

Gnuplot 学习笔记

mr-zhangxj@hotmail.com

本人的Gnuplot的学习笔记，资料来源主要来自Google和Gnuplot自身。不少资料上声称Gnuplot要比Origin跟强大。本人不置可否。显然，如果要快速得到一个图表，Origin甚至不需要学习。鼠标点点即可搞定。相比之下Gnuplot需要记住一些命令，如果对打字比较慢得话还会大大的降低效率。如果你的目的很简单，就是为了文章中得一个数据图，仅此而已，选择Origin，It's a best choice. 如果你喜欢Linux，喜欢在命令行下做事情。学一学Gnuplot还是不错的。

PS: 本人水平一般，难免出错，多多指正

文本内容:

Gnuplot 学习笔记.....	1
获得Gnuplot和帮助.....	1
获得软件.....	1
获得帮助.....	2
函数绘图.....	3
plot, label, tics, title.....	3
font.....	3
format, margin.....	4
key.....	4
lines, points.....	5
导出图片.....	6
脚本文件.....	7
数据绘图.....	8

获得 Gnuplot 和帮助

获得软件

对于Debian和Ubuntu 使用命名

```
# apt-get install gnuplot
```

```
$ sudo apt-get install install gnuplot
```

即可获得Gnuplot，当然这样安装的一般不是最新版本，如果想要见识一下

最新版可以到GNU的官方网站

<http://www.gnuplot.info/>

也可以到GNU的官方网站

<http://www.gnu.org/>

这里可以找到所有的GNU软件。

其他的Linux我就知道了，本人一直在用Debian。

获得帮助

Gnuplot官网上有大量帮助。特别是其中的例子（Demos）这是可以快速开始使用Gnuplot的有效方法。如果想要获得详细信息，使用Gnuplot自带的帮助系统，当然你就要看英文了。

安装好Gnuplot后，在终端输入gnuplot就启动了。Gnuplot顺便给出了版本信息。如果要退出Gnuplot，输入q回车即可

```
gnuplot> q
```

上面给出的命令，只需要输入q，“gnuplot>”是gnuplot给出的提示符。在开始提到 gnuplot 自带的帮助之前，我们先画第一个图：启动Gnuplot，输入：

```
gnuplot> plot sin(x)
```

这时就会看到一个图跳出来了。如果你没有在桌面系统下，比如 Gnome。那么当然就不会右图蹦出来。启动自带帮助使用命名 help，比如，我们看看plot 命令的帮助是什么：

```
gnuplot> help plot
```

系统给出了一大堆帮助信息，当然我们最关心的 plot 函数的格式：

```
Syntax:
  plot {<ranges>}
        {<iteration>}
        {<function> | {"<datafile>" {fatafile-modifiers}} }
        ...
```

我只是列出了函数形式一部分。{}的意思是可选，| 的意思是“或者”（这是我猜的）比如 plot sin(x)，只是用到了{<function>}部分。注意到终端的最下面写着：

```
Press return for more
```

那就按enter键吧，后面的信息就显示出来的了，你还可以看到一些例子。所有的信息显示完毕后，就推出帮助了，回到了

```
gnuplot>
```

如果你已经推出了帮助。再进入plot的帮助：

```
gnuplot> help plot
```

还有东西没说完。下面按q（可能是question的简称），终端上显示出了：

```
Subtopics available for plot:
  acsplines axes bezier
  ....(太多了，我省略了)
Subtopic of plot:
```

它给出了plot相关的下一层帮助，输入axes，你就可以看到关于axes的帮助了，然后你还可以看axes的下一层帮助。如何回到上一层？我还不知道。如果想要中途推出帮助系统，按Ctrl+c

函数绘图

plot, label, tics, titile

实际上面我们已经使用了函数绘图了：

```
gnuplot> gnuplot sin(x)
```

gnuplot中还集成了很多函数，查看这些函数还是使用help

```
gnuplot> help functions
```

是functions，不要少了s。如果看其中的一个函数的具体帮助，你会发现某个函数帮助不在functions的下一层。需要使用 `help sin` 来查看sin的帮助。如果你要画x从50到100区间的sin函数，那么

```
gnuplot> plot [50:100] sin(x)
```

画两条曲线，并使用t作为自变量

```
gnuplot> plot [t=50:100] sin(t), cos(t)
```

加入标题，和横轴、纵轴标示，输入

```
gnuplot> set title "sin"  
gnuplot> set xlabel "t"  
gnuplot> set ylabel "sin and cos"
```

这是你发现，图上没有显示出来，那么需要使用命令

```
gnuplot> replot
```

这样就显示出来了。

font

下面就存在这样一个问题。在提交文章的时候，编辑可能说你的图片的xlabel的字太小了，那么你就需要变大字号。这个问题我们可以通过help来解决：

```
gnuplot> help xlabel
```

看到了xlabel的函数形式了吧：

Syntax:

```
set xlabel {"<label>"}{offset <offset>}{font "<font>{,<size>}"  
  {{textcolor | tc} {lt <line-type> | default}} {{no} enhanced}  
  {rotate by <degrees>}...
```

这些单词肯定认识吧。font就是关于字体和字号的设置。按照help给出的提示设置一下

```
gnuplot> set xlabel "t" font "Courier, 20"  
gnuplot> replot
```

纵轴的标签ylabel也是一样的。还有就是轴上的标度的字符大小，是一样的

```
gnuplot> set xtics font "Courier, 15"  
gnuplot> replot
```

关于gnuplot里的字体。我不是很明白，因为help中没找到太多的信息。我想这取决于你的桌面系统中存在什么字体。测试了一下，下面这些字体在我的系统中是可以调用的。

Courier Arial

如果你不是对字体有强迫症的话，这些足够了。

format, margin

有的时候，需要xtics显示成科学计数法。设置这个东东有点儿像C语言

```
gnuplot> set format x "0.1e"
```

也可是同时设置x轴和y轴

```
gnuplot> set format xy "0.1e"
```

呵呵。搞到现在的时候，你肯定已经发现了一个问题，那就是那个xlabel跑到了图片的外面去了，因为它太大了。这时你需要设置一下边界。一共有上、下、左、右四个边界：tmargin, bmargin, lmargin, rmargin。我们设置下边界 bmargin，看看怎么设置

```
gnuplot> help margin
```

(help bmargin 没什么有用的信息)，help margin 给出了所有的margin的设置方法：

```
Syntax:  
  set bmargin {{at screen} <margin>}  
  ...
```

我们这样设置

```
gnuplot> set bmargin at screen 0.2  
gnuplot> replot
```

基本上搞定了，这个0.2值的是0.2倍的图片高度，同样的方法可以设置tmargin, lmargin, rmargin。

key

key，这个东西中文的名字是图例吧？如果你不知道什么key。输入

```
gnuplot> set key off  
gnuplot> replot
```

或者

```
gnuplot> unset key  
gnuplot> replot
```

什么东西没了，什么东西就是key。然后输入命令

```
gnuplot> set key on  
gnuplot> replot
```

或者

```
gnuplot> set key  
gnuplot> replot
```

让它显示出来。关于key我想最关心的就是key的位置和key的字符的大小。审稿的编辑很在乎字符的大小。还是help key查看帮助：

```
gnuplot> help key
```

这样你就找到了font，就像设置xlabel一样

```
gnuplot> set key font "Courier, 15"
```

改变位置有两种做法，都在help key的下面的帮助里：

```
set key at 1, 5  
set key left
```

不止可以left哦，还有right center bottom top。

还有一个问题，就是这个自动生成的key可能不是你想要的。那么就需要禁止它自动生成key

```
gnuplot> set key noautotitle
```

重新设置的方法是在plot里加入命令，如

```
gnuplot> plot sin(x) title "zhang"
```

这样key里是zhang了。如果是两条曲线，那就逗号隔开

```
gnuplot> plot sin(x) title "zhang", cos(x) title "xiao"
```

这个title不是图片的title。图片的title已经在上面提到过。你可以在help key 中看到很多其他的设置，其实都是一个模式的设置方法 而且帮助信息里还有好多例子。自己研究一下吧。

lines, points

下面就是用线画图，还是点画图的问题。很简单

```
gnuplot> plot sin(x) with lines  
gnuplot> plot sin(x) with points
```

分别是用线图和用点画图。还有其他的模式。比如 dots，这些东西叫做 style。你可以使用help style 找到他们。我们就集中在lines和points

gnuplot中可以使用简称，这个很方便，with简称w，lines简称l，point简称p

```
gnuplot> plot sin(x) w p
```

用点画图。

还可以设置线的样式(`linetype`, 简称`lt`), 宽度(`linewidth`, 简称`lw`), 颜色: (`linecolor`, 简称`lc`)

```
gnuplot> plot sin(x) w l lt 2 lw 4
```

数字表示这使用了什么样的样式和宽度, 至于是什么, 就自己试吧
使用点画图几乎是一样的, 就是指定 `pointtype(pt)`, `pointsizes(ps)`

导出图片

导出图片需要设置两个东西, 一个是你想要导出图片的类型, Gnuplot里叫做`terminal` (简称`term`)。 还有就是图片的名称。

比如你要保存成`eps`, 写文章用`latex`时经常用吧。

```
gnuplot> set terminal postscript eps
```

可以简写为

```
gnuplot> set term post eps
```

然后设置保存文件的名称

```
gnuplot> set output "sin.eps"
```

名称中可以使用路径, 如

```
gnuplot> set output "../pic/sin.eps"
```

但是这样是不能保存图片的。必须要

```
gnuplot> replot
```

才能保存成图片。如果需要带颜色。就这样

```
set term term post eps color
```

还有其他的图片格式, 查看帮助

```
gnuplot> help terminal
```

比如`jpeg`格式

```
gnuplot> set term jpeg
gnuplot> set output "sin.jpeg"
gnuplot> replot
```

脚本文件

下面的问题就是脚本文件，在命令行里输入这么多行，从编辑的角度上来看是不方便的，可以把所有的这些命令写在一个文件中。Linux不强调后缀名，所以可以随便取名字啦

比如我们建立一个文件sin.gpt，内容如下

```
# test gnuplot
set term post eps color
set output "sin.eps"
set xlabel "x"
set ylabel "y"
set key noautotitle
plot [0:3.14] sin(x) title "zhang" w l ls 1 lw 5
```

在终端运行

```
$ gnuplot sin.gpt
```

就搞定了。（上面的东西可能有错，我可能敲错字母）

可能只这样还是觉得麻烦，要不停地输入 `gnuplot sin.gpt`，还要到图形界面下去点击图片。你也可以使用一个bash脚本来搞定：比如文件 `plot.sh`

```
#!/bin/bash
# just for gnuplot
gnuplot << EOF
set term post eps color
set output "sin.eps"
set xlabel "x"
set ylabel "y"
set key noautotitle
plot [0:3.14] sin(x) title "zhang" w l ls 1 lw 5
EOF

evince sin.eps

# end of file plot.sh
```

其中evince是Gnome的图片查看工具。不要忘了改可执行属性

```
$ chmod a+x plot.sh
```

然后：

```
$ ./plot.sh
```

这样就开始画图了，简单多了。

数据绘图

数据画图可能是最有用的，比如我们有一个数据文件：data.dat, 内容如下

```
0  0
1  1
2  4
3  9
...
```

可以这样直接画图

```
gnuplot> plot "data.dat" w l ls 1 lw 5
```

第一列就是x轴，第二列是y轴。当然也可以反着来，第二列做x轴，第一列是y轴。

```
gnuplot> plot "data.dat" using 2:1
```

如果数据有很多列，比如data2.dat文件是

```
0  0  1
1  1  1
2  4  0.6
.....
```

同样的可以用第二列和第三列画图

```
gnuplot> plot "data2.dat" using 2:3
```

还可以只画出某一个区间的图

```
gnuplot> set xrange [2:4]
gnuplot> replot
```

这样x轴就只取2到4范围内的值。

还有就是你可能需要拟合一下：这个命令是fit, 例如

```
gnuplot> zhang(x)=a*x**2 + b*x + c
gnuplot> fit zhang(x) "data.dat" via a, b, c
```

(**是乘方)

Gnuplot会生成一个fit.log文件。里面有 a, b, c 的拟合结果。马上就可画图了

```
gnuplot "data.dat", f(x)
```

看看拟合的是不是很好呢？

画三维图的命令式splot，和plot几乎是一样的。不过他不同于matlab，splot你需要提供三列数据。就像origin那样。你可以help splot。这里不详细说明了

这里只是一个小小的总结。我相信很适合初学者：更多东西就靠你不停地发现了。

、
=====

```
ps:
set size 0.5, 0.5    #设置图片大小
set origin 0,0       #设置零点的位置
unset命令可以把对象弄没了
unset border
unset xtics
unset key
可以自己弄着看
offset 可以设置对象位置的偏移，查看帮助。
```

pps: **Every thing is about HELP**