Class #16: Linear Approximations

Purpose: The objective of this experiment is to begin to become familiar with linear approximations of electrical quantities in circuits and engineering in general.

Background: Before doing this experiment, students should be able to

- Analyze simple circuits consisting of combinations of resistors.
- Build simple circuits consisting of combinations of resistors on protoboards.
- Make voltage measurements using M1K Board and Alice tools.
- Review the background for the previous experiments.

Learning Outcomes: Students will be able to

- Apply curve fitting to data samples to determine linear characteristics of resistive circuits
- Determine the statistical distributions of collected data

Equipment Required:

- M1K Board (Alice) or Analog Discovery (with Waveforms Software)
- Voltmeter, Meter Source, Ohm Meter tools (Alice)
- Parts Kit
- Matlab

Helpful links for this experiment can be found on the course website under Class #16.

Pre-Lab

Required Reading: Before beginning the lab, at least one team member must read over and be generally acquainted with this document and the other **required reading** materials.

Part A – Linear Regression

Background

In a number of our experiments, we have investigated the linear relationship between inputs and outputs. For example, in our introduction of Ohm's Law, we stressed that V = IR is a linear relationship between voltage and current for a resistor. In that experiment we used the linear relationship to plot voltage against current, verifying that the slope was equal to the resistance. Similarly, in the superposition experiment we looked at the linear relationship between two inputs and a single output. In both examples, we knew that a linear relationship existed based on the defined physics. In practice, data does not behave as ideally as we see when making calculations by hand or running simulations. There are several reasons why measured data may not fall exactly on an expected curve (line). Our equipment is not always perfect and there is often inherent noise in the system we are using to make our measurements. This can be easily seen on the Discovery Board by looking at how the least significant digits of a measurements tend to change. Another reason is that the components themselves have manufacturing tolerances. Our resistors are gold band resistors, which means that they have a tolerance of 5%. In practice, this means that a 1kΩ resistor can actually have a value in the range of 950-1050Ω. The environment itself can lead to problems, with the parameters of the system changing over time as the environment changes. In electronics, temperature effects are a classic example of environmental effects, where a parameter like resistance changes over time as the circuit gets hotter.

When trying to extract information from noisy data, we can take advantage of our expectations to approximate an appropriate curve. In the case of linear expectations, we can use linear regression to obtain a best fit line that approximates the data. For example, in my courses, I often look at the relationship between homework scores and final course grade, as shown in Figure A-1. While there is no overall expectation that the relationship must be linear, the linear fit demonstrates a trend (about the importance of doing homework).

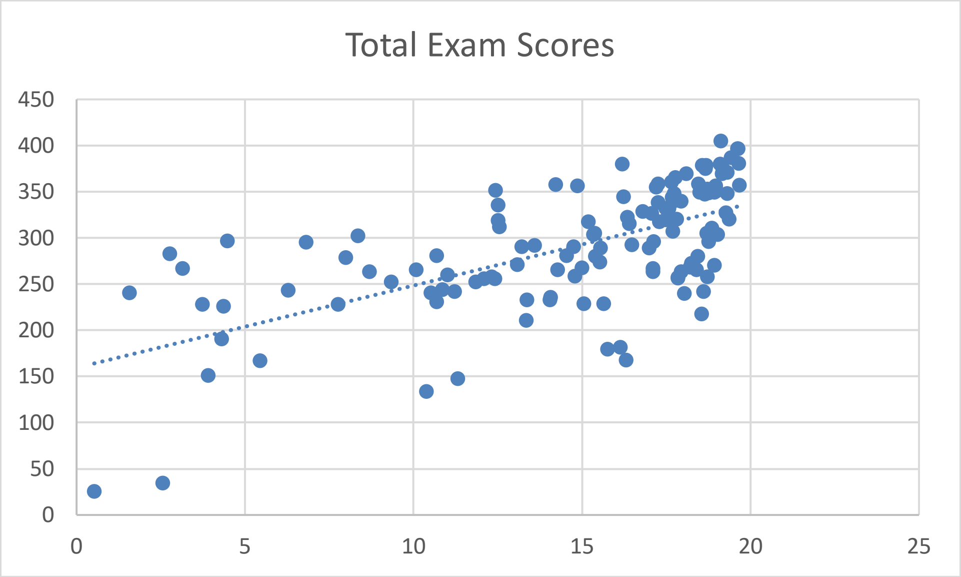


Figure A-2: Exam Scores vs. Homework Scores

There are several approaches to curve-fitting. We will use linear regression (also called ordinary least squares), which minimizes the total square of the vertical distance between the data points and the fitted line. Mathematically, we can write this expression as $\min\left[\sum_{i=1}^{N}(mx_i + b - y_i)\right]$, where *m* and *b* are the familiar slope and intercept of a line, and x_i and y_i are the coordinates of the *i*th data point out of *N* total data points. Since we are interested in finding the slope and intercept of the line, we can express them mathematically as

$$m = \frac{\sum_{i=1}^{N}(x_i y_i - x_i \overline{y})}{\sum_{i=1}^{N}(x_i^2 - x_i \overline{x})},$$

$$b = \overline{y} - m\overline{x}$$

where x_i and y_i are the coordinates of the *i*th data point out of *N* total data points (as above) and $\overline{x}$ and $\overline{y}$ are the average *x* and *y* values.

The average is the sum, divided by the total number of points

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

As an example, consider three points (1,1), (2,3), (3,2). If we plot them (in Matlab), as seen in Figure A-3, we can see that they are not collinear.

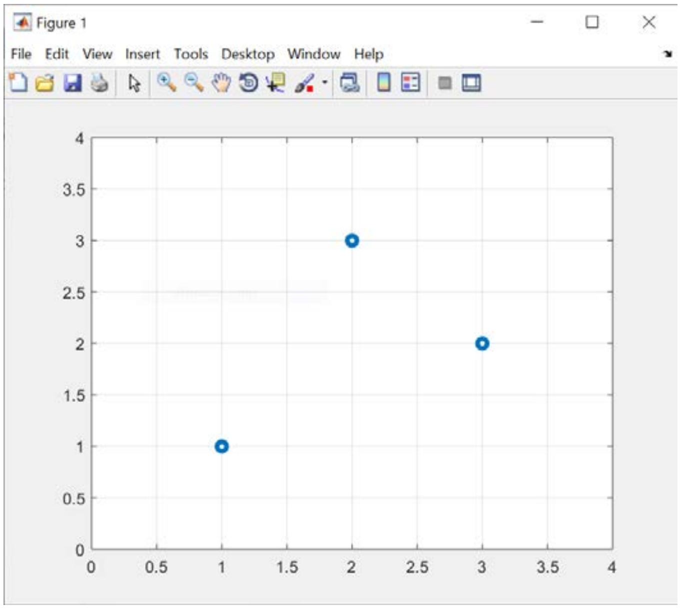


Figure A-3: Example Data

To find the best linear fit using the least squares method, we use the following steps.

1) Find the average x and y values for our data set. The average is the sum, divided by the total number of points

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{3}\sum_{i=1}^{3} x_i = \frac{1}{3}\left[1+2+3\right] = 2$$

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i = \frac{1}{3}\sum_{i=1}^{3} y_i = \frac{1}{3}\left[1+3+2\right] = 2$$

2) Find the slope of the line, using the summation on the previous page

$$m = \frac{\sum_{i=1}^{N}\left(x_i y_i - x_i \overline{y}\right)}{\sum_{i=1}^{N}\left(x_i^2 - x_i \overline{x}\right)} = \frac{\sum_{i=1}^{3}\left(x_i y_i - x_i 2\right)}{\sum_{i=1}^{3}\left(x_i^2 - x_i 2\right)} =$$

$$\frac{\left[(1)(1)-(1)(2)\right]+\left[(2)(3)-(2)(2)\right]+\left[(3)(2)-(3)(2)\right]}{\left[(1)^2-(1)(2)\right]+\left[(2)^2-(2)(2)\right]+\left[(3)^2-(3)(2)\right]} = \frac{1}{2}$$

3) Find the intercept of the line

$$b = \overline{y} - m\overline{x} = 2 - \left(\frac{1}{2}\right)2 = 1$$

We can now plot our line on the data and 'see' how the fit behaves with the real data.

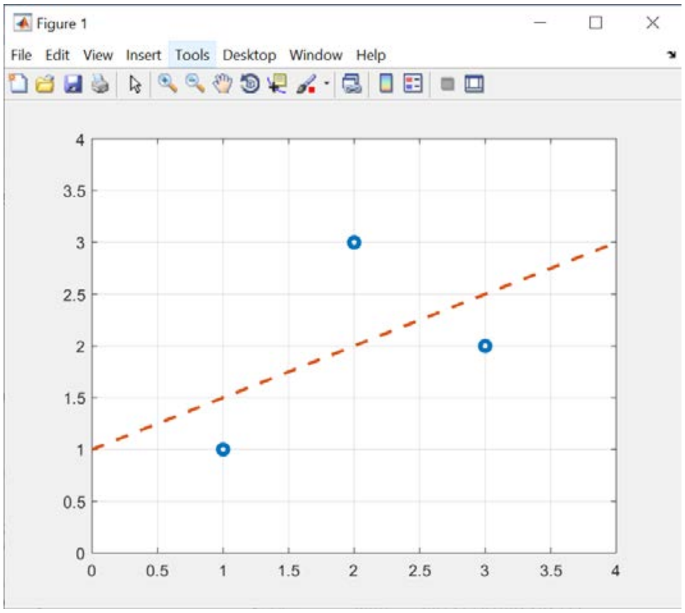


Figure A-4: Best Fit Line for Data

Exercise

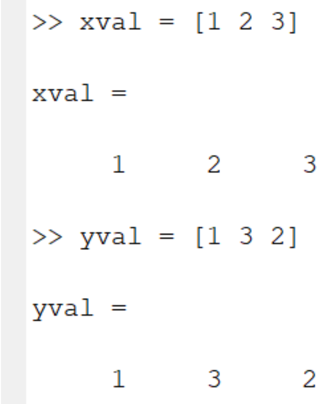
Follow the same process to find the best fit slope for the following sets of points

a) (2,1),(3,2),(4,3), note, this set is collinear

b) (1,1),(3,3),(5,2)

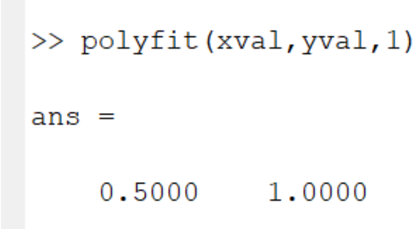
The instructors considered making you do this calculation by hand for hundreds of data points, but instead decided using Matlab would be much more practical (but less fun). You can check your above work using the Matlab `polyfit` function.

First, you need to enter the coordinate data as an array of the x-coordinates and another array of the y-coordinates. Using the example problem, we can create array variables called `xval` and `yval` in Matlab.



```
>> xval = [1 2 3]

xval =

     1     2     3

>> yval = [1 3 2]

yval =

     1     3     2
```

Note, the order of array elements has been the same for both variables. Now we can find the slope and intercept of the best fit line. The '1' in the command indicates that you are looking for a linear fit. (You can find higher order polynomial fits, ie. quadratic, cubic, etc.)



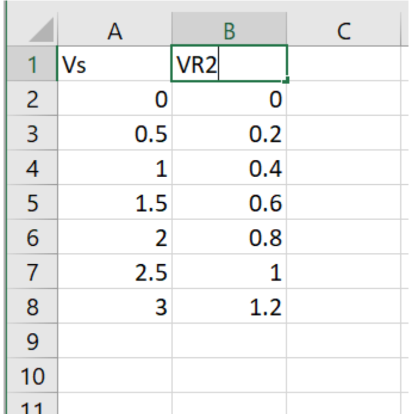
```
>> polyfit(xval,yval,1)

ans =

    0.5000    1.0000
```

The first element of the answer is the slope, *m*, and the second element is the y-intercept, *b*.

c) Check your work on problems a and b using Matlab.

Experiment

Revisit your data from Experiment 5, Part C, where you made voltage measurements across R2. Using voltage divider relationships, we know that the slope of the line relating VR2 to the Source Voltage is, $\frac{R2}{R2+R1}=\frac{1000}{1000+470}=0.6803$. We also expect the intercept to be zero, since the voltage across R2 should be zero when the source voltage is zero. We will test this values in Matlab. There are several ways to load data into Matlab rather than entering the arrays manually. For this experiment, we will take advantage of Excel. Open an Excel file and label the first to elements of the top row with meaningful data names. Then, for each column enter the data from Experiment 5, with the first column being the Source voltage (CH A Voltage) and the second column being the measured R2 voltage. An example is shown in Figure A-5.

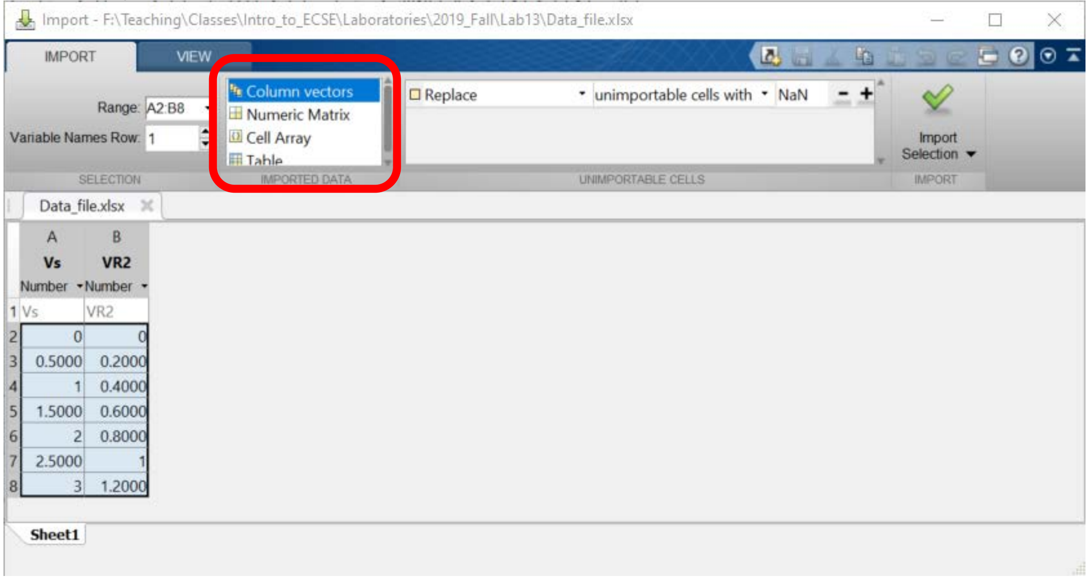


	A	B	C
1	Vs	VR2	
2	0	0	
3	0.5	0.2	
4	1	0.4	
5	1.5	0.6	
6	2	0.8	
7	2.5	1	
8	3	1.2	
9			
10			
11			

Figure A-5: Example of Excel Data

The file can be imported directly into Matlab

1. Click the "Import Data" icon on the top menu bar
2. Navigate to the directly you saved your Excel file and double click the file
3. Make sure you select "Column Vectors" in the "Imported Data" tab



Import - F:\Teaching\Classes\Intro_to_ECSE\Laboratories\2019_Fall\Lab13\Data_file.xlsx

Range: A2:B8

Variable Names Row: 1

	A	B
	Vs	VR2
	Number	Number
1	Vs	VR2
2	0	0
3	0.5000	0.2000
4	1	0.4000
5	1.5000	0.6000
6	2	0.8000
7	2.5000	1
8	3	1.2000

Figure A-6: Matlab Import Data Window

4. Click the green check mark
5. Close the window.

The variables (x and y coordinates) have now been loaded into Matlab. We can now plot the data and find the best fit line. Rather than use the command line repeatedly, we want to start using scripts.

6. In the upper left corner, click the “New Script” icon. A new window will open.

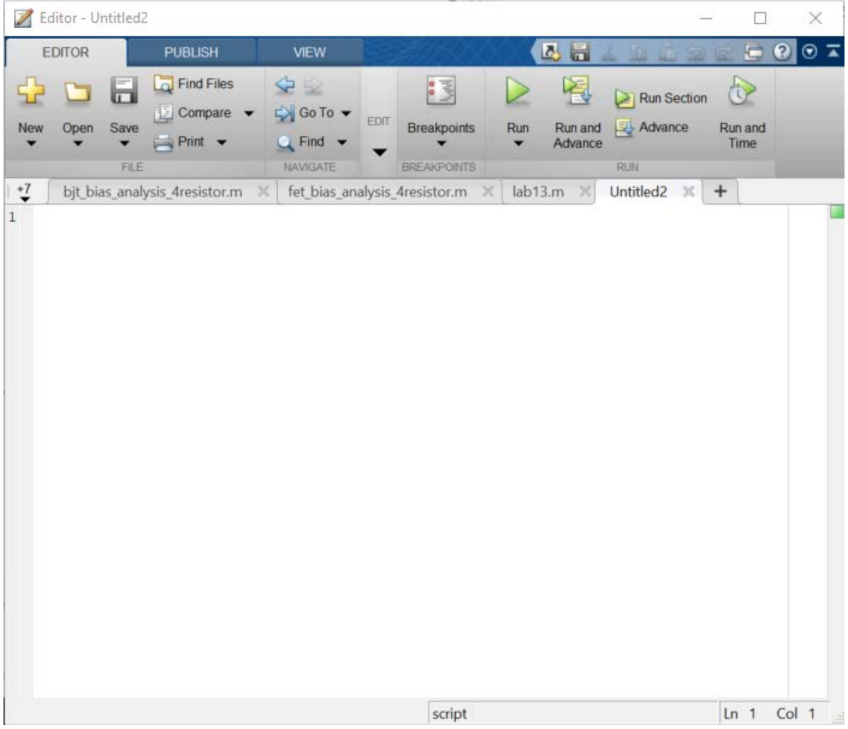
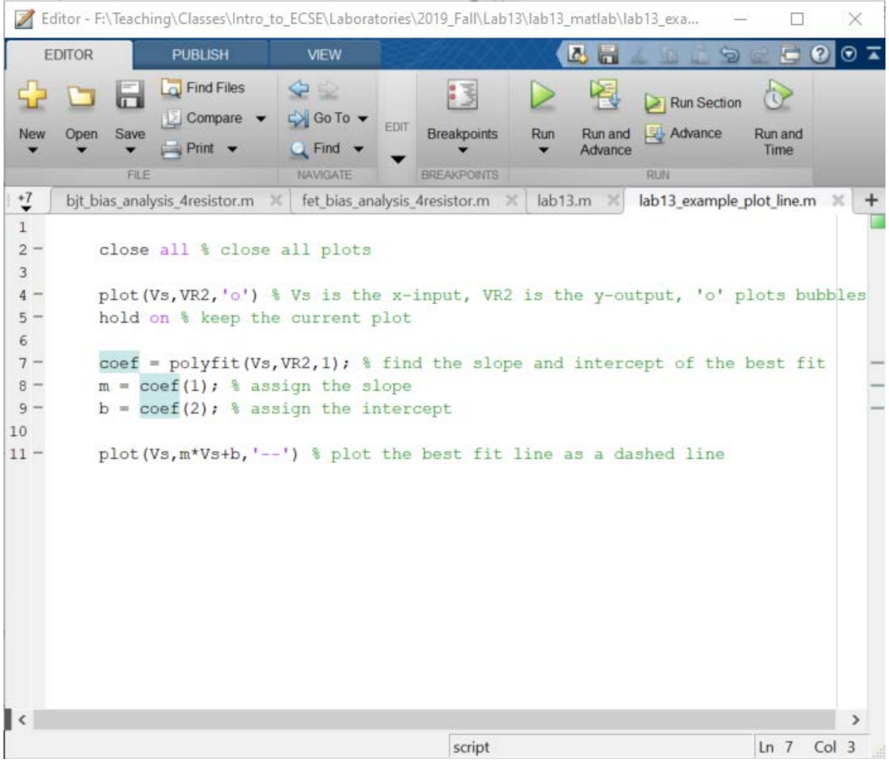


Figure A-7: Matlab Script Window

We can type our commands directly into the window. To generate figures similar to what was done with the opening example, the following script is used



Editor - F:\Teaching\Classes\Intro_to_ECSE\Laboratories\2019_Fall\Lab13\lab13_matlab\lab13_exa...
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
bjt_bias_analysis_4resistor.m
fet_bias_analysis_4resistor.m
lab13.m
lab13_example_plot_line.m
script
Ln 7
Col 3

```

close all % close all plots

plot(Vs,VR2,'o') % Vs is the x-input, VR2 is the y-output, 'o' plots bubbles
hold on % keep the current plot

coef = polyfit(Vs,VR2,1); % find the slope and intercept of the best fit
m = coef(1); % assign the slope
b = coef(2); % assign the intercept

plot(Vs,m*Vs+b,'--') % plot the best fit line as a dashed line
```

Figure A-8: Matlab Script for Least Squares Fit

We encourage you to use the same script, editing the variable names if needed. The resulting plot is shown on the next page

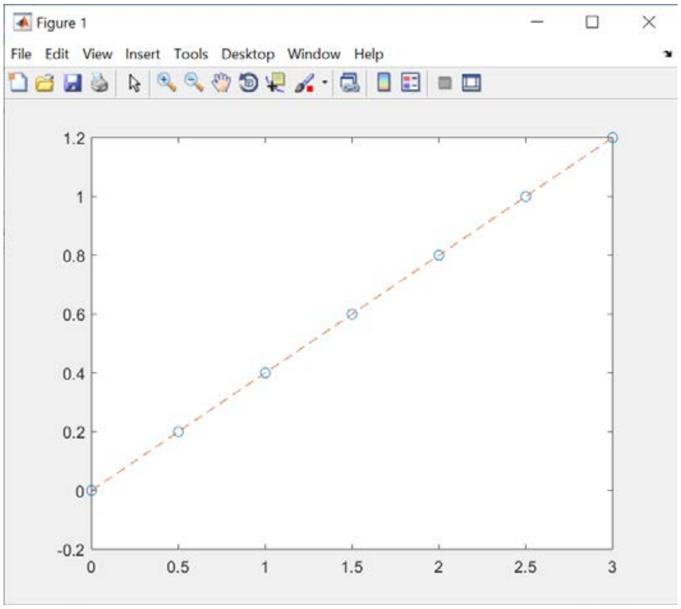


Figure A-9: Data and Linear Fit

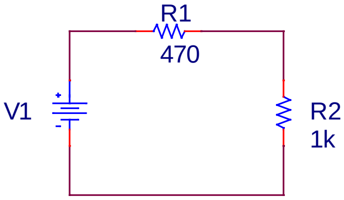
In the above case, the data was linear and the linear fit intersects each data point.

Exercise

d) Following the above process, save your Experiment 5 data into Excel. Plot the data and linear fit using the Matlab script on the previous page. Include the equation of the line and your plots results in the report.

e) Revisit your Experiment 11 Part C results, where you stepped the voltage from 0V to 5V for V1 and V2, respectively, and measured the voltage at the top node. Use the script to find the best fit line of your experimental data when V2 = 0 and then again when V1 = 0. Do the numerically calculated slopes (from `polyfit`) match your experimental observations in Experiment 11?

Plotting data that is linear is not very interesting. We will revisit the Experiment 5 voltage divider, but with a little randomness included.



Before building the above circuit, find your collection of 470Ω and 1kΩ resistors and put them in separate piles. You should have at least five (ten is better) of each resistor. Measure the voltage across R2 for each of the following source voltage. **For each** measurement, select a random resistor from each pile and rebuild the circuit. Due to resistor tolerances, there will be a slightly different voltage divider each time. The data will no longer be linear, though, it will still be close.

Voltage, V1	Resistor Voltage, VR2
0	
0.5	
1	
1.5	
2	
2.5	
3	
3.5	
4	
4.5	
5	

f) Use the Matlab script to plot the data and best fit line.

Since we are confident you did not find those measurements tedious, we are going to ask you to perform a similar experiment to study statistical variance. In this case, pick a single R1 (470Ω) resistor and hold the source voltage at 4V. For each of your 1kΩ resistors, measure the voltage across R2. Assuming R1 is exactly 470Ω and that the source is exactly 4V, use your measured R2 voltage and the voltage divider expression to calculate the real R2 resistance. Once you finish your calculations, add them to the Instructor's Excel file of saved data. We will look at some statistical analysis of data in the next class.

We will also use Alice ohmmeter tool to measure the value of each resistor R2. For this, you need to connect one end of resistor into CH A of M1K board and the other end in the CH B pin of M1K board and "run" the Ohm Meter. Note, close out other Alice applications before you start ohm Meter tool. Only one Alice application can be active at any time.

Resistor Number	Resistor Voltage, VR2	Calculated value of R2	Measured value of R2 (Alice Ohm Meter)
1			
2			
3			
4			
5			

g) Record the calculated and measured R2 values in the instructors Excel file below. Scroll down and append your data to the end of the list.

https://docs.google.com/spreadsheets/d/1PeX5iO9ZlaSKPwtb_WWVJ-t8ghRDLN4I60zg0duKBjo/edit?usp=sharing