Basic stats

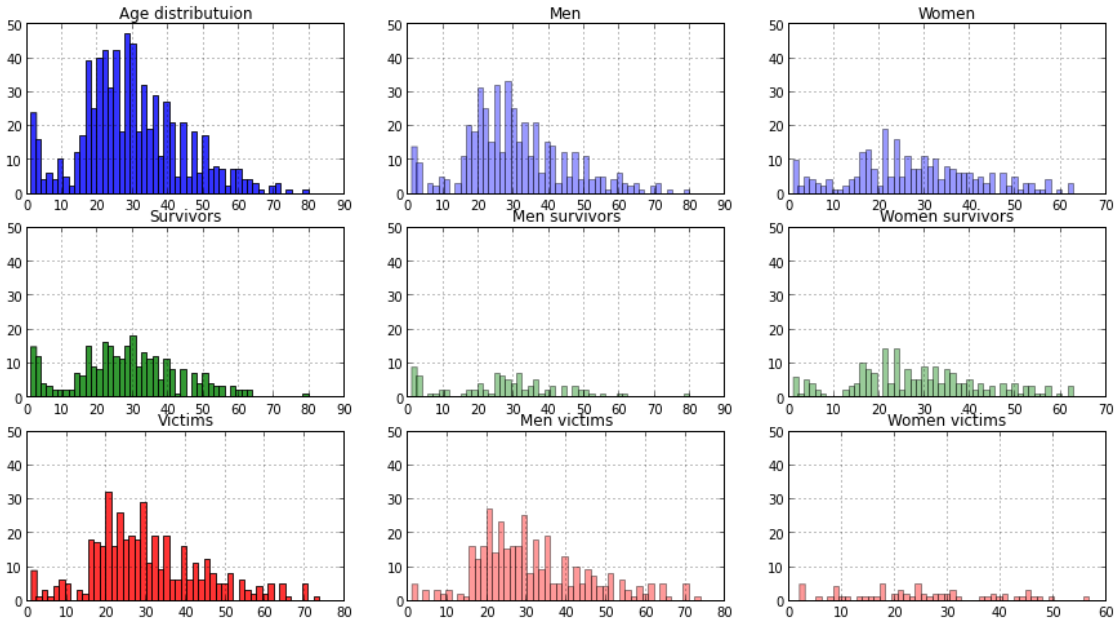
January 29, 2016

In [297]:
```
import pandas as pd
import math
import re

data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')

def mungling(data):
    data.columns = data.columns.map(lambda n: n.lower())
    data.age = data.age.map(math.ceil)

    data['alone'] = data.parch + data.sibsp
    data.alone = data.alone.map(lambda v: 0 if v > 0 else 1)

    data['sex_num'] = data.sex.replace({'male': 0, 'female': 1})
    data['title'] = data.name.map(lambda n: re.search('\w+\.', n).group().lower()[:-1])

    return data

data = mungling(data)
test_data = mungling(test_data)
```

0.0.1 Age

In [298]:
```
fig = plt.figure(figsize(15, 8))

bins = 50

def plot_hist(data, title, idx, *args, **kwargs):
    ax = plt.subplot(3, 3, idx)
    plt.title(title)
    data.hist(bins=bins, *args, **kwargs)
    ax.set_ylim((0, 50))

surv_ind = data.survived == 1
men_ind = data.sex == 'male'
women_ind = data.sex == 'female'

plot_hist(data.age, 'Age distributuion', 1, alpha=0.8)
plot_hist(data[data.sex == 'male'].age, 'Men', 2, alpha=0.4)
plot_hist(data[data.sex == 'female'].age, 'Women', 3, alpha=0.4)

plot_hist(data[surv_ind].age, 'Survivors', 4, alpha=0.8, color='g')
plot_hist(data[surv_ind & men_ind].age, 'Men survivors', 5,
```

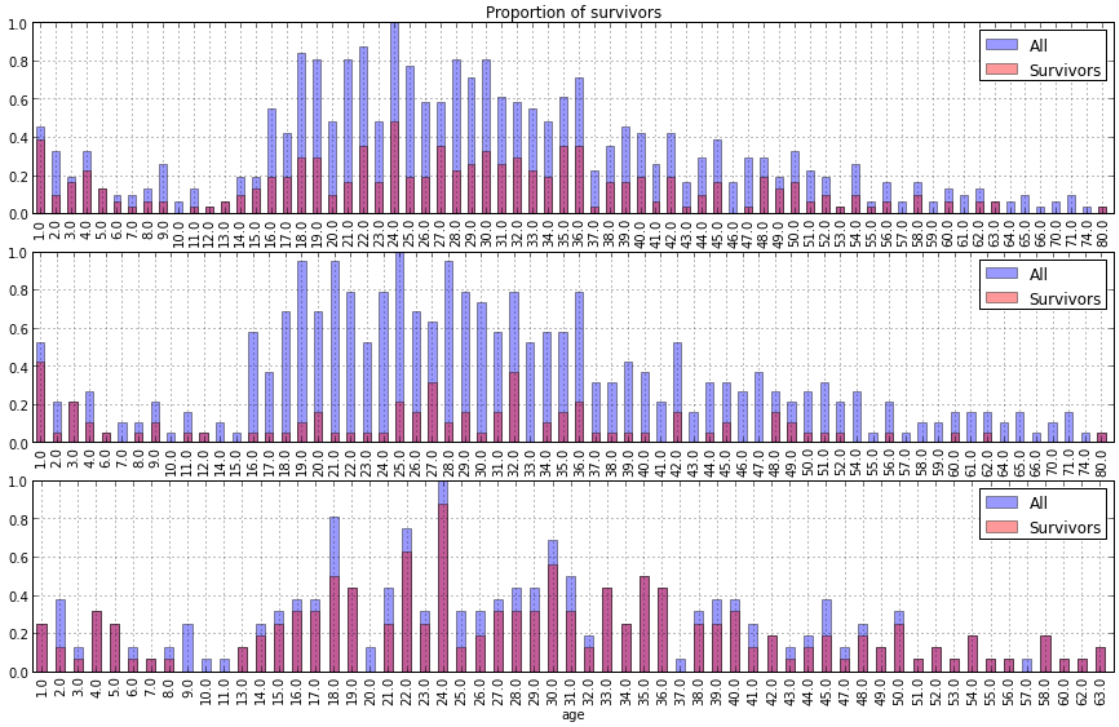
```
        alpha=0.4, color='g')
    plot_hist(data[surv_ind & women_ind].age, 'Women survivors', 6,
        alpha=0.4, color='g')

    plot_hist(data[surv_ind == False].age, 'Victims', 7,
        alpha=0.8, color='r')
    plot_hist(data[(surv_ind == False) & men_ind].age,
        'Men victims', 8, alpha=0.4, color='r')
    plot_hist(data[(surv_ind == False) & women_ind].age,
        'Women victims', 9, alpha=0.4, color='r')
```




```
In [299]: fig = plt.figure(figsize(15, 9))

    def plot_proportion(data):
        age_grp = data.groupby('age').age.size()
        (age_grp / age_grp.max().astype(float)).plot(kind='bar',
            color='b', alpha=0.4, stacked=True,
            label='All')

        grp = data.groupby('age').survived.sum()
        grp = grp / age_grp.max().astype(float)
        grp.plot(kind='bar', color='r', alpha=0.4, stacked=True,
            label='Survivors')

    plt.subplot(3, 1, 1)
    plot_proportion(data)
    plt.title('Proportion of survivors')
    plt.legend(loc='best')

    plt.subplot(3, 1, 2)
    plot_proportion(data[men_ind])
```

```
plt.legend(loc='best')

plt.subplot(3, 1, 3)
plot_proportion(data[women_ind])
plt.legend(loc='best')
```

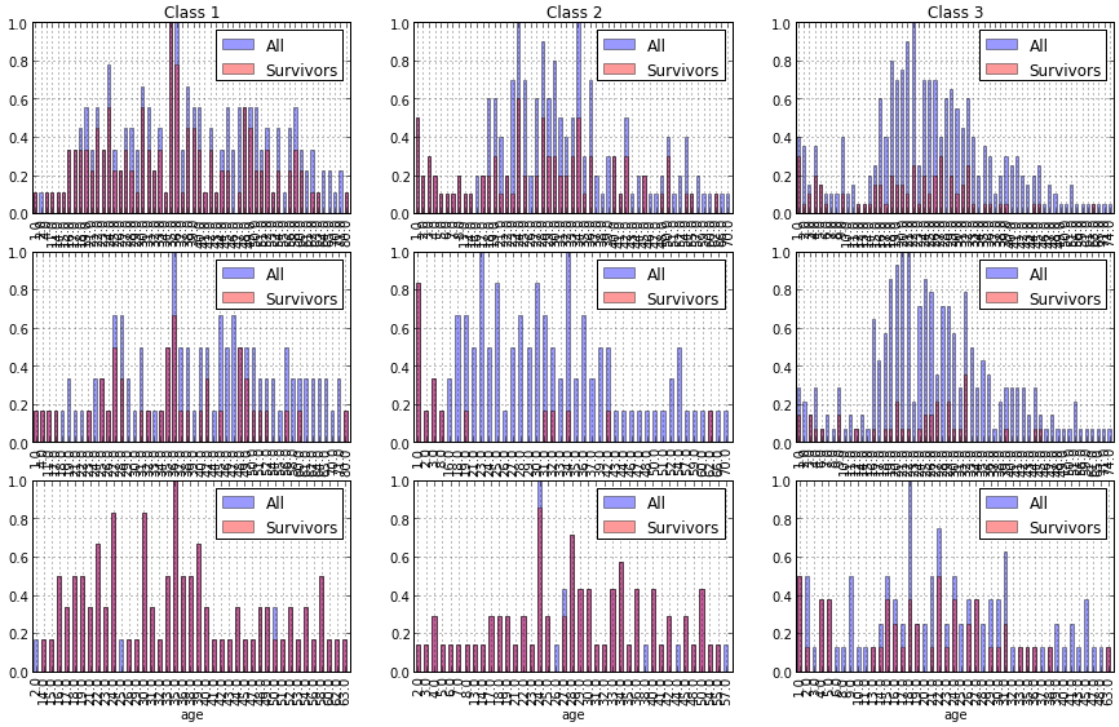
Out[299]: <matplotlib.legend.Legend at 0x357ba350>



In [300]:
```
plt.subplot(3, 3, 1)
plot_proportion(data[data.pclass==1])
plt.title('Class 1')
plt.legend(loc='best')

plt.subplot(3, 3, 2)
plot_proportion(data[data.pclass==2])
plt.title('Class 2')
plt.legend(loc='best')

plt.subplot(3, 3, 3)
plot_proportion(data[data.pclass==3])
plt.title('Class 3')
plt.legend(loc='best')

plt.subplot(3, 3, 4)
plot_proportion(data[(data.pclass==1) & men_ind])
plt.legend(loc='best')

plt.subplot(3, 3, 5)
```

```
plot_proportion(data[(data.pclass==2) & men_ind])
plt.legend(loc='best')

plt.subplot(3, 3, 6)
plot_proportion(data[(data.pclass==3) & men_ind])
plt.legend(loc='best')

plt.subplot(3, 3, 7)
plot_proportion(data[(data.pclass==1) & women_ind])
plt.legend(loc='best')

plt.subplot(3, 3, 8)
plot_proportion(data[(data.pclass==2) & women_ind])
plt.legend(loc='best')

plt.subplot(3, 3, 9)
plot_proportion(data[(data.pclass==3) & women_ind])
plt.legend(loc='best')
```

```
Out[300]: <matplotlib.legend.Legend at 0x36f35890>
```



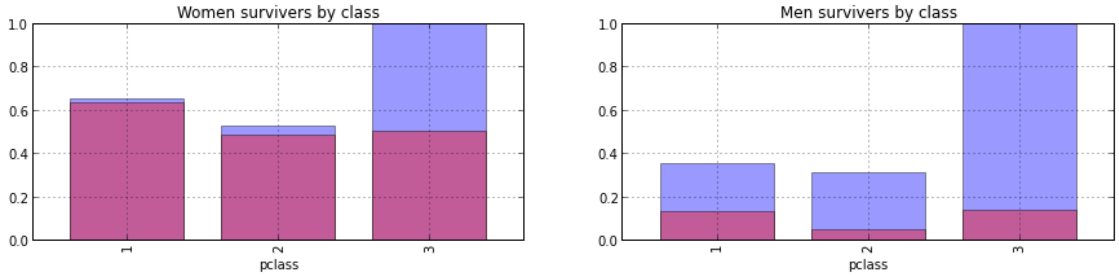
Women from first and second class survived regardless of the age. Yong men from second class dies almost all.

0.0.2 Sex

```
In [301]: plt.figure(figsize(15, 3))

          plt.subplot(1, 2, 1)
```

```
plt.title('Women survivers by class')
grp = data[women_ind].groupby('pclass').survived.size()
(grp / grp.max().astype(float)).plot(kind='bar', alpha=0.4)

surv_grp = data[women_ind].groupby('pclass').survived.sum()
(surv_grp / grp.max().astype(float)).plot(kind='bar', color='r', alpha=0.4)

plt.subplot(1, 2, 2)
plt.title('Men survivers by class')
grp = data[men_ind].groupby('pclass').survived.size()
(grp / grp.max().astype(float)).plot(kind='bar', alpha=0.4)

surv_grp = data[men_ind].groupby('pclass').survived.sum()
(surv_grp / grp.max().astype(float)).plot(kind='bar', color='r', alpha=0.4)
```

Out[301]: <matplotlib.axes.AxesSubplot at 0x37d0b7d0>



In [302]:
```
skip = ['passengerid', 'pclass', 'sex', 'survived', 'name']

for i in range(1, 3):
    print 'Victims-women from %s class' % i

    selection = data[women_ind & (data.pclass == i) & (data.survived == 0)]
    print selection[[c for c in data.columns if c not in skip]]
    print

selection = data[women_ind & (data.pclass == 2) & (data.survived == 1)]

print 'Survivors from second class without cabin information:',\
    selection[selection.cabin.isnull()].cabin.size
print 'Survivors from second class who traveled alone:',\
    selection[(selection.parch == 0) & (selection.sibsp == 0)].cabin.size
print 'Survivors from second class payed in average:',\
    selection[selection.cabin.isnull()].fare.mean()
```

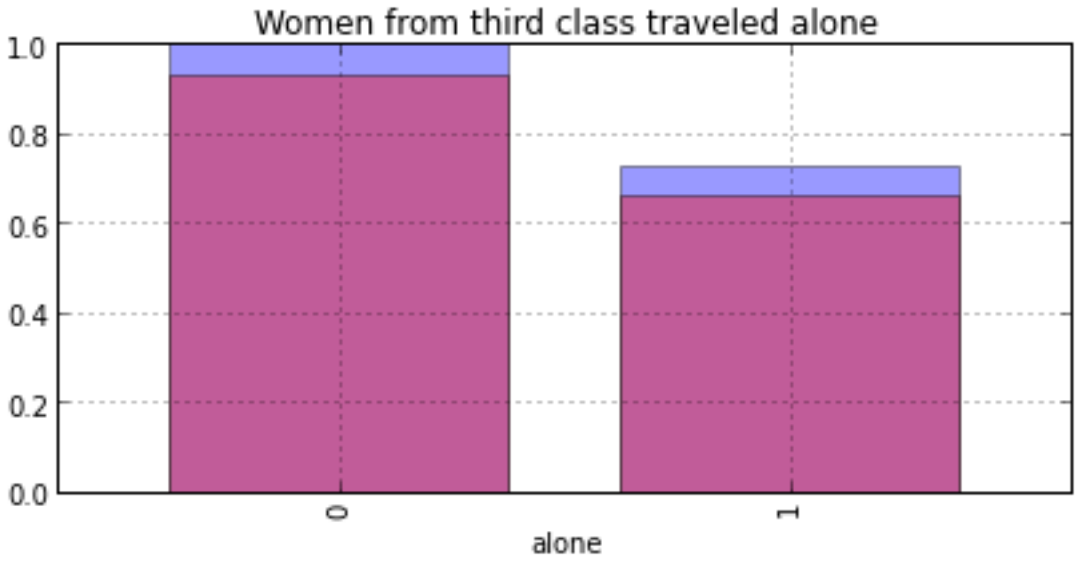
```
Victims-women from 1 class
     age  sibsp  parch    ticket      fare    cabin embarked  alone sex_num title
177   50      0      0  PC 17595   28.7125      C49        C      1       1  miss
297    2      1      2    113781  151.5500  C22 C26        S      0       1  miss
498   25      1      2    113781  151.5500  C22 C26        S      0       1   mrs

Victims-women from 2 class
```

```
     age  sibsp  parch       ticket  fare cabin embarked  alone sex_num title
41    27      1      0        11668  21.0   NaN        S      0       1   mrs
199   24      0      0       248747  13.0   NaN        S      1       1  miss
312   26      1      1       250651  26.0   NaN        S      0       1   mrs
357   38      0      0       237671  13.0   NaN        S      1       1  miss
772   57      0      0  S.O./P.P. 3  10.5   E77        S      1       1   mrs
854   44      1      0       244252  26.0   NaN        S      0       1   mrs

Survivors from second class without cabin information: 61
Survivors from second class who traveled alone: 29
Survivors from second class payed in average: 23.3538934426
```

There is not enough data to say more about women from first and second class.

In [303]:
```
plt.subplot(1, 2, 1)

selection = data[women_ind & (data.pclass == 2)]

grp = selection.groupby('alone')
size_grp = grp.survived.size()

(size_grp / size_grp.max().astype(float)).plot(kind='bar', alpha=0.4)
(grp.survived.sum() / size_grp.max().astype(float)).\
    plot(kind='bar', color='r', alpha=0.4)

plt.title('Women from third class traveled alone')
```

Out[303]:
```
<matplotlib.text.Text at 0x37552b10>
```



In [304]:
```
selection = data[women_ind & (data.pclass < 3)]

print 'Women from 1st and 2nd class are %.2f%% of all the data' % (
    selection.shape[0] / float(data.shape[0]) * 100.)
```

```
print 'Probability to survive for woman from 1st or 2nd class:',\
    float(selection[selection.survived == 1].shape[0]) / selection.shape[0]

selection = data[women_ind & (data.pclass == 3)]

print 'Probability to survive for woman from 3rd class:',\
    float(selection[selection.survived == 1].shape[0]) / selection.shape[0]
```

```
Women from 1st and 2nd class are 19.08% of all the data
Probability to survive for woman from 1st or 2nd class: 0.947058823529
Probability to survive for woman from 3rd class: 0.5
```

The probabilities can be used to check a prediction algorithm.

```
In [320]: from sklearn import ensemble
          from sklearn import cross_validation

          clf = ensemble.RandomForestClassifier(n_estimators=100, random_state=1)

          def predict(clf, fields, train_data, test_data, sex='female'):
              clf.fit(train_data[fields], train_data.survived)

              # for some reason RandomForest returns ndarray and SVC
              # returns Series
              prediction = np.array(clf.predict(test_data[fields]))

              ind = (test_data.sex == sex) & (test_data.pclass < 3)
              print float(prediction[ind].sum()) / prediction[ind].size

              ind = (test_data.sex == sex) & (test_data.pclass == 3)
              print float(prediction[ind].sum()) / prediction[ind].size

              scores = cross_validation.cross_val_score(
                  clf, train_data[fields], train_data.survived, cv=5)

              print 'Accuracy: %0.2f (+/- %0.2f)' % (scores.mean(), scores.std() * 2)

          predict(clf, ['sex_num', 'pclass'], data, test_data)
```

```
1.0
0.0
Accuracy: 0.78 (+/- 0.05)
```

Accorting to our prediction survived 100% women from first and second class and 0% from third class which is in contradiction with our assumptions. Which makes sense because there is no other features to distinquish between potentioal surviver and victims within a group (women, class). So we need some more features.

```
In [306]: from sklearn.svm import SVC

          # the same but using support vector classifier
          clf = SVC(C=1., gamma=0.1)
          predict(clf, ['sex_num', 'pclass'], data, test_data)
```

```
1.0
1.0
Accuracy: 0.79 (+/- 0.03)
```

```
In [307]: from sklearn.naive_bayes import GaussianNB

          clf = GaussianNB()
          predict(clf, ['sex_num', 'pclass'], data, test_data)
```

```
1.0
1.0
Accuracy: 0.79 (+/- 0.03)
```

```
In [308]: data['age_fixed'] = data.age
          data.age_fixed.fillna(data.age.mean(), inplace=True)
          data['fare_fixed'] = data.fare
          data.fare_fixed.fillna(data[women_ind].fare.mean(), inplace=True)

          test_data['age_fixed'] = test_data.age
          test_data.age_fixed.fillna(test_data.age.mean(), inplace=True)
          test_data['fare_fixed'] = test_data.fare
          test_data.fare_fixed.fillna(test_data.fare.mean(), inplace=True)

          clf = ensemble.RandomForestClassifier(n_estimators=100, random_state=1)
          predict(clf, ['sex_num', 'pclass', 'age_fixed'], data, test_data)
```

```
0.9625
0.361111111111
Accuracy: 0.80 (+/- 0.03)
```

Adding age makes results much more realistic. But still not from perfect.

```
In [312]: third_class = data[women_ind & (data.pclass == 3)]

          third_class.groupby(['title', 'alone']).age.transform(
              lambda grp: grp.fillna(grp.mean()))

          clf = ensemble.RandomForestClassifier(n_estimators=100, random_state=1)
          predict(clf, ['sex_num', 'pclass', 'age_fixed'], data, test_data)
```

```
0.9625
0.361111111111
Accuracy: 0.80 (+/- 0.03)
```

Restoring age with more preciesly doesn't really change anything.

```
In [318]: clf = ensemble.RandomForestClassifier(n_estimators=100, random_state=1)
          predict(clf, ['sex_num', 'pclass', 'age_fixed', 'alone', 'fare'], data[women_ind],
              test_data[test_data.sex == 'female'])
```

```
1.0
0.513888888889
Accuracy: 0.81 (+/- 0.09)
```

This gives us the best result.

0.0.3 Sex (men)

```
In [344]: clf = ensemble.RandomForestClassifier(n_estimators=100, random_state=1)
          predict(clf, ['sex_num', 'pclass', 'age_fixed', 'alone', 'fare_fixed'], data[men_ind],
              test_data[test_data.sex == 'male'], 'male')
```

```
0.125
0.116438356164
Accuracy: 0.82 (+/- 0.04)
```

```
In [348]: selection = data[men_ind & (data.pclass < 3)]

          print 'Men from 1st and 2nd class are %.2f%% of all the data' % (
              selection.shape[0] / float(data.shape[0]) * 100.)

          print 'Probability to survive for men from 1st or 2nd class:',\
              float(selection[selection.survived == 1].shape[0]) / selection.shape[0]

          selection = data[men_ind & (data.pclass == 1)]
          print 'Probability to survive for men from 1st class:',\
              float(selection[selection.survived == 1].shape[0]) / selection.shape[0]

          selection = data[men_ind & (data.pclass == 2)]
          print 'Probability to survive for men from 2nd class:',\
              float(selection[selection.survived == 1].shape[0]) / selection.shape[0]

          selection = data[men_ind & (data.pclass == 3)]
          print 'Probability to survive for men from 3rd class:',\
              float(selection[selection.survived == 1].shape[0]) / selection.shape[0]
```

```
Men from 1st and 2nd class are 25.81% of all the data
Probability to survive for men from 1st or 2nd class: 0.269565217391
Probability to survive for men from 1st class: 0.368852459016
Probability to survive for men from 2nd class: 0.157407407407
Probability to survive for men from 3rd class: 0.135446685879
```