Package 'FedData'

March 10, 2023

Type Package

Title Functions to Automate Downloading Geospatial Data Available from Several Federated Data Sources

Version 3.0.3

Date 2023-03-10

Description Functions to automate downloading geospatial data available from several federated data sources (mainly sources maintained by the US Federal government). Currently, the package enables extraction from seven datasets: The National Elevation Dataset digital elevation models (1 and 1/3 arc-second; USGS); The National Hydrography Dataset (USGS); The Soil Survey Geographic (SSURGO) database from the National Cooperative Soil Survey (NCSS), which is led by the Natural Resources Conservation Service (NRCS) under the USDA; the Global Historical Climatology Network (GHCN), coordinated by National Climatic Data Center at NOAA; the Daymet gridded estimates of daily weather parameters for North America, version 3, available from the Oak Ridge National Laboratory's Distributed Active Archive Center (DAAC); the International Tree Ring Data Bank; and the National Land Cover Database (NLCD).

License MIT + file LICENSE

URL https://docs.ropensci.org/FedData/, https://github.com/ropensci/FedData

BugReports https://github.com/ropensci/FedData/issues

SystemRequirements GDAL (>= 3.0.0)

Depends R (>= 3.2.0)

Imports curl, httr, data.table, dplyr, tibble, tidyr, stringr, igraph, jsonlite, xml2, lifecycle, lubridate, magrittr, methods, progress, purrr, readr, terra (>= 1.0), raster (>= 3.6.3), sf (>= 1.0), sp

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

RoxygenNote 7.2.3

Suggests covr, ggplot2, testthat, ncdf4, usethis

Author R. Kyle Bocinsky [aut, cre, cph],
Dylan Beaudette [ctb],
Scott Chamberlain [ctb, rev],
Jeffrey Hollister [ctb],
Julia Gustavsen [rev]

Maintainer R. Kyle Bocinsky <bocinsky@gmail.com>

Date/Publication 2023-03-10 22:40:10 UTC

R topics documented:

`get_daymet` *Download and crop the 1-km DAYMET v4 daily weather dataset.*

Description

`get_daymet` returns a `RasterBrick` of weather data cropped to a given template study area.

Usage

```
get_daymet(
  template,
  label,
  elements = c("dayl", "prcp", "srad", "swe", "tmax", "tmin", "vp"),
  years = 1980:(lubridate::year(Sys.time()) - 1),
  region = "na",
  tempo = "day",
```

```
  extraction.dir = paste0(tempdir(), "/FedData/extractions/daymet/", label, "/"),
  raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9", "INTERLEAVE=BAND"),
  force.redo = FALSE,
  progress = TRUE
)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping.
label	A character string naming the study area.
elements	A character vector of elements to extract. The available elements are: dayl = Duration of the daylight period in seconds per day. This calculation is based on the period of the day during which the sun is above a hypothetical flat horizon. prcp = Daily total precipitation in millimeters per day, sum of all forms converted to water-equivalent. Precipitation occurrence on any given day may be ascertained. srad = Incident shortwave radiation flux density in watts per square meter, taken as an average over the daylight period of the day. NOTE: Daily total radiation (MJ/m2/day) can be calculated as follows: ((srad (W/m2) * dayl (s/day)) / l,000,000) swe = Snow water equivalent in kilograms per square meter. The amount of water contained within the snowpack. tmax = Daily maximum 2-meter air temperature in degrees Celsius. tmin = Daily minimum 2-meter air temperature in degrees Celsius. vp = Water vapor pressure in pascals. Daily average partial pressure of water vapor.
years	A numeric vector of years to extract.
region	The name of a region. The available regions are: na = North America hi = Hawaii pr = Puerto Rico
tempo	The frequency of the data. The available tempos are: day = Daily data mon = Monthly summary data ann = Annual summary data
extraction.dir	A character string indicating where the extracted and cropped DEM should be put. Defaults to a temporary directory.
raster.options	a vector of options for raster::writeRaster.
force.redo	If an extraction for this template and label already exists in extraction.dir, should a new one be created?
progress	Draw a progress bar when downloading?

Value

A named list of RasterBricks of weather data cropped to the extent of the template.

Examples

```
## Not run:

# Get the DAYMET (North America only)
# Returns a list of raster bricks
DAYMET <- get_daymet(
  template = FedData::meve,
  label = "meve",
  elements = c("prcp", "tmin", "tmax"),
  years = 1980:1985
)

# Plot with raster::plot
plot(DAYMET$tmin$X1985.10.23)

## End(Not run)
```

---

get_ghcn_daily *Download and crop the Global Historical Climate Network-Daily data.*

Description

get_ghcn_daily returns a named list of length 2:

1. 'spatial': A SpatialPointsDataFrame of the locations of GHCN weather stations in the template, and
2. 'tabular': A named list of data.frames with the daily weather data for each station. The name of each list item is the station ID.

Usage

```
get_ghcn_daily(
  template = NULL,
  label = NULL,
  elements = NULL,
  years = NULL,
  raw.dir = paste0(tempdir(), "/FedData/raw/ghcn"),
  extraction.dir = paste0(tempdir(), "/FedData/extractions/ghcn/", label, "/"),
  standardize = F,
  force.redo = F
)
```

Arguments

`template` A Raster* or Spatial* object to serve as a template for cropping. Alternatively, a character vector providing GHCN station IDs. If missing, all stations will be downloaded!

`label` A character string naming the study area.

`elements` A character vector of elements to extract.
The five core elements are:
PRCP = Precipitation (tenths of mm)
SNOW = Snowfall (mm)
SNWD = Snow depth (mm)
TMAX = Maximum temperature (tenths of degrees C)
TMIN = Minimum temperature (tenths of degrees C)

The other elements are:

ACMC = Average cloudiness midnight to midnight from 30-second ceilometer data (percent)
ACMH = Average cloudiness midnight to midnight from manual observations (percent)
ACSC = Average cloudiness sunrise to sunset from 30-second ceilometer data (percent)
ACSH = Average cloudiness sunrise to sunset from manual observations (percent)
AWDR = Average daily wind direction (degrees)
AWND = Average daily wind speed (tenths of meters per second)
DAEV = Number of days included in the multiday evaporation total (MDEV)
DAPR = Number of days included in the multiday precipitation total (MDPR)
DASF = Number of days included in the multiday snowfall total (MDSF)
DATN = Number of days included in the multiday minimum temperature (MDTN)
DATX = Number of days included in the multiday maximum temperature (MDTX)
DAWM = Number of days included in the multiday wind movement (MDWM)
DWPR = Number of days with non-zero precipitation included in multiday precipitation total (MDPR)
EVAP = Evaporation of water from evaporation pan (tenths of mm)
FMTM = Time of fastest mile or fastest 1-minute wind (hours and minutes, i.e., HHMM)
FRGB = Base of frozen ground layer (cm)
FRGT = Top of frozen ground layer (cm)
FRTH = Thickness of frozen ground layer (cm)
GAHT = Difference between river and gauge height (cm)
MDEV = Multiday evaporation total (tenths of mm; use with DAEV)
MDPR = Multiday precipitation total (tenths of mm; use with DAPR and DWPR, if available)
MDSF = Multiday snowfall total
MDTN = Multiday minimum temperature (tenths of degrees C; use with DATN)
MDTX = Multiday maximum temperature (tenths of degrees C; use with DATX)
MDWM = Multiday wind movement (km)

MNPN = Daily minimum temperature of water in an evaporation pan (tenths of degrees C)
MXPN = Daily maximum temperature of water in an evaporation pan (tenths of degrees C)
PGTM = Peak gust time (hours and minutes, i.e., HHMM)
PSUN = Daily percent of possible sunshine (percent)
SN*# = Minimum soil temperature (tenths of degrees C) where * corresponds to a code for ground cover and # corresponds to a code for soil depth.

Ground cover codes include the following:
0 = unknown
1 = grass
2 = fallow
3 = bare ground
4 = brome grass
5 = sod
6 = straw multch
7 = grass muck
8 = bare muck

Depth codes include the following:
1 = 5 cm
2 = 10 cm
3 = 20 cm
4 = 50 cm
5 = 100 cm
6 = 150 cm
7 = 180 cm

SX*# = Maximum soil temperature (tenths of degrees C) where * corresponds to a code for ground cover and # corresponds to a code for soil depth.
See SN*# for ground cover and depth codes.
TAVG = Average temperature (tenths of degrees C) [Note that TAVG from source 'S' corresponds to an average for the period ending at 2400 UTC rather than local midnight]
THIC = Thickness of ice on water (tenths of mm)
TOBS = Temperature at the time of observation (tenths of degrees C)
TSUN = Daily total sunshine (minutes)
WDF1 = Direction of fastest 1-minute wind (degrees)
WDF2 = Direction of fastest 2-minute wind (degrees)
WDF5 = Direction of fastest 5-second wind (degrees)
WDFG = Direction of peak wind gust (degrees)
WDFI = Direction of highest instantaneous wind (degrees)
WDFM = Fastest mile wind direction (degrees)
WDMV = 24-hour wind movement (km)
WESD = Water equivalent of snow on the ground (tenths of mm)
WESF = Water equivalent of snowfall (tenths of mm)
WSF1 = Fastest 1-minute wind speed (tenths of meters per second)

WSF2 = Fastest 2-minute wind speed (tenths of meters per second)
WSF5 = Fastest 5-second wind speed (tenths of meters per second)
WSFG = Peak gust wind speed (tenths of meters per second)
WSFI = Highest instantaneous wind speed (tenths of meters per second)
WSFM = Fastest mile wind speed (tenths of meters per second)
WT** = Weather Type where ** has one of the following values:

01 = Fog, ice fog, or freezing fog (may include heavy fog)
02 = Heavy fog or heaving freezing fog (not always distinguished from fog)
03 = Thunder
04 = Ice pellets, sleet, snow pellets, or small hail
05 = Hail (may include small hail)
06 = Glaze or rime
07 = Dust, volcanic ash, blowing dust, blowing sand, or blowing obstruction
08 = Smoke or haze
09 = Blowing or drifting snow
10 = Tornado, waterspout, or funnel cloud
11 = High or damaging winds
12 = Blowing spray
13 = Mist
14 = Drizzle
15 = Freezing drizzle
16 = Rain (may include freezing rain, drizzle, and freezing drizzle)
17 = Freezing rain
18 = Snow, snow pellets, snow grains, or ice crystals
19 = Unknown source of precipitation
21 = Ground fog
22 = Ice fog or freezing fog

WV** = Weather in the Vicinity where ** has one of the following values:
01 = Fog, ice fog, or freezing fog (may include heavy fog)
03 = Thunder
07 = Ash, dust, sand, or other blowing obstruction
18 = Snow or ice crystals
20 = Rain or snow shower

`years` A numeric vector indicating which years to get.

`raw.dir` A character string indicating where raw downloaded files should be put. The directory will be created if missing. Defaults to './RAW/GHCN/'.

`extraction.dir` A character string indicating where the extracted and cropped GHCN shapefiles should be put. The directory will be created if missing. Defaults to './EXTRACTIONS/GHCN/'.

`standardize` Select only common year/month/day? Defaults to FALSE.

`force.redo` If an extraction for this template and label already exists, should a new one be created? Defaults to FALSE.

Value

A named list containing the 'spatial' and 'tabular' data.

Examples

```
## Not run:
# Get the daily GHCN data (GLOBAL)
# Returns a list: the first element is the spatial locations of stations,
# and the second is a list of the stations and their daily data
GHCN.prcp <-
  get_ghcn_daily(
    template = FedData::meve,
    label = "meve",
    elements = c("prcp")
  )

# Plot the VEP polygon
plot(meve$geometry)

# Plot the spatial locations
plot(GHCN.prcp$spatial, pch = 1, add = T)
legend("bottomleft", pch = 1, legend = "GHCN Precipitation Records")

# Elements for which you require the same data
# (i.e., minimum and maximum temperature for the same days)
# can be standardized using standardize==T
GHCN.temp <- get_ghcn_daily(
  template = FedData::meve,
  label = "meve",
  elements = c("tmin", "tmax"),
  standardize = T
)

# Plot the VEP polygon
plot(meve$geometry)

# Plot the spatial locations
plot(GHCN.temp$spatial, pch = 1, add = T)
legend("bottomleft", pch = 1, legend = "GHCN Temperature Records")

## End(Not run)
```

`get_itrdb` *Download the latest version of the ITRDB, and extract given parameters.*

Description

`get_itrdb` returns a named list of length 3:

1. 'metadata': A data.table or `SpatialPointsDataFrame` (if `makeSpatial==TRUE`) of the locations and names of extracted ITRDB chronologies,
2. 'widths': A matrix of tree-ring widths/densities given user selection, and
3. 'depths': A matrix of tree-ring sample depths.

Usage

```
get_itrdb(
  template = NULL,
  label = NULL,
  recon.years = NULL,
  calib.years = NULL,
  species = NULL,
  measurement.type = NULL,
  chronology.type = NULL,
  raw.dir = paste0(tempdir(), "/FedData/raw/itrdb"),
  extraction.dir = ifelse(!is.null(label), paste0(tempdir(),
    "/FedData/extractions/itrdb/", label, "/"), paste0(tempdir(),
    "/FedData/extractions/itrdb")),
  force.redo = FALSE
)
```

Arguments

`template`	A Raster* or Spatial* object to serve as a template for selecting chronologies. If missing, all available global chronologies are returned.
`label`	A character string naming the study area.
`recon.years`	A numeric vector of years over which reconstructions are needed; if missing, the union of all years in the available chronologies are given.
`calib.years`	A numeric vector of all required years—chronologies without these years will be discarded; if missing, all available chronologies are given.
`species`	A character vector of 4-letter tree species identifiers; if missing, all available chronologies are given.
`measurement.type`	A character vector of measurement type identifiers. Options include: • 'Total Ring Density' • 'Earlywood Width' • 'Earlywood Density' • 'Latewood Width' • 'Minimum Density' • 'Ring Width' • 'Latewood Density' • 'Maximum Density' • 'Latewood Percent' if missing, all available chronologies are given.
`chronology.type`	A character vector of chronology type identifiers. Options include: • 'ARSTND' • 'Low Pass Filter' • 'Residual'

- 'Standard'
- 'Re-Whitened Residual'
- 'Measurements Only'

if missing, all available chronologies are given.

`raw.dir` A character string indicating where raw downloaded files should be put. The directory will be created if missing.

`extraction.dir` A character string indicating where the extracted and cropped ITRDB dataset should be put. The directory will be created if missing.

`force.redo` If an extraction already exists, should a new one be created? Defaults to FALSE.

Value

A named list containing the 'metadata', 'widths', and 'depths' data.

Examples

```
## Not run:
# Get the ITRDB records
ITRDB <- get_itrdb(template = FedData::meve, label = "meve", makeSpatial = T)

# Plot the VEP polygon
plot(meve$geometry)

# Map the locations of the tree ring chronologies
plot(ITRDB$metadata, pch = 1, add = T)
legend("bottomleft", pch = 1, legend = "ITRDB chronologies")

## End(Not run)
```

`get_nass_cdl` *Download and crop the NASS Cropland Data Layer.*

Description

`get_nass_cdl` returns a `RasterLayer` of NASS Cropland Data Layer cropped to a given template study area.

Usage

```
get_nass_cdl(
  template,
  label,
  year = 2019,
  extraction.dir = paste0(tempdir(), "/FedData/"),
  raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9", "INTERLEAVE=BAND"),
  force.redo = FALSE,
  progress = TRUE
```

```
)

get_nass(template, label, ...)

get_cdl(template, label, ...)

cdl_colors()
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping.
label	A character string naming the study area.
year	An integer representing the year of desired NASS Cropland Data Layer product. Acceptable values are 2007–the last year.
extraction.dir	A character string indicating where the extracted and cropped NASS data should be put. The directory will be created if missing.
raster.options	a vector of options for terra::writeRaster.
force.redo	If an extraction for this template and label already exists, should a new one be created?
progress	Draw a progress bar when downloading?
...	Other parameters passed on to [get_nass_cdl].

Value

A RasterLayer cropped to the bounding box of the template.

Examples

```
## Not run:
# Extract data for the Mesa Verde National Park:

# Get the NASS CDL (USA ONLY)
# Returns a raster
NASS <-
  get_nass_cdl(
    template = FedData::meve,
    label = "meve",
    year = 2011
  )

# Plot with raster::plot
plot(NASS)

## End(Not run)
```

get_ned *Download and crop the 1 (~30 meter) or 1/3 (~10 meter) arc-second National Elevation Dataset.*

Description

get_ned returns a RasterLayer of elevation data cropped to a given template study area.

Usage

```
get_ned(
  template,
  label,
  res = "1",
  extraction.dir = file.path(tempdir(), "FedData", "extractions", "ned", label),
  raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9"),
  force.redo = FALSE
)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping.
label	A character string naming the study area.
res	A character string representing the desired resolution of the NED. '1' indicates the 1 arc-second NED (the default), while '13' indicates the 1/3 arc-second dataset.
extraction.dir	A character string indicating where the extracted and cropped DEM should be put. The directory will be created if missing.
raster.options	a vector of options for terra::writeRaster.
force.redo	If an extraction for this template and label already exists, should a new one be created?

Value

A RasterLayer DEM cropped to the extent of the template.

Examples

```
## Not run:
# Get the NED (USA ONLY)
# Returns a raster
NED <- get_ned(template = FedData::meve, label = "meve")

# Plot with raster::plot
plot(NED)

## End(Not run)
```

get_nhd *Download and crop the National Hydrography Dataset.*

Description

get_nhd returns a list of 'sf' objects extracted from the National Hydrography Dataset.

Usage

```
get_nhd(
  template,
  label,
  nhdplus = FALSE,
  extraction.dir = file.path(tempdir(), "FedData", "extractions", "nhd", label),
  force.redo = FALSE
)
```

Arguments

template	A 'Raster*', 'Spatial*', or 'sf' object to serve as a template for cropping.
label	A character string naming the study area.
nhdplus	Extract data from the USGS NHDPlus High Resolution service (experimental)
extraction.dir	A character string indicating where the extracted and cropped NHD data should be put.
force.redo	If an extraction for this template and label already exists, should a new one be created?

Value

A list of 'sf' collections extracted from the National Hydrography Dataset.

Examples

```
## Not run:
# Get the NHD (USA ONLY)
NHD <- get_nhd(
  template = FedData::meve,
  label = "meve"
)
NHD
NHD %>%
  plot_nhd(template = FedData::meve)

## End(Not run)
```

get_nlcd *Download and crop the National Land Cover Database.*

Description

get_nlcd returns a RasterLayer of NLCD data cropped to a given template study area. nlcd_colors and pal_nlcd return the NLCD legend and color palette, as available through the [MLRC website](https://www.mrlc.gov/data land-cover-database-2016-nlcd2016-legend).

Usage

```
get_nlcd(
  template,
  label,
  year = 2019,
  dataset = c("landcover", "impervious", "canopy"),
  landmass = "L48",
  extraction.dir = paste0(tempdir(), "/FedData/"),
  raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9"),
  force.redo = FALSE
)

nlcd_colors()

pal_nlcd()
```

Arguments

template	A sf, Raster* or Spatial* object to serve as a template for cropping.
label	A character string naming the study area.
year	An integer representing the year of desired NLCD product. Acceptable values are 2019 (default), 2016, 2011, 2008, 2006, 2004, and 2001.
dataset	A character string representing type of the NLCD product. Acceptable values are 'landcover' (default), 'impervious', and 'canopy' (2016 and 2011, L48 only).
landmass	A character string representing the landmass to be extracted Acceptable values are 'L48' (lower 48 US states, the default), 'AK' (Alaska, 2011 and 2016 only), 'HI' (Hawaii, 2001 only), and 'PR' (Puerto Rico, 2001 only).
extraction.dir	A character string indicating where the extracted and cropped NLCD data should be put. The directory will be created if missing.
raster.options	a vector of options for terra::writeRaster.
force.redo	If an extraction for this template and label already exists, should a new one be created?

Value

A RasterLayer cropped to the bounding box of the template.

Examples

```
## Not run:
# Extract data for the Mesa Verde National Park:

# Get the NLCD (USA ONLY)
# Returns a raster
NLCD <-
  get_nlcd(
    template = FedData::meve,
    label = "meve",
    year = 2016
  )

# Plot with raster::plot
plot(NLCD)

## End(Not run)
```

---

get_ssurgo *Download and crop data from the NRCS SSURGO soils database.*

Description

This is an efficient method for spatially merging several different soil survey areas as well as merging their tabular data.

Usage

```
get_ssurgo(
  template,
  label,
  raw.dir = paste0(tempdir(), "/FedData/raw/ssurgo"),
  extraction.dir = paste0(tempdir(), "/FedData/"),
  force.redo = FALSE
)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping; optionally, a vector of area names [e.g., c('IN087','IN088')] may be provided.
label	A character string naming the study area.
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing. Defaults to './RAW/SSURGO/'.

`extraction.dir` A character string indicating where the extracted and cropped SSURGO shapefiles should be put. The directory will be created if missing. Defaults to './EXTRACTIONS/SSURGO/'.

`force.redo` If an extraction for this template and label already exists, should a new one be created? Defaults to FALSE.

Details

`get_ssurgo` returns a named list of length 2:

1. 'spatial': A `SpatialPolygonsDataFrame` of soil mapunits in the template, and
2. 'tabular': A named list of `data.frame`s with the SSURGO tabular data.

Value

A named list containing the 'spatial' and 'tabular' data.

Examples

```
## Not run:
# Get the NRCS SSURGO data (USA ONLY)
SSURGO.MEVE <- get_ssurgo(template = FedData::meve, label = "meve")

# Plot the VEP polygon
plot(meve$geometry)

# Plot the SSURGO mapunit polygons
plot(SSURGO.MEVE$spatial, lwd = 0.1, add = T)

# Or, download by Soil Survey Area names
SSURGO.areas <- get_ssurgo(template = c("CO670", "CO075"), label = "CO_TEST")

# Let's just look at spatial data for CO675
SSURGO.areas.CO675 <- SSURGO.areas$spatial[SSURGO.areas$spatial$AREASYMBOL == "CO075", ]

# And get the NED data under them for pretty plotting
NED.CO675 <- get_ned(template = SSURGO.areas.CO675, label = "SSURGO_CO675")

# Plot the SSURGO mapunit polygons, but only for CO675
plot(NED.CO675)
plot(SSURGO.areas.CO675, lwd = 0.1, add = T)

## End(Not run)
```

get_wbd *Download and crop the Watershed Boundary Dataset.*

Description

get_wbd returns an 'sf' collection of the HUC 12 regions within the specified template.

Usage

```
get_wbd(
  template,
  label,
  extraction.dir = file.path(tempdir(), "FedData", "extractions", "nhd", label),
  force.redo = FALSE
)
```

Arguments

template	A 'Raster*', 'Spatial*', or 'sf' object to serve as a template for cropping.
label	A character string naming the study area.
extraction.dir	A character string indicating where the extracted and cropped NHD data should be put.
force.redo	If an extraction for this template and label already exists, should a new one be created?

Value

A 'sf' collection of the HUC 12 regions within the specified template.

meve *The boundary of Mesa Verde National Park*

Description

A dataset containing the spatial polygon defining the boundary of Mesa Verde National Park in Montana.

Usage

```
meve
```

Format

Simple feature collection with 1 feature and a geometry field.

plot_nhd *A basic plotting function for NHD data.*

Description

This is more of an example than anything

Usage

```
plot_nhd(x, template = NULL)
```

Arguments

x	The result of [get_nhd].
template	template A 'Raster*', 'Spatial*', or 'sf' object to serve as a template for cropping.

Value

A [ggplot2] panel of plots

Examples

```
## Not run:
# Get the NHD (USA ONLY)
NHD <- get_nhd(
  template = FedData::meve,
  label = "meve"
)
NHD
NHD %>%
  plot_nhd(template = FedData::meve)

## End(Not run)
```

replace_null *Replace NULLs*

Description

Replace all the empty values in a list

Usage

```
replace_null(x)
```

Arguments

x	A list

Examples

```
list(a = NULL, b = 1, c = list(foo = NULL, bar = NULL)) %>% replace_null()
```

Index