

Package ‘meta’

May 6, 2019

Type Package

Title Meta-Analysis for multiple data frames

Version 1.0

Date 2019-05-03

Author Jiecheng Song, Allison Emono and Soojin Kim

Maintainer Jiecheng Song <song.jiecheng@stonybrook.edu>, Allison E-
mono <allison.emono@stonybrook.edu>, Soo-
jin Kim <soojin.kim@stonybrook.edu>

Description Meta-Analysis for overall significance of a variable across multiple data frames

License GPL-2

R topics documented:

meta-package	1
meta.analysis	2
oneframe.analysis	4
print.meta	5
print.oneframe	5
print.singlevariable	6
print.summary.meta	6
single.variable.analysis	6
summary.meta	8

meta-package	<i>Meta-Analysis for multiple data frames</i>
--------------	---

Description

Meta-Analysis for overall significance of a variable across multiple data frames

Details

Performs appropriate tests for analyzing the overall significance of a variable across multiple data frames

Index: This package was not yet installed at build time.

Author(s)

Jiecheng Song, Allison Emono and Soojin Kim

Maintainer: Jiecheng Song <song.jiecheng@stonybrook.edu>, Allison Emono <allison.emono@stonybrook.edu>, Soojin Kim <soojin.kim@stonybrook.edu>

References

Chang, Lun-Ching, et al. "Meta-Analysis Methods for Combining Multiple Expression Profiles: Comparisons, Statistical Characterization and an Application Guideline." BMC Bioinformatics, vol. 14, no. 1, 21 Dec. 2013, p. 368., doi:10.1186/1471-2105-14-368.

Examples

```
## generate sample data
set.seed(123)
p = 100 # total number of variables
data1 = data.frame(group=sample(1:3,200,replace=TRUE), matrix(rnorm(p*200), ncol=p))
data2 = data.frame(group=sample(1:2,150,replace=TRUE), matrix(rnorm(p*150), ncol=p))

single.variable.analysis(data1$X1, data1$group, "Mean")
oneframe.analysis(data2, "Remove")
meta.analysis(data1, data2, "Fisher")
```

meta.analysis

Meta-Analysis for overall significance of a variable

Description

Performs appropriate tests to examine the overall significance of a categorical variable across different data frames

Usage

```
meta.analysis(data1 = NULL, data2 = NULL, data3 = NULL, data4 = NULL, data5 = NULL)
```

Arguments

data1	(required) a data frame where first column consists of group numbers and all other columns are different variables. For each variable, there must be at least two unique groups.
data2	(required) a data frame with the same format as data1. Must have the same number of columns as data1.
data3	(optional) a data frame with the same format as data1. If included, must have the same number of columns as the other data frames.
data4	(optional) a data frame with the same format as data1. If included, must have the same number of columns as the other data frames.
data5	(optional) a data frame with the same format as data1. If included, must have the same number of columns as the other data frames.
method	(required) a method to be used for pooling the p-values from different data frames. Must be one of "Fisher", "Stouffer", "Min" or "Max". Defaults to "Fisher". See reference for more details.

`dataclean` a function which indicates what should happen when the input data contains NAs, must be one of "Remove", "Mean" or "Median". Defaults to "Remove". See details for more description.

Details

First column of each data frame must indicate group membership which takes at least two unique values (e.g. 1 for group1 and 2 for group2). Function will not run if the first column does not contain integer values for group memberships.

All data frames must have identical number of columns where the first column indicates group membership and columns 2 to the last column represent the variables.

Each data frame must have consistent number of rows for all columns while the number of rows may be different for different data frames. In other words, the number of samples must be equal in a single data frame but may be different across different data frames.

Value

A summary table of the meta-analysis for the given data frames. Includes the data used, method used for pooling p-values and the test statistics and p-values for each variable with the alternative hypothesis stated at the end. You can also access the following components. See examples for more details.

<code>statistics</code>	the test statistics for each column (variable) for mean comparison.
<code>p.value</code>	the p-value for the test.
<code>data</code>	the data used in the test. If there are NAs in the data, it will be cleaned with the method specified by the user.
<code>method</code>	the method used for pooling the p-values from different data frames. This is specified by the user.

Note

You may print results with `print.meta()` or simply with `print`. See examples for more details. The printed result contains special symbols to indicate considerable p-values. " *** " for p-values less than 0.001, " ** " for p-values between 0.001 and 0.01, " * " for p-values between 0.01 and 0.05, " . " for p-values between 0.05 and 0.1 and no symbol for p-values above 0.1.

References

Chang, Lun-Ching, et al. "Meta-Analysis Methods for Combining Multiple Expression Profiles: Comparisons, Statistical Characterization and an Application Guideline." BMC Bioinformatics, vol. 14, no. 1, 21 Dec. 2013, p. 368., doi:10.1186/1471-2105-14-368.

Examples

```
## generate sample data
set.seed(123)
p = 100          # total number of variables
data1 = data.frame(group=sample(1:3,200,replace=TRUE), matrix(rnorm(p*200), ncol=p))
data2 = data.frame(group=sample(1:2,150,replace=TRUE), matrix(rnorm(p*150), ncol=p))

a = meta.analysis(data1, data2)
print(a)
print.meta(a)
```

oneframe.analysis *One Data frame Analysis*

Description

Compares group mean values for all variables in single data frame

Usage

```
oneframe.analysis(x, dataclean = "Remove")
```

Arguments

<code>x</code>	a data frame where first column consists of group numbers and all other columns are different variables. For each variable, there must be at least two unique groups.
<code>dataclean</code>	a function which indicates what should happen when the input data contains NAs, must be one of "Remove", "Mean" or "Median". Defaults to "Remove". See details for more description.

Details

This is an expansion of the function `single.variable.analysis()` in this package. The `single.variable.analysis()` function is applied to all columns (except the first group membership column) to compare group mean values and returns p-values for each variable.

Value

A summary table with method, test statistic and p-value for all variables. You can also access the following components. See examples for more details.

<code>method</code>	the method used in each column (variable) for mean comparison.
<code>statistic</code>	the test statistic of the chosen method.
<code>p.value</code>	the p-value for the test.
<code>data</code>	the data used in the test. If there are NAs in the data, it will be cleaned with the method specified by the user.

Note

You may print results with `print.oneframe()` or simply with `print`. See examples for more details. The printed result contains special symbols to indicate considerable p-values. " *** " for p-values less than 0.001, " ** " for p-values between 0.001 and 0.01, " * " for p-values between 0.01 and 0.05, " . " for p-values between 0.05 and 0.1 and no symbol for p-values above 0.1.

See Also

```
single.variable.analysis
```

Examples

```
## generate sample data
set.seed(123)
p = 100          # total number of variables
data1 = data.frame(group=sample(1:3,200,replace=TRUE), matrix(rnorm(p*200), ncol=p))

a = oneframe.analysis(data1)
print(a)
print.oneframe(a)
```

print.meta	<i>Print Meta-Analysis</i>
------------	----------------------------

Usage

```
print.meta(x)
```

Arguments

x	Results from the meta.analysis() function. See examples in meta.analysis() for more details.
---	--

See Also

```
meta.analysis
```

print.oneframe	<i>Print One Data frame Analysis</i>
----------------	--------------------------------------

Usage

```
print.oneframe(x)
```

Arguments

x	Results from the oneframe.test() function. See examples in oneframe.test() for more details.
---	--

See Also

```
oneframe.analysis
```

```
print.singlevariable
```

Print Single-variable Analysis

Usage

```
print.singlevariable(x)
```

Arguments

x	Results from the single.variable.analysis() function. See examples in single.variable.analysis() for more details.
---	--

See Also

```
single.variable.analysis
```

```
print.summary.meta
```

Print Summary of Meta-Analysis

Usage

```
print.summary.meta(x)
```

Arguments

x

See Also

```
summary.meta
```

```
single.variable.analysis
```

Single-variable Analysis

Description

Compares group mean values for a given single variable

Usage

```
single.variable.analysis(x, group, dataclean = "Remove")
```

Arguments

<code>x</code>	a list of numerical measurements for a single variable. Measurements must come from at least two different groups.
<code>group</code>	a list of group membership (i.e. 1 for group1 and 2 for group2), must have at least two unique groups and group names must be integers.
<code>dataclean</code>	a function which indicates what should happen when the input data contains NAs, must be one of "Remove", "Mean" or "Median". Defaults to "Remove". See details for more description.

Details

If either `x` or `group` contains NAs, you can choose to remove the missing value ("Remove"), replace it with the mean value ("Mean") or the median value ("Median") of the group of the missing value.

For data with two unique groups, if both groups have normal data, `t.test()` is used. If at least one group has non-normal data, `wilcox.test()` is used.

For data with more than two groups, if all groups have normal data, `anova(lm())` is used. If at least one group has non-normal data, `kruskal.test()` is used.

Value

A summary table of the test performed. Includes the method name, test statistic and the p-value. You can also access the following components. See examples for more details. A list containing the following components:

<code>method</code>	the method used for mean comparison.
<code>statistic</code>	the test statistic of the chosen method.
<code>p.value</code>	the p-value for the test.
<code>data</code>	the data used in the test. If there are NAs in the data, it will be cleaned with the method specified by the user.
<code>model</code>	the final summary table of the test performed.

Note

You may print results with `print.singlevariable()` or simply with `print`. See examples for more details.

See Also

`t.test` `wilcox.test` `anova` `lm` `kruskal.test`

Examples

```
## generate sample data
set.seed(123)
p = 100          # total number of variables
data1 = data.frame(group=sample(1:3,200,replace=TRUE), matrix(rnorm(p*200), ncol=p))

a = single.variable.analysis(data1$X1, data1$group)
print(a)
print.singlevariable(a)
```

summary.meta

*Summary of the Meta-analysis for significance of a variable***Description**

Summarizes the results from the meta-analysis performed on different data frames

Usage

```
summary.meta(x)
```

Arguments

x Results from the meta.analysis() function. See examples for more details.

Value

A summary of the meta-analysis for the given data frames. Includes similar components as those in meta.analysis() function but much clearer with more useful statements at the end. You can also access the following components. See examples for more details.

data	the data used in the test. If there are NAs in the data, it will be cleaned with the method specified by the user.
method	the method used for pooling the p-values from different data frames. This is specified by the user.
result	a data frame with all test statistics and p-value for each column (variable).
samplesize	the sample size for each data frame.
sig1	a data frame with all variables with p-value less than 0.01. Also shows the test statistics and the p-values.
sig2	a data frame with all variables with p-value less than 0.05. Also shows the test statistics and the p-values.
sig3	a data frame with all variables with p-value less than 0.10. Also shows the test statistics and the p-values.

Note

You may print results with print.summary.meta() or simply with print. See examples for more details. The printed result contains special symbols to indicate considerable p-values. " *** " for p-values less than 0.001, " ** " for p-values between 0.001 and 0.01, " * " for p-values between 0.01 and 0.05, " . " for p-values between 0.05 and 0.1 and no symbol for p-values above 0.1.

Examples

```
## generate sample data
set.seed(123)
p = 100 # total number of variables
data1 = data.frame(group=sample(1:3,200,replace=TRUE), matrix(rnorm(p*200), ncol=p))
data2 = data.frame(group=sample(1:2,150,replace=TRUE), matrix(rnorm(p*150), ncol=p))

a = meta.analysis(data1, data2)
summary.meta(a)
```