Package ‘mixtur’

October 13, 2022

Title  Modelling Continuous Report Visual Short-Term Memory Studies

Version  1.2.0

Date  2021-07-27

Description  A set of utility functions for analysing and modelling data from continuous report short-term memory experiments using either the 2-component mixture model of Zhang and Luck (2008) <doi:10.1038/nature06860> or the 3-component mixture model of Bays et al. (2009) <doi:10.1167/9.10.7>. Users are also able to simulate from these models.

Depends  R (>= 4.0)

Imports  dplyr, ggplot2, rlang, tidyr, RColorBrewer

Suggests  knitr, rmarkdown

License  GPL-3

LazyData  true

URL  https://github.com/JimGrange/mixtur

BugReports  https://github.com/JimGrange/mixtur/issues

Encoding  UTF-8

RoxygenNote  7.1.1

Copyright  Some functions have been adapted from Matlab code written by Paul Bays (https://bayslab.com) published under GNU General Public License.

NeedsCompilation  no

Author  Jim Grange [aut, cre] (<https://orcid.org/0000-0002-8352-8390>), Stuart B. Moore [aut] (<https://orcid.org/0000-0002-0747-9304>), Ed D. J. Berry [ctb]

Maintainer  Jim Grange <grange.jim@gmail.com>

Repository  CRAN

Date/Publication  2021-08-03 08:00:02 UTC
R topics documented:

- bays2009_full
- bays2009_sample
- berry_2019
- fit_mixtur
- get_summary_statistics
- oberauer_2017
- plot_error
- plot_model_fit
- plot_model_parameters
- plot_summary_statistic
- simulate_mixtur

Full data set from Bays et al. (2009)

Description

A full data set including data from 12 participants in a continuous report visual short-term memory experiment. The stimuli were coloured squares in the range radians \(-\pi\) to \(\pi\). The experiment had various set sizes and an additional manipulation of duration of the sample array presentation.

Usage

bays2009_full

Format

A data frame with 7271 rows and 10 variables:

- **id** participant identification
- **set_size** the set size of each trial
- **duration** the duration of the sample array (in milliseconds, ms), with levels 100ms, 500ms, 2000ms
- **response** the participant’s recollection of the target orientation in radians \(-\pi\) to \(\pi\)
- **target** the feature value of the target in radians \(-\pi\) to \(\pi\)
- **non_target_1** the feature value of the first non-target in radians \(-\pi\) to \(\pi\)
- **non_target_2** the feature value of the second non-target in radians \(-\pi\) to \(\pi\)
- **non_target_3** the feature value of the third non-target in radians \(-\pi\) to \(\pi\)
- **non_target_4** the feature value of the fourth non-target in radians \(-\pi\) to \(\pi\)
- **non_target_5** the feature value of the fifth non-target in radians \(-\pi\) to \(\pi\)
### Source

The data set is publicly available on the Open Science Framework, with thanks to Paul Bays:
https://osf.io/c2yx5/

### References


---

### bays2009_sample Sample data set from Bays et al. (2009)

### Description

A sample data set including data from 12 participants in a continuous report visual short-term memory experiment. The stimuli were coloured squares in the range radians -pi to pi. The sample data set only consists of trials with a set size of 4 and a sample array duration of 500ms.

### Usage

bays2009_sample

### Format

A data frame with 7271 rows and 10 variables:

- **id**: participant identification
- **response**: the participant’s recollection of the target orientation in radians (-pi to pi)
- **target**: the feature value of the target in radians (-pi to pi)
- **non_target_1**: the feature value of the first non-target in radians (-pi to pi)
- **non_target_2**: the feature value of the second non-target in radians (-pi to pi)
- **non_target_3**: the feature value of the third non-target in radians (-pi to pi)

### Source

The data set is publicly available on the Open Science Framework, with thanks to Paul Bays:
https://osf.io/c2yx5/

### References

Data set from Berry et al. (2019)

Description
A data set including data from 30 participants in a continuous report visual short-term memory experiment. The stimuli were oriented bars within the range 1-180 degrees. The experiment had a set size of 3.

Usage
berry_2019

Format
A data frame with 3600 rows and 6 variables:

- **id** participant identification
- **condition** condition of experiment: whether the task was completed under single-task or dual-task conditions
- **target_ori** the orientation of the target in degrees (1-180)
- **response_ori** the participant’s recollection of the target orientation in degrees (1-180)
- **non_target_1** the orientation of the first non-target in degrees (1-180)
- **non_target_2** the orientation of the second non-target in degrees (1-180)

Source
The data set is publicly available on the Open Science Framework: https://osf.io/59c4g/

References

Fit the mixture model.

Description
This is the function called by the user to fit either the two- or three- component mixture model.
fit_mixtur

Usage

```r
fit_mixtur(
  data,
  model = "3_component",
  unit = "degrees",
  id_var = "id",
  response_var = "response",
  target_var = "target",
  non_target_var = NULL,
  set_size_var = NULL,
  condition_var = NULL,
  return_fit = FALSE
)
```

Arguments

data: A data frame with columns containing (at the very least) trial-level participant response and target values. This data can either be in degrees (1-360 or 1-180) or radians. If the 3-component mixture model is to be fitted to the data, the data frame also needs to contain the values of all non-targets. In addition, the model can be fit to individual individual participants, individual set-sizes, and individual additional conditions; if the user wishes for this, then the data frame should have columns coding for this information.

model: A string indicating the model to be fit to the data. Currently the options are "2_component", "3_component", "slots", and "slots_averaging".

unit: A string indicating the unit of measurement in the data frame: "degrees" (measurement is in degrees, from 1 to 360); "degrees_180" (measurement is in degrees, but limited to 1 to 180); or "radians" (measurement is in radians, from pi to 2 * pi, but could also be already in the range -pi to pi).

id_var: The quoted column name coding for participant id. If the data is from a single participant (i.e., there is no id column) set to NULL.

response_var: The quoted column name coding for the participants' responses.

target_var: The quoted column name coding for the target value.

non_target_var: The quoted variable name common to all columns (if applicable) storing non-target values. If the user wishes to fit the 3-component mixture model, the user should have one column coding for each non-target’s value in the data frame. If there is more than one non-target, each column name should begin with a common term (e.g., the "non_target" term is common to the non-target columns "non_target_1", "non_target_2" etc.), which should then be passed to the function via the non_target_var variable.

set_size_var: The quoted column name (if applicable) coding for the set size of each response.

condition_var: The quoted column name (if applicable) coding for the condition of each response.

return_fit: If set to TRUE, the function will return the log-likelihood of the model fit, Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), as well as the number of trials used in the fit.
Value

Returns a data frame with best-fitting parameters per participant (if applicable), set-size (if applicable), and condition (if applicable). If `return_fit` was set to `TRUE`, the data frame will also include the log-likelihood value and information criteria of the model fit.

Source

The code for the 3-component model has been adapted from Matlab code written by Paul Bays (https://bayslab.com) published under GNU General Public License.

Examples

```r
# load the example data
data <- bays2009_full

# fit the 3-component mixture model ignoring condition
fit <- fit_mixtur(data = data,
                   model = "3_component",
                   unit = "radians",
                   id_var = "id",
                   response_var = "response",
                   target_var = "target",
                   non_target_var = "non_target",
                   set_size_var = "set_size",
                   condition_var = NULL)
```

---

**get_summary_statistics**

Obtain summary statistics of response error

Description

Returns participant-level summary statistic data of response error estimates ready for inferential analysis. Note that the function does not actually conduct the analysis.

Usage

```r
get_summary_statistics(
  data,
  unit = "degrees",
  id_var = "id",
  response_var = "response",
  target_var = "target",
  set_size_var = NULL,
  condition_var = NULL
)
```
Arguments

- **data**: A data frame with columns containing: participant identifier (declared via variable 'id_var'); the participants’ response per trial ('response_var'); the target value ('target_var'); and, if applicable, the set size of each response ('set_size_var'), and the condition of each response ('condition_var').

- **unit**: The unit of measurement in the data frame: "degrees" (measurement is in degrees, from 0 to 360); "degrees_180" (measurement is in degrees, but limited to 0 to 180); or "radians" (measurement is in radians, from pi to 2 * pi, but could also be already in -pi to pi).

- **id_var**: The quoted column name coding for participant id. If the data is from a single participant (i.e., there is no id column) set to NULL.

- **response_var**: The quoted column name coding for the participants’ responses

- **target_var**: The quoted column name coding for the target value.

- **set_size_var**: The quoted column name (if applicable) coding for the set size of each response.

- **condition_var**: The quoted column name (if applicable) coding for the condition of each response.

Value

Returns a data frame containing the summary statistics `mean_absolute_error`, `resultant_vector_length`, `precision`, and `bias` per participant (if applicable), set-size (if applicable), and condition (if applicable).

Examples

```r
# load an example data frame
data(bays2009_full)

# calculate the summary statistics per condition and per set size
summary_data <- get_summary_statistics(data = bays2009_full,
                                        unit = "radians",
                                        condition_var = "duration",
                                        set_size_var = "set_size")
```

Description

A data set including data from 19 participants in a continuous report visual short-term memory experiment. The stimuli were coloured patches within the range 1-360 degrees. The experiment had a set sizes ranging from 1 to 8.

Usage

oberauer_2017
Format

A data frame with 15,200 rows and 11 variables:

- **id**: participant identification
- **set_size**: the set size of each trial
- **response**: the participant’s recollection of the target colour in degrees (1-360)
- **target**: the orientation of the target colour in degrees (1-360)
- **non_target_1**: the orientation of the first non-target in degrees (1-360)
- **non_target_2**: the orientation of the first non-target in degrees (1-360)
- **non_target_3**: the orientation of the second non-target in degrees (1-360)
- **non_target_4**: the orientation of the third non-target in degrees (1-360)
- **non_target_5**: the orientation of the fourth non-target in degrees (1-360)
- **non_target_6**: the orientation of the fifth non-target in degrees (1-360)
- **non_target_7**: the orientation of the sixth non-target in degrees (1-360)

Source

The data set is publicly available on the Open Science Framework: https://osf.io/j24wb/

References


---

**plot_error**

*Plot response error of behavioural data relative to target values.*

Description

Function to plot the response error in behavioural data relative to target values. Requires a data frame that (at least) has target value data and participant response data.

Usage

```r
plot_error(data, 
  unit = "degrees", 
  id_var = "id", 
  response_var = "response", 
  target_var = "target", 
  set_size_var = NULL, 
  condition_var = NULL, 
  n_bins = 18, 
  n_col = 2, 
  return_data = FALSE, 
  palette = "Dark2"
)
```
plot_error

Arguments

data A data frame with columns containing: participant identifier ("id_var"); the participants’ response per trial ("response_var"); the target value ("target_var"); and, if applicable, the set size of each response ("set_size_var"), and the condition of each response ("condition_var").

unit The unit of measurement in the data frame: "degrees" (measurement is in degrees, from 0 to 360); "degrees_180" (measurement is in degrees, but limited to 0 to 180); or "radians" (measurement is in radians, from pi to 2 * pi, but could also be already in -pi to pi).

id_var The column name coding for participant id. If the data is from a single participant (i.e., there is no id column) set to "NULL".

response_var The column name coding for the participants’ responses.

target_var The column name coding for the target value.

set_size_var The column name (if applicable) coding for the set size of each response.

condition_var The column name (if applicable) coding for the condition of each response.

n_bins An integer controlling the number of cells / bins used in the plot.

n_col An integer controlling the number of columns in the resulting plot.

return_data A boolean (TRUE or FALSE) indicating whether the data for the plot should be returned.

palette A character stating the preferred colour palette to use. To see all available palettes, type display.brewer.all() into the console.

Value

If return_data is set to FALSE (which it is by default), the function returns a ggplot2 object visualising the density distribution of response error averaged across participants (if applicable) per set-size (if applicable) and condition (if applicable).

If return_data is set to TRUE, the function returns a list with two components:

* plot: The ggplot2 object.
* data: A data frame with the data used to generate the plot.

Examples

plot_error(bays2009_full,
           unit = "radians",
           set_size_var = "set_size")
**plot_model_fit**  
*Plot model fit against human error data (target errors)*

**Description**

Plot model fit against human error data (target errors)

**Usage**

```r
plot_model_fit(
  participant_data,  
  model_fit, 
  model, 
  unit = "degrees", 
  id_var = "id", 
  response_var = "response", 
  target_var = "target", 
  set_size_var = NULL, 
  condition_var = NULL, 
  n_bins = 18, 
  n_col = 2,  
  palette = "Dark2"
)
```

**Arguments**

- **participant_data**  
  A data frame of the participant data, with columns containing: participant identifier (`id_var`); the participants' response per trial (`response_var`); the target value (`target_var`); and, if applicable, the set size of each response (`set_size_var`), and the condition of each response (`condition_var`).

- **model_fit**  
  The model fit object to be plotted against participant data.

- **model**  
  A string indicating the model that was fit to the data. Currently the options are "2_component", "3_component", "slots", and "slots_averaging".

- **unit**  
  The unit of measurement in the data frame: "degrees" (measurement is in degrees, from 0 to 360); "degrees_180" (measurement is in degrees, but limited to 0 to 180); or "radians" (measurement is in radians, from pi to 2 * pi, but could also be already in -pi to pi).

- **id_var**  
  The column name coding for participant id. If the data is from a single participant (i.e., there is no id column) set to "NULL".

- **response_var**  
  The column name coding for the participants' responses

- **target_var**  
  The column name coding for the target value

- **set_size_var**  
  The column name (if applicable) coding for the set size of each response

- **condition_var**  
  The column name (if applicable) coding for the condition of each response
plot_model_parameters

n_bins An integer controlling the number of cells / bins used in the plot of the behavioural data.
n_col An integer controlling the number of columns in the resulting plot.
palette A character stating the preferred colour palette to use. To see all available palettes, type display.brewer.all() into the console.

Value

The function returns a ggplot2 object visualising the mean observed response error density distribution across participants (if applicable) per set-size (if applicable) and condition (if applicable) together with the model predictions superimposed.

plot_model_parameters  Plot best-fitting parameters of model fit

Description

Function to plot the best-fitting parameters of either the 2-component or 3-component model.

Usage

plot_model_parameters(
  model_fit,
  model,
  id_var = "id",
  set_size_var = NULL,
  condition_var = NULL,
  n_col = 2,
  return_data = FALSE,
  palette = "Dark2"
)

Arguments

model_fit The model fit object containing the parameters to be plotted.
model A string indicating the model that was fit to the data. Currently the options are "2_component", "3_component", "slots", and "slots_averaging".
id_var The column name coding for participant id.
set_size_var The column name (if applicable) coding for the set size of each response.
condition_var The column name (if applicable) coding for the condition of each response.
n_col An integer controlling the number of columns in the resulting plot.
return_data A boolean (TRUE or FALSE) indicating whether the data for the plot should be returned.
palette A character stating the preferred colour palette to use. To see all available palettes, type display.brewer.all() into the console.
plot_summary_statistic

Plot summary statistics of behavioural data

Description

Function to plot model-free summary statistics of behavioural data. Users can plot mean absolute error, resultant vector length, and precision of the behavioural data.

Usage

plot_summary_statistic(
  data,
  statistic = "precision",
  unit = "degrees",
  id_var = "id",
  response_var = "response",
  target_var = "target",
  set_size_var = NULL,
  condition_var = NULL,
  return_data = FALSE,
  palette = "Dark2"
)

Arguments

data A data frame with columns containing: participant identifier ("id_var"); the participants' response per trial ("response_var"); the target value ("target_var"); and, if applicable, the set size of each response ("set_size_var"), and the condition of each response ("condition_var").

statistic The summary statistic to plot. This can be set to "mean_absolute_error", "resultant_vector_length", or "precision".

unit The unit of measurement in the data frame: "degrees" (measurement is in degrees, from 0 to 360); "degrees_180" (measurement is in degrees, but limited to 0 to 180); or "radians" (measurement is in radians, from pi to 2 * pi, but could also be already in -pi to pi).
id_var The column name coding for participant id. If the data is from a single participant (i.e., there is no id column) set to "NULL".
response_var The column name coding for the participants’ responses.
target_var The column name coding for the target value.
set_size_var The column name (if applicable) coding for the set size of each response.
condition_var The column name (if applicable) coding for the condition of each response.
return_data A boolean (TRUE or FALSE) indicating whether the data for the plot should be returned.
palette A character stating the preferred colour palette to use. To see all available palettes, type display.brewer.all() into the console.

Value

If return_data is set to FALSE (which it is by default), the function returns a ggplot2 object visualising the summary statistic averaged across participants (if applicable) per set-size (if applicable) and condition (if applicable).

If return_data is set to TRUE, the function returns a list with two components:

• plot: The ggplot2 object.
• data: A data frame with the data used to generate the plot.

Examples

plot_summary_statistic(bays2009_full,
unit = "radians",
statistic = "precision",
set_size_var = "set_size",
condition_var = "duration")

simulate_mixtur Generate simulated data from mixture models

Description

Generate simulated data from mixture models

Usage

simulate_mixtur(n_trials, model, kappa, p_u, p_n, K, set_size)
**simulate_mixtur**

**Arguments**

- `n_trials` an integer indicating how many trials to simulate
- `model` a string indicating the model to fit to the data. Currently the options are "2_component", "3_component", "slots", and "slots_averaging".
- `kappa` a numeric value indicating the concentration parameter of the von Mises distribution to use in the simulations. Note, when simulating from the 2_component or 3_component model, if multiple values are provided to the set_size argument, kappa must be a vector of parameter values to use for each set size).
- `p_u` a numeric value indicating the probability of uniform guessing to use when simulating from the 2_component and 3_component models. Note, when simulating from the 2_component or 3_component model, if multiple values are provided to the set_size argument, p_u must be a vector of parameter values to use for each set size).
- `p_n` a numeric value indicating the probability of a non-target response when simulating from the 3_component model. Note, when simulating from the 2_component or 3_component model, if multiple values are provided to the set_size argument, p_n must be a vector of parameter values to use for each set size).
- `K` a numeric value indicating the capacity value to use when simulating from the slots and slots_averaging models.
- `set_size` a numeric value (or vector) indicating the set size(s) to use in the simulations.

**Value**

Returns a data frame containing simulated responses from the requested model per set-size (if applicable).

**Examples**

```r
# simulate from the slots model
slots_data <- simulate_mixtur(n_trials = 1000, 
model = "slots",
kappa = 8.2, 
K = 2.5, 
set_size = c(2, 4, 6, 8))

# simulate one set size from the 3_component model
component_data <- simulate_mixtur(n_trials = 1000, 
model = "3_component",
kappa = 8.2, 
p_u = .1, 
p_n = .15, 
set_size = 4)
```

# simulate multiple set sizes from the 3_component model

```r
component_data_multiple_sets <- simulate_mixtur(n_trials = 1000, 
    model = "3_component", 
    kappa = c(10, 8, 6), 
    p_u = c(.1, .1, .1), 
    p_n = c(.1, .15, .2), 
    set_size = c(2, 4, 6))
```
Index

* datasets
  bays2009_full, 2
  bays2009_sample, 3
  berry_2019, 4
  oberauer_2017, 7

  bays2009_full, 2
  bays2009_sample, 3
  berry_2019, 4
  fit_mixtur, 4

  get_summary_statistics, 6

  oberauer_2017, 7

  plot_error, 8
  plot_model_fit, 10
  plot_model_parameters, 11
  plot_summary_statistic, 12

  simulate_mixtur, 13