Package ‘FedData’

March 11, 2016

Type Package

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

Version 2.0.8

Date 2016-03-10

Author R. Kyle Bocinsky [aut, cre], Dylan Beaudette [ctb], Scott Chamberlain [ctb]

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

Description Functions to automate downloading geospatial data available from several federated data sources (mainly sources maintained by the US Federal government). Currently, the package allows for retrieval of five 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; and the International Tree Ring Data Bank. Additional data sources are in the works, including global DEM resources (ETOPO1, ETOPO5, ETOPO30, SRTM), global soils (HWSD), MODIS satellite data products, the National Atlas (US), Natural Earth, PRISM, and WorldClim.

License MIT + file LICENSE

Depends R (>= 3.2.0), sp

Imports data.table, devtools, soilDB, igraph, curl, methods, rgdal (>= 1.1), raster, Hmisc

Repository CRAN

NeedsCompilation no

RoxygenNote 5.0.1

Date/Publication 2016-03-11 00:57:44

R topics documented:

FedData-package *Scripts to automate downloading geospatial data available from the several federated data sources*

Description

This package contains scripts to automate downloading geospatial data available from the several federated data sources (mainly sources maintained by the US Federal government). Currently, the package allows for retrieval of five 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, and
- The Global Historical Climatology Network (GHCN), coordinated by National Climatic Data Center at NOAA.
- The International Tree Ring Data Bank (ITRDB), coordinated by National Climatic Data Center at NOAA.

Additional data sources are in the works, including global DEM resources (ETOPO1, ETOPO5, ETOPO30, SRTM), global soils (HWSD), MODIS satellite data products, the National Atlas (US), Natural Earth, PRISM, and WorldClim.

Details

Package:	FedData
Type:	FedData
Version:	2.0.8
Date:	2016-03-10
License:	MIT

This package is designed with the large-scale GIS use-case in mind: cases where the use of dynamic web-services is impractical due to the scale (spatial and/or temporal) of analysis. It functions primarily as a means of downloading tiled or otherwise spaticially-defined datasets; additionally, it can preprocess those datasets by extracting data within an area of interest (AoI), defined spatially. It relies heavily on the **sp**, **raster**, and **rgdal** packages.

There are three general types of methods available for each dataset (and others for particular datasets):

get...:	High-level function that allows the user to define an AoI ("template") and returns the dataset cropped/masked t
extract...:	Mid-level functions that automate extraction of tabular data from databases (such as the SSURGO soils tabular
download...:	Low-level functions that automate downloading of raw tabular and spatial data from databases. Downloading i

Additionally, most functions can be forced to "start fresh" in downloading or processing data by specifying `force.redo=TRUE` in the function call.

Author(s)

R. Kyle Bocinsky <bocinsky@gmail.com>

References

Gesch, D.B. (2007) The National Elevation Dataset, in Maune, D., ed., *Digital Elevation Model Technologies and Applications: The DEM Users Manual*. 2nd Edition. American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland.

Gesch, D., Oimoen, M., Greenlee, S., Nelson, C., Steuck, M., and Tyler, D. (2002) The National Elevation Dataset. *Photogrammetric Engineering and Remote Sensing* 68(1):5–11.

Grissino-Mayer HD, Fritts HC. (1997) The International Tree-Ring Data Bank: An enhanced global database serving the global scientific community. *The Holocene* 7(2):235–238.

Menne, M., Durre, I., Korzeniewski, B., McNeal, S., Thomas, K., Yin, X., Anthony, S., Ray, R., Vose, R., B.E.Gleason, and Houston, T. (2012) *Global Historical Climatology Network-Daily (GHCN-Daily), Version 3*. http://doi.org/10.7289/V5D21VHZ.

Soil Survey Staff, Natural Resources Conservation Service, United States Department of Agriculture. Soil Survey Geographic (SSURGO) Database. Available online at http://sdmdataaccess.nrcs.usda.gov/.

Examples

```
## Not run:
# FedData Tester
library(FedData)

# Set a directory for testing
testDir <- "~/FedData Test"

dir.create(testDir, showWarnings=F, recursive=T)
setwd(testDir)

# Extract data for the Village Ecodynamics Project "VEPIIN" study area:
# http://village.anth.wsu.edu
vepPolygon <- polygon_from_extent(raster::extent(672800,740000,4102000,4170000),
                                  proj4string="+proj=utm +datum=NAD83 +zone=12")

# Get the NED (USA ONLY)
# Returns a raster
NED <- get_ned(template=vepPolygon,
               label="VEPIIN")
# Plot with raster::plot
plot(NED)


# 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=vepPolygon,
                            label="VEPIIN",
                            elements=c('prcp'))
# 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=vepPolygon,
                            label="VEPIIN",
                            elements=c('tmin','tmax'),
                            standardize=T)
# Plot the NED again
raster::plot(NED)
# Plot the spatial locations
plot(GHCN.temp$spatial, add=T, pch=1)
legend('bottomleft', pch=1, legend="GHCN Temperature Records")


# Get the NHD (USA ONLY)
NHD <- get_nhd(template=vepPolygon,
               label="VEPIIN")
# Plot the NED again
raster::plot(NED)
# Plot the NHD data
plot(NHD$NHDFlowline, add=T)
plot(NHD$NHDLine, add=T)
plot(NHD$NHDArea, col='black', add=T)
plot(NHD$NHDWaterbody, col='black', add=T)

# Get the NRCS SSURGO data (USA ONLY)
SSURGO.VEPIIN <- get_ssurgo(template=vepPolygon,
                     label="VEPIIN")
# Plot the NED again
raster::plot(NED)
# Plot the SSURGO mapunit polygons
plot(SSURGO.VEPIIN$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)


# Get the ITRDB records
ITRDB <- get_itrdb(template=vepPolygon,
                        label="VEPIIN",
                        makeSpatial=T)
# Plot the NED again
```

```
raster::plot(NED)
# 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)
```

---

download_data *Use curl to download a file.*

Description

This function makes it easy to implement timestamping and no-clobber of files.

Usage

```
download_data(url, destdir = getwd(), timestamping = T, nc = F,
  verbose = F, progress = F)
```

Arguments

url	The location of a file.
destdir	Where the file should be downloaded to.
timestamping	Should only newer files be downloaded?
nc	Should files of the same type not be clobbered?
verbose	Should cURL output be shown?
progress	Should a progress bar be shown with cURL output?

Details

If both `timestamping` and `nc` are TRUE, nc behavior trumps timestamping.

Value

A logical vector of the same length as x.

download_ghcn_daily_station — *Download the daily data for a GHCN weather station.*

Description

Download the daily data for a GHCN weather station.

Usage

```
download_ghcn_daily_station(ID, raw.dir, force.redo = F)
```

Arguments

`ID`	A character string giving the station ID.
`raw.dir`	A character string indicating where raw downloaded files should be put.
`force.redo`	If this weather station has been downloaded before, should it be updated? Defaults to FALSE.

Value

A character string representing the full local path of the GHCN station data.

download_huc4 — *Download a zipped directory containing a shapefile of the HUC4 subregions of the NHD.*

Description

Download a zipped directory containing a shapefile of the HUC4 subregions of the NHD.

Usage

```
download_huc4(raw.dir)
```

Arguments

`raw.dir`	A character string indicating where raw downloaded files should be put.

Value

A character string representing the full local path of the HUC4 zipped directory.

`download_itrdb` *Download the latest version of the ITRDB.*

Description

Downloads and parses the latest zipped (numbered) version of the ITRDB. This function includes improvements to the `read_crn` function from the **dplR** library. The principle changes are better parsing of metadata, and support for the Schweingruber-type Tucson format. Chronologies that are unable to be read are reported to the user.

Usage

```
download_itrdb(raw.dir = "./RAW/ITRDB/", force.redo = FALSE)
```

Arguments

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

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

Value

A data.table containing all of the ITRDB data.

`download_ned_tile` *Download a zipped tile from the 1 (~30 meter) or 1/3 (~10 meter) arc-second National Elevation Dataset.*

Description

Tiles are specified by a resolution, northing, and westing; northing and westing refer to the northwest corner of each NED tile, in degrees; tiles are 1x1 degree. Tiles are downloaded in zipped ESRI ArcGrid format. `downloadNED` returns the path to the downloaded zip file.

Usage

```
download_ned_tile(res = "1", tileNorthing, tileWesting, raw.dir)
```

Arguments

`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.

`tileNorthing` An integer representing the northing (latitude, in degrees north of the equator) of the northwest corner of the tile to be downloaded.

`tileWesting` An integer representing the westing (longitude, in degrees west of the prime meridian) of the northwest corner of the tile to be downloaded.

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

Value

A character string representing the full local path of the downloaded directory.

`download_nhd_subregion`

Download a zipped NHD HUC4 subregion.

Description

HUC4 subregion are specified by a unique character string, best obtained using the `get_huc4` function. `download_nhd_subregion` returns the path to the downloaded zip file.

Usage

```
download_nhd_subregion(area, res, raw.dir)
```

Arguments

`area` A 4-character string indicating the HUC4 NHD subregion to download.

`res` A character string defining the resolution of the NHD to download. Either "medium" (the default), or "high".

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

Value

A character string representing the full local path of the downloaded zip file.

`download_ssurgo_inventory`

Download a zipped directory containing a shapefile of the SSURGO study areas.

Description

Download a zipped directory containing a shapefile of the SSURGO study areas.

Usage

```
download_ssurgo_inventory(raw.dir)
```

Arguments

`raw.dir`	A character string indicating where raw downloaded files should be put.

Value

A character string representing the full local path of the SSURGO study areas zipped directory.

---

download_ssurgo_study_area *Download a zipped directory containing the spatial and tabular data for a SSURGO study area.*

Description

`download_ssurgo_study_area` first tries to download data including a state-specific Access template, then the general US template.

Usage

```
download_ssurgo_study_area(area, date, raw.dir)
```

Arguments

`area`	A character string indicating the SSURGO study area to be downloaded.
`date`	A character string indicating the date of the most recent update to the SSURGO area for these data. This information may be gleaned from the SSURGO Inventory (`get_ssurgo_inventory`).
`raw.dir`	A character string indicating where raw downloaded files should be put.

Value

A character string representing the full local path of the SSURGO study areas zipped directory.

extract_ssurgo_data *Extract data from a SSURGO databse pertaining to a set of mapunits.*

Description

`extract_ssurgo_data` creates a directed graph of the joins in a SSURGO tabular dataset, and then iterates through the tables, only retaining data pertinant to a set of mapunits.

Usage

```
extract_ssurgo_data(tables, mapunits)
```

Arguments

`tables`	A list of SSURGO tabular data.
`mapunits`	A character vector of mapunits (likely dropped from SSURGO spatial data) defining which mapunits to retain.

Value

A list of extracted SSURGO tabular data.

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.frame`s 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,
  raw.dir = "./RAW/GHCN/", extraction.dir = "./EXTRACTIONS/GHCN/",
  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 elemets 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 precipiation 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 degress 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 distinquished 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

`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.

`get_ghcn_daily_station` *Download and extract the daily data for a GHCN weather station.*

Description

`get_ghcn_daily_station` returns a named list of `data.frame`s, one for each `elements`. If `elements` is undefined, it returns all available weather tables for the station

Usage

```
get_ghcn_daily_station(ID, elements = NULL, raw.dir, standardize = F,
  force.redo = F)
```

Arguments

`ID` A character string giving the station ID.

`elements` A character vector of elemets 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 precipiation 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 degress 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 distinquished 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

`raw.dir` A character string indicating where raw downloaded files should be put.

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

`force.redo` If this weather station has been downloaded before, should it be updated? Defaults to FALSE.

Value

A named list of `data.frame`s, one for each `elements`.

`get_ghcn_inventory` *Download and crop the inventory of GHCN stations.*

Description

`get_ghcn_inventory` returns a `SpatialPolygonsDataFrame` of the GHCN stations within the specified `template`. If template is not provided, returns the entire GHCN inventory.

Usage

```
get_ghcn_inventory(template = NULL, elements = NULL, raw.dir)
```

Arguments

`template` A Raster* or Spatial* object to serve as a template for cropping.

`elements` A character vector of elemets to extract. Common elements include "tmin", "tmax", and "prcp".

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

Details

Stations with multiple elements will have multiple points. This allows for easy mapping of stations by element availability.

Value

A `SpatialPolygonsDataFrame` of the GHCN stations within the specified `template`

get_huc4 *Download and crop a shapefile of the HUC4 regions of the National Hydrography Dataset.*

Description

get_huc4 returns a SpatialPolygonsDataFrame of the HUC4 regions within the specified template. If template is not provided, returns the entire HUC4 dataset.

Usage

```
get_huc4(template = NULL, raw.dir)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping.
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing.

Value

A SpatialPolygonsDataFrame of the HUC4 regions within the specified template.

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 chrononlogies,
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, makeSpatial = F, raw.dir = "./RAW/ITRDB/",
  extraction.dir = "./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.

`makeSpatial` Should the metadata be presented as a SpatialPointsDataFrame? Defaults to FALSE.

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

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

`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.

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", raw.dir = "./RAW/NED/",
  extraction.dir = "./EXTRACTIONS/NED/", force.redo = F)
```

Arguments

template A Raster* or Spatial* object to serve as a template for cropping, and perhaps resolution.

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.

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

extraction.dir A character string indicating where the extracted and cropped DEM should be put. The directory will be created if missing. Defaults to "./EXTRACTIONS/NED/".

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.

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

Description

get_ned_tile returns a RasterLayer cropped within the specified template. If template is not provided, returns the entire NED tile.

Usage

```
get_ned_tile(template = NULL, res = "1", tileNorthing, tileWesting, raw.dir)
```

Arguments

`template` A Raster* or Spatial* object to serve as a template for cropping. If missing, entire tile is returned.

`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.

`tileNorthing` An integer representing the northing (latitude, in degrees north of the equator) of the northwest corner of the tile to be downloaded.

`tileWesting` An integer representing the westing (longitude, in degrees west of the prime meridian) of the northwest corner of the tile to be downloaded.

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

Value

A `RasterLayer` cropped within the specified `template`.

`get_nhd` *Download and crop the National Hydrography Dataset.*

Description

`get_nhd` returns a list of Spatial* objects extracted from the National Hydrography Dataset.

Usage

```
get_nhd(template, label, res = "medium", raw.dir = "./RAW/NHD/",
  extraction.dir = "./EXTRACTIONS/NHD/", 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 defining the resolution of the NHD to download. Either "medium" (the default), or "high".

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

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

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

Value

A list of Spatial* objects extracted from the National Hydrography Dataset.

get_nhd_subregion — *Download and crop data from a zipped HUC4 subregion of the National Hydrography Dataset.*

Description

get_nhd_subregion returns a list of SpatialPolygonsDataFrames of the layers of the HUC4 subregion, within the specified template. If template is not provided, returns the entire HUC4 subregion.

Usage

```
get_nhd_subregion(template = NULL, area, res, raw.dir)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping.
area	A 4-character string indicating the HUC4 NHD subregion to download and crop.
res	A character string defining the resolution of the NHD to download. Either "medium" (the default), or "high".
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing.

Value

A SpatialPolygonsDataFrame of the HUC4 regions within the specified template.

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 = "./RAW/SSURGO/",
  extraction.dir = "./EXTRACTIONS/SSURGO/", 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.

`get_ssurgo_inventory` *Download and crop a shapefile of the SSURGO study areas.*

Description

`get_ssurgo_inventory` returns a `SpatialPolygonsDataFrame` of the SSURGO study areas within the specified `template`. If template is not provided, returns the entire SSURGO inventory of study areas.

Usage

```
get_ssurgo_inventory(template = NULL, raw.dir)
```

Arguments

`template`	A Raster* or Spatial* object to serve as a template for cropping.
`raw.dir`	A character string indicating where raw downloaded files should be put. The directory will be created if missing.

Value

A `SpatialPolygonsDataFrame` of the SSURGO study areas within the specified `template`.

get_ssurgo_study_area — *Download and crop the spatial and tabular data for a SSURGO study area.*

Description

get_ssurgo_study_area 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.frames with the SSURGO tabular data.

Usage

```
get_ssurgo_study_area(template = NULL, area, date, raw.dir)
```

Arguments

template	A Raster* or Spatial* object to serve as a template for cropping. If missing, whose study area is returned
area	A character string indicating the SSURGO study area to be downloaded.
date	A character string indicating the date of the most recent update to the SSURGO area for these data. This information may be gleaned from the SSURGO Inventory (get_ssurgo_inventory).
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing.

Value

A SpatialPolygonsDataFrame of the SSURGO study areas within the specified template.

pkg_test — *Install and load a package.*

Description

This is a convenience function that checks whether a package is installed, and if not, installs it.

Usage

```
pkg_test(x)
```

Arguments

x	A character string representing the name of a package.

polygon_from_extent *Turn an extent object into a polygon*

Description

Turn an extent object into a polygon

Usage

```
polygon_from_extent(x, proj4string = NULL)
```

Arguments

x	An `extent` object, or an object from which an extent object can be retrieved.
proj4string	A PROJ.4 formatted string defining the required projection. If NULL, the function will attempt to get the projection from x using `projection`

Value

A SpatialPolygons object.

read_crn *Read a Tucson-format chronology file.*

Description

This function includes improvements to the `read.crn` function from the **dplR** library. The principle changes are better parsing of metadata, and support for the Schweingruber-type Tucson format. Chronologies that are unable to be read are reported to the user. This function automatically recognizes Schweingruber-type files.

Usage

```
read_crn(file)
```

Arguments

file	A character string path pointing to a `*.crn` file to be read.

Details

This wraps two other functions: `read_crn_metadata` `read_crn_data`.

Value

A list containing the metadata and chronology.

read_crn_data — *Read chronology data from a Tucson-format chronology file.*

Description

This function includes improvements to the read_crn function from the **dplR** library. The principle changes are better parsing of metadata, and support for the Schweingruber-type Tucson format. Chronologies that are unable to be read are reported to the user. The user (or read_crn) must tell the function whether the file is a Schweingruber-type chronology.

Usage

```
read_crn_data(file, SCHWEINGRUBER)
```

Arguments

file	A character string path pointing to a *.crn file to be read.
SCHWEINGRUBER	Is the file in the Schweingruber-type Tucson format?

Value

A data.frame containing the data, or if SCHWEINGRUBER==T, a list containing four types of data.

read_crn_metadata — *Read metadata from a Tucson-format chronology file.*

Description

This function includes improvements to the read_crn function from the **dplR** library. The principle changes are better parsing of metadata, and support for the Schweingruber-type Tucson format. Chronologies that are unable to be read are reported to the user. The user (or read_crn) must tell the function whether the file is a Schweingruber-type chronology.

Usage

```
read_crn_metadata(file, SCHWEINGRUBER)
```

Arguments

file	A character string path pointing to a *.crn file to be read.
SCHWEINGRUBER	Is the file in the Schweingruber-type Tucson format?

Details

Location information is converted to decimal degrees.

Value

A data.frame containing the metadata.

`sequential_duplicated` *Get a logical vector of which elements in a vector are sequentially duplicated.*

Description

Get a logical vector of which elements in a vector are sequentially duplicated.

Usage

```
sequential_duplicated(x, rows = F)
```

Arguments

`x`	An vector of any type, or, if `rows`, a matrix.
`rows`	Is x a matrix?

Value

A logical vector of the same length as x.

`spdf_from_polygon` *Turn an SpatialPolygons object into a SpatialPolygonsDataFrame.*

Description

Turn an SpatialPolygons object into a SpatialPolygonsDataFrame.

Usage

```
spdf_from_polygon(x)
```

Arguments

`x`	An SpatialPolygons object.

Value

A SpatialPolygonsDataFrame object.

station_to_data_frame *Convert a list of station data to a single data frame.*

Description

station_to_data_frame returns a data.frame of the GHCN station data list.

Usage

```
station_to_data_frame(station.data)
```

Arguments

station.data	A named list containing station data

Details

This function unwraps the station data and merges all data into a single data frame, with the first column being in the Date class.

Value

A data.frame of the containing the unwrapped station data

substr_right *Get the rightmost "n" characters of a character string.*

Description

Get the rightmost "n" characters of a character string.

Usage

```
substr_right(x, n)
```

Arguments

x	A character string.
n	The number of characters to retrieve.

Value

A character string.

unwrap_rows *Unwraps a matrix and only keep the first n elements.*

Description

A function that unwraps a matrix and only keeps the first n elements n can be either a constant (in which case it will be repeated), or a vector

Usage

```
unwrap_rows(mat, n)
```

Arguments

mat	A matrix
n	A numeric vector

Value

A logical vector of the same length as x

Index