Package: dwc.wells (via r-universe)

October 29, 2024

Title A Package for Condition Predictions for Drinking Water Wells

Version 0.2.0

Description This package allows to predict the condition of a drinking water well based on ML models. The models are trained with results from pump tests and a large set of input variables e.g. the well material, the age and the number of regenerations.

License MIT + file LICENSE

URL https://github.com/KWB-R/dwc.wells

BugReports https://github.com/KWB-R/dwc.wells/issues

Depends R (>= 3.50)

Imports corrplot, dplyr, forcats, ggplot2, kwb.db, kwb.utils, lsr, lubridate, magrittr, odbc32, parsnip, RColorBrewer, readr, readxl, rlang, rsample, scales, sema.berlin.utils, stringr, tibble, tidyr, tidyselect, yardstick, zoo

Suggests caret, covr, cowplot, doParallel, DT, knitr, parallel, plotly, randomForest, rmarkdown, rpart.plot, tidymodels, usethis, xgboost

VignetteBuilder knitr

Remotes github::kwb-r/kwb.db, github::kwb-r/kwb.utils, github::kwb-r/sema.berlin.utils

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

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

RemoteUrl https://github.com/KWB-R/dwc.wells

RemoteRef HEAD

RemoteSha 45e8670647c4771fe70d59db0f7cfd1e80242361

Contents

`chi2.CramersV.test` *Title*

Description

Title

Usage

```
chi2.CramersV.test(data)
```

Arguments

`data`	data frame on which to perform Chi-2-test

`classify_Qs` *Transfer Qs_rel into binary factor with low and high specific capacity*

Description

Transfer Qs_rel into binary factor with low and high specific capacity

Usage

```
classify_Qs(x, split_point = 80, class_names = c("low", "high"))
```

Arguments

`x`	vector of Qs_rel values
`split_point`	threshold for classifying numeric Qs_rel values, default: 80
`class_names`	class names, default: c("low", "high")

`combine_pump_test_and_Q_monitoring_data` *Combined Pumptest and Q Monitoring Dataset*

Description

Combined Pumptest and Q Monitoring Dataset

Usage

```
combine_pump_test_and_Q_monitoring_data(
  df_pump_tests_tidy,
  df_Q_monitoring,
  pump_test_vars
)
```

Arguments

`df_pump_tests_tidy`	df_pump_tests_tidy
`df_Q_monitoring`	df_Q_monitoring
`pump_test_vars`	default: `get_pump_test_vars`

Value

combined pumptest and Q monitoring dataset

correlation_plot *plots Qs_rel vs. input variable as box plot (categorical input variable) or scatterplot (numerical input variable)*

Description

plots Qs_rel vs. input variable as box plot (categorical input variable) or scatterplot (numerical input variable)

Usage

```
correlation_plot(df, x, y = "Qs_rel", title = gsub("_", " ", x))
```

Arguments

`df`	data frame
`x`	column name of x variable"
`y`	column name of y variable (default Qs_rel")
`title`	plot title

extdata_file *Get Path to File in This Package*

Description

Get Path to File in This Package

Usage

```
extdata_file(...)
```

Arguments

`...`	parts of path passed to `system.file`

`fill_up_na_with_median_from_lookup` *Fill up NA values with median of lookup table*

Description

Fill up NA values with median of lookup table

Usage

```
fill_up_na_with_median_from_lookup(df, df_lookup, matching_id = "well_id")
```

Arguments

`df`	data frame with NA values
`df_lookup`	data frame to calculate median values
`matching_id`	column with ids for which median should be calculated

`frequency_table` *calculate absolute and relative frequencies of categorical varables*

Description

calculate absolute and relative frequencies of categorical varables

Usage

```
frequency_table(x, perc_digits = 1, sort_freq = FALSE)
```

Arguments

`x`	vector with categorical variable
`perc_digits`	number of decimal digits for percentages, default = 1
`sort_freq`	sort according to frequency counts, logical, default: TRUE

get_pump_test_vars *Get Default Pump Test Variables*

Description

Get Default Pump Test Variables

Usage

```
get_pump_test_vars()
```

Value

vector with column names of pump test variables

Examples

```
get_pump_test_vars()
```

get_W_static_data *Get W_static measurement data from Neubaupumpversuche, Kurzpumpversuche and other sources*

Description

Get W_static measurement data from Neubaupumpversuche, Kurzpumpversuche and other sources

Usage

```
get_W_static_data(path, renamings, df_wells)
```

Arguments

path	path to static water level data (csv-file)
renamings	list with renamings
df_wells	data frame with prepared well data

`interpolate_and_fill` *Interpolate and fill up static water level*

Description

Interpolate and fill up static water level

Usage

```
interpolate_and_fill(df, x_col, y_col, group_by_col, origin_col)
```

Arguments

`df`	data frame
`x_col`	x column, e.g. date, to be used for interpolation
`y_col`	y column, e.g. measured values, to be used for interpolation
`group_by_col`	grouping variable within which interpolation is done
`origin_col`	already existing or to be created column with type of value

`interpolate_Qs` *Interpolates Qs time series data to a given time interval*

Description

Interpolates Qs time series data to a given time interval

Usage

```
interpolate_Qs(df, interval_days = 1)
```

Arguments

`df`	data frame with date and Qs measurements
`interval_days`	interval for interpolation

`load_renamings_csv` *load renaming table from original excel file*

Description

load renaming table from original excel file

Usage

```
load_renamings_csv(infile)
```

Arguments

`infile`	full path to excel file

`load_renamings_excel` *load renaming table from original excel file*

Description

load renaming table from original excel file

Usage

```
load_renamings_excel(
  infile,
  sheet = "DATEN",
  old_name_col = "Feld",
  new_name_col = "Parametername-R"
)
```

Arguments

`infile`	full path to excel file
`sheet`	sheet name
`old_name_col`	name of column with original variable names
`new_name_col`	name of column with new variable names

`model_data_reduced` *Input Data for Well Capacity Prediction*

Description

A reduced dataset for well capacity prediction created with R script in /data-raw/model_data.R

Usage

```
model_data_reduced
```

Format

A data.frame with 6308 rows and 27 variables:

well_id well id, for info

date date of capacity measurement, for info

key measurement key, e.g. operational_start, pump_test_1, pump_test_2, for info

Qs_rel specific capacity of well relative to operational start condition, output

days_since_operational_start days since operational start, redundant

well_age_years years since operationa start, input, numeric

construction_year year of well construction

screen_material screen material

diameter well diameter (mm)

drilling_method drilling_method

admissible_discharge allowed pumping rate

operational_start.Qs initial Qs at construction

aquifer_coverage confined / unconfined

W_static.sd standard deviation of static water level

surface_water.distance distance to surface water

n_rehab number of well rehabilitations

time_since_rehab_years time since last well rehabilitation in years

volume_m3_d.mean mean daily abstraction volume (m3)

quality.EC water quality: electical conductivity (us/cm)

quality.D0 water quality: dissolved oxygen (mg/l)

quality.Temp water quality: temperature (C)

quality.pH water quality: pH

quality.Redox water quality: electical conductivity (us/cm)

quality.Fe_tot water quality: dissolved oxygen (mg/l)

quality.Mn water quality: Mn (mg/l)

quality.NO3 water quality: NO3 (mg/l)

quality.PO4 water quality: PO4 (mg/l)

quality.SO4 water quality: SO4 (mg/l)

quality.TSS water quality: Total Suspended Solids (mg/l)

`paste_percent` *Paste percent sign to numbers*

Description

Paste percent sign to numbers

Usage

```
paste_percent(x)
```

Arguments

`x`	numeric vector

`plot_distribution` *plot frequency distribution of numerical variable*

Description

plot frequency distribution of numerical variable

Usage

```
plot_distribution(
  Data,
  variable,
  binwidth = NULL,
  title,
  vertical_x_axis_labels = TRUE,
  boundary = 0
)
```

Arguments

`Data`	Data to be plotted
`variable`	variable
`binwidth`	binwidrh
`title`	plot title
`vertical_x_axis_labels`	should x-axis labels be ploted vertically (TRUE / FALSE)
`boundary`	left boundary of bars, default: 0

`plot_frequencies` *plot frequency distribution of factor variable*

Description

plot frequency distribution of factor variable

Usage

```
plot_frequencies(
  Data,
  variable,
  title = variable,
  offset_perc_labels = 0.1,
  size_perc_labels = 3,
  vertical_x_axis_labels = TRUE
)
```

Arguments

`Data`	Data to be plotted
`variable`	variable
`title`	plot title
`offset_perc_labels`	distance of labels from bars
`size_perc_labels`	size of percent labels
`vertical_x_axis_labels`	should x-axis labels be ploted vertically (TRUE / FALSE)

`prepare_pump_test_data` *prepare pump test data with one row per Qs-measurement + rehab history*

Description

prepare pump test data with one row per Qs-measurement + rehab history

Usage

```
prepare_pump_test_data(path, renamings, df_wells, pump_test_vars)
```

Arguments

`path`	path to pump test data
`renamings`	list with renamings
`df_wells`	prepared data frame with well characteristics
`pump_test_vars`	default: `get_pump_test_vars`

---

prepare_pump_test_data_1

Prepare pump test data in wide format

Description

Steps: i) read, rename and clean data, ii) correct wrong pump test dates, iii) fill up missing pump test dates, iv) get information for replaced wells, v) calculate Qs and Qs_rel, vi) determine action type, vii) select columns

Usage

```
prepare_pump_test_data_1(path, renamings, df_wells)
```

Arguments

`path`	path to pump test data
`renamings`	list with renamings
`df_wells`	prepared data frame with well characteristics

---

prepare_pump_test_data_2

reformats untidy pump test data from wide into long format

Description

reformats untidy pump test data from wide into long format

Usage

```
prepare_pump_test_data_2(
  df_pump_tests_untidy,
  df_wells,
  pump_test_vars = get_pump_test_vars()
)
```

Arguments

`df_pump_tests_untidy`	pump test data in wide format
`df_wells`	prepared data frame with well characteristics
`pump_test_vars`	default: `get_pump_test_vars`

`prepare_quality_data` *Prepare Quality Data*

Description

Prepare Quality Data

Usage

```
prepare_quality_data(path, renamings)
```

Arguments

`path`	path
`renamings`	renamings

Value

prepared quality day

`prepare_volume_data` *Prepare Volume Data*

Description

Prepare Volume Data

Usage

```
prepare_volume_data(path, renamings, df_wells)
```

Arguments

`path`	path
`renamings`	renamings
`df_wells`	df_wells

Value

Prepared volume data

Qs_heatmap_plot *Heatmap / raster plot for Qs values over time with each well as one line*

Description

Heatmap / raster plot for Qs values over time with each well as one line

Usage

```
Qs_heatmap_plot(
  df,
  colours,
  dummy_labels,
  date_limits,
  title,
  n_wells_per_page
)
```

Arguments

df	data frame with date, well_id, Qs_rel
colours	3 colours for low, middle and high colour limits
dummy_labels	dummy labels if there are less wells than expected
date_limits	vector with two date strings in format "yyyy-mm-dd"
title	plot title
n_wells_per_page	number of wells do be shown

read_csv *read csv data file exported by Sebastian Schimmelpfennig from db2*

Description

read csv data file exported by Sebastian Schimmelpfennig from db2

Usage

```
read_csv(
  file,
  header = TRUE,
  fileEncoding = "UTF-8",
  skip = 2,
  dec = ".",
  sep = "\t",
  na.strings = "(null)"
)
```

Arguments

`file`	path to csv file
`header`	logical, default = TRUE
`fileEncoding`	default = UTF-8
`skip`	number of rows to skip, default = 2
`dec`	decimal separator, default = '.'
`sep`	columns separator, default = 'tab'
`na.strings`	string that represents NA, default = "(null)"

`read_ms_access` *read table from MS Access data base via odbc connection under 64-bit-R*

Description

read table from MS Access data base via odbc connection under 64-bit-R

Usage

```
read_ms_access(path_db, tbl_name)
```

Arguments

`path_db`	full path to database
`tbl_name`	name of database table to be read

`read_select_rename` *read table from MS Access data base; select and rename columns as defined in renamings table ('old_name' -> 'new_name')*

Description

read table from MS Access data base; select and rename columns as defined in renamings table ('old_name' -> 'new_name')

Usage

```
read_select_rename(
  path_db,
  tbl_name,
  renamings,
  old_name_col = "old_name",
  new_name_col = "new_name"
)
```

Arguments

path_db	full path to database
tbl_name	name of database table to be read
renamings	name of data frame with renamings
old_name_col	name of column with original variable names
new_name_col	name of column with new variable names

rename_values *rename values of a character vector according to renamings table*

Description

rename values of a character vector according to renamings table

Usage

```
rename_values(
  x,
  renamings,
  old_name_col = "old_name",
  new_name_col = "new_name"
)
```

Arguments

x	character vector
renamings	data frame consisting of old and new names
old_name_col	name of column with original variable names
new_name_col	name of column with new variable names

replace_na_with_median *Replace NAs with median*

Description

Replace NAs with median

Usage

```
replace_na_with_median(x)
```

Arguments

x	vector, for which NA should be replaced

save_data *Save data frame in different formats: csv, RData, rds*

Description

Save data frame in different formats: csv, RData, rds

Usage

```
save_data(Data, path, filename, formats = c("csv", "RData", "rds"))
```

Arguments

`Data`	data frame
`path`	out path for saving data
`filename`	core of file name
`formats`	export formats: "csv", "RData", "rds" or several using 'c'

scatterplot *scatterplot for comparing numeric predictions with observations*

Description

scatterplot for comparing numeric predictions with observations

Usage

```
scatterplot(df_pred, lines_80perc = FALSE, alpha = 1, pointsize = 1)
```

Arguments

`df_pred`	data frame obtained with tidymodels::collect_predictions() with columns Qs_rel and .pred
`lines_80perc`	logical value; shout 80%-lines be drawn?; default = FALSE
`alpha`	alpha value for point of colours, default: 1
`pointsize`	size value for points, default: 1

`select_rename_cols` *selects and renames columns from a data frame according to a reference table*

Description

selects and renames columns from a data frame according to a reference table

Usage

```
select_rename_cols(
  df,
  renamings,
  old_name_col = "old_name",
  new_name_col = "new_name"
)
```

Arguments

`df`	data frame with cols to be renamed
`renamings`	name of data frame with renamings
`old_name_col`	name of column with original variable names
`new_name_col`	name of column with new variable names

`summarise_marginal_factor_levels` *summarise factor levels with relative frequency below a threshold*

Description

summarise factor levels with relative frequency below a threshold

Usage

```
summarise_marginal_factor_levels(x, perc_threshold, marginal_name)
```

Arguments

`x`	factor variable
`perc_threshold`	percentage threshold under which levels will be summarised
`marginal_name`	for new summary factor level

`tidy_factor` *turn character into factor, sort factor levels and replace NA level*

Description

turn character into factor, sort factor levels and replace NA level

Usage

```
tidy_factor(x, level_sorting = c("frequency", "alphabet")[1])
```

Arguments

`x`	character vector to be turned to factor
`level_sorting`	sorting of factor levels; two options: "frequency" (default) and "alphabet"; level "Unbekannt" is always always at the end

Index