Homework 2 - Corrected

Andrea Yocom

April 22, 2014

Contents

1 Instructions

- np.genfromtxt ended up being useful
- not all functions should be verbose.
- how to write out to file (thanks Jordan)
- how to use ipython notebook

About the ipython notebook. It has strengths and weaknesses:

Strengths:

- easy to test things out quickly
- looks nice when saved to pdf/html

Weaknesses:

- difficult to share/reuse code
- difficult to save to pdf

2 Code

```
import numpy as np
import scipy as sp
data = np.genfromtxt('./data/hw2.csv', delimiter = ',' , usecols = (2,3))

import matplotlib
matplotlib.use('PS')
import matplotlib.pyplot as plt

# Verbosely return std dev of the temperature given a range of years
def annual_temp_stdev(data,begin,end):
    """data must be a 2-col array of year and temp, begin and end are years to define averaging"""
    #check to make sure begin and end are within range
    if begin not in data[:,0]:
        print "Beginning year not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
    elif end not in data[:,0]:
        print "Ending year not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
    else:
        #index the begin and end years
        ibegin = begin - data[0,0]
        iend = end - data[0,0]

        #compute standard deviation of the temperatures in the range indicated by years
        stdev = np.std(data[ibegin:iend+1,1])
        #round
        stdev = round(stdev,3)

        return "Standard deviation for temperatures in years " + str(begin) \
```

```
        + " to " + str(end) + " is " + str(stdev) + " degrees Fahrenheit."

# Computes the standard deviation of the 2nd col given a range of items in the 1st col
def my_stdev(data,begin,end):
    """data must be a 2-col array, begin and end are endpoints to define averaging"""
    #check to make sure begin and end are within range
    if begin not in data[:,0]:
        print "begin not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
    elif end not in data[:,0]:
        print "end not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
    else:
        #index the begin and end points
        ibegin = begin - data[0,0]
        iend = end - data[0,0]

        #compute standard deviation of the data in the range indicated by [begin:end+1]
        stdev = np.std(data[ibegin:iend+1,1])
    return stdev

# Computes the mean of the 2nd col given a range of items in the 1st col
def my_mean(data,begin,end):
    """data must be a 2-col array, begin and end are endpoints to define averaging"""
    #check to make sure begin and end are within range
    if begin not in data[:,0]:
        print "begin not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
    elif end not in data[:,0]:
        print "end not in range " + str(data[0,0]) \
        + " to " + str(data[len(data)-1,0]) + ", try again."
```

```
    else:
        #index the begin and end points
        ibegin = begin - data[0,0]
        iend = end - data[0,0]

        #compute standard deviation of the data in the range indicated by [begin:end+1]
        datamean = np.mean(data[ibegin:iend+1,1])
    return datamean

def bin_plot(data,binsize,begin,end):
    #initialize lists to store binned averages and stdevs
    years = []
    means = []
    stdevs = []

    #loop in increments of binsize to populate lists,
    #up to the last full decade (doesn't work for 2011,2012 in this usn)
    i = 0
    while i < int(round(2012-1850,-1)):
        years.append(begin+i)
        means.append(my_mean(data,begin+i,begin+i+binsize))
        stdevs.append(my_stdev(data,begin+i,begin+i+binsize))
        i += binsize
    return [years,means,stdevs]

plotdata=bin_plot(data,10,1850,2012)

[years,means,stdevs] = plotdata

#Calibrate: report standard deviations for the years 1930 to 1960 and 1980 to 2010.
```

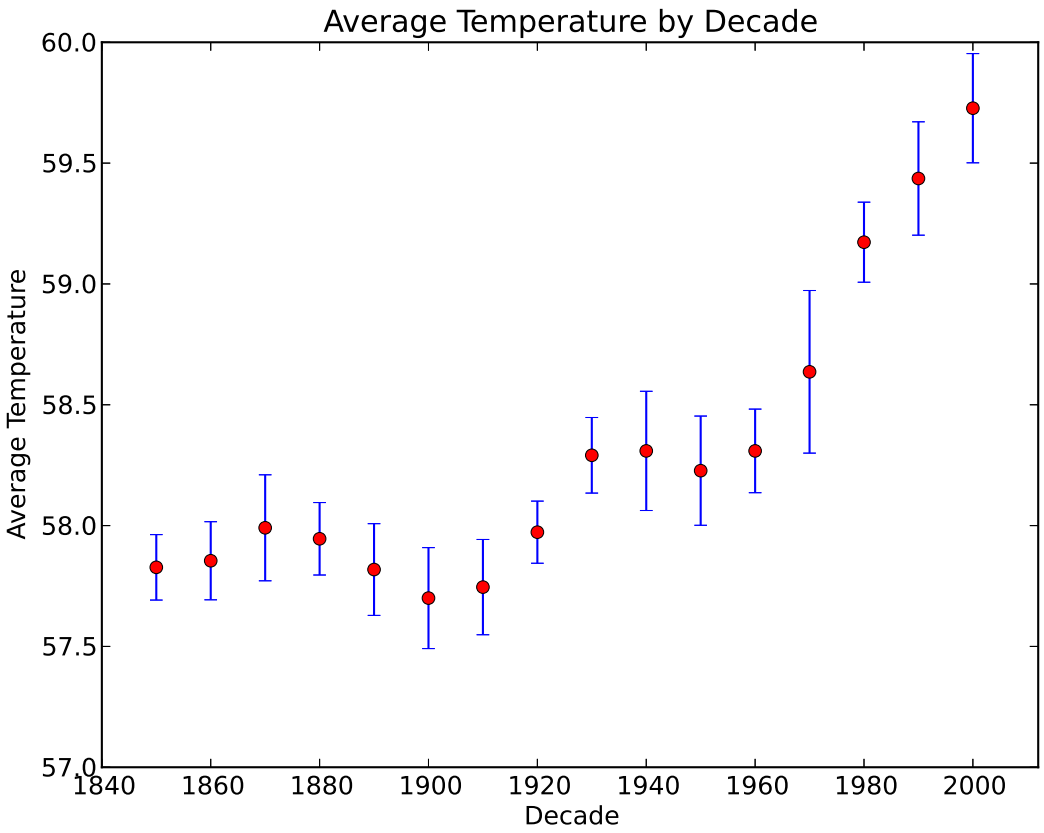
```
thirtyToSixty = annual_temp_stdev(data,1930,1960)
eightyToTen = annual_temp_stdev(data,1980,2010)
#output calibration
with open('./ch/calibration.txt','w') as f:
    # printString =
    f.write(thirtyToSixty + ' ' + eightyToTen)

#plotting code borrowed from Jordan
plt.plot(years,means,'ro')
plt.errorbar(years,means,yerr=stdevs,linestyle = 'None',color='blue')
plt.xlabel('Decade')
plt.ylabel("Average Temperature")
plt.title("Average Temperature by Decade")
plt.xlim([1840,2012])
plt.ylim([57,60])
plt.savefig('./img/temp_plot')
```

3 Output

3.1 Calibration

Standard deviation for temperatures in years 1930 to 1960 is 0.209 degrees Fahrenheit. Standard deviation for temperatures in years 1980 to 2010 is 0.319 degrees Fahrenheit.

3.2 Plot


Average Temperature by Decade
Average Temperature
Decade
60.0
59.5
59.0
58.5
58.0
57.5
57.0
1840
1860
1880
1900
1920
1940
1960
1980
2000