\*"""Python for Data Science - Perform Data Science on Titanic Dataset

a)Load the Titanic dataset into one of the data structures (NumPy or Pandas).

b)Display header rows and description of the loaded dataset.

c) Remove unnecessary features (E.g. drop unwanted columns) from the dataset.

d) Manipulate data by replacing empty column values with a default value.

e) Perform the following visualizations on the loaded dataset:

i) Passenger status (Survived/Died) against Passenger Class

ii) Survival rate of male vs female

iii) No of passengers in each age group

#numpy - Deals multi-dimensional arrays and matrices

#seaborn - Deals with data visualization

#matplotlib - Plotting; pyplot-interactive plotting

#pandas - data structures and data analysis tools"""

#import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

titanic\_df = pd.read\_csv('train.csv')

# Convert the survived column to strings for easier reading

titanic\_df ['Survived'] = titanic\_df ['Survived'].map({

0: 'Died',

1: 'Survived'

})

print("======Data Headers Before Dropping Columns=======")

print(titanic\_df.head(5))

print("\*\*\*\* \n\nDATA TRANSFORMATION \*\*\*\*\*\n")

print("======Data Headers After Dropping Columns - First Way=======")

titanic\_df.drop(['Parch','PassengerId','Name','Ticket'], axis=1, inplace=True)

#axis =1 in drop method shows you are dropping a column

#inplace=True means you are editing original dataframe

print(titanic\_df.head(5))

print("======Data Headers After Dropping Columns - Second Way =======")

titanic\_df = titanic\_df.drop(['SibSp','Fare'], axis=1)

print(titanic\_df.head(5))

# Convert the Class column to strings for easier reading

titanic\_df ['Pclass'] = titanic\_df ['Pclass'].map({

1: 'Luxury Class',

2: 'Economy Class',

3: 'Lower Class'

})

print("======Data Headers After Transforming Class Column =======")

print(titanic\_df.head(5))

titanic\_df["Embarked"] = titanic\_df["Embarked"].fillna("S")

print("======Data Headers After Filling with default value for Embarked Column =======")

print(titanic\_df.head(5))

# Convert the Embarked column to strings for easier reading

titanic\_df ['Embarked'] = titanic\_df ['Embarked'].map({

'C':'Cherbourg',

'Q':'Queenstown',

'S':'Southampton'

})

print("======Data Headers After Transforming Embarked Column =======")

print(titanic\_df.head(5))

print("\n\n\n\*\*\*\* DATA VISUALIZATIONS\*\*\*\*\n\n")

print("Visualization #1 : Survival Rate Based on Passenger Sitting Class")

ax = sns.countplot(x = 'Pclass', hue = 'Survived', palette = 'Set1',data = titanic\_df)

ax.set(title = 'Passenger status (Survived/Died) against Passenger Class',

xlabel = 'Passenger Class', ylabel = 'Total')

plt.show()

#crosstab - Cross tabulation of two or more factors

print("Visualization #2 : Survival Rate Based on Gender")x

print(pd.crosstab(titanic\_df["Sex"],titanic\_df.Survived))

ax = sns.countplot(x = 'Sex', hue = 'Survived', palette = 'Set2', data = titanic\_df)

ax.set(title = 'Total Survivors According to Sex', xlabel = 'Sex', ylabel='Total')

plt.show()

print("Visualization #3 : Survival Rate Based on Passenger Age Group")

# We look at Age column and set Intevals on the ages and the map them to their categories as

# (Children, Teen, Adult, Old)

interval = (0,18,35,60,120)

categories = ['Children','Teens','Adult', 'Old']

#cut - Segment and sort data values into bins

titanic\_df['Age\_cats'] = pd.cut(titanic\_df.Age, interval, labels = categories)

ax = sns.countplot(x = 'Age\_cats', data = titanic\_df, hue = 'Survived', palette = 'Set3')

ax.set(xlabel='Age Categorical', ylabel='Total',

title="Age Categorical Survival Distribution")

plt.show()

print("Visualization #4 : Survival Rate Based on Passenger Embarked Port")

print(pd.crosstab(titanic\_df['Embarked'], titanic\_df.Survived))

ax = sns.countplot(x = 'Embarked', hue = 'Survived', palette = 'Set1', data = titanic\_df)

ax.set(title = 'Survival distribution according to Embarking place')

plt.show()