



EBook Gratis

APRENDIZAJE

Gnuplot

Free unaffiliated eBook created from
Stack Overflow contributors.

#gnuplot

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Gnuplot.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración.....	2
Windows.....	2
Linux.....	3
Arco.....	3
Debian y Ubuntu.....	3
CentOS / RedHat.....	3
Fedora.....	3
Mac OS X.....	3
Usando Homebrew.....	3
Utilizando MacPorts.....	3
Probar la instalación.....	3
Introducción básica a las reglas del lenguaje de programación.....	4
Capítulo 2: Ajustar datos con gnuplot.....	7
Introducción.....	7
Sintaxis.....	7
Parámetros.....	7
Observaciones.....	7
Breve introducción.....	7
Examples.....	8
Ajuste de datos con errores.....	8
Ejemplo de archivo "start.par".....	11
Ajuste: interpolación lineal básica de un conjunto de datos.....	11
Ejemplo con un polinomio de primer grado.....	11
Capítulo 3: Estilos de trazado 2D.....	16

Examples.....	16
Seleccionando un estilo de trazado.....	16
Selección explícita.....	16
Selección de estilo de trazado global.....	16
Capítulo 4: Trazado básico de archivos de datos.....	17
Introducción.....	17
Sintaxis.....	17
Examples.....	17
Trazar un solo archivo de datos.....	17
Trazando múltiples archivos de datos.....	20
Primer método - Concatenación de cuerdas.....	20
Segundo método - Uso de la función sprintf.....	22
Capítulo 5: Usando archivos de script.....	24
Sintaxis.....	24
Observaciones.....	24
Examples.....	24
Archivo de script simple.....	24
Crear un archivo de script.....	24
Ejecuta el script.....	25
Creditos.....	26

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [gnuplot](#)

It is an unofficial and free Gnuplot ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Gnuplot.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Gnuplot

Observaciones

Esta sección proporciona una descripción general de qué es gnuplot y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de gnuplot y vincular a los temas relacionados. Dado que la Documentación para gnuplot es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Versiones

Versión	Último nivel de <i>parche</i>	Última fecha de lanzamiento
5.0.x	5.0.5	2016-10-09
4.6.x	4.6.7	2015-04-28
4.4.x	4.4.4	2011-11-13
4.2.x	4.2.6	2007-07-01
4.0.x	4.0.0	2004-04-01

Examples

Instalación o configuración

Gnuplot es una utilidad de graficación manejada por línea de comandos portátil. Este ejemplo mostrará cómo configurar gnuplot en las distintas plataformas.

Windows

1. Descargue la última versión del instalador desde el [sitio gnuplot](#) .
2. Ejecute el archivo descargado y permita que se ejecute como administrador si se solicita
3. En la ventana de configuración, seleccione el idioma y siga las instrucciones en pantalla.
4. (opcional) Durante la instalación, puede seleccionar el gnuplot que se agregará a la RUTA que le permitirá ejecutar comandos desde cualquier lugar en la línea de comandos. Si elige no hacerlo, puede agregarlo manualmente más tarde o `cd` al directorio de instalación de gnuplot antes de ejecutar los comandos.

La ubicación de instalación predeterminada de gnuplot en Windows es `C:\Program Files (x86)\gnuplot`

NOTA: el nombre del archivo tendrá el formato: `gp<version>-win32-mingw.exe`

Linux

La instalación en Linux se puede realizar a través de los diferentes administradores de paquetes de la siguiente manera.

Arco

```
$ sudo pacman -S gnuplot
```

Debian y Ubuntu

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

CentOS / RedHat

```
$ sudo yum check-update
$ sudo yum install gnuplot
```

Fedora

```
$ sudo dnf check-update
$ sudo dnf install gnuplot
```

Mac OS X

Usando Homebrew

```
$ brew install gnuplot
```

Utilizando MacPorts

```
$ sudo port install gnuplot
```

Probar la instalación

Después de [instalar gnuplot](#) , es una buena idea ejecutar un ejemplo simple para asegurarse de que todo funciona bien.

1. Abre tu terminal
2. Escriba `gnuplot` .
3. Tu mensaje ahora debería cambiar a `gnuplot>`
4. Tipo: `plot sin(x)`

Si todo está bien, debería ver ahora un gráfico `sin(x)` generado por `gnuplot`.

Nota: si está en Windows y no ha agregado `gnuplot` a su `PATH` , tendrá que navegar a la carpeta `<gnuplot_install_path>\bin` . La ubicación predeterminada es: `C:\Program Files (x86)\gnuplot\bin`

Introducción básica a las reglas del lenguaje de programación.

De la documentación en línea oficial de [gnuplot 5.0](#) :

El lenguaje de comandos de `gnuplot` **mayúsculas y minúsculas**, es decir, comandos y nombres de funciones escritas en *minúsculas* no son los mismos que los que están escritos en *mayúsculas*. Todos los nombres de los comandos pueden abreviarse siempre que la abreviatura no sea ambigua. Cualquier número de comandos puede aparecer en una línea, separados por punto ; coma ; . (T. Williams, C. Kelley - *gnuplot 5.0, un programa de trazado interactivo*)

Algunos ejemplos de estas reglas básicas son

1. Un lenguaje sensible a las mayúsculas

Escribir comandos en *minúsculas* y definidos en *mayúsculas* generará una advertencia de `invalid command` .

```
gnuplot> set xlabel "x"
gnuplot> Set xlabel "x"
^
invalid command
```

También la variable `N` será diferente de la `n` .

2. Abreviaturas

Puede encontrar una lista casi completa de abreviaturas [aquí](#) . De todos modos, las primeras tres letras de cualquier comando en *gnuplot* funcionan siempre como abreviaturas. Algunos comandos permiten también una contracción más potente. Un pequeño ejemplo se da a continuación.

```
gnuplot> p sin(x)
gnuplot> rep
gnuplot> q
```

donde `p` significa `plot` , `rep` para `replot` `q` para `quit` .

3. Separadores

El símbolo utilizado para separar los comandos en una sola línea es ;

```
set title "My First Plot"; plot 'data'; print "all done!"
```

5. Comentarios

Los comentarios se admiten de la siguiente manera: un # puede aparecer en la mayoría de los lugares en una línea y gnuplot ignorará el resto de la línea. No tendrá este efecto dentro de las comillas, los números internos (incluidos los números complejos), las sustituciones de comandos internos, etc. En resumen, funciona en cualquier lugar que tenga sentido trabajar. (*Ibidem*)

Solo recuerde la simple regla de "*en cualquier lugar tiene sentido trabajar*".

```
gnuplot> # this is a comment, nothing will happen
gnuplot> plot sin(x) # another valid comment
gnuplot> plot sin(#x)
                ^
            invalid expression
```

4. Extendiendo los comandos.

Los comandos pueden extenderse sobre varias líneas de entrada al finalizar cada línea, pero la última con una barra invertida (\). La barra invertida debe ser el último carácter en cada línea. El efecto es como si la barra invertida y la nueva línea no estuvieran allí. Es decir, no se implica ningún espacio en blanco, ni se termina un comentario. Por lo tanto, al comentar una línea continua, se comenta todo el comando. (*Ibidem*)

Por ejemplo, para dividir el comando de `plot` en varias líneas,

```
plot\
  sin(x),\
  cos(x)
```

trazará lo mismo que

```
plot sin(x), cos(x)
```

Una pequeña nota sobre "*comentar una línea continua comenta todo el comando*". Si escribes el comando

```
plot\
  sin(x),\ # I would like to comment here
  cos(x)
```

se producirá un error:

```
gnuplot> plot\
>   sin(x),\ # I would like to comment here
                ^
            invalid character \
```


Así que es mejor tener cuidado y respetar la regla *"en cualquier lugar que tenga sentido trabajar"* al usar # comentarios.

Lea *Empezando con Gnuplot en línea*: <https://riptutorial.com/es/gnuplot/topic/3284/empezando-con-gnuplot>

Capítulo 2: Ajustar datos con gnuplot

Introducción

El comando de ajuste puede ajustar una función definida por el usuario a un conjunto de puntos de datos (x, y) o (x, y, z) , utilizando una implementación del algoritmo de Marquardt-Levenberg de mínimos cuadrados no lineales (**NLLS**).

Cualquier variable definida por el usuario que aparezca en el cuerpo de la función puede servir como un parámetro de ajuste, pero el tipo de retorno de la función debe ser real.

Sintaxis

- **encajar** *[xrange] [yrange] función "archivo de datos" utilizando modificador a través de parameter_file*

Parámetros

Parámetros	Detalle
Ajustar los parámetros a , b , c cualquier letra que no haya sido utilizada anteriormente	Use letras para representar parámetros que se usarán para ajustar una función. Por ejemplo: $f(x) = a * \exp(b * x) + c$, $g(x, y) = a*x**2 + b*y**2 + c*x*y$
Parámetros de archivo <code>start.par</code>	En lugar de usar parámetros sin inicializar (Marquardt-Levenberg se iniciará automáticamente para usted $a=b=c=\dots=1$) puede ponerlos en un archivo <code>start.par</code> y llamarlos en la sección de archivo de <i>parámetros</i> . Por ejemplo: <code>fit f(x) 'data.dat' u 1:2 via 'start.par'</code> . A continuación se muestra un ejemplo del archivo <code>start.par</code>

Observaciones

Breve introducción

`fit` se usa para encontrar un conjunto de parámetros que "mejor" ajusta sus datos a su función definida por el usuario. El ajuste se juzga sobre la base de la suma de las diferencias al cuadrado o 'residuos' (SSR) entre los puntos de datos de entrada y los valores de la función, evaluados en los mismos lugares. Esta cantidad a menudo se llama 'chisquare' (es decir, la letra griega chi, con el poder de 2). El algoritmo intenta minimizar el SSR, o más precisamente, el WSSR, ya que los residuos son

"ponderados" por los errores de los datos de entrada (o 1.0) antes de cuadrarlos. (*Ibidem*)

El archivo `fit.log`

Después de cada paso de la iteración, se proporciona información detallada sobre el estado del ajuste tanto en la pantalla como en el llamado archivo de registro `fit.log`. Este archivo nunca se borrará sino que siempre se agregará para que el historial de ajuste no se pierda.

Examples

Ajuste de datos con errores.

Puede haber hasta 12 variables independientes, siempre hay 1 variable dependiente y se puede ajustar cualquier número de parámetros. Opcionalmente, se pueden ingresar estimaciones de error para ponderar los puntos de datos. (T. Williams, C. Kelley - [gnuplot 5.0, un programa de trazado interactivo](#))

Si tiene un conjunto de datos y desea ajustar si el comando es muy simple y natural:

```
fit f(x) "data_set.dat" using 1:2 via par1, par2, par3
```

donde en cambio $f(x)$ podría ser también $f(x, y)$. En el caso de que también tengas estimaciones de error de datos, simplemente agrega `{y | xy | z}errors ({ | }` representan las opciones posibles) en la opción de *modificador* (ver **Sintaxis**). Por ejemplo

```
fit f(x) "data_set.dat" using 1:2:3 yerrors via par1, par2, par3
```

donde el `{y | xy | z}errors` opción de `{y | xy | z}errors` requiere, respectivamente, 1 (`y`), 2 (`xy`), 1 (`z`) columna que especifique el valor de la estimación de error.

xyerrors exponencial con **xyerrors** de un archivo.

Las estimaciones de error de datos se utilizan para calcular el peso relativo de cada punto de datos al determinar la suma ponderada de los residuos al cuadrado, WSSR o chisquare. Pueden afectar las estimaciones de los parámetros, ya que determinan cuánta influencia tiene la desviación de cada punto de datos de la función ajustada en los valores finales. Parte de la información de salida de ajuste, incluidas las estimaciones de error de parámetros, es más significativa si se proporcionan estimaciones de error de datos precisas .. (*Ibidem*)

Tomaremos un conjunto de datos de muestra `measured.dat`, compuesto por 4 columnas: las coordenadas del eje x (`Temperature (K)`), las coordenadas del eje y (`Pressure (kPa)`), las estimaciones del error x (`T_err (K)`) y las estimaciones de error y (`P_err (kPa)`).

```
#### 'measured.dat' ####
### Dependence of boiling water from Temperature and Pressure
##Temperature (K) - Pressure (kPa) - T_err (K) - P_err (kPa)
```

```

368.5    73.332    0.66    1.5
364.2    62.668    0.66    1.0
359.2    52.004    0.66    0.8
354.5    44.006    0.66    0.7
348.7    34.675    0.66    1.2
343.7    28.010    0.66    1.6
338.7    22.678    0.66    1.2
334.2    17.346    0.66    1.5
329.0    14.680    0.66    1.6
324.0    10.681    0.66    1.2
319.1     8.015    0.66    0.8
314.6     6.682    0.66    1.0
308.7     5.349    0.66    1.5

```

Ahora, solo componga el prototipo de la función que de la teoría debería aproximar nuestros datos. En este caso:

```

Z = 0.001
f(x) = W * exp(x * Z)

```

donde hemos inicializado el parámetro z porque, de lo contrario, evaluar la función exponencial $\exp(x * z)$ genera valores enormes, lo que lleva a (punto flotante) Infinito y NaN en el algoritmo de ajuste de Marquardt-Levenberg, por lo general no es necesario inicializar variables - eche un vistazo [aquí](#) , si desea saber más acerca de Marquardt-Levenberg.

¡Es hora de ajustar los datos!

```

fit f(x) "measured.dat" u 1:2:3:4 xyerrors via W, Z

```

El resultado se verá como

```

After 360 iterations the fit converged.
final sum of squares of residuals : 10.4163
rel. change during last iteration : -5.83931e-07

degrees of freedom      (FIT_NDF)                : 11
rms of residuals        (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.973105
variance of residuals (reduced chisquare) = WSSR/ndf : 0.946933
p-value of the Chisq distribution (FIT_P)           : 0.493377

Final set of parameters          Asymptotic Standard Error
=====                          =====
W              = 1.13381e-05      +/- 4.249e-06      (37.47%)
Z              = 0.0426853       +/- 0.001047     (2.453%)

correlation matrix of the fit parameters:
      W      Z
W      1.000
Z     -0.999  1.000

```

Donde ahora w y z se llenan con los parámetros deseados y las estimaciones de errores en esos.

El siguiente código produce el siguiente gráfico.

```
set term pos col
set out 'PvsT.ps'

set grid
set key center
set xlabel 'T (K)'
set ylabel 'P (kPa)'

Z = 0.001
f(x) = W * exp(x * Z)
fit f(x) "measured.dat" u 1:2:3:4 xyerrors via W, Z

p [305:] 'measured.dat' u 1:2:3:4 ps 1.3 pt 2 t 'Data' w xyerrorbars,\
f(x) t 'Fit'
```

Parcela con ajuste de `measured.dat` Usando el comando `with xyerrorbars` mostrará errores estima en la X y en la y. `set grid` colocará una grilla discontinua en los tics principales.

80

70

, que incluye los metros cuadrados de una casa en una ciudad determinada y su precio en \$ 1000.

```
### 'house_price.dat'
## X-Axis: House price (in $1000) - Y-Axis: Square meters (m^2)

245    426.72
312    601.68
279    518.16
308    571.50
199    335.28
219    472.44
405    716.28
324    546.76
319    534.34
255    518.16
```

Ajustemos esos parámetros con *gnuplot*. El comando en sí es muy simple, como se puede observar en la sintaxis, simplemente defina su prototipo de adaptación y luego use el comando de `fit` para obtener el resultado:

```
## m, q will be our fitting parameters
f(x) = m * x + q
fit f(x) 'data_set.dat' using 1:2 via m, q
```

Pero también podría ser interesante usar los parámetros obtenidos en la propia gráfica. El código siguiente se ajustará al archivo `house_price.dat` y luego `house_price.dat` los parámetros `m` y `q` para obtener la mejor aproximación de la curva del conjunto de datos. Una vez que tenga los parámetros, puede calcular el `y-value`, en este caso el *precio de la casa*, a partir de cualquier `x-value` (*metros cuadrados de la casa*) simplemente sustituyendo en la fórmula

```
y = m * x + q
```

el `x-value` apropiado. Comentemos el código.

0. Establecer el término

```
set term pos col
set out 'house_price_fit.ps'
```

1. Administración ordinaria para embellecer la gráfica.

```
set title 'Linear Regression Example Scatterplot'
set ylabel 'House price (k$ = $1000)'
set xlabel 'Square meters (m^2)'
set style line 1 ps 1.5 pt 7 lc 'red'
set style line 2 lw 1.5 lc 'blue'

set grid
set key bottom center box height 1.4

set xrange [0:450]
set yrange [0:]
```

2. El ajuste adecuado

Para esto, solo tendremos que escribir los comandos:

```
f(x) = m * x + q  
fit f(x) 'house_price.dat' via m, q
```

3. Guardar valores m y q en una cadena y graficar

Aquí usamos la función `sprintf` para preparar la etiqueta (encuadrada en el `object rectangle` del `object rectangle`) en la que vamos a imprimir el resultado del ajuste. Finalmente trazamos la gráfica completa.

```
mq_value = sprintf("Parameters values\nm = %f k$/m^2\nq = %f k$", m, q)  
set object 1 rect from 90,725 to 200, 650 fc rgb "white"  
set label 1 at 100,700 mq_value  
  
p 'house_price.dat' ls 1 t 'House price', f(x) ls 2 t 'Linear regression'  
set out
```

La salida se verá así.

800

700

600

500

(\$1000)

<https://riptutorial.com/es/gnuplot/topic/8825/ajustar-datos-con-gnuplot>

Capítulo 3: Estilos de trazado 2D

Examples

Seleccionando un estilo de trazado

Selección explícita

Un estilo de trazado suele seleccionarse con la palabra clave `with`, como

```
plot x with points
```

Esto permite utilizar diferentes estilos de trazado para cada `plot`:

```
plot x with points, 2*x with lines
```

Al escribir `help with` en la ventana de comandos de gnuplot, se obtiene una lista de todos los estilos de trazado disponibles.

Selección de estilo de trazado global

Los estilos de trazado también se pueden configurar globalmente para todos los comandos de trazado. Aquí, gnuplot distingue entre la función y los gráficos de datos, para los cuales se pueden establecer diferentes estilos predeterminados.

Para funciones use la función `set style function`:

```
set style function linespoints
plot x, 2*x
```

Para archivos de datos use `set style data`:

```
set style data lines
plot 'file.dat', 'other-file.dat'
```

Tenga en cuenta que para las funciones, el estilo predeterminado son las `lines` y para los archivos de datos son los `points`. Con los `show style data` y la `show style function`, puede inspeccionar los estilos de trazado seleccionados actualmente.

Lea [Estilos de trazado 2D en línea](https://riptutorial.com/es/gnuplot/topic/4302/estilos-de-trazado-2d): <https://riptutorial.com/es/gnuplot/topic/4302/estilos-de-trazado-2d>

Capítulo 4: Trazado básico de archivos de datos

Introducción

Una de las principales características útiles de *gnuplot* es la posibilidad de trazar **archivos de datos**. Trazar un archivo de datos es realmente sencillo con *gnuplot*; en realidad, una vez que haya abierto el software desde el terminal, solo tiene que digitar el `plot 'file'` comando para obtener un gráfico automático.

En primer lugar, antes de trazar, debe asegurarse de estar en el mismo directorio donde se encuentra el archivo de datos, de lo contrario, eventualmente recibirá una `warning`.

Sintaxis

- trazar *archivo de datos* usando *column_expression* con *estilo*

Examples

Trazar un solo archivo de datos

El comando predeterminado *gnuplot* `plot` (también solo `p`) traza un conjunto de datos con columnas, de la forma del archivo `data_set.dat` continuación.

```
# Prototype of a gnuplot data set
# data_set.dat
# X - X^2 - 2*X - Random
0      0      0      5
1      1      2      15
1.4142 2      2.8284 1
2      4      4      30
3      9      6      26.46
3.1415 9.8696 6.2832 39.11
4      16     8      20
4.5627 20.8182 9.1254 17
5.0    25.0   10.0   25.50
6      36     12     0.908
```

Como puede ver, puede escribir en su conjunto de datos en notación de punto flotante. Ahora todo está listo para hacer el gráfico de datos: solo escribiendo

```
plot "data_set.dat"
```

gnuplot producirá un gráfico en su destino de `output`. La configuración predeterminada utilizará las dos primeras columnas de su archivo de datos, respectivamente, x e y. Para especificar las columnas a trazar use el especificador de `using`

```
plot "data_set.dat" using 2:4
```

lo que significa "trazar el archivo usando la columna 2 como X y la columna 4 como Y". En el caso de que su conjunto de datos sea un archivo tridimensional, solo use `splot` ad y agregue la columna z

```
splot "data_set.dat" using 1:2:3
```

También hay diferentes estilos (consulte la documentación de gnuplot o [Selección de un estilo de trazado](#) para obtener más información) para trazar puntos. Como se dijo antes, el estilo por defecto es `point`

```
plot "data_set.dat" using 1:4 with point
```

que trazará el mismo como si no escribieras `with point`. Un estilo útil para el trazado de datos es el punto de `linespoint` que es, obviamente, "líneas + puntos". **P.EJ:**

```
plot "data_set.dat" using 1:4 with linespoint
# the abbreviated form is completely equivalent:
# p "data_set.dat" u 1:4 w lp
```

40

35

, con unos pasos decididos (si no se especifica = 1). Por ejemplo, `for [i = 0:6:2]` se incrementará `i` de 0 a 6 en 2 pasos: `i = 0, 2, 4, 6`. Todos los valores (inicio, parada e incremento) se convierten en valores enteros.

* Grid

La cuadrícula suele ser útil al trazar un conjunto de datos. Para agregar un tipo de rejilla

```
set grid
```

Trazando múltiples archivos de datos

Primer método - Concatenación de cuerdas.

El método más sencillo para trazar múltiples archivos de datos es insertar un bucle `for` dentro del comando `plot` de `gnuplot`. Suponiendo que tiene `N` archivos nombrados de forma secuencial, es decir ,

```
file_1.dat  
file_2.dat  
file_3.dat  
...  
file_N.dat
```

Ejecutando el comando

```
plot for[i = 1:N] "file_".i.".dat"
```

`file_1.dat` todos los archivos entre `file_1.dat` y `file_N.dat` en el mismo gráfico.

Ejemplo con tres archivos de datos

Tabla de conjuntos de datos

Ejes X	Y-Axe file_1.dat	Y-Axe file_2.dat	Y-Axe file_3.dat
1	1	1	1
2	2	4	2
3	3	9	6
4	4	dieciséis	24
5	5	25	120

Comandos

```
set terminal postscript color noenhanced ##setting the term
set output "multiple_files.ps"

set key center ##legend placement

plot [1:5][1:120] \
  for [i = 1:3] "file_".i.".dat" \
  pointsize 1.3 linecolor i+4 \
  title "file\_".i.".dat" \
  with linespoint
```

El bucle comienza con `for [i = 1:3] "file_".i.".dat"` y ejecuta el comando de `plot` hasta que llegue a `i = 3`. El `.i.` Es el número concatenado.

`title "file_".i.".dat"` se ha escrito con el `\` para hacer que el símbolo `_` en el nombre de los archivos aparezca como un *guión bajo* en lugar de un *subíndice*, y el especificador `noenhanced` es fundamental para obtener este resultado.

El resultado final se muestra a continuación.

120

100

`sprintf` lenguaje C. La sintaxis correcta, de la [documentación de gnuplot 5.1](#) es

```
sprintf("format", x, y, ...)
```

Un breve ejemplo aclarará cada duda.

```
file_name(n) = sprintf("file_%d.dat", n)
plot for[i = 1:N] file_name(i) title file_name(i)
```

Lea [Trazado básico de archivos de datos en línea](#):

<https://riptutorial.com/es/gnuplot/topic/3591/trazado-basico-de-archivos-de-datos>

Capítulo 5: Usando archivos de script

Sintaxis

1. `gnuplot -c scriptfile ARG1 ARG2 ...`

Observaciones

El uso básico se puede mostrar escribiendo `gnuplot -h`

```
$ gnuplot -h
Usage: gnuplot [OPTION] ... [FILE]
  -V, --version
  -h, --help
  -p  --persist
  -d  --default-settings
  -c  scriptfile ARG1 ARG2 ...
  -e  "command1; command2; ..."
gnuplot 5.0 patchlevel 3
```

Examples

Archivo de script simple

Gnuplot puede generar un gráfico a partir de un archivo de script que permite que una secuencia de comandos necesaria para dibujar un gráfico se ejecute en secuencia en lugar de escribir manualmente.

A los efectos de este ejemplo, crearemos un script simple para dibujar un `sin(x)` .

Crear un archivo de script

Crema un archivo `sinx.p` con los siguientes contenidos:

```
# Set the output to a png file
set terminal png size 500,500
# The file we'll write to
set output 'sinx.png'
# The graphic title
set title 'Sin(x)'
#plot the graphic
plot sin(x)
```

En el ejemplo anterior, se encuentran los comandos más comunes, sin embargo, hay otros comandos para explorar, como `set xlabel` , `set ylabel` , etc.

Puede personalizar la línea de `set output` con la ruta que desea que el archivo genere.

Ejecuta el script

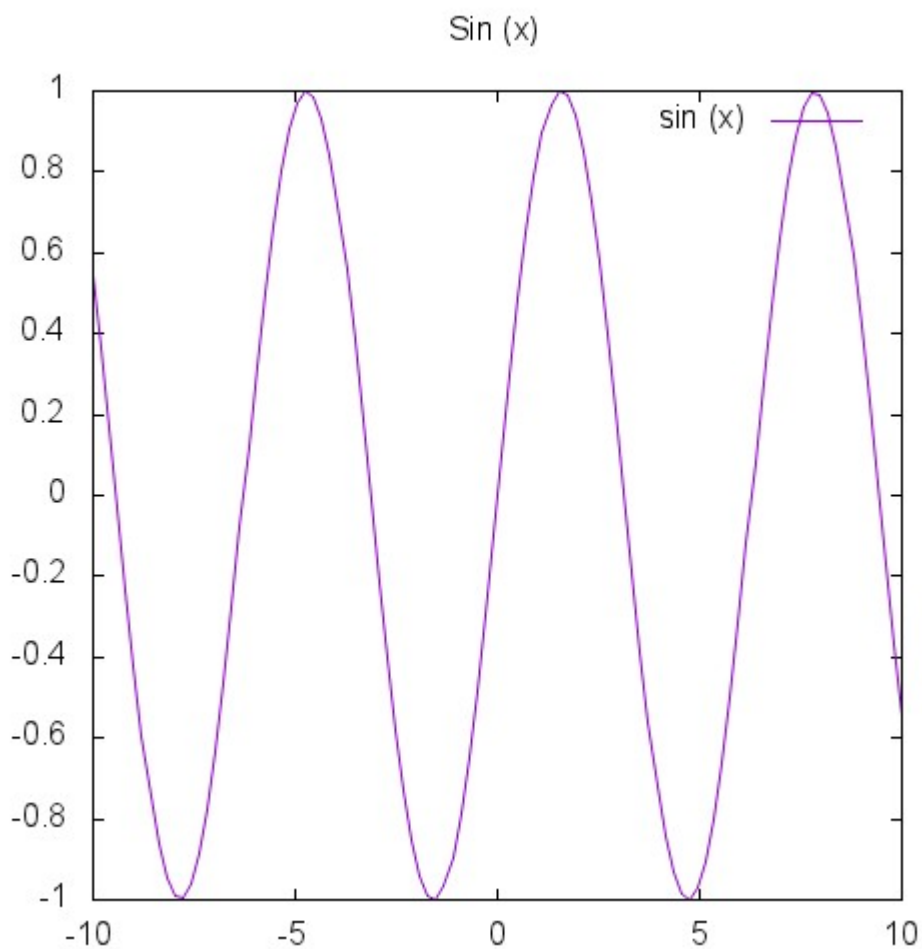
Abre tu terminal y escribe:

```
gnuplot path/to/sinx.p
```

En caso de que su carpeta actual contenga el script, puede ingresar lo siguiente:

```
gnuplot sinx.p
```

La secuencia de comandos se ejecutará y generará el archivo PNG en la ubicación especificada. El gráfico resultante debe ser similar al siguiente:



Lea Usando archivos de script en línea: <https://riptutorial.com/es/gnuplot/topic/4013/usando-archivos-de-script>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Gnuplot	Community , Fawix , opisthofulax
2	Ajustar datos con gnuplot	opisthofulax
3	Estilos de trazado 2D	Christoph
4	Trazado básico de archivos de datos	Christoph , Matthew , opisthofulax , Tom Solid
5	Usando archivos de script	Christoph , Fawix