

Making Plots Using MacAnova

MacAnova can make quite good graphs using these plotting commands:

Command	Description
<code>plot()</code>	Plot each column of matrix <code>y</code> against vector <code>x</code> using standard symbols, with or without connecting lines
<code>chplot()</code>	Same as <code>plot()</code> except you can specify plotting symbols
<code>lineplot()</code>	Same as <code>plot()</code> without plotting symbols but with lines
<code>stringplot()</code>	Write CHARACTER strings at specified positions in a graph
<code>addpoints()</code>	Like <code>plot()</code> , but points are added to an existing graph
<code>addchars()</code>	Like <code>chplot()</code> , but plotting symbols are added to an existing graph
<code>addlines()</code>	Like <code>lineplot()</code> , but lines are added to an existing graph
<code>addstrings()</code>	Like <code>stringplot()</code> , but strings are added to an existing graph
<code>boxplot()</code> and <code>vboxplot()</code>	Make parallel boxplots of data in components of structure argument or columns of matrix argument
<code>hist()</code>	Make histogram of data in vector
<code>plotmatrix()</code>	Make array of small scatter plots of every column of a matrix vs every other column
<code>showplot()</code>	Re-display the most recent plot, possibly with new labels and changed minima or maxima

Carapace MacAnova vs Classic MacAnova: In Carapace MacAnova, the most up-to-date version, commands are entered without the prompt (Cmd>) in the lower panel of the MacAnova window. After hitting **Enter** or **Return**, a command is moved to the upper panel preceded by Cmd> and followed by any output produced.

In Classic MacAnova, commands are entered after Cmd> on the last line of the output window.

In examples below, commands are in *italics* preceded by the prompt Cmd>. You would type only the *italic* part in any version of MacAnova.

The example plots are all produced in Macintosh Classic MacAnova and differ slightly in the orientation and font of labels from graphs produced in Carapace MacAnova.

In Carapace MacAnova, you can get HTML-based help by selecting **Help** on the **Help** menu which opens a browser-like window. For Classic MacAnova, you can download the HTML help files and access them directly using your favorite browser such as Netscape or Firefox. In all versions, you can use commands `help()` and

Graphics in MacAnova

`usage()` to access the same information (it's printed in the output window). In the description below, examples of this usage are given. Clickable topic names are in the index page of the HTML-based help.

Using `plot()`

The simplest usage is

```
Cmd> plot(x,y)
```

which plots the values in variable `y` against the values in variable `x`. `x` must be a REAL vector and `y` must be a REAL vector or matrix. All the other commands in the first two groupings in the table need at least these two arguments `x` and `y`

Shortcut for equally spaced `x` Usually `x` and `y` have the same number of rows. However, except for `stringplot()` and `addstrings()`, you can use an argument `x` with fewer rows than `y` as a "short cut" to specify *equally spaced* values.

When `x` is a scalar (single number), `x` is "expanded" to `vector(x,x+1,x+2,...)`.

When `x` is a vector with two elements, `x` is expanded to `vector(x[1], x[1]+x[2], x[1]+2*x[2], ...)`, that is `x[1]` is the starting value and `x[2]` is the step size or increment.

Examples

```
Cmd> plot(1,y)
```

plots `y[i]` against `i = 1, 2, ..., n`.

```
Cmd> plot(vector(1947,1/12),y),xlab:"Year")
```

might be used to plot a monthly time series starting January 1947 with the time axis labeled in years.

Use **Help** on the **Help** menu or `help()` and `usage()` (for example, type `help(plot)` or `usage(plot)`) to get specific details on these commands.

The following help topics provide more general information on making graphs.

Topic	What it covers
graphs	General information on high and low resolution graphs
graph_keys	Complete summary of keywords used on graphics commands
graph_border	Information on specifying which sides to draw a border
graph_ticks	Information on modifying default tickmark length and placement
graph_files	Summarizes plotting options that allow you to save a plot in a file
GRAPHWINDOWS	Information about special structure with encapsulated information about all graphics windows
graph_assign	Information on how to plot by direct assignment to a component of GRAPHWINDOWS

This handout describes the elements most plotting commands have in common.

Graphics in MacAnova

All plotting commands recognize keywords `xmin`, `xmax`, `ymin`, `ymax` (used to set left, right, bottom and top edges of graphs), `xlab`, `ylab` (used to label X- and Y-axes), `title` (provides header), `xaxis`, `yaxis` (can suppress drawing $x = 0$ and $y = 0$ lines when visible), `logx`, `logy` (used to specify logarithmic spacing), `dumb`, `new`, `show`, `keep`, `file`, and `pause`.

All plotting commands except `boxplot()`, `vboxplot()` and `showplot()` recognize keyword phrase `add:T`.

Some other keywords such as `symbols`, `lines`, `linetype`, `thickness` and `justify` are meaningful only to particular commands.

`showplot()` is particularly useful since it allows you to replot a graph with changed labelling or scaling.

You can get details on any of these keywords using **Help** on the **Help** menu or command `help()`.

```
Cmd> help(graph_keys:"?") # get subtopics for topic graph_keys
Available subtopics for topic 'graph_keys' are:
  x
  y
  symbols
  strings
  keys1
  add1
  title
  xlab
  ylab
  xminmax
  ...
  ...
  assignment_to_GRAPHWINDOWS
Type help(graph_keys,subtopic:vector("subtopicA","subtopicB",...))
```

Note: As of 9/7/05, this usage crashes Carapace MacAnova. It will be fixed soon.

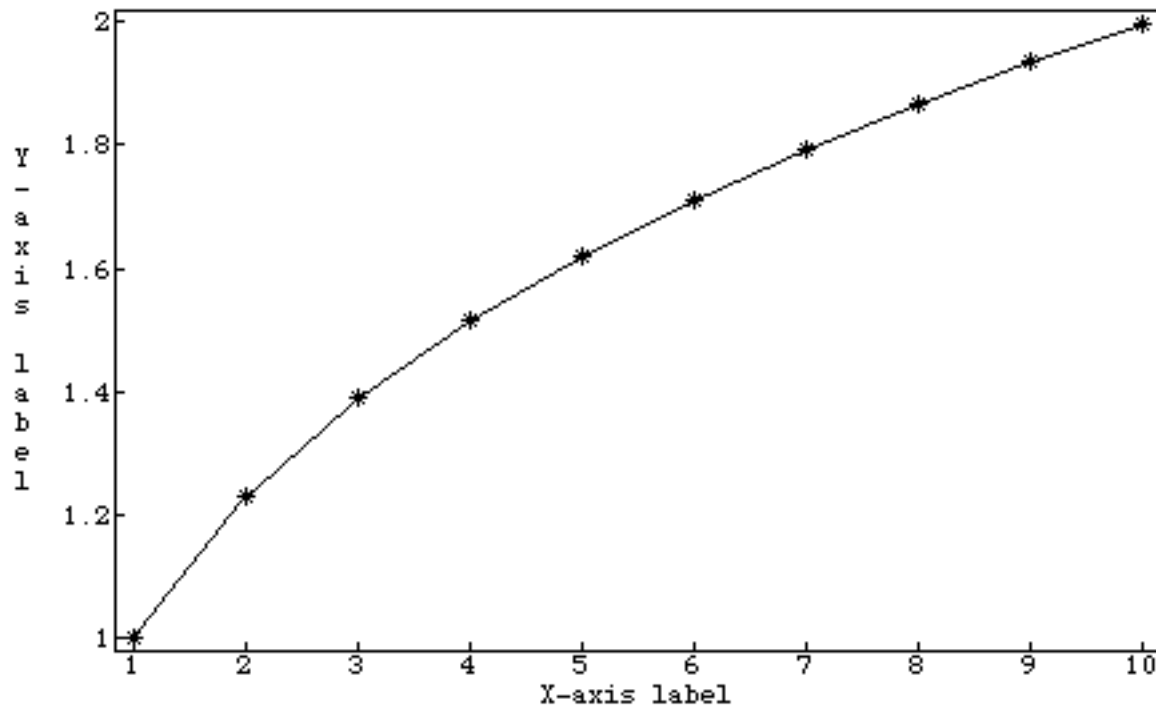
```
Cmd> help(graph_keys:"xminmax") # get help on a subtopic
Subtopic 'xminmax' of help on 'graph_keys'
  xmin:xMinVal or xmin:?      Minimum and maximum values for
  xmax:xMaxVal or xmax:?      x-axis and y-axis. Value ?
  ymin:yMinVal or ymin:?      means compute from new data and
  ymax:yMaxVal or ymax:?      any data in graph being modified.
```

Graphics in MacAnova

Examples Here is a simple example of the use of `plot()`.

```
Cmd> x <- run(10); y <- x^.3 # y is x to the 0.3 power
Cmd> plot(x,y,lines:T,xlab:"X-axis label",ylab:"Y-axis label",\
title:"Sample plot of x^.3 vs x",ticks:"LB")
```

Sample plot of $x^{.3}$ vs x



This plots vector `y` against `x` using a default plotting symbol, connecting the points with lines (because `lines:T` was an argument).

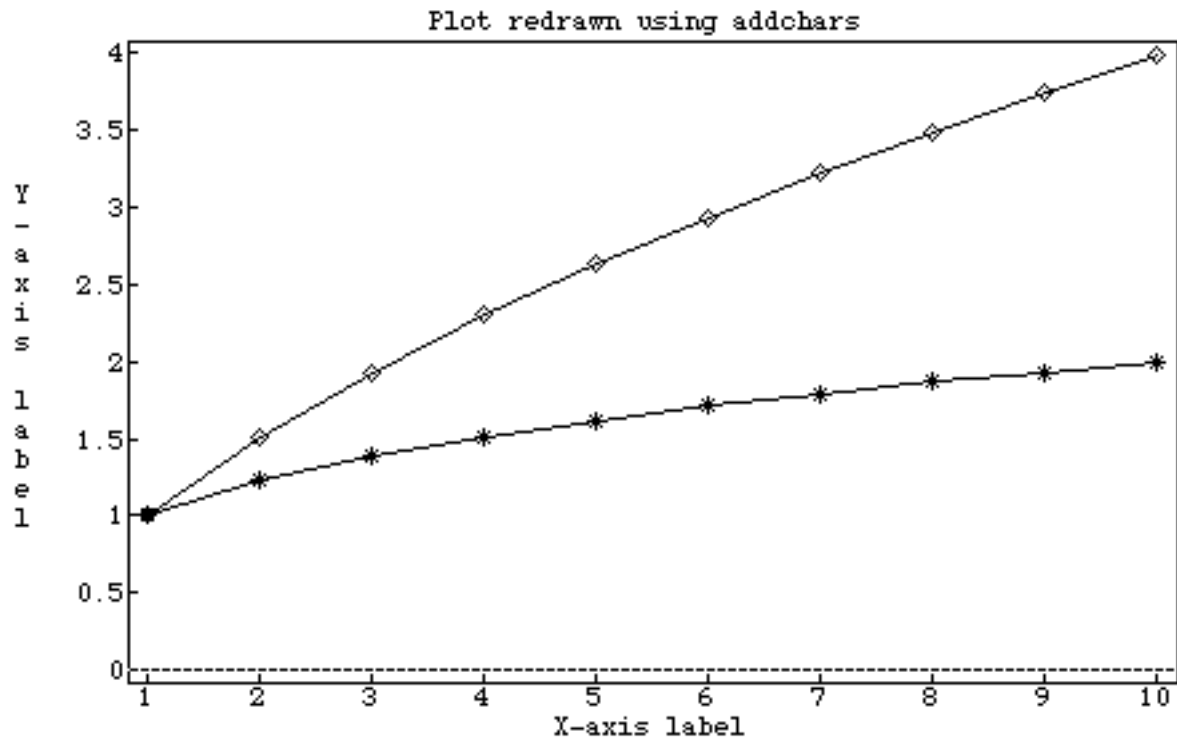
Keyword `title` specified the title above the graph, and keywords `xlab` and `ylab` specified X-axis and Y-axis labels. When you don't use `title`, the graph is untitled and when you don't use `xlab` or `ylab`, the corresponding axis is labelled with the name of the variable being plotted.

Argument `ticks:"LB"` specifies that tick marks are put only on the *Left* and *Bottom* edges of the graph. Without `ticks`, tick marks are drawn on all four sides.

In the following example, we add additional points connected by lines. The plotting symbol is specified by the `symbols:"\1"`, which draws diamonds (\diamond). The plotting symbol in the first plot (*) could have been specified by `symbols:"\6"`. In addition, `ymin` and `ymax` reset the minimum and maximum for the y-axis (`xmin` and `xmax` would do the same for the x-axis). Keyword `title` changes the title.

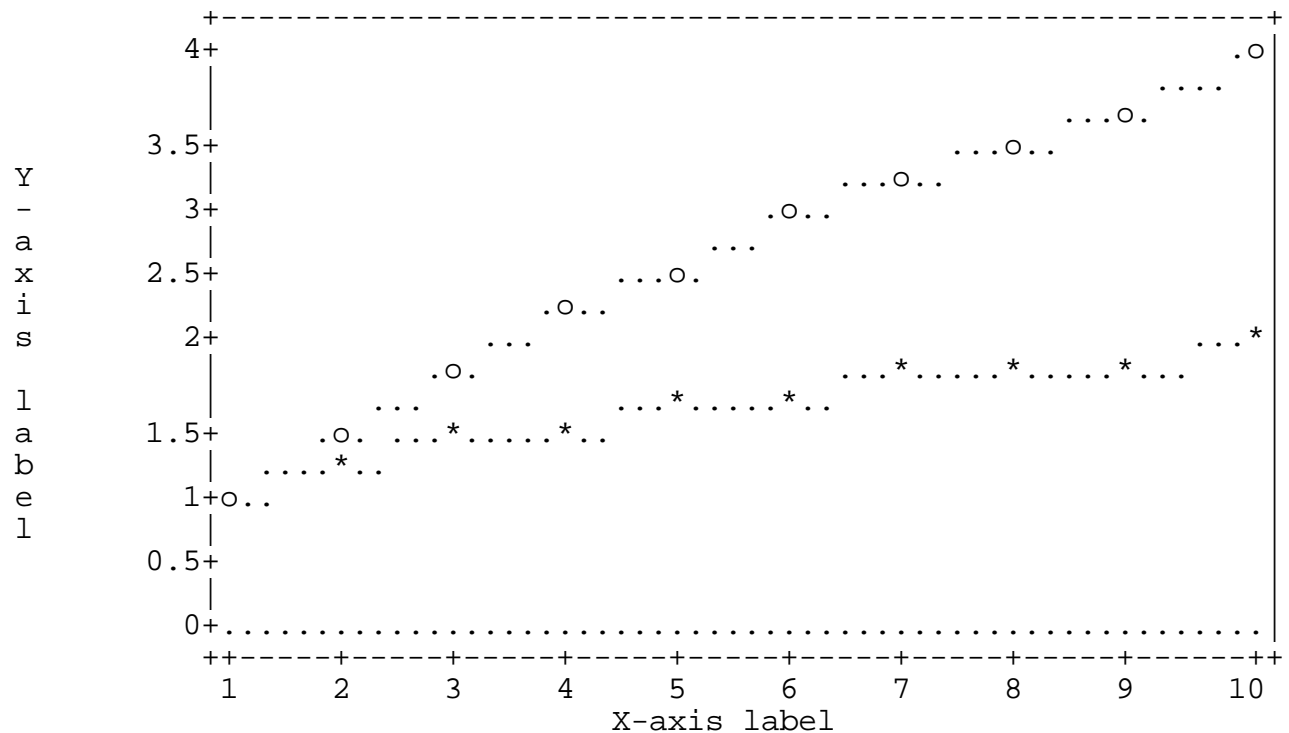
Graphics in MacAnova

```
Cmd> addchars(x,y^2,symbols:"\1",lines:T,\
title:"Plot redrawn using addchars",ymin:0,ymax:y[10]^2)
```



\
Here is an illustration of how showplot() might be used:

```
Cmd> showplot(dumb:T,title:"Plot redrawn using dumb:T",width:70)
Plot redrawn using dumb:T
```



Graphics in MacAnova

Because `dumb:T` was an argument, the plot is a *low resolution* plot constructed from ordinary printing symbols. Such a “dumb” plot is not as elegant as a high resolution plot, but it can be printed on any printer, and included in any word processor document (a font with equal width letters is *essential*). If you are save input and output using `spool()`, “dumb” plots are written to the spooling file, but high resolution plots are not. Argument `width:70` limited the width of the plot to 70 characters so that it would fit on this page.

You can make sure *all* your plots are “dumb” by `setoptions(dumbplot:T)`. If you do this, then to get a high resolution graph you have to use `dumb:F` on a plotting command.

GRAPH variables MacAnova can create and work with GRAPH variables. A GRAPH variable encapsulates all the information used to create a plot, including data, labels, line types and so on.

All plotting commands automatically create a GRAPH variable with name `LASTPLOT`, as a “side effect”. You can assign `LASTPLOT` to another variable (for example, `plot1 <- LASTPLOT`) or redisplay it using `showplot()`.

Just typing the name of a GRAPH variable produces a plot similar to that produced when `dumb:T` is used.

By default, commands such as `addchars()` or `showplot()` that redisplay a graph or add information to it work with the graph encapsulated in `LASTPLOT`. This is also the case for any command with argument `add:T`. You can modify or display the graph in any GRAPH variable, by putting it as first argument to the command, before additional X and Y values, if any. For example, if `plot1` is a GRAPH variable, `showplot(plot1, title:"This is a new title")`, replots the graph with a new title. It does not change `plot1` itself, but updates `LASTPLOT` to reflect the new graph.

Argument `keep:F` on any graphics command suppresses the creation of `LASTPLOT`. One reason for doing this would be if you are short of memory and don’t plan to modify the plot.

MacAnova also saves information about the graphs in all the graph windows in a special structure variable `GRAPHWINDOWS`, each component of which is a GRAPH variable. Help topics `GRAPHWINDOWS` and `graph_assign` give information about this feature. You can modify graphs by appropriate assignments to components of `GRAPHWINDOWS` (e.g., `GRAPHWINDOWS[3] <- structure(title:"New title")`).

Suppressing the display of a graph When you are building up a graph in several steps, it can be a nuisance to display intermediate graphs. To avoid this, use keyword phrase `show:F` on all the plotting commands except the final one. Thus you could produce the second plot above by

```
Cmd> plot(x,y,lines:T,xlab:"X-axis label",ylab:"Y-axis label",show:F)
Cmd> addchars(x,y^2,symbols:"\1",lines:T,ticks:"LB",\
            title:"Plot redrawn using addchars", ymin:0,ymax:y[10]^2)
```

It is illegal to use both `show:F` and `keep:F`.

Graphics in MacAnova

Making a copy of a graph for use in a document The only universally applicable way to do this is to write a “dumb” plot to a file. If you have started spooling your output using `spool()`, “dumb” plots are included in the spooled output automatically. Alternatively, you can use `showplot()` to write GRAPH variable LASTPLOT to a file:

```
Cmd> showplot(file:"myplots.txt",dumb:T)
```

writes the plot in LASTPLOT to file `myplots.txt` as a “dumb” plot.

When a graphics window is the front window, **Copy** on the **Edit** menu (⌘C on a Macintosh or Ctrl+C in Windows or Linux) copies the graph to the Clipboard. You can then paste it into a word processor document or graphics editor using **Paste** on the **Edit** menu (⌘V on a Macintosh or Ctrl+V in Windows or Linux). This may not work with all word processors. If you also copy and paste the commands you used and any printed output, you can have a complete record of your session which can be saved to a file. This is the preferred way to use graphs in homework.

You can get a printed copy using **Print Window** or **Print Graph** on the **File** menu. This prints the front most graphics window. This is not recommended because you get a graph completely out of context.

In Classic Macintosh MacAnova, you can use `screendump: " "` as an argument to a plotting command. This saves a PICT format copy of the plot in a file. Or you can use **Save Graph As...** on the **File** menu to save a graph in PICT format. You can insert PICT files into most Macintosh word processor documents.

Writing plots as PostScript to a file. There is an additional way to save a graph in a file, although it requires some expertise to use the file.

```
Cmd> showplot(file:"myplots.ps")
```

writes a PostScript description of the plot in LASTPLOT to file `myplots.ps`. As of 9/7/05, this does not work in Carapace MacAnova. It should be fixed soon.

PostScript is a page description language that is understood by some printers. It is beyond the scope of this handout to explain in detail what you can do with the PostScript file when you leave MacAnova. On the School of Statistics Linux workstations, you can print such a file using command `lpr`. In addition, there exist programs such as `ghostview` for displaying PostScript and for translating it into various other graphics formats. It is possible to include PostScript directly into documents processed by some programs, including LaTeX.

Customizing plots using keywords You can use keyword phrases to control various aspects of plots such as labels and the minimum and maximum values for each axis.

Here is a summary of the optional keyword arguments common to all plotting commands:

```
title:"Plot title of your choice" (up to 75 characters)
xlab:"X-axis label"                (up to 50 characters)
ylab:"Y-axis label"                (up to 20 characters)
```

Graphics in MacAnova

<code>logx:T</code>	Use log scale for X-axis
<code>logy:T</code>	Use log scale for Y-axis
<code>xmin:xMinVal</code>	Minimum value for X-axis
<code>xmax:xMaxVal</code>	Maximum value for X-axis
<code>ymin:yMinVal</code>	Minimum value for Y-axis
<code>ymax:yMaxVal</code>	Maximum value for Y-axis
<code>xaxis:F</code>	Do not draw X-axis (line $y = 0$)
<code>yaxis:F</code>	Do not draw Y-axis (line $x = 0$)
<code>lines:T</code>	Connect points by lines
<code>symbols:charsyms</code>	Symbols to plot at each plotting position
<code>symbols:numbers</code>	Integers between 0 and 999 to plot at each plotting positions
<code>strings:Ch</code>	CHARACTER strings to draw at each plotting position (not with symbols)
<code>borders:sides</code>	Sides (for example "all" or "LB") where a graph border should be drawn
<code>ticks:sides</code>	Sides (for example "all" or "LB") where tick marks should be drawn
<code>xticks:RealVec</code>	Locations for x- or y-axis tick marks and labels. <code>xticks:?</code> and <code>yticks:?</code> mean compute from data. <code>xticks:NULL</code> and <code>yticks:NULL</code> mean no tick marks or labels.
<code>yticks:RealVec</code>	Length of x- or y-axis ticks, where $\text{length} \geq -1$ $\text{length} < 0$ means outside frame; $\text{length} > 2$ means full gridline.
<code>xticklen:length</code>	CHARACTER strings to label tick marks
<code>yticklen:length</code>	
<code>xticklab:charVec</code>	
<code>yticklab:charVec</code>	
<code>show:F</code>	Do not display plot, only save
<code>keep:F</code>	Do not save plot as LASTPLOT
<code>dumb:T</code>	Make low resolution plot that could be printed on a typewriter
<code>height:h</code>	Use h lines when printing a "dumb" plot.
<code>width:w</code>	Width of a "dumb" plot will be w characters.
<code>pause:T or pause:F</code>	Forces or suppresses a pause after each graph; <code>pause:F</code> is the Macintosh and Windows default; <code>pause:T</code> is the default in other versions.
<code>file:fileName</code>	Write plot file <code>fileName</code> , normally as PostScript commands. (On 09/07/05 does not work in Carapace MacAnova.)
<code>new:T</code>	Delete file <code>fileName</code> before writing

If you supply MISSING as a value for `xmin`, `xmax`, `ymin` or `ymax` (for instance, `xmin:?`) the minimum or maximum of all the data for that axis is used as value.

Graphics in MacAnova

On any command which draws lines, the following additional keyword phrases are recognized.

<code>linetype:n</code>	Use line types $n, n+1, \dots$ for successive columns of y , where $n > 0$ is an integer. The default is $n=1$, a solid line. What the line types are depends on the particular computer
<code>thickness:w</code>	Sets the line thickness to w times normal thickness, default is 1. w must be between .1 and 10. Has no effect with <code>dumb:T</code> or where otherwise not feasible.

The following are not defined for all versions of MacAnova

<code>window:n</code>	Draw plot in window $n, 0 \leq n \leq 24$. If n is 0, use window most recently used (only versions with graph windows). In Classic MacAnova, $0 \leq n \leq 8$.
<code>epsf:T</code>	Encapsulated PostScript file is written. You must also have <code>file:fileName</code> (only Macintosh Classic MacAnova)

Graphs are put in graphics windows named **Graph 1, Graph 2, ...**. You can have up to 24 such windows (8 in Classic MacAnova). A new plot normally goes in the lowest numbered available window. However, you can specify a specific window by keyword, for example `plot(x,y,window:3)`. You can use `window:0` to draw the graph in most recently used window.

There are up to 6 additional “panel” windows, **Panel 1, Panel 2, ... (Panel of Graphs 1-4 and Panel of Graphs 5-8** in Macintosh Classic MacAnova) each containing the contents of up to 4 regular graphics window in reduced form. Clicking on any of the small graphs, brings the corresponding full size graph to the front.

When a graphics window is in front, hitting **Return** or **Enter** brings the command window forward.

Each graphics window is listed on submenu **Graph Windows** on the **Windows** menu (listed directly on **Windows** menu in Classic Macintosh MacAnova). Selecting a window’s menu entry brings it to the front. You can use the keyboard to switch to graph window 2, say, by pressing **F2** (**⌘F2** or **⌘2** in Macintosh Classic MacAnova). In Macintosh Classic MacAnova, pressing **⌘G** displays one of the **Panel of Graphs** windows; pressing **⌘G** again toggles between them if there are two.

The size of a dumb graph is determined by the values of options `height` and `width` (which can be set by `setoptions()`), or by graphics keywords `height` and `width`; see above. In Carapace MacAnova, options `height` and `width` are preset to 25 (9/7/25 preset to 0 in error) and 80 and their values do not change when the window is resized. In Classic Macintosh MacAnova, options `height` and `width` are changed

Graphics in MacAnova

whenever you resize a command/output window so resizing a window changes the size of subsequent dumb plots.