* **Read about Seaborn library** - <http://seaborn.pydata.org/examples/>
* **Read about citipy library** - <https://pypi.python.org/pypi/citipy>
  + <https://github.com/wingchen/citipy> - Review the README.md for installation details.
  + Installation from Git-Bash prompt > pip install citipy
  + Installation from Jupyter Notebook > !pip install citipy
  + *#Test Looking up with coordinates*

*from citipy import citipy as cp*

*city = cp.nearest\_city(22.99, 120.21)*

*print(city) # <citipy.City instance at 0x1069b6518>*

*print(city.city\_name) # Tainan, my home town 'tainan'*

*print(city.country\_code) # And the country is surely Taiwan 'tw'*

* + citipy creator says: You can get Maxmind's Free World Cities Database here: <https://www.maxmind.com/en/free-world-cities-database>
* **Your objective:** build a series of scatter plots to showcase the following relationships:
  + Temperature (F) vs. Latitude
  + Humidity (%) vs. Latitude
  + Cloudiness (%) vs. Latitude
  + Wind Speed (mph) vs. Latitude
* **Your final notebook must:**
  + Randomly select **\*\*at least\*\*** 500 unique (non-repeat) cities based on latitude and longitude.
  + Perform a weather check on each of the cities using a series of successive API calls.
  + Include a print log of each city as it's being processed with the city number, city name, and requested URL.
  + Save both a CSV of all data retrieved and png images for each scatter plot.
* **As final considerations:**
  + You must use the Matplotlib and Seaborn libraries.
  + You must include a written description of three observable trends based on the data.
  + You must use proper labeling of your plots, including aspects like: Plot Titles (with date of analysis) and Axes Labels.
  1. You must include an exported markdown version of your Notebook called README.md in your GitHub repository.
     1. Open Git-Bash
     2. My version:

jupyter nbconvert --to markdown PyWeather\_hw5.ipynb

* + 1. Submit version:

jupyter nbconvert --to markdown PyWeather\_hw5-submit.ipynb

* + See the example PDF file for a a reference of expected format.
* **Hints and Considerations**
* You may want to start this assignment by refreshing yourself on 4th grade geography, in particular, the [geographic coordinate system](http://desktop.arcgis.com/en/arcmap/10.3/guide-books/map-projections/about-geographic-coordinate-systems.htm).
* Next, spend the requisite time necessary to study the OpenWeatherMap API. Based on your initial study, you should be able to answer  basic questions about the API: Where do you request the API key? Which Weather API in particular will you need? What URL endpoints does it expect? What JSON structure does it respond with? Before you write a line of code, you should be aiming to have a crystal clear understanding of your intended outcome.
* Though we've never worked with the <https://pypi.python.org/pypi/citipy>, push yourself to decipher how it works, and why it might be relevant. Before you try to incorporate the library into your analysis, start by creating simple test cases outside your main script to confirm that you are using it correctly. Too often, when introduced to a new library, students get bogged down by the most minor of errors -- spending hours investigating their entire code -- when, in fact, a simple and focused test would have shown their basic utilization of the library was wrong from the start. Don't let this be you!
* Part of our expectation in this challenge is that you will use critical thinking skills to understand how and why we're recommending the tools we are. What is Citipy for? Why would you use it in conjunction with the OpenWeatherMap API? How would you do so?
* In building your script, pay attention to the cities you are using in your query pool. Are you getting coverage of the full gamut of latitudes and longitudes? Or are you simply choosing 500 cities concentrated in one region of the world? Even if you were a geographic genius, simply rattling 500 cities based on your human selection would create a biased dataset. Be thinking of how you should counter this. (Hint: Consider the full range of latitudes).
* Lastly, remember -- this is a challenging activity. Push yourself! If you complete this task, then you can safely say that you've gained a strong mastery of the core foundations of data analytics and it will only go better from here. Good luck!

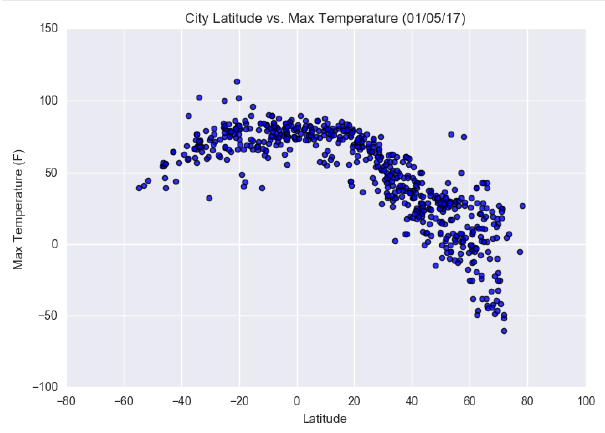
**My steps to complete HW5 PyWeather:**

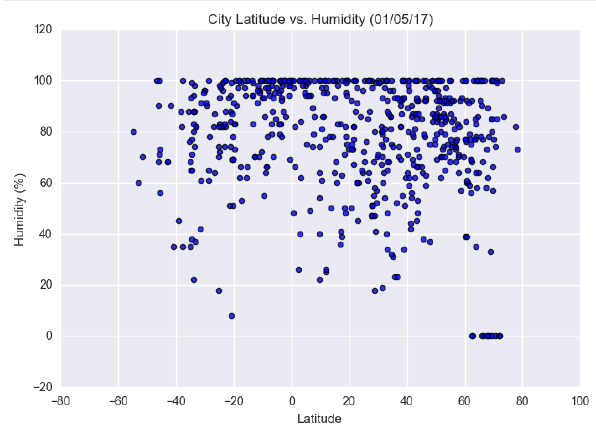
1. Create folders under Python-Challenges and gather homework documents.
   1. Download README.md file and rename as README\_hw5.md, and download WeatherPy\_Example.pdf and rename as WeatherPy\_Example\_hw5.pdf.
   2. Create PyWeather\_hw5.ipynb and paste in the appropriate libraries - especially seaborn and citipy libraries which are required by this assignment.
   3. Download citipy.zip, install, and test with the code snippet above.
2. Steps to my program:
   1. Reserve a cell for the 3 observed trends.
   2. Generate a list of 500 unique cities that are widely distributed.

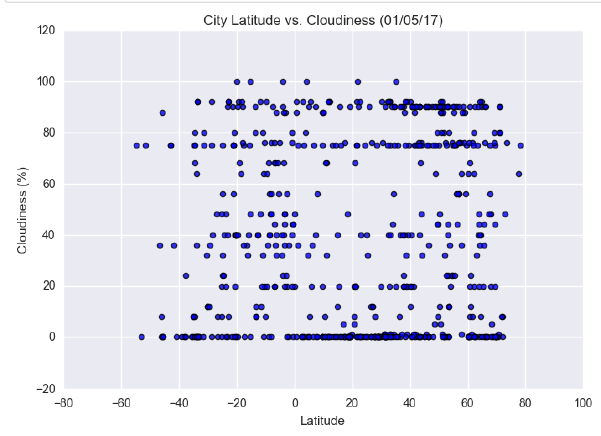
In building your script, pay attention to the cities you are using in your query pool. Are you getting coverage of the full gamut of latitudes and longitudes? Or are you simply choosing 500 cities concentrated in one region of the world? Even if you were a geographic genius, simply rattling 500 cities based on your human selection would create a biased dataset. Be thinking of how you should counter this. (Hint: Consider the full range of latitudes).

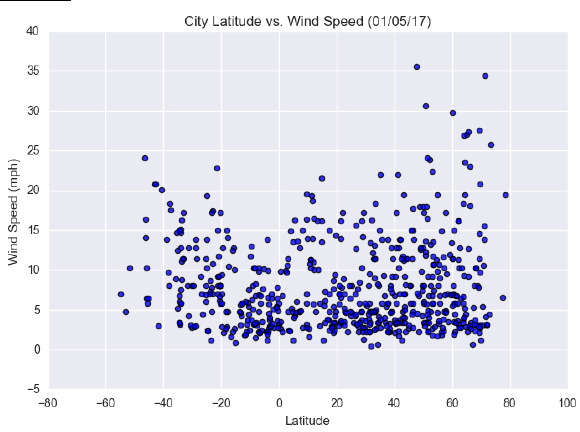
* 1. Perform API calls to the OpenWeatherMap.
     1. Create a data frame with the city numbers, city name, and requested url, and add empty columns like we did for the popular tweets activity.
     2. Log information of each city processed including the city number, city name, and requested URL.
     3. Collect temperature, humidity, cloudiness, wind speed values.
     4. Save the city data as a csv file.

1. build a series of scatter plots to showcase the following relationships:
2. Temperature (F) vs. Latitude
3. Humidity (%) vs. Latitude
4. Cloudiness (%) vs. Latitude
5. Wind Speed (mph) vs. Latitude
6. 4 Plot examples from README.md









1. Create a markdown file of the notebook and folder of images.
2. Create a markdown version of your Notebook called README.md.

Open Git-Bash

**Markdown code for my version:**

jupyter nbconvert --to markdown PyWeather\_hw5.ipynb

**Markdown code for submit version:**

jupyter nbconvert --to markdown PyWeather\_hw5-submit-version.ipynb

1. Clean up files: rename and or delete if necessary.
2. Submit city csv file, the Jupyter Notebook, and the markdown file to GitHub and maybe Google drive.
3. For creating the 500 cities, divide the earth into quadrants: <http://i37.tinypic.com/345izb4.jpg>

