Package 'ukbbhelpr'

December 3, 2021

Type Package

Title Helper functions for UK Biobank data

Version 0.2.0

Description A collection of helper functions for working with UK Biobank data.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Depends R (>= 2.10)

Imports ConceptLibraryClient,
data.table,
lubridate,
readxl,
stringr,
checkmate

Remotes github::SwanseaUniversityMedical/ConceptLibraryClient

RoxygenNote 7.1.2

Suggests testthat

R topics documented:

ehr_extract *Extract EHR records and values*

Description

Extracts records/values from linked EHR data. If `values` is TRUE, values are extracted the `value1` field with the exception of data provider 2 where values are extracted from `value2` if `value1` is empty. Units are taken from `value3` for data provider 2 (otherwise units are unavailable). `NA`, zero and duplicate values are dropped.

Usage

```
ehr_extract(ehr_data, read_codes, values = NULL)
```

Arguments

`ehr_data`	Data table with UK Biobank clinical event data i.e. `gp_clinical.txt`.
`read_codes`	Extract records matching these codes. Data table/frame with `read_2` (Read v2) and `read_3` (CTV3) columns.
`values`	Extract values from EHR records. Replaces the `value1`, `value2`, `value3` columns with a single `value` column and a `unit` column. Default is currently TRUE but will change to FALSE in a future release.

Value

Filtered EHR data with `value` and `unit` columns if `values` is TRUE.

ehr_hdr_extract *TODO*

Description

TODO

Usage

```
ehr_hdr_extract(
  ehr_data,
  id,
  version_id = NULL,
  api = NULL,
  values = FALSE,
  format = "raw"
)
```

Arguments

`ehr_data` TODO

`id` Concept ID.

`version_id` Optional concept version ID. Otherwise latest version of concept is returned.

`api` Optional HttpClient to connect to phenotype library (default uses https://phenotypes.healthdatagateway.or

`values` TODO

`format` TODO

Value

TODO

`ehr_map_codes` *Map EHR codes to another terminology*

Description

Map EHR codes from one terminology to another using UK Biobank mapping dictionaries. Currently supports mapping Read v2 to CTV 3 only.

Usage

```
ehr_map_codes(codes, from = "read2", to = "ctv3", overwrite = FALSE)
```

Arguments

`codes` Vector of codes to map.

`from` Terminology to map from. Currently supports Read v2 codes only (first 5 digits excluding the final 2 digit term). Full 5 digit codes must be passed i.e. no partial matching.

`to` Terminology to map to. Currently supports CTV3 only.

`overwrite` Overwrite existing mapping dictionary (default FALSE).

Value

Data table with mapped codes and description.

get_coding *Get UK Biobank coding*

Description

Returns the UK Biobank coding file for the supplied id. The file is downloaded if it is unavailable locally.

Usage

```
get_coding(id, overwrite = FALSE)
```

Arguments

id ID for coding file. Obtain this from the data dictionary for your application, or directly from the Data Showcase (https://biobank.ndph.ox.ac.uk/showcase/), by looking up the relevant data field.

overwrite Overwrite existing file (default FALSE).

Value

Data table with contents of coding file.

get_hdr_concept *Get concept from HDR UK phenotype library*

Description

Queries HDR UK phenotype library for a given concept. Read v2 codes are mapped to CTV3 if the read argument is TRUE.

Usage

```
get_hdr_concept(id, version_id = NULL, api = NULL, read = FALSE)
```

Arguments

id Concept ID.

version_id Optional concept version ID. Otherwise latest version of concept is returned.

api Optional HttpClient to connect to phenotype library (default uses https://phenotypes.healthdatagateway.or

read Set to TRUE if concept contains Read v2 codes. Read v2 codes of length less than 5 are padded with a "." e.g. "C10" becomes "C10..". Codes are mapped to CTV3 using UK Biobank mapping dictionaries.

Value

Data table with phenotype codes. `read_2` (Read v2) and `read_3` (CTV3) columns are added if `read` is TRUE.

`get_hdr_phenotype` *Get phenotype from HDR UK phenotype library*

Description

Queries HDR UK phenotype library for a given phenotype.

Usage

```
get_hdr_phenotype(id, version_id = NULL, api = NULL)
```

Arguments

`id`	Phenotype ID.
`version_id`	Optional phenotype version ID. Otherwise latest version of phenotype is returned.
`api`	Optional HttpClient to connect to phenotype library (default uses https://phenotypes.healthdatagateway.org

Value

Data table with phenotype codes.

`visit` *Example UK Biobank visit data set.*

Description

Synthetic UK Biobank visit data for function testing and demonstration.

Usage

```
visit
```

Format

Data table with columns as set out below.

Details

Data is provided for 100 synthetic participants with two instances (visits) for 40% of participants. Fields provided are `50` (height), `53` (visit date) and `21002` (weight).

`visit_cancer` *Extract self-reported cancer history*

Description

Extracts self-reported cancer history in a "long" format that is easier to work with than "wide" as provided by UK Biobank (NOTE: watch for type coercion of different data types). Wrapper function for `visit_mult_array()`.

Usage

```
visit_cancer(visit_data)
```

Arguments

`visit_data`	Data frame/table with UK Biobank data. Must include fields `20001` and `20006`.

Value

Data table with the following columns:

eid UK Biobank identifier.

date Reported date of diagnosis.

condition Cancer type coded as in https://biobank.ctsu.ox.ac.uk/crystal/coding.cgi?id=3.

desc Description of cancer type (added if coding data can be downloaded).

reported Visit date at which the cancer was self-reported.

`visit_conditions` *Extract self-reported non-cancer medical history*

Description

Extracts self-reported non-cancer medical history in a "long" format that is easier to work with than "wide" as provided by UK Biobank (NOTE: watch for type coercion of different data types). Wrapper function for `visit_mult_array()`.

Usage

```
visit_conditions(visit_data)
```

Arguments

`visit_data`	Data frame/table with UK Biobank data. Must include fields `20002` and `20008`.

Value

Data table with the following columns:

eid UK Biobank identifier.

date Reported date of diagnosis.

condition Health condition coded as in https://biobank.ctsu.ox.ac.uk/crystal/coding.cgi?id=6.

desc Description of health condition (added if coding data can be downloaded).

reported Visit date at which the condition was self-reported.

`visit_extract` *Extract field data from UK Biobank visit data*

Description

Extracts all instances/arrays of data for a UK Biobank field(s) in clean "long" format (NOTE: watch for type coercion of different data types). See https://biobank.ndph.ox.ac.uk/showcase/ to identify field codes. Wrapper for `visit_fields()` which extracts raw field data.

Usage

```
visit_extract(visit_data, fields, format = NULL)
```

Arguments

`visit_data`	Data frame/table with UK Biobank data.
`fields`	Vector of fields to extract e.g. `50` or `c(50,21002)`. Field name will be identified from UK Biobank schema. Alternatively, field names can be set using a named vector e.g. `c("height" = 50,"weight" = 21002)`.
`format`	Format of output table (`raw` or `source`). Default is currently raw but will change to source in a future release.

Value

Data table with values of all instances/arrays for each field in "long" format. The following columns are provided:

eid UK Biobank identifier.

date Visit date.

field/variable Field name (see below).

array Provided if any fields have multiple arrays (more than one value recorded on the same date e.g repeated blood pressure).

value Value recorded.

If `format` = "source", an additional column `source` = "ukbb" is added to indicate data was recorded by UK Biobank and the field column is renamed `variable`. This will be the default in a future release. Use `format` = "raw" to keep current format.

Examples

```
## Not run:
# Load data
data_path <- ""  # add path to your data
visit_data <- fread(data_path)

# Extract a field
visit_extract(visit_data, 50)

# Extract multiple fields
visit_extract(visit_data, c(50, 21002))

# Manually specify a field name
visit_extract(visit_data, c("height" = 50, 21002))

## End(Not run)
```

visit_family_history *Determine family history of a specified condition*

Description

Determines presence of a specified condition in the self-reported family history data. If multiple history fields are provided (e.g. history of mother and father), presence of the condition in either field determines a positive family history.

Usage

```
visit_family_history(
  visit_data,
  fields,
  condition,
  collapse = TRUE,
  name = NULL
)
```

Arguments

`visit_data`	Data frame/table with UK Biobank data.
`fields`	Vector of family history fields to extract e.g. one or more from `20107` (father), `20110` (mother) or `20111` (sibling).
`condition`	Code for condition. Only a single condition at a time is currently supported. See https://biobank.ndph.ox.ac.uk/showcase/coding.cgi?id=1010 to identify condition codes.
`collapse`	Summarise results across all visit dates (default `TRUE`). If `FALSE`, the presence of the condition is provided at each visit date in the output table.
`name`	Optional column name for condition (default `history`).

Value

Data table with TRUE (condition was reported), FALSE (condition was not reported) or NA (unknown/no response).

Examples

```
## Not run:
# Load data
data_path <- ""  # add path to your data
visit <- fread(data_path)

# Extract history for father
visit_family_history(visit, 20107, 9)

# Extract history for father, mother and siblings
visit_family_history(visit, c(20107, 20110, 20111), 9)

# Name column in output
visit_family_history(visit, c(20107, 20110, 20111), 9, name = "diabetes")

# Get history reported at each visit date
visit_family_history(visit, c(20107, 20110, 20111), 9, collapse = FALSE)

## End(Not run)
```

visit_fields *Extract raw field data from UK Biobank visit data*

Description

Extracts all instances/arrays of data for a UK Biobank field(s). See https://biobank.ndph.ox.ac.uk/showcase/ to identify field codes.

Usage

```
visit_fields(visit_data, fields, format = "wide")
```

Arguments

visit_data	Data frame/table with UK Biobank data.
fields	Vector of fields to extract e.g. 50 or c(50,21002).
format	Format of output table i.e. "wide" or "long" (default "wide"). NOTE: watch for type coercion if use long format.

Value

Data table with all instances/arrays for each field.

Examples

```
## Not run:
# Load data
data_path <- ""  # add path to your data
visit_data <- fread(data_path)

# Extract a field
visit_fields(visit_data, 50)

# Extract multiple fields
visit_fields(visit_data, c(50, 21002))

# Extract multiple fields in long format
visit_fields(visit_data, c(50, 21002), format = "long")

## End(Not run)
```

`visit_mult_array` *Extract data from two UK Biobank fields jointly*

Description

Some fields relate to each other e.g. self-reported medical history where field 20002 contains the disclosed conditions and 20008 the date of diagnosis. The date in array `i` of 20008 corresponds to the condition in array `i` of 20002. `visit_mult_array()` jointly extracts such fields in a "long" format that is easier to work with than "wide" as provided by UK Biobank (NOTE: watch for type coercion of different data types).

Usage

```
visit_mult_array(visit_data, fields)
```

Arguments

`visit_data`	Data frame/table with UK Biobank data.
`fields`	Vector of fields to extract e.g. `c(50,21002)`. Must be length two. Field name will be identified from UK Biobank schema. Alternatively, field names can be set using a named vector e.g. `c("height" = 50,"weight" = 21002)`.

Value

Data table with `eid`, `reported`, and columns corresponding to the `fields` argument. `reported` is the date corresponding to the field instance e.g. the UK Biobank visit at which the data was collected. Each row shows the data for an array.

`visit_subset` *Subset fields from UK Biobank visit data*

Description

Loads UK Biobank data and subsets required fields. Use to avoid loading full data each time. See https://biobank.ndph.ox.ac.uk/showcase/ to identify field codes.

Usage

```
visit_subset(data_path, fields, ..., save = NULL)
```

Arguments

`data_path`	Path to raw UK Biobank data unpacked using ukbunpack utility.
`fields`	Vector of fields to extract e.g. `50` or `c(50,21002)`.
`...`	Passed to `fread` e.g. to set file separator.
`save`	Optional path to save output.

Value

Data table with all instances/arrays for each field. Note the date field (53) is always returned.

Index