Python and pdf introduction 2-4

February 4, 2016

In [35]:
```
%matplotlib inline

import numpy as np #matrices and data structures
import scipy.stats as ss #standard statistical operations
import pandas as pd #keeps data organized, works well with data
import matplotlib.pyplot as plt #plot visualization
```

0.0.1 Find the weather data here: Januaries in Central Park NYC

In [4]:
```
#read a csv
nyw = pd.read_csv('NYC-CParkWeather.csv')
nyw = nyw.set_index('year') #represents observations
nyw.head()
```

Out[4]:
```
      low  high  warm_min  cold_max  avg_min  avg_max  mean precip  snowfall  \
year
2016   11    59        42        27     28.2     40.8  34.5   4.41      27.2
2015    8    56        41        21     23.7     36.1  29.9   5.23      16.9
2014    4    58        44        17     21.9     35.5  28.7   2.79      19.7
2013   11    61        43        20     29.3     40.8  35.1   2.76       1.5
2012   13    62        46        27     30.4     44.2  37.3   3.23       4.3

     max24precip  max24snow
year
2016        2.31       26.6
2015         2.1        5.5
2014         0.5       11.0
2013         0.9        1.5
2012        1.38        4.3
```

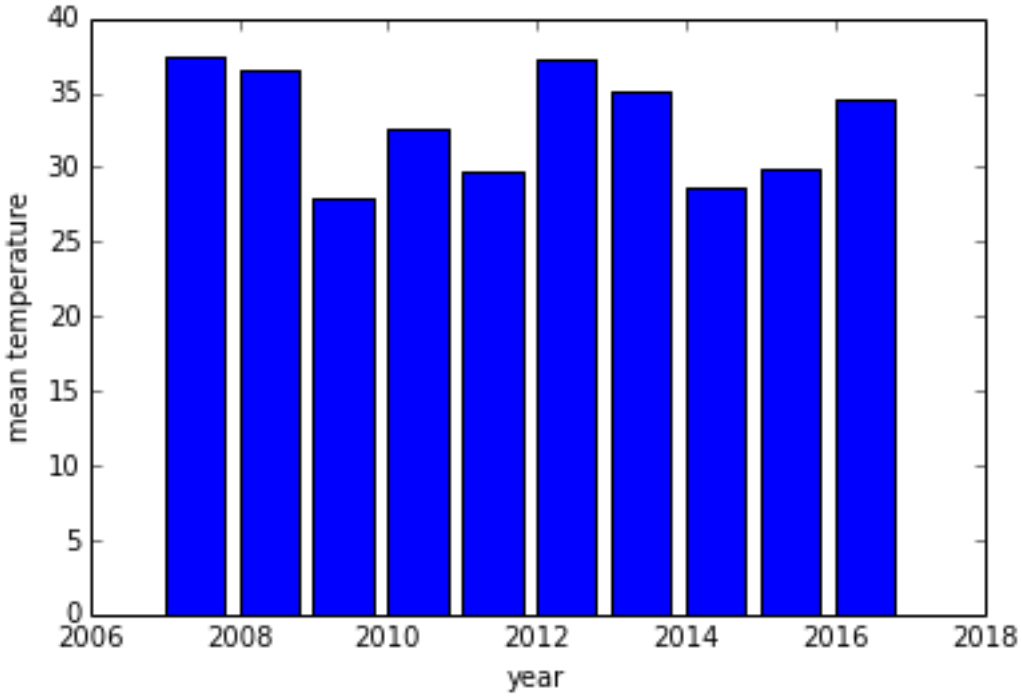
In [6]:
```
nyw.describe()
```

Out[6]:
```
              low        high    warm_min    cold_max     avg_min     avg_max  \
count  117.000000  117.000000  117.000000  117.000000  117.000000  117.000000
mean     8.299145   57.521368   42.743590   21.111111   26.070085   38.656410
std      6.407518    6.010907    5.623464    5.686914    4.531104    4.538008
min     -5.000000   44.000000   28.000000    8.000000   15.600000   27.700000
25%      5.000000   54.000000   39.000000   17.000000   23.500000   35.600000
50%      8.000000   57.000000   42.000000   21.000000   26.100000   38.600000
75%     12.000000   62.000000   46.000000   25.000000   28.900000   41.800000
max     25.000000   72.000000   59.000000   33.000000   37.700000   48.600000

             mean    snowfall   max24snow
count  117.000000  117.000000  117.000000
```

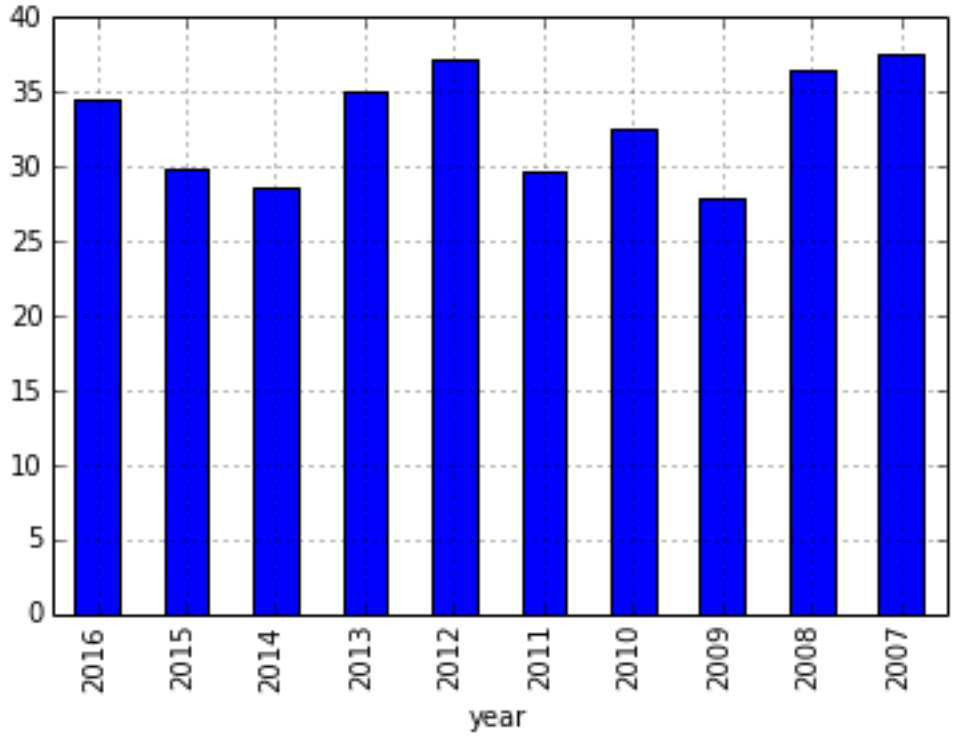
```
mean    32.365812    7.589744    3.996581
std      4.494521    7.270872    3.859246
min     21.700000    0.000000    0.000000
25%     29.500000    2.000000    1.300000
50%     32.300000    5.500000    3.000000
75%     35.200000   11.400000    5.600000
max     43.200000   36.000000   26.600000
```

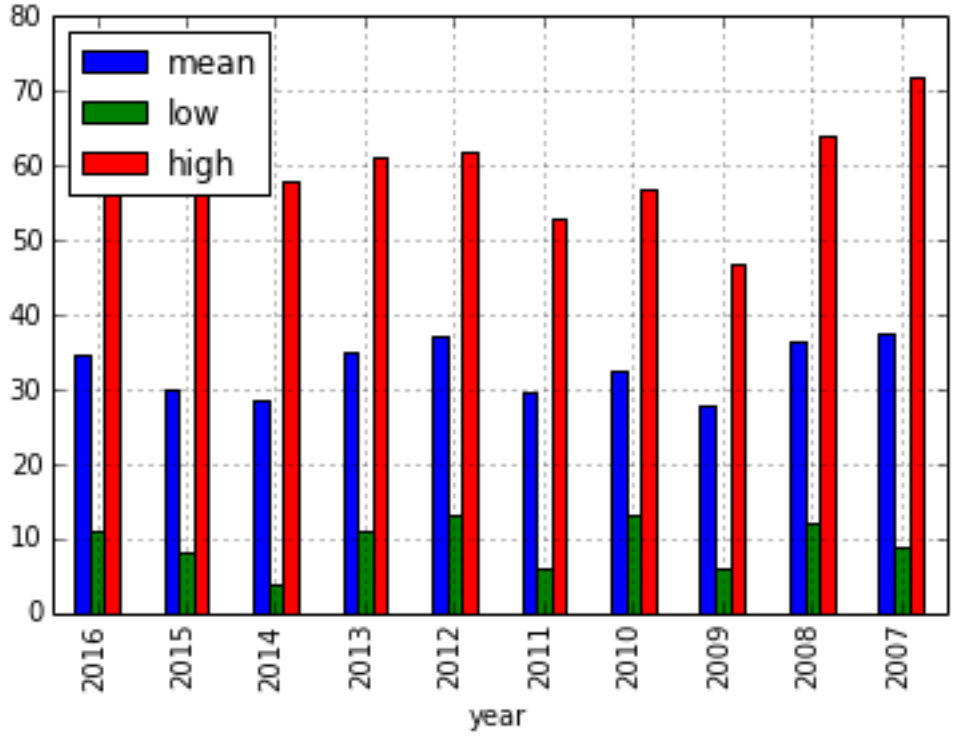
In [13]:
```
#plot some data
nyw_lastten = nyw[:10]
plt.bar(nyw_lastten.index,nyw_lastten['mean'])
plt.ylabel('mean temperature')
plt.xlabel('year')
plt.show()
```



In [16]:
```
#directly plotting with pandas wrapper
nyw_lastten['mean'].plot(kind='bar')
nyw_lastten[['mean', 'low', 'high']].plot(kind='bar')
```

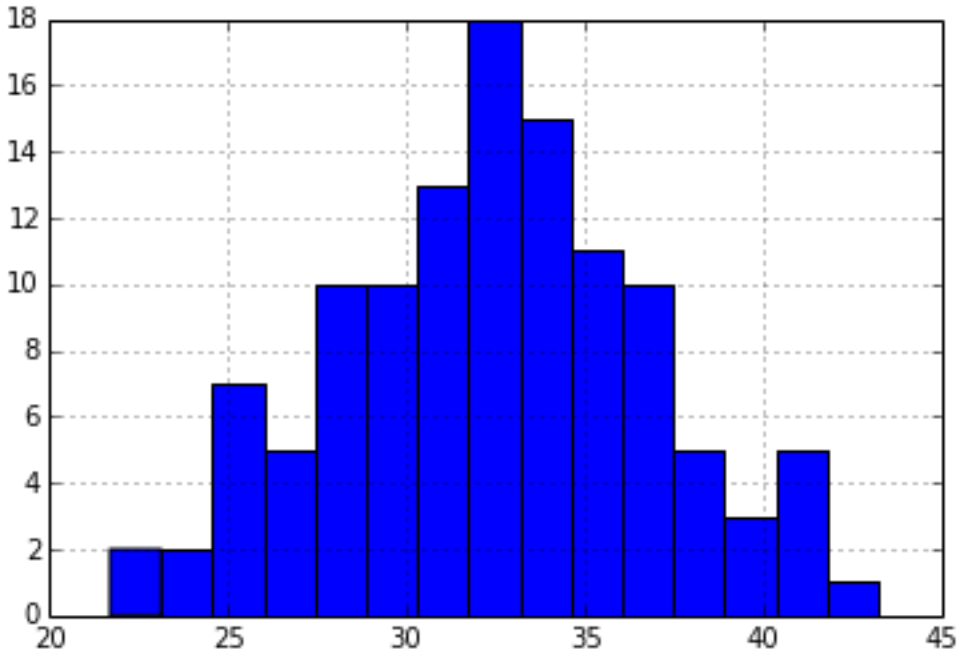
Out[16]:
```
<matplotlib.axes.AxesSubplot at 0x7f17a489b590>
```


40
35
30
25
20
15
10
5
0
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
year


80
70
60
50
40
30
20
10
0
mean
low
high
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
year

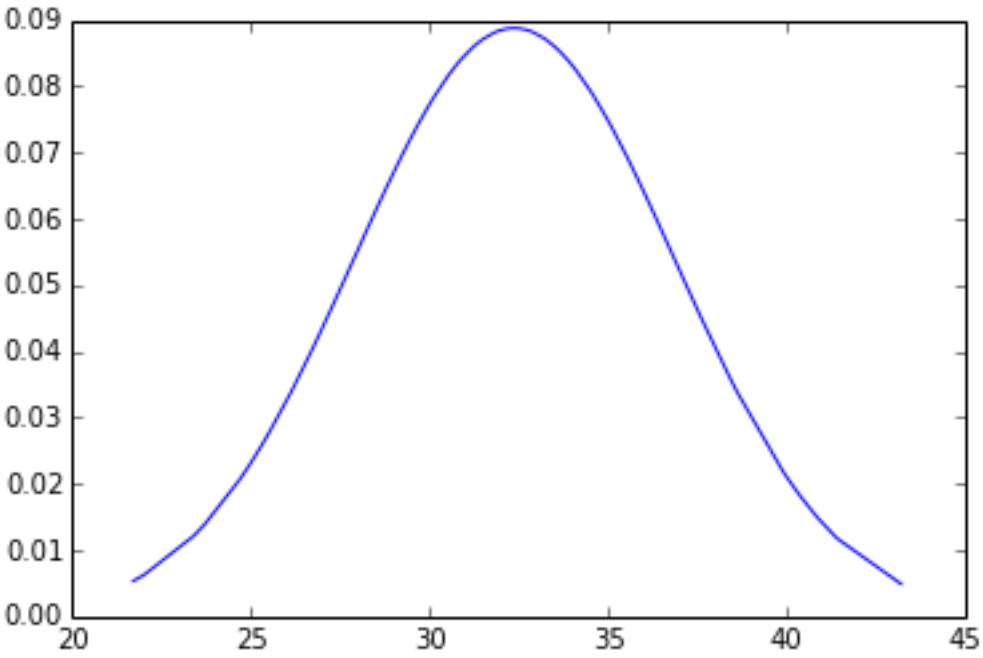
In [37]:
```
#plot a histogram
nyw_avgtemp = nyw['mean']
nyw_avgtemp.hist(bins=15)
#hint: try other bin sizes!
nyw_avgtemp.min(), nyw_avgtemp.max()
```

Out[37]: (21.699999999999999, 43.200000000000003)



In [38]:
```
#plot a normal distribution with parameters fit to the data:
fit = ss.norm.pdf(sorted(nyw_avgtemp), nyw_avgtemp.mean(), nyw_avgtemp.std())
plt.plot(sorted(nyw_avgtemp), fit)
```

Out[38]: [<matplotlib.lines.Line2D at 0x7f17a3f0c6d0>]

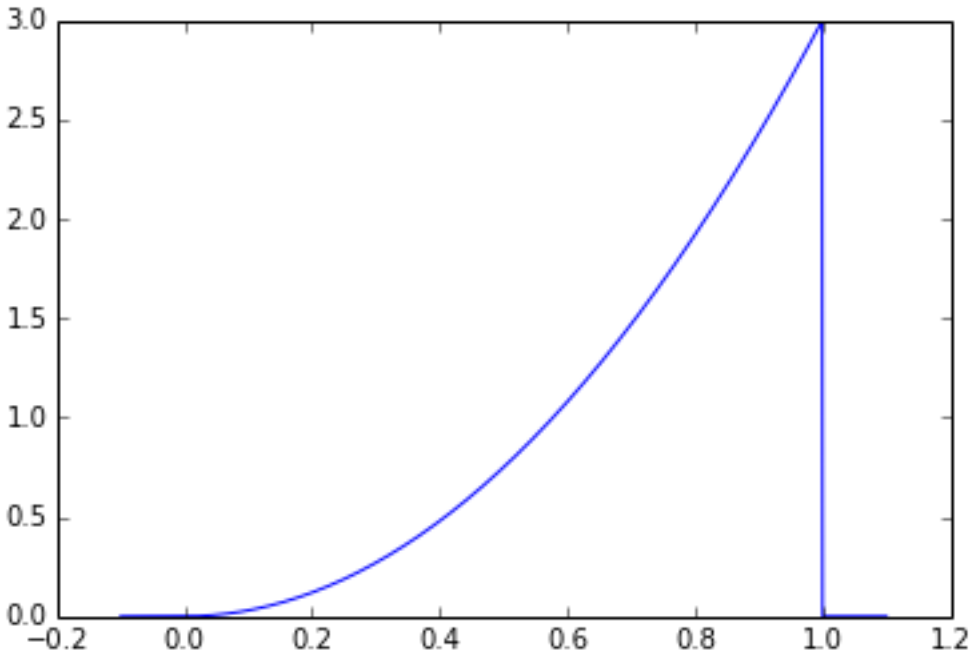


In [46]:
```
#create a custom distribution (based on x-squared): class definition
class sq_dist(ss.rv_continuous):
    def _pdf(self,x):
        return 3*x**2.0

#create an instance of the random variable:
sq_crv = sq_dist(a=0.0, b=1.0, name="sq_pdf") #a to b is the range

#set a range to plot
#x = np.arange(0.0, 1.0, .05)
x = np.arange(-0.1, 1.1, .001) #to see the edges of the distribution, use this
plt.plot(x, sq_crv.pdf(x))
```

Out[46]:
```
[<matplotlib.lines.Line2D at 0x7f17a3b8ae50>]
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2