Package ‘wqtrends’

August 17, 2023

Title Assess Water Quality Trends with Generalized Additive Models
Version 1.4.1
Date 2023-08-16


Depends R (>= 3.5)
Imports dplyr, ggplot2, lubridate, mgcv, mixmeta, plotly, purrr, tibble, tidyr, viridisLite
License CC0
Encoding UTF-8
LazyData true
RoxygenNote 7.2.3
Suggests testthat (>= 2.1.0), covr, english, knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no

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

Maintainer Marcus Beck <mbeck@tbep.org>
Repository CRAN

Date/Publication 2023-08-17 15:20:02 UTC
R topics documented:

- `anlz_avgseason` 
- `anlz_backtrans` 
- `anlz_fit` 
- `anlz_gam` 
- `anlz_metseason` 
- `anlz_mixmeta` 
- `anlz_perchg` 
- `anlz_prd` 
- `anlz_prdday` 
- `anlz_prdmatrix` 
- `anlz_pvalformat` 
- `anlz_smooth` 
- `anlz_sumtrndseason` 
- `anlz_trans` 
- `anlz_trndseason` 
- `rawdat` 
- `show_metseason` 
- `show_perchg` 
- `show_prd3d` 
- `show_prddoy` 
- `show_prdseason` 
- `show_prdseries` 
- `show_sumtrndseason` 
- `show_sumtrndseason2` 
- `show_trndseason` 

Index 27

`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)`

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
Value

A data frame of period averages

Examples

```r
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

```r
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

```r
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**

```r
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**

```r
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 parameter
- `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**

```r
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, ...)

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
...    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

```r
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)
```
Fit a mixed meta-analysis regression model of trends

Usage

```r
anlz_mixmeta(metseason, yrstr = 2000, yrend = 2019, yromit = NULL)
```

Arguments

- `metseason`: output from `anlz_metseason`
- `yrstr`: numeric for starting year
- `yrend`: numeric for ending year
- `yromit`: optional numeric vector for years to omit from, inherited from `show_metseason`

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

```r
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)
```
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  
umeric vector of starting years

testyr  
umeric 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)
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% c('chl')) %>%
  filter(yr > 2015)

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

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

**Usage**

```
anlz_prdday(mod)
```
Get prediction matrix for a fitted GAM

**Arguments**

- **mod**: input model object as returned by `anlz_gam`

**Value**

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

**Examples**

```r
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)
```

---

**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
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

**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_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
  )
Transform response variable prior to fitting GAM

Usage

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

\texttt{moddat} \hspace{1cm} \text{input raw data, one station and parameter}
\texttt{trans} \hspace{1cm} \text{chr string indicating desired type of transformation, one of } \log_{10} \text{ or ident (no transformation)}

Value

\texttt{moddat} with the \texttt{value} column transformed as indicated

See Also

Other analyze: \texttt{anlz_sumtrndseason()}, \texttt{anlz_trndseason()}

Examples

\begin{verbatim}
library(dplyr)
tomod <- rawdat %>%
  filter(station %in% 34) %>%
  filter(param %in% 'chl')
anlz_trans(tomod, trans = 'log10')
\end{verbatim}

---

\textbf{anlz_trndseason} \hspace{1cm} \textsl{Estimate seasonal rates of change based on average estimates}

Description

Estimate seasonal rates of change based on average estimates

Usage

\begin{verbatim}
anlz_trndseason(
  mod, metfun = mean, doystr = 1, doyend = 364, justify = c("center", "left", "right"), win = 5, nsim = 10000, useave = FALSE, 
  ...
)
\end{verbatim}
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.
- **...**: 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`).

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., for 1990 it 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).

Value

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

See Also

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

Examples

```r
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)

**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](https://github.com/fawda123/SFbaytrends)

**show_metseason**

Plot period (seasonal) averages from fitted GAM

**Description**

Plot period (seasonal) averages from fitted GAM
show_metseason

Usage

show_metseason(
  mod,
  metfun = mean,
  doystr = 1,
  doyend = 364,
  yrstr = 2000,
  yrend = 2019,
  yromit = NULL,
  ylab,
  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
nsim         numeric indicating number of random draws for simulating uncertainty
useave       logical indicating if `anlz_avgseason` is used for the seasonal metric calculation
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 `omityr` 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.
show_perchg

**Value**

A `ggplot` object

**Examples**

```r
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 = 2015, yrend = 2019,
  yromit = 2018, 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**

```r
show_perchg(
  mod,
  baseyr,
  testyr,
  ylab,
  base_size = 11,
  xlim = NULL,
  ylim = NULL
)
```
show_prd3d

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 = 1990, 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, size = 0.5, alpha = 1, base_size = 11)

Arguments

mod input model object as returned by anlz_gam
ylab chr string for y-axis label
size numeric indicating line size
alpha numeric from 0 to 1 indicating line transparency
base_size numeric indicating base font size, passed to theme_bw

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, base_size = 11, xlim = NULL, ylim = NULL)

Arguments

mod input model object as returned by anlz_gam
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') %>%
  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

```r
show_prdseries(
    mod,
    ylab,
    alpha = 0.7,
    base_size = 11,
    xlim = NULL,
    ylim = NULL
)
```

Arguments

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

Value

A `ggplot` object

Examples

```r
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)')
```

Description

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

Usage

show_sumtrndseason(
  mod,
  doystr = 1,
  doyend = 364,
  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
  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.

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,
  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
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.

Value

A ggplot2 plot

See Also

Other show: show_sumtrndseason()
Examples

```r
library(dplyr)

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

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

Description

Plot seasonal rates of change based on average estimates

Usage

```r
show_trndseason(
  mod, 
  metfun = mean, 
  doystr = 1, 
  doyend = 364, 
  type = c("log10", "approx"), 
  justify = c("left", "right", "center"), 
  win = 5, 
  ylab, 
  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
- `type` chr string indicating if log slopes are shown (if applicable)
- `justify` chr string indicating the justification for the trend window
show_trndseason

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
useave logical indicating if \texttt{anlz_avgseason} is used for the seasonal metric calculation
base_size numeric indicating base font size, passed to \texttt{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 \texttt{metfun}, e.g., \texttt{na.rm = TRUE})

Value

A \texttt{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

* analyze
  anlz_avgseason, 2
  anlz_backtrans, 3
  anlz_fit, 4
  anlz_gam, 5
  anlz_metseason, 6
  anlz_mixmeta, 7
  anlz_perchg, 8
  anlz_prd, 9
  anlz_prdday, 9
  anlz_prdmatrix, 10
  anlz_pvalformat, 11
  anlz_smooth, 11
  anlz_sumtrndseason, 12
  anlz_trans, 13
  anlz_trndseason, 14

* datasets
  rawdat, 16

* show
  show_metseason, 16
  show_perchg, 18
  show_prd3d, 19
  show_prdday, 20
  show_prdseries, 21
  show_sumtrndseason, 22
  show_sumtrndseason2, 24
  show_trndseason, 25

* utilities
  rawdat, 16
  s, 5
  show_metseason, 7, 16
  show_perchg, 18
  show_prd3d, 19
  show_prdday, 20
  show_prdseries, 21
  show_sumtrndseason, 22
  show_sumtrndseason2, 23
  show_trndseason, 25
  theme_bw, 17, 19–24, 26

anlz_prdday, 9
anlz_prd3d, 9
anlz_prdmatrix, 10
anlz_pvalformat, 11
anlz_smooth, 11
anlz_sumtrndseason, 12, 14, 15, 23
anlz_trans, 3, 5, 13, 15
anlz_trndseason, 13, 14, 15

gam, 5
gamDiff, 8
ggplot, 18–22, 26
ggplot2, 23, 24
mixmeta, 7
rawdat, 16

s, 5
show_metseason, 7, 16
show_perchg, 18
show_prd3d, 19
show_prdday, 20
show_prdseries, 21
show_sumtrndseason, 22
show_sumtrndseason2, 23
show_trndseason, 25

theme_bw, 17, 19–24, 26

anlz_avgseason, 2, 17
anlz_backtrans, 3
anlz_fit, 4
anlz_gam, 2, 4, 5, 6, 8–10, 12, 13, 15, 17, 19–25
anlz_metseason, 6, 7, 10, 15
anlz_mixmeta, 7, 15
anlz_perchg, 8
anlz_prd, 3, 9