Package: poptrend (via r-universe)

September 17, 2024

Title Estimate Smooth and Linear Trends from Population Count Survey Data

Version 0.2.0

Description Functions to estimate and plot smooth or linear population trends, or population indices, from animal or plant count survey data.

Depends R (>= 3.1.2)

License GPL-3

LazyData true

Imports mgcv (>= 1.7.0), stats, graphics

RoxygenNote 7.2.3

URL https://github.com/jknape/poptrend

BugReports https://github.com/jknape/poptrend/issues

Repository https://jknape.r-universe.dev

RemoteUrl https://github.com/jknape/poptrend

RemoteRef HEAD

RemoteSha 72235c5c0170c68cb083eb78c982156e97ea5f47

Contents

`change` *Compute the change in the population over a time interval.*

Description

Computes the estimated percentual change in the population between two given time points, and an approximate confidence interval for the change.

Usage

```
change(trend, start, end, alpha = 0.05)
```

Arguments

`trend`	A fitted object of class trend.
`start`	Start time for the comparison.
`end`	End time for the comparison.
`alpha`	alpha-level for approximate confidence interval.

Details

The function computes the estimated change between two chosen time points. When random effects are present, the change is computed for the underlying linear or smooth trend term. For index models, the change is estimated from the difference between indices. Changes can only be computed between time points that were included in the `trendGrid` argument to ptrend, if the two time points are not included the nearest points in the grid are chosen.

Confidence intervals are computed using quantiles of the bootstrapped trends.

Value

A list containing the estimated change, and start and end points.

Note

If `start` or `end` are not contained in the `trendgrid` argument of the `ptrend` function, the change is computed between the values in the grid that are closest to these points.

Author(s)

Jonas Knape

Examples

```
## Simulate a data set with 10 sites and 30 years
data = simTrend(30, 10)
## Fit a smooth trend with fixed site effects, random time effects,
## and automatic selection of degrees of freedom
trFit = ptrend(count ~ trend(year, type = "smooth") + site, data = data)
## Check the estimated percent change from year 2 to 20
change(trFit, 10, 20)
```

`checkFit` *Check goodness of fit of a trend model.*

Description

Produces various goodness of fit plots and diagnostic measures.

Usage

```
checkFit(trend, residuals = TRUE, ...)
```

Arguments

`trend`	A fitted object of class trend.
`residuals`	Should residuals be plotted?
`...`	Further arguments passed to `plot.gam`.

Details

The function simply calls plot.gam and gam.check on the underlying gam model for checking goodness of fit.

Author(s)

Jonas Knape

See Also

`plot.gam`, `gam.check`

`goldcrest` *Data for goldcrest from the Swedish Bird Survey.*

Description

Transect count survey data for goldcrest from the Swedish Bird Survey from 1998 to 2012.

Usage

```
goldcrest
```

Format

A data frame with 5728 rows and 9 columns.

- count Total transect count.
- site Site identity.
- year Year of the survey.
- observerAge Age of the observer.
- day Day of season.
- firstSurvey A binary variable which is 1 if the route was surveyed for the first time by the observer in that year, and 0 otherwise.
- latitude Latitude of the transect.
- lineCov The percentage of the transect that was surveyed.

Source

`http://www.fageltaxering.lu.se/english`

`greenfinch` *Data for greenfinch from the Swedish Bird Survey.*

Description

Transect count survey data for greenfinch from the Swedish Bird Survey from 1998 to 2012.

Usage

```
greenfinch
```

Format

A data frame with 5728 rows and 9 columns.

- count Total transect count.
- site Site identity.
- year Year of the survey.
- observerAge Age of the observer.
- day Day of season.
- firstSurvey A binary variable which is 1 if the route was surveyed for the first time by the observer in that year, and 0 otherwise.
- latitude Latitude of the transect.
- lineCov The percentage of the transect that was surveyed.

Source

`http://www.fageltaxering.lu.se/english`

`hessBootstrap` *Compute bootstrap confidence intervals based on sampling from the variance-covariance matrix.*

Description

Draws bootstrap samples using the estimated variance matrix of the fitted gam model.

Usage

```
hessBootstrap(trend, nBoot = 500)
```

Arguments

`trend`	An object of class trend.
`nBoot`	The number of bootstrap samples to draw.

Details

This function is used by ptrend and would typically not be called directly. Bootstrap samples are drawn using the estimated coefficients and covariance matrix vcov.gam of the fitted gam model. The default values of vcov.gam which gives the Bayesian posterior covariance matrix.

Bootstrapped samples computed in this way do not account for any uncertainty in the selection of degrees of freedom.

Value

A trend object with the bootstrapped trend estimates appended.

Author(s)

Jonas Knape

plot.trend *Plot an estimated trend.*

Description

The function plots an estimated trend or index, as well as estimates of any temporal random effects included in the trend term.

Usage

```
## S3 method for class 'trend'
plot(
  x,
  baseline = NULL,
  alpha = 0.05,
  ylab = "abundance index",
  trendCol = "black",
  lineCol = adjustcolor("black", alpha.f = 0.05),
  shadeCol = adjustcolor("#0072B2", alpha.f = 0.4),
  incCol = "#009E73",
  decCol = "#D55E00",
  plotGrid = TRUE,
  plotLines = FALSE,
  ranef = "pointCI",
  secDeriv = TRUE,
  ...
)
```

Arguments

`x` A fitted object of class trend.

`baseline` A time point or function used to compute the baseline of the trend (see Knape 2023). If the argument is numeric, the point in the *trendGrid* argument of the function `ptrend` closest to this value will be taken as the baseline (i.e. the estimated trend will be 1 at this point). If the argument is a function, the function is applied to trends and the resulting value is used as the baseline. By default, the first time point is taken as the reference.

`alpha` The alpha level of confidence intervals.

`ylab` The label of the y-axis.

`trendCol` The color of the trend line.

`lineCol` The color of bootstrapped trend lines, if plotted.

`shadeCol` The color of the confidence region.

`incCol` The color of regions where the first or second derivative is significantly increasing.

`decCol` The color of regions where the first or second derivative is significantly decreasing.

`plotGrid` If true, grid lines are plotted.

`plotLines` If true, the bootstrapped trends are plotted.

`ranef` String indicating whether to plot point estimates and/or confidence intervals for random effects. One of 'pointCI', 'point', 'CI' or 'no'.

`secDeriv` If true, coloured boxes at the bottom of the plot shows segments where the second derivative of the smooth is significantly different from zero.

`...` Further arguments passed to `plot.default`.

Details

Trends and indexes are relative measures and therefore are compared against some reference value. By default, the first observed time point is used as the reference value.

If the estimated trend contains bootstrap samples, confidence intervals are plotted as well. For smooth trend models, time periods where the trend is significantly declining or increasing are marked with a different color (set by arguments *decCol* and *incCol*). Periods where the second derivative is significantly positive or negative are marked by coloured boxes at the bottom of the plot.

There is an additional option of plotting each of the bootstrapped trends.

Author(s)

Jonas Knape

References

Knape, J. (2023). Effects of choice of baseline on the uncertainty of population and biodiversity indices. Environmental and Ecological Statistics, 30, 1–16. <doi:10.1007/s10651-022-00550-7>

`poptrend` *Analyze population trends from survey count data.*

Description

The package provides functions for fitting and analysing trend models of data obtained from population count surveys.

Details

The package provides functions for estimating smooth trends with generalized additive mixed models, as well as linear trends and population indices. It is intended as a simple interface to basic trend estimation, allowing estimation of trends accounting for effects of covariates in the form of both smooth terms and random effects. The model fitting engine is the function `gam` of package `mgcv`. Background for the package is given in Knape (2016).

References

Knape, J. 2016. Decomposing trends in Swedish bird populations using generalized additive mixed models. Journal of Applied Ecology, 53:1852-1861. DOI:10.1111/1365-2664.12720.

`print.trend` *Print a trend object.*

Description

Prints basic information about a trend object.

Usage

```
## S3 method for class 'trend'
print(x, ...)
```

Arguments

`x`	A trend object.
`...`	Not used.

Details

Prints the family, formula and type of trend.

Author(s)

Jonas Knape

`ptrend` *Fit a smooth or linear trend to count survey data.*

Description

The function estimates a trend from count survey data.

Usage

```
ptrend(
  formula,
  data = list(),
  family = quasipoisson(),
  nGrid = 500,
  nBoot = 500,
  bootType = "hessian",
  gamModel = TRUE,
  engine = "gam",
  ...
)
```

Arguments

`formula` A trend formula. This is a GAM formula with an extra term `trend` describing the time variable and properties of the trend. All terms except the trend term are treated as covariates. Effect of temporal variation in these covariates are not included in the calculation of the trend.

`data` A data frame containing response variables and covariates.

`family` The distributional family of the response. The family most use a log-link, it defaults to a quasi-Poisson.

`nGrid` The number of grid points over which to compute the trend. If the length of the argument is one, an equally spaced grid over the survey period of length nGrid is set up. nGrid can also be a vector of length 3, in which case the first element is the number of grid points and the second and third elements give, respectively, the start and endpoints of the grid.

`nBoot` The number of bootstrap samples to draw.

`bootType` Only one method, "hessian", currently implemented. Type "hessian", draws bootstrap samples using the Bayesian covariance matrix of the parameters (see `vcov.gam`).

`gamModel` If true, the fit of the underlying gam model is saved in the output. May be set to FALSE to save memory, but with the side effect that the fit of the gam model cannot be checked.

`engine` If 'gam', the default, model fitting is done via `gam`. If 'bam', model fitting is done via `bam`, which is less memory hungry and can be faster for large data sets.

`...` Further arguments passed to `gam`.

Details

The function estimates smooth or loglinear population trends, or indexes from simple design count survey data. It is essentially a wrapper around a call to `gam`, processing its output using `predict.gam` to produce a trend estimate. For smooth trends, cubic regression splines for the temporal variable are set up by the term `s(var, k = k, fx = fx , bs = "cr")` where `var` is the first argument to `trend` in the formula. For loglinear trends, the identity of `var` is used, and for index models a factor variable is constructed from `var`.

Temporal random effects are set up by converting the temporal variable supplied to `trend` to a factor variable and adding this factor variable to the data supplied to `gam`.

Bootstrap confidence intervals are computed by drawing normally distributed random variable with means equal to the estimated coefficients and covariance matrix equal to the Bayesian posterior covariance matrix (see vcov.gam).

Value

An object of class trend.

Author(s)

Jonas Knape

References

Knape, J. (2016). Decomposing trends in Swedish bird populations using generalized additive mixed models. Journal of Applied Ecology, 53, 1852–1861. <doi:10.1111/1365-2664.12720>

Examples

```
## Simulate a data set with 15 sites and 25 years
data = simTrend(15, 25)
## Fit a smooth trend with fixed site effects, random time effects,
## and automatic selection of degrees of freedom
trFit = ptrend(count ~ trend(year, tempRE = TRUE, type = "smooth") + site, data = data)
## Check the model fit
checkFit(trFit)
## Plot the trend
plot(trFit)
summary(trFit)
## Check the estimated percent change from year 8 to 25
change(trFit, 8, 25)

## Fit a loglinear trend model with random site effects and random time effects
## to the same data set.
trLin = ptrend(count ~ trend(year, tempRE = TRUE, type = "loglinear") +
                 s(site, bs = "re"), data = data)
plot(trLin)
summary(trLin)

## Fit an index model with fixed site effects and an (unrelated) continous covariate
## as a smooth effect.
# Simulate mock covariate unrelated to data.
cov = rnorm(nrow(data))
trInd = ptrend(count ~ trend(year, type = "index") + site + s(cov), data = data)
plot(trInd)
summary(trInd)
```

`simTrend` *Simulate population survey data.*

Description

Simulates count survey data with a non-linear trend, and site and temporal random effects. The logistic function is used to create a trend the reduces the expected population size to half its initial value over the time period.

Usage

```
simTrend(nyear = 30, nsite = 40, mu = 3, timeSD = 0.1, siteSD = 0.3)
```

Arguments

`nyear`	The number of years in the simulated survey.
`nsite`	The number of sites in the simulated survey
`mu`	The expected mean of the counts at the start of the survey.
`timeSD`	Standard deviation (at log-scale) of annual mean deviation from the trend.
`siteSD`	Standard deviation (at log-scale) of simulated among site variation.

Value

A data frame containing simulated data.

Author(s)

Jonas Knape

`summary.trend` *Summary of trend estimates*

Description

Computes a trend or index estimate for each time point in the survey.

Usage

```
## S3 method for class 'trend'
summary(object, baseline = NULL, alpha = 0.05, ...)
```

Arguments

object — A trend object returned by ptrend.

baseline — A time point or function used to compute the baseline of the trend. If the argument is numeric, the point in the trendGrid argument of the function ptrend closest to this value will be taken as the baseline (i.e. the estimated trend will be 1 at this point). If the argument is a function, the function is applied to trends and the resulting value is used as the baseline. By default, the first time point is taken as the reference.

alpha — alpha level for approximate confidence intervals.

... — Not used.

Details

For a smooth or loglinear trend model the function computes an estimate of the trend value for each time point in the survey. By default, the reference value is the first time point. Note that if the trend model was fitted with random effects, the random effects are not included in the estimate. Thus the estimate refers to the long-term component.

For an index trend model the index at each time point is computed.

If bootstrap samples are available, bootstrap confidence intervals for the trend or index values are also computed.

Author(s)

Jonas Knape

trend — *Define a trend component.*

Description

The function is used to set up the trend component used in ptrend formulas.

Usage

```
trend(var, tempRE = FALSE, type = "smooth", by = NA, k = -1, fx = FALSE)
```

Arguments

var — A numeric time variable over which a trend or index will be computed.

tempRE — If TRUE, this will set up random time effects. The random effects will be constructed by converting the var argument to a factor. Note that this yields a random effect level for each unique value in var. If this is not appropriate, an alternative is to set tempRE to false and manually add temporal random effects in the trend formula (using s(..., bs = "re")). Temporal random effects cannot be used with index estimation.

`type` The type of trend to be estimated. One of "smooth", "loglinear" or "index".

`by` Currently ignored.

`k` The dimension of the basis for the cubic regression spline of smooth trend fits.

`fx` If true, automatic selection of degrees of freedom are used for smooth trends.

Details

The function extracts information about the trend component of a formula supplied to ptrend. It returns a list containing variable names, information, and `s` components as strings used in subsequent calls to gam.

Value

A list containing information to set up the trend.

Author(s)

Jonas Knape

Examples

```
## Simulate a data set with 15 sites and 25 years
data = simTrend(15, 25)
## Fit a smooth trend with fixed site effects, but no random time effects,
## and fixed degrees of freedom
trFit = ptrend(count ~ trend(year, tempRE = FALSE, k =  8, fx = FALSE, type = ”smooth”) +
                 site, data = data)
plot(trFit)
```

Index