Bootstrap

The bootstrap as a statistical method was invented in 1979 by Bradley Efron, one of the most influential statisticians still alive. The idea is nonparametric, but is not based on ranks, and is very computationally intensive.

The idea behind the bootstrap is a way of simulating the distribution of the sampling distribution for certain statistics, particularly when it is difficult to derive the distribution from theory. The sampling distribution is usually used in order to get confidence intervals. For example, if you want a confidence interval for a proportion, you can use properties of the binomial distribution to derive the appropriate standard error for $\widehat{p}$.

Bootstrap

Sometimes, instead of the proportion, people report the odds, $\widehat{p}/(1-\widehat{p})$. This statistic is just a function of the data, but its variance (and therefore standard error) are more complicated to derive. Usually approximations to normal distributions are used, and the interval is just approximate.

Other statistics that are functions of sample proportions include the risk ratio, when you have two groups,

$$\widehat{p}_1/\widehat{p}_2$$

and the odds ratio

$$\frac{\widehat{p}_1/(1-\widehat{p}_1)}{\widehat{p}_2/(1-\widehat{p}_2)}$$

Bootstrap

To get a confidence interval for the median, sometimes the Wilcoxon test is used. However, this is based on ranks, and doesn't take full advantage of the data, since the ranks is a simplification of the data. What are other ways to get a confidence interval for the population median? The sampling distribution for the sample mean is a normal distribution if the data comes from a normally distributed population. If the underlying population isn't normal, then the sample mean is still roughly normal due to the Central Limit Theorem. There isn't a Central Limit Theorem that applies to sample medians, however. So if the sample median is used to estimate the population median, it is usually difficult to know what an appropriate standard error is needed, especially if the underlying distribution is unknown.

Bootstrap

The bootstrap is a way to get confidence intervals for quantities like odds, medians, and other aspects of a distribution where the standard error can be difficult to derive.

The way the bootstrap works is it assumes that the data is representative of the population. Consequently, if you sample from the data in your sample, then this is similar to sampling from the population as a whole.

Bootstrap

The idea, is that if we want to know how variable something like the odds is, then ideally we could sample multiple times from the population, and make a histogram of the sample odds, $\widehat{p}/(1-\widehat{p})$. Here we are not using theory to understand the variability, but using the population itself.

Since we usually just have one sample to work with, not many, instead of sampling repeatedly from the population, we sample repeatedly from the sample itself, hoping that the sample is representative of the population. This procedure is called resampling.

Bootstrap

I'll use an example with getting an estimate of the mean μ from a normal distribution, and we'll compare the bootstrap and t-based confidence intervals. The population is from a standard normal with mean 0 and standard deviation 1. Here the sample size is $n = 20$.

```
> x <- rnorm(20,0,1)
> x <- sort(x)
> options(digits=3)
> x
-3.2139 -0.6799 -0.6693 -0.2472 -0.2196
-0.1190 -0.0459 -0.0148  0.0733 0.1220
0.1869  0.2759  0.3283  0.4984  0.5429
0.9491  1.0510  1.4324 1.4534  1.7554
```

Bootstrap

To get a resample—a sample from the sample–we then sample from these 20 observations. To do the sample, we sample with replacement. This means that we think of the there being balls with the values, we draw one ball from a hat, write down the number, put it back in the hat, mix it up, and draw again. We repeat until we have 20 observations. Usually this means some observations won't show up in the sample, and some will show up several times.

```
b <- sample(x,replace=T)
> sort(b)
-0.6799 -0.6799 -0.6693 -0.2196 -0.0459
-0.0148 -0.0148  0.1220  0.1220 0.1869
0.1869  0.2759  0.3283  0.4984  0.4984
0.5429  0.5429  1.0510 1.0510  1.4324
```

Bootstrap

In the previous example, the observation `-0.6799` shows up twice in the resample, while `-3.2139` doesn't show up at all. The observation `0.4033` shows up three times in the resample.

Part of the idea is that the resample will be similar to the original sample, but not exactly the same as the original sample. It should have approximately the same mean, median, and variance as the original, for example. Part of the idea is that if you had obtained a different sample, then your inferences should be similar to what occurred with the sample that you got. If ABC News and Gallup both perform a poll for presidential candidates, hopefully their results are similar to each other, but you don't expect them to be exactly the same.

Bootstrap

```
> mean(x)
[1] 0.173
> mean(b)
[1] 0.226
> median(x)
[1] 0.154
> median(b)
[1] 0.187
> sd(x)
[1] 1.05
> sd(b)
[1] 0.564
```

Bootstrap

The idea with the bootstrap is to repeat this procedure many times and see how variable the resampled values are, such as the mean, median, and standard deviation. Usually this requires something like a `for` loop. Here `b` stores the temporary resample, that gets replaced each iteration.

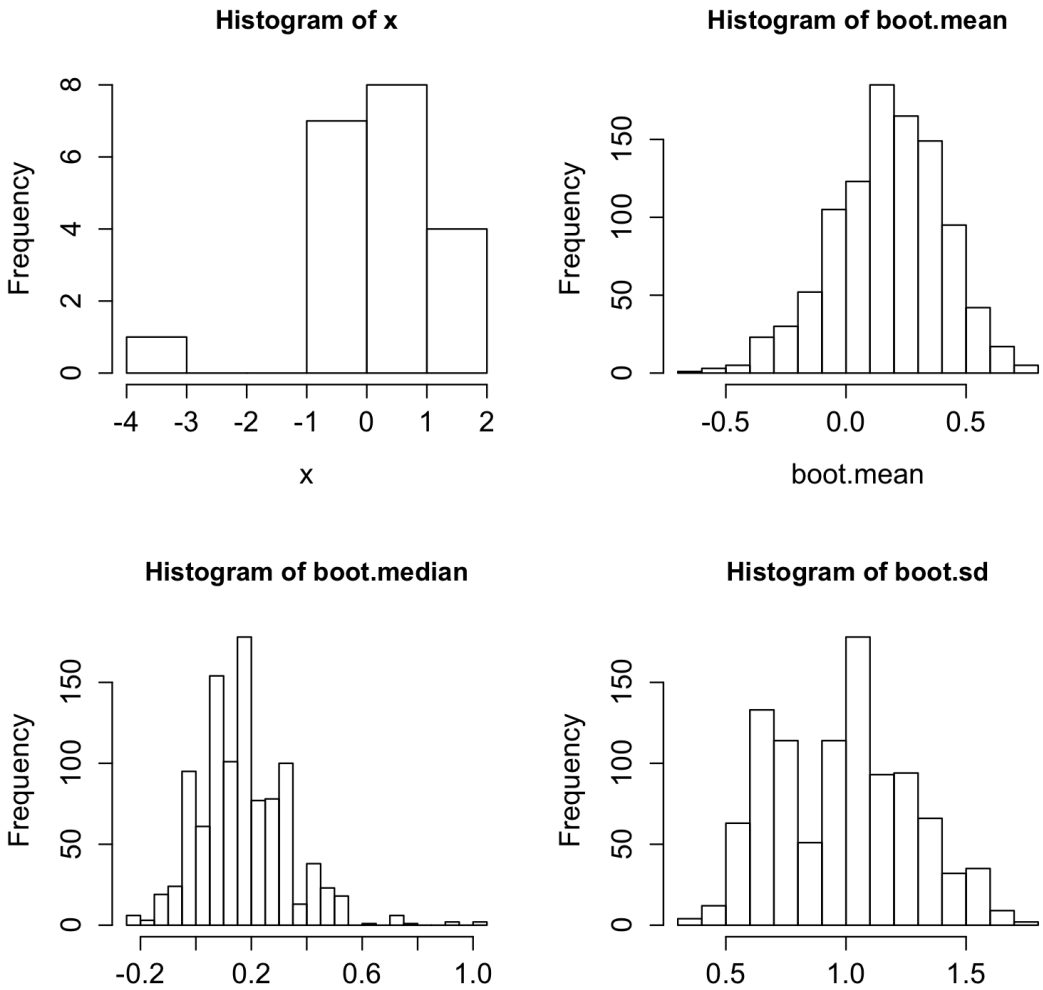
```
I <- 1000
boot.mean <- 1:I
boot.median <- 1:I
boot.sd <- 1:I
for(i in 1:I) {
b <- sample(x,replace=T)
boot.mean[i] <- mean(b)
boot.median[i] <- median(b)
boot.sd[i] <- sd(b)
}
```

Bootstrap

To report a bootstrap CI, you can look at the 2.5 and 97.5 percentiles of the bootstrap distribution. This means sorting the variables `boot.mean`, `boot.median`, and `boot.sd` and examining the appropriate values. The bootstrap distribution can be visualized by a histogram of the bootstrapped sample statistics. For $I = 1000$ bootstraps, the 25th and 976th observations can be used since observations $26, 27, \ldots, 975$ is exactly 950 observations, the middle 95% of the bootstrap distribution.

Bootstrap

There is an outlier, but it was simulated using `x <- rnorm(20)` in R.



Bootstrap

```
boot.mean <- sort(boot.mean)
boot.median <- sort(boot.median)
boot.sd <- sort(boot.sd)
CI.mean <- c(boot.mean[25],boot.median[976])
CI.median <- c(boot.median[25],boot.median[976])
CI.sd <- c(boot.sd[25],boot.sd[976])
CI.mean
#[1] -0.315  0.521
CI.median
# -0.119  0.521
CI.sd
#[1] 0.522 1.563
```

Bootstrap

In this case, the bootstrap intervals include the parameter from the distribution used for the simulation. It is interesting to compare the t-based interval for the mean and Wilcoxon-based interval for the for median.

```
> t.test(x)$conf.int
[1] -0.317  0.663
> wilcox.test(x,conf.int=T)$conf.int
[1] -0.117  0.657
```

The bootstrap CI for the mean, is quite similar to the t-based CI for the mean, and the bootstrap CI for the median is similar to the Wilcoxon-based CI for the median.

Bootstrap

In addition to means and medians, you can get intervals for other quantities, such as the 80th percentile of the distribution (here sort each bootstrap data set, sort it, and pick the 80th percentile, corresponding to observation 16 or 17 in the sorted sample).

For proportion data, you get functions of proportions such as risk ratio and odds ratios.

Bootstrap

As another example, we'll look at the distribution of the risk ratio. Risk ratios are often used in medicine. For example, given either aspirin or placebo, the number of strokes is recorded for subjects in a study. The results are as follow:

	stroke	no stroke	subjects
aspirin	119	10918	11037
placebo	98	10936	11034

To compare the proportion of strokes for aspirin versus placebo takers, we can compare the two proportions:

$$\widehat{p}_1 = \frac{119}{11037} = 0.0108, \quad \widehat{p}_2 = \frac{98}{11034} = 0.00888$$

where p_1 is the proportion of aspirin takers who had a stroke and p_2 is the proportion of placebo takers who experienced a stroke.

Bootstrap

The proportions can be compared by using a test of proportions. However, an issue with this is that the proportions involved are very small:

```
> prop.test(c(119,98),c(11037,11034),correct=F)

2-sample test for equality of proportions without continuity

data:  c(119, 98) out of c(11037, 11034)
X-squared = 2, df = 1, p-value = 0.2
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.000703  0.004504
sample estimates:
 prop 1  prop 2
0.01078 0.00888
```

Bootstrap

For this type of problem, often instead a risk ratio, or relative risk is reported. This gives you an idea of how much more risky it is to have one treatment than another in relative terms, without giving an idea of the absolute risk. In this case, an estimate for the relative risk is

$$\widehat{p}_1/\widehat{p}_2 = 1.21$$

The relative risk is 1.21, which indicates that a random person selected from the aspirin group was 21% more likely to experience a stroke than a person from the placebo group, even though both groups had a fairly low risk (both close to 1%) of experiencing a stroke. In medical examples, a relative risk of 1.21 is fairly large.

We'd also like to get an interval for the relative risk.

Bootstrap

The usual approach for doing this is to take the logarithm of the relative risk, get an interval for the logarithm of the relative risk, and the transform the interval back into the original scale. The reason for this is that the logarithm of a ratio is a difference, and for sums and differences, it is much easier to derive reasonable standard errors.

Bootstrap

	outcome (e.g., stroke)	no outcome	subjects
treatment	x_1	$n_1 - x_1$	n_1
placebo (or control)	x_2	$n_2 - x_2$	n_2

Let $\widehat{RR} = \widehat{p}_1/\widehat{p}_2$ be the estimated relative risk or risk ratio. The standard large sample CI for the log is

$$CI = \log(\widehat{RR}) \pm z_{crit}\sqrt{\frac{(n_1 - x_1)/x_1}{n_1} + \frac{(n_2 - x_2)/x_2}{n_2}}$$

$$= \log(\widehat{RR}) \pm z_{crit}\sqrt{\frac{1}{x_1} - \frac{1}{n_1} + \frac{1}{x_2} - \frac{1}{n_2}}$$

Bootstrap

The to get the interval on the original scale, you then expontiate both endpoints. In the stroke example,

$$SE = \sqrt{\frac{1}{x_1} - \frac{1}{n_1} + \frac{1}{x_2} - \frac{1}{n_2}} = \sqrt{\frac{1}{119} - \frac{1}{11037} + \frac{1}{98} - \frac{1}{11034}} = 0.136$$

The 95% interval is for log RR is therefore (here, $\log 1.21 = 1.91$):

$$0.191 \pm 1.96(0.136) = (-0.0756, 0.458)$$

This is an interval for the log of the relative risk.

Bootstrap

Exponentiating the interval, we get $(0.927, 1.58)$. This is done using

```
> exp(.191-1.96*.136)
[1] 0.927
> exp(.191+1.96*.136)
[1] 1.58
```

The interval includes 1.0, which is the value that corresponds to equal risks. The value 0.927 corresponds to the risk for the aspirin group being 92.7% of the risk of the placebo group, while 1.58 corresponds to the aspririn gorup have a risk that is 58% higher than the placebo group.

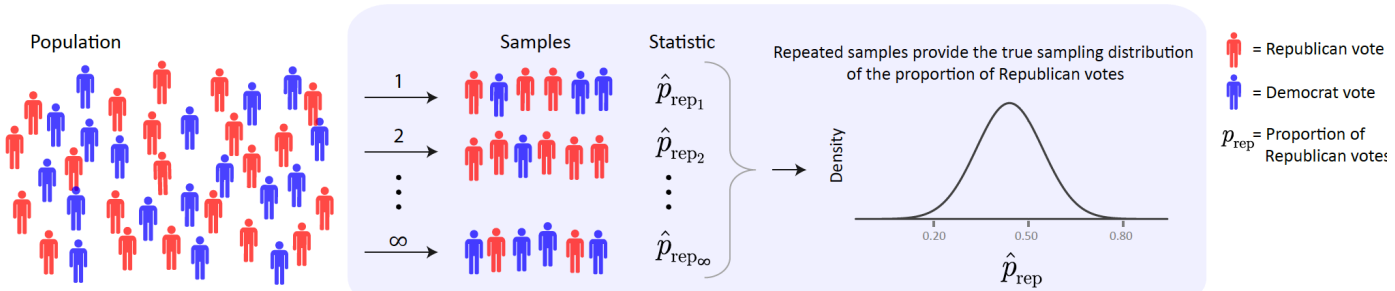
Bootstrap

How to do bootstrapping for proportion data?

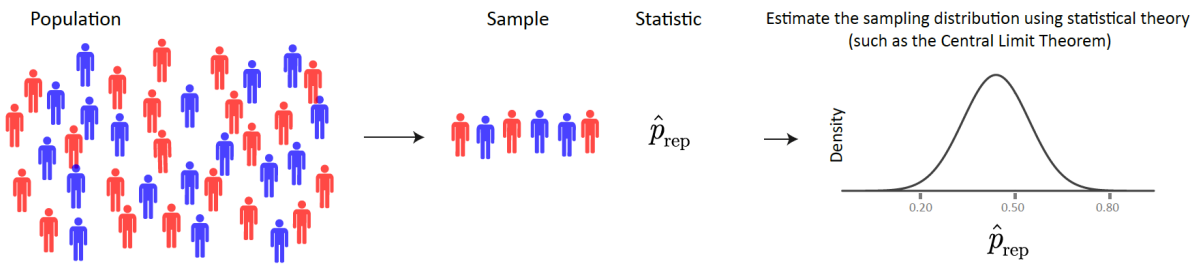
Here we create data sets of 0s and 1s and bootstrap those data sets.

Bootstrap

The ideal case: draw repeated (infinite) samples from the population

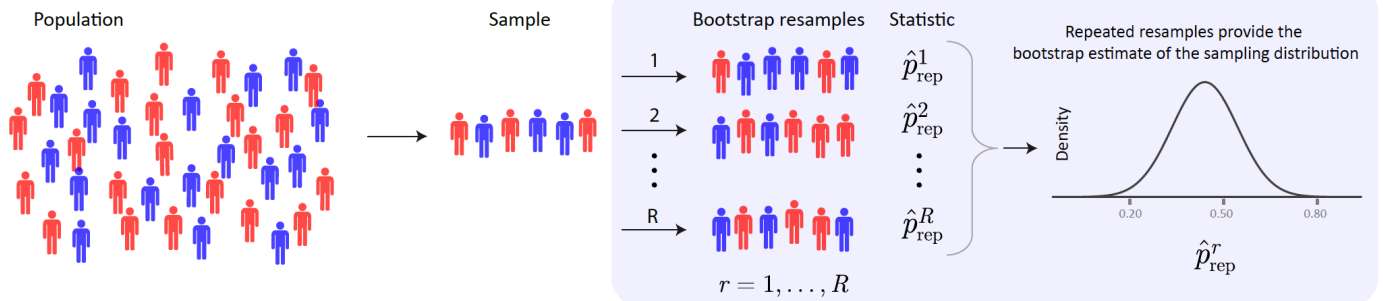


The traditional case: a single sample is observed

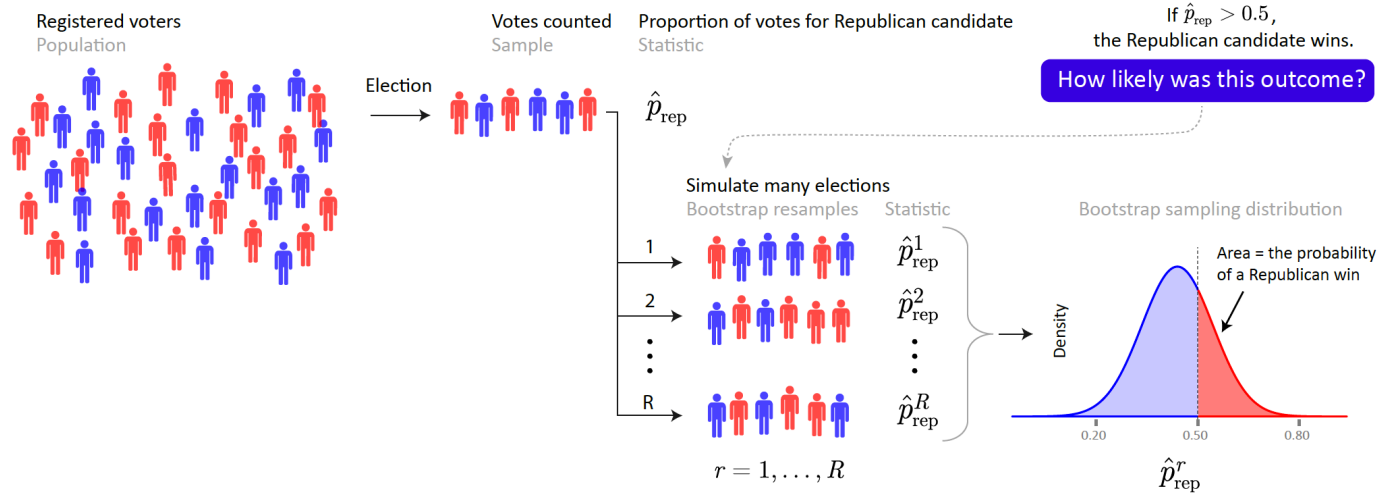


Bootstrap

The bootstrap case: a single sample is resampled



How is the bootstrap used in this scenario?



Bootstrap

For a two-sample proportion case, we need two sets of 0s and 1s (i.e., red and blue) to represent the placebo group and the treatment (aspirin) group).

Bootstrap code

```
aspirin <- c(rep(1,119),rep(0,11037-98))
placebo <- c(rep(1,98),rep(0,11034-98))
boot.rr <- 1:1000
boot.or <- 1:1000
for(i in 1:1000) {
  aspirin.b <- sample(aspirin,replace=TRUE)
  placebo.b <- sample(placebo,replace=TRUE)
  boot.rr[i] <- mean(aspirin.b)/mean(placebo.b)
  p1hat <- mean(aspirin.b)
  p2hat <- mean(placebo.b)
  boot.rr[i] <- p1hat/p2hat
  boot.or[i] <- p1hat*(1-p1hat)/(p2hat*(1-p2hat))
}
> c(sort(boot.rr)[25],sort(boot.rr)[976])
[1] 0.9286731 1.6014550
> c(sort(boot.or)[25],sort(boot.or)[976])
[1] 0.929285 1.594332
```

Bootstrap code

The bootstrap intervals are remarkably close to the interval obtained by exponentiating the interval for the log of the relative risk.

Bootstrap code

Bootstrapping can also be applied to more complex data sets such as regression problems.

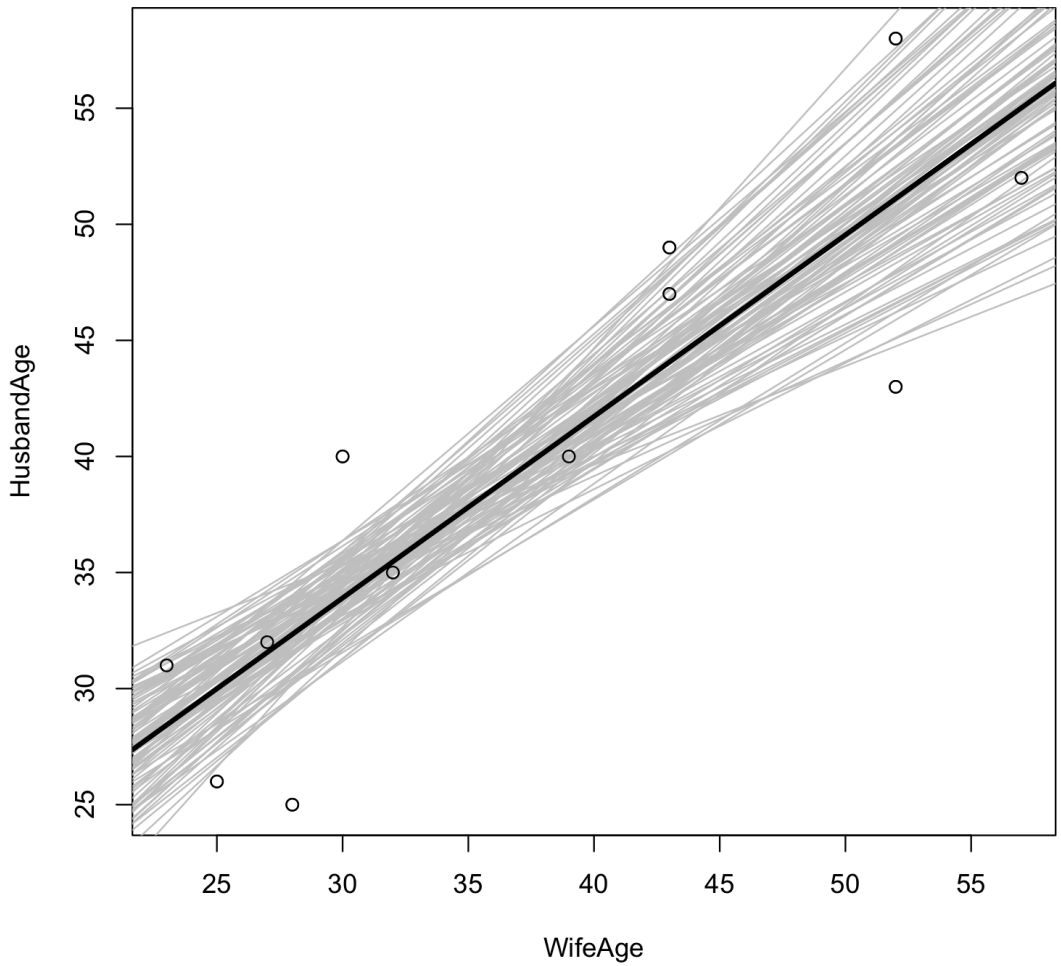
Here, you bootstrap each row in the data set. This means that if x_i appears in the bootstrap sample, then so does the pair (x_i, y_i). This is very different from the permutation test which shuffles x with respect to y.

To sample rows of the data set, you can randomly bootstrap the index for the row you want to include in the bootstrap sample, then apply the rows to a new, temporary data set, or just new vectors for the x and y variables.

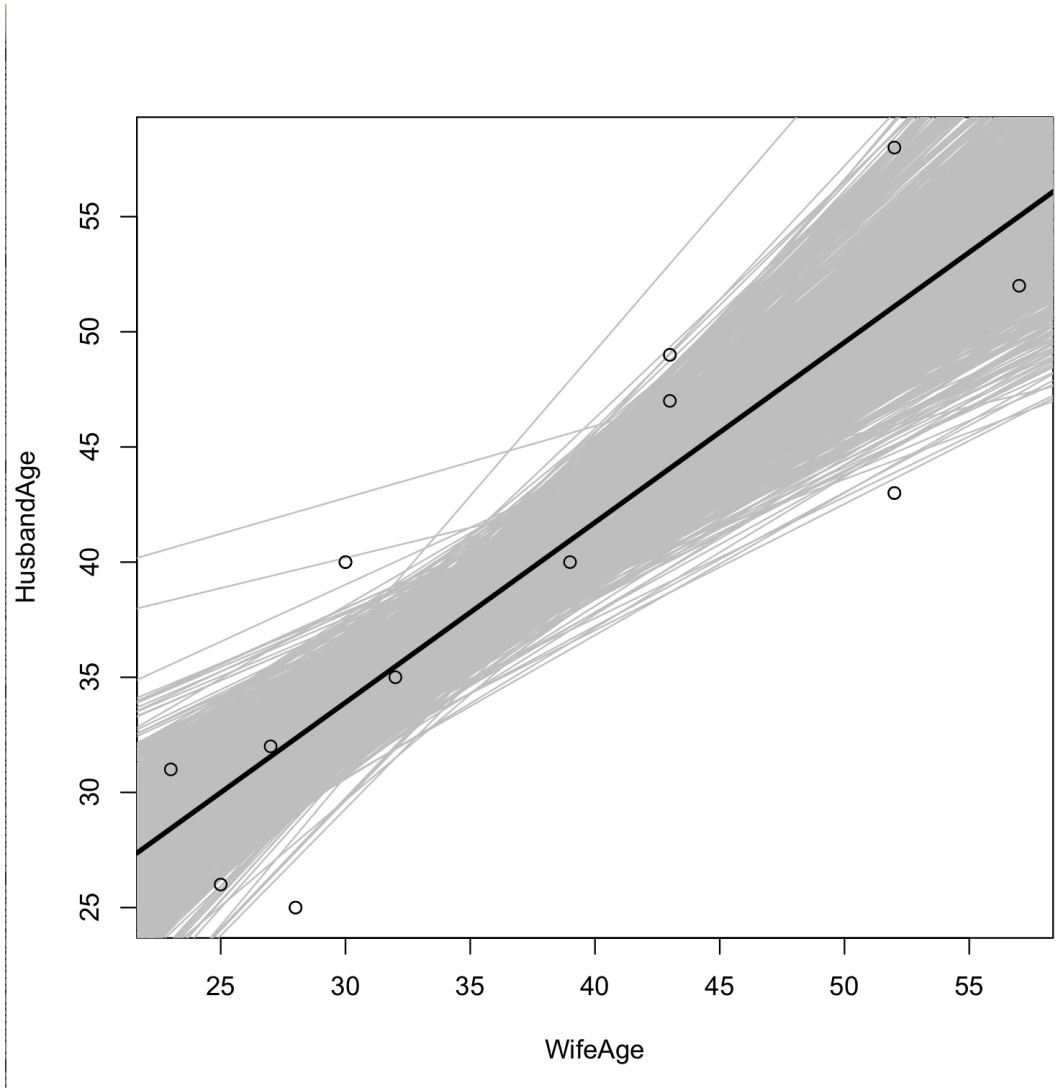
Bootstrap code

```
x <- read.table("couples.txt",header=T)
attach(x)
a <- lm(HusbandAge ~ WifeAge)
abline(a,lwd=3)
plot(WifeAge,HusbandAge)
abline(a,lwd=3)
for(i in 1:100) {
  boot.obs <- sample(1:length(WifeAge),replace=T)
  boot.WifeAge <- WifeAge[boot.obs]
  boot.HusbandAge <- HusbandAge[boot.obs]
  atemp <- lm(boot.HusbandAge ~ boot.WifeAge)
  abline(atemp,col="grey")
}
abline(a,lwd=3) # original is hidden by bootstrap lines
```

Bootstrap, 100 replicates



Bootstrap, 100 replicates



Bootstrap, 100 replicates

The bootstrap is also used extensively in genetics. Here if you have an alignment of DNA sequences, you can sample each site at random to create a new alignment. (On board.)

Bootstrap

An interesting feature of the bootstrap is how it handles outliers. If a data set has an outlier, what is the probability that the outlier is included in one bootstrap sample?

The probability that the outlier is not included is

$$P(\text{no outlier}) = \left(1 - \frac{1}{n}\right)^n$$

where n is the number of observations. The reason is that each observation in the bootstrap sample is not the outlier with probability

$$\frac{n-1}{n} = 1 - \frac{1}{n}$$

because there are $n - 1$ ways to got an observation other than the outlier, and each of the n observations is equally likely.

Bootstrap

If n is large, then

$$P(\text{no outlier}) = \left(1 - \frac{1}{n}\right)^n \approx 0.368$$

How large is large?

n	$(1 - 1/n)^n$
2	0.25
3	0.296
6	0.335
12	0.352
20	0.358
30	0.361
100	0.366

Bootstrap

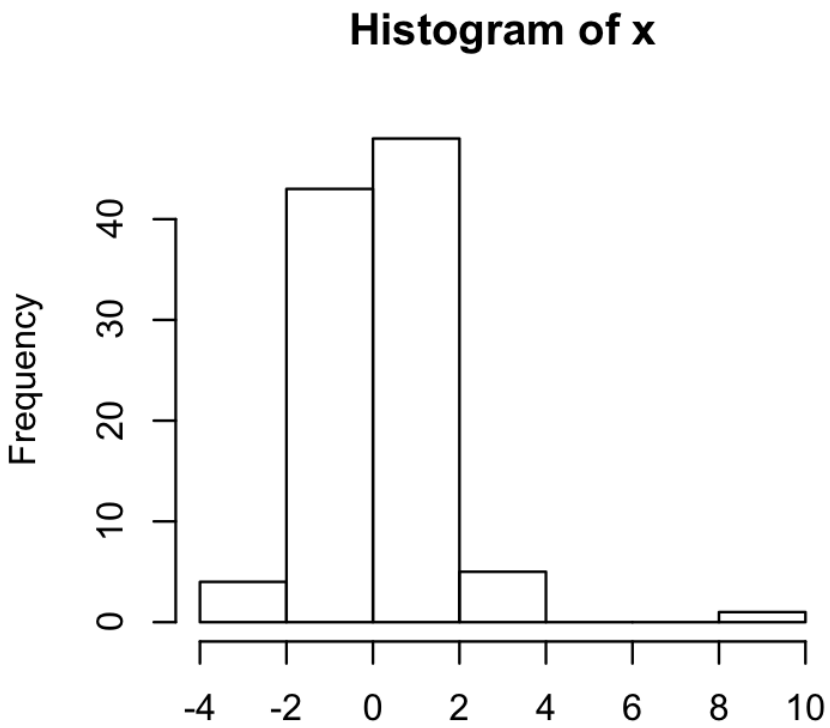
What this means is that approximately $1 - e^{-1} \approx 63\%$ of bootstrap replicates DO have the outlier, but a substantial proportion do not have the outlier. This can lead to interesting bootstrap histograms, where if the outlier is strong enough, the bootstrap samples can be bi- or multi-modal, where the number of modes is the number of times that the outlier was included in the bootstrap sample (recall that in a bootstrap sample, an original observation can occur $0, 1, 2, \ldots, n$ times in theory.

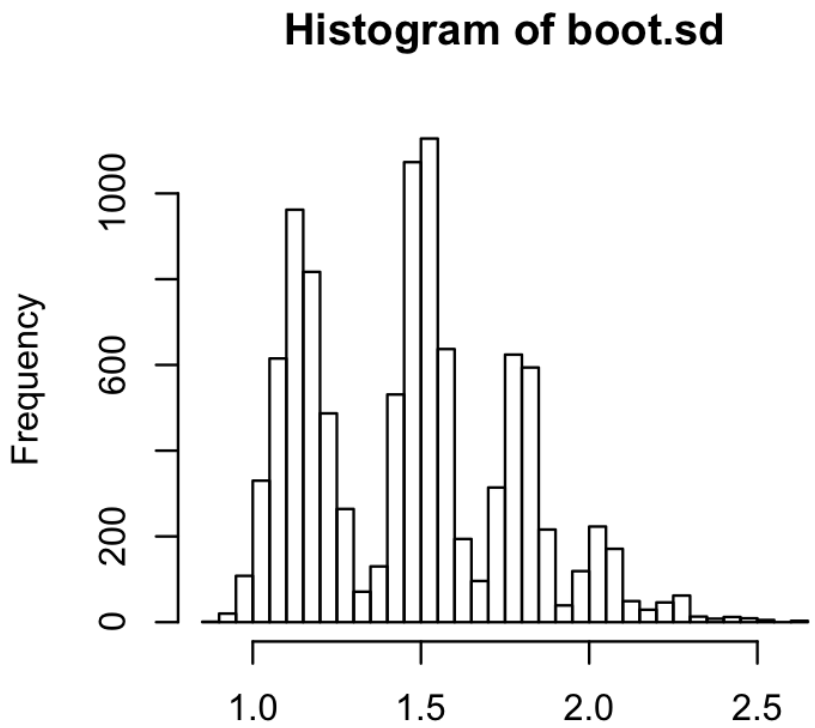
The number of times the outlier appears in a bootstrap sample is a binomial random variable with parameters n and $p = 1/n$. For a data set with 100 regular observations and 1 outlier, the probability that the outlier occurs k times, for $k = 0, \ldots 4$ is

```
> dbinom(0:4,101,p=1/101)
[1] 0.36605071 0.36971121 0.18485561 0.06100235 0.01494558
```

Bootstrap code

```
> x <- rnorm(100)
> x <- c(x,10)
> boot.sd <- 1:10000
> for(i in 1:10000) {
+ temp <- sample(x,replace=T)
+ boot.sd[i] <- sd(temp)
+ }
> hist(boot.sd,nclass=30)
```


Histogram of x
Frequency
0
10
20
30
40
-4
-2
0
2
4
6
8
10


Histogram of boot.sd
Frequency
0
200
600
1000
1.0
1.5
2.0
2.5