Package 'ukbtools'

May 15, 2019

Version 0.11.3

Title Manipulate and Explore UK Biobank Data

Maintainer Ken Hanscombe <ken.hanscombe@gmail.com>

Description A set of tools to create a UK Biobank <http://www.ukbiobank.ac.uk/> dataset from a UKB fileset (.tab, .r, .html), visualize primary demographic data for a sample subset, query ICD diagnoses, retrieve genetic metadata, read and write standard file formats for genetic analyses.

License GPL-2

URL https://kenhanscombe.github.io/ukbtools/

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports data.table (>= 1.12), dplyr, purrr, readr, ggplot2, XML, magrittr, grid, tibble, tidyr, scales, stringr, foreach, parallel, doParallel

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Ken Hanscombe [aut, cre]

Repository CRAN

Date/Publication 2019-05-15 11:40:03 UTC

R topics documented:

`icd10chapters` *International Classification of Diseases Revision 10 (ICD-10) chapters*

Description

A dataset containing the ICD-10 chapter titles - a top level description of diagnoses classes (or blocks)

Usage

```
icd10chapters
```

Format

An object of class `data.frame` with 21 rows and 3 columns.

icd10codes — *International Classification of Diseases Revision 10 (ICD-10) codes*

Description

A dataset containing the full set ICD-10 diagnoses

Usage

```
icd10codes
```

Format

An object of class `data.frame` with 18761 rows and 2 columns.

icd9chapters — *International Classification of Diseases Revision 9 (ICD-9) chapters*

Description

A dataset containing the ICD-9 chapter titles - a top level description of diagnoses classes (or blocks)

Usage

```
icd9chapters
```

Format

An object of class `data.frame` with 19 rows and 3 columns.

icd9codes — *International Classification of Diseases Revision 9 (ICD-9) codes*

Description

A dataset containing the full set ICD-9 diagnoses

Usage

```
icd9codes
```

Format

An object of class `data.frame` with 13679 rows and 2 columns.

ukbcentre *UKB assessment centre*

Description

A dataset containing the 22 assessment centres (as well as pilot test centre and a revisit centre)

Usage

```
ukbcentre
```

Format

An object of class `data.frame` with 24 rows and 2 columns.

ukbtools *ukbtools: Manipulate and Explore UK Biobank Data*

Description

A set of tools to create a UK Biobank dataset from a UKB fileset (.tab, .r, .html), visualize primary demographic data for a sample subset, query ICD diagnoses, retrieve genetic metadata, read and write standard file formats for genetic analyses.

UKB Dataframe

Functions to wrangle the UKB data into a dataframe with meaningful column names.

- ukb_df
- ukb_df_field
- ukb_df_full_join
- ukb_df_duplicated_name
- ukb_centre
- ukb_context

Genetic Metadata

Functions to query the associated genetic sample QC information.

- ukb_gen_read_fam
- ukb_gen_read_sample
- ukb_gen_rel_count
- ukb_gen_related_with_data
- ukb_gen_samples_to_remove
- ukb_gen_sqc_names
- ukb_gen_write_bgenie
- ukb_gen_write_plink

Disease Diagnoses

Functions to query the UKB hospital episodes statistics.

- ukb_icd_code_meaning
- ukb_icd_diagnosis
- ukb_icd_freq_by
- ukb_icd_keyword
- ukb_icd_prevalence

Datasets

- ukbcentre
- icd10chapters
- icd10codes
- icd9chapters
- icd9codes

---

ukb_centre *Inserts UKB centre names into data*

Description

Inserts a column with centre name, ukb_centre, into the supplied data.frame. Useful if your UKB centre variable uk_biobank_assessment_centre_0_0 has not been populated with named levels.

Usage

```
ukb_centre(data, centre.var = "^uk_biobank_assessment_centre.*0_0")
```

Arguments

data	A UKB dataset created with ukb_df.
centre.var	The UKB column containing numerically coded assessment centre. The default is a regular expression "^uk_biobank_assessment_centre.*0_0".

Value

A dataframe with an additional column ukb_centre - UKB assessment centre names

ukb_context *Demographics of a UKB sample subset*

Description

Describes a subset of the UKB sample, relative to a reference subsample, on the UKB primary demographics (sex, age, ethnicity, Townsend deprivation) and assessment centre and current employment status. The "subset" and "reference" samples are defined either by a variable of interest (`nonmiss.var` - those with data form the "subset" of interest and samples with missing data are the "reference" sample), or a logical vector (`subset.var` - where `TRUE` values define the "subset" and `FALSE` the "reference" samples) . This function is intended as an exploratory data analysis and quality control tool.

Usage

```
ukb_context(data, nonmiss.var = NULL, subset.var = NULL,
  bar.position = "fill", sex.var = "^sex.*0_0",
  age.var = "^age_when_attended_assessment_centre.*0_0",
  socioeconomic.var = "^townsend_deprivation_index_at_recruitment.*0_0",
  ethnicity.var = "^ethnic_background.*0_0",
  employment.var = "^current_employment_status.*0_0",
  centre.var = "^uk_biobank_assessment_centre.*0_0")
```

Arguments

`data`	A UKB dataset constructed with `ukb_df`.
`nonmiss.var`	The variable of interest which defines the "subset" (samples with data) and "reference" (samples without data, i.e., NA) samples.
`subset.var`	A logical vector defining a "subset" (`TRUE`) and "reference" subset (`FALSE`). Length must equal the number of rows in your `data`.
`bar.position`	This argument is passed to the `position` in `geom_bar`. The default value is `"fill"` which shows reference and subset of interest as proportions of the full dataset. Useful alternatives are `"stack"` for counts and `"dodge"` for side-by-side bars.
`sex.var`	The variable to be used for sex. Default value is the regular expression "^sex.*0_0".
`age.var`	The variable to be use for age. Default value is the regular expression "^age_when_attended_assessment_c
`socioeconomic.var`	The variable to be used for socioeconomic status. Default value is deprivation at baseline, the regular expression "^townsend_deprivation_index_at_recruitment.*0_0".
`ethnicity.var`	The variable to be used for ethnicity. Default value is the regular expression "^ethnic_background.*0_0".
`employment.var`	The variable to be used for employment status. Default value is employment status at baseline "^current_employment_status.*0_0".
`centre.var`	The variable to be used for assessment centre. Default value is the regular expression "^uk_biobank_assessment_centre.*0_0".

See Also

ukb_df

Examples

```
## Not run:
# Compare those with data to those without
ukb_context(my_ukb_data, nonmiss.var = "my_variable_of_interest")

# Define a subset of interest as a logical vector
subgroup_of_interest <- (my_ukb_data$bmi > 40 & my_ukb_data$age < 50)
ukb_context(my_ukb_data, subset.var = subgroup_of_interest)

## End(Not run)
```

---

ukb_defunct *Defunct genetic metadata functions*

Description

The genetic metadata functions were written to retrieve genetic metadata from the phenotype file for the interim genotype release. **The fields retrieved became obselete when the full genotyping results (500K individuals) were released at the end of 2017.** With the release of the full genotyping results, sample QC (ukb_sqc_v2.txt) and marker QC (ukb_snp_qc.txt) data are now supplied as separate files. The contents of these files, along with all other genetic files are described fully in UKB Resource 531.

- ukb_gen_meta(data)
- ukb_gen_pcs(data)
- ukb_gen_excl(data)
- ukb_gen_rel(data)
- ukb_gen_het(data, all.het = FALSE)
- ukb_gen_excl_to_na(data, x, ukb.id = "eid", data.frame = FALSE)
- ukb_gen_write_plink_excl(path)

Usage

```
ukb_defunct()
```

Details

See also lists new functionality that works with the files described in UKB Resource 531.

See Also

ukb_gen_sqc_names, ukb_gen_rel_count, ukb_gen_related_with_data, ukb_gen_samples_to_remove

ukb_df *Reads a UK Biobank phenotype fileset and returns a single dataset.*

Description

A UK Biobank *fileset* includes a *.tab* file containing the raw data with field codes instead of variable names, an *.r* (*sic*) file containing code to read raw data (inserts categorical variable levels and labels), and an *.html* file containing tables mapping field code to variable name, and labels and levels for categorical variables.

Usage

```
ukb_df(fileset, path = ".", n_threads = "dt", data.pos = 2)
```

Arguments

`fileset`	The prefix for a UKB fileset, e.g., ukbxxxx (for ukbxxxx.tab, ukbxxxx.r, ukbxxxx.html)
`path`	The path to the directory containing your UKB fileset. The default value is the current directory.
`n_threads`	Either "max" (uses the number of cores, ‘parallel::detectCores()‘), "dt" (default - uses the data.table default, ‘data.table::getDTthreads()‘), or a numerical value (in which case n_threads is set to the supplied value, or ‘parallel::detectCores()‘ if it is smaller).
`data.pos`	Locates the data in your .html file. The .html file is read into a list; the default value data.pos = 2 indicates the second item in the list. (The first item in the list is the title of the table). You will probably not need to change this value, but if the need arises you can open the .html file in a browser and identify where in the file the data is.

Details

The **index** and **array** from the UKB field code are preserved in the variable name, as two numbers separated by underscores at the end of the name e.g. *variable_index_array*. **index** refers the assessment instance (or visit). **array** captures multiple answers to the same "question". See UKB documentation for detailed descriptions of index and array.

Value

A dataframe with variable names in snake_case (lowercase and separated by an underscore).

See Also

`ukb_df_field` `ukb_df_full_join`

Examples

```
## Not run:
# Simply provide the stem of the UKB fileset.
# To read ukb1234.tab, ukb1234.r, ukb1234.html

my_ukb_data <- ukb_df("ukb1234")


If you have multiple UKB filesets, read each then join with your preferred
method (ukb_df_full_join is
a thin wrapper around dplyr::full_join applied recursively with
purrr::reduce).

ukb1234_data <- ukb_df("ukb1234")
ukb2345_data <- ukb_df("ukb2345")
ukb3456_data <- ukb_df("ukb3456")

ukb_df_full_join(ukb1234_data, ukb2345_data, ukb3456_data)

## End(Not run)
```

---

ukb_df_duplicated_name

Checks for duplicated names within a UKB dataset

Description

Checks for duplicated names within a UKB dataset

Usage

```
ukb_df_duplicated_name(data)
```

Arguments

data	A UKB dataset created with ukb_df.

Details

Duplicates *within* a UKB dataset are unlikely to occur, however, ukb_df creates variable names by combining a snake_case descriptor with the variable's **index** and **array**. If an index_array combination is incorrectly repeated in the original UKB data, this will result in a duplicated variable name. . See `vignette(topic = "explore-ukb-data", package = "ukbtools")` for further details.

Value

Returns a named list of numeric vectors, one for each duplicated variable name. The numeric vectors contain the column indices of duplicates.

`ukb_df_field` *Makes a UKB data-field to variable name table for reference or lookup.*

Description

Makes either a table of Data-Field and description, or a named vector handy for looking up descriptive name by column names in the UKB fileset tab file.

Usage

```
ukb_df_field(fileset, path = ".", data.pos = 2, as.lookup = FALSE)
```

Arguments

`fileset`	The prefix for a UKB fileset, e.g., ukbxxxx (for ukbxxxx.tab, ukbxxxx.r, ukbxxxx.html)
`path`	The path to the directory containing your UKB fileset. The default value is the current directory.
`data.pos`	Locates the data in your .html file. The .html file is read into a list; the default value data.pos = 2 indicates the second item in the list. (The first item in the list is the title of the table). You will probably not need to change this value, but if the need arises you can open the .html file in a browser and identify where in the file the data is.
`as.lookup`	If set to TRUE, returns a named `vector`. The default `as.look = FALSE` returns a dataframe with columns: field.showcase (as used in the UKB online showcase), field.data (as used in the tab file), name (descriptive name created by `ukb_df`)

Value

Returns a data.frame with columns `field.showcase`, `field.html`, `field.tab`, `names`. `field.showcase` is how the field appears in the online UKB showcase; `field.html` is how the field appears in the html file in your UKB fileset; `field.tab` is how the field appears in the tab file in your fileset; and `names` is the descriptive name that `ukb_df` assigns to the variable. If `as.lookup = TRUE`, the function returns a named character vector of the descriptive names.

See Also

`ukb_df`

Examples

```
## Not run:
# UKB field-to-description for ukb1234.tab, ukb1234.r, ukb1234.html

ukb_df_field("ukb1234")

## End(Not run)
```

`ukb_df_full_join` *Recursively join a list of UKB datasets*

Description

A thin wrapper around `purrr::reduce` and `dplyr::full_join` to merge multiple UKB datasets.

Usage

```
ukb_df_full_join(..., by = "eid")
```

Arguments

`...`	Supply comma separated unquoted names of to-be-merged UKB datasets (created with `ukb_df`). Arguments are passed to `list`.
`by`	Variable used to merge multiple dataframes (default = "eid").

Details

The function takes a comma separated list of unquoted datasets. By explicitly setting the join key to "eid" only (Default value of the by parameter), any additional variables common to any two tables will have ".x" and ".y" appended to their names. If you are satisfied the additional variables are identical to the original, the copies can be safely deleted. For example, if `setequal(my_ukb_data$var, my_ukb_data$var.x)` is `TRUE`, then my_ukb_data$var.x can be dropped. A `dlyr::full_join` is like the set operation union in that all observations from all tables are included, i.e., all samples are included even if they are not included in all datasets.

NB. `ukb_df_full_join` will fail if any variable names are repeated **within** a single UKB dataset. This is unlikely to occur, however, `ukb_df` creates variable names by combining a snake_case descriptor with the variable's **index** and **array**. If an index_array combination is incorrectly repeated, this will result in a duplicated variable. If the join fails, you can use `ukb_df_duplicated_name` to find duplicated names. See `vignette(topic = "explore-ukb-data", package = "ukbtools")` for further details.

See Also

`ukb_df_duplicated_name`

Examples

```
## Not run:
# If you have multiple UKB filesets, tidy then merge them.

ukb1234_data <- ukb_df("ukb1234")
ukb2345_data <- ukb_df("ukb2345")
ukb3456_data <- ukb_df("ukb3456")

my_ukb_data <- ukb_df_full_join(ukb1234_data, ukb2345_data, ukb3456_data)

## End(Not run)
```

---

ukb_gen_excl *Sample exclusions*

Description

Defunct. See help("ukb_defunct").

This list of sample exclusions includes UKB's "recommended", "affymetrix quality control", and "genotype quality control" exclusions. UKB have published full details of genotyping and quality control for the interim genotype data.

Usage

```
ukb_gen_excl(data)
```

Arguments

data	A UKB dataset created with ukb_df.

Examples

```
## Not run:
# For a vector of IDs
recommended_excl_ids <- ukb_gen_excl(my_ukb_df)

## End(Not run)
```

ukb_gen_excl_to_na *Inserts NA into phenotype for genetic metadata exclusions*

Description

Defunct. See help("ukb_defunct").

Replaces data values in a vector (a UKB phenotype) with NA where the sample is to-be-excluded, i.e., is either a UKB recommended exclusion, a heterozygosity outlier, a genetic ethnicity outlier, or a randomly-selected member of a related pair.

Usage

```
ukb_gen_excl_to_na(data, x, ukb.id = "eid", data.frame = FALSE)
```

Arguments

data	A UKB dataset created with ukb_df.
x	The phenotype to be updated (as it is named in data) e.g. "height"
ukb.id	The name of the ID variable in data. Default is "eid"
data.frame	A logical vector indicating whether to return a vector or a data.frame (header: id, meta_excl, pheno, pheno_meta_na) containing the original and updated variable. Default = FALSE returns a vector.

See Also

ukb_gen_write_plink_excl

Examples

```
## Not run:
my_ukb_data$height_excl_na <- ukb_gen_excl_to_na(my_ukb_data, x = "height")

## End(Not run)
```

ukb_gen_het *Heterozygosity outliers*

Description

Defunct. See help("ukb_defunct").

Heterozygosity outliers are typically removed from genetic association analyses. This function returns either a vector of heterozygosity outliers to remove (+/- 3sd from mean heterozygosity), or a data frame with heterozygosity scores for all samples.

Usage

```
ukb_gen_het(data, all.het = FALSE)
```

Arguments

data	A UKB dataset created with ukb_df.
all.het	Set all.het = TRUE for heterozygosity scores for all samples. By default all.het = FALSE returns a vector of sample IDs for individuals +/-3SD from the mean heterozygosity.

Details

UKB have published full details of genotyping and quality control for the interim genotype data.

Value

A vector of IDs if all.het = FALSE (default), or a dataframe with ID, heterozygosity and PCA-corrected heterozygosity if all.het = TRUE.

Examples

```
## Not run:
#' # Heterozygosity outliers (+/-3SD)
outlier_het_ids <- ukb_gen_het(my_ukb_data)

# Retrieve all raw and pca-corrected heterozygosity scores
ukb_het <- ukb_gen_het(my_ukb_data, all.het = TRUE)

## End(Not run)
```

---

ukb_gen_meta *Genetic metadata*

Description

Defunct. See help("ukb_defunct").

UKB have published full details of genotyping and quality control for the interim genotype data. This function retrieves UKB assessment centre codes and assessment centre names, genetic ethnic grouping, genetically-determined sex, missingness, UKB recommended genomic analysis exclusions, BiLeve unrelatedness indicator, and BiLeve Affymetrix and genotype quality control.

Usage

```
ukb_gen_meta(data)
```

Arguments

data	A UKB dataset created with ukb_df.

ukb_gen_pcs *Genetic principal components*

Description

Defunct. See help("ukb_defunct").

These are the principal components derived on the UK Biobank subsample with interim genotype data. UKB have published full details of genotyping and quality control for the interim genotype data.

Usage

```
ukb_gen_pcs(data)
```

Arguments

data	A UKB dataset created with ukb_df.

ukb_gen_read_fam *Reads a PLINK format fam file*

Description

This is wrapper for read_table that reads a basic PLINK fam file. For plink hard-called data, it may be useful to use the fam file ids as a filter for your phenotype and covariate data.

Usage

```
ukb_gen_read_fam(file, col.names = c("FID", "IID", "paternalID",
  "maternalID", "sex", "phenotype"), na.strings = "-9")
```

Arguments

file	A path to a fam file.
col.names	A character vector of column names. Default: c("FID", "IID", "paternalID", "maternalID", "sex", "phenotype")
na.strings	Character vector of strings to use for missing values. Default "-9". Set this option to character() to indicate no missing values.

See Also

ukb_gen_read_sample to read a sample file

ukb_gen_read_sample *Reads an Oxford format sample file*

Description

This is a wrapper for `read_table` that reads an Oxford format .sample file. If you use the unedited sample file as supplied with your genetic data, you should only need to specify the first argument, file.

Usage

```
ukb_gen_read_sample(file, col.names = c("id_1", "id_2", "missing"),
  row.skip = 2)
```

Arguments

`file`	A path to a sample file.
`col.names`	A character vector of column names. Default: c("id_1", "id_2", "missing")
`row.skip`	Number of lines to skip before reading data.

See Also

`ukb_gen_read_fam` to read a fam file

ukb_gen_rel *Creates a table of related individuals*

Description

Defunct. See help("ukb_defunct").

Makes a data.frame containing all related individuals with columns UKB ID, pair ID, KING kinship coefficient, and proportion of alleles IBS = 0. UKB have published full details of genotyping and quality control including details on relatedness calculations for the interim genotype data.

Usage

```
ukb_gen_rel(data)
```

Arguments

`data`	A UKB dataset created with `ukb_df`.

See Also

`ukb_gen_rel_count`

ukb_gen_related_with_data *Subset of the UKB relatedness dataframe with data*

Description

Subset of the UKB relatedness dataframe with data

Usage

```
ukb_gen_related_with_data(data, ukb_with_data, cutoff = 0.0884)
```

Arguments

data	The UKB relatedness data as a dataframe (header: ID1, ID2, HetHet, IBS0, Kinship)
ukb_with_data	A character vector of ukb eids with data on the phenotype of interest
cutoff	KING kingship coefficient cutoff (default 0.0884 includes pairs with greater than 3rd-degree relatedness)

Value

A dataframe (header: ID1, ID2, HetHet, IBS0, Kinship) for the subset of individuals with data.

See Also

ukb_gen_rel_count, ukb_gen_samples_to_remove

ukb_gen_rel_count *Relatedness count*

Description

Creates a summary count table of the number of individuals and pairs at each degree of relatedness that occurs in the UKB sample, and an optional plot.

Usage

```
ukb_gen_rel_count(data, plot = FALSE)
```

Arguments

data	A dataframe of the genetic relatedness data including KING kinship coefficient, and proportion of alleles IBS = 0. See Details.
plot	Logical indicating whether to plot relatedness figure. Default = FALSE.

Details

Use UKB supplied program 'ukbgene' to retrieve genetic relatedness data file ukbA_rel_sP.txt. See UKB Resource 664. The count and plot include individuals with IBS0 >= 0.

Value

If `plot = FALSE` (default), a count of individuals and pairs at each level of relatedness. If `plot = TRUE`, reproduces the scatterplot of genetic relatedness against proportion of SNPs shared IBS=0 (each point representing a pair of related UKB individuals) from the genotyping and quality control documentation.

See Also

`ukb_gen_related_with_data`, `ukb_gen_samples_to_remove`

Examples

```
## Not run:
# Use UKB supplied program `ukbgene` to retrieve genetic relatedness file ukbA_rel_sP.txt.
See \href{http://biobank.ctsu.ox.ac.uk/crystal/refer.cgi?id=664}{UKB Resource 664}.
With the whitespace delimited file read into R as e.g. ukb_relatedness,
generate a dataframe of counts or a plot as follows:

ukb_gen_rel_count(ukb_relatedness)
ukb_gen_rel_count(ukb_relatedness, plot = TRUE)

## End(Not run)
```

---

ukb_gen_samples_to_remove

Related samples (with data on the variable of interest) to remove

---

Description

There are many ways to remove related individuals from phenotypic data for genetic analyses. You could simply exclude all individuals indicated as having "excess relatedness" and include those "used in pca calculation" (these variables are included in the sample QC data, ukb_sqc_v2.txt) - see details. This list is based on the complete dataset, and possibly removes more samples than you need to for your phenotype of interest. Ideally, you want a maximum independent set, i.e., to remove the minimum number of individuals with data on the phenotype of interest, so that no pair exceeds some cutoff for relatedness. `ukb_gen_samples_to_remove` returns a list of samples to remove in to achieve a maximal set of unrelateds for a given phenotype.

Usage

```
ukb_gen_samples_to_remove(data, ukb_with_data, cutoff = 0.0884)
```

Arguments

`data`	The UKB relatedness data as a dataframe (header: ID1, ID2, HetHet, IBS0, Kinship)
`ukb_with_data`	A character vector of ukb eids with data on the phenotype of interest
`cutoff`	KING kingship coefficient cutoff (default 0.0884 includes pairs with greater than 3rd-degree relatedness)

Details

Trims down the UKB relatedness data before selecting individuals to exclude, using the algorithm: step 1. remove pairs below KING kinship coefficient 0.0884 (3rd-degree or less related, by default. Can be set with `cutoff` argument), and any pairs if either member does not have data on the phenotype of interest. The user supplies a vector of samples with data. step 2. count the number of "connections" (or relatives) each participant has and add to "samples to exclude" the individual with the most connections. This is the greedy part of the algorithm. step 3. repeat step 2 till all remaining participants only have 1 connection, then add one random member of each remaining pair to "samples to exclude" (adds all those listed under ID2)

Another approach from the UKB email distribution list:

To: UKB-GENETICS@JISCMAIL.AC.UK Date: Wed, 26 Jul 2017 17:06:01 +0100 **Subject: A list of unrelated samples**

(...) you could use the list of samples which we used to calculate the PCs, which is a (maximal) subset of unrelated participants after applying some QC filtering. Please read supplementary Section S3.3.2 for details. You can find the list of samples using the "used.in.pca.calculation" column in the sample-QC file (ukb_sqc_v2.txt) (...). Note that this set contains diverse ancestries. If you take the intersection with the white British ancestry subset you get ~337,500 unrelated samples.

Value

An integer vector of UKB IDs to remove.

See Also

`ukb_gen_rel_count`, `ukb_gen_related_with_data`

`ukb_gen_sqc_names` *Sample QC column names*

Description

The UKB sample QC file has no header on it.

Usage

```
ukb_gen_sqc_names(data, col_names_only = FALSE)
```

Arguments

data	The UKB ukb_sqc_v2.txt data as dataframe. (Not necessary if column names only are required)
col_names_only	If TRUE returns a character vector of column names (data argument not required). Useful if you would like to supply as header when reading in your sample QC data. If FALSE (Default), returns the supplied dataframe with column names (Checks number of columns in supplied data. See Details.).

Details

From UKB Resource 531: There are currently 2 versions of this file (UKB ukb_sqc_v2.txt) in circulation. The newer version is described below and contains column headers on the first row. The older (deprecated) version lacks the column headers and has two additional Affymetrix internal values prefixing the columns listed below.

Value

A sample QC dataframe with column names, or a character vector of column names if `col_names_only = TRUE`.

---

ukb_gen_write_bgenie *Writes a BGENIE format phenotype or covariate file.*

Description

Writes a space-delimited file with a header, missing character set to "-999", and observations (i.e. UKB subject ids) in sample file order. Use this function to write phenotype and covariate files for downstream genetic analysis in BGENIE - the format is the same.

Usage

```
ukb_gen_write_bgenie(x, ukb.sample, ukb.variables, path, ukb.id = "eid",
  na.strings = "-999")
```

Arguments

x	A UKB dataset.
ukb.sample	A UKB sample file.
ukb.variables	A character vector of either the phenotypes for a BGENIE phenotype file, or covariates for a BGENIE covariate file.
path	A path to a file.
ukb.id	The eid variable name (default = "eid").
na.strings	Character string to be used for missing value in output file. Default = "-999"

Details

Uses a `dplyr::left_join` to the sample file to match sample file order. Any IDs in the sample file not included in the phenotype or covariate data will be missing for all variables selected. See BGENIE usage for descriptions of the `--pheno` and `--covar` flags to read phenotype and covariate data into BGENIE.

See Also

`ukb_gen_read_sample` to read a sample file, `ukb_gen_excl_to_na` to update a phenotype with NAs for samples to-be-excluded based on genetic metadata, and `ukb_gen_write_plink` to write phenotype and covariate files to PLINK format.

Examples

```
## Not run:

# Automatically sorts observations to match UKB sample file and writes missing values as -999

my_ukb_sample <- ukb_gen_read_sample("ukb.sample")

ukb_gen_write_bgenie(
   my_ukb_data,
   ukb.sample = my_ukb_sample,
   ukb.variables = c("height", "weight", "iq")
   path = "my_ukb_bgenie.pheno",
)

ukb_gen_write_bgenie(
   my_ukb_data,
   ukb.sample = my_ukb_sample,
   ukb.variables = c("age", "socioeconomic_status", "genetic_pcs")
   path = "my_ukb_bgenie.cov",
)

## End(Not run)
```

---

ukb_gen_write_plink *Writes a PLINK format phenotype or covariate file*

Description

This function writes a space-delimited file with header, with the obligatory first two columns FID and IID. Use this function to write phenotype and covariate files for downstream genetic analysis in plink - the format is the same.

Usage

```
ukb_gen_write_plink(x, path, ukb.variables, ukb.id = "eid",
  na.strings = "NA")
```

Arguments

x	A UKB dataset.
path	A path to a file.
ukb.variables	A character vector of either the phenotypes for a PLINK phenotype file, or covariates for a PLINK covariate file.
ukb.id	The id variable name (default = "eid").
na.strings	String used for missing values. Defaults to NA.

Details

The function writes the id variable in your dataset to the first two columns of the output file with the names FID and IID - you do not need to have two id columns in the data.frame passed to the argument `x`. Use the `--pheno-name` and `--covar-name` PLINK flags to select columns by name. See the PLINK documentation for the `--pheno`, `--mpheno`, `--pheno-name`, and `--covar`, `--covar-name`, `--covar-number` flags.

See Also

`ukb_gen_read_sample` to read a sample file, and `ukb_gen_write_bgenie` to write phenotype and covariate files to BGENIE format.

Examples

```
## Not run:

# Automatically inserts FID IID columns required by PLINK

ukb_gen_write_plink(
   my_ukb_data,
   path = "my_ukb_plink.pheno",
   ukb.variables = c("height", "weight", "iq")
)

ukb_gen_write_plink(
   my_ukb_data,
   path = "my_ukb_plink.cov",
   ukb.variables = c("age", "socioeconomic_status", "genetic_pcs")
)

## End(Not run)
```

ukb_gen_write_plink_excl *Writes a PLINK format file for combined exclusions*

Description

Defunct. See help("ukb_defunct"). Writes a combined exclusions file including UKB recommended exclusions, heterozygosity exclusions (+/- 3*sd from mean), genetic ethnicity exclusions (based on the UKB genetic ethnic grouping variable, field 1002), and relatedness exclusions (a randomly-selected member of each related pair). For exclusion of individuals from a genetic analysis, the PLINK flag `--remove` accepts a space/tab-delimited text file with family IDs in the first column and within-family IDs in the second column (i.e., FID IID), without a header.

Usage

```
ukb_gen_write_plink_excl(path)
```

Arguments

path	A path to a file.

See Also

ukb_gen_meta, ukb_gen_pcs which retrieve variables to be included in a covariate file. ukb_gen_excl_to_na to update a phenotype with NAs for samples to-be-excluded based on genetic metadata, and ukb_gen_write_plink and ukb_gen_write_bgenie

Examples

```
## Not run:
# Supply name of a file to write PLINK format combined exclusions
ukb_gen_write_plink_excl("combined_exclusions.txt")

## End(Not run)
```

ukb_icd_code_meaning *Retrieves description for a ICD code.*

Description

Retrieves description for a ICD code.

Usage

```
ukb_icd_code_meaning(icd.code, icd.version = 10)
```

Arguments

icd.code	The ICD diagnosis code to be looked up.
icd.version	The ICD version (or revision) number, 9 or 10.

See Also

ukb_icd_diagnosis, ukb_icd_keyword, ukb_icd_prevalence

Examples

```
ukb_icd_code_meaning(icd.code = "I74", icd.version = 10)
```

---

ukb_icd_diagnosis *Retrieves diagnoses for an individual.*

Description

Retrieves diagnoses for an individual.

Usage

```
ukb_icd_diagnosis(data, id, icd.version = NULL)
```

Arguments

data	A UKB dataset (or subset) created with ukb_df.
id	An individual's id, i.e., their unique eid reference number.
icd.version	The ICD version (or revision) number, 9 or 10.

See Also

ukb_df, ukb_icd_code_meaning, ukb_icd_keyword, ukb_icd_prevalence

Examples

```
## Not run:
ukb_icd_diagnosis(my_ukb_data, id = "123456", icd.version = 10)

## End(Not run)
```

ukb_icd_freq_by *Frequency of an ICD diagnosis by a target variable*

Description

Produces either a dataframe of diagnosis frequencies or a plot. For a quantitative reference variable (e.g. BMI), the plot shows frequency of diagnosis within each group (deciles of the reference variable by default) at the (max - min) / 2 for each group.

Usage

```
ukb_icd_freq_by(data, reference.var, n.groups = 10,
  icd.code = c("^(I2[0-5])", "^(I6[0-9])",
  "^(J09|J1[0-9]|J2[0-2]|P23|U04)"),
  icd.labels = c("coronary artery disease", "cerebrovascular disease",
  "lower respiratory tract infection"), plot.title = "",
  legend.col = 1, legend.pos = "right", icd.version = 10,
  freq.plot = FALSE, reference.lab = "Reference variable",
  freq.lab = "UKB disease frequency")
```

Arguments

data	A UKB dataset (or subset) created with ukb_df.
reference.var	UKB ICD frequencies will be calculated by levels of this variable. If continuous, by default it is cut into 10 intervals of approximately equal size (set with n.groups).
n.groups	Number of approximately equal-sized groups to split a continuous variable into.
icd.code	ICD disease code(s) e.g. "I74". Use a regular expression to specify a broader set of diagnoses, e.g. "I" captures all Diseases of the circulatory system, I00-I99, "C|D[0-4]." captures all Neoplasms, C00-D49. Default is the WHO top 3 causes of death globally in 2015, see http://www.who.int/healthinfo/global_burden_disease/GlobalCOD_method_2000_2015.pdf?ua=1. Note. If you specify 'icd.codes', you must supply corresponding labels to 'icd.labels'.
icd.labels	Character vector of ICD labels for the plot legend. Default = V1 to VN.
plot.title	Title for the plot. Default describes the default icd.codes, WHO top 6 cause of death 2015.
legend.col	Number of columns for the legend. (Default = 1).
legend.pos	Legend position, default = "right".
icd.version	The ICD version (or revision) number, 9 or 10.
freq.plot	If TRUE returns a plot of ICD diagnosis by target variable. If FALSE (default) returns a dataframe.
reference.lab	An x-axis title for the reference variable.
freq.lab	A y-axis title for disease frequency.

ukb_icd_keyword *Retrieves diagnoses containing a description.*

Description

Returns a dataframe of ICD code and descriptions for all entries including any supplied keyword.

Usage

```
ukb_icd_keyword(description, icd.version = 10, ignore.case = TRUE)
```

Arguments

description	A character vector of one or more keywords to be looked up in the ICD descriptions, e.g., "cardio", c("cardio", "lymphoma"). Each keyword can be a regular expression, e.g. "lymph*".
icd.version	The ICD version (or revision) number, 9 or 10. Default = 10.
ignore.case	If 'TRUE' (default), case is ignored during matching; if 'FALSE', the matching is case sensitive.

See Also

ukb_icd_diagnosis, ukb_icd_code_meaning, ukb_icd_prevalence

Examples

```
ukb_icd_keyword("cardio", icd.version = 10)
```

ukb_icd_prevalence *Returns the prevalence for an ICD diagnosis*

Description

Returns the prevalence for an ICD diagnosis

Usage

```
ukb_icd_prevalence(data, icd.code, icd.version = 10)
```

Arguments

data	A UKB dataset (or subset) created with ukb_df.
icd.code	An ICD disease code e.g. "I74". Use a regular expression to specify a broader set of diagnoses, e.g. "I" captures all Diseases of the circulatory system, I00-I99, "C|D[0-4]." captures all Neoplasms, C00-D49.
icd.version	The ICD version (or revision) number, 9 or 10. Default = 10.

See Also

ukb_icd_diagnosis, ukb_icd_code_meaning, ukb_icd_keyword

Examples

```
## Not run:
# ICD-10 code I74, Arterial embolism and thrombosis
ukb_icd_prevalence(my_ukb_data, icd.code = "I74")

# ICD-10 chapter 9, disease block I00I99, Diseases of the circulatory system
ukb_icd_prevalence(my_ukb_data, icd.code = "I")

# ICD-10 chapter 2, C00-D49, Neoplasms
ukb_icd_prevalence(my_ukb_data, icd.code = "C|D[0-4].")

## End(Not run)
```

Index