

8 Feb 2013

eBay-Google Bayesian short course

How to plot multiple functions on the same graph in R

Suppose (for example) that you want to plot three Gaussian distributions on the same graph in R, e.g., $\{N(0, 1), N(-1, 2), N(2, 3)\}$, where as usual $N(\mu, \sigma^2)$ is the normal distribution with mean μ and variance σ^2 .

Pretending for the sake of illustration that R had no built-in Gaussian density function, here's how it might go, if you wanted to plot the three densities (say) for $\theta \in (-6, 8)$:

```
greco 2432> R
```

```
R : Copyright 2003, The R Development Core Team
Version 1.6.2 (2003-01-10)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.
```

```
[Previously saved workspace restored]
```

```
> dgauss <- function( theta, mu, sigma2 ) {
  return( exp( - ( theta - mu )^2 / ( 2 * sigma2 ) ) / sqrt( 2 * pi *
    sigma2 ) )
}
```

This function will return a vector of $N(\theta; \mu, \sigma^2)$ density values if the input `theta` is a vector.

```
> x11( ) # Needed under some operating systems to get an X-window.
```

The plot command in R has the basic form `plot(theta, y)`, where `theta` is a vector of x -axis locations and `y` is the corresponding vector of y -axis values, i.e., its basic purpose is

to make a scatterplot. To use this command to make a smooth plot of a function $f(\theta)$ you therefore need to do two things: (1) create a grid `theta` of points at which you want the function to be evaluated, and (2) ensure that when the plot command is invoked it connects the points together with little line segments to give the appearance of a smooth curve.

```
> theta <- seq( -6, 8, length = 500 )
```

This creates the grid; typically 500 points yields a smooth graph.

```
> plot( theta, dgauss( theta, 0, 1 ), type = 'l',  
       xlab = 'theta', ylab = 'Density' )
```

The `type = 'l'` option generates the smooth curve; `xlab` and `ylab` define labels for the horizontal and vertical axes, respectively.

To superimpose the second function on the first you can use the `lines` command, as follows:

```
> lines( theta, dgauss( theta, -1, 2 ), lty = 2 )
```

This works just like `plot` except that it adds the smooth curve to the existing graph; `lty = 2` chooses a particular type of dotted line.

```
> lines( theta, dgauss( theta, 2, 3 ), lty = 3 )
```

Now to be extra fancy you can label the three functions, like so:

```
> text( 1.25, 0.375, "N( 0, 1 )", cex = 1.25 )
```

This does just what it seems: it plots the text `N(0, 1)` with the center of the text string at the point (1.25, 0.375); `cex = 1.25` makes the text 25% larger than the default size.

```
> text( -3.0, 0.25, "N( -1, 2 )", cex = 1.25 )
```

```
> text( 4.25, 0.2, "N( 2, 3 )", cex = 1.25 )
```

You can use the `postscript` command (as demonstrated in the lecture notes) to make a PostScript file that saves the resulting graph (shown on the next page); the `pdf` command does the same thing as a way to save a PDF version of the plot.

