

# The leaps Package

February 16, 2008

**Title** regression subset selection

**Version** 2.7

**Author** Thomas Lumley <tlumley@u.washington.edu> using Fortran code by Alan Miller

**Description** Regression subset selection including exhaustive search

**Depends**

**License** GPL version 2 or later

**Maintainer** Thomas Lumley <tlumley@u.washington.edu>

## R topics documented:

leaps . . . . .	1
leaps.setup . . . . .	3
plot.regsubsets . . . . .	4
regsubsets . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

leaps *all-subsets regression*

---

## Description

leaps() performs an exhaustive search for the best subsets of the variables in x for predicting y in linear regression, using an efficient branch-and-bound algorithm. It is a compatibility wrapper for [regsubsets](#) does the same thing better.

## Usage

```
leaps(x=, y=, wt=rep(1, NROW(x)), int=TRUE, method=c("Cp", "adjr2", "r2"), nbest=10)
```

**Arguments**

<code>x</code>	A matrix of predictors
<code>y</code>	A response vector
<code>wt</code>	Optional weight vector
<code>int</code>	Add an intercept to the model
<code>method</code>	Calculate Cp, adjusted R-squared or R-squared
<code>nbest</code>	Number of subsets of each size to report
<code>names</code>	vector of names for columns of <code>x</code>
<code>df</code>	Total degrees of freedom to use instead of <code>nrow(x)</code> in calculating Cp and adjusted R-squared
<code>strictly.compatible</code>	Implement misfeatures of <code>leaps()</code> in S

**Value**

A list with components

<code>which</code>	logical matrix. Each row can be used to select the columns of <code>x</code> in the respective model
<code>size</code>	Number of variables, including intercept if any, in the model
<code>cp</code>	or <code>adjr2</code> or <code>r2</code> is the value of the chosen model selection statistic for each model
<code>label</code>	vector of names for the columns of <code>x</code>

**Note**

With `strictly.compatible=T` the function will stop with an error if `x` is not of full rank or if it has more than 31 columns. It will ignore the column names of `x` even if `names==NULL` and will replace them with "0" to "9", "A" to "Z".

**References**

Alan Miller "Subset Selection in Regression" Chapman & Hall

**See Also**

[regsubsets](#), [regsubsets.formula](#), [regsubsets.default](#)

**Examples**

```
x<-matrix(rnorm(100),ncol=4)
y<-rnorm(25)
leaps(x,y)
```

---

leaps.setup	<i>Internal functions for leaps(), subsets()</i>
-------------	--

---

## Description

These functions are used internally by `regsubsets` and `leaps`. They are wrappers for Fortran routines that construct and manipulate a QR decomposition.

## Usage

```
leaps.setup(x, y, wt=rep(1, length(y)), force.in=NULL, force.out=NULL, intercept=TRUE, nvmax)
leaps.seqrep(leaps.obj)
leaps.exhaustive(leaps.obj, really.big=FALSE)
leaps.backward(leaps.obj)
leaps.forward(leaps.obj)
```

## Arguments

<code>x</code>	A matrix of predictors
<code>y</code>	A response vector
<code>wt</code>	Optional weight vector
<code>intercept</code>	Add an intercept to the model
<code>force.in</code>	vector indicating variable that must be in the model
<code>force.out</code>	vector indicating variable that must not be in the model
<code>nbest</code>	Number of subsets of each size to report
<code>nvmax</code>	largest subset size to examine
<code>warn.dep</code>	warn if <code>x</code> is not of full rank
<code>leaps.obj</code>	An object of class <code>leaps</code> as produced by <code>leaps.setup</code>
<code>really.big</code>	required before R gets sent off on a long uninterruptible computation

## See Also

[regsubsets](#), [leaps](#)

---

plot.regsubsets      *Graphical table of best subsets*

---

### Description

Plots a table of models showing which variables are in each model. The models are ordered by the specified model selection statistic. This plot is particularly useful when there are more than ten or so models and the simple table produced by `summary.regsubsets` is too big to read.

### Usage

```
## S3 method for class 'regsubsets':  
plot(x, labels=obj$xnames, main=NULL, scale=c("bic", "Cp", "adjr2", "r2"), col=gray
```

### Arguments

x	regsubsets object
labels	variable names
main	title for plot
scale	which summary statistic to use for ordering plots
col	Colors: the last color should be close to but distinct from white
...	other arguments

### Value

None

### Author(s)

Thomas Lumley, based on a concept by Merlise Clyde

### See Also

[regsubsets](#), [summary.regsubsets](#)

### Examples

```
data(swiss)  
a<-regsubsets(Fertility~., nbest=3, data=swiss)  
par(mfrow=c(1, 2))  
plot(a)  
plot(a, scale="r2")
```

---

regsubsets                      *functions for model selection*

---

### Description

Generic function for regression subset selection with methods for formula and matrix arguments.

### Usage

```
regsubsets(x=, ...)
```

```
## S3 method for class 'formula':
regsubsets(x=, data=, weights=NULL, nbest=1, nvmax=8, force.in=NULL, force.out=NULL, ...)
```

```
## Default S3 method:
regsubsets(x=, y=, weights=rep(1, length(y)), nbest=1, nvmax=8,
force.in=NULL, force.out=NULL, intercept=TRUE, method=c("exhaustive",
"backward", "forward", "seqrep"), really.big=FALSE, ...)
```

```
## S3 method for class 'regsubsets':
summary(object, all.best=TRUE, matrix=TRUE, matrix.logical=FALSE, df=NULL, ...)
```

### Arguments

x	design matrix or model formula for full model
data	Optional data frame
y	response vector
weights	weight vector
nbest	number of subsets of each size to record
nvmax	maximum size of subsets to examine
force.in	index to columns of design matrix that should be in all models
force.out	index to columns of design matrix that should be in no models
intercept	Add an intercept?
method	Use exhaustive search, forward selection, backward selection or sequential replacement to search.
really.big	Must be TRUE to perform exhaustive search on more than 50 variables.
object	regsubsets object
all.best	Show all the best subsets or just one of each size
matrix	Show a matrix of the variables in each model or just summary statistics
matrix.logical	With <code>matrix=TRUE</code> , the matrix is logical TRUE/FALSE or string <code>"*/code"</code>

<code>df</code>	Specify a number of degrees of freedom for the summary statistics. The default is <code>n-1</code>
<code>...</code>	Other arguments for future methods

**Value**

An object of class "regsubsets" containing no user-serviceable parts. It is designed to be processed by `summary.regsubsets`.

**Note**

This function improves on `leaps` in several ways. The design matrix need not be of full rank. The ability to restrict `nvmax` speeds up exhaustive searches considerably. There is no hard-coded limit to the number of variables.

**See Also**

[leaps](#)

**Examples**

```
data(swiss)
a<-regsubsets(as.matrix(swiss[,-1]),swiss[,1])
summary(a)
b<-regsubsets(Fertility~.,data=swiss)
summary(a)
```

# Index

## \*Topic **hplot**

plot.regsubsets, 4

## \*Topic **regression**

leaps, 1

leaps.setup, 3

plot.regsubsets, 4

regsubsets, 5

leaps, 1, 3, 6

leaps.backward(*leaps.setup*), 3

leaps.exhaustive(*leaps.setup*), 3

leaps.forward(*leaps.setup*), 3

leaps.seqrep(*leaps.setup*), 3

leaps.setup, 3

plot.regsubsets, 4

print.regsubsets(*regsubsets*), 5

print.summary.regsubsets

(*regsubsets*), 5

regsubsets, 1–4, 5

regsubsets.default, 2

regsubsets.formula, 2

summary.regsubsets, 4, 6

summary.regsubsets(*regsubsets*), 5