

# Package: diffusion (via r-universe)

October 24, 2024

**Type** Package

**Title** Forecast the Diffusion of New Products

**Version** 0.4.0

**URL** <https://github.com/mamut86/diffusion>

**BugReports** <https://github.com/mamut86/diffusion/issues>

**Description** Various diffusion models to forecast new product growth.  
Currently the package contains Bass, Gompertz, Gamma/Shifted  
Gompertz and Weibull curves. See Meade and Islam (2006)  
<[doi:10.1016/j.ijforecast.2006.01.005](https://doi.org/10.1016/j.ijforecast.2006.01.005)>.

**License** LGPL-2.1

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**LazyData** true

**Imports** nloptr, systemfit, optimx

**RoxygenNote** 7.3.1

**Repository** <https://mamut86.r-universe.dev>

**RemoteUrl** <https://github.com/mamut86/diffusion>

**RemoteRef** HEAD

**RemoteSha** 13cce4653f8be7e409683fc2faaf9b371b83d515

## Contents

difcurve . . . . .	2
diffusion . . . . .	3
is.diffusion . . . . .	6
plot.diffusion . . . . .	7
predict.diffusion . . . . .	8
print.diffusion . . . . .	9
tsAc . . . . .	10
tsCarstock . . . . .	10

tsChicken . . . . .	11
tsCovid . . . . .	11
tsIbm . . . . .	12
tsMetal . . . . .	12
tsSafari . . . . .	13
tsWindows . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

difcurve	<i>Calculates the values for various diffusion curves, given some parameters.</i>
----------	---

---

### Description

This function calculates the values of diffusion curves that can be of "bass", "gompertz", "gsgompertz" or "weibull" type, given some parameters.

### Usage

```
difcurve(
  n,
  w = c(0.01, 0.1, 10),
  type = c("bass", "gompertz", "gsgompertz", "weibull"),
  curve = NULL
)
```

### Arguments

n	number of periods to calculate values for.
w	vector of curve parameters (see note). If argument curve is used, this is ignored.
type	diffusion curve to use. This can be "bass", "gompertz" and "gsgompertz". If argument curve is used, this is ignored.
curve	if provided w and type are taken from an object of class diffusion, the output of <a href="#">diffusion</a> .

### Value

Returns a matrix of values with each row being a period.

### Note

w needs to be provided for the Bass curve in the order of ("m", "p", "q"), where "p" is the coefficient of innovation, "q" is the coefficient of imitation and "m" is the market size coefficient.

For the Gompertz curve, vector w needs to be in the form of ("m", "a", "b"). Where "a" is the x-axis displacement coefficient, "b" determines the growth rate and "m" sets, similarly to Bass model, the market potential (saturation point).

For the Shifted-Gompertz curve, vector  $w$  needs to be in the form of ("m", "a", "b", "c"). Where "a" is the x-axis displacement coefficient, "b" determines the growth rate, "c" is the shifting parameter and "m" sets, similarly to Bass model, the market potential (saturation point).

For the Weibull curve, vector  $w$  needs to be in the form of ("m", "a", "b"). Where "a" is the scale parameter, "b" determines the shape. Together, "a" and "b" determine the stepness of the curve. The "m" parameter sets the market potential (saturation point).

### Author(s)

Oliver Schaer, <info@oliverschaer.ch>,  
Nikolaos Kourentzes, <nikolaos@kourentzes.com>

### See Also

[diffusion](#) for fitting a diffusion curve.

### Examples

```
difcurve(w=c(0.01,0.1,10),20)
```

---

diffusion

*Fit various diffusion curves.*

---

### Description

This function fits diffusion curves that can be of "bass", "gompertz", "gsgompertz" (Gamma/Shifted Gompertz curve) or "Weibull" type.

### Usage

```
diffusion(
  y,
  w = NULL,
  cleanlead = c(TRUE, FALSE),
  loss = 2,
  cumulative = c(TRUE, FALSE),
  verbose = c(FALSE, TRUE),
  type = c("bass", "gompertz", "gsgompertz", "weibull"),
  method = c("L-BFGS-B", "Nelder-Mead", "BFGS", "hjk", "Rcgmin", "bobyqa"),
  maxiter = 500,
  opttol = 1e-06,
  multisol = c(FALSE, TRUE),
  initpar = c("linearize", "preset"),
  mscal = c(TRUE, FALSE),
  ...
)
```

**Arguments**

y	vector with adoption per period
w	vector of curve parameters (see note). Parameters set to NA will be optimized. If w = NULL (default) all parameters are optimized.
cleanlead	removes leading zeros for fitting purposes (default == TRUE)
loss	the l-norm (1 is absolute errors, 2 is squared errors).
cumulative	If TRUE optimisation is done on cumulative adoption.
verbose	if TRUE console output is provided during estimation (default == FALSE)
type	diffusion curve to use. This can be "bass", "gomperz" and "gsgomperz"
method	optimization method to use. These can be "Nelder-Meade", "L-BFGS-B", "BFGS", "hjk", "Rcgmin", "bobyqa". Typically, good performance is achieved with "Nelder-Meade" and "L-BFGS-B". "hjk" and "Rcgmin" might be an alternative for complex shapes but have substantially higher computational costs. For further details on optimization algorithms we refer to the optimx package documentation
maxiter	number of iterations the optimiser takes (default == 5000)
opttol	Tolerance for convergence (default == 1.e-06)
multisol	when "TRUE" multiple optimisation solutions from different initialisations of the market parameter are used (default == "FALSE")
initpar	vector of initialisation parameters. If set to preset a predefined set of internal initialisation parameters is used while "linearize" uses linearized initialisation methods (default == "linearize").
mscal	scales market potential at initialisation with the maximum of the observed market potential for better optimization results (default == TRUE)
...	accepts pvalreps, bootstrap repetitions to estimate (marginal) p-values; eliminate, if TRUE eliminates insignificant parameters from the estimation (forces pvalreps = 1000 if left to 0) sig, significance level used to eliminate parameters.

**Value**

Returns an object of class diffusion, which contains:

- type diffusion curve type used
- call calls function fitted
- w named vector of fitted parameters
- y actuals
- fit fitted values of model
- frc forecasts for future periods. This is NULL until `predict.diffusion` is called.
- mse insample Mean Squared Error
- prew the w of the previous generation
- pval p-values for w
- init the initial values that have been used for the optimizer

**Bass curve**

The optimization of the Bass curve is initialized by the linear approximation suggested in Bass (1969).

**Gompertz curve**

The initialization of the Gompertz curve uses the approach suggested by Jukic et al. (2004), but is adapted to allow for the non-exponential version of the Gompertz curve. This makes the market potential parameter equivalent to the Bass curves and the market potential from Bass curve is used for initialization.

**Gamma/Shifted Gompertz**

The curve is initialized by assuming the shift operator to be 1 and becomes equivalent to the Bass curve, as shown in Bemmaor (1994). A Bass curve is therefore used as an estimator for the remaining initial parameters.

**Weibull**

The initialization is obtained through by a linear approximation median-ranked OLS described in Sharif and Islam 1980.

**Note**

vector  $w$  needs to be provided for the Bass curve in the order of "m", "p", "q", where "p" is the coefficient of innovation, "q" is the coefficient of imitation and "m" is the market size coefficient.

For the Gompertz curve, vector  $w$  needs to be in the form of ("m", "a", "b"). Where "a" is the x-axis displacement coefficient, "b" determines the growth rate and "m" sets, similarly to the Bass curve, the market potential (saturation point).

For the Shifted-Gompertz curve, vector  $w$  needs to be in the form of ("m", "a", "b", "c"). Where "a" is the x-axis displacement coefficient, "b" determines the growth rate, "c" is the shifting parameter and "m" sets, similarly to the Bass curve, the market potential (saturation point).

For the Weibull curve, vector  $w$  needs to be in the form of ("m", "a", "b"). Where "a" is the scale parameter, "b" determines the shape. Together, "a" and "b" determine the steepness of the curve. The "m" parameter sets the market potential (saturation point).

Parameters are estimated by minimising the Mean Squared Error with a subplex algorithm from the optimx package. Optionally p-values of the coefficients can be determined via bootstrapping. Furthermore, the bootstrapping allows removing insignificant parameters from the optimization process.

**Author(s)**

Oliver Schaer, <info@oliverschaer.ch>,

Nikolaos Kourentzes, <nikolaos@kourentzes.com>

## References

- For an introduction to diffusion curves see Ord K., Fildes R., Kourentzes N. (2017) **Principles of Business Forecasting 2e**. *Wessex Press Publishing Co.*, Chapter 12.
- Bass, F.M., 1969. A new product growth for model consumer durables. *Management Science* 15(5), 215-227.
- Bemmaor, A. 1994. Modeling the Diffusion of New Durable Goods: Word-of-Mouth Effect versus Consumer Heterogeneity. In G. Laurent, G.L. Lilien and B. Pras (Eds.). *Research Traditions in Marketing*. Boston: Kluwer, pp. 201-223.
- Jukic, D., Kralik, G. and Scitovski, R., 2004. Least-squares fitting Gompertz curve. *Journal of Computational and Applied Mathematics*, 169, 359-375.
- Sharif, N.M. and Islam, M.N. 1980. The Weibull Distribution as a General Model for Forecasting Technological Change. *Technological Forecasting and Social Change*, 18, 247-256.

## See Also

[predict.diffusion](#), [plot.diffusion](#) and [print.diffusion](#).

[seqdiffusion](#) for sequential diffusion model fitting across product generations.

## Examples

```
fitbass <- diffusion(diff(tsChicken[, 2]), type = "bass")
fitgomp <- diffusion(diff(tsChicken[, 2]), type = "gompertz")
fitgsg <- diffusion(diff(tsChicken[, 2]), type = "gsgompertz")
fitgwb <- diffusion(diff(tsChicken[, 2]), type = "weibull")

# Produce some plots
plot(fitbass)
plot(fitgomp)
plot(fitgsg)
plot(fitgwb)
```

---

is.diffusion

*Diffusion class checkers*

---

## Description

Functions to check if an object is of the specified class

## Usage

```
is.diffusion(x)
```

```
is.bass(x)
```

**Arguments**

x                    The object to check.

**Details**

The list of functions includes:

- `is.diffusion()` tests if the object was produced by a `diffusion()` function.
- `is.bass()` checks if the forecast was produced by the `bass()` function.

**Value**

TRUE if this is the specified class and FALSE otherwise.

**Author(s)**

Ivan Svetunkov, <ivan@svetunkov.ru>,  
Oliver Schaer, <info@oliverschaer.ch>

---

plot.diffusion                    *Plot a fitted diffusion curve.*

---

**Description**

Produces a plot of a fitted diffusion curve.

**Usage**

```
## S3 method for class 'diffusion'  
plot(x, cumulative = c(FALSE, TRUE), ...)
```

**Arguments**

x                    diffusion object, produced using [diffusion](#).  
cumulative        If TRUE plot cumulative adoption.  
...                Unused argument.

**Value**

None. Function produces a plot.

**Author(s)**

Oliver Schaer, <info@oliverschaer.ch>,  
Nikolaos Kourentzes, <nikolaos@kourentzes.com>

**See Also**

[diffusion](#).

**Examples**

```
fit <- diffusion(tsChicken[, 2])
plot(fit)
```

---

predict.diffusion	<i>Predict future periods of a fitted diffusion curve.</i>
-------------------	--

---

**Description**

Calculates the values for h future periods of a fitted diffusion curve.

**Usage**

```
## S3 method for class 'diffusion'
predict(object, h = 10, ...)
```

**Arguments**

object	diffusion object, produced using <a href="#">diffusion</a> .
h	Forecast horizon.
...	Unused argument.

**Value**

Returns an object of class diffusion, which contains:

- type diffusion curve type used
- call calls function fitted
- w named vector of fitted parameters
- y actuals
- fit fitted values of model
- frc forecasts for future periods.
- mse insample Mean Squared Error
- prew the w of the previous generation
- pval p-values for w

**Note**

This function populates the matrix frc of the diffusion object used as input.



**Author(s)**

Oliver Schaer, <info@oliverschaer.ch>,  
Nikoloas Kourentzes, <nikoloas@kourentzes.com>

**See Also**

[diffusion.](#)

**Examples**

```
fit <- diffusion(tsChicken[, 2])  
fit <- predict(fit, 20)  
plot(fit)
```

---

<code>print.diffusion</code>	<i>Print a fitted diffusion curve.</i>
------------------------------	--

---

**Description**

Outputs the result of a fitted diffusion curve.

**Usage**

```
## S3 method for class 'diffusion'  
print(x, ...)
```

**Arguments**

<code>x</code>	diffusion object, produced using <a href="#">diffusion</a> .
<code>...</code>	Unused argument.

**Value**

None. Console output only.

**Author(s)**

Oliver Schaer, <info@oliverschaer.ch>,  
Nikolaos Kourentzes, <nikolaos@kourentzes.com>

**See Also**

[diffusion.](#)

**Examples**

```
fit <- diffusion(tsChicken[, 2])
print(fit)
```

---

tsAc *Time series: Assassins Creeds*

---

**Description**

A dataset containing the weekly sales of Assassins Creeds game.

**Format**

A matrix with 380 observations and 8 variables

**ac1** Assassins Creed 1

**ac2** Assassins Creed 2

**ac3** Assassins Creed 3

**ac4** Assassins Creed 4

**ac5** Assassins Creed 5

**ac6** Assassins Creed 6

**ac7** Assassins Creed 7

**ac8** Assassins Creed 8

**References**

VGChartz

---

tsCarstock *Time series: Stock of cars*

---

**Description**

A dataset containing the yearly stock of cars in the Netherlands (1965-1989).

**Format**

A data frame with 25 observations and 3 variables

**year** Year

**raw** Raw stock numbers

**smoothed** Smoothed stock numbers as described by Franses (1994)

**References**

Franses, P.H. 1994. Fitting a Gompertz curve. *Journal of Operational Research Society*, 45, 109-113.

---

tsChicken                      *Time series: Chicken weight*

---

**Description**

A dataset containing the average weekly female chicken weight.

**Format**

A data frame with 13 observations and 2 variables

**time** Weeks since birth

**weight** Weight of the female chicken in Kg

**References**

Jukic, D., Kralik, G. and Scitovski, R. 2004. Least-square fitting Gompertz curve. *Journal of Computational and Applied Mathematics*, 169, 359-375.

---

tsCovid                              *Time series: COVID-19 confirmed cases US*

---

**Description**

A dataset containing the number of confirmed COVID-19 cases in the US.

**Format**

A ts object with 107 days of observations

**tsCovid** Daily confirmed COVID-19 cases

**Source**

<https://github.com/CSSEGISandData/COVID-19>

**References**

COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University

---

tsIbm

*Time series: Sales of IBM Computers*

---

**Description**

A dataset containing the first four generations of yearly IBM general-purpose computers installations in the USA.

**Format**

A data frame with 24 observations and 4 variables

**SIU1** 1st generation

**SIU2** 2nd generation (starts 6 years after first generation)

**SIU3** 3rd generation (starts 11 years after first generation)

**SIU4** 4th generation (starts 16 years after first generation)

**Source**

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8bbf197bc39a27ccf44cf5ed22b5db3da0c>

**References**

Bass, P.I. and Bass, F.M. 2004. IT Waves: Two Completed Generational Diffusion Models. Working Paper Basseconomics, 1-33.

---

tsMetal

*Time series: U.S. Merchant Marine conversion to metal*

---

**Description**

A dataset with conversion of U.S. Merchant Marine from wood to metal.

**Format**

A data frame with 17 observations and 2 variables

**year** Year

**substitution** Conversion to metal

**References**

Martino, J.P. 1993. Technological Forecasting for Decision Making. 3rd edition. New York: McGraw-Hill.

---

tsSafari

*Time series: Safari Browser market share*

---

### Description

A dataset containing the monthly market share of Safari browser generations from Safari 4.0 to Safari 10.

### Format

A data frame with 98 observations and 13 variables

**Date** Log file date

**Safari10.0** Usage of Windows 10

**Safari9.1** Market share of Safari browser v 10.0

**Safari9.0** Market share of Safari browser v 9.1

**Safari8.0** Market share of Safari browser v 9.0

**Safari7.1** Market share of Safari browser v 8.0

**Safari7.0** Market share of Safari browser v 7.1

**Safari6.1** Market share of Safari browser v 6.1

**Safari6.0** Market share of Safari browser v 6.0

**Safari5.1** Market share of Safari browser v 5.1

**Safari5.0** Market share of Safari browser v 5.0

**Safari4.1** Market share of Safari browser v 4.1

**Safari4.0** Market share of Safari browser v 4.0

### Source

<https://gs.statcounter.com/browser-version-market-share>

---

tsWindows

*Time series: Windows OS Platform Statistics*

---

### Description

A dataset containing the 3WSchools monthly log files of Windows operating system usage from March 2003 until February 2017.

**Format**

A data frame with 168 observations and 9 variables

**Date** Log file date

**Win10** Usage of Windows 10

**Win8** Usage of Windows 8

**Win7** Usage of Windows 7

**Vista** Usage of Windows Vista

**WinXP** Usage of Windows XP

**Win2000** Usage of Windows 2000

**Win98** Usage of Windows 98

**Win95** Usage of Windows 95

**Note**

From March 2003 until January 2008 log file is only available bi-monthly. To retain monthly consistency, values have been linearly interpolated

**Source**

[https://www.w3schools.com/browsers/browsers\\_os.asp](https://www.w3schools.com/browsers/browsers_os.asp)

# Index

## \* **ts**

is.diffusion, 6

## \* **univar**

is.diffusion, 6

difcurve, 2

diffusion, 2, 3, 3, 7–9

is.bass (is.diffusion), 6

is.diffusion, 6

plot.diffusion, 6, 7

predict.diffusion, 4, 6, 8

print.diffusion, 6, 9

seqdiffusion, 6

tsAc, 10

tsCarstock, 10

tsChicken, 11

tsCovid, 11

tsIbm, 12

tsMetal, 12

tsSafari, 13

tsWindows, 13