Package ‘fit.models’

February 19, 2015

Version 0.5-10
Date 2013-02-20
Title fit.models
Description A framework for comparing fitted models
Author Kjell Konis <kjell.konis@me.com>
Maintainer Kjell Konis <kjell.konis@me.com>
Depends R (>= 2.15.0), lattice
Suggests MASS
License GPL
Repository CRAN
Repository/R-Forge/Project robust
Repository/R-Forge/Revision 110
Repository/R-Forge/DateTimeStamp 2013-02-20 22:04:25
Date/Publication 2013-02-23 08:06:07
NeedsCompilation no

R topics documented:

  fit.models .................................................. 2
  fmregistry ................................................ 4
  indexPlot.lmfm ........................................... 4
  kernDenPlot.lmfm ......................................... 5
  leverage .................................................. 6
  overlaidKernDenPlot.lmfm ............................... 7
  overlaidQQPlot.lmfm .................................... 7
  plot.glmfm ............................................... 8
  plot.lmfm ................................................ 9
  qqPlot.lmfm ............................................... 10
  rmodified ............................................... 11
  scatterPlot.lmfm ....................................... 12
  simpleRegPlot.lmfm ..................................... 13
  summary.glmfm ......................................... 13
  summary.lmfm ............................................ 14
Description

Fit a statistical model using different estimators (e.g., robust and least-squares) and/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

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

... 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.

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

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

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

in the first element and

```r
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 already 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 most 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

data(stackloss)

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

# Step 1: rlm (robust linear model) is in the MASS package.
require(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
plot(fm1)

# Second, use `fit.models` to combine already 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)
### fmregistry

**Register Comparable Functions**

**Description**

The fit.models package maintains a list of comparable models called the fit.models registry. These functions provide a method for adding new classes of models to the fit.models registry.

**Usage**

```r
fmclass.register(fmclass, classes, validation.function = null)
fmclass.add.class(fmclass, class, warn = TRUE)
```

**Arguments**

- `fmclass`: a character string naming a fit.models class to be added to the fit.models registry.
- `classes`: a character vector naming one or more classes that can be compared by the methods defined for the fit.models class in `fmclass`.
- `class`: a character string specifying a class compatible with the methods of `fmclass`.
- `validation.function`: a function returning `TRUE` when the models are comparable.
- `warn`: a logical value. If TRUE, a warning is printed if `class` is already registered in the fit.models registry.

**Details**

See the package vignette.

**Value**

A null value is invisibly returned.

### indexPlot.lmfm

**Comparison Index (Time) Plot**

**Description**

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

**Usage**

```r
indexPlot.lmfm(x, fun, level = 0.95, id.n = 3, ...)
```
\textbf{kerndenPlot.lmfm}

\textit{Comparison Kernel Density Estimate Plot}

\textbf{Description}

Produces side-by-side kernel density estimate plots.

\textbf{Usage}

\texttt{kerndenPlot.lmfm(x, fun, \ldots)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \quad \text{an \texttt{lmfm} object.}
  \item \texttt{fun} \quad \text{a function to extract the appropriate quantity from \texttt{x}.}
  \item \texttt{level} \quad \text{a numeric value between 0 and 1 specifying the confidence level used to draw the threshold in the plot.}
  \item \texttt{id.n} \quad \text{a non-negative integer value specifying the number of extreme points to identify.}
  \item \ldots \quad \text{any additional arguments are passed to \texttt{xyplot}.}
\end{itemize}

\textbf{Value}

the \texttt{trellis} object is invisibly returned.
leverage  Leverage

Description

Returns a leverage measure for each sample point in the data.

Usage

leverage(object, ...)  
## Default S3 method:
leverage(object, ...)  
## S3 method for class 'lm'
leverage(object, ...)

Arguments

object  a fitted model object.
...
additional arguments are passed to the dispatched method.

Details

For least-squares linear models the leverages are the diagonal elements of the hat matrix. This function is generic. The default method assigns a leverage of $p/n$ to every sample point.

Value

a numeric vector containing the leverages.

Examples

## Example from lm:
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2,10,20, labels=c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)

leverage(lm.D9)
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.
Comparison Diagnostic Plots for Generalized Linear Models

Description

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

(2) Deviance Residuals vs. Predicted Values,
(3) Response vs. Fitted Values,
(4) Normal QQ Plot of Modified Pearson Residuals,
(5) Normal QQ Plot of Modified Deviance Residuals,
(6) Modified Pearson Residuals vs. Leverage,
(7) Scale-Location.

Usage

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

Arguments

- **x**: a `glmfm` object.
- **which.plot**: 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.
**plot.lmfm**

### Examples

```r
# 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(118,58,42,35,27,25,19,18),
    lot2 = c(69,35,26,21,18,16,13,12))

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)
```

### Description

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

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

### Usage

```r
## S3 method for class 'lmfm'
plot(x, which.plots = c(5, 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.
- **...**: other parameters to be passed through to plotting functions.
Details

The *modified residuals* are defined to be

\[ r_i = \frac{e_i}{\sqrt{1 - h_i}} \]

where \( h_i = H_{ii} \) is the \( i^{th} \) diagonal element of the hat matrix. The modified residuals are identically distributed with variance \( \sigma^2 \). The modified residuals are used instead of the standardized residuals (which are identically distributed with variance 1) so that the comparison plots emphasize differences in the variance estimates.

Value

\( x \) is invisibly returned.

Side Effects

The selected plots are drawn on a graphics device.

References


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. See `rmodified` for modified residuals.

Examples

```r
data(stackloss)
stackNlm <- lm(stackNloss ~ NL data = stackloss)
stackNclean <- lm(stackNloss ~ .., data = stacklossL subset = U:RP)
fm <- fitNmodels(stackNcleanL stackNlm)
plot(fm)
```

```r
qqPlot.lmfm
```

Description

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

```r
qqPlot.lmfm(x, fun, envelope = TRUE, half.normal = FALSE, n.samples = 250,
level = .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.

---

**rmodified**

*Extract Modified Residuals*

**Description**

Extract the *modified residuals*

\[
 r_i = e_i / \sqrt{1 - h_{ii}}
\]

from a fitted model object.

**Usage**

```r
rmodified(object, ...) // Default S3 method:
rmodified(object, ...)
```

**Arguments**

- `object`: a fitted model object containing a qr component and residuals.
- `...`: additional arguments are passed to the dispatched method.
Details

The modified residuals are rescaled to have variance $\sigma^2$ (as opposed to the standardized residuals which have variance 1). In the context of the fit.models package, the modified residuals are preferred since diagnostic plots of the modified residuals will emphasize differences between residual scale estimates.

The name modified residuals comes from model-based resampling (bootstrap). See, for example, section 6.6 in MASS.

Value

a numeric vector containing the modified residuals.

References


Examples

```r
## Example from lm:
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels=c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
rmmodified(lm.D9)
```

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

**Arguments**  
- **object**: a `glmfm` object.
- **correlation**: a logical value. If `TRUE`, correlation matrices of the coefficient estimates are included in each summary.
- **...**: additional arguments required by the generic `summary` function.
Value

a list with class summary.glmm 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)
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.lmfm, 4
  kernDenPlot.lmfm, 5
  overlaidKernDenPlot.lmfm, 7
  overlaidQQPlot.lmfm, 7
  plot.glmfm, 8
  plot.lmfm, 9
  qqPlot.lmfm, 10
  scatterPlot.lmfm, 12
  simpleRegPlot.lmfm, 13
*Topic methods
  leverage, 6
  plot.glmfm, 8
  plot.lmfm, 9
  rmodified, 11
  summary.glmfm, 13
  summary.lmfm, 14
*Topic misc
  fmregistry, 4
*Topic models
  fit.models, 2
*Topic regression
  leverage, 6
  rmodified, 11
  summary.glmfm, 13
  summary.lmfm, 14

densityplot, 7
fit.models, 2
fmclass.add.class, 3
fmclass.add.class(fmregistry), 4
fmclass.register, 3
fmclass.register(fmregistry), 4
fmregistry, 4
indexPlot.lmfm, 4, 10
kernDenPlot.lmfm, 5, 10
leverage, 6

overlaidKernDenPlot.lmfm, 7, 10
overlaidQQPlot.lmfm, 7, 10
par, 13
plot.glmfm, 8
plot.lmfm, 9
qqmath, 7
qqPlot.lmfm, 8, 10, 10
rmodified, 10, 11
scatterPlot.lmfm, 8, 10, 12
simpleRegPlot.lmfm, 10, 13
summary, 13, 14
summary.glmfm, 13
summary.lmfm, 14

xypplot, 5, 11, 13