All the relevant code, reporting and contentfor completing the P2 project will be done here

Data Analysis:Ask one or more questions regarding the data that will be answered during the course of the project execution

1. How does survival rate vary across socio-economic classes?
2. How did survivability vary across age groups in this sample? Can we use some graphs to visualise this? Will it be safe to say that some age-groups were more likely to survive?
3. Similar to Qn 2, can we evaluate how survivability changes across gender?

In [1]:
```
# Import all the relevant modules for running code on this project. Numpy, Pandas & matplotlib.pyplot for plotting the
 data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#Loading data from the csv file on a pandas dataframe
filename = 'C:/Users/pvatsa/Downloads/Personal/Coursework/Udacity/Data Analyst Nanodegree/P2 - Analyze a dataset/P2 pr
oject submission/titanic_data.csv'
titanicdata_df = pd.read_csv(filename)
```

In [3]:
```
#Checking data types
titanicdata_df.dtypes
```

Out[3]:
```
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
```

In [4]:
```
#defining a function to return every element in the dataframe after squaring it
def square(x):
    return x**2
```

In [5]:
```
#Getting a feel for the data
titanicdata_df.describe()
```

Out[5]:

	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

In [6]:
```
titanicdata_df[['Ticket', 'Cabin','Sex','Name', 'Embarked']].describe()
```

Out[6]:

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

Data Wrangling

In [7]:
```
#Checking for NaN in the data
titanicdata_df.loc[pd.isnull(titanicdata_df['Embarked'])]
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
61	62	1	1	Icard, Miss. Amelie	female	38	0	0	113572	80	B28	NaN
829	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62	0	0	113572	80	B28	NaN

In [8]:
```
#Checking for NaN in the dependent variable. Omit rows wherever found. This is a critical variable in our analysis
titanicdata_df.loc[pd.isnull(titanicdata_df['Survived'])]
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [9]:
```
#Checking for NaN in the data
titanicdata_df.loc[pd.isnull(titanicdata_df['Ticket'])]
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [12]:
```
#Checking for NaN in the data
print titanicdata_df.loc[pd.isnull(titanicdata_df['Cabin'])]
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
2	3	1	3	
4	5	0	3	
5	6	0	3	
7	8	0	3	
8	9	1	3	
9	10	1	2	
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	
22	23	1	3	
24	25	0	3	
25	26	1	3	
26	27	0	3	
28	29	1	3	
29	30	0	3	
30	31	0	1	
32	33	1	3	
33	34	0	2	
34	35	0	1	
35	36	0	1	
36	37	1	3	
37	38	0	3	
38	39	0	3	
..	...	...	...	
852	853	0	3	
854	855	0	2	
855	856	1	3	
856	857	1	1	
858	859	1	3	
859	860	0	3	
860	861	0	3	
861	862	0	2	
863	864	0	3	
864	865	0	2	

```
865          866         1       2
866          867         1       2
868          869         0       3
869          870         1       3
870          871         0       3
873          874         0       3
874          875         1       2
875          876         1       3
876          877         0       3
877          878         0       3
878          879         0       3
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
888          889         0       3
890          891         0       3

                                                 Name     Sex  Age  SibSp  \
0                             Braund, Mr. Owen Harris    male   22      1
2                              Heikkinen, Miss. Laina  female   26      0
4                            Allen, Mr. William Henry    male   35      0
5                                    Moran, Mr. James    male  NaN      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
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
22                           McGowan, Miss. Anna "Annie"  female   15      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
```

28	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0
29	Todoroff, Mr. Lalio	male	NaN	0
30	Uruchurtu, Don. Manuel E	male	40	0
32	Glynn, Miss. Mary Agatha	female	NaN	0
33	Wheadon, Mr. Edward H	male	66	0
34	Meyer, Mr. Edgar Joseph	male	28	1
35	Holverson, Mr. Alexander Oskar	male	42	1
36	Mamee, Mr. Hanna	male	NaN	0
37	Cann, Mr. Ernest Charles	male	21	0
38	Vander Planke, Miss. Augusta Maria	female	18	2
..	...	...	...	...
852	Boulos, Miss. Nourelain	female	9	1
854	Carter, Mrs. Ernest Courtenay (Lilian Hughes)	female	44	1
855	Aks, Mrs. Sam (Leah Rosen)	female	18	0
856	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45	1
858	Baclini, Mrs. Solomon (Latifa Qurban)	female	24	0
859	Razi, Mr. Raihed	male	NaN	0
860	Hansen, Mr. Claus Peter	male	41	2
861	Giles, Mr. Frederick Edward	male	21	1
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
868	van Melkebeke, Mr. Philemon	male	NaN	0
869	Johnson, Master. Harold Theodor	male	4	1
870	Balkic, Mr. Cerin	male	26	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
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
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1
890	Dooley, Mr. Patrick	male	32	0

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
2	0	STON/O2. 3101282	7.9250	NaN	S
4	0	373450	8.0500	NaN	S
5	0	330877	8.4583	NaN	Q
7	1	349909	21.0750	NaN	S
8	2	347742	11.1333	NaN	S
9	0	237736	30.0708	NaN	C
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
22	0	330923	8.0292	NaN	Q
24	1	349909	21.0750	NaN	S
25	5	347077	31.3875	NaN	S
26	0	2631	7.2250	NaN	C
28	0	330959	7.8792	NaN	Q
29	0	349216	7.8958	NaN	S
30	0	PC 17601	27.7208	NaN	C
32	0	335677	7.7500	NaN	Q
33	0	C.A. 24579	10.5000	NaN	S
34	0	PC 17604	82.1708	NaN	C
35	0	113789	52.0000	NaN	S
36	0	2677	7.2292	NaN	C
37	0	A./5. 2152	8.0500	NaN	S
38	0	345764	18.0000	NaN	S
..	...	...	...	...	...
852	1	2678	15.2458	NaN	C
854	0	244252	26.0000	NaN	S
855	1	392091	9.3500	NaN	S
856	1	36928	164.8667	NaN	S
858	3	2666	19.2583	NaN	C
859	0	2629	7.2292	NaN	C
860	0	350026	14.1083	NaN	S
861	0	28134	11.5000	NaN	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
868      0            345777    9.5000   NaN        S
869      1            347742   11.1333   NaN        S
870      0            349248    7.8958   NaN        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
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
888      2        W./C. 6607   23.4500   NaN        S
890      0            370376    7.7500   NaN        Q

[687 rows x 12 columns]
```

In [14]:
```
titanicdata_df.loc[pd.isnull(titanicdata_df['Fare'])]
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [15]:
```
print titanicdata_df.loc[titanicdata_df['Fare'].isin([0, 0.00,0.0000,0.0])]
```

```
     PassengerId  Survived  Pclass                              Name   Sex  \
179          180         0       3               Leonard, Mr. Lionel  male
263          264         0       1             Harrison, Mr. William  male
271          272         1       3      Tornquist, Mr. William Henry  male
277          278         0       2       Parkes, Mr. Francis "Frank"  male
302          303         0       3   Johnson, Mr. William Cahoone Jr  male
413          414         0       2    Cunningham, Mr. Alfred Fleming  male
466          467         0       2             Campbell, Mr. William  male
481          482         0       2  Frost, Mr. Anthony Wood "Archie"  male
597          598         0       3                Johnson, Mr. Alfred  male
633          634         0       1     Parr, Mr. William Henry Marsh  male
674          675         0       2        Watson, Mr. Ennis Hastings  male
732          733         0       2              Knight, Mr. Robert J  male
806          807         0       1            Andrews, Mr. Thomas Jr  male
815          816         0       1                 Fry, Mr. Richard  male
822          823         0       1   Reuchlin, Jonkheer. John George  male

     Age  SibSp  Parch  Ticket  Fare Cabin Embarked
179   36      0      0    LINE     0   NaN        S
263   40      0      0  112059     0   B94        S
271   25      0      0    LINE     0   NaN        S
277  NaN      0      0  239853     0   NaN        S
302   19      0      0    LINE     0   NaN        S
413  NaN      0      0  239853     0   NaN        S
466  NaN      0      0  239853     0   NaN        S
481  NaN      0      0  239854     0   NaN        S
597   49      0      0    LINE     0   NaN        S
633  NaN      0      0  112052     0   NaN        S
674  NaN      0      0  239856     0   NaN        S
732  NaN      0      0  239855     0   NaN        S
806   39      0      0  112050     0   A36        S
815  NaN      0      0  112058     0  B102        S
822   38      0      0   19972     0   NaN        S
```

In [16]:
```
titanicdata_df.loc[~titanicdata_df['Sex'].isin(['male','female'])]
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [17]:
```
titanicdata_df.loc[titanicdata_df['Age'].isin([None])]
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [18]:
```
titanicdata_df.loc[titanicdata_df['Age'].isin([0])]
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [19]:
```
titanicdata_df.loc[pd.isnull(titanicdata_df['Age'])]
```

Out[19]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	C
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Q
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7.8958	NaN	S
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	female	NaN	1	0	PC 17569	146.5208	B78	C
32	33	1	3	Glynn, Miss. Mary Agatha	female	NaN	0	0	335677	7.7500	NaN	Q
36	37	1	3	Mamee, Mr. Hanna	male	NaN	0	0	2677	7.2292	NaN	C
42	43	0	3	Kraeff, Mr. Theodor	male	NaN	0	0	349253	7.8958	NaN	C
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	8.0500	NaN	S
46	47	0	3	Lennon, Mr. Denis	male	NaN	1	0	370371	15.5000	NaN	Q
47	48	1	3	O'Driscoll, Miss. Bridget	female	NaN	0	0	14311	7.7500	NaN	Q
48	49	0	3	Samaan, Mr. Youssef	male	NaN	2	0	2662	21.6792	NaN	C
55	56	1	1	Woolner, Mr. Hugh	male	NaN	0	0	19947	35.5000	C52	S
64	65	0	1	Stewart, Mr. Albert A	male	NaN	0	0	PC 17605	27.7208	NaN	C
65	66	1	3	Moubarek, Master. Gerios	male	NaN	1	1	2661	15.2458	NaN	C
76	77	0	3	Staneff, Mr. Ivan	male	NaN	0	0	349208	7.8958	NaN	S
77	78	0	3	Moutal, Mr. Rahamin Haim	male	NaN	0	0	374746	8.0500	NaN	S
82	83	1	3	McDermott, Miss. Brigdet Delia	female	NaN	0	0	330932	7.7875	NaN	Q

87	88	0	3	Slocovski, Mr. Selman Francis	male	NaN	0	0	SOTON/OQ 392086	8.0500	NaN	S
95	96	0	3	Shorney, Mr. Charles Joseph	male	NaN	0	0	374910	8.0500	NaN	S
101	102	0	3	Petroff, Mr. Pastcho ("Pentcho")	male	NaN	0	0	349215	7.8958	NaN	S
107	108	1	3	Moss, Mr. Albert Johan	male	NaN	0	0	312991	7.7750	NaN	S
109	110	1	3	Moran, Miss. Bertha	female	NaN	1	0	371110	24.1500	NaN	Q
121	122	0	3	Moore, Mr. Leonard Charles	male	NaN	0	0	A4. 54510	8.0500	NaN	S
126	127	0	3	McMahon, Mr. Martin	male	NaN	0	0	370372	7.7500	NaN	Q
128	129	1	3	Peter, Miss. Anna	female	NaN	1	1	2668	22.3583	F E69	C
140	141	0	3	Boulos, Mrs. Joseph (Sultana)	female	NaN	0	2	2678	15.2458	NaN	C
154	155	0	3	Olsen, Mr. Ole Martin	male	NaN	0	0	Fa 265302	7.3125	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
718	719	0	3	McEvoy, Mr. Michael	male	NaN	0	0	36568	15.5000	NaN	Q
727	728	1	3	Mannion, Miss. Margareth	female	NaN	0	0	36866	7.7375	NaN	Q
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0000	NaN	S
738	739	0	3	Ivanoff, Mr. Kanio	male	NaN	0	0	349201	7.8958	NaN	S
739	740	0	3	Nankoff, Mr. Minko	male	NaN	0	0	349218	7.8958	NaN	S
740	741	1	1	Hawksford, Mr. Walter James	male	NaN	0	0	16988	30.0000	D45	S
760	761	0	3	Garfirth, Mr. John	male	NaN	0	0	358585	14.5000	NaN	S
766	767	0	1	Brewe, Dr. Arthur Jackson	male	NaN	0	0	112379	39.6000	NaN	C
768	769	0	3	Moran, Mr. Daniel J	male	NaN	1	0	371110	24.1500	NaN	Q

773	774	0	3	Elias, Mr. Dibo	male	NaN	0	0	2674	7.2250	NaN	C
776	777	0	3	Tobin, Mr. Roger	male	NaN	0	0	383121	7.7500	F38	Q
778	779	0	3	Kilgannon, Mr. Thomas J	male	NaN	0	0	36865	7.7375	NaN	Q
783	784	0	3	Johnston, Mr. Andrew G	male	NaN	1	2	W./C. 6607	23.4500	NaN	S
790	791	0	3	Keane, Mr. Andrew "Andy"	male	NaN	0	0	12460	7.7500	NaN	Q
792	793	0	3	Sage, Miss. Stella Anna	female	NaN	8	2	CA. 2343	69.5500	NaN	S
793	794	0	1	Hoyt, Mr. William Fisher	male	NaN	0	0	PC 17600	30.6958	NaN	C
815	816	0	1	Fry, Mr. Richard	male	NaN	0	0	112058	0.0000	B102	S
825	826	0	3	Flynn, Mr. John	male	NaN	0	0	368323	6.9500	NaN	Q
826	827	0	3	Lam, Mr. Len	male	NaN	0	0	1601	56.4958	NaN	S
828	829	1	3	McCormack, Mr. Thomas Joseph	male	NaN	0	0	367228	7.7500	NaN	Q
832	833	0	3	Saad, Mr. Amin	male	NaN	0	0	2671	7.2292	NaN	C
837	838	0	3	Sirota, Mr. Maurice	male	NaN	0	0	392092	8.0500	NaN	S
839	840	1	1	Marechal, Mr. Pierre	male	NaN	0	0	11774	29.7000	C47	C
846	847	0	3	Sage, Mr. Douglas Bullen	male	NaN	8	2	CA. 2343	69.5500	NaN	S
849	850	1	1	Goldenberg, Mrs. Samuel L (Edwiga Grabowska)	female	NaN	1	0	17453	89.1042	C92	C
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	C
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

177 rows × 12 columns

In [20]:
```
titanicdata_df.loc[~titanicdata_df['Pclass'].isin([1,2,3])]
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [21]:
```
titanicdata_df.loc[~titanicdata_df['Survived'].isin([0,1])]
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

Analyze the missing data information that we've gathered above for each of the columns in the dataframe.

1. 'Survived' and 'Pclass' have no missing information and have all the values accurately filled as per the permitted values for these categorical variables.
2. 'Age' has 177 rows missing with values = NaN. These will have to be omitted at a later stage when we're doing age related computations and comparisons.
3. I see some noticeable information missing for columns 'Embarked' and 'Cabin' but I don't intend to answer any question around these variables hence this can be ignored.
4. For column 'Fare', I notice something very strange. These values are 0 for a substantial number of rows in the data. This will be fixed in the next section of the workbook.
5. For column 'Sex', all the rows are correctly filled with either 'Male' or 'Female', the only two values for this categorical variable as far as this study is concerned.
6. No missing data tests have been done for the columns: 'Name', 'SibSp' & 'Parch'

In [22]:
```
#Handle missing data. Define a function to handle the missing data in the fare column as per point #4 above
def replace_fare(grp):
    grp['FareNew'] = grp['Fare'].median()
    return grp
```

In [23]:
```
#Create a new column in the dataset consisting of the median fare of each group
titanicdata_df = titanicdata_df.groupby('Pclass').apply(replace_fare)
```

In [24]:
```
#Replace fare median values for the actual fare values that we have where original value not equal to zero
titanicdata_df['FareNew'] = np.where((titanicdata_df['Fare'] !=0), titanicdata_df['Fare'],titanicdata_df['FareNew'])
```

In [25]:
```
#View of a dataframe after dropping the 'Fare' column. Inplace argument is set to false by default. Hence no change ha
s been made
#to the original dataframe. THis is just a view of the dataframe. I'm retaining the fare column just in case.
titanicdata_df.drop('Fare', axis = 'columns')
```

Out[25]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked	FareNew
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	NaN	S	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	C85	C	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	NaN	S	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	C123	S	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	NaN	S	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	NaN	Q	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	E46	S	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	NaN	S	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	NaN	S	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	NaN	C	30.0708
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	G6	S	16.7000
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	C103	S	26.5500
12	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	NaN	S	8.0500
13	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	NaN	S	31.2750
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	NaN	S	7.8542
				Hewlett, Mrs. (Mary D								

15	16	1	2	Kingcome)	female	55	0	0	248706	NaN	S	16.0000
16	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	NaN	Q	29.1250
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	NaN	S	13.0000
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31	1	0	345763	NaN	S	18.0000
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	NaN	C	7.2250
20	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	NaN	S	26.0000
21	22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	D56	S	13.0000
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	NaN	Q	8.0292
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	A6	S	35.5000
24	25	0	3	Palsson, Miss. Torborg Danira	female	8	3	1	349909	NaN	S	21.0750
25	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38	1	5	347077	NaN	S	31.3875
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	NaN	C	7.2250
27	28	0	1	Fortune, Mr. Charles Alexander	male	19	3	2	19950	C23 C25 C27	S	263.0000
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	NaN	Q	7.8792
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	NaN	S	7.8958
...	...	...	...	...	...	...	...	...	...	...	...	...
861	862	0	2	Giles, Mr. Frederick Edward	male	21	1	0	28134	NaN	S	11.5000

862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48	0	0	17466	D17	S	25.9292
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	NaN	S	69.5500
864	865	0	2	Gill, Mr. John William	male	24	0	0	233866	NaN	S	13.0000
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42	0	0	236852	NaN	S	13.0000
866	867	1	2	Duran y More, Miss. Asuncion	female	27	1	0	SC/PARIS 2149	NaN	C	13.8583
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31	0	0	PC 17590	A24	S	50.4958
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	NaN	S	9.5000
869	870	1	3	Johnson, Master. Harold Theodor	male	4	1	1	347742	NaN	S	11.1333
870	871	0	3	Balkic, Mr. Cerin	male	26	0	0	349248	NaN	S	7.8958
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47	1	1	11751	D35	S	52.5542
872	873	0	1	Carlsson, Mr. Frans Olof	male	33	0	0	695	B51 B53 B55	S	5.0000
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47	0	0	345765	NaN	S	9.0000
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28	1	0	P/PP 3381	NaN	C	24.0000
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15	0	0	2667	NaN	C	7.2250
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20	0	0	7534	NaN	S	9.8458
877	878	0	3	Petroff, Mr. Nedelio	male	19	0	0	349212	NaN	S	7.8958

878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	NaN	S	7.8958
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56	0	1	11767	C50	C	83.1583
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25	0	1	230433	NaN	S	26.0000
881	882	0	3	Markun, Mr. Johann	male	33	0	0	349257	NaN	S	7.8958
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22	0	0	7552	NaN	S	10.5167
883	884	0	2	Banfield, Mr. Frederick James	male	28	0	0	C.A./SOTON 34068	NaN	S	10.5000
884	885	0	3	Sutehall, Mr. Henry Jr	male	25	0	0	SOTON/OQ 392076	NaN	S	7.0500
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39	0	5	382652	NaN	Q	29.1250
886	887	0	2	Montvila, Rev. Juozas	male	27	0	0	211536	NaN	S	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19	0	0	112053	B42	S	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	NaN	S	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	C148	C	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32	0	0	370376	NaN	Q	7.7500

891 rows × 12 columns

Data is cleaned now. We will now run some statistical tests to check for correlation, visualization and statistical significance wherever applicable. For my analysis, I've taken survived as the dependent variable and Pclass, age, sex as the independent variables.

Exploration Phase

Question 1.

In [26]:
```
titanicdata_df.corr()
```

Out[26]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658	0.018753
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	0.250635
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	-0.561243
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.099377
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	0.155423
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	0.212103
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	0.995568
FareNew	0.018753	0.250635	-0.561243	0.099377	0.155423	0.212103	0.995568	1.000000

Pearson correlation coefficient,r, measures the linear relationship between two variables. Hence it is only relevant for a bivariate data set. The value of coefficient varies between -1 and 1.

For our first question, we see a r of -0.338481 which indicates that there is negative correlation to an extent. As Pclass value increases, survived value decreases. This indicates that socio-economic-status has some correlation with survival although it does not appear to be very pronounced. 33% of variation in survival can be explained by Pclass, whereas rest depends on other factors. On a side note, there is also a negative r of -0.54 between Fare and Pclass. This is somewhat obvious as the more expensive tickets are purchased by individuals who're better off socio-economically. The limitation of the correlation coefficient is that it doesn't tell us how strong and significant is the relationship between independent variables

In [27]:
```
#Let's look at some visualizations for this relationship.
titanicdata_df.groupby('Pclass').mean()
```

Out[27]:

	PassengerId	Survived	Age	SibSp	Parch	Fare	FareNew
Pclass							
1	461.597222	0.629630	38.233441	0.416667	0.356481	84.154687	85.550231
2	445.956522	0.472826	29.877630	0.402174	0.380435	20.662183	21.126857
3	439.154786	0.242363	25.140620	0.615071	0.393075	13.675550	13.741131

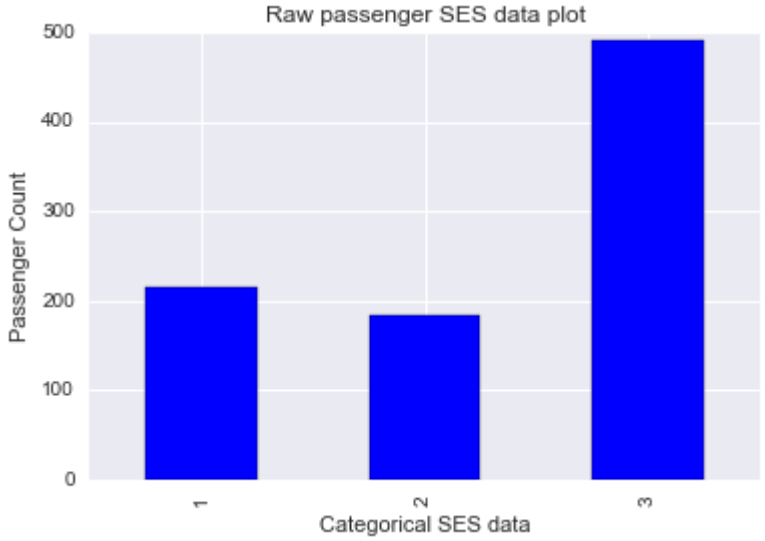
In [28]:
```
titanicdata_df.groupby('Pclass').count()
```

Out[28]:

	PassengerId	Survived	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FareNew
Pclass												
1	216	216	216	216	186	216	216	216	216	176	214	216
2	184	184	184	184	173	184	184	184	184	16	184	184
3	491	491	491	491	355	491	491	491	491	12	491	491

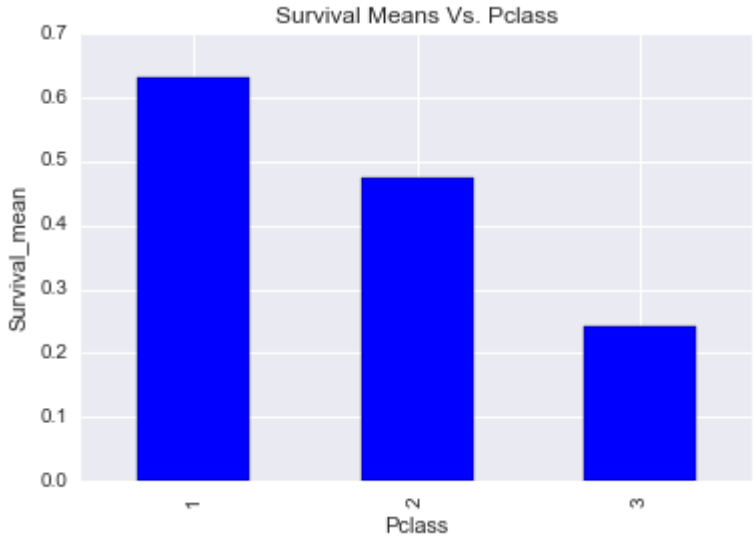
In [29]:
```
#1D analysis
%matplotlib inline
total_passengers_by_class = titanicdata_df.groupby('Pclass').count()['PassengerId']
total_passengers_by_class.plot.bar()
plt.ylabel('Passenger Count')
plt.xlabel('Categorical SES data')
plt.title('Raw passenger SES data plot')
```

Out[29]: <matplotlib.text.Text at 0x1b16f9e8>



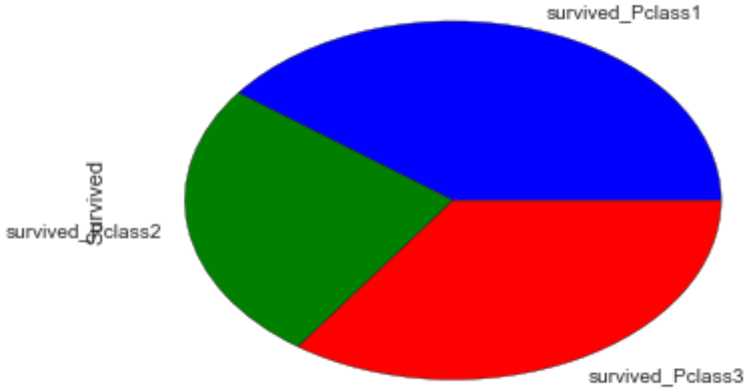
In [30]:
```
#2D analysis
%matplotlib inline
survival_mean_by_class = titanicdata_df.groupby('Pclass').mean()['Survived']
survival_summary_by_class = titanicdata_df.groupby('Pclass').sum()['Survived']
survival_mean_by_class.plot.bar()
plt.ylabel('Survival_mean')
plt.xlabel('Pclass')
plt.title('Survival Means Vs. Pclass')
```

Out[30]: <matplotlib.text.Text at 0x1b39fe80>



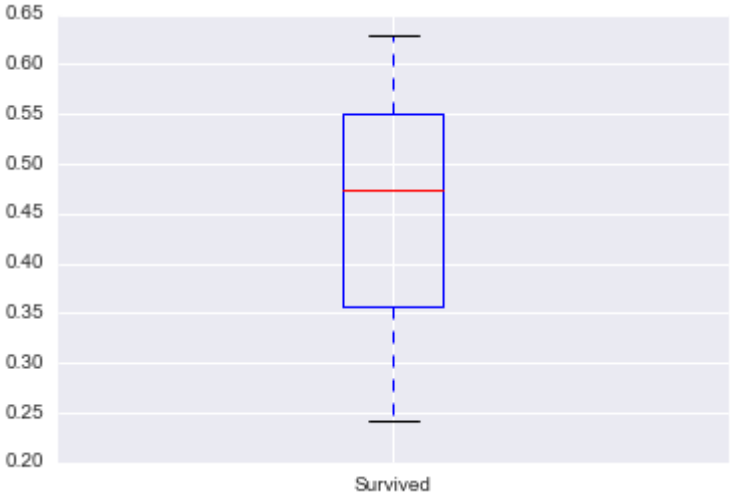
In [31]:
```
survived_Pclass1 = titanicdata_df.groupby('Pclass').get_group(1).sum()['Survived']
survived_Pclass2 = titanicdata_df.groupby('Pclass').get_group(2).sum()['Survived']
survived_Pclass3 = titanicdata_df.groupby('Pclass').get_group(3).sum()['Survived']
survival_summary_by_class.plot.pie(labels = ['survived_Pclass1', 'survived_Pclass2', 'survived_Pclass3'])
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1b0c96d8>



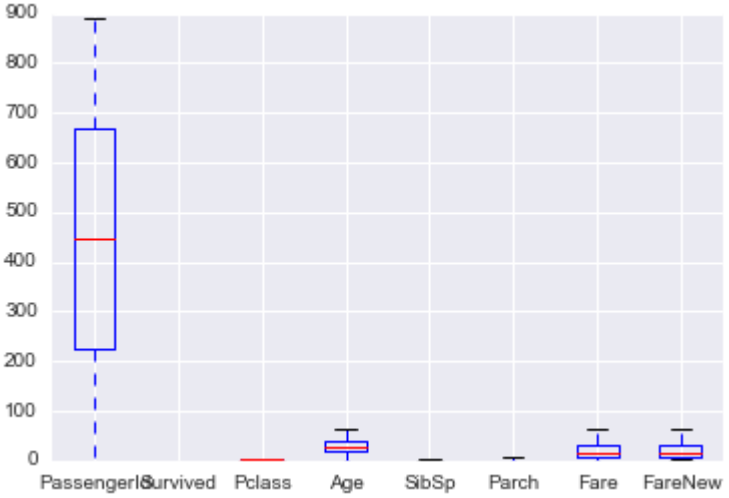
In [32]:
```
%matplotlib inline
survival_mean_by_class.plot.box()
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x1b62d748>



In [33]:
```
%matplotlib inline
titanicdata_df.plot.box()
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x209083c8>



A statistical test is required to prove any association between gender and survivability. Since these both are categorical variables, I'd choose a chi-square test for the same. In this test, there are four steps to come to a conclusion: Step I: Build a contingency table Step II: State the hypotheses for the tests Step III: Formulate an analysis plan Step IV: Analyze data Step V: Interpret results

In [34]:
```
#Step I: A contingency table
ct_Pclass_survivability = pd.pivot_table(data = titanicdata_df[['Survived', 'Pclass']], index = 'Survived', columns = 'Pclass',
                                         aggfunc = len, margins = True)
ct_Pclass_survivability
```

Out[34]:

Pclass	1	2	3	All
Survived				
0	80	97	372	549
1	136	87	119	342
All	216	184	491	891

In [35]:
```
ct_Pclass_survivability_no_margins = pd.pivot_table(data = titanicdata_df[['Survived', 'Pclass']], index = 'Survived',
                                                    columns = 'Pclass',  aggfunc = len, margins = False)
```

In [36]:
```
#Step II: Hypotheses
#Ho: Pclass and survival rates are not associated or independent
#Ha: Pclass and survival rates are associated
```

In [37]:
```
#Step III: Formulate an analysis plan
#Significance level: alpha = 0.05
#chi-square test for independence will be the test that we will do here
```

In [38]:
```
#Step IV
#Computing the degrees of freedom
df = (len(ct_Pclass_survivability_no_margins.index) - 1)*(len(ct_Pclass_survivability_no_margins.columns)- 1) #df is d
egrees of freedom
print df
n = ct_Pclass_survivability.loc['All', 'All'] #this is the common denominator for our computations
print n
E11 = ct_Pclass_survivability.loc['All', 1]* ct_Pclass_survivability.loc[0, 'All']/float(n)
E12 = ct_Pclass_survivability.loc['All', 2]* ct_Pclass_survivability.loc[0, 'All']/float(n)
E13 = ct_Pclass_survivability.loc['All', 3]* ct_Pclass_survivability.loc[0, 'All']/float(n)
E21 = ct_Pclass_survivability.loc['All', 1]* ct_Pclass_survivability.loc[1, 'All']/float(n)
E22 = ct_Pclass_survivability.loc['All', 2]* ct_Pclass_survivability.loc[1, 'All']/float(n)
E23 = ct_Pclass_survivability.loc['All', 3]* ct_Pclass_survivability.loc[1, 'All']/float(n)
arrays = [[0,1]]
tuples = list(zip(*arrays))
index = pd.MultiIndex.from_tuples(tuples, names = ['Survived'])
print index
expected_values_Pclass_df = pd.DataFrame(
                    data  = [[E11, E12, E13],
                             [E21, E22, E23]],
                    columns = [1, 2, 3],
                    index = index
)
expected_values_Pclass_df
```

```
2
891.0
Int64Index([0, 1], dtype='int64', name=u'Survived')
```

Out[38]:

	1	2	3
Survived			
0	133.090909	113.373737	302.535354
1	82.909091	70.626263	188.464646

In [39]:
```
Pclass_diff_df = ct_Pclass_survivability_no_margins - expected_values_Pclass_df
Pclass_diff_df
```

Out[39]:

Pclass	1	2	3
Survived			
0	-53.090909	-16.373737	69.464646
1	53.090909	16.373737	-69.464646

In [40]:
```
Pclass_squared_df = Pclass_diff_df.applymap(square)
Pclass_squared_df
```

Out[40]:

Pclass	1	2	3
Survived			
0	2818.644628	268.099276	4825.337108
1	2818.644628	268.099276	4825.337108

In [41]:
```
X2_df = Pclass_squared_df/expected_values_Pclass_df
X2_df
```

Out[41]:

Pclass	1	2	3
Survived			
0	21.178341	2.364739	15.949664
1	33.996810	3.796028	25.603407

In [42]:
```
X2 = X2_df.values.sum()
X2
```

Out[42]:
```
102.88898875696056
```

Step V: Interpreting the result The above value, X2, is the test statistic, the chi square random variable. Now, at this value of chi square statistic and df = 2, we find the P-value from the chi-square distribution calculator. P(X^2 > 102.8889) ~ 0.0 which is less than the significance level of 0.05 Therefore, we reject the null hypothesis.Therefore, there is a definite relationship between Pclass and survival. We can't ascertain for sure, whether this is causal or not.

Question 2.

In [43]:
```
#cleaning the titanic data of all the NaN values in the age column
age_cleaned_titanic_df = titanicdata_df[titanicdata_df.Age.notnull()]
```

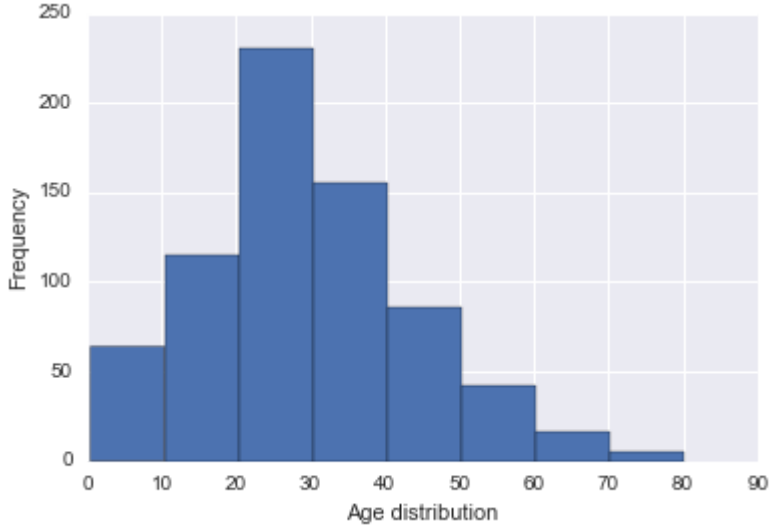
In [44]:
```
#the new dataframe has no NaN values. Now for all age related comparisons we can use this dataframe.
age_cleaned_titanic_df.loc[pd.isnull(titanicdata_df['Age'])]
```

Out[44]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FareNew

In [45]:
```
#1D analysis
%matplotlib inline
age_cleaned_titanic_df['Age'].plot.hist(bins = 8)
plt.xlabel('Age distribution')
```

Out[45]: <matplotlib.text.Text at 0x20f5d9e8>

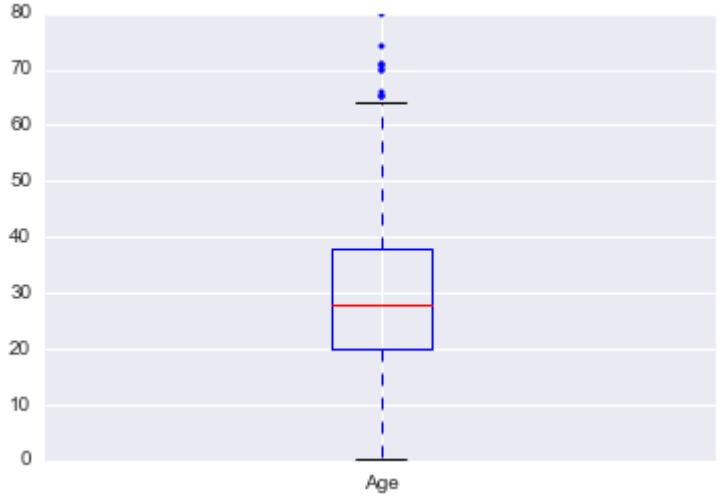


In [46]:
```
age_cleaned_titanic_df['Age'].describe()
```

Out[46]:
```
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
```

In [47]:
```
#1D analysis using box plot
%matplotlib inline
age_cleaned_titanic_df['Age'].plot.box( sym = 'k.')
#IQR = Q3 - Q1
IQR = 38 - 20.125
print IQR
#Outlier_1 > Q3 + 1.5*IQR or <Q1 - 1.5*IQR
Outlier_1 = 38 + 1.5*IQR
Outlier_2 = 20.125 - 1.5*IQR
print Outlier_1, Outlier_2
#Since there are no negative age values in the data, therefore all outliers are data points > 64.8125
```

```
17.875
64.8125 -6.6875
```



In [48]:
```
#With age, it'll be more relevant to convert ages into age-groups to analyse the data. This will give us more insight.
 Writing a
#function for the same. This function returns half-open bins with right intervals included and left intervals excluded
def age_group(df):
    bins = np.linspace(0,80,9)
    df.loc[:,'Age_group'] = pd.cut(df.loc[:,'Age'], bins, include_lowest = True)
    return df
```

In [50]:
```
age_group(age_cleaned_titanic_df)
```

Out[50]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FareNew	Ag
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NaN	S	7.2500	(20
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C	71.2833	(30
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NaN	S	7.9250	(20
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S	53.1000	(30
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NaN	S	8.0500	(30
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S	51.8625	(50
7	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750	NaN	S	21.0750	[0,
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333	NaN	S	11.1333	(20
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	NaN	C	30.0708	(10

10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7000	G6	S	16.7000	[0,
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S	26.5500	(50
12	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.0500	NaN	S	8.0500	(10
13	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.2750	NaN	S	31.2750	(30
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542	NaN	S	7.8542	(10
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16.0000	NaN	S	16.0000	(50
16	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.1250	NaN	Q	29.1250	[0,
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31	1	0	345763	18.0000	NaN	S	18.0000	(30
20	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26.0000	NaN	S	26.0000	(30
21	22	1	2	Beesley, Mr.	male	34	0	0	248698	13.0000	D56	S	13.0000	(30

				Lawrence										
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292	NaN	Q	8.0292	(10
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5000	A6	S	35.5000	(20
24	25	0	3	Palsson, Miss. Torborg Danira	female	8	3	1	349909	21.0750	NaN	S	21.0750	[0,
25	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38	1	5	347077	31.3875	NaN	S	31.3875	(30
27	28	0	1	Fortune, Mr. Charles Alexander	male	19	3	2	19950	263.0000	C23 C25 C27	S	263.0000	(10
30	31	0	1	Uruchurtu, Don. Manuel E	male	40	0	0	PC 17601	27.7208	NaN	C	27.7208	(30
33	34	0	2	Wheadon, Mr. Edward H	male	66	0	0	C.A. 24579	10.5000	NaN	S	10.5000	(60
34	35	0	1	Meyer, Mr. Edgar Joseph	male	28	1	0	PC 17604	82.1708	NaN	C	82.1708	(20
35	36	0	1	Holverson, Mr. Alexander Oskar	male	42	1	0	113789	52.0000	NaN	S	52.0000	(40
				Cann, Mr.										

37	38	0	3	Ernest Charles	male	21	0	0	A./5. 2152	8.0500	NaN	S	8.0500	(20
38	39	0	3	Vander Planke, Miss. Augusta Maria	female	18	2	0	345764	18.0000	NaN	S	18.0000	(10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
856	857	1	1	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45	1	1	36928	164.8667	NaN	S	164.8667	(40
857	858	1	1	Daly, Mr. Peter Denis	male	51	0	0	113055	26.5500	E17	S	26.5500	(50
858	859	1	3	Baclini, Mrs. Solomon (Latifa Qurban)	female	24	0	3	2666	19.2583	NaN	C	19.2583	(20
860	861	0	3	Hansen, Mr. Claus Peter	male	41	2	0	350026	14.1083	NaN	S	14.1083	(40
861	862	0	2	Giles, Mr. Frederick Edward	male	21	1	0	28134	11.5000	NaN	S	11.5000	(20
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48	0	0	17466	25.9292	D17	S	25.9292	(40
864	865	0	2	Gill, Mr. John William	male	24	0	0	233866	13.0000	NaN	S	13.0000	(20
				Bystrom,										

865	866	1	2	Mrs. (Karolina)	female	42	0	0	236852	13.0000	NaN	S	13.0000	(40
866	867	1	2	Duran y More, Miss. Asuncion	female	27	1	0	SC/PARIS 2149	13.8583	NaN	C	13.8583	(20
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31	0	0	PC 17590	50.4958	A24	S	50.4958	(30
869	870	1	3	Johnson, Master. Harold Theodor	male	4	1	1	347742	11.1333	NaN	S	11.1333	[0,
870	871	0	3	Balkic, Mr. Cerin	male	26	0	0	349248	7.8958	NaN	S	7.8958	(20
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47	1	1	11751	52.5542	D35	S	52.5542	(40
872	873	0	1	Carlsson, Mr. Frans Olof	male	33	0	0	695	5.0000	B51 B53 B55	S	5.0000	(30
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47	0	0	345765	9.0000	NaN	S	9.0000	(40
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28	1	0	P/PP 3381	24.0000	NaN	C	24.0000	(20
875	876	1	3	Najib, Miss. Adele Kiamie	female	15	0	0	2667	7.2250	NaN	C	7.2250	(10

				"Jane"									
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20	0	0	7534	9.8458	NaN	S	9.8458	(10
877	878	0	3	Petroff, Mr. Nedelio	male	19	0	0	349212	7.8958	NaN	S	7.8958	(10
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56	0	1	11767	83.1583	C50	C	83.1583	(50
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25	0	1	230433	26.0000	NaN	S	26.0000	(20
881	882	0	3	Markun, Mr. Johann	male	33	0	0	349257	7.8958	NaN	S	7.8958	(30
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22	0	0	7552	10.5167	NaN	S	10.5167	(20
883	884	0	2	Banfield, Mr. Frederick James	male	28	0	0	C.A./SOTON 34068	10.5000	NaN	S	10.5000	(20
884	885	0	3	Sutehall, Mr. Henry Jr	male	25	0	0	SOTON/OQ 392076	7.0500	NaN	S	7.0500	(20
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39	0	5	382652	29.1250	NaN	Q	29.1250	(30
886	887	0	2	Montvila, Rev. Juozas	male	27	0	0	211536	13.0000	NaN	S	13.0000	(20

887	888	1	1	Graham, Miss. Margaret Edith	female	19	0	0	112053	30.0000	B42	S	30.0000	(10
889	890	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	30.0000	C148	C	30.0000	(20
890	891	0	3	Dooley, Mr. Patrick	male	32	0	0	370376	7.7500	NaN	Q	7.7500	(30

714 rows × 14 columns

In [51]:
```
age_cleaned_titanic_df.groupby('Age_group').sum()
```

Out[51]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew
Age_group								
[0, 10]	27574	38	169	273.17	118	91	1947.8041	1947.8041
(10, 20]	51481	44	291	1991.50	68	45	3395.8961	3403.9461
(20, 30]	98597	84	549	5847.50	74	55	6510.5453	6518.5953
(30, 40]	72647	69	324	5433.00	58	61	6586.8955	6775.8080
(40, 50]	41581	33	165	3902.00	32	37	3540.0336	3548.0836
(50, 60]	18892	17	64	2305.50	13	13	1880.5417	1880.5417
(60, 70]	7325	4	26	1086.00	3	6	780.4833	780.4833
(70, 80]	2191	1	9	366.50	0	0	129.6834	129.6834

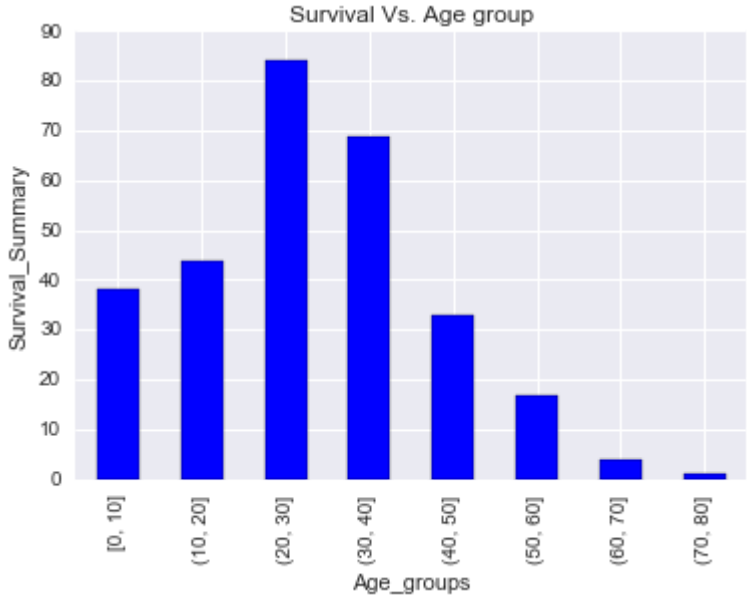
In [52]:
```
age_cleaned_titanic_df.groupby('Age_group').sum()['Survived']
```

Out[52]:
```
Age_group
[0, 10]     38
(10, 20]    44
(20, 30]    84
(30, 40]    69
(40, 50]    33
(50, 60]    17
(60, 70]     4
(70, 80]     1
Name: Survived, dtype: int64
```

In [53]:
```
survival_summary_by_age = age_cleaned_titanic_df.groupby('Age_group').sum()['Survived']
```

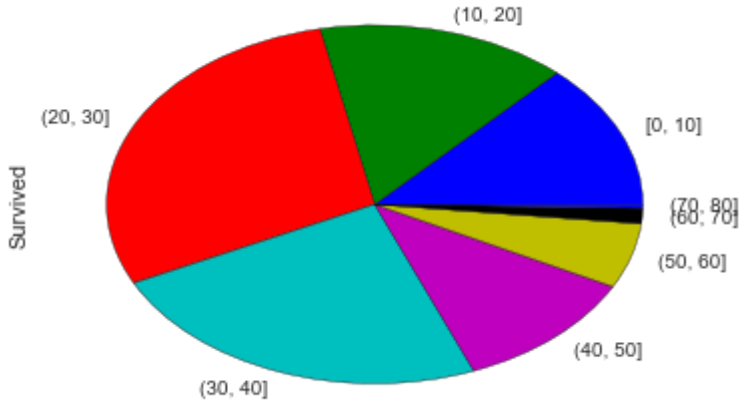
In [54]:
```
#2D Analysis
#Plotting a bar graph to plot the age group data Vs. Survival summary
%matplotlib inline
survival_summary_by_age.plot.bar()
plt.ylabel('Survival_Summary')
plt.xlabel('Age_groups')
plt.title('Survival Vs. Age group')
```

Out[54]: <matplotlib.text.Text at 0x21021e10>



In [55]:
```
#Plotting a pie chart to summarize the survival summar.plot.pie()
survival_summary_by_age.plot.pie()
```

Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x214d0860>



This doesn't give me enough insight because this is absolute information regarding age_group data for survival summary. What will be interesting to see is what is the mean value of Survived class in each age group. Since Survived is a categorical variable with 1 assigned only to those who've survived, higher mean survived value will indicate higher likelyhood of surival in that age group.

Nevertheless, the pie plot gives me information about which is the largest group in terms of number of survivors. (20,30] seems to have the most number of survivors but it could be due to higher number of passengers belonging to that group. Therefore, let's look at mean survival rate per age group.

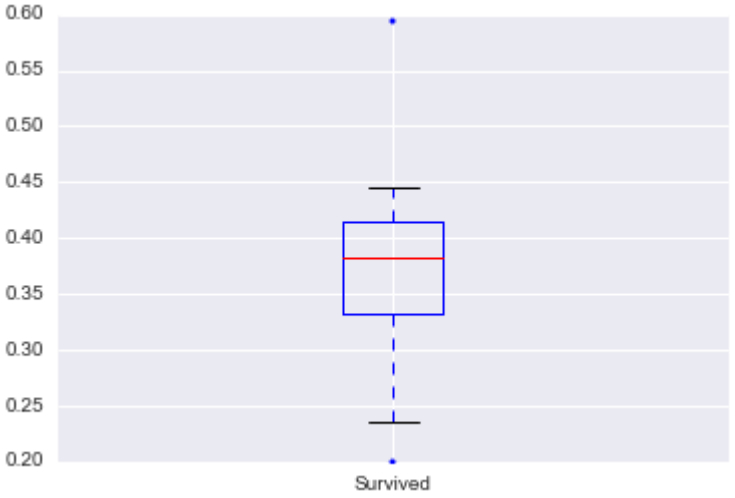
In [56]:
```
age_cleaned_titanic_df.groupby('Age_group').mean()['Survived']
```

Out[56]:
```
Age_group
[0, 10]     0.593750
(10, 20]    0.382609
(20, 30]    0.365217
(30, 40]    0.445161
(40, 50]    0.383721
(50, 60]    0.404762
(60, 70]    0.235294
(70, 80]    0.200000
Name: Survived, dtype: float64
```

In [57]:
```
survival_mean_by_age_group = age_cleaned_titanic_df.groupby('Age_group').mean()['Survived']
```

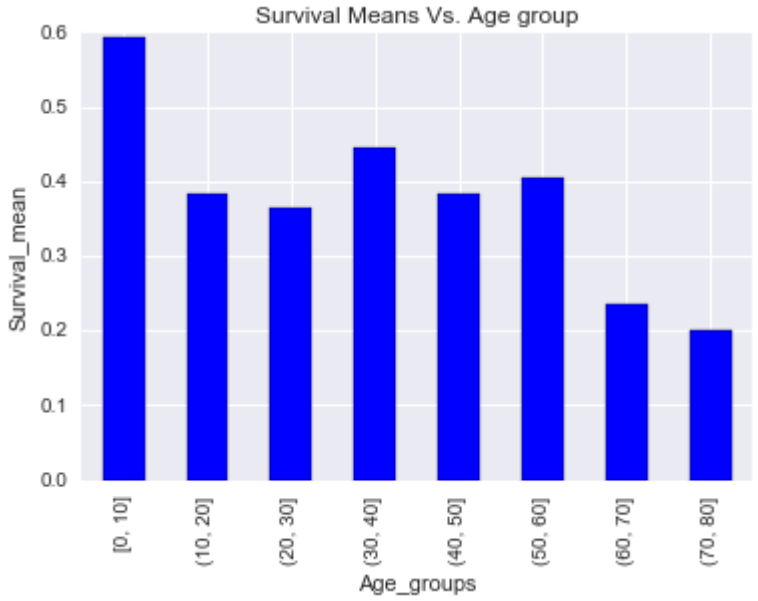
In [58]:
```
%matplotlib inline
survival_mean_by_age_group.plot.box(sym = 'k.')
```

Out[58]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x215e5c88>
```



In [59]:
```
#2D Analysis
survival_mean_by_age_group.plot.bar()
plt.ylabel('Survival_mean')
plt.xlabel('Age_groups')
plt.title('Survival Means Vs. Age group')
```

Out[59]: <matplotlib.text.Text at 0x218fae80>



This looks like following. The peak is at (0,10] thus indicating that saving infants and kids was somewhat a priority. Maybe, even rescue missions focused on rescuing children first as that is a more common strategy in case of catastrophes. The correlation coefficient between age and survived is ~0 (from our earlier correlation matrix) which suggests that there was weak linear relation between age of the passenger and odds of his survival. Now let's run some statistical tests to explore further. We'll first create a contingency table to plot the two categorical variables age_group and survived

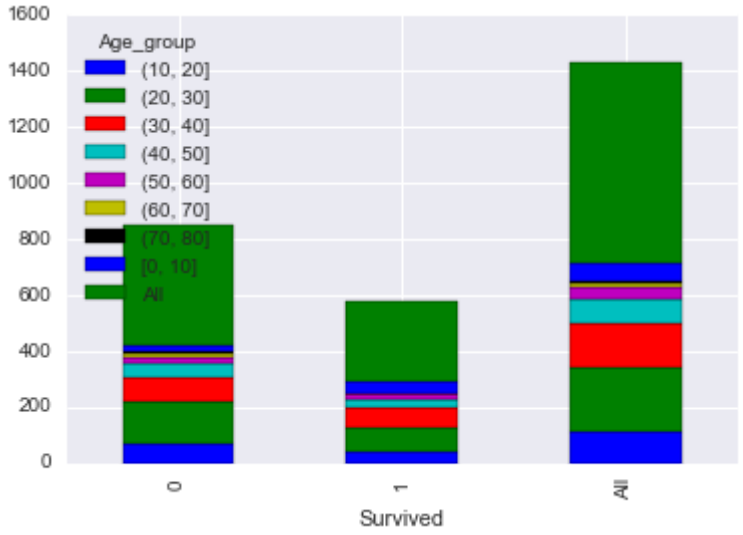
In [60]:
```
#create contingency table
#Age groups to survivability
ct_table_age_group = pd.pivot_table(data = age_cleaned_titanic_df[['Survived', 'Age_group']], index = 'Survived', \
                                    columns = ['Age_group'], aggfunc=len, margins = True)
ct_table_age_group
```

Out[60]:

Age_group	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	[0, 10]	All
Survived									
0	71	146	86	53	25	13	4	26	424
1	44	84	69	33	17	4	1	38	290
All	115	230	155	86	42	17	5	64	714

In [61]:
```
ct_table_age_group.plot.bar(stacked = True)
```

Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0x2166c630>



Chi-square test for independence: This test is used to compute whether there is a significant association between two categorical variables. Firstly, I will state the hypotheses. Ho: The two categorical variables, 'Survived' and 'Age_group' are independent i.e. there is no impact of age group on survival Ha: The two variables are not independent i.e. there is impact of age group on survival Assumption: The significance level is alpha = 0.05 for this test.

In [62]:
```
ct_table_age_group_no_margin = pd.pivot_table(data = age_cleaned_titanic_df[['Survived', 'Age_group']], index = 'Survi
ved', \
                                      columns = ['Age_group'], aggfunc=len, margins = False)
```

In [63]:
```
arrays = [[0,1]]
tuples = list(zip(*arrays))
index = pd.MultiIndex.from_tuples(tuples, names = ['Survived'])
index
```

Out[63]: Int64Index([0, 1], dtype='int64', name=u'Survived')

In [64]:
```
#Doing the Chi-square test of independence
#Computing the degrees of freedom
df = (len(ct_table_age_group_no_margin.index) - 1)*(len(ct_table_age_group_no_margin.columns)- 1) #df is degrees of fr
eedom
print df
n = ct_table_age_group.loc['All', 'All'] #this is the common denominator for our computations
print n
E11 = ct_table_age_group.loc['All', '(10, 20]']* ct_table_age_group.loc[0, 'All']/float(n)
E12 = ct_table_age_group.loc['All', '(20, 30]']* ct_table_age_group.loc[0, 'All']/float(n)
E13 = ct_table_age_group.loc['All', '(30, 40]']* ct_table_age_group.loc[0, 'All']/float(n)
E14 = ct_table_age_group.loc['All', '(40, 50]']* ct_table_age_group.loc[0, 'All']/float(n)
E15 = ct_table_age_group.loc['All', '(50, 60]']* ct_table_age_group.loc[0, 'All']/float(n)
E16 = ct_table_age_group.loc['All', '(60, 70]']* ct_table_age_group.loc[0, 'All']/float(n)
E17 = ct_table_age_group.loc['All', '(70, 80]']* ct_table_age_group.loc[0, 'All']/float(n)
E18 = ct_table_age_group.loc['All', '[0, 10]']* ct_table_age_group.loc[0, 'All']/float(n)
E21 = ct_table_age_group.loc['All', '(10, 20]']* ct_table_age_group.loc[1, 'All']/float(n)
E22 = ct_table_age_group.loc['All', '(20, 30]']* ct_table_age_group.loc[1, 'All']/float(n)
E23 = ct_table_age_group.loc['All', '(30, 40]']* ct_table_age_group.loc[1, 'All']/float(n)
E24 = ct_table_age_group.loc['All', '(40, 50]']* ct_table_age_group.loc[1, 'All']/float(n)
E25 = ct_table_age_group.loc['All', '(50, 60]']* ct_table_age_group.loc[1, 'All']/float(n)
E26 = ct_table_age_group.loc['All', '(60, 70]']* ct_table_age_group.loc[1, 'All']/float(n)
E27 = ct_table_age_group.loc['All', '(70, 80]']* ct_table_age_group.loc[1, 'All']/float(n)
E28 = ct_table_age_group.loc['All', '[0, 10]']* ct_table_age_group.loc[1, 'All']/float(n)
expected_values_df = pd.DataFrame(
                    data = [[E11, E12, E13, E14, E15, E16, E17, E18],
                            [E21, E22, E23, E24, E25, E26, E27, E28]],
                    columns = ['(10, 20]', '(20, 30]' , '(30, 40]', '(40, 50]', '(50, 60]', '(60, 70]', '(70, 80]',
'[0, 10]'],

                    index = index
    )
```

```
7
714.0
```

In [65]:
```
expected_values_df
```

Out[65]:

	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	[0, 10]
Survived								
0	68.291317	136.582633	92.044818	51.070028	24.941176	10.095238	2.969188	38.005602
1	46.708683	93.417367	62.955182	34.929972	17.058824	6.904762	2.030812	25.994398

In [66]:
```
ct_table_age_group_no_margin - expected_values_df
```

Out[66]:

Age_group	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	[0, 10]
Survived								
0	2.708683	9.417367	-6.044818	1.929972	0.058824	2.904762	1.030812	-12.005602
1	-2.708683	-9.417367	6.044818	-1.929972	-0.058824	-2.904762	-1.030812	12.005602

In [67]:
```
X2_array = ct_table_age_group_no_margin - expected_values_df
def square(x):
    return x**2
X2_array.applymap(square)
```

Out[67]:

Age_group	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	[0, 10]
Survived								
0	7.336966	88.6868	36.539824	3.724792	0.00346	8.437642	1.062574	144.134485
1	7.336966	88.6868	36.539824	3.724792	0.00346	8.437642	1.062574	144.134485

In [68]:
```
X2_df = X2_array.applymap(square)
X2_df/expected_values_df
```

Out[68]:

Age_group	(10, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 60]	(60, 70]	(70, 80]	[0, 10]
Survived								
0	0.107436	0.649327	0.396979	0.072935	0.000139	0.835804	0.357867	3.792454
1	0.157079	0.949361	0.580410	0.106636	0.000203	1.222003	0.523226	5.544829

In [69]:
```
X2_df_sum = X2_df/expected_values_df
print X2_df_sum.values.sum() #doing summation to get the test statistic for chi-square test for independence
```

```
15.2966877495
```

In [70]:
```
X2 = X2_df_sum.values.sum()
#This is the test statistic, the chi square random variable. Now, at this value of chi square statistic and df = 7, we
 find the
#P-value from the chi-square distribution calculator. P(X^2 > 15.29668) = 0.03 which is less than the significance lev
el of 0.05
#Therefore, we reject the null hypothesis.Therefore, there is a relationship between age group and survival. We can't
 ascertain
#for sure, whether this is causal or not.
#One must notice the importance of alpha = 0.05 as a significance level here. Had the significance level been 0.01, H0
 would've
#been retained and we could've said that there is no relationship between age group and survival.
```

Question 3

Now we look at grouping this data by gender and visualizing the data for these group labels. Since our column for gender has no missing/incorrect data, we use the original dataframe titanicdata_df to perform these computations.

In [71]:
```
titanicdata_df.groupby('Sex').sum()
```

Out[71]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew
Sex								
female	135343	233	678	7286.00	218	204	13966.6628	13966.6628
male	262043	109	1379	13919.17	248	136	14727.2865	15146.4240

In [72]:
```
titanicdata_df.groupby('Sex').size()
```

Out[72]:
```
Sex
female    314
male      577
dtype: int64
```

In [73]:
```
titanicdata_df.groupby('Sex').mean()
```

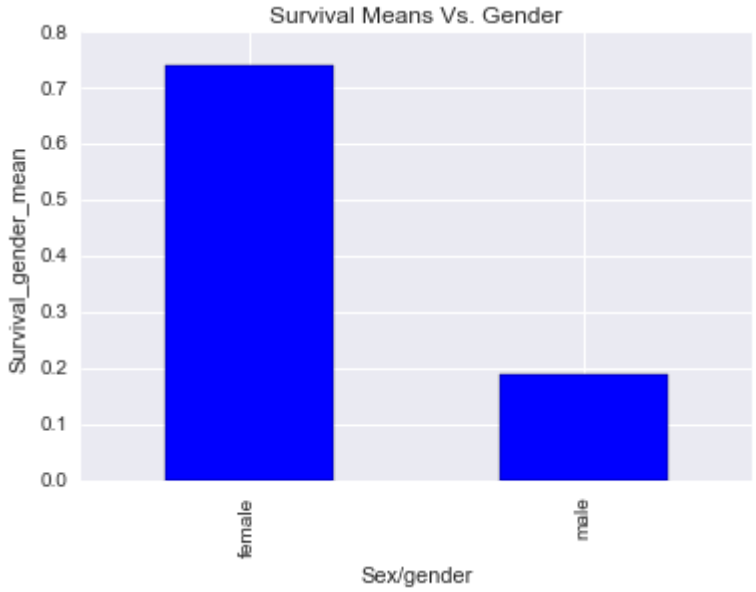
Out[73]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew
Sex								
female	431.028662	0.742038	2.159236	27.915709	0.694268	0.649682	44.479818	44.479818
male	454.147314	0.188908	2.389948	30.726645	0.429809	0.235702	25.523893	26.250302

This indicates that <20% of total males in the dataset have survived since the sum and mean of survived data is only contributed by 1s and hence takes into account those who have survived the crash. Almost 75% of females have survived the crash. These are telling statistic to measure the survival probability based on gender. Let's plot a bar graph now to visualize this.

In [74]:
```
survival_by_gender = titanicdata_df.groupby('Sex').mean()['Survived']
survival_by_gender.plot.bar()
plt.ylabel('Survival_gender_mean')
plt.xlabel('Sex/gender')
plt.title('Survival Means Vs. Gender')
```

Out[74]: <matplotlib.text.Text at 0x21f88f28>



In [75]:
```
#Now let's look at some linear correlation between the two variables. For this, we need to convert gender variable int
o numeric
#values. For the sake of this exercise, female = 0 and male = 1.
def gender_numeric(df):
    df['gender_numeric'] = np.where(df['Sex'] == 'female', 0, 1)
    return df
```

In [76]:
```
gender_numeric(titanicdata_df)
```

Out[76]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FareNew	ge
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	NaN	S	7.2500	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C85	C	71.2833	0
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	NaN	S	7.9250	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C123	S	53.1000	0
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	NaN	S	8.0500	1
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q	8.4583	1
6	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S	51.8625	1
7	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750	NaN	S	21.0750	1
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333	NaN	S	11.1333	0
				Nasser, Mrs.										

9	10	1	2	Nicholas (Adele Achem)	female	14	1	0	237736	30.0708	NaN	C	30.0708	0
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7000	G6	S	16.7000	0
11	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.5500	C103	S	26.5500	0
12	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.0500	NaN	S	8.0500	1
13	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.2750	NaN	S	31.2750	1
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542	NaN	S	7.8542	0
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16.0000	NaN	S	16.0000	0
16	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.1250	NaN	Q	29.1250	1
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S	13.0000	1
				Vander Planke, Mrs. Julius										

18	19	0	3	(Emelia Maria Vande...	female	31	1	0	345763	18.0000	NaN	S	18.0000	0
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C	7.2250	0
20	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26.0000	NaN	S	26.0000	1
21	22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13.0000	D56	S	13.0000	1
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292	NaN	Q	8.0292	0
23	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5000	A6	S	35.5000	1
24	25	0	3	Palsson, Miss. Torborg Danira	female	8	3	1	349909	21.0750	NaN	S	21.0750	0
25	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38	1	5	347077	31.3875	NaN	S	31.3875	0
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	C	7.2250	1
27	28	0	1	Fortune, Mr. Charles Alexander	male	19	3	2	19950	263.0000	C23 C25 C27	S	263.0000	1
				O'Dwyer,										

28	29	1	3	Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Q	7.8792	0
29	30	0	3	Todoroff, Mr. Lalio	male	NaN	0	0	349216	7.8958	NaN	S	7.8958	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
861	862	0	2	Giles, Mr. Frederick Edward	male	21	1	0	28134	11.5000	NaN	S	11.5000	1
862	863	1	1	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48	0	0	17466	25.9292	D17	S	25.9292	0
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S	69.5500	0
864	865	0	2	Gill, Mr. John William	male	24	0	0	233866	13.0000	NaN	S	13.0000	1
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42	0	0	236852	13.0000	NaN	S	13.0000	0
866	867	1	2	Duran y More, Miss. Asuncion	female	27	1	0	SC/PARIS 2149	13.8583	NaN	C	13.8583	0
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31	0	0	PC 17590	50.4958	A24	S	50.4958	1
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S	9.5000	1

869	870	1	3	Johnson, Master. Harold Theodor	male	4	1	1	347742	11.1333	NaN	S	11.1333	1
870	871	0	3	Balkic, Mr. Cerin	male	26	0	0	349248	7.8958	NaN	S	7.8958	1
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47	1	1	11751	52.5542	D35	S	52.5542	0
872	873	0	1	Carlsson, Mr. Frans Olof	male	33	0	0	695	5.0000	B51 B53 B55	S	5.0000	1
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47	0	0	345765	9.0000	NaN	S	9.0000	1
874	875	1	2	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28	1	0	P/PP 3381	24.0000	NaN	C	24.0000	0
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15	0	0	2667	7.2250	NaN	C	7.2250	0
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20	0	0	7534	9.8458	NaN	S	9.8458	1
877	878	0	3	Petroff, Mr. Nedelio	male	19	0	0	349212	7.8958	NaN	S	7.8958	1
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S	7.8958	1

879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56	0	1	11767	83.1583	C50	C	83.1583	0
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)	female	25	0	1	230433	26.0000	NaN	S	26.0000	0
881	882	0	3	Markun, Mr. Johann	male	33	0	0	349257	7.8958	NaN	S	7.8958	1
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22	0	0	7552	10.5167	NaN	S	10.5167	0
883	884	0	2	Banfield, Mr. Frederick James	male	28	0	0	C.A./SOTON 34068	10.5000	NaN	S	10.5000	1
884	885	0	3	Sutehall, Mr. Henry Jr	male	25	0	0	SOTON/OQ 392076	7.0500	NaN	S	7.0500	1
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39	0	5	382652	29.1250	NaN	Q	29.1250	0
886	887	0	2	Montvila, Rev. Juozas	male	27	0	0	211536	13.0000	NaN	S	13.0000	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19	0	0	112053	30.0000	B42	S	30.0000	0
888	889	0	3	Johnston, Miss. Catherine	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	23.4500	0

				Helen "Carrie"									
889	890	1	1	Behr, Mr. Karl Howell	male	26	0	0	111369	30.0000	C148	C	30.0000	1
890	891	0	3	Dooley, Mr. Patrick	male	32	0	0	370376	7.7500	NaN	Q	7.7500	1

891 rows × 14 columns

In [77]:
```
titanicdata_df.corr()
```

Out[77]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew	gender_numeric
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658	0.018753	0.042939
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	0.250635	-0.543351
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	-0.561243	0.131900
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.099377	0.093254
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	0.155423	-0.114631
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	0.212103	-0.245489
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	0.995568	-0.182333
FareNew	0.018753	0.250635	-0.561243	0.099377	0.155423	0.212103	0.995568	1.000000	-0.175647
gender_numeric	0.042939	-0.543351	0.131900	0.093254	-0.114631	-0.245489	-0.182333	-0.175647	1.000000

The correlation coefficient and gender_numeric r, is -0.54. This indicates that as we move from 0 to 1 in gender_numeric variable, there is a decrease in survived variable from 1 to 0. 54% of this variation can be correlated to gender change. Let's plot a scatterplot to view this more.

In [78]:
```
titanicdata_df.groupby('gender_numeric', as_index = False).mean()
```

Out[78]:

	gender_numeric	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	FareNew
0	0	431.028662	0.742038	2.159236	27.915709	0.694268	0.649682	44.479818	44.479818
1	1	454.147314	0.188908	2.389948	30.726645	0.429809	0.235702	25.523893	26.250302

With this we can say the mean survival rate of women is much higher than men. So being a woman is likely to help you survive. That said, these findings are tentative at best and a more rigourous statistic test is required to understand whether these results hold statistical ground.

A statistical test is required to prove any association between gender and survivability. Since these both are categorical variables, I'd choose a chi-square test for the same. In this test, there are four steps to come to a conclusion: Step I: Build a contingency table Step II: State the hypotheses for the tests Step III: Formulate an analysis plan Step IV: Analyze data Step V: Interpret results

In [79]:
```
#Step I: A contingency table
ct_gender_survivability = pd.pivot_table(data = titanicdata_df[['Survived', 'Sex']], index = 'Survived', columns = 'Se
x',
                                         aggfunc = len, margins = True)
ct_gender_survivability
```

Out[79]:

Sex	female	male	All
Survived			
0	81	468	549
1	233	109	342
All	314	577	891

In [80]:
```
ct_gender_survivability_no_margin = pd.pivot_table(data = titanicdata_df[['Survived', 'Sex']], index = 'Survived', col
umns = 'Sex',
                                         aggfunc = len, margins = False)
```

In [81]:
```
#Step II: Hypotheses
#Ho: Gender and survival rates are not associated or independent
#Ha: Gender and survival rates are associated
```

In [82]:
```
#Step III: Analysis plan
#Significance level is set at alpha = 0.05
#Statistical test to be performed: Chi-square test for independence
```

In [84]:
```
#Step IV
#Computing the degrees of freedom
df = (len(ct_gender_survivability_no_margin.index) - 1)*(len(ct_gender_survivability_no_margin)- 1) #df is degrees of
 freedom
print df
n = ct_gender_survivability.loc['All', 'All'] #this is the common denominator for our computations
print n
E11 = ct_gender_survivability.loc['All', 'female']* ct_gender_survivability.loc[0, 'All']/float(n)
E12 = ct_gender_survivability.loc['All', 'male']* ct_gender_survivability.loc[0, 'All']/float(n)
E21 = ct_gender_survivability.loc['All', 'female']* ct_gender_survivability.loc[1, 'All']/float(n)
E22 = ct_gender_survivability.loc['All', 'male']* ct_gender_survivability.loc[1, 'All']/float(n)
arrays = [[0,1]]
tuples = list(zip(*arrays))
index = pd.MultiIndex.from_tuples(tuples, names = ['Survived'])
print index
expected_values_gender_df = pd.DataFrame(
                    data  = [[E11, E12],
                             [E21, E22]],
                    columns = ['female', 'male'],
                    index = index
)
expected_values_gender_df
```

```
1
891.0
Int64Index([0, 1], dtype='int64', name=u'Survived')
```

Out[84]:

	female	male
Survived		
0	193.474747	355.525253
1	120.525253	221.474747

In [85]:
```
chi_table_df = ct_gender_survivability_no_margin - expected_values_gender_df
chi_table_df
```

Out[85]:

Sex	female	male
Survived		
0	-112.474747	112.474747
1	112.474747	-112.474747

In [86]:
```
chi_table_df.applymap(square)
```

Out[86]:

Sex	female	male
Survived		
0	12650.56882	12650.56882
1	12650.56882	12650.56882

In [87]:
```
X2_sum_gender_df = chi_table_df.applymap(square)/expected_values_gender_df
X2_sum_gender_df
```

Out[87]:

Sex	female	male
Survived		
0	65.386150	35.582757
1	104.961977	57.119690

In [88]:
```
X2_value = X2_sum_gender_df.values.sum()
X2_value
```

Out[88]: 263.05057407065567

Step V: This is the test statistic, the chi square random variable. Now, at this value of chi square statistic and df = 1, we find the P-value from the chi-square distribution calculator. P(X^2 > 263.0505) = 0.0 which is less than the significance level of 0.05 Therefore, we reject the null hypothesis.Therefore, there is a definite relationship between gender and survival. We can't ascertain for sure, whether this is causal or not.

In [89]:
```
#Now, let's try to see the interplay of data within each sub-groups of independent variables.
titanicdata_df.groupby(('Pclass','Sex')).mean()
```

Out[89]:

		PassengerId	Survived	Age	SibSp	Parch	Fare	FareNew	gender_numeric
Pclass	**Sex**								
1	**female**	469.212766	0.968085	34.611765	0.553191	0.457447	106.125798	106.125798	0
	male	455.729508	0.368852	41.281386	0.311475	0.278689	67.226127	69.696926	1
2	**female**	443.105263	0.921053	28.722973	0.486842	0.605263	21.970121	21.970121	0
	male	447.962963	0.157407	30.740707	0.342593	0.222222	19.741782	20.533449	1
3	**female**	399.729167	0.500000	21.750000	0.895833	0.798611	16.118810	16.118810	0
	male	455.515850	0.135447	26.507589	0.498559	0.224784	12.661633	12.754428	1

In [90]:
```
titanicdata_df.groupby(('Pclass','Sex')).count()
```

Out[90]:

		PassengerId	Survived	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	FareNew	gender_numeric
Pclass	**Sex**												
1	**female**	94	94	94	85	94	94	94	94	81	92	94	94
	male	122	122	122	101	122	122	122	122	95	122	122	122
2	**female**	76	76	76	74	76	76	76	76	10	76	76	76
	male	108	108	108	99	108	108	108	108	6	108	108	108
3	**female**	144	144	144	102	144	144	144	144	6	144	144	144
	male	347	347	347	253	347	347	347	347	6	347	347	347

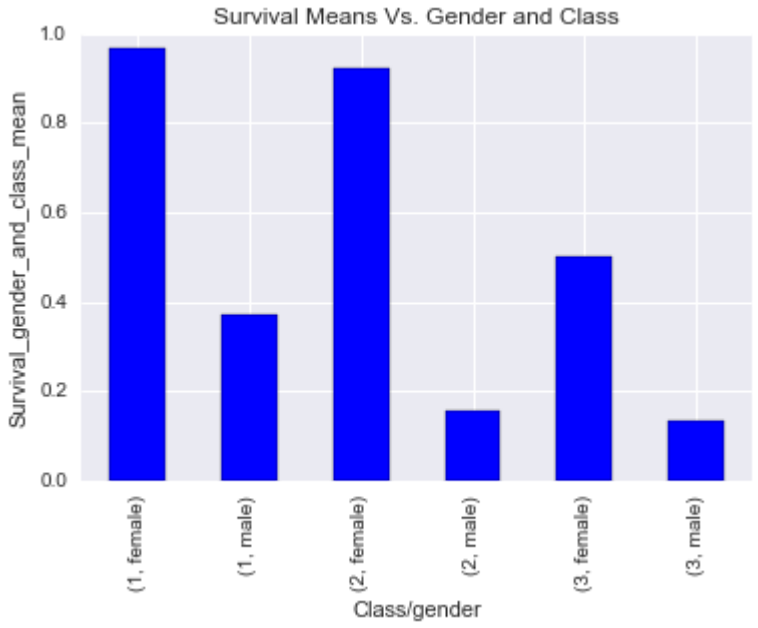
In [91]:
```
survival_by_class_and_gender = titanicdata_df.groupby(('Pclass','Sex')).mean()['Survived']
```

In [92]:
```
print survival_by_class_and_gender
```

```
Pclass  Sex
1       female    0.968085
        male      0.368852
2       female    0.921053
        male      0.157407
3       female    0.500000
        male      0.135447
Name: Survived, dtype: float64
```

In [93]:
```
survival_by_class_and_gender.plot.bar()
plt.ylabel('Survival_gender_and_class_mean')
plt.xlabel('Class/gender')
plt.title('Survival Means Vs. Gender and Class')
```

Out[93]: <matplotlib.text.Text at 0x220a91d0>



This plot is consistent with our first analysis about correlation between socio-economic status and survival rates. There we saw a negative correlation between values of Pclass and Survived. Even in the plot above, mean survived rates come down for both males and females as socio-economic status decreases from 1 to 3. (P.S. Higher number indicates lower status).