Package ‘gaseous’

December 11, 2023

Type Package

Title Greenhouse Gas Data Workflow Functions for the Margenot Lab

Version 0.1.0

Author Ezra Moses

Maintainer Ezra Moses <moses8@illinois.edu>

Description Contains functions and a project template specifically designed for greenhouse gas data retrieved from the Gasmet. For use within the lab of Dr. Andrew Margenot at the University of Illinois Urbana-Champaign.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports conflicted,
dplyr,
purrr,
stringr,
tibble,
cowplot,
ggpubr,
readxl

RoxygenNote 7.2.3

URL https://github.com/redseasoils/gaseous

BugReports https://github.com/redseasoils/gaseous/issues

R topics documented:

`add_missing_rows` *Add rows to data frame to make missing values explicit*

Description

Add rows to data frame to make missing values explicit

Usage

```
add_missing_rows(
  data,
  site_var = site,
  plot_var = plot,
  date_var = Date,
  year_var = year,
  plots_unique_by = c(year, site),
 plots_unique_all = paste(treatment_key$year, treatment_key$site, treatment_key$plot),
  addl_id_vars = c(treatment, block)
)
```

Arguments

`data` A data frame

`site_var` Name of column in `data` containing site names. Defaults to `site`.

`plot_var` Name of column in `data` containing plot IDs. Defaults to `plot`.

`date_var` Name of column in `data` containing sampling dates. Defaults to `Date`.

`year_var` Name of column in `data` containing sampling year. Defaults to `year`.

`plots_unique_by`
Names of columns in `data` within which plot IDs are unique. Defaults to `c(year, site)`.

`plots_unique_all`
Vector of space-separated strings containing all unique plots preceded by their corresponding levels of `plots_unique_by`. Defaults to `paste(treatment_key$year,treatment_ke`
`treatment_key$plot)`.

`addl_id_vars` Additional names of ID columns (those not already listed in other arguments) in `data`. The values of these variables will be filled in added ’missing’ rows in the result. Set to `NULL` if there are no additional ID variables to be filled in. Defaults to c(treatment, block).

Value

`data` with additional rows of previously missing levels of specified variable combinations.

`assumptions_plot_sub` *Make Plot Subtitle Containing Levene and Shapiro-Wilk Test p-values*

Description

Make Plot Subtitle Containing Levene and Shapiro-Wilk Test p-values

Usage

```
assumptions_plot_sub(
  data,
  var,
  trans_label,
  levene_label = "levene",
  shapiro_label = "shapiro",
  var_then_label = TRUE,
  sep = "_"
)
```

Arguments

`data` Data frame containing columns with Levene and Shapiro p values. There must be only one unique p-value for each test within the data frame or current group.

`var` Column name of variable for which ANOVA assumptions are being tested. E.g. `co2`

`trans_label` Optional. String describing data transformation performed on `var` prior to assumptions checks. E.g. ’log’.

levene_label — String pattern which is appended on column names containing the Levene test p-value. If `var_then_label` is `TRUE` (the default), the name of the Levene p-value column should be `paste0(var, sep, levene_label)`. Defaults to.

shapiro_label — String pattern which is appended on column names containing the Shapiro-Wilk test p-value. If `var_then_label` is `TRUE` (the default), the name of the Shapiro p-value column should be `paste0(var, sep, shapiro_label)`. Defaults to `"shapiro"`.

var_then_label — Logical. Are column names containing p-values formatted so that the name specified in `var` comes before the label specified in `levene_label` and `shapiro_label`, separated by `sep`? Defaults to `TRUE`.

sep — Character pattern separating `var` from `levene_label` or `shapiro_label` in names of columns containing p-values. Defaults to "_".

Value

A string that can be used as a formatted plot subtitle.

attr_new_col *Data Attributes*

Description

Data attributes are used in this workflow to record the reason(s) certain data points are marked for exclusion from modeling. These attributes are recorded for each gas in a single column, usually named `attributes`, created with `attr_new_col()`. `attr_update()` is used to add attributes to this column. `attr_show_codes()` prints each unique attribute code and its meaning to the console.

Usage

```
attr_new_col(
  data,
  attr_var = attributes,
  prefixes = c("all", "co2", "n2o", "ch4", "nh3"),
  sep = ","
)

attr_show_codes()

attr_update(data, attr_var = attributes, prefix, attr_code, sep = ",")
```

Arguments

data — A data frame

attr_var — Name of column in `data` where attributes are stored. Defaults to `attributes`.

prefixes — Character. Vector of prefixes to assign in new attribute column with `attr_new_col()`. Defaults to `c('all', 'co2','n2o', 'ch4', 'nh3')`.

sep — String that separates attribute elements in `attr_var`. Defaults to `','`.

prefix — Character. Unique prefix preceding attribute codes to be updated with `attr_update()`.

attr_code — Character. Attribute code to be appended with `attr_update()`.

Value

`attr_new_col()` and `attr_update()` return a data frame with new or updated `attr_var`, respectively. `attr_show_codes()` returns a message to the console.

Examples

```
attr_show_codes()
dat <- data.frame(x = rnorm(3))
dat <- attr_new_col(dat)
attr_update(dat, prefix = 'co2', attr_code = '02')
```

`build_gas_lmer` *Build linear mixed effect models for GHG data*

Description

Build `lmer()` models on multiple gas variables simultaneously.

Usage

```
build_gas_lmer(data, formula, gas_vars = c(co2, n2o, ch4, nh3))
```

Arguments

`data`	A data frame
`gas_vars`	Names of columns in `data` containing gas data to be substituted in `formula`. Each name should be an unquoted expression. Defaults to `c(co2, n2o, ch4, nh3)`.

Value

A list of models

`build_manual` *Build and Prep Manual for Export*

Description

For internal use only. Builds manual to inst/instructions and renames it to match version annotation of pptm instructions.

Usage

```
build_manual()
```

Value

Nothing in memory.

calculate_gas_loads *Calculate cumulative per-day gas loads from imputed data*

Description

Calculate cumulative per-day gas loads after imputing gas data. This function calculates cumulative load using area under the curve ("AUC"). Each gas data point is assumed to be the hourly load for all days surrounding the measurement, up to one half of the distance between the measurement and its nearest preceding and following measurement.

Additional methods, like linear interpolation, may be added in the future as options in the `method` argument.

Usage

```
calculate_gas_loads(
  data,
 gas_vars = c(co2_kg_ha_day_imputed, n2o_kg_ha_day_imputed, ch4_kg_ha_day_imputed,
    nh3_kg_ha_day_imputed),
  date_var = Date,
  site_var = site,
  plot_var = plot,
  method = "AUC",
  per_hour = FALSE
)
```

Arguments

`data`	A data frame
`gas_vars`	Names of columns in `data` containing imputed (i.e. containing no NA values) gas data in mg kg^{-1}. Defaults to `c(co2_kg_ha_day_imputed, n2o_kg_ha_day_imputed,ch4_kg_h` `nh3_kg_ha_day_imputed)`.
`date_var`	Name of column in `data` containing sampling dates in YYYY-MM-DD format. Defaults to `Date`.
`site_var`	Name of column in `data` containing site names. Defaults to `site`.
`plot_var`	Name of column in `data` containing plot IDs. Defaults to `plot`.
`method`	Defaults to `'AUC'` (area under the curve), currently the only option. Linear interpolation methods to be added.
`per_hour`	Logical. Are the units of `gas_vars` per hour (`per_hour = TRUE`) or per day (`per_hour = FALSE`). Defaults to `FALSE`.

Value

`data` with new columns for gas loads

calib_optim *Optimization of Carbon Dioxide ppm per Second Models*

Description

Finds optimum values of `excl_var` column in `data` to make the R^2 value of the `lm()` performed in `ppm_seconds_lm_optim()` at least 0.98 while minimizing the amount of data points excluded from the model.

Usage

```
calib_optim(x, data, excl_var, min_n = 4)
```

Arguments

`x`	Vector of `0`s and `1`s which will be randomly generated in `calib_range()`
`data`	Data frame containing CO_2 data
`excl_var`	Logical column in `data` which tracks observations that should be excluded from the model

Details

See `model_co2()` for details about the parameters and method of model optimization.

Value

Number of data points excluded from the model

See Also

`model_co2()`, `calib_range()`

calib_range *Optimization of Carbon Dioxide ppm per Second Models*

Description

Generates random instances of the logical column specified in `excl_var`, contained in `data`. In each random instance, existing `TRUE` observations are retained, and a random number of points at the head and tail of the column are replaced with `TRUE`. Maximum number of `TRUE` observations in the final instance is set at `min_n`. Converts randomized vector to integer for compatibility with `optim()`.

Usage

```
calib_range(data, excl_var, min_n = 4, x)
```

Arguments

data	Data frame containing CO_2 data
excl_var	Logical column in data which tracks observations that should be excluded from the model
min_n	Integer. Minimum number of data points to retain in final model. Defaults to 4.
x	Passed from calib_optim; not used in this function. Removing this parameter causes an error in optim, but it does not need to be explicitly specified when the function is called.

Value

Vector of 0s and 1s with the same length as data.

See Also

calib_optim, model_co2

---

chron_linear_interp *Linear interpolation of missing values between dates*

Description

Linear interpolation of missing values between dates

Usage

```
chron_linear_interp(
  data,
  gas_vars = c(co2_mg_kg_day, n2o_mg_kg_day, ch4_mg_kg_day, nh3_mg_kg_day),
  date_var = Date,
  site_var = site,
  year_var = year,
  plot_var = plot
)
```

Arguments

data	A data frame
gas_vars	Names of columns in data containing gas data to be interpolated. Defaults to c(co2_mg_kg_day, n2o_mg_kg_day,ch4_mg_kg_day, nh3_mg_kg_day).
date_var	Name of column in data containing sampling dates of format "YYYY-MM-DD". Defaults to Date.
site_var	Name of column in data containing site names. Defaults to site.
year_var	Name of column in data containing year of sampling. Defaults to year.
plot_var	Name of column in data containing plot ID. Defaults to plot.

Value

data with new columns containing interpolated data, named with suffix "_interp".

exclude_one_rep *Exclude observations in groups of length one*

Description

Exclude observations in groups of length one

Usage

```
exclude_one_rep(
  data,
  excl_vars = c(co2_exclude, n2o_exclude, ch4_exclude, nh3_exclude),
  attr_var = attributes,
  prefixes = c("co2", "n2o", "ch4", "nh3"),
  by_vars = c(site, Date, treatment)
)
```

Arguments

data	A data frame
excl_vars	Names of logical columns in data which should be changed to TRUE when the total instances of FALSE in the group (grouped by by_vars) is equal to one. Defaults to c(co2_exclude, n2o_exclude, ch4_exclude, nh3_exclude).
attr_var	Name of column in data containing data attributes. Defaults to attributes.
by_vars	Variables to group_by() when calculating totals where excl_vars == FALSE.

Value

data with updated excl_vars

exclude_outliers *Mark outliers for exclusion*

Description

Determine outliers using is_outlier() and mark them for exclusion from downstream processing.

Usage

```
exclude_outliers(
  data,
  gas_var,
  excl_var,
  attr_var = attributes,
  prefix,
  sep = ",",
  n_IQR = 3,
  datatable = FALSE
)
```

Arguments

data Data frame

gas_var Name of column in data containing gas flux data to be screened for outliers.

excl_var Name of logical column in data which is equal to TRUE when an observation of gas_var should be excluded from downstream processing and analysis.

attr_var Passed to attr_update(). Name of column in data which records attributes. Defaults to attributes.

prefix Unique prefix denoting attribute element in attr_var to be updated with attr_update(). If not specified, it will be inferred from gas_var.

sep Passed to attr_update(). String that separates attribute elements in attr_var. Defaults to ','.

n_IQR Integer. Passed to is_outlier. Number of interquartile ranges below 25 value will be considered an outlier. Defaults to 3.

datatable Logical. Should a datatable (using datatable() of extracted negative flux values be printed? Defaults to FALSE.

Value

data with updated excl_var and attr_var.

f *Make formula with sprintf-style replacements*

Description

Shortcut for, e.g. as.formula(sprintf(%s ~ x, "y")). To be used inside a modeling function call.

Usage

```
f(formula, repl, verbose = FALSE)
```

Arguments

formula String to be converted to formula, with instances of %s replaced by elements of repl

repl character vector of replacements for each instance of %s in formula

verbose Logical. Should replaced formula be printed to the console? Defaults to FALSE.

Value

A formula with specified replacements

Examples

```
x <- rnorm(20)
y1 <- rnorm(20)
y2 <- rnorm(20)

# Single replacement
formula <- '%s ~ x'
lm(f(formula, 'y1'))
lm(f(formula, 'y2'))

# Mutliple replacements
formula2 <- 'x ~ %s * %s'
lm(f(formula, c('y1', 'y2')))
```

gasmet_bind_check *Check that Gasmet data list is ready for binding*

Description

Performs several checks that evaluate if dfls is ready for binding with bind_rows. Gives errors with specifics if binding will be unsuccessful. Otherwise, the output of the function is not useful.

Usage

```
gasmet_bind_check(dfls, issues_path = "gasmet_data_import_issues.txt")
```

Arguments

dfls	List of data frames to check before binding
issues_path	File path to .txt file generated with gasmet_file_issues(). Defaults to 'gasmet_data_import_issues.txt'

Value

TRUE if data frames in dfls are ready for binding

gasmet_file_issues *Detect Issues that Will Affect Gasmet TXT File Import*

Description

Detects issues in TXT files from the Gasmet such as missing columns, mismatches between dates in data columns and file/folder names, multiple dates contained in the same file, etc. Uses cat() to write a message with details about detected issues (and if they need to be resolved for processing to continue). Does not fix the issues - that must be done manually by the user.

Usage

```
gasmet_file_issues(
  dfls,
  check_co2 = TRUE,
  check_n2o = TRUE,
  check_ch4 = TRUE,
  check_nh3 = TRUE,
  check_co = FALSE,
  ...
)
```

Arguments

`dfls`	List of data frames, each list item containing data from a single site/date
`check_co2`	Logical. Whether or not to check issues in CO_2 related columns. Defaults to `TRUE`.
`check_n2o`	Logical. Whether or not to check issues in N_2O related columns. Defaults to `TRUE`.
`check_ch4`	Logical. Whether or not to check issues in CH_4 related columns. Defaults to `TRUE`.
`check_nh3`	Logical. Whether or not to check issues in NH_3 related columns. Defaults to `TRUE`.
`check_co`	Logical. Whether or not to check issues in CO related columns. Defaults to `FALSE`.
`...`	Passed to `cat()` in final message. Use `file = 'file/path.txt'` to write final message to a text file.

Value

Message with details about fatal and non-fatal issues. Writes to the console by default. When `file` is specified, writes to that file.

---

get_latest_instructions

Get Latest Version of PowerPoint Package Instructions

Description

Get Latest Version of PowerPoint Package Instructions

Usage

```
get_latest_instructions(directory = "instructions")
```

Arguments

`directory`	String. Directory in which to copy PowerPoint file. Defaults to `"instructions"`.

Value

Nothing in memory. Copies latest instructions to `directory` if the latest version is not already contained there. If the latest version is already present, a message is printed to the console.

`ghg_sum_stats` *Summary Statistics for GHG Data*

Description

Shortcut function to get calculate mean, median, min, max, standard deviation, standard error, and count for `sum_vars`. Best used in combination with `group_by()`. See examples.

Usage

```
ghg_sum_stats(.data, ..., showDT = TRUE)
```

Arguments

`.data`	A data frame containing GHG data
`...`	Variable(s) in `.data` to be summarized
`showDT`	Logical. Do you want to print an HTML data table of results made with `datatable()`? Defaults to `TRUE`

Details

Columns in results will have the format ".col_.fn" where ".col" is the original column name in `data` and ".fn" is the function that has been performed in the summary. See examples.

Value

A data frame of summary statistics

Examples

```
data <- data.frame(
  co2 = rnorm(100, mean = 1000),
  n2o = rnorm(100, mean = 1000),
  treatment = rep(1:4, 25),
  site = rep(1:2, each = 50)
)
data %>% group_by(treatment, site) %>% ghg_sum_stats(co2, n2o, showDT = FALSE)
```

guess_prefix *Infer prefix from a gas column name*

Description

Infer prefix from a gas column name

Usage

```
guess_prefix(based_on)
```

Arguments

`based_on`	Name of gas column

Value

Prefix compatible with attributes

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

import_chamber_volume *Import Chamber Volume Data*

Description

Imports and cleans up chamber volume Excel spreadsheets contained in `path`.

Usage

```
import_chamber_volume(path = "data/00_raw/chamber_volume")
```

Arguments

`path`	File path in which to look for chamber volume files. Defaults to `"data/00_raw/chamber_volume"`.

Value

A data frame of all chamber volume data.

impute_missing_values *Impute missing gas data*

Description

Replaces gas_vars with NA according to excl_vars. Then fills NAs with imputed values using the procedure described in **Details**.

Usage

```
impute_missing_values(
  data,
  gas_vars = c(co2_kg_ha_day, n2o_kg_ha_day, ch4_kg_ha_day, nh3_kg_ha_day),
  excl_vars = NULL,
  site_var = site,
  date_var = Date,
  plot_var = plot,
  keep_all_cols = FALSE
)
```

Arguments

data	A data frame
gas_vars	Names of columns in data containing gas data to be imputed. Defaults to c(co2_kg_ha_day, n2o_kg_ha_day, ch4_kg_ha_day,nh3_kg_ha_day).
excl_vars	Names of logical columns in data which are TRUE when the corresponding gas_vars should be replaced with NA and then imputed. The order of names should match those in gas_vars (i.e. the first column specified in gas_vars is imputed when the first column specified in excl_vars is TRUE. Defaults to c(co2_exclude, n2o_exclude, ch4_exclude, nh3_exclude).
site_var	Name of column in data containing site names. Defaults to site.
date_var	Name of column in data containing sampling dates. Defaults to Date.
plot_var	Name of column in data containing plot IDs. Defaults to plot.
keep_all_cols	Logical. Should intermediate columns used in imputation calculations be retained in the output?

Details

Imputation is performed using the following procedure:

1. Data are arranged by site_var, plot_var, and date_var so that each unique plot's samplings appear in chronological order in data.
2. Columns are added that fill NAs with the nearest non-NA observation before AND after each missing value.
3. The distance between the missing values and each of the values it was filled with is calculated (i.e. how many samplings between the missing value and the nearest non-NA values before and after, chronologically?)

4. If the minimum distance between the missing value and a non-NA value is 2 or less, the missing value is replaced with the mean of the previous non-NA value and the next non-NA value chronologically.
5. If the minimum distance is greater than 2, the missing value is replaced with the grand mean of all non-NA observations for the plot.

Value

data with new columns for imputed values. When keep_all_cols is TRUE, intermediate columns generated for imputation calculations (e.g. grand means by plot) are retained in the output.

include_n *Count Observations Not Yet Excluded from Processing*

Description

Count Observations Not Yet Excluded from Processing

Usage

```
include_n(data, excl_var)
```

Arguments

data A data frame containing a logical column tracking processing exclusions by observation.

excl_var Name of logical column which tracks processing exclusions. Defaults to exclude_obs.

Value

Integer count of rows in data where exclude is FALSE.

Examples

```
dat <- data.frame(
x = rnorm(10),
exclude_obs = sample(c(TRUE, FALSE), 10, replace = TRUE)
)
include_n(dat)
```

is_outlier *Detect outliers*

Description

Intended for use inside mutate() or filter(). Detects outliers in column or grouped column x as values less than the 25 ranges or greater than the 75 Returns TRUE for outliers and FALSE for non outliers.

Usage

```
is_outlier(x, n_IQR = 3)
```

Arguments

n_IQR	Integer. Number of interquartile ranges below 25 above 75 to 3.
gas_var	Gas flux column in which to detect outliers

Value

A logical vector the same length as x (or current group)

make_preprocessing_report *Perform Preprocessing Steps and Make HTML Reports*

Description

Perform Preprocessing Steps and Make HTML Reports

Usage

```
make_preprocessing_report(
  data,
  date_var = Date,
  site_var = site,
  rmd_file = "code/110_preprocessing.Rmd",
  output_data_dir,
  output_report_dir,
  output_file,
  parent = rstudioapi::getActiveDocumentContext()$path,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`data` A data frame containing a single unique date and site of sampling

`date_var` Name of column in `data` containing the date of sampling. Defaults to `Date`.

`site_var` Name of column in `data` containing the site of sampling. Defaults to `site`.

`rmd_file` RMarkdown file from which to generate reports using `render()`. Defaults to `"code/110_preprocessing.Rmd"`.

`output_data_dir` Optional. Directory in which to store data outputs. If not specified, data outputs will be stored in `"data/01_converted_to_flux_rate/site_var_1"`, where `site_var_1` is replaced with the name of the single unique site in data$site_var. The directory will be created if it does not already exist.

`output_report_dir` Optional. Directory in which to store HTML report outputs. If not specified, reports will be stored in `"data/01_converted_to_flux_rate/site_var_1/html_reports"`, where `site_var_1` is replaced with the name of the single unique site in data$site_var. The directory will be created if it does not already exist.

`output_file` Optional. Name of file outputs. If not specified, files will be named `"YYYYMMDD_preprocessing"`, where `"YYYYMMDD"` is replaced with the single unique date in `data$date_var`.

`overwrite` Logical. Should `render()` be performed if `output_file`.html already exists at `output_report_dir`? Defaults to `FALSE`.

`verbose` Logical. Should messages be printed to the console? Defaults to `TRUE`.

Value

Nothing in R memory. Exports reports and data files to specified directories.

`mutate_levene_shapiro_p` *Add levene and shapiro p values to model data*

Description

Add levene and shapiro p values to model data

Usage

```
mutate_levene_shapiro_p(
  data,
  formula,
  gas_vars = c(co2, n2o, ch4, nh3),
  resid_vars = NULL
)
```

Arguments

`data` A data frame

`gas_vars` Names of columns in `data` containing gas data to be substituted in `formula`. Each name should be an unquoted expression. Defaults to `c(co2, n2o, ch4, nh3)`.

`resid_vars` Unquoted expressions. Names of columns in `data` containing model residuals. Order should correspond with `gas_vars`. If NULL, `gas_vars` will be appended with "_resid" to deduce the residual variable names.

Value

`data` with `length(gas_vars) * 2` new columns

`mutate_residuals` *Add model residuals to data*

Description

Add model residuals to data frame used in modeling. Typically used after `build_gas_lmer`.

Usage

```
mutate_residuals(data, models, suffix = "_resid")
```

Arguments

`data` A data frame

`models` List of models generated with `build_gas_lmer`.

`suffix` String. By default, names of new columns containing residuals will be based on the names of the models in `models` (usually gas variable names), plus a suffix of `"_resid"`. A different suffix can be specified with this argument.

Value

`data` with `length(models)` new columns.

`negative_flux` *Replace or Exclude Negative Values*

Description

Replace or Exclude Negative Values

Usage

```
negative_flux(
  data,
  gas_var,
  excl_var,
  method = c("excl", "zero"),
  attr_var = attributes,
  prefix,
  sep = ",",
  datatable = FALSE
)
```

Arguments

`data`	A data frame
`gas_var`	Name of column in `data` containing gas flux data to be screened for negative values
`excl_var`	Name of logical column in `data` which is equal to `TRUE` when an observation of `gas_var` should be excluded from downstream processing and analysis.
`method`	Character. Method of handling negative values: exclusion or replacement with 0. Options are `"excl"` or `"zero"`.
`attr_var`	Passed to `attr_update()`. Name of column in `data` which records attributes. Defaults to `attributes`.
`prefix`	Passed to `attr_update()`. Character. Unique prefix denoting attribute element in `attr_var` to be updated. If not specified, it will be inferred from `gas_var`.
`sep`	Passed to `attr_update()`. String that separates attribute elements in `attr_var`. Defaults to `','`.
`datatable`	Logical. Should a datatable (using `datatable()` of extracted negative flux values be printed? Defaults to `FALSE`.

Value

`data` with, if `method = "excl"`, updated `excl_var` and `attr_var`, or, if `method = "zero"`, updated `gas_var`.

`noaa_temp_fig` *Daily Minimum and Maximum Air Temperatures Barplot*

Description

Daily Minimum and Maximum Air Temperatures Barplot

Usage

```
noaa_temp_fig(
  data,
  date_var = Date,
  tmax = NOAA_airtemp_c_max,
  tmin = NOAA_airtemp_c_min,
  site_var = NULL,
  sample_dates,
  bar_fill_colors = c("#D2B4DE", "#AED6F1", "#A569BD", "#3498DB"),
  date_breaks = "1 month",
  date_labels = "%b %Y",
  text_size = 14,
  x_lab = "Date",
  y_lab = "Daily Minimum and Maximum\nAir Temperature (°C)"
)
```

Arguments

data	A data frame containing one row per calendar day
date_var	Name of column in data containing dates in "YYYY-MM-DD" format. Defaults to Date.
tmax	Name of column in data containing daily maximum air temperature. Defaults to NOAA_airtemp_c_max.
tmin	Name of column in data containing daily minimum air temperature. Defaults to NOAA_airtemp_c_min.
site_var	Name of column in data containing site names. If provided, it will be used in the rows argument of facet_grid(). Defaults to NULL.
sample_dates	Vector of all unique dates of sampling. By default, bars on these dates will be darker than non-sampling-date bars.
bar_fill_colors	Vector of length 4 specifying colors for, respectively, daily max temp on a non-sampling date, daily min temp on a non-sampling date, daily max temp on a sampling date, and daily min temp on a sampling date. Defaults to c('#D2B4DE', '#AED6F1', '#A569BD','#3498DB').
date_breaks	String. Passed to scale_x_date(). Defaults to '1 month'.
text_size	Integer. Size of text on plots. Defaults to 14.
x_lab	String or expression. X-axis label. Defaults to "Date".
y_lab	String or expression. Y-axis label. Defaults to "Daily Minimum and Maximum\nAir Temperature (°C)".

Value

A ggplot object

---

ppm_seconds_lm *Model Gas ppm per Second*

Description

Create a model of gas ppm per second, with or without use of an optimization algorithm to remove endpoints attributable to Gasmet warm-up or cool-down.

When optimization = TRUE, the function uses calib_optim() and calib_range() to model CO_2 data with optimization as described in **Details**. Returns data with optimized excl_var column, updated attr_var, and additional columns: co2_rsq, co2_intercept, and co2_slope, which correspond to the R^2, intercept, and slope of the final model, respectively.

When optimization = FALSE (the default), the function creates a linear model of gas_var ~ seconds_var excluding observations where excl_var == TRUE. If the model R^2 is at least 0.1, the function adds the model coefficients and R^2 value in new columns in data. Otherwise, the same columns are added but with all entries NA. Attributes are updated in data as needed (see ?attr_show_codes for details). A logical column is added to track exclusions of the specific gas being modeled.

Typically, this function should be run first on CO_2 data with optimization = TRUE, then on other gases with optimization = FALSE.

Usage

```
ppm_seconds_lm(
  data,
  gas_var,
  excl_var = exclude_obs,
  min_n = 4,
  seconds_var = seconds,
  attr_var = attributes,
  prefix,
  optimization = FALSE
)

ppm_seconds_lm_optim(
  data,
  gas_var = Carbon.dioxide.CO2,
  excl_var = exclude_obs,
  min_n = 4,
  seconds_var = seconds,
  attr_var = attributes,
  prefix = "all",
  optimization = TRUE
)

ppm_seconds_lm_asis(
  data,
  gas_var,
  min_n = 4,
  excl_var = exclude_obs,
  seconds_var = seconds,
  attr_var = attributes,
  prefix,
  optimization = FALSE
)
```

Arguments

data Data frame containing CO_2 data

gas_var Name of column in data containing gas ppm data to be modeled. Defaults to Carbon.dioxide.CO2 when optimization = TRUE.

excl_var Logical column in data which is equal to TRUE when an observation of gas_var should be excluded from modeling and downstream analyses. Defaults to exclude_obs.

`min_n` Integer. Minimum number of data points for optimization algorithm to retain in final model. Passed to `calib_optim()` and `calib_range()`. Defaults to 4.

`seconds_var` Name of column in `data` containing seconds since start of gas measurement. Defaults to `seconds`.

`attr_var` Name of column in `data` where attributes are stored. Defaults to `attributes`.

`prefix` Passed to `attr_update()`. Defaults to `"all"` when `optimization = TRUE`. If missing when `optimization = FALSE`, it will attempt to be inferred from `gas_var`.

`optimization` Logical. Whether or not to perform optimization algorithm descibed in **Details**. Defaults to `FALSE`.

Details

When `optimization = TRUE`, the optimization proceeds as follows:

1. If `data` has less than `min_n` observations where `excl_var == FALSE`, modeling is not performed and '02' is added to `data$attr_var` for all gases. New columns are added with all values `NA`.
2. Otherwise, `optim()` is called:
   - Random instances of `data$excl_var` are generated using `calib_range()`:
     - Existing observations where `data$excl_var == TRUE` are retained
     - n observations at the head (n_{head}) and/or tail (n_{tail}) of data\$excl_var are replaced with `TRUE`, where `n = 0 - m`, and `m` is a dynamic integer which keeps the sum of n_{head} and n_{tail} less than `nrow(data) - min_n` (i.e. the number of observations included remains $\geq$ `min_n`)
   - Each random instance of `data$excl_var` is fed to `calib_optim()` which executes the following scheme:
     - A model of CO_2 per second, excluding observations marked for exclusion in the random instance, is generated
     - If the model R^2 value is ≥ 0.98, the result fed to optim is the number of observations excluded from the model (this is the value we want to minimize)
     - If the model R^2 value is < 0.98, an arbitrary, extremely large value is returned to discourage `optim()` from choosing that model
     - The previous three steps are repeated 1000 or `nrow(data) * 10` times, whichever is larger.
3. If there is never a model with $R^2 \geq 0.98$ in any iteration, the modeling is considered a failure case, all values in `data$excl_var` are updated to `TRUE`, '03' is added to `data$attributes`, and new columns are added with all values `NA`.
4. Otherwise, the model is considered a success case and the resulting data frame contains the values of `data$excl_var` used in the final model, and new columns with model coefficients.

Value

`data` with new columns containing model coefficients and updated `attr_var`.

See Also

`ppm_seconds_plot()`, `calib_optim()`, `calib_range()`

ppm_seconds_plot *Create ggplot for ppm Gas per Second Linear Models*

Description

Create side-by-side ggplots of pre- and post-optimization models of gas ppm per second, or a plot displaying a failure message when models do not meet quality control parameters.

Usage

```
ppm_seconds_plot(
  data,
  gas_var,
  gas_name = c("co2", "n2o", "ch4", "nh3"),
  excl_var = paste0(gas_name, "_exclude"),
  rsq_var = paste0(gas_name, "_rsq"),
  co2_excl_var = exclude_obs,
  co2_rsq_var = co2_rsq,
  plot_var = plot,
  seconds_var = seconds
)
```

Arguments

`data`	A data frame
`gas_var`	Name of column in `data` containing gas data to be plotted
`gas_name`	Character. Name of gas that should be used in messages on plots. Options are `'co2'`, `'n2o'`, `'ch4'`, and `'nh3'`. Should correspond to `gas_var`.
`excl_var`	Name of logical column in `data` which tracks observations of `gas_var` which were excluded from ppm per second modeling. Defaults to `paste0(gas_name, "_excl")`.
`rsq_var`	Name of column containing R-squared values for the model of `gas_var` ~ `seconds_col`. Defaults to `paste0(gas_name,"_rsq")`.
`co2_rsq_var`	Name of column containing R-squared value for CO2 ppm ~ seconds model. If `model_co2()` was used, this should be `co2_rsq`. Defaults to `co2_rsq`.
`plot_var`	Name of column containing plot IDs. Defaults to `plot`.
`seconds_var`	Name of column containing number seconds since the start of the gas measurement in the plot. Defaults to `seconds`.

Details

`data` should contain data for a single plot on a single sampling date.

Value

A plot

See Also

`model_co2()`

qqplots *Make Normal Quantile-Quantile Plots for Multiple Gases*

Description

Make Normal Quantile-Quantile Plots for Multiple Gases

Usage

```
qqplots(
  data,
  gas_vars = c(co2, n2o, ch4, nh3),
  resid_vars = NULL,
  one_plot = TRUE
)
```

Arguments

data	A data frame
gas_vars	Unquoted expressions. Names of columns in data containing gas data.
resid_vars	Unquoted expressions. Names of columns in data containing model residuals. Order should correspond with gas_vars. If NULL, gas_vars will be appended with "_resid" to deduce the residual variable names.
one_plot	Logical. Should plots be arranged on a single grid in the output? If FALSE, a list of plots is returned.

Value

If one_plot = TRUE, a single plot. If one_plot = FALSE, a list of ggplots.

regression_fig *Scatterplot Figure with Regression Line, Equation, and Stats*

Description

Scatterplot Figure with Regression Line, Equation, and Stats

Usage

```
regression_fig(
  data,
  x_var,
  y_var,
  group_var,
  facet_rows_var = gas,
  facet_cols_var = site,
  labeller = "label_value",
  group_colors,
  x_lab = "",
  y_lab = "",
  color_lab = "",
  text_size = 14,
  point_shape = 16,
  point_alpha = 0.5,
  legend_position = "top",
  eqn_x = 0,
  eqn_y = Inf,
  eqn_vjust = 1.5,
  rr_x = 0,
  rr_y = Inf,
  rr_vjust = 1
)
```

Arguments

`data` A data frame. Should contain one row per experimental unit.

`x_var` Name of column in `data` to be used as the x variable.

`y_var` Name of column in `data` to be used as the y variable.

`group_var` Optional. Name of column in `data` to be used as a grouping variable. One regression line will appear per group.

`facet_rows_var` Optional. Name of column in `data` to be used as `rows` argument in `facet_grid()`. Set to `NULL` to remove facet rows. Defaults to `gas`.

`facet_cols_var` Optional. Name of column in `data` to be used as `cols` argument in `facet_grid()`. Set to `NULL` to remove facet columns. Defaults to `site`.

`labeller` Passed to `facet_grid()`. A labeller object containing specifications to be used in `labeller()`. See `?labeller` for details. Defaults to `"label_value"`.

`group_colors` Optional. Color palette for `group_var`. If `group_var` is provided but `group_colors` is missing, `brewer.pal()`'s 'Set3' will be used as the color palette.

`x_lab` String. X axis label. Defaults to empty.

`y_lab` String. Y axis label. Defaults to empty.

`color_lab` String. Color legend title. Defaults to empty.

`text_size` Size of text on plot. Defaults to `14`.

`point_shape` Integer. Shape of points. Defaults to `16`.

`point_alpha` Numeric 0-1. Opacity of points. Defaults to `0.5`.

`legend_position`
String. Position of legend on plot. Options `'top'`, `'left'`, `'right'`, `'bottom'`, or `'none'`. Defaults to `'top'`.

`eqn_x` Numeric. X axis position of regression equation. Defaults to `0`.

`eqn_y` Numeric. Y axis position of regression equation. Defaults to `Inf`.

`eqn_vjust` Numeric. Vertical justification of regression equation. Defaults to `1.5`.

`rr_x` Numeric. X axis position of R^2 and p-value. Defaults to `0`.

`rr_y` Numeric. Y axis position of R^2 and p-value. Defaults to `Inf`.

`rr_vjust` Numeric. Vertical justification of R^2 and p-value. Defaults to `1`.

Value

A plot

replace_gas_with_na *Replace gas variable entries with NA based on their corresponding exclusion tracking columns*

Description

Replace gas variable entries with NA based on their corresponding exclusion tracking columns

Usage

```
replace_gas_with_na(
  data,
  gas_vars = c(co2_kg_ha_day, n2o_kg_ha_day, ch4_kg_ha_day, nh3_kg_ha_day),
  excl_vars = c(co2_exclude, n2o_exclude, ch4_exclude, nh3_exclude)
)
```

Arguments

`data`	A data frame
`gas_vars`	Names of columns in `data` containing gas data to be replaced with NA when the corresponding variable in `excl_vars` is `TRUE`. Defaults to `c(co2_kg_ha_day, n2o_kg_ha_day,ch4_kg_ha_day, nh3_kg_ha_day)`.
`excl_vars`	Names of logical columns in `data` which are `TRUE` when the corresponding `gas_vars` should be replaced with NA. The order of names should match those in `gas_vars` (i.e. the first column specified in `gas_vars` is replaced with NA when the first column specified in `excl_vars` is `TRUE`. Defaults to `c(co2_exclude, n2o_exclude, ch4_exclude, nh3_exclude)`.

Value

`data` with updated `gas_vars` columns

resplots *Make Model Residual Plots for Multiple Gases*

Description

Make Model Residual Plots for Multiple Gases

Usage

```
resplots(
  data,
  gas_vars = c(co2, n2o, ch4, nh3),
  resid_vars = NULL,
  one_plot = TRUE
)
```

Arguments

`data`	A data frame
`gas_vars`	Unquoted expressions. Names of columns in `data` containing gas data.
`resid_vars`	Unquoted expressions. Names of columns in `data` containing model residuals. Order should correspond with `gas_vars`. If NULL, `gas_vars` will be appended with "_resid" to deduce the residual variable names.
`one_plot`	Logical. Should plots be arranged on a single grid in the output? If `FALSE`, a list of plots is returned.

Value

A list of ggplots

`rm_random_fx` *Remove random effects from lmer-type model formula*

Description

Convenience function to remove random effects from `lmer` type model formulas, which is sometimes necessary for model checking functions like Levene's test.

Usage

```
rm_random_fx(formula)
```

Arguments

`formula`	String. Formula of the type `"y ~ x + (1|block)"`.

Value

String containing only fixed effects from formula

`round_p` *Round P Values for Plot Labels*

Description

Rounds `p` to three decimal places. If the rounded value is less than 0.001, the output is `"< 0.001"`. Otherwise, the output is `"= 0.000"` where `0.000` is the rounded `p`.

Usage

```
round_p(p)
```

Arguments

`p`	Numeric. P value to be rounded

Value

Character "< 0.001" or "= p" where p is rounded to 3 decimal places.

Examples

```
round_p(0.000025038)
round_p(0.424610370)
```

---

summary_boxplots_flux *Summary Boxplots of Daily Flux Data*

Description

Summary Boxplots of Daily Flux Data

Usage

```
summary_boxplots_flux(
  data,
  x_var = treatment,
  y_vars = c(co2_kg_ha_day, n2o_kg_ha_day, ch4_kg_ha_day, nh3_kg_ha_day),
  by = c(site, Date),
  x_lab = "Treatment",
 y_labs = c("Carbon Dioxide\n(kg CO2-C/ha/day)", "Nitrous Oxide\n(kg N2O-N/ha/day)",
    "Methane\n(kg CH4-C/ha/day)", "Ammonia\n(kg NH3-N/ha/day)"),
  box_color = "sienna2",
  box_fill = "transparent",
  box_alpha = 1,
  point_shape = 21,
  point_alpha = 0.5,
  point_color = "black",
  point_fill = "gray",
  plot_var = plot,
  facet_scales = "free_y"
)
```

Arguments

data	A data frame
x_var	Name of column in data that should be plotted as the x variable. Defaults to treatment.
y_vars	Name(s) of column(s) in data that should be plotted as the y variable(s). If mutliple names are given, plots will be faceted so that there is one facet per y variable. See facet_grid() for details.
by	Split the data by these variables and make a plot for each. Defaults to c(site, Date). Set to NULL to generate a single plot.
x_lab	String. Label that will appear on the x axis. Defaults to 'Treatment'.
y_labs	String. Label(s) that will appear on the y axis or facet labels. Order should correspond to y_vars.

`box_color` String. Outline color of boxes in boxplots. Defaults to `'sienna2'`.

`box_fill` String. Fill color of boxes in boxplots. Defaults to `"transparent"`.

`box_alpha` Numeric 0-1. Opacity of boxes in boxplots. Defaults to `1`.

`point_shape` Integer. Shape of points on boxplots. Defaults to `21`.

`point_alpha` Numeric 0-1. Opacity of points on boxplots. Defaults to `0.5`.

`point_color` String. Outline color of points on boxplots. Defaults to `'black'`.

`point_fill` String. Fill color of points on boxplots. Defaults to `"gray"`.

`plot_var` Name of column in `data` containing plot IDs. Defaults to `plot`.

`facet_scales` String. Passed to `scales` arg of `facet_grid()`. Defaults to `"free_y"`.

Value

A single boxplot if by is NULL, or else a list of boxplots.

`summary_boxplots_loads`

Summary Boxplots of Cumulative Gas Load Data

Description

Summary Boxplots of Cumulative Gas Load Data

Usage

```
summary_boxplots_loads(
  data,
  x_var = treatment,
  y_vars = c(co2_kg_ha_load, n2o_kg_ha_load, ch4_kg_ha_load, nh3_kg_ha_load),
  by = site,
  x_lab = "Treatment",
  y_labs = c("Carbon Dioxide Load\n(kg CO2-C/ha)", "Nitrous Oxide Load\n(kg N2O-N/ha)",
    "Methane Load\n(kg CH4-C/ha)", "Ammonia Load\n(kg NH3-N/ha)"),
  box_color = "sienna2",
  box_fill = "transparent",
  box_alpha = 1,
  point_shape = 21,
  point_alpha = 0.5,
  point_color = "black",
  point_fill = "gray",
  plot_var = plot,
  date_var = Date,
  facet_scales = "free_y"
)
```

Arguments

data A data frame

x_var Name of column in data that should be plotted as the x variable. Defaults to treatment.

y_vars Name(s) of column(s) in data that should be plotted as the y variable(s). If mutliple names are given, plots will be faceted so that there is one facet per y variable. See facet_grid() for details.

by Split the data by these variables and make a plot for each. Defaults to c(site, Date). Set to NULL to generate a single plot.

x_lab String. Label that will appear on the x axis. Defaults to 'Treatment'.

y_labs String. Label(s) that will appear on the y axis or facet labels. Order should correspond to y_vars.

box_color String. Outline color of boxes in boxplots. Defaults to 'sienna2'.

box_fill String. Fill color of boxes in boxplots. Defaults to "transparent".

box_alpha Numeric 0-1. Opacity of boxes in boxplots. Defaults to 1.

point_shape Integer. Shape of points on boxplots. Defaults to 21.

point_alpha Numeric 0-1. Opacity of points on boxplots. Defaults to 0.5.

point_color String. Outline color of points on boxplots. Defaults to 'black'.

point_fill String. Fill color of points on boxplots. Defaults to "gray".

plot_var Name of column in data containing plot IDs. Defaults to plot.

date_var Name of column in data containing sampling date. Defaults to Date.

facet_scales String. Passed to scales arg of facet_grid(). Defaults to "free_y".

Value

A single boxplot if by is NULL, or else a list of boxplots.

time_series *Time Series of Gas Data*

Description

Time Series of Gas Data

Usage

```
time_series(
  data,
  date_var = Date,
  mean_var = gas_value,
  se_var = se,
  lines_group_var = treatment,
  facet_rows_var = gas,
  facet_cols_var = site,
  labeller = "label_value",
  line_colors,
  x_lab = "Date",
  y_lab = "",
  color_lab = "Treatment",
  date_breaks = "1 month",
  date_labels = "%b %Y",
  text_size = 14
)
```

Arguments

data	A data frame. Should be summarized (in most cases, by site/Date/treatment).
date_var	Name of column in data containing sampling date in "YYYY-MM-DD" format. Defaults to Date.
mean_var	Name of column in data containing mean gas value to be used as the y variable. Defaults to gas_value.
se_var	Name of column in data containing standard error of the mean gas value to be used in error bars. Defaults to se.
lines_group_var	Name of column in data to use as grouping variable for lines. Defaults to treatment. Set to 1 to plot a single line.
facet_rows_var	Name of column in data to be used in faceting rows using facet_grid(). Defaults to gas. Set to NULL to remove row faceting.
facet_cols_var	Name of column in data to be used in faceting columns using facet_grid(). Defaults to site. Set to NULL to remove column faceting.
labeller	Passed to labeller(). A labeller object containing specifications to be used in facet_grid(). See ?facet_grid for details.
line_colors	Vector of colors to be used for lines. Length must match length of unique levels of lines_group_var in data. If not specified, the function will use 'Set3' from brewer.pal().
x_lab	Label for the x axis. Defaults to 'Date'.
y_lab	Label for the y axis. Defaults to empty.
color_lab	Label for color legend. Defaults to "Treatment".
date_breaks	Passed to scale_color_manual(). Defaults to '1 month'.
text_size	Size of text on plot. Defaults to 14.

Value

A plot