TITANIC EXPLORATORY DATA ANALYSIS USING PYTHON/ML

VIDIT AGARWAL PH - 9997302747

Some Background Information

The sinking of the RMS Titanic in the early morning of 15 April 1912, four days into the ship's maiden voyage from Southampton to New York City, was one of the deadliest peacetime maritime disasters in history, killing more than 1,500 people. The largest passenger liner in service at the time, Titanic had an estimated 2,224 people on board when she struck an iceberg in the North Atlantic. The ship had received six warnings of sea ice but was travelling at near maximum speed when the lookouts sighted the iceberg. Unable to turn quickly enough, the ship suffered a glancing blow that buckled the starboard (right) side and opened five of sixteen compartments to the sea. The disaster caused widespread outrage over the lack of lifeboats, lax regulations, and the unequal treatment of the three passenger classes during the evacuation. Inquiries recommended sweeping changes to maritime regulations, leading to the International Convention for the Safety of Life at Sea (1914), which continues to govern maritime safety.
from Wikipedia

EDA TITANIC DATA SET

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning:
  import pandas.util.testing as tm
```

LOAD THE " DATASET"

```
from google.colab import files       ##  code to upload files in "COLAB"
upload = files.upload()
```

Choose Files titanic.csv

- **titanic.csv**(application/vnd.ms-excel) - 61194 bytes, last modified: 6/15/2020 - 100% done

```
Saving titanic.csv to titanic (1).csv
```

```
df = pd.read_csv('titanic.csv')
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs	female	38.0	1	0	PC 17599	71.283

```
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
```

1) From Data Info we came to know that 'Age' and 'Cabin' are the entities which contains Nan values

2) Data contains 12 columns and 891 rows

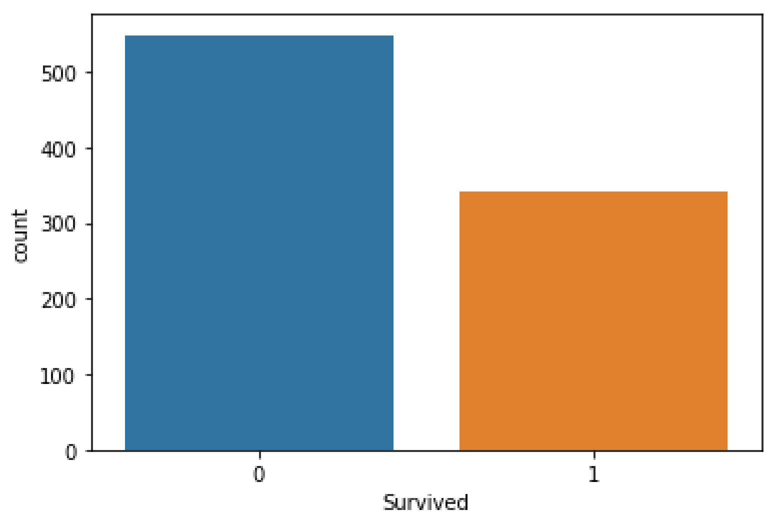
```
df.describe()
```

	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

Univariate EDA:

```
## What is the Count of Survived vs Not Survived?
```

```
sns.countplot(x='Survived', data=df);      ## THROUGH VISUALISATION , there are other plots t
```




```
not_survived = df[df['Survived']==0]     ## LET'S START WITH FIRST METHOD
```

```
len(not_survived)  ## not Survived
```

549

```
## Hence no. of survived
```

```
891-549          ## (Survived total count) - (not-survived)
```

```
342
```

```
df['Survived'].value_counts()            ## THIS IS ANOTHER METHOD TO COUNT THE DIFFERENCE
```

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

SURVIVED - 342

NOT-SURVIVED - 549

Find out the Numerical Columns Basic Statistics:

```
df.count()
```

```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

```
df.max()
```

```
PassengerId                        891
Survived                             1
Pclass                               3
Name           van Melkebeke, Mr. Philemon
Sex                               male
Age                                 80
SibSp                                8
Parch                                6
Ticket                       WE/P 5735
Fare                           512.329
dtype: object
```

```
df.min()
```

```
PassengerId                   1
Survived                      0
Pclass                        1
Name           Abbing, Mr. Anthony
Sex                      female
Age                        0.42
SibSp                         0
Parch                         0
Ticket                   110152
```

```
df.mean()
```

```
PassengerId    446.000000
Survived         0.383838
Pclass           2.308642
Age             29.699118
SibSp            0.523008
Parch            0.381594
Fare            32.204208
dtype: float64
```

```
df.median()
```

```
PassengerId    446.0000
Survived         0.0000
Pclass           3.0000
Age             28.0000
SibSp            0.0000
Parch            0.0000
Fare            14.4542
dtype: float64
```

```
df.select_dtypes
```

```
<bound method DataFrame.select_dtypes of      PassengerId  Survived  Pclass  ...     Far
0              1         0       3  ...   7.2500   NaN         S
1              2         1       1  ...  71.2833   C85         C
2              3         1       3  ...   7.9250   NaN         S
3              4         1       1  ...  53.1000  C123         S
4              5         0       3  ...   8.0500   NaN         S
..           ...       ...     ...  ...      ...   ...       ...
886          887         0       2  ...  13.0000   NaN         S
887          888         1       1  ...  30.0000   B42         S
888          889         0       3  ...  23.4500   NaN         S
889          890         1       1  ...  30.0000  C148         C
890          891         0       3  ...   7.7500   NaN         Q

[891 rows x 12 columns]>
```

```
df.sum()
```

```
PassengerId                                                    397386
Survived                                                          342
Pclass                                                           2057
Name           Braund, Mr. Owen HarrisCumings, Mrs. John Brad...
Sex            malefemalefemalefemalemalemalemalemalefemalefe...
Age                                                           21205.2
SibSp                                                             466
Parch                                                             340
Ticket         A/5 21171PC 17599STON/O2. 31012821138033734503...
Fare                                                          28693.9
```

```
df.std()
```

```
PassengerId    257.353842
Survived         0.486592
Pclass           0.836071
Age             14.526497
SibSp            1.102743
Parch            0.806057
Fare            49.693429
dtype: float64
```

OR

We can use a single code as used above ie df.describe()

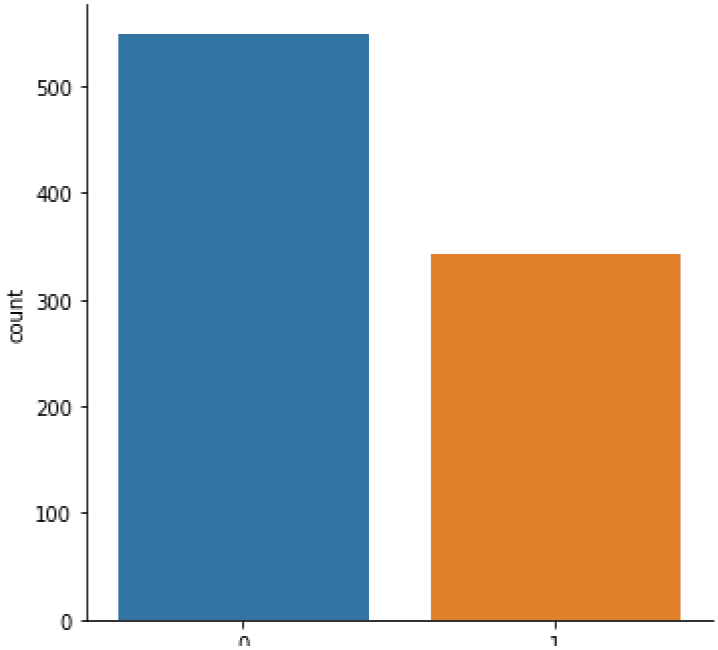
```
df.describe()
```

	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

Visualize Survived vs Not Survived:

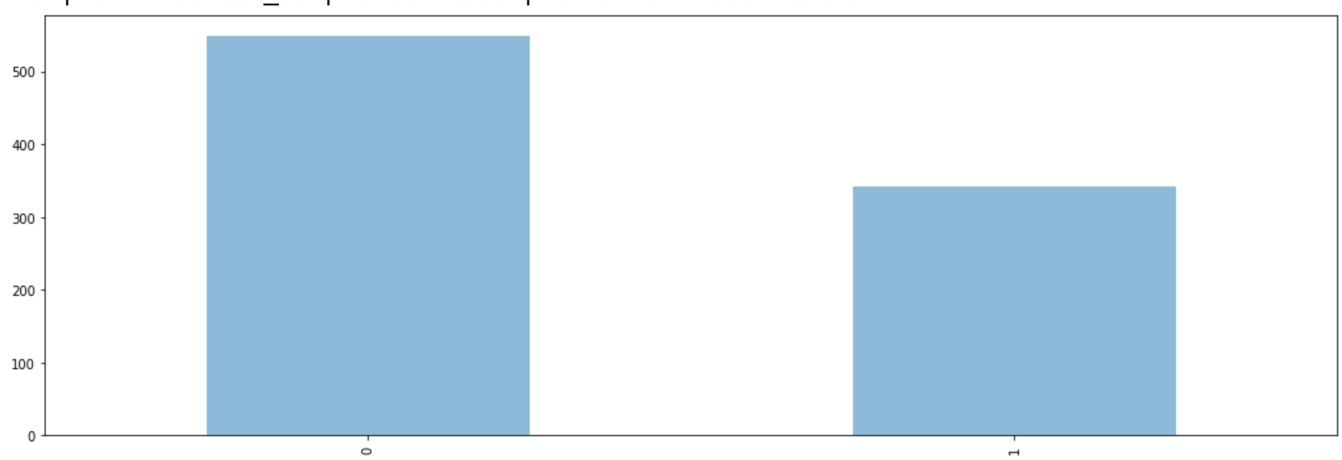
```
sns.factorplot('Survived', data=df, kind='count')     ## First Plot
```

<seaborn.axisgrid.FacetGrid at 0x7f1b4a36e0b8>



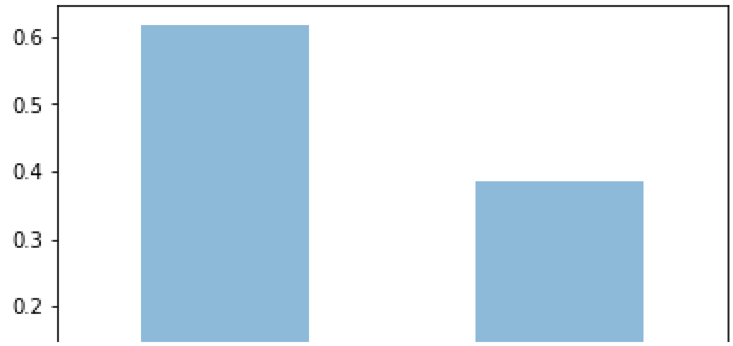

```
fig = plt.figure(figsize=(18,6))      ## To get a figure with proper structure

df.Survived.value_counts().plot(kind="bar",alpha=0.5)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6ee22aa898>



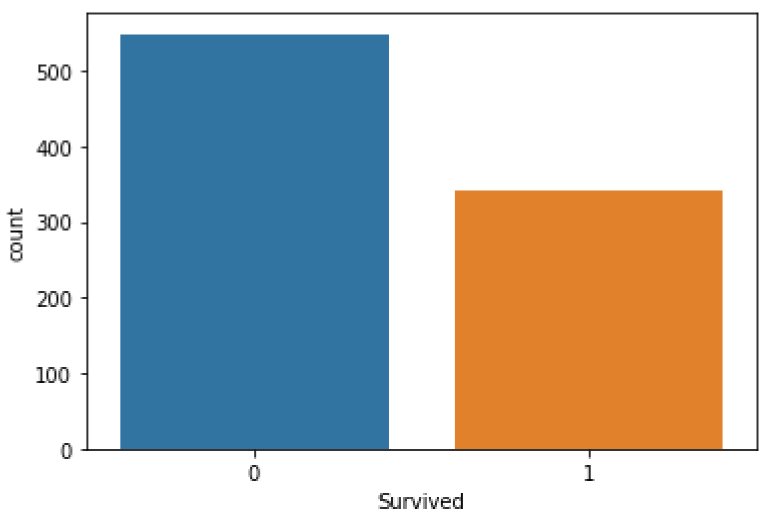

```
df.Survived.value_counts(normalize=True).plot(kind="bar",alpha=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6ee22173c8>
```



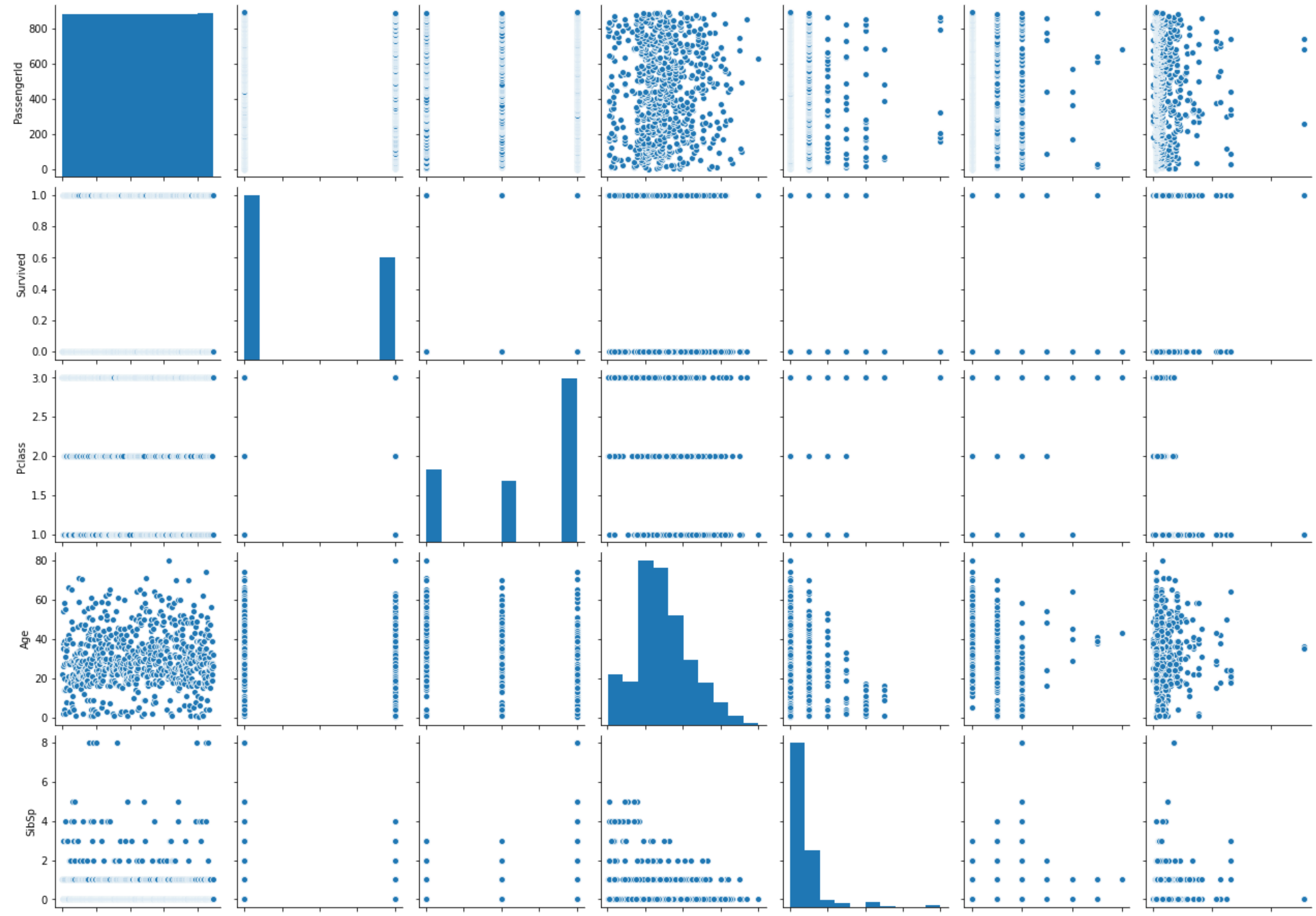
This plot was to get max accuracy in Terms as we came to know that 60% died and 40% Survived

```
sns.countplot(x='Survived', data=df);      ## THROUGH VISUALISATION , there are other plots t
```



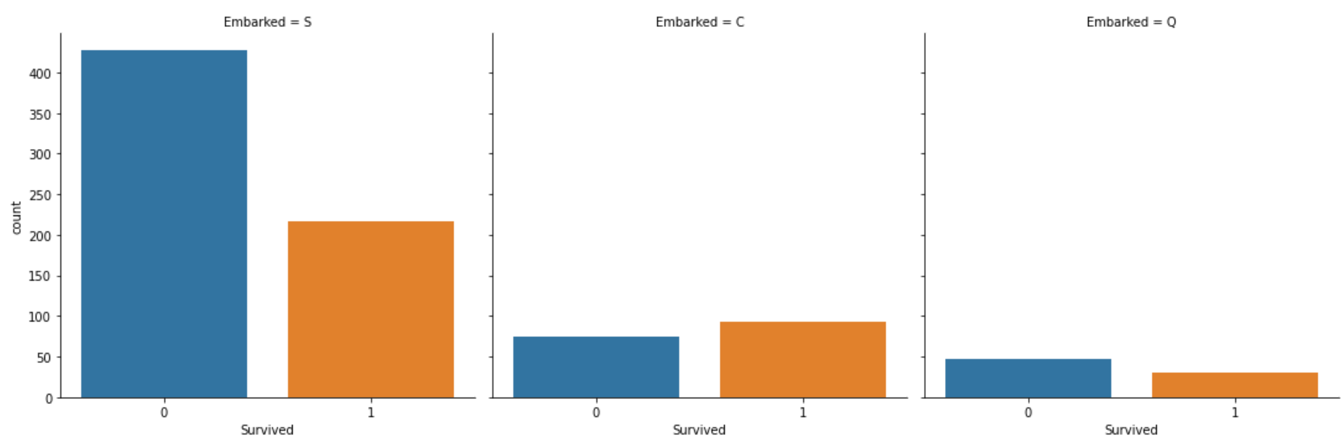

```
sns.catplot(x="Survived", kind="count", palette="ch:.25", data=df);
```


```
sns.pairplot(df);   ## Full explanation of every columns
```

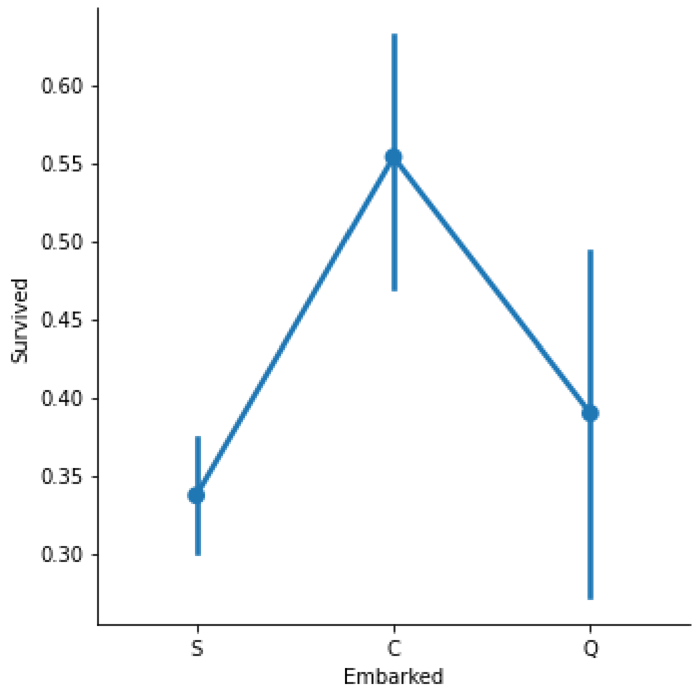


Visual EDA for single Categorical Column: "Embarked"

```
sns.catplot(x='Survived', col='Embarked', kind='count', data=df);
```



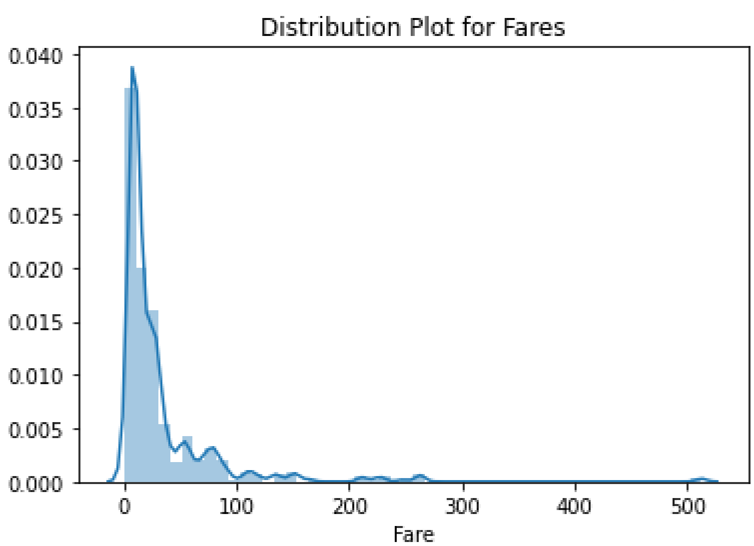
```
sns.catplot('Embarked','Survived', kind='point', data=df);
```



Visual EDA for single Continuous Column: "Fare" using Distribution Plot

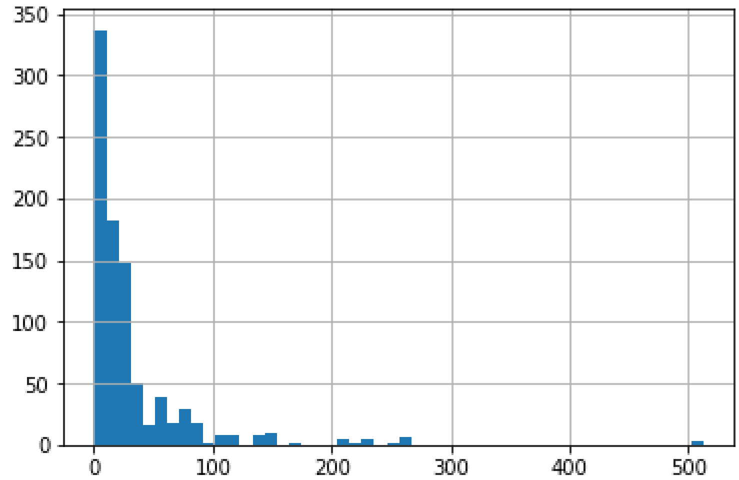
```
fare = df['Fare']
dist = sns.distplot(fare)
dist.set_title("Distribution Plot for Fares")
```

```
Text(0.5, 1.0, 'Distribution Plot for Fares')
```




```
df['Fare'].hist(bins=50)      ## If wanna analyse through graphs...Histograms are also used,
```

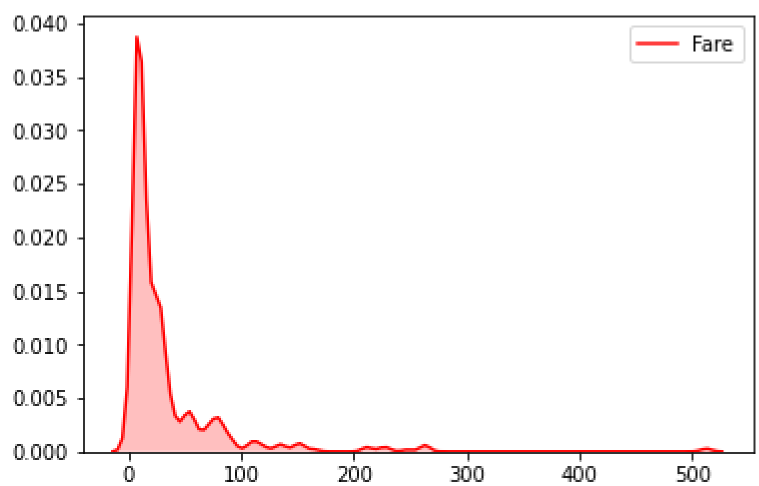
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1b1a6484a8>
```



Visual EDA for single Continuous Column: "Fare" using KDE(Kernel Density Estimation) Plot

```
sns.kdeplot(df['Fare'],color='r',shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1b1a34e2e8>
```



Bivariate EDA:

What is the count of Males and Females Survived and Not Survived in each Class?

```
df.groupby(['Survived','Sex'])['Survived'].count()    ## Total Male and Female survived
```

```
Survived  Sex   
0         female     81
          male      468
1         female    233
          male      109
Name: Survived, dtype: int64
```

```
df.groupby(['Survived','Sex'])['Survived'].sum()          ## Hence we can differentiate
```

```
Survived  Sex   
0         female      0
          male        0
1         female    233
          male      109
Name: Survived, dtype: int64
```

```
# Number of passengers who survived in each class grouped by sex. Also total was found for ea
df.pivot_table('Survived', 'Sex', 'Pclass', aggfunc=np.sum, margins=True)
```

Pclass	1	2	3	All
Sex				
female	91	70	72	233
male	45	17	47	109
All	136	87	119	342

```
# Number of men and women in each of the passenger class
df.groupby(['Sex', 'Pclass'])['Sex'].count()
```

```
Sex     Pclass
female  1          94
        2          76
        3         144
male    1         122
        2         108
        3         347
Name: Sex, dtype: int64
```

```
not_survived = df[df['Survived']==0]
```

```
# Number of passengers who did not survive in each class grouped by sex.
not_survived.pivot_table('Survived', 'Sex', 'Pclass', aggfunc=len, margins=True)
```

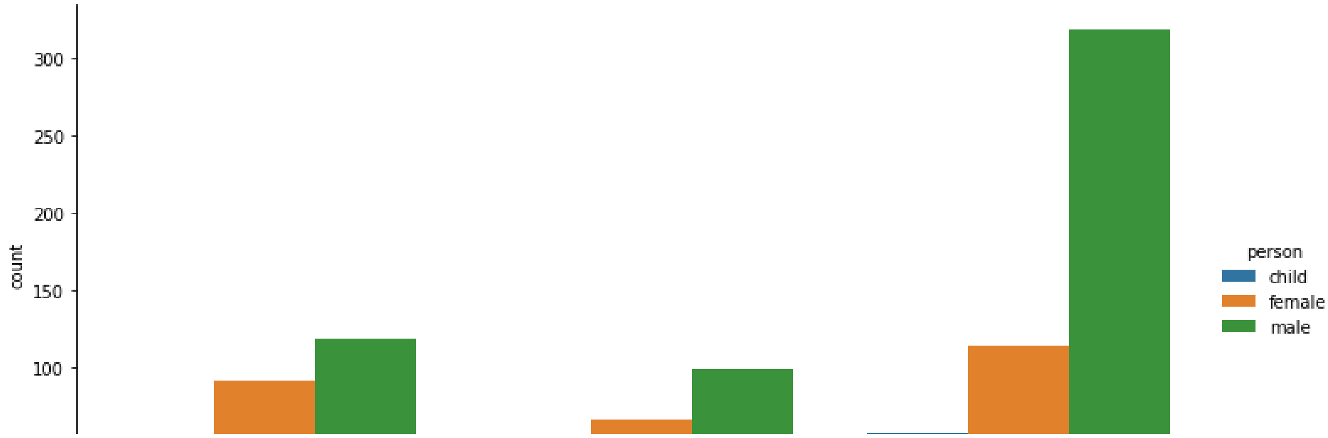
Pclass	1	2	3	All
Sex				
female	3	6	72	81
male	77	91	300	468

```
# Create a function to define those who are children (less than 16)
def male_female_child(passenger):
    age, sex = passenger

    if age < 16:
        return 'child'
    else:
        return sex


df['person'] = df[['Age', 'Sex']].apply(male_female_child, axis=1)

df.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques	female	35.0	1	0	113803	53.100

```
# Lets do a factorplot of passengers splitted into sex, children and class
sns.factorplot('Pclass', data=df, kind='count', hue='person', order=[1,2,3],  hue_order=['chi
```

```
<seaborn.axisgrid.FacetGrid at 0x7fa67a708f60>
```

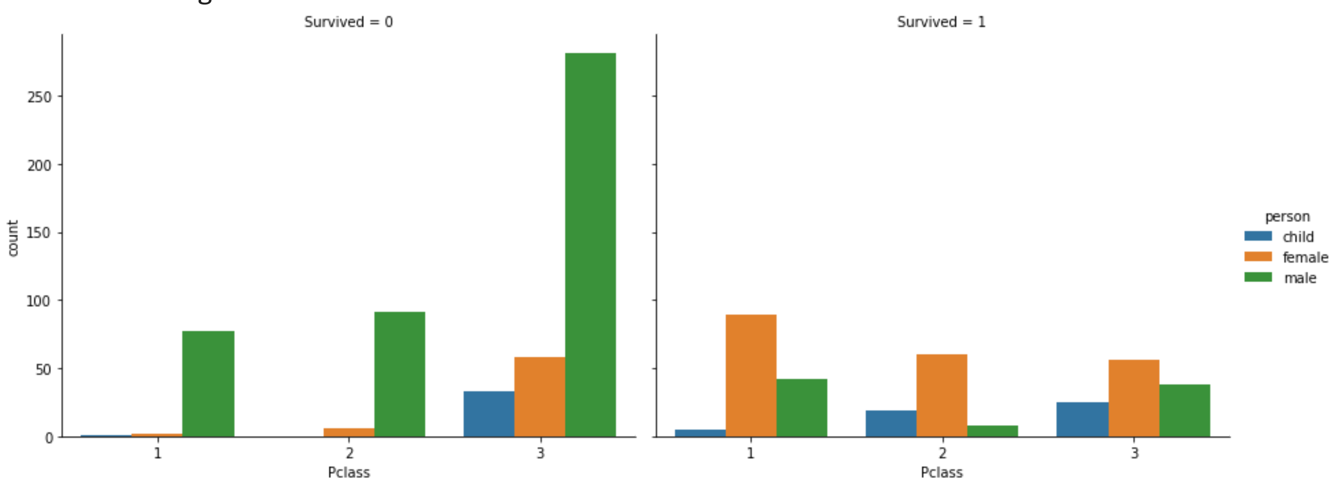



```
# Count number of men, women and children
df['person'].value_counts()
```

```
male      537
female    271
child      83
Name: person, dtype: int64
```

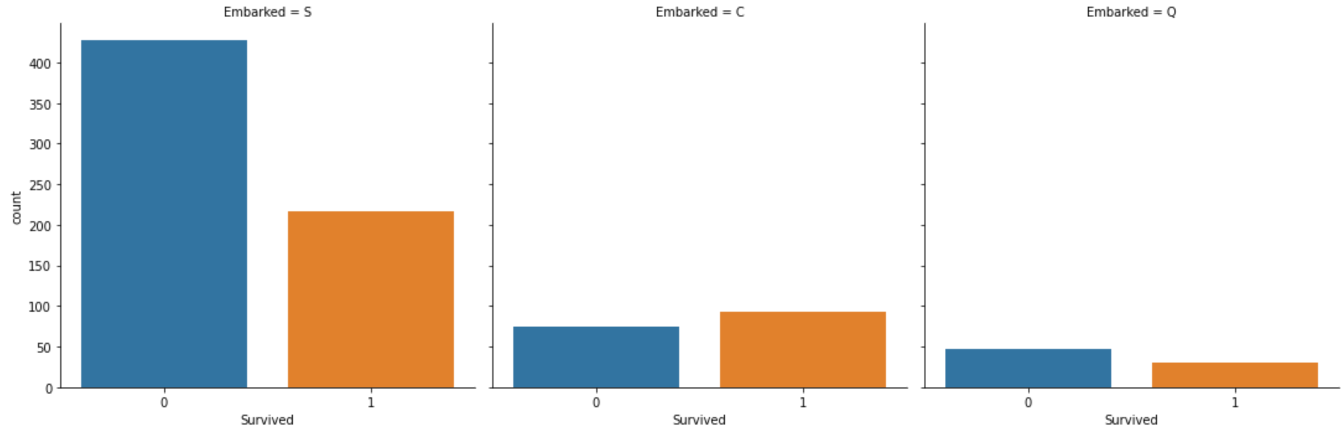
```
# Do the same as above, but split the passengers into either survived or not
sns.factorplot('Pclass', data=df, kind='count', hue='person', col='Survived', order=[1,2,3],
```

```
<seaborn.axisgrid.FacetGrid at 0x7fa67a602e48>
```

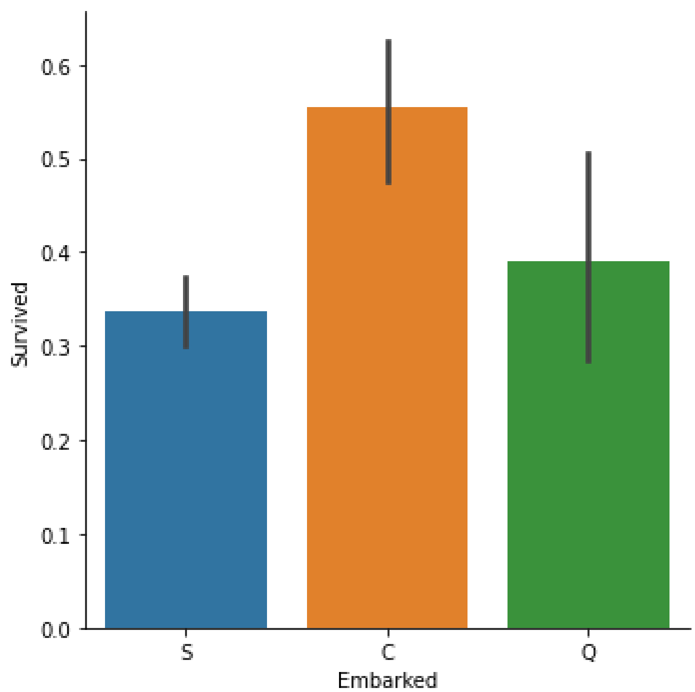


Visualize Survived and Not Survived with respect to the 'Embarked' Column:

```
sns.catplot(x='Survived', col='Embarked', kind='count', data=df);
```




```
sns.catplot('Embarked','Survived', kind='bar', data=df);
```



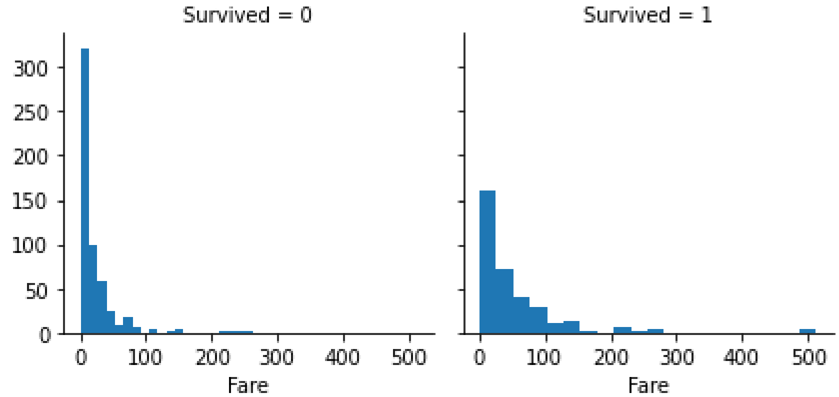
Embarked : Survival rate lowest for S and highest for C

Plot a Desnity Graph based on Fare and Survival Rate:

```
g = sns.FacetGrid(df, col='Survived')
```

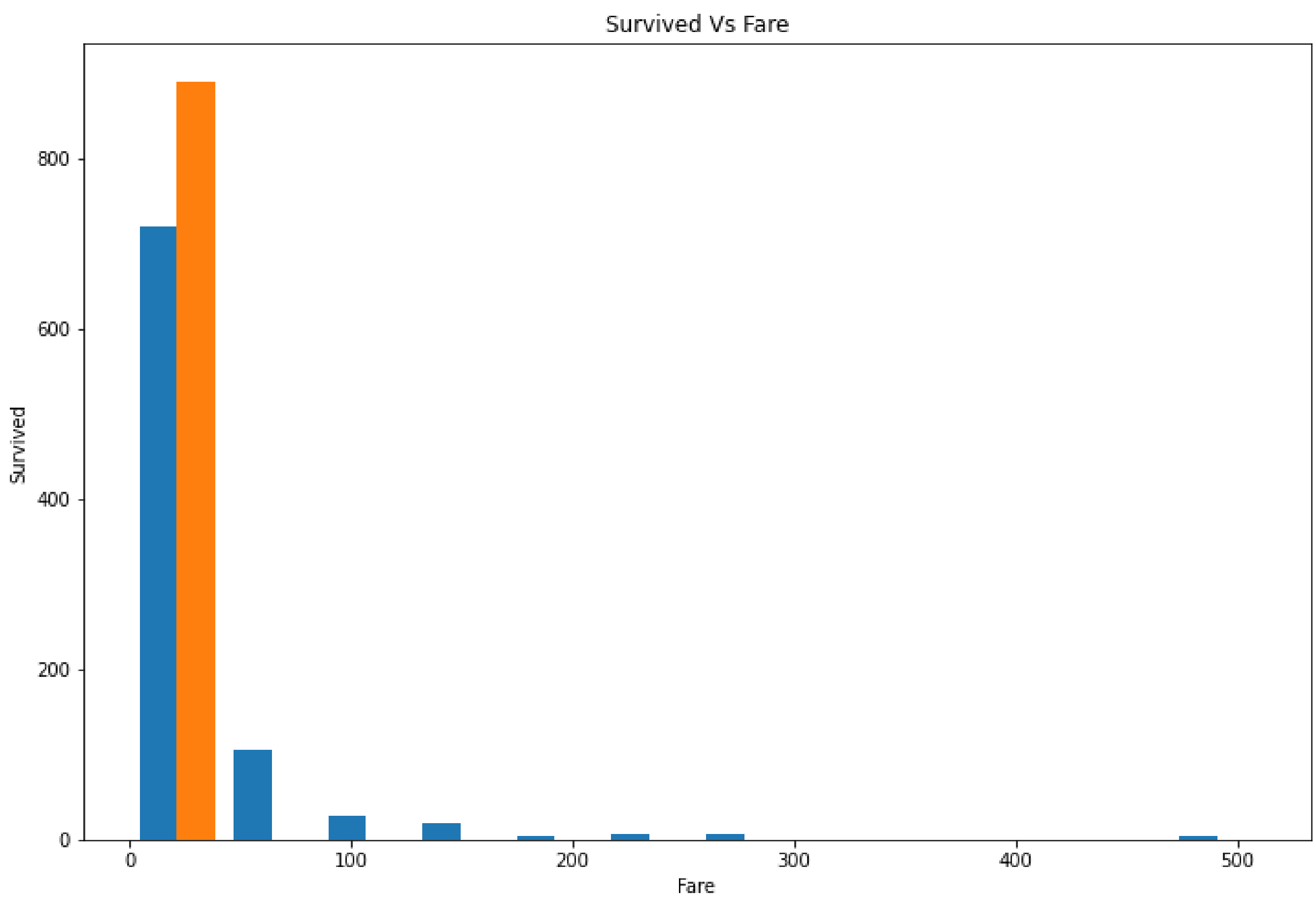
```
g = sns.FacetGrid(df, col='Survived')
g.map(plt.hist, 'Fare', bins=20)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fa67735fdd8>
```

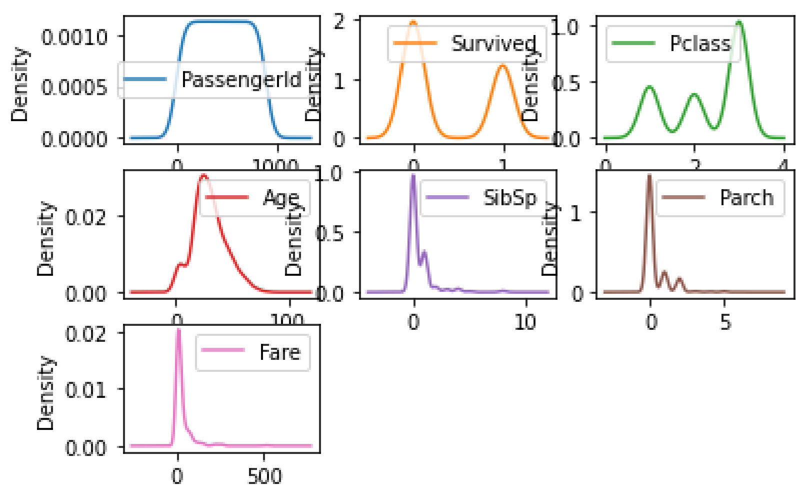



```
plt.figure(figsize=(12,8))
x = df['Fare']
y = df['Survived']
plt.hist([x,y], bins = int(180/15))
plt.xlabel('Fare')
plt.ylabel('Survived')
plt.title('Survived Vs Fare')
```

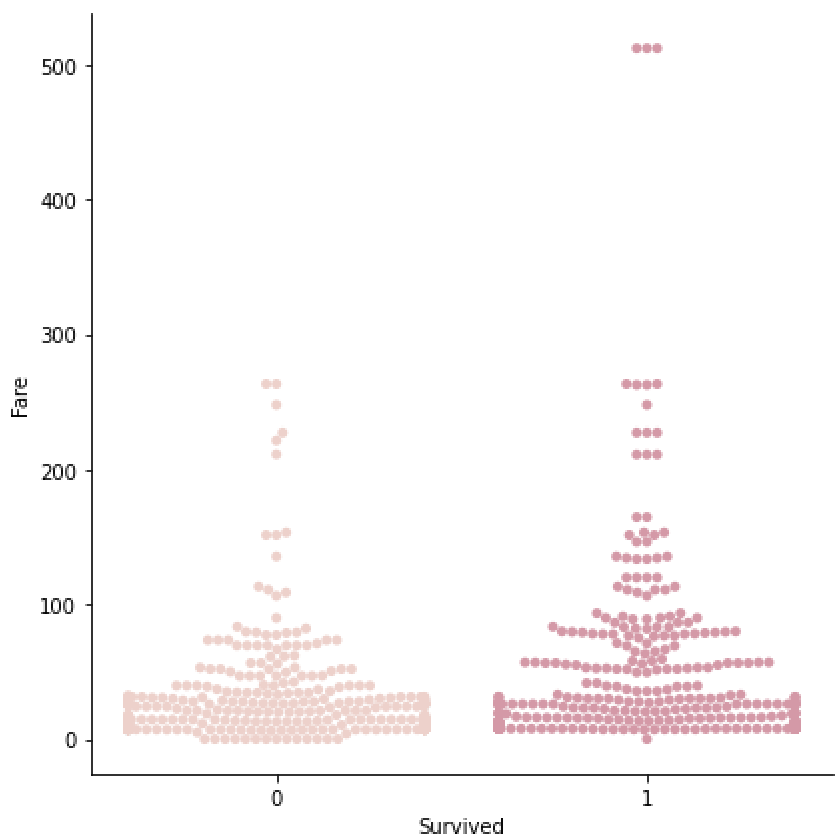
```
Text(0.5, 1.0, 'Survived Vs Fare')
```



```
df.plot(kind='density', subplots=True, layout=(3,3), sharex=False)    ## Analysis of densitie
plt.show()
```

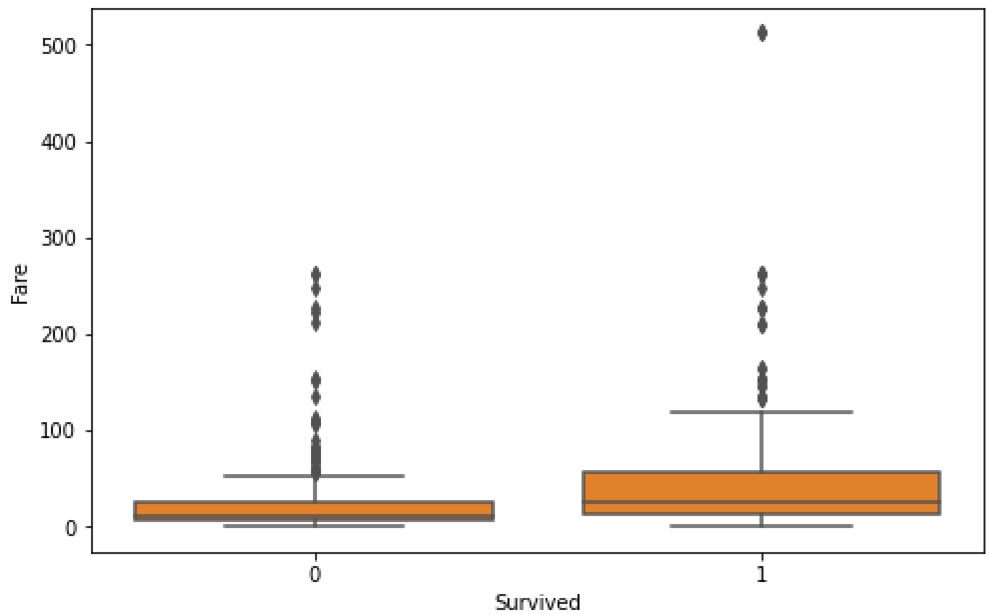



```
sns.catplot(x="Survived", y="Fare", kind="swarm", data=df, palette=sns.cubehelix_palette(5, s

plt.tight_layout()                                          ## Another kind of density plot
```



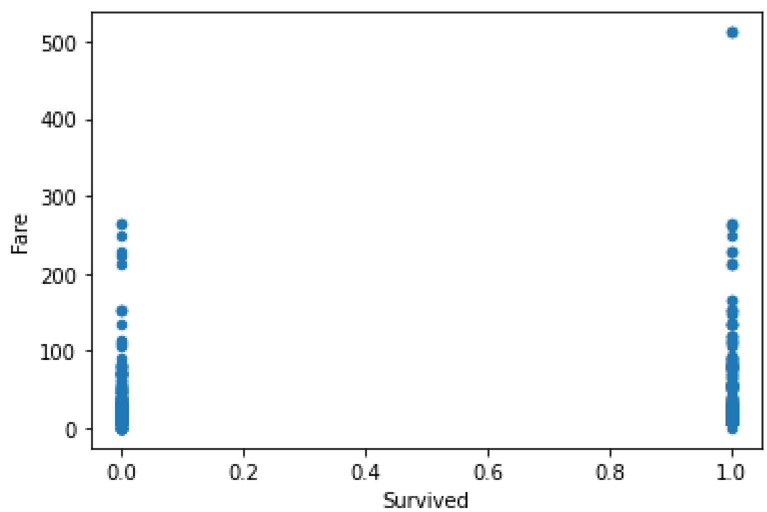

```
plt.figure(figsize = [8, 5])
base_color = sns.color_palette()[1]
sns.boxplot(data = df, x = 'Survived', y = 'Fare', color = base_color)
plt.xlabel('Survived')
```

```
plt.ylabel('Fare')
plt.show()
```



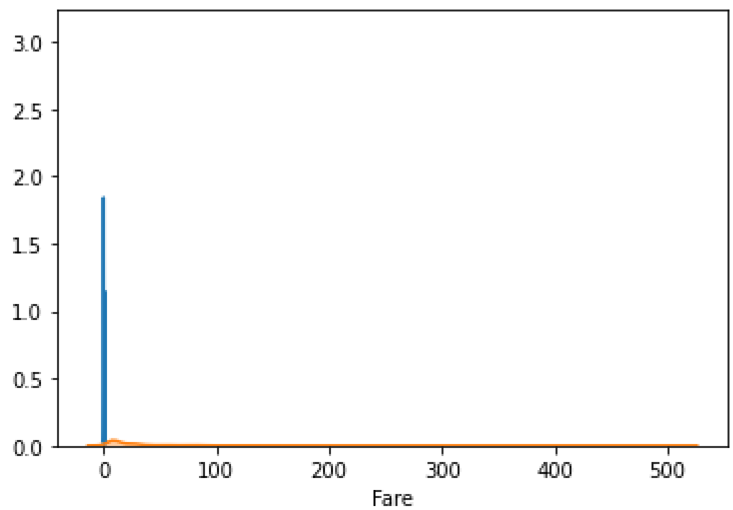

```
df.plot.scatter(x='Survived',y='Fare')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7faf2b61be80>




```
sns.relplot(x="Survived", y="Fare", data=df);
```


```
sns.distplot(df['Survived'])
sns.distplot(df['Fare']);
```



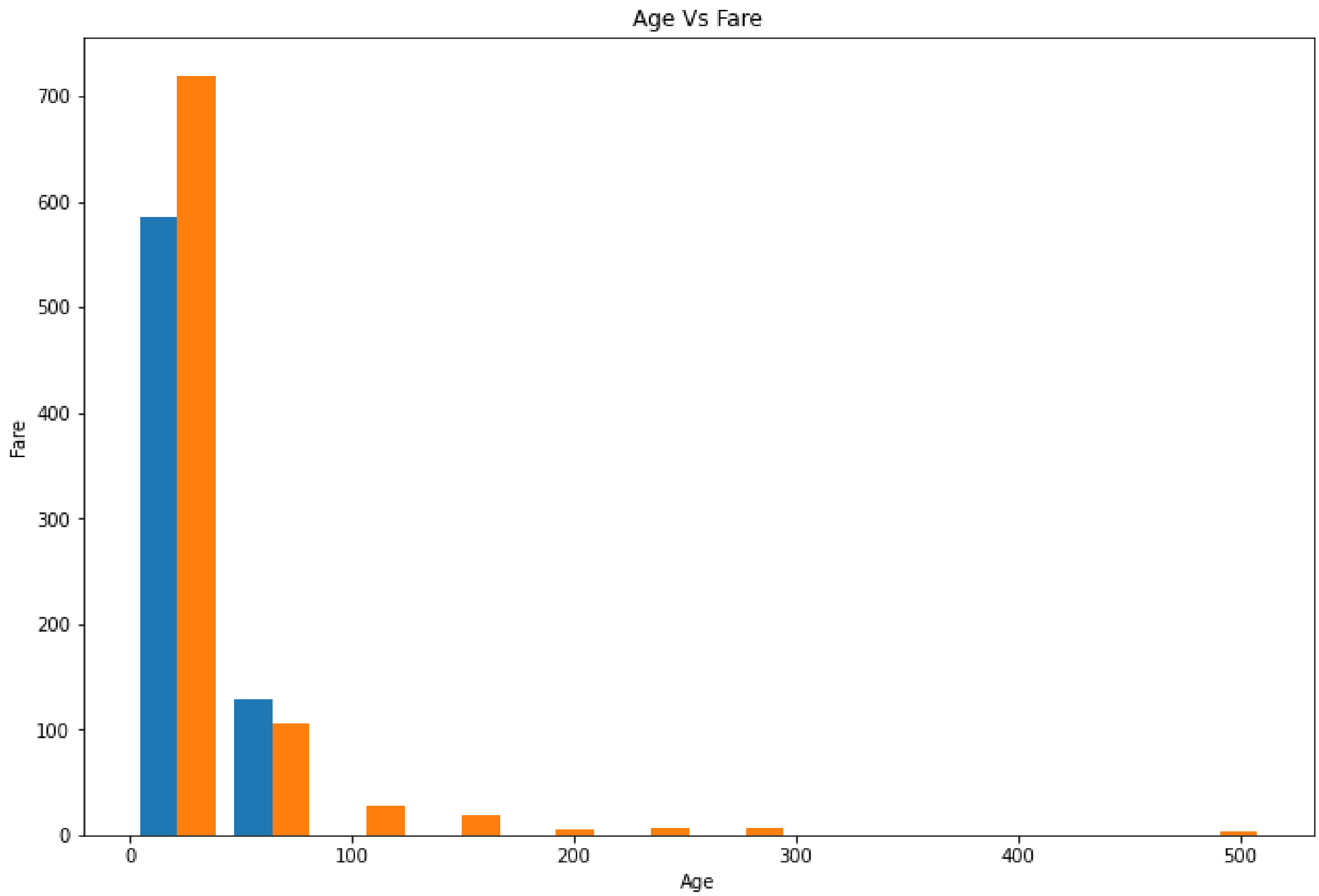
How are "Age" and "Fare" Columns related? Plot a Graph for the same:

```
sns.boxplot(x='Age', y='Fare',data=df)          ## boxplot
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc985bb1518>
```


```
plt.figure(figsize=(12,8))
x = df['Age']
y = df['Fare']
plt.hist([x,y], bins = int(180/15))
plt.xlabel('Age')
plt.ylabel('Fare')
plt.title('Age Vs Fare')
```

```
Text(0.5, 1.0, 'Age Vs Fare')
```



Multivariate EDA:

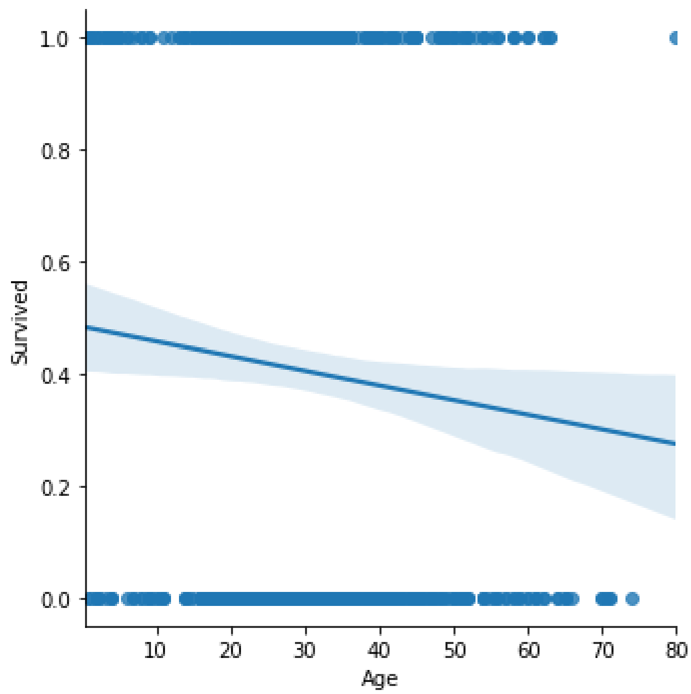
Does Age have an impact on Survival Rate for each Sex and Class group?

```
df['Age'].describe().T
```

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

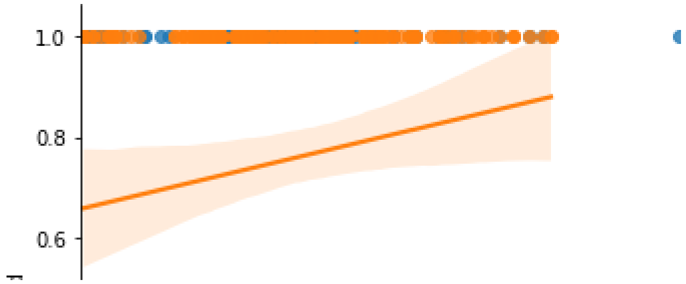
```
# Linear plot of age vs. survived
sns.lmplot('Age', 'Survived', data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0f5d232e10>
```




```
# Survived vs. Age grouped by Sex
sns.lmplot('Age', 'Survived', data=df, hue='Sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0f2c270780>
```



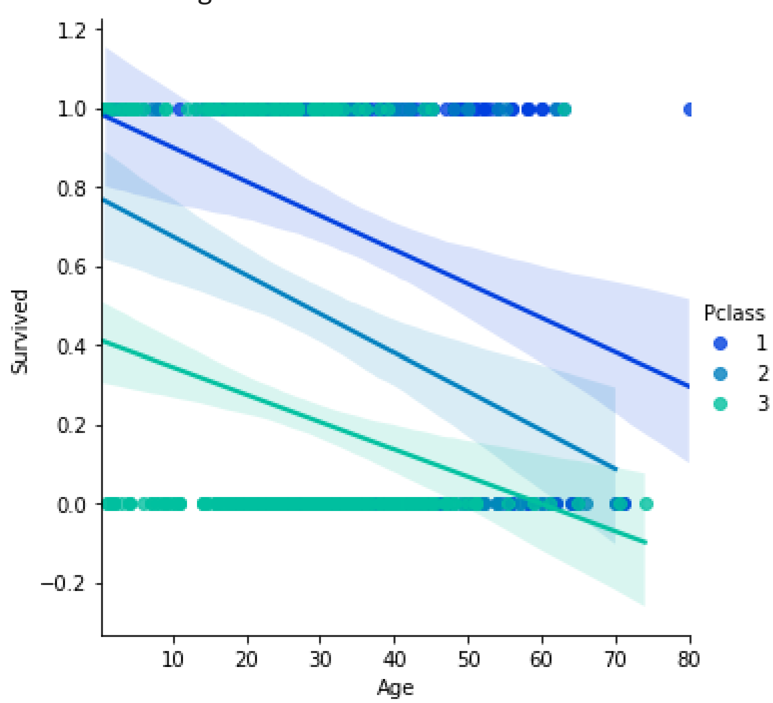
```
## The chances of Survival Decreases with increase in age. "" From the above graph"""
```


```
# Number of passengers  in each class grouped by sex. Also total was found for each class gro
df.pivot_table('Age', 'Sex', 'Pclass', aggfunc=np.sum, margins=True)
```

Pclass	1	2	3	All
Sex				
female	2942.00	2125.50	2218.50	7286.00
male	4169.42	3043.33	6706.42	13919.17
All	7111.42	5168.83	8924.92	21205.17

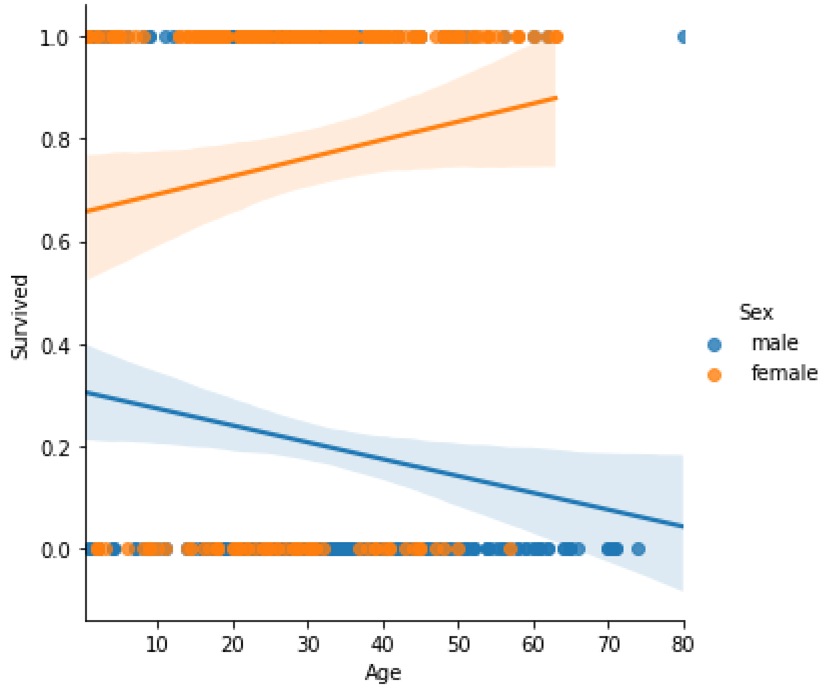
```
sns.lmplot('Age', 'Survived', hue='Pclass', data=df, palette='winter', hue_order=range(1,4))
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0f2b04d860>
```




```
# Survived vs. Age grouped by Sex
sns.lmplot('Age', 'Survived', data=df, hue='Sex')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f0f2ec1ce48>
```

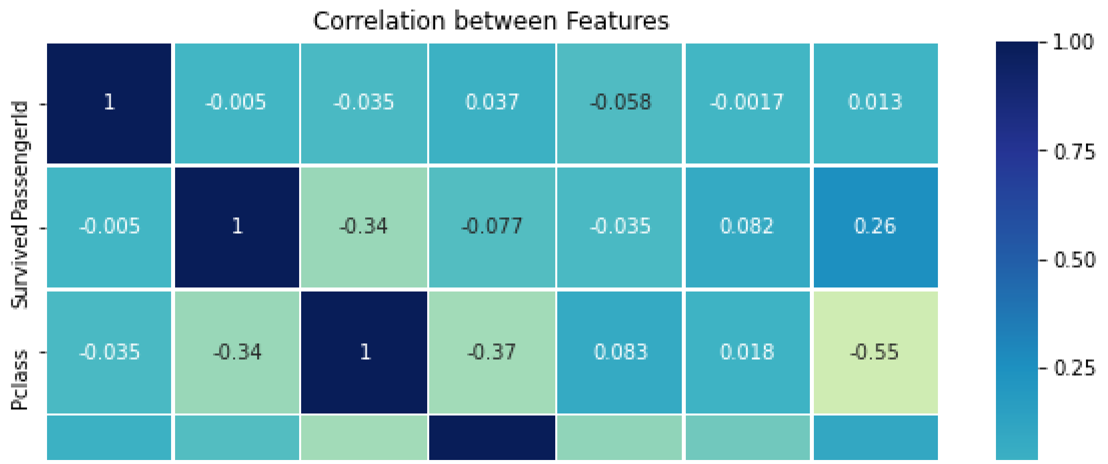


In all three classes, the chance to survive reduced as the passengers got older.

Mens have more Survival Rate than Womens

Plot a HEATMAP showing the correlations between different features:

```
corr = df.corr()


fig,axes = plt.subplots(figsize=(10,8))
sns.heatmap(corr, vmin=-1, vmax=1, annot=True, linewidths=.5, ax=axes, cmap="YlGnBu")
plt.title('Correlation between Features');
```



CONCLUSION

From the above dataset,I analyse that there were many factors on which Survival Rate was dependent Upon:

1) Age (As the age goes higher Survival rete decreases

2) Sex ie Men or women ## As Men has more rate of Survival

3) PClass ie Passenger Class (As these people were based on that part of ship which has maximum damage.Hence more prone and Less survival)