Ouermi Timbwaoga Aime Judicael
Course : Phys 410

All my work can be found here.
https://github.com/touermi/UO-2015-PHYS-410-510.git

Introduction

This assignment is much less involved than the previous ones we had. Here, I believe we are focusing on of learning how to better format our output data be ready for softwares so we don't have to waste a lot time copying and pasting JSFIDDLE. In addition we are looking at the data analysis aspect of this project.

1. Understanding of data

There is not much to do here after reading at prom we realized we only need column C and D for the different calculation and they were already cleaned up. All I did was save those data in a .txt files for so I can easily use it because I was not familiar on how to manipulate excel file from python.

2. Standard Deviation

On timescales of a decade, compute or determine the average and standard deviations per decade. It would be most useful if you coded your own standard deviation program.

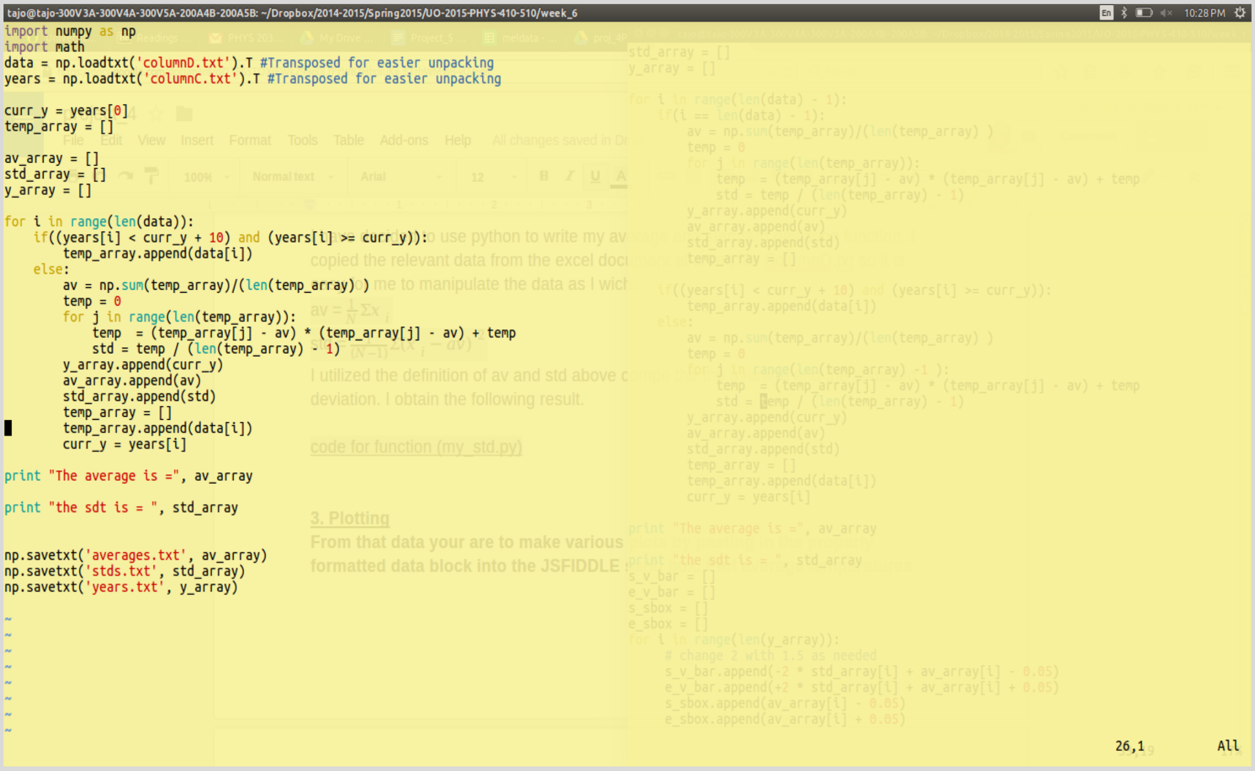
I have decided to use python to write my average and standard deviation function. I copied the relevant data from the excel document at a text file ColumnD.txt so it is easy for me to manipulate the data as I wich.
av = $\frac{1}{N}\Sigma x_i$
std = $\frac{1}{(N-1)}\Sigma(x_i - av)^2$
I utilized the definition of av and std above compe the the average and the standard deviation. I obtain the following result.

```
The averages =
[57.829999999999998, 57.839999999999996, 57.990000000000009,
57.970000000000006, 57.789999999999999, 57.720000000000006,
57.729999999999997, 57.950000000000003, 58.269999999999996,
58.350000000000009, 58.219999999999992, 58.299999999999997,
```

58.589999999999996, 59.140000000000008, 59.429999999999993, 59.689999999999998]

the sdts = [0.022333333333333035, 0.029333333333333325, 0.058777777777777693, 0.020111111111111168, 0.034333333333333139, 0.048444444444444353, 0.044555555555555432, 0.013888888888889284, 0.024555555555555275, 0.053888888888888884, 0.061777777777777793, 0.035555555555555299, 0.11211111111111095, 0.020444444444444262, 0.066777777777777533, 0.045444444444443774]

Code for function (my_std.py)



tajo@tajo-300V3A-300V4A-300V5A-200A4B-200A5B: ~/Dropbox/2014-2015/Spring2015/UO-2015-PHYS-410-510/week_6

```
import numpy as np
import math
data = np.loadtxt('columnD.txt').T #Transposed for easier unpacking
years = np.loadtxt('columnC.txt').T #Transposed for easier unpacking

curr_y = years[0]
temp_array = []

av_array = []
std_array = []
y_array = []

for i in range(len(data)):
    if((years[i] < curr_y + 10) and (years[i] >= curr_y)):
        temp_array.append(data[i])
    else:
        av = np.sum(temp_array)/(len(temp_array) )
        temp = 0
        for j in range(len(temp_array)):
            temp  = (temp_array[j] - av) * (temp_array[j] - av) + temp
            std = temp / (len(temp_array) - 1)
        y_array.append(curr_y)
        av_array.append(av)
        std_array.append(std)
        temp_array = []
        temp_array.append(data[i])
        curr_y = years[i]

print "The average is =", av_array

print "the sdt is = ", std_array


np.savetxt('averages.txt', av_array)
np.savetxt('stds.txt', std_array)
np.savetxt('years.txt', y_array)

~
~
~
~
~
~
~
~
                                                          26,1          All
```

3. Plotting

From that data your are to make various plots by pasting in the properly formatted data block into the JSFIDDLE sand box. Plot average temperatures and deviations as a function of time using this specific Chart Library Click on the JSFiddle link after the first example.

- **The width and height of the box should be 2 years x 0.1 C**
- **The size of the vertical bars should be +/- 1.5 standard deviations.**
- **Your plot should look something like this:**

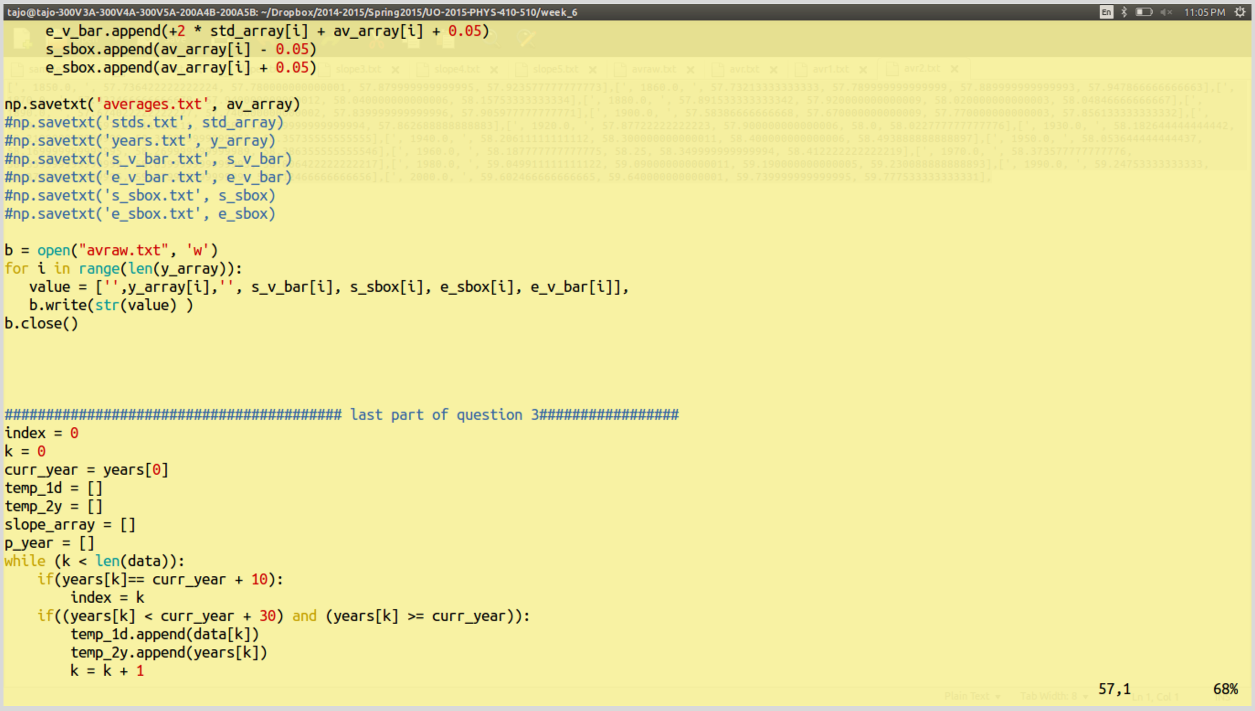
In this section we are produce multiple plot using JSFIDDLE. In order to do so we are going to reuse the previous code. We run the program and save

years.txt (which contains the outputs years we need),
averages.txt (which contains the averages we need), and
stds.txt (which contains the stds)

In order to produce the plots I need 5 entrees by values. Since we do not have many much data I could calculate the entrees by hand and input. However, for good practice I will write a code that calculate the 5 entrees for me.

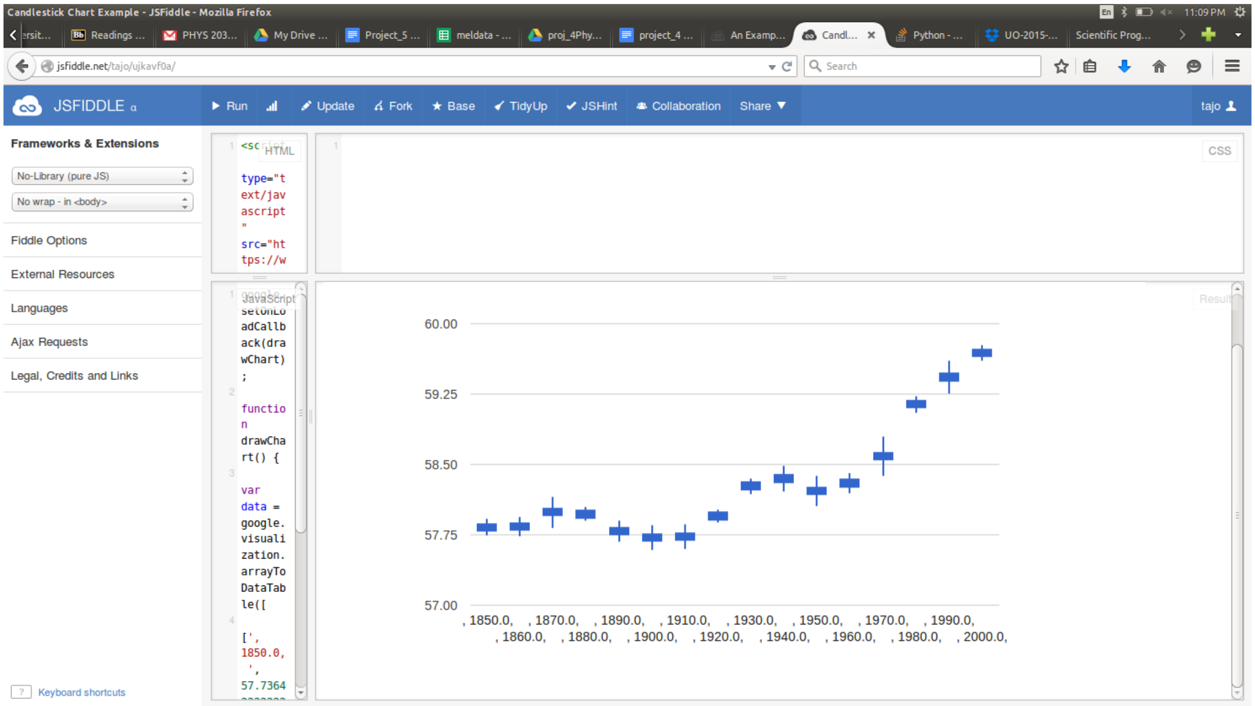
['year', start of vertical bar, start of sandbox, and of sandbox, end of vertical line]
start of vertical bar = - 1.5 * std + av -0.05
end of vertical bar = - 1.5 * std + av + 0.05
start of sandbox = av - 0.05
start of sandbox = av + 0.05

Formating Code


tajo@tajo-300V3A-300V4A-300V5A-200A4B-200A5B: ~/Dropbox/2014-2015/Spring2015/UO-2015-PHYS-410-510/week_6
11:05 PM
57,1
68%

```
    e_v_bar.append(+2 * std_array[i] + av_array[i] + 0.05)
    s_sbox.append(av_array[i] - 0.05)
    e_sbox.append(av_array[i] + 0.05)

np.savetxt('averages.txt', av_array)
#np.savetxt('stds.txt', std_array)
#np.savetxt('years.txt', y_array)
#np.savetxt('s_v_bar.txt', s_v_bar)
#np.savetxt('e_v_bar.txt', e_v_bar)
#np.savetxt('s_sbox.txt', s_sbox)
#np.savetxt('e_sbox.txt', e_sbox)

b = open("avraw.txt", 'w')
for i in range(len(y_array)):
   value = ['',y_array[i],'', s_v_bar[i], s_sbox[i], e_sbox[i], e_v_bar[i]],
   b.write(str(value) )
b.close()



######################################### last part of question 3#################
index = 0
k = 0
curr_year = years[0]
temp_1d = []
temp_2y = []
slope_array = []
p_year = []
while (k < len(data)):
    if(years[k]== curr_year + 10):
        index = k
    if((years[k] < curr_year + 30) and (years[k] >= curr_year)):
        temp_1d.append(data[k])
        temp_2y.append(years[k])
        k = k + 1
```

After I get the avraw.txt file we then use seed to finished the from formating by removing all the garbages that we don’t want.



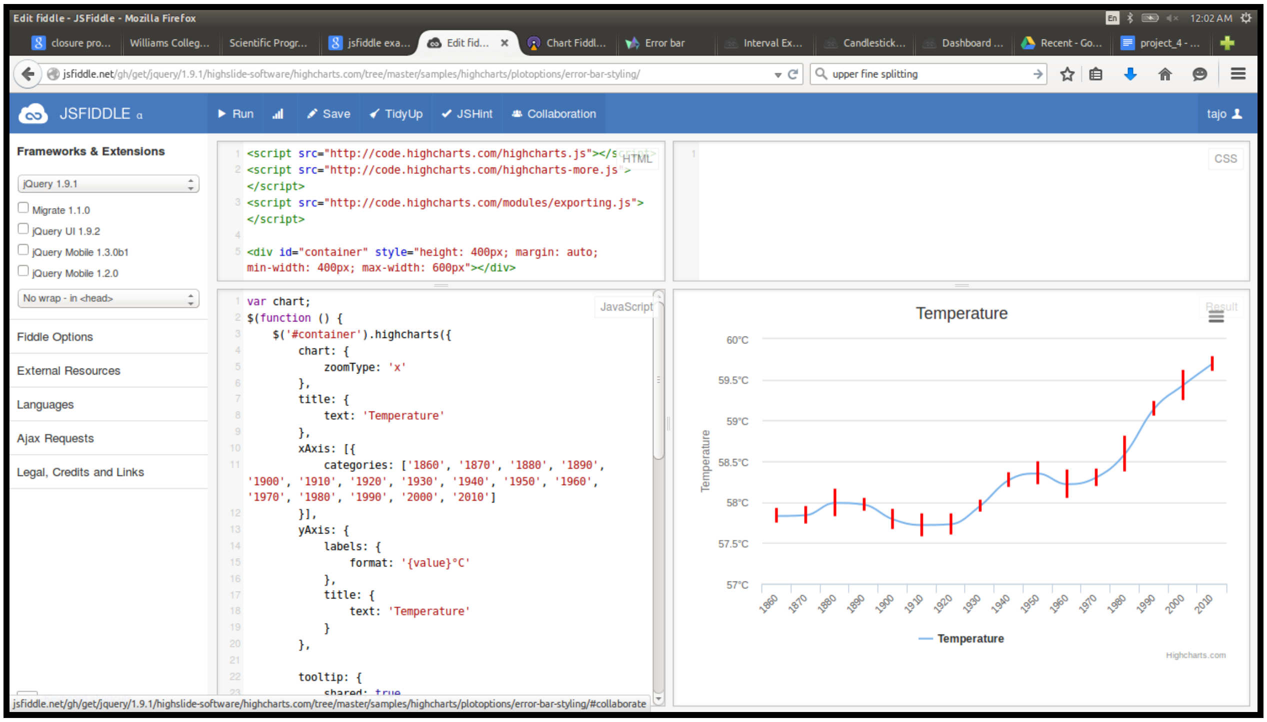
```
<script
type="text/javascript"
src="https://w
```

```
google.setOnLoadCallback(drawChart);
function drawChart() {
var data = google.visualization.arrayToDataTable([
[',
1850.0,
 ',
57.7364
```

Fig1.

Now plot the average temperature line along with the +/- 2 standard deviation envelopes using This Chart Library

In this section we need to change +/-1.5 standard deviation to +/-2 standard deviation to reflect the output we want. For this plot we need the actual average and the range of the error bar. we can save those in file and use it.
After feed the data to JSFIDDLE we get



```
<script src="http://code.highcharts.com/highcharts.js"></s
<script src="http://code.highcharts.com/highcharts-more.js">
</script>
<script src="http://code.highcharts.com/modules/exporting.js">
</script>

<div id="container" style="height: 400px; margin: auto;
min-width: 400px; max-width: 600px"></div>
```

```
var chart;
$(function () {
    $('#container').highcharts({
        chart: {
            zoomType: 'x'
        },
        title: {
            text: 'Temperature'
        },
        xAxis: [{
            categories: ['1860', '1870', '1880', '1890',
'1900', '1910', '1920', '1930', '1940', '1950', '1960',
'1970', '1980', '1990', '2000', '2010']
        }],
        yAxis: {
            labels: {
                format: '{value}°C'
            },
            title: {
                text: 'Temperature'
            }
        },

        tooltip: {
            shared: true
```

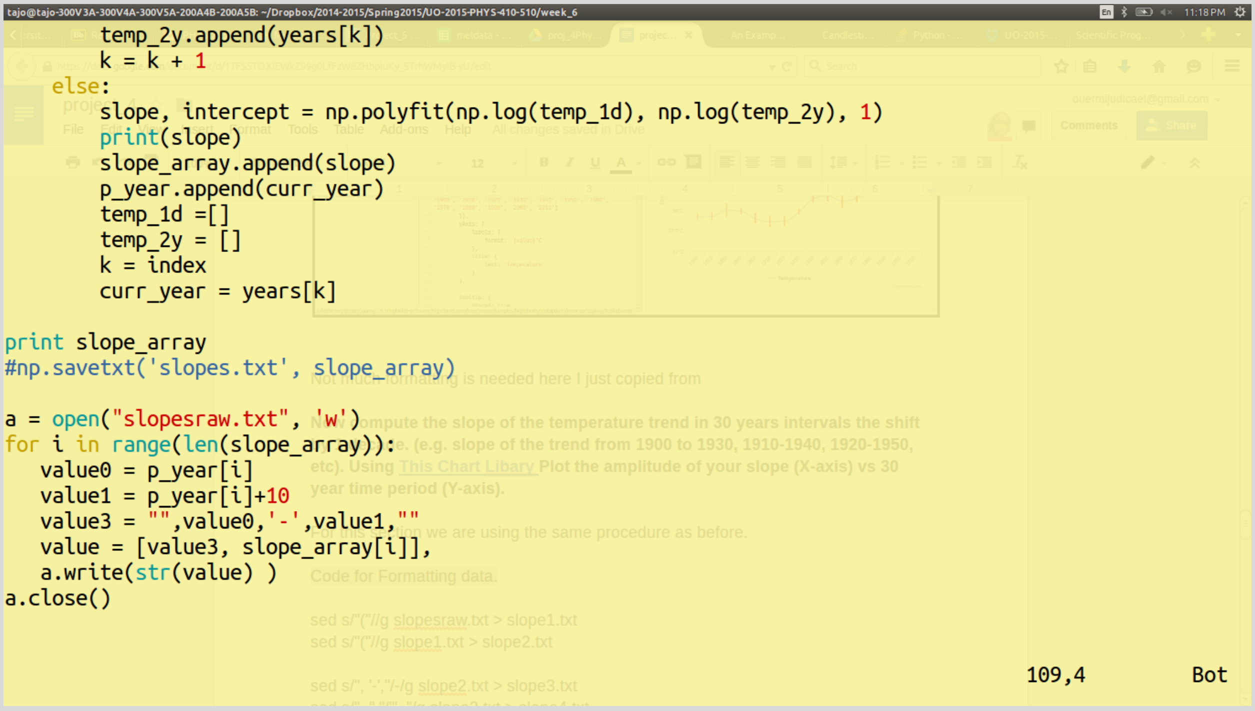
Link

Not much formatting is needed here I just copied data from .txt files I saved and passed it them into JSFIDDLE.

Now compute the slope of the temperature trend in 30 years intervals the shift by 1 decade. (e.g. slope of the trend from 1900 to 1930, 1910-1940, 1920-1950, etc). Using This Chart Libary Plot the amplitude of your slope (X-axis) vs 30 year time period (Y-axis).

For this section we are using the same procedure as before.

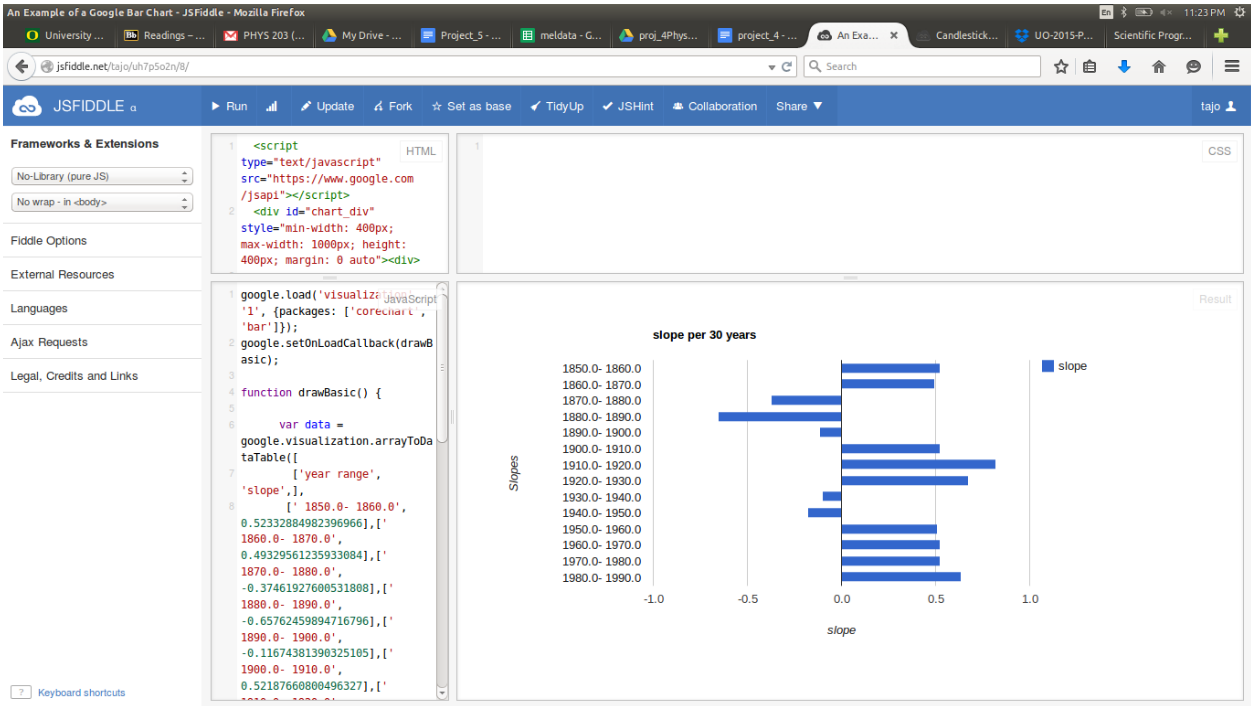
Code for Formatting data.


tajo@tajo-300V3A-300V4A-300V5A-200A4B-200A5B: ~/Dropbox/2014-2015/Spring2015/UO-2015-PHYS-410-510/week_6
11:18 PM

```
        temp_2y.append(years[k])
        k = k + 1
    else:
        slope, intercept = np.polyfit(np.log(temp_1d), np.log(temp_2y), 1)
        print(slope)
        slope_array.append(slope)
        p_year.append(curr_year)
        temp_1d =[]
        temp_2y = []
        k = index
        curr_year = years[k]

print slope_array
#np.savetxt('slopes.txt', slope_array)

a = open("slopesraw.txt", 'w')
for i in range(len(slope_array)):
   value0 = p_year[i]
   value1 = p_year[i]+10
   value3 = "",value0,'-',value1,""
   value = [value3, slope_array[i]],
   a.write(str(value) )
a.close()

                                                                      109,4         Bot
```

Below are different use of sed to obtain final data

```
sed s/"("//g slopesraw.txt > slope1.txt
sed s/"("//g slope1.txt > slope2.txt
sed s/", '-',"/-/g slope2.txt > slope3.txt
sed s/", ","/'", "/g slope3.txt > slope4.txt
```



```
  <script
type="text/javascript"
src="https://www.google.com
/jsapi"></script>
  <div id="chart_div"
style="min-width: 400px;
max-width: 1000px; height:
400px; margin: 0 auto"><div>
```

```
google.load('visualization',
'1', {packages: ['corechart',
'bar']});
google.setOnLoadCallback(drawB
asic);

function drawBasic() {

      var data =
google.visualization.arrayToDa
taTable([
        ['year range',
'slope',],
       [' 1850.0- 1860.0',
0.52332884982396966],['
1860.0- 1870.0',
0.49329561235933084],['
1870.0- 1880.0',
-0.37461927600531808],['
1880.0- 1890.0',
-0.65762459894716796],['
1890.0- 1900.0',
-0.11674381390325105],['
1900.0- 1910.0',
0.52187660800496327],['
```

[Link]

Note: I just realised that I made in labeling my plot above. However the result is correct.

Conclusion

This Assignment made me realise how much time I could save by formatting my data before I start feeding to an external software. This was also an opportunity to better explore JSFIDDLE