Module 6 Challenge

Submit Assignment

Due Monday by 2:59am **Points** 100 **Submitting** a text entry box or a website url

> Jack loves the app. Beta testers love the app. And, with any new product, there are some changes that could take it from a nice app to an awesome app. Your mission, should you choose to accept it, is to implement the feedback from the beta testers, which is listed below.
>
> 1. A weather description to the pop-up markers for customers so that they know what the weather is as they are traveling
> 2. A notation in the search criteria to indicate if it is raining or snowing for customers who are making travel decisions in real-time
> 3. A map that shows the directions for customers' travel itinerary

In this challenge, you will continue to build on your Python programming skills using `try-except` blocks, input statements and nested decision statements, logical expressions, and Pandas methods and attributes to make an API request and build a new DataFrame. You will then filter the DataFrame to create a marker layer map, and a directions layer Google map that shows directions to travel between multiple cities.

Background

For the new modifications to the PlanMyTrip app, you are asked to add more data to the database, or cities DataFrame, so that customers know the weather in the cities when they click on a pop-up marker. You'll also need to add the amount of rainfall or snowfall within the last three hours so that customers can filter the DataFrame using input statements based on the temperature range and whether or not it is raining or

snowing. Finally, you'll need to create a directions layer Google map that shows the directions between multiple cities for travel.

Objectives

The goals for this challenge are for you to:

- Use nested `try-except` blocks.
- Use Pandas methods and attributes on a DataFrame or Series.
- Create a new DataFrame from a new API search with new weather parameters.
- Filter DataFrames based on input and nested decision statements, and logical expressions.
- Create pop-up markers on a Google map from a filtered DataFrame.
- Add a directions layer on a Google map between cities in the filtered DataFrame.

Part 1 Instructions

Get the Weather Description and Amount of Precipitation for Each City

To complete this task, follow these steps:

1. Create a new Jupyter Notebook file and name it `Weather_Database.ipynb`.
2. Generate a new set of 1,500 random latitudes and longitudes.
3. Get the nearest city using the citipy module.
4. Perform an API call with the OpenWeatherMap.
5. Retrieve the following information from the API call:
    - Latitude and longitude
    - Maximum temperature
    - Percent humidity

- Percent cloudiness
- Wind speed
- Weather description (e.g., clouds, fog, light rain, clear sky)
- Using a `try-except` block, in the `try` block if it is raining, get the amount of rainfall in inches for the last three hours. In the `except` block handle the KeyError if there is no rainfall and add 0 inches for the rainfall amount.
- Using a `try-except` block, in the `try` block if it is snowing, get the amount of snowfall in inches for the last three hours. In the `except` block handle the KeyError if there is no snowfall and add 0 inches for the snowfall amount.

6. Add the data to a new DataFrame.
7. Save the new DataFrame as a CSV file to be used for Part 2.
8. Upload the CSV file as part of your submission as `WeatherPy_challenge.csv`.
9. **Answer this question using Pandas methods**: **How many cities have recorded rainfall or snow?**

Your new DataFrame should look similar to the following image:

	City	Country	Date	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Current Description	Rain inches (last 3 hrs)	Snow inches (last 3 hrs)
0	Castro	CL	2019-08-26 17:25:49	-42.48	-73.76	48.20	61	40	14.99	scattered clouds	0.000	0
1	Lebu	ET	2019-08-26 17:25:49	8.96	38.73	58.69	83	72	1.45	light rain	2.187	0
2	Sitka	US	2019-08-26 17:25:49	37.17	-99.65	90.00	46	6	21.00	clear sky	0.000	0
3	Torbay	CA	2019-08-26 17:25:49	47.66	-52.73	75.00	43	20	10.29	few clouds	0.000	0
4	Grandview	US	2019-08-26 17:25:50	38.89	-94.53	69.80	93	40	4.70	heavy intensity rain	0.000	0
5	Mataura	NZ	2019-08-26 17:25:50	-46.19	168.86	36.91	86	87	8.93	light rain	0.124	0
6	Cidreira	BR	2019-08-26 17:25:50	-30.17	-50.22	72.01	50	48	3.44	scattered clouds	0.000	0
7	Hilo	US	2019-08-26 17:25:50	19.71	-155.08	75.20	88	90	4.70	heavy intensity rain	0.000	0
8	Great Falls	US	2019-08-26 17:25:51	47.50	-111.29	63.00	58	1	2.39	clear sky	0.000	0
9	Lata	IN	2019-08-26 17:25:51	30.78	78.62	41.77	93	87	1.66	light rain	0.375	0

Part 2 Instructions

Have Customers Narrow Their Travel Searches Based on Temperature and Precipitation

To complete this task, follow these steps:

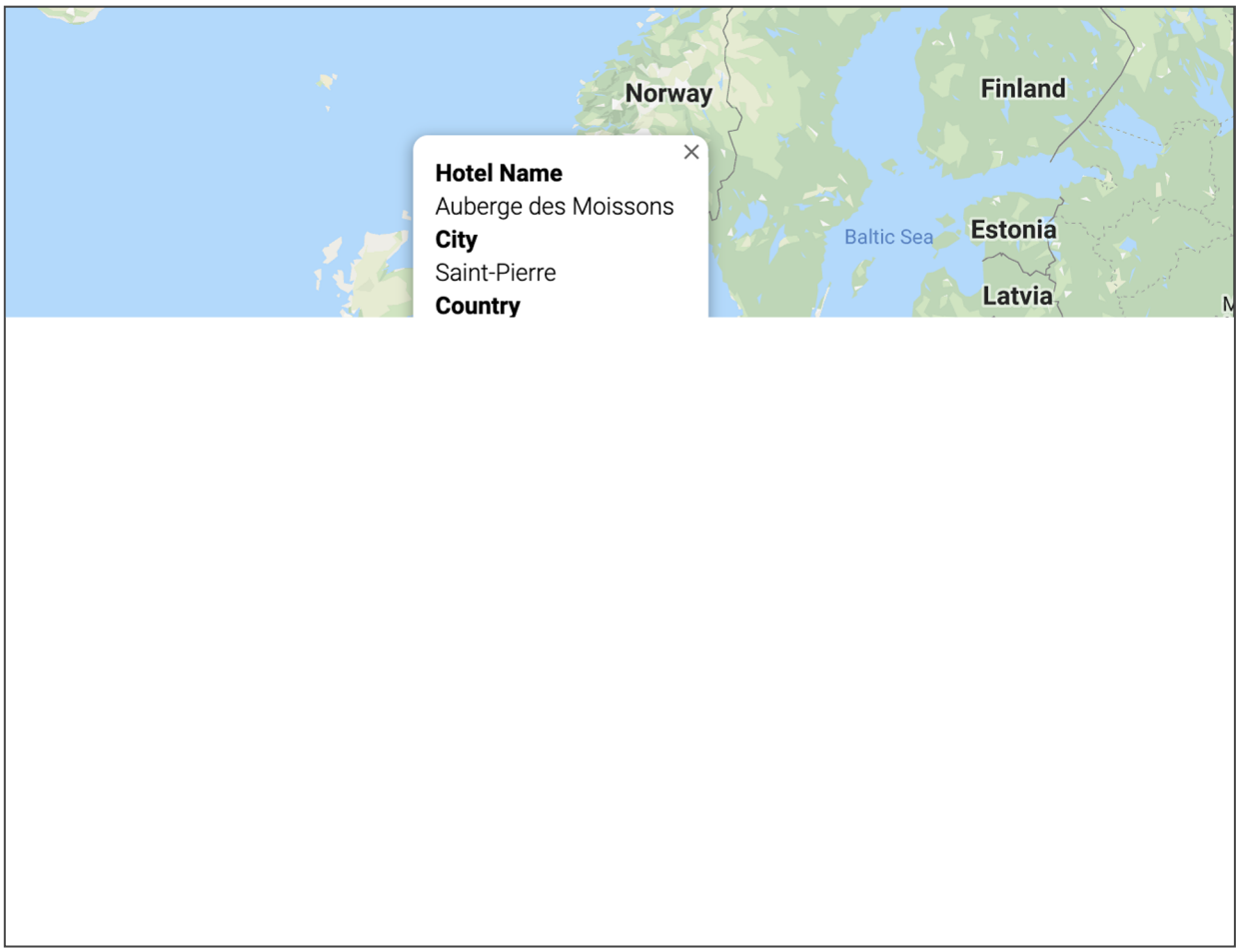
1. Create a new Jupyter Notebook file and name it `Vacation_Search.ipynb`.
2. Import the `WeatherPy_vacation.csv` file from Part 1 as a new DataFrame.
3. Filter the DataFrame for minimum and maximum temperature preferences, and if the rain or snow accumulation is 0 inches or not using conditional statements. Do the following:
    - Prompt the customer for the minimum temperature preference.
    - Prompt the customer for the maximum temperature preference.
    - Prompt the customer to answer if he or she would like it to be raining or not, using `input("Do you want it to be raining? (yes/no) ")`.
    - Prompt the customer to answer if he or she would like it to be snowing or not, using `input("Do you want it to be snowing? (yes/no) ")`.
4. Add the cities to a marker layer map with a pop-up marker for each city that includes:
    - Hotel name
    - City
    - Country
    - Current weather description with the maximum temperature
5. Save and upload the new DataFrame as `WeatherPy_vacation.csv`.
6. Save and upload the new marker layer map as `WeatherPy_vacation_map.png`.

Your new hotel DataFrame should look similar to the following image:

	City	Country	Max Temp	Current Description	Lat	Lng	Hotel Name
2	Sitka	US	90.00	clear sky	37.17	-99.65	
15	Avera	US	86.00	broken clouds	33.19	-82.53	
17	Kapaa	US	82.40	broken clouds	22.08	-159.32	Courtyard by Marriott Kaua'i at Coconut Beach
19	Rock Sound	BS	84.43	overcast clouds	24.90	-76.20	Rock Sound Club
23	Fazilka	IN	86.59	broken clouds	30.41	74.03	Hotel Gulmarg
34	Souillac	FR	91.40	clear sky	45.60	-0.60	Jolysable
35	Yatou	CM	82.40	broken clouds	3.63	9.81	Village Belart
37	Gazanjyk	TM	84.97	clear sky	39.24	55.52	
41	Dodge City	US	90.00	clear sky	37.75	-100.02	Stay Suites of America-Dodge City
45	Villa Maria	CO	80.60	broken clouds	4.50	-74.09	Rumbo al llano "www.musicallanera.co"

Your marker layer map should look similar to the following image:

A pop-up marker for each city should look similar to the following image:



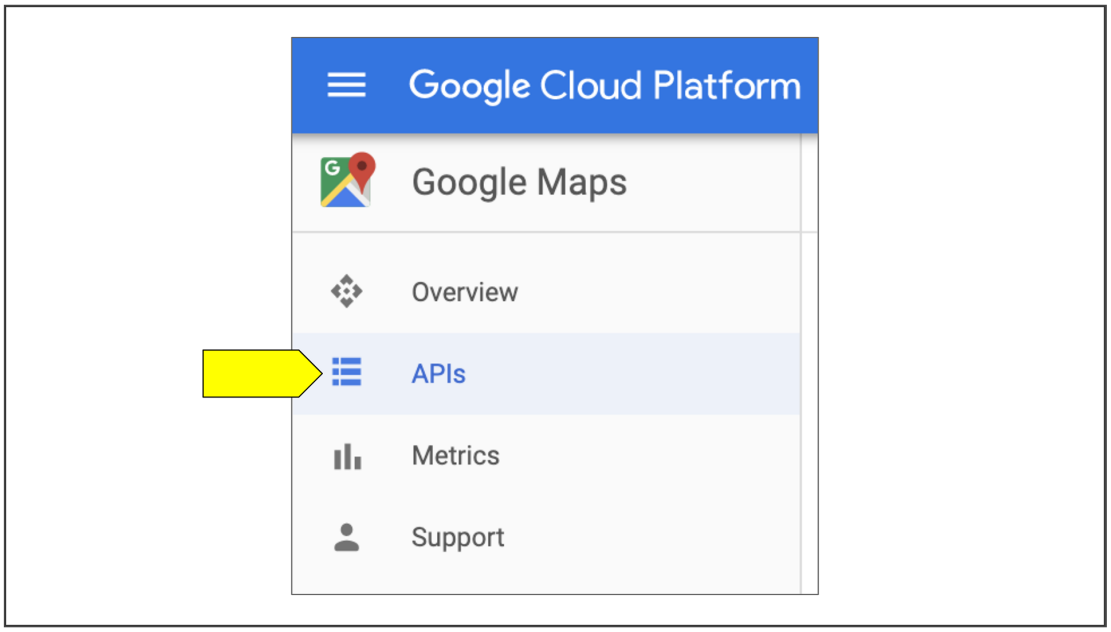
Part 3 Instructions

Create a Travel Itinerary with a Corresponding Map

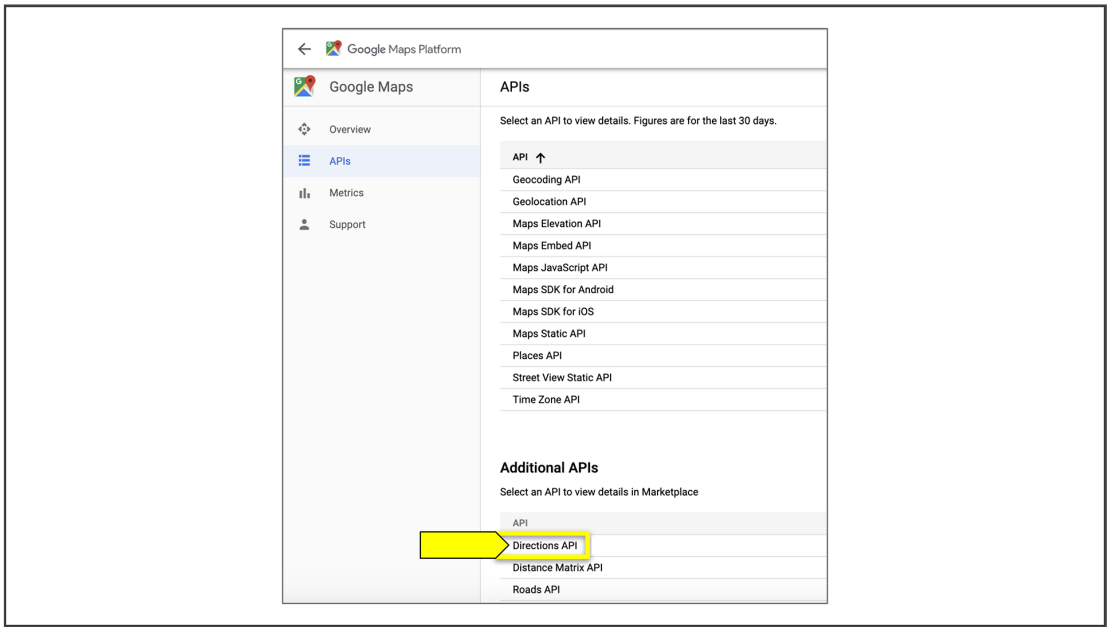
Finally, you will create a map (travel itinerary) that shows the route between four cities from the customer's possible travel destinations, and then create a map with pop-up markers for the four cities. To complete these tasks, follow these steps:

1. Enable the "Directions API" in your Google account for your API key.

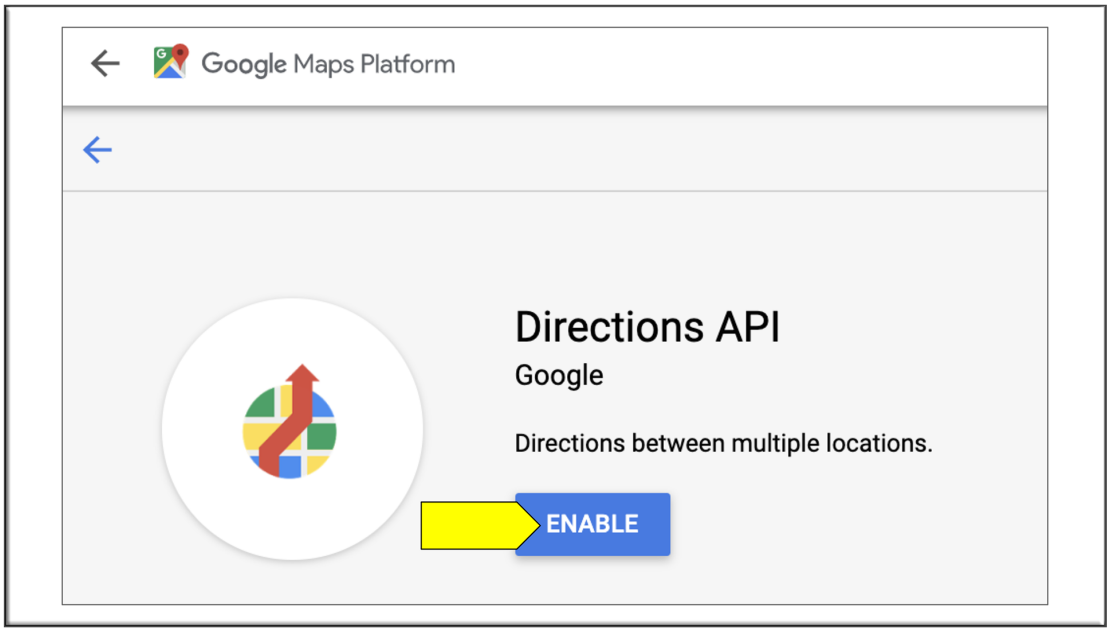
- On the Google Cloud Platform, select APIs from the left-hand side.



- Then, select "Directions API."



- Click “Enable” on the Directions API.



2. Create a new Jupyter Notebook file and label it `Vacation_Itinerary.ipynb`.
3. Import the `WeatherPy_vacation.csv` file as a new DataFrame.
4. From the vacation search map, ***choose at least four cities*** in close proximity on your map that are on the same continent that a customer might travel to, and then create a directions layer map.

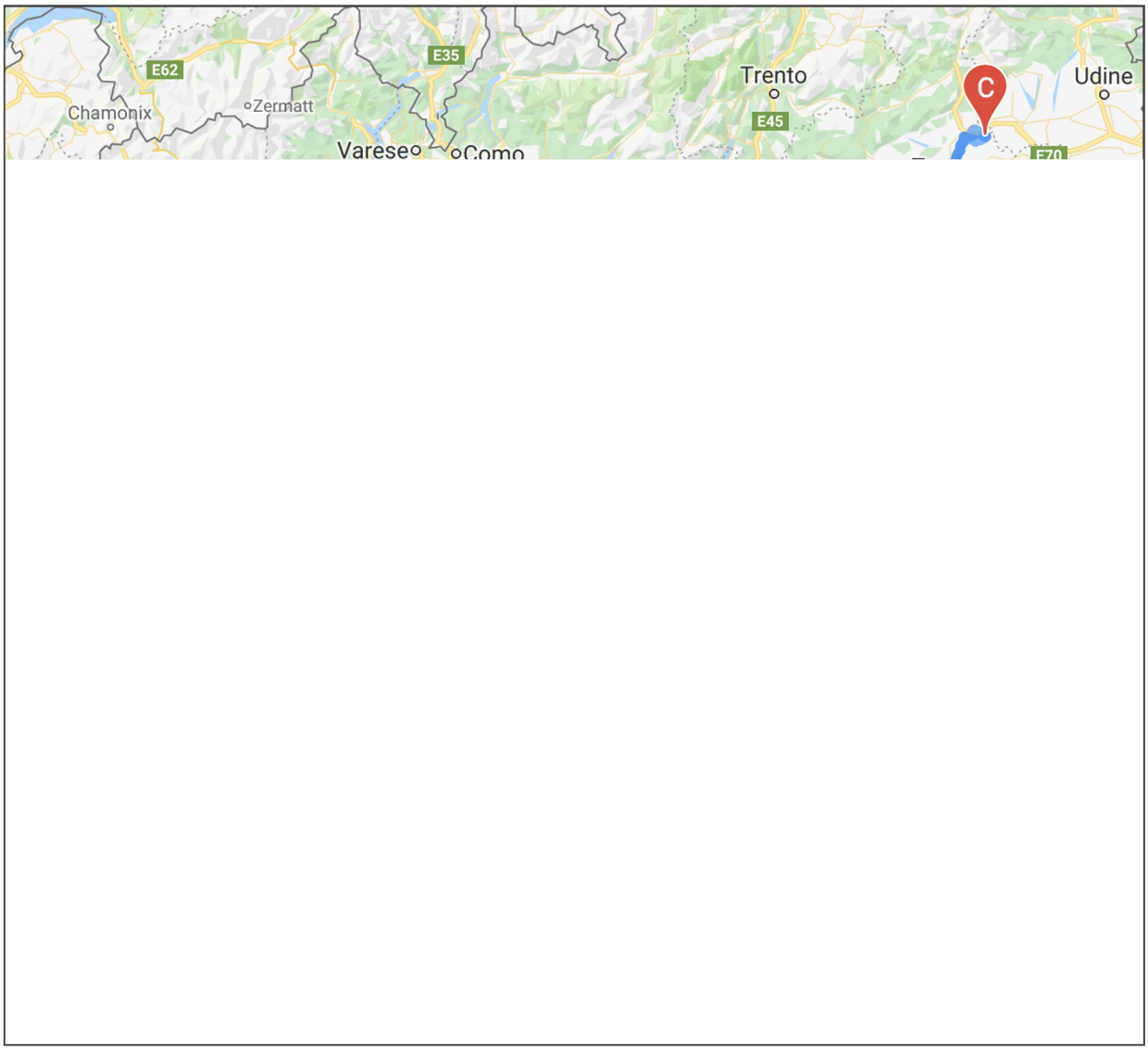
   Hints:

   - Filter the DataFrame for each city you want to go to and create separate DataFrames for each city.
   - Use the **directions Layer instructions from the gmaps documentation (https://jupyter-gmaps.readthedocs.io/en/latest/tutorial.html#directions-layer)**.
   - Use the list indexing and Pandas methods to get the latitude-longitude pairs for each city DataFrame as tuples.
5. For the `travel_mode`, use either DRIVING, BICYCLING, or WALKING.
   Hint: If the cities are too far apart, some travel modes will not be available.
6. Take a screenshot of the route and save it as `WeatherPy_travel_map.png`.
7. Create a marker layer map for the four cities.

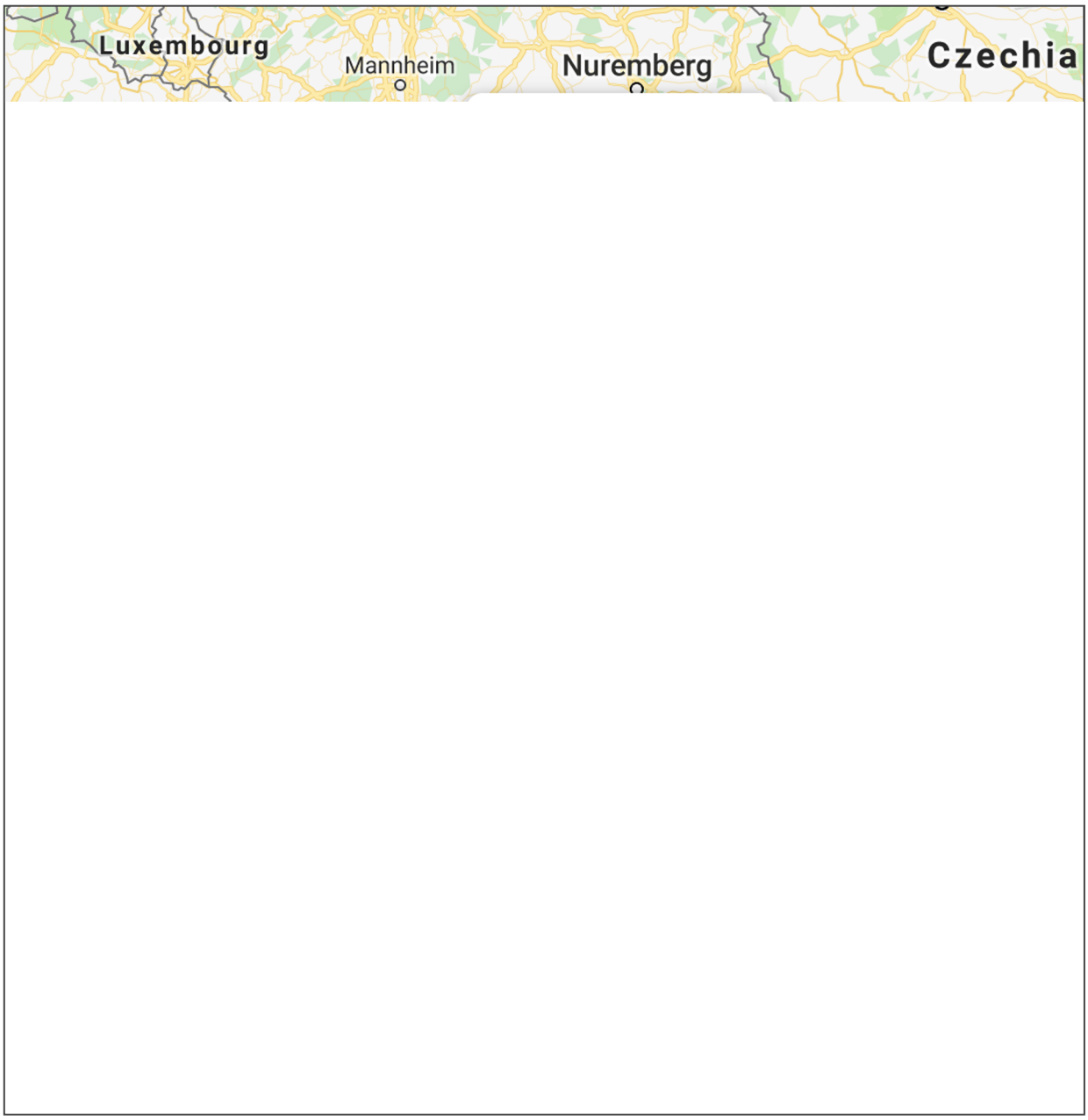
   Hint: Create a new DataFrame that has all the individual city DataFrames you created.

8. On the marker layer map, make sure each city has a pop-up marker that contains the following:
    - Hotel name
    - City
    - Country
    - Current weather description with the maximum temperature
9. Take a screenshot of the marker layer map for the route and save it as `WeatherPy_travel_map_markers.png`.

The directions layer map should look similar to the following image:



The pop-up marker for each city in the vacation itinerary look similar to the following image:



Add, Commit, Push

When you are done. Commit the following to your WeatherPy GitHub repository:

1. Your code for Parts 1, 2, and 3 of the challenge.
    1. `Weather_Database.ipynb`.
    2. `Vacation_Search.ipynb`
    3. `Vacation_Itinerary.ipynb`
2. A “data” folder containing the following CSV files.
    1. `WeatherPy_challenge.csv`
    2. `WeatherPy_vacation.csv`

> 3. An “image” folder containing the images of your maps for Parts 2 and 3.
>   1. `WeatherPy_vacation_map.png`
>   2. `WeatherPy_travel_map.png`
>   3. `WeatherPy_travel_map_markers.png`

Submission

To submit your challenge assignment, click Submit, then provide the URL of your WeatherPy GitHub repository for grading.

Rubric

Please **download the detailed rubric** to access the assessment criteria.

Note: You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Submit then indicate you are skipping by typing “I choose to skip this assignment” in the text box.

Some Rubric

Criteria	Ratings					Pts
Retrieve basic information with API call Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**10.0 pts Mastery**	**7.0 pts Approaching Mastery**	**4.0 pts Progressing**	**1.0 pts Emerging**	**0.0 pts Incomplete**	10.0 pts
Try-except block Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**20.0 pts Mastery**	**15.0 pts Approaching Mastery**	**10.0 pts Progressing**	**5.0 pts Emerging**	**0.0 pts Incomplete**	20.0 pts
Have Customers Narrow Their Travel Searches Based on Temperature and Precipitation using if/elif/else statements Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**20.0 pts Mastery**	**15.0 pts Approaching Mastery**	**10.0 pts Progressing**	**5.0 pts Emerging**	**0.0 pts Incomplete**	20.0 pts
Create a New Dataframe with Hotel Information Using Google API Places and try/except Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**10.0 pts Mastery**	**7.0 pts Approaching Mastery**	**4.0 pts Progressing**	**1.0 pts Emerging**	**0.0 pts Incomplete**	10.0 pts
Create a pop-up marker with city and weather data and hotel name Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**10.0 pts Mastery**	**7.0 pts Approaching Mastery**	**4.0 pts Progressing**	**1.0 pts Emerging**	**0.0 pts Incomplete**	10.0 pts

Criteria	Ratings					Pts
Create a directions layer map to travel between cities. Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**20.0 pts Mastery**	**15.0 pts Approaching Mastery**	**10.0 pts Progressing**	**5.0 pts Emerging**	**0.0 pts Incomplete**	20.0 pts
Create a pop-up marker for each city on the itinerary. Please see complete description, including description of Mastery levels, hyperlinked in assignment description.	**10.0 pts Mastery**	**7.0 pts Approaching Mastery**	**4.0 pts Progressing**	**1.0 pts Emerging**	**0.0 pts Incomplete**	10.0 pts
						Total Points: 100.0