

# Package: cABCanalysis (via r-universe)

June 5, 2026

**Type** Package

**Title** Computed ABC Analysis

**Version** 1.0

**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](https://doi.org/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](https://github.com/AndreHDev/cABC_Analysis)

**Encoding** UTF-8

**Date** 2026-04-20

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Config/testthat/edition** 3

**Repository** <https://andrehdev.r-universe.dev>

**Date/Publication** 2026-05-06 14:35:15 UTC

**RemoteUrl** [https://github.com/andrehdev/cabc\\_analysis](https://github.com/andrehdev/cabc_analysis)

**RemoteRef** HEAD

**RemoteSha** ac00c13047549c7829382594d68f9659611c45d0

## Contents

cABC_analysis . . . . .	2
SwissInhabitants . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

cABC_analysis	<i>ABC Classification</i>
---------------	---------------------------

---

### 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)
```

### 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.

### 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.

**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")
abc <- cABC_analysis(SwissInhabitants, PlotIt = TRUE)

# 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]
```

---

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 Volkszählung: Bevoelkerungsentwicklung der Gemeinden. Bundesamt für 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, [4](#)

cABC\_analysis, [2](#)

cABC\_curve, [2](#)

cABC\_handle\_specials, [2](#)

cABC\_postprocess\_classes, [2](#)

SwissInhabitants, [4](#)

SwissInhabitants1900

(SwissInhabitants), [4](#)