# Package 'states'

December 11, 2020

**Type** Package

**Title** Create Panels of Independent States

**Version** 0.3.0

**Maintainer** Andreas Beger <adbeger@gmail.com>

**Description** Create panel data consisting of independent states from 1816 to the present. The package includes the Gleditsch & Ward (G&W) and Correlates of War (COW) lists of independent states, as well as helper functions for working with state panel data and standardizing other data sources to create country-year/month/etc. data.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 2.10)

**Imports** dplyr, rlang, methods, lifecycle

**Suggests** testthat (>= 2.1.0), ggplot2, stringr, knitr, rmarkdown, covr, pkgdown, DT

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**URL** https://github.com/andybega/states, https://www.andybeger.com/states/

**BugReports** https://github.com/andybega/states/issues

**NeedsCompilation** no

**Author** Andreas Beger [cre, aut] (<https://orcid.org/0000-0003-1883-3169>)

**Repository** CRAN

**Date/Publication** 2020-12-11 10:00:02 UTC

# R topics documented:



---

`.warner` *Temporary helper function to warn about argument order change*

---

## Description

Temporary helper function to warn about argument order change

## Usage

```
.warner(data, x)
```

## Arguments

| | |
|---|---|
| `data` | data.frame |
| `x` | string |

---

`compare` *Compare two statelists*

---

## Description

Check set overlap between two state lists / data frames, e.g. prior to merging them.

## Usage

```
compare(
  df1,
  df2,
  state1 = "gwcode",
  time1 = "year",
  state2 = "gwcode",
  time2 = "year"
)

report(x)
```

## Arguments

| | |
|---|---|
| df1 | data frame |
| df2 | data frame |
| state1 | (character(1)) Name of the country ID var in df1, default "gwcode" |
| time1 | (character(1)) Name of the time ID var in df1, default "year" |
| state2 | (character(1)) Name of the country ID var in df2, default "gwcode" |
| time2 | (character(1)) Name of the time ID var in df2, default "year" |
| x | a "state_sets" object produced by compare() |

## Details

This is a helper for interactively debugging data merges for data that may have slightly different state lists. For example, these differences in case sets could be because of country code differences.

## Examples

```
# df2 has all countries in 2018 but some values in x1 are missing
df1 <- state_panel(2018, 2018, partial = "any")
df1$x1 <- round(runif(nrow(df1))*5)
df1$x1[sample.int(nrow(df1), size = 20, replace = FALSE)] <- NA

# df2 is missing some countries and also has missing values in x2
df2 <- state_panel(2018, 2018, partial = "any")
df2 <- df2[sample.int(nrow(df2), size = 150), ]
df2$x2 <- round(runif(nrow(df2))*5)
df2$x2[sample.int(nrow(df2), size = 20, replace = FALSE)] <- NA

comp <- compare(df1, df2)
comp

report(comp)
```

## country_names *Country names*

### Description

Country names

### Usage

```
country_names(x, list = "GW", shorten = FALSE)
```

### Arguments

| | |
|---|---|
| x | (numeric())<br>A vector of numeric country codes |
| list | (logical(1))<br>Which states list to use? Only "GW" at this time. |
| shorten | (logical(1))<br>Shorten some of the longer country names like "Macedonia, the former Yugoslav Republic of"? |

### Examples

```
data("gwstates")
codes <- gwstates$gwcode
cn    <- country_names(codes)
cs    <- country_names(codes, shorten = TRUE)
data.frame(gwcode = codes, country_name = cn, short_names = cs)
```

## cowstates *Correlates of War list of independent states*

### Description

A list of independent states and microstates from 1816 on by the Correlates of War project.

### Usage

```
cowstates
```

## Format

Data frame

`ccode` Gleditsch and Ward country code.

`cowc` ISO 3 character country code.

`country_name` Long form country name

`start` Country start of independence.

`end` Country end of independence.

`microstate` Logical flag for whether state is a microstates with less than 250,000 population.

## Source

Correlates of War Project. 2011. "State System Membership List, v2011." Online, https://correlatesofwar.org

## Examples

```
data(cowstates)
head(cowstates)
```

---

# gwstates *Gleditsch and Ward list of independent states*

---

## Description

A list of independent states and microstates from 1816 on by Gleditsch and Ward

## Usage

```
gwstates
```

## Format

Data frame

`gwcode` Gleditsch and Ward country code.

`gwc` G&W character country code. This is derived from the COW character codes.

`country_name` Long form country name

`start` Country start of independence.

`end` Country end of independence.

`microstate` Logical flag for whether state is a microstates with less than 250,000 population.

## Source

http://ksgleditsch.com/data-4.html

## References

Gleditsch, Kristian S. and Michael D. Ward. 1999. "Interstate System Membership: A Revised List of the Independent States since 1816." *International Interactions* 25.

## Examples

```
data(gwstates)
head(gwstates)
```

---

`id_date_sequence` *Identify date sequences*

---

## Description

For correctly plotting country-time period spells

## Usage

```
id_date_sequence(x, pd = NULL)
```

## Arguments

| | |
|---|---|
| x | a Date sequence |
| pd | what is the time aggregation period in the data? |

## Examples

```
library("ggplot2")
d1 <- as.Date("2018-01-01")
d2 <- as.Date("2025-01-01")
seq1 <- seq(d1, d2, by = "year")
data.frame(seq1, id=id_date_sequence(seq1, "year"))
# With a gap, should be two ids
df <- data.frame(date = seq1[-4], id=id_date_sequence(seq1[-4], "year"), cowcode = 999)
df

# The point is to plot countries with interrupted independence correctly:
df$y <- c(rep(1, 3), rep(2, 4))
df$id <- paste0(df$cowcode, df$id)
df
ggplot(df, aes(x = date, y = y, group = cowcode)) + geom_line()
ggplot(df, aes(x = date, y = y, group = id)) + geom_line()

# Shortcut for integer years:
yr <- c(2002:2005, 2007:2010)
data.frame(year = yr, id = id_date_sequence(yr))
```

## parse_date *Parse date*

### Description

Try to parse input as a date

### Usage

```
parse_date(x)
```

### Arguments

| | |
|---|---|
| x | a integer or character vector, see examples |

### Examples

```
parse_date(2006)
parse_date("2006")
parse_date("2006-06")
parse_date("2006-06-01")
parse_date(as.Date("2006-06-01"))
```

## plot_missing *Visualize missing and non-proper cases for state panel data*

### Description

Plot missing values by country and date, and additionally identify country-date cases that do or do not match an independent state list.

### Usage

```
plot_missing(
  data,
  x = NULL,
  ccode = NULL,
  time = NULL,
  period = NULL,
  statelist = NULL,
  partial = "any",
  skip_labels = 5,
  space = deprecated()
)
```

```
missing_info(
  data,
  x = NULL,
  ccode = NULL,
  time = NULL,
  period = NULL,
  statelist = NULL,
  partial = NULL,
  space = deprecated()
)
```

## Arguments

| | |
|---|---|
| data | State panel data frame |
| x | Variable names(s), e.g. "x" or c("x1", "x2"). Default is NULL, in which case all columns expect the ccode and time ID columns will be used. |
| ccode | Name of variable identifying state country codes. If NULL (default) and one of "gwcode" or "cowcode" is a column in the data, it will be used. |
| time | Name of time identifier. If NULL and a "date" or "year" column are in the data, they will be used ("year", preferentially, if both are present) |
| period | Time period in which the data are. NULL by default and inferred to be "year" if the "time" column has name "year" or contains integers with a range between 1799 and 2050. Required if the "time" column is a `base::Date()` vector to avoid ambiguity. |
| statelist | Check not only missing values, but presence or absence of observations against a list of independent states? One of "GW", "COW" or "none". NULL by default, in which case it will be inferred if the ccode columns have the name "gwcode" or "cowcode", and "none" otherwise. |
| partial | Option for how to handle edge cases where a state is independent for only part of a time period (year, month, etc.). Options include "exact", and "any". See `state_panel()` for details. If NULL (default) and the "time" column is a date, it will be set to "exact", for yearly "time" columns it will be set to "any". |
| skip_labels | Only plot the label for every n-th country on the y-axis to avoid overplotting. |
| space | Deprecated, use "ccode" argument instead. |

## Details

`missing_info` provides the information that is plotted with `plot_missing`. The latter returns a ggplot, and thus can be chained with other ggplot functions as usual.

## Value

`plot_missing` returns a ggplot2 object.

`missing_info` returns a data frame with components:

| | |
|---|---|
| ccode | ccode identifier, with name equal to the "ccode" argument, e.g. "ccode". |

| | |
|---|---|
| time | Time identifier, with name equal to the "time" argument, e.g. "date". |
| independent | A logical vector, is the statelist argument is none, NA. |
| missing_value | A logical vector indicating if that record has missing values |
| status | The label used for plotting, combining the independence and missing value information for a case as appropriate. |

## Examples

```
# Create an example data frame with missing values
cy <- state_panel(as.Date("1980-06-30"), as.Date("2015-06-30"), by = "year",
useGW = TRUE)
cy$myvar <- rnorm(nrow(cy))
set.seed(1234)
cy$myvar[sample(1:nrow(cy), nrow(cy)*.1, replace = FALSE)] <- NA
str(cy)

# Visualize missing values:
plot_missing(cy, statelist = "none")

# missing_info() generates the data underlying plot_missing():
head(missing_info(cy, statelist =  "none"))

# if we specify a statelist to check against, 'independent' will have values
# now:
head(missing_info(cy, statelist = "GW"))

# Check data also against G&W list of independent states
head(missing_info(cy, statelist = "GW"))
plot_missing(cy, statelist = "GW")

# Live example with Polity data
data("polity")
head(polity)
plot_missing(polity, x = "polity", ccode = "ccode", time = "year",
             statelist = "COW")
# COW starts in 1816; Polity has excess data for several non-independent
# states after that date, and is missing coverage for several countries.

# The date option is relevant for years in which states gain or lose
# independence, so this will be slighlty different:
polity$date <- as.Date(paste0(polity$year, "-01-01"))
polity$year <- NULL
plot_missing(polity, x = "polity", ccode = "ccode", time = "date",
             period = "year", statelist = "COW")

# plot_missing returns a ggplot2 object, so you can do anything you want
polity$year <- as.integer(substr(polity$date, 1, 4))
polity$date <- NULL
plot_missing(polity, ccode = "ccode", statelist = "COW") +
  ggplot2::coord_flip()
```

## polity *Polity IV combined Polity scores*

### Description

Polity scores reflect how democratic or autocratic countries are from a scale of -10 (autocratic) to 10 (democratic). There are also three special codes for foreign "interruption" (-66), anarchy (-77), and transition periods (-88).

The data are included here for as an example for use with the missing plot. Thus they do not contain all available Polity indicators, which are available at the Polity project website www.systemicpeace.org.

### Usage

```
polity
```

### Format

Data frame

`ccode` Correlates of War (COW) country code.

`year` Year of the observation.

`polity` Combined Polity score.

### Source

Marshall, Monty G., Ted Robert Gurr, and Keith Jaggers. 2017. "Polity IV Project: Dataset Users' Manual." http://www.systemicpeace.org/inscr/p4manualv2016.pdf

### Examples

```
data("polity")
head("polity")
```

## prettyc *Shorten country names*

### Description

Shorten country names

### Usage

```
prettyc(x)
```

### Arguments

| | |
|---|---|
| x | (character())<br>Country names, e.g. from country_names(). |

### Examples

```
cn <- c(
  "Macedonia, the former Yugoslav Republic of",
  "Congo, the Democratic Republic of the",
  "Tanzania, United Republic of")
prettyc(cn)
```

---

## sfind *Lookup country codes or names*

---

### Description

Helper to look up state list entries by country code or name

### Usage

```
sfind(x, list = "both")
```

### Arguments

| | |
|---|---|
| x | The search string or number. |
| list | Which state list to search (both, GW, or COW only) |

### Examples

```
# Works with either integer or strings
sfind(325)
sfind("ALG")
sfind("Algeria")

# Search strings are treated as regular expressions (see stringr::str_detect)
sfind("Germany")
sfind("German")
```

## states *State system membership*

### Description

Create data based on the Gleditsch & Ward (G&W) or Correlates of War (COW) state system memberships lists. This is useful as a template for merging other sources of data that have conflicting sets of states.

### Details

See static docs at https://andybeger.com/states and the source code at https://www.github.com/andybega/states

### References

Gleditsch, Kristian S. & Michael D. Ward. 1999. ``Interstate System Membership: A Revised List of the Independent States since 1816.'' International Interactions 25: 393-413.

Correlates of War Project. 2017. ``State System Membership List, v2016.'' Online, http://correlatesofwar.org

## state_panel *Create state panel data*

### Description

Create panel data consisting of independent states in the international system.

### Usage

```
state_panel(start, end, by = NULL, partial = "any", useGW = TRUE)
```

### Arguments

| | |
|---|---|
| start | Beginning date for data, see parse_date() for format options. |
| end | End date for data, see parse_date() for format options. |
| by | Temporal resolution, "year", "month", or "day". If NULL, inferred from start and end input format, e.g. start = 2006`` implies by = "year"`. |
| partial | Option for how to handle edge cases where a state is independent for only part of a time period (year, month, etc.). Options include "exact", "first", "last", and "any". See details. |
| useGW | Use Gleditsch & Ward statelist or Correlates of War state system membership list. |

## Details

The partial option determines how to handle instances where a country gains or loses independence during a time period specified in the by option:

- "exact": the exact date in start is used for filtering
- "any": a state-period is included if the state was independent at any point in that period.
- "first": same as "exact" with the first date in a time period, e.g. "2006-01-01".
- "last": last date in a period. For "yearly" data, this is the same as "exact" with a start date like "YYYY-12-21", but for calendar months the last date varies, hence the need for this option.

## Value

A base::data.frame() with 2 columns for the country code and date information. The column names and types differ slightly based on the "useGW" and "by" arguments.

- The first column will be "gwcode" if useGW = TRUE (the default), and "cowcode" otherwise.
- The second column is an integer vector with name "year" for country-year data (if by or the inferred by value is "year"), and a base::Date() vector with the name "date" otherwise.

## Examples

```
# Basic usage with full option set specified:
gwlist  <- state_panel("1991-01-01", "2015-01-01", by = "year",
                       partial = "any", useGW = TRUE)
head(gwlist, 3)
cowlist <- state_panel("1991-01-01", "2015-01-01", by = "year",
                       partial = "any", useGW = FALSE)
head(cowlist, 3)

# For yearly data, a proper date is not needed, and by = "year" and
# partial = "any" are inferred.
gwlist <- state_panel(1990, 1995)
sfind(265, list = "GW")
265 %in% gwlist$gwcode

# Partials
# Focus on South Sudan--is there a record for 2011, first year of indendence?
data(gwstates)
dplyr::filter(gwstates, gwcode==626)

# No 2011 because SSD was not indpendent on January 1st 2011
x <- state_panel(2011, 2013, partial = "first")
dplyr::filter(x, gwcode==626)

# Includes 2011 because 12-31 date is used for filtering
x <- state_panel("2011-12-31", "2013-12-31", by = "year", partial = "exact")
dplyr::filter(x, gwcode==626)

# Includes 2011 because partial = "any"
x <- state_panel("2011-01-01", "2013-01-01", by = "year", partial = "any")
```

```
    dplyr::filter(x, gwcode==626)
```

# Index