

# Package: BSW (via r-universe)

February 4, 2025

**Type** Package

**Title** Fitting a Log-Binomial Model using the Bekhit-Schöpe-Wagenpfeil (BSW) Algorithm

**Version** 0.1.1

**Date** 2021-03-20

**Author** Adam Bekhit, Jakob Schöpe, Stefan Wagenpfeil

**Maintainer** Adam Bekhit <imbei@med-imbei.uni-saarland.de>

**Description** Implements a modified Newton-type algorithm (BSW algorithm) for solving the maximum likelihood estimation problem in fitting a log-binomial model under linear inequality constraints.

**License** GPL (>=3)

**Encoding** UTF-8

**URL** <https://github.com/adam-bec/BSW>

**BugReports** <https://github.com/adam-bec/BSW/issues>

**VignetteBuilder** knitr

**Depends** Matrix, matrixStats, quadprog

**Suggests** knitr, rmarkdown

**Imports** methods

**RoxygenNote** 7.1.0

**Repository** <https://uds-mf-imbei.r-universe.dev>

**RemoteUrl** <https://github.com/uds-mf-imbei/bsw>

**RemoteRef** HEAD

**RemoteSha** ebe14a3083455cd2559e46a689d3749490b57773

## Contents

bsw	2
bsw-class	3
coef,bsw-method	4
confint,bsw-method	4
constr	5
gradF	6
hess	6
summary,bsw-method	7
<b>Index</b>	<b>8</b>

---

bsw	<i>Fitting a log-binomial model using the Bekhit-Schöpe-Wagenpfeil (BSW) algorithm</i>
-----	--

---

## Description

bsw() fits a log-binomial model using a modified Newton-type algorithm (BSW algorithm) for solving the maximum likelihood estimation problem under linear inequality constraints.

## Usage

```
bsw(formula, data, maxit = 200L)
```

## Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	A data frame containing the variables in the model.
maxit	A positive integer giving the maximum number of iterations.

## Value

An object of S4 class "bsw" containing the following slots:

call	An object of class "call".
formula	An object of class "formula".
coefficients	A numeric vector containing the estimated model parameters.
iter	A positive integer indicating the number of iterations.
converged	A logical constant that indicates whether the model has converged.
y	A numerical vector containing the dependent variable of the model.
x	The model matrix.
data	A data frame containing the variables in the model.

**Author(s)**

Adam Bekhit, Jakob Schöpe

**References**

Wagenpfeil S (1996) Dynamische Modelle zur Ereignisanalyse. Herbert Utz Verlag Wissenschaft, Munich, Germany

Wagenpfeil S (1991) Implementierung eines SQP-Verfahrens mit dem Algorithmus von Ritter und Best. Diplomarbeit, TUM, Munich, Germany

**Examples**

```
set.seed(123)
x <- rnorm(100, 50, 10)
y <- rbinom(100, 1, exp(-4 + x * 0.04))
fit <- bsw(formula = y ~ x, data = data.frame(y = y, x = x))
summary(fit)
```

---

bsw-class

*S4 Class "bsw"*

---

**Description**

S4 Class "bsw"

**Slots**

`call` An object of class "call".

`formula` An object of class "formula".

`coefficients` A numeric vector containing the estimated model parameters.

`iter` A positive integer indicating the number of iterations.

`converged` A logical constant that indicates whether the model has converged.

`y` A numeric vector containing the dependent variable of the model.

`x` The model matrix.

`data` A data frame containing the variables in the model.

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

coef,bsw-method	<i>Extracting the estimated model parameters of bsw()</i>
-----------------	---

---

**Description**

For objects of class "bsw", coef() extracts the estimated model parameters of bsw().

**Usage**

```
## S4 method for signature 'bsw'
coef(object)
```

**Arguments**

object	An object of class "bsw".
--------	---------------------------

**Value**

A numeric vector containing the estimated model parameters.

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

confint,bsw-method	<i>Estimating confidence intervals of the estimated model parameters of bsw()</i>
--------------------	---

---

**Description**

For objects of class "bsw", confint() estimates confidence intervals of the estimated model parameters of bsw().

**Usage**

```
## S4 method for signature 'bsw'
confint(object, parm, level = 0.95, method = "wald", R = 1000L)
```

**Arguments**

object	An object of class "bsw".
parm	A specification of which model parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all model parameters are considered.
level	A numeric value that indicates the level of confidence.
method	A character giving the estimation method of the confidence intervals ("bca" or "wald").
R	A positive integer giving the number of bootstrap replicates.

**Details**

confint provides Wald (default) and bias-corrected accelerated bootstrap confidence intervals of the estimated model parameters of bsw().

**Value**

A matrix with columns giving the lower and upper confidence limits of each estimated model parameter.

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

constr	<i>Setting the linear inequality constraints for bsw()</i>
--------	--

---

**Description**

constr() sets the linear inequality constraints for bsw().

**Usage**

```
constr(x)
```

**Arguments**

x            A model matrix.

**Value**

A matrix containing the linear inequality constraints for bsw().

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

gradF	<i>Deriving the first derivatives of the log likelihood function of the log-binomial model in bsw()</i>
-------	---

---

**Description**

gradF() derives the first derivatives of the log likelihood function of the log-binomial model.

**Usage**

```
gradF(theta, y, x)
```

**Arguments**

theta	A numeric vector containing the initial values of the model parameters.
y	A numeric vector containing the dependent variable of the model.
x	The model matrix.

**Value**

A numeric vector containing the first derivatives of the log likelihood function of the log-binomial model.

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

hess	<i>Deriving the second partial derivatives of the log likelihood function of the log-binomial model in bsw() (Hessian matrix)</i>
------	---

---

**Description**

hess() derives the second partial derivatives of the log likelihood function of the log-binomial model.

**Usage**

```
hess(theta, y, x)
```

**Arguments**

theta	A numeric vector containing the initial values of the model parameters.
y	A numeric vector containing the dependent variable of the model.
x	The model matrix.

**Value**

A numeric matrix containing the second partial derivatives of the log likelihood function of the log-binomial model (Hessian matrix).

**Author(s)**

Adam Bekhit, Jakob Schöpe

---

summary,bsw-method      *Summarizing the estimated model parameters of bsw()*

---

**Description**

For objects of class "bsw", summary() summarizes the estimated model parameters of bsw().

**Usage**

```
## S4 method for signature 'bsw'  
summary(object)
```

**Arguments**

object            An object of class "bsw".

**Value**

A list containing the following elements:

coefficients	A numeric vector containing the estimated model parameters.
std.err	A numeric vector containing the estimated standard errors of the model parameters.
z.value	A numeric vector containing the estimated z test statistic of the model parameters.
p.value	A numeric vector containing the estimated p values of the model parameters.

**Author(s)**

Adam Bekhit, Jakob Schöpe

# Index

bsw, [2](#)

bsw-class, [3](#)

coef, bsw-method, [4](#)

confint, bsw-method, [4](#)

constr, [5](#)

gradF, [6](#)

hess, [6](#)

summary, bsw-method, [7](#)