22S:166
More on the Bootstrap

Lecture 9
September 21, 2009

Kate Cowles
374 SH, 335-0727
kcowles@stat.uiowa.edu

Choosing the number of bootstrap datasets

- approximately 1000 to 2000 is minumum for reasonable performance in most cases
- choosing R = 999 or 1999 facilitates calculation of percentile confidence intervals (see below)

Another version of the function for calculating the statistic for the city data

```
> meanratio
function( df, indices)
{
        #df must be data frame with 2 colur
mean( df[indices, "x"]) / mean( df[indices,

}
```


Running the bootstrap with different settings of R

```
> library(boot)

Attaching package: 'boot'

        The following object(s) are masked _by_ .GlobalEnv :

         city

> data(city)
>
> boot.out <- boot( city, meanratio, R=999)
> boot.out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = city, statistic = meanratio, R = 999)


Bootstrap Statistics :
    original    bias    std. error
t1* 1.520312 0.0338232    0.218307
>

> boot.out <- boot( city, meanratio, R=999)
> boot.out

ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
Call:
boot(data = city, statistic = meanratio, R = 999)


Bootstrap Statistics :
    original     bias    std. error
t1* 1.520312 0.04969103   0.2316369
>

> boot.out <- boot( city, meanratio, R=1999)
> boot.out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = city, statistic = meanratio, R = 1999)


Bootstrap Statistics :
    original     bias    std. error
t1* 1.520312 0.04079779   0.2294637

> boot.out <- boot( city, meanratio, R=1999)
> boot.out

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = city, statistic = meanratio, R = 1999)
```



```
Bootstrap Statistics :
    original     bias    std. error
t1* 1.520312 0.03994395    0.222143
```


Interpreting the boot object

R code for the City Data

```
> library(boot)
> help(boot, package="boot")
```

```
boot                    package:boot                    R Documentation

Bootstrap Resampling

Description:

     Generate 'R' bootstrap replicates of a statistic applied to data.
     Both parametric and nonparametric resampling are possible.  For
     the nonparametric bootstrap, possible resampling methods are the
     ordinary bootstrap, the  balanced bootstrap, antithetic
     resampling, and permutation. For nonparametric multi-sample
     problems stratified resampling is used.  This is specified by
     including a vector of strata in the call to boot. Importance
     resampling weights may be specified.

Usage:

     boot(data, statistic, R, sim="ordinary", stype="i",
          strata=rep(1,n), L=NULL, m=0, weights=NULL,
          ran.gen=function(d, p) d, mle=NULL, ...)

Arguments:

    data: The data as a vector, matrix or data frame.  If it is a
          matrix or data frame then each row is considered as one
          multivariate observation.

statistic: A function which when applied to data returns a vector
          containing the statistic(s) of interest.  When
          'sim="parametric"', the first argument to 'statistic' must be
          the data.  For each replicate a simulated dataset returned by
          'ran.gen' will be passed.  In all other cases 'statistic'
          must take at least two arguments.  The first argument passed
          will always be the original data. The second will be a vector
          of indices, frequencies or weights which define the bootstrap
          sample. Further, if predictions are required, then a third
          argument is required which would be a vector of the random
          indices used to generate the bootstrap predictions.  Any
```

further arguments can be passed to 'statistic' through the '...[]' arguments.

R: The number of bootstrap replicates. Usually this will be a single positive integer. For importance resampling, some resamples may use one set of weights and others use a different set of weights. In this case 'R' would be a vector of integers where each component gives the number of resamples from each of the rows of weights.

sim: A character string indicating the type of simulation required. Possible values are '"ordinary"' (the default), '"parametric"', '"balanced"', '"permutation"', or '"antithetic"'. Importance resampling is specified by including importance weights; the type of importance resampling must still be specified but may only be '"ordinary"' or '"balanced"' in this case.

stype: A character string indicating what the second argument of statistic represents. Possible values of stype are '"i"' (indices - the default), '"f"' (frequencies), or '"w"' (weights).

Details:

The statistic to be bootstrapped can be as simple or complicated as desired as long as its arguments correspond to the dataset and (for a nonparametric bootstrap) a vector of indices, frequencies or weights. 'statistic' is treated as a black box by the 'boot' function and is not checked to ensure that these conditions are met.

Value:

The returned value is an object of class '"boot"', containing the following components :

t0: The observed value of 'statistic' applied to 'data'.

t: A matrix with 'R' rows each of which is a bootstrap replicate of 'statistic'.

R: The value of 'R' as passed to 'boot'.

data: The 'data' as passed to 'boot'.

seed: The value of '.Random.seed' when 'boot' was called.

statistic: The function 'statistic' as passed to 'boot'.

sim: Simulation type used.

stype: Statistic type as passed to 'boot'.

Example of nonparametric bootstrap with boot package:

```
# define "statistic" function
> meanratio <-
function( mydat, indices )
{
if (!(is.matrix( mydat)  && ncol(mydat) == 2 & length(indices)==
nrow(mydat) ))
{
    stop("invalid arguments")
}

mean( mydat[indices,2]) / mean(mydat[indices,1])
}

# call boot function
> boot.out <- boot( as.matrix(city), meanratio, 999)

# summarize results

> boot.out

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = as.matrix(city), statistic = meanratio, R = 999)


Bootstrap Statistics :
    original     bias    std. error
t1* 1.520312 0.04051090   0.2263570
```

```
# bootstrap c.i.

> help(boot.ci, package="boot")

boot.ci                 package:boot                 R Documentation

Nonparametric Bootstrap Confidence Intervals

Description:

     This function generates 5 different types of equi-tailed two-sided
     nonparametric confidence intervals.  These are the first order
     normal  approximation, the basic bootstrap interval, the
     studentized bootstrap  interval, the bootstrap percentile
     interval, and the adjusted bootstrap  percentile (BCa) interval.
     All or a subset of these intervals can be  generated.

Usage:

     boot.ci(boot.out, conf = 0.95, type = "all",
             index = 1:min(2,length(boot.out$t0)), var.t0 = NULL,
             var.t = NULL, t0 = NULL, t = NULL, L = NULL, h = function(t) t,
             hdot = function(t) rep(1,length(t)), hinv = function(t) t, ...)

Arguments:

boot.out: An object of class '"boot"' containing the output of a
          bootstrap calculation.

    conf: A scalar or vector containing the confidence level(s) of the
          required interval(s).


    type: A vector of character strings representing the type of
          intervals required. The value should be any subset of the
          values 'c("norm","basic", "stud", "perc", "bca")' or simply
          '"all"' which will compute all five types of intervals.
```

```
> boot.ci(boot.out)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates
```



```
CALL :
boot.ci(boot.out = boot.out)

Intervals :
Level      Normal              Basic
95%   ( 1.036,  1.923 )   ( 0.848,  1.786 )

Level     Percentile            BCa
95%   ( 1.254,  2.192 )   ( 1.264,  2.231 )
Calculations and Intervals on Original Scale
Warning message:
In boot.ci(boot.out = boot.out) :
        bootstrap variances needed for studentized intervals
```