

Package: FABInference (via r-universe)

October 22, 2026

Title FAB p-Values and Confidence Intervals

Version 0.1

Description Frequentist assisted by Bayes (FAB) p-values and confidence interval construction. See Hoff (2019) [<doi:10.48550/arXiv.1907.12589>](https://doi.org/10.48550/arXiv.1907.12589) ``Smaller p-values via indirect information", Hoff and Yu (2019)  [<doi:10.1214/18-EJS1517>](https://doi.org/10.1214/18-EJS1517)  ``Exact adaptive confidence intervals for linear regression coefficients", and Yu and Hoff (2018) [<doi:10.1093/biomet/asy009>](https://doi.org/10.1093/biomet/asy009) ``Adaptive multigroup confidence intervals with constant coverage".

Date 2019-12-27

Author Peter Hoff

Maintainer Peter Hoff [<peter.hoff@duke.edu>](mailto:peter.hoff@duke.edu)

License GPL-3

Imports MASS

LazyData true

RoxygenNote 7.1.1

Repository <https://pdhoff.r-universe.dev>

Date/Publication 2021-02-19 12:38:37 UTC

RemoteUrl <https://github.com/pdhoff/fabinference>

RemoteRef HEAD

RemoteSha 7da4211bf15b26af82dcb58acc68fcfd21298896

Contents

fabtzCI	2
fabzCI	3
glmFAB	4
lmFAB	5
mmleFH	6
mmleFHP	7
qr.lmFAB	8

rssSplit	8
sfabz	9
summary.glmFAB	10
summary.lmFAB	11

Index	13
--------------	-----------

fabtzCI	<i>z-optimal FAB t-interval</i>
---------	---------------------------------

Description

Computation of a 1-alpha FAB t-interval using z-optimal spending function

Usage

```
fabtzCI(y, s, dof, alpha = 0.05, psi = list(mu = 0, tau2 = 1e+05, sigma2 = 1))
```

Arguments

y	a numeric scalar, a normally distributed statistic
s	a numeric scalar, the standard error of y
dof	positive integer, degrees of freedom for s
alpha	the type I error rate, so 1-alpha is the coverage rate
psi	a list of parameters for the spending function, including <ol style="list-style-type: none"> 1. mu, the prior expectation of E[y] 2. tau2, the prior variance of E[y] 3. sigma2 the variance of y

Value

a two-dimensional vector of the left and right endpoints of the interval

Author(s)

Peter Hoff

Examples

```
n<-10
y<-rnorm(n)
fabtzCI(mean(y),sqrt(var(y)/n),n-1)
t.test(y)$conf.int
```

fabzCI	<i>FAB z-interval</i>
--------	-----------------------

Description

Computation of a 1-alpha FAB z-interval

Usage

```
fabzCI(y, mu, t2, s2, alpha = 0.05)
```

Arguments

y	a numeric scalar
mu	a numeric scalar
t2	a positive numeric scalar
s2	a positive numeric scalar
alpha	the type I error rate, so 1-alpha is the coverage rate

Details

A FAB interval is the "frequentist" interval procedure that is Bayes optimal: It minimizes the prior expected interval width among all interval procedures with exact 1-alpha frequentist coverage. This function computes the FAB z-interval for the mean of a normal population with an known variance, given a user-specified prior distribution determined by psi. The prior is that the population mean is normally distributed. Referring to the elements of psi as mu, t2, s2, the prior and population variance are determined as follows:

1. mu is the prior expectation of the mean
2. t2 is the prior variance of the mean
3. s2 is the population variance

Value

a two-dimensional vector of the left and right endpoints of the interval

Author(s)

Peter Hoff

Examples

```
y<-0
fabzCI(y,0,10,1)
fabzCI(y,0,1/10,1)
fabzCI(y,2,10,1)
fabzCI(y,0,1/10,1)
```

glmFAB

FAB inference for generalized linear models

Description

asymptotic FAB p-values and confidence intervals for parameters in generalized linear regression models

Usage

```
glmFAB(cformula, FABvars, lformula = NULL, alpha = 0.05, silent = FALSE, ...)
```

Arguments

cformula	formula for the control variables
FABvars	matrix of regressors for which to make FAB p-values and CIs
lformula	formula for the linking model (just specify right-hand side)
alpha	error rate for CIs (1-alpha CIs will be constructed)
silent	show progress (TRUE) or not (FALSE)
...	additional arguments to be passed to glm

Value

an object of the class glmFAB which inherits from glm

Author(s)

Peter Hoff

Examples

```
# n observations, p FAB variables, q=2 control variables
n<-100 ; p<-25

# X is design matrix for params of interest
# beta is vector of true parameter values
# v a variable in the linking model - used to share info across betas

v<-rnorm(p) ; beta<-(2 - 2*v + rnorm(p))/3 ; X<-matrix(rnorm(n*p),n,p)/8

# control coefficients and variables
alpha1<- .5 ; alpha2<- -.5
w1<-rnorm(n)/8
w2<-rnorm(n)/8

# simulate data
```

```

lp<-1 + alpha1*w1 + alpha2*w2 + X%%beta
y<-rpois(n,exp(lp))

# fit model
fit<-glmFAB(y~w1+w2,X,~v,family=poisson)

fit$FABpv
fit$FABci
summary(fit) # look at p-value column

```

lmFAB

*FAB inference for linear models***Description**

FAB p-values and confidence intervals for parameters in linear regression models

Usage

```

lmFAB(
  cformula,
  FABvars,
  lformula = NULL,
  alpha = 0.05,
  rssSplit = TRUE,
  silent = FALSE
)

```

Arguments

cformula	formula for the control variables
FABvars	matrix of regressors for which to make FAB p-values and CIs
lformula	formula for the linking model (just specify right-hand side)
alpha	error rate for CIs (1-alpha CIs will be constructed)
rssSplit	use some residual degrees of freedom to help fit linking model (TRUE/FALSE)
silent	show progress (TRUE) or not (FALSE)

Value

an object of the class lmFAB which inherits from lm

Author(s)

Peter Hoff

Examples

```
# n observations, p FAB variables, q=2 control variables

n<-100 ; p<-25

# X is design matrix for params of interest
# beta is vector of true parameter values
# v a variable in the linking model - used to share info across betas

v<-rnorm(p) ; beta<-(2 - 2*v + rnorm(p))/3 ; X<-matrix(rnorm(n*p),n,p)/8

# control coefficients and variables
alpha1<- .5 ; alpha2<- -.5
w1<-rnorm(n)/8
w2<-rnorm(n)/8

# simulate data
lp<-1 + alpha1*w1 + alpha2*w2 + X%*%beta
y<-rnorm(n,lp)

# fit model
fit<-lmFAB(y~w1+w2,X,~v)

fit$FABpv
fit$FABci
summary(fit) # look at p-value column
```

mmleFH

Marginal MLEs for the Fay-Herriot model

Description

Marginal MLEs for the Fay-Herriot random effects model where the covariance matrix for the sampling model is known to scale.

Usage

```
mmleFH(y, X, V, ss0 = 0, df0 = 0)
```

Arguments

y	direct data following normal model $y \sim N(\theta, V\sigma^2)$
X	linking model predictors $\theta \sim N(X\beta, \tau^2 I)$
V	covariance matrix to scale
ss0	prior sum of squares for estimate of σ^2
df0	prior degrees of freedom for estimate of σ^2

Value

a list of parameter estimates including

1. beta, the estimated regression coefficients
2. t2, the estimate of τ^2
3. s2, the estimate of σ^2

Author(s)

Peter Hoff

Examples

```
n<-30 ; p<-3
X<-matrix(rnorm(n*p),n,p)
beta<-rnorm(p)
theta<-X%%beta + rnorm(n)
V<-diag(n)
y<-theta+rnorm(n)
mmleFH(y,X,V)
```

mmleFHP

Marginal MLEs for the Fay-Herriot model with known covariance

Description

Marginal MLEs for the Fay-Herriot random effects model where the covariance matrix for the sampling model is known

Usage

```
mmleFHP(y, X, Sigma)
```

Arguments

y	direct data following normal model $y \sim N(\theta, \Sigma)$
X	linking model predictors $\theta \sim N(X\beta, \tau^2 I)$
Sigma	covariance matrix in sampling model

Value

a list of parameter estimates including

1. beta, the estimated regression coefficients
2. t2, the estimate of τ^2

Author(s)

Peter Hoff

Examples

```
n<-30 ; p<-3
X<-matrix(rnorm(n*p),n,p)
beta<-rnorm(p)
theta<-X%%beta + rnorm(n)
Sigma<-diag(n)
y<-theta+rnorm(n)
mmleFHP(y,X,Sigma)
```

qr.lmFAB

QR decomposition

Description

QR decomposition for lmFAB objects

Usage

```
## S3 method for class 'lmFAB'
qr(x, ...)
```

Arguments

x lmFAB object
... see qr.lm, if you can find it

Value

qr decomposition for a design matrix

rssSplit

Residual sum of squares split

Description

Split residual sum of squares from normal linear regression

Usage

```
rssSplit(fit, df0 = max(1, floor(fit$df/10)), seed = -71407)
```

Arguments

<code>fit</code>	lm object
<code>df0</code>	degrees of freedom for the smaller of the two residual sums of squares
<code>seed</code>	random seed for constructing the basis vectors of the split

Value

a two-dimensional vector of independent sums of squares

Author(s)

Peter Hoff

Examples

```
n<-30 ; p<-6 ; sigma2<-1.5
X<-matrix(rnorm(n*p),n,p)
y<-X%*%rnorm(6) + sqrt(sigma2)*rnorm(n)
ss<-rssSplit(lm(y~ -1+X))
df<-as.numeric( substring(names(ss),first=3))
ss/df
```

sfabz	<i>Bayes-optimal spending function</i>
-------	--

Description

Compute Bayes optimal spending function

Usage

```
sfabz(theta, psi, alpha = 0.05)
```

Arguments

<code>theta</code>	value of theta being tested
<code>psi</code>	a list of parameters for the spending function, including <ol style="list-style-type: none"> 1. mu, the prior expectation of $E[y]$ 2. tau2, the prior variance of $E[y]$ 3. sigma2 the variance of y
<code>alpha</code>	level of test

Details

This function computes the value of s that minimizes the acceptance probability of a biased level- α test for a normal population with known variance, under a specified prior predictive distribution.

Value

a scalar value giving the optimal tail-area probability

Author(s)

Peter Hoff

Examples

```
thetas<-seq(-1,1,length=100)
s<-NULL
for(theta in thetas){ s<-c(s,sfabz(theta,list(mu=0,tau2=1,sigma2=1)) ) }
plot(thetas,s,type="l")
```

summary.glmFAB	<i>Summarizing Generalized Linear Model Fits with FAB Inference</i>
----------------	---

Description

summary method for class glmFAB

Usage

```
## S3 method for class 'glmFAB'
summary(
  object,
  dispersion = NULL,
  correlation = FALSE,
  symbolic.cor = FALSE,
  ...
)
```

Arguments

object	an object of class glmFAB
dispersion	see summary.glm
correlation	see summary.glm
symbolic.cor	see summary.glm
...	see summary.glm

Details

A mod of summary.glm that shows FAB p-values in table

Value

A list of summary statistics of the fitted generalized linear model

Examples

```
# n observations, p FAB variables, q=2 control variables

n<-100 ; p<-25

# X is design matrix for params of interest
# beta is vector of true parameter values
# v a variable in the linking model - used to share info across betas

v<-rnorm(p) ; beta<-(2 - 2*v + rnorm(p))/3 ; X<-matrix(rnorm(n*p),n,p)/8

# control coefficients and variables
alpha1<- .5 ; alpha2<- -.5
w1<-rnorm(n)/8
w2<-rnorm(n)/8

# simulate data
lp<-1 + alpha1*w1 + alpha2*w2 + X%*%beta
y<-rpois(n,exp(lp))

# fit model
fit<-glmFAB(y~w1+w2,X,~v,family=poisson)

fit$FABpv
fit$FABci
summary(fit) # look at p-value column
```

summary.lmFAB

*Summarizing Linear Model Fits with FAB Inference***Description**

summary method for class lmFAB

Usage

```
## S3 method for class 'lmFAB'
summary(object, correlation = FALSE, symbolic.cor = FALSE, ...)
```

Arguments

object	an object of class lmFAB
correlation	see summary.lm
symbolic.cor	see summary.lm
...	see summary.lm

Details

A mod of summary.lm that shows FAB p-values in table

Value

A list of summary statistics of the fitted linear model

Examples

```
# n observations, p FAB variables, q=2 control variables
n<-100 ; p<-25

# X is design matrix for params of interest
# beta is vector of true parameter values
# v a variable in the linking model - used to share info across betas

v<-rnorm(p) ; beta<-(2 - 2*v + rnorm(p))/3 ; X<-matrix(rnorm(n*p),n,p)/8

# control coefficients and variables
alpha1<- .5 ; alpha2<- -.5
w1<-rnorm(n)/8
w2<-rnorm(n)/8

# simulate data
lp<-1 + alpha1*w1 + alpha2*w2 + X%*%beta
y<-rnorm(n,lp)

# fit model
fit<-lmFAB(y~w1+w2,X,~v)

fit$FABpv
fit$FABci
summary(fit) # look at p-value column
```

Index

`fabtzCI`, [2](#)

`fabzCI`, [3](#)

`glmFAB`, [4](#)

`lmFAB`, [5](#)

`mmleFH`, [6](#)

`mmleFHP`, [7](#)

`qr.lmFAB`, [8](#)

`rssSplit`, [8](#)

`sfabz`, [9](#)

`summary.glmFAB`, [10](#)

`summary.lmFAB`, [11](#)