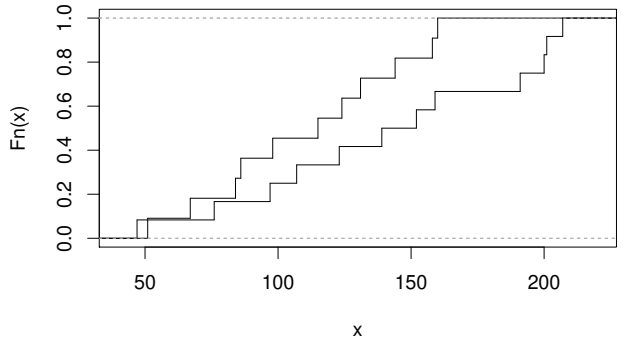
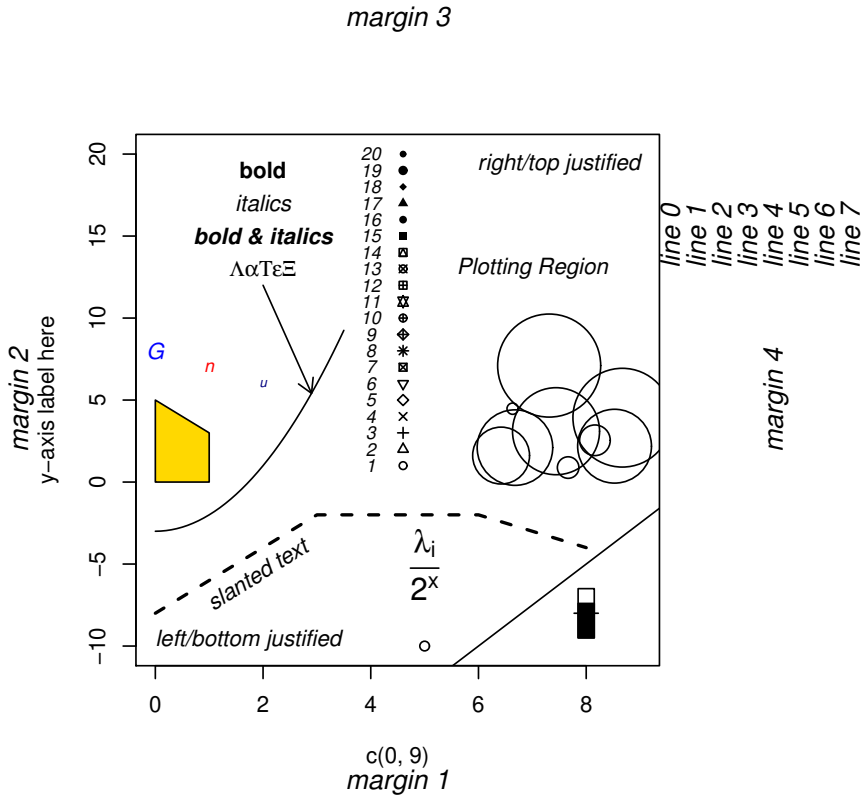


to `par` or reset to their default values by opening a new graphics device with the `dev.new` function. Graphical parameters set by other graphical functions (such as `plot`, `lines`, `points`, or `text`) hold temporary values that are only in effect during the call to the function.

The first R command below calls the `par` function with the `mar` (margin) argument, which controls the sizes of the four margins—bottom, left, top, and right, in that order—measured in number of lines of text. This ordering of the margins is used by other R graphical functions. The second R command calls the `plot` function, which would ordinarily plot the points $(0,-10)$ and $(9,20)$, but setting the `type` argument to the string `"n"` means that no points will be plotted. This sets the axes. The `text` function is used to plot the string `"Plotting Region"` centered at the point $(7,13)$. The `points` function is called to plot the default style point at $(5,-10)$. The first 20 integers are plotted with the `text` function, followed by the first 20 symbols with the `points` function. The `points` function then prints three different characters in three different colors (`col`) and in three different sizes (`cex` for character expand). The `polygon` function plots a shaded polygon with the points associated with the x-values given in the first argument and the y-values given in the second argument. The `lines` function plots a line by connecting the points associated with the x-values given in the first argument and the y-values given in the second argument, with the `lwd` (line width) argument specifying the thickness of the line and the `lty` (line type) argument specifying the type of line (1 for solid, 2 for dashed, 3 for dotted, etc.). The `srt` (string rotate) argument in the `text` function allows for slanted text (the argument is in degrees). The `mtext` (margin text) function places text in the margins. The `font` argument in the `text` function can be used for bold text (`font = 2`), italics text (`font = 3`), bold italix text (`font = 4`), and Greek text (`font = 5`). The `adj` (adjust) argument in the `text` function can assume any value on the interval $[0,1]$, and specifies whether the text should be left justified (`adj = 0`), centered (`adj = 0.5`, the default), or right justified (`adj = 1`). The `expression` function can be used for plotting mathematical expressions like $\lambda_i/2^x$. The `lines` function plots the parabola $y = x^2 - 3$, and the `arrows` function draws an arrow from $(2,12)$ to a point on the parabola. The `abline` function plots a line with specified intercept and slope. Finally, the `symbols` function is used to plot nine circles and a thermometer.

```
> par(mar = c(5, 8, 5, 8))
> plot(c(0, 9), c(-10, 20), type = "n", ylab = "y-axis label here")
> text(7, 13, "Plotting Region")
> points(5, -10)
> text(rep(4, 20), 1:20, 1:20, cex = 0.8)
> points(rep(4.6, 20), 1:20, pch = 1:20, cex = 0.8)
> points(0:2, 8:6, pch = c("G", "n", "u"), col = c("blue", "red", "navy"),
+        cex = seq(1.0, 0.6, by = -0.2))
> polygon(c(0, 0, 1, 1), c(0, 5, 3, 0), col = "gold")
```

```
> lines(c(0, 3, 6, 8), c(-8, -2, -2, -4), lwd = 2, lty = 2)
> text(1.9, -5.6, "slanted text", srt = 32)
> mtext(paste("margin", 1:4), side = 1:4, line = 4)
> mtext(paste("line", 0:7), side = 4, line = 0:7, at = 15)
> text(rep(2, 4), c(19, 17, 15, 13),
+      c("bold", "italics", "bold & italics", "LaTeX"), font = 2:5)
> text(0, -10, "left/bottom justified", adj = c(0, 0))
> text(9, 20, "right/top justified", adj = c(1, 1))
> text(5, -5, expression(frac(lambda[i], 2 ^ x)), cex = 1.5)
> x = seq(0, 3.5, by = 0.1)
> y = x ^ 2 - 3
> lines(x, y)
> arrows(2, 12, x[30], y[30], length = 0.1)
> abline(-25, 2.5)
> symbols(runif(9, 6, 9), 8 * runif(9), circles = runif(9), add=T, inches=F)
> symbols(8, -8, thermometers = matrix(c(.3, 3, .7), 1, 3), add=T, inches=F)
```



The `plot` function can also be used for plotting univariate data. The R command

```
> plot(precip)           # plot precipitation data
```