Analysis of Titanic Data

On April 15, 1912, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew.This data set contains demographics and passenger information from 891 of the 2224 passengers and crew on board the Titanic.

Investigation of which factors made people more likely to survive in Titanic disaster is aimed with this analysis.

Questions:

1. Were "age of passengers" an important factor for rescue?
2. Did "ticket fare and Class" have effect on the survival chance?
3. Did "women and children passengers" have more chance to survive from the disaster than "adult men"?
4. Did family size on board help to keep alive?

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)

- 1st = Upper
- 2nd = Middle
- 3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...

- Sibling = brother, sister, stepbrother, stepsister
- Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...

- Parent = mother, father
- Child = daughter, son, stepdaughter, stepson
- Some children travelled only with a nanny, therefore parch=0 for them.

We can start with importing of functional libraries and loading dataset as dataframe:

In [1]:

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

data = pd.read_csv("./titanic-data.csv")
```

We can look through first 5 rows with head() function and get insight on data:

In [2]:

```
data.head()
```

Out[2]:

	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	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

We can get information of number of non-null entries in columns and type of variables:

In [3]:

```
data.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
```

We can look at main properties of numeric columns by "describe":

In [4]:

```
data.describe()
```

Out[4]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.0
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.9
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512

These are summary of findings :

- There are 891 passengers.
- 38.38 % of people was survived.
- More than half of the passengers were from 3rd Class.
- Average age on board was 29.699.
- The oldest people was 80 years old, the youngest was nearly 5 months old.
- Less than 25 % of people has a parent or child.
- Average fare was 32.20 when the most expensive fare was 512.33.

We detected that there are 177 missing values in Age column. We can use .fillna to assign values instead of these missing ones. However this can mislead us for that dataset. So they will be just removed but size of our dataset get smaller.

Also large amount of cabin values, are missing, just 204 values exists. So we can neglect it and loose some information. We can also create new column with cabin values exist or not.

Embarked column has 2 missing values and we can neglect it.

We can use this new cleaned dataset for investigation related with age:

In [5]:

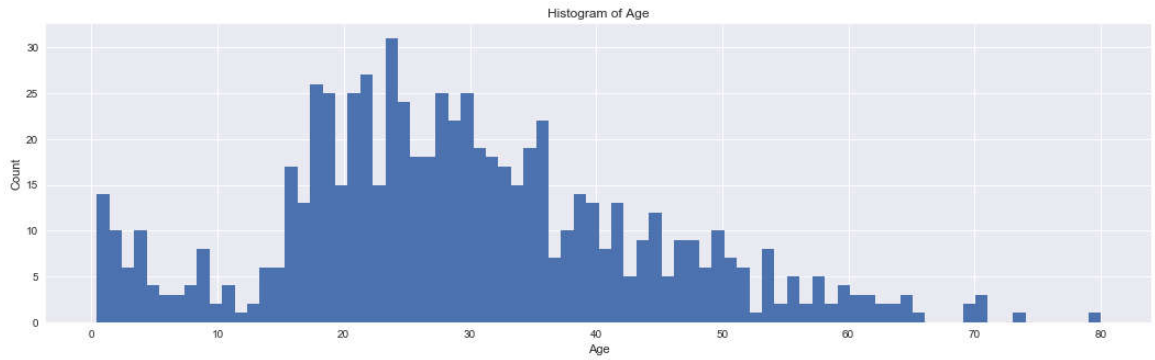
```
clean_data=data.drop(['Cabin'], axis = 1) #Delete "Cabin" column
clean_data = clean_data.dropna(axis=0, how='any') #Delete all rows which any column val
ue is NAN or Null.
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 0 to 890
Data columns (total 11 columns):
PassengerId    712 non-null int64
Survived       712 non-null int64
Pclass         712 non-null int64
Name           712 non-null object
Sex            712 non-null object
Age            712 non-null float64
SibSp          712 non-null int64
Parch          712 non-null int64
Ticket         712 non-null object
Fare           712 non-null float64
Embarked       712 non-null object
dtypes: float64(2), int64(5), object(4)
memory usage: 66.8+ KB
```

Let's create histogram of age values. Each bar is indivual age value due to bins equal to 80.

In [6]:

```
clean_data.hist(column='Age',figsize=(18,5), bins=80)
plt.xlabel("Age")
plt.ylabel("Count")
plt.title("Histogram of Age");
```



Most of the passengers are between 18 and 36 years old. We can also investigate age statistics according to survival.

In [7]:

```
clean_data.groupby('Survived').Age.describe().T
```

Out[7]:

Survived	0	1
count	424.000000	288.000000
mean	30.626179	28.193299
std	14.172110	14.859146
min	1.000000	0.420000
25%	21.000000	19.000000
50%	28.000000	28.000000
75%	39.000000	36.000000
max	74.000000	80.000000

We can use Mann Whitney U Test to determine this two groups are statiscally different from each according to age distribution.

Null hypothesis is that they are same. If p-value is found less than 0.05; we can change our null to alternative one which states they are statistically different.

In [8]:

```
survived_age = clean_data[clean_data['Survived']==1].Age
non_survived_age = clean_data[clean_data['Survived']==0].Age
```

In [9]:

```
from scipy import stats
stats.mannwhitneyu(survived_age,non_survived_age, alternative="two-sided") # alternativ
es: 'less', 'two-sided','greater'
```

Out[9]:

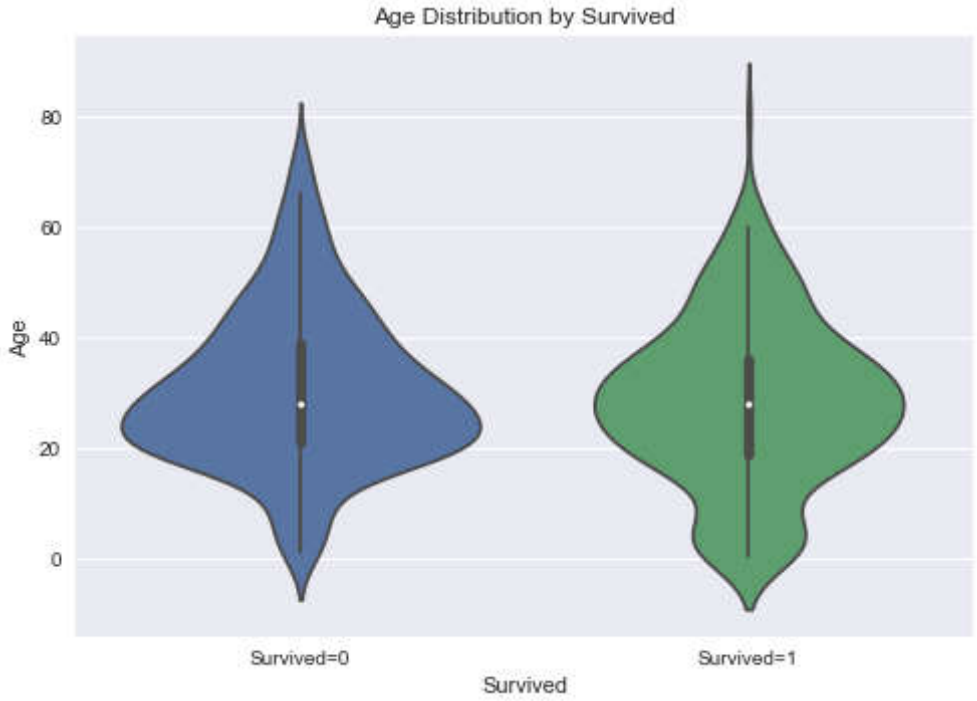
```
MannwhitneyuResult(statistic=56961.0, pvalue=0.12837188836279587)
```

So null hypothesis is preserved and there is no statiscally difference between survived and non-survived people according to age.

We can also visualize age distribution:

In [10]:

```
age_violin = sns.violinplot(data = data, x = 'Survived', y = 'Age')
age_violin.set(title = 'Age Distribution by Survived',
        xlabel = 'Survived',
        ylabel = 'Age',
        xticklabels = ['Survived=0', 'Survived=1']);
```



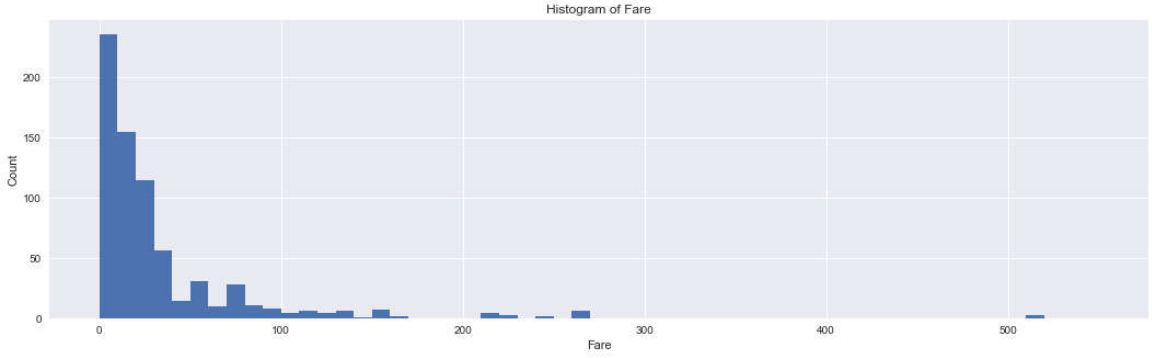
Answer for Question-1:

Age is not important factor solely. Maybe it becomes important by interaction with other attributes.

Let's create histogram of fare values. Each bin is incremented with 10$ value.

In [11]:

```
clean_data.hist(column='Fare',figsize=(18,5), bins=55, range=(0,550))
plt.xlabel("Fare")
plt.ylabel("Count")
plt.title("Histogram of Fare");
```



In [12]:

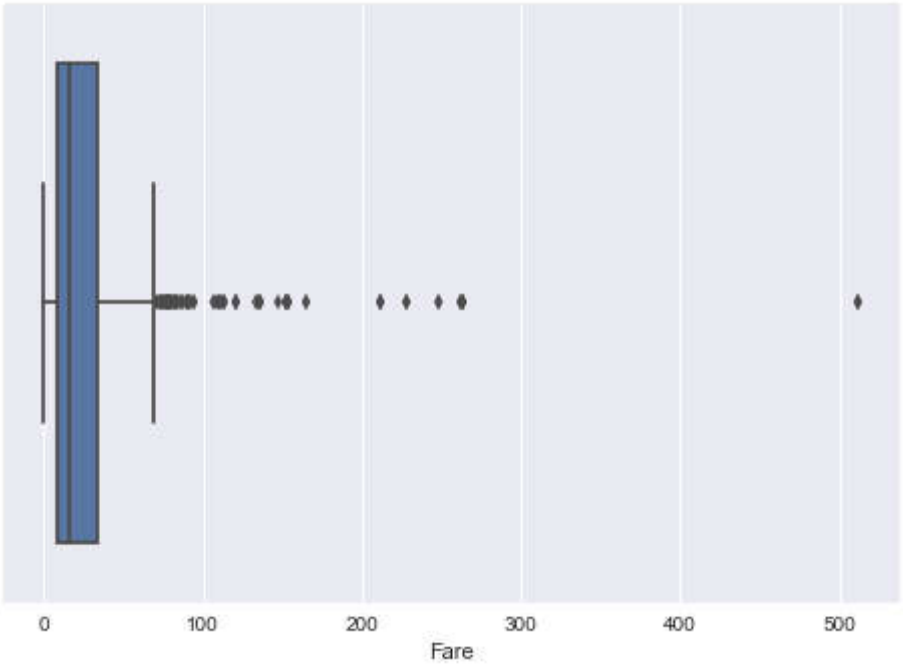
```
clean_data.Fare.describe()
```

Out[12]:

```
count    712.000000
mean      34.567251
std       52.938648
min        0.000000
25%        8.050000
50%       15.645850
75%       33.000000
max      512.329200
Name: Fare, dtype: float64
```

In [13]:

```
ax = sns.boxplot(x="Fare", data=clean_data)
```



Most of the passengers paid for tickets less than 30$.There are some outliers in fare attribute. We can clasify tickets as normal(<=30), expensive (30-80) and outlier(>=80); and then investigate survival rates of passengers who bought those tickets:

In [14]:

```
def fare_status(Fare):
    if Fare  > 80:
        return 'Outlier'
    elif Fare  > 30:
        return 'Expensive'
    else:
        return 'Normal'
```

In [15]:

```
clean_data['Fare_Status'] = clean_data['Fare'].apply(fare_status)
```

In [16]:

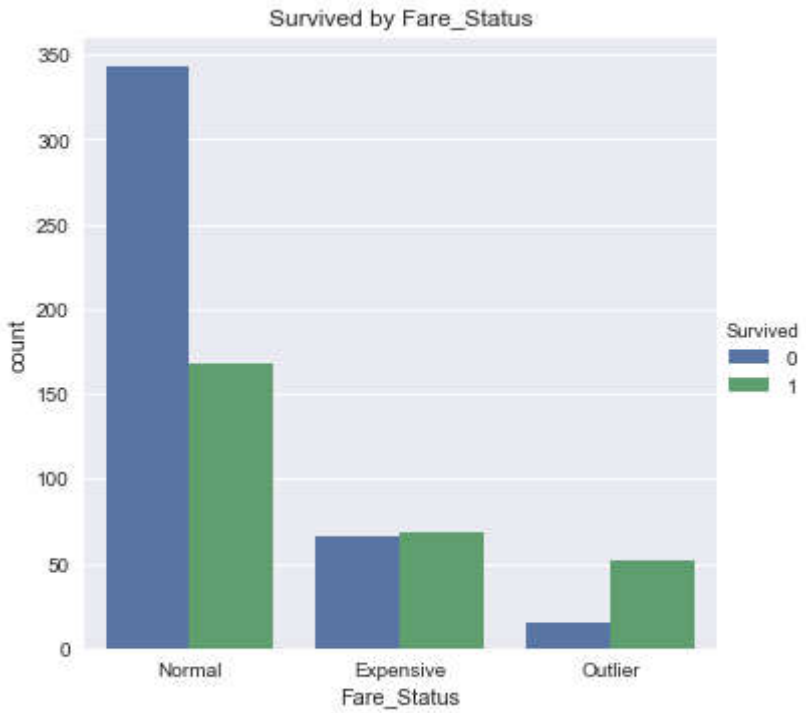
```
clean_data.head()
```

Out[16]:

	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	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

In [17]:

```
sns.factorplot('Fare_Status', hue = "Survived", size = 5, data = clean_data, kind = 'co
unt')
plt.title('Survived by Fare_Status');
```



In [18]:

```
clean_data.groupby('Fare_Status')["Survived"].describe().T
```

Out[18]:

Fare_Status	Expensive	Normal	Outlier
count	134.000000	511.000000	67.000000
mean	0.507463	0.328767	0.776119
std	0.501820	0.470225	0.419989
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	1.000000
50%	1.000000	0.000000	1.000000
75%	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000

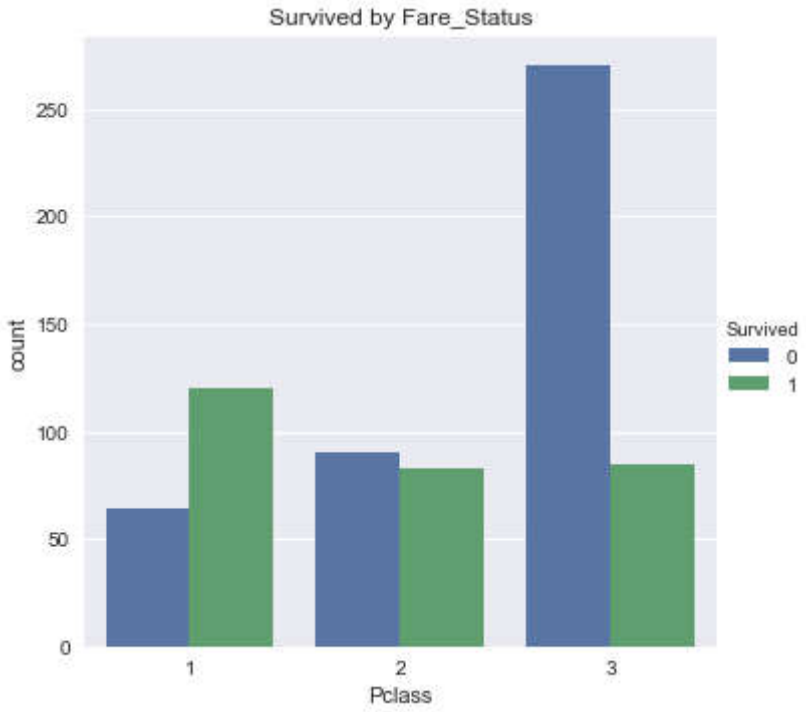
As seen from factorplot, fare is related with survival rates. The higher fare class, the higher survival chance.

Survival rates of Normal,Expensive and Outlier fares are 32.9%, 50.7% and 77.6% respectively.

Let's look at the Ticket Class also:

In [19]:

```
sns.factorplot('Pclass', hue = "Survived", size = 5, data = clean_data, kind = 'count')
plt.title('Survived by Fare_Status');
```



In [20]:

```
clean_data.groupby('Pclass')["Survived"].describe().T
```

Out[20]:

Pclass	1	2	3
count	184.000000	173.000000	355.000000
mean	0.652174	0.479769	0.239437
std	0.477580	0.501041	0.427342
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000
75%	1.000000	1.000000	0.000000
max	1.000000	1.000000	1.000000

As seen from factorplot, ticket class is related with survival rates. The higher ticket class (Class-1 is higher,Class-3 is lower), the higher survival chance.

Survival rates of Class-1,Class-2 and Class-3 tickets are 65.22%, 47.98% and 23.94% respectively.

We can also look at survival rates group by Fare_Status and Ticket Fare together:

In [21]:

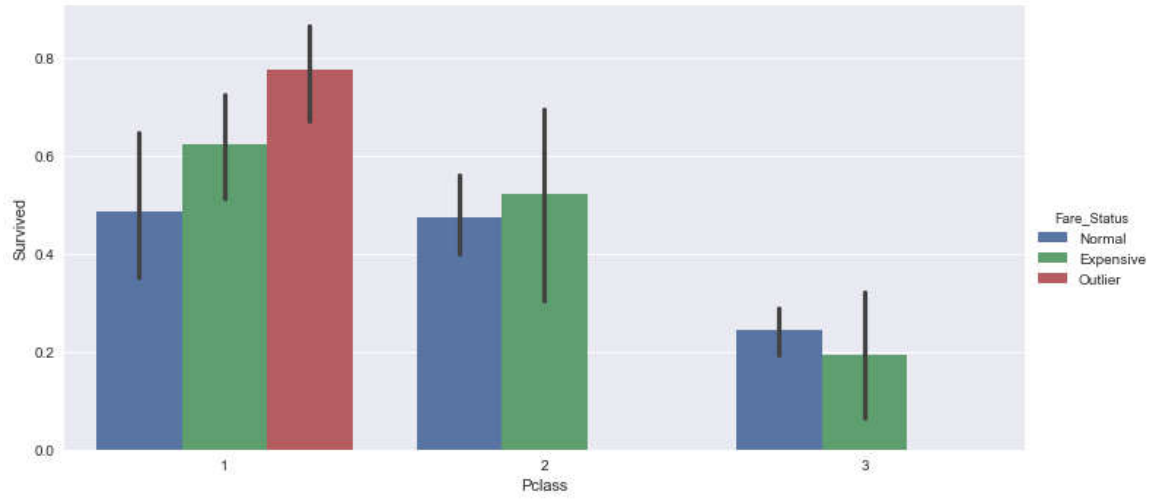
```
clean_data.groupby(['Fare_Status',"Pclass"])["Survived"].describe().T
```

Out[21]:

Fare_Status	Expensive			Normal			Ou
Pclass	1	2	3	1	2	3	1
count	80.000000	23.000000	31.000000	37.000000	150.000000	324.000000	67
mean	0.625000	0.521739	0.193548	0.486486	0.473333	0.243827	0.7
std	0.487177	0.510754	0.401610	0.506712	0.500961	0.430054	0.4
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.0
50%	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.0
75%	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	1.0
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

In [22]:

```
sns.factorplot(x="Pclass", y="Survived", hue="Fare_Status",data=clean_data, kind="bar",
 size=5, aspect=2);
```



It is seen difference for Pclass1 and PClass2 passengers whether they paid expensive or normal rate. We need to check that they are statistically different.

Null hypothesis of Mann Whitney U test is both sample are same. If there is statistically difference, p-value should be less than 0.05.

In [23]:

```
exp_class1 = clean_data[(clean_data['Pclass']==1) & (clean_data['Fare_Status']=="Expens
ive")].Survived
norm_class1 = clean_data[(clean_data['Pclass']==1) & (clean_data['Fare_Status']=="Norma
l")].Survived
```

In [24]:

```
stats.mannwhitneyu(exp_class1,norm_class1, alternative="two-sided") # alternatives: 'le
ss', 'two-sided','greater'
```

Out[24]:

```
MannwhitneyuResult(statistic=1685.0, pvalue=0.16071676824049996)
```

According to test result, Expensive and Normal Rate of Fare statistically do not create difference for Class-1 passengers.

In [25]:

```
exp_class2 = clean_data[(clean_data['Pclass']==2) & (clean_data['Fare_Status']=="Expens
ive")].Survived
norm_class2 = clean_data[(clean_data['Pclass']==2) & (clean_data['Fare_Status']=="Norma
l")].Survived
```

In [26]:

```
stats.mannwhitneyu(exp_class2,norm_class2, alternative="two-sided")
```

Out[26]:

```
MannwhitneyuResult(statistic=1808.5, pvalue=0.66803451018574633)
```

According to test result, Expensive or Normal Rate of Fare statistically do not create difference for Class-2 passengers.

Answer for Question-2:

Fare and Ticket class could be important factor to survive. Maybe it was related with priority to transfer to boats and location, but result is tend to passenger with expensive tickets were more likely to survive disaster.

Let's organize "Sex" column by defining passengers less than 12 years old as a child.

In [27]:

```
def to_child(passenger):
    age,sex = passenger
    if age <= 12:
        return 'child'
    else:
        return sex

clean_data['Sex'] = clean_data[['Age','Sex']].apply(to_child,axis=1)
```

In [28]:

```
clean_data.tail()
```

Out[28]:

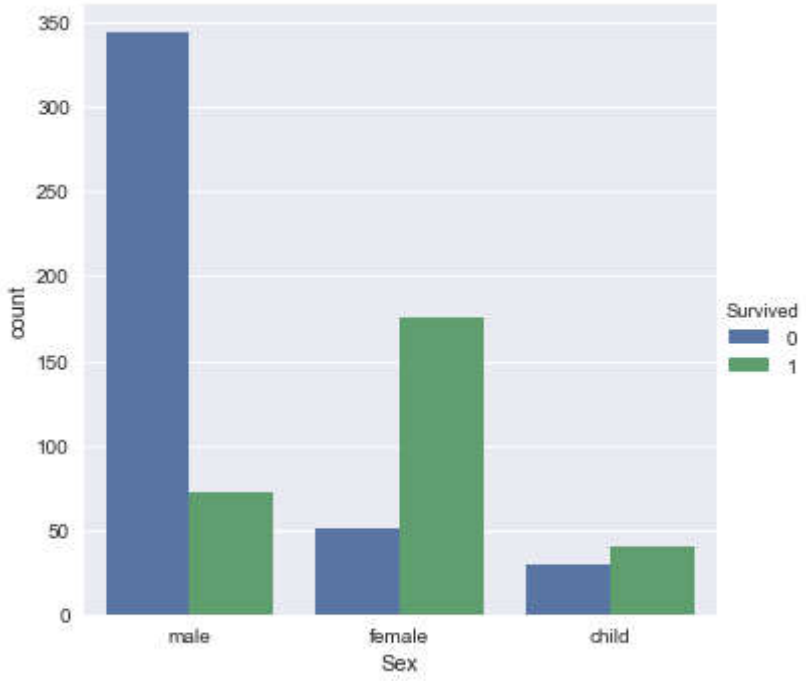
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

In [29]:

```
sns.factorplot('Sex', hue = "Survived", size = 5, data = clean_data, kind = 'count')
```

Out[29]:

```
<seaborn.axisgrid.FacetGrid at 0xbe93cf8>
```



In that structure, we can observe difference for females but not obvious for children.

Ticket class difference may affect this distribution. So we can group by ticket class and sex to observe:

In [30]:

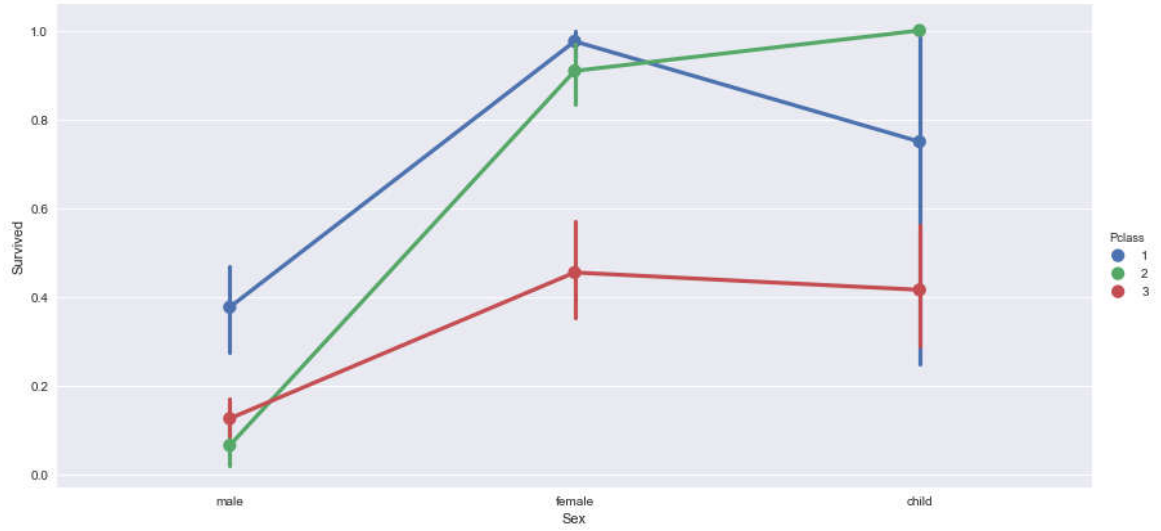
```
clean_data.groupby(['Sex',"Pclass"])["Survived"].describe().T
```

Out[30]:

Sex	**child**			**female**			**male**	
Pclass	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**
count	4.00	17.0	48.000000	82.000000	66.000000	79.000000	98.000000	90.000000
mean	0.75	1.0	0.416667	0.975610	0.909091	0.455696	0.377551	0.066667
std	0.50	0.0	0.498224	0.155207	0.289683	0.501216	0.487267	0.250841
min	0.00	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.75	1.0	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000
50%	1.00	1.0	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000
75%	1.00	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
max	1.00	1.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [31]:

```
sns.factorplot(x="Sex", y="Survived", hue="Pclass", data=clean_data, size=6, aspect=2);
```



Answer for Question-3:

Ladies and children may had priority to be placed boats on that day. Factorplot helps us to observe easily.

Ladies and children are more likely to survive, especially with class 1 and 2 tickets.

To answer 4th question, we will create new column as "Family_Size" by summing siblings-spouses and parents-child:

In [32]:

```
clean_data["Family_Size"] = clean_data["SibSp"]+clean_data["Parch"]
```

In [33]:

```
clean_data.head()
```

Out[33]:

	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	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	5
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

We can obtain survival rate of different family sizes by using groupby and describe:

In [34]:

```
clean_data.groupby(['Family_Size'])["Survived"].describe().T
```

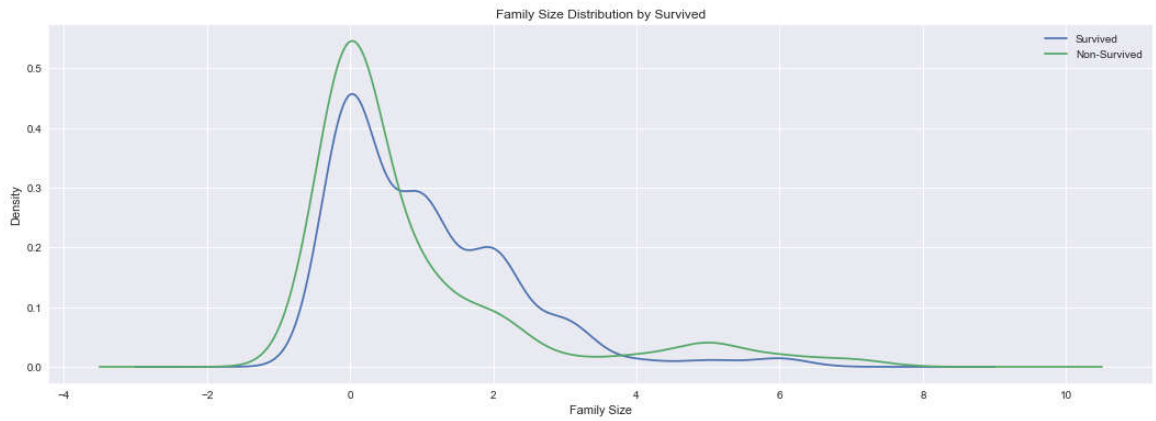
Out[34]:

Family_Size	0	1	2	3	4	5	
count	402.000000	139.000000	93.000000	27.000000	11.000000	22.000000	12.
mean	0.318408	0.546763	0.569892	0.777778	0.272727	0.136364	0.3
std	0.466439	0.499609	0.497774	0.423659	0.467099	0.351250	0.4
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.0
50%	0.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.0
75%	1.000000	1.000000	1.000000	1.000000	0.500000	0.000000	1.0
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0

According to this result, most survival rate is 77.78 % with Family Size 3. However sample size is small. Continue to investigation by kernel density estimate:

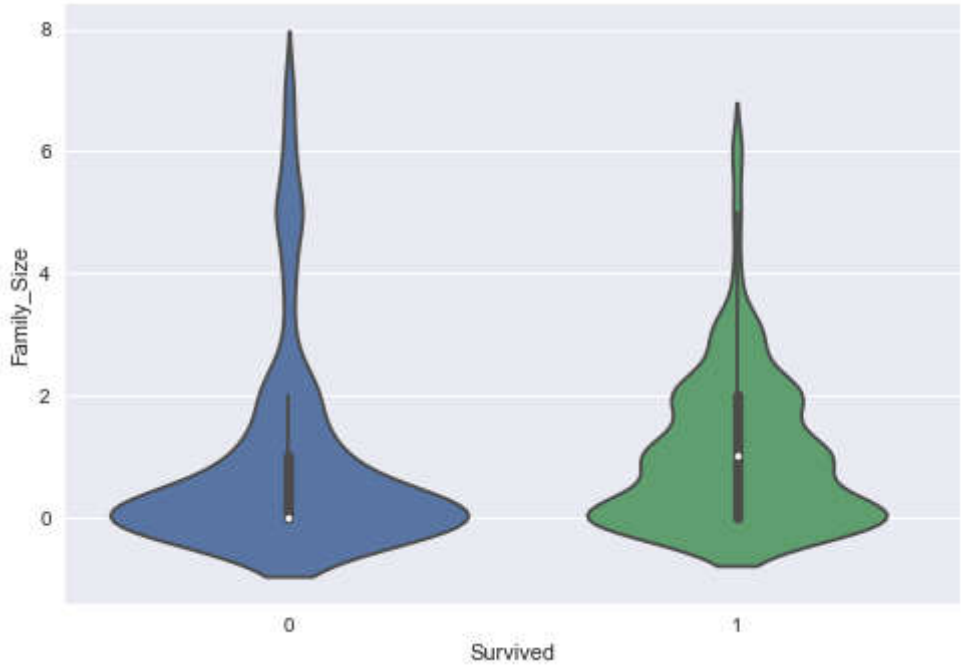
In [35]:

```
clean_data.Family_Size[clean_data.Survived == 1].plot(kind='kde', figsize=(18,6))
clean_data.Family_Size[clean_data.Survived == 0].plot(kind='kde', figsize=(18,6))
plt.title("Family Size Distribution by Survived")
plt.xlabel("Family Size")
plt.legend(('Survived','Non-Survived'),loc='best')
plt.grid(b=True)
```



In [36]:

```
sns.violinplot(data = clean_data, x = 'Survived', y = 'Family_Size');
```



Due to the smaller sample sizes, comparing of Family Sizes does not provide us significant information. So we can aggregate Family Sizes with "1"(With family) and "0"(Without family) as two main category.

In [37]:

```
def to_family(passenger):
    if passenger>0:
        return 1
    else:
        return 0

clean_data['Family'] = clean_data["Family_Size"].apply(to_family)
```

In [38]:

```
clean_data.tail()
```

Out[38]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	I
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

We can obtain survival rate of passengers with or without family members by using groupby and describe:

In [39]:

```
clean_data.groupby(['Family'])["Survived"].describe().T
```

Out[39]:

Family	0	1
count	402.000000	310.000000
mean	0.318408	0.516129
std	0.466439	0.500548
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	1.000000
75%	1.000000	1.000000
max	1.000000	1.000000

According to result, 51.61 % of passengers with family member survived from disaster while 31.84 % of passengers without family member survived.

Also check with Mann Whitney U test:

In [40]:

```
without_family = clean_data[clean_data['Family']==0].Survived
with_family = clean_data[clean_data['Family']==1].Survived
```

In [41]:

```
stats.mannwhitneyu(without_family,with_family, alternative="two-sided")
```

Out[41]:

```
MannwhitneyuResult(statistic=49990.0, pvalue=1.0050981924035183e-07)
```

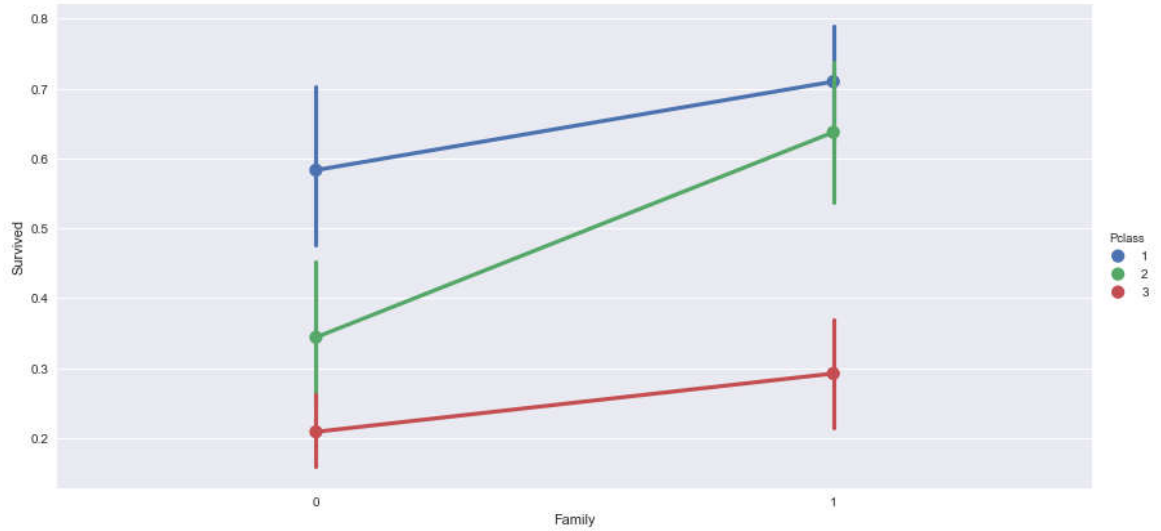
According to test result, null hypothesis will change to alternative.

Statistically there is difference between travel with at least one family member and without any family member.

We can also observe interaction with Family and Pclass on Survival rate:

In [42]:

```
sns.factorplot(x="Family", y="Survived", hue="Pclass", data=clean_data, size=6, aspect=
2);
```



and also interaction with Family and Sex_C on Survival rate: