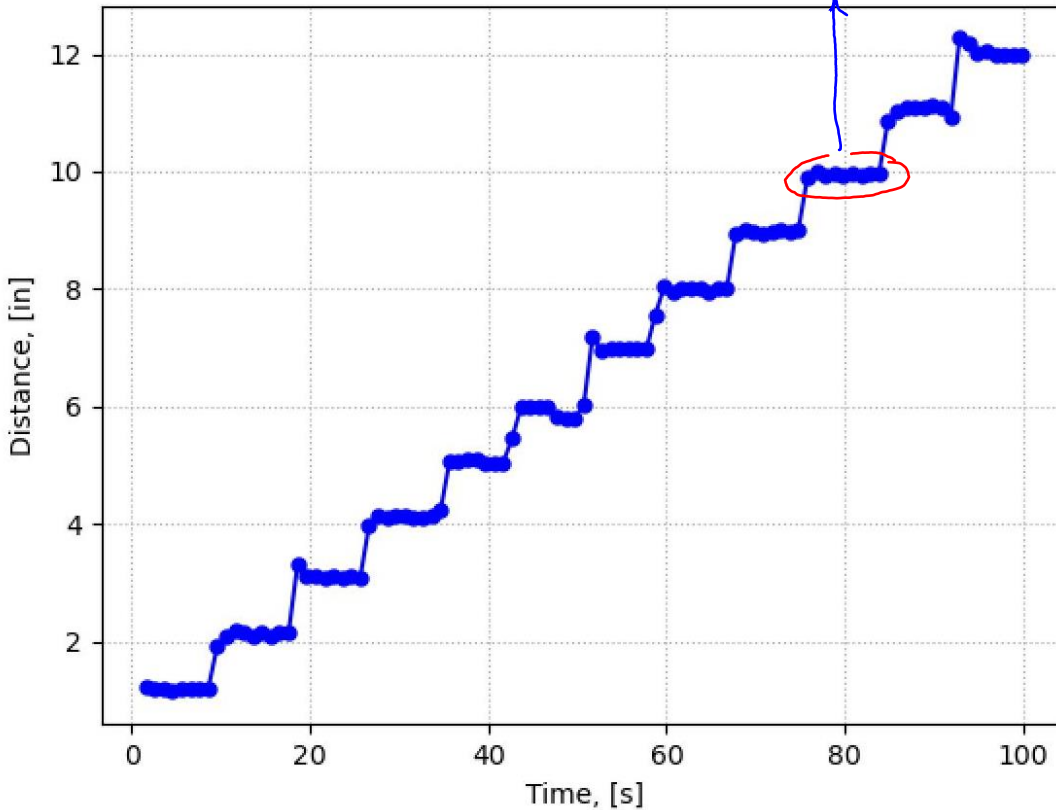


Statistical Analysis of Measured Data

95% Confidence Interval $\left\{ \begin{array}{l} \left[\right] \\ \left| \right. \end{array} \right\}$ mean value
two standard deviations



comma separated skip the header line

```
26 # ----- Import the csv data file that was saved -----
27 dat = np.loadtxt('Distance_Measurement_Wood_block.csv', delimiter=',', skiprows = 1)
28 Time = dat[:,0] # The first column was time in seconds
29 Dist = dat[:,1] # The second column was the distance
```

indexing in Python, C, & LabVIEW starts from zero

```
41 for i in range(1,21):
42     # the 'where' numpy function helps to get the indexes of the values when i=2
43     Indexes = np.where(np.logical_and(Dist >= i-0.5, Dist <= i+0.5))
44     Sorted = Dist[Indexes] # Drop the distance values within each bin 2-0.5 ≤ dist ≤ 2+0.5
45     Mean = np.mean(Sorted) # Calculate the mean value of the bin
46     Std = np.std(Sorted) # Calculate the standard deviation value of the bin
47     Means.append(Mean) # Append the calculated mean value in the array
48     Stds.append(Std) # Append the calculated std in the array
49
```

$i = 1, 2, 3, \dots, 20$ since by default the step is 1
we omitted the 3rd value (1,21,1)

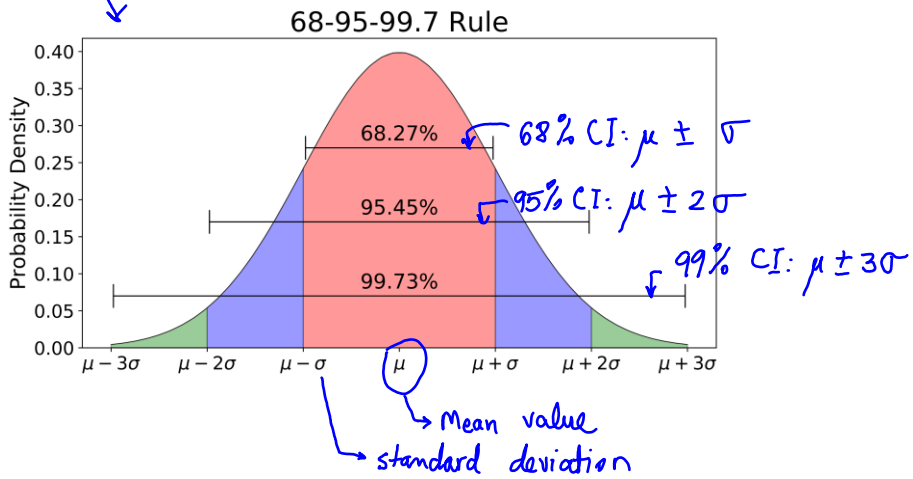
Our true values were 1, 2, 3, ... 20 inch (I went up to 20-inch)

```

54 # Generate an array of equally spaced values from 1 to 20 as our true values
55 TrueValues = np.linspace(1,20,20)
56
57 # Calculate the Confidence Intervals as two standard deviation above and two standard
58 # deviation below the mean value. This is required to get a probability of 95%
59 CI = np.multiply(Stds,2)
60

```

start — End — # of Equally spaced intervals

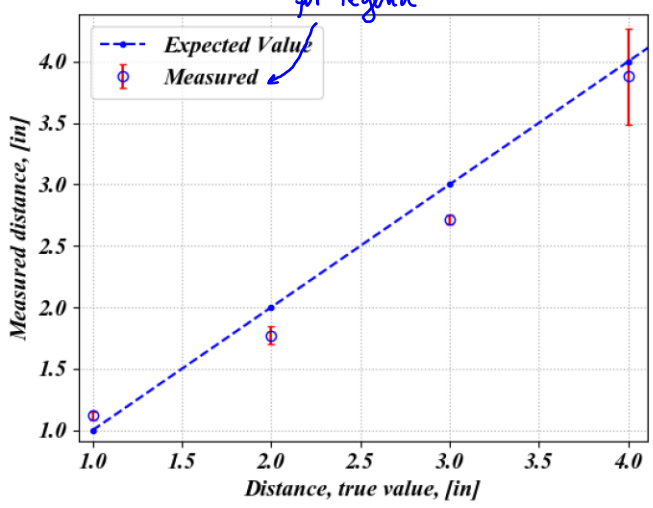


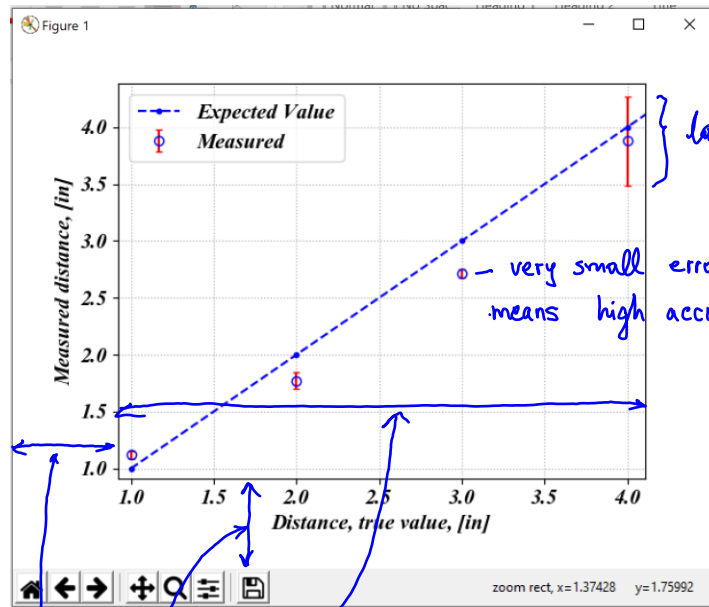
```

61 # ----- Plot the Measured Distance and Its Confidence Intervals -----
62 fig = plt.figure(1)
63 plt.clf()
64 plt.errorbar(TrueValues, Means, yerr=CI, fmt='bo', markerfacecolor='none',
65             capsize=2, ecolor='red', label='Measured')
66
67 plt.plot(TrueValues, TrueValues, '--b.', label='Expected Value')

```

size of the error bars format \pm
 y-axis
 x-axis
 for legend
 since some bars are small \textcircled{I}





```

74 ax = plt.gca()
75 ax.set_position([0.15, 0.17, 0.75, 0.75]) set the margins for our axes
76 plt.legend() show legend
77
78
79 plt.show() since we use IDLE we need it to make visible our figures

```