Project - Investigate a Dataset

January 3, 2016

1 Investigate a Dataset

The following data analysis is done on the titanic data about the passenger provided by kaggle which contains demographics and passenger information from 891 of the 2224 passengers and crew on board the Titanic. This will consist of stating few questions and addressing them in the rest of the analysis process.

1.0.1 Below is the dataset of the Titanic passenger information

```
In [1]: import pandas
        df = pandas.read_csv('titanic_data.csv')
        df
```

Out[1]:

	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	
6	7	0	1	
7	8	0	3	
8	9	1	3	
9	10	1	2	
10	11	1	3	
11	12	1	1	
12	13	0	3	
13	14	0	3	
14	15	0	3	
15	16	1	2	
16	17	0	3	
17	18	1	2	
18	19	0	3	
19	20	1	3	
20	21	0	2	
21	22	1	2	
22	23	1	3	
23	24	1	1	
24	25	0	3	
25	26	1	3	
26	27	0	3	
27	28	0	1	
28	29	1	3	
29	30	0	3	

..	...	...	...
861	862	0	2
862	863	1	1
863	864	0	3
864	865	0	2
865	866	1	2
866	867	1	2
867	868	0	1
868	869	0	3
869	870	1	3
870	871	0	3
871	872	1	1
872	873	0	1
873	874	0	3
874	875	1	2
875	876	1	3
876	877	0	3
877	878	0	3
878	879	0	3
879	880	1	1
880	881	1	2
881	882	0	3
882	883	0	3
883	884	0	2
884	885	0	3
885	886	0	3
886	887	0	2
887	888	1	1
888	889	0	3
889	890	1	1
890	891	0	3

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	
2	Heikkinen, Miss. Laina	female	26	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	
4	Allen, Mr. William Henry	male	35	0	
5	Moran, Mr. James	male	NaN	0	
6	McCarthy, Mr. Timothy J	male	54	0	
7	Palsson, Master. Gosta Leonard	male	2	3	
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	
10	Sandstrom, Miss. Marguerite Rut	female	4	1	
11	Bonnell, Miss. Elizabeth	female	58	0	
12	Saundercock, Mr. William Henry	male	20	0	
13	Andersson, Mr. Anders Johan	male	39	1	
14	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	
15	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	
16	Rice, Master. Eugene	male	2	4	
17	Williams, Mr. Charles Eugene	male	NaN	0	
18	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31	1	
19	Masselmani, Mrs. Fatima	female	NaN	0	
20	Fynney, Mr. Joseph J	male	35	0	

21	Beesley, Mr. Lawrence	male	34	0	
22	McGowan, Miss. Anna "Annie"	female	15	0	
23	Sloper, Mr. William Thompson	male	28	0	
24	Palsson, Miss. Torborg Danira	female	8	3	
25	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38	1	
26	Emir, Mr. Farred Chehab	male	NaN	0	
27	Fortune, Mr. Charles Alexander	male	19	3	
28	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	
29	Todoroff, Mr. Lalio	male	NaN	0	
..	...	...	...	...	
861	Giles, Mr. Frederick Edward	male	21	1	
862	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48	0	
863	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	
864	Gill, Mr. John William	male	24	0	
865	Bystrom, Mrs. (Karolina)	female	42	0	
866	Duran y More, Miss. Asuncion	female	27	1	
867	Roebling, Mr. Washington Augustus II	male	31	0	
868	van Melkebeke, Mr. Philemon	male	NaN	0	
869	Johnson, Master. Harold Theodor	male	4	1	
870	Balkic, Mr. Cerin	male	26	0	
871	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47	1	
872	Carlsson, Mr. Frans Olof	male	33	0	
873	Vander Cruyssen, Mr. Victor	male	47	0	
874	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28	1	
875	Najib, Miss. Adele Kiamie "Jane"	female	15	0	
876	Gustafsson, Mr. Alfred Ossian	male	20	0	
877	Petroff, Mr. Nedelio	male	19	0	
878	Laleff, Mr. Kristo	male	NaN	0	
879	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56	0	
880	Shelley, Mrs. William (Imanita Parrish Hall)	female	25	0	
881	Markun, Mr. Johann	male	33	0	
882	Dahlberg, Miss. Gerda Ulrika	female	22	0	
883	Banfield, Mr. Frederick James	male	28	0	
884	Sutehall, Mr. Henry Jr	male	25	0	
885	Rice, Mrs. William (Margaret Norton)	female	39	0	
886	Montvila, Rev. Juozas	male	27	0	
887	Graham, Miss. Margaret Edith	female	19	0	
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	
889	Behr, Mr. Karl Howell	male	26	0	
890	Dooley, Mr. Patrick	male	32	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
6	0	17463	51.8625	E46	S
7	1	349909	21.0750	NaN	S
8	2	347742	11.1333	NaN	S
9	0	237736	30.0708	NaN	C
10	1	PP 9549	16.7000	G6	S
11	0	113783	26.5500	C103	S

```
12       0         A/5. 2151    8.0500          NaN        S
13       5            347082   31.2750          NaN        S
14       0            350406    7.8542          NaN        S
15       0            248706   16.0000          NaN        S
16       1            382652   29.1250          NaN        Q
17       0            244373   13.0000          NaN        S
18       0            345763   18.0000          NaN        S
19       0              2649    7.2250          NaN        C
20       0            239865   26.0000          NaN        S
21       0            248698   13.0000          D56        S
22       0            330923    8.0292          NaN        Q
23       0            113788   35.5000           A6        S
24       1            349909   21.0750          NaN        S
25       5            347077   31.3875          NaN        S
26       0              2631    7.2250          NaN        C
27       2             19950  263.0000  C23 C25 C27        S
28       0            330959    7.8792          NaN        Q
29       0            349216    7.8958          NaN        S
..     ...               ...       ...          ...      ...
861      0             28134   11.5000          NaN        S
862      0             17466   25.9292          D17        S
863      2          CA. 2343   69.5500          NaN        S
864      0            233866   13.0000          NaN        S
865      0            236852   13.0000          NaN        S
866      0     SC/PARIS 2149   13.8583          NaN        C
867      0          PC 17590   50.4958          A24        S
868      0            345777    9.5000          NaN        S
869      1            347742   11.1333          NaN        S
870      0            349248    7.8958          NaN        S
871      1             11751   52.5542          D35        S
872      0               695    5.0000  B51 B53 B55        S
873      0            345765    9.0000          NaN        S
874      0         P/PP 3381   24.0000          NaN        C
875      0              2667    7.2250          NaN        C
876      0              7534    9.8458          NaN        S
877      0            349212    7.8958          NaN        S
878      0            349217    7.8958          NaN        S
879      1             11767   83.1583          C50        C
880      1            230433   26.0000          NaN        S
881      0            349257    7.8958          NaN        S
882      0              7552   10.5167          NaN        S
883      0  C.A./SOTON 34068   10.5000          NaN        S
884      0   SOTON/OQ 392076    7.0500          NaN        S
885      5            382652   29.1250          NaN        Q
886      0            211536   13.0000          NaN        S
887      0            112053   30.0000          B42        S
888      2        W./C. 6607   23.4500          NaN        S
889      0            111369   30.0000         C148        C
890      0            370376    7.7500          NaN        Q

[891 rows x 12 columns]
```

1.0.2 We try to answer the following question for the given data

- What factors made people more likely to survive?

1.0.3 Data Cleaning

Before analysing the data we need to remove the data which is not needed for answering the given question. So the Name, Ticket, fare and Embarked in the data is not useful for answering the question since the survival of the passenger does not depend on it. So we remove the following columns from the given data.

```
In [2]: df = df.drop(['Name', 'Ticket', 'Fare', 'Embarked'], axis=1)
        df
```

Out[2]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin
0	1	0	3	male	22	1	0	NaN
1	2	1	1	female	38	1	0	C85
2	3	1	3	female	26	0	0	NaN
3	4	1	1	female	35	1	0	C123
4	5	0	3	male	35	0	0	NaN
5	6	0	3	male	NaN	0	0	NaN
6	7	0	1	male	54	0	0	E46
7	8	0	3	male	2	3	1	NaN
8	9	1	3	female	27	0	2	NaN
9	10	1	2	female	14	1	0	NaN
10	11	1	3	female	4	1	1	G6
11	12	1	1	female	58	0	0	C103
12	13	0	3	male	20	0	0	NaN
13	14	0	3	male	39	1	5	NaN
14	15	0	3	female	14	0	0	NaN
15	16	1	2	female	55	0	0	NaN
16	17	0	3	male	2	4	1	NaN
17	18	1	2	male	NaN	0	0	NaN
18	19	0	3	female	31	1	0	NaN
19	20	1	3	female	NaN	0	0	NaN
20	21	0	2	male	35	0	0	NaN
21	22	1	2	male	34	0	0	D56
22	23	1	3	female	15	0	0	NaN
23	24	1	1	male	28	0	0	A6
24	25	0	3	female	8	3	1	NaN
25	26	1	3	female	38	1	5	NaN
26	27	0	3	male	NaN	0	0	NaN
27	28	0	1	male	19	3	2	C23 C25 C27
28	29	1	3	female	NaN	0	0	NaN
29	30	0	3	male	NaN	0	0	NaN
..	...	...	...	...	...	...	...	...
861	862	0	2	male	21	1	0	NaN
862	863	1	1	female	48	0	0	D17
863	864	0	3	female	NaN	8	2	NaN
864	865	0	2	male	24	0	0	NaN
865	866	1	2	female	42	0	0	NaN
866	867	1	2	female	27	1	0	NaN
867	868	0	1	male	31	0	0	A24
868	869	0	3	male	NaN	0	0	NaN
869	870	1	3	male	4	1	1	NaN
870	871	0	3	male	26	0	0	NaN

```
871          872         1       1  female   47      1      1          D35
872          873         0       1    male   33      0      0  B51 B53 B55
873          874         0       3    male   47      0      0          NaN
874          875         1       2  female   28      1      0          NaN
875          876         1       3  female   15      0      0          NaN
876          877         0       3    male   20      0      0          NaN
877          878         0       3    male   19      0      0          NaN
878          879         0       3    male  NaN      0      0          NaN
879          880         1       1  female   56      0      1          C50
880          881         1       2  female   25      0      1          NaN
881          882         0       3    male   33      0      0          NaN
882          883         0       3  female   22      0      0          NaN
883          884         0       2    male   28      0      0          NaN
884          885         0       3    male   25      0      0          NaN
885          886         0       3  female   39      0      5          NaN
886          887         0       2    male   27      0      0          NaN
887          888         1       1  female   19      0      0          B42
888          889         0       3  female  NaN      1      2          NaN
889          890         1       1    male   26      0      0         C148
890          891         0       3    male   32      0      0          NaN

[891 rows x 8 columns]
```

1.0.4 Analysing the data

We try to analyse the data to find different patterns to address our question i.e. finding factors that made people more likely to survive.

```
In [3]: def getlen(sex, pclass):
            return len(df[(df['Sex']==sex)&(df['Pclass']==pclass)])
        def getlensurvived(sex, pclass):
            return len(df[(df['Sex']==sex)&(df['Pclass']==pclass)&(df['Survived']==1)])

        maleupperclass = getlen("male", 1)
        malemiddleclass = getlen("male", 2)
        malelowerclass = getlen("male", 3)
        femaleupperclass = getlen("female", 1)
        femalemiddleclass = getlen("female", 2)
        femalelowerclass = getlen("female", 3)
        totalupperclass = maleupperclass + femaleupperclass
        totalmiddleclass = malemiddleclass + femalemiddleclass
        totallowerclass = malelowerclass + femalelowerclass
        totalmale = maleupperclass + malemiddleclass + malelowerclass
        totalfemale = femaleupperclass + femalemiddleclass + femalelowerclass
        totaltotal = totalmale + totalfemale

        malesurviveduppperclass = getlensurvived("male", 1)
        malesurvivedmiddleclass = getlensurvived("male", 2)
        malesurvivedlowerclass = getlensurvived("male", 3)
        femalesurviveduppperclass = getlensurvived("female", 1)
        femalesurvivedmiddleclass = getlensurvived("female", 2)
        femalesurvivedlowerclass = getlensurvived("female", 3)
        totalsurviveduppperclass = malesurviveduppperclass + femalesurviveduppperclass
        totalsurvivedmiddleclass = malesurvivedmiddleclass + femalesurvivedmiddleclass
        totalsurvivedlowerclass = malesurvivedlowerclass + femalesurvivedlowerclass
```

```
totalsurvivedmale = malesurviveduppperclass + malesurvivedmiddleclass + malesurvivedlowerclass
totalsurvivedfemale = femalesurviveduppperclass + femalesurvivedmiddleclass + femalesurvivedlowe
totalsurvivedtotal = totalsurvivedmale + totalsurvivedfemale

def TableSeries(upperclass, middleclass, lowerclass, total):
    return pandas.Series([upperclass, middleclass, lowerclass, total],
                                index=['Upper Class', 'Middle Class', 'Lower Class', 'Total'])

total = {'Male': TableSeries(maleupperclass, malemiddleclass, malelowerclass, totalmale),
         'Female': TableSeries(femaleupperclass, femalemiddleclass, femalelowerclass, totalfema
         'Total': TableSeries(totaluppperclass, totalmiddleclass, totallowerclass, totaltotal)
        }
dftotal = pandas.DataFrame(total)
totalsurvived = {'Male': TableSeries(malesurviveduppperclass, malesurvivedmiddleclass, malesurvi
                 'Female': TableSeries(femalesurviveduppperclass, femalesurvivedmiddleclass,
                                       femalesurvivedmiddleclass, totalsurvivedfemale),
                 'Total': TableSeries(totalsurviveduppperclass, totalsurvivedmiddleclass,
                                      totalsurvivedlowerclass, totalsurvivedtotal)
        }
dftotalsurvived = pandas.DataFrame(totalsurvived)
```

Total People in Dataset

```
In [4]: dftotal
```

Out[4]:

	Female	Male	Total
Upper Class	94	122	216
Middle Class	76	108	184
Lower Class	144	347	491
Total	314	577	891

Total People that survived in Dataset

```
In [5]: dftotalsurvived
```

Out[5]:

	Female	Male	Total
Upper Class	91	45	136
Middle Class	70	17	87
Lower Class	70	47	119
Total	233	109	342

From the above tables it could be inferred that the probability of female to survive was more than the probability of male. Also female belonging to Upper class was highly probable to survive then the female belonging the middle class was likely to survive. Amongst females, those belonging to lower class were least likely to survive. Also the likelihood of surviving of the people belonging to upper class was more than the people belonging to other two classes.

There are people whose age is not available to us. So we can't really analyse them by considering age as a factor. So we calculate the mean and median of the age of the people whose age is known to us. We also calculate the mean and median of age of people who survived. We also find the same statistics of the people travelling in different class and people with different sex.

```
In [6]: print('Number of people whose Age data is available - ' + str(len(df[pandas.notnull(df['Age'])]
        print('Number of people whose Age data is missing - ' + str(len(df[pandas.isnull(df['Age'])])))

        def getAge(sex):
```

```
    return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))]['Age']
def getSurvivedAge(sex):
    return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))&(df['Survived']==1)]['Age']
def getClassAge(sex, pclass):
    return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))&(df['Pclass']==pclass)]['Age']
def getSurvivedClassAge(sex, pclass):
    return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))&(df['Pclass']==pclass)&(df['Survived
def getTotalClassAge(pclass):
    return df[(pandas.notnull(df['Age']))&(df['Pclass']==pclass)]['Age']
def getTotalSurvivedClassAge(pclass):
    return df[(pandas.notnull(df['Age']))&(df['Pclass']==pclass)&(df['Survived']==1)]['Age']

meantotal = df[pandas.notnull(df['Age'])]['Age'].mean()
mediantotal = df[pandas.notnull(df['Age'])]['Age'].median()
meantotalsurvived = df[(pandas.notnull(df['Age']))&(df['Survived']==1)]['Age'].mean()
mediantotalsurvived = df[(pandas.notnull(df['Age']))&(df['Survived']==1)]['Age'].median()

malemeantotal = getAge('male').mean()
malemediantotal = getAge('male').median()
malemeantotalsurvived = getSurvivedAge('male').mean()
malemediantotalsurvived = getSurvivedAge('male').median()

femalemeantotal = getAge('female').mean()
femalemediantotal = getAge('female').median()
femalemeantotalsurvived = getSurvivedAge('female').mean()
femalemediantotalsurvived = getSurvivedAge('female').median()

malemeanupper = getClassAge('male', 1).mean()
malemedianupper = getClassAge('male', 1).median()
malemeanuppersurvived = getSurvivedClassAge('male', 1).mean()
malemedianuppersurvived = getSurvivedClassAge('male', 1).median()

femalemeanupper = getClassAge('female', 1).mean()
femalemedianupper = getClassAge('female', 1).median()
femalemeanuppersurvived = getSurvivedClassAge('female', 1).mean()
femalemedianuppersurvived = getSurvivedClassAge('female', 1).median()

malemeanmiddle = getClassAge('male', 2).mean()
malemedianmiddle = getClassAge('male', 2).median()
malemeanmiddlesurvived = getSurvivedClassAge('male', 2).mean()
malemedianmiddlesurvived = getSurvivedClassAge('male', 2).median()

femalemeanmiddle = getClassAge('female', 2).mean()
femalemedianmiddle = getClassAge('female', 2).median()
femalemeanmiddlesurvived = getSurvivedClassAge('female', 2).mean()
femalemedianmiddlesurvived = getSurvivedClassAge('female', 2).median()

malemeanlower = getClassAge('male', 3).mean()
malemedianlower = getClassAge('male', 3).median()
malemeanlowersurvived = getSurvivedClassAge('male', 3).mean()
malemedianlowersurvived = getSurvivedClassAge('male', 3).median()

femalemeanlower = getClassAge('female', 3).mean()
femalemedianlower = getClassAge('female', 3).median()
```

```
femalemeanlowersurvived = getClassAge('female', 3).mean()
femalemedianlowersurvived = getClassAge('female', 3).median()

totalmeanupper = getTotalClassAge(1).mean()
totalmedianupper = getTotalClassAge(1).median()
totalmeanuppersurvived = getTotalSurvivedClassAge(1).mean()
totalmedianuppersurvived = getTotalSurvivedClassAge(1).median()

totalmeanmiddle = getTotalClassAge(2).mean()
totalmedianmiddle = getTotalClassAge(2).median()
totalmeanmiddlesurvived = getTotalSurvivedClassAge(2).mean()
totalmedianmiddlesurvived = getTotalSurvivedClassAge(2).median()

totalmeanlower = getTotalClassAge(3).mean()
totalmedianlower = getTotalClassAge(3).median()
totalmeanlowersurvived = getTotalSurvivedClassAge(3).mean()
totalmedianlowersurvived = getTotalSurvivedClassAge(3).median()

totalmean = {'Male': TableSeries(malemeanupper, malemeanmiddle, malemeanlower, malemeantotal),
             'Female': TableSeries(femalemeanupper, femalemeanmiddle, femalemeanlower, femaleme
             'Total': TableSeries(totalmeanupper, totalmeanmiddle, totalmeanlower, meantotal)
        }
dftotalmean = pandas.DataFrame(totalmean)
totalmeansurvived = {'Male': TableSeries(malemeanuppersurvived, malemeanmiddlesurvived,
                                         malemeanlowersurvived, malemeantotalsurvived),
                     'Female': TableSeries(femalemeanuppersurvived, femalemeanmiddlesurvived,
                                           femalemeanlowersurvived, femalemeantotalsurvived),
                     'Total': TableSeries(totalmeanuppersurvived, totalmeanmiddlesurvived,
                                          totalmeanlowersurvived, meantotalsurvived)
        }
dftotalmeansurvived = pandas.DataFrame(totalmeansurvived)

totalmedian = {'Male': TableSeries(malemedianupper, malemedianmiddle, malemedianlower, malemedi
               'Female': TableSeries(femalemedianupper, femalemedianmiddle, femalemedianlower,
               'Total': TableSeries(totalmedianupper, totalmedianmiddle, totalmedianlower, medi
        }
dftotalmedian = pandas.DataFrame(totalmedian)
totalmediansurvived = {'Male': TableSeries(malemedianuppersurvived, malemedianmiddlesurvived,
                                           malemedianlowersurvived, malemediantotalsurvived),
                       'Female': TableSeries(femalemedianuppersurvived, femalemedianmiddlesurvi
                                             femalemedianlowersurvived, femalemediantotalsurvive
                       'Total': TableSeries(totalmedianuppersurvived, totalmedianmiddlesurvived
                                            totalmedianlowersurvived, mediantotalsurvived),
        }
dftotalmediansurvived = pandas.DataFrame(totalmediansurvived)
```

```
Number of people whose Age data is available - 714
Number of people whose Age data is missing - 177
```

Mean Table for all people

```
In [7]: dftotalmean
```

```
Out[7]:                  Female       Male      Total
        Upper Class   34.611765  41.281386  38.233441
```

```
Middle Class  28.722973  30.740707  29.877630
Lower Class   21.750000  26.507589  25.140620
Total         27.915709  30.726645  29.699118
```

Mean Table for people survived

```
In [8]: dftotalmeansurvived
```

Out[8]:

	Female	Male	Total
Upper Class	34.939024	36.248000	35.368197
Middle Class	28.080882	16.022000	25.901566
Lower Class	21.750000	22.274211	20.646118
Total	28.847716	27.276022	28.343690

Median Table for all people

```
In [9]: dftotalmedian
```

Out[9]:

	Female	Male	Total
Upper Class	35.0	40	37
Middle Class	28.0	30	29
Lower Class	21.5	25	24
Total	27.0	29	28

Median Table for people survived

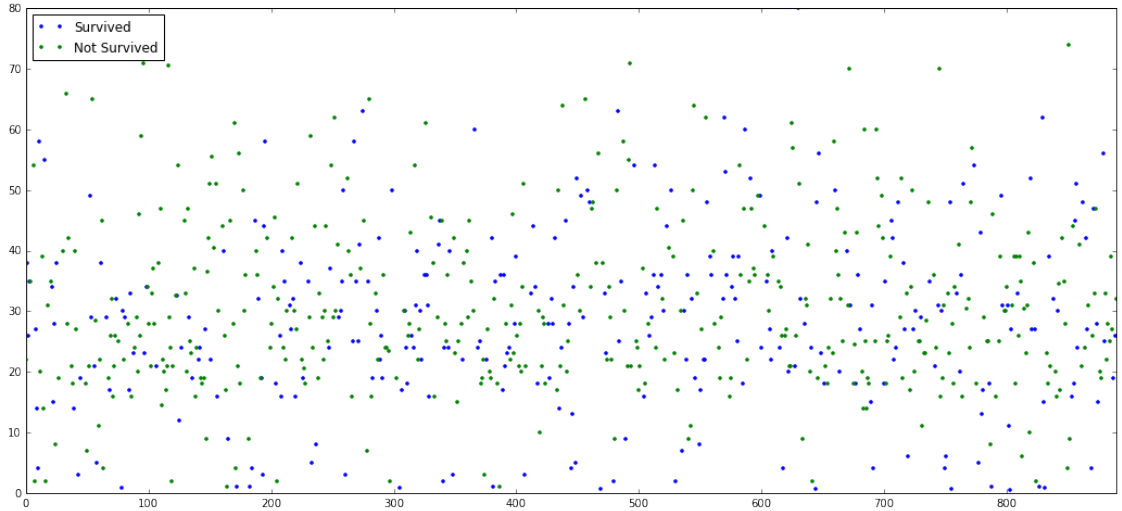
```
In [10]: dftotalmediansurvived
```

Out[10]:

	Female	Male	Total
Upper Class	35.0	36	35
Middle Class	28.0	3	28
Lower Class	21.5	25	22
Total	28.0	28	28

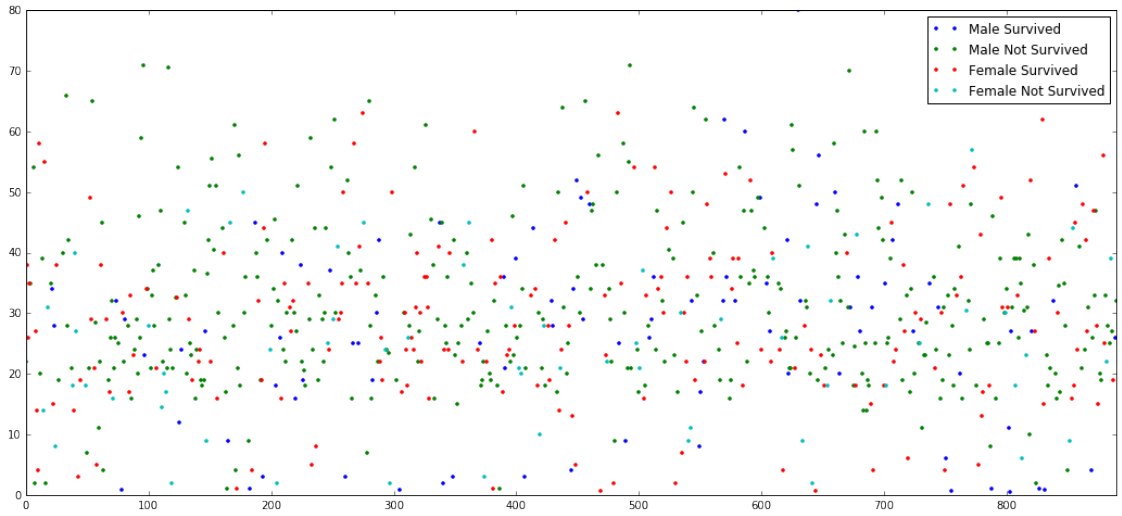
Let us visualize the data according to age to get a better insight

```
In [11]: %matplotlib inline
         def getAllPeopleAge(survived):
             return df[(pandas.notnull(df['Age']))&(df['Survived']==survived)]['Age']
         df2 = pandas.concat([getAllPeopleAge(1), getAllPeopleAge(0)],
                             axis=1, keys=['Survived','Not Survived'])
         df2.plot(style=".", figsize=[18, 8]);
```



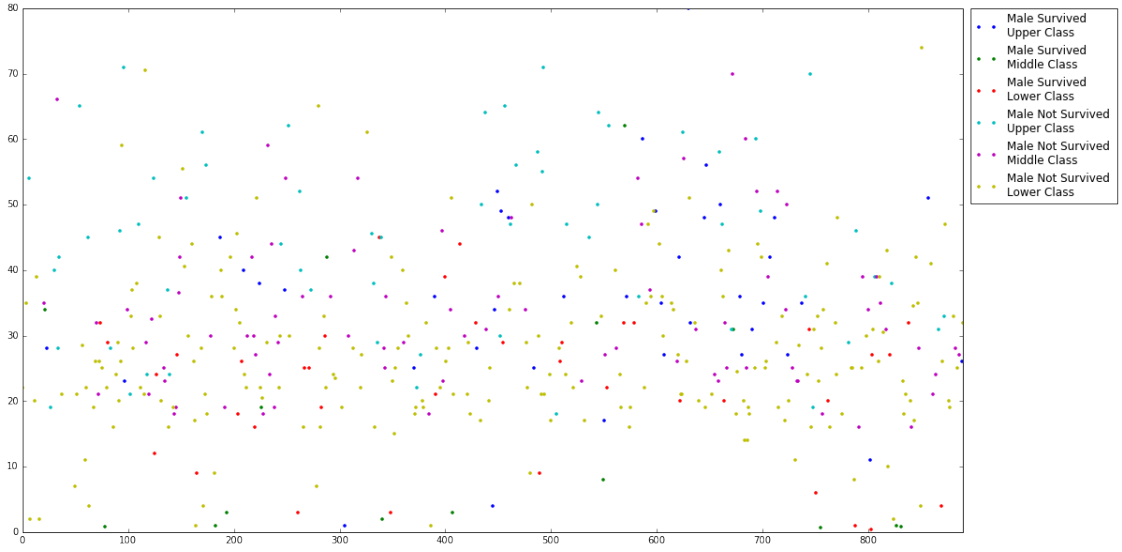
The above graph gives no pattern about the people who survived or didn't survived. So we try to explore further by creating a plot consisting of males who survived and females who survived

```
In [12]: def getPeopleAge(sex, survived):
             return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))&(df['Survived']==survived)]['Age']
         %matplotlib inline
         df3 = pandas.concat([getPeopleAge('male', 1), getPeopleAge('male', 0),
                              getPeopleAge('female', 1), getPeopleAge('female', 0)],
                             axis=1, keys=['Male Survived','Male Not Survived', 'Female Survived','Fema
         df3.plot(style=".", figsize=[18, 8]);
```

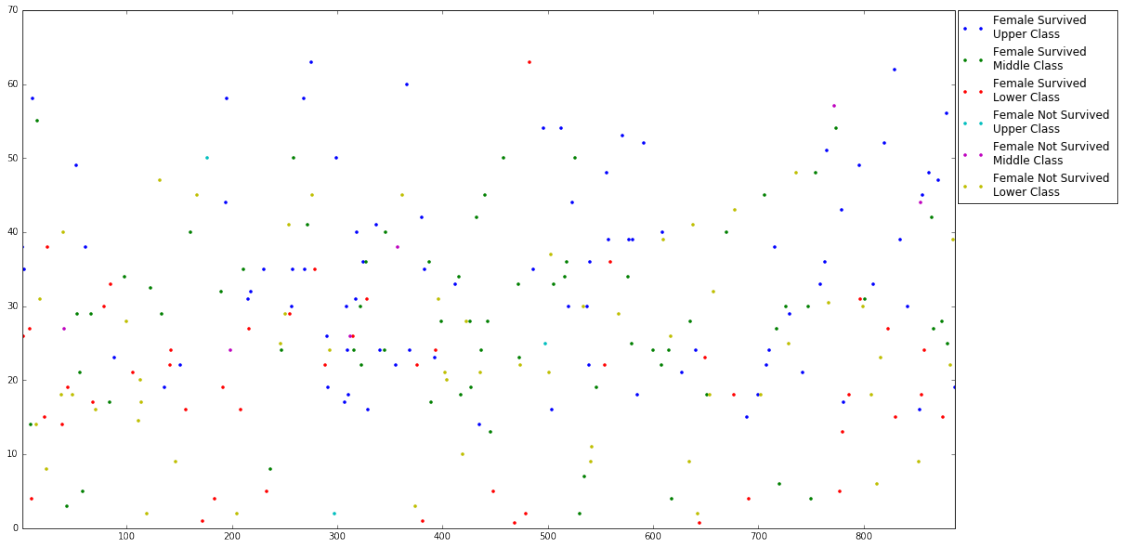


The plot gives a little idea that most of the Female survived. But still there are many female who didn't survive. The males above age of 65 didn't survive while all females above age of 65 survived. Let us explore further by dividing them into different class of people find a better pattern. First we plot just the male and then the females.

```
In [13]: def getPeopleClassAge(sex, survive, pclass):
             return df[(df['Sex']==sex)&(pandas.notnull(df['Age']))&(df['Survived']==survive)&(df['Pcla
         df4 = pandas.concat([getPeopleClassAge('male', 1, 1), getPeopleClassAge('male', 1, 2),
                              getPeopleClassAge('male', 1, 3), getPeopleClassAge('male', 0, 1),
                              getPeopleClassAge('male', 0, 2), getPeopleClassAge('male', 0, 3)],
                             axis=1, keys=['Male Survived \nUpper Class', 'Male Survived \nMiddle Class
                                           'Male Survived \nLower Class', 'Male Not Survived \nUpper Cl
                                           'Male Not Survived \nMiddle Class', 'Male Not Survived \nLow
         ax = df4.plot(style=".", figsize=[18, 10]);
         lx = ax.legend(bbox_to_anchor=(1.17, 1.01))
```



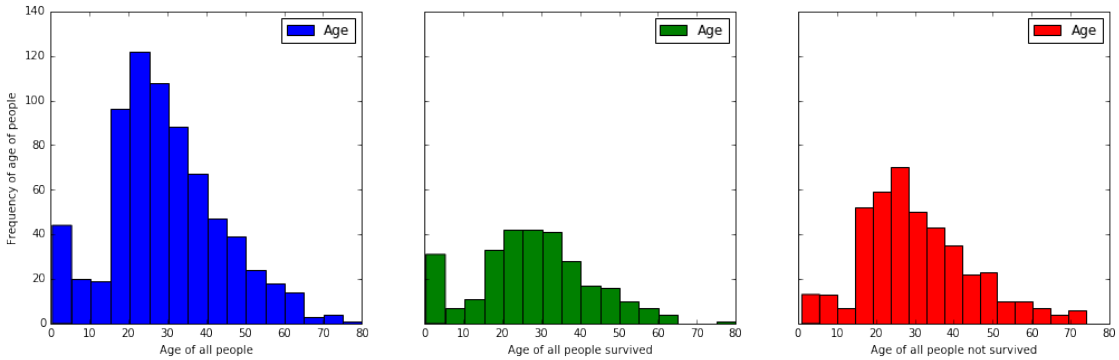

```
In [14]: df5 = pandas.concat([getPeopleClassAge('female', 1, 1), getPeopleClassAge('female', 1, 2),
                              getPeopleClassAge('female', 1, 3), getPeopleClassAge('female', 0, 1),
                              getPeopleClassAge('female', 0, 2), getPeopleClassAge('female', 0, 3)],
                             axis=1, keys=['Female Survived \nUpper Class', 'Female Survived \nMiddle Cl
                                           'Female Survived \nLower Class', 'Female Not Survived \nUppe
                                           'Female Not Survived \nMiddle Class', 'Female Not Survived \n
         ax = df5.plot(style=".", figsize=[18, 10]);
         lx = ax.legend(bbox_to_anchor=(1.18, 1.01))
```



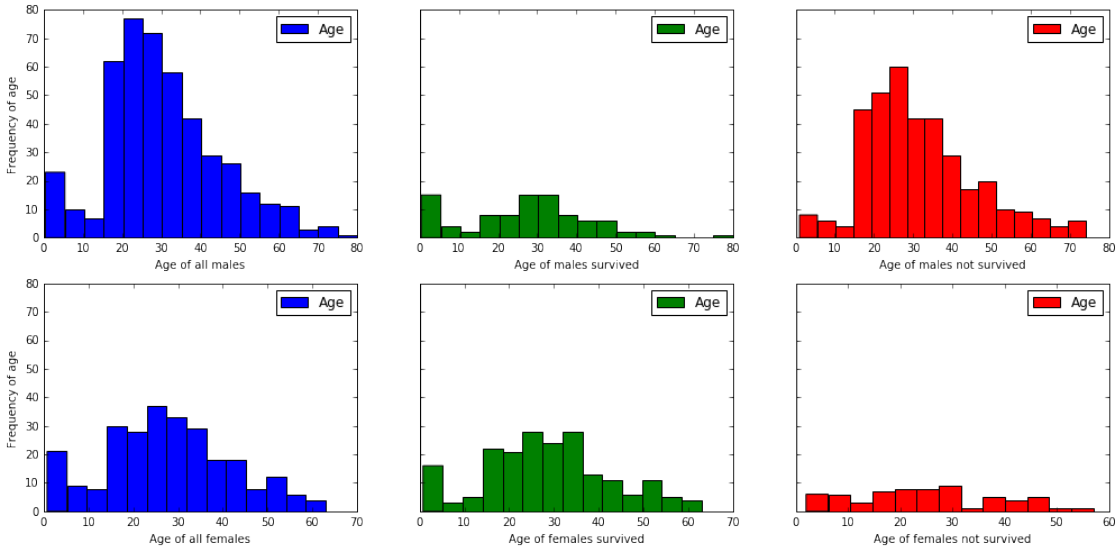
Finally before inferring any analysed data from above plots let us take a look at the age distribution of the people in data.

```
In [15]: import matplotlib.pyplot as plt
```

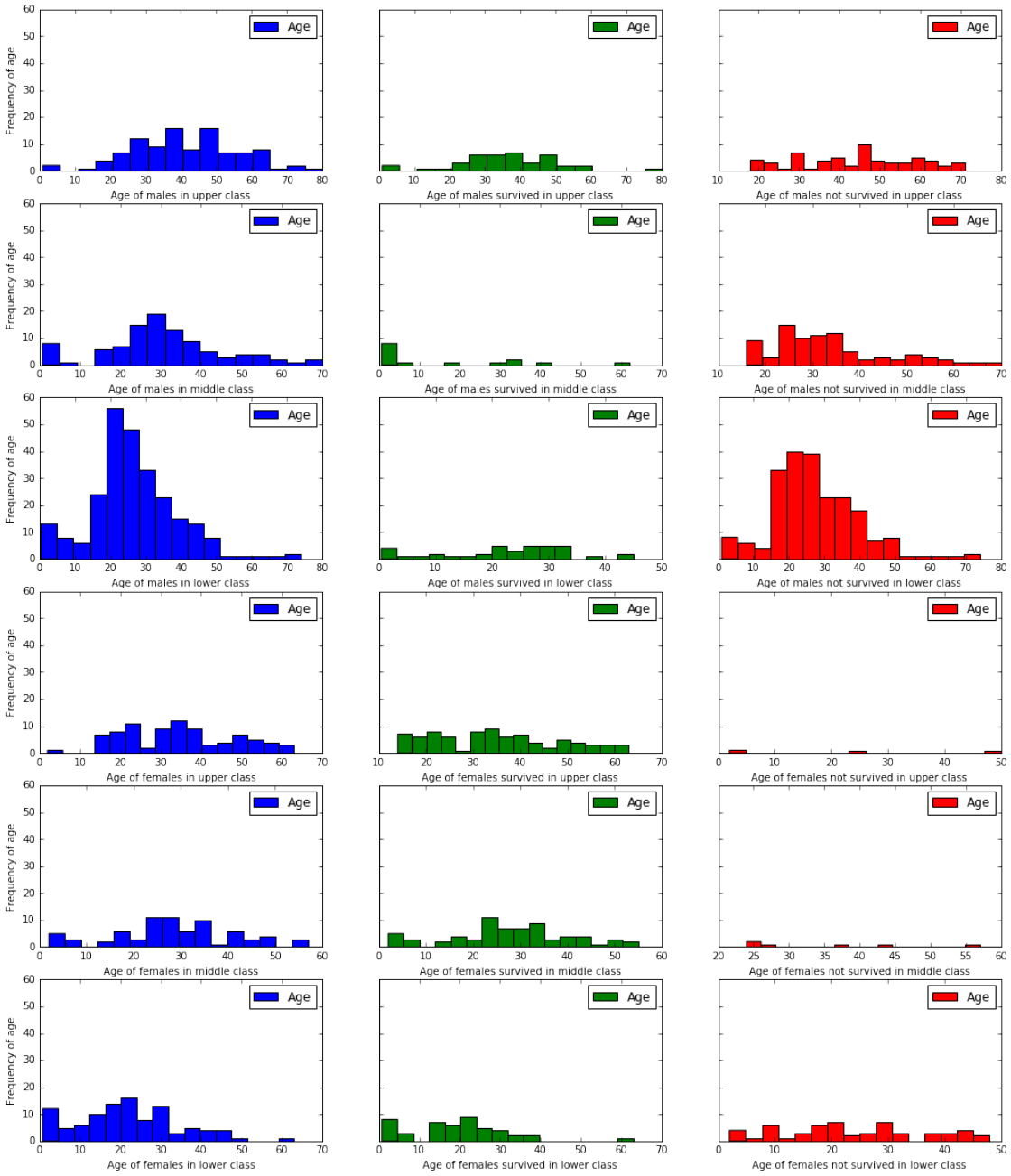
```
fig, axes = plt.subplots(nrows=1, ncols=3, sharey=True)
def subAgePlot(seriest, col, bins, size, tcolor, xaxis, yaxis):
    dft = pandas.DataFrame(seriest)
    ax = dft.plot(kind='hist', ax=axes[col], bins=bins, figsize=size, color=tcolor)
    x = ax.set_xlabel(xaxis)
    y = ax.set_ylabel(yaxis)

subAgePlot(df[(pandas.notnull(df['Age']))]['Age'], 0, 16, [17, 5], "BLUE", 'Age of all people'
subAgePlot(getAllPeopleAge(1), 1, 16, [17, 5], "GREEN", 'Age of all people survived', 'Frequen
subAgePlot(getAllPeopleAge(0), 2, 16, [17, 5], "RED", 'Age of all people not survived', 'Freque
```



In [16]:
```
fig, axes = plt.subplots(nrows=2, ncols=3, sharey=True)

subAgePlot(getAge('male'), (0,0), 16, [17, 8], "BLUE", 'Age of all males', 'Frequency of age')
subAgePlot(getPeopleAge('male', 1), (0,1), 16, [17, 8], "GREEN", 'Age of males survived', 'Fre
subAgePlot(getPeopleAge('male', 0), (0,2), 16, [17, 8], "RED", 'Age of males not survived', 'F

subAgePlot(getAge('female'), (1,0), 14, [17, 8], "BLUE", 'Age of all females', 'Frequency of ag
subAgePlot(getPeopleAge('female', 1), (1,1), 14, [17, 8], "GREEN", 'Age of females survived',
subAgePlot(getPeopleAge('female', 0), (1,2), 13, [17, 8], "RED", 'Age of females not survived'
```



```
In [17]: fig, axes = plt.subplots(nrows=6, ncols=3, sharey=True)
         subAgePlot(getClassAge('male', 1), (0,0), 16, [17, 20], "BLUE", 'Age of males in upper class',
         subAgePlot(getPeopleClassAge('male', 1, 1), (0,1), 16, [17, 20], "GREEN",
                    'Age of males survived in upper class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('male', 0, 1), (0,2), 16, [17, 20], "RED",
                    'Age of males not survived in upper class', 'Frequency of age')

         subAgePlot(getClassAge('male', 2), (1,0), 16, [17, 20], "BLUE", 'Age of males in middle class'
         subAgePlot(getPeopleClassAge('male', 1, 2), (1,1), 16, [17, 20], "GREEN",
                    'Age of males survived in middle class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('male', 0, 2), (1,2), 16, [17, 20], "RED",
                    'Age of males not survived in middle class', 'Frequency of age')

         subAgePlot(getClassAge('male', 3), (2,0), 16, [17, 20], "BLUE", 'Age of males in lower class',
         subAgePlot(getPeopleClassAge('male', 1, 3), (2,1), 16, [17, 20], "GREEN",
                    'Age of males survived in lower class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('male', 0, 3), (2,2), 16, [17, 20], "RED",
                    'Age of males not survived in lower class', 'Frequency of age')


         subAgePlot(getClassAge('female', 1), (3,0), 16, [17, 20], "BLUE", 'Age of females in upper clas
         subAgePlot(getPeopleClassAge('female', 1, 1), (3,1), 16, [17, 20], "GREEN",
                    'Age of females survived in upper class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('female', 0, 1), (3,2), 16, [17, 20], "RED",
                    'Age of females not survived in upper class', 'Frequency of age')

         subAgePlot(getClassAge('female', 2), (4,0), 16, [17, 20], "BLUE", 'Age of females in middle cl
         subAgePlot(getPeopleClassAge('female', 1, 2), (4,1), 16, [17, 20], "GREEN",
                    'Age of females survived in middle class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('female', 0, 2), (4,2), 16, [17, 20], "RED",
                    'Age of females not survived in middle class', 'Frequency of age')

         subAgePlot(getClassAge('female', 3), (5,0), 16, [17, 20], "BLUE", 'Age of females in lower clas
         subAgePlot(getPeopleClassAge('female', 1, 3), (5,1), 16, [17, 20], "GREEN",
                    'Age of females survived in lower class', 'Frequency of age')
         subAgePlot(getPeopleClassAge('female', 0, 3), (5,2), 16, [17, 20], "RED",
                    'Age of females not survived in lower class', 'Frequency of age')
```



1.0.5 Conclusion

On analysing the above plots it could be said that the females belonging to upper Class and middle Class are very likely to survive while the females in the lower class with age below 40 were more likely to survive than any other females in the lower class.

There is no particular pattern of the survival of males in the data expect the male passengers with the age of less than 15 and belonging to upper class and middle class were more likely to survive.