GP Tools

for **GAUSSplot**™

Aptech Systems, Inc.

Powered by Tecplot®

Documentation Revision: 444 April 1, 2005

Contents

1 Installation **1**

1.1 UNIX/Linux 1

1.1.1 Download 1

1.1.2 CD 2

1.1.3 Floppy 2

1.2 Windows/NT/2000 3

1.2.1 Download 3

1.2.2 CD 3

1.2.3 Floppy 3

1.3 Differences Between the UNIX and Windows/NT/2000 Versions 4

2 Introduction **5**

3 Command Reference **7**

gptKernelDensity 8

gptKernelDensityControlCreate 11

gptXY 12

Index **15**

Chapter 1

Installation

1.1 UNIX/Linux

If you are unfamiliar with UNIX/Linux, see your system administrator or system documentation for information on the system commands referred to below.

1.1.1 Download

1. Copy the `.tar.gz` or `.zip` file to `/tmp`.
2. If the file has a `.tar.gz` extension, unzip it using `gunzip`. Otherwise skip to step 3.

   `gunzip` *appxxx*`.tar.gz`
3. `cd` to your **GAUSS** or **GAUSS Engine** installation directory. We are assuming `/usr/local/gauss` in this case.

   `cd /usr/local/gauss`
4. Use `tar` or `unzip`, depending on the file name extension, to extract the file.

   `tar xvf /tmp/`*appxxx*`.tar`

   – or –

   `unzip /tmp/`*appxxx*`.zip`

1.1.2 CD

1. Insert the Apps CD into your machine's CD-ROM drive.

2. Open a terminal window.

3. `cd` to your current **GAUSS** or **GAUSS Engine** installation directory. We are assuming `/usr/local/gauss` in this case.

   ```
   cd /usr/local/gauss
   ```

4. Use `tar` or `unzip`, depending on the file name extensions, to extract the files found on the CD. For example:

   ```
   tar xvf /cdrom/apps/app_myapps_1.0_unix.tar
   ```
   – or –
   ```
   unzip /cdrom/apps/app_myapps_1.0_unix.zip
   ```
   However, note that the paths may be different on your machine.

Documentation for the application(s) can be found in the `apps/MANUALS` subdirectory of the CD.

1.1.3 Floppy

1. Make a temporary directory.

   ```
   mkdir /tmp/workdir
   ```

2. `cd` to the temporary directory.

   ```
   cd /tmp/workdir
   ```

3. Use `tar` to extract the files.

   `tar xvf` *device_name*

   If this software came on diskettes, repeat the `tar` command for each diskette.

4. Read the README file.

   ```
   more README
   ```

5. Run the `install.sh` script in the work directory.

   ```
   ./install.sh
   ```

   The directory the files are installed to should be the same as the install directory of **GAUSS** or the **GAUSS Engine**.

6. Remove the temporary directory (optional).

1.2 Windows/NT/2000

1.2.1 Download

Unzip the `.zip` file into your **GAUSS** or **GAUSS Engine** installation directory.

1.2.2 CD

1. Insert the Apps CD into your machine's CD-ROM drive.
2. Unzip the `.zip` files found on the CD to your **GAUSS** or **GAUSS Engine** installation directory, using your current `.zip` file extraction utility.

Documentation for the application(s) can be found in the `MANUALS` subdirectory of the CD.

1.2.3 Floppy

1. Place the diskette in a floppy drive.
2. Call up a DOS window
3. In the DOS window log onto the root directory of the diskette drive. For example:

```
A:<enter>
cd\<enter>
```

4. Type: **ginstall** *source_drive target_path*

source_drive	Drive containing files to install with colon included For example: **A:**
target_path	Main drive and subdirectory to install to without a final \ For example: **C:\GAUSS**

A directory structure will be created if it does not already exist and the files will be copied over.

*target_path***src**	source code files
*target_path***lib**	library files
*target_path***examples**	example files

1.3 Differences Between the UNIX and Windows/NT/2000 Versions

- If the functions can be controlled during execution by entering keystrokes from the keyboard, it may be necessary to press *Enter* after the keystroke in the UNIX version.
- On the Intel math coprocessors used by the Windows/NT/2000 machines, intermediate calculations have 80-bit precision, while on the current UNIX machines, all calculations are in 64-bit precision. For this reason, **GAUSS** programs executed under UNIX may produce slightly different results, due to differences in roundoff, from those executed under Windows/NT/2000.

Chapter 2

Introduction

GP TOOLS contains additional procedures that may be used with **GAUSSplot**. It includes quick and dirty procedures for producing simple graphs as well as procedures that make use of **GAUSSplot**'s graphing capabilities in a specific application.

To use the procedures in *GP TOOLS*, you must include both `gaussplot` and `gptools` in a **library** statement at the top of your program:

```
library gaussplot,gptools;
```

You may also need to include a file with structure definitions when calling certain procedures. See the reference pages for more information.

Chapter 3

Command Reference

Purpose

Computes and plots kernel density estimates

Library

`gptools`

Format

out = **gptKernelDensity(***dataset*, *ctl***);**

Input

dataset string, name of **GAUSS** dataset.

ctl instance of gptKernelDensityControl structure.

ctl.`VarNames` K × 1 string array, names of selected columns

ctl.`Kernel` K × 1 vector, type of kernel:

1 normal kernel
2 Epanechnikov kernel
3 biweight kernel
4 triangular kernel
5 rectangular kernel
6 truncated normal kernel

If ctl.`Kernel` is scalar, the kernel is the same for all parameters. Default = 1;

ctl.`NumPoints` scalar, number of points to be computed for plots

ctl.`EndPoints` K × 2 matrix, lower (in first column) and upper (in second column) endpoints of density. Default is minimum and maximum, respectively, of the parameter values. If 1 × 2 matrix, endpoints will be the same for all parameters.

ctl.`Smoothing` K × 1 vector or N × 1 vector or N × K matrix, smoothing coefficients for each plot. If scalar, smoothing coefficient will be the same for each plot. If zero, smoothing coefficient will be computed by CMLdensity. If matrix, smoothing coefficient will be different for each observation. Default = 0;

ctl.`Truncate` K × 2 matrix, lower (in first column) and upper (in second column) truncation limits for truncated normal kernel. If 1 × 2 matrix, truncations limits will be the same for all plots. Default is minimum and maximum, respectively.

Output

out instance of gptKernelDensityResults structure

out.px `ctl.NumPoints` × 1 vector, abscissae.

out.py `ctl.NumPoints` × 1 vector, ordinates.

out.sm K × 1, or N × k, or N × 1 smoothing coefficients.

Remarks

You need to include the file `gptkern.sdf` in your program in order to use this command.

Adapted from a procedure written by Gary King, Department of Government, Harvard University

Reference: B.W. Silverman. 1986. `Density Estimation for Statistics and Data Analysis`. London: Chapman and Hall.

An Application: Gary King. "Constituency Service and Incumbency Advantage," `British Journal of Political Science`, 21, 1 (January, 1991): 119–128.

Example

This example computes and plots a kernel density estimation of a simulated mixed Normal, and computes the modes.

```
library gaussplot, gptools;
#include gptkern.sdf

rndseed 34563456;
x = rndn(500,1) | (1.2*(rndn(500,1) + 5));
call saved(x,"sim","X");

struct gptKernelDensityControl k0;
k0.NumPoints = 100;

struct gptKernelDensityResults out;
out = gptKernelDensity("sim",k0);

dd = trimr(out.py,1,0) - trimr(out.py,0,1);
modes = {};

d0 = dd[1]/abs(dd[1]);  // d0 should be positive, if not
                        // there's a problem
```

```
for i(2,rows(dd),1);
   d1 = dd[i]/abs(dd[i]);
   if d1 /= d0;
       if d0 == 1;
           m = dd[i-1] / (dd[i-1] - dd[i]);
           modes = modes | out.px[i] + m*abs(out.px[i+1]-out.px[i]);
       endif;
       d0 = d1;
   endif;
endfor;

print "modes " modes;
```

Source

gptkern.src

Purpose

Creates a default instance of type **gptKernelDensityControl**.

Library

`gptools`

Format

s = **gptKernelDensityControlCreate;**

Output

s	instance of type **gptKernelDensityControl**.

Remarks

You need to include the file `gptkern.sdf` in your program in order to use this command.

Source

`gptKern.src`

Purpose

Shortcut command that draws a graph containing an XY Line plot with one frame.

Library

gptools

Format

ret = **gptXY**(*x*, *y*);

Input

x	$N \times 1$ vector or $N \times K$ matrix, the variable(s) to be associated with the X-axis.
y	$N \times 1$ vector or $N \times K$ matrix, the variable(s) to be associated with the Y-axis.

Output

ret	scalar, return code, 0 if successful, otherwise an error code.

Remarks

This command is a convenience function that allows you to create an XY Line plot with only the following two lines of code:

```
library gaussplot,gptools;
call gptXY(x,y);
```

assuming that the variables *x* and *y* already exist in the **GAUSS** workspace.

gptXY is NOT thread-safe. It should be used only as a means of quickly visualizing data. Use **gpMakeXYLinePlot** or **gpPlot** to create thread-safe **GAUSSplot** programs.

An XY Line plot created with **gptXY** is only minimally configurable. The following global variables control display options for **gptXY**:

Global variable	Description
_gpt_layers	N × 1 vector, $1 \leq N \leq 4$, controls which line map layers are displayed, may contain one or more of the following options: **1** Show line layer (default) **2** Show symbol layer **3** Show error bar layer **4** Show bar layer
_gpt_xtitle	string, title for the X-axis. Default = "X".
_gpt_ytitle	string, title for the Y-axis. Default = "Y".

A graph created with **gptXY** will have one frame containing one or more line maps. If both x and y are $N \times 1$ vectors, then only one line map will be created. If they are both $N \times K$ matrices, then K line maps will be created, and each column in x will be plotted against the corresponding column in y. If x is an $N \times 1$ vector and y is an $N \times K$ matrix, or vice versa, then the $N \times 1$ vector will be plotted against each of the columns in the $N \times K$ matrix. To create a more complicated graph with multiple frames, use **gpPlot**.

This command generates temporary data, macro and structure files on the **GAUSS** temporary file path (edit `gauss.cfg` to change this path). It is your responsibility to clean these files out of the temporary directory.

Observations containing missing values will be deleted before the data is written to the data file. If there are missing values, and x and/or y contain multiple columns, then each column will be written to a separate zone in the data file, since variables contained in the same zone must be equal in size. If the variables are written to multiple zones, then only the first two variable names in *vnames* will be used, since variable names are set once for all zones in the data file. See **gpMakeXYLinePlot** for more information.

Example

```
library gaussplot,gptools;

x = seqa(0.2,0.2,20);
y = log(x)~ln(x);
ret = gptXY(x,y);
```

Source

gptools.src

Index

gptKernelDensity, 8
gptKernelDensityControlCreate, 11
gptXY, 12

I

Installation, 1

K

Kernel Density plot, 8

U

UNIX/Linux, 1
UNIX, 4

W

Windows/NT/2000, 3, 4

X

XY Line plot, 12