Package: kwb.flusshygiene (via r-universe)

August 24, 2024

Title Functions used within FLUSSHYGIENE project (BMBF)

Version 0.3.0

Description Easy and transferable functions for creating and managing models for hygiene data in rivers. This package is developed within the FLUSSHYGIENE project. See https://bmbf.nawam-rewam.de/en/projekt/flusshygiene/ for details.

License MIT + file LICENSE

URL `https://github.com/KWB-R/kwb.flusshygiene`

BugReports `https://github.com/KWB-R/kwb.flusshygiene/issues`

Depends R (>= 3.4), stats, utils, graphics, grDevices

Imports MASS (>= 7.3-47), dplyr (>= 0.7.4), tidyr (>= 0.7.2), tibble (>= 1.3.4), readr (>= 1.1.1), rlang (>= 0.4.0), lubridate (>= 1.7.1), ggplot2 (>= 2.2.1), rstanarm (>= 2.15.3), kwb.utils (>= 0.4.3), purrr (>= 0.3.2)

Suggests knitr (>= 1.23), readxl (>= 1.3.1), rmarkdown (>= 1.13), testthat (>= 2.2.1)

VignetteBuilder knitr

Remotes github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

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

RemoteUrl https://github.com/KWB-R/kwb.flusshygiene

RemoteRef HEAD

RemoteSha b9fee176e392f7932673e8efb092114cfdcbd9d0

Contents

assess_bathing_quality_eu 2
build_model 3
build_new_model 4
choose_model 4
correlation_scatterplot 5
get_mpn_ci 6
get_paths 6
import_riverdata 7
kwb.flusshygiene 8
plot_data_overview 9
plot_hygiene_overview 9
plot_predicted_quality 10
plot_q_overview 10
plot_rain_overview 11
plot_stan_model 11
predict_quality 12
readTableData 13
river_model_prediction 13
unroll_physical_data 14

Index **15**

`assess_bathing_quality_eu`

Bathing quality assessment (EU)

Description

Computes the quality assessment according to european bathing directive 2006/7/EC from E.Coli values. The four possible quality levels are: excellent, good, sufficient and poor

Usage

```
assess_bathing_quality_eu(e.coli, log = TRUE)
```

Arguments

`e.coli`	A numeric vector with e.coli values
`log`	logical. Are the values log-values?

Value

Returns a single factor with all quality levels

`build_model` *Build a Model for E.Coli*

Description

Functions for modelbuilding \n `build_model`: takes the riverdata, handles the other functions and invokes `stan_lm`

Usage

```
build_model(riverdata, variables = ask_for_variables(riverdata),
  with_interaction = TRUE)

ask_for_variables(riverdata)

process_model_riverdata(riverdata, variables)

create_formula(variables, with_interaction = FALSE)
```

Arguments

`riverdata`	a list with riverdata (hygiene + physical data)
`variables`	character. Selected variables for the model
`with_interaction`	logical. Formula with interactions? Default set to TRUE

Details

Build the model from hygiene data and physical data like flow, rain, wwtp. Asks for user input to select variables. Computes the data.frame with data for hygiene and chosen variables and creates a formula of the form: `Q*(K + R)` while multiple Qs will be multiplied, multiple Ks and Rs will be added.

Value

Returns a model of the riverdata.

Returns a character-vector with the chosen model variables

Returns a data.frame with data for hygiene and chosen variables

Returns parsed model-formula. (Like model$formula)

Functions

- `ask_for_variables`: Internal function. Quite time consuming
- `process_model_riverdata`: Internal usage
- `create_formula`: Internal usage

Examples

```
## Not run: variables <- c("e.coli","q_havel",...)
lm(formula = eval(create_formula(variables)),
data = process_model_riverdata(riverdata, variables))
## End(Not run)

create_formula(c("log_e.coli","q_havel","ka_ruhleben","r_berlin"))
create_formula(c("e.coli","r_mitte","r_charlottenburg","r_spandau"))
```

---

build_new_model *Create a new Model*

Description

Main function for creation, object handling and saving of new models. The models will be saved in a own subdirectory as r-objects as a side-effect. The models will not be returned. They have to be loaded by other functions.

Usage

```
build_new_model(river)
```

Arguments

`river`	character. river you want to build a model on

Value

This function returns merely a message what happend.

---

choose_model *Set User Input for a List of Models*

Description

Choose model asks the user inside the console for input. Options are Exit, New Model, or one of a list of existing models. If no integer number was presented by the user an ERROR message will be created but no ERROR will be thrown. This way this can be inserted inside a loop.

Usage

```
choose_model(rivermodels)
```

Arguments

`rivermodels`	A list of named models

Value

Returns the user input as character vector, or an ERROR message.

Examples

```
choose_model(list())
choose_model(list(fake_river_model = 1))
```

`correlation_scatterplot`
Scatterplotmatrix of similar Variables to E.Coli

Description

Takes similar named variables and produces a matrix of scatterplots and their correlation coefficients to E.Coli.

Usage

```
correlation_scatterplot(df, ...)

correlation_values(df, ...)
```

Arguments

`df`	data.frame with data for e.coli and chosen variables in lagdays
`...`	Arguments passed to `stats::cor`

Value

Plotting function. Returns a plot.

Returns correlation values.

Functions

- `correlation_values`: Internal function

Examples

```
correlation_values(data.frame(datum = rep("egal",10), e.coli = 1:10, var = 1:10), variable = "var")
```

get_mpn_ci *Get MPN Confidence Intervals for E.Coli*

Description

Lookup laboratory tables for MPN values for E.Coli to get upper and lower 0.95 confidence interval for the given values. If value is not directly found in table it will be generated by interpolating nearest neighbors.

Usage

```
get_mpn_ci(e.coli)
```

Arguments

`e.coli`	numeric. A vector for e.coli values

Value

A data.frame with 3 columns: e.coli, lo, up

Examples

```
## Not run:
print(get_mpn_ci(c(15,30,35,60,61,71,120,1959,25000,369990)))

## End(Not run)
```

get_paths *Get List of Paths used in the Flusshygiene Project*

Description

Get List of Paths used in the Flusshygiene Project

Usage

```
get_paths(resolve = TRUE, ...)
```

Arguments

`resolve`	if `TRUE` (default) path placeholders are resolved
`...`	arguments passed to `resolve` if `resolve` is `TRUE`

Examples

```
## Not run:
paths <- get_paths()

# Paths to the different work package folders
paths$ap2
paths$ap3
paths$ap4

# What tables are contained in the ODM database?
kwb.db::hsTables(paths$odm)

# Get all Flusshygiene data into one data frame
data <- kwb.ogre.model::get_lab_values(paths$odm)

## End(Not run)
```

import_riverdata *Read existing River Data*

Description

Read existing, preprocessed csv files with first column datetime and other columns variable information.

Usage

```
import_riverdata(path)
```

Arguments

path	character-string to a DATA_preprocessed_csv directory

Value

Returns a list of data.frames containing the river data `kwb.flusshygiene` *A package for creating and handling river hygiene models*

Description

The kwb.flusshygiene package provides functions in three major categories: model handling, model creation and model prediction.

Model handling

`river_model_prediction` is the main function in this package. It uses all of the following functions from within.

`get_paths` reads a serverpath library for easy directory accessing

`search_existing_models` searches saved R-objects in the river directories.

Model creation

`build_new_model` again a overhead function for model creation. Asks the user whether or not the new model shall be saved as R-object.

`import_riverdata` reads all river data from a directory.

`build_model` is a small function handling model creation and invokes `stan_lm` for model building.

`ask_for_variables` asks the user which variables shall be included in the model. Creates plots as a side effect.

`process_model_riverdata` processes a data.frame with the necessary data for the data argument in `stan_lm`

`create_formula` creates a hygiene formula out of the variables with the form `e.coli ~ Q * (R + Ka)`

Model prediction

`predict_quality` is the overhead function for the prediction. It also invokes `posterior_predict.stanreg`

`get_newdata` gathers the latest data for prediction.

`print_latest` prints the prediction of the latest day.

`plot_predicted_quality` plots a whole season with quality assessment.

Utility functions

`unroll_physical_data` unrolls a list with data with lagday combinations to 5 days (default)

`correlation_scatterplot` plots a scatterplot matrix of the unrolled physical data together with correlation values.

Plotting functions

`plot_stan_model`	plots a stan_lm posterior predction with quality assessment.
`plot_data_overview`	plot data overview
`plot_hygiene_overview`	a statistical hygiene data overview
`plot_q_overview`	plot a overview of all q values
`plot_rain_overview`	plot a monthly overview of all rain gauges

`plot_data_overview` *Plot Data Timeline Overview*

Description

Creates a plot with segments or points of the data availability.

Usage

```
plot_data_overview(riverdata, type = "segment")
```

Arguments

`riverdata`	A list of hygiene and physical data of the river
`type`	Either "segment" or "point" for more precise information

Value

Returns a plot

`plot_hygiene_overview` *Plot Hygiene Overview*

Description

Creates a plot with three graphs: Histogramm of all e.coli values, a density curve of the last 16 values, and a boxplot of all values again

Usage

```
plot_hygiene_overview(hygiene_df)
```

Arguments

`hygiene_df`	A data.frame with the hygiene data of a given river

Value

Returns a plot

`plot_predicted_quality` *Plot Quality*

Description

Window function for plot_stan_model

Usage

```
plot_predicted_quality(model, prediction, ...)
```

Arguments

`model`	stan.lm model for the river
`prediction`	list of season, ppd of predcit and ppd of means
`...`	Further parameter passed to plot.default

Value

Plotting function. Returns a plot.

`plot_q_overview` *Plot Flowing Conditions*

Description

Creates a plot with the standard flowing conditions over the year. The data of all years will be taken into account.

Usage

```
plot_q_overview(q_df)
```

Arguments

`q_df`	The data.frame with 2 columns: datum and q

Value

Returns a plot

`plot_rain_overview` *Plot Monthly Rain Summary*

Description

Creates a plot with a monthly summary overview over the different rain sites

Usage

```
plot_rain_overview(df)
```

Arguments

`df`	A data frame with different rain gauges.

Value

Returns a plot

`plot_stan_model` *Plot Model Prediction with Quality Assessment*

Description

Plots a sample of posterior predictions and means. Furthermore colours an hygiene quality assessment as background (see EU Bathing Water Directive) Dark blue means excellent quality. Steelblue means good quality. Yellow means sufficient quality. Red means insufficient quality.

Usage

```
plot_stan_model(timestamp, predict, linpred, log = FALSE, q90, q95,
  nlines = 250, nlinesCenter = 100, ...)
```

Arguments

`timestamp`	POSIX. The x-axis timestamp
`predict`	ppd. The posterior prediction of the model
`linpred`	ppd. The linpred (predicted means) of the model
`log`	logical. Is E.Coli log01-transformed?
`q90`	numeric. The 90. percentile of predict.
`q95`	numeric. The 95. precentile of predict.
`nlines`	numeric. How many lines for posterion predictions?
`nlinesCenter`	numeric. How many lines for predicted means?
`...`	Further parameters for plot.default

Value

Plotting function. Returns a plot.

`predict_quality` *Predict Hygiene Quality*

Description

Main function for invoking and object handling. E.Coli hygiene models will be used to predict hygiene quality on differnt scopes.

Usage

```
predict_quality(model, river_dir, output = "season")

get_newdata(variables, river_dir)

print_latest(model, newdata)

get_latest_season(newdata)
```

Arguments

`model`	stan_lm. A model of e.coli concentration in given river
`river_dir`	character. Path to river-data for up-to-date predictions.
`output`	character. `"season"` will return a list with prediction, `"latest"` will return console output
`variables`	character. A vector with all variables used in the model
`newdata`	data.frame with physical data used in the model

Value

Returns a list of physical data and prediction and linpred from model

Returns a data.frame with the merged data found

Functions

- `get_newdata`: Internal Usage
- `print_latest`: Internal Usage
- `get_latest_season`: Internal Usage

readTableData *Read Data for ODM Tables*

Description

Read data for ODM tables from CSV files stored in the package

Usage

```
readTableData(sourcedir = system.file("extdata", "ODM", package =
  "kwb.flusshygiene"))
```

Arguments

sourcedir	path to input directory

Value

list of data frames

river_model_prediction *Programm for Model Handling and Prediction*

Description

This function is the front-end for model search on the server, model building with existing data, or prediction with new data. It invokes all other functions and handles their objects. It is a main function.

Usage

```
river_model_prediction(river)

search_existing_models(river_dir)
```

Arguments

river	character. The desired river, like "isar".
river_dir	character. Path to server and river directory

Value

(invisible) The data.frame returned by the prediction plus a date column for easy plotting.

Returns a list with the existing models for that river (empty if no model was found).

Functions

- `search_existing_models`: directory searching

Examples

```
river_model_prediction(river = "isar")



serverpath <- "//poseidon/projekte$/SUW_Department/Projects/FLUSSHYGIENE/Data-Work packages/Daten"
river_dir <- search_existing_river_dir(river = "isar", server = serverpath)
search_existing_models(river_dir = river_dir)
```

`unroll_physical_data` *Unroll Lagdays of Data*

Description

Unrolls the lagdays of data.frames.

Usage

```
unroll_physical_data(physical_data)

unroll_lagdays(df, n = 5)
```

Arguments

`physical_data`	list of river data (without hygiene)
`df`	data.frame of 2 columns: datum and var
`n`	numeric. unto to which day shall be lagged behind?

Value

Returns a list of data.frames for each variable. The data.frames contain the unrolled lagdays (with maxday = 5, length(df) == 17)

Functions

- `unroll_lagdays`: Internal usage mostly

Examples

```
df1 <- data.frame(datum = rep("egal", 25), var = 1:25)
df2 <- data.frame(datum = rep("egal", 25), var2 = 51:75, var3 = 101:125)
unroll_lagdays(df1)
summary(unroll_physical_data(list(var1 = df1, var2 = df2)))
```

Index

ask_for_variables, *8*
ask_for_variables *(*build_model*)*, 3
assess_bathing_quality_eu, 2

build_model, 3, *8*
build_new_model, 4, *8*

choose_model, 4
correlation_scatterplot, 5, *8*
correlation_values
*(*correlation_scatterplot*)*, 5
create_formula, *8*
create_formula *(*build_model*)*, 3

get_latest_season *(*predict_quality*)*, 12
get_mpn_ci, 6
get_newdata, *8*
get_newdata *(*predict_quality*)*, 12
get_paths, 6, *8*

import_riverdata, 7, *8*

kwb.flusshygiene, 8
kwb.flusshygiene-package
*(*kwb.flusshygiene*)*, 8

plot_data_overview, *9*, 9
plot_hygiene_overview, *9*, 9
plot_predicted_quality, *8*, 10
plot_q_overview, *9*, 10
plot_rain_overview, *9*, 11
plot_stan_model, *9, 10*, 11
posterior_predict.stanreg, *8*
predict_quality, *8*, 12
print_latest, *8*
print_latest *(*predict_quality*)*, 12
process_model_riverdata, *8*
process_model_riverdata *(*build_model*)*, 3

readTableData, 13
resolve, *6*
river_model_prediction, *8*, 13

search_existing_models, *8*
search_existing_models
*(*river_model_prediction*)*, 13
stan_lm, *3*, *8*

unroll_lagdays *(*unroll_physical_data*)*,
14
unroll_physical_data, *8*, 14