RI CEP Year 2 - Problem Set (20 marks)	**Due Date: 14 Sept, 2017 at 23:59**

In this assignment, we're going to look at weather data. You'll eventually be collecting your own weather data using your Raspberry Pis, but for now, we'll use historical and current weather data available online. You may use either IDLE or Jupyter notebook for this assignment (note that the school computers do not have the matplotlib module installed for IDLE).

Part 1 - Data Acquisition (1 mark)
Oftentimes, data scientists will have to make tough decisions when they're searching for data. They might face incomplete data, inaccurate data, or data that is hard to access. The website www.weather.gov.sg/climate-historical-daily/ offers daily historical weather data for us to download. In this case, data is available from 63 different stations across the island, and the data comes nicely formatted, but separated by month. We'll need to manually download data for each month (sadly they don't have an API) and combine them into one dataset.

We'll be working with data from Ang Mo Kio station: download the data for each month from January 2017 through July 2017 in csv format. Import the data and combine them into one dataframe.

Hint: When importing these CSV files, you'll need to specify the encoding, for example:

```
pd.read_csv("DAILYDATA_S109_201701.csv", encoding="ISO-8859-1")
```

Part 2 - Plotting with Matplotlib (8 marks)
Our next goal is to create two plots using matplotlib: one plotting mean, maximum, and minimum temperature, and the other plotting mean and max wind speed.

Hint: Use the following code to create a new column of datetime values to use for your x coordinates:

```
#combine Year/Month/Day columns into datetime column
data["datetime"] = pd.to_datetime(data.ix[:,'Year':'Day'])
```

You are required to include:

- A plot title
- Line labels (i.e. Max Temp, Mean Temp, Min Temp, etc.)
- Use of a matplotlib stylesheet (use one already created, no need to create your own!)

When done plotting, save your plot as a png file using the following format:

yourname_temp.png
yourname_wind.png

Part 3 - Plotting with plot.ly (11 marks)
Finally, we want to create a visualisation using live weather data. For this, we'll use Particulate Matter 2.5 readings.

a. Request the latest Particulate Matter 2.5 readings from data.gov.sg. Store the returned data in a python dictionary. (1 mark)
b. PM 2.5 readings are taken from five different geographical locations. Average these five readings to obtain an average PM 2.5 reading for the entire island of Singapore. (1 mark)
c. Use the region_metadata field in the response to plot the longitude and latitude coordinates of the five locations on a map of Singapore. Use plotly and Mapbox to accomplish this (you'll need a Mapbox account for an API access token). Your code should read the longitude and latitude coordinates directly from the response data; coordinates should not be hardcoded. Your hover text should display:
   - the location name (north, east, central, etc.)
   - the current PM reading for that location.

   (9 marks)

When done plotting, save your plot as an html file using the following format:

yourname_PM25.html

Submit your .py file via direct message in Slack (submitting your .png files isn't required as your python program should create them). Please name your .py file using the following convention:

<lastname> <firstname> PS1.py (e.g. "Gonsalves Mike PS1.py")