Package: appc (via r-universe)

January 2, 2025

Title Air Pollution Predictor Commons

Version 0.5.0

URL https://github.com/geomarker-io/appc, http://geomarker.io/appc/

Description Functions for geomarker assessment for s2 locations and dates. These are used to train and predict daily ambient air pollution concentrations across the contiguous US 2016 - 2022.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports cli, dplyr, fs, glue, grf, httr, purrr, qs, rlang, s2, sf, terra, tibble, tidyr, withr

RoxygenNote 7.3.2

Suggests curl, dotenv, ggplot2, hexbin, knitr, lubridate, lwgeom, mappp, readr, rmarkdown, scales, testthat (>= 3.0.0), tidync, tigris, viridis

Config/testthat/edition 3

Config/testthat/parallel true

Depends R (>= 2.10)

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

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

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

RemoteRef v0.5.0

RemoteSha 16bb5d124c69679a3e8ca4a71a30d9af570cc1ef

Contents

`appc_clean_data_directory` *delete all installed data files in the user's data directory for the* `appc` *package*

Description

delete all installed data files in the user's data directory for the `appc` package

Usage

```
appc_clean_data_directory()
```

`assemble_predictors` *Assemble a tibble of required predictors for the exposure assessment model*

Description

Assemble a tibble of required predictors for the exposure assessment model

Usage

```
assemble_predictors(x, dates, pollutant = c("pm25"))
```

Arguments

`x`	a vector of s2 cell identifers (`s2_cell` object); currently required to be within the contiguous united states
`dates`	a list of date vectors for the predictions, must be the same length as `x`
`pollutant`	ignored now, but reserved for future sets of predictors specific to different pollutants

Value

a tibble with one row for each unique s2 location - date combination where columns are predictors required for the exposure assessment model

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2023-11-06")),
  "8841a45555555555" = as.Date(c("2023-06-22", "2023-08-15", "2024-09-30"))
)
assemble_predictors(x = s2::as_s2_cell(names(d)), dates = d) |>
  tibble::glimpse()
```

`get_closest_year` *Get the closest years to a vector of dates*

Description

The time between a date and year is calculated using July 1st of the year.

Usage

```
get_closest_year(x, years = as.character(1800:2400))
```

Arguments

`x`	a date vector
`years`	vector of characters (or numerics) representing years to choose from

Value

a character vector of the closest year in `years` for each date in `x`

Examples

```
get_closest_year(x = as.Date(c("2021-05-15", "2022-09-01")), years = c("2020", "2022"))
```

`get_daily_aqs` *Get daily AQS concentrations*

Description

Pre-generated daily summary files are downloaded from the EPA AQS website and filtered/harmonized as described in the Details.

Usage

```
get_daily_aqs(
  pollutant = c("pm25", "ozone", "no2"),
  year = as.character(2017:2024)
)
```

Arguments

`pollutant`	one of "pm25", "ozone", or "no2"
`year`	calendar year

Details

For PM2.5 (FRM, non-FRM, and speciation), data is filtered to only observations with a sample duration of "24 HOURS". All pollutants measurements are removed if the observation percent for the sampling period is less than 75. When a pollutant is measured by more than one device on the same day at the same s2 location, the average measurement is returned, ensuring unique measurements for each pollutant-location-day

Note: Historical measurements are subject to change and the EPA AQS website only stores the latest versions. Since this function always downloads the latest data from EPA AQS, that means that it will could different results depending on the date it was run.

Get all the files on the page and the date they were last updated: `readr::read_csv("https://aqs.epa.gov/aqsweb/airda`

Value

data.frame/tibble of pollutant concentrations with site id, lat/lon, and date

Examples

```
get_daily_aqs("pm25", "2024")
```

get_elevation_summary *Get elevation summary data*

Description

The fun (e.g. median() or sd()) of the elevations (captured at a spatial resolution of 800 by 800 m) within the buffer distance of each s2 geohash.

Usage

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

install_elevation_data()
```

Arguments

x	a vector of s2 cell identifers (s2_cell object)
fun	function to summarize extracted data
buffer	distance from s2 cell (in meters) to summarize data

Value

for get_elevation_summary(), a numeric vector of elevation summaries, the same length as x

for install_elevation_data(), a character string path to elevation raster

References

https://prism.oregonstate.edu/normals/

Examples

```
get_elevation_summary(s2::as_s2_cell(c("8841b399ced97c47", "8841b38578834123")))
```

get_gridmet_data *Get gridMET surface meteorological data*

Description

Daily, high spatial resolution (~4-km) data comes from the Climatology Lab and is available for the contiguous US from 1979-yesterday.

Usage

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

install_gridmet_data(
  gridmet_var = c("tmmx", "tmmn", "pr", "srad", "vs", "th", "rmax", "rmin", "sph"),
  gridmet_year = as.character(1979:format(Sys.Date(), "%Y"))
)
```

Arguments

`x`	a vector of s2 cell identifers (`s2_cell` object)
`dates`	a list of date vectors for the NARR data, must be the same length as `x`
`gridmet_var`	a character string that is the name of a gridMET variable
`gridmet_year`	a character string that is the year for the gridMET data; see details

Details

GRIDMET data comes as 1/24th degree gridded data, which is about 4 sq km resolution. s2 geohashes are intersected with this grid for matching with daily weather values.

gridMET variables are named:

```
gridmet_variable <- c(
  temperature_max = "tmmx",
  temperature_min = "tmmn",
  precipitation = "pr",
  solar_radiation = "srad",
  wind_speed = "vs",
  wind_direction = "th",
  relative_humidity_max = "rmax",
  relative_humidity_min = "rmin"
  specific_humidity = "sph"
)
```

Value

for `get_gridmet_data()`, a list of numeric vectors of gridMET values (the same length as `x` and `dates`)

for `install_gridmet_data()`, a character string path to gridMET raster data

References

https://www.climatologylab.org/gridmet.html

https://www.northwestknowledge.net/metdata/data/

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2024-05-18", "2024-11-06")),
  "8841a45555555555" = as.Date(c("2023-06-22", "2023-08-15"))
)
get_gridmet_data(x = s2::as_s2_cell(names(d)), dates = d, gridmet_var = "tmmx")
get_gridmet_data(x = s2::as_s2_cell(names(d)), dates = d, gridmet_var = "pr")
```

`get_hms_smoke_data` *Get smoke plume data from the NOAA's Hazard Mapping System*

Description

The HMS operates daily in near real-time by outlining the smoke polygon of each distinct smoke plume and classifying it as "light", "medium", and "heavy". Since multiple plumes of varying or the same classification can cover one another, the total smoke plume exposure is estimated as the weighted sum of all plumes, where "light" = 1, "medium" = 2, and "heavy" = 3.

Usage

```
get_hms_smoke_data(x, dates)

install_hms_smoke_data(
  hms_smoke_start_date = as.Date("2017-01-01"),
  hms_smoke_end_date = as.Date("2024-12-31")
)
```

Arguments

`x`	a vector of s2 cell identifers (`s2_cell` object); currently required to be within the contiguous united states
`dates`	a list of date vectors for the predictions, must be the same length as `x`
`hms_smoke_start_date`	a date object that is the first day of hms smoke data installed
`hms_smoke_end_date`	a date object that is the last day of hms smoke data installed

Details

Daily HMS shapefiles are missing for 7 days within 2017-2023 ("2017-04-27", "2017-05-31", "2017-06-01", "2017-06-01" "2017-06-22", "2017-11-12", "2018-12-31") and will return zero values. If files are available but no smoke plumes intersect, then a zero values is also returned.

Value

for `get_hms_smoke_data()`, a list of numeric vectors of smoke plume scores (the same length as `x` and `dates`)

for `install_hms_smoke_data()`, a character string path to the installed RDS file

References

https://www.ospo.noaa.gov/Products/land/hms.html#about

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2017-11-06")),
  "8841a45555555555" = as.Date(c("2017-06-22", "2023-08-15", "2024-12-30"))
)
get_hms_smoke_data(x = s2::as_s2_cell(names(d)), dates = d)
```

get_merra_data *Get MERRA-2 aerosol diagnostics data*

Description

Total and component (Dust, OC, BC, SS, SO4) surface PM2.5 concentrations from the MERRA-2 M2T1NXAER v5.12.4 product. Because installing MERRA-2 data takes a long time, "pre-compiled" data binaries for each year are available as pre-releases specific to MERRA data on GitHub.

Usage

```
get_merra_data(x, dates, merra_release = "merra-2025-01-02")

install_merra_data(
  merra_year = as.character(2017:2024),
  merra_release = "merra-2025-01-02"
)

create_daily_merra_data(merra_date)
```

Arguments

`x` a vector of s2 cell identifers (`s2_cell` object)

`dates` a list of date vectors for the MERRA data, must be the same length as `x`

`merra_release` a character string of a release tag from which "pre-compiled" MERRA data binary is used instead of installing latest data from source; see details

`merra_year` a character string that is the year for the merra data

`merra_date` a date object that is the date for the merra data

Details

- Installed data are filtered to a bounding box around the contiguous US, averaged to daily values, and converted to micrograms per cubic meter (ug/m^3).
- Total surface PM2.5 mass is calculated according to the formula in `https://gmao.gsfc.nasa.gov/reanalysis/MERRA-2/FAQ/#Q4`
- Set options("appc_install_data_from_source"), or the environment variable `APPC_INSTALL_DATA_FROM_SOURCE` to any non-empty value to install MERRA-2 directly from their sources instead of using the released GitHub data binary.
  - An Earthdata account linked with permissions for GES DISC is required. The `EARTHDATA_USERNAME` and `EARTHDATA_PASSWORD` must be set. If a `.env` file is present, environment variables will be loaded using the dotenv package.
  - Set a proxy to be used by all httr calls in the merra functions with `httr::set_config(httr::use_proxy( ... ))`

Value

for `get_merra_data()`, a list of tibbles the same length as `x`, each containing merra data columns (`merra_dust`, `merra_oc`, `merra_bc`, `merra_ss`, `merra_so4`, `merra_pm25`) with one row per date in `dates`

for `install_merra_data()`, a character string path to the merra data

for `create_daily_merra_data()`, a tibble with columns for s2, date, and concentrations of PM2.5 total, dust, oc, bc, ss, so4

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2023-11-06")),
  "8841a45555555555" = as.Date(c("2023-06-22", "2023-08-15"))
)
get_merra_data(x = s2::as_s2_cell(names(d)), dates = d)
```

`get_narr_data` *Get daily North American Regional Reanalysis (NARR) weather data*

Description

Get daily North American Regional Reanalysis (NARR) weather data

Installs NARR raster data into user's data directory for the appc package

Installs annual NLCD Fractional Impervious Surface raster data into user's data directory for the appc package

Usage

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

install_narr_data(
  narr_var = c("air.2m", "hpbl", "acpcp", "rhum.2m", "vis", "pres.sfc", "uwnd.10m",
    "vwnd.10m"),
  narr_year = as.character(2016:2024)
)

install_nlcd_frac_imperv_data(year = as.character(2024:2017))
```

Arguments

`x` a vector of s2 cell identifers (`s2_cell` object)

`dates` a list of date vectors for the NARR data, must be the same length as `x`

`narr_var` a character string that is the name of a NARR variable

`narr_year` a character string that is the year for the NARR data

Details

NARR data comes as 0.3 degrees gridded data, which is about 32 sq km resolution. s2 geohashes are intersected with this 0.3 degree grid for matching with daily weather values.

Value

for `get_narr_data()`, a list of numeric vectors of NARR values (the same length as `x` and `dates`)

for `install_narr_data()`, a character string path to NARR raster data

for `install_nlcd_frac_imperv_data()`, a character string path to NLCD raster data

References

https://psl.noaa.gov/data/gridded/data.narr.html

https://www.ncei.noaa.gov/products/weather-climate-models/north-american-regional

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2023-11-06")),
  "8841a45555555555" = as.Date(c("2023-06-22", "2023-08-15"))
)
get_narr_data(x = s2::as_s2_cell(names(d)), dates = d, narr_var = "air.2m")
```

get_nlcd_frac_imperv *Get NLCD Fractional Impervious Surface*

Description

Get NLCD Fractional Impervious Surface

Usage

```
get_nlcd_frac_imperv(x, dates, fun = stats::median, buffer = 400)
```

Arguments

x	a vector of s2 cell identifers (s2_cell object)
dates	a list of date vectors for the NLCD data, must be the same length as x; each date is matched to the closest available year of Annual NLCD data
fun	function to summarize extracted data
buffer	distance from s2 cell (in meters) to summarize data

Value

for get_nlcd_frac_imperv(), a list of numeric vectors of fractional impervious surface pixel summaries, the same length as x; each vector has values for each date in dates, named according to the NLCD product year

References

https://www.usgs.gov/centers/eros/science/annual-nlcd-fractional-impervious-surface

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2023-11-06")),
  "8841a45555555555" = as.Date(c("2022-06-22", "2022-08-15"))
)
get_nlcd_frac_imperv(x = s2::as_s2_cell(names(d)), dates = d)
get_nlcd_frac_imperv(x = s2::as_s2_cell(names(d)), dates = d, fun = mean, buffer = 1000)
```

`predict_pm25` *Get daily PM2.5 model predictions*

Description

Get daily PM2.5 model predictions

Usage

```
predict_pm25(x, dates)
```

Arguments

`x` a vector of s2 cell identifers (`s2_cell` object)

`dates` a list of date vectors for the predictions, must be the same length as `x`

Value

a list of tibbles the same length as `x`, each containing columns for the predicted (`pm25`) and its standard error (`pm25_se`); with one row per date in `dates`. These numerics are the concentrations of fine particulate matter, measured in micrograms per cubic meter. See `vignette("cv-model-performance")` for more details on the cross validated accuracy of the daily PM2.5 model predictions.

Examples

```
d <- list(
  "8841b39a7c46e25f" = as.Date(c("2023-05-18", "2023-11-06")),
  "8841a45555555555" = as.Date(c("2023-06-22", "2023-08-15"))
)
predict_pm25(x = s2::as_s2_cell(names(d)), dates = d)
```

Index