

# Package ‘multifear’

September 23, 2023

**Type** Package

**Title** Multiverse Analyses for Conditioning Data

**Version** 0.1.3

**Maintainer** Angelos-Miltiadis Kryptos <amkryptos@gmail.com>

**Description** A suite of functions for performing analyses, based on a multiverse approach, for conditioning data. Specifically, given the appropriate data, the functions are able to perform t-tests, analyses of variance, and mixed models for the provided data and return summary statistics and plots. The function is also able to return for all those tests p-values, confidence intervals, and Bayes factors. The methods are described in Lonsdorf, Gerlicher, Klingelhofer-Jens, & Kryptos (2022) <[doi:10.1016/j.brat.2022.104072](https://doi.org/10.1016/j.brat.2022.104072)>.

**Depends** R (>= 3.6.3)

**License** GPL-3

**LazyData** true

**Encoding** UTF-8

**URL** <https://github.com/AngelosPsy/multifear>

**BugReports** <https://github.com/AngelosPsy/multifear/issues>

**Imports** dplyr (>= 0.8.4), purrr (>= 0.3.3), stats (>= 3.6.2), ez (>= 4.4.0), stringr (>= 1.4.0), reshape2 (>= 1.4.3), tibble (>= 2.1.3), ggplot2 (>= 3.2.1), effsize (>= 0.7.8), nlme (>= 3.1.144), BayesFactor (>= 0.9.12.4.2), bayestestR (>= 0.10.0), broom (>= 0.5.5), effectsize (>= 0.4.1), esc (>= 0.5.1), forestplot (>= 1.10), bootstrap (>= 2019.6)

**RoxygenNote** 7.2.3

**Suggests** gridExtra (>= 2.3), fastDummies (>= 1.6.1), vctrs (>= 0.3.1), tidyselect (>= 1.0.0), tidyr (>= 1.0.2), plyr (>= 1.8.6), ggraph (>= 2.0.1), testthat (>= 2.1.0), cowplot (>= 1.0.0), covr, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Angelos-Miltiadis Kryptos [aut, cre, cph]

**Repository** CRAN

**Date/Publication** 2023-09-23 19:50:02 UTC

**R topics documented:**

<code>bt_test_mf</code> . . . . .	2
<code>chop_cs</code> . . . . .	4
<code>chop_css</code> . . . . .	5
<code>combine_cs</code> . . . . .	6
<code>example_data</code> . . . . .	6
<code>exclusion_criteria</code> . . . . .	7
<code>forestplot_mf</code> . . . . .	8
<code>inference_cs</code> . . . . .	9
<code>inference_plot</code> . . . . .	9
<code>mixed_mf</code> . . . . .	10
<code>multiverse_cs</code> . . . . .	13
<code>rm_anova_mf</code> . . . . .	14
<code>rm_banova_mf</code> . . . . .	16
<code>t_test_mf</code> . . . . .	18
<code>universe_cs</code> . . . . .	20

<b>Index</b>	<b>23</b>
--------------	-----------

---

`bt_test_mf`

*bt\_test\_mf*

---

**Description**

Basic function for running the Bayesian t-tests included in the main analyses

**Usage**

```
bt_test_mf(
  cs1,
  cs2,
  data,
  subj,
  group = NULL,
  na.rm = FALSE,
  paired = TRUE,
  rscale = "medium",
  phase = "acquisition",
  dv = "scr",
  exclusion = "full data",
  cut_off = "full data"
)
```

**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
group	the name of the group, if included, default to NULL
na.rm	Whether NAs should be removed, default to FALSE
paired	Whether the t-test refers to dependent (i.e., paired) or to independent sample(s). Default to TRUE
rscale	r scale to be used in the prior of the alternative hypothesis, default to "medium".
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off Name of the cut_off applied. Default to full data

**Details**

This is a wrapper function function around the `BayesFactor::ttestBF()`, running multiple Bayesian t-tests. Similar to the `t_test_mf` function, the function will run different t-tests based on the phase that the t-tests refer to. So, in case of the acquisition phase, there will be a t-test of differences and positive differences, whereas for the extinction phase a t-test for differences and negative differences.

**Value**

A tibble with the following column names:

x: the name of the independent variable (e.g., cs)

y: the name of the dependent variable as this defined in the dv argument

exclusion: see exclusion argument

model: the model that was run (e.g., t-test)

controls: ignore this column for this test

method: the model that was run

p.value: irrelevant here

effect.size: irrelevant here

effect.size.ma: irrelevant here

estimate: the estimate of the test run

statistic: the t-value

conf.low: the lower confidence interval for the estimate  
 conf.high: the higher confidence interval for the estimate  
 framework: were the data analysed within a NHST or Bayesian framework?  
 data\_used: a list with the data used for the specific test

### Examples

```
# Load example data
data(example_data)

# Paired sample t-tests
bt_test_mf(cs1 = "CSP1", cs2 = "CSM1", subj = "id", data = example_data)

# Independent sample t-tests
bt_test_mf(cs1 = "CSP1", cs2 = "CSM1", subj = "id", group = "group", data = example_data)
```

---

chop_cs	<i>chop_cs</i>
---------	----------------

---

### Description

Function for separating the conditioned responses into multiple pieces

### Usage

```
chop_cs(cs, data, subj, group = NULL, prefix = "cs", na.rm = FALSE)
```

### Arguments

cs	The column name(s) of the conditioned responses
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
group	default to NULL
prefix	prefix to be included in the end data frame
na.rm	Whether NAs should be removed, default to FALSE

### Value

A tibble with the following column names: "id" = id column; "top10per" = mean of top 10 "bottom10per" = mean of bottom 10 "minfirst" = all trials minus the first one; "all" = mean of all trials; "t33per" = mean of top 33 "m33per" = mean of medium 33 "b33per" = mean of bottom 33 "fhalf" = mean of first half of the trials; "lhalf" = mean of last half of the trials; "ftrial" = first trial; "ltrial" = last trial; "t20per" = mean of top 20 "b20per" = mean of bottom 20 "f2trial" = mean of first two trials; "l2trial" = mean of last two trials; "per2\_X" = mean per two trials (i.e., block), with X referring to the number of block.

**Examples**

```
data(example_data)
chop_cs(cs = paste0("CSP", 1:10), data = example_data, subj = "id")
```

---

chop\_css

*chop\_css*


---

**Description**

Function for separating the conditioned responses into multiple pieces for two CSs.

**Usage**

```
chop_css(cs1, cs2, data, subj, cs_paired = NULL, group = NULL, na.rm = FALSE)
```

**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
cs_paired	A character vector with the trials that were paired. Default is set to NULL, suggesting that there was full reinforcement @return A tibble with the following column names – separately for cs1 and cs2: "id" = id column; "top10per" = mean of top 10 "bottom10per" = mean of bottom 10 "minfirst" = all trials minus the first one; "all" = mean of all trials; "t33per" = mean of top 33 "m33per" = mean of medium 33 "b33per" = mean of bottom 33 "fhalf" = mean of first half of the trials; "lhalf" = mean of last half of the trials; "ftrial" = first trial; "ltrial" = last trial; "t20per" = mean of top 20 "b20per" = mean of bottom 20 "f2trial" = mean of first two trials; "l2trial" = mean of last two trials; "per2_X" = mean per two trials (i.e., block), with X referring to the number of block.
group	the name of the group, if included, default to NULL
na.rm	Whether NAs should be removed, default to FALSE

---

combine_cs	<i>combine_cs</i>
------------	-------------------

---

### Description

Function for computing mean responses across CSs

### Usage

```
combine_cs(cs1, cs2, data, na.rm = FALSE)
```

### Arguments

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
na.rm	Whether NAs should be removed, default to FALSE

### Value

A tibble with the initial data frame (given by the data argument) together with an additional column with the means for the columns defined in the cs1 and cs2 arguments.

---

example_data	<i>Simulated data sets of skin conductance responses</i>
--------------	--

---

### Description

A simulated data set containing conditioned skin conductance responses for 10 conditioned stimulus plus trials and a equal number of conditioned stimulus minus trials The variables are as follows:

### Usage

```
example_data
```

**Format**

A data frame with 40 rows and 22 variables:

**CSP1** Conditioned stimulus plus, trial 1  
**CSP2** Conditioned stimulus plus, trial 2  
**CSP3** Conditioned stimulus plus, trial 3  
**CSP4** Conditioned stimulus plus, trial 4  
**CSP5** Conditioned stimulus plus, trial 5  
**CSP6** Conditioned stimulus plus, trial 6  
**CSP7** Conditioned stimulus plus, trial 7  
**CSP8** Conditioned stimulus plus, trial 8  
**CSP9** Conditioned stimulus plus, trial 9  
**CSP10** Conditioned stimulus plus, trial 10  
**CSM1** Conditioned stimulus minus, trial 1  
**CSM2** Conditioned stimulus minus, trial 2  
**CSM3** Conditioned stimulus minus, trial 3  
**CSM4** Conditioned stimulus minus, trial 4  
**CSM5** Conditioned stimulus minus, trial 5  
**CSM6** Conditioned stimulus minus, trial 6  
**CSM7** Conditioned stimulus minus, trial 7  
**CSM8** Conditioned stimulus minus, trial 8  
**CSM9** Conditioned stimulus minus, trial 9  
**CSM10** Conditioned stimulus minus, trial 10  
**id** Participant number  
**group** Group assignment

---

exclusion\_criteria      *exclusion\_criteria*

---

**Description**

Exclusion criteria

**Usage**

```
exclusion_criteria(data, cutoff = 0)
```

**Arguments**

**data**                    a data object generated by the `multics::chop_css` function  
**cutoff**                    cut off score

**Details**

Here the different exclusion criteria are applied to the provided data.

**Value**

A tibble with a brace of smaller tibbles, with each sub-tibble including a data frame after each one of the exclusion criteria – mentioned in the `multifear::chop_css` function – is applied.

---

<code>forestplot_mf</code>	<i>forestplot_mf</i>
----------------------------	----------------------

---

**Description**

Basic function for forest plot

**Usage**

```
forestplot_mf(
  data,
  ci = TRUE,
  include_label_text = TRUE,
  reported_effect = "common",
  ...
)
```

**Arguments**

<code>data</code>	a <code>universe_mf</code> or <code>multiverse_mf</code> object
<code>ci</code>	should confidence intervals be included – default to TRUE
<code>include_label_text</code>	Whether the labels for each effect should be include. Default to TRUE
<code>reported_effect</code>	Option as to what effect size to report. The options are: "common" (default option), "anova", or "ttest".
<code>...</code>	any additional argument

**Details**

This is a wrapper around the `forestplot::forestplot` function. The function only uses the ANOVAs and the t-tests. For the t-tests though it includes only the two-sided, as they are the same with the one-sided ones and having both would probably give a false picture of the effect.

The `include_label_text` can be used when multiple plots need to be combined side by side, so one of them does not include the labels.

**Value**

A plot #'



---

inference_cs	<i>inference_cs</i>
--------------	---------------------

---

**Description**

Function for summarizing the multiverse results.

**Usage**

```
inference_cs(data, alpha_level = 0.05, na.rm = FALSE, framework = "Both")
```

**Arguments**

data	a data frame with the results of a multiverse analyses.
alpha_level	What should be the alpha level used (default to 0.05).
na.rm	Should NA's be removed (default to FALSE). See details.
framework	Inference framework. Values could be "NHST", "Bayesian", or "Both" (no case sensitivity)

**Details**

For now the function returns mean, median, standard deviations of p values and proportion of p values below a criterion defined by the `alpha_level` parameter (default to 0.05) as well as mean Bayes factors (please see the 'framework' argument). The user may choose to drop the NAs for the summary statistic.

**Value**

A data frame with summaries of the results.

---

inference_plot	<i>inference_plot</i>
----------------	-----------------------

---

**Description**

Function for plotting the multiverse results.

**Usage**

```
inference_plot(
  data,
  alpha_level = 0.05,
  add_line = TRUE,
  na.rm = FALSE,
  framework = "Both",
  col = c("gray45", "maroon4", "brown1"),
  return_plot = TRUE
)
```

**Arguments**

<code>data</code>	a data frame with the results of a multiverse analysis
<code>alpha_level</code>	What should be the alpha level used (default to 0.05)
<code>add_line</code>	Whether to add a line with the alpha level in the produced histogram (default to TRUE)
<code>na.rm</code>	Should NA's be removed (default to FALSE). See details for more information
<code>framework</code>	Inference framework. Values could be "NHST", "Bayesian", or "Both" (no case sensitivity)
<code>col</code>	A length three vector with the colors to be used for ANOVAS, t-tests, and mixed models (in this order)
<code>return_plot</code>	Whether to return a plot or not (default too TRUE)

**Details**

For the plot the NAs in the `p.value` column are removed automatically – so what `ggplot2` does automatically but here no message is returned.

The `return_plot` argument is there in case you want to combine multiple panels and you do not want to have a plot returned every time you run the code.

**Value**

A histogram summarizing the results.

---

mixed\_mf

*mixed\_mf*

---

**Description**

Basic function for running mixed models for the multiverse analysis

**Usage**

```

mixed_mf(
  cs1,
  cs2,
  data,
  subj,
  group = NULL,
  phase = "acquisition",
  dv = "scr",
  exclusion = "full data",
  cut_off = "full data"
)

```

**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
group	the name of the group, if included, default to NULL
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off Name of the cut_off applied. Default to full data

**Details**

The function assumes that you include more than 1 trial per CS. The function returns an error if that is not the function.

The function performs by default two dependent variable standardizations, the one per subject and the other one without taking subject into account.

In case time is included, the function computes the intercept – i.e., the 0 point – on the middle of the time sequence.

The following models are run and compared: a) Intercept only model, b) Intercept plus CS model, and c) Intercept plus CS x Time interaction.

Separate models are run with ‘Subject’ as random factor, as well as ‘Subject and Time’ as random factors.

The model is fit by maximizing the log-likelihood (i.e., "ML" term in nlme::lme).

The model comparison is done using ‘BIC’.

**Value**

A data frame with the results.

The data frame returned is the standard one returned in all function in the package. Specifically we have:

A tibble with the following column names:

x: the name of the independent variable (e.g., cs). There, you can see the term of the model that is returned. So, not the full model is returned but only this particular term.

y: the name of the dependent variable as this defined in the dv argument

exclusion: see exclusion argument

model: the model that was run (e.g., mixed\_model)

controls: ignore this column for this test

method: the model that was run

p.value: the p-value for each factor

effect.size: irrelevant here

effect.size.ma: irrelevant here

effect.size.ma.lci: irrelevant here

effect.size.ma.hci: irrelevant here

statistic: the t-value for each factor

conf.low: the lower confidence interval for the estimate

conf.high: the higher confidence interval for the estimate

data\_used: a list with the data used for the specific test

**See Also**

[lme](#)

**Examples**

```
# Load example data
data(example_data)

cs1 <- paste0("CSP", 1:2)
cs2 <- paste0("CSM", 1:2)
subj <- "id"

# mixed models without groups
mixed_mf(cs1 = cs1, cs2 = cs2, subj = subj, data = example_data)

# mixed models with groups
mixed_mf(cs1 = cs1, cs2 = cs2, subj = subj, group = "group", data = example_data)
```

---

multiverse_cs	<i>multiverse_cs</i>
---------------	----------------------

---

## Description

Basic function for conducting multiverse analyses of conditioning data

## Usage

```
multiverse_cs(
  cs1,
  cs2,
  data,
  subj,
  group = NULL,
  cs_paired = NULL,
  include_bayes = TRUE,
  include_mixed = FALSE,
  phase = "acquisition",
  cutoff = c(0, 1, 2, 3),
  print_output = TRUE,
  correction = FALSE,
  meta.effect = "d_to_eta2"
)
```

## Arguments

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
group	the name of the group, if included, default to NULL
cs_paired	A character vector with the trials that were paired. Default is set to NULL, suggesting that there was full reinforcement
include_bayes	Whether the Bayesian analyses should be run. Default to TRUE
include_mixed	Whether the mixed model results should be run. Default to FALSE
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
cutoff	A numeric vector of the cutoff criteria applied. Default to 0, 0.05, .1
print_output	Whether to print the output or not. Default set to TRUE
correction	whether the Greenhouse-Geisser correction should be applied or not. Default to FALSE
meta.effect	How the meta-analytic effect should be computed.

**Details**

In case of higher order interaction, only the highest order effect is returned.

In case the CSs include only 1 observation per participant, or of unequal numbers of CS trials, the function will return the warning ""Skipping ANOVA due to the number of trials for the cs1 and/or cs2."".

In principle the `multiverse_cs` function runs the `universe_cs` function multiple times, so whatever holds for the `universe_cs` – e.g., in terms of warnings, holds for here as well.

**Value**

A tibble with the following column names: `x`: the name of the independent variable (e.g., `cs`) `y`: the name of the dependent variable as this defined in the `dv` argument `exclusion`: see `exclusion` argument `model`: the model that was run (e.g., t-test) `controls`: ignore this column for this test `method`: the method used `p.value`: the reported p-value `effect.size`: the reported effect size estimate: the estimate of the test run statistic: the value of the test `conf.low`: the lower confidence interval for the estimate `conf.high`: the higher confidence interval for the estimate `framework`: were the data analysed within a NHST or Bayesian framework? `data_used`: a list with the data used for the specific test

---

`rm_anova_mf`

*rm\_anova\_mf*

---

**Description**

Basic function for running the frequentist's repeated measures analysis of variance (ANOVA)

**Usage**

```
rm_anova_mf(
  cs1,
  cs2,
  data,
  subj,
  time = TRUE,
  group = NULL,
  phase = "acquisition",
  dv = "scr",
  exclusion = "full data",
  cut_off = "full data",
  correction = FALSE
)
```

**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
time	should time be included? Default to TRUE
group	the name of the group, if included, default to NULL
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off Name of the cut_off applied. Default to full data
correction	whether the Greenhouse-Geisser correction should be applied or not. Default to FALSE

**Details**

In case the `time` argument is set to `TRUE` (default value), the function will include this as a within subjects factor, assuming that the columns in `cs1` and `cs2` correspond to ascending time points (e.g., `cs1 trial 1`, `cs1 trial 2` ... `cs1 trial n`). If this is not the case, the results are not to be trusted.

The function uses the `ez::ezANOVA` function. The function gives by default a warning regarding the collapsing of factors. This function here suppresses this warning but the user should be aware of it. Please note that at the moment no sphericity correction is performed. The reported effect size is omega squared as this is computed by `sjstats::omega_sq`. The meta-analytic effect size is eta squared.

**Value**

A basic function for running repeated measures ANOVAs.

A tibble with the following column names:

`x`: the name of the independent variable (e.g., `cs`)

`y`: the name of the dependent variable as this defined in the `dv` argument

`exclusion`: see `exclusion` argument

`model`: the model that was run (e.g., t-test)

`controls`: ignore this column for this test

`method`: the model that was run

`p.value`: the p-value of the test

`effect.size`: the estimated effect size

`effect.size.ma`: the estimated effect size for the meta-analytic plots

effect.size.ma.lci: low confidence intervals for the meta-analytic effect size  
 effect.size.ma.hci: high confidence intervals for the meta-analytic effect size  
 estimate: the estimate of the test run  
 statistic: the F-value  
 conf.low: the lower confidence interval for the estimate  
 conf.high: the higher confidence interval for the estimate  
 framework: were the data analysed within a NHST or Bayesian framework?  
 data\_used: a list with the data used for the specific test

### Examples

```

# Load example data
data(example_data)

# Briefly define argument values that will be plugged in later on in the functions
cs1 <- paste0("CSP", 1:10)
cs2 <- paste0("CSM", 1:10)
subj <- "id"
group <- "group"

# Repeated measures ANOVA without groups
rm_anova_mf(cs1 = cs1, cs2 = cs2, subj = subj, data = example_data, time = TRUE)

# Repeated measures ANOVA with groups
rm_anova_mf(cs1 = cs1, cs2 = cs2, subj = subj, group = "group",
data = example_data, time = TRUE)

```

---

*rm\_banova\_mf*

*rm\_banova\_mf*

---

### Description

Basic function for running the Bayesian repeated measures analysis of Variance

### Usage

```

rm_banova_mf(
  cs1,
  cs2,
  data,
  subj,
  time = TRUE,
  group = NULL,
  phase = "acquisition",
  dv = "scr",
  exclusion = "full data",
  cut_off = "full data"
)

```



**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
time	should time be included? Default to TRUE
group	the name of the group, if included, default to NULL
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off Name of the cut_off applied. Default to full data

**Details**

In case the `time` argument is set to true, the function will include this as a within subjects factor, assuming that the columns in `cs1` and `cs2` correspond to ascending time points (e.g., `cs1 trial 1`, `cs1 trial 2` ... `cs1 trial n`). If this is not the case, the results are not to be trusted.

The ANOVA will run *\*all\** possible models and combinations. Please note that in case of many factors, this will mean that the analysis will take a long time to be completed.

**Value**

A tibble with the following column names:

`x`: the name of the independent variable (e.g., `cs`)

`y`: the name of the dependent variable as this defined in the `dv` argument

`exclusion`: see `exclusion` argument

`model`: the model that was run (e.g., rep ANOVA)

`controls`: ignore this column for this test

`method`: the model that was run

`p.value`: irrelevant here

`effect.size`: irrelevant here

`effect.size.ma`: irrelevant here

`effect.size.lci`: irrelevant here

`effect.size.hci`: irrelevant here

`estimate`: the estimate of the test run

`statistic`: the Bayes factor `conf.low`: the lower confidence interval for the estimate

`conf.high`: the higher confidence interval for the estimate

`framework`: were the data analysed within a NHST or Bayesian framework?

`data_used`: a list with the data used for the specific test

**Examples**

```
# Briefly define argument values that will be plugged in later on in the functions.
# We only use two trials as the function takes a long time to run.

data(example_data)
cs1 <- paste0("CSP", 1:2)
cs2 <- paste0("CSM", 1:2)
subj <- "id"

# Bayesian Repeated measures ANOVA without groups
rm_banova_mf(cs1 = cs1, cs2 = cs2, subj = subj,
data = example_data, time = TRUE)
```

---

*t\_test\_mf*


---

*t\_test\_mf*


---

**Description**

Basic function for running the frequentist's t-tests included in the main analyses

**Usage**

```
t_test_mf(
  cs1,
  cs2,
  data,
  subj,
  group = NULL,
  na.rm = FALSE,
  paired = TRUE,
  quanz = c(0.05, 0.95),
  meta.effect = "d_to_eta2",
  phase = "acquisition",
  dv = "scr",
  exclusion = "full data",
  cut_off = "full data"
)
```

**Arguments**

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses

subj	The name of the column including the participant numbers. Unique numbers are expected
group	the name of the group, if included, default to NULL
na.rm	Whether NAs should be removed, default to FALSE
paired	Whether the t-test refers to dependent (i.e., paired) or to independent sample(s). Default to TRUE
quanz	Quantiles for the meta-analytic effect sizes. Default to .05 (lower) and .95 (upper)
meta.effect	How the meta-analytic effect should be computed, Default to "d_to_eta2" (see details for more information)
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off Name of the cut_off applied. Default to full data

### Details

Given the correct names for the cs1, cs2, subj, and data, the function will run one- and two-sided frequentist's t-tests. In case cs1 or cs2 refer to multiple columns, the mean – per row – for each one of these variables will be computed first before running the t-test. Please note that cs1 is implicitly referred to the cs that is reinforced, and cs2 to the cs that is not reinforced. Depending on whether the data refer to an acquisition or extinction phase (as defined in the phase argument), the function will return a positive one-sided, or negative one-sided, respectively t-test in addition to the two-sided t-test. The returned effect size is Hedge's  $g$  in the column effect size. For the meta-analytic effect size (effect.size.ma), the returned effect size is eta-squared.

The function by default runs a Welch t-test, meaning it assumes unequal variances. This is due to calls that such a test should be preferred over Student's t-test, at least for paired samples t-test. Please note that if we let R decide which test to run – this is done by default in `stats::t.test`, then for some test there would be a Student t-test whereas in some others not. There are two different ways to compute the meta-analytic effect sizes but the results may differ. The option "t\_to\_eta2" computes the eta squared via the t values whereas the "d\_to\_eta2" the eta squared is computed via the Cohen's  $d$  value.

### Value

A tibble with the following column names:

x: the name of the independent variable (e.g., cs)

y: the name of the dependent variable as this defined in the dv argument exclusion: see exclusion argument

model: the model that was run (e.g., t-test)

controls: ignore this column for this test

method: the model that was run

p.value: the p-value of the test

effect.size: the estimated effect size  
effect.size.ma: the estimated effect size for the meta-analytic plots. Here we used eta squared  
effect.size.ma.lci: low confidence intervals for the meta-analytic effect size  
effect.size.ma.hci: high confidence intervals for the meta-analytic effect size  
estimate: the estimate of the test run. For the t-test is the mean of the differences  
statistic: the t-value  
conf.low: the lower confidence interval for the estimate  
conf.high: the higher confidence interval for the estimate  
framework: were the data analysed within a NHST or Bayesian framework?  
data\_used: a list with the data used for the specific test

### Examples

```
# Load example data
data(example_data)

# Paired sample t-tests
t_test_mf(cs1 = "CSP1", cs2 = "CSM1", subj = "id", data = example_data)

# Independent sample t-tests
t_test_mf(cs1 = "CSP1", cs2 = "CSM1", subj = "id", group = "group", data = example_data)
```

---

universe\_cs

*universe\_cs*

---

### Description

Basic function for running a multiverse analysis for a single data set

### Usage

```
universe_cs(
  cs1,
  cs2,
  data,
  subj,
  group = NULL,
  include_bayes = TRUE,
  include_mixed = FALSE,
  phase = "acquisition",
  dv = "scr",
  print_output = TRUE,
  exclusion = "full data",
  cut_off = "full data",
```

```

    correction = FALSE,
    meta.effect = "d_to_eta2"
  )

```

### Arguments

cs1	The column name(s) of the conditioned responses for the first conditioned stimulus
cs2	The column name(s) of the conditioned responses for the second conditioned stimulus
data	A data frame containing all the relevant columns for the analyses
subj	The name of the column including the participant numbers. Unique numbers are expected
group	the name of the group, if included, default to NULL
include_bayes	Whether the Bayesian analyses should be run. Default to TRUE
include_mixed	Whether the mixed model results should be run. Default to FALSE
phase	The conditioned phase that the analyses refer to. Accepted values are acquisition, acq, extinction, or ext
dv	name of the measured conditioned response. Default to "SCR"
print_output	Whether to print the output or not. Default set to TRUE
exclusion	Name of the data reduction procedure used. Default to full data
cut_off	cut off score
correction	whether the Greenhouse-Geisser correction should be applied or not. Default to FALSE
meta.effect	How the meta-analytic effect should be computed.

### Details

In case of higher order interaction, only the highest order effect is shown.

In case the CSs include only 1 observation per participant, or of unequal numbers of CS trials, the function will return the warning ""Skipping ANOVA due to the number of trials for the cs1 and/or cs2."".

### Value

A tibble with the following column names: x: the name of the independent variable (e.g., cs) y: the name of the dependent variable as this defined in the dv argument exclusion: see exclusion argument model: the model that was run (e.g., t-test) controls: ignore this column for this test method: the method used p.value: the reported p-value effect.size: the reported effect size estimate: the estimate of the test run statistic: the value of the test conf.low: the lower confidence interval for the estimate conf.high: the higher confidence interval for the estimate framework: were the data analysed within a NHST or Bayesian framework? data\_used: a list with the data used for the specific test

**Examples**

```
# Load data and define argument to be fed in universe_cs
data("example_data", package = "multifear")
example_data <- example_data[1:10, ]
cs1 <- paste0("CSP", 1:10)
cs2 <- paste0("CSM", 1:10)
subj = "id"
multifear::universe_cs(cs1, cs2, subj = subj, data = example_data, include_bayes = FALSE)
```

# Index

## \* datasets

example\_data, 6

bt\_test\_mf, 2

chop\_cs, 4

chop\_css, 5

combine\_cs, 6

example\_data, 6

exclusion\_criteria, 7

forestplot\_mf, 8

inference\_cs, 9

inference\_plot, 9

lme, 12

mixed\_mf, 10

multiverse\_cs, 13

rm\_anova\_mf, 14

rm\_banova\_mf, 16

t\_test\_mf, 18

universe\_cs, 20