Package 'TwoRegression'

March 19, 2018

Type Package

Title Process Data from Wearable Research Devices Using Two-Regression Algorithms

Version 0.1.2

Depends R (>= 2.10)

Description Application of two-regression algorithms for wearable research devices. It provides an easy way for users to read in device data files and apply an appropriate two-regression algorithm. More information is available from Hibbing PR, LaMunion SR, Kaplan AS, & Crouter SE (2017) <doi:10.1249/MSS.0000000000001532>.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Imports data.table (>= 1.10.4), dplyr (>= 0.5.0), seewave (>= 2.0.5), magrittr (>= 1.5), utils (>= 3.2.4), stats (>= 3.2.4)

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/paulhibbing/TwoRegression

BugReports https://github.com/paulhibbing/TwoRegression/issues

NeedsCompilation no

Author Paul R. Hibbing [aut, cre],
Vincent T. van Hees [ctb]

Maintainer Paul R. Hibbing <paulhibbing@gmail.com>

Repository CRAN

Date/Publication 2018-03-19 11:16:07 UTC

R topics documented:

`all_data` *Two-regression-ready data frame*

Description

A dataset with pre-processed primary accelerometer and IMU data that is ready for applying a two-regression algorithm.

Usage

```
all_data
```

Format

A data frame with 299 rows and 17 variables:

PID Participant ID

file_source_PrimaryAccel The filename of the primary accelerometer file

date_processed_PrimaryAccel The date the primary accelerometer file was processed

file_source_IMU The filename of the IMU file

date_processed_IMU The date the IMU file was processed

Timestamp The corresponding time for each row of data

day_of_year The numeric day of the year, i.e., the Julian date

minute_of_day The numeric minute of the day

ENMO Euclidian Norm Minus One, in milli-g

Gyroscope_VM_DegPerS Gyroscope vector magnitude, in degrees per second

mean_abs_Gyroscope_x_DegPerS Rotation in x axis, degrees per second

mean_abs_Gyroscope_y_DegPerS Rotation in y axis, degrees per second

mean_abs_Gyroscope_z_DegPerS Rotation in z axis, degrees per second

mean_magnetometer_direction Cardinal direction of magnetometer signal, averaged over one second

ENMO_CV10s Coefficient of variation per 10-s, applied to Euclidian Norm Minus One

GVM_CV10s Coefficient of variation per 10-s, applied to gyroscope vector magnitude

Direction Direction changes per 5-s

`get_cvPER` *Calculate coefficient of variation in sliding windows*

Description

Calculates coefficient of variation using the approach of Crouter et al. (2010, *Med Sci Sports Exerc*)

Usage

```
get_cvPER(big_data, window_secs = 10, Algorithm, verbose = FALSE)
```

Arguments

`big_data` a numeric vector on which to perform the calculation

`window_secs` size of the sliding window, in seconds

`Algorithm` A numeric vector giving the algorithm(s) to apply to the data from the primary accelerometer and (if applicable) IMU

`verbose` A logical scalar: print progress updates?

Value

a numeric vector of values, giving the lowest coefficient of variation among the sliding windows that correspond to each epoch of data

Examples

```
data(raw_for_cv)
get_cvPER(raw_for_cv$ENMO, Algorithm = 1)
```

get_directions *Calculate direction changes per five seconds in sliding windows*

Description

Calculate direction changes per five seconds in sliding windows

Usage

```
get_directions(big_data, window_secs = 5)
```

Arguments

big_data	a numeric vector on which to perform the calculation
window_secs	size of the sliding window, in seconds

Value

a numeric vector of values, giving the number of direction changes in the sliding window that corresponds to each epoch of data

Examples

```
## Not run:
##All possible directions
directions <-
  c("N", "NNE", "NE", "ENE",
    "E", "ESE", "SE", "SSE",
    "S", "SSW", "SW", "WSW",
    "W", "WNW", "NW", "NNW")

##Reproducible results
set.seed(55)
direction_vector <- sample(directions, 50, replace = TRUE)

##Vector of direction changes per 5-s. First and last two values are always NA
get_directions(direction_vector)

## End(Not run)
```

hibbing18_twoReg_process

Process GT9X Files with Hibbing Two-Regression Algorithms

Description

Process GT9X primary accelerometer and (if applicable) IMU files using one or more of the algorithms from Hibbing et al. (2018, *Med Sci Sports Exerc*).

Usage

```
hibbing18_twoReg_process(RAW, IMU = NULL, Wear_Location = c("Hip",
  "Left Wrist", "Right Wrist", "Left Ankle", "Right Ankle"), PID,
  Algorithm = 1, verbose = FALSE, IMU_ignore_A1 = TRUE)
```

Arguments

`RAW`	A character scalar giving path to primary accelerometer data file
`IMU`	A character scalar giving path to IMU data file
`Wear_Location`	A character scalar indicating the device's attachment site
`PID`	A character scalar giving the participant identification
`Algorithm`	A numeric vector giving the algorithm(s) to apply to the data from the primary accelerometer and (if applicable) IMU
`verbose`	A logical scalar: print progress updates?
`IMU_ignore_A1`	A logical scalar. If Algorithm = 1, should IMU files be ignored?

Value

A data frame giving the data and predictions

Examples

```
## Not run:
raw_file <-
    system.file("extdata",
        "TestID_LeftWrist_RAW.csv",
        package = "TwoRegression")

imu_file <-
    system.file("extdata",
        "TestID_LeftWrist_IMU.csv",
        package = "TwoRegression")

wear <- "Left Wrist"
id <- "Test"
alg <- 1:2
```

```
hibbing18_twoReg_process(raw_file, imu_file, wear, id, alg)

## End(Not run)
```

imu_to_check *IMU data to check*

Description

A dataset for demonstrating checks that are applied to IMU data.

Usage

```
imu_to_check
```

Format

A data frame with 300 rows and 8 variables:

file_source_IMU The filename of the IMU file

date_processed_IMU The date the IMU file was processed

Timestamp The corresponding time for each row of data

Gyroscope_VM_DegPerS Gyroscope vector magnitude, in degrees per second

mean_abs_Gyroscope_x_DegPerS Rotation in x axis, degrees per second

mean_abs_Gyroscope_y_DegPerS Rotation in y axis, degrees per second

mean_abs_Gyroscope_z_DegPerS Rotation in z axis, degrees per second

mean_magnetometer_direction Cardinal direction of magnetometer signal, averaged over one second

imu_to_collapse *IMU data to collapse*

Description

A partially-processed IMU dataset ready to be collapsed from raw samples to one-second summaries.

Usage

```
imu_to_collapse
```

Format

A data frame with 1500 rows and 17 variables:

Timestamp The corresponding time for each row of data

Accelerometer.X Secondary accelerometer x-axis data, in G

Accelerometer.Y Secondary accelerometer y-axis data, in G

Accelerometer.Z Secondary accelerometer z-axis data, in G

Temperature Temperature of the IMU, in Celcius

Gyroscope.X Gyroscope x-axis data, in degrees per second

Gyroscope.Y Gyroscope y-axis data, in degrees per second

Gyroscope.Z Gyroscope z-axis data, in degrees per second

Magnetometer.X Magnetometer x-axis data, in micro-Teslas

Magnetometer.Y Magnetometer y-axis data, in micro-Teslas

Magnetometer.Z Magnetometer z-axis data, in micro-Teslas

file_source_IMU The filename of the IMU file

date_processed_IMU The date the IMU file was processed

ms The millisecond value of the timestamp

mean_Accel_VM Vector magnitude of the secondary accelerometer signal, in G

Gyroscope_VM_DegPerS Gyroscope vector magnitude, in degrees per second

Magnetometer_VM_MicroT Vector magnitude of the magnetometer signal, in micro-Teslas

`raw_for_cv` *Primary accelerometer data to calculate coefficient of variation per 10-s*

Description

A partially-processed primary accelerometer dataset ready to calculate the coefficient of variation per 10-s

Usage

```
raw_for_cv
```

Format

A data frame with 299 rows and 2 variables:

Block A vestigial variable synonymous with row number

ENMO Euclidian Norm Minus One, in milli-g

`raw_to_collapse` *Primary accelerometer data to collapse*

Description

A partially-processed primary accelerometer dataset ready to be collapsed from raw samples to one-second summaries.

Usage

```
raw_to_collapse
```

Format

A data frame with 24000 rows and 3 variables:

Accelerometer X Primary accelerometer x-axis data, in G

Accelerometer Y Primary accelerometer y-axis data, in G

Accelerometer Z Primary accelerometer z-axis data, in G

`read_AG_raw` *File reading function for primary accelerometer files*

Description

File reading function for primary accelerometer files

Usage

```
read_AG_raw(file, output_window_secs = 1, verbose = FALSE)
```

Arguments

`file` A character scalar giving path to primary accelerometer file

`output_window_secs`
the desired epoch length; defaults to one second

`verbose` A logical scalar: print progress updates?

Value

A dataframe giving processed raw data from the primary accelerometer in the specified epoch length

Examples

```
raw_file <-
    system.file("extdata",
    "TestID_LeftWrist_RAW.csv",
    package = "TwoRegression")

read_AG_raw(raw_file)
```

`read_IMU` *File reading function for IMU files*

Description

File reading function for IMU files

Usage

```
read_IMU(file, output_window_secs = 1, verbose = FALSE)
```

Arguments

`file` character scalar giving the path to the IMU file

`output_window_secs`
the desired epoch length; defaults to one second

`verbose` A logical scalar: print progress updates?

Value

A dataframe giving processed IMU data in the specified epoch length

Examples

```
## Not run:
imu_file <-
    system.file("extdata",
        "TestID_LeftWrist_IMU.csv",
        package = "TwoRegression")

read_IMU(imu_file)

## End(Not run)
```

TwoRegression *Process Data from Wearable Research Devices Using Two-Regression Algorithms*

Description

The TwoRegression package is designed to make implementation of two-regression algorithms quick, easy, and accurate.

Core functions

get_cvPER

hibbing18_twoReg_process

Associated References

Hibbing PR, LaMunion SR, Kaplan AS, & Crouter SE (2017). Estimating energy expenditure with ActiGraph GT9X Inertial Measurement Unit. *Medicine and Science in Sports and Exercise*. Advance online publication. doi: 10.1249/MSS.0000000000001532

Examples

```
## Not run:
raw_file <-
    system.file("extdata",
        "TestID_LeftWrist_RAW.csv",
        package = "TwoRegression")

imu_file <-
    system.file("extdata",
        "TestID_LeftWrist_IMU.csv",
        package = "TwoRegression")

wear <- "Left Wrist"
id <- "Test"
alg <- 1:2

hibbing18_twoReg_process(raw_file, imu_file, wear, id, alg)

## End(Not run)
```

Index