

Package ‘auditor’

September 19, 2018

Title Model Audit - Verification, Validation, and Error Analysis

Version 0.3.1

Description Provides an easy to use unified interface for creating validation plots for any model. The 'auditor' helps to avoid repetitive work consisting of writing code needed to create residual plots. This visualizations allow to asses and compare the goodness of fit, performance, and similarity of models.

Depends R (>= 3.0.0)

License GPL

Encoding UTF-8

LazyData true

Imports car, factoextra, fdrtool, GGally, ggplot2, ggrepel, gridExtra, hnp, plotROC, ROCR, tseries

RoxygenNote 6.1.0

Suggests aods3, breakDown, carData, DALEX, e1071, gamlss, knitr, lme4, MASS, mlbench, randomForest, rmarkdown, testthat, covr

VignetteBuilder knitr

URL <https://mi2datalab.github.io/auditor/>

BugReports <https://github.com/MI2DataLab/auditor/issues>

NeedsCompilation no

Author Alicja Gosiewska [aut, cre],
Przemyslaw Biecek [aut, ths]

Maintainer Alicja Gosiewska <alicjagosiewska@gmail.com>

Repository CRAN

Date/Publication 2018-09-19 09:50:03 UTC

R topics documented:

audit	3
auditorData	4
modelEvaluation	4
modelFit	5
modelPerformance	6
modelResiduals	6
observationInfluence	7
plot.modelAudit	8
plotACF	9
plotAutocorrelation	10
plotCooksDistance	10
plotHalfNormal	11
plotLIFT	12
plotModelCorrelation	13
plotModelPCA	14
plotModelRanking	15
plotPrediction	16
plotREC	17
plotResidual	18
plotResidualBoxplot	19
plotResidualDensity	19
plotROC	20
plotRROC	21
plotScaleLocation	22
plotTwoSidedECDF	23
score	24
scoreCooksDistance	25
scoreDW	26
scoreHalfNormal	26
scoreMAE	27
scoreMSE	28
scorePeak	28
scoreREC	29
scoreRMSE	30
scoreROC	31
scoreRROC	31
scoreRuns	32

audit	<i>Create modelAudit object</i>
-------	---------------------------------

Description

Function `audit` create `modelAudit` object for further validation of a model. Models may have very different structures. This function creates a unified representation of a model and calculates residuals, which can be further processed by various error analysis functions.

Usage

```
audit(object, data = NULL, y = NULL, predict.function = yhat,
      residual.function = NULL, label = NULL)
```

Arguments

<code>object</code>	An object containing a model or object of class <code>explainer</code> (see explain).
<code>data</code>	<code>Data.frame</code> or matrix - data that will be used by further validation functions. If not provided, will be extracted from the model.
<code>y</code>	Response vector that will be used by further validation functions. Some functions may require an integer vector containing binary labels with values 0,1. If not provided, will be extracted from the model.
<code>predict.function</code>	Function that takes two arguments: model and data. It should return a numeric vector with predictions.
<code>residual.function</code>	Function that takes three arguments: model, data and response vector. It should return a numeric vector with model residuals for given data. If not provided, response residuals ($y - \hat{y}$) are calculated.
<code>label</code>	Character - the name of the model. By default it's extracted from the <code>'class'</code> attribute of the model.

Value

An object of class `ModelAudit`, which contains: #'

- `model.class` class of the audited model,
- `label` the name of the model,
- `model` the audited model,
- `fitted.values` fitted values from model,
- `data` data used for fitting the model,
- `y` vector with values of predicted variable used for fitting the model,
- `predict.function` function that were used for model predictions,
- `residual.function` function that were used for calculating model residuals,
- `residuals`
- `std.residuals` standardized residuals - the residuals divided by theirs standard deviation.

Examples

```
library(MASS)
model.glm <- glm(Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian, data = anorexia)
audit.glm <- audit(model.glm)
```

```
p.fun <- function(model, data){predict(model, data, response = "link")}
audit.glm.newpred <- audit(model.glm, predict.function = p.fun)
```

```
library(randomForest)
model.rf <- randomForest(Species ~ ., data=iris)
audit.rf <- audit(model.rf)
```

auditorData

auditorData

Description

The auditor Data is an artificial data set. It consists of 2000 observations. First four of simulated variables are treated as continuous while the fifth one is categorical.

Examples

```
data("auditorData", package = "auditor")
head(auditorData)
```

modelEvaluation

Create Model Evaluation explainer

Description

Creates modelEvaluation object to be plotted. Model evaluation concentrates on classification models.

Usage

```
modelEvaluation(object, variable = NULL)
```

Arguments

object	An object of class ModelAudit.
variable	Optional. Name of variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the <code>audit</code> function).

Examples

```
library(mlbench)
data("PimaIndiansDiabetes")
Pima <- PimaIndiansDiabetes
Pima$diabetes <- ifelse(Pima$diabetes == "pos", 1, 0)
glm_model <- glm(diabetes~., family=binomial,data=Pima)
glm_au <- audit(glm_model, data = Pima, y = Pima$diabetes)

modelEvaluation(glm_au)
```

modelFit

Create Model Fit explainer

Description

Creates modelFit object to be plotted.

Usage

```
modelFit(object, quant.scale = FALSE, ...)
```

Arguments

object	An object of class ModelAudit.
quant.scale	if TRUE values on axis are on quantile scale.
...	other parameters passed do hnp function.

Examples

```
library(MASS)
model.glm <- glm(Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian, data = anorexia)
audit.glm <- audit(model.glm)

mf.glm <- modelFit(audit.glm)
```

modelPerformance *Create Model Performance Explainer*

Description

Creates observationInfluence object to be plotted.

Usage

```
modelPerformance(object, scores = c("MAE", "MSE", "REC", "RROC"),
  new.score = NULL)
```

Arguments

object	An object of class ModelAudit.
scores	Vector of score names to be plotted.
new.score	A named list of functions that take one argument: object of class ModelAudit and return a numeric value. The measure calculated by the function should have the property that lower score value indicates better model.
...	other parameters.

Examples

```
library(MASS)
model.glm <- glm(Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian, data = anorexia)
audit.glm <- audit(model.glm)

mp.glm <- modelPerformance(audit.glm)
```

modelResiduals *Create Model Residuals explainer*

Description

Creates modelResiduals object to be plotted.

Usage

```
modelResiduals(object, variable = NULL)
```

Arguments

object	An object of class ModelAudit.
variable	Optional. Name of variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the <code>audit</code> function).

Examples

```
library(MASS)
model.glm <- glm(Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian, data = anorexia)
audit.glm <- audit(model.glm)

mr.glm <- modelResiduals(audit.glm)
```

observationInfluence *Create Observation Influence Explainer*

Description

Creates observationInfluence object to be plotted.

Usage

```
observationInfluence(object, ...)
```

Arguments

object	An object of class ModelAudit.
...	other parameters.

Examples

```
library(MASS)
model.glm <- glm(Postwt ~ Prewt + Treat + offset(Prewt), family = gaussian, data = anorexia)
audit.glm <- audit(model.glm)

oi.glm <- observationInfluence(audit.glm)
```

plot.modelAudit *Model diagnostic plots*

Description

This function provides several diagnostic plots for regression and classification models.

Usage

```
## S3 method for class 'modelAudit'
plot(x, ..., type = "Residual", ask = TRUE,
     grid = TRUE)

## S3 method for class 'modelFit'
plot(x, ..., type = "Residual", ask = TRUE,
     grid = TRUE)

## S3 method for class 'modelPerformance'
plot(x, ..., type = "Residual", ask = TRUE,
     grid = TRUE)

## S3 method for class 'modelResiduals'
plot(x, ..., type = "Residual", ask = TRUE,
     grid = TRUE)

## S3 method for class 'observationInfluence'
plot(x, ..., type = "Residual",
     ask = TRUE, grid = TRUE)
```

Arguments

x	object of class modelAudit, modelResiduals or observationInfluence.
...	other arguments dependent on the type of plot or additional objects of class modelAudit
type	the type of plot. Possible values: 'ACF', 'Autocorrelation', 'CooksDistance', 'HalfNormal', 'Residuals', 'LIFT', 'ModelPCA', 'ModelRanking', 'ModelCorrelation', 'Prediction', 'REC', 'Residual', 'ResidualBoxplot', 'ResidualDensity', 'ROC', 'RROC', 'ScaleLocation', 'TwoSidedECDF' (for detailed description see functions in see also section).
ask	logical; if TRUE, the user is asked before each plot, see par (ask=).
grid	logical; if TRUE plots will be plotted on the grid.

See Also

[plotACF](#), [plotAutocorrelation](#), [plotCooksDistance](#), [plotHalfNormal](#), [plotResidual](#), [plotResidualBoxplot](#), [p](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plot(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plot(lm_au, rf_au, type = "ModelRanking")
```

plotACF

*Autocorrelation Function Plot***Description**

Plot Autocorrelation Function of models residuals.

Usage

```
plotACF(object, ..., variable = NULL, alpha = 0.95)
```

Arguments

object	An object of class modelAudit or modelResiduals.
...	Other modelAudit or modelResiduals objects to be plotted together.
variable	Only for modelAudit object. Name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).
alpha	Confidence level of the interval.

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotACF(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotACF(lm_au, rf_au)
```

plotAutocorrelation *Autocorrelation Plot*

Description

Plot of i-th residual vs i+1-th residual.

Usage

```
plotAutocorrelation(object, ..., variable = NULL, score = FALSE)
```

Arguments

object	An object of class modelAudit or modelResiduals.
...	Other modelAudit or modelResiduals objects to be plotted together.
variable	Only for modelAudit object. Name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).
score	Logical, if TRUE values of scoreDW and scoreRuns will be added to plot.

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotAutocorrelation(lm_au)
```

plotCooksDistance *Influence of observations plot*

Description

Cook's distances are used for estimate the influence of an single observation.

Usage

```
plotCooksDistance(object, nlabel = 3, ...)
```

Arguments

object	An object of class ModelAudit.
nlabel	Number of observations with the biggest Cook's distances to be labeled.
...	Other arguments passed to scoreCooksDistance .

Details

Cook's distance is a tool for identifying observations that may negatively affect the model. They may be also used for indicating regions of the design space where it would be good to obtain more observations. Data points indicated by Cook's distances are worth checking for validity.

Cook's Distances are calculated by removing the i -th observation from the data and recalculating the model. It shows how much all the values in the model change when the i -th observation is removed.

For model classes other than `lm` and `glm` the distances are computed directly from the definition.

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotCooksDistance(lm_au)
```

plotHalfNormal	<i>Half-Normal plot</i>
----------------	-------------------------

Description

The half-normal plot is one of the tools designed to evaluate the goodness of fit of a statistical models. It is a graphical method for comparing two probability distributions by plotting their quantiles against each other. Points on the plot correspond to ordered absolute values of model diagnostic (i.e. standardized residuals) plotted against theoretical order statistics from a half-normal distribution.

Usage

```
plotHalfNormal(object, score = TRUE, quant.scale = FALSE,
  xlab = "half-normal Quantiles", ylab = "residuals", main = "", ...)
```

Arguments

object	modelAudit object, modelFit object.
score	If TRUE score based on probability density function is displayed on the plot.
quant.scale	if TRUE values on axis are on quantile scale.
xlab	The text for the x axis.
ylab	The text for the y axis.
main	Title of plot.
...	extra arguments passed to hnp .

Value

An object of class `ggplot`

See Also[scoreHalfNormal](#)

`plotLIFT`*LIFT*

Description

LIFT is a plot of the rate of positive prediction against true positive rate for the different thresholds. It is useful for measuring and comparing the accuracy of the classifiers.

Usage`plotLIFT(object, ...)`**Arguments**

`object` An object of class `modelAudit` or `modelEvaluation`.
`...` Other `modelAudit` objects to be plotted together.

Value`ggplot` object**See Also**[plot.modelAudit](#)**Examples**

```
library(mlbench)
data("PimaIndiansDiabetes")
Pima <- PimaIndiansDiabetes
Pima$diabetes <- ifelse(Pima$diabetes == "pos", 1, 0)
glm_model <- glm(diabetes~., family=binomial,data=Pima)
glm_au <- audit(glm_model, data = Pima, y = Pima$diabetes)
plotLIFT(glm_au)
```

plotModelCorrelation *Model Correlation Plot*

Description

Matrix of plots

Usage

```
plotModelCorrelation(object, ..., values = "Fitted values")
```

Arguments

object	An object of class modelAudit or modelResiduals.
...	Other modelAudit or modelResiduals objects to be plotted together.
values	"Fitted values" or "Predicted response" for model fitted values or "Residuals" for residual values.

Value

ggplot object

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotModelCorrelation(lm_au, rf_au)
```

`plotModelPCA`*Principal Component Analysis of models*

Description

Principal Component Analysis of models residuals. PCA can be used to assess the similarity of the models.

Usage

```
plotModelPCA(object, ..., scale = TRUE, invisible = "none")
```

Arguments

<code>object</code>	An object of class <code>modelAudit</code> or <code>modelResiduals</code> ,
<code>...</code>	Other <code>modelAudit</code> or <code>modelResiduals</code> objects to be plotted together.
<code>scale</code>	A logical value indicating whether the models residuals should be scaled before the analysis.
<code>invisible</code>	A text specifying the elements to be hidden on the plot. Default value is "none". Allowed values are "model", "observ".

Value

ggplot object

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotModelPCA(lm_au, rf_au)
```

plotModelRanking	<i>Model Ranking Plot</i>
------------------	---------------------------

Description

Radar plot with model scores. Scores are scaled to [0,1], each score is inverted and divided by maximum score value.

Usage

```
plotModelRanking(object, ..., scores = c("MAE", "MSE", "REC", "RROC"),
  new.score = NULL, table = TRUE)
```

Arguments

object	An object of class ModelAudit.
...	Other modelAudit objects to be plotted together.
scores	Vector of score names to be plotted.
new.score	A named list of functions that take one argument: object of class ModelAudit and return a numeric value. The measure calculated by the function should have the property that lower score value indicates better model.
table	Logical. Specifies if table with score values should be plotted

Value

ggplot object

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotModelRanking(lm_au, rf_au)
```

plotPrediction *Predicted response vs Observed or Variable Values*

Description

Plot of predicted response vs observed or variable Values.

Usage

```
plotPrediction(object, ..., variable = NULL, smooth = FALSE,
               abline = TRUE, split = "none")
```

Arguments

object	An object of class modelAudit or modelResiduals.
...	Other modelAudit or modelResiduals objects to be plotted together.
variable	Only for modelAudit objects. Name of model variable to order residuals. If value is NULL data order is taken. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).
smooth	Logical, indicates whenever smooth line should be added.
abline	Logical, indicates whenever function y=x should be added.
split	Character. If "model" plot will be splitted by model.

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotPrediction(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotPrediction(lm_au, rf_au)
```

plotREC	<i>Regression Error Characteristic Curves (REC)</i>
---------	---

Description

Error Characteristic curves are a generalization of ROC curves. On the x axis of the plot there is an error tolerance and on the y axis there is a percentage of observations predicted within the given tolerance.

Usage

```
plotREC(object, ...)
```

Arguments

object	An object of class ModelAudit or modelResiduals.
...	Other modelAudit or model Residuals objects to be plotted together.

Details

REC curve estimates the Cumulative Distribution Function (CDF) of the error
Area Over the REC Curve (REC) is a biased estimate of the expected error

Value

ggplot object

References

Bi J., Bennett K.P. (2003). Regression error characteristic curves, in: Twentieth International Conference on Machine Learning (ICML-2003), Washington, DC.

See Also

[plot.modelAudit](#), [plotROC](#), [plotRRROC](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotREC(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotREC(lm_au, rf_au)
```

`plotResidual`*Plot Residuals vs Observed, Fitted or Variable Values*

Description

A plot of residuals against fitted values, observed values or any variable.

Usage

```
plotResidual(object, ..., variable = NULL, points = TRUE,
             lines = FALSE, std.residuals = FALSE, nlabel = 0)
```

Arguments

<code>object</code>	An object of class <code>modelAudit</code> or <code>modelResiduals</code> .
<code>...</code>	Other <code>modelAudit</code> objects to be plotted together.
<code>variable</code>	Only for <code>modelAudit</code> object. Name of model variable to order residuals. If value is <code>NULL</code> data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).
<code>points</code>	Logical, indicates whenever observations should be added as points.
<code>lines</code>	Logical, indicates whenever smoothed lines should be added.
<code>std.residuals</code>	Logical, indicates whenever standardized residuals should be used.
<code>nlabel</code>	Number of observations with the biggest Cook's distances to be labeled.

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotResidual(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotResidual(lm_au, rf_au)
```

plotResidualBoxplot *Plot Boxplots of Residuals*

Description

A plot of residuals.

Usage

```
plotResidualBoxplot(object, ...)
```

Arguments

object An object of class modelAudit.
... Other modelAudit objects to be plotted together.

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotResidual(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotResidualBoxplot(lm_au, rf_au)
```

plotResidualDensity *Residual Density Plot*

Description

Density of model residuals.

Usage

```
plotResidualDensity(object, ..., split.var = TRUE, variable = NULL)
```

Arguments

object	An object of class ModelAudit.
...	Other modelAudit objects to be plotted together.
split.var	Logical. Indicates whenever plot should be splitted by variable.
variable	variable name o split. Optional. Should be provided only for modelAudit object.

Value

ggplot object

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotResidualDensity(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotResidualDensity(lm_au, rf_au)
```

plotROC

Receiver Operating Characteristic (ROC)

Description

Receiver Operating Characteristic Curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) for the different thresholds. It is useful for measuring and comparing the accuracy of the classifiers.

Usage

```
plotROC(object, ...)
```

Arguments

object	An object of class ModelAudit or modelEvaluation.
...	Other modelAudit objects to be plotted together.

Value

ggplot object

See Also

[plot.modelAudit](#), [plotRROC](#), [plotREC](#)

Examples

```
library(mlbench)
data("PimaIndiansDiabetes")
Pima <- PimaIndiansDiabetes
Pima$diabetes <- ifelse(Pima$diabetes == "pos", 1, 0)
glm_model <- glm(diabetes~., family=binomial,data=Pima)
glm_au <- audit(glm_model, data = Pima, y = Pima$diabetes)
plotROC(glm_au)
```

plotRROC

Regression Receiver Operating Characteristic (RROC)

Description

The basic idea of the ROC curves for regression is to show model asymmetry. The RROC is a plot where on the x-axis we depict total over-estimation and on the y-axis total under-estimation.

Usage

```
plotRROC(object, ...)
```

Arguments

`object` An object of class `modelAudit` or `modelResiduals`.
`...` Other `modelAudit` or `modelResiduals` objects to be plotted together.

Details

For RROC curves we use a shift, which is an equivalent to the threshold for ROC curves. For each observation we calculate new prediction: $\hat{y}' = \hat{y} + s$ where s is the shift. Therefore, there are different error values for each shift: $e_i = \hat{y}'_i - y_i$

Over-estimation is calculated as: $OVER = \sum(e_i | e_i > 0)$.

Under-estimation is calculated as: $UNDER = \sum(e_i | e_i < 0)$.

The shift equals 0 is represented by a dot.

The Area Over the RROC Curve (AOC) equals to the variance of the errors multiplied by $frac{n^2}{2}$.

Value

ggplot object

References

Hernández-Orallo, José. 2013. 'ROC Curves for Regression'. Pattern Recognition 46 (12): 3395–3411.

See Also

[plot.modelAudit](#), [plotROC](#), [plotREC](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotRRROC(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotRRROC(lm_au, rf_au)
```

plotScaleLocation *Scale location plot*

Description

Variable values vs square root of the absolute value of the residuals. A vertical line corresponds to median.

Usage

```
plotScaleLocation(object, ..., variable = NULL, score = FALSE,
  peaks = TRUE)
```

Arguments

object	An object of class modelAudit or modelResiduals.
...	Other modelAudit objects to be plotted together.
variable	Only for modelAudit object. Name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).
score	A logical value. If TRUE value of scorePeak will be added.
peaks	A logical value. If TRUE peaks are marked on plot by black dots.

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotScaleLocation(lm_au)
```

plotTwoSidedECDF *Two-sided Cumulative Distribution Function*

Description

Cumulative Distribution Function for positive and negative residuals.

Usage

```
plotTwoSidedECDF(object, ..., error.scaled = TRUE, outliers = NA,
  residuals = TRUE, y.reversed = FALSE)
```

Arguments

object	An object of class ModelAudit.
...	Other modelAudit objects to be plotted together.
error.scaled	A logical value indicating whether ECDF should be scaled by proportions of positive and negative proportions.
outliers	Number of outliers to be marked.
residuals	A logical value indicating whether residuals should be marked.
y.reversed	A logical value indicating whether values on y axis should be reversed.

Value

ggplot object

See Also

[plot.modelAudit](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
plotTwoSidedECDF(lm_au)

library(randomForest)
rf_model <- randomForest(prestige~education + women + income, data = Prestige)
```

```
rf_au <- audit(rf_model, data = Prestige, y = Prestige$prestige)
plotTwoSidedECDF(lm_au, rf_au, y.reversed = TRUE)
```

score

Model Scores computations

Description

This function provides several scores for model validation and performance assessment. Scores can be also used to compare models.

Usage

```
score(object, type = "MSE", ...)
```

Arguments

object	Object An object of class modelAudit.
type	The type of score to be calculated. Possible values: 'Cook', 'DW', 'Peak', 'HalfNormal', 'MAE', 'MSE', 'REC', 'RMSE', 'ROC', 'RROC', 'Runs' (for detailed description see functions in see also section).
...	Other arguments dependent on the type of score.

Value

an object of class scoreAudit, except Cooks distance, where numeric vector is returned

See Also

[scoreCooksDistance](#), [scoreDW](#), [scorePeak](#), [scoreHalfNormal](#), [scoreMAE](#), [scoreMSE](#), [scoreREC](#), [scoreROC](#), [scoreRROC](#), [scoreRuns](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
score(lm_au, type = 'Runs')
```

scoreCooksDistance *Score based on Cooks Distance*

Description

Cook's distance are used for estimate of the influence of an single observation.

Usage

```
scoreCooksDistance(object, print = TRUE)
```

Arguments

object	An object of class ModelAudit.
print	If TRUE progress is printed.

Details

Cook's distance is a tool for identifying observations that may negatively affect the model. They may be also used for indicating regions of the design space where it would be good to obtain more observations. Data points indicated by Cook's distances are worth checking for validity.

Cook's Distances are calculated by removing the i -th observation from the data and recalculating the model. It shows how much all the values in the model change when the i -th observation is removed.

Models of classes other than `lm` and `glm` the distances are computed directly from the definition, so this may take a while.

Value

numeric vector

See Also

[score](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreCooksDistance(lm_au)
```

scoreDW	<i>Durbin-Watson Score</i>
---------	----------------------------

Description

Score based on Durbin-Watson test statistic. The score value is helpful in comparing models. It is worth pointing out that results of tests like p-value makes sense only when the test assumptions are satisfied. Otherwise test statistic may be considered as a score.

Usage

```
scoreDW(object, variable = NULL)
```

Arguments

object	An object of class modelAudit or modelResiduals.
variable	Name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (<i>y</i> parameter passed to the audit function).

Value

an object of class scoreAudit

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreDW(lm_au)
```

scoreHalfNormal	<i>Half-Normal Score</i>
-----------------	--------------------------

Description

Score is approximately: $\sum \#[res_i \leq simres_{i,j}] - n$ with the distinction that each element of sum is also scaled to take values from [0,1].

res_i is a residual for *i*-th observation, $simres_{i,j}$ is the residual of *j*-th simulation for *i*-th observation, and *n* is the number of simulations for each observation. Scores are calculated on the basis of simulated data, so they may differ between function calls.

Usage

```
scoreHalfNormal(object, ...)
```

Arguments

```
object      modelAudit or modelFit object.  
...        Extra arguments passed to hnp.
```

Examples

```
library(car)  
lm_model <- lm(prestige~education + women + income, data = Prestige)  
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)  
plotHalfNormal(lm_au)
```

scoreMAE	<i>Mean Absolute Error</i>
----------	----------------------------

Description

Mean Absolute Error.

Usage

```
scoreMAE(object)
```

Arguments

```
object      An object of class ModelAudit or modelResiduals.
```

Value

an object of class scoreAudit

See Also

[score](#)

Examples

```
library(car)  
lm_model <- lm(prestige~education + women + income, data = Prestige)  
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)  
scoreMAE(lm_au)
```

scoreMSE	<i>Mean Square Error</i>
----------	--------------------------

Description

Mean Square Error.

Usage

```
scoreMSE(object)
```

Arguments

object An object of class modelAudit or modelResiduals.

Value

an object of class scoreAudit

See Also

[score](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreMSE(lm_au)
```

scorePeak	<i>Peak Score</i>
-----------	-------------------

Description

This score is calculated on the basis of Peak test, which is used for checking for homoscedasticity of residuals in regression analyses.

Usage

```
scorePeak(object, variable = NULL)
```

Arguments

object	Object An object of class modelAudit or modelResidual.
variable	Name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the audit function).

Value

an object of class scoreAudit

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scorePeak(lm_au)
```

scoreREC

Area Over the Curve for REC Curves

Description

The area over the Regression Error Characteristic curve is a measure of the expected error for the regression model.

Usage

```
scoreREC(object)
```

Arguments

object	An object of class ModelAudit.
--------	--------------------------------

Value

an object of class scoreAudit

References

J. Bi, and K. P. Bennet, "Regression error characteristic curves," in Proc. 20th Int. Conf. Machine Learning, Washington DC, 2003, pp. 43-50

See Also

[plotREC](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreREC(lm_au)
```

scoreRMSE

Root Mean Square Error

Description

Root Mean Square Error.

Usage

```
scoreRMSE(object)
```

Arguments

object An object of class ModelAudit.

Value

an object of class scoreAudit

See Also

[score](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreRMSE(lm_au)
```

scoreROC	<i>Area Under ROC Curve (AUC)</i>
----------	-----------------------------------

Description

Area Under Curve (AUC) for Receiver Operating Characteristic.

Usage

```
scoreROC(object)
```

Arguments

object An object of class ModelAudit.

Value

an object of class scoreAudit

See Also

[plotROC](#)

Examples

```
library(mlbench)
data("PimaIndiansDiabetes")
Pima <- PimaIndiansDiabetes
Pima$diabetes <- ifelse(Pima$diabetes == "pos", 1, 0)
glm_model <- glm(diabetes~., family=binomial,data=Pima)
glm_au <- audit(glm_model, data = Pima, y = Pima$diabetes)
scoreROC(glm_au)
```

scoreRROC	<i>Area Over the Curve for RROC Curves</i>
-----------	--

Description

The area over the Regression Receiver Operating Characteristic.

Usage

```
scoreRROC(object)
```

Arguments

object An object of class ModelAudit.

Value

an object of class scoreAudit

References

Hernández-Orallo, José. 2013. 'ROC Curves for Regression'. Pattern Recognition 46 (12): 3395–3411.

See Also

[plotRROC](#)

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreRROC(lm_au)
```

scoreRuns

Runs Score

Description

Score based on Runs test statistic. Note that this test is not very strong. It utilizes only signs of the residuals. The score value is helpful in comparing models. It is worth pointing out that results of tests like p-value makes sense only when the test assumptions are satisfied. Otherwise test statistic may be considered as a score.

Usage

```
scoreRuns(object, variable = NULL)
```

Arguments

object object An object of class ModelAudit.

variable name of model variable to order residuals. If value is NULL data order is taken. If value is "Predicted response" or "Fitted values" then data is ordered by fitted values. If value is "Observed response" the data is ordered by a vector of actual response (y parameter passed to the [audit](#) function).

Examples

```
library(car)
lm_model <- lm(prestige~education + women + income, data = Prestige)
lm_au <- audit(lm_model, data = Prestige, y = Prestige$prestige)
scoreRuns(lm_au)
```

Index

audit, [3](#), [4](#), [7](#), [9](#), [10](#), [16](#), [18](#), [22](#), [26](#), [29](#), [32](#)
auditorData, [4](#)

explain, [3](#)

hnp, [5](#), [11](#), [27](#)

modelEvaluation, [4](#)
modelFit, [5](#)
modelPerformance, [6](#)
modelResiduals, [6](#)

observationInfluence, [7](#)

par, [8](#)
plot.modelAudit, [8](#), [12–23](#)
plot.modelFit (plot.modelAudit), [8](#)
plot.modelPerformance
 (plot.modelAudit), [8](#)
plot.modelResiduals (plot.modelAudit), [8](#)
plot.observationInfluence
 (plot.modelAudit), [8](#)
plotACF, [8](#), [9](#)
plotAutocorrelation, [8](#), [10](#)
plotCooksDistance, [8](#), [10](#)
plotHalfNormal, [8](#), [11](#)
plotLIFT, [8](#), [12](#)
plotModelCorrelation, [8](#), [13](#)
plotModelPCA, [8](#), [14](#)
plotModelRanking, [8](#), [15](#)
plotPrediction, [8](#), [16](#)
plotREC, [8](#), [17](#), [21](#), [22](#), [29](#)
plotResidual, [8](#), [18](#)
plotResidualBoxplot, [8](#), [19](#)
plotResidualDensity, [8](#), [19](#)
plotROC, [8](#), [17](#), [20](#), [22](#), [31](#)
plotRROC, [8](#), [17](#), [21](#), [21](#), [32](#)
plotScaleLocation, [8](#), [22](#)
plotTwoSidedECDF, [8](#), [23](#)

score, [24](#), [25](#), [27](#), [28](#), [30](#)
scoreCooksDistance, [10](#), [24](#), [25](#)
scoreDW, [10](#), [24](#), [26](#)
scoreHalfNormal, [12](#), [24](#), [26](#)
scoreMAE, [24](#), [27](#)
scoreMSE, [24](#), [28](#)
scorePeak, [22](#), [24](#), [28](#)
scoreREC, [24](#), [29](#)
scoreRMSE, [30](#)
scoreROC, [24](#), [31](#)
scoreRROC, [24](#), [31](#)
scoreRuns, [10](#), [24](#), [32](#)