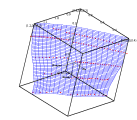
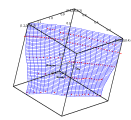
Agenda for today

Graphics!



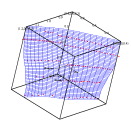
Graphics

```
plot (x1 {, x2 {, ... {x5}}})
      plots the three-dimensional data sets x1, ...,
        x5
gr = grsurface(x {, col})
      generates surface from the function $f(x,y)$
gr = grcontour2(x {, c {,col}})
      generates the contour lines $f(x,y)=c$
gr = grcontour3(x {, c {,col}})
      generates the contour lines $f(x,y,z)=c$
gr = grsunflower(x {, d {, o {,col}}})
      generates a sunflower plot
gr = grlinreg(x {,col})
      generates the linear regression line
gr = grlinreg2 (x {, n {,col}})
      generates the linear regression plane
```

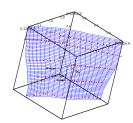
Graphics

```
plot2(x {, prep {,col}})
     plots two variables
plotstar (x {, prep {,col}})
     plots a star diagram
plotscml (x {, varnames})
     plots a scatter-plot matrix
plotandrews (x {, prep {,col}})
     plots Andrews curves
plotpcp (x {, prep {,col}})
    plots parallel coordinates
```

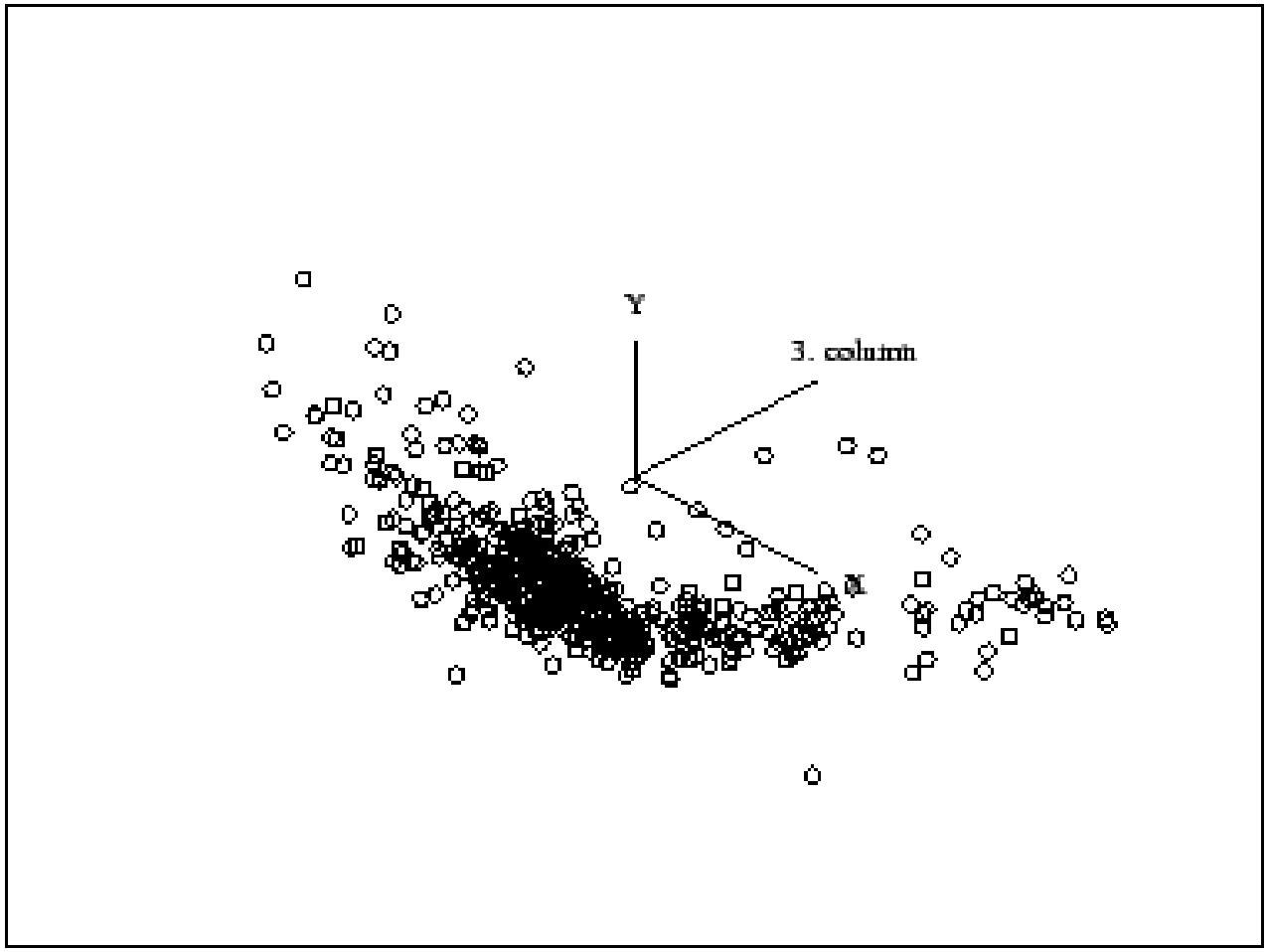



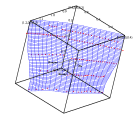
Graphics

```
1 library("plot")
2 data = read ("bostonh")
3 x = data[,6|13|14]
4 plot(x)
```

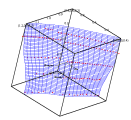
3D Scatterplot



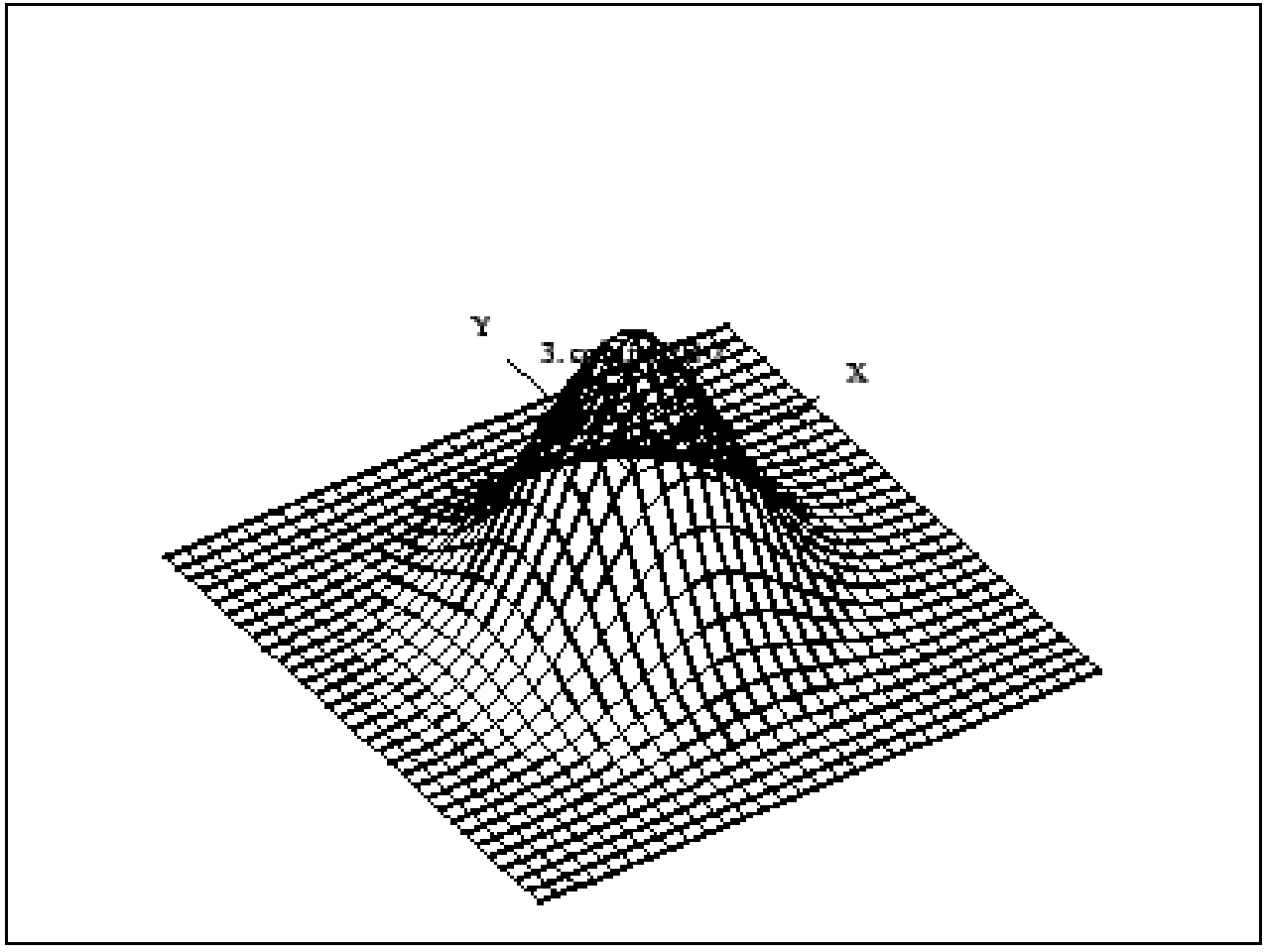


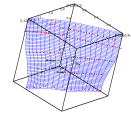
Graphics

```
1  library ("plot")
2  x0 = #(-3, -3)
3  h  = #(0.2, 0.2)
4  n  = #(31, 31)
5  x  = grid(x0, h, n)
6  f  = exp(-(x[,1]^2+x[,2]^2)/1.5)/(1.5*pi)
7  gr = grsurface(x~f)
8  plot(gr)
```

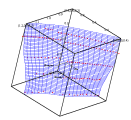
3D Surfaceplot



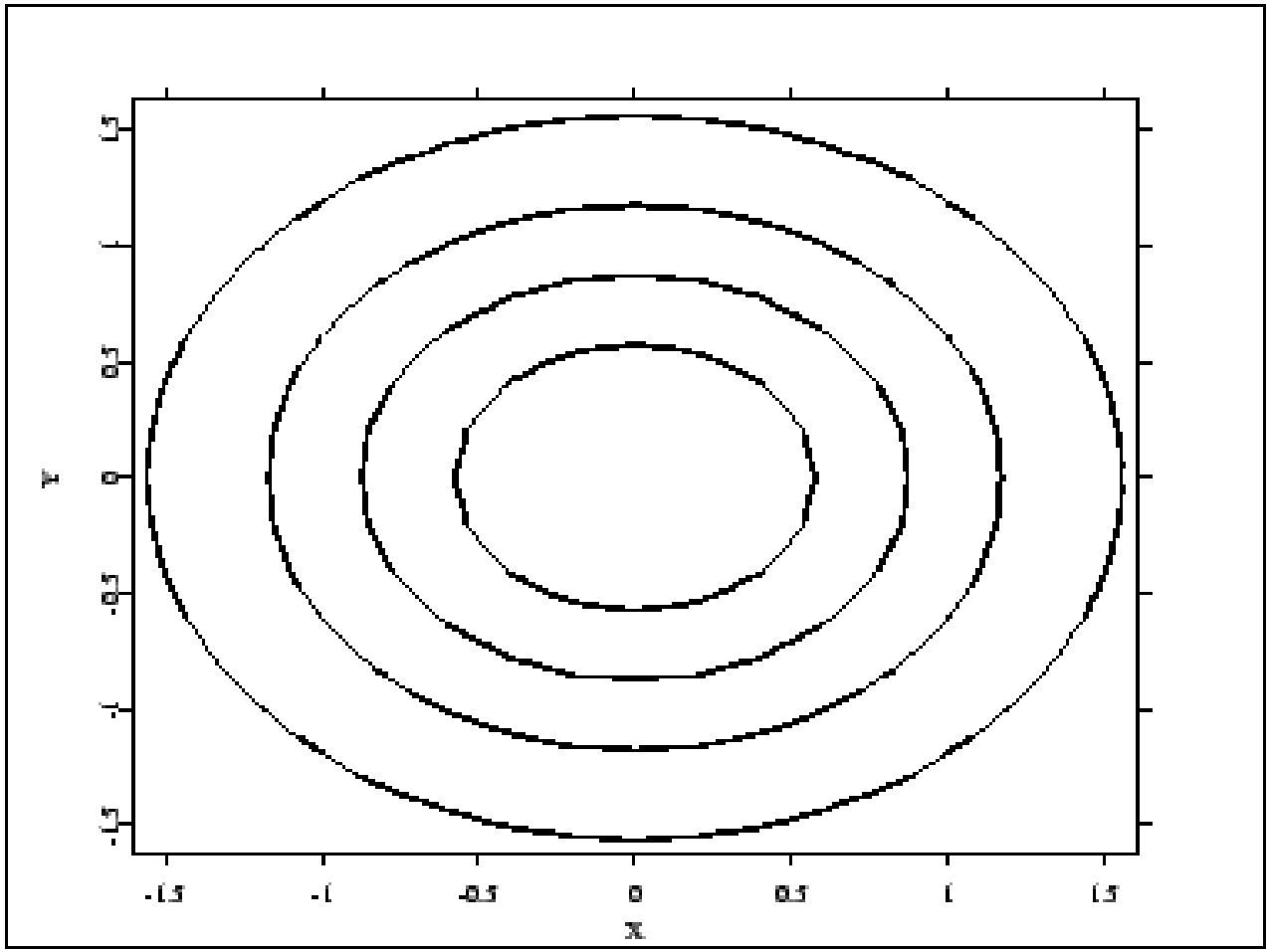


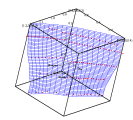
Graphics

```
1 library ("plot")
2 x0 = #(-3, -3)
3 h  = #(0.2, 0.2)
4 n  = #(31, 31)
5 x  = grid(x0, h, n)
6 f  = exp(-(x[,1]^2+x[,2]^2)/1.5)/(1.5*pi)
7 c  = 0.2*(1:4).*max(f)
8 gr = grcontour2(x~f, c)
9 plot(gr)
```

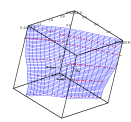
3D Contourplot



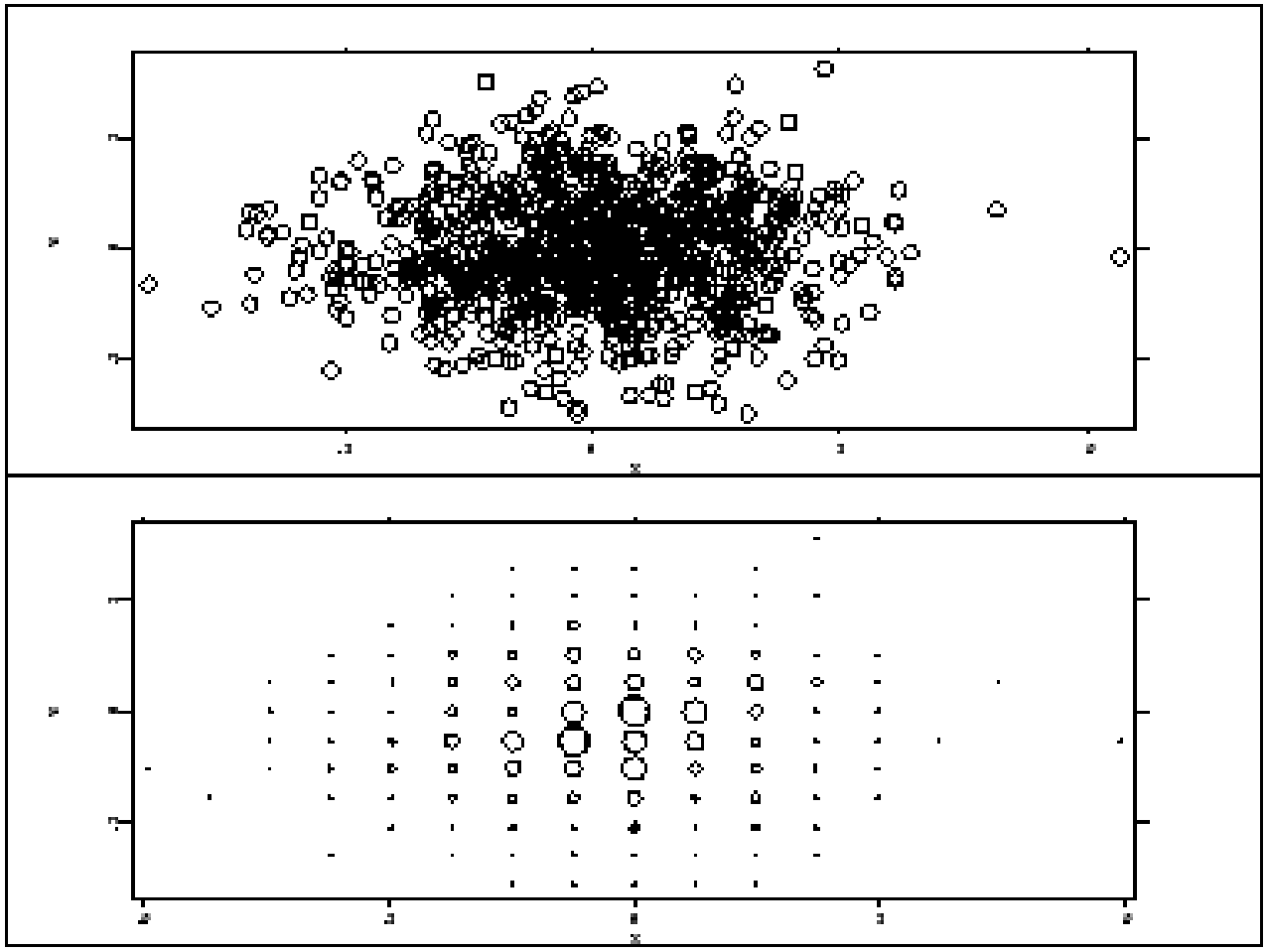


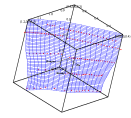
Graphics

```
1 library ("graphic")
2 x  = normal(1000, 2)
3 d  = createdisplay(2,1)
4 show (d, 1, 1, x)
5 gr = grsunflower(x)
6 show (d, 2, 1, gr)
```

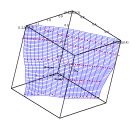
Sunflowerplot



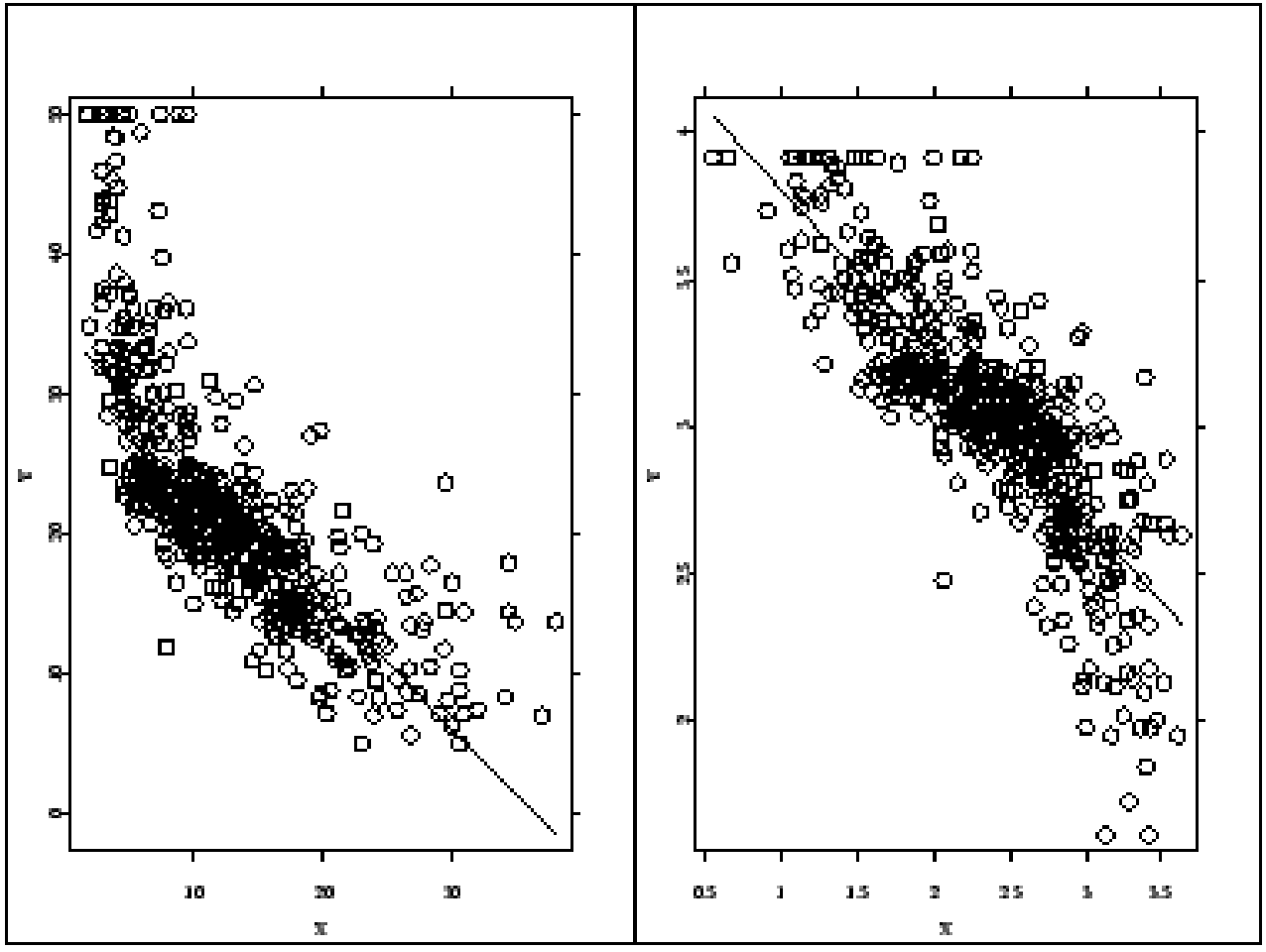


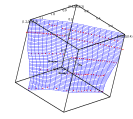
Graphics

```
1 library("plot")
2 data = read ("bostonh")
3 x0 = data[,13:14]
4 l0 = grlinreg(x0)
5 x1 = log(data[,13:14])
6 l1 = grlinreg(x1)
7 d  = createdisplay(1,2)
8 show (d, 1, 1, x0, l0)
9 show (d, 1, 2, x1, l1)
```

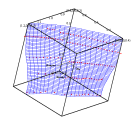
Linear Regression Plot



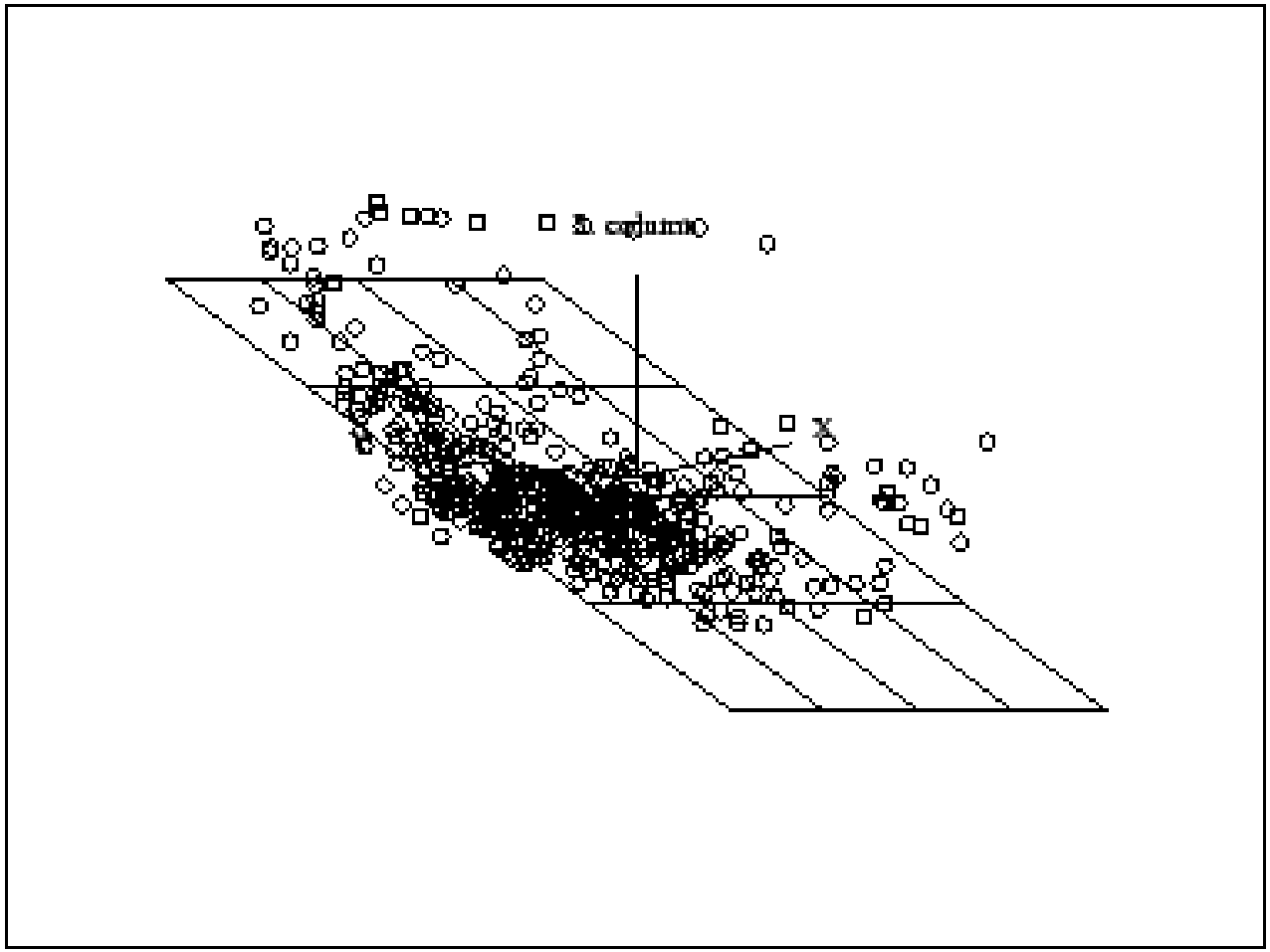


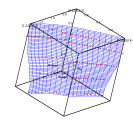
Graphics

```
1 library("plot")
2   data = read ("bostonh")
3   x   = data[,5|6|14]
4   p   = grlinreg2(x, 5|5)
5   plot(x, p)
```

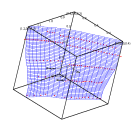
3D Regression Plot



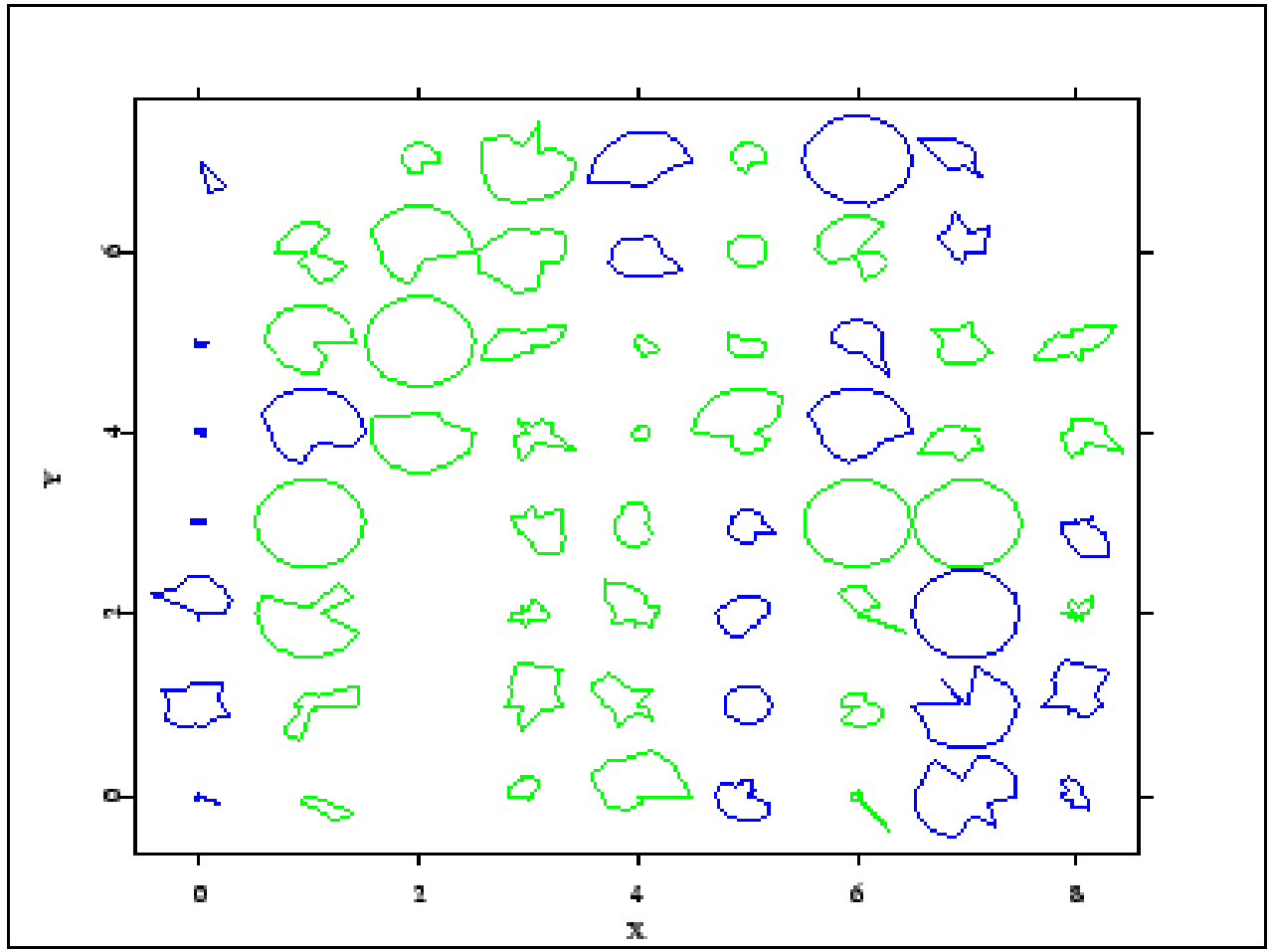


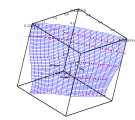
Graphics

```
1 library("plot")
2 data = read ("bostonh")
3 data = data[1:70,]
4 col  = grc.col.green-grc.col.blue
5 col  = grc.col.blue+col*(data[,14]<mean(data[,14]))
6 plotstar (data, grc.prep.zeroone, col)
```

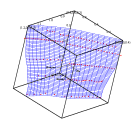
Starplot



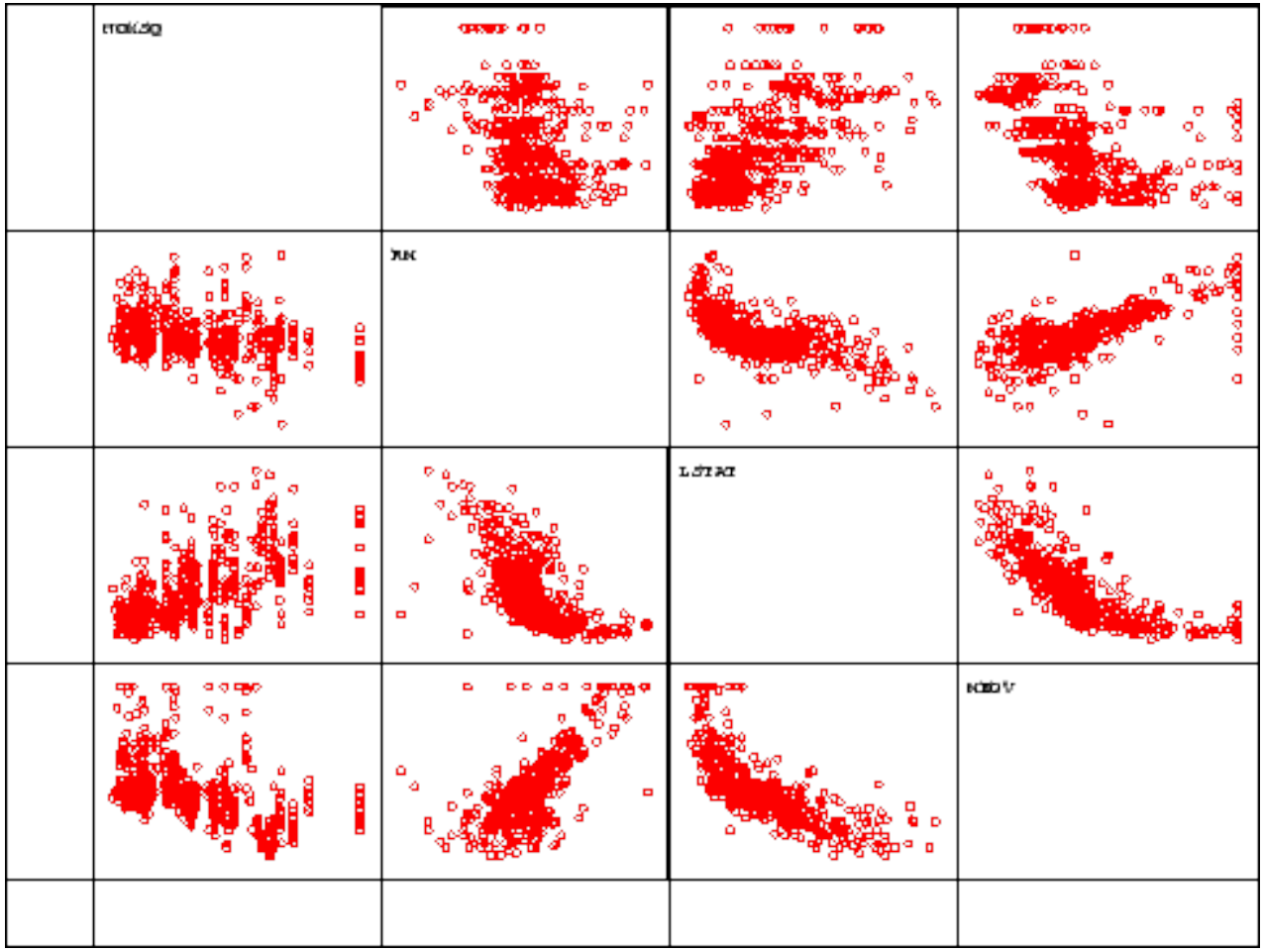


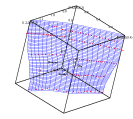
Scatterplot Matrices

```
library("plot")
data = read ("bostonh")
x = data[,5|6|13|14]
names="NOXSQ"~"RM"~"LSTAT"~"MEDV"
plotscml (x, names)
```

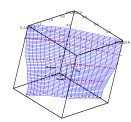
Scatterplot Matrix



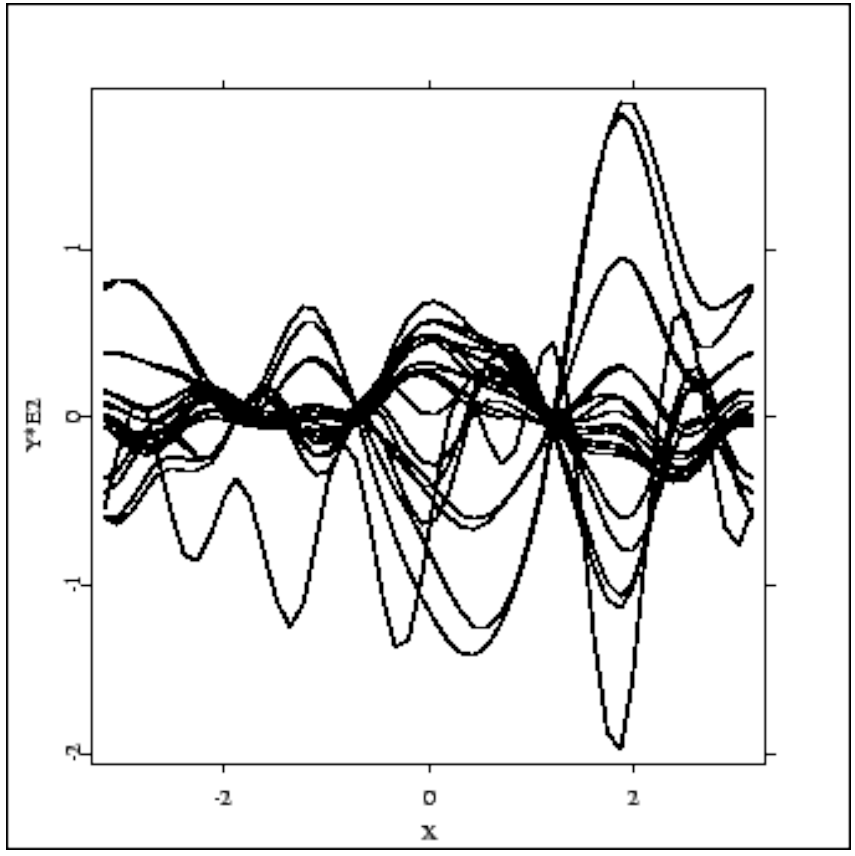


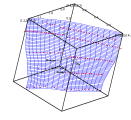
Andrews Curves

```
1  library("plot")
2  data = read ("bostonh")
3  data = data[,1:3]~data[,5:14]
4  data = data[21:40]
5  plotandrews (data, grc.prep.pcacorr)
```

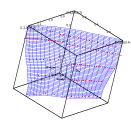
Andrews Curves



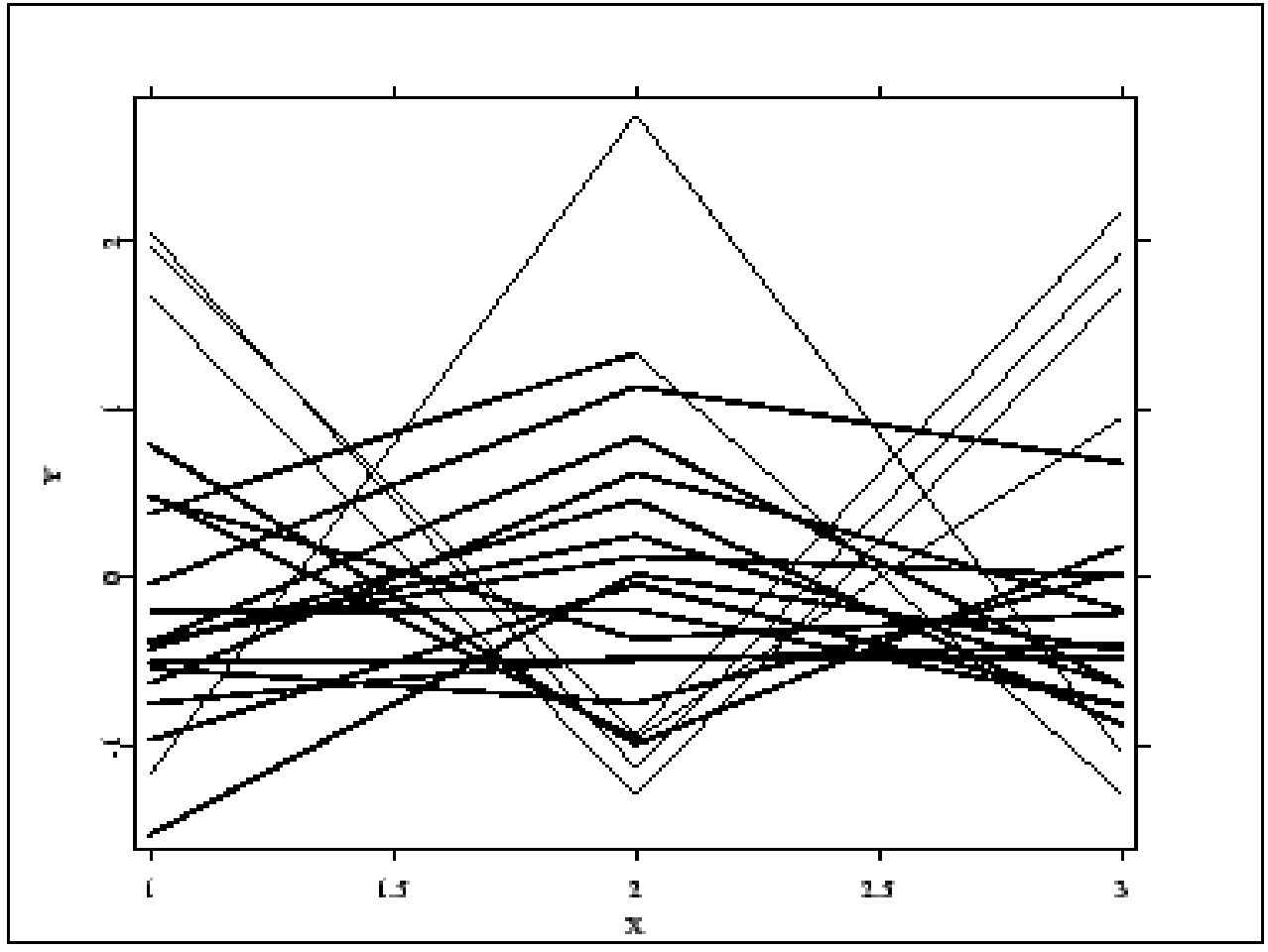


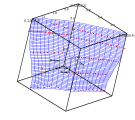
Parallel Coordinate Plots

```
1 library("plot")
2 data = read ("bostonh")
3 data = data[21:40]
4 x = data[,6|13|14]
5 plotpcp (x, grc.prep.standard)
```

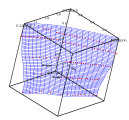
Parallel Coordinate Plot





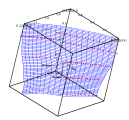
Graphics Primitives

```
grmove(grin, shf)
    moves a graphical object by shf
grrot(grin, rot)
    rotates a graphical object by rot times 90
      degree rotations
grxline(x, v {, col})
    generates a vertical line at v
gryline(y, v {, col})
    generates a horizontal line at v
```

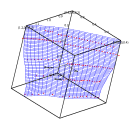
Graphics Primitives

```
grcircle (radius {, col})
    generates a circle or ellipse, centered at (0,0)
hls2rgb(hls)
    generates RGB colors from the HLS color model
rgb2hls (rgb)
    generates HLS colors from the RGB color model
createcolor(rgb)
    sets a palette of colors
```

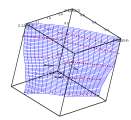
Graphics Primitives

```
1   library("plot")
2   data = read ("bostonh")
3   gro1 = grbox (data[,11])
4   gro2 = grbox (data[,13])
5   gro3 = grbox (data[,14])
6   gro1 = grrot (gro1, 1)
7   gro2 = grrot (gro2, 1)
8   gro2 = grmove (gro2, #(1.5,0))
9   gro3 = grrot (gro3, 1)
10  gro3 = grmove (gro3, #(3,0))
11  plot(gro1, gro2, gro3)
```

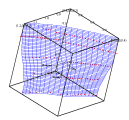
Graphics Primitives

```
1  library("graphic")
2  randomize(0)
3  x  = normal(200,2)
4  xl = grxline(0, x[,1])
5  yl = gryline(0, x[,2])
6  cl = grcircle(2.44775)
7  d  = createdisplay(1,1)
8  show(d,1,1,x,xl,yl,cl)
```

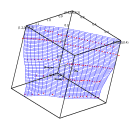
setmask{p,l,t} and setgopt

```
1 setmaskp (data, color, layout, size)
2     influences the layout, size and color of data
        points
3 setmaskl (data, lines, color, type, thickness)
4     influences the layout, size, type and color
        between data points
5 setmaskt (data, labels, color, direction, size)
6     influences the appearance of text at the data
        points
7 setgopt (d,row,col, optname,optval,..., optnameN,
    optvalN)
8     influences several parameters of plots and
        displays
```

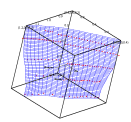
Controlling Data Points

```
x   = 1:100
y   = sin(x/20)+uniform(100)/5
data = x~y
setmaskp(data, 4, 3, 8)
d = createdisplay(1, 1)
show(d, 1, 1, data)
;  data = setmask(data, "red", "medium", "circle")
```

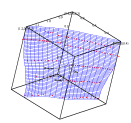
Controlling Data Points

```
x   = 1:100
y   = sin(x/20)+uniform(100)/5
data = x~y
color = 4*matrix(50)|5*matrix(50)
setmaskp(data, color, 3, 8)
d = createdisplay(1, 1)
show(d, 1, 1, data)
```

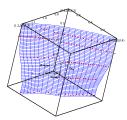
Controlling Data Points

```
x       = 1:100
y       = sin(x/20) +uniform(100, 1) /10
data    = x~y
color   = 4*matrix(50)|5*matrix(50)
layout  = 3*matrix(25)|4*matrix(25)|5*matrix(25)|6*
  matrix(25)
setmaskp(data, color, layout, 8)
d       = createdisplay(1, 1)
show(d, 1, 1, data)
```

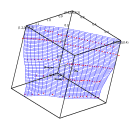
Controlling Data Points - Alternative

```
1  library("plot")
2  x        = 1:100
3  y        = sin(x/20) +uniform(100, 1) /10
4  data     = x~y
5  mycolor  = string("red", 1:50)|string("magenta",
     51:100)
6  mystyle  = string("circle", 1:25)|string("triangle",
     26:50)
7  mystyle  = mystyle|string("xsymbol",51:75)
8  mystyle  = mystyle|string("rhomb",76:100)
9  data     = setmask(data, mycolor, mystyle)
10 d        = createdisplay(1, 1)
11 show(d, 1, 1, data)
```

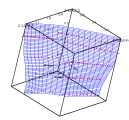
Controlling Data Points - Size

```
x       = 1:100
y       = sin(x/20) + uniform(100, 1)/10
data    = x~y
color   = 4*matrix(50)|5*matrix(50)
layout  = 3*matrix(25)|4*matrix(25)|5*matrix(25)|6*
  matrix(25)
size    = 4*matrix(50)|15*matrix(50)
setmaskp(data, color, layout, size)
d       = createdisplay(1, 1)
show(d, 1, 1, data)
```

Controlling Data Points - Alternative

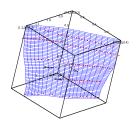
```
1  library("plot")
2  x        = 1:100
3  y        = sin(x/20) + uniform(100, 1)/10
4  data     = x~y
5  mycolor  = string("red", 1:50)|string("magenta",
     51:100)
6  mystyle  = string("circle", 1:25)|string("triangle",
     26:50)
7  mystyle  = mystyle|string("xsymbol" ,51:75)
8  mystyle  = mystyle|string("rhomb", 76:100)
9  mysize   = string("small", 1:50)|string("huge",
     51:100)
10 data     = setmask(data, mycolor, mystyle, mysize)
11 d        = createdisplay(1, 1)
12 show(d, 1, 1, data)
```

Controlling Data Points

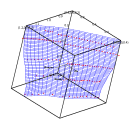
Connection of Data Points (setmaskl)

```
randomize(666)
n = 6
x = 4:(3+n)
y = 2*x+normal(n)
z = x~y
d = createdisplay(1, 1)
pm    = (1:n)'
color = 1
art   = 1
thick = 5
setmaskl(z, pm, color ,art ,thick)
show(d, 1, 1, z)
```

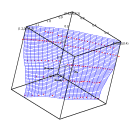
Controlling Data Points - Alternative

```
1 library("plot")
2 randomize(666)
3 n = 6
4 x = 4:(3+n)
5 y = 2 *x +normal(n)
6 z = x~y
7 z = setmask(z, "line", "blue", "thick", "solid")
8 show(d, 1, 1, z)
```

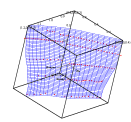
Controlling Data Points - Labels

```
x        = 1:6
x        = x~x
text     = "Point1"|"Point2"|"Point3"|"Point4"|"Point5"
  |"Point6"
color    = 1
position = 3
size     = 16
setmaskt(x, text, color, position, size)
d        = createdisplay(1, 1)
show(d, 1, 1, x)
```

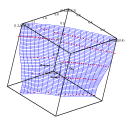
Label Alternative

```
1 library("plot")
2 x = 1:6
3 x = x~x
4 mytext = "Point1"|"Point2"|"Point3"|"Point4"|"Point5
    "|"Point6"
5 x = setmask(x, "points","text",mytext,"blue","right"
    ,"medium")
6 d = createdisplay(1, 1)
7 show(d, 1, 1, x)
```

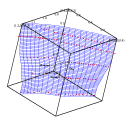
Label Alternative

```
x     = 1:6
x     = x~x
text      = "Right"|"Under"|"Left"|"Over"|"Center"|"
  No"
color     = 1|2|3|4|5|6
position  = 3|6|9|12|0|(-1)
size      = 12|13|14|15|16|16
setmaskt(x, text, color,position, size)
d     = createdisplay(1, 1)
show(d, 1, 1, x)
```

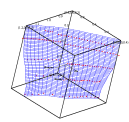
Label Alternative

```
x = 1:6
x = x~x
mytext   = "Right"|"Under"|"Left"|"Over"|"Center"|"No
  "
mycolor = "blue"|"green"|"cyan"|"red"|"magenta"|"
  yellow"
mypos    = "right"|"below"|"left"|"above"|"center"|"
  request"
mysize   = "small"|"medium"|"medium"|"large"|"large"|
  "huge"
x = setmask(x, "points", "text",mytext,mycolor,
  mysize,mypos)
d = createdisplay(1, 1)
show(d, 1, 1, x)
```

Title and Axes Labels

```
x     = 1:100
y     = sqrt(x)
data  = x~y
d     = createdisplay(1, 1)
show(d, 1, 1, data)
title  = "Plot of Sqrt(x)"
ylabel = "y = sqrt(x)"
setgopt(d, 1, 1,"title",title, "xlabel","x", "ylabel
  ",ylabel)
```

Axes

xlim/ylim change limits of x/y-axis

xoffset/yoffset change the width of axis border

xvalue/yvalue transformation $m + k \cdot x$

xorigin/yorigin change origin for tickmark

xmajor/ymajor change major for tickmark

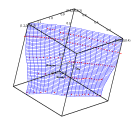
xlabel/ylabel change labels

rotpoint change rotation point

rotcos change rotation matrix

scal change scale matrix

transl change translation vector



Axes

```
1 disp = createdisplay(1,1)
```