Package 'JOP'

February 19, 2015

Type Package

Title Joint Optimization Plot

Version 3.6

Date 2013-08-16

Author Sonja Kuhnt and Nikolaus Rudak

Maintainer Nikolaus Rudak <rudak@statistik.tu-dortmund.de>

Depends Rsolnp, dglm

Description JOP is a tool for simultaneous optimization of multiple responses and visualization of the results. The visualization is done by the joint optimization plot introduced by Kuhnt and Erdbruegge (2004).

LazyData Yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-16 10:17:21

R topics documented:

datax 2
datay 2
JOP 3
locate 5
plot.JOP 7

Index **9**

`datax` *Examplary dataset containing parameter settings*

Description

dataset coming from a sheet metal forming process

Usage

```
data(datax)
```

Format

Table containing parameter settings, X1 and X2

References

Sonja Kuhnt and Martina Erdbruegge (2004). A strategy of robust paramater design for multiple responses, Statistical Modelling; 4: 249-264, TU Dortmund.

`datay` *Examplary dataset containing Responses*

Description

dataset coming from a sheet metal forming process

Usage

```
data(datay)
```

Format

Table containing responses, Y1 and Y2.

References

Sonja Kuhnt and Martina Erdbruegge (2004). A strategy of robust paramater design for multiple responses, Statistical Modelling; 4: 249-264, TU Dortmund.

JOP — *Main function to minimize the risc function of a sequence of cost matrices*

Description

JOP calculates optimal design parameters associated with a given sequence of cost matrices based on the minimization of a risk function introduced by Pignatiello (1993). Furthermore JOP visualizes the optimal design parameters and the appropriate predicted responses using the joint optimization plot introduced by Kuhnt and Erdbruegge (2004).

Usage

```
JOP(datax, datay, tau = "min", Wstart = -5, Wend = 5, numbW = 10, d = NULL,
optreg = "sphere", Domain = NULL, form.mean = NULL, form.disp = NULL,
family.mean = gaussian(), dlink = "log", mean.model = NULL, var.model = NULL,
joplot = FALSE, solver = "solnp")
```

Arguments

`datax`	data set with parameter settings from an experimental design (data.frame). Columns have to be named.
`datay`	data set with responses resulting from an experimental design (data.frame). Columns have to be named.
`tau`	list of target values or single character value for the corresponding responses, where also "min" for minimization or "max" for maximization is possible. If `tau="min"` or `tau="max"`, then all responses are minimized or maximized.
`Wstart`	value to calculate the sequence of weight matrices (see Details)
`Wend`	value to calculate the sequence of weight matrices (see Details)
`numbW`	value to calculate the sequence of weight matrices (see Details)
`d`	a vector with values to calculate the sequence of weight matrices (see Details)
`optreg`	User can choose the Optimization region. `optreg="box"`: box constraints `optreg="sphere"`: sphere
`Domain`	box constraints. Column 1 for lower contraints and Column 2 for upper contraints. Row i corresponds to Parameter i.
`form.mean`	list of formulas for mean of each response
`form.disp`	list of formulas for dispersion of each response
`family.mean`	`family` for the mean
`dlink`	`list of names of link functions for the dispersion models`
`mean.model`	list of functions that model the mean for the corresponding response
`var.model`	list of functions that model the variance for the corresponding response
`joplot`	logical, if TRUE then the joint optimization plot is displayed.
`solver`	Default is "solnp" for three different starting points. Alternatively, "gosolnp" is especially recommended for complex programs.

Details

The main function `JOP` is a package for multiresponse optimization which aims to minimize a risc function for a prespecified sequence of cost matrices. This sequence of cost matrices is specified by the arguments Wstart, Wend, numbW and d. The user can plug in target values for the responses or set to the target value to "min" or "max" in order to minimize or maximize the corresponding response.

JOP needs models for the mean and dispersion of each response which can be plugged in by means three different possibilities.

- First, the user can pass the models for mean and dispersion as lists of functions in the parameter vector through the arguments var.model and mean.model.
- Secondly, the user can plug in a list of formulas for each response for the mean and dispersion via the arguments form.mean and form.disp. Furthermore, a suitable link and distribution assumption can be specified both for the mean and dispersion
- Finally, if the user does not plug in neither formulas nor models then JOP calculates automatically double generalized linear models by means of the function `dglm` from package dglm. Furthermore, JOP performs a backward selection, starting from the full model with main effects, interactions and quadratic terms, and afterwards dropping the least significant covariate in each step.

The data sets `datax` and `datay` are needed for model building. Both `datax` and `datay` have to be data frames where `datax` contains an experimental design with settings for each parameter columnwise and `datay` contains the experimental results columnwise for every response. Additionally, the columns of the data sets should be named, as exemplary demonstrated by data(datax) and data(datay). The optimization is performed by the procedure `solnp` out of the package Rsolnp. JOP returns an object of class "JOP" which can be visualized by means of `plot.JOP`. Details on the JOP method can be found in Erdbruegge et al. (2011).

Value

JOP returns a list containing the following elements:

`Parameters` The i-th row of this matrix contains the optimal Parameter setting appropriate to the i-th weight matrix

`Responses` The i-th row of this matrix contains the predicted Responses appropriate to the i-th weight matrix

`StandardDeviation` The i-th row of this matrix contains the standard deviation value for each response

`OptimalValue` This vector contains the optimal value of the risk function for each optimal parameter setting

`TargetValueJOP` Contains the target values for the correspoding responses used internally by JOP

`TargetValueUSER` Contains the target values for the correspoding responses specified by the user

`DGLM` If no models assigned then the list DGLM contains the calculated models for the mean and dispersion for every response

`RiskminimalParameters`
: Parameters that minimize the squared sum of single risks among all calculated Parameters

`RiskminimalResponses`
: Responses for the risk minimal parameters

`ValW` Values for Wstart and Wend

`d` Slope vector

`numbW` Number of weight matrices

References

Sonja Kuhnt and Martina Erdbruegge (2004). A strategy of robust paramater design for multiple responses, Statistical Modelling; 4: 249-264, TU Dortmund.

Martina Erdbruegge, Sonja Kuhnt and Nikolaus Rudak (2011). Joint optimization of independent multiple responses based on loss functions, Quality and Reliability Engineering International 27, doi: 10.1002/qre.1229.

Joseph J. Pignatiello (1993). Strategies for robust multiresponse quality engineering, IIE Transactions 25, 5-15, Texas A M University.

Alexios Ghalanos and Stefan Theussl (2012). Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method. R package version 1.12.

Peter K Dunn and Gordon K Smyth (2012). dglm: Double generalized linear models, R package version 1.6.2.

Sonja Kuhnt, Nikolaus Rudak (2013). Simultaneous Optimization of Multiple Responses with the R Package JOP, Journal of Statistical Software, 54(9), 1-23, URL http://www.jstatsoft.org/v54/i09/.

Examples

```
# Example: Sheet metal hydroforming process
# Run JOP without Model specification

outtest <- JOP(datax = datax, datay = datay, tau = list(0, 0.05), numbW = 5, joplot = TRUE)
```

`locate` *Selection of a compromise*

Description

The function `locate` allows the user to choose a point as a good compromise on the right plot and `locate` returns the corresponding design parameters.

Usage

```
locate(x, ncom = 1 ,xlu = NULL ,no.col = FALSE ,standard = TRUE ,col = 1 ,lty = 1,
bty = "l" ,las = 1 ,adj = 0.5 ,cex = 1 ,cex.lab = 1 ,cex.axis = 1,
xlab = c("Stretch Vector" , "Stretch Vector"),
ylab = c("Parameter Setting" , "Predicted Response"),lwd=1,...)
```

Arguments

`x`	object from JOP
`ncom`	number of compromises that the user seeks to select, default is 1
`xlu`	The vector of x-coordinate that indicates where the user assumes a good compromise, see Details
`no.col`	If TRUE the plot will be gray scaled. Otherwise the plot will be coloured.
`standard`	If TRUE the standard deviations will be displayed on the right hand plot.
`col`	Graphical argument, see details.
`lty`	Graphical argument, see details.
`xlab`	Graphical argument, see details.
`ylab`	Graphical argument, see details.
`bty,las,cex,adj,cex.lab,cex.axis,lwd`	Graphical arguments
`...`	Further graphical arguments passed to `plot`.

Details

The function `locate` asks the user to choose a compromise on the right hand plot and it displays the nearest calculated points by means of vertical lines. Furthermore it returns the chosen responses together with the corresponding parameters.

Let nx be the number of parameters (number of columns of datax) and ny be the number of responses (number of columns of datay). Then col and lty must have length nx+ny. Otherwise predefined grey colors (for no.col=TRUE) or standard colors 1, 2, ..., nx+ny are used. The arguments xlab and ylab must have length two, where the first entry contains the label for x-axis and y-axis of the left hand plot and the second entry contains the label for x-axis and y-axis of the right hand plot. Additional graphical arguments can be plugged in.

Value

locate returns a list containing the following elements:

`ChosenResponses`
: selected responses by user

`ChosenParameters`
: corresponding selected parameters

Author(s)

Sonja Kuhnt and Nikolaus Rudak

References

Sonja Kuhnt and Martina Erdbruegge (2004). A strategy of robust paramater design for multiple responses, Statistical Modelling; 4: 249-264, TU Dortmund.

Martina Erdbruegge, Sonja Kuhnt and Nikolaus Rudak (2011). Joint optimization of independent multiple responses based on loss functions, Quality and Reliability Engineering International 27, doi: 10.1002/qre.1229.

Joseph J. Pignatiello (1993). Strategies for robust multiresponse quality engineering, IIE Transactions 25, 5-15, Texas A M University.

Alexios Ghalanos and Stefan Theussl (2012). Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method. R package version 1.12.

Peter K Dunn and Gordon K Smyth (2012). dglm: Double generalized linear models, R package version 1.6.2.

Sonja Kuhnt, Nikolaus Rudak (2013). Simultaneous Optimization of Multiple Responses with the R Package JOP, Journal of Statistical Software, 54(9), 1-23, URL http://www.jstatsoft.org/v54/i09/.

Examples

```
# Example: Sheet metal hydroforming process
outtest <- JOP(datax = datax, datay = datay, tau = list(0, 0.05), numbW = 5)

# Location of points
locate(outtest, xlu = c(3, 4))
```

`plot.JOP` *Displaying the Joint Optimization Plot*

Description

The function `plot.JOP` takes the output produced by `JOP` and returns the joint optimization plot.

Usage

```
## S3 method for class 'JOP'
plot(x, no.col = FALSE, standard = TRUE, col = 1, lty = 1, bty = "l",
las = 1 ,adj = 0.5 ,cex = 1 ,cex.lab = 1 ,cex.axis = 1,
xlab = c("Stretch Vector" , "Stretch Vector"),
ylab = c("Parameter Setting" , "Predicted Response"),lwd=1,...)
```

Arguments

`x`	object from JOP
`no.col`	If TRUE the plot will be gray scaled. Otherwise the plot will be coloured.
`standard`	If TRUE the standard deviations will be displayed on the right hand plot.
`col`	Graphical argument, see details.
`lty`	Graphical argument, see details.
`xlab`	Graphical argument, see details.
`ylab`	Graphical argument, see details.
`bty,las,cex,adj,cex.lab,cex.axis,lwd`	Graphical arguments
`...`	Further graphical arguments passed to `plot`.

Details

Let nx be the number of parameters (number of columns of datax) and ny be the number of responses (number of columns of datay). Then col and lty must have length nx+ny. Otherwise predefined grey colors (for no.col=TRUE) or standard colors 1, 2, ..., nx+ny are used. The arguments xlab and ylab must have length two, where the first entry contains the label for x-axis and y-axis of the left hand plot and the second entry contains the label for x-axis and y-axis of the right hand plot. Additional graphical arguments can be plugged in.

References

Sonja Kuhnt and Martina Erdbruegge (2004). A strategy of robust paramater design for multiple responses, Statistical Modelling; 4: 249-264, TU Dortmund.

Martina Erdbruegge, Sonja Kuhnt and Nikolaus Rudak (2011). Joint optimization of independent multiple responses based on loss functions, Quality and Reliability Engineering International 27, doi: 10.1002/qre.1229.

Joseph J. Pignatiello (1993). Strategies for robust multiresponse quality engineering, IIE Transactions 25, 5-15, Texas A M University.

Alexios Ghalanos and Stefan Theussl (2012). Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method. R package version 1.12.

Peter K Dunn and Gordon K Smyth (2012). dglm: Double generalized linear models, R package version 1.6.2.

Sonja Kuhnt, Nikolaus Rudak (2013). Simultaneous Optimization of Multiple Responses with the R Package JOP, Journal of Statistical Software, 54(9), 1-23, URL http://www.jstatsoft.org/v54/i09/.

Examples

```
# Example: Sheet metal hydroforming process
outtest <- JOP(datax = datax, datay = datay, tau = list(0 , 0.05), numbW = 5)

# Several graphical parameters can be plugged in
plot(outtest, col = 5:8)
```

Index

*Topic **datasets**
    datax, 2
    datay, 2

datax, 2
datay, 2
dglm, *4*

family, *3*

JOP, 3

locate, 5

plot, *6*, *7*
plot.JOP, *4*, 7

solnp, *4*