Class #17: Statistical Analysis and Rectangular Matrices

Purpose: The objective of this experiment is to familiarize yourself with common statistical analysis terms and revisit linear curve fitting with multiple inputs.

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

- Analyze simple circuits consisting of combinations of resistors.
- Apply Ohm's Law to determine current from voltage measurements
- Make differential voltage measurements using M1K board and Alice tools.
- Invert a matrix and perform matrix multiplication
- Use Matlab to perform linear regression and matrix manipulation
- Review the background for the previous experiments.

Learning Outcomes: Students will be able to

- Understand the properties of Gaussian distributions
- Use statistical parameters to determine the quality of their least squares curve fitting
- Use matrix mathematics to determine a linear fit for multiple input variables

Equipment Required:

- M1K board (with Alice tools) or Analog Discovery (with Waveforms Software)
- Voltmeter tool (Alice)
- Meter-Source tool (Alice)
- Parts kit
- Matlab

Keywords:

- Histogram
- Mean
- Standard Deviation
- Median
- Correlation Coefficient

Helpful links for this experiment can be found on the course website under Class #18.

Part A – Statistical Analysis (Gaussian and Uniform Distribution)

Background

In the last experiment, we looked at fitting a linear approximation to data. In this experiment, we will look at a more general discussion of data distributions and then revisit the data from last experiment. To start off, we can consider a very common data distribution called the **Gaussian distribution** (Gaussian function). This distribution is one we see frequently when discussing test grades. Mathematically, we can represent the Gaussian distribution as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

where $\bar{x}$ is the **mean** (average), which we used in experiment 16 (previous class) when determining the slope and intercept of the least squares fit, and σ is the **standard deviation**. The standard deviation is a measure of how 'spread out' the distribution appears. Considering the following two figures, Figure A-1 is an example of 'wide' Gaussian distribution with a 'larger' standard deviation and Figure A-2 is an example of a 'narrow' Gaussian distribution with a 'smaller' standard deviation. We can clearly see that a smaller standard deviation looks much 'narrower'. As mentioned above, this type of distribution is commonly associated with grades, where the mean is average assigned grade (C range perhaps) and a positive standard deviation would be one letter higher (B range perhaps). A negative standard deviation would be one letter grade lower (D range perhaps). Figures A-1 and A-2 are continuous plots, where the input can be any real value between 0 and 100, not just integers. When we consider data gathered, our measurements from last experiment as an example, the data set is then discrete. We can fit a Gaussian curve to that data, using a procedure similar to the least squares operation in the last class. Another approach is to use a **histogram**, which counts the frequency of data points within a range of values. This type of plot is especially useful when considering a data set that is integers. Again, using grades as an example, Figure A-3 is a histogram plot of exam grades. Observationally, we can see that the data is approximately Gaussian. Discrete data analysis (matrix analysis), indicates that the mean is 64.9 and the standard deviation is 16.2. If we implement curve fitting, the plot in Figure A-3 would be very close to the Gaussian fit.

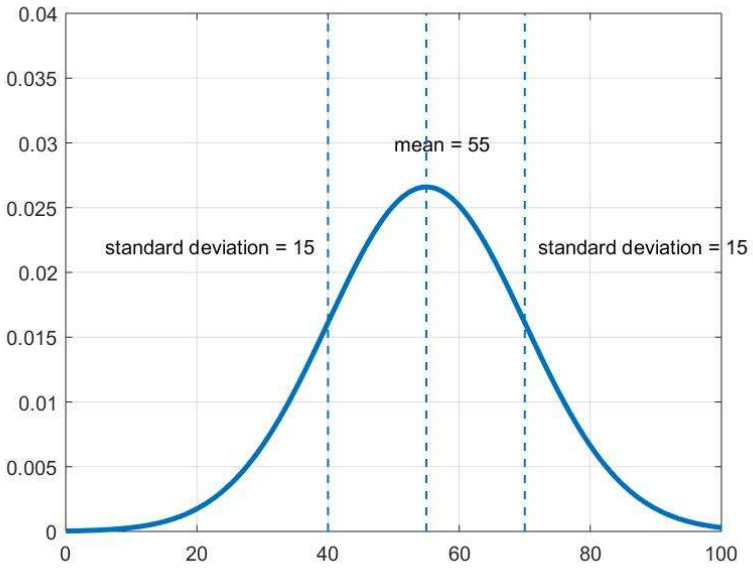

Figure A-1: Gaussian Distribution, Mean = 55, Std. Dev. = 15 (largish)

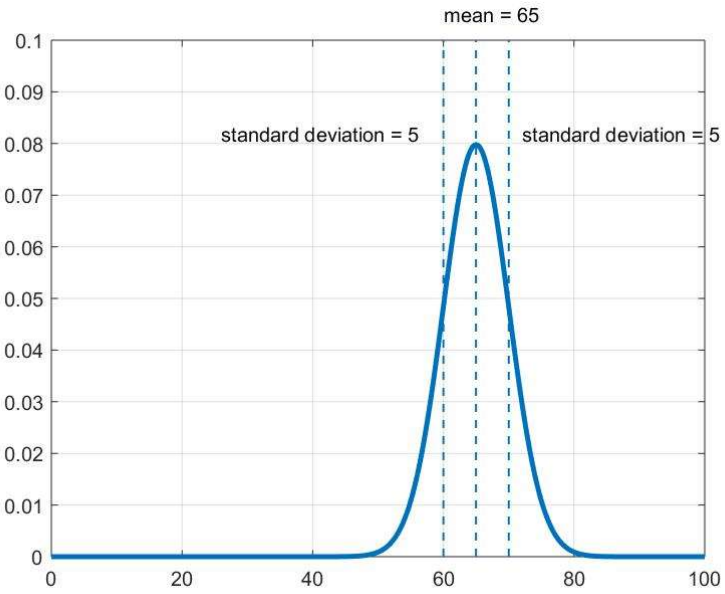

Figure A-2: Gaussian Distribution, Mean = 65, Std. Dev. = 15 (smallish)

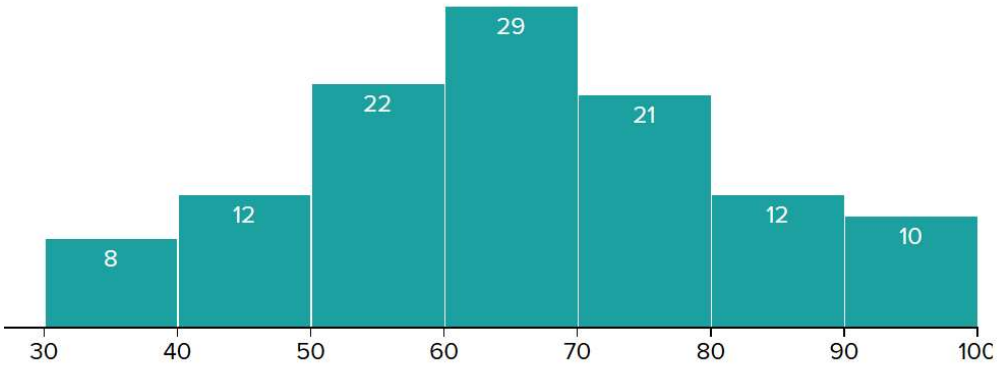

Figure A-3: Histogram of Exam grades, Mean = 64.9, Std. Dev. = 16.6

In the case of discrete data sets, another metric of interest is the **median**, which is the 'middle' data point when sorting the data from smallest to largest (or largest to smallest). For example, the array

$$\begin{bmatrix}1 & 9 & 4 & 9 & 8 & 2 & 3\end{bmatrix}$$

After sorting, the array becomes

$$\begin{bmatrix}1 & 2 & 3 & 4 & 8 & 9 & 9\end{bmatrix}$$

and the median (middle number) is 4. In general, if the data set is symmetrical, like the Gaussian distribution in Figure A-3, the **median** and **mean** are very close to each other. The median of the Figure A-3 data was 64, which is very close to the mean.

There are other types of data distributions that we see frequently. A variation on the above plots is a skewed Gaussian distribution, where the data is not symmetric. An example is shown in Figure A-4. An important observation for the skewed Gaussian is that the median and the mean are no longer close together.

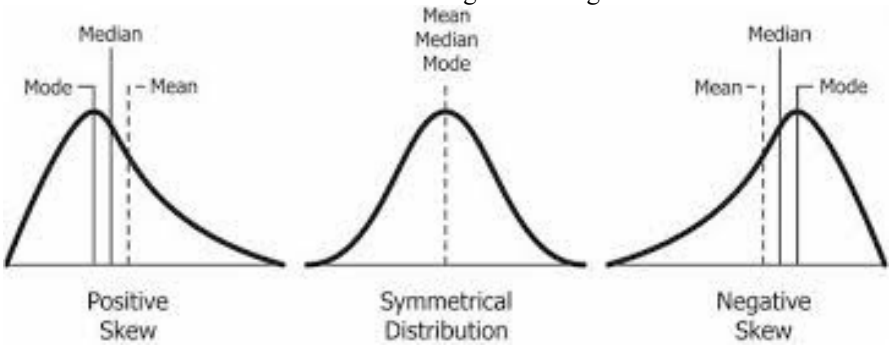

Figure A-4: Skewed Gaussian Distributions

Using grades as an example, Figure A-5 presents asymmetric data. The results can be interpreted as a **skewed Gaussian** distribution. In this case, an argument could be made that the data distribution is exponential, having a maximum frequency of occurrence at 50 and then a decaying frequency of lower scores.

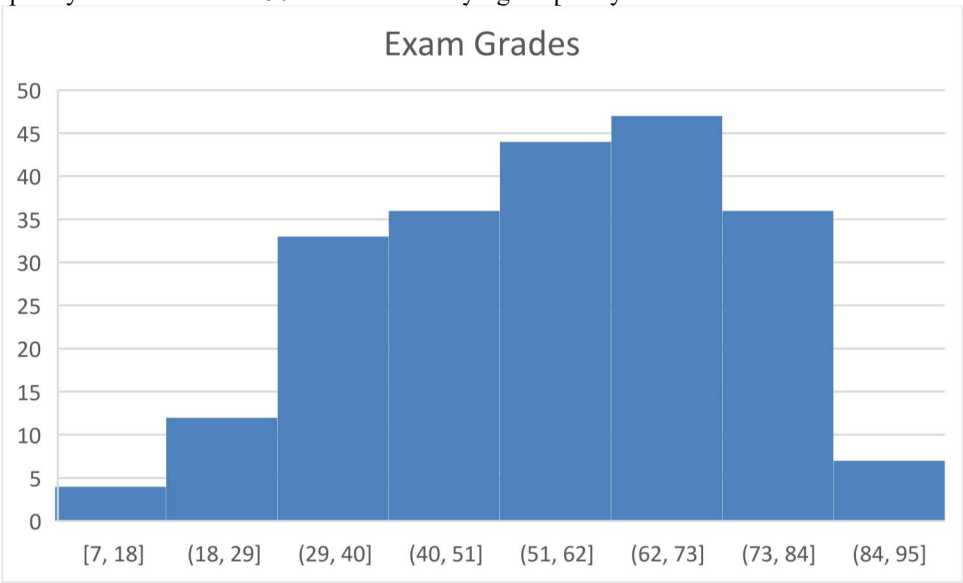

Figure A-5: Skewed Gaussian Distribution with Mean = 56.0 and Median = 57.5

Two other common distributions are the **Uniform** distribution, where the frequency of occurrence is flat over the range of data. In this type, each outcome (score) is equally likely. You can see an example of that in Figure A-6. A **Bimodal distribution** is a data set with two different Gaussian distributions. Examples are shown Figures A-7 (example curve) and A-8 (real exam data). In the case of scores, instructors are frequently disturbed by Bimodal distributions, leading to a lot of discussion about the course.

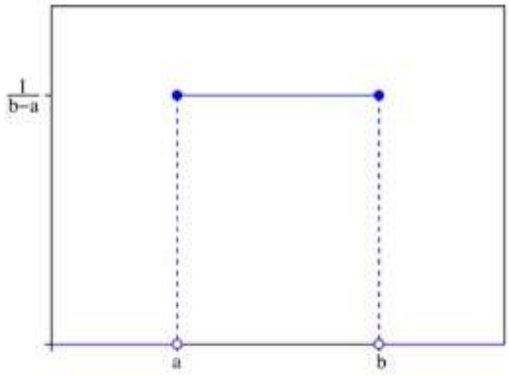

Figure A-6: Example of a Uniform Distribution

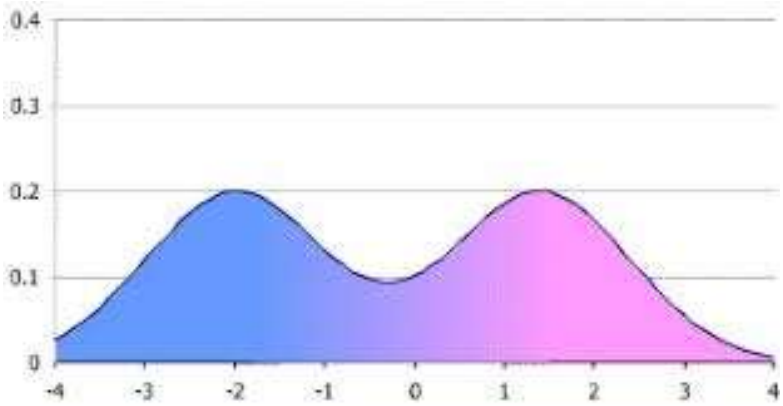

Figure A-7: Example of a Bimodal Distribution

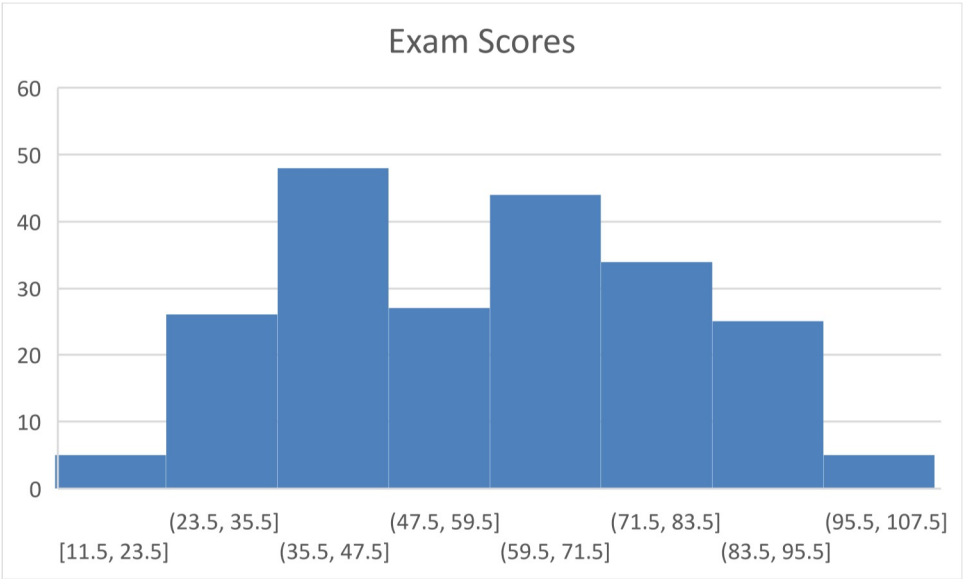

Figure A-8: Bimodal Distribution of Exam Scores

Important note: A fair bit of mathematics is applied when studying probability distributions and you will see some of that in a future course (ECSE-2500). We don't expect you to have a deep understanding of the mathematics, but we do expect to recognize common distribution functions/histograms and concepts like mean, median and standard deviation.

At the end of experiment 16, you collected data about the actual value of the 1kΩ resistors in your parts kit. Remember, the fourth color band indicates the tolerance of the resistor, i.e. the range of values close to 1kΩ that is considered acceptable manufacturing. Gold band like ours resistors are 5% tolerance, meaning that the 1kΩ resistor can actually have some value between 950Ω and 1050Ω. The data we collected is available in an Excel file linked in the Class 17 section of the web page. You can also see a detailed histogram that Dr. Hameed generated.

Since there are a large number of data points, we will use Matlab to do the heavy lifting. The commands in Matlab are fairly straightforward. For some array called `data`,

The mean is determined using the `mean` command

```
>> mean(data)
```

The median is determined using the `median` command

```
>> median(data)
```

The standard deviation is determined using the `std` command

```
>> std(data)
```

A histogram plot is obtained using the `histogram` command

```
>> histogram(data)
```

We can change the range of data for the histogram plot (these are called bins), by adding an integer value to the command. For example

```
>> histogram(data,12)
```

Changing the bin size can make the data distribution look a little different. Though, the statistical curve fitting is essentially the same.

Exercise:

1) Download the Excel file and load it into Matlab. Use the above commands to determine the mean, median and standard deviation for the resistor measurements. Are the mean or median or both close to the expected value of 1kΩ? What experimental issues/assumptions in experiment 16 may have introduced error? Is the standard deviation within the range expected by the tolerance?
2) Make a histogram plot. In this plot, you should notice an outlier point, which should be clearly a bad data point. Delete that data point from the Excel file and reload the data into Matlab. Recalculated your values from part 1. One of your results should have changed significantly. Given the new results, what observation can you make about the actual tolerance of this batch of resistors?
3) For your histogram plot, which of the discussed distributions most closely matches what you see? Make a couple more histogram plots, changing the number of bins. Are the new plots consistent with your chosen distribution type (there are no wrong answers).

Part B – Quality of Linear Regression Curve Fitting

Background

In experiment 16, after finding the linear fit for the data points, we can ask the question of how good is the fit. One of the metrics used to analyze the **correlation coefficient**. We can express this mathematically as

$$\rho = \frac{1}{\sigma_x \sigma_y} \sum_{i=1}^{N} \left[(x_i - \bar{x})(y_i - \bar{y}) \right]$$

where x_i and y_i are the coordinates of the data points, $\bar{x}$ and $\bar{y}$ are the mean values of the *x* and *y* coordinates, and σ_x and σ_y are the standard deviations of the *x* and *y* coordinates. We will again use Matlab to do the analysis, using the command `corrcoef`. For our data, this command will return a 2x2 array. Given our relatively simple data, the main diagonals of the array will be 1 and the off-diagonal terms will be a value less than or equal to 1. The off-diagonal terms will be the result of interest for us. If that value is close to 1, we know the least squares fit is very good and the data is nearly linear. In the limit, for linear data, that value will be 1. The closer the value is to zero, the less we confidence we have that the linear fit is an accurate approximation.

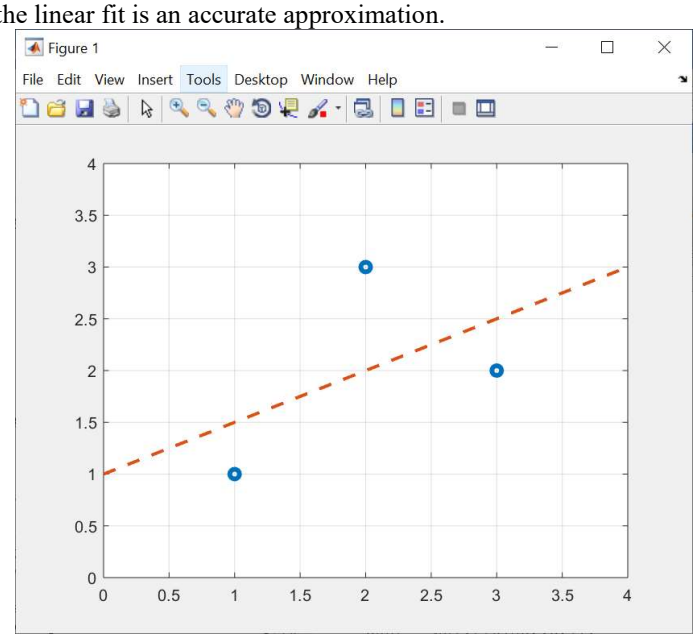

Figure B-1: Best Fit Line for Data

Using the three element example data set from Experiment 16, (1,1), (2,3), (3,2) that had a linear fit shown in Figure B-1, we can assess the quality of the fit using correlation coefficient,

```
>> corrcoef([1 2 3],[1 3 2])

ans =

    1.0000    0.5000
    0.5000    1.0000
```

(Note: the first array is the x-coordinates and the second array is the y-coordinates.)

The off-diagonal terms are 0.5, indicating the linear fit is 'not great'. This would be expected for so few data points.

Exercise:

4) Revisit the three element arrays from experiment 16 and find the correlation coefficients
   a. (2,1),(3,2),(4,3)
   b. (1,1),(3,3),(5,2)

   Based on the Matlab calculations, would you characterize this data sets as linear?

5) Do the same for your linear fit analysis of the experiment 5, experiment 11, Part C, and from experiment 16, Part F data. In this case, when you import the Excel data, you can use the column labels in the `corrcoef` command instead of writing the arrays directly. Using the example from experiment 16, you could write

```
>> corrcoef(Vs,VR2)
```

   Again, based on the Matlab calculations, would you characterize the results as linear?

Part C –Linear Approximations with Two Inputs

Background

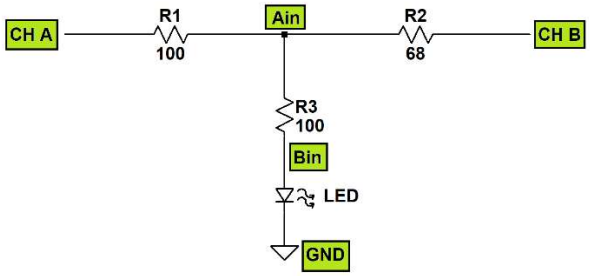

Figure C-1: Non-linear Circuit

In Figure C-1, we see a circuit with two input voltages. This circuit is very similar to the superposition circuit seen in experiment 11, In this case, we are going to look at the current through resistor R3. If the circuit was linear, then using experiment 11 concepts, we could write the current as

$$I_{R3} = aV_1 + bV_2$$

where *a* and *b* are the coefficients for a linear relationship. With the LED in the circuit, the relationship is not linear, but a linear approximation may be valid. To find the coefficients *a* and *b*, we can use matrix analysis. We still want a solution to the expression,

$$Ax = b$$

Again, we are using measured data to extract the unknown coefficients *a* and *b*. We will have multiple measurements, but only two unknowns, which means that the matrix *A* is rectangular, not square. Each row of *A* is the values of V1 and V2, with the corresponding element of *b* being the measurement result. We need to make changes to our matrix expression in order to find a solution. In order to make the systems square, we multiply both sides of the expression by the **transpose** of A

$$\left(A^T A\right)x = A^T b$$

The transpose of a matrix is obtained by switching the row and column elements, an element in row *i* and column *j* is placed in row *j* and column *i*. As an example,

For matrix $A = \begin{bmatrix} 1 & 2 \\ 3 & 7 \\ 5 & 1 \end{bmatrix}$, with a transpose $A^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 7 & 1 \end{bmatrix}$

In Matlab, the transpose command is `transpose`,

```
>> transpose(A)
```

6) Implement the circuit in Figure C-1 on your protoboard and measure the voltage across R3 with the following voltages
   a. V1 = 2.0V, V2 = 2.0V
   b. V1 = 2.0V, V2 = 2.5V

   Use Ohm's Law to calculate the current through R3.
7) Your matrix and arrays are now

$$A = \begin{bmatrix} 2 & 2 \\ 2 & 2.5 \end{bmatrix}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix}, \qquad b = \begin{bmatrix} measurement1 \\ measurement2 \end{bmatrix}$$

   a) Find the transpose of A (by hand)
   b) Multiply matrices, $\left(A^T A\right)$ (by hand)
   c) Find the inverse of your part b result (by hand)
   d) Find your values for the *x* array by completing the matrix multiplication, $\left(A^T A\right)^{-1} A^T b$ (by hand)
   e) Use your result to predict the current when V1 = 2.5V and V2 = 2.5V

8) Collect two more data points and repeat parts a-e. You can use Matlab for each step, but should include the results for each step in the report. At least one of your voltages should be equal to or larger than 2.0V. Do not pick V1 = 2.5, V2 = 2.5V as a data point.
9) Collect four more points, with the same conditions as 8).
10) Set V1 = 2.5V and V2 = 2.5V and find the current through R3.
11) Compare your actual measurement to the estimates of parts 7, 8 and 9. As you collected more data, did the linear approximation approach the measured value?