Choosing the number of bootstrap datasets

• approximately 1000 to 2000 is minimum 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

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

Running the bootstrap with different settings of R

```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 0.049691 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 0.03994395 0.222143

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

ORDINARY NONPARAMETRIC BOOTSTRAP

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

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

R code for the City Data

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

Interpreting the boot object

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

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

> 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

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

ORDINARY NONPARAMETRIC BOOTSTRAP

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

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

R code for the City Data

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

Interpreting the boot object

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

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

> 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

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

ORDINARY NONPARAMETRIC BOOTSTRAP

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

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

R code for the City Data

> library(boot)
> help(boot, package="boot")
further arguments can be passed to 'statistic' through the '...' argument.

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 the 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'.

Example of nonparametric bootstrap with boot package:

```r
# 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")

Nonparametric Bootstrap Confidence Intervals

This function generates 5 different types of equi-tailed two-sided nonparametric confidence intervals. These are the first or der 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:

```r
boot.ci(boot.out, conf = 0.95, type = "all",
       index = 1:dim(boot.out)[[2]], var = NULL,
       var.t = NULL, t0 = NULL, t = NULL, L = NULL, R = 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.

Example of parametric bootstrap with boot package:

```r
# 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")

Nonparametric Bootstrap Confidence Intervals

This function generates 5 different types of equi-tailed two-sided nonparametric confidence intervals. These are the first or der 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:

```r
boot.ci(boot.out, conf = 0.95, type = "all",
       index = 1:dim(boot.out)[[2]], var = NULL,
       var.t = NULL, t0 = NULL, t = NULL, L = NULL, R = 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.

Example of parametric bootstrap with boot package:

```r
# 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")

Nonparametric Bootstrap Confidence Intervals

This function generates 5 different types of equi-tailed two-sided nonparametric confidence intervals. These are the first or der 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:

```r
boot.ci(boot.out, conf = 0.95, type = "all",
       index = 1:dim(boot.out)[[2]], var = NULL,
       var.t = NULL, t0 = NULL, t = NULL, L = NULL, R = 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.