

Short Rubric for WeatherPy:

Criteria	Demonstrating Proficiency 25 to > 21 points	Approaching Proficiency 21 to > 18 points	Developing Proficiency 18 to > 15 points	Emerging 15 to > 0 points	Incomplete 0 points	Pts
Deliverable 1: Retrieve Weather Data	<ul style="list-style-type: none"> ✓The deliverable fulfills the “Emerging” criteria AND the following: ✓All the weather data is added to a new DataFrame. ✓The DataFrame is exported and saved as a CSV. 	<ul style="list-style-type: none"> ✓The deliverable fulfills the “Emerging” criteria AND the following: ✓All the weather data is added to a new DataFrame. ✓Code is written to export the DataFrame as a CSV, but there is an error to save it. 	<ul style="list-style-type: none"> ✓The deliverable fulfills the “Emerging” criteria AND the following: ✓Most of the weather data is added to a new DataFrame. 	<ul style="list-style-type: none"> ✓All of the required data from the API is retrieved. 		25.0
	Demonstrating Proficiency 35 to > 32 points	Approaching Proficiency 32 to > 27 points	Developing Proficiency 27 to > 24 points	Emerging 24 to > 0 points		
Deliverable 2: Create a Customer Travel Destinations Map	<ul style="list-style-type: none"> ✓Input statements are written to get the minimum and maximum temperature. ✓A new DataFrame is created based on the weather criteria, and empty rows are dropped. ✓The hotel name is added to the DataFrame, and the empty rows are dropped. ✓The DataFrame is exported and saved as a CSV file. ✓A marker layer map is created with a pop-up marker for each city that has all the correct data. ✓The marker layer map is saved as a PNG 	<ul style="list-style-type: none"> ✓Input statements are written to get the minimum and maximum temperature. ✓A new DataFrame is created based on the weather criteria, and empty rows are dropped. ✓The hotel name is added to the DataFrame, and the empty rows are dropped. ✓The DataFrame is exported and saved as a CSV file. ✓A marker layer map is created with a pop-up marker for each city, but some cities don't have all the data. ✓The marker layer map is saved as a PNG 	<ul style="list-style-type: none"> ✓Input statements are written to get the minimum and maximum temperature. ✓A new DataFrame is created based on the weather criteria, and empty rows are dropped. ✓The hotel name is added to the DataFrame, but the empty rows are not dropped. ✓The DataFrame is exported and saved as a CSV file. ✓A marker layer map is created with a pop-up marker for each city, but some cities don't have all the data. ✓The marker layer map is saved as a PNG 	<ul style="list-style-type: none"> ✓Input statements are written to get the minimum and maximum temperature. ✓A new DataFrame is created based on the weather criteria, but the empty rows are not dropped. ✓The hotel name is added to the DataFrame, but the empty rows are not dropped. ✓The DataFrame is exported and saved as a CSV file. ✓A marker layer map is created with a pop-up marker for each city, but some cities don't have all the data. ✓The marker layer map is saved as a PNG 		35.0

	Demonstrating Proficiency 40 to > 36 points	Approaching Proficiency 36 to > 34 points	Developing Proficiency 34 to > 31 points	Emerging 30 to > 0 points		
Deliverable 3: Create a Travel Itinerary Map	<p>✓Four DataFrames are created, one for each city in the itinerary.</p> <p>✓The latitude and longitude pairs for each city are retrieved to create the directions layer map.</p> <p>✓A directions layer map between the cities and the travel map is uploaded as a PNG.</p> <p>✓A DataFrame that contains the four cities on the itinerary is created.</p> <p>✓A marker layer map with a pop-up marker for the cities in the itinerary is created, and is uploaded as a PNG.</p>	<p>✓Four DataFrames are created, one for each city in the itinerary.</p> <p>✓The latitude and longitude pairs for each city are retrieved to create the directions layer map.</p> <p>✓There is a directions layer map between THREE of the FOUR cities, and the travel map is uploaded as a PNG.</p> <p>✓A DataFrame that contains the four cities on the itinerary is created.</p> <p>✓A marker layer map with a pop-up marker for the cities in the itinerary is created, and is uploaded as a PNG.</p>	<p>✓Four DataFrames are created, one for each city in the itinerary.</p> <p>✓The latitude and longitude pairs for each city are retrieved to create the directions layer map.</p> <p>✓There is a directions layer map between TWO of the FOUR cities, and the travel map is uploaded as a PNG.</p> <p>✓A DataFrame that contains the four cities on the itinerary is created.</p> <p>✓A marker layer map with a pop-up marker for the cities in the itinerary is created, and is uploaded as a PNG.</p>	<p>✓Four DataFrames are created, one for each city in the itinerary.</p> <p>✓Code is written to retrieve the latitude and longitude pairs for each of the four cities.</p> <p>✓Code is written but a directions layer map isn't created between the cities.</p> <p>✓A DataFrame that contains the four cities on the itinerary is created.</p> <p>✓A marker layer map with a pop-up marker for the cities in the itinerary is created, and is uploaded as a PNG.</p>		40.0