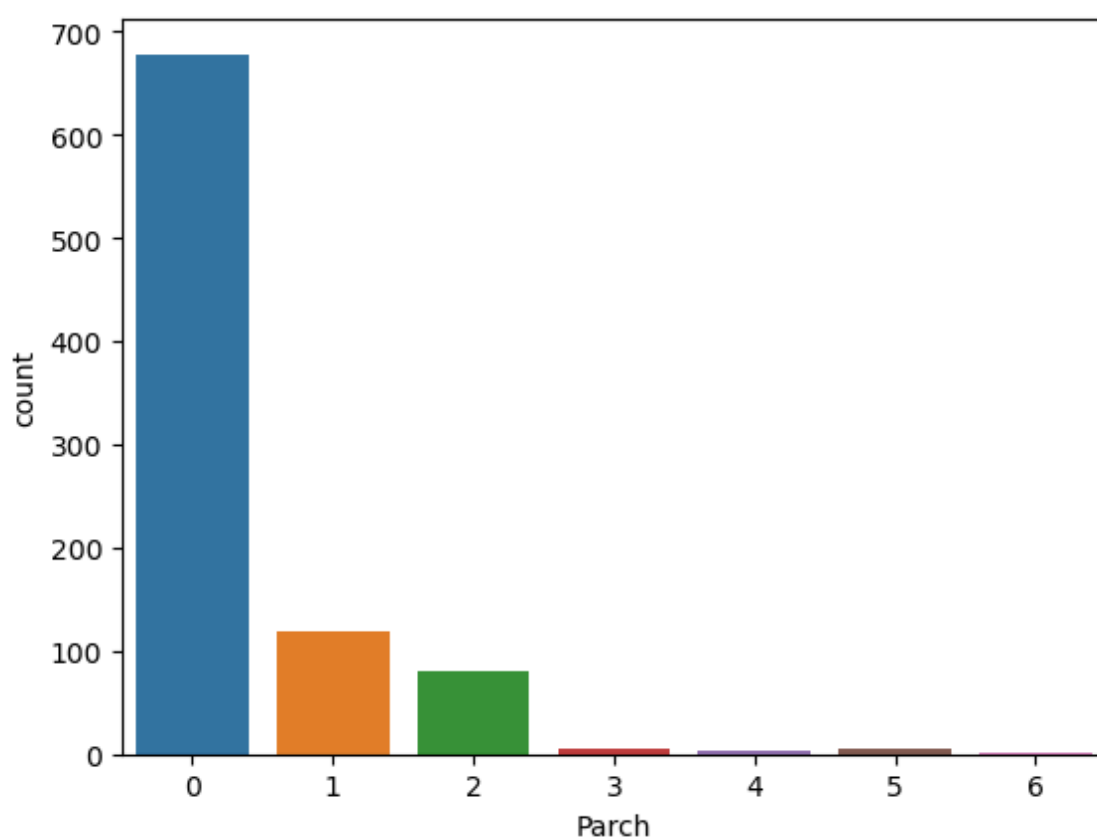
```
In [9]: sns.countplot(train['SibSp'])
```

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



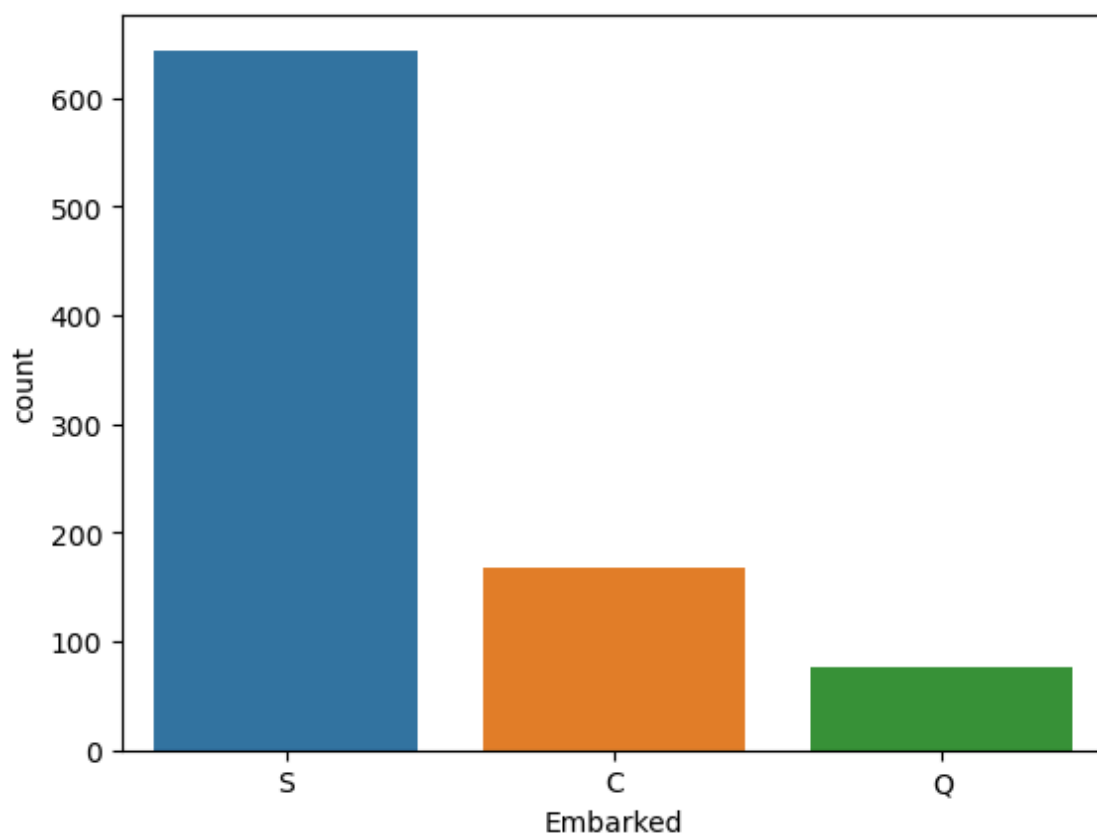
```
In [10]: sns.countplot(train['Parch'])
```

```
Out[10]: <AxesSubplot:xlabel='Parch', ylabel='count'>
```



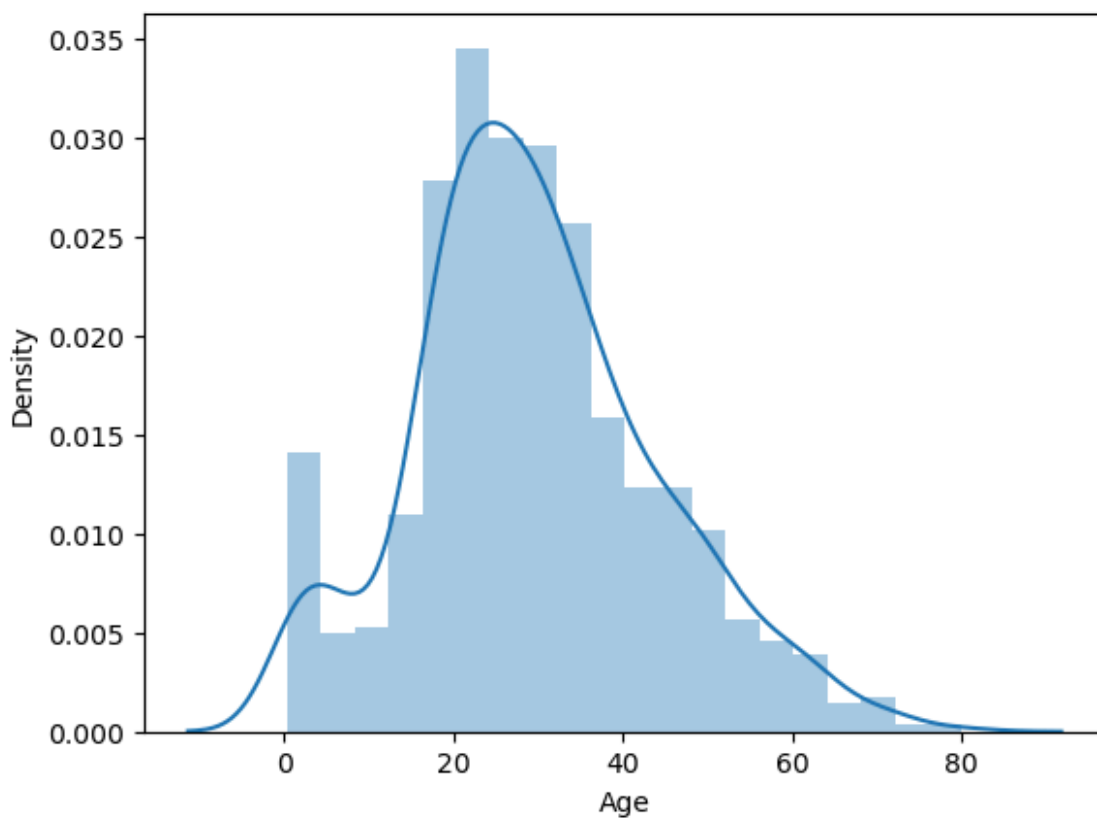
```
In [11]: sns.countplot(train['Embarked'])
```

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



```
In [12]: sns.distplot(train['Age'])
```

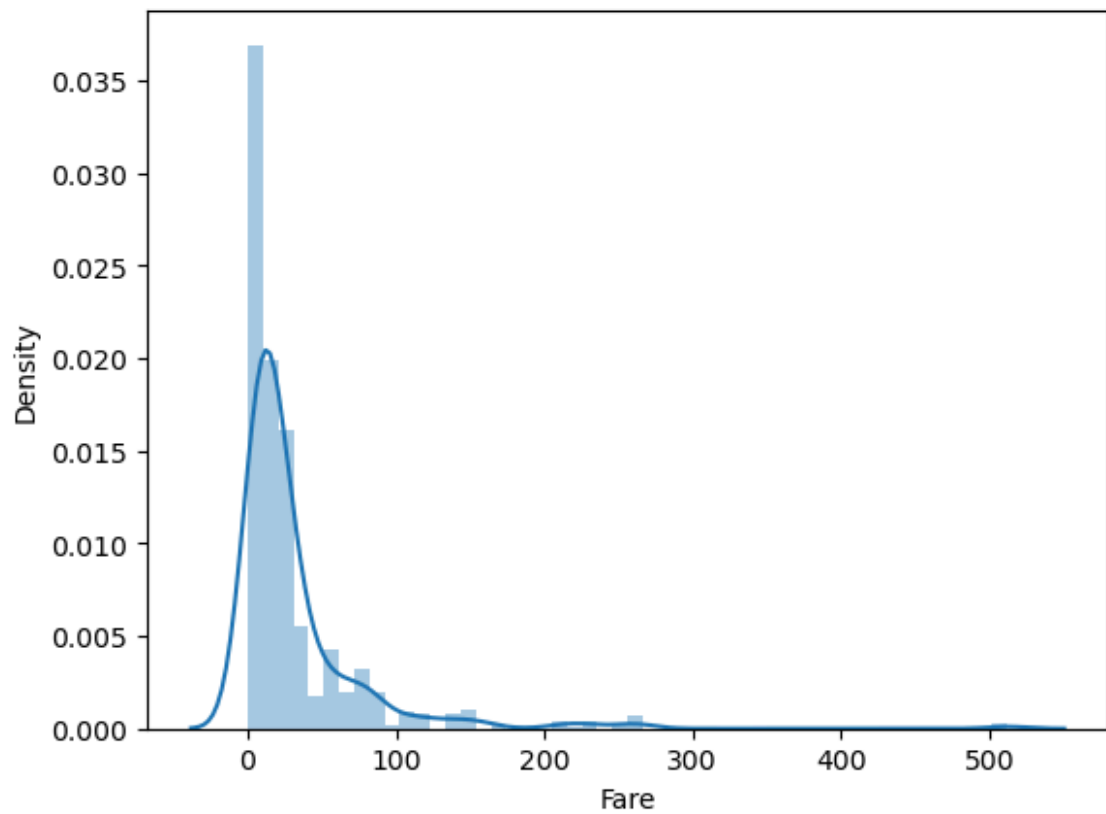
```
Out[12]: <AxesSubplot:xlabel='Age', ylabel='Density'>
```



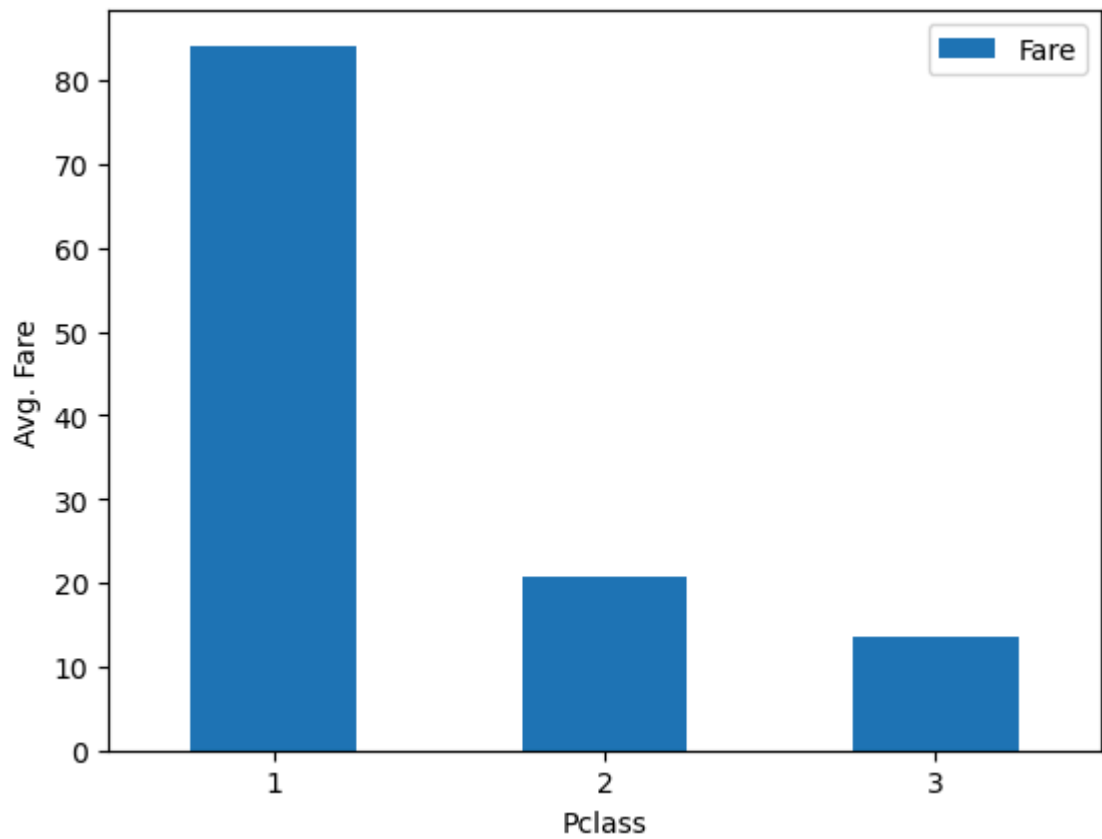
## numerical attributes

```
In [13]: sns.distplot(train['Fare'])
```

```
Out[13]: <AxesSubplot:xlabel='Fare', ylabel='Density'>
```

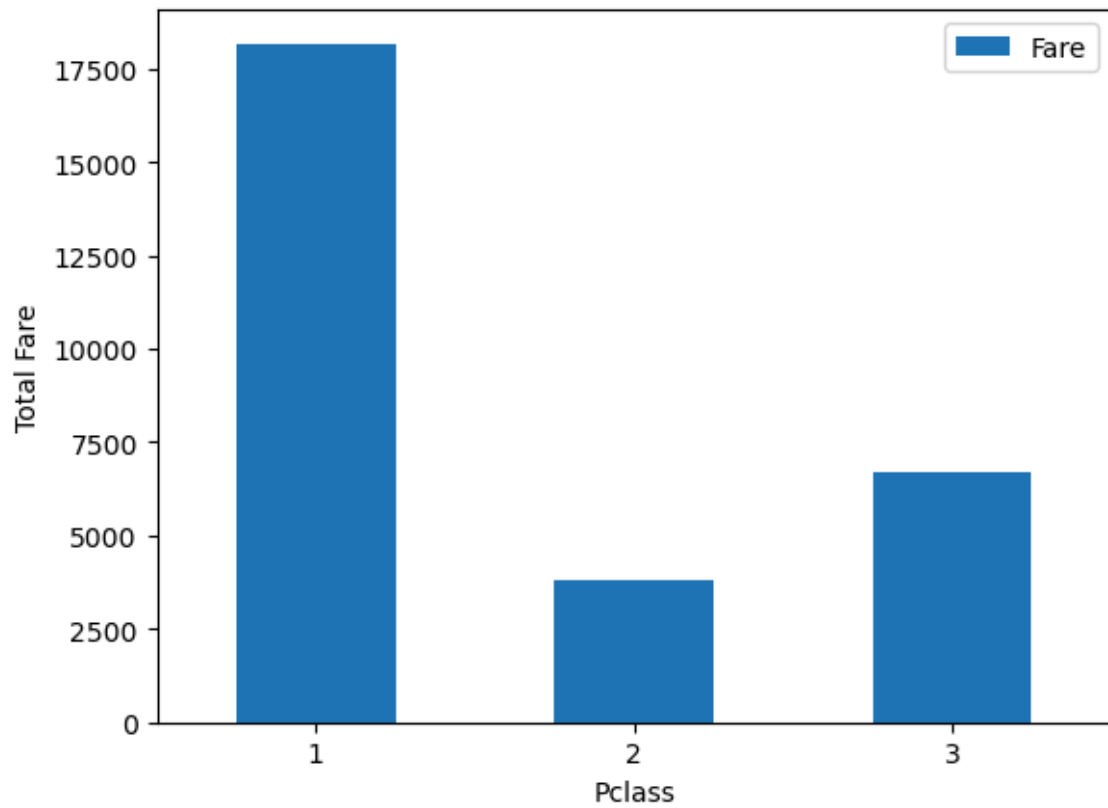


```
In [14]: class_fare = train.pivot_table(index='Pclass', values='Fare')
class_fare.plot(kind='bar')
plt.xlabel('Pclass')
plt.ylabel('Avg. Fare')
plt.xticks(rotation=0)
plt.show()
```



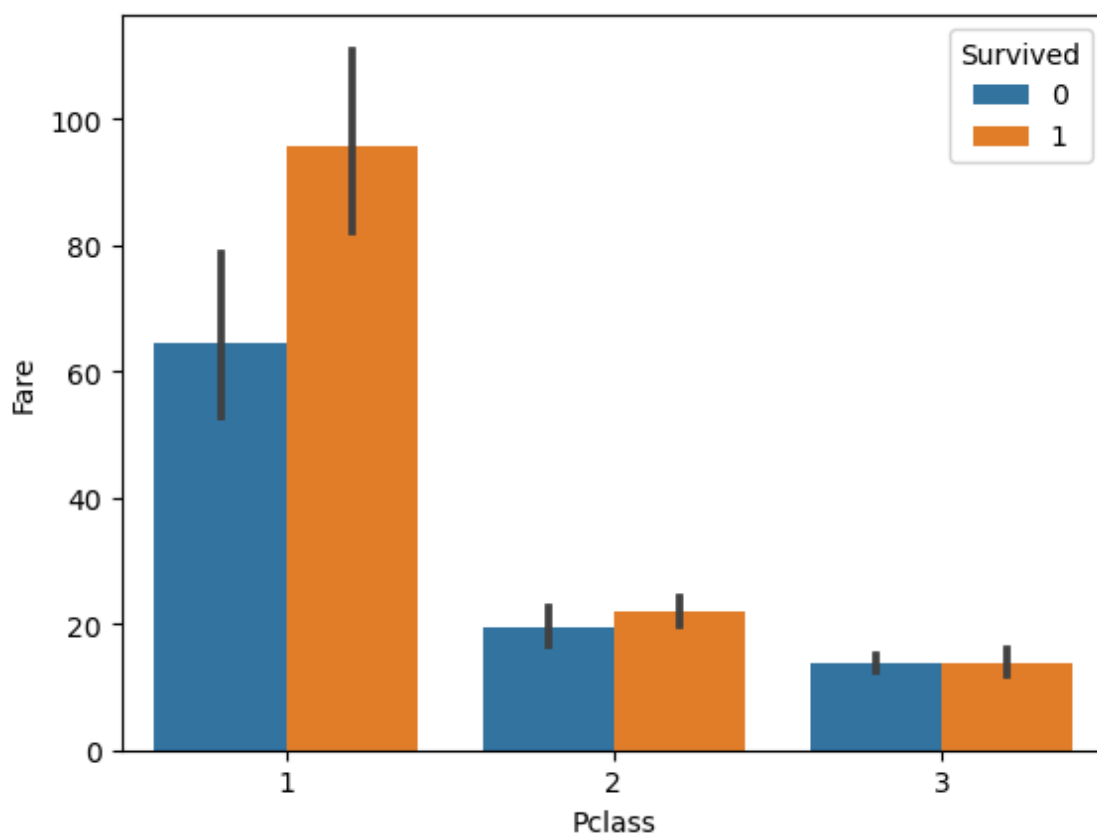


```
In [15]: class_fare = train.pivot_table(index='Pclass', values='Fare', aggfunc=np.sum)
class_fare.plot(kind='bar')
plt.xlabel('Pclass')
plt.ylabel('Total Fare')
plt.xticks(rotation=0)
plt.show()
```



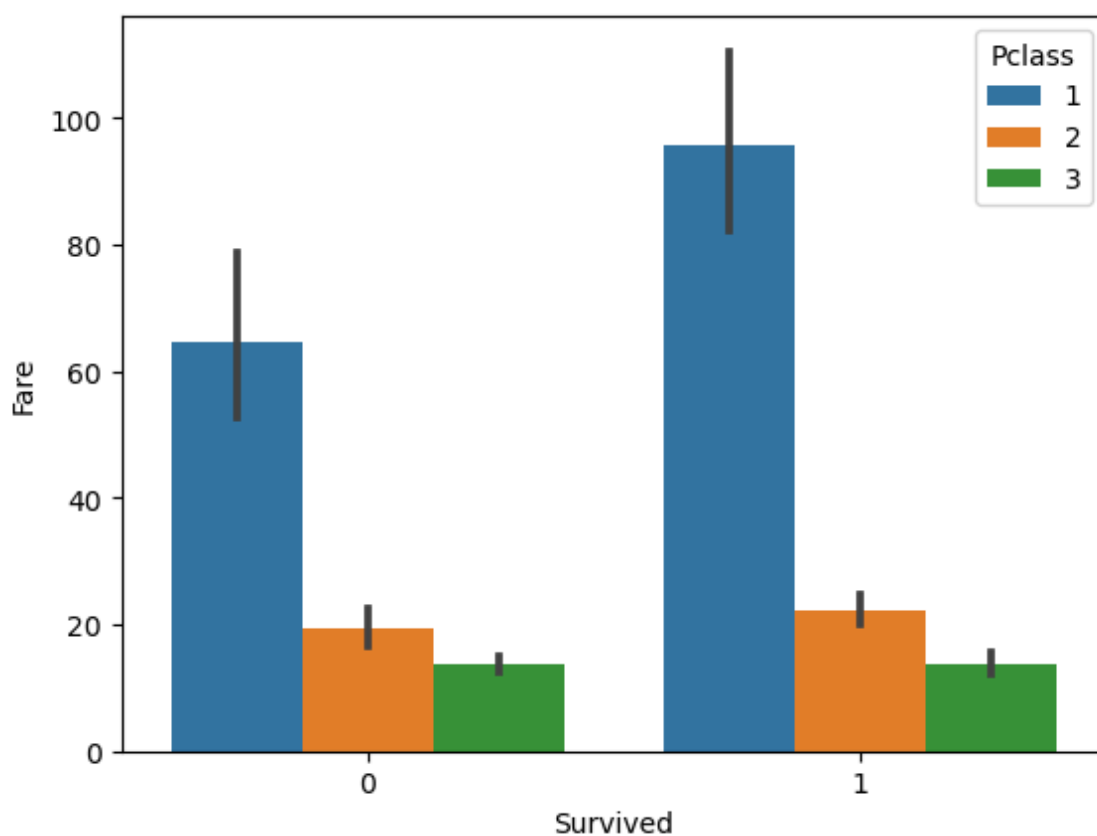
```
In [16]: sns.barplot(data=train, x='Pclass', y='Fare', hue='Survived')
```

```
Out[16]: <AxesSubplot:xlabel='Pclass', ylabel='Fare'>
```



```
In [17]: sns.barplot(data=train, x='Survived', y='Fare', hue='Pclass')
```

```
Out[17]: <AxesSubplot:xlabel='Survived', ylabel='Fare'>
```



## Data Preprocessing

```
In [18]: train_len = len(train)
df = pd.concat([train, test], axis=0)
df = df.reset_index(drop=True)
df.head()
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

## combine two dataframes

In [19]: `df.tail()`

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1304	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236
1305	1306	NaN	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758 10
1306	1307	NaN	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262
1307	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309
1308	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668 2

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

Out[20]:

```

PassengerId      0
Survived          418
Pclass            0
Name              0
Sex               0
Age              263
SibSp             0
Parch             0
Ticket            0
Fare              1
Cabin           1014
Embarked          2
dtype: int64

```

## find the null values

In [21]: `df = df.drop(columns=['Cabin'], axis=1)`

## drop or delete the column

In [22]: `df['Age'].mean()`

Out[22]: 29.881137667304014

```
In [23]: df['Age'] = df['Age'].fillna(df['Age'].mean())  
df['Fare'] = df['Fare'].fillna(df['Fare'].mean())
```

## fill missing values using mean of the numerical column

```
In [24]: df['Embarked'].mode()[0]
```

```
Out[24]: 'S'
```

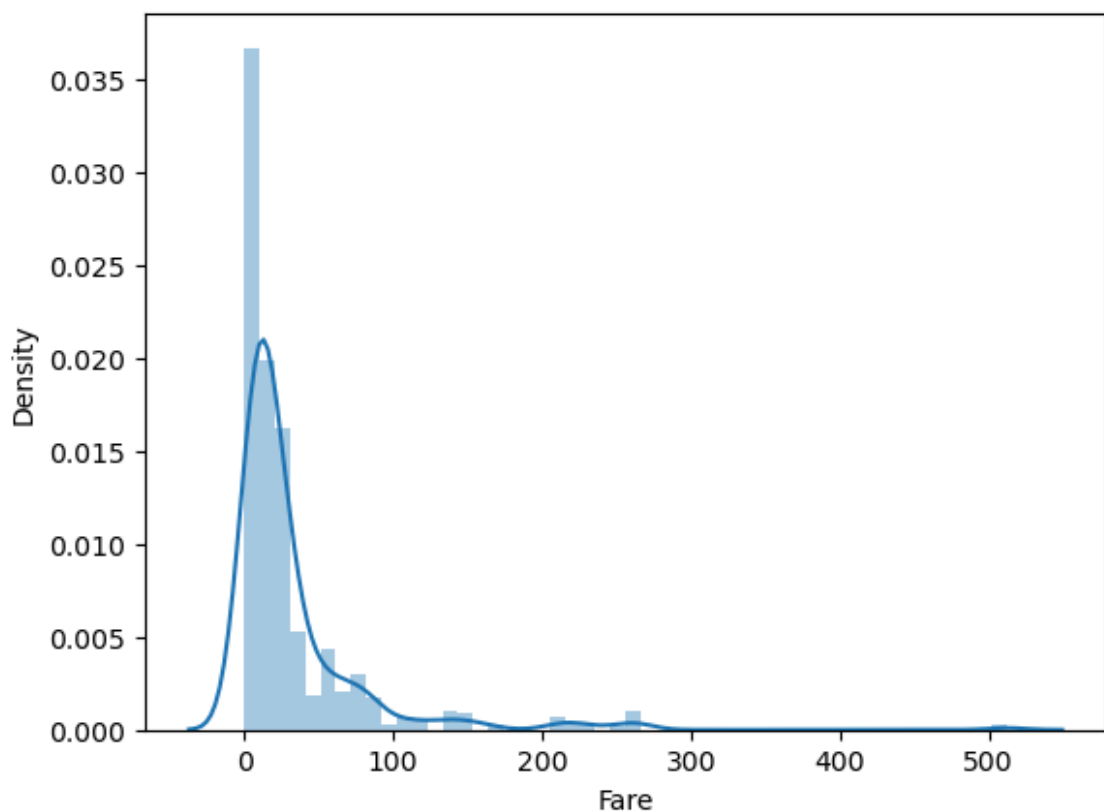
```
In [25]: df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])
```

## fill missing values using mode of the categorical column

### Log transformation for uniform data distribution

```
In [26]: sns.distplot(df['Fare'])
```

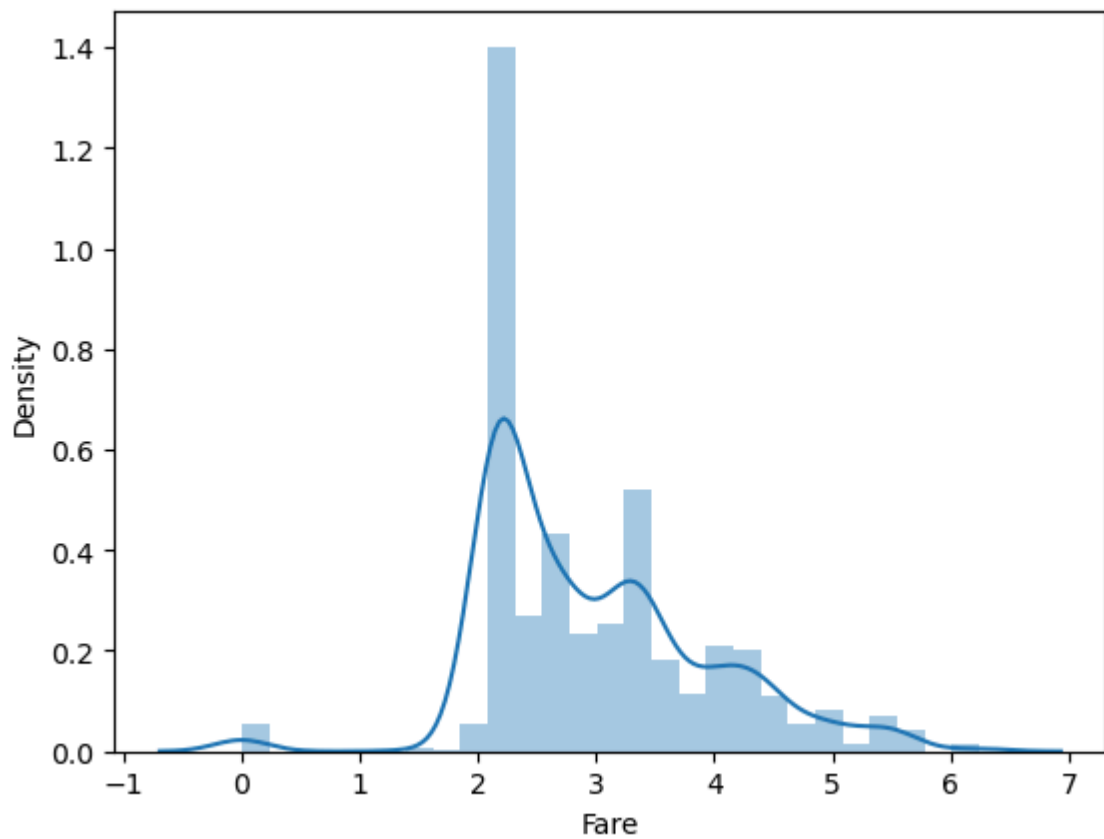
```
Out[26]: <AxesSubplot:xlabel='Fare', ylabel='Density'>
```



```
In [27]: df['Fare'] = np.log(df['Fare']+1)
```

```
In [28]: sns.distplot(df['Fare'])
```

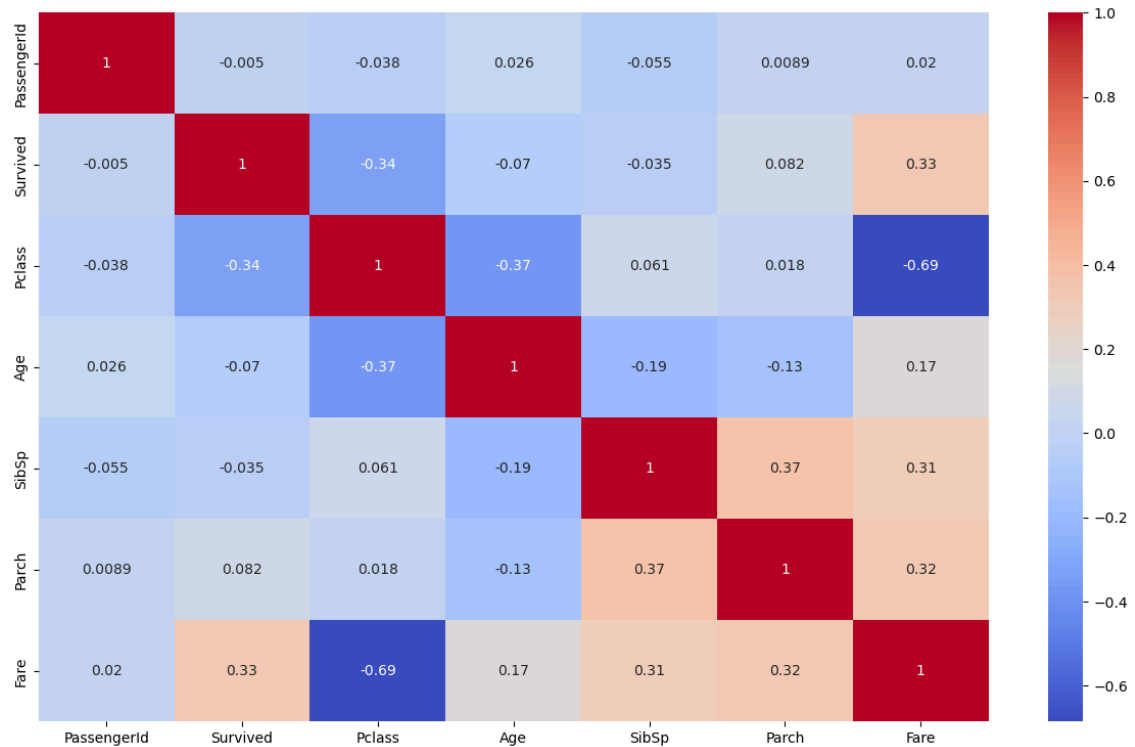
```
Out[28]: <AxesSubplot:xlabel='Fare', ylabel='Density'>
```



## Correlation Matrix

```
In [29]: corr = df.corr()
plt.figure(figsize=(15, 9))
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

Out[29]: <AxesSubplot:>



```
In [30]: df.head()
```

Out[30]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0.0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	2.11021
1	2	1.0	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	4.28059
2	3	1.0	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	2.18885
3	4	1.0	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	3.99083
4	5	0.0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	2.20276

```
In [31]: df = df.drop(columns=['Name', 'Ticket'], axis=1)
df.head()
```

```
Out[31]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0.0	3	male	22.0	1	0	2.110213	S
1	2	1.0	1	female	38.0	1	0	4.280593	C
2	3	1.0	3	female	26.0	0	0	2.188856	S
3	4	1.0	1	female	35.0	1	0	3.990834	S
4	5	0.0	3	male	35.0	0	0	2.202765	S

## drop unnecessary columns

## Label Encoding

```
In [32]: from sklearn.preprocessing import LabelEncoder
cols = ['Sex', 'Embarked']
le = LabelEncoder()

for col in cols:
    df[col] = le.fit_transform(df[col])
df.head()
```

```
Out[32]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0.0	3	1	22.0	1	0	2.110213	2
1	2	1.0	1	0	38.0	1	0	4.280593	0
2	3	1.0	3	0	26.0	0	0	2.188856	2
3	4	1.0	1	0	35.0	1	0	3.990834	2
4	5	0.0	3	1	35.0	0	0	2.202765	2

## Train-Test Split

```
In [33]: train = df.iloc[:train_len, :]
test = df.iloc[train_len:, :]
```

```
In [34]: train.head()
```

```
Out[34]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0.0	3	1	22.0	1	0	2.110213	2
1	2	1.0	1	0	38.0	1	0	4.280593	0
2	3	1.0	3	0	26.0	0	0	2.188856	2
3	4	1.0	1	0	35.0	1	0	3.990834	2
4	5	0.0	3	1	35.0	0	0	2.202765	2



In [35]: `test.head()`

Out[35]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
891	892	NaN	3	1	34.5	0	0	2.178064	1
892	893	NaN	3	0	47.0	1	0	2.079442	2
893	894	NaN	2	1	62.0	0	0	2.369075	1
894	895	NaN	3	1	27.0	0	0	2.268252	2
895	896	NaN	3	0	22.0	1	1	2.586824	2

In [36]: `X = train.drop(columns=['PassengerId', 'Survived'], axis=1)`  
`y = train['Survived']`

## input split

In [37]: `X.head()`

Out[37]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	2.110213	2
1	1	0	38.0	1	0	4.280593	0
2	3	0	26.0	0	0	2.188856	2
3	1	0	35.0	1	0	3.990834	2
4	3	1	35.0	0	0	2.202765	2

## Model Training

In [38]: `from sklearn.model_selection import train_test_split, cross_val_score`  
`def classify(model):`  
 `x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.1)`  
 `model.fit(x_train, y_train)`  
 `print('Accuracy:', model.score(x_test, y_test))`  
  
`score = cross_val_score(model, X, y, cv=5)`  
`print('CV Score:', np.mean(score))`

## classify column

In [39]: `from sklearn.linear_model import LogisticRegression`  
`model = LogisticRegression()`  
`classify(model)`

Accuracy: 0.8071748878923767  
 CV Score: 0.7833971502102819

```
In [40]: from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
classify(model)
```

Accuracy: 0.7309417040358744  
CV Score: 0.7710752620676667

```
In [41]: from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
classify(model)
```

Accuracy: 0.8071748878923767  
CV Score: 0.8137342288619672

```
In [42]: from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier()
classify(model)
```

Accuracy: 0.8116591928251121  
CV Score: 0.7935157868307073

```
In [43]: !pip install xgboost
```

Defaulting to user installation because normal site-packages is not write able  
Requirement already satisfied: xgboost in c:\users\kodad\appdata\roaming\python\python39\site-packages (2.1.3)  
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (from xgboost) (1.21.5)  
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (from xgboost) (1.9.1)

```
In [44]: from xgboost import XGBClassifier
model = XGBClassifier()
classify(model)
```

Accuracy: 0.7847533632286996  
CV Score: 0.8148327160881301

```
In [45]: !pip install lightgbm
```

Defaulting to user installation because normal site-packages is not write able  
Requirement already satisfied: lightgbm in c:\users\kodad\appdata\roaming\python\python39\site-packages (4.5.0)  
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages (from lightgbm) (1.9.1)  
Requirement already satisfied: numpy>=1.17.0 in c:\programdata\anaconda3\lib\site-packages (from lightgbm) (1.21.5)

```
In [46]: from lightgbm import LGBMClassifier  
model = LGBMClassifier()  
classify(model)
```

And if memory is not enough, you can set `force\_col\_wise=true`.

[LightGBM] [Info] Total Bins 204

[LightGBM] [Info] Number of data points in the train set: 713, number of used features: 7

[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.384292 -> initscore=-0.471371

[LightGBM] [Info] Start training from score -0.471371

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

[LightGBM] [Warning] No further splits with positive gain, best gain: -inf

```
In [47]: !pip install catboost
```

```
Defaulting to user installation because normal site-packages is not write
able
Requirement already satisfied: catboost in c:\users\kodad\appdata\roaming
\python\python39\site-packages (1.2.7)
Requirement already satisfied: graphviz in c:\users\kodad\appdata\roaming
\python\python39\site-packages (from catboost) (0.20.3)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-p
ackages (from catboost) (1.16.0)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site
-packages (from catboost) (1.9.1)
Requirement already satisfied: pandas>=0.24 in c:\programdata\anaconda3\l
ib\site-packages (from catboost) (1.4.4)
Requirement already satisfied: numpy<2.0,>=1.16.0 in c:\programdata\anaco
nda3\lib\site-packages (from catboost) (1.21.5)
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib
\site-packages (from catboost) (3.5.2)
Requirement already satisfied: plotly in c:\programdata\anaconda3\lib\sit
e-packages (from catboost) (5.9.0)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\l
ib\site-packages (from pandas>=0.24->catboost) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\programdata\anaconda3\lib\site-packages (from pandas>=0.24->catboost) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\programdata\anaconda3
\lib\site-packages (from matplotlib->catboost) (9.2.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anacon
da3\lib\site-packages (from matplotlib->catboost) (4.25.0)
Requirement already satisfied: packaging>=20.0 in c:\programdata\anaconda
3\lib\site-packages (from matplotlib->catboost) (21.3)
Requirement already satisfied: cyclers>=0.10 in c:\programdata\anaconda3\l
ib\site-packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\programdata\anacon
da3\lib\site-packages (from matplotlib->catboost) (1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\programdata\anacond
a3\lib\site-packages (from matplotlib->catboost) (3.0.9)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda
3\lib\site-packages (from plotly->catboost) (8.0.1)
```

```
In [48]: from catboost import CatBoostClassifier
model = CatBoostClassifier(verbose=0)
classify(model)
```

```
Accuracy: 0.8295964125560538
CV Score: 0.8226790534178645
```

## Complete Model Training with Full Data

```
In [49]: model = LGBMClassifier()
model.fit(X, y)
```

```
[LightGBM] [Info] Number of positive: 342, number of negative: 549
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000531 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 222
[LightGBM] [Info] Number of data points in the train set: 891, number of
used features: 7
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.383838 -> initscore=-0.
473288
[LightGBM] [Info] Start training from score -0.473288
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
Out[49]: LGBMClassifier()
```

```
In [50]: test.head()
```

```
Out[50]:
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
891	892	NaN	3	1	34.5	0	0	2.178064	1
892	893	NaN	3	0	47.0	1	0	2.079442	2
893	894	NaN	2	1	62.0	0	0	2.369075	1
894	895	NaN	3	1	27.0	0	0	2.268252	2
895	896	NaN	3	0	22.0	1	1	2.586824	2

```
In [51]: X_test = test.drop(columns=['PassengerId', 'Survived'], axis=1)
```

## input split for test data

```
In [52]: X_test.head()
```

```
Out[52]:
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
891	3	1	34.5	0	0	2.178064	1
892	3	0	47.0	1	0	2.079442	2
893	2	1	62.0	0	0	2.369075	1
894	3	1	27.0	0	0	2.268252	2
895	3	0	22.0	1	1	2.586824	2

```
In [53]: pred = model.predict(X_test)
pred
```

```
Out[53]: array([0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 1.,
0.,
        1., 1., 1., 0., 0., 1., 1., 1., 0., 1., 1., 1., 0., 0., 0., 0.,
0.,
        1., 0., 0., 0., 0., 1., 0., 1., 0., 1., 1., 0., 0., 0., 1., 1.,
1.,
        0., 1., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 1., 1., 1.,
0.,
        0., 1., 1., 0., 0., 0., 1., 1., 0., 1., 0., 1., 1., 0., 0., 0.,
0.,
        0., 1., 1., 1., 1., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0., 1.,
0.,
        0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 0., 0., 1.,
1.,
        1., 1., 0., 1., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
0.,
        0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0.,
0.,
        0., 0., 0., 1., 0., 1., 1., 1., 1., 1., 0., 0., 0., 0., 0., 1.,
0.,
        0., 1., 0., 0., 0., 1., 1., 0., 1., 1., 0., 0., 1., 0., 1., 0.,
1.,
        0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 1., 1., 0.,
1.,
        0., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 1., 0.,
1.,
        0., 1., 0., 1., 0., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 0.,
1.,
        1., 1., 1., 1., 0., 0., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0.,
0.,
        0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0.,
0.,
        1., 1., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0., 1., 0., 0., 0.,
0.,
        0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1.,
1.,
        0., 1., 0., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0.,
0.,
        1., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0.,
0.,
        0., 0., 0., 1., 0., 1., 0., 1., 0., 1., 1., 0., 0., 0., 1., 0.,
1.,
        0., 0., 1., 0., 1., 1., 1., 1., 0., 0., 0., 1., 0., 0., 1., 0.,
0.,
        1., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
0.,
        1., 0., 0., 0., 1., 0., 1., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
0.,
        1., 1., 1., 1., 0., 0., 1., 0., 0., 1.] )
```

## Test Submission

```
In [54]: sub = pd.read_csv('gender_submission.csv')
sub.head()
```

```
Out[54]:
```

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

```
In [55]: sub.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     418 non-null   int64
1   Survived        418 non-null   int64
dtypes: int64(2)
memory usage: 6.7 KB
```

```
In [56]: sub['Survived'] = pred
sub['Survived'] = sub['Survived'].astype('int')
```

```
In [57]: sub.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId     418 non-null   int64
1   Survived        418 non-null   int32
dtypes: int32(1), int64(1)
memory usage: 5.0 KB
```

```
In [58]: sub.head()
```

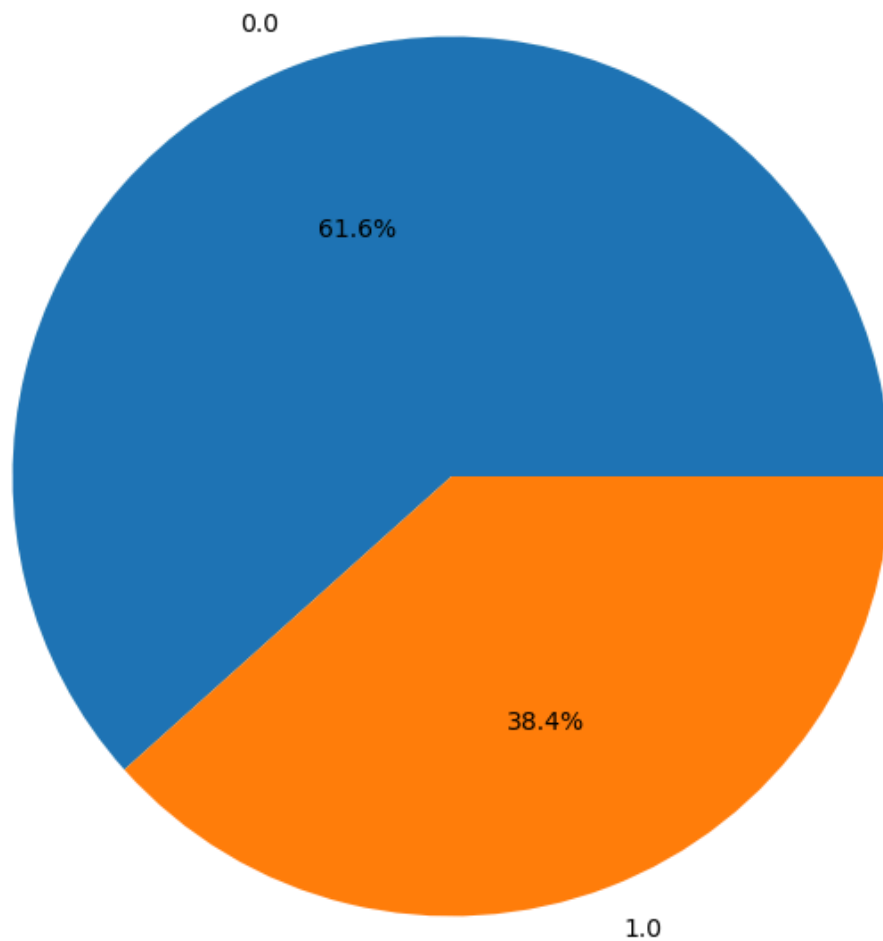
```
Out[58]:
```

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	1
4	896	0

```
In [59]: sub.to_csv('submission.csv', index=False)
```

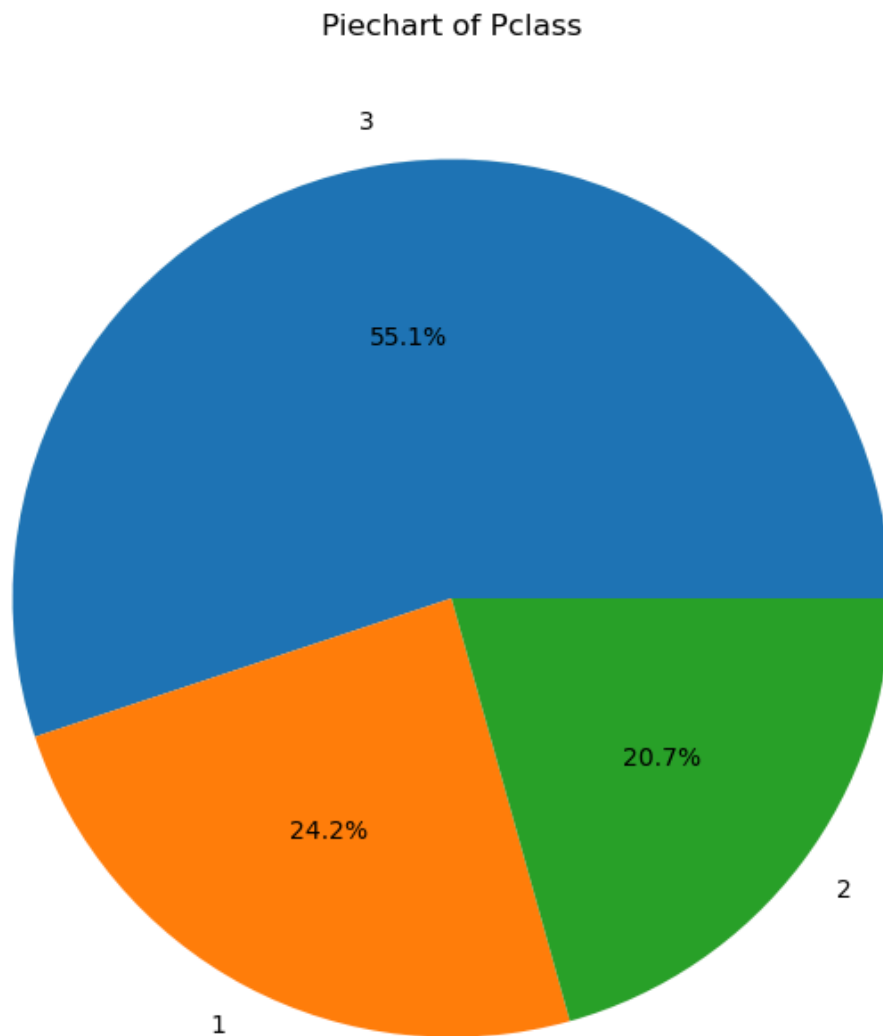
```
In [60]: df=train.Survived.value_counts()  
df.plot(kind='pie', figsize=(8, 8), autopct='%1.1f%%')  
plt.title("Piechart of Survived")  
plt.ylabel("")  
plt.show()
```

Piechart of Survived





```
In [61]: train=train.Pclass.value_counts()  
train.plot(kind='pie', figsize=(8, 8), autopct='%1.1f%%')  
plt.title("Piechart of Pclass")  
plt.ylabel("")  
plt.show()
```



In [ ]: