```
## Warning:  package ’quadprog’ was built under R version 3.6.0
```

Bootstrapping: Examples and Counterexamples

W. John Braun

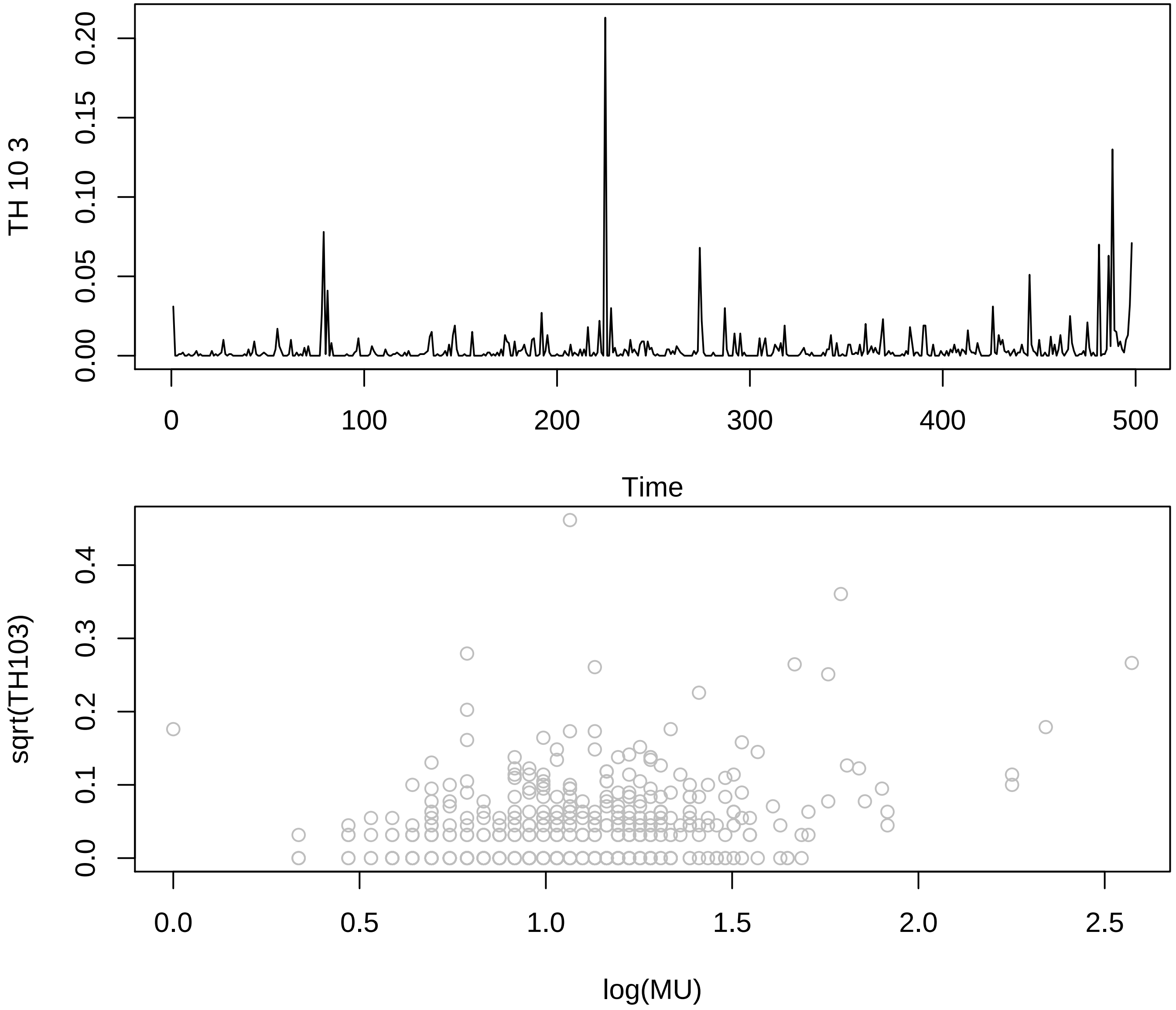
University of British Columbia

12 June 2018

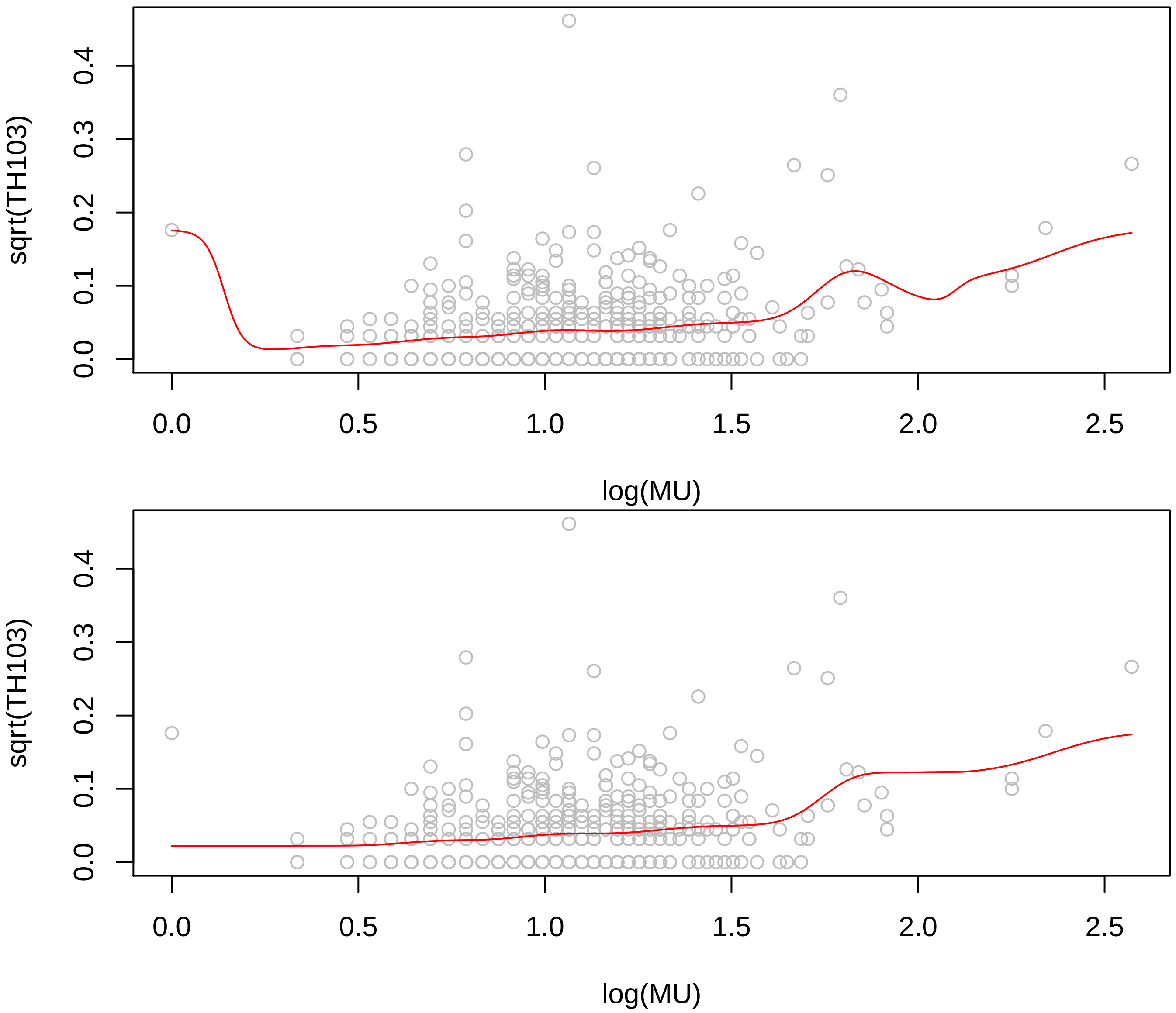
Outline

1. **Preamble - Identification of TH 103 threshold**
   (a) **data sharpening for local polynomial regression**
   (b) **quantile regression**
   (c) **Tatum's robust estimate of variance**
2. **Introduction to the Bootstrap**
3. **Confidence Intervals**
4. **Implementation in R with the `boot()` Library**
5. **Bootstrap Failure**

Preamble: Identification of TH 103 threshold

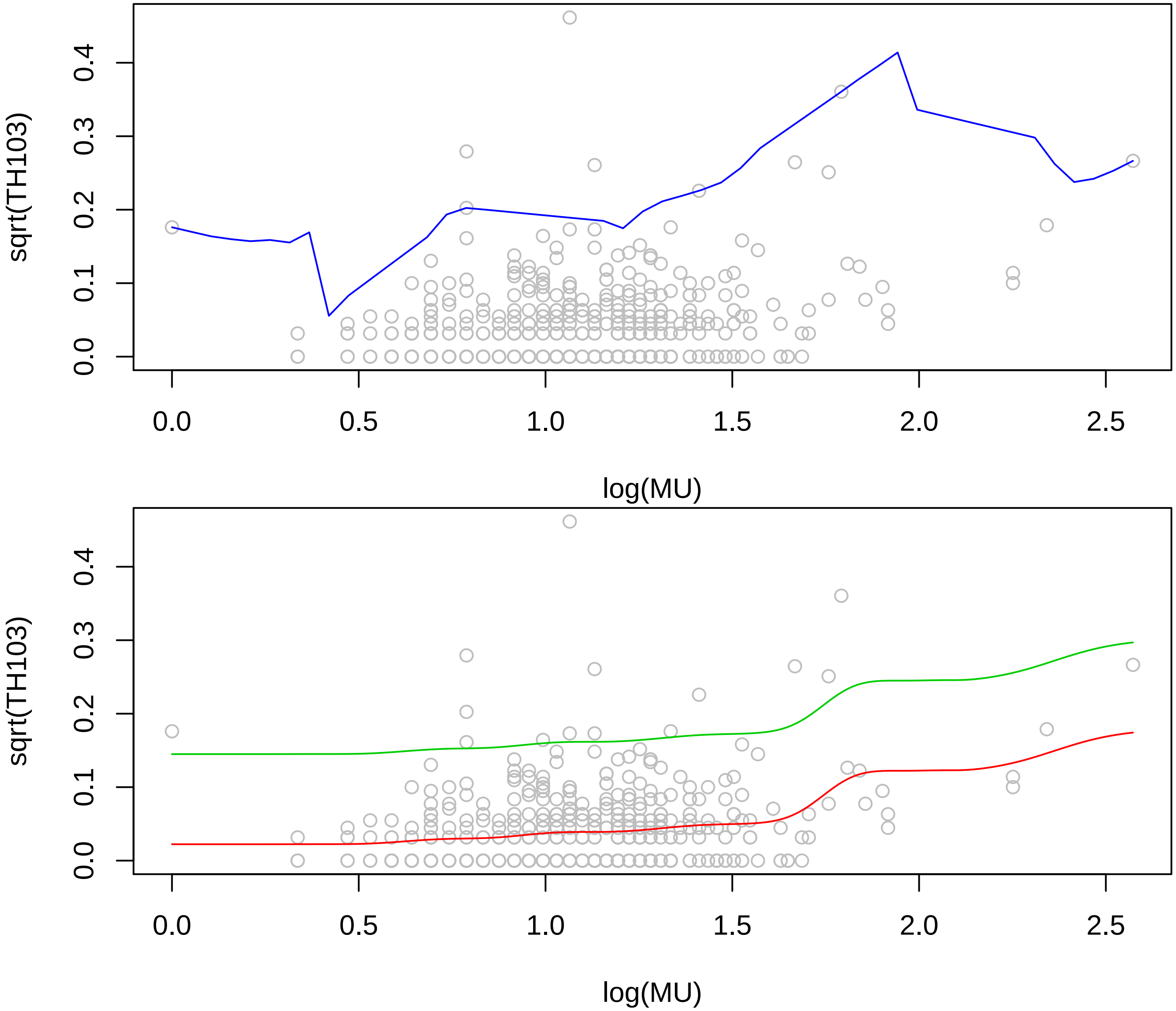


Preamble: Identification of TH 103 threshold



Top panel: local constant regression (Nadaraya, 1964); bottom panel: data sharpened local constant regression subject to a monotonicity constraint (Braun and Hall, 2001).

Preamble: Identification of TH 103 threshold



Top panel: quantile regression at the 99th percentile (Koenker, 2004, 2018); bottom panel: data sharpened local constant regression subject to a monotonicity constraint plus robustly estimated 3σ limits (Tatum, 1997) mitigating the effects of outliers.

Need to Assess Estimation Uncertainty in...

local constant regression estimates

data sharpened monotonic regression estimates

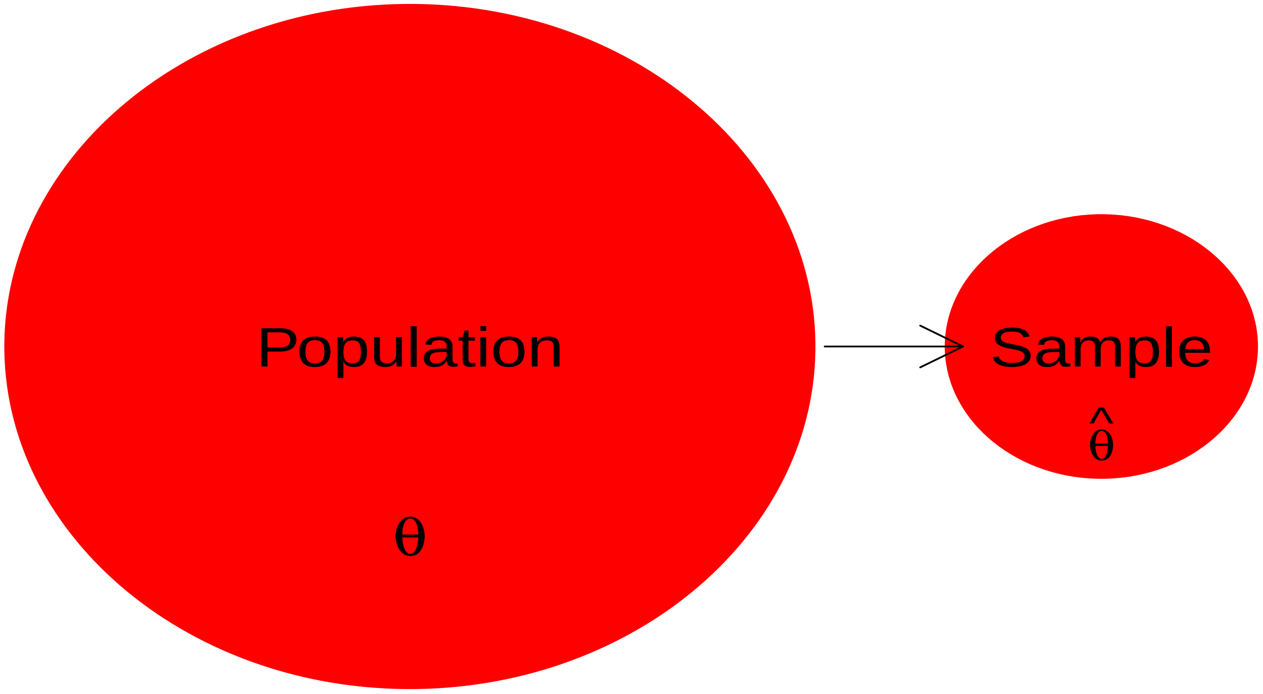
quantile regression estimates

robustly estimated standard deviation (avoiding the effects of outliers)

.... when the statistic is complicated, and it is difficult to derive standard errors and confidence intervals, we often resort to the bootstrap; for some of the above statistics, the bootstrap will work well but for some, it might fail... how can we tell beforehand?

Introduction

The usual statistical problem:



Estimate the population parameter θ using the sample estimate $\widehat{\theta}$.

Introduction

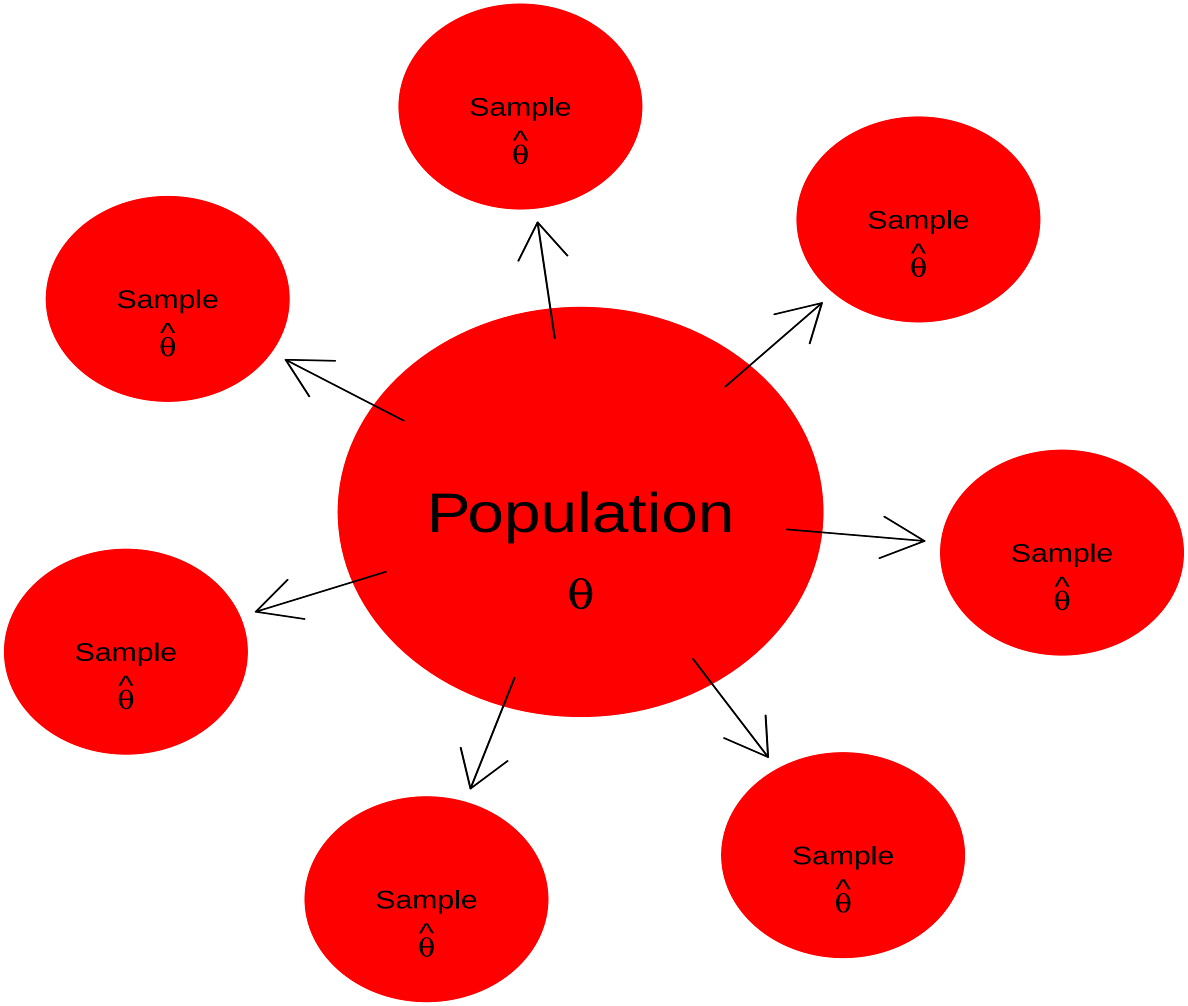
Fact: $\theta \neq \widehat{\theta}$

Question: How wrong is the estimate?

Statistical Answer: assess variability of $\widehat{\theta}$

standard errors, confidence intervals, p**-values for hypothesis tests about** θ

In an ideal world:



Assess variability of the sample estimate $\widehat{\theta}$ by taking additional samples, obtaining new estimates of θ each time.

Introduction

Statistical theory, based on independence and normality assumptions, for example, has allowed us to obtain formulas for variances, standard errors, confidence intervals and p-values for a large number of statistics.

e.g. recall: standard error of the sample mean, $\bar{X}$: $\sigma/\sqrt{n}$

Problem! What if our assumptions are not quite right? Or, what if we can't obtain a useful formula?

Introduction: Example

What is the standard error of the trimmed mean?

Introduction: Standard error of the trimmed mean

Consider the sample

```
78 86 92 81 3 89 94 88 79 85
```

The α trimmed mean is the average of the inner $(1 - 2\alpha)$ values in the sample.

For example, if $\alpha = .2$, then the trimmed mean is the average of the inner 60% of the observations.

Introduction: Standard error of the trimmed mean

We can compute the $\alpha = .2$ trimmed mean in R for the above sample with the following code:

```
x <- c(78, 86, 92, 81, 3, 89, 94, 88, 79, 85)
x.tmean <- mean(x, trim=.2)
x.tmean


## [1] 84.66667
```

This is the average of the 6 innermost observations, i.e. excluding $3, 78, 92$ and 94.

Note: untrimmed mean is 77.5, due to outlier.

Introduction: Standard error of the trimmed mean

It is always good practice to supply a standard error estimate.

For the trimmed mean, this is not straightforward.

Recall that the standard error is the standard deviation of the estimator based on the underlying distribution of the data.

Introduction: Parametric Bootstrap

For definiteness, let's suppose that the data have really come from a normal distribution. (The outlier is thus assumed anomalous.)

Estimated mean: 84.7. **(trimmed mean)**

Estimated standard deviation: 6.86. **(IQR/**1.349**)**
i.e.

```
diff(quantile(x, c(.25, .75)))/1.349


##      75%
## 6.856931
```

Introduction: Parametric bootstrap

We can simulate from this fitted distribution repeatedly, computing the trimmed mean in each case.

The collection of trimmed means is then used to estimate the standard deviation of the trimmed mean for such a population.

This technique pre-dates the bootstrap: Monte Carlo simulation with a new name.

Introduction: Parametric bootstrap

One such simulated sample and estimate is obtained as follows:

```
set.seed(1038087)  # this allows you to obtain
             # the same numerical values below.
x1 <- rnorm(10, mean=x.tmean, sd=6.86)
x1trim <- mean(x1, trim=.2)
x1trim

## [1] 82.46431
```

Introduction: Parametric bootstrap

We generate 10000 samples and the respective trimmed mean estimates. Then we calculate the standard deviation of the whole collection as follows:

```
x10000 <- replicate(10000,
   mean(rnorm(10, mean=x.tmean, sd=6.86), trim=.2))
sd(x10000)


## [1] 2.285725
```

Note: that the standard error of the ordinary sample mean in this case is $s/\sqrt{n} = 6.86/\sqrt{10} = 2.17$**.**

Introduction: Standard error of the trimmed mean

We often do not know the true underlying distribution.

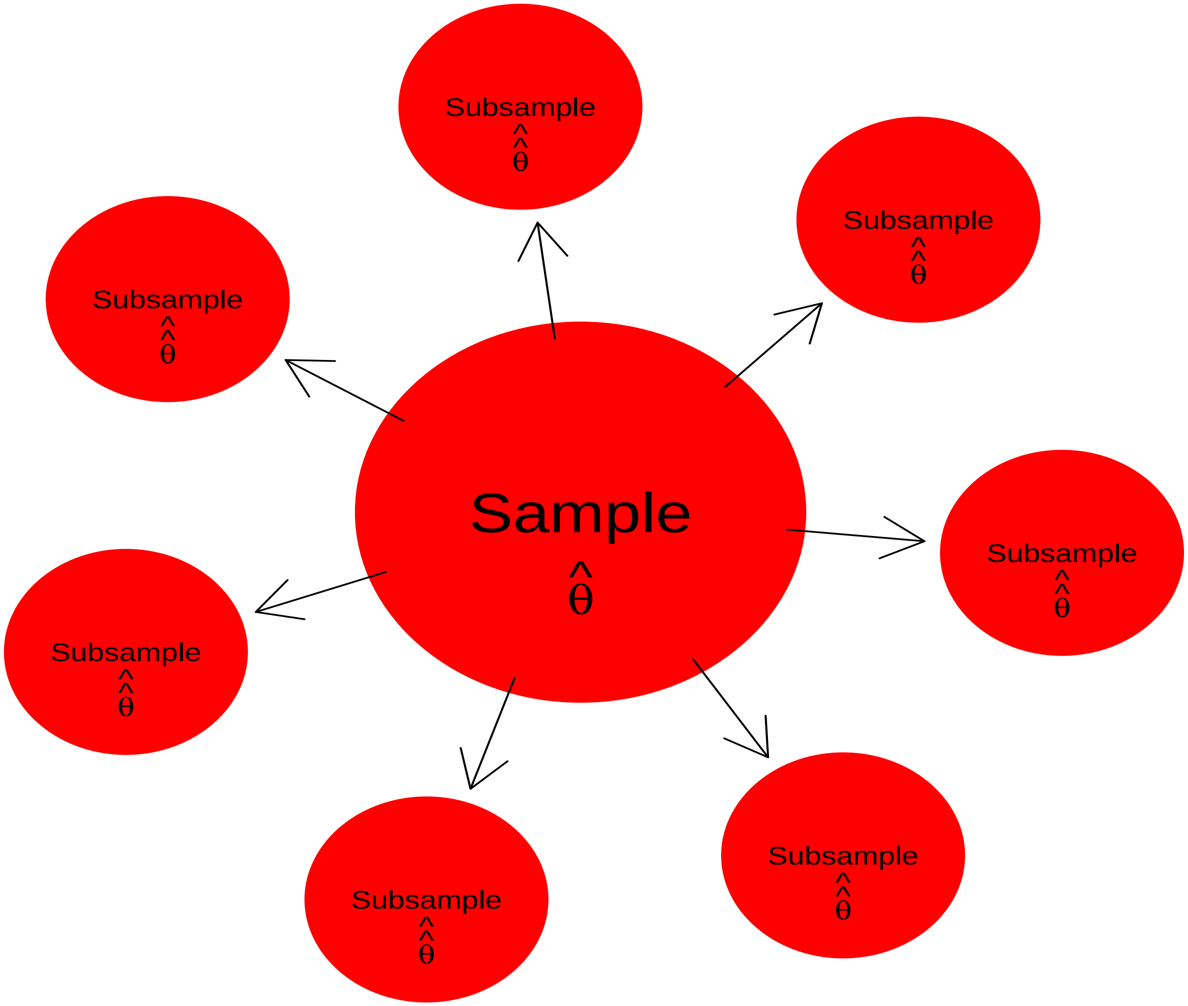
$\rightsquigarrow$ **We cannot simulate from a parametric distribution.**

However, we can use the sample itself as an estimate of the true underlying population.

In other words, by viewing the sample as a population, and simulating from that, we can assess variability of an estimate.

Introduction: Subsampling

In the real world:



Assess variability of $\widehat{\theta}$ by estimating θ for each subsample (where sample size is $m < n$).

Introduction: Subsampling

Subsample from the original sample (using $m = 5$):

```
indices <- sample(1:10, size=5, replace=FALSE)
x.sub1 <- x[indices]
x.sub2 <- x[-indices]
x.sub1

## [1] 85  3 94 92 89

x.sub2

## [1] 78 86 81 88 79
```

Introduction: Subsampling

Compute the trimmed mean for these subsamples:

```
mean(x.sub1, trim=.2)

## [1] 88.66667

mean(x.sub2, trim=.2)

## [1] 82
```

Introduction: Subsampling

Estimate the standard error by taking the standard deviation of the sample means:

```
sd(c(mean(x.sub1), mean(x.sub2)))


## [1] 6.929646
```

Introduction: Subsampling

... provides good assessments of variability in $\widehat{\theta}$

... but assumes a sample size of m

Adjust if you can. E.g.,

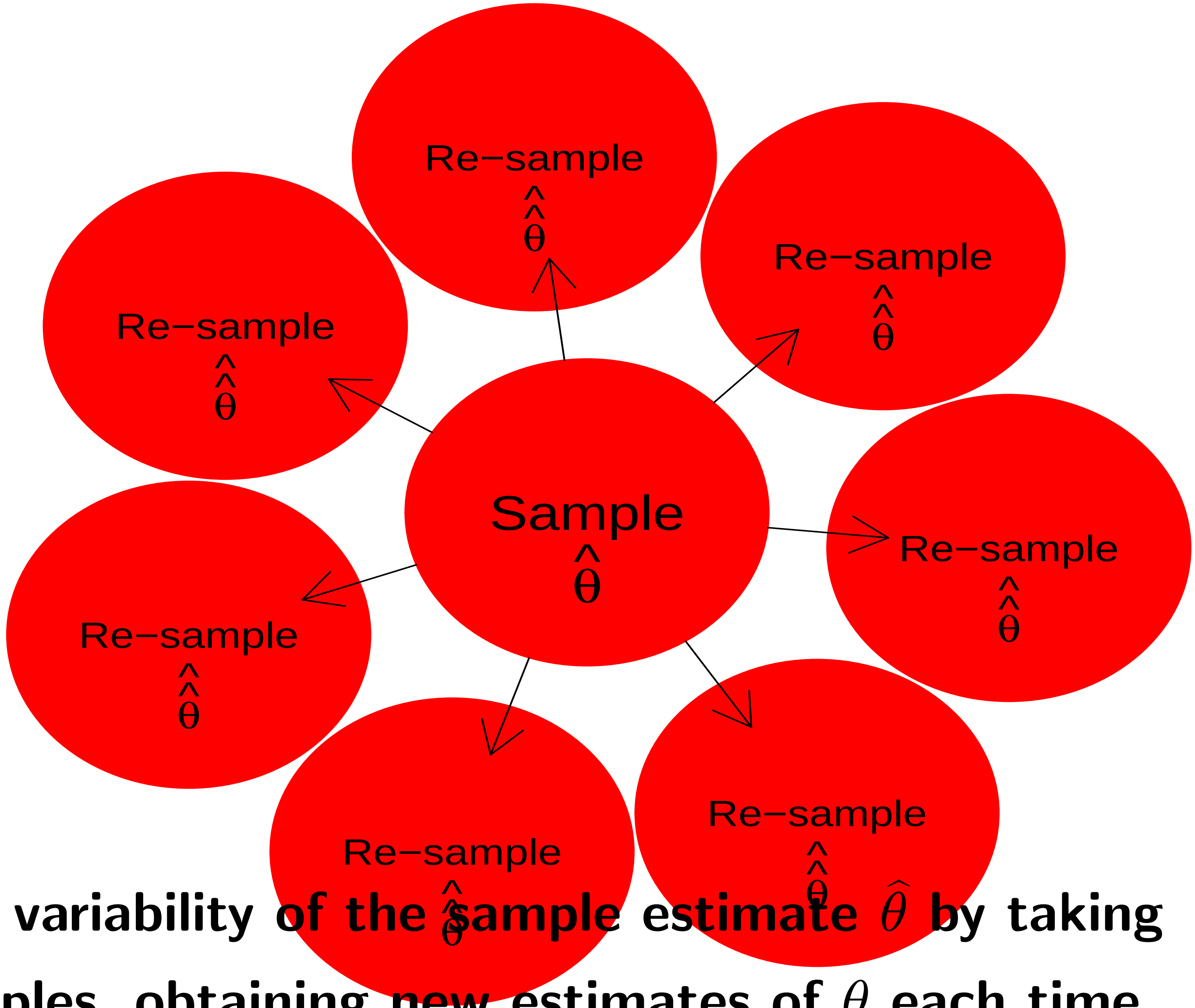
```
sd(c(mean(x.sub1), mean(x.sub2)))*sqrt(5/10)


## [1] 4.9
```

Subsampling is often available as a reliable (if inefficient) backup for the nonparametric bootstrap.

Introduction: Nonparametric bootstrap

In the bootstrap world:



Assess variability of the sample estimate $\widehat{\theta}$ by taking re-samples, obtaining new estimates of θ each time.

Introduction: Nonparametric bootstrap

Re-sample from the original sample with replacement:

```
x.nonparbs1 <- sample(x, size=10, replace=TRUE)
x.nonparbs1
```

```
##  [1] 79 94 89 85 79 86 89 86 92 94
```

Compute the trimmed mean for this resample:

```
mean(x.nonparbs1, trim=.2)
```

```
## [1] 87.83333
```

Introduction: Nonparametric bootstrap

Repeat the calculation a large number of times:

```r
x.nonparbs10000 <- replicate(10000,
  mean(sample(x, size=10, replace=TRUE), trim=.2))
sd(x.nonparbs10000)


## [1] 5.419144
```

Introduction: Nonparametric bootstrap

Problem! The outlier can appear several times in a resample, allowing it to enter the calculation of the estimate ⇝ inflation of the standard error estimate.

e.g. standard error of sample mean:

```r
xbar10000 <- replicate(10000,
  mean(sample(x, size=10, replace=TRUE)))
sd(xbar10000)    # compare with 2.17


## [1] 8.127203
```

Confidence Intervals: Normal type

1. **Estimate the standard error of $\widehat{\theta}$ using the nonparametric bootstrap: $\widehat{s}$**

2. **Interval endpoints: $\widehat{\theta} \pm \widehat{s} z_{\alpha/2}$**

where z_α is the α upper percentile of the standard normal distribution.

This technique obviously requires that the distribution of $\widehat{\theta}$ be well approximated by normality (which is often not true).

Basic Confidence Intervals

To estimate θ with a symmetric $(1-2\alpha)$ confidence interval centered at $\widehat{\theta}$, we need to find t in

$$P(\widehat{\theta} - t \leq \theta \leq \widehat{\theta} + t | \text{ population}) = 1 - 2\alpha.$$

Bootstrap principle: approximate the above equation with

$$P(\widehat{\widehat{\theta}} - t \leq \widehat{\theta} \leq \widehat{\widehat{\theta}} + t | \text{ sample}) = 1 - 2\alpha. \qquad (1)$$

The solution, $\widehat{t}$: $(1-2\alpha)$ percentile of $|\widehat{\theta} - \widehat{\widehat{\theta}}|$.

Basic confidence interval:

$$\widehat{\theta} \pm \widehat{t}.$$

Studentized Confidence Intervals

These confidence intervals are require an estimate of the standard deviation of $\widehat{\theta}$.

$$\widehat{\theta} \pm \widehat{s}\widehat{t}$$

where $\widehat{t}$ is the $1 - 2\alpha$ percentile of the distribution of

$$\frac{|\widehat{\widehat{\theta}} - \widehat{\theta}|}{\widehat{\widehat{s}}}.$$

Studentized Confidence Intervals

The standard deviation estimate $\widehat{s}$ need not be consistent.

When regularity conditions are satisfied by the data, these intervals can be much more accurate than the basic intervals. (Unfortunately, conditions are not nearly always satisfied.)

Percentile Confidence Intervals

α **and** $1 - \alpha$ **percentiles of the distribution of** $\hat{\hat{\theta}}$**.**

Originally proposed by Efron (1979)

Efron (1987) discovered a way of improving confidence interval accuracy using what he called accelerated bias correction (BCa).

BCa intervals are often more accurate than percentile intervals, but can be unstable.

Implementation using the `boot()` library (Canty, 2002)

The following data which were collected during an escape drill on an oil platform: times (in seconds) for a sample of individuals to escape.

```
escape <-
  c(389, 356, 359, 363, 375, 424, 325, 394, 402, 373,
 373, 370, 364, 366, 364, 325, 339, 393, 392, 369,
 374, 359, 356, 403, 334, 397)
```

Implementation

How much warning time is necessary to be almost certain that everyone could escape during an emergency?

A possible measure: $\theta = \mu + 4\sigma$**,**
where μ **and** σ **are the escape time mean and standard deviation. We can estimate** θ **with** $\widehat{\theta} = \bar{X} + 4s$**.**

In order to use higher order bootstrap confidence intervals, we need an estimate of the variance of $\widehat{\theta}$**.**

Implementation

In this case, we could do an exact calculation, but it is tedious. Use of an iterated bootstrap to estimate the variance is one alternative.

Another alternative is to use a delta method. If the data are normally distributed, we can show that

$$\textbf{Var}(\widehat{\theta}) \doteq \frac{\sigma^2}{n} + \frac{8\sigma^2}{n-1} - \frac{\sigma^2}{(n-1)^2}.$$

Estimate with:

$$\frac{s^2}{n} + \frac{8s^2}{n-1} - \frac{s^2}{(n-1)^2}.$$

Implementation

R Function to compute sample statistic and variance estimate:

```r
escape.fun <- function (data, indices) {
    x <- data[indices]
    xbar <- mean(x)
    s2 <- var(x)
    s <- sqrt(s2)
    n <- length(x)
    c(xbar + 4*s, s2/n + 8*s2/(n-1) - s2/(n-1)^2)
}
```

Implementation

Perform resampling and calculation of the statistics for each resample using the `boot()` function:

```r
library(boot)
escape.boot <- boot(escape, escape.fun, R=999)
```

Implementation

Output:

```
escape.boot


##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = escape, statistic = escape.fun, R = 999)
##
##
## Bootstrap Statistics :
##     original    bias    std. error
## t1* 468.1267 -2.899631    13.78719
## t2* 211.7408 -8.183151    55.70149
```

Implementation

Observations:

The bootstrap standard error of the statistic is roughly the square root of the variance estimate.

The estimate of θ is biased. This is due, largely, to the fact that s is a biased estimator for σ.

Confidence intervals can now be computed using the `boot.ci()` **function.**

Implementation

```
boot.ci(escape.boot)


## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = escape.boot)
##
## Intervals :
## Level      Normal              Basic             Studentized
## 95%   (444.0, 498.0 )   (445.0, 497.8 )   (449.2, 508.8 )
##
## Level     Percentile            BCa
## 95%   (438.4, 491.3 )   (444.0, 502.9 )
## Calculations and Intervals on Original Scale
## Some BCa intervals may be unstable
```

A GLM example

A sequence of trials were conducted using a detector.

In each trial, a number of objects were to be detected.

The observed number of detected objects were recorded for each trial.

The quality of the detector can be assessed by computing the proportion detected:

$$\widehat{p} = \frac{\#\ \text{Observed}}{\#\ \text{Expected}}$$

A normal approximation can be used to obtain the confidence interval. (Bootstrap not needed.)

A GLM example

The data:

```
detect <- read.csv("testdata.csv", header=TRUE)
head(detect)
```

```
##   Observed Expected
## 1        3        3
## 2        1        1
## 3        5        7
## 4       12       14
## 5        6        6
## 6       23       23
```

A GLM example

Fitting the binomial model using `glm()` (this is overkill, but instructive)

```
detect.glm <- glm(I(Observed/Expected) ~ ., data=detect,
     weights=Expected, family=binomial)
summary(detect.glm)$coef
```

```
##             Estimate Std. Error  z value     Pr(>|z|)
## (Intercept) 2.320604  0.1824228 12.72102 4.519306e-37
```

A GLM example

Problem! The sample of observed detections might be underdispersed relative to the binomial distribution.

If so, we could use quasi-likelihood. But this does not have the same simple interpretation.

What is value of the dispersion parameter?

```
detect.glm <- glm(I(Observed/Expected) ~ .,
    data=detect, weights=Expected,
    family=quasibinomial)
disp <- summary(detect.glm)$dispersion
disp

## [1] 0.7093484
```

This is less than 1, but is it significantly less? We can use the bootstrap to find out.

Bootstrapping the dispersion parameter

```
disp.fn <- function(data, indices) {
    Observed <- data$Observed[indices]
    Expected <- data$Expected[indices]
    bootdata <- data.frame(Observed, Expected)
    detect.glm <- glm(I(Observed/Expected) ~ .,
    data=bootdata, weights=Expected,
        family=quasibinomial)
    disp <- summary(detect.glm)$dispersion
    disp
    }

disp.boot <- boot(data=detect, statistic=disp.fn, R=999)
```

Bootstrapping the dispersion parameter

```
boot.ci(disp.boot)


## Warning in boot.ci(disp.boot):  bootstrap variances needed
for studentized intervals


## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = disp.boot)
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 0.4051,  1.0335 )   ( 0.3809,  0.9889 )
##
## Level     Percentile            BCa
## 95%   ( 0.4298,  1.0378 )   ( 0.4827,  1.2670 )
## Calculations and Intervals on Original Scale
## Some BCa intervals may be unstable
```

But do we believe the confidence intervals?

The confidence intervals do not all tell us the same thing.

BCa is supposedly the most accurate ... usually. Is it in this case?

Since it is telling us that the dispersion parameter is not significantly different from 1, let's suppose the binomial model is true and do a parametric bootstrap to see how accurate the BCa interval for dispersion is.

A model for the parametric bootstrap

Binomial model parameters:

```
p <- exp(2.32)/(1+exp(2.32))
n <- length(detect$Expected)
p; n

## [1] 0.9105199
## [1] 30
```

A small bootstrap simulation study:

Create 100 simulated binomial data sets, and construct 95% BCa confidence intervals for the dispersion parameter in each case.

A small bootstrap simulation study

```
> ncorrect <- 0
+ for (i in 1:100) {
+ detectboot <- rbinom(n, size=detect$Expected, prob=p)
+ detectboot <- data.frame(Observed=detectboot,
+      Expected=detect$Expected)
+ disp.boot2 <- boot(data=detectboot, statistic=disp.fn,
+      R=499)
+ ci <- boot.ci(disp.boot2, type="bca")$bca[4:5]
+ ncorrect <- ncorrect + (ci[1]<1)*(ci[2]>1)
+ }
+ ncorrect
[1] 94
```

Conclusion of simulation study

The BCa intervals appear to be reliable in this case.

It seems okay to believe the original data are not underdispersed.

Bootstrap failure

We now give some examples where the bootstrap is invalid. In some cases, it is possible to remedy the problem.*

*This discussion follows parts of Section 3.6 of Shao and Tu (1995).

Nonsmooth functions of the sample mean

The bootstrap is often consistent for smooth functions of the sample mean.

The smoothness condition is necessary.

E.g., consider the parameter $\theta = g(\mu) = \sqrt{|\mu|}$**.**

The statistic is $\widehat{\theta} = \sqrt{|\bar{X}|}$**.**

Nonsmooth functions of the sample mean

When $\mu = 0$,

Percentile confidence intervals for $\sqrt{|\mu|}$ never contain 0.

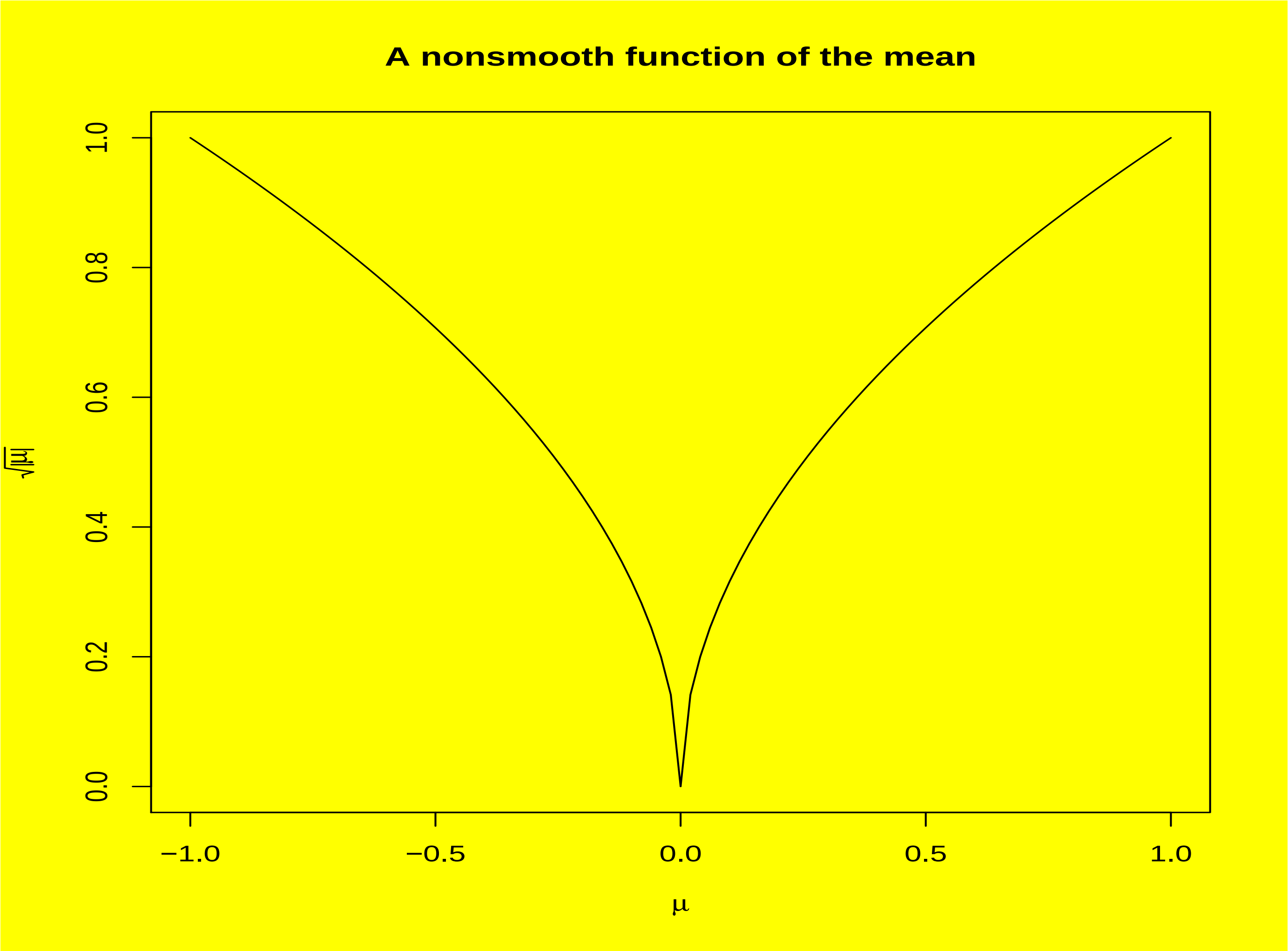
BCa confidence intervals cannot be computed.

Studentized 95% confidence intervals cannot always be computed.

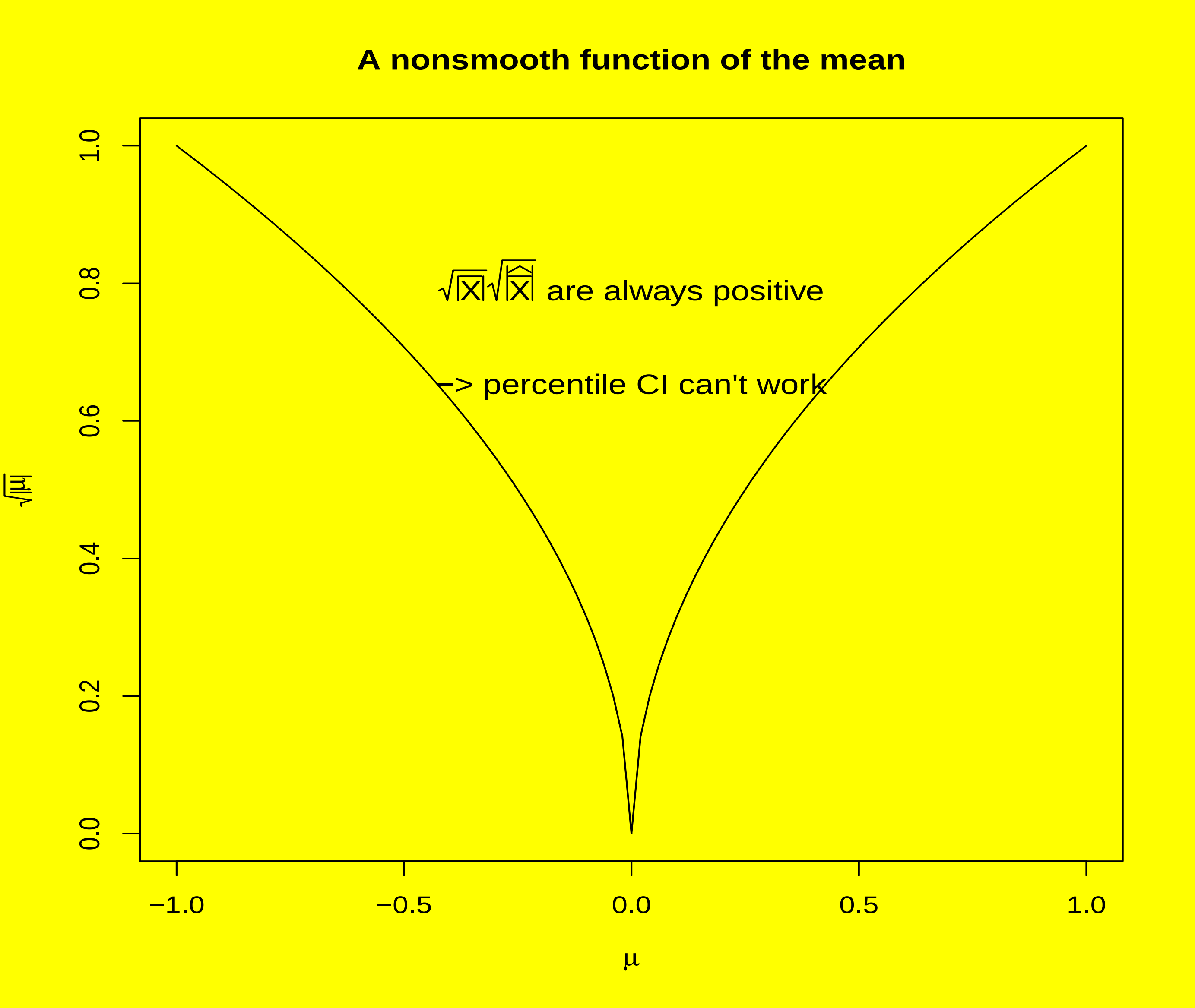
When $n = 100$, 47% of the 95% basic intervals contain 0.

Nonsmooth functions of the sample mean

What is causing the problem?

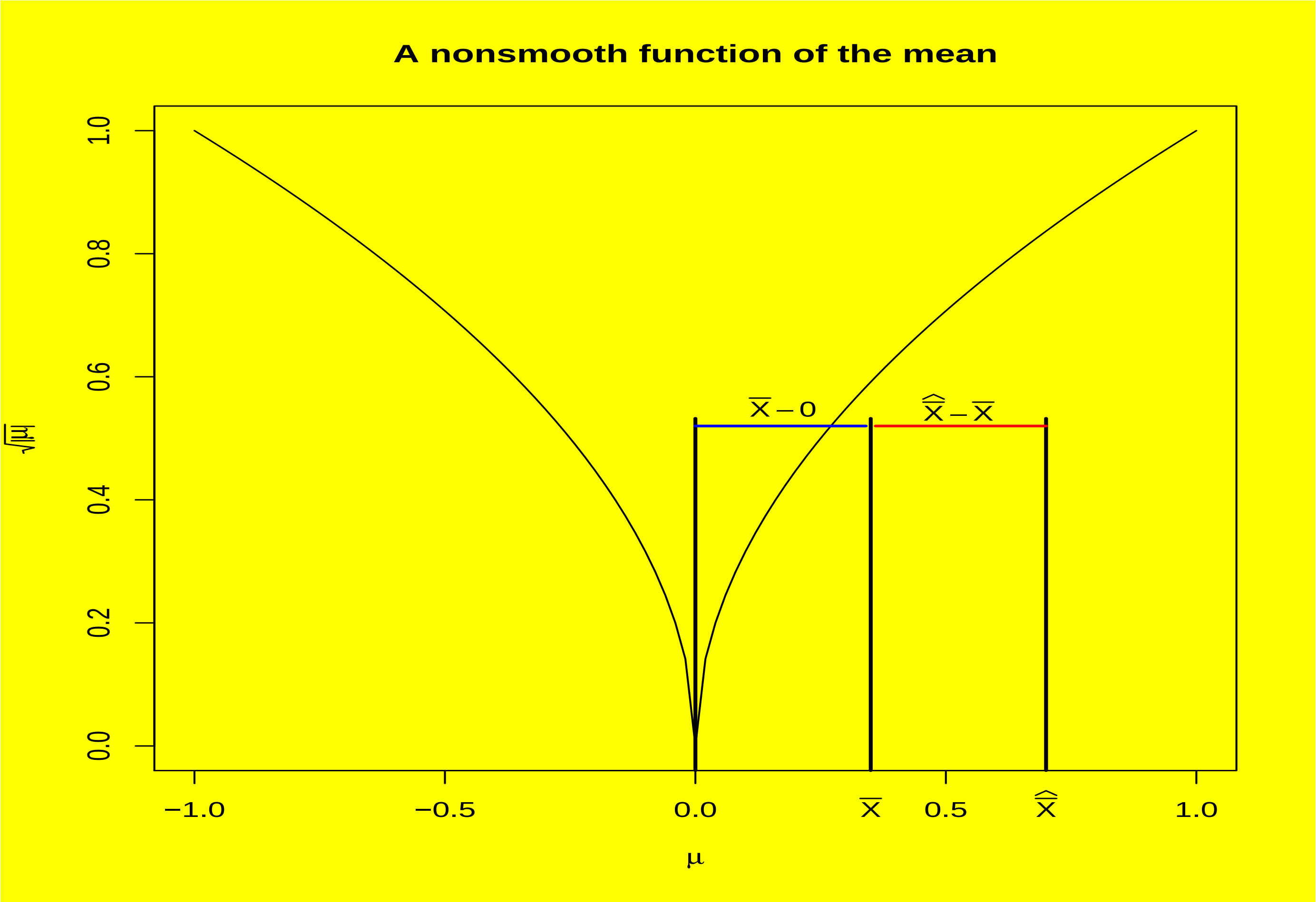


What is causing the problem? (Percentile Case)



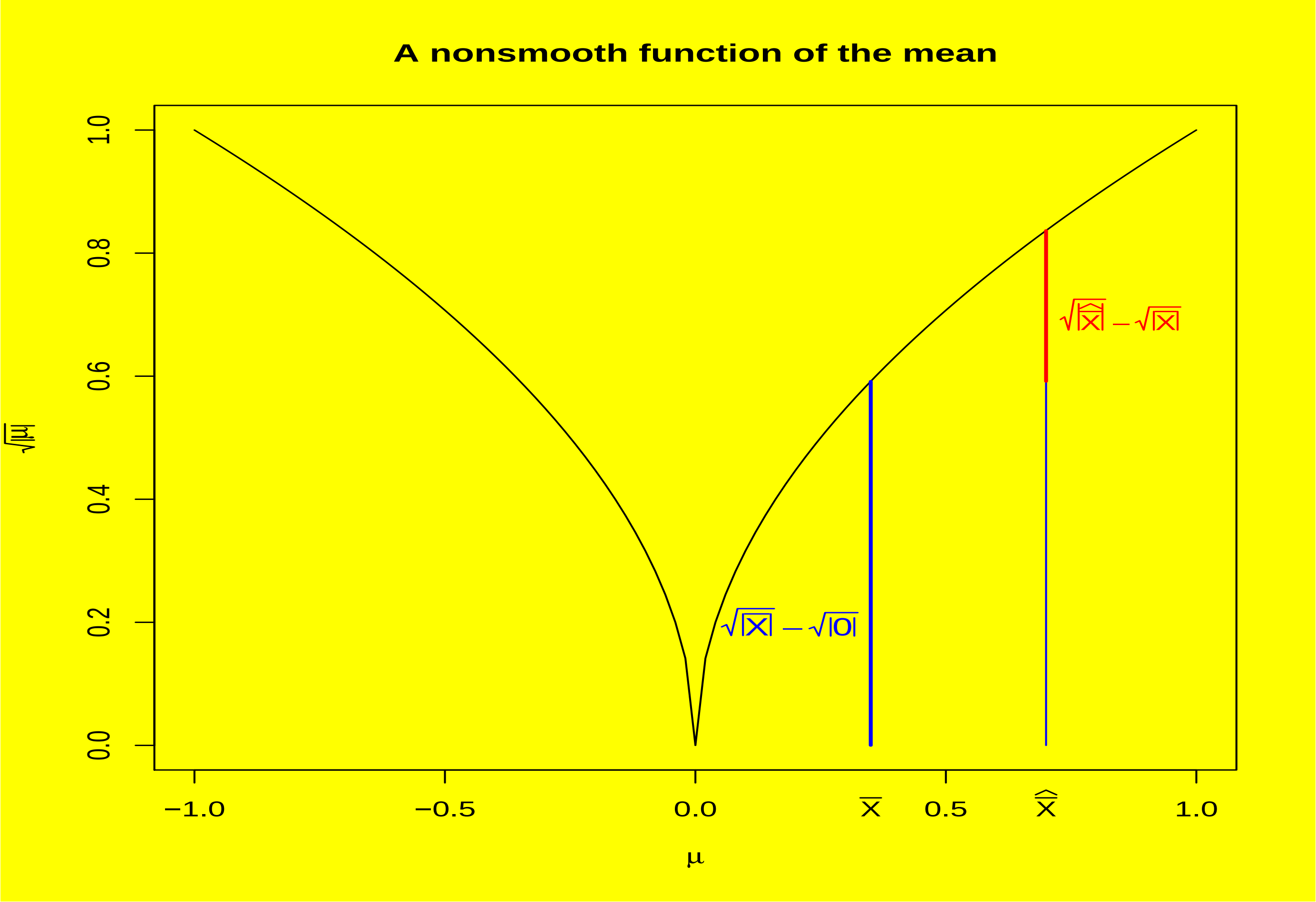
What is causing the problem? (Basic Case)

Deviations between $\bar{X}$ and μ are similar to deviations between $\widehat{\bar{X}}$ and $\bar{X}$...



What is causing the problem? (Basic Case)

... which implies that deviations between $\sqrt{|\bar{X}|}$ and $\sqrt{|\mu|}$ tend to be larger than deviations between $\sqrt{\widehat{\bar{X}}}$ and $\sqrt{|\bar{X}|}$.



What is causing the problem? (Basic Case)

$\sqrt{|\bar{X}|}$ is not converging to 0 fast enough, compared with the speed at which $\sqrt{|\widehat{\bar{X}}|}$ is converging to $\sqrt{|\bar{X}|}$.

Remedy: Increase sample size used to calculate $\sqrt{|\bar{X}|}$ to speed up convergence.

Really?!!

What is causing the problem? (Basic Case)

Actual Remedy: Slow down convergence of $\sqrt{|\widehat{\bar{X}}|}$ by reducing bootstrap sample size.

i.e. Use subsampling.

Percentage of correct basic CIs when $m = 25$: 71%

Percentage correct when $m = 15$: 85%.

Percentage correct when $m = 10$: 91%.

Functions of the mean with stationary points

Smooth functions of the mean usually pose no difficulty for the bootstrap, but not always!

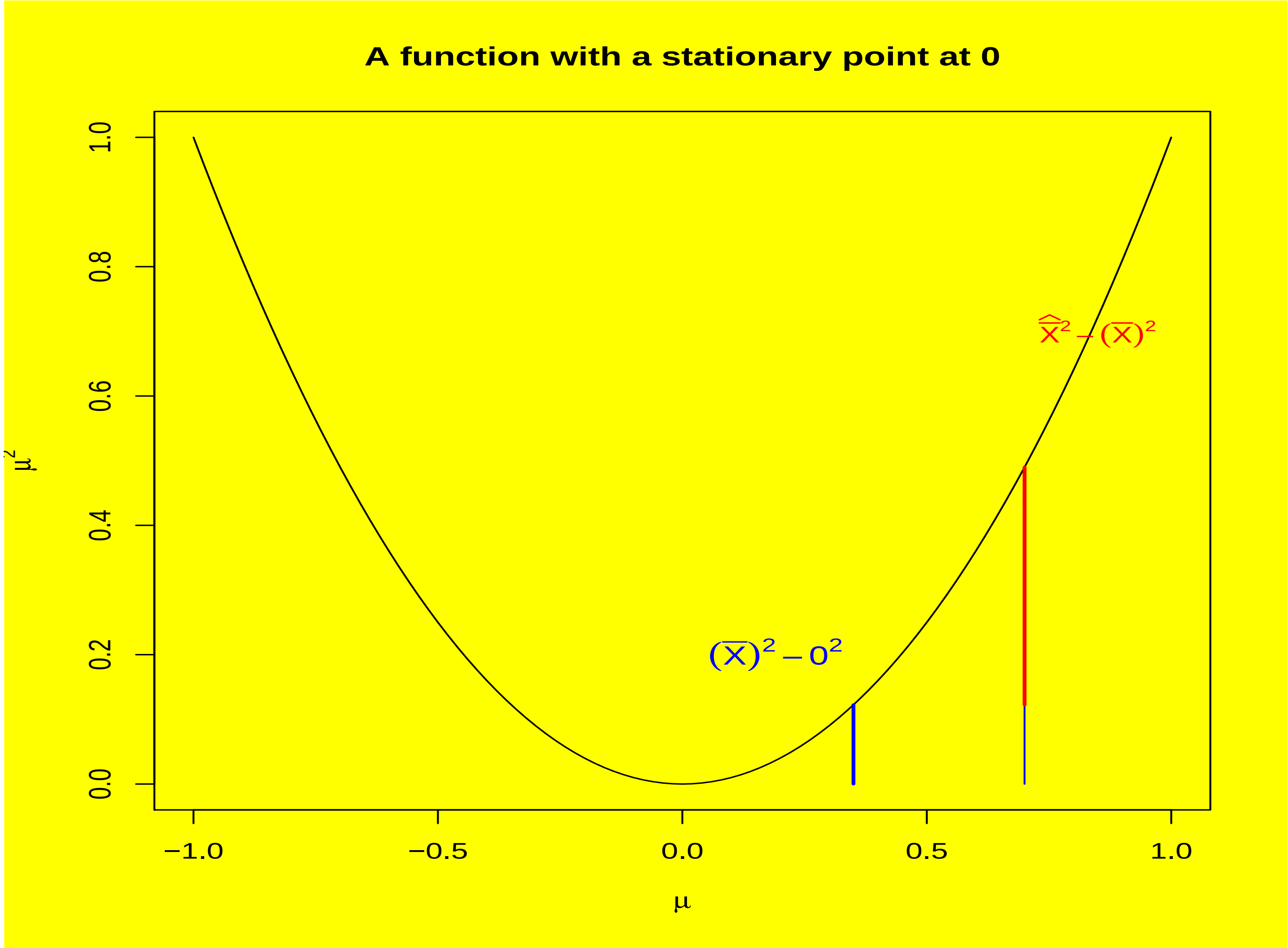
Again, suppose $X_1, X_2, \ldots, X_n$ are independent and identically distributed with mean μ and variance σ^2.

The parameter of interest is $\theta = \mu^2$ and the statistic is $\widehat{\theta} = \bar{X}^2$.

When $\mu = 0$, all percentile confidence intervals have 0% coverage, and 95% basic intervals have over 99% coverage.

Functions of the mean with stationary points

What is causing the problem?



Maximum order statistics (Knight, 1989)

Suppose $X_1, X_2, \ldots, X_n$ are independent uniform random variables on the interval $[0, \theta]$. The maximum likelihood estimator for θ is $X_{(n)}$, the maximum data point.

$$P(\theta - X_{(n)} \leq 0) = 0.$$

The bootstrap estimate:

$$P(X_{(n)} - \widehat{X}_{(n)} \leq 0) = 1 - (1 - n^{-1})^n \to 1 - e^{-1} \neq 0.$$

Once again, subsampling will alleviate this difficulty.

Dependent data

The conventional form of the bootstrap assumes independent and identically distributed data.

In the case of time series data, the ordinary bootstrap fails badly because of serial dependence.

The most popular method for handling stationary time series data is the so-called moving blocks bootstrap: resample subsequences of consecutive observerations (blocks) ...

sounds like subsampling

Summary

Bootstrapping can be used to obtain confidence intervals and to test hypotheses in situations where analytic calculations are difficult or impossible.

Studentized and BCa confidence intervals are preferred ... when they work.

The bootstrap is not robust to the influence of outliers.

The bootstrap can fail for many kinds of statistics: exercise caution when working with extremes, absolute values, ...

The bootstrap requires remediation when data are not independent.

Subsampling is often a simple, though inefficient, remedy for bootstrap failure.

Pop Quiz

Q1: Which of the 4 statistics mentioned in the preamble will the bootstrap work well on?

Q2: Which will cause trouble for the bootstrap? Why?

Further reading

1. Boos, D. (2003) Introduction to the Bootstrap World. Statistical Science 18 168–174.

2. Canty, A. Resampling Methods in R: The boot package. R News, December, 2002, 2-7.

3. Davison, A. and Hinkley, D. (1997) Bootstrap Methods and Their Application Cambridge University Press, Cambridge.

4. Efron, B. (1979) Bootstrap Methods: Another Look at the Jackknife. Annals of Statistics 7 1–26.

5. Efron, B. (1987) Better bootstrap confidence intervals. Journal of the American Statistical Association 82 171–200.

6. Efron, B. (2003) Second Thoughts on the Bootstrap. Statistical Science 18 135–140.

7. Efron, B. and Tibshirani, R. (1993) An Introduction to the Bootstrap Chapman and Hall, London.

8. Hall, P. (1992) The Bootstrap and Edgeworth Expansion Springer, New York.

9. Harrell, F. (2001) Regression Modelling Strategies Springer, New York.

10. Hjorth, J.S.U. (1994) Computer Intensive Statistical Methods Chapman and Hall, London.

11. Shao, J. and Tu, D. (1995) The Jackknife and Bootstrap Springer, New York.