Introduction to resampling methods

Vivien Rossi

CIRAD - UMR Ecologie des forêts de Guyane
vivien.rossi@cirad.fr

Master 2 - Ecologie des Forêts Tropicale
AgroParisTech - Université Antilles-Guyane
Kourou, novembre 2010

objectives of the course

1. to present resampling technics
   - randomization tests
   - cross-validation
   - jackknife
   - bootstrap
2. to apply with R

outlines

Resampling in statistics

Description

set of statistical inference methods based on *new* samples drawn from the initial sample

Implementation

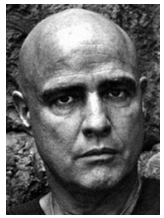
- computer simulation of these *new* samples
- analysing these *new* data to refine the inference

Classical uses

- estimation/bias reduction of an estimate (**jackknife, bootstrap**)
- estimation of confidence intervalle without normality assumption (**bootstrap**)
- exacts tests (**permutation tests**)
- model validation (**cross validation**)

History of resampling techniques

1935 *randomization tests*, Fisher 

1948 *cross-validation*, Kurtz 

1958 *jackknife*, Quenouille-Tukey 

1979 *bootstrap*, Efron 

Resampling mecanism

Why it works ?

- can we expect an improvement by resampling from the same sample?
- no new information is brought back !
- but it can help to extract useful information from the base sample

the idea

- to consider the sample like the population
- to simulate samples that we could see
- to handle "scale" relationship into the sample

Illustration with Russian dolls

How many flowers is there on the biggest doll?

population

sample

sub-samples

randomization tests or *permutation tests*

goal: testing assumption

$$\begin{cases} H_0 : & X \text{ and } Y \text{ are independants} \\ H_1 : & X \text{ and } Y \text{ are dependants} \end{cases}$$

principle: data are randomly re-assigned so that a p-value is calculated based on the permutated data

permutation tests exhaust all possible outcomes $\Rightarrow$ exact tests $\Rightarrow$ exact p-value

randomization tests resampling simulates a large number of possible outcomes $\Rightarrow$ approximate p-value

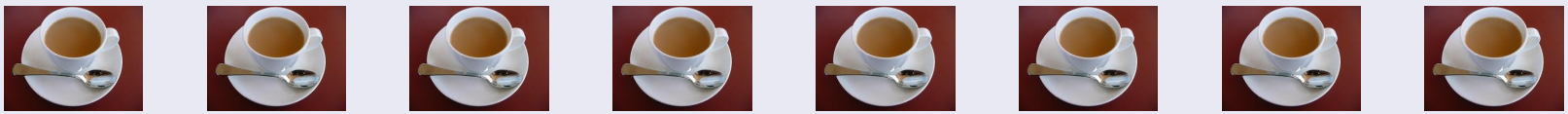
example from Fisher: Lady tasting tea

a typical british story

In 1920, R. A. Fisher met a lady who insisted that her tongue was sensitive enough to detect a subtle difference between a cup of tea with the milk being poured first and a cup of tea with the milk being added later. Fisher was skeptical . . .

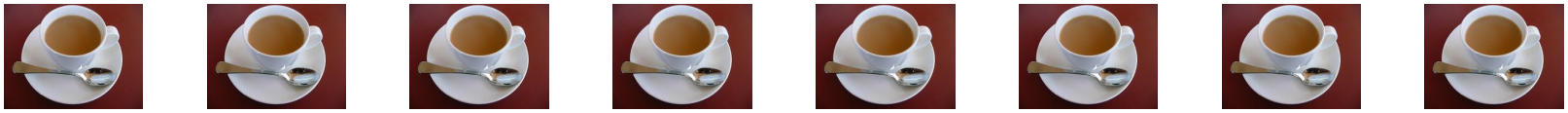
Fisher experiment: he presented 8 cups of tea to this lady

4 cups were 'milk-first' and 4 others were 'tea-first':



tea or milk first ?

example from Fisher: Lady tasting tea

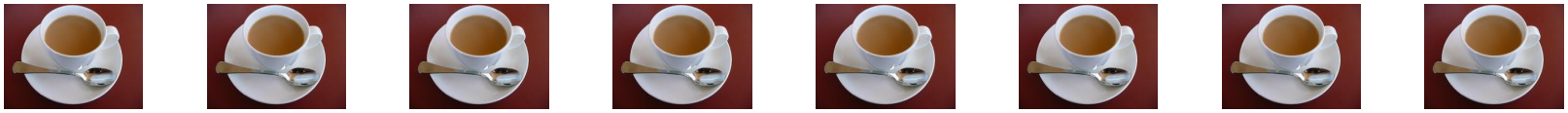


tea or milk first ?

experiment result: the lady well detected the 8 cups

Did the woman really have a super-sensitive tongue?

example from Fisher: Lady tasting tea



tea or milk first ?

experiment result: the lady well detected the 8 cups

Did the woman really have a super-sensitive tongue?

Reformulation as a statistical test

H_0: the order in which milk or tea is poured in a cup and the lady's detection of the order are independent.

H_1: the lady can correctly tell the order in which milk or tea is poured in a cup.

example from Fisher: Lady tasting tea

Reformulation as a statistical test

H_0: the order in which milk or tea is poured in a cup and the lady's detection of the order are independent.

H_1: the lady can correctly tell the order in which milk or tea is poured in a cup.

example from Fisher: Lady tasting tea

Reformulation as a statistical test

H_0: the order in which milk or tea is poured in a cup and the lady's detection of the order are independent.

H_1: the lady can correctly tell the order in which milk or tea is poured in a cup.

probabilities of all the possibilities under H_0

(1,0,1,1,0,0,1,0)	number of match	probability
(1,1,1,1,0,0,0,0)	6	1/nb possibilities
(1,1,1,0,1,0,0,0)	4	1/nb possibilities
⋮	⋮	⋮
(1,0,1,1,0,0,1,0)	8	1/nb possibilities
⋮	⋮	⋮
(0,0,0,0,1,1,1,1)	2	1/nb possibilities

example from Fisher: Lady tasting tea

Reformulation as a statistical test

H_0: the order in which milk or tea is poured in a cup and the lady's detection of the order are independent.
H_1: the lady can correctly tell the order in which milk or tea is poured in a cup.

test result

The probability of matching the 8 cups under H_0 is 1/(nb possibilities)

number of possibilities $\left(\begin{array}{c} 8 \\ 4 \end{array} \right) = 70$

i.e.The probability that the Lady matched by chance the 8 cups is $1/70 \approx 0.014$

example from Fisher: Lady tasting tea

Reformulation as a statistical test

H_0: the order in which milk or tea is poured in a cup and the lady's detection of the order are independent.
H_1: the lady can correctly tell the order in which milk or tea is poured in a cup.

test result

The probability of matching the 8 cups under H_0 is 1/(nb possibilities)

number of possibilities $\left(\begin{array}{c} 8 \\ 4 \end{array} \right) = 70$

i.e.The probability that the Lady matched by chance the 8 cups is $1/70 \approx 0.014$

exercise

What can we say if the Lady had matched only 6 cups ?

example from Fisher: Lady tasting tea

exercise

What can we say if the Lady had matched only 6 cups ?

some tips to do it with R

- exhausting all possibilities: see function *combn*
- matching of all the possibilities: loops *for* or function *apply*

example from Fisher: Lady tasting tea

exercise

What can we say if the Lady had matched only 6 cups ?

some tips to do it with R

- exhausting all possibilities: see function *combn*
- matching of all the possibilities: loops *for* or function *apply*

a solution

```
TeaCup <- function(NbCup=8){
      NbMilk=NbCup/2
      ref=rep(0,NbCup)
      ref[sample(NbCup,NbMilk)]=1
      possibilities=combn(1:NbCup,NbMilk)
      score=rep(0,ncol(possibilities))
      for (i in 1:ncol(possibilities)){
          score[i]=sum(ref[possibilities[,i]])*2
     }
     Probabilities=table(score)/choose(NbCup,NbMilk)
     return(Probabilities)
}
```

example: Comparison of two groups

Hot-dog data

type	calories
meat	186 181 176 149 184 190 158 139 175 148 152 111 141 153 190 157 131 149 135 132
poultry	129 132 102 106 94 102 87 99 107 113 135 142 86 143 152 146 144

example: Comparison of two groups

Hot-dog data

type	calories
meat	186 181 176 149 184 190 158 139 175 148 152 111 141 153 190 157 131 149 135 132
poultry	129 132 102 106 94 102 87 99 107 113 135 142 86 143 152 146 144

Hypothesis test

H_0: the hot-dog type and calories are independents
H_1: the hot-dog type and calories are dependants

Exercise with R

- Make the test, functions from package *coin* may be helpful
- What test is it if you use the ranks ?

example: test significance of covariate in linear regression

Linear model

$$Y_i = \beta_0 + \beta_1 x_i^1 + \cdots + \beta_p x_i^p + \varepsilon_i \quad i = 1, \ldots, n$$

example: test significance of covariate in linear regression

Linear model

$$Y_i = \beta_0 + \beta_1 x_i^1 + \cdots + \beta_p x_i^p + \varepsilon_i \quad i = 1, \ldots, n$$

hypothesis test

H_0: Y and $(X_1, \ldots, X_p)$ are linearly independents
H_1: Y and $(X_1, \ldots, X_p)$ are linearly dependents

example: test significance of covariate in linear regression

Linear model

$$Y_i = \beta_0 + \beta_1 x_i^1 + \cdots + \beta_p x_i^p + \varepsilon_i \quad i = 1, \ldots, n$$

hypothesis test

H_0: Y and $(X_1, \ldots, X_p)$ are linearly independents
H_1: Y and $(X_1, \ldots, X_p)$ are linearly dependents

exercise with R

- simulate data according to the model
- proceed a permutation test with the function *lmp* from the package *lmPerm*

Cross-validation

goal

to estimate how accurately a predictive model will perform in practice

principle

1. splits the dataset into training and validation data
2. fit model on training data
3. assess predictive accuracy on validation data (RMSE)

some types of cross-validation

random subsampling (randomly splits the data into training and validation data) $\times s$

K fold randomly splits the data into K subsamples ($K-1$ for training and 1 for validation) $\times K$

leave-one-out ($N-1$ obs for training and 1 obs for validation) $\times N$

...

Example: K-fold

- data: $(x_i, y_i)_{i=1,\ldots,100}$
- 5 folds: $d_1 = (x_i, y_i)_{i=1,\ldots,20}$, $d_2 = (x_i, y_i)_{i=21,\ldots,40}$, $d_3 = (x_i, y_i)_{i=41,\ldots,60}$, $d_4 = (x_i, y_i)_{i=61,\ldots,80}$ and $d_5 = (x_i, y_i)_{i=81,\ldots,100}$

CV for model $\mathcal{M}$: $\hat{y} = f(x, \theta)$

- assess prediction for subsample d_1
  1. training: $\theta^1 = \arg\min_\theta \sum_{i=21}^{100} (f(x_i, \theta) - y_i)^2$
  2. predicting: $\hat{y}_i = f(x_i, \theta^1)$ for $i = 1, \ldots, 20$
  3. validation: $MSE_1 = \sum_{i=1}^{20} (\hat{y}_i - y_i)^2 / 20$
- idem for $d_2, \ldots, d_5 \rightarrow MSE_2, \ldots, MSE_5$
- $CV = mean(MSE_1, MSE_2, MSE_3, MSE_4, MSE_5)$

excercise with R

linear regression

1. simulate data according to the model
   $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \varepsilon$
2. compute CV for several covariate combination (use function *cv.glm* form package *boot*

excercise with R

linear regression

1. simulate data according to the model
   $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \varepsilon$
2. compute CV for several covariate combination (use function *cv.glm* form package *boot*

generalised linear model

1. simulate data according to the model
   $y \sim \mathcal{P}\big(\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)\big)$
2. compute CV for several covariate combination (use function *cv.glm* form package *boot*

excercise with R

linear regression

1. simulate data according to the model
   $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \varepsilon$
2. compute CV for several covariate combination (use function *cv.glm* form package *boot*

generalised linear model

1. simulate data according to the model
   $y \sim \mathcal{P}\big(\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)\big)$
2. compute CV for several covariate combination (use function *cv.glm* form package *boot*

any model

- see function *crossval* from package *boostrap*
- make your own function

Cross-validation caracteristics

generalities

- easy to implement, the prediction method need only be available as a "black box"
- provides *ECV* a nearly unbiased estimate of the expectation of the fitting criterion
- can be very slow since the training must be carried out repeatedly, methods to speed up exists but . . .

limitations

- validation set and test set must drawn from the same population
- be careful with dynamical systems and stratified population
- instability for small sample
- variance of *CV* can be large $\rightarrow$ to take into account for model selection

the Jackknife

overview

- proposed to reduce the bias of an estimator (delete if $bias = a/n$)
- generally used to estimate the variance or covariance
- useless for unbiased estimators (mean)
- close to leave one out method
- only adapted for estimators which are smooth function of the observation

principle

- let a sample $x_1, \ldots, x_n \sim \mathcal{L}(\theta)$
- let T an estimator of θ computed with $x_1, \ldots, x_n$
- for $i = 1, \ldots, n$: T_{-i} is the estimator computed with $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$
- the Jackknife estimators: $T_J = T - \frac{n-1}{n} \sum_{i=1}^{n} (T_{-i} - T)$
- bias of the estimator: $-\frac{n-1}{n} \sum_{i=1}^{n} (T_{-i} - T)$

excercise: Jackknife estimator for the variance

estimators of variance

Let $x_1, \ldots, x_n \sim \mathcal{N}(\mu, \sigma^2)$, the two usual estimators of σ^2:

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \text{and} \quad \tilde{S}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

their expectations:

$$E[S^2] = \frac{n-1}{n}\sigma^2 \quad \text{and} \quad E[\tilde{S}^2] = \sigma^2$$

excercise: Jackknife estimator for the variance

estimators of variance

Let $x_1, \ldots, x_n \sim \mathcal{N}(\mu, \sigma^2)$, the two usual estimators of σ^2:

$$S^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \text{and} \quad \tilde{S}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

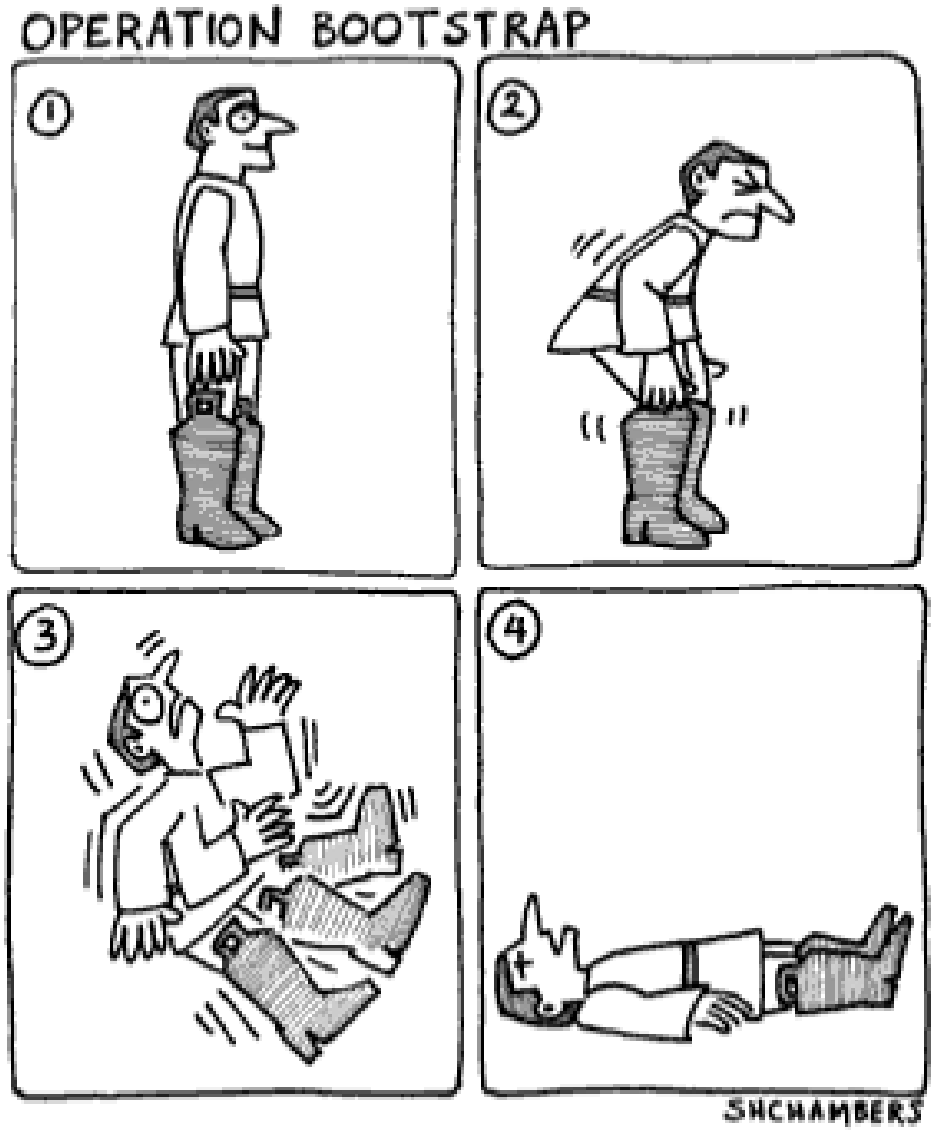
their expectations:

$$E[S^2] = \frac{n-1}{n}\sigma^2 \quad \text{and} \quad E[\tilde{S}^2] = \sigma^2$$

exercise with R

- simulate a sample
- compute the jackknife estimator of S^2 using the function *jackknife* from package *bootstrap*
- compare with the estimator $\tilde{S}^2$

The bootstrap



The bootstrap

goal

- estimate confidence interval of a parameter with or with assumption on data distribution
- estimate bias of an estimator

principle of the nonparametric bootstrap

- starting point: $x_1, \ldots, x_n$ *i.i.d* $\sim \mathcal{L}(\theta)$
- repeat for $k = 1, \ldots, K$
  1. sample with replacement $x_1^*, \ldots, x_n^*$ among $x_1, \ldots, x_n$
  2. compute θ_k^* the estimates of θ with $x_1^*, \ldots, x_n^*$
- $\theta_1^*, \ldots, \theta_K^* \rightsquigarrow$ inference on θ

many variants

- parametric bootstrap
- smooth bootstrap ...

The bootstrap: generalities

- In Jackknife, the number of resamples is confined by the number of observations (n-1).
- In bootstrap, the original sample could be duplicated as many times as the computing resources allow, and then this expanded sample is treated as a virtual population.

- Unlike cross validation and Jackknife, the bootstrap employs sampling with replacement

- In cross-validation and Jackknife, the n in the sub-sample is smaller than that in the original sample
- In bootstrap every resample has the same number of observations as the original sample.
- Thus, the bootstrap method has the advantage of modeling the impacts of the actual sample size

example: median estimation based on bootstrap

sample: 10, 15, 17, 8 and 11 $\rightarrow$ median estimate = 11

resampling	median
8 11 15 10 8	10
10 15 10 10 11	10
15 17 10 10 17	15
11 15 10 10 17	11
17 17 15 17 10	15
10 15 10 17 10	10
10 8 8 8 15	8

median estimate by bootstrap: mean(10,10,15,11,15,10,8)=11.29
sd of median estimate by bootstrap: sd(10,10,15,11,15,10,8)=2.69

example: median estimation based on bootstrap

sample: 10, 15, 17, 8 and 11 $\rightarrow$ median estimate = 11

resampling	median
8 11 15 10 8	10
10 15 10 10 11	10
15 17 10 10 17	15
11 15 10 10 17	11
17 17 15 17 10	15
10 15 10 17 10	10
10 8 8 8 15	8

median estimate by bootstrap: mean(10,10,15,11,15,10,8)=11.29
sd of median estimate by bootstrap: sd(10,10,15,11,15,10,8)=2.69

exercise with R

estimate the confidence interval using the function *bootstrap* from package *bootstrap*

example: parametric bootstrap

principle of the parametric bootstrap

- starting point: $x_1, \ldots, x_n$ $i.i.d \sim \mathcal{L}(\theta)$ where $\mathcal{L}$ known
- compute $\hat{\theta}$ the mle of θ
- repeat for $k = 1, \ldots, K$
  1. sample $x_1^*, \ldots, x_n^*$ $i.i.d \sim \mathcal{L}(\hat{\theta})$
  2. compute θ_k^* the mle of θ with $x_1^*, \ldots, x_n^*$
- $\theta_1^*, \ldots, \theta_K^* \rightsquigarrow$ inference on θ

example: parametric bootstrap

principle of the parametric bootstrap

- starting point: $x_1, \ldots, x_n$ $i.i.d \sim \mathcal{L}(\theta)$ where $\mathcal{L}$ known
- compute $\hat{\theta}$ the mle of θ
- repeat for $k = 1, \ldots, K$
  1. sample $x_1^*, \ldots, x_n^*$ $i.i.d \sim \mathcal{L}(\hat{\theta})$
  2. compute θ_k^* the mle of θ with $x_1^*, \ldots, x_n^*$
- $\theta_1^*, \ldots, \theta_K^* \rightsquigarrow$ inference on θ

Exercise with R: the exponential distribution

- simulate a sample from $Exp(\lambda)$ with $\lambda = 5$
- estimate λ using both parametric bootstrap and nonparametric bootstrap (function *boot* from package *boot*)

AgroParisTech

example: resampling residual

principle of the resampling residual

- starting point: $(x_1, y_1), \ldots, (x_n, y_n)$ and a model $f(\cdot, \theta)$
- calibrate the model $\rightarrow \hat{\theta}$: $\hat{y}_i = f(x_i, \hat{\theta}) \rightsquigarrow \varepsilon_i = \hat{y}_i - y_i$
- repeat for $k = 1, \ldots, K$
  1. $\forall i$, $y_i^* = y_i + \tilde{\varepsilon}$ with ε sampled from $\varepsilon_1, \ldots, \varepsilon_n$
  2. calibrate the model with $(x_1, y_1^*), \ldots, (x_n, y_n^*) \rightsquigarrow \theta_k^*$
- $\theta_1^*, \ldots, \theta_K^* \rightsquigarrow$ inference on θ

exercise with R: the case of the linear model

- simulate data according to a linear model
- estimate regression parameters using the function *lm.boot* from the package *simpleboot*

Rationale of supporting resampling

Empirical empirical-based resampling do not require assumptions on the sample or the population.

Clarity resampling is clean and simple. High mathematical background is not required to comprehend it

Small sample size Distributional assumptions requierd by classical procedures are usually met by a large sample size. Bootstrapping could treat a small sample as the virtual population to "generate" more observations

Non-random sample Resampling is valid for any kind of data, including random and non-random data.

Large sample size Given a very large sample size, one can reject virtually any null hypothesis $\rightarrow$ divide the sample into subsets, and then apply a simple or double cross-validation.

Replications Repeated experiments in resampling such as cross-validation and bootstrap can be used as internal replications.

Criticisms of resampling

Assumption "You're trying to get something for nothing". Every theory and procedure is built on certain assumptions and requires a leap of faith to some degree. Indeed, the classical statistics framework requires more assumptions than resampling does

Generalization resampling is based on one sample and therefore the generalization cannot go beyond that particular sample.

Bias confidence intervals obtained by simple bootstrapping are always biased though the bias decreases with sample size. (for normal case the bias in is at least n/(n-1))

Accuracy for small sample resampling may be less accurate than conventional parametric methods. Not very convincing argument because today computers are very powerful.

pros and cons in both traditional and resampling methods carry certain valid points. $\rightarrow$ the appropriateness of the methodology highly depends on the situation

R packages for resampling methods

boot	quite a wide variety of bootstrapping tricks.
bootstrap	relatively simple functions for bootstrapping and related techniques.
coin	permutation tests
Design	includes bootcov for bootstrapping the covariance of estimated regression parameters and validate for testing the quality of fitted models by cross validation or bootstrapping.
MChtest	Monte Carlo hypothesis tests: tests using some form of resampling.
meboot	a method of bootstrapping a time series.
permtest	a function for permutation tests of microarray data.
resper	for doing restricted permutations.
scaleboot	produces approximately unbiased hypothesis tests via bootstrapping.
simpleboot	performs bootstraps in simple situations: one and two samples, and linear regression.