Package: contentiousR (via r-universe)

October 15, 2026

Type Package

Title Contentious Politics and Civil Conflict Research

Version 0.1.0

Description Constructs reproducible country-year panels for research on contentious politics and civil conflict, as described in Guseinov (2026) <doi:10.2139/ssrn.7506799>. Functions build state panels keyed by Correlates of War or Gleditsch-Ward country codes and join curated socioeconomic, leadership, democracy, conflict, protest, and revolutionary-episode data on that common key, with diagnostics and plots for comparing source coverage. Bundled sources include Gapminder; the Nonviolent and Violent Campaigns and Outcomes (NAVCO) 2.1 data, Chenoweth and Shay (2019) <doi:10.7910/DVN/MHOXDV>; the Social Conflict Analysis Database, Salehyan et al. (2012) <doi:10.1080/03050629.2012.697426>; the Uppsala Conflict Data Program and Peace Research Institute Oslo (UCDP/PRIO) Armed Conflict Dataset, Gleditsch et al. (2002) <doi:10.1177/0022343302039005007>; and the Major Episodes of Contention data, Chenoweth and Kang (2026) <doi:10.1093/jopres/xjaf008>. Varieties of Democracy (V-Dem) indicators require the optional 'vdemdata' package, which is not on CRAN and can be installed from <https://github.com/vdeminstitute/vdemdata>.

License MIT + file LICENSE

Copyright See file COPYRIGHTS for the third-party data bundled in inst/extdata.

Encoding UTF-8

Depends R (>= 4.1.0)

Imports countrycode, dplyr, readr, states, stats, tidyr

Suggests brglm2, ggplot2, knitr, peacesciencer (>= 1.2.0), readxl, rmarkdown, testthat (>= 3.0.0), vdemdata

Config/testthat/edition 3

VignetteBuilder knitr

URL https://rguseinov.github.io/contentiousR/,
https://github.com/rguseinov/contentiousR

BugReports https://github.com/rguseinov/contentiousR/issues

Config/Needs/website r-lib/pkgdown

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ruslan Guseinov [aut, cre] (ORCID:
<https://orcid.org/0009-0000-1156-9495>)

Maintainer Ruslan Guseinov <rigusseinov@gmail.com>

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

Date/Publication 2026-10-15 16:20:09 UTC

RemoteUrl https://github.com/cran/contentiousR

RemoteRef HEAD

RemoteSha 32977158d0c743ffb9befbe469f59f68c6dc094d

Contents

add_conflict . . . 3
add_from_peacesciencer . . . 5
add_gdp . . . 6
add_lag . . . 7
add_leader_data . . . 7
add_military_expenditure . . . 8
add_population . . . 9
add_regions . . . 10
add_spell_duration . . . 11
add_vdem . . . 12
as_peacesciencer_panel . . . 13
build_states_panel . . . 14
cn_bib . . . 15
cn_cite . . . 15
conflict_data . . . 16
harmonize_conflict_data . . . 19
load_gdp_data . . . 21
load_leader_data . . . 23
load_military_expenditure_data . . . 24
load_population_data . . . 26
load_vdem_data . . . 28
plot_coverage . . . 29
plot_regional_coverage . . . 30
plot_temporal_coverage . . . 31

Index **33**

`add_conflict` *Add conflict or protest data to a state panel*

Description

Joins a conflict, protest, or revolutionary episode dataset onto an existing state panel at the country-year level.

Country-year datasets (`"scad"`, `"ucdp_prio"`, `"ucdp_vpp"`, `"mm"`, `"mmad"`) are already at the country-year level when loaded, so the `aggregate` argument has no effect for them.

Campaign and episode datasets (`"navco1.3"`, `"navco2.1"`, `"beissinger"`, `"csra"`, `"mec"`) can have multiple rows per country-year. With `aggregate = TRUE` (the default) these are collapsed to country-year by taking the maximum of all numeric columns, and an `n_campaigns` count column is added. Set `aggregate = FALSE` to keep campaign-level rows (may produce duplicates).

Usage

```
add_conflict(panel, dataset, aggregate = TRUE)
```

Arguments

`panel`	A data frame produced by `build_states_panel()`.
`dataset`	Dataset to load. One of: `"navco1.3"` NAVCO 1.3 campaign onsets (1900-2019). `"navco2.1"` NAVCO 2.1 campaign-years (1945-2013). `"beissinger"` Beissinger revolutionary episode onsets (1900-2014). `"csra"` HSE CSRA revolutionary episodes (2000-2024). `"scad"` SCAD 2018 social conflict events, Africa and Latin America (recorded starts 1989-2017). Prefix: `scad_`. `"ucdp_prio"` UCDP/PRIO Armed Conflict Dataset v26.1 (1946-2025). Prefix: `ucdp_prio_`. `"ucdp_vpp"` UCDP Violent Political Protest v26.1 (1989-2025). Prefix: `ucdp_vpp_`. Requires the `readxl` package. `"mm"` Mass Mobilization Project v4 (1990-2020). Prefix: `mm_`. `"mmad"` Mass Mobilization in Autocracies Database (2003-2022). Prefix: `mmad_`. `"mec"` Major Episodes of Contention (1955-2018), with global episode-level coverage. Prefix: `mec_`.
`aggregate`	Logical. If `TRUE` (default), aggregates to country-year before joining (relevant for campaign and episode datasets only). If `FALSE`, performs a raw left join.

Value

The input panel with conflict/protest columns added via left join. Unmatched country-years remain `NA`; the function does not assume that an unobserved event indicator is zero.

References

Chenoweth, E., & Shay, C. W. (2020). *List of Campaigns in NAVCO 1.3*. Harvard Dataverse. doi:10.7910/DVN/ON9XND

Chenoweth, E., & Shay, C. W. (2019). *NAVCO 2.1 Dataset*. Harvard Dataverse. doi:10.7910/DVN/MHOXDV

Beissinger, M. (2022). *Revolutionary Episodes Dataset*.

Ustyuzhanin, V., Semichev, D., Grinin, L., & Korotaev, A. (2026). *Revolutionary Database (2000-2024)*, Version 1.102. HSE University, Centre for Stability and Risk Analysis. https://social.hse.ru/mr/rev_bd

Salehyan, I., Hendrix, C. S., Hamner, J., Case, C., Linebarger, C., Stull, E., & Williams, J. (2012). Social conflict in Africa: A new database. *International Interactions*, 38(4), 503-511. doi:10.1080/03050629.2012.697426

Gleditsch, N. P., Wallensteen, P., Eriksson, M., Sollenberg, M., & Strand, H. (2002). Armed conflict 1946-2001: A new dataset. *Journal of Peace Research*, 39(5), 615-637. doi:10.1177/0022343302039005007

Svensson, I., Schaftenaar, S., & Allansson, M. (2022). Violent political protest: Introducing a new Uppsala Conflict Data Program data set on organized violence, 1989-2019. *Journal of Conflict Resolution*, 66(9), 1703-1730.

Clark, D. H., & Regan, P. M. (2016). *Mass Mobilization Protest Data*. Harvard Dataverse. doi:10.7910/DVN/HTTWYL

Weidmann, N. B., & Rød, E. G. (2019). *The Internet and Political Protest in Autocracies*. Oxford University Press.

Chenoweth, E., & Kang, S. (2026). The Major Episodes of Contention (MEC) Data Project: An introduction. *Journal of Peace Research*, 63(4), 724-734. doi:10.1093/jopres/xjaf008

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_conflict(dataset = "navco2.1")

panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_conflict(dataset = "ucdp_prio")

panel <- build_states_panel(2005, 2020, coding_system = "cow") |>
  add_conflict(dataset = "mmad")

# Raw join for legacy datasets — researcher handles aggregation manually
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_conflict(dataset = "navco1.3", aggregate = FALSE)

# MEC is episode-level; keep individual episodes with aggregate = FALSE
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_conflict(dataset = "mec", aggregate = FALSE)
```

add_from_peacesciencer

Add data from peacesciencer and restore the contentiousR schema

Description

Temporarily adapts a contentiousR state-year panel to the column names and metadata expected by a `peacesciencer` `add_*()` function. After that function returns, the temporary `ccode` or `gwcode` alias and dispatch attributes are removed. The original `cow` or `gw` key is retained.

Usage

```
add_from_peacesciencer(panel, .fun, ...)
```

Arguments

`panel`	A contentiousR state-year data frame containing exactly one of `cow` or `gw` and a `year` column.
`.fun`	A `peacesciencer` function whose first argument is a state-year data frame, such as `peacesciencer::add_archigos`.
`...`	Additional arguments passed to `.fun`.

Details

This helper is intended for `peacesciencer` functions that support state-year data. It does not alter country codes: `ccode` is simply `peacesciencer`'s name for the COW code stored as `cow` by contentiousR.

Use `as_peacesciencer_panel()` instead when several `peacesciencer` functions will be applied in one uninterrupted pipeline and retaining the compatibility metadata until the end is preferable.

Value

The data frame returned by `.fun`, restored to the input contentiousR key schema. A pre-existing `ccode` or `gwcode` column and pre-existing `ps_system` or `ps_data_type` attributes are preserved.

Examples

```
if (requireNamespace("peacesciencer", quietly = TRUE)) {
  library(peacesciencer)

  panel <- build_states_panel(1990, 1995, coding_system = "cow") |>
    add_from_peacesciencer(add_archigos)
}
```

add_gdp *Add GDP data to a state panel*

Description

Joins GDP data onto an existing state panel. The coding system ("cow" or "gw") and year range are detected automatically from the panel.

Usage

```
add_gdp(panel, dataset = c("gapminder", "fariss"))
```

Arguments

panel	A data frame produced by build_states_panel(), containing a cow or gw column and a year column.
dataset	Dataset to use: "gapminder" (default) or "fariss". See load_gdp_data() for details and added columns.

Value

The input panel with GDP columns added via left join.

References

Gapminder. (2024). *GDP per capita in constant PPP dollars*. https://www.gapminder.org/gdp-per-capita/

Fariss, C. J., Anders, T., Markowitz, J. N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591. doi:10.1177/00220027211054432

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_gdp()

panel <- build_states_panel(1900, 2010, coding_system = "cow") |>
  add_gdp(dataset = "fariss")
```

add_lag — *Add lagged variables to a state panel*

Description

Adds a lagged version of each requested column, computed within each state (`group_by(cow)` or `group_by(gw)`, matching the panel's coding system) after sorting by year — required for lags to be meaningful in panel data. Unlike a plain `dplyr::lag()`, the lag is `NA` wherever the preceding row is not exactly `n` years earlier, so gaps in the year sequence (e.g. after filtering with `tidyr::drop_na()`) do not silently produce a lag from the wrong year.

Usage

```
add_lag(panel, vars, n = 1)
```

Arguments

`panel`	A data frame produced by `build_states_panel()`, containing a `cow` or `gw` column and a `year` column.
`vars`	Character vector of column names in `panel` to lag.
`n`	Integer. Number of years to lag by. Default `1`.

Value

The input panel with one additional column per entry in `vars`, named `<var>_l`, holding the value of `<var>` from `n` years earlier for the same state (or `NA` if that year is not present in `panel`).

Examples

```
panel <- build_states_panel(1990, 2000, coding_system = "cow") |>
  add_gdp() |>
  add_lag(vars = c("gdp_pcap", "gdp_growth"), n = 1)
```

add_leader_data — *Add leader data to a state panel*

Description

Joins country-year leader data onto an existing state panel. The coding system and year range are detected automatically from the panel.

Each country-year is assigned the leader who held power at the end of that year (i.e., the leader with the latest start date when multiple leaders served in the same year).

Usage

```
add_leader_data(panel, dataset = c("archigos", "reign"))
```

Arguments

panel	A data frame produced by `build_states_panel()`.
dataset	Dataset to use: `"archigos"` (default) or `"reign"`. See `load_leader_data()` for details.

Value

The input panel with leader columns added via left join.

References

Goemans, H. E., Gleditsch, K. S., & Chiozza, G. (2009). Introducing Archigos: A dataset of political leaders. *Journal of Peace Research*, 46(2), 269-283.

Bell, C., Besaw, C., & Frank, M. (2021). *The Rulers, Elections, and Irregular Governance (REIGN) Dataset*.

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_leader_data(dataset = "archigos")

panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_leader_data(dataset = "reign")
```

---

add_military_expenditure

Add military expenditure data to a state panel

Description

Joins country-year military expenditure data onto an existing state panel. The coding system and year range are detected automatically from the panel.

Usage

```
add_military_expenditure(panel, dataset = c("sipri", "nmc", "barnum"))
```

Arguments

panel	A data frame produced by `build_states_panel()`.
dataset	Dataset to use: `"sipri"` (default), `"nmc"`, or `"barnum"`. See `load_military_expenditure_data()` for details and added columns.

Value

The input panel with military expenditure columns added via left join.

References

Singer, J. D., Bremer, S., & Stuckey, J. (1972). Capability distribution, uncertainty, and major power war, 1820-1965. In B. Russett (Ed.), *Peace, War, and Numbers* (pp. 19-48). Sage.

Singer, J. D. (1988). Reconstructing the Correlates of War dataset on material capabilities of states, 1816-1985. *International Interactions*, 14, 115-132.

Stockholm International Peace Research Institute. (2025). *SIPRI Military Expenditure Database*. doi:10.55163/CQGC9685

Barnum, M., Fariss, C. J., Markowitz, J. N., & Morales, G. (2025). Measuring arms: Introducing the Global Military Spending Dataset. *Journal of Conflict Resolution*, 69(2-3), 540-567. doi:10.1177/00220027241232964

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_military_expenditure()

panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_military_expenditure(dataset = "nmc")
```

---

add_population *Add population data to a state panel*

Description

Joins country-year population data onto an existing state panel. The coding system and year range are detected automatically from the panel.

Usage

```
add_population(panel, dataset = c("wpp", "nmc", "fariss"))
```

Arguments

panel	A data frame produced by `build_states_panel()`.
dataset	Dataset to use: `"wpp"` (default), `"nmc"`, or `"fariss"`. See `load_population_data()` for details and added columns.

Value

The input panel with population columns added via left join.

References

United Nations, Department of Economic and Social Affairs, Population Division. (2024). *World Population Prospects 2024*. https://population.un.org/wpp/

Fariss, C. J., Anders, T., Markowitz, J. N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591. doi:10.1177/00220027211054432

Singer, J. D., Bremer, S., & Stuckey, J. (1972). Capability distribution, uncertainty, and major power war, 1820-1965. In B. Russett (Ed.), *Peace, War, and Numbers* (pp. 19-48). Sage.

Singer, J. D. (1988). Reconstructing the Correlates of War dataset on material capabilities of states, 1816-1985. *International Interactions*, 14, 115-132.

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_population(dataset = "wpp")

panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_population(dataset = "nmc")
```

---

add_regions — *Add region names to a state panel*

Description

Joins region names onto an existing state panel via `countrycode::countrycode()`, mapping the panel's `cow`/`gw` codes to one or more of `countrycode`'s regional classifications. The coding system is detected automatically from the panel.

Usage

```
add_regions(panel, region = "region")
```

Arguments

`panel`	A data frame produced by `build_states_panel()`, containing a `cow` or `gw` column.
`region`	Character vector selecting one or more `countrycode` destination classifications to add, from: • `"region"` (default): World Bank 7-region classification. • `"region23"`: World Bank's finer, 23-region classification. • `"un.region.name"`: UN macro-region (continent-level). • `"un.regionsub.name"`: UN sub-region. See `countrycode::countrycode()` for details on these classifications.

Value

The input panel with one additional column per entry in `region`, named after the classification (e.g. `region`, `region23`), holding the corresponding region name for each row (`NA` where `countrycode()` finds no match).

Examples

```
panel <- build_states_panel(1990, 2000, coding_system = "cow") |>
  add_regions()

panel <- build_states_panel(1990, 2000, coding_system = "cow") |>
  add_regions(region = c("region", "un.regionsub.name"))
```

---

add_spell_duration *Add conflict spell (peace-years) durations to a state panel*

Description

Computes a BTSCS-style spell/duration counter from a binary conflict column already present in the panel (typically added by `add_conflict()`): the number of years since the last event for that state, restarting after each new event. This is the same construct as `peacesciencer::add_spells()`, generalized to work with any binary column rather than a fixed set of bundled column names. Named `add_spell_duration()`, not `add_spells()`, to avoid colliding with `peacesciencer::add_spells()` when both packages are attached.

Usage

```
add_spell_duration(panel, event = NULL, ongoing = NULL)
```

Arguments

`panel`	A data frame produced by `build_states_panel()`, optionally already joined with conflict data via `add_conflict()`.
`event`	Character. Name of the binary (0/1) column to compute spells from. If `NULL` (the default), `add_spell_duration()` looks for exactly one column in `panel` whose name ends in `_onset`, `_incidence`, or `_ongoing`; if zero or more than one match, it errors and asks for an explicit `event`.
`ongoing`	Logical. If `TRUE`, `event` is treated as an "ongoing" series where a run of consecutive `1`s is one continuous event (only the first year of the run counts as a new failure; the remaining years of that run get `NA` in the output, matching `peacesciencer::add_spells()`'s convention of excluding within-event years from a spell that measures time since the last event). If `FALSE`, every `1` in `event` is treated as its own event, appropriate for an already onset-coded column. If `NULL` (the default), this is inferred from the column name: `TRUE` for `_incidence`/`_ongoing` suffixes, `FALSE` for `_onset`.

Details

event must not contain NA. add_conflict()'s output frequently has NA for country-years unmatched by the source (documented behavior: absence from a source is not evidence of zero events), and a spell count cannot be computed across a gap of unknown conflict status. Filter or otherwise resolve these NAs before calling add_spell_duration().

Value

The input panel with one additional integer column, named after event with its onset/incidence/ongoing suffix (if any) replaced by _spell. In a fresh spell, the first observed year is 0 and each subsequent year without a new event increments by one; the event year itself carries the final count of the spell it closes.

Examples

```
panel <- build_states_panel(1990, 2010, coding_system = "gw") |>
  add_conflict(dataset = "ucdp_prio") |>
  tidyr::drop_na(ucdp_prio_onset) |>
  add_spell_duration(event = "ucdp_prio_onset")
```

---

add_vdem *Add V-Dem indicators to a state panel*

Description

Joins V-Dem country-year data onto an existing state panel. The coding system and year range are detected automatically from the panel.

Requires the vdemdata package: remotes::install_github("vdeminstitute/vdemdata")

Usage

```
add_vdem(panel, vars = NULL)
```

Arguments

panel	A data frame produced by build_states_panel().
vars	Character vector of V-Dem variable names. If NULL, a default set of democracy, civil liberties, civil society, and rule-of-law indicators is used. See load_vdem_data() for the full default list.

Value

The input panel with V-Dem columns added via left join.

References

Coppedge, M., Gerring, J., Knutsen, C. H., Lindberg, S. I., Teorell, J., Altman, D., Bernhard, M., Fish, M. S., Glynn, A., Hicken, A., et al. (2025). *V-Dem [Country-Year/Country-Date] Dataset v15*. Varieties of Democracy.

Examples

```
if (requireNamespace("vdemdata", quietly = TRUE)) {
panel <- build_states_panel(1990, 2010, coding_system = "cow") |>
  add_vdem(vars = c("v2x_polyarchy", "v2x_libdem", "v2x_rule"))
}
```

---

as_peacesciencer_panel

Make a state panel compatible with peacesciencer

Description

Adds the key alias and metadata expected by state-year functions from the `peacesciencer` package. The original `cow` or `gw` column is retained, so the result remains compatible with all `contentiousR` functions.

Usage

```
as_peacesciencer_panel(panel)
```

Arguments

`panel`	A state-year data frame containing exactly one of `cow` or `gw` and a `year` column.

Details

`peacesciencer` dispatches its `add_*()` functions using attributes created by `peacesciencer::create_stateyears()`. It also expects its own key names: `ccode` for COW panels and `gwcode` for Gleditsch-Ward panels. This adapter supplies that interface without replacing contentiousR's `cow` or `gw` key.

Place the adapter immediately before the first `peacesciencer` function, or earlier in the pipeline. Dplyr joins used by contentiousR preserve these attributes.

Value

The input data frame with an additional `ccode` (COW) or `gwcode` (Gleditsch-Ward) alias and the attributes `ps_system` and `ps_data_type = "state_year"`. Rows and analytical variables are not changed.

Examples

```
panel <- build_states_panel(1990, 1995, coding_system = "cow") |>
  add_gdp() |>
  as_peacesciencer_panel()

names(panel)
attr(panel, "ps_system")
attr(panel, "ps_data_type")
```

---

build_states_panel *Build a state panel*

Description

Build a state panel

Usage

```
build_states_panel(
  start_year = 1946,
  end_year = 2013,
  coding_system = c("cow", "gw"),
  exclude_microstates = TRUE,
  exclude_non_un = TRUE,
  exclude_islands = FALSE
)
```

Arguments

`start_year`	First year. Must be `1816` or later.
`end_year`	Last year. Must be `2025` or earlier.
`coding_system`	Country coding system. Either `"cow"` or `"gw"`.
`exclude_microstates`	Logical. Exclude entities identified as microstates by `states::state_panel()`.
`exclude_non_un`	Logical. Exclude the package's fixed set of non-UN entities (Kosovo, Taiwan, Hong Kong, the European Union, and South Vietnam for COW; Kosovo, Taiwan, and South Vietnam for GW).
`exclude_islands`	Logical. Exclude six small Caribbean island states: Dominica, Grenada, Saint Lucia, Saint Vincent and the Grenadines, Antigua and Barbuda, and Saint Kitts and Nevis.

Value

A data frame with state-year observations.

cn_bib — *Bundled bibliography for cn_cite()*

Description

A `bibentry` object with one entry per bundled or interoperable data source, used by `cn_cite()`. Each entry's `keywords` field lists the `contentiousR` function(s) and/or `dataset =` value(s) it documents.

Usage

```
cn_bib
```

Format

A `bibentry` object of length 22.

Source

`data-raw/contentiousR_bib.bib` and `data-raw/contentiousR_bib.R`.

cn_cite — *Cite a bundled data source*

Description

Looks up the bundled bibliography (`cn_bib`) by keyword or BibTeX key and returns the matching entries. Printing the result (e.g. at the console) shows them as BibTeX text, ready to paste into your own `.bib` file.

Usage

```
cn_cite(x, column = c("keywords", "bibtexkey"))
```

Arguments

`x`	A search string. Matched against the `keywords` field by default (function names such as `"add_conflict()"` or `"load_gdp_data()"`, and `dataset =` values such as `"archigos"` or `"mec"`) as an exact match against one comma-separated keyword, or against BibTeX keys as a fixed substring match when `column = "bibtexkey"`. Neither is a regular expression.
`column`	Which field to search: `"keywords"` (default) or `"bibtexkey"`.

Value

A `bibentry` object (with the additional class `cn_citation`) holding the matching entries. Its `print()` method displays them as BibTeX; all other `bibentry` methods, such as `format()` and `toBibtex()`, work as usual.

Examples

```
cn_cite("archigos")
cn_cite("add_conflict()")
cn_cite("gleditschArmedConflict194620012002", column = "bibtexkey")
```

---

conflict_data *Load conflict and mobilization data*

Description

Loads one of ten conflict, protest, or revolutionary episode datasets bundled with the package. `scad`, `ucdp_prio`, `ucdp_vpp`, `mm`, and `mmad` are returned at the country-year level; their event- or conflict-level source rows are aggregated inside this function.

`navco1.3`, `beissinger`, `csra`, and `mec` return campaign or episode records; `navco2.1` returns campaign-year rows. These sources can contain multiple rows for one country-year. Use `add_conflict(aggregate = TRUE)` (the default) to collapse them to country-year.

Usage

```
conflict_data(
  start_year = 1945,
  end_year = 2013,
  dataset = c("navco1.3", "navco2.1", "beissinger", "csra", "scad", "ucdp_prio",
    "ucdp_vpp", "mm", "mmad", "mec"),
  coding_system = c("cow", "gw")
)
```

Arguments

`start_year` First year to include.

`end_year` Last year to include.

`dataset` Dataset to load. One of:

- `"navco1.3"` NAVCO 1.3 campaign onsets. Coverage: 1900-2019.
- `"navco2.1"` NAVCO 2.1 campaign-years. Coverage: 1945-2013.
- `"beissinger"` Beissinger revolutionary episode onsets. Coverage: 1900-2014.
- `"csra"` HSE CSRA revolutionary episodes. Coverage: 2000-2024.
- `"scad"` SCAD 2018 social conflict events, Africa and Latin America. Recorded event starts: 1989-2017 (the 1989 event continues into the source's 1990 coverage period). Prefix: `scad_`.

"ucdp_prio" UCDP/PRIO Armed Conflict Dataset v26.1. Coverage: 1946-2025. Prefix: ucdp_prio_.

"ucdp_vpp" UCDP Violent Political Protest Dataset v26.1. Coverage: 1989-2025. Prefix: ucdp_vpp_. Requires the readxl package.

"mm" Mass Mobilization Project v4 (Clark and Regan). Coverage: 1990-2020. Prefix: mm_.

"mmad" Mass Mobilization in Autocracies Database. Coverage: 2003-2022. Prefix: mmad_.

"mec" Major Episodes of Contention. Global coverage: 1955-2018. Episode records; prefix: mec_.

coding_system Country coding system: "cow" or "gw". ucdp_prio and ucdp_vpp use GW codes natively; when coding_system = "cow" they are converted via countrycode and countries without a COW equivalent are dropped.

Details

All outputs use the requested cow or gw key and year. Dataset-specific columns have a stable source prefix. A missing country-year row is not created by this loader; after add_conflict(), an unmatched year is NA, not an assumed zero.

SCAD is deduplicated to one row per event before country-year aggregation. Its UCDP placeholder rows are excluded, and documented negative codes for unknown death counts are treated as missing.

For UCDP/PRIO, ucdp_prio_incidence identifies any active conflict-year and ucdp_prio_onset identifies a new episode using the source's start_date2. The VPP source does not include an episode-start variable; ucdp_vpp_incidence should be preferred. The legacy ucdp_vpp_onset column is retained as an incidence alias for compatibility.

The Mass Mobilization source mixes exact participant counts with text such as ranges and inequalities. mm_participants is the maximum among genuinely numeric values; mm_participants_reported preserves the source strings.

MEC is kept at its published episode level. Its source byear becomes year, ccode becomes cow, and all other source columns receive the mec_ prefix. mec_episode equals one for every returned episode record. Fourteen left-censored records have an actual mec_bdate before 1955 while their source byear is 1955. With GW coding, COW 679 (Yemen) maps to GW 678 and COW 817 (South Vietnam) maps to GW 817; Tonga has no GW state equivalent and is omitted.

Value

A data frame. navco1.3, beissinger, csra, and mec contain one row per campaign or episode record; navco2.1 contains campaign-year rows. The other datasets contain one row per country-year.

Source

NAVCO 2.1: doi:10.7910/DVN/MHOXDV.

Beissinger Revolutionary Episodes: https://web.archive.org/web/20250331220952/https://mbeissinger.scholar.princeton.edu/revolutionary-episodes-dataset.

HSE Revolutionary Database: https://social.hse.ru/mr/rev_bd.

SCAD: https://www.strausscenter.org/ccaps-research-areas/social-conflict/database/.

UCDP: https://ucdp.uu.se/downloads/replication_data.html.

MMAD: https://mmadatabase.org/.

MEC: Chenoweth and Kang (2026), doi:10.1093/jopres/xjaf008; data release doi:10.7910/DVN/JQWQNW.

References

Chenoweth, E., & Shay, C. W. (2020). *List of Campaigns in NAVCO 1.3*. Harvard Dataverse. doi:10.7910/DVN/ON9XND

Chenoweth, E., & Shay, C. W. (2019). *NAVCO 2.1 Dataset*. Harvard Dataverse. doi:10.7910/DVN/MHOXDV

Beissinger, M. (2022). *Revolutionary Episodes Dataset*.

Ustyuzhanin, V., Semichev, D., Grinin, L., & Korotaev, A. (2026). *Revolutionary Database (2000-2024)*, Version 1.102. HSE University, Centre for Stability and Risk Analysis. https://social.hse.ru/mr/rev_bd

Salehyan, I., Hendrix, C. S., Hamner, J., Case, C., Linebarger, C., Stull, E., & Williams, J. (2012). Social conflict in Africa: A new database. *International Interactions*, 38(4), 503-511. doi:10.1080/03050629.2012.697426

Gleditsch, N. P., Wallensteen, P., Eriksson, M., Sollenberg, M., & Strand, H. (2002). Armed conflict 1946-2001: A new dataset. *Journal of Peace Research*, 39(5), 615-637. doi:10.1177/0022343302039005007

Svensson, I., Schaftenaar, S., & Allansson, M. (2022). Violent political protest: Introducing a new Uppsala Conflict Data Program data set on organized violence, 1989-2019. *Journal of Conflict Resolution*, 66(9), 1703-1730.

Clark, D. H., & Regan, P. M. (2016). *Mass Mobilization Protest Data*. Harvard Dataverse. doi:10.7910/DVN/HTTWYL

Weidmann, N. B., & Rød, E. G. (2019). *The Internet and Political Protest in Autocracies*. Oxford University Press.

Chenoweth, E., & Kang, S. (2026). The Major Episodes of Contention (MEC) Data Project: An introduction. *Journal of Peace Research*, 63(4), 724-734. doi:10.1093/jopres/xjaf008

Examples

```
navco <- conflict_data(1990, 2010, dataset = "navco2.1", coding_system = "cow")

scad  <- conflict_data(1995, 2015, dataset = "scad",      coding_system = "cow")

ucdp  <- conflict_data(1990, 2020, dataset = "ucdp_prio", coding_system = "gw")

mec   <- conflict_data(1990, 2010, dataset = "mec",       coding_system = "cow")
```

harmonize_conflict_data

Harmonize externally supplied conflict/event data onto a state panel

Description

Joins a conflict or event dataset that the caller supplies directly – downloaded via GDELT, ICEWS, ACLED, or any other source not bundled with `contentiousR` – onto an existing state panel. Unlike `add_conflict()`, this function never downloads or bundles any data itself: `data` is whatever data frame the caller already has in R, in whatever raw column names and country/date encodings that source happens to use.

Usage

```
harmonize_conflict_data(
  panel,
  data,
  country_col,
  origin_code,
  prefix,
  year_col = NULL,
  date_col = NULL,
  date_format = NULL,
  aggregate = TRUE
)
```

Arguments

`panel`	A data frame produced by `build_states_panel()`, containing a `cow` or `gw` column and a `year` column.
`data`	A data frame of external conflict/event data, one row per event (the default; see `aggregate`).
`country_col`	Character. Name of the column in `data` holding country identifiers.
`origin_code`	Character. The `countrycode::countrycode()` origin code matching `country_col`'s format, e.g. `"country.name"`, `"iso3c"`, `"fips"` (GDELT's two-letter country codes), `"cown"`, or `"gwc"`. See `countrycode::codelist` for the full list of supported origins.
`prefix`	Character. Prefix for the new column(s), e.g. `"gdelt"` produces `gdelt_onset`/`gdelt_n_events` (see `aggregate`).
`year_col`	Character. Name of a column in `data` already holding an integer year. Exactly one of `year_col` or `date_col` must be supplied.
`date_col`	Character. Name of a column in `data` holding a date to extract the year from. Exactly one of `year_col` or `date_col` must be supplied.

date_format — Character. A format string for as.Date(), e.g. "%Y%m%d" for GDELT's SQLDATE field, or "%d %B %Y" for ACLED's event_date. If NULL (the default), date_col is parsed with as.Date() directly, which expects ISO 8601 ("%Y-%m-%d") or an already-Date-typed column. Ignored if year_col is supplied.

aggregate — Logical. If TRUE (default), data is treated as one row per event and is collapsed to one row per country-year, adding <prefix>_onset (always 1, since a row only exists where an event occurred) and <prefix>_n_events (event count that country-year). If FALSE, data is assumed to already be one row per country-year; every other column in data (besides country_col/year_col/ date_col) is renamed with <prefix>_ and joined as-is.

Details

Country matching uses countrycode::countrycode(), so any row whose country_col value it cannot map to the panel's coding system (an unrecognized name, a sub-national/organizational entry, etc.) is dropped, with a message reporting how many rows were dropped.

contentiousR deliberately does not depend on gdeltr2, icews, or acled.api – this function is a generic bridge instead, so the package is not exposed to any of those clients' own dependencies, rate limits, or licensing terms. Cite the original data source directly (this function does not know anything about the specific provenance of data).

Value

panel with the new column(s) joined in via left_join(). Country-years absent from data remain NA – as with add_conflict(), that is not evidence of zero events, only that data doesn't cover that country-year.

Examples

```
panel <- build_states_panel(2015, 2018, coding_system = "cow")

# An ACLED-style extract: one row per event, country as free text.
acled_like <- data.frame(
  country    = c("Nigeria", "Nigeria", "Mali", "Mali", "Mali"),
  event_date = c("2016-03-01", "2017-11-20", "2015-06-14",
                 "2015-09-02", "2018-01-30"),
  stringsAsFactors = FALSE
)

panel |>
  harmonize_conflict_data(
    data        = acled_like,
    country_col = "country",
    origin_code = "country.name",
    date_col    = "event_date",
    prefix      = "acled"
  )

# A GDELT-style extract: FIPS country codes, SQLDATE as YYYYMMDD.
gdelt_like <- data.frame(
```

```
  Actor1CountryCode = c("NI", "NI", "ML"),
  SQLDATE           = c(20160301L, 20171120L, 20150614L)
)

panel |>
  harmonize_conflict_data(
    data        = gdelt_like,
    country_col = "Actor1CountryCode",
    origin_code = "fips",
    date_col    = "SQLDATE",
    date_format = "%Y%m%d",
    prefix      = "gdelt"
  )
```

`load_gdp_data` *Load GDP data*

Description

Loads GDP data from one of two sources included with the package and returns a country-year dataset using either COW or Gleditsch-Ward country codes.

Usage

```
load_gdp_data(
  start_year = 1945,
  end_year = 2019,
  dataset = c("gapminder", "fariss"),
  coding_system = c("cow", "gw")
)
```

Arguments

`start_year`	Integer. First year to include.
`end_year`	Integer. Last year to include.
`dataset`	Character. Dataset to load: `"gapminder"` Gapminder GDP per capita (v32). Coverage: 1945-2019 by default (full source coverage is wider). `"fariss"` Latent GDP and GDP per capita estimates (Fariss, Anders, Markowitz & Barnum 2022), combined from their `fariss_gdp` and `fariss_gdppc` replication files. Coverage: 1500-2019.
`coding_system`	Character. Country coding system to use. Either `"cow"` for Correlates of War codes or `"gw"` for Gleditsch-Ward codes.

Details

gdp_growth (gapminder only) is NA unless the preceding row is genuinely one year earlier for the same country, the same gap-safety check add_lag() uses, so a missing year never silently produces a growth rate computed against the wrong base year.

fariss_gdp and fariss_gdppc are the model's own latent-scale estimates, taken directly from the "latent_gdp"/"latent_gdppc" rows of the replication files (as opposed to the underlying raw indicator series also present in those files). Their absolute units are not independently verified here; consult Fariss et al. (2022) before using them outside of relative/comparative analysis.

Value

A data frame with country-year GDP indicators. If coding_system = "cow", the data frame contains a cow column. If coding_system = "gw", it contains a gw column. Columns differ by dataset:

gapminder: year, gdp_pcap, log_gdp_pcap, gdp_growth.

fariss: year, fariss_gdp, fariss_gdppc.

Source

Gapminder: GDP per capita data from Gapminder. Gapminder data are distributed under CC BY 4.0; please cite Gapminder and the original data providers.

Fariss: Fariss, C.J., Anders, T., Markowitz, J.N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591.

References

Gapminder. (2024). *GDP per capita in constant PPP dollars*. https://www.gapminder.org/gdp-per-capita/

Fariss, C. J., Anders, T., Markowitz, J. N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591. doi:10.1177/00220027211054432

Examples

```
gdp_cow <- load_gdp_data(coding_system = "cow")

gdp_gw <- load_gdp_data(coding_system = "gw")

fariss_gdp <- load_gdp_data(1900, 2015, dataset = "fariss", coding_system = "cow")
```

load_leader_data *Load leader data*

Description

Loads country-year leader data from one of two datasets included with the package. Each row in the output represents one country-year, with the leader who held power at the end of the year. When multiple leaders served in the same year (i.e., a leadership transition occurred), the leader with the latest start date is retained.

Usage

```
load_leader_data(
  start_year = 1945,
  end_year = 2015,
  dataset = c("archigos", "reign"),
  coding_system = c("cow", "gw")
)
```

Arguments

`start_year`	Integer. First year to include.
`end_year`	Integer. Last year to include.
`dataset`	Character. Dataset to load: `"archigos"` Archigos 4.1 (Goemans, Gleditsch & Chiozza). Leader spells with entry/exit mode coding. Coverage: 1875–2015. `"reign"` REIGN Leader List (Bell 2021). Leader spells with military background coding. Coverage: 1921–2021.
`coding_system`	Character. Country coding system: `"cow"` or `"gw"`.

Value

A country-year data frame. Columns differ by dataset:

archigos: `cow/gw`, `year`, `leader`, `entry`, `exit`, `irregular_entry`, `irregular_exit`, `female_leader`, `yrborn`, `posttenurefate`, `leader_tenure`.

reign: `cow/gw`, `year`, `leader`, `female_leader`, `military_bg`, `birthyear`, `leader_tenure`.

Source

Archigos: Goemans, H.E., Gleditsch, K.S., & Chiozza, G. (2009). Introducing Archigos: A Dataset of Political Leaders. *Journal of Peace Research*, 46(2), 269–283. https://web.archive.org/web/20241206095059/http://ksgleditsch.com/archigos.html

REIGN: Bell, C. (2021). REIGN: Rulers, Elections, and Irregular Governance Dataset. One Earth Future Foundation. https://oefdatascience.github.io/REIGN.github.io/

References

Goemans, H. E., Gleditsch, K. S., & Chiozza, G. (2009). Introducing Archigos: A dataset of political leaders. *Journal of Peace Research*, 46(2), 269-283.

Bell, C., Besaw, C., & Frank, M. (2021). *The Rulers, Elections, and Irregular Governance (REIGN) Dataset*.

Examples

```
arch <- load_leader_data(1990, 2010, dataset = "archigos", coding_system = "cow")

reign <- load_leader_data(1990, 2010, dataset = "reign", coding_system = "gw")
```

---

load_military_expenditure_data

Load military expenditure data

Description

Loads country-year military expenditure data from one of the sources included with the package.

Usage

```
load_military_expenditure_data(
  start_year = 1945,
  end_year = 2019,
  dataset = c("sipri", "nmc", "barnum"),
  coding_system = c("cow", "gw")
)
```

Arguments

`start_year` Integer. First year to include.

`end_year` Integer. Last year to include.

`dataset` Character. Dataset to load:

- `"sipri"` SIPRI Military Expenditure Database v1.2: spending in constant (2024) USD millions (`sipri_milex`) and military burden as a share of GDP (`sipri_milburden`, a fraction, not a percentage). Coverage: 1949-2025.
- `"nmc"` Correlates of War National Material Capabilities v7.0 (abridged): military expenditure in thousands of current USD (`nmc_milex`). Coverage: 1816-2022.
- `"barnum"` Global Military Spending Dataset (Barnum, Fariss, Markowitz & Morales 2025): military burden (`barnum_milburden`, a fraction of GDP) plus latent-model expenditure estimates on two unit bases (`barnum_sipri`, `barnum_nmc`; see Details). Coverage: 1816-2019.

`coding_system` Character. Country coding system: `"cow"` or `"gw"`.

Details

NMC's `-9` missing-data sentinel is recoded to `NA` in `nmc_milex`.

The SIPRI workbook lists a small number of rows (e.g. "Africa", "NATO") that are regional or organizational aggregates rather than countries; these have no matching COW/Gleditsch-Ward code and are dropped when `countrycode::countrycode()` returns `NA`.

Barnum et al.'s replication files contain one row per country-year for each of 24 underlying source indicators (SIPRI, WMEAT, NCD, IISS, and others), each already reported in its own original currency/unit. Unlike Fariss et al. (2022)'s GDP and population estimates, the paper does not publish a single combined "latent" value on a real monetary scale: "the average by itself does not have a real or direct monetary unit" (p. 550). Instead, `barnum_sipri` and `barnum_nmc` are the model's posterior estimate of two specific indicators (the `"milex_con_2022_sipri"` and `"milex_con_2017_nmc"` series respectively), which extends each one's coverage back to 1816 using information from all 24 sources, still expressed in that indicator's own original units. `barnum_milburden` is `milexgdp` from Barnum et al.'s separate military-burden estimates file, which is unit-free and so is published directly.

Both `"_con_"` indicators are in constant (inflation-adjusted) dollars, unlike `nmc_milex` from `dataset = "nmc"`, which NMC documents in current-year dollars. `barnum_nmc` is therefore not directly comparable to `nmc_milex`: their ratio varies smoothly over time (e.g. around 2.4 for the United States in 1990, falling toward 1 by the mid-2010s), which reflects cumulative inflation adjustment rather than disagreement between the two sources.

Value

A country-year data frame. Columns differ by dataset:

sipri: `cow/gw`, `year`, `sipri_milex`, `sipri_milburden`.

nmc: `cow/gw`, `year`, `nmc_milex`.

barnum: `cow/gw`, `year`, `barnum_milburden`, `barnum_sipri`, `barnum_nmc`.

Source

SIPRI: Stockholm International Peace Research Institute (2025). SIPRI Military Expenditure Database. https://www.sipri.org/databases/milex

NMC: Singer, J. David, Bremer, Stuart & Stuckey, John (1972). Capability Distribution, Uncertainty, and Major Power War, 1820-1965. National Material Capabilities v7.0. https://correlatesofwar.org/data-sets/national-material-capabilities/

Barnum: Barnum, M., Fariss, C.J., Markowitz, J.N., & Morales, G. (2025). Measuring Arms: Introducing the Global Military Spending Dataset. *Journal of Conflict Resolution*, 69(2-3), 540-567.

References

Singer, J. D., Bremer, S., & Stuckey, J. (1972). Capability distribution, uncertainty, and major power war, 1820-1965. In B. Russett (Ed.), *Peace, War, and Numbers* (pp. 19-48). Sage.

Singer, J. D. (1988). Reconstructing the Correlates of War dataset on material capabilities of states, 1816-1985. *International Interactions*, 14, 115-132.

Stockholm International Peace Research Institute. (2025). *SIPRI Military Expenditure Database*. doi:10.55163/CQGC9685

Barnum, M., Fariss, C. J., Markowitz, J. N., & Morales, G. (2025). Measuring arms: Introducing the Global Military Spending Dataset. *Journal of Conflict Resolution*, 69(2-3), 540-567. doi:10.1177/00220027241232964

Examples

```
milex_sipri <- load_military_expenditure_data(1990, 2015, dataset = "sipri", coding_system = "cow")

milex_nmc <- load_military_expenditure_data(1900, 2015, dataset = "nmc", coding_system = "cow")
```

---

load_population_data *Load population data*

Description

Loads country-year population data from one of three sources included with the package.

Usage

```
load_population_data(
  start_year = 1945,
  end_year = 2019,
  dataset = c("wpp", "nmc", "fariss"),
  coding_system = c("cow", "gw")
)
```

Arguments

`start_year`	Integer. First year to include.
`end_year`	Integer. Last year to include.
`dataset`	Character. Dataset to load: `"wpp"` UN World Population Prospects, total population in thousands. Coverage: 1949-2023. `"nmc"` Correlates of War National Material Capabilities v7.0 (abridged): total population (`nmc_tpop`) and urban population (`nmc_upop`), both in thousands. Coverage: 1816-2022. `"fariss"` Latent population estimates (Fariss, Anders, Markowitz & Barnum 2022). Coverage: 1500-2019.
`coding_system`	Character. Country coding system: `"cow"` or `"gw"`.

Details

The bundled UN WPP extract has three rows for some China country-years: a combined mainland+Hong Kong+Macao+Taiwan figure, a mainland-only figure, and a Hong Kong-only figure, all of which matched COW code 710 during country-name conversion upstream. The mainland-only figure is consistently the median of the three, so `wpp_pop` is computed as the median `pop` value within each `cow-year` group; for every other country-year (a single row) this is a no-op.

NMC's `-9` missing-data sentinel is recoded to `NA` in `nmc_tpop` and `nmc_upop`.

`fariss_pop` is the model's own latent-scale estimate, taken directly from the `"latent_pop"` rows of the replication file. Its absolute units are not independently verified here; consult Fariss et al. (2022) before using it outside of relative/comparative analysis.

Value

A country-year data frame. Columns differ by dataset, each prefixed with the source's `dataset` key:

wpp: `cow/gw`, `year`, `wpp_pop`.

nmc: `cow/gw`, `year`, `nmc_tpop`, `nmc_upop`.

fariss: `cow/gw`, `year`, `fariss_pop`.

Source

UN WPP: United Nations, Department of Economic and Social Affairs, Population Division (2024). World Population Prospects 2024. https://population.un.org/wpp/

NMC: Singer, J. David, Bremer, Stuart & Stuckey, John (1972). Capability Distribution, Uncertainty, and Major Power War, 1820-1965. National Material Capabilities v7.0. https://correlatesofwar.org/data-sets/national-material-capabilities/

Fariss: Fariss, C.J., Anders, T., Markowitz, J.N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591.

References

United Nations, Department of Economic and Social Affairs, Population Division. (2024). *World Population Prospects 2024*. https://population.un.org/wpp/

Fariss, C. J., Anders, T., Markowitz, J. N., & Barnum, M. (2022). New estimates of over 500 years of historic GDP and population data. *Journal of Conflict Resolution*, 66(3), 553-591. doi:10.1177/00220027211054432

Singer, J. D., Bremer, S., & Stuckey, J. (1972). Capability distribution, uncertainty, and major power war, 1820-1965. In B. Russett (Ed.), *Peace, War, and Numbers* (pp. 19-48). Sage.

Singer, J. D. (1988). Reconstructing the Correlates of War dataset on material capabilities of states, 1816-1985. *International Interactions*, 14, 115-132.

Examples

```
pop_wpp <- load_population_data(1990, 2015, dataset = "wpp", coding_system = "cow")

pop_nmc <- load_population_data(1900, 2015, dataset = "nmc", coding_system = "cow")
```

load_vdem_data *Load V-Dem data*

Description

Loads selected country-year variables from the V-Dem dataset provided by the `vdemdata` package. The returned data can use either COW or Gleditsch-Ward country codes.

Usage

```
load_vdem_data(
  vars = NULL,
  start_year = 1945,
  end_year = 2019,
  coding_system = c("cow", "gw")
)
```

Arguments

`vars`	Character vector of V-Dem variable names to include. If `NULL`, a default set of civil society, repression, civil liberties, democracy, corruption, and rule-of-law indicators is returned.
`start_year`	Integer. First year to include.
`end_year`	Integer. Last year to include.
`coding_system`	Character. Country coding system to use. Either `"cow"` for Correlates of War codes or `"gw"` for Gleditsch-Ward codes.

Details

This function requires the `vdemdata` package. If it is not installed, install it with:

`remotes::install_github("vdeminstitute/vdemdata")`. Because `vdemdata` is not distributed through CRAN, it is always checked conditionally.

For `coding_system = "gw"`, V-Dem COW codes are converted to Gleditsch-Ward codes using `countrycode::countrycode()`.

Value

A data frame with country-year V-Dem indicators. If `coding_system = "cow"`, the data frame contains a `cow` column. If `coding_system = "gw"`, it contains a `gw` column.

Source

Varieties of Democracy (V-Dem) Project. Please cite V-Dem when using these data.

References

Coppedge, M., Gerring, J., Knutsen, C. H., Lindberg, S. I., Teorell, J., Altman, D., Bernhard, M., Fish, M. S., Glynn, A., Hicken, A., et al. (2025). *V-Dem [Country-Year/Country-Date] Dataset v15*. Varieties of Democracy.

Examples

```
if (requireNamespace("vdemdata", quietly = TRUE)) {
vdem_cow <- load_vdem_data(coding_system = "cow")

vdem_selected <- load_vdem_data(
  vars = c("v2x_polyarchy", "v2x_libdem", "v2x_rule"),
  coding_system = "cow"
)

vdem_gw <- load_vdem_data(
  vars = c("v2x_polyarchy", "v2x_libdem"),
  coding_system = "gw"
)
}
```

---

plot_coverage *Plot country-year onset/event coverage of a binary variable*

Description

Draws a country-by-year heatmap of a binary (0/1) event column: `1` years are shaded as an event, `0` years as no event, and years where the state does not appear in `panel` at all (e.g. before independence) or `var` is `NA` are left blank. Useful for eyeballing when and where events (e.g. conflict onsets) actually occurred, as opposed to just where a source has any data.

Usage

```
plot_coverage(panel, var, drop_empty = TRUE, show_labels = TRUE)
```

Arguments

`panel`	A data frame produced by `build_states_panel()`, containing a `cow` or `gw` column, a `year` column, and `var`.
`var`	Character. Name of a binary (0/1) column in `panel`, such as `"ucdp_prio_onset"`.
`drop_empty`	Logical. If `TRUE` (default), drop states with zero years where `var == 1` before plotting. A state that never has the event contributes a row with no blue tiles at all, which adds clutter without showing where/when events happened; set to `FALSE` to include these states anyway.
`show_labels`	Logical. If `TRUE` (default), label each row with its country (or `cow`/`gw` code, if `panel` has no `country` column). Set to `FALSE` to drop these labels, which otherwise overlap and become illegible once `panel` covers more than a few dozen states.

Value

A ggplot object (one tile per country-year, filled by "Event" / "No event", with unobserved country-years left blank). States are ordered top-to-bottom by their number of events, most first. Requires the ggplot2 package.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  build_states_panel(1990, 2015, coding_system = "cow") |>
    add_conflict("ucdp_prio") |>
    plot_coverage("ucdp_prio_onset")

  build_states_panel(1990, 2015, coding_system = "cow") |>
    add_conflict("ucdp_prio") |>
    plot_coverage("ucdp_prio_onset", show_labels = FALSE)
}
```

---

plot_regional_coverage

Compare regional coverage across conflict/campaign datasets

Description

Draws a grouped bar chart of the regional distribution of events (World Bank 7-region classification) for any set of conflict_data() campaign or episode datasets, in the style of Guseinov, Ustyuzhanin, and Korotayev (2026), Figures 1-3. Comparing several datasets this way shows whether they cover similar parts of the world, or whether one systematically over/under-represents a region relative to the others.

Usage

```
plot_regional_coverage(
  datasets,
  start_year,
  end_year,
  coding_system = c("cow", "gw"),
  metric = c("share", "count")
)
```

Arguments

datasets	Character vector of conflict_data() dataset keys to compare, e.g. c("navco1.3", "beissinger", "csra").
start_year, end_year	Integer. Year range to compare (applied to every dataset).
coding_system	"cow" or "gw".
metric	"share" (default; each dataset's bars sum to 100%, so datasets of very different size remain comparable) or "count" (raw number of events).

Value

A `ggplot` object. Requires the `ggplot2` package.

Source

Guseinov, R., Ustyuzhanin, V., & Korotayev, A. (2026). Talking about the [same] revolution? A comparative analysis of main datasets of revolutionary events. *Defence and Peace Economics*, 1–31. doi:10.1080/10242694.2026.2704178

References

Guseinov, R., Ustyuzhanin, V., & Korotayev, A. (2026). Talking about the [same] revolution? A comparative analysis of main datasets of revolutionary events. *Defence and Peace Economics*, 1-31. doi:10.1080/10242694.2026.2704178

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot_regional_coverage(
    c("navco1.3", "navco2.1", "beissinger", "csra"),
    start_year = 1950, end_year = 2013
  )
}
```

---

plot_temporal_coverage

Compare temporal coverage across conflict/campaign datasets

Description

Draws a line chart of the number of events per time period for any set of `conflict_data()` campaign or episode datasets, in the style of Guseinov, Ustyuzhanin, and Korotayev (2026), Figures 4-5. Comparing several datasets this way shows whether they track the same broad historical trends, and where one dataset picks up a surge or lull that the others miss.

Usage

```
plot_temporal_coverage(
  datasets,
  start_year,
  end_year,
  coding_system = c("cow", "gw"),
  period_length = 5,
  by_region = FALSE
)
```

Arguments

datasets	Character vector of `conflict_data()` dataset keys to compare, e.g. `c("navco1.3", "beissinger", "csra")`.
start_year, end_year	Integer. Year range to compare (applied to every dataset).
coding_system	`"cow"` or `"gw"`.
period_length	Integer. Width in years of each time bin (default 5, matching the source article).
by_region	Logical. If `TRUE`, facet by World Bank region (one panel per region, as in Figure 4). If `FALSE` (default), a single global panel (as in Figure 5).

Value

A `ggplot` object. Requires the `ggplot2` package.

Source

Guseinov, R., Ustyuzhanin, V., & Korotayev, A. (2026). Talking about the [same] revolution? A comparative analysis of main datasets of revolutionary events. *Defence and Peace Economics*, 1–31. doi:10.1080/10242694.2026.2704178

References

Guseinov, R., Ustyuzhanin, V., & Korotayev, A. (2026). Talking about the [same] revolution? A comparative analysis of main datasets of revolutionary events. *Defence and Peace Economics*, 1-31. doi:10.1080/10242694.2026.2704178

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot_temporal_coverage(
    c("navco1.3", "navco2.1", "beissinger", "csra"),
    start_year = 1950, end_year = 2013
  )
}
```

Index

∗ **datasets**
    cn_bib, 15

add_conflict, 3
add_conflict(), *11*, *17*, *19*, *20*
add_from_peacesciencer, 5
add_gdp, 6
add_lag, 7
add_leader_data, 7
add_military_expenditure, 8
add_population, 9
add_regions, 10
add_spell_duration, 11
add_vdem, 12
as.Date(), *20*
as_peacesciencer_panel, 13
as_peacesciencer_panel(), *5*

build_states_panel, 14

cn_bib, 15
cn_cite, 15
cn_cite(), *15*
conflict_data, 16

harmonize_conflict_data, 19

load_gdp_data, 21
load_leader_data, 23
load_military_expenditure_data, 24
load_population_data, 26
load_vdem_data, 28

plot_coverage, 29
plot_regional_coverage, 30
plot_temporal_coverage, 31

states::state_panel(), *14*