Package: RWiener (via r-universe)

September 7, 2024

Version 1.3-3

Date 2020-05-04

Title Wiener Process Distribution Functions

Author Dominik Wabersich [aut, cre]

Maintainer Dominik Wabersich <dominik.wabersich@gmail.com>

Depends R (>= 3.0.0)

Suggests MASS

Imports stats, graphics

License GPL (>= 2)

URL https://github.com/yeagle/RWiener

Description Provides Wiener process distribution functions, namely the Wiener first passage time density, CDF, quantile and random functions. Additionally supplies a modelling function (wdm) and further methods for the resulting object.

Repository https://yeagle.r-universe.dev

RemoteUrl https://github.com/yeagle/rwiener

RemoteRef HEAD

RemoteSha 1431cd2e9b2e8a0d693b9b9cde63dd089dc953a8

Contents

deprecated *Wiener likelihood and criterion functions (deprecated)*

Description

`wiener_likelihood` computes the log-likelihood for given parameter values and data. `wiener_deviance` computes the deviance. `wiener_aic` computes the AIC. `wiener_bic` computes the BIC. These functions can be very useful in combination with the optim funcion, to estimate parameters (see example below).

Usage

```
wiener_likelihood(x, data)
wiener_deviance(x, data)
wiener_aic(x, data, loss=NULL)
wiener_bic(x, data, loss=NULL)
```

Arguments

`x`	vector with the four parameter values: alpha, tau, beta, delta.
`data`	dataframe with data. Needs a reaction time column and a accuracy/response column.
`loss`	Defaults to NULL, which means that the default computation is done by using -2*wiener_likelihood(x,data) in the formula. If not NULL, this can be a function to replace the default -2*wiener_likelihood(x,data) in the code and use a custom function instead.

Details

The described functions are deprecated, but still fully supported. They are kept for backwards compatibility and to ensure one can reproduce the examples from Wabersich & Vandekerckhove (2014).

These functions are simple wrapper functions for the generic R functions reported below.

The User is encouraged to use the generic R functions instead: `logLik.wdm`, `deviance.wdm`, `AIC.wdm`, `BIC.wdm`. See the corresponding help pages for more information on these functions.

References

Wabersich, D., & Vandekerckhove, J. (2014). The RWiener package: An R package providing distribution functions for the Wiener diffusion model. The R Journal, 6(1), 49-56.

Examples

```
### Example 1: Parameter estimation
## generate random data
dat <- rwiener(100,2,.3,.5,0)

## compute likelihood
wiener_likelihood(c(2,.3,.5,0), dat)

## estimate parameters with optim
onm <- optim(c(1,.1,.1,1),wiener_deviance,data=dat, method="Nelder-Mead")
est <- optim(onm$par,wiener_deviance,data=dat, method="BFGS",hessian=TRUE)
est$par # parameter estimates

## the following code needs the MASS package
## Not run: sqrt(diag(MASS::ginv(est$hessian))) # sd for parameters

### Example 2: Simple model comparison
## compare two models with deviance
wiener_deviance(c(3,.3,.5,0), dat)
wiener_deviance(c(3,.3,.5,0.5), dat)
## log-likelihood difference
wiener_likelihood(c(3,.3,.5,0), dat)-wiener_likelihood(c(3,.3,.5,0.5), dat)

## Not run:
### Example 3: likelihood-ratio test and Wald test
## Suppose we have data from 2 conditions
dat1 <- rwiener(100,2,.3,.5,-.5)
dat2 <- rwiener(100,2,.3,.5,.5)
onm1 <- optim(c(1,.1,.1,1),wiener_deviance,data=dat1, method="Nelder-Mead")
est1 <- optim(onm1$par,wiener_deviance,data=dat1, method="BFGS",hessian=TRUE)
wiener_likelihood(est1$par,dat1)+wiener_likelihood(est1$par,dat2) # combined loglike
model_ll <- function(pars,delta,dat1,dat2) {
  wiener_likelihood(pars,dat1)+
    wiener_likelihood(c(pars[1:3],pars[4]+delta),dat2)
}
## likelihood-ratio test
## 0-model: delta=0; alt-model: delta=1
model_ll(est1$par,1,dat1,dat2)
## compute likelihood ratio
LR <- -2*model_ll(est1$par,0,dat1,dat2)+2*model_ll(est1$par,1,dat1,dat2)
## compare with critical X^2(1) quantile, alpha=0.05
LR > qchisq(0.95,1)
## get p-value from X^2(1)
pchisq(LR,1, lower.tail=FALSE)
## Wald-Test
## estimate parameter delta and test for significance
onm2 <- optim(c(1,.1,.1,1),wiener_deviance,data=dat2, method="Nelder-Mead")
est2 <- optim(onm2$par,wiener_deviance,data=dat2, method="BFGS",hessian=TRUE)
delta <- est2$par[4]-est1$par[4]
## the following code needs the MASS package
est1.sd <- sqrt(diag(MASS::ginv(est1$hessian))) # sd for parameters
WT <- (est1$par[4]-(est1$par[4]+delta))/est1.sd[4]
```

```
## compare with critical quantile N(0,1), alpha=0.05
abs(WT) > qnorm(0.975)
## get p-value from N(0,1)
pnorm(WT)

## End(Not run)

### Example 4: Custom AIC loss function
many_drifts <- function(x,datlist) {
  l = 0
  for (c in 1:length(datlist)) {
    l = l + wiener_deviance(x[c(1,2,3,c+3)],datlist[[c]])
  }
  return(l)
}
dat1 <- rwiener(n=100, alpha=2, tau=.3, beta=.5, delta=0.5)
dat2 <- rwiener(n=100, alpha=2, tau=.3, beta=.5, delta=1)
datlist <- list(dat1,dat2)
wiener_aic(x=c(2,.3,.5,.5,1), data=datlist, loss=many_drifts)
```

`likelihood` *Likelihood and criterion functions for wdm*

Description

`logLik.wdm` computes the log-likelihood. `deviance.wdm` computes the deviance. `AIC.wdm` computes the AIC. `BIC.wdm` computes the BIC.

Usage

```
## S3 method for class 'wdm'
logLik(object, ...)
## S3 method for class 'wdm'
deviance(object, ...)
## S3 method for class 'wdm'
AIC(object, ...)
## S3 method for class 'wdm'
BIC(object, ...)
```

Arguments

`object`	a wdm object file or a list containing a `$par` vector with the model parameters, a `$data` data.frame with the data and optionally a `$loss` function.
`...`	optional arguments

Details

The $par vector with the (four) parameter values should be in the following order: alpha, tau, beta, delta.

The $data data.frame with data needs a reaction time column and a accuracy/response column.

The $loss function defaults to NULL, which means that the default computation is done by using the default formula. If not NULL, this can be a function to replace the default computation in the code and use a custom function instead. The custom function takes two arguments: the parameter vector and the data.frame with the data.

These functions can be very useful in combination with the optim funcion, to estimate parameters manually. Check the examples below to see how to use the provided generic functions in a manual estimation routine.

References

Wabersich, D., & Vandekerckhove, J. (2014). The RWiener package: An R package providing distribution functions for the Wiener diffusion model. The R Journal, 6(1), 49-56.

Examples

```
## generate random data
dat <- rwiener(100,3,.25,.5,0.8)

## fit wdm
wdm1 <- wdm(dat, alpha=3, tau=.25, beta=0.5)

## compute likelihood, AIC, BIC, deviance
logLik(wdm1)
AIC(wdm1)
BIC(wdm1)
deviance(wdm1)

## Not run:
## estimate parameters by calling optim manually
## first define necessary wrapper function
nll <- function(x, data) {
  object <- wdm(data, alpha=x[1], tau=x[2], beta=x[3], delta=x[4])
  -logLik(object)
}
## call estimation routine
onm <- optim(c(1,.1,.1,1),nll,data=dat, method="Nelder-Mead")
est <- optim(onm$par,nll,data=dat, method="BFGS",hessian=TRUE)
est$par # parameter estimates
## use the obtained parameter estimates to create wdm object
wdm2 <- wdm(dat, alpha=est$par[1], tau=est$par[2], beta=est$par[3],
  delta=est$par[4])
## now the generic functions can be used again
logLik(wdm2)

## End(Not run)
```

`miscellaneous` *Miscellaneous Wiener Diffusion model functions*

Description

Miscellaneous functions for the RWiener package.

Usage

```
is.wiener(data)
as.wiener(data, yvar=c("q", "resp"))
## S3 method for class 'numdata.wiener'
revamp(data, ...)
## S3 method for class 'data.wiener'
revamp(data, ...)
```

Arguments

`data`	can be a data.wiener and/or data.frame with data (needs a reaction time column and a accuracy/response column). Further it can be a numdata.wiener and/or numeric with the data as single variable (lower bound reaction times are then represented as negative numbers, upper bound reaction times as positive numbers).
`yvar`	represents an optional vector, that can be used to define the names of the reaction time column (first value) and the accuracy/response column (second value), if a data.wiener and/or data.frame is given as data.
`...`	optional arguments: `yvar` (as described above) and `direction`: character string that can be used to define the desired format of the returned data. `"wide"` returns a numdata.wiener, `"long"` returns a data.wiener.

Details

`data.wiener` and `numdata.wiener` are data objects that represent data coming from a Wiener Diffusion process. `data.wiener` uses a data.frame with 2 columns for the 2 response variables (`"q"` and `"resp"` by default). `numdata.wiener` emulates a single variable representation by using a vector, that stores the responses for the upper boundary as positive numbers and the responses for the lower boundary as negative numbers. This is similar to the transformation: Y=(2D-1)RT; where Y is the single variable, that preserves all the information from the decision variable D (1 or 0) and the reaction time variable RT.

The `as.wiener` function can be used to create wiener data objects (`data.wiener` or `numdata.wiener`), that can be used by generic functions, e.g. `plot`.

`is.wiener` checks if the given data is a wiener data object (`data.wiener` or `numdata.wiener`).

`revamp.data.wiener` and `revamp.numdat.wiener` can be used to transform `data.wiener` objects to `numdata.wiener` objects and vice versa. The generic function `revamp(data, ...)` can be called for convenience.

Examples

```
## generate data
dat <- rwiener(100, 4,.35,.5,0.8)

## simple function calls
is.wiener(dat)
dat <- as.data.frame(dat)
dat <- as.wiener(dat)
y <- revamp(dat)
y
revamp(y)
```

`plot` *Wiener plot function*

Description

`plot` creates a density plot of correct and wrong responses for a given dataset.

Usage

```
## S3 method for class 'data.wiener'
plot(x, ...)
```

Arguments

`x`	data.wiener object, which is basically a data.frame with data. Needs a reaction time column and a accuracy/response column.
`...`	Arguments to be passed to methods, such as graphical parameters.

References

Wabersich, D., & Vandekerckhove, J. (2014). The RWiener package: An R package providing distribution functions for the Wiener diffusion model. The R Journal, 6(1), 49-56.

Examples

```
## generate random data
dat <- rwiener(100,2,.3,.5,0)

## plot
plot(dat)
```

RWiener internal *Wiener functions internals*

Description

These functions are not to be used by the user.

scorefun *Extract Empirical Estimating Functions*

Description

!EXPERIMENTAL FUNCTION!

Generic function for extracting the empirical estimating functions of a fitted model.

!EXPERIMENTAL FUNCTION!

Usage

```
scorefun(x, ...)
estfun(x, ...)
```

Arguments

x	a fitted model object.
...	arguments passed to methods.

Value

A matrix containing the empirical estimating functions. Typically, this should be an $n \times k$ matrix corresponding to n observations and k parameters. The columns should be named as in `coef` or `terms`, respectively.

The estimating function (or score function) for a model - `scorefun` - is the derivative of the objective function with respect to the parameter vector. The empirical estimating functions is the evaluation of the estimating function at the observed data (n observations) and the estimated parameters (of dimension k).

The `estfun` function is basically the score function, but with the additional functionality to sum up the scores by the given covariable `id` in the dataset.

References

Zeileis A (2006), Object-Oriented Computation of Sandwich Estimators. *Journal of Statistical Software*, **16**(9), 1–16. URL http://www.jstatsoft.org/v16/i09/.

Examples

```
## generate random data
dat <- rwiener(100,1,.2,.5,0.5)

## fit wdm
wdm1 <- wdm(dat)

## estimating function
scores <- scorefun(wdm1)

## print
head(scores)

## plot
par(mfrow=c(2,2))
plot(scores[,1]);plot(scores[,2]);plot(scores[,3]);plot(scores[,4])
```

`tests` *Wiener Diffusion model test functions*

Description

Calculates test scores and further information for `wdm` model objects.

Usage

```
  ## S3 method for class 'wdm'
anova(object, ..., test="LRT")
  ## S3 method for class 'wdm'
waldtest(object, ..., theta="delta", theta0=0)
```

Arguments

`object`	a wdm model object.
`test`	Statistical test to calculate, so far the only option is a likelihood-ratio test (LRT).
`...`	Further model objects or other arguments passed to methods.
`theta`	the name of the parameter to be tested.
`theta0`	the value of the parameter under the null hypothesis.

Details

The `anova.wdm` function calls the specified test and calculates the test statistics and other information for two or more nested `wdm` model objects.

The `waldtest.wdm` function can be used to conduct a Wald test for a single parameter.

Examples

```
# a random dataset
dat <- rbind(cbind(rwiener(100, 2,.3,.5,0), grp=factor("A", c("A","B"))),
cbind(rwiener(100,2,.3,.5,1), grp=factor("B", c("A","B"))))

# create nested wdm model objects (from specific to general)
wdm1 <- wdm(dat)
wdm2 <- wdm(dat,
            alpha=coef(wdm1)[1], tau=coef(wdm1)[2], beta=coef(wdm1)[3],
            xvar="grp")
wdm3 <- wdm(dat,
            tau=coef(wdm1)[2],
            xvar="grp")

# conduct LRT tests
anova1 <- anova(wdm1,wdm2,wdm3)
anova1

# waldtest
wt1 <- waldtest(wdm1, theta="delta", theta0=0)
wt1
```

---

wdm *Wiener diffusion model fit function*

Description

wdm creates parameter estimates for the four parameters of the Wiener model.

Usage

```
  wdm(data, yvar=c("q", "resp"), alpha=NULL, tau=NULL, beta=NULL,
    delta=NULL, xvar=NULL, start=NULL, fixed=0)
  ## S3 method for class 'wdm'
vcov(object, ..., method="hessian")
```

Arguments

data	is the data object containing data coming from a (hypothetical) Wiener diffusion process. For further details on the data object, see is.wiener.
yvar	represents an optional vector, that can be used to define the names of the reaction time column. For further details on the data object, see is.wiener.
alpha	optional, can be used to fix the alpha parameter to the given value.
tau	optional, can be used to fix the tau parameter to the given value.
beta	optional, can be used to fix the beta parameter to the given value.
delta	optional, can be used to fix the delta parameter to the given value.

xvar	optional: group factor variable to estimate all unfixed parameters separate for the given groups.
start	an optional vector with the four starting parameter values in the following order: alpha, tau, beta, delta.
fixed	a number indicating how many of the parameters are fixed (not free). This number will be subtracted from the number of free parameters. Defaults to 0.
method	the method to use for estimating the covariance matrix of the parameter estimates. Options are "opg" for outer product of gradients or "hessian" to use the hessian matrix from the estimation routine. Defaults to "hessian".
object	a wdm object file or a list containing a $par vector with the model parameters, a $data data.frame with the data and optionally a $loss function.
...	arguments passed to methods.

Details

The wdm function calls an estimation routine, to estimate the model parameters.

If all but one parameters are fixed, a "Brent (optim)" type algorithm is used. For the estimation of more than one parameter, first a "BFGS (optim)" type algorithm is tried, if unsuccessful, a "Newton type (nlm)" algorithm is tried, if again unsuccessful, a "Nelder-Mead (optim)" algorithm is used.

In case all parameters are set to fixed values, no estimation routine is called, but a wdm object will still be created.

The returned wdm object is basically a list containing the parameter estimates in $coefficients. $hessian contains the numerically differentiated Hessian matrix (if available, else NULL). $data contains the data passed to the wdm function call. $loglik contains the log-likelihood value for the wdm object and its parameter estimates. $estpar contains a vector, that is TRUE if the respective parameter was estimated and FALSE if the respective parameter was set to a fixed value. Additional information is given in other list objects.

The standard R functions coef, vcov, confint, summary can be used with wdm objects.

Examples

```
## generate random data
dat <- rbind(cbind(rwiener(100, 2,.3,.5,1), group=factor("A", c("A","B"))),
        cbind(rwiener(100,2,.3,.5,-1), group=factor("B", c("A", "B"))))

## fit wdm
wdm1 <- wdm(dat)

## extract parameters
coef(wdm1)

## further models
wdm2 <- wdm(dat, beta=.5)
wdm3 <- wdm(dat, alpha=wdm1$coefficients[1], tau=wdm1$coefficients[2],
            beta=wdm1$coefficients[3], xvar="group")
```

`wienerdist` *Wiener process distribution functions*

Description

`dwiener` computes the wiener first passage time density. `pwiener` computes the CDF for the wiener first passage time density. `qwiener` computes the quantile for a given CDF value. `rwiener` generates random quantiles from a wiener process distribution, based on the rejection based method. For all functions, the standard deviation of the diffusion process is fixed to 1.

Usage

```
dwiener(q, alpha,tau,beta,delta, resp="upper", give_log=FALSE)
pwiener(q, alpha,tau,beta,delta, resp="upper")
qwiener(p, alpha,tau,beta,delta, resp="upper")
rwiener(n, alpha,tau,beta,delta)
```

Arguments

`q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`alpha`	boundary separation parameter.
`tau`	non-decision time parameter.
`beta`	bias parameter.
`delta`	drift rate parameter.
`resp`	respone: "upper", "lower", or "both"
`give_log`	function returns log, if this argument is TRUE

References

Wabersich, D., & Vandekerckhove, J. (2014). The RWiener package: An R package providing distribution functions for the Wiener diffusion model. The R Journal, 6(1), 49-56.

Examples

```
## calculate density for reactiontime 1.45, upper bound and some parameters
dwiener(1.45, 2,0.3,0.5,0)

## calculate CDF for reactiontime 1.45, upper bound and some parameters
pwiener(1.45, 2,0.3,0.5,0)

## calculate quantile for CDF value of 0.5, upper bound and some parameters
qwiener(0.5, 2,0.3,0.5,0)
```

```
## generate one random value
rwiener(1, 2,0.3,0.5,0)
```

Index