

# Package ‘roistats’

October 14, 2022

**Title** Convenience Functions for Applying Basic Stats to Sub-Groups

**Version** 0.1.1

**Description** Easily applying same t-tests/basic data description across several sub-groups, with the output as a nice arranged data.frame. Multiple comparison and the significance symbols are also provided.

**License** MIT + file LICENSE

**Depends** R (>= 3.5.0)

**URL** <https://github.com/Irisfee/roistats>

**BugReports** <https://github.com/Irisfee/roistats/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** dplyr, tidyr, magrittr, purrr, stats, rlang

**Suggests** testthat (>= 3.0.0), covr, knitr, rmarkdown

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Yufei Zhao [aut, cre] (<<https://orcid.org/0000-0002-6511-8507>>),  
Daniel Anderson [ctb] (<<https://orcid.org/0000-0003-4699-4680>>)

**Maintainer** Yufei Zhao <[fay.zz0213@gmail.com](mailto:fay.zz0213@gmail.com)>

**Repository** CRAN

**Date/Publication** 2021-03-10 17:20:02 UTC

## R topics documented:

color_index . . . . .	2
color_index_two_sample . . . . .	2
df_sem . . . . .	3
p_range . . . . .	4
t_test_one_sample . . . . .	4
t_test_two_sample . . . . .	5

**Index**

7

---

color_index	<i>Color index</i>
-------------	--------------------

---

**Description**

The pre-processed data for identifying which brain regions is sensitive to the color memory of learned objects.

**Usage**

color\_index

**Format**

A tibble with 232 rows and 3 variables with one group attribute:

**subj\_id** Subject identity number

**roi\_id** Brain sub-region that of interest for the analysis. Served as the grouping variable here.

**color\_index** A value we want to test if it is significantly different from 0 across subjects.

**References**

Zhao, Y., Chanals, A.J.H. & Kuhl, B.A. (2021). Adaptive memory distortions are predicted by feature representations in parietal cortex. *Journal of Neuroscience*

**Examples**

color\_index

---

color_index_two_sample	<i>Color effect data for each group</i>
------------------------	---

---

**Description**

The pre-processed data for identifying which brain regions is sensitive to the color memory of learned objects.

**Usage**

color\_index\_two\_sample

**Format**

A tibble with 464 rows and 4 variables with one group attribute:

**subj\_id** Subject identity number

**roi\_id** Brain sub-region that of interest for the analysis. Served as the grouping variable here.

**group** A within-group variable for each subject. Indicating whether the color effect value is for the Paired or Control condition

**color\_effect** A value we want to test between the two groups (Paired vs Control).

**References**

Zhao, Y., Chanals, A.J.H. & Kuhl, B.A. (2021). Adaptive memory distortions are predicted by feature representations in parietal cortex. *Journal of Neuroscience*

**Examples**

```
color_index_two_sample
```

---

df_sem	<i>Generate standard error of mean</i>
--------	--

---

**Description**

Generate standard error of mean

**Usage**

```
df_sem(data, x)
```

**Arguments**

**data** A data frame, generally grouped by the intended sub-groups which you want to compare for the same t-test.

**x** A (bare) column name of the variable which you want to get the mean, sd, and standard error of the mean (SEM).

**Value**

A data.frame with consisting of characters. The columns that are always present are: group variable(s), mean, sd, n, and se(SEM).

**Examples**

```
df_sem(color_index, color_index)
```

```
library(magrittr)
color_index %>%
  df_sem(color_index)
```

---

p\_range *Create significant symbols for p-values*

---

**Description**

Create significant symbols for p-values

**Usage**

```
p_range(p)
```

**Arguments**

p A numeric p value (usually yielded from a statistical test).

**Value**

A character significant symbol. \* represents the p is within the range of (0.05, 0.01], \*\* for (0.01, 0.001], and \*\*\* for (0.001, +inf]

**Examples**

```
p_range(0.02)

library(dplyr)
t_test_one_sample(color_index, "color_index", mu = 0) %>% mutate(sig = p_range(p))
```

---

t\_test\_one\_sample *Generate one-sample t-test results for multiple sub-groups*

---

**Description**

This function produce one-sample t-test (two-tailed with confident interval at 0.95) results for multiple sub-groups and provides with a nice output in a table format. It can also add adjusted p values for multiple comparison issue.

**Usage**

```
t_test_one_sample(data, x, mu = 0, p_adjust = "bonferroni")
```

**Arguments**

data	A grouped data frame. It should be grouped by the intended sub-groups which you want to do the same t-test.
x	Column name of the variable which contains data values that you want to test (see <a href="#">t.test</a> and details).
mu	A number indicating the true value of the mean (or difference in means if you are performing a two sample test).
p_adjust	character indicating which method should be used for adjusting multiple comparisons (see <a href="#">p.adjust</a> and details). The default "bonferroni" corresponds to Bonferroni adjustment.

**Value**

A data.frame with the t-statistics table consisting of characters. The columns that are always present are: group variable(s), tvalue, df (degrees of freedom), p, and p\_adjustmethod(s).

**Examples**

```
t_test_one_sample(color_index, "color_index", mu = 0)

# use bonferroni and fdr method for adjusted p values.
library(magrittr)
color_index %>%
  t_test_one_sample("color_index", mu = 0, p_adjust = c("bonferroni", "fdr"))
```

---

t_test_two_sample	<i>Generate two-sample t-test results for multiple sub-groups</i>
-------------------	---

---

**Description**

This function produce two-sample t-test (two-tailed with confident interval at 0.95) results for multiple sub-groups and provides with a nice output in a table format. It can also add adjusted p values for multiple comparison issue.

**Usage**

```
t_test_two_sample(data, x, y, paired = FALSE, p_adjust = "bonferroni")
```

**Arguments**

data	A grouped data frame. It should be grouped by the intended sub-groups which you want to do the same t-test.
x	Column name of the variable which contains data values that you want to test (see <a href="#">t.test</a> and details).
y	Column name of the variable which contains data values of group assignments for the test values (see <a href="#">t.test</a> and details).

`paired` a logical indicating whether you want a paired t-test.  
`p_adjust` character indicating which method should be used for adjusting multiple comparisons (see [p.adjust](#) and details). The default "bonferroni" corresponds to Bonferroni adjustment.

**Value**

A data.frame with the t-statistics table consisting of characters. The columns that are always present are: group variable(s), tvalue, df (degrees of freedom), p, and p\_adjustmethod(s).

**Examples**

```
t_test_two_sample(color_index_two_sample, x = "color_effect", y = "group", paired = TRUE)

# use bonferroni and fdr method for adjusted p values.
library(magrittr)
color_index_two_sample %>%
  t_test_two_sample(
    x = "color_effect", y = "group", paired = TRUE, p_adjust = c("bonferroni", "fdr")
  )
```

# Index

## \* datasets

color\_index, 2

color\_index\_two\_sample, 2

color\_index, 2

color\_index\_two\_sample, 2

df\_sem, 3

p.adjust, 5, 6

p\_range, 4

t.test, 5

t\_test\_one\_sample, 4

t\_test\_two\_sample, 5