Package ‘fingertipsR’

April 16, 2020

Type Package
Version 1.0.3
Title Fingertips Data for Public Health
Description Fingertips (<http://fingertips.phe.org.uk/>) contains data for many indicators of public health in England. The underlying data is now more easily accessible by making use of the API.
Maintainer Sebastian Fox <sebastian.fox@phe.gov.uk>
Depends R (>= 3.4.0)
URL https://fingertips.phe.org.uk,
https://github.com/ropensci/fingertipsR,
https://docs.ropensci.org/fingertipsR/
BugReports https://github.com/ropensci/fingertipsR/issues
Imports curl, dplyr, DT, httr, jsonlite, miniUI, readr, rlang, shiny, shinyjsloaders
Suggests ggplot2, knitr, pander, rmarkdown, testthat
VignetteBuilder knitr
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Author Sebastian Fox [aut, cre],
    Julian Flowers [aut, ctb],
    Simon Thelwall [ctb] (<https://orcid.org/0000-0002-0434-2724>),
    Duncan Gormansway [ctb],
    Carl Ganz [ctb],
    David Whiting [ctb],
    Crown Copyright 2018 [cph]
Repository CRAN
Date/Publication 2020-04-16 09:00:02 UTC
### add_timestamp

#### Description

Add timestamp onto end of api url to prevent caching issues

#### Usage

```r
add_timestamp(api_path)
```

#### Arguments

- `api_path` string; the API url to retrieve data from
Description

Outputs a data frame of area type ids, their descriptions, and how they map to parent area types. To understand more on mappings of areas, see the Where to start section of the Life Expectancy vignette.

Usage

area_types(AreaTypeName = NULL, AreaTypeID = NULL, ProfileID = NULL, path)

Arguments

AreaTypeName Character vector, description of the area type; default is NULL
AreaTypeID Numeric vector, the Fingertips ID for the area type; default is NULL
ProfileID Numeric vector, id of profiles of interest
path String; Fingertips API address. Function will default to the correct address

Value

A data frame of area type ids and their descriptions

See Also

indicators for indicator lookups, profiles for profile lookups, deprivation_decile for deprivation decile lookups, category_types for category lookups, indicator_areatypes for indicators by area types lookups, indicators_unique for unique indicator ids and their names, nearest_neighbours for a vector of nearest neighbours for an area and indicator_order for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: category_types(), deprivation_decile(), indicator_areatypes(), indicator_metadata(), indicator_order(), indicators_unique(), indicators(), nearest_neighbours(), profiles()

Examples

## Not run:
# Returns a data frame with all levels of area and how they map to one another
area_types()

# Returns a data frame of county and unitary authority mappings
area_types("counties")

# Returns a data frame of both counties, district
# and unitary authorities and their respective mappings
areas <- c("counties","district")
area_types(areas)

# Uses AreaTypeID to filter area types
area_types(AreaTypeID = 152)
## End(Not run)

category_types  Category types

Description

Outputs a data frame of category type ids, their name (along with a short name)

Usage

category_types(path)

Arguments

path  String: Fingertips API address. Function will default to the correct address

Value

A data frame of category type ids and their descriptions

See Also

indicators for indicator lookups, profiles for profile lookups, deprivation_decile for deprivation decile lookups, area_types for area type lookups, indicator_areatypes for indicators by area types lookups, indicators_unique for unique indicatorids and their names, nearest_neighbours for a vector of nearest neighbours for an area and indicator_order for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: area_types(), deprivation_decile(), indicator_areatypes(), indicator_metadata(), indicator_order(), indicators_unique(), indicators(), nearest_neighbours(), profiles()

Examples

## Not run:
# Returns the deprivation category types
cats <- category_types()
cats[cats$CategoryTypeId == 1,]
## End(Not run)
Description
Outputs a data frame allocating deprivation decile to area code based on the Indices of Multiple
Deprivation (IMD) produced by Department of Communities and Local Government

Usage
deprivation_decile(AreaTypeID, Year = 2019, path)

Arguments
AreaTypeID  Integer value; this function uses the IndicatorIDs 91872, 93275 and 93553,
please use the indicator_areatypes() function to see what AreaTypeIDs are
available
Year        Integer value, representing the year of IMD release to be applied, limited to 2015
or 2019
path        String; Fingertips API address. Function will default to the correct address

Details
This function uses the fingertips_data function to filter for the Index of multiple deprivation score
for the year and area supplied, and returns the area code, along with the score and the deprivation
decile, which is calculated using the ntile function from dplyr

Value
A lookup table providing deprivation decile and area code

See Also
indicators for indicator lookups, profiles for profile lookups, indicator_metadata for the
metadata for each indicator, area_types for area types and their parent mappings, category_types
for category lookups, indicator_areatypes for indicators by area types lookups, indicators_unique
for unique indicatorids and their names, nearest_neighbours for a vector of nearest neighbours
for an area and indicator_order for the order indicators are presented on the Fingertips website
within a Domain
Other lookup functions: area_types(), category_types(), indicator_areatypes(), indicator_metadata(),
indicator_order(), indicators_unique(), indicators(), nearest_neighbours(), profiles()

Examples
## Not run:
# Return 2019 deprivation scores for Sustainability and Transformation Footprints
deprivation_decile(120, 2019)
## End(Not run)
fingertipsR

A package for extracting the data behind the Fingertips website (https://fingertips.phe.org.uk/)

Description

The fingertipsR package provides two categories of important functions: lookup and data extract.

Lookup functions

The lookup functions are to provide users the ability to understand the ID inputs for the data extract functions.

Data extract functions

Using ID codes as inputs, the data extract functions allow the user to extract data from the Fingertips API.

fingertips_data

Fingertips data

Description

Outputs a data frame of data from Fingertips. Note, this function can take up to a few minutes to run (depending on internet connection speeds and parameter selection).

Usage

fingertips_data(
  IndicatorID = NULL,
  AreaCode = NULL,
  DomainID = NULL,
  ProfileID = NULL,
  AreaTypeID,
  ParentAreaTypeID = NULL,
  categorytype = FALSE,
  rank = FALSE,
  url_only = FALSE,
  path
)

**Arguments**

- **IndicatorID**: Numeric vector, id of the indicator of interest
- **AreaCode**: Character vector, ONS area code of area of interest
- **DomainID**: Numeric vector, id of domains of interest
- **ProfileID**: Numeric vector, id of profiles of interest. Indicator polarity can vary between profiles therefore if using one of the comparison fields it is recommended to complete this field as well as IndicatorID. If IndicatorID is populated, ProfileID can be ignored or must be the same length as IndicatorID (but can contain NAs).
- **AreaTypeID**: Numeric vector, the Fingertips ID for the area type. This argument accepts "All", which returns data for all available area types for the indicator(s), though this can take a long time to run
- **ParentAreaTypeID**: Numeric vector, the comparator area type for the data extracted; if NULL the function will use the first record for the specified ‘AreaTypeID’ from the area_types() function
- **categorytype**: TRUE or FALSE, determines whether the final table includes categorytype data where it exists. Default to FALSE
- **rank**: TRUE or FALSE, the rank of the area compared to other areas for that combination of indicator, sex, age, categorytype and category along with the indicator’s polarity. 1 is lowest NAs will be bottom and ties will return the average position. The total count of areas with a non-NA value are returned also in AreaValuesCount
- **url_only**: TRUE or FALSE, return only the url of the api call as a character vector
- **path**: String; Fingertips API address. Function will default to the correct address

**Details**

Note, polarity of an indicator is not automatically returned (e.g., whether a low value is good, bad or neither). Use the rank field for this to be returned (though it adds a lot of time to the query)

**Value**

A data frame of data extracted from the Fingertips API

**See Also**

Other data extract functions: `fingertips_redred()`

**Examples**

```r
## Not run:
# Returns data for the two selected domains at county and unitary authority geography
doms <- c(1000049,1938132983)
fingdata <- fingertips_data(DomainID = doms, AreaTypeID = 202)

# Returns data at local authority district geography (AreaTypeID = 101)
# for the indicator with the id 22401
```
fingdata <- fingertips_data(22401, AreaTypeID = 101)

# Returns same indicator with different comparisons due to indicator polarity
# differences between profiles on the website
# It is recommended to check the website to ensure consistency between your
# data extract here and the polarity required
fingdata <- fingertips_data(rep(90282,2),
    ProfileID = c(19,93),
    AreaTypeID = 202,
    AreaCode = "E06000008")
fingdata <- fingdata[order(fingdata$TimeperiodSortable, fingdata$Sex),]

# Returns data for all available area types for an indicator
fingdata <- fingertips_data(10101, AreaTypeID = "All")
## End(Not run)

---

**fingertips_endpoint**  
*Get the default fingertips API endpoint*

**Description**

Get the default fingertips API endpoint

**Usage**

fingertips_endpoint()

**Value**

A character string with the HTTP URL of the Fingertips API

---

**fingertips_ensure_api_available**  
*Check if the given Fingertips API endpoint is available*

**Description**

Check if the given Fingertips API endpoint is available

**Usage**

fingertips_ensure_api_available(endpoint = fingertips_endpoint())

**Arguments**

- **endpoint** string, the API base URL to check
Value

TRUE if the API is available, otherwise stop() is called.

Description

Filters data returned by the fingertips_data function for values for areas that are trending statistically significantly worse and the spot value is significantly worse than the comparator (England or Parent) value in the latest year of that indicator

Usage

fingertips_redred(Comparator = "England", ...)

Arguments

Comparator String, either "England" or "Parent" to determine which field to compare the spot value significance to

... Parameters provided to fingertips_data()

Value

A data frame of data extracted from the Fingertips API

See Also

Other data extract functions: fingertips_data()

Examples

```r
## Not run:
# Returns data for the two selected domains at county and unitary authority geography
reddata <- fingertips_redred(ProfileID = 26, AreaTypeID = 182)
## End(Not run)
```
fingertips_stats

High level statistics on Fingertips data

Description
A sentence that summarises the number of indicators, unique indicators and profiles

Usage
fingertips_stats()

Value
A string that summarises the high level statistics of indicators and profiles in Fingertips

Examples
## Not run:
# Returns a sentence describing number of indicators and profiles in Fingertips
fingertips_stats()
## End(Not run)

get_fingertips_api

Retrieve data from a given Fingertips API url

Description
Retrieve data from a given Fingertips API url

Usage
get_fingertips_api(api_path)

Arguments
api_path string; the API url to retrieve data from
**indicators**

*Live indicators and the profiles and domains they belong to*

**Description**

Outputs a data frame of indicators within a profile or domain

**Usage**

```r
indicators(ProfileID = NULL, DomainID = NULL, path)
```

**Arguments**

- **ProfileID**: Numeric vector, id of profiles of interest
- **DomainID**: Numeric vector, id of domains of interest
- **path**: String; Fingertips API address. Function will default to the correct address

**Value**

A data frame of indicators within a profile or domain.

**See Also**

- `area_types` for area type and their parent mappings
- `indicator_metadata` for indicator metadata
- `profiles` for profile lookups
- `deprivation_decile` for deprivation decile lookups
- `category_types` for category lookups
- `indicator_areatypes` for indicators by area types lookups
- `indicators_unique` for unique indicator ids and their names
- `nearest_neighbours` for a vector of nearest neighbours for an area
- `indicator_order` for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: `area_types()`, `category_types()`, `deprivation_decile()`, `indicator_areatypes()`, `indicator_metadata()`, `indicator_order()`, `indicators_unique()`, `nearest_neighbours()`, `profiles()`

**Examples**

```r
## Not run:
# Returns a complete data frame of indicators and their domains and profiles
indicators()

# Returns a data frame of all of the indicators in the Public Health Outcomes Framework
indicators(ProfileID = 19)
## End(Not run)
```
**indicators_unique**  
*Live indicators*

**Description**  
Outputs a data frame of indicators (their id and name only). Note, this function can take up to a few minutes to run (depending on internet connection speeds).

**Usage**  
```
indicators_unique(ProfileID = NULL, DomainID = NULL, path)
```

**Arguments**  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProfileID</td>
<td>Numeric vector, id of profiles of interest</td>
</tr>
<tr>
<td>DomainID</td>
<td>Numeric vector, id of domains of interest</td>
</tr>
<tr>
<td>path</td>
<td>String: Fingertips API address. Function will default to the correct address</td>
</tr>
</tbody>
</table>

**Value**  
A data frame of indicator ids and names

**See Also**
- `indicators` for indicators and their parent domains and profiles,  
- `area_types` for area type and their parent mappings,  
- `indicator_metadata` for indicator metadata and  
- `profiles` for profile lookups and  
- `deprivation_decile` for deprivation decile lookups and  
- `category_types` for category lookups,  
- `indicator_areatypes` for indicators by area types lookups and  
- `indicator_order` for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: `area_types()`, `category_types()`, `deprivation_decile()`, `indicator_areatypes()`, `indicator_metadata()`, `indicator_order()`, `indicators()`, `nearest_neighbours()``, `profiles()`

**Examples**
```
## Not run:
indicators_unique(ProfileID = 21)
## End(Not run)
```
**indicator_areatypes**  

*Area types by indicator*

**Description**

Outputs a data frame of indicator ids and the area type ids that exist for that indicator

**Usage**

```
indicator_areatypes(IndicatorID, AreaTypeID, path)
```

**Arguments**

- `IndicatorID` integer; the Indicator ID (can be ignored or of length 1). Takes priority over AreaTypeID if both are entered
- `AreaTypeID` integer; the Area Type ID (can be ignored or of length 1)
- `path` String; Fingertips API address. Function will default to the correct address

**Value**

A data frame of indicator ids and area type ids

**See Also**

- `indicators` for indicator lookups, `profiles` for profile lookups, `deprivation_decile` for deprivation decile lookups, `area_types` for area type lookups, `category_types` for category type lookups, `indicators_unique` for unique indicator ids and their names, `nearest_neighbours` for a vector of nearest neighbours for an area and `indicator_order` for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: `area_types()`, `category_types()`, `deprivation_decile()`, `indicator_metadata()`, `indicator_order()`, `indicators_unique()`, `indicators()`, `nearest_neighbours()`, `profiles()`

**Examples**

```r
## Not run:
indicator_areatypes(IndicatorID = 10101)
## End(Not run)
```
indicator_metadata

Description

Outputs a data frame containing the metadata for selected indicators. Note, this function can take up to a few minutes to run (depending on internet connection speeds)

Usage

indicator_metadata(IndicatorID = NULL, DomainID = NULL, ProfileID = NULL, path)

Arguments

IndicatorID  Numeric vector, id of the indicator of interest
DomainID  Numeric vector, id of domains of interest
ProfileID  Numeric vector, id of profiles of interest. Indicator polarity can vary between profiles therefore if using one of the comparison fields it is recommended to complete this field as well as IndicatorID. If IndicatorID is populated, ProfileID can be ignored or must be the same length as IndicatorID (but can contain NAs).
path  String; Fingertips API address. Function will default to the correct address

Value

The metadata associated with each indicator/domain/profile identified

See Also

indicators for indicator lookups, profiles for profile lookups, deprivation_decile for deprivation lookups, area_types for area types and their parent mappings, category_types for category lookups, indicator_areatypes for indicators by area types lookups, indicators_unique for unique indicatorids and their names, nearest_neighbours for a vector of nearest neighbours for an area and indicator_order for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: area_types(), category_types(), deprivation_decile(), indicator_areatypes(), indicator_order(), indicators_unique(), indicators(), nearest_neighbours(), profiles()

Examples

```r
## Not run:
# Returns metadata for indicator ID 90362 and 1107
indicatorIDs <- c(90362, 1107)
indicator_metadata(indicatorIDs)

# Returns metadata for the indicators within the domain 1000101
indicator_metadata(DomainID = 1000101)
```
# Returns metadata for the indicators within the profile with the ID 129
indicator_metadata(ProfileID = 129)
## End(Not run)

---

## indicator_order

### Indicator order number

#### Description

Outputs a tibble of indicator ids and their sequence number for the provided domain and area type. This enables the user to order the indicators as they are ordered on the Fingertips website.

#### Usage

```r
indicator_order(DomainID, AreaTypeID, ParentAreaTypeID, path)
```

#### Arguments

- **DomainID**: Numeric vector, id of domains of interest
- **AreaTypeID**: Numeric vector, the Fingertips ID for the area type. This argument accepts "All", which returns data for all available area types for the indicator(s), though this can take a long time to run
- **ParentAreaTypeID**: Numeric vector, the comparator area type for the data extracted; if NULL the function will use the first record for the specified 'AreaTypeID' from the area_types() function
- **path**: String; Fingertips API address. Function will default to the correct address

#### Value

A data frame of indicator ids and sequence number

#### See Also

- `indicators` for indicators and their parent domains and profiles, `area_types` for area type and their parent mappings, `indicator_metadata` for indicator metadata, `profiles` for profile lookups, `deprivation_decile` for deprivation decile lookups, `category_types` for category lookups, `indicator_areatypes` for indicators by area types lookups and `nearest_neighbours` for a vector of nearest neighbours for an area

Other lookup functions: `area_types()`, `category_types()`, `deprivation_decile()`, `indicator_areatypes()`, `indicator_metadata()`, `indicators_unique()`, `indicators()`, `nearest_neighbours()`, `profiles()`

#### Examples

```r
## Not run:
indicator_order(DomainID = 1938133161, AreaTypeID = 102, ParentAreaTypeID = 6)
## End(Not run)
```
nearest_neighbours  Nearest neighbours

Description

Outputs a character vector of similar areas for given area. Currently returns similar areas for Clinical Commissioning Groups (old and new) based on NHS England’s similar CCG explorer tool or lower and upper tier local authorities based on CIPFA’s Nearest Neighbours Model or upper tier local authorities based on Children’s services statistical neighbour benchmarking tool

Usage

nearest_neighbours(AreaCode, AreaTypeID = 101, measure, path)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AreaCode</td>
<td>Character vector, ONS area code of area of interest</td>
</tr>
<tr>
<td>AreaTypeID</td>
<td>AreaTypeID of the nearest neighbours (see area_types) for IDs. Only returns information on AreaTypeIDs 101, 102, 201, 202, 152, 153 and 154</td>
</tr>
<tr>
<td>measure</td>
<td>string; when AreaTypeID = 102 measure must be either &quot;CIPFA&quot; for CIPFA local authority nearest neighbours or &quot;CSSN&quot; for Children’s services statistical neighbours</td>
</tr>
<tr>
<td>path</td>
<td>String; Fingertips API address. Function will default to the correct address</td>
</tr>
</tbody>
</table>

Details

Use AreaTypeID = 102 for the AreaTypeID related to Children’s services statistical neighbours

Value

A character vector of area codes

See Also

indicators for indicator lookups, profiles for profile lookups, deprivation_decile for deprivation decile lookups, area_types for area type lookups, category_types for category type lookups, indicators_unique for unique indicatorids and their names, indicator_areatypes for indicators by area types lookups and indicator_order for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: area_types(), category_types(), deprivation_decile(), indicator_areatypes(), indicator_metadata(), indicator_order(), indicators_unique(), indicators(), profiles()

Examples

```r
## Not run:
nearest_neighbours(AreaCode = "E38000002", AreaTypeID = 154)
## End(Not run)
```
**profiles**

### Live profiles

**Description**

Outputs a data frame of live profiles that data are available for in Fingertips http://fingertips.phe.org.uk/

**Usage**

```r
profiles(ProfileID = NULL, ProfileName = NULL, path)
```

**Arguments**

- **ProfileID** Numeric vector, id of profiles of interest
- **ProfileName** Character vector, full name of profile(s)
- **path** String; Fingertips API address. Function will default to the correct address

**Value**

A data frame of live profile ids and names along with their domain names and ids.

**See Also**

- `area_types` for area type and their parent mappings,
- `indicators` for indicator lookups,
- `indicator_metadata` for indicator metadata,
- `deprivation_decile` for deprivation decile lookups,
- `category_types` for category lookups,
- `indicator_areatypes` for indicators by area types lookups,
- `indicators_unique` for unique indicatorids and their names,
- `nearest_neighbours` for a vector of nearest neighbours for an area and
- `indicator_order` for the order indicators are presented on the Fingertips website within a Domain

Other lookup functions: `area_types()`, `category_types()`, `deprivation_decile()`, `indicator_areatypes()`, `indicator_metadata()`, `indicator_order()`, `indicators_unique()`, `indicators()`, `nearest_neighbours()`

**Examples**

```r
## Not run:
# Returns a complete data frame of domains and their profiles
profiles()

# Returns a data frame of all of the domains in the Public Health Outcomes Framework
profiles(ProfileName = "Public Health Outcomes Framework")
## End(Not run)
```
**select_indicators**

---

**Description**

Point and click method of selecting indicators and assigning them to object. Note, this function can take up to a few minutes to run (depending on internet connection speeds).

**Usage**

```
select_indicators()
```

**Value**

A numeric vector of indicator IDs

**Examples**

```
## Not run:
# Opens a browser window allowing the user to select indicators by their name, domain and profile
inds <- select_indicators()
## End(Not run)
```
Index

add_timestamp, 2
area_types, 3, 4, 5, 11–17

category_types, 3, 4, 5, 11–17
deprivation_decile, 3, 4, 5, 11–17

fingertips_data, 6, 9
fingertips_endpoint, 8
fingertips.ensure_api_available, 8
fingertips_redred, 7, 9
fingertips_stats, 10
fingertipsR, 6

get_fingertips_api, 10

indicator_areatypes, 3–5, 11, 12, 13, 14–17
indicator_metadata, 3–5, 11–13, 14, 15–17
indicator_order, 3–5, 11–14, 15, 16, 17
indicators, 3–5, 11, 12–17
indicators_unique, 3–5, 11, 12, 13–17

nearest_neighbours, 3–5, 11–15, 16, 17
profiles, 3–5, 11–16, 17

select_indicators, 18