Package ‘ccoptimalmatch’

October 12, 2022

Type Package
Title Implementation of Case-Control Optimal Matching
Version 0.1.0
Description Cases are matched to controls in an efficient, optimal and computationally flexible way. It uses the idea of sub-sampling in the level of the case, by creating pseudo-observations of controls. The user can select between replacement and without replacement, the number of controls, and several covariates to match upon. See Mamouris (2021) <doi:10.1186/s12874-021-01256-3> for an overview.
Depends R (>= 2.10)
License GPL-2
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports dplyr, rlang
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Pavlos Mamouris [aut, cre], Vahid Nassiri [aut, ctb]
Maintainer Pavlos Mamouris <pavlos.mamouris@kuleuven.be>
Repository CRAN
Date/Publication 2021-04-21 07:40:10 UTC

R topics documented:

being_processed .................................................. 2
ccoptimalmatch .................................................. 2
not_processed ..................................................... 3
optimal_matching ............................................... 3

Index 5
### being_processed  
*Data for matching cases with controls*

**Description**
A dataset containing cases and controls using the Intego registry data. The variables are as follows:

**Usage**
```r
data(being_processed)
```

**Format**
A data frame with 77110 rows and 11 variables

**Details**
- `cluster_case`: each case forms a cluster with all possible controls to be matched
- `Patient_Id`: Unique identifier for each patient
- `case_control`: binary, if case==Colorectal Cancer, else control
- `case_ind`: binary, if 1==case, else control
- `JCG`: Year of Contact
- `entry_year`: the year that the patient first entered the database
- `CI`: Comorbidity Index. Count of chronic diseases before index data
- `age_diff`: difference of age between cases and controls
- `fup_diff`: difference of follow-up between cases and controls
- `total_control_per_case`: total controls that are available to be pooled per case
- `freq_of_controls`: how many times the control is available to be matched for different cases

### ccoptimalmatch  
*ccoptimalmatch: Optimal Case Control matching*

**Description**
Fast and optimal matching for cases and controls

**Author(s)**
- **Maintainer**: Pavlos Mamouris <pavlos.mamouris@kuleuven.be>
- Authors:
  - Vahid Nassiri <vahid.nassiri@openanalytics.eu> [contributor]
Description

A dataset containing cases and controls using the Intego registry data. But not the final dataset. The variables are as follows:

Usage

data(not_processed)

Format

A data frame with 656506 rows and 9 variables

Details

- Patient_Id: Unique identifier for each patient
- JCG: Year of Contact
- Birth_Year: Patient’s year of birth
- Gender: Patient’s Gender
- Practice_Id: Patient’s general practice
- case_control: binary, if case==Colorectal Cancer, else control
- entry_year: the year that the patient first entered the database
- fup_diff: difference of follow-up between cases and controls
- CI: Comorbidity Index. Count of chronic diseases before index data

Description

optimal_matching is performing the optimal match between cases and controls in an iterative way and computational efficient way
Usage

optimal_matching(
    total_database,
    n_con,
    cluster_var,
    Id_Patient,
    total_cont_per_case,
    case_control,
    with_replacement = FALSE
)

Arguments

total_database  a data frame that contains the cases and controls
n_con           number of controls to be matched
cluster_var     a variable that contains one case with all available controls to be pooled
Id_Patient      Id of the patient
total_cont_per_case
    total number of controls that are available for each case
case_control    a variable containing "case" and "control"
with_replacement Use replacement or not

Details

Here is where I should put all my details. This is where I should give more examples if necessary

Value

a data frame containing the cases and the corresponding number of controls

Examples

optimal_matching(being_processed, n_con=2, cluster_var=cluster_case,
    Id_Patient=Patient_Id, total_cont_per_case=total_control_per_case, case_control = case_control)
Index

* datasets
  - being_processed, 2
  - not_processed, 3

being_processed, 2

ccoptimalmatch, 2
ccoptimalmatch-package
  - (ccoptimalmatch), 2

not_processed, 3

optimal_matching, 3