Gnuplot 4.6 Frequenty Used Commands Guide

Author: Sumant Raykar
sraykar@mit.edu

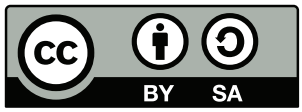
Started: March 3, 2013

Last Updated: August 26, 2013

Overview

`gnuplot` is a powerful command-line based graphing software, often used in scientific publishing. `gnuplot` generates high-quality vector images, and is highly versatile. However, as I started using it, I found that there wasn't a single comprehensive manual or resource that I could refer to when I ran intro trouble. There are several online resources [see page 12], and Stack Overflow and other forums offer some help, but no one resource was always useful every time. So, I wrote this manual as a guide for myself to keep track of my favorite `gnuplot` commands and functions. This repeatability is one of the perks of the command-line interface. I thought others could benefit from using this manual and hence I am putting it out there for any one to use, free of cost. I have tried to explain all the commands in detail, and I encourage you to verify that my examples work as suggested. I have tested the commands on Windows 7 and Debian Wheezy. When I say GNU/Linux in the manual, I'm referring to the Debian distribution.

This manual is provided as-is with no guarantees. I am not responsible if you mess-up your graphs, or for any damage you may inflict on yourself or your computer while using this manual. I do not work for or represent `gnuplot`. I do not guarantee that the methods shown here are the best ones or recommended ones. However, they work. I am a beginner-level user of `gnuplot`, and I update the document when I learn new or better methods. If you would like the source data files, images, or the LaTeX source file for this document, email me at the address above.



Plotting data

1. Put the data in a `.txt` file. The first column should be the x-axis values, and the second column will be the y-axis values.

2. Start `gnuplot` and navigate to the directory where the data text file is saved.

```
cd 'D:\Documents\....'  # for Windows
cd './Documents/...'    # for GNU/Linux
```

3. Plot the data as follows:

```
plot 'foo.txt'              # plots scatter plot
plot 'foo.txt' with lines   # plots lines
plot 'foo.txt', 'bar.txt'   # plots two data sets on the same graph
```

Axes

1. Set the axes labels as follows:

```
set xlabel 'Time'
set ylabel 'Service Rate'
```

Setting the line types

1. We first define the line styles for each line that we want to plot by using the following command:

```
set style line 1 lt 1 lc rgb "red" lw 1 pt 4
```
for line number '1', sets line type to be 1, color to red, line weight to 1, and point type 4

```
set style line 2 lt 1 lc rgb "green" lw 1 pt 1
```
for line number '2', sets line type to be 1, color to green, line weight to 1, and point type 1

lt	Output
1	Solid
2	Dashed
3	Smaller dashes
4	Smaller dashes

Table 1: Line types

pt	Output
1	+ sign
2	cross
3	3 lines cross
4	empty square
5	filled square
6	empty circle
7	filled circle
8	empty triangle
9	filled triangle
10	empty inverse triangle
11	filled inverse triangle
12	empty tilted square
13	filled titled square
14	empty pentagon
15	filled pentagon
16	...

Table 2: Line marker shapes and fill

2. The line type argument can take the values as shown in table 1.

3. The marker shapes can be decided by using the arguments shown in table 2.

4. The color names can be obtained by the command:

```
show colornames  # shows the standard dictionary of color names
```

Key

1. The key can be removed by the command:

2. The key position can be set by the command:

```
unset key
```

```
set key top left  # other arguments are bottom and right
```

3. To place the key outside the graph window, see: info gnuplot

Plotting several data sets on a graph

1. All the information above can finally be used to plot several data sets on to a single graph. Suppose you have a data file listing the rainfall (in mm) in each quarter of a year for four cities, as shown in table 3.

Quarter	Boston	NYC	Seattle	SF
Q1	10	8	30	5
Q2	35	30	45	10
Q3	65	55	80	25
Q4	15	10	35	10

Table 3: Source file `rainfall.txt` listing rainfall in four cities (in mm)

This information can be plotted using the following set of commands.

```
set xrange [-0.5:3.5]                                          # set the x-axis range
set yrange [0:90]                                              # set the y-axis range
set style line 1 lt 1 lc rgb "red" lw 1 pt 1                   # line style for line 1
set style line 2 lt 1 lc rgb "green" lw 1 pt 2                 # line style for line 2
set style line 3 lt 1 lc rgb "blue" lw 1 pt 3                  # line style for line 3
set style line 4 lt 1 lc rgb "magenta" lw 1 pt 4               # line style for line 4
set xlabel ‘Quarter’                                           # set X-axis label
set ylabel ‘Rainfall (in mm)’                                  # set Y-axis label
set key top left                                               # set the position of the key
set key autotitle columnheader                                 # column-headers in the source file are to be used as titles in the legend

plot ‘‘rainfall.txt’’ u 2:xticlabels(1) w linespoints          # plot all the data sets, taking the x-axis labels from column 1
ls 1, ’’ u 3:xticlabels(1) w linespoints ls 2, ’’ u
4:xticlabels(1) w linespoints ls 3, ’’ u 5:xticlabels(1)
w linespoints ls 4
```

Plotting several data sets on a graph

2. The output should look like figure 1.

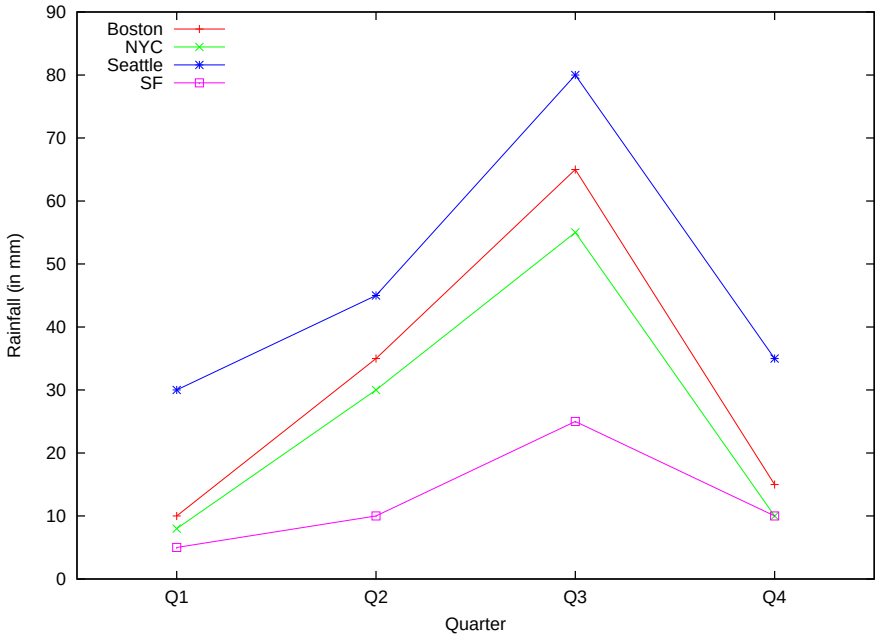


Figure 1: Plotting several lines on the same graph

Plotting several columns on the same graph can be accomplished using the `histogram` style as shown below. The `axes x1y1` and `axes x1y2` arguments are critical for matching the data and axes correctly.

The output should appear as shown in figure 2.

`set xrange [-1:24]`	# set the range for the common x-axis
`set yrange [0:700]`	# set the range for the left-side y-axis
`set y2range [0:0.2]`	# set the range for the right-side y-axis
`set ytics 100 tc lt 1 nomirror`	# set the ytics interval and line type; `nomirror` means tics are not shown on the y2 axis
`set y2tics 0.05 tc lt 3`	# set the y2tics interval and line type
`set ylabel ‘‘Number of messages’’`	# set the label for the y axis
`set y2label ‘‘Probability of posting’’`	# set the label for the y2 axis
`set style data histogram`	# set the histogram style
`set style histogram cluster gap 1`	# set the gap between the clusters of boxes
`set boxwidth 0.9`	# set the width of the histogram boxes
`set style fill solid border -1`	# set the column fill color and border
`plot ‘test.txt’ using 2:xtic(1) axes x1y1, ’’ u 3 axes x1y2 lc rgb ‘‘blue’’`	# plot the second column from the text file against the first column xtics, and from the same source file, graph the third column too but against the y2 axis

Use of the `histogram` style

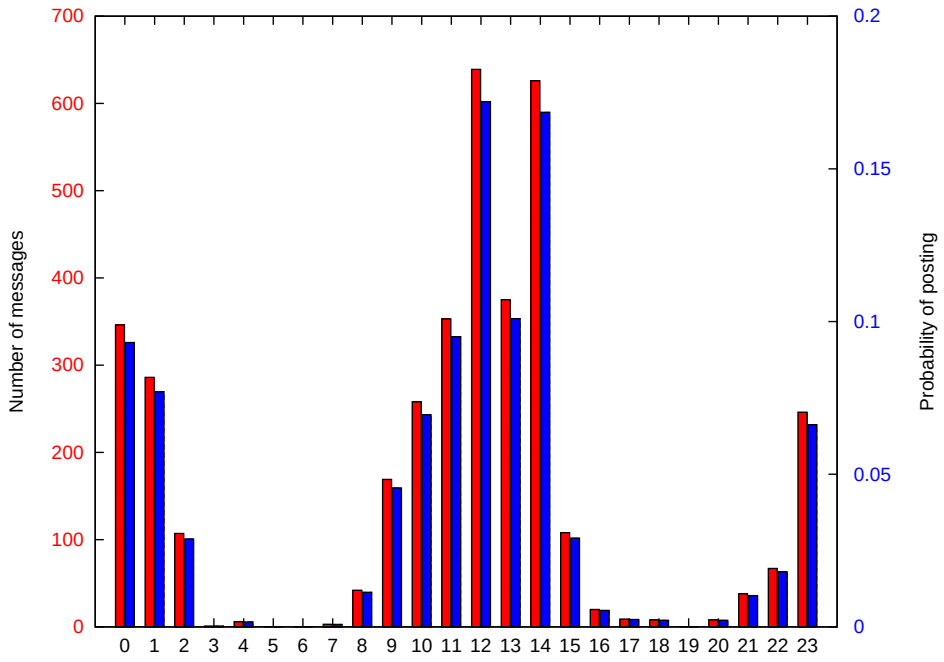


Figure 2: Example of the `histogram` style

Plotting time on the X-axis

`gnuplot` has a special format for plotting time on the X-axis. This format, called `timefmt`, reads time data from a source file and interprets it in the correct manner. Suppose the data file contains about 60 lines in the format shown in table 6.

13:30	6
13:31	4
13:32	6
13:33	3
13:34	3
13:35	4
13:36	5
13:37	5
13:38	5
13:39	3

Table 6: Snippet of the source file

This can be plotted as follows. At the `gnuplot` terminal, type

`set xdata time`	# indicate that the X-axis carries time values
`set timefmt "%H:%M"`	# indicate that the input file will be read by `timefmt` in the format indicated

Plotting time on the X-axis: set `timefmt`

Then, we can plot the data in the source file as follows:

`set xtics format ‘‘%H:%M’’`	# sets the format to display the xtics
`set xtics ‘‘13:30’’, 300, ‘‘14:30’’`	# set the start, interval in seconds and end point for the X-axis
`plot ’time-example.txt’ u 1:2 w lines`	

Plotting time on the X-axis

The graph looks like figure 3.
Different time formats are found at http://gnuplot.sourceforge.net/docs_4.2/node274.html.

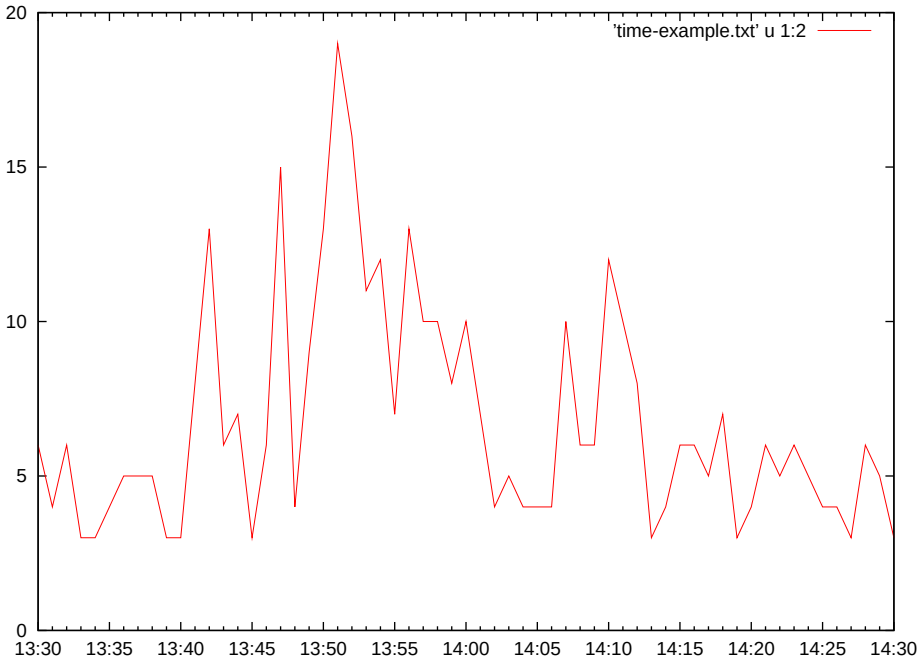


Figure 3: Using `timefmt` for the X-axis

Exporting to PDF

1. The default terminal is `wxt` for GNU/Linux and `windows` for Windows.

2. To export to an EPS file, use the following commands at the `gnuplot` terminal:

```
set terminal postscript eps enhanced color
set output 'foobar.eps'
replot
```

3. To export the EPS file to PDF in GNU/Linux, open another terminal window and move to the path where the EPS file is stored. Then run the `epstopdf` command.

```
epstopdf foobar.eps
```

4. In Windows, just use Ghostscript to export the EPS file to PDF. Then use Foxit PDF Editor to crop the PDF file to size.

5. To export to a WinFIG or XFig compatible format, use the following commands at the `gnuplot` terminal:

```
set terminal fig color
set output 'foobar.fig'
replot
```

6. The file can then be opened in WinFIG or XFig and you can add LaTeX math to it, and then export it as usual. In GNU/Linux, first export the file using 'Combined PS/Latex (both parts)'. Then follow this procedure:

```
latex launcher.tex                          # compile the launcher file (see reference 3) containing the .pslatex_t file as input
dvips -E launcher.dvi -o outputfilename.eps # converts DVI file to EPS file
epstopdf outputfilename.eps                 # convert the EPS file to PDF
```

Favorite commands

```
plot [-3:3][0:1] norm(x)                                    # plots the CDF of the normal distribution

Norm(x,m,s) = 1./(sqrt(2*pi)*s) * exp(-(x-m)**2/(2*s*s))    # define the normal distribution PDF

plot [-5:5] Norm(x,0,1)                                     # plot the PDF for mean 0 and sd 1
```

Normal Distribution

```
geometric(x,p) = (1-p)**(x-1)*p                             # define the geometric distribution PDF

xmin = 0                                                    # set the lower limit of xmin to be sampled

xmax = 8                                                    # set the upper limit of xmin to be sampled

set sample (xmax - xmin) + 1                                # sets the number of points to be sampled

ymax = 1.1*p                                                # define the max value on the y-axis

set yrange [0:ymax]                                         # set the max value on the y-axis

plot geometric(x,0.5) with impulses title "p = 0.5"         # plot the geometric distribution with impulses, and for given p-value
```

Geometric Distribution

```
expo(x,a,b) = a*exp(b*x)        # define a new function expo in terms of standard gnuplot function exp

plot [-1:5] expo(x,1,1)         # plot the function for given a and b
```

Exponential Distribution

Useful references

1. http://www.manpagez.com/info/gnuplot/gnuplot-4.4.0/gnuplot_463.php#Concept_005fIndex
2. http://www.gnuplot.info/demo/index.html
3. Guide to convert fig files containing LaTeX math to pdf files:
   http://graphics.stanford.edu/lab/howto/xfig_latex.html
4. http://gnuplot.sourceforge.net/