Package ‘PhysicalActivity’

May 19, 2018

Version 0.2-2
Date 2018-05-18
Title Process Accelerometer Data for Physical Activity Measurement
Description It provides a function ‘wearingMarking’ for classification of monitor wear and nonwear time intervals in accelerometer data collected to assess physical activity. The package also contains functions for making plots for accelerometer data and obtaining the summary of various information including daily monitor wear time and the mean monitor wear time during valid days. The revised package version 0.2-1 improved the functions in the previous version regarding speed and robustness. In addition, several functions were added: ‘markDelivery’ can classify days for ActiGraph delivery by mail; ‘markPAI’ can categorize physical activity intensity level based on user-defined cut-points of accelerometer counts. It also supports importing ActiGraph AGD files with ‘readActigraph’ and ‘queryActigraph’ functions. The package also better supports time zones and daylight saving.

License GPL (>= 3)
Depends R (>= 2.10)
Suggests RSQLite, DBI
RoxygenNote 6.0.1
NeedsCompilation no
Author Leena Choi [aut, cre], Cole Beck [aut], Zhouwen Liu [aut], Charles E. Matthews [aut], Maciej S. Buchowski [aut]
Maintainer Leena Choi <leena.choi@Vanderbilt.Edu>
Repository CRAN
Date/Publication 2018-05-18 22:03:34 UTC

R topics documented:

PhysicalActivity-package .................................................. 2
Description

It provides a function `wearingMarking` for classification of monitor wear and nonwear time intervals in accelerometer data collected to assess physical activity. The package also contains functions for making plot for accelerometer data and obtaining the summary of various information including daily monitor wear time and the mean monitor wear time during valid days.

Details

The revised package version 0.2-2 improved the functions in the previous version regarding speed and robustness. In addition, several functions were added: `markDelivery` can classify days for ActiGraph delivery by mail; `markPAI` can categorize physical activity intensity level based on user-defined cut-points of accelerometer counts. It also supports importing ActiGraph AGD files with `readActigraph` and `queryActigraph` functions. The package also better supports time zones and daylight saving.

Classify wear and nonwear time status for accelerometer data by epoch-by-epoch basis by `wearingMarking`.

Classify mail delivery and non-delivery day status for accelerometer data by `markDelivery`.

Three options are available for the package: `pa.validCut=600`, `pa.timeStamp='TimeStamp'`, and `pa.cts='axis'`. When these options are specified (as in `markDelivery`), the other functions will automatically respect these values as defaults. For instance, the count variable in `data(dataSec)` is "counts". Running `options(pa.cts='counts')` allows the user to avoid specifying the "cts" argument in `wearingMarking`. The options for `validCut` and `timeStamp` are rarely changed.

Shiny app called `Actigraph` can be used to visualize accelerometer data and summarize the data. Please see [https://github.com/couthcommander/PhysicalActivityShiny](https://github.com/couthcommander/PhysicalActivityShiny).
Author(s)

Leena Choi <leena.choi@vanderbilt.edu>, Cole Beck <cole.beck@vumc.org>, Zhouwen Liu <zhouwen.liu@vumc.org>, Charles E. Matthews <Charles.Matthews2@nih.gov>, and Maciej S. Buchowski <maciej.buchowski@vanderbilt.edu>

Maintainer: Leena Choi <leena.choi@vanderbilt.edu>

References


Examples

data(dataSec)

mydata1m = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 60)
options(pa.cnts = 'counts') # change cnt variable from "axis1" to "counts"
data1m = wearingMarking(dataset = mydata1m, frame = 90)

sumVct(data1m, id="sdata1m")

plotData(data=data1m)

summaryData(data=data1m, validCut=600, perMinuteCts=1, markingString = "w")

---

**dataCollapser**  
*Collapse Accelerometer Data to a Dataset with a Longer Epoch*

Description

The function collapses counts in data collected with a short epoch to make a data set with a longer epoch. For example, this function collapses data with 1-sec epoch to 10-sec epoch or 1-min epoch data.

Usage

dataCollapser(dataset, TS, by, col, func = sum, ...)

Arguments

- **dataset**: The source dataset, in dataframe format, that needs to be collapsed.
- **TS**: The column name for timestamp.
- **by**: Epoch in seconds for a collapsed dataset. For example, to collapse second data to minute data, set by = 60; to collapse 10-second data to minute data, set by = 60.
- **col**: The column name(s) to collapse. If not provided, will default to all numeric columns.
- **func**: A method for collapsing counts. The default is the summation of counts.
- **...**: Argument settings that to be used by user-defined "func" setting.

Value

A collapsed data with user specified epoch.

Author(s)

Zhouwen Liu <zhouwen.liu@vumc.org>

References


Examples

```r
data(dataSec)

## collapse 1-sec epoch data to 10-sec epoch data
mydata10s = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 10)

## collapse 1-sec epoch data to 1-min epoch data
mydata1m = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 60)
```

---

**dataSec**

**Accelerometer Data Example**

Description

Approximately 3 days of accelerometer data collected with 1-sec epoch in the correct data format that can be used by `wearingMarking` to classify wear and nonwear time.

Usage

dataSec
**deliveryData**

**Format**

A data frame with 238140 observations on the following 2 required variables.

- **TimeStamp**  A character vector, timestamp of accelerometer measurements
- **counts**   A numeric vector, counts as accelerometer measurements

**References**


**Examples**

```r
data(dataSec)
```

---

**deliveryData**

*Data Example for Mail Delivery Day Classification*

**Description**

Approximately 15 days of 3-axis accelerometer data collected with 1-minute epoch.

**Usage**

```r
deliveryData
```

**Format**

A data frame with 20987 observations on the following variables.

- **TimeStamp**  A character vector, timestamp of accelerometer measurements
- **axis1**  A numeric vector, counts from axis1 of 3-axis accelerometer
- **axis2**  A numeric vector, counts from axis2 of 3-axis accelerometer
- **axis3**  A numeric vector, counts from axis3 of 3-axis accelerometer
- **steps**  A numeric vector, the number of steps
- **vm**   A numeric vector, the vector magnitude calculated from counts of axis1, axis2 and axis3

**Author(s)**

Cole Beck <cole.beck@vumc.org>, Leena Choi <leena.choi@Vanderbilt.Edu>

**Examples**

```r
data(deliveryData)
```
Description

This function adds an indicator variable for accelerometer delivery days based on a delivery classification algorithm. The algorithm classifies each day as delivery or non-delivery day within each participant data using summary statistics of accelerometer counts for each day. As the summary statistics, the 95th percentile, mean and standard deviation (sd) of accelerometer counts can be used. Using the summary statistics for each day, the algorithm defines a set of days that are used to estimate the 95% confidence interval (CI) based on t-distribution (default) or normal distribution. The lower bound of the 95% CI is used to classify delivery days; if the summary statistics for a day is below the lower bound of the 95% CI, this day is classified as delivery day. Three methods for defining a set of days are available: trim (default), consecutive, and valid.

Usage

markDelivery(data, cts = getOption("pa.cts"), markingString = "w", window = c("trim", "consecutive", "valid"), method = c("95", "mean", "sd"), validCut = getOption("pa.validCut"), wearThreshold = 300, dist = c("t", "normal"))

Arguments

data Data with classified wear (nonwear) status by wearingMarking.
cts The name of the counts column. The default is “axis1”.
markingString Option for summarizing wear (markingString = “w”) or nonwear time (markingString = “nw”).
window A character. It should be one of ‘trim’, ‘consecutive’, or ‘valid’.
method A character. It should be one of ‘95’, ‘mean’ or ‘sd’.
validCut A cutoff for the total minutes of classified monitor wear time per day to be considered as a valid monitor day.
wearThreshold A numeric value specifying a pseudo-valid day cutoff similar to “validCut”, which is used to define a set of days to estimate the 95% CI.
dist Option for distribution used to calculate the 95% CI.

Value

A data frame with summary information about daily counts.

Author(s)

Cole Beck <cole.beck@vumc.org>, Leena Choi <leena.choi@Vanderbilt.Edu>
markingTime

Examples

data(deliveryData)

options(pa.c = "vm")
wm <- wearingMarking(dataset = deliveryData)

markDelivery(wm)
plotData(data=wm) # days 1, 2, 10 - 15 are delivery or invalid days based on the result above
markDelivery(wm, window='valid', method='mean')
markDelivery(wm, method='mean')
markDelivery(wm, method='sd')

markingTime  Mark Days

Description

This function adds a "day" variable to the source dataset. The day is marked in numeric order, according to the timestamp variable.

Usage

markingTime(dataset, timestamp, startTime = "00:00:00", tz = "UTC")

Arguments

dataset  The source dataset, in dataframe format, which needs to be marked.
timestamp  The column name in the dataset that will be used as timestamp.
startTime  Define the starting time of a day. It must be in the format of "hh:mm:ss".
tz  Local time zone, defaults to UTC.

Value

A dataframe with an extra day marking column.

Author(s)

Zhouwen Liu <zhouwen.liu@vumc.org>

References

**Mark Physical Activity Intensity (PAI) Level**

**Description**

This function adds a physical activity intensity level variable “pai” to the source dataset. The “pai” is an ordered factor variable. It will be NA for nonwear times.

**Usage**

```r
markPAI(data, cts = getOption("pa.cts"), markingString = "w",
        breaks = c(-Inf, 100, 760, 2020, Inf), labels = c("sedentary", "light",
                "moderate", "vigorous"))
```

**Arguments**

- **data**: Data with classified wear (nonwear) status by `wearingMarking`.
- **cts**: The name of the counts column. The default is “axis1”.
- **markingString**: Option for summarizing wear (markingString = “w”) or nonwear time (markingString = “nw”).
- **breaks**: A numeric vector of cut-points. The default cut-points are based on Matthews et al. (2016).
- **labels**: A character vector labelling intensity levels.

**Value**

A data frame with an additional PAI-level column.

**Author(s)**

Cole Beck <cole.beck@vumc.org>, Leena Choi <leena.choi@vanderbilt.edu>

**References**


**Examples**

```r
data(dataSec)

mydata1m = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 60)

data1m = wearingMarking(dataset = mydata1m, perMinuteCts = 1, cts = "counts")```
plotData

markPAI(data = data1m, cts = 'counts')[1:10,]

plotData  

Plot Accelerometer Data over Time

Description

This function makes plot for accelerometer collected data (counts) over time for the whole monitor period, or a user specified time period or day with a midnight marking to separate monitored days.

Usage

plotData(data, day = NULL, start = NULL, end = NULL, 
cts = getOption("pa.cts"), TS = getOption("pa.timeStamp"), 
summary = NULL)

Arguments

data  Data with classified wear and nonwear status from wearingMarking.
day  A part of data during a user specified day for plot.
start  Define a starting time for plot.
end  Define a ending time for plot.
cts  The name of the counts column. The default is "axis1".
TS  The column name for timestamp. The default is "TimeStamp".
summary  List output of summaryData function.

Details

If a local time-zone is specified for wearingMarking, it is possible that daylight savings starts or ends during the period shown. In this case a dotted line will indicate its position and the appropriate time-zone abbreviations will be included.

Value

Plot with midnight marking.

Author(s)

Leena Choi <leena.choi@Vanderbilt.Edu>

See Also

wearingMarking, sumVct, summaryData
Examples

```r
data(dataSec)
mydata1m = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 60)
data1m = wearingMarking(dataset = mydata1m,
    frame = 90,
    perMinuteCts = 1,
    TS = "TimeStamp",
    cts = "counts",
    streamFrame = NULL,
    allowanceFrame= 2,
    newcolname = "wearing")

## change "cts" default from "axis1" to "counts"
options(pa.cts = "counts")
## plot the whole data
plotData(data=data1m)

## plot the data from 60 min to 900 min
plotData(data=data1m, start=60, end=900)

## plot the data for day 2
plotData(data=data1m, day=2)

## include summaryData
sumdat <- summaryData(data=data1m)
plotData(data=data1m, summary=sumdat)

## present daylight saving time change
data(deliveryData)
options(pa.cts = "vm")
wm <- wearingMarking(dataset = deliveryData, TS="TimeStamp", tz="America/Chicago")
sumdat <- summaryData(wm)
plotData(data=wm, summary = sumdat)
## valid data after delivery marking
del <- markDelivery(wm)
sumdat <- summaryData(wm, delivery = del)
plotData(data=wm, summary = sumdat)
```

queryActigraph | Query ActiGraph File

Description

This function executes a SELECT query on an ActiGraph AGD file.

Usage

```r
queryActigraph(datfile, qry)
```
Arguments

`datfile` An AGD file.

`qry` An SQL SELECT statement.

Details

AGD files are actually SQLite databases. This function requires the `RSQlite` package. The user is encouraged to directly interface with the database by creating a connection with the `DBI` package. This has been tested with AGD files produced with ActiLife v6.11.

Value

A data frame with query results.

Author(s)

Cole Beck <cole.beck@vumc.org>

See Also

`readActigraph`

Examples

```r
## Not run:
dat <- queryActigraph("actfile.agd", "SELECT * FROM data LIMIT 5")

queryActigraph("actfile.agd", "SELECT * FROM settings")

## directly interface using DBI package
con <- DBI::dbConnect(RSQLite::SQLite(), "actfile.agd")
DBI::dbListTables(con)
DBI::dbDisconnect(con)

## End(Not run)
```

---

**readActigraph**  
*Read ActiGraph Accelerometer Data*

**Description**

This function reads an ActiGraph AGD file into R as a data frame. If accelerometer data are collected with three axes, it creates vector magnitude (vm). The counts at any axis or "vm" can be used to classify with wear and nonwear time using `wearingMarking`.

**Usage**

```r
readActigraph(datfile, convertTime = TRUE)
```
Arguments

datfile An AGD file.
convertTime Convert the timestamp from a character string into POSIXct.

Details

AGD files are SQLite databases. This function requires the RSQLite package.

Value

A data frame with accelerometer data.

Author(s)

Cole Beck <cole.beck@vumc.org>

See Also

wearingMarking, queryActigraph

Examples

## Not run:
dat <- readActigraph("actfile.agd")
dat1s <- wearingMarking(dataset = dat,
frame = 90,
perMinuteCts = 1,
TS = "TimeStamp",
cts = "axis1",
streamFrame = NULL,
allowanceFrame= 2,
newcolname = "wearing",
getMinuteMarking = FALSE)

## End(Not run)

---

readCountsData Convert Accelerometer Output Data to a Correct Data Format

Description

This function converts accelerometer output data to a correct data format to classify wear and non-wear time using wearingMarking. This function can accept accelerometer output data with various epochs (for example, 1-sec, 10-sec or 1-min). If accelerometer data are collected with three axes, it creates vector magnitude (vm).

Usage

readCountsData(filename, ctPerSec, mode = 0)
**Arguments**

- **filename**
  A filename of accelerometer output to be read.

- **ctPerSec**
  Data collection epoch. This argument tells the program the number of counting will be performed in every second. For examples: for 1-sec epoch data, set ctPerSec = 1; for 10-sec epoch data, set ctPerSec = 1/10; for 1-min epoch data, set ctPerSec = 1/60.

- **mode**
  The mode of the ActiLife dat file. Defaults to 0, and should be listed in the file header.

**Value**

A data frame with the correct format (TimeStamp, counts) to be used for `wearingMarking`.

**Note**

Warning: It can be very slow if accelerometer data were collected with 1-sec epoch for many days.

**Author(s)**

Zhouwen Liu <zhouwen.liu@vumc.org>

**References**


**See Also**

`wearingMarking`

**Examples**

```
# Read accelerometer output and convert to a correct format (TimeStamp, counts)
# Suppose "rawActigraphOutput.dat" is an Actigraph output with header as follows:
# MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
# This raw data with 1-sec epoch can be converted to a correct data format to classify wear and nonwear time using "wearingMarking" by the following code:

# Not run: mydata$ = readCountsData("rawActigraphOutput.dat", ctPerSec=1)
```
summarizeData

Summarize Classified Wear Time by Daily Basis

Description

This function summarizes accelerometer data and the classified wear or nonwear time by daily basis.

Usage

```r
summaryData(data, validCut =getOption("pa.validCut"), perMinuteCts = 1,
markingString = "w", TS = getOption("pa.timeStamp"),
cts = getOption("pa.cts"), delivery = NULL)
```

Arguments

- `data`: Data with classified wear (nonwear) status by `wearingMarking`.
- `validCut`: A cutoff for the total minutes of classified monitor wear time per day to be considered as a valid monitor day.
- `perMinuteCts`: The number of data rows per minute. The default is 1-min epoch (perMinuteCts = 1) and we recommend to use 1-min epoch data for this summary. For examples: for data with 10-sec epoch, set perMinuteCts = 6; for data with 1-sec epoch, set perMinuteCts = 60.
- `markingString`: Option for summarizing wear (markingString = "w") or nonwear time (markingString = "nw").
- `TS`: The column name for timestamp. The default is “TimeStamp”.
- `cts`: The name of the counts column. The default is “axis1”.
- `delivery`: data.frame. Delivery information created by `markDelivery`.

Details

This function summarizes the total number of days, weekdays and weekend days in accelerometer data. It provides the total number of valid days, valid weekdays and valid weekend days based on a user defined cutoff for the total minutes of classified monitor wear time per day. This function also summarizes the classified wear (nonwear) time by day and by valid day, and the mean wear (nonwear) time for valid days during weekday and weekends, and for overall valid days. If mail delivery days are classified by `markDelivery`, it also summarizes the classified delivery (non-delivery) days with argument “delivery”. If “pai” column is present in the data, which can be created by `markPAI`, then physical activity intensity (PAI) level will be summarized in the output.

Value

- `unit`: epoch for data.
- `totalNumDays`: the total number of days in accelerometer data.
summaryData

- totalNumWeekWeekend: the total number of weekdays and weekend days in accelerometer data.
- validCut: a user defined cutoff for the total minutes of classified monitor wear time per day to be considered as a valid monitor day.
- totalValidNumDays: the total number of valid days based on the user defined cutoff ("validCut") for the total minutes of wear time and the classified wear time.
- totalValidNumWeekWeekend: the total number of valid weekdays and valid weekend days based on the user defined cutoff ("validCut") for the total minutes of classified monitor wear time per day.
- wearTimeByDay: the classified total wear (nonwear) time by day.
- deliveryDays: marked delivery days.
- validWearTimeByDay: the classified total wear (nonwear) time by valid day.
- meanWearTimeValidDays: the mean wear (nonwear) time for valid days during weekdays and weekends.
- meanWearTimeOverallValidDays: the mean wear (nonwear) time for overall valid days.
- dayInfo: information about wear time and mean counts for each day.
- intensity: optional output depending on "pai" column in the data; the total time in hours of physical activity intensity by day.
- meanValidIntensity: optional output depending on "pai" column in the data; the mean physical activity intensity (PAI) level for valid days.

Author(s)
Cole Beck <cole.beck@vumc.org>, Leena Choi <leena.choi@Vanderbilt.Edu>

References

See Also
wearingMarking, sumVct, markPAI, markDelivery

Examples

data(dataSec)

mydata1m = dataCollapser(dataSec, TS = "TimeStamp", col = "counts", by = 60)

data1m = wearingMarking(dataset = mydata1m,
    frame = 90,
    perMinuteCts = 1,
sumVct

**Summarize Wear and Nonwear Time Interval**

**Description**

This function summarizes the classified wear (nonwear) time by interval basis from the epoch-by-epoch classified wear (nonwear) status classified by `wearingMarking`.

**Usage**

```r
sumVct(dataVct, wearing = "wearing", TS = getOption("pa.timeStamp"), markingString = "w", by = "days", id = NULL)
```

**Arguments**

- **dataVct**: Data with classified wear (nonwear) status classified by `wearingMarking`.
- **wearing**: The column name for classified wear and nonwear status. The default is "wearing".
- **TS**: The column name for timestamp. The default is "TimeStamp".
- **markingString**: Option for summarizing wear (markingString="w") or nonwear time interval (markingString="nw").
- **by**: A sequence of days for classified wear (nonwear) time intervals.
- **id**: Optional output for subject identification or file name.

**Value**

The summary data for wear and nonwear time intervals.
wearingMarking

Author(s)
Leena Choi <leena.choi@vanderbilt.edu>, Cole Beck <cole.beck@vumc.org>, Zhouwen Liu <zhouwen.liu@vumc.org>, Charles E. Matthews <Charles.Matthews@nih.gov>, and Maciej S. Buchowski <maciej.buchowski@vanderbilt.edu>

References

See Also
wearingMarking, summaryData

Examples

```r
data(dataSec)
mydata1m = dataCollapse(dataSec, TS = "TimeStamp", col = "counts", by = 60)

data1m = wearingMarking(dataset = mydata1m,
    frame = 90,
    perMinuteCts = 1,
    TS = "TimeStamp",
    cts = "counts",
    streamFrame = NULL,
    allowanceFrame = 2,
    newcolname = "wearing")

sumVct(data1m, id="sdata1m")
sumVct(data1m, id="sdata1m", markingString = "nw")
```

wearingMarking  Classify Wear and Nonwear Time for Accelerometer Data

Description
This function classifies wear and nonwear time status for accelerometer data by epoch-by-epoch basis.

Usage

```r
wearingMarking(dataset, frame = 90, perMinuteCts = 60,
    TS =getOption("pa.timeStamp"), cts = getOption("pa.cts"),
    streamFrame = NULL, allowanceFrame = 2, newcolname = "wearing",
    getMinuteMarking = FALSE, dayStart = "00:00:00", tz = "UTC", ...)
```
wearingMarking

Arguments

- **dataset**: The source dataset, in dataframe format, which needs to be marked.
- **frame**: The size of time interval to be considered; Window 1 described in Choi et al. (2011). The default is 90.
- **perMinuteCts**: The number of data rows per minute. The default is 1-sec epoch (perMinuteCts = 60). For examples: for data with 10-sec epoch, set perMinuteCts = 6; for data with 1-min epoch, set perMinuteCts = 1.
- **TS**: The column name for timestamp. The default is “TimeStamp”.
- **cts**: The column name for counts. The default is “axis1”.
- **streamFrame**: The size of time interval that the program will look back or forward if activity is detected; Window 2 described in Choi et al. (2011). The default is the half of the frame.
- **allowanceFrame**: The size of time interval that zero counts are allowed; the artifactual movement interval described in Choi et al. (2011). The default is 2.
- **newcolname**: The column name for classified wear and nonwear status. The default is “wearing”. After the data is processed, a new field will be added to the original dataframe. This new field is an indicator for the wearing (“w”) or nonwearing (“nw”).
- **getMinuteMarking**: Return minute data with wear and nonwear classification. If the source is not a minute dataset, the function will collapse it into minute data. The default is FALSE.
- **dayStart**: Define the starting time of day. The default is the midnight, "00:00:00". It must be in the format of "hh:mm:ss".
- **tz**: Local time zone, defaults to UTC.
- **...**: Parameter settings that will be used in `dataCollapser` function.

Details

A detailed description of the algorithm implemented in this function is described in Choi et al. (2011).

Value

A data frame with the column for wear and nonwear classification indicator by epoch-by-epoch basis.

Note

Warning: It will be very slow if accelerometer data with 1-sec epoch for many days are directly classified. We recommend to collapse a dataset with 1-sec epoch to 1-min epoch data using `dataCollapser` and then classify wear and nonwear status using a dataset with a larger epoch.
wearingMarking

Author(s)

Leena Choi <leena choi@vanderbilt.edu>, Cole Beck <cole.beck@vumc.org>, Zhouwen Liu <zhouwen liu@vumc.org>, Charles E. Matthews <Charles Matthews@nih.gov>, and Maciej S. Buchowski <maciej buchowski@vanderbilt.edu>

References


See Also

readCountsData, sumVct

Examples

data(dataSec)

## mark data with 1-min epoch
mydata1m = dataCollapsers(dataSec, TS = "TimeStamp", col = "counts", by = 60)

data1m = wearingMarking(dataset = mydata1m,
                      frame = 90,
                      perMinuteCts = 1,
                      TS = "TimeStamp",
                      cts = "counts",
                      streamFrame = NULL,
                      allowanceFrame = 2,
                      newcolname = "wearing")

sumVct(data1m, id = "dataid")

## mark data with 1-sec epoch
## Not run:
data1s = wearingMarking(dataset = dataSec,
                      frame = 90,
                      perMinuteCts = 60,
                      TS = "TimeStamp",
                      cts = "counts",
                      streamFrame = NULL,
                      allowanceFrame = 2,
                      newcolname = "wearing",
                      getMinuteMarking = FALSE)

sumVct(data1s, id = "dataid")
sumVct(data1s, id = "dataid", markingString = "nw")

## End(Not run)
Index

*Topic accelerometer
  PhysicalActivity-package, 2
*Topic and
  PhysicalActivity-package, 2
*Topic classification,
  PhysicalActivity-package, 2
*Topic classification
  PhysicalActivity-package, 2
*Topic datasets
  dataSec, 4
deliveryData, 5
*Topic data
  PhysicalActivity-package, 2
*Topic day
  PhysicalActivity-package, 2
*Topic delivery
  PhysicalActivity-package, 2
*Topic mail
  PhysicalActivity-package, 2
*Topic nonwear
  PhysicalActivity-package, 2
*Topic process,
  PhysicalActivity-package, 2
*Topic wear
  PhysicalActivity-package, 2

dataCollapser, 3, 18
dataSec, 4
deliveryData, 5

markDelivery, 2, 6, 14, 15
markingTime, 7
markPlai, 2, 8, 14, 15

PhysicalActivity
  (PhysicalActivity-package), 2
PhysicalActivity-package, 2
plotData, 9

queryActigraph, 2, 10, 12

readActigraph, 2, 11, 11
readCountsData, 12, 19

summaryData, 9, 14, 17
sumVct, 9, 15, 16, 19

wearingMarking, 2, 6, 8, 9, 11–17, 17