

# Package ‘ukbbhelpr’

November 29, 2021

**Type** Package

**Title** Helper functions for UK Biobank data

**Version** 0.1.0

**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** data.table,  
lubridate,  
stringr

**RoxygenNote** 7.1.2

**Suggests** testthat

## R topics documented:

ehr_extract . . . . .	2
get_coding . . . . .	2
visit . . . . .	3
visit_cancer . . . . .	3
visit_conditions . . . . .	4
visit_extract . . . . .	4
visit_family_history . . . . .	5
visit_fields . . . . .	6
visit_mult_array . . . . .	7
visit_subset . . . . .	8
<b>Index</b>	<b>9</b>

---

ehr_extract	<i>Extract observations/test results from raw EHR data</i>
-------------	--

---

### Description

Extracts values from linked EHR data. Data is extracted from 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)
```

### Arguments

ehr_data	Data table/frame with UK Biobank clinical event data i.e. gp_clinical.txt.
read_codes	Values are extracted from these codes. Data table/frame with Read v2 (read_2) and CTV3 (read_3) columns.

### Value

Data table with value and unit columns.

---

get_coding	<i>Get UK Biobank coding</i>
------------	------------------------------

---

### 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 ( <a href="https://biobank.ndph.ox.ac.uk/showcase/">https://biobank.ndph.ox.ac.uk/showcase/</a> ), by looking up the relevant data field.
overwrite	Overwrite existing file (default FALSE).

### Value

Data table with contents of coding file.

---

visit	<i>Example UK Biobank visit data set.</i>
-------	---

---

### 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	<i>Extract self-reported cancer history</i>
--------------	---

---

### 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	<i>Extract self-reported non-cancer medical history</i>
------------------	---

---

### 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	<i>Extract field data from UK Biobank visit data</i>
---------------	--

---

### 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)
```

### Arguments

visit_data	Data frame/table with UK Biobank data.
------------	--

fields	Vector of fields to extract e.g. 50 or <code>c(50, 21002)</code> . Field name will be identified from UK Biobank schema. Alternatively, field names can be set using a named vector e.g. <code>c("height" = 50, "weight" = 21002)</code> .
--------	--

**Value**

Data table with values of all instances/arrays for each field in "long" format i.e. eid, date of visit, field name and value recorded by UK Biobank. array is provided if any fields have multiple arrays (more than one value recorded on the same date e.g repeated blood pressure).

**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	<i>Determine family history of a specified condition</i>
----------------------	--

---

**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 <a href="https://biobank.ndph.ox.ac.uk/showcase/coding.cgi?id=1010">https://biobank.ndph.ox.ac.uk/showcase/coding.cgi?id=1010</a> 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	<i>Extract raw field data from UK Biobank visit data</i>
--------------	--

---

### 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	<i>Extract data from two UK Biobank fields jointly</i>
------------------	--

---

**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. <code>c(50,21002)</code> . 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. <code>c("height" = 50, "weight" = 21002)</code> .

**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	<i>Subset fields from UK Biobank visit data</i>
--------------	---

---

**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 <code>ukbunpack</code> utility.
fields	Vector of fields to extract e.g. <code>50</code> or <code>c(50,21002)</code> .
...	Passed to <code>fread</code> 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

## \*Topic **datasets**

visit, [3](#)

ehr\_extract, [2](#)

get\_coding, [2](#)

visit, [3](#)

visit\_cancer, [3](#)

visit\_conditions, [4](#)

visit\_extract, [4](#)

visit\_family\_history, [5](#)

visit\_fields, [6](#)

visit\_mult\_array, [7](#)

visit\_subset, [8](#)