

Scatter plots

Created using Maple 14.01

Jake Bobowski

```
> restart;
with(StringTools) :
FormatTime("%m-%d-%Y, %H:%M");
"08-04-2012, 21:04"
```

(1)

Scatter plots are useful for plotting experimental data. In Maple, you have to load the *Statistics* package to generate a scatter plot.

```
> with(Statistics) :
```

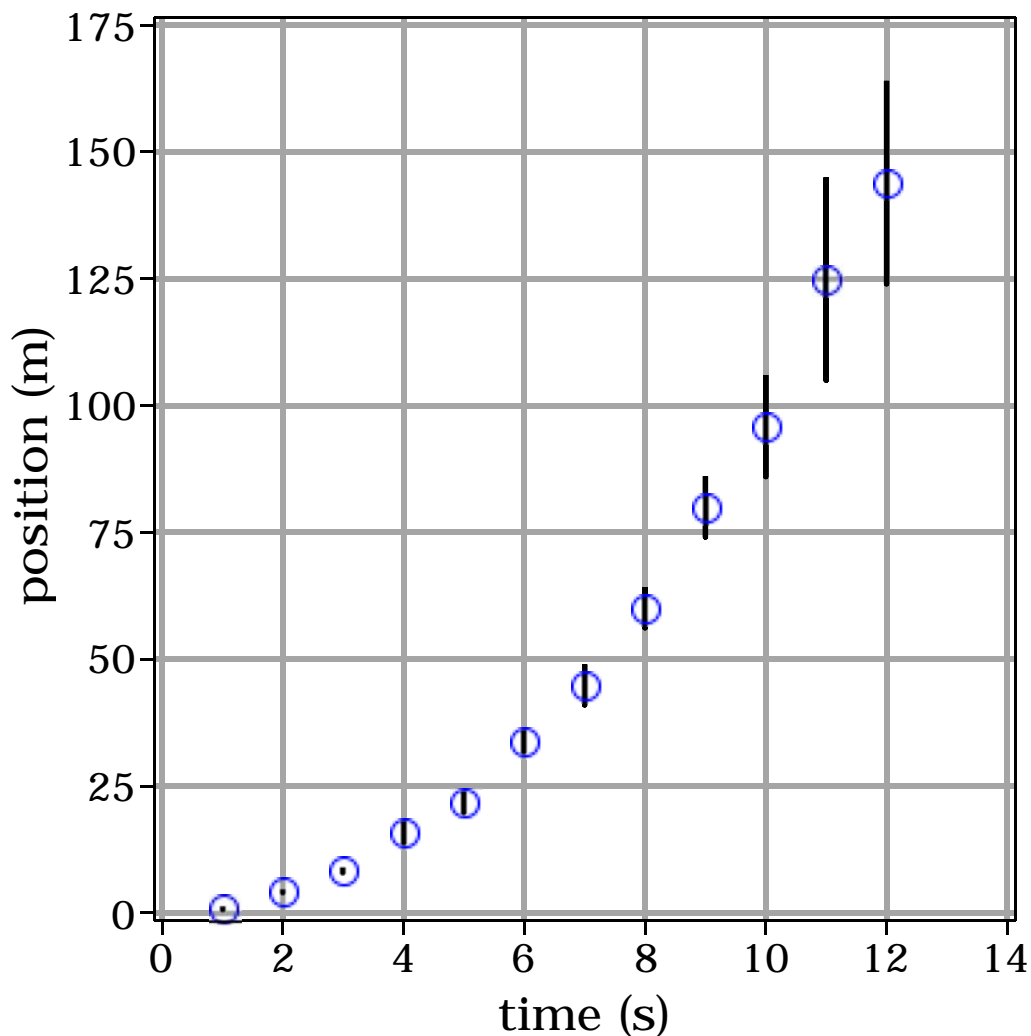
Here's some sample data.

```
> t := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12];
y := [1.02, 4.3, 8.6, 16, 22, 34, 45, 60.2, 80.1, 96, 125, 144];
dy := [.3, .3, .5, 2, 2, 2, 4, 4, 6, 10, 20, 20];
nops(t);
nops(y);
nops(dy);
t := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
y := [1.02, 4.3, 8.6, 16, 22, 34, 45, 60.2, 80.1, 96, 125, 144]
dy := [0.3, 0.3, 0.5, 2, 2, 2, 4, 4, 6, 10, 20, 20]
12
12
12
```

(2)

This is the scatter plot previously generated in "scatter plots.mw".

```
> ScatterPlot(t, y, errors = dy, axes = boxed, view = [0 .. 14, 0 .. 175], labels = [typeset(
"time (s)", typeset("position (m)"), labeldirections = ["horizontal", "vertical"], symbol
= circle, symbolsize = 20, thickness = 2, tickmarks = [8, 8], colour = blue, axesfont
= [Times, 12], labelfont = [Times, 14], axis = [gridlines = [thickness = 2]]);
```

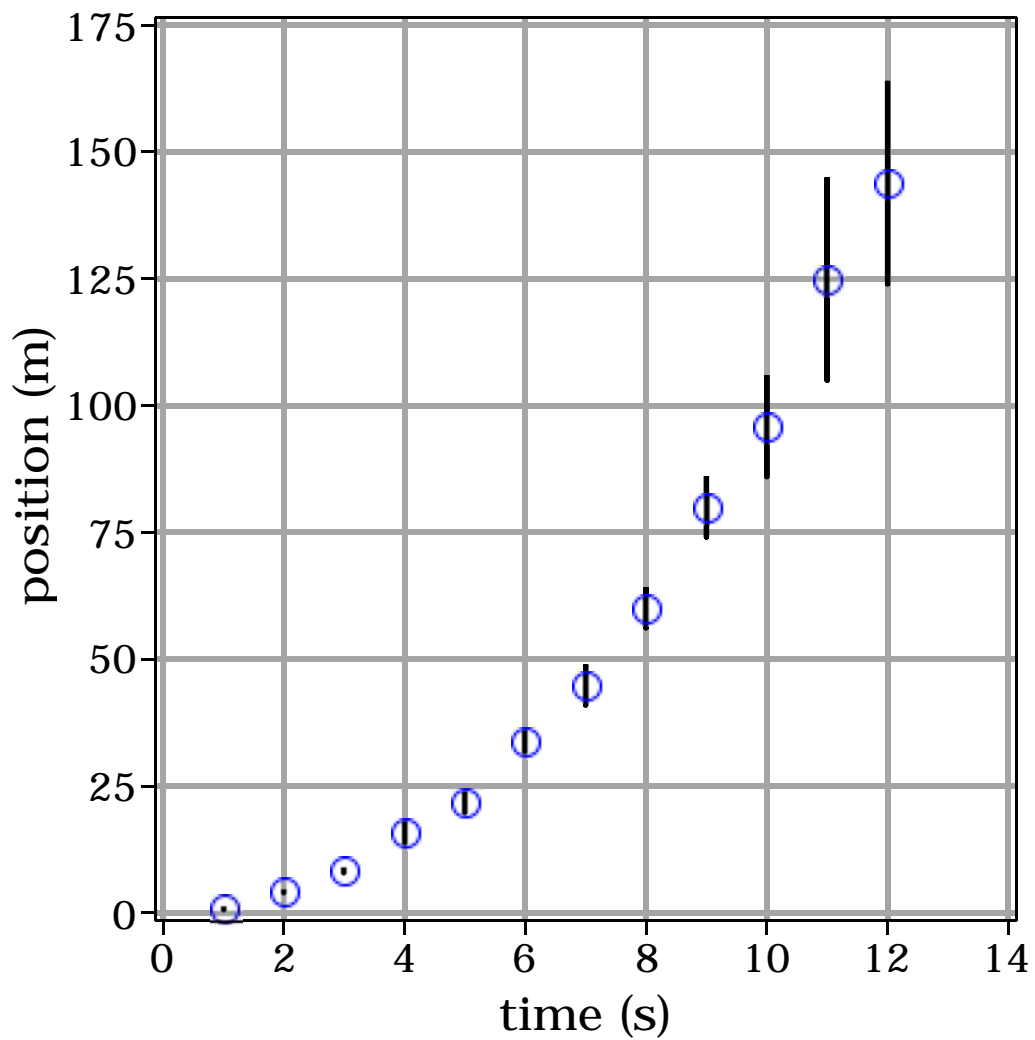


First, let's assign the name "plot1" to this scatter plot

```
> plot1 := ScatterPlot(t, y, yerrors = dy, axes = boxed, view = [0 .. 14 , 0 .. 175], labels
= [typeset("time (s)", typeset("position (m)")), labeldirections = ["horizontal",
"vertical"], symbol = circle, symbolsize = 20, thickness = 2, tickmarks = [8, 8], colour
= blue , axesfont = [Times, 12], labelfont = [Times, 14], axis = [gridlines = [thickness
= 2]]) :
```

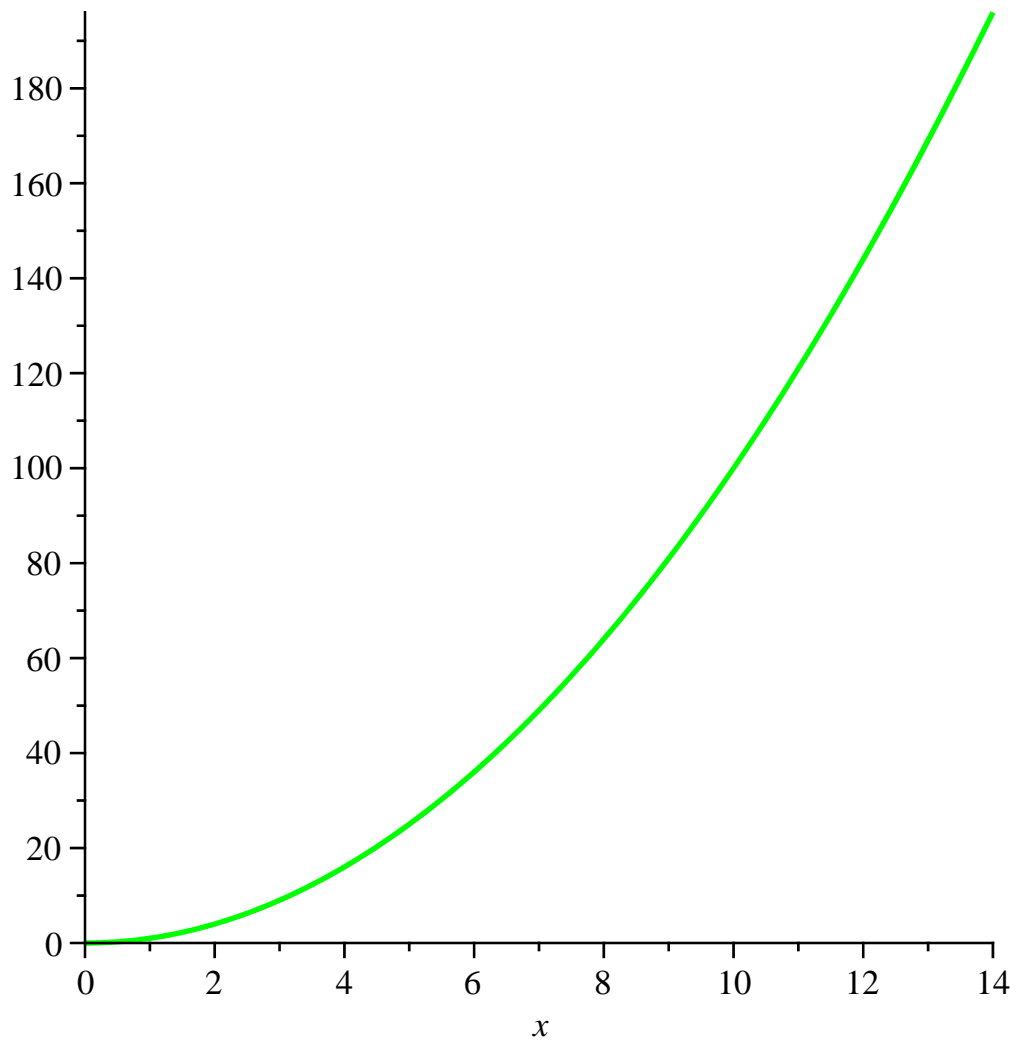
To display *plot1*, use the *display* command. Before doing so, the package *plots* must be loaded. If you know that you're going to be using the *plots* package, it might make sense to place *with(plots)* at the beginning of the document.

```
> with(plots) :
display(plot1);
```



Now use `plot` to display a function that is supposed to model the data. The data approximately follow $y = t^2$. This plot will be called `plot2`.

```
> plot2 := plot(x^2, x=0..14, colour = green, thickness = 2) :
  display(plot2);
```



To show the data and the curve on the same plot, use *display*.

```
> display(plot1, plot2);
```

