Stat 401B Fall 2015 Lab #5 (Due October 8)

There is an R code set at the end of this lab and posted on the course web page that will prove useful for doing this lab. Use it in doing the following.

1. Simulate 10,000 sets of observations \((x_1, x_2, x_3, x_4, x_5, x_6)\) from Vardeman's "brown bag" (a distribution that is normal with mean 5.0 and standard deviation 1.715). Treat the first 5 observations in each set as "the sample" available upon which to do inference.

   a) Make 10,000 two-sided 90\% prediction intervals for \(x_6\) (one additional observation beyond the data in hand) and determine the fraction of the 10,000 intervals that cover their respective new observation.

   b) Make 10,000 two-sided 99\% prediction intervals for 90\% of the brown bag distribution and determine the fraction of them that contain at least 90\% of that normal distribution.

2. Do some plotting of \(\chi^2\) and F probability densities. (Mark on your plots which density is which.)

   a) Plot \(\chi^2\) densities for \(\nu = 2, 4, 6, \) and 10.

   b) Plot F densities for pairs of degrees of freedom (2,20), (3,20), (20,3), and (3,3).

3. The \(t\) confidence limits for a mean are widely believed to be fairly "robust" (i.e. unlikely to go badly wrong unless the distribution being sampled is very non-bell-shaped).

   a) There is code for simulating \(n = 5\) confidence intervals for a mean when (presumably unknown to the user) the distribution sampled is not normal, but rather is \(U(0,1)\). Do the intervals have at least the nominal coverage probability?

   b) Modify the code referred to above, making the distribution sampled the Exponential distribution with parameter 1. (You'll need to change not only the data, but also the mean to be covered by the intervals.) Do the intervals have actual coverage probability close to the nominal coverage probability in this case?

4. Vary sample size \(n\) and (unknown to the investigator) value of \(p\) across the grid of values with \(n = 50, 100, 400, 1600\) and \(p = .2, .1, .01, .001\) and approximate the actual coverage probabilities of nominally 90\% confidence limits for \(p\) made using the method in the text (on page 404) and the method on formula sheet for these 16 situations. What do these results suggest about the reliabilities of the two methods?

5. Modify the code in the code set to recreate Figure 3.18 of the Vardeman and Jobe text (see page 90).

6. There is code included that expedites the computations of the HW problems from Chapter 7. Modify and run it on the data of Example 1 of Chapter 4 and on the data of Exercise 2 Section 1 Exercises page 460 of Vardeman and Jobe.
Here is some code for Lab #5

First we study prediction and tolerance intervals

```r
M <- matrix(rnorm(60000, mean=5, sd=1.715), nrow=10000, byrow=T)
av <- 1:10000
s <- 1:10000
Lowpi <- rep(0, 10000)
Uppi <- rep(0, 10000)
Lowti <- rep(0, 10000)
Upti <- rep(0, 10000)
chkpi <- rep(0, 10000)
chkti <- rep(0, 10000)

for (i in 1:10000) {
  av[i] <- mean(M[i, 1:5])
}
for (i in 1:10000) {
  s[i] <- sd(M[i, 1:5])
}
for (i in 1:10000) {Lowpi[i] <- av[i] - 2.132 * s[i] * sqrt(1 + (1/5))}
for (i in 1:10000) {Uppi[i] <- av[i] + 2.132 * s[i] * sqrt(1 + (1/5))}
for (i in 1:10000) {if ((Lowpi[i] < M[i, 6]) & (M[i, 6] < Uppi[i])) chkpi[i] <- 1}
cbind(Lowpi[1:10], Uppi[1:10], M[1:10, 6], chkpi[1:10])
mean(chkpi)
```
for(i in 1:10000) {Lowti[i]<-av[i]-6.655*s[i]}
for(i in 1:10000) {Upti[i]<-av[i]+6.655*s[i]}
for(i in 1:10000) {if(pnorm(Upti[i],mean=5,sd=1.715)-pnorm(Lowti[i],mean=5,sd=1.715)>.9) chkti[i]<-1}
cbind(Lowti[1:10],Uppi[1:10],pnorm(Upti[1:10],mean=5,sd=1.715)-pnorm(Lowti[1:10],mean=5,sd=1.715),chkti[1:10])
mean(chkti)

#Now we plot some Chi-square densities
curve(dchisq(x,df=2),xlim=c(0,25))
curve(dchisq(x,df=4),add=TRUE,xlim=c(0,25))
curve(dchisq(x,df=6),add=TRUE,xlim=c(0,25))
curve(dchisq(x,df=10),add=TRUE,xlim=c(0,25))

#Now we plot some F densities
curve(df(x,df1=20,df2=20),xlim=c(0,10))
curve(df(x,df1=3,df2=20),add=TRUE,xlim=c(0,10))
curve(df(x,df1=20,df2=3),add=TRUE,xlim=c(0,10))
curve(df(x,df1=3,df2=3),add=TRUE,xlim=c(0,10))

#Here's a bit of a "robustness" study for a t CI for a mean
M<-matrix(runif(50000,min=0,max=1),nrow=10000,byrow=T)
av<-1:10000
s<-1:10000
Lowci<-rep(0,10000)
Upcii<-rep(0,10000)
chkci<-rep(0,10000)
for (i in 1:10000) {
    av[i] <- mean(M[i, 1:5])
}
for (i in 1:10000) {
    s[i] <- sd(M[i, 1:5])
}
for (i in 1:10000) {Lowci[i] <- av[i] - 2.132 * s[i] / sqrt(5)}
for (i in 1:10000) {Upci[i] <- av[i] + 2.132 * s[i] / sqrt(5)}
for (i in 1:10000) {if ((Lowci[i] < .5) & (.5 < Upci[i])) chkci[i] <- 1}
cbind(Lowci[1:10], Upci[1:10], chkci[1:10])
mean(chkci)

# Here is code for studying the effectiveness of CIs for p
p <- .05
n <- 100
X <- rbinom(10000, size = n, p)
phat <- X / n
phattilde <- (X + 2) / (n + 4)

Lowcip <- 1:10000
Upcip <- 1:10000
Lowciptilde <- 1:10000
Upciptilde <- 1:10000

chkcip <- rep(0, 10000)
chkciptilde <- rep(0, 10000)
for(i in 1:10000) {Lowcip[i]<-phat[i]-1.645*sqrt(phat[i]*(1-phat[i])/n)}

for(i in 1:10000) {Upcip[i]<-phat[i]+1.645*sqrt(phat[i]*(1-phat[i])/n)}

for(i in 1:10000) {if((Lowcip[i]<p) & (p<Upcip[i])) chkcip[i]<-1}

for(i in 1:10000) {Lowciptilde[i]<-phat[i]-1.645*sqrt(phattilde[i]*(1-phattilde[i])/n)}

for(i in 1:10000) {Upciptilde[i]<-phat[i]+1.645*sqrt(phattilde[i]*(1-phattilde[i])/n)}

for(i in 1:10000) {if((Lowciptilde[i]<p) & (p<Upciptilde[i])) chkciptilde[i]<-1}

mean(chkcip)

mean(chkciptilde)

#Here is a quick way to make a single normal plot

#(There is code for making multiple normal plots on a single set of axes in the HW)

qqnorm(c(1,4,3,6,8,34,3,6,2,7),datax=TRUE)

#Here is some R code for the class "Depth of Cut" example

pulses<-c(rep("A(100)",4),rep("B(500)",4),rep("C(1000)",4))
depth<-c(7.4,8.6,5.6,8.0,24.2,29.5,26.5,23.8,33.4,37.5,35.9,34.8)

Depth<-data.frame(depth,pulses)

Depth

plot(depth ~ pulses,data=Depth)
plot(as.factor(pulses), depth)

summary(Depth)

aggregate(Depth$depth, by=list(Depth$pulses), mean)

aggregate(Depth$depth, by=list(Depth$pulses), sd)

depth.aov<-aov(depth ~ pulses, data=Depth)

summary(depth.aov)