

Worksheet 8: GnuPlot

Producing Graphs With GnuPlot

1. You will be making an artificial graph of data by plotting the numbers of most popular boys' names vs. girls' names as two lines on a graph. First, however, modify the sample GnuPlot input file (but not the data yet), so it has reasonable information for the graph you'll create. Copy the file from my directory to yours:

```
-> cp ~stacia/ws05/gp.in gp.in
```

You should change the axis labels, the graph title, and the names of the curves. Now produce a PostScript graph and use view it with GhostScript through Xwindows. (Start Xwindows from inside Utilities, ssh -X bovine, then type gs followed by the postscript file.)

2. Now write a Perl program that will extract out the relevant information from the file with boys' and girls' names in it and create an input data file. This file should have 3 columns, the rank of the name (x-coordinate), and the count of the name for girls and for boys.
3. Modify your GnuPlot input file (gp.in) so that it can create the graph from the datafile created in the previous step. Try it out:

```
-> gnuplot gp.in
```

Then view the graph.

4. Now modify the Perl program so that it creates the PostScript graph as output (in a file).
5. Create a GnuPlot template by copying gp.in to gnuplot.template, then replace references to input data and output data with pertinent words in all caps. Often people want to create the same format of graph over and over and only change labels or the data file name, which makes a standard template useful. You can then use your template by replacing the words in all caps using a Perl script. For example, you can decide all data files will end with .data and all output files will end in .ps, then you could replace 2D with NAME. Then say you were graphing bowling scores and the data was in bowl.data, you can go through your GnuPlot template replacing NAME with bowl:

```
$_ = s/NAME/bowl/g;
```

Perl Subroutines

1. Modify Worksheet 6, number 5 so that the encryption is performed in a subroutine called `&encrypt` which takes a string as an argument.
2. Write a subroutine (a.k.a. method, function) called `&total` which returns the sum of a list of numbers when passed an array of numbers.
3. Write a subroutine called `&min` which returns the minimum of a list of numbers when the numbers are passed as individual parameters (the number of parameters may vary).