Package: codec (via r-universe)

November 3, 2026

Title Community Data Explorer for Cincinnati

Version 3.1.0.9000

Description This repository serves as the definition of the CoDEC data specifications and provides helpers to create, validate, release, and read CoDEC data.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests base64enc, bslib, crosstalk, testthat (>= 3.0.0), DT, devtools, htmltools, leaflet, markdown, plotly, roxygen2, rmarkdown, dpkg, knitr, shiny

Config/testthat/edition 3

URL https://geomarker.io/codec

BugReports https://github.com/geomarker-io/codec/issues

Imports rlang, tidyselect, stringr, scales, glue, here, pins, tibble, sf, dplyr, withr, s2

Depends R (>= 3.5)

LazyData true

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

Repository https://geomarker-io.r-universe.dev

Date/Publication 2026-08-05 14:06:00 UTC

RemoteUrl https://github.com/geomarker-io/codec

RemoteRef v3.1.0

RemoteSha e381b37b4aeeee3c60248f287b4d142e18e01192

Contents

as_codec_tbl 2
cincy_addr_geo 3
cincy_census_geo 4
cincy_city_geo 5
cincy_neighborhood_geo 5
cincy_zcta_geo 6
codec_as_sf 6
codec_colors 7
codec_interpolate 8
codec_latest_annual 9
codec_read 9
install_cagis_data 11

Index **13**

`as_codec_tbl` *Convert a data frame, name, and description into a CoDEC table*

Description

CoDEC Specifications:

1. The data must include a census tract identifier column (i.e., `census_tract_id_2010` or `census_tract_id_2020`). The column must contain 11-digit GEOID identifiers for every census tract in Hamilton County, OH.
2. Data must be structured in a tidy format such that each row is an observation for a specific census tract at a specific year (and month). This means that the data includes a year column (`year`), an integer year representing the vintage of the data (e.g. `2021`). The data can optionally include a month column (`month`), an integer month of the year.
3. The name must only contain lower case alphanumeric characters, `-`, or `_`
4. The description should be markdown text and the first line must contain the title of the CoDEC data table as a level one header (e.g., `# My Community Data`). Titles must be less than 80 characters.

Usage

```
as_codec_tbl(x, name, description = character())
```

Arguments

`x`	data.frame or tibble meeting CoDEC data specifications above
`name`	name of CoDEC table
`description`	markdown text describing the CoDEC table

Value

a codec_tbl object

Examples

```
tibble::tibble(
  census_tract_id_2020 = cincy_census_geo("tract", "2020")$geoid,
  n_things = 823,
  year = 2024L
) |>
  as_codec_tbl(
    name = "n_things",
    paste0(
      "# Number of Things\n",
      "Number of things were averaged by census tract ",
      "using the survey from 2024"
    )
  )
```

---

cincy_addr_geo *Cincy address geographies*

Description

CAGIS data (see `install_cagis_data()`) provides a list of all addresses in Hamilton County. The s2 cell is derived from LONGITUDE and LATITUDE fields in the CAGIS address database.

Usage

```
cincy_addr_geo(packaged = TRUE)
```

Arguments

packaged	logical; use the data included with the package instead of (down)loading from the source data?

Details

This function previously excluded addresses with type milemarker (`MM`), park (`PAR`), infrastructure project (`PRJ`), cell tower (`CTW`), vacant or commercial lot (`LOT`), and other miscellaneous non-residential addresses (`MIS`, `RR`, `TBA`). Now that they are included, see the examples for how to filter these out.

Value

a simple features object with columns `cagis_address`, `cagis_address_place`, `cagis_address_type`, `cagis_s2`, `cagis_parcel_id`, `cagis_is_condo`, and a geometry column (`s2_geography`)

Examples

```
cincy_addr_geo() |>
  dplyr::filter(
    !cagis_address_type %in% c(
      "MM", "PAR", "PRJ", "CTW",
      "LOT", "MIS", "RR", "TBA"
    )
  )
```

---

cincy_census_geo *Cincy census tracts and block groups*

Description

Read tract and block group ("bg") geographies from the online Census TIGER/Line files into R.

Usage

```
cincy_census_geo(
  geography = c("tract", "bg"),
  vintage = as.character(2024:2013),
  packaged = TRUE
)
```

Arguments

geography	which type of cincy census geography to return
vintage	a character vector of a year corresponding to the vintage of TIGER/Line data
packaged	logical; use the data included with the package instead of (down)loading from the source data?

Details

Compressed shapefiles are downloaded from TIGER into an R user data directory and will be cached for use across other R sessions.

Value

a simple features object with a geographic identifier column (geoid) and a geometry column (s2_geography)

Examples

```
cincy_census_geo("tract", "2020")
cincy_census_geo("bg", "2020")
```

`cincy_city_geo` *cincy_city_geo()*

Description

cincy_city_geo()

Usage

```
cincy_city_geo()
```

Examples

```
## Not run:
cincy_city_geo()

## End(Not run)
```

`cincy_neighborhood_geo` *Cincy neighborhood geographies*

Description

CAGIS data (see `install_cagis_data()`) provides community council boundaries, but these boundaries can overlap and do not align with census geographies or ZIP codes. By default, the statistical neighborhood approximations are instead returned, which are calculated by aggregating census tracts into 50 matching neighborhoods.

Usage

```
cincy_neighborhood_geo(
  geography = c("statistical_neighborhood_approximations", "community_council"),
  packaged = TRUE
)
```

Arguments

`geography`	which type of cincy neighborhood geography to return
`packaged`	logical; use the data included with the package instead of (down)loading from the source data?

Value

a simple features object with a geographic identifier column (`geoid`) and a geometry column (`s2_geography`)

Examples

```
cincy_neighborhood_geo()
```

`cincy_zcta_geo` *Cincy ZIP Code Tabulation Areas*

Description

Read ZIP Code Tabulation Areas (ZCTAs) geographies from the online Census TIGER/Line files into R.

Usage

```
cincy_zcta_geo(vintage = as.character(2024:2013), packaged = TRUE)
```

Arguments

`vintage`	a character vector of a year corresponding to the vintage of TIGER/Line data
`packaged`	logical; use the data included with the package instead of (down)loading from the source data?

Value

a simple features object with a geographic identifier column (`geoid`) and a geometry column (`s2_geography`)

Examples

```
cincy_zcta_geo("2020")
```

`codec_as_sf` *Coerce a CoDEC data table into a simple features object*

Description

The name of the census tract column in the CoDEC data table is used to add the appropriate census tract s2 geography column.

Usage

```
codec_as_sf(x)
```

Arguments

`x`	a CoDEC data table

Details

Tract identifiers do not change across decennial censuses, but the digital representation of their boundaries may be improved over time. Here, data tables using 2010 tract identifiers use the TIGER/Line 2019 tract shapefiles and data tables using 2020 tract identifiers use the TIGER/Line 2020 tract shapefiles.

Value

a simple features object with a geometry column (`s2_geography`) in addition to the columns in `x`

Examples

```
codec_as_sf(codec_read("property_code_enforcements"))
```

`codec_colors` *CoDEC colors*

Description

CoDEC colors

Usage

```
codec_colors(n = NULL)
```

Arguments

`n`	a numeric vector of color numbers or character vector of color names; if NULL returns named vector of available colors

Examples

```
plot(1:8, rep(1, 8), col = codec_colors(1:8),
  pch = 19, cex = 10, axes = FALSE, xlab = "", ylab = "")
text(1:8, rep(1.1, 8), labels = names(codec_colors()))
```

`codec_interpolate` *Spatially interpolate community-level data*

Description

Census block-level weights are used to spatially interpolate CoDEC data packages from the census tract-level to other Cincy geographies.

Usage

```
codec_interpolate(from, to, weights = c("pop", "homes", "area"))
```

Arguments

`from`	a CoDEC data package
`to`	A simple features object returned by one of the `cincy_*_geo()` functions (for example `cincy_census_geo()`, `cincy_neighborhood_geo()`, or `cincy_zcta_geo()`)
`weights`	which census block-level weights to use; see details

Details

Block-level total population (`pop`), total number of homes (`homes`), or total land area (`area`) from the 2020 Census can be chosen to use for the weights. Geospatial intersection happens after transforming geographies to epsg:5072. See `codec_as_sf()` for adding geography to a CoDEC data package. Variables beginning with "n_" are interpolated using a weighted sum; all other variables are interpolated using a weighted mean.

Value

a tibble with a new geographic identifier column for the `to` target geography (`geoid`) in addition to the interpolated columns in `from`

Examples

```
codec_interpolate(codec_read("acs_measures"),
                  cincy_neighborhood_geo())
codec_interpolate(codec_read("property_code_enforcements"),
                  cincy_census_geo("tract", "2019"))
```

codec_latest_annual *CoDEC latest annual*

Description

CoDEC data is harmonized as the most recently available annual (or annual average of monthly) values at the census tract 2020 geography. The year represents the year this table was assembled and mainly used as a placeholder so this object can be used with functions that take a `codec_tbl` object (see `?as_codec_tbl`). See the description metadata for the actual latest years used for each CoDEC table.

Usage

```
codec_latest_annual
```

Format

An object of class `codec_tbl` (inherits from `tbl_df`, `tbl`, `data.frame`) with 226 rows and 63 columns.

Examples

```
glue::glue(attr(codec_latest_annual, "description"))
tibble::glimpse(codec_latest_annual)
```

codec_read *CoDEC online data catalog*

Description

The CoDEC data catalog is shipped with the package and older versions can be read from GitHub alongside the source code for this package.

- Use `codec_read()` as a shortcut to read a CoDEC table into R as a `codec_tbl` object (see `?as_codec_tbl`)
- Use `codec_list()` as a shortcut to list available CoDEC table pins
- `codec_board()` can be used to create a pin board object (see `?pins::pins`) based on a specific version of the codec package

Usage

```
codec_read(
  name,
  board = codec_board(),
  to = NULL,
  weights = c("pop", "homes", "area"),
  include_geography = FALSE
)

codec_list(board = codec_board())

codec_board(
  version = paste0("v", utils::packageVersion("codec")),
  cache = NULL,
  use_cache_on_failure = rlang::is_interactive(),
  headers = NULL
)
```

Arguments

name	the name of the CoDEC table in the CoDEC data catalog.
board	a pins board object; create with codec_board() to read from the bundled catalog or earlier versions of the catalog, or to change the caching behavior of the pins package
to	optional target geography for interpolation; supply the output of cincy_census_geo(), cincy_neighborhood_geo(), or cincy_zcta_geo() to interpolate the table while reading
weights	which census block-level weights to use when to is supplied; passed to codec_interpolate()
include_geography	logical; include the s2_geography column in the result? Defaults to FALSE
version	specify a version of the online data catalog using a commit SHA, tag, or branch of geomarker-io/codec; uses the bundled board for the installed package version by default
cache	Cache path. Every board requires a local cache to avoid downloading files multiple times. The default stores in a standard cache location for your operating system, but you can override if needed.
use_cache_on_failure	If the pin fails to download, is it ok to use the last cached version? Defaults to is_interactive() so you'll be robust to poor internet connectivity when exploring interactively, but you'll get clear errors when the code is deployed.
headers	Named character vector for additional HTTP headers (such as for authentication). See connect_auth_headers() for Posit Connect support.

Details

The pin for each CoDEC table has versions (see ?pins::pin_versions), but codec_board() can be used to specify a state of the online data catalog based on the version of the codec package. (See

examples)

Value

For `codec_read()`, a `codec_tbl` by default, or an interpolated tibble / simple-features tibble when `to` or `include_geography` is used

For `codec_list()`, a character vector CoDEC table names

For `codec_board()`, a pins_board object

Examples

```
# list available CoDEC tables
codec_list()

# read a CoDEC table and inspect its metadata
d <- codec_read("traffic")
head(d)
attr(d, "title")
message(attr(d, "description"))

# interpolate while reading
codec_read("acs_measures", to = cincy_neighborhood_geo())
codec_read("acs_measures", include_geography = TRUE)

# inspect the bundled board or read from an older online version
codec_board() |>
  pins::pin_versions("crime")
codec_board("v3.0.0") |>
  pins::pin_versions("crime")
```

---

install_cagis_data *Install CAGIS GIS database*

Description

This installs the CAGIS Open Data GIS database (`.gdb`) into the data directory for the codec package. Once downloaded, it will be reused across R sessions on the same computer. The geodatabase contains many layers that are updated quarterly. Historical geodatabases are not available here.

Usage

```
install_cagis_data(
  cagis_data_url =
   "https://www.cagis.org/Opendata/Quarterly_GIS_Data/CAGISOpenDataQ1_2025.gdb.zip"
)
```

Arguments

`cagis_data_url`	the url to the CAGIS Open Data .gdb.zip file; this changes quarterly, so check the CAGIS Open Data page for something more recent if the file cannot be found.

See Also

This function is called by `cincy_neighborhood_geo()` and `cincy_city_geo()`, plus other functions that import individual layers.

Examples

```
## Not run:
install_cagis_data()
sf::st_layers(install_cagis_data())$name

## End(Not run)
```

Index

∗ **datasets**
codec_latest_annual, 9

as_codec_tbl, 2

cincy_addr_geo, 3
cincy_census_geo, 4
cincy_city_geo, 5
cincy_neighborhood_geo, 5
cincy_zcta_geo, 6
codec_as_sf, 6
codec_board *(*codec_read*)*, 9
codec_colors, 7
codec_interpolate, 8
codec_latest_annual, 9
codec_list *(*codec_read*)*, 9
codec_read, 9
connect_auth_headers(), *10*

install_cagis_data, 11