Package 'medhist'

January 29, 2021

Type Package

Title A preprocessor to construct medical history table from data source

Version 0.1.0

Date 2021-01-21

Depends R (>= 4.0.2)

Description This package construct a medical-history table from several tables
of an electronic medical record database. The medical-history table may be
utilized for both causal and predictive modeling.

License GPL-3

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports devtools,
tidyverse,
pbapply,
parallel,
BiocGenerics,
lubridate,
Biobase,
broom,
survival,
imputeTS,
caret,
MLeval

Suggests BiocStyle,
knitr,
rmarkdown,
kableExtra,
dslabs,
magick,
testthat

URL https://github.com/herdiantrisufriyana/medhist

BugReports https://github.com/herdiantrisufriyana/medhist/issues

VignetteBuilder knitr

1

R topics documented:

---

annotation   *Annotation of diagnosis/procedure code.*

Description

A dataset containing the annotation of diagnosis/procedure code.

Usage

    annotation

Format

A data frame with 12,846 rows and 2 columns:

code ICD-10 code of a diagnosis/procedure (2016). This connects to `diagnosis` data.

desc A character describing diagnosis/procedure.

Source

Artificial data

---

compile_mh_outcome   *Construct unified data as a TidySet*

Description

This function compiles medical history and the outcome table into a TidySet (i.e. ExpressionSet). One example of the output can be called by `data(medhistdata)`.

Usage

```
compile_mh_outcome(mh_table, outcome)
```

Arguments

`mh_table`	Target population data, a data frame with rows for visits and standardized columns (please see Details below). This is an output of `extract_medical_history()`.
`outcome`	Subject list, a data frame with rows for unique subjects and columns of `subject_id`, `latest_date`, and `outcome`. The last column is a factor of which non-event is the first class between non-event and event.

Details

Target population data consisted `visit_id`,`subject_id`, `healthcare_id`, `admission_date`, and `db_start_date` beyond columns for medical histories. Columns of `visit_id`,`subject_id`, and `healthcare_id` are characters of identification numbers idenitfying respectively unique visits, subjects, and healthcare providers. Columns of `admission_date` and `db_start_date` are dates of subject admission in a visit and database start of recording period, respectively. The remaining columns are named using ICD-10 code for either diagnosis or procedure regardless the number of digit or k-mer. Each code are spread as a column consisting the number of days from the latest `admission_date` on which this code have ever been encountered to `admission_date` of each visit. If no encounter is found, then the value returns NA to denote censored data. This is because the code may be encountered before `db_start_date`. If the latest date and that of each visit are the same, then the value returns 0.

Value

A TidySet (i.e. ExpressionSet) containing the visits of subjects in medical history and the outcome datasets.

Examples

```
## Create input example
data(visit_cap)
data(visit_ffs)
data(visit_drg)
data(diagnosis)

population=
  list(visit_cap,visit_ffs,visit_drg) %>%
  lapply(select,visit_id,subject_id,healthcare_id,admission_date) %>%
  do.call(rbind,.) %>%
```

```
    left_join(diagnosis,by='visit_id') %>%
    filter(!code_type%in%c('Admission diagnosis')) %>%
    select(-code_type) %>%
    mutate(db_start_date=as.Date('2015-01-01')) %>%
    .[!duplicated(.),]

## Extract outcome of subjects and sample some of them
outcome=
    extract_outcome(population,'O1[4-5]',first,-1,'Z3[3-7]',last,0) %>%
    group_by(outcome) %>%
    slice(sample(seq(n()),ceiling(n()*0.0125),F)) %>%
    ungroup()

## Filter medical history before the date of either event or non-event
input=
    outcome %>%
    right_join(population,by='subject_id') %>%
    select(visit_id, everything()) %>%
    filter(admission_date<latest_date) %>%
    select(-outcome,-latest_date)

## Extract medical history of subjects per healthcare provider
mh_table=extract_medical_history(input)

## Construct unified data as a TidySet
medhisdata=compile_mh_outcome(mh_table,outcome,'ICD-10 (2016)')
```

---

diagnosis   *Diagnosis and procedure.*

Description

A dataset containing the diagnoses and procedures (if any) Every visit is encountered at least a code of discharge, primary diagnosis. But, a visit may be encountered 1 or more other code of diagnoses/procedures.

Usage

```
diagnosis
```

Format

A data frame with 4,663,453 rows and 3 columns:

visit_id Visit ID. Each ID is unique to one episode (from admission to discharge) per provider. This connects to visit_cap, visit_ffs, and visit_drg data.

code ICD-10 code of a diagnosis/procedure (2016). This connects to annotation data.

code_type Categorical variable of code type, consisting: (1) Admission diagnosis; (2) Discharge,primary diagnosis; (3) Discharge,secondary diagnosis; and (4) Procedure.

Source

Artificial data

edata	*Experimenter information for medical history datasets with specific outcome.*

Description

Experimenter information for experiment data table.

Usage

```
edata
```

Format

A MIAME object with 6 attributes:

name A character of the experimenter name.

lab A character of the laboratory name.

contact A character of the experimenter email.

title A character of the dataset title.

abstract A character of the dataset abstract.

url A character of the dataset URL.

Source

Artificial data

extract_medical_history	
	Extract medical history of subjects per healthcare provider

Description

This function spread medical histories into different columns.

Usage

```
extract_medical_history(population, n_batch = 50, cl = 1)
```

Arguments

population	Target population data, a data frame with rows for visits and standardized columns (please see Details below). If only a single healthcare provider, then make `healthcare_id` consisting only a single identification number.
n_batch	Number of batch, a non-negative integer of number of batch in which a number of subjects per provider are grouped. This is intended to ensure the computation fits to the memory.
cl	Parallel cluster, a non-negative integer of number of CPU cluster used for computation in parallel. Set to 1 if no parallelism is expected.

Details

Target population data consisted visit_id,subject_id, healthcare_id, admission_date, code, and db_start_date. Columns of visit_id,subject_id, and healthcare_id are characters of identification numbers identifying respectively unique visits, subjects, and healthcare providers. Columns of admission_date and db_start_date are dates of subject admission in a visit and database start of recording period, respectively. Column of code is a character of ICD-10 code for either diagnosis or procedure regardless the number of digit or k-mer.

Value

Target population data, a data frame with rows for visits and standardized columns (please see Details above), except code. Each code are spread as a column consisting the number of days from the latest admission_date on which this code have ever been encountered to admission_date of each visit. If no encounter is found, then the value returns NA to denote censored data. This is because the code may be encountered before db_start_date. If the latest date and that of each visit are the same, then the value returns 0.

Examples

```
## Create input example
data(visit_cap)
data(visit_ffs)
data(visit_drg)
data(diagnosis)

population=
  list(visit_cap,visit_ffs,visit_drg) %>%
  lapply(select,visit_id,subject_id,healthcare_id,admission_date) %>%
  do.call(rbind,.) %>%
  left_join(diagnosis,by='visit_id') %>%
  filter(!code_type%in%c('Admission diagnosis')) %>%
  select(-code_type) %>%
  mutate(db_start_date=as.Date('2015-01-01')) %>%
  .[!duplicated(.),]

## Extract outcome of subjects and sample some of them
outcome=
  extract_outcome(population,'O1[4-5]',first,-1,'Z3[3-7]',last,0) %>%
  group_by(outcome) %>%
  slice(sample(seq(n()),ceiling(n()*0.0125),F)) %>%
  ungroup()

## Filter medical history before the date of either event or non-event
input=
  outcome %>%
  right_join(population,by='subject_id') %>%
  select(visit_id, everything()) %>%
  filter(admission_date<latest_date) %>%
  select(-outcome,-latest_date)

## Extract medical history of subjects per healthcare provider
mh_table=extract_medical_history(input)
```

---

extract_nps_mh *Extract medical history without perfect separation*

Description

This function makes a list of medical histories without perfect separation. This means a medical history per outcome have non-zero variance.

Usage

```
extract_nps_mh(tidy_set, sd_cutoff = 0)
```

Arguments

tidy_set   A TidySet (i.e. ExpressionSet) containing the visits of subjects in outcome dataset, paid by any payment systems. This TidySet also accomodates outcome dataset. This is an output of `compile_mh_outcome()`.

sd_cutoff   Cutoff of standard deviation, a non-negative numeric that is used for cutoff. If standard deviation of a medical history metric for an outcome is more than the cutoff for that outcome, either non-event or event, then that medical history is accounted to the list.

Value

Medical history list, a data frame with rows for unique medical history and columns of key, nonevent and event. The last two columns are standard deviation of the medical history for non-event and event. Missing value or NA is not accounted when computing the standard deviation.

Examples

```
## Create input example
data(medhistdata)

## Extract medical history without perfect separation
ps_remover=extract_nps_mh(medhistdata)
```

---

extract_nzv_mh *Extract medical history without zero variance*

Description

This function makes a list of medical histories without zero variance.

Usage

```
extract_nzv_mh(tidy_set, sd_cutoff = 0)
```

Arguments

tidy_set   A TidySet (i.e. ExpressionSet) containing the visits of subjects in outcome
                  dataset, paid by any payment systems. This TidySet also accomodates outcome
                  dataset. This is an output of `compile_mh_outcome()`.

sd_cutoff   Cutoff of standard deviation, a non-negative numeric that is used for cutoff. If
                  standard deviation of a medical history metric is more than the cutoff, then that
                  medical history is accounted to the list.

Value

Medical history list, a data frame with rows for unique medical history and columns of `key` and
`sd_value`. Missing value or NA is not accounted when computing the standard deviation.

Examples

```
## Create input example
data(medhistdata)

## Extract medical history without zero variance
zv_remover=extract_nzv_mh(medhistdata)
```

---

extract_outcome   *Extract outcome of subjects*

Description

This function makes a list of subjects with either event or non-event outcome, that visit healthcare
providers up to the predefined latest date.

Usage

```
extract_outcome(
  population,
  icd10_event,
  latest_event = first,
  day_to_event = -1,
  icd10_nonevent,
  latest_nonevent = last,
  day_to_nonevent = 0,
  verbose = T
)
```

Arguments

population   Target population data, a data frame with rows for visits and standardized columns
                  (please see Details below). If only a single healthcare provider, then make
                  `healthcare_id` consisting only a single identification number.

icd10_event	Event code, a character of ICD-10 code for either diagnosis or procedure, that identify subjects with event outcome. This parameter should be a regular expression if detecting >1 ICD-10 codes. The detection is not exact but anywhere within the character sequence; thus, the code with more than k-mer will be extracted.
latest_event	Latest date of event, a function to choose `admission_date` with the `icd10_event` among many for each subject. This date will be adjusted with `day_to_event` to limit visits accounted for medical histories, since these are considerably before the outcome. The function is conceivably `first()` or `last()` from dplyr package, preferably `first()`. This may be respectively `min()` or `max()` if using base R.
day_to_event	Day to event, an integer to adjust `latest_event` for limiting visits accounted for medical histories. These are considerably before the outcome, i.e. up to -1 day.
icd10_nonevent	Non-event code, a character of ICD-10 code for either diagnosis or procedure, that identify subjects with non-event outcome. This parameter should be a regular expression if detecting >1 ICD-10 codes. Make this parameter as `''` (character of none) if all except event codes are used. The detection is not exact but anywhere within the character sequence; thus, the code with more than k-mer will be extracted.
latest_nonevent	
	Latest date of non-event, a function to choose `admission_date` with the `icd10_nonevent` among many for each subject. This date will be adjusted with `day_to_nonevent` to limit visits accounted for medical histories, since these are considerably up to the day of the outcome. The function is conceivably `first()` or `last()` from dplyr package, preferably `last()`. This may be respectively `min()` or `max()` if using base R.
day_to_nonevent	
	Day to non-event, an integer to adjust `latest_nonevent` for limiting visits accounted for medical histories. These are considerably up to the day of the outcome, i.e. up to 0 day.
verbose	Verbosity, a logical indicating whether progress should be shown.

Details

Target population data consisted `visit_id,subject_id`, `healthcare_id`, `admission_date`, `code`, and `db_start_date`. Columns of `visit_id,subject_id`, and `healthcare_id` are characters of identification numbers identifying respectively unique visits, subjects, and healthcare providers. Columns of `admission_date` and `db_start_date` are dates of subject admission in a visit and database start of recording period, respectively. Column of `code` is a character of ICD-10 code for either diagnosis or procedure regardless the number of digit or k-mer.

Value

Subject list, a data frame with rows for unique subjects and columns of `subject_id`, `latest_date`, and `outcome`. The last column is a factor of which non-event is the first class between non-event and event.

Examples

```
## Create input example
```

```
data(visit_cap)
data(visit_ffs)
data(visit_drg)
data(diagnosis)

population=
  list(visit_cap,visit_ffs,visit_drg) %>%
  lapply(select,visit_id,subject_id,healthcare_id,admission_date) %>%
  do.call(rbind,.) %>%
  left_join(diagnosis,by='visit_id') %>%
  filter(!code_type%in%c('Admission diagnosis')) %>%
  select(-code_type) %>%
  mutate(db_start_date=as.Date('2015-01-01')) %>%
  .[!duplicated(.),]

## Extract outcome of subjects
outcome=extract_outcome(population,'O1[4-5]',first,-1,'Z3[3-7]',last,0)
```

---

healthcare   *Healthcare providers.*

Description

A dataset containing the healthcare providers. Subjects are sampled from all available providers in all of the countries/cities, weighted by household numbers in a provider and subject numbers in a household. Based on insurance utilization, there are three categories of household. Category 1 refers to a household that all of the subjects never utilize the insurance. Category 2 refers to a household that all or some of the subjects have ever utilized the insurance for primary care only. Category 3 refers to a household that all or some of the subjects have ever utilized the insurance for primary and secondary/tertiary care. In this table, sample_cat implicitly refers to these categories but not the same.

Usage

```
healthcare
```

Format

A data frame with 237,717 rows and 5 columns:

healthcare_id Provider ID. This connects to subject, visit_cap, visit_ffs, and visit_drg data.

sample_cat Category of this provider. Category 1 and 2 refer to primary care. Category 3 refers to secondary/tertiary care.

sampling_weight Weight that determines how likely a subject sample taken from this provider over others.

healthcare_country Country code where this provider is located.

healthcare_city City code where this provider is located.

Source

Artificial data

---

mdata *Metadata for medical history datasets with specific outcome.*

Description

Metadata for phenotype and protocol tables.

Usage

```
mdata
```

Format

A data frame with 7 rows and 1 column:

labelDescription A character to describe the variables.

Source

Artificial data

---

medhistdata *Medical history datasets with specific outcome.*

Format

A data frame with 3,434 rows and 7 columns:

exprs ICD-10 code of a diagnosis/procedure (2016). This connects to `pData()` by columns (visits) and `fData()` by rows (medical histories).

outcome A factor of which non-event is the first class between non-event and event. Subjects with events are those encountered by either O14 or O15 while those with non-events encountered by Z33 to Z37. Called by `pData()`.

desc A character describing diagnosis/procedure. Called by `fData()`.

visit_id Visit ID. Each ID is unique to one episode (from admission to discharge) per provider. Called by pData(`protocolData()`).

subject_id Subject ID. Each ID is unique to one subject across healthcare providers. Called by pData(`protocolData()`).

latest_date Latest date of event/non-event. For event, this is admission_date of subjects at 2 days before the date of event, which is earliest date of either O14 or O15 encounter. For non-event, this is admission_date of subjects at the date of non-event, which is the latest date of Z33 to Z37 encounters. Called by pData(`protocolData()`).

healthcare_id Provider ID. The provider is the one of which the subject visits (not always one the subject registered to). This connects to healthcare data. Called by pData(`protocolData()`).

admission_date Admission date of this visit. Called by pData(`protocolData()`). \itemdb_start_dateStart date of the database. Called by pData(`protocolData()`).

Artificial data

medhistdata

A TidySet (i.e. ExpressionSet) containing the visits of subjects in outcome dataset, paid by any payment systems. This TidySet also accomodates outcome dataset.

datasets

---

subject   *Subject sample (single-payer health insurance client).*

Description

A dataset containing the subjects or clients. Subjects are sampled from all available providers in all of the countries, weighted by household numbers in a provider and subject numbers in a household. Based on insurance utilization, there are three categories of household. Category 1 refers to a household that all of the subjects never utilize the insurance. Category 2 refers to a household that all or some of the subjects have ever utilized the insurance for primary care only. Category 3 refers to a household that all or some of the subjects have ever utilized the insurance for primary and secondary/tertiary care.

Usage

```
subject
```

Format

A data frame with 1,697,452 rows and 11 columns:

subject_id Subject ID. This connects to `healthcare`, `visit_cap`, `visit_ffs`, and `visit_drg` data.

householder_id Householder ID. A household only has one householder. Several subjects may be registered to a household. This ID is the `subject_id` of the householder.

healthcare_id Provider ID. The provider is the one of which this subject registered to (not always one where this subject visits).

birth_date Birth date of this subject.

family_status Categorical variable of family status, consisting `person` and `other` for non-family household, or `husband`, `wife`, and `child` for family household. These may be mixed.

sex Categorical variable of sex, consisting `female`, `male`, and `unspecified`.

marital_status Categorical variable of marital status, consisting `single`, `married`, `divorced/widowed`, and `unspecified`.

insurance_class Categorical variable of insurance class, consisting `first`, `second`, and `third`. The `first` class is the one indicating the highest socioeconomic class compared to the `second` and `third` classes

occupation_segment Categorical variable of occupation segment of the householder (not necessarily this subject). This consists of `central-government-paid` householder, `local-government-paid` householder, `employee` householder (private company), `employer` householder, and `unemployed` householder.

subject_country country code where the provider is located.

subject_city city code where the provider is located.

Source

Artificial data

---

trans_binary	*Transform medical history by binarizing day interval*

Description

This function transforms each medical history from day interval to binary one. If there is no record, then the value is 0; otherwise, the value is 1.

Usage

```
trans_binary(tidy_set, verbose = T)
```

Arguments

tidy_set
: A TidySet (i.e. ExpressionSet) containing the visits of subjects in outcome dataset, paid by any payment systems. This TidySet also accomodates outcome dataset. This is an output of `compile_mh_outcome()`.

verbose
: Verbosity, a logical indicating whether progress should be shown.

Value

A TidySet (i.e. ExpressionSet) containing the visits of subjects in medical history and the outcome datasets. Since a code may be encountered before `db_start_date`, causing the value returns NA to denote censored data, that medical history is transformed to 0. But, if the interval is 0, the value returns 1 because this means the record exists.

Examples

```
## Create input example
data(medhistdata)

## Transform medical history by binarizing day interval
mh_binary=trans_binary(medhistdata)
```

---

trans_hist_rate	*Transform medical history into a rate by utilizing Kaplan-Meier esti-mator*

Description

This function transforms each medical history from day interval to a rate by utilizing Kaplan-Meier estimator in time-to-event analysis. Intuition behind this procedure is that a medical history may be absent because a patient do not visit a healthcare provider although there is a symptom indicating a medical condition. Another situation is a patient visit another provider; thus, a healthcare provider do not have record of a medical history, while the patient do not report (e.g. forget) that medical history to the physician. Therefore, the transformed value is the probability of a code not encountered before t that denotes a time when at least a code encountered once. A censored code encounter is treated as 'not encountered'. This probability is also computed across healthcare providers.

Usage

```
trans_hist_rate(tidy_set, hist_rate = NULL, interpolation = NULL, verbose = T)
```

Arguments

tidy_set   A TidySet (i.e. ExpressionSet) containing the visits of subjects in outcome
                  dataset, paid by any payment systems. This TidySet also accomodates out-
                  come dataset. This is an output of compile_mh_outcome(). Only non-zero
                  variance medical histories are taken. This is applied automatically. No need
                  to use extract_nzv_mh() before plugging data for this function. It does not
                  matter if extract_nps_mh() is applied, since non-perfect separation medical
                  histories are also non-zero variance ones, conceivably.

hist_rate   Historical rate data, a data frame with three columns which are mh for medical
                  history, time for days before the visit, and estimate for historical rate for each
                  medical history at the time.

interpolation   Interpolation method, a character of interpolation method. This accepts follow-
                  ing input: 'none', 'linear', 'spline', and 'stine'. If no interpolation is expected,
                  then set to 'none'; thus, the same rate is applied as the later one. This will show
                  a stair-step appearance in a typical Kaplan-Meier plot. Instead of a flat line
                  among the times of the estimates, interpolation may be applied by linear, spline,
                  and Stineman approximation.

verbose   Verbosity, a logical indicating whether progress should be shown.

Value

A TidySet (i.e. ExpressionSet) containing the visits of subjects in medical history and the outcome
datasets. Since a code may be encountered before db_start_date, the value returns NA to denote
censored data. This is accounted for Kaplan-Meier estimation.

Examples

```
## Create input example
data(medhistdata)

## Transform medical history by normalizing day up to maximum database days
mh_hist_rate=trans_hist_rate(medhistdata)
```

---

trans_norm_day   *Transform medical history by normalizing day up to maximum*
                             database days

Description

This function transforms each medical history from day interval to normalized one based on max-
imum period of database, inversely. For example, if maximum period is 730 days, and a code
medical history is encountered at 365 days at the latest up to a visit by a subject, then this function
will normalize this number into (730-365)/730=0.5. Therefore, normalized values of 0 and 1 are
respectively day intervals of 730 and 0. Missing value or NA day interval is also normalized to 0.

Usage

```
trans_norm_day(tidy_set, day_cutoff, verbose = T)
```

Arguments

`day_cutoff`	Cutoff of the maximum days, a non-negative numeric that is used for cutoff. If day interval of a medical is more than the cutoff, then that medical history is transformed to 0. Missing value or NA is also transformed to 0.
`verbose`	Verbosity, a logical indicating whether progress should be shown.
`mh_table`	Target population data, a data frame with rows for visits and standardized columns (please see Details below). This is an output of `extract_medical_history()`.

Value

A TidySet (i.e. ExpressionSet) containing the visits of subjects in medical history and the outcome datasets. Since a code may be encountered before `db_start_date`, causing the value returns NA to denote censored data, that medical history is transformed to 0.

Examples

```
## Create input example
data(medhistdata)

## Transform medical history by normalizing day up to maximum database days
mh_norm_day=trans_norm_day(medhistdata,730)
```

---

visit_cap	*Visits paid by capitation payment system.*

Description

A dataset containing the visits. All visits of a subject sample are included of which admission dates range from January 1st, 2015 to December 31st, 2016. The discharge date may be up to 2017. In this table, only visits from primary care are included. These visits are not paid. Instead, the provider is monthly paid per subject regardless the number of visits. Therefore, there is no claimed or verified cost.

Usage

```
visit_cap
```

Format

A data frame with 1,733,757 rows and 6 columns:

visit_id Visit ID. Each ID is unique to one episode (from admission to discharge) per provider. This connects to `diagnosis` data.

subject_id Subject ID. This connects to `subject` data.

healthcare_id Provider ID. The provider is the one of which the subject visits (not always one the subject registered to). This connects to `healthcare` data.

admission_date Admission date of this visit.

discharge_date Discharge date of this visit.

discharge_status Categorical variable of discharge status, consisting `discharged to home`, `return for outpatient services`, `transferred within this facility`, `transferred to higher-level care`, `left against medical advice or discontinued care`, `expired`, and `other`.

Source

Artificial data

| visit_drg | *Visits paid by diagnosis-related group payment system.* |

Description

A dataset containing the visits. All visits of a subject sample are included of which admission dates range from January 1st, 2015 to December 31st, 2016. The discharge date may be up to 2017. In this table, only visits from secondary/tertiary care are included. These visits are paid based on diagnosis-related group. Therefore, there is claimed or verified cost.

Usage

```
visit_drg
```

Format

A data frame with 906,905 rows and 8 columns:

visit_id Visit ID. Each ID is unique to one episode (from admission to discharge) per provider. This connects to `diagnosis` data.

subject_id Subject ID. This connects to `subject` data.

healthcare_id Provider ID. The provider is the one of which the subject visits (not always one the subject registered to). This connects to `healthcare` data.

admission_date Admission date of this visit.

discharge_date Discharge date of this visit.

discharge_status Categorical variable of discharge status, consisting `discharged to home`, `transferred to other facility`, `transferred to higher-level care`, `expired`, and `expired,place unknown`.

claimed_cost Amount of cost based on claimed diagnosis-related group, in a cost unit.

verified_cost Amount of cost based on verified diagnosis-related group, in a cost unit.

Source

Artificial data

---

visit_ffs *Visits paid by fee-for-service payment system.*

Description

A dataset containing the visits. All visits of a subject sample are included of which admission dates range from January 1st, 2015 to December 31st, 2016. The discharge date may be up to 2017. In this table, only visits from primary care are included. These visits are paid based on fee for service. Therefore, there is claimed or verified cost.

Usage

```
visit_ffs
```

Format

A data frame with 67,870 rows and 7 columns:

visit_id Visit ID. Each ID is unique to one episode (from admission to discharge) per provider. This connects to `diagnosis` data.

subject_id Subject ID. This connects to `subject` data.

healthcare_id Provider ID. The provider is the one of which the subject visits (not always one the subject registered to). This connects to `healthcare` data.

admission_date Admission date of this visit.

discharge_date Discharge date of this visit.

claimed_cost Amount of cost based on claimed fee for service, in a cost unit.

verified_cost Amount of cost based on verified fee for service, in a cost unit.

Source

Artificial data

Index