**ENGR 102 Sect 508 Lab 10b**

**110 points**

**Reading assignment:**

|  |  |
| --- | --- |
| **Lecture Slides** | **L10** |
| **zyBook chapter 10** | **Complete all participation and challenge activities** |

*Attention!!*

*Individual submission.*

*Submit* *your Py-files together with your word/pdf file with screenshots of your tests outputs. Include any derivations, comments and supplemental notes in your word/pdf files.*

*No pictures by the phone – it is impossible to read. You will be allowed to resubmit and reupload HW as many times as you want to within the due date/time, only last submission will be graded. No late submissions. For submission you may use this file as template: rename file including your name. Do not forget to put your name inside of this file as well. If it is a team work use Team Header, include the team number and all team members.*

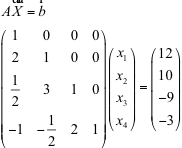
This week’s lab and individual assignments are meant to familiarize you with two of the most commonly used engineering packages in Python, numpy and matplotlib.

This individual assignment is meant to follow on to the team lab work. Be sure you have completed the team lab work, first. You are to individually create two programs, and turn in the two programs.

Question 1

1. [10pts] Solve for X by hand, I need to see all steps

x = 12,-14, 27, -52

* Steps: first we find the inverse of A by:
  + Multiply row 1 by -2 and add to row 2.  
    Multiply row 1 by -1/2 and add to row 3.  
    Multiply row 1 by 1 and add to row 4.
  + Multiply row 2 by -3 and add to row 3.  
    Multiply row 2 by 1/2 and add to row 4.
  + Multiply row 3 by -2 and add to row 4.
  + Divide row 1 by 1.  
    Divide row 2 by 1.  
    Divide row 3 by 1.  
    Divide row 4 by 1.
  + Result: A-1
* The solution of the equation is:  
  A-1B = x

**[35 points] Activity #1: Matrix multiply and plot.**

1. You will create a program that repeatedly multiplies a matrix by a point and plots the data to the screen.

In this case, we will have a 2D point, (x,y). The point can be represented as a vector: . We can also define a 2x2 matrix, . Computing the product of *M* with *v* gives us a new point *v’*: . We can then use *v’* as the new point, multiply by matrix *M* again, and get another point, i.e. . This can go on indefinitely, creating a long sequence of points.

Create a program that uses numpy to create a matrix and a point. Specifically, you should begin with the point (1, 0), and the matrix: . Then, repeatedly multiply the matrix by the point to get a new point. You should repeat this between 150 and 250 times.

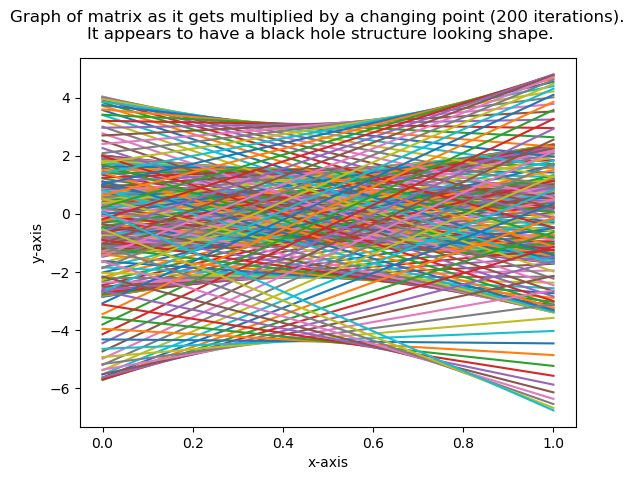
Note: the purpose of this part of the assignment is to get practice with numpy, so you should use numpy for your operations, even if you find it easier to perform this computation a different way.

1. Then, make your program plot the data points, using matplotlib. Be sure to label the axes, and include a title. Your title should give a brief description of the shape that the points “trace” out.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 10B  
# Date: 04-11-2018***import** numpy **as** np  
**import** matplotlib  
matplotlib.use(**"TkAgg"**)  
**from** matplotlib **import** pyplot **as** plt  
  
*# Declare initial point (v) and initial 2D matrix :*v = np.array([[1],[0]])  
  
M = np.array([[1.00583,-0.087156],[0.087156,1.00583]])  
  
*# Operations***for** i **in** range (200):  
 Prod = M@v  
 v = Prod  
 plt.plot(v)  
  
*# Print graph*plt.suptitle(**'Graph of matrix as it gets multiplied by a changing point (200 iterations). \n'  
 'It appears to have a black hole structure looking shape.'**)  
plt.xlabel(**'x-axis'**)  
plt.ylabel(**'y-axis'**)  
plt.show()

Output:

****

**[65 points] Activity #2: Plotting multiple data points**

This program will be a follow-on to the program from last week’s individual assignment, Activity #3. In that assignment, you read in weather data from a file.

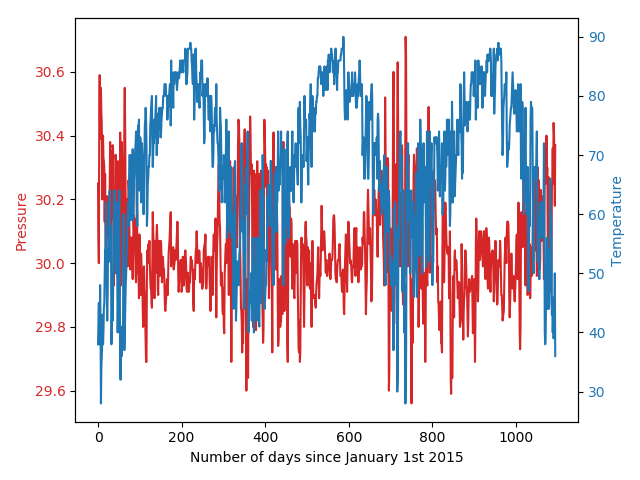
In this assignment, you are to again read in that weather data from the file, but instead of answering statistical queries, you are to plot the data in a set of graphs. You may re-use the code from the previous assignment for reading in data from a file. Using matplotlib, you are to create the following 4 graphs.

1. Create a line graph that shows both average temperature and pressure plotted over the period of time. Both should be plotted on the same graph, with date on the x-axis, and different y axes for the two different measurements.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 10B  
# Date: 04-11-2018***import** numpy **as** np  
**import** matplotlib  
matplotlib.use(**"TkAgg"**)  
**from** matplotlib **import** pyplot **as** plt  
  
file = open(**'WeatherDataMac.csv'**,**'r'**)  
days = 0  
temperature = []  
pressure = []  
day\_list = []  
file.readline() *# skip the first line***for** row **in** file:  
 t = float(row.split(**','**)[2])  
 p = float(row.split(**','**)[11])  
 temperature.append(t)  
 pressure.append(p)  
 day\_list.append(days)  
 days += 1  
  
print(temperature)  
print(pressure)  
print(day\_list)  
  
*# Plotting of graph:*fig, ax1 = plt.subplots()  
  
color = **'tab:red'**ax1.set\_xlabel(**'Number of days since January 1st 2015'**)  
ax1.set\_ylabel(**'Pressure'**, color=color)  
ax1.plot(day\_list, pressure, color=color)  
ax1.tick\_params(axis=**'y'**, labelcolor=color)  
  
ax2 = ax1.twinx() *# instantiate a second axes that shares the same x-axis*color = **'tab:blue'**ax2.set\_ylabel(**'Temperature'**, color=color) *# we already handled the x-label with ax1*ax2.plot(day\_list, temperature, color=color)  
ax2.tick\_params(axis=**'y'**, labelcolor=color)  
  
fig.tight\_layout() *# otherwise the right y-label is slightly clipped*plt.show()

Output:

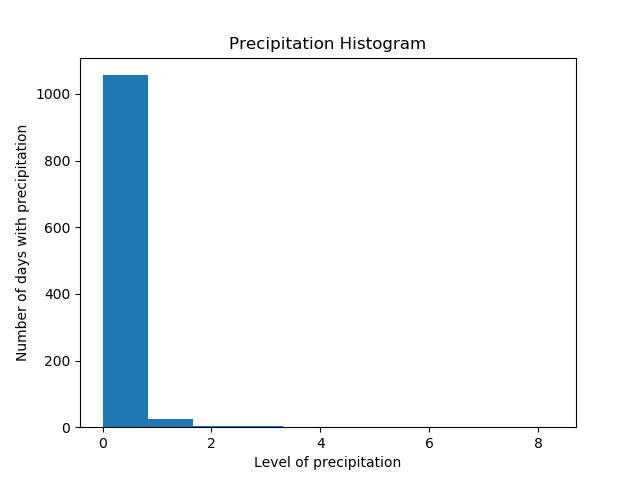


1. Create a histogram showing amounts of precipitation. The x axis should cover a reasonable range of precipitation levels, and the y axis should show number of days that had precipitation in the specific range.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 10B  
# Date: 04-11-2018***import** numpy **as** np  
**import** matplotlib  
matplotlib.use(**"TkAgg"**)  
**from** matplotlib **import** pyplot **as** plt  
  
  
file = open(**'WeatherDataMac.csv'**,**'r'**)  
precipitation = []  
  
file.readline() *# skip the first line***for** row **in** file:  
 num = float(row.split(**','**)[13])  
 precipitation.append(num)  
  
plt.hist(precipitation)  
plt.title(**"Precipitation Histogram"**)  
plt.xlabel(**"Level of precipitation"**)  
plt.ylabel(**"Number of days with precipitation"**)  
plt.show()

Output:

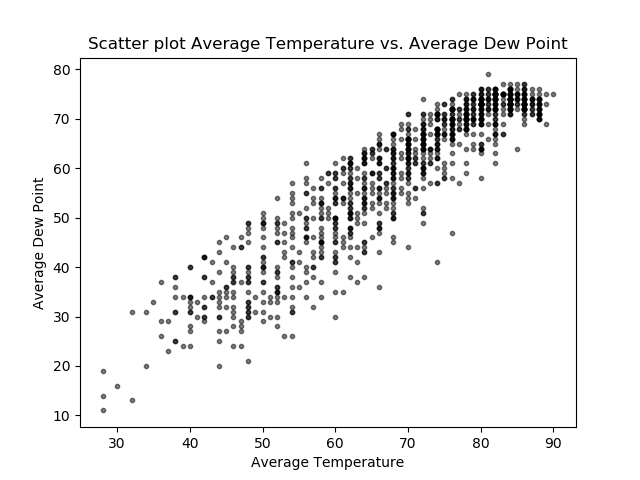


1. Create a scatterplot indicating relationships between average temperature and average dew point (one on each axis).

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 10B  
# Date: 04-11-2018***import** numpy **as** np  
**import** matplotlib  
matplotlib.use(**"TkAgg"**)  
**from** matplotlib **import** pyplot **as** plt  
  
  
file = open(**'WeatherDataMac.csv'**,**'r'**)  
  
average\_temperature = []  
average\_dewpoint = []  
  
file.readline() *# skip the first line***for** row **in** file:  
 t = float(row.split(**','**)[2])  
 d = float(row.split(**','**)[5])  
 average\_temperature.append(t)  
 average\_dewpoint.append(d)  
  
*# Scatterplot:*plt.scatter(average\_temperature, average\_dewpoint, s=(np.pi\*3), c=(0,0,0), alpha=0.5)  
plt.title(**'Scatter plot Average Temperature vs. Average Dew Point'**)  
plt.xlabel(**'Average Temperature'**)  
plt.ylabel(**'Average Dew Point'**)  
plt.show()

Output:

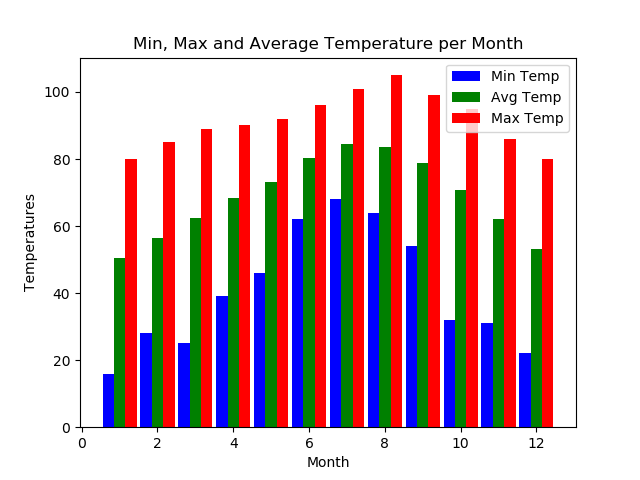


1. Create a bar chart, with one bar per month, showing the average temperature along with error bars indicating the high and low temperatures from that month.
   1. Note: You will want to create new lists of data, but you may find it useful to use the max/min/sum functions on lists.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 10B  
# Date: 04-11-2018***import** numpy **as** np  
**import** matplotlib  
matplotlib.use(**"TkAgg"**)  
**from** matplotlib **import** pyplot **as** plt  
  
  
*# First we need to calculate the average for each month:***def** average\_temperature(x):  
 file = open(**'WeatherDataMac.csv'**, **'r'**)  
 d = 0  
 total\_temp = 0  
 count = 0  
 file.readline() *# skip the first line* **for** row **in** file:  
 **if** int(row.split(**'/'**)[0]) == x:  
 d = float(row.split(**','**)[2])  
 total\_temp += d  
 count += 1  
 file.close()  
 **return** total\_temp/count  
  
  
*# Functions to get max and min temperatures for each month:***def** max(x):  
 file = open(**'WeatherDataMac.csv'**, **'r'**)  
 max = 0  
 file.readline() *# skip the first line* **for** row **in** file:  
 **if** int(row.split(**'/'**)[0]) == x:  
 d = float(row.split(**','**)[1])  
 **if** d > max:  
 max = d  
 file.close()  
 **return** max  
  
**def** min(x):  
 file = open(**'WeatherDataMac.csv'**, **'r'**)  
 min = 100  
 file.readline() *# skip the first line* **for** row **in** file:  
 **if** int(row.split(**'/'**)[0]) == x:  
 d = float(row.split(**','**)[3])  
 **if** d < min:  
 min = d  
 file.close()  
 **return** min  
  
  
average\_list = []  
max\_list = []  
min\_list = []  
**for** i **in** range (1,13,1):  
 average\_list.append((average\_temperature(i)))  
 max\_list.append((max(i)))  
 min\_list.append((min(i)))  
  
print(average\_list)  
print(max\_list)  
print(min\_list)  
  
*# Plot graph:*x = np.array([1,2,3,4,5,6,7,8,9,10,11,12])  
ax = plt.subplot(111)  
w = 0.3  
p1= ax.bar(x-w,min\_list,width=w,color=**'b'**)  
p2= ax.bar(x,average\_list,width=w,color=**'g'**)  
p3= ax.bar(x+w,max\_list,width=w,color=**'r'**)  
plt.title(**'Min, Max and Average Temperature per Month'**)  
plt.xlabel(**'Month'**)  
plt.ylabel(**'Temperatures'**)  
plt.legend((p1[0],p2[0],p3[0]),(**'Min Temp'**,**'Avg Temp'**,**'Max Temp'**))  
plt.show()

Output:



Be sure to include legends, labels, title, etc. on all charts, and be sure your scales/ranges and such are set appropriately to display all the data.