Bootstrap

Lauren Perry

The Bootstrap

- The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

Where does the name come from?

▶ The use of the term *bootstrap* derives from the phrase "to pull oneself up by ones bootstraps", widely thought to be based on the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:

The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

▶ It is not the same as the term "bootstrap" used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.

Plug-In and the Bootstrap

▶ A big mistake one can make in statistics is to confuse the sample and the population or to confuse estimators and parameters.

▶ That is, $\hat{\theta}$ is not θ.

▶ Plug-in principle seems to say the opposite
  ▶ Sometimes it is okay to just plug in an estimate for an unknown parameter.
  ▶ We do this, for example, when plugging in s for σ when doing a t-test.

▶ The *bootstrap* is a cutesy name for a vast generalization of the plug-in principle.

Illustrative Example

Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, where X and Y are random quantities.

▶ We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.

▶ We wish to choose α to minimize the total risk of our investment. In other words, we want to minimize $\text{Var}(\alpha X + (1 - \alpha) Y)$.

▶ It turns out the value that minimizes risk is

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Illustrative Example

- In this case, all of σ_X^2, σ_Y^2, and $2\sigma_{XY}$ are unknown.
- We can compute estimates $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$, and $2\hat{\sigma}_{XY}$ using a dataset with measurements for X and Y.

Then we can estimate risk as

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$
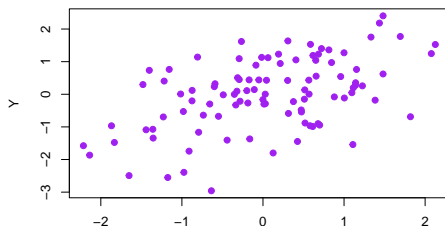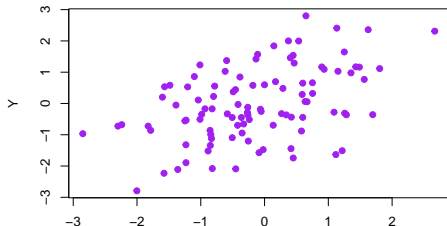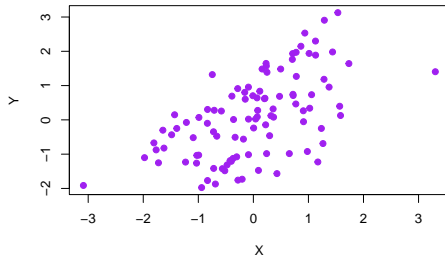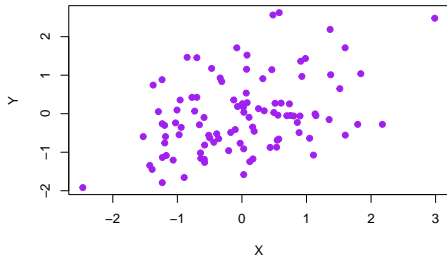
Illustrative Example

Here we simulate 100 paired observations for X, Y several times and estimate α.

```r
set.seed(0)
par(mfrow=c(2,2))
alp <- numeric(4)
for(i in 1:4){
  XY <- rmvnorm(100,sigma=matrix(c(1, 0.5, 0.5, 1.25), 2))
  X <- XY[,1]; Y <- XY[,2]
  plot(X, Y, pch=19, col="purple")

  sig2.x <- var(X)
  sig2.y <- var(Y)
  cov.xy <- cov(X,Y)
  alp[i] <- (sig2.y - cov.xy)/(sig2.x + sig2.y - 2*cov.xy)
}
```
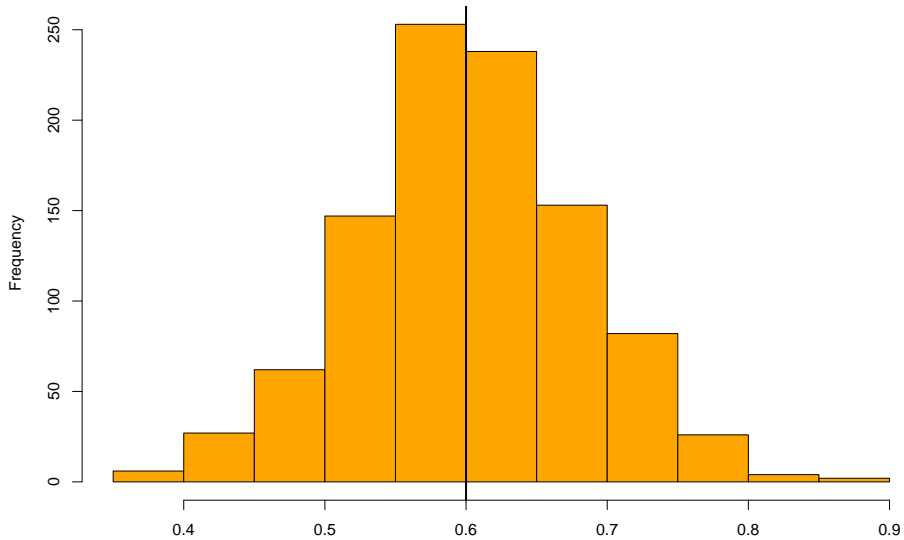
Illustrative Example

Illustrative Example

To estimate the standard deviation of α, we repeat the process 1000 times, estimating α each time.

```r
set.seed(0)
alp <- numeric(1000)
for(i in 1:1000){
  XY <- rmvnorm(100,sigma=matrix(c(1, 0.5, 0.5, 1.25), 2))
  X <- XY[,1]; Y <- XY[,2]

  sig2.x <- var(X)
  sig2.y <- var(Y)
  cov.xy <- cov(X,Y)
  alp[i] <- (sig2.y - cov.xy)/(sig2.x + sig2.y - 2*cov.xy)
}
```

Illustrative Example

Illustrative Example

```r
mean(alp)
```

```
## [1] 0.6029877
```

```r
sd(alp)
```

```
## [1] 0.07990753
```

So we would expect an estimate $\hat{\alpha}$ to differ from $\alpha = 0.6$ by about 0.08, on average.

Bootstrap

- The bootstrap does basically the same thing as the simulations we've done so far.
- The difference is that, instead of simulating from some known data-generating mechanism, we simulate from the data.

Nonparametric Bootstrap

- The notation $X_1^*, \ldots, X_n^*$ IID $\hat{F}_n$ means $X_1^*, \ldots, X_n^*$ are independent and identically distributed from the empirical distribution of the real data
- Sampling from the empirical distribution is just like sampling from a finite population, where the population is the real data $X_1, \ldots, X_n$
  - To be IID sampling must be with replacement
- $X_1^*, \ldots, X_n^*$ are a sample with replacement from $X_1, \ldots, X_n$
- Called *resampling*

Nonparametric Bootstrap

- We want to know the sampling distribution of $\hat{\theta}_n$ or $\hat{\theta}_n - \theta$ or $\frac{\hat{\theta}_n - \theta}{s(\hat{F}_n)}$ or some function thereof
  - This sampling distribution depends on the true unknown distribution F of the real data
- May be very difficult or impossible to calculate theoretically

Nonparametric Bootstrap

- In the bootstrap world everything is known, $\hat{F}_n$ plays the role of the true unknown distribution and $\hat{\theta}_n$ plays the role of the true unknown parameter value
- The sampling distribution of θ_n^* or $\theta_n^* - \hat{\theta}_n$ or $\frac{\theta_n^* - \hat{\theta}_n}{s(F_n^*)}$ may still be difficult to calculate theoretically, but it can always be calculated by simulation.
- The bootstrap analogy works when the empirical distribution $\hat{F}_n$ is close to the true unknown distribution F
  - Usually the case when the sample size n is large and not otherwise

Illustrative Example
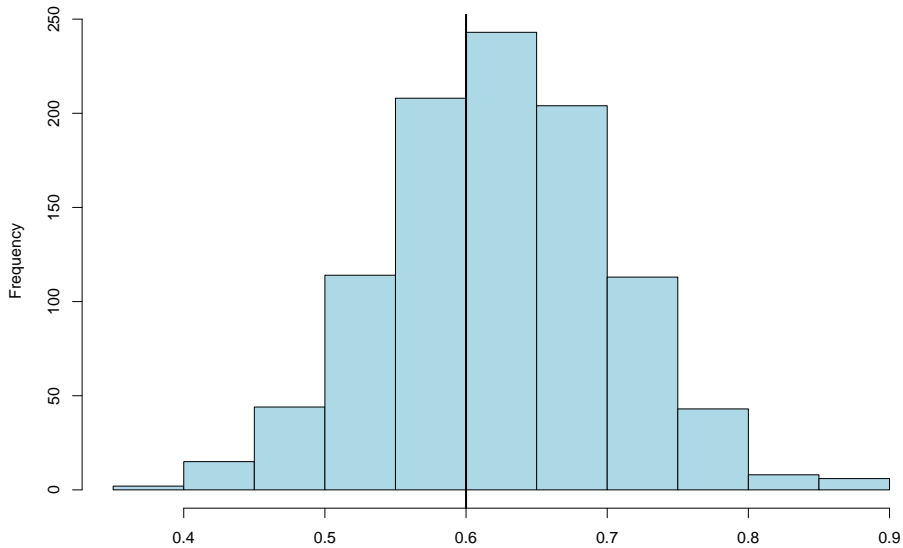
```
set.seed(1)
XY <- rmvnorm(100,sigma=matrix(c(1, 0.5, 0.5, 1.25), 2))
X <- XY[,1]; Y <- XY[,2]

alp <- numeric(1000)
for(i in 1:1000){
  ind <- sample(nrow(XY), replace=TRUE)
  xy.boot <- XY[ind,]

  sig2.x <- var(xy.boot[,1])
  sig2.y <- var(xy.boot[,2])
  cov.xy <- cov(xy.boot[,1],xy.boot[,2])
  alp[i] <- (sig2.y - cov.xy)/(sig2.x + sig2.y - 2*cov.xy)
}
```
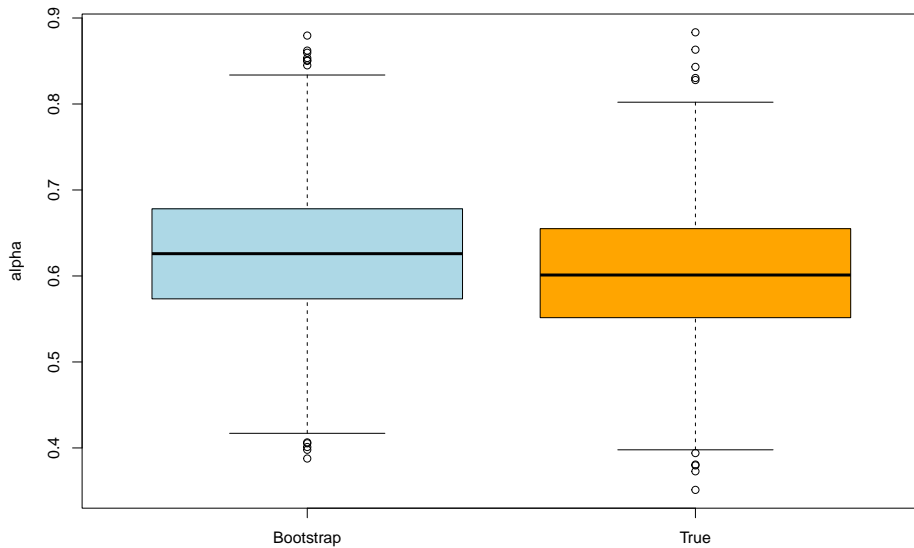
Illustrative Example

Illustrative Example

Bootstrap Percentile Intervals

- Simplest method of making confidence intervals for the unknown parameter is to take $\alpha/2$ and $1 - \alpha/2$ quantiles of the bootstrap distribution of the estimator θ_n^* as endpoints of the $100(1 - \alpha)\%$ confidence interval
- Percentile method only makes sense when there is a symmetrizing transformation (some function of $\hat{\theta}$ has an approximately symmetric distribution with the center of symmetry being the true unknown parameter value θ)
- The symmetrizing transformation does not have to be known, but it does have to exist

Parametric Bootstrap

- The parametric bootstrap is just like the nonparametric bootstrap except for one difference in the analogy
- We use a parametric model $F_{\hat{\theta}_n}$ rather than the empirical distribution $\hat{F}_n$ as the analog of the true unknown distribution in the bootstrap world

Parametric Bootstrap

- Simulation from the parametric model $F_{\hat{\theta}_n}$ not analogous to finite population sampling and does not resample the data like the nonparametric bootstrap does
- Instead we simulate the parametric model
- May be easy (when R has a function to provide such random simulations) or difficult

Nonparametric versus Parametric

▶ When the parameter θ is defined in terms of the parametric statistical model and can only be estimated using the parametric model (by maximum likelihood perhaps), the statistical model is needs to be correct for the parameter estimate $\hat{\theta}_n$ to make sense

▶ Since we already need the statistical model to be correct, the parametric bootstrap is the logical choice
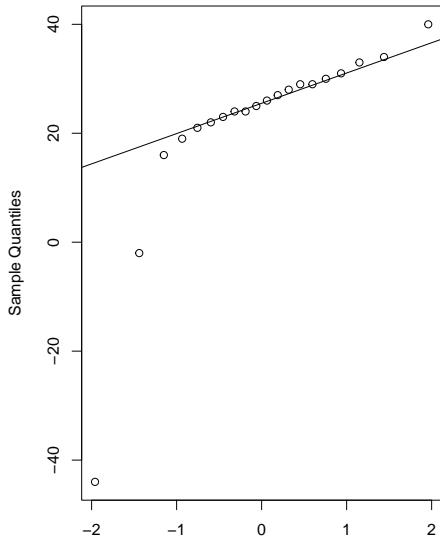
Abnormal speed of light data

- ▶ In 1882 Simon Newcomb performed an experiment to measure the speed of light
- ▶ Measured time it took for light to travel from Fort Myer on the west bank of the Potomac River to a fixed mirror at the foot of the Washington monument 3721 meters away
- ▶ In the units of the data, the currently accepted "true" speed of light is 33.02
- ▶ Note: to convert these units to time in the millionths of a second, multiply by 10^{-3} and add 24.8
- ▶ Does the data support the current accepted speed of 33.02?

Abnormal speed of light data

Histogram of speed
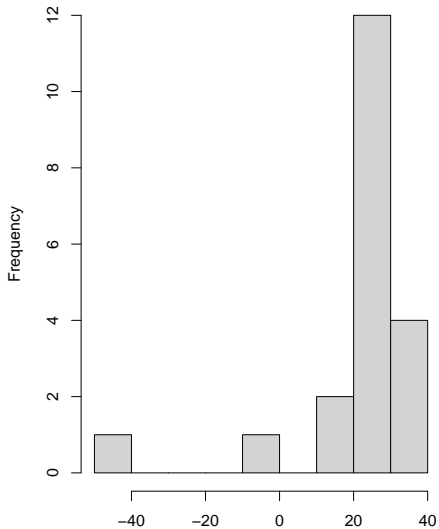
Normal Q–Q Plot

Abnormal speed of light data

- A t-test assumes the population of measurements is normally distributed
- With this small sample size and a severe departure from normality, we can't be guaranteed a good approximation
- Instead, we can consider the bootstrap

Abnormal speed of light data

1. State null and alternative hypotheses

$$H_0 : \mu = 33.02 \text{ versus } H_a : \mu \neq 33.02$$

2. Choose a significance level, in our case 0.05
3. Choose a test statistic, since we wish to estimate the mean speed we can use the sample average
4. Find the observed value of the test statistic
5. Calculate a p-value?

Abnormal speed of light data

- We now need a p-value, but we don't have the sampling distribution of our test statistic when the null hypothesis is true
    - It is approximately normal, but that is a poor approximation here
    - Instead we can perform a simulation under conditions in which we know the null hypothesis is true
    - Use our data to represent the population, but first we shift it over so that the mean really is 33.02

```
newspeed <- speed - mean(speed) + 33.02
```

- Histogram of newspeed will have exactly the same shape as speed, but will be shifted

Abnormal speed of light data

- ▶ Now we reach into our fake population and take out 20 observations at random, with replacement
- ▶ We take out 20 because that's the size of our initial sample
- ▶ We calculate the average and save it, then repeat this process many, many times
- ▶ Now we have a sampling distribution with mean 33.02
- ▶ Can compare this to our observed sample average and obtain a p-value

```
n <- 10000
bstrap <- double(n)
for (i in 1:n){
  newsample <- sample(newspeed, 20, replace=T)
  bstrap[i] <- mean(newsample)
}
```

Bootstrap Sampling Distribution

Frequency

Bootstrap Samples
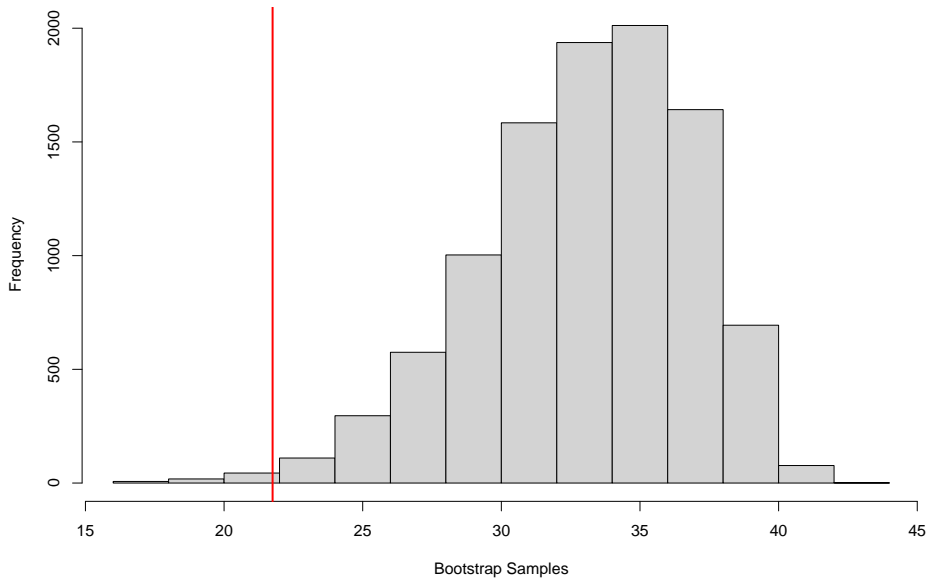
Abnormal speed of light data

- Doesn't look normal, which means we did the right thing
  - Not impossible for the sample average to be 21.75
  - But it's not all that common, either

Abnormal speed of light data

- ▶ The p-value is the probability of getting something more extreme than what we observed
- ▶ Notice 21.75 is $33.02 - 21.75 = 11.27$ units away from the null hypothesis
- ▶ So p-value is the probability of being more than 11.27 units away from 33.02

```
(sum(bstrap < 21.75) + sum(bstrap > 44.29))/10000
```

```
## [1] 0.0061
```

- ▶ Since our significance level is 5%, we reject H_0 and conclude that Newcomb's measurements were not consistent with the currently accepted figure

Example: Sleep study

▶ The two sample t-test checks for differences in means according to a known null distribution
▶ Similar to permutation tests
▶ Let's resample and generate the sampling distribution under the bootstrap assumption

Example: Sleep study

```
bootstrap.resample <- function(object){
  sample(object, length(object), replace=TRUE)
}
diff.in.means <- function(df){
  mean(df[df$group==1,"extra"]) - mean(df[df$group==2,"extra"])
}
resample.diffs <- replicate(2000,diff.in.means(
    sleep[bootstrap.resample(1:nrow(sleep)),]
  ))
```

Example: Sleep study

Bootstrap Sampling Distribution
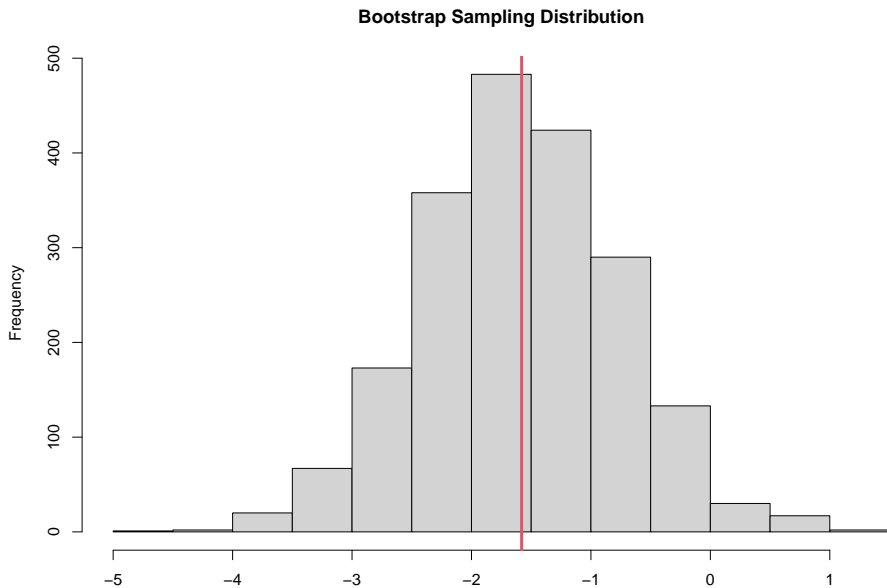
Bootstrapping functions

- ▶ R has numerous built in bootstrapping functions, too many to mention
  - ▶ See `boot` library
- ▶ Bootstrap of the **ratio of means** using the `city` data included in the `boot` package

```
library(boot)
data(city)
ratio <- function(d, w) sum(d$x * w)/sum(d$u * w)
results <- boot(city, ratio, R=1000, stype="w")
```

Bootstrapping functions

```
results
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = city, statistic = ratio, R = 1000, stype = "w")
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 1.520313 0.05128372   0.2248831
```

Bootstrapping functions

```
boot.ci(results, type="bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca")
##
## Intervals :
## Level       BCa
## 95%   ( 1.223,  2.085 )
## Calculations and Intervals on Original Scale
```

Bootstrapping a single statistic

▶ Can use the bootstrap to generate a 95% confidence interval for R-squared
▶ Linear regression of miles per gallon (mpg) on car weight (wt) and displacement (disp)
▶ Data source is mtcars
▶ The bootstrapped confidence interval is based on 1000 replications

```
rsq <- function(formula, data, indices) {
 d <- data[indices,]
 fit <- lm(formula, data=d)
 return(summary(fit)$r.square)
}
results <- boot(data=mtcars, statistic=rsq,
    R=1000, formula=mpg~wt+disp)
```

Bootstrapping a single statistic

```
results
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = mtcars, statistic = rsq, R = 1000, formula = mpg ~
##     wt + disp)
##
##
## Bootstrap Statistics :
##      original     bias    std. error
## t1* 0.7809306 0.01583864  0.04751768
```
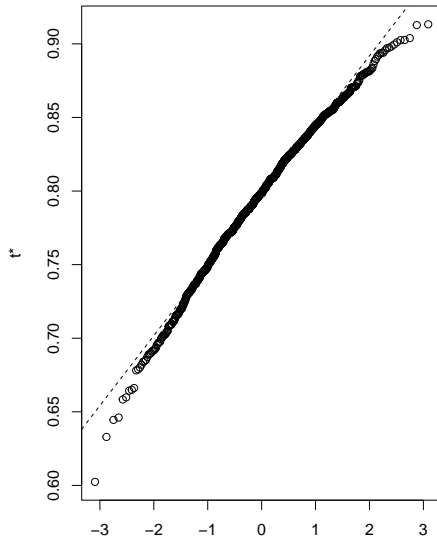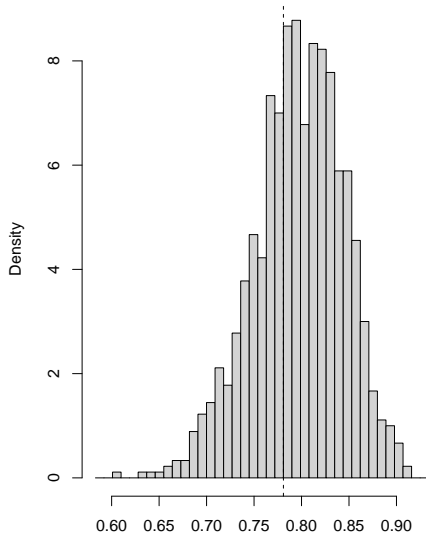
Bootstrapping a single statistic

```
boot.ci(results, type="bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "bca")
##
## Intervals :
## Level       BCa
## 95%   ( 0.6437,  0.8516 )
## Calculations and Intervals on Original Scale
## Some BCa intervals may be unstable
```

Bootstrapping a single statistic

Histogram of t

Summary

- ▶ Bootstrapping provides a nonparametric approach to statistical inference when distributional assumptions may not be met
- ▶ Enables calculation of standard errors and confidence intervals in a variety of situations, e.g. medians, correlation coefficients, regression parameters, . . .
- ▶ Hypothesis tests are a little more challenging
- ▶ The bootstrap is large sample, approximate, and asymptotic!
- ▶ Works when the empirical distribution $\hat{F}_n$ is close to the true unknown distribution F
- ▶ Usually the case when the sample size n is large and not otherwise, no method can save bad data!