moonboot: An R Package Implementing m-out-of-n Bootstrap Methods

by Christoph Dalitz and Felix Lögler

Abstract The m-out-of-n bootstrap is a possible workaround to compute confidence intervals for bootstrap inconsistent estimators, because it works under weaker conditions than the n-out-of-n bootstrap. It has the disadvantage, however, that it requires knowledge of an appropriate scaling factor tau(n) and that the coverage probability for finite n depends on the choice of m. This article presents an R package moonboot which implements the computation of m-out-of-n bootstrap confidence intervals and provides functions for estimating the parameters tau(n) and m. By means of Monte Carlo simulations, we evaluate the different methods and compare them for different estimators.

1 Introduction

The *bootstrap* is a resampling method introduced by **?**, which repeatedly simulates samples of the same size n as the observed data by n-fold drawing with replacement. Let us call this method the *n-out-of-n bootstrap*, in order to distinguish it from different sampling schemes. For each of these samples, the estimator T of interest is computed, thereby simulating a sample $T_1^*, \ldots, T_R^*$ for the distribution of T, where R is the number of bootstrap repetitions. For sufficiently smooth estimators T, this bootstrap distribution asymptotically approaches the true distribution of T (**??**), and it can thus be used to construct confidence intervals for T. The coverage probability of confidence intervals based on flipping the quantiles of the bootstrapped distribution $T_1^*, \ldots, T_R^*$ at the point estimate ("basic bootstrap") can be shown to be first order accurate, i.e. up to $O(n^{-1/2})$. Under certain conditions, it is even possible to make the confidence intervals second order accurate, i.e. up to $O(n^{-1})$, by studentized sampling ("bootstrap-t") or accelerated bias correction ("BC_a bootstrap") (**???**). Compared to other methods for estimating confidence intervals, the bootstrap is more versatile because it neither relies on a likelihood function, nor does it require asymptotic normality. The bootstrap has thus become a standard technique for estimating confidence intervals, and it is even included in vanilla R with the package **boot** (**?**), which implements five different methods in `boot.ci()`.

There are, however, *bootstrap inconsistent* estimators, i.e., estimators for which the n-out-of-n bootstrap fails to yield confidence intervals with an asymptotically correct coverage probability. Examples include extreme order statistics (**?**), the Grenander estimator of a monotonous density (**?**), Chernoff's estimator of the mode and Tukey's shorth (**?**), or Chatterjee's rank correlation index (**??**). See the discussion by **?** for a literature review and further examples. A possible workaround in these cases is the *m-out-of-n bootstrap*, which samples only $m < n$ observations. This can be done with or without replacement. **?**, who introduced this method, only considered sampling without replacement, which they called *subsampling*. They have shown that the m-out-of-n bootstrap without replacement works under much weaker conditions than the n-out-of-n bootstrap, provided m is chosen such that, for $n \to \infty$, $m \to \infty$ and $m/n \to 0$. The only condition is that the estimator, suitably scaled by some factor τ_n, possesses a limit distribution, whereas no smoothness of the estimator or uniformity of the convergence are required. If the sampling is instead done with replacement, **?** have shown that a similar result requires additional restrictions on the estimator. Sampling without replacement thus works under weaker conditions than sampling with replacement, and, in the present article, we therefore only consider sampling without replacement. Our package **moonboot**[1] supports sampling both with and without replacement by means of an argument `replace`, which is set to `FALSE` by default.

[1] This article used version 2.0.x, available, e.g., from `https://github.com/cdalitz/moonboot`.

The wider applicability of the m-out-of-n bootstrap comes at a price, though. One drawback is that the scaling factor τ_n must be known. **?** explained that it must be chosen such that $\tau_n^2 \operatorname{Var}(T)$ converges to some constant, and they also suggested a method to estimate the scaling factor with another bootstrap. For root-n consistent estimators, it is $\tau_n = \sqrt{n}$, but not all estimators are root-n consistent, especially if they are bootstrap inconsistent, and even if root-n consistency is conjectured it can be difficult to prove (**?**). Moreover, the m-out-of-n bootstrap can have less than first order accuracy: **?** gave a theoretical example with accuracy $O(n^{-1/3})$ for an optimal choice of m, and even worse for other choices of m. That the accuracy depends on the choice of m was demonstrated, too, in a simulation study by **?**. And for the sample quantiles and sampling with replacement, **?** has shown that the choice $m \propto n^{2/3}$ is optimal. This raises the problem of how to choose m, and several heuristics have been suggested in the literature (**?????**).

Due to these shortcomings, the usual n-out-of-n bootstrap is generally preferable for bootstrap consistent estimators. However, there *are* bootstrap inconsistent estimators and there is thus need for an R package that facilitates the application of the m-out-of-n bootstrap by providing the required functions. To this end, we present a new package **moonboot**, evaluate the algorithms implemented therein by means of Monte Carlo simulations, and give recommendations and use cases for their application. The name of the package was inspired by an article by **?**, who abbreviated the m-out-of-n bootstrap as "moon bootstrap".

2 The m-out-of-n bootstrap

Let $X_1, \ldots, X_n$ be i.i.d. random variables, θ be some parameter of their underlying distribution, and $T_n = T_n(X_1, \ldots, X_n)$ be an estimator for θ. Let us additionally assume[2] that $E(T_n^2) < \infty$. The m-out-of-n bootstrap requires that, for some scaling factor τ_n which behaves for $n \to \infty$ as

$$\frac{\tau_{m(n)}}{\tau_n} \to 0 \quad \text{for} \quad m(n) \to \infty \quad \text{with} \quad \frac{m(n)}{n} \to 0, \tag{1}$$

the estimator converges in distribution to some limit law when centered around the true parameter θ and scaled with τ_n:

$$S_n = \tau_n(T_n - \theta) \overset{n\to\infty}{\longrightarrow} S \text{ in distribution} \tag{2}$$

Note that condition (**??**) implies that $\tau_n \to \infty$, and thus the convergence (**??**) requires that T_n converges to θ in probability (in other words: T_n must be consistent), because it follows from (**??**) that

$$P(|T_n - \theta| > \varepsilon) = F_n(-\tau_n\varepsilon) + (1 - F_n(\tau_n\varepsilon)) \overset{n\to\infty}{\longrightarrow} 0$$

where F_n is the cumulative distribution function of S_n. Moreover, if additionally the second moment of S_n converges to some constant, the scaling factor τ_n is related to the rate of decrease of the variance of T_n:

$$V = \lim_{n\to\infty} \operatorname{Var}(S_n) = \lim_{n\to\infty} \operatorname{Var}(\tau_n T_n) \quad \Rightarrow \quad \operatorname{Var}(T_n) \sim \frac{V}{\tau_n^2} \quad \text{for} \quad n \to \infty \tag{3}$$

This relationship allows one to determine τ_n theoretically by means of an analytical calculation, or to estimate it by means of Monte Carlo simulations.

? have shown that, under the assumptions (**??**) and (**??**), a confidence interval with asymptotic coverage probability $(1 - \alpha)$ can be constructed as

$$\left[T_n - \frac{q(1-\alpha/2)}{\tau_n},\ T_n - \frac{q(\alpha/2)}{\tau_n}\right] \tag{4}$$

[2] **?** did not make this assumption, but it is necessary for the estimation of the scaling factor τ_n with the method by **?**.

where $q(1-\alpha/2)$ and $q(\alpha/2)$ are the quantiles of the scaled bootstrap distribution $\tau_m(T^*_m - T_n)$, where T^*_m denotes the bootstrap samples obtained by m-fold drawing without replacement. In **moonboot**, this interval is implemented by `mboot.ci(...,method="politis")`.

If the variance of S_n converges to some value σ^2 and the limiting distribution of S is the *normal distribution* with standard deviation σ, then the m-out-of-n bootstrap can alternatively be used to estimate the variance σ^2 and compute the standard confidence interval

$$T_n \pm z_{1-\alpha/2}\,\hat{\sigma}^*_m \quad \text{with} \quad \hat{\sigma}^*_m = \frac{\tau_m}{\tau_n}\sqrt{\mathrm{Var}(T^*_m)} \tag{5}$$

In **moonboot**, this is implemented by `mboot.ci(...,method="norm")`. Although this interval is only reasonable for asymptotically normal estimators, its coverage probability can have a faster convergence to the nominal value than the interval (**??**) provided asymptotic normality indeed holds (**?**).

? observed that the scaling with τ_m/τ_n in (**??**) can be omitted if the interval is centered around one of the T_m instead of T_n. They suggested using the first m samples for computing T_m, but this is an arbitrary choice and thus even the location of the confidence interval is not uniquely determined by the data. Moreover, the lack of scaling increases the interval length. The greater length is compensated by the higher volatility of the location so that the coverage probability still is close to the nominal value. This does not hold, however, for drawing without replacement as m approaches n, and a compromise must be made between small interval length and approximate coverage probability. **?** suggested a heuristic method for choosing m on the basis of a double bootstrap, and they gave the rule of thumb to choose m between $n^{1/2}$ and $n^{2/3}$. However, due to the greater interval length, this method should only be used as a last resort if the convergence rate τ_n is unknown and cannot be estimated from the data. In **moonboot**, this method is implemented by `mboot.ci(...,method="sherman")` and the method for estimating m by `estimate.m.sherman()`.

2.1 Estimation of τ_n

According to Eq. (**??**), the formula for the scaling factor τ_n can be determined by an analytic investigation of $\mathrm{Var}(T_n)$, which can not only be difficult for some estimators, but it requires an ad hoc study of the specific estimator T_n under consideration. This thwarts the application of the m-out-of-n bootstrap out-of-the-box.

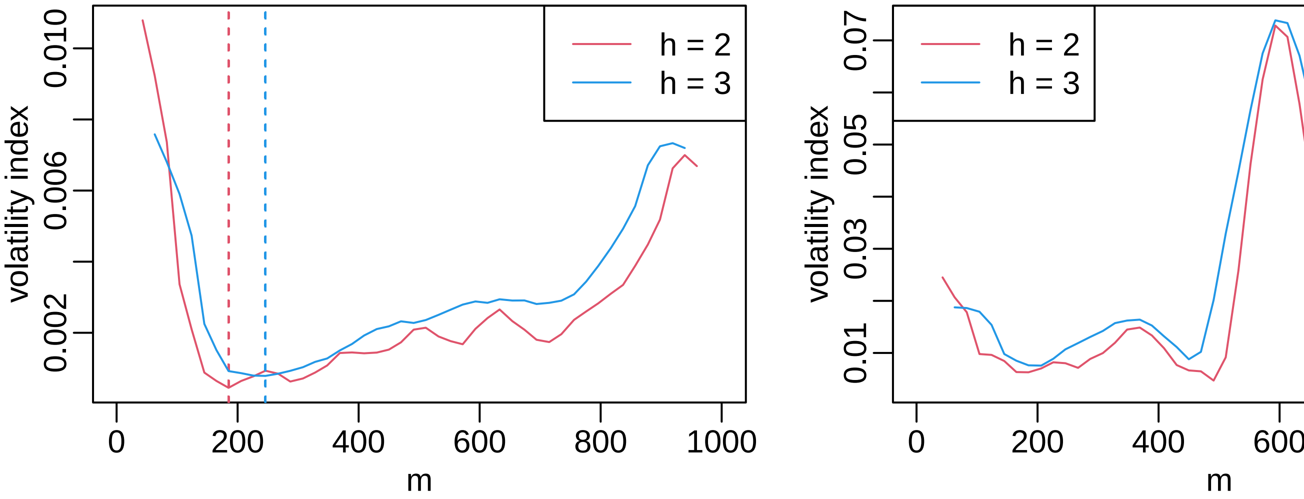
Fortunately, the relationship (**??**) makes it possible to estimate τ_n by another bootstrap (**?**). If the variance $V_m = \mathrm{Var}(T^*_m)$ is estimated by sampling with different subsampling sizes m and τ_n is assumed to be of the form $\tau_n = n^\beta$, then the asymptotic relationship (**??**) becomes

$$\log V_m \approx -2\beta \log m + \log V \tag{6}$$

and β can be estimated with a least-squares fit. For the choice of the test values m, **?** suggested $m_i = n^{\gamma_i}$, but wrote that "the difficult problem of choosing the γ_i's requires more work on a case-by-case basis"[3]. This casts doubt on the whole method, because removing the necessity of a detailed asymptotic analysis of the estimator is the whole point of estimating τ_n from the data. Indeed, we have not found a sequence of values for the γ_i that worked equally well in all of our examples. We thus leave the choice of the γ-sequence as an option to the user and use the default of five values between 0.2 and 0.7, which worked reasonably well for some of our estimators.

? suggested the alternative of using quantiles instead of the variance, but as the method involves taking logarithms, this makes no sense when the estimator tends to be negative or close to zero. As a workaround, (**?**) suggested replacing V_m in Eq. (**??**) with the average of

[3] Note that **?** wrote n^{β_i} for the test values m_i, but we have renamed them to n^{γ_i} to avoid confusion with the power β that is to be estimated.



1. xicor

2. shorth

Figure 1: Volatility index after **?** as a function of m for Chatterjee's correlation (*xicor*) and Tukey's shorth for $n = 1000$.

quantile ranges

$$Q_m = \frac{1}{J}\sum_{j=1}^{J}\left(\text{quantile}(T_m^*, \alpha_j^{high}) - \text{quantile}(T_m^*, \alpha_j^{low})\right) \tag{7}$$

We have implemented this, too, with the choice $\alpha_j^{high} = 0.75 + j \cdot 0.05$ and $\alpha_j^{low} = 0.25 - j \cdot 0.05$ for $j = 0, \ldots, 4$, but the results in section **??** show that this makes almost no difference in comparison to using the variance.

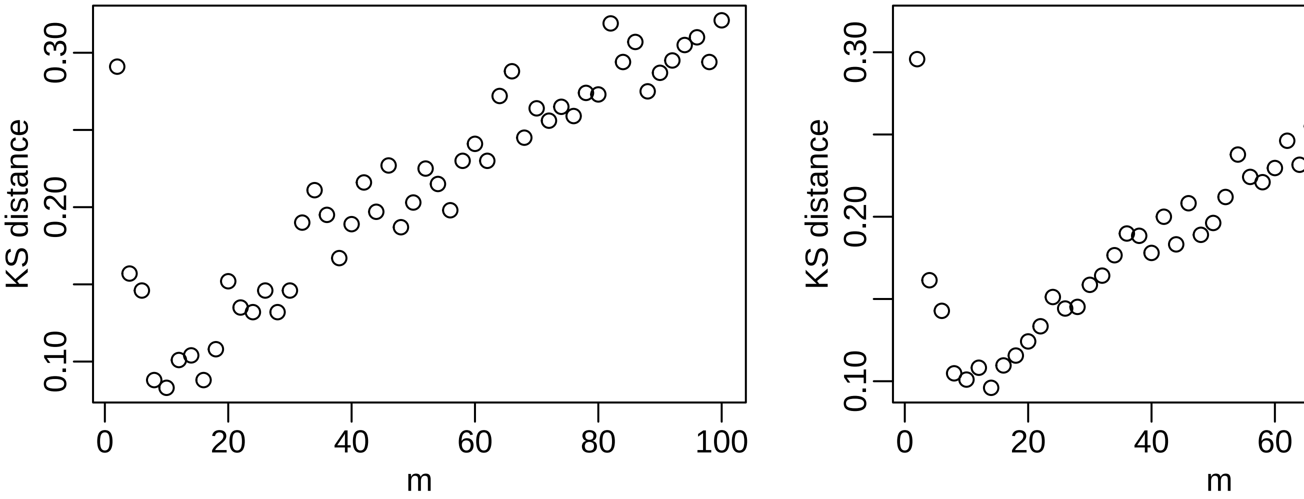
This method is implemented in **moonboot** by `estimate.tau()`, which is automatically called by `mboot.ci()` when no rate for the parameter `tau` is provided.

2.2 Choice of m

According to **?**, the m-out-of-n bootstrap asymptotically works for any choice of m satisfying $m \to \infty$ and $m/n \to 0$ as $n \to \infty$. This, however, leaves a wide range of choices, e.g. $m = cn^\beta$ for any $\beta \in (0, 1)$, and the simulations in section **??** show that the convergence rate of the coverage probability depends on the choice of m. It might thus be desirable to choose m in a data-dependent way, for which different methods have been suggested in the literature (**?????**).

?, ch. 9.3.2 suggested a *minimum volatility method* for estimating m, which is based on the idea that there should be some range for m where its choice has little effect on the estimated confidence interval endpoints. The optimal m is that with the lowest "volatility" of the confidence interval, which is defined as the running standard deviation of the interval endpoints around each specific choice for m. Before computing the standard deviation, the interval endpoints should be smoothed out, too, by averaging over the endpoints computed for the neighboring choices for m.

The algorithm has two parameters, the window width h_{ci} for smoothing the interval end points and the window width h_σ for computing the running standard deviation. **?** recommended $h_{ci} = h_\sigma = 2$ or 3, which corresponds to a range of 5 or 7 values. To avoid finding a minimum that is too close to n, as happens in Fig. **??**b, we have restricted the search range to values $m < n/2$. For efficiency reasons, **?** recommended not to try every m, but only a grid of equidistant values. Our implementation in the function `estimate.m.volatility()` in the package **moonboot** therefore only tries out a maximum of 50 values for m.



1. single sample

2. averaged over multiple sa

Figure 2: Kolmogorov distance between the CDFs of S^*_m and $S^*_{m/2}$ for the estimator of the maximum of a uniform distribution with $n = 200$. The left results are for a single sampling per m, and the right results for averaging over fivefold sampling.

The methods by **?** and **?** are both based on minimizing the distance between the distributions of S^*_m and $S^*_{m'}$ for different choices m and m' of the subsampling size. As a distance measure, both articles utilized the Kolmogorov distance, i.e., the maximum distance between the cumulative distribution functions. The methods differ in the values that are tried out for m and m': **?** suggested searching the minimum among all values for m and $m/2$, whereas **?** searched the minimum distance between m_j and m_{j+1} for $m_j = \lceil q^j n \rceil$, with the recommendation to set $q = 0.75$. These methods are provided by **moonboot** as `estimate.m(...,method="goetze")` and `estimate.m(...,method="bickel")`, respectively.

Trying out all combinations of m and $m/2$, as suggested by **?**, has a runtime of order $O(Rn^2)$, and it would thus be preferable to use a more efficient minimum search like, e.g., a Golden Section Search (**?**). Such algorithms, however, are devised for deterministic functions, not for random functions. As can be seen in Fig. **??**, the Kolmogorov distance $d(m, m/2)$ between two randomly sampled distributions scatters considerably and this is only somewhat remedied by repeated sampling. This means that even the usual initial bracketing process by going downhill with increasing step size is unreliable when sampling from specific data, even if, on average, the function actually has a non-boundary minimum[4]. We therefore stuck to the original suggestion by **?** and have implemented an exhaustive search over all even values for m.

Two other methods have been suggested in the literature, which we have not included in **moonboot**. **?** based their method on the assumptions of root-n consistency and asymptotic normality of the estimator. Although the assumption of root-n consistency might be overcome by replacing factors $n^{1/2}$ with τ_n in the formulas, the normality assumption is essential for that method because it is based on an Edgeworth expansion. In our tests with non-normal estimators, the coverage probabilities were very low and did not even converge to the nominal value for large n. **?** suggested using a double bootstrap for estimating the coverage probability for different values of m and to choose the m where this estimate is close to the nominal value. In our experiments, however, the estimated coverage probability based on an in-sample bootstrap was either way too low or almost one and thus rarely close to the nominal value. For *xicor*, this behavior was not even monotone in m, which made the method difficult to apply. Moreover, it has a runtime complexity of $O(R^2n^2)$, which made Monte Carlo simulations for its evaluation infeasible.

[4] Fig. 2 in the article by **?** shows that even this is not guaranteed, and the method can thus fail.

2.3 Functions provided by *moonboot*

Like in the **boot** package, the computation of the confidence interval is split into two steps: the sampling procedure for simulating the bootstrap distribution T_m^*, and the computation of the confidence interval therefrom. Using the same interface makes it easy for a user to switch between **boot** and **moonboot**. Moreover, this interface is very flexible because it allows for statistics defined on complicated data structures for which only the user knows how to do the sub-indexing. The interface consists of the following two functions:

```
mboot(data, statistic, m, R=1000, replace=FALSE, ...)
```

Simulates the bootstrap distribution of the given estimator *statistic*, which must have been defined as a function with two arguments: `statistic(data,indices)`. For multidimensional data, each row in *data* is assumed to be one data point. Additional arguments are passed to *statistic*.

```
mboot.ci(boot, conf=0.95, tau=NULL, type=c("all", "basic", "norm"))
```

Estimates the confidence interval with Eqs. (**??**) ("basic") or (**??**) ("norm"). *tau* must be a function that computes τ_n from its argument n. If it is not provided, it is estimated with `estimate.tau()` with the default settings of this function. According to the results in Section **??**, this is not recommended, though, and it is preferable to provide the correct value for *tau*.

Calling these functions requires knowledge of the scaling factor τ_n and a choice for the subsample size m. For the cases that τ_n is not known or that m shall be chosen in a data-based way, two other functions are provided:

```
estimate.tau(data, statistic, R=1000, replace=FALSE, min.m=3,           gamma=seq(0.2, 0.7, leng
```

Estimates the scaling factor with the method by **?**. The values for m are tried out as $m_i = n^{\gamma_i}$. *min.m* is the minimum sample size for which the statistic makes sense and can be computed. The estimation is based on the scaling behavior of the variance (`method="variance"`) or of the quantile ranges (`method="quantile"`).

```
estimate.m(data, statistic, tau, R=1000, replace=FALSE, min.m=3,          method="bickel", params
```

Estimates m with the method by **?** (`method="bickel"`), the method by **?** (`method="goetze"`), or with the volatility index (`method="politis"`). *params* is a list that can be used to pass additional parameters for the underlying method, e.g. `params=list(q=0.75)` for the method by **?**. *min.m* is the minimum sample size for which the statistic makes sense and can be computed.

2.4 Typical usage

Let us assume that you want to estimate a confidence interval for an estimator `my.stat` on the basis of some data `x`. Then you must first define a wrapper function that computes `my.stat` only for the data selected according to the provided indices:

```
boot.stat <- function(dat, indices) {
  my.stat(dat[indices])
}
```

Without any knowledge about the asymptotic properties of `my.stat`, you can estimate a 95% confidence interval with:

```
boot.out <- mboot(x, boot.stat, m=sqrt(NROW(x)))
ci <- mboot.ci(boot.out, type="basic")
print(ci)
```

estimator $\hat{\theta}$	*data model*	*true* θ	τ_n
`max(x)`	$x_i \sim \mathrm{unif}(0,1)$	1	n
`moonboot::shorth(x)`	$x_i \sim \mathrm{norm}(0,1)$	0	$n^{1/3}$
`XICOR::xicor(x,y)`	$x_i \sim \mathrm{unif}(-1,1), y_i \sim x_i + \mathcal{N}(0,0.5)$	0.3818147	$\sqrt{n}$
`mean(x)`	$x_i \sim \mathrm{power}(2,0,1)$	3/4	$\sqrt{n}$

Table 1: Tested estimators and data generation processes. $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are the simulated data sets.

Beware that this automatically estimates the asymptotic convergence rate $1/\tau_n^2$ of the variance of `my.stat` with the method by **?**. It is thus better to provide this rate in the parameter `tau`, if you happen to know it. For a root-n consistent estimator, e.g., it is $\tau_n = \sqrt{n}$ and you can call `mboot.ci()` as follows[5]:

```
ci <- mboot.ci(boot.out, tau=function(n) { n^0.5 }, type="basic")
```

3 Examples

In order to evaluate the implemented methods, we applied them to four different estimators, three of which were bootstrap inconsistent. A summary can be found in Tbl. **??**, and the detailed description follows. $x_1, \ldots, x_n$ denote the i.i.d. sample values drawn from the given distributions.

Maximum of a uniform distribution (max). The maximum likelihood estimator for the upper bound of a uniform distribution between zero and θ is $\hat{\theta} = \max\{x_1, \ldots, x_n\}$. This estimator was already given by **?** as an example of a bootstrap inconsistent estimator. It has the nice property that its probability density $g(t)$ can be readily computed as

$$g(t) = \begin{cases} nt^{n-1}\theta^{-n} & \text{for } 0 \leq t \leq \theta \\ 0 & \text{else} \end{cases} \tag{8}$$

This estimator is neither root-n consistent, nor asymptotically normal. The scaling factor τ_n can be computed from Eq. (**??**) as

$$\mathrm{Var}(\hat{\theta}) = \theta^2 \frac{n}{(n+2)(n+1)^2} \overset{n\to\infty}{\sim} \frac{\theta^2}{n^2} \quad \Rightarrow \quad \tau_n = n \tag{9}$$

Tukey's Shorth (shorth). This is the mean of the data points in the shortest interval that contains half of the data. For symmetric distributions with a strongly unimodal density, **?**, p. 50ff has shown that this estimator is cube root consistent, i.e., $\tau_n = n^{1/3}$. Bootstrap inconsistency of the shorth was mentioned by **?**. We have implemented this estimator in **moonboot** as `shorth()` and applied it to normally distributed data.

Chatterjee's rank correlation (xicor). This coefficient ξ_n was introduced by **?** as an estimator for an index $\xi(X,Y)$ that is zero, if the two random variables X and Y are independent, and one, if Y is a measurable function of X. For continuous X and Y, this is an interesting example of a root-n consistent estimator (**?**) that is bootstrap inconsistent (**??**). We have used the implementation `xicor()` from the package XICOR (**?**) to compute ξ_n, and the function given by **?** to compute $\xi(X,Y)$ for the model $X \sim Y + \varepsilon$, where ε is normally distributed with zero mean and $\sigma = 0.5$. Note that we have used the original definition of ξ_n by **?**, not the bias reduced form suggested by **?**.

[5] It is more efficient to just set `tau=sqrt`, of course, but the example uses a more complicated way for the sake of clarity.

method	*estimator*	*true β*	(γ_1, γ_5)	*n*	*estimated β*	*n*	*estimated β*
variance	mean	0.5	$(\mathbf{0.2}, \mathbf{0.5})$	100	0.4723	500	0.4870
			$(0.4, 0.8)$		0.5964		0.5480
	max	1.0	$(0.2, 0.5)$		0.6789		0.7730
			$(\mathbf{0.4}, \mathbf{0.8})$		0.9595		1.0043
	xicor	0.5	$(0.2, 0.5)$		0.1821		0.2945
			$(\mathbf{0.4}, \mathbf{0.8})$		0.4914		0.5052
quantile range	mean	0.5	$(\mathbf{0.2}, \mathbf{0.5})$	100	0.5394	500	0.5164
			$(0.4, 0.8)$		0.6167		0.5600
	max	1.0	$(0.2, 0.5)$		0.8355		0.8722
			$(\mathbf{0.4}, \mathbf{0.8})$		1.1068		1.0872
	xicor	0.5	$(0.2, 0.5)$		0.0332		0.2128
			$(\mathbf{0.4}, \mathbf{0.8})$		0.4880		0.5171

Table 2: Dependency of the estimation of $\tau_n = n^\beta$ on the chosen endpoints for the sequence $m_i = n^{\gamma_i}$ and two different values for n. The better choice is marked in bold for each estimator. The estimated values for the power β in τ_n have been averaged over 100 estimations.

Mean of an unsymmetric distribution (mean). The mean is a very well-behaved statistic: it is asymptotically normal, root-n consistent and bootstrap consistent. It can thus serve as a simple test case of how well the m-out-of-n bootstrap performs for bootstrap consistent estimators. We have simulated data according to the density $f(x) = 3x^2$ for $x \in [0, 1]$, which was already used in a study by **?**, and for which we have implemented the random number generator `rpower()` in **moonboot**.

4 Results

For all four estimators, we have evaluated the methods for estimating τ_n and the different choices for m in the m-out-of-n bootstrap without replacement by means of Monte Carlo simulations. We have computed confidence intervals with a nominal coverage probability P_{cov} of 0.95 and estimated the actual coverage probability by repeating the computation $N = 10^4$ times. This means that the accuracy (i.e. the width of a 95% confidence interval) of the estimated P_{cov} is about ± 0.005. Using a larger N would have made the simulation for the method by Goetze intractable. All simulations were done with $R = 1000$ bootstrap repetitions.

4.1 Estimation of τ_n

Our simulations confirmed that the choice for the extreme values of the trial values $m_i = n^{\gamma_i}$ for $i = 1, \ldots, 5$ indeed influence the accuracy of the estimation of τ_n. For the *mean* estimator, e.g., we achieved the best results for $\gamma_1 = 0.2$ and $\gamma_5 = 0.5$, whereas this choice was poor for *xicor* and *max* (see Tbl. **??**). The results for estimation by means of the variance or quantile ranges are similar. We thus conclude that it does not matter which method is used. The results for *xicor*, however, are a warning that the estimation of τ_n with the method by **?** can be grossly inaccurate, unless the range of the γ_i has been luckily guessed.

4.2 Choice of m

To see how the coverage probability depends on the choice of m, we have first used the fixed formula $m = n^\beta$ with different choices for β. The results in Fig. **??** show that the coverage probability indeed depends on the choice of m. Moreover, the optimal choice for β depends on the estimator: $\beta = 1/3$ was, e.g., a decent choice for *shorth*, but a poor choice for *xicor*.

The results for the data based methods for choosing m are summarized in Fig. **??**. The method by **?** cannot be recommended, because the resulting coverage probability never approached the nominal value in our simulations. For the *shorth*, it was only about 0.7 even for high values of n. For the other estimators, it was somewhat greater, but nevertheless too small. This behavior can be understood from the curves showing $m(n)$ in Fig. **??**: The method by **?** always chooses an m proportional to n, which means that the condition $\lim_{n\to\infty} m/n = 0$ is violated.

The methods according to **?** and **?**, on the other hand, both showed for all estimators a similar coverage probability which approached the nominal value with increasing n. The method by **?** was slightly better for three of the four tested estimators. It should be noted, though, that there was considerable fluctuation in the estimated values for m, as can be concluded from the wide error bars in Fig. **??**. As the method by **?** has a much higher runtime of order $O(Rn^2)$, the method by **?**, which has a runtime of order $O(Rn)$, is preferable from a runtime perspective, too.

Out of curiosity, we have also compared the m-out-of-n bootstrap without replacement with the ordinary n-out-of-n bootstrap for the estimator *mean*, which is bootstrap consistent and thus allows for a comparison. For bootstrap consistent estimators with additional smoothness properties[6], it was generally proven by **?** that estimates based on the m-out-of-n bootstrap with or without replacement are less efficient than those based on the n-out-of-n bootstrap. The mean is an example of such an estimator. As can be seen in Fig. **??**, the m-out-of-n bootstrap is competitive with the basic bootstrap only for large $n \gtrsim 2000$ and a luckily chosen m. For a data based choice of m with the method by **?**, the coverage probability is smaller, although the difference in the interval length is small. This again demonstrates that, for bootstrap consistent estimators, the usual n-out-of-n bootstrap is preferable, especially because the best choice for m is not known, in general.

5 Violation of the assumptions

The assumptions (**??**) & (**??**), under which the m-out-of-n bootstrap works, are quite weak, but there are nevertheless some estimators that violate these conditions. For example, inconsistent estimators, i.e., estimators that do not converge in probability to the true parameter value, violate the assumptions because consistency is a necessary condition for the assumptions to hold, as shown in section **??**. Using **moonboot** in such a case is an error on the side of the user, and if the user has determined the scaling factor τ_n by an analysis of the estimator, he usually will be aware of this. It might be, however, that a user relies on the fully automated estimation of τ_n provided by **moonboot** and thus is not aware of a violation of the assumptions.

We therefore have tested our package with two unbiased, yet inconsistent estimators, too. Please note that these estimators are bizarre examples and are not meant to be used in practice. The first estimator is only the very first observation as an estimator for the mean μ of the unsymmetric distribution `moonboot::dpower(..,2,0,1)`:

$$\hat{\mu}_1 = X_1 \tag{10}$$

The distribution of this estimator is the same for all sample sizes, and the scaling factor is thus $\tau_n \propto 1$, which fulfills condition (**??**), but violates condition (**??**). The second estimator estimates the parameter λ of a Poisson distribution. For a Poisson distribution, both mean and variance are λ, which allows for the construction of an unbiased estimator for λ from the unbiased estimators $\overline{X}$ for the mean and S^2 for the variance as follows:

$$\hat{\lambda}_n = n\overline{X} - (n-1)S^2 = \sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2 \tag{11}$$

As both $\overline{X}$ and S^2 are root-n consistent estimators, the scaling factor for condition (**??**) to hold is $\tau_n \propto n^{-1/2}$, which again is in violation of condition (**??**).

As the assumptions (**??**) & (**??**) are only sufficient, but not necessary conditions for the m-out-of-n bootstrap, we first estimated the coverage probability of the "basic" m-out-of-n

[6] Most notably that it is twice Fréchet differentiable.

bootstrap interval when the correct scaling factor τ_n is provided. For $\hat{\mu}_1$, the coverage probability obviously neither depends on n nor on the subsampling size m, and it turned out to be about 0.52 for a nominal 0.95 interval. For $\hat{\lambda}_n$, the coverage probability decreased with increasing n and was only about 0.17 for $n = 5000$ and $m = \sqrt{n}$. These results show that the m-out-of-n bootstrap indeed does not work for these inconsistent estimators.

We then estimated the scaling factors with `estimate.tau()` for $n = 100$ and $n = 500$, and, for all parameter settings, the mean estimates for β in $\tau_n = n^\beta$ were less than 0.002 for $\hat{\mu}_1$ and less than -0.40 for $\hat{\lambda}_n$. This means that, for these estimators, the violation of the assumptions can be automatically detected due to suspiciously small estimated values for β. In order to make the user aware of this issue, `mboot.ci()` gives a warning if τ_n decreases or if its increase is suspiciously small, i.e., if it increases at a slower rate than $n^{0.01}$.

6 Conclusions

The **moonboot** package provides ready-to-run implementations of the m-out-of-n bootstrap and methods for estimating its parameters. Our simulations have shown that the quality of the data based method for estimating the scaling factor τ_n is highly sensitive to a parameter for which no universal recommendations can be made. It is thus recommended to estimate τ_n by a different method, e.g., by model based Monte Carlo simulations of the variance of the estimator and estimating it by means of Eq. (**??**). For a data based choice of m, the method by **?** as implemented in `estimate.m(...,method="bickel")` can be recommended, according to our simulations.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and for the suggestion to investigate the case when the assumptions are violated.

References

D. F. Andrews, P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey. *Robust Estimators of Location*. Princeton University Press, Princeton, USA, 1972. [p131]

M. A. Arcones. On the asymptotic accuracy of the bootstrap under arbitrary resampling size. *Annals of the Institute of Statistical Mathematics*, 55:563–583, 2003. [p126]

P. Bertail, D. N. Politis, and J. P. Romano. On subsampling estimators with unknown rate of convergence. *Journal of the American Statistical Association*, 94(446):569–579, 1999. [p126, 127, 130, 131, 133]

P. J. Bickel and A. Sakov. On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statistica Sinica*, 18(3):967–985, 2008. [p126, 128, 129, 130, 133, 134]

P. J. Bickel, F. Götze, and W. R. van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. *Statistica Sinica*, 7(1):1–31, 1997. [p125, 131, 133]

A. Canty and B. D. Ripley. *boot: Bootstrap R (S-Plus) Functions*, 2021. R package version 1.3-28. [p125]

S. Chatterjee. A new coefficient of correlation. *Journal of the American Statistical Association*, 116(536):2009–2022, 2021. doi: 10.1080/01621459.2020.1758115. [p132]

S. Chatterjee and S. Holmes. *XICOR: Robust and generalized correlation coefficients*, 2023. R package. https://CRAN.R-project.org/package=XICOR. [p132]

K.-H. Chung and S. M. Lee. Optimal bootstrap sample size in construction of percentile confidence bounds. *Scandinavian Journal of Statistics*, 28(1):225–239, 2001. [p126, 128, 129]

C. Dalitz. Construction of confidence intervals. Technical Report 2017-01, Hochschule Niederrhein, Fachbereich Elektrotechnik und Informatik, 2017. [p132]

C. Dalitz, J. Arning, and S. Goebbels. A simple bias reduction for Chatterjee's correlation. *Journal of Statistical Theory and Practice*, 18:51, 2024. doi: 10.1007/s42519-024-00399-y. [p125, 127, 132]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, 1997. [p125]

T. J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Statistical Science*, 11(3):189–228, 1996. [p125]

B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7(1):1–26, 1979. [p125]

E. Giné and J. Zinn. Bootstrapping general empirical measures. *The Annals of Probability*, pages 851–869, 1990. [p125]

F. Götze and A. Račkauskas. Adaptive choice of bootstrap sample sizes. *Lecture Notes-Monograph Series*, 36 (State of the Art in Probability and Statistics):286–309, 2001. [p126, 128, 129, 130, 133]

P. Hall. Theoretical comparison of bootstrap confidence intervals. *The Annals of Statistics*, 16 (3):927–953, 1988. [p125]

A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(4):795–816, 2014. [p126]

S. M. Lee and P. Yang. Bootstrap confidence regions based on M-estimators under nonstandard conditions. *The Annals of Statistics*, 48(1):274–299, 2020. [p126, 128, 129]

C. Léger and B. MacGibbon. On the bootstrap in cube root asymptotics. *Canadian Journal of Statistics*, 34(1):29–44, 2006. [p125, 131]

Z. Lin and F. Han. Limit theorems of Chatterjee's rank correlation, 2022. Preprint. `https://arxiv.org/abs/2204.08031`. [p132]

Z. Lin and F. Han. On the failure of the bootstrap for Chatterjee's rank correlation. *Biometrika*, page asae004, 2024. doi: 10.1093/biomet/asae004. [p125, 126, 132]

D. N. Politis and J. P. Romano. Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of Statistics*, 22(4):2031–2050, 1994. doi: 10.1214/aos/1176325770. [p125, 126, 128]

D. N. Politis, J. P. Romano, and M. Wolf. *Subsampling*. Springer, New York, 1999. [p126, 128, 129, 133]

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, USA, 2nd edition, 1992. [p129]

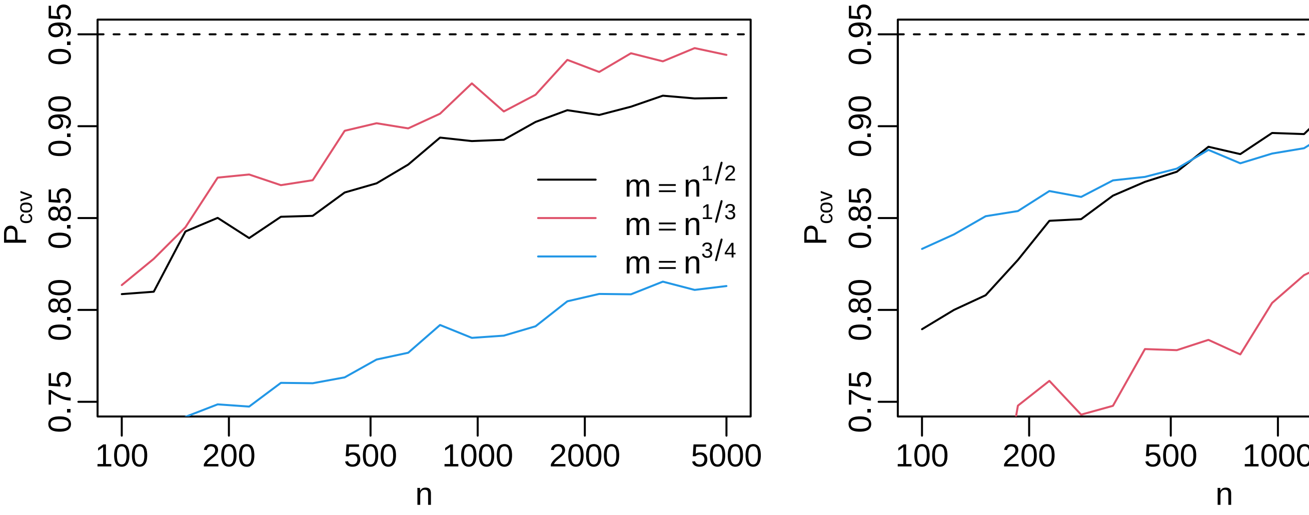
B. Sen, M. Banerjee, and M. Woodroofe. Inconsistency of bootstrap: The Grenander estimator. *The Annals of Statistics*, 38(4):1953–1977, 2010. [p125]

J. Shao and D. Tu. *The jackknife and bootstrap*. Springer, 1995. [p125]

M. Sherman and E. Carlstein. Confidence intervals based on estimators with unknown rates of convergence. *Computational Statistics & Data Analysis*, 46(1):123–139, 2004. [p127]

Christoph Dalitz
Niederrhein University of Applied Sciences
Institute for Pattern Recognition
Reinarzstr. 49
Germany
(ORCiD 0000-0002-7004-5584)
`christoph.dalitz@hs-niederrhein.de`

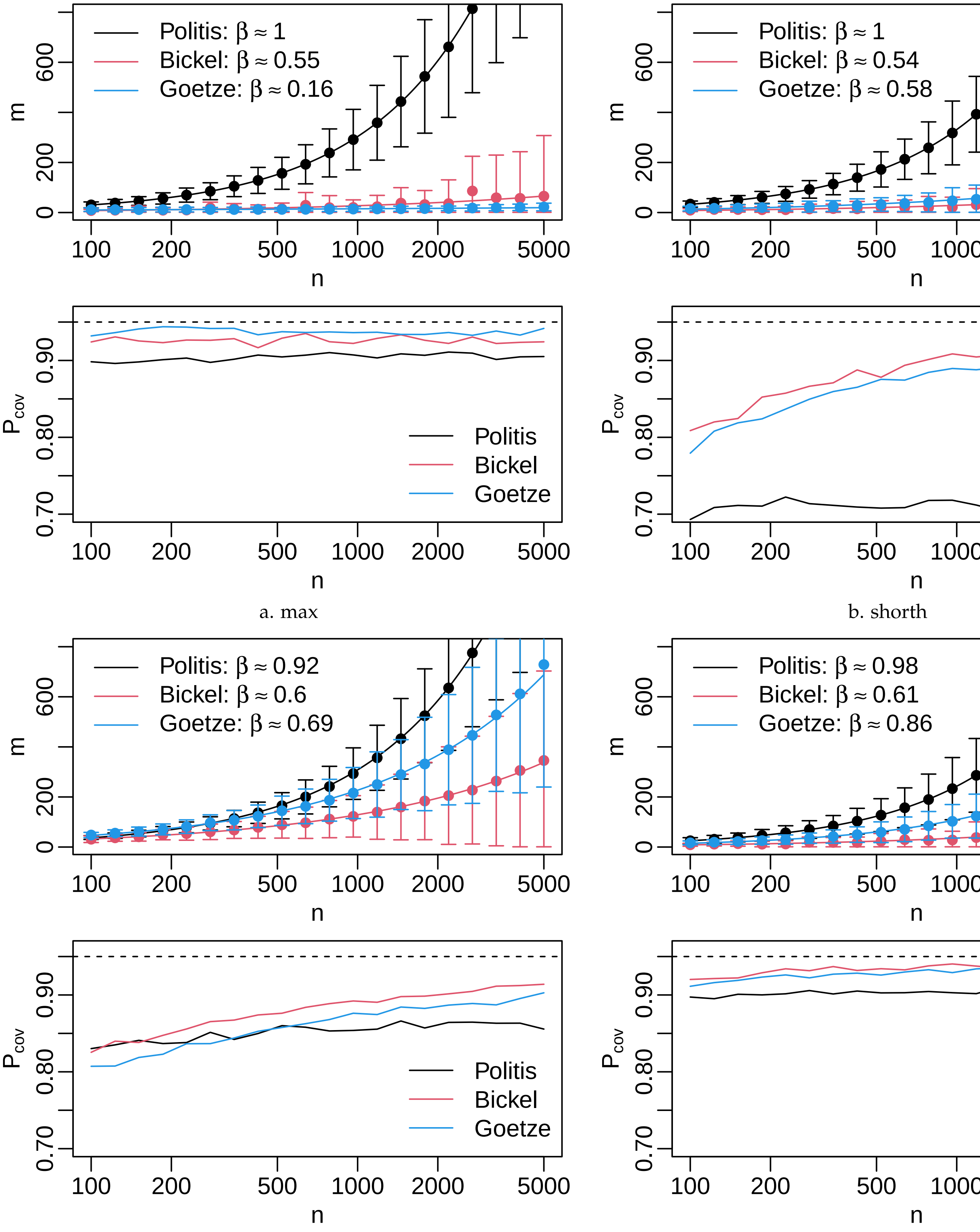
Felix Lögler
Niederrhein University of Applied Sciences
Institute for Pattern Recognition
Reinarzstr. 49
Germany



a. shorth

b. xicor

Figure 3: Coverage probability P_{cov} of the m-out-of-n bootstrap without replacement for the choices $m = n^{\beta}$ with $\beta \in \{\frac{1}{2}, \frac{1}{3}, \frac{3}{4}\}$. Note that the n-axis uses a logarithmic scale.



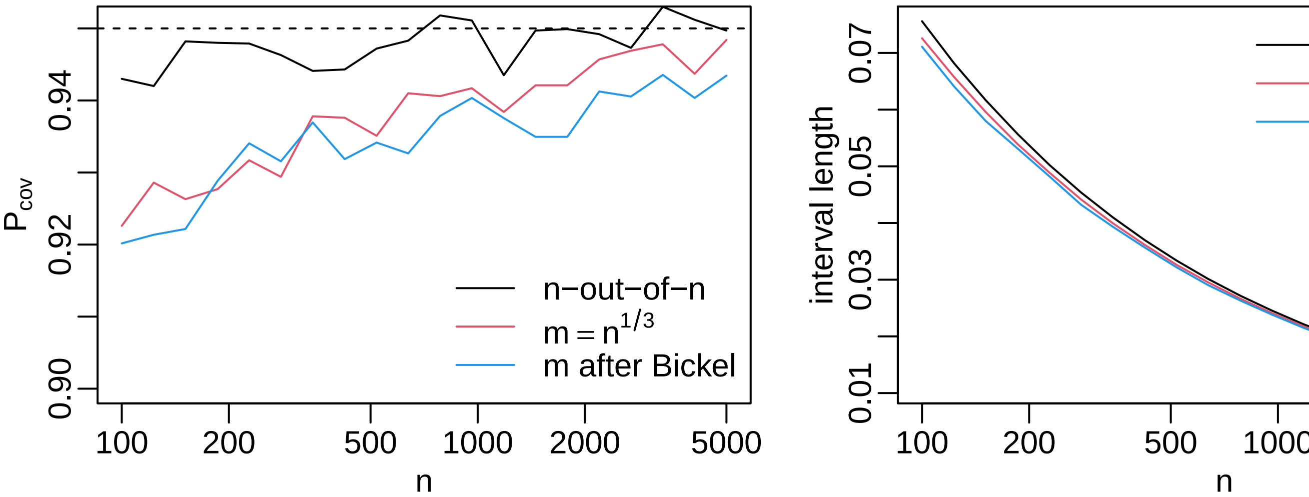
a. max

b. shorth

c. xicor

d. mean

Figure 4: Comparison of the three data based methods for choosing m in the m-out-of-n bootstrap without replacement. The upper plots show the mean value $m(n)$ together with error bars $\pm\,\mathrm{sd}(m)$ and a least squares fit $\log(m) = \log(cn^{\beta})$. The bottom plots show the resulting coverage probabilities. Note that the n-axis uses a logarithmic scale.



a. coverage probability

b. average interval leng

Figure 5: Comparison of the m-out-of-n bootstrap without replacement with the ordinary n-out-of-n (basic) bootstrap for the bootstrap consistent estimator *mean*. Note that the n-axis uses a logarithmic scale.