Package ‘nhdR’

January 3, 2022

Title Tools for Working with the National Hydrography Dataset

Version 0.5.7


URL https://github.com/jsta/nhdR

BugReports https://github.com/jsta/nhdR/issues

Depends R (>= 3.5.0), maps

License GPL

Imports rappdirs, rgdal, sf, httr, rvest, xml2, foreign, ggplot2, rlang, dplyr, curl, units, stringr, memoise, purrr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, wikilake, sp, rgeos, testthat (>= 2.1.0), covr, crul, lwgeom, s2

VignetteBuilder knitr

SystemRequirements 7-zip command line tool (7z)

Language en-US

NeedsCompilation no

Author Jemma Stachelek [aut, cre] (<https://orcid.org/0000-0002-5924-2464>)

Maintainer Jemma Stachelek <stachel2@msu.edu>

Repository CRAN

Date/Publication 2022-01-03 17:50:13 UTC
R topics documented:

nhdR-package ................................................. 2
bbox2poly .................................................. 3
extract_network .............................................. 3
find_state .................................................... 5
find_vpu ...................................................... 6
great_lakes .................................................. 6
gull .......................................................... 7
gull_flow ..................................................... 7
leaf_reaches .................................................. 8
mendota ...................................................... 9
mendota_network ............................................ 9
nhd_dl_state ................................................. 10
nhd_get ...................................................... 11
nhd_info ..................................................... 11
nhd_list ..................................................... 12
nhd_load ..................................................... 13
nhd_plus_get ............................................... 14
nhd_plus_info .............................................. 15
nhd_plus_list ............................................... 16
nhd_plus_load ............................................... 16
nhd_plus_query ............................................. 18
nhd_query ................................................... 19
select_point_overlay ...................................... 21
select_poly_overlay ...................................... 21
sunapee ..................................................... 22
sunapee_network .......................................... 22
terminal_reaches .......................................... 23
tip_reaches ................................................ 25
toUTM ....................................................... 25
vpu_shp ..................................................... 26

Index 27

| nhdR-package | R interface to the National Hydrography Dataset |

Description

R interface to the National Hydrography Dataset

Author(s)

<stachel2@msu.edu>
bbox2poly

Convert a bounding box to polygon

Description

Convert a bounding box to polygon

Usage

bbox2poly(bbox)

Arguments

bbox object of class bbox from sf

Value

An sfc object from the sf package

Examples

## Not run:
library(sf)
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")

pnt <- st_as_sf(wk, coords = c("Lon", "Lat"), crs = 4326)
pnt <- st_transform(pnt, st_crs(vpu_shp))
qry <- nhd_plus_query(wk$Lon, wk$Lat,
    dsn = c("NHDWaterbody"), buffer_dist = 0.05)
wbd <- qry$sp$NHDWaterbody[which.max(st_area(qry$sp$NHDWaterbody)), ]
bbox2poly(st_bbox(wbd))

## End(Not run)

extract_network

Return nhd plus stream network upstream of a waterbody

Description

Return nhd plus stream network upstream of a waterbody
Usage

```
extract_network(
  lon = NA,
  lat = NA,  
  lines = NA,
  lines_network = TRUE,  
  buffer_dist = 0.01,
  maxsteps = 3,
  approve_all_dl = FALSE,
  temporary = TRUE,
  ...
)
```

Arguments

- `lon` numeric decimal degree longitude
- `lat` numeric decimal degree latitude
- `lines` sf spatial lines object to limit extent of the network search
- `lines_network` boolean treat lines as the complete network object. If FALSE, simply start network extraction at the terminal reach of the lines object.
- `buffer_dist` numeric buffer around lat-lon point in dec. deg.
- `maxsteps` maximum number of stream climbing iterations
- `approve_all_dl` logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- `temporary` logical set FALSE to save data to a persistent rappdirs location
- `...` parameters passed on to sf::st_read

Details

The lon and lat arguments are used for querying the corresponding lake polygon layer which is then used to climb its intersecting stream network.

Value

An sf data frame with LINESTRING geometries

Examples

```
## Not run:
library(mapview)
library(sf)

# headwater lakes have no upstream network
coords <- data.frame(lat = 46.32711, lon = -89.58893)
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

# fails if no lake nhdp lake found within the buffer at the query point
```
coords <- data.frame(lat = 43.62453, lon = -85.47164)
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

coords <- data.frame(lat = 20.79722, lon = -156.47833)
# use a non-geographic (projected) buffer size
res <- extract_network(coords$lon, coords$lat, maxsteps = 9,
   buffer_dist = units::as_units(5, "km"))

# use a projected buffer size
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

# no upstream network for lakes intersecting the Great Lakes
coords <- data.frame(lat = 44.6265, lon = -86.23121)
res <- extract_network(coords$lon, coords$lat, maxsteps = 3)

coords <- data.frame(lat = 42.96523, lon = -89.2527)
res <- extract_network(coords$lon, coords$lat, maxsteps = 9)

mapview(res)

## End(Not run)

---

**find_state**

---

**Description**

find_state

**Usage**

`find_state(pnt, abb = FALSE)`

**Arguments**

- `pnt`: an sf point object
- `abb`: logical, return a state abbreviation?

**Examples**

## Not run:

```r
pnt <- st_as_sf(data.frame(Lon = -107.2, Lat = 39.45),
   coords = c("Lon", "Lat"), crs = 4326)

## End(Not run)```
find_vpu  

Find VPU

Description

Find Vector Processing Unit from sf object

Usage

find_vpu(pnt)

Arguments

pnt    sf object

Value

A character vector of vpu ids

Examples

## Not run:
library(sf)

# vpu centers
pnt <- st_cast(st_point_on_surface(nhdR::vpu_shp), "POINT")

find_vpu(pnt[1, ])
find_vpu(pnt)

find_vpu(nhdR::gull$sp$NHDWaterbody[1, ])
find_vpu(nhdR::gull$sp$NHDWaterbody)

## End(Not run)

great_lakes  

Data and spatial polygons of the Great Lakes

Description

Data and spatial polygons of the Great Lakes

Usage

great_lakes(spatial = FALSE)
Arguments

spatial logical, return Great Lakes polygons?

Value

A data frame of North America Great Lakes with optional geometry column

Examples

```r
gl <- great_lakes()
## Not run:
gl <- great_lakes(spatial = TRUE)
## End(Not run)
```

gull

List of simple features lake polygons and flowlines within a buffer around Gull Lake Michigan.

description

Data from NHD Plus

details

List of simple features lake polygons and flowlines within a buffer around Gull Lake Michigan.

gull_flow

Flowlines within a buffer around Gull Lake Michigan including flow information.

description

Data from NHD Plus

details

List of simple features lake polygons and flowlines within a buffer around Gull Lake Michigan.
Description

A leaf reach is a stream flowline that has upstream connections but is not in the focal set.

Usage

leaf_reaches(
    lon = NA,
    lat = NA,
    network = NA,
    approve_all_dl = FALSE,
    temporary = TRUE,
    ...
)

Arguments

lon numeric decimal degree longitude. optional. See Details section.
lat numeric decimal degree latitude. optional. See Details section.
network sf lines collection. optional. See Details section.
approve_all_dl logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
temporary logical set FALSE to save data to a persistent rappdirs location
... parameters passed on to sf::st_read

Value

An sf data frame with LINESTRING geometries

Examples

```r
## Not run:
coords <- data.frame(lat = 20.79722, lon = -156.47833)
# nhd_plus_get(
#  nhdR::find_vpu(
#    sf::st_as_sf(coords, coords = c("lon", "lat"), crs = 4326)),
#    temporary = FALSE)
leaf_reaches(coords$lon, coords$lat)

coords <- data.frame(lat = 41.42217, lon = -73.24189)
l_reach <- leaf_reaches(coords$lon, coords$lat)

network_focal <- nhd_plus_query(lon = coords$lon, lat = coords$lat,
                                 dsn = "NHDFlowline", buffer_dist = units::as_units(2, "km"))$sp$NHDFlowline
```
network <- nhd_plus_query(lon = coords$lon, lat = coords$lat,  
  dsn = "NHDFlowline", buffer_dist = units::as_units(5, "km"))$sp$NHDFlowline
l_reach <- leaf_reaches(network = network_focal)

plot(network$geometry)
plot(network_focal$geometry, col = "darkgreen", add = TRUE)
plot(l_reach$geometry, col = "red", add = TRUE)

## End(Not run)

---

**mendota**

List of simple features lake polygons and flowlines within a buffer around Lake Mendota.

---

**Description**

Data from NHD Plus

**Details**

mendota

---

**mendota_network**

Upstream flowlines connected to Lake Mendota.

---

**Description**

Data from NHD Plus

**Details**

mendota_network
Description

`nhd_dl_state`

Usage

```r
nhd_dl_state(
  state,
  state_exists,
  yes_dl,
  file_ext,
  dsn = NA,
  wkt_filter = NA,
  temporary = FALSE,
  ...
)
```

Arguments

- `state`: state abbreviation
- `state_exists`: 1 for file exists on disk
- `yes_dl`: 1 for downloading the state gdb file
- `file_ext`: file extension ("gdb", etc)
- `dsn`: name of gdb layer
- `wkt_filter`: a text string of coordinates see `sf::st_read`
- `temporary`: logical set FALSE to save data to a persistent rappdirs location
- `...`: other arguments passed to `sf::st_read`

Examples

```r
## Not run:
nhd_dl_state("RI", 1, 0, NA, "NHDWaterbody")

## End(Not run)
```
**nhd_get**  
*Download and cache NHD data by state*

**Description**  
Download and cache NHD data by state

**Usage**  
\[ \text{nhd\_get}(\text{state} = \text{NA}, \text{force\_dl} = \text{FALSE}, \text{force\_unzip} = \text{FALSE}, \text{temporary} = \text{TRUE}) \]

**Arguments**
- **state**: character state abbreviation includes "DC", "PR", and "VI"  
- **force\_dl**: logical force a re-download of the requested data  
- **force\_unzip**: logical force an unzip of downloaded data  
- **temporary**: logical set FALSE to save data to a persistent rappdirs location

**Value**  
An invisible list of file paths to NHD data for the specified state

**Examples**

```r  
## Not run:  
nhd_get(state = c("DC"))  
nhd_get(state = c("RI", "CT"))  
## End(Not run)  
```

---

**nhd_info**  
*Return NHD layer metadata and field listing*

**Description**  
Return NHD layer metadata and field listing

**Usage**  
\[ \text{nhd\_info}(\text{state}, \text{dsn}) \]

**Arguments**
- **state**: character  
- **dsn**: character
Value

An ogrinfo object from the sf package

Examples

```r
## Not run:
nhd_info("DC", "NHDWaterbody")
## End(Not run)
```

---

**nhd_list**

*List available locally cached NHD layers per state*

Description

List available locally cached NHD layers per state

Usage

`nhd_list(state)`

Arguments

- `state` character state abbreviation

Value

A character vector of NHD layers for the specified state

Examples

```r
## Not run:
nhd_list(state = "DC")
## End(Not run)
```
Load NHD layers into current session

**Usage**

```r
nhd_load(
  state, 
  dsn, 
  file_ext = NA, 
  approve_all_dl = FALSE, 
  temporary = FALSE, 
  wkt_filter = NA, 
  ... 
)
```

**Arguments**

- `state` character state abbreviation
- `dsn` character name of a NHD layer
- `file_ext` character choice of "shp" for spatial data and "dbf" or "gpkg" for non-spatial. optional
- `approve_all_dl` logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- `temporary` logical set FALSE to save data to a persistent rappdirs location
- `wkt_filter` character. WKT spatial filter for selection. See sf::st_read
- `...` arguments passed to sf::st_read

**Details**

This function will ask the user to approve downloading missing data unless approve_all_dl is set to TRUE.

**Value**

Spatial simple features object or data frame depending on the dsn type and value passed to file_ext
nhd_plus_get

Description

Download and cache NHDplus data by vector processing unit

Usage

nhd_plus_get(
  vpu = NA,
  component = "NHDSnapshot",
  force_dl = FALSE,
  force_unzip = FALSE,
  temporary = TRUE
)

Arguments

vpu numeric vector processing unit
component character component name
force_dl logical force a re-download of the requested data
force_unzip logical force an unzip of downloaded data
temporary logical set FALSE to save data to a persistent rappdirs location

Value

An invisible list of file paths to NHDplus data for the specified vpu
Examples

```r
## Not run:
# Spatial
nhd_plus_get(vpu = 4)
nhd_plus_get(vpu = "10L")
nhd_plus_get(vpu = 1, component = "NHDPlusAttributes")

# Non-spatial
nhd_plus_get(vpu = "National", component = "V1_To_V2_Crosswalk")
nhd_plus_get(vpu = 4, component = "EROMExtension")

## End(Not run)
```

---

### nhd_plus_info

Return NHDplus layer metadata and field listing

#### Description

Return NHDplus layer metadata and field listing

#### Usage

```r
nhd_plus_info(vpu, component, dsn, file_ext = NA)
```

#### Arguments

- `vpu`: numeric vector processing unit
- `component`: character component name
- `dsn`: character data source name
- `file_ext`: character choice of "shp" for spatial data and "dbf" for non-spatial. optional

#### Value

An ogrinfo object from the sf package

#### Examples

```r
## Not run:
nhd_plus_info(vpu = 4, component = "NHDSnapshot", dsn = "NHDWaterbody")
nhd_plus_info(vpu = 1, component = "NHDPlusAttributes", dsn = "PlusFlow")

## End(Not run)
```
nhd_plus_list  
*List available locally cached NHDplus layers per state*

**Description**

List available locally cached NHDplus layers per state

**Usage**

```
    nhd_plus_list(vpu, component = "NHDSnapshot", file_ext = NA, ...)
```

**Arguments**

- `vpu`: numeric vector processing unit
- `component`: character component name
- `file_ext`: character choice of "shp" for spatial data and "dbf" for non-spatial. optional
- `...`: arguments passed to list.files. optional.

**Value**

A character vector of NHD layers for the specified vpu

**Examples**

```r
    # Not run:
    nhd_plus_list(vpu = 4)
    nhd_plus_list(vpu = 4, full.names = TRUE)

    nhd_plus_list(vpu = 1, component = "NHDPlusAttributes")
    nhd_plus_list(vpu = "National", component = "V1_To_V2_Crosswalk")

    # End(Not run)
```

---

nhd_plus_load  
*Load NHDplus layers into current session*

**Description**

Load NHDplus layers into current session
Usage

```r
nhd_plus_load(
  vpu,
  component = "NHDSnapshot",
  dsn,
  file_ext = NA,
  approve_all_dl = FALSE,
  force_dl = FALSE,
  temporary = FALSE,
  pretty = FALSE,
  wkt_filter = NA,
  ...
)
```

Arguments

- `vpu`: numeric vector processing unit
- `component`: character component name
- `dsn`: data source name
- `file_ext`: character choice of "shp" for spatial data and "dbf" for non-spatial. optional
- `approve_all_dl`: logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive
- `force_dl`: logical force a re-download of the requested data
- `temporary`: logical set FALSE to save data to a persistent rappdirs location
- `pretty`: more minimal pretty printing st_read relative to "quiet"
- `wkt_filter`: character. WKT spatial filter for selection. See sf::st_read
- `...`: parameters passed on to sf::st_read

Details

This function will ask the user to approve downloading missing data unless `approve_all_dl` is set to TRUE. Output of this function is saved in active memory (memoized) to speed up repeated function calls.

Value

spatial object

Examples

```r
## Not run:
# Spatial
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody")
dt <- nhd_plus_load(c(1, 2), "NHDSnapshot", "NHDWaterbody")
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDFlowline")
dt <- nhd_plus_load(4, "NHDPlusCatchment", "Catchment")
```
# Quieter printing
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", pretty = TRUE)
# Quietest printing
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", quiet = TRUE)

# Non-spatial
dt <- nhd_plus_load(1, "NHDPlusAttributes", "PlusFlow")
dt <- nhd_plus_load("National", "V1_To_V2_Crosswalk", "NHDPlusV1Network_V2Network_Crosswalk")
gridcode <- nhd_plus_load(1, "NHDPlusCatchment", "featuregridcode")
flowline_vaa <- nhd_plus_load(1, "NHDPlusAttributes", "PlusFlowlineVAA")
eromflow <- nhd_plus_load(4, "EROMExtension", "EROM_010001")

# Character VPU
plusflow <- nhd_plus_load(vpu = "10L", "NHDPlusAttributes", "PlusFlow")

# Spatial filtering via wkt_filter
dt <- nhd_plus_load(4, "NHDSnapshot", "NHDWaterbody", wkt_filter = "POINT (-85.411 42.399)")

## End(Not run)

---

nhd_plus_query

Select NHDplus features via polygon or circular buffer of coordinate pair

## Description

Select NHDplus features via polygon or circular buffer of coordinate pair

## Usage

nhd_plus_query(
  lon = NA,
  lat = NA,
  poly = NA,
  dsn,
  buffer_dist = units::as_units(4.75, "km"),
  approve_all_dl = FALSE,
  temporary = TRUE,
  ...
)

## Arguments

- **lon**: numeric longitude. optional
- **lat**: numeric latitude. optional
- **poly**: sfc polygon. optional
- **dsn**: character data source
buffer_dist numeric buffer in units of coordinate degrees
approve_all_dl logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
temporary logical set FALSE to save data to a persistent rappdirs location

... parameters passed on to sf::st_read

Value

A list of sf spatial objects

Examples

## Not run:
library(sf)
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")
pnt <- st_as_sf(wk, coords = c("Lon", "Lat"), crs = 4326)
pnt <- st_transform(pnt, st_crs(vpu_shp))
# nhd_plus_list(nhdR::find_vpu(pnt))

qry <- nhd_plus_query(wk$Lon, wk$Lat,
  dsn = c("NHDWaterbody", "NHDFlowLine"), buffer_dist = units::as_units(4.75, "km"))

plot(qry$sp$NHDWaterbody$geometry, col = "blue")
plot(qry$sp$NHDFlowLine$geometry, col = "cyan", add = TRUE)
plot(qry$pnt, col = "red", pch = 19, add = TRUE)
axis(1)
axis(2)

library(ggplot2)
ggplot(qry$sp$NHDWaterbody) + geom_sf()

# query with a polygon
wbd <- qry$sp$NHDWaterbody[which.max(st_area(qry$sp$NHDWaterbody)), ]
qry_lines <- nhd_plus_query(poly = st_as_sfc(st_bbox(wbd)),
  dsn = "NHDFlowLine")
ggplot() +
  geom_sf(data = qry$sp$NHDWaterbody) +
  geom_sf(data = qry_lines$sp$NHDFlowLine, color = "red")

## End(Not run)
Usage

nhd_query(
  lon = NA,
  lat = NA,
  poly = NA,
  dsn,
  approve_all_dl = FALSE,
  buffer_dist = units::as_units(4.75, "km"),
  temporary = TRUE,
  ...
)

Arguments

lon numeric longitude
lat numeric latitude
poly sfc polygon. optional
dsn character data source
approve_all_dl logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
buffer_dist numeric buffer with specified units
temporary logical set FALSE to save data to a persistent rappdirs location
...
other arguments passed to sf::st_read

Examples

## Not run:
library(sf)
wk <- wikilake::lake_wiki("Worden Pond")
qry <- nhd_query(wk$Lon, wk$Lat, dsn = c("NHDWaterbody", "NHDFlowLine"),
  buffer_dist = units::as_units(1, "km"))
qry$sp$NHDWaterbody <- dplyr::filter(qry$sp$NHDWaterbody, FType != 466)
plot(sf::st_geometry(qry$sp$NHDWaterbody), col = "blue")
plot(sf::st_geometry(qry$sp$NHDFlowLine), col = "red", add = TRUE)
plot(qry$pnt, col = "red", pch = 19, add = TRUE)
axis(1)
axis(2)

# query with a polygon
wbd <- qry$sp$NHDWaterbody[
  order(st_area(qry$sp$NHDWaterbody), decreasing = TRUE), ][1, ]
qry_lines <- nhd_query(poly = st_as_sfc(st_bbox(wbd)), dsn = "NHDFlowLine")
library(ggplot2)
ggplot() +
  geom_sf(data = qry$sp$NHDWaterbody) +
  geom_sf(data = qry_lines$sp$NHDFlowLine, color = "red")

## End(Not run)
select_point_overlay  Select features clipped by a point buffer around a point

Description

Select features clipped by a point buffer around a point

Usage

```r
select_point_overlay(pnt, sp, buffer_dist = units::as_units(4.75, "km"))
```

Arguments

- `pnt`: geographic point of class sfc
- `sp`: list of sf data frames
- `buffer_dist`: numeric buffer with specified units

Value

A list of sf spatial objects

Examples

```r
## Not run:
wk <- wikilake::lake_wiki("Gull Lake (Michigan)")
pnt <- sf::st_sfc(sf::st_point(c(wk$Lon, wk$Lat)))
sf::st_crs(pnt) <- 4326
sp <- lapply(c("NHDWaterbody", "NHDFlowLine"),
             function(x) nhd_plus_load(vpu = 4, dsn = x))
names(sp) <- c("NHDWaterbody", "NHDFlowLine")
qry <- select_point_overlay(pnt = pnt, sp = sp)
plot(qry$NHDWaterbody$geometry, col = "blue")
plot(qry$NHDFlowLine$geometry, col = "cyan", add = TRUE)
## End(Not run)
```

select_poly_overlay  Select features clipped by a polygon

Description

Select features clipped by a polygon

Usage

```r
select_poly_overlay(poly, sp)
```
Arguments

poly     sf *polygon object
sp      list of sf data frames

Value

A list of sf spatial objects

---

sunapee   List of simple features lake polygons and flowlines within a buffer around Lake Sunapee.

---

Description

Data from NHD Plus

Details

sunapee

---

sunapee_network   Upstream flowlines connected to Lake Sunapee.

---

Description

Data from NHD Plus

Details

sunapee_network
terminal_reaches

**Return terminal reaches from collection intersecting lake**

**Description**

In the case of a network query, a terminal reach is a stream flowline that has no downstream reaches in-network. In the case of a point query, a terminal reach is a flowline that exits the intersecting surface waterbody.

**Usage**

```r
terminal_reaches(
  lon = NA,
  lat = NA,
  buffer_dist = 0.01,
  network = NA,
  lakepoly = NA,
  lakewise = FALSE,
  lakesize_threshold = 4,
  approve_all_dl = FALSE,
  temporary = TRUE,
  ...
)
```

**Arguments**

- `lon` numeric decimal degree longitude. optional. See Details section.
- `lat` numeric decimal degree latitude. optional. See Details section.
- `buffer_dist` numeric buffer around lat-lon point in dec. deg.
- `network` sf lines collection. optional. See Details section.
- `lakepoly` sf polygon. optional. See Details section.
- `lakewise` logical. If TRUE, return the terminal reaches of all lakes in the stream network rather than a single terminal reach of the focal lake.
- `lakesize_threshold` numeric above which to count as a lake (ha).
- `approve_all_dl` logical blanket approval to download all missing data. Defaults to TRUE if session is non-interactive.
- `temporary` logical set FALSE to save data to a persistent rappdirs location
- `...` parameters passed on to sf::st_read
Details

There are multiple ways to execute `terminal_reaches`:

- Only providing lon + lat arguments - this will query the corresponding lake polygon layer and find the terminal reach of the lake intersecting a buffer around the specified point.
- Only providing a lake polygon - this is essentially the same as above except there is no preliminary lake polygon query.
- Only providing a network of stream lines - this provides the most downstream reach irrespective of lakes.

Value

An sf data frame with LINESTRING geometries

Examples

```r
## Not run:
library(sf)
library(mapview)

coords <- data.frame(lat = 46.32711, lon = -89.58893)
t_reach <- terminal_reaches(coords$lon, coords$lat)

cords <- data.frame(lat = 20.79722, lon = -156.47833)
# use a non-geographic (projected) buffer size
   t_reach <- terminal_reaches(coords$lon, coords$lat, buffer_dist = units::as_units(5, "km"))

cords <- data.frame(lat = 42.96628, lon = -89.25264)
t_reach <- terminal_reaches(coords$lon, coords$lat)

cords <- data.frame(lat = 41.42217, lon = -73.24189)
t_reach <- terminal_reaches(coords$lon, coords$lat)

mapview(st_as_sf(coords, coords = c("lon", "lat"), crs = 4326)) +
mapview(t_reach$geometry, color = "red")

coords <- data.frame(lat = 41.859080, lon = -71.575422)
network <- nhd_plus_query(lon = coords$lon, lat = coords$lat, dsn = "NHDFlowline", buffer_dist = 0.05)$sp$NHDFlowline
t_reach <- terminal_reaches(network = network)
t_reach_lake <- terminal_reaches(network = network, lakewise = TRUE, lakesize_threshold = 1)

mapview(network) + mapview(t_reach_lake, color = "green") +
mapview(t_reach, color = "red")

## End(Not run)
```
tip_reaches

Return tip reaches from a network

Description
A tip reach is a stream flowline with no upstream connections.

Usage
tip_reaches(network = NA)

Arguments
network sf lines collection. optional. See Details section.

Value
An sf data frame with LINESTRING geometries

Examples
## Not run:
coords <- data.frame(lat = 41.42217, lon = -73.24189)
network <- nhd_plus_query(lon = coords$lon, lat = coords$lat,
    dsn = "NHDFlowline", buffer_dist = units::as_units(5, "km"))$sp$NHDFlowline
t_reaches <- tip_reaches(network = network)

plot(network$geometry)
plot(t_reaches$geometry, col = "red", add = TRUE)

## End(Not run)

toUTM

Re-project to appropriate UTM zone

Description
Re-project to appropriate UTM zone

Usage
toUTM(sf_object)

Arguments
sf_object an sf object
Value

A transformed sf object

Examples

```r
## Not run:
data(gull)
gull_ <- gull$sp$NHDWaterbody
st_crs(gull_)
gull_ <- st_transform(gull_, 4326)
st_crs(gull_)
st_crs(toUTM(gull_[1, ]))

## End(Not run)
```
Index

* datasets
  - gull, 7
  - gull_flow, 7
  - mendota, 9
  - mendota_network, 9
  - sunapee, 22
  - sunapee_network, 22
  - vpu_shp, 26

- sunapee, 22
- sunapee_network, 22
- terminal_reaches, 23, 24
- tip_reaches, 25
- toUTM, 25
- vpu_shp, 26

- bbox2poly, 3
- extract_network, 3
- find_state, 5
- find_vpu, 6
- great_lakes, 6
- gull, 7
- gull_flow, 7
- leaf_reaches, 8
- mendota, 9
- mendota_network, 9
- nhd_dl_state, 10
- nhd_get, 11
- nhd_info, 11
- nhd_list, 12
- nhd_load, 13
- nhd_plus_get, 14
- nhd_plus_info, 15
- nhd_plus_list, 16
- nhd_plus_load, 16
- nhd_plus_query, 18
- nhd_query, 19
- nhdR (nhdR-package), 2
- nhdR-package, 2
- select_point_overlay, 21
- select_poly_overlay, 21