Package ‘selfea’

June 23, 2015

Type Package
Title Select Features Reliably with Cohen’s Effect Sizes
Version 1.0.1
Date 2015-06-20
Author Lang Ho Lee, Arnold Saxton, Nathan Verberkmoes
Maintainer Lang Ho Lee <langholee@gmail.com>
Depends R (>= 3.1.0), pwr, MASS, plyr, ggplot2
Description Functions using Cohen’s effect sizes (Cohen, Jacob. Statistical power analysis for the behavioral sciences. Academic press, 2013) are provided for reliable feature selection in biology data analysis. In addition to Cohen’s effect sizes, p-values are calculated and adjusted from quasi-Poisson GLM, negative binomial GLM and Normal distribution ANOVA. Significant features (genes, RNAs or proteins) are selected by adjusted p-value and minimum Cohen’s effect sizes, calculated to keep certain level of statistical power of biology data analysis given p-value threshold and sample size.
License GPL-2
NeedsCompilation no
Repository CRAN
Date/Publication 2015-06-23 01:14:10

R topics documented:

  selfea-package .................................................. 2
calculate_cohen_f2 ............................................. 3
draw_scatter_plots ................................................. 4
example_data ........................................................ 5
example_data1 ....................................................... 5
example_data2 ....................................................... 6
example_group ...................................................... 7
get_statistics_from_dataFrame .................................... 7
get_statistics_from_file .......................................... 10
glm_anova ............................................................ 12
top_table ............................................................. 13
ttest_cohens_d ..................................................... 15
**selfea-package**

**Index**

| selfea-package | Selfea: R package for reliable feature selection using Cohen’s effect sizes |

**Description**

Functions using Cohen’s effect sizes (Cohen, Jacob. Statistical power analysis for the behavioral sciences. Academic press, 2013) are provided for reliable feature selection in biology data analysis. In addition to Cohen’s effect sizes, p-values are calculated and adjusted from quasi-Poisson GLM, negative binomial GLM and Normal distribution ANOVA. Significant features (genes, RNAs or proteins) are selected by adjusted p-value and minimum Cohen’s effect sizes, calculated to keep certain level of statistical power of biology data analysis given p-value threshold and sample size.

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>selfea</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0.1</td>
</tr>
<tr>
<td>Date:</td>
<td>2015-06-20</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

**Author(s)**

Lang Ho Lee, Arnold Saxton, Nathan Verberkmoes

Maintainer: Lang Ho Lee <llee27@utk.edu>

**References**

Lang Ho Lee, Arnold Saxton, Nathan Verberkmoes, Selfea: A R package for reliable feature selection in process

**See Also**

get_statistics_from_dataFrame, get_statistics_from_file, top_table, ttest_cohens_d

**Examples**

```r
library(selfea)

## Test to calculate p-value of Student's t-test and Cohen's d
values <- c(8,18,8,8,11,29,26,22,27,26)
groups <- c("U200", "U200", "U200", "U200", "U600", "U600", "U600", "U600", "U600")
list_result <- ttest_cohens_d (values, groups, 0.05, 0.90)
```
calculate_cohen_f2

## Test selfea for single protein expression
values <- c(6,8,10,29,26,22)
groups <- c("U200", "U200", "U200", "U600", "U600", "U600")
experiments <- c("exp1", "exp2", "exp3", "exp4", "exp5", "exp6")

df_expr <- data.frame(ID="Protein_1", exp1=6, exp2=8, exp3=10, exp4=29, exp5=26, exp6=22)
df_group <- data.frame(Col_Name=experiments, Group=groups)
list_result <- get_statistics_from_dataframe(df_expr, df_group)
top_table(list_result)

## Load Gregori's data and test Selfea

## Josep Gregori, Laura Villarreal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013).
## An Effect Size Filter Improves the Reproducibility
## in Spectral Counting-based Comparative Proteomics.
## Journal of Proteomics, DOI http://dx.doi.org/10.1016/j.jprot.2013.05.030'

## Description:
## Each sample consists in 500ng of standard yeast lissate spiked with
## 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich).
## The dataset contains a different number of technical replimessages of each sample

## Import Gregori data
## data(example_data2)  ## if you want to test whole Gregori dataset
data(example_data1)  ## example_data1 has only 50 proteins for fast run

df_contrast <- example_data
df_group <- example_group

## calculate statistics including Cohen's effect sizes and p-values
## To see detail of method option, read R document about get_statistics_from_dataFrame.
list_result <- get_statistics_from_dataFrame(df_contrast, df_group, padj = 'fdr')

## get significant features by desired statistical power and alpha
## For this example, we set p-value threshold = 0.05, power = 0.84
## To see detail of method option, read R document about top_table.
significant_qpf <- top_table(list_result, pvalue=0.05, power_desired=0.84, method='QPF')

### calculate_cohen_f2


#### Usage

calculate_cohen_f2(model_glm, df_aov)
Arguments

- **model_glm**: GLM model generated by `glm` function
- **df_aov**: A data frame containing groups in 'Run' column and values in 'SC' column

Value

Cohen’s f2 (an effect size for linear models)

---

**draw_scatter_plots**

**draw_scatter_plots**

---

Description

Draw a scatterplot to show how significant IDs are distinguished from the total

Usage

```r
draw_scatter_plots(input_data_frame, max_pvalue, min_ES, power_desired, x_label, y_label)
```

Arguments

- **input_data_frame**: A data frame that consists of 'x' (P-value), 'y' (Effect size), 'cat' (significant or not).
- **max_pvalue**: P-value threshold
- **min_ES**: Effect size filter threshold
- **power_desired**: Give the statistical power you desired for output significant list
- **x_label**: Label of X axis
- **y_label**: Label of y axis

Value

A scatter plot
example_data

---

**example_data**

*Gregori Data: Yeast lisate samples spiked with human proteins*

---

**Description**

The spectral counts matrix has samples in the columns, and proteins in the rows. Each sample consists in 500ng of standard yeast lisate spiked with 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich). The dataset contains a different number of technical replicates of each sample. This dataset has only 100 proteins of total 685 proteins in the original data for fast example execution. If you want to use whole dataset, go for 'example_data2'.

**Usage**

```r
data(example_data)
```

**Format**

A data frame containing protein IDs and their expression profile.

**References**

Josep Gregori, Laura Villareal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013). An Effect Size Filter Improves the Reproducibility in Spectral Counting-based Comparative Proteomics. Journal of Proteomics, DOI [http://dx.doi.org/10.1016/j.jprot.2013.05.030](http://dx.doi.org/10.1016/j.jprot.2013.05.030)

**Examples**

```r
data(example_data)
```
Format

Two data frames, df_contrast (protein expression profile) and df_group (experiment group information).

References


Examples

data(example_data)

data(example_data2)

Description

The spectral counts matrix has samples in the columns, and proteins in the rows. Each sample consists in 500ng of standard yeast lysate spiked with 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich). The dataset contains a different number of technical replicates of each sample.

Usage

data(example_data2)

Format

Two data frames, df_contrast (protein expression profile) and df_group (experiment group information).

References


Examples

data(example_data2)
**Description**

The spectral counts matrix has samples in the columns, and proteins in the rows. Each sample consists in 500ng of standard yeast lisate spiked with 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich). The dataset contains a different number of technical replicates of each sample. This dataset has only 100 proteins of total 685 proteins in the original data for fast example execution. If you want to use whole dataset, go for `example_data2`.

**Usage**

data(example_data1)

**Format**

A data frame containing MS Run names and their corresponding experiment groups

**References**

Josep Gregori, Laura Villareal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013). An Effect Size Filter Improves the Reproducibility in Spectral Counting-based Comparative Proteomics. Journal of Proteomics, DOI [http://dx.doi.org/10.1016/j.jprot.2013.05.030](http://dx.doi.org/10.1016/j.jprot.2013.05.030)

**Examples**

data(example_data1)

---

**get_statistics_from_dataFrame**

**get_statistics_from_dataFrame**

**Description**

This function computes Cohen's f, f2 and w, adjusted p-value from GLM quasi-Poisson, negative binomial and Normal distribution.

**Usage**

get_statistics_from_dataFrame(df_contrast, df_group, padj = "fdr")
Arguments

- **df_contrast**: A data frame that consists of `ID` column and expression profile (columns after `ID` column). `ID` column should be unique. Column names after `ID` column should be unique. Only positive numbers are allowed in expression data. Here is an example.
get_statistics_from_dataFrame

<table>
<thead>
<tr>
<th>ID</th>
<th>Y500U100_001</th>
<th>Y500U100_002</th>
<th>Y500U200_001</th>
<th>Y500U200_002</th>
</tr>
</thead>
<tbody>
<tr>
<td>YKL060C</td>
<td>151</td>
<td>195</td>
<td>188</td>
<td>184</td>
</tr>
<tr>
<td>YDR155C</td>
<td>154</td>
<td>244</td>
<td>237</td>
<td>232</td>
</tr>
<tr>
<td>YOL086C</td>
<td>64</td>
<td>89</td>
<td>128</td>
<td>109</td>
</tr>
<tr>
<td>YJR104C</td>
<td>161</td>
<td>155</td>
<td>158</td>
<td>172</td>
</tr>
<tr>
<td>YGR192C</td>
<td>157</td>
<td>161</td>
<td>173</td>
<td>175</td>
</tr>
<tr>
<td>YLR150W</td>
<td>96</td>
<td>109</td>
<td>113</td>
<td>115</td>
</tr>
<tr>
<td>YPL037C</td>
<td>23</td>
<td>28</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>YNL007C</td>
<td>53</td>
<td>58</td>
<td>64</td>
<td>63</td>
</tr>
<tr>
<td>YBR072W</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>44</td>
</tr>
<tr>
<td>YDR418W_1</td>
<td>76</td>
<td>53</td>
<td>62</td>
<td>74</td>
</tr>
</tbody>
</table>

df_group  A data frame that consists of 'Col_Name' and 'Group' columns This parameter is to match experiment groups to expression profiles of df_contrast. 'Col_Name' should be corresponding to column names of expression profile of df_contrast. 'Group' columns have experiment informaion of columns in expression profile of df_contrast. Here is an example. See the example of df_contrast together.

<table>
<thead>
<tr>
<th>Col_Name</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y500U100_001</td>
<td>U100</td>
</tr>
<tr>
<td>Y500U100_002</td>
<td>U100</td>
</tr>
<tr>
<td>Y500U200_001</td>
<td>U200</td>
</tr>
<tr>
<td>Y500U200_002</td>
<td>U200</td>
</tr>
</tbody>
</table>

padj  Choose one of these c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). "fdr" is default option. The option is same to p.adjust.

Value

A list that consists of the following items:

- $data_table  A data frame that have statistics for each IDs
- $min_rep  Common number of replicates in your group information.
- $max_rep  Maximum number of replicates in your group information.
- $nt  The number of total experiments in your expression profile.
- $ng  The number of groups in your group information.
- $method_pvalue_adjustment  The selected method for p-value adjustment

data_table's elements

- Cohens_W  Cohen’s w
- Cohens_F  Cohen’s f
- Cohens_F2  Cohen’s f2
- Max_FC  Maximum fold change among all the possible group pairs
- QP_Pval_adjusted  Adjusted p-value from GLM quasi-Poisson
- NB_Pval_adjusted  Adjusted p-value from GLM negative binomial
get_statistics_from_file

Normal_Pval_adjusted  Adjusted p-value from Normal ANOVA

Examples

library(selfea)

## Test selfea for single protein expression
values <- c(6, 8, 10, 29, 26, 22)
groups <- c("U200", "U200", "U600", "U600", "U600")
experiments <- c("exp1", "exp2", "exp3", "exp4", "exp5", "exp6")

df_expr <- data.frame(ID="Protein_1", exp1=6, exp2=8, exp3=10, exp4=29, exp5=26, exp6=22)
df_group <- data.frame(Col_Name=experiments, Group=groups)
list_result <- get_statistics_from_dataframe(df_expr, df_group)
top_table(list_result)

## For this example we will import Gregori data
## Josep Gregori, Laura Villareal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013).
## An Effect Size Filter Improves the Reproducibility
## in Spectral Counting-based Comparative Proteomics.
## Journal of Proteomics, DOI http://dx.doi.org/10.1016/j.jprot.2013.05.030'

## Description:
## Each sample consists in 500ng of standard yeast llsate spiked with
## 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich).
## The dataset contains a different number of technical replilmmages of each sample

## import Gregori data
data(example_data)
df_contrast <- example_data
df_group <- example_group

## Get statistics through 'get_statistics_from_dataframe' function
list_result <- get_statistics_from_dataframe(df_contrast, df_group)

## Get significant features (alpha >= 0.05 and power >= 0.90)
significant_qpf <- top_table(list_result, pvalue=0.05, power_desired=0.90, method='QPF')

get_statistics_from_file

description

This function computes Cohen’s f, f2 and w, adjusted p-value from GLM quasi-Poisson, negative binomial and Normal distribution.

Usage

get_statistics_from_file(file_expr = "", file_group = "", padj = "fdr")
get_statistics_from_file

Arguments

file_expr: a CSV type file, comma (,) separated file format, that has unique "ID" at the first column and expression data for the corresponding ID. Here is an example.

ID,Y500U100_001,Y500U100_002,Y500U200_001,Y500U200_002
YKL060C,151,195,221,201
YDR155C,154,244,190,187
YOL086C,64,89,116,119

file_group: a CSV type file, comma (,) separated file format, that consists of "Col_Name", column names of "file_expr" parameter, and "Group" information of the corresponding column name. The order of "Col_Name" column have to be same to order of columns in "file_expr". Here is an example. See also the example above.

Col_Name,Group
Y500U100_001,U100
Y500U100_002,U100
Y500U200_001,U200
Y500U200_002,U200

padj: Choose one of these c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). "fdr" is default option. The option is same to p.adjust.

Value

A list that consists of the following items:

$data_table: A data frame that have statistics for each IDs
$min_rep: Common number of replicates in your group information.
$max_rep: Maximum number of replicates in your group information.
$nt: The number of total experiments in your expression profile.
$ng: The number of groups in your group information.
$method_pvalue_adjustment: The selected method for p-value adjustment

$data_table’s elements
Cohens_W: Cohen’s w
Cohens_F: Cohen’s f
Cohens_F2: Cohen’s f2
Max_FC: Maximum fold change among all the possible group pairs
QP_Pval_adjusted: Adjusted p-value from GLM quasi-Poisson
NB_Pval_adjusted: Adjusted p-value from GLM negative binomial
Normal_Pval_adjusted: Adjusted p-value from Normal ANOVA
### Examples

```r
library(selfea)

## For this example we will import Gregori data
## Josep Gregori, Laura Villareal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013).
## An Effect Size Filter Improves the Reproducibility
## in Spectral Counting-based Comparative Proteomics.
## Journal of Proteomics, DOI http://dx.doi.org/10.1016/j.jprot.2013.05.030

## Description:
## Each sample consists in 500ng of standard yeast lisate spiked with
## 100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1. Sigma-Aldrich).
## The dataset contains a different number of technical replimessagees of each sample

## Import Gregori data
data(example_data)
df_contrast <- example_data
df_group <- example_group

## Write Gregori data to use 'get_statistics_from_file' function
write.csv(df_contrast,"expression.csv",row.names=FALSE)
write.csv(df_group,"group.csv",row.names=FALSE)

## Get statistics
list_result <- get_statistics_from_file("expression.csv","group.csv","fdr")

## Get significant features (alpha >= 0.05 and power >= 0.90)
significant_qpf <- top_table(list_result,pvalue=0.05,power_desired=0.90,method='QPF')
```

---

### Description

Calculate P-values from ANOVA using Normal, Quasi-Poisson and Negative Binomial distribution and Cohen’s effect sizes

### Usage

```r
glm_anova(dataset.expr, dataset.ID, group, padj = "fdr")
```

### Arguments

- `dataset.expr`: A data frame that has column names for distinguishing experiments and numerical values for expression levels
- `dataset.ID`: A vector of the obtained expression profile’s ID column
- `group`: A data frame that consists of ‘Col_Name’ and ‘Group’ obtained from the user file through `get_statistics_from_file`. 
top_table

padj

Choose one of these c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). "fdr" is default option.

Value

A data frame containing ID, Cohen’s W, Cohen’s F, Max fold change, GLM Negative Binomial P-value, GLM Quasi-Poisson P-value and ANOVA with Normal P-value.

description

Get IDs that pass two filters, p-value and effect-size. This top_table will make a significant list that is less than p-value and greater than effect-size. Effect-size are calculated by obtained power level. This function requires four parameters. ex) top_table(input_data, pvalue=0.05, power_desired=0.90, method='QPF')

Usage

top_table(input_list, pvalue = 0.05, power_desired = 0.9, method = "QPF", FC_threshold = 2)

Arguments

input_list

The list should be produced by 'get_statistics_from_file' or 'get_statistics_from_dataFrame' function. See get_statistics_from_file and get_statistics_from_dataFrame for more information. It consists of the following items:

$data_table A data frame that have statistics for each IDs
$min_rep Common number of replicates in your group information.
$max_rep Maximum number of replicates in your group information.
$nt The number of total experiments in your expression profile.
$ng The number of groups in your group information.

pvalue p-value should be ranged between 0 to 1. default is 0.05.
power_desired Give the statistical power you desired for output significant list
method Choose statistics method you want to use for making significant list

"QPF" combination of Quasi-Poisson and Cohen’s f. Default.
"QPF2" combination of Quasi-Poisson and Cohen’s f2.
"QPCF" combination of Quasi-Poisson and Fold change.
"NBW" combination of Negative Binomial and Cohen’s w.
"NBF2" combination of Negative Binomial and Cohen’s f2.
"NBFC" combination of Negative Binomial and Fold change.
"NORF" combination of ANOVA with normal distribution and Cohen’s f.
"NORFC" combination of ANOVA with normal distribution and Fold change.
**FC_threshold**  
Fold change you want to use. Default is 2.

**Value**

A list containing the following items and a scatter plot that x-axis is effect size and y-axis is probability. Vertical line the plot is minimum effect size and horizontal line is maximum probability threshold. Red dots means insignificant, while blue dots are significant.

- **top_table**: a data frame that have calculated statistics for top table IDs
- **minimum_effect_size**: Minimum effect size threshold
- **selected_effect_size_filter**: The selected effect size filter
- **minimum_power**: Minimum statistical power in the top_table
- **selected_model**: The selected probability model for calculating p-value
- **alpha**: Maximum adjusted p-value
- **method_pvalue_adjustment**: The selected method for p-value adjustment
- **num_group**: The number of groups used for generating the top_table
- **common_replicates**: The number of common replicates
- **num_columns**: The number of columns (samples or experiments)

**top_table’s elements**

- **Cohens_W**: Cohen’s w
- **Cohens_F**: Cohen’s f
- **Cohens_F2**: Cohen’s f2
- **Max_FC**: Maximum fold change among all the possible group pairs
- **QP_Pval_adjusted**: Adjusted p-value from GLM quasi-Poisson
- **NB_Pval_adjusted**: Adjusted p-value from GLM negative binomial
- **Normal_Pval_adjusted**: Adjusted p-value from Normal ANOVA

**Examples**

```r
library(selfea)

## Test selfea for single protein expression
values <- c(6,8,10,29,26,22)
groups <- c("U200","U200","U200","U600","U600","U600")
experiments <- c("exp1","exp2","exp3","exp4","exp5","exp6")

df_expr <- data.frame(ID="Protein_1",exp1=6,exp2=8,exp3=10,exp4=29,exp5=26,exp6=22)
df_group <- data.frame(Col_Name=experiments,Group=groups)
list_result <- get_statistics_from_dataframe(df_expr,df_group)
top_table(list_result)

## For this example we will import Gregori data
## Josep Gregori, Laura Villareal, Alex Sanchez, Jose Baselga, Josep Villanueva (2013).
```
## An Effect Size Filter Improves the Reproducibility
## in Spectral Counting-based Comparative Proteomics.
## Journal of Proteomics, DOI http://dx.doi.org/10.1016/j.jprot.2013.05.030

### Description:
Each sample consists in 500ng of standard yeast lystate spiked with
100, 200, 400 and 600fm of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich).
The dataset contains a different number of technical replications of each sample

### import Gregori data
data(example_data)
df_contrast <- example_data
df_group <- example_group

### Get statistics through `get_statistics_from_dataframe` function
list_result <- get_statistics_from_dataframe(df_contrast, df_group)

### Get significant features (alpha >= 0.05 and power >= 0.90)
significant_qpf <- top_table(list_result, pvalue=0.05, power_desired=0.90, method='QPF')

### Description
Fulfill Welch Two Sample t-test (\texttt{t.test}) and calculate Cohen’s d as well as determine significance by p-value and effect size threshold.

### Usage
\texttt{ttest_cohens_d(values, groups, alpha = 0.05, power = 0.9, alternative = "two.sided", paired = FALSE, var.equal = FALSE)}

### Arguments
- \texttt{values}: A scalar vector. Length of both of two vectors, values and groups, should be same.
- \texttt{groups}: Experiment groups for the vector ‘values’. Length of both of two vectors, values and groups, should be same. The number of groups is not limited to two, such as \texttt{group <- c(‘A’,’A’,’A’,’B’,’B’)}.
- \texttt{alpha}: P-value threshold
- \texttt{power}: Give the statistical power you desired for output significant list
- \texttt{alternative}: Choose one of these c("two.sided", "less", "greater"). Default is "two.sided".
- \texttt{paired}: if two groups are paired, set it to TRUE. Default is FALSE.
- \texttt{var.equal}: if two groups are assumed to have same variance, set it to TRUE. Default is FALSE.
Value

A list containing the followings:

- observed_pvalue: Calculated P-value from T-test
- observed_cohens_d: Calculated Cohen's d
- threshold_cohens_d: Cohen's d threshold at the desired power
- threshold_pvalue: Desired p-value threshold
- flag_pvalue: TRUE=passed the pvalue threshold, FALSE=not
- flag_cohens_d: TRUE=passed the Cohen's d threshold, FALSE=not
- power_desired: Statistical power in your input parameters
- method: 'Welch Two Sample t-test'
- alternative: alternative option in your input parameters
- paired: paired option in your input parameters
- var.equal: var.equal option in your input parameters

Examples

```r
library(selfea)

values <- c(8,10,8,8,11,29,26,22,27,26)
groups <- c("U200","U200","U200","U200","U200","U600","U600","U600","U600")
list_result <- ttest_cohens_d(values, groups, 0.85, 0.90)
```
Index

*Topic Quasi-Poisson, Negative Binomial, Cohen’s f, Cohen’s w, Cohen’s f2, Cohen’s d, ANOVA
  selfea-package, 2
*Topic datasets
  example_data, 5
  example_data1, 5
  example_data2, 6
  example_group, 7

calculate_cohen_f2, 3
draw_scatter_plots, 4
  example_data, 5
  example_data1, 5
  example_data2, 6
  example_group, 7

get_statistics_from_dataFrame, 2, 7, 13
get_statistics_from_file, 2, 10, 13
glm_anova, 12

p.adjust, 9, 11

selfea (selfea-package), 2
selfea-package, 2

t.test, 15
top_table, 2, 13
ttest_cohens_d, 2, 15