How to Create a Function in R

How to Step Through Debugging an R Function

How to Extend Text Functionality with Stringr in R

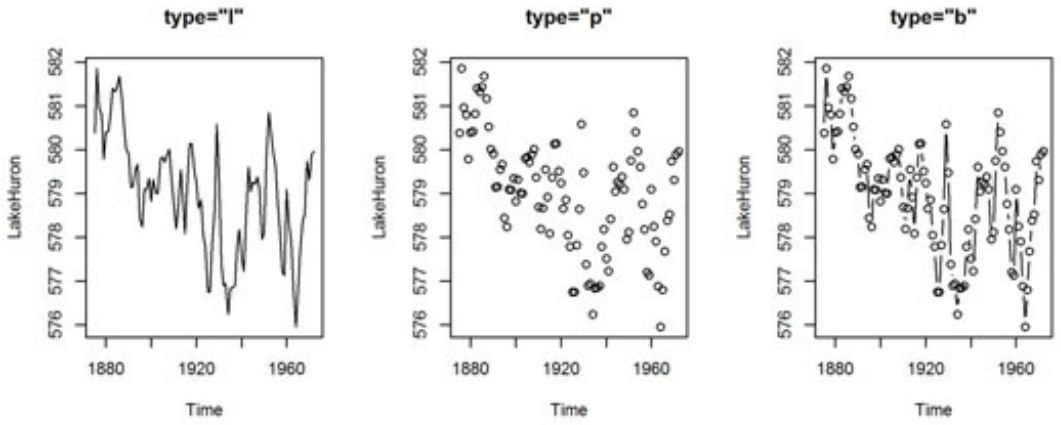
How to Create Different Plot Types in R

By Andrie de Vries and Joris Meys from R For Dummies

The plot function in R has a `type` argument that controls the type of plot that gets drawn. For example, to create a plot with lines between data points, use `type="l"`; to plot only the points, use `type="p"`; and to draw both lines and points, use `type="b"`:

```
> plot(LakeHuron, type="l", main='type="l"')
> plot(LakeHuron, type="p", main='type=p"')
> plot(LakeHuron, type="b", main='type="b"')
```

The plot with lines only is on the left, the plot with points is in the middle, and the plot with both lines and points is on the right.



The Help page for `plot()` has a list of all the different types that you can use with the `type` argument:

- **`"p"`:** Points
- **`"l"`:** Lines
- **`"b"`:** Both
- **`"c"`:** The lines part alone of `"b"`
- **`"o"`:** Both “overplotted”
- **`"h"`:** Histogram like (or high-density) vertical lines
- **`"n"`:** No plotting

    It seems odd to use a plot function and then tell R not to plot it. But this can be very useful when you need to create just the titles and axes, and plot the data later using `points()`, `lines()`, or any of the other graphical functions.

This flexibility may be useful if you want to build a plot step by step (for example, for presentations or documents). Here’s an example:

```
> x <- seq(0.5, 1.5, 0.25)
> y <- rep(1, length(x))
> plot(x, y, type="n")
> points(x, y)
```

Aside from `plot()`, which gives you tremendous flexibility in creating your own plots, R also provides a variety of functions to make specific types of plots. Here are a few to explore:

- **Scatterplot:** If you pass two numeric vectors as arguments to `plot()`, the result is a scatterplot. Try:

  ```
  > with(mtcars, plot(mpg, disp))
  ```

- **Box-and-whisker plot:** Use the `boxplot()` function:

  ```
  > with(mtcars, boxplot(disp, mpg))
  ```

- **Histogram:** A histogram plots the frequency of observations. Use the `hist()` function:

  ```
  > with(mtcars, hist(mpg))
  ```

- **Matrix of scatterplots:** The `pairs()` function is useful in data exploration, because it plots a matrix of scatterplots. Each variable gets plotted against another.

```
> pairs(iris)
```