

# Package ‘FedData’

January 24, 2018

**Type** Package

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

**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).

**Version** 2.5.1

**Date** 2018-1-24

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

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

**URL** <https://github.com/ropensci/FedData>

**BugReports** <https://github.com/ropensci/FedData/issues>

**License** MIT + file LICENSE

**Depends** R (>= 3.2.0), sp

**Imports** data.table, devtools, soilDB, igraph, curl, methods, rgdal,  
raster, Hmisc, rgeos, readr, lubridate, dplyr, magrittr,  
foreach, ncdf4, stringr

**Repository** CRAN

**NeedsCompilation** no

**RoxygenNote** 6.0.1

**Suggests** testthat, covr, roxygen2

**LazyData** true

**Date/Publication** 2018-01-24 17:52:40 UTC

## R topics documented:

daymet_tiles . . . . .	2
get_daymet . . . . .	3
get_ghcn_daily . . . . .	4
get_itrdb . . . . .	8
get_ned . . . . .	10
get_nhd . . . . .	11
get_nlcd . . . . .	12
get_ssurgo . . . . .	14
nlcd_canopy_pam . . . . .	15
nlcd_impervious_pam . . . . .	15
nlcd_landcover_pam . . . . .	16
nlcd_tiles . . . . .	16
<b>Index</b>	<b>17</b>

---

daymet_tiles	<i>The DAYMET tiles SpatialPolygonsDataFrame.</i>
--------------	---

---

### Description

A dataset containing the DAYMET tiles.

### Usage

```
daymet_tiles
```

### Format

A SpatialPolygonsDataFrame with 1060 features and 5 variables:

**TileID** the numeric identifier of the tile

**XMin** the minimum longitude of the tile

**XMax** the maximum longitude of the tile

**YMin** the minimum latitude of the tile

**YMax** the maximum latitude of the tile

### Source

[https://github.com/khufkens/daymetr/blob/master/data/tile\\_outlines.rda](https://github.com/khufkens/daymetr/blob/master/data/tile_outlines.rda)

---

get_daymet	<i>Download and crop the 1-km DAYMET daily weather dataset.</i>
------------	---

---

### Description

get\_daymet returns a RasterBrick of weather data cropped to a given template study area.

### Usage

```
get_daymet(template, label, elements = NULL, years = NULL,
  raw.dir = "./RAW/DAYMET", extraction.dir = paste0("./EXTRACTIONS/", label,
  "/DAYMET"), force.redo = F)
```

### 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)) / 1,000,000) 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.
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing. Defaults to './RAW/DAYMET/'.
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/DAYMET/'.
force.redo	If an extraction for this template and label already exists, should a new one be created?

**Value**

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

**Examples**

```
## Not run:
# 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 DAYMET (North America only)
# Returns a list of raster bricks
DAYMET <- get_daymet(template=vepPolygon,
  label='VEPIIN',
  elements = c('prcp','tmin','tmax'),
  years = 1980:1985)

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

## End(Not run)
```

---

get_ghcn_daily	<i>Download and crop the Global Historical Climate Network-Daily data.</i>
----------------	--

---

**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 = "./RAW/GHCN",
  extraction.dir = paste0("./EXTRACTIONS/", label, "/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	<p>A character vector of elements to extract.</p> <p>The five core elements are:</p> <p>PRCP = Precipitation (tenths of mm)</p> <p>SNOW = Snowfall (mm)</p> <p>SNWD = Snow depth (mm)</p> <p>TMAX = Maximum temperature (tenths of degrees C)</p> <p>TMIN = Minimum temperature (tenths of degrees C)</p> <p>The other elements are:</p> <p>ACMC = Average cloudiness midnight to midnight from 30-second ceilometer data (percent)</p> <p>ACMH = Average cloudiness midnight to midnight from manual observations (percent)</p> <p>ACSC = Average cloudiness sunrise to sunset from 30-second ceilometer data (percent)</p> <p>ACSH = Average cloudiness sunrise to sunset from manual observations (percent)</p> <p>AWDR = Average daily wind direction (degrees)</p> <p>AWND = Average daily wind speed (tenths of meters per second)</p> <p>DAEV = Number of days included in the multiday evaporation total (MDEV)</p> <p>DAPR = Number of days included in the multiday precipitation total (MDPR)</p> <p>DASF = Number of days included in the multiday snowfall total (MDSF)</p> <p>DATN = Number of days included in the multiday minimum temperature (MDTN)</p> <p>DATX = Number of days included in the multiday maximum temperature (MDTX)</p> <p>DAWM = Number of days included in the multiday wind movement (MDWM)</p> <p>DWPR = Number of days with non-zero precipitation included in multiday precipitation total (MDPR)</p> <p>EVAP = Evaporation of water from evaporation pan (tenths of mm)</p> <p>FMTM = Time of fastest mile or fastest 1-minute wind (hours and minutes, i.e., HHMM)</p> <p>FRGB = Base of frozen ground layer (cm)</p> <p>FRGT = Top of frozen ground layer (cm)</p> <p>FRTH = Thickness of frozen ground layer (cm)</p> <p>GAHT = Difference between river and gauge height (cm)</p> <p>MDEV = Multiday evaporation total (tenths of mm; use with DAEV)</p> <p>MDPR = Multiday precipitation total (tenths of mm; use with DAPR and DWPR, if available)</p> <p>MDSF = Multiday snowfall total</p> <p>MDTN = Multiday minimum temperature (tenths of degrees C; use with DATN)</p> <p>MDTX = Multiday maximum temperature (tenths of degrees C; use with DATX)</p> <p>MDWM = Multiday wind movement (km)</p> <p>MNPN = Daily minimum temperature of water in an evaporation pan (tenths of degrees C)</p> <p>MXPN = Daily maximum temperature of water in an evaporation pan (tenths of degrees C)</p> <p>PGTM = Peak gust time (hours and minutes, i.e., HHMM)</p>

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 mulch  
 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:
# 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 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 VEP polygon
plot(vepPolygon)

# 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 VEP polygon
plot(vepPolygon)

# 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, makeSpatial = F, raw.dir = "./RAW/ITRDB",
  extraction.dir = ifelse(!is.null(label), paste0("./EXTRACTIONS/", label,
  "/ITRDB"), "./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: <ul style="list-style-type: none"> <li>• 'Total Ring Density'</li> <li>• 'Earlywood Width'</li> <li>• 'Earlywood Density'</li> <li>• 'Latewood Width'</li> <li>• 'Minimum Density'</li> <li>• 'Ring Width'</li> <li>• 'Latewood Density'</li> <li>• 'Maximum Density'</li> <li>• 'Latewood Percent'</li> </ul> if missing, all available chronologies are given.
chronology.type	A character vector of chronology type identifiers. Options include: <ul style="list-style-type: none"> <li>• 'ARSTND'</li> <li>• 'Low Pass Filter'</li> <li>• 'Residual'</li> <li>• 'Standard'</li> <li>• 'Re-Whitened Residual'</li> <li>• 'Measurements Only'</li> </ul> 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.

### Examples

```
## Not run:
# 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 ITRDB records
ITRDB <- get_itrdb(template=vepPolygon, label='VEPIIN', makeSpatial=T)

# Plot the VEP polygon
plot(vepPolygon)

# 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)
```

---

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

---

### 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 = paste0("./EXTRACTIONS/", label, "/NED"),
  raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9", "INTERLEAVE=BAND"),
  force.redo = F)
```

**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.
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/'.
raster.options	a vector of options for raster::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:
# 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)

## End(Not run)
```

---

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, raw.dir = "./RAW/NHD",
  extraction.dir = paste0("./EXTRACTIONS/", label, "/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.
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.

**Examples**

```
## Not run:
# 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 NHD (USA ONLY)
NHD <- get_nhd(template=vepPolygon, label='VEPIIN')

# Plot the VEP polygon
plot(vepPolygon)

# 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)

## 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.

**Usage**

```
get_nlcd(template, label, year = 2011, dataset = "landcover",
  raw.dir = "./RAW/NLCD", extraction.dir = paste0("./EXTRACTIONS/", label,
  "/NLCD"), raster.options = c("COMPRESS=DEFLATE", "ZLEVEL=9",
  "INTERLEAVE=BAND"), force.redo = F)
```

**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 NLCD product. Acceptable values are 2011 (default), 2006, and 2001.
dataset	A character string representing type of the NLCD product. Acceptable values are 'landcover' (default), 'impervious', and 'canopy'.
raw.dir	A character string indicating where raw downloaded files should be put. The directory will be created if missing. Defaults to './RAW/NLCD/'.
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/NLCD/'.
raster.options	a vector of options for raster::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:
# 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 NLCD (USA ONLY)
# Returns a raster
NLCD <- get_nlcd(template=vepPolygon, label='VEPIIN')

# 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 = "./RAW/SSURGO",
  extraction.dir = paste0("./EXTRACTIONS/", label, "/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.frames](#) with the SSURGO tabular data.

### Value

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

### Examples

```
## Not run:
# 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 NRCS SSURGO data (USA ONLY)
```

```
SSURGO.VEPIIN <- get_ssurgo(template=vepPolygon, label='VEPIIN')

# Plot the VEP polygon
plot(vepPolygon)

# 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('C0670', 'C0075'), label='CO_TEST')

# Let's just look at spatial data for C0675
SSURGO.areas.C0675 <- SSURGO.areas$spatial[SSURGO.areas$spatial$AREASYMBOL=='C0075',]

# And get the NED data under them for pretty plotting
NED.C0675 <- get_ned(template=SSURGO.areas.C0675, label='SSURGO_C0675')

# Plot the SSURGO mapunit polygons, but only for C0675
plot(NED.C0675)
plot(SSURGO.areas.C0675, lwd=0.1, add=T)

## End(Not run)
```

---

nlcd\_canopy\_pam

*The NLCD canopy PAM attributes.*

---

### **Description**

A dataset containing the PAM attributes.

### **Usage**

```
nlcd_canopy_pam
```

### **Format**

An object of class character of length 2345.

---

nlcd\_impervious\_pam

*The NLCD impervious PAM attributes.*

---

### **Description**

A dataset containing the PAM attributes.

### **Usage**

```
nlcd_impervious_pam
```

**Format**

An object of class character of length 2345.

---

nlcd\_landcover\_pam      *The NLCD landcover PAM attributes.*

---

**Description**

A dataset containing the PAM attributes.

**Usage**

```
nlcd_landcover_pam
```

**Format**

An object of class character of length 2606.

---

nlcd\_tiles              *The NLCD tiles SpatialPolygonsDataFrame.*

---

**Description**

A dataset containing the NLCD tiles.

**Usage**

```
nlcd_tiles
```

**Format**

A SpatialPolygonsDataFrame with 203 features and 1 variable:

**Name** the name of the tile



# Index

## \*Topic **datasets**

- daymet\_tiles, [2](#)
- nlcd\_canopy\_pam, [15](#)
- nlcd\_impervious\_pam, [15](#)
- nlcd\_landcover\_pam, [16](#)
- nlcd\_tiles, [16](#)

- data.frame, [4](#), [14](#)
- daymet\_tiles, [2](#)

- get\_daymet, [3](#)
- get\_ghcn\_daily, [4](#)
- get\_itrdb, [8](#)
- get\_ned, [10](#)
- get\_nhd, [11](#)
- get\_nlcd, [12](#)
- get\_ssurgo, [14](#)

- nlcd\_canopy\_pam, [15](#)
- nlcd\_impervious\_pam, [15](#)
- nlcd\_landcover\_pam, [16](#)
- nlcd\_tiles, [16](#)