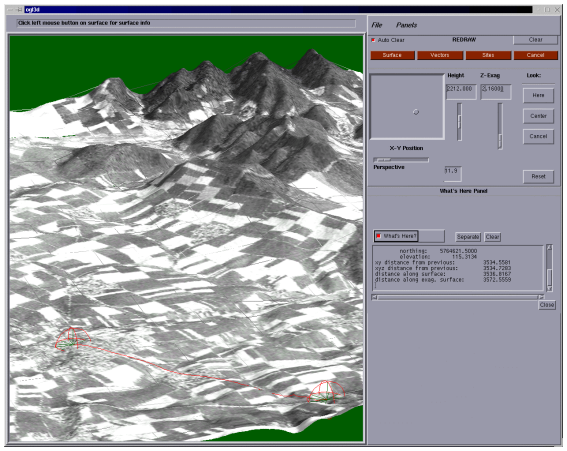
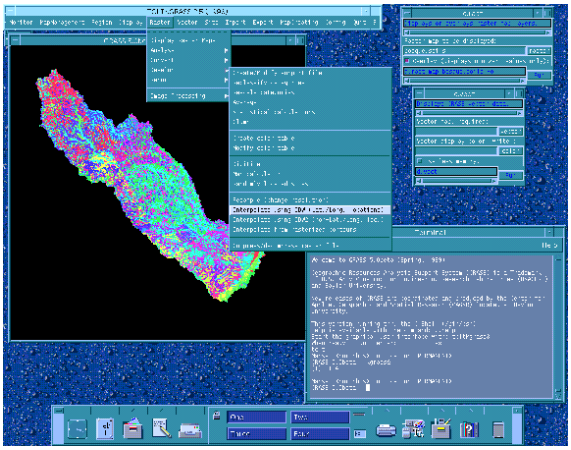
GRASS Reference Manual

Shell Script Commands





GRASS Development Team

USA Headquarters
Center for Applied Geographic & Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas 76798-7351
USA

European Headquarters
Institute of Physical Geography-Landscape Ecology
University of Hannover
Schneiderberg 50
30167 Hannover
Germany

grass@baylor.edu

http://www.baylor.edu/~grass
http://www.geog.uni-hannover.de/grass/

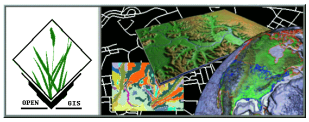


Table of Contents

GRASS Introduction

GRASS (Geographic Resources Analysis Support System) is a raster based GIS, vector GIS, image processing system, and graphics production system. GRASS contains over 200 programs and tools to render maps and images on monitor and paper; manipulate raster, vector, and sites data; process multi-spectral image data; and create, manage, and store spatial data. GRASS uses both an intuitive windows interface as well as command line syntax for ease of operations. GRASS can interface with commercial printers, plotters, digitizers, and databases to develop new data as well as manage existing data.

GRASS is ideal for use in engineering and land planning applications. Like other GIS packages, GRASS can display and manipulate vector data for roads, streams, boundaries, and other features. GRASS can also be used to keep maps updated with its integral digitizing functions. Another feature of GRASS is its ability to use raster, or cell, data. This is particularly important in spatial analysis and design. GRASS functions can convert between vector data to raster data for seamless integration.

GRASS' strengths lie in several fields. The simple user interface makes it an ideal platform for those learning about GIS for the first time. GRASS is capable of reading and writing maps and data to many popular commercial GIS packages including ARC/Info and Idrisi. Users wishing to write their own code can do so by examining existing source code, interfacing with the documented GIS libraries, and using the GRASS Programmers Manual. This allows more sophisticated functionality to be integrated in GRASS.

The ability to work with raster data gives GRASS the unique ability to function as a surface modeling system. GRASS contains more than 100 multi-function raster analysis and manipulation commands. Surface processes such as rainfall-runoff modeling, flowline construction (as shown), slope stability analysis, and spatial data analysis are just a few of the many applications of GRASS to engineering and land planning. Since many of the raster tools are multi-functional, users can create their own maps from GRASS data analysis.

In addition to standard two-dimensional analysis, GRASS allows users to view data in three-dimensions. Raster maps, vector maps, and sites data can be used for visualization. Example applications of such capabilities include airspace analysis for airport planning (as shown), terrain analysis and “flybys”, and spatial trends. Tools in GRASS allow the user to animate any spatial data available with options to switch between data layers “on-the-fly”. Data used in 3-D visualization may also be saved as still pictures, or as mpeg movie files for later replay and analysis.

Accompanying its land planning and engineering applications, GRASS contains a suite of tools to aid in hydrologic modeling and analysis. Currently, tools are also available for performing such functions as watershed analysis, curve number generation, flood analysis, and stream channel characteristics for comprehensive watershed modeling. Other GRASS programs can generate graphs, statistics, and charts of modeled and calibrated data. Additionally, GRASS can use field data for model input or simulate parameters based on numerical data.

In addition to the traditional command line version of GRASS, a new user interface, based on Tcl/Tk has been written. This puts the power of spatial analysis and modeling into an easy to use Graphical User Interface that is platform-independent. This intuitive user interface lets users quickly and easily view, manipulate, and use data. Nearly all of the programs available in GRASS are available in the new GUI, with the standard command-line still available, giving users all of the functionality of GRASS.

This manual is part of a comprehensive set of documentation written to support GRASS. This Users Guide consists of a complete set of command references for all current GRASS functions and tools, including examples. An installation guide and fact sheet guides users through the installation process. For those wishing to write their own spatial analysis and modeling applications for GRASS, a Programmers Guide is also available. GRASS runs on a variety of UNIX and Linux platforms including SUN SPARCstations and Ultras, HP, Silicon Graphics, and PC’s running Windows 95 and Windows NT.

The GRASS Development Team is currently working to further upgrade and enhance the capabilities of GRASS. Future developments include tools that give the user the ability to work completely in 3-D, a capability that does not exist in any other GIS package. Users will be able to work with raster elevation data as well as vector and sites data in the 3-D environment, adding to the

visualization capabilities of GRASS. Enhancements in the numerical processing functions of GRASS also now allow for floating-point operations to be performed on data.

For the latest information on GRASS contact the GRASS Development Team at grass@baylor.edu or visit our web sites at:

http://www.baylor.edu/~grass if you're in the U.S.

http://www.geog.uni-hannover.de/grass if you're in Europe

Look for our worldwide mirrors!

The GRASS Development Team is:

Bruce Byars and Markus Neteler are the development team leaders and coordinators.

Helena Mitasova and Bill Brown of the GMS Lab at UIUC have made significant contributions with the development of GRASS 5.

Additional authors include:
Lisa Zygo, Edward Zarecky, Jacques Bouchard, Steve Clamons, Brent Duncan, Jason Cipriano, Jim Westervelt, Michael Shapiro, Darrell McCauley, Dave Gerdes, Bill Hughes, Bernhard Reiter, Brook Milligan, Eliot Cline, Jaro Hofierka, Clay Cockrell, and Bob Lozar. See the web pages for author affiliations.

Note:

Many other people have contributed to the GRASS GIS. Without any one of them, GRASS would not exist in its current form. The authors of the individual programs are listed at the end of their manual page in the GRASS users manual, however, numerous authors of bug fixes and enhancements as well as people who have been working on coordination, integration, documentation and testing are not mentioned.

Please allow us to extend our most cordial thanks to all of you. If you contributed to GRASS at any point during its existence, let us know your name and e-mail address so we can add your name to the comprehensive on-line list.

To reference GRASS:

GRASS Development Team, 1999, Geographic Resources Analysis and Support System - GRASS: Baylor University, Waco, Texas.

GRASS Development Team
Center for Applied Geographic and Spatial Research
Baylor University
P.O. Box 97351
Waco, Texas, U.S.A. 76798-7351

3d.view.sh

NAME
3d.view.sh - Displays several 3-dimensional views of a landscape on the user's graphics monitor. (GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
3d.view.sh
3d.view.sh help
3d.view.sh file=mapname ef=mapname vh=viewing_height sv=sink_value exag=exag lf=line_frequency back=background_color

DESCRIPTION
3d.view.sh is a Bourne shell (sh(1)) script that displays several 3-dimensional views of a landscape on the user's graphics monitor. It erases the graphics monitor and then prepares it for the display of nine equally-sized frames. The user-specified raster map layer (given by file=name) is displayed *d.rast* in the middle frame. The remaining frames are then used to display 3-d perspective views. The top middle panel is a view from the north, the top right from the north-east, the right from the east, and so on. Each is drawn with a call to the *d.3d* program. The viewing angles are calculated automatically.

OPTIONS
If options are not stated on the command line, default values will be used. These values are listed under Parameters, below.

Parameters:
file=mapname Name of raster map layer to be displayed.
Default: elevation

ef=mapname Name of raster map layer whose category values will supply the elevation values used to generate 3-d perspective views.
Default: elevation

vh=viewing_height Height (in meters) of the location from which scenes will be viewed.
Default: 30000

sv=sink_value Sink factor value, causing the image to be displayed lower, or higher, on the graphics screen.
Default: 0

exag=vertical_exaggeration Vertical exaggeration factor of the values in the elevation file.
Default: 3

lf=line_frequency Contour intervals at which vector grid lines will be drawn, in meters.
Default: 20

back=background_color Color of the background of the display frames.
Options: red, orange, yellow, green, blue, indigo, violet, magenta, brown, gray, white, and black
Default: black

NOTES
In the spearfish sample data base, the user must specify a viewing height when running *3d.view.sh*. Note also that the raster elevation map layers in the PERMANENT mapset under spearfish are named elevation.dem and elevation.dma.

This program will not prompt the user for inputs; if the user types *3d.view.sh* without program arguments on the command line, default values will be used.

FILES
This program is simply a shell script. Users are encouraged to make their own shell scripts using similar techniques. See $GISBASE/scripts/3d.view.sh.

SEE ALSO
d.3d, d.rast

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

DOS.show

NAME
DOS.show - Reads screen images stored on the PC6300 MS-DOS hard disk by *DOS.save*.
(SCS GRASS Display Program) (Available for PC6300 workstations, only)

GRASS VERSION
4.x

SYNOPSIS
DOS.show
DOS.show help

DESCRIPTION
This program allows a user to interactively view PC6300 screen images produced by *d.display*, *d.rast*, etc. and saved to the PC6300 hard disk by DOS.save.

NOTES
This program is for use with PC6300 workstations, only.

SEE ALSO
DOS.delete, DOS.list, DOS.save

AUTHOR
P.W. Carlson, USDA, SCS, NHQ-CGIS

blend.sh

NAME
blend.sh - Combines the red, green, and blue color components of two raster map layers.
(GRASS Shell Script)

SYNOPSIS
blend.sh file1 file2 perc outbase

DESCRIPTION
blend.sh is a Bourne shell (sh(1)) script that extracts the red (R), green (G), and blue (B) color components from each of two raster map layers, and creates three new raster map layers whose category values respectively represent the combined red, combined blue, and combined green color values from the two input layers. Category values in each of the output map layers will fall within the range of 0 - 255.

The R,G,B values from the two input map layers (file1 and file2) are not simply added together, but are instead combined by a user-named percentage (perc) of the R,G,B values in file1. Specifically, *blend.sh* executes three *r.mapcalc* statements that:

1) convert the R,G,B values in file1 and file2 to the range 0 - 255;

2) multiply the R, G, and B values in file1 by a user-named percentage (perc);

3) multiply the R, G, and B values in file2 by (100 - perc)%;

creating three new raster map layers, whose category values represent the summed R, summed G, or summed B values resulting from (2) and (3). Resulting R, G, and B values will respectively be stored in three new raster map layers named outbase.r, outbase.g and outbase.b.

OPTIONS
This program runs non-interactively; the user must state all parameter values on the command line.

Parameters:
file1 Name of a first raster map layer, whose R, G, and B color components will be combined with those of the second raster map layer (file2) named. The percent value (perc) given will apply to file1.

file2 Name of a second raster map layer, whose color components will be combined with those of file1. The percent value (perc) given will apply to the R,G,B values in file1. The R, G, and B values in file2 will be multiplied by (100 - perc)%.

perc Percentage or amount of the color contribution in terms of color intensity. This value is multiplied by the R,G,B values in file1.

outbase The root name assigned to each of the three output files created. A suffix is added to each file name, indicating which hold the red, green, and blue color values.

NOTES
blend.sh
executes three *r.mapcalc* statements:

```
r.mapcalc "outbase.r = r#file1 * .perc + (1.0 - .perc) * r#file2"
r.mapcalc "outbase.g = g#file1 * .perc + (1.0 - .perc) * g#file2"
r.mapcalc "outbase.b = b#file1 * .perc + (1.0 - .perc) * b#file2"
```

It uses the # operator to separately extract the red, green, and blue components in the named raster map layers, essentially allowing color separates to be made.

EXAMPLE
Typing the following at the command line:

blend.sh aspect elevation 40 elev.asp

will create three new raster map layers named elev.asp.r, elev.asp.g, and elev.asp.b, that, respectively, contain 40% of the red, green, and blue components of the elevation map layer and contain 60% of the red, green, and blue components of the aspect map layer.

FILES
This program is simply a shell script. Users are encouraged to make their own shell scripts using similar techniques. See $GISBASE/scripts/blend.sh.

SEE ALSO
r.colors, r.mapcalc

AUTHOR
Dave Gerdes, U.S. Army Construction Engineering Research Laboratory

bug.report.sh

NAME
bug.report.sh - A mechanism for writing, storing, and e-mailing to the GRASS Research Group users' bug reports on GRASS 4.1 commands.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
bug.report.sh 4.1_program.name [program_arguments]

DESCRIPTION
bug.report.sh is a Bourne shell (sh(1)) script which, when given a GRASS 4.1 program name, allows the user to complete a bug report for that program. Completed bug reports can be e-mailed to the GRASS development group at Baylor, saved to a file in the user's home directory, or discarded.

OPTIONS
This program is not interactive; the user must specify the name of a 4.1 program on the command line. Program arguments can optionally be entered by the user.

Parameters:
4.1_program.name The name of an existing GRASS version 4.1 program tested by the user.

program_argument(s) Program arguments (parameters and/or flags) for the 4.1_program.name tested by the user. The user should enter whatever command arguments were actually run when the program bug occurred.

EXAMPLE
For example, if the user wished to complete a bug report on *v.to.rast* after running the program, the user might then type:

bug.report.sh v.to.rast

A standard bug report form would then be displayed on the user's text terminal, containing the user's current GRASS region settings, and the machine name, GRASS data base, location, and mapset, on which the user is currently running GRASS. Program parameters and flag settings, and the command entered by the user on the command line, are also automatically entered on the bug report form. The user is put into a text editor and expected to enter additional information describing the nature of the bug found or the results of program testing.

The user is then asked whether the completed bug report is to be:

1 - mailed to westerve@zorro.cecer.army.mil

2 - added to the file grass.bugs in the user's home directory

3 - both 1 and 2

4 - thrown away

NOTES
This program prints whatever region settings, GRASS data base, location, and mapset are current when the user runs *bug.report.sh*. The user is therefore advised to run *bug.report.sh* immediately after experiencing a program bug, to ensure that the settings current when the bug occurred are reported on the bug report form.

FILES
This shell script is stored under the $GISBASE/scripts directory on the user's system. The user is encouraged to examine the shell script commands stored in this directory and to produce similar scripts for their own use. Users might modify this shell script to e-mail reports of program bugs to a local systems' administrator in addition to someone at USACERL.

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

d.6386.show

NAME
d.6386.show - Reads screen images stored on the PC6386 hard disk by *d.6386.save*.
"(SCS GRASS Display Program)
"(Available for PC6386 UNIX workstations, only)"

GRASS VERSION
4.x

SYNOPSIS
d.6386.show
d.6386.show help

DESCRIPTION
allows a user to interactively view PC6386 screen images produced by GRASS display programs (*d.display*, *d.rast*, etc.) and saved to the PC6386 hard disk by *d.6386.save*.

NOTES
This program is available for PC6386 UNIX workstations, only.

SEE ALSO
d.6386.delete, d.6386.save

AUTHOR
P.W. Carlson, USDA, SCS, NHQ-CGIS

d.rast.leg.sh

NAME
d.rast.leg.sh - Displays a raster map and its legend on a graphics window.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.rast.leg.sh
d.rast.leg.sh help
d.rast.leg.sh rast_map [num_of_lines]

DESCRIPTION
d.rast.leg.sh is a UNIX Bourne shell macro that clears the entire screen, divides it into a main (left) and a minor (right) frames, and then display a raster map in the main frame and the map legend in the minor frame. The main frame remains active when the program finishes.

OPTIONS
The user can run the program interactively or non-interactively.

Parameters:
rast_map A raster map to be displayed.

num_of_lines Number of lines to appear in the legend. If this number is not given, the legend frame will display as many lines as number of categories in the map, otherwise, it will display the first num_of_lines minus 1 categories with the rest being truncated.

NOTES

The user may adjust the num_of_lines parameter or the size of graphics window to get an appropriate result.

FILES
See the file *d.rast.leg.sh* under $GISBASE/scripts.

SEE ALSO
d.legend, d.rast

AUTHORS
Jianping Xu, Scott Madry, Rutgers University

d.zoom.last.sh

NAME
d.zoom.last.sh - Allows the user to change the current geographic region settings interactively and tracks the last region settings. (An addition to the GRASS display program *d.zoom*).
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
d.zoom.last.sh

DESCRIPTION
d.zoom.last.sh is a UNIX Bourne shell macro that allows the user to interactively adjust the settings of the current geographic region using a pointing device such as a mouse, and automatically saves the last region settings. The last region can be restored by a GRASS command *g.region region=last*
or *g.region last.*

This script and the above restoring command function as an 'undo' in editing.

NOTES
Only the last region is saved. When zooming multiple times, regions before the last are NOT available. Region-file 'last' is reserved for *d.zoom.last.sh.*

FILES
See the file $GISBASE/scripts/d.zoom.last.sh.

SEE ALSO
d.zoom, g.region

AUTHOR
Jianping Xu, John Bognar, Rutgers University

dcorrelate.sh

NAME
dcorrelate.sh - Graphically displays the correlation among from two to four raster map layers in the active frame on the graphics monitor.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
dcorrelate.sh layer1 layer2 [layer3 [layer4]]

DESCRIPTION
dcorrelate.sh is a C-shell (csh(1)) script that graphically displays the results of an *r.stats* run on two raster map layers. This shell script is useful for highlighting the correlation (or lack of it) among data layers.

The results are displayed in the active display frame on the user's graphics monitor. *dcorrelate.sh* erases the active frame before displaying results.

OPTIONS
Parameters:
layer1 layer2 [layer3 [layer4]] The names of from two to four existing raster map layers to be included in the correlation.

NOTES
This is a shell script that uses *r.stats* and the UNIX awk command to calculate the correlation among data layers, and uses *d.text* and *d.graph* to display the results.

If three or four map layers are specified, the correlation among each combination of two data layers is displayed.

This command is written for /bin/csh. If your system doesn't support this shell, don't install this script.

FILES
This program is simply a shell script. Users are encouraged to make their own shell script programs using similar techniques. See $GISBASE/scripts/dcorrelate.sh.

SEE ALSO
d.text, d.graph, r.coin, r.stats
The UNIX awk command

AUTHOR
Michael Shapiro, U.S. Army Construction Engineering Research Laboratory

g.man2html

NAME
g.man2html - convert a GRASS manual page to HTML
(GRASS Shell Script)

GRASS VERSION
4.x

SYNOPSIS
g.man2html
g.man2html help
g.man2html name

DESCRIPTION
g.man2html is a Bourne shell script that converts the roff source of a GRASS manual page to HyperText Markup Language (HTML) and prints the results to standard output.

OPTIONS
Parameter:
name Name of a GRASS man page (full path to roff source)

NOTES
g.man2html is likely to be used by programmers when preparing documentation for their code.

FILES
$GISBASE/scripts/g.man2html

SEE ALSO
g.nroff, g.manual, start.man.sh

AUTHOR
James Darrell McCauley, Agricultural Engineering Purdue University

grass.logo.sh

NAME
grass.logo.sh - Displays a GRASS/Army Corps of Engineers logo in the active display frame on the graphics monitor.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
grass.logo.sh

DESCRIPTION
grass.logo.sh is primarily a demonstration of the GRASS *d.graph* program in a UNIX Bourne shell macro that generates the U.S. Army Corps of Engineers logo. Users are encouraged to generate their own unique logos by writing similar macro shell scripts. Examine the contents of the input file called *grass.logo.sh* located in the GRASS shell script command directory ($GISBASE/scripts) to see how the GRASS logo was generated using *d.graph* graphics commands. The coordinates for this logo were taken from a drawing done on graph paper.

To view the graphics described by this file use *d.graph* or *d.mapgraph*, making sure that a copy of this file is either in your current directory or is given by its full path name.

NOTES
grass.logo.sh, like, *d.rast* will overwrite (not overlay) whatever display appears in the active graphics frame.

This program requires no command line arguments.

SEE ALSO
d.font, d.graph, d.mapgraph, d.rast

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

hsv.rgb.sh

NAME
hsv.rgb.sh - Converts HSV (hue, saturation, and value) cell values to RGB (red, green, and blue) values. (GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
hsv.rgb.sh file1 file2 file3

DESCRIPTION
hsv.rgb.sh is a Bourne shell (sh(1)) program that converts HSV to RGB using *r.mapcalc*. The Foley and Van Dam algorithm is the basis for this program. Input must be three raster files - each file supplying the HSV values. Three new raster files are created representing RGB.

NOTES
Do not use the same names for input and output.

FILES
This program is simply a shell script stored in the file *hsv.rgb.sh* under the $GISBASE/scripts directory. Users are encouraged to make their own shell script programs using similar techniques.

SEE ALSO
rgb.hsv.sh

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

old.cmd.sh

NAME
old.cmd.sh - Provides the new GRASS version 4.2 program name for any program name in GRASS version 3.2.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
old.cmd.sh 3.2_program.name

DESCRIPTION
old.cmd.sh is a Bourne shell (sh(1)) script which, when given a GRASS 3.2 program name, returns the name of the new GRASS version 4.2 program performing its functions. This program is useful as a quick on -line cross-reference between GRASS versions 3 and 4. This program is not interactive; the user must specify the name of a 3.2 program on the command line.

OPTIONS
Parameters:
3.2_program.name The name of an old GRASS version 3.2 program.

Output will be in the form:

old 3.2 program name replaced with: new 4.2 program name

EXAMPLE
For example, to learn which GRASS 4.1 command performs the function of the GRASS 3.2 list command, the user might type:

old.cmd.sh list

The user would then see the following message displayed to standard output:

list replaced with: g.list

FILES
This shell script is stored under the $GISBASE/scripts directory on the user's system. The user is encouraged to examine the shell script commands stored in this directory and to produce similar scripts for their own use.

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

rgb.hsv.sh

NAME
rgb.hsv.sh - Converts RGB (red, green, and blue) cell values to RGB (red, green, and blue) values. (GRASS Shell Script)"

GRASS VERSION
4.x, 5.x

SYNOPSIS
rgb.hsv.sh file1 file2 file3

DESCRIPTION
rgb.hsv.sh is a Bourne shell (sh(1)) program that converts RGB values to HSV using *r.mapcalc*. The Foley and Van Dam algorithm is the basis for the program. Input must be three raster files - each file supplying the RGB values. Three new raster files are created representing HSV.

NOTES
Do not use the same names for input and output.

FILES
This program is simply a shell script stored under the <KBD>$GISBASE/scripts</KBD> directory. The user is encouraged to examine the shell script programs stored here and to produce other such programs.

SEE ALSO
hsv.rgb.sh

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

r.univar

NAME
r.univar - Univariate statistics for a GRASS raster map.
(GRASS Script)

GRASS VERSION
4.x,5.x

SYNOPSIS
r.univar
r.univar help
r.univar name

DESCRIPTION
r.univar calculates univariate statistics of a raster map. This includes the number of cells counted, minimum and maximum cell values, range, arithmetic mean, variance, standard deviation and coefficient of variation.

Parameter:
name Name of an existing raster map.

SEE ALSO
s.univar

AUTHOR
Markus Neteler
neteler@geog.uni-hannover.de

shade.rel.sh

NAME
shade.rel.sh - Creates a shaded relief map based on current resolution settings and sun altitude and azimuth values entered by the user.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
shade.rel.sh

DESCRIPTION
shade.rel.sh is a Bourne shell (sh(1)) script that creates a raster shaded relief map based on current resolution settings and on sun altitude and azimuth values entered by the user. The new shaded relief map is named shade and stored in the user's current mapset. The map is assigned a grey-scale color table.

This program is interactive; the user enters the command:

shade.rel.sh

The program then prompts the user to enter values for:

1) the altitude of the sun in degrees above the horizon (a value between 0 and 90 degrees),

2) the azimuth of the sun in degrees to the east of north (a value between -1 and 360 degrees),

3) the name of a raster map layer whose cell category values are to provide elevation values for the shaded relief map. Typically, this would be a map layer of elevation; however, any raster map layer can be named.

Specifically, *shade.rel.sh* executes the following *r.mapcalc* statement:

```
r.mapcalc; EOF

shade = eval( \\
 x=($elev[-1,-1] + 2*$elev[0,-1] + $elev[1,-1] \\
   -$elev[-1,1] - 2*$elev[0,1] - $elev[1,1])/(8.*$ewres) , \\
 y=($elev[-1,-1] + 2*$elev[-1,0] + $elev[-1,1] \\
   -$elev[1,-1] - 2*$elev[1,0] - $elev[1,1])/(8.*$nsres) , \\
 slope=90.-atan(sqrt(x*x + y*y)), \\
 a=round(atan(x,y)), \\
 aspect=if(x||y,if(a,a,360)), \\
 cang = sin($alt)*sin(slope) + cos($alt)*cos(slope) \\
 * cos($az-aspect), \\
 if(cang &lt; 0,0,100*cang))
EOF
```

Refer to the manual entry for *r.mapcalc* for an explanation of the filtering syntax shown in the above expression. See, for example, the section on "The Neighborhood Modifier".

shade.rel.sh then runs *r.colors* to assign a grey-scale color table to the new shaded relief map shade, by executing the command:

r.colors shade color=grey

FILES
This program is simply a shell script. Users are encouraged to make their own shell scripts using similar techniques. See $GISBASE/scripts/shade.rel.sh.

SEE ALSO
blend.sh, g.ask, g.region, r.colors, r.mapcalc
An Algebra for GIS and Image Processing,
by Michael Shapiro and Jim Westervelt, U.S. Army Construction Engineering Research Laboratory (March/1991).

AUTHOR
Jim Westervelt, U.S. Army Construction Engineering Research Laboratory

show.color.sh

NAME
show.color.sh - Displays and names available primary colors used by GRASS programs, in frames on the graphics monitor.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
show.color.sh

DESCRIPTION
show.color.sh is a UNIX Bourne shell macro that displays and names available primary colors used by GRASS programs in frames on the graphics monitor. Available colors are: red, orange, yellow, green, blue, indigo, violet, white, black, gray, brown, magenta, and aqua.

No program arguments are required to run this program.

NOTES
The full monitor screen is made the active frame after this program ends.

This macro is located in $GISBASE/scripts/show.color.sh.

SEE ALSO
d.colormode, d.colors, d.colortable, d.display, d.frame, grass.logo.sh, show.fonts.sh

AUTHOR
David Gerdes, U.S. Army Construction Engineering Research Laboratory

show.fonts.sh

NAME
show.fonts.sh - Displays and names available font types in the active display frame on the graphics monitor.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
show.fonts.sh

DESCRIPTION
*show.fonts.*sh is a UNIX Bourne shell macro that runs the *d.erase-a* command, then names and displays the font types that can be selected using *d.font*. This macro also runs the GRASS commands *d.font* and *d.text*. See the manual entry for *d.font* for instructions on choosing a font type.

No program arguments are required to run this program.

BUGS
The font is set to romans (Roman simplex) after running *show.fonts.sh.*
There is no mechanism to query the current font, so there is no way to automatically restore the font. The user will have to reset the font type using *d.font* if romans is not desired.

FILES
This program is simply a shell script stored under the $GISBASE/scripts directory. Users are encouraged to examine the shell script programs stored here and to produce others for their own use.

SEE ALSO
d.display, d.erase, d.font, grass.logo.sh, d.label, d.legend, d.paint.labels, d.text, d.title

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

slide.show.sh

NAME
slide.show.sh - Displays a series of raster map layers existing in the user's current mapset search path on the graphics monitor.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
slide.show.sh [across=value] [down=value] [mapsets=list]

DESCRIPTION
slide.show.sh is a UNIX Bourne shell macro which clears the entire screen, creates a series of display frames on the graphics monitor, and displays in slideshow format each of the raster map layers listed in the user-specified mapsets. This is a shell script example that makes extensive use of GRASS and UNIX commands. Users are encouraged to examine this macro and develop similar on-line demos using their own data files.

OPTIONS
Parameters:
across=value The number of display frames across the graphics monitor screen to be used for map display.
Default: 4

down=value The number of display frames down the graphics monitor screen to be used for map display.
Default: 3

mapsets=list The names of the mapsets under the user's current location whose raster map layers are to be displayed, separated by commas.
Default: All mapsets listed in the user's current mapset search path.

FILES
See the file $GISBASE/scripts/slide.show.sh.

SEE ALSO
d.display, d.erase, d.text, g.mapsets, 3d.view.sh, grass.logo.sh, show.fonts.sh

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

split.sh

NAME
split.sh - Divides the graphics monitor into two frames and then displays two maps in these frames. (GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
split.sh mapname mapname [cmd=GRASS_command] [cmd2=GRASS_command] [view=horiz]

DESCRIPTION
split.sh is a Bourne shell (sh) script that clears the entire graphics screen and divides it into two display frames. Map layers are then displayed in each of the two frames. This command is very useful for visually comparing maps (raster, vector, and 3-d views) and can be used by other GRASS shell macros. It is also useful for creating demos.

OPTIONS

Parameters:
view=horiz The graphics screen can be split either horizontally or vertically. The default view splits the screen into two frames, one on the left and one on the right (a vertical split). Some maps (3-d views) are better represented with more width then height (horizontal split). The first map name listed on the command line will be displayed in the top or left window (depending on whether the screen was split horizontally or vertically), and The second map will be displayed in the bottom or right window.

cmd=GRASS_command The GRASS command used to display the named mapnames. If no command is specified by the user, *d.rast* is used by default. However, any GRASS display command (e.g., *d.3d, d.vect*, etc...) an be entered.

cmd2=GRASS_command This command will be used to display map data in the second frame only.

If the user fails to specify the values of both *cmd* and *cmd2, split.sh* will use the default command (*d.rast*) to display user-specified map layer names in both frames. If the user specifies only the value of cmd on the command line, then that command will be executed for both frames. If the user specifies the values of both *cmd* and *cmd2* on the command line, the *cmd* command will be executed in frame 1 and the *cmd2* command will be executed in frame 2.

EXAMPLES

split.sh soils vegcover

split.sh soils cmd2=d.legend "soils red"

split.sh elevation vegcover cmd=d.3d view=horiz

NOTES
split.sh leaves the frame that the last map was drawn in as the active frame. The order in which the options (*cmd, cmd2, view*) are placed on the command line doesn't matter, but the order is important for the map names.

FILES
This program is simply a shell script. Users are encouraged to make their own shell scripts using similar techniques. See $GISBASE/scripts/split.sh.

SEE ALSO
d.3d, d.frame, d.rast, d.sites, d.vect

AUTHOR
Michael Higgins, U.S. Army Construction Engineering Research Laboratory

start.man.sh

NAME
start.man.sh - Creates the template for a manual entry in standard User's Reference Manual format for a user-specified GRASS 4.2 command.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
start.man.sh 4.2_program.name

DESCRIPTION
start.man.sh is a Bourne shell (sh) script which, when given a GRASS 4.2 program name, creates a basic manual entry for that program in the same standard format as that used by the GRASS User's Reference Manual. The named program must already exist under a directory for GRASS main, alpha, or contributed source code.

OPTIONS

This program is not interactive; the user must specify the name of a 4.2 program on the command line.

By default, program output will be sent to standard output (i.e., displayed to the user's text terminal). If the user wishes to save the manual entry created by *start.man.sh*, program output can be redirected into a file. For example, the below command will create a manual entry for the program new program, and save output to the file new.program.man in the user's current directory.

start.man.sh new.program > new.program.man

Parameters:
4.1_program.name The name of an existing GRASS program located in a source code directory for main, alpha, or contributed software.

FILES
This shell script is stored under the $GISBASE/scripts directory on the user's system. The user is encouraged to examine the shell script commands stored in this directory and to produce similar scripts for their own use.

SEE ALSO
bug.report.sh, g.help, g.manual

AUTHOR
James Westervelt, U.S. Army Construction Engineering Research Laboratory

tig.rim.sh

NAME
tig.rim.sh - Generates various vector maps from a rim/TIGER data base.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
tig.rim.sh help
tig.rim.sh dbname tractN1 tractN2 ...

DESCRIPTION
tig.rim.sh is a shell script that queries information from a rim data base using the GRASS command *v.db.rim*, which is an interface between GRASS and RIM.

The dbname given on the command line should be the name of a rim data base created using *v.in.tiger*. *tig.rim.sh* will create several new vector files. Three of these are: a county outline map, a map of tract boundaries within the county, and a map showing block group boundaries. For every tract number given on the input line, additional vector maps will be created showing: the tract outline, a block group boundary map for each block group within the tract, and a map showing block boundaries within each block group. Output files will be named dbname.county, dbname.tract and dbname.bg for the map layers showing the county outline, the tract outlines within the county, and the block group boundaries within the county. The vector file showing the boundary for an individual tract will be named TtractN, where tractN is the tract number given on the command line. The vector files created to show an individual block group boundary and block boundaries within that block group will be named using the appropriate tract number and block group number as part of the name, with suffixes of .bg and .bk respectively.

OPTIONS
Parameters:
dbname Name of an existing rim data base.

tractN Number of a tract located within this county.

NOTES
This command must be installed separately as part of the package of routines dealing with the import of Census (TIGER) data. It requires the use of rim and *v.db.rim*, which must be compiled first.

You must include at least one tract number on the command line for this command to function. Use *itiger.info.sh* to obtain all tract numbers for a given TIGER type1 data file.

If the master binary vector file created using *v.in.tiger* is modified after it is in GRASS, this program will probably not work. In that situation, the processes from this shell script may simply be run by hand, using *v.db.rim* directly, but searching the old vector file to find lines for the new vector files without using the binary offset field (vectoff).

Vector files showing the block boundaries within a block group may contain hydrology lines, which in fact define the edge of a census block.

SEE ALSO
v.in.tig.rim, v.db.rim, tiger.info.sh

AUTHOR

Jim Hinthorne and David Satnik, GIS Lab, Central Washington University, Ellensburg, WA.

tiger.info.sh

NAME
tiger.info.sh - Provides tract number(s) and classification codes found within a given U.S. Census Bureau TIGER type1 data file.
(GRASS Shell Script)

GRASS VERSION
4.x, 5.x

SYNOPSIS
tiger.info.sh help
tiger.info.sh infile

DESCRIPTION
tiger.info.sh is a shell script that outputs tract number(s) and classification codes found within the TIGER type1 data file infile. Output is written to standard out, and can be captured in a file by redirecting output. This information is useful when querying (using *v.db.rim*) from the master binary vector file created by *v.in.tig.rim*. It also provides tract number(s) that can be used as input to the command Gen.Maps.

OPTIONS
Parameters:
infile Name of a TIGER type1 data file.

NOTES
This command must be installed separately as part of the package of routines dealing with the import of Census (TIGER) data.

SEE ALSO
m.tiger.region, v.in.tig.rim, v.db.rim, Gen.Maps , Gen.tractmap

AUTHOR
Marjorie Larson, U.S. Army Construction Engineering Research Laboratory