Package ‘apc’

October 12, 2022

Type Package

Title Age-Period-Cohort Analysis

Version 2.0.0

Date 2020-09-28

Author Zoe Fannon, Bent Nielsen

Maintainer Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk>

Description Functions for age-period-cohort analysis. Aggregate data can be organised in matrices indexed by age-cohort, age-period or cohort-period. The data can include dose and response or just doses. The statistical model is a generalized linear model (GLM) allowing for 3,2,1 or 0 of the age-period-cohort factors. Individual-level data should have a row for each individual and columns for each of age, period, and cohort. The statistical model for repeated cross-section is a generalized linear model. The statistical model for panel data is ordinary least squares. The canonical parametrisation of Kuang, Nielsen and Nielsen (2008) <DOI:10.1093/biomet/asn026> is used. Thus, the analysis does not rely on ad hoc identification.

Imports lattice, plyr, reshape, plm, survey, lmtest, car, ISLR, AER, ggplot2, ChainLadder

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-01 23:20:06 UTC

R topics documented:

`apc-package` *Age-period-cohort analysis*

Description

The package includes functions for age-period-cohort analysis. The statistical model is a generalized linear model (GLM) allowing for age, period and cohort factors, or a sub-set of the factors. The canonical parametrisation of Kuang, Nielsen and Nielsen (2008a) is used. The outline of an analysis is described below.

Details

Package:	apc
Type:	Package
Version:	2.0.0
Date:	2020-09-28
License:	GPL-3

The apc package uses the canonical parameters suggested by Kuang, Nielsen and Nielsen (2008a) and generalized by Nielsen (2014). These evolve around the second differences of age, period and cohort factors as well as an three parameters (level and two slopes) for a linear plane. The age, period and cohort factors themselves are not identifiable. They could be ad hoc identified by associating the levels and two slopes to the age, period and cohort factors in a particular way. This should be done with great care as such ad hoc identification easily masks which information is coming from the data and which information is coming from the choice of ad hoc identification scheme. An illustration is given below. A short description of the package can be found in Nielsen (2015).

A formal analysis of the identification of the age-period-cohort model can be found in Nielsen and Nielsen (2014). Forecasting is discussed in Kuang, Nielsen and Nielsen (2008b, 2011) and Martinez Miranda, Nielsen and Nielsen (2015). Methods for cross section data are introduced in Fannon, Monden and Nielsen (2019). Methods for panel data are introduced in Fannon (2020). For a recent overview see Fannon and Nielsen (2019).

The package covers age-period-cohort models for three types of data.

1. Tables of aggregate data.
2. Repeated cross sectional data.
3. Panel data.

The apc package can be used as follows.

1. **Aggregate data**. For a vignette with an introduction to analysis of aggregate data, see see `IntroductionAggregateData.pdf`, `IntroductionAggregateData.R` on `Vignettes`.
   (a) Organize the data in as an `apc.data.list`. Data are included in matrix format. Information needs to be given about the original data format. Optionally, information can be given about the labels for the time scales.
   (b) Construct descriptive plots using `apc.plot.data.all`. This gives a series of descriptive plots. The plots can be called individually through
      i. Plot data sums using `apc.plot.data.sums`. Numerical values can be obtained through `apc.data.sums`.
      ii. Sparsity plots of data using `apc.plot.data.sparsity`.
      iii. Plot data using all combinations of two time scales using `apc.plot.data.within`.
   (c) Get an deviance table for the age-period-cohort model through `apc.fit.table`.
   (d) Estimate a particular (sub-model of) age-period-cohort model through `apc.fit.model`.
   (e) Plot probability transforms of observed responses given fit using `apc.plot.fit.pt`.
   (f) Plot estimated parameters through `apc.plot.fit`. Numerical values of certain transformations of the canonical parameter can be obtained through `apc.identify`.

   (g) Recursive analysis can be done by selecting a subset of the observations through `apc.data.list.subset` and then repeating analysis. This will reveal how sensitive the results are to particular age, period and cohort groups.
   (h) Forecasting. Some functions have been been added for forecasting in from a Poisson response-only model with an age-cohort parametrization `apc.forecast.ac` and with an age-period parametrization `apc.forecast.ap`. See also the overview on `apc.forecast`

2. **Repeated cross section** and **Panel Data**. For a vignette with an introduction to analysis of repeated cross section data and panel data, see `IntroductionIndividualData.pdf`, `IntroductionIndividualData.R` on `Vignettes` Further examples can be found in a second vignette, see `IntroductionIndividualDataFurtherExampl` `IntroductionIndividualDataFurtherExample.R`.

Data examples include

1. **Aggregate data**
   (a) `data.asbestos` includes counts of deaths from mesothelioma in the UK. This dataset has no measure for exposure. It can be analysed using a Poisson model with an "APC" or an "AC" design. Source: Martinez Miranda, Nielsen and Nielsen (2015). Also used in Nielsen (2015).
   (b) `data.Italian.bladder.cancer` includes counts of deaths from bladder cancer in the Italy. This dataset includes a measure for exposure. It can be analysed using a Poisson model with an "APC" or an "AC" design. Source: Clayton and Schifflers (1987a).
   (c) `data.Belgian.lung.cancer` includes counts of deaths from lung cancer in the Belgium. This dataset includes a measure for exposure. It can be analysed using a Poisson model with an "APC", "AC", "AP" or "Ad" design. Source: Clayton and Schifflers (1987a).
   (d) `data.Japanese.breast.cancer` includes counts of deaths from breast cancer in the Japan. This dataset includes a measure for exposure. It can be analysed using a Poisson model with an "APC" design. Source: Clayton and Schifflers (1987b).

   Repeated cross section data
   (a) `Wage` data from the package ISLR

   Repeated cross section data
   (a) `PSID7682` data from the package AER. These are panel data on earnings for 595 individuals for the years 1976-1982.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 29 Jan 2015 updated 26 Aug 2020.

References

Clayton, D. and Schifflers, E. (1987a) Models for temperoral variation in cancer rates. I: age-period and age-cohort models. *Statistics in Medicine* 6, 449-467.

Clayton, D. and Schifflers, E. (1987b) Models for temperoral variation in cancer rates. II: age-period-cohort models. *Statistics in Medicine* 6, 469-481.

Fannon, Z. (2020). D.Phil. thesis. University of Oxford.

Fannon, Z., Monden, C. and Nielsen, B. (2018) Age-period cohort modelling and covariates, with an application to obesity in England 2001-2014. *Download*: Nuffield DP. Supplement Code for replication: Nuffield DP supplement.

Fannon, Z. and Nielsen, B. (2019) Age-period-cohort models. *Oxford Research Encyclopedia of Economics and Finance*. Oxford University Press. *Download*: `doi.org/10.1093/acrefore/9780190625979.013.495`; Earlier version Nuffield DP.

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008b) Forecasting with the age-period-cohort model and the extended chain-ladder model. Biometrika 95, 987-991. *Download*: Article; Earlier version Nuffield DP.

Kuang, D., Nielsen, B. and Nielsen, J.P. (2011) Forecasting in an extended chain-ladder-type model. Journal of Risk and Insurance 78, 345-359. *Download*: Article; Earlier version: Nuffield DP.

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

Nielsen, B. (2015) apc: An R package for age-period-cohort analysis. *R Journal* 7, 52-64. *Download*: Open access.

Nielsen, B. (2014) Deviance analysis of age-period-cohort models. *Download*: Nuffield DP.

Nielsen, B. and Nielsen, J.P. (2014) Identification and forecasting in mortality models. The Scientific World Journal. vol. 2014, Article ID 347043, 24 pages. *Download*: Article.

See Also

Vignettes are available on `Vignettes`.

Further information, including minor upgrades and a python version can be found on `apc development web page`.

Examples

```
# see vignettes
```

`apc-internal` *Internal apc Functions*

Description

Internal apc functions

Details

These are not to be called by the user.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 1 Feb 2016 `apc.data.list` *Arrange data as an apc.data.list*

Description

This is step 1 of the apc analysis.

The apc package is aimed at range of data types. This analysis and labelling of parameters depends on the choice data type. In order to keep track of this choice the data first has to be arranged as an apc.data.list. The function purpose of this function is to aid the user in constructing a list with the right information.

Age period cohort analysis is used in two situations. A dose-response situation, where both doses (exposure, risk set, cases) and responses (counts of deaths, outcomes) are available. And a response situation where only a response is available. If the aim is to directly model mortality ratios (counts of death divided by exposure) this will be thought of a response

The `apc.data.list` gives sufficient information for the further analysis. It is sufficient to store this information. It has 2 obligatory arguments, which are a response matrix and a character indicating the data format. It also has some further optional arguments, which have certain default values. Some times it may be convenient to add further arguments to the `apc.data.list`. This will not affect the apc analysis.

`apc.data.list` generates default row and column names for the response and dose matrices when these are not provided by the user.

Usage

```
apc.data.list(response, data.format, dose=NULL,
age1=NULL, per1=NULL, coh1=NULL, unit=NULL,
per.zero=NULL, per.max=NULL,
time.adjust=NULL, label=NULL,
n.decimal=NULL)
```

Arguments

`response` matrix (or vector). Numbers of responses. It should have a format matching `data.format`. Time should be increasing with the row/column index of the matrix. For instance, consider a 10x5 matrix in "AP" format: Then the row index is for age, and it should be increasing in age. Thus, higher ages are further down the rows of the matrix. In the same way, the column index is for period.

`data.format` character. The following options are implemented:

"AC" has age/cohort as increasing row/column index.

"AP" has age/period as increasing row/column index.

"CA" has cohort/age as increasing row/column index.

"CL" has cohort/age as increasing row/column index, triangular.

"CP" has cohort/period as increasing row/column index.

"PA" has period/age as increasing row/column index.

"PC" has period/cohort as increasing row/column index.

"trapezoid" has age/period as increasing row/column index, period-diagonals are NA for period <= per.zero and >per.zero+per.max.

`dose` *Optional.* matrix or NULL. Numbers of doses. It should have same format as `response`.

`age1` *Optional.* Numeric or NULL. Time label for youngest age group. Used if `data.format` is "AC", "AP", "CA", "CL", "PA", "trapezoid". If NULL default is unit.

`per1` *Optional.* Numeric or NULL. Time label for oldest period group. Used if `data.format` is "AP", "CP", "PA", "PC". If NULL default is unit.

`coh1` *Optional.* Numeric or NULL. Time label for youngest age group. Used if `data.format` is "AC", "CA", "CL", "CL.vector.by.row", "CP", "PC", "trapezoid". If NULL default is unit.

`unit` *Optional.* Numeric or NULL. Common time steps for age, period and cohort. For quarterly data use `1/4`. For monthly data use `1/12`. If NULL default is 1.

`per.zero` *Optional.* Numeric or NULL. Needed if data format is "trapezoid".

`per.max` *Optional.* Numeric or NULL. Needed if data format is "trapezoid".

`time.adjust` *Optional.* Numeric. Time labels are based on two of age1, per1 and coh1. The third time label is computed according to the formula age1+coh1=per1+time.adjust. Default is 0. If age1=coh=1 it is natural to choose time.adjust=1.

`label` *Optional.* Character. Useful when working with multiple data sets. Some internal functions use the first three characters of the label for identification of the two datasets.

`n.decimal` *Optional.* Numeric or NULL. The labels for parameters involves a date. This is found by converting a number into a character. If the value is set to `d` package uses `sprintf`. If the value is set to `NULL` and `unit==1/4` for quarterly data or `unit==1/12` for monthly data or `1/20<=unit && unit<1` then package uses `sprintf`. If the value is set to `NULL` and `1/20>unit || unit>=1` then package uses `as.character`, which looks nice for integers, but can be messy otherwise.

Details

If the user does not set values for any of `age1`, `per1`, `coh1`, `unit` then the value is set to `unit`.

The user can set values of `age1`, `per1`, `coh1` that are incongruent. The functions only use two these that are relevant for the chosen `data.format`. Example: the `data.format` may be `"AC"` and the user sets `age1`, `per1`, but `age1`, `coh1` are relevant for this data format. The `apc.data.list` then sets `coh1=unit`, by default, while ignoring the value for `per1`. Other commands such as `apc.data.list.subset` or `apc.fit.table`, will internally, as default option, call the function `apc.get.index`. That function will, in this example, set `per1` according to the values of `age1` and `coh1`.

If the user does not set a value for `time.adjust` this is set equal to `unit` when the user does not specify at least two `age1`, `per1`, `coh1`. Otherwise it is set to 0. The former choice matches the values in the theory papers, where indices count 1,2,... to follow standard notation for row/column indices for matrices, so that age+coh=per+unit. The latter choice seeks to match a real time scale the user sets according to age+coh=per.

Value

`response` matrix (or vector). Numbers of responses.

`dose` matrix (or NULL). Numbers of doses.

`data.format` character.

`age1` Numeric. Default is NULL.

`per1` Numeric. Default is NULL.

`coh1` Numeric. Default is NULL.

`unit` Numeric. Default is NULL. For monthly data one use `unit=1/12`.

`per.zero` Numeric. If data.format is not "trapezoid" the value is NULL. If data.format is "trapezoid" the coordinate system is in age-cohort format and this value counts how many periods are cut off. The default is `per.zero=0`.

`per.max` Numeric. If data.format is not "trapezoid" the value is NULL. If data.format is "trapezoid" the coordinate system is in age-cohort format and this value counts how many periods are included in the data array. The default is `per.max=nrow(response)+ncol(respon`

`time.adjust` Numeric. Default is NULL.

`label` Character. Default of NULL.

`n.decimal` Numeric or NULL.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 17 Nov 2016

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014) Deviance analysis of age-period-cohort models. *Download*: Nuffield DP.

Nielsen, B. (2015) apc: An R package for age-period-cohort analysis. *R Journal* 7, 52-64. *Download*: Open access.

See Also

The below example shows how the `data.Japanese.breast.cancer` data.list was generated. Other provided data sets include `data.asbestos` `data.Belgian.lung.cancer` `data.Italian.bladder.cancer`.

A subset of the data can be selected using `apc.data.list.subset`.

Examples

```
###############
# Artificial data
# (1) Generate a 5x7 matrix and make arbitrary decisions for rest

response <- matrix(data=seq(1:35),nrow=5,ncol=7)
data.list <- apc.data.list(response=response,data.format="AP",
age1=25,per1=1955,coh1=NULL,unit=5,
per.zero=NULL,per.max=NULL)
data.list

# (2) Chain Ladder data

k <- 5
v.response  <- seq(1:(k*(k+1)/2))
data.list <- apc.data.list(response=vector.2.triangle(v.response,k),
data.format=”CL.vector.by.row”,age1=2001)
data.list

###############
# Japanese breast cancer
# This is the code used to generate the data.Japanese.breast.cancer
v.rates <- c( 0.44, 0.38, 0.46, 0.55, 0.68,
    1.69, 1.69, 1.75, 2.31, 2.52,
  4.01, 3.90, 4.11, 4.44, 4.80,
  6.59, 6.57, 6.81, 7.79, 8.27,
  8.51, 9.61, 9.96,11.68,12.51,
 10.49,10.80,12.36,14.59,16.56,
 11.36,11.51,12.98,14.97,17.79,
 12.03,10.67,12.67,14.46,16.42,
 12.55,12.03,12.10,13.81,16.46,
 15.81,13.87,12.65,14.00,15.60,
 17.97,15.62,15.83,15.71,16.52)
v.cases <- c(   88,   78,  101,  127,  179,
   299,  330,  363,  509,  588,
   596,  680,  798,  923, 1056,
   874,  962, 1171, 1497, 1716,
  1022, 1247, 1429, 1987, 2398,
  1035, 1258, 1560, 2079, 2794,
   970, 1087, 1446, 1828, 2465,
   820,  861, 1126, 1549, 1962,
   678,  738,  878, 1140, 1683,
   640,  628,  656,  900, 1162,
   497,  463,  536,  644,  865)
# see also example below for generating labels

rates <- matrix(data=v.rates,nrow=11, ncol=5,byrow=TRUE)
cases <- matrix(data=v.cases,nrow=11, ncol=5,byrow=TRUE)

#  A data list is now constructed as follows
# note that list entry rates is redundant,
# but included since it represents original data

data.Japanese.breast.cancer <- apc.data.list(response=cases,
dose=cases/rates,data.format=”AP”,
age1=25,per1=1955,coh1=NULL,unit=5,
per.zero=NULL,per.max=NULL,time.adjust=0,
label=”Japanese breast cancer”)

# or when exploiting the default values
```

```
data.Japanese.breast.cancer <- apc.data.list(response=cases,
dose=cases/rates,data.format="AP",
age1=25,per1=1955,unit=5,
label="Japanese breast cancer")

####################################################
#  Code for generating labels

row.names <- paste(as.character(seq(25,75,by=5)),"-",as.character(seq(29,79,by=5)),sep="")
col.names <- paste(as.character(seq(1955,1975,by=5)),"-",as.character(seq(1959,1979,by=5)),sep="")
```

apc.data.list.subset *Cut age, period and cohort groups from data set.*

Description

For a recursive analysis it is useful to be able to cut age, period and cohort groups from a data set. Function returns an apc.data.list with data.format "trapezoid".

When used with default values the function turns an apc.data.list into a new apc.data.list with data.format "trapezoid" without reducing dataset.

Usage

```
apc.data.list.subset(apc.data.list,
age.cut.lower=0,age.cut.upper=0,
per.cut.lower=0,per.cut.upper=0,
coh.cut.lower=0,coh.cut.upper=0,
apc.index=NULL,
suppress.warning=FALSE)
```

Arguments

apc.data.list	List. See apc.data.list for a description of the format.
age.cut.lower	*Optional*. Numeric. Specifies how many age groups to cut at lower end. Default is zero.
per.cut.lower	*Optional*. Numeric. Specifies how many period groups to cut at lower end. Default is zero.
coh.cut.lower	*Optional*. Numeric. Specifies how many cohort groups to cut at lower end. Default is zero.
age.cut.upper	*Optional*. Numeric. Specifies how many age groups to cut at upper end. Default is zero.
per.cut.upper	*Optional*. Numeric. Specifies how many period groups to cut at upper end. Default is zero.

`coh.cut.upper` *Optional*. Numeric. Specifies how many cohort groups to cut at upper end. Default is zero.

`apc.index` *Optional*. List. See `apc.get.index` for a description of the format. If not provided this is computed internally.

`suppress.warning`
Optional. Logical. Suppresses warnings. This is useful when generating data sums using `apc.data.sums` but reducing the data set so much that models cannot be fitted.

Value

`response` matrix (or vector). Numbers of responses.

`dose` matrix (or NULL). Numbers of doses.

`data.format` "trapezoid"

`age1` Numeric.

`per1` Numeric.

`coh1` Numeric.

`unit` Numeric.

`per.zero` Numeric.

`per.max` Numeric.

Arguments: Notes

If apc.index is supplied then the input can be simplified. It suffices to write `apc.data.list = list(response=response,data.format=data.format,dose=dose)`, where dose could be `dose=NULL`. Likewise `apc.index` does not need to be a full `apc.index` `list`. It suffices to construct a list with entries `age.max`, `per.max`, `coh.max`, `age1`, `per1`, `coh1`, `unit`, `per.zero`, `index.trap`, `index.data`.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 4 Dec 2013 recoded 26 Apr 2017

See Also

The below example uses artificial data. For an example using `data.asbestos` see `apc.plot.fit`.

Examples

```
###############
# Artificial data
# Generate a 5x7 matrix and make arbitrary decisions for rest

response <- matrix(data=seq(1:35),nrow=5,ncol=7)
data.list <- list(response=response,dose=NULL,data.format="AP",
age1=25,per1=1955,coh1=NULL,unit=5,
per.zero=NULL,per.max=NULL,time.adjust=0)
data.list
```

```
apc.data.list.subset(data.list,1,1,0,0,0,0)
```

apc.data.sums *Computes age, period and cohort sums of a matrix*

Description

Computes age, period and cohort sums of a matrix. This is the same as taking column, row and diagonal sums. The match between the age, period and cohort sums and column, row and diagonal sums depends on the data format

Usage

```
apc.data.sums(apc.data.list,data.type="r",
average=FALSE,keep.incomplete=TRUE,apc.index=NULL)
```

Arguments

`apc.data.list` List. See `apc.data.list` for a description of the format.

`data.type` Optional. Character. "r","d","m" if sums are computed for responses,dose,(mortality) rates. Rates are computed as responses/doses. "r" is default.

`average` Optional. Logical. If TRUE/FALSE reports averages/sums. Default is FALSE.

`keep.incomplete` Optional. Logical. If true perform calculation for incomplete sequences by removing NA. If false incomplete sequences are NA. See example. Default=TRUE.

`apc.index` Optional. List. See `apc.get.index` for a description of the format. If not provided this is computed.

Value

`sums.age` Vector. Sums/Averages over data.matrix by age.

`sums.per` Vector. Sums/Averages over data.matrix by period.

`sums.coh` Vector. Sums/Averages over data.matrix by cohort.

Arguments: Notes

If apc.index is supplied then the input can be simplified. For instance if `data.type="r"` then, for the first argument, it suffices to write `apc.data.list` = `list(response=response)`. Likewise `apc.index` does not need to be a full `apc.index list`. It suffices to construct a list with entries `age.max`, `per.max`, `coh.max`, `index.trap`, `index.data`, `per.zero`.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 15 Aug 2018 (15 Dec 2013)

See Also

The example below uses Japanese breast cancer data, see data.Japanese.breast.cancer

Examples

```
#####################
#  EXAMPLE with artificial data
#  generate a 3x4 matrix in "AP" data.format with the numbers 1..12

m.data   <- matrix(data=seq(length.out=12),nrow=3,ncol=4)
m.data
data.list <- apc.data.list(m.data,"AP")
apc.data.sums(data.list)

# $sums.age
#  [1] 22 26 30
# $sums.per
# [1]  6 15 24 33
# $sums.coh
# [1]  3  8 15 24 18 10

apc.data.sums(data.list,average=TRUE)
# $sums.age
# [1] 5.5 6.5 7.5
# $sums.per
# [1]  2  5  8 11
# $sums.coh
# [1]  3  4  5  8  9 10

apc.data.sums(data.list,keep.incomplete=FALSE)
# $sums.age
#  [1] 22 26 30
# $sums.per
# [1]  6 15 24 33
# $sums.coh
# [1]  NA NA 15 24 NA NA

#####################
# EXAMPLE with Japanese breast cancer data

data.list <- data.Japanese.breast.cancer() # function gives data list
apc.data.sums(data.list)

# $sums.age
# [1]  573 2089 4053 6220 8083 8726 7796 6318 5117 3986 3005
# $sums.per
# [1]  7519  8332 10064 13183 16868
# $sums.coh
# [1]  497 1103 1842 2858 4474 5550 6958 7471 7531 6931 5111 3080 1666  715  179

# Compare with the response matrix
```

```
data.list$response

#       1955-1959 1960-1964 1965-1969 1970-1974 1975-1979
# 25-29        88        78       101       127       179
# 30-34       299       330       363       509       588
# 35-39       596       680       798       923      1056
# 40-44       874       962      1171      1497      1716
# 45-49      1022      1247      1429      1987      2398
# 50-54      1035      1258      1560      2079      2794
# 55-59       970      1087      1446      1828      2465
# 60-64       820       861      1126      1549      1962
# 65-69       678       738       878      1140      1683
# 70-74       640       628       656       900      1162
# 75-79       497       463       536       644       865
```

`apc.fit.model` *Fits an age period cohort model*

Description

`apc.fit.model` fits the age period cohort model as a Generalized Linear Model using `glm.fit`. The model is parametrised in terms of the canonical parameter introduced by Kuang, Nielsen and Nielsen (2008), see also the implementation in Martinez Miranda, Nielsen and Nielsen (2015). This parametrisation has a number of advantages: it is freely varying, it is the canonical parameter of a regular exponential family, and it is invariant to extentions of the data matrix. Other parametrizations can be computed using `apc.identify`.

`apc.fit.model` can be be used for all three age period cohort factors, or for submodels with fewer of these factors.

`apc.fit.model` can be used either for mortality rates through a dose-response model or for mortality counts through a pure response model without doses/exposures.

The GLM families include Poisson regressions (with log link) and Normal/Gaussian least squares regressions.

apc.fit.table produces a deviance table for 15 combinations of the three factors and linear trends: "APC", "AP", "AC", "PC", "Ad", "Pd", "Cd", "A", "P", "C", "t", "tA", "tP", "tC", "1".

Usage

```
apc.fit.model(apc.data.list,model.family,model.design,apc.index=NULL,
replicate.version.1.3.1=FALSE)
apc.fit.table(apc.data.list,model.family,model.design.reference="APC",
apc.index=NULL)
```

Arguments

`apc.data.list` List. See `apc.data.list` for a description of the format.

`model.family` Character. The following options are implemented. These are used internally when calling `glm.fit`.

"poisson.response" This sets family=poisson(link="log"). Only responses are used. Inference is done in a multinomial model, conditioning on the overall level as documented in Martinez Miranda, Nielsen and Nielsen (2015).

"od.poisson.response" This sets family=quasipoisson(link="log") in the estimation step, but then reverts to family=poisson(link="log") when computing standard errors, which are then corrected. Only responses are used. Inference is done in an over-dispersed Poisson model as documented in Harnau and Nielsen (2016). Note that limit distributions are t and F not normal and chi2.

"poisson.dose.response" This sets family=poisson(link="log"). Doses are used as offset.

"binomial.dose.response" This sets family=binomial(link="logit") and gives a logistic regression.

"gaussian.rates" This sets family=gaussian(link="identity"). The dependent variable is the mortality rates, which are computed as response/dose.

"gaussian.response" This sets family=gaussian(link="identity"). Only responses are used. The dependent variable is the responses.

"log.normal.rates" Gaussian regression for log(rates) and with identity link (Least Squares).

"log.normal.response" Gaussian regression for log(response) and with identity link (Least Squares).

`model.design` Character. This indicates the design choice. The following options are possible.

"APC" Age-period-cohort model.

"AP" Age-period model. Nested in "APC"

"AC" Age-cohort model. Nested in "APC"

"PC" Period-cohort model. Nested in "APC"

"Ad" Age-trend model, including age effect and two linear trends. Nested in "AP", "AC".

"Pd" Period-trend model, including period effect and two linear trends. Nested in "AP", "PC".

"Cd" Cohort-trend model, including cohort effect and two linear trends. Nested in "AC", "PC".

"A" Age model. Nested in "Ad".

"P" Period model. Nested in "Pd".

"C" Cohort model. Nested in "Cd".

"t" Trend model, with two linear trends. Nested in "Ad", "Pd", "Cd".

"tA" Single trend model in age index. Nested in "A", "t".

"tP" Single trend model in period index. Nested in "P", "t".

"tC" Single trend model in cohort index. Nested in "C", "t".

"1" Constant model. Nested in "tA", "tP", "tC".

`model.design.reference`

Character. This indicates the reference design choice for the deviance table. Choices are "APC","AP","AC","PC","Ad","Pd","Cd","A","P","C","t". Default is "APC".

apc.index *Optional*. List. See apc.get.index for a description of the format. If not provided this is computed internally. If apc.fit.model is used in a simulation study computational effort can be saved when using this option.

replicate.version.1.3.1

Optional. Logical. Replicate error in covariance calculation for "poisson.response","od.poisson.response" in versions 1.2.3-1.3.1. Default=FALSE

Value

apc.fit.table produces a deviance table. There are 15 rows corresponding to all possible design choices. The columns are as follows.

"-2logL" -2 log Likelihood up to some constant. If the model family is Poisson or binomial (logistic) this is the same as the glm deviance: That is the difference in -2 log likelihood value between estimated model and the saturated model. If the model family is Gaussian it is different from the traditional glm deviance. Here the -2 log likelihood value is measured in a model with unknown variance, which is the standard in regression analysis, whereas in the glm package the deviance is the residual sum of squares, which can be interpreted as the -2 log likelihood value in a model with variance set to one.

"df.residual" Degrees of freedom of residual: nrow x ncol - dim(parameter). If the model.family="poisson.response" the degrees of freedom is one lower.

"prob(>chi_sq)"

p-value of the deviance, -2logL. Left out in Gaussian case which has no saturated model

"LR vs APC" the likelihood ratio statistic against the "APC" model.

"df" Degrees of freedom against the "APC" model.

"prob(>chi_sq)"

p-value of log likelihood ratio statistic.

"aic" Akaike's "An Information Criterion", minus twice the maximized log-likelihood plus twice the number of parameters upto a constant. It is take directly from the glm function. For the "poisson.dose.response" and "binomial.dose.response" model families the dispersion is fixed at one and the number of parameters is the number of coefficients. The "poisson.response" model is conditional on the level. The number of parameters should therefore be adjusted by subtracting 2 to take this into account to get the proper AIC. However, in practice this does not matter, since we are only interested in relative effects. For the "gaussian.response" and "gaussian.dose.response" model families the dispersion is estimated from the residual deviance.

"F" Only for "od.poisson.response". F statistic: Ratio of deviance for submodel divided by degrees of freedom to deviance of apc model divided by degrees of freedom.

"prof(>F)" Only for "od.poisson.response". F statistic: with degrees of freedom given by differences between sub-model and apc model and between apc model and saturated model.

apc.fit.model returns a list. The entries are as follows.

`fit` List. Values from `glm.fit`.

`apc.index` List. Values from `apc.get.index`.

`coefficients.canonical`
Matrix. For each coordinate of the canonical parameters is reported coefficient, standard deviation, z-value, which is the ratio of those, and asymptotically normal p-values. Note, for "od.poisson.response" the reported standard errors corrected by the deviance and p-values are asymptotically t distributed, see Harnau and Nielsen (2016). Other parametrizations can be computed using `apc.identify`.

`covariance.canonical`
Matrix. Estimated covariance matrix for canonical parameters.

`slopes` Vector. Length three. The design matrix found by `apc.get.design.collinear` has age, period, and cohort linear trends. `slopes` indicates which of these are actually used in estimation.

`difdif` Vector. Length three. The design matrix found by `apc.get.design.collinear` has age, period, and cohort double differences. `slopes` indicates which of these are actually used in estimation.

`index.age` Vector. Indices for age double difference parameters within `coefficients.canonical`. NULL if age double differences are not estimated.

`index.per` Vector. Indices for period double difference parameters within `coefficients.canonical`. NULL if period double differences are not estimated.

`index.coh` Vector. Indices for cohort double difference parameters within `coefficients.canonical`. NULL if cohort double differences are not estimated.

`dates` Vector. Indicates the dates for the double difference parameters within `coefficients.canonical`.

`model.family` Character. Argument.

`model.design` Character. Argument.

`RSS` Numeric. Residual sum of squares. NULL for non-gaussian families.

`sigma2` Numeric. Maximum likelihood estimator for variance: RSS/n. NULL for non-gaussian families.

`s2` Numeric. Least squares estimator for variance: RSS/df. NULL for non-gaussian families.

`n.decimal` Numeric. From `apc.data.list`.

`predictors` Vector. Design*Estimates. Same as the `glm.fit` value `linear.predictors` when there is no offset.

Note

For gaussian families *deviance* is defined differently in apc and `glm`. Here it is -2 log likelihood. In `glm` it is RSS.

The values for `apc.fit.model` include the `apc.data.list` and the `apc.index` returned by `apc.get.index`.

For the `poisson.response` the inference is conditional on the level, see Martinez Miranda, Nielsen and Nielsen (2015). The `coefficients.canonical` computed by `apc` are therefore different from the default `coefficients` computed by `glm`.

For the `od.poisson.response` an asymptotic theory is used that mimics the conditioning for `poisson.response`. The asymptotic distribution are, however, asymptotically t or F distributed, see Harnau and Nielsen (2017).

For the `log.normal.response` standard normal theory applies for quantities on the log scale including estimators. An asymptotic theory for quantities on the original scale is provided in Kuang and Nielsen (2018).

For `coefficients` the 3rd and 4th columns have headings `t value` and `Pr(>|t|)` for `od.poisson.response` to indicate an asymptotic t theory and otherwise `z value` and `Pr(>|z|)` to indicate an asymptotic normal theory. The labels are inherited from `glm.fit`.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 15 Aug 2018 (27 Aug 2014)

References

Harnau, J. and Nielsen (2016) Over-dispersed age-period-cohort models. To appear in *Journal of the American Statistical Association*. *Download*: Nuffield DP

Kuang, D, Nielsen B (2018) Generalized log-normal chain-ladder. mimeo Nuffield Collge.

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

See Also

The fit is done using `glm.fit`.

The examples below use Italian bladder cancer data, see `data.Italian.bladder.cancer` and Belgian lung cancer data, see `data.Belgian.lung.cancer`.

In example 3 the design matrix is called is called using `apc.get.design`.

Examples

```
#####################
# EXAMPLE 1 with Italian bladder cancer data

data.list <- data.Italian.bladder.cancer() # function gives data list
apc.fit.table(data.list,"poisson.dose.response")

#        -2logL df.residual prob(>chi_sq) LR.vs.APC df.vs.APC prob(>chi_sq)        aic
# APC    33.179          27         0.191        NA        NA            NA    487.624
# AP    512.514          40         0.000   479.335        13         0.000    940.958
# AC     39.390          30         0.117     6.211         3         0.102    487.835
# PC   1146.649          36         0.000  1113.470         9         0.000   1583.094
# Ad    518.543          43         0.000   485.364        16         0.000    940.988
# Pd   4041.373          49         0.000  4008.194        22         0.000   4451.818
# Cd   1155.629          39         0.000  1122.450        12         0.000  1586.074
# A    2223.800          44         0.000  2190.621        17         0.000  2644.245
# P   84323.944          50         0.000 84290.765        23         0.000 84732.389
# C   23794.205          40         0.000 23761.026        13         0.000 24222.650
# t    4052.906          52         0.000  4019.727        25         0.000  4457.351
# tA   5825.158          53         0.000  5791.979        26         0.000  6227.602
# tP  84325.758          53         0.000 84292.579        26         0.000 84728.203
# tC  33446.796          53         0.000 33413.617        26         0.000 33849.241
# 1   87313.678          54         0.000 87280.499        27         0.000 87714.123
#
# Table suggests that "APC" and "AC" fit equally well.  Try both

fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")
fit.ac <- apc.fit.model(data.list,"poisson.dose.response","AC")

# Compare the estimates: They are very similar

fit.apc$coefficients.canonical
fit.ac$coefficients.canonical

#####################
# EXAMPLE 2 with Belgian lung cancer data
# This example illustrates how to find the linear predictors

data.list <- data.Belgian.lung.cancer()

# Get an APC fit

fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")

# The linear predictor of the fit is a vector.
# But, we would like it in the same format as the data.
# Thus create matrix of same dimension as response data
# This can be done in two ways

m.lp <- data.list$response # using original information
m.lp <- fit.apc$response #  using information copied when fitting

# the fit object index.data is used to fill linear predictor in
# vector format into matrix format

m.lp[fit.apc$index.data] <-fit.apc$linear.predictors
exp(m.lp)

#####################
# EXAMPLE 3 with Belgian lung cancer data
# This example illustrates how apc.fit.model works.

data.list <- data.Belgian.lung.cancer()

# Vectorise data
index <- apc.get.index(data.list)
v.response <- data.list$response[index$index.data]
```

```
v.dose <- data.list$dose[index$index.data]

# Get design
m.design <- apc.get.design(index,"APC")$design

# Fit using glm.fit from stats package
fit.apc.glm <- glm.fit(m.design,v.response,family=poisson(link="log"),offset=log(v.dose))

# Get canonical coefficients
v.cc <- fit.apc.glm$coefficients

# Find linear predictors and express in matrix form
m.fit <- data.list$response # create matrix
m.fit[index$index.data] <- m.design
m.fit.offset <- m.fit + log(data.list$dose) # add offset
exp(m.fit.offset)

# Compare with linear.predictors from glm.fit
# difference should be zero
sum(abs(m.fit.offset[index$index.data]-fit.apc.glm$linear.predictors))

#####################
# EXAMPLE 4 with Taylor-Ashe loss data
# This example illustrates the over-dispersed poisson response model.

data <- data.loss.TA()
fit.apc.od <- apc.fit.model(data,"od.poisson.response","APC")
fit.apc.od$coefficients.canonical[1:5,]
fit.apc.no.od <- apc.fit.model(data,"poisson.response","APC")
fit.apc.no.od$coefficients.canonical[1:5,]
```

apc.forecast *Forecasts from age-period-cohort models.*

Description

In general forecasts from age-period-cohort models require extrapolation of the estimated parameters. This has to be done without introducing identifications problems, see Kuang, Nielsen and Nielsen (2008b,2011). There are many different possibilities for extrapolation for the different sub-models. The extrapolation results in point forecasts. Distribution forecasts should be build on top of these, see Martinez Miranda, Nielsen and Nielsen (2015) and Harnau and Nielsen (2016). At present three experimental functions apc.forecast.ac, apc.forecast.apc and apc.forecast.ap are available.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 10 Sep 2016 (1 Feb 2016)

References

Harnau, J. and Nielsen (2016) Over-dispersed age-period-cohort models. To appear in *Journal of the American Statistical Association*. *Download*: Nuffield DP

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008b) Forecasting with the age-period-cohort model and the extended chain-ladder model. Biometrika 95, 987-991. *Download*: Article; Earlier version Nuffield DP.

Kuang, D., Nielsen B. and Nielsen J.P. (2011) Forecasting in an extended chain-ladder-type model. Journal of Risk and Insurance 78, 345-359. *Download*: Article; Earlier version: Nuffield DP.

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

`apc.forecast.ac` *Forecast for responses model with AC or CL structure.*

Description

Computes forecasts for a model with AC or Chain Ladder structure. Forecasts of the linear predictor are given for all models. Distributions forecasts are provided for a Poisson response model (using Martinez Miranda, Nielsen and Nielsen, 2015), for an over-dispersed Poisson response model (using Harnau and Nielsen, 2017) and for a log normal response model (using Kuang and Nielsen, 2018) This is done for the triangle which shares age and cohort indices with the data.

Usage

```
apc.forecast.ac(apc.fit,sum.per.by.age=NULL,
sum.per.by.coh=NULL, quantiles=NULL, suppress.warning=TRUE)
```

Arguments

`apc.fit` List. Output from `apc.fit.model`. Note: apc.fit.model should be run with AC structure so that `apc.fit$model.design=="AC"`. Distribution forecasts are only provided for a Poisson response model where `apc.fit$model.family=="poisson.response"` for an over-disperse Poisson response model where `apc.fit$model.family=="poisson.response"` and for a log normal response model where `apc.fit$model.family=="log.normal.response"`. For other models only point forecasts of the linear predictor are provided, that is the first two values `linear.predictors.forecast` and `index.trap.J`.

`sum.per.by.age` *Optional*. Vector. If not NULL it will generate forecasts by period, where, for each period, the point forecasts are cummulated over certain age groups. Indicates which age groups. If `sum.per.by.age` is a scalar or vector of length one it represents a single age group. Point forecasts are made for the indicated age group. If `sum.per.by.age` is a vector of length two it represents lower and upper values of an range of age groups. Point forecasts are cummulated over the indicated age groups.

`sum.per.by.coh` *Optional*. Vector. Same as `sum.per.by.age`, but for cohort instead of age.

quantiles *Optional*. Vector. Generates forecast quantiles for indicated quantiles. Example: quantiles=c(0.05,0.50,0.95). Default is NULL.

suppress.warning Logical. If true, suppresses warnings from apc.data.list.subset, which is called internally. Default is "TRUE".

Details

The default output only reports standard errors. By setting the argument quantiles to, for instance, quantiles=c(0.05,0.50,0.95) forecast quantiles are reported.

Poisson response forecast errors. The asymptotic theory for the Poisson forecast standard errors is presented in Martinez Miranda, Nielsen and Nielsen (2015). The sampling theory is based on multinomial model, conditional on the total number of outcomes. Asymptotically this gives a normal theory. There are two independent contributions to the forecast error: a process error and an estimation error. The empirical example of that paper uses the data data.asbestos. The results of that paper are reproduced in the vignette ReproducingMMNN2015.pdf, ReproducingMMNN2015.R on Vignettes.

Overdispersed Poisson response forecast errors. The asymptotic theory for the overdispersed Poission forecast standard errors is presented in Harnau and Nielsen (2018). The sampling theory is based on infinitely devisible distributions, with the compound Poisson distribution as a special case. This results in scale nuisance parameter, which is estimated by the deviance of the AC model divided by the degrees of freedom. Asymptotically this gives a t/F theory. There are three independent contributions to the forecast error: a process error, an estimation error and a sampling error for the overall mean.

Generalized log normal forecast errors. Uses the asymptotic theory present in Kuang and Nielsen (2018). The sampling theory is based on infinitely devisible distributions, using small sigma asymptotics. There are two independent contributions to the forecast error: a process error and an estimation error.

The examples below are based on the smaller data reserving sets data.loss.VNJ, data.loss.TA. See also data.loss.XL.

Value

linear.predictors.forecast
Vector. Linear predictors for forecast area.

index.trap.J Matrix. age-coh coordinates for vector. Similar structure to index.trap in apc.index, see apc.get.index.

trap.response.forecast
Matrix. Includes data and point forecasts. Forecasts in lower right triangle. Trapezoid format.

response.forecast.cell
Matrix. 4 columns. 1: Point forecasts. 2: corresponding forecast standard errors 3: process standard errors 4: estimation standard errors Note that the square of column 2 equals the sums of squares of columns 3 and 4 Note that index.trap.J gives the age-coh coordinates for each entry.

response.forecast.age
Same as response.forecast.cell, but point forecasts by age cumulated over period/cohort.

`response.forecast.per`
: Same as `response.forecast.cell`, but point forecasts by `per` cumulated over age/cohort.

`response.forecast.per.ic`
: Same as response.forecast.cell, but point forecasts cumulated by per and intercept corrected by multiplying column 1 of `response.forecast.per` by `intercept.correction.per`.

`response.forecast.coh`
: Same as `response.forecast.cell`, but point forecasts by `coh` cumulated over age/period.

`response.forecast.all`
: Same as `response.forecast.cell`, but point forecasts cumulated by `age` and `coh`.

`response.forecast.per.by.age`
: Only if `sum.per.by.age!=NULL`. Same as `response.forecast.per`, but point forecasts cumulated over ages indicated by `sum.per.by.age`.

`response.forecast.per.by.age.ic`
: Only if `sum.per.by.age!=NULL`. Same as `response.forecast.per.by.age`, but intercept corrected using `intercept.correction.per.by.age`.

`response.forecast.per.by.coh`
: Only if `sum.per.by.coh!=NULL`. Same as `response.forecast.per`, but point forecasts cumulated over cohorts indicated by `sum.per.by.coh`.

`response.forecast.per.by.coh.ic`
: Only if `sum.per.by.coh!=NULL`. Same as `response.forecast.per.by.coh`, but intercept corrected using `intercept.correction.per.by.coh`.

`intercept.correction.per`
: Numeric. The intercept correction is constructed as the ratio of the sum of data entries for the last period and the sum of the corresponding fitted values.

`intercept.correction.per.by.age`
: Numeric. Only if `sum.per.by.age!=NULL`.

`intercept.correction.per.by.coh`
: Numeric. Only if `sum.per.by.coh!=NULL`.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 18 November 2019 (2 Mar 2016)

References

Harnau, J. and Nielsen (2018) Over-dispersed age-period-cohort models. *Journal of the American Statistical Association* 113, 1722-1732. *Download*: Nuffield DP

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

Martinez Miranda, M.D., Nielsen, B., Nielsen, J.P. and Verrall, R. (2011) Cash flow simulation for a model of outstanding liabilities based on claim amounts and claim numbers. *ASTIN Bulletin* 41, 107-129.

Kuang, D, Nielsen B (2018) Generalized log-normal chain-ladder. mimeo Nuffield Collge.

See Also

The example below uses Japanese breast cancer data, see data.Japanese.breast.cancer

Examples

```
#####################
#  EXAMPLE with reserving data: data.loss.VNJ()
#  Data used in Martinez Miranda, Nielsen, Nielsen and Verrall (2011)
#  Point forecasts are the Chain-Ladder forecasts
#  *NOTE* Data are over-dispersed,
# so distribution forecast are *NOT* reliable
#  The same could be done data.asbestos(),
# which are not over-dispersed
# see vignette.

data <- data.loss.VNJ()
fit.ac <- apc.fit.model(data,"poisson.response","AC")
forecast <- apc.forecast.ac(fit.ac)

# forecasts by "policy-year"
forecast$response.forecast.coh
#            forecast         se    se.proc     se.est
# coh_2      1684.763   57.69067   41.04586   40.53949
# coh_3     29379.085  220.53214  171.40328  138.76362
# coh_4     60637.929  313.33867  246.24770  193.76066
# coh_5    101157.697  385.69930  318.05298  218.18857
# coh_6    173801.522  501.42184  416.89510  278.60786
# coh_7    249348.589  595.21937  499.34816  323.94060
# coh_8    475991.739  864.06580  689.92155  520.20955
# coh_9    763918.643 1182.70450  874.02440  796.78810
# coh_10 1459859.526 2216.80272 1208.24647 1858.58945

# forecasts of "cash-flow"
forecast$response.forecast.per
# reproduces Table 6 of MMNNV (2011)
#           forecast         se    se.proc   se.est
# per_11 1353858.32 1456.92459 1163.55417 876.7958
# per_12  754180.12 1017.37629  868.43544 529.9758
# per_13  488612.42  816.62860  699.00817 422.2202
# per_14  318043.00  664.36135  563.95302 351.1880
# per_15  184610.86  508.97704  429.66366 272.8494
# per_16  115022.56  414.64945  339.14976 238.5615
# per_17   63145.15  320.93564  251.28700 199.6360
# per_18   35812.79  255.08766  189.24267 171.0466
# per_19    2494.27   78.10439   49.94266  60.0502

# forecast  of "total reserve"
# reproduces Table 6 of MMNNV (2011)
forecast$response.forecast.all
#     forecast       se  se.proc   se.est
# all  3315779 3182.737 1820.928 2610.371
```

```
#####################
# Forecast of cashflows for 7th cohort (policy year)
# Note a series of warnings are given because
# this is done by truncating the data
# which generates the warnings associated
# with apc.data.list.subset()
forecast<- apc.forecast.ac(fit.ac,sum.per.by.coh=7)
forecast$response.forecast.per.by.coh
#          forecast        se   se.proc    se.est
# per_11 102975.337 355.97444 320.89771 154.08590
# per_12  58061.306 267.24671 240.95914 115.58329
# per_13  40466.866 226.40049 201.16378 103.87646
# per_14  21615.765 170.90637 147.02301  87.13910
# per_15  24410.927 194.70158 156.23997 116.17994
# per_16   1818.389  61.09857  42.64257  43.75668
#
# This can also be intercept corrected
# Such intercept corrections are useful when
# analysing data.asbestos().
# Unclear if they are useful for
# reserving.
forecast$intercept.correction.per.by.coh
#   > [1] 1.241798
forecast$response.forecast.per.by.coh.ic
#          forecast        se   se.proc    se.est
# per_11 127874.573 355.97444 320.89771 154.08590
# per_12  72100.417 267.24671 240.95914 115.58329
# per_13  50251.675 226.40049 201.16378 103.87646
# per_14  26842.415 170.90637 147.02301  87.13910
# per_15  30313.441 194.70158 156.23997 116.17994
# per_16   2258.071  61.09857  42.64257  43.75668

#####################
# Forecast of cashflows cumulated for
# 6th and 7th cohort (policy year)
forecast<- apc.forecast.ac(fit.ac,sum.per.by.coh=c(6,7))
forecast$response.forecast.per.by.coh.ic
#          forecast        se   se.proc    se.est
# per_11 226219.380 460.52781 414.62816 200.42295
# per_12 139628.153 366.48699 325.74697 167.93339
# per_13  87022.435 295.86605 257.16360 146.29970
# per_14  66584.160 277.64858 224.94656 162.75067
# per_15  34962.678 206.77289 163.00324 127.22018
# per_16   2392.759  61.09857  42.64257  43.75668

#####################
#  EXAMPLE with reserving data: data.loss.TA()
#  Data used in Harnau and Nielsen (2016)
data <- data.loss.TA()
fit.ac <- apc.fit.model(data,"od.poisson.response","AC")
forecast <- apc.forecast.ac(fit.ac,quantiles=c(0.01,0.05,0.5,0.95,0.99))
forecast$response.forecast.all
#     forecast      se se.proc  se.est  tau.est
# all 18680856 2675417 1007826 2474680 134561.2
# ...
#  t-0.010  t-0.050  t-0.500  t-0.950  t-0.990
# 12158931 14160544 18680856 23201167 25202781
# ...
#  G-0.010  G-0.050  G-0.500  G-0.950  G-0.990
# 12760202 14398564 18553290 23417098 25792423
forecast$response.forecast.per

#####################
#  EXAMPLE with reserving data: data.loss.XL()
#  see helpfile for data.loss.XL
```

---

apc.forecast.ap *Forecast for Poisson response model with AP structure.*

Description

Computes forecasts for a model with AP structure. The data can have any form allowed in, see `apc.data.list`. These are all special cases of generalised trapezoids. If the "lower triangle" with the largest (age,coh) values are not observed, they can be forecast using this function. The function extrapolates the AP model to the lower triangle where `per.zero+per.max` < `per` <= `age.max+coh.max-1`. The estimates of the age parameters can be used for the lower triangle. The estimates of the period parameters need to be extrapolated for the lower triangle. Thus, the function extrapolates `per.forecast.J=age.max+coh.max-1-per.zero-per.max` period values. The extrapolation method has to chosen so as not to introduce an identification problem, see Kuang, Nielsen and Nielsen (2008b,2011). Two such extrapolation methods are implemented in this function: "I0" and "I1". The default is to report the linear predictor.

If the `model.family="binominal.dose.response"`, that is a logistic model, then forecasts of dose, response and survival probability are given for lower triangle.

Usage

```
apc.forecast.ap(apc.fit,extrapolation.type="I0",suppress.warning=TRUE)
```

Arguments

`apc.fit` List. Output from `apc.fit.model`. Note: apc.fit.model should be run with AP structure so that `apc.fit$model.design=="AP"`. Only point forecasts of the linear predictor are provided.

`extrapolation.type`
Character. Choices for extrapolating the differenced period parameter ("Delta.beta_per"). Default is "I0".

"I0" extrapolates the first out-of-sample differenced period parameter by the average of cumulated sums of the in-sample estimated differenced period parameters. The subsequent out-of-sample differenced period parameters are zero.

"I1" extrapolates all out-of-sample differenced period parameters by zero.

Both methods are invariant to ad hoc identification of the implied period time effect, by following the ideas put forward in Kuang, Nielsen and Nielsen (2008b). Internally, the extrapolation is done as follows. The estimated differenced period parameters are found from "apc.fit$coefficients.canonical" using `apc.identify` with `type="dif"`. These imply period time effects by ad hoc identification: choose an arbitrary value for the first period time effect and add partial sums of the differenced period parameter. Fit a time series model: an intercept model with "I0" and a random walk model for "I1". Then extrapolate and take differences. These extrapolation methods are invariant to the actual choice of the arbitrary value for the first period time effect.

`suppress.warning`
: Logical. If true, suppresses warnings from `apc.data.list.subset`, which is called internally. Default is "TRUE".

Details

When `model.family=binomial.dose.response` forecasts are made by the component method, see Cox (1976). It is intended to be used for a population analysis situation where the response equals cohort-decrease of dose. For cell in forecast array with index `(age,cohort)` then: Survival probability is `survival=1/(1+exp(predictor_(a,c)))`. Dose is `dose_(a,c)=max(0,dose_(a-1,c)-response_(a-1,c)`
Response is `response_(a,c)=dose_(a,c)*(1-survival_(a,c))`.

Value

`trap.predictors.forecast`
: Matrix. Includes estimates and point forecasts of linear predictor. That is design*coefficient. Same as the `glm.fit` value `linear.predictors` when there is no offset. Forecasts in lower right triangle. Trapezoid format.

`index.trap.J`
: Matrix. age-coh coordinates for forecast area. Similar structure to `index.trap` in `apc.index`, see `apc.get.index`.

`D.xi.per.extrapolated`
: Matrix. Extrapolated parameters. Dimension `per.forecast.J=age.max+coh.max-1-per.zero-per.ma` rows, 1 column.

`trap.dose.forecast`
: Matrix. Includes data and point forecasts. Forecasts in lower right triangle. Dose in cell age,coh equal to dose in cell age-1,coh minus response in cell age-1,coh. Only implemented for `model.family="binomial.dose.response"`. See details.

`trap.response.forecast`
: Matrix. Includes data and point forecasts. Forecasts in lower right triangle. Response in cell age,coh equal to dose in cell age,coh times 1 minus probability of surviving in that cell. Only implemented for `model.family="binomial.dose.response"`. See details.

`trap.survival.forecast`
: Matrix. Point forecasts. Forecasts in lower right triangle Probability of surviving computed from `trap.predictors.forecast` using logistic link function. Only implemented for `model.family="binomial.dose.response"`. See details.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 2 May 2016 (2 Mar 2016)

References

Cox, P.R. (1976) Demography. 5th Edition. Cambridge: Cambridge University Press. (page 324).

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008b) Forecasting with the age-period-cohort model and the extended chain-ladder model. Biometrika 95, 987-991. *Download*: Article; Earlier version Nuffield DP.

Kuang, D., Nielsen B. and Nielsen J.P. (2011) Forecasting in an extended chain-ladder-type model. Journal of Risk and Insurance 78, 345-359. *Download*: Article; Earlier version: Nuffield DP.

`apc.forecast.apc` *Forecast models with APC structure.*

Description

Computes forecasts for a model with APC structure. Forecasts of the linear predictor are given for all models. This is done for the triangle which shares age and cohort indices with the data.

Usage

```
apc.forecast.apc(apc.fit,extrapolation.type="I0",
suppress.warning=TRUE)
```

Arguments

`apc.fit` List. Output from `apc.fit.model`. Note: apc.fit.model should be run with APC structure so that `apc.fit$model.design=="APC"`. Point forecasts of the response are only provided for a Poisson response model where `apc.fit$model.family=="poisson.resp` and for an over-disperse Poisson response model where `apc.fit$model.family=="od.poisson.respor` For other models only point forecasts of the linear predictor are provided, that is the first two values `linear.predictors.forecast` and `index.trap.J`.

`extrapolation.type`
Character. Choices for extrapolating the differenced period parameter ("Delta.beta_per"). Default is "I0".

"I2" Extrapolates future DDbeta by 0.

"I1" Extrapolates future DDbeta as follows. Compute Dbeta=cumsum(DDbeta) for j=3,...,J. This determines Dbeta upto arbitrary level. Compute average mean(Dbeta). Forecast DDbeta[J+1]=mean(Dbeta)-Dbeta[J]. Forecast DDbeta[J+h]=0 for h>1. This forecast is invariant to arbitrary level.

"I0" Extrapolates future DDbeta as follows. Compute beta=cumsum(cumsum(DDbeta)) for j=3,...,J. This determines beta upto arbitrary linear trend. Regress on 1 and demeaned trend=j-(n+1)/2 giving estimates mu1 and mu2. Forecast beta[J+1]=mu1 + mu2*(n+1-(n+1)/2). Forecast beta[J+2]=mu1 + mu2*(n+2-(n+1)/2). Forecast DDbeta[J+h]=beta[J+h]-2*beta[J+h-1]+beta[J+h-2] for h=1,2. Forecast DDbeta[J+h]=0 for h>2. This forecast is invariant to arbitrary linear trend.

All methods are invariant to ad hoc identification of the implied period time effect, by following the ideas put forward in Kuang, Nielsen and Nielsen (2008b).

`suppress.warning`
Logical. If true, suppresses warnings from `apc.data.list.subset`, which is called internally. Default is "TRUE".

Details

The example below is based on the smaller data reserving sets `data.loss.TA`.

Value

`linear.predictors.forecast`
Vector. Linear predictors for forecast area.

`index.trap.J` Matrix. age-coh coordinates for vector. Similar structure to `index.trap` in `apc.index`, see `apc.get.index`.

`trap.response.forecast`
Matrix. Includes data and point forecasts. Forecasts in lower right triangle. Trapezoid format.

`response.forecast.cell`
Matrix. 4 columns. 1: Point forecasts. 2: corresponding forecast standard errors 3: process standard errors 4: estimation standard errors Note that the square of column 2 equals the sums of squares of columns 3 and 4 Note that `index.trap.J` gives the age-coh coordinates for each entry.

`response.forecast.age`
Same as `response.forecast.cell`, but point forecasts by age cumulated over period/cohort.

`response.forecast.per`
Same as `response.forecast.cell`, but point forecasts by `per` cumulated over age/cohort.

`response.forecast.coh`
Same as `response.forecast.cell`, but point forecasts by `coh` cumulated over age/period.

`response.forecast.all`
Same as `response.forecast.cell`, but point forecasts cumulated by `age` and `coh`.

`xi.per.dd.extrapolated`
The extrapolated double differences.

`xi.extrapolated`
The extrapolated parameters.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 10 Sep 2016

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008b) Forecasting with the age-period-cohort model and the extended chain-ladder model. Biometrika 95, 987-991. *Download*: Article; Earlier version Nuffield DP.

See Also

The example below uses Taylor and Ashe reserving see data.loss.TA

Examples

```
#####################
#  EXAMPLE with reserving data: data.loss.TA()

data <- data.loss.TA()
fit.apc <- apc.fit.model(data,"poisson.response","APC")
forecast <- apc.forecast.apc(fit.apc)

# forecasts by "policy-year"
forecast$response.forecast.coh
#          forecast
# coh_2    91718.82
# coh_3   464661.38
# coh_4   704591.94
# coh_5  1025337.23
# coh_6  1503253.81
# coh_7  2330768.44
# coh_8  4115906.56
# coh_9  4257958.30
# coh_10 4567231.84
# forecasts of "cash-flow"
forecast$response.forecast.per
#         forecast
# per_11 5274762.58
# per_12 4213526.23
# per_13 3188451.80
# per_14 2210649.45
# per_15 1644203.06
# per_16 1236495.32
# per_17  764552.75
# per_18  444205.71
# per_19   84581.44
# forecast  of "total reserve"
forecast$response.forecast.all
#     forecast
# all 19061428
```

apc.get.design *Create design matrices*

Description

Functions to create the apc design matrix for the canonical parameters. Based on Nielsen (2014b), which generalises introduced by Kuang, Nielsen and Nielsen (2008). In normal use these function are needed for internal use by apc.fit.model.

The resulting function design matrix is collinear, so a sub-set of the columns have to be selected. The columns are: intercept, age/period/cohort slopes, age/period/cohort double differences. Thus, there are three slopes instead of two. Before use, one has to select which parameters are needed. This should include at either one/two of age/cohort slopes or period slope or no slope.

Usage

```
apc.get.design(apc.index,model.design)
apc.get.design.collinear(apc.index)
```

Arguments

apc.index List. See apc.get.index for a description of the format. Note, apc.index can be replace by an apc.fit list. This is extended version of apc.index is the output from apc.fit.model.

model.design Character. This indicates the design choice. The following options are possible.

- **"APC"** Age-period-cohort model.
- **"AP"** Age-period model. Nested in "APC"
- **"AC"** Age-cohort model. Nested in "APC"
- **"PC"** Period-cohort model. Nested in "APC"
- **"Ad"** Age-trend model, including age effect and two linear trends. Nested in "AP", "AC".
- **"Pd"** Period-trend model, including period effect and two linear trends. Nested in "AP", "PC".
- **"Cd"** Cohort-trend model, including cohort effect and two linear trends. Nested in "AC", "PC".
- **"A"** Age model. Nested in "Ad".
- **"P"** Period model. Nested in "Pd".
- **"C"** Cohort model. Nested in "Cd".
- **"t"** Trend model, with two linear trends. Nested in "Ad", "Pd", "Cd".
- **"tA"** Single trend model in age index. Nested in "A", "t".
- **"tP"** Single trend model in period index. Nested in "P", "t".
- **"tC"** Single trend model in cohort index. Nested in "C", "t".
- **"1"** Constant model. Nested in "tA", "tP", "tC".
- **NULL** The function then looks for information on model design in the first argument.

The model.design argument is not needed if the first argument is of type apc.fit. If given, the model.design argument is used.

Value

apc.get.design returns a list with

`design` Matrix. The design matrix. The number of rows is the number of observations, that is `apc.index$n.data`. The order of the observations corresponds to the internal choice made in `apc.get.index`.

`slopes` Vector. For internal use. Length 3 of logicals, indicate presence of age/period/cohort linear slopes at most two slopes can be present if neither age/cohort present then period may be presents, which is the case for model.design "P","tP"

`difdif` Vector. For internal use. Length 3 of logicals

apc.get.design.collinear returns a collinear design matrix for the unrestricted "APC" model. It has an extra column. The columns 2-4 are linear trends in age, period and cohort directions. At most two of these should be used. They are selected by `slopes`.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 1 Mar 2015

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014b) Deviance analysis of age-period-cohort models.

See Also

The vignette `NewDesign.pdf`, `NewDesign.R` on `Vignettes`.

Examples

```
#####################
# EXAMPLE 1 with Belgian lung cancer data
# This example illustrates how apc.fit.model works.

data.list <- data.Belgian.lung.cancer()

# Vectorise data
index <- apc.get.index(data.list)
v.response <- data.list$response[index$index.data]
v.dose <- data.list$dose[index$index.data]

# Get design
m.design.apc <- apc.get.design(index,"APC")$design

# Fit using glm.fit from stats package
fit.apc.glm <- glm.fit(m.design.apc,v.response,family=poisson(link="log"),offset=log(v.dose))
fit.apc.glm$deviance
```

```
# Compare with standard output from apc.fit.model
apc.fit.model(data.list,"poisson.dose.response","APC")$deviance

#####################
# EXAMPLE 2 with Belgian lung cancer data
# The age-drift model gives a good fit.
# This fit can be refined to a cubic or quadratic age effect.
# The latter is not precoded so one will have to work directly with the design matrix.
# SEE ALSO VIGNETTE

data.list <- data.Belgian.lung.cancer()

# Vectorise data
index <- apc.get.index(data.list)
v.response <- data.list$response[index$index.data]
v.dose <- data.list$dose[index$index.data]

# Get design matrix for "Ad"
m.design.ad <- apc.get.design(index,"Ad")$design

# Modify design matrix for cubic or quadratic age effect
# Note this implies a linear or constant double difference
# Quadractic age effect: restrict double differences to be equal
p <- ncol(m.design.ad)
m.rest.q <- matrix(data=0,nrow=p,ncol=4)
m.rest.q[1,1] <- 1
m.rest.q[2,2] <- 1
m.rest.q[3,3] <- 1
m.rest.q[4:p,4] <- 1
m.design.adq <- m.design.ad %*% m.rest.q
# Cubic age effect: restrict double differences to be linear
m.rest.c <- matrix(data=0,nrow=p,ncol=5)
m.rest.c[1,1] <- 1
m.rest.c[2,2] <- 1
m.rest.c[3,3] <- 1
m.rest.c[4:p,4] <- 1
m.rest.c[4:p,5] <- seq(1,p-3)
m.design.adc <- m.design.ad %*% m.rest.c

# Poisson regression for dose-response and with log link
fit.ad <- glm.fit(m.design.ad,v.response,family=poisson(link="log"),offset=log(v.dose))
fit.adc <- glm.fit(m.design.adc,v.response,family=poisson(link="log"),offset=log(v.dose))
fit.adq <- glm.fit(m.design.adq,v.response,family=poisson(link="log"),offset=log(v.dose))

# Deviance tests
fit.adc$deviance - fit.ad$deviance
fit.adq$deviance - fit.ad$deviance
# Degrees of freedom
ncol(m.design.ad) - ncol(m.design.adc)
ncol(m.design.ad) - ncol(m.design.adq)
```

`apc.get.index` *Get indices for mapping data into trapezoid formation*

Description

This function does the internal book keeping between the original data format and the trapezoid format. It creates index matrices to transform data between original format, trapezoid format and a vector, as well as values to keep track of the labels for the time scales.

The generalized trapezoids are introduced in Kuang, Nielsen and Nielsen (2008), see also Nielsen (2014).

Usage

```
apc.get.index(apc.data.list)
```

Arguments

`apc.data.list` See `apc.data.list` for a description of the format

Value

A list containing the following values.

`response` Matrix. An argument

`dose` Matrix or NULL. An argument

`data.format` Character. An argument

`unit` Numeric. An argument.

`data.xmax` Numeric. Number of rows of response matrix.

`data.ymax` Numeric. Number of columns of response matrix.

`data.xlab` Character. Label for row index of response matrix. Derived from `data.format`.

`data.ylab` Character. Label for column index of response matrix. Derived from `data.format`.

`data.xlab1` Numeric. Year for smallest row index of response matrix.

`data.ylab1` Numeric. Year for smallest column index of response matrix.

`n.data` Numeric. Number of observations.

`index.data` Matrix of dimension `n.datax2`. Index pairs for observations in the original coordinate system as given by `data.format`. Same order as in `index.trap`.

`index.trap` Matrix of dimension `n.datax2`. Index pairs for observations in an age/cohort system. Hence the coordinates of a trapezoid matrix. Same order as in `index.data`.

`age.max` Numeric. Number of age groups.

`per.max` Numeric. Number of period groups.

`coh.max` Numeric. Number of cohort groups.

`per.zero` Numeric. Anchor for period index, so that period starts from `per.zero`+1.

`per.odd` Logic. TRUE if per.zero is odd.

`U` Numeric. Integer value of (per.zero+3)/2.

`age1` Numeric. Year for smallest age index. Derived for data.format="CP", "PC", otherwise an argument.

`per1` Numeric. Year for smallest period index. Derived for data.format="AC","CA","CL","CL.vector.by.row"," otherwise an argument.

`coh1` Numeric. Year for smallest cohort index. Derived for data.format="AP", "PA", otherwise an argument.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 31 Mar 2015

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014) Deviance analysis of age-period-cohort models. Nuffield DP.

Examples

```
#################
# Artificial data

################
# Artificial data
# Generate a 3x5 matrix and make arbitrary decisions for rest

response <- matrix(data=seq(1:15),nrow=3,ncol=5)
data.list <- list(response=response,dose=NULL,data.format="AP",
age1=25,per1=1955,coh1=NULL,
unit=5,per.zero=NULL,per.max=NULL,time.adjust=0)
apc.get.index(data.list)
```

`apc.hypothesis` *Imposing hypotheses on age-period-cohort models.*

Description

apc has a set of standard hypotheses that can be imposed on the age-period-cohort model. A deviance table can be found on `apc.fit.table`, while fits of restricted models can be found using `apc.fit.model`.

Other linear hypotheses can be imposed using a little bit of coding, see the vignette `NewDesign.pdf`, `NewDesign.R` on `Vignettes`.

For over-dispersed Poisson models for responses and no doses the theory is worked out in Harnau and Nielsen (2017).

In general forecasts from age-period-cohort models require extrapolation of the estimated parameters. This has to be done without introducing identifications problems, see Kuang, Nielsen and Nielsen (2008b,2011). There are many different possibilities for extrapolation for the different sub-models. The extrapolation results in point forecasts. Distribution forecasts should be build on top of these, see Martinez Miranda, Nielsen and Nielsen (2015) and Harnau and Nielsen (2016). At present three experimental functions `apc.forecast.ac`, `apc.forecast.apc` and `apc.forecast.ap` are available.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 10 Sep 2016 (1 Feb 2016)

References

Harnau, J. and Nielsen (2016) Over-dispersed age-period-cohort models. To appear in *Journal of the American Statistical Association*. *Download*: Nuffield DP

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008b) Forecasting with the age-period-cohort model and the extended chain-ladder model. Biometrika 95, 987-991. *Download*: Article; Earlier version Nuffield DP.

Kuang, D., Nielsen B. and Nielsen J.P. (2011) Forecasting in an extended chain-ladder-type model. Journal of Risk and Insurance 78, 345-359. *Download*: Article; Earlier version: Nuffield DP.

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

`apc.identify` *Identification of time effects*

Description

Computes ad hoc identified time effects.

Usage

```
apc.identify(apc.fit.model)
```

Arguments

`apc.fit.model`	List. See `apc.fit.model` for a description of the format.

Details

Forms ad hoc identified time effects from the canonical parameter. These are used either indirectly by `apc.plot.fit` or they are computed directly with this command.

The ad hoc identifications are based on Nielsen (2014b). For details see also the vignette `Identification.pdf`, `Identification.R` on `Vignettes` or in the notes below.

For model designs of any type two ad hoc identified time effects.

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored in the middle of the first period diagonal.

(2) The type "detrend" gives double sums that start in zero and end in zero.

For model designs with only two time effects, that is "AC", "AP", "PC" there is a further ad hoc identification.

(3) The type "demean" gives single sums of single differences. Derived from "detrend" where the linear trends are attributed to the double sums of double differences. Level unchanged.

(4) The type "dif" gives the single differences derived from "demean". Could also have been chosen as canonical parametrisation for these models.

Value

`index.age.max` Vector. Indices for age parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.max` Vector. Indices for period parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.max` Vector. Indices for cohort parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.max` Vector. Indicates the dates for the parameters when using coefficients.ssdd or coefficients.detrend. The length is six longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`index.age.sub` * Vector. Indices for age parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.sub` * Vector. Indices for period parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.sub` * Vector. Indices for cohort parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.sub` * Vector. Indicates the dates for the parameters when using coefficients.demean. The length is six longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`index.age.dif` * Vector. Indices for age parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.dif` * Vector. Indices for period parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.dif` * Vector. Indices for cohort parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.dif` * Vector. Indicates the dates for the parameters when using coefficients.dif. The length is three longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`coefficients.ssdd`

Matrix. Coefficients of the double sum of double differences. Normalised to be zero at two values chosen so age=cohort and period is at the minimal value. For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.ssdd`

Matrix. Estimated covariance matrix for double sums.

`coefficients.detrend`

Matrix. Coefficients of the double sum of double differences. Normalised to be zero for first and last value. For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.detrend`

Matrix. Estimated covariance matrix for detrended double sums.

`coefficients.demean`

* Matrix. Coefficients of the sum of differences. Normalised to be zero for first value. Does not apply is design is "APC" For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.demean`

* Matrix. Estimated covariance matrix for demeaned sums.

`coefficients.dif`

* Matrix. Coefficients of the differences. Does not apply is design is "APC" For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.dif`

* Matrix. Estimated covariance matrix for differences.

Note

* indicates that values only implemented for designs "AC", "AP", "PC".

The differences are not identified for design "APC". An arbitrary level can be moved between differences for age, period and cohort.

The differences are not identified for designs "Ad", "Pd", "Cd". These models have two linear trends and one set of double differences. In the model "Ad", as an example, one linear trend will be associated with age, but it is arbitrary whether the second linear trend should be associated with period or cohort. The slope of the age trend will depend on that arbitrary choice. In turn the level of the age differences will be arbitrary.

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored to be zero in the three points where `age=cohort=U`, `age=U+1,cohort=U` `age=U,cohort=U+1` with `apc.fit.model$U` and where `U` is the integer value of `(per.zero+3)/2` This corresponds to the representation in Nielsen (2014b). The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=U`; an age slope, which is the difference of the values of the predictor at `age=U+1,cohort=U` and `age=cohort=U`; an cohort slope, which is the difference of the values of the predictor at `age=U,cohort=U+1` and `age=cohort=U`.

(2) The type "detrend" gives double sums that start in zero and end in zero. The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=1`, which is usually outside the index set for the data; while age and cohort slopes are adjusted for the ad hoc identification of the time effects.

(3) Subsumes `var.apc.identify` from `apc.indiv` (25 Sep 2020)

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> & Zoe Fannon 25 Sep 2020 (12 Apr 2015)

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014b) Deviance analysis of age-period-cohort models. Work in progress.

See Also

The vignette Identification.pdf.

Examples

```
########################
# Belgian lung cancer
#  first an example with APC design, note that demean and dif not defined.

data.list <- data.Belgian.lung.cancer()

fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")
fit.apc$coefficients.canonical
id.apc <- apc.identify(fit.apc)
id.apc$coefficients.ssdd
id.apc$coefficients.detrend
id.apc$coefficients.demean
id.apc$coefficients.dif

fit.ap <- apc.fit.model(data.list,"poisson.dose.response","AP")
fit.ap$coefficients.canonical
id.ap <- apc.identify(fit.ap)
id.ap$coefficients.ssdd
id.ap$coefficients.detrend
id.ap$coefficients.demean
id.ap$coefficients.dif
```

apc.indiv.compare.direct *Implements direct tests between APC models*

Description

This function allows the user to directly compare any of the APC model, its submodels, or the TS model to any smaller model. For example, the function can be used to compare the TS to the Ad model or the Ad model to the A model. Comparisons are by likelihood ratio or Wald tests.

Usage

```
apc.indiv.compare.direct(data, big.model, small.model, unit=1,
         dep.var, covariates=NULL, model.family,
                    n.coh.excl.start=0, n.coh.excl.end=0,
                    n.age.excl.start=0, n.age.excl.end=0,
                    n.per.excl.start=0, n.per.excl.end=0,
                    NR.controls=NULL, test, dist,
                    wt.var=NULL, plmmodel="notplm", id.var=NULL)
        apc.indiv.waldtest.fullapc(data, dist="F", big.model="APC",
                    small.model, dep.var, covariates=NULL,
                    model.family="gaussian", unit=1,
                    n.coh.excl.start=0, n.coh.excl.end=0,
                    n.age.excl.start=0, n.age.excl.end=0,
                    n.per.excl.start=0, n.per.excl.end=0,
                    existing.big.model.fit=NULL,
                    existing.small.model.fit=NULL,
                    existing.collinear=NULL,
plmmodel = "notplm", id.var=NULL, wt.var=NULL)
        apc.indiv.waldtest.TS(data, dist="F", small.model="APC",
                    dep.var, covariates=NULL,
                    model.family="gaussian", unit=1,
                    n.coh.excl.start=0, n.coh.excl.end = 0,
                    n.age.excl.start=0, n.age.excl.end = 0,
                    n.per.excl.start=0, n.per.excl.end = 0,
                    existing.small.model.fit=NULL,
                    existing.big.model.fit=NULL,
                    existing.collinear=NULL)
        apc.indiv.LRtest.fullapc(data,  big.model="APC",
                    small.model,
                    dep.var, covariates=NULL,
                    model.family="binomial", unit=1,
                    n.coh.excl.start=0, n.coh.excl.end=0,
                    n.age.excl.start=0, n.age.excl.end=0,
                    n.per.excl.start=0, n.per.excl.end=0,
                    existing.big.model.fit=NULL,
                    existing.small.model.fit=NULL,
                    existing.collinear=NULL)
       apc.indiv.LRtest.TS(data, small.model=”APC”, dep.var, covariates=NULL,
                    model.family=”binomial”, unit=1,
                    n.coh.excl.start=0, n.coh.excl.end=0,
                    n.age.excl.start=0, n.age.excl.end=0,
                    n.per.excl.start=0, n.per.excl.end=0,
                    existing.small.model.fit=NULL,
                    existing.big.model.fit=NULL,
                    existing.collinear=NULL,
                    NR.controls=NULL)
```

Arguments

`data` The data.frame in use.

`big.model` The name of the larger of the two models to be tested.

`small.model` The name of the smaller of the two models to be tested.

`unit` The interval at which age, period, and cohort are recorded (must be the same for each). Default 1.

`dep.var` The name of the dependent variable as it appears in the data

`covariates` A vector of the names of covariates as they appear in the data. Default NULL.

`model.family` Either "gaussian" or "binomial"

`n.coh.excl.start` If any cohorts have been censored (AP data only). Default 0.

`n.coh.excl.end` If any cohorts have been censored (AP data only). Default 0.

`n.per.excl.start` If any periods have been censored (AC data only). Default 0.

`n.per.excl.end` If any periods have been censored (AC data only). Default 0.

`n.age.excl.start` If any ages have been censored (PC data only). Default 0.

`n.age.excl.end` If any ages have been censored (PC data only). Default 0.

`NR.controls` Optional list to modify aspects of the Newton-Rhapson iteration for binomial TS model. See details in `apc.indiv.est.model`.

`test` The type of test. One of "LR", "Wald".

`dist` The distribution against which the test statistic is compared. One of "F", "Chisq".

`wt.var` Only if using survey weights. The name of the weights variable.

`plmmodel` Used to indicate whether a panel data model is to be estimated and if so what type. Default is "notplm", for not panel data. Other values are "pooling", "within", "random". Further details in `plm`.

`id.var` Only if using panel data. The name of the individual ID variable.

existing.big.model.fit
: Optional specify the output of apc.indiv.fit.model, if already run for the big model.

existing.small.model.fit
: Optional specify the output of apc.indiv.fit.model, if already run for the small model.

existing.collinear
: Optional specify the output of apc.indiv.design.collinear, if already run.

Details

These functions are designed to facilitate direct comparison between sub-models. The functions are used to construct the rows of tables in `apc.indiv.model.table` but can also more helpfully be used to compare nested sub-models that gain similar levels of suport from such a table, e.g. PC to P.

Value

test.type	The type of test, one of "LR", "Wald".
dist.type	The distribution against which the test statistic is compared. One of "F", "Chisq".
test.stat	The value of the test statistic.
df	Degrees of freedom.
df.num	Gaussian models only. Degrees of freedom used in the numerator of the F-statistic.
df.denom	Gaussian models only. Degrees of freedom used in the denominator of the F-statistic.
p.value	P-value from testing against a chi-square or F distribution.
aic.big	AIC of the big model.
aic.small	AIC of the small model.
lik.big	Log-likelihood of the big model.
lik.small	Log-likelihood of the small model.
NR.report	Binomial TS model only. Report on the Newton-Rhapson algorithm.

Author(s)

Zoe Fannon <zoe.fannon@economics.ox.ac.uk> 26 Jun 2020

References

Fannon, Z. (2018) `apc.indiv`: R tools to estimate age-period-cohort models with repeated cross section data. Mimeo. University of Oxford.

Fannon, Z., Monden, C. and Nielsen, B. (2018) Age-period-cohort modelling and covariates, with an application to obesity in England 2001-2014. Mimeo. University of Oxford.

See Also

For model estimation: apc.indiv.est.model. The data in these examples are the Wage data from the package ISLR and the PSID7682 data from the package AER.

For examples, see the vignette IntroductionIndividualData.pdf, IntroductionIndividualData.R on Vignettes. Further examples in the vignette IntroductionIndividualDataFurtherExamples.pdf, IntroductionIndividualDataFurtherExamples.R.

Examples

```
#### see vignettes
```

apc.indiv.est.model *Estimate a single APC model*

Description

The function apc.indiv.est.model is used to estimate any of: the APC model, any APC submodel, or the time-saturated model. To estimate the APC model or a submodel, it calls apc.indiv.design.collinear, apc.indiv.design.model, and apc.indiv.fit.model in that order. To estimate the time-saturated (TS) model it calls either apc.indiv.estimate.TS or apc.indiv.logit.TS, depending on the selected model.family. These functions can also be called directly by the user.

Usage

```
apc.indiv.est.model(data, unit = 1,
                    n.coh.excl.start=0, n.coh.excl.end=0,
                    n.per.excl.start=0, n.per.excl.end=0,
                    n.age.excl.start=0, n.age.excl.end=0,
                    model.design = "APC", dep.var = NULL,
                    covariates = NULL, model.family = "gaussian",
                    NR.controls = NULL,
                    existing.collinear = NULL,
                    existing.design = NULL,
plmmodel = "notplm", id.var = NULL,
                    wt.var = NULL)
apc.indiv.design.collinear(data, unit = 1,
                           n.coh.excl.start = 0, n.coh.excl.end = 0,
                           n.per.excl.start = 0, n.per.excl.end = 0,
                           n.age.excl.start = 0, n.age.excl.end = 0)
apc.indiv.design.model(apc.indiv.design.collinear,
                       model.design = "APC", dep.var = NULL,
                       covariates = NULL, plmmodel = "notplm",
                       wt.var = NULL, id.var = NULL)
apc.indiv.fit.model(apc.indiv.design.model, model.family = "gaussian", DV = NULL)
apc.indiv.estimate.TS(data, dep.var, covariates = NULL)
apc.indiv.logit.TS(data, dep.var, covariates = NULL, NR.controls = NULL)
```

Arguments

`data` The data.frame in use

`unit` The interval at which age, period, and cohort are recorded (must be the same for each). Default 1.

`n.coh.excl.start` If any cohorts have been censored (AP data only). Default 0.

`n.coh.excl.end` If any cohorts have been censored (AP data only). Default 0.

`n.per.excl.start` If any periods have been censored (AC data only). Default 0.

`n.per.excl.end` If any periods have been censored (AC data only). Default 0.

`n.age.excl.start` If any ages have been censored (PC data only). Default 0.

`n.age.excl.end` If any ages have been censored (PC data only). Default 0.

`model.design` The name of the model to be estimated. One of "TS", "APC", "AC", etc.

`dep.var` The name of the dependent variable as it appears in the data

`DV` apc.indiv.fit.model only. Optional. Vector containing dependent variable.

`covariates` A vector of the names of covariates as they appear in the data. Default NULL.

`plmmodel` Used to indicate whether a panel data model is to be estimated and if so what type. Default is "notplm", for not panel data. Other values are "pooling", "within", "random". Further details in `plm`.

`id.var` Only if using panel data. The name of the individual ID variable.

`wt.var` Only if using survey weights. The name of the weights variable.

`model.family` Either "gaussian" or "binomial". Default "gaussian".

`NR.controls` Optional list to modify aspects of the Newton-Rhapson iteration for binomial TS model. Further information in "Details", below.

`existing.collinear` Optional specify the output of apc.indiv.design.collinear, if already run.

`existing.design` Optional specify the output of apc.indiv.design.model, if already run.

`apc.indiv.design.collinear` Output from the command `apc.indiv.design.collinear`.

`apc.indiv.design.model` Output from the command `apc.indiv.design.model`.

Details

The casual user should start with the general function `apc.indiv.est.model` for analysis. The underlying functions should be employed if the user needs to run many models using the same relatively large dataset, in which case time can be saved by running `apc.indiv.design.collinear` just once and using `apc.indiv.design.model` and `apc.indiv.fit.model` to estimate each of the models.

The time-saturated (TS) binomial model is estimated by a customized Newton-Rhapson iteration. Aspects of this iteration can be controlled by specifying the `NR.controls` option of `apc.indiv.est.model` or of apc.indiv.logit.TS. NR.controls is a named list of length 8. In order, the elements are: maxit.loop, maxit.linesearch, tolerance, init, inv.tol, d1.tol, custom.kappa, custom.zeta. maxit.loop sets the maximum number of Newton-Rhapson iterations, and has a default of 10. maxit.linesearch sets the maximum number of linesearch iterations within each Newton-Rhapson iteration, and has a default of 20. tolerance sets the condition for convergence, i.e. the tolerated difference between likelihoods from one Newton-Rhapson iteration to the next; the default is .002. init sets the starting values for the iteration. The default is "ols", meaning that estimates from the linear probability model are the starting values; one can also use "zero" to set the starting values to zero, or use "custom" and specify custom starting values using custom.kappa and custom.zeta. inv.tolsets the tolerance of small values when inverting a matrix (using solve), and the default is the machine precision. d1.tol sets the magnitude of norm of first derivative to be tolerated in Newton-Rhapson iteration, and has a default of .002. custom.kappa is used to specify custom starting values for the TS indicator parameters, while custom.zeta is used to specify custom starting values for parameters on any covariates.

Value

fit The output of either glm, svyglm, or plm for repeated cross-section, repeated cross-section with survey weights, or panel models respectively. Can be used directly with follow-on functions like waldtest

.

coefficients.canonical
Matrix of estimates, standard error, t-statistic, and p-value of canonical parameter.

coefficients.covariates
Matrix of estimates, standard error, t-statistic, and p-value of covariates.

coefficients.TS
TS model only: matrix of estimates, standard error, t-statistic, and p-value of TS indicators.

aic TS model only: Akaike Information Criterion.

likelihood model likelihood.

model.design which APC submodel has been estimated.

fixef When plmmodel = "within", estimated individual fixed effects. Otherwise NULL.

full.design.collinear
from apc.indiv.design.collinear only. The collinear design matrix.

full.design from apc.indiv.design.model only. The design matrix used to estimate the model.

DV from apc.indiv.design.model only, if dep.var specified. A vector of the outcome variable.

ID from apc.indiv.design.model only, if panel model. A vector of the individual ID variable.

PER from apc.indiv.design.model only, if panel model. A vector of the period variable.

WT from apc.indiv.design.model only, if wt.var specified. A vector of the survey weight variable.

model.formula from apc.indiv.design.model only, the implied model formula. NULL if dep.var not specified.

model.string from apc.indiv.design.model only, the implied model formula as a character string. RHS only if dep.var not specified.

Author(s)

Zoe Fannon <zoe.fannon@economics.ox.ac.uk> 26 Jun 2020

References

Fannon, Z. (2018) apc.indiv: R tools to estimate age-period-cohort models with repeated cross section data. Mimeo. University of Oxford.

Fannon, Z., Monden, C. and Nielsen, B. (2018) Age-period-cohort modelling and covariates, with an application to obesity in England 2001-2014. Mimeo. University of Oxford.

See Also

For model estimation: glm, svyglm, plm For model testing: apc.indiv.model.table, codeapc.indiv.compare.direct, waldtest, linearHypothesis For plotting: apc.plot.fit. The data in these examples are the Wage data from the package ISLR and the PSID7682 data from the package AER.

For examples, see the vignette IntroductionIndividualData.pdf, IntroductionIndividualData.R on Vignettes. Further examples in the vignette IntroductionIndividualDataFurtherExamples.pdf, IntroductionIndividualDataFurtherExamples.R.

Examples

```
#### see vignettes
```

---

apc.indiv.model.table *Generate table to select APC submodel*

Description

These functions test, for a given choice of dependent variable and covariates, which of the TS, APC, and APC submodels provides the best fit to the data. Comparison is by Wald or likelihood ratio test and where appropriate by Akaike Information Criterion. A table is generated with these statistics for each model considered.

Usage

```
apc.indiv.model.table(data, dep.var, covariates = NULL,
unit = 1, n.coh.excl.start = 0, n.coh.excl.end = 0,
n.age.excl.start = 0, n.age.excl.end = 0,
n.per.excl.start = 0, n.per.excl.end = 0,
model.family, NR.controls = NULL,
test, dist,
TS=FALSE, wt.var=NULL, plmmodel="notplm",
id.var=NULL)
   apc.indiv.waldtable(data, dep.var, covariates = NULL,
   dist="F", unit = 1, model.family,
     n.coh.excl.start = 0, n.coh.excl.end = 0,
n.age.excl.start = 0, n.age.excl.end = 0,
n.per.excl.start = 0, n.per.excl.end = 0,
 wt.var=NULL, plmmodel="notplm",
id.var=NULL)
apc.indiv.waldtable.TS(data, dep.var, covariates=NULL, dist = "F",
                                unit=1, model.family = "gaussian",
                                n.coh.excl.start=0, n.coh.excl.end=0,
                                n.age.excl.start=0, n.age.excl.end=0,
                                n.per.excl.start=0, n.per.excl.end=0)
     apc.indiv.LRtable(data,  dep.var, covariates=NULL,
                              model.family, unit=1,
                              n.coh.excl.start=0, n.coh.excl.end=0,
                              n.age.excl.start=0, n.age.excl.end=0,
                              n.per.excl.start=0, n.per.excl.end=0)
   apc.indiv.LRtable.TS(data,  dep.var, covariates=NULL,
                                 model.family, unit=1,
                                 n.coh.excl.start=0, n.coh.excl.end=0,
                                 n.age.excl.start=0, n.age.excl.end=0,
                                 n.per.excl.start=0, n.per.excl.end=0,
                                 NR.controls=NR.controls)
```

Arguments

`data`	The data.frame in use
`dep.var`	The name of the dependent variable as it appears in the data
`covariates`	A vector of the names of covariates as they appear in the data. Default NULL.
`unit`	The interval at which age, period, and cohort are recorded (must be the same for each). Default 1.
`n.coh.excl.start`	If any cohorts have been censored (AP data only). Default 0.
`n.coh.excl.end`	If any cohorts have been censored (AP data only). Default 0.
`n.age.excl.start`	If any ages have been censored (PC data only). Default 0.
`n.age.excl.end`	If any ages have been censored (PC data only). Default 0.
`n.per.excl.start`	If any periods have been censored (AC data only). Default 0.
`n.per.excl.end`	If any periods have been censored (AC data only). Default 0.
`model.family`	Either "gaussian" or "binomial"
`NR.controls`	Optional list to modify aspects of the Newton-Rhapson iteration for binomial TS model.See details in `apc.indiv.est.model`.
`test`	The type of test. One of "LR", "Wald".

`TS` ...

`dist` The distribution against which the test statistic is compared. One of "F", "Chisq".

`wt.var` Only if using survey weights. The name of the weights variable.

`plmmodel` Used to indicate whether a panel data model is to be estimated and if so what type. Default is "notplm", for not panel data. Other values are "pooling", "within", "random". Further details in `plm`.

`id.var` Only if using panel data. The name of the individual ID variable.

Details

Each row of the table corresponds to a single sub-model of the APC model. The first three columns test the sub-model in question against the time-saturated model. The next three columns test the sub-model against the full APC model. The final two columns report the likelihood and AIC of the estimated sub-model. The model with the lowest AIC value which is also not rejected in tests against the APC and TS models should be selected.

Value

`table` contains the table of comparison statistics.

`NR.report` for logit models only, a report on the Newton-Rhapson algorithm used to estimate the time-saturated model.

Author(s)

Zoe Fannon <zoe.fannon@economics.ox.ac.uk> 26 Jun 2020

References

Fannon, Z. (2018) `apc.indiv`: R tools to estimate age-period-cohort models with repeated cross section data. Mimeo. University of Oxford.

Fannon, Z., Monden, C. and Nielsen, B. (2018) Age-period-cohort modelling and covariates, with an application to obesity in England 2001-2014. Mimeo. University of Oxford.

See Also

For model estimation: `apc.indiv.est.model` For pairwise model comparison: `apc.indiv.model.table`, `waldtest`, `linearHypothesis`. The data in these examples are the `Wage` data from the package ISLR and the `PSID7682` data from the package AER.

For examples, see the vignette `IntroductionIndividualData.pdf`, `IntroductionIndividualData.R` on `Vignettes`. Further examples in the vignette `IntroductionIndividualDataFurtherExamples.pdf`, `IntroductionIndividualDataFurtherExamples.R`.

Examples

```
#### see vignettes
```

apc.plot.data.all *Make all descriptive plots.*

Description

Plots data sums using apc.plot.data.sums. Sparsity plots of data using apc.plot.data.sparsity. Plots data using all combinations of two time scales using apc.plot.data.within. Level plots of data using apc.plot.data.level. The latter plot is done for responses and if applicable also for doses and mortality rates.

Usage

```
apc.plot.data.all(apc.data.list,log ="",rotate=FALSE)
```

Arguments

apc.data.list	List. See apc.data.list for a description of the format.
log	Optional plot argument. Character. "y" if y-scale is logarithmic, otherwise "". Default is "y".
rotate	Optional. Logical. If TRUE rotates apc.plot.data.level 90 degrees clockwise (or anti-clockwise if data.format is "CL"). Default is FALSE.

Warning

A warning is produced if dimension is not divisible by thin, so that one group is smaller than other groups.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 25 Apr 2015

See Also

The example below uses Italian bladder cancer data, see data.Italian.bladder.cancer

Examples

```
#####################
#  EXAMPLE with artificial data
#  generate a 3x4 matrix in "AP" data.format with the numbers 1..12

m.data   <- matrix(data=seq(length.out=12),nrow=3,ncol=4)
m.data
data.list <- apc.data.list(m.data,"AP")
apc.plot.data.all(data.list,log="")

#####################
# EXAMPLE with Italian bladder cancer data
```

```
#
# get data list, then make all descriptive plots.
#  Note that warnings are given in relation to the data chosen thinning
# This can be avoided by working with the individual plots, and in particular
# with apc.plot.data.within where the thinning happens.
#
# data.list <- data.Italian.bladder.cancer()
# apc.plot.data.all(data.list)
```

`apc.plot.data.level` *Level plot of data matrix.*

Description

This plot shows level plot of data matrix based on `levelplot` in the package `lattice`.

Usage

```
apc.plot.data.level(apc.data.list,data.type="r",
  rotate=FALSE,apc.index=NULL,
main=NULL,lab=NULL,
contour=FALSE,colorkey=TRUE)
```

Arguments

`apc.data.list`	List. See `apc.data.list` for a description of the format.
`data.type`	Optional. Character. "r"="response" / "d"="dose" / "m"="mortality"="rates" if sums are computed for responses/dose/rates, where rates are found through division response/dose. It also takes data types "residual" / "fitted.values" / "linear.predictors" when the argument `apc.data.list` is the output of the fitting function `apc.fit.model`, which is an extended apc.data.list. "r" is default.
`rotate`	Optional. Logical. If TRUE rotates plot 90 degrees clockwise (or anti-clockwise if data.format is "CL"). Default is FALSE.
`apc.index`	Optional. List. See `apc.get.index` for a description of the format. If not provided this is computed.
`main`	Optional. Character. Main title.
`lab`	Optional `plot` parameter. A numerical vector of the form c(x, y, len) which modifies the default way that axes are annotated. The values of x and y give the (approximate) number of tickmarks on the x and y axes. len is not implemented.
`contour`	Optional `levelplot` (`lattice`) parameter. Logical. Contour lines drawn if TRUE. Default FALSE.
`colorkey`	Optional `levelplot` (`lattice`) parameter. Logical or list. Determines color key. Default TRUE.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 26 Apr 2015

See Also

`data.Japanese.breast.cancer` for information on the data used in the example.

Examples

```
#####################
# EXAMPLE with Japanese breast cancer data
# Clayton and Shifflers (1987b) use APC design
# Make a data list
# Then plot data.
# Note: No plot appears to have approximately parallel lines.

data.list <- data.Japanese.breast.cancer()
apc.plot.data.level(data.list,"r")
dev.new()
apc.plot.data.level(data.list,"d",contour=TRUE)

# It also works with a single argument, but then a default log scale is used.
#  Note that warnings are given in relation to the data chosen thinning

apc.plot.data.within(data.list)

#####################
# EXAMPLE with Italian bladder cancer data
# Clayton and Shifflers (1987a) use AC design
# Note: plot of within cohort against age appears to have approximately parallel lines.
#   This is Figure 2 in Clayton and Shifflers (1987a)
# Note: plot of within age against cohort appears to have approximately parallel lines.
#   Indicates that interpretation should be done carefully.

data.list <- data.Italian.bladder.cancer()
apc.plot.data.within(data.list,"m",1,log="y")

#####################
# EXAMPLE with asbestos data
# Miranda Martinex, Nielsen and Nielsen (2014).
# This is Figure 1d

data.list <- data.asbestos()
apc.plot.data.within(data.list,type="l",lty=1)
```

---

apc.plot.data.sparsity *This plot shows heat map of the sparsity of a data matrix.*

Description

The plot shows where the data matrix is sparse.

Usage

```
apc.plot.data.sparsity(apc.data.list,
data.type="a",swap.axes=FALSE,
apc.index=NULL,
sparsity.limits=c(1,2),
cex=NULL,pch=15,
main.outer=NULL)
```

Arguments

`apc.data.list` List. See `apc.data.list` for a description of the format.

`data.type` Optional. Character. "r"/"d"/"m" if sums are computed for responses/dose/all. "r" is default.

`swap.axes` Optional. Logical. If true swap axes in plot. Default is FALSE unless data.format="CL"

`apc.index` Optional. List. See `apc.get.index` for a description of the format. If not provided this is computed.

`sparsity.limits`

Optional. vector with two values in increasing order. Default is c(1,2). The sparsity plot is a heat map with three colours: black if the observation is smaller than first index (default 1), grey if the observation is smaller than the second index (default 2) and otherwise white.

`cex` Optional `plot` argument. A numerical value giving the amount by which plotting text and symbols should be magnified. Default is NULL in which case program chooses.

`pch` Optional. vector with two values. Either integers specifying a symbol or characters. See `points` for possible values and their interpretation. Default is c(15,15), which is filled square.

`main.outer` Optional. Character. Main title for plot, to be shown in outer margin. Default is NULL, in which case a title is generated internally.

Details

The default values is used to highlight where a matrix of counts has values of zero and one. Estimation can be very noise in those areas.

Note

Note that the axes for plots grow from bottom left while axes for matrices grow from top left. The exception is when data.format="CL", in which case both grow from top left.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 25 Apr 2015 updated 27 Apr 2015

See Also

The example below uses asbestos data, see `data.asbestos`

Examples

```
#####################
#  EXAMPLE with artificial data
#  generate a 3x4 matrix in "AP" data.format with the numbers 1..12

m.data   <- matrix(data=seq(length.out=12),nrow=3,ncol=4)
m.data
data.list <- apc.data.list(m.data,"AP")
apc.plot.data.sparsity(data.list)

#####################
# EXAMPLE with Japanese breast cancer data
# get data list, then make sparsity plots.

data.list <- data.asbestos()
apc.plot.data.sparsity(data.list)
```

`apc.plot.data.sums` *This plot shows sums of data matrix by age, period or cohort.*

Description

Produces plots showing age, period and cohort sums. As a default this is done both for responses and dose, giving a total of six plots.

Usage

```
apc.plot.data.sums(apc.data.list,data.type="a",
average=FALSE,keep.incomplete=TRUE,apc.index=NULL,
type="o",log="",main.outer=NULL,main.sub=NULL)
```

Arguments

`apc.data.list` List. See `apc.data.list` for a description of the format.

`data.type` Optional. Character. "r","d","m","a" if sums are computed for responses, dose, (mortality rates), all. Rates are computed as responses/doses. Default is "a".

`average` Optional. Logical. Sums are reported if FALSE, Averages are reported if TRUE. Default is FALSE.

`keep.incomplete`
Optional. Logical. If true perform calculation for incomplete sequences by removing NA. If false incomplete sequences are NA. See example in `apc.data.sums`. Default=TRUE.

`apc.index` Optional. List. See `apc.get.index` for a description of the format. If not provided this is computed.

`type` Optional `plot` argument. Character. "o" if overlaid points and lines. "l" if lines. "p" if points. Default is "o".

`log` Optional `plot` argument. Character. "y" if y-scale is logarithmic, otherwise "". Default is "".

`main.outer` Optional. Character. Main title for plot, to be shown in outer margin. Default is NULL, in which case a title is generated internally.

`main.sub` Optional. Titles for sub plots. Use with data.type "r","d","m". For data.type "a" use default. Default is NULL, in which case a title is generated internally.

Details

The data sums are computed using `apc.data.sums`. Then plotted as requested.

Note

Use `apc.data.sums` if numerical values needed.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 15 Aug 2018 (15 Dec 2013)

References

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

See Also

The example below uses Japanese breast cancer data, see `data.Japanese.breast.cancer`

Examples

```
#####################
#   EXAMPLE with artificial data
#   Generate a 3x4 matrix in ”AP” data.format with the numbers 1..12
# Then make a data list
# Then plot data sums.
# Note only 3 plots are made as there are no doses

m.data   <- matrix(data=seq(length.out=12),nrow=3,ncol=4)
m.data
data.list <- apc.data.list(m.data,”AP”)
apc.plot.data.sums(data.list)
apc.plot.data.sums(data.list,average=TRUE)
apc.plot.data.sums(data.list,keep.incomplete=FALSE)

#####################
# EXAMPLE with Japanese breast cancer data
# Make a data list
# Then plot data sums for both responses and doses.

data.list <- data.Japanese.breast.cancer()
```

```
apc.plot.data.sums(data.list)

#  Or plot data sums for responses only

apc.plot.data.sums(data.list,data.type="r")

#####################
# EXAMPLE with asbestos data
# Miranda Martinex, Nielsen and Nielsen (2013).
# This is Figure 1,a-c

data.list <- data.asbestos()
apc.plot.data.sums(data.list,type="l")
```

apc.plot.data.within *This plot shows time series of matrix within age, period or cohort.*

Description

apc.plot.data.within produces plot showing time series of matrix within age, period or cohort against one of the other two indices. apc.plot.data.within.all.six produces all six plots in one panel plot.

These plots are sometimes used to gauge how many of the age, period, cohort factors are needed: If lines are parallel when dropping one index the corresponding factor may not be needed. In practice these plots should possibly be used with care, see Italian bladder cancer example below.

Usage

```
apc.plot.data.within(apc.data.list,
data.type="r",plot.type="awc",
average=FALSE,
thin=NULL,apc.index=NULL,
ylab=NULL,type="o",log="",legend=TRUE,
lty=1:5,col=1:6,bty="n",main=NULL,
x="topleft",return=FALSE)
apc.plot.data.within.all.six(apc.data.list,
data.type="r",
average=FALSE,
thin=NULL,apc.index=NULL,
ylab=NULL,type="o",log="",legend=TRUE,
lty=1:5,col=1:6,bty="n",main.outer=NULL,
x="topleft")
```

Arguments

apc.data.list List. See apc.data.list for a description of the format.

`data.type` Optional. Character. "r"="response" / "d"="dose" / "m"="mortality"="rates" if sums are computed for responses/dose/rates, where rates are found through division response/dose. "r" is default.

`plot.type` Optional. "awp", "pwa" "awc", "cwa, "cwp", "pwc": for example: "awp" gives time series in age within each period level: for an AP data-array these are the column sums.

`average` Optional. Logical. If TRUE/FALSE reports averages/sums. Default is FALSE.

`thin` Optional. Numerical. age/periods/cohorts are grouped in groups of size thin. Default is computed from dimensions of data. A warning is produced if dimension is not divisible by thin, so that one group is smaller than other groups.

`apc.index` Optional. List. See `apc.get.index` for a description of the format. If not provided this is computed.

`ylab` Optional `plot` argument. Character. Common label for y-axes. Default is "".

`type` Optional `plot` argument. Character. "o" if overlaid points and lines. "l" if lines. "p" if points. Default is "o".

`log` Optional `plot` argument. Character. "y" if y-scale is logarithmic, otherwise "". Default is ""

`legend` Optional `plot` argument. Logical. Should legends be drawn? Default is TRUE.

`lty` Optional `plot` argument. Vector of line types. The first element is for the first column, the second element for the second column, etc., even if lines are not plotted for all columns. Line types will be used cyclically until all plots are drawn. Default is 1:5

`col` Optional `plot` argument. Vector of colors. The first element is for the first column, the second element for the second column, etc., even if lines are not plotted for all columns. Colors will be used cyclically until all plots are drawn. Default is 1:6.

`bty` Optional `plot` argument. Character. The type of box to be drawn around the legend. The allowed values are "n" and "o". Default is "n".

`main` Optional. Character. Main title for single plot. Default is NULL, in which case a title is generated internally.

`main.outer` Optional. Character. Main title for panel of six plots, to be shown in outer margin. Default is NULL, in which case a title is generated internally.

`x` Optional `legend` argument. Default is "topleft".

`return` Optional. If TRUE return matrix that is plotted. Default is FALSE

Warning

A warning is produced if dimension is not divisible by thin, so that one group is smaller than other groups.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 17 Nov 2016 (25 Apr 2015)

References

Clayton, D. and Schifflers, E. (1987a) Models for temperoral variation in cancer rates. I: age-period and age-cohort models. *Statistics in Medicine* 6, 449-467.

Clayton, D. and Schifflers, E. (1987b) Models for temperoral variation in cancer rates. II: age-period-cohort models. *Statistics in Medicine* 6, 469-481.

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

See Also

`data.Japanese.breast.cancer`, `data.Italian.bladder.cancer` and `data.asbestos` for information on the data used in the example.

Examples

```
#####################
#   EXAMPLE with artificial data
#   Generate a 3x4 matrix in "AP" data.format with the numbers 1..12
# Then make a data list
# Then plot data.
#   Note: this deterministic matrix has neither age, period, or cohort factors,
#  only linear trends.  Thus all 6 plots have parallel lines.

m.data   <- matrix(data=seq(length.out=12),nrow=3,ncol=4)
m.data
data.list <- apc.data.list(m.data,"AP")
apc.plot.data.within(data.list,log="")

# It also works with a single argument, but then a default log scale is used.

apc.plot.data.within(data.list)

#####################
# EXAMPLE with Japanese breast cancer data
# Clayton and Shifflers (1987b) use APC design
# Make a data list
# Then plot data.
# Note: No plot appears to have approximately parallel lines.

data.list <- data.Japanese.breast.cancer()
apc.plot.data.within(data.list,"m",1,log="y")

# It also works with a single argument, but then a default log scale is used.
#  Note that warnings are given in relation to the data chosen thinning

apc.plot.data.within(data.list)

#####################
# EXAMPLE with Italian bladder cancer data
# Clayton and Shifflers (1987a) use AC design
# Note: plot of within cohort against age appears to have approximately parallel lines.
#   This is Figure 2 in Clayton and Shifflers (1987a)
# Note: plot of within age against cohort appears to have approximately parallel lines.
#   Indicates that interpretation should be done carefully.

data.list <- data.Italian.bladder.cancer()
apc.plot.data.within(data.list,"m",1,log="y")

#####################
# EXAMPLE with asbestos data
# Miranda Martinex, Nielsen and Nielsen (2014).
# This is Figure 1d

data.list <- data.asbestos()
apc.plot.data.within(data.list,type="l",lty=1)
```

---

apc.plot.fit *Plots of apc estimates*

Description

Functions to plot the apc estimates found by apc.fit.model. The function apc.plot.fit detects the type of model.design and model.family from the fit values and makes appropriate plots.

Depending on the model.design the plot has up to 9 sub plots. The type of these can be chosen using type

Model designs of any type. If type is "detrend" or "sum.sum" the canonical age period cohort parametrisation is used. This involves double differences of the time effects. The first row of plots are double differences of the time effects. The next two rows of plots illustrate the representation theorem depending on the choice of type. In both cases the sum of the plots add up to the predictor.

"detrend" The last row of plots are double sums of double differences detrend so that that each series starts in zero and ends in zero. The corresponding level and (up to) two linear trends are shown in the middle row of plots. The linear trends are identified to be 0 for age, period or cohort equal to its smallest value. See note 2 below.

"sum.sum" The last row of plots are double sums of double differences anchored as in the derivation of Nielsen (2014b). The corresponding level and (up to) two linear trends are shown in the middle row of plots. The linear trends are identified to be 0 for the anchoring point U of age, period or cohort as described in Nielsen (2014b). See note 1 below.

Model designs with 2 factors. If type is "dif" the canonical two factor parametrisation is used. This involves single differences. It is only implemented for model.design of "AC", "AP", "PC". It does not apply for model.design of "APC" because single differences are not identified. It does not apply for the drift models where model.design is "Ad", "Pd", "Cd", "t" because it is not clear which time scale the second linear trend should be attributed to. It is not implemented for model.design of "tA, "tP", "tC", "1". The first row of plots are single differences of the time effects. The next two rows of plots illustrate the representation theorem. In the second row the level is given and in the third row plots of single sums of single differences are given, normalised to start in zero.

Appearance may vary. Note, the plots "detrend" and "dif" can give very different appearance of the time effects. The "dif" plots are dominated by linear trends. They can therefore be more difficult to interpret than the "detrend" plots, where linear trends are set aside.

Standard deviations. All plots include plots of 1 and 2 standard deviations. The only exception is the intercept in the case `model.family` is "poisson.response" as this uses a multinomial sampling scheme, where the intercept is set to increase in the asymptotic experiment. The default is to plot standard deviations around zero, so that they represent a test for zero values of the parameters. Using the argument `sdv.at.zero` the standard deviations can be centered around the estimates. This can give a very complicated appearance.

Values of coefficients. These can be found using `apc.identify`.

Usage

```
apc.plot.fit(apc.fit.model,scale=FALSE,
sdv.at.zero=TRUE,type="detrend",
include.linear.plane=TRUE,
include.double.differences=TRUE,
sub.plot=NULL,main.outer=NULL,main.sub=NULL,
cex=NULL,cex.axis=NULL,cex.lab=NULL,cex.main=NULL,
cex.main.outer=1.2,
line.main=0.5,line.main.outer=NULL,
las=NULL,mar=NULL,oma=NULL,mgp=c(2,1,0),
vec.xlab=NULL)
```

Arguments

`apc.fit.model` List. See `apc.fit.model` for a description of the format.

`scale` Optional. Logical. If (TRUE) FALSE use scale of (inverse) link function. Default is FALSE.

`sdv.at.zero` Optional. Logical. If FALSE/TRUE standard deviations are plotted around estimates/zero. Default is TRUE.

`type` Optional. Character. If "detrend" double sums start and end in zero. If "sum.sum" double sums anchored as discussed in Nielsen (??). Default is "detrend".

`include.linear.plane`
Optional. Logical. If true include plots of linear plane. Default TRUE

`include.double.differences`
Optional. Logical. If true include plots of double differences. Default TRUE

`sub.plot` Optional. Character: "a","b",...,"i". Only the indicated sub plot is plotted. Default is NULL so all plots shown.

`main.outer` Optional. Character. Main title in outer margin. Default is generated internally.

`main.sub` Optional. Vector of 9 characters. Main titles for individual plots. Default is generated internally, see note 3 below.

`cex` Optional. Plot parameter, see `par`. Controls size of text. Default is NULL so that R default is used.

`cex.axis` Optional. Plot parameter, see `par`. Controls magnification of axis annotations. Default is NULL so that R default is used.

`cex.lab` Optional. Plot parameter, see `par`. Controls magnification of axis labels. Default is NULL so that R default is used.

`cex.main` Optional. Plot parameter, see `par`. Controls magnification of main title. Default is NULL so that R default is used.

`cex.main.outer` Optional. Controls magnification of outer main title if an array of plots is shown. Default is 1.2 (same as cex.main).

`line.main` Optional. Specifies the line position of main title in individual plots. Default is 0.5.

`line.main.outer`

Optional. Specifies the line position of outer main title if an array of plots is shown. Default is NULL so that R default is used.

`las` Optional. Plot parameter, see `par`. Numeric. The style of axis labels. Default is NULL so that R default is used.

`mar` Optional. Gives the number of lines of margin to be specified on the four sides of the plot. Default: `c(4,3,2,0)` for array of plots, `c(4,4,3,1)` for a single plot.

`oma` Optional. Gives the size of the outer margins in lines of text. Default: `c(0,0,5,1)` for array of plots, `c(0,0,0,0)` for a single plot.

`mgp` Optional. Plot parameter, see `par`. The margin line for the axis title, axis label and axis line. Defauls is `c(2,1,0)`, different from R default.

`vec.xlab` Optional. Controls title for xaxis. Should be a 9-vector of characters for an array of plots and a character for a single plot. As R recycles entries if a vector is too short, then `vec.xlab=""` will remove titles on x-axis. Default: NULL.

Note

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored to be zero in the three points where `age=cohort=U`, `age=U+1,cohort=U` `age=U,cohort=U+1` with `apc.fit.model$U` and where `U` is the integer value of `(per.zero+3)/2` This corresponds to the representation in Nielsen (2014b). The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=U`; an age slope, which is the difference of the values of the predictor at `age=U+1,cohort=U` and `age=cohort=U`; an cohort slope, which is the difference of the values of the predictor at `age=U,cohort=U+1` and `age=cohort=U`.

(2) The type "detrend" gives double sums that start in zero and end in zero. The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=1`, which is usually outside the index set for the data; while age and cohort slopes are adjusted for the ad hoc identification of the time effects.

(3) The default of the titles `main.sub` are generated internally depending on model specification. In the case of `model.design="APC"` and a dose-response model family the default value is `c(expression(paste("(a) ",Delta^2,alpha)),expression(paste("(b) ",Delta^2,beta)),expression(paste("(",Delta^2,gamma)),"(d) first linear trend","(e) level","(f) second linear trend",expression(paste("(g) detrended ",Sigma^2,Delta^2,alpha)),expression(paste("(h) detrended ",Sigma^2,Delta^2,beta)),expressi detrended ",Sigma^2,Delta^2,gamma)))`

(4) Default values of parameters changed (28 Sep 2020). The old appearance can be reproduced by setting `cex.lab=1.5`. For example:

`data.list <- data.Italian.bladder.cancer()`

`fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")`

`apc.plot.fit(fit.apc,cex.lab=1.5)`

The code subsumes `var.apc.plot.fit` by Zoe Fannon.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> & Zoe Fannon 28 September 2020 (12 Apr 2015).

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014b) Deviance analysis of age-period-cohort models. Work in progress.

See Also

`data.asbestos` and `data.Italian.bladder.cancer` for information on the data used in the example.

Values of coefficients can be found using `apc.identify`.

Further information on the identification in the vignette `Identification.pdf`, `Identification.R` on `Vignettes`.

Examples

```
#####################
# Example with Italian bladder cancer data
# Note that the model.design "AC" cannot be rejected against "APC"
# so there is little difference between the two plots of those fits.

data.list <- data.Italian.bladder.cancer()
apc.fit.table(data.list,"poisson.dose.response")
fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")
apc.plot.fit(fit.apc)
# now try an AC model
# can use dev.new() to see both
fit.ac <- apc.fit.model(data.list,"poisson.dose.response","AC")
apc.plot.fit(fit.ac)

# to check the numerical values for the last two rows of plots use
apc.identify(fit.ac)$coefficients.detrend

# to get only a sub plot and playing with titles
# main.outer not used with individual plot
apc.plot.fit(fit.ac,sub.plot="a",main.outer="My outer title",main.sub="My sub title")
# to play with
# titles (main.outer/main.sub),
# label orientation (las),
# axis titles (vec.xlab)
apc.plot.fit(fit.ac,main.outer="My outer title",
main.sub=c("1","2","3","4","5","6","7","8","9"),
las=1,
vec.xlab=c("a","b","c","d","e","f","g","h","i"))
```

apc.plot.fit.all *Make all fit plots.*

Description

Plots estimates using apc.plot.fit. Probability transform plot of residuals using apc.plot.fit.pt. Level plot of residuals using apc.plot.fit.residuals. Level plot of fitted values using apc.plot.fit.fitted.values. Level plot of linear predictors using apc.plot.fit.linear.predictors. Level plots of responses and rates (if dose is availble) using apc.plot.data.level.

Usage

```
apc.plot.fit.all(apc.fit.model,log ="",rotate=FALSE)
```

Arguments

apc.fit.model	List. Output from apc.fit.model. See there for a description of the format.
log	Optional plot argument. Character. "y" if y-scale is logarithmic, otherwise "". Default is "".
rotate	Optional. Logical. If TRUE rotates level plots 90 degrees clockwise (or anti-clockwise if data.format is "CL"). Default is FALSE.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 2t Apr 2015

See Also

The example below uses Italian bladder cancer data, see data.Italian.bladder.cancer

Examples

```
#####################
# EXAMPLE with Italian bladder cancer data

# get data list, then make all descriptive plots.
#  Note that warnings are given in relation to the data chosen thinning
# This can be avoided by working with the individual plots, and in particular
# with apc.plot.data.within where the thinning happens.

data.list <- data.Italian.bladder.cancer()
fit <- apc.fit.model(data.list,"poisson.dose.response","APC")
apc.plot.fit.all(fit)
```

`apc.plot.fit.pt` *Plot probability transform of responses given fitted values*

Description

Constructs probability transforms of responses given fitted values from `apc.fit.model`. The plot is given in the original coordinate system. Colours and symbols are used to indicate whether responses are central to the fitted distribution or in the tails of the fitted distribution.

Usage

```
apc.plot.fit.pt(apc.fit.model,
   do.plot=TRUE,do.value=FALSE,
   pch=c(21,24,25),
   col=c("black","green","blue","red"),
   bg=NULL,cex=NULL,main=NULL)
```

Arguments

`apc.fit.model` List. See `apc.fit.model` for a description of the format.

`do.plot` Optional. Logical. If FALSE plot is not produced. Default is TRUE.

`do.value` Optional. Logical. If TRUE value is produced. Default is FALSE.

`pch` Optional `points` argument. Numeric. Default is 21/24/25. 21 is a circle used for the central 80% of distribution. 24/25 are triangle point up/down used for right tail and left tail.

`col` Optional `plot` argument. Character or Numeric. Default is "black"/"green"/"blue"/"red". Black is use for central 80%, Green is used for 90-95% and 5-10%, Blue is used for 95-99% and 1-5%, Red is used for tails.

`bg` Optional `plot` argument. Character or Numeric. Default is bg=col.

`cex` Optional `plot` argument. Numeric. Magnification. Default is internally computed.

`main` Optional `plot` argument. Character. Main title. Default is internally computed.

Value

Vector of probability transforms. Only produced if `do.value` is set to TRUE. See example below.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 2 Dec 2013

See Also

`data.Italian.bladder.cancer` for information on the data used in the example.

Examples

```
#####################
# Example with Italian bladder cancer data
# HOW TO USE VALUE

data.list <- data.Italian.bladder.cancer()
fit <- apc.fit.model(data.list,"poisson.dose.response","APC")
v.pt <- apc.plot.fit.pt(fit,do.value=TRUE)
m.pt <- matrix(data=NA,nrow=fit$data.xmax,ncol=fit$data.ymax)
m.pt[fit$index.data] <- v.pt
m.pt

#             [,1]      [,2]       [,3]       [,4]      [,5]
#  [1,] 0.63782311 0.5651585 0.33982477 0.91299734 0.5759652
#  [2,] 0.82676269 0.8992667 0.26378120 0.28795884 0.3708787
#  [3,] 0.54139571 0.2445995 0.51923747 0.63451773 0.7955547
#  [4,] 0.87364488 0.8228499 0.07219437 0.38789788 0.5938305
#  [5,] 0.86797473 0.3934085 0.34525271 0.38955656 0.5097203
#  [6,] 0.65027598 0.8377994 0.29018594 0.03694977 0.7990229
#  [7,] 0.43769468 0.1099946 0.50261364 0.56777485 0.8916552
#  [8,] 0.67518708 0.5519831 0.67817803 0.19793887 0.5354669
#  [9,] 0.02717016 0.2066092 0.77035122 0.89047749 0.5017919
# [10,] 0.71037782 0.9464356 0.36897847 0.41790169 0.2080577
# [11,] 0.50922468 0.3085978 0.55261186 0.77592343 0.3597815
```

apc.plot.fit.residuals

Level plots of residuals / fitted values / linear predictors

Description

Level plots of residuals / fitted values / linear predictors. Returns residuals / fitted values / linear predictors as matrices when requested. The plots use apc.plot.data.level. They plot are given in the original coordinate system.

Usage

```
apc.plot.fit.residuals(apc.fit.model,
rotate=FALSE,main=NULL,lab=NULL,
contour=FALSE,colorkey=TRUE,return=FALSE)
   apc.plot.fit.fitted.values(apc.fit.model,
rotate=FALSE,main=NULL,lab=NULL,
contour=FALSE,colorkey=TRUE,return=FALSE)
   apc.plot.fit.linear.predictors(apc.fit.model,
rotate=FALSE,main=NULL,lab=NULL,
contour=FALSE,colorkey=TRUE,return=FALSE)
```

Arguments

`apc.fit.model` List. Output from `apc.fit.model`. See there for a description of the format.

`rotate` Optional. Logical. If TRUE rotates plot 90 degrees clockwise (or anti-clockwise if data.format is "CL"). Default is FALSE.

`main` Optional. Character. Main title.

`lab` Optional `plot` parameter. A numerical vector of the form c(x, y, len) which modifies the default way that axes are annotated. The values of x and y give the (approximate) number of tickmarks on the x and y axes. len is not implemented.

`contour` Optional `levelplot` (`lattice`) parameter. Logical. Contour lines drawn if TRUE. Default FALSE.

`colorkey` Optional `levelplot` (`lattice`) parameter. Logical or list. Determines color key. Default TRUE.

`return` Optional. Logical. If TRUE returns matrix with values. Default is FALSE.

Value

Matrix of the original format with residuals / fitted values /linear predictors as entries. Only produced if `return` is set to TRUE.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 26 Apr 2015

See Also

`data.Italian.bladder.cancer` for information on the data used in the example.

Examples

```
#####################
# Example with Italian bladder cancer data

data.list <- data.Italian.bladder.cancer()
fit <- apc.fit.model(data.list,"poisson.dose.response","APC")
apc.plot.fit.fitted.values(fit,return=TRUE)

#       1955-1959   1960-1964   1965-1969   1970-1974   1975-1979
# 25-29   3.04200    3.368944    2.261518    2.327538   12.000000
# 30-34  13.11980   12.835733   13.955859   10.416142    9.672462
# 35-39  24.15536   33.591644   33.388355   37.542301   26.322340
# 40-44  69.89262   68.842728   96.652963   98.478793  113.132896
# 45-49 217.97285  189.375728  189.115063  272.281239  285.255119
# 50-54 450.44864  529.823519  462.504305  469.869189  701.354350
# 55-59 724.88451  904.298410 1069.452434  969.346982  966.017661
# 60-64 877.17820 1226.088350 1532.521380 1877.331703 1807.880364
# 65-69 950.36106 1296.011123 1798.196048 2336.012274 3028.419493
# 70-74 903.94495 1187.708772 1598.021907 2302.605072 3222.719298
# 75-79 831.00000  953.055049 1280.930166 1755.788768 2678.226017
```

apc.polygon *Add connected line and standard deviation polygons to a plot*

Description

Draws a line for point forecasts and adds shaded region for forecast distribution around it. This is added to a plot in the same way as lines and polygon add lines and polygons to a plot.

Usage

```
apc.polygon(m.forecast,x.origin=1,
plot.se=TRUE,plot.se.proc=FALSE,plot.se.est=FALSE,
unit=1,
col.line=1,lty.line=1,lwd.line=1,
q.se=c(2,2,2),
angle.se=c(45,45,45),
border.se=c(NA,NA,NA),
col.se=gray(c(0.50,0.80,0.90)),
density.se=c(NULL,NULL,NULL),
lty.se=c(1,1,1))
```

Arguments

m.forecast Matrix. Up to 4 columns. Column 1: point forecasts. Column 2: forecast standard errors. Column 3: process standard errors. Column 4: estimation standard errors.

x.origin *Optional.* Numerical. x-coordinate for last observation. The first point forecast is made at x.origin+unit, where unit (with default 1) is defined in apc.data.list. Default: 1.

plot.se *Optional.* Logical. Should forecast standard errors be plotted? Default: TRUE.

plot.se.proc *Optional.* Logical. Should process standard errors be plotted? Default: FALSE.

plot.se.est *Optional.* Logical. Should estimation standard errors be plotted? Default: FALSE.

unit *Optional.* Numerical. step length for point forecasts. Default=1.

col.line *Optional.* Point forecasts: Colour of line. Same as col for lines. Default: 1.

lty.line *Optional.* Point forecasts: Type of line. Same as lty for lines. Default: 1.

lwd.line *Optional.* Point forecasts: Width of line. Same as lwd for lines. Default: 1.

q.se *Optional.* Vector of length 3. Multiplication factors for standard errors. Default: c(2,2,2).

angle.se *Optional.* Standard error polygon: 3-vector: Angle of shading. Same as angle for polygon. Default: =c(45,45,45).

border.se *Optional.* Standard error polygon: 3-vector: Border of polygon. Same as border for polygon. Default: =c(NA,NA,NA).

`col.se`	*Optional.* Standard error polygon: 3-vector: Colour of polygon. Same as `col` for `polygon`. Default: `gray(c(0.50,0.80,0.90))`.
`density.se`	*Optional.* Standard error polygon: 3-vector: Density of shading. Same as `density` for `polygon`. Default: `=c(NULL,NULL,NULL)`.
`lty.se`	*Optional.* Standard error polygon: 3-vector: Type of shading. Same as `lty` for `polygon`. Default: `=c(1,1,1)`.

Details

The empirical example of Martinez Miranda, Nielsen and Nielsen (2015) uses the data `data.asbestos`. The results of that paper are reproduced in the vignette `ReproducingMMNN2015.pdf`, `ReproducingMMNN2015.R` on `Vignettes`. The function is used there.

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 6 Jan 2016

References

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Article, Nuffield DP.

`data.aids` *UK aids data*

Description

Function that organises UK aids data in `apc.data.list` format.

The data set is taken from table 1 of De Angelis and Gilks (1994). The data are also analysed by Davison and Hinkley (1998, Example 7.4). The data are reporting delays for AIDS counting the number of cases by the date of diagnosis and length of reporting delay, measured by quarter.

The data set is in "trapezoid"-format. The original data set is unbalanced in various ways: first column covers a reporting delay of less than one month (or should it be less than one quarter?); last column covers a reporting delay of at least 14 quarters; last diagonal include incomplete counts. The default data set excludes the incomplete counts in the last diagonal, but includes the unbalanced first and last columns.

Usage

```
data.aids(all.age.groups = FALSE)
```

Arguments

`all.age.groups`	logical. If FALSE (default), the last calendar year with incomplete counts is ignored.

Value

The value is a list in `apc.data.list` format.

`response` matrix of cases

`data.format` logical equal to "trapezoid".

`age1` numeric equal to 0. This is the label for the reporting delay.

`per1` NULL. Not needed when data.format="trapezoid"

`coh1` numeric equal to 1983.5. This is the label for the diagnosis quarter (1983, third quarter).

`unit` numeric equal to 1/4. This is the width of the age and period groups.

`per.zero` numeric equal to 0.

`per.max` numeric equal to 38.

`time.adjust` numric equal to 0.

`label` character. Default data has "UK AIDS - clean".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 7 Feb 2016

Source

Table 1 of De Angelis and Gilks (1994). Also analysed by Davison and Hinkley (1998, Example 7.4).

References

De Angelis, D. and Gilks, W.R. (1994) Estimating acquired immune deficiency syndrome incidence accounting for reporting delay. *Journal of the Royal Statistical Sociey* A 157, 31-40.

Davison, A.C. and Hinkley, D.V. (1998) *Bootstrap methods and their application*. Cambridge: Cambridge University Press.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable
data <- data.Belgian.lung.cancer()
## To see the content of the data
data

#########################
# Forecast AIDS incidences by diagonsis year (cohort).
# uses as poisson response model with an AC structure
# although there is evidence of overdispersion and the
# period effect appears significant.
# The omission of the period effect follows
# Davison and Hinkley and a parsimoneous model may be
# advantageous when forecasting.
#
apc.fit.table(data.aids(),"poisson.response")
fit <- apc.fit.model(data.aids(),"poisson.response","AC")
forecast <- apc.forecast.ac(fit)
data.sums.coh <- apc.data.sums(data.aids())$sums.coh
forecast.total <- forecast$response.forecast.coh
forecast.total[,1] <- forecast.total[,1]+data.sums.coh[25:38]
x <- seq(1983.5,1992.75,by=1/4)
y <- data.sums.coh
xlab<- "diagnosis year (cohort)"
ylab<- "diagnoses"
main<- "Davison and Hinkley, Fig 7.6, parametric version"
plot(x,y,xlim=c(1988,1993),ylim=c(200,600),xlab=xlab,ylab=ylab,main=main)
apc.polygon(forecast.total,x.origin=1989.25,unit=1/4)
```

`data.asbestos` *Asbestos data*

Description

Function that organises asbestos data in `apc.data.list` format.

Counts of mesothelioma deaths in the UK by age and period. Mesothelioma is most often caused by exposure to asbestos.

The data set is in "PA"-format.

`data.asbestos` is for men 1967-2012 `data.asbestos.2013` is the same as `data.asbestos.2013.men` and is for men 1968-2013. `data.asbestos.2013.women` and is for women 1968-2013.

The primary data set includes ages 25-89, which is obtained when using the function without arguments or with argument `all.age.groups=FALSE`. The secondary data includes younger and older age groups, which is obtained when using the function with argument `all.age.groups=TRUE`. The `apc.package` is at present not aimed at such unbalanced data.

Usage

```
data.asbestos(all.age.groups = FALSE)
data.asbestos.2013(all.age.groups = FALSE)
data.asbestos.2013.women(all.age.groups = FALSE)
data.asbestos.2013.men(all.age.groups = FALSE)
```

Arguments

`all.age.groups` logical. If FALSE (default), only age groups 25-89 are included.

Value

The value is a list in `apc.data.list` format.

`response` matrix of cases. Numbers of mesothelioma deaths by period and age. Period runs 1967-2007. Age runs 25-89 when `all.age.groups=FALSE`. "PA"-format.

`dose` NULL

`data.format` logical equal to "PA". Data organised with period-groups in rows and age-groups in columns.

`age1` numeric equal to 25. This is the label for the first age group of 25.

`per1` numeric equal to 1967. This is the label for the first period group of 1967.

`coh1` NULL. Not needed when data.format="PA"

`unit` numeric equal to 1. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="PA"

`per.max` NULL. Not needed when data.format="PA"

`time.adjust` 0. Thus age=89 in period=1967 corresponds to cohort=1967-89+0=1878.

`label` character. "UK asbestos".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 30 April 2016

Source

Data were prepared for the Asbestos Working Party by the UK Health and Safety Executive. An APC analysis of these data can be found in Martinez Miranda, Nielsen and Nielsen (2015). The results of that paper are reproduced in the vignette `ReproducingMMNN2015.pdf`, `ReproducingMMNN2015.R` on `Vignettes`. These data are also used in Nielsen (2015).

The updated data set `data.asbestos.2013` is for 1968-2013 and has the same structure. This is analysed in Martinez-Miranda, Nielsen and Nielsen (2016).

References

Martinez Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2015) Inference and forecasting in the age-period-cohort model with unknown exposure with an application to mesothelioma mortality. *Journal of the Royal Statistical Society* A 178, 29-55. *Download*: Nuffield DP.

Martinez-Miranda, M.D., Nielsen, B. and Nielsen, J.P. (2016) A simple benchmark for mesothelioma projection for Great Britain. To appear in *Occupational and Environmental Medicine*. *Download*: Nuffield DP.

Nielsen, B. (2015) apc: An R package for age-period-cohort analysis. R Journal 7, 52-64. *Download*: Open access.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
# apc data list

data.list <- data.asbestos()
objects(data.list)

#####################
# Figure 1,a-c from
# Miranda Martinex, Nielsen and Nielsen (2015).

data.list <- data.asbestos()
apc.plot.data.sums(data.list,type="l")

#####################
# Figure 1,d from
# Miranda Martinex, Nielsen and Nielsen (2015).
data.list <- data.asbestos()
apc.plot.data.within(data.list,type="l",lty=1)
```

`data.Belgian.lung.cancer` *Belgian lung cancer data*

Description

Function that organises Belgian lung cancer data in `apc.data.list` format.

The data set is taken from table VIII of Clayton and Schifflers (1987a), which contains age-specific incidence rates (per 100,000 person-years observation) of lung cancer in Belgian females during the period 1955-1978. Numerators are also available. The original source was the WHO mortality database.

The data set is in "AP"-format. The original data set is unbalanced since the first four period groups cover 5 years, while the last covers 4 years. The primary data set has 4 period groups, which is obtained when using the function without arguments or with argument `unbalanced=FALSE`. The secondary data set has 5 uneven sized period groups, wwhich is obtained when using the function with argument `unbalanced=TRUE`. The `apc.package` is at present not aimed at such unbalanced data.

Usage

```
data.Belgian.lung.cancer(unbalanced = FALSE)
```

Arguments

`unbalanced` logical. If TRUE (default), the last 4-year group column of the data is ignored.

Value

The value is a list in `apc.data.list` format.

`rates` matrix of mortality rates. This is not needed for the `apc.data.list` format, but included as this is the original data formats

`response` matrix of cases

`dose` matrix of cases/rates

`data.format` logical equal to "AP". Data organised with age-groups in rows and period-groups in columns.

`age1` numeric equal to 25. This is the label for the first age group covering ages 25-29.

`per1` numeric equal to 1955. This is the label for the first period group covering period 1955-1959.

`coh1` NULL. Not needed when data.format="AP"

`unit` numeric equal to 5. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="AP"

`per.max` NULL. Not needed when data.format="AP"

`time.adjust` 0. Thus age=25 in period=1955 corresponds to cohort=1955-25+0=1930, and indeed the centers of the age and period groups, that is age=27 and period=1957 translate into cohort=1957-27+0=1930.

`label` character. "Belgian lung cancer".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (24 Oct 2013)

Source

Table VIII of Clayton and Schifflers (1987a).

References

Clayton, D. and Schifflers, E. (1987a) Models for temperoral variation in cancer rates. I: age-period and age-cohort models. *Statistics in Medicine* 6, 449-467.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.Belgian.lung.cancer()

## To see the content of the data

data
```

`data.Italian.bladder.cancer` *Italian bladder cancer data*

Description

Function that organises Italian bladder data in `apc.data.list` format.

The data set is taken from table IV of Clayton and Schifflers (1987a), which contains age-specific incidence rates (per 100,000 person-years observation) of bladder cancer in Italian males during the period 1955-1979. Numerators are also available. The original source was the WHO mortality database.

The data set is in "AP"-format.

Usage

```
data.Italian.bladder.cancer()
```

Value

The value is a list in `apc.data.list` format.

`rates` matrix of mortality rates. This is not needed for the `apc.data.list` format, but included as this is the original data formats

`response` matrix of cases

`dose` matrix of cases/rates

`data.format` logical equal to "AP". Data organised with age-groups in rows and period-groups in columns.

`age1` numeric equal to 25. This is the label for the first age group covering ages 25-29.

`per1` numeric equal to 1955. This is the label for the first period group covering period 1955-1959.

`coh1` NULL. Not needed when data.format="AP"

`unit` numeric equal to 5. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="AP"

`per.max` NULL. Not needed when data.format="AP"

`time.adjust` 0. Thus age=25 in period=1955 corresponds to cohort=1955-25+0=1930, and indeed the centers of the age and period groups, that is age=27 and period=1957 translate into cohort=1957-27+0=1930.

`label` character. "Italian bladder cancer".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (24 Oct 2013)

Source

Table IV of Clayton and Schifflers (1987a).

References

Clayton, D. and Schifflers, E. (1987a) Models for temperoral variation in cancer rates. I: age-period and age-cohort models. *Statistics in Medicine* 6, 449-467.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.Italian.bladder.cancer()

## To see the content of the data

data
```

---

data.Japanese.breast.cancer
Japanese breast cancer data

Description

Function that organises Japanese breast data in `apc.data.list` format.

The data set is taken from table I of Clayton and Schifflers (1987b), which contains age-specific mortality rates (per 100,000 person-years observation) of breast cancer in Japan, during the period 1955-1979. Reported in 5 year age groups and 5 year period groups. Numbers of cases on which rates are based are also available. The original source was WHO mortality data base.

The data set is in "AP"-format.

Usage

```
data.Japanese.breast.cancer()
```

Value

The value is a list in `apc.data.list` format.

`rates`	matrix of mortality rates. This is not needed for the `apc.data.list` format, but included as this is the original data formats
`response`	matrix of cases

`dose` matrix of cases/rates

`data.format` logical equal to "AP". Data organised with age-groups in rows and period-groups in columns.

`age1` numeric equal to 25. This is the label for the first age group covering ages 25-29.

`per1` numeric equal to 1955. This is the label for the first period group covering period 1955-1959.

`coh1` NULL. Not needed when data.format="AP"

`unit` numeric equal to 5. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="AP"

`per.max` NULL. Not needed when data.format="AP"

`time.adjust` 0. Thus age=25 in period=1955 corresponds to cohort=1955-25+0=1930, and indeed the centers of the age and period groups, that is age=27 and period=1957 translate into cohort=1957-27+0=1930.

`label` character. "Japanese breast cancer".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (24 Oct 2013)

Source

Table I of Clayton and Schifflers (1987b)

References

Clayton, D. and Schifflers, E. (1987b) Models for temperoral variation in cancer rates. II: age-period-cohort models. *Statistics in Medicine* 6, 469-481.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.Japanese.breast.cancer()

## To see the content of the data

data
```

`data.loss.BZ` *Motor data*

Description

Function that organises loss data in `apc.data.list` format.

The data set is taken from table 3.5 of Barnett & Zehnwirth (2000). Source of data unclear. It includes a run-off triangle: "response" (X) is paid amounts (units not reported) along with measures of exposure.

Data also analysed in e.g. Kuang, Nielsen, Nielsen (2011).

The data set is in "CL"-format.

At present `apc.package` does not have functions for either forecasting or for exploiting the counts. For this one can with advantage use the `DCL.package`.

Usage

```
data.loss.BZ
```

Value

The value is a list in `apc.data.list` format.

`response`	vector of paid amounts, X
`counts`	vector of number of reported claims, N
`dose`	NULL.
`data.format`	logical. Equal to "CL.vector.by.row". Data organised in vectors.
`age1`	numeric. Equal to 1.
`per1`	NULL. Not needed when data.format="CL"
`coh1`	numeric. Equal to 1.
`unit`	numeric. Equal to 1.
`per.zero`	NULL. Not needed when data.format="CL"
`per.max`	NULL. Not needed when data.format="CL"
`time.adjust`	0. Thus age=1 in cohort=1 corresponds to period=1+1-1+0=1.
`label`	character. "loss BZ".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (18 Mar 2015)

Source

Tables 1,2 of Verrall, Nielsen and Jessen (2010).

References

Barnett G, Zehnwirth B (2000) Best estimates for reserves. Proc. Casualty Actuar. Soc. 87, 245–321.

Kuang D, Nielsen B, Nielsen JP (2011) Forecasting in an extended chain-ladder-type model *Journal of Risk and Insurance* 78, 345-359

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.loss.BZ()

## To see the content of the data

data

#########################
# Fit geometric chain-ladder model

apc.fit.table(data,"log.normal.response")
```

`data.loss.TA` *Motor data*

Description

Function that organises loss data in `apc.data.list` format.

The data set is taken from Table 1 of Verrall (1991), who attributes the data to Taylor and Ashe (1983). It includes a run-off triangle: "response" (X) is paid amounts (units not reported).

Data also analysed in various papers, e.g. England and Verrall (1999).

The data set is in "CL"-format.

At present `apc.package` does not have functions for either forecasting or for exploiting the counts. For this one can with advantage use the `DCL.package`.

Usage

```
data.loss.TA
```

Value

The value is a list in apc.data.list format.

response	vector of paid amounts, X
dose	NULL.
data.format	logical. Equal to "CL.vector.by.row". Data organised in vectors.
age1	numeric. Equal to 1.
per1	NULL. Not needed when data.format="CL"
coh1	numeric. Equal to 1.
unit	numeric. Equal to 1.
per.zero	NULL. Not needed when data.format="CL"
per.max	NULL. Not needed when data.format="CL"
time.adjust	0. Thus age=1 in cohort=1 corresponds to period=1+1-1+0=1.
label	character. "loss TA".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (18 Mar 2015)

Source

Tables 1 of Verrall (1991).

References

England, P., Verrall, R.J. (1999) Analytic and bootstrap estimates of prediction errors in claims reserving Insurance: Mathematics and Economics 25, 281-293

Taylor, G.C., Ashe, F.R. (1983) Second moments of estimates of outstanding claims Journal of Econometrics 23, 37-61

Verrall, R.J. (1991) On the estimation of reserves from loglinear models Insurance: Mathematics and Economics 10, 75-80

See Also

General description of apc.data.list format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.loss.TA()

## To see the content of the data

data
```

```
#########################
# Fit chain-ladder model

apc.fit.table(data,"poisson.response")

# The overdispersed poisson model is experimental at the moment,
# so not documented
apc.fit.table(data,"od.poisson.response")
```

data.loss.VNJ *Motor data*

Description

Function that organises motor data in apc.data.list format.

The data set is taken from tables 1,2 of Verrall, Nielsen and Jessen (2010). Data from Codan, Danish subsiduary of Royal & Sun Alliance. It is a portfolio of third party liability from motor policies. The time units are in years. There are two run-off triangles: "response" (X) is paid amounts (units not reported) "counts" (N) is number of reported claims.

Data also analysed in e.g. Martinez Miranda, Nielsen, Nielsen and Verrall (2011) and Kuang, Nielsen, Nielsen (2015).

The data set is in "CL"-format.

At present apc.package does not have functions for either forecasting or for exploiting the counts. For this one can with advantage use the DCL.package.

Usage

```
data.loss.VNJ
```

Value

The value is a list in apc.data.list format.

response	vector of paid amounts, X
counts	vector of number of reported claims, N
dose	NULL.
data.format	logical. Equal to "CL.vector.by.row". Data organised in vectors.
age1	numeric. Equal to 1.
per1	NULL. Not needed when data.format="CL"
coh1	numeric. Equal to 1.
unit	numeric. Equal to 1.
per.zero	NULL. Not needed when data.format="CL"

`per.max` NULL. Not needed when data.format="CL"

`time.adjust` 0. Thus age=1 in cohort=1 corresponds to period=1+1-1+0=1.

`label` character. "loss VNJ".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 18 Mar 2015 updated 4 Jan 2016

Source

Tables 1,2 of Verrall, Nielsen and Jessen (2010).

References

Verrall R, Nielsen JP, Jessen AH (2010) Prediction of RBNS and IBNR claims using claim amounts and claim counts *ASTIN Bulletin* 40, 871-887

Martinez Miranda, M.D., Nielsen, B., Nielsen, J.P. and Verrall, R. (2011) Cash flow simulation for a model of outstanding liabilities based on claim amounts and claim numbers. *ASTIN Bulletin* 41, 107-129

Kuang D, Nielsen B, Nielsen JP (2015) The geometric chain-ladder *Scandinavian Acturial Journal* 2015, 278-300.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.loss.VNJ()

## To see the content of the data

data

#########################
# Fit chain-ladder model

fit.ac <- apc.fit.model(data,"poisson.response","AC")
fit.ac$coefficients.canonical
id.ac <- apc.identify(fit.ac)
id.ac$coefficients.dif

#########################
# Compare output with table 7.2 in
# Kuang D, Nielsen B, Nielsen JP (2015)
#                Estimate   Std. Error    z value      Pr(>|z|)
# level       13.07063963 0.0000000000        Inf  0.000000e+00
# D_age_2     -0.06543495 0.0006018694 -108.71950  0.000000e+00
# D_age_3     -0.80332424 0.0008757527 -917.29576  0.000000e+00
# D_age_4     -0.41906516 0.0012294722 -340.84965  0.000000e+00
# D_age_5     -0.29097802 0.0015627740 -186.19329  0.000000e+00
# D_age_6     -0.57299006 0.0021628918 -264.91850  0.000000e+00
# D_age_7     -0.36101594 0.0030016569 -120.27222  0.000000e+00
# D_age_8     -0.62706059 0.0046139466 -135.90547  0.000000e+00
# D_age_9      0.12160793 0.0061126021   19.89463  4.529830e-88
# D_age_10    -2.59708012 0.0245028290 -105.99103  0.000000e+00
# D_cohort_2  -0.02591843 0.0009037977  -28.67724 7.334840e-181
# D_cohort_3   0.18973130 0.0011301184  167.88621  0.000000e+00
# D_cohort_4   0.12354693 0.0010508785  117.56539  0.000000e+00
# D_cohort_5  -0.10114701 0.0010566534  -95.72392  0.000000e+00
# D_cohort_6   0.03594882 0.0010913718   32.93912 6.056847e-238
# D_cohort_7  -0.17175409 0.0011676536 -147.09336  0.000000e+00
# D_cohort_8   0.20671145 0.0012098255  170.86055  0.000000e+00
# D_cohort_9   0.04056617 0.0012325163   32.91329 1.418555e-237
# D_cohort_10  0.06876759 0.0015336998   44.83771  0.000000e+00

#########################
# Get deviance table.
# APC strongly rejected => overdispersion?
#   AC (Chain-ladder) rejected against APC (inference invalid anyway)
#    => one should be careful with distribution forecasts

apc.fit.table(data,"poisson.response")

#########################
#         -2logL df.residual prob(>chi_sq)  LR.vs.APC df.vs.APC prob(>chi_sq)        aic
# APC   176030.0          28             0         NA        NA            NA   176841.7
# AP    305784.6          36             0   129754.6         8             0   306580.3
# AC    374155.2          36             0   198125.2         8             0   374950.9
# PC    553555.1          36             0   377525.0         8             0   554350.7
# Ad    486013.4          44             0   309983.4        16             0   486793.0
# Pd    710009.6          44             0   533979.6        16             0   710789.3
# Cd    780859.4          44             0   604829.4        16             0   781639.1
# A     575389.6          45             0   399359.6        17             0   576167.3
# P    9483688.1          45             0  9307658.0        17             0  9484465.7
# C    7969034.0          45             0  7793004.0        17             0  7969811.7
# t     898208.1          52             0   722178.1        24             0   898971.7
# tA    987389.4          53             0   811359.4        25             0   988151.1
# tP   9690623.4          53             0  9514593.4        25             0  9691385.1
# tC   8079187.6          53             0  7903157.6        25             0  8079949.3
# 1   10815443.5          54             0 10639413.5        26             0 10816203.2


#########################
# Fit geometric chain-ladder model

fit.ac <- apc.fit.model(data,"log.normal.response","AC")
fit.ac$coefficients.canonical
id.ac <- apc.identify(fit.ac)
id.ac$coefficients.dif
```

```
#########################
# Compare output with table 7.2 in
# Kuang D, Nielsen B, Nielsen JP (2015)
#                  Estimate Std. Error     t value     Pr(>|t|)
# level       13.0846325168  0.1322711 98.92285585 0.000000e+00
# D_age_2     -0.0721758004  0.1291053 -0.55904595 5.761304e-01
# D_age_3     -0.8180698189  0.1350216 -6.05880856 1.371335e-09
# D_age_4     -0.3945325384  0.1433094 -2.75301253 5.904964e-03
# D_age_5     -0.3354312554  0.1538274 -2.18056918 2.921530e-02
# D_age_6     -0.6322104515  0.1673396 -3.77800844 1.580875e-04
# D_age_7     -0.3020293471  0.1854134 -1.62895114 1.033234e-01
# D_age_8     -0.5225495852  0.2112982 -2.47304367 1.339678e-02
# D_age_9      0.0078494549  0.2531172  0.03101115 9.752607e-01
# D_age_10    -2.5601846890  0.3415805 -7.49511273 6.624141e-14
# D_cohort_2  -0.1025686798  0.1291053 -0.79445748 4.269292e-01
# D_cohort_3   0.0820931043  0.1350216  0.60799994 5.431875e-01
# D_cohort_4   0.3800465893  0.1433094  2.65193088 8.003292e-03
# D_cohort_5  -0.0920821506  0.1538274 -0.59860701 5.494350e-01
# D_cohort_6  -0.0530061052  0.1673396 -0.31675768 7.514275e-01
# D_cohort_7  -0.2053813051  0.1854134 -1.10769405 2.679940e-01
# D_cohort_8   0.2705853742  0.2112982  1.28058555 2.003393e-01
# D_cohort_9  -0.0009224552  0.2531172 -0.00364438 9.970922e-01
# D_cohort_10  0.0736954734  0.3415805  0.21574845 8.291838e-01

#########################
# Get deviance table.
# AC marginally rejected against APC

apc.fit.table(data,”log.normal.response”)

#########################
#      -2logL df.residual LR.vs.APC df.vs.APC prob(>chi_sq)     aic
# APC -28.528          28        NA        NA            NA  27.472
# AP   -3.998          36    24.530         8         0.002  36.002
# AC   -9.686          36    18.842         8         0.016  30.314
# PC   31.722          36    60.250         8         0.000  71.722
# Ad    6.251          44    34.779        16         0.004  30.251
# Pd   41.338          44    69.866        16         0.000  65.338
# Cd   38.919          44    67.447        16         0.000  62.919
# A    12.765          45    41.292        17         0.001  34.765
# P   171.283          45   199.811        17         0.000 193.283
# C   162.451          45   190.979        17         0.000 184.451
# t    46.300          52    74.827        24         0.000  54.300
# tA   49.541          53    78.069        25         0.000  55.541
# tP  171.770          53   200.298        25         0.000 177.770
# tC  163.280          53   191.808        25         0.000 169.280
# 1   182.166          54   210.694        26         0.000 186.166
```

data.loss.XL *US Casualty data, XL Group*

Description

Function that organises US Casualty data from XL Group in `apc.data.list` format.

The data set is taken from table 1.1 Kuang and Nielsen (2020). Data are for US Casualty data from the XL Group. They are gross paid and reported loss and allocated loss adjustment expense in 1000 USD.

The data set is in "CL"-format.

Usage

```
data.loss.XL
```

Value

The value is a list in `apc.data.list` format.

`response`	matrix of paid amounts, incremental
`dose`	NULL.
`data.format`	logical. Equal to "CL".
`age1`	numeric. Equal to 1.
`per1`	NULL. Not needed when data.format="CL"
`coh1`	numeric. Equal to 1997.
`unit`	numeric. Equal to 1997.
`per.zero`	NULL. Not needed when data.format="CL"
`per.max`	NULL. Not needed when data.format="CL"
`time.adjust`	-1996. Thus age=1 in cohort=1997 corresponds to period=1997+1997-1+(-1996)=1997.
`label`	character. "loss, US casualty, XL Group".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 26 August 2020 (10 Mar 2018)

Source

Table 1.1 of Kuang and Nielsen (2020) and in turn *download*: xls file from XL Group files.

References

Kuang, D. and Nielsen B. (2020) Generalized log-normal chain-ladder. *Scandinavian Actuarial Journal* 2020, 553-576. *Download*: Open access. Earlier version: Nuffield DP.

See Also

General description of `apc.data.list` format.

For explanation for Chain Ladder forecast, see `apc.forecast.ac`.

The analysis in Kuang and Nielsen (2020) is reproduced in the vignette `ReproducingKN2020.pdf`, `ReproducingKN2020.R` on `Vignettes`.

Examples

```
#########################
## It is convenient to construct a data variable for paid data

data <- data.loss.XL()
## To see the content of the data
data

#########################
# Get deviance table.
# reproduce Table 4.1 in Kuang and Nielsen (2018).

apc.fit.table(data,”log.normal.response”)
apc.fit.table(data,”log.normal.response”,model.design.reference=”AC”)

#########################
# > apc.fit.table(data,”log.normal.response”)
#      -2logL df.residual LR vs.APC df vs.APC prob(>chi_sq) F vs.APC prob(>F)     aic
# APC 170.003         153       NaN       NaN           NaN      NaN      NaN 286.003
# AP  243.531         171    73.527        18         0.000    3.564    0.000 323.531
# AC  179.873         171     9.869        18         0.936    0.409    0.984 259.873
# PC  633.432         171   463.428        18         0.000   68.736    0.000 713.432
# Ad  258.570         189    88.567        36         0.000    2.230    0.000 302.570
# Pd  643.892         189   473.888        36         0.000   36.340    0.000 687.892
# Cd  649.142         189   479.139        36         0.000   37.368    0.000 693.142
# A   357.359         190   187.355        37         0.000    5.956    0.000 399.359
# P   644.176         190   474.172        37         0.000   35.412    0.000 686.176
# C   672.392         190   502.388        37         0.000   41.099    0.000 714.392
# t   664.488         207   494.484        54         0.000   27.015    0.000 672.488
# tA  681.993         208   511.989        55         0.000   29.072    0.000 687.993
# tP  664.746         208   494.742        55         0.000   26.560    0.000 670.746
# tC  686.181         208   516.178        55         0.000   29.713    0.000 692.181
# 1   690.399         209   520.396        56         0.000   29.830    0.000 694.399
#
# > apc.fit.table(data,”log.normal.response”,model.design.reference=”AC”)
#     -2logL df.residual LR vs.AC df vs.AC prob(>chi_sq) F vs.AC prob(>F)     aic
# AC 179.873         171      NaN      NaN           NaN     NaN      NaN 259.873
# Ad 258.570         189   78.698       18             0   4.319        0 302.570
# Cd 649.142         189  469.269       18             0  79.257        0 693.142
# A  357.359         190  177.486       19             0  11.955        0 399.359
# C  672.392         190  492.519       19             0  84.930        0 714.392
# t  664.488         207  484.615       36             0  42.993        0 672.488
# tA 681.993         208  502.120       37             0  45.869        0 687.993
# tC 686.181         208  506.308       37             0  46.886        0 692.181
# 1  690.399         209  510.526       38             0  46.670        0 694.399


#########################
# Fit log normal chain-ladder model
# reproduce Table 4.2 in Kuang and Nielsen (2018).

fit.ac <- apc.fit.model(data,”log.normal.response”,”AC”)
```

```
id.ac <- apc.identify(fit.ac)
id.ac$coefficients.dif
fit.ac$s2
fit.ac$RSS

#########################
# > id.ac$coefficients.dif
#                   Estimate Std. Error     t value     Pr(>|t|)
# level          7.660055032  0.1377951 55.59016605 0.000000e+00
# D_age_1998     2.272100342  0.1335080 17.01846386 5.992216e-65
# D_age_1999     0.932530550  0.1362610  6.84370899 7.716860e-12
# D_age_2000     0.235606356  0.1398301  1.68494782 9.199864e-02
# D_age_2001     0.088886609  0.1438733  0.61781154 5.366996e-01
# D_age_2002    -0.176044303  0.1483681 -1.18653717 2.354102e-01
# D_age_2003    -0.144445459  0.1533567 -0.94189218 3.462478e-01
# D_age_2004    -0.427608601  0.1589136 -2.69082462 7.127565e-03
# D_age_2005    -0.300527594  0.1651428 -1.81980421 6.878883e-02
# D_age_2006    -0.399729999  0.1721838 -2.32153023 2.025824e-02
# D_age_2007    -0.189656058  0.1802245 -1.05233225 2.926471e-01
# D_age_2008    -0.242063670  0.1895226 -1.27722853 2.015216e-01
# D_age_2009    -0.260459607  0.2004421 -1.29942545 1.937980e-01
# D_age_2010    -0.555317528  0.2135164 -2.60081872 9.300158e-03
# D_age_2011    -0.303234088  0.2295651 -1.32090683 1.865324e-01
# D_age_2012     0.405830766  0.2499291  1.62378389 1.044219e-01
# D_age_2013    -0.895278068  0.2769988 -3.23206421 1.228994e-03
# D_age_2014     0.116668873  0.3156054  0.36966685 7.116307e-01
# D_age_2015    -0.383048241  0.3777268 -1.01408813 3.105407e-01
# D_age_2016    -0.273419402  0.5083832 -0.53782152 5.907003e-01
# D_cohort_1998  0.288755900  0.1335080  2.16283663 3.055375e-02
# D_cohort_1999  0.163424236  0.1362610  1.19934721 2.303930e-01
# D_cohort_2000 -0.264981486  0.1398301 -1.89502518 5.808907e-02
# D_cohort_2001  0.149829430  0.1438733  1.04139815 2.976908e-01
# D_cohort_2002 -0.374386828  0.1483681 -2.52336417 1.162380e-02
# D_cohort_2003 -0.198735893  0.1533567 -1.29590632 1.950078e-01
# D_cohort_2004 -0.008807130  0.1589136 -0.05542087 9.558032e-01
# D_cohort_2005 -0.005337953  0.1651428 -0.03232325 9.742143e-01
# D_cohort_2006 -0.132272851  0.1721838 -0.76820710 4.423642e-01
# D_cohort_2007 -0.021862643  0.1802245 -0.12130783 9.034472e-01
# D_cohort_2008 -0.472602270  0.1895226 -2.49364600 1.264386e-02
# D_cohort_2009 -0.437572798  0.2004421 -2.18303804 2.903301e-02
# D_cohort_2010  0.295511564  0.2135164  1.38402260 1.663515e-01
# D_cohort_2011  0.310545832  0.2295651  1.35275725 1.761332e-01
# D_cohort_2012 -0.268692406  0.2499291 -1.07507473 2.823413e-01
# D_cohort_2013  0.142131410  0.2769988  0.51311192 6.078730e-01
# D_cohort_2014  0.201777590  0.3156054  0.63933494 5.226051e-01
# D_cohort_2015 -0.092672697  0.3777268 -0.24534320 8.061907e-01
# D_cohort_2016  0.872997251  0.5083832  1.71720334 8.594203e-02
# > fit.ac$s2
# [1] 0.1693316
# > fit.ac$RSS
# [1] 28.9557
# > fit.ac$RSS
```

```
forecast <- apc.forecast.ac(fit.ac,quantiles=c(0.995))
forecast$response.forecast.coh

#########################
# > forecast$response.forecast.coh
#   forecast         se    se.proc      se.est     t-0.995
# coh_2    1871.073   1026.463   707.4405    743.7428    4544.891
# coh_3    5099.330   1874.681  1375.8435   1273.3744    9982.659
# coh_4    7171.317   2123.128  1622.5220   1369.3412   12701.822
# coh_5   11699.350   2984.949  2274.8292   1932.6338   19474.801
# coh_6   13717.388   3345.138  2654.4080   2035.6984   22431.090
# coh_7   14343.522   3188.410  2471.3130   2014.5886   22648.964
# coh_8   18377.001   3834.057  2910.9751   2495.2390   28364.281
# coh_9   25488.052   5241.618  3976.5389   3414.9225   39141.867
# coh_10  30524.942   6213.652  4662.3320   4107.5694   46710.794
# coh_11  40078.245   8115.990  5976.5789   5490.8835   61219.471
# coh_12  32680.319   6603.511  4727.4210   4610.6241   49881.712
# coh_13  28509.077   5895.265  4143.1332   4193.8760   43865.568
# coh_14  51760.526  11013.030  7540.3989   8026.7807   80448.208
# coh_15  98747.731  22063.641 14798.3216  16365.0210  156220.991
# coh_16 100330.677  23254.845 14704.7084  18015.5316  160906.889
# coh_17 149813.314  36629.836 21310.2885  29792.8931  245229.846
# coh_18 221549.649  58610.037 29815.3239  50459.7158  374222.093
# coh_19 229480.904  69931.745 29102.9866  63588.2473  411645.102
# coh_20 575343.178 235016.967 70362.1087 224236.8135 1187535.497
```

data.RH.mortality *2-sample mortality data.*

Description

Function that organises mortality data from Riebler and Held (2010) in `apc.data.list` format.

The data set is taken from the supplementary data of Riebler and Held (2010). Mortality data for women in Denmark and Norway

The original source was Jacobsen et al. (2004).

The data set is in "AP"-format.

Usage

```
data.RH.mortality.dk()
data.RH.mortality.no()
```

Value

The value is a list in `apc.data.list` format.

`response`	matrix of cases

`dose` matrix of cases/rates

`data.format` logical equal to "AP". Data organised with age-groups in rows and period-groups in columns.

`age1` numeric equal to 0.

`per1` numeric equal to 1960.

`coh1` NULL. Not needed when data.format="AP"

`unit` numeric equal to 5. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="AP"

`per.max` NULL. Not needed when data.format="AP"

`time.adjust` 0. Thus age=0 in period=1960 corresponds to cohort=1960-0+0=1960, and indeed the centers of the age and period groups, that is age=2 and period=1962 translate into cohort=1962-2+0=1960.

`label` character. "RH mortality Denmark" or "RH mortality Norway".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 17 Sep 2016

Source

Riebler and Held (2010), supplementary material.

References

Jacobsen, R, von Euler, M, Osler, M, Lynge, E and Keiding, N (2004) Women's death in Scandinavia - what makes Denmark different? *European Journal of Epidemiology* 19, 117-121.

Riebler, A and Held, L. (2010) The analysis of heterogeneous time trends in multivariate age-period-cohort models. *Biostatistics* 11, 57–59. *Download*: Open access, Supplementary material.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.US.prostate.cancer()

## To see the content of the data

data
```

`data.US.prostate.cancer`

Japanese breast cancer data

Description

Function that organises US prostate data in `apc.data.list` format.

The data set is taken from table 2 of Holford (1983), which contains age-specific counts of deaths and midperiod population measured in 1000s, during the period 1935-1969. Reported in 5 year age groups and 5 year period groups.

The original source was Cancer deaths: National Center for Health Statistics, 1937-1973 Population 1935-60: Grove and Hetzel, 1968 Population 1960-69: Bureau of the Census, 1974

The data set is in "AP"-format.

Usage

```
data.US.prostate.cancer()
```

Value

The value is a list in `apc.data.list` format.

`response` matrix of cases

`dose` matrix of cases/rates

`data.format` logical equal to "AP". Data organised with age-groups in rows and period-groups in columns.

`age1` numeric equal to 50. This is the label for the first age group covering ages 25-29.

`per1` numeric equal to 1935. This is the label for the first period group covering period 1955-1959.

`coh1` NULL. Not needed when data.format="AP"

`unit` numeric equal to 5. This is the width of the age and period groups.

`per.zero` NULL. Not needed when data.format="AP"

`per.max` NULL. Not needed when data.format="AP"

`time.adjust` 0. Thus age=50 in period=1935 corresponds to cohort=1935-50+0=1885, and indeed the centers of the age and period groups, that is age=52 and period=1937 translate into cohort=1937-52+0=1885.

`label` character. "US prostate cancer".

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 8 Sep 2015 (28 Apr 2015)

Source

Table 2 of Holford (1983)

References

Holford, T.R. (1983) The estimation of age, period and cohort effects for vital rates. *Biometrics* 39, 311-324.

See Also

General description of `apc.data.list` format.

Examples

```
#########################
## It is convient to construct a data variable

data <- data.US.prostate.cancer()

## To see the content of the data

data
```

`new.apc.identify` *Identification of time effects*

Description

Computes ad hoc identified time effects.

Usage

```
new.apc.identify(apc.fit.model)
```

Arguments

`apc.fit.model` List. See `apc.fit.model` for a description of the format.

Details

Forms ad hoc identified time effects from the canonical parameter. These are used either indirectly by `apc.plot.fit` or they are computed directly with this command.

The ad hoc identifications are based on Nielsen (2014b). For details see also the vignette `Identification.pdf`, `Identification.R` on `Vignettes` or in the notes below.

For model designs of any type two ad hoc identified time effects.

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored in the middle of the first period diagonal.

(2) The type "detrend" gives double sums that start in zero and end in zero.

For model designs with only two time effects, that is "AC", "AP", "PC" there is a further ad hoc identification.

(3) The type "demean" gives single sums of single differences. Derived from "detrend" where the linear trends are attributed to the double sums of double differences. Level unchanged.

(4) The type "dif" gives the single differences derived from "demean". Could also have been chosen as canonical parametrisation for these models.

Value

`index.age.max` Vector. Indices for age parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.max` Vector. Indices for period parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.max` Vector. Indices for cohort parameters when using coefficients.ssdd or coefficients.detrend. The length is two longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.max` Vector. Indicates the dates for the parameters when using coefficients.ssdd or coefficients.detrend. The length is six longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`index.age.sub` * Vector. Indices for age parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.sub` * Vector. Indices for period parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.sub` * Vector. Indices for cohort parameters when using coefficients.demean. The length is two longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.sub` * Vector. Indicates the dates for the parameters when using coefficients.demean. The length is six longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`index.age.dif` * Vector. Indices for age parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.age` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.per.dif` * Vector. Indices for period parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.per` if `model.design` is "APC. NULL if age double differences are not estimated.

`index.coh.dif` * Vector. Indices for cohort parameters when using coefficients.dif. The length is one longer that that of `apc.model.fit$index.coh` if `model.design` is "APC. NULL if age double differences are not estimated.

`dates.dif` * Vector. Indicates the dates for the parameters when using coefficients.dif. The length is three longer that that of `apc.model.fit$index.coh` if `model.design` is "APC.

`coefficients.ssdd`
: Matrix. Coefficients of the double sum of double differences. Normalised to be zero at two values chosen so age=cohort and period is at the minimal value. For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.ssdd`
: Matrix. Estimated covariance matrix for double sums.

`coefficients.detrend`
: Matrix. Coefficients of the double sum of double differences. Normalised to be zero for first and last value. For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.detrend`
: Matrix. Estimated covariance matrix for detrended double sums.

`coefficients.demean`
: * Matrix. Coefficients of the sum of differences. Normalised to be zero for first value. Does not apply is design is "APC" For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.demean`
: * Matrix. Estimated covariance matrix for demeaned sums.

`coefficients.dif`
: * Matrix. Coefficients of the differences. Does not apply is design is "APC" For each parameter is reported coefficient, standard deviation, z-value, which is the ratio of those, and p-value.

`covariance.dif`
: * Matrix. Estimated covariance matrix for differences.

Note

* indicates that values only implemented for designs "AC", "AP", "PC".

The differences are not identified for design "APC". An arbitrary level can be moved between differences for age, period and cohort.

The differences are not identified for designs "Ad", "Pd", "Cd". These models have two linear trends and one set of double differences. In the model "Ad", as an example, one linear trend will be associated with age, but it is arbitrary whether the second linear trend should be associated with period or cohort. The slope of the age trend will depend on that arbitrary choice. In turn the level of the age differences will be arbitrary.

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored to be zero in the three points where `age=cohort=U`, `age=U+1,cohort=U` `age=U,cohort=U+1` with `apc.fit.model$U` and where `U` is the integer value of `(per.zero+3)/2` This corresponds to the representation in Nielsen (2014b). The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=U`; an age slope, which is the difference of the values of the predictor at `age=U+1,cohort=U` and `age=cohort=U`; an cohort slope, which is the difference of the values of the predictor at `age=U,cohort=U+1` and `age=cohort=U`.

(2) The type "detrend" gives double sums that start in zero and end in zero. The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=1`, which is usually outside the index set for the data; while age and cohort slopes are adjusted for the ad hoc identification of the time effects.

(3) Subsumes `var.apc.identify` from `apc.indiv` (25 Sep 2020)

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> & Zoe Fannon 25 Sep 2020 (12 Apr 2015)

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014b) Deviance analysis of age-period-cohort models. Work in progress.

See Also

The vignette Identification.pdf.

Examples

```
########################
# Belgian lung cancer
#  first an example with APC design, note that demean and dif not defined.

data.list <- data.Belgian.lung.cancer()

fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")
fit.apc$coefficients.canonical
id.apc <- apc.identify(fit.apc)
id.apc$coefficients.ssdd
id.apc$coefficients.detrend
id.apc$coefficients.demean
id.apc$coefficients.dif

fit.ap <- apc.fit.model(data.list,"poisson.dose.response","AP")
fit.ap$coefficients.canonical
id.ap <- apc.identify(fit.ap)
id.ap$coefficients.ssdd
id.ap$coefficients.detrend
id.ap$coefficients.demean
id.ap$coefficients.dif
```

`new.apc.plot.fit` *Plots of apc estimates*

Description

Functions to plot the apc estimates found by `apc.fit.model`. The function apc.plot.fit detects the type of `model.design` and `model.family` from the fit values and makes appropriate plots.

Depending on the `model.design` the plot has up to 9 sub plots. The type of these can be chosen using `type`

Model designs of any type. If `type` is "detrend" or "sum.sum" the canonical age period cohort parametrisation is used. This involves double differences of the time effects. The first row of plots are double differences of the time effects. The next two rows of plots illustrate the representation theorem depending on the choice of `type`. In both cases the sum of the plots add up to the predictor.

"detrend" The last row of plots are double sums of double differences detrend so that that each series starts in zero and ends in zero. The corresponding level and (up to) two linear trends are shown in the middle row of plots. The linear trends are identified to be 0 for age, period or cohort equal to its smallest value. See note 2 below.

"sum.sum" The last row of plots are double sums of double differences anchored as in the derivation of Nielsen (2014b). The corresponding level and (up to) two linear trends are shown in the middle row of plots. The linear trends are identified to be 0 for the anchoring point U of age, period or cohort as described in Nielsen (2014b). See note 1 below.

Model designs with 2 factors. If `type` is "dif" the canonical two factor parametrisation is used. This involves single differences. It is only implemented for `model.design` of "AC", "AP", "PC". It does not apply for `model.design` of "APC" because single differences are not identified. It does not apply for the drift models where `model.design` is "Ad", "Pd", "Cd", "t" because it is not clear which time scale the second linear trend should be attributed to. It is not implemented for `model.design` of "tA, "tP", "tC", "1". The first row of plots are single differences of the time effects. The next two rows of plots illustrate the representation theorem. In the second row the level is given and in the third row plots of single sums of single differences are given, normalised to start in zero.

Appearance may vary. Note, the plots "detrend" and "dif" can give very different appearance of the time effects. The "dif" plots are dominated by linear trends. They can therefore be more difficult to interpret than the "detrend" plots, where linear trends are set aside.

Standard deviations. All plots include plots of 1 and 2 standard deviations. The only exception is the intercept in the case `model.family` is "poisson.response" as this uses a multinomial sampling scheme, where the intercept is set to increase in the asymptotic experiment. The default is to plot standard deviations around zero, so that they represent a test for zero values of the parameters. Using the argument `sdv.at.zero` the standard deviations can be centered around the estimates. This can give a very complicated appearance.

Values of coefficients. These can be found using `apc.identify`.

Usage

```
new.apc.plot.fit(apc.fit.model,scale=FALSE,
sdv.at.zero=TRUE,type="detrend",
include.linear.plane=TRUE,
include.double.differences=TRUE,
sub.plot=NULL,main.outer=NULL,main.sub=NULL,
cex=NULL,cex.axis=NULL,cex.lab=NULL,cex.main=NULL,
cex.main.outer=1.2,
line.main=0.5,line.main.outer=NULL,
mar=NULL,oma=NULL,mgp=c(2,1,0))
```

Arguments

`apc.fit.model` List. See `apc.fit.model` for a description of the format.

`scale` Optional. Logical. If (TRUE) FALSE use scale of (inverse) link function. Default is FALSE.

`sdv.at.zero` Optional. Logical. If FALSE/TRUE standard deviations are plotted around estimates/zero. Default is TRUE.

`type` Optional. Character. If "detrend" double sums start and end in zero. If "sum.sum" double sums anchored as discussed in Nielsen (??). Default is "detrend".

`include.linear.plane`
Optional. Logical. If true include plots of linear plane. Default TRUE

`include.double.differences`
Optional. Logical. If true include plots of double differences. Default TRUE

`sub.plot` Optional. Character: "a","b",...,"i". Only the indicated sub plot is plotted. Default is NULL so all plots shown.

`main.outer` Optional. Character. Main title in outer margin. Default is generated internally.

`main.sub` Optional. Vector of 9 characters. Main titles for individual plots. Default is generated internally, see note 3 below.

`cex` Optional. Plot parameter, see `par`. Controls size of text. Default is NULL so that R default is used.

`cex.axis` Optional. Plot parameter, see `par`. Controls magnification of axis annotations. Default is NULL so that R default is used.

`cex.lab` Optional. Plot parameter, see `par`. Controls magnification of axis labels. Default is NULL so that R default is used.

`cex.main` Optional. Plot parameter, see `par`. Controls magnification of main title. Default is NULL so that R default is used.

`cex.main.outer` Optional. Controls magnification of outer main title if an array of plots is shown. Default is 1.2 (same as cex.main).

`line.main` Optional. Specifies the line position of main title in individual plots. Default is 0.5.

`line.main.outer`
Optional. Specifies the line position of outer main title if an array of plots is shown. Default is NULL so that R default is used.

`mar` Optional. Gives the number of lines of margin to be specified on the four sides of the plot. Default: `c(4,3,2,0)` for array of plots, `c(4,4,3,1)` for a single plot.

`oma` Optional. Gives the size of the outer margins in lines of text. Default: `c(0,0,5,1)` for array of plots, `c(0,0,0,0)` for a single plot.

`mgp` Optional. Plot parameter, see `par`. The margin line for the axis title, axis label and axis line. Defauls is `c(2,1,0)`, different from R default.

Note

(1) The type "sum.sum" (same as "ss.dd") gives double sums anchored to be zero in the three points where `age=cohort=U`, `age=U+1,cohort=U` `age=U,cohort=U+1` with `apc.fit.model$U` and where `U` is the integer value of `(per.zero+3)/2` This corresponds to the representation in Nielsen (2014b). The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=U`; an age slope, which is the difference of the values of the predictor at `age=U+1,cohort=U` and `age=cohort=U`; an cohort slope, which is the difference of the values of the predictor at `age=U,cohort=U+1` and `age=cohort=U`.

(2) The type "detrend" gives double sums that start in zero and end in zero. The linear plane is parametrised in terms of a level, which is the value of the predictor at `age=cohort=1`, which is usually outside the index set for the data; while age and cohort slopes are adjusted for the ad hoc identification of the time effects.

(3) The default of the titles `main.sub` are generated internally depending on model specification. In the case of `model.design="APC"` and a dose-response model family the default value is `c(expression(paste("(a) ",Delta^2,alpha)),expression(paste("(b) ",Delta^2,beta)),expression(paste("(` `",Delta^2,gamma)),"(d) first linear trend","(e) level","(f) second linear trend",expression(paste("(g)` `detrended ",Sigma^2,Delta^2,alpha)),expression(paste("(h) detrended ",Sigma^2,Delta^2,beta)),expressi` `detrended ",Sigma^2,Delta^2,gamma)))`

(4) Default values of parameters changed (25 Sep 2020). The old appearance can be reproduced by setting `cex.lab=1.5`. For example:

```
data.list <- data.Italian.bladder.cancer()

fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")

apc.plot.fit(fit.apc,cex.lab=1.5)
```

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 12 Apr 2015 updated 24 September 2020 vs 2.0.0. Subsumes `var.apc.plot.fit` by Zoe Fannon.

References

Kuang, D., Nielsen, B. and Nielsen, J.P. (2008a) Identification of the age-period-cohort model and the extended chain ladder model. Biometrika 95, 979-986. *Download*: Article; Earlier version Nuffield DP.

Nielsen, B. (2014b) Deviance analysis of age-period-cohort models. Work in progress.

See Also

`data.asbestos` and `data.Italian.bladder.cancer` for information on the data used in the example.

Values of coefficients can be found using `apc.identify`.

Further information on the identification in the vignette `Identification.pdf`, `Identification.R` on `Vignettes`.

Examples

```
#####################
# Example with Italian bladder cancer data
# Note that the model.design "AC" cannot be rejected against "APC"
# so there is little difference between the two plots of those fits.

data.list <- data.Italian.bladder.cancer()
apc.fit.table(data.list,"poisson.dose.response")
fit.apc <- apc.fit.model(data.list,"poisson.dose.response","APC")
new.apc.plot.fit(fit.apc)
# now try an AC model
# can use dev.new() to see both
fit.ac <- apc.fit.model(data.list,"poisson.dose.response","AC")
new.apc.plot.fit(fit.ac)

# to check the numerical values for the last two rows of plots use
new.apc.identify(fit.ac)$coefficients.detrend

# to get only a sub plot and playing with titles
# main.outer not used with individual plot
new.apc.plot.fit(fit.ac,sub.plot="a",main.outer="My outer title",main.sub="My sub title")
# to get only a all plots and playing with titles
new.apc.plot.fit(fit.ac,main.outer="My outer title",main.sub=c("1","2","3","4","5","6","7","8","9"))
```

`triangle` *Triangular matrices used in reserving*

Description

Triangular matrices are used for reserving in general insurance. A matrix is triangular if it is square and it has NAs in lower triangle where row+col>dim. The apc package uses incremental triangles.

The function `is.triangle` tests if an object is a triangular matrix.

The function `triangle.cumulative` forms the cumulative version of an incremental matrix by taking partial sums in each row.

The function `triangle.incremental` forms the incremental version of an cumulative matrix by taking differences in each row.

The function `vector.2.triangle` turns a k*(k+1)/2 vector into a triangular matrix of dimension k.

Usage

```
is.triangle(m)
triangle.cumulative(m)
triangle.incremental(m)
vector.2.triangle(v,k)
```

Arguments

v	vector. Length k*(k+1)/2
k	integer. Dimension
m	matrix. Square matrix

Author(s)

Bent Nielsen <bent.nielsen@nuffield.ox.ac.uk> 21 Nov 2019 (7 Feb 2015)

Examples

```
#########################

m <- vector.2.triangle(1:10,4)
m
is.triangle(m)
triangle.cumulative(m)
triangle.incremental(m)
```

Index