



[add 3.0 solution, 5.3 activities, 6.0 homework](#)
[Hunter S Carlisle](#) authored 2 weeks ago

3bc51034

Name	Last commit	Last update
..		
Images	add 3.0 solution, 5.3 activities, 6.0 homework	2 weeks ago
output_data	add 3.0 solution, 5.3 activities, 6.0 homework	2 weeks ago
starter_code	add 3.0 solution, 5.3 activities, 6.0 homework	2 weeks ago
Python_API_grading_rubric.pdf	add 3.0 solution, 5.3 activities, 6.0 homework	2 weeks ago
README.md	add 3.0 solution, 5.3 activities, 6.0 homework	2 weeks ago

[README.md](#)

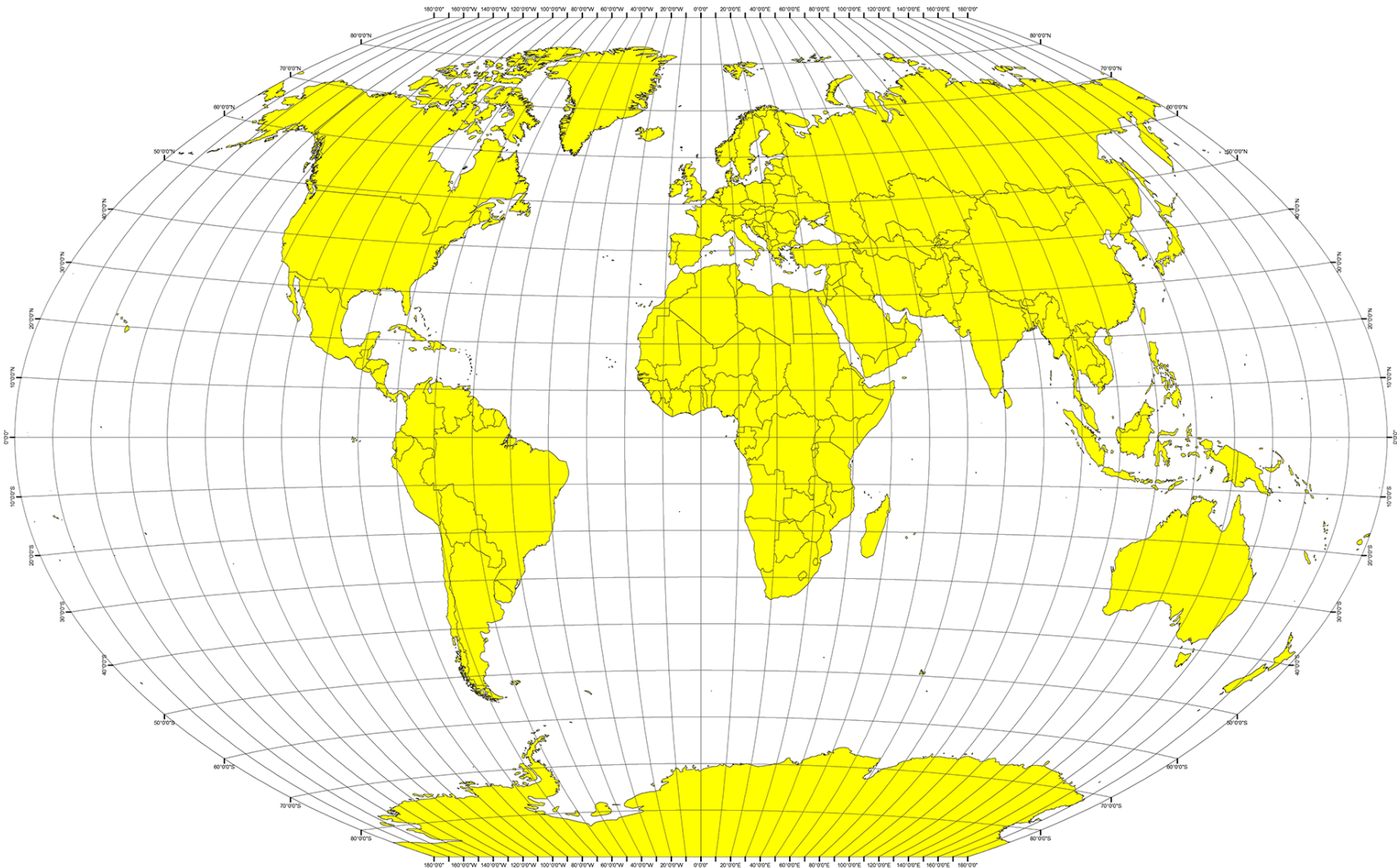
Python API Homework - What's the Weather Like?

Background

Whether financial, political, or social -- data's true power lies in its ability to answer questions definitively. So let's take what you've learned about Python requests, APIs, and JSON traversals to answer a fundamental question: "What's the weather like as we approach the equator?"

Now, we know what you may be thinking: "*Duh. It gets hotter...*"

But, if pressed, how would you **prove** it?



Before You Begin

1. Create a new repository for this project called `python-api-challenge` . **Do not add this homework to an existing repository.**
2. Clone the new repository to your computer.

3. Inside your local git repository, create a directory for both of the Python Challenges. Use folder names corresponding to the challenges: **WeatherPy**.
4. Inside the folder that you just created, add new files called `WeatherPy.ipynb` and `VacationPy.ipynb`. These will be the main scripts to run for each analysis.
5. Push the above changes to GitHub.

Part I - WeatherPy

In this example, you'll be creating a Python script to visualize the weather of 500+ cities across the world of varying distance from the equator. To accomplish this, you'll be utilizing a [simple Python library](#), the [OpenWeatherMap API](#), and a little common sense to create a representative model of weather across world cities.

Your first requirement is to create 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

After each plot add a sentence or too explaining what the code is and analyzing.

Your second requirement is to run linear regression on each relationship, only this time separating them into Northern Hemisphere (greater than or equal to 0 degrees latitude) and Southern Hemisphere (less than 0 degrees latitude):

- Northern Hemisphere - Temperature (F) vs. Latitude
- Southern Hemisphere - Temperature (F) vs. Latitude
- Northern Hemisphere - Humidity (%) vs. Latitude
- Southern Hemisphere - Humidity (%) vs. Latitude
- Northern Hemisphere - Cloudiness (%) vs. Latitude
- Southern Hemisphere - Cloudiness (%) vs. Latitude
- Northern Hemisphere - Wind Speed (mph) vs. Latitude
- Southern Hemisphere - Wind Speed (mph) vs. Latitude

After each pair of plots explain what the linear regression is modeling such as any relationships you notice and any other analysis you may have.

Optional You will be creating multiple linear regression plots. To optimize your code, write a function that creates the linear regression plots.

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 and city name.
- Save a CSV of all retrieved data and a PNG image for each scatter plot.

Part II - VacationPy

Now let's use your skills in working with weather data to plan future vacations. Use jupyter-gmaps and the Google Places API for this part of the assignment.

- **Note:** if you having trouble displaying the maps try running `jupyter nbextension enable --py gmaps` in your environment and retry.
- Create a heat map that displays the humidity for every city from the part I of the homework.

- Narrow down the DataFrame to find your ideal weather condition. For example:
 - A max temperature lower than 80 degrees but higher than 70.
 - Wind speed less than 10 mph.
 - Zero cloudiness.
 - Drop any rows that don't contain all three conditions. You want to be sure the weather is ideal.

- **Note:** Feel free to adjust to your specifications but be sure to limit the number of rows returned by your API requests to a reasonable number.
- Using Google Places API to find the first hotel for each city located within 5000 meters of your coordinates.
- Plot the hotels on top of the humidity heatmap with each pin containing the **Hotel Name, City,** and **Country.**

As final considerations:

- Create a new GitHub repository for this project called **API-Challenge** (note the kebab-case). **Do not add to an existing repo**
- You must complete your analysis using a Jupyter notebook.
- You must use the Matplotlib or Pandas plotting libraries.
- For Part I, 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.
- For max intensity in the heat map, try setting it to the highest humidity found in the data set.

Hints and Considerations

- The city data you generate is based on random coordinates as well as different query times; as such, your outputs will not be an exact match to the provided starter notebook.
- You may want to start this assignment by refreshing yourself on the [geographic coordinate system](#).
- 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.
- A starter code for Citipy has been provided. However, if you're craving an extra challenge, push yourself to learn how it works: [citipy Python library](#). 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).
- Once you have computed the linear regression for one chart, the process will be similar for all others. As a bonus, try to create a function that will create these charts based on different parameters.
- Remember that each coordinate will trigger a separate call to the Google API. If you're creating your own criteria to plan your vacation, try to reduce the results in your DataFrame to 10 or fewer cities.
- 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!

Copyright

Trilogy Education Services © 2019. All Rights Reserved.