

# Package ‘tsModel’

April 19, 2009

**Title** Time Series Modeling for Air Pollution and Health

**Depends** R (>= 2.7.0)

**Imports** splines, stats, tlnise, MASS

**Suggests** gam, NMMAPSLite

**LazyLoad** yes

**Date** 2008-08-12

**Version** 0.5-1

**Author** Roger D. Peng <rpeng@jhsph.edu>, with contributions from Aidan McDermott

**Maintainer** Roger D. Peng <rpeng@jhsph.edu>

**Description** Tools for specifying time series regression models

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2008-08-12 14:36:18

## R topics documented:

balt . . . . .	2
fitCitySeason . . . . .	3
ModelTerms . . . . .	4
seasonal . . . . .	5
spatialgibbs . . . . .	6
tsdecomp . . . . .	7
<b>Index</b>	<b>9</b>

---

`balt`*Baltimore City data*

---

**Description**

Mortality, air pollution, and weather data for Baltimore City, Maryland, USA, 1987–2000.

**Usage**

```
data(balt)
```

**Format**

A data frame with 15342 observations on the following 20 variables.

**cvd** daily counts of deaths from cardiovascular disease

**death** daily counts of deaths from all causes excluding accident

**resp** daily counts of deaths from respiratory disease

**tmpd** daily average temperature (Fahrenheit)

**rmtmpd** daily running mean of temperature for lags 1–3

**dptp** daily average dew point temperature

**rmdptp** daily running mean of dew point temperature for lags 1–3

**time** day/time indicator

**date** date

**agecat** a factor with levels `under65` `65to74` `75p`

**dow** a factor with levels `Sunday` `Monday` `Tuesday` `Wednesday` `Thursday` `Friday` `Saturday`

**pm10tmean** daily detrended PM10

**11pm10tmean** lag 1 PM10

**12pm10tmean** lag 2 PM10

**13pm10tmean** lag 3 PM10

**14pm10tmean** lag 4 PM10

**15pm10tmean** lag 5 PM10

**16pm10tmean** lag 6 PM10

**17pm10tmean** lag 7 PM10

**Age2Ind** indicator for age category 2 (65 to 74)

**Age3Ind** indicator for age category 3 (75 and above)

**Source**

See <http://www.ihapss.jhsph.edu/>.

**Examples**

```
## None
```

---

fitCitySeason	<i>Fit seasonally varying coefficient model</i>
---------------	---

---

## Description

Fit a seasonally varying coefficient model to NMMAPS data

## Usage

```
fitCitySeason(data, pollutant = "11pm10tmean", cause = "death",
              season = c("none", "periodic", "factor2"),
              tempModel = c("default", "rm7", "tempInt", "SeasonInt"),
              dfyr.Time = 7, pdfyr.time = 0.15, df.Temp = 6, df.Dew = 3,
              df.Season = 1, obsThreshold = 50, extractors = NULL)
```

## Arguments

data	data frame for an NMMAPS city
pollutant	name of pollutant variable
cause	name of cause of death; choices are "death" (all non-accidental), "cvd" (cardio-vascular), or "resp" (respiratory)
season	type of seasonal model to fit
tempModel	type of temperature model to use
dfyr.Time	number of degrees of freedom per calendar year of data to use in the smooth function of time
pdfyr.time	fraction of the degrees of freedom used in the overall smooth function of time to use in the smooth function of time specific to the older age category
df.Temp	degrees of freedom to use in the smooth function of temperature
df.Dew	degrees of freedom to use in the smooth function of dew point temperature
df.Season	number of sine/cosine pairs to include in the "periodic" model
obsThreshold	minimum number of observations required before a model can be fit to the data
extractors	a list of functions which extract elements of the "glm" object.

## Details

This function fits a seasonally varying coefficient model to data from the NMMAPS study using two different seasonal models. The first is a step function model ("factor2") which uses indicators for the seasons. The second is a hamonic model ("periodic") which uses sines and cosines.

## Value

An object of class "glm" is returned.

## References

Peng RD, Dominici F, Pastor-Barriuso R, Zeger SL, Samet JM (2005). "Seasonal Analyses of Air Pollution and Mortality in 100 US Cities," *American Journal of Epidemiology*, 161 (6), 585–595.

---

ModelTerms

*Model terms for time series models*

---

## Description

Tools for creating model/formula terms in time series models

## Usage

```
Lag(v, k, group = NULL)
runMean(v, lags = 0, group = NULL, filter = NULL)
harmonic(x, nfreq, period, intercept = FALSE)
```

## Arguments

<code>v, x</code>	a numeric vector
<code>k, lags</code>	an integer vector giving lag numbers
<code>group</code>	a factor or a list of factors defining groups of observations
<code>filter</code>	a vector specifying a linear filter
<code>nfreq</code>	number of sine/cosine pairs to include
<code>period</code>	period
<code>intercept</code>	should basis matrix include a column of 1s?

## Value

`Lag` returns a `length(v)` by `length(k)` matrix of lagged variables. `runMean` returns a numeric vector of length `length(v)`. `harmonic` returns a matrix of sine/cosine basis functions.

## Author(s)

Roger D. Peng

## Examples

```
## Ten day "time series"
x <- rnorm(10)

## Lag 1 of `x`
Lag(x, 1)

## Lag 0, 1, and 2 of `x`
Lag(x, 0:2)
```

```
## Running mean of lag 0, 1, and 2
runMean(x, 0:2)
```

---

seasonal

*Reproducible Seasonal Analysis of Air Pollution and Mortality*


---

## Description

Functions for conducting a reproducible seasonal analysis of air pollution and mortality in the United States.

## Usage

```
coefSeasonal(results, pollutant, method = "factor2",
             Seasons = c("Winter", "Spring", "Summer", "Fall"))
multiDFFit(dfVec, city, ...)
poolCoef(b, cov = NULL, w = NULL, method = c("tlnise", "fixed"),
        extractors = NULL, ...)
extractBetaCov(results, pollutant)
LouisFormat(x, type = c("stderr", "confint"), digits = 2)
gibbspool(b, v, scale = 1, maxiter = 2000, burn = 500,
          pScale = 1e-5)
pooling(estimate, var)
```

## Arguments

results	list of individual city-specific regression fits
pollutant	name of pollutant of interest
method	type of seasonal model, or method of pooling
Seasons	character vector of season names
dfVec	vector of degrees of freedom per year to use in the smooth function of time
city	name of a city in the NMMAPS database
b	a vector of N coefficients or an N x p matrix of coefficient vectors
cov, v	a vector of N variances or a p x p x N array of covariances matrices
w	second-stage covariates to include in the hierarchical model
extractors	a list of functions
x	a numeric matrix
type	type of output to produce
digits	number of digits of output to use
scale	prior standard deviation of the national average
maxiter	maximum number of iterations in the Gibbs sampler

burn	number of iterations to discard
pScale	scaling coefficient for the prior variance of the heterogeneity parameter
estimate	a vector of coefficients
var	a vector of coefficient variances
...	other arguments passed to <code>fitCitySeason</code>

### Details

These functions are used to reproduce a seasonal analysis of air pollution and mortality in the United States (see the reference below).

### Author(s)

Roger D. Peng (rpeng@jhsph.edu)

### References

Peng RD, Dominic F (2008). *Statistical Methods for Environmental Epidemiology in R: A Case Study in Air Pollution and Health*, Springer.

---

spatialgibbs      *~~function to do ... ~~*

---

### Description

~~ A concise (1-5 lines) description of what the function does. ~~

### Usage

```
spatialgibbs(b, v, x, y, phi = 0.1, scale = 1, maxiter = 1000, burn = 500, a0 = 10,
```

### Arguments

b	a vector of regression coefficients
v	a vector of regression coefficient variances
x	a vector of x-coordinates
y	a vector of y-coordinates
phi	scale parameter for exponential covariance function
scale	scaling parameter for the prior variance of the national average estimate
maxiter	maximum number of iterations in the Gibbs sampler
burn	number of iterations to discard
a0	parameter for Gamma prior on heterogeneity variance
b0	parameter for Gamma prior on heterogeneity variance

**Details**

This function is used to produce pooled national average estimates of air pollution risks taking into account potential spatial correlation between the risks. The function uses a Gibbs sampler to produce the posterior distribution of the national average estimate and the heterogeneity variance. See the reference below for more details.

**Author(s)**

Roger D. Peng (rpeng@jhsph.edu)

**References**

Peng RD, Dominic F (2008). *Statistical Methods for Environmental Epidemiology in R: A Case Study in Air Pollution and Health*, Springer.

---

tsdecomp

*Time scale decomposition*

---

**Description**

Decompose a vector into frequency components

**Usage**

```
tsdecomp(x, breaks)
```

**Arguments**

<code>x</code>	a numeric vector with no missing data
<code>breaks</code>	a numeric constant or a vector of break points into which <code>x</code> should be broken. If <code>breaks</code> is a constant then <code>x</code> will be broken into that number of frequencies. This argument is passed directly to <code>cut</code> to determine the break points. See <code>cut</code> for more details.

**Value**

A matrix with dimension  $n \times m$  where  $n$  is the length of `x` and  $m$  is the number of break categories.

**Author(s)**

Original by Aidan McDermott; revised by Roger Peng (rpeng@jhsph.edu)

**References**

Dominici FD, McDermott A, Zeger SL, Samet JM (2003). "Airborne particulate matter and mortality: Timescale effects in four US cities", *American Journal of Epidemiology*, 157 (12), 1055–1065.

**Examples**

```
x <- rnorm(101)
freq.x <- tsdecomp(x, c(1, 10, 30, 80))

## decompose x into 3 frequency categories.
## x[,1] represents from 1 to 9 cycles in 101 data points
## x[,2] represents from 10 to 29 cycles in 101 data points
## x[,3] represents from 30 to 50 cycles in 101 data points
## you can only have up to 50 cycles in 101 data points.
```

# Index

## \*Topic **datasets**

balt, [2](#)

## \*Topic **models**

fitCitySeason, [3](#)

ModelTerms, [4](#)

seasonal, [5](#)

spatialgibbs, [6](#)

tsdecomp, [7](#)

balt, [2](#)

coefSeasonal (*seasonal*), [5](#)

extractBetaCov (*seasonal*), [5](#)

fitCitySeason, [3](#)

gibbspool (*seasonal*), [5](#)

harmonic (*ModelTerms*), [4](#)

Lag (*ModelTerms*), [4](#)

LouisFormat (*seasonal*), [5](#)

ModelTerms, [4](#)

multiDFFit (*seasonal*), [5](#)

poolCoef (*seasonal*), [5](#)

pooling (*seasonal*), [5](#)

runMean (*ModelTerms*), [4](#)

seasonal, [5](#)

spatialgibbs, [6](#)

tsdecomp, [7](#)