



EBook Gratuito

APPENDIMENTO

Gnuplot

Free unaffiliated eBook created from
Stack Overflow contributors.

#gnuplot

Sommario

Di.....	1
Capitolo 1: Iniziare con Gnuplot.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione o configurazione.....	2
finestre.....	2
Linux.....	3
Arco.....	3
Debian e Ubuntu.....	3
CentOS / RedHat.....	3
Fedora.....	3
Mac OSX.....	3
Utilizzando Homebrew.....	3
Utilizzando MacPorts.....	3
Testare l'installazione.....	3
Introduzione di base alla programmazione delle regole del linguaggio.....	4
Capitolo 2: Adatta i dati con gnuplot.....	7
introduzione.....	7
Sintassi.....	7
Parametri.....	7
Osservazioni.....	7
Breve introduzione.....	7
Examples.....	8
Adattamento dei dati con errori.....	8
Esempio di file "start.par".....	11
Fit: interpolazione lineare di base di un set di dati.....	11
Esempio con un polinomio di primo grado.....	11
Capitolo 3: Stili di plottaggio 2D.....	16

Examples.....	16
Selezione di uno stile di tracciamento.....	16
Selezione esplicita.....	16
Selezione dello stile di tracciamento globale.....	16
Capitolo 4: Tracciamento di base di file di dati.....	17
introduzione.....	17
Sintassi.....	17
Examples.....	17
Traccia un singolo file di dati.....	17
Tracciare più file di dati.....	22
Primo metodo - Concatenazione di stringhe.....	22
Secondo metodo: utilizzo della funzione sprintf.....	24
Capitolo 5: Utilizzando i file di script.....	26
Sintassi.....	26
Osservazioni.....	26
Examples.....	26
Semplice file di script.....	26
Crea un file di script.....	26
Esegui lo script.....	27
Titoli di coda.....	28

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

Capitolo 1: Iniziare con Gnuplot

Osservazioni

Questa sezione fornisce una panoramica su cosa sia gnuplot e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno di gnuplot e collegarsi agli argomenti correlati. Poiché la Documentazione per gnuplot è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Versioni

Versione	Ultimo <i>patchlevel</i>	Data dell'ultimo rilascio
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

Installazione o configurazione

Gnuplot è un programma di utilità per la grafica guidato da linea di comando portatile. Questo esempio mostrerà come configurare gnuplot nelle varie piattaforme.

finestre

1. Scarica l'ultima versione del programma di installazione dal [sito gnuplot](#) .
2. Esegui il file scaricato e consenti che venga eseguito come amministratore, se richiesto
3. Nella finestra di installazione selezionare la lingua e seguire le istruzioni sullo schermo.
4. (facoltativo) Durante l'installazione è possibile selezionare il gnuplot da aggiungere al PERCORSO che consentirà di eseguire comandi da qualsiasi punto della riga di comando. Se si sceglie di non farlo, è possibile aggiungerlo manualmente in un secondo momento o effettuare un `cd` nella directory installata di gnuplot prima di eseguire i comandi.

Il percorso di installazione predefinito di gnuplot su Windows è `C:\Program Files (x86)\gnuplot`

NOTA: il nome del file sarà del formato: `gp<version>-win32-mingw.exe`

Linux

L'installazione su Linux può essere eseguita attraverso i diversi gestori di pacchetti come segue.

Arco

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

Debian e 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 OSX

Utilizzando Homebrew

```
$ brew install gnuplot
```

Utilizzando MacPorts

```
$ sudo port install gnuplot
```

Testare l'installazione

Dopo aver [installato gnuplot](#) è una buona idea eseguire un semplice esempio per assicurarsi che tutto funzioni correttamente.

1. Apri il tuo terminale

2. Digita `gnuplot .`
3. Il tuo prompt dovrebbe ora passare a `gnuplot>`
4. Tipo: `plot sin(x)`

Se tutto va bene dovresti vedere ora un $\sin(x)$ grafico generato da `gnuplot`.

Nota: se si è su Windows e non è stato aggiunto `gnuplot` al `PATH` è necessario accedere alla `<gnuplot_install_path>\bin` . Il percorso predefinito è: `C:\Program Files (x86)\gnuplot\bin`

Introduzione di base alla programmazione delle regole del linguaggio

Dalla documentazione online ufficiale di [gnuplot 5.0](#) :

Il linguaggio di comando di `gnuplot` fa **distinzione tra maiuscole e minuscole** , cioè i comandi e i nomi di funzioni scritti in *lettere minuscole* non sono gli stessi di quelli scritti in *lettere maiuscole* . Tutti i nomi dei comandi possono essere abbreviati a condizione che l'abbreviazione non sia ambigua. Qualsiasi numero di comandi può apparire su una linea, separati da punti ; virgola ; . (T. Williams, C. Kelley - *gnuplot 5.0, Un programma di plottaggio interattivo*)

Alcuni esempi di queste regole di base sono

1. Linguaggio case sensitive

Digitando comandi -defined *minuscole* in *maiuscolo* genererà un `invalid command` avvertimento.

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

Anche la variabile `N` sarà diversa da quella di `n` .

2. Abbreviazioni

È possibile trovare un elenco quasi completo delle abbreviazioni [qui](#) . In ogni caso le prime tre lettere di qualsiasi comando in *gnuplot* funzionano sempre come abbreviazioni. Alcuni comandi consentono anche una contrazione più potente. Di seguito è riportato un piccolo esempio.

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

dove `p` sta per `plot` , `rep` per `replot` e `q` per `quit` .

3. Separatori

Il simbolo utilizzato per separare i comandi su una linea singola è ;

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

5. Commenti

I commenti sono supportati come segue: un # può apparire nella maggior parte dei posti in una riga e gnuplot ignorerà il resto della linea. Non avrà questo effetto tra virgolette, numeri interni (compresi numeri complessi), sostituzioni di comandi interni, ecc. In breve, funziona ovunque abbia senso funzionare. (*Ibidem*)

Basta ricordare la semplice regola "*dovunque abbia senso lavorare*".

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

4. Estendere i comandi

I comandi possono estendersi su più righe di input terminando ciascuna riga ma l'ultima con una barra rovesciata (\). Il backslash deve essere l'ultimo carattere su ogni riga. L'effetto è come se il backslash e il newline non fossero lì. Cioè, nessuno spazio bianco è implicito, né un commento è terminato. Pertanto, commentando un commento di linea continua l'intero comando. (*Ibidem*)

Ad esempio, per dividere il comando di `plot` su più linee,

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

tratterà lo stesso di

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

Una piccola nota su "*commentare un commento di linea continua sull'intero comando*". Se si digita il comando

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

si verificherà un errore:

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

Quindi è meglio stare attenti e rispettare la regola "*ovunque abbia senso lavorare*" mentre si usano # commenti.

Leggi Iniziare con Gnuplot online: <https://riptutorial.com/it/gnuplot/topic/3284/iniziare-con-gnuplot>

Capitolo 2: Adatta i dati con gnuplot

introduzione

Il comando di adattamento può adattare una funzione definita dall'utente a un insieme di punti dati (x, y) o (x, y, z) , utilizzando un'implementazione dell' algoritmo Marquardt-Levenberg dei minimi quadrati non lineari (**NLLS**).

Qualsiasi variabile definita dall'utente che si verifica nel corpo della funzione può servire come parametro di adattamento, ma il tipo restituito della funzione deve essere reale.

Sintassi

- `fit [xrange] [yrange] function " datafile " usando il modificatore tramite parameter_file`

Parametri

parametri	Dettaglio
Adattamento dei parametri <code>a</code> , <code>b</code> , <code>c</code> e di qualsiasi lettera che non era stata precedentemente utilizzata	Utilizzare le lettere per rappresentare i parametri che verranno utilizzati per adattare una funzione. Es: $f(x) = a * \exp(b * x) + c$, $g(x, y) = a*x**2 + b*y**2 + c*x*y$
Parametri file <code>start.par</code>	Invece usando i parametri non inizializzati (il Marquardt-Levenberg si inizierà automaticamente per te <code>a=b=c=...=1</code>) puoi metterli in un file <code>start.par</code> e chiamarli con nella sezione <i>parameter_file</i> . Ad esempio: <code>fit f(x) 'data.dat' u 1:2 via 'start.par'</code> . Di seguito è riportato un esempio per il file <code>start.par</code>

Osservazioni

Breve introduzione

`fit` è usato per trovare una serie di parametri che "meglio" si adatta ai tuoi dati alla tua funzione definita dall'utente. L'adattamento viene giudicato sulla base della somma delle differenze al quadrato o dei "residui" (SSR) tra i punti dei dati di input e i valori delle funzioni, valutati nelle stesse posizioni. Questa quantità viene spesso chiamata "chisquare" (cioè la lettera greca chi, alla potenza di 2). L'algoritmo tenta di ridurre al minimo SSR, o più precisamente, WSSR, poiché i residui vengono "pesati" dagli errori dei dati di input (o 1.0) prima di essere quadrati. (*Ibidem*)

Il file `fit.log`

Dopo ogni passaggio di iterazione vengono fornite informazioni dettagliate sullo stato `fit.log` sia sullo schermo che su un file di registro cosiddetto `fit.log`. Questo file non verrà mai cancellato ma verrà sempre aggiunto in modo che la cronologia del fit non vada persa.

Examples

Adattamento dei dati con errori

Possono esserci fino a 12 variabili indipendenti, c'è sempre 1 variabile dipendente e qualsiasi numero di parametri può essere montato. Facoltativamente, è possibile immettere stime di errore per la ponderazione dei punti dati. (T. Williams, C. Kelley - [gnuplot 5.0, Un programma di plottaggio interattivo](#))

Se si dispone di un set di dati e si desidera adattarlo se il comando è molto semplice e naturale:

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

dove invece `f(x)` potrebbe essere anche `f(x, y)`. Nel caso in cui si abbiano anche stime di errori di dati, basta aggiungere `{y | xy | z}errors ({ | })` rappresentano le possibili scelte) nell'opzione *modificatore* (vedi **sintassi**). Per esempio

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

dove il `{y | xy | z}errors` opzione di `{y | xy | z}errors` richiede rispettivamente 1 (`y`), 2 (`xy`), 1 (`z`) colonna che specifica il valore della stima dell'errore.

xyerrors esponenziale con xyerrors di un file

Le stime degli errori dei dati vengono utilizzate per calcolare il peso relativo di ciascun punto dati quando si determina la somma ponderata dei residui quadratici, WSSR o chisquare. Possono influenzare le stime dei parametri, poiché determinano l'influenza che la deviazione di ciascun punto di dati dalla funzione adattata ha sui valori finali. Alcune delle informazioni di output di adattamento, incluse le stime degli errori dei parametri, sono più significative se sono state fornite stime accurate degli errori dei dati. (*Ibidem*)

Prendiamo un campione di dati `measured.dat`, composto da 4 colonne: le coordinate dell'asse x (Temperature (K)), le coordinate dell'asse y (Pressure (kPa)), le stime dell'errore x (`T_err` (K)) e le stime dell'errore 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

Adesso componi il prototipo della funzione che dalla teoria dovrebbe approssimare i nostri dati. In questo caso:

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

dove abbiamo inizializzato il parametro Z perché altrimenti la valutazione della funzione esponenziale $\exp(x * Z)$ produce valori enormi, che porta a (in virgola mobile) Infinity e NaN nell'algoritmo di adattamento Marquardt-Levenberg, in genere non è necessario inizializzare il variabili - date un'occhiata [qui](#), se volete saperne di più su Marquardt-Levenberg.

È il momento di adattare i dati!

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

Il risultato sarà simile

```
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
```

Dove ora w e Z sono riempiti con i parametri desiderati e le stime degli errori su quelli.

Il codice sottostante produce il seguente grafico.

```
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'
```

Terreno con impeto di `measured.dat` Utilizzando il comando `with xyerrorbars` visualizzerà errori stima sulla x e y. `set grid` posizionerà una griglia tratteggiata sui tic principali.

80

70

, che include i metri quadrati di una casa in una determinata città e il suo prezzo in \$ 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
```

Adatta questi parametri con *gnuplot* Il comando stesso è molto semplice, come puoi notare dalla sintassi, basta definire il tuo prototipo di adattamento, e poi usare il comando `fit` per ottenere il risultato:

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

Ma potrebbe essere interessante anche usare i parametri ottenuti nella trama stessa. Il seguente codice si inserisce il `house_price.dat` file e quindi tracciare le m e q parametri per ottenere la migliore approssimazione della curva del set di dati. Una volta che hai i parametri puoi calcolare il y -value, in questo caso il *prezzo della casa*, da un dato x -vaule (*metri quadrati della casa*) appena sostituendo nella formula

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

il x -value appropriato. Diamo un commento al codice.

0. Impostazione del termine

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

1. Amministrazione ordinaria per abbellire il grafico

```
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. La giusta misura

Per questo, dovremo solo digitare i comandi:

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

3. Salvare i valori di m e q in una stringa e tracciare

Qui usiamo la funzione `sprintf` per preparare l'etichetta (racchiusa nel `object rectangle`) in cui stamperemo il risultato della misura. Infine tracciamo l'intero grafico.

```
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
```

L'output sarà simile a questo.

800

700

600

500

(\$1000)

<https://riptutorial.com/it/gnuplot/topic/8825/adatta-i-dati-con-gnuplot>

Capitolo 3: Stili di plottaggio 2D

Examples

Selezione di uno stile di tracciamento

Selezione esplicita

Solitamente uno stile di tracciamento viene selezionato usando la parola chiave `with`, come

```
plot x with points
```

Ciò consente di utilizzare diversi stili di stampa per ogni `plot` :

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

Digitando `help with` nella finestra dei comandi gnuplot fornisce un elenco di tutti gli stili di plottaggio disponibili.

Selezione dello stile di tracciamento globale

Gli stili di plottaggio possono anche essere impostati globalmente per tutti i comandi di stampa. Qui gnuplot distingue tra grafici di funzioni e di dati, per i quali è possibile impostare diversi stili di default.

Per le funzioni usa la `set style function` :

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

Per i file di dati utilizzare i `set style data` :

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

Si noti che per le funzioni lo stile predefinito è `lines` e per i file di dati si tratta di `points`. Con i `show style data` `show style function` e la `show style function` è possibile esaminare gli stili di disegno attualmente selezionati.

Leggi Stili di plottaggio 2D online: <https://riptutorial.com/it/gnuplot/topic/4302/stili-di-plottaggio-2d>

Capitolo 4: Tracciamento di base di file di dati

introduzione

Una delle principali caratteristiche utili di *gnuplot* è la possibilità di tracciare i **file di dati** . Tracciare un file di dati è davvero semplice con *gnuplot* , in realtà, una volta aperto il software dal terminale, è sufficiente digitare il comando `plot 'file'` per ottenere un grafico automatico.

Prima di tutto, prima di stampare, devi essere sicuro di trovarti nella stessa directory in cui si trova il file di dati, altrimenti alla fine riceverai un `warning` .

Sintassi

- trama *file di dati* usando *column_expression* con *stile*

Examples

Traccia un singolo file di dati

Il `plot` comando *gnuplot* predefinito (anche solo `p`) traccia il set di dati con le colonne, sotto forma del file `data_set.dat` sottostante.

```
# 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
```

Come puoi vedere puoi scrivere nel tuo set di dati in notazione a virgola mobile. Ora tutto è pronto per tracciare i dati: digitando solo

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

gnuplot produrrà un grafico nella destinazione di `output` . Le impostazioni predefinite useranno le prime due colonne del tuo file di dati, rispettivamente `key`. Per specificare le colonne da tracciare usa lo specificatore `using`

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

che significa "traccia il file usando la colonna 2 come X e la colonna 4 come Y". Nel caso in cui il set di dati sia un file tridimensionale, basta usare `splot` ad aggiungere la z-column

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

Ci sono anche stili diversi (vedi documentazione gnuplot o [Selezione di uno stile di tracciamento](#) per ulteriori informazioni) per i punti di tracciamento. Come detto prima, lo stile predefinito è il `point`

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

che tratterà lo stesso come se non si digiti `with point`. Uno stile utile per il `linespoint` dati è il punto di `linespoint` che è, ovviamente, "linee + punti". **PER ESEMPIO:**

```
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

for iterazione risultati ciclo molto utile:

```
p for [col = 2:4] "data_set.dat" using 1:col w lp
```

che dà l'output

40

35

, con passaggi deciso (se non specificato = 1). Ad esempio `for [i = 0:6:2]` incrementerà `i` da 0 a 6 in 2 passaggi: `i = 0, 2, 4, 6`. Tutti i valori (start, stop e increment) vengono convertiti in valori interi.

* Grid

La griglia è spesso utile quando si traccia un set di dati. Per aggiungere un tipo di griglia

```
set grid
```

Tracciare più file di dati

Primo metodo - Concatenazione di stringhe

Il metodo più semplice per tracciare più file di dati è inserire un ciclo `for` all'interno del comando `plot` di gnuplot. Supponendo che tu abbia `N` file chiamati sequently, `ie`

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

Esecuzione del comando

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

`file_1.dat` tutti i file tra `file_1.dat` e `file_N.dat` nello stesso grafico.

Esempio con tre file di dati

Tabella dei set di dati

X-Assi	Y-Ax file_1.dat	Y-Ax file_2.dat	Y-Ax file_3.dat
1	1	1	1
2	2	4	2
3	3	9	6
4	4	16	24
5	5	25	120

comandi

```
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
```

Il ciclo inizia con `for [i = 1:3] "file_".i.".dat"` ed esegue il comando `plot` fino a raggiungere `i = 3`. Il `.i.` è il numero concatenato.

`title "file_".i.".dat"` è stato scritto con `\` per far sì che il simbolo `_` nel nome dei file appaia come carattere di *sottolineatura* piuttosto che come *indice*, e l' `noenhanced` è fondamentale per ottenere questo risultato.

Il risultato finale è mostrato sotto

120

100

che funziona fondamentalmente come lo `printf` linguaggio C. La sintassi corretta è la [documentazione di gnuplot 5.1](#)

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

Un breve esempio chiarirà ogni dubbio.

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

Leggi Tracciamento di base di file di dati online:

<https://riptutorial.com/it/gnuplot/topic/3591/tracciamento-di-base-di-file-di-dati>

Capitolo 5: Utilizzando i file di script

Sintassi

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

Osservazioni

L'utilizzo di base può essere visualizzato digitando `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

Semplice file di script

Gnuplot è in grado di generare un'immagine da un file di script che consente una sequenza di comandi necessari per disegnare un'immagine da eseguire in sequenza anziché digitare manualmente.

Ai fini di questo esempio creeremo un semplice script per disegnare un `sin(x)`.

Crea un file di script

Creare un file `sinx.p` con il seguente contenuto:

```
# 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)
```

Nell'esempio sopra trovi i comandi più comuni, tuttavia, ci sono molti altri comandi da esplorare come `set xlabel`, `set ylabel`, ecc.

È possibile personalizzare la riga di `set output` con il percorso in cui si desidera che il file generi il

file.

Esegui lo script

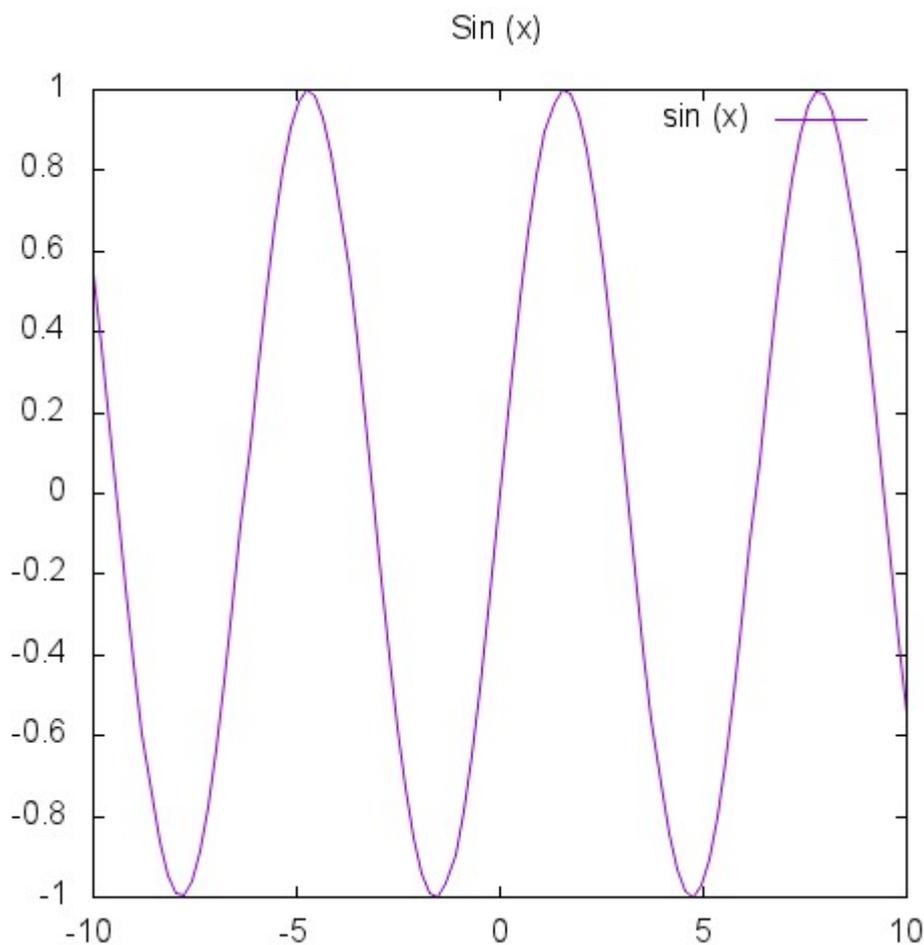
Apri il tuo terminale e digita:

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

Nel caso in cui la cartella corrente contenga lo script, è possibile inserire quanto segue:

```
gnuplot sinx.p
```

Lo script verrà eseguito e genererà il file PNG nella posizione specificata. Il grafico risultante dovrebbe essere simile al seguente:



Leggi Utilizzando i file di script online: <https://riptutorial.com/it/gnuplot/topic/4013/utilizzando-i-file-di-script>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con Gnuplot	Community , Fawix , opisthofulax
2	Adatta i dati con gnuplot	opisthofulax
3	Stili di plottaggio 2D	Christoph
4	Tracciamento di base di file di dati	Christoph , Matthew , opisthofulax , Tom Solid
5	Utilizzando i file di script	Christoph , Fawix