Project 2 Statistical Computing

Tyler Grimes

February 25, 2016

Introduction

The goal of this project is to determine the average price, in dollars, that a student at UNF spends on a haircut. A sample was obtained from the STA 6106 Spring class. This was not a random sample, but we will assume it is representative of the population of UNF students. The sample is shown in the table below.

Observation	1	2	3	4	5	6	7	8	9	10	11	12
Price	17.00	16.49	45.00	25.00	12.00	18.00	13.00	20.00	10.00	0.00	15.00	50.00

We will estimate the average price by creating a 90% confidence interval. Three different methods will be considered; a t-based confidence interval with normality assumed, a bootstrap confidence interval, and a double bootstrap interval.

Assuming Normality

Since our estimator is the sample mean, we might first want to consider creating a simple confidence interval using the t-distribution. Remember, if the estimator has a normal distribution, we can create a 90% confidence interval by

$$(\bar{x} + t_{0.9,df=n-1}SE_{\bar{x}}, \bar{x} + t_{0.1,df=n-1}SE_{\bar{x}})$$

where $t_{\alpha,df}$ is the α^{th} percentile of the t-distribution with df degrees of freedom, and $SE_{\bar{x}}$ is the standard error of $\bar{x}$. Given that our sample size of $n = 12$ is fairly small, we can not rely on the central limit theorem to assert that $\bar{x}$ is approximately normal. On the other hand, if we have reason to believe that a normal distribution might be a good model for the haircut prices, we could proceed under this assumption; in which case, we assume the population is normal, hence $\bar{x}$ will be normal.

The first step now would be to verify our model assumption. A quick look at the histogram of our sample (figure 1) suggests that the population might be skewed right, or possibly bimodal. However, it is difficult to make a strong judgement with only 12 observations.

We will perform a Shapiro-Wilk test, which tests the hypothesis that our sample is from a normal distribution. The p-value for this test is 0.0408, which is small enough for us to reject our normality assumption.

We could also look at a quantile-quantile plot (Q-Q plot) with our sample against a normal distribution (figure 2). If the sample truly does come from a normal population, the plots on the Q-Q plot should fall very close to the $y = x$ line, and should be distributed randomly above and below this line.

The Q-Q plot shows a clear pattern in the observations. We should be very suspect of our normality assumption. However, if we proceed anyway, the 90% confidence interval for the mean haircut price is $(12.77064, 27.47769)$.

Bootstrap

In the previous approach, the normality assumption does not appear to be valid. And if we think about haircut prices, it is easy to imagine the distribution having three modes; students who pay \$0, students who

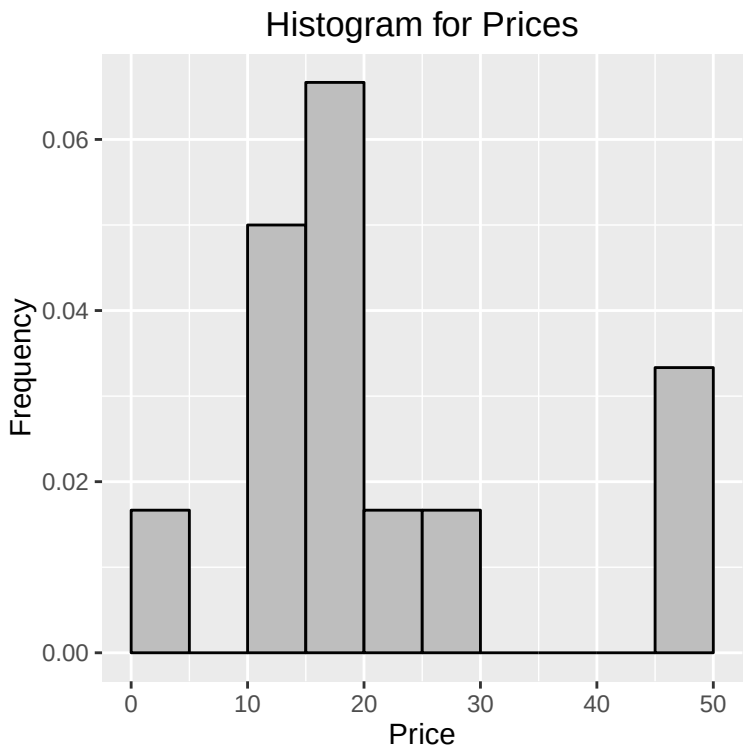

Figure 1: Histogram of sample prices

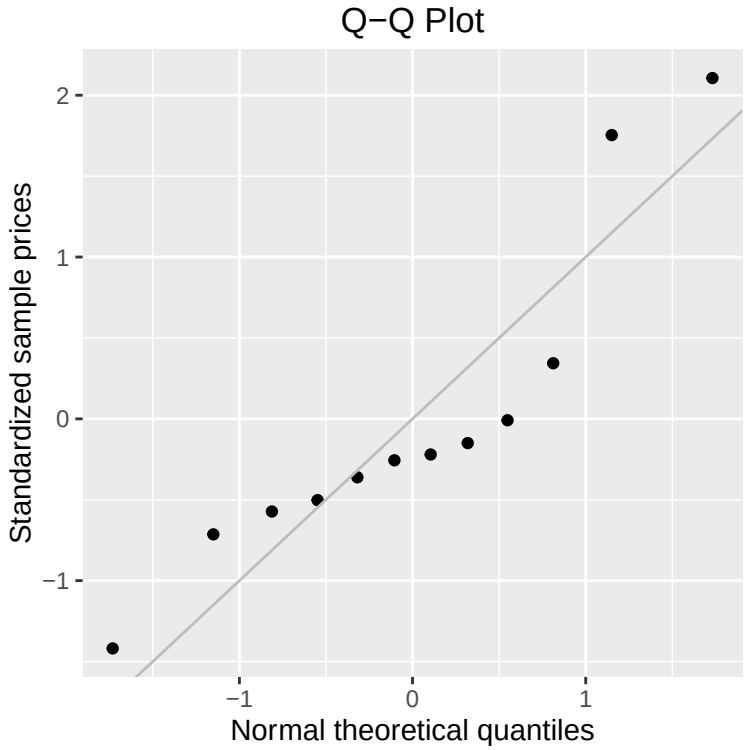

Figure 2: Q-Q Plot for sample prices with normal distirbution.

pay a medium amount of around \$10-\$20, and students who pay a premium amount >\$50. This suggests a mixture model would be more appropriate, but then the t-based confidence interval is no longer valid.

We now consider an approach that does not depend on the population's distribution; the non-parametric bootstrap method.

Percentile Method

For bootsrapping in general, we use resampling with replacement to simulate random draws from the population. These simulated samples are refered to as bootstrap samples. In the most basic implementation of the bootstrap, an estimate is calulated for each bootstrap sample, which results in a distribution of estimators, and the middle 90% of these estimators is used to construct a confidence interval. This is the percentile method.

Algorithm

Let $(x_1, ..., x_n)$ denote our random sample.

1. Obtain a bootstrap sample $(x_1^*, ..., x_n^*)$ from the original sample $(x_1, ..., x_n)$ by sampling with replacement. Compute $\widehat{\widetilde{x}}^*_{(b)}$.
2. Repeat 1 B times to obtain B estimates $(\widehat{\widetilde{x}}^*_{(1)}, ..., \widehat{\widetilde{x}}^*_{(B)})$.
3. Order the $\widehat{\widetilde{x}}^*_{(b)}$'s from smallest to largest.
4. Construct the 90% confidence interval by $\widehat{\widetilde{x}}^*_{(ceiling(B*0.05))}, \widehat{\widetilde{x}}^*_{(celing(B*0.95))}$

Results

The resulting confidence interval is $(14.33333, 26.99833)$. The concern with the percentile method is that, although the confidence level is 90%, the actual coverage probability may not be 90%. That is, the procedure of constructing a percentile bootstrap interval may not actually contain the true population mean 90% of the time. Usually, this coverage probability is much lower.

Double Bootstrap

A method for constructing a confidence interval that provides a coverage probability that is closer to the nominal level is the double bootstrap. The idea is to construct confidence intervals for several different confidence levels. We then do a second layer of resampling (simulating second-order bootstrap samples on from the first set of bootstrap samples) and use these to estimate the coverage probability for each confidence level. We can then interpolate these (x = coverage probability, y = confidence level) points with a Lagrange polynomial, and calculate an appropriate confidence level that will give us the desired coverage.

Algorithm

Let $(x_1, ..., x_n)$ denote our random sample. Compute the sample mean $\bar{x} = \sum_{i=1}^{n} frac(x_i)n$.

1. Obtain $B1$ bootstrap samples by sampling with replacement from the original sample.
2. For each of the $B1$ bootstrap samples, obtain $B2$ bootstrap samples. Using these $B2$ second-order samples, construct 4 confidence intervals using the significance levels 0.10, 0.05, 0.01, 0.001, by the percentile method.

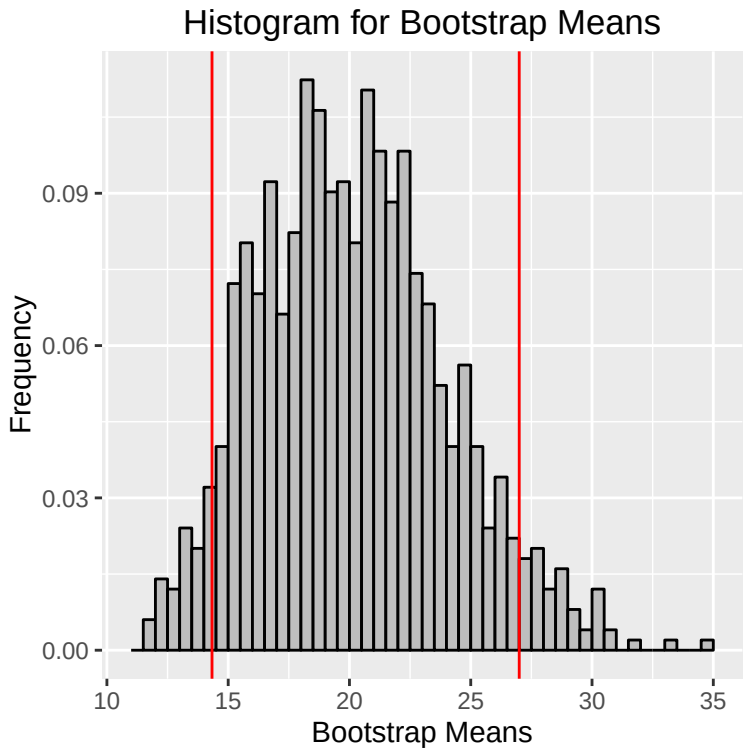

Figure 3: A histogram of means from the B bootstrap samples. The red vertical lines correspond to the cut-off points for the 90% confidence interval. The first vertical line is at the 5th percentile, the second is at the 95th percentile.

3. For each significance level considered, compute the proportion of the $B1$ intervals that contain $\bar{x}$. These proportions are the corresponding coverage probabilities for each confidence level.
4. Consider the coverage probabilities and significance levels as (x, y) coordinates and interpolate using the Lagrange polynomial $f(x)$.
5. Compute $f(0.10) = \widehat{\alpha}$.
6. Construct the the $(1 - \widehat{\alpha}) \bullet 100\%$ confidence interval by using the percentile method with $\widehat{\alpha}$ on the original $B1$ bootstrap samples.

Results

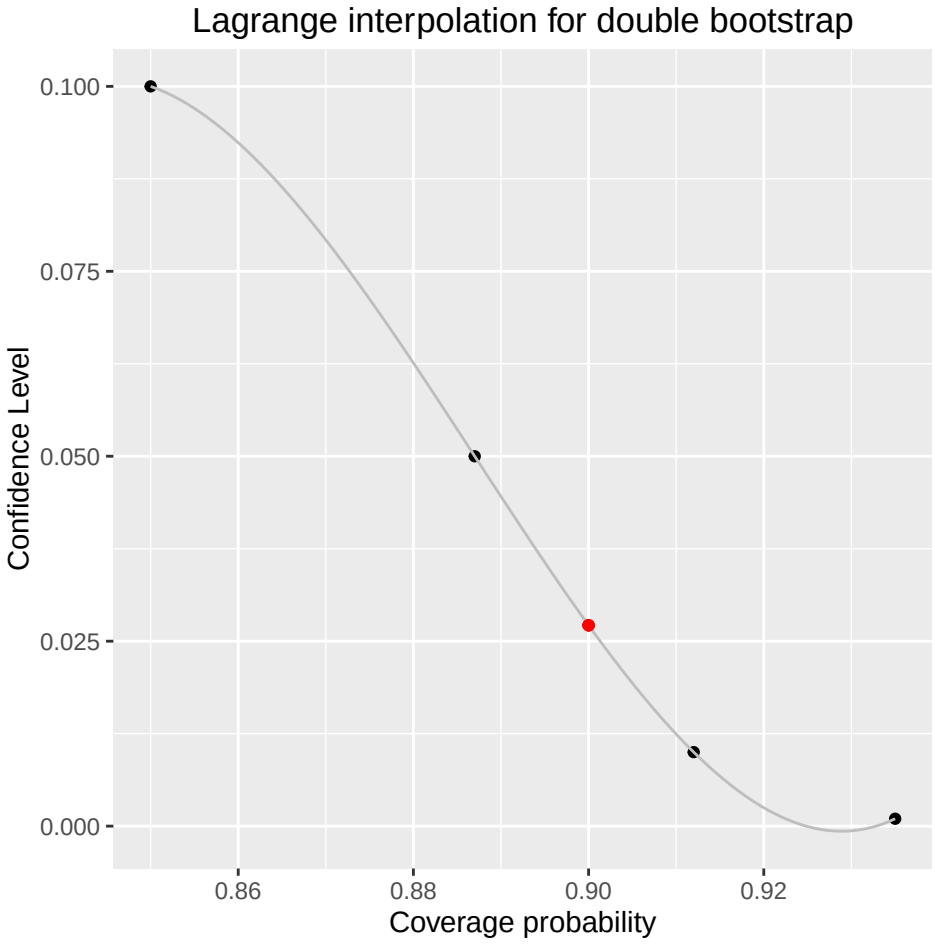

Figure 4: For the double bootstrap, four alpha levels were considered: 0.10, 0.05, 0.01, and 0.001. This graph shows the Lagrange interpolating polynomial through these four points. The red point is the estimated alpha level needed for a 90% confidence interval.

The Lagrage polynomial is shown in figure 4. For a 90% confidence interval, the estimated significance level from the double bootstrap is $\widehat{\alpha} = 0.02714687$. The resulting confidence interval is $(12.54083, 29.4575)$

Comparison

In comparing the intervals obtained from the percentile method and the double bootstrap, we see that the double bootstrap produces a wider interval. This is because of the smaller significance level used, which

allows the coverage probability is much closer to the nominal level than the percentile method. However, the two intervals are still very similar, so the percentile method provides a decent estimation for a relatively cheaper computational cost.

Appendix (R code)

```
library("ggplot2") #For nice plots.
library("knitr") #For nice tables.
library("polynom") #For Lagrange interpolation used in double bootstrap.

#Set the random number generator seed.
set.seed(3)

#Haircut data:
#   x = Amount spent on last haircut
data <- data.frame(x = c(17, 16.49, 45, 25, 12, 18, 13, 20, 10, 0, 15, 50))
x <- data$x
n <- length(data$x)

kable(matrix(c("Observation", 1:12, "Price", format(x, nsmall = 2)), nrow = 2, byrow = TRUE))

#Sample mean and standard deviation of x.
x.mean <- mean(x)
x.sd <- sd(x)

#Histogram of x.
ggplot(data.frame(Price = x), aes(Price)) +
  geom_histogram(aes(y = ..density..), breaks = seq(0, 50, 5),
                 fill = "grey", col = "black") +
  labs(title = "Histogram for Prices", x = "Price", y = "Frequency")
  #geom_vline(xintercept = x.mean, linetype = "longdash", color = "red")

##Shapiro-Wilk test
pval <- shapiro.test(x)

#Significance level used throughout.
alpha <- 0.1

#Confidence interval from normality assumption
CI <- c(x.mean + qt(alpha/2, n-1)*x.sd/sqrt(n),
        x.mean + qt(1-alpha/2, n-1)*x.sd/sqrt(n))

#QQplot of x.
ggplot(data.frame(y = (x-x.mean)/x.sd), aes(sample = y)) +
  stat_qq() +
  geom_abline(intercept = 0, slope = 1, color = "grey") +
  labs(title = "Q-Q Plot", x = "Normal theoretical quantiles",
       y = "Standardized sample prices")

B = 1000
#Create B bootstrap samples and store in an n by B matrix.
```

```r
boot_samples <- matrix(sample(x, n*B, replace = TRUE), nrow = n, byrow = FALSE)

#Compute the mean for each bootstrap sample and store in a vector of length B.
boot_means <- apply(boot_samples, 2, mean)

#We'll need to obtain percentile confidence intervals again later on, so
# here we create a function that returns a percentile CI given a vector
# of estimates.
get_percentile_CI <- function(estimates, alpha = 0.1) {
  #Get the number of estimates in the vector.
  B <- length(estimates)

  #Put the estimates in ascending order.
  estimates_ordered <- sort(estimates)

  #Use the appropriate percentiles for the upper and lower confidence limit.
  CI_percentile <- c(estimates_ordered[ceiling(B*alpha/2)],
                     estimates_ordered[ceiling(B*(1 - alpha/2))])
}

#Print out the confidence interval.
CI <- get_percentile_CI(boot_means, 0.1)

ggplot(data.frame(y = boot_means), aes(y)) +
  geom_histogram(aes(y = ..density..), breaks = seq(11, 35, 0.5),
                 fill = "grey", col = "black") +
  labs(title = "Histogram for Bootstrap Means",
       x = "Bootstrap Means", y = "Frequency") +
  geom_vline(xintercept = c(CI[1], CI[2]), color = "red")

B1 = 1000
B2 = 1000

#Confidence levels to consider.
deltas <- c(0.10, 0.05, 0.01, 0.001)

#For each confidence level, we will estimate the coverage probability.
coverage <- vector(mode = "numeric", length = length(deltas))

#Create B1 bootstrap samples and store in an n by B1 matrix.
boot_samples <- matrix(sample(x, n*B1, replace = TRUE), nrow = n, byrow = FALSE)

second_boot_means <- vector("list", B1)
for(i in 1:B1) {
  #Create B2 bootstrap samples and store in an n by B2 matrix.
  second_boot_means[[i]] <- apply(matrix(sample(boot_samples[ ,i], n*B2, replace = TRUE),
                                  nrow = n, byrow = FALSE), 2, mean)
}

j <- 1
for(delta in deltas) {
  CI <- NULL
```

```
  #Create confidence interval for each of the B2 bootstrap samples
  CI <- matrix(sapply(second_boot_means, get_percentile_CI, delta), nrow = 2)

  #Estimate the coverage probability.
  coverage[j] <- sum(CI[1, ] < x.mean & x.mean < CI[2, ])/B2
  j <- j + 1
}

#Calculate the Lagrange polynomial using (coverage, deltas) as (x, y) points.
L <- poly.calc(x = coverage, y = deltas)

#Evaluate the interpolated polynomial at x = alpha.
alpha.corrected <- predict(L, 1 - alpha)

CI <- get_percentile_CI(boot_means, alpha = alpha.corrected)

ggplot(data.frame(x = coverage, y = deltas)) +
  geom_point(aes(x = x, y = y)) +
  stat_function(fun = function(x) { predict(L, x) }, color = "grey") +
  labs(title = "Lagrange interpolation for double bootstrap",
       x = "Coverage probability", y = "Confidence Level") +
  geom_point(aes(x = 0.9, y = alpha.corrected), color = "red")
```