**Name : Mohit Manish Bhavsar**

**Roll No : 20U437**

**Div : 4**

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import os

test = pd.read\_csv("test.csv")  
train = pd.read\_csv("train.csv")

train.info()

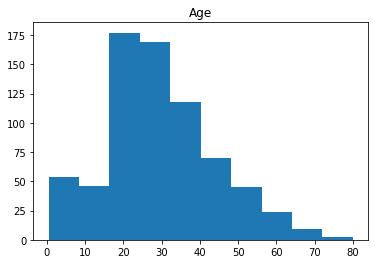
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 891 entries, 0 to 890  
Data columns (total 12 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 PassengerId 891 non-null int64   
 1 Survived 891 non-null int64   
 2 Pclass 891 non-null int64   
 3 Name 891 non-null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

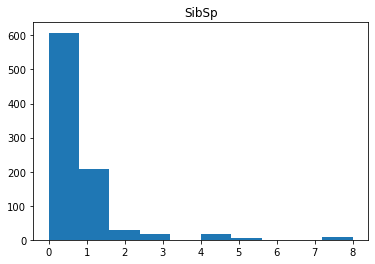
train.describe()

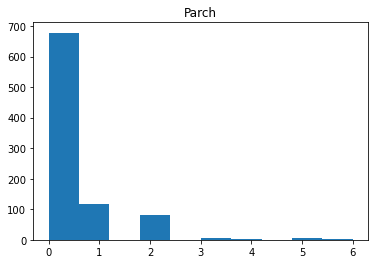
PassengerId Survived Pclass Age SibSp \  
count 891.000000 891.000000 891.000000 714.000000 891.000000   
mean 446.000000 0.383838 2.308642 29.699118 0.523008   
std 257.353842 0.486592 0.836071 14.526497 1.102743   
min 1.000000 0.000000 1.000000 0.420000 0.000000   
25% 223.500000 0.000000 2.000000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

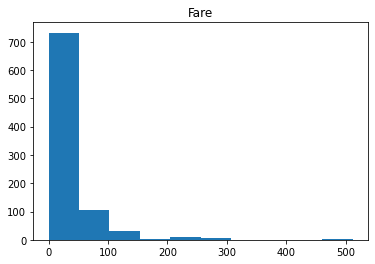
# look at numerical and categorical values seperately  
df\_num = train[['Age','SibSp','Parch','Fare']]  
df\_cat = train[['Survived','Pclass','Sex','Ticket','Cabin','Embarked']]

for i in df\_num.columns:  
 plt.hist(df\_num[i])  
 plt.title(i)  
 plt.show()







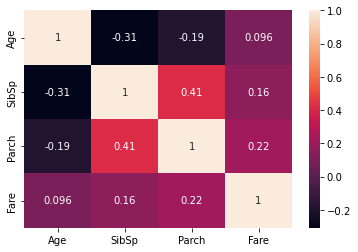


print(df\_num.corr())

Age SibSp Parch Fare  
Age 1.000000 -0.308247 -0.189119 0.096067  
SibSp -0.308247 1.000000 0.414838 0.159651  
Parch -0.189119 0.414838 1.000000 0.216225  
Fare 0.096067 0.159651 0.216225 1.000000

sns.heatmap(df\_num.corr(),annot=True)

<AxesSubplot:>

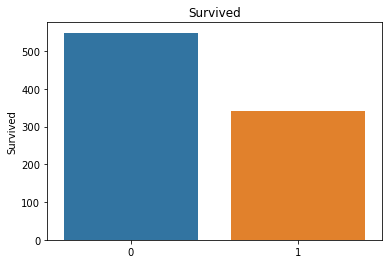


pd.pivot\_table(train,index='Survived', values=['Age','SibSp','Parch','Fare'])

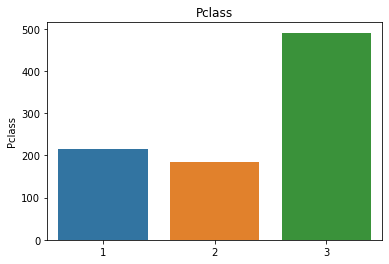
Age Fare Parch SibSp  
Survived   
0 30.626179 22.117887 0.329690 0.553734  
1 28.343690 48.395408 0.464912 0.473684

for i in df\_cat.columns:  
 sns.barplot(df\_cat[i].value\_counts().keys(),df\_cat[i].value\_counts()).set\_title(i)  
 plt.title(i)  
 plt.show()

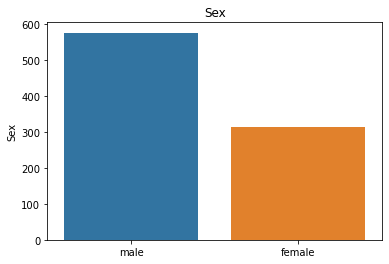
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



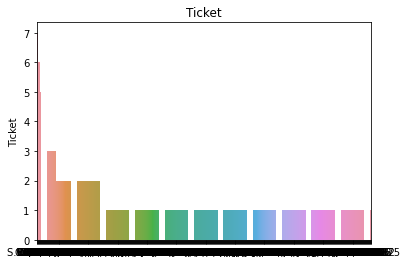
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



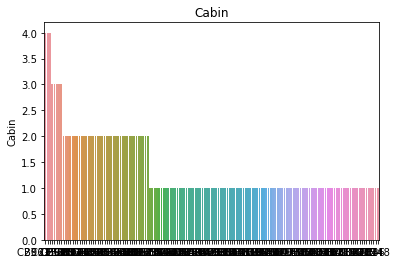
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



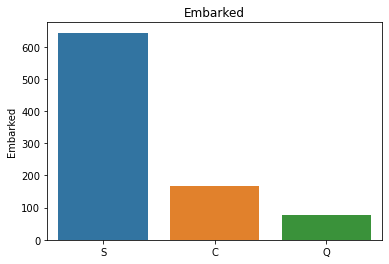
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



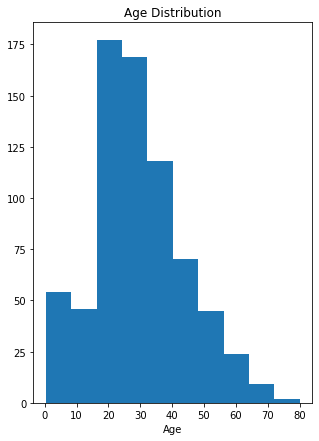
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



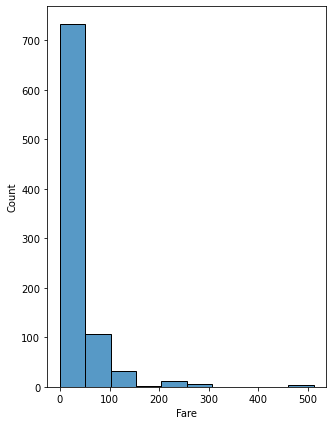
C:\Users\sahil\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
 warnings.warn(



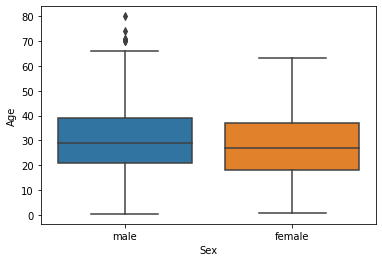
plt.figure(figsize=(5,7))  
plt.hist(train['Age'])  
plt.title("Age Distribution")  
plt.xlabel("Age")  
plt.show()



plt.figure(figsize=(5,7))  
sns.histplot(train['Fare'], bins=10)  
plt.show()



sns.boxplot(x='Sex',y='Age', data=train)  
plt.show()



test1 = pd.DataFrame()

test1 = train[['Sex','Age','Survived']]

test1

Sex Age Survived  
0 male 22.0 0  
1 female 38.0 1  
2 female 26.0 1  
3 female 35.0 1  
4 male 35.0 0  
.. ... ... ...  
886 male 27.0 0  
887 female 19.0 1  
888 female NaN 0  
889 male 26.0 1  
890 male 32.0 0  
  
[891 rows x 3 columns]

plt.figure(figsize=(20,15))  
sns.barplot(x=train['Age'], y=train['Fare'])

<AxesSubplot:xlabel='Age', ylabel='Fare'>

