

# Package ‘FindMinIC’

February 19, 2015

**Type** Package

**Title** Find Models with Minimum IC

**Version** 1.6

**Date** 2013-11-15

**Author** Nicholas Lange, Tom Fletcher, Kristen Zygmunt

**Maintainer** Kristen Zygmunt <kriszmz@sci.utah.edu>

**Imports** nlme, sets

**Description** Creates models from all combinations of a list of variables and sorts by minimum IC (information criterion).

**License** LGPL (>= 3.0)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-12-18 01:26:03

## R topics documented:

FindMinIC-package . . . . .	1
cm . . . . .	2
FindMinIC . . . . .	4
FindMinIC-internal . . . . .	6
<b>Index</b>	<b>8</b>

---

FindMinIC-package      *Find Minimum Information Criterion package*

---

## Description

Find and rank the best models by information criterion such as AIC. Looks at models of all possible combinations of the candidate variables with fixed variables always included.

**Author(s)**

Nicholas Lange, Tom Fletcher, Kristen Zygmunt

**References**

*Burnham, K. P.; Anderson, D. R. (2004), "Multimodel inference: understanding AIC and BIC in Model Selection", Sociological Methods and Research 33: 261-304.*

**See Also**

[FindMinIC](#)

---

cm

*Handling "cm" and "cmList" objects*

---

**Description**

Methods for working with candidate model ("cm") and candidate model list ("cmList") objects.

**Usage**

```
getNthModel(object, index)
getFirstModel(object)

summaryTable(object, index, ...)

## S3 method for class 'cmList'
summary(object, ...)

## S3 method for class 'summary.cmList'
print(x, ...)

## S3 method for class 'cm'
formula(x, ...)

## S3 method for class 'cm'
IC(object)

## S3 method for class 'cm'
summary(object, ...)

## S3 method for class 'summary.cm'
print(x, ...)
```

**Arguments**

object	a list of candidate models of class "cmList" that was returned from a call to <code>fmi</code> or <code>FindMinIC</code> .
index	an index into the <code>cmList</code> , for instance, using <code>index = 4</code> in <code>getNthModel</code> will return the 4th best candidate model from the list as ranked by IC.
x	a candidate model of class "cm"
...	some generic methods such as <code>print</code> and <code>summary</code> can take additional arguments, these are passed in to those methods appropriately.

**Details**

Since `FindMinIC` returns a potentially large list of candidate models, functions such as `getNthModel`, `getFirstModel`, `summaryTable` and `summary` can be used to either return specific models or summarize the list of models. Once a particular candidate model (`cm`) has been extracted, functions such as `IC`, `formula`, and `summary` can be used to further understand that particular model.

**Value**

a `cmList` is a list containing the following components:

<code>results</code>	a list of candidate models of class "cm".
<code>data</code>	the data passed in including any changes made for <code>groupedData</code> .
<code>best</code>	the best candidate model
<code>modeltype</code>	the model type provided to the <code>FindMinIC</code> call

a `cm` object is a candidate model containing the following components:

<code>call</code>	the call to <code>lm</code> or <code>lme</code> that generated the candidate model
<code>IC</code>	the IC of the model fit
<code>formula</code>	the formula of the model fit

**Author(s)**

Nicholas Lange, Tom Fletcher, Kristen Zygmunt

**See Also**

[FindMinIC](#)

**Examples**

```
data(iris)

coly="Sepal.Length"
fixed="Sepal.Width"
candidates=c("Species", "-1", "Sepal.Width:Species")

results.lm = FindMinIC(coly, candidates, fixed, iris)
```

```

print(summary(results.lm))

# best model:
print(summary(getFirstModel(results.lm)))
print(summaryTable(results.lm, 1))
print(summary(results.lm$results[[1]]))

# 4th best model:
print(summary(getNthModel(results.lm, 4)))
print(summary(results.lm$results[[4]]))

```

---

FindMinIC

*Find Model with Minimum IC*


---

### Description

Evaluates all models in a set of candidates, and ranks them by IC such as AIC. Either `lm` or `lme` can be used for the model.

### Usage

```

# Find the minimum IC
## Default S3 method:
FindMinIC(coly, candidates = c(""), fixed = c(""), data = list(),
          modeltype = "lm", random = ~1, ic = "AIC", ...)
## S3 method for class 'formula'
FindMinIC(formula, data=list(), na.action=na.omit, fixed = c(""), random = ~1, ...)

# find the minimum IC, fmi is the shorter name form of FindMinIC
## Default S3 method:
fmi(coly, candidates = c(""), fixed = c(""), data = list(),
    modeltype = "lm", random = ~1, ic = "AIC", ...)
## S3 method for class 'formula'
fmi(formula, data=list(), na.action=na.omit, fixed = c(""), random = ~1, ...)

```

### Arguments

<code>formula</code>	A formula containing the response variable and terms. All the terms of the formula become candidates for inclusion as covariates.
<code>na.action</code>	action to use when data contains NAs. Options include <code>na.omit</code> , <code>na.exclude</code> , <code>na.fail</code>
<code>coly</code>	The name of the column to use for the response variable <code>y</code> of the model
<code>candidates</code>	A list of names of columns that are candidates for inclusion as covariates in the model

fixed	A list of names of columns (can be empty) that must always be included in every model
data	An object containing the variables for use in the model.
modeltype	Currently a choice between "lm" (the default) and "lme". If a model follows the calling convention of <code>lm</code> , it might work here, but it is not guaranteed.
random	When modeltype = "lme", use random the same way as would inside a call to <code>lme</code> and to indicate the variable for <code>groupedData</code>
ic	Type of information criterion to used. Defaults to "AIC". Other options are "AICc" or "BIC"
...	Extra arguments are passed directly into the call to <code>lm</code> or <code>lme</code> .

### Details

FindMinIC tries all possible model combinations of the candidate covariates, while always including the same response variable and fixed variables. It returns a list of candidate models ranked by IC. The model combinations include all 2-way interactions among the candidate variables. Other interactions (like  $age^2$ ) can be directly included in the candidates or fixed lists.

### Value

FindMinIC returns a list of candidate models sorted by information criterion IC. The first model has the "best" IC. The list is of class("cmList") while each element of that list is of class("cm") see `cmList` for more details

### Author(s)

Nicholas Lange, Tom Fletcher, Kristen Zygmunt

### References

Burnham, K. P.; Anderson, D. R. (2004), "Multimodel inference: understanding AIC and BIC in Model Selection", *Sociological Methods and Research* 33: 261-304.

### See Also

`getFirstModel`

### Examples

```
data(iris)

coly="Sepal.Length"
fixed="Sepal.Width"
candidates=c("Species", "-1", "Sepal.Width:Species")

results.lm = FindMinIC(coly, candidates, fixed, iris)

# model with lowest IC:
```

```

first.model = getFirstModel(results.lm)
print(summary(first.model))

# model with 3rd lowest IC:
third.model = getNthModel(results.lm, 3)
print(summary(third.model))

# list of first 5 models, ordered by AIC
print(summary(results.lm)$table[1:5,])

# list of first 5 models, ordered by BIC
results.bic = FindMinIC(coly, candidates, fixed, iris, ic="BIC")
print(summary(results.bic)$table[1:5,])

fm = FindMinIC(Infant.Mortality ~ ., data = swiss)
summary(fm)

fm2 = FindMinIC(Infant.Mortality ~ Fertility + Agriculture + Education * Catholic,
               data = swiss)
summary(fm2)

# list of first 5 models, ordered by AICc
if (require(nlme)) {
  results.aicc = FindMinIC(distance~age, data=Orthodont,
                          ic="AICc", model="lme",
                          random= ~ 1 | Subject)
  print(summary(results.aicc))
}

```

---

FindMinIC-internal      *FindMinIC internal functions*

---

## Description

These are internal functions used by FindMinIC. These should generally *not* be used directly.

## Usage

```

splitvars(fixed)
getx(formula)
gety(formula)
getIC(fit, ictype)

```

## Arguments

fixed	a list of strings
formula	a formula object

`fit` a model fit object such as an object returned by `lm`  
`ic type` type of IC to use, options are "AIC", "AICc", "BIC"

**Details**

`splitvars` splits the variables in fixed into their component variables. For instance, `splitvars(c("A","B",C*A","-1","A:D","E:F:G","H|I|J"))` will return `c("A","B","C","1", "D", "E", "F","G","H","I","J"`

`getx` and `gety` will return the x and y sides of the given formula respectively

`getIC` returns the AIC, AICc, or BIC for the given model based on which type was passed in )

**Value**

see details above

**Author(s)**

Nicholas Lange, Tom Fletcher, Kristen Zygmunt

**See Also**

[FindMinIC](#) which should be used directly instead of these methods.

# Index

- \*Topic **methods**
  - cm, [2](#)
  - FindMinIC-internal, [6](#)
- \*Topic **models**
  - FindMinIC, [4](#)
  - FindMinIC-package, [1](#)
- \*Topic **multivariate**
  - FindMinIC, [4](#)
  - FindMinIC-package, [1](#)
- \*Topic **package**
  - FindMinIC-package, [1](#)
- \*Topic **print**
  - cm, [2](#)

  

- cm, [2](#)
- cmList, [5](#)
- cmList (cm), [2](#)

  

- FindMinIC, [2](#), [3](#), [4](#), [7](#)
- FindMinIC-internal, [6](#)
- FindMinIC-package, [1](#)
- fmi, [3](#)
- fmi (FindMinIC), [4](#)
- formula.cm (cm), [2](#)

  

- getFirstModel, [5](#)
- getFirstModel (cm), [2](#)
- getIC (FindMinIC-internal), [6](#)
- getNthModel (cm), [2](#)
- getx (FindMinIC-internal), [6](#)
- gety (FindMinIC-internal), [6](#)
- groupedData, [5](#)

  

- IC.cm (cm), [2](#)

  

- lm, [5](#)
- lme, [5](#)

  

- print.summary.cm (cm), [2](#)
- print.summary.cmList (cm), [2](#)

  

- splitvars (FindMinIC-internal), [6](#)
- summary.cm (cm), [2](#)
- summary.cmList (cm), [2](#)
- summaryTable (cm), [2](#)