

# R sampling app version 2

Posted on [May 20, 2013](#) by [Matt Leonawicz](#)

Continuing from my previous post, [Introducing R Shiny applications \[app\]](#), here I expand upon the example Shiny app that was presented, which generates random samples from some common probability distributions. Specifically, I enhance the histogram plots of random samples from different probability distributions by adding the pdf (probability density function) for the currently selected distribution above the histogram. And, naturally, the pdf is not complete without specifying the support, so that is included as well.

## Distributions of Random Variables



You can click the image to go to the [app](#) page.

See my post, [Mathematical Notation in R Plots](#), for some examples and conveniently supplied **R** code for displaying the equation of a pdf on an **R** plot for many common distributions. The code shown in this post for version 2 of the random sampling app will make use of all these **R** expressions, but you won't see the expressions explicitly, so please see the other post if you are interested in how they work or just copying them for your convenience.

The first change is to simply add this line somewhere prior to the call to `shinyServer` in the `server.R` script:

```
load("plotmathExpressions.RData", envir=.GlobalEnv)
```

This workspace contains an expression object for each density function which will be used for annotating the histograms. They are loaded here because they only need to be loaded once. There is no reason to put them inside `shinyServer` and have them redefined every time the user alters inputs in their browser. I choose to keep these in an `.RData` file so that the `server.R` script is not unnecessarily cluttered, but it is fine to define objects at the top of

the `server.R` script, as I've already done as well. Again, you can see the code for these expression objects at the notation post linked to above.

I chose to define these objects with names such as `expr.norm`, `expr.t`, `expr.gam`, etc. This is convenient for grabbing the right expression based on the abbreviated object name for the currently selected distribution, as shown in the following code addition/change within `output$plot`:

I get the appropriate expression. I increase the top margin to make sufficient room for plotting the expression – some of the more complex formulas need more vertical space than others, but it doesn't look bad to leave the top margin fixed at 10 for all distributions since the overall graphic is fairly basic. I've set the `main` argument in `hist` to `expr`. And that's that. The `ui.R` script has not changed at all. Here is the complete `server.R` script, version 2:

Go to [version 2](#) of the sampling app. In my next post, [R sampling app version 3 \[app\]](#), I add more distributions to sample from, including discrete distributions. So in addition to pdf's, there will be pmf's (probability mass functions), and of course, with their supports included. I also have a [follow up](#) to my post, [Mathematical notation in R plots](#), to include pmf's as well to provide my flavor (for what it's worth) of expression code for common discrete distributions.