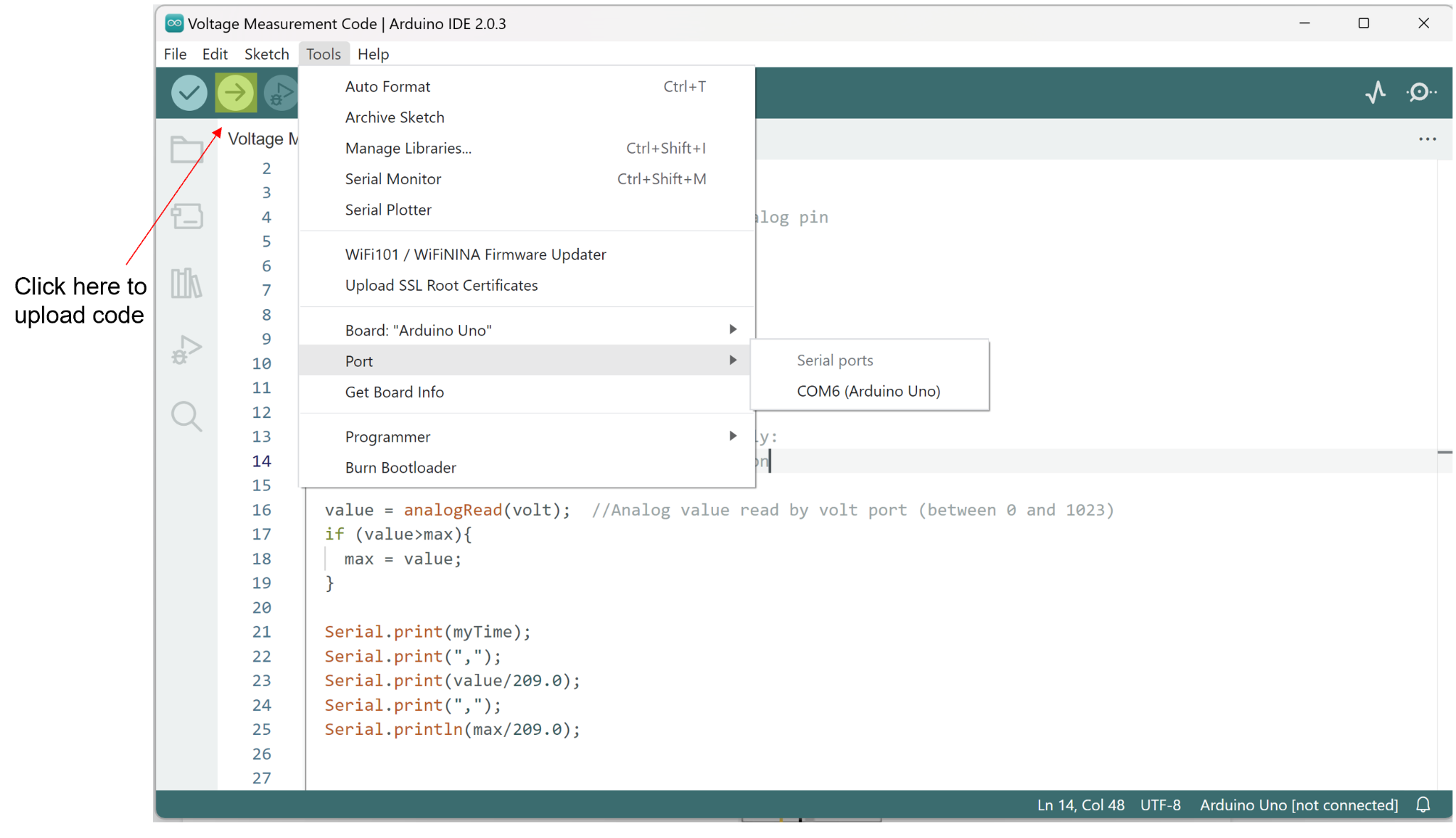
Guide to Using Arduino To Measure Voltage

Abrar Sheikh, February 16th 2023

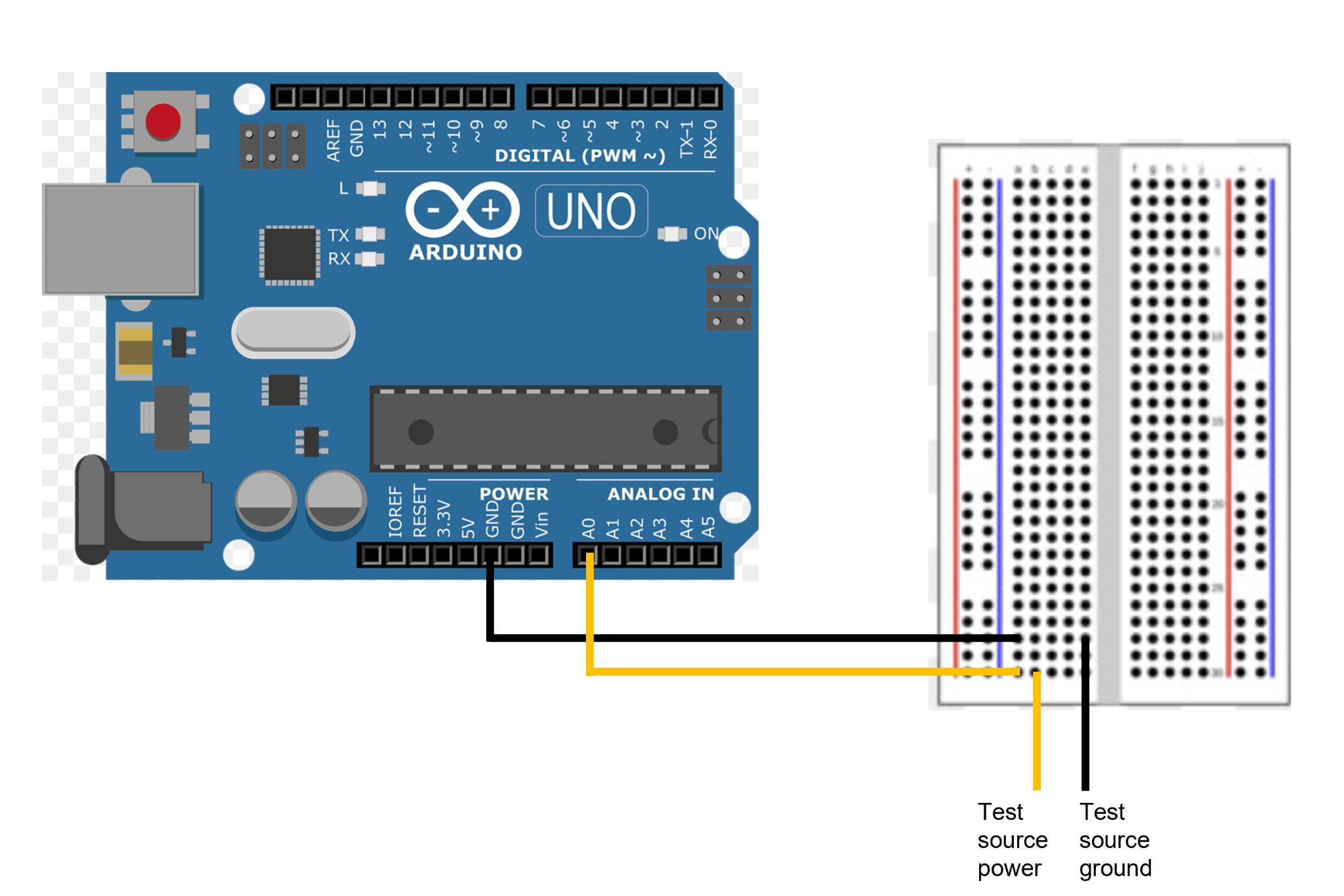
**Note: you can turn off circuit just by unplugging Arduino from computer.**

**Wiring Arduino and Circuit:**

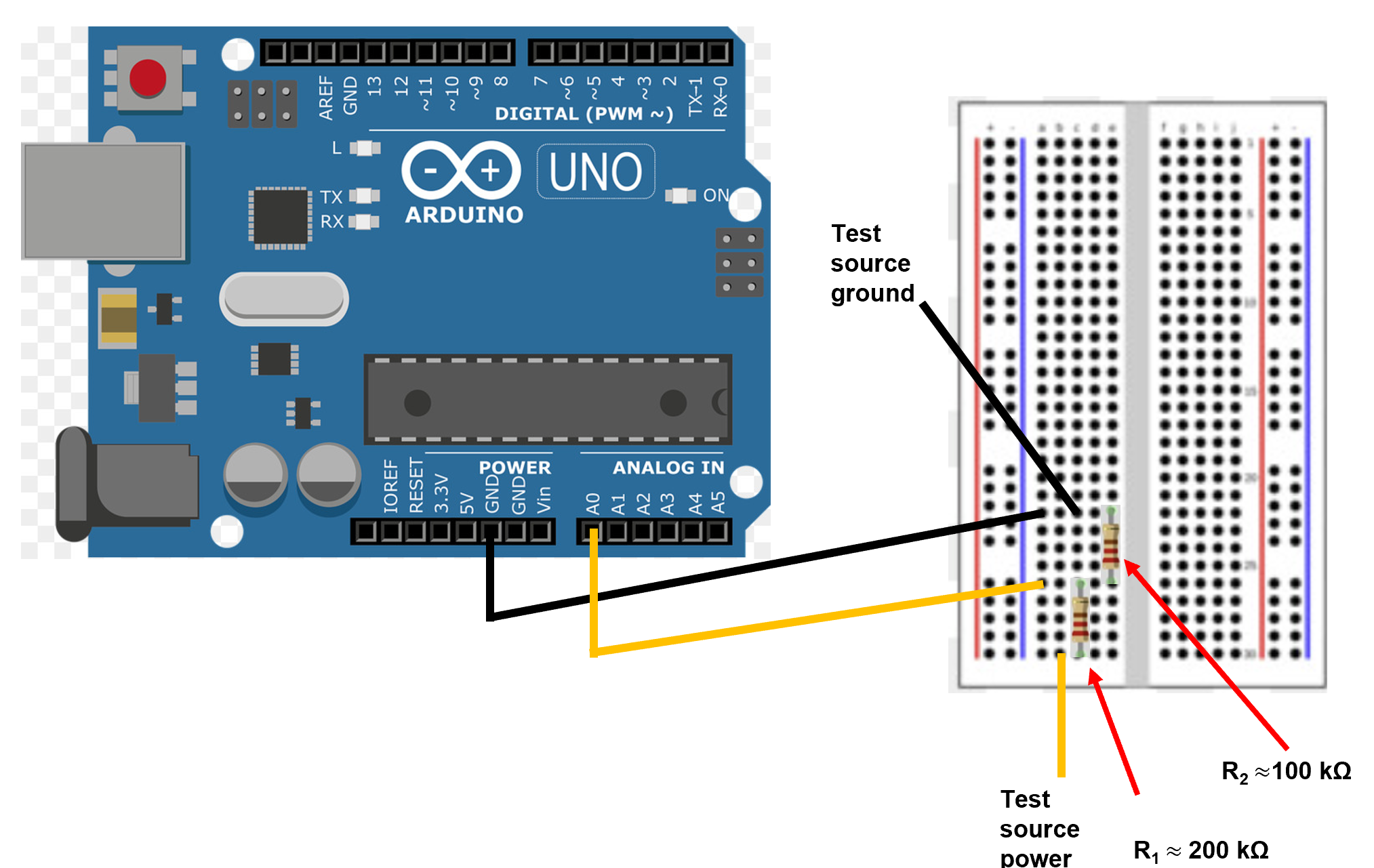
1. **If you are doing Iodine clock, disregard steps 2 and beyond for this section, and follow the instructions below.** Your scaling factor will be 1.
   1. Place rover on a raised platform (so it won’t move if its motors turn on).
   2. You won’t need to adjust any wiring. Select the port your Arduino is connected to as shown below and then upload the “Voltage Measurement Code” onto the Arduino board by clicking the highlighted button.



1. Follow the circuit diagram below if your expected voltage is less than 5 volts. In this case, your scaling factor will be 1. Make sure that Arduino is connected to your laptop (cables are in CEID on the table where Chem-E-Car keeps its stuff)!



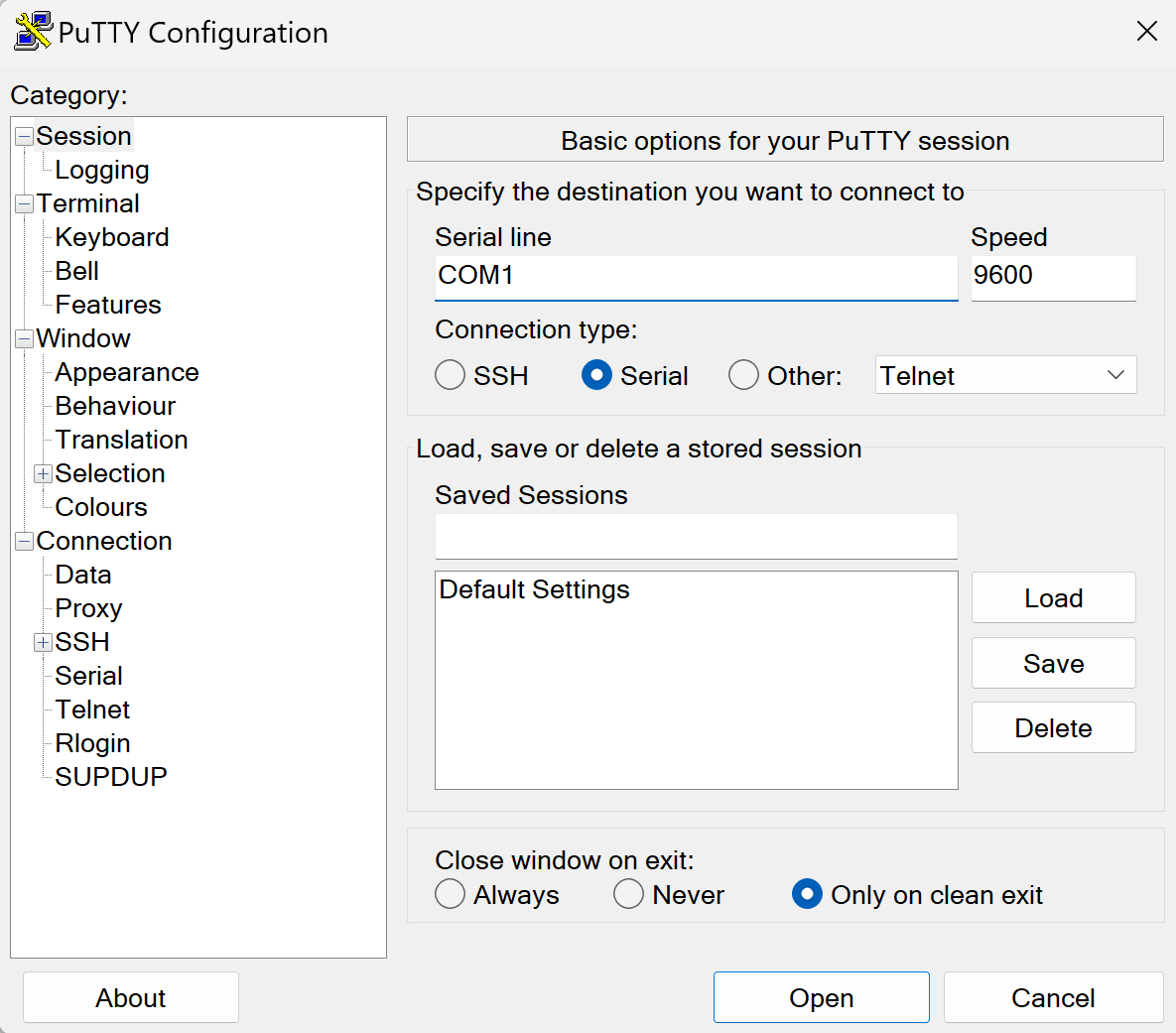
1. If your expected voltage is more than 5 volts and less than 15 volts, you MUST follow the diagram below. Please measure the resistances R1 and R2 to get their actual resistances, and your scaling factor will be



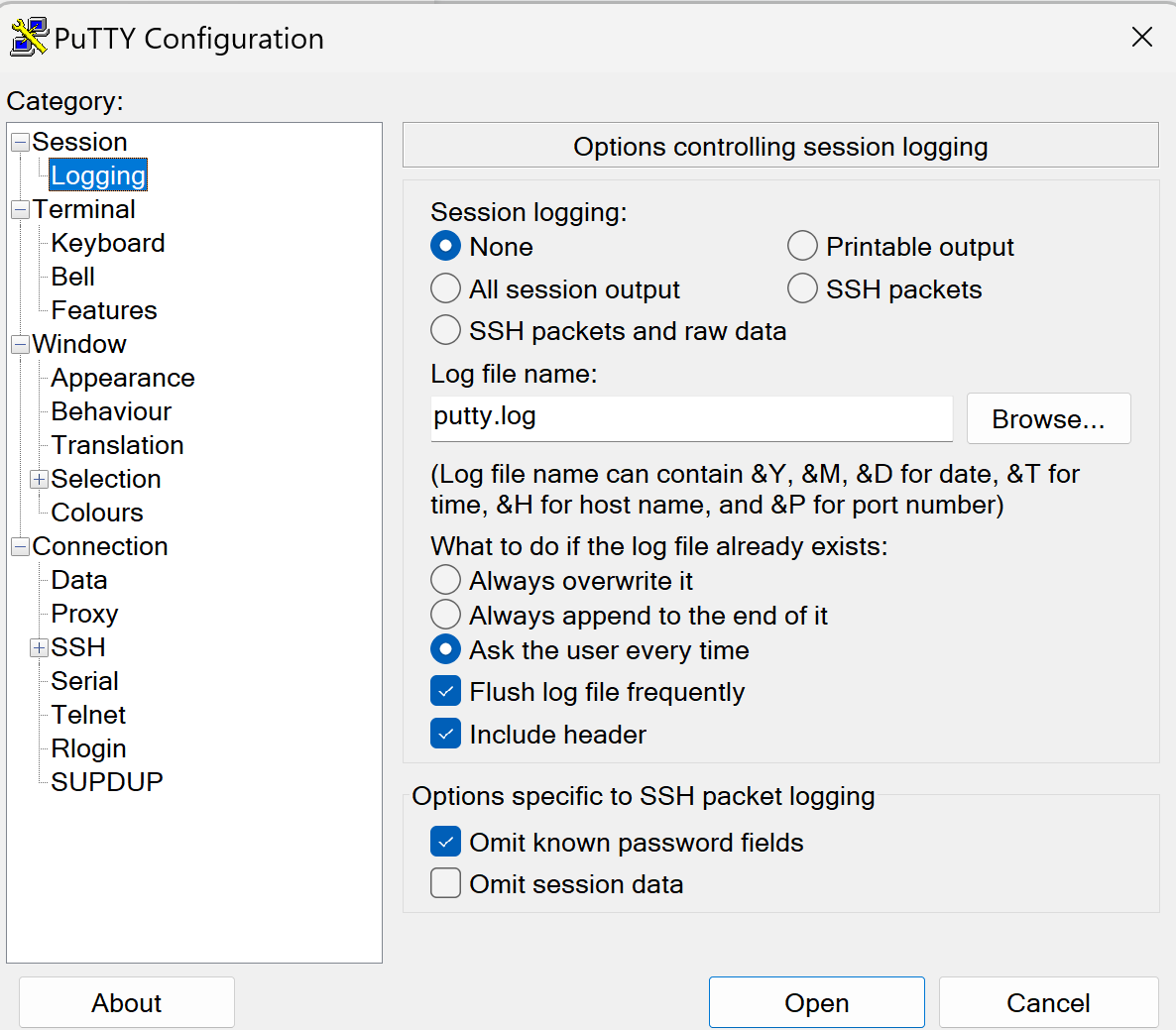
1. If voltage is greater than 15 volts, make R2 10 times larger than R1 and proceed as above.
2. Navigate to Desktop/Yale/Chem-E-Car/ElectriCar Code/Voltage Measurement Code and click on INO file within the folder.
3. Plug in Arduino board to USB port on laptop, and connect board in IDE. Note COM port that it is connected to.
4. Upload code to Arduino Board.

**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.



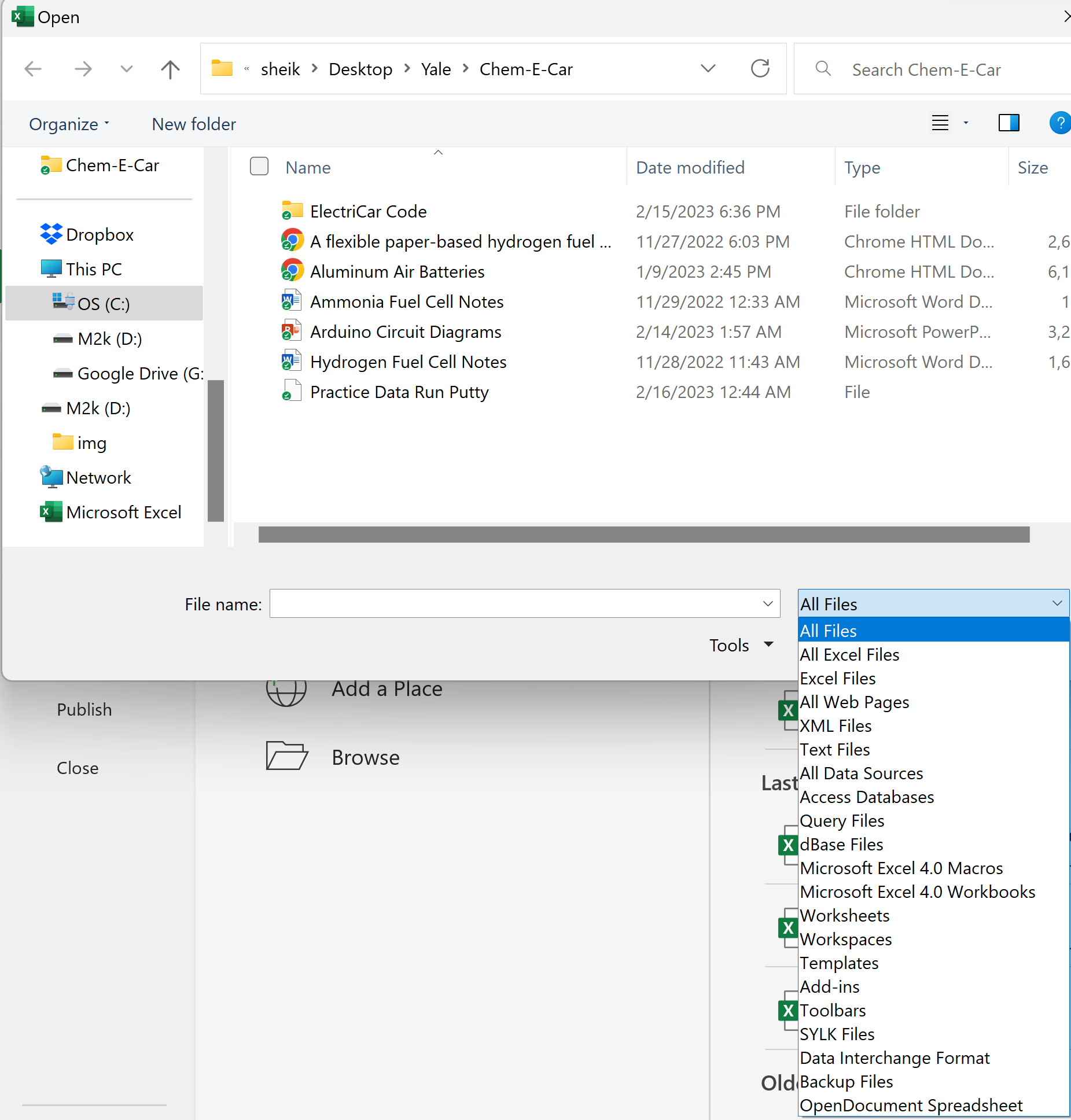
1. Click on “Logging” and under “Session Logging” click on “All Session Output” radio button.



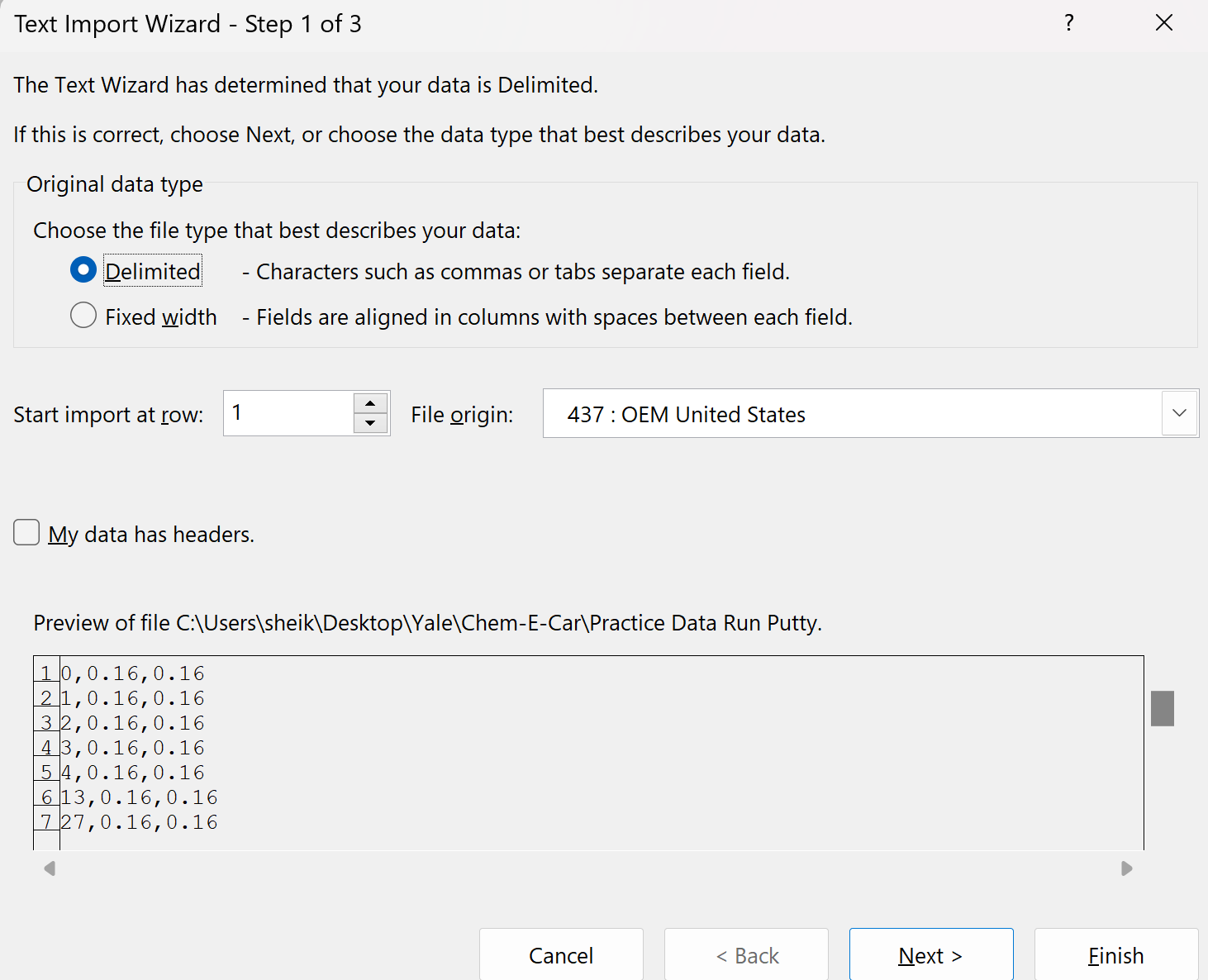
1. Select “Browse” button, navigate to directory **where JupyterLab data analysis file is**, and save with name of your choice.
2. Click “open” when ready. You should see a black screen where data is being logged.
3. Close window when finished- data should be automatically saved.

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

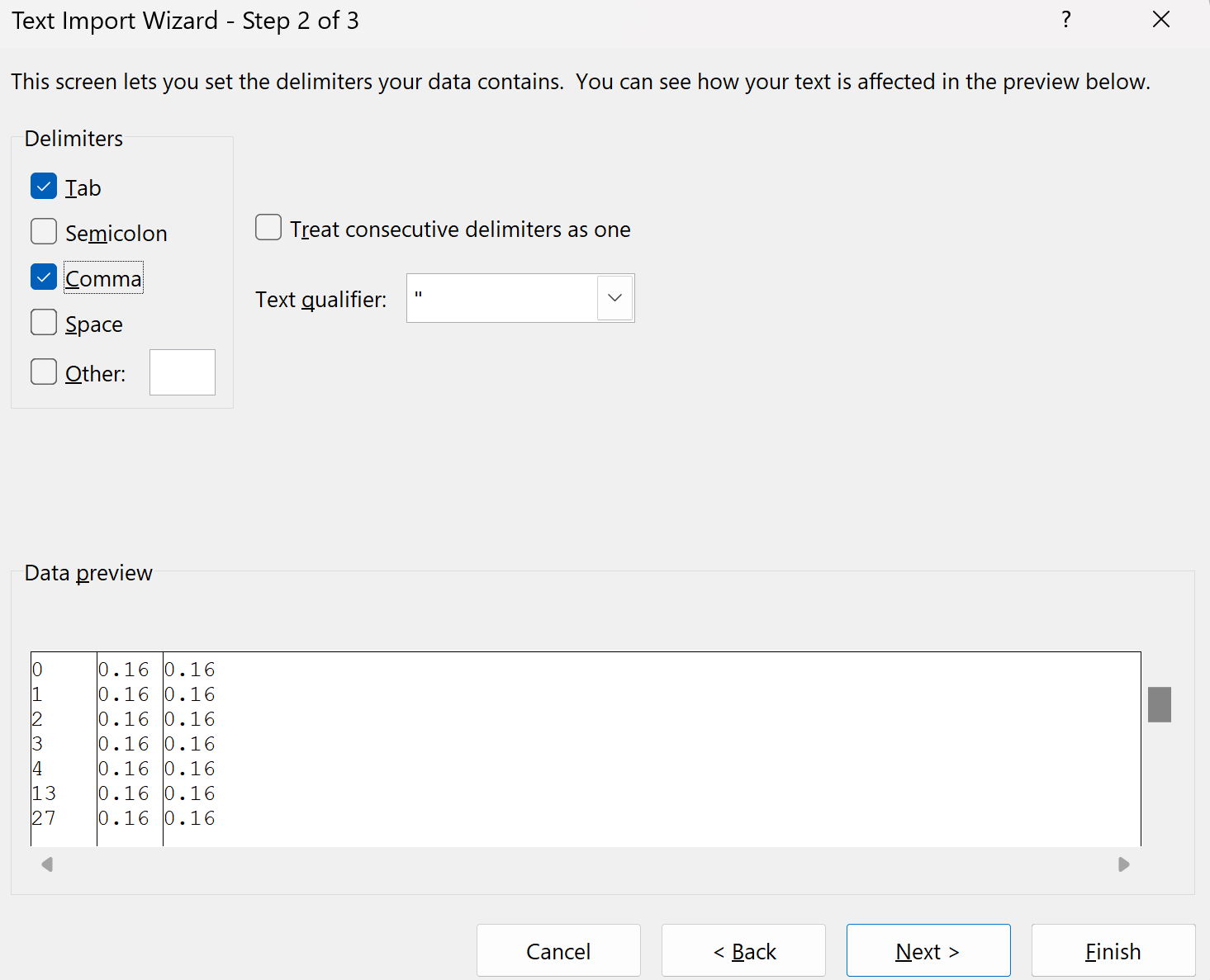
1. Find file where data was saved. Open it up using NotePad, and delete first line (as well as any repeated 0 lines).
2. Scroll down to bottom of file. Delete any incomplete lines.
3. Save and close file.
4. Open up the data file in Excel. Make sure that when searching for files, you have the “All Files”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



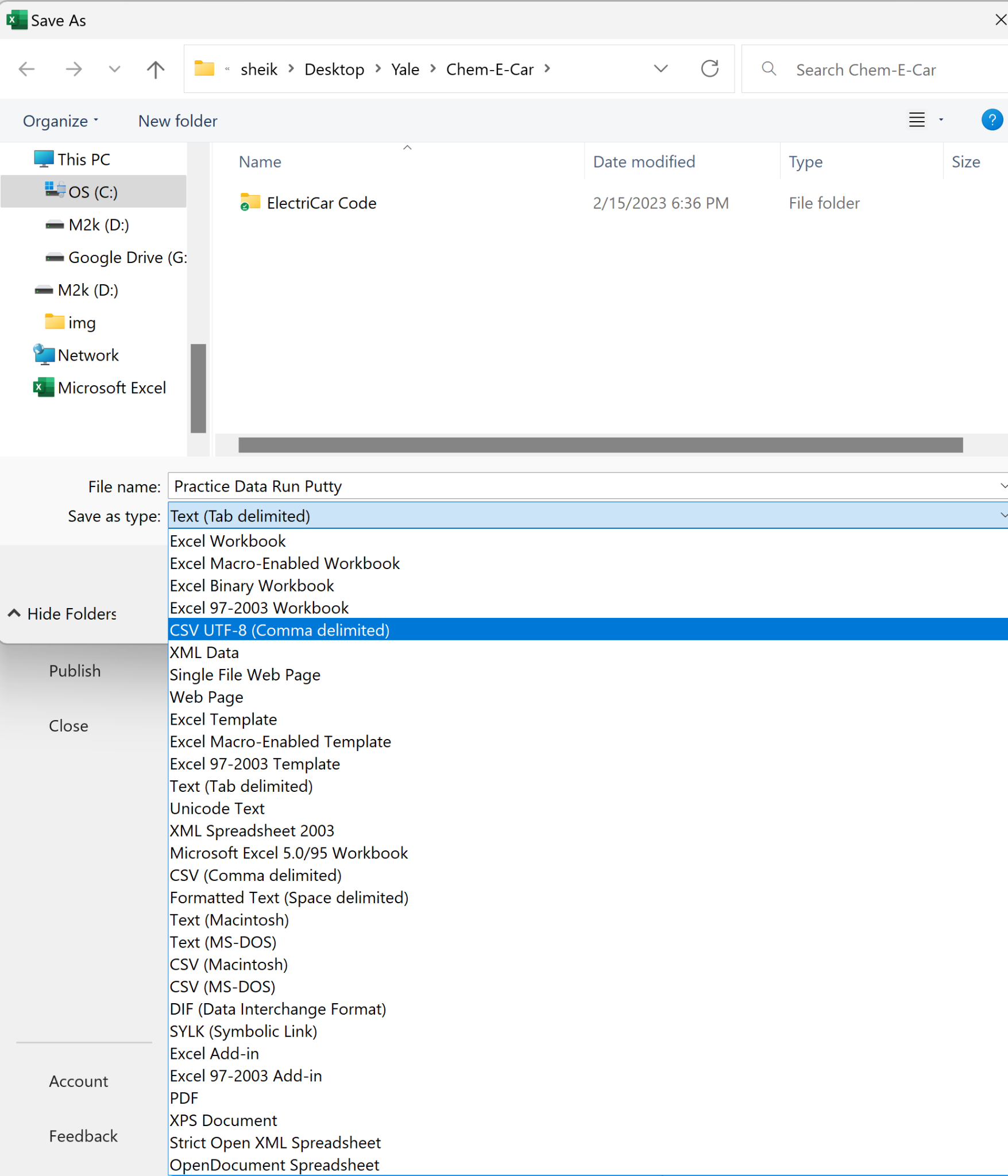
1. When they warn about incompatibility, say “Yes” you are sure, and then they will take you to a wizard. Make sure that you click the radio button with “Delimited”.



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

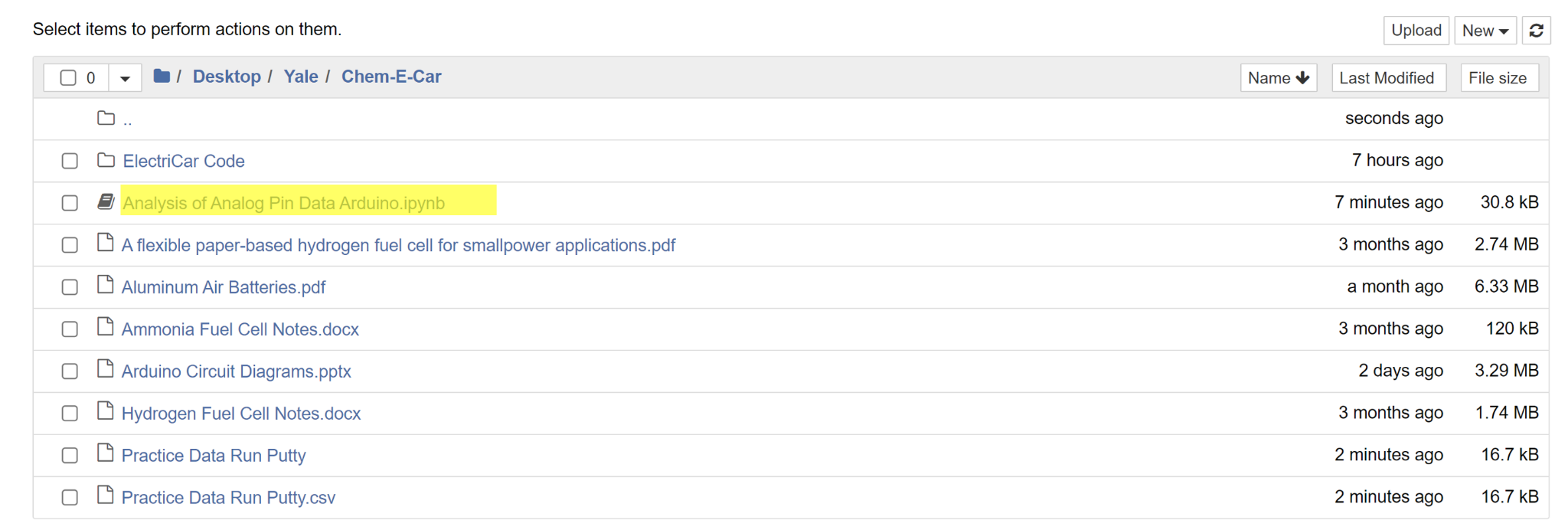


1. Save file as a CSV UTF-8 (Comma delimited) file, as shown below. Make sure this is in same folder as analysis code (else the code will not work).

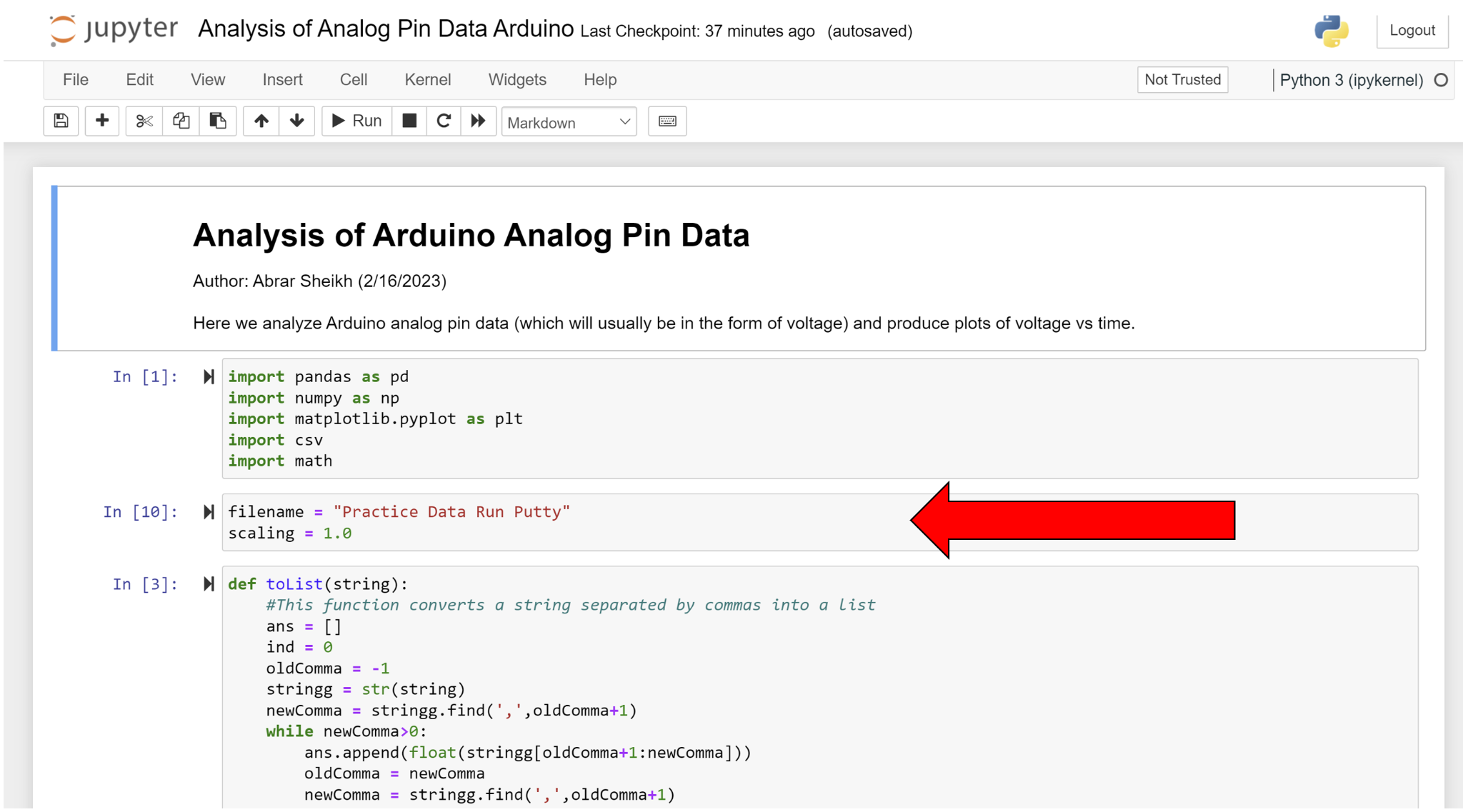


**Data Analysis:**

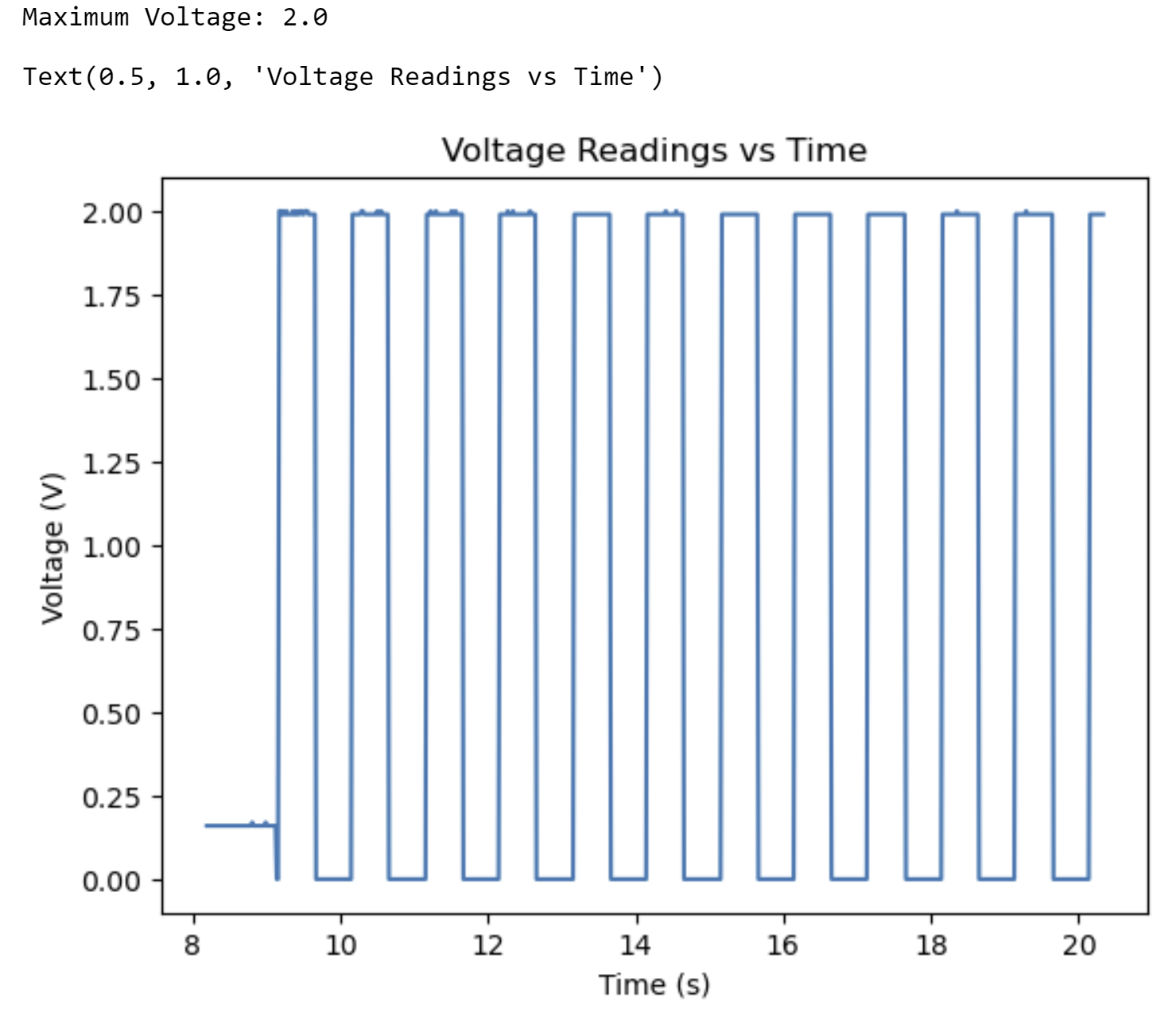
1. Type “Jupyter Notebook” into search bar, and let it open up (it should open in Chrome if using Abrar’s laptop).
2. Click on “Desktop” then “Yale” then “Chem-E-Car” then “ElectriCar” and finally open “Analysis of Analog Pin Data Arduino.ipynb”.



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 (slightly adjusted for clarity- that’s why it doesn’t start at 0 time):



(The input voltage fed to Arduino was a square wave of peak-to-peak voltage 2V and offset 1V, which is what we see in the graph).