

Package ‘ripc’

September 6, 2023

Title Download and Tidy IPC and CH Data

Version 0.2.0

Description Utilities to access Integrated Food Security Phase Classification (IPC) and Cadre Harmonisé (CH) food security data. Wrapper functions are available for all of the 'IPC-CH' Public API (<<https://docs.api.ipcinfo.org>>) simplified and advanced endpoints to easily download the data in a clean and tidy format.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

URL <https://github.com/ocha-dap/ripc>

BugReports <https://github.com/ocha-dap/ripc/issues>

Imports countrycode, dplyr, httr, lifecycle, lubridate, magrittr, purrr, rlang, stringr, tidyr

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Seth Caldwell [aut, cre, cph]

Maintainer Seth Caldwell <caldwellst@gmail.com>

Repository CRAN

Date/Publication 2023-09-06 18:00:10 UTC

R topics documented:

| | |
|------------------------------|---|
| ipc_get | 2 |
| ipc_get_analyses | 2 |
| ipc_get_areas | 4 |
| ipc_get_country | 5 |
| ipc_get_icons | 7 |
| ipc_get_points | 8 |
| ipc_get_population | 9 |

Index**12**

| | |
|---------|----------------------------------|
| ipc_get | <i>Get resource from IPC API</i> |
|---------|----------------------------------|

Description

Back-end function used to drive the API calls of the other `ipc_get_...()` family of functions.

Usage

```
ipc_get(resource, api_key = NULL, ..., drop_metadata = FALSE)
```

Arguments

| | |
|---------------|-------------------------------------------------------------------------------------------------|
| resource | One of the resources exposed in the IPC API, such "areas" , "analyses", "points", or "country". |
| api_key | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| ... | Named parameters passed to the API call URL in the form of argument=value. |
| drop_metadata | Logical, whether or not to drop metadata from the original list return. |

Value

Data frame from the API

| | |
|------------------|------------------------------------------------|
| ipc_get_analyses | <i>Get analyses resources from the IPC API</i> |
|------------------|------------------------------------------------|

Description

Accesses the areas resources on the IPC API. Contains detailed analysis information. If country, year and/or type parameters are passed, accesses the **analyses** simplified API endpoint and pulls in all analyses or filtered to those parameters. To get the details for a specific analysis available on the **analyses/id/period** advanced API endpoint, pass in id. You cannot pass in both sets of parameters.

Usage

```
ipc_get_analyses(
  country = NULL,
  year = NULL,
  type = NULL,
  id = NULL,
  api_key = NULL,
  tidy_df = TRUE
)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| country | ISO2 country code. |
| year | Single numeric year to filter analysis, calculated from the maximum year of current period dates. If NULL, the default, returns data for all years. |
| type | Single string value of 'A' or 'C', corresponding to food security conditions, either acute or chronic. If NULL, the default, returns data for all types. |
| id | Analysis ID. |
| api_key | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| tidy_df | If TRUE, returns a tidy data frame wrangled as described in the Tidy section of the documentation. If FALSE, returns the data frame as returned direct from the IPC-CH API. |

Details

Analyses data is metadata related to specific analyses conducted by the IPC, including title of the analysis, link to its release page on the IPC website, and creation/modification dates.

Value

Data frame of analysis metadata. Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When tidy_df is TRUE, the following changes are made to the initial output to ensure each row represents a single analysis:

1. created and modified columns are converted from character columns to date columns.
2. id column is renamed to be analysis_id.

Examples

```
# get all analyses details from the simplified API
ipc_get_analyses()

# get analysis details for a specific analysis ID
ipc_get_analyses(id = 12856213)
```

ipc_get_areas

*Get areas resources from the IPC-CH API***Description**

Accesses the areas resources on the IPC-CH API. Contains detailed area and population data. If country, year and/or type parameters are passed, accesses the **areas** simplified API endpoint and pulls in all areas filtered to those parameters. To get all areas for a specific analysis and period, available on the **areas/id/period** advanced API endpoint, pass in id and period. You cannot pass in both sets of parameters.

Usage

```
ipc_get_areas(
  country = NULL,
  year = NULL,
  type = NULL,
  id = NULL,
  period = NULL,
  api_key = NULL,
  tidy_df = TRUE
)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| country | ISO2 country code. |
| year | Single numeric year to filter analysis, calculated from the maximum year of current period dates. If NULL, the default, returns data for all years. |
| type | Single string value of 'A' or 'C', corresponding to food security conditions, either acute or chronic. If NULL, the default, returns data for all types. |
| id | Analysis ID. |
| period | Period code, either 'C', 'P', or 'A' for current, projection, and second projection. |
| api_key | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| tidy_df | If TRUE, returns a tidy data frame wrangled as described in the Tidy section of the documentation. If FALSE, returns the data frame as returned direct from the IPC-CH API. |

Details

Areas data is the typical unit of analysis in IPC-CH outputs. These are typically administrative units (or clusters of them together). For each area, estimates of the population in each phase is provided and a general phase classification is assigned. Data is currently output only for current periods through **areas** endpoint. Use `ipc_get_population()` to get detailed population data and classifications for all analysis periods.

See the [IPC website](#) and [API documentation](#) for more information.

Value

Data frame of IPC and CH analysis at the areas level. Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When `tidy_df` is `TRUE`, the following changes are made to the initial output to ensure each row represents a single area analysis, and all estimates and values are stored as columns:

1. `analysis_period_start` and `analysis_period_end` created as `Date` columns from the `from` and `to` columns respectively, allocating the day of the start and end periods to be the 15th of the month.
2. `phases` is unnested from a list column to bring the phase data into the main data frame.
3. The population estimates are pivoted to a wider format with names `phase#_num` and `phase#_pct`.
4. `title` column is renamed to be `area_name`, `anl_id` to `analysis_id`, and `id` and `aar_id` are changed to `area_id`.

Examples

```
# get all areas from the simplified API
ipc_get_areas()

# get areas for specific analysis ID and period from advanced API
ipc_get_areas(id = 12856213, period = "P")
```

| | |
|------------------------------|-----------------------------------------------|
| <code>ipc_get_country</code> | <i>Get country resources from the IPC API</i> |
|------------------------------|-----------------------------------------------|

Description

Accesses the country resources on the IPC API. Contains detailed country-level data. If `country`, `year` and/or `type` parameters are passed, accesses the **country** simplified API endpoint and pulls in all country data filtered to those parameters.

Usage

```
ipc_get_country(  
  country = NULL,  
  year = NULL,  
  type = NULL,  
  api_key = NULL,  
  tidy_df = TRUE  
)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| country | ISO2 country code. |
| year | Single numeric year to filter analysis, calculated from the maximum year of current period dates. If NULL, the default, returns data for all years. |
| type | Single string value of 'A' or 'C', corresponding to food security conditions, either acute or chronic. If NULL, the default, returns data for all types. |
| api_key | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| tidy_df | If TRUE, returns a tidy data frame wrangled as described in the Tidy section of the documentation. If FALSE, returns the data frame as returned direct from the IPC-CH API. |

Details

Country data is the highest level of aggregation for IPC analysis. Data is the population estimates and other details aggregated from area and point analyses within the country for that specific analysis. No phase classifications are made at the country level. Data is currently output only for current periods through the **country** endpoint. Use `ipc_get_population()` to get detailed population data for all analysis periods.

See the [IPC website](#) and [API documentation](#) for more information.

Value

Data frame of IPC and CH analysis at the country level. Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When `tidy_df` is TRUE, the following changes are made to the initial output to ensure each row represents a single country analysis, and all estimates and values are stored as columns:

1. `analysis_period_start` and `analysis_period_end` created as Date columns from the `from` and `to` columns respectively, allocating the day of the start and end periods to be the 15th of the month.
2. `phases` is unnested from a list column to bring the phase data into the main data frame.
3. The population estimates are pivoted to a wider format with names `phase#_num` and `phase#_pct`.
4. `id` column is renamed to be `analysis_id`.

Examples

```
# get all areas from the simplified API
ipc_get_country()

# get country data just for Somalia
ipc_get_country(country = "SO")
```

`ipc_get_icons`*Get icons resources from the IPC API*

Description

Accesses the icons resources on the IPC API. Contains links from analysis and area IDs to the icon resource the IPC uses in maps and publications. This is likely **not useful for the general user**. If year and/or type parameters are passed, accesses the **icons** simplified API endpoint, pulls in data filtered to those parameters. To get all icons for a specific analysis and period, available on the **types/id/period** advanced API endpoint, pass in id and period. You cannot pass in both sets of parameters.

Usage

```
ipc_get_icons(  
  year = NULL,  
  type = NULL,  
  id = NULL,  
  period = NULL,  
  api_key = NULL,  
  tidy_df = TRUE  
)
```

Arguments

| | |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>year</code> | Single numeric year to filter analysis, calculated from the maximum year of current period dates. If NULL, the default, returns data for all years. |
| <code>type</code> | Single string value of 'A' or 'C', corresponding to food security conditions, either acute or chronic. If NULL, the default, returns data for all types. |
| <code>id</code> | Analysis ID. |
| <code>period</code> | Period code, either 'C', 'P', or 'A' for current, projection, and second projection. |
| <code>api_key</code> | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| <code>tidy_df</code> | If TRUE, returns a tidy data frame wrangled as described in the Tidy section of the documentation. If FALSE, returns the data frame as returned direct from the IPC-CH API. |

Details

Icons data is used internally by the IPC to link up analyses for areas and a relevant icon for display on maps and in publications. The icons are stored in an Amazon S3 bucket that is not publicly accessible and therefore not useful for general users. Provided here for convenience.

See the [IPC website](#) and [API documentation](#) for more information.

Value

Data from of icons for analysis publications. Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When `tidy_df` is `TRUE`, `aar_id` is renamed to `area_id` and `area` to `area_name`.

Examples

```
# get all icons from the simplified API
ipc_get_icons()

# get icons for specific analysis ID and period from advanced API
ipc_get_icons(id = 12135625, period = "C")
```

```
ipc_get_points
```

```
Get points resources from the IPC API
```

Description

Accesses the points resources on the IPC API. Contains detailed area and population data. If year and/or type parameters are passed, accesses the **types** simplified API endpoint and pulls in all types data or filtered to either year or type. To get all types for a specific analysis and period, available on the **types/id/period** advanced API endpoint, pass in `id` and `period`. You cannot pass in both sets of parameters.

Usage

```
ipc_get_points(
  year = NULL,
  type = NULL,
  id = NULL,
  period = NULL,
  api_key = NULL,
  tidy_df = TRUE
)
```

Arguments

| | |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>year</code> | Single numeric year to filter analysis, calculated from the maximum year of current period dates. If <code>NULL</code> , the default, returns data for all years. |
| <code>type</code> | Single string value of 'A' or 'C', corresponding to food security conditions, either acute or chronic. If <code>NULL</code> , the default, returns data for all types. |
| <code>id</code> | Analysis ID. |
| <code>period</code> | Period code, either 'C', 'P', or 'A' for current, projection, and second projection. |
| <code>api_key</code> | IPC API key. If <code>NULL</code> (the default), looks for <code>IPC_API_KEY</code> in the environment. |
| <code>tidy_df</code> | If <code>TRUE</code> , returns a tidy data frame wrangled as described in the Tidy section of the documentation. If <code>FALSE</code> , returns the data frame as returned direct from the IPC-CH API. |

Details

Points data is IPC data generated from analysis on geographic entities that are smaller than the standard areas. These are typically urban areas or displacement sites where food insecurity conditions might differ significantly from the wider context and justify specific analysis. Population estimates as well as an overall phase classification are provided for each point. Data is currently output only for current periods through the **points** endpoint. Use `ipc_get_population()` to get detailed population data and classifications for all analysis periods. Points data is contained within the areas dataset returned from `ipc_get_population()`.

See the [IPC website](#) and [API documentation](#) for more information.

Value

Data frame of IPC and CH analysis at the point level. Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When `tidy_df` is TRUE, the following changes are made to the initial output to ensure each row represents a single point analysis, and all estimates and values are stored as columns:

1. phases is unnested from a list column to bring the phase data into the main data frame.
2. The population estimates are pivoted to a wider format with names `phase#_num` and `phase#_pct`.
3. `aar_id` is renamed to `area_id` and `anl_id` to `analysis_id`.

Examples

```
# get all areas from the simplified API
ipc_get_points()

# get areas for specific analysis ID and period from advanced API
ipc_get_points(id = 18978466, period = "P")
```

`ipc_get_population` *Get population resources from the IPC API*

Description

Accesses the population resources on the IPC API. Contains detailed population data. If country and/or start and end parameters are passed, accesses the **population** advanced API endpoint and pulls in all data. filtered by those parameters. To get the population data for a specific analysis, available on the **types/id** advanced API endpoint, pass in `id`. You cannot pass in both sets of parameters.

Usage

```
ipc_get_population(  
  country = NULL,  
  start = NULL,  
  end = NULL,  
  id = NULL,  
  api_key = NULL,  
  tidy_df = TRUE  
)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| country | ISO2 country code. |
| start | Start year. |
| end | End year. |
| id | Analysis ID. |
| api_key | IPC API key. If NULL (the default), looks for IPC_API_KEY in the environment. |
| tidy_df | If TRUE, returns a tidy data frame wrangled as described in the Tidy section of the documentation. If FALSE, returns the data frame as returned direct from the IPC-CH API. |

Details

Unlike the other `ipc_get_...()` family of functions, this returns a list of datasets, corresponding to country, areas, and groups data. The benefit of `ipc_get_population()` is that the returned data for each level of analysis contains all periods of analysis.

Groups data, where available, are geographies within a country that comprise multiple areas and/or points. Areas and points data is the lowest level of IPC analysis where population estimates for each phase are provided and a general area-level classification is made. There is no phase classification at the group level, but populations in each phase are provided. The same applies to country-level data.

These datasets are available elsewhere through:

- Country data: [ipc_get_country\(\)](#)
- Areas data: [ipc_get_areas\(\)](#)
- Groups data: Not available through other functions

See the respective function documentation for more details on what each dataset comprises or the [IPC website](#) and [API documentation](#) for more detailed and comprehensive information on the data and analysis.

Value

A list of 3 data frames:

- Country data frame.
- Areas data frame.

- Groups data frame.

Refer to the [IPC-CH Public API documentation](#) for details on the returned values, with variables described in full in the [extended documentation](#).

Tidy

When `tidy_df` is `TRUE`, the data returned from the population end point is transformed into a list of 3 data frames to ensure that each row represents a single analysis, and all estimates and values are stored as columns, while data at different levels of aggregation are in completely separate data frames. The steps are:

1. `analysis_period_start` and `analysis_period_end` created as `Date` columns from the `period_dates` column respectively, allocating the day of the start and end periods to be the 15th of the month.
2. `analysis_date` converted to a date column, using the 15th day of the month.
3. `phases` is unnested from a list column to bring the phase data into the main data frame.
4. The population estimates are pivoted to a wider format with names `phase#_num` and `phase#_pct`.
5. `id` column renamed to `analysis_id`.
6. Data frames are split out so multiple aggregations not present in a single single data frame.

Examples

```
# get all populations from the simplified API
ipc_get_population()

# get populations for specific analysis ID from advanced API
ipc_get_population(id = 12856213)
```

Index

`ipc_get`, [2](#)
`ipc_get_analyses`, [2](#)
`ipc_get_areas`, [4](#)
`ipc_get_areas()`, [10](#)
`ipc_get_country`, [5](#)
`ipc_get_country()`, [10](#)
`ipc_get_icons`, [7](#)
`ipc_get_points`, [8](#)
`ipc_get_population`, [9](#)
`ipc_get_population()`, [4, 6, 9](#)