Bootstrap Lab #2

1 Stats Lab # 2: Robust Tests of Central Location

1.1 Initialize R

Enter the following commands in R:

```
> source(url("http://www-rcf.usc.edu/~rwilcox/Rallfun-v9"))
> load(url("http://psycserv.mcmaster.ca/bennett/rdata/sgData.Rdata"))
```

The first line loads Wilcox's functions for doing robust analyses. The second loads a data file.

1.2 Mean

Use the functions `mean`, `sd`, and `t.test` to calculate the mean, standard deviation, and 95% confidence interval for the data in `sgData`. Use `boxplot` to inspect the data visually.

```
> (m <- mean(sgData) )

[1] 753.6653

> (s <- sd(sgData) )

[1] 3334.578

> n <- length(sgData)
> alpha <- .05
> t.low <- qt(alpha/2,df=n-1)
> t.high <- qt(1-alpha/2,df=n-1)
> confinterval <- c(m-t.high*s/sqrt(n),m-t.low*s/sqrt(n))
> names(confinterval)<- c("2.5%","97.5%")
> confinterval

     2.5%     97.5%
-806.9654 2314.2960

> # faster way:
> t.test(sgData)

        One Sample t-test

data:  sgData
t = 1.0108, df = 19, p-value = 0.3248
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -806.9654 2314.2960
sample estimates:
mean of x
 753.6653
```

A boxplot on the original scores doesn't show very much because of the presence of two extreme scores. Therefore, I am going to plot the the log-transformed scores instead (see Figure 1). You can see that there are two scores that are a couple of log units above the median score (which is indicated by the horizontal line in the box).

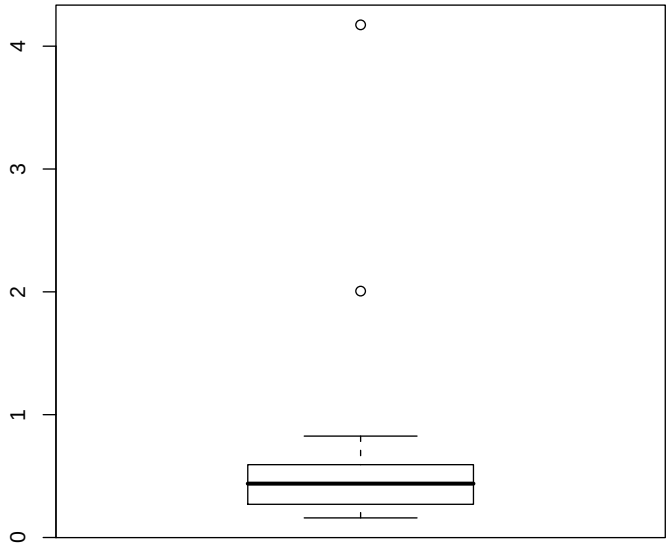

Figure 1: Boxplot of log-transformed scores in `sgData`.

- Do the mean and confidence interval reflect what you think represents a good measure of a typical score? Explain.

  No, the mean is not a good measure of a "typical" score. The median of `sgData` is 2.75; all of the scores except two are less than seven. Hence, the mean, which is 753, is not representative of most scores. Also, the confidence interval $[-806.9, 2314.3]$ seems inconsistent with the fact that 90% of the observations (18 out of 20) are between 1.4 and 6.7.

- Let's define an "outlier" as any score that is more than 3 standard deviations from the mean. Now, identify the outliers in `sgData`. Has this procedure worked satisfactorily? Why or why not?

```
> m <- mean(sgData)
> s <- sd(sgData)
> z <- (sgData - m)/s # calculate z scores
> sgData[abs(z)>3] # list all outliers

[1] 14920.41

> range(sgData[abs(z)<=3]) # range of all non-outliers

[1]   1.444219 101.425927
```

No, the procedure did not work well because it identified only the most extreme score as an outlier: the remaining values still contained one obvious outlier (Figure 2). The problem – referred to as outlier masking – is that the outliers in the original sample inflated the mean and standard deviation to such a degree that the odd scores no longer fit the definition of an unusual or outlying score (see).

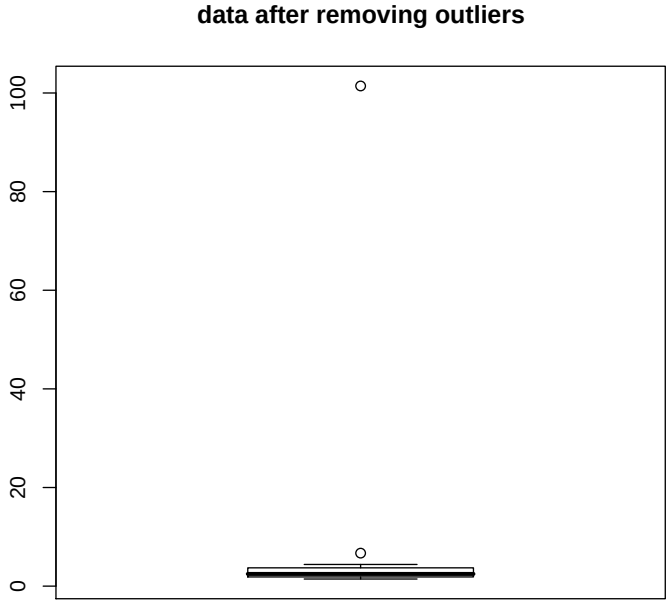

Figure 2: Boxplot of scores in `sgData` after removing outliers.

1.3 Trimmed Mean

Use the commands `mean` and `tmean` to calculate the 10% and 20% trimmed means. Next, use `trimci` and `trimpb` to calculate 95% confidence intervals for the 10% and 20% trimmed means.

Here are the trimmed means:

```
> mean(sgData,trim=0.1)

[1] 3.030476

> mean(sgData,trim=0.2)

[1] 2.816027

> tmean(sgData,tr=0.1)

[1] 3.030476

> tmean(sgData,tr=0.2)

[1] 2.816027
```

Here are the confidence intervals calculated with `trimci`:

```
> trimci(sgData,tr=0.1)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1] 2.020424 4.040529

$test.stat
[1] 6.395023

$p.value
[1] 1.204647e-05

> trimci(sgData,tr=0.2)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1] 2.063349 3.568706

$test.stat
[1] 8.234635

$p.value
[1] 4.957173e-06
```

Here are the confidence intervals calculated with `trimpb`:

```
> trimpb(sgData,tr=0.1,nboot=2000)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "Taking bootstrap samples. Please wait."
$ci
[1]   2.332242 942.280313

$p.value
[1] 0

> trimpb(sgData,tr=0.2,nboot=2000)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "Taking bootstrap samples. Please wait."
$ci
[1]  2.195048 11.478736

$p.value
[1] 0
```

- How do the confidence intervals for the trimmed means compare to the confidence interval for the mean? Which do you think is a more accurate estimate of the correct confidence interval? Why?

The confidence interval for the 10% trimmed mean is smaller than the one for the mean, but the confidence interval for the 20% trimmed mean is *much* smaller than the confidence interval for the mean. Technically, all of the intervals are correct, but I think that the interval for the 20% trimmed mean gives a more accurate indication of the reliability, or consistency, of the bulk of the measurements.

- Do the confidence intervals for the 10% and 20% trimmed means differ? Does this make sense?

  Yes, and yes. Trimming by 10% removes the two highest and lowest scores from the original sample of 20. Therefore, if samples of 20 scores contain two or fewer outliers at each tail, then the 10% trimmed mean will be stable across samples. However, if samples sometimes contain more than two outliers at either tail, then the 10% trimmed mean can vary significantly across samples. In our case, the original sample suggests that extreme high scores constitute approximately 10% of the population of scores, and therefore we should occasionally get samples of 20 that contain 3 extreme high scores. Hence, the 10% trimmed mean will vary significantly across samples. However, the 20% trimmed mean still should be stable because samples rarely will contain more than 4 extreme high scores.

- Do the intervals calculated by `trimci` and `trimpb` differ? What might this result imply about the assumptions that underlie `trimci`?

  Yes, they do differ: the intervals calculated with `trimci` are narrower than the ones calculated with `trimpb`. I suspect this difference is due to the fact that the assumption that the sample trimmed mean follows a t distribution – an assumption made by `trimci` but not `trimpb` – is not valid in this case.

1.4 M-estimators

- Use the commands `onestep` and `mom` to calculate the one-step M-estimator and the modified one-step M-estimator for `sgData`. How do these values compare the mean and trimmed means?

```
> onestep(sgData)

[1] 2.950505

> mom(sgData)

[1] 2.634062
```

  The M-estimators are much less than the means and similar to the trimmed means.

- Use `momci` and `onesampb` to calculate 95% confidence interval for $\hat{\mu}_{os}$ and $\hat{\mu}_{mom}$. How do these intervals compare to the intervals for the mean and trimmed means?

```
> momci(sgData,nboot=2000)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] 1.870453 3.511839

> onesampb(sgData,est=onestep,nboot=2000)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] 2.126626 4.061419
```

  The widths of the confidence intervals are approximately 2, and therefore both are significantly narrower than any of the confidence intervals calculated previously.

- Explain how the MOM is computed for the `sgData`.

  The modified one-step M-estimator, or MOM, is computed in two steps. First, outliers are identified and discarded from the data. An outlier is defined as any score, X_i that satisfies the condition

  $$\frac{|X_i - M|}{MADN} > K$$

  where M is the median of the entire sample, $MADN$ is (1/0.6745) times the median absolute deviation of scores, and K is a constant, sometimes referred to as *bend* parameter. Typically, $K = 2.24$. Second, the mean of the remaining values are calculated.

- Are the data in `sgData` skewed? Might the presence or absence of skew influence your choice of using a trimmed mean or an M-estimator? Explain.

  Yes, the data are skewed positively: the outliers are all greater than the median. One advantage of the M-estimator over the trimmed mean is that it does not necessarily discard scores on both sides of the median. In other words, M-estimators can reduce the influence of only very large scores, or very small scores. Therefore an M-estimator might be preferred when dealing with skewed data.