Package ‘wqtrends’

January 12, 2026

Title Assess Water Quality Trends with Generalized Additive Models

Version 1.5.2

Date 2026-01-05

Description Assess Water Quality Trends for Long-Term Monitoring Data in Estuaries using Generalized Additive Models following Wood (2017) <doi:10.1201/9781315370279> and Error Propagation with Mixed-Effects Meta-
Analysis following Sera et al. (2019) <doi:10.1002/sim.8362>.
Methods are available for model fitting, assessment of fit, annual and seasonal trend tests, and visualization of results.

Depends R (>= 3.5)

Imports dplyr, ggplot2, lubridate, mgcv, mixmeta, plotly, purrr, tibble, tidyr, viridisLite

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

URL <`https://github.com/tbep-tech/wqtrends/`>, <`https://tbep-tech.github.io/wqtrends/`>

BugReports `https://github.com/tbep-tech/wqtrends/issues`

Suggests testthat (>= 2.1.0), covr, english, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Marcus Beck [aut, cre] (ORCID: <`https://orcid.org/0000-0002-4996-0059`>),
Perry de Valpine [aut],
Rebecca Murphy [aut],
Ian Wren [aut],
Ariella Chelsky [aut],
Melissa Foley [aut] (ORCID: <`https://orcid.org/0000-0002-5832-6404`>),
David Senn [aut] (ORCID: <`https://orcid.org/0000-0002-4869-3550`>)

Maintainer Marcus Beck <`mbeck@tbep.org`>

Repository CRAN

Date/Publication 2026-01-12 13:50:02 UTC

Contents

`anlz_avgseason` *Extract period (seasonal) averages from fitted GAM*

Description

Extract period (seasonal) averages from fitted GAM

Usage

```
anlz_avgseason(mod, doystr = 1, doyend = 364, yromit = NULL)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`yromit`	optional numeric vector for years to omit from the output

Value

A data frame of period averages

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_avgseason(mod, doystr = 90, doyend = 180)
```

`anlz_backtrans` *Back-transform response variable*

Description

Back-transform response variable after fitting GAM

Usage

```
anlz_backtrans(dat)
```

Arguments

`dat`	input data with `trans` argument

Details

`dat` can be output from `anlz_trans` or `anlz_prd`

Value

`dat` with the `value` column back-transformed using info from the `trans` column

Examples

```
library(dplyr)

tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')
dat <- anlz_trans(tomod, trans = 'log10')
backtrans <- anlz_backtrans(dat)
head(backtrans)
```

```
mod <- anlz_gam(tomod, trans = 'log10')
dat <- anlz_prd(mod)
backtrans <- anlz_backtrans(dat)
head(backtrans)
```

anlz_fit *Return summary statistics for GAM fits*

Description

Return summary statistics for GAM fits

Usage

```
anlz_fit(mod)
```

Arguments

mod	input model object as returned by anlz_gam

Details

Results show the overall summary of the model as Akaike Information Criterion (AIC), the generalized cross-validation score (GCV), and the R2 values. Lower values for AIC and GCV and higher values for R2 indicate improved model fit.

Value

A data.frame with summary statistics for GAM fits

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <-  anlz_gam(tomod, trans = 'log10')
anlz_fit(mod)
```

anlz_gam *Fit a generalized additive model to a water quality time series*

Description

Fit a generalized additive model to a water quality time series

Usage

```
anlz_gam(moddat, kts = NULL, ...)
```

Arguments

`moddat`	input raw data, one station and paramater
`kts`	optional numeric vector for the upper limit for the number of knots in the term `s(cont_year)`, see details
`...`	additional arguments passed to other methods, i.e., `trans = 'log10'` (default) or `trans = 'ident'` passed to `anlz_trans`

Details

The model structure is as follows:

model S: chl ~ s(cont_year, k = large)

The `cont_year` vector is measured as a continuous numeric variable for the annual effect (e.g., January 1st, 2000 is 2000.0, July 1st, 2000 is 2000.5, etc.) and doy is the day of year as a numeric value from 1 to 366. The function `s` models `cont_year` as a smoothed, non-linear variable. The optimal amount of smoothing on `cont_year` is determined by cross-validation as implemented in the mgcv package and an upper theoretical upper limit on the number of knots for `k` should be large enough to allow sufficient flexibility in the smoothing term. The upper limit of `k` was chosen as 12 times the number of years for the input data. If insufficient data are available to fit a model with the specified `k`, the number of knots is decreased until the data can be modelled, e.g., 11 times the number of years, 10 times the number of years, etc.

Value

a `gam` model object

Examples

```
library(dplyr)
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')
anlz_gam(tomod, trans = 'log10')
```

anlz_metseason *Extract period (seasonal) metrics from fitted GAM*

Description

Extract period (seasonal) metrics from fitted GAM

Usage

```
anlz_metseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  nsim = 10000,
  yromit = NULL,
  ...
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`metfun`	function input for metric to calculate, e.g., `mean`, `var`, `max`, etc
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`nsim`	numeric indicating number of random draws for simulating uncertainty
`yromit`	optional numeric vector for years to omit from the output
`...`	additional arguments passed to `metfun`, e.g., `na.rm = TRUE`

Details

This function estimates a metric of interest for a given seasonal period each year using results from a fitted GAM (i.e., from `anlz_gam`). The estimates are based on the predicted values for each seasonal period, with uncertainty of the metric based on repeated sampling of the predictions following uncertainty in the model coefficients.

Value

A data frame of period metrics

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_metseason(mod, mean, doystr = 90, doyend = 180, nsim = 100)
```

---

anlz_mixmeta *Fit a mixed meta-analysis regression model of trends*

Description

Fit a mixed meta-analysis regression model of trends

Usage

```
anlz_mixmeta(metseason, yrstr = 2000, yrend = 2019)
```

Arguments

metseason	output from anlz_metseason
yrstr	numeric for starting year
yrend	numeric for ending year

Details

Parameters are not back-transformed if the original GAM used a transformation of the response variable

Value

A list of mixmeta fitted model objects

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)
```

```
mod <- anlz_gam(tomod, trans = 'log10')
metseason <- anlz_metseason(mod, doystr = 90, doyend = 180)
anlz_mixmeta(metseason, yrstr = 2016, yrend = 2019)
```

`anlz_perchg` *Estimate percent change trends from GAM results for selected time periods*

Description

Estimate percent change trends from GAM results for selected time periods

Usage

```
anlz_perchg(mod, baseyr, testyr)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`baseyr`	numeric vector of starting years
`testyr`	numeric vector of ending years

Details

Working components of this function were taken from the gamDiff function in the baytrends package.

Value

A data frame of summary results for change between the years.

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <- anlz_gam(tomod, trans = 'log10')
anlz_perchg(mod, baseyr = 1990, testyr = 2016)
```

anlz_prd *Get predicted data from fitted GAMs across period of observation*

Description

Get predicted data from fitted GAMs across period of observation

Usage

```
anlz_prd(mod, annual = FALSE)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`annual`	logical indicating if predictions only for the `cont_year` smoother are returned

Value

a `data.frame` with predictions

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_prd(mod)
```

anlz_prdday *Get predicted data from fitted GAMs across period of observation, every day*

Description

Get predicted data from fitted GAMs across period of observation, every day

Usage

```
anlz_prdday(mod)
```

Arguments

`mod`	input model object as returned by `anlz_gam`

Value

a `data.frame` with predictions

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_prdday(mod)
```

`anlz_prdmatrix` *Get prediction matrix for a fitted GAM*

Description

Get prediction matrix for a fitted GAM

Usage

```
anlz_prdmatrix(mod, doystr = 1, doyend = 364, avemat = FALSE)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`avemat`	logical indicating if the prediction matrix is to be passed to `anlz_metseason` (default) or `anlz_avgseason`

Details

Used internally by `anlz_metseason`, not to be used by itself

Value

a `data.frame` with predictors to use with the fitted GAM

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_prdmatrix(mod, doystr = 90, doyend = 180)
```

`anlz_pvalformat` *Format p-values for show functions*

Description

Format p-values for show functions

Usage

```
anlz_pvalformat(x)
```

Arguments

`x`	numeric input p-value

Value

p-value formatted as a text string, one of `p < 0.001`, `'p < 0.01'`, `p < 0.05`, or `ns` for not significant

Examples

```
anlz_pvalformat(0.05)
```

`anlz_smooth` *Return summary statistics for smoothers of GAMs*

Description

Return summary statistics for smoothers of GAMs

Usage

```
anlz_smooth(mod)
```

Arguments

mod	input model object as returned by anlz_gam

Details

Results show the individual effects of the modelled components of each model as the estimated degrees of freedom (edf), the reference degrees of freedom (Ref.df), the test statistic (F), and significance of the component (p-value). The significance of the component is in part based on the difference between edf and Ref.df.

Value

a data.frame with summary statistics for smoothers in each GAM

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <- anlz_gam(tomod, trans = 'log10')
anlz_smooth(mod)
```

---

anlz_sumstats *Retrieve summary statistics for seasonal metrics and trend results*

Description

Retrieve summary statistics for seasonal metrics and trend results

Usage

```
anlz_sumstats(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  yrstr = 2000,
  yrend = 2019,
  yromit = NULL,
  nsim = 10000,
  confint = 0.95,
  useave = FALSE,
  ...
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`metfun`	function input for metric to calculate, e.g., `mean`, `var`, `max`, etc
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`yrstr`	numeric for starting year for trend model, see details
`yrend`	numeric for ending year for trend model, see details
`yromit`	optional numeric vector for years to omit from the plot, see details
`nsim`	numeric indicating number of random draws for simulating uncertainty
`confint`	numeric from zero to one indicating confidence interval level for summarizing the mixed-effects meta-analysis model, see details
`useave`	logical indicating if `anlz_avgseason` is used for the seasonal metric calculation, see details
`...`	additional arguments passed to `metfun`, e.g., `na.rm = TRUE`

Details

This function is primarily for convenience to return summary statistics of a fitted GAM from `anlz_gam`.

Note that `confint` only applies to the `summary` and `coeffs` list outputs. It does not apply to the `metseason` list element output that is default set to 95

Set `useave = T` to speed up calculations if `metfun = mean`. This will use `anlz_avgseason` to estimate the seasonal summary metrics using a non-stochastic equation.

Value

A list object with named elements:

- `mixmet`: `mixmeta` object of the fitted mixed-effects meta-analysis trend model
- `metseason`: tibble object of the fitted seasonal metrics as returned by `anlz_metseason` or `anlz_avgseason`
- `summary`: summary of the `mixmet` object
- `coeffs`: tibble object of the slope estimate coefficients from the `mixmet` model. An approximately linear slope estimate will be included as `slope.approx` if `trans = 'log10'` for the GAM used in `mod`.

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)
```

```
mod <- anlz_gam(tomod, trans = 'log10')

anlz_sumstats(mod, metfun = mean, doystr = 90, doyend = 180, yrstr = 2016,
  yrend = 2019, nsim = 100)
```

anlz_sumtrndseason *Estimate seasonal rates of change based on average estimates for multiple window widths*

Description

Estimate seasonal rates of change based on average estimates for multiple window widths

Usage

```
anlz_sumtrndseason(
  mod,
  doystr = 1,
  doyend = 364,
  justify = c("center", "left", "right"),
  win = 5:15,
  yromit = NULL
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`justify`	chr string indicating the justification for the trend window
`win`	numeric vector indicating number of years to use for the trend window
`yromit`	optional numeric vector for years to omit from the plot, see details

Details

The optional `yromit` vector can be used to omit years from the plot and trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

This function is a wrapper to `anlz_trndseason` to loop across values in `win`, using `useave = TRUE` for quicker calculation of average seasonal metrics. It does not work with any other seasonal metric calculations.

Value

A data frame of slope estimates and p-values for each year

See Also

Other analyze: `anlz_trans()`, `anlz_trndseason()`

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_sumtrndseason(mod, doystr = 90, doyend = 180, justify = 'center', win = 2:3)
```

`anlz_trans` *Transform response variable*

Description

Transform response variable prior to fitting GAM

Usage

```
anlz_trans(moddat, trans = c("log10", "ident"))
```

Arguments

`moddat`	input raw data, one station and paramater
`trans`	chr string indicating desired type of transformation, one of `log10` or `ident` (no transformation)

Value

`moddat` with the `value` column transformed as indicated

See Also

Other analyze: `anlz_sumtrndseason()`, `anlz_trndseason()`

Examples

```
library(dplyr)
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')
anlz_trans(tomod, trans = 'log10')
```

anlz_trndseason *Estimate rates of change based on seasonal metrics*

Description

Estimate rates of change based on seasonal metrics

Usage

```
anlz_trndseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  justify = c("center", "left", "right"),
  win = 5,
  nsim = 10000,
  yromit = NULL,
  useave = FALSE,
  ...
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`metfun`	function input for metric to calculate, e.g., `mean`, `var`, `max`, etc
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`justify`	chr string indicating the justification for the trend window
`win`	numeric indicating number of years to use for the trend window, see details
`nsim`	numeric indicating number of random draws for simulating uncertainty
`yromit`	optional numeric vector for years to omit from the output
`useave`	logical indicating if `anlz_avgseason` is used for the seasonal metric calculation, see details
`...`	additional arguments passed to `metfun`, e.g., `na.rm = TRUE`

Details

Trends are based on the slope of the fitted linear trend within the window, where the linear trend is estimated using a meta-analysis regression model (from `anlz_mixmeta`) for the seasonal metrics (from `anlz_metseason`). Set `useave = T` to speed up calculations if `metfun = mean`. This will use `anlz_avgseason` to estimate the seasonal summary metrics using a non-stochastic equation.

Note that for left and right windows, the exact number of years in `win` is used. For example, a left-centered window for 1990 of ten years will include exactly ten years from 1990, 1991, ... ,

1999. The same applies to a right-centered window, e.g., 1990 would include 1981, 1982, ..., 1990 (if those years have data). However, for a centered window, picking an even number of years for the window width will create a slightly off-centered window because it is impossible to center on an even number of years. For example, if `win = 8` and `justify = 'center'`, the estimate for 2000 will be centered on 1997 to 2004 (three years left, four years right, eight years total). Centering for window widths with an odd number of years will always create a symmetrical window, i.e., if `win = 7` and `justify = 'center'`, the estimate for 2000 will be centered on 1997 and 2003 (three years left, three years right, seven years total).

The optional `yromit` vector can be used to omit years from the trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

Value

A data frame of slope estimates and p-values for each year

See Also

Other analyze: `anlz_sumtrndseason()`, `anlz_trans()`

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
anlz_trndseason(mod, doystr = 90, doyend = 180, justify = 'center', win = 4)
```

`rawdat` *Raw data from San Francisco Estuary (South Bay)*

Description

Raw data from San Francisco Estuary (South Bay)

Usage

```
rawdat
```

Format

A `data.frame` object with 12411 rows and 8 columns

date Date

station int

param chr

value num

doy num

cont_year num

yr num

mo Ord.factor

Details

Data from `datprc` object in https://github.com/fawda123/SFbaytrends

---

`show_metseason` *Plot period (seasonal) averages from fitted GAM*

Description

Plot period (seasonal) averages from fitted GAM

Usage

```
show_metseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  yrstr = 2000,
  yrend = 2019,
  yromit = NULL,
  ylab,
  width = 0.9,
  size = 1.5,
  seascol = "deepskyblue3",
  trndcol = "pink",
  nsim = 10000,
  useave = FALSE,
  base_size = 11,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

mod	input model object as returned by anlz_gam
metfun	function input for metric to calculate, e.g., mean, var, max, etc
doystr	numeric indicating start Julian day for extracting averages
doyend	numeric indicating ending Julian day for extracting averages
yrstr	numeric for starting year for trend model, see details
yrend	numeric for ending year for trend model, see details
yromit	optional numeric vector for years to omit from the plot, see details
ylab	chr string for y-axis label
width	numeric for width of error bars
size	numeric for point size
seascol	chr string for color of the seasonal averages
trndcol	chr sting for color of the trend line
nsim	numeric indicating number of random draws for simulating uncertainty
useave	logical indicating if anlz_avgseason is used for the seasonal metric calculation, see details
base_size	numeric indicating base font size, passed to theme_bw
xlim	optional numeric vector of length two for x-axis limits
ylim	optional numeric vector of length two for y-axis limits
...	additional arguments passed to metfun, e.g., na.rm = TRUE

Details

Setting yrstr or yrend to NULL will suppress plotting of the trend line for the meta-analysis regression model.

The optional yromit vector can be used to omit years from the plot and trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

Set useave = T to speed up calculations if metfun = mean. This will use anlz_avgseason to estimate the seasonal summary metrics using a non-stochastic equation.

Value

A ggplot object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'ident')

show_metseason(mod, doystr = 90, doyend = 180, yrstr = 2016, yrend = 2019,
     ylab = 'Chlorophyll-a (ug/L)')


# show seasonal metrics without annual trend
show_metseason(mod, doystr = 90, doyend = 180, yrstr = NULL, yrend = NULL,
     ylab = 'Chlorophyll-a (ug/L)')

# omit years from the analysis
show_metseason(mod, doystr = 90, doyend = 180, yrstr = 2016, yrend = 2019,
     yromit = 2017, ylab = 'Chlorophyll-a (ug/L)')
```

`show_mettrndseason` *Plot seasonal metrics and rates of change*

Description

Plot seasonal metrics and rates of change

Usage

```
show_mettrndseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  justify = c("center", "left", "right"),
  win = 5,
  nsim = 10000,
  useave = FALSE,
  yromit = NULL,
  ylab,
  width = 0.9,
  size = 3,
  nms = NULL,
  fils = NULL,
  cmbn = F,
  base_size = 11,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`metfun`	function input for metric to calculate, e.g., `mean`, `var`, `max`, etc
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`justify`	chr string indicating the justification for the trend window
`win`	numeric indicating number of years to use for the trend window, see details
`nsim`	numeric indicating number of random draws for simulating uncertainty
`useave`	logical indicating if `anlz_avgseason` is used for the seasonal metric calculation, see details
`yromit`	optional numeric vector for years to omit from the plot, see details
`ylab`	chr string for y-axis label
`width`	numeric for width of error bars
`size`	numeric for point size
`nms`	optional character vector for trend names, see details
`fils`	optional character vector for the fill of interior point colors, see details
`cmbn`	logical indicating if the no trend and no estimate colors should be combined, see details
`base_size`	numeric indicating base font size, passed to `theme_bw`
`xlim`	optional numeric vector of length two for x-axis limits
`ylim`	optional numeric vector of length two for y-axis limits
`...`	additional arguments passed to `metfun`, e.g., `na.rm = TRUE`

Details

The plot is the same as that returned by `show_metseason` with the addition of points for the seasonal metrics colored by the trends estimated from `anlz_trndseason` for the specified window and justification.

Four colors are used to define increasing, decreasing, no trend, or no estimate (i.e., too few points for the window). The names and the colors can be changed using the `nms` and `fils` arguments, respectively. The `cmbn` argument can be used to combine the no trend and no estimate colors into one color and label. Although this may be desired for aesthetic reasons, the colors and labels may be misleading with the default names since no trend is shown for points where no estimates were made.

The optional `yromit` vector can be used to omit years from the plot and trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

Value

A `ggplot` object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
show_mettrndseason(mod, metfun = mean, doystr = 90, doyend = 180, justify = 'center',
  win = 4, ylab = 'Chlorophyll-a (ug/L)')
```

show_perchg *Plot percent change trends from GAM results for selected time periods*

Description

Plot percent change trends from GAM results for selected time periods

Usage

```
show_perchg(
  mod,
  baseyr,
  testyr,
  ylab,
  base_size = 11,
  xlim = NULL,
  ylim = NULL
)
```

Arguments

mod	input model object as returned by anlz_gam
baseyr	numeric vector of starting years
testyr	numeric vector of ending years
ylab	chr string for y-axis label
base_size	numeric indicating base font size, passed to theme_bw
xlim	optional numeric vector of length two for x-axis limits
ylim	optional numeric vector of length two for y-axis limits

Value

A ggplot object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <- anlz_gam(tomod, trans = 'log10')

show_perchg(mod, baseyr = 1995, testyr = 2016, ylab = 'Chlorophyll-a (ug/L)')
```

show_prd3d *Plot a 3-d surface of predictions*

Description

Plot a 3-d surface of predictions

Usage

```
show_prd3d(mod, ylab)
```

Arguments

mod	input model object as returned by anlz_gam
ylab	chr string for y-axis label

Value

a plotly surface

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')

show_prd3d(mod, ylab = 'Chlorophyll-a (ug/L)')
```

show_prddoy *Plot predictions for GAMs against day of year*

Description

Plot predictions for GAMs against day of year

Usage

```
show_prddoy(
  mod,
  ylab,
  yromit = NULL,
  linewidth = 0.5,
  alpha = 1,
  base_size = 11
)
```

Arguments

mod	input model object as returned by anlz_gam
ylab	chr string for y-axis label
yromit	optional numeric vector for years to omit from the plot, see details
linewidth	numeric indicating line width
alpha	numeric from 0 to 1 indicating line transparency
base_size	numeric indicating base font size, passed to theme_bw

Details

The optional yromit vector can be used to omit years from the plot. This may be preferred if the predicted values from the model deviate substantially from other years likely due to missing data.

Value

A ggplot object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <- anlz_gam(tomod, trans = 'log10')

show_prddoy(mod, ylab = 'Chlorophyll-a (ug/L)')
```

show_prdseason *Plot predictions for GAMs over time, by season*

Description

Plot predictions for GAMs over time, by season

Usage

```
show_prdseason(
  mod,
  ylab,
  yromit = NULL,
  base_size = 11,
  xlim = NULL,
  ylim = NULL
)
```

Arguments

mod	input model object as returned by anlz_gam
ylab	chr string for y-axis label
yromit	optional numeric vector for years to omit from the plot, see details
base_size	numeric indicating base font size, passed to theme_bw
xlim	optional numeric vector of length two for x-axis limits
ylim	optional numeric vector of length two for y-axis limits

Value

A ggplot object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
show_prdseason(mod, ylab = 'Chlorophyll-a (ug/L)')
```

show_prdseries *Plot predictions for GAMs over time series*

Description

Plot predictions for GAMs over time series

Usage

```
show_prdseries(
  mod,
  ylab,
  yromit = NULL,
  alpha = 0.7,
  base_size = 11,
  xlim = NULL,
  ylim = NULL,
  col = "brown",
  minomit = 8
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`ylab`	chr string for y-axis label
`yromit`	optional numeric vector for years to omit from the plot, see details
`alpha`	numeric from 0 to 1 indicating line transparency
`base_size`	numeric indicating base font size, passed to `theme_bw`
`xlim`	optional numeric vector of length two for x-axis limits
`ylim`	optional numeric vector of length two for y-axis limits
`col`	optional chr string for line color
`minomit`	numeric indicating number of observations in the observed data to exclude in the predicted series for a given year if provided to `yromit`

Details

The optional `yromit` vector can be used to omit years from the plot. This may be preferred if the predicted values from the model deviate substantially from other years likely due to missing data.

The `yromit` argument behaves differently for this function compared to others because of a mismatch in the timestep for the predicted time series and the observed data. The function will attempt to find "bookends" in the observed data that match with the predicted time series for each year in `yromit`. For example, if there is a gap in the observed data that spans multiple years and a single year value is included with `yromit` that is within the gap, the predicted time series will not be shown for the entire gap in the observed data even if additional years within the gap were not provided to `yromit`.

Entire years where observed data are present can also be removed from the predicted time series if they exceed a minimum number of observations defined by `minomit`. For example, if a year has at least 8 observations and `yromit` includes that year, the predicted time series for that entire year will be removed.

Value

A `ggplot` object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')

mod <- anlz_gam(tomod, trans = 'log10')

show_prdseries(mod, ylab = 'Chlorophyll-a (ug/L)')
```

`show_sumtrndseason` *Plot seasonal rates of change based on average estimates for multiple window widths*

Description

Plot seasonal rates of change based on average estimates for multiple window widths

Usage

```
show_sumtrndseason(
  mod,
  doystr = 1,
  doyend = 364,
  yromit = NULL,
  justify = c("center", "left", "right"),
  win = 5:15,
  txtsz = 6,
  cols = c("lightblue", "lightgreen"),
  base_size = 11
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`doystr`	numeric indicating start Julian day for extracting averages
`doyend`	numeric indicating ending Julian day for extracting averages
`yromit`	optional numeric vector for years to omit from the plot, see details
`justify`	chr string indicating the justification for the trend window
`win`	numeric vector indicating number of years to use for the trend window
`txtsz`	numeric for size of text labels inside the plot
`cols`	vector of low/high colors for trends
`base_size`	numeric indicating base font size, passed to `theme_bw`

Details

This function plots output from `anlz_sumtrndseason`.

The optional `yromit` vector can be used to omit years from the plot and trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

Value

A `ggplot2` plot

See Also

Other show: `show_sumtrndseason2()`

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
show_sumtrndseason(mod, doystr = 90, doyend = 180, justify = 'center', win = 2:3)
```

show_sumtrndseason2 *Plot seasonal rates of change in quarters based on average estimates for multiple window widths*

Description

Plot seasonal rates of change in quarters based on average estimates for multiple window widths

Usage

```
show_sumtrndseason2(
  mod,
  yromit = NULL,
  justify = c("center", "left", "right"),
  win = 5:15,
  txtsz = 6,
  cols = c("lightblue", "lightgreen"),
  base_size = 11
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`yromit`	optional numeric vector for years to omit from the plot, see details
`justify`	chr string indicating the justification for the trend window
`win`	numeric vector indicating number of years to use for the trend window
`txtsz`	numeric for size of text labels inside the plot
`cols`	vector of low/high colors for trends
`base_size`	numeric indicating base font size, passed to `theme_bw`

Details

This function is similar to `show_sumtrndseason` but results are grouped into seasonal quarters as four separate plots with a combined color scale.

The optional `yromit` vector can be used to omit years from the plot and trend assessment. This may be preferred if seasonal estimates for a given year have very wide confidence intervals likely due to limited data, which can skew the trend assessments.

Value

A `ggplot2` plot

See Also

Other show: `show_sumtrndseason()`

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
show_sumtrndseason2(mod, justify = 'center', win = 2:3)
```

`show_trndseason` *Plot rates of change based on seasonal metrics*

Description

Plot rates of change based on seasonal metrics

Usage

```
show_trndseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  type = c("log10", "approx"),
  justify = c("left", "right", "center"),
  win = 5,
  ylab,
  nsim = 10000,
  yromit = NULL,
  useave = FALSE,
  base_size = 11,
  nms = NULL,
  fils = NULL,
  cols = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`mod`	input model object as returned by `anlz_gam`
`metfun`	function input for metric to calculate, e.g., `mean`, `var`, `max`, etc
`doystr`	numeric indicating start Julian day for extracting averages

`doyend`	numeric indicating ending Julian day for extracting averages
`type`	chr string indicating if log slopes are shown (if applicable)
`justify`	chr string indicating the justification for the trend window
`win`	numeric indicating number of years to use for the trend window, see details
`ylab`	chr string for y-axis label
`nsim`	numeric indicating number of random draws for simulating uncertainty
`yromit`	optional numeric vector for years to omit from the output
`useave`	logical indicating if `anlz_avgseason` is used for the seasonal metric calculation, see details
`base_size`	numeric indicating base font size, passed to `theme_bw`
`nms`	optional character vector for trend names
`fils`	optional character vector for the fill of interior point colors
`cols`	optional character vector for confidence interval colors
`xlim`	optional numeric vector of length two for x-axis limits
`ylim`	optional numeric vector of length two for y-axis limits
`...`	additional arguments passed to `metfun`, e.g., `na.rm = TRUE`

Value

A `ggplot` object

Examples

```
library(dplyr)

# data to model
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl') %>%
  filter(yr > 2015)

mod <- anlz_gam(tomod, trans = 'log10')
show_trndseason(mod, doystr = 90, doyend = 180, justify = 'left', win = 4,
     ylab = 'Slope Chlorophyll-a (ug/L/yr)')
```

Index