

## Introduction

Let go of a helium filled balloon and watch it race skyward. How high will it go? Where will it end up? Does it continue hurtling upward at the same rate that it left your hand? Maxime Dehaye Web Developer and avid photographer released a weather balloon with a payload consisting of a range of telemetric devices. Whilst some schools are duplicating this project for themselves, Dehaye recorded the event and posted the results on his YouTube channel. In this investigation you will use the first minute of data from Dehaye's video to predict the altitude of the balloon some 30 minutes later.

# Equipment

- YouTube video:
- TI-Nspire Calculator

# **Collecting the Data**

Open a TI-Nspire document and insert a **Spreadsheet** application. Label the first two columns:

Time	Entar the data:	
nme	Enter the data:	{U,1U,2U,3U,4U,5U,6U}

Altitude Enter this data using the YouTube video

Follow the link to Maxime Dehaye's video on YouTube, watch and record the time (10 second intervals) and altitude for the first minute of the balloon flight.

http://bit.ly/HighAltitudeBalloon

Insert a Data and Statistics application.

Use the [Tab] key to identify and select the independent variable (time) and then again to for the dependent variable (altitude).



•	1.1	HAB 🤜	7	RAD 🚺 🗙
P	A time	<sup>B</sup> altitude	С	D
=				
1	0			
2	10			
3	20			
4	30			
5	40			
B1				•



## Question: 1.

Describe the general relationship between time and altitude for the first minute of flight. Data is linear! (Refer above screen capture.)

© Texas Instruments 2016. You may copy, communicate and modify this material for non-commercial educational purposes provided all acknowledgements associated with this material are maintained.

Author: P. Fox



# **Determining an Equation**

Insert a **Calculator** application and following the next set of instructions *very* carefully.

To check that your lists are complete, try both of the following:

dim(time)

dim(altitude)

Both results should return a '7' as shown, signifying that there are 7 values entered in the time and altitude lists.

<1.1 1.2 1.3 ▶ *HAB ->	RAD 🚺 🗙
dim( <i>time</i> )	7
dim(altitude)	7
1	
	$\sim$

The **[VAR**] key can be used to recall stored variables. This is a very useful way of recalling values stored on other pages within the same problem and also avoids excessive typing.

The regression equation can be generated directly from the calculator; however the following instructions show how this equation is generated. In this example the equation will be of the form: y = a + bx. The values for  $s_x$  and  $s_y$  will be calculated directly from the data. While the value for r can also be computed from the data, in this activity it will be approximated.

The regression equation: y = a + bx can be determined using the formulas:

and  $a = \overline{y} - b\overline{x}$ 

 $b = r \frac{s_y}{s}$ 

The first step is to calculate the statistic:  $s_x = \sqrt{\sum \frac{(x - \overline{x})^2}{n - 1}}$ ,

in this activity *x* represents time. The statistic will be calculated in two steps.

The first step is to calculate:  $(x - \overline{x})^2$ 

Note that the answer will be a list of values that represent the squared differences between each time and the mean time.

<1.1 1.2 1.3 ▶ *HAB ->	RAD 🚺	×
$(time-mean(time))^2$		
{900,400,100,0,100,400	,900}	

Sample data above

© Texas Instruments 2016. You may copy, communicate and modify this material for non-commercial educational purposes provided all acknowledgements associated with this material are maintained.

The second step requires the values to be summed, divided by n-1 and finally square-rooted.

In this example there were 7 values  $\{0, 10, 20 \dots 60\}$  so the previous answer must be divided by 7 - 1 = 6.

In the following calculation, use [Ctrl] + [Enter] to get an approximate value.

Evaluate the expression:

Type: 
$$\sqrt{\frac{sum(time - mean(time))^2}{6}}$$

Note: The previous answer has been hidden.  $\bigcirc$ 

Store your answer in sx.

Store can be done in at least two ways. The 'assigned' equals has been used here.

1.1 1.2 1.3 ▶ *HAB	RAD 🚺 🔀
{ 900, 400, 100, 0, 10	0,400,900}
$\int \frac{\operatorname{sum}((time-\operatorname{mean}(time))^2)}{6}$	21.6025
$s_{x:=} \boxed{\frac{sum((time-mean(time))^2)}{s_{x:=}}}$	21.6025
6	
Sample data above	$\checkmark$

#### Question: 2.

Write down the value you have determined for  $s_{y}$ . Answers will vary:  $s_{y} \approx 21.6$ 

## **Question: 3.**

Repeat the previous two calculator steps for the **altitude** variable. Answers will vary:  $s_v \approx 108$ 

Write down the value you have determined for  $s_{_{\rm V}}$  and store your result on the calculator as  $s_{_{\rm V}}$  .

## Question: 4.

In this case the data correlates almost perfectly so the *r* value is approximately 1. (Sample data typically produces:  $r^2 = 0.999$ ). Using  $r \approx 1$ , combined with the previous two calculations,

determine the value for *b* where: 
$$b = r \frac{s_y}{s_x}$$
,  $b \approx 4.998$ 

The value for *a* can be determined where  $a = \overline{y} - b\overline{x}$ 

Type the following: (shown opposite)

◀ 1.1	1.2 1.3 ▶ *HAB 🗢	RAD 🚺	X
s	$sum ((altitude - mean(altitude))^2)$		2
sy	б		
		107.976	
b:=1	$\frac{Sy}{Sx}$	4.99832	l
a:=n	$mean(altitude) - b \cdot mean(time)$	123.193	l
I			

#### Sample data above

© Texas Instruments 2016. You may copy, communicate and modify this material for non-commercial educational purposes provided all acknowledgements associated with this material are maintained.

Author: P. Fox



<1.1 1.2 1.3 ▶ *HAB ->	RAD 🚺 🗙
( <i>time</i> -mean( <i>time</i> )) <sup>2</sup> { 900,400,100,0,10	00,400,900}
$\sqrt{\frac{\operatorname{sum}((\operatorname{time-mean}(\operatorname{time}))^2)}{6}}$	21.6025
1	



Return to the Data and Statistics application.

From the [Menu] select Analyse followed by Plot Function.

Enter the equation: a + bx

Remember to use a multiplication sign between the b and x.



#### **Question: 5.**

How well does your equation fit the data? The equation is almost a perfect fit!

#### **Question: 6.**

While still in the **Data and Statistics** application, from the menu select **Analyse**, followed by **Regression** and then **Show Linear** (a+bx). Compare the equation generated by the calculator with the one determined in questions 2 to 4.

Students may use a screen capture to show the equations are the same.



## Question: 7.

One reason for determining an equation for a set of data is so that it can be used to forecast or extrapolate. Use your equation to predict the altitude of the weather balloon at the 30 *minute* mark of its journey.  $123.2 + 4.998 \times 1800 \approx 9120 m$ 

#### **Question: 8.**

Return to the YouTube video by Maxime Dehaye "Journey to the Statosphere" and locate the 30 minute mark of the journey. Comment on the accuracy of your prediction. Video (actual) = 9161m. The predication is remarkably accurate given the level of extrapolation!





### **Question: 9.**

It is estimated that the weather balloon should reach a final altitude of approximately 23km to 24km. Determine an approximate time (journey) for which the balloon will burst. Watch the video to see how high it really does go.

The weather balloon actually bursts at 23,227m, 1hr and 27m into the journey.

Using the equation and altitude of 23,000m:  $solve(a + bx \approx 23000m, x) = 4576s$  or 76 minutes

Using the equation and altitude of 24,000m:  $solve(a + bx \approx 24000m, x) = 4777s$  or 79 minutes

The combination of the extraordinary extrapolation and potential speed variation at the higher altitudes contribute to the error margin produced here. Students may believe they have made an error in their calculations considering the extremely reliable prediction at the 30 minute mark of the balloon's flight, so this question serves as a reminder of the limitations of even very precise models!

**Note**: The cruising altitude of a 747 is approximately 10,000m. This means the weather balloon reaches a height more than twice the cruising altitude of a 747, so it makes sense that anyone releasing such a balloon obtains clearance from the appropriate authorities before release!