

Package: wqbc (via r-universe)

September 17, 2024

Type Package

Title Tidy Water Quality Data and Calculate Thresholds for British Columbia

Version 0.3.1.9003

Description Tidies water quality data and calculates water quality thresholds for British Columbia.

URL <https://github.com/bcgov/wqbc>

BugReports <https://github.com/bcgov/wqbc/issues>

License Apache License (== 2.0) | file LICENSE

LazyData true

Depends R (>= 3.4)

Imports chk, plyr, dplyr, rlang, zyp, magrittr, NADA, purrr, grDevices, stringr, stats, tibble, tidyr, mgcv, lubridate, lifecycle

Suggests testthat, knitr, ggplot2, rmarkdown, xtable, rems (>= 0.5.1), canwqdata, Kendall, covr

Remotes bcgov/remdata, bcgov/canwqdata

VignetteBuilder knitr

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

Encoding UTF-8

RdMacros lifecycle

Repository <https://poissonconsulting.r-universe.dev>

RemoteUrl <https://github.com/poissonconsulting/wqbc>

RemoteRef HEAD

RemoteSha d6172fc71da74d7fe7d5f55ec94495b55c2f44e9

Contents

calc_limits	2
ccme	4
clean_wqdata	5
codes	6
compress_ems_codes	7
convert_values	7
dummy	8
ems_codes	9
error	9
expand_ems_codes	10
fraser	10
geomean1	11
limits	12
lookup_codes	12
lookup_limits	13
lookup_units	14
lookup_use	14
lookup_variables	15
plot_timeseries	15
set_non_detects	16
site_limits	17
standardize_wqdata	17
stations	18
substitute_units	19
substitute_variables	19
summarise_for_trends	20
summarise_wqdata	21
test_trends	22
tidy_ec_data	23
tidy_ems_data	24
vmv_codes	25
vmv_ems	25
yuepilon	26
Index	28

calc_limits	<i>Calculate Limits</i>
-------------	-------------------------

Description

Calculates the approved "short" or "long"-term or the "long-daily" upper water quality thresholds for freshwater life in British Columbia. The water quality data is automatically cleaned using [clean_wqdata](#) prior to calculating the limits to ensure: all variables are recognised, all values are non-negative and in the standard units, divergent replicates are filtered and all remaining replicates are averaged. Only limits whose conditions are met are returned.

Usage

```
calc_limits(
  x,
  by = NULL,
  term = "long",
  dates = NULL,
  keep_limits = TRUE,
  delete_outliers = FALSE,
  estimate_variables = FALSE,
  clean = TRUE,
  limits = wqbc::limits,
  messages = getOption("wqbc.messages", default = TRUE),
  use = "Freshwater Life"
)
```

Arguments

x	A data.frame of water quality readings to calculate the limits for.
by	A optional character vector of the columns in x to calculate the limits by.
term	A string indicating whether to calculate the "long" or "short"-term or "long-daily" limits.
dates	A optional date vector indicating the start of 30 day long-term periods.
keep_limits	A flag indicating whether to keep values with user supplied upper or lower limits.
delete_outliers	A flag indicating whether to delete outliers or merely flag them.
estimate_variables	A flag indicating whether to estimate total hardness, total chloride and pH for all dates.
clean	Should the data be run through clean_wqdata before calculating limits? Default TRUE
limits	A data frame of the limits table to use.
messages	A flag indicating whether to print messages.
use	A string indicating the Use.

Details

If a limit depends on another variable such as pH, Total Chloride, or Total Hardness and no value was recorded for the date of interest then the pH, Total Chloride or Total Hardness value is assumed to be the average recorded value over the 30 day period. The one exception is if `estimate_variables = TRUE` in which case a parametric model is used to predict the pH, Total Chloride and Total Hardness for all dates with a value of any variable. Existing values are replaced. If, in every year, there are less than 12 pH/Total Chloride/Total Hardness then an average value is taken. Otherwise, if there is only one year with 12 or more values a simple seasonal smoother is used. If there is two years with 12 or more values then a seasonal smoother with a trend is fitted. Otherwise a model with trend and a dynamic seasonal component is fitted.

When considering long-term limits there must be at least 5 values spanning 21 days. As replicates are averaged prior to calculating the limits each of the 5 values must be on a separate day. The first 30 day period begin at the date of the first reading while the next 30 day period starts at the date of the first reading after the previous period and so on. The only exception to this is if the user provides dates in which case each period extends for 30 days or until a provided date is reached. It is important to note that the averaging of conditional variables, the 5 in 30 rule and the assignment of 30 day periods occurs independently for all combination of factor levels in the columns specified by by.

If the user wishes to consider the long-term thresholds without the above requirements that there are at least 5 values spanning 21 days etc then they should set term = "long-daily"

See Also

[clean_wqdata](#) and [lookup_limits](#)

Examples

```
## Not run:
demo(fraser)

## End(Not run)
```

ccme

CCME Water Quality Index User's Manual Example Data

Description

The Canadian Council of Ministers of the Environment (CCME) Water Quality Index 1.0 User's Manual example dataset in tidy format.

Usage

```
ccme
```

Format

A data frame with 120 rows and 7 columns:

Date The date of the reading.

Variable The name of the variable.

Value The value of the reading.

DetectionLimit The detection limit.

LowerLimit The minimum permitted value.

UpperLimit The maximum permitted value.

Units The units of the value, detection limit and lower and upper limits.

clean_wqdata	<i>Clean Water Quality Data</i>
--------------	---------------------------------

Description

Cleans water quality data. After standardization using `standardize_wqdata` replicates (two or more readings for the same variable on the same date) are averaged using the mean function. Readings for the same variable on the same date but at different levels of the columns specified in by are not considered replicates. The `clean_wqdata` function is automatically called by `calc_limits` prior to calculating limits.

Usage

```
clean_wqdata(
  x,
  by = NULL,
  max_cv = Inf,
  sds = 10,
  ignore_undetected = TRUE,
  large_only = TRUE,
  delete_outliers = FALSE,
  remove_blanks = FALSE,
  messages = getOption("wqbc.messages", default = TRUE),
  FUN = mean
)
```

Arguments

<code>x</code>	The data.frame to clean.
<code>by</code>	A character vector of the columns in <code>x</code> to perform the cleaning by. If you have multiple stations specify the column name that contains the station IDs.
<code>max_cv</code>	A number indicating the maximum permitted coefficient of variation for replicates.
<code>sds</code>	The number of standard deviations above which a value is considered an outlier.
<code>ignore_undetected</code>	A flag indicating whether to ignore undetected values when calculating the average deviation and identifying outliers.
<code>large_only</code>	A flag indicating whether only large values which exceed the <code>sds</code> should be identified as outliers.
<code>delete_outliers</code>	A flag indicating whether to delete outliers or merely flag them.
<code>remove_blanks</code>	Should blanks be removed? Blanks are assumed to be denoted by a value of "Blank..." in the <code>SAMPLE_CLASS</code> column. Default FALSE
<code>messages</code>	A flag indicating whether to print messages.
<code>FUN</code>	The function to use for summaries, e.g. median, mean, or max. Default mean

Details

If there are three or more replicates with a coefficient of variation (CV) in exceedance of `max_cv` then the replicates with the highest absolute deviation is dropped until the CV is less than or equal to `max_cv` or only two values remain. By default all values are averaged.

A `max_cv` value of 1.29 is exceeded by two zero and one positive value (CV = 1.73) or by two identical positive values and a third value an order or magnitude greater (CV = 1.30). It is not exceed by one zero and two identical positive values (CV = 0.87).

See Also

[calc_limits](#) and [standardize_wqdata](#)

Examples

```
clean_wqdata(wqbc::dummy, messages = TRUE)
```

codes

Water Quality Parameter Codes and Units for British Columbia

Description

The standard variables and codes recognised by the `wqbc` package with their standard units and the R function to use when averaging multiple samples.

Usage

```
codes
```

Format

A data frame with 4 variables:

Variable The name of the variable.

Code The EMS code in expanded form.

Units The standard units for the variable.

EC_Code The Variable Code in the Environment Canada data.

See Also

[calc_limits](#)

compress_ems_codes *Compress EMS Codes*

Description

Compresses EMS codes by removing EMS_ from start and replacing all '_' with '-'. This function is provided because wqbc stored EMS codes in expanded form.

Usage

```
compress_ems_codes(x)
```

Arguments

x A character vector of codes to compress.

See Also

[expand_ems_codes](#)

Examples

```
compress_ems_codes(c("EMS_0014", "EMS_KR-P", "0-15"))
```

convert_values *Convert values to different units*

Description

Convert values to different units

Usage

```
convert_values(x, from, to, messages)
```

Arguments

x a numeric vector of values to convert
 from units to convert from
 to units to convert to
 messages should messages be printed when

Details

Currently supported units for from and to are: c("ng/L", "ug/L", "mg/L", "g/L", "kg/L", "pH", "degC", "C", "CFU/dL", "MPN/dL", "CFU/100mL", "MPN/100mL", "CFU/g", "MPN/g", "CFU/mL", "MPN/mL", "Col.unit", "Rel", "NTU", "m", "uS/cm")

Value

a numeric vector of the converted values

Examples

```
convert_values(1, "ug/L", "mg/L", messages = FALSE)

df <- data.frame(
  value = c(1.256, 5400000, 12300, .00098),
  units = c("mg/L", "ng/L", "ug/L", "g/L"),
  stringsAsFactors = FALSE
)
df

df$units_mg_L <- convert_values(df$value, from = df$units, to = "mg/L", messages = FALSE)
df
```

dummy

Dummy Water Quality Data

Description

A dummy data set to illustrate various data cleaning functions.

Usage

```
dummy
```

Format

A data frame with 4 columns:

Date The date of the reading.

Variable The name of the variable.

Value The value of the reading.

Units The units of the value.

See Also

[clean_wqdata](#)

Examples

```
demo(dummy)
```

`ems_codes`*Water Quality Parameter EMS Names and Codes for British Columbia*

Description

The standard variables and codes stored in the EMS database.

Usage`ems_codes`**Format**

A tibble with 4 variables:

Variable The name of the variable.

Code The EMS code.

`error`*Error*

Description

Throws an error without the call as part of the error message.

Usage`error(...)`**Arguments**

... zero or more objects which can be coerced to character (and which are pasted together with no separator) or a single condition object.

See Also`base::stop`

expand_ems_codes	<i>Expand EMS Codes</i>
------------------	-------------------------

Description

Expands EMS codes by adding EMS_ to start if absent and replacing all '-' with '_'. This function is provided because wqbc stored EMS codes in expanded form.

Usage

```
expand_ems_codes(x)
```

Arguments

x	A character vector of codes to expand
---	---------------------------------------

See Also

[compress_ems_codes](#)

Examples

```
expand_ems_codes(c("0014", "KR-P", "0_15", "EMS_ZN_T"))
```

fraser	<i>Fraser River Basin Long-term Water Quality Monitoring 1979-Present</i>
--------	---

Description

Surface freshwater quality monitoring in the Fraser River Basin is carried out under the Canada-British Columbia Water Quality Monitoring Agreement. Monitoring is conducted to assess water quality status and long-term trends, detect emerging issues, establish water quality guidelines and track the effectiveness of remedial measures and regulatory decisions.

Usage

```
fraser
```

Format

A data frame with 8 columns:

SiteID The unique water quality station number.

Date The date of the reading.

Variable The name of the variable.

Value The value of the reading.

Units The units of the value.

Site The full name of the station.

Lat The latitude of the station in decimal degrees.

Long The longitude of the station in decimal degrees.

Source

<http://open.canada.ca/data/en/dataset/9ec91c92-22f8-4520-8b2c-0f1cce663e18>

Examples

```
## Not run:
demo(fraser)

## End(Not run)
```

geomean1

Geometric Mean Plus-Minus 1

Description

Calculates the geometric mean by adding 1 before logging and subtracting 1 before exponentiating so that provides results even with zero counts. Not used by any wqbc functions but provided as may be helpful if averaging bacterial counts.

Usage

```
geomean1(x, na.rm = FALSE)
```

Arguments

x A numeric vector of non-negative numbers.

na.rm A flag indicating whether to remove missing values.

Examples

```
mean(0:9)
geomean1(0:9)
```

 limits

Water Quality Limits for British Columbia

Description

The short and long term water quality limits for British Columbia recognised by the wqbc package.

Usage

```
limits
```

Format

A data frame with 6 variables:

Variable The name of the variable.

Use The name of the Use.

Term The term of the limit i.e. "Short" versus "Long".

Condition A logical R expression to test for the required condition.

UpperLimit The upper limit or an R expression defining the upper limit.

Units The units of the upper limit.

Statistic R function to calculate statistic of value.

See Also

[calc_limits](#)

 lookup_codes

Lookup Codes

Description

Returns compressed recognised water quality EMS codes. If `variables = NULL` the function returns all recognised codes. Otherwise it first substitutes the provided variables for recognised variables using [substitute_variables](#) and then looks up the matching codes from [codes](#).

Usage

```
lookup_codes(
  variables = NULL,
  messages = getOption("wqbc.messages", default = TRUE)
)
```

Arguments

variables An optional character vector of variables to lookup codes.
 messages A flag indicating whether to print messages.

See Also

[lookup_limits](#) and [expand_ems_codes](#)

Examples

```
lookup_codes()
lookup_codes(c("Aluminum", "Arsenic Total", "Boron Something", "Kryptonite"),
  messages = TRUE
)
```

lookup_limits	<i>Lookup Limits</i>
---------------	----------------------

Description

Looks up the long or short-term water quality limits for BC. If the limits depend on on the pH, total hardness (CaCO₃), total chloride or the concentration of methyl mercury and site specific values are not provided then the dependent limits are returned as missing values.

Usage

```
lookup_limits(
  ph = NULL,
  hardness = NULL,
  chloride = NULL,
  methyl_mercury = NULL,
  term = "long",
  use = "Freshwater Life"
)
```

Arguments

ph A number indicating the pH in pH units at the site of interest.
 hardness A number indicating the total hardness (CaCO₃) in mg/L at the site of interest.
 chloride A number indicating the total chloride concentration in mg/L at the site of interest.
 methyl_mercury A number indicating the total concentration of methyl mercury in ug/L at the site of interest.
 term A string indicating whether to lookup the "long" or "short"-term limits.
 use A string indicating the Use.

See Also[calc_limits](#)**Examples**

```
lookup_limits(ph = 8, hardness = 100, chloride = 50, methyl_mercury = 2)
lookup_limits(term = "short")
```

lookup_units	<i>Lookup Units</i>
--------------	---------------------

Description

Returns a character vector of the recognised units.

Usage

```
lookup_units()
```

See Also[lookup_limits](#)**Examples**

```
lookup_units()
```

lookup_use	<i>Lookup Use</i>
------------	-------------------

Description

Returns a character vector of the recognised uses.

Usage

```
lookup_use()
```

See Also[lookup_limits](#)**Examples**

```
lookup_use()
```

lookup_variables	<i>Lookup Variables</i>
------------------	-------------------------

Description

Returns recognised water quality variables. If `codes = NULL` the function returns all recognised variable names. Otherwise it looks up the matching variables from [codes](#). Whether or not the codes are compressed or expanded is unimportant.

Usage

```
lookup_variables(  
  codes = NULL,  
  messages = getOption("wqbc.messages", default = TRUE)  
)
```

Arguments

<code>codes</code>	An optional character vector of codes to look up variables.
<code>messages</code>	A flag indicating whether to print messages.

See Also

[lookup_limits](#) and [expand_ems_codes](#)

Examples

```
lookup_variables()  
lookup_variables(c("AL-D", "EMS_AS_T", "B--T", "KRYP"), messages = TRUE)
```

plot_timeseries	<i>Plot Time Series Data</i>
-----------------	------------------------------

Description

If `by = NULL` `plot_timeseries` returns a `ggplot` object. Otherwise it returns a list of `ggplot` objects.

Usage

```
plot_timeseries(  
  data,  
  by = NULL,  
  y0 = TRUE,  
  size = 1,  
  messages = getOption("wqbc.messages", default = TRUE)  
)
```

Arguments

data	A data frame of the data to plot.
by	A character vector of the columns to plot the time series by.
y0	A flag indicating whether to expand the y-axis limits to include 0.
size	A number of the point size.
messages	A flag indicating whether to print messages.

Examples

```
plot_timeseries(ccme[ccme$Variable == "As", ])
plot_timeseries(ccme, by = "Variable")
```

set_non_detects	<i>Set value for 'non-detects'</i>
-----------------	------------------------------------

Description

Set a value where the actual value of a measurement is less than the method detection limit (MDL)

Usage

```
set_non_detects(
  value,
  mdl_flag = NULL,
  mdl_value = NULL,
  mdl_action = c("zero", "mdl", "half", "na")
)
```

Arguments

value	a numeric vector of measured values
mdl_flag	a character vector the same length as value that has a "flag" (assumed to be "<") for values that are below the MDL
mdl_value	a numeric vector the same length as value that contains the MDL values.
mdl_action	What to do with values below the detection limit. Options are "zero" (set the value to 0; the default), # "half" (set the value to half the MDL), "mdl" (set the value to equal to the MDL), or "na" (set the value to NA).

Details

You must supply either `mdl_flag` or `mdl_value`, or both. When only `mdl_flag` is supplied, it is assumed that the original value has been set to the MDL, and will be adjusted according to the `mdl_action`. When only `mdl_value` is supplied then any value less than that will be adjusted appropriately using the corresponding `mdl_value`. When both `mdl_flag` and `mdl_value` are supplied, any value with a corresponding `<` in the `mdl_flag` vector will be adjusted appropriately using the corresponding `mdl_value`.

Value

a numeric vector the same length as value with non-detects adjusted accordingly

site_limits	<i>Example Site Specific Limits</i>
-------------	-------------------------------------

Description

The standard variables, VMV codes and Variable Codes provided by Environment Canada.

Usage

```
site_limits
```

Format

A tibble

Station_Name The name of the station as a character.

Variable The variable name as character.

Term The term ('Short' or 'Long') as a character.

Condition The condition as a character.

UpperLimit The upper limit as a character.

Units The units as a character.

Code The EMS codes as a character.

EMS_ID The station EMS ID as a character.

Use The use name as character.

standardize_wqdata	<i>Standardize Water Quality Data</i>
--------------------	---------------------------------------

Description

Standardizes a water quality data set using [substitute_variables](#) and [substitute_units](#) so that all remaining values have the recognised codes and variables in [codes](#) and the standard units. If column Code is present then a Variable column is created using [lookup_variables](#). The standardize_wqdata function is called by clean_wqdata prior to cleaning.

Usage

```
standardize_wqdata(
  x,
  strict = TRUE,
  messages = getOption("wqbc.messages", default = TRUE)
)
```

Arguments

<code>x</code>	A data.frame to standardize.
<code>strict</code>	A flag indicating whether to require all words in a recognised variable name to be present in <code>x</code> (<code>strict = TRUE</code>) or only the first one (<code>strict = FALSE</code>) [Soft-deprecated].
<code>messages</code>	A flag indicating whether to print messages.

See Also

[clean_wqdata](#)

Examples

```
standardize_wqdata(wqbc::dummy, messages = TRUE)
```

stations

Water Quality Stations for British Columbia

Description

The water quality stations for British Columbia with their coordinates.

Usage

```
stations
```

Format

A tibble with 4 variables:

EMS_ID The station ID (chr).

Station_Name The EMS name of the station (chr).

Latitude The station latitude in decimal degrees (dbl).

Longitude The station longitude in decimal degrees (dbl).

substitute_units	<i>Substitute Units</i>
------------------	-------------------------

Description

Substitutes provided unit names for recognised units. Before matching all spaces and "units" or "UNITS" are removed. The case is not important. Where there are no matches missing values are returned.

Usage

```
substitute_units(x, messages = getOption("wqbc.messages", default = TRUE))
```

Arguments

x	The character vector of units to substitute.
messages	A flag indicating whether to print messages.

See Also

[substitute_variables](#)

Examples

```
substitute_units(c("mg/L", "MG/L", "mg /L ", "Kg/l", "gk1"), messages = TRUE)
```

substitute_variables	<i>Substitute Variables</i>
----------------------	-----------------------------

Description

Substitutes provided variable names for recognised names. The case is not important. Where there are no matches missing values are returned. When `strict = TRUE` all words in a recognised variable must be present in `x` but when `strict = FALSE` (soft-deprecated) the only requirement is that the first word is present. When `strict = FALSE` recognised variables with the same first word such as "Iron Dissolved" and "Iron Total" are excluded from matches. In both cases the only requirement is that all words or just the first word are present in `x`. The order of the words does not matter nor does the presence of other words. This means that a value such as "Total Fluoride Hardness" matches two recognised variables which causes an error. The code also considers Aluminium to be a match with Aluminum.

Usage

```
substitute_variables(  
  x,  
  strict = TRUE,  
  messages = getOption("wqbc.messages", default = TRUE)  
)
```

Arguments

x	A character vector of variable names to substitute.
strict	A flag indicating whether to require all words in a recognised variable name to be present in x (strict = TRUE) or only the first one (strict = FALSE) [Soft-deprecated].
messages	A flag indicating whether to print messages.

See Also

[substitute_units](#)

Examples

```
substitute_variables(c(
  "ALUMINIUM SOMETHING", "ALUMINUM DISSOLVED",
  "dissolved aluminium", "BORON Total", "KRYPTONITE",
  "Total Fluoride Hardness"
), messages = TRUE)
```

summarise_for_trends *Summarise data by year and month*

Description

Compute annual summaries of water quality observations.

Usage

```
summarise_for_trends(
  data,
  breaks = NULL,
  FUN = "median",
  messages = getOption("wqbc.messages", default = TRUE)
)
```

Arguments

data	The data.frame to analyse.
breaks	A numeric vector used to create groups of consecutive months, if NULL the full year is used.
FUN	The function to use for yearly summaries, e.g. median, mean, or max.
messages	A flag indicating whether to print messages.

Details

The data must contain the columns Station, Date, Variable, Value, and Units.

Value

A tibble data.frame with rows for each Station, Variable, Year and month grouping.

Examples

```
# select one station
data(yuepilon)
data <- yuepilon[yuepilon$Station == "02EA005", ]
# estimate trend (using simple sen slope)
trend <- test_trends(data, messages = TRUE)
# get the data used in the test
datasum <- summarise_for_trends(data)
plot(datasum$Year, datasum$Value,
      main = paste("p-value =", round(trend$significance, 3)),
      ylab = "Value", xlab = "Year", las = 1
    )
```

summarise_wqdata

Summarise Water Quality Data

Description

Calculates summary statistics for water quality data using log-normal maximum-likelihood models.

Usage

```
summarise_wqdata(
  x,
  by = NULL,
  censored = FALSE,
  na.rm = FALSE,
  conf_level = 0.95,
  quan_range = 0.5
)
```

Arguments

x	The data.frame to summarise.
by	A character vector specifying the columns in x to independently summarise by.
censored	A flag specifying whether to account for non-detects.
na.rm	A flag specifying whether to exclude missing Value values when summarising.
conf_level	A number between 0 and 1 specifying confidence limits. By default calculates 95% confidence intervals.
quan_range	A number between 0 and 1 specifying the quantile range. By default calculates the inter-quartile range.

Details

The data set must include a numeric 'Value' and a character or factor 'Variable' column.

By default the summary statistics are independently calculated for each Variable. The user can specify additional columns to independently calculate the statistics by using the by argument.

If the user wishes to account for non-detects using left-censored maximum-likelihood (by setting censored = TRUE) the data set must also include a numeric DetectionLimit column.

Missing values in the DetectionLimit column are assumed to indicate that the Values are not censored. Missing values in the Value column are always considered to be missing values. If the user wishes to exclude missing values in the Value column they should set na.rm = TRUE.

Value

A tibble of the summary statistics.

Examples

```
data.frame(Variable = "var", Value = 1:5, stringsAsFactors = FALSE)
```

test_trends

Thiel-Sen Trend Test

Description

Analyses time series using the Thiel-Sen estimate of slope. It requires at least 6 years of data.

Usage

```
test_trends(
  data,
  breaks = NULL,
  FUN = "median",
  messages = getOption("wqbc.messages", default = TRUE)
)
```

Arguments

data	The data.frame to analyse.
breaks	A numeric vector used to create groups of consecutive months, if NULL the full year is used.
FUN	The function to use for yearly summaries, e.g. median, mean, or max.
messages	A flag indicating whether to print messages.

Details

The data must contain the columns Station, Date, Variable, Value, and Units.

Value

A tibble data.frame with rows for each Station, Variable, and month grouping, and additional columns for the sen slope estimate, 95\

See Also

[zyp.sen](#)

Examples

```
data <- wqbc::yuepilon
trend <- test_trends(data, breaks = 6, messages = TRUE)
## Not run:
demo(test_trends)

## End(Not run)
```

tidy_ec_data

Tidy Environment Canada Data

Description

Tidies water quality data downloaded from Environment Canada website. It is recommended to obtain the data via [canwqdata::wq_site_data\(\)](#) or [canwqdata::wq_basin_data\(\)](#) It retains and renames required columns and sets the timezone to PST.

Usage

```
tidy_ec_data(
  x,
  cols = character(0),
  mdl_action = c("zero", "mdl", "half", "na", "none")
)
```

Arguments

x	The data to tidy.
cols	additional columns from the EMS data to retain specified as a character vector of column names that exist in the data. The default columns retained are: <ul style="list-style-type: none"> • "SITE_NO" • "DATE_TIME_HEURE" (Renamed to "DateTime") • "VARIABLE" (Renamed to "Variable") • "VMV_CODE" (Renamed to "Code") • "VALUE_VALEUR" (Renamed to "Value") • "UNIT_UNITE" (Renamed to "Units") • "DSL_LDE" (Renamed to "DetectionLimit")

- "FLAG_MARQUEUR" (Renamed to "ResultLetter")
- mdl_action What to do with values below the detection limit. Options are "zero" (set the value to 0; the default), # "half" (set the value to half the MDL), "mdl" (set the value to equal to the MDL), or "na" (set the value to NA). Can also be set to "none" to leave as is.

Value

A tibble of the tidied rems data.

tidy_ems_data	<i>Tidy EMS Data</i>
---------------	----------------------

Description

Tidies water quality data downloaded from EMS database using the bcgov/rems package. It retains and renames required columns and sets the timezone to PST.

Usage

```
tidy_ems_data(
  x,
  cols = character(0),
  mdl_action = c("zero", "mdl", "half", "na", "none")
)
```

Arguments

- x The data to tidy.
- cols additional columns from the EMS data to retain specified as a character vector of column names that exist in the data. The default columns retained are:
- "EMS_ID"
 - "MONITORING_LOCATION" (Renamed to "Station")
 - "COLLECTION_START" (Renamed to "DateTime")
 - "PARAMETER" (Renamed to "Variable")
 - "PARAMETER_CODE" (Renamed to "Code")
 - "RESULT" (Renamed to "Value")
 - "UNIT" (Renamed to "Units")
 - "METHOD_DETECTION_LIMIT" (Renamed to "DetectionLimit")
 - "RESULT_LETTER" (Renamed to "ResultLetter")
 - "SAMPLE_STATE"
 - "SAMPLE_CLASS"
 - "SAMPLE_DESCRIPTOR"
- mdl_action What to do with values below the detection limit. Options are "zero" (set the value to 0; the default), # "half" (set the value to half the MDL), "mdl" (set the value to equal to the MDL), or "na" (set the value to NA). Can also be set to "none" to leave as is.

Details

It sets values that are flagged as being less than the detection limit to zero. It does not alter values that are flagged as being greater than the detection limit - that is left up to the user.

Value

A tibble of the tidied rems data.

vmv_codes	<i>Water Quality Parameter VMV and Variable Codes for Canada</i>
-----------	--

Description

The standard variables, VMV codes and Variable Codes provided by Environment Canada.

Usage

vmv_codes

Format

A tibble

Variable The name of the variable.

VMV_Code The VMV code.

EC_Code The Variable Code in the Environment Canada data.

Details

There can be more than one VMV Code for a variable!

vmv_ems	<i>A crosswalk table linking EMS codes to Environment and Climate Change Canada VMV codes</i>
---------	---

Description

A crosswalk table linking EMS codes to Environment and Climate Change Canada VMV codes

Usage

vmv_ems

Format

A tibble

EMS_CODE EMS Code for the variable

EMS_VARIABLE EMS name for the variable

EMS_UNIT EMS name for the unit

EMS_UNIT_CODE EMS code for the unit

EMS_METHOD_CODE EMS code for the method

EMS_METHOD_TITLE EMS name for the method

EMS_MDL EMS method detection limit

VMV_CODE VMV code (unique for variable, method, and unit)

VMV_VARIABLE_CODE VMV code for the variable

VMV_VARIABLE VMV name for the variable

VMV_VARIABLE_TYPE VMV name for the variable type

VMV_UNIT VMV name for the unit

VMV_UNIT_NAME VMV name for the unit

VMV_METHOD_CODE VMV code for the method

VMV_METHOD_TITLE VMV name for the method

yuepilon

Example data used in Yue, Pilon et al. 2001 taken from the Canadian National Water Data Archive (HYDAT) 1949-1998

Description

Hydrometric data are collected and compiled by Water Survey of Canada's eight regional offices. The information is housed in two centrally-managed databases: HYDEX and HYDAT.

Usage

yuepilon

Format

A data frame with 5 columns:

Station Unique 7-character station identification code.

Date The year of the data stored as if it was taken on the 1st Jan.

Variable The name of the variable.

Value The value of the reading.

Units The units of the value.

Site The full name of the station.

Lat The latitude of the station in decimal degrees.

Long The longitude of the station in decimal degrees.

Details

HYDAT is a relational database that contains the actual computed data for the stations listed in HYDEX. These data include: daily and monthly means of flow, water levels and sediment concentrations (for sediment sites). For some sites, peaks and extremes are also recorded.

WSC now offers hydrometric data and station information in a single downloadable file, either in Microsoft Access Database format or in SQLite format, updated on a quarterly basis.

This database was used to derive the yuepilon dataset, which is a table of annual mean river flows for four sites: 02FB007, 02KB001, 02EA005 and 02GA010.

Source

<http://www.ec.gc.ca/rhc-wsc/default.asp?lang=En&n=9018B5EC-1>

Index

* datasets

- ccme, 4
 - codes, 6
 - dummy, 8
 - ems_codes, 9
 - fraser, 10
 - limits, 12
 - site_limits, 17
 - stations, 18
 - vmv_codes, 25
 - vmv_ems, 25
 - yuepilon, 26
- calc_limits, 2, 5, 6, 12, 14
- canwqdata::wq_basin_data(), 23
- canwqdata::wq_site_data(), 23
- ccme, 4
- clean_wqdata, 2–4, 5, 8, 18
- codes, 6, 12, 15, 17
- compress_ems_codes, 7, 10
- convert_values, 7
- dummy, 8
- ems_codes, 9
- error, 9
- expand_ems_codes, 7, 10, 13, 15
- fraser, 10
- geomean1, 11
- limits, 12
- lookup_codes, 12
- lookup_limits, 4, 13, 13, 14, 15
- lookup_units, 14
- lookup_use, 14
- lookup_variables, 15, 17
- plot_timeseries, 15
- set_non_detects, 16
- site_limits, 17
- standardise_wqdata
(standardize_wqdata), 17
- standardize_wqdata, 5, 6, 17
- stations, 18
- substitute_units, 17, 19, 20
- substitute_variables, 12, 17, 19, 19
- summarise_for_trends, 20
- summarise_wqdata, 21
- test_trends, 22
- tidy_ec_data, 23
- tidy_ems_data, 24
- vmv_codes, 25
- vmv_ems, 25
- yuepilon, 26
- zyp.sen, 23