GREG tutorial: I. Basics

Presentation by
Sébastien BARDEAU
(IRAM/Grenoble)

Current kernel developers: Jérôme PETY,
Sébastien BARDEAU, & Stéphane GUILLOTEAU
on behalf of the GILDAS developers over time

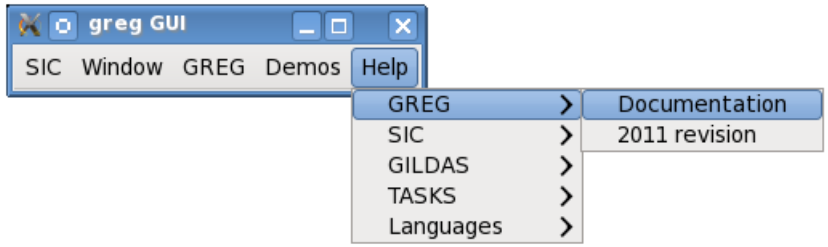
9th IRAM Millimeter Interferometry School
Oct. 10-14 2016, St Martin d'Hères

Table of contents

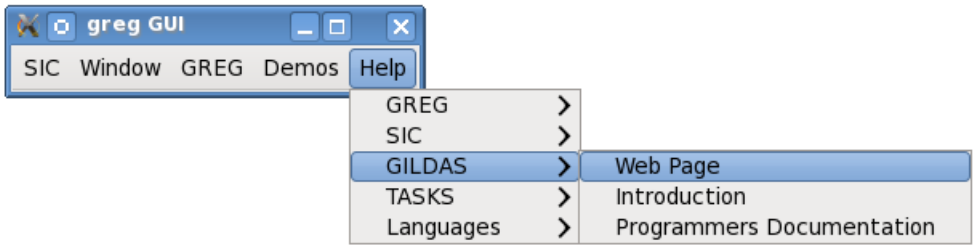
Need help (bis)?

Need help (bis)? I. Browsing the widgets

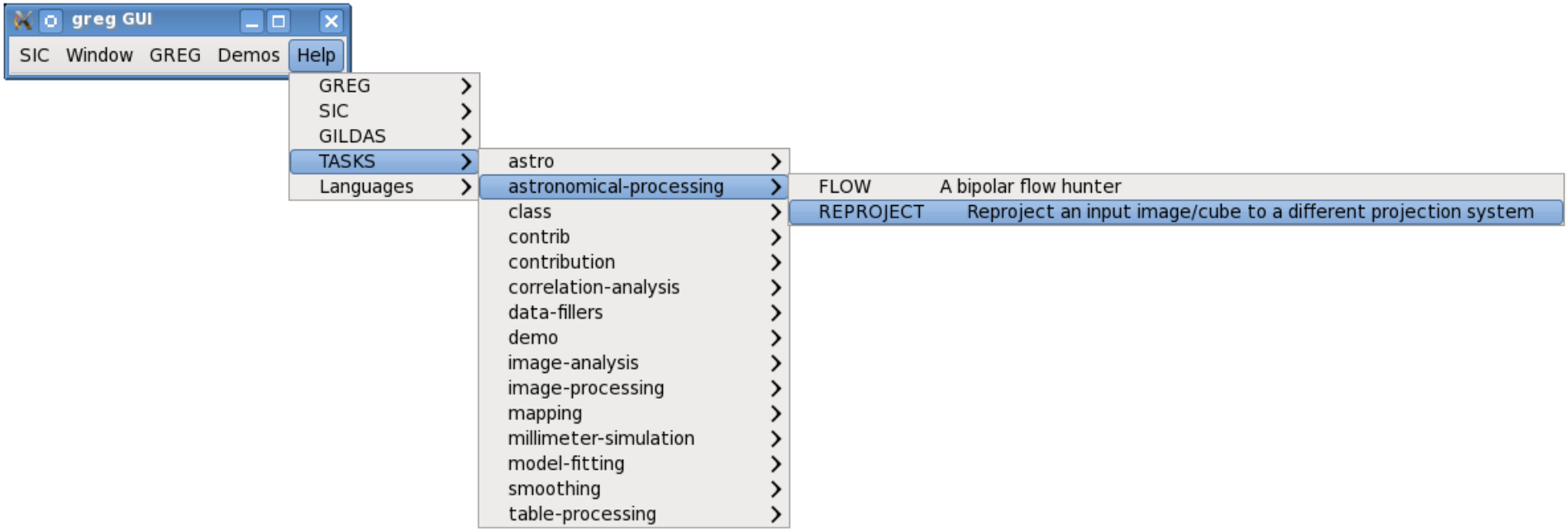
PDF document opened in your PDF viewer:



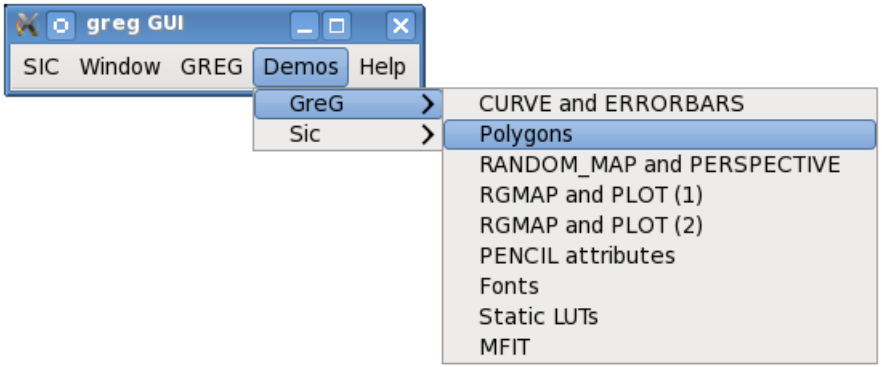
Web pages opened in your web browser:



Online HELP displayed in the terminal window:

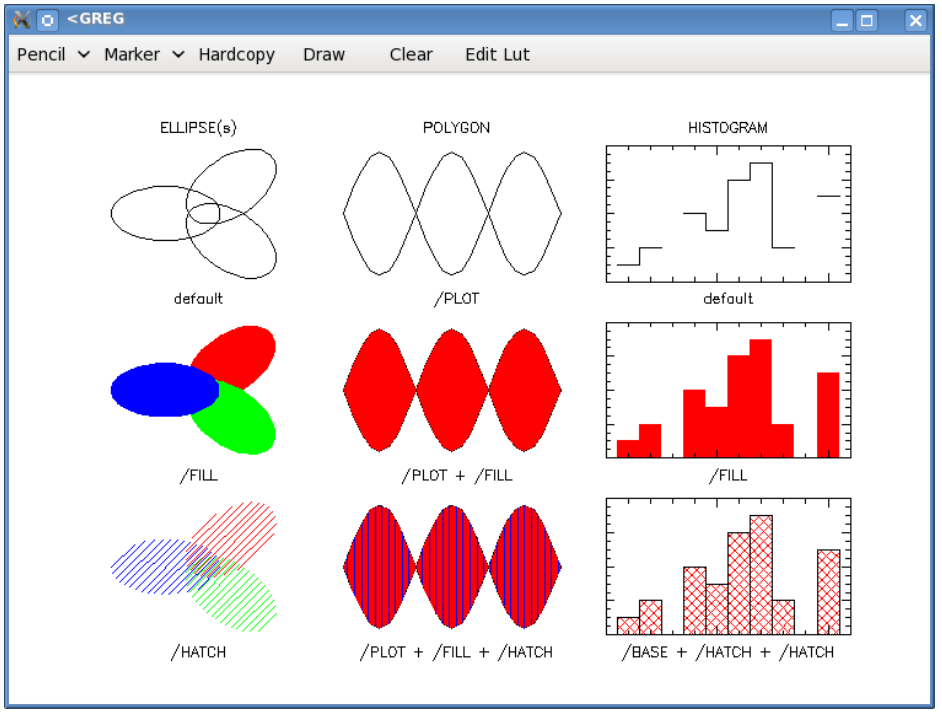


Need help (bis)? II. Demonstration procedures



Demo executed in the terminal and/or the plotting window:

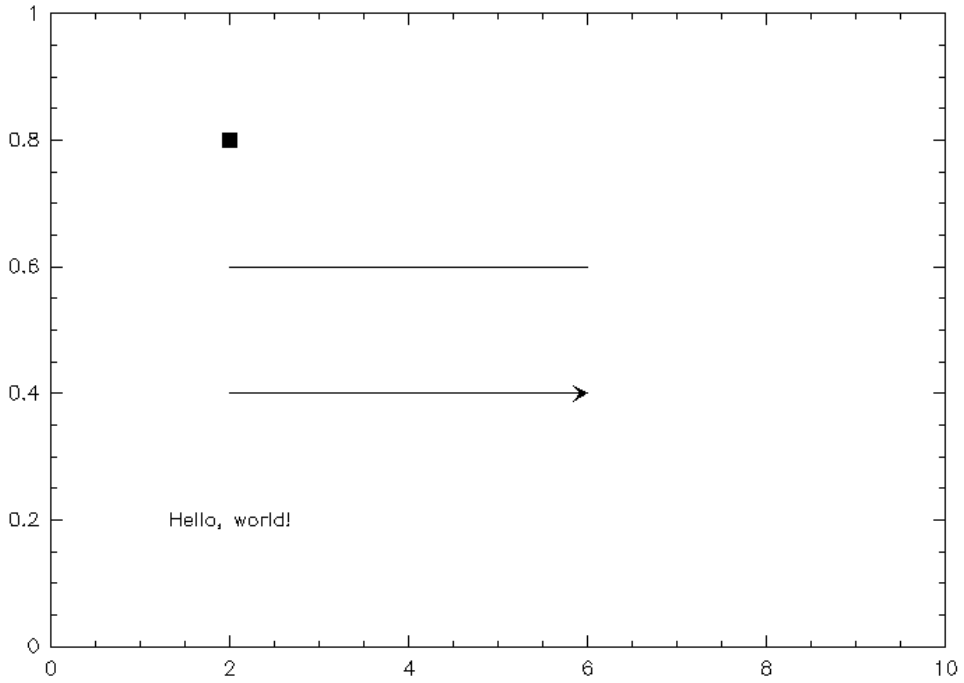
```
GREG>
I-DEMO,  Executing demonstration procedure
         @ gag_demo:demo-polygon.greg
```



First plots

First plots I. Basics

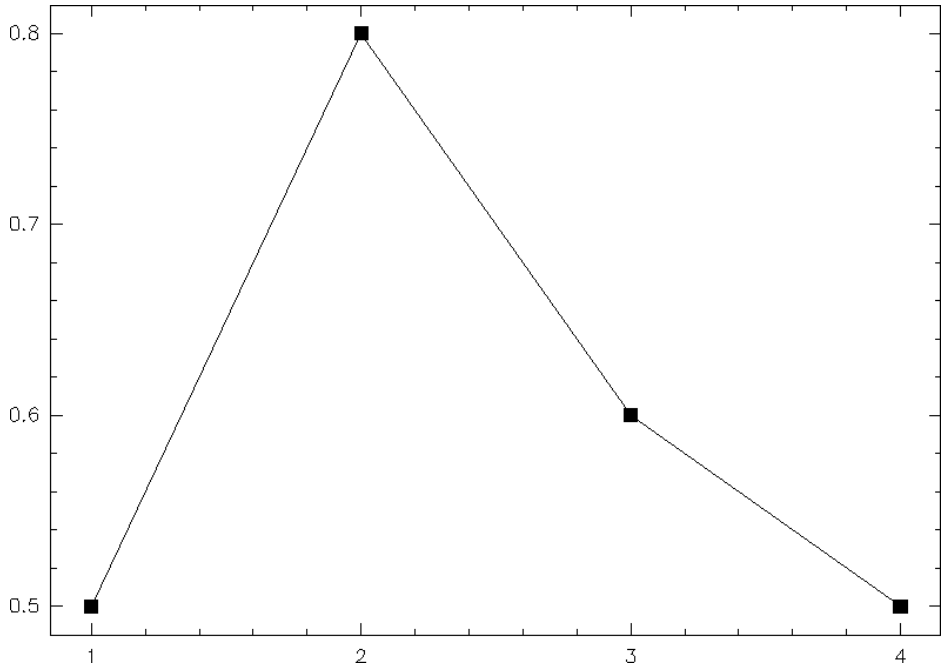
```
GREG> limits 0 10 0 1              ! Define X and Y ranges
GREG> box                          ! Draw a box at default location

GREG> set marker 4 3 0.5 0         ! Custom marker (square, filled, size 0.5cm, angle 0)
GREG> draw marker 2 0.8 /user      ! Draw 1 marker at these user coordinates

GREG> draw relocate 2 0.6 /user    ! Move pen to this position (nothing drawn)
GREG> draw line 6 0.6 /user        ! Draw a line from previous position to this position

GREG> draw relocate 2 0.4 /user    ! Move pen to this position (nothing drawn)
GREG> draw arrow 6 0.4 /user       ! Draw an arrow from previous position to this position

GREG> draw text 2 0.2 "Hello, world!" /user  ! Draw text centered at this position
```



First plots II. 1D data

Load the data in two X and Y arrays, use e.g. COLUMN command:

```
GREG> type values.dat
! X   Y
  1  0.5
  2  0.8
  3  0.6
  4  0.5
GREG> column X 1 Y 2 /file values.dat
GREG> examine nxy
NXY     =      4       ! Integer GLOBAL RO
GREG> examine x y
X is a double precision Array of dimensions 4
   1.00000000000000000        2.00000000000000000
   3.00000000000000000        4.00000000000000000
Y is a double precision Array of dimensions 4
   0.50000000000000000        0.80000000000000000
   0.60000000000000000        0.50000000000000000
```

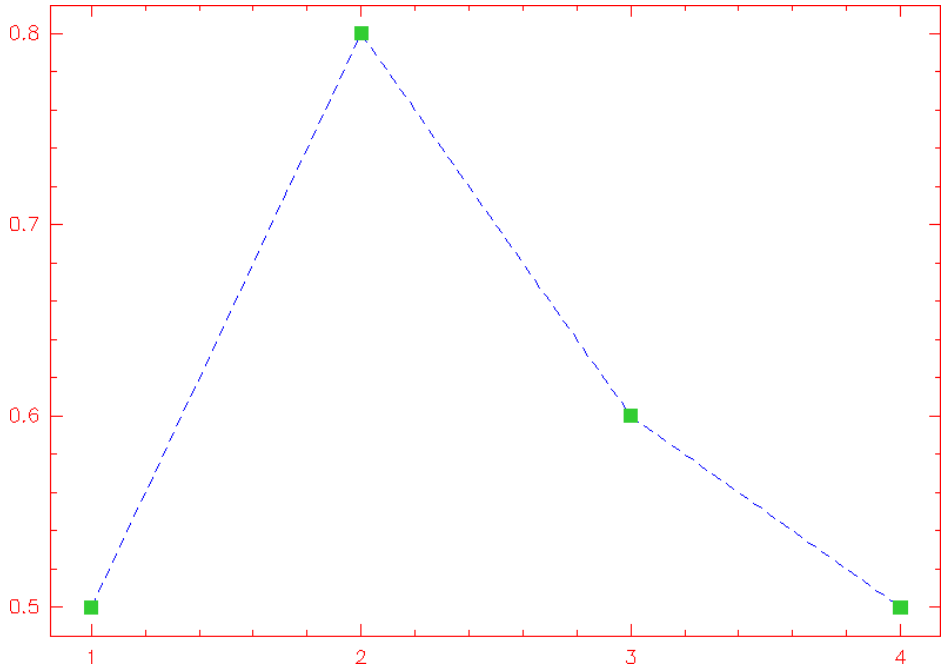


Plot these vectors:

```
GREG> clear             ! Erase the previous plot
GREG> limits /var x y   ! Automatic limits from X and Y with 5% margins
GREG> box
GREG> connect x y       ! Connect the values with current pen
GREG> points x y        ! Draw points with current marker
```

First plots III. Changing pen attributes

```
GREG> clear
GREG> pen /colour red
GREG> box                       ! Red box
GREG> pen /colour blue /dash 2
GREG> connect x y               ! Dashed blue line
GREG> pen /colour lime_green
GREG> points x y                ! Green markers
GREG> pen /default              ! Revert to defaults
```



Get the exhaustive list of colours with:

```
GREG> pen /colour ?
```

or execute demo:
Widget > Demos > Greg > Pencil colour names

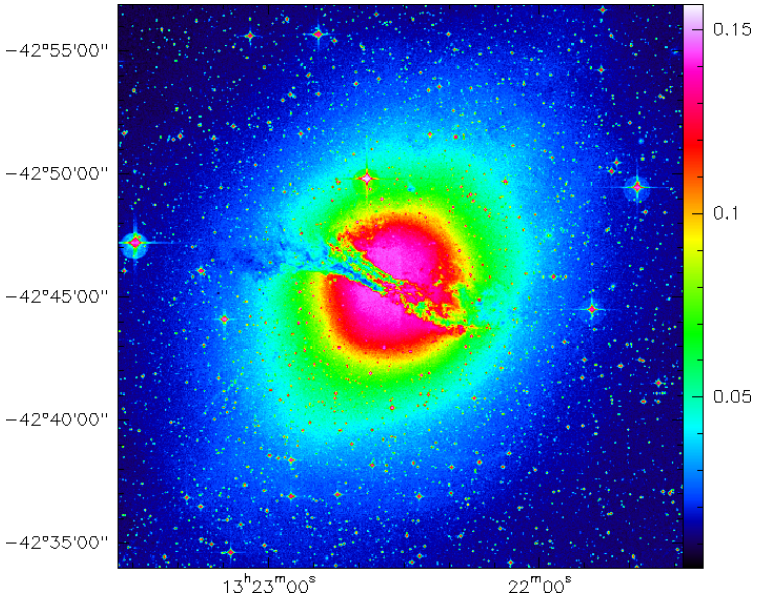
First plots IV. 2D data

Load the image in the RG (Regular Grid) buffer:

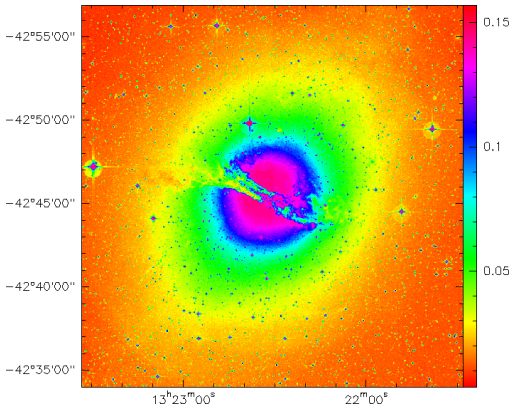
```
GREG> clear                          ! Erase the previous plot
GREG> image gag_demo:centaurus.gdf   ! Load demo image in RG buffer
W-GREG3,  Defining GREG3 variables
W-GDF,  UNKNOWN Velocity type defaulted to LSR
I-GIO_RIH,  File is  [EEEI to IEEE] , Header Version 1 (32 bit)
```

Plot the RG buffer:

```
GREG> limits /rg      ! Automatic limits from RG description
GREG> set box match   ! Best box matching X and Y scales
GREG> plot            ! Plot the RG buffer
GREG> box /absolute   ! Overlay the box with absolute coordinates
GREG> wedge           ! Add the colour scale on right side
```



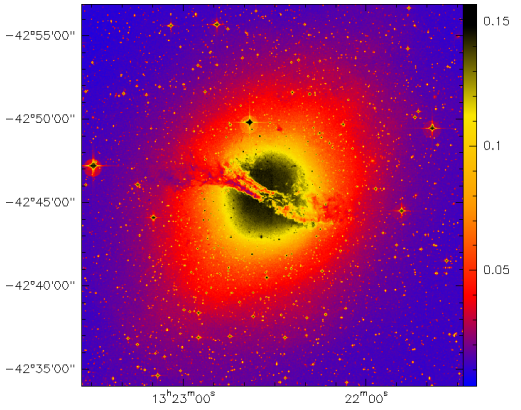
First plots V. Changing Look-Up-Table



Do not CLEAR current PLOT, just use command LUT:

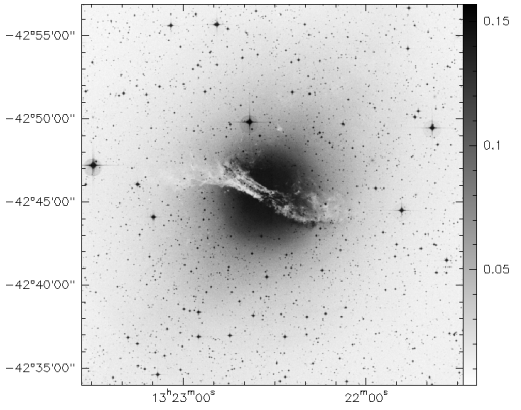
```
GREG> lut color
GREG> lut smooth
GREG> lut black
```

⇒ dynamically updated!



Get the exhaustive list of Look-Up-Tables with:

```
GREG> lut ?
```

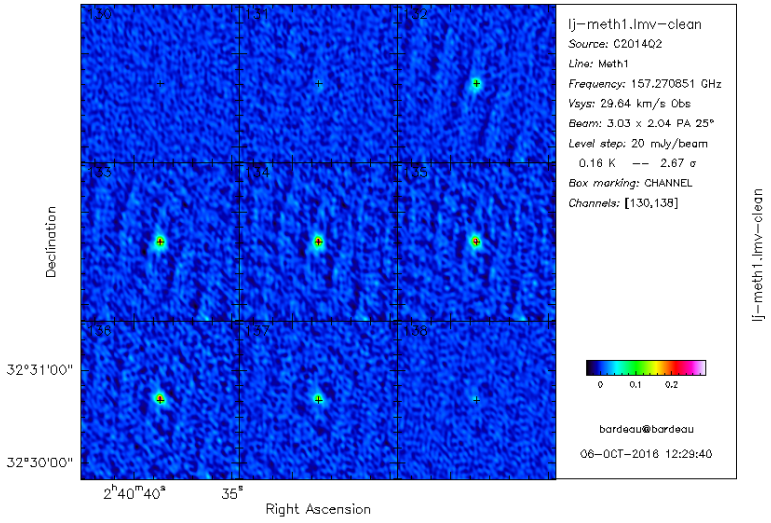


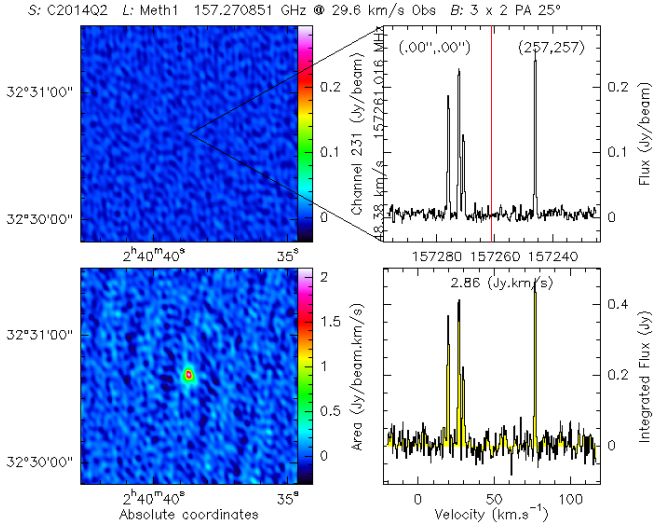
First plots VI. 3D data

Basic support : plot one plane at a time.

```
GREG> image lj-meth1.lmv-clean /plane 135
```

Advanced support : use integrated tools to display channel maps (GO BIT), interactive exploration (GO VIEW), and many more ⇒ see Greg 3D presentation.





Hardcopies

Hardcopies

The plot in the current plotting window can be exported to an output file with the command HARDCOPY, e.g.

```
GREG> hardcopy test.ps /device ps
I-HARDCOPY,  test.ps created
GREG> hardcopy test.ps /device ps              ! No overwrite by default
E-HARDCOPY,  File test.ps already exists
GREG> hardcopy test.ps /device ps /overwrite   ! Allow overwrite
I-HARDCOPY,  test.ps created
```

Supported *devices* are:

- PostScript (PS), vectorial, well suited for printing,
- Encapsulated PostScript (EPS), vectorial, well suited for inclusion in LaTeX/PDF documents,
- Portable Network Graphics (PNG), bitmap,
- Scalable Vector Graphics (SVG), vectorial, well suited for inclusion in web page.

Pros/cons:

- Vectorial formats:
  ⊕ no resolution loss,
  ⊖ the richer the plot the larger the file,
- Bitmap formats:
  ⊕ lossless compression, limited size even for rich plots,
  ⊖ limited resolution.

Controlling the position of your plot on the page

Controlling the position of your plot on the page

I. Basics

set plot_page Arg(s) LANDSCAPE shortcut for lanscape A4, PORTRAIT shortcut for portrait A4, X Y for custom plotting page.

set box_location Gx1 Gx2 Gy1 Gy2 Define the X and Y coordinates of the box in physical units of the paper page (default A4 in landscape mode: $0 \leq$ Gx1 and Gx2 ≤ 30 cm, and $0 \leq$ Gy1 and Gy2 ≤ 21 cm).

set viewport Px1 Px2 Py1 Py2 Same "set box_location" but the coordinates are given in terms of fraction of the plot page, i.e., $0 \leq$ Px1 and Px2 ≤ 1.0, and $0 \leq$ Py1 and Py2 1.0.

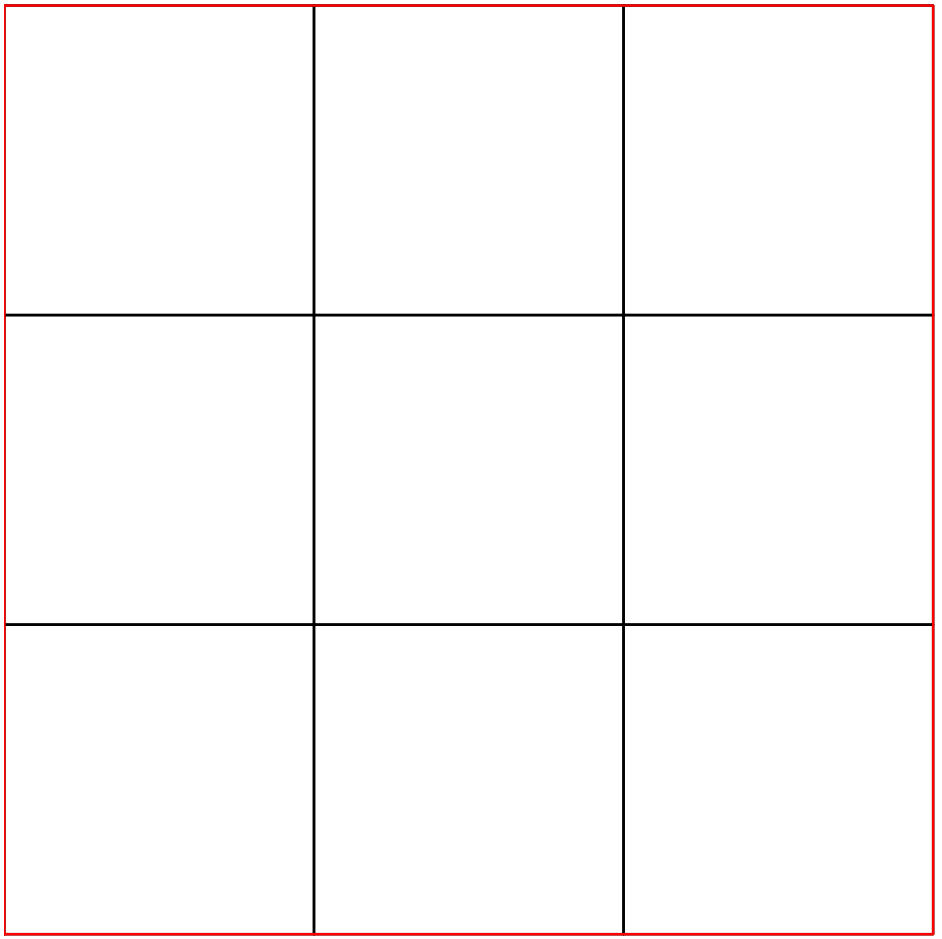
Advantage Extremely flexible.

Inconvenient Imply many human computations.

Controlling the position of your plots on the page

II. Easy way: 1. Syntax #1: @ window_xy nx ny ith

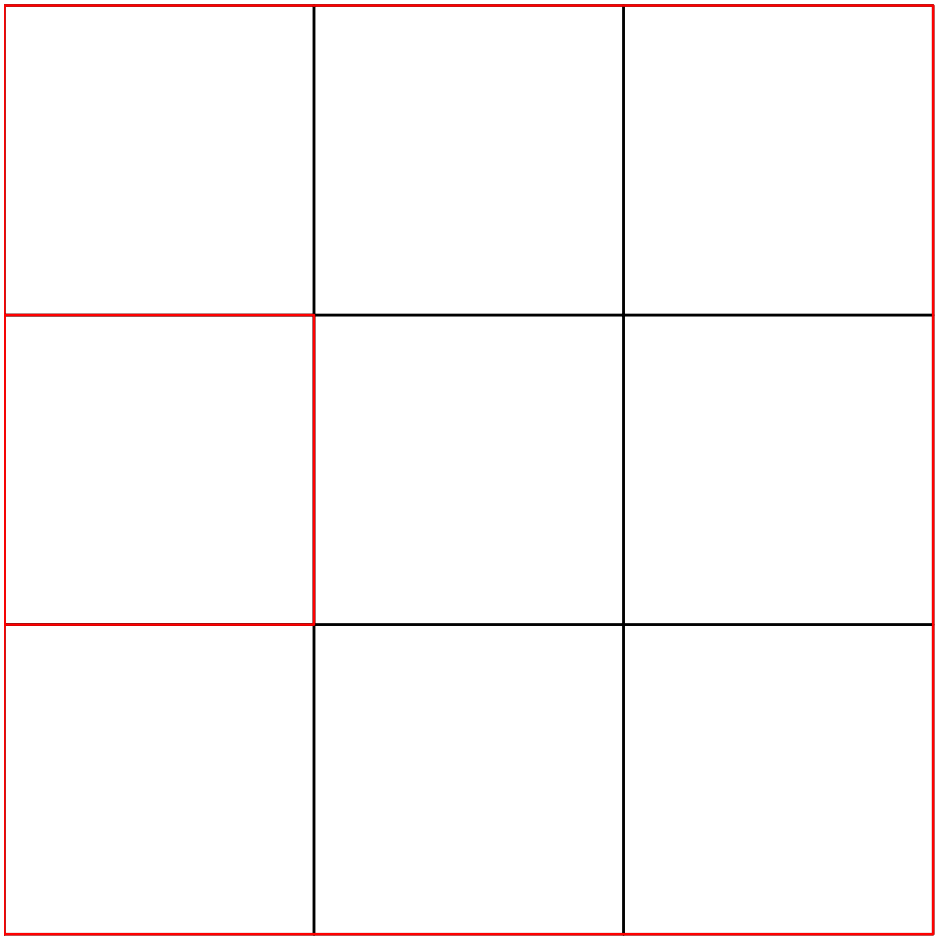
```
@ window_tools.greg
clear plot
@ window_init
pen 0
for ibox 1 to 9
   @ window_xy 3 3 'ibox'
   box n n n
next
pen 1
@ window_xy 1 1 1
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 2. Syntax #2: @ window_xy nx ny ix iy

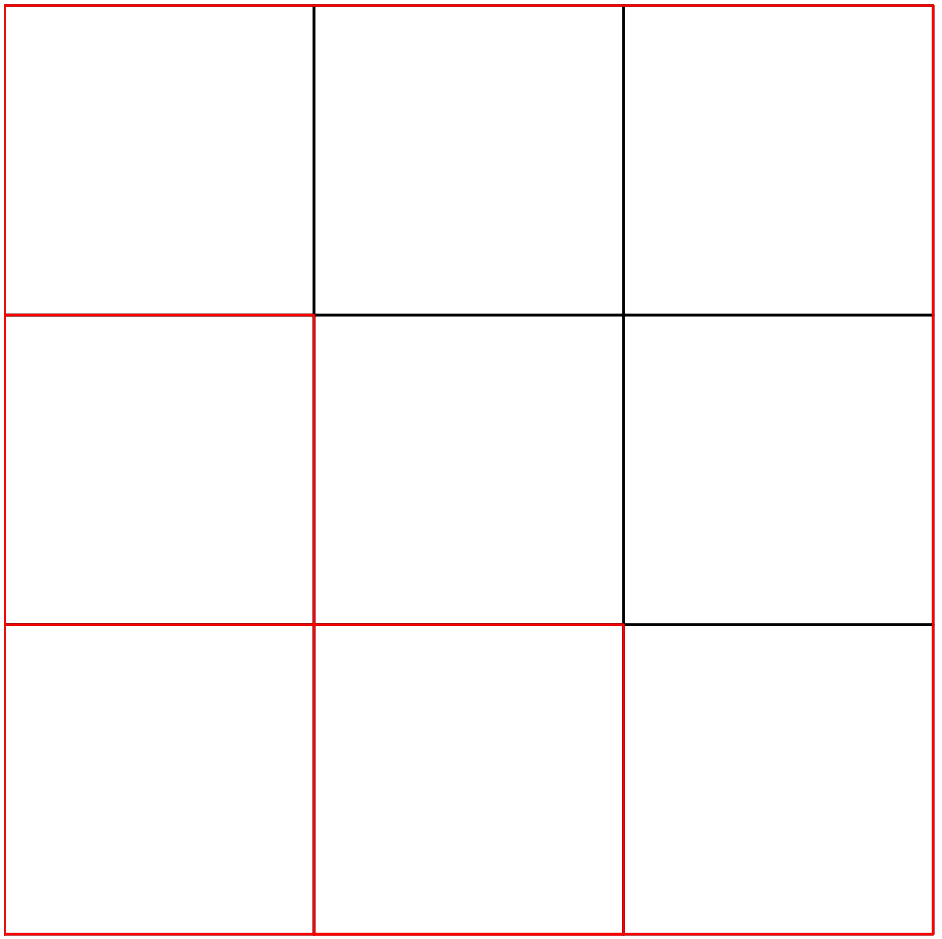
```
pen 0
@ window_xy 1 1 1 1
box n n n
@ window_xy 3 3 1 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 2. Syntax #2: @ window_xy nx ny ix iy

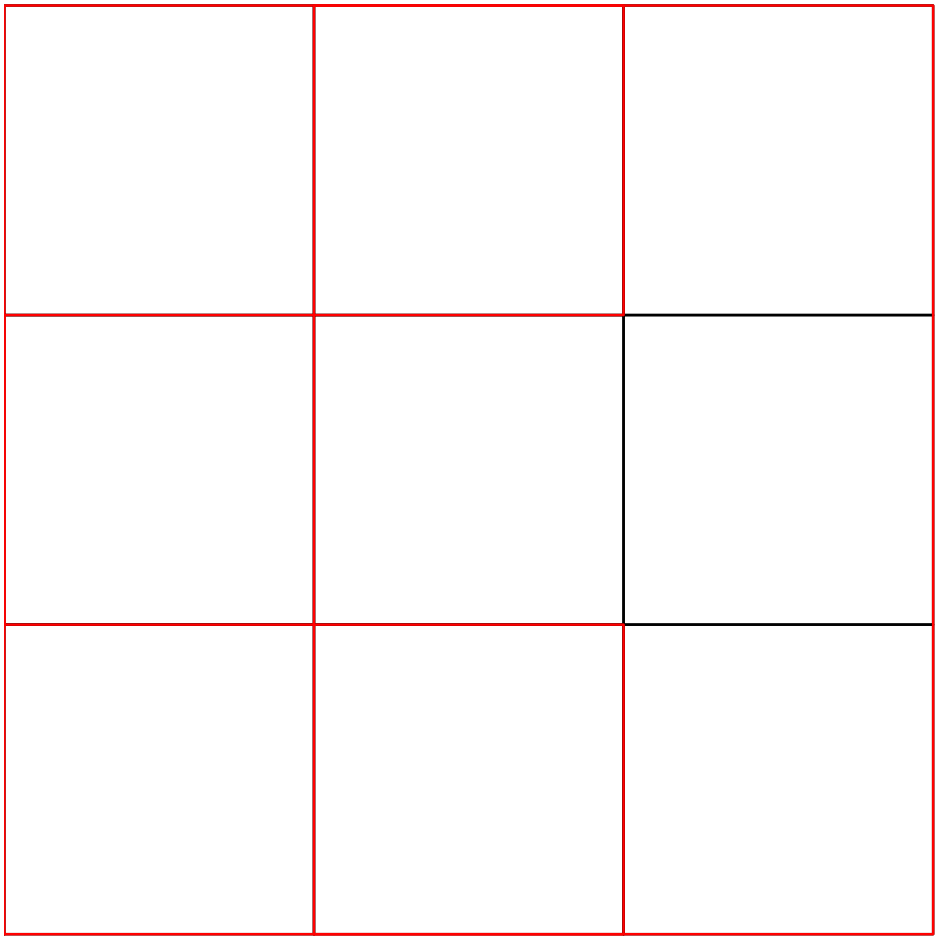
```
pen 0
@ window_xy 1 1 1 1
box n n n
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 2. Syntax #2: @ window_xy nx ny ix iy

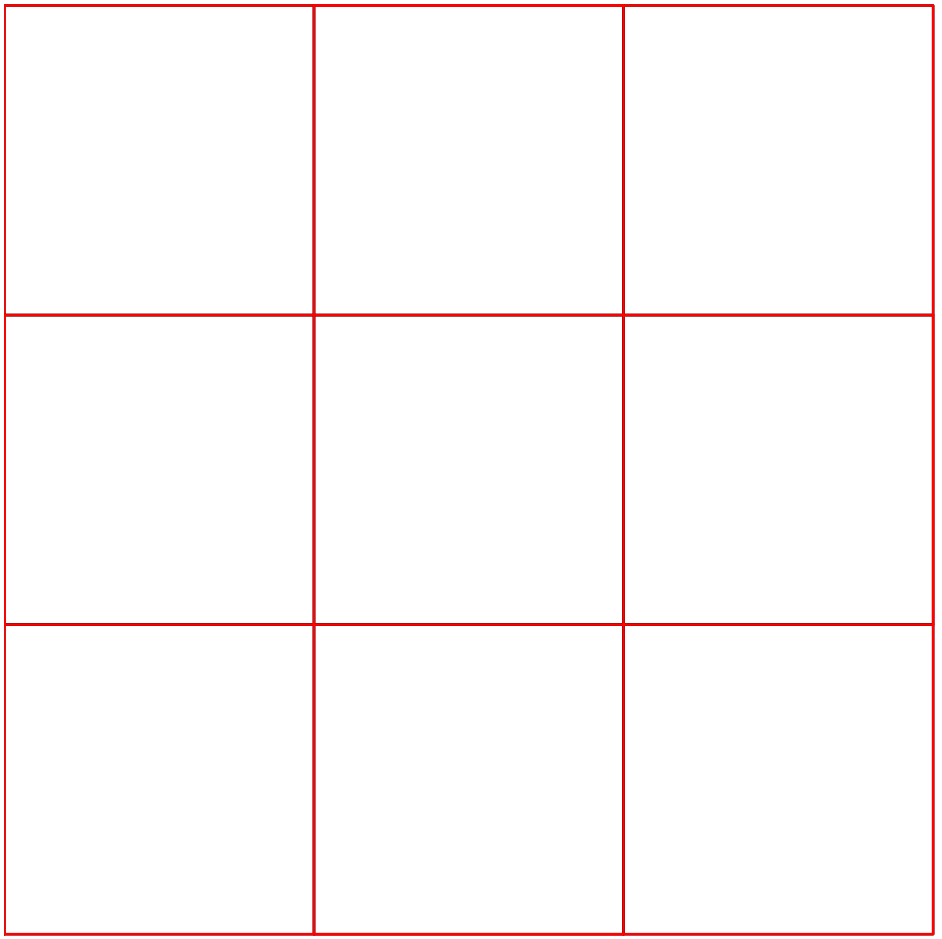
```
pen 0
@ window_xy 1 1 1 1
box n n n
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 2 1
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 2. Syntax #2: @ window_xy nx ny ix iy

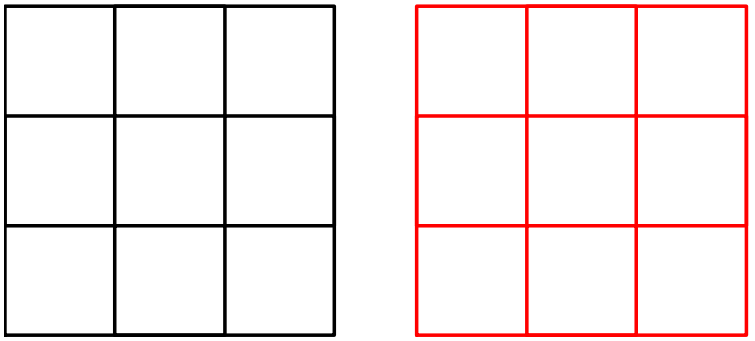
```
pen 0
@ window_xy 1 1 1 1
box n n n
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 3. Choosing the position of the external box

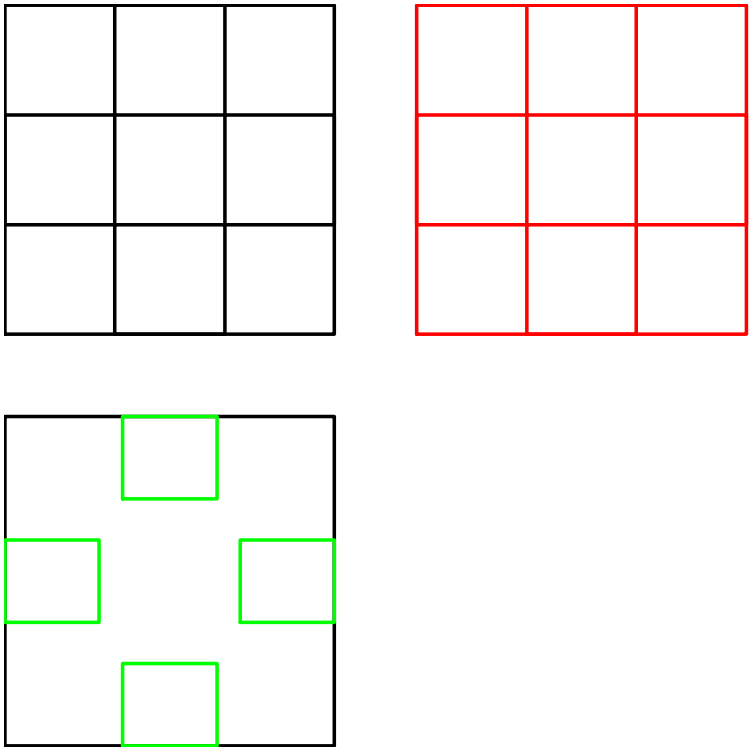
```
pen 0
let position 0.55 0.75 0.65 0.85
@ window_xy 1 1 1 1
box n n n
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
!
pen 0
let position 0.80 1.0 0.65 0.85
@ window_xy 1 1 1 1
box n n n
pen 1
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 4. Getting margins between the internal boxes

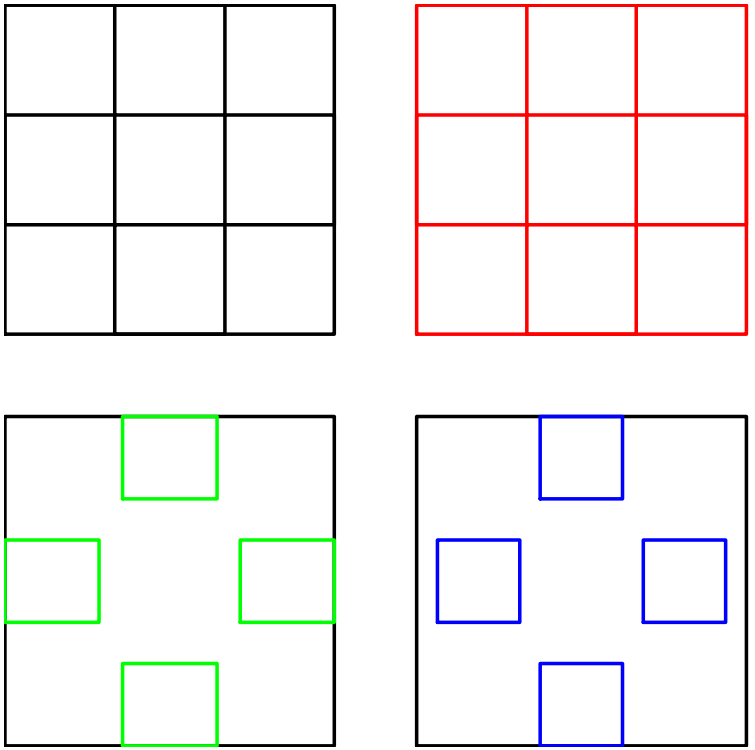
```
pen 0
let position 0.55 0.75 0.40 0.60
let inter 0.25 0.5
let aspect_ratio no
@ window_xy 1 1 1 1
box n n n
pen 2
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 4. Getting margins between the internal boxes

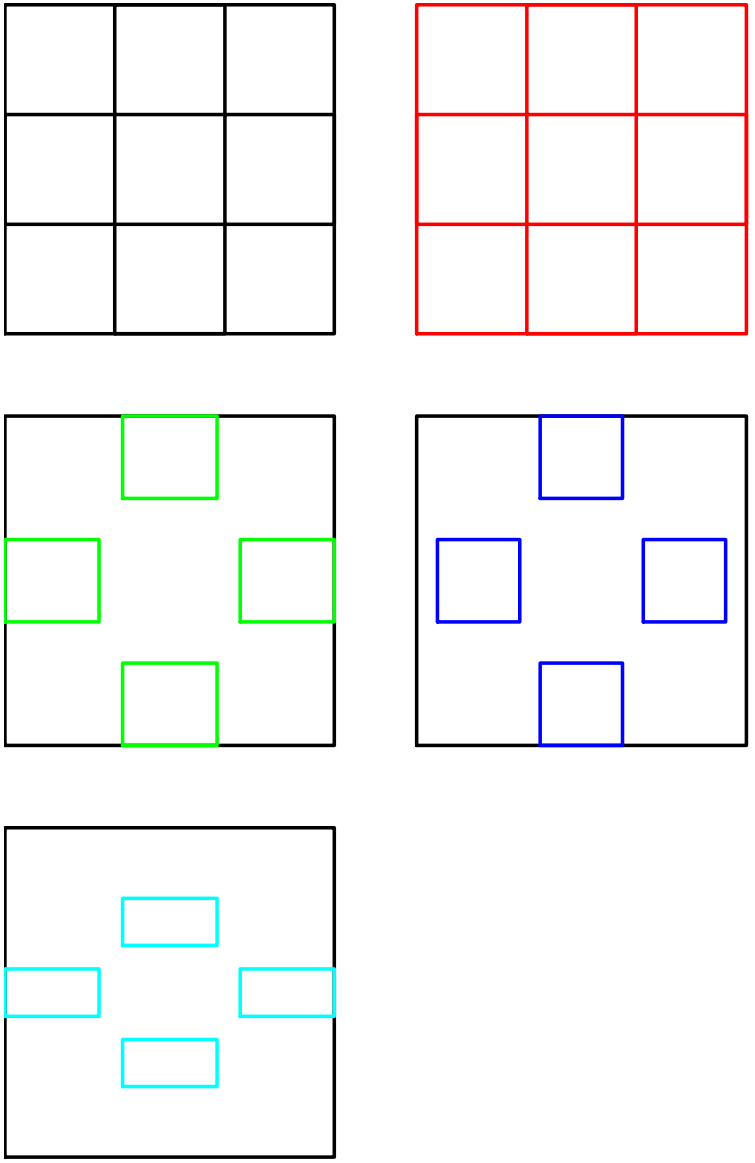
```
pen 0
let position 0.80 1.00 0.40 0.60
let inter 0.25 0.5
let aspect_ratio yes
@ window_xy 1 1 1 1
box n n n
pen 3
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 4. Getting margins between the internal boxes

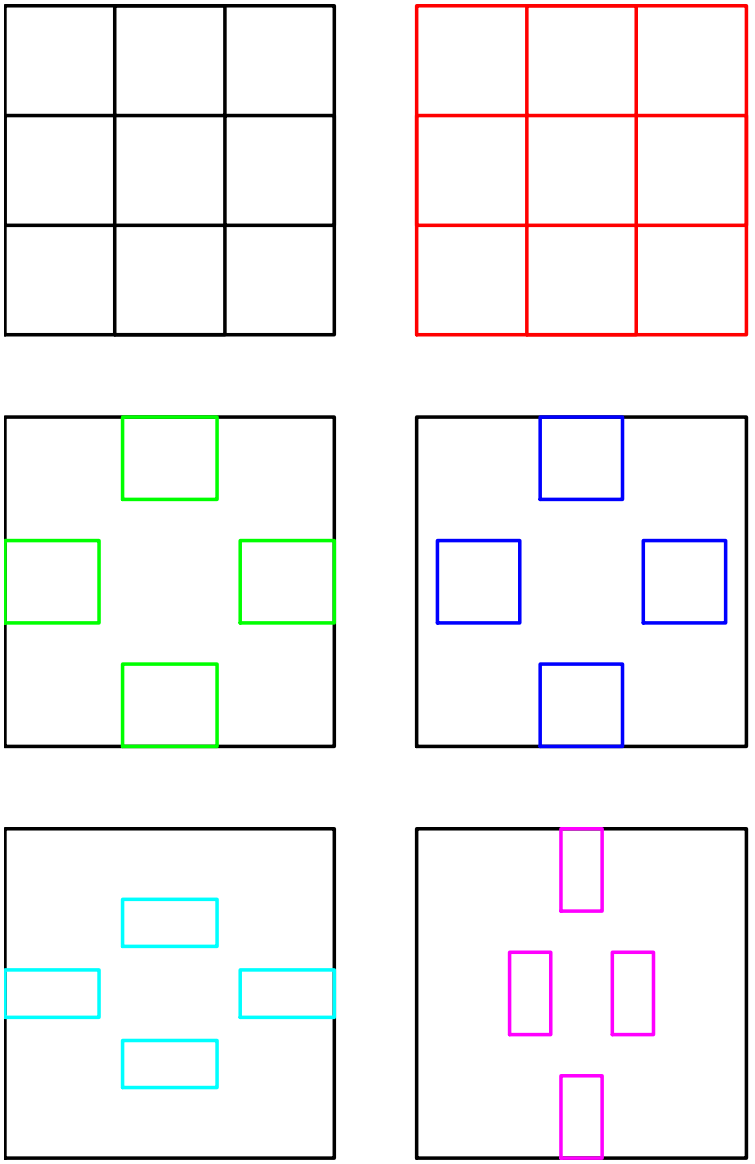
```
pen 0
let position 0.55 0.75 0.15 0.35
let inter 0.25 0.5
let aspect_ratio yes
@ window_xy 1 1 1 1
box n n n
pen 4
let aspect 0.5
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 4. Getting margins between the internal boxes

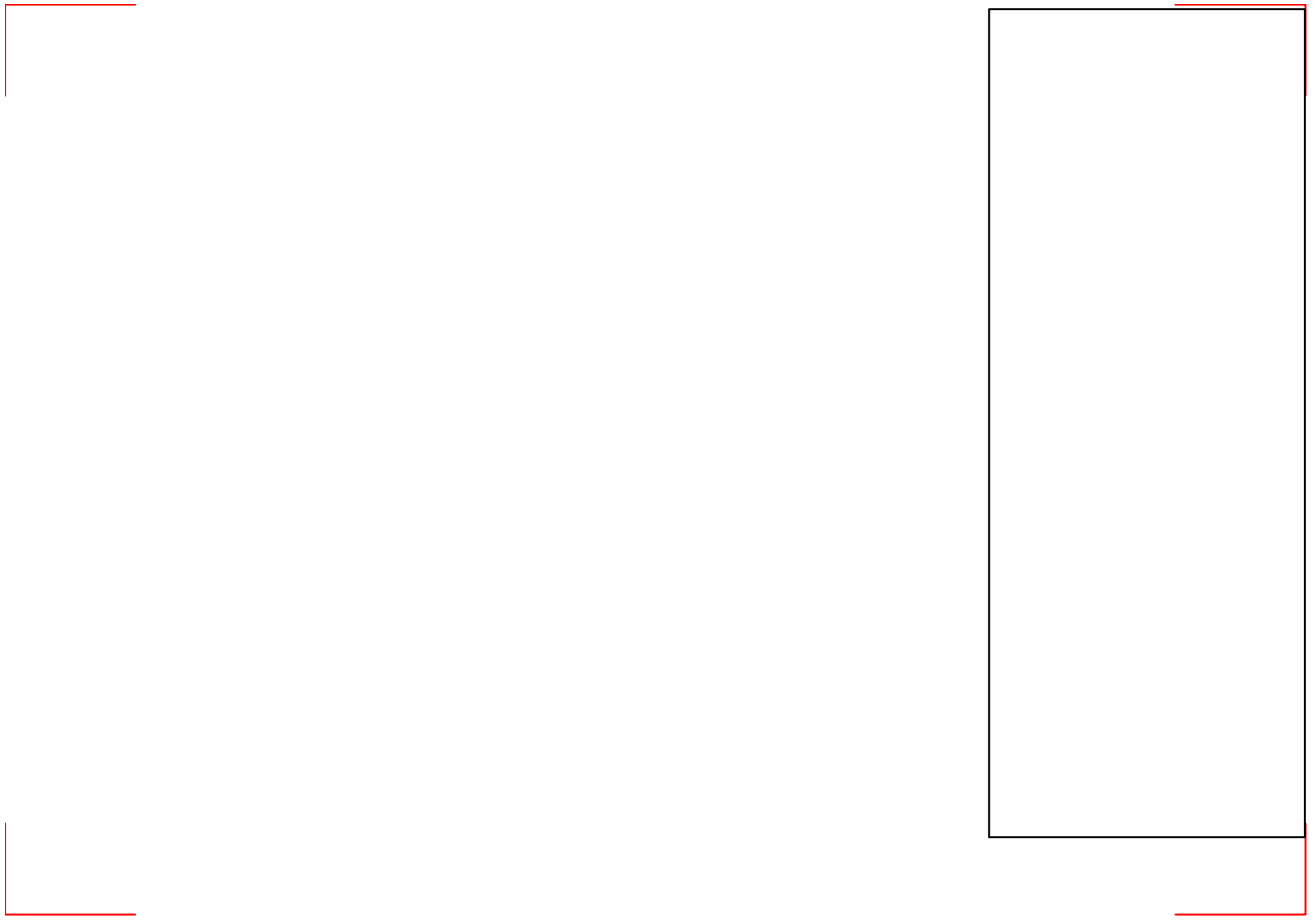
```
pen 0
let position 0.80 1.00 0.15 0.35
let inter 0.25 0.5
let aspect_ratio yes
@ window_xy 1 1 1 1
box n n n
pen 6
let aspect 2.0
@ window_xy 3 3 1 2
box n n n
@ window_xy 3 3 2 1
box n n n
@ window_xy 3 3 2 3
box n n n
@ window_xy 3 3 3 2
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 5. Header position

```
clear
pen 1
corner
pen 0
@ window_init
let aspect_ratio no
!
@ header_position
@ window_xy 1 1 1
box n n n
```



Controlling the position of your plots on the page

II. Easy way: 6. Data position

```
clear
pen 1
corner
pen 0
@ window_init
let aspect_ratio no
!
@ header_position
@ window_xy 1 1 1
@ mybox pause 12
!
@ plot_position
@ window_xy 1 1 1
box n n n
```