Bootstrapping-Methods for Linear Regression

Jack DeGroot

Section 1: Description of Problem

We consider that a bootstrap is used to learn and estimate a sampling distribution by resampling and utilizing given data; this is done by assigning measures of accuracy to sample estimates. A bootstrap percentile confidence interval, after bootstrapping, uses the calculated empirical distribution as an estimate for a sampling distribution to produce a percentile confidence interval. In order to get an accurate percentile confidence interval from bootstrapping we will need to replicate this process a very large amount of times so our percentile confidence intervals have enough data to compute. In this problem we will be comparing the percentile confidence intervals of five different bootstrap methods, run in four different simulations that involve distinct noises in a simple linear regression model. The four different bootstrap methods to create our percentile confidence intervals that we will be observing are the resampling cases, parametric, resampling errors, and wild bootstrap methods. First, we consider a simple linear regression model defined as followed:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad \text{for } i = 1, \ldots.. , n$$

where we have y_i as our ith response variable, β_0 as our intercept term, β_1 as our slope, x_i as our explanatory ith variable, and ε_i as our noise. We will be changing our noise for each of our four simulations. For our first simulation we will have standard normal noise. For our second simulation we will have heavy tail noise. For our third simulation we will have skewed noise. For our fourth simulation we will have heteroscedastic noise. Specifics on how these different noise will be implemented into our linear regression model will be discussed in section 2.

Aims:

Our goal in this problem is to compare these five different bootstrap percentile confidence intervals to observe which bootstrap method produces the most accurate percentile confidence interval for our slope. In order to determine which bootstrap method produces the most accurate percentile confidence interval we will be observing the empirical coverage probability produced from each of the simulations as well as evaluating the lengths of the percentile confidence intervals.

Section 2: Design Of Study

In Section 1 we briefly described that we will be running four simulations, each with a different noise for our linear model, to produce percentile confidence intervals based on five different methods of bootstrapping. This will result in twenty groups of simulated confidence intervals that we will evaluate on the basis of proportion that include our true value as well as the length of the confidence interval. These two metrics will provide answers to which bootstrapping method produced the most accurate percentile confidence intervals. We first start our design of this study with determining the metrics in simulation.

Data-generating Mechanisms:

Like all things in statistics/data science, the more simulations or data produced the more fair representation we will have at determining our results. specifically, in our simulation we want to make sure that we are producing enough percentile confidence intervals that will give us a fair representation of if that specific bootstrapping method creates percentile confidence intervals that hold our true β_1. For example, if we ran ten simulations for this study that would only give us a group of ten confidence intervals to evaluate if β_1 is within the lower bound and upper bound of each of the ten confidence intervals. Hence, ten simulations will not be sufficient for our study. Based on the time we have for this project and the strength of our programming language R, we will be doing 150 simulations for our study. This will provide us with 150 percentile confidence intervals for each bootstrapping method in each of our four simulations. In order to produce 150 simulations I utilize the replicate() function which allows for repeated evaluations of an expression.

Each of our four simulations have the 5 same bootstrapping methods applied to a linear regression model based on different noise. In our first simulation our linear regression model will have standard normal noise also known as gaussian noise. This kind of noise can take on values that are based on the Normal/Gaussian distribution. We implement this method into our experiment by utilizing the rnom() function which produces random deviations from our standard normal distribution . In our second simulation our linear regression model will have heavy tail noise. This kind of noise implies that for large values of x, the density approaches zero much slower compared to the standard normal noise. We implement this method by setting our error, ε_i = rt(n ,df=3)/sqrt(3) where our rt() function is for random student t distribution, and since variance of a t distribution with degree of freedom, denoted df, is df/(df-2) we divide the random samples by the square root of 3 so that the noise in this setting also has variance of 1 which is comparable to the standard normal. In our third simulation our linear regression model will have skewed noise. This kind of noise implies some sort of bias/distortion in our distribution where values can be taken from. We implement this method by setting our error, ε_i = rgamma(m ,shape=4, rate=2)-2 where rgamma() is a function that produces random deviations from our gamma distribution which is skewed with mean 2 and variance 1. We make sure we subtract 2 from our random samples so that the noise has zero mean. In our fourth simulation, our linear regression model will have heteroscedastic noise. This implies that the variance of our noise in our regression model varies widely. We implement this method by setting our error, ε_{i1} = rnorm(n)*x^2/sqrt(mean(x^4) and ε_{i2} = ε_{i1} - mean(ε_{i1}) where we see again our rnom() function. We also observe that Var(ε_i) $\propto x_i^4$ but still the variance of the esp vector is 1 and the mean is zero. All in all, these four noises will impact our linear regression model which, we think, will produce different confidence intervals based on the five bootstrap methods.

We now apply each of our five bootstrapping methods to each of our linear regression models. As mentioned we will be replicating the procedure of our bootstrap so that our percentile confidence intervals have enough data to accurately produce each of our 150 percentile confidence intervals. We implement this by utilizing our replicate() function again and setting it so that it replicates 2000 bootstrap replicates for our percentile confidence intervals to pull from. 2000 bootstrap replicates will give us plenty of data so we can compute each of the 150 percentile confidence intervals for our four simulations. We now fit our bootstrapping method to each simulation's linear regression model.

Methods:

For all cases in each bootstrapping method. A linear model is fit with n=2000 with x being n data taken from a uniform distribution form -1 to 1 where we set β_0 = 1 and β_1 = 3. The error term ε_i is equal to whatever simulation we are in. If it's simulation 1 it is standard normal noise, if it is simulation two it is heavy tail noise, if it is simulation three it is skewed noise, and finally if it is simulation four it is heteroscedastic noise. Setting our error term to the appropriate error of the specified simulation we then fit our model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ We then fit each of the bootstrapping methods.

The resampling cases bootstrap: also known as the nonparametric bootstrap, we first independently sample our x explanatory variable, creating x_i^* n number of times with replacement. Next, we set our explanatory variable for our bootstrap equal to our explanatory variable as well as set our y response variable for our bootstrap to our response variable in our linear regression model for our bootstrap replication. Our error term will be set based on whichever simulation we are running the bootstrap method in. For resampling we will not have an error for our bootstrap replication linear model denoted eps.star. Fitting the pairs of our bootstrap explanatory and response variables we produce a bootstrap linear regression model which gives us the estimates for $\widehat{\beta_0^{(b)}}$ and $\widehat{\beta_1^{(b)}}$. This is one replication of the 2000 replications that will be computed. After 2000 replications are completed, this process is replicated once again until 150 confidence intervals are produced in a list. I unlist the output, sort it, and then compute the necessary operations to find the proportion of percentile confidence intervals that include our true value of $\beta_1 = 3$. I also compute the average length of all of the percentile confidence intervals. Discussion of this chosen metric is under "Performance measures" Results of this are discussed in Section 3.

The parametric bootstrap: we first estimate the parameters from our linear regression model (based on whichever simulation) and then simulate from the estimated parametric model. We simulate our eps.star from a Normal Distribution with mean zero and standard deviation based on the linear regression model based on simulation number. Once again, we set our explanatory variable for our bootstrap equal to our explanatory variable as well as set our y response variable for our bootstrap to our response variable in our linear regression model for our bootstrap replication. We fit the pairs of our bootstrap explanatory and response variables and we produce a bootstrap linear regression model which gives us the estimates for $\widehat{\beta_0^{(b)}}$ and $\widehat{\beta_1^{(b)}}$. This is one replication of the 2000 replications that will be computed. After 2000 replications are completed, this process is replicated once again until 150 confidence intervals are produced in a list.This process is the same where I unlist the output, sort it, compute the necessary operations to find the proportion of percentile confidence intervals that include our true value of $\beta_1 = 3$ and compute the average length of all of the percentile confidence intervals. Results of this are discussed in Section 3.

The resampling errors bootstrap: we have two assumptions that the linear model is assumed and that there isn't any error distribution. We add to our linear regression model based assumption with our error being equal to $\varepsilon_i = \frac{y_i - \hat{y}}{\sqrt{1 - h_i}}$ where h_i is the hat value and our residual error equal to $r_i = \varepsilon_i - \hat{\varepsilon}$ where $\hat{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i$ With this, we independently sample our residual variable, creating r_j^* n number of times with replacement as well as set our explanatory variable for our bootstrap equal to our explanatory variable for our linear regression model. In our resampling errors case, however, we set our y response variable for our bootstrap to our response variable in our linear regression model plus our resid.star , hence $y_j^* = \widehat{y_j} + r_j^*$ for our bootstrap replication. Now, we fit the pairs of our bootstrap explanatory and response variables and we produce a bootstrap linear regression model which gives us the estimates for $\widehat{\beta_0^{(b)}}$ and $\widehat{\beta_1^{(b)}}$. This is one replication of the 2000 replications that will be computed. After 2000 replications are completed, this process is replicated once again until 150 confidence intervals are produced in a list.This process is the same where I unlist the output, sort it, compute the necessary operations to find the proportion of percentile confidence intervals that include our true value of $\beta_1 = 3$ and compute the average length of all of the percentile confidence intervals. Results of this are discussed in Section 3.

The wild bootstrap: we want to simulate the errors for each observation from different distributions. Like resampling errors we add to our linear regression model based assumption with our error being equal to $\varepsilon_i = \frac{y_i - \hat{y}}{\sqrt{1 - h_i}}$ where h_i is the hat value and our residual error equal to $r_i = \varepsilon_i - \hat{\varepsilon}$ where $\hat{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i$ as well as we add the term v_j^* where v_j^* are independent and identically distributed random variables from Normal Distribution with mean zero and variance one. With this, we independently sample our residual variable, creating r_j^* n number of times with replacement as well as set our explanatory variable for our bootstrap equal to our explanatory variable for our linear regression model. In our wild bootstrap case, however, we set our y response variable for our bootstrap to our response variable in our linear regression model plus our residuals multiplied by v_j^*, hence $y_j^* = \widehat{y_j} + r_j (v_j^*)$ for our bootstrap replication. Now, we fit the pairs of our bootstrap explanatory and response variables and we produce a bootstrap linear regression model which gives us the estimates for $\widehat{\beta_0^{(b)}}$ and $\widehat{\beta_1^{(b)}}$. This is one replication of the 2000 replications that will be computed. After 2000 replications are completed, this process is replicated once again until 150 confidence intervals are produced in a list.This process is the same where I unlist the output, sort it, compute the necessary operations to find the proportion of percentile confidence intervals that include our true value of $\beta_1 = 3$ and compute the average length of all of the percentile confidence intervals. Results of this are discussed in Section 3.

Performance Measures:

After each simulation finishes fitting their linear regression model to each bootstrapping method and runs through all the associated replications, each simulation will produce a summary table. The summary table will include the number of percentile confidence intervals that include $\beta_1 = 3$ in their percentile confidence intervals for each bootstrap method. This consequently will be used to produce the proportion of percentile confidence intervals for each bootstrapping method that included $\beta_1 = 3$ in the 150 percentile confidence intervals produced. The closer the proportion is to 1.00 means that the 100% of the 150 confidence intervals produced from the associated bootstrap method included $\beta_1 = 3$ in their percentile confidence intervals. The closer the proportion is to 0.00 means that the 0% of the 150 confidence intervals produced from the associated bootstrap method included $\beta_1 = 3$ in their percentile confidence intervals. So we will be looking for proportions that are closer to 1.00. We will consider the length of those confidence intervals to understand if those confidence intervals are truly precise in their accuracy.

The summary table will also consist of the average percentile confidence interval length of each bootstrapping method. Computing the average length will be a sufficient representation and an easy way to compare each of the confidence interval's lower bound and upper bound values with seeing the difference (length) between them. The computation for the average length is taking the mean of the upper bound and lower bound of all associated 150 percentile confidence intervals for each bootstrap method in whichever specified simulation. We will be looking for averages that are small and closer to zero. The smaller the average means that the length for the confidence interval was on average more precise on determining $\beta_1 = 3$.The wider the length of the confidence interval the less precise our confidence interval is. For example if we are looking for confidence intervals where $\beta_1 = 3$ and we have a 95% percentile confidence interval that is [1.50000 , 4.50000] . We say that we are 95% confident the true value is between 1.5 and 4.5. While our value 3 is included in that confidence interval, the length of the confidence interval is very wide. Hence, the confidence interval may be accurate in having the true value within it, however the precision of that confidence interval is not suitable because there is a large range of numbers that can be drawn out within that interval. A more accurate confidence interval would be [2.929292 , 3.112795] which has a length of 0.183503. This smaller length gives us a more precise interval on determining the true value of β_1 making that interval much more accurate when compared to the previous interval of [1.50000 , 4.50000]. In total there is one summary table for each simulation, so four summary simulation tables in total.

There are two additional tables as well, the first is the whole study summary of the proportion of percentile confidence intervals for each bootstrapping method that included $\beta_1 = 3$ in their percentile confidence intervals based on bootstrapping method (x-column) and simulation number (y-column). The second table consists of the average percentile confidence interval length based on bootstrapping method (x-column) and simulation number (y-column). We will now discuss these tables and results in Section 3.

Section 3: Results and Discussion

Table 1: Simulation 1 Results (Shown Below)

Simulation 1	Number of C.I. That Include B1 = 3	Proportion of C.I. that Include B1 = 3	Average Length of C.I.
ReSampling Case Bootstrap	140	0.9333333	0.1274949
Parametric Bootstrap	136	0.9066667	0.1267270
ReSampling Errors Bootstrap	136	0.9066667	0.1273656
Wild Bootstrap	124	0.8266667	0.1277831

We observe the ReSampling case bootstrap and see that it produced the second highest proportion in proportion of Confidence Intervals that included our $\beta_1 = 3$ while also having the second shortest average length out of all the other bootstraps. From this, we can determine that for a linear regression model with standard normal noise, the resampling bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$

Table 2: Simulation 2 Results (Shown Below)

Simulation 2	Number of C.I. That Include B1 = 3	Proportion of C.I. that Include B1 = 3	Average Length of C.I.
ReSampling Case Bootstrap	138	0.9200000	0.1252067
Parametric Bootstrap	133	0.8866667	0.1262886
ReSampling Errors Bootstrap	131	0.8733333	0.1282585
Wild Bootstrap	139	0.9266667	0.1260277

We now observe the Wild bootstrap and see that it produced the second highest proportion. We also observe that the wild bootstrap had the second shortest average length out of all bootstraps. We take into consideration that the resampling case bootstrap produced a proportion smaller than wild bootstrap which was the result of one less confidence interval including $\beta_1 = 3$. We also see that the average length of the ReSampling bootstrap is 0.000821 shorter than the wild bootstrap. Even though, the wild bootstrap performed one confidence interval better in capturing the true β_1

= 3 value the resampling bootstrap provided a tighter interval that was nearly as accurate in capturing $\beta_1 = 3$

in its percentile confidence interval. From this, we can determine that for a linear regression model with heavy tail noise, the resampling case bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$.

Table 3: Simulation 3 Results (Shown Below)

Simulation 3	Number of C.I. That Include B1 = 3	Proportion of C.I. that Include B1 = 3	Average Length of C.I.
ReSampling Case Bootstrap	125	0.8333333	0.1271938
Parametric Bootstrap	131	0.8733333	0.1275900
ReSampling Errors Bootstrap	136	0.9066667	0.1273264
Wild Bootstrap	124	0.8266667	0.1263581

We observe the ReSampling Errors bootstrap and see that it produced the second highest proportion. The ReSampling Errors bootstrap also produced the smallest average length of confidence intervals. This is a clear indication that the ReSampling Errors bootstrap performed best in this simulation. We can determine that for a linear regression model with skewed noise, the ReSampling Errors bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$.

Table 4: Simulation 4 Results (Shown Below)

Simulation 4	Number of C.I. That Include B1 = 3	Proportion of C.I. that Include B1 = 3	Average Length of C.I.
ReSampling Case Bootstrap	139	0.9266667	0.1866798
Parametric Bootstrap	109	0.7266667	0.1269003
ReSampling Errors Bootstrap	110	0.7333333	0.1272146
Wild Bootstrap	140	0.9333333	0.1864845

We observe the Wild bootstrap and see that it produced the second highest proportion, however it did produce the 3rd lowest average length of confidence intervals. Looking at the two bootstrap methods, Parametric and ReSampling, that produced the lowest two average lengths of confidence intervals, we can see that they did not perform well in capturing the true value of $\beta_1 = 3$ in their confidence intervals. This low proportion does not outweigh the larger width of the wild bootstraps average length in confidence intervals. We note that ReSampling bootstrap has a high proportion of confidence intervals that include $\beta_1 = 3$ however it was 1 confidence interval short of wild bootstrap. In addition, the wild bootstrap has a smaller average length in its confidence intervals compared to resampling. From this, we can determine that for a linear regression model with heteroscedastic noise, the wild bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$.

Table 5: Summary Table for Proportion of Confidence Intervals that Include $\beta_1 = 3$

Proportion of C.I. That Include B1 = 3	Simulation 1	Simulation 2	Simulation 3	Simulation 4
ReSampling Case Bootstrap	0.9333333	0.9200000	0.8333333	0.9266667
Parametric Bootstrap	0.9066667	0.8866667	0.8733333	0.7266667
ReSampling Errors Bootstrap	0.9066667	0.8733333	0.9066667	0.7333333
Wild Bootstrap	0.8266667	0.9266667	0.8266667	0.9333333

From our study summary table above, we can see that the resampling case bootstrap produced the most consistent percentile confidence intervals that included $\beta_1 = 3$. We see that the resampling case bootstrap struggled when fitting to a linear regression model with skewed noise; however it did not perform the worst in that simulation. We observe that our Parametric bootstrap was the worst bootstrap overall in capturing $\beta_1 = 3$ in its percentile confidence intervals. Parametric struggled the most when it tried to fit with a linear regression model with heteroscedastic noise. For our ReSampling Errors bootstrap, it performed the best when being fit to a linear regression model with skewed noise, however it struggled the most when trying to fit a linear regression model with heteroscedastic noise. Finally, our wild bootstrap was the second most consistent in producing percentile confidence intervals that included $\beta_1 = 3$. The wild bootstrap struggled with fitting linear regression models with standard normal noise as well as linear regression models with skewed noise.

Table 6: Summary of Average Confidence Interval Lengths

Average Length of C.I.	Simulation 1	Simulation 2	Simulation 3	Simulation 4
ReSampling Case Bootstrap	0.1274949	0.1252067	0.1271938	0.1866798
Parametric Bootstrap	0.1267270	0.1262886	0.1275900	0.1269003
ReSampling Errors Bootstrap	0.1273656	0.1282585	0.1273264	0.1272146
Wild Bootstrap	0.1277831	0.1260277	0.1263581	0.1864845

Between each simulation of each bootstrap we see that there is not much difference between their average lengths. Between each bootstrap method within a simulation there is not much difference between their average lengths. Overall, the main takeaway from the average confidence interval length is that most of the methods in each simulation were a thousandths digit away from one another which meant that most of the percentile confidence intervals were very precise in their accuracy.

Section 4: Conclusion

We conclude that for a linear regression model with standard normal noise and a linear regression model with heavy tail noise, that the resampling case bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$. In addition, we conclude that for a linear regression model with skewed noise, the ReSampling Errors bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$. Finally, we conclude that for a linear regression model with heteroscedastic noise, the wild bootstrap provided us with the most accurate percentile confidence intervals for capturing $\beta_1 = 3$.