Guide to Using Arduino To Measure Battery Discharge Curve

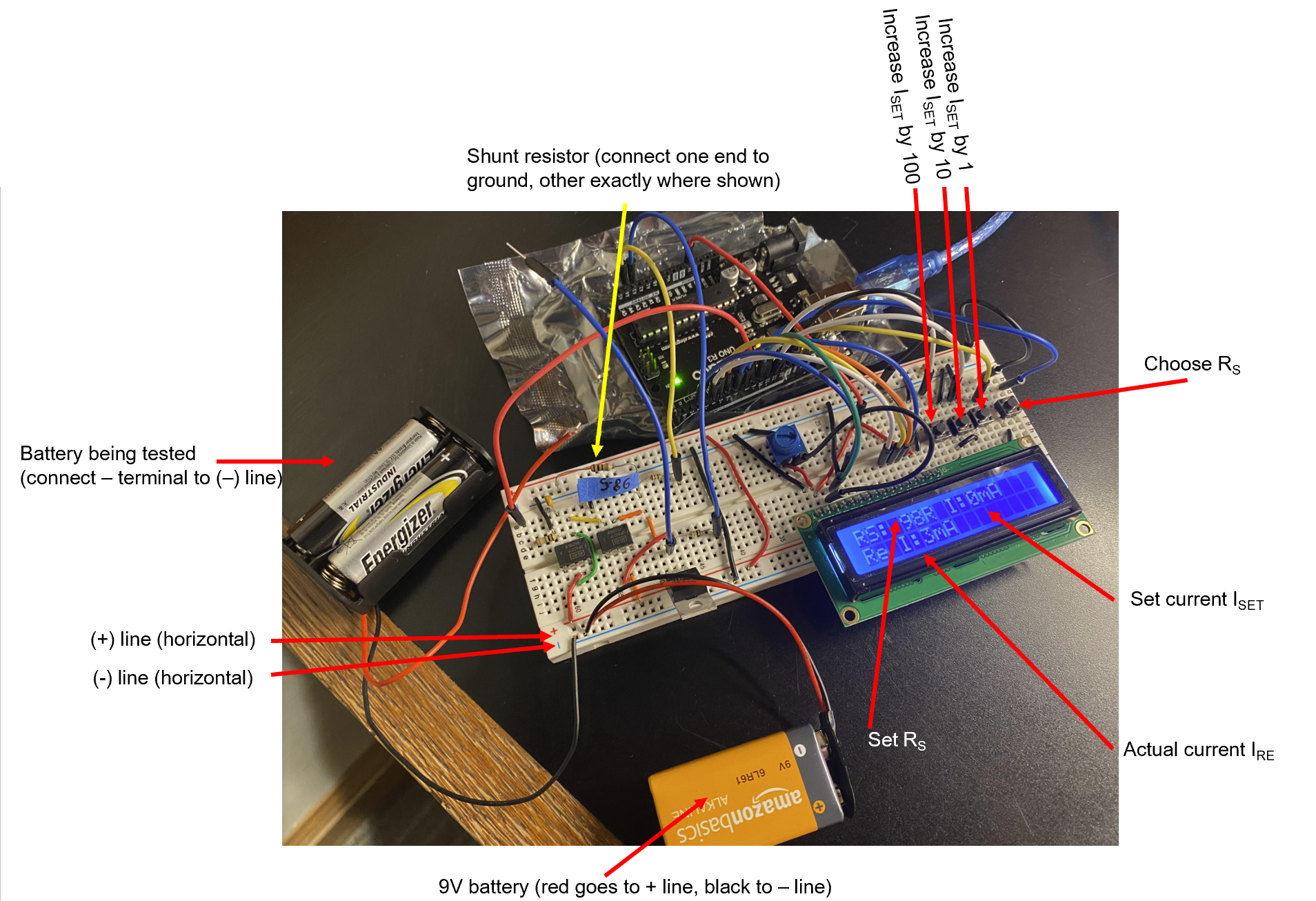
Abrar Sheikh, February 21st 2023

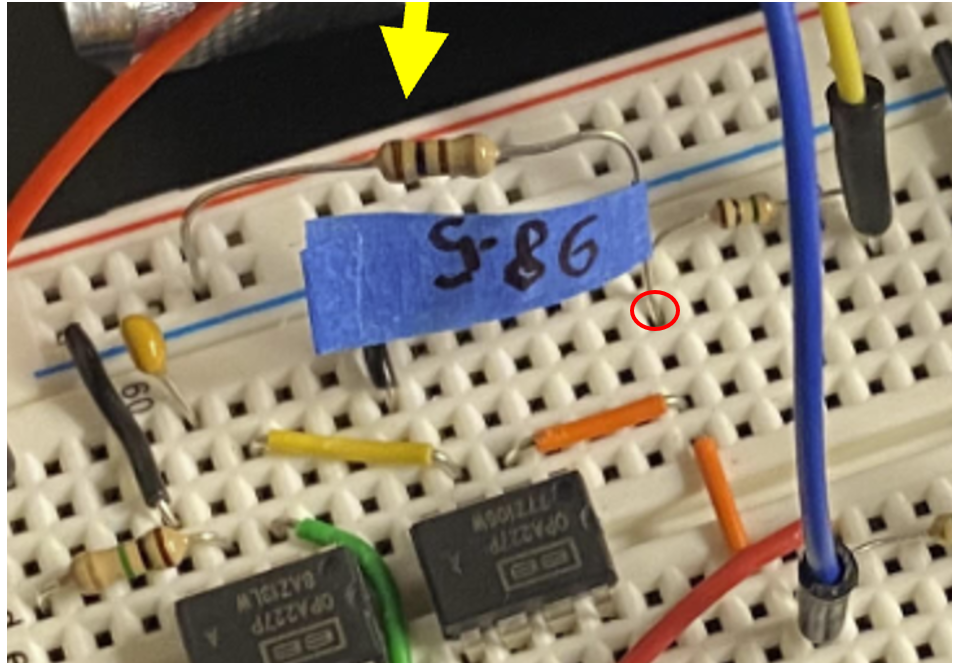
**Note: you can turn off circuit just by unplugging Arduino from computer.**

Also note that Arduino should be placed on top of a piece of paper or a **non-conducting** surface (otherwise it may short-circuit and be damaged).

**Wiring Arduino and Circuit:**

1. Note that this circuit can only measure up to 5V. Please **DO NOT** try to connect more than 5V to the power supply (the Arduino could get damaged). Instead, if it is required, ask Abrar to adjust the circuit.
2. Choose your desired discharge current ISET and determine which shunt resistor is appropriate. Your six choices are RS = 2.0 ohm (max 2500 mA), 5.2 ohm (max 865 mA), 10.1 ohm (max 445 mA), 46.6 ohm (max 96 mA), 98.6 ohm (max 45.63 mA), or 997 ohm (max 4 mA).
   1. Caution: for currents > 100 mA, transistor (with large metal flap) and shunt resistor get very hot!
   2. Above 800 mA smell of burning plastic is noticeable- at most we will discharge 650 mA since that is the measured current consumption of the rover
3. Wire up the circuit as below (note that there are two (+) and (-) lines- both are equivalent to each other.)



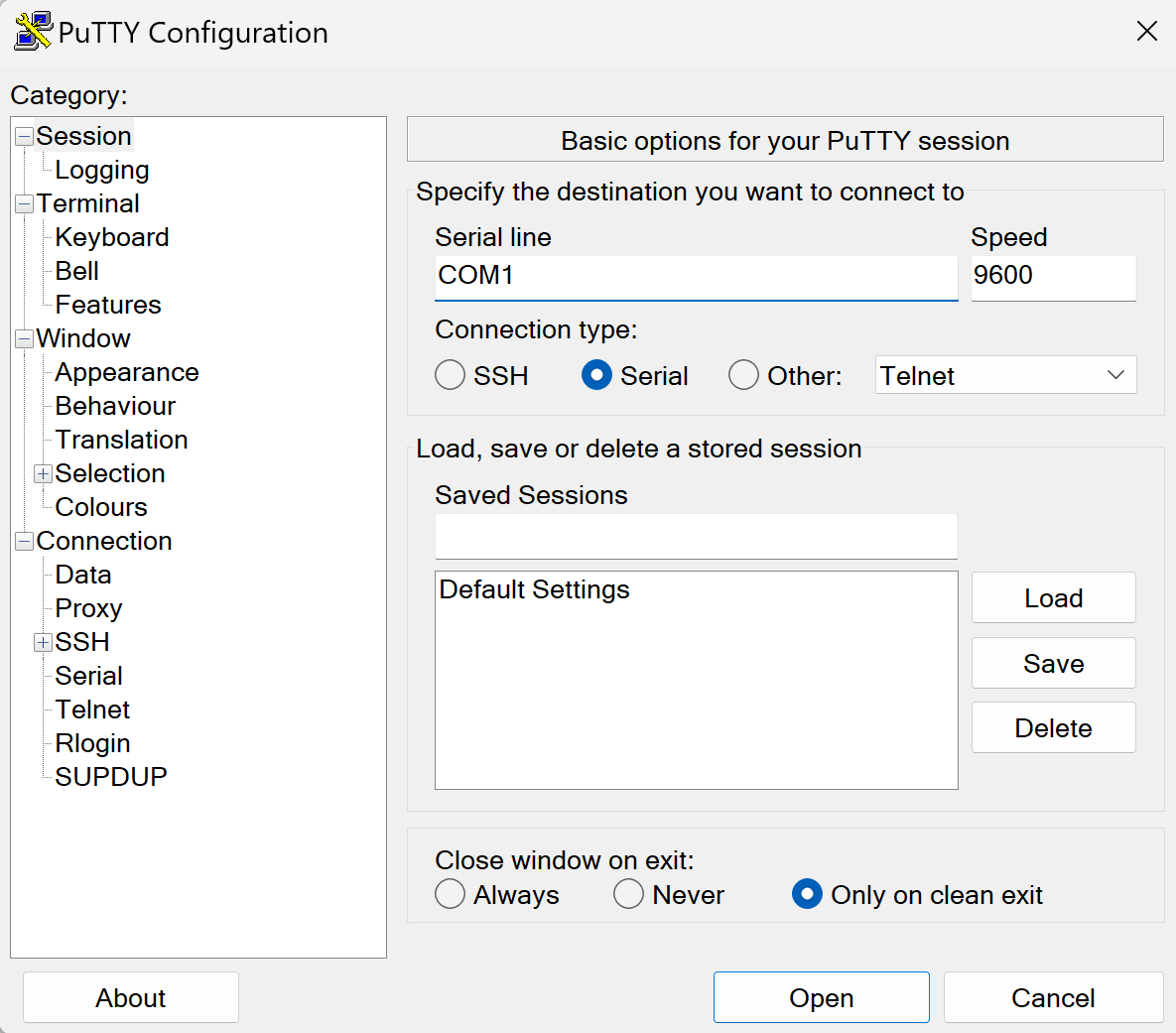


**Using Device to Set Current:**

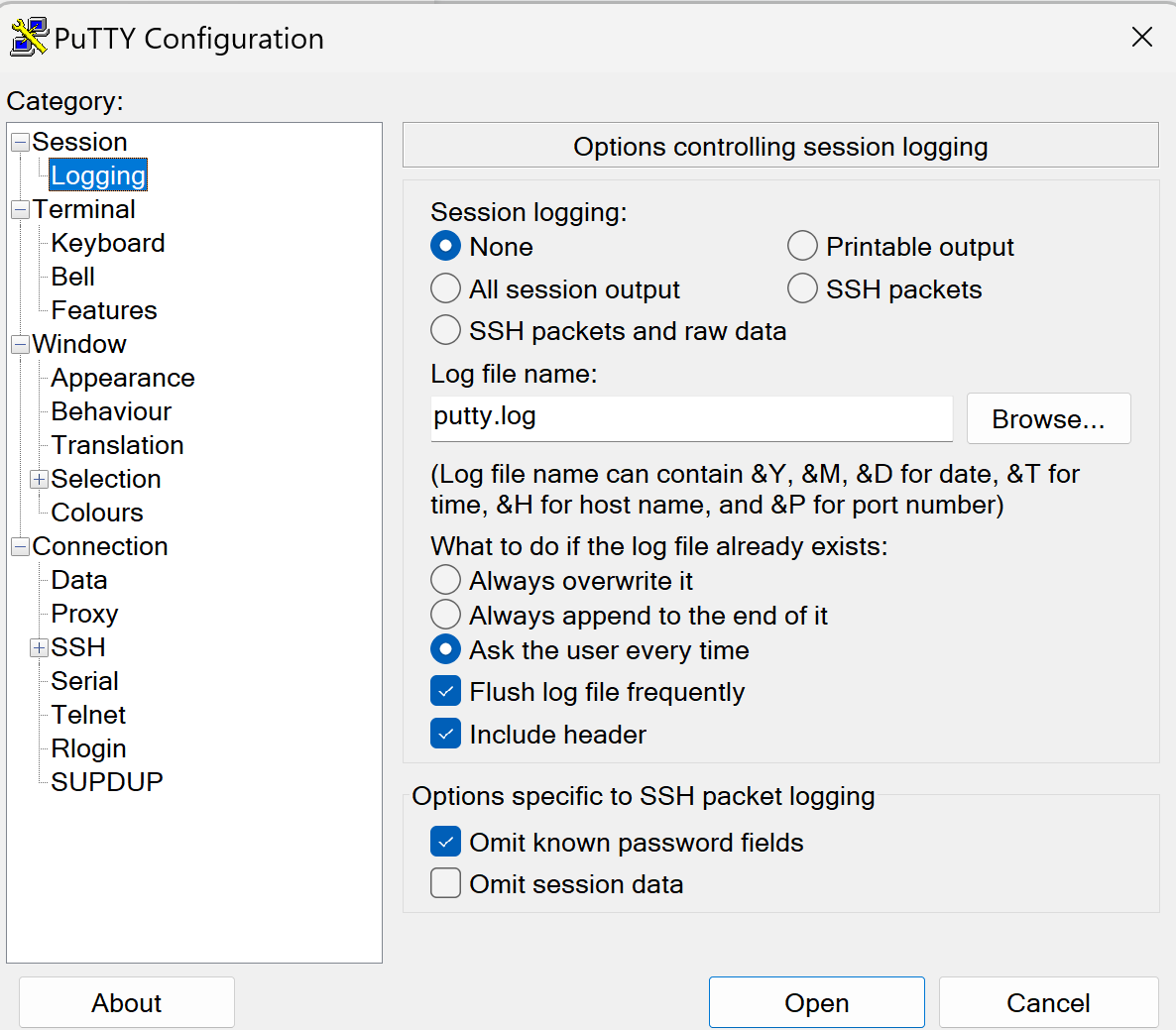
1. Disconnect the alligator clip connection for now.
2. Press the rightmost button until RS reads on the screen.
3. The left three buttons adjust current as follows:
   1. Left most adds 100 mA
   2. Middle adds 10 mA
   3. Third adds 1 mA
4. Press buttons as needed to get to set current. If you overshoot, just keep pressing the same button until it goes back to 0.

**Procedure to use PuTTy to measure voltage:**

1. Open up PuTTy. Under “Connection Type” click “Serial”.
2. Type “COM(number)” into “Serial Line” bar, replacing “(number)” with appropriate number.



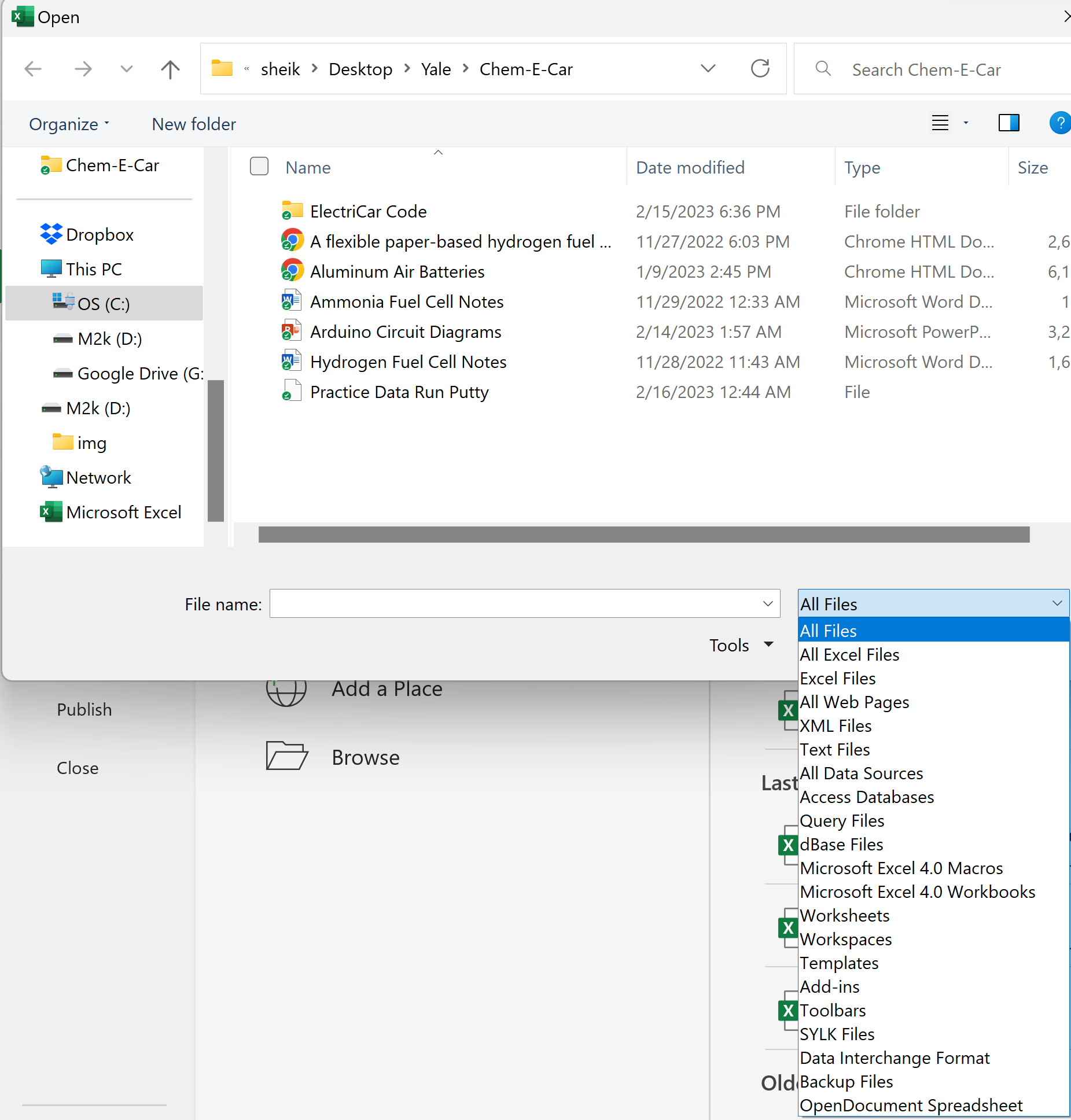
1. Click on “Logging” and under “Session Logging” click on “All Session Output” radio button.



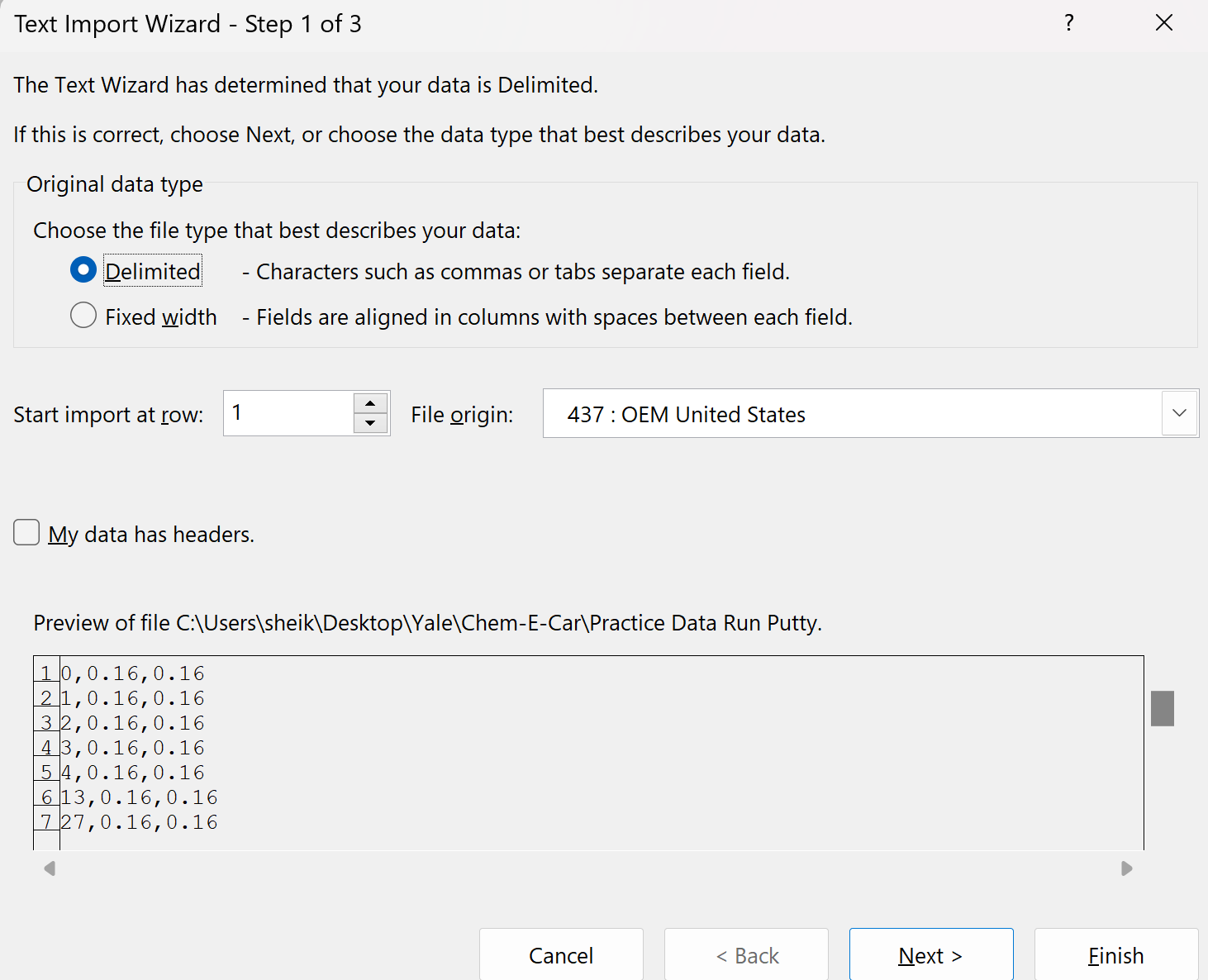
1. Select “Browse” button, navigate to directory **where JupyterLab data analysis file is**, and save with name of your choice.
2. Click “open” when ready. You should see a black screen where data is being logged.
3. At the same time, reconnect alligator clip.
4. Close window when finished- data should be automatically saved.

**Procedure to convert data into CSV File:**

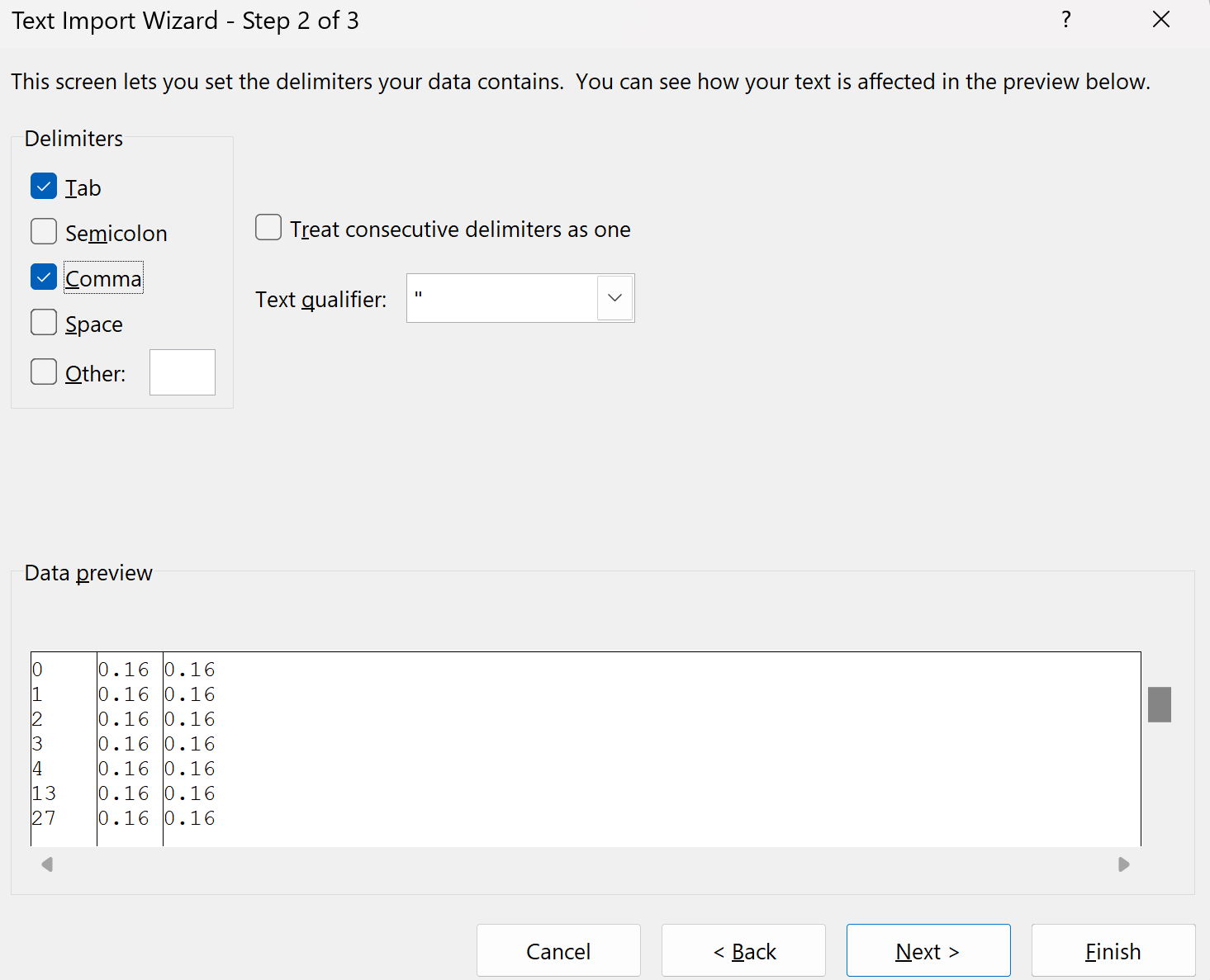
1. Find file where data was saved. Open it up using NotePad, and delete first line (as well as any repeated 0 lines).
2. Scroll down to bottom of file. Delete any incomplete lines.
3. Save and close file.
4. Open up the data file in Excel. Make sure that when searching for files, you have the “All Files” option chosen.



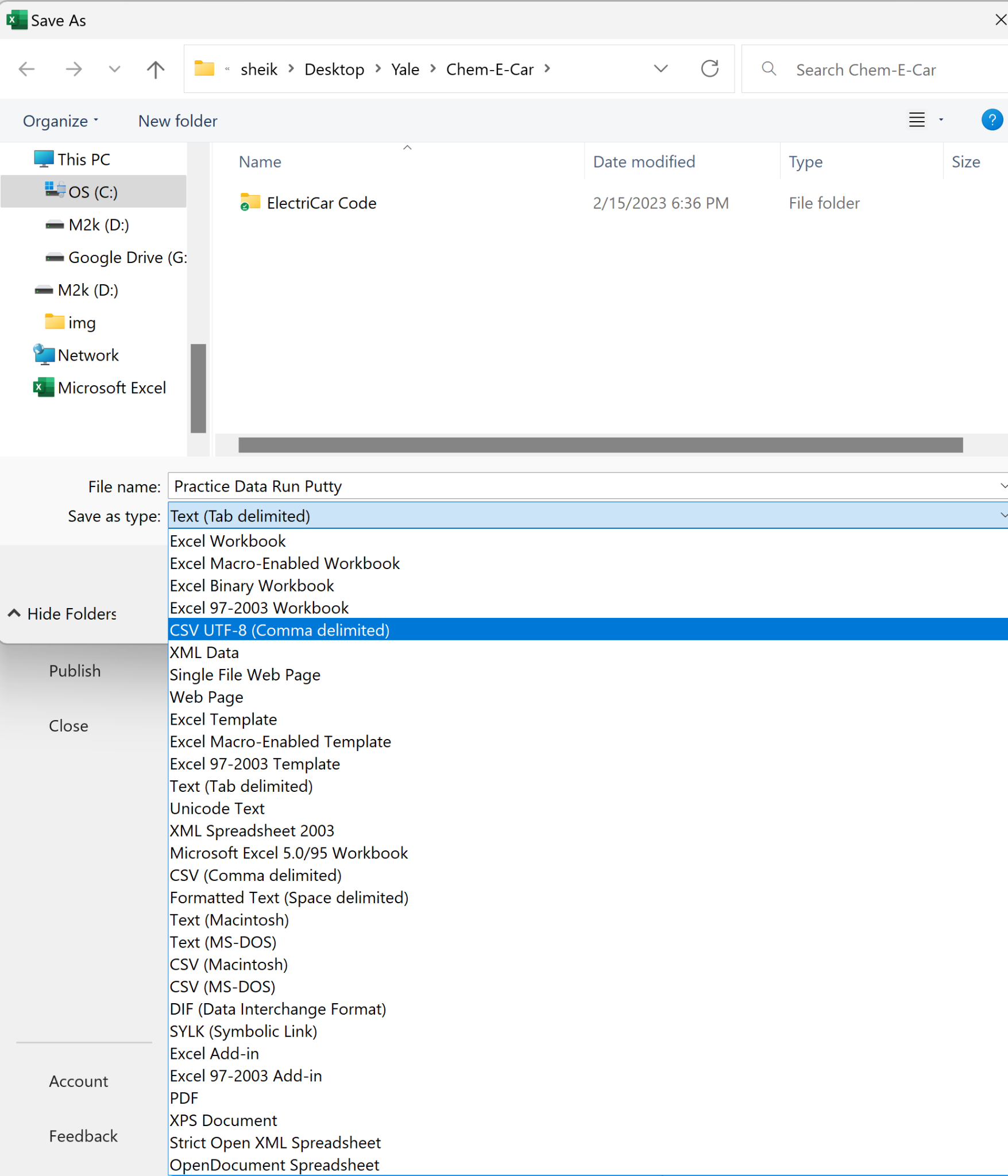
1. When they warn about incompatibility, say “Yes” you are sure, and then they will take you to a wizard. Make sure that you click the radio button with “Delimited”.



1. Now click “Next”, and make sure to select “Tab” and “Comma” as delimiters in Step 2. Click “Finish” once done.

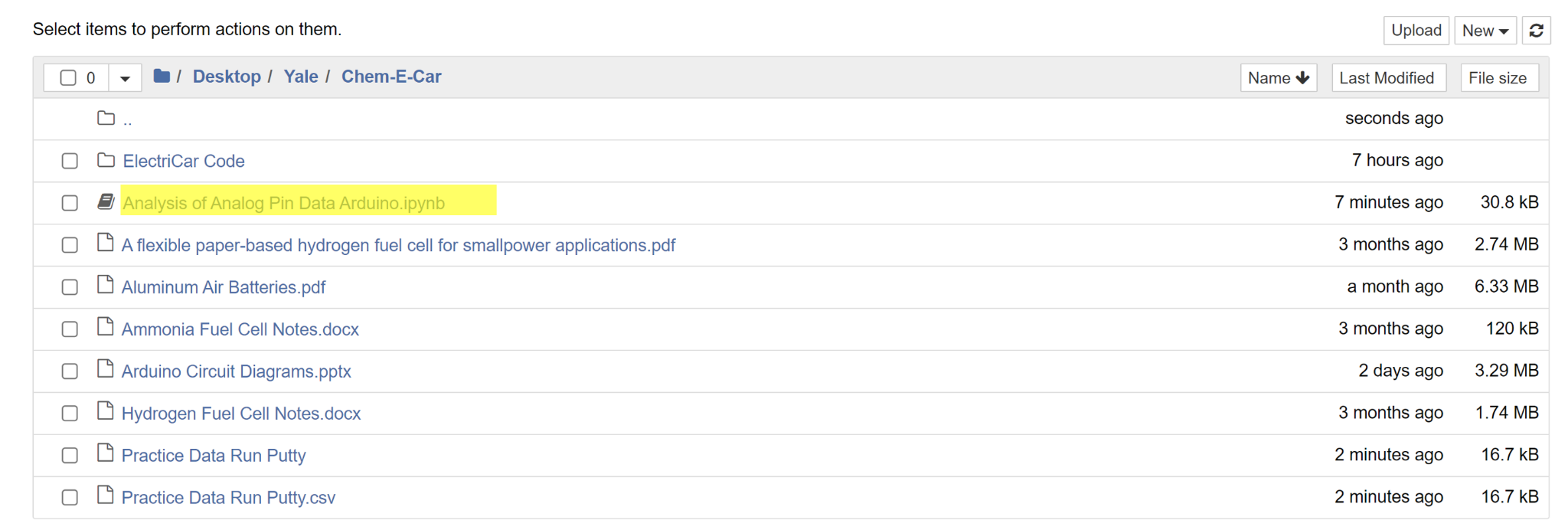


1. Save file as a CSV UTF-8 (Comma delimited) file, as shown below. Make sure this is in same directory as the analysis code (else the code will not work).

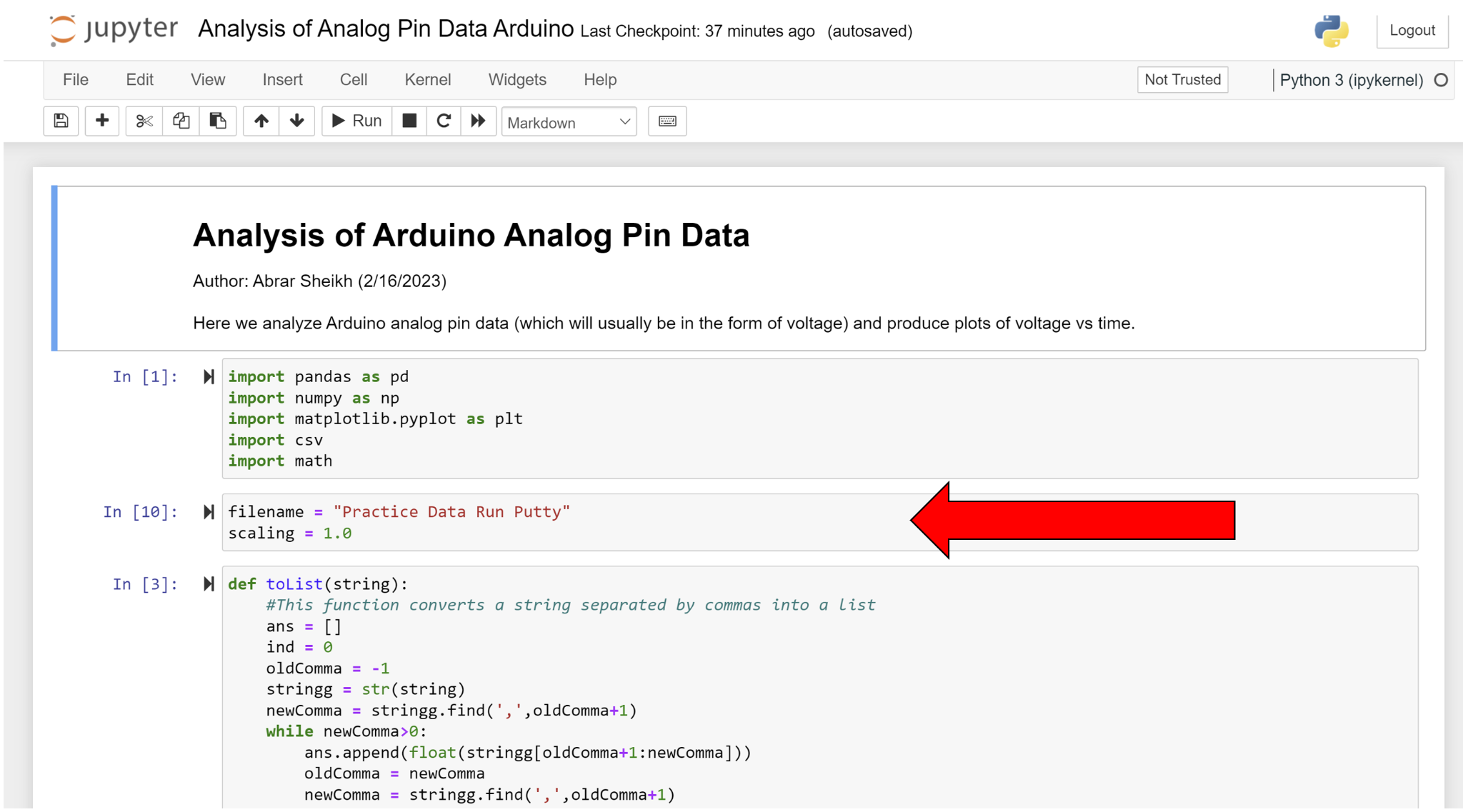


**Data Analysis:**

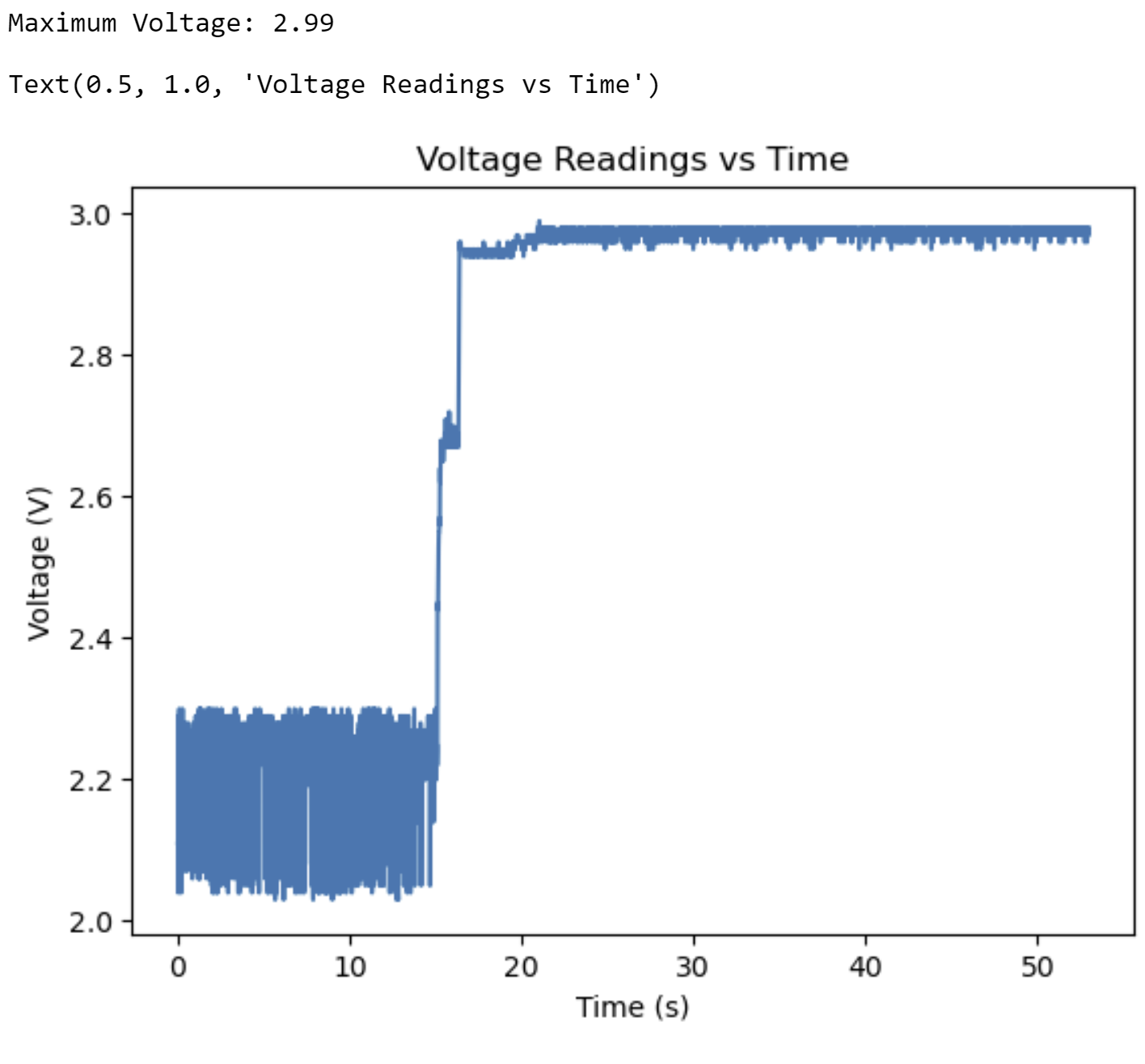
1. Type “Jupyter Notebook” into search bar, and let it open up (it should open in Chrome if using Abrar’s laptop).
2. Click on “Desktop” then “Yale” then “Chem-E-Car” then “ElectriCar” and finally open “Analysis of Analog Pin Data Arduino.ipynb”.



1. Adjust the filename as appropriate in the file. Also adjust the scaling factor as determined in the first step.



1. Click on the first cell. Press and hold SHIFT while repeatedly pressing ENTER.
2. You should get a graph like the below:



(The source connected was 2 AA batteries being discharged at 45 mA with shunt resistance 98 ohm- I’m pretty sure the first 15 seconds are just noise).