

Package: oneclust (via r-universe)

September 26, 2026

Type Package

Title Maximum Homogeneity Clustering for Univariate Data

Version 0.3.0

Description Maximum homogeneity clustering algorithm for one-dimensional data described in W. D. Fisher (1958) [doi:10.1080/01621459.1958.10501479](https://doi.org/10.1080/01621459.1958.10501479) via dynamic programming.

License GPL-3

URL <https://nanx.me/oneclust/>, <https://github.com/nanxstats/oneclust>

Encoding UTF-8

VignetteBuilder knitr

BugReports <https://github.com/nanxstats/oneclust/issues>

Depends R (>= 3.5.0)

LinkingTo Rcpp

Imports Rcpp

Suggests knitr, rmarkdown

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository <https://nanxstats.r-universe.dev>

Date/Publication 2025-08-29 12:47:30 UTC

RemoteUrl <https://github.com/nanxstats/oneclust>

RemoteRef HEAD

RemoteSha 121505b16462d5bbbaa64c366d86e41d84245c19

Contents

cud	2
oneclust	2
sim_postcode_levels	3
sim_postcode_samples	4

Index	5
--------------	----------

 cud

Masataka Okabe and Kei Ito's Color Universal Design palette

Description

Masataka Okabe and Kei Ito's Color Universal Design palette

Usage

```
cud(x, shift = TRUE, reverse = FALSE)
```

Arguments

x	Vector, color index.
shift	Start from the second color in the CUD palette?
reverse	Reverse the color order?

Value

A vector of color hex values.

Examples

```
barplot(rep(1, 7), col = cud(1:7))
barplot(rep(1, 8), col = cud(1:8, shift = FALSE))
barplot(rep(1, 8), col = cud(1:8, shift = FALSE, reverse = TRUE))
```

 oneclust

Maximum homogeneity clustering for one-dimensional data

Description

Maximum homogeneity clustering for one-dimensional data

Usage

```
oneclust(x, k, w = NULL, sort = TRUE)
```

Arguments

x	Numeric vector, samples to be clustered.
k	Integer, number of clusters.
w	Numeric vector, sample weights (optional). Note that the weights here should be sampling weights (for example, a certain proportion of the population), not frequency weights (for example, number of occurrences).
sort	Should we sort x (and w) before clustering? Default is TRUE. Otherwise the order of the data is respected.

Value

A list containing:

- cluster - cluster id of each sample.
- cut - index of the optimal cut points.

References

Fisher, Walter D. 1958. On Grouping for Maximum Homogeneity. *Journal of the American Statistical Association* 53 (284): 789–98.

Examples

```
set.seed(42)
x <- sample(c(
  rnorm(50, sd = 0.2),
  rnorm(50, mean = 1, sd = 0.3),
  rnorm(100, mean = -1, sd = 0.25)
))
oneclust(x, 3)
```

sim_postcode_levels	<i>Simulate the levels and their sizes in a high-cardinality feature</i>
---------------------	--

Description

Simulate the levels and their sizes in a high-cardinality feature

Usage

```
sim_postcode_levels(nlevels = 100L, seed = 1001)
```

Arguments

nlevels	Number of levels to generate.
seed	Random seed.

Value

A data frame of postal codes and sizes.

Note

The code is derived from the example described in the "rare levels" vignette in the vtreat package.

Examples

```
df_levels <- sim_postcode_levels(nlevels = 500, seed = 42)
head(df_levels)
```

sim_postcode_samples *Simulate a high-cardinality feature and a binary response*

Description

Simulate a high-cardinality feature and a binary response

Usage

```
sim_postcode_samples(  
  df_levels,  
  n = 2000L,  
  threshold = 1000,  
  prob = c(0.3, 0.1),  
  seed = 1001  
)
```

Arguments

df_levels	Number of levels.
n	Number of samples.
threshold	The threshold for determining if a postal code is rare.
prob	Occurrence probability vector of the class 1 event in rare and non-rare postal codes.
seed	Random seed.

Value

A data frame of samples with postal codes, response labels, and level rarity status.

Note

The code is derived from the example described in the "rare levels" vignette in the vtreat package.

Examples

```
df_levels <- sim_postcode_levels(nlevels = 500, seed = 42)  
df_postcode <- sim_postcode_samples(  
  df_levels,  
  n = 10000, threshold = 3000, prob = c(0.2, 0.1), seed = 43  
)  
head(df_postcode)
```

Index

`cud`, [2](#)

`oneclust`, [2](#)

`sim_postcode_levels`, [3](#)

`sim_postcode_samples`, [4](#)