Package `saws`

December 16, 2019

Type Package

Title Small-Sample Adjustments for Wald Tests Using Sandwich Estimators

Version 0.9-6.2

Date 2019-12-16

Author Michael P. Fay

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

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

Depends R (>= 2.6.0), gee, stats

Suggests MASS

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-16 20:40:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>saws-package</td>
<td>2</td>
</tr>
<tr>
<td>clogistCalc</td>
<td>3</td>
</tr>
<tr>
<td>dietfat</td>
<td>5</td>
</tr>
<tr>
<td>geeUOmega</td>
<td>6</td>
</tr>
<tr>
<td>lmfitSaws</td>
<td>7</td>
</tr>
<tr>
<td>mgee</td>
<td>8</td>
</tr>
<tr>
<td>micefat</td>
<td>9</td>
</tr>
<tr>
<td>print.saws</td>
<td>10</td>
</tr>
<tr>
<td>saws</td>
<td>11</td>
</tr>
<tr>
<td>SDcorn</td>
<td>13</td>
</tr>
</tbody>
</table>

Index 14
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, linear regression, and conditional logistic regression.

Details

Package: saws
Type: Package
Version: 0.9-6.2
Date: 2019-12-16
License: GPL-2 or greater

The main function of this package is `saws`, which takes output from some regression models (gee, linear regression, conditional logistic regression) and gives inferences (confidence intervals, p-values) using small sample adjusted sandwich estimators of variance. Using methods described in Fay and Graubard (2001).

The output from the regression model must be a list including the following three elements: 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.

Since the 'u' matrix and 'omega' array are not normally part of standard output, there are three specialized functions for creating regression output for use in the `saws` function: `mgee` (for gee), `lmfitSaws` (for linear models) and `clogistCalc` (for conditional logistic regression). For example, the function `mgee` does a gee analysis using the gee function of the gee package, takes the output and runs it through the `geeUOmega` function to create the 'u' matrix and the 'omega' array, and adds those onto the output from the gee (in the process other output from the gee may be corrected, see `geeUOmega`).

The cox regression function is not included in this version of the package. Unless there is demand (and I have time) it will not 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>
clogistCalc

References


Examples

```r
library(gee)
data(warpbreaks)
g<-gee(breaks~tension,id=wool, data=warpbreaks, corstr="exchangeable")
guo<-geeUOmega(g)
saws(guo)
```

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

```r
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

- `n`: vector of number at risk
- `m`: vector of number of events
- `x`: matrix of covariates
- `set`: vector of denoting clusters
- `initb`: vector of initial parameter estimates, `initb=NA` uses unconditional logistic regression for initial estimate
- `h`: small value for numeric integration
- `maxitr`: maximum number of iterations
- `epsilon`: convergence criteria (see details)
- `conf.level`: confidence level for confidence intervals
- `beta`: 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*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)
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 Midhune (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, Graubard, BI. Biometrics 57: 1198-1206.

**Examples**

data(dietfat)

### maybe str(dietfat) ; plot(dietfat) ...
Description

This function is normally not to be called directly, but one should usually use `mgee` (see warning below).

This function takes output from the `gee` function from the gee package and creates a score matrix (i.e., estimating equation) and information array (i.e., minus the derivative of the estimating equation). Note the function creates the X matrix assuming the data set is the same as it was for the original call to gee, see Warning section.

Usage

```r
geeUOmega(geeOutput)
```

Arguments

- `geeOutput` object of class gee, output from gee function

Value

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

Warning

It is safer to use the `mgee` function, which internally calls gee then geeUOmega. If you do not use mgee, and instead call geeUOmega directly, there could be a problem if the data set has been changed after the initial gee call. This is because the model matrix (i.e., the X matrix) is not saved as part of the gee object, we must recreate it from the gee call. So it is created assuming that the data argument in gee means the same thing that it did when gee was called. So if you change the data set between the original gee call and using the geeUOmega function, there may be problems.

Note

The function recalculates the fitted.values and the residuals to the gee object, since in gee (version 4.13-18 at least) the fitted.values and residuals can be wrong if there is an offset or if y is a matrix (as in the binomial model).

Author(s)

M.P. Fay, with some lines copied from gee function

See Also

- `gee`, `mgee`
Examples

```r
## example from gee help
data(warpbreaks)
geeout<-gee(breaks~tension,id=wool,data=warpbreaks,corstr="exchangeable")
gu<-geeUOmega(geeout)
saws(guo)
```

Description

This is a very basic linear model function. It outputs only the objects needed for input into `saws`.

Usage

```r
lmfitSaws(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,] * residual[i]`. The value `omega` is a K by p by p array, where `omega[i,]` is the derivative of the ith estimating equation with respect to the parameter vector. For the linear model `omega[i,] = t(Xi)`.

Value

A list with the following elements

- `coefficients`: p by 1 coefficient vector
- `u`: K by p matrix of estimating equations
- `omega`: K by p by p array, see details

Author(s)

M.P. Fay

References

See Also

link{lm}

Examples

```r
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 function calls the `gee` function from the gee package, then applies the `geeUOmega` function to it to create a score matrix (i.e., estimating equation) and information array (i.e., minus the derivative of the estimating equation). Since the `mgee` function just calls the `gee` function all help for `gee` applies to `mgee`.

Usage

```r
mgee(formula = formula(data), id = id, data = parent.frame(),
    subset, na.action, R = NULL, b = NULL, tol = 0.001,
    maxiter = 25, family = gaussian, corstr = "independence",
    Mv = 1, silent = TRUE, contrasts = NULL, scale.fix = FALSE,
    scale.value = 1, v4.4compat = FALSE)
```

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
**Value**

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

**Note**

You can alternatively take the output from gee and apply the `geeUOmega` function. But see the warning for that function.

**Author(s)**

last few lines by M.P. Fay, for the rest see gee package DESCRIPTION

**See Also**

`gee, geeUOmega`

**Examples**

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

---

**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)
print.saws

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


Examples

data(micefat)
head(micefat)

---

print.saws  

Print saws object

Description

Prints confidence intervals and p-values from saws object.

Usage

```r
## 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

Small sample Adjustments for Wald-type tests using Sandwich estimator of variance

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, `lmfitSaws`, `geeUOmega` and `clogistCalc`. The function `lmfitSaws` is a simple linear model function that creates all the output needed. The function `geeUOmega` takes output from the gee function of the gee package and 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.

Suppose that the coefficient vector from the regression is beta. Then we test r hypotheses, based on the the matrix product, TEST (beta-beta0)=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<-(2,5), tests that beta[2]-beta0[2]=0 and that beta[5]-beta0[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:

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

Author(s)

M.P. Fay

References


See Also

For examples, see geeUOmega and clogistCalc. See also print.saws
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, Graubard, BI. Biometrics 57: 1198-1206.

Examples

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

*Topic datasets
dietfat, 5
micefat, 9
SDcorn, 13

*Topic htest
lmfitSaws, 7
saws, 11

*Topic misc
print.saws, 10

*Topic nonlinear
clogistCalc, 3
ggeeUOmega, 6
mgee, 8

*Topic package
saws-package, 2

*Topic regression
saws, 11

clogistCalc, 2, 3, 11, 12
clogistInfo(clogistCalc), 3
clogistLoglike(clogistCalc), 3
dietfat, 5
ggee, 6, 8, 9
ggeeUOmega, 2, 6, 8, 9, 11, 12

lmfitSaws, 2, 7, 11

mgee, 2, 6, 8
micefat, 9

print.saws, 10, 12

saws, 2–4, 6, 7, 9, 11, 13
saws-package, 2
SDcorn, 13