

Package: geomarker (via r-universe)

September 24, 2026

Title Geomarker Assessment using S2 Cells and Dates

Version 0.1.1

Description The `s2_cell_dates` vector class stores an `s2::s2_cell` vector with a list of date vectors, one for each s2 cell. Geomarker assessment functions use this format to complete spatiotemporal linkage of extant environmental and sociodemographic data.

Depends R (>= 4.2.0)

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports s2, digest, stow (>= 0.3.0), vctrs

Remotes cole-brokamp/stow

Suggests curl, fs, jsonlite, dplyr, sf, tibble, pillar, readr, terra, tidync, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libssl-dev

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

Date/Publication 2026-08-25 15:30:13 UTC

RemoteUrl <https://github.com/geomarker-io/geomarker>

RemoteRef v0.1.1

RemoteSha 2e1255b90a1d744d61bd084d58a35513226cf400

Contents

<code>as_s2cd</code>	2
<code>get_daily_smoke_data</code>	3
<code>get_elevation_summary</code>	4
<code>get_evi_summary</code>	5

get_gridmet_data	6
get_merra_data	7
get_narr_data	8
get_nei_point_summary	9
get_nlcd_fct_imp_summary	11
get_smoke_summary	12
get_tiger_bg	12
get_traffic_summary	13
impute_date_ranges	14
is_s2cd	16
s2cd	16
s2cd_dates	18
s2cd_example	18
s2cd_example_cincy	19
s2cd_within	19
s2cd_years	20

Index	21
--------------	-----------

as_s2cd	<i>Convert other objects into a s2_cell_dates (s2cd) vector</i>
---------	---

Description

Convert other R objects into s2cd vectors

Usage

```
as_s2cd(x, ...)
```

Arguments

- | | |
|-----|--|
| x | an object to convert |
| ... | passed to methods; for the data.frame method, passed to s2cd() |

Methods implemented for

- data.frame: must have column "s2_cell" and list-column "dates"
- s2cd: returned as-is

Examples

```
data.frame(  
  s2_cell = s2::as_s2_cell(c("8841b39a7c46e25f", "8841a45555555555")),  
  dates = I(list(  
    as.Date("2026-01-01"),  
    c(as.Date("2024-09-13"), as.Date("2024-09-20"))  
  ))  
)
```

```

    ) |>
    as_s2cd()

as_s2cd(s2cd_example())

```

get_daily_smoke_data *Get wildfire smoke plume data*

Description

NOAA's Hazard Mapping System (HMS) operates daily in near real-time by outlining the smoke polygon of each distinct smoke plume and classifying it as "Heavy", "Light", or "Medium" based on its apparent thickness.

Usage

```
get_daily_smoke_data(x, ...)
```

Arguments

x	Date vector; files are organized and retrieved by date for all of CONUS
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Details

Daily files for HMS smoke data are used again instead of re-downloading, unless there is an updated version of the daily data available. For more details, see <https://www.ospo.noaa.gov/products/land/hms.html#about>.

Value

a list of daily tibbles, each with columns for geometry and density

Examples

```

withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_daily_smoke_data(as.Date(c("2024-02-09", "2024-06-10", "2024-06-11", "2024-06-12")))

```

get_elevation_summary *Summarize nearby elevation*

Description

Summarizes elevations within the buffer distance of each s2 cell; dates are not used to link to elevation data.

Usage

```
get_elevation_summary(x, fun = stats::median, buffer = 800, ...)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
fun	function to summarize extracted values
buffer	distance from s2 cell (in meters) to summarize data
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker, and etag is always set to FALSE for this versioned release asset.

Details

Elevation data come from a reproducible snapshot of the U.S. Geological Survey 3D Elevation Program (3DEP) Bare Earth DEM Dynamic service. The service snapshot, reflecting 3DEP data published through July 20, 2026, was exported directly as a single-band signed 16-bit GeoTIFF in EPSG:5070 at an 800 m grid and is distributed as a versioned geomarker release asset. See <https://www.usgs.gov/3d-elevation-program/about-3dep-products-services> and <https://elevation.nationalmap.gov/arcgis/rest/services/3DEPElevation/ImageServer>. Data courtesy of the U.S. Geological Survey.

Value

numeric vector of elevation summaries

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_elevation_summary(s2cd_example())
get_elevation_summary(s2cd_example(), stats::sd, 1400)
```

get_evi_summary	<i>MODIS enhanced vegetation index summary</i>
-----------------	--

Description

Links annual enhanced vegetation index (EVI) based on the location of each s2 cell and dates. For each year, EVI is defined as the mean EVI within buffer meters of the location, extracted from a 250 m annual MODIS Terra/Aqua EVI composite whose pixels are the median of all good or marginal-quality 16-day EVI observations in that year.

Usage

```
get_evi_summary(x, buffer = 400, overwrite = FALSE, quiet = FALSE)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
buffer	distance from s2 cell (in meters) to summarize data
overwrite	logical; replace existing annual EVI rasters stored as durable managed local copies?
quiet	logical; show download progress messages?

Details

Data are downloaded from the [Microsoft Planetary Computer](https://planetarycomputer.microsoft.com/) mirror of NASA LP DAAC MOD13Q1/MYD13Q1 Version 6.1 Cloud Optimized GeoTIFFs. See <https://planetarycomputer.microsoft.com/dataset/modis-13Q1-061> for collection details. Source 16-day EVI and pixel reliability rasters are staged under the get_evi_summary subdirectory of geomarker's durable managed local copy directory only when an annual composite is missing. If a build fails, complete source downloads are reused on the next attempt and removed after the annual composite is successfully written. EVI source rasters are downloaded by MODIS tile; 14 MODIS tiles intersect the contiguous United States. Planetary Computer assets are accessed with reusable container SAS tokens to reduce signing API calls. Annual composite rasters are retained there as durable managed local copies. EVI source raster values are scaled by 0.0001 while creating annual composites. EVI is a greenness index designed to emphasize photosynthetically active vegetation while reducing atmospheric and soil background effects. In this MODIS product, scaled EVI values are expected to range from about -0.2 to 1.0; higher values indicate denser or more vigorous green vegetation, values near 0 indicate little green vegetation, and negative values usually indicate water, snow, cloud, or other non-vegetated surfaces or artifacts.

Value

a list of numeric vectors of annual EVI summaries.

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_evi_summary(s2cd_example_cincy(2L))
get_evi_summary(s2cd_example_cincy(2L), buffer = 1200, quiet = TRUE)
```

get_gridmet_data	<i>gridMET surface meteorological data</i>
------------------	--

Description

Links daily, high spatial resolution (~4-km) meteorological data based on the location of each s2_cell and dates.

Usage

```
get_gridmet_data(
  x,
  gridmet_var = c("tmmx", "tmmn", "pr", "srad", "vs", "th", "rmax", "rmin", "sph"),
  ...
)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
gridmet_var	character; name of gridMET variable
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Details

Data is downloaded from the [Climatology Lab](#) and is available for the contiguous US from 1979-yesterday. Data for dates within the last 60 days are considered preliminary and subject to change.

Value

a list of numeric vectors of gridMET values

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_gridmet_data(s2cd_example_cincy(2L), "tmmx")
```

get_merra_data	<i>MERRA-2 aerosol diagnostics data</i>
----------------	---

Description

Links daily surface PM2.5 and its aerosol components from NASA's MERRA-2 M2T1NXAER v5.12.4 product to each location and date in an s2cd vector.

Usage

```
get_merra_data(x, ...)

install_merra_data(
  merra_month,
  source = c("release", "earthdata"),
  overwrite = FALSE,
  quiet = FALSE
)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
...	passed to <code>stow::stow()</code> when an official release asset is needed. The package and subdir arguments are fixed by geomarker.
merra_month	character vector of calendar months in YYYY-MM format, from 2017 onward
source	either "release" to install the matching official half-year asset or "earthdata" to build each complete month from NASA
overwrite	logical; recreate monthly data from the latest CMR listing? Matching staged daily source files are still reused.
quiet	logical; suppress progress messages?

Details

Each package release records one complete January–June and one complete July–December asset per available year. A fully available calendar month may instead be built locally with `install_merra_data(source = "earthdata")`. Local monthly data take precedence over a released half-year asset.

Source builds require a free NASA Earthdata Login account. Follow NASA's [registration guide](#) to create and activate an account. Before the first source build, sign in to the [Cloud OPeNDAP service](#) and authorize the Hyrax in the cloud application when prompted. NASA also documents how to [preauthorize an application](#) if the prompt does not appear.

Set the account credentials in `EARTHDATA_USER` and `EARTHDATA_PASSWORD`. They are written to temporary private netrc and cookie files in geomarker's durable managed local copy directory while NASA's authenticated redirects are followed.

Source concentrations are converted to micrograms per cubic meter and averaged across each day. Total PM_{2.5} is calculated as $DUSMASS25 + OCSMASS + BCSMASS + SSSMASS25 + SO4SMASS * 132.14 / 96.06$.

Value

A named list with one data frame per input location and one row per requested date. Columns are `merra_dust`, `merra_oc`, `merra_bc`, `merra_ss`, `merra_so4`, and `merra_pm25`. Unavailable periods contain missing values and produce one warning.

`install_merra_data()` returns a named character vector of installed paths. Months in the same released half-year may have the same path.

References

https://disc.gsfc.nasa.gov/datasets/M2T1NXAER_5.12.4/summary

https://gmao.gsfc.nasa.gov/gmao-products/merra-2/faq_merra-2/

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package("geomarker", "gmrkr--8841"),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_merra_data(s2cd_example())
```

get_narr_data

NARR surface meteorological data

Description

Links daily, 0.3 degrees gridded (~ 32 sq km) meteorological monolevel data based on the location of each `s2_cell` and dates.

Usage

```
get_narr_data(
  x,
  narr_var = c("air.2m", "hpb1", "acpcp", "rhum.2m", "vis", "pres.sfc", "uwnd.10m",
    "vwnd.10m"),
  ...
)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
narr_var	character; name of NARR variable
...	passed to <code>stow::stow()</code> . The package and subdirs arguments are fixed by geomarker.

Details

Data is downloaded from the [NCEP North American Regional Reanalysis](#) project page and is available for the entire US from 1979 - near present. Data is updated monthly.

Value

a list of numeric vectors of NARR values

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_narr_data(s2cd_example_cincy(2L), "air.2m")
```

get_nei_point_summary *Summarize nearby National Emissions Inventory point emissions*

Description

Summarizes annual facility-level point emissions within buffer meters of each s2 cell as tons per year. Each date is mapped by calendar year to the nearest scheduled triennial National Emissions Inventory (NEI) cycle (2016-2018 to 2017 NEI, 2019-2021 to 2020 NEI, 2022-2024 to 2023 NEI). Dates mapping to an unavailable or unsupported cycle, including 2025 through 2027 mapping to the not-yet- available 2026 NEI, return NA_real_.

Usage

```
get_nei_point_summary(
  x,
  pollutant_code = c("PM25-PRI", "EC", "OC", "SO4", "NO3", "PMFINE"),
  fun = sum,
  buffer = 1000,
  ...
)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
pollutant_code	one of PM25-PRI, EC, OC, SO4, NO3, or PMFINE
fun	function used to summarize nearby annual emissions
buffer	distance from s2 cell (in meters) to summarize data
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Details

Facility-summary ZIP files for each requested pollutant are downloaded from the Environmental Protection Agency and read directly on each call. Rows are retained only when the pollutant code, latitude, longitude, and total emissions are complete. Latitude, longitude, and total emissions must also be finite, with latitude between -90 and 90 degrees and longitude between -180 and 180 degrees.

fun receives the numeric vector of annual emissions from facilities within the buffer and must return one numeric value. With the default sum, a location with no nearby facilities returns zero.

Value

a list of numeric vectors of point-source emissions summaries, one vector per input location and one value per associated date. Values passed to fun are in tons per year. Dates mapping to unavailable inventory cycles return NA_real_.

References

<https://www.epa.gov/air-emissions-inventories/national-emissions-inventory-nei>
<https://www.epa.gov/air-emissions-inventories/what-are-units-nei-emissions-data>

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_nei_point_summary(s2cd_example())
```

```
get_nei_point_summary(s2cd_example(), "PMFINE", buffer = 1600)
get_nei_point_summary(s2cd_example_cincy(5L), "PMFINE", buffer = 10000)
```

```
get_nlcd_fct_imp_summary
```

Summarize nearby imperviousness

Description

Summarizes fraction imperviousness (fractional surface area covered with artificial substrate or structures) within the buffer distance of each s2 cell; the year of each date is used to link to the corresponding annual snapshot.

Usage

```
get_nlcd_fct_imp_summary(x, fun = mean, buffer = 800, ...)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
fun	function to summarize extracted data
buffer	distance from s2 cell (in meters) to summarize data
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Details

Fractional Impervious Surface rasters from Annual NLCD Collection 1.2 are retrieved as full conterminous U.S. GeoTIFFs from the MRLC direct-download bundles at a 30 m grid. The currently available conterminous U.S. data span 1985 through 2025; dates outside the available years return NA_real_ values. See <https://www.usgs.gov/centers/eros/science/annual-nlcd-data-access> for data-access details and <https://www.usgs.gov/centers/eros/science/usgs-eros-archive-land-cover-annual> for product details.

Value

a list of numeric vectors of NLCD fraction imperviousness summaries. Dates in unavailable years return NA_real_.

Examples

```
get_nlcd_fct_imp_summary(s2cd_example_cincy(2L))
```

get_smoke_summary	Summarize wildfire smoke plume exposures
-------------------	--

Description

For each s2_cell location and Date vector, the intersections with NOAA’s HMS daily smoke polygons (see ?get_daily_smoke_data) are calculated and summarized as the maximum intensity ("Light", "Medium", "Heavy"). If no smoke polygons are intersected, "None" is used to summarize the maximum intensity.

Usage

```
get_smoke_summary(x, ...)
```

Arguments

- x a s2_cell_dates vector (see ?s2cd)
- ... passed to get_daily_smoke_data() and then stow::stow().

Value

a list of ordered factors (Levels: None > Light > Medium > Heavy)

Examples

```
withr::local_envvar(  
  R_USER_DATA_DIR = fs::path_package(  
    "geomarker",  
    "gmrkr--8841"  
  ),  
  R_GEOMARKER_NO_DOWNLOAD = "true"  
)  
s2cd(s2::as_s2_cell(c("8841b39a7c46e25f", "8841a45555555555")),  
  dates = list(as.Date(c("2024-05-18", "2024-11-06")),  
    as.Date(c("2024-06-22", "2024-08-15", "2024-12-30"))))  
) |>  
  get_smoke_summary()
```

get_tiger_bg	Census Block Group Linkage
--------------	----------------------------

Description

Link an object coercible to an s2_cell vector with the closest census block group using the US Census TIGER/Line shapefiles from a specific year.

Usage

```
get_tiger_bg(x, year = as.character(2024:2013), ...)
```

Arguments

x	object coercible to an s2_cell vector; non-missing cells must be full-resolution (level 30) cells
year	vintage of TIGER/Line block group geography files
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Value

An unnamed character vector of matched census block group identifiers, with one element per input cell in the same order. Duplicate and missing input cells retain their original positions.

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
set.seed(123)
get_tiger_bg(s2cd_example_cincy(8L))
```

get_traffic_summary	<i>Summarize nearby traffic</i>
---------------------	---------------------------------

Description

Summarizes average annual daily traffic meters (aadt_m) traveled on F1 and F2 roadways within buffer distance of each s2 cell as totals per vehicle class (trucks_buses, tractor_trailer, and passenger); dates are not used to link to traffic data.

Usage

```
get_traffic_summary(x, buffer = 400, ...)
```

Arguments

x	a s2_cell_dates vector (see ?s2cd)
buffer	distance from s2 cell (in meters) to summarize data
...	passed to <code>stow::stow()</code> . The package and subdir arguments are fixed by geomarker.

Details

Traffic data are derived from the final 2024 Highway Performance Monitoring System (HPMS) File Geodatabase published by the Bureau of Transportation Statistics. Only roads with F_SYSTEM classification of 1 ("interstate") or 2 ("principal arterial - other freeways and expressways") are used. Passenger vehicles (FHWA 1-3) are calculated as the total minus FHWA class 4-7 (single unit) and 8-13 (combination). FHWA documents substantial missing portions of the 2024 data for North Dakota and New Jersey. In addition, 55 retained roadway sections report single-unit plus combination AADT greater than total AADT; the documented passenger-vehicle subtraction is preserved for those source records rather than silently modifying them.

The processed traffic data are distributed separately from the R package as a GeoPackage asset attached to the geomarker package release. The file is saved as a durable managed local copy in the `get_traffic_summary` directory the first time it is required and reused on subsequent calls.

Value

data frame with one row per input location and numeric traffic summaries `aadtm_trucks_buses`, `aadtm_tractor_trailer`, and `aadtm_passenger`

References

<https://www.arcgis.com/home/item.html?id=5e6a977c2d7c4ec1bdc82e684d3384f2>

https://github.com/FHWA/HPMS/blob/2024-HPMS/README_2024_HPMS_All.md

Examples

```
withr::local_envvar(
  R_USER_DATA_DIR = fs::path_package(
    "geomarker",
    "gmrkr--8841"
  ),
  R_GEOMARKER_NO_DOWNLOAD = "true"
)
get_traffic_summary(s2cd_example_cincy(), buffer = 1500)
```

<code>impute_date_ranges</code>	<i>Impute date ranges from a chronological sequence of dates</i>
---------------------------------	--

Description

In observational data, the date associated with an address is often the date of the observation instead of the date of a change in address. For each interior observation, the imputed start date is the midpoint between that observation date and the previous observation date. The imputed end date is the next imputed start date. The first start date and last end date are optionally extended by `start_early` and `end_late`.

Usage

```
impute_date_ranges(x, start_early = 0L, end_late = 0L, expand = FALSE)
```

Arguments

<code>x</code>	a chronologically sorted Date vector
<code>start_early</code>	start the first imputed date range this many days early as a length-one integer
<code>end_late</code>	end the last imputed date range this many days late as a length-one integer
<code>expand</code>	logical; if TRUE, return a list of daily Date sequences, one for each input date, instead of the default start and end vectors.

Details

Use this function to impute an effective date range for a set of addresses or location identifiers that were collected on unrelated days. For example, residential addresses collected during a specific healthcare encounter do not reflect when a patient actually changed addresses. Imputing address date ranges can reduce reliance on observation dates alone and may help avoid differential assumptions about when address changes occurred.

Value

a list of start and end Date vectors for the imputed ranges for each input date in `x`, or, if `expand = TRUE`, a list of Date vectors where each element is the daily sequence from the imputed start date to the imputed end date.

Examples

```
impute_date_ranges(as.Date(c("2024-01-01", "2024-03-17", "2024-09-21")))

impute_date_ranges(as.Date(c("2024-01-01", "2024-03-17", "2024-09-21")),
  start_early = 30L, end_late = 60L
)

impute_date_ranges(
  as.Date(c("2024-01-01", "2024-03-17", "2024-09-21")),
  expand = TRUE
)

# use within a data.frame with multiple individuals
tibble::tibble(
  id = rep(c("A", "B"), each = 3),
  encounter = rep(1:3, 2),
  date = as.Date(c(
    "2024-01-01", "2024-03-17", "2024-09-21",
    "2023-11-29", "2024-09-22", "2024-09-29"
  ))
) |>
  dplyr::arrange(id, date) |>
  dplyr::mutate(
    imputed_start_date = impute_date_ranges(date)$start,
    imputed_end_date = impute_date_ranges(date)$end,
    .by = "id"
  )
```

```
# expand to daily dates for use with s2cd()
s2cd(
  s2::as_s2_cell(rep("8841b39a7c46e25f", 3)),
  dates = impute_date_ranges(
    as.Date(c("2024-01-01", "2024-03-17", "2024-09-21")),
    expand = TRUE
  )
)
```

is_s2cd	<i>Test whether input is a s2cd vector</i>
---------	--

Description

Tests if x inherits from class s2cd

Usage

is_s2cd(x)

Arguments

x any object to test

Value

logical

Examples

```
is_s2cd(s2cd_example())

is_s2cd(letters)
```

s2cd	<i>Create a new s2_cell_dates (s2cd) vector</i>
------	---

Description

A s2cd vector (short for **s2_cell_dates**) stores an s2::s2_cell vector alongside a list of date vectors, where each vector of chronologically, non-missing dates corresponds to each valid, level 30 s2 cell. Create a new s2cd vector with s2cd() or coerce another object into a s2cd vector with as_s2cd().

Usage

s2cd(.data, dates = list(), sort_dates = FALSE)

Arguments

<code>.data</code>	an <code>s2_cell</code> vector that is valid and is at level 30 resolution
<code>dates</code>	a list of date vectors, each in chronological order and without missing values
<code>sort_dates</code>	logical; sort each date vector chronologically when needed. This defaults to FALSE so accidentally unordered dates continue to error.

Details

Each position in the `s2_cells` vector corresponds to the same-indexed element in `dates`, allowing multiple dates to be associated with a single level-30 S2 cell.

At level 30, S2 cells have an approximate spatial resolution of *1 centimeter*; in this context, they are intended to function as point-like representations rather than true spatial areas.

Value

A `s2cd` vector

See Also

`as_s2cd`

Examples

```
# create directly with s2_cell vector and list of Date vectors
d <- s2cd(
  s2::as_s2_cell(c("8841b39a7c46e25f", "8841a45555555555")),
  dates = list(
    as.Date("2026-01-01"),
    c(as.Date("2024-09-13"), as.Date("2024-09-20"))
  )
)

# check if d is a s2cd vector
is_s2cd(d)

# use a s2cd vector as a tibble column
# the display includes each cell's number of dates
tibble::tibble(id = seq_along(d), location = d)

# create using data.frame with s2_cell and dates columns
data.frame(
  s2_cell = s2::as_s2_cell(c("8841b39a7c46e25f", "8841a45555555555")),
  dates = I(list(
    as.Date("2026-01-01"),
    c(as.Date("2024-09-13"), as.Date("2024-09-20"))
  ))
) |>
  as_s2cd()
```

s2cd_dates	<i>Get date vectors from a s2_cell_dates (s2cd) vector</i>
------------	--

Description

Get date vectors from a s2_cell_dates (s2cd) vector

Usage

```
s2cd_dates(x)
```

Arguments

x a s2_cell_dates vector (see ?s2cd)

Value

a list of Date vectors

Examples

```
s2cd_dates(s2cd_example())
```

s2cd_example	<i>Simple example s2cd vector</i>
--------------	-----------------------------------

Description

s2cd_example() returns a very simple example

Usage

```
s2cd_example()
```

Value

A s2cd vector

Examples

```
s2cd_example()  
  
str(s2cd_example())
```

s2cd_example_cincy	<i>More complex example s2cd vector</i>
--------------------	---

Description

s2cd_example_cincy() randomly generates an example s2cd vector by sampling locations from the center of level 15 cells covering the s2 cell "8841" and sampling dates from 2024.

Usage

```
s2cd_example_cincy(
  n_locations = 100L,
  n_dates_each = 3L,
  n_dates_variation = c("fixed", "poisson+1")
)
```

Arguments

n_locations	integer; number of s2 locations to simulate
n_dates_each	integer; number of dates to simulate per location
n_dates_variation	"poisson+1" randomly draws the number of dates per location from a Poisson distribution with lambda equal to n_dates; "fixed" randomly draws n_dates for every location

Examples

```
set.seed(923)
s2cd_example_cincy()
s2cd_example_cincy(100L, 5L, "poisson+1")
```

s2cd_within	<i>Test if a s2cd vector is within a date range or s2 cell union</i>
-------------	--

Description

Dates within the s2cd vector are compared to the date range, inclusively. An s2_cell vector or s2_cell_union object is used to test for intersection with the s2 cells. If both a date range and s2 cell union are provided, s2cd_within() will return TRUE only if both are within.

Usage

```
s2cd_within(x, date_range = NULL, s2_cell = NULL)
```

Arguments

x an s2_cell_dates vector
date_range length two Date vector representing a minimum and maximum allowable date
s2_cell a s2_cell vector, often a covering represented by many cells of different resolutions; see examples

Value

logical

Examples

```

# create compact representations of arbitrary polygons:
# s2::s2_covering_cell_ids_agg(
#   codec::cincy_city_geo(),
#   max_level = 12, max_cells = 100)

```

s2cd_years	<i>Get unique years from s2cd vector</i>
------------	--

Description

Get unique years from s2cd vector

Usage

```
s2cd_years(x)
```

Arguments

x a s2_cell_dates vector (see ?s2cd)

Value

a sorted character vector of years present in x

Examples

```
s2cd_years(s2cd_example())
```

Index

`as_s2cd`, [2](#)

`get_daily_smoke_data`, [3](#)
`get_elevation_summary`, [4](#)
`get_evi_summary`, [5](#)
`get_gridmet_data`, [6](#)
`get_merra_data`, [7](#)
`get_narr_data`, [8](#)
`get_nei_point_summary`, [9](#)
`get_nlcd_fct_imp_summary`, [11](#)
`get_smoke_summary`, [12](#)
`get_tiger_bg`, [12](#)
`get_traffic_summary`, [13](#)

`impute_date_ranges`, [14](#)
`install_merra_data` (`get_merra_data`), [7](#)
`is_s2cd`, [16](#)

`s2cd`, [16](#)
`s2cd_dates`, [18](#)
`s2cd_example`, [18](#)
`s2cd_example_cincy`, [19](#)
`s2cd_within`, [19](#)
`s2cd_years`, [20](#)
`stow::stow()`, [3](#), [4](#), [6](#), [7](#), [9–13](#)