

In [1]:

```
import pandas as pd
import numpy as np
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]:

```
train_df=pd.read_csv(r"C:\Users\RAMADEVI SURIPAKA\Downloads\train.gender_submission.csv")
train_df
```

Out[2]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...
886	887	0	2Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

891 rows × 12 columns



In [27]:

In [3]:

```
train_df.head()
```

Out[3]:

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



In [4]:

```
train_df.tail()
```

Out[4]:

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



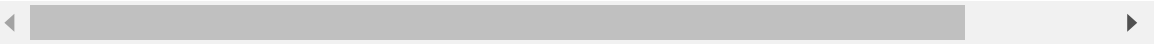
In [5]:

```
test_df=pd.read_csv(r"C:\Users\RAMADEVI SURIPAKA\Downloads\test.gender_submission.csv")
test_df
```

Out[5]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cat
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	N
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	N
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	N
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	N
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	N
...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	N
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C1
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	N
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	N
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	N

418 rows × 11 columns



In [6]:

```
test_df.head()
```

Out[6]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
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	

In [7]:

```
test_df.tail()
```

Out[7]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
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	C101	
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	

In [8]:

```
train_df.describe()
```

Out[8]:

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

In [9]:

```
test_df.describe()
```

Out[9]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [10]:

```
train_df.shape
```

Out[10]:

```
(891, 12)
```

In [11]:

```
test_df.shape
```

Out[11]:

```
(418, 11)
```

In [12]:

```
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 [13]:

```
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
```



In [14]:

```
train_df.isnull().sum()
```

Out[14]:

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

In [15]:

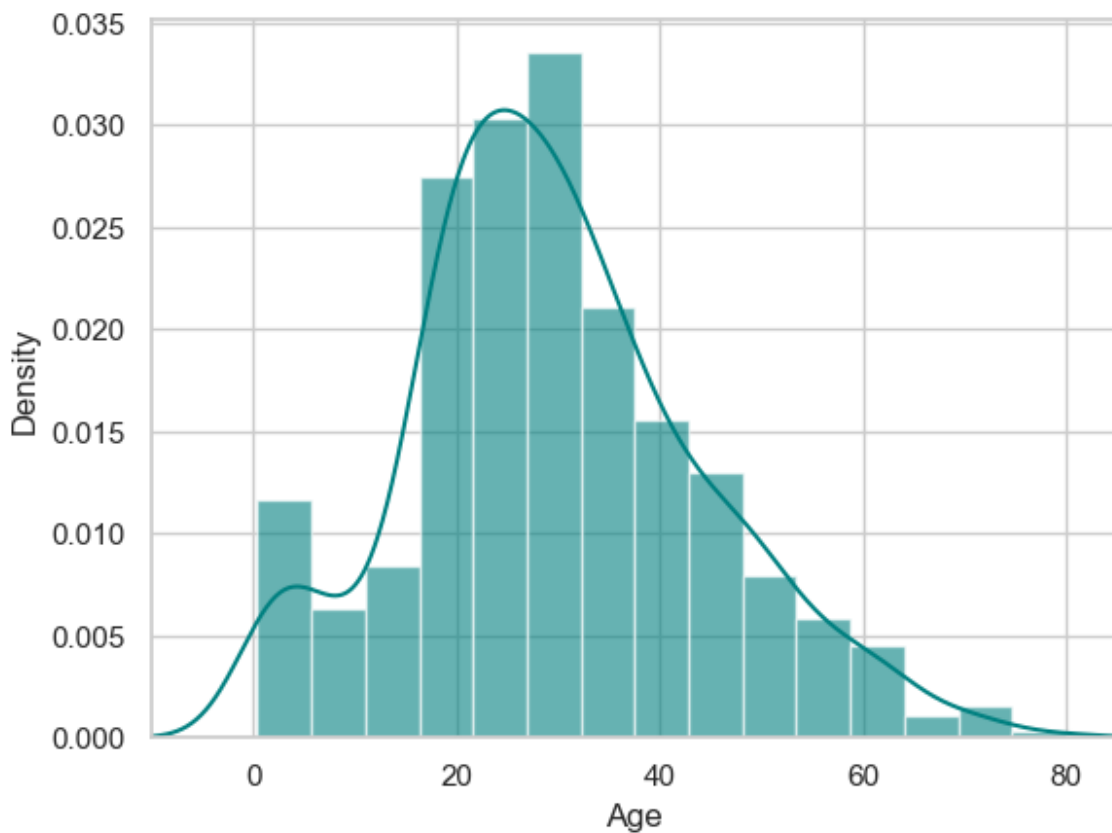
```
test_df.isnull().sum()
```

Out[15]:

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

In [16]:

```
ax=train_df["Age"].hist(bins=15,density=True,stacked=True,color='teal',alpha=0.6)
train_df["Age"].plot(kind='density',color='teal')
ax.set(xlabel="Age")
plt.xlim(-10,85)
plt.show()
```



In [17]:

```
print(train_df["Age"].mean(skipna=True))
print(train_df["Age"].median(skipna=True))
```

```
29.69911764705882
28.0
```

In [18]:

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

```
77.10437710437711
```

In [19]:

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

```
0.22446689113355783
```

In [21]:

```
print('Boarded passengers grouped by part of embarkation (C = Cherbourg,Q=Queenstown,S=Southampton)')
print(train_df['Embarked'].value_counts())
sns.countplot(x='Embarked',data=train_df,palette='Set2')
plt.show()
```

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

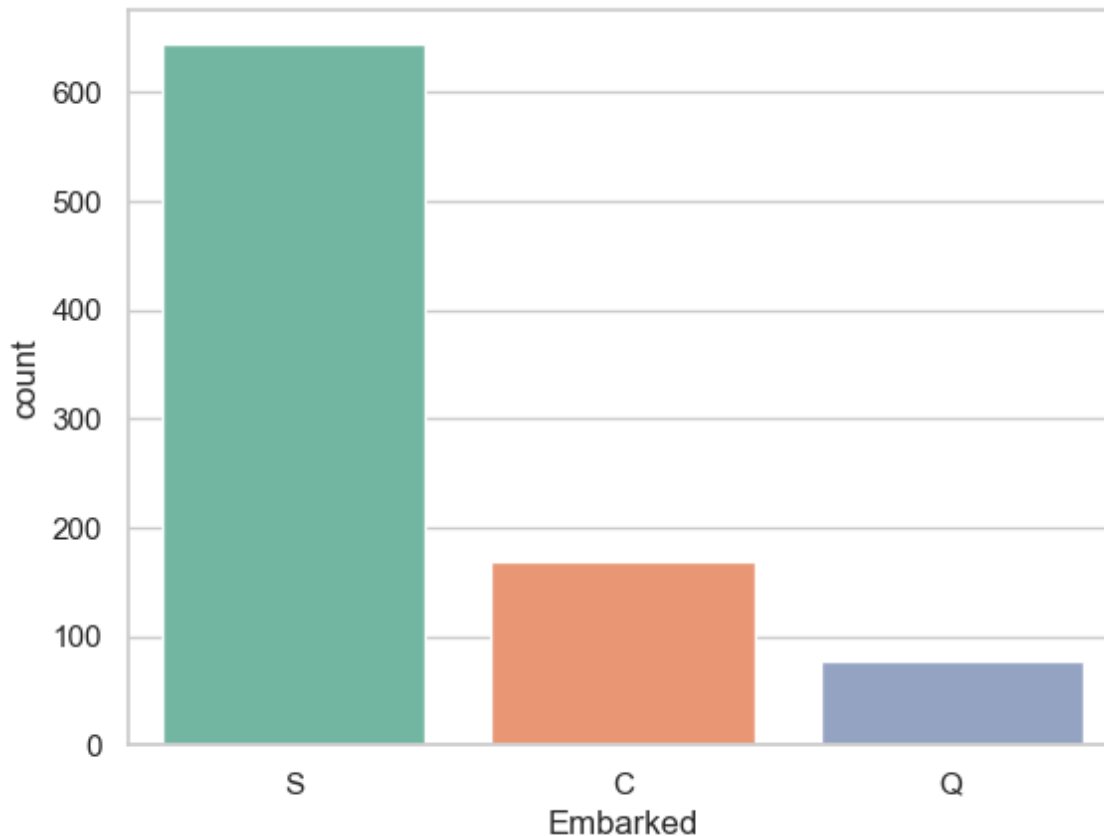
Embarked

S 644

C 168

Q 77

Name: count, dtype: int64



In [22]:

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

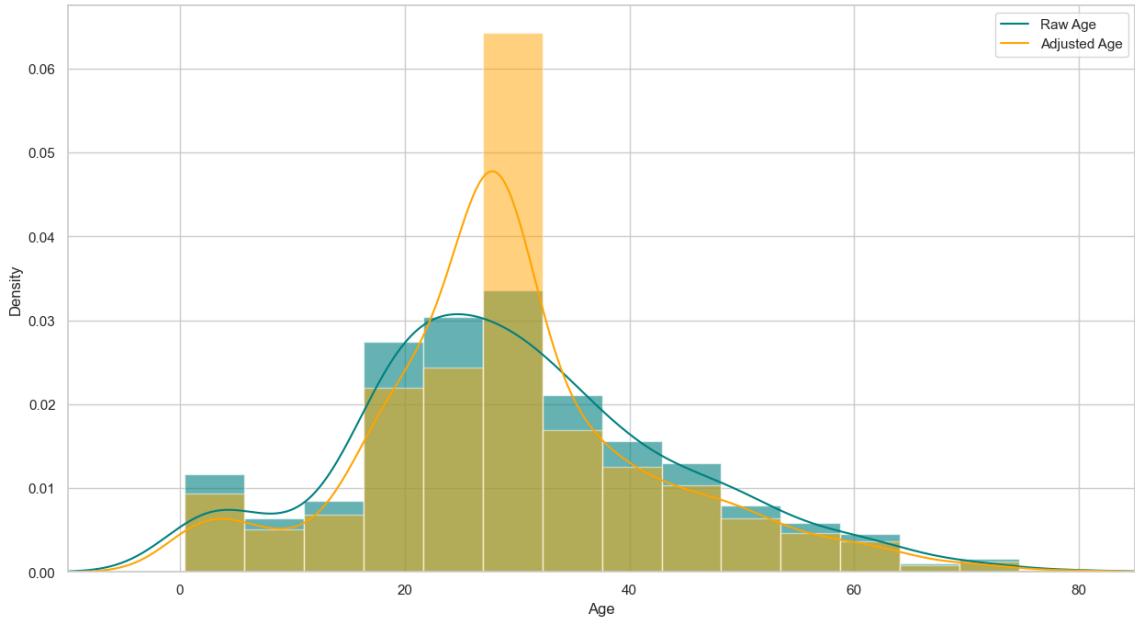
S

In [25]:

```
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 [26]:

```
plt.figure(figsize=(15,8))
ax=train_df["Age"].hist(bins=15,density=True,stacked=True,color='teal',alpha=0.6)
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()
```



In [27]:

```
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[27]:

	Survived	Age	SibSp	Parch	Fare	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Emba
0	0	22.0	1	0	7.2500	False	False	True	False	
1	1	38.0	1	0	71.2833	True	False	False	True	
2	1	26.0	0	0	7.9250	False	False	True	False	
3	1	35.0	1	0	53.1000	True	False	False	False	
4	0	35.0	0	0	8.0500	False	False	True	False	

In [29]:

```
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['TravelAlone']=np.where((test_data["SibSp"]+test_data["Parch"])>0,0,1)
test_data.drop("SibSp",axis=1,inplace=True)
test_data.drop("Parch",axis=1,inplace=True)
testing=pd.get_dummies(train_data,columns=["Pclass","Embarked","Sex"])
testing.drop("Sex_female",axis=1,inplace=True)
testing.drop("PassengerId",axis=1,inplace=True)
testing.drop("Name",axis=1,inplace=True)
testing.drop("Ticket",axis=1,inplace=True)
final_test=testing
final_test.head()
```

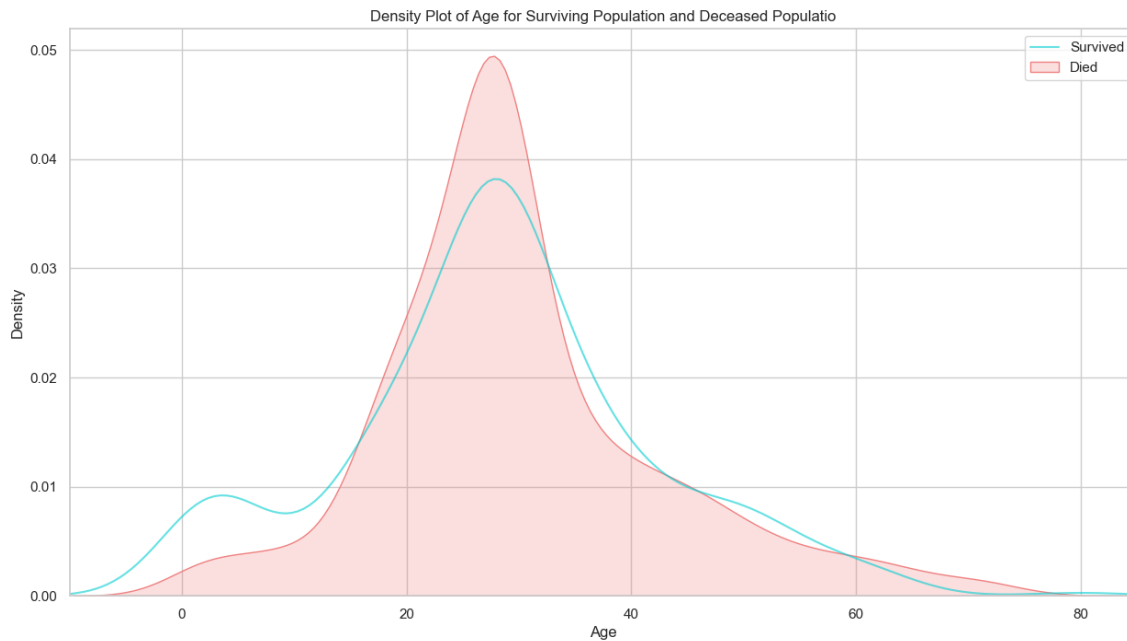
Out[29]:

	Survived	Age	SibSp	Parch	Fare	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Emba
0	0	22.0	1	0	7.2500	False	False	True	False	
1	1	38.0	1	0	71.2833	True	False	False	True	
2	1	26.0	0	0	7.9250	False	False	True	False	
3	1	35.0	1	0	53.1000	True	False	False	False	
4	0	35.0	0	0	8.0500	False	False	True	False	

EXPLORATORY DATA ANALASYS

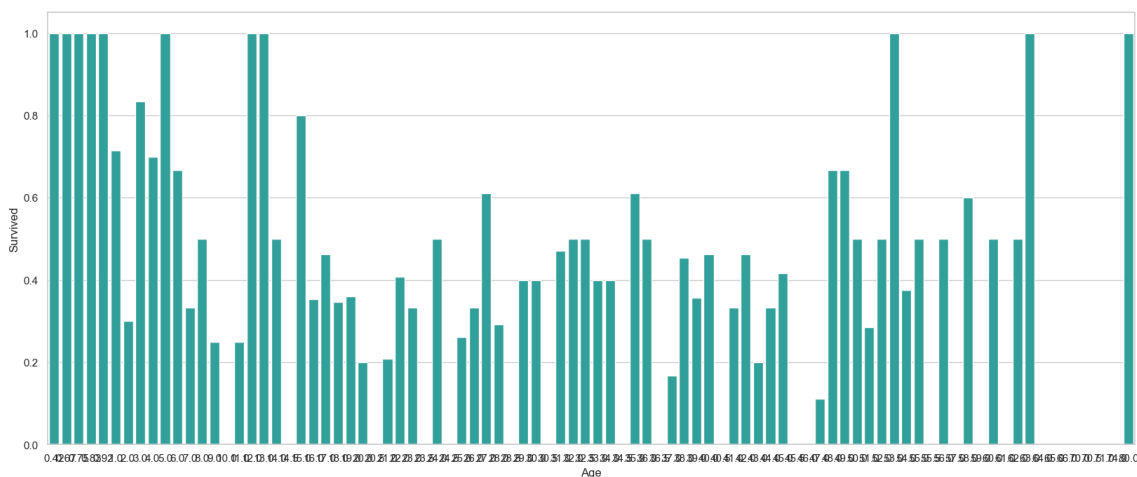
In [31]:

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



In [32]:

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



In [33]:

```
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 [34]:

```
final_test['IsMinor']=np.where(final_test['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 [ ]: