Package 'Rgnuplot'

July 29, 2015

Type Package

Title R Interface for Gnuplot

Version 1.0.3

Encoding UTF-8

SystemRequirements gnuplot

Description Interface for gnuplot
Based on gnuplot_i version 1.11, the GPL code from Nicolas Devillard.

License GPL (>= 3)

Depends R (>= 2.7.0), methods

Enhances colorspace, png, rgdal, sp

Author Jose Gama [aut, cre],
Nicolas Devillard [aut],
Mauricio Galo [ctb],
Patrick J. Bartlein [ctb],
Oscar Perpiñán Lamigueiro [ctb]

Maintainer Jose Gama <rxprtgama@gmail.com>

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-07-29 00:20:58

R topics documented:

`GpcheckHandle` *Check if the gnuplot session handle is valid*

Description

`GpcheckHandle` checks a gnuplot session handle and returns an error message if it is invalid

Usage

```
GpcheckHandle(handle)
```

Arguments

`handle`	handle to the connection

Value

none

Author(s)

Jose' Gama

See Also

`Gpinit`

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
#returns TRUE for valid handle
## Not run: GpcheckHandle(h1)
#close gnuplot handle
h1<-Gpclose(h1)
#this generates an error
#GpcheckHandle(h1)
```

Gpclose *Close a gnuplot session*

Description

Gpclose closes a gnuplot session and resets the handle to prevent inadvertently trying to reuse and thus avoid a crash

Usage

```
Gpclose(handle)
```

Arguments

handle	handle to the connection

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
```

```
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpcmd *Send one or more commands to gnuplot*

Description

Gpcmd sends one or more commands to an open gnuplot session

Usage

```
Gpcmd(handle, cmd, ...)
```

Arguments

handle	handle to the connection
cmd, ...	cmd is an optional format string as in the printf command, followed by one or more commands

Value

none

Author(s)

Jose' Gama

See Also

Gpinit Gpclose

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
```

```
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot and add a legend
## Not run: GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

`Gpcolorhistogram` *Plot a color histogram*

Description

`Gpcolorhistogram` plots a color histogram

Usage

```
Gpcolorhistogram(fileIN, Ncols=1, vectCols=NULL, gtitle='',gxlabel='',gylabel='')
```

Arguments

`fileIN`	input file
`Ncols`	number of columns to plot
`vectCols`	column indices to use
`gtitle`	title
`gxlabel`	x-axis label
`gylabel`	y-axis label

Value

none

Author(s)

Jose' Gama

Gpcols2rows *convert a file with columns to rows*

Description

Gpcols2rows converts a file with columns yyyy,m1,m2,m3,..., m12 to rows yyyy,m1 yyyy,m3 ... yyyy,m12 readable by gnuplot

Usage

```
Gpcols2rows(filename, newfile, filecolseparator=' ', fileheader = FALSE,
newfilerowseparator='\n\n')
```

Arguments

filename	file with columns yyyy,m1,m2,m3,...,m12
newfile	file with rows yyyy,m1 yyyy,m3 ... yyyy,m12
filecolseparator	string that separates the columns
fileheader	optional column header
newfilerowseparator	string that separates the rows

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
#set gnuplot's additional search directories, to the extdata directory
# from Rgnuplot (default)
Gpsetloadpath(h1)
#change gnuplot's working directory to be the same as
#R's working directory (default)
Gpsetwd(h1)
## Not run:
nordklim <- read.table(system.file('extdata/NordklimData.tab', package =
"Rgnuplot"), stringsAsFactors=FALSE, header=TRUE)
```

```
NKmonths <- c('January','February','March','April','May','June','July','August',
'September','October','November','December')
#choose Helsinki (code 304) and country (code 'FIN') Precipitation (code 601)
nordklimHelsinkiPrecipitation <- nordklim[which((nordklim$NordklimNumber==304) &
(nordklim$CountryCode=='FIN') & (nordklim$ClimateElement==601)),c('FirstYear',
NKmonths)]
nordklimHelsinkiPrecipitation <- as.matrix(nordklimHelsinkiPrecipitation)
#save to a data file
Gpmatrixr2gnu(nordklimHelsinkiPrecipitation, 'NORDKLIM-Helsinki-prec.dat')

#convert the data file from 12 columns for the monthly data to 2 rows
Gpcols2rows('NORDKLIM-Helsinki-prec.dat','NORDKLIM-Helsinki-prec-columns.dat')

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpdivergingColormap *Create diverging colormaps*

Description

GpdivergingColormap

Usage

```
GpdivergingColormap(s,rgb1,rgb2, outColorspace='sRGB')
```

Arguments

s	vector with values between zero and one
rgb1	colorspace element for lower bound
rgb2	colorspace element for upper bound
outColorspace	name of the output color space (RGB, sRGB, HLS, HSV, LAB, LUV, PolarLAB, PolarLUV, XYZ)

Value

none

Author(s)

Kenneth Moreland, Andy Stein and Jose' Gama

See Also

Gprun

Gperrmsg *Get gnuplot's error messages*

Description

Gperrmsg gets gnuplot's error messages

Usage

```
Gperrmsg(handle)
```

Arguments

handle	handle to the connection

Value

string containing gnuplot's error messages

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run: Gpcmd(h1,'plot 0/0')
Gperrmsg(h1)
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpext2terminal *Determine a suitable terminal from a file extension*

Description

Gpext2terminal

Usage

```
Gpext2terminal(filetype='PNG')
```

Arguments

filetype	file extension

Value

terminal name

Author(s)

Jose' Gama

See Also

GpsetTerm

Gpfile2string *Read a text file to a string*

Description

Gpfile2string reads the contents of a text file to a string

Usage

```
Gpfile2string(mfile)
```

Arguments

mfile	file to be read

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - loading and executing a gnuplot script from Rgnuplot
#Initialize the gnuplot handle
h1<-Gpinit()
#set gnuplot's additional search directories, to the extdata directory from Rgnuplot (default)
Gpsetloadpath(h1)
## Not run:
#the filename has Rgnuplot's extdata path
gpfile <- system.file(package='Rgnuplot')
#load script into a string
s<-Gpfile2string(gpfile)
#send gnuplot commands
Gpcmd(h1,s)
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpfitAllprogress *Get the value of all parameters from a stderr fit file*

Description

`GpfitAllprogress` returns the value of all parameters from a logged stderr fit file

Usage

```
GpfitAllprogress(fitprogressfile, boolScreenOut = TRUE)
```

Arguments

`fitprogressfile`	logged stderr fit file
`boolScreenOut`	optional, if TRUE then the output will also be displayed on the screen

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpfitProgress *Get the value of a parameter from a stderr fit file*

Description

GpfitProgress returns the value of a parameter from a logged stderr fit file

Usage

```
GpfitProgress(fitprogressfile, fitparameter)
```

Arguments

fitprogressfile	logged stderr fit file
fitparameter	fit parameter

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Gpgetfontpath *Get gnuplot's additional directories, for fonts*

Description

Gpgetfontpath lists gnuplot's additional directories, for fonts

Usage

```
Gpgetfontpath(handle)
```

Arguments

handle	handle to the connection

Value

String with a list of gnuplot's additional directories, for fonts

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set gnuplot's additional search directories for fonts, to the extdata directory
#from Rgnuplot (default)
Gpsetfontpath(h1)
#get gnuplot's additional search directories for fonts
Gpgetfontpath(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpgetloadpath *Get gnuplot's additional directories, for data and scripts*

Description

Gpgetloadpath gets a list of gnuplot's additional directories, for data and scripts

Usage

```
Gpgetloadpath(handle)
```

Arguments

handle	handle to the connection

Value

String with a list of gnuplot's additional directories, for data and scripts

Author(s)

Jose' Gama

See Also

Gpsetloadpath

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set gnuplot's additional search directories, to the extdata directory from Rgnuplot (default)
Gpsetloadpath(h1)
#get gnuplot's additional search directories
Gpgetloadpath(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpgetvariable *Get the value of a system or environment variable*

Description

Gpgetvariable returns the value of a system or environment variable

Usage

```
Gpgetvariable(handle, variablename)
```

Arguments

handle	handle to the connection
variablename	system or environment variable name

Value

none

Author(s)

Jose' Gama

See Also

Gpsetvariable

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
Gpgetvariable(h1,'pi')

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpgetwd *Get gnuplot's working directory*

Description

Gpgetwd returns gnuplot's working directory

Usage

```
Gpgetwd(handle)
```

Arguments

handle	handle to the connection

Value

String with gnuplot's working directory

Author(s)

Jose' Gama

See Also

Gpsetwd

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#change gnuplot's working directory to be the same as R's working directory (default)
Gpsetwd(h1)
Gpgetwd(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gph *Show gnuplot's help*

Description

Gph shows the output from gnuplot's help command

Usage

```
Gph(handle, gnustring)
```

Arguments

handle	handle to the connection
gnustring	string to be searched for by gnuplot's help system

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
Gph(h1,'plot')

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GphexRGB *RGB sequence, from decimal to hexadecimal, ready for gnuplot*

Description

GphexRGB returns an RGB sequence, from decimal to hexadecimal

Usage

```
GphexRGB(r,g=NA,b=NA)
```

Arguments

r	value for red channel
g	value for green channel
b	value for blue channel

Value

hexadecimal RGB sequence

Author(s)

Jose' Gama

See Also

GpimageRGBchange

GpimageCrop *Simple image manipulation*

Description

GpimageCrop crops an image GpimageDecimate decimates an image GpimageFlip flips a png image on the X, Y or XY axis GpimagePlot displays an image file on the screen or to a chosen terminal except to a file GpimageResize resizes an image GpimageRotate rotates an image GpimageTile tiles multiple image files together into one Gpimage2PNG converts from any supported format to PNG, the file type is automatically detected Gpimage2image converts from any supported format to other format, the file type is automatically detected

Usage

```
GpimageCrop(fileIN, fileOUT,x1,y1,x2,y2, filetype='PNG', terminal=NULL)
GpimageDecimate(fileIN, fileOUT, Xdec=2,Ydec=2, filetype='PNG', terminal=NULL)
GpimageFlip(fileIN, fileOUT,flipX=TRUE,flipY=FALSE, filetype='PNG',
terminal=NULL)
GpimagePlot(fileIN,alpha=FALSE, backgroundColor='', filetype='',
terminal=NULL)
GpimageResize(fileIN, fileOUT,newWidth, newHeight, filetype='PNG',
terminal=NULL)
GpimageRotate(fileIN, fileOUT,degrees, filetype='PNG', terminal=NULL)
GpimageTile(fileOUT, matrixFilenamesIn, vectorWidths, vectorHeights,
matrixXscale=NULL, matrixYscale=NULL,alpha=FALSE, filetype='PNG', terminal=NULL)
Gpimage2PNG(fileIN, fileOUT, optional256=FALSE,alpha=FALSE, backgroundColor='')
Gpimage2image(fileIN, fileOUT , filetype='PNG', terminal=NULL,alpha=FALSE,
backgroundColor='')
```

Arguments

fileIN	input file name
fileOUT	output file name
x1	x coord corner
y1	y coord corner
x2	x coord opposite corner
y2	y coord opposite corner
filetype	image file type
terminal	terminal name
Xdec	x value to decimate
Ydec	y value to decimate
flipX	boolean, TRUE=flip on the x-axis
flipY	boolean, TRUE=flip on the y-axis
alpha	value for alpha channel
backgroundColor	value for background color
newWidth	new value for the width
newHeight	new value for the Height
degrees	angle of rotation in degrees
matrixFilenamesIn	matrix with the file names to tile
vectorWidths	vector with the widths of the files to tile
vectorHeights	vector with the heights of the files to tile
matrixXscale	matrix with values for scaling the images in the x-axis
matrixYscale	matrix with values for scaling the images in the y-axis
optional256	boolean, TRUE=256 color output

Value

none

Author(s)

Jose' Gama

See Also

GpimageRGBchange

GpimageRGBchange *Change image colors*

Description

GpimageRGBchange modifies an image with either a formula for each color component of RGB or for all GpimageRgbfalsecolor converts an image from true color to false color and vice-versa GpimageRgbfiltercolorBlue filters an image using the blue channel GpimageRgbfiltercolorGreen filters an image using the green channel GpimageRgbfiltercolorRed filters an image using the red channel GpimageRgbfiltercolorSepia filters an image using a sepia algorithm GpimageRgbfiltercolorSepia2 filters an image using a 2nd sepia algorithm GpimageRgbgreyscaleBT709 filters an image using greyscale algorithm - BT709 GpimageRgbgreyscaleLinear filters an image using greyscale algorithm - linear GpimageRgbgreyscaleLuminosity filters an image using greyscale algorithm - luminosity GpimageRgbgreyscaleRMY filters an image using greyscale algorithm - RMY GpimageRgbgreyscaleY filters an image using greyscale algorithm - Y GpimageRgbgreyscaleavg filters an image using greyscale algorithm - average

Usage

```
GpimageRGBchange(fileIN, fileOUT, filetype='PNG', terminal=NULL,
rgbformula,rgbformulaG='',rgbformulaB='')
GpimageRgbfalsecolor(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbfiltercolorBlue(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbfiltercolorGreen(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbfiltercolorRed(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbfiltercolorSepia(fileIN, fileOUT, filetype='PNG', terminal=NULL,
sepiaDepth=20,sepiaIntensity=10)
GpimageRgbfiltercolorSepia2(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleBT709(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleLinear(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleLuminosity(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleRMY(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleY(fileIN, fileOUT, filetype='PNG', terminal=NULL)
GpimageRgbgreyscaleavg(fileIN, fileOUT, filetype='PNG', terminal=NULL)
```

Arguments

fileIN	input file name
fileOUT	output file name
filetype	image file type
terminal	terminal name
rgbformula	formula for the 3 RGB channels or for channel R
rgbformulaG	formula for for channel G
rgbformulaB	formula for for channel B
sepiaDepth	depth value for the sepia algorithm
sepiaIntensity	intensity value for the sepia algorithm

Value

none

Author(s)

Jose' Gama

See Also

GpimagePlot

Gpinit *Open a gnuplot session*

Description

Gpinit opens a gnuplot session and returns a handle for using the session

Usage

```
Gpinit(optcmd = "gnuplot")
```

Arguments

optcmd	optional string for opening a gnuplot session

Value

returns a handle to the connection

Author(s)

Jose' Gama

See Also

Gpclose Gpcmd

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpinitSaveStderr *Initialize gnuplot, save stderr to a log file*

Description

GpinitSaveStderr initializes gnuplot and saves stderr to a log file

Usage

```
GpinitSaveStderr(logfile)
```

Arguments

logfile	log file name

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpisWindowOpen *Check if an X11 window is opened*

Description

GpisWindowOpen returns TRUE if an X11 window is opened

Usage

```
GpisWindowOpen(windowid)
```

Arguments

windowid	X11 window id

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpkillpidX11 *Kill X11 windows*

Description

GpkillpidX11 kills X11 windows (gnuplot), useful for testing purposes only- experimental, use at your own risk!

Usage

```
GpkillpidX11(windowid=0)
```

Arguments

windowid	X11 window id

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GploadDemo *Load .dem gnuplot files*

Description

GploadDemo loads a .dem gnuplot file and executes it, allowing pause statements

Usage

```
GploadDemo(handle, mfile)
```

Arguments

handle	handle to the connection
mfile	.dem gnuplot file name

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#example of using GploadDemo and GpreadURL2string
#Initialize the gnuplot handle
h1<-Gpinit()
#change gnuplot's working directory to be the same as
# R's working directory (default)
Gpsetwd(h1)
#load the file 'simple.dem'
#Gpcmd(h1, 'set terminal postscript eps color;set output "simple.eps"\n'
# %s% GpURL2string('http://gnuplot.sourceforge.net/demo_svg/simple.1.gnu') %s%
#'\nset terminal X11;set output')
## Not run:
```

```
if (!file.exists('/usr/share/doc/gnuplot-doc/examples/simple.dem'))
stop('Please install gnuplot-doc')
GploadDemo(h1, '/usr/share/doc/gnuplot-doc/examples/simple.dem')
#pause R and gnuplot
Gppause()
# example of GpreadURL2string
#Kuen's Surface
gpcode<-GpURL2string('http://gnuplot.sourceforge.net/demo/transparent_solids.2.gnu')
#send gnuplot script
Gpcmd(h1, gpcode)
#Gpcmd(h1, 'set terminal postscript eps color
#set output "KuensSurface.eps"\n'
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpmandel *Recursive implementation of the Mandelbrot set function*

Description

Gpmandel returns the values calculated from a recursive implementation of the Mandelbrot set function

Usage

```
Gpmandel(x,y,z,n,maxiterations)
```

Arguments

x	x-coordinate
y	y-coordinate
z	z-coordinate
n	iteration number
maxiterations	maximum number of iterations

Value

none

Author(s)

Kawano and Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
#set gnuplot's additional search directories, to the extdata directory from Rgnuplot (default)
Gpsetloadpath(h1)
#change gnuplot's working directory to be the same as R's working directory (default)
Gpsetwd(h1)
## Not run:
#create the fractal data from R calling a C function, with more points and more iterations
mandelxy2<-function(x,y) Gpmandel(x,y,c(0,0),0,1000)
GpR2splot(mandelxy2,'mandel4.dat',c(-1.5,0.5),c(-1,1),c(500,500),c(300,300),TRUE)
Gpcmd(h1,'reset
splot "mandel4.dat" w  pm3d notitle')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handles
h1<-Gpclose(h1)
```

GpmapMerpar *Work with vector files for plotting maps*

Description

GpmapMerpar creates a data file with meridians or parallels Gpmapsr2gnu saves the lines and points from a maps object to a data file readable by gnuplot

Usage

```
GpmapMerpar(parfile='worldpar.dat',merfile='worldmer.dat',pardeg=10,
merdeg=10,splot=FALSE)
Gpmapsr2gnu(mapsObj,mapFileName)
```

Arguments

parfile	output file with parallels
merfile	output file with meridians
pardeg	resolution is degrees for parallels
merdeg	resolution is degrees for meridians
splot	boolean, TRUE=output for splot
mapsObj	object of type maps
mapFileName	output data file name

Value

none

Author(s)

Jose' Gama

See Also

GpresampleDEM

`Gpmath3dPlot` *3d plots like Mathematica*

Description

`Gpmath3dPlot`

Usage

```
Gpmath3dPlot(foo, xrange=c(0,1), yrange=c(0,1))
```

Arguments

`foo`	mathematical expression
`xrange`	x-axis range
`yrange`	y-axis range

Value

none

Author(s)

Ben Ruijl and Jose' Gama

See Also

Gpcolorhistogram

Gpmatrix2GimpPalette *Work with palette files*

Description

Gpmatrix2GimpPalette saves a matrix into a Gimp palette file (gpl) Gpmatrix2palette save a palette from matrix data with optional palette indeces and the option to modify the palette for solid colors GpcreateIndexFromMatrixAndPalette from a matrix with RGB colors (decimal 24bit) from an image file and its palette as a vector, create a matrix with indices GpcreatePaletteFromMatrix from a matrix with RGB colors (decimal 24bit) from an image file create a palette of 256 colors (decimal 24bit), as a vector GpgimpPalette2matrix reads a Gimp palette into a matrix, optionally the index can be returned too GppalettePlot plots a palette from an indexed PNG file GpshowPaletteColornames get gnuplot's RGB color names as a dataframe - 'ColorName','ColorHex','R','G','B'

Usage

```
Gpmatrix2GimpPalette(paletteMatrix, gplFileName, GimpColumns=16)

Gpmatrix2palette(paletteData, paletteFileName,paletteIndeces=0,SolidColor=FALSE)

GpcreateIndexFromMatrixAndPalette(matrixRGB, paletteRGB)

GpcreatePaletteFromMatrix(matrixRGB, sortType='')

GpgimpPalette2matrix(paletteGimp,returnIndex=FALSE)

GppalettePlot(filepal, sortType='', TheGimp=FALSE)

GpshowPaletteColornames()
```

Arguments

paletteMatrix	matrix with palette values
gplFileName	palette file from The GIMP
GimpColumns	number of columns on gpl file header
paletteData	matrix with the palette data
paletteFileName	output palette file
paletteIndeces	number of indices
SolidColor	boolean, TRUE=solid colors
matrixRGB	matrix with RGB true colors
paletteRGB	palette file
paletteGimp	palette file from The GIMP
returnIndex	boolean, TRUE=include the index value

filepal	palette file
sortType	not implemented yet
TheGimp	boolean, TRUE=GIMP file

Value

none

Author(s)

Jose' Gama

See Also

GpresampleDEM

Gpmatrix2XYdata *Converting a file to X, Y format*

Description

Gpmatrix2XYdata converts a file with data in matrix format to an X, Y format readable by gnuplot

Usage

```
Gpmatrix2XYdata(fileName1,fileName2,optMatrix=NA,surfacegrid=FALSE,overwrite=TRUE)
```

Arguments

fileName1	file with data in matrix format
fileName2	file to save the data in X, Y format
optMatrix	optional matrix to add to the output file's 4th data column
surfacegrid	if TRUE then the data is in grid format
overwrite	if set to TRUE then it will overwrite an existing file without warning

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpmatrixfilePad *Working with matrix and XY data files*

Description

GpmatrixfilePad adds 1 row and 1 column (by duplicating the last row and last column) to a matrix file GpsaveXYfile saves an R matrix with coords as a XY data file Gpxydata2matrix converts a file with splot data to a gnuplot matrix file GpXYcoordsConvertFun reads a XY data file and changes the coords according to 2 functions GpXYcoords2shpere reads a XY data file and changes the coords to fit a sphere

Usage

```
GpmatrixfilePad(fileIN,fileOUT,overwrite=TRUE)
GpsaveXYfile(Rmatrix,xyfile)
Gpxydata2matrix(fileIN,fileOUT)
GpXYcoordsConvertFun(xyfile,newxyfile,fun1,fun2,swapXY=FALSE)
GpXYcoords2shpere(xyfile,newxyfile)
```

Arguments

fileIN	input file name
fileOUT	output file name
overwrite	boolean, TRUE=overwrite the output file
Rmatrix	R matrix
xyfile	input XY file name
newxyfile	output XY file name
fun1	function for the x values
fun2	function for the y values
swapXY	boolean, TRUE=swap x and y

Value

none

Author(s)

Jose' Gama

See Also

Gpmatrix2GimpPalette

Gpmatrixr2gnu *Save R matrix in gnuplot format*

Description

Gpmatrixr2gnu saves an R matrix in a format that can be read by gnuplot

Usage

```
Gpmatrixr2gnu(rmatrix, gnufile)
```

Arguments

rmatrix	R matrix
gnufile	gnuplot file name

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
#set gnuplot's additional search directories, to the extdata directory from
#Rgnuplot (default)
Gpsetloadpath(h1)
#change gnuplot's working directory to be the same as R's working directory
#(default)
Gpsetwd(h1)
## Not run:
nordklim <- read.table(system.file('extdata/NordklimData.tab', package =
"Rgnuplot"), stringsAsFactors=FALSE, header=TRUE)
NKmonths <- c('January','February','March','April','May','June','July',
'August','September','October','November','December')
#choose Helsinki (code 304) and country (code 'FIN') Precipitation (code 601)
nordklimHelsinkiPrecipitation <- nordklim[which((nordklim$NordklimNumber==304) &
(nordklim$CountryCode=='FIN') & (nordklim$ClimateElement==601)),c('FirstYear',
NKmonths)]
nordklimHelsinkiPrecipitation <- as.matrix(nordklimHelsinkiPrecipitation)
#save to a data file
```

```
Gpmatrixr2gnu(nordklimHelsinkiPrecipitation, 'NORDKLIM-Helsinki-prec.dat')

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gppausableterm *Determine if the current terminal can be paused in synch with R*

Description

Gppausableterm returns TRUE if the current terminal can be paused

Usage

```
Gppausableterm()
```

Value

boolean

Author(s)

Jose' Gama

See Also

Gppause

Gppause *Pauses the system*

Description

Gppause pauses the system for a number of seconds and then waits for the user to press a key, detecs X11 beforehand

Usage

```
Gppause(delaySecs=2)
```

Arguments

delaySecs	number of seconds to wait

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GppauseX *Pauses the system (X11)*

Description

GppauseX pauses the system for a number of seconds and then waits for the user to press a key, X11 only

Usage

```
GppauseX(delaySecs=2)
```

Arguments

delaySecs	number of seconds to wait

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GppidX11 *Get the pid from an X11 window*

Description

GppidX11 returns the pid from an X11 window (gnuplot)

Usage

```
GppidX11(windowid=0)
```

Arguments

windowid	window id (X11)

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpplotEquation *Plot the curve of an equation in 2D*

Description

GpplotEquation plots an expression, data or a function in 2D

Usage

```
GpplotEquation(handle, equation, title)
```

Arguments

handle	handle to the connection
equation	equation to be plotted
title	title for the graphic

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpplotFunction *Plots an expression in one function call*

Description

GpplotFunction plots an expression without the need to define handles

Usage

```
GpplotFunction(x, xlab='x', ylab='y', main='', type='l',...)
```

Arguments

x	expression
xlab	label for the x-axis
ylab	label for the y-axis
main	main title for the graphic
type	type of plot: lines or points
...	extra parameters passed to the function

Value

none

Author(s)

Oscar Perpi\~n\'an Lamigueiro

See Also

Gpinit

Examples

```
## Not run: GpplotFunction ('sin(x)', 'x', 'y', 'sine function')
```

GpplotOnce *Plots arrays in one function call*

Description

GpplotOnce plots arrays for the x and y values without the need to define handles

Usage

```
GpplotOnce(title, style, label.x, label.y, x, y, n)
```

Arguments

title	title for the graphic
style	gnuplot style (lines, points, linespoints, ...)
label.x	label for the x-axis
label.y	label for the y-axis
x	array of doubles for the x-axis
y	array of doubles for the y-axis
n	number of elements of the list

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpplotPolyFit *Plots a polynomial fit in one function call*

Description

GpplotPolyFit Plots a polynomial fit without the need to define handles

Usage

```
GpplotPolyFit(x, y, order)
```

Arguments

x	array with the values of x
y	array with the values of y = f(x)
order	order of the polynomial

Value

none

Author(s)

Oscar Perpi\~n\'an Lamigueiro

See Also

Gpinit

Examples

```
polnorder <- 7# order of the polynomial
npoints <- 20 # number of points to plot
xpoints <- ( 0:npoints ) * 0.1 # x values
wpoints <- c(1,10^ -( 0:polnorder )) # "a" to "h" values
xPower <- outer(xpoints, 0:7, '^')
ypoints <- colSums(wpoints[1:8] * t(xPower))
## Not run: GpplotPolyFit(xpoints, ypoints, 7)
```

GpplotSlope *Plots a slope*

Description

GpplotSlope plots a slope from a list of doubles

Usage

```
GpplotSlope(handle, a, b, title)
```

Arguments

handle	handle to the connection
a	list of doubles
b	number of elements of the list
title	title for the graphic

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpplotX *Plot one list of doubles in 2D*

Description

GpplotX plots one list of doubles in 2D

Usage

```
GpplotX(handle, d, n, title)
```

Arguments

handle	handle to the connection
d	list of doubles
n	number of elements of the list
title	title for the graphic

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpplotXY *Plot two list of doubles in 2D*

Description

GpplotXY plots two list of doubles in 2D

Usage

```
GpplotXY(handle, x, y, n, title)
```

Arguments

handle	handle to the connection
x	list of doubles for the x-axis
y	list of doubles for the y-axis
n	number of elements of the list
title	title for the graphic

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpPNG2color *Convert PNG to text*

Description

GpPNG2color converts a PNG file to a text format readable by gnuplot

Usage

```
GpPNG2color(fileName)
```

Arguments

fileName	PNG file name

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpPNG2RGB *Convert PNG to RGB*

Description

GpPNG2RGB converts a PNG file to an RGB or RGBA file

Usage

```
GpPNG2RGB(PNGfile, RGBfile,forceRGB=FALSE)
```

Arguments

PNGfile	PNG file
RGBfile	RGB or RGBA file
forceRGB	if forceRGB is TRUE then the alpha channel is ignored

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpPNG4DEM *Convert PNG plus DEM to text*

Description

GpPNG4DEM converts a PNG file and a DEM tab separated to a text format readable by gnuplot

Usage

```
GpPNG4DEM(filePNG, fileDEMtab, file3Ddat)
```

Arguments

filePNG	PNG file
fileDEMtab	DEM tab separated file
file3Ddat	text file (gnuplot)

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpR2plot *Save 2D R to gnuplot*

Description

GpR2plot saves the output from a 2D R function to a file readable by gnuplot

Usage

```
GpR2plot(Rfunction,filename,gnuxrange,gnuyrange,gnusamples)
```

Arguments

Rfunction	2D R function
filename	file name (gnuplot)
gnuxrange	gnuplot xrange
gnuyrange	gnuplot yrange
gnusamples	gnuplot samples

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpR2splot *Save 3D R to gnuplot*

Description

GpR2splot saves the output from a 3D R function to a file readable by gnuplot

Usage

```
GpR2splot(Rfunction,filename,gnuxrange,gnuyrange,gnusamples,gnuisosamples,hidden3d=FALSE)
```

Arguments

Rfunction	2D R function
filename	file name (gnuplot)
gnuxrange	gnuplot xrange
gnuyrange	gnuplot yrange
gnusamples	gnuplot samples
gnuisosamples	gnuplot gnuisosamples
hidden3d	gnuplot hidden3d

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Gpresetplot *Reset the session*

Description

Gpresetplot resets the session

Usage

```
Gpresetplot(handle)
```

Arguments

handle	handle to the connection

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpRGB2image *Work with RGB data*

Description

GpRGB2image converts an RGB file to another image file GpRGB1to3channels converts a vector or matrix of RGB 1-channel decimal values (24 bit) to 3-channel decimal values (3x8 bit) GpRGB2DAT converts an RGB file to a matrix data file GpRGBsample plots a square filled with an RGB color

Usage

```
GpRGB2image(RGBfile, imagefile, width, height, filetype='PNG', gpterminal=NULL)
GpRGB1to3channels(RGB1channel=NULL, fileRGB1channel=NULL, fileRGB3channel=NULL)
GpRGB2DAT(RGBfile, DATfile,width,height)
GpRGBsample(xRGB='#FFFFFF', optionalTitle='')
```

Arguments

RGBfile	raw RGB file name
imagefile	image file name
width	image width
height	image height
filetype	image type
gpterminal	terminal name
RGB1channel	vector or matrix of RGB 1-channel decimal values (24 bit)
fileRGB1channel	file with RGB 1-channel decimal values (24 bit)
fileRGB3channel	file with RGB 3-channel decimal values (3x8 bit)
DATfile	data file name
xRGB	RGB color in hexadecimal
optionalTitle	title for plot

Value

none

Author(s)

Jose' Gama

See Also

Gpimage2image

Gprun *executes gnuplot code directly*

Description

Gprun returns the gnuplot version

Usage

```
Gprun(cmd, optPause=FALSE)
```

Arguments

cmd	String with gnuplot commands
optPause	boolean, TRUE means to pause after the execution

Value

none

Author(s)

Jose' Gama

See Also

Gppause

`Gpsetfontpath` *Set gnuplot's additional directories, for fonts*

Description

`Gpsetfontpath` set gnuplot's additional directories, for fonts, default path = extdata directory from Rgnuplot

Usage

```
Gpsetfontpath(handle,fontpath)
```

Arguments

`handle`	handle to the connection
`fontpath`	string with gnuplot's additional directories, for fonts

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set gnuplot's additional search directories for fonts, to the extdata
#directory from Rgnuplot (default)
Gpsetfontpath(h1)
#get gnuplot's additional search directories for fonts
Gpgetfontpath(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

`Gpsetloadpath` *Set gnuplot's additional directories, for data and scripts*

Description

`Gpsetloadpath` sets gnuplot's additional directories, for data and scripts, default path = extdata directory from Rgnuplot

Usage

```
Gpsetloadpath(handle, loadpath)
```

Arguments

`handle`	handle to the connection
`loadpath`	string with gnuplot's additional directories, for data and scripts

Value

none

Author(s)

Jose' Gama

See Also

`Gpinit`

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set gnuplot's additional search directories, to the extdata directory from Rgnuplot (default)
Gpsetloadpath(h1)
#get gnuplot's additional search directories
Gpgetloadpath(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpsetstyle *Set the plotting style*

Description

`Gpsetstyle` sets the plotting style (lines, points, linespoints, ...)

Usage

```
Gpsetstyle(handle, plot.style)
```

Arguments

`handle`	handle to the connection
`plot.style`	gnuplot style (lines, points, linespoints, ...)

Value

none

Author(s)

Jose' Gama

See Also

`Gpinit`

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpsetTerm *List or change gnuplot's current terminal*

Description

GpsetTerm sets gnuplot's current terminal or get a list of all available terminals

Usage

```
GpsetTerm(optionalNewTerminal)
```

Arguments

optionalNewTerminal
: should be either empty or the name of an available terminal

Value

the name of gnuplot's current terminal

Author(s)

Jose' Gama

See Also

Gpshowterm

Gpsetvariable *Set a system or environment variable*

Description

Gpsetvariable sets a system or environment variable "variablename", with value "variabledata"

Usage

```
Gpsetvariable(handle,variablename,variabledata)
```

Arguments

handle	handle to the connection
variablename	variable name
variabledata	data value

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Gpsetwd *Set gnuplot working directory*

Description

Gpsetwd sets gnuplot working directory, default path = R's working directory

Usage

```
Gpsetwd(handle, wd)
```

Arguments

handle	handle to the connection
wd	gnuplot working directory, default = R's working directory

Value

none

Author(s)

Jose' Gama

See Also

Gpgetwd

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#change gnuplot's working directory to be the same as R's working directory (default)
Gpsetwd(h1)
#check it out
Gpgetwd(h1)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpsetXlabel *Set the x-axis label*

Description

GpsetXlabel sets the x-axis label

Usage

```
GpsetXlabel(handle, label)
```

Arguments

handle	handle to the connection
label	label for the x-axis

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpsetYlabel *Set the y-axis label*

Description

GpsetYlabel sets the y-axis label

Usage

```
GpsetYlabel(handle, label)
```

Arguments

handle	handle to the connection
label	label for the y-axis

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#"Hello World!" - text on legend
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
#set output to a postscript file
#Gpcmd(h1,'set terminal postscript eps color;set output "helloworld1.eps"')
#label the x and y axis
GpsetXlabel(h1, 'x')
GpsetYlabel(h1, 'y')
#set plot style to "lines"
Gpsetstyle(h1, 'lines')
#plot sin(x) and add a legend
GpplotEquation(h1,'sin(x)','Hello World!')
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpshowDatafileBinaryFiletypes

List gnuplot's RGB binary types

Description

GpshowDatafileBinaryFiletypes get gnuplot's RGB binary types as a vector

Usage

```
GpshowDatafileBinaryFiletypes()
```

Value

vector with the names of gnuplot's RGB binary types

Author(s)

Jose' Gama

See Also

GpimagePlot

Gpshowterm *Get gnuplot's current terminal*

Description

Gpshowterm gets gnuplot's current terminal

Usage

```
Gpshowterm()
```

Value

the name of gnuplot's current terminal

Author(s)

Jose' Gama

See Also

GpsetTerm

GpSHP2gnu *Work with shapefiles*

Description

GpSHP2gnu given a shapefile (SHP) with full path and the shapefile layer name, the coordinates are saved to a text file readable by gnuplot

GpresampleDEM resamples DEM data, similarly to what GIS applications do

GpboxXY returns the box (coordinates) around a location, from a shapefile

GpmapPNG2lines draws "squarish" coastlines given a gridded map from a PNG image

Gpmatrix2PNG saves a matrix with a mask of a map to a PNG file

GpplotMap plots a map with vector and raster overlays

Usage

```
GpSHP2gnu(SHPfilename, SHPlayername,gnufilename,toCRS='+init=epsg:4326')
GpresampleDEM(fileIN, fileOUT, xrange, yrange, interpolationMethod,
XYformat=FALSE)
GpboxXY(fileName)
GpmapPNG2lines(PNGfile, landoutlinefile)
Gpmatrix2PNG(matM,PNGfile)
GpplotMap(mapvectfiles=NA, projection='PlateCarree', linetype='l',
linestyle=1, plotTitle=NA, maprastfile=NA, maprastpalette=NA,
AdditionalCode=NA,projectionInit=NA,returnCode=FALSE)
```

Arguments

SHPfilename	shapefile name
SHPlayername	shape layer name
gnufilename	output data file name
toCRS	optional CRS string to modify the projection
fileIN	input data file name
fileOUT	output data file name
xrange	x-axis range
yrange	y-axis range
interpolationMethod	dgrid3d interpolation method
fileName	shapefile name
PNGfile	PNG input file name
landoutlinefile	vector output file name
matM	matrix with map raster data
XYformat	boolean, TRUE=use XY format in input file
mapvectfiles	filenames with vector data
projection	cartographic projection
linetype	line type
linestyle	line style
plotTitle	title for the plot
maprastfile	filename of the raster file
maprastpalette	filename of the raster file's palette
AdditionalCode	additional code to run before the plot
projectionInit	initialization values for the projection
returnCode	boolean, if TRUE the code is returned as a string

Value

none

Author(s)

Pat Bartlein and Jose' Gama

See Also

Gpmapsr2gnu

Gpsplot — *Plots an expression in 3D with one function call*

Description

Gpsplot plots an expression in 3D without the need to define handles

Usage

```
Gpsplot(x, type = c('hidden3d', 'pm3d', 'map', 'contour'))
```

Arguments

x	expression
type	type of 3D plot: hidden3d, pm3d, map or contour

Value

none

Author(s)

Oscar Perpi\~n\'an Lamigueiro

See Also

Gpinit

Examples

```
## Not run: Gpsplot(volcano)
Gpsplot(volcano, 'pm3d')
Gpsplot(volcano, 'map')
Gpsplot(volcano, 'contour')
## End(Not run)
```

GpURL2string — *Read a text file from the web to a string*

Description

GpURL2string reads the contents of a text file from the web to a string

Usage

```
GpURL2string(mURL)
```

Arguments

mURL	URL with the file to be read

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#example of using GploadDemo and GpreadURL2string
#Initialize the gnuplot handle
h1<-Gpinit()
#change gnuplot's working directory to be the same as
#R's working directory (default)
Gpsetwd(h1)
#load the file 'simple.dem'
#Gpcmd(h1, 'set terminal postscript eps color;set output "simple.eps"\n'
# %s% GpURL2string('http://gnuplot.sourceforge.net/demo_svg/simple.1.gnu') %s%
# '\nset terminal X11;set output')
## Not run:
if (!file.exists('/usr/share/doc/gnuplot-doc/examples/simple.dem'))
stop('Please install gnuplot-doc')
GploadDemo(h1, '/usr/share/doc/gnuplot-doc/examples/simple.dem')
#pause R and gnuplot
Gppause()
# example of GpreadURL2string
#Kuen's Surface
gpcode<-GpURL2string('http://gnuplot.sourceforge.net/demo/transparent_solids.2.gnu')
#send gnuplot script
#Gpcmd(h1, 'set terminal postscript eps color
#set output "KuensSurface.eps"\n'
Gpcmd(h1, gpcode)
#pause R and gnuplot
Gppause()
## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

Gpversion *Get the gnuplot version*

Description

Gpversion returns the gnuplot version

Usage

```
Gpversion(handle)
```

Arguments

handle	handle to the connection

Value

gnuplot version

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
# get the version number
## Not run: Gpversion(h1)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpWindowStatus *Status of an X11 window*

Description

GpWindowStatus returns the status of an X11 window

Usage

```
GpWindowStatus(windowid)
```

Arguments

windowid	window id (X11)

Value

none

Author(s)

Joel VanderWerf and Jose' Gama

See Also

Gpinit

Examples

```
#Initialize the gnuplot handle
h1<-Gpinit()
## Not run:
GpWindowStatus(0)
#plot sin(x)
GpplotEquation(h1,'sin(x)','')
GpWindowStatus(0)
#pause R and gnuplot
Gppause()
GpWindowStatus(0)

## End(Not run)
#close gnuplot handle
h1<-Gpclose(h1)
```

GpwriteMultiCsv *Write multi column CSV file*

Description

GpwriteMultiCsv Writes a multi column CSV file for use with gnuplot commands later

Usage

```
GpwriteMultiCsv(fileName,xListPtr,n,numColumns,title)
```

Arguments

fileName	CSV file name
xListPtr	A list of pointers to column buffers
n	number of columns
numColumns	Length of xListPtr lis
title	Title to write for the first line of the .csv file, will be preceeded by "#"

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpwriteXcsv *Write single column CSV file*

Description

GpwriteXcsv writes a single column CSV file for use with gnuplot commands later

Usage

```
GpwriteXcsv(fileName,x,n,title)
```

Arguments

fileName	CSV file name
x	A column buffer
n	number of columns
title	Title to write for the first line of the .csv file, will be preceeded by "#"

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpwriteXYcsv *Write double column CSV file*

Description

GpwriteXYcsv writes a double column CSV file for use with gnuplot commands later

Usage

```
GpwriteXYcsv(fileName,x,y,n,title)
```

Arguments

fileName	CSV file name
x	A column buffer
y	A column buffer
n	number of columns
title	Title to write for the first line of the .csv file, will be preceeded by "#"

Value

none

Author(s)

Jose' Gama

See Also

Gpinit

GpX11Present *Check for X11 in the system*

Description

GpX11Present returns TRUE if X11 is present in the system

Usage

```
GpX11Present()
```

Value

TRUE if X11 is present in the system

Author(s)

Jose' Gama

See Also

Gpinit

Examples

```
## Not run: GpX11Present()
```

---

sOp *String concatenation operator*

Description

%s% concatenates 2 strings

Usage

```
x %s% y
```

Arguments

x	string
y	another string

Value

none

Author(s)

Jose' Gama

Examples

```
# create two strings
## Not run:
t1 <- 'Hello'
t2 <- 'World'
# concatenate several strings with %s%
t3 <- t1 %s% ' ' %s% t2
print(t3)
print(t3 %s% '!')

## End(Not run)
```

Index