

# Package ‘biasdetection’

October 20, 2020

**Title** Bias Detection for Stated Preference Experiments

**Version** 0.0.0.9000

**Description** Identifies the different types of biases in a stated preference dataset.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, stringr, rlist

**RoxygenNote** 7.1.1

**URL** <https://github.com/Rupanjana22/biasdetection>

**BugReports** <https://github.com/Rupanjana22/biasdetection/issues>

**Roxygen** list(markdown = TRUE)

**NeedsCompilation** no

**Author** Rupanjana Chakraborty [aut, cre]

**Maintainer** Rupanjana Chakraborty <rupanjana.chakraborty@tum.de>

## R topics documented:

attribute_non_attendance_function . . . . .	2
create_plots_ers . . . . .	2
create_plots_mrs . . . . .	3
create_plots_rr . . . . .	3
ers_function . . . . .	4
get_time_groups . . . . .	4
lrs_general_function . . . . .	5
mrs_function . . . . .	5
random_responding_time_filter . . . . .	6
remove_duplicates . . . . .	6
rr_function . . . . .	7
rr_time_filter . . . . .	7
<b>Index</b>	<b>8</b>

---

attribute\_non\_attendance\_function

*Attribute Non Attendance Function (for Inconsistent Bias and Non Trading)*

---

### Description

Attribute Non Attendance Function (for Inconsistent Bias and Non Trading)

### Usage

```
attribute_non_attendance_function(
  csv_file,
  total_scenarios,
  scenarios,
  alternatives,
  attribute,
  attribute_short
)
```

### Arguments

csv_file	The file we want to process
total_scenarios	A list containing all the SP scenarios
scenarios	A list containing the relevant SP scenarios for the specific attribute of LRS
alternatives	A list containing the alternatives for the SP experiment
attribute	The attribute chosen for LRS
attribute_short	The short form of the attribute name (for table names)

### Value

A list containing the updated data attribute\_non\_attendance\_function()

---

create\_plots\_ers

*Create Plots Extreme Response Style*

---

### Description

This function creates plots from the resulting data in extreme response style (question wise bar plot and respondent wise bar plot (for positive, negative and total), kernel density plot for respondent wise).

### Usage

```
create_plots_ers(ers_data, likert_columns)
```

**Arguments**

ers\_data            Resulting data from the ers\_function  
 likert\_columns    A list containing the column names to use for ERS create\_plots\_ers()

---

create_plots_mrs	<i>Create Plots Mid point Response Style</i>
------------------	--

---

**Description**

This function creates plots from the resulting data in mid point response style (question and respondent wise bar plots).

**Usage**

```
create_plots_mrs(mrs_data, likert_columns)
```

**Arguments**

mrs\_data            Resulting data from the mrs\_function  
 likert\_columns    A list containing the column names to use for MRS create\_plot\_mrs()

---

create_plots_rr	<i>Create Plots Random Responding</i>
-----------------	---------------------------------------

---

**Description**

This function creates plots from the resulting data in random responding (bar plot for group wise and bar plot, density plot and histogram for respondent wise)

**Usage**

```
create_plots_rr(rr_data, excluded_time_groups, num_time_groups)
```

**Arguments**

rr\_data            A list containing unique responses (based on id) from the original panel data  
 excluded\_time\_groups    A list containing the indices of excluded time groups  
 num\_time\_groups    Number of time groups create\_plots\_rr()

---

ers_function	<i>Extreme Responding Style</i>
--------------	---------------------------------

---

**Description**

Extreme Responding Style

**Usage**

```
ers_function(csv_file, likert_columns, max_value, min_value)
```

**Arguments**

csv_file	The file we want to process
likert_columns	A list containing the column names to use for ERS
max_value	Maximum rate in the likert scale
min_value	Minimum rate in the likert scale

**Value**

A list containing the updated data ers\_function()

---

get_time_groups	<i>Get number of time groups</i>
-----------------	----------------------------------

---

**Description**

This function reads the .csv file which is originated from Lime Survey and calculates the number of time groups automatically for detecting random responses based on the time spent in filling out parts of the survey.

**Usage**

```
get_time_groups(csv_file)
```

**Arguments**

csv_file	The .csv file we want to process
----------	----------------------------------

**Value**

Number of time groups get\_time\_groups()

---

lrs\_general\_function    *Lexicographic Response Style Function*


---

**Description**

Lexicographic Response Style Function

**Usage**

```
lrs_general_function(
  csv_file,
  total_scenarios,
  scenarios,
  alternatives,
  attribute,
  attribute_short
)
```

**Arguments**

csv_file	The file we want to process
total_scenarios	A list containing all the SP scenarios
scenarios	A list containing the relevant SP scenarios for the specific attribute of LRS
alternatives	A list containing the alternatives for the SP experiment
attribute	The attribute chosen for LRS
attribute_short	The short form of the attribute name (for table names)

**Value**

A list containing the updated data lrs\_general\_function()

---

mrs\_function    *Mid point Response Styles*


---

**Description**

Mid point Response Styles

**Usage**

```
mrs_function(csv_file, likert_columns, mid_value)
```

**Arguments**

csv_file	The file we want to process
likert_columns	A list containing the column names to use for MRS
mid_value	Mid rate in the likert scale

**Value**

A list containing the updated data `mrs_function()`

---

`random_responding_time_filter`

*Random responding time filter*

---

**Description**

For each Time Group and for each respondent, the results of the random responses are recorded as binary. It also creates 2 additional columns (`RR_Total` and `percent_RR_Total`) containing the sum of the random responses for each Time Group, and the total percentage of random responses for each respondent, respectively.

**Usage**

```
random_responding_time_filter(rr_data, excluded_time_groups, num_time_groups)
```

**Arguments**

`rr_data` A list containing unique responses (based on id) from the original panel data

`excluded_time_groups`

A list containing the indices of excluded time groups

`num_time_groups`

Number of time groups

**Value**

A list containing the updated data `random_responding_time_filter()`

---

`remove_duplicates`

*Remove duplicate*

---

**Description**

Removes the duplicate entries from the panel data based on the id of the respondent to detect random responses.

**Usage**

```
remove_duplicates(csv_data)
```

**Arguments**

`csv_data` A list containing the responses

**Value**

A list containing unique responses (based on id) from the original panel data `remove_duplicates()`

---

rr_function	<i>Write the updated .csv file</i>
-------------	------------------------------------

---

**Description**

Given a .csv file with panel data, excludes duplicate entries, applies time filter for random responses and writes the result in another .csv file.

**Usage**

```
rr_function(csv_file, excluded_time_groups, num_time_groups)
```

**Arguments**

csv_file	The file we want to process
excluded_time_groups	A list containing the indices of excluded time groups
num_time_groups	Number of time groups

**Value**

A list containing the updated data rr\_function()

---

rr_time_filter	<i>Random response time filter</i>
----------------	------------------------------------

---

**Description**

Calculates the first quartiles of each Time Group to identify the random responses. It displays them as well.

**Usage**

```
rr_time_filter(rr_data, num_time_groups)
```

**Arguments**

rr_data	A list containing unique responses (based on id) from the original panel data
num_time_groups	Number of time groups rr_time_filter()

# Index

`attribute_non_attendance_function`, [2](#)

`create_plots_ers`, [2](#)

`create_plots_mrs`, [3](#)

`create_plots_rr`, [3](#)

`ers_function`, [4](#)

`get_time_groups`, [4](#)

`lrs_general_function`, [5](#)

`mrs_function`, [5](#)

`random_responding_time_filter`, [6](#)

`remove_duplicates`, [6](#)

`rr_function`, [7](#)

`rr_time_filter`, [7](#)