Package 'stddiff.spark'

November 3, 2026

Title Calculate the Standardized Difference for Numeric, Binary and Category Variables in Apache Spark

Version 1.1.0

Description Provides functions to compute standardized differences for numeric, binary, and categorical variables on Apache Spark DataFrames using 'sparklyr'. The implementation mirrors the methods used in the 'stddiff' package but operates on distributed data. See Zhicheng Du, Yuantao Hao (2022) <doi:10.32614/CRAN.package.stddiff> for reference.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.0), tidyr (>= 1.3.0), sparklyr (>= 1.8.4)

Suggests stddiff (>= 3.1), testthat (>= 3.0.0), withr

Config/testthat/edition 3

SystemRequirements Apache Spark (tested with 3.4.4 and 4.2.0)

URL https://github.com/alicja-januszkiewicz/stddiff.spark

BugReports https://github.com/alicja-januszkiewicz/stddiff.spark/issues

NeedsCompilation no

Author Alicja Januszkiewicz [aut, cre, cph]

Maintainer Alicja Januszkiewicz <cran.alicja.januszkiewicz@gmail.com>

Repository CRAN

Date/Publication 2026-11-03 05:30:02 UTC

Contents

stddiff.binary 2
stddiff.category 3
stddiff.numeric 5

Index **7**

`stddiff.binary` *Compute Standardized Differences for Binary Variables (Spark)*

Description

Calculates standardized differences for binary variables using a Spark DataFrame. Equivalent to `stddiff::stddiff.binary` but operates on Spark data.

Usage

```
stddiff.binary(data, gcol, vcol, verbose = FALSE)
```

Arguments

`data`	A Spark DataFrame (`tbl_spark`) containing the variables.
`gcol`	Integer; column index of the binary grouping variable (e.g., treatment vs control).
`vcol`	Integer vector; column indices of the binary variables to analyze. Repeated indices give repeated rows, and `gcol` may be included like any other column, as in `stddiff`.
`verbose`	Logical; if TRUE, prints progress messages. Default is FALSE.

Details

Spark DataFrames have no factor type, so values are ordered the way `as.factor()` orders them in `stddiff`: numerically for numeric columns, alphabetically for strings, and `FALSE` before `TRUE`. The first value of each variable becomes 0 and the second becomes 1, so `p.c` and `p.t` are proportions of the second value. The group column is ordered the same way, and its first value is the control group.

Like `stddiff`, this does not check that variables are binary. A variable with more than two distinct values is coded 0, 1, 2, ... in that order, so its results are not meaningful: the proportions can exceed 1, and the standardized difference can be `NaN`, with R's "NaNs produced" warning. Each distinct value is collected, so don't pass continuous or ID-like columns.

Differences from `stddiff` on Spark data: the group column must have exactly two values, or the function stops with an error. `stddiff` doesn't check this, and with more than two groups it silently compares the first two. `NaN` counts as missing, where `stddiff` keeps it as a value, because sparklyr's `copy_to()` turns some R `NA`s into `NaN`. With more than about 46,000 rows per group, the confidence limits are computed where `stddiff`'s integer counts overflow and give `NA`. Spark also has no factor type, so the data cannot carry a custom level order. Groups and values follow their sorted order, which is the default order in `stddiff`. Local data frames are passed to `stddiff`, which uses the level order of factor columns.

The standardized difference is computed as:

$$d = \frac{|p_t - p_c|}{\sqrt{(p_t(1-p_t) + p_c(1-p_c))/2}}$$

Value

A numeric matrix with one row per variable and columns:

- `p.c`: Proportion in control group (first value of the group column)
- `p.t`: Proportion in treatment group (second value of the group column)
- `missing.c`: Number of missing values in control group
- `missing.t`: Number of missing values in treatment group
- `stddiff`: Standardized difference
- `stddiff.l`: Lower bound of 95% confidence interval
- `stddiff.u`: Upper bound of 95% confidence interval

See Also

`stddiff.category`, `stddiff.numeric`

Examples

```
sc <- sparklyr::spark_connect(master = "local")

spark_df <- sparklyr::copy_to(sc, mtcars)

result <- stddiff.binary(
  data = spark_df,
  gcol = 9,   # column index of grouping variable
  vcol = c(8) # columns of binary variables
)

sparklyr::spark_disconnect(sc)
```

`stddiff.category` *Compute Standardized Differences for Categorical Variables (Spark)*

Description

Calculates standardized differences for categorical variables using a Spark DataFrame. Equivalent to `stddiff::stddiff.category` but operates on Spark data.

Usage

```
stddiff.category(data, gcol, vcol, verbose = FALSE)
```

Arguments

data	A Spark DataFrame (tbl_spark) containing the variables.
gcol	Integer; column index of the binary grouping variable.
vcol	Integer vector; column indices of the categorical variables to analyze. Repeated indices give repeated rows, and gcol may be included like any other column, as in stddiff.
verbose	Logical; if TRUE, prints progress messages. Default is FALSE.

Details

For categorical variables with K levels, the standardized difference is computed using a multivariate approach that accounts for all K-1 levels simultaneously (excluding the reference level). Levels are ordered the way as.factor() orders them in stddiff (numerically for numeric columns, alphabetically for strings), and the first level serves as the reference. The standardized difference does not depend on which level is the reference. Each distinct value is collected, so don't pass continuous or ID-like columns.

Differences from stddiff on Spark data: where stddiff stops with an error for a variable, such as a singular matrix, that variable gets NA with a warning and the other variables still get results. Local data frames are passed to stddiff, so the error remains there. The group column must have exactly two values, or the function stops with an error that says so. stddiff doesn't check this, and with more than two groups it stops with an unrelated error. NaN counts as missing, where stddiff keeps it as a level, because sparklyr's copy_to() turns some R NAs into NaN. With more than about 46,000 rows per group, the confidence limits are computed where stddiff's integer counts overflow and give NA. Spark also has no factor type, so the data cannot carry a custom level order: groups and levels follow their sorted order, which is the default order in stddiff.

Value

A numeric matrix with one row per category level and columns:

- p.c: Proportion in control group
- p.t: Proportion in treatment group
- missing.c: Number of missing values in control group (first row only)
- missing.t: Number of missing values in treatment group (first row only)
- stddiff: Standardized difference (first row only)
- stddiff.l: Lower CI bound (first row only)
- stddiff.u: Upper CI bound (first row only)

Row names are formatted as "variable_name level_name". A variable with no non-missing values gets one row, named after the variable.

See Also

stddiff.binary, stddiff.numeric

Examples

```
sc <- sparklyr::spark_connect(master = "local")

spark_df <- sparklyr::copy_to(sc, as.data.frame(Titanic))

result <- stddiff.category(
  data = spark_df,
  gcol = 4,   # column index of grouping variable
  vcol = c(1) # columns of categorical variables
)

sparklyr::spark_disconnect(sc)
```

`stddiff.numeric` *Compute Standardized Differences for Numeric Variables (Spark)*

Description

Calculates standardized differences for continuous numeric variables using a Spark DataFrame. Equivalent to `stddiff::stddiff.numeric` but operates on Spark data.

Usage

```
stddiff.numeric(data, gcol, vcol, verbose = FALSE)
```

Arguments

`data`	A Spark DataFrame (`tbl_spark`) containing the variables.
`gcol`	Integer; column index of the binary grouping variable.
`vcol`	Integer vector; column indices of the numeric variables to analyze. Repeated indices give repeated rows, and `gcol` may be included like any other column, as in `stddiff`.
`verbose`	Logical; if TRUE, prints progress messages. Default is FALSE.

Details

The standardized difference for continuous variables is computed as:

$$d = \frac{|\bar{x}_t - \bar{x}_c|}{\sqrt{(s_t^2 + s_c^2)/2}}$$

where $\bar{x}$ represents means and s^2 represents variances.

Differences from `stddiff` on Spark data: where `stddiff` stops with an error for a variable, as when it has no non-missing values, that variable gets `NA` with a warning and the other variables still get results. Local data frames are passed to `stddiff`, so the error remains there. The group column must have exactly two values, or the function stops with an error. `stddiff` doesn't check this, and

with more than two groups it silently compares the first two. Strings are converted as `as.numeric()` converts decimals, `Inf` and `NaN`, but unusual formats it also reads, such as hexadecimal (`"0x1A"`), become `NA` with its coercion warning. With more than about 46,000 rows per group, the confidence limits are computed where `stddiff`'s integer counts overflow and give `NA`. Spark also has no factor type, so the data cannot carry a custom level order: groups follow their sorted order, which is the default order in `stddiff`.

Value

A numeric matrix with one row per variable and columns:

- `mean.c`: Mean in control group
- `sd.c`: Standard deviation in control group
- `mean.t`: Mean in treatment group
- `sd.t`: Standard deviation in treatment group
- `missing.c`: Number of missing values in control group
- `missing.t`: Number of missing values in treatment group
- `stddiff`: Standardized difference
- `stddiff.l`: Lower bound of 95% confidence interval
- `stddiff.u`: Upper bound of 95% confidence interval

See Also

`stddiff.binary`, `stddiff.category`

Examples

```
sc <- sparklyr::spark_connect(master = "local")

spark_df <- sparklyr::copy_to(sc, mtcars)

result <- stddiff.numeric(
  data = spark_df,
  gcol = 8,          # column index of grouping variable
  vcol = c(1, 2, 5)  # columns of numeric variables
)

sparklyr::spark_disconnect(sc)
```

Index

stddiff.binary, 2, *4*, *6*
stddiff.category, *3*, 3, *6*
stddiff.numeric, *3*, *4*, 5