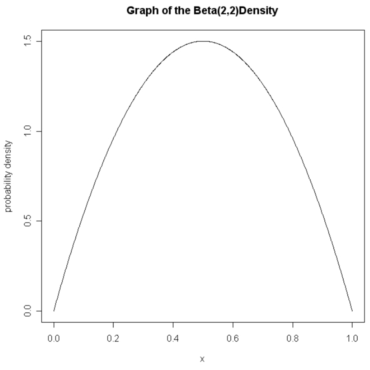
Course EPIB-613 - Introduction to Statistical Software

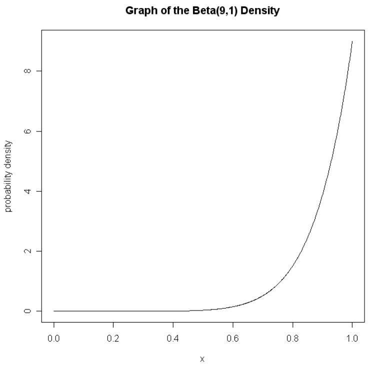
Assignment 4

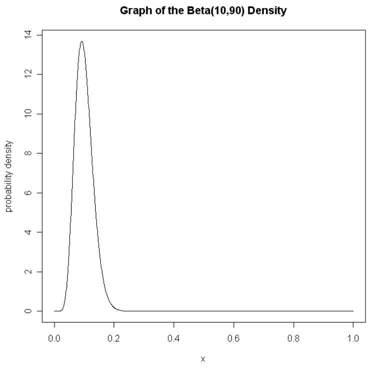
1. In this assignment we will be putting together many of the R skills learned so far in EPIB-613 in creating one new useful function.

The EPIB-613 class notes provide a function for carrying out a Bayesian analysis of a normal mean. Using similar ideas, here you will create your own R function that does Bayesian analysis for simple binomial parameters. The basic Bayesian theory says to start with a prior density on the unknown parameter (here the binomial probability), add in information in the data through the likelihood function (here binomial), and then multiple the two to derive the posterior density. We omit the mathematical details, and just provide these three functions:

Prior density $p \sim beta(a, b)$: A beta density is continuous on the range [0, 1], and is zero otherwise. Thus, it is ideal for binomial probability parameters, which are also always in this range. A beta(1,1) density is identical to a uniform density on the range [0, 1], and some other examples of beta densities are:


Graph of the Beta(2,2)Density


Graph of the Beta(9,1) Density


Graph of the Beta(10,90) Density

Data and Likelihood Function: Binomial(x, n, p), where n is the number of trials, x is the number of successes observed within these n trials, and p is the unknown probability of success that we are trying to draw inferences about.

Posterior density: $p \sim beta(a + x, b + n - x)$. Note how simple it is to form the posterior density given the data x and n, and the prior parameters, a and b. One just adds a to the number of successes x and b to the number of non-successes, $n - x$.

To create a function that automatically implements the above theory in R I suggest that you proceed along the steps outlined below.

(a) Start by creating an R function that takes as input the information needed to provide the posterior distribution given beta prior parameters and number of successes and trial of the binomial data. Use parameter names that will help the user of your program understand the inputs required. The output should be the parameters of the beta posterior distribution.

(b) Add some graphics to your function, in particular the prior and posterior beta density curves. Here are a couple of hints:

- Recall the difference between the plot command (to start a new graph) versus points (to add points to graph). Note also the use of the lty option for making different line types, and the type option for making continuous lines (e.g., lty="l"). We have seen similar examples which used the dnorm function, but here of course you must use the dbeta function.
- To avoid cutting off the tops of the graphs, think about the order in which the plots are created. Starting with the tallest plot (i.e., the posterior) may be a good idea.
- Use the range of (0,1) for your x-axis in all graphs. [A more sophisticated function might first examine all curves in the plot, and automatically decide on a range based on this information.]

(c) Add a 95% credible interval to your output. (Hint: Investigate the use of the qbeta function, which gives quantiles of the beta density. For example perhaps using qbeta(0.025, a, b) is useful for the lower limit, and qbeta(0.975, a, b) for the upper limit.)

(d) Run your program with a beta(5, 25) prior, and a data set with $x = 10$ successes in $n = 70$ trials. Check that the answer is correct, and that the plots seem reasonable (for example, the credible interval should roughly match 95% area under the posterior density curve, which you can approximately guess by eye.