Math 20
Lab 1
Due: July 6 2015

The goal of this lab is to get you acquainted with the R programming language. R is a statistical programming language that is the open source version of S (another statistical programming language). The first thing you should do is download the file `basic_intro.R` from the course web page. Open RStudio and use the file navigator in the bottom right quadrant to open the folder that you downloaded the file to. Click the file name and you should see the file appear in the upper left quadrant. Clicking the `source` tab in the top of this quadrant will execute the file and the output can be viewed in the lower left quadrant (the console).

Read through the file until you understand the code and commands. As you read be sure to 1) check what commands in the file get printed in the console 2) type certain commands from the file into the console and run them there. Typing `help(function)` into the console will bring up a dialogue that gives you information about the function in question. For instance `help(plot)` brings up the plotting dialogue.

Once you have read through the `basic_intro.R` file you can start on the following tasks. Each task asks you to create an R file. These R files will be graded and should be emailed to me before the start of class on the due date.

1. Using RStudio open a new R script titled `sign_yourlastname.R`. In this file you should define a function called `sgn.1` which takes a number x as an input and returns:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

   Then plot a square wave (i.e. the function $\text{sgn}(\sin(x))$ for $-3\pi \leq x \leq 3\pi$). When executed, your file should produce an image that looks similar to figure 1.

2. Open a new R script titled `flips_yourlastname.R`. The goal of this task is to simulate the experiment of flipping a coin 50 times where the random variable is the number of flips until observing a H for the first time (here if all 50 tosses are Ts we will just say the random variable is 50). An easy way of doing this is randomly generating a vector of zeros and ones of length 50. A 1 will represent observing a H and a 0 will represent observing a T. Generate a random vector of length 50 ten thousand times. Each time you generate a string be sure to record the first instance of a 1 in the string (here the `which` command might be useful). Once you have a record of the number of tosses make a histogram of the data. Over the histogram plot points that give the probability of the observed number of tosses (i.e. over the n-tosses bin plot a point of height $\frac{1}{2^n}$). There is a large spread of values for the histogram, so I suggest just displaying a y and x range similar to figure 2.

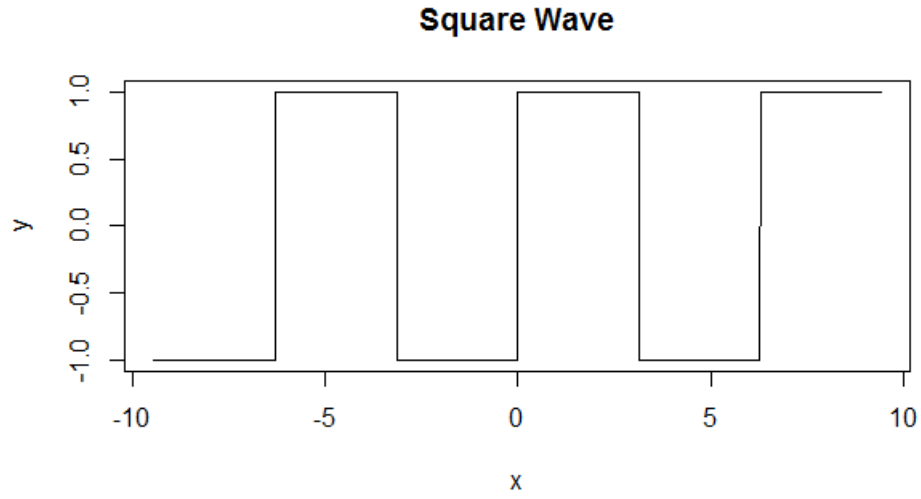

Figure 1: The script for task 1 should output something similar to this.

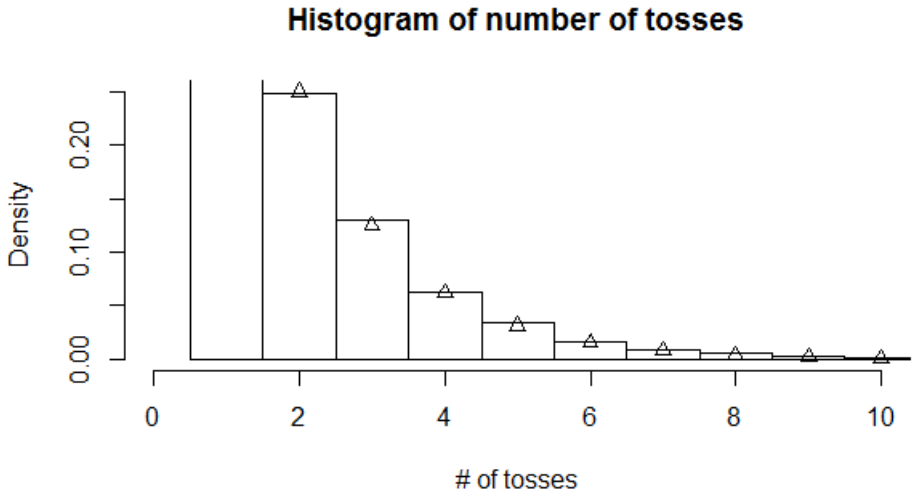

Figure 2: The script for task 2 should output something similar to this.