Package: eegUtils (via r-universe)

August 25, 2024

Type Package

Title Utilities for Electroencephalographic (EEG) Analysis

Version 0.7.0

Date 2022-02-03

Maintainer Matt Craddock <matt@mattcraddock.com>

Description EEG processing and visualization tools. Includes import functions for 'BioSemi' (.BDF), 'Neuroscan' (.CNT), 'Brain Vision Analyzer' (.VHDR), 'EEGLAB' (.set) and 'Fieldtrip' (.mat). Many preprocessing functions such as referencing, epoching, filtering, and ICA are available. There are a variety of visualizations possible, including timecourse and topographical plotting.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports ggplot2, dplyr (>= 1.0.0), scales, purrr, shiny, tidyr (>= 1.0.0), miniUI, rlang (>= 0.4.0), MASS, Matrix, signal, tibble, stats, matrixStats, pracma, abind, data.table, plotly, future.apply, Rcpp, isoband

Depends R (>= 3.3.0)

RoxygenNote 7.1.2

Suggests testthat, vdiffr, covr, knitr, rmarkdown, JADE, ica, geigen, whitening, fICA, edfReader, ini, R.matlab, hdf5r, mgcv, infomax (>= 0.1.0)

URL https://github.com/craddm/eegUtils, https://craddm.github.io/eegUtils

BugReports https://github.com/craddm/eegUtils/issues

VignetteBuilder knitr

LinkingTo Rcpp, RcppArmadillo

Roxygen list(markdown = TRUE)

Remotes eegverse/infomax

Repository https://craddm.r-universe.dev

RemoteUrl https://github.com/craddm/eegUtils

RemoteRef HEAD

RemoteSha aeb0ec00c243702a8c95c328fa2852f8d0e69730

Contents

`apply_ica` *Recreate channel timecourses from ICA decompositions.*

Description

This function can be used to either recreate "mixed" (i.e. channel level) timecourses from an ICA decomposition, or to apply a set of ICA weights to a given dataset for the purpose of removing specific ICA components from that dataset.

Usage

```
apply_ica(data, ...)

## S3 method for class 'eeg_ICA'
apply_ica(data, comps = NULL, ...)

## S3 method for class 'eeg_epochs'
apply_ica(data, decomp, comps, ...)
```

Arguments

`data`	An `eeg_ICA` or `eeg_epochs` object.
`...`	Other parameters.
`comps`	Components to remove.
`decomp`	An `eeg_ICA` object.

Methods (by class)

- `eeg_ICA`: From given `eeg_ICA` object, recreate channel timecourses.
- `eeg_epochs`: Combine a specific set of ICA weights with any `eeg_epochs` object.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
test_ica <- run_ICA(demo_epochs, pca = 10)
plot_butterfly(demo_epochs)
# Reconstruct the original data from the ICA decomposition.
# Note that the ICA process subtracts the mean from each epoch,
# so the reconstructed plot may look slightly different to the original.
plot_butterfly(apply_ica(test_ica))
# Remove component 2 from the data
plot_butterfly(apply_ica(demo_epochs, test_ica, comps = 2))
```

---

ar_acf *Detect low autocorrelation of ICA components*

Description

Low autocorrelation can be a sign of a poor quality channel or component. Often these are noisy, poor contact, or heavily contaminated with muscle noise. Low autocorrelation at a lag of 20ms is often associated with muscle noise.

Usage

```
ar_acf(data, ...)

## S3 method for class 'eeg_ICA'
ar_acf(data, ms = 20, plot = TRUE, verbose = TRUE, threshold = NULL, ...)
```

Arguments

data	eeg_ICA object
...	additional parameters
ms	Time lag to check ACF, in milliseconds. Defaults to 20 ms.
plot	Produce plot showing ACF and threshold for all EEG components.
verbose	Print informative messages. Defaults to TRUE.
threshold	Specify a threshold for low ACF. NULL estimates the threshold automatically.

Value

A character vector of component names that break the threshold.

Methods (by class)

- eeg_ICA: Autocorrelation checker for eeg_ICA objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

Chaumon, M., Bishop, D.V., Busch, N.A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. J Neurosci Methods. Jul 30;250:47-63. doi: 10.1016/j.jneumeth.2015.02.025

Examples

```
demo_sobi <- run_ICA(demo_epochs, pca = 10)
ar_acf(demo_sobi)
```

`ar_chanfoc` *Detect high channel focality of ICA components*

Description

Detect components that load heavily on a single channel. Looks for components that have one particular channel that has a particularly high z-score.

Usage

```
ar_chanfoc(
  data,
  plot = TRUE,
  threshold = NULL,
  verbose = TRUE,
  measure = "max",
  ...
)
```

Arguments

`data`	`eeg_ICA` object
`plot`	Produce plot showing max z-scores and threshold for all ICA components.
`threshold`	Specify a threshold for high focality. NULL estimates the threshold automatically.
`verbose`	Print informative messages.
`measure`	Use maximum "max" or "kurtosis".
`...`	additional parameters

Value

A character vector of component names that break the threshold.

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

Chaumon, M., Bishop, D.V., Busch, N.A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. J Neurosci Methods. Jul 30;250:47-63. doi: 10.1016/j.jneumeth.2015.02.025

Examples

```
demo_sobi <- run_ICA(demo_epochs, pca = 10)
ar_chanfoc(demo_sobi)
```

`ar_eogcor` *Detect high component correlation with eye channels*

Description

Checks the correlation between individual components of an `eeg_ICA` decomposition and the electrooculogram channels of an `eeg_epochs` dataset.

Usage

```
ar_eogcor(decomp, data, ...)

## S3 method for class 'eeg_ICA'
ar_eogcor(
  decomp,
  data,
  HEOG,
  VEOG,
  threshold = NULL,
  plot = TRUE,
  bipolarize = TRUE,
  ...
)
```

Arguments

`decomp`	ICA decomposition
`data`	Original data
`...`	Other parameters
`HEOG`	Horizontal eye channels
`VEOG`	Vertical eye channels
`threshold`	Threshold for correlation (r). Defaults to NULL, automatically determining a threshold.
`plot`	Plot correlation coefficient for all components
`bipolarize`	Bipolarize the HEOG and VEOG channels?

Value

A character vector of component names that break the threshold.

Methods (by class)

- `eeg_ICA`: Method for eeg_ICA objects.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

References

Chaumon, M., Bishop, D.V., Busch, N.A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. J Neurosci Methods. Jul 30;250:47-63. doi: 10.1016/j.jneumeth.2015.02.025

`ar_eogreg` *Remove EOG using regression*

Description

Calculates and removes the contribution of eye movements to the EEG signal using least-squares regression. Specifically, it generate regression weights based on EOG channels that are used to estimate how much activity eye movements are responsible for across all channels.

Usage

```
ar_eogreg(data, heog, veog, bipolarize = TRUE)

## S3 method for class 'eeg_data'
ar_eogreg(data, heog, veog, bipolarize = TRUE)

## S3 method for class 'eeg_epochs'
ar_eogreg(data, heog, veog, bipolarize = TRUE)
```

Arguments

`data`	Data to regress - `eeg_data` or `eeg_epochs`
`heog`	Horizontal EOG channel labels
`veog`	Vertical EOG channel labels
`bipolarize`	Bipolarize the EOG channels. Only works when four channels are supplied (2 HEOG and 2 VEOG).

Value

An `eeg_data` or `eeg_epochs` object with corrections applied.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

---

ar_FASTER *FASTER EEG artefact rejection*

Description

An implementation of the FASTER artefact rejection method for EEG by Nolan, Whelan & Reilly (2010) FASTER: Fully Automated Statistical Thresholding for EEG artifact Rejection. J Neurosci Methods.

Usage

```
ar_FASTER(data, ...)

## S3 method for class 'eeg_epochs'
ar_FASTER(
  data,
  exclude = NULL,
  test_chans = TRUE,
  test_epochs = TRUE,
  test_cine = TRUE,
  ...
)

## S3 method for class 'eeg_group'
ar_FASTER(
  data,
  exclude = NULL,
  test_chans = TRUE,
  test_epochs = TRUE,
  test_cine = TRUE,
  EOG = NULL,
  ...
)

eeg_FASTER(data, ...)
```

Arguments

data	An object of class eeg_epochs
...	Parameters passed to FASTER
exclude	Channels to be ignored by FASTER.
test_chans	Logical. Run tests of global channel statistics

test_epochs	Logical. Run tests of globally bad epochs.
test_cine	Logical. Run tests for locally bad channels within epochs.
EOG	names of EOG channels to be used when computed maximum EOG values.

Value

An eeg_epochs object with artefact correction applied.

Methods (by class)

- eeg_epochs: Run FASTER on eeg_epochs
- eeg_group: Run FASTER on eeg_group objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

Nolan, Whelan & Reilly (2010). FASTER: Fully Automated Statistical Thresholding for EEG artifact Rejection. J Neurosci Methods.

Examples

```
ar_FASTER(demo_epochs)
```

---

ar_thresh *Simple absolute value thresholding*

Description

Reject data based on a simple absolute amplitude threshold. This marks any timepoint from any electrode.

Usage

```
ar_thresh(data, threshold, reject = FALSE)

## S3 method for class 'eeg_data'
ar_thresh(data, threshold, reject = FALSE)

## S3 method for class 'eeg_epochs'
ar_thresh(data, threshold, reject = FALSE)
```

Arguments

data	An object of class eeg_data or eeg_epochs.
threshold	In microvolts. If one value is supplied, it will be treated as a +- value.
reject	If TRUE, remove marked data immediately, otherwise mark for inspection/rejection. Defaults to FALSE.

Value

An object of class eeg_data or eeg_epochs

Methods (by class)

- eeg_data: Reject data using a simple threshold.
- eeg_epochs: Reject data using a simple threshold.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
ar_thresh(demo_epochs, c(100))
```

---

ar_trialfoc *Detect high trial focality of ICA components*

Description

Detect components that load heavily on a small number of trials. Looks for components that have one particular trial that has a particularly high z-score.

Usage

```
ar_trialfoc(data, plot = TRUE, threshold = NULL, verbose = TRUE)
```

Arguments

data	eeg_ICA object
plot	Produce plot showing max z-scores and threshold for all ICA components.
threshold	Specify a threshold (z-score) for high focality. NULL estimates the threshold automatically.
verbose	Print informative messages.

Value

A character vector of component names that break the threshold.

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

Chaumon, M., Bishop, D.V., Busch, N.A. (2015). A practical guide to the selection of independent components of the electroencephalogram for artifact correction. J Neurosci Methods. Jul 30;250:47-63. doi: 10.1016/j.jneumeth.2015.02.025

Examples

```
demo_sobi <- run_ICA(demo_epochs, pca = 10)
ar_trialfoc(demo_sobi)
```

---

as.data.frame.eeg_data

Convert eeg_data *to* data.frame

Description

Convert an object of class eeg_data into a standard data.frame.

Usage

```
## S3 method for class 'eeg_data'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  events = FALSE,
  coords = FALSE,
  ...
)
```

Arguments

x	Object of class eeg_data
row.names	Kept for compatibility with S3 generic, ignored.
optional	Kept for compatibility with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE
events	Include events in output.
coords	Include electrode coordinates in output. Only possible when long = TRUE.
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

`as.data.frame.eeg_epochs` *Convert* `eeg_epochs` *object to data.frame*

Description

Convert an `eeg_epochs` object to a data.frame for use with whatever packages you desire.

Usage

```
## S3 method for class 'eeg_epochs'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  events = FALSE,
  cond_labels,
  coords = TRUE,
  ...
)
```

Arguments

`x`	Object of class `eeg_epochs`
`row.names`	Kept for compatability with S3 generic, ignored.
`optional`	Kept for compatability with S3 generic, ignored.
`long`	Convert to long format. Defaults to FALSE.
`events`	Include events in output. Defaults to FALSE. Currently ignored.
`cond_labels`	Add column tagging epochs with events that have matching labels. Deprecated. Metainfo from the epochs structure is now added automatically.
`coords`	Include electrode coordinates in output. Ignored if long == FALSE.
`...`	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

as.data.frame.eeg_evoked *Convert* eeg_evoked *object to data frame*

Description

Convert eeg_evoked object to data frame

Usage

```
## S3 method for class 'eeg_evoked'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  coords = TRUE,
  ...
)
```

Arguments

x	Object of class eeg_evoked
row.names	Kept for compatability with S3 generic, ignored.
optional	Kept for compatability with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE
coords	include electrode coordinates in output. Ignored if long = FALSE.
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

as.data.frame.eeg_ICA *Convert* eeg_ICA *object to data frame*

Description

Convert eeg_ICA object to data frame

Usage

```
## S3 method for class 'eeg_ICA'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  cond_labels,
  mixing = FALSE,
  coords = TRUE,
  ...
)
```

Arguments

x	Object of class eeg_ICA
row.names	Kept for compatability with S3 generic, ignored.
optional	Kept for compatability with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE
cond_labels	add condition labels to data frame. Deprecated.
mixing	If TRUE, outputs the mixing matrix. If FALSE, outputs source activations.
coords	Adds electrode coordinates if TRUE; only if long data and the mixing matrix are requested.
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

---

as.data.frame.eeg_lm *Convert* eeg_lm *to data.frame*

Description

Convert an object of class eeg_data into a standard data.frame.

Usage

```
## S3 method for class 'eeg_lm'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  coords = TRUE,
```

```
  values = c("coefficients", "std_err", "t_stats", "r_sq"),
  ...
)
```

Arguments

x	Object of class eeg_data
row.names	Kept for compatibility with S3 generic, ignored.
optional	Kept for compatibility with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE.
coords	Include electrode coordinates in output. Only possible when long = TRUE.
values	Defaults to "coefficients", returning fitted coefficient values.
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

---

as.data.frame.eeg_stats

Convert eeg_stats *objects to data frames*

Description

Convert eeg_stats objects to data frames

Usage

```
## S3 method for class 'eeg_stats'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  long = FALSE,
  coords = FALSE,
  ...
)
```

Arguments

x	Object of class eeg_stats
row.names	Kept for compatability with S3 generic, ignored.
optional	Kept for compatability with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE.
coords	Include electrode coordinates in output (ignored if long = FALSE)
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

---

as.data.frame.eeg_tfr *Convert* eeg_tfr *objects to data frames*

Description

Convert eeg_tfr objects to data frames

Usage

```
## S3 method for class 'eeg_tfr'
as.data.frame(x, row.names = NULL, optional = FALSE, long = FALSE, ...)
```

Arguments

x	Object of class eeg_tfr
row.names	Kept for compatability with S3 generic, ignored.
optional	Kept for compatability with S3 generic, ignored.
long	Convert to long format. Defaults to FALSE.
...	arguments for other as.data.frame commands

Author(s)

Matt Craddock <matt@mattcraddock.com>

---

browse_data *Browse EEG data.*

Description

A Shiny gadget for browsing EEG data and ICA decompositions interactively. With EEG data (epoched or continuous), data can be viewed as a butterfly plot (all electrodes overlaid) or as individual traces (electrodes "stacked"). Currently, the scale cannot be manually set and is determined by the range of the viewable data. With

Usage

```
browse_data(data, ...)

## S3 method for class 'eeg_ICA'
browse_data(data, ...)

## S3 method for class 'eeg_data'
browse_data(data, sig_length = 5, n_elecs = NULL, downsample = TRUE, ...)

## S3 method for class 'eeg_epochs'
browse_data(data, sig_length = 5, n_elecs = NULL, downsample = FALSE, ...)
```

Arguments

data	eeg_data, eeg_epochs, or eeg_ICA object to be plotted.
...	Other parameters passed to browsing functions.
sig_length	Length of signal to be plotted initially (seconds if continuous, epochs if epoched).
n_elecs	Number of electrodes to be plotted on a single screen. (not yet implemented)
downsample	Only works on eeg_data or eeg_epochs objects. Reduces size of data by only plotting every 4th point, speeding up plotting considerably. Defaults to TRUE for eeg_data, FALSE for eeg_epochs

Methods (by class)

- eeg_ICA: View eeg_ICA component properties
- eeg_data: Browse continuous EEG data.
- eeg_epochs: Browse epoched EEG data.

Author(s)

Matt Craddock <matt@mattcraddock.com>

---

channels *Modify channel information*

Description

Get or set the contents of the channel information inside eegUtils objects.

Usage

```
channels(.data)

channels(.data) <- value
```

Arguments

.data	eegUtils object to view
value	Value to replace chan_info structure with.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
channels(demo_epochs)
```

---

channel_names *Retrieve signal/channel names*

Description

Get the names of the signals element of eegUtils objects.

Usage

```
channel_names(.data)
```

Arguments

.data	eegUtils object

Examples

```
channel_names(demo_epochs)
```

---

channel_stats *Channel statistics*

Description

Channel statistics

Usage

```
channel_stats(data, ...)

## S3 method for class 'eeg_data'
channel_stats(data, ...)
```

Arguments

data	Data as a eeg_data or eeg_epochs object.
...	Other parameters passed to the functions.

Value

A data frame with statistics for each channel.

Methods (by class)

- eeg_data: Calculate channel statistics for eeg_data objects.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
channel_stats(demo_epochs)
```

---

compute_csd *Convert to Current Source Density*

Description

Convert an eeg_data or eeg_epochs object to using Current Source Densities. This command uses a spherical spline algorithm (Perrin et al., 1989) to compute scalp surface Laplacian/current source density estimates of scalp potentials, a reference-free measure of electrical activity that emphasises more local spatial features

Usage

```
compute_csd(data, ...)

## Default S3 method:
compute_csd(data, ...)

## S3 method for class 'eeg_data'
compute_csd(data, m = 4, smoothing = 1e-05, scaling = 1, ...)

## S3 method for class 'eeg_epochs'
compute_csd(data, m = 4, smoothing = 1e-05, scaling = 1, ...)
```

Arguments

data	eeg_data or eeg_epochs object
...	Other parameters
m	smoothing constraint (higher = more rigid splines)
smoothing	lambda constant. Added to the Defaults to 1e-05
scaling	Default scaling (1) is uV / m^2. Note that this depends on the units of the electrode co-ordinates.

Methods (by class)

- default: Default method to detect unknown classes.
- eeg_data: Transform eeg_data to CSD
- eeg_epochs: Transform eeg_data to CSD

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

- Perrin, F., Pernier, J., Bertrand, O., Echallier, J.F. (1989). Spherical splines for scalp potential and current density mapping. Electroencephalography and Clinical Neurophysiology, 72(2), 184-187. PMID: 2464490
- Kayser, J., Tenke, C.E. (2006). Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: I. Evaluation with auditory oddball tasks. Clinical Neurophysiology, 117(2), 348-368.
- Kayser, J., Tenke, C.E. (2015). Issues and considerations for using the scalp surface Laplacian in EEG/ERP research: A tutorial review. International Journal of Psycholphysiology, 97(3), 189-209

Examples

```
csd_epochs <- compute_csd(demo_epochs)
plot_butterfly(csd_epochs)

# Compare the topographies of the CSD vs average referenced data
topoplot(demo_epochs, c(.2, .21))
topoplot(csd_epochs, c(.2, .21))
```

compute_itc *Calculate inter-trial coherence*

Description

Calculates inter-trial coherence (ITC), a measure of phase consistency across single trial data. Input data must be provided as complex Fourier coefficients within an `eeg_tfr` object

Usage

```
compute_itc(data)
```

Arguments

`data`	An `eeg_tfr` object

Value

An `eeg_tfr` object

compute_psd *Compute power spectral density*

Description

`compute_psd` returns the PSD calculated using Welch's method for every channel in the data. The output is in microvolts ^2 / Hz. If the object has multiple epochs, it will perform Welch's FFT separately for each epoch and then average them afterwards.

Usage

```
compute_psd(data, ...)

## S3 method for class 'eeg_data'
compute_psd(
  data,
  seg_length = NULL,
  noverlap = NULL,
  n_fft = NULL,
  method = "Welch",
  demean = TRUE,
  verbose = TRUE,
  ...
)

## S3 method for class 'eeg_epochs'
```

```
compute_psd(
  data,
  seg_length = NULL,
  noverlap = NULL,
  n_fft = 256,
  method = "Welch",
  keep_trials = TRUE,
  demean = TRUE,
  verbose = TRUE,
  ...
)

## S3 method for class 'eeg_evoked'
compute_psd(
  data,
  seg_length = NULL,
  noverlap = NULL,
  n_fft = 256,
  method = "Welch",
  demean = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

data	Data to be plotted. Accepts objects of class eeg_data
...	any further parameters passed to specific methods
seg_length	Length of rolling data segments. Defaults to n_fft. Must be <= n_fft.
noverlap	Number of (sampling) points of overlap between segments. Must be <= seg_length.
n_fft	Length of FFT to be calculated in sampling points. See details.
method	Defaults to "Welch". No other method currently implemented.
demean	Remove channel/epoch means. TRUE by default.
verbose	Print informative messages. TRUE by default.
keep_trials	Include FFT for every trial in output, or average over them if FALSE.

Details

Welch's FFT splits the data into multiple segments, calculates the FFT separately for each segment, and then averages over segments. Each segment is windowed with a Hanning window to counter spectral leakage. For epoched data, Welch's FFT is calculated separately for each trial.

The number of sampling points used for the FFT can be specified using n_fft. n_fft defaults to 256 sampling points for eeg_epochs data, or the minimum of 2048 or the length of the signal for continuous eeg_data.

seg_length defaults to be n_fft, and must be less than or equal to it.

noverlap specifies the amount of overlap between windows in sampling points. If NULL, it defaults to 50\

Value

Currently, a data frame with the PSD for each channel separately.

Methods (by class)

- `eeg_data`: Compute PSD for an `eeg_data` object
- `eeg_epochs`: Compute PSD for an `eeg_epochs` object
- `eeg_evoked`: Compute PSD for an `eeg_evoked` object

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
out <- compute_psd(demo_epochs)

out <- compute_psd(demo_epochs, n_fft = 256, seg_length = 128)
```

---

compute_tfr *Compute Time-Frequency representation of EEG data*

Description

This function creates a time frequency representation of EEG time series data. Currently, it is possible to use either Morlet wavelets or Hanning tapers during the decomposition, which uses convolution in the frequency domain.

Usage

```
compute_tfr(data, ...)

## Default S3 method:
compute_tfr(data, ...)

## S3 method for class 'eeg_epochs'
compute_tfr(
  data,
  method = "morlet",
  foi,
  n_freq,
  spacing = "linear",
  n_cycles = 7,
  keep_trials = FALSE,
  output = "power",
  downsample = 1,
```

```
  verbose = TRUE,
  ...
)

## S3 method for class 'eeg_evoked'
compute_tfr(
  data,
  method = "morlet",
  foi,
  n_freq,
  spacing = "linear",
  n_cycles = 7,
  keep_trials = FALSE,
  output = "power",
  downsample = 1,
  verbose = TRUE,
  ...
)
```

Arguments

data	An object of class eeg_epochs.
...	Further TFR parameters
method	Time-frequency analysis method. Defaults to "morlet".
foi	Frequencies of interest. Scalar or character vector of the lowest and highest frequency to resolve.
n_freq	Number of frequencies to be resolved. Must be an integer number of frequencies.
spacing	Use "linear" or "log" spacing for the frequency vector and number of cycles.
n_cycles	Number of cycles at each frequency. If a single integer, use a constant number of cycles at each frequency. If a character vector of length 2, the number of cycles will scale with frequency from the minimum to the maximum.
keep_trials	Keep single trials or average over them before returning. Defaults to FALSE.
output	Sets whether output is power, phase, or fourier coefficients.
downsample	Downsampling factor. Integer. Selects every n samples after performing time-frequency analysis on the full sampling rate data.
verbose	Print informative messages in console.

Value

An object of class eeg_tfr

Methods (by class)

- default: Default method for compute_tfr
- eeg_epochs: Default method for compute_tfr
- eeg_evoked: Method for eeg_evoked objects.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
out <- compute_tfr(demo_epochs, method = "morlet", foi = c(4, 30), n_freq = 10, n_cycles = 3)
out
out$freq_info$morlet_resolution
out <- compute_tfr(demo_epochs, method = "morlet", foi = c(4, 30), n_freq = 10, n_cycles = c(3, 10))
out$freq_info$morlet_resolution
```

---

cycle_calc *Calculate cycles*

Description

A helper function for calculating the appropriate min-max cycles for a fixed time window/frequency resolution for use with `compute_tfr`. For some analyses you may wish to keep a fixed frequency resolution across the range being analysed, which requires using a fixed time window. `compute_tfr` expects the minimum and maximum number of cycles to be supplied. Use this function to calculate the equivalent number of cycles at each frequency.

Usage

```
cycle_calc(time_win, frex)
```

Arguments

`time_win`	Time window in seconds.
`frex`	Frequencies of interest.

Value

the number of cycles for each frequency of interest

See Also

`compute_tfr`

Examples

```
cycle_calc(.5, seq(3, 30, length.out = 10))
no_scale_tfr <- compute_tfr(demo_epochs, foi = c(3, 30),
 n_cycles = range(cycle_calc(0.5, seq(3, 30, length.out = 10))),
  n_freq = 10)
```

demo_epochs *A demo* eeg_epochs *dataset*

Description

This is a small dataset with 11 electrodes and 80 epochs used for demonstrating some features of the eegUtils package.

Usage

```
demo_epochs
```

Format

An eeg_epochs object

demo_spatial *A demo* eeg_epochs *dataset*

Description

This is a small dataset with 23 electrodes and 80 epochs used for demonstrating some features of the eegUtils package. It is part of a recording from a visual spatial cueing task.

Usage

```
demo_spatial
```

Format

An eeg_epochs object

eeg_average *Calculate averages (e.g. ERPs) for single datasets*

Description

This function is used to create an eeg_evoked object from eeg_epochs. By default, it will try to keep different conditions in the data separate using the epochs metadata from the object, thus yielding one average per condition. Alternatively, the user can specify which averages they want using the cols argument.

Usage

```
eeg_average(data, ...)

## Default S3 method:
eeg_average(data, ...)

## S3 method for class 'eeg_epochs'
eeg_average(data, cols = NULL, ...)

## S3 method for class 'eeg_evoked'
eeg_average(data, cols = NULL, ...)

## S3 method for class 'eeg_tfr'
eeg_average(data, cols = NULL, ...)
```

Arguments

data	An eeg_epochs of eeg_tfr object.
...	Other arguments passed to the averaging functions
cols	Columns from the epochs structure that the average should group on. NULL, the default, uses all columns other than the epoch column.

Value

An object of class eeg_evoked if applied to eeg_epochs; eeg_tfr if applied to eeg_tfr.

Methods (by class)

- default: Default method for averaging EEG objects
- eeg_epochs: Create evoked data from eeg_epochs
- eeg_evoked: average an eeg_epochs object over epochs.
- eeg_tfr: average an eeg_tfr object over epochs.

Author(s)

Matt Craddock <matt@mattcraddock.com>

eeg_combine *Combine* eegUtils *objects*

Description

Combine multiple eeg_epochs, eeg_data, or eeg_evoked objects into a single object. The function will try to check the participant_id entry in the epochs structure to see if the data comes from a single participant or from multiple participants. If the data is from a single participant, it will concatenate the objects and attempt to correct them so that the trial numbers and timings are correct.

Usage

```
eeg_combine(data, ...)

## S3 method for class 'list'
eeg_combine(data, ...)

## S3 method for class 'eeg_data'
eeg_combine(data, ..., check_timings = TRUE)

## S3 method for class 'eeg_epochs'
eeg_combine(data, ..., check_timings = TRUE)

## S3 method for class 'eeg_evoked'
eeg_combine(data, ...)
```

Arguments

data	An eeg_data, eeg_epochs, or eeg_evoked object, or a list of such objects.
...	additional eeg_data or eeg_epochs objects
check_timings	Check whether sample times / epoch numbers are continuously ascending; if not, modify so that they are. Useful when, for example, combining epochs derived from multiple recording blocks. Defaults to TRUE

Value

If all objects have the same participant_id, returns an object of the same class as the original input object. If the objects have different participant_id numbers, an object of both class eeg_group and the same class as the original input object.

Methods (by class)

- list: Method for combining lists of eeg_data and eeg_epochs objects.
- eeg_data: Method for combining eeg_data objects.
- eeg_epochs: Method for combining eeg_epochs objects
- eeg_evoked: Method for combining eeg_evoked objects

Author(s)

Matt Craddock, <matt@mattcraddock.com>

`eeg_decompose` *Generalized eigenvalue decomposition based methods for EEG data*

Description

Implements a selection of Generalized Eigenvalue based decomposition methods for EEG signals. Intended for isolating oscillations at specified frequencies, decomposing channel-based data into components reflecting distinct or combinations of sources of oscillatory signals. Currently, spatio-spectral decomposition (Nikulin, Nolte, & Curio, 2011) and Rhythmic Entrainment Source Separation (Cohen & Gulbinate, 2017) are implemented. The key difference between the two is that the former returns the results of the data-derived spatial filters applied to the bandpass-filtered "signal" data, whereas the latter returns the results of the filters applied to the original, broadband data.

Usage

```
eeg_decompose(data, ...)

## S3 method for class 'eeg_epochs'
eeg_decompose(
  data,
  sig_range,
  noise_range,
  method = "ssd",
  verbose = TRUE,
  order = 2,
  ...
)
```

Arguments

`data`	An `eeg_data` object
`...`	Additional parameters
`sig_range`	Vector with two inputs, the lower and upper bounds of the frequency range of interest
`noise_range`	Range of frequencies to be considered noise (e.g. bounds of flanker frequencies)
`method`	Type of decomposition to apply. Currently only "ssd" is supported.
`verbose`	Informative messages printed to console. Defaults to TRUE.
`order`	Filter order for filter applied to signal/noise

Value

An `eeg_ICA` object. Note that

Methods (by class)

- eeg_epochs: method for eeg_epochs objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

Cohen, M. X., & Gulbinate, R. (2017). Rhythmic entrainment source separation: Optimizing analyses of neural responses to rhythmic sensory stimulation. NeuroImage, 147, 43-56. https://doi.org/10.1016/j.neuroimage.2016.

Haufe, S., Dähne, S., & Nikulin, V. V. (2014). Dimensionality reduction for the analysis of brain oscillations. NeuroImage, 101, 583–597. https://doi.org/10.1016/j.neuroimage.2014.06.073

Nikulin, V. V., Nolte, G., & Curio, G. (2011). A novel method for reliable and fast extraction of neuronal EEG/MEG oscillations on the basis of spatio-spectral decomposition. NeuroImage, 55(4), 1528–1535. https://doi.org/10.1016/j.neuroimage.2011.01.057

See Also

Other decompositions: run_ICA()

Examples

```
# The default method is Spatio-Spectral Decomposition, which returns
# spatially and temporally filtered source timecourses.
 decomposed <-
   eeg_decompose(demo_epochs,
                 sig_range = c(9, 11),
                 noise_range = c(8, 12),
                 method = "ssd")
 plot_psd(decomposed)
 # We can plot the spatial filters using `topoplot()`
 topoplot(decomposed, 1:2)
 plot_timecourse(decomposed, 1)
# method = "ress" returns spatially but not temporally filtered timecourses.
 with_RESS <-
   eeg_decompose(demo_epochs,
                 sig_range = c(9, 11),
                 noise_range = c(8, 12),
                 method = "ress")
 plot_psd(with_RESS)
 # The topographical plots are identical to those using "ssd", as the
 # spatial filters are the same.
 topoplot(with_RESS, 1:2)
 plot_timecourse(with_RESS, 1)
```

`eeg_downsample` *Downsampling EEG data*

Description

Performs low-pass anti-aliasing filtering and downsamples EEG data by a specified factor. This is a wrapper for `decimate` from the `signal` package. Note that this will also adjust the event table, moving events to the nearest time remaining after downsampling

Usage

```
eeg_downsample(data, ...)

## S3 method for class 'eeg_data'
eeg_downsample(data, q, ...)

## S3 method for class 'eeg_epochs'
eeg_downsample(data, q, ...)
```

Arguments

`data`	An `eeg_data` object to be downsampled
`...`	Parameters passed to functions
`q`	Integer factor to downsample by

Methods (by class)

- `eeg_data`: Downsample eeg_data objects
- `eeg_epochs`: Downsample eeg_epochs objects

Author(s)

Matt Craddock `<matt@mattcraddock.com>`

Examples

```
eeg_downsample(demo_epochs, 2)
```

eeg_epochs *Function to create an S3 object of class* eeg_epochs.

Description

Function to create an S3 object of class eeg_epochs.

Usage

```
eeg_epochs(
  data,
  srate,
  events = NULL,
  chan_info = NULL,
  timings = NULL,
  reference = NULL,
  epochs = NULL
)
```

Arguments

data	Raw data - signals from electrodes/channels.
srate	Sampling rate in Hz.
events	Event table
chan_info	String of character names for electrodes.
timings	Timing information - samples and sample /sampling rate.
reference	Reference channel information, including names of reference channels, excluded channels etc.
epochs	Information about the epochs contained in the data.

Author(s)

Matt Craddock <matt@mattcraddock.com>

eeg_filter *Filter EEG data*

Description

Perform IIR or FIR filtering on input EEG data of class eeg_data or eeg_epochs. WARNING: with epoched data, epoch boundaries are currently ignored, which can result in minor edge artifacts.

Usage

```
eeg_filter(data, ...)

## S3 method for class 'eeg_data'
eeg_filter(
  data,
  low_freq = NULL,
  high_freq = NULL,
  filter_order = "auto",
  trans_bw = "auto",
  method = "fir",
  window = "hamming",
  demean = TRUE,
  ...
)

## S3 method for class 'eeg_epochs'
eeg_filter(
  data,
  low_freq = NULL,
  high_freq = NULL,
  filter_order = "auto",
  trans_bw = "auto",
  method = "fir",
  window = "hamming",
  demean = TRUE,
  ...
)

## S3 method for class 'eeg_group'
eeg_filter(
  data,
  low_freq = NULL,
  high_freq = NULL,
  filter_order = "auto",
  trans_bw = "auto",
  method = "fir",
  window = "hamming",
  demean = TRUE,
  ...
)
```

Arguments

data	An eeg_data or eeg_epochs object to be filtered.
...	Additional parameters.
low_freq	Low cutoff frequency.

high_freq	High cutoff frequency.
filter_order	Defaults to "auto", which automatically estimates filter order for the specified filter characteristics (defaults to 4 if method = "iir").
trans_bw	Transition bandwidth of the filter. "auto" or an integer. "auto" attempts to determine a suitable transition bandwidth using the heuristic given below. Ignored if method = "iir".
method	"fir" (Finite Impulse Response) or "iir" (Infinite Impulse Response). Defaults to "fir".
window	Windowing function to use (FIR filtering only). Defaults to "hamming"; currently only "hamming" available.
demean	Remove DC component (i.e. channel/epoch mean) before filtering. Defaults to TRUE.

Details

low_freq and high_freq are the low and high cutoff frequencies. Pass low freq or high freq alone to perform high-pass or low-pass filtering respectively. For band-pass or band-stop filters, pass both low_freq and high_freq.

If low_freq < high_freq, bandpass filtering is performed.

If low_freq > high_freq, bandstop filtering is performed.

Note that it is recommended to first zero-mean the signal using either channel means or by-channel epoch means.

The function allows parallelization using the `future` package, e.g. using `plan(multiprocess)`

Value

An object of the original class with signals filtered according to the user's specifications

FIR versus IIR filtering

Finite Impulse Response (FIR) filtering is performed using an overlap-add FFT method. Note that this only performs a single-pass; the data is shifted back in time by the group delay of the filter to compensate for the phase delay imposed by the linear filtering process. Infinite Impulse Response (IIR) filtering is performed using a two-pass (once forwards, once reversed) method to correct for phase alignment. Note that the Butterworth filter designs used here can become numerically unstable with only a small increase in filter order. For most purposes, use FIR filters.

Examples

```
plot_psd(eeg_filter(demo_epochs, low_freq = 1, high_freq = 30))
plot_psd(eeg_filter(demo_epochs, low_freq = 12, high_freq = 8))
plot_psd(eeg_filter(demo_epochs, low_freq = 12, high_freq = 8, method = "iir"))
```

eeg_ICA *Function to create an S3 object of class* eeg_ICA.

Description

Function to create an S3 object of class eeg_ICA.

Usage

```
eeg_ICA(
  mixing_matrix,
  unmixing_matrix,
  signals,
  timings,
  events,
  chan_info,
  srate,
  epochs,
  algorithm
)
```

Arguments

mixing_matrix	ICA mixing matrix
unmixing_matrix	ICA unmixing matrix
signals	ICA component timecourses.
timings	Unique timepoints remaining in the data.
events	event table
chan_info	A data frame containing electrode labels and coordinates.
srate	Sampling rate in Hz. A numeric value.
epochs	A data frame containing meta-information about the epochs contained in the data, such as participant ID label and condition labels for epochs.
algorithm	The method used to calculate the decomposition.

Value

An object of class eeg_ICA.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Matt Craddock <matt@mattcraddock.com>

`eeg_reference` *Referencing*

Description

Used to reference the EEG data to a specified electrode or electrodes. Defaults to average reference. When specific electrodes are used, they are removed from the data. Meta-data about the referencing scheme is held in the `eeg_data` structure.

Usage

```
eeg_reference(data, ...)

## Default S3 method:
eeg_reference(data, ...)

## S3 method for class 'eeg_data'
eeg_reference(
  data,
  ref_chans = "average",
  exclude = NULL,
  robust = FALSE,
  implicit_ref = NULL,
  verbose = TRUE,
  ...
)

## S3 method for class 'eeg_epochs'
eeg_reference(
  data,
  ref_chans = "average",
  exclude = NULL,
  robust = FALSE,
  implicit_ref = NULL,
  verbose = TRUE,
  ...
)

reref_eeg(data, ...)
```

Arguments

`data`	Data to re-reference. Primarily meant for use with data of class `eeg_data`.
`...`	Further parameters to be passed to `eeg_reference`
`ref_chans`	Channels to reference data to. Defaults to "average" i.e. average of all electrodes in data. Character vector of channel names or numbers.

exclude	Electrodes to exclude from average reference calculation.
robust	Use median instead of mean; only used for average reference. Defaults to FALSE.
implicit_ref	Implicit reference channel - use this to add a channel back that was previously used as a reference. E.g. if the LM (left mastoid) channel was used in recording and is absent from the data, passing "LM" adds an "LM" channel back to the data, populated with zeroes.
verbose	Print informative messages in console. Defaults to TRUE.

Value

object of class eeg_data, re-referenced as requested.

Methods (by class)

- default: Default method
- eeg_data: Rereference objects of class eeg_data
- eeg_epochs: Rereference objects of class eeg_epochs

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
# demo_epochs is average referenced by default
demo_epochs
# Rereference it but exclude B5 from calculation of the average
eeg_reference(demo_epochs, exclude = "B5")
# Reference data using the median of the reference channels rather than the mean
eeg_reference(demo_epochs, robust = TRUE)

eeg_reference(demo_spatial)
eeg_reference(demo_spatial, ref_chans = "Fz")
eeg_reference(demo_spatial, implicit_ref = "LM")
```

---

electrode_locations *Get standard electrode locations*

Description

Joins standard electrode locations to EEG data from eegUtils internal data.

Usage

```
electrode_locations(data, ...)

## S3 method for class 'data.frame'
electrode_locations(
  data,
  electrode = "electrode",
  drop = FALSE,
  montage = NULL,
  ...
)

## S3 method for class 'eeg_data'
electrode_locations(data, drop = FALSE, montage = NULL, overwrite = FALSE, ...)

## S3 method for class 'eeg_epochs'
electrode_locations(data, drop = FALSE, montage = NULL, overwrite = FALSE, ...)

## S3 method for class 'eeg_tfr'
electrode_locations(data, drop = FALSE, montage = NULL, overwrite = FALSE, ...)
```

Arguments

data	An EEG dataset.
...	Passed to S3 methods.
electrode	The column name containing electrode names in data. (Defaults to "electrode").
drop	Should electrodes in data for which default locations are not available be removed? (Defaults to FALSE).
montage	Name of an existing montage set. Defaults to NULL.
overwrite	Overwrite existing channel info. Defaults to FALSE.

Details

The standard locations are from the 10-05 system derived by Oostenveld & Praamstra (2001). In addition, there are multiple specific montages for BioSemi systems included. These can be added using the montage parameter: "biosemi16", "biosemi32", biosemi64", "biosemi64alpha", "biosemi128", "biosemi160", "biosemi256"

Value

A tibble (or data.frame), or ggplot2 object if plot = TRUE.

Methods (by class)

- data.frame: Adds standard locations to a data frame in long format
- eeg_data: Adds standard locations to the chan_info field of an eeg_data object.
- eeg_epochs: Adds standard locations to the chan_info field of an eeg_data object.
- eeg_tfr: Adds standard locations to the chan_info field of an eeg_tfr object.

References

Oostenveld, R. & Praamstra, P. (2001). The five percent electrode system for high-resolution EEG and ERP measurements. Clinical Neurophysiology, 112, 4, 713-719

Examples

```
channels(demo_epochs)
electrode_locations(demo_epochs, overwrite = TRUE, montage = "biosemi64alpha")
```

---

epochs *Modify the epochs structure*

Description

Get or set the epochs structure of an `eegUtils` object.

Usage

```
epochs(data)

epochs(data) <- value
```

Arguments

`data`	`eegUtils` object to view
`value`	Structure to replace `epochs` structure with.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
epochs(demo_spatial)
```

epoch_data *Create epochs from EEG data*

Description

Creates epochs around specified event triggers. Requires data of class `eeg_data`. Where multiple events are specified, epochs will be created around each event.

Usage

```
epoch_data(data, ...)

## S3 method for class 'eeg_data'
epoch_data(
  data,
  events,
  time_lim = c(-1, 1),
  baseline = "none",
  epoch_labels = NULL,
  ...
)
```

Arguments

`data`	Continuous data to be epoched.
`...`	Parameters passed to functions
`events`	Character vector of events to epoch around.
`time_lim`	Time in seconds to form epoch around the events. Defaults to one second either side.
`baseline`	Baseline times to subtract. Can be set to a numeric vector of length two to specify a time window to use as a baseline in each epoch (e.g. c(-.1, 0)), "none", which will perform no baseline correction, or `NULL` to use the mean of the whole epoch. As of v0.6 of `eegUtils`, the default is "none".
`epoch_labels`	Character vector of same length as events which'll be used to label the epochs.

Value

Returns an epoched object of class `eeg_epochs`

Methods (by class)

- `eeg_data`: Epoch `eeg_data` objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

epoch_data.default *Create epochs from EEG data*

Description

Create epochs from EEG data

Usage

```
## Default S3 method:
epoch_data(data, ...)
```

Arguments

data	Continuous data to be epoched.
...	Parameters passed to functions

epoch_stats *Epoch statistics*

Description

Calculate various statistics for each epoch in the data

Usage

```
epoch_stats(data, ...)

## S3 method for class 'eeg_epochs'
epoch_stats(data, ...)
```

Arguments

data	Data as a eeg_data or eeg_epochs object.
...	Other parameters passed to the functions.

Methods (by class)

- eeg_epochs: Calculate statistics for each epoch.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
epoch_stats(demo_epochs)
```

erp_image *Plot ERP images*

Description

Plot an ERP image from a single electrode. Uses a boxcar smooth over a series of trials in order to make across-trial patterns more apparent.

Usage

```
erp_image(data, ...)

## S3 method for class 'data.frame'
erp_image(
  data,
  electrode = "Cz",
  time_lim = NULL,
  smoothing = 10,
  clim = NULL,
  interpolate = FALSE,
  na.rm = TRUE,
  ...
)

## S3 method for class 'eeg_epochs'
erp_image(
  data,
  electrode = "Cz",
  time_lim = NULL,
  smoothing = 10,
  clim = NULL,
  interpolate = FALSE,
  na.rm = TRUE,
  ...
)

## S3 method for class 'eeg_ICA'
erp_image(
  data,
  component = "Comp001",
  smoothing = 10,
  clim = NULL,
  interpolate = FALSE,
  na.rm = TRUE,
  ...
)
```

```
## S3 method for class 'eeg_tfr'
erp_image(
  data,
  electrode = "Cz",
  time_lim = NULL,
  smoothing = 10,
  clim = NULL,
  interpolate = FALSE,
  freq_range = NULL,
  na.rm = TRUE,
  ...
)
```

Arguments

data	Data frame or eegUtils object to be plotted.
...	Other arguments passed to the method.
electrode	Electrode for which to generate an ERP image.
time_lim	Time limits of plot.
smoothing	Number of trials to smooth over when generating image.
clim	Character vector of min and max values of plotting colour range. e.g. c(-5,5). Defaults to min and max.
interpolate	Perform interpolation to produce smoother looking plots. Defaults to FALSE.
na.rm	Remove trials with NA amplitudes after smoothing. Defaults to TRUE.
component	eeg_ICA component to plot
freq_range	A numeric vector specify the range of frequencies to average over. A single number will find the closest matching frequency. A vector of length two will match average over frequencies within that range.

Value

A ggplot object

Methods (by class)

- data.frame: Default function operates on normal data frames
- eeg_epochs: Create an erp_image from eeg_epochs
- eeg_ICA: Plot component image from eeg_ICA
- eeg_tfr: Plot component image from eeg_tfr

Author(s)

Matt craddock <matt@mattcraddock.com>

Examples

```
erp_image(demo_epochs, electrode = "A31")
erp_image(demo_epochs, electrode = "A31", interpolate = TRUE)
erp_image(demo_epochs, electrode = "A31", smoothing = 5)
erp_image(compute_tfr(demo_epochs,
 foi = c(4, 30), n_cycles = 3, n_freq = 20, verbose = FALSE, keep_trials = TRUE),
 electrode = "A31", freq_range = c(8, 12))
```

---

erp_raster *ERP raster plot*

Description

Create a plot showing the ERP at every channel as a single ERP image. By default, this attempts to find channel locations and rearrange the channels such that spatial patterns on the scalp are more discernible. It orders the rows from the most posterior electrode on the right hemisphere to the most anterior electrode on the left hemisphere, with midline electrodes in the middle. If no locations are found, it simply displays the data in its original order.

Usage

```
erp_raster(
  data,
  anat_order = TRUE,
  time_lim = NULL,
  clim = NULL,
  interpolate = FALSE
)
```

Arguments

data	An eeg_epochs object
anat_order	Arrange the channels in a more anatomically representative order. Defaults to TRUE.
time_lim	Time limits of plot - should be a character vector (e.g. c(-.2, .5))
clim	Character vector of min and max values of plotting colour range. e.g. c(-5,5). Defaults to min and max.
interpolate	Smooth the raster plot. Defaults to FALSE.

Value

A ggplot object

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
library(ggplot2)
erp_raster(demo_epochs)
erp_raster(demo_epochs, interpolate = TRUE)
erp_raster(rm_baseline(demo_epochs, c(-.1, 0)), interpolate = TRUE)
erp_raster(demo_spatial) + facet_wrap(~epoch_labels)
```

`erp_scalp` *Plot event-related potentials using a scalp based layout*

Description

Creates a `ggplot2` figure showing the ERP at each electrode, with each electrode placed according to its location on the scalp.

Usage

```
erp_scalp(data, ...)

## Default S3 method:
erp_scalp(
  data,
  electrode = "electrode",
  amplitude = "amplitude",
  time = "time",
  color = NULL,
  colour = NULL,
  size = 0.8,
  baseline = NULL,
  show_guide = TRUE,
  chan_info = NULL,
  montage = NULL,
  ...
)
```

Arguments

`data`	An EEG dataset.
`...`	Various arguments passed to specific functions
`electrode`	Column name containing electrode names in data. Defaults to "electrode".
`amplitude`	Column name containing amplitudes in data. Defaults to "amplitude".
`time`	Column name containing time in data. Defaults to "time".
`color`	Alias for `colour`.
`colour`	Variable to colour lines by. If no variable is passed, only one line is drawn for each electrode.

size	Size of the line(s) for the ERPs.
baseline	Character vector of times to subtract for baseline correct.
show_guide	Should a guide showing the scale of the ERP plots be shown. Defaults to TRUE.
chan_info	Pass channel information in the standard eegUtils format directly.
montage	Name of an existing montage set. Defaults to NULL, which will attempt to us locations from the data. If none are found, it will attempt to use standard 10-05 locations.

Value

Returns a ggplot2 plot object.

Methods (by class)

- default: Plot an ERP scalp map

Author(s)

Matti Vuorre, <mv2521@columbia.edu>

Matt Craddock, <matt@mattcraddock.com>

See Also

interactive_scalp() for interactive plots of ERPs in a scalp-based layout.

Other scalp-based maps: interactive_scalp(), topoplot()

Examples

```
erp_scalp(demo_epochs, montage = "biosemi64alpha")
```

---

events *Modify events structure*

Description

Get or set the values in the events structure of an eegUtils object.

Usage

```
events(.data)

events(.data) <- value

## S3 replacement method for class 'eeg_epochs'
events(.data) <- value

## S3 replacement method for class 'eeg_data'
events(.data) <- value
```

Arguments

.data	eegUtils object to view
value	Value to replace events structure with.

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

Other event handlers: list_epochs(), list_events(), tag_events()

Examples

```
events(demo_epochs)
events(demo_epochs) <- mutate(events(demo_epochs),
 sf = dplyr::case_when(
         event_type %% 2 == 0 ~ "HSF",
         event_type %% 2 == 1 ~ "LSF",
 ))
events(demo_epochs)
```

---

export_bva *Export continuous data in Brain Vision Analyzer format*

Description

Export continuous EEG data in Brain Vision Analyzer format. This is one of the recommended formats for BIDS https://bids-specification.readthedocs.io/en/stable/04-modality-specific-files/03-electroencephalography.html

Usage

```
export_bva(.data, filename, orientation, verbose = TRUE)

## S3 method for class 'eeg_data'
export_bva(.data, filename, orientation = "VECTORIZED", verbose = TRUE)
```

Arguments

.data	eeg_data object to be exported.
filename	String giving filename to export to. File extensions will be removed when supplied.
orientation	VECTORIZED or MULTIPLEXED. This relates to the way the data is stored in the binary file. VECTORIZED is the default and recommended.
verbose	print informative messages to console

Methods (by class)

- eeg_data: Method for eeg_data

---

fit_glm *Fit a linear model to EEG data*

Description

Fits a linear model to each timepoint separately for each electrode.

Usage

```
fit_glm(formula, data, ...)

## S3 method for class 'eeg_epochs'
fit_glm(formula, data, time_lim = NULL, ...)
```

Arguments

formula	An object of class formula. Right-hand side A regression formula for a GLM. See ?formula and notes on use below
data	An eegUtils object.
...	Any other arguments passed to (LM/GLM)
time_lim	Numeric vector of length 2 specifying time period to be used as a baseline.

Methods (by class)

- eeg_epochs: GLM fitting for eeg_epochs

Notes On Usage

The fit_glm function will fit a linear model to each timepoint for each electrode in the input dataset.

The formula is a standard R formula. Specify only the right-hand side of the formula i.e. any predictors.

The function allows flexible fitting of a baseline covariate, recognising the term baseline in the specified formula.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

geom_topo *Create a topographical plot*

Description

geom_topo() creates a topographical plot as a ggplot2 object. This function automatically combines a number of distinct geom_* and stat_* functions to create a default topographical scalp map. Since geom_raster does not allow unevenly spaced grids, the function creates an interpolated surface.

Usage

```
geom_topo(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  show.legend = NA,
  na.rm = TRUE,
  inherit.aes = TRUE,
  interpolate = FALSE,
  interp_limit = "skirt",
  chan_markers = "point",
  chan_size = rel(2),
  head_size = rel(1.5),
  r = NULL,
  grid_res = 200,
  method = "biharmonic",
  bins = 6,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).

stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().
interpolate	If TRUE interpolate linearly, if FALSE (the default) don't interpolate.
interp_limit	Topoplot with a "skirt" or inside the "head".
chan_markers	Defaults to "point". Mark electrode positions with points or text.
chan_size	Size for channel markers, if any.
head_size	Size of the head shape.
r	Head circumference
grid_res	Smoothness of the interpolation grid.
method	"biharmonic" or ""gam".
bins	Number of bins to use for contour lines.
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

See Also

Other topoplot functions: stat_scalpcontours(), stat_scalpmap()

Examples

```
library(ggplot2)
ggplot(demo_epochs, aes(x = x, y = y, fill = amplitude, z = amplitude)) + geom_topo()
```

---

get_participant_id *Query and set elements of the* epochs *metadata structures*

Description

These functions are used to query and set the participant_id and recording fields of an epochs structure within any eegUtils object. Note that the two set_* functions do not operate on eeg_group objects.

Usage

```
get_participant_id(data)

get_recording(data)

set_participant_id(data, participant_id)

set_recording(data, recording)
```

Arguments

data	An eegUtils object.
participant_id	A character vector giving the name to use for the participant_id for a particular object.
recording	A character vector giving the name to use for the EEG recording.

Value

- get_participant_id() returns a character vector containing the participant_id
- get_recording() returns a character vector containing the recording from values of recording from the eegUtils object
- set_participant_id() returns the full eegUtils object with the participant_id field modified in the epochs structure
- set_recording() returns the full eegUtilsobject with the recording field modified in the epochs structure

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
get_participant_id(demo_epochs)
get_recording(demo_epochs)
set_participant_id(demo_epochs, "002")
set_recording(demo_epochs, "test_recording")
```

---

get_scalpmap *Calculate an interpolated scalpmap*

Description

Calculate and return an interpolated scalpmap from a data.frame or eeg_epochs object. This provides the basis for a raster plot, as used to create topographical plots.

Usage

```
get_scalpmap(data, ...)

## S3 method for class 'data.frame'
get_scalpmap(
  data,
  method = "biharmonic",
  grid_res = 100,
  interp_limit = "skirt",
  quantity = "amplitude",
  facets = NULL,
  r = 95,
  k = -1,
  ...
)

## S3 method for class 'eeg_epochs'
get_scalpmap(
  data,
  method = "biharmonic",
  grid_res = 100,
  interp_limit = "skirt",
  quantity = "amplitude",
  facets = NULL,
  r = 95,
  ...
)
```

Arguments

data	An object of class 'data.frame'
...	Other arguments
method	"biharmonic" or "gam" smooth
grid_res	Grid resolution
interp_limit	Interpolate up to the "head" or add a "skirt"
quantity	Quantity to be plotted. Defaults to "amplitude".
facets	Any facets you plan to use
r	Size of headshape
k	Degrees of freedom used for spline when using method = gam. Defaults to -1, which attempts to automatically determine k.

Value

A tibble with the columns x, y, fill

Methods (by class)

- data.frame: Get scalpmap from a data.frame
- eeg_epochs: Get scalpmap from an eeg_epochs object.

Examples

```
head(get_scalpmap(demo_epochs))
```

---

iir_filt *Butterworth IIR filter*

Description

Construct a Butterworth IIR filter and filter input data. This uses signal::filt_filt, which filters the signal twice to - once forwards, then again backwards).

Usage

```
iir_filt(data, ...)

## S3 method for class 'data.frame'
iir_filt(
  data,
  low_freq = NULL,
  high_freq = NULL,
  filter_order = 4,
  srate,
  silent = FALSE,
  ...
)

## S3 method for class 'eeg_data'
iir_filt(
  data,
  low_freq = NULL,
  high_freq = NULL,
  filter_order = 4,
  silent = FALSE,
  ...
)

## S3 method for class 'eeg_epochs'
iir_filt(
  data,
  low_freq = NULL,
  high_freq = NULL,
```

```
  filter_order = 4,
  silent = FALSE,
  ...
)
```

Arguments

data	Data to be filtered.
...	Parameters passed to S3 methods
low_freq	Low passband edge.
high_freq	High passband edge.
filter_order	Order of the Butterworth filter.
srate	Sampling rate of the signal.
silent	Turns off filtering messages.

Details

low_freq and high_freq are passband edges. Pass low freq or high freq alone to perform high-pass or low-pass filtering respectively. For band-pass or band-stop filters, pass both low_freq and high_freq.

If low_freq < high_freq, bandpass filtering is performed.

If low_freq > high_freq, bandstop filtering is performed.

Note that the signal is first zero-meaned using either channel means or by-channel epoch means.

Methods (by class)

- data.frame: Filter a data frame
- eeg_data: Filter eeg_data objects
- eeg_epochs: Filter eeg_epochs objects.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
plot_psd(iir_filt(demo_epochs, low_freq = 1, high_freq = 30))
plot_psd(iir_filt(demo_epochs, low_freq = 12, high_freq = 8))
```

import_chans *Import channel locations from various file formats*

Description

Currently only ASA `.elc` format with Cartesian x-y-z coordinates is supported.

Usage

```
import_chans(file_name, format = "spherical", file_format = "auto")
```

Arguments

`file_name`	Name and full path of file to be loaded.
`format`	If the file is not `.elc` format, "spherical", "geographic". Default is "spherical".
`file_format`	Default is `auto`, which will use the file extension to determine file format. Other options include `ced`, `besa`, `elp`, `elc`

Value

A `tibble` containing electrode names and locations in several different coordinate systems.

Author(s)

Matt Craddock <matt@mattcraddock.com>

import_erplab *Import from ERPLAB .erp files*

Description

ERPLAB is a toolbox for Event Related Potential analysis associated with EEGLAB. It exports files in Matlab v6.5 format, either with an .erp or .mat extension. Note that ERPLAB files can contain data from multiple subjects, individual subjects, or averaged over multiple subjects. The files do not distinguish between these possibilities, so users will need to apply their own knowledge of the contents to import them correctly.

Usage

```
import_erplab(
  file_name,
  df_out = FALSE,
  participant_id = NULL,
  recording = NULL
)
```

Arguments

file_name	Filename (and path if not in present working directory)
df_out	Defaults to FALSE - outputs an object of class eeg_epochs. Set to TRUE to output a data frame.
participant_id	By default, the filename will be used as the id of the participant.
recording	By default, the filename will be used as the name of the recording.

Value

An object of class eeg_epochs or a data.frame.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
## Not run: import_set("your_data.set")
```

---

import_ft *Import Fieldtrip files*

Description

Fieldtrip is a Matlab package for EEG/MEG processing and analysis.

Usage

```
import_ft(file_name, participant_id = NULL, recording = NULL, verbose = TRUE)
```

Arguments

file_name	Name of file to be imported.
participant_id	Identifier for the participant.
recording	Name of the recording. By default, the filename will be used.
verbose	Informative messages printed to console. Defaults to TRUE.

Value

An object of class eeg_data, eeg_epochs, or eeg_tfr, depending on the type of input data.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
## Not run: import_ft("fieldtrip_test.mat")
```

`import_raw` *Function for reading raw data.*

Description

Currently BDF/EDF, 32-bit .CNT, and Brain Vision Analyzer files are supported. Filetype is determined by the file extension.The `edfReader` package is used to load BDF/EDF files, whereas custom code is used for .CNT and BVA files. The function creates an `eeg_data` structure for subsequent use.

Usage

```
import_raw(
  file_name,
  file_path = NULL,
  recording = NULL,
  participant_id = NA,
  fast_bdf = TRUE
)
```

Arguments

`file_name`	File to import. Should include file extension.
`file_path`	Path to file name, if not included in filename.
`recording`	Name of the recording. By default, the filename will be used.
`participant_id`	Identifier for the participant. Defaults to NA.
`fast_bdf`	New, faster method for loading BDF files. Experimental.

Value

An object of class `eeg_data`

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
## Not run:
import_raw("test_bdf.bdf")

## End(Not run)
```

import_set *Load* EEGLAB *.set files*

Description

Load EEGLAB .set files and convert them to eeg_epochs objects. Supports import of files saved in either Matlab v6.5 or Matlab v7.3 formats. Currently, any ICA weights or decompositions are discarded.

Usage

```
import_set(
  file_name,
  df_out = FALSE,
  participant_id = NULL,
  recording = NULL,
  drop_custom = FALSE,
  verbose = TRUE
)
```

Arguments

file_name	Filename (and path if not in present working directory)
df_out	Defaults to FALSE - outputs an object of class eeg_epochs. Set to TRUE for a normal data frame.
participant_id	Character vector. By default, the filename will be used as the id of the participant.
recording	Character vector. By default, the filename will be used as the name of the recording.
drop_custom	Drop custom event fields. TRUE by default.
verbose	Print informative messages. TRUE by default.

Value

An object of class eeg_epochs

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
## Not run: import_set("your_data.set")
```

interactive_scalp *Interactive scalp maps*

Description

Launches a Shiny Gadget for an interactive version of `erp_scalp`, allowing clicking of individual electrodes to plot them in a separate panel. In that panel, they can be averaged over or plotted as individual electrodes.

Usage

```
interactive_scalp(data, colour = NULL, baseline = NULL, montage = NULL)
```

Arguments

`data`	An EEG dataset.
`colour`	Variable to colour lines by. If no variable is passed, only one line is drawn for each electrode.
`baseline`	Character vector of times to subtract for baseline correction.
`montage`	Name of an existing montage set. Defaults to NULL; (currently only 'biosemi64alpha' available other than default 10/20 system)

Author(s)

Matt Craddock, <matt@mattcraddock.com>

See Also

`erp_scalp()` for non-interactive plots of ERPs in a scalp-based layout.

Other scalp-based maps: `erp_scalp()`, `topoplot()`

interp_elecs *Channel interpolation*

Description

Interpolate EEG channels using a spherical spline (Perrin et al., 1989; 1990). The data must have channel locations attached.

Usage

```
interp_elecs(data, bad_elecs, ...)

## S3 method for class 'eeg_data'
interp_elecs(data, bad_elecs, ...)
```

Arguments

data	Data as a eeg_data or eeg_epochs object.
bad_elecs	Name(s) of electrode(s) to interpolate.
...	Other parameters passed to the functions.

Methods (by class)

- eeg_data: Interpolate EEG channel(s)

Author(s)

Matt Craddock <matt@mattcraddock.com>

References

- Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. F. (1989). Spherical splines for scalp potential and current density mapping. Electroencephalography and Clinical Neurophysiology, 72, 184-187
- Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. F. (1990). Corrigenda EEG 02274. Electroencephalography and Clinical Neurophysiology, 76, 565

---

is.eeg_epochs *Check if object is of class* eeg_epochs.

Description

Check if object is of class eeg_epochs.

Usage

```
is.eeg_epochs(x)
```

Arguments

x	Object to check.

Author(s)

Matt Craddock <matt@mattcraddock.com>

is.eeg_evoked *Check if object is of class* eeg_evoked

Description

Check if object is of class eeg_evoked

Usage

```
is.eeg_evoked(x)
```

Arguments

x	Object to check.

Author(s)

Matt Craddock <matt@mattcraddock.com>

is.eeg_group *Check if object is of class* eeg_group

Description

Check if object is of class eeg_group

Usage

```
is.eeg_group(x)
```

Arguments

x	Object to check.

is.eeg_ICA *Check if object is of class* eeg_ICA

Description

Check if object is of class eeg_ICA

Usage

```
is.eeg_ICA(x)
```

Arguments

x	Object to check.

Author(s)

Matt Craddock <matt@mattcraddock.com>

is.eeg_stats *Check if object is of class* eeg_stats

Description

Check if object is of class eeg_stats

Usage

```
is.eeg_stats(x)
```

Arguments

x	Object to check.

is.eeg_tfr *Check if object is of class* eeg_tfr

Description

Check if object is of class eeg_tfr

Usage

```
is.eeg_tfr(x)
```

Arguments

x	Object to check.

Author(s)

Matt Craddock <matt@mattcraddock.com>

list_epochs *List epochs*

Description

List trigger types and any labels found in an eeg_epochs object.

Usage

```
list_epochs(data, ...)

## S3 method for class 'eeg_epochs'
list_epochs(data, ...)

## S3 method for class 'eeg_ICA'
list_epochs(data, ...)
```

Arguments

data	An object of class eeg_epochs
...	Additional arguments

Methods (by class)

- eeg_epochs: List epochs and associated events from eeg_epochs objects
- eeg_ICA: List epochs and associated events from eeg_ICA objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

tag_events() and list_events()

Other event handlers: events(), list_events(), tag_events()

---

list_events *List events*

Description

List trigger types and any labels found in an eeg_data object.

Usage

```
list_events(data)
```

Arguments

data	An object of class eeg_data

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

tag_events() and list_epochs()

Other event handlers: events(), list_epochs(), tag_events()

Examples

```
list_events(demo_epochs)
```

plot_butterfly *Create a butterfly plot from timecourse data*

Description

Typically event-related potentials/fields, but could also be timecourses from frequency analyses for single frequencies. Output is a ggplot2 object. CIs not possible.

Usage

```
plot_butterfly(data, ...)

## Default S3 method:
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  colourmap = NULL,
  legend = TRUE,
  continuous = FALSE,
  browse_mode = FALSE,
  allow_facets = FALSE,
  ...
)

## S3 method for class 'eeg_evoked'
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  colourmap = NULL,
  legend = TRUE,
  continuous = FALSE,
  browse_mode = FALSE,
  allow_facets = FALSE,
  ...
)

## S3 method for class 'eeg_data'
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  legend = TRUE,
  allow_facets = FALSE,
  browse_mode = FALSE,
  ...
```

```
)

## S3 method for class 'eeg_epochs'
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  legend = TRUE,
  allow_facets = FALSE,
  browse_mode = FALSE,
  ...
)

## S3 method for class 'eeg_stats'
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  legend = TRUE,
  allow_facets = FALSE,
  browse_mode = FALSE,
  quantity = "statistic",
  ...
)

## S3 method for class 'eeg_lm'
plot_butterfly(
  data,
  time_lim = NULL,
  baseline = NULL,
  legend = TRUE,
  allow_facets = FALSE,
  browse_mode = FALSE,
  quantity = "coefficients",
  ...
)
```

Arguments

data	EEG dataset. Should have multiple timepoints.
...	Other parameters passed to plot_butterfly
time_lim	Character vector. Numbers in whatever time unit is used specifying beginning and end of time-range to plot. e.g. c(-.1,.3)
baseline	Character vector. Times to use as a baseline. Takes the mean over the specified period and subtracts. e.g. c(-.1, 0)
colourmap	Attempt to plot using a different colourmap (from RColorBrewer). (Not yet implemented)

legend	Include plot legend. Defaults to TRUE.
continuous	Is the data continuous or not (I.e. epoched)
browse_mode	Custom theme for use with browse_data.
allow_facets	Allow use of ggplot2 facetting. See note below. Defaults to FALSE.
quantity	Which aspect of the linear model you want to be plotted. only applies to eeg_lm objects

Value

A ggplot object

ggplot2 object showing ERPs for all electrodes overlaid on a single plot.

Methods (by class)

- default: Default plot_butterfly method for data.frames, eeg_data
- eeg_evoked: Plot butterfly for eeg_evoked objects
- eeg_data: Create butterfly plot for eeg_data objects
- eeg_epochs: Create butterfly plot for eeg_epochs objects
- eeg_stats: Create butterfly plot for eeg_stats objects
- eeg_lm: Create butterfly plot for eeg_lm objects

Notes on ggplot2 facetting

In order for ggplot2 facetting to work, the data has to be plotted using stat_summary() rather than geom_line(), so that the plots can still be made when not all categorical variables are reflected in the facets. e.g. if there are two variables with two levels each, but you want to average over one of those variables, stat_summary() is required. However, stat_summary() is extremely slow.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
plot_butterfly(demo_epochs)
plot_butterfly(demo_epochs,
time_lim = c(-.1, .4),
legend = FALSE)
```

plot_difference *Plot ERP difference waves*

Description

Calculates the difference between the event-related potentials from two conditions and plots it.

Usage

```
plot_difference(data, ...)

## S3 method for class 'eeg_epochs'
plot_difference(
  data,
  electrode = NULL,
  time_lim = NULL,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  conditions = "epoch_labels",
  ...
)
```

Arguments

data	eegUtils object. Should have multiple timepoints.
...	Other arguments passed to methods.
electrode	Electrode(s) to plot.
time_lim	Character vector. Numbers in whatever time unit is used specifying beginning and end of time-range to plot. e.g. c(-.1, .3)
baseline	Character vector. Times to use as a baseline. Takes the mean over the specified period and subtracts. e.g. c(-.1,0)
colour	Variable to colour lines by. If no variable is passed, only one line is drawn.
color	Alias for colour.
mapping	A ggplot2 aes() mapping.
conditions	Defaults to "epoch_labels".

Value

Returns a ggplot2 plot object

Methods (by class)

- eeg_epochs: Plot an ERP difference wave from an eeg_epochs object

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
plot_difference(demo_spatial, conditions = "epoch_labels", electrode = "P8")
```

---

plot_electrodes *Plot electrode locations*

Description

Produces either a 2D plot of the electrode locations or an interactive plot of electrode locations in 3D space.

Usage

```
plot_electrodes(data, interact = FALSE)

## Default S3 method:
plot_electrodes(data, interact = FALSE)

## S3 method for class 'eeg_data'
plot_electrodes(data, interact = FALSE)

## S3 method for class 'eeg_tfr'
plot_electrodes(data, interact = FALSE)
```

Arguments

data	Data with associated electrode locations to be plotted.
interact	Choose 2D cartesian layout, or, if set to TRUE, an interactive 3D plot of electrode locations. Defaults to FALSE.

Value

A ggplot or plotly figure showing the locations of the electrodes

Methods (by class)

- default: generic plot electrodes function
- eeg_data: Plot electrodes associated with an eeg_data object.
- eeg_tfr: Plot electrodes associated with an eeg_data object.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
plot_electrodes(demo_epochs)
```

`plot_psd` *Plot Power Spectral Density*

Description

Calculate and plot the PSD for `eeg_*` objects. Output units are dB. The PSD is calculated using Welch's method.

Usage

```
plot_psd(data, freq_range = NULL, ...)

## S3 method for class 'eeg_epochs'
plot_psd(
  data,
  freq_range = NULL,
  n_fft = 256,
  seg_length = NULL,
  noverlap = NULL,
  demean = TRUE,
  keep_trials = TRUE,
  ...
)

## S3 method for class 'eeg_data'
plot_psd(
  data,
  freq_range = NULL,
  n_fft = 2048,
  noverlap = NULL,
  seg_length = NULL,
  ...
)

## S3 method for class 'eeg_ICA'
plot_psd(
  data,
  freq_range = NULL,
  components = NULL,
  seg_length = NULL,
  noverlap = NULL,
  n_fft = 256,
  ...
```

```
)

## S3 method for class 'data.frame'
plot_psd(data, freq_range = NULL, ...)

## S3 method for class 'eeg_evoked'
plot_psd(
  data,
  freq_range = NULL,
  n_fft = 256,
  seg_length = NULL,
  noverlap = NULL,
  keep_trials = TRUE,
  ...
)

## S3 method for class 'eeg_group'
plot_psd(
  data,
  freq_range = NULL,
  n_fft = 256,
  seg_length = NULL,
  noverlap = NULL,
  demean = TRUE,
  ...
)
```

Arguments

data	Object of class eeg_epochs, eeg_data, or eeg_ICA.
freq_range	Vector of lower and upper frequencies to plot. (e.g. c(1, 40))
...	Additional parameters.
n_fft	Number of points to use for the underlying FFTs. Defaults to 256 for eeg_epochs or minimum of 2048 or the signal length for eeg_data.
seg_length	Length of individual segments. Defaults to n_fft. Must be <= n_fft.
noverlap	Amount of overlap between segments, in sampling points. Defaults to 50%.
demean	Remove epoch means before FFT.
keep_trials	Whether to keep trial information in the output or average over all trials
components	Which components to compute the PSD for. Defaults to all.

Details

Welch's method splits the data into multiple segments and then averages over those segments. For epoched data, Welch's FFT is calculated separately for each trial.

Specific parameters such as the number of FFT points and the amount of overlap between segments can be passed to Welch's FFT.

Value

A ggplot object.

Methods (by class)

- eeg_epochs: Plot PSD for eeg_epochs.
- eeg_data: Plot PSD for eeg_data.
- eeg_ICA: Plot PSD for eeg_ICA objects
- data.frame: Plot PSD for data.frames.
- eeg_evoked: Plot PSD for eeg_evoked objects
- eeg_group: Plot PSD for eeg_group objects is not currently supported

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
plot_psd(demo_epochs)
plot_psd(demo_epochs, seg_length = 256)
```

---

plot_tfr *Time-frequency plot*

Description

Creates a time-frequency plot of an eeg_tfr object. The plot has time on the x-axis and frequency on the y-axis. If no electrode is supplied, it will average over all electrodes.

Usage

```
plot_tfr(
  data,
  electrode = NULL,
  time_lim = NULL,
  freq_range = NULL,
  baseline_type = "none",
  baseline = NULL,
  fill_lims = NULL,
  interpolate = FALSE,
  na.rm = TRUE,
  fun.data = mean
)
```

Arguments

data	Object of class eeg_tfr
electrode	Electrode to plot. If none is supplied, averages over all electrodes.
time_lim	Time limits of plot.
freq_range	Vector of two numbers. (e.g. c(8, 40)).
baseline_type	baseline correction to apply. Defaults to "none".
baseline	Baseline period
fill_lims	Custom colour scale (i.e. range of power). e.g. c(-5, 5).
interpolate	Interpolation of raster for smoother plotting.
na.rm	Remove NA values silently (TRUE) or with a warning (FALSE). Defaults to TRUE.
fun.data	Statistical function to use for averaging over electrodes/conditions. Defaults to mean.

Details

Various different baseline options can be applied here (e.g. "db" for decibels, "pc" for percent change, "divide" for division; see rm_baseline for details).

Value

A ggplot

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

rm_baseline()

---

plot_timecourse *Plot 1-D timecourse data.*

Description

Typically event-related potentials/fields, but could also be timecourses from frequency analyses for single frequencies. Averages over all submitted electrodes. Output is a ggplot2 object.

Usage

```
plot_timecourse(data, ...)

## S3 method for class 'data.frame'
plot_timecourse(
  data,
  electrode = NULL,
  time_lim = NULL,
  add_CI = FALSE,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  ...
)

## S3 method for class 'eeg_evoked'
plot_timecourse(
  data,
  electrode = NULL,
  time_lim = NULL,
  add_CI = FALSE,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  ...
)

## S3 method for class 'eeg_ICA'
plot_timecourse(
  data,
  component = NULL,
  time_lim = NULL,
  add_CI = FALSE,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  ...
)

## S3 method for class 'eeg_epochs'
plot_timecourse(
  data,
  electrode = NULL,
  time_lim = NULL,
  add_CI = FALSE,
```

```
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  ...
)

## S3 method for class 'eeg_group'
plot_timecourse(
  data,
  electrode = NULL,
  time_lim = NULL,
  add_CI = FALSE,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  ...
)

## S3 method for class 'eeg_tfr'
plot_timecourse(
  data,
  electrode = NULL,
  time_lim = NULL,
  add_CI = FALSE,
  baseline = NULL,
  colour = NULL,
  color = NULL,
  mapping = NULL,
  freq_range = NULL,
  type = "divide",
  ...
)
```

Arguments

data	EEG dataset. Should have multiple timepoints.
...	Other arguments passed to methods.
electrode	Electrode(s) to plot.
time_lim	Character vector. Numbers in whatever time unit is used specifying beginning and end of time-range to plot. e.g. c(-.1, .3)
add_CI	Add confidence intervals to the graph. Defaults to 95 percent between-subject CIs.
baseline	Character vector. Times to use as a baseline. Takes the mean over the specified period and subtracts. e.g. c(-.1,0)
colour	Variable to colour lines by. If no variable is passed, only one line is drawn.

color	Alias for colour.
mapping	A ggplot2 aes() mapping.
component	name or number of ICA component to plot
freq_range	Choose a specific frequency range to plot
type	Type of baseline correction to use for eeg_tfr objects

Value

Returns a ggplot2 plot object

Methods (by class)

- data.frame: Plot a data.frame timecourse
- eeg_evoked: plot eeg_evoked timecourses
- eeg_ICA: Plot individual components from eeg_ICA components
- eeg_epochs: Plot timecourses from eeg_epochs objects.
- eeg_group: Plot timecourses from eeg_group objects.
- eeg_tfr: Plot timecourses from eeg_tfr objects.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

Examples

```
plot_timecourse(demo_epochs, "A29")
plot_timecourse(demo_epochs, "A29", add_CI = TRUE)
```

---

print.eeg_data *Print* eeg_data *summary*

Description

Print a basic summary of the contents of an eeg_data object

Usage

```
## S3 method for class 'eeg_data'
print(x, ...)
```

Arguments

x	eeg_data object to be printed
...	Further arguments passed

print.eeg_epochs *Print* eeg_epochs *summary*

Description

Print a basic summary of the contents of an eeg_epochs object

Usage

```
## S3 method for class 'eeg_epochs'
print(x, ...)
```

Arguments

x	eeg_epochs object to be printed
...	Further arguments passed

print.eeg_evoked *Print* eeg_evoked *summary*

Description

Print a basic summary of the contents of an eeg_epochs object

Usage

```
## S3 method for class 'eeg_evoked'
print(x, ...)
```

Arguments

x	eeg_epochs object to be printed
...	Further arguments passed

print.eeg_group *Print* eeg_group *summary*

Description

Print a basic summary of the contents of an eeg_group object

Usage

```
## S3 method for class 'eeg_group'
print(x, ...)
```

Arguments

x	eeg_group object to be printed
...	Further arguments passed

print.eeg_ICA *Print* eeg_epochs *summary*

Description

Print a basic summary of the contents of an eeg_epochs object

Usage

```
## S3 method for class 'eeg_ICA'
print(x, ...)
```

Arguments

x	eeg_epochs object to be printed
...	Further arguments passed

print.eeg_lm *Print* eeg_lm *summary*

Description

Print a basic summary of the contents of an eeg_lm object

Usage

```
## S3 method for class 'eeg_lm'
print(x, ...)
```

Arguments

x	eeg_lm object to be printed
...	Further arguments passed

print.eeg_stats *Print* eeg_stats *summary*

Description

Print a basic summary of the contents of an eeg_stats object

Usage

```
## S3 method for class 'eeg_stats'
print(x, ...)
```

Arguments

x	eeg_stats object to be printed
...	Further arguments passed

print.eeg_tfr *Print* eeg_tfr *summary*

Description

Print a basic summary of the contents of an eeg_tfr object

Usage

```
## S3 method for class 'eeg_tfr'
print(x, ...)
```

Arguments

x	eeg_tfr object to be printed
...	Further arguments passed

rm_baseline *Baseline correction*

Description

Used to correct data using the mean of a specified time period. For time-domain data, this will subtract the mean from all data. For eeg_tfr objects, a variety of methods are available, including subtraction, and conversion to "dB" change. With a data frame, it will search for "electrode" and "epoch" columns, and groups on these when found. An electrode column is always required; an epoch column is not. Note that baseline correction is always applied on single-trial basis. For baseline correction based on subtraction, this makes no difference compared to averaging first and then baseline correcting, but for divisive measures used with time-frequency data, this distinction can be very important, and can lead to counterintuitive results.

Usage

```
rm_baseline(data, time_lim = NULL, ...)

## S3 method for class 'eeg_data'
rm_baseline(data, time_lim = NULL, verbose = TRUE, ...)

## S3 method for class 'eeg_epochs'
rm_baseline(data, time_lim = NULL, verbose = TRUE, ...)

## S3 method for class 'data.frame'
rm_baseline(data, time_lim = NULL, verbose = TRUE, ...)

## S3 method for class 'eeg_tfr'
```

```
rm_baseline(data, time_lim = NULL, type = "divide", verbose = TRUE, ...)

## S3 method for class 'eeg_evoked'
rm_baseline(data, time_lim = NULL, verbose = TRUE, ...)
```

Arguments

data	Data to be baseline corrected.
time_lim	Numeric character vector (e.g. time_lim <- c(-.1, 0)) defining the time period to use as a baseline. If the value is NULL, it uses the mean of the whole of each epoch if the data is epoched, or the channel mean if the data is continuous.
...	other parameters to be passed to functions
verbose	Defaults to TRUE. Output descriptive messages to console.
type	Type of baseline correction to apply. Options are ("divide", "ratio", "absolute", "db", and "pc")

Value

An eegUtils object or a data.frame, depending on the input.

Methods (by class)

- eeg_data: remove baseline from continuous eeg_data
- eeg_epochs: Remove baseline from eeg_epochs
- data.frame: Legacy method for data.frames
- eeg_tfr: Method for eeg_tfr objects
- eeg_evoked: Method for eeg_evoked objects

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
rm_baseline(demo_epochs)
rm_baseline(demo_epochs, c(-.1, 0))
```

rotate_angle *Rotate channel locations*

Description

On import, channel locations may be rotated (e.g. Fpz pointing towards ears.)

Usage

```
rotate_angle(chan_info, degrees)
```

Arguments

chan_info	channel information structure
degrees	degrees by which to rotate

Value

A tibble()

Examples

```
plot_electrodes(demo_epochs)
channels(demo_epochs) <- rotate_angle(channels(demo_epochs), 90)
plot_electrodes(demo_epochs)

rotate_angle(channels(demo_epochs), 90)
```

run_ICA *Independent Component Analysis for EEG data*

Description

Performs Independent Component Analysis for electroencephalographic data. Currently only available with on epoched data. Implements three different methods of ICA - 'fastica', 'extended Infomax', and 'Second-Order Blind Identification (SOBI)'. The resulting eeg_ICA objects can be used largely like eeg_epochs objects.

Usage

```
run_ICA(data, ...)

## S3 method for class 'eeg_epochs'
run_ICA(
  data,
  method = "sobi",
  maxit = 1000,
  tol = 1e-06,
  pca = NULL,
  centre = TRUE,
  alg = "gradient",
  rateanneal = c(60, 0.9),
  rate = 0.1,
  verbose = TRUE,
  ...
)
```

Arguments

data	Data to be ICAed.
...	Other parameters passed to function.
method	"sobi" (default), "fastica", "infomax", or "imax". "infomax" uses the implementation from the ica package, whereas imax uses the implementation from the infomax package, which is based on the EEGLAB implementation.
maxit	Maximum number of iterations of the Infomax and Fastica ICA algorithms.
tol	Convergence tolerance for fastica and infomax. Defaults to 1e-06.
pca	Reduce the number of dimensions using PCA before running ICA. Numeric, >1 and < number of channels
centre	Defaults to TRUE. Centre the data on zero by subtracting the column mean. See notes on usage.
alg	Use "gradient descent" or "newton" algorithm for extended infomax. Defaults to "gradient". Ignored if method != "infomax".
rateanneal	Annealing rate for extended infomax. Ignored if method != "infomax".
rate	Learning rate for extended infomax. Ignored if method != "infomax".
verbose	Print informative messages to console.

Value

An eeg_ICA object containing an ICA decomposition

Methods (by class)

- eeg_epochs: Run ICA on an eeg_epochs object

Notes on ICA usage

It is recommended to mean-centre your data appropriately before running ICA. The implementations of FASTICA and extended-Infomax from the `ica` package, and of SOBI ICA have this as an option which is enabled by default, while the implementation of FASTICA in the fICA package enforces mean-centring of the columns of the data. With epoched data, it is recommended to centre each epoch on zero, rather than centre on the overall channel mean. This can be achieved with the `rm_baseline()` function. SOBI ICA will do this automatically, whereas the other ICA implementations will centre on the channel means, not the epoch means.

In addition, PCA will be required if the data is not full rank. This is typical when using average reference, when the data rank will be n_electrodes - 1.

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

Other decompositions: `eeg_decompose()`

Examples

```
sobi_demo <-
  run_ICA(demo_epochs,
          pca = 10)
 sobi_demo
 # We can plot the resulting spatial filters using `topoplot()`
 topoplot(sobi_demo, 1:2)
 ## Not run:  view_ica(sobi_demo)
```

---

select_elecs *Select electrodes from a given dataset*

Description

This is a generic function for selection of electrodes from an EEG dataset.

Usage

```
select_elecs(data, ...)

## Default S3 method:
select_elecs(data, electrode = NULL, keep = TRUE, ...)

## S3 method for class 'eeg_data'
select_elecs(data, electrode, keep = TRUE, df_out = FALSE, ...)

## S3 method for class 'eeg_evoked'
```

```
select_elecs(data, electrode = NULL, keep = TRUE, df_out = FALSE, ...)

## S3 method for class 'eeg_ICA'
select_elecs(data, component, keep = TRUE, df_out = FALSE, ...)

## S3 method for class 'eeg_tfr'
select_elecs(data, electrode, keep = TRUE, df_out = FALSE, ...)
```

Arguments

data	An EEG dataset.
...	Arguments used with related methods
electrode	A character vector of electrode labels for selection or removal.
keep	Defaults to TRUE. Set to false to *remove* the selected electrodes.
df_out	Defaults to FALSE. Set to TRUE to return a dataframe rather than an eeg_data object.
component	Component to select

Value

Data frame with only data from the chosen electrodes

eeg_data object with selected electrodes removed/kept.

Methods (by class)

- default: Select electrodes from a generic data frame.
- eeg_data: Select electrodes from a eeg_data object.
- eeg_evoked: Select electrode from an eeg_evoked object
- eeg_ICA: Select components from eeg_ICA objects.
- eeg_tfr: Select electrodes from eeg_tfr objects.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

See Also

select_times() and select_epochs()

Other Data selection functions: select_times()

Examples

```
names(demo_epochs$signals)
keep_A5 <- select_elecs(demo_epochs, electrode = "A5")
remove_A5 <- select_elecs(demo_epochs, electrode = "A5", keep = FALSE)

select_elecs(demo_epochs, c("A21", "A29"))
```

select_epochs *Select epochs*

Description

This is a generic function for selecting epochs from an epoched data set.

Usage

```
select_epochs(data, ...)

## Default S3 method:
select_epochs(data, ...)

## S3 method for class 'eeg_epochs'
select_epochs(
  data,
  epoch_events = NULL,
  epoch_no = NULL,
  keep = TRUE,
  df_out = FALSE,
  ...
)

## S3 method for class 'eeg_ICA'
select_epochs(
  data,
  epoch_events = NULL,
  epoch_no = NULL,
  keep = TRUE,
  df_out = FALSE,
  ...
)

## S3 method for class 'eeg_tfr'
select_epochs(
  data,
  epoch_events = NULL,
  epoch_no = NULL,
  keep = TRUE,
  df_out = FALSE,
  ...
)
```

Arguments

data	eeg_epochs object from which to select epochs.

...	Parameters passed to specific methods
epoch_events	Select epochs containing any of the specified events. Can be numeric or a character string. Will override any epoch_no input.
epoch_no	Select epochs by epoch number.
keep	Defaults to TRUE, meaning select the specified epochs. Set to FALSE to remove specified epochs.
df_out	Output a data.frame instead of an eeg_data object.

Methods (by class)

- default: Select from generic object
- eeg_epochs: Selection of epochs from eeg_epochs objects.
- eeg_ICA: Selection of epochs from eeg_ICA objects.
- eeg_tfr: Selection of epochs from eeg_tfr objects.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

See Also

select_times() and select_elecs()

Examples

```
select_epochs(demo_epochs, epoch_no = 1:5)
demo_ica <- run_ICA(demo_epochs, pca = 10)
select_epochs(demo_ica, epoch_no = 1:5)
```

---

select_freqs *Select frequencies*

Description

Select specific frequencies from eeg_tfr objects. Can be used to selecting either single frequencies or anything within a range.

Usage

```
select_freqs(data, freq_range)

## S3 method for class 'eeg_tfr'
select_freqs(data, freq_range)
```

Arguments

data	An eeg_tfr object.
freq_range	The range of frequencies to retain. Can be a scale or the lower and upper bounds. (e.g. c(5, 30))

Methods (by class)

- eeg_tfr: Function for selecting specific frequencies from eeg_tfr objects.

Examples

```
demo_tfr <- compute_tfr(demo_epochs, foi = c(4, 30), n_freq = 10, n_cycles = 5)
select_freqs(demo_tfr, c(8, 12))
```

---

select_times *Select timerange*

Description

Generic function for selecting specific time ranges from a given dataset. Input can be a dataframe, or an object of class eeg_data, eeg_epochs, or eeg_evoked. Note this finds the closest times to those specified, so the time range returned may be slightly longer or shorter than that requested.

Usage

```
select_times(data, ...)

## Default S3 method:
select_times(data, time_lim = NULL, ...)

## S3 method for class 'eeg_data'
select_times(data, time_lim = NULL, df_out = FALSE, ...)

## S3 method for class 'eeg_epochs'
select_times(data, time_lim, df_out = FALSE, ...)

## S3 method for class 'eeg_evoked'
select_times(data, time_lim, df_out = FALSE, ...)

## S3 method for class 'eeg_tfr'
select_times(data, time_lim = NULL, df_out = FALSE, ...)
```

Arguments

data	Data from which to select
...	Further arguments passed to or from other methods.
time_lim	A character vector of two numbers indicating the time range to be selected e.g. c(min, max)
df_out	Returns a data frame rather than an object of the same type that was passed in.

Value

Data frame with only data from within the specified range.

eeg_data object

Methods (by class)

- default: Default select times function
- eeg_data: Select times from an eeg_data object
- eeg_epochs: Select times in eeg_epochs objects
- eeg_evoked: Select times in eeg_evoked objects
- eeg_tfr: Select times from an eeg_tfr object

Author(s)

Matt Craddock, <matt@mattcraddock.com>

See Also

select_elecs() and select_epochs()

Other Data selection functions: select_elecs()

Examples

```
## Select timepoints from -.1 to .3
demo_epochs
short_epochs <- select_times(demo_epochs, time_lim = c(-.1, .3))
short_epochs
```

`stat_scalpcontours` *Create an interpolated scalp surface*

Description

`stat_scalpcontours` creates an interpolated surface for an irregular set of x-y coordinates, as is typically required for a topographical EEG plot, and then calculates contours.

Usage

```
stat_scalpcontours(
  mapping = NULL,
  data = NULL,
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  grid_res = 200,
  interp_limit = c("skirt", "head"),
  method = "biharmonic",
  r = NULL,
  bins = 6,
  binwidth = NULL,
  breaks = NULL,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()` or `aes_()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`na.rm`	If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().
grid_res	Resolution of the interpolation grid. (Defaults to 200 points).
interp_limit	Interpolate to the "skirt" or inside the "head".
method	"biharmonic" or "gam" smooth to create interpolated surface
r	Radius of interpolated surface
bins	Number of contour bins. Overridden by binwidth.
binwidth	The width of the contour bins. Overridden by breaks.
breaks	Numeric vector to set the contour breaks. Overrides binwidth and bins. By default, this is a vector of length ten with pretty() breaks.
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

Other topoplot functions: geom_topo(), stat_scalpmap()

---

stat_scalpmap *Create an interpolated scalp surface*

Description

stat_scalpmap creates an interpolated surface for an irregular set of x-y coordinates, as is typically required for a topographical EEG plot. Since the surface should be approximately round, the function attempts to blank out portions of the surface that lay outside the area within the electrodes.

geom_head() adds a headshape to a plot.

geom_mask() adds a masking ring to smooth the edges of a scalp map generated by stat_scalpmap(), to give it a circular appearance.

geom_ears simply draws a pair of ears attached to the head shape.

geom_channels adds either points or text labels at channel locations. This is a convenience function to prevent overplotting when the input data contains many rows of data.

Usage

```
stat_scalpmap(
  mapping = NULL,
  data = NULL,
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  grid_res = 200,
  interpolate = FALSE,
  interp_limit = c("skirt", "head"),
  method = "biharmonic",
  r = NULL,
  ...
)

geom_head(
  mapping = NULL,
  data = NULL,
  show.legend = NA,
  na.rm = TRUE,
  inherit.aes = TRUE,
  interp_limit = "skirt",
  r = 95,
  ...
)

geom_mask(
  mapping = NULL,
  data = NULL,
  show.legend = NA,
  na.rm = FALSE,
  colour = "white",
  size = rel(5),
  r = 95,
  interp_limit = "skirt",
  ...
)

geom_ears(
  mapping = NULL,
  data = NULL,
  show.legend = NA,
  na.rm = FALSE,
  r = 95,
  ...
)
```

```
geom_channels(
  mapping = NULL,
  data = NULL,
  geom = "point",
  show.legend = NA,
  inherit.aes = TRUE,
  na.rm = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().
grid_res	Resolution of the interpolation grid. (Defaults to 200 points).
interpolate	If TRUE interpolate linearly, if FALSE (the default) don't interpolate.
interp_limit	Topoplot with a "skirt" or inside the "head".
method	"biharmonic" or "gam"
r	Radius of head
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
colour	For geom_mask, colour of the masking ring.
size	For geom_mask, width of the masking ring.
geom	"point" for points or "text" for labels. Default is "point".

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

Other topoplot functions: geom_topo(), stat_scalpcontours()

Other topoplot functions: geom_topo(), stat_scalpcontours()

Other topoplot functions: geom_topo(), stat_scalpcontours()

---

tag_epochs *Tag epochs with labels*

Description

Tag epochs with labels indicating details such as experimental condition, based on the occurrence of event triggers from the events() structure. This adds a new column to the epochs structure in an eeg_epochs object.

Usage

```
tag_epochs(.data, ...)

## Default S3 method:
tag_epochs(.data, ...)

## S3 method for class 'eeg_epochs'
tag_epochs(.data, event_type = NULL, event_label = NULL, ...)
```

Arguments

.data	An eegUtils object
...	Additional arguments.
event_type	Label epochs according to specific event_types (typically a trigger)
event_label	Label epochs according to specific event_labels

Methods (by class)

- eeg_epochs: Tag epochs in an eeg_epochs object.

tag_events *Tag events*

Description

Give trigger events meaningful labels. Existing labels will be overwritten. Use hierarchical labelling to tag an event with multiple labels: separate labels with a "/" symbol. (e.g. "cond1" for a trigger that belongs to one condition, "cond1/cond2" for a trigger that could belong to more than one condition).

Usage

```
tag_events(data, ...)

## S3 method for class 'eeg_data'
tag_events(data, trigs, event_label, ...)

## S3 method for class 'eeg_epochs'
tag_events(data, trigs, event_label, ...)
```

Arguments

data	An object of class eeg_data or eeg_epochs
...	Parameters passed to S3 methods
trigs	Character vector of trigger numbers
event_label	Labels for the events.

Methods (by class)

- eeg_data: Tag events in an eeg_data object
- eeg_epochs: Tag events in an epoched dataset

Author(s)

Matt Craddock <matt@mattcraddock.com>

See Also

Other event handlers: events(), list_epochs(), list_events()

topoplot *Topographical Plotting Function for EEG*

Description

Allows topographical plotting of functional data. Output is a ggplot2 object.

Usage

```
topoplot(data, ...)

## Default S3 method:
topoplot(data, ...)

## S3 method for class 'data.frame'
topoplot(
  data,
  time_lim = NULL,
  limits = NULL,
  chanLocs = NULL,
  method = "Biharmonic",
  r = NULL,
  grid_res = 200,
  palette = "RdBu",
  interp_limit = "skirt",
  contour = TRUE,
  chan_marker = "point",
  quantity = "amplitude",
  montage = NULL,
  highlights = NULL,
  scaling = 1,
  groups = NULL,
  verbose = TRUE,
  k = 40,
  ...
)

## S3 method for class 'eeg_data'
topoplot(
  data,
  time_lim = NULL,
  limits = NULL,
  chanLocs = NULL,
  method = "Biharmonic",
  r = NULL,
  grid_res = 200,
  palette = "RdBu",
```

```
  interp_limit = "skirt",
  contour = TRUE,
  chan_marker = "point",
  quantity = "amplitude",
  montage = NULL,
  highlights = NULL,
  scaling = 1,
  verbose = TRUE,
  groups = NULL,
  k = 40,
  ...
)

## S3 method for class 'eeg_epochs'
topoplot(
  data,
  time_lim = NULL,
  limits = NULL,
  chanLocs = NULL,
  method = "Biharmonic",
  r = NULL,
  grid_res = 200,
  palette = "RdBu",
  interp_limit = "skirt",
  contour = TRUE,
  chan_marker = "point",
  quantity = "amplitude",
  montage = NULL,
  highlights = NULL,
  scaling = 1,
  groups = NULL,
  verbose = TRUE,
  k = 40,
  ...
)

## S3 method for class 'eeg_ICA'
topoplot(
  data,
  component,
  time_lim = NULL,
  limits = NULL,
  chanLocs = NULL,
  method = "Biharmonic",
  r = NULL,
  grid_res = 200,
  palette = "RdBu",
  interp_limit = "skirt",
```

```
  contour = TRUE,
  chan_marker = "point",
  quantity = "amplitude",
  montage = NULL,
  highlights = NULL,
  scaling = 1,
  verbose = TRUE,
  groups = NULL,
  k = 40,
  ...
)

## S3 method for class 'eeg_tfr'
topoplot(
  data,
  time_lim = NULL,
  limits = NULL,
  chanLocs = NULL,
  method = "Biharmonic",
  r = NULL,
  grid_res = 200,
  palette = "RdBu",
  interp_limit = "skirt",
  contour = TRUE,
  chan_marker = "point",
  quantity = "power",
  montage = NULL,
  highlights = NULL,
  scaling = 1,
  freq_range = NULL,
  verbose = TRUE,
  groups = NULL,
  k = 40,
  ...
)
```

Arguments

data	An EEG dataset. If the input is a data.frame, then it must have columns x, y, and amplitude at present. x and y are (Cartesian) electrode co-ordinates), amplitude is amplitude.
...	Various arguments passed to specific functions
time_lim	Timepoint(s) to plot. Can be one time or a range to average over. If none is supplied, the function will average across all timepoints in the supplied data.
limits	Limits of the fill scale - should be given as a character vector with two values specifying the start and endpoints e.g. limits = c(-2,-2). Will ignore anything else. Defaults to the range of the data.
chanLocs	Allows passing of channel locations (see `electrode_locations`)

method	Interpolation method. "Biharmonic" or "gam". "Biharmonic" implements the same method used in Matlab's EEGLAB. "gam" fits a Generalized Additive Model with k = 40 knots. Defaults to biharmonic spline interpolation.
r	Radius of cartoon head_shape in mm. Default value is 95 (mm) when using `interp_limit = "skirt"`.
grid_res	Resolution of the interpolated grid. Higher = smoother but slower. Defaults to 200 points per edge.
palette	Defaults to RdBu if none supplied. Can be any palette from RColorBrewer or viridis. If an unsupported palette is specified, switches to Greens.
interp_limit	"skirt" or "head". Defaults to "skirt". "skirt" interpolates just past the farthest electrode and does not respect the boundary of the head_shape. "head" interpolates up to the radius of the plotted head, and moves all electrodes inside the head.
contour	Plot contour lines on topography (defaults to TRUE)
chan_marker	Set marker for electrode locations. "point" = point, "name" = electrode name, "none" = no marker. Defaults to "point".
quantity	Allows plotting of an arbitrary quantitative column. Defaults to amplitude. Use quoted column names. E.g. "p.value", "t_statistic".
montage	Name of an existing montage set. Defaults to NULL; (currently only 'biosemi64alpha' available other than default 10/20 system)
highlights	Electrodes to highlight (in white).
scaling	Scaling multiplication factor for labels and any plot lines. Defaults to 1.
groups	Column name for groups to retain. This is required to create facetted plots.
verbose	Warning messages when electrodes do not have locations. Defaults to TRUE.
k	Degrees of freedom used for spline when using `method = gam`. Defaults to 40.
component	Component to plot (numeric)
freq_range	Range of frequencies to average over.

Methods (by class)

- `default`: Default method for data frames.
- `data.frame`: Topographical plotting of data.frames and other non eeg_data objects.
- `eeg_data`: Topographical plotting of `eeg_data` objects.
- `eeg_epochs`: Topographical plotting of `eeg_epochs` objects.
- `eeg_ICA`: Topographical plot for `eeg_ICA` objects
- `eeg_tfr`: Topographical plotting of `eeg_tfr` objects.

Notes on usage of Generalized Additive Models for interpolation

The function fits a GAM using the `gam` function from `mgcv`. Specifically, it fits a spline using the model function gam(z ~ s(x, y, bs = "ts", k = 40). Using GAMs for smooths is very much experimental. The surface is produced from the predictions of the GAM model fitted to the supplied data. Values at each electrode do not necessarily match actual values in the data: high-frequency variation will tend to be smoothed out. Thus, the method should be used with caution. In addition the method can only be used when there are more than 40 electrodes.

Author(s)

Matt Craddock, <matt@mattcraddock.com>

See Also

Other scalp-based maps: `erp_scalp()`, `interactive_scalp()`

Examples

```
topoplot(demo_epochs)
topoplot(demo_epochs, time_lim = c(.1, .2))
```

---

view_artefacts *Artefact browser*

Description

An interactive Shiny app that allows exploration of channel and epoch statistics.

Usage

```
view_artefacts(data)
```

Arguments

data	Object to be explored.

Value

Nothing.

Author(s)

Matt Craddock <matt@mattcraddock.com>

Examples

```
## Not run:  view_artefacts(demo_epochs)
```

`view_ica` *EEG component viewer*

Description

A Shiny viewer for ICA or SSD/RESS components that provides an interface for looking at topographies, timecourses, and power spectral densities of all or individual components.

Usage

```
view_ica(data)
```

Arguments

`data`	An `eeg_ICA` object

Author(s)

Matt craddock <matt@mattcraddock.com>

Index