PREDICT 401 Introduction to Statistical Analysis Data Analysis Assignment #3

Part 1: Presentation of the Chi Square Distribution

Use the code in Appendix A of this assignment. Comment statements document the program. The chi square family of distributions will be used to illustrate various aspects of sampling distribution convergence and bootstrapping. This assignment involves executing the code supplied with a designated expected value (mu =1, variance = 2mu) for the chi square distribution. This is an example of an asymmetric distribution. Using the R script supplied, execute the first part of the program to see the shape of the chi square distribution involved. Part 2: Bootstrap Approach

Continue using the code in Appendix A. Two different bootstrapping methods will be introduced and compared to the traditional t-test. In some cases the three methods agree closely. When this is not the case, the bootstrap t method is preferred particularly when asymmetric distributions are involved.

mu = 1 n = 20	mu = 1 n = 40	mu = 1 n = 200	mu = 1 n = 200

This portion of the script will be executed four times using the same sample sizes shown above. Keep set.seed(124) unchanged. Change the sample size n for each iteration. **Execute each of the four iterations starting with set.seed(124).** The program proceeds in stages.

1) First a simple random sample (srs) of size n is drawn from the chi square distribution.
2) Summary statistics are calculated and a histogram of srs is presented.
3) This is followed by resampling 10,000 times from srs with N = 10,000. Each sample is drawn from srs with replacement.
4) For each sample, the mean and t-statistic are calculated and retained.
5) Histograms of the values calculated at step (3) are produced. It is apparent both histograms are skewed, with the t-statistic histogram more so.
6) Confidence intervals are generated using the traditional t-statistic method, the bootstrap percentile method and the bootstrap t method. Fill in a table such as the one below with these confidence intervals. Note the convergence of the intervals as n is increased.

	95% Confidence Intervals		
n	traditional t intervals	percentile bootstrap	bootstrap t
20			
40			
200			
500			

Note: Chihara and Hesterberg, *Mathematical Statistics with Resampling and R.* Chapter 5 page 113 and Chapter 7 page 197 provide explanations for confidence interval construction. What is explained there is used in the R script for this part of the assignment. These explanations, with minor modification, can be adapted for one-sided confidence interval construction.

Part 3 Analyzing Data

Use the code given in Appendix B of this assignment. Refer to Black *Business Statistics* Chapter 8 page 294 problem 2. This part of the assignment will use the data in "Hospital.csv". Based on the results from Part 2 and the size of the data file, all the data will be used for resampling as shown in the R script supplied. Refer to Black *Business Statistics* page 15 for the data dictionary. Review the problem and execute the code as presented.

1) Observe the histogram of Census, and the resulting histograms for the resampled mean values and the resampled t-statistic.
2) The script produces two sets of three confidence intervals. 90% confidence intervals are in one set and 99% confidence intervals in the other.
3) Take note of these intervals noting any similarities and differences. What do these intervals suggest about the sample size? How does a change in confidence level affect the intervals?
4) It may be advantageous to record them in a table like the following.

Confidence Interval	traditional t intervals	percentile bootstrap	bootstrap t
90%			
99%			

Part 4 Hypothesis Testing

Refer to Black *Business Statistics* Chapter 9 page 351 problem 2. The data in "Hospital.csv" will be used. Refer to Black *Business Statistics* page 15 for the data dictionary. We will resample with replacement from these data. Partial script for this part is provided in Appendix C. To start the assignment, execute the supplied script. **It will be necessary to supplement the script with statements to perform one-sided hypothesis tests using one-sided confidence intervals as required by the problem. Use t.test() to generate the one-sided t statistic confidence intervals. One-sided bootstrapping confidence intervals only require use of one quantile that corresponds to the confidence level required. You must pick the correct one for this assignment.**

1) Test the hypothesis that the average hospital averages more than 700 births per year. Do this twice using one-sided 99% and 95% confidence intervals. Calculate a traditional one-sided t confidence interval, a one-sided percentile confidence interval and a one-sided bootstrap t confidence interval. It may be advantageous to record results in a table like the following. Are there any statistically significant differences? If so, where?

Confidence Interval	traditional t intervals	percentile bootstrap	bootstrap t
95%			
99%			

2) Test the hypothesis that on average hospitals in the USA employ fewer than 900 personnel. Do this using one-sided 90% confidence intervals. It may be advantageous to record results in a table like the following. Are the results statistically significant?

Confidence Interval	traditional t intervals	percentile bootstrap	bootstrap t
90%			

3) Determine how you would report these results to the hospital. This concludes the data analysis. Don't forget the quiz.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# Predict 401 Data Analysis Assignment #3

#-------------------------------------------------------------------------
# Part 1: Presentation of the Chi Square Distribution
# Appendix A
#-------------------------------------------------------------------------

require(moments)

# Plot the chi square density function. mu equals the expectation which is the
# degrees of freedom for the chi square density.  However, for the chi square
# distribution if the mean is mu then the variance is 2mu.  We will be using mu = 1.0.

mu <- 1  # This is where different mean values may be substituted.

#-------------------------------------------------------------------------
limit <- round(0.9*mu + 11)   # This generates a plotting limit.

X <- seq(1,2*limit)/2   # This generates values for computing the density.
plot(X, dchisq(X, df = mu, ncp = 0), type = "b",  xlim = c(0, limit),
    col = "darkred", lwd = 2, main = "Chi Square Density")

#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
# Part 2: Bootstrap Approach.
# Appendix A

# Two different bootstraping methods will be used.  In some cases the traditional
# t statistic works as well as the bootstrap t method.  When this is not the case,
# the bootstrap t method is preferred since it will compensate for asymmetry.

# Draw a random sample of size n. It is designated as srs and will be used later.
```

```
n <- 20  #  This is the sample size which will be changed repeatedly for this part.

set.seed(124)  # Retain this random seed. Use the same seed for each iteration.
mu <- 1  # This is the mean value for the chi square distribution.

# First a simple random sample is taken from the population.
srs <- rchisq(n, mu, ncp = 0)
# srs is the initial sample which is resampled for the bootstrap.

# Calculate the sample mean and standard deviation for use later.
mu.boot <- mean(srs)
std.boot <- sd(srs)

# Produce the histogram of the simple random sample srs.
cells <- seq(from = 0, to = max(srs)+0.5, by = 0.5)
hist(srs, breaks = cells, main = "Histogram of Initial Simple Random Sample", col = "blue")
# This is the distribution used in bootstrapping.

#-------------------------------------------------------------------------
# What follows is resampling with replacement from the simple random sample srs.
# This resampling with replacement is the basis of bootstrapping.

N <- 10^4      # Number of iterations.
# Define vectors for storage purposes.
my.boot <-numeric(N)
t.my.boot <- numeric(N)

for (i in 1:N)
{
  x <- sample(srs, n, replace = TRUE)                  # Sample size is n.
  my.boot[i] <- mean(x)                              # Calculate mean value for srs.
  t.my.boot[i] <- (mean(x)-mu.boot)/(sd(x)/sqrt(n))      # Calculate t statistic for srs.
}

#-------------------------------------------------------------------------
# Construct a histogram of the resampled mean values and superimpose normal density function.
m <- mean(my.boot)
s <- sd(my.boot)
x0 <- min(my.boot)
x1 <- max(my.boot)+1
x <- seq(x0,x1,length=1000)
y <- dnorm(x, mean=m, sd = s, log=FALSE)
ylim <- max(y)+0.05

# Sampling distribution of mean values with quantiles at 2.5% and 97.5% shown as vertical lines.
```

```
hist(my.boot, main = "Resampling distribution of mean values", probability = TRUE, col = "red")
abline(v= m, col = "green", lty = 2, lwd = 2)
abline(v = quantile(my.boot, probs = c(0.025, 0.975)), col = "green", lty = 2, lwd = 2)
lines(x,y, col="green", lwd = 2)

# It is apparent that the resampled mean values have a histogram which is skewed right.
#---------------------------------------------------------------------
# Construct a histogram of the resampled t-statistic values and superimpose the t density function.
x0 <- min(t.my.boot)
x1 <- max(t.my.boot)
x <- seq(x0,x1,length=1000)
y <- dt(x, df = n-1)
ymax <- max(y)

# Sampling distribution of t values with quantiles at 2.5% and 97.5% shown.
hist(t.my.boot, col = "green", main = "Resampling Distribution of t-statistic",
    probability = TRUE, ylim = c(0,ymax))
abline(v = 0.0, col = "darkred", lty = 2, lwd = 2)
abline(v = quantile(t.my.boot, probs = c(0.025, 0.975)), col = "darkred", lty = 2, lwd = 2)
lines(x,y, col="darkred", lwd = 2)

# It is apparent that the resampled t statistic values have a left-skewed histogram.
#---------------------------------------------------------------------

# This exercise demonstrates the confidence intervals for the three methods converge
# as the sample size increases. With a sample size of 200 the results are becoming
# much closer.  This exercise also reveals that the bootstrap t interval adjusts for
# the skewness in the sampling distribution which in many cases results in
# better coverage of the true population mean than what the symmetric traditional
# t statistic confidence interval provides.

#---------------------------------------------------------------------
# Traditional confidence interval for the mean using srs and t-statistic.
t.test(srs, conf.level=0.95, alternative = c("two.sided"))
# We will compare to this confidence interval.
#---------------------------------------------------------------------
# Percentile bootstrapping confidence interval.
round(quantile(my.boot, prob = c(0.025,0.975)), digits = 3)
#---------------------------------------------------------------------
# Determine a two-sided confidence interval using bootstrap t distribution.
Q1 <- quantile(t.my.boot, prob = c(0.025), names = FALSE)
Q2 <- quantile(t.my.boot, prob = c(0.975), names = FALSE)
round(mu.boot - Q2*(std.boot/sqrt(n)), digits = 3)
round(mu.boot - Q1*(std.boot/sqrt(n)), digits = 3)
#---------------------------------------------------------------------
#---------------------------------------------------------------------
```

```
#-----------------------------------------------------------------------
# Part 3 Analyzing Data
# Appendix B
#-----------------------------------------------------------------------
# Analyzing the Databases Problem 2 page 294

hospital <- read.csv(file.path("c:/RBlack/","Hospital.csv"),sep=",")
str(hospital)
require(moments)

# EDA on census reveals an asymmetric distribution.
census <- hospital$Census
summary(census)
hist(census, col = "blue")
boxplot(census, col = "blue")
skewness(census)

# The distribution of census is similar to what was shown in the exercises above.
# For what follows we will consider census a sample from a larger population.
# Resampling will be used to generate sampling distributions.

# Set the stage for resampling.
mu <- mean(census)
n <- length(census)                # The sample size is the number of observations in census.
N <- 10^4
census.mean<-numeric(N)
census.t <- numeric(N)
set.seed(124)

for (i in 1:N)
{
  x <- sample(census,n,replace = TRUE)
  census.mean[i] <- mean(x)
  census.t[i] <- (mean(x)-mu)/(sd(x)/sqrt(n))}

#-----------------------------------------------------------------------
# Construct histogram and superimpose normal density function.
m <- mean(census.mean)
s <- sd(census.mean)
x0 <- min(census.mean)
x1 <- max(census.mean)+1
x <- seq(x0,x1,length=1000)
y <- dnorm(x, mean=m, sd = s, log=FALSE)
ylim <- max(y)+0.05

hist(census.mean, main="Bootstrap distribution of mean values", probability = TRUE, col = "red")
```

```
abline(v = quantile(census.mean, probs = c(0.025, 0.975)), col = "green", lty = 2, lwd = 2)
abline(v= m, col = "green", lty = 2, lwd = 2)   # observed mean
lines(x,y, col="green", lwd = 2)

#-----------------------------------------------------------------------
# Construct a histogram of the resampled t-statistic values and superimpose the t density function.

x0 <- min(census.t)
x1 <- max(census.t)
x <- seq(x0,x1,length=1000)
y <- dt(x, df = n-1)
ymax <- max(y)

hist(census.t, main="Bootstrap distribution of t statistic", probability = TRUE, col = "green")
abline(v=0.0, col = "red", lty = 2, lwd = 2)
abline(v = quantile(census.t, probs = c(0.025, 0.975)), col = "red", lty = 2, lwd = 2)
lines(x,y, col="darkred", lwd = 2)
#-------------------------------------------------------------------------------------------

# Construct two-sided confidence interval using t-statistic.  The following
# calculation gives a traditional t-statistic confidence interval using
# 90% and 99% for comparison.
t.test(census, conf.level = 0.9, alternative = c("two.sided"))
t.test(census, conf.level = 0.99, alternative = c("two.sided"))

# Determine two-sided bootstrap percentile confidence intervals.
round(quantile(census.mean, probs=c(0.05,0.95)), digits = 2)
round(quantile(census.mean, probs=c(0.005,0.995)), digits = 2)

# Determine two-sided bootstrap t confidence intervals.
Q2 <- quantile(census.t, prob=c(0.95), names = FALSE)
Q1 <- quantile(census.t, prob=c(0.05), names = FALSE)
round(mu -Q2*sd(census)/sqrt(n), digits = 2)
round(mu -Q1*sd(census)/sqrt(n), digits = 2)

Q2 <- quantile(census.t, prob=c(0.995), names = FALSE)
Q1 <- quantile(census.t, prob=c(0.005), names = FALSE)
round(mu -Q2*sd(census)/sqrt(n), digits = 2)
round(mu -Q1*sd(census)/sqrt(n), digits = 2)


# The resulting 90% confidence intervals are similar as a consequence of the sample size.
# Such results do not always result, particularly when outliers are common.
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
# Part 4 Hypothesis Testing
```

```
# Appendix C
#-------------------------------------------------------------------------
# Problem 2 Page 351
# For what follows we will consider the data a sample from a larger population.
# It will be necessary to develop the necessary confidence intervals based on the above examples.
# The intervals required are one-sided, so adjustments will be needed.
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
# Does the average hospital have more than 700 births per year?

summary(hospital$Births)
hist(hospital$Births, col = "blue")
boxplot(hospital$Births, col = "blue")

# Distribution of births is asymmetric and right skewed.

# Form the sampling distribution for the mean.

births <- hospital$Births
n <- length(births)
N <- 10^4
mu <- mean(births)
births.t <- numeric(N)
births.mean <- numeric(N)
set.seed(124)

for (i in 1:N)
{
  x <- sample(births,n,replace = TRUE)
  births.mean[i] <- mean(x)
  births.t[i] <- (mean(x)-mu)/(sd(x)/sqrt(n))
}

#-------------------------------------------------------------------------------
# Construct histogram and superimpose normal density function.
m <- mean(births.mean)
s <- sd(births.mean)
x0 <- min(births.mean)
x1 <- max(births.mean)+1
x <- seq(x0,x1,length=1000)
y <- dnorm(x, mean=m, sd = s, log=FALSE)
ylim <- max(y)+0.05

hist(births.mean, main="Bootstrap distribution of mean values", probability = TRUE, col = "red")
abline(v= m, col = "green", lty = 2, lwd = 2)   # observed mean
abline(v=700,col="green",lty=2, lwd = 2)         # null hypothesis value for mean
```

```
abline(v= quantile(births.mean, probs=0.01),
                col="green", lty =2, lwd = 2)  # quantile for 99% confidence interval
lines(x,y, col="green", lwd = 2)

#-------------------------------------------------------------------------------
# Construct a histogram of the resampled t-statistic values and superimpose the t density function.
x0 <- min(births.t)
x1 <- max(births.t)
x <- seq(x0,x1,length=1000)
y <- dt(x, df = n-1)
ymax <- max(y)

# Sampling distribution of t values with quantiles at 2.5% and 97.5% shown.
hist(births.t, col = "green", main = "Resampling Distribution of t-statistic",
    probability = TRUE, ylim = c(0,ymax))
abline(v = 0.0, col = "darkred", lty = 2, lwd = 2)
abline(v = quantile(births.t, probs = 0.99, names = FALSE),
      col = "darkred", lty = 2, lwd = 2)          # quantile for the 99% confidence interval
lines(x,y, col="darkred", lwd = 2)

#-------------------------------------------------------------------------
# Construct confidence intervals and perform the necessary hypothesis tests.
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
# Personnel question-----------------------------------------------------
# On average, do hospitals employ fewer than 900 people?

personnel <- hospital$Personnel

hist(personnel, col = "blue")
boxplot(personnel, col = "blue")
mu <- mean(personnel)

# Distribution of personnel is asymmetric and right skewed.

n <- length(personnel)
N <- 10^4
personnel.t<-numeric(N)
personnel.mean <- numeric(N)
set.seed(124)

for (i in 1:N)
{
  x <- sample(personnel,n,replace = TRUE)
  personnel.mean[i] <- mean(x)
  personnel.t[i] <- (mean(x)-mu)/(sd(x)/sqrt(n))
```

```
}

#-------------------------------------------------------------------------
# Construct histogram and superimpose normal density function.
m <- mean(personnel.mean)
s <- sd(personnel.mean)
x0 <- min(personnel.mean)
x1 <- max(personnel.mean)+1
x <- seq(x0,x1,length=1000)
y <- dnorm(x, mean=m, sd = s, log=FALSE)
ylim <- max(y)+0.05

hist(personnel.mean, main="Bootstrap distribution of mean values",
    probability = TRUE, col = "red", ylim = c(0.0, 0.007))
abline(v= m, col = "green", lty = 2, lwd = 2)   # observed mean
abline(v=900,col="green",lty=2, lwd = 2)        # null hypothesis value for mean
abline(v=quantile(personnel.mean, probs=0.9),
      col="green", lty=2, lwd= 2)     # quantile for the confidence interval
lines(x,y, col="green", lwd = 2)

#-------------------------------------------------------------------------------
# Construct a histogram of the resampled t-statistic values and superimpose the t density function.
x0 <- min(personnel.t)
x1 <- max(personnel.t)
x <- seq(x0,x1,length=1000)
y <- dt(x, df = n-1)
ymax <- max(y)

# Sampling distribution of t values with quantiles at 2.5% and 97.5% shown.
hist(personnel.t, col = "green", main = "Resampling Distribution of t-statistic",
    probability = TRUE, ylim = c(0,ymax))
abline(v = 0.0, col = "darkred", lty = 2, lwd = 2)
abline(v = quantile(personnel.t, probs = 0.1, names = FALSE),
      col = "darkred", lty = 2, lwd = 2)           # quantile for the confidence interval
lines(x,y, col="darkred", lwd = 2)

#-------------------------------------------------------------------------
# Construct confidence intervals and perform the necessary hypothesis tests.
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
```