Bootstrap Methods Using Simulated Data

jdt

12/28/2020

Contents

Bootstrap Procedure

Some Theory

Derive the standard error using numerical methods, such as, Bootstrapping. This topic will be covered in much greater detail in an advanced computational statistics course.

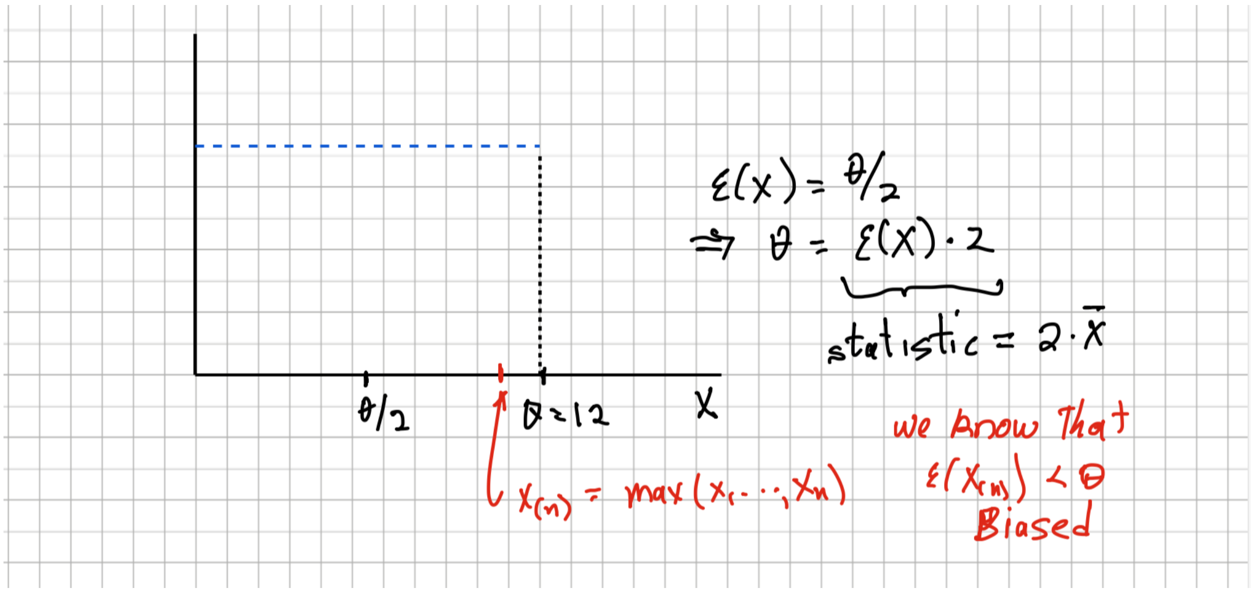
Suppose that one has a realization of a simple random sample of size n, given by $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$. A single bootstrap sample given by, $\mathbf{x}_b^* = (x_1^*, x_2^*, \ldots, x_n^*)$, is a sample of size n taken from the above realization when the sampling is done **with replacement**. Suppose that T_n is any statistic. The standard error of T_n can be computed as:

1. Select B (large number) independent bootstrap samples $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_B^*$, where $\mathbf{x}_b^* = (x_1^*, x_2^*, \ldots, x_n^*)$.
2. Compute the statistic $T_n = T_n^{*b}$ for each of the $b = 1, 2, \ldots, B$ bootstrap samples.
3. Estimate the bootstrap standard error by

$$s.e._{boot}(T_n) = \left\{ \frac{1}{B-1} \sum_{b=1}^{B} \left(T_N^{*b} - \bar{T_n^*} \right)^2 \right\}^{1/2}$$

   where $\bar{T_n^*} = \frac{1}{B} \sum_{b=1}^{B} T_N^{*b}$.

In the next section, we will consider several statistics for the problem $x \sim U(0, \theta)$ when $\theta > 0$. Consider the following figure.



R

Simulation for Order statistic from a U(0,θ)

```
if (!require("boot")) install.packages("boot", dep=TRUE)
```

```
## Loading required package: boot
```

```
library("boot")
```

Generate data

```
set.seed(123)
theta = 12  # parameter for the uniform (0, theta)
dat = c(runif(100)*theta)
```

Define functions for the bootstrap.

Define function using the statistics $2 * (\bar{x})$

```
fc_mean <- function(d, i){
    d2 <- d[i]
    return(2*mean(d2))
}
```

Perform Bootstrap

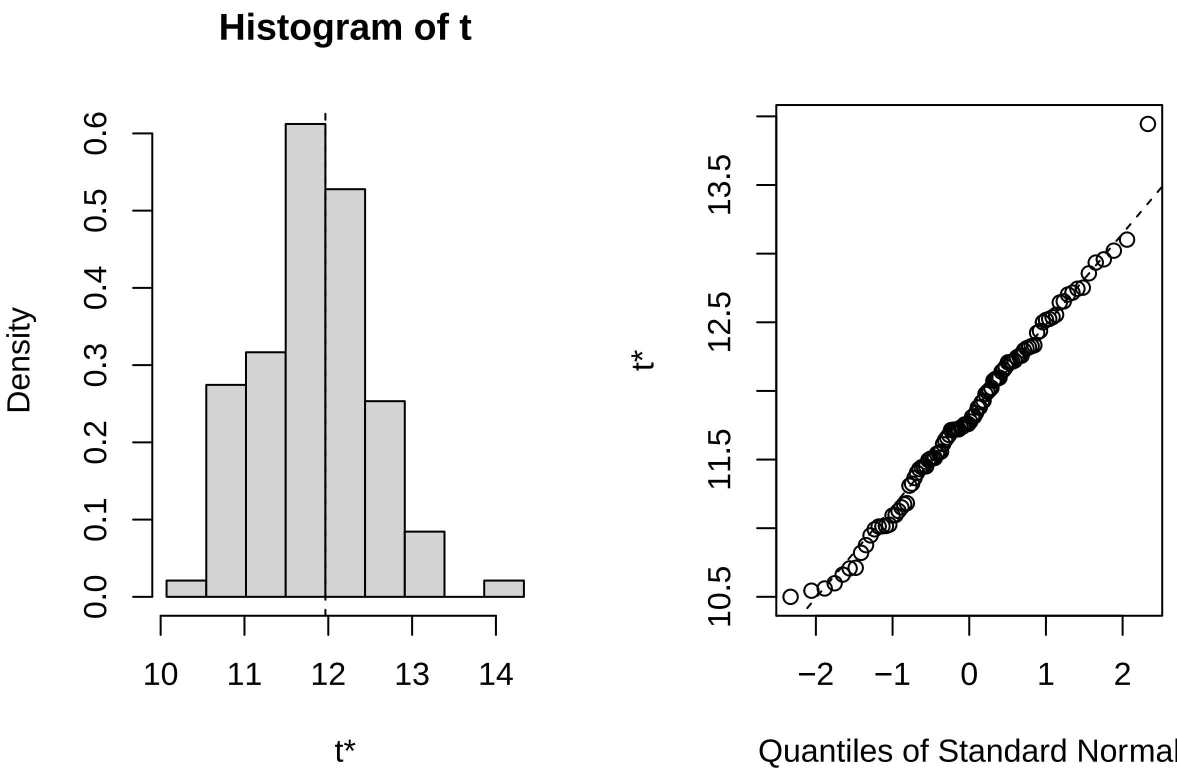
```
set.seed(321)
b.mean = boot(dat, fc_mean, R=100)
b.mean
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = dat, statistic = fc_mean, R = 100)
##
```

```
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 11.96542 -0.1457932   0.6634344
```

Notice the magnitude of the std. error

```
plot(b.mean)
```



Define function using the maximum $x_{(n)}$

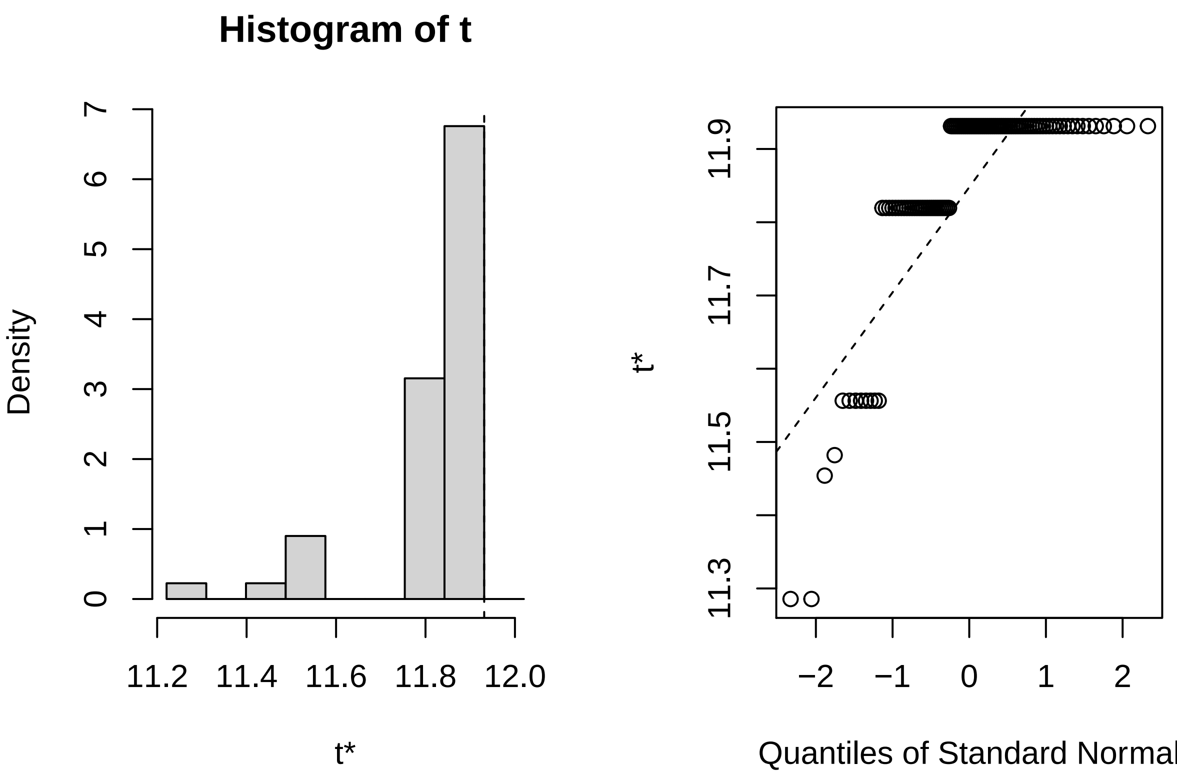
```
fc_max <- function(d, i){
    d2 <- d[i]
    return(max(d2))
}
```

Perform Bootstrap

```
set.seed(321)  #same bootstrap sample as with the mean
b.max = boot(dat, fc_max, R=100)
b.max
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = dat, statistic = fc_max, R = 100)
##
##
## Bootstrap Statistics :
##     original       bias    std. error
## t1* 11.93124 -0.08346362    0.1435816
```

```
plot(b.max)
```



SAS

Code

```
options center nodate pagesize=100 ls=70;
*libname LDATA '/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/';

/* Simplified LaTeX output that uses plain LaTeX tables *
ods tagsets.simplelatex
file="/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/simdata_boot.tex"
stylesheet="/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/sas.sty"(url="sas");
*/
title1 'Simulate Data for BootStrap Example';
title2 'Generate Uniform (0, theta) Data';

/*********************************************************************
 Simulation by Using the DATA Step and SAS Procedures
 ********************************************************************/

%let theta=12;            /* Right endpoint      */
%let N = 30;              /* size of each sample */
%let NumSamples = 250;    /* number of samples   */

/* 1. Simulate data */
data SimUni;
call streaminit(123);
   do i = 1 to &N;
      x = &theta*rand("Uniform");
      output;
```

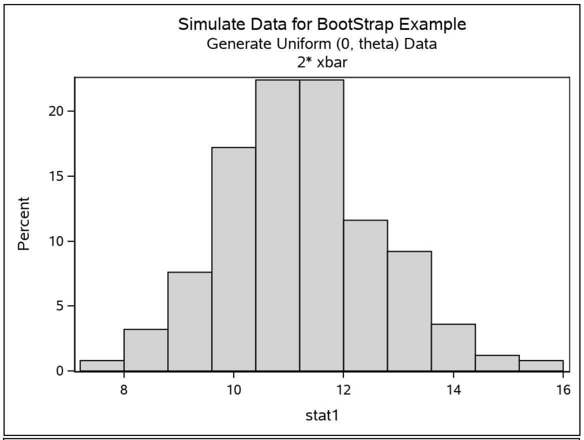
```
   end;
run;

*******************************************************;
* Using PROC SURVEYSELECT;
*******************************************************;

proc surveyselect data=SimUni NOPRINT seed=1
     out=BootSS
     method=balboot
     reps=&NumSamples;
run;

proc summary data=BootSS;by replicate;
var x;
   output out=Bootdist (drop=_freq_ _type_) mean=mean_x max=max_x min=min_x;
run;


data OutStatsUni; set Bootdist;
stat1 = 2*mean_x; stat2=max_x; stat3=min_x + max_x; run;

title3 '2* xbar';
proc sgplot data=OutStatsUni;
   histogram stat1; run;

title3 'x_(n)';
proc sgplot data=OutStatsUni;
   histogram stat2; run;

title3 'x_(1) + x_(n)';
proc sgplot data=OutStatsUni;
   histogram stat3; run;

title3 'Average over bootstrap samples';
proc means data=OutStatsUni mean std min max; var stat1 stat2 stat3;
output out=Outstats2;run;
```

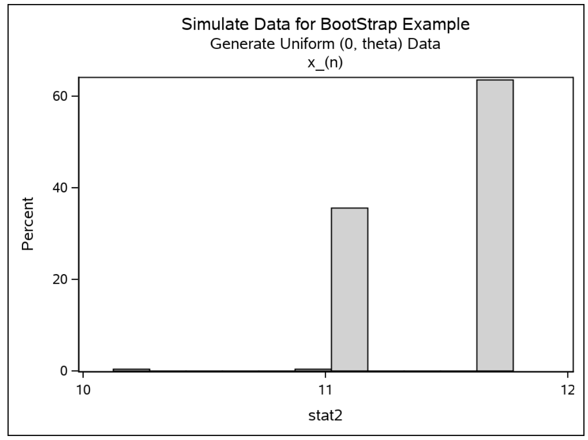
Output

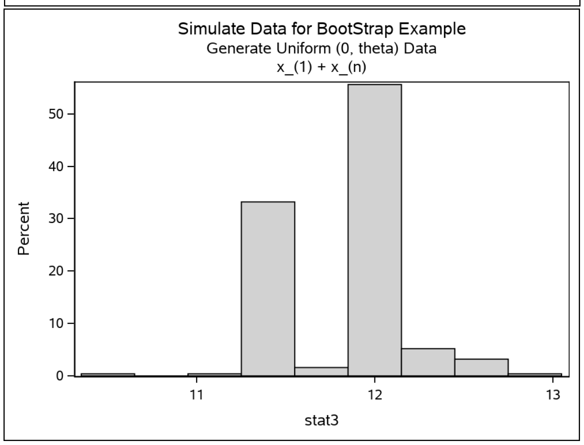
Simulate Data for BootStrap Example

Generate Uniform (0, theta) Data

Average over bootstrap samples







The MEANS Procedure

Variable	Mean	Std Dev	Minimum	Maximum
$2 * \bar{x}$	11.2266360	1.4252300	7.6495045	15.6029231
$x_{(n)}$	11.4967737	0.3040389	10.1544953	11.7216010
$x_{(1)} + x_{(n)}$	11.8337423	0.3839096	10.3705517	12.9819779

From this table one observes that $x_{(n)}$ is biased but has the smallest std. error(Std Dev).

Assignment

1. Repeat the R simulation with a new function for the statistic $V = Y_{(1)} + Y_{(n)}$.
2. Generate data from an exponential distribution with mean $\lambda = 8$. Perform the simulation using the bootstrap code with the statistic $\bar{x}$ and the sample median(x).