

## Problem statement:

**To predict and analyze which gender has a high chance of survival at the time of disaster**

\*Import Datasets,Python packages and libraries

```
In [1]: import numpy as np
import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid",color_codes=True)
import warnings
warnings.simplefilter(action="ignore")
```

```
In [2]: test_df=pd.read_csv(r"C:\Users\Sushma sree\Downloads\test.gender_submission.csv")
test_df
```

Out[2]:

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

418 rows × 11 columns



```
In [3]: train_df=pd.read_csv(r"C:\Users\Sushma sree\Downloads\train.gender_submission.  
train_df
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns



```
In [4]: train_df.head(10)
```

```
Out[4]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	I
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	I
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	I
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	I
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	I
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	I
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	I


```
In [5]: train_df.shape
```

```
Out[5]: (891, 12)
```

```
In [6]: test_df.head(10)
```

Out[6]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
5	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NaN	
6	898	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	
7	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000	NaN	
8	900	3	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292	NaN	
9	901	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500	NaN	



```
In [7]: test_df.shape
```

Out[7]: (418, 11)

```
In [8]: train_df.describe
```

```
Out[8]: <bound method NDFrame.describe of      PassengerId  Survived  Pclass
0                1         0        3  \
1                2         1        1
2                3         1        3
3                4         1        1
4                5         0        3
..            ...         ...         ...
886            887         0         2
887            888         1         1
888            889         0         3
889            890         1         1
890            891         0         3
```

	Name	Sex	Age	SibSp
0	Braund, Mr. Owen Harris	male	22.0	1
\				
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1
2	Heikkinen, Miss. Laina	female	26.0	0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1
4	Allen, Mr. William Henry	male	35.0	0
..	...	...	...	...
886	Montvila, Rev. Juozas	male	27.0	0
887	Graham, Miss. Margaret Edith	female	19.0	0
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1
889	Behr, Mr. Karl Howell	male	26.0	0
890	Dooley, Mr. Patrick	male	32.0	0

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S
..	...	...	...	...	...
886	0	211536	13.0000	NaN	S
887	0	112053	30.0000	B42	S
888	2	W./C. 6607	23.4500	NaN	S
889	0	111369	30.0000	C148	C
890	0	370376	7.7500	NaN	Q

```
[891 rows x 12 columns]>
```

```
In [9]: train_df.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
```

```
In [10]: test_df.describe
```

```
Out[10]: <bound method NDFrame.describe of      PassengerId  Pclass
Name
0           892      3              Kelly, Mr. James \
1           893      3      Wilkes, Mrs. James (Ellen Needs)
2           894      2      Myles, Mr. Thomas Francis
3           895      3              Wirz, Mr. Albert
4           896      3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..          ...      ...
413        1305      3              Spector, Mr. Woolf
414        1306      1      Oliva y Ocana, Dona. Fermina
415        1307      3      Saether, Mr. Simon Sivertsen
416        1308      3      Ware, Mr. Frederick
417        1309      3      Peter, Master. Michael J

      Sex  Age  SibSp  Parch      Ticket    Fare Cabin Embarked
0   male  34.5     0     0    330911     7.8292   NaN      Q
1  female  47.0     1     0    363272     7.0000   NaN      S
2   male  62.0     0     0    240276     9.6875   NaN      Q
3   male  27.0     0     0    315154     8.6625   NaN      S
4  female  22.0     1     1    3101298    12.2875   NaN      S
..     ...   ...     ...     ...     ...     ...     ...     ...
413  male   NaN     0     0      A.5. 3236     8.0500   NaN      S
414  female  39.0     0     0      PC 17758    108.9000  C105      C
415  male  38.5     0     0  SOTON/O.Q. 3101262     7.2500   NaN      S
416  male   NaN     0     0    359309     8.0500   NaN      S
417  male   NaN     1     1      2668     22.3583   NaN      C

[418 rows x 11 columns]>
```

```
In [11]: test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  -
 0   PassengerId   418 non-null   int64  
 1   Pclass        418 non-null   int64  
 2   Name          418 non-null   object  
 3   Sex           418 non-null   object  
 4   Age           332 non-null   float64 
 5   SibSp         418 non-null   int64  
 6   Parch         418 non-null   int64  
 7   Ticket        418 non-null   object  
 8   Fare          417 non-null   float64 
 9   Cabin         91 non-null    object  
10   Embarked      418 non-null   object  
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

## To find Missing values

```
In [12]: train_df.isnull().sum()
```

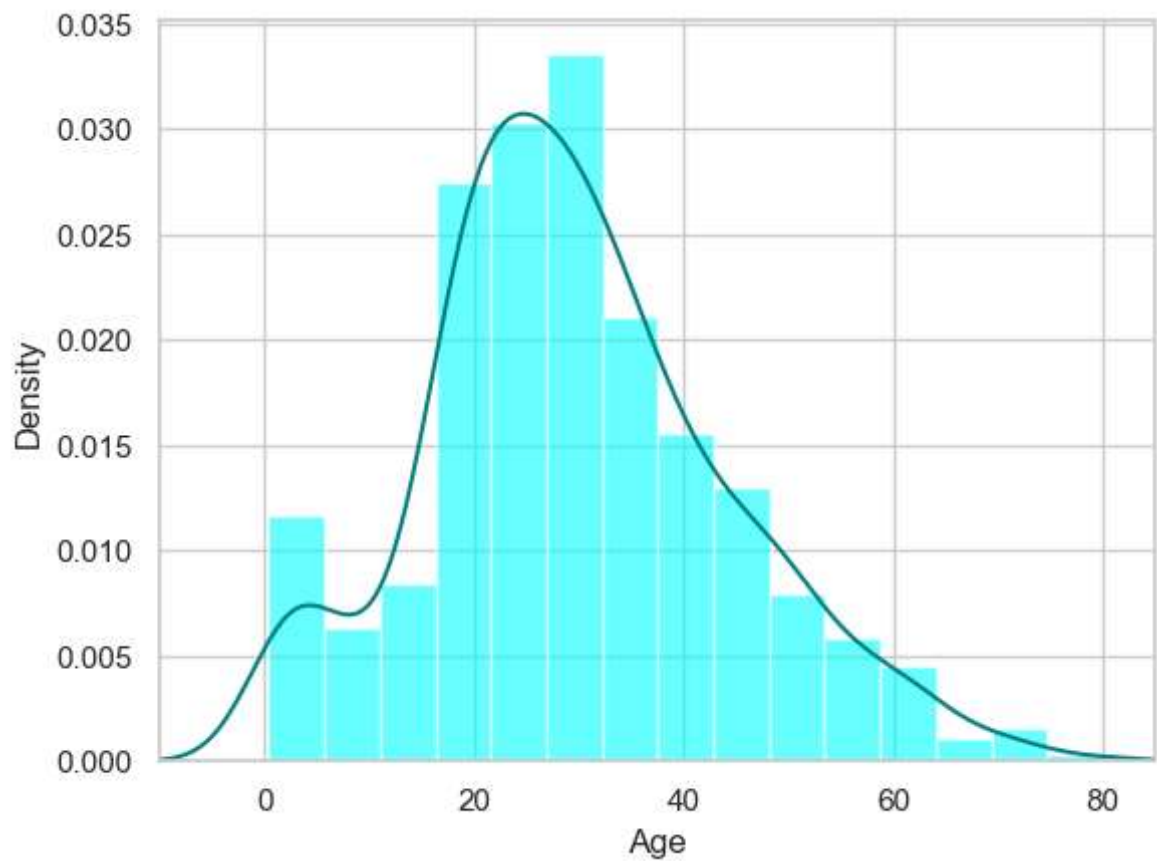
```
Out[12]: PassengerId      0
Survived                0
Pclass                  0
Name                    0
Sex                     0
Age                    177
SibSp                   0
Parch                   0
Ticket                  0
Fare                    0
Cabin                   687
Embarked                2
dtype: int64
```

```
In [13]: test_df.isnull().sum()
```

```
Out[13]: PassengerId      0
Pclass                  0
Name                    0
Sex                     0
Age                     86
SibSp                   0
Parch                   0
Ticket                  0
Fare                    1
Cabin                   327
Embarked                0
dtype: int64
```



```
In [16]: ax=train_df['Age'].hist(bins=15,density=True,stacked=True,color='cyan',alpha=0.5)
train_df['Age'].plot(kind='density',color='teal')
ax.set(xlabel='Age')
plt.xlim(-10,85)
plt.show()
```



```
In [18]: print(train_df['Age'].mean(skipna=True))
print(train_df['Age'].median(skipna=True))
```

```
29.69911764705882
28.0
```

```
In [20]: print((train_df['Cabin'].isnull().sum()/train_df.shape[0])*100)
print((train_df['Embarked'].isnull().sum()/train_df.shape[0])*100)
```

```
77.10437710437711
0.22446689113355783
```

```
In [27]: print('Boarded passengers grouped by port of embarkation(C=Cherbang,Q=Queensto
print(train_df['Embarked'].value_counts())
sns.countplot(x='Embarked',data=train_df,palette='Set2')
plt.show()
```

Boarded passengers grouped by port of embarkation(C=Cherbang,Q=Queenstown,S=Southampton):

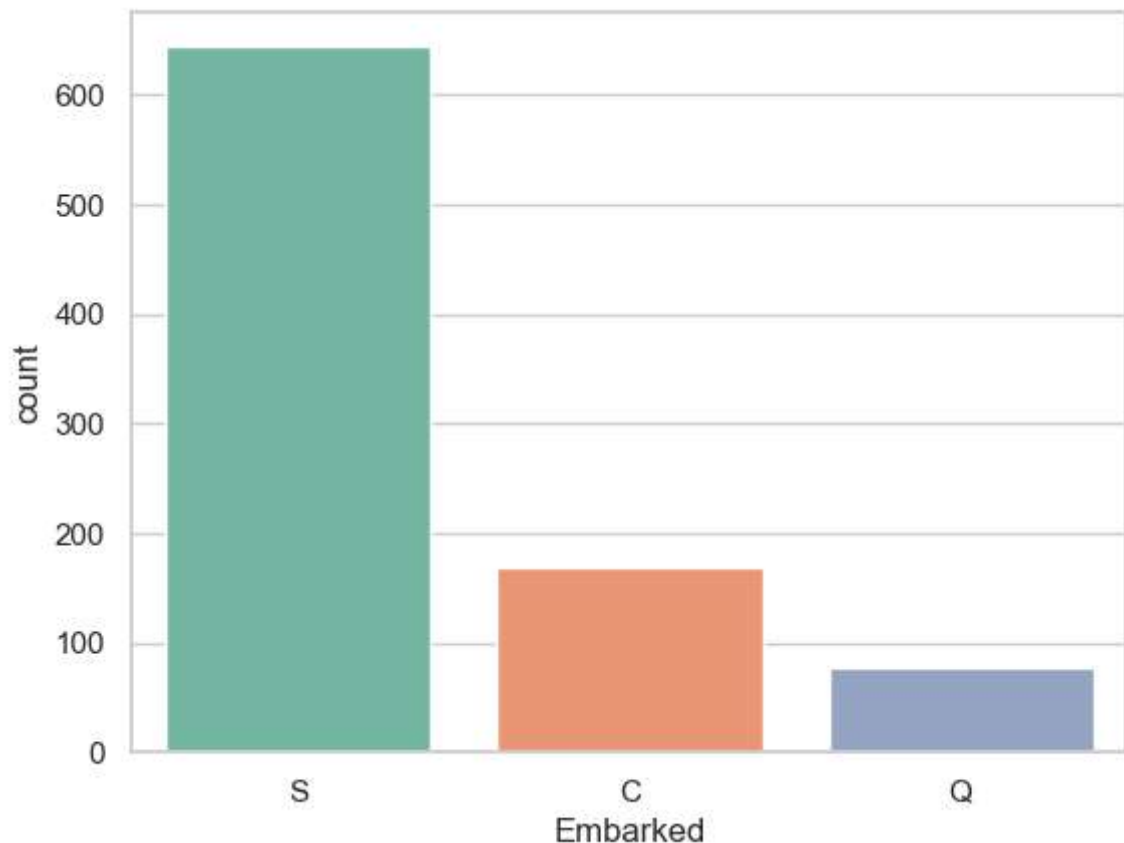
Embarked

S 644

C 168

Q 77

Name: count, dtype: int64



```
In [21]: print(train_df['Embarked'].value_counts().idxmax())
```

S

```
In [23]: train_data=train_df.copy()
train_data['Age'].fillna(train_df['Age'].median(skipna=True),inplace=True)
train_data['Embarked'].fillna(train_df['Embarked'].value_counts().idxmax(),inplace=True)
train_data.drop('Cabin',axis=1,inplace=True)
```

```
In [24]: train_data.isnull().sum()
```

```
Out[24]: PassengerId    0
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Ticket      0
Fare        0
Embarked    0
dtype: int64
```

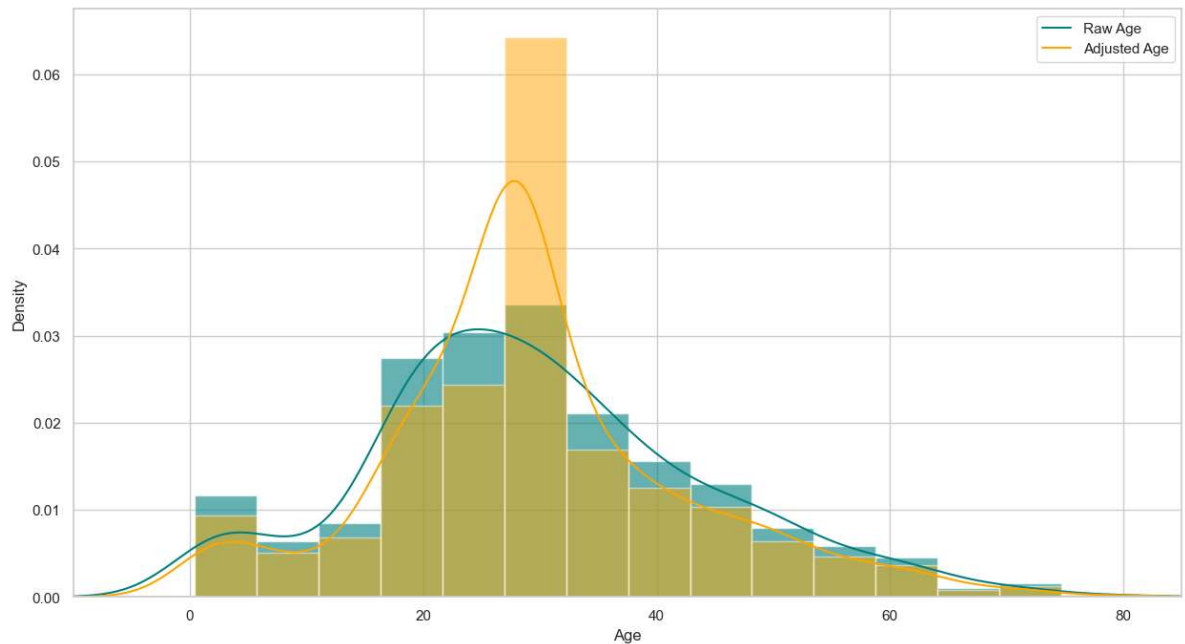
```
In [25]: train_data.head()
```

```
Out[25]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Er
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 [30]: plt.figure(figsize=(15,8))
ax=train_df['Age'].hist(bins=15,density=True,stacked=True,color='teal',alpha=0.5)
train_df['Age'].plot(kind='density',color='teal')
ax=train_data['Age'].hist(bins=15,density=True,stacked=True,color='orange',alpha=0.5)
train_data['Age'].plot(kind='density',color='orange')
ax.legend(['Raw Age','Adjusted Age'])
ax.set(xlabel='Age')
plt.xlim(-10,85)
plt.show()
```



## Create categorical variable for travelling alone

```
In [32]: train_data['TravelAlone']=np.where((train_data["SibSp"]+train_data["Parch"])>0,1,0)
train_data.drop('SibSp',axis=1,inplace=True)
train_data.drop('Parch',axis=1,inplace=True)
```

## Create categorical variables and drop some variables

```
In [34]: training=pd.get_dummies(train_data,columns=["Pclass","Embarked","Sex"])
training.drop('Sex_female',axis=1,inplace=True)
training.drop('PassengerId',axis=1,inplace=True)
training.drop('Name',axis=1,inplace=True)
training.drop('Ticket',axis=1,inplace=True)
final_train=training
final_train.head()
```

```
Out[34]:
```

	Survived	Age	Fare	TravelAlone	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Embarked_
0	0	22.0	7.2500	0	False	False	True	False	Fals
1	1	38.0	71.2833	0	True	False	False	True	Fals
2	1	26.0	7.9250	1	False	False	True	False	Fals
3	1	35.0	53.1000	0	True	False	False	False	Fals
4	0	35.0	8.0500	1	False	False	True	False	Fals

```
In [35]: test_df.isnull().sum()
```

```
Out[35]: PassengerId      0
Pclass      0
Name        0
Sex         0
Age         86
SibSp       0
Parch       0
Ticket      0
Fare        1
Cabin      327
Embarked    0
dtype: int64
```

```

In [39]: test_data=test_df.copy()
test_data['Age'].fillna(train_df['Age'].median(skipna=True),inplace=True)
test_data['Fare'].fillna(train_df['Fare'].median(skipna=True),inplace=True)
test_data.drop('Cabin',axis=1,inplace=True)
test_data['TravelAlnoe']=np.where((test_data['SibSp']+test_data['Parch'])>0,0,0,
test_data.drop('SibSp',axis=1,inplace=True)
test_data.drop('Parch',axis=1,inplace=True)
testing=pd.get_dummies(test_data,columns=['Pclass','Embarked','Sex'])
testing.drop('Sex_female',axis=1,inplace=True)
testing.drop('Name',axis=1,inplace=True)
testing.drop('Ticket',axis=1,inplace=True)
final_test=testing
final_test.head()

```

```

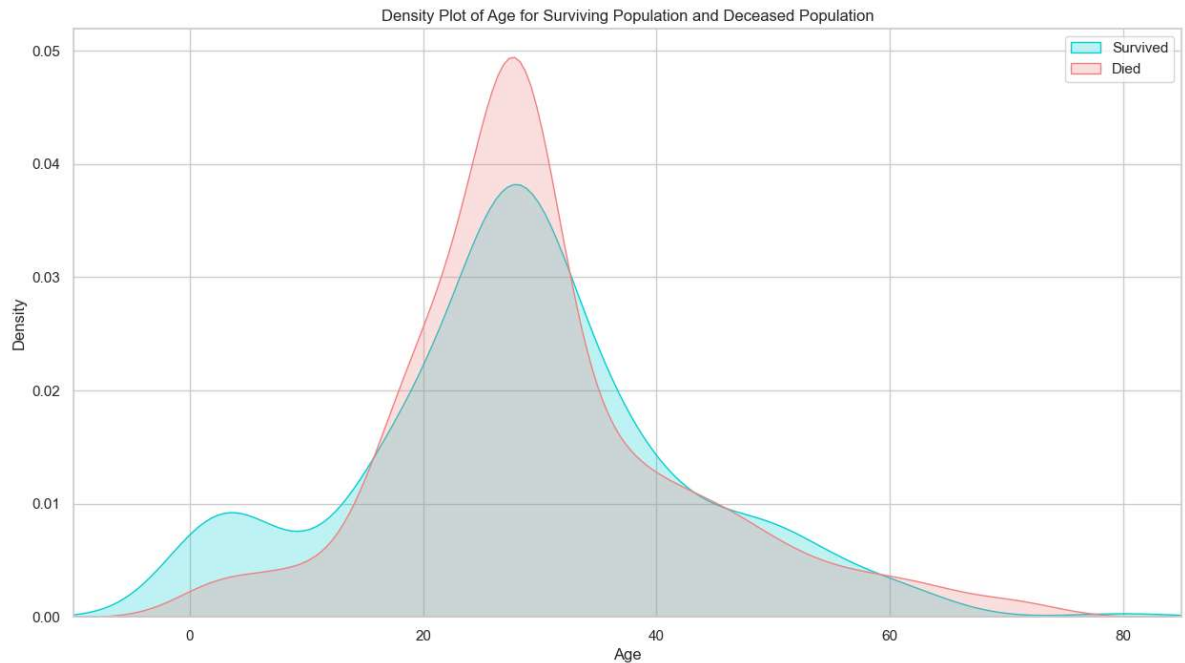
Out[39]:

```

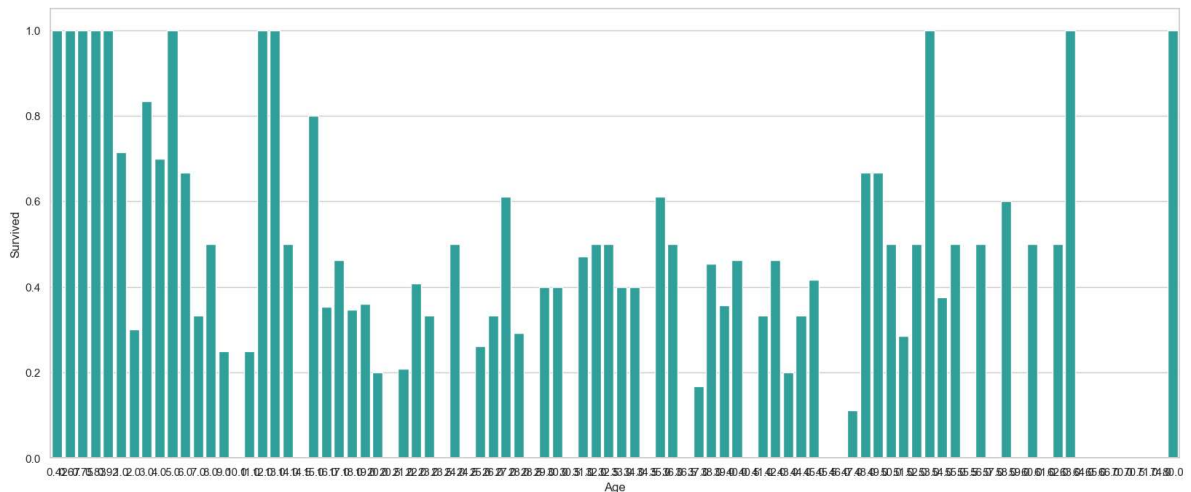
	PassengerId	Age	Fare	TravelAlnoe	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Embark
0	892	34.5	7.8292	1	False	False	True	False	
1	893	47.0	7.0000	0	False	False	True	False	
2	894	62.0	9.6875	1	False	True	False	False	
3	895	27.0	8.6625	1	False	False	True	False	
4	896	22.0	12.2875	0	False	False	True	False	

# Exploratory Data Analysis

```
In [40]: plt.figure(figsize=(15,8))
ax = sns.kdeplot(final_train["Age"][final_train.Survived == 1], color="darkturquoise", fill=True)
sns.kdeplot(final_train["Age"][final_train.Survived == 0], color="lightcoral", fill=True)
plt.legend(['Survived', 'Died'])
plt.title('Density Plot of Age for Surviving Population and Deceased Population')
ax.set(xlabel='Age')
plt.xlim(-10,85)
plt.show()
```



```
In [41]: plt.figure(figsize=(20,8))
avg_survival_byage = final_train[["Age", "Survived"]].groupby(['Age'], as_index=False).mean()
g = sns.barplot(x='Age', y='Survived', data=avg_survival_byage, color="LightSeaGreen")
plt.show()
```



```
In [42]: final_train['IsMinor']=np.where(final_train['Age']<=16, 1, 0)
print(final_train['IsMinor'])
```

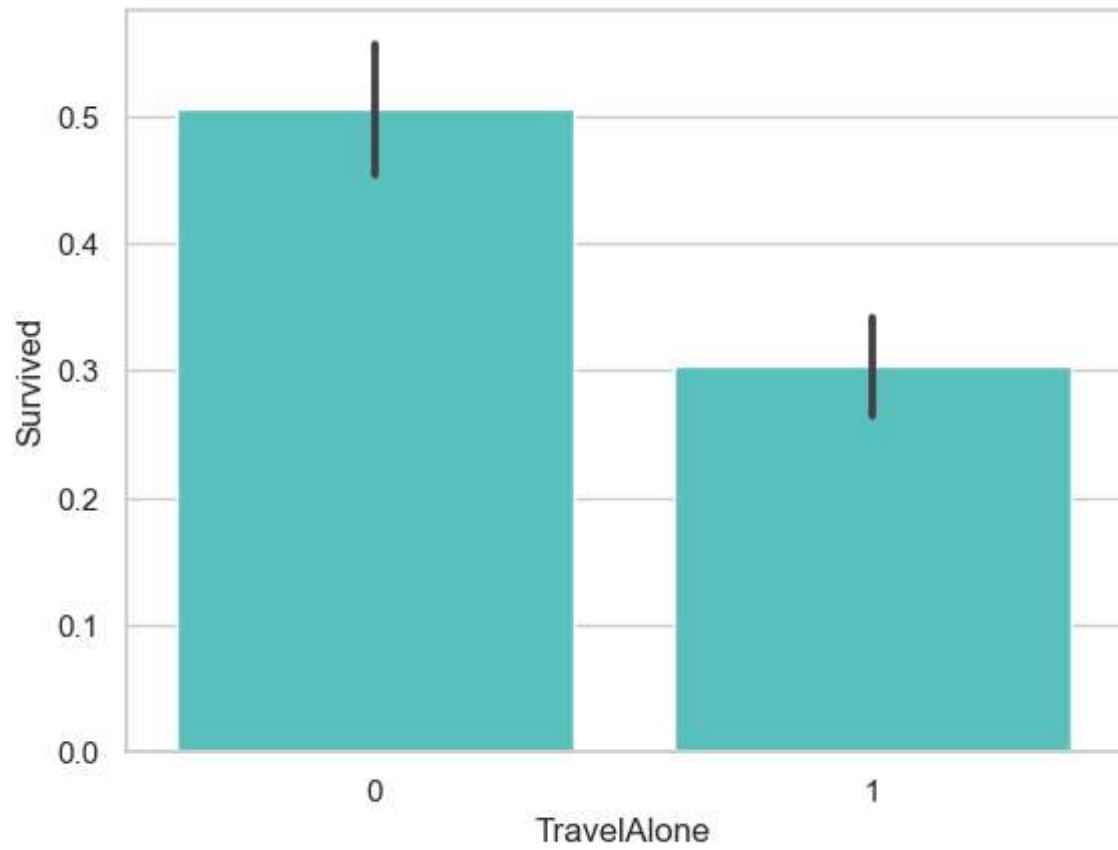
```
0      0
1      0
2      0
3      0
4      0
..
886    0
887    0
888    0
889    0
890    0
Name: IsMinor, Length: 891, dtype: int32
```

```
In [43]: final_test['IsMinor']=np.where(final_test['Age']<=16, 1, 0)
print(final_test['IsMinor'])
```

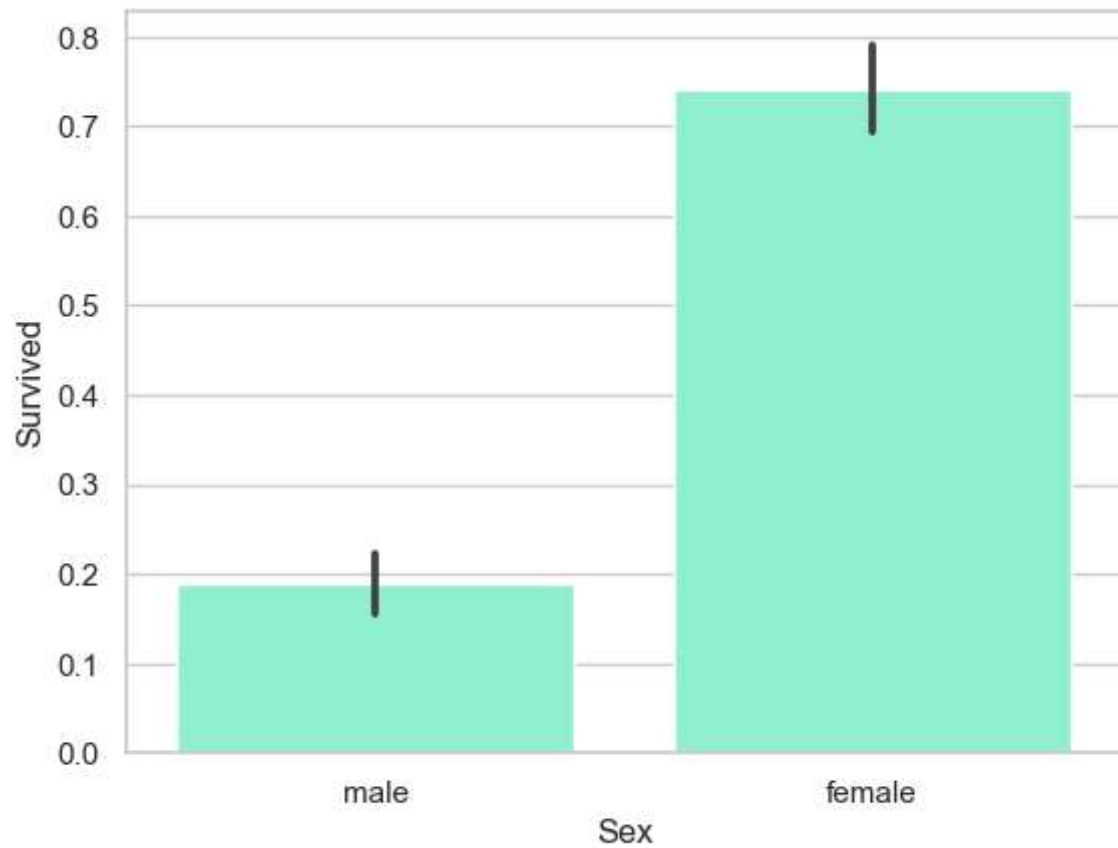
```
0      0
1      0
2      0
3      0
4      0
..
413    0
414    0
415    0
416    0
417    0
Name: IsMinor, Length: 418, dtype: int32
```



```
In [44]: sns.barplot(x='TravelAlone', y='Survived', data=final_train, color="mediumturquoise",  
plt.show())
```



```
In [45]: import seaborn as sns
import matplotlib.pyplot as plt
sns.barplot(x='Sex', y='Survived', data=train_df, color='aquamarine')
plt.show()
```



In [ ]: