This tutorial will show you how to create animations using Evapotranspiration data from ECOSTRESS using Python on MacOS. Outputs will include GIF and MP4 files, with the option to standardize the values.

# Table of Contents

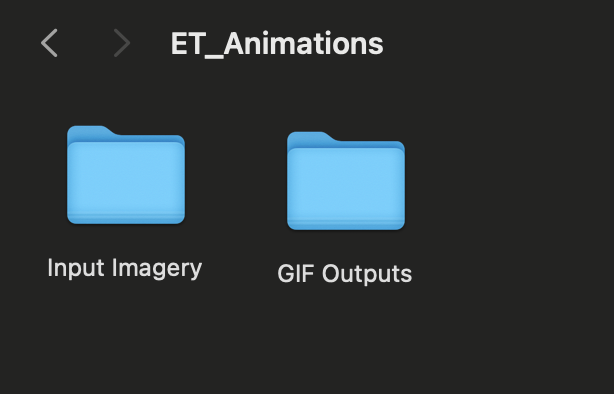
Create ET GIF

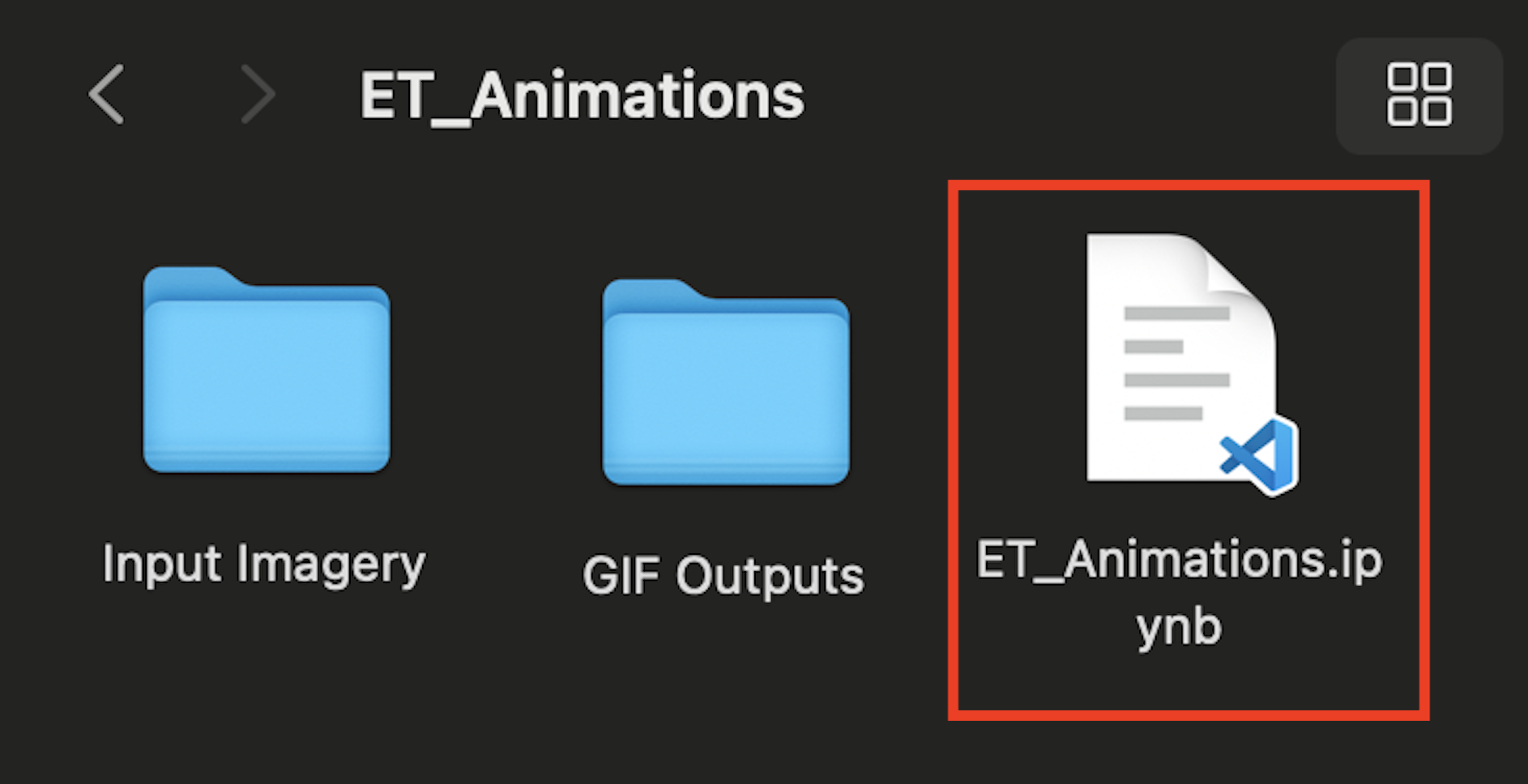
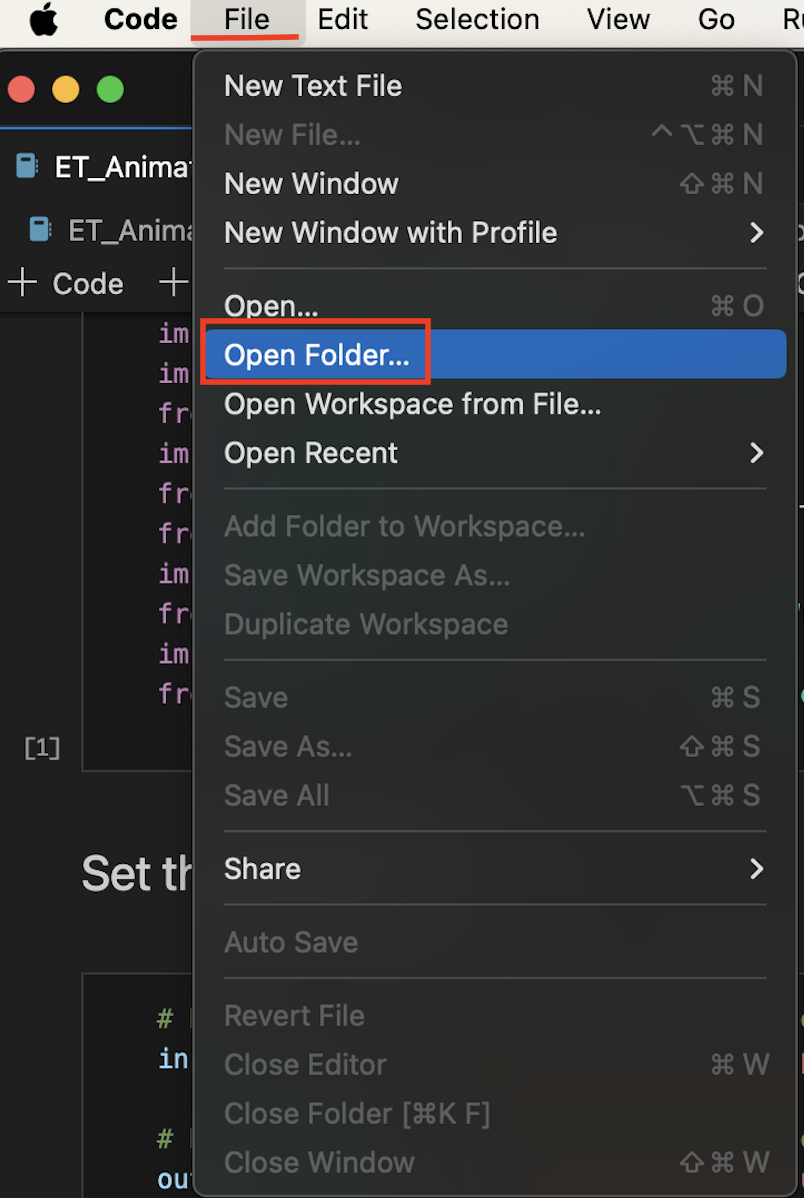
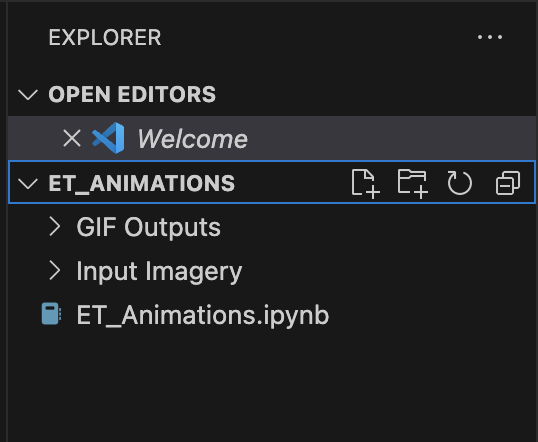
Creating Animated GIF

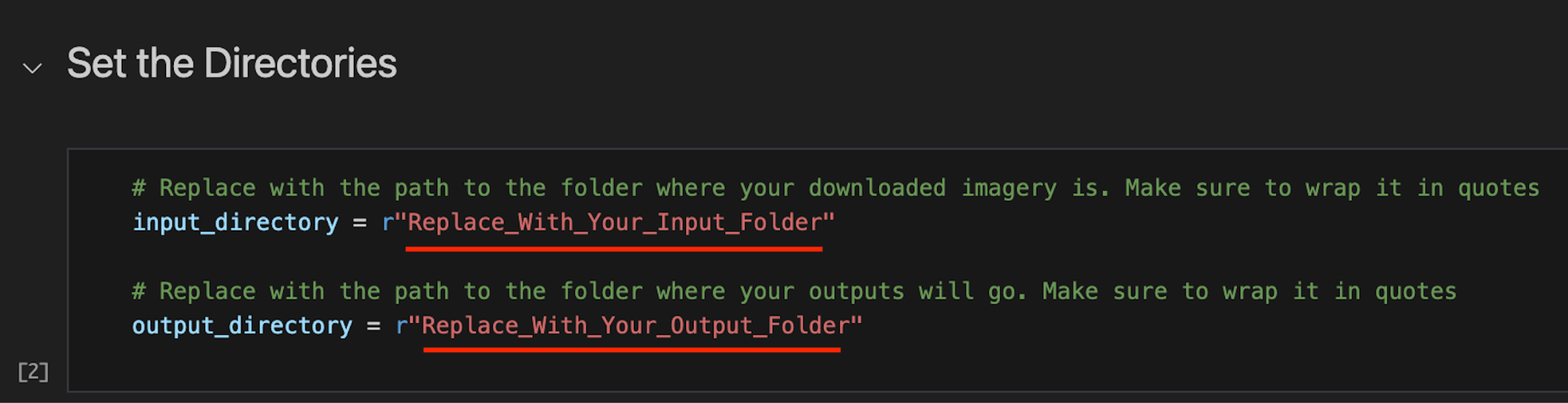
Animations with Standardized Values

Create ET GIF

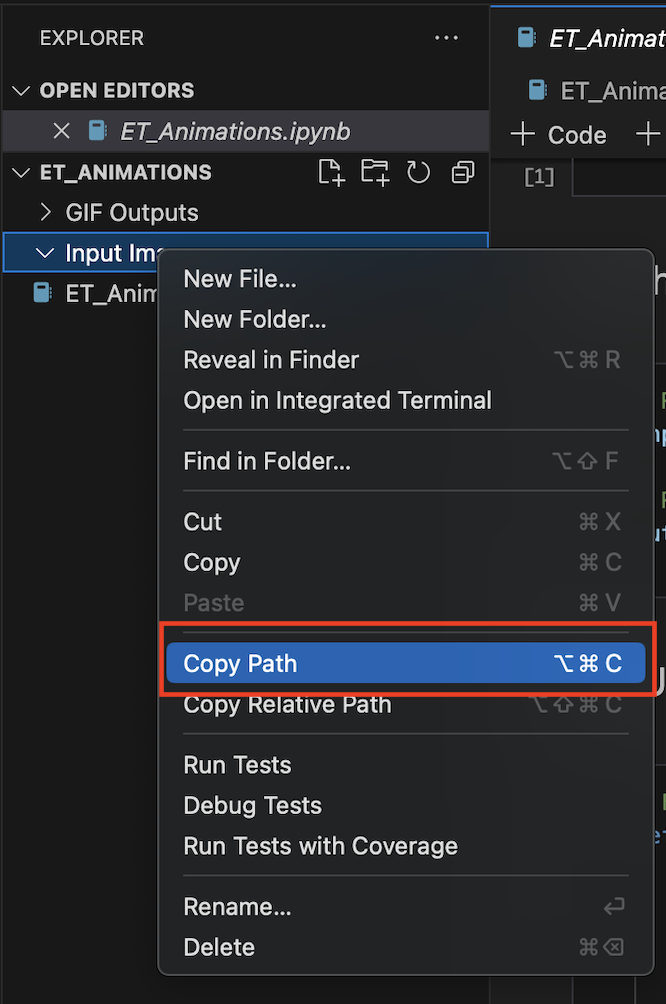
1. Download **ET Animations** code from <https://github.com/ECOSTRESS-Tutorials/ECOSTRESS-ET-Animations>.
2. Open your Finder. Create a project folder called “**ET\_Animations**” with a folder within called “**Input Imagery**” to hold your ECOSTRESS ET images, and a subsequent output folder called “**GIF Outputs**” to store the GIF outputs you will create.



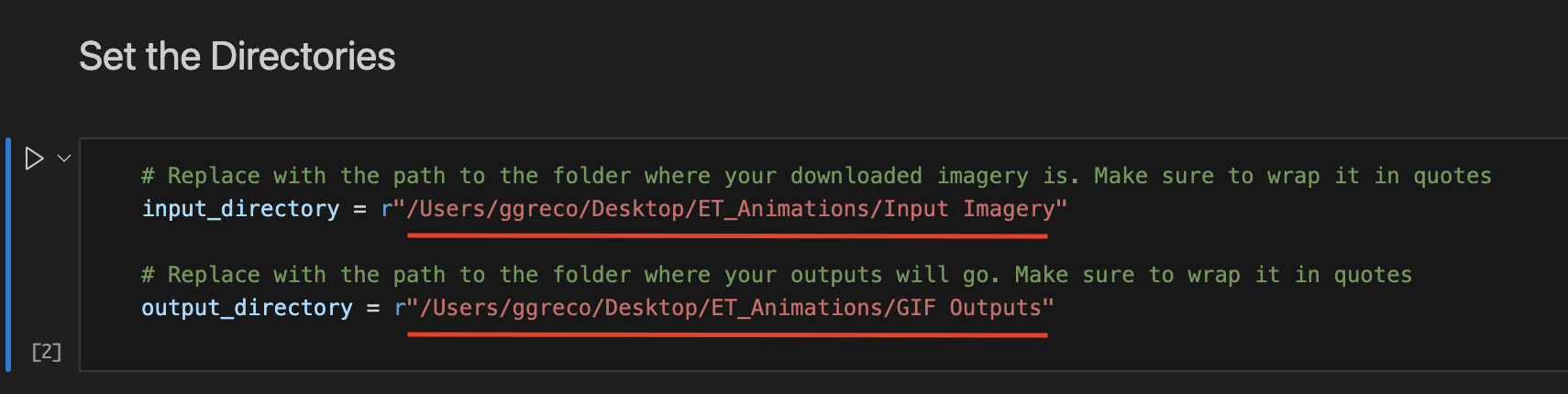
1. Move the downloaded ET Animations code to the “ET\_Animations” project folder.
2. Open **Visual Studio Code** and use **File > Open Folder…** to connect to the main project folder containing your imagery and animation code.
3. You’ll see the **ET\_Animations.ipynb** code under the **Explorer** tab. Click on it to open. (**Tip**: If you want to know more about what each line of the code does, read the **comments** in the code. Comments in the code are identified by **#**. These comments do not actually change how the code runs, but they can be helpful to put notes on how the code works for yourself or other users. This can also be helpful if you want to customize the code because it will guide you to which parts you may want to change!)
4. Locate the second block of code, labeled **Set the Directories**. For the variable that says “Replace\_With\_Your\_Input\_Folder”, change it to reflect the actual path of the Input Imagery folder you created that is holding the downloaded ET ECOSTRESS imagery. Next, for the output directory, replace the variable with the actual path to the GIF output folder you created.



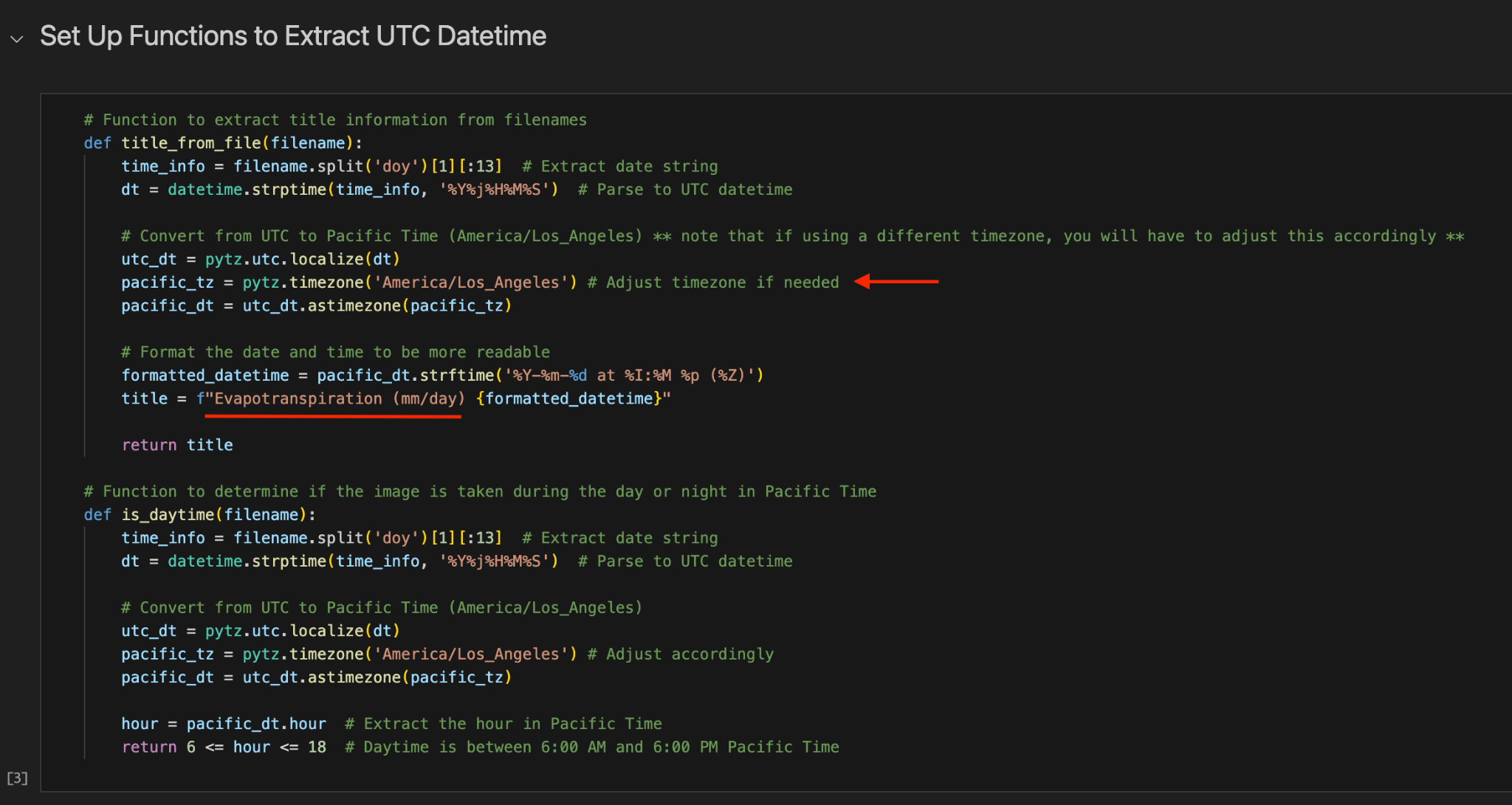
To **copy the folder path**, use the **EXPLORER** panel on the left side of Visual Studio Code to find the folder you are interested in. Once you have found it, **right click** on it and select **Copy Path**. Now you can paste the path into your code. Make sure it is still **wrapped in quotes** and has **r** outside the first quote.



Paste the folder path into the **input\_directory** quotes. Do the same for the **output\_directory**.



1. For the “Set Up Functions to Extract UTC Datetime” code cell, leave it as is *except* if you wish to change the Timezone and the units that the ET data uses.



1. For the Timezone, adjust as needed from PT timezone. To find the nomenclature for your timezone, you can search ‘List of timezones using python pytz’ to find yours.
2. For the ET units, the animation code is set up to use ET daily measurements. If you are using the ET instantaneous dataset, then change the units to be **W/m²**. NOTE: if you make this change, you will have to change the units throughout the entire code.

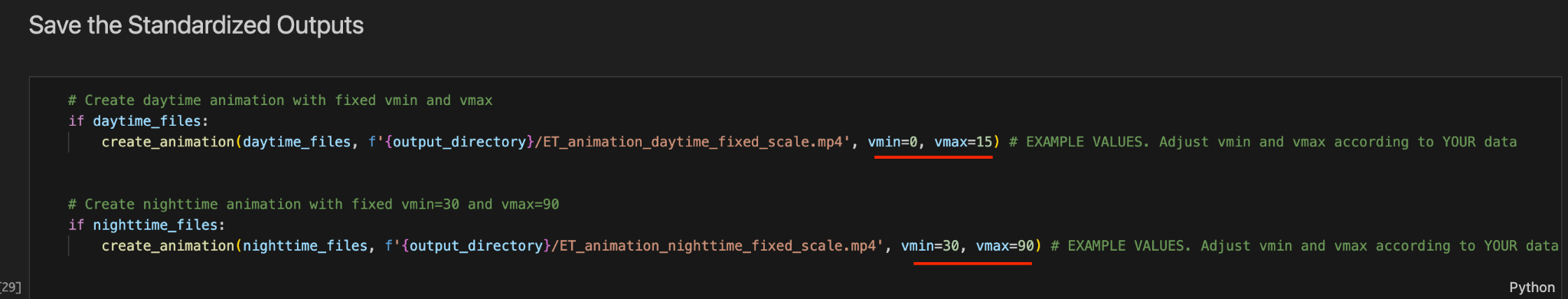
Leave the “Identify if Sharpened Image is Daytime or Nighttime,” “Make a Helper Function to Create the GIF,” and “Save the Outputs” cells as is.

Creating Animated GIF

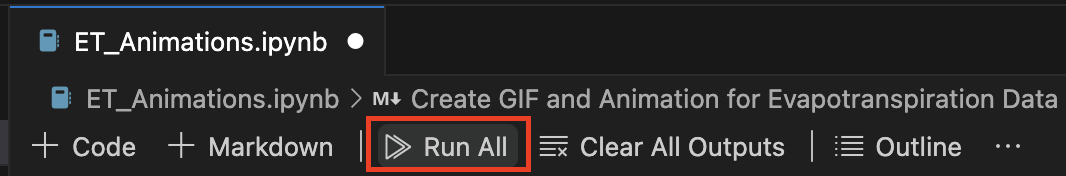
1. The only adjustments you will need to make in this part of the code are either the Timezones or ET units, if you did in the previous portion. Make sure it reflects the changes above, if it applies. If you did not adjust anything then you can keep these cells as is.

Animations With Standardized Values

1. Again, if you make adjustments to the Timezones or ET units in the previous portion, make sure it reflects the changes above, if it applies. If you did not adjust anything then you can keep these cells as is.
2. In the “Save the Standardized Outputs”, make sure to adjust the min and max values according to YOUR data. This is very important as not every timeseries will have the same vales. You can do this by looking at the GIF or non-standardized MP4 outputs and manually find the maximum ET value and the minimum ET value.



Run all cells by clicking Run All at the top of the VS Code page.



You should now see a series of outputs for both daytime and nighttime imagery:

1. GIF
2. Animated GIF
3. Animated GIF with standardized values