Package 'epr'

July 2, 2014

Type Package

Title Easy polynomial regression

Version 2.0

Date 2013-07-30

Author Emmanuel Arnhold

Maintainer Emmanuel Arnhold <emmanuelarnhold@yahoo.com.br>

Description The package performs analysis of polynomial regression in simple designs with quantitative treatments

Depends R (>= 2.13), car

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-07-30 21:36:47

R topics documented:

epr-package *Easy polynomial regression*

Description

The package performs analysis of polynomial regression in simple designs with quantitative treatments.

Details

Package:	epr
Type:	Package
Version:	2.0
Date:	2013-07-30
License:	GPL-2

Author(s)

Emmanuel Arnhold <emmanuelarnhold@yahoo.com.br>

References

KAPS, M. and LAMBERSON, W. R. Biostatistics for Animal Science: an introductory text. 2nd Edition. CABI Publishing, Wallingford, Oxfordshire, UK, 2009. 504p.

SAMPAIO, I. B. M. Estatistica aplicada a experimentacao animal. 3nd Edition. Belo Horizonte: Editora FEPMVZ, Fundacao de Ensino e Pesquisa em Medicina Veterinaria e Zootecnia, 2010. 264p.

Examples

```
# analysis in completely randomized design
data(data1)
r1=pr2(data1)
names(r1)
r1
r1[1]

# analysis in randomized block design
data(data2)
r2=pr2(data2, design=2)
r2

# analysis in latin square design
data(data3)
```

```
r3=pr2(data3, design=3)
r3

# analysis in several latin squares
data(data4)
r4=pr2(data4, design=4)
r4

# data
treatments=rep(c(0.5,1,1.5,2,2.5,3), c(3,3,3,3,3,3))
r1=rnorm(18,60,3)
r2=r1*1:18
r3=r1*18:1
r4=r1*c(c(1:10),10,10,10,10,10,10,10,10)
data6=data.frame(treatments,r1,r2,r3, r4)

# use the argument list = TRUE
pr2(data6, design=1, list=TRUE)

# graphs
regplot(data6,variable=1, poly=2)
regplot(data6,variable=2, poly=1)
regplot(data6,variable=3, poly=1)
regplot(data6,variable=4, poly=4)
regplot(data6,variable=4, poly=3)
```

`data1` *data1: Sampaio (2010): page 134*

Description

Quantitative treatments in completely randomized design.

Usage

```
data(data1)
```

Format

A data frame with 24 observations on the following 2 variables.

`treatment` a numeric vector

`gain` a numeric vector

References

SAMPAIO, I. B. M. Estatistica aplicada a experimentacao animal. 3nd Edition. Belo Horizonte: Editora FEPMVZ, Fundacao de Ensino e Pesquisa em Medicina Veterinaria e Zootecnia, 2010. 264p.

Examples

```
data(data1)
summary(data1)
```

`data2` *data2: Kaps and Lamberson (2009): page 434*

Description

Quantitative treatments in randomizad block design.

Usage

```
data(data2)
```

Format

A data frame with 25 observations on the following 3 variables.

`protein_level` a numeric vector

`litter` a factor with levels `l1` `l2` `l3` `l4` `l5`

`feed_conversion` a numeric vector

References

KAPS, M. and LAMBERSON, W. R. Biostatistics for Animal Science: an introductory text. 2nd Edition. CABI Publishing, Wallingford, Oxfordshire, UK, 2009. 504p.

Examples

```
data(data2)
summary(data2)
```

`data3` *data3: fictional example*

Description

Quantitative treatments in latin square design.

Usage

```
data(data3)
```

Format

A data frame with 25 observations on the following 4 variables.

`treatment` a numeric vector

`animal` a factor with levels `a1` `a2` `a3` `a4` `a5`

`period` a factor with levels `p1` `p2` `p3` `p4` `p5`

`milk_fat` a numeric vector

Examples

```
data(data3)
summary(data3)
```

`data4` *data4: fictional example*

Description

Quantitative treatments in several latin squares design.

Usage

```
data(data4)
```

Format

A data frame with 50 observations on the following 5 variables.

`treatment` a numeric vector

`square` a numeric vector

`animal` a factor with levels `a1` `a2` `a3` `a4` `a5`

`period` a factor with levels `p1` `p2` `p3` `p4` `p5`

`milk_fat` a numeric vector

Examples

```
data(data4)
summary(data4)
```

data5 *data5: fictional example*

Description

Quantitative treatments and three response variable.

Usage

```
data(data5)
```

Format

A data frame with 24 observations on the following 4 variables.

`treatments` a numeric vector

`variable1` a numeric vector

`variable2` a numeric vector

`variable3` a numeric vector

Examples

```
data(data5)
summary(data5)
```

pr1 *Analysis of polynomial regression*

Description

The function performs analysis of polynomial regression in simple designs with quantitative treatments. The function also performs model fits with plateaus (plateaus linear and quadratic).

Usage

```
pr1(data, plateau = FALSE, x.plateau = NULL)
```

Arguments

`data`	data is a data.frame The first column should contain the treatments (explanatory variable) and the remaining columns the response variables.
`plateau`	FALSE = function returns the linear and quadratic TRUE = function returns the linear, quadratic, linear.plateau and quadratic.plateau
`x.plateau`	NULL = starting value for the linear plateau will be the point of maximum (or minimum) of the quadratic equation

Value

Returns coefficients of the models, t test for coefficients, R squared, adjusted R squared, AIC, BIC and the maximum (or minimum) values of y and critical point of x.

Author(s)

Emmanuel Arnhold <emmanuelarnhold@yahoo.com.br>

See Also

lm, ea1(easyanova package), pr2, regplot

Examples

```
# data
data(data5)

# linear and quadratic models
results1=pr1(data5)
results1

# including plateaus models
results2=pr1(data5, plateau=TRUE)
results2
```

---

pr2 *Analysis of polynomial regression*

Description

The function performs analysis of polynomial regression in simple designs with quantitative treatments. This function performs analysis the lack of fit .

Usage

```
pr2(data, design = 1, list = FALSE, type = 2)
```

Arguments

data	data is a data.frame
	data frame with two columns, treatments and response (completely randomized design)
	data frame with three columns, treatments, blocks and response (randomized block design)
	data frame with four columns, treatments, rows, cols and response (latin square design)
	data frame with five columns, treatments, square, rows, cols and response (several latin squares)

design	1 = completely randomized design 2 = randomized block design 3 = latin square design 4 = several latin squares
list	FALSE = a single response variable TRUE = multivariable response
type	type is form of obtain sum of squares 1 = a sequential sum of squares 2 = a partial sum of squares

Details

The response and the treatments must be numeric. Other variables can be numeric or factors.

Value

Returns analysis of variance, models, t test for coefficients and R squared and adjusted R squared.

Author(s)

Emmanuel Arnhold <emmanuelarnhold@yahoo.com.br>

References

KAPS, M. and LAMBERSON, W. R. Biostatistics for Animal Science: an introductory text. 2nd Edition. CABI Publishing, Wallingford, Oxfordshire, UK, 2009. 504p.

SAMPAIO, I. B. M. Estatistica aplicada a experimentacao animal. 3nd Edition. Belo Horizonte: Editora FEPMVZ, Fundacao de Ensino e Pesquisa em Medicina Veterinaria e Zootecnia, 2010. 264p.

See Also

lm, lme(package nlme), ea1(package easyanova), pr1, regplot

Examples

```
# analysis in completely randomized design
data(data1)
r1=pr2(data1)
names(r1)
r1
r1[1]

# analysis in randomized block design
data(data2)
r2=pr2(data2, design=2)
r2

# analysis in latin square design
```

```
data(data3)
r3=pr2(data3, design=3)
r3

# analysis in several latin squares
data(data4)
r4=pr2(data4, design=4)
r4

# data
treatments=rep(c(0.5,1,1.5,2,2.5,3), c(3,3,3,3,3,3))
r1=rnorm(18,60,3)
r2=r1*1:18
r3=r1*18:1
r4=r1*c(c(1:10),10,10,10,10,10,10,10,10)
data6=data.frame(treatments,r1,r2,r3, r4)

# use the argument list = TRUE
pr2(data6, design=1, list=TRUE)

# graphs
regplot(data6,variable=1, poly=2)
regplot(data6,variable=2, poly=1)
regplot(data6,variable=3, poly=1)
regplot(data6,variable=4, poly=4)
regplot(data6,variable=4, poly=3)
```

`regplot` *Regression graphics*

Description

The function generates the scatter plot with the regression equation.

Usage

```
regplot(data, xlab = NULL, ylab = NULL, poly = 1, position = 6, colors = TRUE,
mean = TRUE, variable = 1, x.plateau = NULL)
```

Arguments

`data`	data is a data.frame the first column contain the explanatory variable the others columns contain the responses variables
`xlab`	name of variable x
`ylab`	name of variable y

poly	indicates which regression plot 1 = linear (default) 2 = quadratic 3 = linear.plateau 4 = quadratic.plateau
position	position of equation in the graph top=1 bottomright=2 bottom=3 bottomleft=4 left=5 topleft=6 (default) topright=7 right=8 center=9
colors	TRUE = the line is red (default) FALSE = the line is black
mean	TRUE = scatter plots with averages (default) FALSE = scatter plots with all data
variable	1 = second column of data.frame (default) 2 = third column of data.frame see examples
x.plateau	default is NULL = starting value for the linear plateau will be the point of maximum (or minimum) of the quadratic equation

Value

The function generates the scatter plot with the regression equation.

Author(s)

Emmanuel Arnhold <emmanuelarnhold@yahoo.com.br>

See Also

lm, lme, ea1(easyanova package), pr2, pr2, dplot(ds package)

Examples

```
# data
data(data5)

# first response variable
par(mfrow=c(2,2))
regplot(data5, "Variable X", "Variable Y")
```

```
regplot(data5, "Variable X", "Variable Y", poly=2)
regplot(data5, "Variable X", "Variable Y", poly=3)
regplot(data5, "Variable X", "Variable Y", poly=4)

# second response variable
par(mfrow=c(2,2))
regplot(data5, "Variable X", "Variable Y", variable=2)
regplot(data5, "Variable X", "Variable Y", variable=2, poly=2)
regplot(data5, "Variable X", "Variable Y", variable=2,  poly=3)
regplot(data5, "Variable X", "Variable Y", variable=2, poly=4)

# third response variable
par(mfrow=c(2,2))
regplot(data5, variable=3, colors=FALSE, position=4, mean=FALSE)
regplot(data5, variable=3, poly=2, mean=FALSE)
regplot(data5, variable=3,  poly=3, mean=FALSE)
regplot(data5, variable=3, poly=4, mean=FALSE)

# data
treatments=rep(c(0.5,1,1.5,2,2.5,3), c(3,3,3,3,3,3))
r1=rnorm(18,60,3)
r2=r1*1:18
r3=r1*18:1
r4=r1*c(c(1:10),10,10,10,10,10,10,10,10)
data6=data.frame(treatments,r1,r2,r3, r4)

# graphs
regplot(data6,variable=1, poly=2)
regplot(data6,variable=2, poly=1)
regplot(data6,variable=3, poly=1)
regplot(data6,variable=4, poly=4)
regplot(data6,variable=4, poly=3)
```

Index