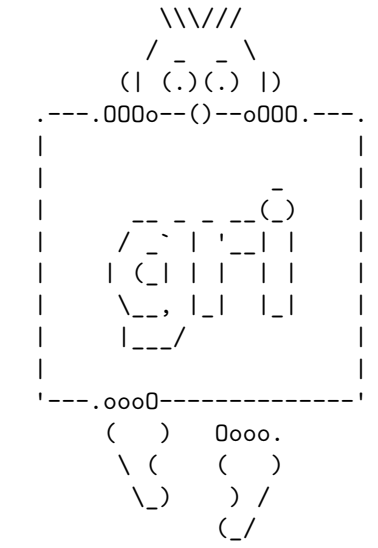


```
Codeblock class: gri

    sudo apt-get install gri imagemagick
    http://gri.sourceforge.net
    Notes
    - insists on creating a <fname>.ps in current working directory
    - requires `convert` from imagemagick
    - ImageMagick's security policy might need massaging

    runs:
    > gri {im_opt} -c 0 -b <fname>.gri

    class->cmd
      gri -> gri

Metadata options
    imagine.im_out: img,fcb
    imagine.im_log: 4
```

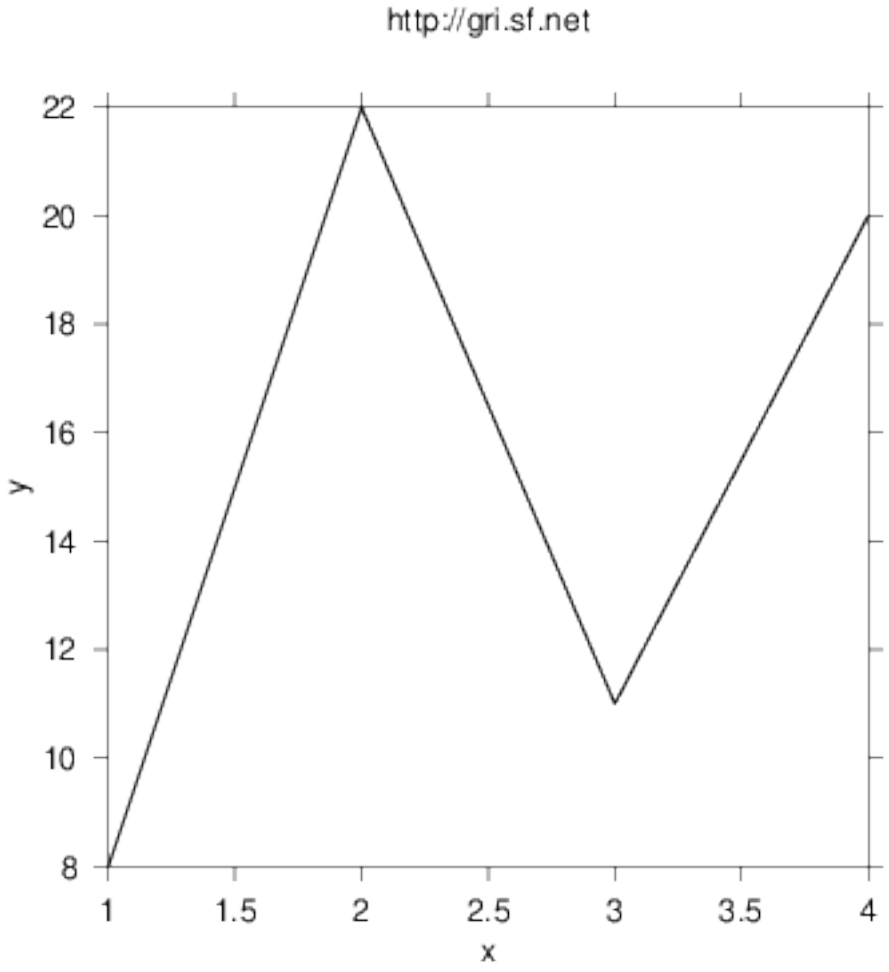
GRI

Single plot

With the following in `gri-01.dat`

```
1  8 11  9
2 22 21 20
3 11 10  9
4 20 15 10
```

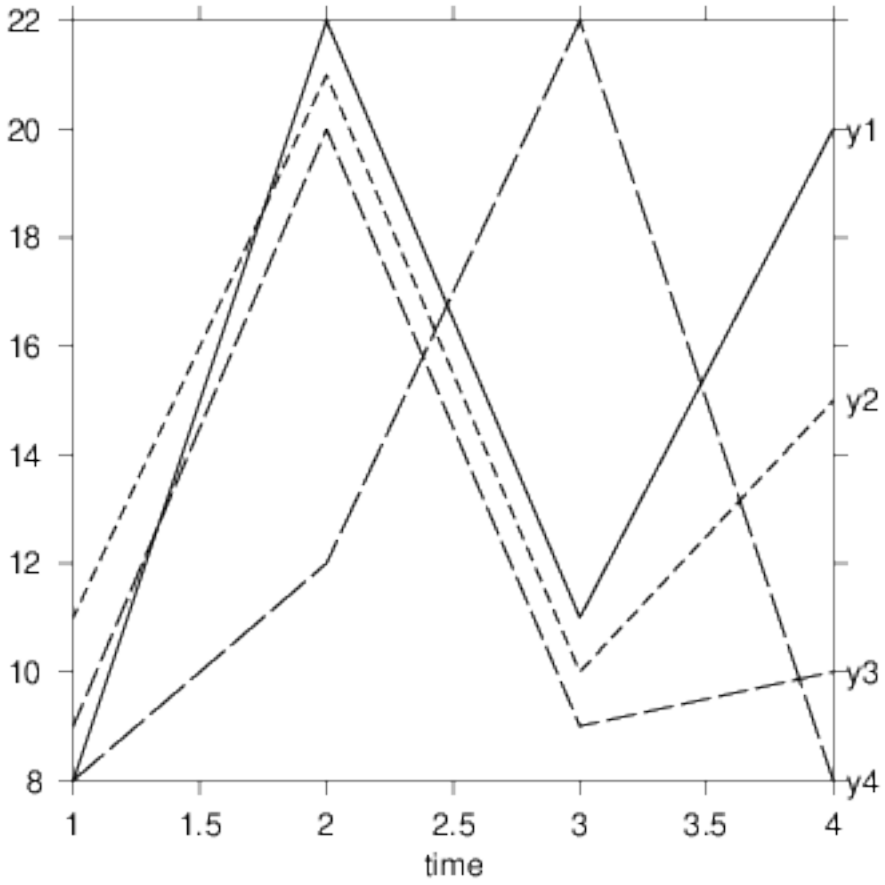
plot the first two columns like so:




```gri
open dta/gri-01.dat
```

```
read columns x y
draw curve
draw title "http://gri.sf.net"
```

Multiple curves




```gri
`draw curves' \xname \y1name ...'`
Draw multiple y columns versus an x column.  Assumes
that the datafile is open, and that x is in the first
column, with the y values in one or more following
columns.


The number of columns is figured out from the options,
as is the name of the x-axis, and the labels to be
used on each of the y curves.
{
  # NB. the 3 below lets us skip the words 'draw'
  # and 'curves', and the name of the x-column.
  .num_of_y_columns. = {rpn wordc 3 -}
```

```
  if {rpn .num_of_y_columns. 1 >}
    show "ERROR: 'draw curves' needs at least 1 y column!"
    quit
  end if


  set x name {rpn 2 wordv}
  set y name ""


  # Loop through the columns.
  .col. = 0
  while {rpn .num_of_y_columns. .col. <}
    # The x-values will be in column 1, with y-values
    # in columns 2, 3, ..., of the file.
    .ycol. = {rpn .col. 2 +}
    rewind
    read columns x=1 y=.ycol.
    # At this point, you may want to change line thickness,
    # thickness, color, dash-type, etc.  For illustration,
    # let's set dash type to the column number.
    set dash .col.
    draw curve
    draw label for last curve {rpn .col. 3 + wordv}
    .col. += 1
  end while
}


open dta/gri-01.dat
draw curves time y1 y2 y3 y4
```

Documentation

gri -h

```
NAME
     gri - draw scientific graphs

SYNOPSIS
     gri [OPTIONS] [command_file [optional_arguments]]

DESCRIPTION
     If a command file (command_file) is named, commands are read from that file;
     otherwise they are read from the keyboard.  If a command_file is named, then
     a file in which to store the PostScript output may also be named; otherwise
     it is stored in a file named gr-00.ps (or gr-01.ps if gr-00.ps exists, etc).

     There are 3 special forms that do no graphing:
       `gri -creator postscript_file'
         Extracts the Gri commands that created the Gri PostScript file,
         but only if the Gri invocation that created the PostScript file
         had used the -no_private commandline option, or if the version of
         Gri that produced the file was earlier than 2.12.10.
       `gri -help' or `gri -h'
         Prints this help message.
       `gri -version' or `gri -v'
         Prints the version number of Gri.

     In normal usage, where drawing is expected, Gri takes these options:
     -batch or -b
             Stops printing of prompts and hints.
     -chatty[N] or -c[N]
             Let gri print info messages
     -debug or -d
             Turns debugging on (sets variable ..debug.. to value 1).
     -warn_offpage
             Warn if any item is drawn far off a 8.5x11" page.
             (This is the default.)
     -no_warn_offpage
             Don't warn if any item is drawn far off a 8.5x11" page
     -directory pathname
             Specifies the directory where Gri looks for startup files;
             otherwise it looks in /opt/gri/lib or at whatever
             directory is defined in configure shellscript, at compile time.
     -directory_default
             Reports directory where gri.cmd should be found,
```

```
        if not supplied by -directory.
-no_bounding_box
        Make bounding-box be full page.
-no_expecting
        Prevent warning message if `expecting version .n.'
        command is missing.
-no_startup_message
        Stops printing of startup message.
-output file_name
        Specify the name of the file to hold the graphical output.  If
        this flag is not specified, the file will be PostScript,
        and its name will be derived from the name of the
        commandfile, e.g. `mygraph.gri'
        will produce `mygraph.ps'), or, for interactive use,
        it will have a name like `gri-00.ps', or
        `gri-01.ps' if the former file exists, etc.
-private
        Prevents inserting any information about the user into
        the PostScript file (see -no_private, next).  As of
        version 2.12.10, this privacy option is assumed by default.
-no_private
        Instructs Gri to include comments in the PostScript file that
        identify the user, state the commandline arguments used in
        invoking Gri, and that list all the commands that were executed.
        This information can be recovered by calling Gri on the
        PostScript file, with the -creator commandline argument.
        Until version 2.12.10, the default was to include this
        information, but a change was made out of privacy concerns.
-publication or -p
        Sets the builtin variable ..publication.. to 1; normally it is 0.
        One might use if statements (`if !..publication..' ...) on drafts.
-superuser
        Used mainly by Gri programmers (who can check the value with the
        C function `superuser()'.)  An optional value can be supplied
        without spaces (e.g. `-s2') to set the debugging level.
        The flags are as follows:
            1: print cmdline before/after substituting synonyms
            2: print cmdline before/after substituting rpn expressions
            4: print new commands being defined
            8: print system commands and `open "... | "'
               commands before they are passed to the system
          128: for author's use only
          256: for author's use only
        Note that all flags are equal to 2 raised to an
        integer power.  Since the flag values are detected by
        a bitwise OR, you can combine flags by adding; thus
```

```
        specifying a flag of 5 yields flags 1 and 4 together; specifying
        15 yields flags 1, 2, 4 and 8.
-trace or -t
        Makes Gri print out command lines as they are executed.
-true or -y
        Makes Gri think the answer to all `query's is RETURN.
```

man page

GRI(1) General Commands Manual GRI(1)

NAME

gri - scientific graphics language

SYNOPSIS

gri [OPTIONS] [CommandFile [optional_arguments]]

DESCRIPTION

Gri is a programming language for scientific graphics. It can make x-y graphs, contour-graphs, and image graphs. In addition, Gri has a full suite of low-level graphical elements and sufficient programming capabilities (loops, subroutines, etc) to permit complex customization.

Gri is not point-click. In some ways it is analogous to TeX. Extensive power rewards tolerance of a modest learning curve.

OPTIONS

If a command file (CommandFile) is named, commands are read from that file; otherwise they are read from the keyboard. If a command file is named, then a file in which to store the PostScript output may also be named; otherwise it is stored in a file named by substituting the .ps extension instead of .gri in CommandFile. If no command file is named, the output is named gri-00.ps (or gri-01.ps if gri-00.ps exists, etc).

There are 3 special forms that do no graphing:

`gri -creator postscript_file'

Extracts the Gri commands that created the Gri PostScript file.

`gri -help' or `gri -h'

Prints this help message.

`gri -version' or `gri -v'

Prints the version number of Gri.

In normal usage, where drawing is expected, Gri takes these

options:

-batch or -b

Stops printing of prompts and hints.

-chatty[N] or -c[N]

Let gri print info messages

-debug or -d

Turns debugging on (sets variable ..debug.. to value 1).

-warn_offpage

Warn if any item is drawn far off a 8.5x11 inch page. (This is the default.)

-nowarn_offpage

Don't warn if any item is drawn far off a 8.5x11 inch page

-directory pathname

Specifies the directory where Gri looks for startup files; otherwise it looks in /opt/gri/lib or at whatever directory is defined in configure shellscript, at compile time.

-directory_default

Reports directory where gri.cmd should be found, if not supplied by -directory.

-no_bounding_box

Make bounding-box be full page.

-no_expecting

Prevent warning message if `expecting version .n.' command is missing.

-no_startup_message

Stops printing of startup message.

-publication or -p

Sets the builtin variable ..publication.. to 1; normally it is 0. One might use if statements (`if !..publication..' ...) on drafts.

-superuser or -s

Used only by Gri programmers (who can check the value with the C function `superuser()'.) An optional value can be supplied without spaces (e.g. `-s2') to set the debugging level. Flags are listed below; add flags to get several actions at once

1: print cmdline before/after substituting synonyms

2: print cmdline before/after substituting rpn expressions

4: print new commands being defined

8: print system commands and `open "... | "' commands before

they are passed to the system

128: for author's use only

256: for author's use only

Note that all flags are equal to 2 raised to an integer power. Since the flag values are detected by a bitwise OR, you can combine flags by adding; thus specifying a flag of 5 yields flags 1 and 4 together; specifying 15 yields flags 1, 2, 4 and 8.

-trace or -t

Makes Gri print out command lines as they are executed.

-true or -y

Makes Gri think the answer to all `query's is RETURN.

SEE ALSO

For more information, please consult online info and html manuals.

The info manual included in the main gri Debian package is normally accessed by typing

info gri

(or from within Emacs' own info).

There are also reference cards in postscript format. See /usr/share/doc/gri/*refcard.ps

The Debian package gri-html-doc provides the html manual, which when installed is then located at

/usr/share/doc/gri/html/index.html

or, if you have a web server installed, at

http://localhost/doc/gri/html/index.html

The HTML manual is accessible via dwww and dhelp Debian help interfaces. The html FAQ is located at

/usr/share/doc/gri/html/FAQ.html

The gri-html-doc package also includes examples in /usr/share/doc/gri/examples/ which are described in the manual, and are included as a quick start primer.

The gri-pdf-doc package is a PDF version of the manual suitable for printing.

GRI_MERGE AND GRI_UNPAGE COMMANDS

Two Perl scripts are provided with Gri to manipulate the PostScript output.

gri_merge is used to merge multiple Gri output files into a single PostScript file. See gri_merge -h and its man page for usage information.

gri_unpage is used is split a multi-page Gri output file (in which the new page command was used) into separate PostScript files, one for each page.

See their respective man pages.

EMACS SUPPORT

An emacs mode is provided with Gri. It is documented in the gri Info or HTML manual.

The mode is installed automatically in Debian by the elisp file:

/etc/emacs/site-start.d/50gri-el.el

The emacs mode itself is gri-mode.el and is installed on Debian as /usr/share/emacs/site-lisp/gri-mode.el

Byte-compiled versions of this file are produced for every flavour of Emacs that is installed, and are located in places like /usr/share/emacs/23.1/site-lisp/gri-el/gri-mode.elc

SEE ALSO

gri_merge(1), gri_unpage(1)

AUTHOR

This manual page by Peter S Galbraith <psg@debian.org>.