Introduction to Scientific Typesetting
Lesson 16: Plotting data using LaTeX

Ryan Higginbottom

January 26, 2012

Line Graphs

Bar Graphs

Using Data in LaTeX Documents

You can imagine situations where data are collected and then need to be presented in a document or presentation. Of course, we'll want to use LaTeX for this document/presentation!

How should we get graphs into our LaTeX documents?

There are two main ways to proceed:

1. Create a graph in some other program (Excel, Mathematica, SPSS, etc), export the image and include it in your LaTeX document with the `\includegraphics` command.
   - **Positive**: Familiar, quick
   - **Negative**: Have little control over the appearance
2. Create the graph within LaTeX itself.

Line Graphs

The psgraph environment

The container for our line graphs will be the psgraph environment.

```
\begin{psgraph}[Options]{<arrows>}
(x0,y0)(xm,ym)(xM,yM){xL}{yL} ...
\end{psgraph}
```

x0	x-coordinate of the origin
y0	y-coordinate of the origin
xm	smallest x-coordinate
ym	smallest y-coordinate
xM	biggest x-coordinate
yM	biggest y-coordinate
xL	horizontal length of graph
yL	vertical length of graph

The Options in the psgraph environment are the same ones that apply for the psaxes environment.

Making LaTeX read your data

The key part of all of this (which is not LaTeX-dependent) is *producing the data*. For line graphs, LaTeX is looking for ordered pairs of numbers to plot — you must supply the *coordinates* of the data points for the graph. There are two ways that LaTeX can read your data once it is produced.

1. Keeping the data in a separate file, like `clinic.dat`.
   - ordered pairs can be separated by spaces, commas, parentheses, etc.
2. Writing the data in the `.tex` file itself.

Look at the first example file (`.tex`) for examples of both of these methods. Make sure that you have the file `data1.dat` in the same folder as your example `.tex` file.

Commands for the initial graph

Two important commands:

- `\readdata{macro}{file}` — this stores your data in a macro that LaTeX can use
  - `macro` should be something like `\data` or `\dataA` — no numbers allowed!
  - `file` is the name of the data file, including file extension
- `\listplot[options]{macro}` — the command to plot the data
  - `options` can include line thickness, line color, whether or not to show the data points

Plotting multiple data sets on one axis

It is easy to plot multiple sets of data on the same set of axes.

Open up the second example file (`.tex`), build and view.

When plotting multiple sets of data on one set of axes, it is standard to include a *key* or *legend* to help your reader distinguish between the two sets of data.

Uncomment the bottom part of the last example file, build and view.

Legends for graphs

The command to make a legend for your graph is this:

```
\pslegend[reference](xoffset,yoffset){text}
```

- `reference` — must be one of `lb`, `lt`, `rb`, or `rt`, where `rt` is the default
- `xoffset` and `yoffset` — units (multiples of 1 point) to move the legend away from the specified corner of the graph
- `text` — the contents of the legend, typeset in a `tabular` environment

Example of legend text:

```
\pslegend[lt]{
    \red\rule[1ex]{2em}{1pt} & Data I\\
    \blue\rule[1ex]{2em}{1pt} & Data II}
```

More on the legend

Two notes on legends:

- Any color that has *already been defined* can be used as a command (as in the previous example with `\red` and `\blue`).
- The legend must be defined *before* the `psgraph` environment.

Labels for the axis

Obtaining labels for the axes:

```
\psset{xAxisLabel=Time,yAxisLabel=Height}
```

This command needs to come *before* the `psgraph` environment.

To position the axis labels:

```
\psset{xAxisLabelPos={c,-.4in},
     yAxisLabelPos={-.4in,c}}
```

Open the third example file (`.tex`), build and view.

Let's practice!

Open the fourth example file (`.pdf`) and reproduce it.

Bar Graphs

The data

It is best for LaTeX to read `.csv` files when creating bar graphs. The first row should be the labels, the second row should be the values. The `.csv` file can either be written within LaTeX using the `filecontents*` environment (note the `*`) or written with an external program.

- The command to store the data in a macro is `\readpsbardata{macro}{file}`.
- The command to produce the bar chart is `\psbarchart[options]{macro}`

Open the fifth example file (`.tex`), build and view.

Different bar styles

The appearance of the bars is set through the `barstyle=style` option on `\psbarchart`. The available `barstyles`:

- `black`
- `gray`
- `darkgray`
- `lightgray`
- `white`
- `red`
- `green`
- `blue`

It's not too hard to define one's own `barstyle`:

```
\newrgbcolor{mypurple}{.5 0 .5}
\newpsbarstyle{spiffy}
    {fillcolor=mypurple,fillstyle=solid}
...
\psbarchart[barstyle=spiffy]{...}
```

Example

Open the sixth example file (`.tex`) posted on Sakai.

Build and view.

Bar width and separation

Changing the bar width:

`\setlength{\psxunit}{newwidth}`

- This is a command itself that must go *within* the `psgraph` environment but before the `\psbarchart` command.

Changing the space between bars:

`barcolsep=newseparation`

- This is an option in the `\psbarchart` command.
- Default is `0.4`.

Uncomment the last part of the previous example file. Build and view.

The appearance of the bar labels

The bar labels can be rotated.

`barlabelrot=angle`

- This is an option in the `\psbarchart` command.
- Default is `0`.

The separation between the bar labels and the horizontal axis can be changed.

`labelsep=newseparation`

- This is an option in the `\psbarchart` command.

Open the seventh example file (`.tex`) from Sakai. Build and view.

Let's practice!

Open the eighth example file (`.pdf`) and reproduce it.