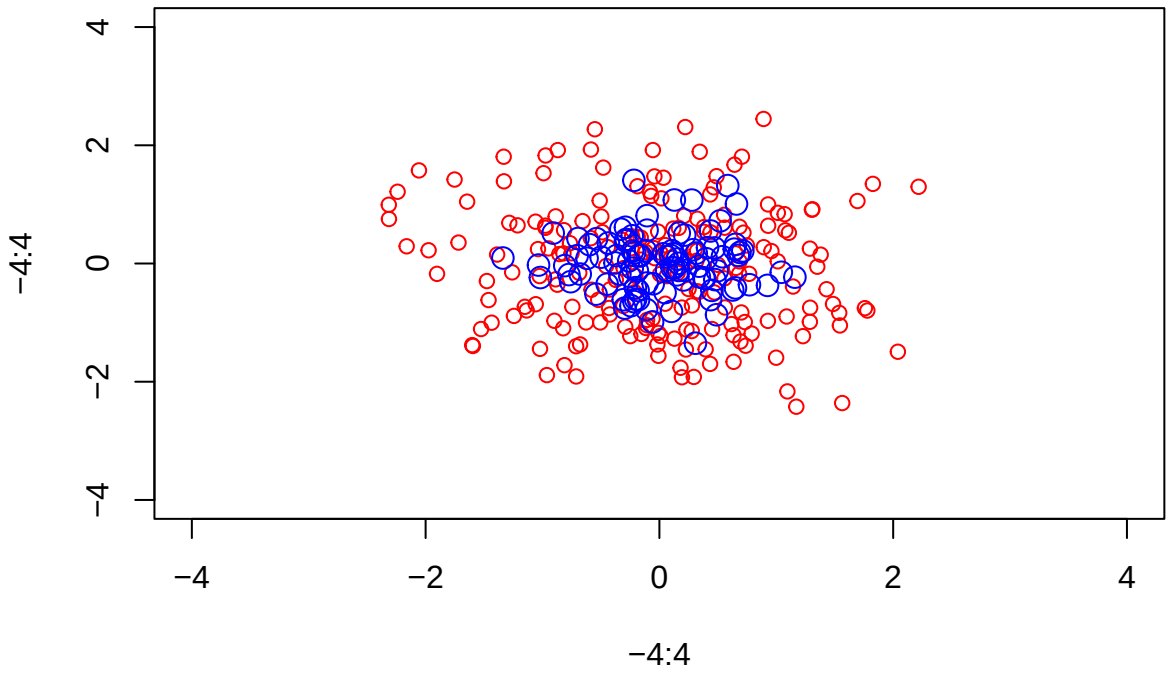
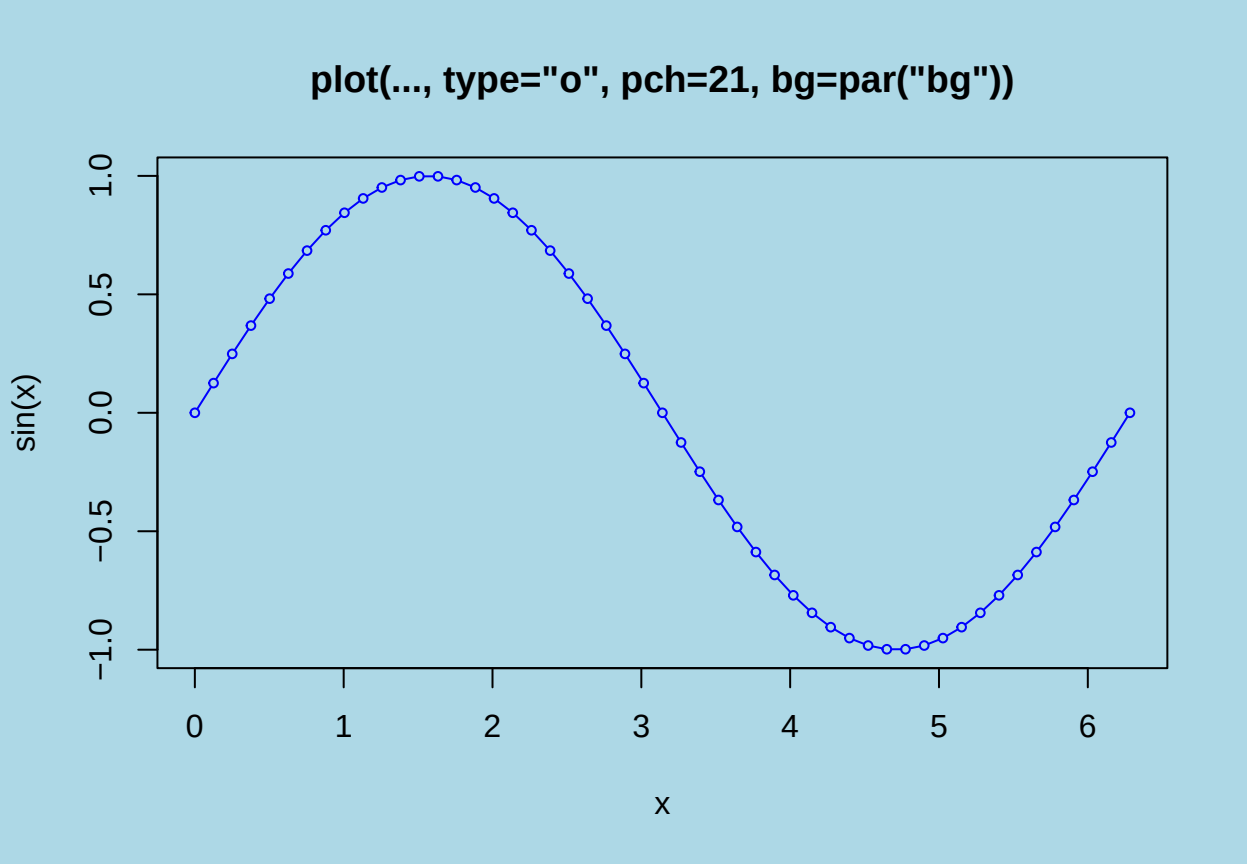
SCATTER

```
example(points)
```

```
##
## points> require(stats) # for rnorm
##
## points> plot(-4:4, -4:4, type = "n")  # setting up coord. system
```

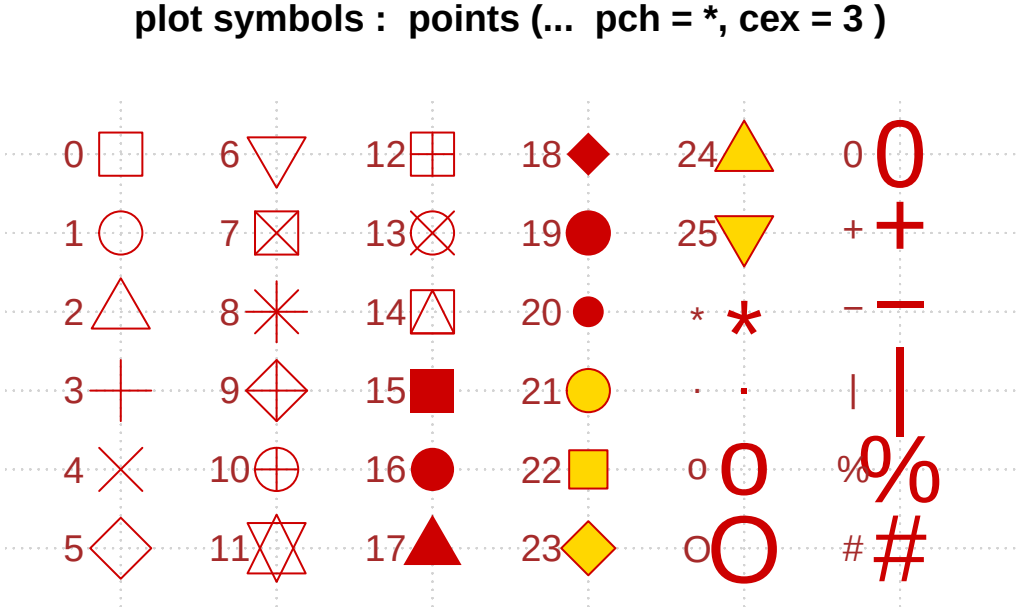



```
##
## points> points(rnorm(200), rnorm(200), col = "red")
##
## points> points(rnorm(100)/2, rnorm(100)/2, col = "blue", cex = 1.5)
##
## points> op <- par(bg = "light blue")
##
## points> x <- seq(0, 2*pi, length.out = 51)
##
## points> ## something "between type='b' and type='o'":
## points> plot(x, sin(x), type = "o", pch = 21, bg = par("bg"), col = "blue", cex = .6,
## points+  main = 'plot(..., type="o", pch=21, bg=par("bg"))')
```

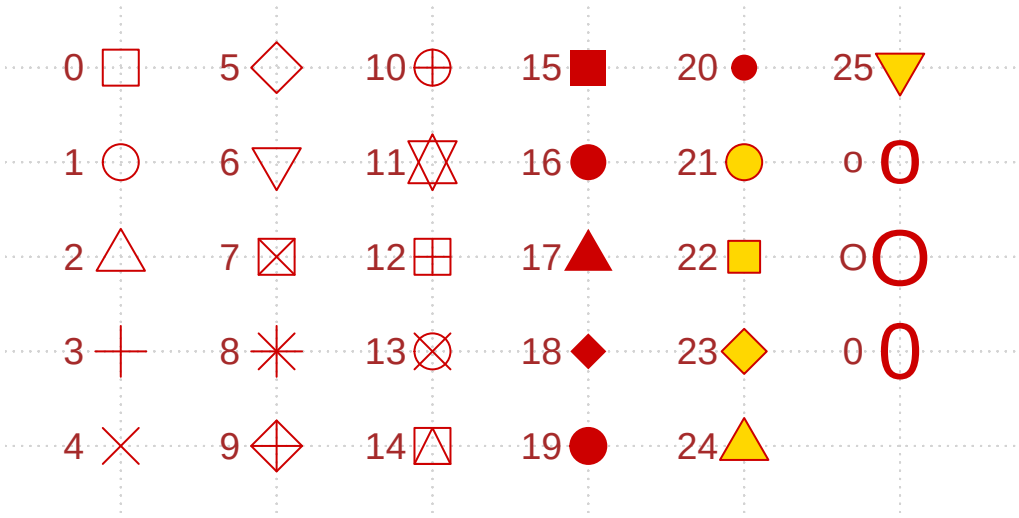



```
##
## points> par(op)
##
## points> ## Not run:
## points> ##D ## The figure was produced by calls like
## points> ##D png("pch.png", height = 0.7, width = 7, res = 100, units = "in")
## points> ##D par(mar = rep(0,4))
## points> ##D plot(c(-1, 26), 0:1, type = "n", axes = FALSE)
## points> ##D text(0:25, 0.6, 0:25, cex = 0.5)
## points> ##D points(0:25, rep(0.3, 26), pch = 0:25, bg = "grey")
## points> ## End(Not run)
## points>
## points> ##-------- Showing all the extra & some char graphics symbols ---------
## points> pchShow <-
## points+   function(extras = c("*",".", "o","O","0","+","-","|","%","#"),
## points+            cex = 3, ## good for both .Device=="postscript" and "x11"
## points+            col = "red3", bg = "gold", coltext = "brown", cextext = 1.2,
## points+            main = paste("plot symbols :  points (...  pch = *, cex =",
## points+                         cex,")"))
## points+   {
## points+     nex <- length(extras)
## points+     np  <- 26 + nex
## points+     ipch <- 0:(np-1)
## points+     k <- floor(sqrt(np))
## points+     dd <- c(-1,1)/2
## points+     rx <- dd + range(ix <- ipch %/% k)
```

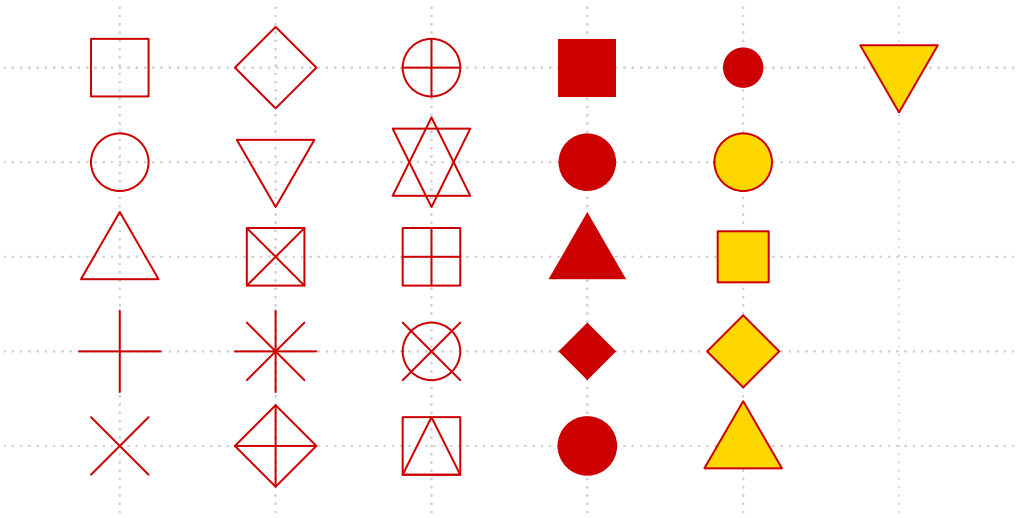
```
## points+     ry <- dd + range(iy <- 3 + (k-1)- ipch %% k)
## points+     pch <- as.list(ipch) # list with integers & strings
## points+     if(nex > 0) pch[26+ 1:nex] <- as.list(extras)
## points+     plot(rx, ry, type = "n", axes  =  FALSE, xlab = "", ylab = "", main = main)
## points+     abline(v = ix, h = iy, col = "lightgray", lty = "dotted")
## points+     for(i in 1:np) {
## points+       pc <- pch[[i]]
## points+       ## 'col' symbols with a 'bg'-colored interior (where available) :
## points+       points(ix[i], iy[i], pch = pc, col = col, bg = bg, cex = cex)
## points+       if(cextext > 0)
## points+           text(ix[i] - 0.3, iy[i], pc, col = coltext, cex = cextext)
## points+     }
## points+   }
##
## points> pchShow()
```




```
##
## points> pchShow(c("o","O","0"), cex = 2.5)
```




```
##
## points> pchShow(NULL, cex = 4, cextext = 0, main = NULL)
```

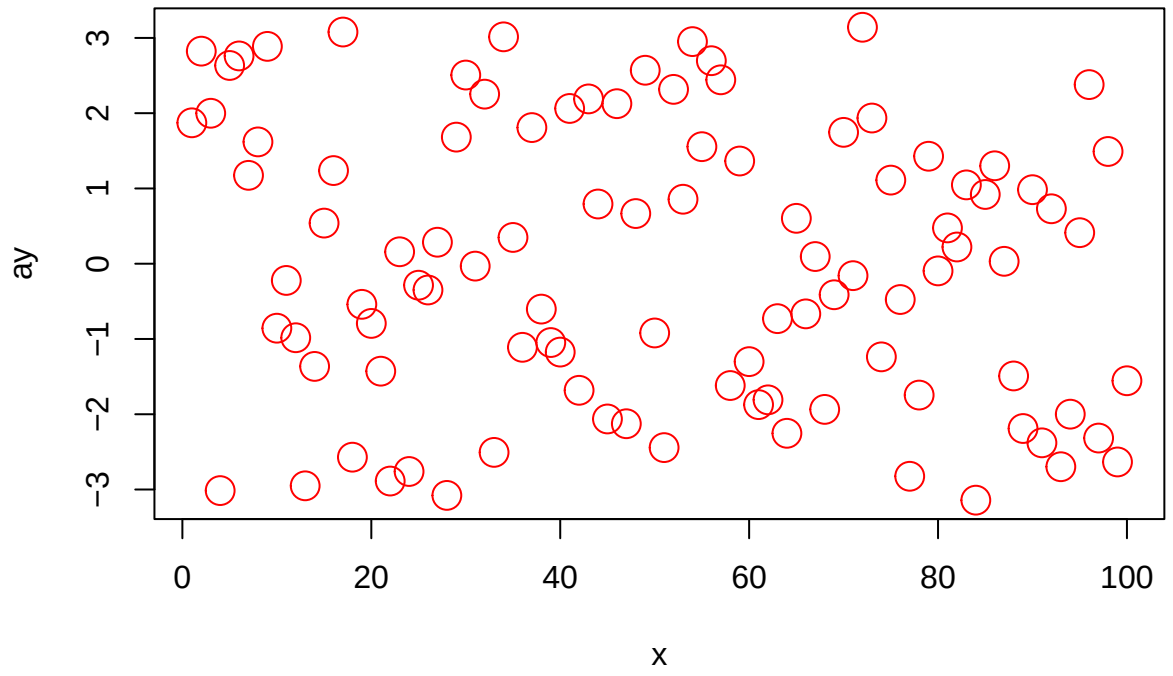


```
##
## points> ## No test:
## points> ##D ## ------------ test code for various pch specifications -------------
## points> ##D # Try this in various font families (including Hershey)
## points> ##D # and locales.  Use sign = -1 asserts we want Latin-1.
## points> ##D # Standard cases in a MBCS locale will not plot the top half.
## points> ##D TestChars <- function(sign = 1, font = 1, ...)
## points> ##D {
## points> ##D    MB <- l10n_info()$MBCS
## points> ##D    r <- if(font == 5) { sign <- 1; c(32:126, 160:254)
## points> ##D        } else if(MB) 32:126 else 32:255
## points> ##D    if (sign == -1) r <- c(32:126, 160:255)
## points> ##D    par(pty = "s")
## points> ##D    plot(c(-1,16), c(-1,16), type = "n", xlab = "", ylab = "",
## points> ##D         xaxs = "i", yaxs = "i",
## points> ##D         main = sprintf("sign = %d, font = %d", sign, font))
## points> ##D    grid(17, 17, lty = 1) ; mtext(paste("MBCS:", MB))
## points> ##D    for(i in r) try(points(i%%16, i%/%16, pch = sign*i, font = font,...))
## points> ##D }
## points> ##D TestChars()
## points> ##D try(TestChars(sign = -1))
## points> ##D TestChars(font = 5)  # Euro might be at 160 (0+10*16).
## points> ##D                      # macOS has apple at 240 (0+15*16).
## points> ##D try(TestChars(-1, font = 2))  # bold
## points> ## End(No test)
## points>
```

```
## points>
```

```
x <- c(1:100)
y <- seq(-pi, pi, length.out = 100)
ay <- sample(y, 100, replace = FALSE)

plot(x, ay, col = "red", cex = 2)
```

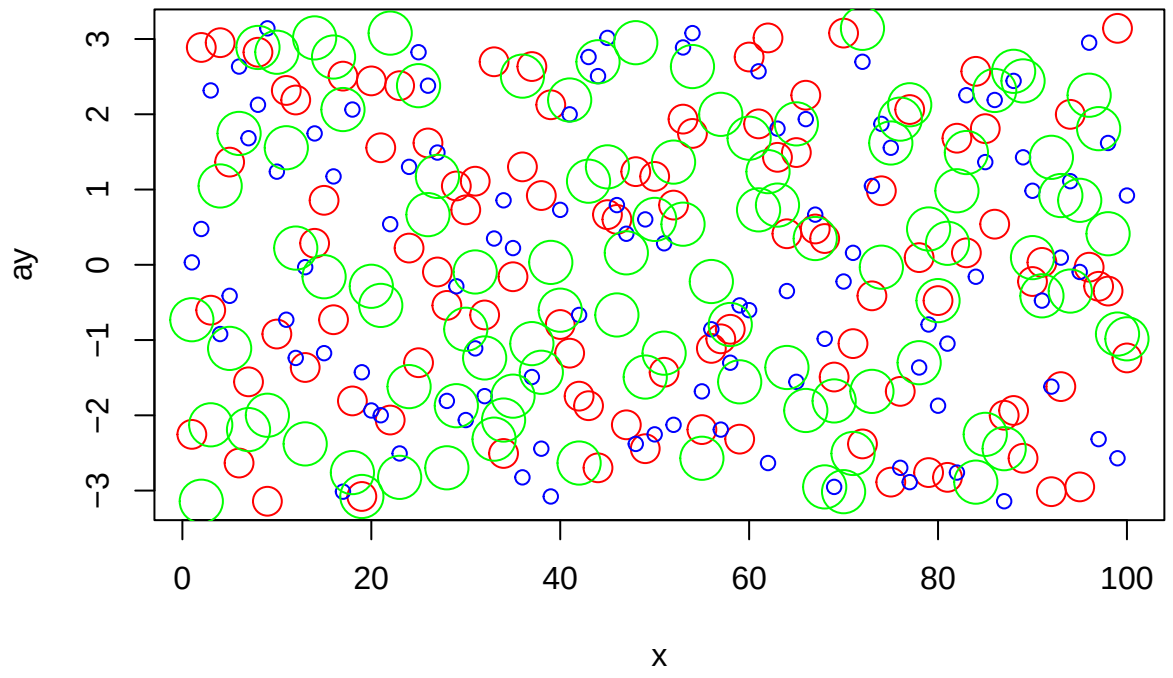



```
x <- c(1:100)
y <- seq(-pi, pi, length.out = 100)
ay <- sample(y, 100, replace = FALSE)

by <- sample(y, 100, replace = FALSE)

cy <- sample(y, 100, replace = FALSE)

plot(x, ay, col = "red", cex = 2)
points(x, by, col = "blue", cex = 1)
points(x, cy, col = "green", cex = 3)
```

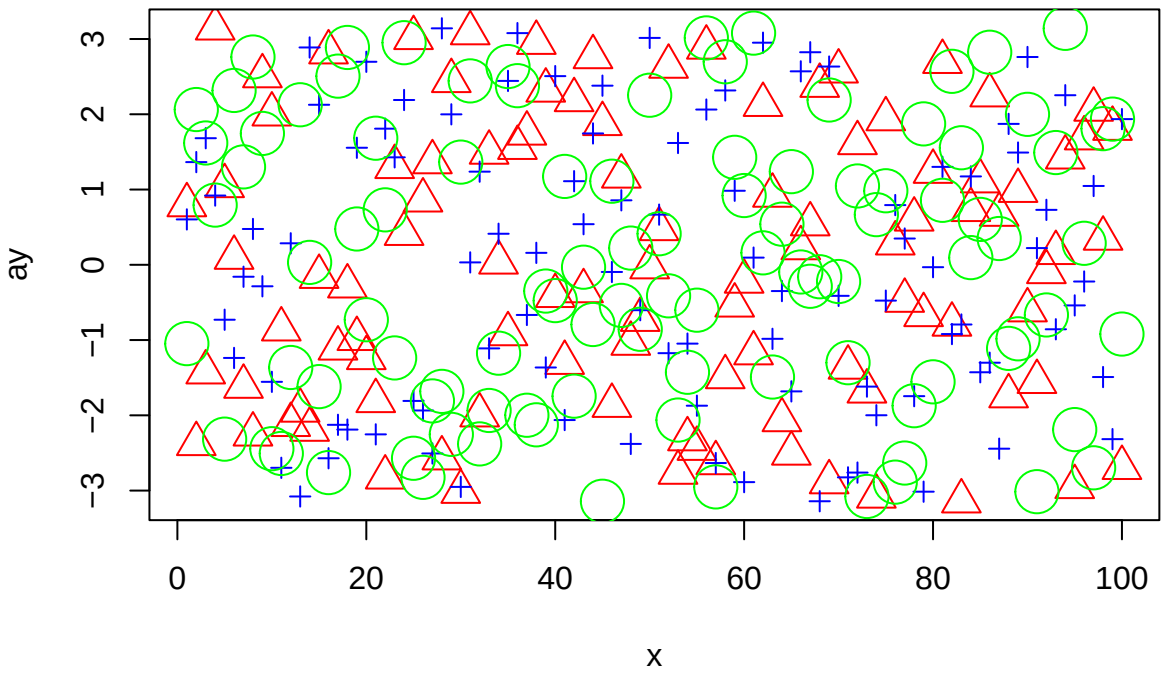



```
x <- c(1:100)
y <- seq(-pi, pi, length.out = 100)
ay <- sample(y, 100, replace = FALSE)

by <- sample(y, 100, replace = FALSE)

cy <- sample(y, 100, replace = FALSE)

plot(x, ay, col = "red", cex = 2, pch = 2)
points(x, by, col = "blue", cex = 1, pch = 3)
points(x, cy, col = "green", cex = 3)
```


ay
3
2
1
0
−1
−2
−3
0
20
40
60
80
100
x