Package ‘csalert’

June 15, 2023

Title  Alerts from Public Health Surveillance Data
Version  2023.6.17
Description  Helps create alerts and determine trends by using various methods to analyze public health surveillance data. The primary analysis method is based upon a published analytics strategy by Benedetti (2019) <doi:10.5588/pha.19.0002>.
Depends  R (>= 3.3.0)
License  MIT + file LICENSE
BugReports  https://github.com/csids/csalert/issues
Encoding  UTF-8
Imports  data.table, magrittr, ggplot2, glm2, cstdy, cstime, stringr
Suggests  testthat, knitr, rmarkdown, rstudioapi, glue, covidnor, cdata, csmaps, ggrepel, plnr
RoxygenNote  7.2.3
VignetteBuilder  knitr
NeedsCompilation  no
Author  Beatriz Valcarcel Salamanca [aut]
        (<https://orcid.org/0000-0000-0000-0000>),
        Chi Zhang [aut] (<https://orcid.org/0000-0003-0501-5909>),
        Richard Aubrey White [aut, cre]
        (<https://orcid.org/0000-0002-6747-1726>),
        CSIDS [cph]
Maintainer  Richard Aubrey White <hello@rwhite.no>
Repository  CRAN
Date/Publication  2023-06-15 17:30:06 UTC

R topics documented:

  short_term_trend .................................................. 2

Index  ................................................. 4
**Description**


**Usage**

```r
short_term_trend(x, ...) 
```

```r
# S3 method for class 'csfmt_rts_data_v1'
short_term_trend(
  x,
  numerator,
  denominator = NULL,
  prX = 100,
  trend_dates = 42,
  remove_last_dates = 0,
  forecast_dates = trend_dates,
  trend_isoyearweeks = ceiling(trend_dates/7),
  remove_last_isoyearweeks = ceiling(remove_last_dates/7),
  forecast_isoyearweeks = trend_isoyearweeks,
  numerator_naming_prefix = "from_numerator",
  denominator_naming_prefix = "from_denominator",
  statistics_naming_prefix = "universal",
  remove_training_data = FALSE,
  ... 
)
```

**Arguments**

- `x` Data object
- `...` Not in use.
- `numerator` Character of name of numerator
- `denominator` Character of name of denominator (optional)
- `prX` If using denominator, what scaling factor should be used for numerator/denominator?
- `trend_dates` Number of dates you want to check the trend
- `remove_last_dates` Number of dates you want to remove at the end (due to unreliable data)
- `forecast_dates` Number of dates you want to forecast into the future
- `trend_isoyearweeks` Same as `trend_dates`, but used if `granularity_geo`='isoyearweek'
- `remove_last_isoyearweeks` Same as `remove_last_dates`, but used if `granularity_geo`='isoyearweek'
short_term_trend

forecast_isoyearweeks
  Same as forecast_dates, but used if granularity_geo='isoyearweek'

numerator_naming_prefix
  "from_numerator", "generic", or a custom prefix

denominator_naming_prefix
  "from_denominator", "generic", or a custom prefix

statistics_naming_prefix
  "universal" (one variable for trend status, one variable for doubling dates), "from_numerator_and_prX"
  (If denominator is NULL, then one variable corresponding to numerator. If denominator exists, then one variable for each of the prXs)

remove_training_data
  Boolean. If TRUE, removes the training data (i.e. 1:(trend_dates-1) or 1:(trend_isoyearweeks-1)) from the returned dataset.

Value

The original csfmt_rts_data_v1 dataset with extra columns. *_trend*_status contains a factor with levels c("training", "forecast", "decreasing", "null", "increasing"), while *_doublingdays* contains the expected number of days before the numerator doubles.

Examples

```r
  d <- cstidy::nor_covid19_icu_and_hospitalization_csfmt_rts_v1
  d <- d[granularity_time="isoyearweek"]
  res <- csalert::short_term_trend(
    d, 
    numerator = "hospitalization_with_covid19_as_primary_cause_n", 
    trend_isoyearweeks = 6 
  )
  print(res[, .( 
    isoyearweek, 
    hospitalization_with_covid19_as_primary_cause_n, 
    hospitalization_with_covid19_as_primary_cause_trend0_42_status 
  )])
```
Index

short_term_trend, 2