Data Analysis- Fictional Experiment of Earth's Gravitational Acceleration Using Mathematica

Darin Mihalik & Jonathan Pachter

Spring 2018

Installing Mathematica

To get Mathematica, please go to the following link and follow the instructions to download

https://it.stonybrook.edu/software/title/mathematica

Theoretical Predictions

From a well known theory, we know the acceleration from gravity precisely.

$$g = 9.81\ [\frac{m}{s^2}]$$

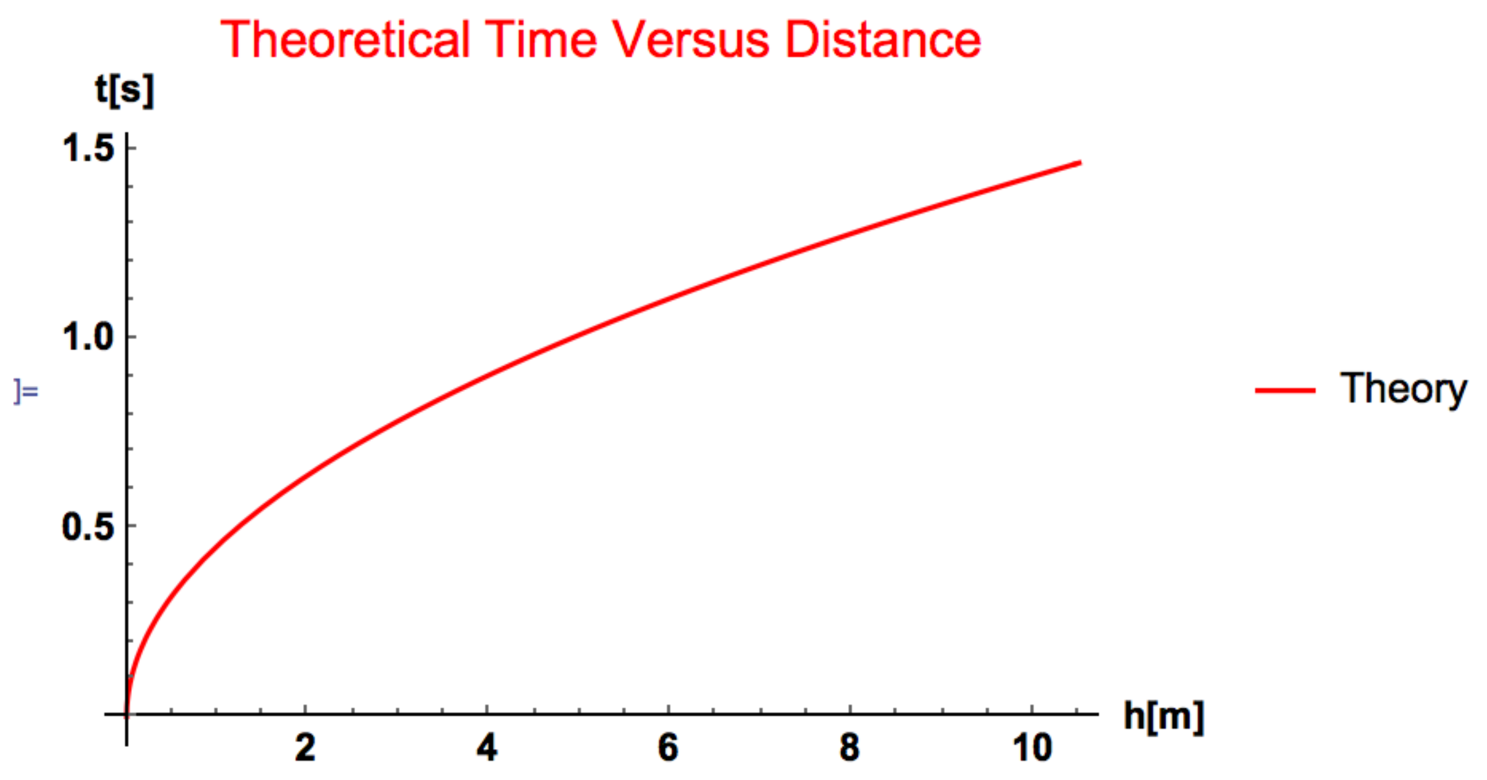
Knowing this information, we know what the theoretical plot of time (in seconds) versus distance fallen (in meters) should look like.

We can show this by plotting the continuous function. Or by solving the equation for *t* at each point *h* and plotting each data point. Or we can do both.

Plot

First, we will plot the continuous function.

```
plot1 = Plot[√(2 h / 9.81), {h, 0, 10.5}, PlotStyle → Red, AxesLabel → {"h[m]", "t[s]"},
  AxesStyle → {{Black, 12, Bold}, {Black, 12, Bold}}, PlotLegends → {"Theory"},
  PlotLabel → Style["Theoretical Time Versus Distance", 16, Red]]
```



* once you type the code above, press **shift+enter** to execute

ListPlot

First, we need to put our data into a list.

```
timetheory = {{0, 0}, {1, .451524}, {2, .638551}, {3, .782062}, {4, .903047},
  {5, 1.00964}, {6, 1.106}, {7, 1.19462}, {8, 1.2771}, {9, 1.35457}, {10, 1.42784}}
```

We can access each part of the list by

```
timetheory[[1, 1]]

0

timetheory[[3, 1]]

2
```

x-values

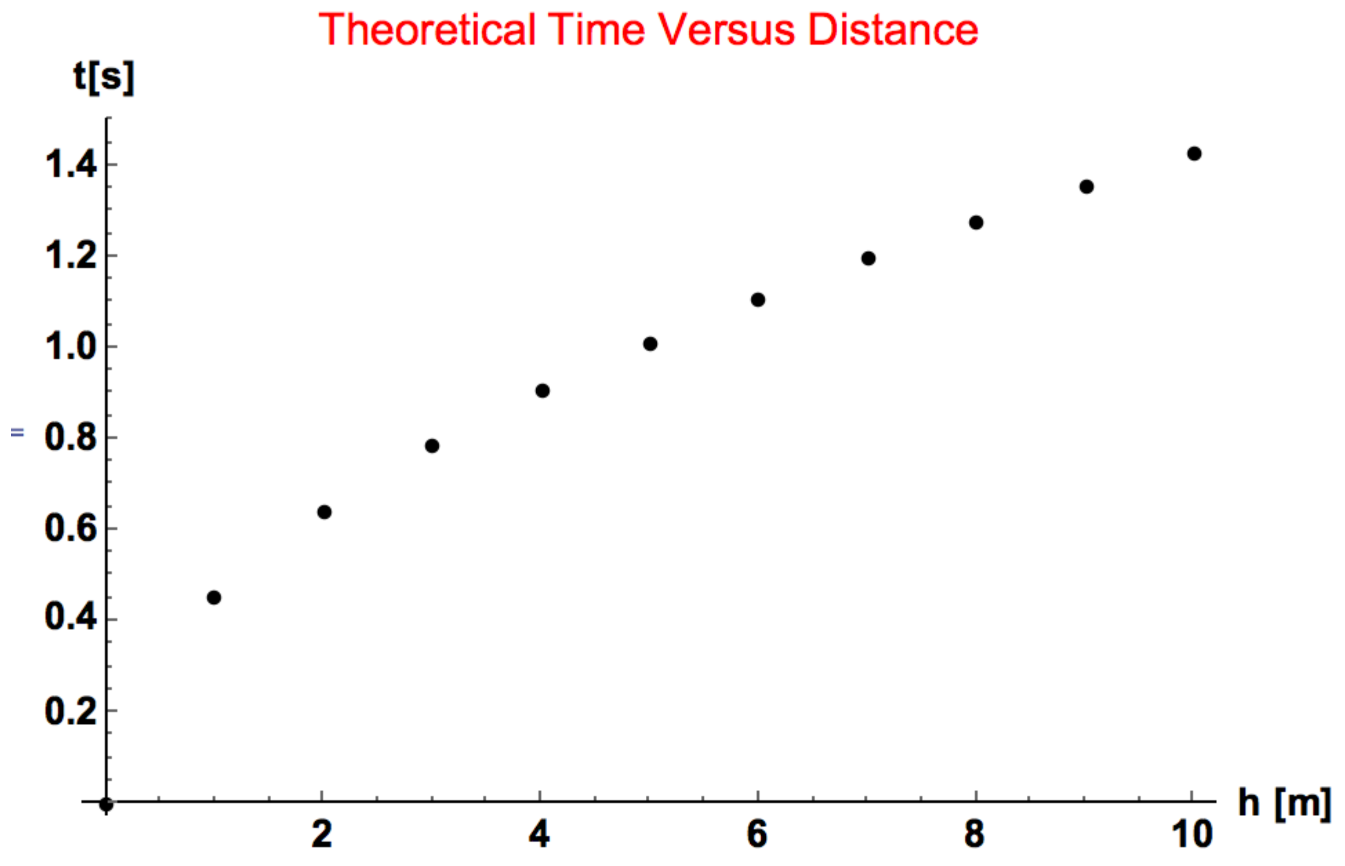
```
timetheory[[6, 2]]

1.00964

timetheory[[1, 2]]

0
```

y-values

ListPlot

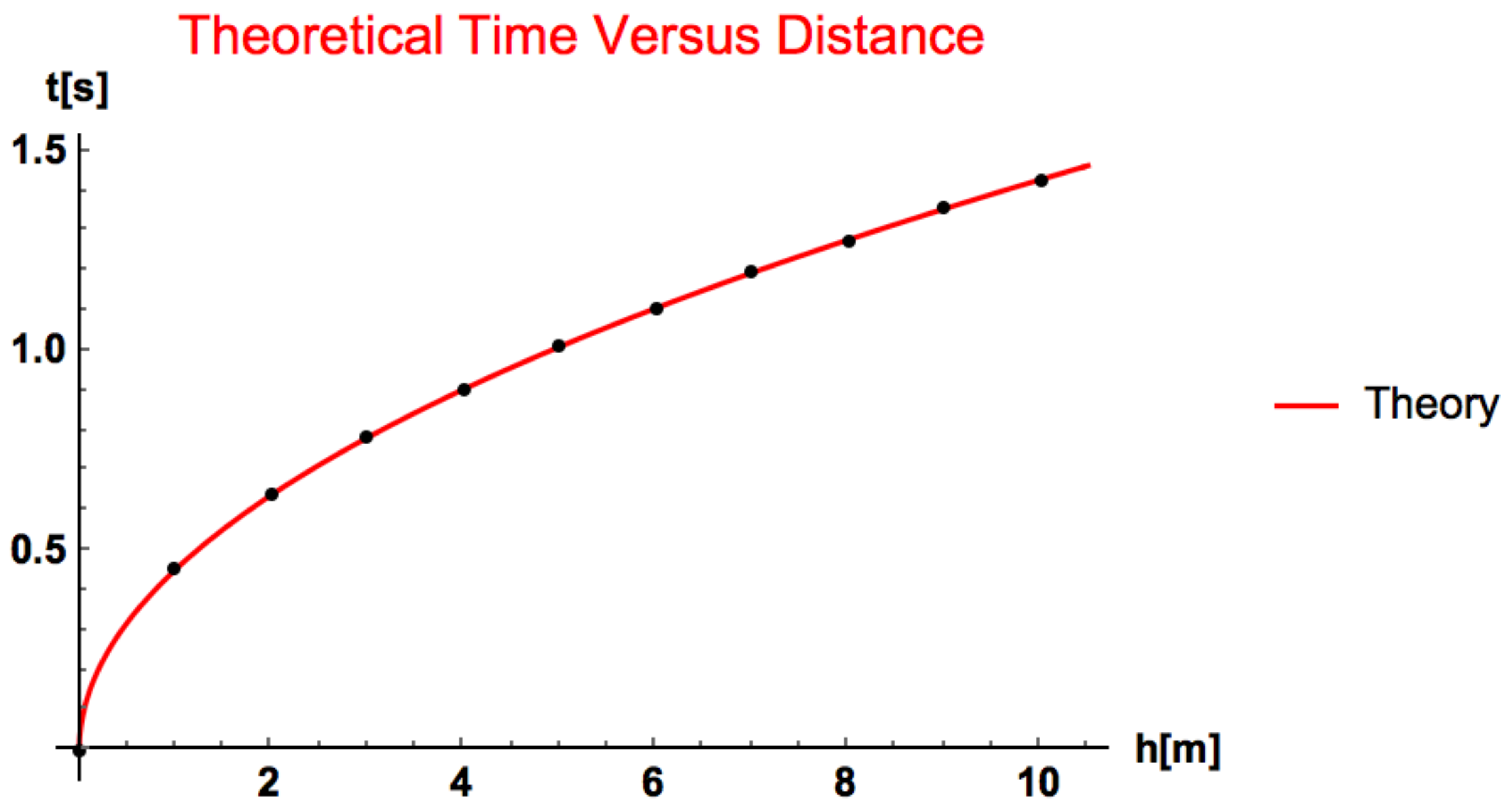
Now, we can plot this dataset at discrete datapoint

```
plot2 = ListPlot[timetheory, PlotStyle → Black, AxesLabel → {"h[m]", "t[s]"},
  AxesStyle → {{Black, 14, Bold}, {Black, 14, Bold}},
  PlotLabel → Style["Theoretical Time Versus Distance", 16, Red]]
```

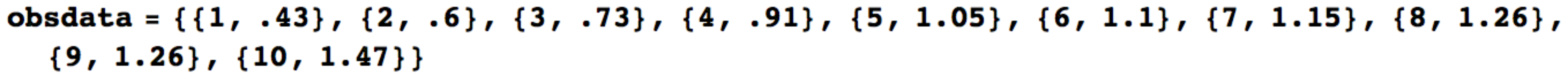


Show Together

```
plot3 = Show[plot1, plot2]
```



Observed Data



```
obsdata = {{1, .43}, {2, .6}, {3, .73}, {4, .91}, {5, 1.05}, {6, 1.1}, {7, 1.15}, {8, 1.26},
  {9, 1.26}, {10, 1.47}}
```

We know that this data should/will resemble a function close to that shown in the theory section.

Solving for the experimentally determined value of "*g*" analytically for this function would be difficult. However, we can mathematically manipulate the equation to make it easier to solve.

Manipulation to a Line

$$t(h) = \sqrt{\frac{2h}{g}}$$

Taking the log of both sides

$$\ln(t) = \frac{1}{2}\ln(h) + \frac{1}{2}\ln(\frac{2}{g})$$

Therefore, by comparison to

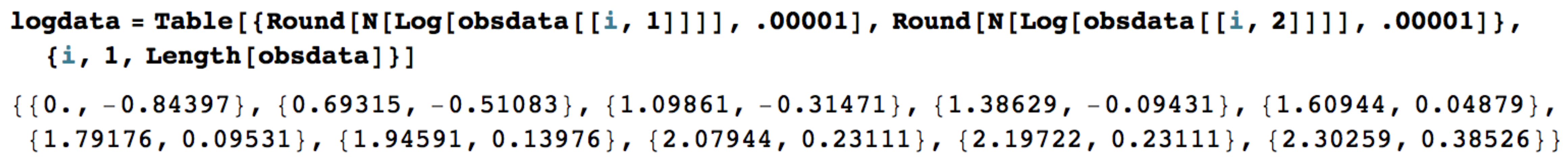
$$y = mx + b$$

We find the following for our parameters

$$y = \ln(t) \qquad m = \frac{1}{2} \qquad x = \ln(h) \qquad b = \frac{1}{2}\ln(\frac{2}{g})$$

Manipulation in Mathematica

We first want to log our data and we will round to the hundred thousandths place for both variables.



```
logdata = Table[{Round[N[Log[obsdata[[i, 1]]]], .00001], Round[N[Log[obsdata[[i, 2]]]], .00001]},
  {i, 1, Length[obsdata]}]

{{0., -0.84397}, {0.69315, -0.51083}, {1.09861, -0.31471}, {1.38629, -0.09431}, {1.60944, 0.04879},
 {1.79176, 0.09531}, {1.94591, 0.13976}, {2.07944, 0.23111}, {2.19722, 0.23111}, {2.30259, 0.38526}}
```

This may look "scary" at first, but it will be easy soon. A few pointers

- Each set of {} and [] closes a function or a certain grouping of code per the Mathematica syntax
- N[] tells Mathematica you want to see a number
- Under "Help", go to **Wolfram Documentation** and type in any function name and it will show you in an interactive window (you can change parameters to see how they would change the output) with the correct syntax…this will be your best friend.

Error Bars

It is always important to include error bars on your plot. To do easily, make a table of your data points and include the ErrorBar function. First, you must call

```
Needs["ErrorBarPlots`"]
```

Now we must calculate what the error bar will be with propagation of error

$$y_i = \ln(t_i) \quad \rightarrow \quad \sigma_{y_i} = |\frac{\partial y_i}{\partial t_i}|\sigma_{t_i} \quad \rightarrow \quad \frac{\sigma_{t_i}}{t_i}$$

Where “t” is the time from the observed data (no log)

Error Bars on the Plot

Now we need to calculate what the error bar will be

```
newerrorbars = Table[.05/obsdata[[i, 2]], {i, 1, Length[logdata]}]
```

Then we need the following command (output is also shown)

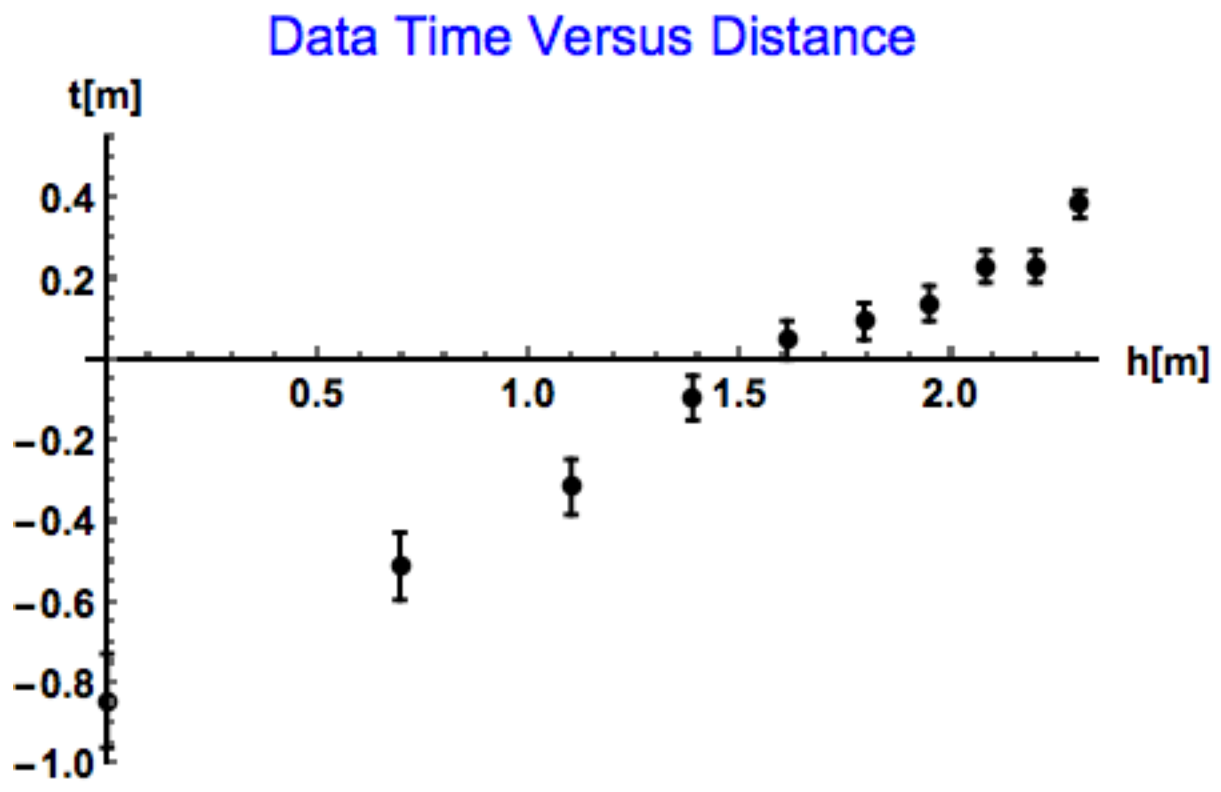
```
errdata = Table[{logdata[[i]], ErrorBar[0, newerrorbars[[i]]]}, {i, 1, Length[obsdata]}]
```

```
{{{0., -0.84397}, ErrorBar[0, 0.116279]}, {{0.69315, -0.51083}, ErrorBar[0, 0.0833333]},
 {{1.09861, -0.31471}, ErrorBar[0, 0.0684932]}, {{1.38629, -0.09431}, ErrorBar[0, 0.0549451]},
 {{1.60944, 0.04879}, ErrorBar[0, 0.047619]}, {{1.79176, 0.09531}, ErrorBar[0, 0.0454545]},
 {{1.94591, 0.13976}, ErrorBar[0, 0.0434783]}, {{2.07944, 0.23111}, ErrorBar[0, 0.0396825]},
 {{2.19722, 0.23111}, ErrorBar[0, 0.0396825]}, {{2.30259, 0.38526}, ErrorBar[0, 0.0340136]}}
```

Now we can call ErrorListPlot, which has the same syntax and command structure as ListPlot

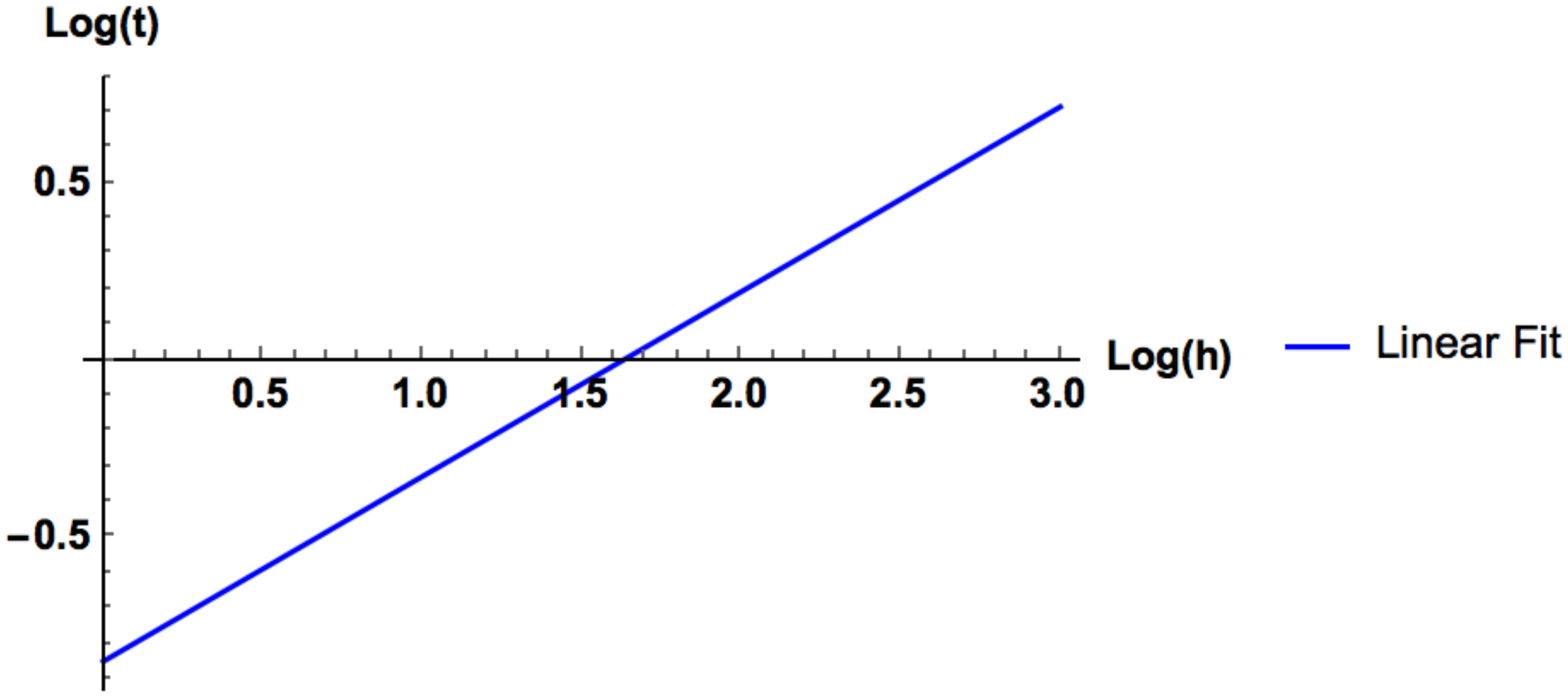
Plotting the Data

```
plot4 = ErrorListPlot[errdata, PlotStyle → {Black, PointSize[.02]}, AxesLabel → {"h[m]", "t[m]"},
  AxesStyle → {{Black, 12, Bold}, {Black, 12, Bold}},
  PlotLabel → Style["Data Time Versus Distance", 16, Blue], PlotRange → {-1, .55}]
```



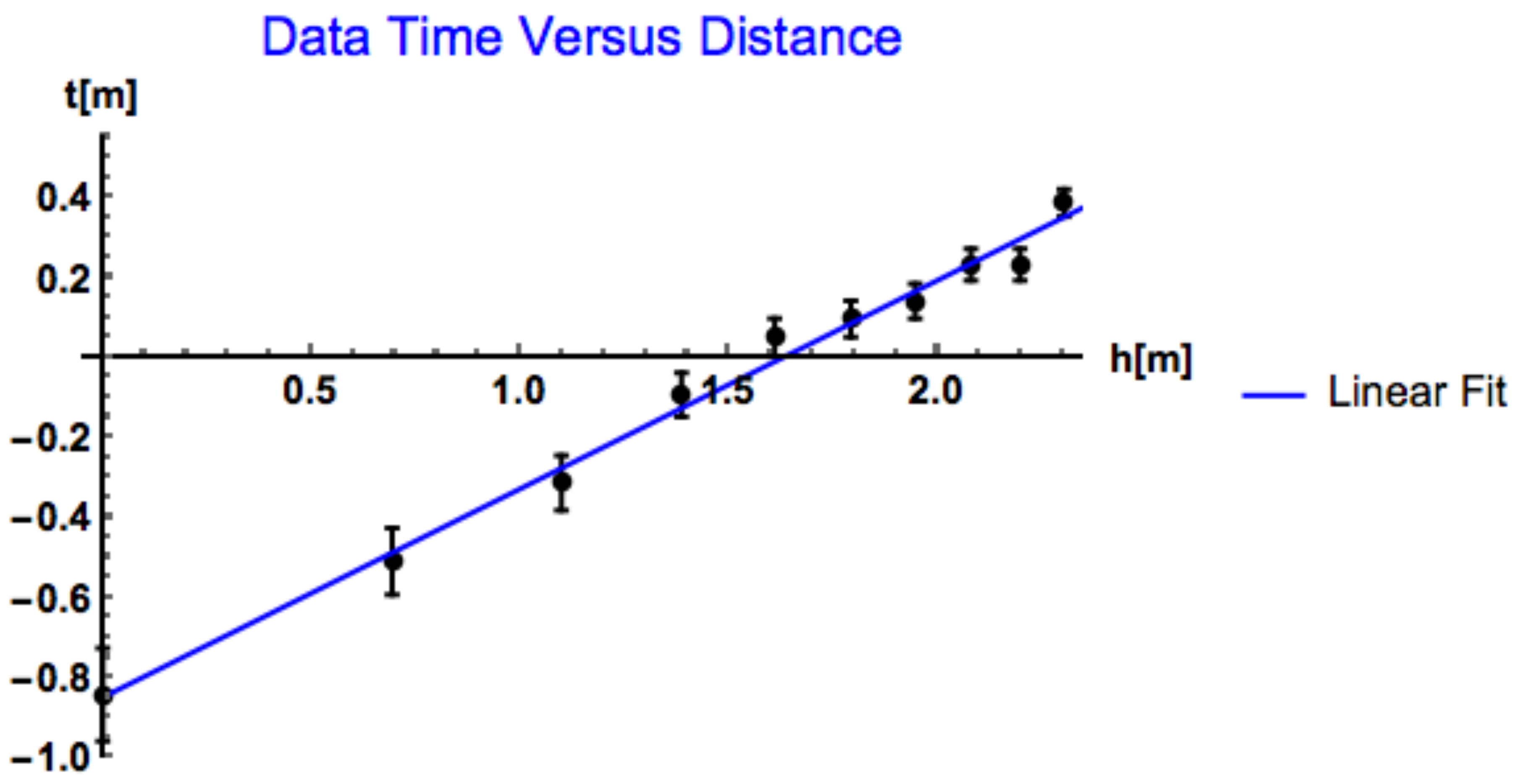
Plotting the Fit

```
plot5 = Plot[model["BestFit"], {h, 0, 3}, PlotStyle → Blue, AxesLabel → {"Log(h)", "Log(t)"},
  AxesStyle → {{Black, 12, Bold}, {Black, 12, Bold}}, PlotLegends → {"Linear Fit"}]
```



Showing Data and Fit

```
Show[plot4, plot5]
```



Solving for “A” and “B”

$$A = \frac{1}{\Delta}(\sum \frac{x_i^2}{\sigma_i^2} \sum \frac{y_i}{\sigma_i^2} - \sum \frac{x_i}{\sigma_i^2} \sum \frac{x_i y_i}{\sigma_i^2})$$

$$\sigma_A^2 = \frac{1}{\Delta} \sum (\frac{x_i^2}{\sigma_i^2})$$

$$\Delta = \sum \frac{1}{\sigma_i^2} \sum \frac{x_i^2}{\sigma_i^2} - (\sum \frac{x_i}{\sigma_i^2})^2$$

$$B = \frac{1}{\Delta}(\sum \frac{1}{\sigma_i^2} \sum \frac{x_i y_i}{\sigma_i^2} - \sum \frac{x_i}{\sigma_i^2} \sum \frac{y_i}{\sigma_i^2})$$

$$\sigma_B^2 = \frac{1}{\Delta} \sum (\frac{1}{\sigma_i^2})$$

What do you notice about these equations?

…..common terms

Solving for One Sum at a Time

$$\sum \frac{x_i}{\sigma_i^2}$$

```
sumX = Total[Table[logdata[[n, 1]]/(logdata[[n, 3]])^2, {n, 1, Length[logdata]}]]
```

$$\sum \frac{x_i y_i}{\sigma_i^2}$$

```
sumXY = N[Total[Table[logdata[[n, 1]] logdata[[n, 2]]/(logdata[[n, 3]])^2, {n, 1, Length[logdata]}]]]
```

We will go over the rest on the first day of class.

Mathematica is relatively user friendly, especially with Wolfram Documentation. In addition, there are numerous forums online. Some of these can get pretty in-depth but for the purposes you will need for this class they should be basic. I have given you examples for just about every plot option you will need.

Due to all of this, please don't come to me for help with a blank Mathematica notebook and tell me you "don't get it". I **will** correct your mistakes and give suggestions but I **will not** just write you a code from scratch…you must put in the effort to get help.

Please come to my office hours for help with Mathematica. They Wednesday 11-12 in the lab (A133).