

# Package ‘HydroData’

October 16, 2018

**Type** Package  
**Title** HydroData  
**Description** A set of functions to find, process, export and visualize land surface, infrastructural and station based datasets relevant to geography, hydrology, and earth system science.  
**Version** 0.0.7  
**Maintainer** Mike Johnson <jmj00@ucsb.edu>  
**BugReports** <https://github.com/mikejohnson51/HydroData/issues>  
**URL** <https://github.com/mikejohnson51/HydroData>  
**Depends** R(>= 3.3.0),  
AOI,  
leaflet  
**Imports** crayon,  
dams,  
foreach,  
htmlwidgets,  
httr,  
jsonlite,  
magrittr,  
sf,  
sp,  
utils,  
raster  
**Remotes** mikejohnson51/AOI  
**License** MIT + file LICENSE  
**Encoding** UTF-8  
**LazyData** true  
**RoxygenNote** 6.1.0  
**Suggests** testthat

## R topics documented:

ap	2
col_elev	3
download.url	4

explore . . . . .	4
findAirports . . . . .	5
findCDL . . . . .	6
findGAGESII . . . . .	6
findNearestAirports . . . . .	8
findNearestCOMID . . . . .	9
findNearestHUC12 . . . . .	10
findNearestNWIS . . . . .	11
findNearestWaterbodies . . . . .	12
findNED . . . . .	13
findNHD . . . . .	13
findNID . . . . .	14
findNLCD . . . . .	15
findNLDI . . . . .	16
findNWIS . . . . .	17
findRoads . . . . .	18
findSnotel . . . . .	19
findSSURGO . . . . .	20
findWaterbodies . . . . .	20
findWaterPoly . . . . .	21
findWBD . . . . .	22
HydroData . . . . .	23
meta.data . . . . .	23
mosaic.hd . . . . .	24
query_cida . . . . .	24
return.what . . . . .	25
saveHD . . . . .	25
snotel . . . . .	26
to_sf . . . . .	27
usgsStations . . . . .	27
writeHD . . . . .	28
%>% . . . . .	28

<b>Index</b>	<b>29</b>
--------------	-----------

ap

*Airports*

## Description

Dataset containing information global Airports subset to the Daymet Domain

## Usage

ap

**Format**

a dataframe instance, 1 row per station with columns:

- 'name': A character Name of airport. May or may not contain the City name.
- 'city': A character Main city served by airport. May be spelled differently from Name.
- 'country': A character Country or territory where airport is located.
- 'IATA': A character 3-letter IATA code
- 'ICAO': A numeric 4-letter ICAO code
- 'lat': A numeric Latitude of airport
- 'lon': A numeric Longitude of airport

**Source**

[OpenFlights](#)

**Examples**

```
## Not run:  
airports = HydroData::ap  
  
## End(Not run)
```

---

col\_elev

*Color Palletes Usefull for NLCD, CDL and NED plotting*

---

**Description**

Color palletes for visualizing land use and elevation raster data

**Usage**

```
col_elev
```

**Format**

An object of class character of length 23.

**Author(s)**

Mike Johnson

---

download.url	<i>Download data from URL via http::GET</i>
--------------	---

---

**Description**

Download file from URL using http:GET. File is written to the tmp directory with the basename of the URL call.

**Usage**

```
download.url(url)
```

**Arguments**

url	url
-----	-----

**Value**

a status and path to downloaded file

**Author(s)**

Mike Johnson

---

explore	<i>Explore Spatial HydroData Objects</i>
---------	--

---

**Description**

explore can be used to explore spatial HydroData objects obtained from Namely: findGHCN find-NHD findReservoir findtiger findSnotel findUSGS findWaterbodies findWS findap getCLD getKopen getNED getNLCD getWaterUse

**Usage**

```
explore(input = NULL, save = FALSE)
```

**Arguments**

input	a single, or list, of HydroData objects
save	(logical) should the leaflet HTML be saved to disk?

**Value**

leaflet map object

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
# Find USGS station near UCSB

ucsb.nwis = findUSGS(clip_unit = list("UCSB", 10, 10))
explore(ucsb.nwis)

## End(Not run)
```

findAirports

*Find Airport Data***Description**

findAirports returns a SpatialPointsDataframe of all airports within an AOI. Data comes from the **Openflights** database and includes the following attributes:

- 'name': character Name of airport. May or may not contain the City name.
- 'city': character Main city served by airport. May be spelled differently from Name.
- 'country': character Country or territory where airport is located.
- 'IATA': character 3-letter IATA code
- 'ICAO': numeric 4-letter ICAO code
- 'lat': numeric Latitude of airport
- 'lon': numeric Longitude of airport

**Usage**

```
findAirports(AOI = NULL, ids = FALSE)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of airport ICAO codes is added to returned list (default = FALSE)

**Value**

a list of minimum length 2: AOI and ap

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
ap = getAOI(state = "CO", county = "El Paso") %>% findAirports()

## End(Not run)
```

findCDL

*Find Cropland Land Cover Data from the Cropland Data Layer***Description**

findCDL returns Raster land cover data from the USDA Cropland Data Layer Dataset (**Cropscape**) for an AOI.

**Usage**

```
findCDL(AOI, year = 2017)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
year	the year(s) to download. Options include 2008-2017 for most locations. Default = 2017

**Value**

a list() of minimum length 2: AOI and cld

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
dt = getAOI(clip = list("Devil Tower")) %>% findCDL()
dt = getAOI(clip = list("Devil Tower")) %>% findCDL(2011:2016)

## End(Not run)
```

findGAGESII

*Find GAGESII points and basins***Description**

The 'Geospatial Attributes of Gages for Evaluating Streamflow, version II' (**GAGESII**) provides geospatial data and classifications for 9,322 stream gages maintained by the U.S. Geological Survey (USGS). findGAGESII returns a SpatialPointsDataFrame\* of GAGESII outlets within an AOI. Data comes from the USGS CIDA server and contains the following attributes:

- 'id' : character Internal feature number
- 'STAIID' : character USGS NWIS Station ID
- 'STANAME' : character USGS NWIS Station Name
- 'CLASS' : character Classification (Ref or Non-ref)
- 'AGGECOREGION' : numeric Aggregated ecoregion

- 'DRAIN\_SQKM' : numeric Drainage area, sq km
- 'HUC02' : numeric Hydrologic Unit Code, 2-digit
- 'LAT\_GAGE' : character
- 'LNG\_GAGE' : character Longitude, decimal degrees
- 'STATE' : character State at gage location
- 'HCDN\_2009' : character If gage is part of HCDN-2009
- 'ACTIVE09' : numeric If gage active in water year 2009
- 'FLYRS1990' : numeric Number of complete years of flow data between 1900 and 2009
- 'FLYRS1950' : numeric Number of complete years of flow data between 1950 and 2009
- 'FLYRS1990' : numeric Number of complete years of flow data between 1990 and 2009

If basins = TRUE a SpatialPolygonsDataFrame of the gage drainage basin will also be appended to the returned list and contain the following attributes:

- 'id' : character Internal feature number and data identifier
- 'ogr\_fid' : character Internal feature number
- 'area' : character Basin Area
- 'perimeter' : character Basein Parameters
- 'gage\_id' : numeric USGS NWIS Station ID

### Usage

```
findGAGESII(AOI = NULL, basins = FALSE, ids = FALSE)
```

### Arguments

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
basins	If TRUE, returns a list of GAGESII basin in addition
ids	If TRUE, a vector of gage IDs is added to retuned list (default = FALSE)

### Value

a list() of minimum length 2: AOI and gagesII

### Author(s)

Mike Johnson

### Examples

```
## Not run:
#Get GAGESII outlets for AOI
bas = getAOI(clip = list("UCSB", 10, 10)) %>% findGAGESII()

#Get GAGESII outlets and basins for AOI
bas = getAOI(clip = list("UCSB", 10, 10)) %>% findGAGESII(basins = TRUE)

## End(Not run)
```

---

findNearestAirports	<i>Find the Airports nearest a point</i>
---------------------	--

---

## Description

findNearestAirports returns a SpatialPointsDataFrame of the 'n' number of airports closest to a declared point.

## Usage

```
findNearestAirports(point = NULL, n = 5, ids = FALSE, bb = FALSE)
```

## Arguments

point	a point described by lat/long, can be piped from <a href="#">geocode</a>
n	the number of Airports to find (default = 5)
ids	If TRUE, a vector of airport ICAO IDs is added to returned list (default = FALSE)
bb	If TRUE, the geometry of the minimum bounding area of the features is added to returned list (default = FALSE)

## Value

a list() of minimum length 2: AOI and ap

## Author(s)

Mike Johnson

## See Also

findAirports

## Examples

```
## Not run:  
pt = geocode("UCSB") %>% findNearestAirports(n = 5)  
  
## End(Not run)
```



---

findNearestCOMID	<i>Find Neareast COMIDs</i>
------------------	-----------------------------

---

## Description

findNearestCOMID returns a SpatialPointsDataFrame of the 'n' number of COMIDs closest to a declared point.

## Usage

```
findNearestCOMID(point = NULL, n = 5, ids = FALSE, bb = FALSE)
```

## Arguments

point	a point described by lat/long, can be piped from <a href="#">geocode</a>
n	the number of COMIDs to find (default = 5)
ids	If TRUE, a vector of NHD COMIDs is added to returned list (default = FALSE)
bb	If TRUE, the geometry of the minimum bounding area of the features is added to returned list (default = FALSE)

## Value

a list() of minimum length 2: AOI and ap

## Author(s)

Mike Johnson

## See Also

findAirports

## Examples

```
## Not run:  
pt = geocode("University of Oregon") %>% findNearestCOMID(n = 5)  
  
## End(Not run)
```

---

findNearestHUC12	<i>Find Neareast HUC12 boundaries</i>
------------------	---------------------------------------

---

## Description

findNearestHUC12 returns a SpatialPolygonDataFrame of the 'n' number of HUC12 basins closest to a declared point.

## Usage

```
findNearestHUC12(point = NULL, n = 1, ids = FALSE, bb = FALSE)
```

## Arguments

point	a point described by lat/long, can be piped from <a href="#">geocode</a>
n	the number of basins to find (default = 5)
ids	If TRUE, a vector of basin HUC IDs is added to returned list (default = FALSE)
bb	If TRUE, the geometry of the minimum bounding area of the features is added to returned list (default = FALSE)

## Value

a list() of minimum length 2: AOI and wbd

## Author(s)

Mike Johnson

## See Also

findWBD

## Examples

```
## Not run:
pt = geocode("UCSB") %>% findNearestHUC12(n = 5)

## End(Not run)
```

---

findNearestNWIS	<i>Find Neareast NWIS stations</i>
-----------------	------------------------------------

---

## Description

findNearestNWIS returns a SpatialPointsDataFrame of the 'n' number of NWIS gages closest to a declared point.

## Usage

```
findNearestNWIS(point = NULL, n = 5, ids = FALSE, bb = FALSE)
```

## Arguments

point	a point described by lat/long, can be piped from <a href="#">geocode</a>
n	the number of NWIS gages to find (default = 5)
ids	If TRUE, a vector of station IDs is added to returned list (default = FALSE)
bb	If TRUE, the geometry of the minimum bounding area of the features is added to returned list (default = FALSE)

## Value

a list() of minimum length 2: AOI and nwis

## Author(s)

Mike Johnson

## See Also

findNWIS

## Examples

```
## Not run:  
pt = geocode("UCSB") %>% findNearestNWIS(n = 5)  
  
## End(Not run)
```

---

findNearestWaterbodies

*Find Neareast NHD Waterbodies*


---

## Description

findNearestWaterbodies returns a SpatialPointsDataFrame of the 'n' number of NHDwaterbodies closest to a declared point.

## Usage

```
findNearestWaterbodies(point = NULL, n = 5, ids = FALSE,
  bb = FALSE)
```

## Arguments

point	a point described by lat/long, can be piped from <a href="#">geocode</a>
n	the number of NWIS gages to find (default = 5)
ids	If TRUE, a vector of station IDs is added to returned list (default = FALSE)
bb	If TRUE, the geometry of the minimum bounding area of the features is added to returned list (default = FALSE)

## Value

a list() of minimum length 2: AOI and nwis

## Author(s)

Mike Johnson

## See Also

findNWIS

## Examples

```
## Not run:
pt = geocode("UCSB") %>% findNearestWaterbodies(n = 5)

## End(Not run)
```

findNED

*Find National Elevation Data (NED)***Description**

findNED returns Raster elevation data from the National Elevation Dataset (**NED**) for an AOI. Data comes from the USA National Map.

**Usage**

```
findNED(AOI = NULL, res = 1)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
res	resolution of NED data. 1 equals 1 arc second, 13 equals 1/3 arc second.

**Value**

a list() of minimum length 2: AOI and NED

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
el.paso.elev = getAOI(state = "CO", county = "El Paso") %>% findNED(res = 1)

## End(Not run)
```

findNHD

*Find National Hydrography River Networks***Description**

findNHD returns a SpatialLinesDataframe of all NHDFlowlines reaches within an AOI. Data comes from the USGS CIDA server and contain 92 attributes, perhaps most notably:

- 'comid' : integer Integer value that uniquely identifies the occurrence of each feature in the NHD
- 'reachcode' : character Unique identifier for a 'reach'. The first eight numbers are the WBD\_HUC8
- 'fdate': POSIXct Date of last feature modification
- 'gnis\_id' : character Unique identifier assigned by GNIS
- 'gnis\_name' : character Proper name, term, or expression by which a particular geographic entity is known
- 'lengthkm' : numeric Length of linear feature based on Albers Equal Area

- 'areasqkm' : numeric Area of areal feature based on Albers Equal Area,
- 'flowdir' : character Direction of flow relative to coordinate order.
- 'wbareacomi' : integer The COMID of the waterbody through which the flowline flows.
- 'ftype' : character unique identifier of a feature type

### Usage

```
findNHD(AOI = NULL, ids = FALSE)
```

### Arguments

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of NHD COMIDs is added to returned list (default = FALSE)

### Value

a list() of minimum length 2: AOI and nhd

### Author(s)

Mike Johnson

### Examples

```
## Not run:
nhd = getAOI(clip = list("Tuscaloosa, AL", 10, 10)) %>% findNHD()

## End(Not run)
```

---

findNID

*Find Dams in the US Army Core National Inventory (NID)*

---

### Description

findNID returns a SpatialPointsDataFrame of all US Army Corps Dams for an Area of Interest from the National Inventory of Dams dataset. The National Inventory of Dams (NID) is a congressionally authorized database documenting dams in the United States and its territories This dataset is accessed through the dams R package and contains 61 attributes, perhaps most notably:

- 'Dam\_Name' : character Dam Name
- 'NID\_ID' : character Unique ID for the dam
- 'River' : character Name of the river
- 'Owner\_Type' : character Type of Owner
- 'Dam\_Type' : character Type of Dam
- 'Primary\_Purpose' : numeric Primary Purpose served
- 'Dam\_Length' : numeric Length of the dam
- 'Dam\_Height' : numeric Height of the dam
- 'Max\_Discharge' : numeric Maximum Discharge
- 'Max\_Storage' : character Maximum Storage
- 'Normal\_Storage' : character Normal Storage

**Usage**

```
findNID(AOI = NULL, ids = FALSE)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of Dam IDs is added to returned list (default = FALSE)

**Value**

a list() of minimum length 2: AOI and dams

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
# Find all dams in Texas
tx.dams = getAOI(state = "TX") %>% findNID()

## End(Not run)
```

---

findNLCD

---

*Find National Land Cover Products (NLCD)*


---

**Description**

findNLCD returns Raster land cover data from the National Land Cover Dataset (**NLCD**) for an AOI. Data comes from the USA National Map and is available for years 2001, 2006, 2011. In addition to landcover, users can get data reflecting impervious surface and canopy cover.

**Usage**

```
findNLCD(AOI = NULL, year = 2011, type = "landcover")
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
year	the year(s) to download. Options include 2001, 2006, 2011. Default = 2011
type	the type of data to download. Options include landcover, canopy, and impervious. Default = landcover

**Value**

a list() of minimum length 2: AOI and NLCD

**Author(s)**

Mike Johnson

## Examples

```
## Not run:
dt = getAOI(clip = list("Devil Tower")) %>% findNLDC(2006, 'landcover')
dt = getAOI(clip = list("Devil Tower")) %>% findNLDC(2011, 'canopy')
dt = getAOI(clip = list("Devil Tower")) %>% findNLDC(2011, 'impervious')

## End(Not run)
```

---

findNLDI

*Get NLDI Data*


---

## Description

This function acts as a formal query to the NLDI

## Usage

```
findNLDI(comid = NULL, nwis = NULL, huc12 = NULL, type = NULL,
  spatial = TRUE, resources = NULL)
```

## Arguments

comid	a COMID value (see findNHD and findNearestCOMID)
nwis	a USGS station values (see findUSGS and findNearestUSGS)
huc12	a HUC12 id (see findHUC)
type	a type of data to return ("UT", "UM", "DD", "DM", "basin")
spatial	if TRUE returned data will be sp, else of type sf
resources	what type of resources to add ("comid", "huc12pp", "npdes_rad", "nwis", "wqp")

## Value

a list of spatial / sf features

## Author(s)

Mike Johnson



findNWIS

*Find USGS NWIS Stream Gages***Description**

findNWIS returns a SpatialPointsDataFrame of all USGS NWIS gages for an Area of Interest. This dataset is accessed through the NWIS web portal and contains the following attributes:

- 'OBJECTID' : character Unique ID in dataset
- 'feature\_id' : character NHD COMID for reach
- 'site\_no' : character NWIS ID number
- 'site\_name' : character Name of site
- 'da\_sqkm' : character Area drainign to gage in square kilometers
- 'lat\_reachCent' : numeric Latitude of the reach center, decimil degrees
- 'lon\_reachCent' : numeric Longitude of the reach center, decimil degrees

**Usage**

```
findNWIS(AOI = NULL, ids = FALSE, comids = FALSE)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of NIWS gage IDs are added to retuned list (default = FALSE)
comids	f TRUE, a vector of NHD COMIDs IDs are added to retuned list (default = FALSE)

**Value**

a list() of minimum length 2: AOI and nwis

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
co = getAOI(state = "CO") %>% findNWIS()

## End(Not run)
```

findRoads

*Find US Census Bureau TIGER Road Networks***Description**

findRoads returns a SpatialLinesDataFrame of TIGER road networks cropped to an Area of Interest. This dataset is accessed through the US Census geo portal and contains the following attributes:

- 'LINEARID' : character Linear feature identifier
- 'FULLNAME' : character Feature Name
- 'RTTYP' : character Route Type Code
- 'MTFCC' : character Road Classification
  - S1100 Primary Roads
  - R1011 Railroad Feature (Main, Spur, or Yard)
  - R1051 Carline, Streetcar Track, Monorail, Other Mass Transit Rail)
  - R1052 Cog Rail Line, Incline Rail Line, Tram
  - S1100 Primary Road
  - S1200 Secondary Road

**Usage**

```
findRoads(AOI = FALSE)
```

**Arguments**

AOI                      A Spatial\* or simple features geometry, can be piped from [getAOI](#)

**Value**

a list() of minimum length 2: AOI and tiger

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
AOI = getAOI(clip = list("UCSB", 10, 10)) %>% findRoads()

## End(Not run)
```

findSnotel

*Find USDA NRCS Snotel Stations***Description**

findSnotel returns a SpatialPointsDataFrame\* Objects cropped to an Area of Interest. Station data is extracted from **NRCS reports** and contains 11 attributes:

- 'NETWORK' : integer Description of Network (all = 'SNTL')
- 'STATE' : character The state the station is located in
- 'NAME': POSITct Unique station name
- 'ID' : character Unique identifier assigned by NRCS
- 'START.DATE' : character Date the station made first measurment ("Year-Month")
- 'LAT' : numeric Latitude of station, decimil degrees
- 'LON' : numeric Longitude of station, decimil degrees
- 'ELEV' : character Elevation of station
- 'COUNTY' : integer The county the station is located in
- 'HUC12.NAME': character The HUC12 name the station is located in
- 'HUC12.ID': character The HUC12 ID the station is located in

**Usage**

```
findSnotel(AOI = NULL, ids = FALSE)
```

**Arguments**

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of NHD COMIDs is added to returned list (default = FALSE)

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
CA.sno = getAOI(state = 'CA') %>% findSnotel()

## End(Not run)
```

findSSURGO

*Find SSURGO Maps and Mapunit Aggregated Attributes***Description**

SSURGO is a geospatial database of soils produced by the Natural Resources Conservation Service (NRCS). findSSURGO returns a SpatialPolygonsDataFrame of SSURGO map units and the Mapunit aggregated attributes (muaggatt table) associated with each MUKEY. In total 43 attributes are returned, all of which can be found in the [SSURGO documentation](https://sdmdataaccess.sc.egov.usda.gov/documents/). The MUAGGATT values cover a wide range of soil related questions however if a user is interested in the full range of tabular data associated with the SSURGO map units we suggest checking out the FedData package.

**Usage**

```
findSSURGO(AOI)
```

**Arguments**

AOI                      A Spatial\* or simple features geometry, can be piped from [getAOI](#)

**Value**

a list() of minimum length 2: AOI and ssurgo

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
AOI = getAOI(clip = list("UCSB", 10, 10)) %>% findSSURGO()

## End(Not run)
```

findWaterbodies

*Find National Hydrography Dataset Waterbodies***Description**

findWaterbodies returns a SpatialPolgonsDataframe of all NHDwaterbodies within an AOI. Data comes from the USGS CIDA server and contain 23 attributes, perhaps most notably:

- 'objectid' : integer Integer value that uniquely identifies the waterbody of each feature in the NHD
- 'comid' : character The COMID draining into the feature
- 'fdate': POSIXt Date of last feature modification
- 'gnis\_id' : character Unique identifier assigned by GNIS

- 'gnis\_name' : character Proper name, term, or expression by which a particular geographic entity is known
- 'meandepth' : numeric Mean depth of the waterbody
- 'lakevolume' : numeric Total waterbody volume
- 'maxdepth' : character Maximum depth of waterbody
- 'meanused' : integer The average amount of water used

### Usage

```
findWaterbodies(AOI = NULL, ids = FALSE)
```

### Arguments

AOI	A Spatial* or simple features geometry, can be piped from <a href="#">getAOI</a>
ids	If TRUE, a vector of waterbody IDs is added to returned list (default = FALSE)

### Value

a list() of minimum length 2: AOI and waterboies

### Author(s)

Mike Johnson

### Examples

```
## Not run:
getAOI(clip = "Tuscaloosa") %>% findWaterbodies()

## End(Not run)
```

---

findWaterPoly

*Find Water Polygon*


---

### Description

A function to join all NHD water featrues into a single polygon. Flowlines are sized according to there Strahler stream order multiplied by 10 m. Useful for vidualization and masking operations where a water surface is needed.

### Usage

```
findWaterPoly(AOI)
```

### Arguments

AOI	an AOI
-----	--------

### Value

a single polygon of all NHD water features.

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
getAOI(state = 'CO', county = 'El Paso') %>% findWaterPoly()

## End(Not run)
```

findWBD

*Find Watershed Boundary Geometries (WBD/HUC)***Description**

The United States Geological Survey maintains a hierarchical system of hydrologic units each assigned a unique code (HUC). The hierarchical level is described by the number of digits in the code. A two-digit code (eg. HUC 2) is the coarsest unit of aggregation while the HUC 12 is the finest resolution. The spatial geometries of these units are stored in the Watershed Boundary Dataset with coverage of the United States. `findWBD` returns a `SpatialPolygonsDataFrame*` of WBD boundaries for the specified level within an AOI. Pending the query, data comes from the USGS CIDA server or the USGS staged products FTP.

Below you can see the general factors for each HUC level:\

Name	Digits	Average Size (sqmiles)	Example Name	Example Code
Region	2	177,560	Pacific Northwest	17
Subregion	4	16,800	Lower Snake	1706
Basin	6	10,596	Lower Snake	170601
Subbasin	8	700	Imnaha River	17060102
Watershed	10	227	Upper Imnaha River	1706010201
Subwatershed	12	40	North Fork Imnaha River	170601020101

**Usage**

```
findWBD(AOI, level = 8, subbasins = FALSE, crop = TRUE,
        ids = FALSE)
```

**Arguments**

AOI	A <code>Spatial*</code> or simple features geometry, can be piped from <a href="#">getAOI</a>
level	defines the HUC level of interest (default = 8)
subbasins	If TRUE, all subbasins of the supplied level will be joined to returned list
crop	If TRUE, all objects are cropped to the AOI boundaries (default = TRUE)
ids	If TRUE, a vector of finest resolution HUC codes is added to returned list (default = FALSE)

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:
# Get Cropped HUC8s for AOI
getAOI(list("UCSB", 10, 10)) %>% findWBD()

# Get Cropped HUC10s for AOI
getAOI(list("UCSB", 10, 10)) %>% findWBD(level = 10)

# Get Cropped HUC8s, HUC10s and HUC12s for AOI
getAOI(clip = list("UCSB", 10, 10)) %>% findWBD(level = 8, subbasins = TRUE)

# Get uncropped HUC10s for AOI
getAOI(clip = list("UCSB", 10, 10)) %>% findWBD(level = 10, crop = FALSE)

## End(Not run)
```

---

HydroData	<i>HydroData</i>
-----------	------------------

---

**Description**

HydroData package

**Details**

Access earth systems geospatial and observation data programatically  
See the README on

---

meta.data	<i>Return HydroData Object metadata</i>
-----------	---

---

**Description**

Function for generating descriptive meta data for all HyrdroData objects

**Usage**

```
meta.data(AOI)
```

**Arguments**

AOI and AOI object with appended HydroData features

**Value**

a data.frame of meta.data describing a HydroData object

**Author(s)**

Mike Johnson

---

`mosaic.hd`*Mosaic*

---

**Description**

Internal function for mosaicing a list of rasters

**Usage**

```
mosaic.hd(input)
```

**Arguments**

<code>input</code>	raster stack
--------------------	--------------

**Author(s)**

Mike Johnson

---

`query_cida`*CIDA Server Query*

---

**Description**

CIDA Server Query

**Usage**

```
query_cida(AOI, type, spatial = TRUE)
```

**Arguments**

<code>AOI</code>	and AOI object generated with the AOI package, or a bounding box
<code>type</code>	the WBD or nhdplus object to return
<code>spatial</code>	return 'sp' (default) if FALSE return 'sf'

**Value**

a Spatial object



---

`return.what`*Choose what to return for HydroData Calls*

---

**Description**

A function defining what should be returned in a HydroData object

**Usage**

```
return.what(AOI, type, report, vals)
```

**Arguments**

AOI	Spatial object
type	the index of the AOI object to report on
report	character sting to append values to
vals	the ids call

**Value**

a list of HydroData components

**Author(s)**

Mike Johnson

---

`saveHD`*Save File*

---

**Description**

Save File

**Usage**

```
saveHD(AOI = NULL, path = NULL, ext = "shp")
```

**Arguments**

AOI	What data to write
path	State abb
ext	country call

---

snotel	<i>Snotel Stations</i>
--------	------------------------

---

## Description

snotelNRCS SNOTEL station metadata

## Usage

snotel

## Format

a dataframe instance, 1 row per station with columns:

- 'NETWORK': A integer Network of interest
- 'STATE': A integer State abbreviation
- 'NAME': A character Site name
- 'START.DATE': A integer Day of first measurement
- 'LAT': A integer Station Latitude
- 'LONG': A character Station Longitude
- 'ELEV': A character Station Elevation
- 'COUNTY': A character County
- 'HUC12.NAME': A character HUC 12 Name
- 'HUC12.ID': A character HUC 12 code

## Source

**SNOTEL INFORMATION**

## Examples

```
## Not run:  
tiles = HydroData::snotelStations.rda  
  
## End(Not run)
```

---

`to_sf`*Convert Spatial HydroData Objects to Simple Features*

---

**Description**

A function to convert all Spatial\* HydroData objects to simple feature geometries. Non-Spatial objects (eg raster and list components) will be skipped over.

**Usage**

```
to_sf(hydro.data.object = NULL)
```

**Arguments**

`hydro.data.object`  
a HydroData object with Spatial components

**Value**

a list with the same length as the input

**Author(s)**

Mike Johnson

**Examples**

```
## Not run:  
AOI = getAOI(clip = 'UCSB') %>% findNED %>% findNHD %>% findNWIS %>% to_sf  
  
## End(Not run)
```

---

`usgsStations`*USGS NWIS station information*

---

**Description**

Dataset containing information about USGS stations in the United States

**Usage**

```
usgsStations
```

**Format**

- a dataframe instance, 1 row per station with columns:
- 'OBJECTID': A character Object id in the dataset
  - 'feature\_id': A character NHD COMID of reach
  - 'site\_no': A character USGS site number
  - 'site\_name': A character USGS site name
  - 'da\_sqkm': A numeric Area that drains to the location in square kilometers
  - 'lat\_reachCent': A numeric Latitude of NHD reach center
  - 'lon\_reachCent': A numeric Longitude of NHD reach center

**Source**

Compiled from USGS and NHD datasets

**Examples**

```
## Not run:
usgs = HydroData::usgsStations

## End(Not run)
```

---

writeHD	<i>Write Data to disk</i>
---------	---------------------------

---

**Description**

internal function used to write data to disk according to it meta.data found via HydroData::meta.data()

**Usage**

```
writeHD(AOI, class, dir, feature_name, ext = "shp")
```

**Arguments**

AOI	a HydroData object with components to write to disk
class	objects class
dir	directory
feature_name	object id
ext	shp or gpkg

---

%>%	<i>re-export magrittr pipe operator</i>
-----	---

---

**Description**

re-export magrittr pipe operator

# Index

## \*Topic **datasets**

- ap, [2](#)
- col\_elev, [3](#)
- snotel, [26](#)
- usgsStations, [27](#)
- %>%, [28](#)
- ap, [2](#)
- col\_elev, [3](#)
- download.url, [4](#)
- explore, [4](#)
- findAirports, [5](#)
- findCDL, [6](#)
- findGAGESII, [6](#)
- findNearestAirports, [8](#)
- findNearestCOMID, [9](#)
- findNearestHUC12, [10](#)
- findNearestNWIS, [11](#)
- findNearestWaterbodies, [12](#)
- findNED, [13](#)
- findNHD, [13](#)
- findNID, [14](#)
- findNLCD, [15](#)
- findNLDI, [16](#)
- findNWIS, [17](#)
- findRoads, [18](#)
- findSnotel, [19](#)
- findSSURGO, [20](#)
- findWaterbodies, [20](#)
- findWaterPoly, [21](#)
- findWBD, [22](#)
- geocode, [8–12](#)
- getAOI, [5–7](#), [13–15](#), [17–22](#)
- HydroData, [23](#)
- HydroData-package (HydroData), [23](#)
- meta.data, [23](#)
- mosaic.hd, [24](#)
- query\_cida, [24](#)
- return.what, [25](#)
- saveHD, [25](#)
- snotel, [26](#)
- to\_sf, [27](#)
- usgsStations, [27](#)
- writeHD, [28](#)