Overview of Gri 2.8 Commands

1 Introduction

This reference card describes the commands in version 2.8 of the Gri plotting language. See also the companion “Gri Reference Card” and the online manuals.

2 Control Statements

2.1 If Statements

The if statement has ancillary else if and else statements, and is ended by the end if statement, e.g.

```c
if (rpn .i, 10 >)
  show "The variable .i, is less than 10"
else if (rpn .x > 20)
  show "The variable .x, is between 10 and 20"
else
  show "The variable .x, is greater than 20"
end if
```

2.2 Loops

while loops are provided. The statements between while and end while are repeated until the RPN expression on the while command line is false.

Here is an infinite loop ended by a break statement when the file contents are exhausted:

```c
while i
  read .x, .y
  if .eof.
    break
  end if
show ",x, is , .x."
end while
```

Here is a loop that will print the numbers 0, 1, ..., 9, i = 0 while (rpn .i, 10 >)

```c
show .i, .i = 1
end while
```

3 List of Gri Commands

What follows is a complete list of built-in Gri commands. For more help on a given command, see the full manual, or use the Gri online help facility (e.g., type gri to launch Gri, then type help; exit by typing)

The notation is as follows:

- Items written within square brackets are optional.
- Items written within dots are either raw numbers, RPN expressions, or variable names.
- Items preceded by backslashes are any given string.
- Items separated by vertical bars are alternatives.
- Curly brackets group words that must appear together.

Thus, for example, the syntax

```c
set dash [n.][dash, .blank, ][off]
```

means that set dash is a possible Gri command (meaning use the default dash style). Several forms of optional items may be present also. For example, set dash 2 is legal; it means use the dash style numbered 2. Gri will check any single number presented in this place on this command against the list of acceptable ,n, values. If two numbers are present, Gri interprets the first as the length of dashes and the second as the length of blanks; notice the braces indicating that these two parameters must appear together. Finally, the keyword off is allowed (it means go back to a solid line).

Here are the commands:

```c
assert condition, ["message"]
cd ([pathname])
close ([]filename)
copy columns to grid [neighbor[]{objective}boxcar, .x, .y, [.z, .e, .i] [barnes [.x, .y, .gamma, .iter]}
copy columns to spline [gamma, .alpha], [.min, .max, .xinc]}
copy grid to columns
copy grid to image [size, width, height] [box .ll.x, .ll.y, .ur.x, .ur.y]
copy image to grid
cREATE columns from function
create image grayscale banded, .band, create image grayscale banded, .band, drawaxis (n] ][clipped values in drawcommand] [off
delete [variable, [synonym, ...]] [columns [[randomly .fraction, .where missing]] [grid, [iter] scale]}
differentiate [x, y, w, i, y, x] [grid, w, i, y, x]
draw arrows from .x0, .y0, to .x1, .y1, [cm]
draw arrows
draw axes if needed
draw axes, [style, [frame, none]}
draw border box [box .ll.x, .ll.y, .ur.x, .ur.y, .width, .cm, .brightness]}
draw box filled, ll.x, ll.y, ur.x, ur.y, [cm]
draw circle with radius, .r, cm, at .x, .y, .cm}
draw contour [[value, [unlabelled, labelled]["label", ]]
[.min, .max, .inc, [.unlabelled, [unlabelled]]]}
draw curve overlying
draw curve filled [to [.y, y, [.x, x, x]]}
draw curve
draw curve [text, [reset]
draw grid [x, .xcm, .ycm, .height, .cm, .style, [fgcolor, [bgcolor]}
draw grid
draw image [palette [axis], left, [axis], right, [axis], top, [axis], bottom] [left, .l, right, .r, .right, .right, [increment, .inc]] [box .ll.x, .ll.y, .ur.x, .ur.y, .cm, .width, .cm, .height, .cm, .style, [fgcolor, [bgcolor]}
draw image grayscale [left, .l, right, .r, .right, .right, [increment, .inc]] [box .ll.x, .ll.y, .ur.x, .ur.y, .cm, .width, .cm, .height, .cm, .style, [fgcolor, [bgcolor]}
draw image histogram [box .ll.x, .ll.y, .ur.x, .ur.y, .cm, .width, .cm, .height, .cm, .style, [fgcolor, [bgcolor]}
draw image

draw isopencil [unlabelled], [density, [.P, sigma, [.P, theta, a]]}
draw isopcn [unlabelled, [unlabelled]}
draw label boxed ["string", at .ll.x, .ll.y, .cm]}
draw label white under ["string", at .ll.x, .ll.y, .cm]}
draw label for last curve ["label",}
draw label ["string", [centered, right, justified] at .x, .y, [cm], [rotated, .deg]}
draw line from .x0, .y0, to .x1, .y1, [cm]}
draw line legend ["label", at .x, .y, [cm], [length, .cm]}
draw lines [vertical, [left, .right, .inc]] [horizontally, [bottom, .top, .inc, .]}]
draw patches, [.width, .height, .cm]}
draw polygon [filled, .x0, .y0, .x1, .y1, .x2, .y2, ...}
draw regression line [clipped]
draw symbol legend [symbol, name, ["label", at .x, .y, [cm]}
draw symbol [code, [name, at .x, .y, [cm]], [graylevel, ] 0][color, hue, .h, [brightness, .b, .b], [saturation, .s, .s]}
draw time stamp [fontsize, points, at .x, .y, .cm, [angle, .deg]}
draw title ["string", [draw values, [.dx, .dy], [format] [separator, .xcm, .ymcm]}
draw x axis at [bottom, [top], [.y, [.cm]], [.lower, [.upper]]}
draw x box plot at .y, [.size, .cm]}
draw y axis at [left, [right], [.x, .cm], [.left, [right]]}
draw y box plot at .x, [.size, .cm]}
draw zero line [horizontally, [v), vertically]
expecting version, .n}
filter columns [x, .y] recursively .a0, .a1, ... .b0, .b1, ...
filter grid rows,[columns recursively .a0, .a1, ... .b0, .b1, ...
filter image highpass, [lowpass]
fill gridimage [x, y]}
get env [result, environment variable]
help [command, name, [topic]}
if [[], [flag, [flag, ["string", == "string2",]]]
ignore last, .n}
input [ justification, [.xcm, .ymcm, [.xmag, .ymag, [.rot, .deg]]]
```
set x margin {[bigger][smaller] .size.} | default
set x margin .size.
set x margin bigger .size.
set x margin smaller .size.
set x margin default
set x name "\name"|default
set x size .width_cm.|default
set x size .width_cm
set x size default
set y axis label horizontal|vertical
set y axis label horizontal
set y axis label vertical
set y axis left|right|increasing|decreasing|[.bottom,.top,[.incBig,.incSml.]]|unknown
set y axis left
set y axis right
set y axis increasing
set y axis decreasing
set y axis .bottom,.top.
set y axis .bottom,.top,.incBig,
set y axis .bottom,.top,.incBig,.incSml.
set y format \format|default|off
set y grid .bottom,.top,.inc|[./.rows.]
set y grid .bottom,.top,.inc,
set y grid .bottom,.top,../.rows.
set y margin {[bigger][smaller] .size.} | default
set y margin .size.
set y margin bigger .size.
set y margin smaller .size.
set y margin default
set y name "\name"|default
set y size .height_cm.|default
set y size .height_cm
set y size default
set y type linear|log|map 2|5|8
set z missing above|below .intercept.|.slope.
set "...
show all
show axes
show color
show colornames
show columns [statistics]
show flags
show grid [mask]
show hint of the day
show image
show license
show misc
show next line
show traceback
show stopwatch
show synonyms
show time
show variables
show .value. | {rpn .} | "\text" {.value.}{rpn .} |text
[...]
skip [forward|backward] [.n.]
sleep .sec.