# Package 'DIMORA'

April 16, 2021

**Date** 2021-04-12

**Type** Package

**Title** Diffusion Models R Analysis

**Version** 0.2.0

**Author** Zanghi Federico

**Depends** R (>= 3.1.0), minpack.lm, numDeriv, stats, forecast, reshape2, deSolve

**Maintainer** Zanghi Federico <federico.zanghi.11@gmail.com>

**Description** The implemented methods are: Bass model, Generalized Bass model (with rectangular shock, exponential shock, mixed shock and harmonic shock, 1 to 3 shocks available), Dynamic market potential model, and UCRCD model. The Bass model consists of a simple differential equation that describes the process of how new products get adopted in a population, the Generalized Bass model is a generalization of the Bass model with a function x(t), capturing the changing speed of diffusion. In some real processes the market potential may be not constant over time and a dynamic market potential model is needed. The Guseo-Guidolin model is a specification of this situation. The UCRCD model (Unbalanced Competition and Regime Change Diachronic) is a diffusion model used to capture the dynamics of competition between two products within the same market.

**Encoding** UTF-8

**License** GPL | file LICENSE

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-04-16 13:10:06 UTC

## R topics documented:

## BASS.generalized *Function that estimates a Generalized Bass model*

### Description

This function allows to estimate parameters of a Generalized Bass model, GBM, (Bass et al., 1994) with Exponential, Rectangular, Harmonic or Mixed shock, (Guseo et al. 2007).

### Usage

```
BASS.generalized(sales, shock = c("exp", "rett", "mixed", "harmonic"),
                 nshock, prelimestimates, alpha = 0.05, ous = 100,
                 display = T,max.iter=100)
```

### Arguments

| | |
|---|---|
| `sales` | Instantaneous sales |
| `shock` | Function characterizing the shock. Available options are: Exponential ('exp'), Rectangular ('rett'), Harmonic ('harmonic') and Mixed ('mixed'). |
| `nshock` | Number of shocks desired (from 1 to 3) |
| `prelimestimates` | Vector containing the preliminary estimates of the parameters |
| `alpha` | Desired significance level, the default value is 0.05 |
| `ous` | Numeric value for the out-of-sample forecasts, the default value is 100 |
| `display` | T or F, to display the plot of the model or not, the default value is T |
| `max.iter` | Maximum number of iterations, the default value is 100 |

### Author(s)

Zanghi Federico <`federico.zanghi.11@gmail.com`>

### See Also

BASS.standard

BASS.standard.generator

BASS.plot

make.instantaneous

GG.model

UCRCD

predict_SARMAX.refinement

## BASS.plot *Function that draws the plot of Bass model*

### Description

Function that draws the plot of a standard Bass model.

### Usage

```
BASS.plot(data)
```

### Arguments

| | |
|---|---|
| data | Result of BASS.standard.generator |

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

### See Also

BASS.standard

BASS.standard.generator

make.instantaneous

BASS.generalized

GG.model

UCRCD

### Examples

```
#Example 1

data = BASS.standard.generator(10,0.1,0.1,8,3)
BASS.plot(data)
```

## BASS.standard — *Function that estimates a standard Bass model*

### Description

This function allows to estimate the parameters of a standard Bass model, BM, (Bass, 1969) by using two different methods: nonlinear least squares, NLS, or OPTIM, which minimizes the residual sum of squares.

### Usage

```
BASS.standard(sales, method = "nls", prelimestimates = c(sum(sales) +
            100, 0.01, 0.1), ous= 100, alpha = 0.05, display = T, max.iter = 100)
```

### Arguments

`sales` Instantaneous sales

`method` Method for parameter estimation, 'nls' or 'optim'

`prelimestimates` A vector containing the starting values for parameter estimation. Default values are m = sum(sales)+100, p = 0.01, q = 0.1

`alpha` Significance level for confidence intervals

`display` T or F, to display the plot of the model or not, the default value is T

`ous` Numeric value for the out-of-sample forecasts, the default value is 100

`max.iter` Maximum number of iterations, the default value is 100. This option works only with method="nls"

### Author(s)

Zanghi Federico <`federico.zanghi.11@gmail.com`>

### See Also

`BASS.standard.generator`

`BASS.plot`

`make.instantaneous`

`BASS.generalized`

`GG.model`

`UCRCD`

### Examples

```
#Example 1
Sales <- c(840,1470,2110,4000,7590,10950,10530,9470,7790,5890)
BASS.standard(sales = Sales,method = 'nls')
```

## BASS.standard.generator *Function that returns the results of standard Bass model*

### Description

Function returning the results of a standard Bass model, BM, (Bass, 1969).

### Usage

```
BASS.standard.generator(m,p,q,tstart=1,n=50)
```

### Arguments

| | |
|---|---|
| m | Market potential |
| p | Coefficient of innovation |
| q | Coefficient of imitation |
| tstart | Inital time |
| n | Number of observations |

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

### See Also

BASS.standard

BASS.plot

make.instantaneous

BASS.generalized

GG.model

UCRCD

### Examples

```
#Example 1

BASS.standard.generator(10,0.1,0.1,8,3)
```

## `classic.plot.sarima` *Plot for* `SARMAX.refinement` *function.*

### Description

Function to obtain the same graphs that are obtained with "display = T" in the SARMAX.refinement function.

### Usage

```
classic.plot.sarima(model,typeog=0,...)
```

### Arguments

| | |
|---|---|
| `model` | An object of class `Dimora`, i.e. `BASS.generalized`, `BASS.standard` or `GG.model`. |
| `typeog` | Three plots available, residuals versus time (`typeog = 1`), residual ACF (`typeog = 2`), fitted values on real data (`typeog = 3`), mode `typeog = 0` shows all plots. The default value is `typeog = 0`. |
| `...` | Other graphic parameters. |

### Author(s)

Zanghi Federico <`federico.zanghi.11@gmail.com`>

## `DIMORA` *DIMORA Package*

### Description

In the following package some innovation diffusion models have been implemented to analyze nonlinear growth processes, for both univariate and bivariate time series data.

The implemented models are: standard Bass model, Generalized Bass model (with rectangular shock, exponential shock, mixed shock and harmonic shock, 1 to 3 shocks available), Dynamic market potential model, and UCRCD model.

The Bass model, BM, (Bass, 1969), consists of a simple differential equation that describes the process of how new products get adopted in a population, while the Generalized Bass model, GBM, (Bass et al., 1994) is a generalization of the Bass model with a function x(t), capturing the changing speed of diffusion. In some real processes the market potential may be not constant over time and a dynamic market potential model is needed. The Guseo-Guidolin model, GGM, (Guseo and Guidolin, 2009) is a specification of this situation.

The UCRCD model, (Guseo and Mortarino, 2014) is a diffusion model useful to capture the dynamics of competition between two products within the same market. See the full documentation for more details.

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

Andrea Savio <svandr97@gmail.com>

### See Also

BASS.standard

BASS.standard.generator

BASS.plot

BASS.generalized

GG.model

UCRCD

make.instantaneous

SARMAX.refinement

classic.plot.sarima

summary.Dimora

predict.Dimora

plot.Dimora

### Examples

```
library(DIMORA)
```

## GG.model *Function that estimates a dynamic market potential model*

### Description

This function allows to estimate the parameters of a model with a time-dependent market potential, m(t). The market potential may be defined according to the form proposed in Guseo and Guidolin (2009), generating the GG model, GGM. Other forms for m(t) may be defined according to the following: m(t) must depend on t and be a cdf.

To use the function, two options are available:

- Use the classic GGM, by setting function with sales, preliminary estimates and alpha

- Use the model with another m(t) function, by setting function with sales, preliminary estimates, mt (as a function object) and alpha. Note that the m(t) function must be a cdf, (must have codomain in (0,1)).

Note: Default preliminary estimates are based on standard Bass model parameters. (This starting point does not always guarantee convergence of the algorithm, and preliminary estimates may be manually selected).

## Usage

```
GG.model(sales, prelimestimates = NULL, mt = "base", alpha = 0.05, ous=100,
display=T,max.iter=100,...)
```

## Arguments

`sales` Instantaneous sales

`prelimestimates` Vector containing the preliminary estimates of the parameters, default values are based on parameter estimates of a standard Bass model

`mt` Function type object, representing the variable market potential, the default for m(t) is that of GGM

`alpha` Desired significance level, the default value is 0.05

`ous` Numeric value for the out-of-sample forecasts, the default value is 100

`display` T or F, to display the plot of the model or not, the default value is T

`max.iter` Maximum number of iterations, the default value is 100

`...` Other graphic parameters

## Author(s)

Zanghi Federico <`federico.zanghi.11@gmail.com`>

## See Also

`BASS.standard`

`BASS.standard.generator`

`BASS.plot`

`make.instantaneous`

`BASS.generalized`

`UCRCD`

## Examples

```
# Example 1


# dati <- c(169,397,1496,2131,2678,3431,3852,4725,5081,4592,
# 6272,6572,6479,7092,6669,7498,7380,5993,5882,9523,9885,9437
# ,10023,10103,9534,11228,10779,10687,11732,11460,12142,11465,
# 11854,11177,11112,11324,12790,12229,12116,11280,14460,13090,
# 12383,13076,13518,13781,13455,13758,14747,12405,8145,11245,
# 12211,14557,13943,14838,14275,14911,14003,14111,14241,13242,
# 15477,15219,14691,14541,12465,15909,16118,10568,11235,17345,
# 15694,15746,17129,16127,15691,16689,16552,16326,16485,15615,
# 17040,16119,13731,16102,14692,14162,17013,17058,15782,14762,
# 16813,16152,15954,16129,16356,16752)
# sp = c(1.69062e+06,2.60513e-03,3.20522e-02,1.00000e-03,1.00000e-01)
# sp1 = c(1.69062e+06,2.60513e-03,3.20522e-02)

# GG.model(sales = dati, prelimestimates = sp1 , function(x) pchisq(x,10),col=2)
# GG.model(sales = dati, mt = function(x) pchisq(x,10),col=2)
# GG.model(sales = dati, prelimestimates = sp,col=2)
# GG.model(sales = dati, col=2)
```

## make.instantaneous *Function that converts the data*

### Description

Function that turns cumulative values into instantaneous values. This function may be useful in some cases, since all procedures in this package are built using instantaneous data.

### Usage

```
make.instantaneous(cumulate.data)
```

### Arguments

| | |
|---|---|
| cumulate.data | Cumulative data |

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

### See Also

BASS.standard

BASS.standard.generator

BASS.plot

BASS.generalized

GG.model

UCRCD

### Examples

```
#Example 1
cumulate.data = c(1,2,3,6,12)
data.inst = make.instantaneous(cumulate.data)
```

## plot.Dimora *plot method for Dimora*

### Description

Three plots (to be selected with `typeog`) are currently available: a plot of fitted values over observed data (typeog=1), a plot of residuals against fitted values (typeog=2), a plot of residuals against time (typeog=3).

### Usage

```
## S3 method for class 'Dimora'
plot(x,..., product=1, mode=c(”i”,”c”),typeog=1,oos=c(”y”,”n”),
                    xlim=NULL,ltycurve=1,lwdcurve=1.5,typecurve=”l”,colcurve=2)
```

### Arguments

| | |
|---|---|
| `x` | An object of class `Dimora`, i.e. `BASS.generalized`, `BASS.standard`, `GG.model`. |
| `product` | 1 or 2, which indicates whether there are one or two products |
| `mode` | Mode "i" for instantaneous values and "c" for cumulative values |
| `typeog` | Fitted values on observed data `typeog=1`, residuals against fitted values `typeog=2` and residuals against time `typeog=3`, and all the plots with `typeog=0` |
| `oos` | "y" or "n" if the out-of-sample plot is desired |
| `xlim` | The limit for the graph, default values is null, if the out-of-sample plot is desired the portion of out-of-sample values may be defined |
| `ltycurve` | "lty" parameter for the fitted values plot (`typeog=1`). |
| `lwdcurve` | "lwd" parameter for the fitted values plot (`typeog=1`). |
| `typecurve` | "type" parameter for the fitted values plot (`typeog=1`). |
| `colcurve` | "col" parameter for the fitted values plot (`typeog=1`). |
| `...` | Other graphic parameters. |

### Author(s)

Zanghi Federico <`federico.zanghi.11@gmail.com`>

## predict.Dimora *predict method for Dimora objects*

### Description

Prediction of test data using any model of the package.

### Usage

```
## S3 method for class 'Dimora'
predict(object,..., newx)
```

### Arguments

| | |
|---|---|
| object | An object of class Dimora, i.e. BASS.generalized, BASS.standard, GG.model. |
| newx | A number or a vector of numeric values, representing time t. |
| ... | Additional arguments affecting the predictions produced. |

### Value

The method returns a vector of predicted values.

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

## predict_SARMAX.refinement *Performs prediction for* SARMAX.refinement *function.*

### Description

predict_SARMAX.refinement is a function for predictions from a SARMAX.refinement function.

### Usage

```
predict_SARMAX.refinement(fit,newx)
```

### Arguments

| | |
|---|---|
| fit | A model object from SARMAX.refinement for which prediction is desired. |
| newx | Vector of new values at which predictions are performed. |

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

## SARMAX.refinement — *SARMAX refinement of a* Dimora *model.*

### Description

This function allows to refine results obtained with a model of the Dimora class through a SARIMAX model, (Guseo et. al., 2007). The function predict_SARMAX.refinement is aviable for this procedure.

### Usage

```
SARMAX.refinement(fit,arima_order,seasonal_order,period=1,display=F)
```

### Arguments

fit — Object of classDimora, i.e. BASS.generalized, codeBASS.standard or codeGG.model.

arima_order — Non-seasonal part of the SARIMA model

seasonal_order — Seasonal part of the SARIMA model

period — Period of the seasonal part of the ARIMA model. Default value is 1 (daily data).

display — Logical "T" and "F", "T" for some graphs to be shown

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

### See Also

BASS.standard

BASS.standard.generator

BASS.plot

make.instantaneous

GG.model

predict_SARMAX.refinement

UCRCD

## summary.Dimora *summary method for Dimora objects*

### Description

Summary method for the Dimora objects.

### Usage

```
## S3 method for class 'Dimora'
summary(object,...)
```

### Arguments

object An object of class Dimora i.e. BASS.generalized, BASS.standard, GG.model and UCRCD.

... Additional arguments related to the summary produced.

### Value

summary is a generic function used to produce summary of results for the models fitted in Dimora. The function invokes particular methods which depend on the class of the first argument, in this case the class is Dimora.

### Author(s)

Zanghi Federico <federico.zanghi.11@gmail.com>

## UCRCD *Function that estimates a UCRCD model*

### Description

This function allows to estimate diachronic competition between two products, according to the Unbalanced Competition Regime Change Diachronic model, UCRCD, (Guseo and Mortarino, 2014).

### Usage

```
UCRCD(sales1, sales2, c2, display=T, alpha=0.05,
      delta=0.01, gamma=0.01, par="double",
      m1 =BASS.standard(sales1,display = F)$Estimate[1,1],
      m2 =BASS.standard(sales2,display = F)$Estimate[1,1],
      p1c=BASS.standard(sales1,display = F)$Estimate[2,1],
      q1c=BASS.standard(sales1,display = F)$Estimate[3,1],
      p2c=BASS.standard(sales2,display = F)$Estimate[2,1],
      q2c=BASS.standard(sales2,display = F)$Estimate[3,1])
```

### Arguments

`sales1` Instantaneous sales of the first product

`sales2` Instantaneous sales of the second product

`c2` Time at which the second product enters the market

`display` T or F, to display the plot of the model or not, the default value is T

`alpha` Desired significance level, the default value is 0.05

`delta` Preliminary estimates for parameter delta

`gamma` Preliminary estimates for parameter gamma

`par` Default value is par = ”double”, where delta is different from gamma. If par = "unique", delta is equal to gamma

`m1` Preliminary estimates of the first product’s market potential under competition

`m2` Preliminary estimates of the second product’s market potential

`p1c` Preliminary estimates of the first product’s innovation coefficient under competition

`p2c` Preliminary estimates of the second product’s innovation coefficient under competition

`q1c` Preliminary estimates of the first product’s imitation coefficient under competition

`q2c` Preliminary estimates of the second product’s imitation coefficient under competition

### Author(s)

Savio Andrea <svandr97@gmail.com>

### See Also

`BASS.standard`

`BASS.standard.generator`

`BASS.generalized`

`BASS.plot`

`make.instantaneous`

`GG.model`

# Index