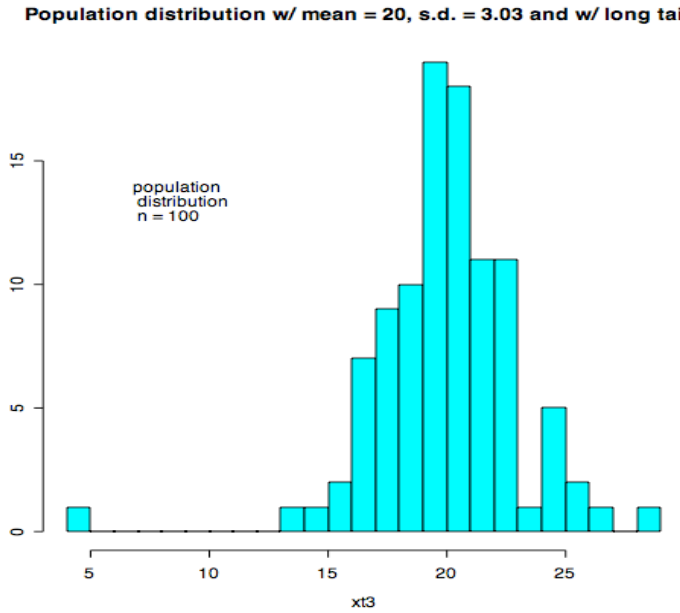
Bootstrapping illustrated
Comparing the standard mean, 10 and 20% trimmed means, & median

Before discussing the main topic, let us quickly review sampling distributions so that everyone is clear on the major theoretical background.

Because the concept of a sampling distribution of a statistic (especially a mean) is so fundamental to bootstrapping – what it's about, why it works as it does – I want to review the following: The sampling distribution of the mean has ***three principal characteristics*** *you should remember:* (1) For any sample size *n*, the mean of *all* (!) sample means [drawn from a finite population] is *necessarily* equal to the population mean (such a statistic is said to be *unbiased*); (2) The *variance* of the distribution of all means (always) equals the population variance divided by *n*; and (perhaps surprisingly), (3) As sample size, *n,* grows larger, the *shape* of the sampling distribution of the mean *tends toward that of a* ***normal curve, regardless of the shape or form of the parent population.*** Thus, it is properly said that the distribution of the sample mean, necessarily has mean μ, and variance σ^2/n, where these Greek letters stand for the population mean and variance respectively. Moreover, such a ***sampling distribution approaches normal form*** as the sample size *n* grows larger for [virtually] *every population!* It is the generality of the last point that is so distinctive. The square root of the variance σ^2/n (written as $\sigma/\sqrt{n}$) is called the ***standard error of the mean*** (*i.e.*, the standard deviation of the sampling distribution of the mean); a term that is frequently encountered in statistical practice. What has just been stated are the principal results of the '***Central limit theorem***.' (Stop to note that the sampling distribution [of any statistic] is itself a population; but such a distribution is wholly distinct from the distribution of the parent population, or from the distribution of any particular sample. Because of it's status as a population, the characteristics of a sampling distribution are generally denoted by Greek letters [consider that all possible samples of a given size were the sources of the statistics]. But don't confuse the sampling distribution (of any statistic…and there is generally a different one for different statistics) with the parent population from which samples were drawn. When we speak about a bootstrap distribution of a statistic we are talking about an *approximate sampling distribution of a particular statistic*, based on a 'large' number of bootstrap samples; and for each sample, the sampling is done *with replacement* from a particular 'sample-as-population'. And each sample is of the same size as the original. Still that 'large' number [1000 below] of statistics is far smaller than the total of *all possible samples*, which is generally *n* to the power *n* in a bootstrap context, or N to the power *n*, for a finite population of size N.)

In practice, bootstrapping entails sampling with replacement from a vector (or 'rows' of matrix or data frame; see next page, bottom for how to do in **R**), so that each bootstrap sample is always the *same size as the original*. But don't confine your thinking to just the mean as we begin to consider bootstrapping; in general, bootstrap distributions can be created for any statistic that can be computed – and each statistic is based on a set of *resampled* data points.

The following illustration begins from a vector ***y*** that contains $n = 100$ values, originally generated as a random sample from the t_3 distribution, i.e. t w/ 3 degrees of freedom, and then scaled to have a mean of 20 and a standard deviation of about 3. This accounts for the relatively long tails of ***y***, compared with a Gaussian (normal) distribution that you see below. See plot of ***y*** (which is both a sample and a population, depending on your point of view – and both ideas are relevant!); its summary statistics (parameters?) are given below.



Here is the R function I used to obtain four central tendency estimates for each of 1000 bootstrap samples: (Copy and paste means4 into your R session)

```
means4 <- function(x,tr1=.1,tr2=.2)
{xm1 <- mean(x)
xmt.1 <- mean(x, trim =tr1)   # 10% trimmed mean
xmt.2 <- mean(x,trim=tr2)     # 20% trimmed mean
xm.5 <- median(x)           # 50% trimmed mean = median
xms <- c(xm1, xmt.1,xmt.2, xm.5)
xms }
```

#the four 'means' above are given as mean 1...mean4 below.

Now, we use the bootstrap function from the library bootstrap:

mns4.y <- bootstrap(y,nboots=1000,means4) ← *command used for bootstrap run (1000 replicates)[nboot >= 1000 for 'good' C.I's]*

I generated 1000 bootstrap replications of the four statistics [for library: bootstrap in R]

Numerical summary of bootstrap results:

```
>cbind(my.summary(xt3),my.summary(mns4.500))
```

	pop.	**mean1**	**mean2**	**mean3**	**mean4**	**#mean1 is just conventional mean.**
means	20.00	19.99	20.01	20.00	20.02	←departures from 20 indicate ***bias***
s.d.s	3.03	***0.30***	***0.26***	***0.26***	***0.24***	#first value is 'popul' s.d./rest
skewns	-1.04	-0.07	0.00	-0.09	-0.06	of s.d.s are bootstrap s.e.'s

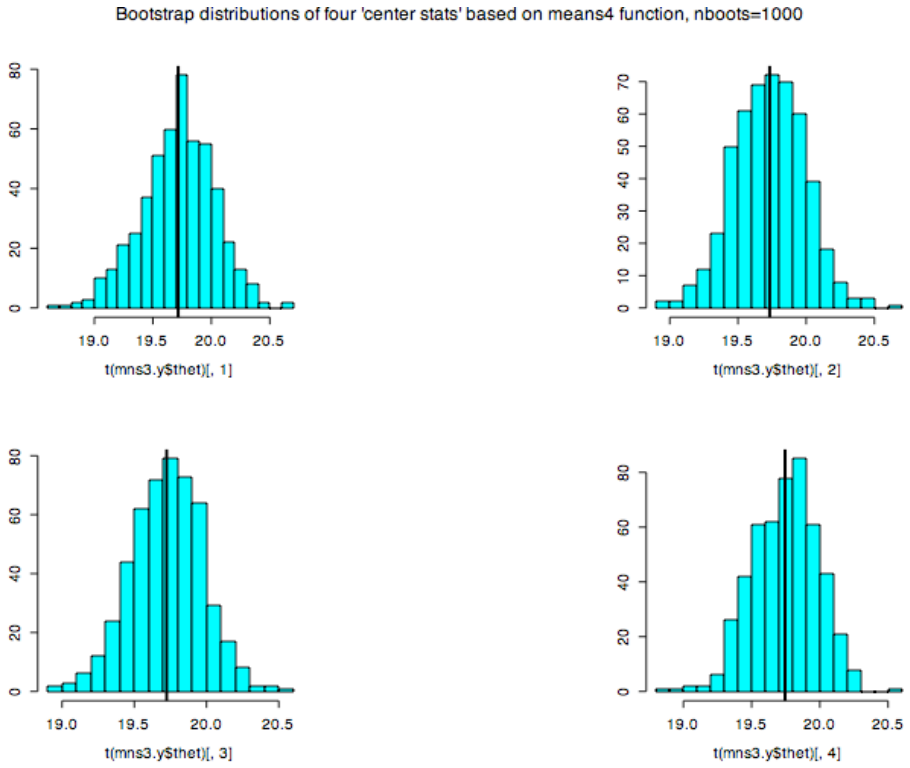
will discuss! See plot next p.

#key results in ***bold italics above AND BELOW.***

A second run, again w/ 1000 bootstrap replictations, gave results:

means	19.99	20.02	20.01	20.03	but I ignore the pop. values here
s.d.s	**0.30**	**0.25**	**0.25**	**0.24**	**as they did not change.**
skewns	-0.07	0.10	0.00	-0.02	For practical purposes, identical.

Following are the four bootstrap distributions for the first set:



NB: The initial 'population' was a long-tailed sample. It's use affords an opportunity to study the conventional sample mean as an estimate of the 'center' of a distribution, when normality does not hold. We in fact see below that the conventional mean is the ***worst*** *of the four estimators of the center of the distribution of the parent population, based on 1000 bootstrap samples. Remember initial sample as population had n=100 scores, so that n = 100 for each sample. Thus, the first s.e., for mean1, can be calculated by theory; that theory says divide the population s.d. by sqrt(n); here 3.03/sqrt(100)=.303.*
We are most informed by the derived standard error estimates; these quantify how well these different estimators of the 'center' of this distribution work in relation to one another.
To repeat, each bootstrap sample entails sampling WITH replacement from the elements of the initial data vector y. Each of the B = 1000 bootstrap samples contains n = 100 resampled scores, and all four statistics ('means') were computed for each bootstrap sample. The summary results, and especially standard error estimates, based on the bootstrap replicates are the principal results on which one will usually focus in studies like this. See the documentation for 'bootstrap' for more information as to what this function does, or can do.

The first major book on the bootstrap was written by Bradley Efron, inventor of the bootstrap, and Tibshirani: '*An introduction to the bootstrap'*, 1993. There are now at least a dozen books, many of them technical, about bootstrapping. The May 2003 issue of *Statistical Science* is devoted exclusively to articles on bootstrapping, for its 25th anniversary. See the first and last web-source below R functions for bootstrapping.

Some things you may find useful about bootstrapping within the world of R:
1. A vector such as y, regarded as y[1:n], where one controls contents, e.g. y[c(1,3)] = 1st and 3rd elements of y; or y[n:1] presents y values in reverse order; or y[sample(1:n,n,repl=T)] yields a bootstrap sample of y, of size n; and the latter, repeated, becomes a basis for bootstrap analysis.
2. A matrix such as yy, regarded as yy[1:n,1:p] (of order n x p) can be examined in parts using bracket notation; e.g. yy[1:3,] displays the first 3 rows of yy; also, to sample the rows of yy, use yy[sample(1:n,n,repl=T),], where comma in [,] separates row and column designations.

Bootstrapping sources on the web:

www.ats.ucla.edu/stat/SPLUS/library/bootstrap.htm ←See programming info (SPLUS is like **R**)
http://www.insightful.com/Hesterberg/bootstrap/ ← See articles and tech. reports in particular.

Latter site provides information about new bootstrap methods and code for SPLUS as well as MANY other interesting things such as how students can get a free copy (!) of SPLUS. Study this site; doing so will uncover articles such as Hesterberg's (with others, 2003?) *Bootstrap Methods and Permutation Tests*, available at http://bcs.whfreeman.com/pbs/cat_160/PBS18.pdf , and also Hesterberg's (1998), *Simulation and Bootstrapping for Teaching Statistics*, Proceedings of the Statistical Education Section, American Statistical Association, 44-52; at least the first of these is readable, and you should certainly examine that one.

R functions for bootstrapping can be found in the **`bootstrap`** and the **`boot`** library, so you should examine the help files for several of the functions in these libraries to see how to proceed. Note that **`bootstrap`** is a much smaller library than **`boot`**, and generally easier to use effectively. I recommend that you begin w/ the function **`bootstrap`** in the library of the same name.

The next page provides a second illustration of the use of the **`means4`** function.

A second illustration using means4 to do boostrapping (R cmds shown)

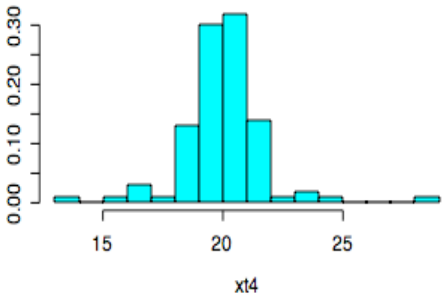
This time we use the following population (also, a sample …to be discussed), of size N = 100:

Mean= 20.00 s.d.= 1.76 skewns =0.28 low= 13.04 high= 28.02

The decimal point is at the |

12 | 0
14 | 7
16 | 6783
18 | 00122467788890001112244444566666777899999
20 | 00111122233334455566677778888999000112467778899
22 | 248
24 | 3
26 |
28 | 0

This population is starting point for 1000 bootstrap samples.
note in, particular that the MEAN = 20.00 and σ = 1.76 AND that it has longer tails than would a normal distribution



Cmds used

```
> bt.xt4<-bootstrap(xt4,1000,theta=means4)
then >par(mfrow(c(2,2)) to set up 4 panels,
>summary(bt.xt4)
                Length Class  Mode
thetastar       4000   -none- numeric
func.thetastar     0   -none- NULL
 . . . .
>truehist(bt.xt4$the[1,],22,col=13,xlim=
c(19,21))              ^ then rows 2,3,4
    repeating same limits to get the plots
that you see below.
```

For bootstraps: **Summary Results**

```
my.summary(t(bt.xt4$the))
       mean tr(.1) tr(.2) median
means 20.01 20.02 20.03  20.04
s.d.s 0.177 0.132 0.127  0.148 ←*
skewns 0.10 -0.02 -0.04  -0.19
krtsis 0.13  0.02 -0.01   0.39
low   19.41 19.52 19.56  19.51
high  20.64 20.48 20.49  20.59
```

See s.d.s especially*, then plots below, where .1 & esp. .2 trimmed means show best recovery of 'population' mean across all reps.

Histograms for 1000 bootstrap reps for: mean, 10%trimd mean, 20%trmd mean, median, n = 100