

Package ‘fit.models’

April 6, 2017

Version 0.5-14

Date 2017-04-06

Title Compare Fitted Models

Description The fit.models function and its associated methods (coefficients, print, summary, plot, etc.) were originally provided in the robust package to compare robustly and classically fitted model objects. The aim of the fit.models package is to separate this fitted model object comparison functionality from the robust package and to extend it to support fitting methods (e.g., classical, robust, Bayesian, regularized, etc.) more generally.

Author Kjell Konis <kjell.konis@me.com>.

Maintainer Kjell Konis <kjell.konis@me.com>

Imports lattice, stats

Suggests MASS

License GPL

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-06 21:34:00 UTC

R topics documented:

designMD	2
fit.models	3
fmclass.register	4
indexPlot.lmf	5
kernDenPlot.lmf	6
overlaidKernDenPlot.lmf	6
overlaidQQPlot.lmf	7
plot.glmfm	7
plot.lmf	8
qqPlot.lmf	10
scatterPlot.lmf	10

simpleRegPlot.lmfm	11
summary.glmfm	12
summary.lmfm	13

Index	14
--------------	-----------

designMD	<i>Design Matrix Mahalanobis Distance</i>
----------	---

Description

Returns the squared Mahalanobis distance of all rows in the design (model) matrix X and the sample mean vector μ of the columns of X with respect to the sample covariance matrix Σ . This is (for vector x' a row of X) defined as

$$d^2 = (x - \mu)' \Sigma^{-1} (x - \mu)$$

where

$$\mu = \text{colMeans}(X)$$

and

$$\Sigma = \text{cov}(X).$$

Usage

```
designMD(object, ...)
```

Arguments

`object` a fitted model object with a `model.matrix` method.
`...` additional arguments are ignored.

Value

a numeric vector containing the squared Mahalanobis distances.

Examples

```
stack.lm <- lm(stack.loss ~ ., data = stackloss)

# Mahalanobis distance (not squared)
sqrt(designMD(stack.lm))
```

fit.models

*Fit dot Models***Description**

Fit a statistical model using different estimators (e.g., robust and least-squares) or combine fitted models into a single object. Generic methods then produce side-by-side comparisons of the parameter estimates and diagnostic plots.

Usage

```
fit.models(model.list, ...)
```

Arguments

<code>model.list</code>	a list or a character vector containing names of modeling functions. Only required when <code>fit.models</code> is being used to fit models (rather than combine already fitted models into a <code>fit.models</code> object).
<code>...</code>	see details.

Details

There are two distinct ways the `fit.models` function can be used.

The first is to fit the same model using different estimators. In this case, `model.list` should be a character vector or a list where each element is the name of a modeling function and the remaining arguments (in `...`) are the common arguments to the functions in `model.list`. For example, the following command fits robust and least squares linear models to Brownlee's Stack Loss Plant Data.

```
fit.models(c("rlm", "lm"), stack.loss ~ ., data = stackloss)
```

The resulting `fit.models` object is a list with the output of

```
rlm(stack.loss ~ ., data = stackloss)
```

in the first element and

```
lm(stack.loss ~ ., data = stackloss)
```

in the second. The class attribute of the returned list is set (in this case) to `"lmfm"` which is the `fit.models` class (`fmclass`) for comparing linear-model-like fits.

The second use of `fit.models` is to combine fitted model objects. In this case, `fit.models` combines its arguments into a `fit.models` object (a list where element i is occupied by argument i and sets the class attribute to the appropriate `fit.models` class).

Value

The returned object is a list containing the fitted models. The class of the returned object depends on the classes of the model objects it contains.

See Also

[fmclass.add.class](#) for adding a class to an existing fit.models class and [fmclass.register](#) to create a new fit.models class.

Examples

```
# First, use fit.models to fit robust and least squares linear
# regression models to Brownlee's Stack Loss Plant Data.

# Step 1: rlm (robust linear model) is in the MASS package.
library(MASS)

# Step 2: tell fit.models rlm can be compared to lm
fmclass.add.class("lmfm", "rlm")

fm1 <- fit.models(c("rlm", "lm"), stack.loss ~ ., data = stackloss)

summary(fm1) #rlm does not provide p-values or Multiple R-squared

# Second, use fit.models to combine fitted models into a
# fit.models object.

lm.complete <- lm(stack.loss ~ ., data = stackloss)
lm.clean <- lm(stack.loss ~ ., data = stackloss, subset = 5:20)

fm2 <- fit.models(lm.clean, lm.complete)

summary(fm2)
plot(fm2)

# Name the models in the fit.models object.

fm3 <- fit.models(c(Robust = "rlm", "Least Squares" = "lm"),
                 stack.loss ~ ., data = stackloss)

fm4 <- fit.models(Clean = lm.clean, Complete = lm.complete)
```

fmclass.register

Register Comparable Functions

Description

The `fit.models` package maintains a list of comparable models. These functions provide an api to modify this list.

Usage

```
fmclass.register(fmclass, classes, validation.function = NULL)
```

```
fmclass.add.class(fmclass, class, warn = TRUE)
```

Arguments

`fmclass` a character string naming the fit.models class to be added.

`classes` a character vector naming one or more classes that can be compared by the methods defined for the fit.models class in `fmclass`.

`validation.function` a function returning TRUE when the models are comparable.

`class` a character string specifying a class compatible with the methods of `fmclass`.

`warn` a logical value. If TRUE, a warning is printed if `class` is already registered.

Details

See the package vignette.

Value

a null value is invisibly returned.

indexPlot.lmfm	<i>Comparison Index (Time) Plot</i>
----------------	-------------------------------------

Description

Produces side-by-side index (time) plots.

Usage

```
indexPlot.lmfm(x, fun, level = 0.95, id.n = 3, ...)
```

Arguments

`x` an lmfm object.

`fun` a function to extract the appropriate quantity from `x`.

`level` a numeric value between 0 and 1 specifying the confidence level used to draw the threshold in the plot.

`id.n` a non-negative integer value specifying the number of extreme points to identify.

`...` any additional arguments are passed to [xyplot](#).

Value

the trellis object is invisibly returned.

kernDenPlot.lmfm *Comparison Kernel Density Estimate Plot*

Description

Produces side-by-side kernel density estimate plots.

Usage

```
kernDenPlot.lmfm(x, fun, ...)
```

Arguments

x an lmfm object.
fun a function to extract the appropriate quantity from x.
... additional arguments are passed to [xyplot](#).

Value

the trellis object is invisibly returned.

overlaidKernDenPlot.lmfm
 Overlaid Kernel Density Estimate Plot

Description

Produces an overlaid kernel density plot.

Usage

```
overlaidKernDenPlot.lmfm(x, fun, ...)
```

Arguments

x an lmfm object.
fun a function to extract the appropriate quantity from x.
... additional arguments are passed to [densityplot](#).

Value

the trellis object is invisibly returned.

overlaidQQPlot.lmfm *Overlaid Normal QQ Plot*

Description

Produces an overlaid normal QQ plot.

Usage

```
overlaidQQPlot.lmfm(x, fun, ...)
```

Arguments

x	an lmfm object.
fun	a function to extract the appropriate quantity from x.
...	additional arguments are passed to qqmath .

Value

the trellis object is invisibly returned.

plot.glmfm *Comparison Diagnostic Plots for Generalized Linear Models*

Description

Produces a set of comparison diagnostic plots. The plot options are

1. (not used)
2. Deviance Residuals vs. Predicted Values,
3. Response vs. Fitted Values,
4. Normal QQ Plot of Pearson Residuals,
5. Normal QQ Plot of Deviance Residuals,
6. Pearson Residuals vs. Mahalanobis Distance,
7. Scale-Location.

Usage

```
## S3 method for class 'glmfm'
plot(x, which.plots = c(2, 5, 7, 6), ...)
```

Arguments

`x` a `glmfm` object.

`which.plots` either "ask", "all", or a vector of integer values specifying which plots to draw. In the latter case, use the plot numbers given in the description above (or in the "ask" menu). Any other values will be silently ignored.

... other parameters to be passed through to plotting functions.

Value

`x` is invisibly returned.

Side Effects

The selected plots are drawn on a graphics device.

See Also

[qqPlot.lmfm](#) for 4 and 5 and [scatterPlot.lmfm](#) for the others.

Examples

```
# From ?glm:
# A Gamma example, from McCullagh & Nelder (1989, pp. 300-2)

clotting <- data.frame(
  u = c(5,10,15,20,30,40,60,80,100),
  lot1 = c(118,58,42,35,27,25,21,19,18),
  lot2 = c(69,35,26,21,18,16,13,12,12))

lot1 <- glm(lot1 ~ log(u), data = clotting, family = Gamma)
lot2 <- glm(lot2 ~ log(u), data = clotting, family = Gamma)

fm <- fit.models(lot1, lot2)
plot(fm)
```

plot.lmfm

Comparison Diagnostic Plots for Linear Regression Models

Description

Produces a set of comparison diagnostic plots. The plot options are

1. (not used)
2. Normal QQ Plot of Residuals,
3. Kernel Density Estimate of Residuals,
4. Residuals vs. Mahalanobis Distance,

5. Residuals vs. Fitted Values,
6. Scale-Location,
7. Response vs. Fitted Values,
8. Residuals vs. Index (Time),
9. Overlaid Normal QQ Plot of Residuals,
10. Overlaid Kernel Density Estimate of Residuals,
11. Scatter Plot with Overlaid Fits (for simple linear regression models).

Usage

```
## S3 method for class 'lmfm'  
plot(x, which.plots = c(5, 2, 6, 4), ...)
```

Arguments

x	an lmfm object.
which.plots	either "ask", "all", or a vector of integer values specifying which plots to draw. In the latter case, use the plot numbers given in the description above (or in the "ask" menu). Any other values will be silently ignored.
...	additional parameters are ignored.

Value

x is invisibly returned.

Side Effects

The selected plots are drawn on a graphics device.

See Also

See [qqPlot.lmfm](#) for 2, [kernDenPlot.lmfm](#) for 3, [indexPlot.lmfm](#) for 8, [overlaidQQPlot.lmfm](#) for 9, [overlaidKernDenPlot.lmfm](#) for 10, [simpleRegPlot.lmfm](#) for 11, and [scatterPlot.lmfm](#) for the others.

Examples

```
data(stackloss)  
stack.lm <- lm(stack.loss ~ ., data = stackloss)  
stack.clean <- lm(stack.loss ~ ., data = stackloss, subset = 5:20)  
fm <- fit.models(stack.clean, stack.lm)  
plot(fm)
```

 qqPlot.lmfm

Compaison QQ Plot

Description

Produces side-by-side QQ plots. An optional simulated confidence envelope can be included in each plot.

Usage

```
qqPlot.lmfm(x, fun, envelope = TRUE, half.normal = FALSE, n.samples = 250,
            level = 0.95, id.n = 3, qqline = TRUE, ...)
```

Arguments

x	an lmfm object.
fun	a function to extract the appropriate quantity from x.
envelope	a logical value. If TRUE a level confidence envelope is simulated for each QQ plot.
half.normal	a logical value. If TRUE the plot is drawn using the absolute values.
n.samples	a positive integer value giving the number of samples to compute in the simulation of the confidence envelope.
level	a numeric value between 0 and 1 specifying the confidence level for the envelope.
id.n	a non-negative integer value specifying the number of extreme points to identify.
qqline	a logical value. If TRUE, a QQ line is included in the plot.
...	additional arguments are passed to xyplot .

Value

the trellis object is invisibly returned.

scatterPlot.lmfm

Comparison Scatter Plot

Description

Produces a side-by-side scatter plot.

Usage

```
scatterPlot.lmfm(object, x.fun, y.fun, ...)
```

Arguments

object	an lmfm object.
x.fun	a function to extract the x-axis quantity.
y.fun	a function to extract the y-axis quantity.
...	additional arguments.

Value

the trellis object is invisibly returned.

simpleRegPlot.lmfm *Scatter Plot with Overlaid Fits*

Description

Produces a scatter plot of the data with overlaid fits.

Usage

```
simpleRegPlot.lmfm(x, lwd.reg, col.reg, ...)
```

Arguments

x	an lmfm object.
lwd.reg	a vector with length equal to the number of fitted models in x specifying the line widths used in the plot. See par for possible values.
col.reg	a vector with length equal to the number of fitted models in x specifying the line colors used in the plot. See par for possible values.
...	additional arguments are passed to xyplot .

Value

the trellis object is invisibly returned.

`summary.glmfm`*Comparison Summaries for Generalized Linear Models*

Description

Compute a summary of each model in a `glmfm` object.

Usage

```
## S3 method for class 'glmfm'  
summary(object, correlation = FALSE, ...)
```

Arguments

<code>object</code>	a <code>glmfm</code> object.
<code>correlation</code>	a logical value. If TRUE, correlation matrices of the coefficient estimates are included in each summary.
<code>...</code>	additional arguments required by the generic summary function.

Value

a list with class `summary.glmfm` whose elements are summaries of each model in `object`.

Examples

```
# From ?glm:  
# A Gamma example, from McCullagh & Nelder (1989, pp. 300-2)  
  
clotting <- data.frame(  
  u = c(5,10,15,20,30,40,60,80,100),  
  lot1 = c(118,58,42,35,27,25,21,19,18),  
  lot2 = c(69,35,26,21,18,16,13,12,12))  
  
lot1 <- glm(lot1 ~ log(u), data = clotting, family = Gamma)  
lot2 <- glm(lot2 ~ log(u), data = clotting, family = Gamma)  
  
fm <- fit.models(lot1, lot2)  
summary(fm)
```

`summary.lmfm`*Comparison Summaries for Linear Regression Models*

Description

Compute a summary of each model in an `lmfm` object.

Usage

```
## S3 method for class 'lmfm'  
summary(object, correlation = FALSE, ...)
```

Arguments

<code>object</code>	an <code>lmfm</code> object.
<code>correlation</code>	a logical value. If <code>TRUE</code> , the correlation matrices for the coefficients are included in the summaries.
<code>...</code>	additional arguments required by the generic <code>summary</code> function.

Value

a list with class `summary.lmfm` whose elements are summaries of each model in `object`.

Examples

```
data(stackloss)  
m1 <- lm(stack.loss ~ ., data = stackloss)  
m2 <- lm(stack.loss ~ . - Acid.Conc., data = stackloss)  
  
fm <- fit.models(m1, m2)  
print(fm.sum <- summary(fm))
```

Index

*Topic **hplot**

- [indexPlot.lmf](#), 5
- [kernDenPlot.lmf](#), 6
- [overlaidKernDenPlot.lmf](#), 6
- [overlaidQQPlot.lmf](#), 7
- [plot.glmf](#), 7
- [plot.lmf](#), 8
- [qqPlot.lmf](#), 10
- [scatterPlot.lmf](#), 10
- [simpleRegPlot.lmf](#), 11

*Topic **methods**

- [designMD](#), 2
- [plot.glmf](#), 7
- [plot.lmf](#), 8
- [summary.glmf](#), 12
- [summary.lmf](#), 13

*Topic **misc**

- [fmclass.register](#), 4

*Topic **models**

- [fit.models](#), 3

*Topic **regression**

- [designMD](#), 2
- [summary.glmf](#), 12
- [summary.lmf](#), 13

[densityplot](#), 6

[designMD](#), 2

[fit.models](#), 3

[fmclass.add.class](#), 4

[fmclass.add.class\(fmclass.register\)](#), 4

[fmclass.register](#), 4, 4

[indexPlot.lmf](#), 5, 9

[kernDenPlot.lmf](#), 6, 9

[model.matrix](#), 2

[overlaidKernDenPlot.lmf](#), 6, 9

[overlaidQQPlot.lmf](#), 7, 9

[par](#), 11

[plot.glmf](#), 7

[plot.lmf](#), 8

[qqmath](#), 7

[qqPlot.lmf](#), 8, 9, 10

[scatterPlot.lmf](#), 8, 9, 10

[simpleRegPlot.lmf](#), 9, 11

[summary](#), 12, 13

[summary.glmf](#), 12

[summary.lmf](#), 13

[xyplot](#), 5, 6, 10, 11