# Package 'dynamicalSystemsNotes'

June 29, 2023

**Title** Dynamical Systems Notes

**Version** 0.9.1

**Description** HDFS 538 Dynamical Systems Methods and Applications personal study notes.

**URL** https://github.com/ijapesigan/dynamicalSystemsNotes,
https://ijapesigan.github.io/dynamicalSystemsNotes/

**BugReports** https://github.com/ijapesigan/dynamicalSystemsNotes/issues

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**VignetteBuilder** knitr

**Depends** R (>= 3.0.0)

**Suggests** knitr, rmarkdown, testthat

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Ivan Jacob Agaloos Pesigan [aut, cre, cph]
(<https://orcid.org/0000-0003-4818-8420>)

**Maintainer** Ivan Jacob Agaloos Pesigan <ijapesigan@gmail.com>

## R topics documented:

## Cobweb *Cobweb Plot*

### Description

Cobweb Plot

### Usage

```
Cobweb(y0, func, tol = sqrt(.Machine$double.eps), max_iter = 1000L)
```

### Arguments

| | |
|---|---|
| y0 | Numeric. Initial condition. |
| func | Function. The input is $y_{i(t-1)}$ and the output is $y_{it}$. |
| tol | Small numeric value. Convergence criteria. |
| max_iter | Large positive integer. Maximum number of iterations. |

### Author(s)

Ivan Jacob Agaloos Pesigan

### Examples

```
# linear
func <- LinearConstructor(alpha = 8.0, beta = 0.8)
Cobweb(y0 = 0.01, func = func)

# logistic
func <- LogisticConstructor(r = 1.5, K = 10)
Cobweb(y0 = 0.01, func = func)
```

## FixedPoint *Fixed Point*

### Description

Fixed Point

### Usage

```
FixedPoint(y0, func, tol = sqrt(.Machine$double.eps), max_iter = 1000L)
```

### Arguments

| | |
|---|---|
| y0 | Numeric. Initial condition. |
| func | Function. The input is $y_{i(t-1)}$ and the output is $y_{it}$. |
| tol | Small numeric value. Convergence criteria. |
| max_iter | Large positive integer. Maximum number of iterations. |

### Author(s)

Ivan Jacob Agaloos Pesigan

### Examples

```
# linear
func <- LinearConstructor(alpha = 8.0, beta = 0.8)
FixedPoint(y0 = 0.01, func = func)

# logistic
func <- LogisticConstructor(r = 1.5, K = 10)
FixedPoint(y0 = 0.01, func = func)
```

---

## LinearConstructor *Simple Linear System Function Constructor*

---

### Description

Simple Linear System Function Constructor

### Usage

```
LinearConstructor(alpha, beta, sigmasq = NULL)
```

### Arguments

| | |
|---|---|
| alpha | Numeric. Intercept $\alpha$. |
| beta | Numeric. Slope $\beta$. |
| sigmasq | Positive numeric value. Variance of the random error term with expected value of zero. If sigmasq = NULL, the system is deterministic. |

### Details

The simple linear system is given by

$$y_{it} = \alpha + \beta y_{i(t-1)}$$

where $\alpha$ is the intercept and $\beta$ is the slope.

### Author(s)

Ivan Jacob Agaloos Pesigan

### Examples

```
func <- LinearConstructor(alpha = 8.0, beta = 0.8)
func(0.1)
```

---

## LogisticConstructor *Simple Logistic System Function Constructor*

---

### Description

Simple Logistic System Function Constructor

### Usage

```
LogisticConstructor(r, K, sigmasq = NULL)
```

### Arguments

| | |
|---|---|
| r | Numeric. Growth rate. |
| K | Numeric Carrying capacity. |
| sigmasq | Positive numeric value. Variance of the random error term with expected value of zero. If sigmasq = NULL, the system is deterministic. |

### Details

The logistic system is given by

$$y_{it} = r y_{i(t-1)} \left(1 - \frac{y_{i(t-1)}}{K}\right)$$

where $r$ is the growth rate and $K$ is the carrying capacity.

### Author(s)

Ivan Jacob Agaloos Pesigan

### Examples

```
func <- LogisticConstructor(r = 1.5, K = 10)
func(0.1)
```

`UnivSeries` *Univariate Time Series*

## Description

Univariate Time Series

## Usage

```
UnivSeries(y0, func, nt)
```

## Arguments

| | |
|---|---|
| `y0` | Numeric. Initial condition. |
| `func` | Function. The input is $y_{i(t-1)}$ and the output is $y_{it}$. |
| `nt` | Positive integer. Maximum discrete time points. |

## Details

The univariate time series is generated using the following equation

$$y_{it} = f\left(y_{i(t-1)}\right).$$

## Author(s)

Ivan Jacob Agaloos Pesigan

## Examples

```
# linear
func <- LinearConstructor(alpha = 8.0, beta = 0.8)
y <- UnivSeries(y0 = 0.01, func = func, nt = 100)
plot(y)

# logistic
func <- LogisticConstructor(r = 1.5, K = 10)
y <- UnivSeries(y0 = 0.01, func = func, nt = 100)
plot(y)
```

# Index