

# Package ‘dabestr’

October 13, 2022

**Type** Package

**Title** Data Analysis using Bootstrap-Coupled Estimation

**Version** 0.3.0

**Maintainer** Joses W. Ho <joseshowh@gmail.com>

**Description** Data Analysis using Bootstrap-Coupled ESTimation.

Estimation statistics is a simple framework that avoids the pitfalls of significance testing. It uses familiar statistical concepts: means, mean differences, and error bars. More importantly, it focuses on the effect size of one's experiment/intervention, as opposed to a false dichotomy engendered by P values.

An estimation plot has two key features:

1. It presents all datapoints as a swarmplot, which orders each point to display the underlying distribution.
2. It presents the effect size as a bootstrap 95% confidence interval on a separate but aligned axes.

Estimation plots are introduced in Ho et al., Nature Methods 2019, 1548-7105. <doi:10.1038/s41592-019-0470-3>.

The free-to-view PDF is located at <<https://rdcu.be/bHhJ4>>.

**License** file LICENSE

**URL** <https://github.com/ACCLAB/dabestr>

**BugReports** <https://github.com/ACCLAB/dabestr/issues>

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0), magrittr, stats, utils

**Imports** boot, cowplot, dplyr, effsize, ellipsis, ggplot2 (>= 3.2), forcats, ggforce, ggbeeswarm, plyr, RColorBrewer, rlang, simpleboot, stringr, tibble, tidyr,

**RoxygenNote** 7.1.0

**Suggests** knitr, rmarkdown, tuftes, testthat, vdiff

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Joses W. Ho [cre, aut],  
Tayfun Tumkaya [aut]

**Repository** CRAN

**Date/Publication** 2020-07-13 08:50:08

**License\_is\_FOSS** yes

## R topics documented:

cohen_d_standardizers . . . . .	2
dabest . . . . .	3
dabestr . . . . .	5
hedges_correction . . . . .	5
lsat_scores . . . . .	6
mean_diff . . . . .	6
plot.dabest_effsize . . . . .	8
print.dabest . . . . .	12
print.dabest_effsize . . . . .	12
transcription_scores . . . . .	13
wellbeing_ind . . . . .	14
wellbeing_paired . . . . .	14
<b>Index</b>	<b>16</b>

---

cohen\_d\_standardizers *Compute the standardizers for Cohen's d*

---

### Description

Compute the standardizers for Cohen's d

### Usage

```
cohen_d_standardizers(x1, x2)
```

### Arguments

x1            A vector of values.  
x2            Another vector of values.

### Value

Named list for pooled and average standardizers. Use the pooled one for unpaired Cohens' d, and the average one for paired Cohens'd d.

---

dabest *Prepare Data for Analysis with dabestr*

---

### Description

dabest prepares a **tidy dataset** for analysis using estimation statistics.

### Usage

```
dabest(.data, x, y, idx, paired = FALSE, id.column = NULL)
```

### Arguments

.data	A data.frame or tibble.
x, y	Columns in .data.
idx	A vector containing factors or strings in the x columns. These must be quoted (ie. surrounded by quotation marks). The first element will be the control group, so all differences will be computed for every other group and this first group.
paired	Boolean, default FALSE. If TRUE, the two groups are treated as paired samples. The first group is treated as pre-intervention and the second group is considered post-intervention.
id.column	Default NULL. A column name indicating the identity of the datapoint if the data is paired. <i>This must be supplied if paired is TRUE.</i>

### Details

Estimation statistics is a statistical framework that focuses on effect sizes and confidence intervals around them, rather than  $P$  values and associated dichotomous hypothesis testing.

dabest() collates the data in preparation for the computation of **effect sizes**. Bootstrap resampling is used to compute non-parametric assumption-free confidence intervals. Visualization of the effect sizes and their confidence intervals using estimation plots is then performed with a specialized **plotting** function.

### Value

A dabest object with 8 elements.

**data** The dataset passed to **dabest**, stored here as a **tibble**.

**x and y** The columns in data used to plot the x and y axes, respectively, as supplied to **dabest**. These are **quoted variables** for **tidy evaluation** during the computation of effect sizes.

**idx** The vector of control-test groupings. For each pair in idx, an effect size will be computed by downstream dabestr functions used to compute **effect sizes** (such as `mean_diff()`).

**is.paired** Whether or not the experiment consists of paired (aka repeated) observations.

**id.column** If **is.paired** is TRUE, the column in data that indicates the pairing of observations.

**.data.name** The variable name of the dataset passed to **dabest**.

**.all.groups** All groups as indicated in the **idx** argument.

**See Also**

- [Effect size computation](#) from the loaded data.
- [Generating estimation plots](#) after effect size computation.

**Examples**

```
# Performing unpaired (two independent groups) analysis.
unpaired_mean_diff <- dabest(iris, Species, Petal.Width,
                             idx = c("setosa", "versicolor"),
                             paired = FALSE)

# Display the results in a user-friendly format.
unpaired_mean_diff

# Compute the mean difference.
mean_diff(unpaired_mean_diff)

# Plotting the mean differences.
mean_diff(unpaired_mean_diff) %>% plot()

# Performing paired analysis.
# First, we munge the `iris` dataset so we can perform a within-subject
# comparison of sepal length vs. sepal width.

new.iris <- iris
new.iris$ID <- 1: length(new.iris)
setosa.only <-
  new.iris %>%
  tidyr::gather(key = Metric, value = Value, -ID, -Species) %>%
  dplyr::filter(Species %in% c("setosa"))

paired_mean_diff <- dabest(setosa.only, Metric, Value,
                           idx = c("Sepal.Length", "Sepal.Width"),
                           paired = TRUE, id.col = ID) %>%
  mean_diff()

# Using pipes to munge your data and then passing to `dabest`.
# First, we generate some synthetic data.
set.seed(12345)
N <- 70
c <- rnorm(N, mean = 50, sd = 20)
t1 <- rnorm(N, mean = 200, sd = 20)
t2 <- rnorm(N, mean = 100, sd = 70)
long.data <- tibble::tibble(Control = c, Test1 = t1, Test2 = t2)

# Munge the data using `gather`, then pass it directly to `dabest`
meandiff <- long.data %>%
  tidyr::gather(key = Group, value = Measurement) %>%
  dabest(x = Group, y = Measurement,
```

```
      idx = c("Control", "Test1", "Test2"),
      paired = FALSE) %>%
mean_diff()
```

---

dabestr

*dabestr: A package for producing estimation plots.*

---

### Description

The dabestr package provides functions to construct bootstrap confidence intervals for differences between groups for a range of [effect sizes](#) and a function to generate [estimation plots](#).

---

hedges\_correction

*Returns the exact Hedges' correction factor for Cohen's d.*

---

### Description

Returns the exact Hedges' correction factor for Cohen's d.

### Usage

```
hedges_correction(x1, x2)
```

### Arguments

x1            A vector of values.  
x2            Another vector of values.

### Value

Hedges' correction for the g of x1 and x2.

---

 lsat\_scores

*LSAT Scores.*


---

### Description

This dataset is taken from Thomason et al. (2014) who studied a group of 12 students who underwent a course in critical thinking. Their scores on the Logical Reasoning section of the Law School Aptitude Test (LSAT) were assessed before and after training.

### Usage

lsat\_scores

### Format

A list with two elements: pretrain and posttrain.

### Details

It is also found in Chapter 8 "The Paired Design" (pp 195 - 199) of Introduction to the New Statistics (Routledge, 2017), by Geoff Cumming and Robert Calin-Jageman.

### Source

Thomason, N. R., Adajian, T., Barnett, A. E., Boucher, S., van der Brugge, E., Campbell, J., Knorpp, W., Lempert, R., Lengbeyer, L., Mandel, D. R., Rider, Y., van Gelder, T., & Wilkins, J. (2014). *Critical thinking final report*. The University of Melbourne, N66001-12-C-2004.

---

 mean\_diff

*Compute Effect Size(s)*


---

### Description

For each pair of observations in a dabest object, a desired effect size can be computed. Currently there are five effect sizes available:

- The **mean difference**, given by `mean_diff()`.
- The **median difference**, given by `median_diff()`.
- **Cohen's *d***, given by `cohens_d()`.
- **Hedges' *g***, given by `hedges_g()`.
- **Cliff's delta**, given by `cliffs_delta()`.

**Usage**

```
mean_diff(x, ci = 95, reps = 5000, seed = 12345)

median_diff(x, ci = 95, reps = 5000, seed = 12345)

cohens_d(x, ci = 95, reps = 5000, seed = 12345)

hedges_g(x, ci = 95, reps = 5000, seed = 12345)

cliffs_delta(x, ci = 95, reps = 5000, seed = 12345)
```

**Arguments**

**x** A dabest object, generated by the [dabest\(\)](#) function.

**ci** float, default 95. The level of the confidence intervals produced. The default ci = 95 produces 95% CIs.

**reps** integer, default 5000. The number of bootstrap resamples that will be generated.

**seed** integer, default 12345. This specifies the seed used to set the random number generator. Setting a seed ensures that the bootstrap confidence intervals for the same data will remain stable over separate runs/calls of this function. See [set.seed](#) for more details.

**Value**

A dabest\_effsize object with 10 elements.

**data** The dataset passed to [dabest\(\)](#), as a [tibble](#).

**x and y** The columns in data used to plot the x and y axes, respectively, as supplied to [dabest\(\)](#). These are [quoted variables](#) for [tidy evaluation](#) during the computation of effect sizes.

**idx** The vector of control-test groupings as initially passed to [dabest\(\)](#).

**is.paired** Whether or not the experiment consists of paired (aka repeated) observations. Originally supplied to [dabest\(\)](#).

**id.column** If **is.paired** is TRUE, the column in data that indicates the pairing of observations. As passed to [dabest\(\)](#).

**effect.size** The effect size being computed. One of the following: `c("mean_diff", "median_diff", "cohens_d", "hedges_g", "cliffs_delta")`.

**.data.name** The variable name of the dataset passed to [dabest\(\)](#).

**summary** A [tibble](#) with a row for the mean or median of each group in the x column of data, as indicated in **idx**.

**result** A [tibble](#) with the following 15 columns:

- control\_group, test\_group** The name of the control group and test group respectively.
- control\_size, test\_size** The number of observations in the control group and test group respectively.
- effect\_size** The effect size used.

- paired** Is the difference paired (TRUE) or not (FALSE)?
- difference** The effect size of the difference between the two groups.
- variable** The variable whose difference is being computed, ie. the column supplied to y.
- ci** The ci passed to this function.
- bca\_ci\_low, bca\_ci\_high** The lower and upper limits of the Bias Corrected and Accelerated bootstrap confidence interval.
- pct\_ci\_low, pct\_ci\_high** The lower and upper limits of the percentile bootstrap confidence interval.
- bootstraps** The vector of bootstrap resamples generated.

### See Also

- [Loading data](#) for effect size computation.
- [Generating estimation plots](#) after effect size computation.
- The [mathematical definitions](#) and equations used to compute each effect size.
- The [effsize](#) package, which is used under the hood to compute Cohen's  $d$ , Hedges'  $g$ , and Cliff's delta.
- The [boot\(\)](#) and [boot.ci\(\)](#) functions from the boot package, which generate the (nonparametric) bootstrapped resamples used to compute the confidence intervals.

### Examples

```
# Loading data for unpaired (two independent groups) analysis.
petal_widths <- dabest(iris, Species, Petal.Width,
                      idx = c("setosa", "versicolor"),
                      paired = FALSE)

# Compute the mean difference.
mean_diff(petal_widths)

# Plotting the mean differences.
mean_diff(petal_widths) %>% plot()
```

---

plot.dabest\_effsize    *Create an Estimation Plot*

---

### Description

An estimation plot has two key features.

1. It presents all datapoints as a [swarmplot](#) or [sinaplot](#), which orders each point to display the underlying distribution.
2. It presents the effect size as a bootstrap 95 percent confidence interval on a separate but aligned axes.



Estimation plots emerge from estimation statistics, an intuitive framework that avoids the pitfalls of significance testing. It uses familiar statistical concepts: means, mean differences, and error bars. More importantly, it focuses on the effect size of one's experiment/intervention, as opposed to a false dichotomy engendered by  $P$  values. This function takes the output of the [effect size](#) functions and produces an estimation plot.

## Usage

```
## S3 method for class 'dabest_effsize'
plot(
  x,
  ...,
  color.column = NULL,
  palette = "Set1",
  float.contrast = TRUE,
  slopegraph = TRUE,
  group.summaries = "mean_sd",
  rawplot.type = c("swarmplot", "sinaplot"),
  rawplot.ylim = NULL,
  rawplot.ylabel = NULL,
  rawplot.markersize = 2,
  rawplot.groupwidth = 0.3,
  effsize.ylim = NULL,
  effsize.ylabel = NULL,
  effsize.markersize = 4,
  theme = ggplot2::theme_classic(),
  tick.fontsize = 11,
  axes.title.fontsize = 14,
  show.legend = TRUE,
  swarmplot.params = NULL,
  sinaplot.params = NULL,
  slopegraph.params = NULL
)
```

## Arguments

<code>x</code>	A <code>dabest_effsize</code> object, generated by one of the <a href="#">effect size</a> functions currently available in <code>dabestr</code> .
<code>...</code>	Signature for S3 generic function.
<code>color.column</code>	default <code>NULL</code> . This is a column in the <code>data.frame</code> passed to the <code>dabest</code> function. This column will be treated as a <a href="#">factor</a> and used to color the datapoints in the rawdata swarmplot.
<code>palette</code>	default <code>"Set1"</code> . Accepts any one of the <code>RColorBrewer</code> palettes, or a vector of colors. Colors can be specified as RGB hexcode or as a named color. See the "Palettes" section in <a href="#">scale_color_brewer</a> for more on palettes. To obtain all 657 named colors in R, enter <code>colors()</code> at the console.
<code>float.contrast</code>	default <code>TRUE</code> . If <code>idx</code> in the <code>dabest</code> object contains only 2 groups, <code>float.contrast</code>

	= TRUE will plot the effect size and the bootstrap confidence interval in a horizontally-aligned axes (also known as a <a href="#">Gardner-Altman plot</a> ).
slopegraph	boolean, default TRUE. If the dabest object contains paired comparisons, slopegraph = TRUE will plot the rawdata as a <a href="#">Tufte slopegraph</a> .
group.summaries	"mean_sd", "median_quartiles", or NULL. Plots the summary statistics for each group. If 'mean_sd', then the mean and standard deviation of each group is plotted as a gapped line beside each group. If 'median_quartiles', then the median and 25th & 75th percentiles of each group is plotted for each group as a gapped line. If group.summaries = NULL, the summaries are not shown.
rawplot.type	default "beeswarm". Accepts either "beeswarm" or "sinaplot". See <a href="#">geom_quasirandom</a> and <a href="#">geom_sina</a> for more information.
rawplot.ylim	default NULL. Enter a custom y-limit for the rawdata plot. Accepts a vector of length 2 (e.g. c(-50, 50)) that will be passed along to <a href="#">coord_cartesian</a> .
rawplot.ylabel	default NULL. Accepts a string that is used to label the rawdata y-axis. If NULL, the column name passed to y is used.
rawplot.markersize	default 2. This is the size (in points) of the dots used to plot the individual datapoints. There are 72 points in one inch. See <a href="#">this article</a> for more info.
rawplot.groupwidth	default 0.3. This is the maximum amount of spread (in the x-direction) allowed, for each group.
effsize.ylim	default NULL. Enter a custom y-limit for the effect size plot. This parameter is ignored if float.contrast = TRUE. Accepts a vector of length 2 (e.g. c(-50, 50)) that will be passed along to <a href="#">coord_cartesian</a> .
effsize.ylabel	default NULL. Accepts a string that is used to label the effect size y-axis. If NULL, this axes will be labeled "(un)paired <effect size>", where <i>effect size</i> is the <a href="#">effect size</a> function used to generate the dabest_effsize object currently being plotted.
effsize.markersize	default 4. This is the size (in points) of the dots used to indicate the effect size.
theme	default <a href="#">theme_classic</a> . This is the ggplot2 theme that is used to style various elements of the estimation plot.
tick.fontsize	default 11. This controls the font size (in points) of all tick labels.
axes.title.fontsize	default 14. This determines the font size (in points) of the axes titles.
show.legend	boolean, default TRUE. If FALSE, the color legend will not be displayed, even if color.column is supplied.
swarmplot.params	default NULL. Supply list of keyword = value pairs to <a href="#">geom_quasirandom</a> .
sinaplot.params	default NULL. Supply list of keyword = value pairs to <a href="#">ggforce::geom_sina()</a> .
slopegraph.params	default NULL. Supply list of keyword = value pairs to ggplot2's <a href="#">geom_line()</a> . This controls the appearance of the lines plotted for a paired slopegraph.

**Value**

A `ggproto` object. This object is actually composed of two `ggplot2` objects (one for the rawdata swarmplot or slopegraph, another for the effect sizes and bootstrapped confidence intervals). These are arranged in the desired configuration (whether as a Gardner-Altman plot or a Cumming plot) by the `plot_grid()` function in the `cowplot` package.

**References**

**Moving beyond P values: Data analysis with estimation graphics.** Nature Methods 2019, 1548-7105. Jose Ho, Tayfun Tumkaya, Sameer Aryal, Hyungwon Choi, Adam Claridge-Chang

**See Also**

- [Loading data](#) for effect size computation.
- [Effect size computation](#) from the loaded data.

Run `vignette("Using dabestr", package = "dabestr")` in the console to read more about using parameters to control the plot features.

**Examples**

```
# Performing unpaired (two independent groups) analysis.
# We want to obtain the mean difference between the petal widths
# of setosa and versicolor species.
unpaired_mean_diff <- dabest(iris, Species, Petal.Width,
                             idx = c("setosa", "versicolor"),
                             paired = FALSE) %>%
  mean_diff()

# Create a Gardner-Altman estimation plot.
plot(unpaired_mean_diff)

## Not run:
# Comparing versicolor and virginica petal widths to setosa petal width.
shared_control_data <- dabest(iris, Species, Petal.Width,
                              idx = c("setosa", "versicolor", "virginica")) %>%
  mean_diff()

# Create a Cumming estimation plot.
plot(shared_control_data)

## End(Not run)
```

print.dabest      *Print a 'dabest' object*

---

**Description**

Print a 'dabest' object

**Usage**

```
## S3 method for class 'dabest'  
print(x, ...)
```

**Arguments**

x                    A [dabest](#) object, generated by the function of the same name.  
...                   S3 signature for generic plot function.

**Value**

A summary of the experimental designs.

**Examples**

```
# Performing unpaired (two independent groups) analysis.  
unpaired_mean_diff <- dabest(iris, Species, Petal.Width,  
                              idx = c("setosa", "versicolor"),  
                              paired = FALSE)  
  
# Display the results in a user-friendly format.  
print(unpaired_mean_diff)
```

---

print.dabest\_effsize      *Print a 'dabest\_effsize' object*

---

**Description**

Print a 'dabest\_effsize' object

**Usage**

```
## S3 method for class 'dabest_effsize'  
print(x, ..., signif_digits = 3)
```

**Arguments**

- x A dabest\_effsize object, generated by one of the [effect size computation functions](#).
- ... S3 signature for generic plot function.
- signif\_digits Integer, default 3. All numeric figures in the printed output will be rounded off to this number of significant digits.

**Value**

A summary of the effect sizes and respective confidence intervals.

**Examples**

```
# Performing unpaired (two independent groups) analysis.
unpaired_mean_diff <- dabest(iris, Species, Petal.Width,
                             idx = c("setosa", "versicolor"),
                             paired = FALSE) %>%
  mean_diff()

# Display the results in a user-friendly format.
print(unpaired_mean_diff)
```

---

transcription\_scores *Transcription Scores*.

---

**Description**

This dataset is taken from Mueller and Oppenheimer (2014), comparing the percentage of notes that was verbatim transcribed during a lecture by two independent groups of students: one using pen and paper, and one using laptops.

**Usage**

```
transcription_scores
```

**Format**

A list with two elements: pen and laptop.

**Details**

It is also found in Chapter 7 "The Independent Groups Design" (pp 160 - 166) of Introduction to the New Statistics (Routledge, 2017), by Geoff Cumming and Robert Calin-Jageman.

**Source**

Mueller, Pam A., and Daniel M. Oppenheimer. "The Pen Is Mightier than the Keyboard: Advantages of Longhand over Laptop Note Taking." *Psychological Science*, vol. 25, no. 6, June 2014, pp. 1159–68, doi:10.1177/0956797614524581.

---

wellbeing\_ind                      *Wellbeing Scores (2 independent groups).*

---

**Description**

This is a synthetic dataset from Geoff Cumming. 20 students were randomly assigned to spend the afternoon reading in the library—the Control condition—or reading in the local botanical gardens—the Test condition. At the end of the session, each student completed a measure of his or her perceived well-being.

**Usage**

wellbeing\_ind

**Format**

A list with two elements: control and test.

**Details**

It is found in Chapter 11 "The Paired Design" (page 286) of *Understanding the New Statistics* (Routledge, 2012) by Geoff Cumming.

**Source**

Cumming, G. *Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis*. Routledge 2012. <https://books.google.com/books?id=AVBDYgEACAAJ>

---

wellbeing\_paired                      *Wellbeing Scores (Before and after design).*

---

**Description**

This is a synthetic dataset from Geoff Cumming. A single group of 10 students first completed a well-being questionnaire (before), spent the afternoon reading in the botanical gardens, then gave well-being scores once again (after).

**Usage**

wellbeing\_paired

**Format**

A list with two elements: before and after.

**Details**

It is found in Chapter 11 "The Paired Design" (page 291) of Understanding the New Statistics (Routledge, 2012) by Geoff Cumming.

**Source**

Cumming, G. Understanding the New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis. Routledge 2012. <https://books.google.com/books?id=AVBDYgEACAAJ>

# Index

## \* datasets

- lsat\_scores, 6
- transcription\_scores, 13
- wellbeing\_ind, 14
- wellbeing\_paired, 14

- boot, 8
- boot.ci, 8

- cliffs\_delta (mean\_diff), 6
- cohen\_d\_standardizers, 2
- cohens\_d (mean\_diff), 6
- coord\_cartesian, 10

- dabest, 3, 3, 12
- dabest(), 7
- dabestr, 5

- effect size, 9, 10
- Effect size computation, 4, 11
- effect size computation, 13
- effect sizes, 3, 5
- effsize, 8
- estimation plots, 5

- factor, 9

- Generating estimation plots, 4, 8

- geom\_line(), 10
- geom\_quasirandom, 10
- geom\_sina, 10
- ggforce::geom\_sina(), 10
- ggproto, 11

- hedges\_correction, 5
- hedges\_g (mean\_diff), 6

- Loading data, 8, 11
- lsat\_scores, 6

- mean\_diff, 6

- median\_diff (mean\_diff), 6

- plot.dabest\_effsize, 8
- plot\_grid, 11
- plotting, 3
- print.dabest, 12
- print.dabest\_effsize, 12

- scale\_color\_brewer, 9
- set.seed, 7

- theme\_classic, 10
- tibble, 3, 7
- transcription\_scores, 13

- wellbeing\_ind, 14
- wellbeing\_paired, 14