

Package ‘activPAL’

May 7, 2026

Type Package

Title Advanced Processing and Chart Generation from activPAL Events Files

Version 0.1.4

Maintainer Craig Speirs <craig@paltechnologies.com>

Description Contains functions to generate pre-defined summary statistics from activPAL events files <<https://www.palt.com/>>. The package also contains functions to produce informative graphics that visualise physical activity behaviour and trends. This includes generating graphs that align physical activity behaviour with additional time based observations described by other data sets, such as sleep diaries and continuous glucose monitoring data.

License GPL-3

Encoding UTF-8

Imports dplyr, ggplot2, tidyr, lubridate, magrittr, methods, devtools

RoxygenNote 6.1.1

Suggests testthat

NeedsCompilation no

Author Craig Speirs [aut, cre],
Kate Lyden [ctb],
Douglas Maxwell [ctb],
David Loudon [ctb]

Repository CRAN

Date/Publication 2025-09-17 14:20:02 UTC

Contents

activity.summary.by.custom.period	2
activity.summary.by.window.duration	2
activity.with.overlay.chart	3
activity.with.overlay.chart.folder	4

daily.stepping.activity.bubble.chart	5
stepping.cadence.bands.folder.four.stepping.groups	5
stepping.cadence.bands.folder.two.stepping.groups	6

Index	8
--------------	----------

activity.summary.by.custom.period

Summarise activity data from a folder of activPAL events files in user defined periods

Description

activity.summary.custom.period reads in a csv file containing a list of events files and csv files with time periods of interest (the first two columns should have the start date and end date in the format dd-mm-yy HH:MM). A summary table is saved to a csv file for each events file. A further csv file giving a summary for all the events file is also generated. The generated summary activity table for each events file is saved to a csv file, while an additional csv file containing the summaries across all the processed events files is also saved.

Usage

```
activity.summary.by.custom.period(location_file, output_folder)
```

Arguments

location_file The filepath to a csv file containing the path for a group of csv files to be processed

output_folder The filepath for the folder where the csv summary files are saved

activity.summary.by.window.duration

Summarise activity data from a folder of activPAL events files into equal sized periods

Description

activity.summary.window processes all the events files (format *Events.csv) in a folder and produces a summary table giving a breakdown of the number and duration of each event type across consecutive periods of a specified duration during the day. If the window_size is not a factor of 24, the final period will be shorter than the other reporting periods, generating a warning to advise of this difference in reporting period duration. The generated summary activity table for each events file is saved to a csv file, while an additional csv file containing the summaries across all the processed events files is also saved.

Usage

```
activity.summary.by.window.duration(input_folder, output_folder,
  window_duration = 2)
```

Arguments

`input_folder` The filepath for the folder where the events files (format *Events.csv) to be processed are saved

`output_folder` The filepath for the folder where the csv summary files are to be saved

`window_duration` The size of each bucket in hours that the data is to be summarised. Default - 2 hours.

Examples

```
input_folder <- paste(system.file("extdata", "", package = "activPAL"),"/",sep="")
output_folder <- paste(tempdir(),"/",sep="")

activPAL::activity.summary.by.window.duration(input_folder,output_folder,2)

# Omitting a value for window_duration sets the default window size to 2 hours
activPAL::activity.summary.by.window.duration(input_folder,output_folder)

# You can set the window_duration to a time period that is not evenly divisible
# by 24 hours. In this case the last reporting period for each day will be
# shorter than the specified window_duration.
activPAL::activity.summary.by.window.duration(input_folder,output_folder,3.5)
```

activity.with.overlay.chart

Combines events file data with observational data for a single events file

Description

Reads in an events file and overlay file, generating spiral and linear charts aligning the events file data with the corresponding overlay data. Where the overlay data is non-continuous data (such as sleeps diaries) the csv file should have the columns `start_time` (dd-mm-YYYY HH:MM), `end_time` (dd-mm-YYYY HH:MM) and `category` (text). Where the overlay data is continuous data (such as continuous glucose monitoring) the csv file should have the columns `start_time`(dd-mm-YYYY HH:MM) and `category` (text). In this case the end time coincides with the start time of the subsequent observation. The charts are saved as png images with two images generated (one spiral chart and one linear chart).

Note: Spiral plots are generated using `ggplot2` and `coord_polar()` and can take 45 - 60 seconds to generate each spiral plot.

Usage

```
activity.with.overlay.chart(events_file, overlay_file, output_folder)
```

Arguments

events_file The filepath of the events file. Must be a valid activPAL events csv file
 overlay_file The filepath of the csv file containing the overlay data csv file
 output_folder The filepath of the folder where the generated chart are to be saved to

Examples

```
events_file <- system.file("extdata", "Test_Events.csv", package = "activPAL")
sleep_file <- system.file("extdata", "Sleep.csv", package = "activPAL")
output_folder <- paste(tempdir(), "/", sep="")
```

```
activPAL::activity.with.overlay.chart(events_file, sleep_file, output_folder)
```

```
activity.with.overlay.chart.folder
```

Combines events file data with observational data across multiple events files to produce and save charts

Description

Reads in a csv file containing a list of events file locations and overlay file locations, generating and saving spiral and linear charts for each events file aligning the events file data with the corresponding overlay data. Where the overlay data is non-continuous data (such as sleeps diaries) the csv file should have the columns start_time (dd-mm-YYYY HH:MM), end_time (dd-mm-YYYY HH:MM) and category (text). Where the overlay data is continuous data (such as continuous glucose monitoring) the csv file should have the columns start_time (dd-mm-YYYY HH:MM) and category (text). In this case the end time coincides with the start time of the subsequent observation. The charts are saved as png images with two images generated for each events file / overlay file pair (one spiral chart and one linear chart).

Note: Spiral plots are generated using ggplot2 and coord_polar() and can take 45 - 60 seconds to generate each spiral plot.

Usage

```
activity.with.overlay.chart.folder(index_file_location, output_folder)
```

Arguments

index_file_location The filepath for the csv file with the following column names file_id, events_file, overlay_file
 output_folder The filepath of the folder where the generated chart are to be saved to

`daily.stepping.activity.bubble.chart`

Processes a folder of events files to output bubble charts showing the daily distribution of stepping cadences

Description

Processes a folder of events files to output bubble charts showing the daily distribution of stepping cadences. A bubble chart shows daily patterns in stepping behaviour. For each valid day of activity data a bubble is plotted on the chart. The centre of bubble indicates the weighted median cadence of all stepping bouts of duration between 10 seconds and one minute (x-axis position) and the weighted median cadence of all stepping bouts of duration greater than or equal to one minute (y-axis position). The height of a bubble indicates the weighted upper quartile (y - max) and lower quartile (y - min) cadence across the stepping bouts of duration greater than or equal to one minute occurring during the given day. The width of a bubble indicates the weighted upper quartile (x - right) and lower quartile (x - left) cadence across the stepping bouts of duration between 10 seconds and one minute occurring during the given day. Each bubble chart is saved as a .png image.

Usage

```
daily.stepping.activity.bubble.chart(input_folder, output_folder)
```

Arguments

`input_folder` The filepath for the folder where the events files to be processed are saved
`output_folder` The filepath for the folder where the bubble charts are to be saved

Examples

```
input_folder <- paste(system.file("extdata", "", package = "activPAL"),"/",sep="")  
output_folder <- paste(tempdir(),"/",sep="")
```

```
activPAL::daily.stepping.activity.bubble.chart(input_folder,output_folder)
```

`stepping.cadence.bands.folder.four.stepping.groups`

Processes events files to produce histograms showing the distribution of stepping and weighted median stepping cadence across four groups of stepping bout duration.

Description

Processes a folder of events files and generates a faceted set of histograms for each events file showing the duration of stepping in different cadence bands (each cadence band has a width of 10 steps per minute) for stepping bouts of duration 10 seconds to 1 minute, 1 minute to 5 minutes, 5 minutes to 10 minutes and 10 minutes plus. The weighted median cadence of stepping at each stepping bout duration is also calculated and indicated on the histogram. The values of the weighted median cadence for each stepping duration across all the processed events files is also returned as a data.frame.

Usage

```
stepping.cadence.bands.folder.four.stepping.groups(input_folder,
  output_folder, generate_charts = TRUE)
```

Arguments

`input_folder` The filepath for the folder where the events files to be processed are saved

`output_folder` The filepath for the folder where the generated files are to be saved

`generate_charts`
Set TRUE if stacked histograms showing the distribution of stepping cadences are to be generated for each events file

Examples

```
input_folder <- paste(system.file("extdata", "", package = "activPAL"),"/",sep="")
output_folder <- paste(tempdir(),"/",sep="")

activPAL::stepping.cadence.bands.folder.four.stepping.groups(input_folder,output_folder,TRUE)
# Omitting a value for generate_charts results in the charts being saved in the output folder.
activPAL::stepping.cadence.bands.folder.four.stepping.groups(input_folder,output_folder)
```

```
stepping.cadence.bands.folder.two.stepping.groups
```

Processes events files to produce histograms showing the distribution of stepping and weighted median stepping cadence across two groups of stepping bout duration.

Description

Processes a folder of events files and generates a faceted set of histograms for each events file showing the duration of stepping in different cadence bands (each cadence band has a width of 10 steps per minute) for stepping bouts of duration 10 seconds to 1 minute and 1 minute plus. The weighted median cadence of stepping at each stepping bout duration is also calculated and indicated on the histogram. The values of the weighted median cadence for each stepping duration across all the processed events files is also returned as a data.frame.

Usage

```
stepping.cadence.bands.folder.two.stepping.groups(input_folder,  
  output_folder, generate_charts = TRUE)
```

Arguments

`input_folder` The filepath for the folder where the events files to be processed are saved

`output_folder` The filepath for the folder where the generated files are to be saved

`generate_charts`
Set TRUE if stacked histograms showing the distribution of stepping cadences are to be generated for each events file. Default = TRUE

Examples

```
input_folder <- paste(system.file("extdata", "", package = "activPAL"),"/",sep="")  
output_folder <- paste(tempdir(),"/",sep="")  
  
activPAL::stepping.cadence.bands.folder.two.stepping.groups(input_folder,output_folder,TRUE)  
# Omitting a value for generate_charts results in the charts being saved in the output folder.  
activPAL::stepping.cadence.bands.folder.two.stepping.groups(input_folder,output_folder)
```

Index

activity.summary.by.custom.period, [2](#)
activity.summary.by.window.duration, [2](#)
activity.with.overlay.chart, [3](#)
activity.with.overlay.chart.folder, [4](#)

daily.stepping.activity.bubble.chart,
[5](#)

stepping.cadence.bands.folder.four.stepping.groups,
[5](#)

stepping.cadence.bands.folder.two.stepping.groups,
[6](#)