INTRODUCTION TO DATA SCIENCE

PROJECT REPORT

Exploratory Data Analysis on Titanic Disaster

Submitted by -
Pyushi Paliwal (16ucs146)
Tanishqa Jain (16ucs194)
Saloni Singh (16ucs167)
Yashashwi Siddharth (16ucc105)

Introduction:

RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early hours of 15 April 1912, after colliding with an iceberg during its maiden voyage from Southampton to New York City, killing 1502 out of 2224 passengers and crew. Translated 32% survival rate, it is one of the deadliest maritime disasters in modern history.

Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others. Women, children, and the upper-class being some of the groups. Our project aims to further explore the factors that played a deciding role in passenger's survival.

Problem Statement:

Given the data set of samples listing the passengers that did/did not survive the Titanic disaster, determine the factors that played a deciding role in one's survival. Perform the required Data Analysis and pre-process the data for ML classification.

About the dataset-

The dataset has been collected from **Kaggle**. The dataset consist of both training data as well as test data. The training data contains 891 rows and 12 columns and the test data consist of 418 rows and 11 columns (The label class-"Survived" is not included in test data set).

Total samples are 891 or 40% of the actual number of passengers on board the Titanic (2,224).

Features available in the dataset are-

1) **Passenger-Id:** Id of every passenger.
2) **Survived:** This feature have value 0 and 1. 0 for not survived and 1 for survived.
3) **Pclass:** There are 3 classes of passengers. Class1, Class2 and Class3.
4) **Name:** Name of passenger.
5) **Sex:** Gender of passenger.
6) **Age:** Age of passenger.
7) **SibSp:** No. of siblings, spouse aboard.
8) **Parch:** Parents or children aboard.
9) **Ticket:** Ticket no of passenger.
10) **Fare:** Indicating the fare.
11) **Cabin:** The cabin no. of passenger.
12) **Embarked:** The entry port to embark on the journey.

1. Import Data-

```
In [1]: import pandas as pd
        import numpy as np

In [2]: train = pd.read_csv('C:\\Users\\hp\\Desktop\\train.csv')
        test = pd.read_csv('C:\\Users\\hp\\Desktop\\test.csv')

In [3]: print(train.columns.values)
        print(test.columns.values)

        ['PassengerId' 'Survived' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch'
         'Ticket' 'Fare' 'Cabin' 'Embarked']
        ['PassengerId' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch' 'Ticket' 'Fare'
         'Cabin' 'Embarked']
```

2. Describing Data-

```
In [4]: #Top few rows of data
        train.head()
```

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

The different types of features in our dataset are-

A. Qualitative Features-> Categorical : Survived, Sex, and Embarked. **Ordinal:** Pclass.

B. Quantitative Features-> Continuous: Age, Fare. **Discrete:** Sibsp, Parch.

C. Mixed data type Features-> Ticket, Cabin.

```
In [5]: #Retrieve data info
        train.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 891 entries, 0 to 890
        Data columns (total 12 columns):
        PassengerId    891 non-null int64
        Survived       891 non-null int64
        Pclass         891 non-null int64
        Name           891 non-null object
        Sex            891 non-null object
        Age            714 non-null float64
        SibSp          891 non-null int64
        Parch          891 non-null int64
        Ticket         891 non-null object
        Fare           891 non-null float64
        Cabin          204 non-null object
        Embarked       889 non-null object
        dtypes: float64(2), int64(5), object(5)
        memory usage: 83.6+ KB
```

Features containing null/blank values: Cabin > Age > Embarked, with cabin having the highest no. of null values.

Data types: int-5, float-2, Object(string)-5.

```
In [6]: test.info()

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

Features containing null/blank values: Cabin > Age>Fare

Data types: int-4, float-2, Object(string)-5.

3. Descriptive Statistics:

To learn the distribution of features across the training data set, we use descriptive analysis-

3.1 For Numerical features-

```
In [7]: # Descriptive statistics on numerical data
        train.describe(include=['number'])
```

Out[7]:

	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

- 38% samples survive, a decent representation for the actual survival rate i.e. 32%.
- More than 50% passengers travelled in Class 3(refers to Pclass).
- Most passengers (> 75%) did not travel with parents or children.
- Nearly 70% passengers did not travel with their spouse or siblings.
- Less than 1% passengers paid as high as $512.(Using Chebyshev's inequality)
- Less than 1% of the aboard passengers aged between 65-80 (Using Chebyshev's inequality).

3.1 For Object type features-

In [8]:
```
# Descriptive statistics on "object" data
train.describe(include=['object'])
```

Out[8]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Tornquist, Mr. William Henry	male	CA. 2343	C23 C25 C27	S
freq	1	577	7	4	644

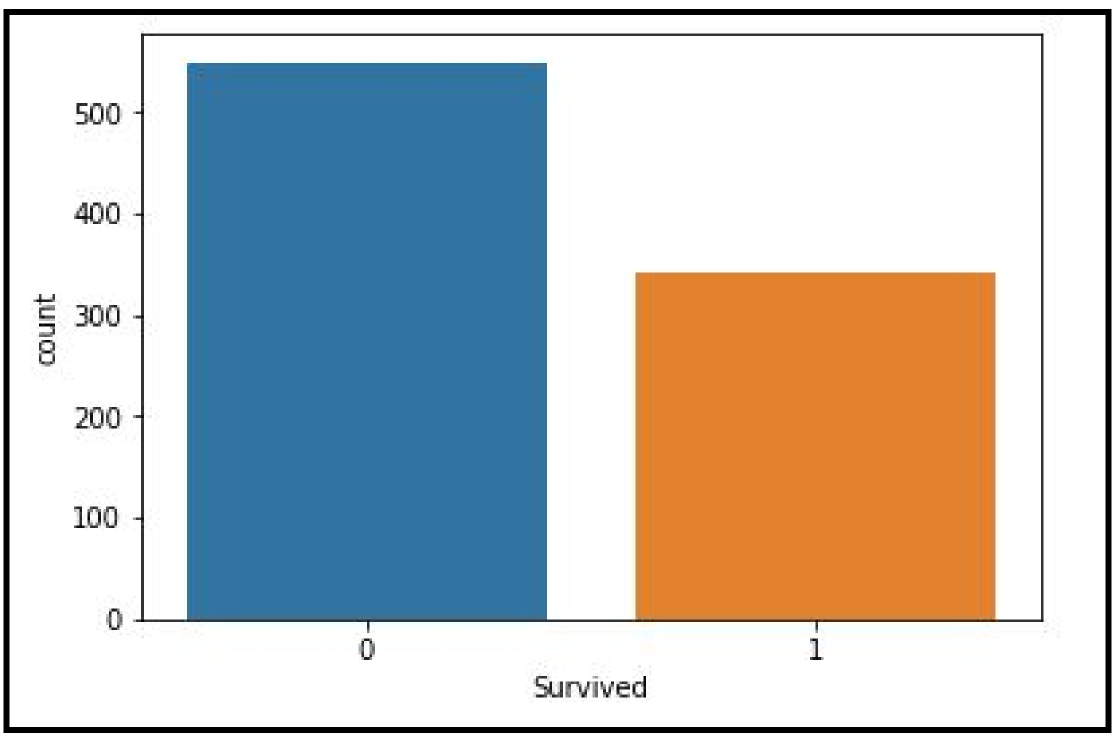
- All the passengers have unique names (as count=unique=891)
- Sex variable has two possible values with 65% male (top=male, (577/891)*100 = 64.75).
- Cabin values are not unique in nature as several passengers shared a cabin.
- Embarked takes three possible values. S port being used by 72% of the passengers((644/891)*100)
- Ticket no. is not unique in nature with 210 duplicate values.

4. Removing Irrelevant features:

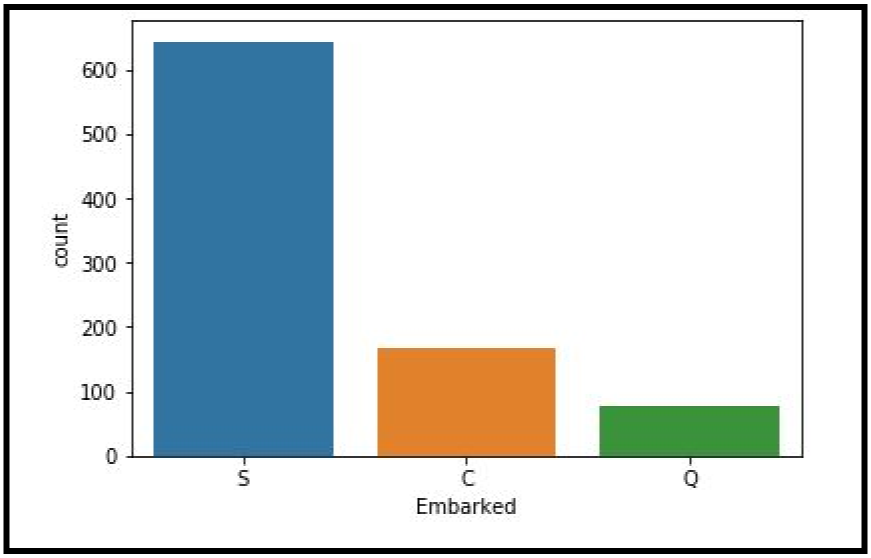
PassengerId, Name may be dropped from training dataset as there seems to be no correlation between these features and survival of the individual. **Ticket** feature may also be dropped as it contains around 22% duplicate values with seemingly no contribution to survival. **Cabin** feature may be dropped as it is highly incomplete both in training and test dataset.

4. Data Visualization:

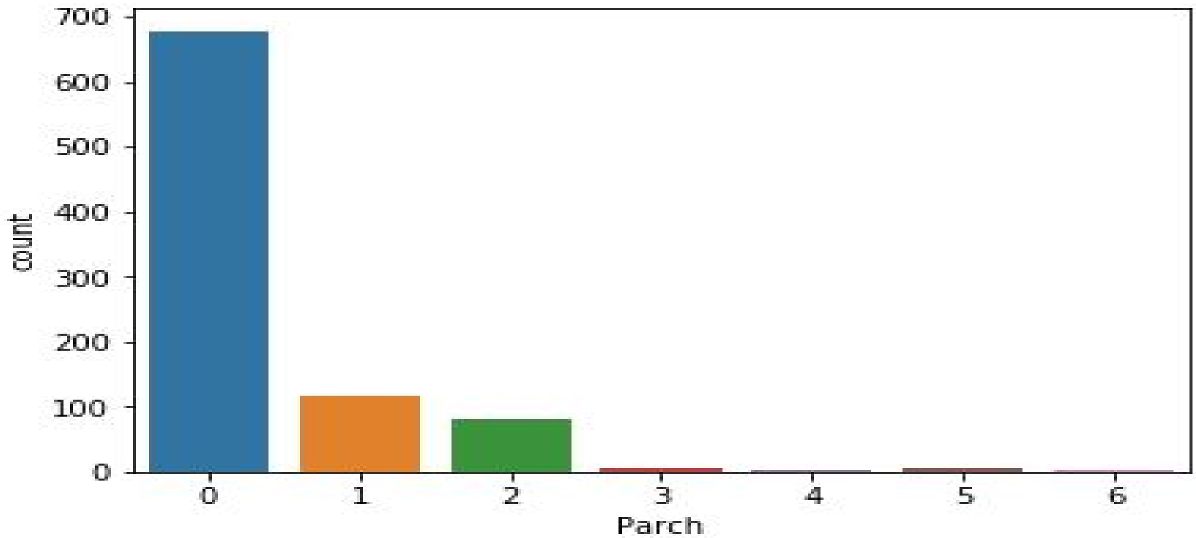
4.1 Univariate Analysis:



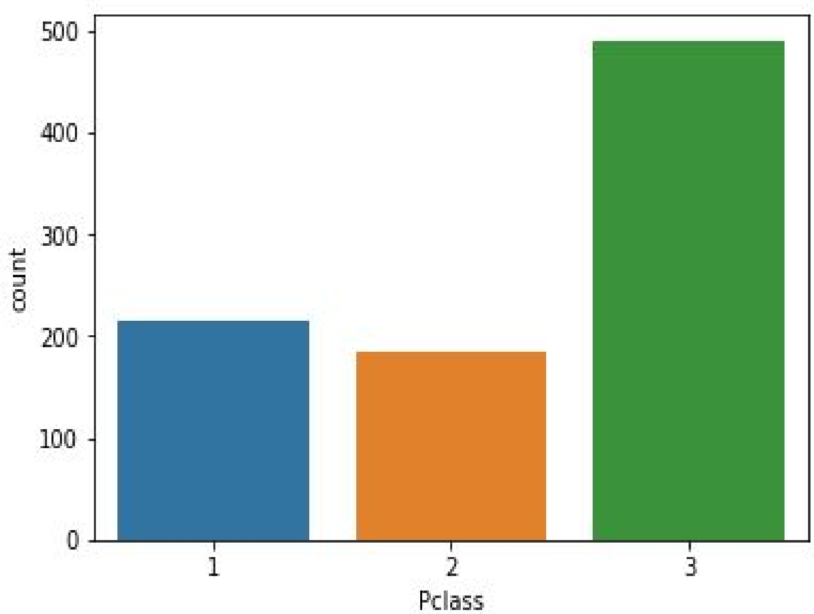
As it is evident from the above plot, only 38% of the passengers survived, whereas a majority 62% the passenger did not survive the disaster.



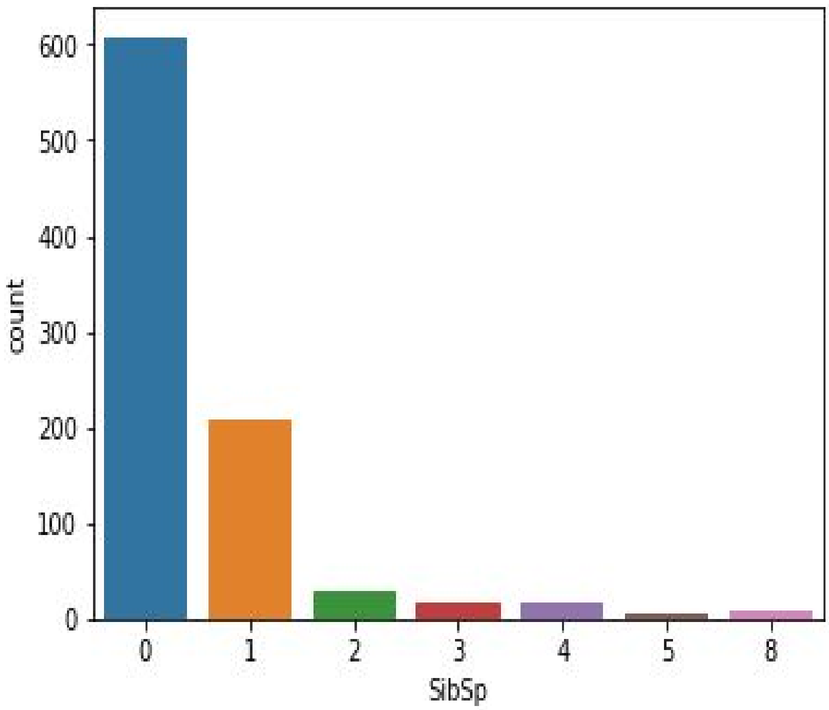
Most of the passengers embarked on the ship from ‘S’ port followed by ‘C’ and least number of people were from ‘Q’ port.



Most Passengers were travelling without their parents or children, some had 1 or 2 of them. Very few passengers had more than 2 parents or children.

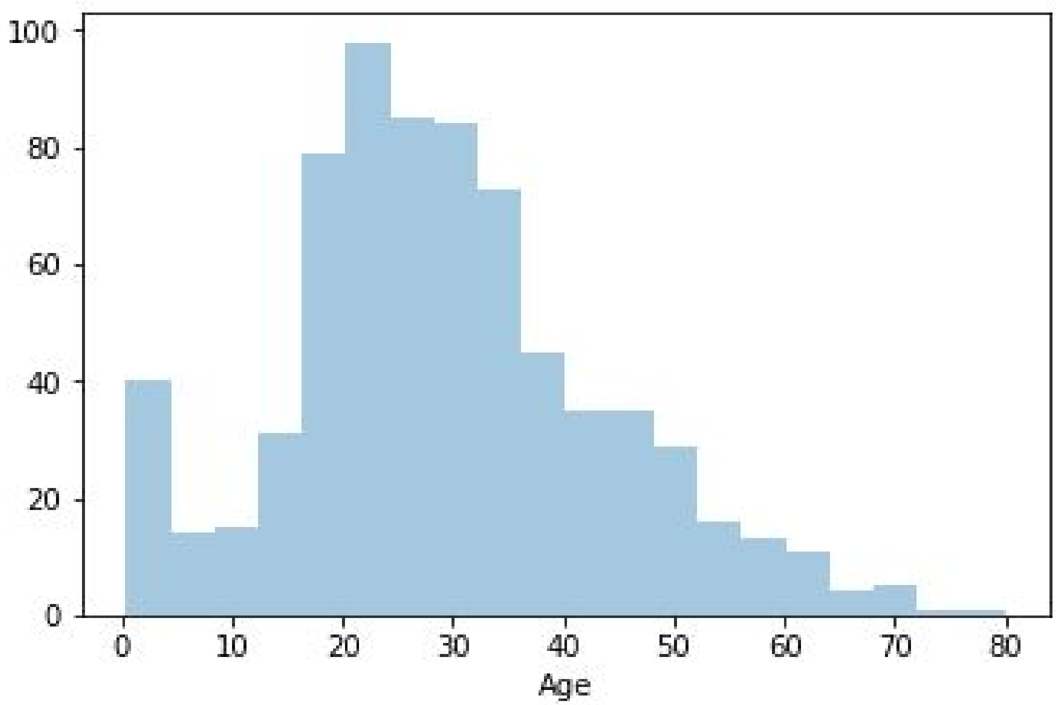


Most Passengers were travelling in Passenger class 3, then in class 1 and least in class 2.

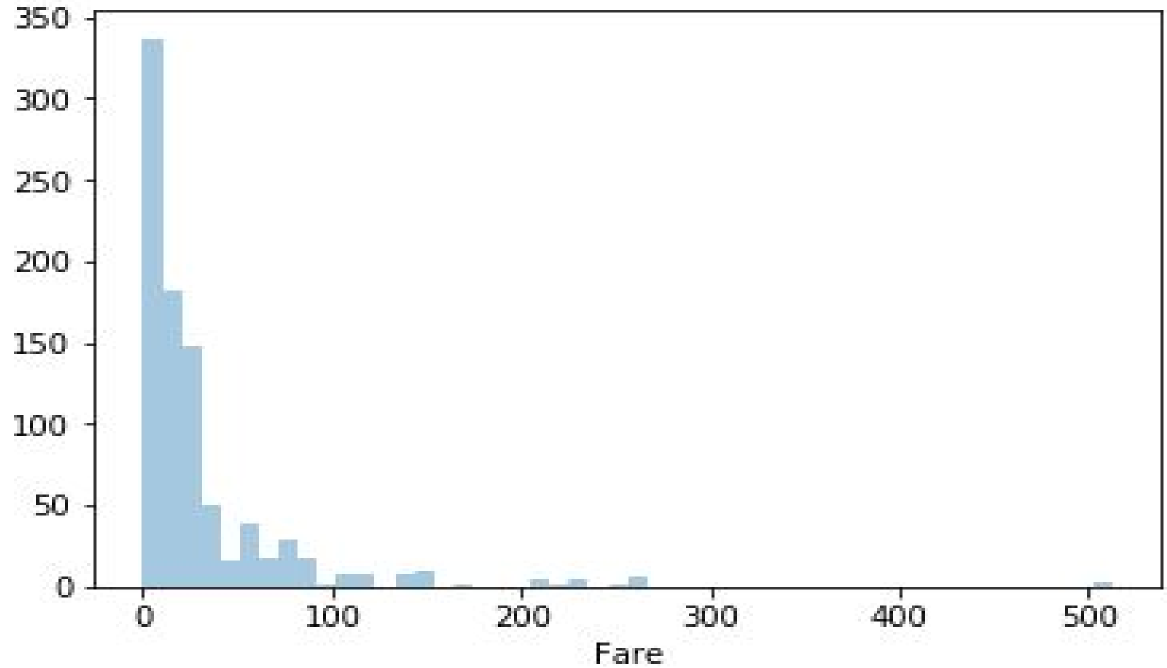


Most Passengers had no sibling or spouse aboard with them on the ship.

For numerical features- “Age”(dropping the NAN values), “Fare”

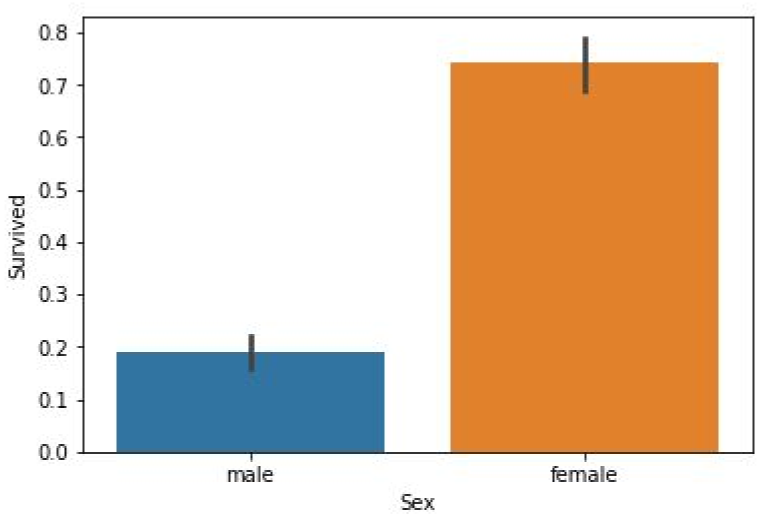


Maximum people are in the age range of 25-35 years. Few elderly passengers (<1%) within age range 65-80 were aboard.

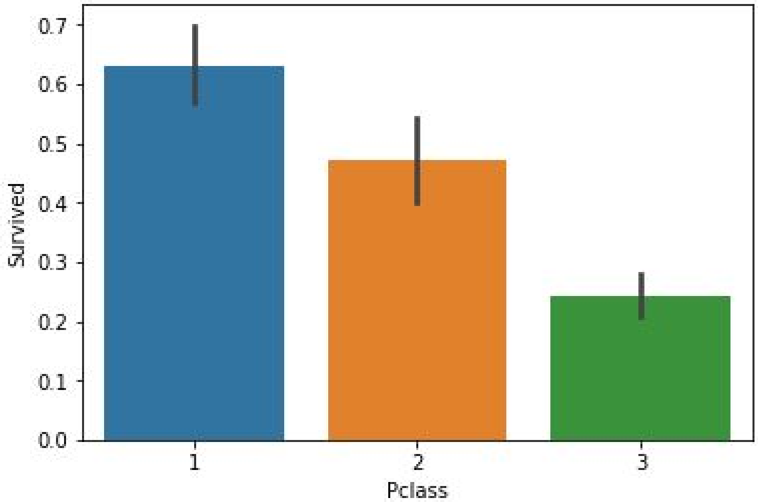


The Fares paid by the passengers displayed significant variation with few passengers (<1%) paying as high as $512 and most passengers paying relatively much lower.

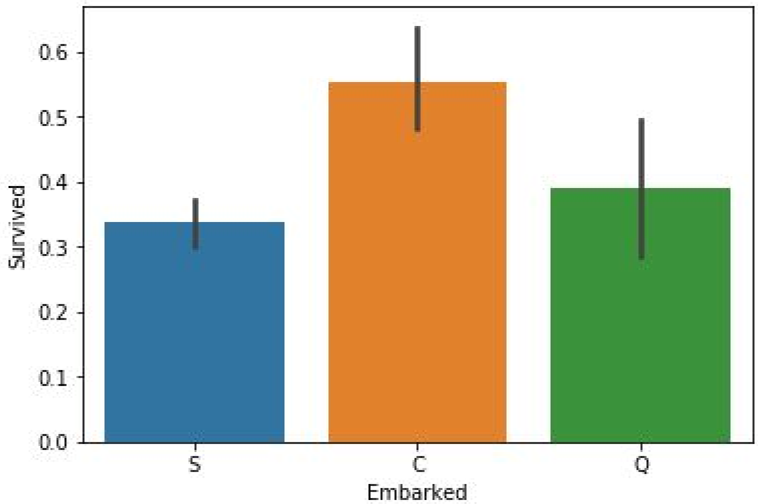
4.2 Bivariate Analysis:



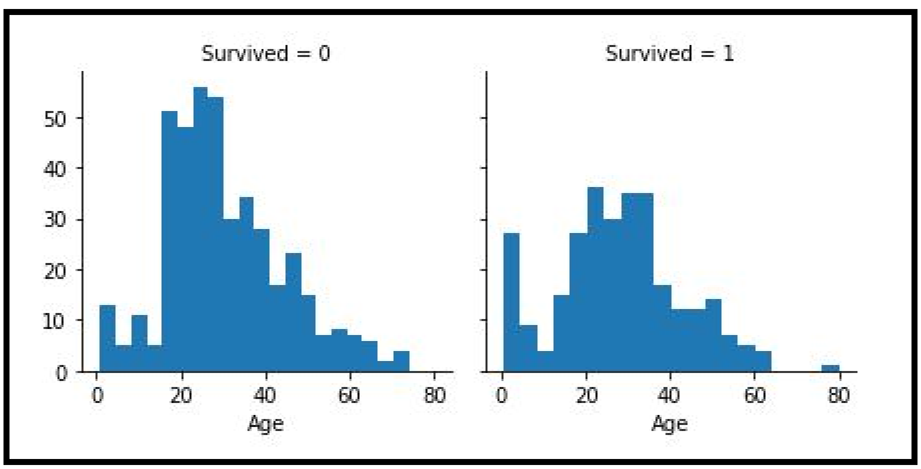
It is evident that average survival rates of male is around 20% whereas in case of a female, chances of survival is around 75%. Therefore the feature 'Sex' is highly correlated with survival.



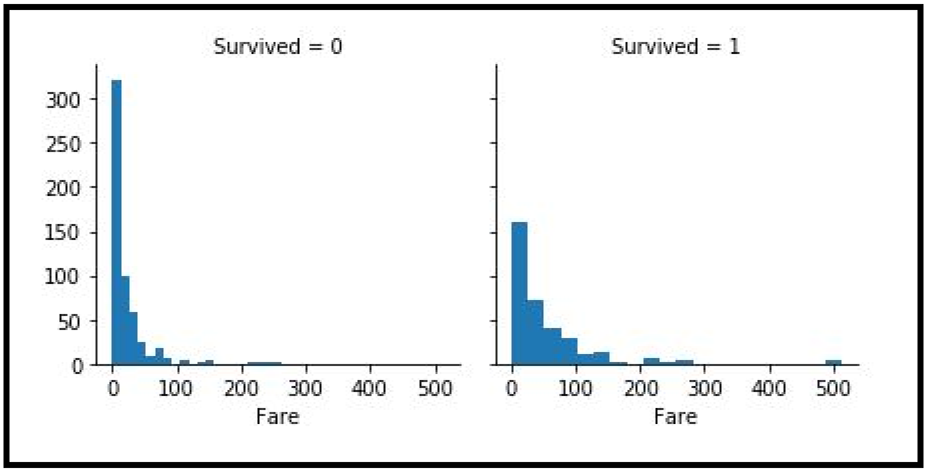
There is a for sure a clear relationship between Pclass and the survival if we consider the plot above. Passengers on Pclass 1 on an average had a better survival rate of approx 60% whereas passengers on Pclass 3 had the worst survival rate of approx 22%



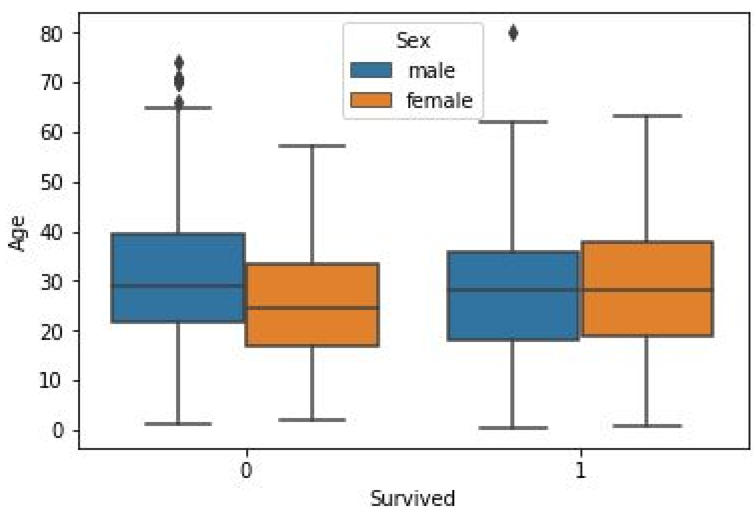
It is clearly visible that passengers from port C had significantly better chances of survival.

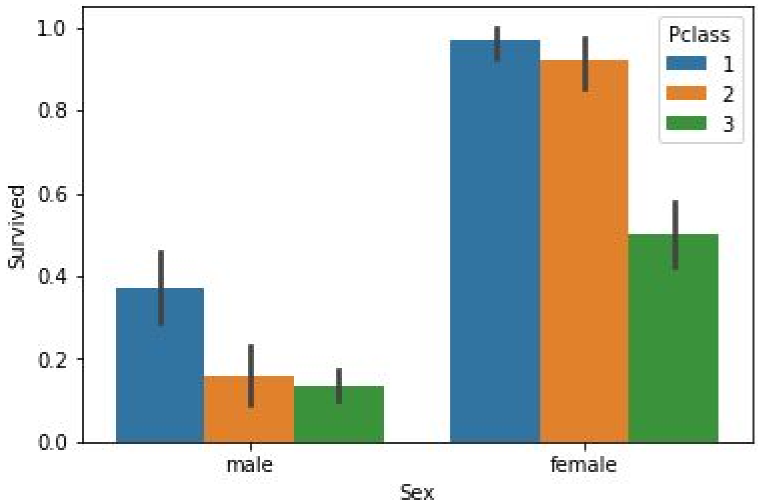


- Large number of 15-25 year olds did not survive.
- Passengers with age less than 5 had high survival rate.
- The passenger with highest age of 80 survived.

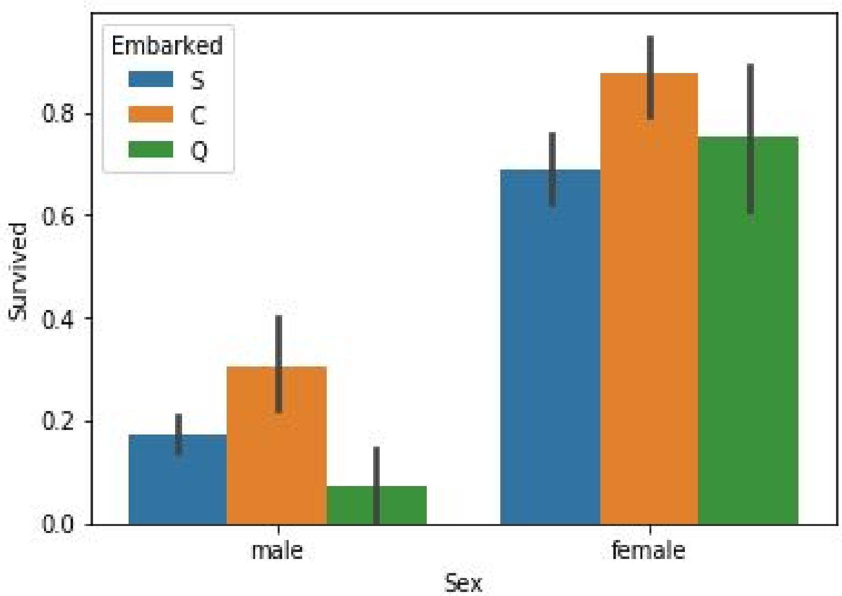


There is a marginal relationship between the fare and survival rate.

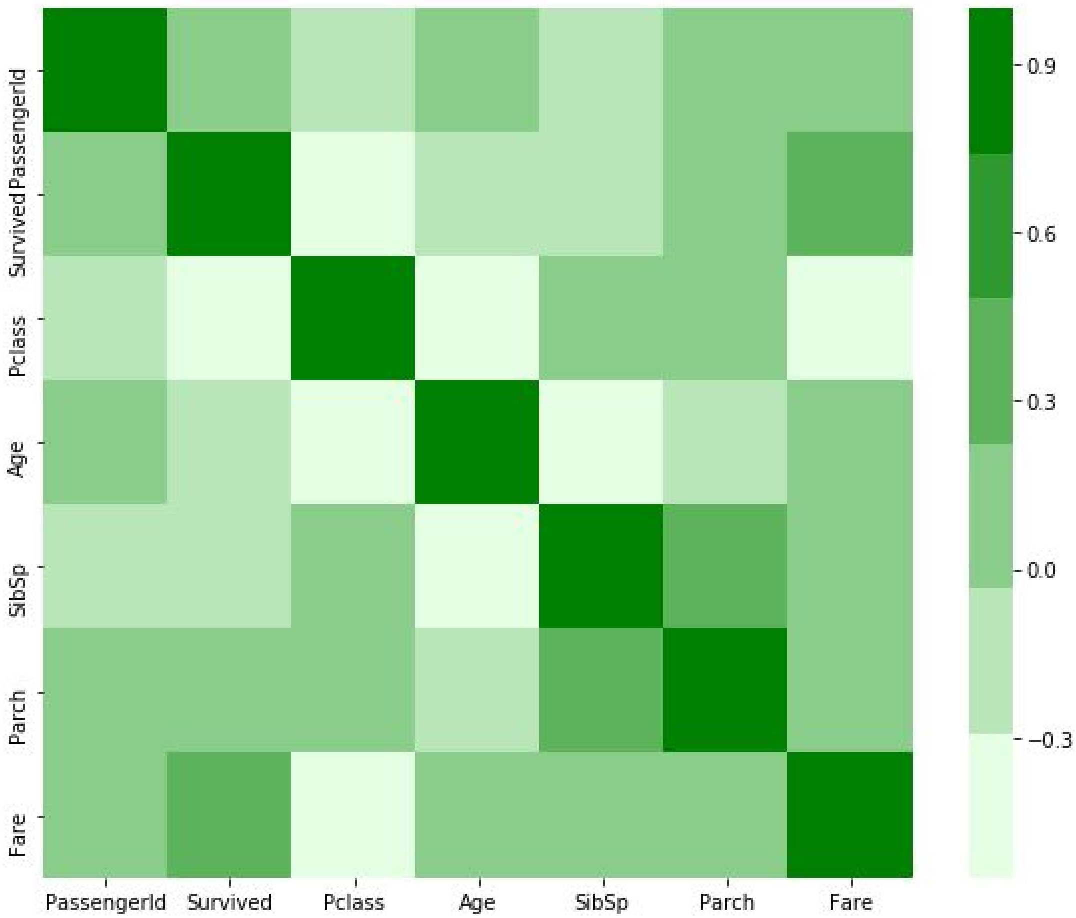




In the above plot, we can see that the average number of survivals of male and female in each class. From the plot we can understand that more number of females survived than males. In both males and females more number of survivals are from first class.



The above plot gives an idea of no. of average passengers of both sex were able to survive who embarked from different ports.



The above heat plot indicates Pclass and survival are strongly correlated and so are age and class of the passenger. Passenger id seems to be un-correlated to survival, verifying our assumption of independence of the two. Survival seems to have little dependence on Parch & Sibsp as there is no strong correlation between them.

Conclusion-Que 1- What were the deciding factors in the survival of a passenger on Titanic?

Answer: The contributing factors that came out from above analysis are-

i) Sex - Sex was the feature that was highly related to the chances of survival. In the tragedy, women had a survival rate of around 75% while men had chances below 20%

ii) Pclass - Pclass was negatively correlated with the chance of survival as titanic sank from the bow of the ship where third class rooms located. Passenger on Pclass 1 had highest chances of surviving.

iii) Age - Age also had significant relationship with Survival. Infants (Age <=4) had high survival rate. Oldest passengers (Age = 80) survived.Large number of 15-25 year olds did not survive. Most passengers are in 15-35 age range.

iv) Fare - There is a positive correlation between Fare and Survived.

Que 2 - Which features had little or zero contribution to the chances of survival?

Answer- PassengerId, Name, SibSp, Parch, Ticket, Cabin, Embarked (No direct correlation).

Data Preprocessing

Removing Irrelevant features:

Name may be dropped from training dataset as there seems to be no correlation between these features and survival of the individual. **Ticket** feature may also be dropped as it contains around 22% duplicate values with seemingly no contribution to survival. **Cabin** feature may be dropped as it is highly incomplete both in training and test dataset.

BEFORE:

	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

AFTER:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
5	6	0	3	male	NaN	0	0	8.4583	Q

Dealing with missing values:

Based on our data analysis, features “Age” and “Embarked” are the attributes with missing values which seem to have correlation with the class label (‘Survived’), thus completing them is important. Since machine learning models work best when data has no missing values, thus we take the following steps to deal with them:

1. FOR AGE : Both for male and female passengers we fill the missing values with their respective median age

Before-

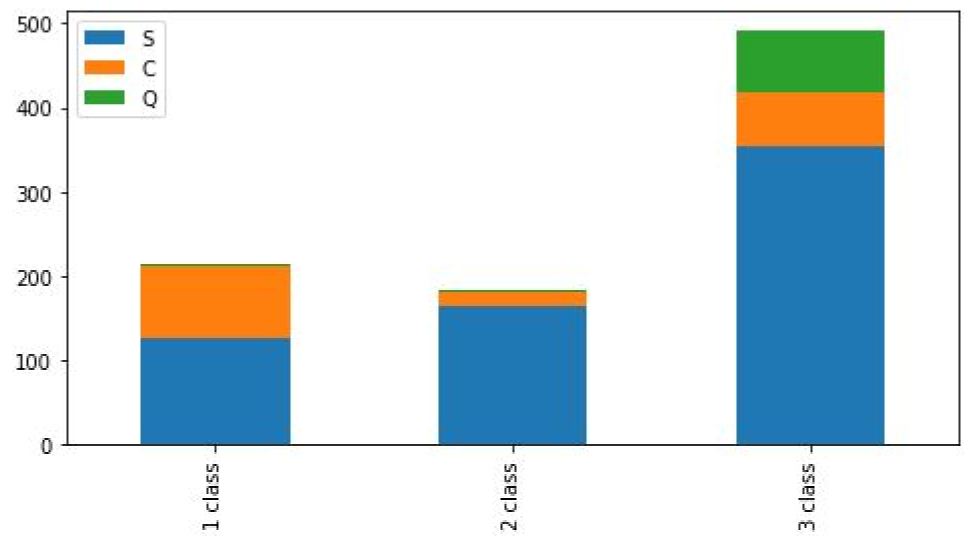
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
5	6	0	3	male	NaN	0	0	8.4583	Q

After-

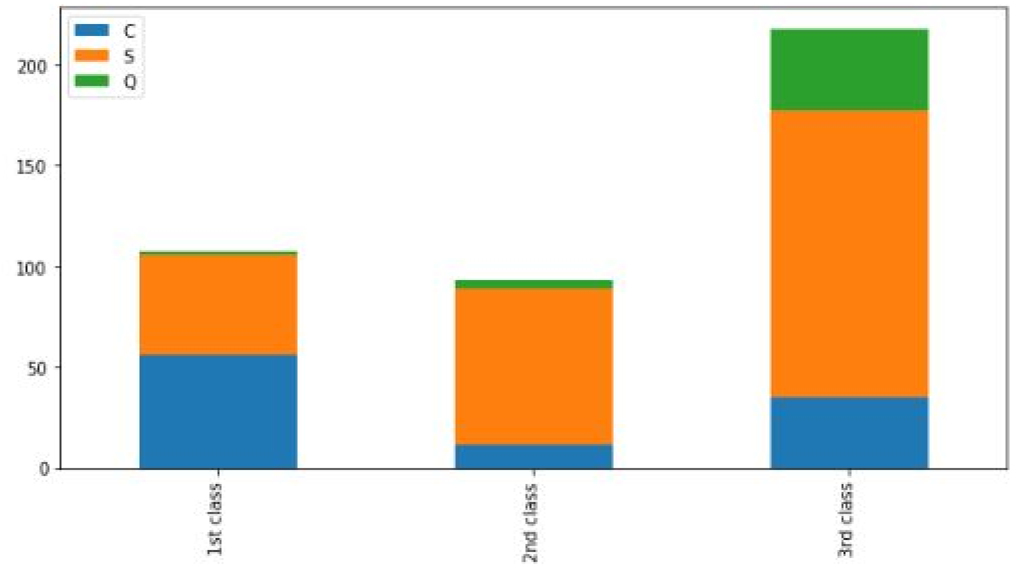
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
5	6	0	3	male	29.0	0	0	8.4583	Q

2. FOR EMBARKED : As can be observed by the univariate graph of feature embarked, the “S” port has the highest count of passengers. The plot below shows that in all classes in both training and test data, we can see that more than 50 percent have S embark, thus we fill in the missing values with "S".

TRAINING SET:



TEST SET:



After embarked missing value correction:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S

Converting Features:

A. **CATEGORICAL TO NUMERICAL:** Since various algorithms used for model training, require the categorical features to be converted to their equivalent numerical ones. So here we convert the text categorical features namely - “Sex” and “Embarked” to their numerical equivalents through the following mapping :

1. **For Embarked** :Mapping = {"S": 0, "Q": 1, "C": 2}

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	0
1	2	1	1	female	38.0	1	0	71.2833	2
2	3	1	3	female	26.0	0	0	7.9250	0
3	4	1	1	female	35.0	1	0	53.1000	0
4	5	0	3	male	35.0	0	0	8.0500	0

2. **For Sex** : Sex_map = {"male" : 1, "female": 0}

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	0
1	2	1	1	0	38.0	1	0	71.2833	2
2	3	1	3	0	26.0	0	0	7.9250	0
3	4	1	1	0	35.0	1	0	53.1000	0
4	5	0	3	1	35.0	0	0	8.0500	0

B. **CONTINUOUS TO DISCRETE :** As observed from our analysis, specific age bands have high correlation with survival, so here we convert the continuous age feature into discrete age bands. Similarly, fare -> discrete fare bands.

Age Bands:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	1.0	1	0	7.2500	0
1	2	1	1	0	3.0	1	0	71.2833	2
2	3	1	3	0	2.0	0	0	7.9250	0
3	4	1	1	0	2.0	1	0	53.1000	0
4	5	0	3	1	2.0	0	0	8.0500	0
5	6	0	3	1	2.0	0	0	8.4583	1

Fare Bands :

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	1.0	1	0	0.0	0
1	2	1	1	0	3.0	1	0	2.0	2
2	3	1	3	0	2.0	0	0	0.0	0
3	4	1	1	0	2.0	1	0	2.0	0
4	5	0	3	1	2.0	0	0	0.0	0
5	6	0	3	1	2.0	0	0	0.0	1

Creating features:

New features are created that follow correlation, conversion and completeness goals. Features SibSp and Parch both when combined form passenger's family count, so in order to reduce these features for better analysis and training we form a new feature "hasFamily" -> "1"(if has a family), "0"(if doesn't have a family).

FINAL PRE-PROCESSED DATA SET:

	PassengerId	Survived	Pclass	Sex	Age	Fare	Embarked	hasFamily
0	1	0	3	1	1.0	0.0	0	1
1	2	1	1	0	3.0	2.0	2	1
2	3	1	3	0	2.0	0.0	0	0
3	4	1	1	0	2.0	2.0	0	1
4	5	0	3	1	2.0	0.0	0	0
5	6	0	3	1	2.0	0.0	1	0
6	7	0	1	1	3.0	2.0	0	0
7	8	0	3	1	0.0	1.0	0	1
8	9	1	3	0	2.0	0.0	0	1
9	10	1	2	0	0.0	2.0	2	1
10	11	1	3	0	0.0	1.0	0	1
11	12	1	1	0	3.0	1.0	0	0
12	13	0	3	1	1.0	0.0	0	0
13	14	0	3	1	3.0	2.0	0	1
14	15	0	3	0	0.0	0.0	0	0

The Data analysis done helps in choosing the appropriate features, this processed data set can finally be used to train the appropriate machine learning model for further classification.

Kindly look below for code files of Data Analysis & Data pre-processing.

IDS_Project_Code

November 17, 2018

In [1]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
train = pd.read_csv('C:\\Users\\paliw\\Downloads\\train_titanic.csv')
test = pd.read_csv('C:\\Users\\paliw\\Downloads\\test_titanic.csv')
```

In [4]:
```
print(train.columns.values)
print(test.columns.values)
```

```
['PassengerId' 'Survived' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch'
 'Ticket' 'Fare' 'Cabin' 'Embarked']
['PassengerId' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch' 'Ticket' 'Fare'
 'Cabin' 'Embarked']
```

In [5]:
```
#top few rows of data
train.head()
```

Out[5]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         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

   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
```

In [6]:
```
#Retrieve data info
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

In [7]:
```
test.info()
```

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

In [8]:
```
# Descriptive statistics on numerical data
train.describe(include=['number'])
```

Out[8]:

	PassengerId	Survived	Pclass	Age	SibSp \
count	891.000000	891.000000	891.000000	714.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008
std	257.353842	0.486592	0.836071	14.526497	1.102743

```
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

In [9]:
```python
# Descriptive statistics on "object" data
train.describe(include=['object'])
```

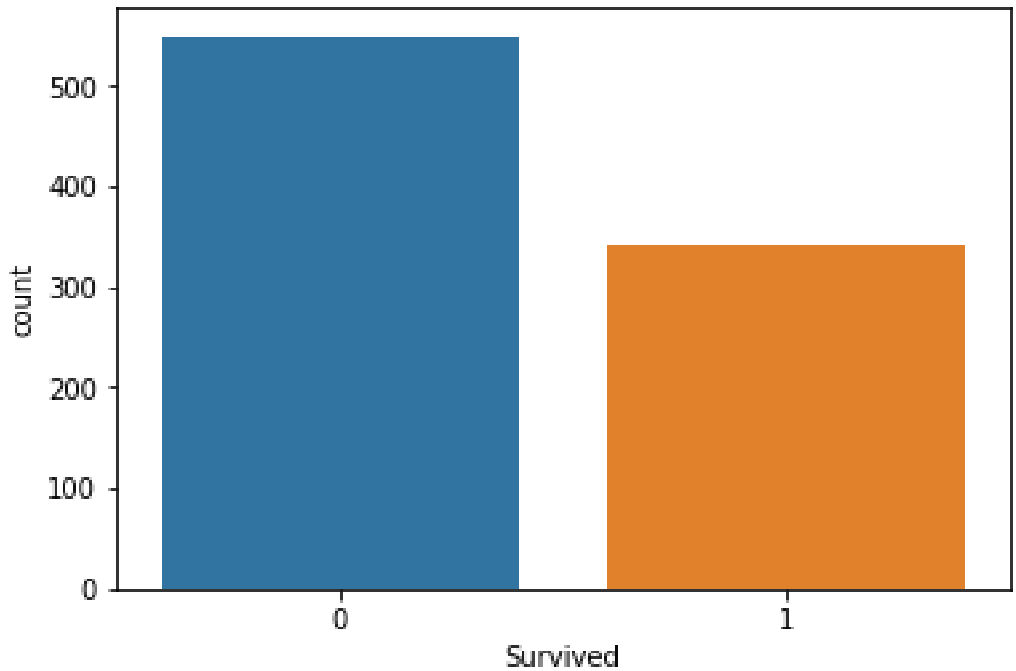
Out[9]:
```
                                                Name   Sex    Ticket  \
count                                            891   891       891
unique                                           891     2       681
top     Taylor, Mrs. Elmer Zebley (Juliet Cummins Wright)  male  CA. 2343
freq                                               1   577         7

              Cabin Embarked
count           204      889
unique          147        3
top     C23 C25 C27        S
freq              4      644
```

In [11]:
```python
# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import Image, display
%matplotlib inline
```

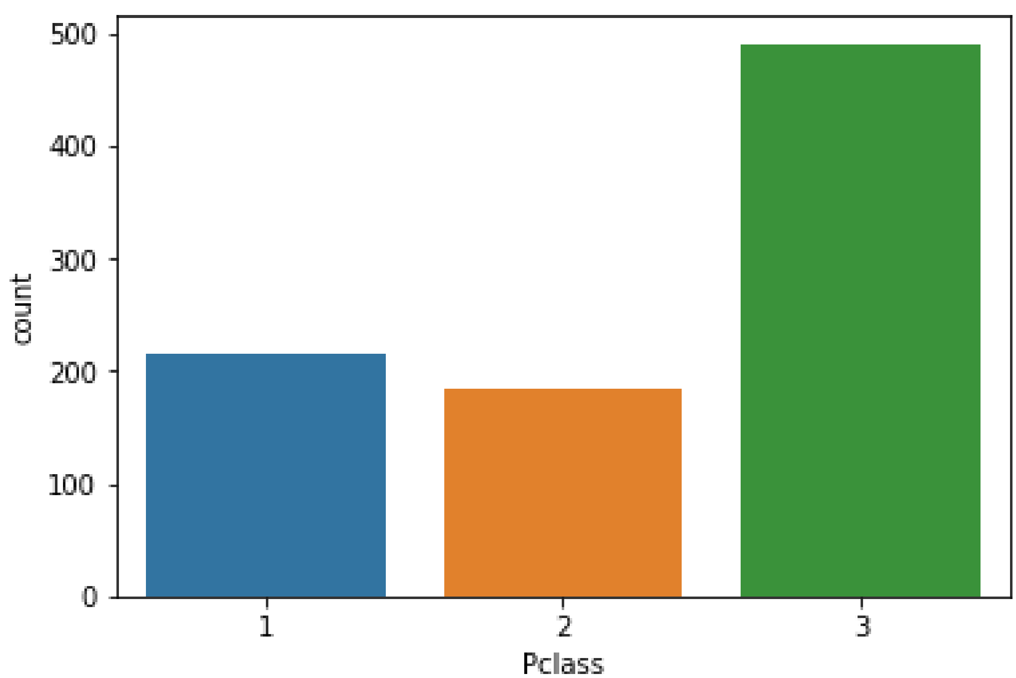
In [12]:
```python
# Univariate Analysis
# For categorical/ordinal features
sns.countplot('Survived',data=train)
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x24756f549b0>



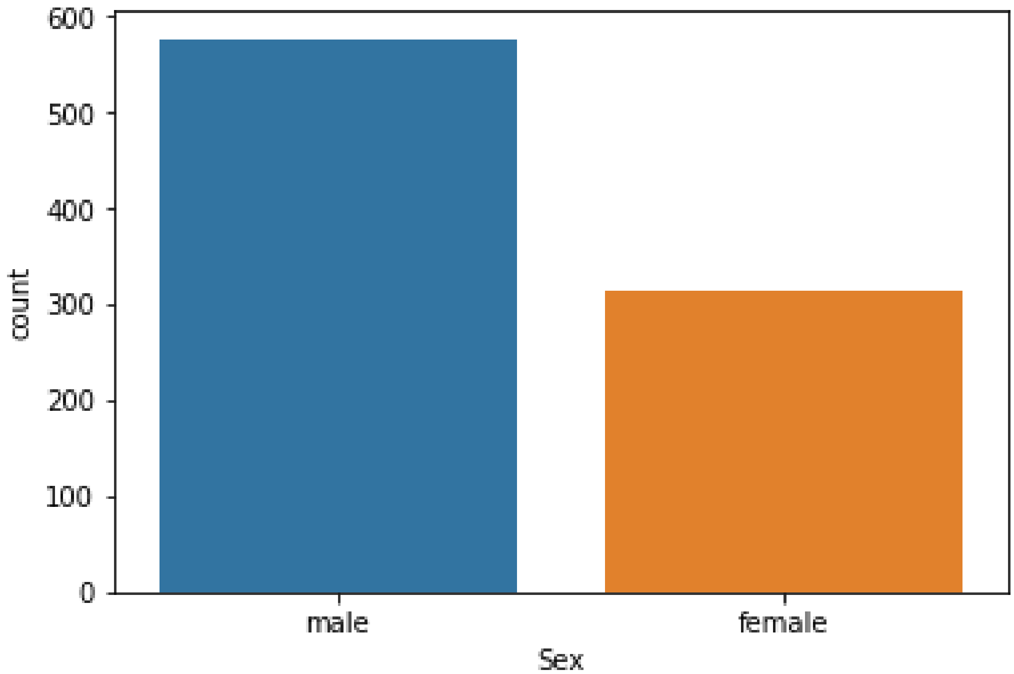

```
In [13]: sns.countplot('Pclass',data=train)
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x24758251ef0>
```



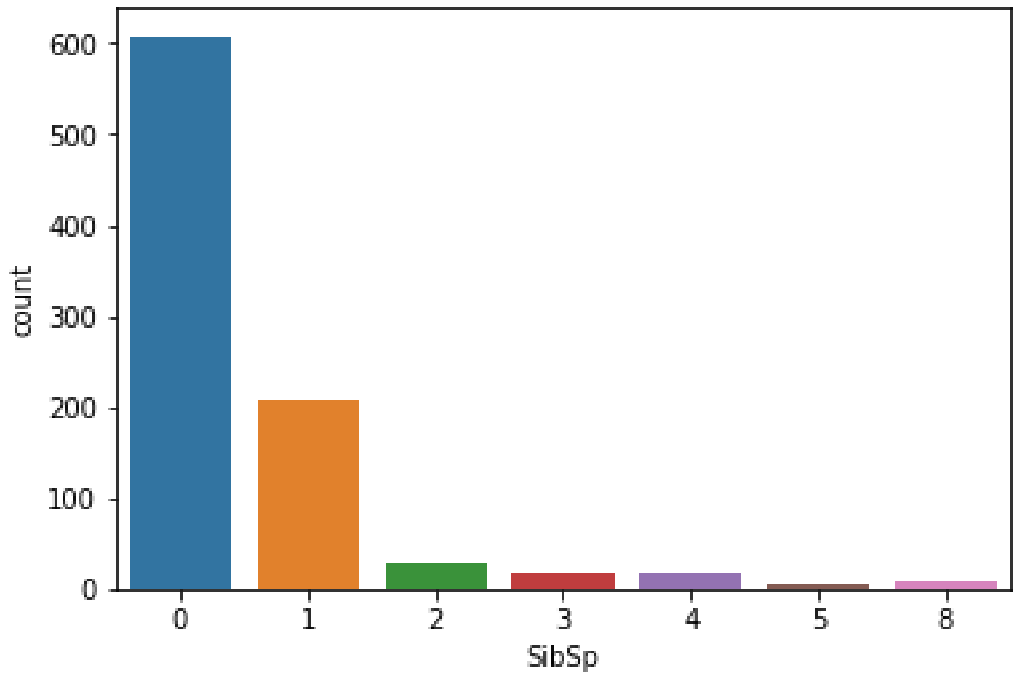
```
In [14]: sns.countplot('Sex',data=train)
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x247582af940>
```



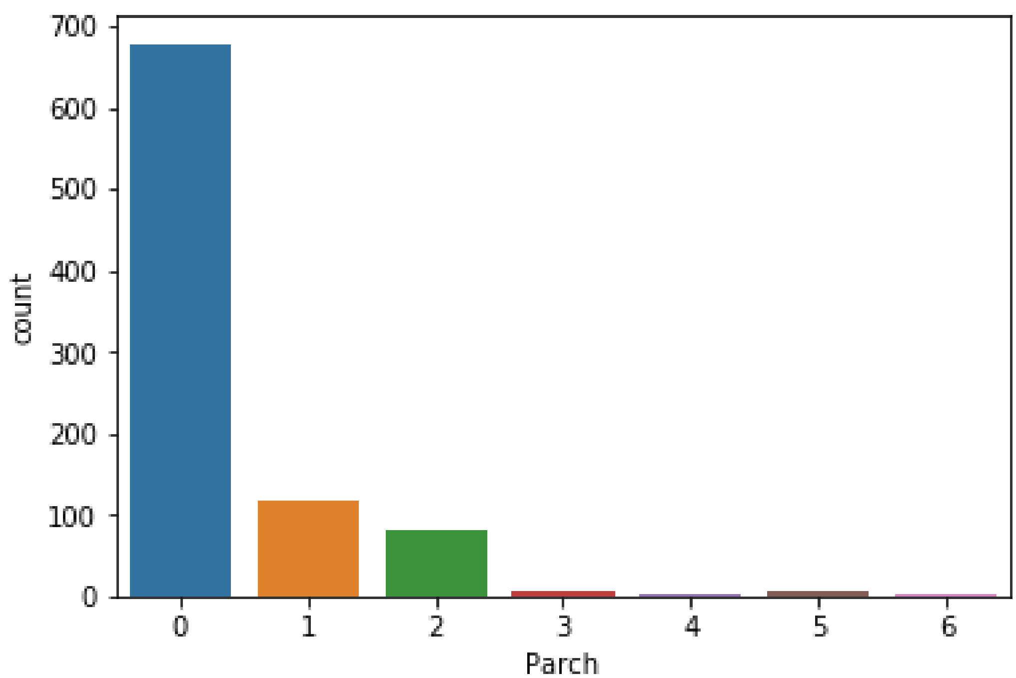

```
In [15]: sns.countplot('SibSp',data=train)
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x247582fa240>
```



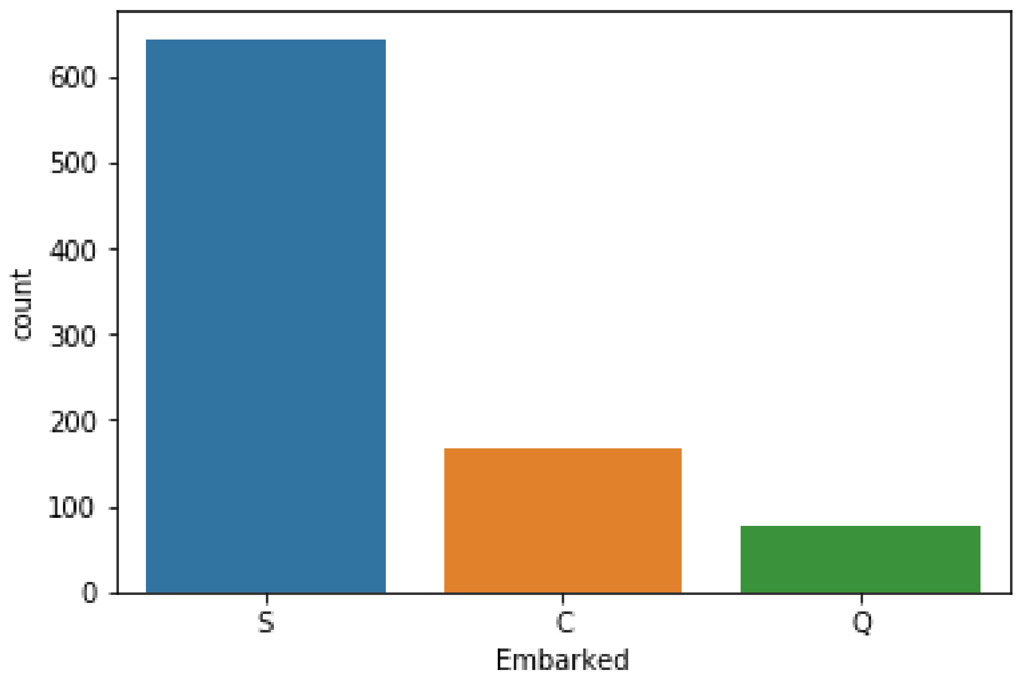

```
In [16]: sns.countplot('Parch',data=train)
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x24758366828>
```



```
In [17]: sns.countplot('Embarked',data=train)
```

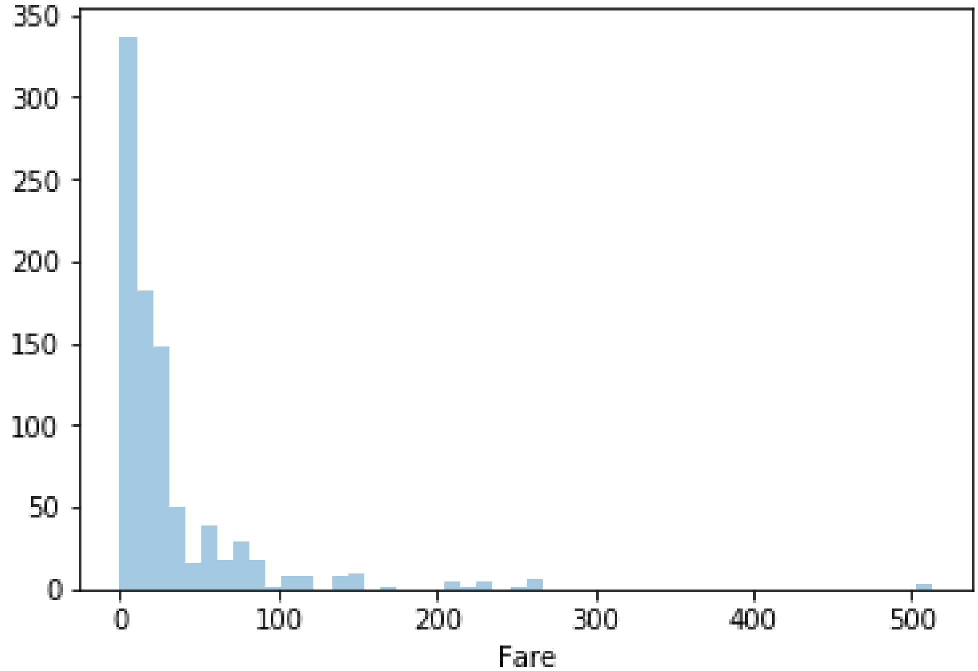
```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x247583a93c8>
```




```
In [18]: # For numerical features
         sns.distplot(train['Fare'], kde=False)
```

```
C:\Users\paliw\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'norm
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

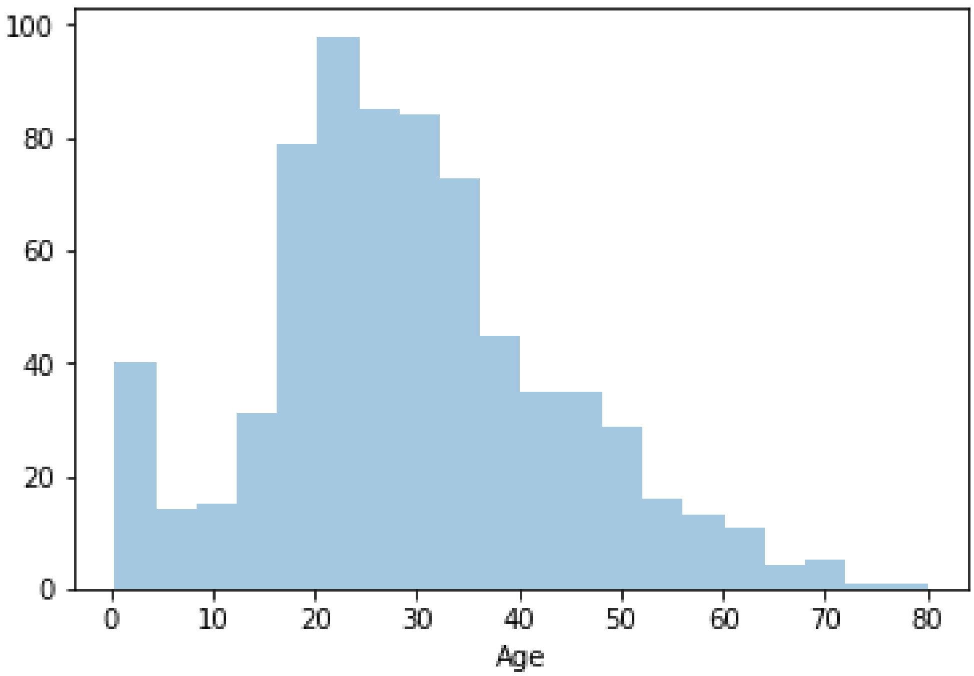
```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x24758416668>
```




```
In [20]: # Drop NAN values & plot Age
         sns.distplot(train['Age'].dropna(), kde=False)
```

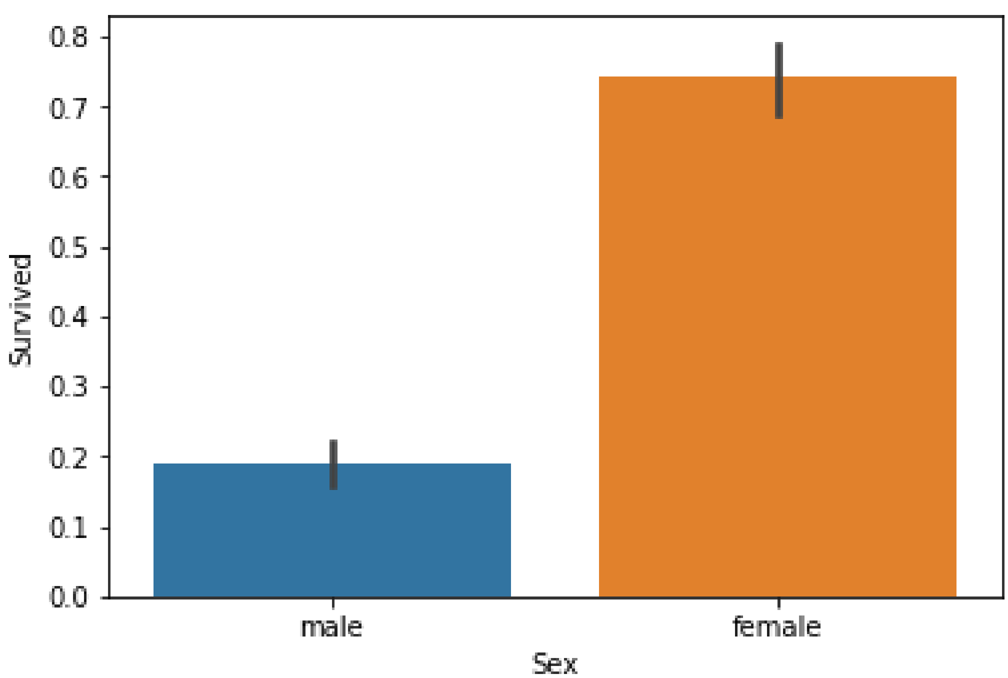
```
C:\Users\paliw\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'norm
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x24758286cf8>
```



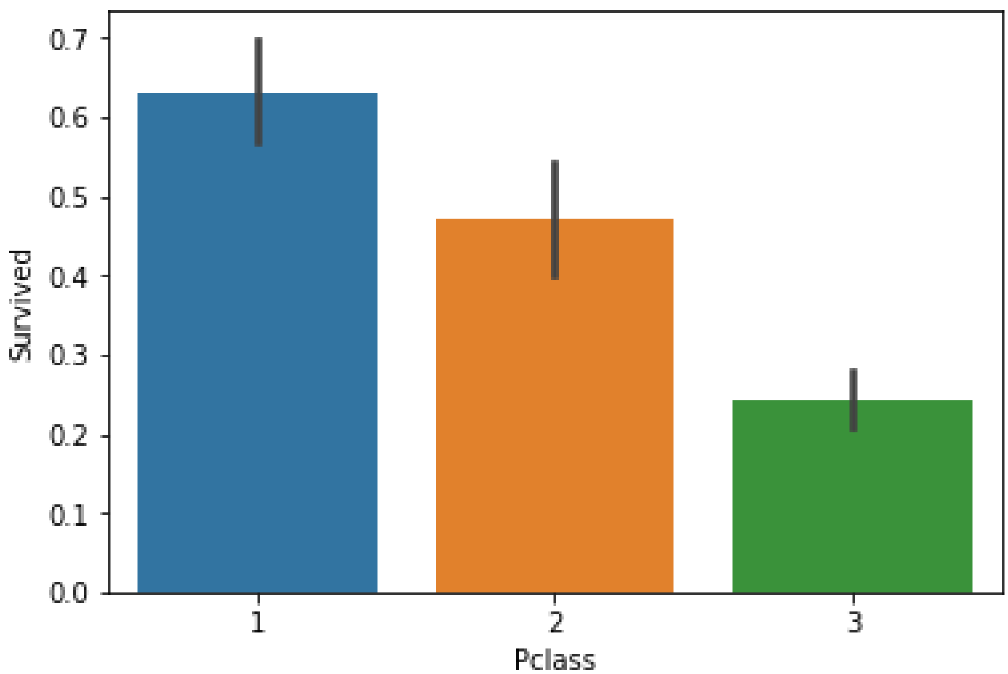

```
In [22]: # Bivariate Analysis
         sns.barplot(x ='Sex', y = 'Survived', data = train)

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x2475871ff28>
```



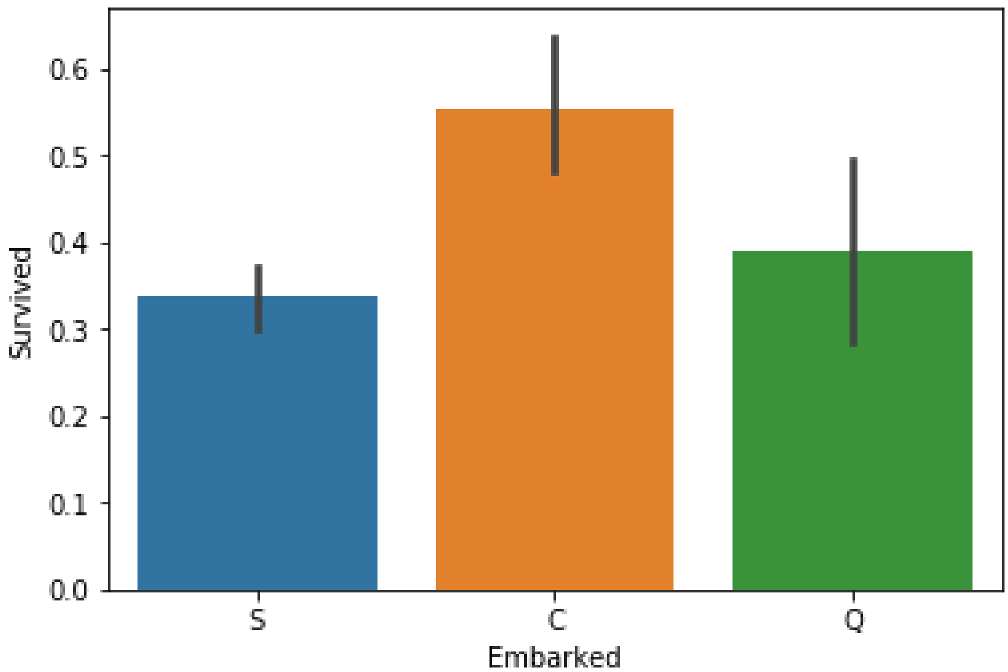
```
In [23]: sns.barplot(x ='Pclass', y = 'Survived', data = train)
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x2475876db00>
```



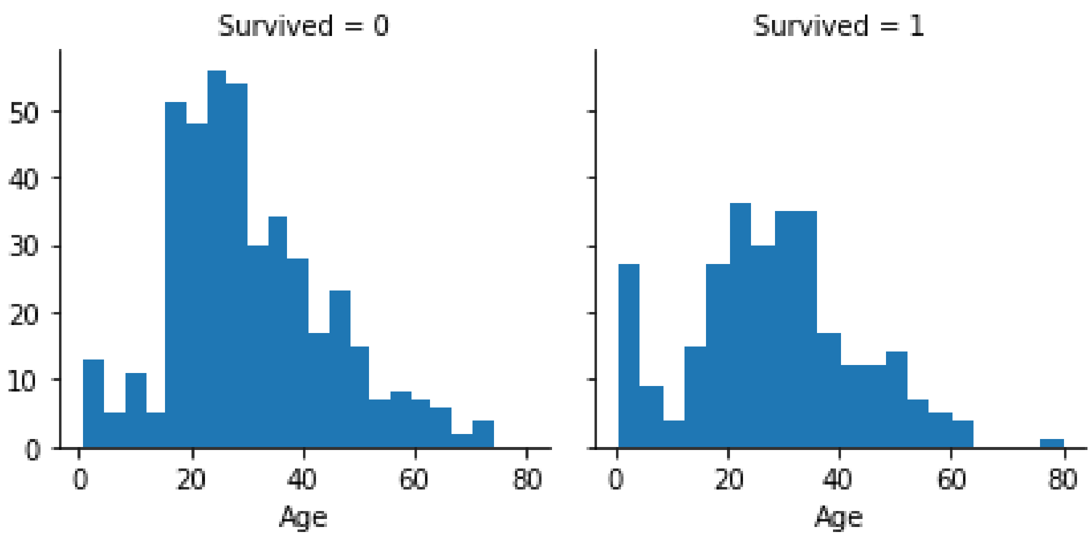

```
In [24]: sns.barplot(x ='Embarked', y = 'Survived', data = train)
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x24758729e10>
```



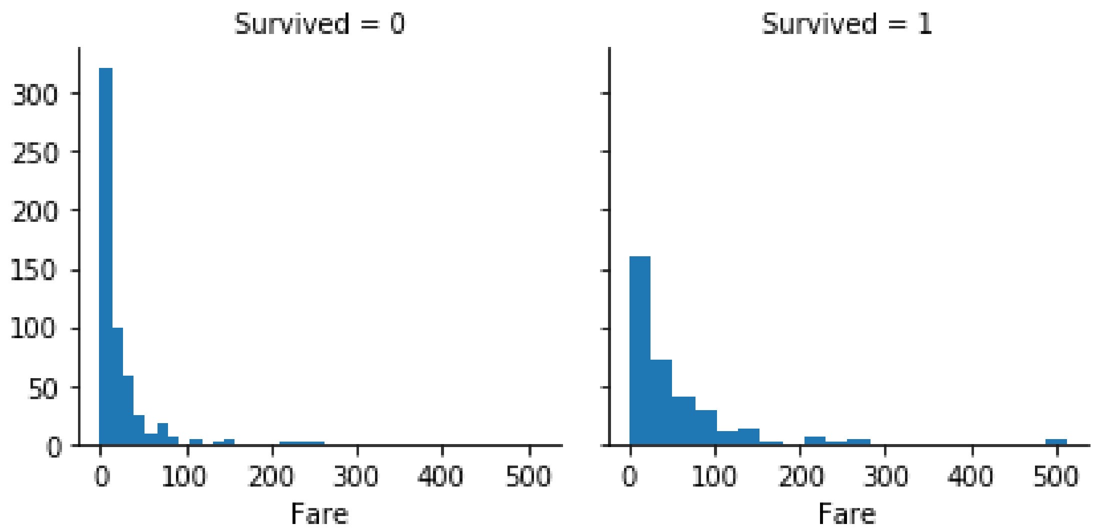

```
In [25]: g = sns.FacetGrid(train, col='Survived')
         g.map(plt.hist, 'Age', bins=20)
```

```
Out[25]: <seaborn.axisgrid.FacetGrid at 0x2475881a9b0>
```



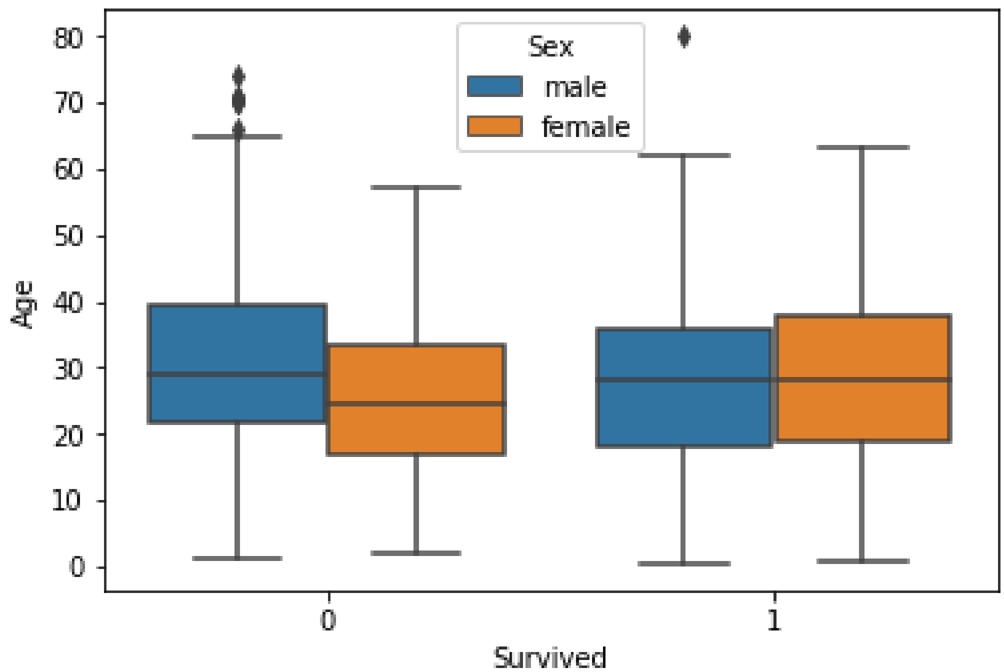

```
In [26]: g = sns.FacetGrid(train, col='Survived')
         g.map(plt.hist, 'Fare', bins=20)
```

```
Out[26]: <seaborn.axisgrid.FacetGrid at 0x247588800f0>
```



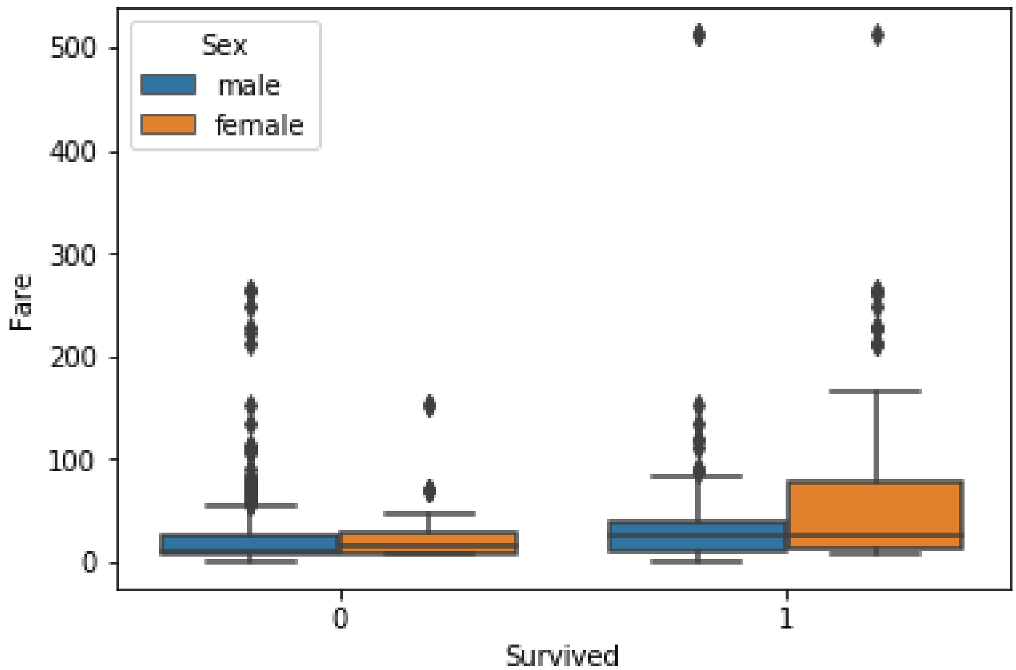

```
In [28]: sns.boxplot(x="Survived", y ="Age", hue = "Sex", data = train)

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x24758aa6f60>
```



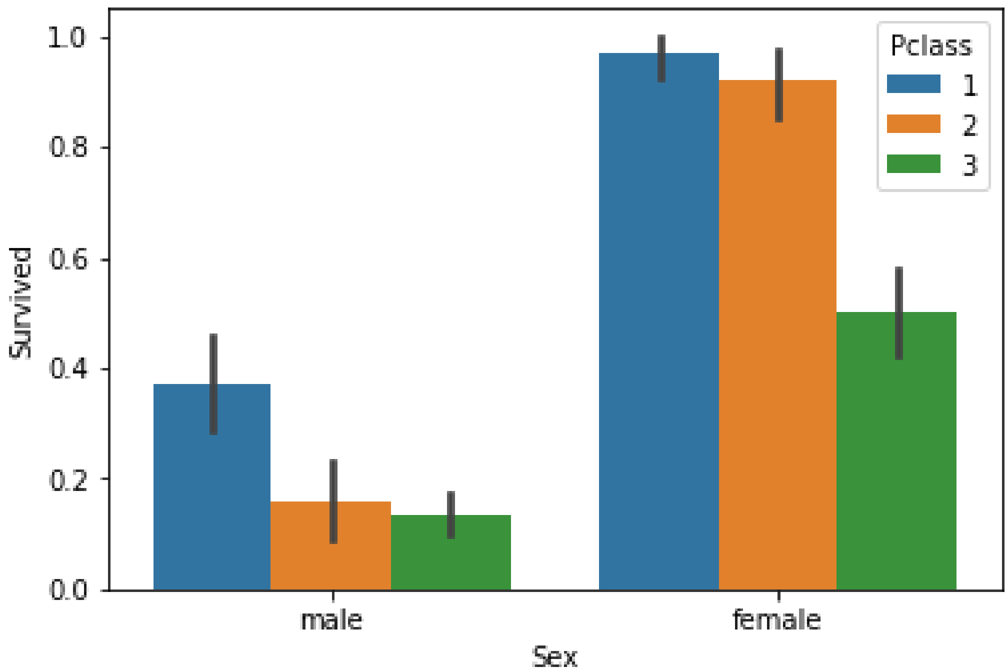
```
In [29]: sns.boxplot(x="Survived", y ="Fare", hue = "Sex", data = train)
```

```
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x24758b4f128>
```



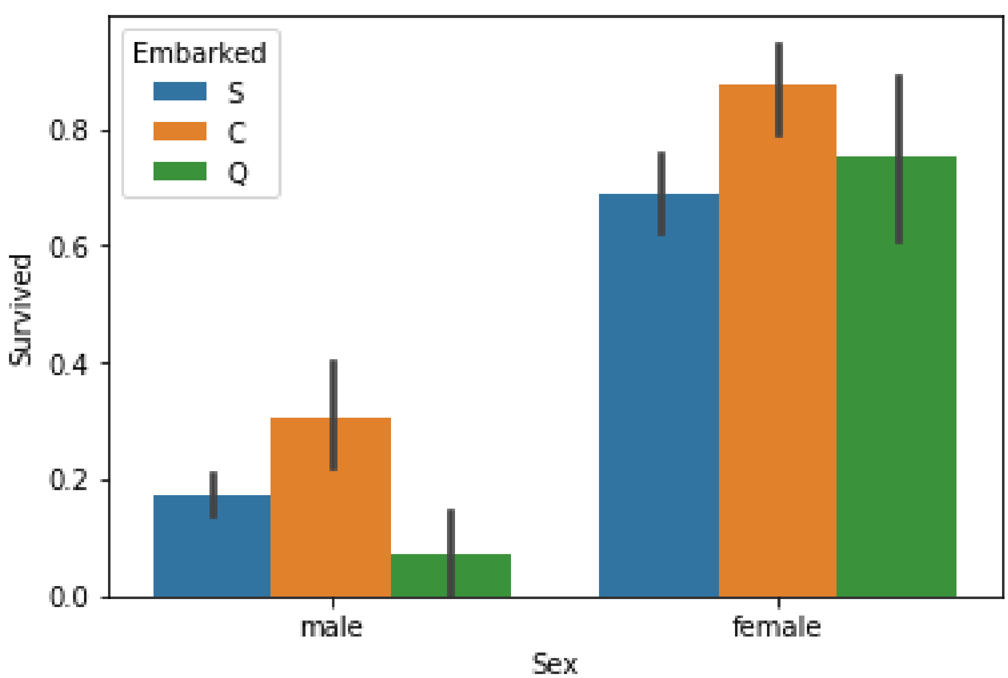

```
In [30]: sns.barplot(x ='Sex', y = 'Survived', hue = 'Pclass', data = train)
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x24758bed828>
```



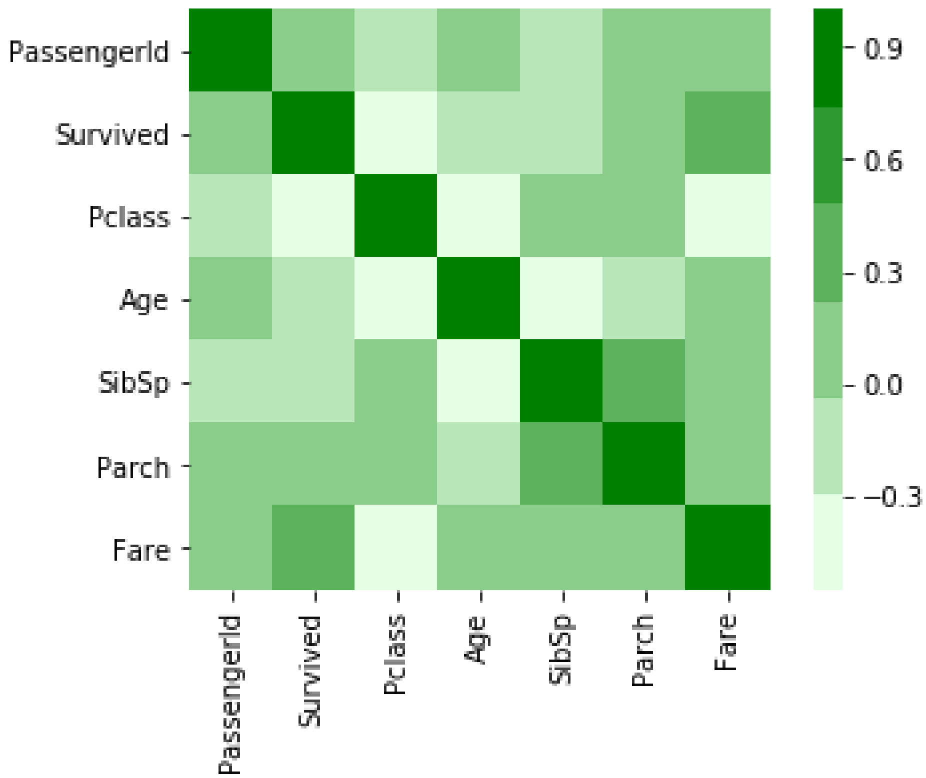

```
In [31]: sns.barplot(x ='Sex', y = 'Survived', hue = 'Embarked', data = train)
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x24758c4ab70>
```



In [34]:
```
correlation = train.corr()
sns.heatmap(correlation, mask=np.zeros_like(correlation, dtype=np.bool),
            cmap=sns.light_palette("green"),
            square=True)
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x24759d14898>



TITANIC DATA PRE PROCESSING

November 18, 2018

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
train = pd.read_csv('C:\\Users\\paliw\\Downloads\\train_titanic.csv')
test = pd.read_csv('C:\\Users\\paliw\\Downloads\\test_titanic.csv')
```

In [4]:
```
train.head(6)
```

Out[4]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         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
5                                   Moran, Mr. James    male   NaN      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
5      0            330877   8.4583   NaN        Q
```

A.Delete unnecessary features from dataset Unimportant features namely Name, Ticket and Cabin are dropped

In [5]:
```
train.drop('Name', axis=1, inplace=True)
test.drop('Name', axis=1, inplace=True)
train.drop('Cabin', axis=1, inplace=True)
test.drop('Cabin', axis=1, inplace=True)
```

```
train.drop('Ticket', axis=1, inplace=True)
test.drop('Ticket', axis=1, inplace=True)
```

In [6]:
```
train.head(6)
```

Out[6]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
5	6	0	3	male	NaN	0	0	8.4583	Q

B.MISSING VALUES :

1. To fill out the missing values of AGE : Both for male and female passengers we fill the missing values with the median age

In [7]:
```
train["Age"].fillna(train.groupby("Sex")["Age"].transform("median"), inplace=True)
test["Age"].fillna(test.groupby("Sex")["Age"].transform("median"), inplace=True)
```

In [8]:
```
train.head(6)
```

Out[8]:

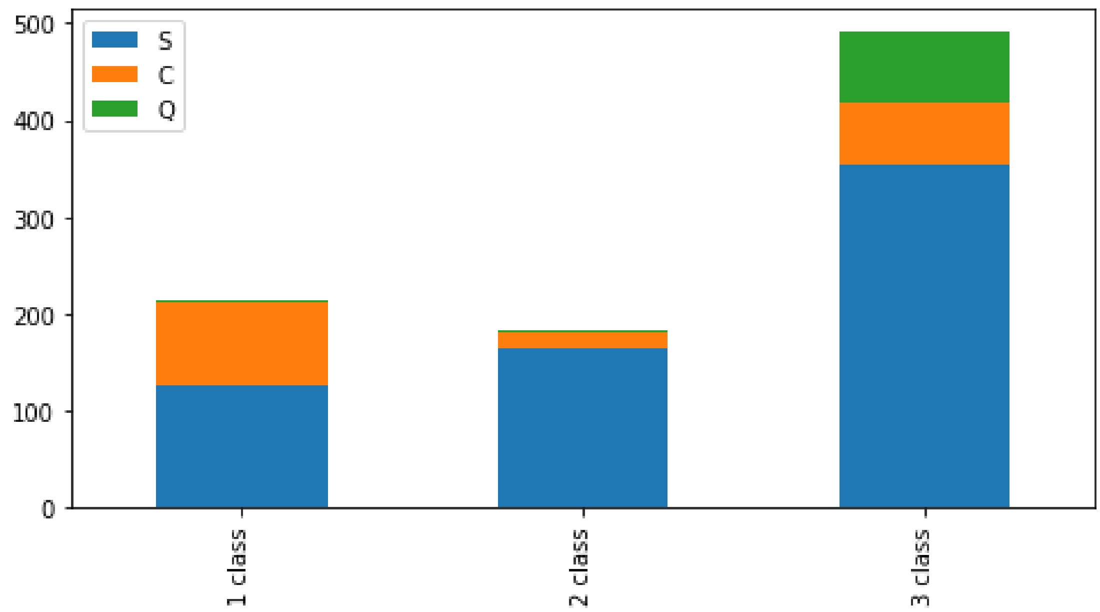
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.0	1	0	7.2500	S
1	2	1	1	female	38.0	1	0	71.2833	C
2	3	1	3	female	26.0	0	0	7.9250	S
3	4	1	1	female	35.0	1	0	53.1000	S
4	5	0	3	male	35.0	0	0	8.0500	S
5	6	0	3	male	29.0	0	0	8.4583	Q

2. MISSING VALUES OF EMBARKED : Fill out missing embark with S embark

In [12]:
```
P1 = train[train['Pclass']==1]['Embarked'].value_counts()
P2 = train[train['Pclass']==2]['Embarked'].value_counts()
P3 = train[train['Pclass']==3]['Embarked'].value_counts()
df = pd.DataFrame([P1, P2, P3])
df.index = ['1 class','2 class', '3 class']
```

In [14]:
```
df.plot(kind='bar',stacked=True, figsize=(8,4))
```

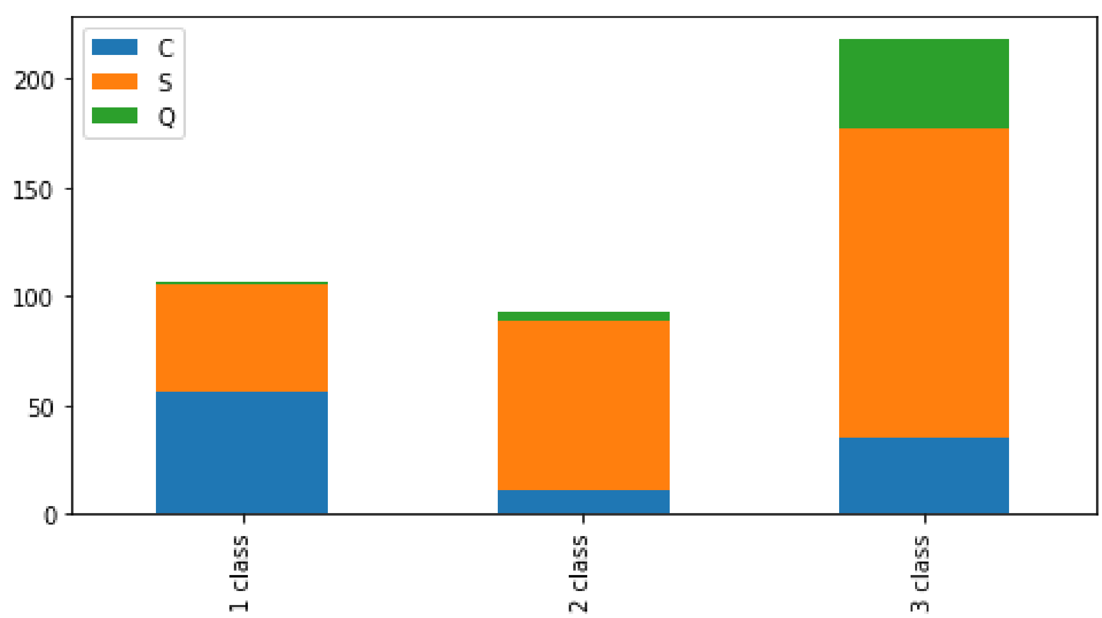
Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x292a96917f0>
```



In [15]:
```
P1 = test[test['Pclass']==1]['Embarked'].value_counts()
P2 = test[test['Pclass']==2]['Embarked'].value_counts()
P3 = test[test['Pclass']==3]['Embarked'].value_counts()
df = pd.DataFrame([P1, P2, P3])
df.index = ['1 class','2 class', '3 class']
```

In [16]:
```
df.plot(kind='bar',stacked=True, figsize=(8,4))
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x292a8727ef0>
```



In all classes in both traininng and test data, we can see that more than 50 percent have S embark. so we will fill missing values by S embark.

```
In [18]: combinedData = [train, test]
         for dataset in combinedData:
             dataset['Embarked'] = dataset['Embarked'].fillna('S')

In [19]: train.head(5)

Out[19]:    PassengerId  Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked
         0            1         0       3    male  22.0      1      0   7.2500        S
         1            2         1       1  female  38.0      1      0  71.2833        C
         2            3         1       3  female  26.0      0      0   7.9250        S
         3            4         1       1  female  35.0      1      0  53.1000        S
         4            5         0       3    male  35.0      0      0   8.0500        S

In [ ]: COVERTING:
        converting categorical features to numeric ones.
        1.coverting embarked values
        2.converting sex values

In [20]: embarkMapping = {"S": 0, "Q": 1, "C": 2}
         for dataset in combinedData:
             dataset['Embarked'] = dataset['Embarked'].map(embarkMapping)

In [21]: train.head(5)

Out[21]:    PassengerId  Survived  Pclass     Sex   Age  SibSp  Parch     Fare  \
         0            1         0       3    male  22.0      1      0   7.2500
         1            2         1       1  female  38.0      1      0  71.2833
         2            3         1       3  female  26.0      0      0   7.9250
         3            4         1       1  female  35.0      1      0  53.1000
         4            5         0       3    male  35.0      0      0   8.0500

            Embarked
         0         0
         1         2
         2         0
         3         0
         4         0

In [22]: Sex_map = {"male" : 1, "female": 0}
         for dataset in combinedData:
             dataset['Sex'] = dataset['Sex'].map(Sex_map)

In [23]: train.head(5)
```

Out[23]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	22.0	1	0	7.2500	0
1	2	1	1	0	38.0	1	0	71.2833	2
2	3	1	3	0	26.0	0	0	7.9250	0
3	4	1	1	0	35.0	1	0	53.1000	0
4	5	0	3	1	35.0	0	0	8.0500	0

CONVERSION : CONTINUOUS NUMERICAL FEATURES INTO DISCRETE FEATURES As observed from our analysis, specific age bands have high correlation with survival, so here we convert the continuos age feature into discrete age bands. Simialrly, fare -> discrete fare bands.

In [24]:
```
for dataset in combinedData:
    dataset.loc[ dataset['Age'] <= 15, 'Age'] = 0,
    dataset.loc[(dataset['Age'] > 15) & (dataset['Age'] <= 25), 'Age'] = 1,
    dataset.loc[(dataset['Age'] > 25) & (dataset['Age'] <= 35), 'Age'] = 2,
    dataset.loc[(dataset['Age'] > 35) & (dataset['Age'] <= 60), 'Age'] = 3,
    dataset.loc[ dataset['Age'] > 60, 'Age'] = 4
```

In [25]:
```
train.head(6)
```

Out[25]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	1.0	1	0	7.2500	0
1	2	1	1	0	3.0	1	0	71.2833	2
2	3	1	3	0	2.0	0	0	7.9250	0
3	4	1	1	0	2.0	1	0	53.1000	0
4	5	0	3	1	2.0	0	0	8.0500	0
5	6	0	3	1	2.0	0	0	8.4583	1

In [26]:
```
for dataset in combinedData:
    dataset.loc[ dataset['Fare'] <= 15, 'Fare'] = 0,
    dataset.loc[(dataset['Fare'] > 15) & (dataset['Fare'] <= 30), 'Fare'] = 1,
    dataset.loc[(dataset['Fare'] > 30) & (dataset['Fare'] <= 100), 'Fare'] = 2,
    dataset.loc[ dataset['Fare'] > 100, 'Fare'] = 3
```

In [27]:
```
train.head(6)
```

Out[27]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	1	1.0	1	0	0.0	0
1	2	1	1	0	3.0	1	0	2.0	2
2	3	1	3	0	2.0	0	0	0.0	0
3	4	1	1	0	2.0	1	0	2.0	0
4	5	0	3	1	2.0	0	0	0.0	0
5	6	0	3	1	2.0	0	0	0.0	1

CREATING NEW FEATURES: Features SibSp and Parch both when combined form passenger's family count, so inorder to reduce these features for better analysis and training we form a new feature "hasFamily" -> "1"(if has a family), "0"(if doesn't have a family).

In [28]:
```
for dataset in combinedData:
    dataset['FamilySize'] = dataset['SibSp'] + dataset['Parch']
```

```
    dataset['hasFamily'] = 1
    dataset.loc[dataset['FamilySize'] == 0, 'hasFamily'] = 0

train = train.drop(['Parch', 'SibSp', 'FamilySize'], axis=1)
test = test.drop(['Parch', 'SibSp', 'FamilySize'], axis=1)
```

Resultant pre processed data set.

In [29]:
```
train.head(15)
```

Out[29]:

	PassengerId	Survived	Pclass	Sex	Age	Fare	Embarked	hasFamily
0	1	0	3	1	1.0	0.0	0	1
1	2	1	1	0	3.0	2.0	2	1
2	3	1	3	0	2.0	0.0	0	0
3	4	1	1	0	2.0	2.0	0	1
4	5	0	3	1	2.0	0.0	0	0
5	6	0	3	1	2.0	0.0	1	0
6	7	0	1	1	3.0	2.0	0	0
7	8	0	3	1	0.0	1.0	0	1
8	9	1	3	0	2.0	0.0	0	1
9	10	1	2	0	0.0	2.0	2	1
10	11	1	3	0	0.0	1.0	0	1
11	12	1	1	0	3.0	1.0	0	0
12	13	0	3	1	1.0	0.0	0	0
13	14	0	3	1	3.0	2.0	0	1
14	15	0	3	0	0.0	0.0	0	0

In [ ]: