Package: cABCanalysis (via r-universe)

October 11, 2026

Type Package

Title Computed ABC Analysis

Version 1.0.2

Description Identify the most relative data points by dividing a numeric data set into three classes A, B, and C, where class A items are the ``import few", class C items are the ``trivial many" with class B items being something in between, resembling the idea of the Pareto principle. This ABC classification is done using an ABC curve, which plots cumulative ``Yield" against ``Effort", similar to a Lorenz curve. Class borders are then precisely mathematically defined on that curve, aiding in interpretation. Based on: Ultsch A, Lotsch J (2015) ``Computed ABC Analysis for rational Selection of most informative Variables in multivariate Data". PLoS ONE 10(6): e0129767. <doi:10.1371/journal.pone.0129767>.

Depends R (>= 2.10.0)

Imports ggplot2, plotrix, grDevices, graphics, stats, utils

LazyData true

Suggests datasets, testthat (>= 3.0.0)

License GPL-3

URL https://github.com/AndreHDev/cABC_Analysis

Encoding UTF-8

Date 2026-08-27

RoxygenNote 7.3.3

NeedsCompilation no

Config/testthat/edition 3

Repository https://andrehdev.r-universe.dev

Date/Publication 2026-09-11 10:52:14 UTC

RemoteUrl https://github.com/andrehdev/cabc_analysis

RemoteRef HEAD

RemoteSha 7d8bcc9525491819095821530507755 0ee1eaf37

Contents

cABC_analysis 2
cABC_plot_style 4
SwissInhabitants 6

Index **7**

`cABC_analysis` *ABC Classification*

Description

Divides a numeric dataset into three classes (A, B, and C) using ABC analysis. The classification is based on geometric properties of the ABC curve and identifies regions of high, balanced, and low efficiency. Class interpretation:

A: Low effort, high yield (Pareto items)
B: Balanced effort and yield
C: High effort, low yield (submarginal items)

Usage

```
cABC_analysis(Data, PlotIt = FALSE, useGGPlot = TRUE, plotArgs = NULL)
```

Arguments

`Data`	Positive numeric vector which is not uniformly distributed. If matrix or dataframe then the first column will be used.
`PlotIt`	Logical. If `TRUE`, an ABC plot is generated.
`useGGPlot`	Logical, default `TRUE`. If `TRUE` a ggplot2 plot is produced; if `FALSE` a base-R plot is produced. Only relevant when `PlotIt = TRUE`.
`plotArgs`	`NULL` (default), or a named list of cosmetic plot options controlling colors, line width, title, and more. See `cABC_plot_style` for the full list of supported options. Only used when `PlotIt = TRUE` and `useGGPlot = TRUE`. Can be build with `cABC_plot_style` for autocomplete and validation, e.g. `cABC_plot_style(LineWidth = 2, CurveColor = "darkred")`, or passed as a raw named list with the same element names. Any option not specified falls back to the package default.

Details

Calculation of Boundaries is done on the ABC Curve (see `cABC_curve`) with:

Pareto Point:	The point with minimal distance to (0,1) -> A|B Boundary
Breakeven Point:	The point where slope equals to 1
Juren Point:	The point with minimal distance to (BreakevenPoint_x,1) -> B|C Boundary

For more calculation details see: Ultsch A, Lotsch J (2015) "Computed ABC Analysis for rational Selection of most informative Variables in multivariate Data". PLoS ONE 10(6): e0129767. <doi:10.1371/journal.pone.0129767>.

Data cleaning: Before classification, non-numeric values and NAs are coerced to `0`, negative values are set to `0`. A warning is issued when items are converted. If a matrix or data frame is supplied, only the first column is used.

Degenerate inputs (single point, all-identical values, very small datasets) are caught before curve fitting, see `cABC_handle_specials` for the full behavior. Boundary duplicate values that span two classes after classification are resolved by `cABC_postprocess_classes`. In both cases a warning is issued when a special case is triggered.

Value

A list containing:

Aind, Bind, Cind Integer vectors of indices (into the original `Data`) for items assigned to classes A, B, and C respectively. In special-case returns (single point or all-identical), only `Aind` is populated; `Bind` and `Cind` are `integer(0)`.

ABexchanged Logical; `TRUE` if the Pareto point and Break-even point were swapped to maintain coordinate logic (i.e. the Break-even point was to the left of the Pareto point on the curve).

A, B, C `c(x, y)` coordinates for the Pareto point (A), the Break-even point (B), and the Sub-marginal point (C). `NULL` in special-case returns.

smallestAData Cumulative yield at the boundary of Class A. `NULL` in special-case returns.

smallestBData Cumulative yield at the boundary of Class B. `NULL` in special-case returns.

AlimitIndInInterpolation Index of the A boundary in the interpolated `[p, ABC]` curve. `NULL` in special-case returns.

BlimitIndInInterpolation Index of the C boundary in the interpolated `[p, ABC]` curve. `NULL` in special-case returns.

Plot Generated plot object

p Numeric vector of effort values (x-axis) of the interpolation curve. `NULL` in special-case returns.

ABC Numeric vector of yield values (y-axis) of the interpolation curve. `NULL` in special-case returns.

ABLimit Data value closest to the threshold separating Class A from Class B. `NULL` in special-case returns.

BCLimit Data value closest to the threshold separating Class B from Class C. `NULL` in special-case returns.

Author(s)

André Himmelspach (01/2026)

Examples

```
data("SwissInhabitants")
# Set plotting options
opts <- cABC_plot_style(LineWidth = 1.5, CurveColor = "darkblue")

abc <- cABC_analysis(SwissInhabitants, PlotIt = TRUE, plotArgs = opts)

# Extract the data belonging to each class
A <- abc$Aind; B <- abc$Bind; C <- abc$Cind
Agroup <- SwissInhabitants[A]
Bgroup <- SwissInhabitants[B]
Cgroup <- SwissInhabitants[C]
```

---

cABC_plot_style *ABC Plot style*

Description

Constructs a validated list of cosmetic options for customizing ABC plots produced by `cABC_analysis`. Pass the result to `cABC_analysis(..., plot_args = cABC_plot_style(...))`.

Usage

```
cABC_plot_style(
  LineWidth = NULL,
  ShowUniform = NULL,
  ShowBoundary = NULL,
  Plot_title = NULL,
  UniformColor = NULL,
  IdentityColor = NULL,
  EquilibriumColor = NULL,
  CurveColor = NULL,
  PointColor = NULL,
  ABoundaryColor = NULL,
  BBoundaryColor = NULL,
  CBoundaryColor = NULL,
  BoundaryLineColor = NULL,
  LabelColor = NULL,
  LegendTextSize = NULL,
  LegendX = NULL,
  LegendY = NULL,
  LegendSpacing = NULL,
  Theme = NULL
)
```

Arguments

LineWidth	Numeric. Line width of the main ABC curve. Default is 1.25.
ShowUniform	Logical. Whether to draw the uniform reference curve. Default is TRUE.
ShowBoundary	Logical. Whether to draw lines indicating boundaries. Default is TRUE.
Plot_title	Character. Title of the plot. Default is "ABC plot".
UniformColor	Color of the uniform reference curve. Any valid R color specification, e.g. a color name ("green"), a hex code ("#00FF00"), or a palette index (colors()[26]). Default is "green".
IdentityColor	Color of the identity line (y = x). Any valid R color specification (see UniformColor). Default is grDevices::colors()[452].
EquilibriumColor	Color of the equilibrium diagonal (y = 1 - x). Any valid R color specification (see UniformColor). Default is grDevices::colors()[175].
CurveColor	Color of the ABC curve itself. Any valid R color specification (see UniformColor). Default is "blue".
PointColor	Color of individual data points (shown when n < 20). Any valid R color specification (see UniformColor). Default is "blue".
ABoundaryColor	Color of the A|B boundary star and label. Any valid R color specification (see UniformColor). Default is "red".
BBoundaryColor	Color of the B boundary star. Any valid R color specification (see UniformColor). Default is "green".
CBoundaryColor	Color of the B|C boundary star and label. Any valid R color specification (see UniformColor). Default is "blue".
BoundaryLineColor	Color of the orthogonal boundary guide lines. Any valid R color specification (see UniformColor). Default is "red".
LabelColor	Color of the A/B/C set labels and counts. Any valid R color specification (see UniformColor). Default is "red".
LegendTextSize	Numeric. Font size of the legend text. Default is 4.
LegendX	Numeric. X-position of the legend. Default is 0.80.
LegendY	Numeric, or NULL (default). Y-position where the legend starts. If NULL, this is calculated automatically based on the A/C boundary positions to avoid overlapping the plot.
LegendSpacing	Numeric. Vertical spacing between legend entries. Default is 0.05.
Theme	A ggplot2 theme object (e.g. ggplot2::theme_minimal()). Default is ggplot2::theme_light() with a few structural tweaks applied on top (no panel grid, a bordered panel, and a centered plot title).

Details

Any option left at NULL falls back to the package default cABC_default_plot_args.

Value

A named list of plot options, suitable for the `plot_args` argument of `cABC_analysis`.

See Also

`cABC_analysis`

Examples

```
opts <- cABC_plot_style(LineWidth = 2, CurveColor = "darkred")
data("SwissInhabitants")
abc <- cABC_analysis(SwissInhabitants, PlotIt = TRUE, plotArgs = opts)
```

SwissInhabitants *SwissInhabitants in 1900*

Description

Number of inhabitants in the 2896 villages of Switzerland in the year 1900.

Usage

```
data("SwissInhabitants")
```

Format

A numeric vector of length 2896 containing the population counts.

Details

This data set consists of the number of inhabitants in the 2896 communes (cities and villages) in Switzerland in 1900. The data is unordered for anonymity reasons.

Source

Schuler, M., Ullmann, D. (2002). Eidgenossische Volkszahlung: Bevoelkerungsentwicklung der Gemeinden. Bundesamt fur Statistik, Neuchatel, Switzerland.

References

Behnisch, M., Ultsch, A. (2010). Population Patterns in Switzerland 1850-2000. In: Gaul, W. et al (Eds), Advances in Data Analysis, Data Handling and Business Intelligence, Springer, Heidelberg, pp. 163-173.

Examples

```
data(SwissInhabitants)
summary(SwissInhabitants)
```

Index

∗ **datasets**
SwissInhabitants, 6

cABC_analysis, 2, *4*, *6*
cABC_curve, *2*
cABC_default_plot_args, *5*
cABC_handle_specials, *3*
cABC_plot_style, *2*, 4
cABC_postprocess_classes, *3*

SwissInhabitants, 6
SwissInhabitants1900 *(*SwissInhabitants*)*, 6