In [1]:

```
import pandas as pd
import numpy as np

test_data = pd.read_csv("test.csv")
df_test_data = pd.DataFrame(test_data)
df_test_data
```

Out[1]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cal
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	N
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	N
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	N
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	N
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	N
...	...	...	...	...	...	...	...	...	...	
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	N
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C1
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	N
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	N
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	N

418 rows × 11 columns

```
A) number of rows in training and test sets
```

In [2]:

```
#A) number of rows in training and test sets

print(df_test_data.count())
print("\nNumber of Rows: ",df_test_data.shape)
```

```
PassengerId    418
Pclass         418
Name           418
Sex            418
Age            332
SibSp          418
Parch          418
Ticket         418
Fare           417
Cabin           91
Embarked       418
dtype: int64

Number of Rows:  (418, 11)
```

In [3]:

```
# display the structure of the dataset along with the datatypes of the fields

df_test_data.dtypes
```

Out[3]:

```
PassengerId      int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [4]:

```
train_data = pd.read_csv("train.csv")
df_train_data = pd.DataFrame(train_data)
df_train_data
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

891 rows × 12 columns

In [5]:

```
#A) number of rows in training and test sets

print(df_train_data.count())
print("\nNumber of Rows: ",df_train_data.shape)
```

```
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

Number of Rows:  (891, 12)
```

In [6]:

```
# display the structure of the dataset along with the datatypes of the fields
df_train_data.dtypes
```

Out[6]:

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

#Data Cleaning: #1. Analyse the data and identify which columns are not relevant for survivor prediction task. Drop those columns from the dataframes. #2. Check how many columns have missing values in them (NA) and how many have NaN values. Logically impute the dataset. #3. Identify any categorical valued columns (non-numeric) and convert them to numeric.

```
PassengerID is not relevent
ticket is not relevent
```

In [7]:

```
# Ticket column has been drop
df_train_data
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.28
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.92
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.10
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

891 rows × 12 columns

In [8]:

```
df_train_data.drop("PassengerId", inplace=True,axis=1)
```

In [9]:

```
# passengerID has been drop
df_train_data
```

Out[9]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

891 rows × 11 columns

In [10]:

```
# Removing NAN values from Age Column

df_train_data.dropna(subset=["Age"])
```

Out[10]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	En
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	
885	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

714 rows × 11 columns

In [11]:

```
# Age Converted from float to int

# df_train_data['Age'] = df_train_data['Age'].fillna(0).astype(float)

df_train_data
```

Out[11]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	
...	...	...	...	...	...	...	...	...	...	...	
886	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	
887	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	
888	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	
889	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	
890	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	

891 rows × 11 columns

In [12]:

```
df_train_data['Age'].dtypes
```

Out[12]:

```
dtype('float64')
```

In [13]:

```
df_train_data['Age'].shape
```

Out[13]:

```
(891,)
```

1. Show how many passengers were male and female and plot using matplotlib. On the same plot depict the people who survived and who died. Make accurate axis and legend. Save the plot in a png file.

In [14]:

```
import matplotlib.pyplot as plt
```

In [15]:

```
gender = df_train_data["Sex"].tolist()

total_male = gender.count("male")
total_female = gender.count("female")

print("total Male : ",total_male)
print("total Female : ",total_female)
```

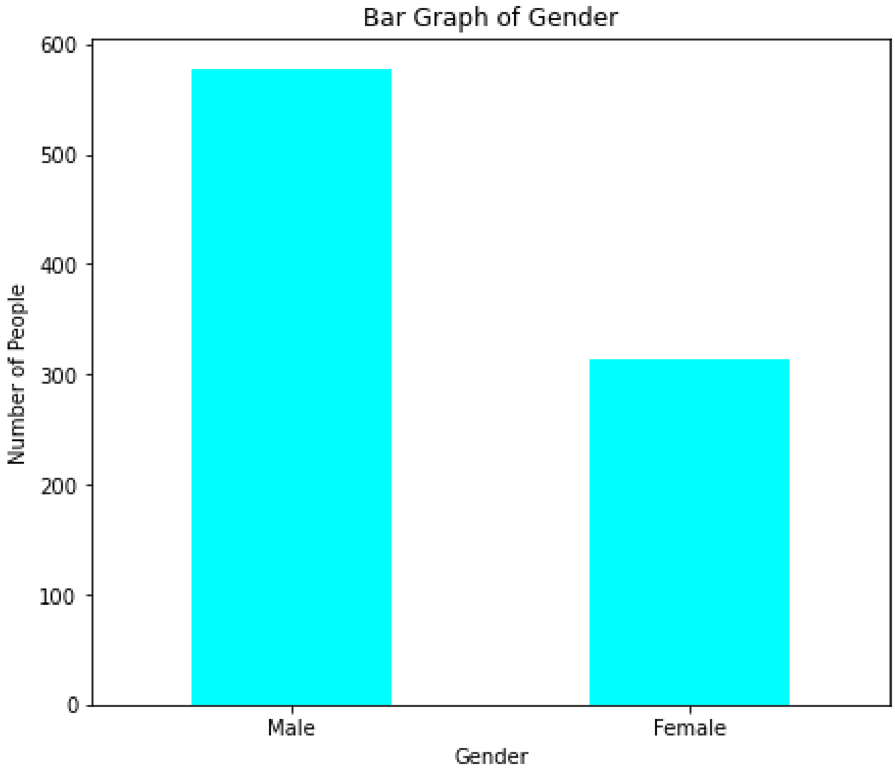
```
total Male :  577
total Female :  314
```

In [16]:

```
gender_count = df_train_data['Sex'].value_counts()
plt.figure(figsize=(7, 6))
ax = gender_count.plot(kind='bar', rot=0, color="cyan")
ax.set_title("Bar Graph of Gender", y = 1)
ax.set_xlabel('Gender')
ax.set_ylabel('Number of People')
ax.set_xticklabels(('Male', 'Female'))
```

Out[16]:

```
[Text(0, 0, 'Male'), Text(1, 0, 'Female')]
```

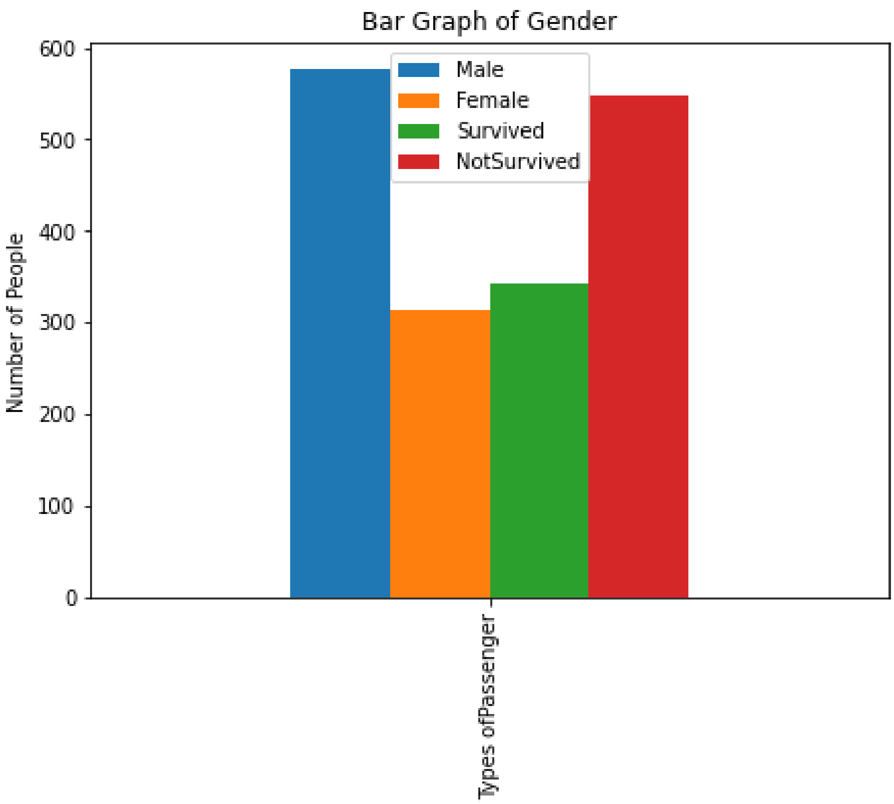


In [17]:

```
#1. Show how many passengers were male and female and plot using matplotlib. On the same 
# the people who survived and who died. Make accurate axis and legend. Save the plot in a

male=len(df_train_data.query("Sex == 'male'"))
female=len(df_train_data.query("Sex != 'male'"))
Survived=len(df_train_data.query("Survived == 1"))
notSurvived=len(df_train_data.query("Survived == 0"))
df = pd.DataFrame({'Male': male, 'Female': female,'Survived':Survived,'NotSurvived':notSu
ax = df.plot.bar(figsize=(7,5),rot=plt.savefig("Q1.png"))
ax.set_title("Bar Graph of Gender", y = 1)
ax.set_ylabel('Number of People')
plt.show()
```

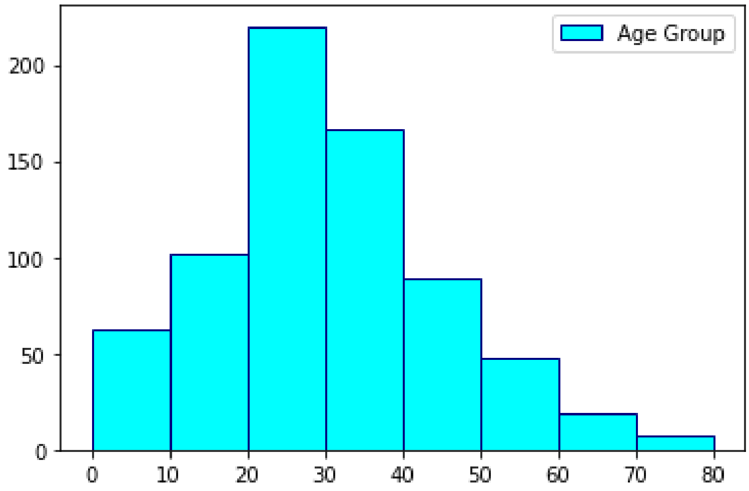
```
<Figure size 432x288 with 0 Axes>
```



Q.2 - Show the histogram of the count of passengers who died (according to their age). Age ranges should be <10, 10 to <20, 20 to <30 and so on. How many minor children died and how many of them survived (<16 years). Create a separate plot for the passengers who survived.

In [18]:

```
age=df_train_data['Age']
plt.hist(age,bins=[0,10,20,30,40,50,60,70,80],edgecolor="navy",label='Age Group',color =
plt.legend()
plt.xlabel="Age Group"
plt.ylabel="Number of Passengar"
plt.savefig("Ques2-P-1.png")
plt.show()
```



In [19]:

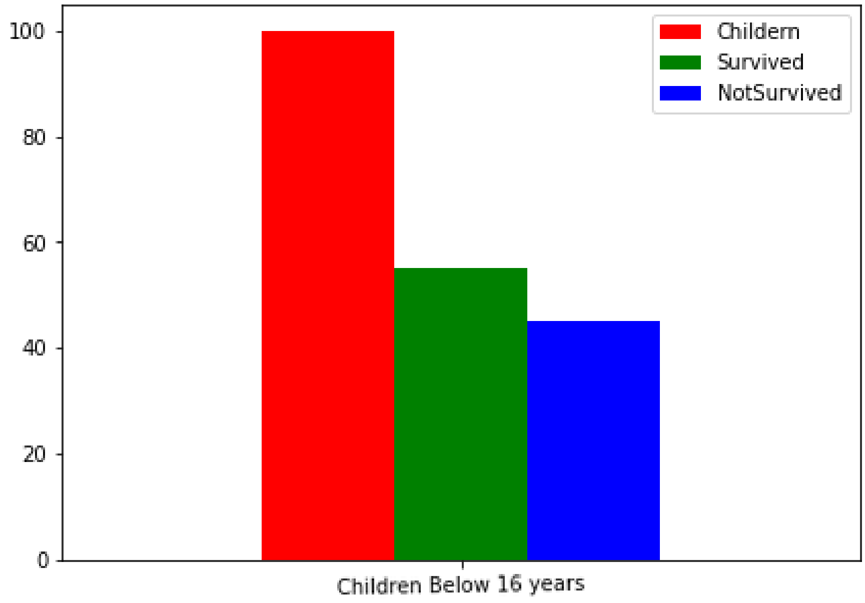
```
#How many minor children died and how many of them survived (<16 years). Create a separat

childAge=pd.DataFrame(df_train_data.query("Age <= 16"))
child=len(childAge)
Survived=len(childAge.query("Survived == 1"))
notSurvived=len(childAge.query("Survived == 0 "))
print("Number of children below 16 years - " , child )
print("total survived - " , Survived)
print("total Deaths - " ,notSurvived)
```

```
Number of children below 16 years -  100
total survived -  55
total Deaths -  45
```

In [20]:

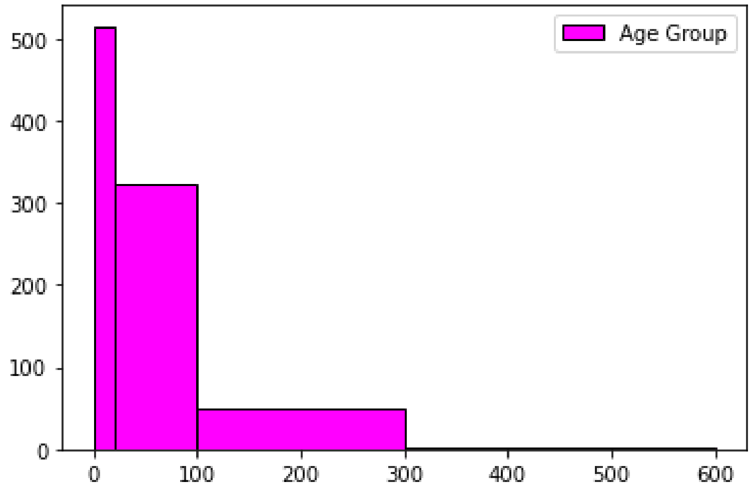
```
df = pd.DataFrame({'Childern': child, 'Survived': Survived,'NotSurvived':notSurvived},ind
ax = df.plot.bar(figsize=(7,5),rot=1,color='rgb')
```



3. Show the distribution on the count of passengers who died (according to the fare they paid). Choose fare ranges such that the mean lies in the middle range. Give the percentage of passengers who survived as had paid more than $100. Justify if there was any bias in the rescue operation towards the rich (Yes/No/not enough evidence).

In [22]:

```
df_train_data['Fare'].mean()
fare=df_train_data['Fare']
plt.hist(fare,bins=[0,20,100,300,600],edgecolor="black",label='Age Group',color='fuchsia'
plt.legend()
plt.xlabel="Age Group"
plt.ylabel="Number of Passengar"
plt.savefig("Q2a.png")
plt.show()
```

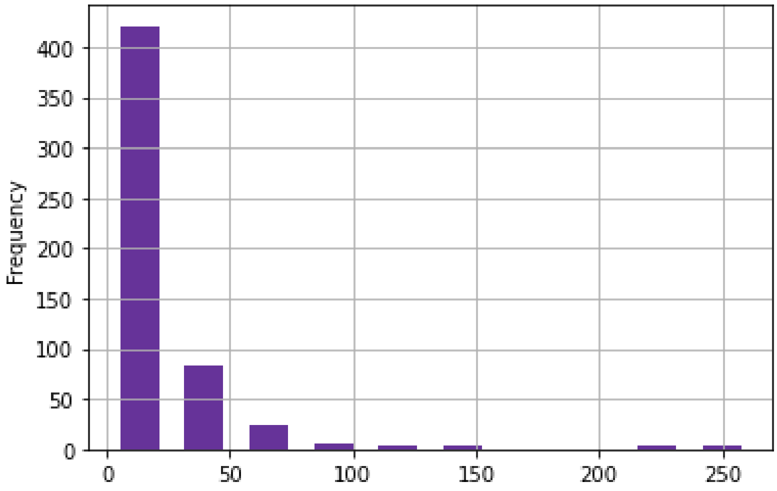


In [23]:

```
pd = df_train_data.query("Survived == 0")
#Survived = 0 represents the people who died.
pd=pd[['Fare','Survived']].copy()
pd
pd['Fare'].plot.hist(grid=True,  rwidth = 0.6 ,color='rebeccapurple')
```

Out[23]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [24]:

```
morefare = df_train_data.query('Fare>=100')
moreandsur = df_train_data.query('Survived ==1 and Fare>=100')
percentage= (len(moreandsur)/len(morefare))*100
print(percentage," % of the passengers survived after paying more than 100$.")
print("its looking biased. ")
```

```
73.58490566037736  % of the passengers survived after paying more than 100
$.
its looking biased.
```

5. Find the number of passengers who were married

In [25]:

```
import re
count=0
for i in df_train_data['Name']:
    if(re.findall("Mrs", i)):
        count=count+1
print("The Number of married couples were:",count)
```

```
The Number of married couples were: 129
```

In [ ]:

```

```