Package 'driftR'

June 14, 2018

Type Package

Title Drift Correcting Water Quality Data

Version 1.1.0

Description A tidy implementation of equations that correct for instrumental drift in continuous water quality monitoring data. There are many sources of water quality data including private (ex: YSI instruments) and open source (ex: USGS and NDBC), each of which are susceptible to errors/inaccuracies due to drift. This package allows the user to correct their data using one or two standard reference values in a uniform, reproducible way. The equations implemented are from Hasenmueller (2011) <doi:10.7936/K7N014KS>.

License GPL-3

URL https://github.com/shaughnessyar/driftR

BugReports https://github.com/shaughnessyar/driftR/issues

Encoding UTF-8

LazyData true

Imports dplyr, glue, janitor (>= 1.0.0), lubridate, magrittr, readr, readxl, rlang, stringr, tibble

RoxygenNote 6.0.1

Suggests covr, knitr, testthat, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Andrew Shaughnessy [aut, cre], Christopher Prener [aut], Elizabeth Hasenmueller [aut]

Maintainer Andrew Shaughnessy <andrew.shaughnessy@slu.edu>

Repository CRAN

Date/Publication 2018-06-13 22:03:03 UTC

R topics documented:

`driftR` *driftR: Drift Correcting Water Quality Data*

Description

There are many sources of water quality data including instruments (ex: YSI instruments) and open source data sets (ex: USGS and NDBC), all of which are susceptible to errors/inaccuracies due to drift. `driftR` provides a grammar for cleaning and correcting these data in a "tidy", reproducible manner.

Details

The `driftR` package implements a series of equations used in Dr. Elizabeth Hasenmueller's hydrology and geochemistry research. These equations correct continuous water quality monitoring data for incremental drift that occurs over time. There are two forms of corrections included in the package - a one-point calibration and a two-point calibration. One-point and two-point calibration values are suited for different types of measurements. The package is currently written for the easiest use with YSI Sonde products.

There are four key verbs that are introduced in `driftR`:

- *read*: The `dr_readSonde` function imports and properly formats output from YSI Sonde instrument
- *factor*: The `dr_factor` function calculates factors based on the time of the observation and the total amount of time that the instrument had been deployed. They are used in the equations for both the one-point and two-point drift corrections.
- *correct*: The `dr_correctOne` and `dr_correctTwo` functions take both the factors and standard values as parameters for calculating drift corrected versions of specific measurements.
- *drop*: The `dr_drop` function allows for removing erroneous observations from both the head and the tail of the data.

Tidy Evaluation

driftR makes use of tidy evaluation and the pronoun .data, meaning that variable references may be either quoted or unquoted (i.e. bare). This also means that driftR works seamlessly with magrittr pipe operators.

Author(s)

Maintainer: Andrew Shaughnessy andrew.shaughnessy@slu.edu

Authors:

- Christopher Prener, Ph.D. chris.prener@slu.edu
- Elizabeth Hasenmueller, Ph.D. elizabeth.hasenmueller@slu.edu

See Also

Useful links:

- Package Website and Documentation
- Source Code on GitHub
- Bug Reports and Feature Requests

---

dr_correctOne *One-point drift correction*

Description

A wrapper around dplyr::mutate() that creates a corrected value for each observation of the specified variable based on one data point.

Usage

```
dr_correctOne(.data, sourceVar, cleanVar, calVal, calStd, factorVar)
```

Arguments

.data	A tbl
sourceVar	Name of variable to correct
cleanVar	New variable name for corrected data
calVal	A numeric value; the value that the instrument was actually reading for the parameter
calStd	A numeric value; the value that the instrument should have been reading for that standard; i.e. the standard value
factorVar	Name of variable generated using dr_factor

Details

This function takes the raw data from the water-quality instrument, utilizes the values generated from dr_factor and returns data that accounts for drift over time. This is done via a one-point calibration standard, which it typical for specific conductivity, dissolved oxygen, and turbidity.

Value

An object of the same class as .data with the new corrected variable added to the other data in .data.

See Also

dr_factor for correction factor creation, dr_correctTwo for the two-point drift correction

Examples

```
testData <- data.frame(
  Date = c("9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015"),
   Time = c("12:10:49", "12:15:50", "12:20:51", "12:25:51", "12:30:51", "12:35:51"),
   Temp = c(14.76, 14.64, 14.57, 14.51, 14.50, 14.63),
   SpCond = c(0.754, 0.750, 0.750, 0.749, 0.749, 0.749),
   corrFac = c(0.0000000, 0.2003995, 0.4007989, 0.6005326, 0.8002663, 1.0000000),
   stringsAsFactors = FALSE
 )

dr_correctOne(testData, sourceVar = SpCond, cleanVar = SpCond_Corr,
    calVal = 1.05, calStd = 1, factorVar = corrFac)
```

---

dr_correctTwo *Two-point drift correction*

Description

A wrapper around dplyr::mutate() that creates a corrected value for each observation of the specified variable based on two data points.

Usage

```
dr_correctTwo(.data, sourceVar, cleanVar, calValLow, calStdLow,
                     calValHigh, calStdHigh, factorVar)
```

Arguments

.data	A tbl
sourceVar	Name of variable to correct
cleanVar	New variable name for corrected data

calValLow	A numeric value; the number that the instrument was actually reading for the low standard
calStdLow	A numeric value; the number that the instrument should have been reading for that standard; i.e. the low standard value
calValHigh	A numeric value; the number that the instrument was actually reading for the high standard
calStdHigh	A numeric value; the number that the instrument should have been reading for that standard; i.e. the high standard value
factorVar	Name of variable generated using dr_factor

Details

This command takes the raw data from the water-quality instrument, utilizes the values generated from dr_factor and returns data that accounts for drift over time. This is done via a two-point calibration standard, which it typical for pH and chloride.

Value

An object of the same class as .data with the new corrected variable added to the other data in .data.

See Also

dr_factor for correction factor creation, dr_correctOne for the two-point drift correction

Examples

```
testData <- data.frame(
  Date = c("9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015"),
   Time = c("12:10:49", "12:15:50", "12:20:51", "12:25:51", "12:30:51", "12:35:51"),
   Temp = c(14.76, 14.64, 14.57, 14.51, 14.50, 14.63),
   pH = c(7.18, 7.14, 7.14, 7.13, 7.13, 7.13),
   corrFac = c(0.0000000, 0.2003995, 0.4007989, 0.6005326, 0.8002663, 1.0000000),
   stringsAsFactors = FALSE
 )

 dr_correctTwo(testData, sourceVar = pH, cleanVar = pH_Corr, calValLow = 7.01, calStdLow = 7,
     calValHigh = 11.8, calStdHigh =  10, factorVar = corrFac)
```

---

dr_drop *Dropping observations from the monitoring period*

Description

dr_drop() includes three approaches for removing observations from the monitoring period. Observations may be removed by specifying the number to remove from the head and/or the tail of the observation. They may also be removed by specifying one or two timepoints in the data where problematic observations begin, end, or fall between. Finally observations may be removed based on a problematic sensor value or range of values using an expression.

Usage

```
dr_drop(.data, head = NULL, tail = NULL, dateVar = NULL, timeVar = NULL, from = NULL,
    to = NULL, tz = NULL, exp)
```

Arguments

.data	A tbl
head	An integer >= 1 specifying the number of rows to be removed from the top of .data (or NULL)
tail	An integer >= 1 specifying the number of rows to be removed from the bottom of .data (or NULL)
dateVar	Name of variable containing date data
timeVar	Name of variable containing time data
from	Beginning date and (optionally) time to remove observations
to	End date and (optionally) time to remove observations
tz	String name of timezone, defaults to system's timezone
exp	Unquoted expression

Details

When taking the instrument out of the water, there are often several observations that pass before the run can be downloaded. Additionally, once the instrument is in the water, it often takes about 30 minutes for the sensors to equilibrate. This function allows you to drop observations from the bottom and top of the data set for each of those issues respectively. This function also provides approaches for removing observations from the middle of the data set.

Value

An object of the same class as .data with specified observations removed.

Examples

```
testData <- data.frame(
  Date = c("9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/19/2015", "9/21/2015"),
   Time = c("12:10:49", "12:15:50", "12:20:51", "12:25:51", "12:30:51", "12:35:51"),
   Temp = c(14.76, 14.64, 14.57, 14.51, 14.50, 14.63),
   SpCond = c(0.754, 0.750, 0.750, 0.749, 0.749, 0.749),
   stringsAsFactors = FALSE
 )
```

```
dr_drop(testData, head = 2)
dr_drop(testData, tail = 1)
dr_drop(testData, head = 2, tail = 1)
dr_drop(testData, dateVar = Date, timeVar = Time, from = "9/19/2015")
dr_drop(testData, dateVar = Date, timeVar = Time, from = "9/18/2015 12:25:51")
dr_drop(testData, dateVar = Date, timeVar = Time, to = "9/19/2015")
dr_drop(testData, dateVar = Date, timeVar = Time, to = "9/18/2015 12:25:51")
dr_drop(testData, dateVar = Date, timeVar = Time, from = "9/18/2015 12:25:51",
    to = "9/19/2015 12:30:51")
dr_drop(testData, dateVar = Date, timeVar = Time, from = "9/18/2015 12:00", to = "9/19/2015 13:00")
dr_drop(testData, exp = Temp > 14.7)
```

`dr_factor` *Creating correction factors*

Description

A wrapper around `dplyr::mutate()` that creates a correction factor for each observation.

Usage

```
dr_factor(.data, corrFactor, dateVar, timeVar, tz = NULL,
    format = c("MDY", "YMD"), keepDateTime = TRUE)
```

Arguments

`.data`	A tbl
`corrFactor`	New variable name for correction factor data
`dateVar`	Name of variable containing date data
`timeVar`	Name of variable containing time data
`tz`	String name of timezone, defaults to system's timezone
`format`	Either "MDY" or "YMD" for `dateVar` - ***deprecated as of*** `driftR` ***v1.1***
`keepDateTime`	A logical statement to keep an intermediate dateTime variable

Details

Correction factors are calculated based on the time of the observation and the total amount of time that the instrument had been deployed. They are used in the equations for both the one-point and two-point drift corrections.

Value

An object of the same class as `.data` with the new correction factor variable added to the other data in `.data` as well as a dateTime variable if keepDateTime = TRUE.

See Also

dr_correctOne for correction factor creation, dr_correctTwo for the two-point drift correction

Examples

```
testData <- data.frame(
  Date = c("9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015"),
   Time = c("12:10:49", "12:15:50", "12:20:51", "12:25:51", "12:30:51", "12:35:51"),
   Temp = c(14.76, 14.64, 14.57, 14.51, 14.50, 14.63),
   SpCond = c(0.754, 0.750, 0.750, 0.749, 0.749, 0.749),
   stringsAsFactors = FALSE
 )

dr_factor(testData, corrFactor = corrFac, dateVar = Date, timeVar = Time, keepDateTime = TRUE)
```

---

dr_read *Import raw data from water quality instrument*

Description

This function imports the raw data from a YSI Sonde 6600 and EXO2 as well as an Onset U24 Conductivity Logger and formats the data set as a tibble. If defineVar is set to TRUE (the default option), units of measurement will not be included in the first observation.

Usage

```
dr_read(file, instrument, defineVar = TRUE, cleanVar = TRUE, case)
```

Arguments

file	The name of the file which the data are to be read from. Each row of the table appears as one line of the file. If it does not contain an absolute path, the file name is relative to the current working directory.
instrument	Which instruments the data was colected with. Options currently include "Sonde", "EXO", and "HOBO".
defineVar	Logical scalar that determines if the units of measurement are included in the first observation. If they are included, all vectors will be read in as character.
cleanVar	Logical scalar. Should the variable names be cleaned to remove spaces and special characters? This is implemented using the janitor package's clean_names function.
case	Case to convert variable names to, see janitor::clean_names for details

Value

A tibble with the formatted data and the variable types defined if defineVar = TRUE

Examples

```
## Not run:
dr_read("data.csv", instrument = Sonde, defineVar = TRUE, cleanVar = TRUE, case = "snake")
dr_read("data.csv", instrument = EXO, defineVar = TRUE, cleanVar = TRUE, case = "lower_camel")
dr_read("data.csv", instrument = HOBO, defineVar = TRUE, cleanVar = TRUE, case = "all_caps")

## End(Not run)
```

---

dr_readSonde *Import raw data from a YSI Multivariable V2 Sonde*

Description

This function imports the raw data from a YSI Sonde and formats the data set as a tibble. If `defineVar` is set to `TRUE` (the default option), units of measurement will not be included in the first observation.

Usage

```
dr_readSonde(file, defineVar = TRUE)
```

Arguments

`file`	The name of the file which the data are to be read from. Each row of the table appears as one line of the file. If it does not contain an absolute path, the file name is relative to the current working directory.
`defineVar`	Logical scalar that determines if the units of measurement are included in the first observation. If they are included, all vectors will be read in as character.

Value

A tibble with the formatted data and the variable types defined if `defineVar = TRUE`

Examples

```
## Not run:
dr_readSonde("data.csv")
dr_readSonde("data.csv", defineVar = TRUE)

## End(Not run)
```

dr_replace *Replacing problematic observations from the monitoring period*

Description

dr_replace() includes two approaches for identifying problematic observations for specific measurements that should be recoded as missing values (NA).

Usage

```
dr_replace(.data, sourceVar, cleanVar = NULL, overwrite = FALSE, dateVar = NULL,
    timeVar = NULL, from = NULL, to = NULL, tz = NULL, exp)
```

Arguments

.data	A tbl
sourceVar	Name of variable to replace missing values in
cleanVar	New variable name for cleaned data
overwrite	A logical scalar. Should the current variable be overwritten instead of creating a new variable?
dateVar	Name of variable containing date data
timeVar	Name of variable containing time data
from	Beginning date and (optionally) time to remove observations
to	End date and (optionally) time to remove observations
tz	String name of timezone, defaults to system's timezone
exp	Unquoted expression

Details

During monitoring, a sensor malfunction may impact only a single element of a given reading. Removing the entire observation may therefore be imprudent. dr_replace() provides two methods for identifying these values and declaring them as missing. Values can be identified by specifying one or two timepoints in the data where problematic measurements begin, end, or fall between. Values can also be identified based on a problematic sensor value or range of values using an expression.

Value

An object of the same class as .data with specified observations recoded as missing.

Examples

```
testData <- data.frame(
  Date = c("9/18/2015", "9/18/2015", "9/18/2015", "9/18/2015", "9/19/2015", "9/21/2015"),
   Time = c("12:10:49", "12:15:50", "12:20:51", "12:25:51", "12:30:51", "12:35:51"),
   Temp = c(14.76, 14.64, 14.57, 14.51, 14.50, 14.63),
   SpCond = c(0.754, 0.750, 0.750, 0.749, 0.749, 0.749),
   stringsAsFactors = FALSE
 )

 dr_replace(testData, sourceVar = Temp, dateVar = Date, timeVar = Time,
     from = "2015-09-19 12:30:51", to = "2015-09-21 12:35:51")
 dr_replace(testData, sourceVar = Temp, dateVar = Date, timeVar = Time,
     from = "2015-09-19", to = "2015-09-21")
 dr_replace(testData, sourceVar = Temp, dateVar = Date, timeVar = Time, from = "2015-09-19")
 dr_replace(testData, sourceVar = Temp, dateVar = Date, timeVar = Time, to = "2015-09-19")
 dr_replace(testData, sourceVar = Temp, cleanVar = temp2, dateVar = Date, timeVar = Time,
     to = "09/19/2015 12:35:51")
 dr_replace(testData, sourceVar = Temp, overwrite = TRUE, exp = Temp > 14.75)
```

---

sondeCal *Calibration values for water quality monitoring data correction*

Description

A data set containing correction values used in practice drift corrections

Usage

```
data(sondeCal)
```

Format

a dataframe with 7 rows and 3 variables

Parameter The parameter that you will be correcting for

'Cal Standard' The value that the instrument should be reading for the given parameter

'Cal Value' The value that the instrument is actually reading for the given parameter

Source

Saint Louis University Geochemistry Lab

Examples

```
str(sondeCal)
head(sondeCal)
```

sondeClean *Corrected water quality monitoring data*

Description

A data set containing corrected measurements from a YSI Sonde 6600

Usage

```
data(sondeClean)
```

Format

a dataframe with 1528 rows and 16 variables

Date Date of measurement

Time Time of measurement

Temp Temperature in degrees C

SpCond Specific conductivity raw in mS/cm

SpCond_Corr Specific conductivity corrected in mS/cm

pH pH raw

pH_Corr pH corrected

pHmV Potential reading from pH sensor in mV

Chloride Chloride raw in mg/L

Chloride_Corr Chloride corrected in mg/L

AmmoniumN Ammonium-Nitrogen in mg/L

NitrateN Nitrate-Nitrogen in mg/L

'Turbidity.' Turbidity raw in NTU

Turbidity_Corr Turbidity corrected in NTU

DO Dissolved Oxygen raw in % sat

DO_Corr Dissolved Oxygen corrected in % sat

corrFactors A list of correction factors based on time spent in the water

Source

Saint Louis University Geochemistry Lab

Examples

```
str(sondeClean)
head(sondeClean)
```

sondeRaw *Uncorrected water quality monitoring data*

Description

A data set containing un-corrected measurements from a YSI Sonde 6600

Usage

```
data(sondeRaw)
```

Format

a dataframe with 1528 rows and 11 variables

Date Date of measurement

Time Time of measurement

Temp Temperature in degrees C

SpCond Specific conductivity in mS/cm

pH pH

pHmV Potential reading from pH sensor in mV

Chloride Chloride in mg/L

AmmoniumN Ammonium-Nitrogen in mg/L

NitrateN Nitrate-Nitrogen in mg/L

'Turbidity+' Turbidity in NTU

DO Dissolved Oxygen in % sat

Source

Saint Louis University Geochemistry Lab

Examples

```
str(sondeRaw)
head(sondeRaw)
```

`%nin%` *Not In Operator*

Description

Provides the compliment to the base R `%in%` operator. Included here instead of via import due to stability issues with the source package, `Hmsic`, during original package development in October, 2017. Used under terms of `Hmisc`'s GPL-3 License.

Usage

```
x %nin% y
```

Arguments

`x`	vector or `NULL`: the values to be matched
`y`	vector or `NULL`: the values to be matched against

Source

`Hmsic`

Examples

```
x <- 2
y <- 2
z <- 3

x %in% y
x %nin% y

x %in% z
x %nin% z
```

Index