1. Deriving .

Define as follows:

Set to find critical points of .

Thus, the only critical point of is at .

Since , then we obtain

Therefore, has an absolute minimum at , so does indeed minimize , and by extension .

1. Bootstrap Sample with *n* observations
   1. The probability that the first bootstrap observation is not the jth observation from the original sample is 1-1/n. There are n samples, and only 1 of them is the jth observation, so the chance of the first observation being the jth observation is 1/n, and the complement of that (chance of first observation NOT being jth observation) is 1-1/n.
   2. Since bootstrap takes observations *with replacement*, every observation selection is an independent event, as there will always be n observations to choose from in every selection. Therefore, for all positive integers k, the chance that the kth observation selected is not the jth observation is the same (1-1/n). Therefore, the probability that the second bootstrap observation is *not* the jth observation from the original sample is 1-1/n.
   3. For all 1<=j<=n, the chance of the jth observation not being selected during any individual selection for the bootstrap set is 1-1/n. Since the bootstrap sample is the same size as the original dataset, there will be n individual selections to be made. Therefore, for any observation j, the chance that it will not be included in the bootstrap sample, or the chance that it will not be chosen in any of the n individual selections in the bootstrapping process, is (1-1/n)^n.
   4. When n=5, the probability that the jth observation *is* in the bootstrap sample is 1- the chance that it isn’t included, or 1-(1-1/n)^n=1-(1-1/5)^5=1-(4/5)^5=0.67232, or about 67%.
   5. Similarly, when n=100, the probability that the jth observation *is* in the bootstrap sample is about 63.4%.
   6. Similarly, when n=10000, the probability that the jth observation *is* in the bootstrap sample is about 63.2%.
   7. When plotting the probability that the jth observation is in the bootstrap sample with every integer value from 1 to 100000, the graph starts at 1, since if there is only one observation, then it must be in the bootstrap sample. It steeply jumps to 0.75 with two observations, then to about 0.7 with three observations, then proceeding to have the difference in probability between adjacent number of observations become extremely small. As the graph approaches about 100, the differences in probability are essentially negligible, with the plot behaving like there is an asymptote at about 0.63, as the graph appears essentially horizontal past 1000, with even n=100000 yielding a probability of about 0.63.
   8. After performing this numerical experiment, the chance of the bootstrap with 100 samples containing the 4th observation, averaged out over 10000 different bootstrap samples, is about 63.4%. This is basically equivalent to the theoretical odds of 63.4%.
2. K-fold cross-validation
   1. K-fold cross-validation is implemented first by splitting up the observation set into k different groups (folds) of essentially equal size. Then, starting with the first fold, that fold is treated as the test set, with the rest of the data being used to train the model. This is repeated for all of the available k folds, with mean squared errors being computed for each, and the k-fold CV estimate is computing by taking the mean of every MS value.
   2. K-fold cross-validation advantages
      1. The validation set approach, while computationally quicker, has much higher variability than k-fold cross-validation. This is because of the fact that k-fold cross-validation essentially performs the validation set approach k times, taking the mean of all of the k validation sets. This leads to a more accurate representation of the true error, due to how having k different validation sets, each with somewhat different results, and averaging them out is a better estimate of the true error than simply taking one of the test errors, which could be a significant outlier and lead to a much higher/lower test error than should be occurring. As well, the validation set approach tends to overestimate the test error due to only half of the data set being used to train the model.
      2. LOOCV is a special case of k-fold cross-validation, but with k=n, the number of observations in the set. It is therefore much more computationally expensive than k-fold cross validation, as instead of having to make 5 or 10 models for cross-validation, it must make n different sets. As well, it has higher variance due to how each of the n folds are highly positively correlated with each other, since all of the n training sets only differ by one value.
3. If we were to use some statistical learning method to predict response Y for predictor X, and we needed to estimate the standard deviation, we would first use bootstrap sampling to create a bootstrap sample (implying with replacement). Then, we would estimate the predictor on this particular bootstrap sample, and repeat the above two steps a large number of times, in effect producing a very large number of bootstrap samples. Finally, we would compute the standard deviation of all of the estimates using the corresponding standard deviation estimate formula.