Statistics 302 Spring 2014 Larget Assignment #5 posted February 17, 2014

Due Wednesday, February 26

Textbook Exercises

4.79, 4.82, 4.86, 4.94, 4.96, 4.100, 4.102, 4.123, 4.124, 4.126, 4.130, 4.132, 4.141, 4.144, 4.158, 4.160, 4.167

Computer Exercises

For each R problem, turn in answers to questions with the written portion of the homework. Send the R code for the problem to Katherine Goode. The answers to questions in the written part should be well written, clear, and organized. The R code should be commented and well formatted.

R problem 1 1. Write a function in R that will compute a p-value by a randomization test for a single population proportion. You may use the following template to begin, but will need to add missing code to make the function work.

```
# function for p-value from a proportion
# n is sample size
# x is number of successes in sample
# p0 is null success probability
# R is the number of samples to generate
pvalue.p = function(n,x,p0,R,alternative=c("not.equal","less","greater")) {
  alternative = match.args(alternative)
  p.hat = numeric(R)
  for ( i in 1:R ) {
    p.hat[i] = mean(sample(c(0,1),size=n,replace=TRUE,probs=c(1-p0,p0)))
  }
  if ( alternative == "not.equal" ) {
## do something
  }
  else if ( alternative == "less" ) {
## do something
  }
  else if ( alternative == "greater" ) {
## do something
  }
## return something
}
```

2. Use the function to compute p-values in the following situations.
   (a) Katherine's first ten tries at the ESP test: $n = 10$, $x = 6$, $H_0 : p = 0.2$, $H_a : p > 0.2$.
   (b) Paul the Octopus for all Euro 2008 and World Cup 2010 matches: $n = 13$, $x = 11$, $H_0 : p = 0.5$, $H_a : p > 0.5$.b
   (c) A genetics hypothesis where if true, $p = 11/16$: $n = 230$, $x = 161$, $H_0 : p = 11/16$, $H_a : p \neq 11/16$.

R problem 2 (a) Write a function that returns a p-value using a randomization distribution for testing a difference in two population means. Do this by repeatedly assigning individuals to the two groups at random. (This is also called a permutation test.) Here is a very brief template for the function.

```
# function for p-value from differences in sample means
# x is the first sample
# y is the second sample
# R is the number of samples to generate
pvalue.mudiff = function(x,y,R,alternative=c("not.equal","less","greater")) {
  alternative = match.args(alternative)
  stat = numeric(R)
  total = c(x,y)
  nx = length(x)
  ny = length(y)
  for ( i in 1:R ) {
    newTotal = sample(total)
    stat[i] = mean(newTotal[1:nx]) - mean(newTotal[(nx+1):(nx+ny)])
  }
  if ( alternative == "not.equal" ) {
## do something
  }
  else if ( alternative == "less" ) {
## do something
  }
  else if ( alternative == "greater" ) {
## do something
  }
## return something
}
```

(b) Use the function to find a p-value for the data in Table 4.11 on page 278.