Comparing Estimators

The Median

For the sake of discussion, assume that we are measuring the heights of randomly selected adult men from the U.S. Also for the sake of discussion, let's suppose that this population follows a N(70", 3") distribution, with height measured in inches.

Suppose we plan to take a random sample of size 10. (Obtaining such a sample isn't easy, but assume we can do it.) Here's one such simulated sample

```
> (def x (+ (* 3 (normal-rand 10)) 70))
X
> x
(70.4137146579604 66.64387360086086 65.02215265160767 68.59955985316095
67.50177488231989 72.28927908590187 72.43601901835297 72.1645650707586
66.94914404152433 69.76333582979076)
>(median x)
```

The median of this list is 69.18 (rounding off), which is pretty close. The error is -0.82. But one wonders whether we were this close just because of luck, or whether this is typical.

So take another sample:

```
(68.33798052106891 70.54899024675639 73.90930552039957 71.20938752129774
69.13832575867107 70.79102487857833 68.91376948312231 69.9605805576679
70.79784104468536 71.2276041211891)
> (median x)
70.67000756266737
```

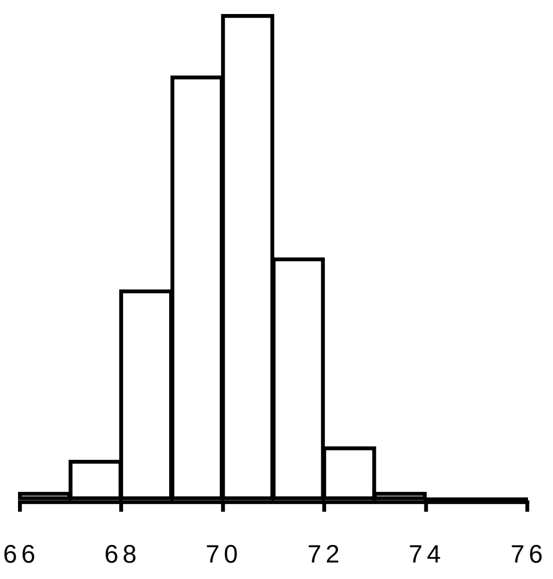
The error is +0.67. Even closer.

To understand how this would do in the long run, we simulate 1000 repetitions:

```
> (def manymedians (sample-distr #'median 70 1000 10 3))
```

(This is a home-brewed function that takes as input an estimator-function (median, preceded by #' to make sure the LISP evaluates it as a function and not a value), the mean (70), the number of simulations (1000), the size of each sample (10), and the standard deviation (3).)

Lets treat this list, manymedians, as we would any data set, and look at a picture:



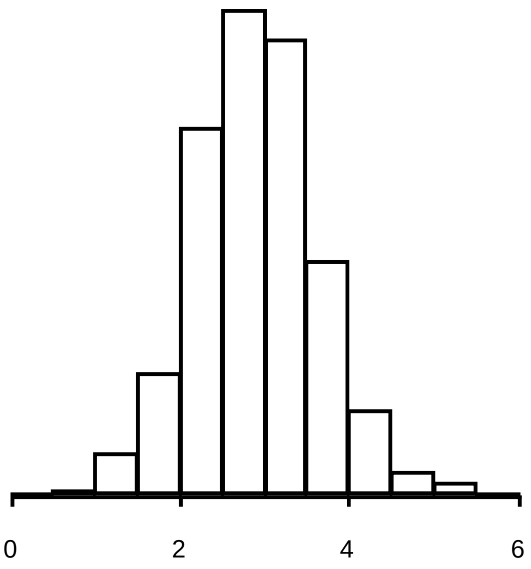
The sampling distribution appears rather symmetric, and indeed appears very close to normal. The median of the medians is 70.11; the average of the medians is 70.09, (which strongly suggests that the median of the sample is an unbiased estimator of this population mean -- at least for this normal population), and the SE is approximately 1.136, which is smaller than the SD of the population (3). Compare this to the theoretical SE for the average, sigma/sqrt(n) = 3/sqrt(10) = 0.9486, and you see that they are roughly in the same ballpark.

In practice, of course, we don't know the mean and standard deviation and median before we begin. But this simulation tells us that if we take the sample median of a sample of 10 people, we can be fairly certain our median will be close (within 1.136, approximately, about 68% of the time, using the "empirical rule".)

Standard-Deviation

Suppose we wish to estimate the standard deviation of this population. There are two popular estimators. The so-called "sample standard deviation" divides by (n-1); the "population standard deviation", which has the distinction of also being the maximum likelihood estimator, divides by n. Let's call the first s1, the second s2. Imagine we take a sample of size 10, again, and compute both estimators.

Here's the sampling distribution of s1:



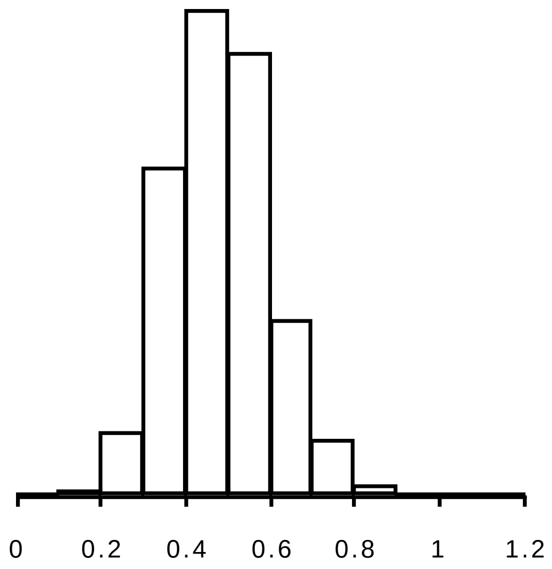
It looks normal, too. But in fact it is not. For one thing, negative values are impossible. The result is a distribution skewed to the right (although only slightly in this case). Note that some of our samples had quite small standard deviations, and some quite large. But these were rare. The average of this list is 2.92, very close to the "target" value of 3.

Now for s2:

The histogram looks pretty much the same. (We did an entirely new simulation, so the histograms are not exactly the same.) The average of s2 was 2.77. A little less close to the target.

As you can see from some algebra, s1 = (sqrt (n/ n-1))s2, which shows that s1 is always slightly larger than s2. This is because s2 is biased downwards: it consistently gives values that are smaller than the target. The "correction" factor fixes this.

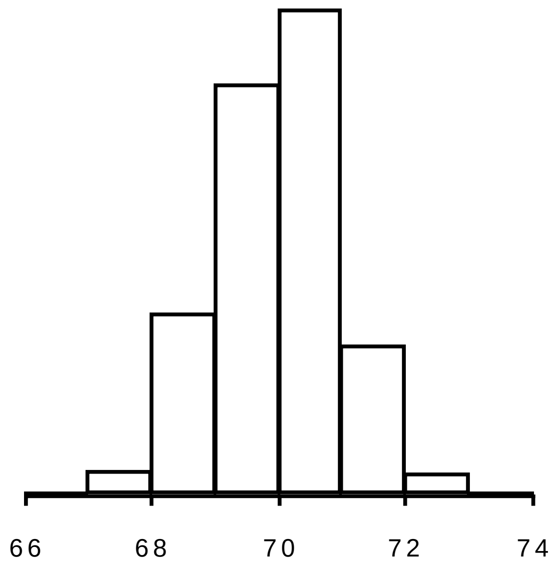
Incidentally, you can get a better sense of the skewed sampling distribution if we instead sample from a N(70,.5) distribution. Now here's a histogram of 1000 such simulations:



This should look a little more skewed to you.

<u>Sample Mean</u>

In the case of the sample mean, we have well-known theory to guide us. We know that the sample mean (the average of the sample) will be N(70, .9486). Let's check:



The average of these averages is 70.0018, and the SD is 0.94568. Both quite close.

The situation in reverse

Suppose we know only that we have sampled from a N(?, 3) population. We see these numbers:

(64.09791552690261 67.03127774411125 71.23499658281865 74.64023086443113 72.74757233302203 74.18724752373062 69.96287696287911 66.52040596863715 64.2658942174255 72.9117640212181)

What's the true value of the mean?

Of course there's no way of knowing. The average of this list is 69.76. But our theory tells us that sample averages are almost always (at least 95% of the time) within 2 SEs of the true mean. So the true mean must be somewhere between 69.76 - 2*.9486 and 69.76 + 2*.9486 or (67.86, 71.66).

Of course there's no guarantee. We only know that 95% of all samples will produce averages that are within 2 SEs of the true mean, and so we can be reasonably confident that ours is one of them. In fact, the true mean here was 68, so we win.

Appendix:
Here's the code for sample-distr:

```
(defun sample-distr (estimator center M n sigma)
(let ( resultlist '())
(dolist (i (iseq 1 M) resultlist)
          (def x (normal-rand n))
          (def newx (+ (* x sigma) center))
          (def resultlist (append (list (funcall estimator newx)) resultlist)))))
#|example (sample-distr #'mean 10 1000 100 4)
or (sample-distr #'median 10 1000 100 4)|#
```