**Examining Alternative Estimators through a Simulation Study in R**

**Abstract**

 By the central limit theorem, the mean of a random sample has been known to be a good estimator for evaluating the true mean of a distribution. But are there other estimators that can perform as well or better? It’s known that a trimmed mean (mean with a fixed number of extreme values removed) and the median (maximal amount of trimming) have been unbiased estimators for symmetric distributions. We would like to examine if the trimmed mean or median could perform as well as the sample mean on a mixture of two normal distributions. Our results suggest that there is a positive relationship between trimming and performance when a random sample contains outliers (contamination). Although this relationship exists in cases with high contamination, differences in efficiency between the sample mean and trimmed mean are not strong as in the low contamination case. When no contamination exists the three estimators show no noticeable difference in efficiency.

**Introduction**

 The Monte Carlo method can be used to estimate an unknown parameter, θ of a distribution. This method is based on repeated random sampling to estimate the true value of θ. When the sample is generated from a known distribution this process is known as parametric bootstrap method (Rizzo, p 162). For instance, sampling from the normal distribution and calculating the sample mean ($\overbar{X}$) again and again to estimate the true mean, µ is a prime implementation of the parametric bootstrap method. By the central limit theorem, the distribution of $\overbar{X} $is standard, normally distributed with the mean at µ (Rizzo, p.35). Thus, the mean of this collection of sample means will produce a quality estimation of θ.

If a distribution is symmetric, then the trimmed mean $\hat{X}$ and median $\hat{X}\_{0.5}$ could also be used in combination with the Monte Carlo method to find µ. A trimmed mean of a sample is the mean of a sample with a defined percentage of the largest and smallest values removed. Trimmed means are said to be robust estimators since the estimator is insensitive to extreme values (Rizzo p.158). The median is also a trimmed mean. However, the median is a special case of trimming where the maximum number of values have been trimmed off leaving a central value. It is unknown how effective $\hat{X}$ and $\hat{X}\_{0.5}$ are in approximating µ of a distribution. In this study, we compare the performance of $\hat{X}$ and $\hat{X}\_{0.5}$ against$ \overbar{X }$on a random sample with varying amounts of outliers using the mean square error as an indicator of estimator efficiency.

The unknown distribution that will be under study is a mixture distribution of two normal distributions with identical means. A mixture distribution is a weighted sum of random variables from multiple probability distributions (Rizzo, p. 63)

$F\_{X}\left(x\right)=\sum\_{}^{}p\_{i}F\_{X\_{i}}(x)$,

where the sum of the mixing probabilities pi add to one. Mixing two normal distributions can reveal how factors such as dispersion and outliers affect the efficiency of these estimators. The “contamination” of one distribution onto another from mixing can allow for some controlled variability. Since the two distributions are normal they are both symmetric about the mean. Also, the identical means centers the two distributions onto the same point. This allows for a precise effect from other factors on the estimations. Parameters such as sample size, mixing probabilities, µ, σ, and a multiplying constant k will be altered to examine the estimators $\overbar{X}$, $\hat{X}$, and$ \hat{X}\_{0.5}$ using the Monte Carlo Method.

**Methods**

 The open source, statistical software R (Version 3.0.2) was used for the simulations of this study.

The model used to generate the random samples {X1, X2, X3,…, Xn} was a mixture of two normal distributions given by,

$$X= pN\left(µ, σ^{2}\right)+ \left(1-p\right)N\left(µ, k^{2}σ^{2}\right) $$

where p is the mixing probability of $X$ taking on a value from$ N(µ, σ^{2})$.

This model was examined under various parameter settings. *Table 1* below, summarizes the values of the parameters used in the model. All random variables were generated from a combination of these five parameters, and every possible combination (total of 48) was evaluated using Bias (1) and Mean Square Error (MSE; 2).

*Table 1.* The various parameters used to run the model. The size of the sample is n, the true mean is µ, standard deviation is σ, the constant multiplier of the standard deviation is k, and the mixing proportion is p.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | µ | σ | k | p |
| 30, 100 | 0, 5 | 1, 5 | 2, 10 | 0.3, 0.5, 0.7 |

The two normal distributions that the mixture sample is generated from share an identical mean value, but one of the variance (k2σ2) is multiplied by a scalar k2. Thus our model is a mixture of two normal distributions centered at a single point with $N(µ, k^{2}σ^{2}) $containing heavier tails. Altering the values of σ, k, or p allows for the manipulation of contamination in this study.

The three estimators tested in the study are the sample mean ($\overbar{X}$), a 10% trimmed mean ($\hat{X}\_{0.1}$), and the median ($\hat{X}\_{0.5}$). To find these estimators of µ, a Monte Carlo simulation was used. After a random sample *x1, x2, x3,…, xn*was generated from our model, the mean and median were found using the built in functions mean() and median(). In order to find the trimmed mean, the sample was first ordered in magnitude from smallest to largest. Then, the number of observations to be trimmed off of each extreme (t) was calculated by

$$t=0.10n.$$

The sample mean of this new sample was then taken to be $\hat{X}\_{0.1}$ (Rizzo, p. 156). This process was repeated 500 times to produce 500 estimations of $\overbar{X}$, $\hat{X}\_{0.1}$, and $\hat{X}\_{0.5}$ . Following this, the final estimators were found. Let $\hat{θ\_{1}}$, $\hat{θ\_{2}}$, and $\hat{θ\_{3}}$ represent the estimators of $\overbar{X}$, $\hat{X}\_{0.1}$, and $\hat{X}\_{0.5}$. Then,

The bias (*bi*) of the ith estimator was calculated by:

$b\_{i}=E\left[\hat{θ}\_{i}\right]-μ, $ for i= 1, 2, 3 (1)

The MSE was used as the performance evaluator for this study, and is given by:

$MSE\left(\hat{θ}\_{i}\right)=E\left[\left(\hat{θ}\_{i}-μ\right)^{2}\right] $ (2)

The second moment of the centered $\hat{θ}\_{i}$ about the true mean. Alternatively,

$$MSE\left(\hat{θ}\_{i}\right)=Var\left(\hat{θ}\_{i}\right)+ (E\left[\hat{θ}\_{i}\right]-μ)^{2} $$

If our estimator of $μ$ is an unbiased one, we would expect $MSE\left(\hat{θ}\_{i}\right)$ = $Var\left(\hat{θ}\_{i}\right)$. Thus, within unbiased family of estimators, the best estimator should produce the smallest MSE.

**Results**

For a comprehensive list of the results under all the parameter combinations see the table in the appendix. *Table 2* below represents a few comparative cases where all parameters except µ constant. It appears that changing the value of µ does not seem to have an effect on the overall behavior of the estimators in the model. No noticeable pattern is apparent when µ is changed from zero to five.

*Table 2.* A sample of the scenario when the true mean, µ alternates from zero to five. All other parameters are kept constant between alternations.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 0, 1, 2, 0.3, 30 | 1.475 | -0.002 | -0.008 | -0.014 | 0.117 | 0.116 | 0.137 |
| 5, 1, 2, 0.3, 30 | 1.825 | 0.023 | 0.025 | 0.019 | 0.11 | 0.101 | 0.123 |
| 0, 1, 2, 0.3, 100 | 1.587 | -0.003 | -0.002 | 0 | 0.031 | 0.028 | 0.035 |
| 5, 1, 2, 0.3, 100 | 1.684 | -0.002 | -0.002 | -0.003 | 0.028 | 0.028 | 0.036 |
| 0, 1, 2, 0.5, 30 | 1.206 | 0.014 | 0.012 | -0.002 | 0.085 | 0.077 | 0.094 |
| 5, 1, 2, 0.5, 30 | 1.648 | 0.003 | 0.005 | 0.002 | 0.08 | 0.071 | 0.078 |
| 0, 1, 2, 0.5, 100 | 1.452 | 0.002 | 0.005 | 0.014 | 0.024 | 0.02 | 0.025 |
| 5, 1, 2, 0.5, 100 | 1.628 | 0 | 0.003 | 0.005 | 0.023 | 0.021 | 0.026 |
| 0, 1, 2, 0.7, 30 | 1.4 | -0.001 | -0.004 | -0.007 | 0.07 | 0.061 | 0.076 |
| 5, 1, 2, 0.7, 30 | 1.741 | -0.001 | 0 | 0 | 0.074 | 0.065 | 0.079 |
| 0, 1, 2, 0.7, 100 | 1.351 | 0.007 | 0.007 | 0.009 | 0.02 | 0.017 | 0.023 |
| 5, 1, 2, 0.7, 100 | 1.277 | -0.012 | -0.012 | -0.016 | 0.018 | 0.015 | 0.02 |

The values of σ changed from one to five in this study, and the results are shown in *table 3*. Changing σ seems to make the performance of all estimators worse. The bias and MSE both increased when the standard deviation was increased.

**Table 3.**  Changes in performance were examined when the parameter σ was changed from one to five. Regardless of the values of other parameters, σ directly increases the bias as well as the MSE for all three estimators.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 0, 1, 2, 0.3, 30 | 1.475 | -0.002 | -0.008 | -0.014 | 0.117 | 0.116 | 0.137 |
| 0, 5, 2, 0.3, 30 | 8.019 | 0.051 | 0.089 | 0.075 | 2.423 | 2.401 | 2.76 |
| 0, 1, 2, 0.3, 100 | 1.587 | -0.003 | -0.002 | 0 | 0.031 | 0.028 | 0.035 |
| 0, 5, 2, 0.3, 100 | 10.39 | 0.021 | 0.019 | -0.014 | 0.763 | 0.73 | 0.911 |
| 0, 1, 2, 0.5, 30 | 1.206 | 0.014 | 0.012 | -0.002 | 0.085 | 0.077 | 0.094 |
| 0, 5, 2, 0.5, 30 | 9.127 | 0.044 | 0.032 | 0.035 | 2.133 | 1.854 | 2.397 |

When n was increased from 30 to 100, the MSE decreased among three parameters (see table in appendix). *Table 4* below reports a selected instances of this phenomenon. Also, *table 4* shows that standard deviation had an inverse relationship with the value of p. As p increased the value of the standard deviation dropped. Interestingly, the MSE also had an inverse relationship with p.

**Table 4.**  Changes in sample size (n) and mixing probability (p) showed that as n or p increased, the standard deviation and MSE decreased.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 0, 1, 10, 0.3, 30 | 9.634 | -0.001 | -0.025 | -0.021 | 2.334 | 1.976 | 0.493 |
| 0, 1, 10, 0.3, 100 | 8.652 | -0.151 | -0.14 | -0.042 | 0.782 | 0.675 | 0.139 |
| 0, 1, 10, 0.5, 30 | 8.98 | 0.036 | 0.052 | 0.011 | 1.667 | 1.059 | 0.177 |
| 0, 1, 10, 0.5, 100 | 5.875 | -0.015 | -0.014 | -0.01 | 0.518 | 0.353 | 0.044 |
| 0, 1, 10, 0.7, 30 | 5.015 | -0.004 | 0.005 | 0.009 | 0.999 | 0.338 | 0.099 |
| 0, 1, 10, 0.7, 100 | 4.555 | 0.004 | -0.004 | 0 | 0.324 | 0.088 | 0.029 |

The effects of k were also examined when all other parameters were fixed. These effects were tested again with a large value of p. **Fig 1** shows the performance of the three estimators when p=0.7 and k=2 (low contamination from the mixture distribution). All three estimators showed to have a similar distribution for this trial. *Table 5*, row 1 indicates that all three estimators have similar bias and MSE values.



**Figure 1.**  The parameters for this simulation are σ=0, σ=1, k=2, p=0.7, n=100. **Fig 1** represents a situation where the constant k (k=2) is small and there is little contamination from the mixture. The majority of data points will come from the standard normal distribution. **Fig 1a** is the distribution for the sample mean, **Fig 1b** is the distribution for the 10% trimmed mean, and **Fig 1c** is the distribution for the median. All three methods yield similar estimations of µ.

**Table 5.**  Comparison of performance when contamination is low, but effects of contamination are strong (k is increased to 10 in row 2). Parameters in row one and two were used for the generation of **Figure 1** and F**igure 2** respectively. Notice that MSE and standard deviation both increase as k is increased, and MSE is reduced if estimators involved trimming.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 0, 1, 2, 0.7, 100 | 1.351 | 0.007 | 0.007 | 0.009 | 0.02 | 0.017 | 0.023 |
| 0, 1, 10, 0.7, 100 | 4.555 | 0.004 | -0.004 | 0 | 0.324 | 0.088 | 0.029 |



**Figure 2.**  The parameters for this simulation are σ=0, σ=1, k=10, p=0.7, n=100. **Fig 2** represents a situation where the multiplying constant k is large (k=10) and there is still a low contamination from the mixture. The majority of data points again come from the standard normal distribution. However, because of the large k value there is a few extreme values mixed in. **Fig 2a** is the distribution for the sample mean, **Fig 2b** is the distribution for the 10% trimmed mean, and **Fig 2c** is the distribution for the median. As the trim level gets higher the estimation of µ gets better.

In **Figure 2,** k was increased from 2 to 10. Comparing **Figure 2a** with **Figure 2b** the distribution of possible values the estimator takes narrows down when 10% of the sample is trimmed off. If the maximum is trimmed off and the median is used for the estimator the distribution slims down even further **Figure 2c**. Also, in row 2 of *table 5* the MSE decreases as the sample is trimmed more and more. This indicates that trimmed mean is a better estimate of µ in cases of low contamination and small multiplier values, and the median seems to be the best. Note, that the MSE and the standard deviation increases when comparing rows 1 (k=2) and 2 (k=10) of table six.

 A similar experiment was examined with the two values of k, except the amount of contamination was changed from low (0.7) to high (p=0.3). As **Fig 3** shows, low values of k will not change the effectiveness of these three estimators even in high contamination situations. Again, the sample mean, trimmed mean, and median method all share similar MSE values (*Table 6,* row 1*)*.

**Table 6.**  Contamination of the mixture distribution was increased to the maximum. Increasing k caused a similar difference above as in *table 5.* However, unlike the scenario when contamination is low, The MSE between the trimmed mean and sample mean estimators did not differ as much. When contamination was increased, and effects of contamination were strong the median had the best performance (MSE=0.139).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 0, 1, 2, 0.3, 100 | 1.587 | -0.003 | -0.002 | 0 | 0.031 | 0.028 | 0.035 |
| 0, 1, 10, 0.3, 100 | 8.652 | -0.151 | -0.14 | -0.042 | 0.782 | 0.675 | 0.139 |



**Figure 3.**  The parameters for this simulation are σ=0, σ=1, k=2, p=0.3, n=100. **Fig 3** represents high contamination from the mixture, but a small multiplying constant k=2. The distribution of this simulation is similar to **Fig 1** except there is a larger variance. **Fig 3a** is the distribution for the sample mean, **Fig 3b** is the distribution for the 10% trimmed mean, and **Fig 3c** is the distribution for the median. As in Fig 1 all three methods yield a similar distribution of the estimators.

Lastly, when the value of k=10 in a high contamination simulation (row two of *table 6*), the median estimator again showed to be the best estimator (**Fig 4**). However, the trimmed mean distribution does not seem to visually differ from the sample mean distribution (**Fig 4a** and **Fig 4b**). Unlike **Fig 2** we don’t have this pattern of a steadily, increasing efficiency as more is trimmed off. The effectiveness of the median method may be the best out of the three in this situation (**Fig 4c**) but it’s performance does not compare to the effectiveness of median method in a low contamination simulation (**Fig 2c,** *table 6 row 2*).



**Figure 4.**  The parameters for this simulation are σ=0, σ=1, k=10, p=0.3, n=100. **Fig 4** represents high contamination from the mixture with a large multiplying constant k=10. There are more extreme values mixed into this distribution. **Fig 3a** is the distribution for the sample mean, **Fig 3b** is the distribution for the 10% trimmed mean, and **Fig 3c** is the distribution for the median. The median method is more efficient although not as efficient as in the case of low contamination.

**Discussion**

 When changing µ in this study, the effectiveness of the three methods of estimation did not change. This is due to the fact that our mixture is

$$X= pN(µ, σ^{2}) + (1-p) N(µ, k^{2}σ^{2}) $$

 Both of the distributions in this mixture are normal and centered on the same value. Changing the value of µ would only shift the mixture’s distribution left or right. The only time the true value of µ was used in statistical analysis portion of this study is when the bias was calculated. There was no apparent pattern that emerged from the biases reported in *Table 2*. Since $\overbar{X}$ is an unbiased estimator of µ, and the normal distribution is symmetric, the trimmed means will also be unbiased. The median $\hat{X}\_{0.5}$ is the maximum a sample can be trimmed. It is also unbiased since the median and mean are the same value in a normal distribution. Therefore, we can expect that all three estimators will produce similar biases.

 Increasing $σ$ from 1 to 5 increased the MSE across all three estimators (*Table 3)*. The larger $σ $is, the more disperse the values of the sample are. This in combination with a relatively small sample size causes our sample mean, trimmed mean, and median to be further away from the center of our distribution than when $σ $is smaller.

 The increase in n from 30 to 100 also produced an overall decrease in MSE (*table 4)*. According to the law of large numbers, increasing n makes the sample mean a better estimator of the true mean (Rizzo, p. 35). Thus our estimation of µ using the sample mean should give us a better estimate of the bias when n is increased. This in turn would decrease the value of the MSE since

$$MSE\left(\hat{θ}\_{i}\right)=Var\left(\hat{θ}\_{i}\right)+ (E\left[\hat{θ}\_{i}\right]-μ)^{2}$$

Likewise, the trimmed mean and median would perform better as expected since there are more sample points and the extreme values are cut off.

 The parameter p controlled the amount of contamination from$ N(µ, k^{2}σ^{2})$. When p took on a larger value the mixture had little contamination. Likewise, a smaller value of p created a larger amount of contamination. When p=0.7, the standard deviation and MSE decreased (*table 4)*. The small amount of contamination produced less values taken from $N(µ, k^{2}σ^{2})$. This meant there was a smaller chance of high dispersion in the sample. As a result the standard deviation would be lower for low contaminated simulations. Also, a reduction in dispersion would cause$\overbar{X}$, $\hat{X}\_{0.1}$, and $\hat{X}\_{0.5}$ to become better estimators by reducing the variance of the estimator.

 When the effects of k were examined at large values of p (p=0.7) and large k (k=10), it was found that $\hat{X}\_{0.5}$ was the best estimator of $µ $ (**Figure 2**). The small amount of contamination combined with the trimming of 50% of the values caused the median estimator to produce the smallest MSE between the three estimators MSE($\hat{X}\_{0.5 })=0.029; Table 5)$ . Trimming of 10% also produced a better estimate than the sample mean although it did not have as strong as an effect as trimming 50% MSE($\hat{X}\_{0.1})=0.088; MSE(\overbar{X})=0.324;Table 5$).

 If k=2 in either the high (**Figure 1**) or low contamination (**Figure 3**) scenarios, there was not a noticeable difference between the three estimators. Even though the sample was from a mixture distribution, both normal distribution had similar variances. Thus, the mixture sample resembled a sample taken from a standard, normal distribution. Therefore, there is not a high expectation of difference between the three methods when k=2. However, When high contamination was present (p=0.3; *Table 6* row 2) with large values of k (k=10), the effects of the trimming methods were dampened when there was low contamination (p=0.7; *table 5* row 2). In either case, $\hat{X}\_{0.5}$ was still the best estimator. The effectiveness of $\hat{X}\_{0.1}$ was questionable with high contamination (**Figure 4**). This could be due to a large number of samples taken from the highly dispersed distribution. Trimming off ten percent may not be enough to show a noticeable difference in performance between $\hat{X}\_{0.1}$ and$ \overbar{X}$. The optimal trimming for strong performance could be determined in a future study. It seems than when contamination exists and the multiplying constant k is high in our study, the estimator $\hat{X}\_{0.5}$ is a good estimator of the true mean. When the distributions being mixed look similar, then the methods don’t seem to matter.

 One of the limitations of this study was that only mixtures with identical µ values were examined. The performance of these estimators may change if the distributions had different µ values. This could be a further study to look into. It was also undetermined if these differences and changes in the parameters caused significant changes in the results. Hypothesis testing in a future study could provide a more holistic view of the effects of these parameters.

**References**

Rizzo, M.L. *Statistical Computing with R*. Chapman & Hall/CRC, Boca Raton, FL, 2008. ISBN 1-58488-545-9.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

**Appendix**

Below is a table of the standard deviation, Bias, and MSE of all three estimators. This table is exhaustive of every parameter combination used in this study.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RowNo. | Parametersµ, σ, k, p, n | StandardDeviation | BiasSample Mean | BiasTrimmedMean | BiasMedian | MSESample Mean | MSETrimmedMean | MSEMedian |
| 1 | 0, 1, 2, 0.3, 30 | 1.475 | -0.002 | -0.008 | -0.014 | 0.117 | 0.116 | 0.137 |
| 2 | 0, 1, 2, 0.3, 100 | 1.587 | -0.003 | -0.002 | 0 | 0.031 | 0.028 | 0.035 |
| 3 | 0, 1, 2, 0.5, 30 | 1.206 | 0.014 | 0.012 | -0.002 | 0.085 | 0.077 | 0.094 |
| 4 | 0, 1, 2, 0.5, 100 | 1.452 | 0.002 | 0.005 | 0.014 | 0.024 | 0.02 | 0.025 |
| 5 | 0, 1, 2, 0.7, 30 | 1.4 | -0.001 | -0.004 | -0.007 | 0.07 | 0.061 | 0.076 |
| 6 | 0, 1, 2, 0.7, 100 | 1.351 | 0.007 | 0.007 | 0.009 | 0.02 | 0.017 | 0.023 |
| 7 | 0, 1, 10, 0.3, 30 | 9.634 | -0.001 | -0.025 | -0.021 | 2.334 | 1.976 | 0.493 |
| 8 | 0, 1, 10, 0.3, 100 | 8.652 | -0.151 | -0.14 | -0.042 | 0.782 | 0.675 | 0.139 |
| 9 | 0, 1, 10, 0.5, 30 | 8.98 | 0.036 | 0.052 | 0.011 | 1.667 | 1.059 | 0.177 |
| 10 | 0, 1, 10, 0.5, 100 | 5.875 | -0.015 | -0.014 | -0.01 | 0.518 | 0.353 | 0.044 |
| 11 | 0, 1, 10, 0.7, 30 | 5.015 | -0.004 | 0.005 | 0.009 | 0.999 | 0.338 | 0.099 |
| 12 | 0, 1, 10, 0.7, 100 | 4.555 | 0.004 | -0.004 | 0 | 0.324 | 0.088 | 0.029 |
| 13 | 0, 5, 2, 0.3, 30 | 8.019 | 0.051 | 0.089 | 0.075 | 2.423 | 2.401 | 2.76 |
| 14 | 0, 5, 2, 0.3, 100 | 10.39 | 0.021 | 0.019 | -0.014 | 0.763 | 0.73 | 0.911 |
| 15 | 0, 5, 2, 0.5, 30 | 9.127 | 0.044 | 0.032 | 0.035 | 2.133 | 1.854 | 2.397 |
| 16 | 0, 5, 2, 0.5, 100 | 7.482 | 0.072 | 0.063 | 0.071 | 0.681 | 0.576 | 0.705 |
| 17 | 0, 5, 2, 0.7, 30 | 4.926 | 0.025 | 0.018 | 0.005 | 1.583 | 1.428 | 1.705 |
| 18 | 0, 5, 2, 0.7, 100 | 6.827 | -0.02 | -0.019 | -0.022 | 0.494 | 0.431 | 0.554 |
| 19 | 0, 5, 10, 0.3, 30 | 47.852 | 0.492 | 0.342 | 0.148 | 60.699 | 55.42 | 16.759 |
| 20 | 0, 5, 10, 0.3, 100 | 38.465 | -0.122 | -0.036 | 0.074 | 16.152 | 14.149 | 3.046 |
| 21 | 0, 5, 10, 0.5, 30 | 29.658 | -0.015 | -0.172 | -0.199 | 41.32 | 27.412 | 4.819 |
| 22 | 0, 5, 10, 0.5, 100 | 29.749 | -0.05 | -0.095 | -0.068 | 12.189 | 8.069 | 1.267 |
| 23 | 0, 5, 10, 0.7, 30 | 21.37 | 0.155 | 0.113 | 0.039 | 24.076 | 8.487 | 2.448 |
| 24 | 0, 5, 10, 0.7, 100 | 26.222 | 0.25 | 0.071 | 0.027 | 8.34 | 2.318 | 0.721 |
| 25 | 5, 1, 2, 0.3, 30 | 1.825 | 0.023 | 0.025 | 0.019 | 0.11 | 0.101 | 0.123 |
| 26 | 5, 1, 2, 0.3, 100 | 1.684 | -0.002 | -0.002 | -0.003 | 0.028 | 0.028 | 0.036 |
| 27 | 5, 1, 2, 0.5, 30 | 1.648 | 0.003 | 0.005 | 0.002 | 0.08 | 0.071 | 0.078 |
| 28 | 5, 1, 2, 0.5, 100 | 1.628 | 0 | 0.003 | 0.005 | 0.023 | 0.021 | 0.026 |
| 29 | 5, 1, 2, 0.7, 30 | 1.741 | -0.001 | 0 | 0 | 0.074 | 0.065 | 0.079 |
| 30 | 5, 1, 2, 0.7, 100 | 1.277 | -0.012 | -0.012 | -0.016 | 0.018 | 0.015 | 0.02 |
| 31 | 5, 1, 10, 0.3, 30 | 8.508 | -0.04 | -0.034 | 0.044 | 2.433 | 2.041 | 0.628 |
| 32 | 5, 1, 10, 0.3, 100 | 8.718 | -0.033 | -0.021 | 0.002 | 0.686 | 0.582 | 0.117 |
| 33 | 5, 1, 10, 0.5, 30 | 6.598 | 0.052 | 0.06 | 0.049 | 1.654 | 1.169 | 0.22 |
| 34 | 5, 1, 10, 0.5, 100 | 5.787 | 0.001 | -0.017 | -0.018 | 0.505 | 0.327 | 0.052 |
| 35 | 5, 1, 10, 0.7, 30 | 6.659 | -0.065 | -0.03 | 0.015 | 1.047 | 0.384 | 0.096 |
| 36 | 5, 1, 10, 0.7, 100 | 6.417 | -0.023 | -0.014 | -0.014 | 0.279 | 0.085 | 0.029 |
| 37 | 5, 5, 2, 0.3, 30 | 9.414 | 0.123 | 0.098 | 0.072 | 2.578 | 2.504 | 2.825 |
| 38 | 5, 5, 2, 0.3, 100 | 7.97 | 0.027 | 0.029 | 0.043 | 0.773 | 0.76 | 0.948 |
| 39 | 5, 5, 2, 0.5, 30 | 10.043 | -0.108 | -0.114 | -0.094 | 2.067 | 1.775 | 2.173 |
| 40 | 5, 5, 2, 0.5, 100 | 8.033 | -0.042 | -0.041 | -0.075 | 0.598 | 0.539 | 0.684 |
| 41 | 5, 5, 2, 0.7, 30 | 10.339 | -0.03 | -0.051 | -0.057 | 1.509 | 1.358 | 1.834 |
| 42 | 5, 5, 2, 0.7, 100 | 7.161 | 0.028 | 0.007 | -0.012 | 0.444 | 0.396 | 0.551 |
| 43 | 5, 5, 10, 0.3, 30 | 32.024 | 0.103 | 0.109 | 0.207 | 55.015 | 47.913 | 12.43 |
| 44 | 5, 5, 10, 0.3, 100 | 40.876 | -0.063 | -0.173 | -0.1 | 19.182 | 16.698 | 3.268 |
| 45 | 5, 5, 10, 0.5, 30 | 38.9 | 0.022 | 0.079 | 0.036 | 42.372 | 27.524 | 4.433 |
| 46 | 5, 5, 10, 0.5, 100 | 35.735 | 0.104 | 0.105 | 0.02 | 11.306 | 7.202 | 1.205 |
| 47 | 5, 5, 10, 0.7, 30 | 42.135 | -0.473 | -0.268 | -0.153 | 29.072 | 11.455 | 2.846 |
| 48 | 5, 5, 10, 0.7, 100 | 19.896 | -0.134 | -0.086 | -0.023 | 7.804 | 2.31 | 0.796 |

R code

##--------------------------------------------------------------------------

#Functions

##--------------------------------------------------------------------------

#Performs the mixture

mixture<-function(mu,sigma,k,n,p){

 sig<-sample(x=c(sigma,k\*sigma),size=n,replace=TRUE,prob=c(p,1-p))

 mix<-sort(rnorm(n,mu,sig))

 return(mix)

}

#Calculates sample mean, trimmed mean, and median

theta.hat<-function(mu,sigma,k,p,n){

 r.s<-mixture(mu,sigma,k,n,p)

 s.mean<-mean(r.s)

 s.med<-median(r.s)

 trim<-.10\*n

 t.mean<-mean(r.s[(trim+1):(n-trim)])

 return(c(s.mean,t.mean,s.med))

}

#Reports the bias and the MSE.

sta<-function(mu,sigma,k,p,n,m){

 data<-t(replicate(m,expr=theta.hat(mu,sigma,k,p,n)))

 bias<-colMeans(data)-mu

 MSE<-apply(data,2,var)+bias^2

 return(c(bias,MSE))

}

##--------------------------------------------------------------------------

#Parameters and Experimentation

##--------------------------------------------------------------------------

true.mean<-c(0,5)

true.std<-c(1,5)

var.const<-c(2,10)

prop<-c(0.3,0.5,0.7)

sample.size<-c(30,100)

m<-500

x<-numeric(0)

st.dev<-numeric(0)

count=0

for (mu in true.mean){

 for (sigma in true.std){

 for(k in var.const){

 for(p in prop){

 for(n in sample.size){

 x<-c(x,sta(mu,sigma,k,p,n,m))

 st.dev<-c(st.dev,sd(mixture(mu,sigma,k,n,p)))

 count=count+1

 }

 }

 }

 }

}

##--------------------------------------------------------------------------

#Table Creation

##--------------------------------------------------------------------------

#Table of all possible scenarios

tab<-matrix(x,nrow=count,ncol=6,byrow=T)

colnames(tab)<-c("s.mean bias","t.mean bias","median bias","s.mean MSE","t.mean MSE","median MSE")

tab<-cbind(st.dev,tab)

tab<-round(x=tab,digits=3)

#output table into excel file for reporting

write.csv(x=tab,file="midpro.csv")

##---------------------------------------------------------------------------

#Visualizations

##---------------------------------------------------------------------------

par(mfrow=c(1,3))

par(ask=T)

vis.l<-replicate(m,expr=theta.hat(mu=0,sigma=1,k=2,p=0.7,n=100))

vis.h<-replicate(m,expr=theta.hat(mu=0,sigma=1,k=10,p=0.7,n=100))

vis.c1<-replicate(m,expr=theta.hat(mu=0,sigma=1,k=2,p=0.3,n=100))

vis.c2<-replicate(m,expr=theta.hat(mu=0,sigma=1,k=10,p=0.3,n=100))

title.l<-c("k=2, Low Contamination: Sample Mean","k=2, Low Contamination: Trimmed Mean","k=2, Low Contamination: Median")

title.h<-c("k=10, Low Contamination: Sample Mean","k=10, Low Contamination: Trimmed Mean","k=10, Low Contamination: Median")

title.c1<-c("k=2, High Contamination: Sample Mean","k=2, High Contamination: Trimmed Mean","k=2, High Contamination: Median")

title.c2<-c("k=10, High Contamination: Sample Mean","k=10, High Contamination: Trimmed Mean","k=10, High Contamination: Median")

#X axis label

lab<-c("X bar", "X hat 0.1", "X hat 0.5")

#Lable each sub figure

fig1<-c("1a","1b","1c")

fig2<-c("2a","2b","2c")

fig3<-c("3a","3b","3c")

fig4<-c("4a","4b","4c")

#Plotting

for (q in 1:3){

hist(vis.l[q,],prob=T,xlim=c(-.8,.8),ylim=c(0,3.4),xlab=lab[q],main=title.l[q])

lines(density(vis.l[q,]))

text(-.5,2.2,fig1[q])

}

for (w in 1:3){

hist(vis.h[w,],prob=T,xlim=c(-2,2),ylim=c(0,2.5),xlab=lab[w],main=title.h[w])

lines(density(vis.h[w,]))

text(-1.5,2.2,fig2[w])

}

for (a in 1:3){

hist(vis.c1[a,],prob=T,xlim=c(-.8,.8),ylim=c(0,2.5),xlab=lab[a],main=title.c1[a])

lines(density(vis.c1[a,]))

text(-.5,2.2,fig3[a])

}

for (b in 1:3){

hist(vis.c2[b,],prob=T,xlim=c(-2,2),ylim=c(0,1.5),xlab=lab[b],main=title.c2[b])

lines(density(vis.c2[b,]))

text(-1.5,1.2,fig4[b])

}

par(mfrow=c(1,1))

par(ask=F)