Exploratory Data Analysis (EDA) of Titanic Survival Problem.

To do the same we will use the Pandas, Seaborn and Matplotlib library.

Dataset contains the details of the passengers who had boarded the ship.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

```
df=pd.read_csv("/content/train.csv")
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	1	0	PC 17599

```
df.shape
```

```
(891, 12)
```

```
df.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
df.fillna(df.mean(), inplace = True)
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
```

```
     Ticket           0
     Fare             0
     Cabin          687
     Embarked         2
     dtype: int64
```

```
# fill values of Embarked column
df["Embarked"].fillna("S", inplace = True)
df.isnull().sum()
```

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

```
# drop Cabin column because it has lot of null values. 687/891
drop_cabin = df.isnull().sum()[df.isnull().sum() > (50/100 * df.shape[0])]
drop_cabin
```

```
     Cabin    687
     dtype: int64
```

```
df.drop(drop_cabin.index, axis = 1, inplace = True)
df.isnull().sum()
```

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

```
df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
PassengerId	1.000000	-0.005007	-0.035144	0.033207	-0.057527	-0.001652	0.01
Survived	-0.005007	1.000000	-0.338481	-0.069809	-0.035322	0.081629	0.25
Pclass	-0.035144	-0.338481	1.000000	-0.331339	0.083081	0.018443	-0.54
Age	0.033207	-0.069809	-0.331339	1.000000	-0.232625	-0.179191	0.09
SibSp	-0.057527	-0.035322	0.083081	-0.232625	1.000000	0.414838	0.15
Parch	-0.001652	0.081629	0.018443	-0.179191	0.414838	1.000000	0.21

```
df.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Tick
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	2115
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	1120
				Johnston					

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.0
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.2
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.6
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.0
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.9
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.4
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.0
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.3

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 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  Embarked     891 non-null    object
 dtypes: float64(2), int64(5), object(4)
 memory usage: 76.7+ KB
```

```
# create a new column Family size by adding SibSp and Parch

df["FamilySize"] = df["SibSp"] + df["Parch"]
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	1	0	PC 17599

```
# drop SibSp and Parch because we create new column FamilySize instaed of them

df.drop(["SibSp", "Parch"], axis = 1, inplace = True)
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	Ticket	Fare	Embar
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	PC 17599	71.2833	

```
df.corr()
```

	PassengerId	Survived	Pclass	Age	Fare	FamilySize
PassengerId	1.000000	-0.005007	-0.035144	0.033207	0.012658	-0.040143
Survived	-0.005007	1.000000	-0.338481	-0.069809	0.257307	0.016639
Pclass	-0.035144	-0.338481	1.000000	-0.331339	-0.549500	0.065997

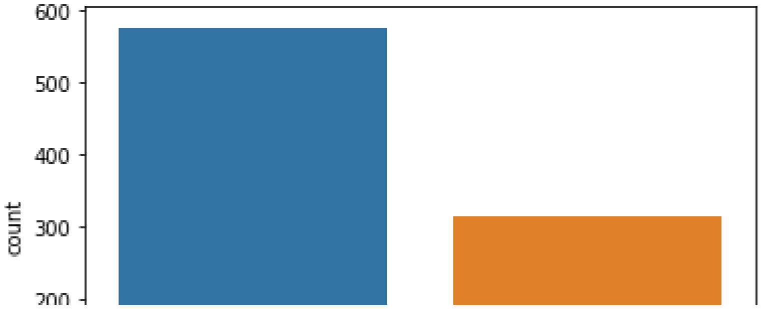
```
# filtered alone persons/passengers

df["Alone"] = [0 if df["FamilySize"][i] > 0 else 1 for i in df.index]
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	Ticket	Fare	Embar
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	PC 17599	71.2833	

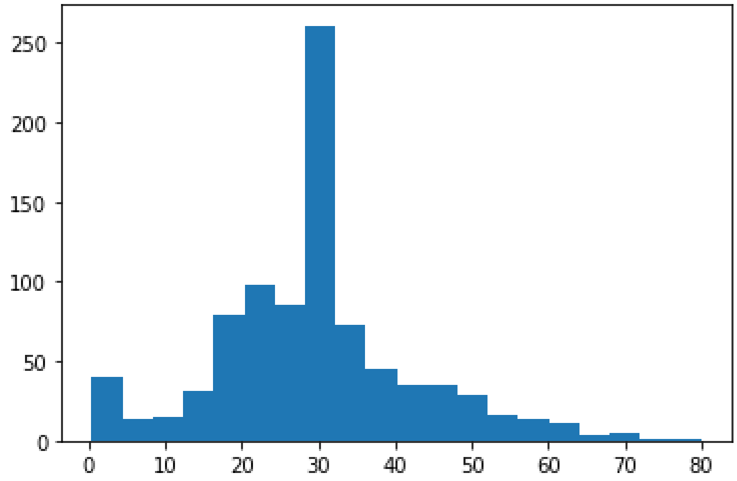
```
df.corr()
```

	PassengerId	Survived	Pclass	Age	Fare	FamilySize	
PassengerId	1.000000	-0.005007	-0.035144	0.033207	0.012658	-0.040143	0.0
Survived	-0.005007	1.000000	-0.338481	-0.069809	0.257307	0.016639	-0.2
Pclass	-0.035144	-0.338481	1.000000	-0.331339	-0.549500	0.065997	0.1
Age	0.033207	-0.069809	-0.331339	1.000000	0.091566	-0.248512	0.1
Fare	0.012658	0.257307	-0.549500	0.091566	1.000000	0.217138	-0.2
FamilySize	-0.040143	0.016639	0.065997	-0.248512	0.217138	1.000000	-0.6
Alone	0.057462	-0.203367	0.135207	0.179775	-0.271832	-0.690922	1.0

```
# sex ratio of passengers

sb.countplot(x = "Sex", data = df);
```

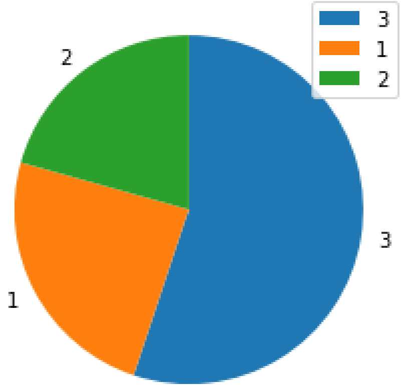


```
# age distribution

plt.hist(x = df["Age"], bins = 20);
```



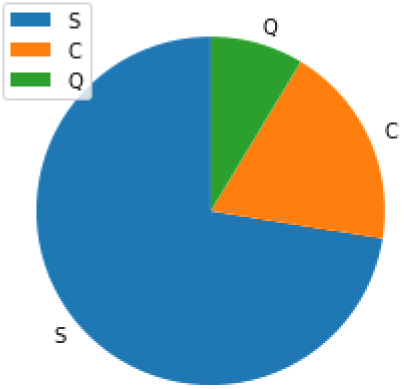
```
# passenger class
x = df["Pclass"].value_counts()
plt.pie(x, labels = x.index, startangle = 90, counterclock = False);
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fc9c1352710>
```

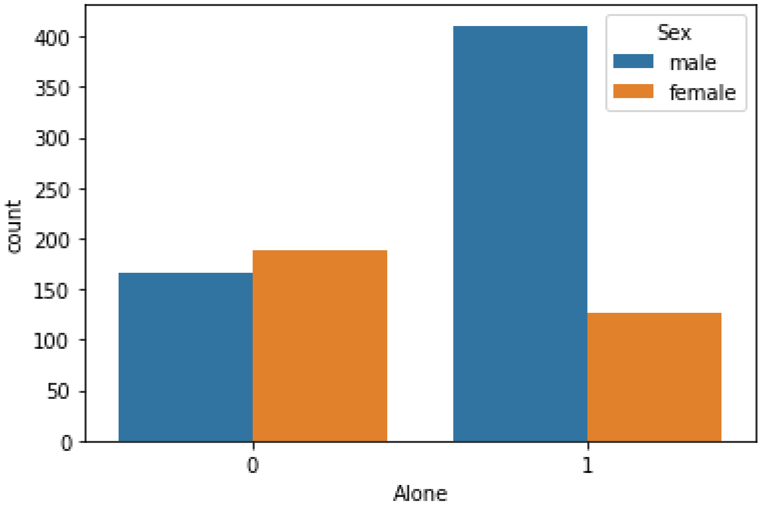


```
#Embarked
y = df["Embarked"].value_counts()
plt.pie(y, labels = y.index, startangle = 90, counterclock = True);
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fc9c2b80f90>
```

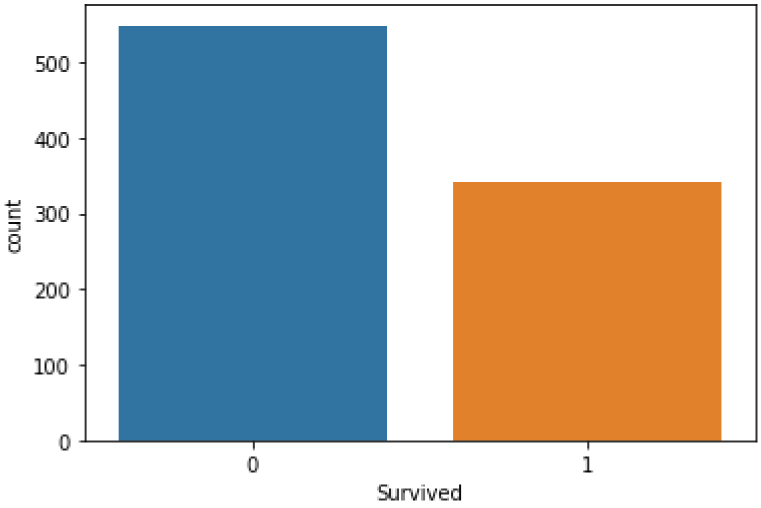


```
# survive rate of alone person according to their sex

sb.countplot(x = "Alone", hue = "Sex", data = df);
```

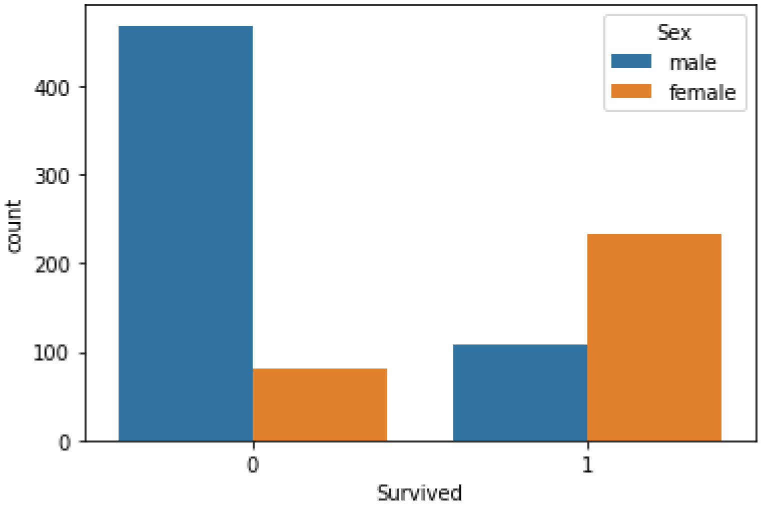


```
# survive rate of family

sb.countplot(x = "FamilySize", data = df);
```

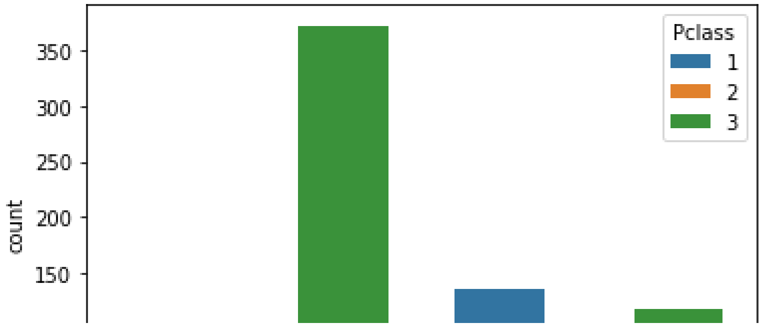
```
# total survived passengers

sb.countplot(x = "Survived", data = df);
```



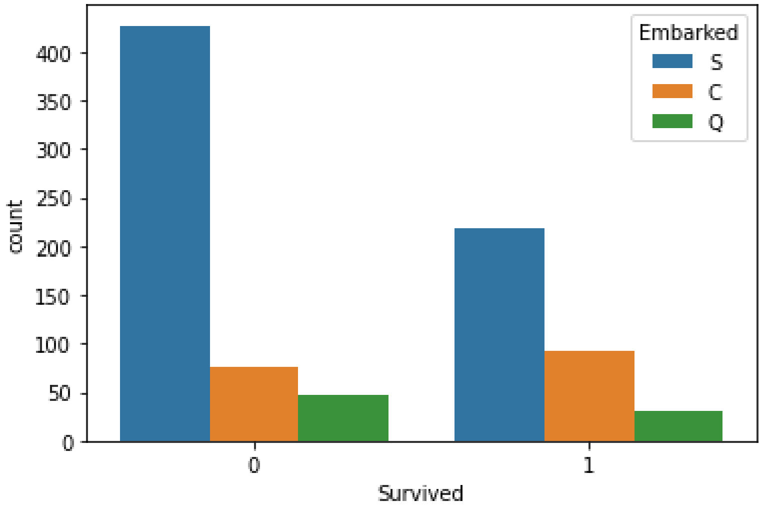
```
# survived ratio according to sex

sb.countplot(x = "Survived", hue = "Sex", data = df);
```



```
# accoring to pclass

sb.countplot(x = "Survived", hue = "Pclass", data = df);
```

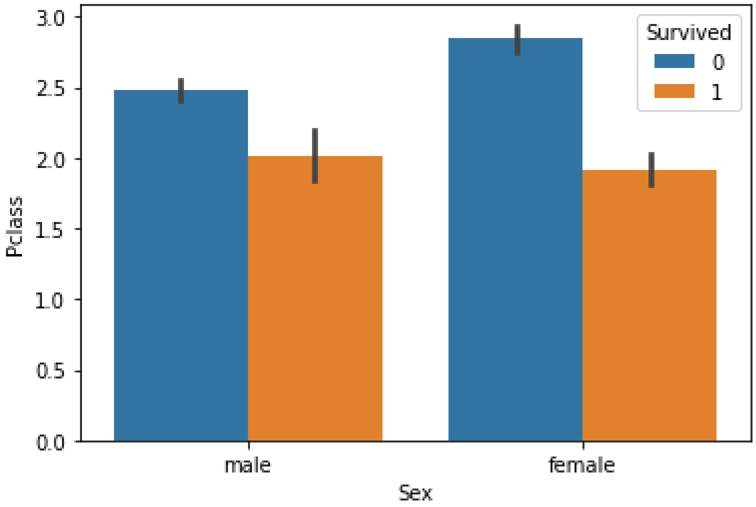


```
# according to embarked

sb.countplot(x = "Survived", hue = "Embarked", data = df);
```



```
# accroding to sex and passenger class

sb.barplot("Sex", "Pclass", hue = "Survived", data = df);
```

```
    /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass tl
      FutureWarning
```



● ✕