Package 'FluMoDL'

September 13, 2019

Type Package

Imports stats, utils, splines, tsModel

Title Influenza-Attributable Mortality with Distributed-Lag Models

Version 0.0.3

Maintainer Theodore Lytras <thlytras@gmail.com>

Depends dlnm, mvmeta

Description Functions to estimate the mortality attributable to influenza and temperature, using distributed-lag nonlinear models (DLNMs), as first implemented in Lytras et al. (2019) <doi:10.2807/1560-7917.ES.2019.24.14.1800118>. Full descriptions of underlying DLNM methodology in Gasparrini et al. <doi:10.1002/sim.3940> (DLNMs), <doi:10.1186/1471-2288-14-55> (attributable risk from DLNMs) and <doi:10.1002/sim.5471> (multivariate meta-analysis).

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Theodore Lytras [aut, cre] (<https://orcid.org/0000-0002-4146-4122>), Antonio Gasparrini [ctb] (<https://orcid.org/0000-0002-2271-3568>), Shuangcai Wang [ctb]

Repository CRAN

Date/Publication 2019-09-13 10:50:02 UTC

R topics documented:

`addPredictions` *Add predictions to summary.FluMoDL objects*

Description

This function uses the data available in a FluMoDL object to generate predictions (in the form of `crosspred` objects) for a `summary.FluMoDL` object.

Usage

```
addPredictions(s, m)
```

Arguments

`s`	An object of class `summary.FluMoDL` (normally holding BLUP or pooled estimates, i.e. `s$type` will equal `"blup"` or `"pooled"`) for which predictions will be generated.
`m`	An object of class FluMoDL, which provides the original (untransformed) predictor data to create the predictions.

Details

Creating a cross-basis matrix (to use as covariate in a Distributed-Lag Nonlinear Model) transforms and scales the original predictor. Interpreting the model coefficients requires revisiting the cross-basis matrix and backtransforming to the original predictor, in order to generate predicted effect estimates for specific values of predictor and lag.

For this reason, `summary.FluMoDL` objects created from a multivariate meta-analysis, containing pooled or BLUP coefficients, do not contain predictions (their `$pred` element is `NULL`) because they have no reference to an original predictor. This is what `addPredictions()` does: it uses the cross-basis matrices from a FluMoDL object `m` to calculate predictions with the coefficients in the `summary.FluMoDL` object `s`. It provides the necessary "context" in which to interpret the model coefficients.

Value

The function returns the summary.FluMoDL object s, with predictions added (as element $pred)

See Also

summary.FluMoDL, fitFluMoDL, crosspred

---

attrdl *Attributable risk from distributed lag nonlinear models*

Description

This is a general function that computes attributable risk (attributable numbers or fractions) from distributed lag nonlinear models.

Usage

```
attrdl(x, basis, cases, model = NULL, coef = NULL, vcov = NULL,
  type = "af", dir = "back", tot = TRUE, cen, range = NULL,
  sim = FALSE, nsim = 5000, sub = 1:length(cases))
```

Arguments

x	An exposure vector OR (only for dir="back") a matrix of lagged exposures, for which the attributable risk needs to be computed.
basis	The object of class "crossbasis" used for fitting the model.
cases	The cases vector OR (only for dir="forw") the matrix of future cases corresponding to x.
model	The fitted model. You need to provide either this, or arguments coef and vcov. *The model MUST have a log link function.*
coef	Coefficients for basis IF model is not provided
vcov	Variance-covariance matrix for basis IF model is not provided
type	Either "an" or "af" for attributable number or attributable fraction
dir	Either "back" or "forw" for backward or forward perspectives of attributable risk
tot	If TRUE, the total attributable risk is computed (number or fraction, depending on argument type)
cen	The reference value used as the counterfactual scenario (the comparator)
range	The range of exposure (for which the attributable risk, compared to cen, is calculated). If NULL, the whole range is used.
sim	Set to TRUE if Monte Carlo simulation samples should be returned.
nsim	Number of simulation samples desired (only for nsim=TRUE).
sub	Subset of cases for which to calculate the attributable risk (as an integer index vector). Defaults to 1:length(cases). Argument cases should be a vector (not a matrix).

Details

Original function and documentation written by Antonio Gasparrini and available here. Slightly amended by Theodore Lytras for use with FluMoDL.

Documentation below copied from the original source.

This function computes the attributable fraction or number for a specific exposure scenario and associated cases, given an estimated exposure-lag-response association defined by a DLNM. Either forward or backward versions of attributable risk measures are available in this setting. The method is described by Gasparrini and Leone (2014), see references below. The function works in combination with other functions in the package dlnm, which is assumed to be available.

The exposure and cases are provided by the arguments x and cases, respectively. The original cross-basis and fitted model containg it used for estimation are provided by the arguments basis and model, respectively. Alternatively, the user can provide estimated coefficients and (co)variance matrix with coef and vcov.

The function works both with time series and non-time series data. In a time series setting, both x and cases represent a complete series of ordered observations. More generally, the user can apply this function for any kind of data: in this case x must be a matrix of lagged exposures when dir="back", and cases must be a matrix of future cases dir="forw". The function can compute the total attributable risk (if tot=TRUE, the default) or the contribution for each observation. The argument cen defines the value used as counterfactual scenario.

If sim=TRUE, the function computes samples of the attributable risk measures by simulating from the assumed normal distribution of the estimated coefficients (only implemented for total estimates). These samples can be used to defined empirical confidence intervals.

Value

By default, a numeric scalar representing the total attributable fraction or number. If sim=TRUE, a vector of the simulated samples with length nsim. If tot=FALSE, a vector with contributions for all the observations (see Note below). These quantities are defined versus a counterfactual scenario defined through the argument cen.

Note

The function handles missing values in both the x and cases objects, excluding incomplete observations (also due to lagging) accordingly. However, the total attributable number is rescaled to match the fraction using as denominator the total observed number in cases. This approach uses the all the available information even in the presence of missing values in x. All of this under the assumption that the missing mechanism is unrelated with both exposure and cases values.

The functions can be also used with estimates from DLNMs reduced to the overall cumulative exposure-response through the function crossreduce in the package dlnm. In this case, the modified coefficients and (co)variance matrix of the reduced cross-basis in basis must be passed using the arguments coef and vcov. This option can be useful when the original estimates from the full cross-basis are not available any more, for example following a meta-analysis. Given the lag-specific estimates are not available in this case, only the forward version of attributable risk (dir="forw") can be computed. See Gasparrini and Leone (2014) for further info.

Author(s)

Original author: Antonio Gasparrini <<antonio.gasparrini@lshtm.ac.uk>>

References

- Gasparrini A, Leone M. Attributable risk from distributed lag models. BMC Med Res Methodol 2014;14:55.

Examples

```
# load the package
library(FluMoDL)  # package dlnm is automatically loaded

# define the cross-basis and fit the model
cb <- crossbasis(chicagoNMMAPS$temp, lag=30, argvar=list(fun="bs",
   knots=c(-10,3,18)), arglag=list(knots=c(1,3,10)))
library(splines)
model <- glm(death ~ cb + ns(time, 7*14) + dow,
   family=quasipoisson(), chicagoNMMAPS)

# global backward attributable risk of temperature (number and fraction)
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,type="an",cen=21)
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,cen=21)

# global forward attributable fraction
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,dir="forw",cen=21)

# empirical confidence intervals
afsim <- attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,cen=21,
   sim=TRUE,nsim=1000)
quantile(afsim,c(2.5,97.5)/100)

# attributable fraction component due to heat and cold
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,cen=21,range=c(21,100))
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,cen=21,range=c(-100,21))

# daily attributable deaths in the second month
attrdl(chicagoNMMAPS$temp,cb,chicagoNMMAPS$death,model,type="an",
   tot=FALSE,cen=21)[31:60]
```

`attrMort` *Influenza- and temperature-attributable mortality for a FluMoDL object*

Description

This function uses an object of class FluMoDL to calculate mortality attributed to influenza and/or temperature.

Usage

```
attrMort(m, par = c("H1", "H3", "B", "temp", "RSV"), sel = "week",
  from = NULL, to = NULL, temprange = "cold", ci = TRUE,
  nsim = 5000, mcsamples = FALSE, progress = TRUE, blup = FALSE)
```

Arguments

m	An object of class FluMoDL.
par	A character vector indicating which exposures to calculate the attributable mortality for. Defaults to c("H1","H3","B","temp","RSV"), which indicates all three influenza proxies, temperature and RSV (if it exists in the model).
sel	For which time period(s) to calculate attributable mortality. This can be one of several choices. For sel="week" (the default) and sel="season" attributable mortality is calculated for each week or each season respectively. One can also provide to sel a list of index vectors (integer or logical) corresponding to particular rows of m$data, or a matrix of logical index vectors, or a single index vector. Note that the index vectors should point to *consecutive rows* of m$data.
from	Week (integer, in YYYYWW format) or season to start from, in case sel="week" or sel="season" respectively.
to	Week (integer, in YYYYWW format) or season to end with, in case sel="week" or sel="season" respectively.
temprange	In case temperature-attributable mortality is calculated (argument par includes "temp"), this argument specifies the temperature range or interest. This can be one of several choices. If temprange="cold" (the default) mortality attributable to cold temperatures is calculated, i.e. temperatures below the MMP (minimum mortality point). If temprange="heat" mortality attributable to hot temperatures is calculated, i.e. those above the MMP. If temperature="all" the entire range of temperatures is used, i.e. any temperature other than the MMP. Alternatively one can provide a numeric vector of length two, indicating a specific temperature range; this can also be provided as a *character* vector of length two, where one of the elements can be the word "MMP", which will be replaced with the MMP temperature.
ci	If TRUE, empirical (Monte Carlo) 95 for all attributable mortality estimates.
nsim	Number of Monte Carlo simulations to run per attributable mortality estimate. Defaults to 5000. Increase if higher precision is required (and you don't mind the wait).
mcsamples	If TRUE, return all Monte Carlo simulation samples in the output. See below.
progress	If TRUE, a progress bar appears if Monte Carlo simulations are run and if there are more than three time periods selected in argument sel. Set to FALSE to suppress the progress bar.

`blup` If `FALSE` (the default), the model coefficients stored in `m$model` are used for the calculation of attributable mortality. If `TRUE`, the coefficients stored in the FluMoDL object are used; if `blup=TRUE` but `blup(m)` is `NULL`, a warning is generated. Alternatively, `blup` can be another object of class `summary.FluMoDL`, whose coefficients are used for the calculation.

Details

All attributable mortalities are calculated using the "backward" perspective, meaning the mortality at any given day that is attributable to exposures up to 30 days previously (=the maximum lag).

Confidence intervals (when `ci=TRUE`) are obtained empirically through Monte Carlo simulations; this can take quite some time if lots of CIs need to be calculated (for example if `sel=TRUE`). For this reason, a progress bar is shown by default in this case (which can be suppressed by `progress=FALSE`).

Temperature-attributable mortalities are by default calculated for cold temperatures, i.e. temperatures lower than the minimum mortality point (MMP). Note, however, that the adjustment in the FluMoDL is made for the entire range of daily mean temperatures, not just for cold. Therefore mortality attributable to any range of temperatures can be calculated, e.g. for heat, extreme cold, extreme heat, etc. See argument `temprange` above for details.

Value

If `mcsamples=FALSE` (the default), a data.frame is returned with columns named 'FluH1', 'FluH3', 'FluB' and 'Temp' (and/or 'RSV'), depending on the argument `par`, and also 'FluH1.lo', 'FluH1.hi', 'FluH3.lo', ..., if `ci=TRUE`. Each row in the output corresponds to a selection made in argument `sel`, for example if `sel="week"` (the default) rows correspond to each week available in the data. If all influenza types/subtypes are selected in `par`, a column named 'AllFlu' is also calculated automatically, with the mortality (and 95 attributable to all influenza types/subtypes.

If `mcsamples=TRUE`, a list is returned with elements 'result' and 'mcsamples'. The first contains the data.frame with point estimates of influenza- and/or temperature-attributable mortality, as before (no 95 element contains a list of the Monte Carlo simulation samples for each parameter in `par`.

References

- Lytras T, Pantavou K, Mouratidou E, Tsiodras S. Mortality attributable to seasonal influenza in Greece, 2013 to 2017: variation by type/subtype and age, and a possible harvesting effect. Euro Surveill. 2019;24(14):pii=1800118 (PubMed)
- Gasparrini A, Leone M. Attributable risk from distributed lag models. BMC Med Res Methodol 2014;14:55.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
```

```
# Calculate influenza-attributable estimates by season, until 2016-17:
attr1 <- attrMort(m, par=c("H1","H3","B"), sel="season", to=2016)
attr1

# Calculate influenza-attributable estimates by week, only point
#    estimates, for the 2014-15 season:
attr2 <- attrMort(m, par=c("H1","H3","B"), sel="week",
    from=201440, to=201520, ci=FALSE)
attr2


# Calculate mortality attributable to temperatures below 5 celsius, for
#    the period of January 2017:
attr3 <- attrMort(m, par="temp",
    sel=with(m$data, which(dates>="2017-1-1" & dates<="2017-1-31")),
    temprange=c(5,-20))


# Calculate attributable mortalities for the entire 2017-18 season, and
#    return the Monte Carlo simulation samples in the output
attr4 <- attrMort(m, sel="season", from=2017, to=2017, mcsamples=TRUE)
```

`blup.FluMoDL` *Get or set BLUP coefficients for a FluMoDL object*

Description

This retrieves or sets the BLUP coefficients for a particular FluMoDL object.

Usage

```
## S3 method for class 'FluMoDL'
blup(object, ...)

blup(object) <- value

## S3 replacement method for class 'FluMoDL'
blup(object) <- value
```

Arguments

`object`	An object of class `FluMoDL`
`...`	Further arguments passed to or from other methods.
`value`	An object of class `summary.FluMoDL`, holding BLUP estimates to be assigned to `x`

Value

For `blup.FluMoDL`, the returned object of class `summary.FluMoDL` holding the BLUP coefficients associated with the FluMoDL object.

`fitFluMoDL` *Fit a FluMoDL object*

Description

This function fits a FluMoDL object. This is a distributed lag nonlinear model (DLNM), of quasipoisson family and with log link, which estimates the association between mortality (as outcome) and daily mean temperatures and type-specific influenza incidence proxies (as exposures), adjusted for covariates.

Usage

```
fitFluMoDL(deaths, temp, dates, proxyH1, proxyH3, proxyB, yearweek,
  proxyRSV = NULL, smooth = TRUE, periodic = TRUE)
```

Arguments

`deaths`	A vector of *daily* deaths, of equal length to argument `` `dates` ``
`temp`	A vector of *daily* mean temperatures, of equal length to argument `` `dates` ``
`dates`	A vector of dates (of class `Date`)
`proxyH1`	A vector of *weekly* influenza A(H1N1)pdm09 incidence proxies, of equal length to argument `` `yearweek` ``
`proxyH3`	A vector of *weekly* influenza A(H3N2) incidence proxies, of equal length to argument `` `yearweek` ``
`proxyB`	A vector of *weekly* influenza B incidence proxies, of equal length to argument `` `yearweek` ``
`yearweek`	An integer vector of weeks, in *yyyyww* format
`proxyRSV`	An *optional* vector of *weekly* RSV incidence proxies, of equal length to argument `` `yearweek` ``. (This is an experimental feature, and this argument might be removed in the future.)
`smooth`	`TRUE` (the default) if smoothing is to be applied to the influenza incidence proxies when converting them to a daily series.
`periodic`	Should a periodic B-spline term be included in the model? Defaults to `TRUE`.

Details

Objects of class 'FluMoDL' contain the model, the associated data, estimates of the predicted associations and other information. These objects can be further used as input for function `attrMort`, to calculate influenza-attributable and temperature-attributable mortalities for any period in the data (and any temperature range). Methods `print()`, `coef()` and `vcov()` have been defined for objects of class 'FluMoDL' (see below), and also `summary()`.

FluMoDL uses a DLNM with the *daily* number of deaths as the outcome. Covariates include the following:

- A `cross-basis matrix` for temperature. The exposure-response relationship is modelled with a quadratic B-spline with internal knots placed at the 10th, 75th and 90th percentile of the temperatures distribution. The lag-response relationship is modelled with a natural cubic spline with three internal knots equidistant in the log scale.
- Three `cross-basis matrices` for influenza incidence proxies for each type/subtype: A(H1N1)pdm09, A(H3N2) and B. These normally are equal to a sentinel Influenza-Like Illness (ILI) rate, times the laboratory swab samples Percentage Positive (implying an approximately constant case fatality ratio for each influenza type. The lag-response relationship is specified as above (for temperature).
- A periodic B-spline term to model seasonality, with three equidistant internal knots according to day of the year. Can optionally be suppressed by setting argument `periodic` to `FALSE`.
- A linear trend, and a factor variable for day of the week.
- *Optionally*, a `cross-basis matrix` for an RSV incidence proxy, with specification identical to those for influenza. If given, it will be included in the model and output, and it will be possible to calculate mortality attributable to RSV with `attrMort`. This is an experimental feature; it might be removed in the future.

Value

An object of class 'FluMoDL'. This is a list containing the following elements:

$data A `data.frame` with the data used to fit the model. Rows correspond to days in argument `dates`. The columns are named: `yearweek`, `dates`, `deaths`, `temp`, (for temperature), `proxyH1`, `proxyH3`, `proxyB`, `t` (linear trend, with values `1:nrow(m$data)`), `doy` (day of year, use to calculate the periodic B-spline term to model seasonality) and `dow` (day of the week). Also column `proxyRSV` if the relevant argument is provided.

$model The fitted model; an object of class `glm` and of 'quasipoisson' family with log link.

$basis A list with names 'temp', 'proxyH1', 'proxyH3' and 'proxyB' (and proxyRSV, if provided in the function arguments), containing the cross-basis matrices that are used as exposures in the model. See `crossbasis`.

$MMP The Minimum Mortality Point, i.e. the temperature where mortality is lowest.

$pred A list with names 'temp', 'proxyH1', 'proxyH3' and 'proxyB' (and 'proxyRSV' if provided in the function arguments), containing predictions (in the form of `crosspred` objects) for each exposure. These can be plotted in both the exposure-response and lag-response dimensions, see `crosspred`, `plot.crosspred` and the examples below.

$blup This element is NULL when creating the object, but can receive a `summary.FluMoDL` object that contains Best Linear Unbiased Predictor (BLUP) coefficients, to be used when estimating attributable mortality. Can be retrieved or set with the `blup.FluMoDL` method

Objects of class 'FluMoDL' have methods `print()`, `coef()` and `vcov()`. `coef()` returns a list of numeric vectors, with names 'proxyH1', 'proxyH3' and 'proxyB' (and 'proxyRSV' if provided in the function arguments), containing the model coefficients for these cross-basis terms. Similarly `vcov()` returns a list of variance-covariance matrices for the same terms.

References

- Lytras T, Pantavou K, Mouratidou E, Tsiodras S. Mortality attributable to seasonal influenza in Greece, 2013 to 2017: variation by type/subtype and age, and a possible harvesting effect. Euro Surveill. 2019;24(14):pii=1800118 (PubMed)
- Gasparrini A, Armstrong B, Kenward MG. Distributed lag non-linear models. Stat Med 2010;29(21):2224–34.
- Gasparrini A, et al. Mortality risk attributable to high and low ambient temperature: a multi-country observational study. Lancet 2015 Jul 25;386(9991):369–75.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
m

# Plot the association between A(H1N1)pdm09 activity and mortality
#   and the overall temperature-mortality association:
plot(m$pred$proxyH1, "overall")
plot(m$pred$temp, "overall")

# Add the Minimum Mortality Point to the plot:
abline(v=m$MMP)

# Check the lag-response dimension for the A(H1N1)pdm09 - mortality
#   association, for all proxy values, and for an indicative value of 30.
plot(m$pred$proxyH1) # Produces a 3D plot, see ?plot.crosspred
plot(m$pred$proxyH1, var=30)

# Have a look at the data associated with this FluMoDL:
str(m$data)
tail(m$data)
```

`greece` *Greece mortality and influenza data*

Description

Surveillance data from Greece used to estimate influenza-attributable mortality using FluMoDL, covering the period from May 2013 to October 2017. Contains the following:

- A daily time series of (all-cause) deaths
- A time series of daily mean temperatures
- A weekly series of Influenza-Like Illness (ILI) rates, calculated via sentinel surveillance
- Three weekly series of laboratory sample percentage positives by influenza type and subtype: A(H1N1)pdm09, A(H3N2) and B. These can be multiplied with the respective ILI rates, to create type-specific influenza incidence proxies (see Goldstein et al. PLoS Med. 2011;8(7):e1001051)

Usage

```
data(greece)
```

Format

An list with two elements of class `data.frame`: ‘greece$daily‘ contains a `data.frame` with columns ‘date‘ (of class `Date`), ‘deaths‘ and ‘temperature‘. ‘greece$weekly‘ contains a `data.frame` with columns ‘yearweek‘ (integer, in YYYYWW format), ‘ILI‘ (ILI rate per 1000 patient consultations), ‘ppH1‘, ‘ppH3‘ and ‘ppB‘ (percentage positives for A(H1N1)pdm09, A(H3N2) and B respectively).

Source

Greek National Public Health Organization (formerly the Hellenic Centre for Disease Control and Prevention)

References

Lytras T, Pantavou K, Mouratidou E, Tsiodras S. Mortality attributable to seasonal influenza in Greece, 2013 to 2017: variation by type/subtype and age, and a possible harvesting effect. Euro Surveill. 2019;24(14):pii=1800118 (PubMed)

Examples

```
data(greece)
str(greece$daily)
str(greece$weekly)
```

hasPeriodic *Does object include a periodic B-spline term?*

Description

This method checks whether a 'FluMoDL' object includes a periodic B-spline term in its parametrization or not. By default FluMoDL objects are created with a periodic term, unless argument `periodic` in `fitFluMoDL` is set to `FALSE`

Usage

```
hasPeriodic(x)
```

Arguments

x	An object of class `FluMoDL`

Value

`TRUE` if the model includes a periodic term, `FALSE` if it does not.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
hasPeriodic(m)   # Returns TRUE
```

hasRSV *Does object have a term for RSV?*

Description

This method checks whether a 'FluMoDL' or 'summary.FluMoDL' object contains a `cross-basis term` for RSV (Respiratory Syncytial Virus) incidence proxy, or contains only terms for influenza incidence proxies.

Usage

```
hasRSV(x)
```

Arguments

x	An object of class FluMoDL or summary.FluMoDL

Value

TRUE if the model contains a term for RSV, FALSE if it does not.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
hasRSV(m)   # Returns FALSE
hasRSV(summary(m))   # Also returns FALSE
```

---

isoweek *Calculate the ISO week & year for a Date*

Description

This function takes a vector of Date objects and calculates the week and year according to ISO 8601. It is flexible in its output.

Usage

```
isoweek(x, type = "both_num", sep = "-", inv = FALSE,
  colnames = c("isoyear", "isoweek"))
```

Arguments

x	A vector of class Date (of length >=1)
type	A string (one of "week", "year", "both_text", "both_num" or "matrix") that determines the kind of output the function returns. See "Return value".
sep	Seperator between year and week, applicable if type="both_text"
inv	If type="both_text", and inv=FALSE, then year comes before week. If inv=TRUE, week comes before year.
colnames	Names for the matrix columns if type="matrix"

Details

This function calculates the week number according to ISO 8601. Note that dates near the start or end of a given year may belong to the previous or next year respectively, thus the year needs to be calculated too.

Value

Different according to the function's type argument. If "both_num" (the default), a vector of 6-digit integers is returned, in a YYYYWW format. If "week" or "year", only the week number or year is returned, respectively. If "both_text", then a character vector of the same length as x is returned, containing both the year and week number, seperated by sep, and inverted if inv=TRUE. Finally, if type="matix", both year and week numbers are returned in a two-column matrix, with the columns named as in colnames.

Examples

```
isoweek(Sys.Date())
isoweek("1980-8-19", "both_text", sep="/", inv=TRUE)
isoweek(c("2004-5-31", "2006-6-10", "2007-8-20"), "matrix")
```

---

isoweekStart *Calculate the start date of a given ISO week*

Description

This function takes a vector of ISO week numbers (of the form YYYYWW) and returns a Date vector with the first Monday of each week. It is essentially the inverse function of isoweek.

Usage

```
isoweekStart(x)
```

Arguments

x	A numeric vector of ISO week numbers (of format YYYYWW)

Value

A vector of class Date and length equal to x, containing the start date (first Monday) of each ISO week.

Examples

```
isoweekStart(201740) # Start of 2017-18 influenza surveillance
isoweekStart(isoweek(Sys.Date()))
```

linterp *Linearly interpolate missing values in a numeric vector*

Description

This value fills in missing values (NAs) in a numeric vector by linear interpolation

Usage

```
linterp(x, max_allow = 3)
```

Arguments

x	The numeric vector to interpolate. The first and last element must not be NA, otherwise an error is generated.
max_allow	Maximum number of consecutive missing values to allow. If there is any number of consecutive NA values in x longer than max_allow, the function will fail with an error. Set to NULL to fully disable this check.

Details

This function can be handy when running fitFluMoDL, for example to fill in small gaps in the temperatures vector. But it can be more generally useful as well.

Value

The numeric vector x, with any missing values replaced by linear interpolants.

metaFluMoDL *Multivariate meta-analysis for FluMoDL objects*

Description

This function runs multivariate meta-analysis (using package mvmeta) on the first-stage coefficients of influenza (and possibly RSV) incidence proxies for multiple 'FluMoDL' object summaries.

Usage

```
metaFluMoDL(summaries, par = c("H1", "H3", "B", "RSV"))
```

Arguments

summaries	A *list* of objects of class summary.FluMoDL (at least two), representing the first-stage analyses. If the list is named, the names are kept in the output object and can be retrieved with names(), see below.
par	For which model terms (sets of coefficients) to run the meta-analysis? Defaults to c("H1","H3","B","RSV"), which indicates all three influenza proxies and RSV (for those summaries that have included an RSV term). It is unlikely that you'll want to alter this default.

Value

Returns an object of class 'metaFluMoDL'. This is a list of objects of class mvmeta, representing the results of the multivariate random-effects meta-analysis for the sets of coefficients corresponding to each term in argument par; they can be accessed directly using the $ operator as $proxyH1, $proxyH3 and $proxyB (and also $proxyRSV if there were RSV terms in at least two elements of summaries and par included "RSV" – in which case, hasRSV() returns TRUE for objects of class 'metaFluMoDL').

However, some methods have been redefined for class 'metaFluMoDL', and do not work the same as in simple lists. In particular: length() returns the number of summaries (number of "studies") meta-analyzed and names() returns the names of these summaries (if the list in summaries argument was named).

In addition, the [[and [operators have been redefined for class 'metaFluMoDL', and now return the Best Linear Unbiased Predictor (BLUP) estimates for the selected summaries ("studies"), as objects of class summary.FluMoDL; selection can be made the usual way, with a logical or numeric index vector, or with the summary names (as provided by names). [returns a *list* of summary.FluMoDL objects, whereas [[returns a single object. The returned objects contain the string "blup" in their $type element, to distinguish them from first-stage model summaries or pooled result summaries. In their $description element, they contain the name of the respective summary ("study") if a named list had been provided in the summaries argument of metaFluMoDL(). And finally, they contain no $pred element, as they are not associated with a particular dataset and cross-basis matrices (which is a prerequisite to create crosspred objects).

The pooled coefficients (for all three or four incidence proxies) can be obtained with function pooled(), which also returns an object of class summary.FluMoDL that you can further use.

References

- Gasparrini A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. Stat Med 2012;31(29):3821–39.

See Also

summary.FluMoDL, pooled

NOAA_allStations *Get list of weather stations from NOAA*

Description

Download the list of all available weather stations from NOAA, or only those for a specific country and period

Usage

```
NOAA_allStations(force_retrieve = FALSE)

NOAA_countryStations(fips, from = NULL, to = NULL)
```

Arguments

force_retrieve	If TRUE download the list again from NOAA (even if it was already downloaded previously). Defaults to FALSE, so that download happens only once per session.
fips	2-letter country FIPS ID (full list of codes at ftp://ftp.ncdc.noaa.gov/pub/data/gsod/country-list.txt).
from	Lower limit of reporting period (as class Date). Only retrieve stations whose period of record ends at or after this date.
to	Upper limit of reporting period (as class Date). Only retrieve stations whose period of record begins at or before this date.

Details

NOAA_allStations() downloads the list of all available weather stations from NOAA, found in ftp://ftp.ncdc.noaa.gov/pub/data/noaa/isd-history.csv, and returns it as a data.frame. The data are downloaded only once per R session, the first time this function is used, and are then stored internally for further retrievals.

NOAA_countryStations() retrieves the list for a specific country only (or several countries, if length(ctry)>1), and possibly only for a specific period of record.

Value

A data.frame with the following columns (copy-pasted from NOAA):

usaf Air Force station ID. May contain a letter in the first position.

wban NCDC WBAN number

ctry FIPS country ID

st State for US stations

icao ICAO ID

lat Latitude in thousandths of decimal degrees

lon Longitude in thousandths of decimal degrees

elev.m. Elevation in meters

begin Beginning Period Of Record. There may be reporting gaps within the P.O.R.

end Ending Period Of Record. There may be reporting gaps within the P.O.R.

Note that columns `begin` and `end` in the output are of class `Date`.

`NOAA_getGSOD` *Get daily weather summaries from NOAA*

Description

Downloads the daily weather summaries for a set of weather stations and a set of years.

Usage

```
NOAA_getGSOD(stations, years, match.columns = "station.name",
  progress = TRUE)
```

Arguments

`stations`	A `data.frame` with the stations for which data are to be retrieved. It can be a subset of the `data.frames` returned from `NOAA_allStations` or `NOAA_countryStations`. At a minimum it should contain columns 'usaf' and 'wban'.
`years`	An integer vector of years (from 1901 to current year) for which data are to be retrieved.
`match.columns`	`NULL` or (optionally) a vector of column names that can be found in `stations`. If given, these are included in the output after matching with the respective weather stations.
`progress`	If `TRUE` (the default), a progress bar appears if more than three files are to be downloaded from NOAA. Set to `FALSE` to suppress the progress bar. For example, one can include a grouping variable in `stations` (such as region code) and give its name in `match.columns` for it to be included in the function output. This facilitates aggregating the output by the grouping variable.

Value

A `data.frame` with the following columns (adapted from NOAA):

usaf Air Force station ID

wban NCDC WBAN number

date Date (of class `Date`

temp Mean temperature for the day, in degrees Celsius to tenths.

tempC Number of observations used in calculating mean temperature.

dewp Dew point for the day, in degrees Celsius to tenths.

dewpC Number of observations used in calculating mean dew point.

slp Mean sea level pressure for the day,in millibars to tenths.

slpC Number of observations used in calculating mean sea level pressure.

stp Mean station pressure for the day in millibars to tenths.

stpC Number of observations used in calculating mean station pressure.

visib Mean visibility for the day in miles to tenths.

visibC Number of observations used in calculating mean visibility.

wdsp Mean wind speed for the day in knots to tenths.

wdspC Number of observations used in calculating mean wind speed.

mxspd Maximum sustained wind speed reported for the day, in knots to tenths.

gust Maximum wind gust reported for the day, in knots to tenths.

maxtemp Maximum temperature reported during the day, in degrees Celsius to tenths. Time of max temp report varies by country and region, so this will sometimes not be the max for the calendar day.

maxtempF Blank indicates max temp was taken from the explicit max temp report and not from the 'hourly' data. An asterisk (*) indicates max temp was derived from the hourly data (i.e., highest hourly or synoptic-reported temperature).

mintemp Minimum temperature reported during the day, in degrees Celsius to tenths. Time of min temp report varies by country and region, so this will sometimes not be the min for the calendar day.

mintempF Blank indicates min temp was taken from the explicit min temp report and not from the 'hourly' data. An asterisk (*) indicates min temp was derived from the hourly data (i.e., lowest hourly or synoptic-reported temperature).

prcp Total precipitation (rain and/or melted snow) reported during the day, in inches and hundredths; will usually not end with the midnight observation, i.e. may include latter part of previous day. Zero indicates no measurable precipitation (includes a trace). Note: Many stations do not report '0' on days with no precipitation, therefore NA will often appear on these days. Also, for example, a station may only report a 6-hour amount for the period during which rain fell. See 'prcpF' field for source of data.

prcpF A = 1 report of 6-hour precipitation amount. B = Summation of 2 reports of 6-hour precipitation amount. C = Summation of 3 reports of 6-hour precipitation amount. D = Summation of 4 reports of 6-hour precipitation amount. E = 1 report of 12-hour precipitation amount. F = Summation of 2 reports of 12-hour precipitation amount. G = 1 report of 24-hour precipitation amount. H = Station reported '0' as the amount for the day (eg, from 6-hour reports), but also reported at least one occurrence of precipitation in hourly observations; this could indicate a trace occurred, but should be considered as incomplete data for the day. I = Station did not report any precip data for the day and did not report any occurrences of precipitation in its hourly observations; it's still possible that precip occurred but was not reported.

sndp Snow depth in inches to tenths–last report for the day if reported more than once. Note: Most stations do not report '0' on days with no snow on the ground–therefore, NA will often appear on these days.

frshtt Indicators (1 = yes, 0 = no/not reported) for the occurrence during the day of: Fog ('F' - 1st digit); Rain or Drizzle ('R' - 2nd digit); Snow or Ice Pellets ('S' - 3rd digit); Hail ('H' - 4th digit); Thunder ('T' - 5th digit); Tornado or Funnel Cloud ('T' - 6th digit).

Note that, compared to the original NOAA output (for details see ftp://ftp.ncdc.noaa.gov/pub/data/gsod/readme.txt), all temperatures are automatically converted to degrees Celsius (instead of Fahrenheit) and all missing indicators are replaced with NAs.

---

pbs *Periodic B-Spline Basis for Polynomial Splines*

Description

Generate the periodic B-spline basis matrix for a polynomial spline.

Usage

```
pbs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE,
  Boundary.knots = range(x))
```

Arguments

x	the predictor variable. Missing values are allowed.
df	degrees of freedom; one can specify 'df' rather than 'knots'; 'pbs()' then chooses 'df - 1' knots at suitable quantiles of 'x' (which will ignore missing values).
knots	the _internal_ breakpoints that define the spline. The number of internal knots must be greater than or equal to degree polynomial regression. See also 'Boundary.knots'.
degree	degree of the piecewise polynomial-default is 3 for cubic splines.
intercept	if 'TRUE', an intercept is included in the basis; default is 'FALSE'
Boundary.knots	boundary points at which to set the period of the perodic B-spline basis(default the range of the data). If both 'knots' and 'Boundary.knots' are supplied, the basis parameters do not depend on 'x'. Data CAN NOT be extended beyond 'Boundary.knots'. Typical Bourday knots are start and end values of period.

Details

This function and documentation is copied from the CRAN package pbs by Shuangcai Wang <<swang1@gmail.com>>

Value

A matrix of dimension 'length(x) * (df)', where either 'df' was supplied or if 'knots' were supplied, 'df = length(knots) + intercept'. Attributes are returned that correspond to the arguments to 'pbs', and explicitly give the 'knots', 'Boundary.knots' etc for use by 'predict.pbs()'.

pbs()' is based on the function 'spline.des()' in package splines. It generates a basis matrix for representing the family of piecewise polynomials with the specified interior knots and degree, evaluated at the values of 'x'. A primary use is in modeling formulas to directly specify a piecewise polynomial term in a model.

Examples

```
require(stats); require(graphics); require(splines)
x = seq(1,628)/100
z = rep(seq(1, 314)/100, 2)

pbs(x, df = 5, Boundary.knots = c(0, 2*pi))
pbs(x, knots=c(pi/2, pi, pi*3/2), Boundary.knots = c(0, 2*pi))
#### example of one periodic functions
y= sin(x) + cos(2*x) +
rnorm(628, 0, 0.1) ## x has a period of 2*pi
## df method, need to use large enough df to get a better fit.
## May use max loglik to choose optimal df
summary( fm1 <- lm(y ~ pbs(x, df = 10, Boundary.knots = c(0, 2*pi))) )
plot(x, y, xlab = "x", ylab = "sin(x)", pch="x", cex=.5)

lines(x, predict(fm1, data.frame(x=x, z=z)), col='blue')
lines(x, sin(x) + cos(2*x), col='red')

## knots methods, usually selected at turning points
summary( fm2 <- lm(y ~ pbs(x, knots=c(pi/2, pi, pi*3/2),
          Boundary.knots = c(0, 2*pi)))
)
plot(x, y, xlab = "x", ylab = "sin(x)", pch="x", cex=.5)

lines(x, predict(fm2, data.frame(x=x, z=z)), col='blue')
lines(x, sin(x) + cos(2*x), col='red')

#### example of two periodic functions
x0 = seq(1,628, by=4)/100
z0 = seq(1, 314, by=3)/100
x = rep(x0, each=length(z0))
z = rep(z0, length(x0))
y = sin(x) + cos(2*z) +
   rnorm(length(x), 0, 0.1) ## x has a period of 2*pi and z of pi

summary( fm3 <- lm(y ~ pbs(x, df = 5, Boundary.knots = c(0, 2*pi))+
                       pbs(z, df = 5, Boundary.knots = c(0, pi)))
)

plot(sin(x) + cos(2*3), predict(fm3, data.frame(x=x, z=3)))
summary(sin(x) + cos(2*3)- predict(fm3, data.frame(x=x, z=3)))
## End(Not run)
```

`pooled` *Get pooled effect estimates from metaFluMoDL object*

Description

This function returns the pooled effect estimates for all incidence proxy terms (three for influenza, and optionally for RSV) from a `metaFluMoDL` object. It returns a `summary.FluMoDL` object that can

be further used in analyses.

Usage

```
pooled(m)
```

Arguments

m	An object of class metaFluMoDL, holding the results of a random-effects multivariate meta-analysis of summary.FluMoDL first-stage model summaries

Value

An object of class summary.FluMoDL, holding the pooled coefficients and variance-covariance matrices for the three influenza incidence proxies (four if hasRSV(m) is TRUE). The returned object contains the string "pooled" in its $type element, to distinguish it from first-stage model summaries or BLUP summaries (Best Linear Unbiased Predictor). The returned summary.FluMoDL object also has no $pred element, as it is not associated with a particular dataset and cross-basis matrices (which is a prerequisite to create crosspred objects).

References

- Gasparrini A, Armstrong B, Kenward MG. Multivariate meta-analysis for non-linear and other multi-parameter associations. Stat Med 2012;31(29):3821–39.

See Also

summary.FluMoDL, metaFluMoDL

---

predict.FluMoDL *Predict method for FluMoDL objects*

Description

Obtains predictions (predicted daily or weekly deaths) and optionally estimates standard errors of those predictions

Usage

```
## S3 method for class 'FluMoDL'
predict(object, temp = NULL, proxyH1 = NULL,
  proxyH3 = NULL, proxyB = NULL, proxyRSV = NULL, se.fit = FALSE,
  byWeek = FALSE, ...)
```

Arguments

object	A FluMoDL object
temp	A vector of daily mean temperatures. See 'Details'.
proxyH1	A vector of daily influenza A(H1N1)pdm09 incidence proxies. See 'Details'.
proxyH3	A vector of daily influenza A(H3N2) incidence proxies. See 'Details'.
proxyB	A vector of daily influenza B incidence proxies. See 'Details'.
proxyRSV	An vector of daily RSV incidence proxies (used only if the FluMoDL object includes an RSV term). See 'Details'.
se.fit	Logical switch indicating if standard errors are required. Requires byWeek=FALSE.
byWeek	If TRUE, aggregate fitted estimates by week. Has priority over argument se.fit. If both se.fit and byWeek are TRUE, se.fit is set to FALSE and a warning is returned.
...	Further arguments passed to or from other methods

Details

Arguments temp, proxyH1, proxyH3, proxyB and (if hasRSV(object) is TRUE) proxyRSV take a numeric vector as input, which is recycled to a length of nrow(object$data). Alternatively they can take NULL, in which case the respective column of object$data is used as input. Argument temp can also take the string "MMP", which is interpreted as the "Minimum Mortality Point", i.e. the temperature at which mortality is lowest (found in object$MMP).

In this way, the predict() method can be flexibly used to calculate the predicted "baseline" mortality (by setting temp="MMP" and all incidence proxies to zero), the model-predicted mortality for the actual input (by leaving all input arguments to their default NULL), or predicted mortalities for any combination of temperature and incidence proxy inputs.

Value

A vector of daily predicted deaths (corresponding to the rows in object$data). If byWeek=TRUE, the predictions are automatically aggregated by week (as per object$data$yearweek) and the vector contains the respective week (in YYYYWW format) as names.

If se.fit=TRUE, a list is returned with elements $fit and $se.fit containing the (daily) predicted deaths and their associated log standard errors.

Note that the first 30 elements (or first 5 elements if byWeek=TRUE) will be NA by default, as FluMoDL uses a maximum lag of 30 days.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
m
```

```
# Calculate FluMoDL baseline
baseline <- predict(m, temp="MMP", proxyH1=0, proxyH3=0, proxyB=0, byWeek=TRUE)

# Calculate fitted predictions
fitted <- predict(m, byWeek=TRUE)


# Plot everything
plot(with(m$data, tapply(deaths, yearweek, sum)), type="l",
     xaxt="n", ylab="Weekly deaths", xlab="Time")
points(baseline, type="l", col="blue")
points(fitted, type="l", col="green")
legend("topleft", c("Actual", "Baseline", "Fitted"), lty="solid",
    col=c("black", "blue", "green"), bty="n")
```

`summary.FluMoDL` *Summary method for FluMoDL objects*

Description

This function creates a summarized version of a 'FluMoDL' object. It contains the sets of coefficients and variance-covariance matrices for the incidence proxy terms (for influenza, and for RSV if provided), and the predictions for these terms.

Usage

```
## S3 method for class 'FluMoDL'
summary(object, ...)
```

Arguments

`object`	An object of class 'FluMoDL'
`...`	Further arguments passed to or from other methods.

Details

These summaries can be used to run a `multivariate meta-analysis` and calculate pooled effect estimates and BLUP (Best Unbiased Linear Predictor) estimates for influenza (and RSV if provided).

Value

An object of class 'summary.FluMoDL'. This is a list containing the following elements:

$type A string describing the meaning of the coefficients. Defaults to "summary", meaning a first-stage model summary. Alternatively, "blup" means Best Unbiased Linear Predictor (BLUP) coefficients, and "pooled" refers to coefficients pooled in the course of a multivariate meta-analysis. See `metaFluMoDL`.

$description A string with an additional description. For objects created with `summary.FluMoDL()` it is an empty string, but see `metaFluMoDL`.

$coef A list of numeric vectors, with names 'proxyH1', 'proxyH3' and 'proxyB' (and 'proxyRSV' if provided in the function arguments), containing the model coefficients for these terms.

$vcov A list of variance-covariance matrices, with names 'proxyH1', 'proxyH3' and 'proxyB' (and 'proxyRSV' if provided in the function arguments), for the respective model coefficients.

$pred A list with names 'proxyH1', 'proxyH3' and 'proxyB' (and 'proxyRSV' if provided in the function arguments), containing predictions (in the form of `crosspred` objects) for each exposure. These can be plotted in both the exposure-response and lag-response dimensions, see `crosspred`, `plot.crosspred` and the example below.

Examples

```
data(greece) # Use example surveillance data from Greece
m <- with(greece, fitFluMoDL(deaths = daily$deaths,
    temp = daily$temp, dates = daily$date,
    proxyH1 = weekly$ILI * weekly$ppH1,
    proxyH3 = weekly$ILI * weekly$ppH3,
    proxyB = weekly$ILI * weekly$ppB,
    yearweek = weekly$yearweek))
summ <- summary(m)
summ

# Plot the association between A(H1N1)pdm09 activity and mortality:
plot(summ$pred$proxyH1, "overall")
```

Index