

Package ‘rbc’

July 23, 2025

Title Regression by Composition

Version 0.1.0

Description Flexible statistical modelling using a modular framework for regression, in which groups of transformations are composed together and act on probability distributions.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports Formula, R6

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Daniel Farewell [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-8871-1653>>),
Simon Schoenbuchner [ctb] (ORCID:
<<https://orcid.org/0000-0003-4892-2170>>)

Maintainer Daniel Farewell <farewelld@cardiff.ac.uk>

Repository CRAN

Date/Publication 2024-11-06 21:00:01 UTC

Contents

AIC.RegressionByComposition	2
append_flow	3
Bernoulli	3
coef.RegressionByComposition	4
fitted.RegressionByComposition	4
logLik.RegressionByComposition	5
LogNormal	5

Moebius	6
Normal	6
Power	7
predict.RegressionByComposition	7
rbc	8
residuals.RegressionByComposition	9
Scale	9
ScaleOdds	10
ScaleRisk0	10
ScaleRisk1	10
starr	11
summary.RegressionByComposition	11
Translate	12
TranslateRisk1	12
vcov.RegressionByComposition	12
Index	13

AIC.RegressionByComposition
Compute Akaike Information Criterion from a regression by composition

Description

Compute Akaike Information Criterion from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'  
AIC(object, ..., k = 2)
```

Arguments

- object a RegressionByComposition object; usually the result of a call to rbc()
- ... ignored
- k numeric, the *penalty* per parameter to be used; 'k = 2' is the classical AIC.

append_flow	<i>Append a flow to a CompositeFamily object</i>
-------------	--

Description

Append a flow to a CompositeFamily object

Usage

```
append_flow(family, flow)
```

Arguments

family	a CompositeFamily object
flow	a Flow object

Value

a new CompositeFamily object

Examples

```
append_flow(Normal(0, 1), Translate)

Reduce(append_flow, list(Scale, Translate), init = Normal(0, 1))
```

Bernoulli	<i>Bernoulli distribution as a CompositeFamily</i>
-----------	--

Description

Bernoulli distribution as a CompositeFamily

Usage

```
Bernoulli(prob = 0.5)
```

Arguments

prob	the probability of a success
------	------------------------------

Value

a new BinaryFamily object

Examples

```
dist <- Bernoulli()
dist$probability()
```

```
coef.RegressionByComposition
```

Extract regression coefficients from a regression by composition

Description

Extract regression coefficients from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'
coef(object, ...)
```

Arguments

object	a RegressionByComposition object; usually the result of a call to rbc()
...	ignored

```
fitted.RegressionByComposition
```

Compute fitted values from a regression by composition

Description

Compute fitted values from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'
fitted(object, ...)
```

Arguments

object	a RegressionByComposition object; usually the result of a call to rbc()
...	further arguments passed to the R6 method \$fitted() associated with the model's CompositeFamily

`logLik.RegressionByComposition`*Extract log-likelihood from a regression by composition*

Description

Extract log-likelihood from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'  
logLik(object, ...)
```

Arguments

<code>object</code>	a <code>RegressionByComposition</code> object; usually the result of a call to <code>rbc()</code>
<code>...</code>	ignored

`LogNormal`*Lognormal distribution as a CompositeFamily*

Description

Lognormal distribution as a `CompositeFamily`

Usage

```
LogNormal(meanlog = 0, sdlog = 1)
```

Arguments

<code>meanlog</code>	the mean of the logarithm
<code>sdlog</code>	the standard deviation of the logarithm

Value

a new `ContinuousFamily` object

Examples

```
dist <- LogNormal()  
log(dist$quantile(0.95))
```

Moebius	<i>Moebius flow</i>
---------	---------------------

Description

Moebius flow

Usage

Moebius

Format

An object of class Flow (inherits from R6) of length 6.

Normal	<i>Normal distribution as a CompositeFamily</i>
--------	---

Description

Normal distribution as a CompositeFamily

Usage

Normal(mean = 0, sd = 1)

Arguments

- | | |
|------|------------------------|
| mean | the mean |
| sd | the standard deviation |

Value

a new ContinuousFamily object

Examples

```
dist <- Normal()
dist$quantile(0.95)
```

Power	<i>Power flow</i>
-------	-------------------

Description

Power flow

Usage

Power

Format

An object of class Flow (inherits from R6) of length 6.

predict.RegressionByComposition	<i>Compute predicted values from a regression by composition</i>
---------------------------------	--

Description

Compute predicted values from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'  
predict(object, newdata, ...)
```

Arguments

object	a RegressionByComposition object; usually the result of a call to rbc()
newdata	data.frame containing new data
...	further arguments passed to the R6 method \$fitted() associated with the model's CompositeFamily

rbc

*Fit a regression by composition model***Description**

Fit a regression by composition model

Usage

```
rbc(formula, init, flows, family, data, par, hessian = TRUE)
```

Arguments

formula	a formula object, with model components separated by ' '
init	the initial distribution
flows	a list of flows
family	(optional) an object of class 'CompositeFamily'; if supplied, 'init' and 'flows' are ignored
data	a data frame
par	a vector of starting values
hessian	logical; use Hessian matrix in model fitting?

Value

an rbc object

Examples

```
## Annette Dobson (1990)
## "An Introduction to Generalized Linear Models".
## Page 9: Plant Weight Data.
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
dobson <- data.frame(
  weight = c(ctl, trt),
  group = gl(2, 10, 20, labels = c("Ctl", "Trt"))
)
dobson_fit <- rbc(weight ~ 1 | 1 + group,
  init = Normal(0, 1),
  flows = list(Scale, Translate),
  data = dobson
)

starr_fit <- rbc(
  height ~ 1 | 0 + I((280 + age)^(-1)) | 1 | 1,
  init = LogNormal(),
  flows = list(Power, Moebius, Scale, Translate),
```

```
data = subset(starr, id %in% unique(id)[1:10])
)
```

residuals.RegressionByComposition

Compute 'residuals' from a regression by composition

Description

Compute 'residuals' from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'
residuals(object, ...)
```

Arguments

- object a RegressionByComposition object; usually the result of a call to rbc()
- ... ignored

Value

a vector of probabilities of the same length as the data

Scale	<i>Scale flow</i>
-------	-------------------

Description

Scale flow

Usage

Scale

Format

An object of class Flow (inherits from R6) of length 6.

ScaleOdds	<i>ScaleOdds flow</i>
-----------	-----------------------

Description

ScaleOdds flow

Usage

ScaleOdds

Format

An object of class Flow (inherits from R6) of length 6.

ScaleRisk0	<i>ScaleRisk0 flow</i>
------------	------------------------

Description

ScaleRisk0 flow

Usage

ScaleRisk0

Format

An object of class Flow (inherits from R6) of length 6.

ScaleRisk1	<i>ScaleRisk1 flow</i>
------------	------------------------

Description

ScaleRisk1 flow

Usage

ScaleRisk1

Format

An object of class Flow (inherits from R6) of length 6.

starr	<i>Growth from birth to 3 years in healthy babies in the US</i>
-------	---

Description

Growth from birth to 3 years in healthy babies in the US

Usage

starr

Format

starr:
A data frame with 104,798 rows and 5 columns:
id Anonymized identifier
sex Sex of baby
age Age of baby, in days
height Jittered height of baby, in cm
weight Jittered weight of baby, in kg ...

Source

[doi:10.5061/dryad.4j0zpc8jf](https://doi.org/10.5061/dryad.4j0zpc8jf)

References

[doi:10.1186/s12874024021451](https://doi.org/10.1186/s12874024021451)

summary.RegressionByComposition	<i>Summary of a regression by composition</i>
---------------------------------	---

Description

Summary of a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'  
summary(object, compact = FALSE, ...)
```

Arguments

object	a RegressionByComposition object; usually the result of a call to rbc()
compact	logical; should coefficients from all flows be compressed into a single matrix?
...	ignored

Translate	<i>Translate flow</i>
-----------	-----------------------

Description

Translate flow

Usage

Translate

Format

An object of class Flow (inherits from R6) of length 6.

TranslateRisk1	<i>TranslateRisk1 flow</i>
----------------	----------------------------

Description

TranslateRisk1 flow

Usage

TranslateRisk1

Format

An object of class Flow (inherits from R6) of length 6.

vcov.RegressionByComposition	<i>Extract variance-covariance matrix from a regression by composition</i>
------------------------------	--

Description

Extract variance-covariance matrix from a regression by composition

Usage

```
## S3 method for class 'RegressionByComposition'  
vcov(object, ...)
```

Arguments

object	a RegressionByComposition object; usually the result of a call to rbc()
...	ignored

Index

* datasets

- Moebius, [6](#)
- Power, [7](#)
- Scale, [9](#)
- ScaleOdds, [10](#)
- ScaleRisk0, [10](#)
- ScaleRisk1, [10](#)
- starr, [11](#)
- Translate, [12](#)
- TranslateRisk1, [12](#)

AIC.RegressionByComposition, [2](#)

append_flow, [3](#)

Bernoulli, [3](#)

coef.RegressionByComposition, [4](#)

fitted.RegressionByComposition, [4](#)

logLik.RegressionByComposition, [5](#)

LogNormal, [5](#)

Moebius, [6](#)

Normal, [6](#)

Power, [7](#)

predict.RegressionByComposition, [7](#)

rbc, [8](#)

residuals.RegressionByComposition, [9](#)

Scale, [9](#)

ScaleOdds, [10](#)

ScaleRisk0, [10](#)

ScaleRisk1, [10](#)

starr, [11](#)

summary.RegressionByComposition, [11](#)

Translate, [12](#)

TranslateRisk1, [12](#)

vcov.RegressionByComposition, [12](#)