# Creating Sparklines with Gnuplot Under Ubuntu

Sparklines are small graphs designed to be embedded within a paragraph as though they were words. This article documents a script that will help you easily create sparklines using Gnuplot. While written under Ubuntu Maverick there's no real reason it shouldn't be able to be modified for other systems that have access to Gnuplot - possibly including Windows using Cygwin (http://www.cygwin.com).

## 1. The Problem

Sparklines are an interesting concept, while they lack the detail of a full-fledged graph they can provide a very high density of information in the space of a word. Because of their size they are inherently simplistic, lacking labels and complex annotations. However, because of Gnuplot's inherent complexity creating the simple can be difficult. Enter sparkline-gp, a script to create sparkline graphs using Gnuplot with a minimum of effort. While the script is rapidly increasing in complexity and options, running **sparkline-gp f=data.dat** is often enough to create a workable sparkline. Details of these options are documented below.

### 1.1. Other Solutions

Sparklines are just a concept, there's nothing magic about my script that makes it a sparkline and all other methods inferior. There is a Ubuntu package libsparkline-php which can also create sparklines, however it has a dependency of apache2 which makes it less than a lightweight solution. Sparklines can be created in the OpenOffice.org (or LibreOffice) spreadsheet programs by simply creating a tiny graph. Microsoft Excel has native support for sparklines (http://blogs.office.com/b/microsoft-excel/archive/2009/07/17/sparklines-in-excel.aspx). There is a jQuery (http://omnipotent.net/jquery.sparkline/) implementation with a much more comprehensive set of graphs than my own. You can also build them as a web service (http://sparklines.bitworking.info/). There's no shortage of ways to create them, but I was looking for a very simple method to create them from a command line. And being able to add the features *I* want to incorporate them into my DocBook (https://www.ebower.com/docs/docbook) documents is just icing on the cake.

# 2. Installing sparkline-gp

The easiest way to ensure you’ve got the latest and greatest is to install the eBower PPA as described in Appendix A. Afterwords run **sudo apt-get update && sudo apt-get install sparkline-gp** and the script will install itself in `/usr/bin/sparkline-gp`.

You can also manually install the script if you prefer, but the script is still in the early stages so you’ll want to check back often for updates. Because sparkline-gp is simply a front end to Gnuplot, it has very limited dependencies. If you decide to manually install the script you’ll need to ensure you have gnuplot-nox installed on your system. The script itself can be found here (sparkline-gp), I fear that embedding the script within my document is tedious so I’ll leave the click-through to the reader.

# 3. Using sparkline-gp

Using **sparkline-gp** only requires an input file as a parameter. However there are many options that can be used to tweak and add to your graph. The format of the parameters is an arbitrarily ordered list of *variable=value* pairs.

## 3.1. File Processing Parameters

The only required variable is the input file, everything else is optional. File processing parameters are fairly-self explanatory - they define the input and output files of the script.

f

This parameter represents the Gnuplot formatted input file. Additional details of this format can be found in Section 3.1.1.

o

This parameter defines the output PNG file. By default it will be of the format `inputfile.png` - simply tagging a `.png` extension to the input file. Note that the output file format will always be PNG, if you need another format you’ll need to use an external program to perform the conversion. The imagemagick package includes a utility called **convert** that can be used for this.

### 3.1.1. Input File Format

The format is just a standard Gnuplot file. Lines starting with # will be ignored as comments. Each row represents a space separated `x y` value. Additional columns will be ignored. A simple example file that outputs is below.

**Example 1. Sample Input File**

```
# X-value Y-value
0 12
1 15
2 18
3 16
4 19
5 21
6 15
7 23
8 18
9 16
10 12
```

## 3.2. Image Scaling

The Y axis may be manually scaled as needed.

min

> This variable will force the graph to a use a defined minimum value. By default this value will be automatically calculated based on the data, however it may be useful to force a lower bound to change the shape of the graph or to normalize multiple graphs displayed simultaneously.

max

> This variable will force the graph to a use a defined maximum value. By default this value will be automatically calculated based on the data, however it may be useful to force an upper bound to change the shape of the graph or to normalize multiple graphs displayed simultaneously.

## 3.3. Marks and Annotations

Sparklines can include marks and annotations to call out certain properties. Colors are represented by text (red, blue, green, orange, etc.) or by an RGB specification of the format #RRGGBB.

c

> This parameter will modify the color of the line which, by default, will be red.

marklast

> This will create a dot of the specified color to highlight the last point on the graph. By default there will be no dot.

markmin

This will create a dot of the specified color to highlight the first instance of the lowest point on the graph. By default there will be no dot.

markmax

This will create a dot of the specified color to highlight the last instance of the highest point on the graph. By default there will be no dot.

## 3.4. Creating a Range

Sometimes it's useful to highlight a range on the graph, for example to show the normal range of values and highlight deviations from this. If the range specified exceeds the automatically calculated minimum and maximum Y values the range will supersede the values. To override this you may manually specify the `min` and/or `max` values.

rangestart

This is the lower value of the range, by default there is no range defined. When defined without a `rangestop` value the script will exit with an error code.

rangestop

This is the higher value of the range, by default there is no range defined. When defined without a `rangestart` value the script will exit with an error code.

rangecolor

This allows you to change the color of the range rectangle, by default it will be a muted grey. The format of this variable is described in Section 3.3.

## 3.5. Image Size and Positioning Parameters

The default parameters are designed to work well with common font settings. In some cases (for example, with different font sizes) these parameters can be tweaked to improve the overall effect.

h

This variable is the height of the image in pixels. By default this will be 15 pixels.

w

This variable is the width of the image in pixels. By default this will be 100 pixels.

tbuff

This variable allows you to add a buffer to the top of the image. This can be useful if the top of a marker is cut off or missing. The default value is 0.

bbuff

This variable allows you to add a buffer to the bottom of the image. This can be useful if the bottom of a marker is cut off or missing. The default value is 0.

rbuff

This variable allows you to add a buffer to the right of the image. This can be useful if the right of a marker is cut off or missing. The default value is 0.

lbuff

This variable allows you to add a buffer to the right of the image. This can be useful if the left of a marker is cut off or missing. The default value is 0.

# A. Installing the eBower PPA

I've created a Personal Packages Archive (PPA) on Launchpad.net. You can add this using the command **sudo add-apt-repository ppa:ubuntu-ebower/ebower** and then **sudo apt-get update** to load the list of packages. Alternatively you can add the following lines to your `/etc/apt/sources.lst` (this is useful if you're running server and don't want to add the **add-apt-repository** command):

```
deb http://ppa.launchpad.net/ubuntu-ebower/ebower/ubuntu precise main
deb-src http://ppa.launchpad.net/ubuntu-ebower/ebower/ubuntu precise main
```

Make sure you replace *precise* above with your distro (lucid, oneiric, etc.). If your distro is not available please contact me with the version and package name and I'll try to update things posthaste. After this is done, run the command **sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys C27A63B3** to import the keys.

By installing my scripts from the PPA you'll be sure to have the most up-to-date versions in case I need to make changes. However, note that I'm not a developer and I only play one on the Internet. If you use my scripts as a baseline and make changes yourself please let me know how you're changing things and I'll try to accommodate your efforts to ensure that I don't stomp on your customizations.

# B. About Me

My name is Jeff Bower, I'm a technology professional (http://www.linkedin.com/in/jdbower) with more years of experience in the telecommunications industry than I'd care to admit. I tend to post with the username jdbower on various forums, including Komodo Kamado (http://komodokamado.com/forum/), Android Central (http://forum.androidcentral.com/), VirtualBox (http://forums.virtualbox.org/), and MakeMKV (http://www.makemkv.com/forum2/). Writing these documents is a hobby of mine, I hope you find them useful and feel free to browse more at https://www.ebower.com/docs.

I also enjoy cooking, especially outdoors with my Komodo Kamado (http://www.komodokamado.com) and using my Stoker (https://www.rocksbarbque.com). Take a look at my recipes stored at https://www.ebower.com/recipes.

If you've got any questions or feedback please feel free to email me at docs@ebower.com (mailto:docs@ebower.com) or follow me on Google+ (https://profiles.google.com/100268310848930740059) or Twitter (http://twitter.com/jdbower).