STAT 440, Homework 2

due 2.5.2015

1 Turn-in on paper

1. For the following examples, show what function of U, a standard uniform random variable will yield a random variable, X, of the distribution mentioned. (Use the inverse CDF method.)

   (a) Let X be a random variable with DENSITY (pdf)

   $$f_X(x) = \frac{1}{2} x e^{-x^2} I_{\{0<x<\infty\}}$$

   (b) Let X be a random variable with DENSITY (pdf)

   $$f(x) = \begin{cases} \frac{1}{2}(x-2), \ 2 \leq x \leq 3 \\ \frac{1}{2}\left(2 - \frac{x}{3}\right), \ 3 < x \leq 6 \\ 0, \ otherwise \end{cases}$$

2. Let X and Y denote the coordinates of a point uniformly chosen in the circle of radius 1 centered at the origin. That is, their joint density is given by

   $$f(x,y) = \frac{1}{\pi} I_{\{x^2+y^2 \leq 1\}}$$

   (Note that X, Y are not independent of each other.) Find the joint density function of the polar coordinates

   $$R = \sqrt{X^2+Y^2} \quad \Theta = \tan^{-1}(Y/X)$$

   where the inverse tangent takes on values from $-\pi$ to π. Then comment on how you could use this transformation to create a simulation scheme for a uniformly random point in a circle if you were given two independent standard uniform random variables.

3. Suppose that X and Y are independent and gamma random variables with parameters (α, λ) and (β, λ), respectively. Just so that we are clear on parameterization, the density of X is then

$$f_X(x) = \frac{\lambda e^{-\lambda x}(\lambda x)^{\alpha-1}}{\Gamma(\alpha)} I_{\{0<x<1\}}$$

   (a) Compute the joint density of $U = X + Y$ and $V = X/(X + Y)$.

   (b) Recall that an exponential random variable with rate λ is a Gamma with parameters $(1, \lambda)$. Use this fact and what you have just shown for the distribution of U to create a simulation scheme for a Gamma random variable with parameters (n, λ) where n is an integer. (You just need to describe the steps to the simulation method; you do not need to code it up in R.)

   (c) Using the first two parts of this problem. Describe a simulation scheme for a beta random variable with parameters m and n, where both are integers.

2 Turn-in through Angel

Note that the following two functions should be included in the same file, called `hw2.R`, and then submitted in the HW2 drop box in Angel.

1. Write a function, which will take three arguments `n` (an integer one or larger), `mu` a real number, and `sigma` another (non-negative) real number. Your function should

   - Return a vector of length `n` of independent normal random variables each with mean `mu` and standard deviation `sigma`. You should use the transform method, and your function should be efficient in that you use at most one additional randomly generated uniform random variable than you return.
   - You may NOT use the `rnorm` command in R; you may only use `runif` to generate random numbers.
   - A useful command for writing this function is `is.odd()`, which takes an integer and returns `TRUE` if the number is odd and `FALSE` otherwise.
   - Name the function `rnormalt` and save it in the file `hw2.R` with only this function and the one from the following exercise.

- The grader will take your file, run it, and then issue the following into R:

  ```
  x=rnormalt(2000,2,3)
  ```

  The grader will then compute the mean and variance of `x`, along with a qqplot to ensure that your function accurately generates normal random variables.

2. Write a function, which will take three arguments `num`, `m` and `n`, both integers

   - Return a `num` number of F-distributed random variable with (m, n) degrees of freedom. You are only permitted to generate random variables using the `rnormalt` function from the last problem. (If you try and fail to accomplish that problem, then use `rnorm`.) Note fact 2 on page 58 of our text book to get you started.
   - Name the function `rfalt` and save it in the file `hw2.R` with only this function and the one from the previous exercise.
   - The grader will take your file, run the script, and then issue the following into R:

     ```
     x=rfalt(2000,8,12)
     ```

     In this case, `num` is 2000, `m` is 8 and `n` is 12. The grader will then compute the mean and variance of `x`, along with a qqplot to ensure that your function accurately generates F-distributed random variables.