

Lab: Two Dimensional Vectors

Question 1 (20 minutes): Analysis of Temperature and Precipitation Data

The Comma Separated Value (.csv) file “Davis_Weather_7012010_6302011.csv” contains daily weather data (average daily temperature and precipitation) from 7/01/2010 to 6/30/2011. The data was recorded by a weather station in Davis. You can open this file with any text editor or spreadsheet software to view its contents.

Manually import this large data set into MATLAB is tiresome and likely to introduce errors. There are several ways the import process can be expedited. In this question we will learn how to use ‘uiimport’.

Task 0: Follow steps below to get started.

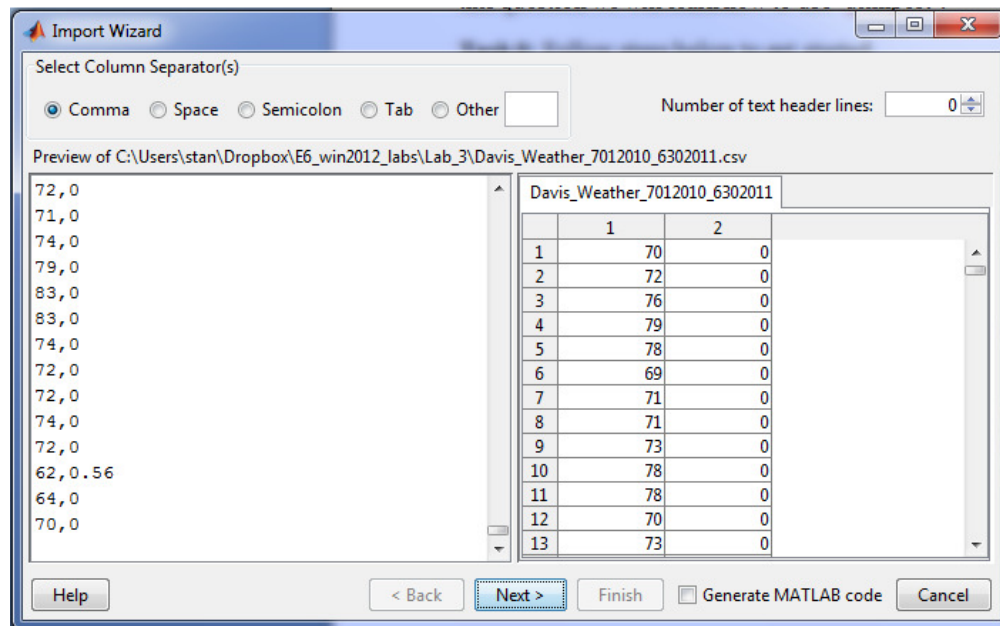
Step 1: Type ‘uiimport’ into the command prompt and a window will pop up.

Step 2: Click ‘File’ when asked to select a source.

Step 3: Navigate to, and select the file, ‘Davis_Weather_7012010_6302011.csv’.
Click Open.

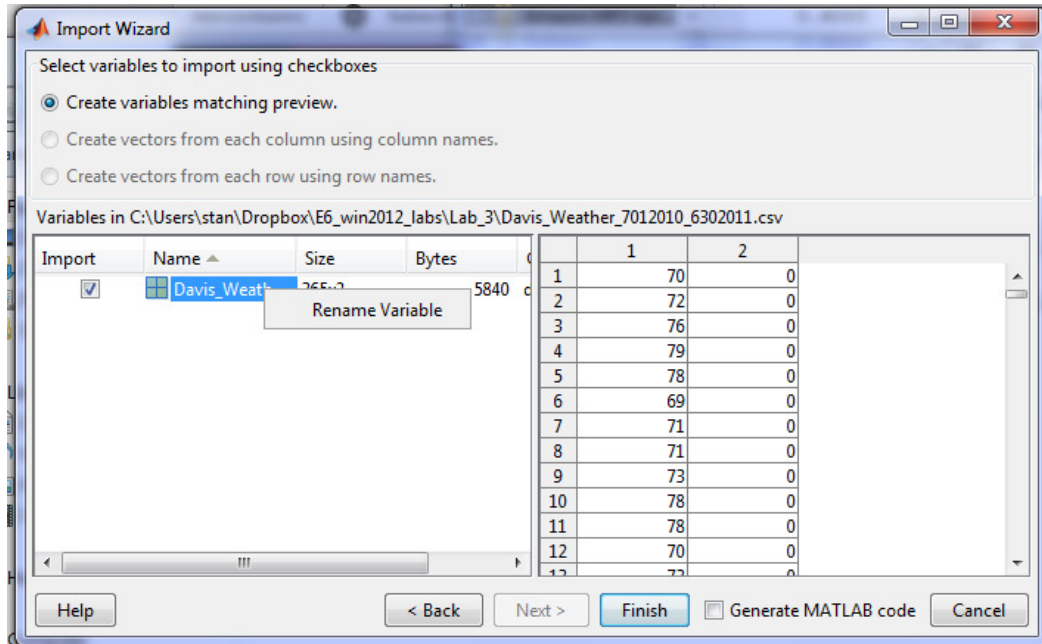
Step 4: You now have a window that looks like the figure below. This window allows you to see how MATLAB will parse this file. Make sure the settings are set as follows and click ‘Next’.

- Select Column Separator(s): Comma
- Number of text header lines: 0 (this is the number of rows to skip)



Lab: Two Dimensional Vectors

Step 5: In this next window (see picture below), you see information (Name, Size, Bytes, and Class) on the new matrix that will be created once the import process is finished. Make sure the “Import” checkbox for ‘Davis_Weather...’ is checked.



Step 6: Right click on the variable Name “Davis_Weather...” and select rename variable. Rename the variable to ‘weather’. Now click ‘Finish’.

Step 7: Now you have a new variable called ‘weather’ with size 365x2 in the workspace. In the workspace, right click on the ‘weather’ variable and select “Save As”. Save it as “weather.mat” at your local directory.

Step 8: In the future, instead of using ‘uimport’ to import the weather database (requires manual interaction), simply type “load weather.mat” in the MATLAB command prompt to load this database.

Task 1: Comment out code for task 0, and add “load weather.mat” to load the weather database. This is how you will turn in your code. The TA will have their copy of “weather.mat” for grading. You do not need to submit your “weather.mat”.

Task 2: Use a MATLAB command to find the *size* (number of rows and columns) of the ‘weather’ matrix.

Task 3: Use a MATLAB command to find the *length* of the ‘weather’ matrix.

Task 4: How many dimensions are in the matrix created above? Hint: ‘ndims()’

Task 5: Find the *maximum* daily temperature in Davis from 7/1/2010 to 6/30/2011. Convert answer to Celsius. Hint: Temp. in Celsius = (5/9)*(Temp. in Fahrenheit-32)

Lab: Two Dimensional Vectors

Task 6: Calculate the *yearly* precipitation in Davis from 7/1/2010 to 6/30/2011. Convert answer to centimeters. Hint: 1 inch = 2.54 cm

In 'Davis_Weather_7012010_6302011.csv', the daily average temperature (in Fahrenheit) is in column 1 and precipitation (in inches) is in column 2. Each row is one day, so there are 365 rows. The first and last row corresponds to 7/1/2010 and 6/30/2011, respectively.

To help with remaining tasks, you may wish to use the data in Table 1 below.

Month	Starting index in database	Ending index in database
July	1	31
August	32	62
September	63	92
October	93	123
November	124	153
December	154	184
January	185	215
February	216	243
March	244	274
April	275	304
May	305	335
June	336	365

Table 1. Starting and ending indexes for each month in the .csv file.

Task 7: Use MATLAB functions 'mean()' and 'std()' to find the mean and standard deviation of temperature and precipitation for the two months specified by the teaching assistant. When using the 'std()' function, make sure to set the second argument to '1'. For example, 'std(vec, 1)'

Task 8: Using formulas below, re-calculate the mean and standard deviation of the solar radiation for the two months in Task 7. Compare results to Task 7, do they match? You may use 'mean()' in your solution for standard deviation. You may not use 'mean()' to calculate mean, and 'std()' to calculate standard deviation.

$$\text{Arithmetic Mean, } \mu = \frac{1}{N} \sum_{i=1}^N a_i$$

where a_i is a data sample in the a set of data with N data samples

$$\text{Standard Deviation, } \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_i - \mu)^2}$$

where μ is the mean calculated above

Lab: Two Dimensional Vectors

We will now plot the mean temperature for the two months on a 2D figure together with error bars. The error bar is the standard deviation for that month. Hint: see 5.6 in textbook.

Task 9: Plot the mean temperature for the two months by using `'bar([1 2], [<mean1> <mean2>])'`. The vector `'[1 2]'` represent the months on the x-axis. Replace quantities in `<>` with your own variables/values.

Task 10: Place the bar plot on hold by typing `'hold'`.

Task 11: Plot a red errobar by using `'errorbar([<mean1> <mean2>], [<std_dev1> <std_dev2>], 'r*')'`. Replace quantities in `<>` with your own variables/values.

Lab: Two Dimensional Vectors

Question 2 (20 minutes): Analysis of Solar Radiation Data

The comma separated value file “SAC_AP_2005.csv” contains *hourly* weather and solar data (for the year 2005) from a weather station at Sacramento Executive Airport. You can open this file with any text editor or spreadsheet tool to view its contents. In this question we will import the data into MATLAB using ‘xlsread()’.

The first column of the database contains values from 1 to 24 where 1 corresponds to 1:00 (24 hour format), and 24 corresponds to 24:00. The first row corresponds to 1:00 on 1/1/2005 and the last row corresponds to 24:00 on 12/31/2005. The solar radiation data is in column 2. The units are Wh/m² (Watt-hour per square meter), which is energy per area.

Task 0: Type the following into MATLAB’s command prompt.

```
inputVec = xlsread('SAC_AP_2005_solar.csv');
```

You now have a matrix in the workspace called ‘inputVec’. Contents and dimensions of ‘inputVec’ are described above.

In the workspace, right click on ‘inputVec’ and select “Save As”. Save it as “solar.mat” at your local directory.

In the future, instead of using ‘xlsread()’ everytime to import the weather database (which is slow for large files), simply type “load solar.mat” in the MATLAB command prompt to load this database.

Task 1: Comment out code for task 0, and add “load solar.mat” to load the solar radiation database. This is how you will turn in your code. The TA will have their copy of “solar.mat” for grading. You do not need to submit your “solar.mat”.

Task 2: Use a MATLAB command to find the *size* (number of rows and columns) of the ‘solar’ matrix.

Task 3: Use a MATLAB command to find the *length* of the ‘solar’ matrix.

Task 4: How many dimensions are in the matrix created above? Hint: ‘ndims()’

Task 5: Find the *maximum* hourly solar radiation in Sacramento.

Task 6: For an area of 1 kilometer square, how much solar energy can be generated in a year in Sacramento? Hint: 1 kilometer = 1000 meters

Hint: yearly solar radiation * 1 kilometer square = total energy generated.

Lab: Two Dimensional Vectors

To help with remaining tasks, you may wish to use the data in Table 2 below.

Month	Starting index in database	Ending index in database
January	1	744
February	745	1416
March	1417	2160
April	2161	2880
May	2881	3624
June	3625	4344
July	4345	5088
August	5089	5832
September	5833	6552
October	6553	7296
November	7297	8016
December	8017	8760

Table 2. Starting and Ending indexes for each month in the .csv file.

Task 7: Using ‘mean()’ and ‘std()’, calculate the mean and the standard deviation of the solar radiation for the two months given by the teaching assistant. When using the ‘std()’ function, make sure to set the second argument to ‘1’. For example, ‘std(vec, 1)’

Task 8: Using formulas below, re-calculate the mean and standard deviation of the solar radiation for the two months in Task 7. Compare results to Task 7, do they match? You may use ‘mean()’ in your solution for standard deviation. You may not use ‘mean()’ to calculate mean, and ‘std()’ to calculate standard deviation.

$$\text{Arithmetic Mean, } \mu = \frac{1}{N} \sum_{i=1}^N a_i$$

where a_i is a data sample in the a set of data with N data samples

$$\text{Standard Deviation, } \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_i - \mu)^2}$$

where μ is the mean calculated above

Task 9: Plot the mean of solar radiation for the two months on a 2D figure together with error bars. The error bar is the standard deviation for that month. Hint: see section 5.6 in textbook.

Lab: Two Dimensional Vectors

Question 3 (20 minutes): Defects in Solar Panels

Defects in Solar Panel Fabrication

A single solar panel consists of photodiodes connected together so the power available from each photodiode contributes to the solar panel's total output. See figure 1 below for a graphical description.

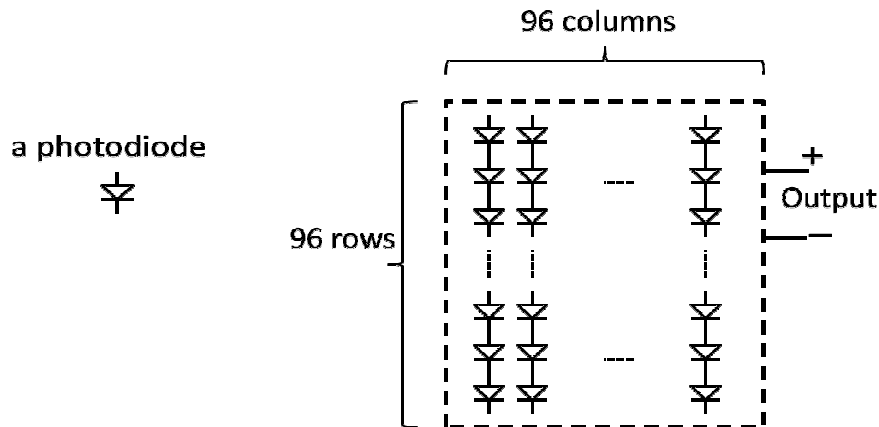


Figure 1. 96x96 matrix of photodiodes inside a solar panel.

Occasionally some photodiodes will become defected (shorted out) due to manufacturing imperfections (shown in red in Figure 2) and this affects the total output power. A defected photodiode contributes no voltage to the panel's overall output.

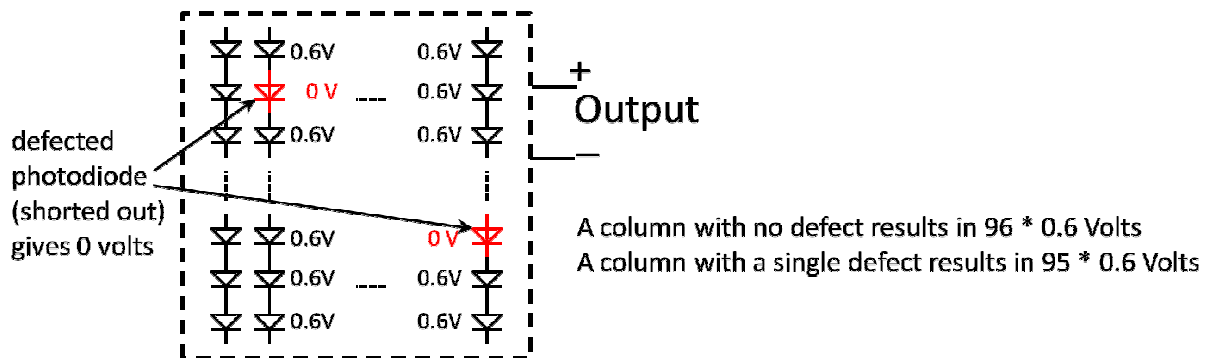


Figure 2. Solar panel with defects shown in red, and their effects on the output.

Lab: Two Dimensional Vectors

You will write a program to calculate the estimated output power of solar panels constructed from 96 rows by 96 columns of photodiodes. We will make the following assumptions for simplicity:

- 1) The total output voltage of the solar panel is determined by the smallest output voltage of any column in the solar panel.
- 2) The maximum current in a column is 100 milli-amperes (mA), and for every defect in a column, the output current for that column decreases by 1 mA. This assumption is described by the equation below:

$$\text{Output Current per column} = 100 \text{ mA} - 1 \text{ mA} * \text{Number of defects in that column}$$

$$\text{Total Solar Panel Output Current} = \text{Sum of all output currents per column}$$

Note: The probability that a photodiode becomes defected in manufacturing will be given to you by the teaching assistant.

Example:

Figure 3 below shows a solar panel that contains a 6x6 matrix of photodiodes.

Table 2 below shows the currents and voltages for each column, the total output current and voltage, and the solar panel's output power.

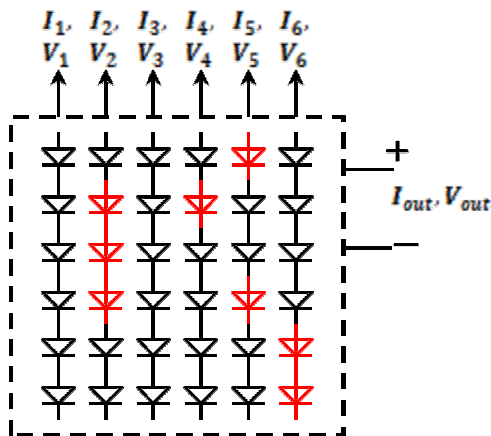


Figure 3.

I1, V1	100 mA, 3.6 V
I2, V2	97 mA, 1.8 V
I3, V3	100 mA, 3.6 V
I4, V4	99 mA, 3.0 V
I5, V5	98 mA, 2.4 V
I6, V6	98 mA, 2.4 V
I _{out} , V _{out}	592 mA, 1.8 V
Panel's output power	592 mA*1.8V*90% = 0.959 Watts

Table 2.

Lab: Two Dimensional Vectors

Calculate the total output power by following the steps below and completing the associated tasks.

Step 1: We will start by modeling the solar panel with a 96x96 matrix with each entry initialized to 0.6 Volts. 0.6 Volts is an estimate of the open circuit voltage of a single photodiode.

Task 1: Write code to create a 96x96 matrix with each entry initialized to 0.6. Call this matrix **panel**. Hint: use 'zeros()' or 'ones()'.

Step 2: We will now learn how to model the defect probability. We will use MATLAB's random number generator, 'randi()' (section 3.7 of textbook) to help us. We will create a new matrix (call it **rand_int**) with size 96x96, and use 'randi()' to generate a random number (between 1 and 100) for each entry.

You can think of the operation of 'randi()' as follows: when generating a random number, 'randi()' "throws" a N-sided dice. (where N is 100 in our case) Type 'help randi' and read through the documentation before asking for help.

Task 2: Write a line of code to generate a 96x96 matrix of uniformly distributed random numbers between 1 and 100. Call this matrix **rand_int**.

Step 3: We will now modify **rand_int** accordingly. Remember that each matrix entry models a photodiode, so **rand_int** from the Task 2 actually models which photodiode is defected, and which is not. Any entry in **rand_int** that contains a value less than or equal to the defect probability will be consider defected.

Task 3: Write a line of code that modifies **rand_int** according to the following rules:

- If entry is less than or equal to the defect probability, replace it with 0.
- If entry is greater than the defect probability, replace it with 1.

Example: The command `aVec(aVec>20)=1` sets all entries greater than 20 in `aVec` to 1

Step 4: Finally, will set the defected photodiode's voltage in the solar panel to zero while keeping the good ones at 0.6 Volts.

Lab: Two Dimensional Vectors

Task 4: Multiply `rand_int` with `panel` to obtain the results.

Step 5: Find the column that has the minimum total voltage.

Task 5: Sum all elements in each column and find the column that gives the smallest sum. This is the solar panel's output voltage.

Step 6: Compute the solar panel's total output current.

Task 6: Count the number of defects in each column. Hint: 'find()'

Task 7: Compute the total output current per column using the equation below.

$$\text{Output Current per column} = 100 \text{ mA} - 1 \text{ mA} * \text{Number of defects in that column}$$

Task 8: Compute the panel's total output current by summing the output current for all columns.

Step 7: Compute the Total output power.

Task 9: Multiply the results from Task 8 with Task 5. The total output power is approximately 90% of this value.

Task 10: Plot the 2D matrix you used to represent the defects in the solar panel.

You may use any following methods: `contour(1:96, 1:96, panel)`, `surf(1:96, 1:96, panel)`, `pcolor(panel)`.

Lab: Two Dimensional Vectors

Question 4 (20 minutes): Variations in Solar Panels

Variations in Solar Panel Fabrication

A single solar panel consists of photodiodes connected together so the power available from each photodiode contributes to the solar panel's total output. See figure 1 below for a graphical description.

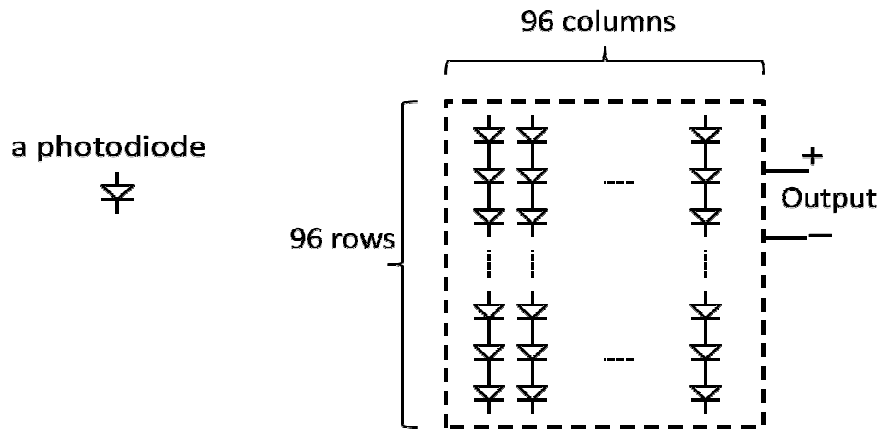


Figure 1. 96x96 matrix of photodiodes inside a single solar panel.

Under direct sunlight, each photodiode will typically output 0.6 volts, which gives a total output voltage of 57.6 volts ($=96 \text{ rows} * 0.6 \text{ volts}$). However, due to variations in manufacturing, the output voltage deviates slightly from 0.6 volts. This variation can be modeled as a normally distributed random process, with a mean and a standard deviation. As you can see, the variation affects the total output power of the solar panel. (power = current * voltage)

You will write a program to estimate the output power of a solar panel constructed from 96 rows by 96 columns of photodiodes while accounting for variations.

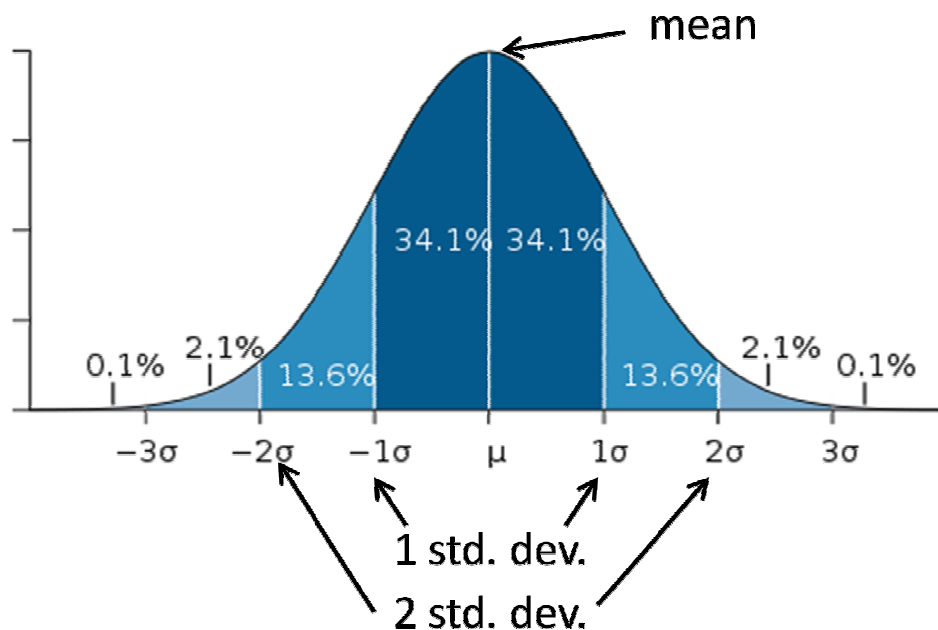
Before starting, read the brief information below on normally distributed random processes.

The normal distribution curve (“bell curve”) is shown below with the mean, and the one and two standard deviation points labeled. This is essentially a histogram. The y-axis is the frequency (how often the corresponding x value shows up) while the x-axis plots the value. As you can see, in a normally

Lab: Two Dimensional Vectors

distributed random experiment, the most likely outcome is the mean, followed by values within \pm one and two standard deviations from the mean. Finally, outcome outside of \pm 3 standard deviations from the mean are unlikely, but not impossible. The probability associated with each outcome region is (the shaded area under the curve) labeled under the curve. The probability for an experiment to have an outcome within \pm one standard deviation is 68.2(=34.1+34.1) percent, while the probability to obtain an outcome outside of \pm 3 standard deviations is < 0.2 percent.

Aside: throwing a six sided dice is a *uniform distribution* (flat line with non-zero y intercept) because each outcome has equal probability ($1/6$). If the dice is thrown a large number of times, on average each of the six outcomes will show up for the same number of times.



Complete the tasks below to construct the program.

We will start by modeling the solar panel with a 96x96 matrix with each entry initialized to 0.6 Volts. 0.6 Volts is an estimate of the open circuit voltage of a single photodiode.

Lab: Two Dimensional Vectors

Task 1: Write code to create a 96x96 matrix with each entry initialized to 0.6. Call this matrix **panel**. Hint: use 'zeros()' or 'ones()'.

Next, we will model the variations.

Task 2: Use MATLAB's random number generator, 'randn()' (section 3.7 of textbook) to modify **panel**, such that it is a 96x96 matrix of random number with mean equal to 0.6, and a standard deviation given by the teaching assistant. Use the formula below. You will fill in the arguments to randn().

$$\text{Random Number } X = \text{standard deviation} * \text{randn}(\dots) + \text{mean}$$

Task 3: Find the column in the 96x96 matrix that has the smallest total voltage (sum of all elements in a column). This will be the solar panel's output voltage.

Task 4: For each column, find the number of photodiodes whose voltage is within 1 standard deviation above OR below the mean, which is 0.6 Volts.

Task 5: For each column, find the number of photodiodes whose voltage is within 2 standard deviation above OR below the mean, which is 0.6 Volts.

Task 6: For each column, find the number of photodiodes whose voltage is within 3 standard deviation above OR below the mean, which is 0.6 Volts.

Task 7: Each column has 100 milli-amps of output current. Compute the panel's total output current by summing the output currents in all columns.

Task 8: The solar panel's output power is calculated by multiplying the output current with the output voltage. The output current and voltage are results from Task 3 and Task 7. The solar panel's output power is approximately 90 percent of the product.

Task 9: Plot the 2D matrix you used to represent the defects in the solar panel. You may use any following methods: *contour(1:96, 1:96, panel)*, *surf(1:96, 1:96, panel)*, *pcolor(panel)*.

Lab: Two Dimensional Vectors

Question 5 (15 minutes): Hybrid Electric Vehicle Sales

The spreadsheet “hev_sales.xlsx” (source: department of energy) contains sales data for hybrid electric vehicles (HEV) from 1999 to 2010.

The data you see (under the “Data” tab) in this spreadsheet is sorted in alphabetical order by the name of car manufactures. The spreadsheet has been saved into a .mat file.

Type “load hev.mat” at the MATLAB command prompt to import this table into MATLAB as the matrix “hev”. Only columns D to O, and rows 4 to 35 in the spreadsheet are imported into “hev”. Hint: see the size of “hev” by typing “size(hev)”

Important: You may use Excel to check your results, but you must provide MATLAB code for all Tasks to receive ANY credit.

Task 1: Calculate the total number of Ford Escape sold from 1999 to 2010.

Task 2: Calculate the total number of Nissan Altima sold from 1999 to 2010.

Task 3: Calculate the number of HEVs sold in 2000.

Task 4: Calculate the number of HEVs sold in 2010.

Task 5: What is the percent increase in HEV sales from 2000 to 2010?

Task 6: What is the average number of total HEVs sold each year from 1999 to 2010?

Task 7: How many more HEVs did Honda sell compared to Toyota from 1999 to 2010? Look at the spreadsheet to see which row corresponds to which car company.

Task 8: Based on this data, at least how many HEVs are out at the end of 2010?

Task 9: Using bar(), plot the number of Prius sold each year from 1999 to 2010.

Task 10: Using bar(), plot the total number of HEVs sold each year from 1999 to 2010.

Lab: Two Dimensional Vectors

Question 6 (15 minutes): Means of Transportation to Work

The spreadsheet “work_transportation.xlsx” (source: department of energy) contains data for how people commute to work from 1980, 1990, 2000, and 2009.

The data you see under the ‘Data’ tab in the spreadsheet are inside the file “work.mat”.

Type “load work.mat” at the command prompt to load this database into MATLAB as the matrix “work”. Only columns C to F, and rows 4 to 17 in the spreadsheet are imported into “hev”. Hint: see the size of “work” by typing “size(work)”

Important: You may use Excel to check your results, but you must provide MATLAB code for all Tasks to receive ANY credit.

Task 1: According to this database, how many total workers are there in 1990?

Task 2: Using data from all four years, what percent of people drive alone to work?

Task 3: Comparing 2009 to 1980, what is the percent decrease of the people who walk to work? Note: percent decrease = absolute value of the difference between the two data, divided by the original data.

Task 4: Assuming the average distance traveled by commuters who drive alone is 30 miles a day, and assume the average miles-per-gallon is 27 Miles. How many gallons of gasoline is consumed everyday by commuters who drive alone?

Task 5: If 30% of the people who drive alone to work now take the bus, how many people utilize public transformation in total?

Task 6: If for every 500 additional people who take the bus, the bus fare can be reduced by 1 cent. How much can the bus fare be reduced if 30% of the people who drive alone to work now take the bus? Assuming bus fare per person is \$1 per person per ride.

Task 7: Using data for the four years, how many people bike to work on average?

Task 8: Using plot(), plot the number of people who take subway to work, versus the four years.

Lab: Two Dimensional Vectors

Question 7 (15 minutes): Alternative Fuel Vehicles

Alternative Fuel Vehicles are vehicles that partially or completely utilize a different energy source as fuel (rather than conventional gasoline).

The spreadsheet “afv.xlsx” (source: department of energy) contains data for the number of Alternative Fuel Vehicles (AFV) in use from 1992 to 2008.

The data you see under the ‘Data’ tab in the spreadsheet are inside the file “afv.mat”.

Type “load afv.mat” at the command prompt to load this database into MATLAB as the matrix “afv”. Only columns C to K, and rows 4 to 20 in the spreadsheet are imported into “afv”. Hint: see the size of “afv” by typing “size(afv)”

Important: You may use Excel to check your results, but you must provide MATLAB code for all Tasks to receive ANY credit.

Task 1: How many total AFVs were there in the year 2004?

Task 2: Which type of AFV was the most common in 1995?

Task 3: In 2008, what percent of total AFVs are electricity based?

Task 4: What is the average increase in number of AFVs that uses E85 per year from 1992 to 2008?

Task 5: What is the average decrease in number of AFVs that uses LPG per year from 1992 to 2008?

Task 6: Using plot(), plot the data for 85% Ethanol (E85) from 1995 to 2008. On the same plot, plot the data for Liquefied Petroleum Gases (LPG), also from 1995 to 2008. By looking at the graph, on which year did E85 surpass LPG? Check your answer by looking at the spreadsheet. Note: x-axis is time, y-axis is the data for the AFV.