Package: rict (via r-universe)

June 1, 2026

Title Redistricting in Clean Tables

Version 0.0.1

Description Provides a suite of tools to create tables that accompany maps. The tools create clean, informative tables for electoral outcomes, compactness, and other district-level quantities. Most tools are aimed at the redistricting context, but are broadly applicable to other electoral data.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports cli, dplyr, geomander, geos, ggplot2, gt, purrr, redist, redistmetrics, rlang, sf, stringr, tibble

Depends R (>= 4.1.0)

LazyData true

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/website christopherkenny/ctktemplate

VignetteBuilder knitr

URL http://christophertkenny.com/rict/, https://github.com/christopherkenny/rict

BugReports https://github.com/christopherkenny/rict/issues

Language en-US

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libnode-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-05-01 16:15:31 UTC

RemoteUrl https://github.com/christopherkenny/rict

RemoteRef HEAD

RemoteSha 982e9ca9ab50bc251667bcddb1e0419d92815e65

Contents

`data_color_party` *Color Columns with Partisan Scales*

Description

Color Columns with Partisan Scales

Usage

```
data_color_party(tab, columns = gt::everything(), ...)
```

Arguments

`tab`	A `gt` table with class `gt::gt_tbl`
`columns`	the columns to color with partisan colors
`...`	additional arguments passed on to `gt::data_color()`

Value

A gt::gt

Examples

```
rict(wv_plans, 'cd_2020') |>
  data_color_party(columns = 'e_dvs')
```

`gt_get_data` *Extract data from a* `gt`

Description

Extract data from a `gt`

Usage

```
gt_get_data(tab)
```

Arguments

`tab`	A `gt` table with class `gt::gt_tbl`

Value

A dplyr::tibble

Examples

```
rict(wv) |> gt_get_data()
```

`gt_hide_lists` *Hide List Columns in* `gt`

Description

Hide List Columns in `gt`

Usage

```
gt_hide_lists(tab)
```

Arguments

`tab`	A `gt` table with class `gt::gt_tbl`

Value

A gt::gt

Examples

```
wv |>
  gt::gt() |>
  gt_hide_lists()
```

`gt_plot_compactness` *Add Compactness Metric Plots to a* `gt`

Description

Add Compactness Metric Plots to a `gt`

Usage

```
gt_plot_compactness(
  tab,
  shp,
  plan,
  measures = guess_comp(tab),
  height = 200,
  ...
)
```

Arguments

`tab`	A `gt` table with class `gt::gt_tbl`
`shp`	An `sf` object
`plan`	A numeric vector with one entry for each precinct in `shp`.
`measures`	A character vector indicating which measures to plot. Uses `guess_comp()` if not supplied.
`height`	height, in pixels, of each image. Default is 200.
`...`	additional arguments. Not currently passed on.

Value

A gt::gt

Examples

```
rict(wv_plans, 'cd_2020') |>
  gt_plot_compactness(wv, wv$cd_2020)
```

gt_plot_sf *Add* sf *Geometry Plots to a* gt

Description

Add sf Geometry Plots to a gt

Usage

```
gt_plot_sf(tab, name, height = 100, ...)
```

Arguments

tab	A gt table with class gt::gt_tbl
name	Name for sf column in gt
height	height, in pixels, of each image. Default is 100
...	additional arguments passed on to geom_sf()

Value

A gt::gt

Examples

```
wv_dist <- wv |>
  dplyr::group_by(cd_2020) |>
  dplyr::summarize()
gt::gt(wv_dist) |> gt_plot_sf()
```

plot_compactness *Create Plots for Common Compactness Metrics*

Description

Create Plots for Common Compactness Metrics

Usage

```
plot_compactness(
  shp,
  plan,
 measure = c("Polsby Popper", "Schwartzberg", "Reock", "Convex Hull", "Length Width",
    "Skew", "Box Reock"),
  fill_color = "deeppink"
)
```

Arguments

shp	An sf object
plan	A numeric vector with one entry for each precinct in shp.
measure	A character indicating which measure to plot. Default is 'Polsby Popper'.
fill_color	hex or color name to fill the shape. A second entry can be provided to fill the background.

Value

list of ggplot2 plots

Examples

```
plot_compactness(wv, wv$cd_2020)
```

---

rict *Create a Summary Table from Redistricting Data*

Description

Creates a formatted gt::gt table summarizing redistricting plans or maps.

Usage

```
rict(x, plan, ...)
```

Arguments

x	A redist_map or redist_plans object.
plan	For redist_plans: draw name or number to display. For redist_map: column or vector of district assignments (defaults to existing plan via redist::get_existing()).
...	Additional arguments passed to methods.

Value

A gt::gt table

Examples

```
rict(wv)
rict(wv_plans, 'cd_2020')
```

rict_boundary *Display boundary information in a table*

Description

Identifies neighboring precincts along a district boundary and displays them as adjacent pairs, with one row per pair of neighboring precincts from different districts.

Usage

```
rict_boundary(map, plan, seam, columns, adj_col = "adj", as_gt = TRUE)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
seam	Pair of districts in plan to focus on.
columns	columns in map to display in the output
adj_col	Name of column in map that contains adjacency information.
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_boundary(map = wv, plan = wv$cd_2020, seam = c(1, 2), columns = pop)
```

rict_compactness *Display compactness measures in a table*

Description

Display compactness measures in a table

Usage

```
rict_compactness(
  map,
  plan,
  measures = list(comp_polsby = redistmetrics::comp_polsby, comp_schwartz =
   redistmetrics::comp_schwartz, comp_reock = redistmetrics::comp_reock, comp_ch =
     redistmetrics::comp_ch),
  as_gt = TRUE
)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
measures	a list of named functions to score compactness
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_compactness(map = wv, plan = wv$cd_2020)
```

---

rict_component *Display population data by administrative unit in a table*

Description

Display population data by administrative unit in a table

Usage

```
rict_component(map, plan, admin, as_gt = TRUE)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
admin	column names in map without NA values to calculate administrative splits for
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_component(map = wv, plan = wv$cd_2020, admin = 'county')
```

rict_contiguity *Display contiguity info in a table*

Description

Display contiguity info in a table

Usage

```
rict_contiguity(map, plan, adj = NULL, adj_col = "adj", as_gt = TRUE)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
adj	An adjacency list (zero-indexed). If provided, used directly instead of looking up adj_col in map.
adj_col	Name of column in map that contains adjacency information.
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_contiguity(map = wv, plan = wv$cd_2020)
```

rict_demographics *Display demographic data in a table*

Description

Display demographic data in a table

Usage

```
rict_demographics(map, plan, normalize = TRUE, as_gt = TRUE)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
normalize	Logical. Should columns be normalized to percentages? Default: TRUE.
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a `gt::gt_tbl` if `as_gt = TRUE`, otherwise a tibble::tibble

Examples

```
rict_demographics(map = wv, plan = wv$cd_2020)
```

---

rict_elections *Display electoral data in a table*

Description

Display electoral data in a table

Usage

```
rict_elections(map, plan, as_gt = TRUE)
```

Arguments

`map`	A `redist_map` or `sf` object.
`plan`	Column in `map` or vector of district assignments.
`as_gt`	Logical. Should output be a `gt` table? Default: `TRUE`.

Value

a `gt::gt_tbl` if `as_gt = TRUE`, otherwise a tibble::tibble

Examples

```
rict_elections(map = wv, plan = wv$cd_2020)
```

---

rict_population *Display population parity in a table*

Description

Display population parity in a table

Usage

```
rict_population(map, plan, as_gt = TRUE)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_population(map = wv, plan = wv$cd_2020)
```

---

rict_splits *Display splits data in a table*

Description

Display splits data in a table

Usage

```
rict_splits(
  map,
  plan,
  admin = NULL,
  subadmin = NULL,
  total = admin,
  multi = admin,
  as_gt = TRUE
)
```

Arguments

map	A redist_map or sf object.
plan	Column in map or vector of district assignments.
admin	column names in map without NA values to calculate administrative splits for
subadmin	column names in map with NA values to calculate administrative splits for
total	column names in map without NA values to calculate total splits for
multi	column names in map without NA values to calculate multi-splits for
as_gt	Logical. Should output be a gt table? Default: TRUE.

Value

a gt::gt_tbl if as_gt = TRUE, otherwise a tibble::tibble

Examples

```
rict_splits(map = wv, plan = wv$cd_2020, admin = 'state')
```

---

wv *West Virginia Geographic Data*

Description

This file contains demographic, partisan, and geographic data for West Virginia at the county level.

Format

`redist_map` object

`GEOID` US Census Geographic Identifier

`NAME` County name

`state` State name

`county` County name

`pop` Total population

`pop_hisp` Hispanic population

`pop_white` White, non-Hispanic population

`pop_black` Black, non-Hispanic population

`pop_aian` American Indian and Alaskan Native, non-Hispanic population

`pop_asian` Asian, non-Hispanic population

`pop_nhpi` Native Hawaiian and Pacific Islander, non-Hispanic population

`pop_other` Other, non-Hispanic population

`pop_two` Two or More Races, non-Hispanic population

`vap` voting age population

`vap_hisp` Hispanic voting age population

`vap_white` White, non-Hispanic voting age population

`vap_black` Black, non-Hispanic voting age population

`vap_aian` American Indian and Alaskan Native, non-Hispanic voting age population

`vap_asian` Asian, non-Hispanic voting age population

`vap_nhpi` Native Hawaiian and Pacific Islander, non-Hispanic voting age population

`vap_other` Other, non-Hispanic voting age population

`vap_two` Two or More Races, non-Hispanic voting age population

`cd_2010` 2010 congressional district lines smoothed to the county level

`cd_2020` 2020 congressional district lines

`pre_20_dem_bid` votes for Biden 2020, President (D)

pre_20_rep_tru votes for Trump 2020, President (R)

arv_20 average Republican vote in 2020

adv_20 average Democratic vote in 2020

nrv normal Republican vote

ndv normal Democratic vote

adj adjacency list, zero-indexed

sample_1 random sampled plan from redist 50 states project

sample_2 random sampled plan from redist 50 states project

sample_3 random sampled plan from redist 50 states project

sample_4 random sampled plan from redist 50 states project

sample_5 random sampled plan from redist 50 states project

sample_6 random sampled plan from redist 50 states project

sample_7 random sampled plan from redist 50 states project

sample_8 random sampled plan from redist 50 states project

sample_9 random sampled plan from redist 50 states project

sample_10 random sampled plan from redist 50 states project

sample_11 random sampled plan from redist 50 states project

sample_12 random sampled plan from redist 50 states project

geometry sf geometry

Examples

```
data(wv)
```

---

wv_plans *West Virginia Redistricting Plans*

Description

This file contains 10 sampled plans from the ALARM Project 50 states project and the 2020 congressional plan for WV.

Format

redist_plans object

draw draw identifier

district district number

total_pop Total population

total_vap voting age population

plan_dev Maximum deviation from perfect population parity

comp_edge Fraction of Edges Kept compactness

comp_polsby Polsby Popper compactness

pop_white White, non-Hispanic population

pop_black Black, non-Hispanic population

pop_hisp Hispanic population

pop_aian American Indian and Alaskan Native, non-Hispanic population

pop_asian Asian, non-Hispanic population

pop_nhpi Native Hawaiian and Pacific Islander, non-Hispanic population

pop_other Other, non-Hispanic population

pop_two Two or More Races, non-Hispanic population

vap_hisp Hispanic voting age population

vap_white White, non-Hispanic voting age population

vap_black Black, non-Hispanic voting age population

vap_aian American Indian and Alaskan Native, non-Hispanic voting age population

vap_asian Asian, non-Hispanic voting age population

vap_nhpi Native Hawaiian and Pacific Islander, non-Hispanic voting age population

vap_other Other, non-Hispanic voting age population

vap_two Two or More Races, non-Hispanic voting age population

pre_20_dem_bid votes for Biden 2020, President (D)

pre_20_rep_tru votes for Trump 2020, President (R)

arv_20 average Republican vote in 2020

adv_20 average Democratic vote in 2020

nrv normal Republican vote

ndv normal Democratic vote

ndshare normal Democratic share in the district

e_dvs expected Democratic share in the district

pr_dem proportion of districts where Democrats win reconstructed elections

e_dem expected number of Democratic seats

pbias partisan bias

egap efficiency gap

Examples

```
data(wv_plans)
```

Index