STAT 517: Takehome 1

1. (25 pts) Load the **state** data. Using logical and/or subsetting operators, complete the following tasks.

   (a) Find the names of all the states with a population greater than 5 million that are not located in the North Central United States.

   (b) Use two different approaches to find the complement of the above set of states.

   (c) Find the names of all the states that either have a population greater than 5 million or are not located in the North Central, but not both. Compare this to your answer in (a).

   (d) Order the states by region and then frost days; comment on any regional trends or unusual observations.

2. Consider the hit-or-miss demonstration we studied in class. Convert the following commands from the simulation into a function.

```
my.fcn <- ex1.fcn
my.max <- 4
my.n <- 10000
my.a <- 0
my.b <- 1
my.rand.x <- runif(my.n, min=my.a, max=my.b)
my.rand.y <- runif(my.n, min=0, max=my.max)
under= my.rand.y < my.fcn(my.rand.x)
my.m <- sum(under)
my.prop=my.m/my.n
I.approx.HM <- my.max*(my.b - my.a)*(my.m/my.n)
print(c(I.approx.HM,my.prop))
```

   The function arguments should include the number of random draws, the upper and lower endpoints of x, the function maximum over the domain of x, and a user-defined function. The function maximum should simply be a numeric value you determined beforehand. Graduate students should *not* provide the function maximum, but compute it within the body of the function. The user-defined function needs to be created beforehand–just like we created `ex1.fcn` in class. Don't be intimidated about including the name of a function as a function argument–it's just another **R** object. The output (which will replace the `print` statement) should include the estimated integral and the proportion of points under your curve.

   (a) Test your program by estimating:

$$\int_0^{\pi} \sin(u)\, du$$

   Use 10000 draws and compare your answer to the actual value of this integral. Remember that this is a simulation and there is no single right answer here.

(b) Grad students should add the necessary commands to generate a graphic of the simulation similar to the one I generated in class.

3. (30 pts) Here is code for Classic Monte Carlo Integration (using 10000 iterations) of the same function as above:

```
my.fcn <- function(x){
h <- sin(x)
return(h)
}
n = 10000
a = 0
b = pi
my.rand.x = runif(n, min=a, max=b)
Int.MC = ((b - a)/n)*sum(my.fcn(my.rand.x))
```

I want you to create a **for** loop that saves the estimate of `Int.MC` from each of 100 separate simulations of size 10000 and stores it in a numeric vector (of length 100). Construct a histogram of your estimates and include a red vertical reference line at 2. Comment on the accuracy of the Classic Monte Carlo integration approach. Graduate students should place the **for** loop in a function that allows the user to vary the sample size, the simulation size, the integration limits and the input function (they do not need to include a reference line for the integral though). Graduate students should demonstrate their function for the integral we studied in class.

4. (25 pts) Load the **iris** data. Use the data to make a plot as similar to the one on the web page as possible. Provide the commands you used as well as the finished plot. Include your name where "Your Name" currently appears. We have already used most of the commands you will need–`help(legend)` should help with the rest. Undergraduate students do not need to include the subheading "Iris Data".