

How to make data-plots and charts in L^AT_EX

v 0.1 (draft)

Hans Henrik Sievertsen*

February 28, 2008

Abstract

This note gives a step-by-step introduction to the making of data-plots using `pstricks` in L^AT_EX. The main focus is on mastering the scaletool `\pstScalePoint`. Data and several examples are presented including the code that generated the examples.

1 Introduction

This little note is mainly written for Economics students at the Institute of Economics, University of Copenhagen. I made my first data-plots and charts in L^AT_EX when I wrote my bachelor thesis at the Institute of Economics. As with everything in L^AT_EX the startup costs are high. But - again as with almost everything in L^AT_EX- the final result is beautiful and after a while it gets quite intuitive and easy to use.

A few words of warning: It takes time to learn it, and even if you have learned it well, complicated figures will still take longer time to create in L^AT_EX than in for example “Microsoft Excel”. A good alternative could be to use “R” or other programmes. Even SAS allows you to save the output as “.tex” files - that can be included directly in L^AT_EX.

2 Documentation

I have found *PsTricks* to be the best tool for making graphs, plots and other beautiful empirical graphs. The best documentation is probably *pstricks-add-doc*. However, the best way to learn PsTricks is of course by trial and error. And

*This note is not, in any way, official documentation nor it is guaranteed to be correct. I thank Niels-Jakob Harbo Hansen and Andreas Noack Jensen for helpful suggestions

the best way to start is by having a look at the *tug-pstricks website*, where you can find examples of almost all graph types.

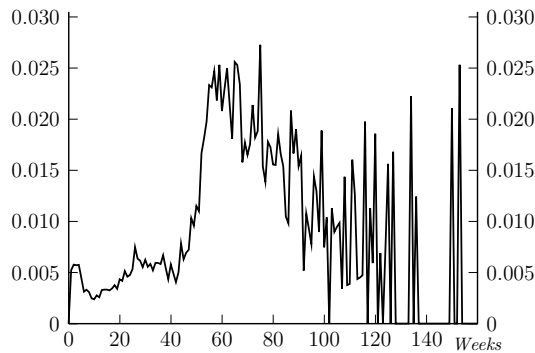
3 My installation

I usually use a MiKTeX installation on Windows machine and TeXnicCenter as an editor. However, the examples in this note should work everywhere. But, you cannot use *pdflatex*.

4 Ok, ok, let's get ready to rumble

To motivate you - let's take a look on a quite easy example of a plot made in graph. Get the code and data and try on your own.

Figure 1: *Example: Kaplan-Meier Hazard rates*



```
----- Start Preamble -----
\documentclass{article}
\usepackage{pst-bar,pst-plot,pstricks-add}
----- End Preamble -----
\readdata{\data}{data1.txt}
\psset{xunit=1 mm,yunit=1 mm}
\pspicture(0,0)(100,75)
  \psaxes[Dx=20,dx=1 cm,0x=0,Dy=0.005,dy=1 cm,tickstyle=below,ticks=5pt](10,10)(90,71)
  \psaxes[xAxis=false,tickstyle=top,Dy=0.005,dy=1cm,tickstyle=below,ticks=-5pt](90,10)(90,71)
  \pstScalePoints(0.5,2000){20 add}{0.005 add}
  \rput(90,6){\footnotesize{\emph{Weeks}}}
  \listplot[linecolor=black,linewidth=1pt, nEnd=160]{\data}
\endpspicture
----- Code end -----
```

4.1 Step 1: the preamble

For the figures in this note it should be enough to add:

```
\usepackage{pst-plot,pstricks-add}
```

It is very important that “pstricks-add” is the last package loaded (of the pst-packages).

4.2 The axes

In figure 1 the axes are drawn by

```
\psaxes[options](startposition)(endposition)
```

. What is important to know, is that the plots, bars or graphs know nothing about where the axes are placed. So it’s your task to make this blind-date working. I won’t spend much time on explaining how to make the axes, since it is quite intuitive. But a few interesting options have to be mentioned:

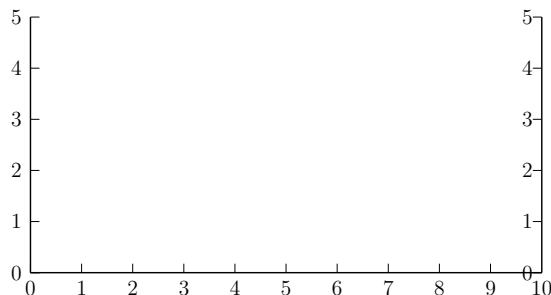
- Dx (or Dy): The interval at which the x-labels (or y-labels) are shown.
- dx (or dy): The distance between the x-labels (or y-labels). You are allowed to specify a unit here (fx. 1 cm).
- 0x (or 0y): The level at which the x-axes (or y-axes) starts.

Now, try to make your own axes and try it without any data. For example: Play a little bit around with the axis.

4.3 Where does the data come from?

The easiest way to read data, is by placing a textfile (*.txt*) in the same directory as the your *.tex* file. The data has to be tabulator separated. The first observation to read is always the first x-value, the second observation is then the first y-value, and the third observation is then the second x-value, and so on. It is also possible to specify a library for datafiles. Remember to use American “.” as decimalseparator and not the European “,”.

Figure 2: *Example: Two y-axes and one x-axis*



```
----- Start Preamble -----
\documentclass{article}
\usepackage{pst-plot,pstricks-add}
----- End Preamble -----
\psset{xunit=1 cm,yunit=1 cm}
\pspicture(0,0)(10,10)
\psaxes[Dx=1,dx=1 cm,0x=0,Dy=1,dy=1 cm,ticks=5pt](0,0)(10,10)
\psaxes[xAxis=false,Dy=1,dy=1cm,ticks=-5pt](10,0)(10.01,10)
\endpspicture
----- Code end -----
```

4.4 Plots, plots, plots

Now it's time to plot. Download the datafile data2.txt. Use the following command to read the data:

```
\readdata{name}{data-reference}
```

Use listplot to make the plots:

```
\listplot[options]{the name you specified in readdata}
```

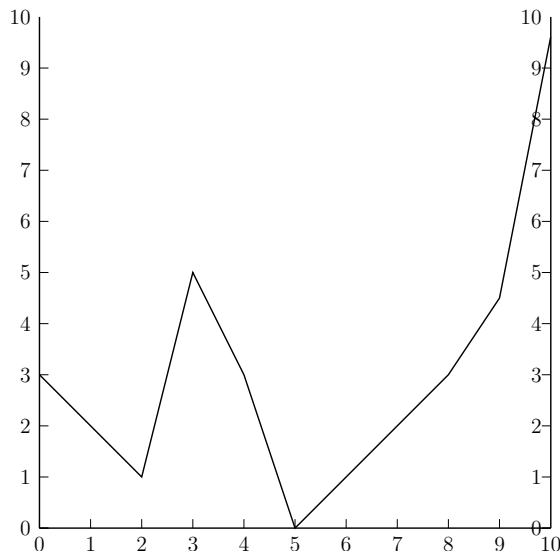
The listplot function is one out of many. There are some very useful options for listplots:

- nEnd specifies the last x-value to read. Fx. nEnd=1999.
- linecolor="colour"
- linewidth=1pt
- linestyle=dashed
- and more... (see pstricks-add-doc)

Lets use the data from data2.txt on the graphs from above:

Simple? Yes! Good, try to make your own plots before you move on to the next section.

Figure 3: *Example: Two y-axes and one x-axis and a plot*



```

----- Start Preamble -----
\documentclass{article}
\usepackage{pst-plot,pstricks-add}
----- End Preamble -----

\psset{xunit=1 cm,yunit=1 cm}
\pspicture(0,0)(10,10)
\psaxes[Dx=1,dx=1 cm,0x=0,Dy=1,dy=1 cm,ticks=5pt](0,0)(10,10)
\psaxes[xAxis=false,Dy=1,dy=1cm,ticks=-5pt](10,0)(10.01,10)
\readdata{\data}{data2.txt}
\listplot{\data}
\endpspicture
----- Code end -----

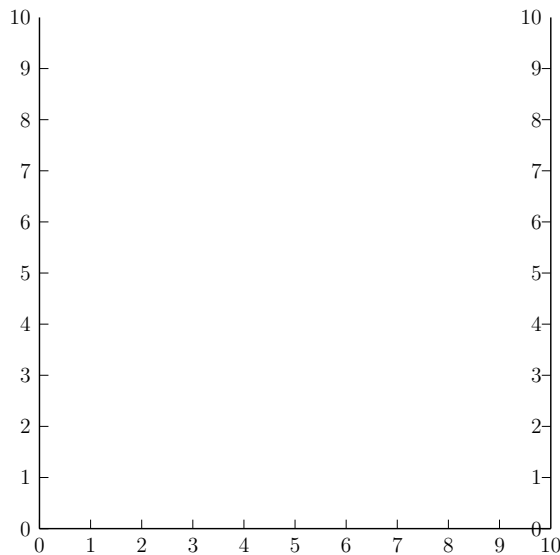
```

4.5 We have a new friend, called “pstScalePoints”

The most important feature when working with plots and graphs in \LaTeX is `\pstScalePoints`. Let’s first see why we have to use `\pstScalePoints`. Assume that you want to show the development of unemployment. from 1970 to 2006. The data is saved in the datafile `data3.txt`. We use example 3 shown above and include `data3.txt` in figure 4.

What happens? No plot is seen. Remember, that the axes do not know where plot is, you have to tell them. Take one step at time. Forget everything about the plot, and create the axes as you want them. First, we want the x-axis to start at 1970 - so we set “`0x=1970`” in the options of the axis. Furthermore, we still want the table to be 10 cm wide, but it should end

Figure 4: *Example: Scaling (I)*



```

----- Start Preamble -----
\documentclass{article}
\usepackage{pst-plot,pstricks-add}
----- End Preamble -----
\psset{xunit=1 cm,yunit=1 cm}
\pspicture(0,0)(10,10)
\psaxes[Dx=1,dx=1cm,Ox=0,Dy=1,dy=1cm,ticksize=5pt](0,0)(10,10)
\psaxes[xAxis=false,Dy=1,dy=1cm,ticksize=-5pt](10,0)(10.01,10)
\readdata{\data}{data3.txt}
\listplot{\data}
\endpspicture
----- Code end -----

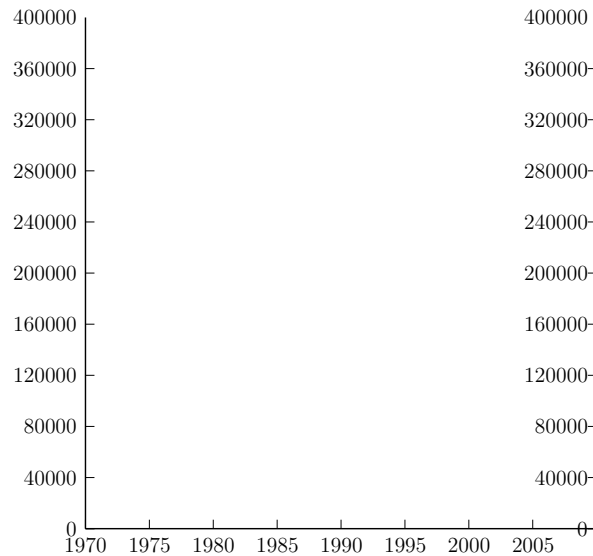
```

at 2006. Lets divide $(2006 - 1970)/10 = 3,6$ between every label, that would mean “1970, 1973,6...” that would not be beautiful. Try a little bit around. I suggest, that we use “Dx=5” and “dx=1.25cm”.

What about the y-axis? Have a look at data. We see, that the lowest level of unemployment was at 30.000 and the highest was around 370.000 individuals. We set “Dy=40000” and “dy=1 cm”. This gives the axes shown in figure 5

The axes are now as we want them. But let us now come to the important part. We need rescale the listplot. To start with, adjust the x-axis. The first value is 1970. In figure 5 that would mean, that the first value is almost 20 meters to the right of the document. We have to rescale it down. So that

Figure 5: *Example: Scaling (II)*



```

----- Start Preamble -----
\documentclass{article}
\usepackage{pst-plot,pstricks-add}
----- End Preamble -----
\psset{xunit=1cm,yunit=1cm}
\pspicture(0,0)(10,10)
\psaxes[Dx=5,dx=1.25 cm,0x=1970,Dy=40000,dy=1 cm,ticks=5pt](0,0)(10,10)
\psaxes[xAxis=false,Dy=40000,dy=1 cm,ticks=-5pt](10,0)(10.01,10)
\endpspicture
----- Code end -----

```

L^AT_EX thinks that 1970 is 0. How does is that work?

`\pstScalePoints(a,b){c add}{d add}`

- First c is added to the x -value, and d is added to the y -value.
- Then the resulting values are multiplied by a and b respectively.

We can formalize this to the two functions for the y -values and x -values:

$$\hat{x} = (x + c)a$$

$$\hat{y} = (y + d)b$$

Where $x(y)$ is the input value from your datafile and $\hat{x}(\hat{y})$ denotes where it is placed using the units set. Let us use the example above starting with

the x-axis. As mentioned above the value 1975 would be 19,75 meters to the right of the document, but we want it to be at “1.25” in the figure and likewise we want 1995 to be at the point $1.25 * 5 = 6.25$. Now, we have two equations with two unknowns:

$$\begin{aligned} 1,25 &= (1975 + c)a \\ 6,25 &= (2005 + c)a \end{aligned} \tag{1}$$

This gives the following values:

$$\begin{aligned} c &= -1970 \\ d &= 0.25 \end{aligned}$$

And now, for the y-axis. We want the input number 40.000 to be at 1 cm, and the input number 200.000 to be at 5 cm, this gives the following two conditions:

$$\begin{aligned} 1,25 &= (40.000 + d)b \\ 6,25 &= (200.000 + d)b \end{aligned} \tag{2}$$

This gives the following values:

$$\begin{aligned} d &= 0,028 \\ b &= 0,000025 \end{aligned}$$

And now, the interesting part, does it work? Have a look at figure 6. Yes, it looks as it should! Play a little bit around by yourself. Mastering `\pstScalePoint` is the hardest part of making dataplots.

It’s easy to plot more graphs in one figure, just read in more data, and make additional listplots. Remember, that `pstScalePoint` is a global option, so if you want to reset the scaling, you have to insert

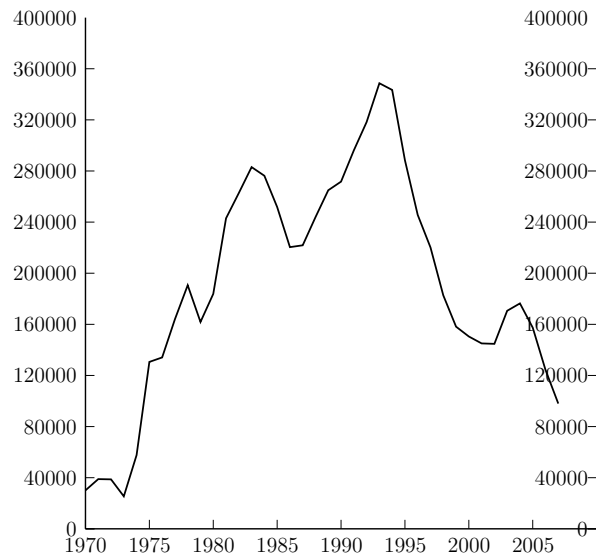
```
\pstScalePoint(1,1){}{}
```

5 To do

Yet to come:

- Labeling
- Grids
- Bars and other plottypes

Figure 6: *Example: Using pstScalePoint*



```

----- Start Preamble -----
\documentclass{article}
\usepackage{pst-plot,pstricks-add}
----- End Preamble -----
\psset{xunit=1 cm,yunit=1 cm}
\pspicture(0,0)(10,10)
\psaxes[Dx=5,dx=1.25 cm,Ox=1970,Dy=40000,dy=1 cm,ticks=5pt](0,0)(10,10)
\psaxes[xAxis=false,Dy=40000,dy=1 cm,ticks=-5pt](10,0)(10.01,10)
\pstScalePoints(0.25,0.000025){1970 sub}{0.028 add}
\readdata{\une}{data3.txt}
\listplot[linecolor=black,linewidth=1pt]{\une}
\endpspicture
----- Code end -----

```