Package: bcgwlreports (via r-universe)

October 6, 2024

Title Create BC Groundwater Level Reports

Version 0.0.0.9000

Description Fetch data, calculate historical percentiles and create reports of BC groundwater levels for set dates.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

Depends R (>= 2.10)

Imports bcdata (>= 0.2.4), bcmaps (>= 1.0.2), dplyr (>= 1.0.7), DT, emo (>= 0.0.0.9000), fasstr (>= 0.3.2), fs, ggplot2 (>= 3.3.5), gt (>= 0.3.1), glue (>= 1.4.2), httr (>= 1.4.2), leaflet (>= 2.0.4.1), lubridate (>= 1.7.10), patchwork (>= 1.1.1), plotly, purrr (>= 0.3.4), rappdirs (>= 0.3.3), readr (>= 1.4.0), rlang (>= 0.4.11), rmarkdown (>= 2.9), sf (>= 0.9.8), shiny, shinydashboard, shinyFiles, shinyjs, shinyWidgets, stringr (>= 1.4.0), tidyr (>= 1.1.3), sessioninfo (>= 1.2.1), readxl (>= 1.3.1)

Suggests covr, knitr, testthat (>= 3.0.0)

Remotes hadley/emo

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://steffilazerte.r-universe.dev

RemoteUrl https://github.com/bcgov/bcgwlreports

RemoteRef HEAD

RemoteSha ecfc693339815ec982bb610c5bba8746a882fb50

Contents

`bcgwlreports-package` *BC Governement Groundwater Level Reports*

Description

Create reproducible reports of current groundwater statuses.

`clean_cache` *Clean cache*

Description

Removes data cache

Usage

```
clean_cache()
```

Examples

```
# clean_cache()
```

`get_obs_in_area` *Get a list of all wells from a natural resource region or area.*

Description

Get a list of all wells from a natural resource region or area.

Usage

```
get_obs_in_area(
  nr_area = c("North Natural Resource Area", "South Coast Region",
    "South Natural Resource Area", "West Coast Region"),
  rm_well = NA
)
```

Arguments

`nr_area` List one or multiple of c("North Natural Resource Area", "South Coast Region", "South Natural Resource Area", "West Coast Region")

`rm_well` Exclude specific wells.

`gw_both_plots` *Create both percentiles and historical plot with selected observation wells*

Description

Create both percentiles and historical plot with selected observation wells

Usage

```
gw_both_plots(data, ows = NA, water_year_start = 10)
```

Arguments

`data` Data object created from `gw_data_prep()` function.

`ows` List a specific well to plot from original listed wells (exports just this well)

gw_conditions *Create a percentiles map with selected observation wells*

Description

Create a percentiles map with selected observation wells

Usage

```
gw_conditions(
  ows = NA,
  report_dates = Sys.Date(),
  n_days = 14,
  years_min = 5,
  cache_age = 7
)
```

Arguments

ows	Character vector. Observation well numbers (e.g, "OW000"). Set to NA for all wells (default).
report_dates	Character vector. Only one date for this function. Default Sys.Date().
n_days	Numeric. If there is no data on the report date chosen, this is the range of days over which to look for alternative dates with data. Defaults to 2 weeks, meaning 2 weeks before and 2 weeks after a given report date, for a total window of 4 weeks.
years_min	Numeric. Minimum number of years required to to calculate a percentiles
cache_age	Logical. Maximum age in days of cached datasets (not obs well data, but metadata related to regional maps, aquifer and wells).

gw_conditions_save *Create a percentiles map with selected observation wells*

Description

Create a percentiles map with selected observation wells

Usage

```
gw_conditions_save(
  ows = NA,
  report_dates = Sys.Date(),
  n_days = 14,
  years_min = 5,
  cache_age = 7,
  save_plots = FALSE,
  csv_dir = "",
  plots_dir = ""
)
```

Arguments

`ows` Character vector. Observation well numbers (e.g, "OW000"). Set to NA for all wells (default).

`report_dates` Character vector. Only one date for this function. Default Sys.Date().

`n_days` Numeric. If there is no data on the report date chosen, this is the range of days over which to look for alternative dates with data. Defaults to 2 weeks, meaning 2 weeks before and 2 weeks after a given report date, for a total window of 4 weeks.

`years_min` Numeric. Minimum number of years required to to calculate a percentiles

`cache_age` Logical. Maximum age in days of cached datasets (not obs well data, but metadata related to regional maps, aquifer and wells).

gw_data_prep *Prepare well data for creating table and plots*

Description

Prepare well data for creating table and plots

Usage

```
gw_data_prep(
  ows,
  report_dates = Sys.Date(),
  remarks = NULL,
  n_days = 14,
  years_min = 5,
  cache_age = 7,
  water_year_start = 10
)
```

Arguments

`ows` Character vector. Observation well numbers (e.g, "OW000")

`report_dates` Character vector. Two current dates to explore. By default a date 2 week ago and 4 weeks before that are used.

`remarks` Character / data frame. Path to file OR data frame containing remarks on specific observation wells to be included in the main summary table (see Details).

`n_days` Numeric. If there is no data on the report date chosen, this is the range of days over which to look for alternative dates with data. Defaults to 2 weeks, meaning 2 weeks before and 2 weeks after a given report date, for a total window of 4 weeks.

`years_min` Numeric. Minimum number of years required to to calculate a percentiles

`cache_age` Logical. Maximum age in days of cached datasets (not obs well data, but metadata related to regional maps, aquifer and wells).

Details

`remarks` can be a file path to a TSV (tab-separated) text file or Excel file contain columns ’ow’ and ’remarks’, or it can be a `data.frame()/tibble()` (see examples) containing the same. Note that CSV is not permitted as ’,’ is used for separating variables which can make it difficult to write out complete, complex remarks.

Examples

```
## Not run:

wells <- c('OW400')

gw_data <- gw_data_prep(ows = wells)


## End(Not run)
```

`gw_historic_data_plot` *Create historical data plot with selected observation wells*

Description

Create historical data plot with selected observation wells

Usage

```
gw_historic_data_plot(data, ows = NA)
```

Arguments

`data` Data object created from `gw_data_prep()` function.

`ows` List a specific well to plot from original listed wells (exports just this well)

gw_percentiles_details_table *Create a wells percentiles details with selected observation wells*

Description

Create a wells percentiles details with selected observation wells

Usage

```
gw_percentiles_details_table(data, gt = TRUE)
```

Arguments

data	Data object created from gw_data_prep() function.
gt	Make the table gt format (TRUE) or a regular data.frame (FALSE)

gw_percentiles_plot *Create an annual hydrograph with percentiles plot with selected observation wells*

Description

Create an annual hydrograph with percentiles plot with selected observation wells

Usage

```
gw_percentiles_plot(data, ows = NA)
```

Arguments

data	Data object created from gw_data_prep() function.
ows	List a specific well to plot from original listed wells (exports just this well)

gw_percentile_class_table *Create a percentiles class table with selected observation wells*

Description

Create a percentiles class table with selected observation wells

Usage

```
gw_percentile_class_table(data, gt = TRUE)
```

Arguments

data	Data object created from `gw_data_prep()` function.
gt	Make the table gt format (TRUE) or a regular data.frame (FALSE)

gw_percentile_map *Create a percentiles map with selected observation wells*

Description

Create a percentiles map with selected observation wells

Usage

```
gw_percentile_map(data)
```

Arguments

data	Data object created from `gw_data_prep()` function.

`gw_wells_below_normal_table` *Create a wells below normal table with selected observation wells*

Description

Create a wells below normal table with selected observation wells

Usage

```
gw_wells_below_normal_table(
  data,
  which = c("totals", "hydraulic_connectivity", "type")[1],
  gt = TRUE
)
```

Arguments

`data`	Data object created from `gw_data_prep()` function.
`which`	Which group to filter by: "totals" for all, 'hydraulic_connectivity' or 'type' for aquifer type.
`gt`	Make the table gt format (TRUE) or a regular data.frame (FALSE)

`run_shiny` *Launch the shiny app*

Description

Launch shiny app

Usage

```
run_shiny()
```

well_report *Compile report*

Description

Compile report

Usage

```
well_report(
  ows,
  name = "report",
 report_dates = c(Sys.Date() - lubridate::weeks(2), Sys.Date() - lubridate::weeks(4)),
  title = NULL,
  description = NULL,
  remarks = NULL,
  n_days = 13,
  years_min = 5,
  out_dir = ".",
  cache_age = 7,
  water_year_start = 10
)
```

Arguments

ows	Character vector. Observation well numbers (e.g, "OW000")
name	Character string. Short text to name the file. Will become name_YYYY_MM_DD.html
report_dates	Character vector. Two current dates to explore. By default a date 2 week ago and 4 weeks before that are used.
title	Character. Title of the report.
description	Character. Descriptive paragraph to place at the start.
remarks	Character / data frame. Path to file OR data frame containing remarks on specific observation wells to be included in the main summary table (see Details).
n_days	Numeric. If there is no data on the report date chosen, this is the range of days over which to look for alternative dates with data. Defaults to 2 weeks, meaning 2 weeks before and 2 weeks after a given report date, for a total window of 4 weeks.
years_min	Numeric. Minimum number of years required to to calculate a percentiles
out_dir	Character. Location of output report. Defaults to working directory.
cache_age	Logical. Maximum age in days of cached datasets (not obs well data, but metadata related to regional maps, aquifer and wells).

Details

remarks can be a file path to a TSV (tab-separated) text file or Excel file contain columns 'ow' and 'remarks', or it can be a data.frame()/tibble() (see examples) containing the same. Note that CSV is not permitted as ',' is used for separating variables which can make it difficult to write out complete, complex remarks.

Examples

```
## Not run:

well_report(ows = c("OW008", "OW217", "OW377", "OW197"))

# If short, easiest to add remarks in script:

library(dplyr)

remarks <- tribble(~ow,     ~remarks,
                   "OW377", "Construction in the area disrupting measurements",
                   "OW008", "No problems")

well_report(ows = c("OW008", "OW217", "OW377", "OW197"),
            remarks = remarks)

# Or load from a file
library(readr)
write_tsv(remarks, "remarks.txt")
check_remarks(remarks = "remarks.txt")

## End(Not run)
```

Index