Package ‘AmbientViewer’

May 6, 2025

Title Filtering and Visualisation for Somnofy Data

Version 0.0.1

Description
This package helps importing, filtering and visualising data from Somnofy devices (VitalThings).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests devtools,
mockery,
testthat,
tibble

Config/testthat/edition 3

Imports cli,
dplyr,
ggplot2,
later,
logging,
lubridate,
readr,
rlang,
scales,
shiny,
shinyjs,
shinyWidgets,
stringr,
svglite,
tidyr

Depends R (>= 4.3)

LazyData true

Contents

`ambient_viewer` *Ambient Viewer app*

Description

This function launches the Ambient Viewer app, a Shiny application for visualizing and analyzing sleep data.

Usage

```
ambient_viewer()
```

convert_angle_to_time *Convert an angle to time*

Description

This function converts an angle in radians to time in the provided unit (can be "second", "minute" or "hour").

Usage

```
convert_angle_to_time(angle, unit = "second")
```

Arguments

`angle` A numeric value representing the angle in radians.

`unit` A string indicating the unit of time. Can be "second", "minute", or "hour".

Value

A numeric value representing the time in the specified unit.

See Also

`convert_times_to_mean_angle()` to calculate the average angle from a vector of time values.

Other time processing: `convert_times_to_mean_angle()`, `group_epochs_by_night()`, `group_sessions_by_night`
`max_time()`, `mean_time()`, `min_time()`, `shift_times_by_12h()`

Examples

```
convert_angle_to_time(pi/2, unit = "hour")
```

convert_times_to_mean_angle *Convert a vector of times to a mean angle*

Description

This function converts a vector of times to a mean angle in radians. It is useful to calculate average times spanning midnight

Usage

```
convert_times_to_mean_angle(times, unit = "second")
```

Arguments

`times` A vector of times in seconds.

`unit` A string indicating the unit of time. Can be "second", "minute", or "hour".

Value

A numeric value representing the mean angle in radians.

See Also

`convert_angle_to_time()` to convert the mean angle back to time format.

Other time processing: `convert_angle_to_time()`, `group_epochs_by_night()`, `group_sessions_by_night()`, `max_time()`, `mean_time()`, `min_time()`, `shift_times_by_12h()`

Examples

```
convert_times_to_mean_angle(c(23, 10, 0), unit = "hour")
```

`example_epochs` *Example Epoch data*

Description

A data frame containing epoch data recorded by a Somnofy device.

Usage

```
example_epochs
```

Format

`example_epochs`**:**
A data frame with 18,755 rows and 15 columns. Each row represents a time-point (or epoch) in a session. Epochs are 30 seconds long. The columns are as follows:

- timestamp: The time at which the epoch was recorded in UTC.
- subject_id: The ID of the subject.
- signal_quality_mean: The mean signal quality of the epoch.
- movement_fast_mean: The mean movement detected during the epoch.
- movement_fast_nonzero_pct
- distance_mean: the distance of the subject from the device in meters.
- motion_data_count: The number of data points in the epoch (30).
- light_ambient_mean: The ambient light level during the epoch.
- sound_amplitude_mean: The sound amplitude during the epoch.
- temperature_ambient_mean: The ambient temperature during the epoch.
- humidity_mean: The ambient humidity during the epoch.
- pressure_mean: The ambient pressure during the epoch.
- indoor_air_quality_mean: The indoor air quality during the epoch.
- epoch_duration: The precise duration of the epoch (seconds).
- sleep_stage: The sleep stage as established with the VT algorithm. They are encoded as numbers 0-5

Source

data-raw/example_epochs.csv

`example_sessions` *Example Sessions data*

Description

A data frame containing sessions recorded by a Somnofy device.

Usage

```
example_sessions
```

Format

`example_sessions`**:**

A data frame with 124 rows and 60 columns. Each row represents a session. Columns contain metadata about the session, including:

- start_time: The start time of the session in UTC.
- end_time: The end time of the session in UTC.
- subject_id: The ID of the subject.
- device_serial_number: The serial number of the device used.
- time_at_sleep_onset: The time at which the subject fell asleep.
- time_at_wakeup: The time at which the subject woke up. Columns also include various metrics averaged over the session, such as:
- mean heart rate
- mean respiration rate Finally, some columns contain environmental parameters, such as:
- Temperature
- Humidity
- Light intensity
- Noise level
- Atmospheric pressure

Source

data-raw/example_sessions.csv

`filter_by_night_range` *Filter sessions for nights within a night range*

Description

Filter sessions for nights within a night range

Usage

```
filter_by_night_range(sessions, from_night, to_night)
```

Arguments

`sessions`	The sessions dataframe
`from_night`	The start night of the range (inclusive) in YYYY-MM-DD format
`to_night`	The end night of the range (inclusive) in YYYY-MM-DD format

Value

The sessions dataframe with only the sessions that fall within the specified night range

See Also

Other filtering: `filter_epochs_from_sessions()`, `remove_sessions_no_sleep()`, `select_devices()`, `select_subjects()`, `set_min_time_in_bed()`, `set_session_sleep_onset_range()`, `set_session_start_time_r`

Examples

```
filtered_sessions <- filter_by_night_range(example_sessions, "2025-04-07", "2025-04-10")
```

---

filter_epochs_from_sessions

Filter epochs based on session IDs

Description

Filter epochs based on session IDs

Usage

```
filter_epochs_from_sessions(epochs, sessions)
```

Arguments

`epochs`	The epochs dataframe
`sessions`	The sessions dataframe

Value

The epochs dataframe with only the epochs that belong to the specified sessions

See Also

`filter_by_night_range()` to filter sessions by night range.

Other filtering: `filter_by_night_range()`, `remove_sessions_no_sleep()`, `select_devices()`, `select_subjects()`, `set_min_time_in_bed()`, `set_session_sleep_onset_range()`, `set_session_start_time_r`

Examples

```
# Apply filtering to sessions to keep specific nights, and filter epochs accordingly
filtered_sessions <- filter_by_night_range(example_sessions, "2025-04-07", "2025-04-10")#
filtered_epochs <- filter_epochs_from_sessions(example_epochs, filtered_sessions)
```

`get_epochs_summary` *Summarise epoch information*

Description

This function displays the number of sessions in the epoch data, as well as the start and end dates of the epoch data

Usage

```
get_epochs_summary(epochs)
```

Arguments

`epochs`	The epochs dataframe

Value

A single-row dataframe summarising epoch information

See Also

`get_sessions_summary()` to summarise session information.

Other data tables: `get_non_complying_sessions()`, `get_removed_sessions()`, `get_sessions_summary()`

Examples

```
get_epochs_summary(example_epochs)
```

`get_non_complying_sessions` *Get non-complying sessions (i.e. where there is more than one session on the same day)*

Description

Get non-complying sessions (i.e. where there is more than one session on the same day)

Usage

```
get_non_complying_sessions(sessions)
```

Arguments

`sessions`	The sessions dataframe

Value

The sessions dataframe with only the sessions that are non-complying

See Also

Other data tables: get_epochs_summary(), get_removed_sessions(), get_sessions_summary()

Examples

```
duplicate_sessions <- get_non_complying_sessions(example_sessions)
```

---

get_removed_sessions *Get a table of sessions that were removed during filtering*

Description

Get a table of sessions that were removed during filtering

Usage

```
get_removed_sessions(sessions, filtered_sessions)
```

Arguments

sessions	The original sessions dataframe
filtered_sessions	The filtered sessions dataframe

Value

The sessions dataframe with only the sessions that were removed during filtering

See Also

Other data tables: get_epochs_summary(), get_non_complying_sessions(), get_sessions_summary()

Examples

```
filtered_sessions <- set_session_start_time_range(example_sessions, "22:00", "06:00")
removed_sessions <- get_removed_sessions(example_sessions, filtered_sessions)
```

---

get_sessions_summary *Make a summary of session information*

Description

This function summarises session information, including the number of sessions, mean session length, mean time at sleep onset and wakeup, subject and device ID.

Usage

```
get_sessions_summary(sessions)
```

Arguments

`sessions`	The sessions dataframe.

Value

A single-row dataframe summarizing session information.

See Also

`get_epochs_summary()` to summarise epoch information.

Other data tables: `get_epochs_summary()`, `get_non_complying_sessions()`, `get_removed_sessions()`

Examples

```
get_sessions_summary(example_sessions)
```

`group_epochs_by_night` *Create a grouping by night for epoch data*

Description

Create a grouping by night for epoch data

Usage

```
group_epochs_by_night(epochs)
```

Arguments

`epochs`	The epochs dataframe

Details

The function creates a new column `night` that groups the epochs by night. Timepoints before 12 PM are considered part of the previous night.

Value

The epochs dataframe with the `night` column added

See Also

`group_sessions_by_night()` to group session data by night.

Other time processing: `convert_angle_to_time()`, `convert_times_to_mean_angle()`, `group_sessions_by_night`
`max_time()`, `mean_time()`, `min_time()`, `shift_times_by_12h()`

Examples

```
epochs <- group_epochs_by_night(example_epochs)
```

group_sessions_by_night

Create a grouping by night for session data

Description

Create a grouping by night for session data

Usage

```
group_sessions_by_night(sessions)
```

Arguments

sessions	The sessions dataframe

Details

The function creates a new column `night` that groups the sessions by night depending on their start time. Sessions that start before 12 PM are considered part of the previous night.

Value

The sessions dataframe with the `night` column added

See Also

`group_epochs_by_night()` to group epoch data by night.

Other time processing: `convert_angle_to_time()`, `convert_times_to_mean_angle()`, `group_epochs_by_night()`, `max_time()`, `mean_time()`, `min_time()`, `shift_times_by_12h()`

Examples

```
sessions <- group_sessions_by_night(example_sessions)
```

load_epochs

Load epoch data

Description

Load epoch data

Usage

```
load_epochs(epochs_file)
```

Arguments

epochs_file	The path to the epochs file

Details

The function loads the epoch data from a CSV file and groups the epochs by night.

Value

A dataframe containing the epoch data

See Also

Other data loading: `load_sessions()`

`load_sessions` *Load session data*

Description

Load session data

Usage

```
load_sessions(sessions_file)
```

Arguments

`sessions_file`	The path to the sessions file

Details

The function loads the session data from a CSV file and groups the sessions by night.

Value

A dataframe containing the session data

See Also

Other data loading: `load_epochs()`

max_time — *Calculate the maximum time from 12pm to 12pm*

Description

This function calculates the maximum time from a vector of time strings in the format "YYYY-MM-DD HH:MM:SS". It considers a time window from 12pm to 12pm the next day, so 11:00 is considered later than 13:00.

Usage

```
max_time(time_vector)
```

Arguments

`time_vector`	A vector of time strings in the format "YYYY-MM-DD HH:MM:SS".

Value

A string representing the maximum time in the format "HH:MM".

See Also

`min_time()` to calculate the minimum time in the same format.

Other time processing: `convert_angle_to_time()`, `convert_times_to_mean_angle()`, `group_epochs_by_night()`, `group_sessions_by_night()`, `mean_time()`, `min_time()`, `shift_times_by_12h()`

Examples

```
max_time(c("2025-04-08 23:00:00", "2025-04-09 01:00:00", "2025-04-09 02:30:00"))
```

mean_time — *Calculate the mean time from a vector of time strings*

Description

This function calculates the mean time from a vector of time strings in the format "YYYY-MM-DD HH:MM:SS".

Usage

```
mean_time(time_vector)
```

Arguments

`time_vector`	A vector of time strings in format "YYYY-MM-DD HH:MM:SS", "HH:MM:SS" or "HH:MM".

Value

A string representing the mean time in the format "HH:MM".

See Also

Other time processing: convert_angle_to_time(), convert_times_to_mean_angle(), group_epochs_by_night(), group_sessions_by_night(), max_time(), min_time(), shift_times_by_12h()

Examples

```
# Use on a vector of time strings representing full dates
time_vector <- c("2025-04-08 23:00:00", "2025-04-09 01:00:00")
mean_time(time_vector)

# Use on time-only strings
time_vector <- c("22:56", "01:32")
mean_time(time_vector)

# Use on a dataframe column
mean_time(example_sessions$time_at_sleep_onset)
```

min_time *Calculate the minimum time from 12pm to 12pm*

Description

This function calculates the minimum time from a vector of time strings in the format "YYYY-MM-DD HH:MM:SS". It considers a time window from 12pm to 12pm the next day, so 11:00 is considered later than 13:00.

Usage

```
min_time(time_vector)
```

Arguments

time_vector	A vector of time strings in the format "YYYY-MM-DD HH:MM:SS".

Value

A string representing the minimum time in the format "HH:MM".

See Also

max_time() to calculate the maximum time in the same format.

Other time processing: convert_angle_to_time(), convert_times_to_mean_angle(), group_epochs_by_night(), group_sessions_by_night(), max_time(), mean_time(), shift_times_by_12h()

Examples

```
min_time(c("2025-04-08 23:00:00", "2025-04-09 01:00:00", "2025-04-09 02:30:00"))
```

`plot_actigram` *Plot an Actigram*

Description

Generate an actigram from the Somnofy epoch data.

Usage

```
plot_actigram(epochs)
```

Arguments

`epochs`	The epochs data frame

Value

A ggplot object representing the actigram

`plot_bedtimes_waketimes` *Plot bedtimes and waketimes*

Description

Plot bedtimes and waketimes

Usage

```
plot_bedtimes_waketimes(sessions, groupby = "night")
```

Arguments

`sessions`	The sessions dataframe
`groupby`	The grouping variable for the plot. Can be "night", "workday", or "weekday".

Value

A ggplot graph showing the bedtimes and waketimes

`plot_hypnogram` *Plot Hypnogram*

Description

Plot Hypnogram

Usage

```
plot_hypnogram(epochs)
```

Arguments

`epochs`	The epochs dataframe

Value

A ggplot object showing the hypnogram as bars

See Also

`plot_sleep_stages()` to show the proportion of each sleep stage per day

Other plot epochs: `plot_sleep_spiral()`, `plot_sleep_stages()`, `plot_timeseries()`

`plot_sleep_bubbles` *Plot Sleep Bubbles*

Description

This function creates a bubble plot of sleep sessions, where the size and colour of the bubbles represents the sleep duration.

Usage

```
plot_sleep_bubbles(sessions)
```

Arguments

`sessions`	The sessions dataframe.

Value

A ggplot object containing the sleep bubbles graph.

See Also

Other plot sessions: `plot_sleep_clock()`, `plot_timeseries_sessions()`

`plot_sleep_clock` *Plot Sleep Clock*

Description

Plot Sleep Clock

Usage

```
plot_sleep_clock(sessions)
```

Arguments

`sessions`	The sessions dataframe

Value

A ggplot object showing the sleep clock

See Also

Other plot sessions: `plot_sleep_bubbles()`, `plot_timeseries_sessions()`

`plot_sleep_spiral` *Plot Sleep Spiral*

Description

Plot Sleep Spiral

Usage

```
plot_sleep_spiral(epochs)
```

Arguments

`epochs`	The epochs dataframe

Value

A ggplot object showing the sleep spiral

See Also

Other plot epochs: `plot_hypnogram()`, `plot_sleep_stages()`, `plot_timeseries()`

`plot_sleep_stages` *Plot Sleep Stages*

Description

Plot Sleep Stages

Usage

```
plot_sleep_stages(epochs)
```

Arguments

`epochs`	The epochs dataframe

Value

A ggplot object showing the proportion of sleep stages for each night

See Also

`plot_hypnogram()` to show the detailed sleep stages over time

Other plot epochs: `plot_hypnogram()`, `plot_sleep_spiral()`, `plot_timeseries()`

`plot_timeseries` *Plot epoch time series data for a given variable*

Description

Plot epoch time series data for a given variable

Usage

```
plot_timeseries(epochs, variable, exclude_zero = FALSE)
```

Arguments

`epochs`	The epochs dataframe
`variable`	The variable to plot (e.g., "temperature_ambient_mean")
`exclude_zero`	Logical, whether to exclude zero values from the plot (default: FALSE)

Value

A ggplot object

See Also

`plot_timeseries_sessions()` to plot session data.

Other plot epochs: `plot_hypnogram()`, `plot_sleep_spiral()`, `plot_sleep_stages()`

plot_timeseries_sessions

Plot session time series data for a given variable

Description

Plot session time series data for a given variable

Usage

```
plot_timeseries_sessions(sessions, variable, exclude_zero = FALSE)
```

Arguments

`sessions`	The sessions dataframe
`variable`	The variable to plot (e.g., "time_at_sleep_onset")
`exclude_zero`	Logical, whether to exclude zero values from the plot (default: FALSE)

Value

A ggplot object

See Also

`plot_timeseries()` to plot epoch data.

Other plot sessions: `plot_sleep_bubbles()`, `plot_sleep_clock()`

remove_sessions_no_sleep

Remove sessions with no sleep

Description

Remove sessions with no sleep

Usage

```
remove_sessions_no_sleep(sessions)
```

Arguments

`sessions`	The sessions dataframe

Value

The sessions dataframe with only the sessions that have a sleep period greater than 0

See Also

Other filtering: filter_by_night_range(), filter_epochs_from_sessions(), select_devices(), select_subjects(), set_min_time_in_bed(), set_session_sleep_onset_range(), set_session_start_time_r

Examples

```
filtered_sessions <- remove_sessions_no_sleep(example_sessions)
```

---

select_devices *Select devices by ID*

Description

Select devices by ID

Usage

```
select_devices(sessions, device_ids)
```

Arguments

sessions	The sessions dataframe
device_ids	The device IDs to select

Value

The sessions dataframe with only the sessions recorded by the specified devices

See Also

select_subjects() to select sessions by subject ID.

Other filtering: filter_by_night_range(), filter_epochs_from_sessions(), remove_sessions_no_sleep(), select_subjects(), set_min_time_in_bed(), set_session_sleep_onset_range(), set_session_start_time_r

---

select_subjects *Select subjects by ID*

Description

Select subjects by ID

Usage

```
select_subjects(sessions, subject_ids)
```

Arguments

sessions	The sessions dataframe
subject_ids	The subject IDs to select

Value

The sessions dataframe with only the sessions that belong to the specified subjects

See Also

`select_devices()` to select sessions by device ID.

Other filtering: `filter_by_night_range()`, `filter_epochs_from_sessions()`, `remove_sessions_no_sleep()`, `select_devices()`, `set_min_time_in_bed()`, `set_session_sleep_onset_range()`, `set_session_start_time_ra`

`set_min_time_in_bed` *Set minimum time in bed*

Description

Set minimum time in bed

Usage

```
set_min_time_in_bed(sessions, min_time_in_bed)
```

Arguments

`sessions`	The sessions dataframe
`min_time_in_bed`	The minimum time in bed in hours

Value

The sessions dataframe with only the sessions that meet the minimum time in bed requirement

See Also

Other filtering: `filter_by_night_range()`, `filter_epochs_from_sessions()`, `remove_sessions_no_sleep()`, `select_devices()`, `select_subjects()`, `set_session_sleep_onset_range()`, `set_session_start_time_range(`

Examples

```
filtered_sessions <- set_min_time_in_bed(example_sessions, 2)
```

set_session_sleep_onset_range

Set sleep onset time range

Description

Set sleep onset time range

Usage

```
set_session_sleep_onset_range(sessions, from_time, to_time)
```

Arguments

`sessions`	The sessions dataframe
`from_time`	Include sessions where sleep started after this time (in format HH:MM)
`to_time`	Include sessions where sleep started before this time (in format HH:MM)

Value

The sessions dataframe with only the sessions where sleep started within the specified time range

See Also

`set_session_start_time_range()` to filter sessions based on start time.

Other filtering: `filter_by_night_range()`, `filter_epochs_from_sessions()`, `remove_sessions_no_sleep()`, `select_devices()`, `select_subjects()`, `set_min_time_in_bed()`, `set_session_start_time_range()`

Examples

```
filtered_sessions <- set_session_sleep_onset_range(example_sessions, "22:00", "06:00")
```

set_session_start_time_range

Set session start time range

Description

Set session start time range

Usage

```
set_session_start_time_range(sessions, from_time, to_time)
```

Arguments

`sessions`	The sessions dataframe
`from_time`	Include sessions that started after this time (in format HH:MM)
`to_time`	Include sessions that started before this time (in format HH:MM)

Value

The sessions dataframe with only the sessions that started within the specified time range

See Also

set_session_sleep_onset_range() to filter sessions based on sleep onset time.

Other filtering: filter_by_night_range(), filter_epochs_from_sessions(), remove_sessions_no_sleep(), select_devices(), select_subjects(), set_min_time_in_bed(), set_session_sleep_onset_range()

Examples

```
filtered_sessions <- set_session_start_time_range(example_sessions, "22:00", "06:00")
```

---

shift_times_by_12h *Shift times to break at 12 pm*

Description

This function shifts times so that the day starts at 12 PM. This is useful for plotting night data

Usage

```
shift_times_by_12h(times)
```

Arguments

times	A vector of times in POSIXct format, character convertible to POSIXct, or numerical (in hours).

Value

A vector of times in POSIXct format (or numerical if numerical provided as input) shifted to start at 12 PM

See Also

Other time processing: convert_angle_to_time(), convert_times_to_mean_angle(), group_epochs_by_night(), group_sessions_by_night(), max_time(), mean_time(), min_time()

Examples

```
# Shift a vector of times in HH:MM format
shift_times_by_12h(c("02:30", "16:00"))
#> "14:30" "04:00"

# Shift times in YYYY-MM-DD HH:MM:SS format
shift_times_by_12h(c("2025-04-08 23:00:00", "2025-04-09 01:00:00"))
#> "2025-04-08 11:00" "2025-04-09 13:00"

# Shift sessions start times to start at 12 PM
shifted_times <- shift_times_by_12h(example_sessions$session_start)
```

```
# Use dplyr::mutate to dicrectly add the shifted times to a dataframe
epochs <- example_epochs |>
  dplyr::mutate(shifted_time = shift_times_by_12h(timestamp))
```

Index