CoDa R Package

Marius Pascariu

2018-07-18

Description

The CoDa R package contains the implementation of the Compositional Data Mortality Model (CoDa). This is a Lee-Carter (1992) type method that is used to modelling and forecasting the life table distribution of deaths (dx) using Principal Component Analysis. In the context of mortality forecasting the CoDa method was fist used in Bergeron-Boucher et al. (2017). The package includes functions for fitting the model, analysing it's goodness-of-fit and performing mortality projections.

Installation

1. Make sure you have the most recent version of R
2. Run the following code in your R console

```
install.packages("CoDa")
```

Updating to the latest version of the package

You can track and contribute to the development of `CoDa` on GitHub. To install it:

1. Install the release version of `devtools` from CRAN with `install.packages("devtools")`.

2. Make sure you have a working development environment.

   - **Windows**: Install Rtools.
   - **Mac**: Install `Xcode` from the Mac App Store.
   - **Linux**: Install a compiler and various development libraries (details vary across different flavors of Linux).

3. Install the development version of `CoDa`.

```
devtools::install_github("mpascariu/CoDa")
```

Help

All functions are documented in the standard way, which means that once you load the package using `library(CoDa)` you can just type `?coda` to see the help file.

Examples

Fit CoDa model

```
rm(list = ls())
library(CoDa)
```

```
M <- coda(CoDa.data, x = 0:110, y = 1960:2014)
M
```

```
##
## Compositional Data Model fit - CoDa (Oeppen 2008)
## Model with predictor: clr d[x] = a[x] + b[x]k[t]
## Call: coda(dx = CoDa.data, x = 0:110, y = 1960:2014)
##
## Years in fit:  1960 - 2014
## Ages in fit:  0 - 110
```

Output objects

```
ls(M)
```

```
## [1] "call"         "coefficients" "fitted"       "input"
## [5] "residuals"    "x"            "y"
```

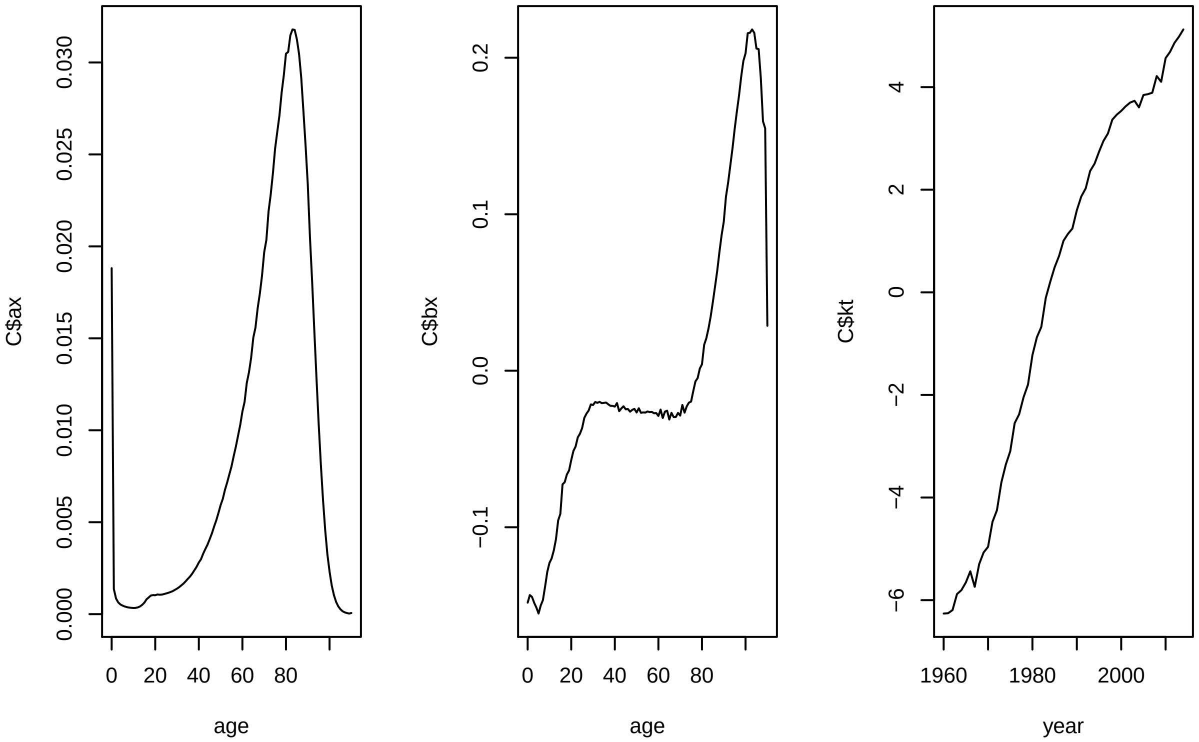
Summary

```
summary(M)
```

```
##
## Compositional Data Model fit - CoDa (Oeppen 2008)
## Model with predictor: clr d[x] = a[x] + b[x]k[t]
##
## Coefficients:
##          ax       bx    .    y       kt
## 0   0.01882 -0.14822    | 1960  -6.2615
## 1   0.00136 -0.14334    | 1961 -6.25469
## 2   0.00085 -0.14457    | 1962 -6.19413
## 3   0.00064 -0.14831    | 1963 -5.88283
## 4   0.00053 -0.15134    | 1964 -5.80534
## 5   0.00046 -0.15515    | 1965 -5.65574
## ...     ...      ... <NA>  ...      ...
## 105 0.00026  0.20582    | 2009  4.10337
## 106 0.00015  0.20556    | 2010  4.56493
## 107   9e-05  0.18714    | 2011  4.68676
## 108   6e-05  0.15935    | 2012  4.86257
## 109   3e-05   0.1549    | 2013  4.98146
## 110   6e-05   0.0287    | 2014  5.12442
```

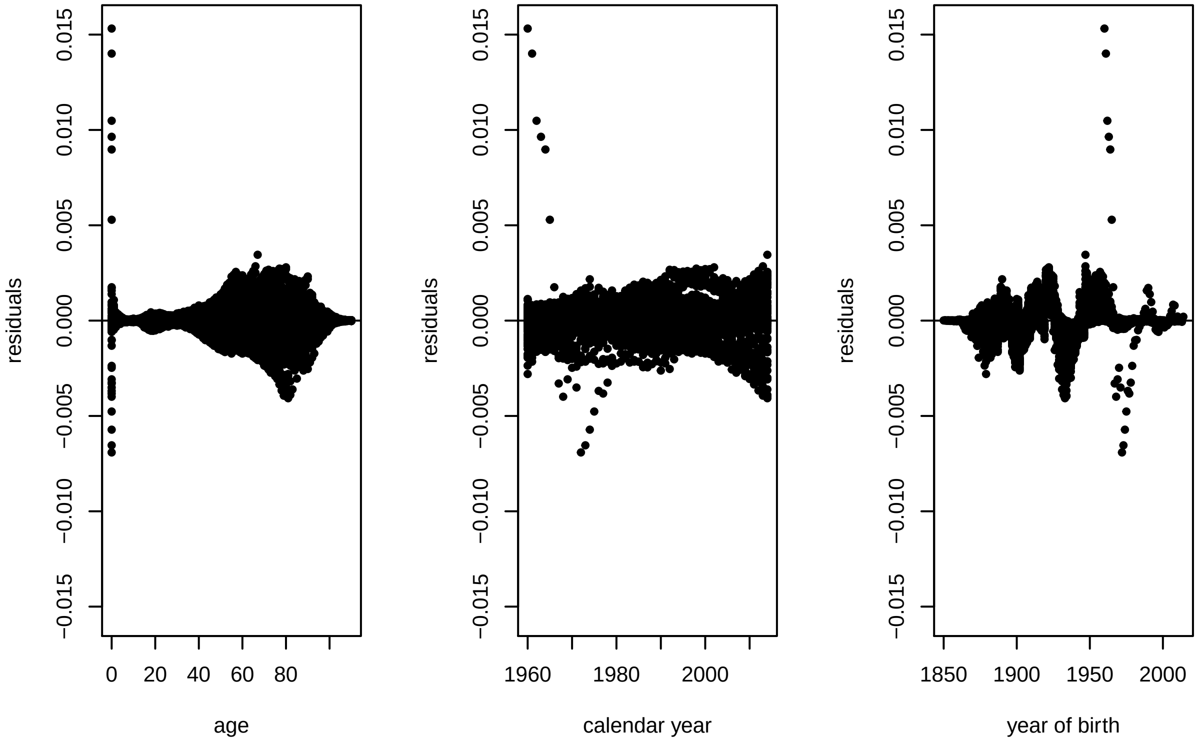
Estimated parameters

```
plot(M)
```

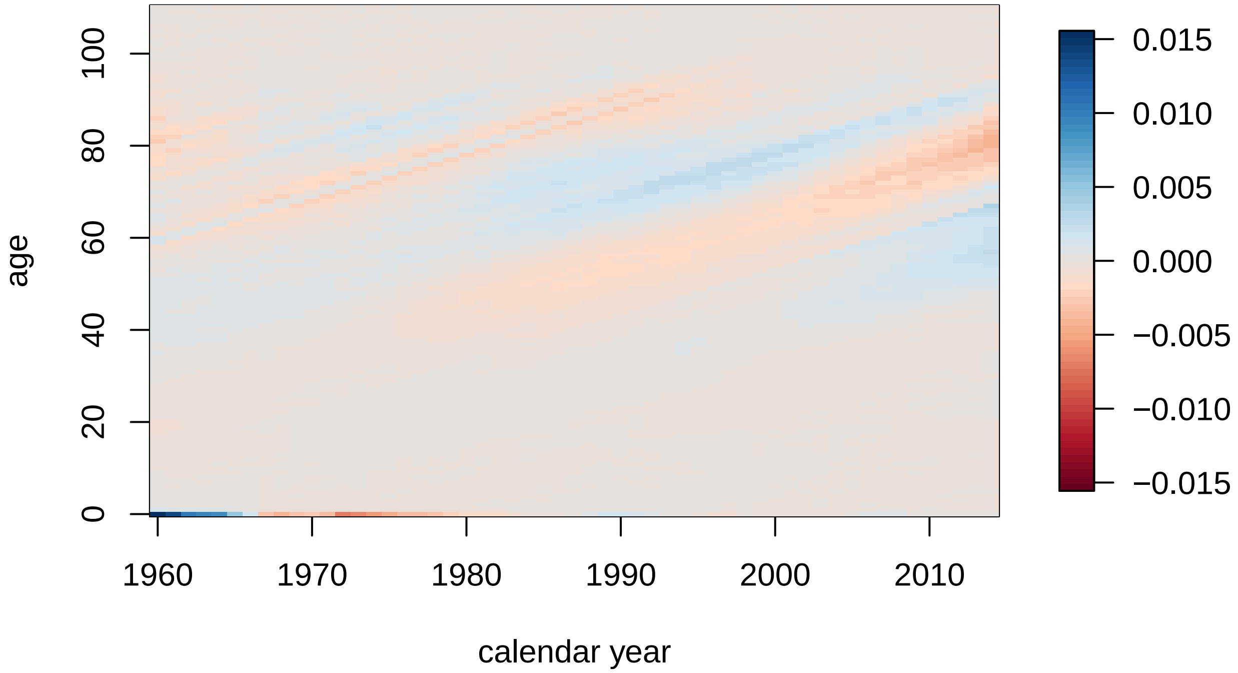


Plot Residuals

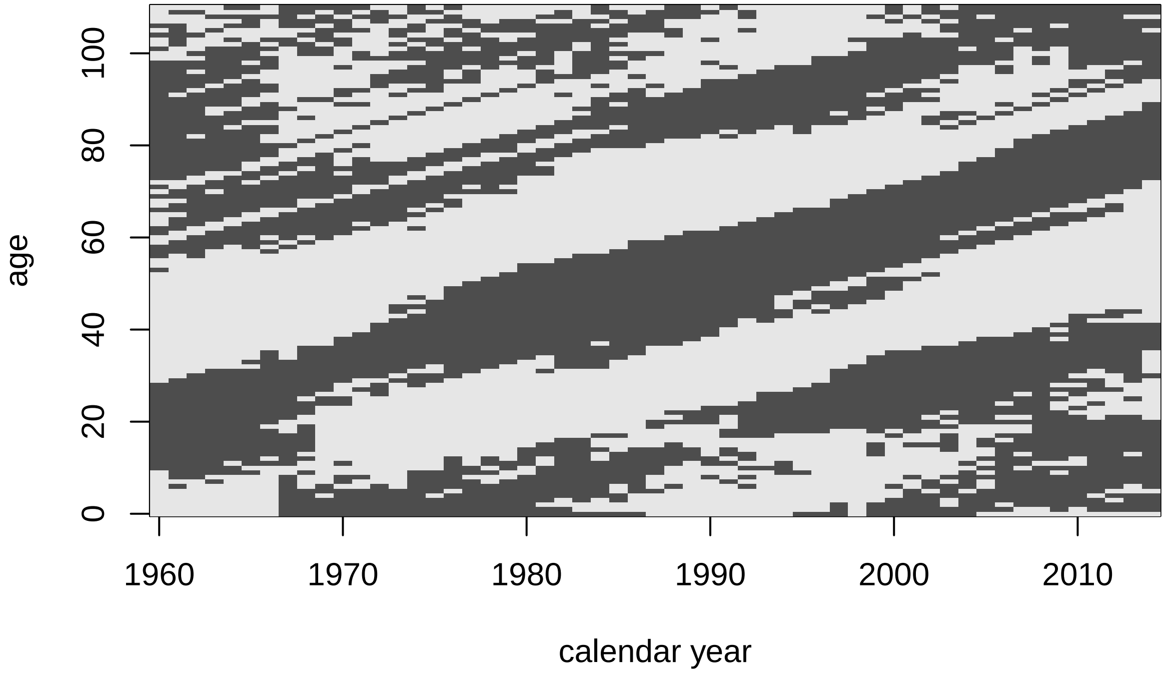
```r
res <- resid(M)
plot(res, type = "scatter")
```



```r
plot(res, type = "colourmap")
```



```r
plot(res, type = "signplot")
```



Mortality projections

Forecasting mortality 30 years in the future using CoDa model:

```r
P <- predict(M, h = 30, jumpchoice = 'actual')
P
```

```
##
## Compositional Data Model forecast
## Ages in forecast:  2015 - 2044
## Time series model (kt): ARIMA(2,2,2)
```

```
# Predicted distribution of death
head(P$predicted.values$mean)
```

```
##           2015         2016         2017         2018         2019
## 0 0.0075851408 0.0075122888 0.0071610721 0.0070100479 0.0067843320
## 1 0.0004882349 0.0004836694 0.0004616400 0.0004521574 0.0004379734
## 2 0.0003116650 0.0003087307 0.0002945754 0.0002884838 0.0002793739
## 3 0.0002283841 0.0002261895 0.0002156096 0.0002110603 0.0002042611
## 4 0.0001900496 0.0001881935 0.0001792500 0.0001754068 0.0001696659
## 5 0.0001766520 0.0001748917 0.0001664159 0.0001627766 0.0001573438
##           2020         2021         2022         2023         2024
## 0 0.0065498611 0.0063901695 0.0061528080 0.0059875747 0.0057904885
## 1 0.0004232242 0.0004131699 0.0003982116 0.0003877887 0.0003753454
## 2 0.0002699034 0.0002634490 0.0002538487 0.0002471609 0.0002391785
## 3 0.0001971984 0.0001923883 0.0001852387 0.0001802618 0.0001743256
## 4 0.0001637062 0.0001596496 0.0001536234 0.0001494310 0.0001444333
## 5 0.0001517084 0.0001478751 0.0001421848 0.0001382290 0.0001335166
##           2025         2026         2027         2028         2029
## 0 0.0056057935 0.0054392728 0.0052553604 0.0050972852 0.0049295726
## 1 0.0003636732 0.0003531400 0.0003414958 0.0003314780 0.0003208397
## 2 0.0002316926 0.0002249387 0.0002174742 0.0002110538 0.0002042374
## 3 0.0001687627 0.0001637473 0.0001582081 0.0001534472 0.0001483962
## 4 0.0001397526 0.0001355350 0.0001308796 0.0001268807 0.0001226404
## 5 0.0001291065 0.0001251354 0.0001207554 0.0001169957 0.0001130120
##           2030         2031         2032         2033         2034
## 0 0.0047720332 0.0046201851 0.0044673466 4.324865e-03 4.181576e-03
## 1 0.0003108373 0.0003011874 0.0002914654 2.823939e-04 2.732625e-04
## 2 0.0001978299 0.0001916496 0.0001854247 1.796176e-04 1.737735e-04
## 3 0.0001436516 0.0001390785 0.0001344756 1.301848e-04 1.258697e-04
## 4 0.0001186598 0.0001148251 0.0001109678 1.073740e-04 1.037619e-04
## 5 0.0001092748 0.0001056774 0.0001020612 9.869453e-05 9.531318e-05
##           2035         2036         2037         2038         2039
## 0 0.0040449341 3.911728e-03 3.781308e-03 3.656261e-03 3.533304e-03
## 1 0.0002645465 2.560419e-04 2.477074e-04 2.397089e-04 2.318368e-04
## 2 0.0001681966 1.627563e-04 1.574260e-04 1.523117e-04 1.472795e-04
## 3 0.0001217548 1.177434e-04 1.138161e-04 1.100506e-04 1.063481e-04
## 4 0.0001003194 9.696550e-05 9.368366e-05 9.053888e-05 8.744848e-05
## 5 0.0000920929 8.895769e-05 8.589207e-05 8.295659e-05 8.007394e-05
##           2040         2041         2042         2043         2044
## 0 3.414920e-03 3.299608e-03 3.187566e-03 3.079219e-03 2.973643e-03
## 1 2.242505e-04 2.168542e-04 2.096610e-04 2.026986e-04 1.959080e-04
## 2 1.424311e-04 1.377052e-04 1.331102e-04 1.286637e-04 1.243278e-04
## 3 1.027833e-04 9.931113e-05 9.593747e-05 9.267512e-05 8.949626e-05
## 4 8.447477e-05 8.157992e-05 7.876883e-05 7.605202e-05 7.340626e-05
## 5 7.730212e-05 7.460574e-05 7.198923e-05 6.946231e-05 6.700321e-05
```