

Package: HASP (via r-universe)

November 22, 2024

Type Package

Title Hydrologic AnalySis Package

Version 1.0.3

Description Data aggregation and plotting of USGS groundwater level data.

License CC0

Copyright This software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior. For more information, see the official USGS copyright policy at https://www.usgs.gov/visual-id/credit_usgs.html#copyright

Depends R (>= 3.5.0)

Imports dataRetrieval, dplyr, tidyr, ggplot2, ggpmisc, ggpp, scales, rkt, checkmate, whisker, xfun, usethis, magrittr, EnvStats

Suggests testthat, knitr, kableExtra, covr, readxl, leaflet, shiny, DT, shinyAce, shinycssloaders, shinydashboard, openxlsx, rmarkdown

BugReports <https://code.usgs.gov/water/stats/hasp/-/issues>

VignetteBuilder knitr

BuildVignettes true

LazyLoad yes

LazyData yes

Encoding UTF-8

RoxygenNote 7.3.1

Config/pak/sysreqs git make libgit2-dev libicu-dev libxml2-dev libssl-dev libx11-dev

Repository <https://decicco-usgs.r-universe.dev>

RemoteUrl <https://code.usgs.gov/water/stats/HASP>

RemoteRef HEAD

RemoteSha 186093666c61e4cd424a2cf88da4fc976fcf8464

Contents

aquifer_data	2
composite_data	3
create_groundwater_report	3
daily_frequency_table	4
daily_gwl_plot	5
daily_gwl_summary	8
data_available	9
explore_aquifers	10
filter_sites	10
first_day	11
get_aquifer_data	12
get_state_data	13
gwl_plot_field	13
L2701_example_data	16
last_day	17
map_hydro_data	18
mid_month	18
monthly_frequency_plot	19
monthly_frequency_table	22
monthly_mean	24
nat_aqfr_state	25
normalized_data	25
plot_composite_data	26
plot_normalized_data	27
Sc_CI_plot	28
site_data_summary	29
site_summary	30
theme_gwl	31
trend_plot	32
trend_test	33
water_year	35
weekly_frequency_plot	35
weekly_frequency_table	37
Index	40

aquifer_data	<i>Example aquifer data</i>
--------------	-----------------------------

Description

Example data representing Basin and Range basin-fill aquifers (N100BSNRGB).

Examples

```
head(aquifer_data)
```

composite_data	<i>Composite hydrograph data</i>
----------------	----------------------------------

Description

Create composite data

Usage

```
composite_data(x, num_years, parameter_cd)
```

Arguments

x	aquifer data
num_years	integer number of years required
parameter_cd	character, 5-digit parameter code, default is "72019".

Value

data frame with year, name, and value

Examples

```
aquifer_data <- aquifer_data
num_years <- 30

comp_data <- composite_data(aquifer_data, num_years, "72019")
```

create_groundwater_report	<i>Generate HASP report</i>
---------------------------	-----------------------------

Description

Opens a template of the blanks sample report. Running the function will create a Rmd file. The file can be "knit" as-is, or adjusted before rendering.

Usage

```
create_groundwater_report(  
  siteID,  
  report_name,  
  report_folder,  
  output_type = "word"  
)
```

Arguments

siteID character. USGS site ID, should be a groundwater site.
report_name character. Base name of report.
report_folder character. Report folder, can be full path or partial.
output_type should be either "word", "html", or "pdf".

Examples

```
## Not run:  
create_groundwater_report("424520070562401",  
                           report_name = "example_report",  
                           report_folder = "reports",  
                           output_type = "word")  
  
create_groundwater_report("424520070562401",  
                           report_name = "example_report",  
                           report_folder = "reports",  
                           output_type = "html")  
  
## End(Not run)
```

daily_frequency_table *Daily frequency table*

Description

Calculates the historical max, mean, minimum, and number of available points for each day of the year

Usage

```
daily_frequency_table(  
  gw_level_dv,  
  gwl_data,  
  parameter_cd = NA,  
  date_col = NA,  
  value_col = NA,  
  approved_col = NA,  
  stat_cd = NA  
)
```

Arguments

gw_level_dv	data frame, daily groundwater level data from readNWISdv
gwl_data	data frame returned from dataRetrieval::readNWISgwl, or data frame with mandatory columns lev_dt (representing date), lev_age_cd (representing approval code), and a column representing the measured value (either lev_va, sl_lev_va, or value).
parameter_cd	If data in gw_level_dv comes from NWIS, the parameter_cd can be used to define the value_col. If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the heading of the date column. The default is NA, which the code will try to get the column name automatically.
value_col	name of value column. The default is NA, which the code will try to get the column name automatically.
approved_col	name of column to get provisional/approved status.
stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.

Value

a data frame giving the max, mean, min, and number of available days of data for each day of the year.

Examples

```
# site <- "263819081585801"
p_code_dv <- "62610"
statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, p_code_dv, statCd = statCd)
gw_level_dv <- L2701_example_data$Daily
daily_frequency_table(gw_level_dv,
                      NULL,
                      parameter_cd = "62610")

gwl_data <- L2701_example_data$Discrete
daily_frequency_table(gw_level_dv,
                      gwl_data,
                      parameter_cd = "62610")
```

daily_gwl_plot

Plot recent data

Description

Calculates daily statistics based on all approved data. Daily, discrete, or both types are included. Historic median or mean are plotted based on all of the approved data.

Usage

```

daily_gwl_plot(
  gw_level_dv,
  gwl_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  start_date = NA,
  end_date = NA,
  historical_stat = "mean",
  month_breaks = FALSE,
  plot_title = "",
  subtitle = "U.S. Geological Survey",
  y_axis_label = "",
  flip = FALSE
)

```

Arguments

<code>gw_level_dv</code>	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
<code>gwl_data</code>	data frame returned from readNWISgwl , or data frame with a date, value, and approval columns. Using the convention: <code>lev_dt</code> (representing date), <code>lev_age_cd</code> (representing approval code), and <code>lev_va</code> or <code>sl_lev_va</code> (representing value) will allow defaults to work. Use NULL for no discrete data.
<code>parameter_cd</code>	If data in <code>gw_level_dv</code> comes from NWIS, the <code>parameter_cd</code> can be used to define the <code>value_col</code> . If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
<code>date_col</code>	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gwl_data</code> data frames require custom column names, the first value of this input defines the date column for <code>gw_level_dv</code> , and the second defines <code>gwl_data</code> .
<code>value_col</code>	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gwl_data</code> data frames require custom column names, the first value of this input defines the value column for <code>gw_level_dv</code> , and the second defines <code>gwl_data</code> .
<code>approved_col</code>	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gwl_data</code> data frames require custom column names, the first value of this input defines the approval column for <code>gw_level_dv</code> , and the second defines <code>gwl_data</code> .#' @param stat_cd If data in <code>gw_level_dv</code> comes from NWIS, the <code>stat_cd</code> can be used to help define the <code>value_col</code> .

stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.
start_date	Date to start plot. If NA (which is the default), the plot will start 2 years before the most recent value.
end_date	Date to end plot. If NA (which is the default), the plot will end with the latest measurement.
historical_stat	the summary statistic to use for middle line of the plot. Either "mean" or "median."
month_breaks	a logical indicating whether to use monthly breaks for the plot
plot_title	the title to use on the plot
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".
y_axis_label	the label to use for the y axis
flip	logical. If TRUE, flips the y axis so that the smallest number is on top. Default is FALSE.

Value

a ggplot object with a ribbon indicating the historical daily range, the historical daily mean or median, and approved and provisional daily data for the last two years

Examples

```

site <- "263819081585801"
p_code_dv <- "62610"
statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, p_code_dv, statCd = statCd)
gw_level_dv <- L2701_example_data$Daily

gwl_data <- L2701_example_data$Discrete

daily_gwl_plot(gw_level_dv,
               NULL,
               parameter_cd = "62610",
               plot_title = "Groundwater Level",
               historical_stat = "median")

daily_gwl_plot(gw_level_dv,
               gwl_data,
               parameter_cd = "62610",
               plot_title = "Groundwater Level",
               historical_stat = "median")

daily_gwl_plot(gw_level_dv,
               gwl_data,
               parameter_cd = "62610",
               plot_title = "Groundwater Level",
               month_breaks = TRUE,
               start_date = "2020-10-01",

```

```

        historical_stat = "median")

daily_gwl_plot(gw_level_dv, gwl_data,
              parameter_cd = "62610",
              plot_title = "Groundwater Level",
              month_breaks = TRUE,
              start_date = "2018-10-01",
              end_date = "2020-10-01",
              historical_stat = "median")

```

daily_gwl_summary	<i>Summary table of daily data</i>
-------------------	------------------------------------

Description

Summary table of daily data

Usage

```

daily_gwl_summary(
  gw_level_dv,
  gwl_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA
)

```

Arguments

gw_level_dv	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
gwl_data	data frame returned from readNWISgwl , or data frame with a date, value, and approval columns. Using the convention: lev_dt (representing date), lev_age_cd (representing approval code), and lev_va or sl_lev_va (representing value) will allow defaults to work. Use NULL for no discrete data.
parameter_cd	If data in gw_level_dv comes from NWIS, the parameter_cd can be used to define the value_col. If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames require custom column names, the first value of this input defines the date column for gw_level_dv, and the second defines gwl_data.

value_col	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames require custom column names, the first value of this input defines the value column for gw_level_dv, and the second defines gwl_data.
approved_col	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames require custom column names, the first value of this input defines the approval column for gw_level_dv, and the second defines gwl_data. # @param stat_cd If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.
stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.

Value

a summary table giving the period of record, completeness and percentile values

Examples

```
# site <- "263819081585801"
p_code_dv <- "62610"
statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, p_code_dv, statCd = statCd)
gw_level_dv <- L2701_example_data$Daily
daily_gwl_summary(gw_level_dv,
                  gwl_data = NULL,
                  parameter_cd = p_code_dv)

gwl_data <- L2701_example_data$Discrete
daily_gwl_summary(gw_level_dv,
                  gwl_data = gwl_data,
                  parameter_cd = p_code_dv)
```

data_available	site_summary
----------------	--------------

Description

Get station summary information

Usage

```
data_available(siteID)
```

Arguments

siteID character. USGS site ID for a groundwater site.

Examples

```
siteID <- "263819081585801"  
site_data_available <- data_available(siteID)
```

explore_aquifers	<i>Explore data in shiny applications</i>
------------------	---

Description

Open an interactive app in a browser. See the "Shiny App" vignette: `vignette("shinyApp", package = "toxEval")` for more details. Using this function is a quick and convenient way to explore data. For more customization, the R-code to produce each graph and table is displayed in the app. That is a good starting-point for a custom analysis.

Usage

```
explore_aquifers(browse = TRUE)  
  
explore_site(browse = TRUE)
```

Arguments

browse Logical. Use browser for running Shiny app.

Examples

```
## Not run:  
explore_aquifers()  
  
## End(Not run)  
## Not run:  
explore_site()  
  
## End(Not run)
```

filter_sites	<i>filter_sites</i>
--------------	---------------------

Description

Filter down to sites with num_years of data

Usage

```
filter_sites(
  x,
  parameter_cd = "72019",
  num_years = NA,
  start_year = NA,
  end_year = NA
)
```

Arguments

x	aquifer data
parameter_cd	character, 5-digit parameter code, default is "72019".
num_years	integer number of years required. This can be NA, in which case the filter will use the full range of the data.
start_year	integer the first year to filter from. If NA, the filter will use the minimum from the data.
end_year	integer the last year to filter from. If NA, the filter will use the last year.

Value

data frame filter of x

Examples

```
aquifer_data <- aquifer_data
num_years <- 30

aq_data <- filter_sites(aquifer_data,
                        parameter_cd = "72019",
                        num_years = num_years)
```

first_day

Find the first day of the month for a given date

Description

Find the first day of the month for a given date

Usage

```
first_day(date)
```

Arguments

date	a vector of dates
------	-------------------

Value

the first day of the month that given dates fall in

Examples

```
date <- as.Date("2020-12-28")
first_day(date)
```

get_aquifer_data *get_aquifer_data*

Description

Get USGS data based on aquiferCd

Usage

```
get_aquifer_data(aquiferCd, startDate, endDate, parameter_cd = "72019")
```

Arguments

<code>aquiferCd</code>	character. To see valid aquifer codes, see the included data frame <code>local_aqfr</code> .
<code>startDate</code>	date or string. Beginning date of when to pull data.
<code>endDate</code>	date of string Ending date to pull data.
<code>parameter_cd</code>	5-digit character USGS parameter code.

Examples

```
end_date <- "2021-01-01"
start_date <- "1989-12-31"

aquiferCd <- "S100CSLLWD"

aq_data <- get_aquifer_data(aquiferCd, start_date, end_date)
```

get_state_data	<i>get_state_data</i>
----------------	-----------------------

Description

Get USGS data based for a single state with specific aquifer codes.

Usage

```
get_state_data(state, aquiferCd, startDate, endDate, parameter_cd = "72019")
```

Arguments

state	character. Can be state abbreviation, long name, or numeric code.
aquiferCd	character. To see valid aquifer codes, see the included data frame local_aqfr.
startDate	date or string. Beginning date of when to pull data.
endDate	date of string Ending date to pull data.
parameter_cd	5-digit character USGS parameter code. Default is "72019".

Examples

```
end_date <- "2021-01-01"
start_date <- "1989-12-31"
aquiferCd <- "S100CSLLWD"

st_data <- get_state_data("WI", aquiferCd,
                        start_date, end_date)
```

gwl_plot_field	<i>Single site groundwater level plots and tables</i>
----------------	---

Description

Function to create the field groundwater level data plot.

Usage

```
gwl_plot_field(
  gwl_data,
  plot_title = "",
  parameter_cd = NA,
  date_col = "lev_dt",
  value_col = NA,
```

```

    approved_col = "lev_age_cd",
    flip = TRUE,
    y_label = "",
    subtitle = "U.S. Geological Survey"
  )

gwl_plot_all(
  gw_level_dv,
  gwl_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  y_label = "",
  subtitle = "U.S. Geological Survey",
  plot_title = "",
  add_trend = FALSE,
  n_years = 10,
  POR_trend = TRUE,
  flip = FALSE,
  days_required_per_month = 14
)

```

Arguments

gwl_data	data frame returned from readNWISgwl , or data frame with a date, value, and approval columns. Using the convention: lev_dt (representing date), lev_age_cd (representing approval code), and lev_va or sl_lev_va (representing value) will allow defaults to work. Use NULL for no discrete data.
plot_title	character, title for plot.
parameter_cd	If data in gw_level_dv comes from NWIS, the parameter_cd can be used to define the value_col. If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames require custom column names, the first value of this input defines the date column for gw_level_dv, and the second defines gwl_data.
value_col	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames require custom column names, the first value of this input defines the value column for gw_level_dv, and the second defines gwl_data.
approved_col	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gwl_data data frames

	require custom column names, the first value of this input defines the approval column for gw_level_dv, and the second defines gwl_data.
flip	logical. If TRUE, flips the y axis so that the smallest number is on top. Default is TRUE.
y_label	character for y-axis label. Consider using readNWISpCode for USGS parameter_nm.
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".
gw_level_dv	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.
add_trend	logical. Uses trend_test.
n_years	integer. This is the number of years to calculate the trend on. Default is 10. This can be a vector of years.
POR_trend	a logical indicating whether to include a trend test for the full period of record. Default is TRUE.
days_required_per_month	integer. Number of days required per month. Default is 14. Only used if add_trend is TRUE using daily data.

Examples

```
# site <- "263819081585801"
# gwl_data <- dataRetrieval::readNWISgwl(site)

# Using package example data:
gwl_data <- L2701_example_data$Discrete
plot_title <- attr(gwl_data, "siteInfo")[["station_nm"]]
pcodes <- dataRetrieval::readNWISpCode(unique(gwl_data$parameter_cd))
gwl_plot_field(gwl_data,
               plot_title = plot_title,
               parameter_cd = "62610",
               flip = FALSE,
               y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"])
gwl_plot_field(gwl_data,
               plot_title = plot_title,
               y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62611"],
               parameter_cd = "62611",
               flip = FALSE)
gwl_plot_field(gwl_data,
               plot_title = plot_title,
               y_label = pcodes$parameter_nm[pcodes$parameter_cd == "72019"],
               parameter_cd = "72019",
               flip = TRUE)
# site <- "263819081585801"
parameterCd <- "62610"
# statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, parameterCd, statCd = statCd)
```

```

# Using package example data:
gw_level_dv <- L2701_example_data$Daily
gw_data <- L2701_example_data$Discrete
plot_title <- attr(gw_data, "siteInfo")[[ "station_nm" ]]
pcodes <- dataRetrieval::readNWISpCode(unique(gw_data$parameter_cd))

gw_plot_all(gw_level_dv,
            NULL,
            parameter_cd = "62610",
            plot_title = plot_title,
            y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"],
            flip = TRUE)

gw_plot_all(gw_level_dv,
            gw_data,
            parameter_cd = "62610",
            plot_title = plot_title,
            y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"],
            add_trend = TRUE,
            flip = FALSE)

gw_plot_all(gw_level_dv,
            gw_data,
            parameter_cd = "62610",
            n_years = c(5, 10, 20),
            POR_trend = TRUE,
            y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"],
            plot_title = plot_title,
            add_trend = TRUE)

gw_plot_all(NULL,
            gw_data,
            parameter_cd = "62610",
            plot_title = plot_title,
            y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"])

gw_plot_all(NULL,
            gw_data,
            parameter_cd = "62610",
            plot_title = plot_title,
            y_label = pcodes$parameter_nm[pcodes$parameter_cd == "62610"],
            add_trend = TRUE)

```


Description

Example data from site 263819081585801 L-2701. Data is a named list of "Daily" for daily groundwater level at the site and "Discrete" for discrete groundwater level measurements from the site. Updated April 9, 2021.

Examples

```
head(L2701_example_data$Daily)
head(L2701_example_data$Discrete)
head(L2701_example_data$QW)
```

last_day

Find the last day of the month for a given date

Description

Find the last day of the month for a given date

Usage

```
last_day(date)
```

Arguments

date a vector of dates

Value

the last day of the month that given dates fall in

Examples

```
date <- as.Date("2020-12-28")
last_day(date)
last_day("2020-02-15")
last_day("2019-02-15")
last_day(c("2020-12-28", "2020-02-15", "2019-02-15"))
```

map_hydro_data	<i>Map data used in composite hydrographs</i>
----------------	---

Description

Map data used in composite hydrographs

Usage

```
map_hydro_data(x, num_years, parameter_cd = "72019")
```

Arguments

x	aquifer data frame. Requires at least 3 columns. Two are required "site_no", "year", and "value".
num_years	integer number of years required
parameter_cd	character, 5-digit parameter code, default is "72019".

Value

leaflet object

Examples

```
aquifer_data <- aquifer_data
num_years <- 30

map_data <- map_hydro_data(aquifer_data, num_years)
map_data
```

mid_month	<i>Find the middle of the month for a given date</i>
-----------	--

Description

Find the middle of the month for a given date

Usage

```
mid_month(date)
```

Arguments

date	a vector of dates
------	-------------------

Value

the middle day of the month the given dates fall in

Examples

```
date <- as.Date("2020-12-28")
mid_month(date)
mid_month(c("2019-02-15", "2020-03-08", "2010-06-01"))
```

monthly_frequency_plot

Plot monthly frequency analysis

Description

This plot uses calculations from `monthly_frequency_table`. Daily, discrete, or both types of data can be included.

Usage

```
monthly_frequency_plot(
  gw_level_dv,
  gwl_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  plot_title = "",
  subtitle = "U.S. Geological Survey",
  plot_range = c("Past year"),
  y_axis_label = "",
  flip = FALSE,
  percentile_colors = NA,
  include_edges = FALSE,
  median_point_size = 2.5,
  data_point_size = 2.5
)
```

Arguments

<code>gw_level_dv</code>	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
<code>gwl_data</code>	data frame returned from readNWISgwl , or data frame with a date, value, and approval columns. Using the convention: <code>lev_dt</code> (representing date), <code>lev_age_cd</code> (representing approval code), and <code>lev_va</code> or <code>sl_lev_va</code> (representing value) will allow defaults to work. Use NULL for no discrete data.

<code>parameter_cd</code>	If data in <code>gw_level_dv</code> comes from NWIS, the <code>parameter_cd</code> can be used to define the <code>value_col</code> . If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
<code>date_col</code>	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw_data</code> data frames require custom column names, the first value of this input defines the date column for <code>gw_level_dv</code> , and the second defines <code>gw_data</code> .
<code>value_col</code>	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw_data</code> data frames require custom column names, the first value of this input defines the value column for <code>gw_level_dv</code> , and the second defines <code>gw_data</code> .
<code>approved_col</code>	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw_data</code> data frames require custom column names, the first value of this input defines the approval column for <code>gw_level_dv</code> , and the second defines <code>gw_data</code> .
<code>stat_cd</code>	If data in <code>gw_level_dv</code> comes from NWIS, the <code>stat_cd</code> can be used to help define the <code>value_col</code> .
<code>plot_title</code>	the title to use on the plot.
<code>subtitle</code>	character. Sub-title for plot, default is "U.S. Geological Survey".
<code>plot_range</code>	the time frame to use for the plot. Either "Past year" to use the last year of data, or "Calendar year" to use the current calendar year, beginning in January.
<code>y_axis_label</code>	the label used for the y-axis of the plot.
<code>flip</code>	logical. If TRUE, flips the y axis so that the smallest number is on top. Default is TRUE.
<code>percentile_colors</code>	Optional argument to provide a vector of 5 or 7 colors used to fill the percentile bars in order from lowest percentile bin to the highest percentile bin. Default behavior (NA) is to use legacy plot colors. If <code>include_edges</code> parameter is set to TRUE, then this vector must be 7 colors long.
<code>include_edges</code>	Optional argument to toggle on the "edge bins" min-5 and 95-max on the plot. Default is FALSE which does not plot those bins.
<code>median_point_size</code>	Optional argument to specify the size of the median point markers which are shown as black triangles on the plot. The default size is 2.5.
<code>data_point_size</code>	Optional argument to specify the size of the data point markers which are shown as red diamonds on the plot. The default size is 2.5.

Value

a ggplot with rectangles representing the historical monthly percentile, black triangles representing the historical monthly median, and red diamonds showing the last year of groundwater level measurements.


```

monthly_frequency_edge_bins
monthly_frequency_custom_point_sizes <- monthly_frequency_plot(gw_level_dv,
  plot_title = "L2701 Groundwater Level",
  flip = FALSE,
  include_edges = TRUE)
  gw_level_dv,
  gw_l_data = gw_l_data,
  parameter_cd = "62610",
  plot_title = "L2701 Groundwater Level",
  y_axis_label = label,
  median_point_size = 0.5,
  data_point_size = 3)
monthly_frequency_custom_point_sizes

```

monthly_frequency_table

Create a table of monthly frequency analysis

Description

The table will accept daily, discrete, or a both types of data. The median of each year/month is calculated. Then using that median, monthly stats are calculated. Percentiles are calculated using the quantile function with "type=6".

Usage

```

monthly_frequency_table(
  gw_level_dv,
  gw_l_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  flip = FALSE
)

```

Arguments

gw_level_dv	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
gw_l_data	data frame returned from readNWISgw1 , or data frame with a date, value, and approval columns. Using the convention: lev_dt (representing date), lev_age_cd (representing approval code), and lev_va or sl_lev_va (representing value) will allow defaults to work. Use NULL for no discrete data.


```
head(monthly_flip)                                flip = TRUE)
```

monthly_mean	<i>Monthly mean</i>
--------------	---------------------

Description

Take the mean of each month, filter if there's not at least 15 days, and create a new date that is the midpoint of each month.

Usage

```
monthly_mean(x, date_col = "Date", value_col = "X_62610_00001")
```

Arguments

x	data.frame
date_col	character name of date column
value_col	character name of value column

Examples

```
# site <- "263819081585801"
parameterCd <- "62610"
# statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, parameterCd,
#                                         statCd = statCd)
# Using package example data:
gw_level_dv <- L2701_example_data$Daily

site_statistics <- monthly_frequency_table(gw_level_dv,
                                           NULL,
                                           parameter_cd = parameterCd)

gw_monthly <- monthly_mean(gw_level_dv)
```

nat_aqfr_state	<i>Aquifer data included</i>
----------------	------------------------------

Description

National aquifer data
 Local aquifer data codes
 markerTable raw data

Value

data frame with columns Country, State, nat_aqfr_cd, long_name, and other_cd
 data frame with columns long_name, nat_aqfr_cd, state_indexes, and states
 data frame with aqfr_cd, Aqfr_Name_prpr
 data frame with columns Trend and MarkerDescription
 data frame with columns trendType, trend, markerDescription, r_lwd

Examples

```
head(nat_aqfr_state)
head(summary_aquifers)
head(local_aqfr)
```

normalized_data	<i>Composite normalized hydrograph data</i>
-----------------	---

Description

Create normalized composite data

Usage

```
normalized_data(x, num_years, parameter_cd = "72019")
```

Arguments

x	aquifer data
num_years	integer number of years required
parameter_cd	character, 5-digit parameter code, default is "72019".

Details

Information can be found here: <https://groundwaterwatch.usgs.gov/composite/help/CompositeGroundwaterLevel.docx.html>

Value

data frame with year, name, and value

Examples

```
aquifer_data <- aquifer_data
num_years <- 30

norm_data <- normalized_data(aquifer_data, num_years, "72019")
```

plot_composite_data *Plot composite hydrograph data*

Description

Create composite hydrograph plot

Usage

```
plot_composite_data(
  x,
  num_years = NA,
  parameter_cd = "72019",
  plot_title = "",
  subtitle = "U.S. Geological Survey"
)
```

Arguments

x	aquifer data frame. Requires at least 3 columns. Two are required "site_no", "year", and "value".
num_years	integer number of years required. If NA, the analysis will default to the range of the data in x.
parameter_cd	character, 5-digit parameter code, default is "72019".
plot_title	character title included on plot.
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".

Value

ggplot2 object

Examples

```

aquifer_data <- aquifer_data

comp_data <- plot_composite_data(aquifer_data,
                                num_years = 30,
                                parameter_cd = "72019",
                                plot_title = "Calendar Year")

comp_data
# Do it on a water year:
aquifer_data$cal_year <- aquifer_data$year
aquifer_data$year <- aquifer_data$water_year
plot_composite_data(aquifer_data,
                    parameter_cd = "72019",
                    num_years = 30,
                    plot_title = "Water Year")

```

plot_normalized_data *Plot normalized composite hydrograph data*

Description

Create composite hydrograph plot

Usage

```

plot_normalized_data(
  x,
  num_years = NA,
  parameter_cd = "72019",
  plot_title = "",
  subtitle = "U.S. Geological Survey"
)

```

Arguments

x	aquifer data frame. Requires at least 3 columns. Two are required "site_no", "year", and "value".
num_years	integer number of years required to the analysis. If NA, the analysis will default to the range of the data in x.
parameter_cd	character, 5-digit parameter code, default is "72019".
plot_title	character title of plot.
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".

Value

ggplot2 object

Examples

```

aquifer_data <- aquifer_data

norm_data <- plot_normalized_data(aquifer_data,
                                parameter_cd = "72019",
                                num_years = 30)

norm_data

aquifer_data$cal_year <- aquifer_data$year
aquifer_data$year <- aquifer_data$water_year
plot_normalized_data(aquifer_data,
                    parameter_cd = "72019",
                    num_years = 30)

```

Sc_Cl_plot

Specific conductance and chloride

Description

Functions to create the individual chloride, specific conductance, and combination plots and tables for a single site.

Usage

```
Sc_Cl_plot(qw_data, plot_title, subtitle = "U.S. Geological Survey")
```

```
Sc_Cl_table(qw_data)
```

```

qw_plot(
  qw_data,
  plot_title,
  y_label = NA,
  CharacteristicName = "Chloride",
  start_date = NA,
  end_date = NA,
  subtitle = "U.S. Geological Survey"
)

```

```
qw_summary(qw_data, CharacteristicName, norm_range = NA)
```

Arguments

qw_data	data frame returned from dataRetrieval::readWQPqw, must include columns sample_dt, parm_cd, result_va
plot_title	character title for plot
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".

y_label	character label for y axis. If left as NA, the function will attempt to use the "variableInfo" attribute of qw_data. This is attached to dataRetrieval output.
CharacteristicName	character CharacteristicName to filter to.
start_date	Date to start plot. If NA (which is the default), the plot will start at the earliest measurement.
end_date	Date to end plot. If NA (which is the default), the plot will end with the latest measurement.
norm_range	a numerical range to potentially group the data. If NA, no grouping is shown.

Examples

```
# site <- "263819081585801"
# parameterCd <- c("00095","90095","00940","99220")
# site_data <- dataRetrieval::readWQPqw(site,
#                                     parameterCd)
# Using package example data:
qw_data <- L2701_example_data$QW
plot_title <- paste(attr(qw_data, "siteInfo")[["station_nm"]], ": Specific Conductance vs Chloride")
Sc_Cl_plot(qw_data, plot_title)

sc_cl <- Sc_Cl_table(qw_data)
plot_title <- attr(qw_data, "siteInfo")[["station_nm"]]
qw_plot(qw_data, plot_title, CharacteristicName = "Chloride")
qw_plot(qw_data, plot_title, CharacteristicName = "Specific conductance")
qw_plot(qw_data,
        plot_title,
        CharacteristicName = "Specific conductance",
        start_date = "1990-01-01")

site <- "USGS-01491000"
qw_data_phos <- dataRetrieval::readWQPqw(site, "Orthophosphate")
qw_plot(qw_data_phos ,
        CharacteristicName = "Orthophosphate",
        plot_title = "Choptank: Orthophosphate")

qw_summary(qw_data, CharacteristicName = "Chloride",
           norm_range = c(230, 860))
qw_summary(qw_data, CharacteristicName = "Specific conductance",
           norm_range = NA)
```

site_data_summary *site_data_summary*

Description

Get summaries of data by site. Requires a column site_no, and will take the summaries

Usage

```
site_data_summary(x, value_col = "value", site_col = "site_no")
```

Arguments

x	data frame
value_col	name of value column. The default is "value".
site_col	name of site column. This is the column we are grouping by.

Value

data frame with 10 columns

Examples

```
aquifer_data <- aquifer_data  
aquifer_data <- aquifer_data[aquifer_data$parameter_cd == "72019", ]  
summary_info <- site_data_summary(aquifer_data)
```

site_summary	<i>site_summary</i>
--------------	---------------------

Description

Get station summary information

Usage

```
site_summary(siteID, markdown = FALSE)
```

Arguments

siteID	character
markdown	logical. Use markdown formatting or console-friendly.

Examples

```
siteID <- "263819081585801"  
site_metadata <- site_summary(siteID)
```

trend_plot	<i>Chloride plot with trend</i>
------------	---------------------------------

Description

Function creates the chloride over time plot with trends.

Usage

```
trend_plot(
  qw_data,
  plot_title,
  y_label = NA,
  n_years = 10,
  POR_trend = TRUE,
  CharacteristicName = c("Chloride"),
  norm_range = c(230, 860),
  subtitle = "U.S. Geological Survey",
  include_table = TRUE
)
```

Arguments

qw_data	data frame returned from dataRetrieval::readWQPqw, must include columns ActivityStartDateTime, CharacteristicName, result_va
plot_title	character title for plot
y_label	character label for y axis. If left as NA, the function will attempt to use the "variableInfo" attribute of qw_data. This is attached to dataRetrieval output.
n_years	integer. This is the number of years to calculate the trend on. Default is 10. This can be a vector of years.
POR_trend	a logical indicating whether to include a trend test for the full period of record. Default is TRUE.
CharacteristicName	character CharacteristicName to plot
norm_range	a numerical range to potentially group the data. If NA, no grouping is shown.
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".
include_table	logical whether or not to include the trend table in the upper left corner.

Examples

```
# site <- "263819081585801"
# parameterCd <- c("00095", "90095", "00940", "99220")
# site_data <- dataRetrieval::readWQPqw(site,
#                                     parameterCd)
# Using package example data:
```



```
qw_data <- L2701_example_data$QW
plot_title <- paste(attr(qw_data, "siteInfo")[["station_nm"]], ": Chloride")
trend_plot(qw_data, plot_title)
```

trend_test

Trend Test

Description

Test for period of record and user-specified ranges. The default trends are calculated for 10 year and the full period of record.

Usage

```
trend_test(
  gw_level_dv,
  gwl_data,
  n_years = 10,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  pctComplete = 0.5,
  days_required_per_month = 14,
  POR_trend = TRUE
)
```

Arguments

gw_level_dv	daily groundwater level data frame. Often obtained from from readNWISdv
gwl_data	data frame returned from dataRetrieval::readNWISgwl, or data frame with mandatory columns lev_dt (representing date), lev_age_cd (representing approval code), and a column representing the measured value (either lev_va, sl_lev_va, or value).
n_years	integer. This is the number of years to calculate the trend on. Default is 10. This can be a vector of years.
parameter_cd	If data in gw_level_dv comes from NWIS, the parameter_cd can be used to define the value_col. If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the heading of the date column. The default is NA, which the code will try to get the column name automatically.
value_col	name of value column. The default is NA, which the code will try to get the column name automatically.
approved_col	name of column to get provisional/approved status.

stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.
pctComplete	number percentage complete. This is a fraction that represents the amount of data that must be included overall in order to calculate a trend. The default is 0.5, which means if gaps in the data span more than 50 total record, a trend will not be calculated.
days_required_per_month	integer. Number of days required per month to include in the trend test. Default is 14.
POR_trend	a logical indicating whether to include a trend test for the full period of record. Default is TRUE.

Details

For data that is at least on a daily interval, the [rkt](#) function is used. For periodic data, the [kendall-TrendTest](#) is used.

Value

a data frame of test results from 5 and 20 year Kendall Seasonal Trend test

Examples

```
# site <- "263819081585801"
# gw_level_data <- dataRetrieval::readNWISgw1(site)

# Using package example data:
gw1_data <- L2701_example_data$Discrete

gw_level_dv <- L2701_example_data$Daily

trend_test(gw_level_dv,
           gw1_data,
           parameter_cd = "62610")

trend_test(gw_level_dv,
           gw1_data,
           POR_trend = FALSE,
           parameter_cd = "62610")

trend_test(gw_level_dv,
           gw1_data,
           parameter_cd = "62610",
           n_years = 5)

trend_test(gw_level_dv,
           gw1_data,
           parameter_cd = "62610",
           n_years = c(5, 10, 20))

# Only periodic data:
```

```
trend_test(NULL,  
           gw1_data,  
           parameter_cd = "62610")
```

water_year	<i>Convert to water year</i>
------------	------------------------------

Description

This function is a little more robust than [calcWaterYear](#)

Usage

```
water_year(x)
```

Arguments

x character vector

Examples

```
x <- c("2010-01-01", "1994-02", "1980", "2009-11-01")  
water_year(x)
```

weekly_frequency_plot	<i>Plot weekly frequency analysis</i>
-----------------------	---------------------------------------

Description

Weekly statistics are calculated using the `weekly_frequency_table` function. Daily, discrete, or both types of data can be used.

Usage

```
weekly_frequency_plot(  
  gw_level_dv,  
  gw1_data,  
  parameter_cd = NA,  
  date_col = NA,  
  value_col = NA,  
  approved_col = NA,  
  stat_cd = NA,  
  plot_range = "Past year",  
  plot_title = "",  
  subtitle = "U.S. Geological Survey",
```

```

    y_axis_label = "",
    flip = FALSE,
    percentile_colors = NA
  )

```

Arguments

gw_level_dv	data frame, daily groundwater level data. Often obtained from readNWISdv . Use NULL for no daily data.
gw1_data	data frame returned from readNWISgw1 , or data frame with a date, value, and approval columns. Using the convention: lev_dt (representing date), lev_age_cd (representing approval code), and lev_va or sl_lev_va (representing value) will allow defaults to work. Use NULL for no discrete data.
parameter_cd	If data in gw_level_dv comes from NWIS, the parameter_cd can be used to define the value_col. If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gw1_data data frames require custom column names, the first value of this input defines the date column for gw_level_dv, and the second defines gw1_data.
value_col	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gw1_data data frames require custom column names, the first value of this input defines the value column for gw_level_dv, and the second defines gw1_data.
approved_col	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both gw_level_dv and gw1_data data frames require custom column names, the first value of this input defines the approval column for gw_level_dv, and the second defines gw1_data.
stat_cd	If data in gw_level_dv comes from NWIS, the stat_cd can be used to help define the value_col.
plot_range	the time frame to use for the plot. Either "Past year" to use the last year of data, or "Calendar year" to use the current calendar year, beginning in January.
plot_title	the title to use on the plot
subtitle	character. Sub-title for plot, default is "U.S. Geological Survey".
y_axis_label	the label used for the y-axis of the plot.
flip	logical. If TRUE, flips the y axis so that the smallest number is on top. Default is FALSE.
percentile_colors	Optional argument to provide a vector of 5 colors used to fill the percentile bars in order from lowest percentile bin to the highest percentile bin. Default behavior (NA) is to use legacy plot colors.

Value

a ggplot object with rectangles representing the historical weekly percentiles, and points representing the historical median and daily values

Examples

```
# site <- "263819081585801"
p_code_dv <- "62610"
statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, p_code_dv, statCd = statCd)
gw_level_dv <- L2701_example_data$Daily
weekly_frequency_plot(gw_level_dv,
                      gw_l_data = NULL,
                      date_col = "Date",
                      value_col = "X_62610_00001",
                      approved_col = "X_62610_00001_cd")

gw_l_data <- L2701_example_data$Discrete

weekly_frequency_plot(gw_level_dv,
                      gw_l_data = gw_l_data,
                      parameter_cd = "62610")

weekly_frequency_plot(gw_level_dv,
                      gw_l_data = gw_l_data,
                      parameter_cd = "62610",
                      flip = TRUE)
```

weekly_frequency_table

Create a table of weekly frequency analysis

Description

The weekly frequency analysis is based on daily, discrete, or both types of data. The median of each year/week combo is calculated, then overall weekly statistics are calculated off of that median.

Usage

```
weekly_frequency_table(
  gw_level_dv,
  gw_l_data,
  parameter_cd = NA,
  date_col = NA,
  value_col = NA,
  approved_col = NA,
  stat_cd = NA,
  flip = FALSE
)
```

Arguments

gw_level_dv	data frame, daily groundwater level data. Often obtained from <code>readNWISdv</code> . Use NULL for no daily data.
gw1_data	data frame returned from <code>readNWISgw1</code> , or data frame with a date, value, and approval columns. Using the convention: <code>lev_dt</code> (representing date), <code>lev_age_cd</code> (representing approval code), and <code>lev_va</code> or <code>sl_lev_va</code> (representing value) will allow defaults to work. Use NULL for no discrete data.
parameter_cd	If data in <code>gw_level_dv</code> comes from NWIS, the <code>parameter_cd</code> can be used to define the <code>value_col</code> . If the data doesn't come directly from NWIS services, this can be set to NA, and this argument will be ignored.
date_col	the name of the date column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw1_data</code> data frames require custom column names, the first value of this input defines the date column for <code>gw_level_dv</code> , and the second defines <code>gw1_data</code> .
value_col	the name of the value column. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw1_data</code> data frames require custom column names, the first value of this input defines the value column for <code>gw_level_dv</code> , and the second defines <code>gw1_data</code> .
approved_col	the name of the column to get provisional/approved status. The default is NA, in which case, the code will try to get the column name automatically based on NWIS naming conventions. If both <code>gw_level_dv</code> and <code>gw1_data</code> data frames require custom column names, the first value of this input defines the approval column for <code>gw_level_dv</code> , and the second defines <code>gw1_data</code> .
stat_cd	If data in <code>gw_level_dv</code> comes from NWIS, the <code>stat_cd</code> can be used to help define the <code>value_col</code> .
flip	logical. If TRUE, flips labels so that the lower numbers are in the higher percentages. Default is TRUE.

Value

a data frame of weekly frequency analysis

Examples

```
# site <- "263819081585801"
p_code_dv <- "62610"
statCd <- "00001"
# gw_level_dv <- dataRetrieval::readNWISdv(site, p_code_dv, statCd = statCd)
gw_level_dv <- L2701_example_data$Daily
weekly_frequency <- weekly_frequency_table(gw_level_dv,
                                           NULL,
                                           parameter_cd = "62610")

head(weekly_frequency)

gw1_data <- L2701_example_data$Discrete
```


Index

- * **datasets**
 - nat_aqfr_state, 25
- * **data**
 - aquifer_data, 2
- * **quality**
 - aquifer_data, 2
- * **water**
 - aquifer_data, 2
- aquifer_data, 2
- calcWaterYear, 35
- composite_data, 3
- create_groundwater_report, 3
- daily_frequency_table, 4
- daily_gwl_plot, 5
- daily_gwl_summary, 8
- data_available, 9
- explore_aquifers, 10
- explore_site (explore_aquifers), 10
- filter_sites, 10
- first_day, 11
- get_aquifer_data, 12
- get_state_data, 13
- gwl_plot_all (gwl_plot_field), 13
- gwl_plot_field, 13
- hasp_framework (theme_gwl), 31
- kendallTrendTest, 34
- L2701_example_data, 16
- last_day, 17
- local_aqfr (nat_aqfr_state), 25
- map_hydro_data, 18
- markerTable (nat_aqfr_state), 25
- markerTable2 (nat_aqfr_state), 25
- mid_month, 18
- monthly_frequency_plot, 19
- monthly_frequency_table, 22
- monthly_mean, 24
- nat_aqfr_state, 25
- normalized_data, 25
- plot_composite_data, 26
- plot_normalized_data, 27
- qw_plot (Sc_Cl_plot), 28
- qw_summary (Sc_Cl_plot), 28
- readNWISdv, 6, 8, 15, 19, 22, 36, 38
- readNWISgwl, 6, 8, 14, 19, 22, 36, 38
- readNWISpCode, 15
- rkt, 34
- Sc_Cl_plot, 28
- Sc_Cl_table (Sc_Cl_plot), 28
- site_data_summary, 29
- site_summary, 30
- summary_aquifers (nat_aqfr_state), 25
- theme_gwl, 31
- trend_plot, 32
- trend_test, 33
- water_year, 35
- weekly_frequency_plot, 35
- weekly_frequency_table, 37