

# Package ‘saws’

April 20, 2012

**Type** Package

**Title** Small-Sample Adjustments for Wald tests Using Sandwich Estimators

**Version** 0.9-5.3

**Date** 2012-04-19

**Author** Michael Fay

**Maintainer** Michael Fay <mfay@niaid.nih.gov>

**Description** Tests coefficients with sandwich estimator of variance and with small samples. Regression types supported are gee, cox regression, and conditional logistic regression.

**Depends** R (>= 2.6.0), gee, stats

**Suggests** MASS

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2012-04-20 06:39:50

## R topics documented:

saws-package . . . . .	2
clogistCalc . . . . .	3
dietfat . . . . .	4
lmfitSaws . . . . .	5
mgee . . . . .	6
micefat . . . . .	8
print.saws . . . . .	8
saws . . . . .	9
SDcorn . . . . .	11
<b>Index</b>	<b>12</b>

---

saws-package

*Small-Sample Adjustments for Wald tests Using Sandwich Estimators*

---

## Description

Tests coefficients with sandwich estimator of variance and with small samples. Regression types supported are gee, cox regression, and conditional logistic regression.

## Details

Package: saws  
Type: Package  
Version: 0.9-5.3  
Date: 2012-04-19  
License: GPL-2 or greater

The main function of this package is `saws`, which takes output from some regression models (gee, cox regression, conditional logistic regression) and gives inferences (confidence intervals, p-values) using small sample adjusted sandwich estimators of variance.

In this version of the package, there are two specialized functions for creating regression output for use in the `saws` function: `mgee` and `clogistCalc`. The function `mgee` is a slight modification of the `gee` function that creates the 'u' matrix and the 'omega' array. The 'coefficients' is a vector with p parameter estimates, and is a standard output from the regression. The matrix 'u' is K by p with  $u[i,]$  the ith estimating equation, where there are K approximately independent estimating equations. The array 'omega' is K by p by p where  $\omega[i,,]$  is a p by p matrix estimating  $-du/dbeta$  (here  $\beta$ =coefficients). See Fay and Graubard (2001) for details.

The cox regression function is not included in this version. It should be included in a future version. There is a demo recreating the example in Fay and Graubard (2001).

## Author(s)

M.P. Fay

Maintainer: Michael Fay <mfay@niaid.nih.gov>

## References

Fay and Graubard (2001). Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators. *Biometrics* 57: 1198-1206. (for copy see `vignette("FayGraubard")`)

## Examples

```
library(gee)
data(warpbreaks)
mout<-mgee(breaks~tension,id=wool, data=warpbreaks, corstr="exchangeable")
saws(mout)
```

---

`clogistCalc`*Conditional Logistic Regression fit*

---

## Description

Perform conditional logistic regression with output formatted for input into `saws` which will give confidence intervals and p-values.

## Usage

```
clogistCalc(n, m, x, set, initb = NA, h = 1e-04, maxitr = 15, epsilon = 1e-08, conf.level = 0.95)
clogistLoglike(n, m, x, beta)
clogistInfo(n, m, x, beta, h)
```

## Arguments

<code>n</code>	vector of number at risk
<code>m</code>	vector of number of events
<code>x</code>	matrix of covariates
<code>set</code>	vector of denoting clusters
<code>initb</code>	vector of initial parameter estimates, <code>initb=NA</code> uses unconditional logistic regression for initial estimate
<code>h</code>	small value for numeric integration
<code>maxitr</code>	maximum number of iterations
<code>epsilon</code>	convergence criteria (see details)
<code>conf.level</code>	confidence level for confidence intervals
<code>beta</code>	vector of current parameter estimate

## Details

The main program is `clogistCalc`. It calls `clogistLoglike` and `clogistInfo` which are not to be called explicitly. The function `clogistLoglike` finds the loglikelihood using recursive methods, and `clogistInfo` calculates score vector and information matrix using numerical methods. Both methods are described in Gail, Lubin and Rubinstein (1981), and the `h` value is the same as is defined in that paper.

The algorithm stops when the largest absolute relative change in either the loglikelihood or in any parameter is less than `epsilon`. For parameters close to zero (i.e., less than 0.01 in absolute value) the relative change is defined as `change/0.01`.

**Value**

A list for input into the [saws](#) function, containing the following elements (K=number of clusters, p=number of parameters):

coefficients	p by 1 vector of parameter estimates
u	K by p matrix of scores or estimating equations
omega	K by p by p array of $-1 \times$ information

**Author(s)**

Michael Fay, modeled after a Fortran program by Doug Midthune

**References**

Gail, Lubin and Rubinstein (1981) *Biometrika*, 703-707

**See Also**

See also [saws](#)

**Examples**

```
data(micefat)
cout<-clogistCalc(micefat$N,micefat$NTUM,micefat[,c("fatCal","totalCal")],micefat$cluster)
## usual model based variance
saws(cout,method="dm")
## sandwich based variance with small sample correction
s3<-saws(cout,method="d3")
s3
print.default(s3)
```

---

dietfat

*Mammary Tumors and Different Types of Dietary Fat in Rodents*

---

**Description**

This is a data set from a meta analysis described in Fay, Freedman, Clifford, and Midthune (1997).

**Usage**

```
data(dietfat)
```

**Format**

A data frame with 442 observations on the following 9 variables.

ARTICLE a numeric vector

SET a numeric vector

N a numeric vector

RESTRICT a numeric vector

PN3 a numeric vector

PN6 a numeric vector

PZERO a numeric vector

PMONO a numeric vector

NTUM a numeric vector

**Details**

For relationship of article numbers to references see Article.numbers.txt in the /doc/ directory.

**References**

Fay, MP, Freedman, LS, Clifford, CK, Midthune, DN. Cancer Research 57: 3979-3988.

Fay, MP, Graubard, BI. Biometrics 57: 1198-1206.

**Examples**

```
data(dietfat)
## maybe str(dietfat) ; plot(dietfat) ...
```

---

ImfitSaws

*Linear model function to output extra objects for saws*

---

**Description**

This is a very basic linear model function. It outputs only the objects needed for input into [saws](#).

**Usage**

```
ImfitSaws(x,y)
```

**Arguments**

x	design matrix
y	response vector

**Details**

The `saws` function requires three inputs, the parameter estimates (coefficients), `u`, and `omega`. The value `u` is the  $K$  by  $p$  matrix of estimating equations evaluated at the coefficient, where each row is an independent estimating equation. For the linear model  $u[i,] = x[i,] * \text{residual}[i]$ . The value `omega` is a  $K$  by  $p$  by  $p$  array, where  $\text{omega}[i,]$  is the derivative of the  $i$ th estimating equation with respect to the parameter vector. For the linear model  $\text{omega}[i,] = t(X_i)$

**Value**

A list with the following elements

<code>coefficients</code>	$p$ by 1 coefficient vector
<code>u</code>	$K$ by $p$ matrix of estimating equations
<code>omega</code>	$K$ by $p$ by $p$ array, see details

**Author(s)**

M.P. Fay

**References**

Fay and Graubard (2001). Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators. *Biometrics* 57: 1198-1206. (for copy see `vignette("FayGraubard")`)

**See Also**

`link{lm}`

**Examples**

```
set.seed(1)
n<-20
x1<-rnorm(n)
x2<-factor(c(rep("a", n/2), rep("b", n/2)))
y<-rnorm(n, x1)
out<-lmfitSaws(model.matrix(~x1*x2), y)
saws(out)
```

---

mgee

*Modified gee function to output extra objects for saws*

---

**Description**

This is just the `gee` function from the `gee` package with extra lines added at the end to create a score matrix (i.e., estimating equation) and information array (i.e., minus the derivative of the estimating equation). Since the code is the same (except the additional lines) all help for `gee` applies to `mgee`.

**Usage**

```
mgee(formula = formula(data), id = id, data = parent.frame(), subset, na.action, R = NULL, b = NULL, tol
```

**Arguments**

formula	see gee help
id	see gee help
data	see gee help
subset	see gee help
na.action	see gee help
R	see gee help
b	see gee help
tol	see gee help
maxiter	see gee help
family	see gee help
corstr	see gee help
Mv	see gee help
silent	see gee help
contrasts	see gee help
scale.fix	see gee help
scale.value	see gee help
v4.4compat	see gee help

**Value**

A gee object with two extra elements to the list, u and omega (see [saws](#)).

**Author(s)**

last few lines by M.P. Fay, for the rest see [gee](#) package DESCRIPTION

**See Also**

[gee](#)

**Examples**

```
## example from gee help
data(warpbreaks)
mout<-mgee(breaks~tension,id=wool,data=warpbreaks,corstr="exchangeable")
saws(mout)
```

micefat

*Dietary fat and Mammary tumors in Mice*

---

**Description**

Data from meta analysis of mice bred for spontaneous tumors and their response to different diets. The sources for the data are from the literature and listed in Freedman et al (1990).

**Usage**

```
data(micefat)
```

**Format**

A data frame with 57 observations on the following 5 variables.

NTUM number of mice in group with any mammary tumor

N number of mice in group

fatCal fat calories per day (kcal)

totalCal total calories per day (kcal)

cluster different experiments

**Source**

Freedman, LS, Clifford, C, and Messina, M (1990). Cancer Research 50: 5710-5719.

**Examples**

```
data(micefat)
head(micefat)
```

---

print.saws*Print saws object*

---

**Description**

Prints confidence intervals and p-values from saws object.

**Usage**

```
## S3 method for class 'saws'
print(x, digits = NULL, ...)
```

**Arguments**

x	object of class 'saws'
digits	number of digits
...	other objects passed to print default

---

saws	<i>Small sample Adjustments for Wald-type tests using Sandwich estimator of variance</i>
------	--

---

**Description**

This function takes an object from a regression function and gives confidence intervals and p-values using the sandwich estimator of variance corrected for small samples.

**Usage**

```
saws(x, test = diag(p), beta0 = matrix(0, p, 1),
     conf.level = 0.95, method = c("d3", "d5", "d1", "d2", "d4", "dm"), bound=.75)
```

**Arguments**

x	a list containing three elements: coefficients, u, omega (see details)
test	either a numeric vector giving elements of coefficient to test, or an r by p matrix of constants for testing (see details)
beta0	null parameters for testing (see details)
conf.level	level for confidence intervals
method	one of "d3", "d5", "d1", "d2", "d4", or "dm" (see details)
bound	bound for bias correction, denoted b in Fay and Graubard, 2001

**Details**

Typically, the x object is created in a specialized function. Currently there are three such functions, `link{lmfitSaws}`, `mgee` and `clologistCalc`. The function `lmfitSaws` is a simple linear model function that creates all the output needed. The function `mgee` is a slight modification of the `gee` function that creates the 'u' matrix and the 'omega' array. The 'coefficients' is a vector with p parameter estimates, and is a standard output from the regression. The matrix 'u' is K by p with  $u_{[i,]}$  the  $i$ th estimating equation, where there are K approximately independent estimating equations. The array 'omega' is K by p by p where  $\omega_{[i,,]}$  is a p by p matrix estimating  $-du/d\beta$  (here  $\beta$ =coefficients). See Fay and Graubard (2001) for details.

Suppose that the coefficient vector from the regression is  $\beta$ . Then we test r hypotheses, based on the the matrix product,  $TEST(\beta - \beta_0) = 0$ , where TEST is an r by p matrix. If the argument 'test' is an r by p matrix (where r is arbitrary), then  $TEST = test$ . If 'test' is a vector, then each element of test corresponds to testing that row of  $\beta$  is 0, i.e.,  $TEST <- diag(p)[test,]$ , where p is the length of the coefficient vector. For example,  $test <- c(2,5)$ , tests that  $\beta[2] - \beta_0[2] = 0$  and that  $\beta[5] - \beta_0[5] = 0$ . The alternatives are always two-sided.

There are several methods available. They are all discussed in Fay and Graubard (2001). The naming of the methods follows that paper (see for example Table 1, where `deltam` corresponds to `dm`, etc.):

- dm** the usual model based method which does not use the sandwich, uses a chi squared distribution
- d1** the standard sandwich method which makes no corrections for small samples
- d2** sandwich method, no bias correction, uses F distribution with  $df=dhat$  (see paper)
- d3** (default method:sandwich method, no bias correction, uses F distribution with  $df=dtilde$  (see paper)
- d4** sandwich method, with bias correction, uses F distribution with  $df=dhatH$  (see paper)
- d5** sandwich method, with bias correction, uses F distribution with  $df=dtildeH$  (see paper)

### Value

An object of class 'saws'. A list with elements:

<code>originalCall</code>	call from the original object
<code>method</code>	method used (see details)
<code>test</code>	test matrix (see details)
<code>beta0</code>	beta0 vector (see details)
<code>coefficients</code>	estimated coefficients
<code>df</code>	a vector of estimated degrees of freedom. This will have as many elements as there are coefficients
<code>V</code>	variance-covariance matrix
<code>se</code>	vector of standard errors of the coefficients
<code>t.value</code>	a vector of t-values: $test (coef - beta0)/se$
<code>p.value</code>	a vector of two-sided p-values
<code>conf.int</code>	p by 2 matrix of confidence intervals

### Author(s)

M.P. Fay

### References

Fay and Graubard (2001). Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators. *Biometrics* 57: 1198-1206. (for copy see `vignette("FayGraubard")`)

### See Also

For examples, see [mgee](#) and [clogistCalc](#). See also [print.saws](#)

---

SDcorn

*Mammary tumors in Sprague-Dawley rats fed Corn Oil*

---

### Description

These data are part of a meta analysis to determine how fat calories and total calories effect the changes of getting a mammary tumor.

### Usage

```
data(SDcorn)
```

### Format

A data frame with 104 observations on the following 10 variables.

ARTICLE a numeric vector

NTUM a numeric vector

N a numeric vector

TFA2 a numeric vector

KCA2 a numeric vector

PFC a numeric vector

LOGIT a numeric vector

KCAL a numeric vector

SET a numeric vector

TEMPSET a numeric vector

### Details

Note the adjustment in Fay, Graubard, Freedman, and Midthune (1998) is slightly different from the one in Fay and Graubard (2001) so the [saws](#) does not match exactly with the 1998 paper.

For relationship of article numbers to references see `Article.numbers.txt` in the `/doc/` directory.

### References

Fay, MP, Freedman, LS, Clifford, CK, Midthune, DN. *Cancer Research* 57: 3979-3988.

Fay, MP, Graubard, BI, Freedman, LS, Midthune, DN. *Biometrics* 54: 195-208.

Fay, MP, Graubard, BI. *Biometrics* 57: 1198-1206.

### Examples

```
data(SDcorn)
## maybe str(SDcorn) ; plot(SDcorn) ...
```

# Index

\*Topic **datasets**

dietfat, [4](#)

micefat, [8](#)

SDcorn, [11](#)

\*Topic **htest**

lmfitSaws, [5](#)

saws, [9](#)

\*Topic **misc**

print.saws, [8](#)

\*Topic **nonlinear**

clogistCalc, [3](#)

mgee, [6](#)

\*Topic **package**

saws-package, [2](#)

\*Topic **regression**

saws, [9](#)

clogistCalc, [2](#), [3](#), [9](#), [10](#)

clogistInfo(clogistCalc), [3](#)

clogistLoglike(clogistCalc), [3](#)

dietfat, [4](#)

gee, [6](#), [7](#)

lmfitSaws, [5](#)

mgee, [2](#), [6](#), [9](#), [10](#)

micefat, [8](#)

print.saws, [8](#), [10](#)

saws, [2–7](#), [9](#), [11](#)

saws-package, [2](#)

SDcorn, [11](#)