Programming for data science - Question 01 - Part A - COMScDS221P-015

Titanic Exploratory Data Analysis

As per the task, we are exploring the famous story dataset of the sinking of the Royal Mail Steamer Titanic ship. This ship had wrecked due to a collision with an iceberg during its maiden voyage from Southampton, England to New York City on April 14th, 1912. There was an estimate of 2240 passengers including crew members. This disaster claimed the lives of more than 1,500 passengers and crew members and since then this has become one of the most unforgettable maritime tragedies in history. Disaster size was reducible but due to the lack of sufficient lifeboats on the ship and the ship's hull was not strong enough to withstand the impact of an iceberg, the titanic disaster happened.

In this exploratory data analysis, we are using several python libraries including Pandas, Numpy, Matplotlib, and Seaborn. And we import and assign the dataset as 'df" as the naming convention for all the analysis and coding below.

```
#Importing the dataset and having a review of the attributes and data in it.
filename= "C:\\Users\\Gihan\\Downloads\\train.csv"

df= pd.read_csv(filename)

df.head(10)
```

✓ 0.1s

	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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

I have run a quick overview of the dataset through the .info( ) function and discovered there are a total of 12 columns and 891 rows.

```
# Quick review of the data types and null counts in the dataset to understand the data preprocessing
df.info()
```

✓ 0.1s

```
<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
```

The dataset abbreviation data is explained below.

Abbreviation	**Definition**	**Assigned information**	**Data type**
PassengerID	Passenger ID		int64
Survived	Survival	0 = No / 1 = Yes	int64
Pclass	Ticket class	1 = 1st 2 = 2nd 3 = 3rd	int64
sex	Gender of the passengers		object
Age	Age of the passengers		float64
SibSp	Sib = No of siblings Sp = No of spouses		int64
Parch	Par = No of Parents Ch = No of Children		int64
Ticket	Ticket number		object
Fare	Passenger Fare		float64
Cabin	Cabin number		object
Embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton	object

We have analyzed the dataset to find out the count of the null values using the .isnull( ) function. It resulted in the Cabin column having 687 null values, the Age column having 177 null values, Embarked column having 2 null values.

Summary Statistics of the Dataset

```
#Identify the numeric data in the dataset from .describe()  function.
#This gives us a overview understanding of count/mean/ standard deviation and central tendencies of the data
df.describe()
```

	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

From the .describe( ) function we have identified the mean, count, standard deviation, minimum value, max value, and quartiles for each numerical value column. From the 891 passengers on board, 38% (mean of 0.383838) survived the disaster. Passengers on board were divided into three classes such as first class, second class, and third class. The majority of the passengers were from third class and then second class. The mean age of the passengers is about 30 years. The mean values of "SibSp" and "Parch" are 0.52 and 0.38 respectively, indicating that most passengers were traveling alone or with a small number of family members. The fare range distributes min 0 to a maximum of 512.33 and mean price of a ticket price is £32.20 British pounds.

Comparison Between the Categorical Variable and Survival Mean Rate

Gender

```
#Value counts of the sex column

df['Sex'].value_counts(dropna=False)
```

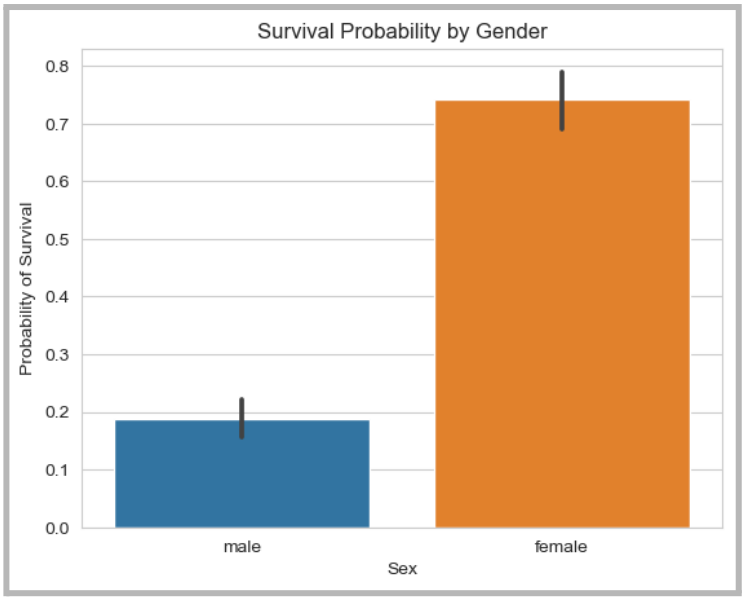
✓ 0.1s

```
#Analysing the mean rate of survival by gender
df[['Sex','Survived']].groupby('Sex', as_index=False).mean().sort_values(by='Survived', ascending=False)
```

```
sns.barplot(x='Pclass',y='Survived', data=df)
plt.title('Survival Probability by Passebger Class')
plt.ylabel('Probability of Survival')
```

```
sns.barplot(x='Sex',y='Survived', data=df)
plt.title('Survival Probability by Gender')
plt.ylabel('Probability of Survival')
```

Sex	Count	Survived(Mean)
Male	577	0.188908
Female	314	0.742038



This bar plot indicates the survival probability by gender of the passengers in the ship. We can analyze there was a higher probability that female passengers survived when compared to the male passengers on the ship.

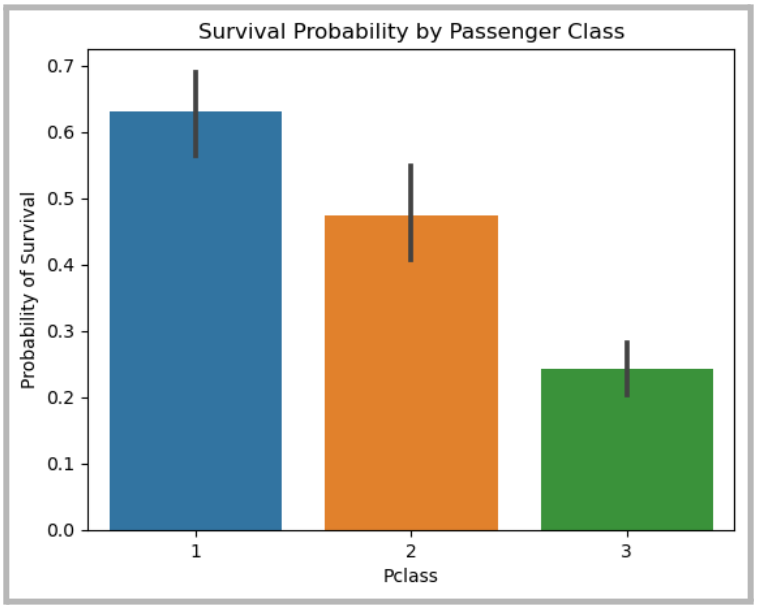
Passenger Class

```
#Calculating the Value counts of the Pclass column
```

```
#Analysing the mean rate of survival by Pclass
df[['Pclass','Survived']].groupby('Pclass', as_index=False).mean().sort_values(by='Survived', ascending=False)
```

```
sns.barplot(x='Pclass',y='Survived', data=df)
plt.title('Survival Probability by Passebger Class')
plt.ylabel('Probability of Survival')
```

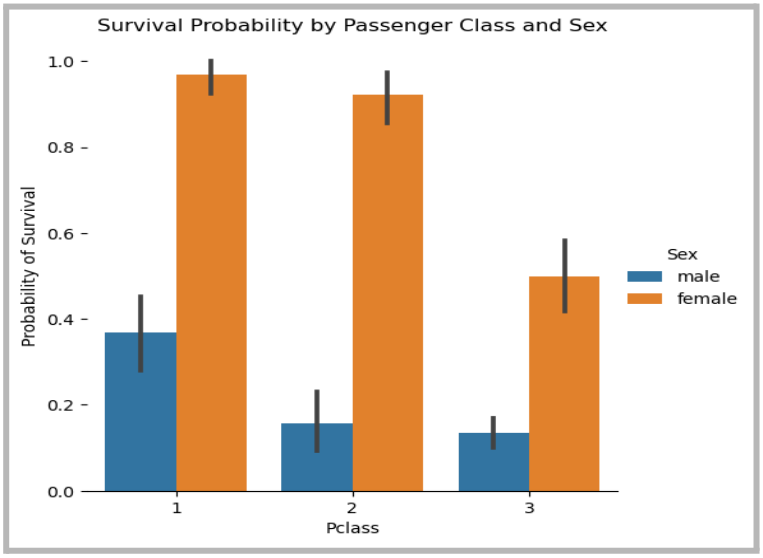
Pclass	Count	Survived(Mean)
1	216	0.629630
2	184	0.472826
3	491	0.242363



This barplot indicates the survival probability by passenger class of the passengers in the ship. We can analyze there was a higher probability of first-class passengers were survived when compared to the second and third passenger class passengers on the ship.

Survival Probability by Passenger Class and Gender

This barplot indicates the survival probability by passenger class of the passengers in the ship. We can analyze there was a higher probability of first-class passengers were survived when compared to the second and third passenger class passengers on the ship. Also first class female and male passengers have a high probability of survival when compared to second and third passenger class travelers.



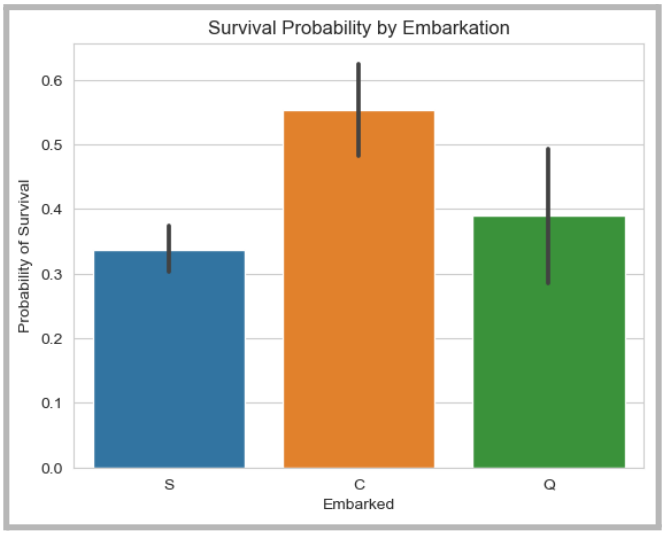
Embarkation

```
#Calculating the Value counts of the Embarked column
df['Embarked'].value_counts(dropna=False)
```

```
sns.barplot(x='Embarked',y='Survived', data=df)
plt.title('Survival Probability by Embarkation')
plt.ylabel('Probability of Survival')
```

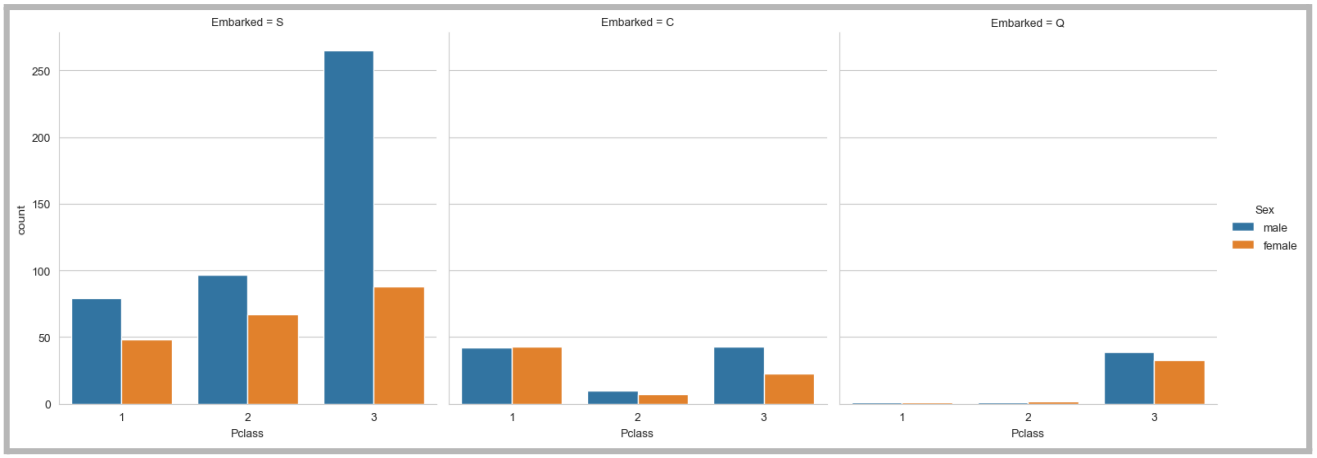
Embarked	Count	Survived(Mean)
S	644	0.553571
C	168	0.553571
Q	77	0.389610

This bar plot indicates the survival probability by embarkation of the passengers in the ship. We can analyze there were higher probability of passengers who have embarked in Cherbourg when compared to the Queenstown and Southampton.



Comparison between embarked locations and Passenger class

```
#Analysing the comparison of Embarked location and Passenger class by Gender
sns.factorplot(x='Pclass',col='Embarked', hue= 'Sex',kind='count', data=df)
```



We can analyze Southampton has the highest count of third class passengers were got on board when contrasting the graph.

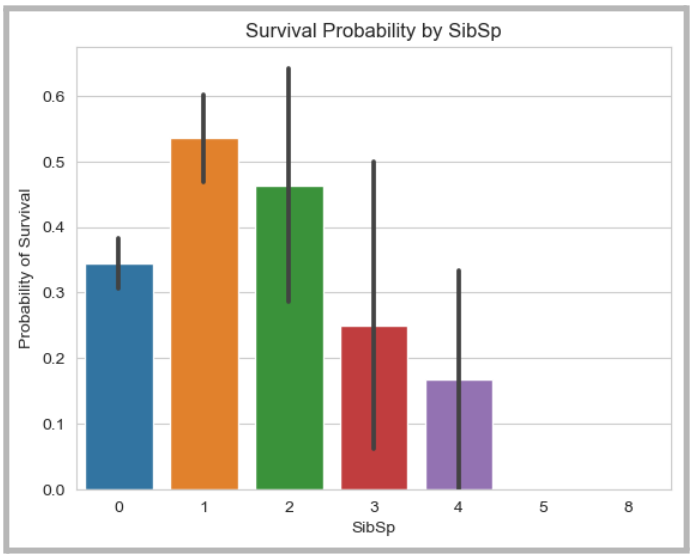
Sibling and Spouse

```
#Calculating the Value counts of the SibSp column

df['SibSp'].value_counts(dropna=False)
```

```
#Analysing the mean rate of survival by SibSp
df[['SibSp','Survived']].groupby('SibSp', as_index=False).mean().sort_values(by='Survived', ascending=False)
```

```
sns.barplot(x='SibSp',y='Survived', data=df)
plt.title('Survival Probability by SibSp')
plt.ylabel('Probability of Survival')
```

SibSp	Count	Survived(Mean)
0	608	0.345395
1	209	0.535885
2	28	0.464286
3	16	0.250000
4	18	0.166667
5	5	0.000000
8	7	0.000000



This barplot indicates the survival probability by siblings and spouses of the passengers in the ship. We can analyze there were a higher probability of survival rate is happened who have one or two siblings or spouse who had no sibling and spouse.

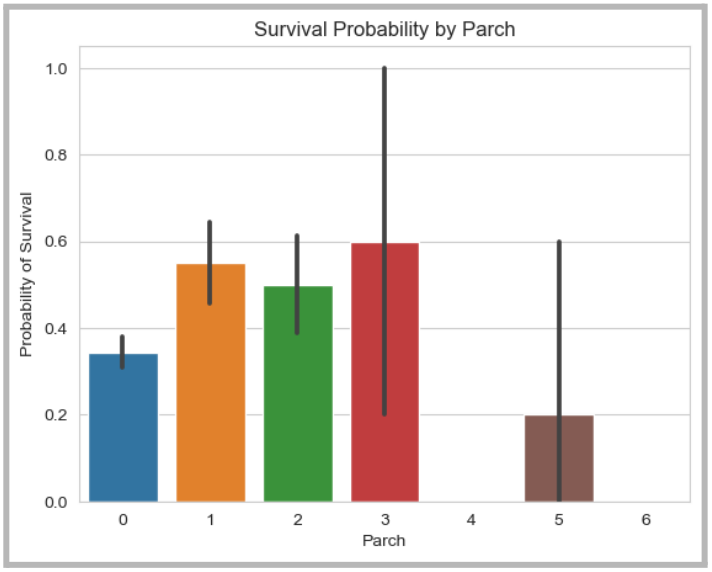
Parent and Children

```
#Calculating the Value counts of the Parch column

df['Parch'].value_counts(dropna=False)
```

```
#Analysing the mean rate of survival by Parch
df[['Parch','Survived']].groupby('Parch', as_index=False).mean().sort_values(by='Survived', ascending=False)
```

```
sns.barplot(x='Parch',y='Survived', data=df)
plt.title('Survival Probability by Parch')
plt.ylabel('Probability of Survival')
```

Parch	Count	Survived(Mean)
0	678	0.343658
1	118	0.550847
2	80	0.500000
3	5	0.600000
4	4	0.000000
5	5	0.200000
6	1	0.000000

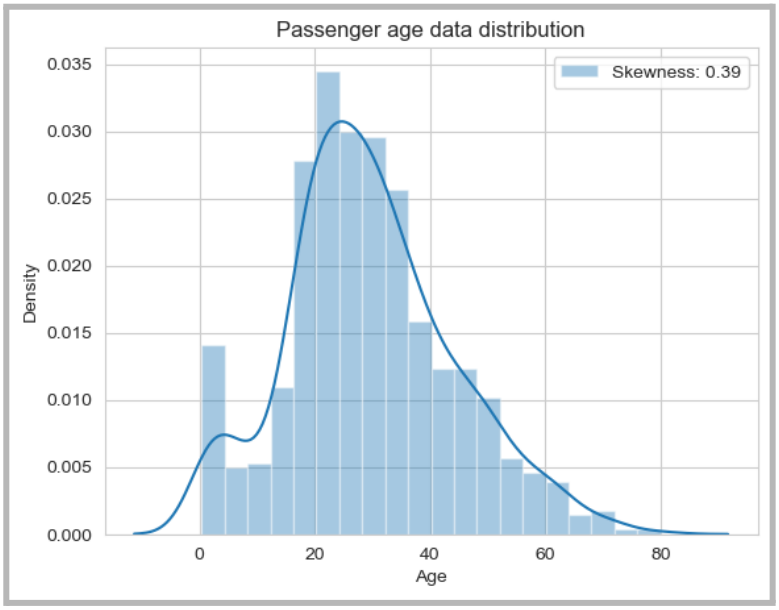


This barplot indicates the survival probability by parents and children of the passengers in the ship. By analyzing this, we can see that individuals who had 1 or 3 parents or children on board the ship had a higher survival rate compared to those who had 4 or more parents or children, or no parents or children.

Age

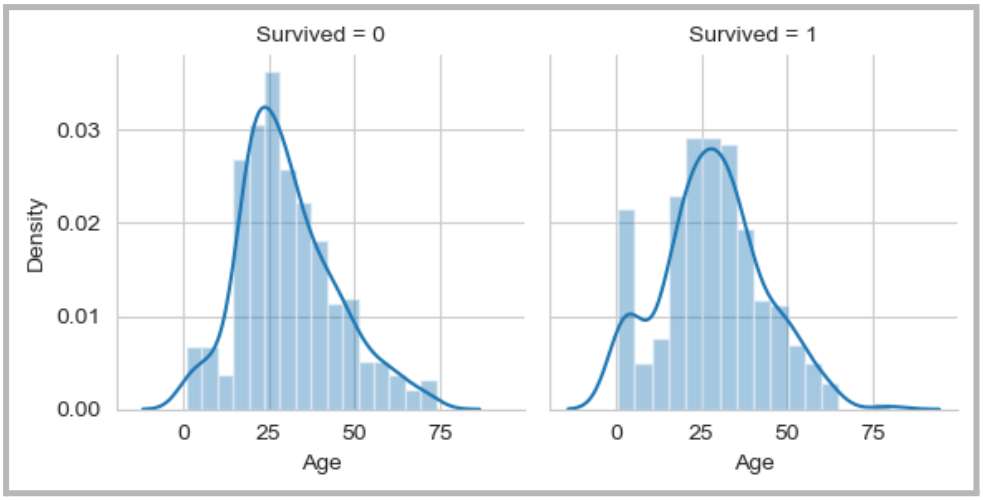
```
# Vusualizing passenger age data distribution
sns.distplot(df['Age'],label= 'Skewness: %.2f'%(df['Age'].skew()))
plt.title('Passenger age data distribution')
plt.legend(loc='best')
```

This distplot indicates the age distribution of the passengers on the ship. We can analyze age has 0 to 85 range of distribution. Most of the age has Distributed between 20 to 40.



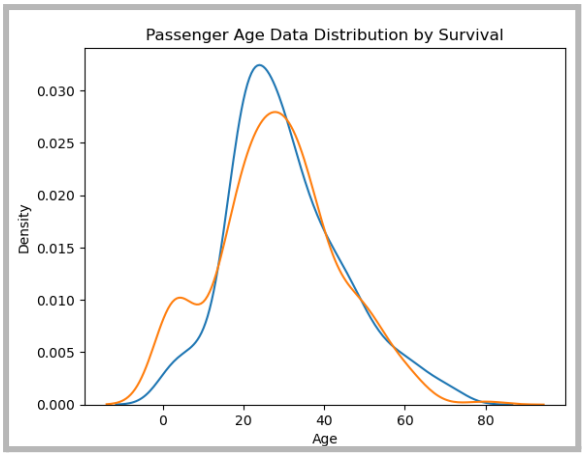
Passenger age data distribution by survival

```
# Vusualizing passenger age data distribution by survival

age_sur= sns.FacetGrid(df,col='Survived')
age_sur.map(sns.distplot,'Age')
```



By analyzing the facetgrid chart indicates a higher survival rate recorded ages between 20 to 40 same as unsurvived passengers.

```
sns.kdeplot(df['Age'][df['Survived']==0], label= 'Not Survived(Dead)')
sns.kdeplot(df['Age'][df['Survived']==1], label= 'Survived(Alive)')
plt.xlabel('Age')
plt.title('Passenger Age Data Distribution by Survival')
```
✓ 0.5s

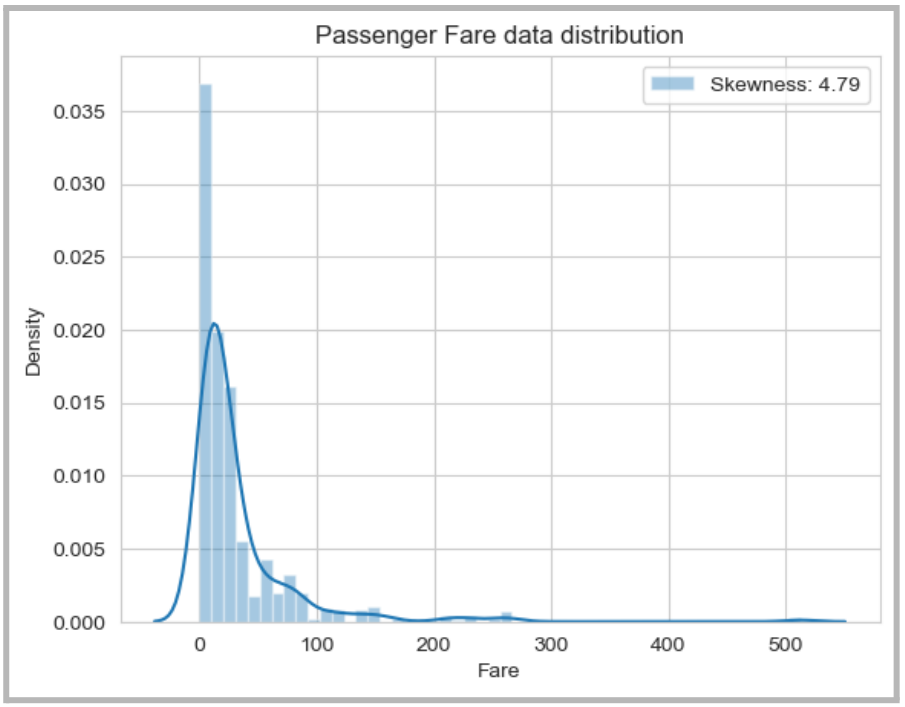


Fare

```
# Visualizing passenger fare data distribution
sns.distplot(df['Fare'],label= 'Skewness: %.2f'%(df['Fare'].skew()))
plt.title('Passenger Fare data distribution')
plt.legend(loc='best')
```

By analyzing this graph we can identify 4.79 higher skewness.

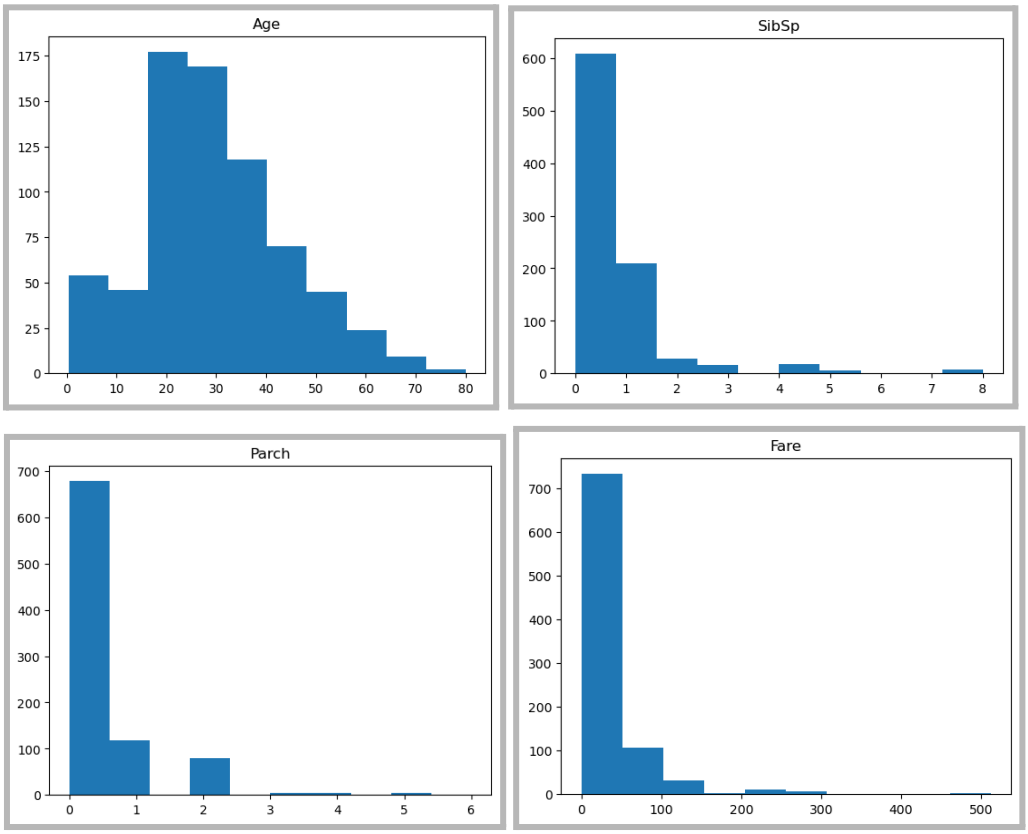
This means that the majority of the values in the distribution are concentrated on the left side of the mean, with a long tail of larger values on the right side.



Here we categorized the dataset into two categories numerical data and categorical data.
Numerical data = ['Age', 'SibSp', 'Parch', 'Fare']
Categorical data = ['Survived', 'Pclass', 'Sex',' Ticket',' Cabin', 'Embarked']
And visualized to get a clear understand the data distribution among each attribute.

Numerical Data

```
#Illustrating histrograms for all the numerical variables of the dataset
for i in df_numerical_data.columns:
    plt.hist(df_numerical_data[i])
    plt.title(i)
    plt.show()
```

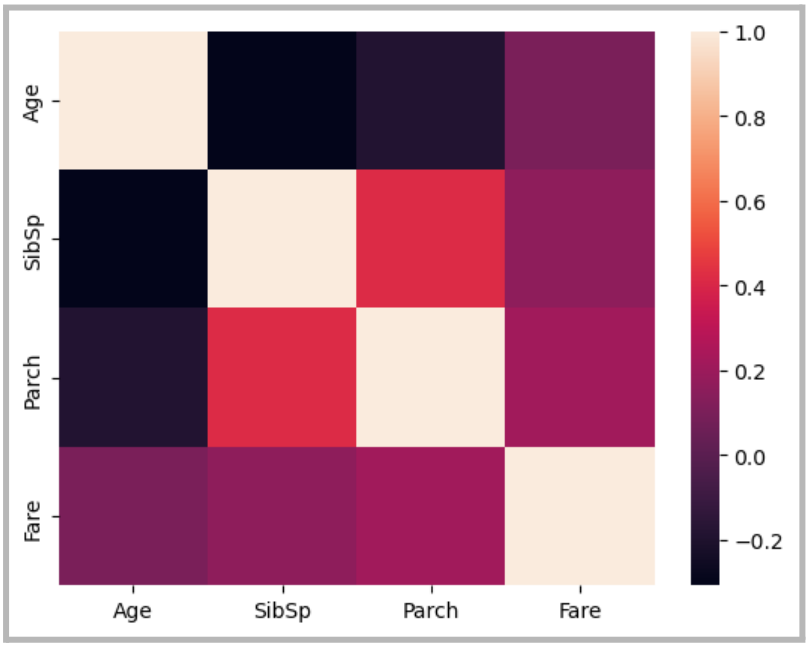


- As we mentioned before Age data has been distributed in a bell curve middle into the age range of 20-40 years. According to the histogram, the highest number of people deviated between the 20-30 age range.
- As we can see on the SibSp graph at least one sibling or spouse has traveled titanic on board with their family members.
- In the Parch graph, we can see at least one parent or child has traveled with their family member.
- In the Fare graph, we can identify that most of the passengers have bought the £0.00 to £150.00 which we can consider the third class and second class passengers' ticket price range.

Analyzing correlation through heatmap

```
#Finding co-relations through vizualizing in a heatmap
print(df_numerical_data.corr())
sns.heatmap(df_numerical_data.corr())
```

```
             Age     SibSp     Parch      Fare
Age     1.000000 -0.308247 -0.189119  0.096067
SibSp  -0.308247  1.000000  0.414838  0.159651
Parch  -0.189119  0.414838  1.000000  0.216225
Fare    0.096067  0.159651  0.216225  1.000000

<AxesSubplot:>
```



In order to analyze the correlation between different variables we initiate a heatmap above.
The value of 0.096067 in the fourth row, the first column indicates that there is a very weak positive correlation between the Age variable and the Fare variable. This means that as the Age variable increases, the Rate variable also increases, but to a very small extent.
Overall, this heat map shows that there is a moderate negative correlation between Age and SibSp, a weak negative correlation between Age and Parch, and a very weak positive correlation between Age and Fare.

Contrast between the number of survival and categorical columns

```
#Contrast the Survival rate within  Age,SibSp, Parch,Fare

pd.pivot_table(df, index='Survived', values=['Age','SibSp','Parch',])
```

Survived	Age	Fare	Parch	SibSp
0	30.626179	22.117887	0.329690	0.553734
1	28.343690	48.395408	0.464912	0.473684

- We initiate a table to analyze the average values of the Age, Fare, Parch, and SibSp variables for individuals who did not survive (labeled "0") and individuals who did survive (labeled "1") the disaster.
- The value of 30.626179 in the first row, the first column indicates that the average age of individuals who did not survive was 30.63 years. The value of 48.395408 in the second row, the second column indicates that the average fare paid by individuals who survived was 48.40.
- The values in the table can be used to compare the characteristics of the two groups and potentially identify patterns or trends. For example, the table shows that individuals who survived tended to be younger and paid a higher average fare than those who did not survive. It also shows that individuals who survived tended to have fewer siblings or spouses (SibSp) and parents or children (Parch) on board the ship.

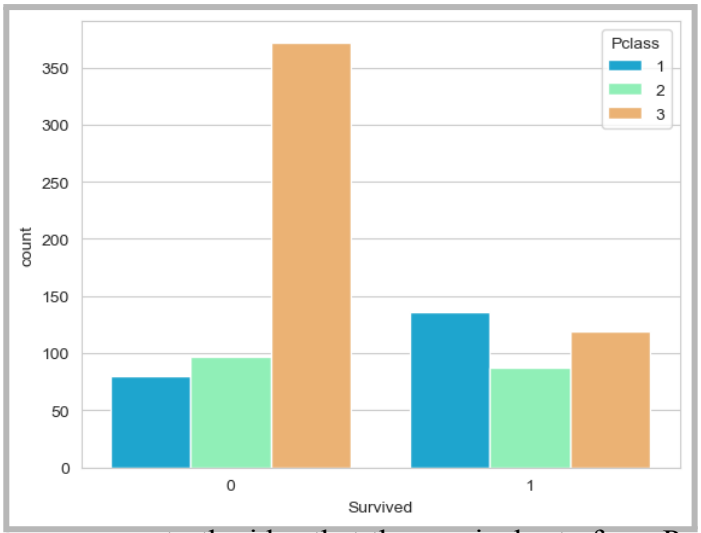
.

Pclass

```
#Contrasting the number of survival and each of the categorical columns of the dataset
print(pd.pivot_table(df,index='Survived', columns= 'Pclass',values='Ticket', aggfunc='count'))
print()
```

```
sns.set_style('whitegrid')
sns.countplot(x='Survived', hue= 'Pclass', data=df, palette='rainbow')
```

Pclass / Survived	1	2	3
0	80	97	372
1	136	87	119



By analyzing the above table and chart we can come to the idea that the survival rate from P classes, first-class survival number is high. And also from the passengers of the third class have a higher number of unsurvived.

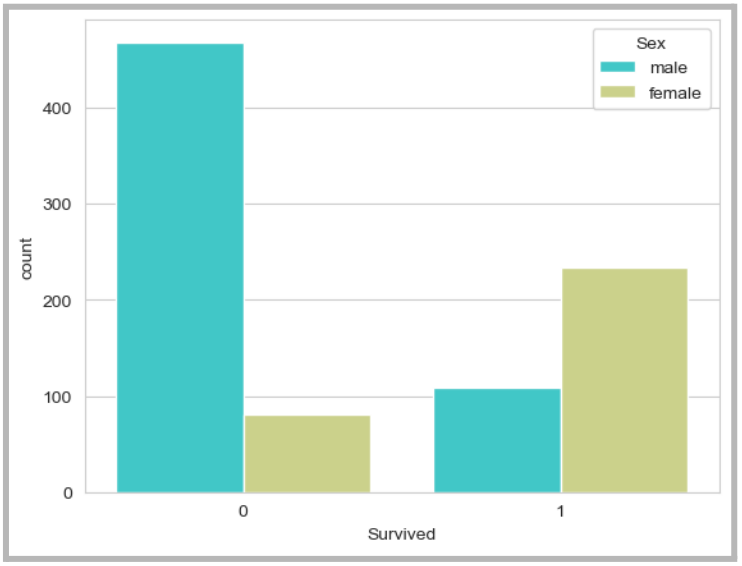
Gender

```
print(pd.pivot_table(df,index='Survived', columns= 'Sex',values='Ticket', aggfunc='count'))
print()
```

```
sns.set_style('whitegrid')
sns.countplot(x='Survived', hue='Sex', data=df, palette='rainbow')
```

0.2s

Sex / Survived	Female	Male
0	81	468
1	233	109



When comparing Gender and number of survival of passengers, 468 male passengers have died when compared to female passengers who have died from the event. And also 233 female passengers have survived when compare to male passengers.

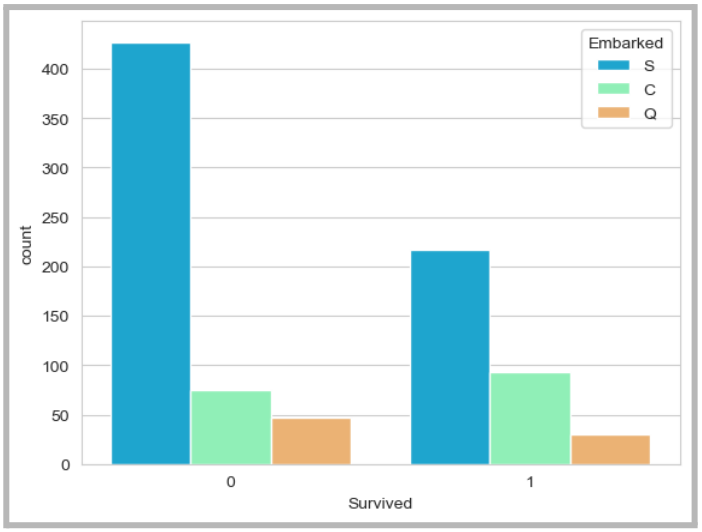
Embarked Location

```
print(pd.pivot_table(df,index='Survived', columns= 'Embarked',values='Ticket', aggfunc='count'))
print()
```

```
sns.set_style('whitegrid')
sns.countplot(x='Survived', hue= 'Embarked', data=df, palette='rainbow')
```

✓ 0.9s

Embarked / Survived	S	C	Q
0	427	75	47
1	217	93	30



When we compared the embarked location and number of Survival of the passengers, 427 passengers died and 217 passengers survived from embarked in Southampton. The lowest survival and deaths recorded from Queenstown were 30 and 47 respectively.

Data Preprocessing

```
#Analysing missing values in the dataset
df.isnull().sum().sort_values(ascending = False)
```

There are 687 missing values in the Cabin column and 177 missing values in the Age column also there were 2 missing values in the Embarked column.

Column	Missing values
Cabin	687
Age	177
Embarked	2
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
SibSp	0
Parch	0
Ticket	0
Fare	0

Engineering the Embarked column

```
#Recognise the most frequent value of Embarked  in the dataset

mode= df['Embarked'].dropna().mode()[0]
mode
```

```
'S'
```

```
#Replace missing values in Embarked with mode value value ('S') in the dataset

df['Embarked'].fillna(mode, inplace= True)
```

Here we have taken the mode of the Embarked column and filled the missing two values with “S” to reduce the missing values into zero.

Engineering the Name Title

```
#Create a new column "Title" in the dataset and extract title

df['Title'] = [name.split(',')[1].split('.')[0].strip() for name in df['Name']]
df[['Name','Title']].head()
```

✓ 0.1s

	Name	Title
0	Braund, Mr. Owen Harris	Mr
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	Mrs
2	Heikkinen, Miss. Laina	Miss
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	Mrs
4	Allen, Mr. William Henry	Mr

We have split the second word from the Name column and extracted the title from the names. After that to get a count of each title we used the value_count function.

Name Title	Count
Mr	517
Miss	182
Mrs	125
Master	40
Dr	7
Rev	6
Mlle	2
Major	2
Col	2
the Countess	1
Capt	1
Ms	1
Sir	1
Lady	1
Mme	1
Don	1
Jonkheer	1

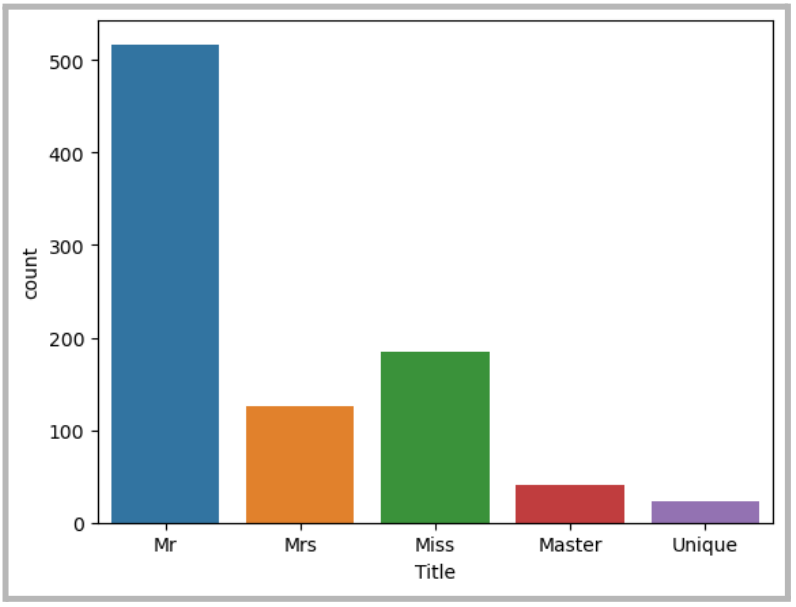
```
#Value count of each title

df['Title'].value_counts()
```

```
#Calculating the Mean of Survived by name title

df[['Title','Survived']].groupby(['Title'], as_index=False).mean().sort_values(by= 'Survived', ascending = False)
```

```
#Summerize the titles except Mr /Miss/ Mrs / Master in to category "Rare"

df['Title'] = df['Title'].replace(['Dr','Rev','Col','Major','Lady','Jonkheer',
                                   'Don','Capt','the Countess','Sir','Dona'],'Rare')


#Summerize the titles Mlle/ Ms in to 'Miss'
df['Title'] = df['Title'].replace(['Mlle','Ms'], 'Miss')


#Summerize the titles Mme in to 'Mrs'
df['Title'] = df['Title'].replace('Mme', 'Mrs')


sns.countplot(df['Title'])
```

We have engineered these titles ['Dr','Rev','Col','Major','Lady','Jonkheer','Don','Capt','the Countess','Sir','Dona'] into a new category called “Unique”.
Then we replaced a new category called ‘Miss’ including ['Mlle','Ms'] titles and replaced 'Mme', 'Mrs' in to “Mrs” category.

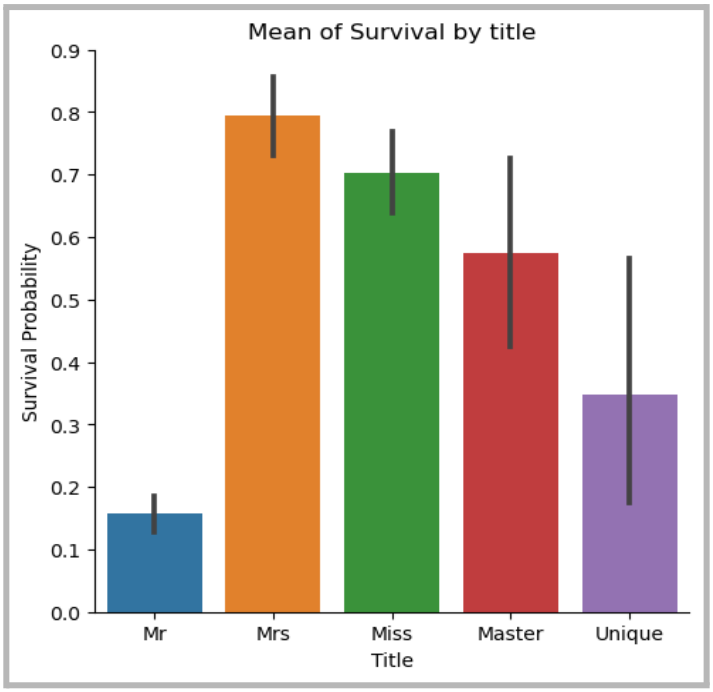
This count-plot interprets the count of each title. By analyzing this chart, Mr title Is the highest count and Miss title is the second highest count.



Analysing Survival Probability Rate with Titles

```
sns.factorplot(x='Title', y='Survived', data = df, kind = 'bar')
plt.ylabel('Survival Probability')
plt.title('Mean of Survival by title')
```

Title	Survived(Mean)
Mrs	0.793651
Miss	0.702703
Master	0.575000
Unique	0.347826
Mr	0.156673



By analyzing the table and chart we can come to a resolution that individuals with the title "Mrs" had a much higher survival rate compared to individuals with the title "Mr". Miss title has the second highest rate of survival rate and the Master title has the third highest survival rate. Mostly women and children have the highest survival rate.