Package: CoronaNetR (via r-universe)

September 13, 2024

Type Package

Title API Access to 'CoronaNet' Data

Version 0.3.0

Maintainer Robert Kubinec <rmk7@nyu.edu>

Description Offers access to a database on government responses to the
COVID-19 pandemic. To date, the 'CoronaNet' dataset provides
the most comprehensive and granular documentation of such
government policies in the world, capturing data for 20 broad
policy categories alongside many other dimensions, including
the initiator, target, and timing of a policy. This package is
a programmatic front-end to up-to-date 'CoronaNet' policy
records and the 'CoronaNet' policy intensity index scores. For
more information, see Cheng et al. (2020)
<doi:10.1038/s41562-020-0909-7>.

License MIT + file LICENSE

Depends R (>= 2.10), httr, R.utils

Imports readr

Suggests rmarkdown, knitr

Encoding UTF-8

Language en-US

LazyData true

URL https://github.com/CoronaNetDataScience/CoronaNetR

BugReports https://github.com/CoronaNetDataScience/CoronaNetR/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

VignetteBuilder knitr

Repository https://coronanetdatascience.r-universe.dev

RemoteUrl https://github.com/coronanetdatascience/coronanetr

RemoteRef HEAD

RemoteSha 1c2fae0d33519c88a6fbe673b63718124813bfed

Contents

---

get_event   *Access CoronaNet Event Data API*

Description

Use this function to obtain the latest policy event data from CoronaNet via an http API.

Usage

```
get_event(
  countries = "All",
  type = "All",
  type_sub_cat = "All",
  default_columns = c("record_id", "policy_id", "entry_type", "update_type",
   "update_level", "update_level_var", "date_announced", "date_start", "date_end",
   "date_end_spec", "country", "init_country_level", "province", "target_init_same",
     "target_country", "target_province", "target_city", "target_intl_org",
     "target_other", "target_who_what", "target_who_gen", "target_direction",
     "compliance", "enforcer", "city", "type", "type_sub_cat", "description"),
  additional_columns = NULL,
  from = "2019-12-31",
  to = "2022-01-01",
  include_no_end_date = TRUE,
  time_out = FALSE
)
```

Arguments

countries	A character vector of country name(s), e.g., c("Yemen", "Saudi Arabia"). "All" is used as the default.
type	A character vector of policy types, e.g., c("Lockdown", "Curfew"). "All" is used as the default. See https://www.coronanet-project.org/taxonomy.html? for a list of policy types.
type_sub_cat	A character vector of policy types, e.g., c("Self-testing", "Drive-in testing centers"). "All" is used as the default. See https://www.coronanet-project.org/taxonomy.html? for a list of policy subtypes and their related policy types.
default_columns	
	A character vector specifying the minimum set of columns of data to retrieve. Defaults to record/policy ID, dates, policy targets, policy type and sub-type, and description

additional_columns

> By default NULL. Select additional columns to include with the query.

from A character vector for the earliest start date, e.g., "2019-12-31".

to A character vector for the last end date, e.g., "2019-06-01".

include_no_end_date

> TRUE/FALSE - whether to include policy records that do not yet have an end date. By default set to TRUE (this is a lot of records).

time_out Whether to set a 5-second time-out on the API call. Beyond 5 seconds, the function will return an empty data-frame. Only useful for complying with CRAN submission requirements. Default is FALSE.

Details

This function offers programmatic access to the CoronaNet public release dataset, comprising over 80,000 distinct policy records and 93 fields. The dataset is updated regularly as policy coding continues. The entire dataset can be downloaded through this function, although by default it selects a subset of columns (see argument details below). To access additional columns, use the additional_columns argument and include a character vector of column names. For a full list of possible columns, see the CoronaNet codebook.

For more information about the data creation, see our paper.

Citation:

Cheng, Cindy; Barcelo, Joan; Spencer Hartnett, Allison; Kubinec, Robert and Messerschmidt, Luca. "COVID-19 Government Response Event Dataset (CoronaNet v1.0)." **Nature Human Behavior** 4, pp. 756-768 (2020).

See code examples for demonstration of filtering syntax.

Value

A dataframe with one record per COVID-19 policy

Examples

```
# Grab all data for Saudi Arabia from first 4 months of pandemic

saudi_data <- get_event(countries = "Saudi Arabia", type = "All",
type_sub_cat = "All", from = "2019-12-31", to = "2020-04-30")

# Each row is one policy record
saudi_data

# Use the additional_columns argument to add additional columns
# beyond the default set
# In this case, we'll add the link column to get the URLs
# for underlying public sources
saudi_data_links <- get_event(countries = "Saudi Arabia", type = "All",
                              type_sub_cat = "All",
                              from = "2019-12-31", to = "2020-04-30",
                              additional_columns = "link")
```

```
head(saudi_data_links$link)

# look at a specific policy type

saudi_data_subcat <- get_event(countries = "Saudi Arabia",
                               type = "Lockdown",
                               type_sub_cat = "All",
                               from = "2019-12-31", to = "2020-04-30")

head(saudi_data_subcat$description)
```

---

get_policy_scores   *Download Policy Intensity Scores*

Description

This function accesses the latest policy intensity scores showing the level of COVID-19 policy-making activity in a given country with measurement error.

Usage

```
get_policy_scores(
  countries = "All",
  type = "All",
  from = "2019-12-31",
  to = "2021-07-01",
  time_out = FALSE
)
```

Arguments

countries	Specify a specific country to query By default all countries.
type	Specify a specific index to query. By default all types.
from	The beginning time period in YYYY-MM-DD format.
to	The end time period in YYYY-MM-DD format. At present the index goes until 04-29-2021.
time_out	Whether to set a 5-second time-out on the API call. Beyond 5 seconds, the function will return an empty data-frame. Only useful for complying with CRAN submission requirements. Default is FALSE.

Details

Use this function to access the latest policy intensity scores for six types: ,

1. Business
2. Health Monitoring

3. Health Resources

4. Masks

5. Schools

6. Social Distancing

By default, all six indices for all countries are downloaded, running from the index start at January 1, 2020 until April 29, 2021. The indices are periodically updated with new data as the CoronaNet project continues coding policies and integrating external datasets.

The scores are Normally-distributed with a mean of 0. It is possible to re-scale the scores but we recommend using the default scale to preserve relationships between units and time points.

You can read more about the index construction and evaluation in our working paper.

Citation:

Kubinec, Robert; Barcelo, Joan; Goldzsmidt, Rafael; Grujic, Vanja; Model, Timothy; Schenk, Caress; Cheng, Cindy; Hale, Thomas; Spencer Hartnett, Allison; Messerschmidt, Luca; Petherick, Anna, and Thorvaldsdottir, Svanhildur. "Cross-National Measures of the Intensity of COVID-19 Public Health Policies." SocArchiv (2022). https://osf.io/preprints/socarxiv/rn9xk/

Because the indices were produced with a measurement model, they include measurement error. The most likely estimate is in the med_estimate column, and the uncertainty interval high/low are in the low_estimate\codehigh_estimate columns. The SD_estimate column has the standard deviation of the measurement error. These measurement error estimates can be used either in a model that explicitly incorporates the SD of the measurement error (so-called errors-in-variables models) or by using the low_estimate and high_estimate scores as robustness checks. The default for estimation and modeling should be the most likely med_estimate column.

Value

A data frame with one row per policy intensity score per country

Examples

```
# Download policy intensity scores for Japan and China

japan_scores <- get_policy_scores(countries=c("Japan","China"),
                from="2020-01-01",
                to="2020-01-05",
                 time_out=TRUE)
```

Index