Package 'acc'

October 12, 2022

Type Package

Title Exploring Accelerometer Data

Version 1.3.3

Date 2016-12-15

Author Jaejoon Song, Matthew G. Cox

Maintainer Jaejoon Song <jjsong2@mdanderson.org>

Description Processes accelerometer data from uni-axial and tri-axial devices, and generates data summaries. Also includes functions to plot, analyze, and simulate accelerometer data.

Depends R (>= 2.10), mhsmm

Imports zoo, PhysicalActivity, nleqslv, plyr, methods, DBI, RSQLite, circlize, ggplot2, R.utils, iterators, Rcpp

License GPL (>= 2)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-12-16 23:29:38

R topics documented:

`acc` *Summarizes accelerometer data for multiple types of physical activities*

Description

Summarizes accelerometer for multiple types of physical activities

Usage

```
acc(data, tri, axis, spuriousDef, nonwearDef, minWear,
    patype, pacut, epoch, boutsize, tolerance)
```

Arguments

`data`	Data which consists of two columns [TimeStamp,counts] (i.e. accelerometer counts file read in by function readCounts)
`tri`	Whether the data is from a tri-axial accelerometer. Default is tri='FALSE'. If tri='TRUE' then option 'axis' should be specified.
`axis`	This option is only used for the tri-axial accelerometer. Options are 'x','y','z','sum', or 'vm'. Options 'x', 'y', or 'z' can be specified to summarize data using only data from a single axis. If the option 'vm' is used, the square root of the squared sum of counts from three axes (i.e. $\sqrt{x^2 + y^2 + z^2}$) are used for the summary. If the option 'sum' is used, sum of the counts from three axes are used.
`spuriousDef`	Definition of spurious observation. Defined as minutes of consecutive zeros. For example, if spuriousDef = 20, this means that an observation point will be determined as a spurious observation if there are consequtive counts of at least 20 zeros before and after the single non-zero observation. Default is spuriousDef = 20.
`nonwearDef`	Definition of non-wear time. Defined as minutes of consecutive zeros. For example, if nonwearDef=60, this means that a period will be defined as non-wear time if there are at least 60 consecutive zeros. Default is nonwearDef=60. To consider all observations as wear time specify nonwearDef='Inf'.
`minWear`	Minimum wear time definition. Defined as minutes of wear time. or example, if minWear = 600, this means that a day will be considered valid only if the wear time is at least 600 minutes. Default is minWear = 600. To return summary for all dates in the data, set minWear = 0.

patype	Types of physical activity for summary. For example, to summarize sedentary and moderate-vigorous physical activities, user specifies patype=c('Sedentary','MVPA'). This labels the summary accordingly.
pacut	Cut points to be used for the physical activity type. For example, if the user specified patype=c('Sedentary','MVPA'), pacut can be specified as pacut=c(c(0,99),c(1952,Inf)). The options requires to have a lower and a upper limit for each activity type (i.e. c(0,99) for sedentary activity). The specified interval includes its lower and upper endpoints (it is a closed interval).
epoch	Epoch size. Default is '1 min'. Other epoch size can be specified using this option (e.g., '1 sec')
boutsize	Boutsize to summarize a physical activity. If multiple patype is specified, boutsize should be specified for each one (e.g., if patype=c('Sedentary','MVPA') then one can use boutsize=c(10,10)).
tolerance	Whether two observations outside the physical activity should be permitted in summarizing a physical activity. If multiple patype is specified, tolerance should be for each one (e.g. if patype=c('Sedentary','MVPA') then one can use tolerance=c('FALSE','TRUE')).

Value

summary	Returns summary for each specified physical activity types (number of bouts and minutes of the activity), for valid dates.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Choi, L., Liu, Z., Matthews, C.E. and Buchowski, M.S. (2011). Validation of Accelerometer Wear and Nonwear Time Classification Algorithm. Med Sci Sports Exerc, 43(2):357-64.

Hall, K. S., Howe, C. A., Rana, S. R., Martin, C. L., and Morey, M. C. (2013). METs and Accelerometry of Walking in Older Adults: Standard versus Measured Energy Cost. Medicine and Science in Sports and Medicine, 45(3). 574-82.

Freedson, P., Melanson, E., and Sirard, J. (1998). Calibration of the Computer Sciences and Applications, Inc. accelerometer. Medicine and Science in Sports and Exercercise, 30(5):777-81.

Swartz, A. M., Strath, S. J., Bassett, D. R. Jr., O'Brien, W. L., King, G. A., and Ainsworth, B. E. (2000). Estimation of energy expenditure using CSA accelerometers at hip and wrist sites. Medicine and Science in Sports and Exercercise, 32: S450-456.

Copeland, J. L., and Esliger, D. W. (2009). Accelerometer assessment of physical activity in active, healthy older adults. J Aging Phys Act, 17: 17-30.

Examples

```
##
## Example: Loading accelerometer counts data using readCounts function
##
```

```
## Not run:
library(acc)
infile <- "CountsDataName.dat"
counts <- readCounts(infile)


##
## Example: Summarizing accelerometer data for a sedentary individual
##

# For this example, data is generated using a Hidden Markov model
# First, a sequence of time is generated
randomTime <- seq(ISOdate(2015,4,1),ISOdate(2015,4,3),"min")
# Load the mhsmm package to generate data using a Hidden Makov model
library(mhsmm)
# It is assumed that the counts are generated from a Hidden Markov model
# with three states being non-wear, sedentary, and moderate-vigorous activity
J <- 3; initial <- rep(1/J, J)
# Set up a transition matrix for the Hidden Markov model.
P <- matrix(c(0.95, 0.04, 0.01,
              0.09, 0.9, 0.01,
              0.1, 0.2, 0.7), byrow='TRUE',nrow = J)
# It is assumed that the counts are realized from a mixture of
# two normal distributions (for sedentary activity and mvpa)
# and a constant at zero (for non-wear time).
b <- list(mu = c(0, 30, 2500), sigma = c(0, 30, 1000))
model <- hmmspec(init = initial, trans = P, parms.emission = b,dens.emission = dnorm.hsmm)
# Generate data!
train <- simulate.hmmspec(model, nsim = (60*24*2), seed = 1234, rand.emis = rnorm.hsmm)
# Now set up a dataset that mimicks the accelerometry data
counts <- data.frame(TimeStamp = randomTime[1:length(train$x)], counts = train$x)
library(acc)
# summarize the data using the acc function.
# Sedentary and moderate-vigorous activity is summarized, using Freedson's cut points by default.
summary1 <- acc(data=counts, tri='FALSE', axis=NULL,
                     spuriousDef=20, nonwearDef=60, minWear=600,
                     patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                     boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary1


##
## Example: Summarizing accelerometer data for an active individual.
##

randomTime <- seq(ISOdate(2015,4,1),ISOdate(2015,4,3),"min")
library(mhsmm)
J <- 3; initial <- rep(1/J, J)
P <- matrix(c(0.95, 0.04, 0.01,
              0.09, 0.7, 0.21,
              0.1, 0.1, 0.8), byrow='TRUE',nrow = J)
b <- list(mu = c(0, 30, 2500), sigma = c(0, 30, 1000))
model <- hmmspec(init = initial, trans = P, parms.emission = b,dens.emission = dnorm.hsmm)
```

```
train <- simulate.hmmspec(model, nsim = (60*24*2), seed = 1234, rand.emission = rnorm.hsmm)

counts <- data.frame(TimeStamp = randomTime[1:length(train$x)], counts = train$x)
library(acc)
summary2 <- acc(data=counts, tri='FALSE', axis=NULL,
                    spuriousDef=20, nonwearDef=60, minWear=600,
                    patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                    boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary2


## End(Not run)
```

---

accBatch *Summarizes multiple accelerometer datafiles*

Description

Summarizes multiple accelerometer datafiles in a batch mode. Summary can be provided for multiple types of physical activities, by day.

Usage

```
accBatch(path, tri, axis, spuriousDef, nonwearDef, minWear,
    patype, pacut, epoch, boutsize, tolerance)
```

Arguments

path	Path to accelerometer data files read in by function readCounts or readCounts-Bath. Files in this path can have both uni-axial and tri-axial data. If at least one tri-axial data is present in the path, please specify tri='TRUE' and axis. This information will be used to summarize tri-axial data.
tri	Is there at least one dataset from a tri-axial accelerometer in the folder? Default is tri='TRUE'. If tri='TRUE' then option 'axis' should be specified. Default axis is axis='vm'.
axis	If the data is from a tri-axial device, this option is applied. Options are 'x','y','z','sum', or 'vm'. Options 'x', 'y', or 'z' can be spefied to summarize data using only data from a single axis. If the option 'vm' is used, the square root of the squared sum of counts from three axes (i.e. $\sqrt{x^2 + y^2 + z^2}$) are used for the summary. If the option 'sum' is used, sum of the counts from three axes are used.
spuriousDef	Definition of spurious observation. Defined as minutes of consecutive zeros. For example, if spuriousDef = 20, this means that an observation point will be determined as a spurious observation if there are consequtive counts of at least 20 zeros before and after the single non-zero observation. Default is spuriousDef = 20.

`nonwearDef` Definition of non-wear time. Defined as minutes of consecutive zeros. For example, if nonwearDef=60, this means that a period will be defined as non-wear time if there are at least 60 consecutive zeros. Default is nonwearDef=60. To consider all observations as wear time specify nonwearDef='Inf'

`minWear` Minimum wear time definition. Defined as minutes of wear time. or example, if minWear = 600, this means that a day will be considered valid only if the wear time is at least 600 minutes. Default is minWear = 600. To return summary for all dates in the data, set minWear = 0.

`patype` Types of physical activity for summary. For example, to summarize sedentary and moderate-vigorous physical activities, user specifies patype=c('Sedentary','MVPA'). This labels the summary accordingly.

`pacut` Cut points to be used for the physical activity type. For example, if the user specified patype=c('Sedentary','MVPA'), pacut can be specified as pacut=c(c(0,99),c(1952,Inf)). The options requires to have a lower and a upper limit for each activity type (i.e. c(0,99) for sedentary activity). The specified interval includes its lower and upper endpoints (it is a closed inerval).

`boutsize` Boutsize to summarize a physical activity. If multiple patype is specified, boutsize should be for each one (e.g., if patype=c('Sedentary','MVPA') then one can use boutsize=c(10,10)).

`epoch` Epoch size. Default is '1 min'. Other epoch size can be specified using this option (e.g., '1 sec')

`tolerance` Whether two observations outside the physical activity should be permitted in summarizing a physical activity. If multiple patype is specified, tolerance should be for each one (e.g., if patype=c('Sedentary','MVPA') then one can use tolerance=c('FALSE','TRUE')).

Value

A folder 'summaryfiles' is created within the specified path. In the folder, summary files are saved by the same filenames as in the accelerometer data for valid days which consists of columns [Date, SedentaryMinutes, wearTime, numberOfBoutsSed, mvpaMinutes, numberOfBoutsMVPA]

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Choi, L., Liu, Z., Matthews, C.E. and Buchowski, M.S. (2011). Validation of Accelerometer Wear and Nonwear Time Classification Algorithm. Med Sci Sports Exerc, 43(2):357-64.

Hall, K. S., Howe, C. A., Rana, S. R., Martin, C. L., and Morey, M. C. (2013). METs and Accelerometry of Walking in Older Adults: Standard versus Measured Energy Cost. Medicine and Science in Sports and Medicine, 45(3). 574-82.

Freedson, P., Melanson, E., and Sirard, J. (1998). Calibration of the Computer Sciences and Applications, Inc. accelerometer. Medicine and Science in Sports and Exercercise, 30(5):777-81.

Swartz, A. M., Strath, S. J., Bassett, D. R. Jr., O'Brien, W. L., King, G. A., and Ainsworth, B. E. (2000). Estimation of energy expenditure using CSA accelerometers at hip and wrist sites. Medicine and Science in Sports and Exercercise, 32: S450-456.

Copeland, J. L., and Esliger, D. W. (2009). Accelerometer assessment of physical activity in active, healthy older adults. J Aging Phys Act, 17: 17-30.

Examples

```
##
## Example
##
## Not run:
mypath <- "C:/Accelerometry files/readfiles"
accBatch(path=mypath, tri='TRUE', axis='vm',
                      spuriousDef=20, nonwearDef=60, minWear=600,
                      patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                      boutsize=c(10,10), tolerance=c('FALSE','TRUE'))

## End(Not run)
```

`accSummary` *Summarizes accelerometer data for a single type of physical activity*

Description

Summarizes accelerometer data for a single type of physical activity. Functionality is same as function acc, except that this function provides a summary for a single type of physical activity, with more detailed information.

Usage

```
accSummary(data, tri, axis, spuriousDef, nonwearDef, minWear,
    patype, pacut, epoch, boutsize, tolerance, returnbout)
```

Arguments

`data`	Data which consists of two columns [TimeStamp,counts] (i.e. raw accelerometer file read in by function readRaw)
`tri`	Whether the data is from a tri-axial accelerometer. Default is tri='FALSE'. If tri='TRUE' then option 'axis' should be specified.
`axis`	This option is only used for the tri-axial accelerometer. Options are 'x','y','z','sum', or 'vm'. Options 'x', 'y', or 'z' can be spefied to summarize data using only data from a single axis. If the option 'vm' is used, the square root of the squared sum of counts from three axes (i.e. $\sqrt{x^2 + y^2 + z^2}$ are used for the summary. If the option 'sum' is used, sum of the counts from three axes are used.

`spuriousDef` Definition of spurious observation. Defined as minutes of consecutive zeros. For example, if spuriousDef = 20, this means that an observation point will be determined as a spurious observation if there are consequtive counts of at least 20 zeros before and after the single non-zero observation. Default is spuriousDef = 20.

`nonwearDef` Definition of non-wear time. Defined as minutes of consecutive zeros. For example, if nonwearDef=60, this means that a period will be defined as non-wear time if there are at least 60 consecutive zeros. Default is nonwearDef=60. To consider all observations as wear time specify nonwearDef='Inf'

`minWear` Minimum wear time definition. Defined as minutes of wear time. or example, if minWear = 600, this means that a day will be considered valid only if the wear time is at least 600 minutes. Default is minWear = 600. To return summary for all dates in the data, set minWear = 0.

`patype` Type of physical activity for summary. For example, to summarize sedentary activity, use option patype=c('Sedentary'). To summarize moderate-vigorous physical activities, user specifies patype=c('MVPA'). This labels the summary accordingly.

`pacut` Cut points to be used for the physical activity type. For example, if the user specified patype=c('Sedentary'), pacut can be specified as pacut=c(c(0,99)). The options requires to have a lower and a upper limit for each activity type (i.e. c(0,99) for sedentary activity). The specified interval includes its lower and upper endpoints (it is a closed inerval).

`boutsize` Boutsize to summarize a physical activity. If multiple patype is specified, boutsize should be for each one (e.g., if patype=c('Sedentary') then one can use boutsize=c(10)).

`epoch` Epoch size. Default is '1 min'. Other epoch size can be specified using this option (e.g., '1 sec')

`tolerance` Whether two observations outside the physical activity cut point should be permitted in summarizing a physical activity (e.g. if patype=c('Sedentary') then one can use tolerance=c('FALSE')).

`returnbout` Whether to return data with bout indicators. If returnbout='FALSE' then only a summary of daily physical activity is returned.

Value

If returnbout='FALSE', then a summary for each specified physical activity types (number of bouts and minutes of the activity), for valid dates. Defaults to 'TRUE'.

If returnbout='TRUE', then a list of summary object is returned with:

`totalDates` Number of unique days available in data.

`validDates` A summary for each specified physical activity types (number of bouts and minutes of the activity), for valid dates (as defined by minWear).

`PA` Data supplied by the user is returned with three additional columns [inPA, nonwear, inboutPA], indicating whether the observation was considered to be in the defined cutpoint of the physical activity (inPA), in the defined cutpoint of nonwear time (nonwear), or in bout (inboutPA), respectively.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Choi, L., Liu, Z., Matthews, C.E. and Buchowski, M.S. (2011). Validation of Accelerometer Wear and Nonwear Time Classification Algorithm. Med Sci Sports Exerc, 43(2):357-64.

Hall, K. S., Howe, C. A., Rana, S. R., Martin, C. L., and Morey, M. C. (2013). METs and Accelerometry of Walking in Older Adults: Standard versus Measured Energy Cost. Medicine and Science in Sports and Medicine, 45(3). 574-82.

Freedson, P., Melanson, E., and Sirard, J. (1998). Calibration of the Computer Sciences and Applications, Inc. accelerometer. Medicine and Science in Sports and Exercercise, 30(5):777-81.

Swartz, A. M., Strath, S. J., Bassett, D. R. Jr., O'Brien, W. L., King, G. A., and Ainsworth, B. E. (2000). Estimation of energy expenditure using CSA accelerometers at hip and wrist sites. Medicine and Science in Sports and Exercercise, 32: S450-456.

Copeland, J. L., and Esliger, D. W. (2009). Accelerometer assessment of physical activity in active, healthy older adults. J Aging Phys Act, 17: 17-30.

Examples

```
##
## Example 1: Loading the activity counts data using readCounts function
##
## Not run:
library(acc)
infile <- "DataName.dat"
counts <- readCounts(infile)


##
## Example 2: Summarizing accelerometer data for a sedentary individual"
##

# For this example, data is generated using a Hidden Markov model
# First, a sequence of time is generated
randomTime <- seq(ISOdate(2015,4,1),ISOdate(2015,4,3),"min")
# Load the mhsmm package to generate data using a Hidden Makov model
library(mhsmm)
# It is assumed that the counts are generated from a Hidden Markov model
# with three states, being non-wear, sedentary, and moderate-vigorous activity
J <- 3; initial <- rep(1/J, J)
# Set up a transition matrix for the Hidden Markov model.
P <- matrix(c(0.95, 0.04, 0.01,
              0.09, 0.9, 0.01,
              0.1, 0.2, 0.7), byrow='TRUE',nrow = J)
# It is assumed that the counts are realized from a mixture of
# two normal distributions (for sedentary activity and mvpa)
# and a constant at zero (for non-wear time).
b <- list(mu = c(0, 30, 2500), sigma = c(0, 30, 1000))
model <- hmmspec(init = initial, trans = P, parms.emission = b,dens.emission = dnorm.hsmm)
```

```
# Generate data!
train <- simulate.hmmspec(model, nsim = (60*24*2), seed = 1234, rand.emis = rnorm.hsmm)
# Now set up a dataset that mimicks the accelerometry data
counts <- data.frame(TimeStamp = randomTime[1:length(train$x)], counts = train$x)
library(acc)
# summarize the data using the acc function.
# Sedentary and moderate-vigorous activity is summarized, using Freedson's cut points by default.
# Option returnbout='TRUE' returns a more detailed information on how the summary was calculated.
summary1 <- accSummary(data=counts, tri='FALSE', axis=NULL,
                     spuriousDef=20, nonwearDef=60, minWear=600,
                     patype='MVPA',pacut=c(1952,Inf),
                     boutsize=10, tolerance='TRUE',returnbout='TRUE')
summary1$validDates # This returns the same summary as when returnbout='FALSE'
# summary1$PA # This returns the activity classification and bout information

##
## Example 3: Summarizing accelerometer data for an active individual.
##

randomTime <- seq(ISOdate(2015,4,1),ISOdate(2015,4,3),"min")
library(mhsmm)
J <- 3; initial <- rep(1/J, J)
P <- matrix(c(0.95, 0.04, 0.01,
              0.09, 0.7, 0.21,
              0.1, 0.1, 0.8), byrow='TRUE',nrow = J)
b <- list(mu = c(0, 30, 2500), sigma = c(0, 30, 1000))
model <- hmmspec(init = initial, trans = P, parms.emission = b,dens.emission = dnorm.hsmm)
train <- simulate.hmmspec(model, nsim = (60*24*2), seed = 1234, rand.emission = rnorm.hsmm)

counts <- data.frame(TimeStamp = randomTime[1:length(train$x)], counts = train$x)
library(acc)
# Option returnbout='TRUE' returns a more detailed information on how the summary was calculated.
summary2 <- accSummary(data=counts, tri='FALSE', axis=NULL,
                     spuriousDef=20, nonwearDef=60, minWear=600,
                     patype='MVPA',pacut=c(1952,Inf),
                     boutsize=10, tolerance='TRUE',returnbout='TRUE')
summary2$validDates # This returns the same summary as when returnbout='FALSE'
# summary2$PA # This returns the activity classification and bout information


## End(Not run)
```

---

aee *Creates an aee object*

Description

Creates an aee object for semiparametric regression with augmented estimating equation.

Usage

```
aee(ID, time, minutes)
```

Arguments

ID	Individual identifier
time	Observation time
minutes	Minutes of physical activity

Value

psDF	A data frame, part of original input data frame with variable "ID", "time" and "count"
timeGrid	Ordered distinct observation times in the set of all observation times
panelMatrix	a matrix representation of panel count data, one row per subject, one column per time point in "timeGrid"

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Wang, X. and Yan, J. (2011). Fitting semiparametric regressions for panel count survival data with an R package spef. Computer Methods and Programs in Biomedicine, 104, 278-285.

Wang, X., Ma, S., and Yan, J. (2013). Augmented estimating equations for semiparametric panel count regression with informative observation times and censoring time. Statistica Sinica, 23, 359-381.

Examples

```
## Not run:
# We illustrate the use of function aeexfit
# with the sample data from the National Health and Nutrition Examination Survey (NHANES)
# to examine the association between the cardiorespiratory function (i.e., VO2max)
# and daily minutes of moderate to vigorous physical activity (MVPA).

data(NHANES)
formula <- aee(ID, Day, mvpaMinutes) ~ VO2max
# Standard errors are obtained using sandwich estimation
fitted <- aeexfit(formula = formula, data = NHANES, se = "Sandwich")
summary(fitted)

## End(Not run)
```

`aeefit` *Fits semiparametric regression models for irregularly observed physical activity data*

Description

Fits semiparametric regression models for irregularly observed physical activity data under conditional independent censoring

Usage

```
aeefit(formula, data, weight, se, control=list(), boot)
```

Arguments

`formula`	A formula object as returned by aee.
`data`	A data frame which includes individuals' ID, observation times, and minutes of physical activity since the last observation time.
`weight`	A vector of sampling weights, for each individual.By default, no sampling weights are applied.
`se`	The method of estimating standard errors can be chosen by the argument se. Two options are available: i) the sandwhich estimation (se = 'Sandwich'), or ii) the bootstrap procedure (se = 'Bootstrap').
`control`	A list of control parameters. See 'Details'.
`boot`	The number of resamples generated for the bootstrap procedure.

Details

The control argument is a list athat can supply any of the following components:

- betaInit: Initial value for covariate coefficient, default is 0.
- interval: Initial search interval for solving beta. Default is (-5,5).
- maxIter: Maximum iterations allowed. Default is 150.
- absTol: Absolute tolerance. Default is 1e-6.
- relTol: Relative tolerance. Default is 1e-6.
- a: A tune parameter. Default is .1. In case of gamma fraility, "a" corresponds to the value of both shape and rate parameters.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Wang, X. and Yan, J. (2011). Fitting semiparametric regressions for panel count survival data with an R package spef. Computer Methods and Programs in Biomedicine, 104, 278-285.

Wang, X., Ma, S., and Yan, J. (2013). Augmented estimating equations for semiparametric panel count regression with informative observation times and censoring time. Statistica Sinica, 23, 359-381.

Examples

```
## Not run:
data(NHANES)
# Example of analyzing NHANES data
# Example 1: Not adjusted for sampling weights
nhanesToFit <- NHANES[ which(NHANES$mvpaMinutes!=0), ]
formula <- aee(ID, Day, mvpaMinutes) ~ Age+raceBi+VO2max+Gender
fitted1 <- aeefit(formula=formula, data=nhanesToFit)
summary(fitted1)

## End(Not run)
```

`aeexfit` *Fits semiparametric regression models robust to informative observation times and censoring*

Description

Fits semiparametric regression models robust to informative observation times and censoring

Usage

```
aeexfit(formula, data, weight, se, control=list(), boot)
```

Arguments

`formula`	A formula object as returned by aee.
`data`	A data frame which includes individuals' ID, observation times, and minutes of physical activity since the last observation time.
`weight`	A vector of sampling weights, for each individual.By default, no sampling weights are applied.
`se`	The method of estimating standard errors can be chosen by the argument se. Two options are available: i) the sandwhich estimation (se = 'Sandwich'), or ii) the bootstrap procedure (se = 'Bootstrap').
`control`	A list of control parameters. See 'Details'.
`boot`	The number of resamples generated for the bootstrap procedure.

Details

The control argument is a list athat can supply any of the following components:

- betaInit: Initial value for covariate coefficient, default is 0.
- interval: Initial search interval for solving beta. Default is (-5,5).
- maxIter: Maximum iterations allowed. Default is 150.
- absTol: Absolute tolerance. Default is 1e-6.
- relTol: Relative tolerance. Default is 1e-6.
- a: A tune parameter. Default is .1. In case of gamma fraility, "a" corresponds to the value of both shape and rate parameters.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

References

Wang, X. and Yan, J. (2011). Fitting semiparametric regressions for panel count survival data with an R package spef. Computer Methods and Programs in Biomedicine, 104, 278-285.

Wang, X., Ma, S., and Yan, J. (2013). Augmented estimating equations for semiparametric panel count regression with informative observation times and censoring time. Statistica Sinica, 23, 359-381.

Examples

```
## Not run:

data(NHANES)
formula <- aee(ID, Day, mvpaMinutes) ~ VO2max
# Standard errors are obtained using sandwich estimation
fitted <- aeexfit(formula = formula, data = NHANES, se = "Sandwich")
summary(fitted)

## End(Not run)
```

---

aggAcc *Aggregates multiple accelerometer summary files*

Description

Aggregates multiple accelerometer summary files (e.g. summary files for multiple individuals) by day. This function can be used in sequence with readRawBatch and accBatch.

Usage

```
aggAcc(path)
```

Arguments

path	Path to accelerometer summary files created by function acc or accBatch

Value

A folder 'aggregate' is created within the specified path. In the folder, aggregate.Rdata file (object 'aggregate') consists of columns [ID, Date, SedentaryMinutes, wearTime, numberOfBoutsSed, mvpaMinutes, numberOfBoutsMVPA], where ID is the filename.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## Example
##
## Not run:
mypath <- "C:/Accelerometry files/readfiles/summaryfiles"
aggAcc(mypath)

## End(Not run)
```

NHANES *NHANES data*

Description

NHANES 2003 data

Usage

```
data(NHANES)
```

Format

A data frame with 19323 rows and 11 variables. This consists physical activity data for 2763 individuals.

Details

- ID: Individual's ID
- Day: Order of the seven day measurement perior 1-7.
- SedentaryMinutes: Sedentary minutes per day in 10 minute bouts.
- numberOfBoutsSed: Number of sedentary bouts per day in 10 minute bouts.

- mvpaMinutes: Minutes of moderate to vigorous physical activity per day in 10 minute bouts.
- numberOfBoutsMVPA: Number of moderate to vigorous physical activity bouts per day in 10 minute bouts.
- wearTime: Wear time per day.
- Gender: Participant's gender.
- Age: Participant's age.
- VO2max: Participant's V02max at baseline.
- raceBI: Participant's race (white=1, otherwise=0).

`plotAcc` *Plots accelerometer data*

Description

Plots accelerometer data. This function receives summary object from function accsummary.

Usage

```
plotAcc(object,markbouts)
```

Arguments

`object`	An object returned from either the function accsummary.
`markbouts`	Whether to mark bouts. If markbout='TRUE' a bar along the time axis will indicate whether the epoch was counted as in bout or not. Default is false.

Value

A plot is returned.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
## Not run:
##
## Example: Simulate a dataset for two days, for an individual with low MVPA level.
##
mvpaLowData <- simAcc(timelength=(60*24*2),paLevel='low')
summary <- accSummary(data=mvpaLowData)
summary$validDates
plotAcc(summary,markbouts='FALSE')

##
## Example: Simulate a dataset for two days, for an individual with moderate MVPA level.
```

```
##
mvpaModData <- simAcc(timelength=(60*24*2),paLevel='moderate')
summary <- accSummary(data=mvpaModData, tri='FALSE', axis=NULL,
             spuriousDef=20, nonwearDef=60, minWear=600,
             patype='MVPA',pacut=c(1952,Inf), boutsize=10,
             tolerance='TRUE', returnbout='TRUE')
summary$validDates
plotAcc(summary,markbouts='FALSE')

##
## Example: Simulate a dataset for two days, for an individual with high MVPA level.
##
mvpaHighData <- simAcc(timelength=(60*24*2),paLevel='high')
summary <- accSummary(data=mvpaHighData, tri='FALSE', axis=NULL,
             spuriousDef=20, nonwearDef=60, minWear=600,
             patype='MVPA',pacut=c(1952,Inf), boutsize=10,
             tolerance='TRUE', returnbout='TRUE')
summary$validDates
plotAcc(summary,markbouts='FALSE')


##
## Example: Simulate a tri-axial dataset for five days.
##
  library(acc)
  library(mhsmm)
  seedset=1234
  minutes=(60*24*5)
  randomTime <- seq(ISOdate(2015,1,1),ISOdate(2020,1,1),"min")
  J <- 3; initial <- rep(1/J, J)
  P <- matrix(rep(NA,9),byrow='TRUE',nrow=J)

  P1 <- matrix(c(0.95, 0.04, 0.01,
                 0.09, 0.9, 0.01,
                 0.1, 0.2, 0.7), byrow='TRUE',nrow = J)

  b <- list(mu = c(0, 30, 2500), sigma = c(0, 30, 1000))
  model1 <- hmmspec(init = initial, trans = P1, parms.emis = b,dens.emis = dnorm.hsmm)
  x <- simulate.hmmspec(model1, nsim = (minutes), seed = seedset, rand.emis = rnorm.hsmm)

  seedset=12345
  P2 <- matrix(c(0.95, 0.04, 0.01,
                 0.09, 0.8, 0.11,
                 0.1, 0.1, 0.8), byrow='TRUE',nrow = J)
  model2 <- hmmspec(init = initial, trans = P2, parms.emis = b,dens.emis = dnorm.hsmm)
  y <- simulate.hmmspec(model2, nsim = (minutes), seed = seedset, rand.emis = rnorm.hsmm)

  seedset=123456
  P3 <- matrix(c(0.95, 0.04, 0.01,
                 0.09, 0.8, 0.11,
                 0.1, 0.1, 0.8), byrow='TRUE',nrow = J)
  model3 <- hmmspec(init = initial, trans = P3, parms.emis = b,dens.emis = dnorm.hsmm)
  z <- simulate.hmmspec(model3, nsim = (minutes), seed = seedset, rand.emis = rnorm.hsmm)
```

```
  counts <- data.frame(TimeStamp = randomTime[1:minutes], x=x$x, y=y$x, z=z$x)
  summary <- accSummary(data=counts, tri='TRUE', axis='vm',
                        spuriousDef=20, nonwearDef=60, minWear=600,
                        patype='MVPA',pacut=c(1952,Inf), boutsize=10, tolerance='TRUE',
                        returnbout='TRUE')
summary$validDates

plotAcc(summary,markbouts='FALSE')

## End(Not run)
```

---

racePlot *Race plots of minutes of activity per day*

Description

Race plots of minutes of activity per day, for several activity types

Usage

```
racePlot(summary,title,cex.title,cex.text,cex.center,color)
```

Arguments

summary	An object returned from function acc.
title	Title of the plot
cex.title	Font size of the title
cex.text	Font size of the race track labels
cex.center	Font size of the day indicator in the center of the plot
color	Color of the race tracks

Value

A plot is returned.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
## Not run:
library(acc)
##
## Example: Simulate a dataset for seven days, for an individual with low MVPA level.
##
mvpaModData <- simAcc(timelength=(60*24*7),paLevel='moderate')

summary1 <- acc(data=mvpaModData, tri='FALSE', axis='NULL',
spuriousDef=20, nonwearDef=60, minWear=600,epoch=c('1 min','1 min'),
patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary1

racePlot(summary1,title="Summary of Physical Activity Per Day",
cex.title=1,cex.text=1.2)

## End(Not run)
```

---

readCounts — *Reads counts data in .dat, .agd, or .csv format for Actigraph GT1M and GT3X devices*

Description

Reads counts data in .dat, .agd, or .csv format for Actigraph GT1M and GT3X devices. Device type and epoch is automatically detected and reported in the console.

Usage

```
readCounts(filename)
```

Arguments

`filename`	Speficy full file path and file name. e.g. C:/mydata.dat or C:/mydata.csv

Value

For uni-axial accelerometer (GT1M), two columns are returned, consisting of: [TimeStamp,Counts]
For tri-axial accelerometer (GT3X), four columns are returned, consisting of: [TimeStamp,x,y,z]

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## A example to read counts data
##
## Not run:
accData1 <- readCounts("C:/mydata.agd")
accData1 <- readCounts("C:/mydata.dat")
accData2 <- readCounts("C:/mydata.csv")

## End(Not run)
```

`readCountsBatch` *Reads multiple accelerometer counts data in a folder*

Description

Reads multiple accelerometer counts data in a folder. This is a batch mode of the readCounts function.

Usage

```
readCountsBatch(path,filetype)
```

Arguments

`path`	Path to a folder which contains accelerometer counts data in .dat or .csv format.
`filetype`	Specify whether the data to read is in dat or csv format. Options are either 'dat' or 'csv'. For example if filetype = 'csv' is specified, all csv data will be read and all other types in the same folder will be ignored. By defalt, it is assumed that all files in the specified path are either csv or dat files and are intended to be read.

Value

A folder 'readfiles' is created inside the specified 'path'. In the folder, files are saved by the same filenames as in the raw data.

For uni-axial accelerometer (GT1M), two columns are returned, consisting of: [TimeStamp,Counts]
For tri-axial accelerometer (GT3X), four columns are returned, consisting of: [TimeStamp,x,y,z]

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## Example
##
# filepath to locate the activity counts data files
## Not run:
mypath <- "C:/Accelerometry files"
readCountsBatch(mypath,filetype='csv')

## End(Not run)
```

---

readRaw *Reads raw data in .csv format for GT3X devices*

Description

Reads raw data in .csv format for Actigraph GT3X devices. Device type and epoch is automatically detected and reported in the console. Marks activity intensity based on two existing methods.

Usage

```
readRaw(filepath,type,resting)
```

Arguments

filepath	Speficy full file path and file name. (e.g. 'C:/mydata.csv')
type	Specify how activity intensity is identified. Options: i) 'mad' for Vaha-Ypya method, ii) 'ai' for Bai et al.'s method, or iii) 'resting' to calculate resting state intensity level required for Bai's method
resting	Resting state intensity level, required for Bai et al.'s method.

Value

If the option 'type' is 'mad' or 'ai', five columns are returned, consisting of first four columns [TimeStamp,x,y,z], and the intensity category/level. If the option 'type' is 'resting' then a vector of resting state intensity level is returned.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## A example to read counts data
##
## Not run:
# For Vaha-Ypya et al.'s method
accData1 <- readRaw("C:/mydata.csv",type='mad')
# For Bai et al.'s method with known resting state intensities
accData2 <- readRaw("C:/mydata.csv",type='ai',resting=c(.15,.16,.17))
# Calculating resting state intensity for Bai et al.'s method
restingIntensity <- readRaw("C:/mydata.csv",type='resting')
accData3 <- readRaw("C:/mydata.csv",type='ai',resting=restingIntensity)

## End(Not run)
```

`simAcc` *Simulates accelerometer data based on a hidden Markov model*

Description

Simulates accelerometer data. The simulation function is based on a hidden Markov model, as described in the example for function acc. This function is provided for convenience to generate data from a pre-specified transition probabilities to mimic activity levels of low, moderate and high. To generate data from a specific transition probabilities and distributions, please refer to the example for function acc.

Usage

```
simAcc(timelength,paLevel,epoch,startDate,endDate,
       mu,sigma,seedset,tpm)
```

Arguments

`timelength`	Number of observations to be generated.
`paLevel`	Pre-specified levels of physical activity for convenience. User can specify all parameter as preferred, by stating the option as paLevel=NULL. Default is 'moderate'. Options: 1) 'low', 'moderate', or 'high'. Low specifies a hidden markov model with transition probabilities 0.95, 0.04, 0.01, 0.09, 0.9, 0.01, 0.1, 0.2, 0.7, respectively for P11, P12, P13, P21, P22, P23, P31, P32, P33, respectively. Moderate specifies a hidden markov model with transition probabilities 0.95, 0.04, 0.01, 0.09, 0.8, 0.11, 0.1, 0.1, 0.8 respectively for P11, P12, P13, P21, P22, P23, P31, P32, P33, respectively. High specifies a hidden markov model with transition probabilities 0.95, 0.04, 0.01, 0.09, 0.7, 0.21, 0.1, 0.1, 0.8, respectively for P11, P12, P13, P21, P22, P23, P31, P32, P33, respectively. For all levels, it is assumed that the activity intensities are realized from a mixture

	of two normal distributions (for sedentary activity and mvpa) and a constant at zero (for non-wear time), with means mu = c(0, 30, 2500) and variance sigma = c(0, 30, 1000).
epoch	Epoch size. User can specify desired epoch size in units of time larger than seconds. Defaults to 1 minute epoch.
startDate	Start date in ISOdate format. For example ISOdate(2017,1,1,hour=0,min=0,sec=0,tz="GMT").
endDate	End date in ISOdate format. For example ISOdate(2017,1,1,hour=0,min=0,sec=0,tz="GMT").
mu	Mean levels for each activity type.
sigma	Standard deviations for each activity type.
seedset	Sets seed for random data generation. Defaults to 1234.
tpm	Transition probability matrix that specify probability of change from one activity state to another.

Value

A simulated dataset is returned with two columns: [TimeStamp, counts]

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
## Not run:
##
## Example: Simulate a dataset for two days, for an individual with low activity level.
##
mvpaLowData <- simAcc(timelength=(60*24*2),paLevel='low')
summary <- acc(data=mvpaLowData, tri='FALSE', axis=NULL,
                     spuriousDef=20, nonwearDef=60, minWear=600,
                     patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                     boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary

##
## Example: Simulate a dataset for two days, for an individual with moderate activity level.
##
mvpaLowData <- simAcc(timelength=(60*24*2),paLevel='moderate')
summary <- acc(data=mvpaLowData, tri='FALSE', axis=NULL,
                     spuriousDef=20, nonwearDef=60, minWear=600,
                     patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                     boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary

##
## Example: Simulate a dataset for two days, for an individual with high activity level.
##
mvpaLowData <- simAcc(timelength=(60*24*2),paLevel='high')
summary <- acc(data=mvpaLowData, tri='FALSE', axis=NULL,
```

```
                    spuriousDef=20, nonwearDef=60, minWear=600,
                    patype=c('Sedentary','MVPA'),pacut=c(c(0,99),c(1952,Inf)),
                    boutsize=c(10,10), tolerance=c('FALSE','TRUE'))
summary


## End(Not run)
```

---

simPA *Simulates minutes of physical activity per day*

Description

Simulates minutes of physical activity per day with realistic missing data patterns

Usage

```
simPA(n, type, beta, minday, maxday)
```

Arguments

n	Number of individuals in the simulated data.
type	Whether to simulate data from informative/non-informative observation/censoring patterns. Options: i) 'inf', ii) 'noninf'.
beta	True coefficient for the binary covariate.
minday	Minimun number of observation days.
maxday	Maximum number of observation days.

Value

A simulated dataset is returned with four columns: [ID, day, min, x1, z].

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## Simulauting data for a single individual
## with noninformative observation patterns
##
simdata <- simPA(n=1, beta=1.5,type='noninf', minday=6, maxday=7)
```

summary.aeefit *Summary method for aeefit objects*

Description

Prints estimated parameters for aeefit object

Usage

```
## S3 method for class 'aeefit'
summary(object, digits = 3, dig.tst = 2, ...)
```

Arguments

object	A aeefit object
digits	Minimum number of significant digits to be used for most numbers.
dig.tst	Minimum number of significant digits for the test statistics
...	...

Value

Estimated parameters for aeefit objects

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##Will put an example here
```

summary.aeexfit *Summary method for aeexfit objects*

Description

Prints estimated parameters for aeexfit object

Usage

```
## S3 method for class 'aeexfit'
summary(object, digits = 3, dig.tst = 2, ...)
```

Arguments

object	A aeexfit object
digits	Minimum number of significant digits to be used for most numbers.
dig.tst	Minimum number of significant digits for the test statistics
...	...

Value

Estimated parameters for aeefit objects

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##Will put an example here
```

---

visGroup *Group level plots*

Description

Group level plots of minutes of activity per day

Usage

```
visGroup(data,ID,activity,type,title,yaxis,xaxis,time,legendTitle,
         groupBy,groupFun,levels,heatcol)
```

Arguments

data	Summary data by day
ID	Name of column that identifies groups or individuals
activity	Name of column for activity to be plotted
type	Type of plot. Options 'boxplot' or 'heatmap'
title	Title of the plot
yaxis	Axis title for vertical axis
xaxis	Whether to pring the x axis label. Options are TRUE or FALSE
time	Name of column that contains time index
legendTitle	Title of legend for 'type=heatmap'
groupBy	Option to plot by group, for 'type=heatmap'

groupFun	If groupBy is specified, how are two groups defined? Provide a function to specify a statistic that divides the groups. For example, specify 'median' to divide the two groups by the overall median of the data.
levels	Specifies thresholds for heatmap. For example, user can specify 'levels = c(0,300,600,1440)' to plot a heatmap of four levels.
heatcol	Specifies colors for the heatmap.

Value

A plot is returned.

Author(s)

Jaejoon Song <jjsong2@mdanderson.org>

Examples

```
##
## Example: A box plot
##
## Not run:
library(acc)
data(NHANES)
NHANES_subset <- NHANES[1:300,]
visGroup(
data = NHANES_subset,
activity = 'wearTime',
ID = 'ID',
type = 'boxplot',
title = "Wear time at baseline",
yaxis = "Wear time (minutes per day)",xaxis=TRUE)


##
## Example: A heatmap
##
library(acc)
data(NHANES)
NHANES_subset <- NHANES[1:1000,]

visGroup(
data = NHANES_subset,
ID = 'ID',activity = 'wearTime',
type = 'heatmap',title = "Wear time at baseline",
yaxis = "ID",time = 'Day',
legendTitle = "Wear time",
groupBy = 'VO2max',groupFun = 'median',
levels = c(0,300,600,1440),
heatcol = c("white","mistyrose","lightpink","violetred1"))

## End(Not run)
```

Index