Introduction to Data Analysis - P2 Project - Titanic Data

This project will analyze passanger data from the Titanic and will evaluate the corrolation between passanger survivor rate and the following data points:

1. Did the passanger's gender play any role in their chance of survival?
2. Did the passanger's Age play any role in their chance of survival?
3. Did the amount payed by the passenger for their ticket play any role in their chance of survival?
4. Do passangers with any family memebers aboard have a higher survival rate?

Coloumn Heading Legend:

- Survived: Survived (1) or died (0)
- Pclass: Passenger’s class
- Name: Passenger’s name
- Sex: Passenger’s sex
- Age: Passenger’s age
- SibSp: Number of siblings/spouses aboard
- Parch: Number of parents/children aboard
- Ticket: Ticket Number
- Fare: Fare Paid ofr Ticket
- Cabin: Cabin Number
- Embarked: Port of Embarkation

Loading the Data from CSV

In [834]:

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

# load csv file into dataframe
filename = 'titanic_data.csv'
titanic_df = pd.read_csv(filename)

# display top 5 rows of data
titanic_df.head()
```

```
Populating the interactive namespace from numpy and matplotlib
```

Out[834]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Evaluate Data Values

In [835]:

```
# dataframe statistics
titanic_df.describe()
```

Out[835]:

	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	NaN	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	NaN	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	NaN	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [836]:

```
titanic_df.dtypes
```

Out[836]:

```
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 [837]:

```
# evaluate unique values we might use for analysis later
for column in titanic_df:
    if column in ['Survived', 'Pclass', 'Sex', 'SibSp', 'Parch']:
        print "{} values: {}".format(column, titanic_df[column].unique())
```

```
Survived values: [0 1]
Pclass values: [3 1 2]
Sex values: ['male' 'female']
SibSp values: [1 0 3 4 2 5 8]
Parch values: [0 1 2 5 3 4 6]
```

In [838]:

```
# Examine groups
titanic_df.count()
```

Out[838]:

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

Data Cleanup

In [839]:

```
# Intentionally leaving loaded data as is and populating new columns with clean data to make analysis easier

# Create Alone column based on non-zero values in SibSp and ParCh
titanic_df['Alone'] = (titanic_df['SibSp'] + titanic_df['Parch']) > 0

# Create Adult True/False values for passangers by age
titanic_df['Adult'] = (titanic_df['Age'] >= 18)

# Create Male True/False column
titanic_df['Male'] = (titanic_df['Sex'] == 'male')

# Create Survival True/False column
titanic_df['SurvivedTF'] = (titanic_df['Survived'] == 1)

titanic_df.head()
```

Out[839]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Alone	Adult	Male	Su
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	True	True	True	Fal
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	True	True	False	Tru
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	False	True	False	Tru
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	True	True	False	Tru
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	False	True	True	Fal

Data Analysis

In [840]:

```
survival_grouping = titanic_df.groupby(['Survived','Male','Adult','Alone'], as_index=False)['PassengerId'].count()
survival_grouping
```

Out[840]:

	Survived	Male	Adult	Alone	PassengerId
0	0	False	False	False	8
1	0	False	False	True	26
2	0	False	True	False	19
3	0	False	True	True	28
4	0	True	False	False	105
5	0	True	False	True	38
6	0	True	True	False	242
7	0	True	True	True	83
8	1	False	False	False	29
9	1	False	False	True	45
10	1	False	True	False	70
11	1	False	True	True	89
12	1	True	False	False	14
13	1	True	False	True	25
14	1	True	True	False	50
15	1	True	True	True	20

In [841]:

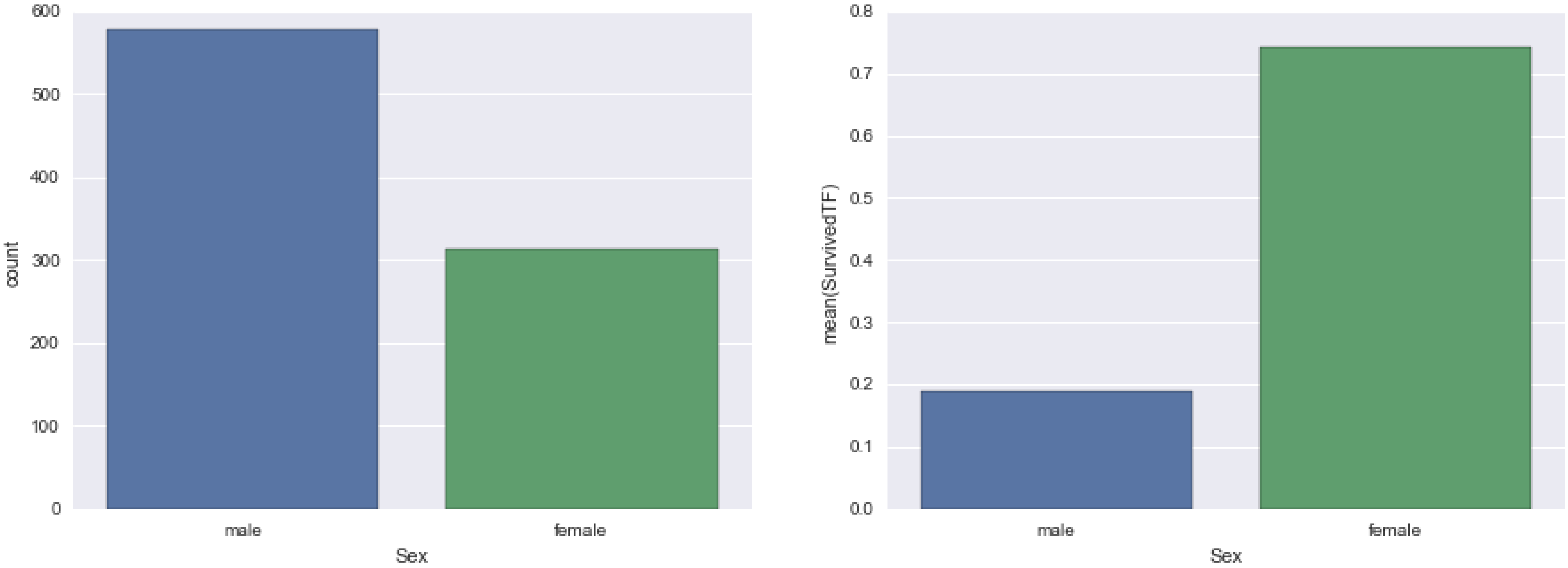
```
# Split main table into two for the ones that made it and the ones that didn't
madeit_df = titanic_df.query('SurvivedTF == True')
didnt_df = titanic_df.query('SurvivedTF == False')
```

Did the passanger's gender play any role in their chance of survival?

```
In [842]: # Count of Passangers by Sex
          fig, (axis1,axis2) = plt.subplots(1,2,figsize=(15,5))
          sns.countplot(x='Sex', data=titanic_df, ax=axis1)

          # average of survived by Sex
          grouped_by_sex = titanic_df[['Sex', 'SurvivedTF']].groupby(['Sex'],as_index=False).mean()
          sns.barplot(x='Sex', y='SurvivedTF', data=grouped_by_sex, ax=axis2, order=['male','female'])
```

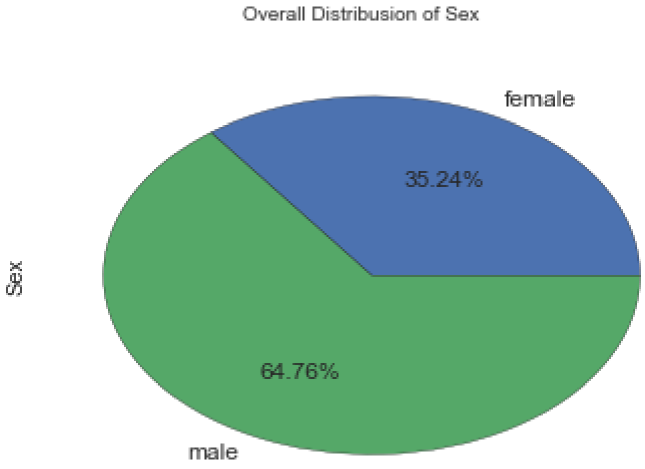
```
Out[842]: <matplotlib.axes._subplots.AxesSubplot at 0x5405f6a0>
```



```
In [843]: titanic_df.groupby(['Sex'], as_index=True)['Sex'].count() \
          .plot.pie(subplots=True, title='Overall Distribusion of Sex', autopct='%.2f%%', fontsize=12)
          titanic_df.groupby(['Sex'], as_index=False)['PassengerId'].count()
```

Out[843]:

	Sex	PassengerId
0	female	314
1	male	577

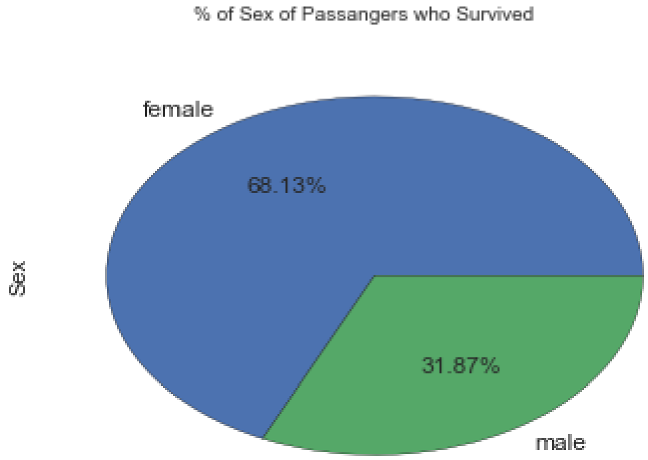


In [844]:

```
madeit_df.groupby(['Sex'], as_index=True)['Sex'].count() \
.plot.pie(subplots=True, title='% of Sex of Passangers who Survived', autopct='%.2f%%', fontsize=12)
madeit_df.groupby(['Sex'], as_index=False)['PassengerId'].count()
```

Out[844]:

	Sex	PassengerId
0	female	233
1	male	109

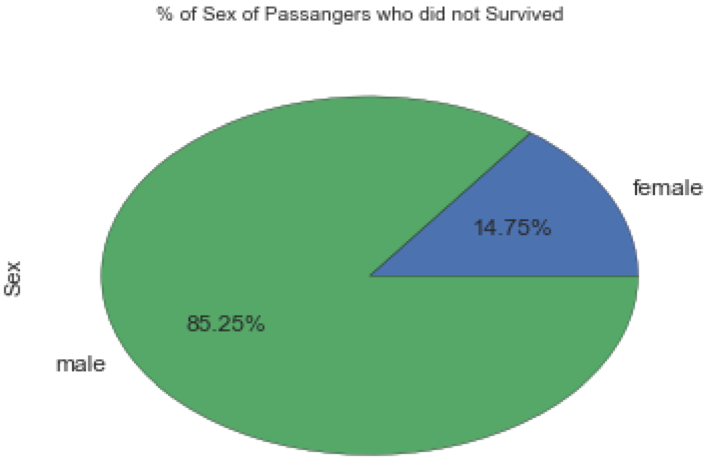


In [845]:

```
didnt_df.groupby(['Sex'], as_index=True)['Sex'].count() \
.plot.pie(subplots=True, title='% of Sex of Passangers who did not Survived', autopct='%.2f%%', fontsize=12)
didnt_df.groupby(['Sex'], as_index=False)['PassengerId'].count()
```

Out[845]:

	Sex	PassengerId
0	female	81
1	male	468



Did the passanger's Age play any role in their chance of survival?

In [846]:

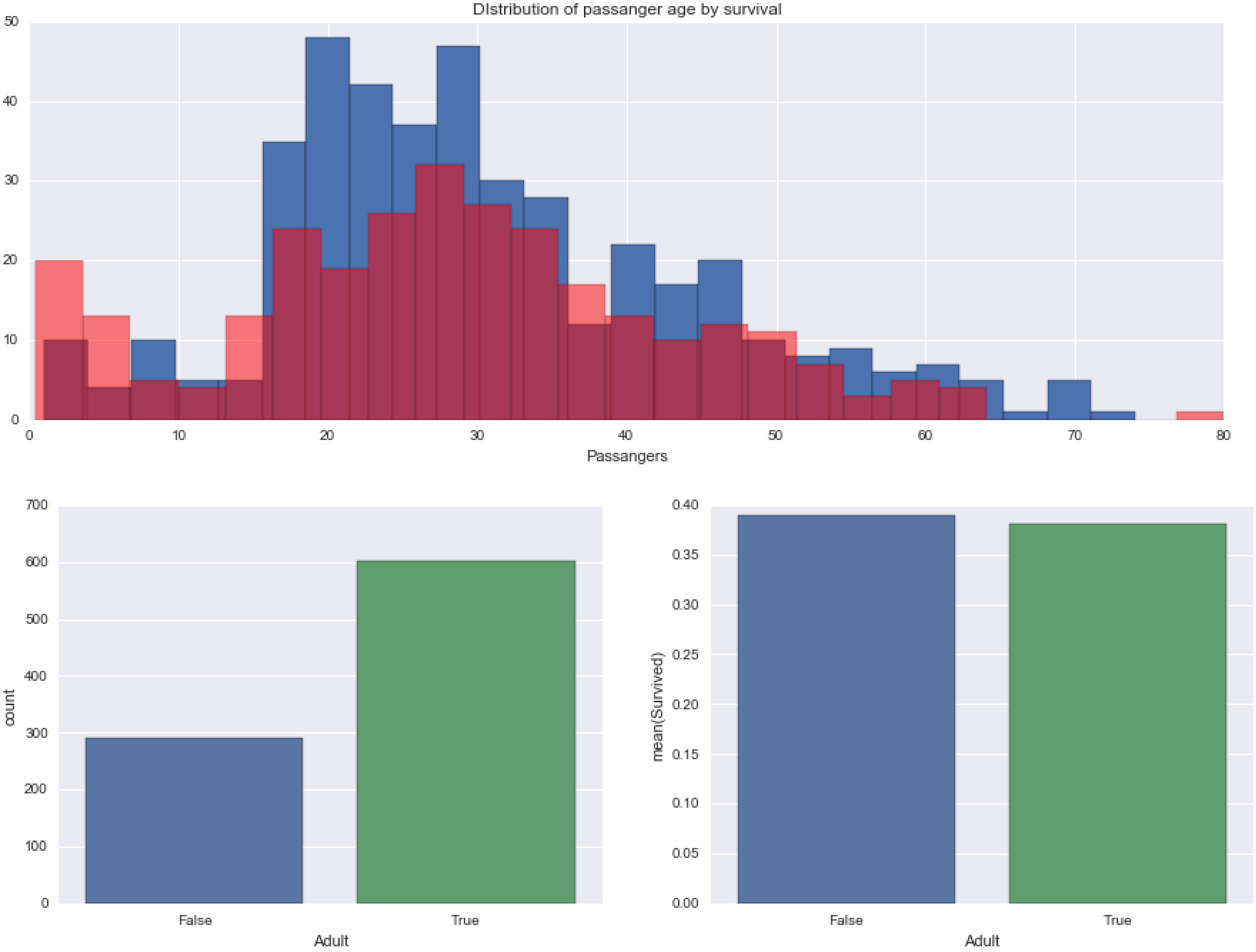
```
# Survival Rate by Age
fig, (axis1) = plt.subplots(1,figsize=(15,5))

didnt_df['Age'].hist(bins=25)
madeit_df['Age'].hist(bins=25, color = 'r', alpha=0.50)
plt.xlabel('Passangers')
plt.title('DIstribution of passanger age by survival')


# Count of Adults / Children
fig, (axis1,axis2) = plt.subplots(1,2,figsize=(15,5))
sns.countplot(x='Adult', data=titanic_df, ax=axis1)

# average of survived by Sex
grouped_by_age = titanic_df[['Adult', 'Survived']].groupby(['Adult'],as_index=False).mean()
sns.barplot(x='Adult', y='Survived', data=grouped_by_age, ax=axis2, order=[False,True])
```

```
Out[846]:  <matplotlib.axes._subplots.AxesSubplot at 0x549ab898>
```


DIstribution of passanger age by survival
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
Passangers
700
600
500
400
300
200
100
0
count
False
True
Adult
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
mean(Survived)
False
True
Adult

Did the amount payed by the passenger for their ticket play any role in their chance of survival?

In [847]:

```
# Survival Rate by Fair
titanic_df.groupby(['SurvivedTF'], as_index=False)['Fare'].mean()
```

Out[847]:

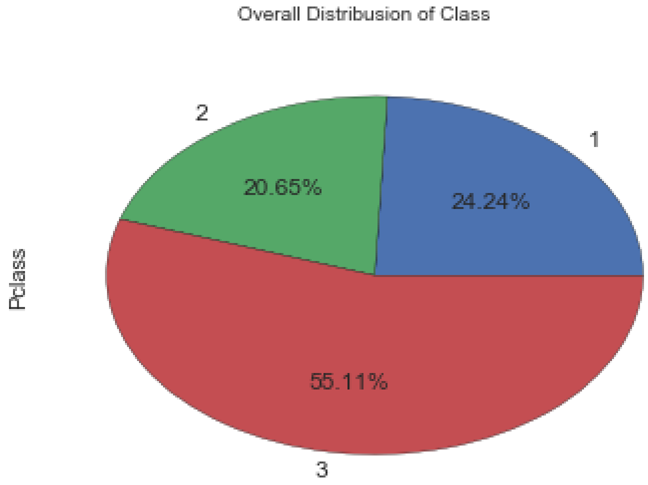
	SurvivedTF	Fare
0	False	22.117887
1	True	48.395408

In [848]:

```
titanic_df.groupby(['Pclass'], as_index=True)['Pclass'].count() \
.plot.pie(subplots=True, title='Overall Distribusion of Class', autopct='%.2f%%', fontsize=12)
titanic_df.groupby(['Pclass'], as_index=False)['PassengerId'].count()
```

Out[848]:

	Pclass	PassengerId
0	1	216
1	2	184
2	3	491

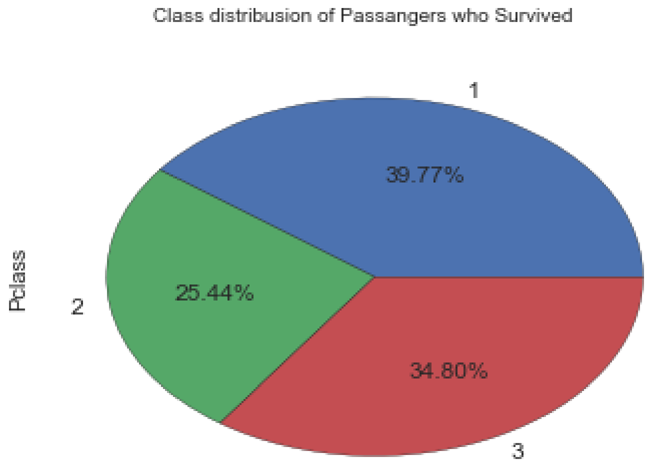


In [849]:

```
madeit_df.groupby(['Pclass'], as_index=True)['Pclass'].count() \
.plot.pie(subplots=True, title='Class distribusion of Passangers who Survived', autopct='%.2f%%', fontsize=12)
madeit_df.groupby(['Pclass'], as_index=False)['PassengerId'].count()
```

Out[849]:

	Pclass	PassengerId
0	1	136
1	2	87
2	3	119

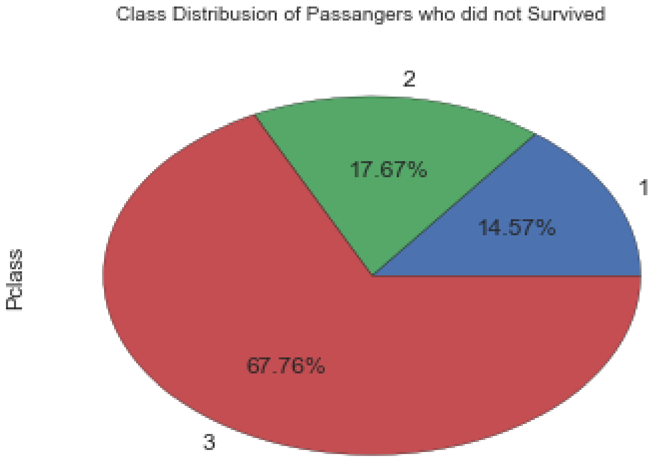


In [850]:

```
didnt_df.groupby(['Pclass'], as_index=True)['Pclass'].count() \
.plot.pie(subplots=True, title='Class Distribusion of Passangers who did not Survived', autopct='%.2f%%', fontsize=12)
didnt_df.groupby(['Pclass'], as_index=False)['PassengerId'].count()
```

Out[850]:

	Pclass	PassengerId
0	1	80
1	2	97
2	3	372

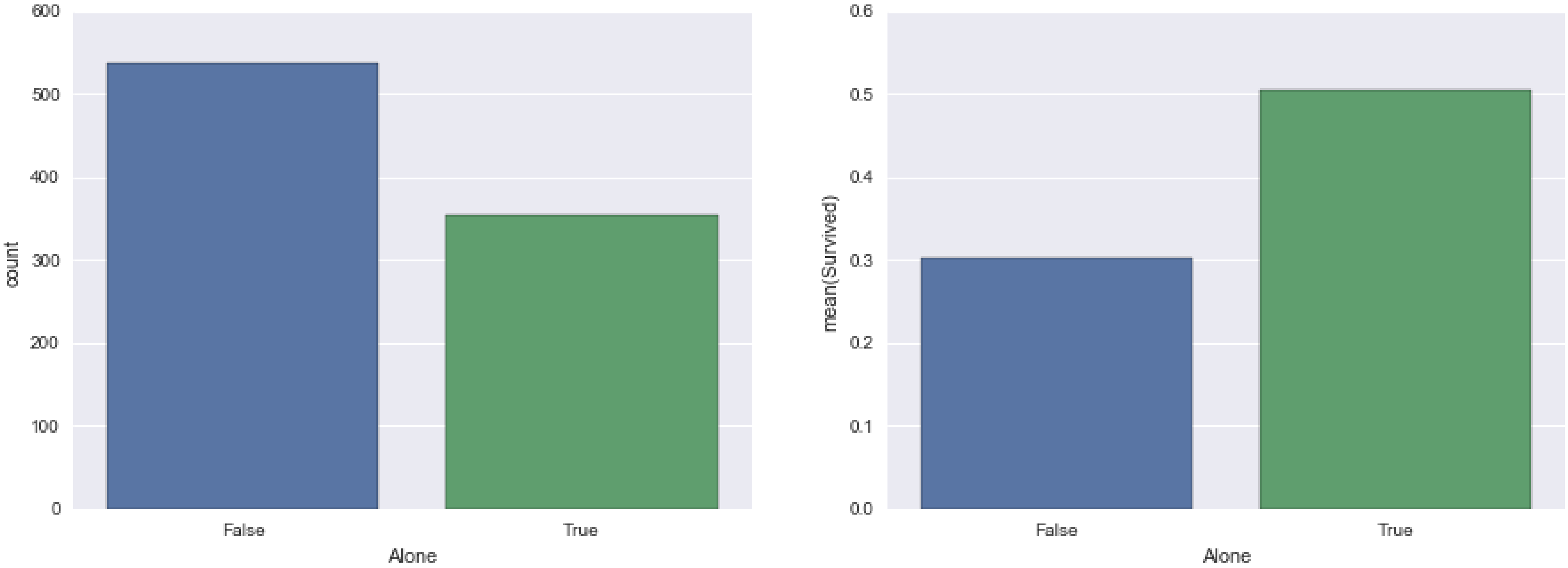


Do passangers with any family memebers aboard have a higher survival rate?

```
In [851]: # Count of passangers without family memebers on board
          fig, (axis1,axis2) = plt.subplots(1,2,figsize=(15,5))
          sns.countplot(x='Alone', data=titanic_df, ax=axis1)

          # Survival rate of passangers without family members
          grouped_by_age = titanic_df[['Alone', 'Survived']].groupby(['Alone'],as_index=False).mean()
          sns.barplot(x='Alone', y='Survived', data=grouped_by_age, ax=axis2, order=[False,True])
```

```
Out[851]: <matplotlib.axes._subplots.AxesSubplot at 0x552f5c18>
```



```
In [ ]:
```