In [2]:

```
greetings = "Assalam-o-Alaikum!"
print(greetings)
```

```
Assalam-o-Alaikum!
```

In [3]:

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

Import Data Set

In [4]:

```
df = pd.read_csv("titanic.csv")
df.head(5)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

How many columns are there in the dataset?

In [5]:

```
no_columns = df.shape[1]
print("Number Of Columns =", no_columns)
```

```
Number Of Columns = 12
```

2. Which column contains information about whether a passenger survived or not?

In [6]:

```
column_info = df["Survived"].value_counts()
column_info
```

Out[6]:

```
0    549
1    342
Name: Survived, dtype: int64
```

In [7]:

```
column_info = column_info.to_frame().reset_index()
column_info
```

Out[7]:

	index	Survived
0	0	549
1	1	342

In [8]:

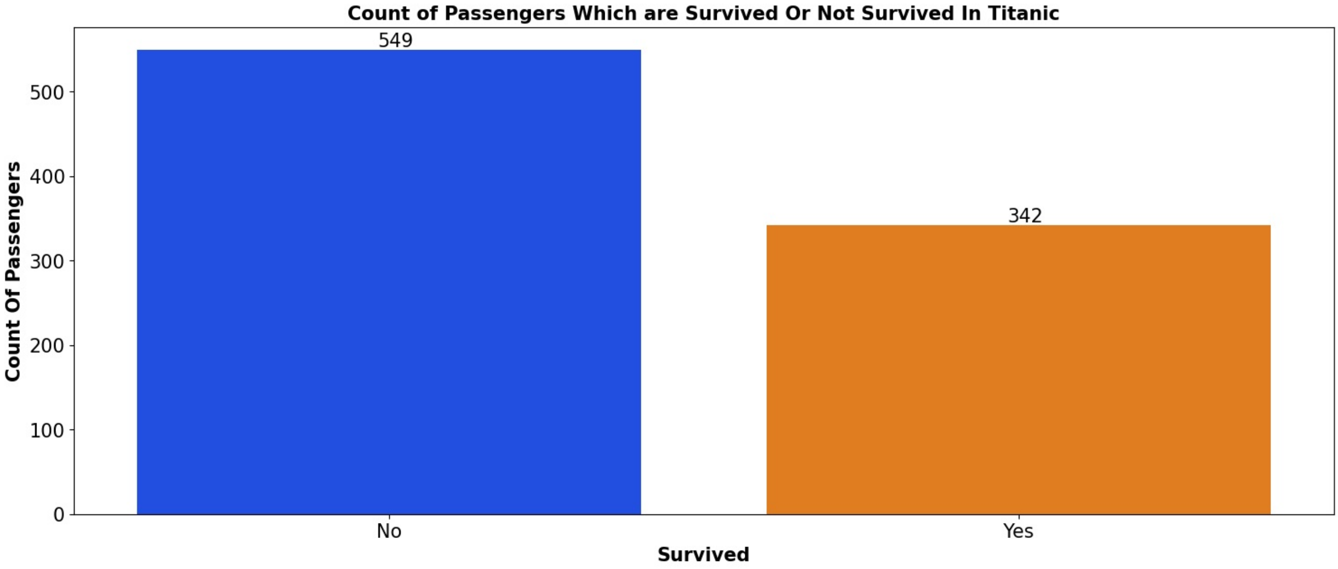
```
column_info.columns = ["Survived", "Passengers"]
column_info
```

Out[8]:

	Survived	Passengers
0	0	549
1	1	342

In [9]:

```
plt.figure(figsize =(18, 7))
graph = sns.barplot(x ="Survived", y = "Passengers", data = column_info, palette = "bright")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                      (p.get_x()+0.41, p.get_height()),
                       ha='center', va='bottom',color= 'black', size = 15)
plt.title("Count of Passengers Which are Survived Or Not Survived In Titanic", weight =("bold"), size ="15")
plt.xticks([0, 1], ["No", "Yes"], size ="15")
plt.yticks(size ="15")
plt.xlabel("Survived", weight ="bold", size ="15")
plt.ylabel("Count Of Passengers", weight ="bold", size ="15")
plt.show()
```



What is the average age of the passengers in the dataset?

Average Age Of Passengers

In [10]:

```
avg_age = df["Age"].mean()
print("Average Age Of Passengers = " + str(int(avg_age)) , "Years")
```

```
Average Age Of Passengers = 29 Years
```

In [11]:

```
avg_m_f = df.groupby("Sex")["Age"].agg(["mean"]).reset_index().round(2)
avg_m_f
```

Out[11]:

	Sex	mean
0	female	27.92
1	male	30.73

In [12]:

```
avg_m_f.columns = ["Gender", "Average age"]
avg_m_f
```

Out[12]:

	Gender	Average age
0	female	27.92
1	male	30.73

In [13]:

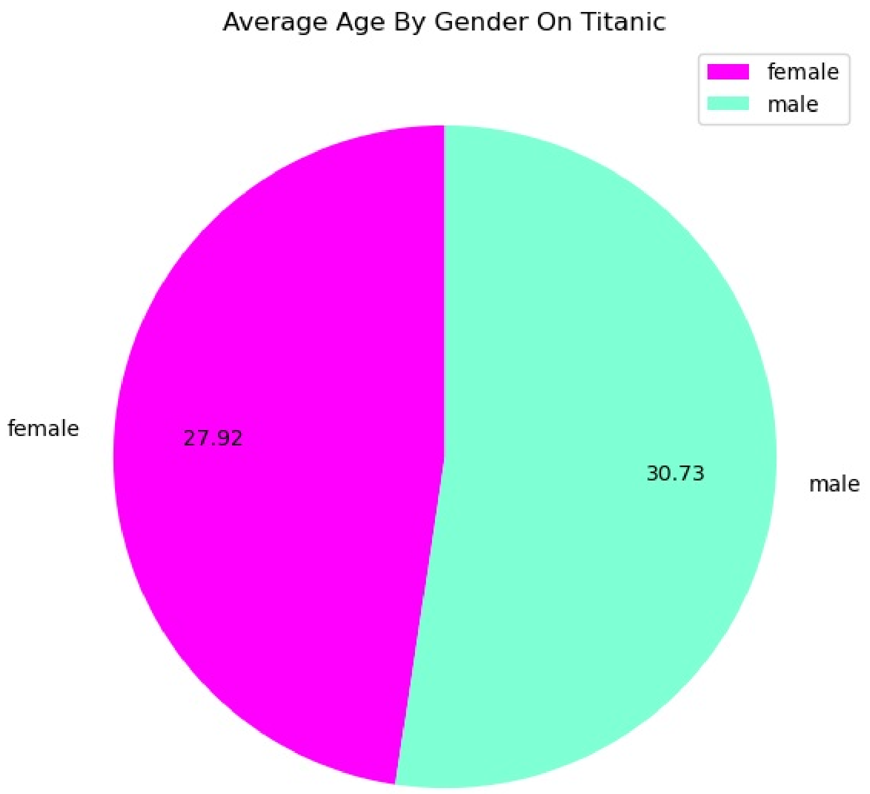
```
plt.figure(figsize=(18, 7))
pie = plt.pie(avg_m_f['Average age'], labels=avg_m_f['Gender'], startangle=90, colors = ["magenta", "aquamarine



# Adding numbers as labels
for i, (age, label) in enumerate(zip(avg_m_f['Average age'], avg_m_f['Gender'])):
    angle = (pie[0][i].theta2 + pie[0][i].theta1) / 2
    x = pie[0][i].r * 0.7 * np.cos(np.deg2rad(angle))
    y = pie[0][i].r * 0.7 * np.sin(np.deg2rad(angle))
    plt.text(x, y, str(age), ha='center', va='center', color = 'black')



# Adding a title
plt.title('Average Age By Gender On Titanic')
plt.legend()
# Display the chart
plt.show()
```



Average Age of Genders Who survived or not survived in titanic

In [14]:

```
sga = df.groupby(["Survived", "Sex"])["Age"].agg("mean").to_frame().round(2).reset_index()
sga
```

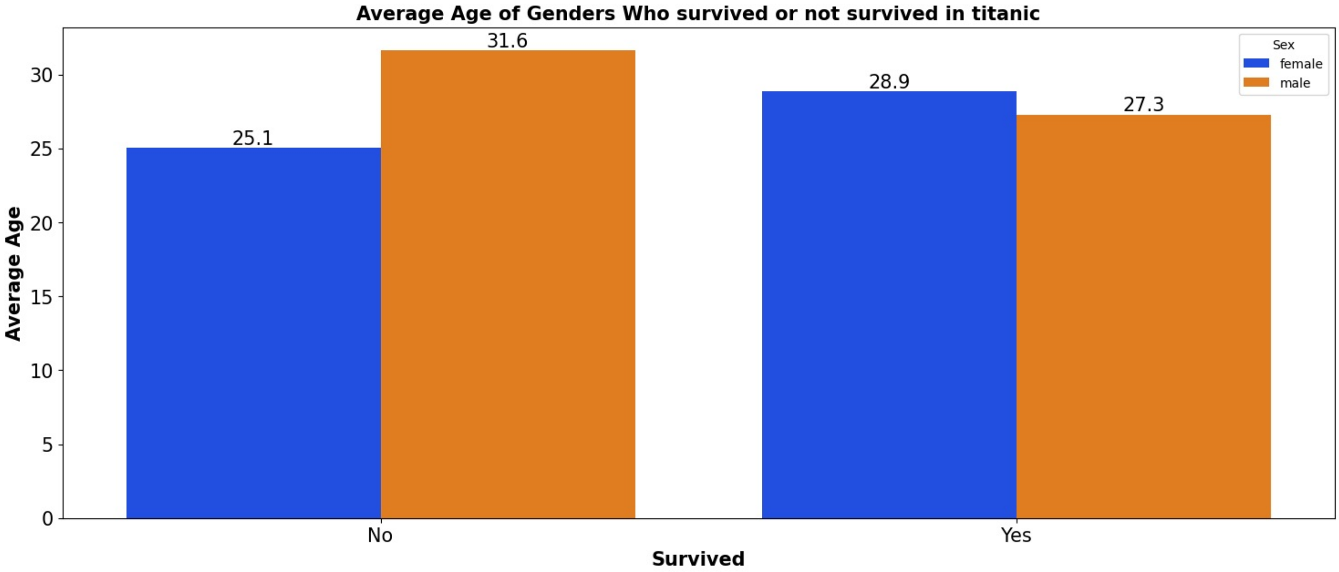
Out[14]:

	Survived	Sex	Age
0	0	female	25.05
1	0	male	31.62
2	1	female	28.85
3	1	male	27.28

Create a bar chart showing the number of survivors and non-survivors.

In [15]:

```
plt.figure(figsize = (18, 7))
graph = sns.barplot(x ="Survived", y ="Age", hue ="Sex", data = sga, palette ="bright")
for p in graph.patches:
        graph.annotate('{:.01f}'.format(p.get_height()),
                      (p.get_x()+0.2, p.get_height()),
                       ha='center', va='bottom',color= 'black', size = 15)
plt.title("Average Age of Genders Who survived or not survived in titanic", weight ="bold", size =15)
plt.xticks([0, 1], ["No", "Yes"], size ="15")
plt.yticks(size =15)
plt.xlabel("Survived", weight ="bold", size =15)
plt.ylabel("Average Age", weight ="bold", size =15)
plt.show()
```



In [16]:

```
df.head(1)
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S

How many male and female Survived or Died in Titanic

In [17]:

```
no_male_female = df.groupby(["Survived", "Sex"])["Sex"].agg("count").to_frame()
no_male_female.columns = ["Count"]
no_male_female =no_male_female.reset_index()
no_male_female
```

Out[17]:

	Survived	Sex	Count
0	0	female	81
1	0	male	468
2	1	female	233
3	1	male	109

How many male and female passengers are there in the dataset?

In [18]:

```
no_male_female = df["Sex"].value_counts()
no_male_female
```

Out[18]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [19]:

```
df.head(1)
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S

What is the highest fare paid by a passenger?

In [26]:

```
highest_fare = df["Fare"].max()
print("Highest Fare Paid by a Passenger = $" + str(round(highest_fare, 2)))
```

```
Highest Fare Paid by a Passenger = $512.33
```

Lowest Fare Paid by a passenger

In [27]:

```
lowest_fare = df["Fare"].min()
print("Lowest Fare Paid by a Passenger = $" +str(lowest_fare))
```

```
Lowest Fare Paid by a Passenger = $0.0
```

How many passengers embarked from the 'S' port?

In [29]:

```
count_s_embarked = len(df[df["Embarked"] == "S"])
```

```
print("Number of passengers embarked from 'S' port =", count_s_embarked)
```

```
Number of passengers embarked from 'S' port = 644
```

How many passengers were traveling alone (without any siblings, spouse, parents, or children)?

In [31]:

```
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 [34]:

```
pas_trv_aln = len(df[df["SibSp"] == 0])
print("Total Passengers Which were Trv.Alone =", pas_trv_aln)
```

```
Total Passengers Which were Trv.Alone = 608
```

What is the most common cabin class among the passengers?

In [42]:

```
cabin_counts = df["Cabin"].value_counts().to_frame().reset_index().head(10)
cabin_counts.columns =["Cabin Class", "Counts Of Cabins"]
cabin_counts
```

Out[42]:

	Cabin Class	Counts Of Cabins
0	B96 B98	4
1	G6	4
2	C23 C25 C27	4
3	C22 C26	3
4	F33	3
5	F2	3
6	E101	3
7	D	3
8	C78	2
9	C93	2

Can you identify any missing values in the dataset? If yes, in which columns are they present?

In [49]:

```
missing_values = df.isnull().sum().to_frame().reset_index()
missing_values.columns =["Colums", "Missing Values"]
missing_values
```

Out[49]:

	Colums	Missing Values
0	PassengerId	0
1	Survived	0
2	Pclass	0
3	Name	0
4	Sex	0
5	Age	177
6	SibSp	0
7	Parch	0
8	Ticket	0
9	Fare	0
10	Cabin	687
11	Embarked	2

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js