Introduction to the Bootstrap and Robust Statistics

PSY711/712

Winter Term 2009

1 Confidence Intervals of a Trimmed Mean

1.1 Definition of Trimmed Mean and Standard Error of Trimmed Mean

Consider a set of observations $[x_1, x_2, x_3, \ldots, x_n]$ that are ordered from smallest to largest. The $k\%$ **trimmed mean**, $\overline{X}_t$ is the average of the values that remain after removing the bottom $k\%$ (i.e., smallest) *and* the top $k\%$ (i.e., the largest) observations from the original set. The variance of the trimmed mean is

$$\frac{s_w^2}{n \cdot (1 - 2\gamma)^2} \tag{1}$$

where s_w^2 is the winsorized variance (see below), n is the sample size (before trimming), and γ is the proportion of trimming. The standard error of the trimmed mean is

$$\frac{s_w}{\sqrt{n} \cdot (1 - 2\gamma)} \tag{2}$$

What is the best amount of trimming? This depends on your particular circumstances, but Wilcox (2005) recommends 20%. Obviously, smaller sample sizes might compel you to use less trimming (e.g., 10%).

1.2 Definition of Winsorized Mean and Variance

Another robust estimtor of central tendency is the winsorized mean. Like the trimmed mean, the winsorized mean eliminates the outliers at both ends of an ordered set of observations. Unlike the trimmed mean, the **winsorized mean** *replaces* the outliers with observed values, rather than discarding them. The k-th winsorized mean is the average of the observations after each of the first k smallest values are replaced by the *(k+1)*-th smallest value, and the k largest values are replaced by the *(k + 1)*-th largest value. So, supposed our sorted observations are <11, 13, 21, 22, 22, 24, 25, 25, 42, 55>, the 20% winsorized mean is the average of the winsorized values <21, 21, 21, 22, 22, 24, 25, 25, 25, 25>.

The **winsorized variance**, s_w^2, of a sample is simply the variance of the winsorized set of values, W,

$$s_w^2 = \frac{\Sigma_{i=1}^{n} \left(W_i - \overline{W}\right)^2}{n - 1} \tag{3}$$

where n is the sample size.

1.2.1 R examples

R's built-in function, `mean`, can be used to calculate the trimmed mean of a set of numbers;

```
> set.seed(55)
> x <- rchisq(100, df = 3)
> mean(x)
```

```
[1] 3.662562

> mean(x, trim = 0.2)

[1] 3.015199

> mean(x, trim = 0.1)

[1] 3.164719
```

Rand Wilcox, a Professor of Psychology at the University of Southern California, has created a large set of functions that can be used with R to perform a wide range of robust statistical analyses. The name of the current file containing the routines is `Rallfun-v9`, and it can be loaded from his website into R with the `source` command:

```
> source(url("http://www-rcf.usc.edu/~rwilcox/Rallfun-v9"))
```

This command loads *many* functions into R's workspace; you can list them with the `ls()` command. One of the commands, `tmean`, can be used to calculate trimmed means:

```
> tmean(x, tr = 0.1)

[1] 3.164719
```

And the function `trimse` can be used to estimate the standard error of the trimmed mean:

```
> trimse(x, tr = 0.1)

[1] 0.2774806
```

Note to Matlab users: a matlab function, `trimmean`, computes the trimmed mean, but it uses a definition of a trimmed mean that differs slightly from the one used here. Specifically, `trimmean(x,p)` returns the mean of the array `x` after trimming the top (`p`/2)% and bottom (`p`/2)%. Note that our definition requires that `p`% of the values be trimmed off of the top *and* bottom.

The Winsorized mean and variance can be calculated with, respectively, `win()` and `winvar()`:

```
> win(x, tr = 0.2)

[1] 3.135361

> winvar(x, tr = 0.2)

[1] 2.565369
```

1.3 t approximation

You should be familiar with the equation that defines Student's t. The analog, when working with trimmed means is

$$T_t = \frac{(1-2\gamma)\sqrt{n}(\bar{X}_t - \mu_t)}{s_w} \tag{4}$$

where γ is the proportion of trimmed scores. T_t has $n_t - 1$ degrees of freedom, where n_t is the number of scores that remain *after* trimming. Assuming that T_t follows the t distribution (df=$n_t - 1$), then the $100(1-\alpha)\%$ confidence interval for the trimmed mean is

$$\bar{X}_t \pm t_{\alpha/2}\frac{s_w}{1-2\gamma\sqrt{n_t}} \tag{5}$$

where $t_{\alpha/2}$ is the value of t that is exceed with a probability equal to $\alpha/2$.

In R, confidence intervals for a trimmed mean based on the t distribution are calculated with Wilcox's function `trimci()`:

```
> trimci(x, tr = 0.2, alpha = 0.05)

[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] 2.481041 3.549357

$test.stat
[1] 11.29516

$p.value
[1] 2.220446e-16

> trimci(x, tr = 0.1, alpha = 0.05)

[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] 2.612408 3.717031

$test.stat
[1] 11.40519

$p.value
[1] 0
```

1.4 bootstrapped confidence intervals

If T_t does not follow the t distribution, then the confidence interval calculated using Equation 5 may be misleading. Alternative methods of calculating confidence intervals that do not rely on the t assumption use, instead, the percentile and percentile-t bootstrap procedures.

```
> trimpb(x, tr = 0.2, alpha = 0.05, 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] 2.493637 3.544321

$p.value
[1] 0

> trimcibt(x, tr = 0.2, alpha = 0.05, nboot = 2000)

[1] "Taking bootstrap samples. Please wait."
[1] "NOTE: p.value is computed only when side=T"
$ci
[1] 2.515170 3.611451

$test.stat
[1] 11.29516

$p.value
[1] NA
```

2 Confidence interval of a one-step M-estimator

2.1 Definition of the one-step Huber M-estimator

A trimmed mean is based on a predetermined amount of trimming. Also, the amount of trimming is symmetrical, in the sense that the same number of low and high values are trimmed. Another approach would be to adjust the amount and kind of trimming based on the the distribution of outliers in your sample. In a sense, this is what is done by M-estimators of central tendency. Essentially, M-estimators are derived by a repetitive process. Let $\overline{X}_m$ represent the M-estimator of central tendency of a sample. First, an initial guess, $\overline{X}_{m0}$, is made of central tendency. For example, the initial guess might be the median. Next, values that differ significantly from $\overline{X}_{m0}$ (i.e., outliers) are identified. Finally, a new value for the M-estimator, $\overline{X}_{m1}$is calculated with a formula that reduces the influence of the values identified as outliers. If $\overline{X}_{m1}$ and $\overline{X}_{m0}$differ significantly, then the process is repeated: i.e., values that differ a lot from $\overline{X}_{m1}$ are identified as outliers, and a new estimate $\overline{X}_{m2}$ is calculated. The cycle continues until there is no significant difference between $\overline{X}_{m(k)}$ and $\overline{X}_{m(k+1)}$. You can probably tell from this description that M-estimator refers to a large class of measures that might differ in how they define outliers, how outliers and non-outliers are combined, and the rule for stopping the iteration process. The M-estimator that we will use is called a one-step M-estimator because the iteration process ends after the very first step (i.e., after $\overline{X}_{m1}$ is calculated). It is sometimes referred to as a Huber M-estimator on the pioneering work of Peter Huber [1].

To calculate our one-step Huber M-estimator, we need to define an outlier. First, let's come up with a measure of variation in our sample. The median absolute deviation, MAD, is a robust measure of the variability in a sample. It is calculated by subtracting the median from each element in a sample, taking the absolute value of each difference, and then finding the median of the absolute values MADN is simply a scaled version of MAD:

$$MADN = \frac{MAD}{0.6745} \tag{6}$$

If the data are drawn from a normal distribution, then MADN is approximately equal to the population standard deviation. We define an outlier to be any value, x, where the following statement is true: abs(x - median) / MADN $>$ 1.28,

$$\frac{|x - M|}{MADN} > K \tag{7}$$

where $||$ is absolute value, and M is the median of the original sample, and K is a constant. A common choice for K is 1.28, and that is the value that we will use. Let L be the number of outliers in our sample that are *less than* M, U be the number of outliers that are *greater than* M, and B be the sum of all of the remaining (i.e., non-outlier) values. The one-step Huber M-estimator with $K = 1.28$ is

$$\hat{\mu}_{os} = \frac{1.28 \cdot MADN \cdot (U - L) + B}{(n - L - U)} \tag{8}$$

where n is the sample size. Since I do not want to continue typing 'one-step-Huber-M-estimator-with-$K = 1.28$' everywhere, I will simply refer to this value as our M-estimator, or $\overline{X}_m$.

2.2 Modified One-step M-estimator (MOM)

The modified one-step estimator (MOM) is calculated by identifying outliers, removing them from the data, and averaging the remaining values. A score, X_i, is declared an outlier if

$$\frac{|X_i - M|}{MADN} > 2.24 \tag{9}$$

where M is the sample median and MADN is given by Equation 6. The value of 2.24 is sometimes referred to as so-called *bend* parameter; 2.24 is a standard value, but others (e.g., 3.5) are used occasionally. The average of the non-outliers is an estimate of the population MOM ($\hat{\mu}_{mom}$).

2.2.1 R example: `onestep`

Wilcox's function `onestep` can be used to calculate the one-step M-estimator given by Equation 8.

```
> onestep(x, bend = 1.28)

[1] 3.179459
```

The MOM is calculated thusly:

```
> mom(x, bend = 2.24)

[1] 2.904912
```

2.3 Standard error of the one-step M-estimator

The standard error of μ_{os} can be estimated using the bootstrap. The original sample, X, is sampled with replacement to generate B bootstrapped samples, X_b^*. Calculate the value of the M-estimator for each sample $\overline{X}^*_{m(b)}$. The estimated standard error of μ_m , represented by the symbol $\hat{\sigma}_m$, is the *standard deviation* of the $\overline{X}^*_{m(b)}$values. Wilcox [3] recommends that the standard error be based on at least 100 bootstrapped samples.

2.4 Confidence Intervals of an M-estimator

Wilcox [3] states that the percentile-t method of constructing confidence intervals for μ_{os} yields unsatisfactory results when $n \leq 40$, and recommends the percentile method be used instead. As before, the original sample, X, is sampled with replacement to generate B bootstrapped samples, X_b^*, and we calculate the value of the M-estimator for each sample $\overline{X}^*_{m(b)}$. The M-estimators are sorted in ascending order, $\left[\overline{X}^*_{m(1)} \leq \overline{X}^*_{m(2)} \leq \overline{X}^*_{m(3)} \leq \cdots \leq X^*_{m(B)}\right]$, and the estimated $(100 \times \alpha)\%$ confidence interval are $\left(\overline{X}^*_{m(lo+1)}, \overline{X}^*_{m(hi)}\right)$, where $lo = ROUND(\alpha B/2)$ and $hi = B - lo$. We reject the null hypothesis, $H_0 : \overline{X}_m = \mu_m$, in favor of the alternative hypothesis, $H_1 : \overline{X}_m \neq \mu_m$, if $\mu_m < \overline{X}^*_{m(lo+1)}$or $\mu_m > \overline{X}^*_{m(hi)}$.

2.4.1 R examples

Confidence intervals for MOMs can be calculated using a command provided by Wilcox:

```
> momci(x, alpha = 0.05, nboot = 2000)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] 2.330498 3.487020
```

The function `onesampb` can be used to calculate the confidence interval for a one-step M-estimator:

```
> onesampb(x, est = onestep, alpha = 0.05, nboot = 2000, bend = 1.28)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] 2.620462 3.673801
```

Note that `onesampb` can be used to calculate the confidence interval (based on the percentile bootstrap) for any estimator. For example, here is how to calculate the confidence interval for the 10% trimmed mean:

```
> onesampb(x, est = mean, alpha = 0.05, nboot = 599, tr = 0.1)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] 2.645527 3.796974
```

3 Two-sample t test

The classical two-sample t test probably is the most common way of assessing the difference between two independent groups. The value of t is given by

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{10}$$

where s_p^2 – the pooled variance estimate – is derived from the sample variances, s_1^2 and s_2^2, according to the equation

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{11}$$

Often the hypothesis being evaluated is that the population means do not differ, in which case Eq 10 simplifies to

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{12}$$

When the scores are drawn randomly and independently from normal distributions, **and when the variances of the two populations are equal**, then the t statistic calculated with Eq 10 (or 12) follows a theoretical t distribution with $n_1 + n_2 - 2$ degrees of freedom. Therefore, we can use the theoretical distribution to calculate the probability of obtaining values of t that are at least as extreme as the observed value. When this probability is small (typically $p < .\alpha$, where $\alpha \leq .05$), then common practice is to reject the null hypothesis ($\mu_1 = \mu_2$) in favor of an alternative hypothesis ($\mu_1 \neq \mu_2$). The probability of making a so-called Type I error – that is, of incorrectly rejecting the null hypothesis – is α.

The following code shows how to do a t test in R that evaluates the hypothesis that the two means are equal:

```
> set.seed(93)
> x <- rnorm(n = 20, mean = 0, sd = 1)
> y <- rnorm(n = 20, mean = 3, sd = 4)
> t.test(x, y, var.equal = T, alpha = 0.05)

        Two Sample t-test

data:  x and y
t = -3.88, df = 38, p-value = 0.0004024
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.373507 -1.374503
sample estimates:
mean of x mean of y
0.4036250 3.2776300
```

Here we evaluate the hypothesis that $\mu_x - \mu_y = -3$:

```
> t.test(x, y, mu = -3, var.equal = T, alpha = 0.05)

        Two Sample t-test

data:  x and y
t = 0.1701, df = 38, p-value = 0.8658
```

```
alternative hypothesis: true difference in means is not equal to -3
95 percent confidence interval:
 -4.373507 -1.374503
sample estimates:
mean of x mean of y
0.4036250 3.2776300
```

3.1 Heteroscedasticity

The t test assumes that the population distributions are normal with *equal* variance. Random variables that have equal variance are said to be homoscedastic. When the variances are *un*equal, variables are hetreroscedastic. It is important to understand that heteroscedasticity causes problems with our t test: specifically, when the equal variance[1] assumption is not valid, then our t statistic will not follow the theoretical t distribution, and therefore probability statements derived from the t distribution will be incorrect. So, for example, the probability of making a Type I error will not equal the nominal value of α.

The classical solution to this problem has been to use the so-called Welch-Satterthwaite equation to adjust the degrees of freedom:

$$df' = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}} \tag{13}$$

When the populations variances are unequal, t is distributed *approximately* as a t variable with df' degrees of freedom.

The following code shows how to do a t test in R that assumes the variances are unequal, and adjusts the degrees of freedom using Equation 13

```
> t.test(x, y, var.equal = F, alpha = 0.05)

        Welch Two Sample t-test

data:  x and y
t = -3.88, df = 23.195, p-value = 0.0007484
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.405582 -1.342428
sample estimates:
mean of x mean of y
0.4036250 3.2776300
```

Notice how the degrees of freedom differ from the previous test. Also note that the default in R is to assume that the variances are unequal. Therefore the same results are obtained when `var.equal` is not specified:

```
> t.test(x, y, alpha = 0.05)

        Welch Two Sample t-test

data:  x and y
t = -3.88, df = 23.195, p-value = 0.0007484
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

[1] I'm not going to keep saying or typing homoscedasticity

```
 -4.405582 -1.342428
sample estimates:
mean of x mean of y
0.4036250 3.2776300
```

3.2 Yuen's robust alternative to the t test

The Welch-Satterthwaite correction yields reasonable results when both samples are drawn from normal distributions. When the distributions are not normal – for example, if they have heavy tails or if they are skewed – then correcting the degrees of freedom may not yield accurate results. This problem can be especially severe when the two samples differ in size. Many of these problems are discussed by Wilcox [2].

Yuen [4] addressed this problem by generalizing the t test to a test of timmed means (rather than means). The basic idea here is that the difference between two trimmed means will be less influenced by deviations from normality, and therefore a t-like statistic based on trimmed means would be more likely to follow the t distribution. We denote the sample γ-trimmed mean of samples one and two as $\bar{X}_{t1}$ and $\bar{X}_{t2}$, where γ equals the proportion of trimming. Yuen's statistic for testing the hypothesis of equal trimmed means is

$$t_y = \frac{\bar{X}_{t1} - \bar{X}_{t2}}{\sqrt{d_1 + d_2}} \tag{14}$$

where

$$d_j = \frac{(n_j - 1)\hat{\sigma^2_{wj}}}{h_j(h_j - 1)} \tag{15}$$

and σ^2_{wj} is the γ-Winsorized variance for group j and h_j is the effective sample size (i.e., the sample size *after* trimming). Yuen's test assumes that t_y follows a t distribution with degrees of freedom equal to

$$\nu_y = \frac{(d_1 + d_2)^2}{d_1^2/(h_1 - 1) + d_2^2/(h_2 - 1)} \tag{16}$$

The following code illustrates how to perform Yuen's t test on trimmed means for two independent groups, using Wilcox's R function `yuen`. A classical t also is performed for comparison:

```
> source(url("http://www-rcf.usc.edu/~rwilcox/Rallfun-v9"))
> set.seed(72)
> x <- rnorm(40, mean = 0, sd = 5)
> y <- c(rnorm(35, mean = 5, sd = 5), rnorm(5, mean = 5, sd = 25))
> t.test(x, y, alpha = 0.05)

        Welch Two Sample t-test

data:  x and y
t = -2.4, df = 53.697, p-value = 0.01989
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.5376122 -0.7652945
sample estimates:
 mean of x  mean of y
-0.1441972  4.5072561

> test.results <- yuen(x, y, tr = 0.2, alpha = 0.05)
> test.results$ci

[1] -9.039814 -3.220781
```

```
> test.results$p.value

[1] 0.0001119766
```

Note how – in this case – the confidence interval for the difference between 20% trimmed means is significantly smaller than the confidence interval for the difference between means. According to Yuen's test, we would conclude that the difference between 20% trimmed means is statistically significant ($p = .00048$).

3.2.1 Bootstrapped versions of Yuen's test

Wilcox [3] suggests that Yuen's robust t test may yield inaccurate results when the normality and equal-variance assumptions are violated and sample sizes are small (e.g., $n < 30$). In such cases, a bootstrap version of Yuen's test may be preferable.

Wilcox's routine `yuenbt` uses the percentile-t bootstrap to estimate a confidence interval for the difference between two trimmed means:

```
> set.seed(2109)
> x <- rnorm(20, mean = 3, sd = 5)
> y <- rchisq(20, df = 3, ncp = 4)
> yuenbt(x, y, tr = 0.2, alpha = 0.05, nboot = 1999)

[1] "NOTE: p-value computed only when side=T"
[1] "Taking bootstrap samples. Please wait."
$ci
[1] -7.528124 -2.008034

$test.stat
[1] -3.549337

$p.value
[1] NA
```

`yuenbt` returns a p-value only when the `side` parameter is TRUE:

```
> yuenbt(x, y, tr = 0.2, alpha = 0.05, nboot = 1999, side = T)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] -7.496828 -2.003231

$test.stat
[1] -3.549337

$p.value
[1] 0.003001501
```

Setting `side` to TRUE forces `yuenbt` to calculate a so-called **symmetric confidence interval**. How is such a thing calculated? Imagine that we calculated R bootstrapped values of t. The typical, non-symmetrical confidence interval is calculated by extracting the t values corresponding to the $\alpha/2$ and $1 - \alpha/2$ quantiles. Now imagine that we take the absolute value of all bootstrapped t's, and select T_c at the $1-\alpha$ quantile. This single value, T_c, is used to construct symmetrical confidence intervals. Wilcox [3] (page 120) argues that symmetric confidence intervals tend to produce more accurate results when using the percentile-t bootstrap.

Wilcox also has written an R function that calculates confidence interval using a percentile bootstrap. The function returns the estimated population difference between trimmed means (`est.dif`), a confidence interval (`ci`) and a p-value for the null hypothesis of no difference (`p.value`):

```
> trimpb2(x, y, tr = 0.2, alpha = 0.05, nboot = 1999)

$p.value
[1] 0.002001001

$ci
[1] -7.434011 -1.548752

$est.dif
[1] -4.750029
```

3.3 Comparing M-estimators between two independent groups

Wilcox recommends that the percentile bootstrap method be used when comparing M-estimators across groups. The following R example shows how to use the routine `pb2gen` to evaluate the hypothesis of no group difference in Modified One-Step M-estimators (MOM). The function returns a confidence interval, a p-value for the null hypothesis test, and the squared standard error of the difference between MOMs:

```
> pb2gen(x, y, alpha = 0.05, nboot = 1999, est = mom)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] -7.485629 -1.275413

$p.value
[1] 0.004002001

$sq.se
[1] 2.527066
```

3.4 Permutation (Randomization) tests

An alternative to the bootstrap is the so-called permutation, or randomization, test. The logic behind the permutation test is simple. In most situations, the null hypothesis is that means, or trimmed means, or MOMs, do not differ between groups. Let's alter this hypothesis slightly and say that the two groups were drawn from the same population. If this hypothesis is true, then the observed difference between groups is due to pure chance (i.e., the random assignment of individuals to two groups). We can examine these random effects on our statistic – be it the difference between group mean, trimmed means, or whatever – by randomly reassigning our scores to the two groups. By repeating this randomization procedure many times, we can see how the statistic varies when the null hypothesis is true. If the observed statistic – i.e., the one calculated for our data – is unusually large or small, then we might conclude that the null hypothesis is false.

Note that the randomization procedure differs from the bootstrap because it relies on sampling *without* replacement. Also note that the null hypothesis differs slightly from the ones evaluated with the bootstrap. Specifically, the randomization test evaluates the null hypothesis that the two distributions are identical (which is equivalent to saying that the scores came from one distribution).

Wilcox provides the routine `permg` for conducting a permutation/randomization test on two independent groups:

```
> permg(x, y, alpha = 0.05, est = mom, nboot = 1999)

$dif
[1] -4.157222
```

```
$lower
[1] -3.081557

$upper
[1] 3.163626

$reject
[1] "yes"
```

References

[1] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[2] R. R. Wilcox and H. J. Keselman. Modern robust data analysis methods: measures of central tendency. *Psychol Methods*, 8(3):254–74, 2003.

[3] Rand R. Wilcox. *Introduction to robust estimation and hypothesis testing*. Elsevier Academic Press, 2005.

[4] K.K. Yuen. The two-sample trimmed t for unequal population variances. *Biometrika*, 61:165–70, 1974.