Package 'adklakedata'

November 28, 2025

Title Adirondack Long-Term Lake Data

Version 0.6.4

Maintainer Jeremy Farrell <jeremyfarrell@gmail.com>

Description
Package for the access and distribution of long-term lake datasets from lakes in the Adirondack Park, northern New York state. Includes a wide variety of physical, chemical, and biological parameters from 28 lakes. Data are from multiple collection organizations and have been harmonized in both time and space for ease of reuse.

License MIT + file LICENSE

URL https://github.com/jeremylfarrell/adklakedata

Encoding UTF-8

RoxygenNote 7.3.2

BugReports https://github.com/jeremylfarrell/adklakedata/issues

Depends R (>= 3.5.0)

Imports httr, rappdirs, tibble

Suggests sf, testthat (>= 3.0.0)

NeedsCompilation no

Author Jeremy Farrell [aut, cre],
Luke Winslow [aut],
Taylor Leach [aut],
Tobi Hahn [aut]

Repository CRAN

Date/Publication 2025-11-28 13:00:37 UTC

Contents

`adk_data` *Load ADK Data*

Description

Loads data from locally downloaded CSV files. Run `check_dl_data` before using this function.

Usage

```
adk_data(data_name)
```

Arguments

`data_name`	A string choosing the data to load.

Data name (data_name)	Data Description
chem	Lake Chemistry
crustacean	Crustacean Zooplankton Biomass
meta	Lake-specific metadata (type, location, morphology)
nutrient	Lake Nutrients
phyto	Phytoplankton Biomass Observations
rotifer	Rotifer Zooplankton Biomass
secchi	Lake Secchi Depth Observations
tempdo	Temperature and Dissolved Oxygen Profiles
met	Lake-specific Meterology (air temp, wind, precip, etc)

Examples

```
## Not run:

#grab secchi data and plot it
secchi = adk_data('secchi')
plot(as.POSIXct(secchi$date), secchi$secchi)

## End(Not run)
```

adk_lakes *Load the lake shapefile for ADK*

Description

This function loads the primary geospatial boundary data for the Adirondack parks.

Usage

```
adk_lakes()
```

Examples

```
## Not run:
sites = adk_lakes()

## End(Not run)
```

adk_lake_shapes *Return path to Lake Polygons Shapefile*

Description

Returns the path to the shapefile for the study Lake polygons. The source is a locally stored shapefile that can be used for mapping and analysis.

Usage

```
adk_lake_shapes()
```

Examples

```
## Not run:
#' library(sf)
bl = sf::read_sf(adklakedata::adk_shape())
lakes = sf::read_sf(adklakedata::adk_lake_shapes())
plot(st_geometry(bl))
plot(st_geometry(lakes), add=TRUE, col='blue')

## End(Not run)
```

adk_metadata *Get data table metadata info*

Description

Function to recall metadata about each dataset. Includes units and long-name of parameters. Prints info to console as well as returning text.

Usage

```
adk_metadata(data_name)
```

Arguments

data_name	character name of dataset. See adk_data documentation for dataset names.

Examples

```
## Not run:
#Get chemistry metadata
adk_metadata('chem')

## End(Not run)
```

adk_shape *Return path to Adirondack Park Shapefile*

Description

Returns the path to the shapefile for the Adirondack Park outline (The "Blue Line"). Returns the path to a locally stored shapefile that can be used for mapping and analysis.

Usage

```
adk_shape()
```

Examples

```
## Not run:
#' library(sf)
bl = sf::read_sf(adklakedata::adk_shape())
lakes = sf::read_sf(adklakedata::adk_lake_shapes())
plot(st_geometry(bl))
plot(st_geometry(lakes), add=TRUE, col='blue')

## End(Not run)
```

check_dl_data *Download lake data from internet*

Description

Check that we have local cache of ADK lake data. If it is not locally available, download the data from the internet and prepare it for local use. This only needs to be run once for each install of the package. Note: you will be required to re-download data when a new version of the package is released. This ensures stale data are not being accidentally used.

Usage

```
check_dl_data()
```

check_dl_file *Verify and download data files*

Description

Checks if local data files as defined in master file exist and match MD5 hash. Downloads data if necessary.

Usage

```
check_dl_file(master_file, fname = NULL, md5check = TRUE, dest = local_path())
```

Arguments

master_file	Character path to master file
fname	Character vector of specific file names to check
md5check	boolean
dest	Character path to download destination

local_path *Get local file path*

Description

Data files are locally cached (they are too large to be distributed with the CRAN package). These cached files are stored in your user data directory, or a custom directory set using `set_local_path`.

Usage

```
local_path()
```

Value

Path to local file cache location

Examples

```
# set custom path to local temp directory
set_local_path(tempdir())

#returns current local path directory
local_path()
```

set_local_path *Set custom local file path*

Description

Data files are locally cached (they are too large to be distributed with the CRAN package). These cached files are stored in your user data directory, or a custom directory set using `set_local_path`.

Usage

```
set_local_path(path)
```

Arguments

`path`	Full path to custom folder, will be created if it doesn't exist.

Examples

```
# set custom path to local temp directory
set_local_path(tempdir())
```

Index