Package ‘gamlss.demo’

February 19, 2015

Description  Demos for smoothing and gamlss.family distributions.
Title        Demos for GAMLSS
LazyLoad     yes
Version      4.3-1
Date         2015-02-04
Depends      R (>= 2.4.0), rpanel (>= 1.1-1), gamlss.dist, gamlss.tr, tcltk
Suggests     MASS
Author       Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>, Bob Rigby <<r.rigby@londonmet.ac.uk>>, Paul Eilers <<p.eilers@erasmusmc.nl>>, Brian Marx \{email\{bmarx@LSU.EDU\}}, Konstantinos Patras <<kostas.pateras@gmail.com>> with contributions from Larisa Kosidou.
Maintainer   Mikis Stasinopoulos <<d.stasinopoulos@londonmet.ac.uk>>
License      GPL-2 | GPL-3
URL          http://www.gamlss.org/
NeedsCompilation  no
Repository    CRAN
Date/Publication  2015-02-04 10:35:00

R topics documented:

demo.BSplines .............................................. 2
demo.LocalRegression ...................................... 3
demo.Locmean .............................................. 4
demo.NO ...................................................... 5
demoDist ..................................................... 8
demoLpolyS ................................................ 9
demoPsplines ............................................... 10
gamlss.demo .............................................. 11
Locmean ..................................................... 12

Index  15
Demos for smoothing techniques

Description
These are demos for teaching smoothing techniques to students

Usage

demo.BSplines()
demo.RandomWalk(y = NULL, ...)
demo.histSmo(y = NULL, ...)
demo.interpolateSmo(y = NULL, w = NULL, ...)
demo.PSsplines(y = NULL, x = NULL, ...)

Arguments

y for y variable if needed otherwise it is generated
w for weights if needed
x for explanatory variable if needed
... for adding parameters in the plot

Value
An rpanel plot

Author(s)
Paul Eilers <p.eilers@erasmusmc.nl>, Brian Marx <bmarx@LSU.EDU>, and Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN
**demo.LocalRegression**

**Examples**

demo.PSplines()

demo.LocalRegression

**Local Regression Smoothing**

**Description**

This function demonstrates some characteristics of local regression smoothing.

**Usage**

```r
demo.LocalRegression(y = NULL, x = NULL, span = 0.5,
                      position = trunc((n - 1)/2),
                      deg = 1)
LPOL(y, x, span = 0.5, position = trunc((n - 1)/2),
     w = rep(1, length(y)), deg = 1)
WLPLAN(y, x, sd = 0.5, position = trunc((n - 1)/2),
       w = rep(1, length(y)), deg = 1)
```

**Arguments**

- `y` The response variable
- `x` The explanatory variable
- `span` The smoothing parameters
- `sd` The standard deviation of a normal kernel used as smoothing parameter
- `position` The position of the target values in the x axis
- `w` weights
- `deg` The degree of the local polynomial

**Details**

The function `demo.LocalRegression` demonstrates some aspects of the Local (unweighed) polynomial regression. The functions `LPOL()` and `WLPLAN()` produce plots related to unweighed and weighted local polynomial regression respectively.

**Value**

All function produce plots.

**Author(s)**

Mikis Stasinopoulos
References

R Development Core Team (2010) tcltk package, CRAN.
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN

See Also

See also *demoDist.gamlss.demo*

Examples

demo.Locmean()
n <- 100
x <- seq(0, 1, length = n)*1.4
y <- 1.2 + .3*sin(5 * x) + rnorm(n) * 0.2
op <- par(mfrow=c(2,2))
LPOL(y,x, deg=0, position=5)
title("(a) moving average")
LPOL(y,x, deg=1, position=75)
title("(b) linear poly")
WL Polynomial(x, deg=2, position=30)
title("(c) quadratic poly")
WL Polynomial(x, deg=3, position= 50)
title("(b) cubic poly")
par(op)

---

demo.Locmean  Demos for local polynomial smoothing

Description

Those are four demos to show weighed and unweighed local mean and polynomial smoothing.

Usage

demo.Locmean(y = NULL, x = NULL, ...)
demo.Locpoly(y = NULL, x = NULL, ...)
demo.WLocpoly(y = NULL, x = NULL, ...)
demo.WLocmean(y = NULL, x = NULL, ...)

Arguments

y the response variable. If null it generates its own data
x explanatory variable
... for extra argument in the plot

Value

It produces an rpanel plot

Author(s)

Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>

References

Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN

See Also

demo.PSplines

Examples

demo.Locmean()

demo.NO  Demos for different gamlss.family distributions

Description

The demo functions for showing the gamlss.family distributions. The functions use the package rpanel.
Usage
demo.NO()
demo.LO()
demo.NO.LO()
demo.GU()
demo.RG()
demo.exGAUS()
demo.PE()
demo.PE.NO()
demo.TF()
demo.TF.NO()
demo.EGB2()
demo.GT()
demo.JSU()
demo.JSUo()
demo.NET()
demo.SHASH()
demo.SEP1()
demo.SEP2()
demo.SEP3()
demo.SEP4()
demo.ST1()
demo.ST2()
demo.ST3()
demo.ST4()
demo.ST5()
demo.EXP()
demo.GA()
demo.LOGNO()
demo.NO.LOGNO()
demo.IG()
demo.WEI()
demo.WEI2()
demo.WEI3()
demo.BCCG()
demo.GG()
demo.GIG()
demo.ZAGA()
demo.ZAIG()
demo.BCT()
demo.BCPE()
demo.GB2()
demo.EGB2()
demo.BE()
demo.BEo()
demo.GB1()
demo.GT()
An rpanel plot
Author(s)
Mikis Stasinopoulos <d.stasinopoulos@londonmet.ac.uk>, Bob Rigby <r.rigby@londonmet.ac.uk> with contribution from Larisa Kosidou.

References
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN


Examples

demo.NO()

demoDist

Interface for demonstrating the gamlss.family distributions

Description
The function demoDist is an tcltk interface for plotting all the available gamlss.family distributions.

Usage
demoDist()

Value
It creates a tcltk menu

Author(s)
Konstantinos Pateras <kostas.pateras@gmail.com>
References

R Development Core Team (2010) tcltk package, CRAN.
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN

Examples

## do not run
demoDist()

demoLpolyS

Demo for local polynomial fits

Description

It starts the gamlss local polynomial demos demos. It is an tcltk interface for using the local polynomial demos.

Usage

demoLpolyS()

Value

It creates a tcltk menu

Author(s)

Konstantinos Pateras <kostas.pateras@gmail.com>

References

R Development Core Team (2010) tcltk package, CRAN.
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN
See Also

See also demoDist, gamlss.demo.

Examples

demoLpoly5()
Examples
demoPsplines()

---

gamlss.demo The demo for gamlss distributions and smoothing

Description
It starts the gamlss demos. It is an tcltk interface for using the gamlss demos.

Usage
gamlss.demo()

Value
It creates a tcltk menu

Author(s)
Konstantinos Pateras <kostas.pateras@gmail.com>

References
R Development Core Team (2010) tcltk package, CRAN.
Bowman, Bowman, Gibson and Crawford (2008) rpanel, CRAN
Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and
GAMLSS package in R. Accompanying documentation in the current GAMLSS help files, (see
also http://www.gamlss.org/).
org/v23/i07.

See Also
See also demoDist, gamlss.demo,

Examples
gamlss.demo()
There are four functions here to illustrate the fitting of local regressions. i) `locmean`, which uses local means within a symmetric local window, ii) `locpoly`, which uses a local polynomial fit within a symmetric local window. iii) `wlocmean`, which uses a Gaussian kernel and iv) `wlocpoly`, which uses local polynomials weighted by a Gaussian kernel.

### Usage

- `locmean(y, x = seq(1, length(y)), w = rep(1, length(y)), span = 0.5)`
- `locpoly(y, x = seq(1, length(y)), w = rep(1, length(y)), span = 0.5, order = 1)`
- `wlocmean(y, x = seq(1, length(y)), w = rep(1, length(y)), lambda = 0.5)`
- `wlocpoly(y, x = seq(1, length(y)), w = rep(1, length(y)), lambda = 0.5, order = 1)`

### Arguments

- `y` - the response variable
- `x` - the x-variable
- `w` - prior weights
- `span` - the side of the local window compared as a proportion to the total number of observations
- `lambda` - the smoothing parameter for the Gaussian kernel
- `order` - the order of the polynomial

### Details

Those functions can be used for illustration of the basic concepts of smoothing using small data sets. Do not use them with large data because they are computationally inefficient.

### Value

The functions return a `locW` object with values

- `fitted.values` - the fitted values
- `residuals` - the residuals
- `edf` - the effective degrees of freedom
- `rss` - the residual sum of squares
- `lambda` - the smoothing parameter
- `y` - the y variable
- `x` - the x variable
- `w` - the prior weights
Author(s)

Mikis Stasinopoulos, <d.stasinopoulos@londonmet.ac.uk>

References


See Also

loess, ksmooth

Examples

library(MASS)
data(mcycle)
# local means
m0<-locmean(mcycle$accel, mcycle$times, span=.1)
m1<-locmean(mcycle$accel, mcycle$times, span=.2)
m2<-locmean(mcycle$accel, mcycle$times, span=.3)
span <- c("span=0.1", "span=0.2", "span=0.3")
plot(accel~times, data=mcycle, main="local mean")
lines(fitted(m0)~mcycle$times, col=1, lty=1)
lines(fitted(m1)~mcycle$times, col=2, lty=2)
lines(fitted(m2)~mcycle$times, col=3, lty=3)
legend(1.5,50, legend = span, col = 1:3, lty = 1:3, cex = .8, y.intersp = 1)

# kernel estimation
k0<-wlocmean(mcycle$accel, mcycle$times, lambda=1)
k1<-wlocmean(mcycle$accel, mcycle$times, lambda=2)
k2<-wlocmean(mcycle$accel, mcycle$times, lambda=3)
lambda <- c("lambda=1", "lambda=2", "lambda=3")
plot(accel~times, data=mcycle, main="Gaussian kernel fit")
lines(fitted(k0)~mcycle$times, col=1, lty=1)
lines(fitted(k1)~mcycle$times, col=2, lty=2)
lines(fitted(k2)~mcycle$times, col=3, lty=3)
legend(1.5,50, legend = lambda, col = 1:3, lty = 1:3, cex = .8, y.intersp = 1)

# local polynomials
l1<-locpoly(mcycle$accel, mcycle$times, span=.1)
l2<-locpoly(mcycle$accel, mcycle$times, span=.2)
l3<-locpoly(mcycle$accel, mcycle$times, span=.3)
span <- c("span=0.1", "span=0.2", "span=0.3")
plot(accel~times, data=mcycle, main="local linear fit")
lines(fitted(l1)~mcycle$times, col=1, lty=1)
lines(fitted(l2)~mcycle$times, col=2, lty=2)
lines(fitted(l2)~mcycle$times, col=3, lty=3)
legend(1.5,50, legend = span, col = 1:3,
       lty = 1:3, cex = .8, y.intersp = 1)

# weighted local polynomials
lw1<-wLocpoly(mcycle$accel, mcycle$times, lambda=1.5, order=1)
lw2<-wLocpoly(mcycle$accel, mcycle$times, lambda=1.5, order=2)
lw3<-wLocpoly(mcycle$accel, mcycle$times, lambda=1.5, order=3)

span <- c("linear", "quadratic", "cubic")
plot(accel~times, data=mcycle, main="Weighted local linear, quadratic and cubic fits")
lines(fitted(lw1)~mcycle$times, col=1, lty=1)
lines(fitted(lw2)~mcycle$times, col=2, lty=2)
lines(fitted(lw3)~mcycle$times, col=3, lty=3)
legend(1.5,50, legend = span, col = 1:3,
       lty = 1:3, cex = .8, y.intersp = 1)
demo.ST2 (demo.NO), 5
demo.ST3 (demo.NO), 5
demo.ST4 (demo.NO), 5
demo.ST5 (demo.NO), 5
demo.TF (demo.NO), 5
demo.TF2 (demo.NO), 5
demo.WARING (demo.NO), 5
demo.WEI (demo.NO), 5
demo.WEI2 (demo.NO), 5
demo.WEI3 (demo.NO), 5
demo.WLocmean (demo.Locmean), 4
demo.WLocpoly (demo.Locmean), 4
demo.YULE (demo.NO), 5
demo.ZABB (demo.NO), 5
demo.ZABI (demo.NO), 5
demo.ZAGA (demo.NO), 5
demo.ZAIG (demo.NO), 5
demo.ZALG (demo.NO), 5
demo.ZANBI (demo.NO), 5
demo.ZAP (demo.NO), 5
demo.ZIBB (demo.NO), 5
demo.ZIBI (demo.NO), 5
demo.ZINBI (demo.NO), 5
demo.ZIP (demo.NO), 5
demo.ZIP2 (demo.NO), 5
demo.ZIPIG (demo.NO), 5
demoDist, 4, 8, 10, 11
demoLpolyS, 9
demoPsplines, 10
gamlss.demo, 4, 10, 11, 11
ksmooth, 13
Locmean, 12
Locpoly (Locmean), 12
loess, 13
LPOL (demo.LocalRegression), 3
WLocmean (Locmean), 12
WLocpoly (Locmean), 12
WLPOL (demo.LocalRegression), 3