

# Demographic Forecasting

*Lecture 4: the Lee-Carter method*

Ugo Filippo Basellini

basellini@demogr.mpg.de

Max Planck Institute for Demographic Research

May 7, 2025



# Course overview

- Lecture 1: direct extrapolation by (generalized) linear models
- Lecture 2: direct extrapolation by time-series methods
- Lecture 3: parametric approaches
- Lecture 4: the Lee-Carter method



# Mortality forecasting

- Crucial for sustainability of pensions, insurances, elderly care; predicting population ageing and projecting populations; ...
- Until the 1980s, the methods used to forecast mortality were **deterministic**, based on mathematical formulae or expert judgment
- Revived interest in recent years following the introduction of the Lee-Carter method in 1992
- One of the firstly introduced **stochastic** mortality models  $\Rightarrow$  the model “revolutionized probabilistic mortality and population forecasting” (Raftery 2023)



# The Lee-Carter method (1992)

- Proposed in 1992 to model and forecast US mortality
- After 30+ years, Lee-Carter (LC) still widely employed by variety of users: governments, private companies, international organizations, ...
- The landmark model in mortality forecasting
- An **extrapolation** method:
  - model the mortality surface over age and time
  - extrapolate trends in the future, assuming that observed trends will continue
- Simplicity, robustness and objectivity have made the model so successful
- Nonetheless, some limitations of the model have stimulated several extensions over the years (see Basellini et al. (2023) for a recent comprehensive review)



# The LC method

- A simple log-bilinear functional form for mortality rates  $m_{x,t}$  at age  $x$  and time  $t$

$$\ln(m_{x,t}) = \alpha_x + \beta_x \kappa_t + \epsilon_{x,t}$$

where:

- $\alpha_x$  is the general shape of log-mortality at age  $x$
- $\beta_x$  is the rate of mortality improvement at age  $x$
- $\kappa_t$  is the general level of mortality at time  $t$
- $\epsilon_{x,t}$  is the error term with mean 0 and variance  $\sigma_\epsilon^2$ , reflecting residual age-specific influences not captured by the model
- Modelling log-rates  $\Rightarrow$  fitted and forecast rates constrained to be positive
- Log transformation partially counters heteroscedasticity of observed rates



# The LC method: a schematic view

$$\begin{aligned}
 \ln(m_{x,t}) &\simeq \alpha_x + \beta_x \kappa_t \\
 \begin{pmatrix} \ln(m_{x_1,t_1}) & \ln(m_{x_1,t_2}) & \dots & \ln(m_{x_1,t_n}) \\ \ln(m_{x_2,t_1}) & \ln(m_{x_2,t_2}) & \dots & \ln(m_{x_2,t_n}) \\ \ln(m_{x_3,t_1}) & \ln(m_{x_3,t_2}) & \dots & \ln(m_{x_3,t_n}) \\ \vdots & \vdots & \ddots & \vdots \\ \ln(m_{x_m,t_1}) & \ln(m_{x_m,t_2}) & \dots & \ln(m_{x_m,t_n}) \end{pmatrix} &\simeq \\
 \begin{pmatrix} \alpha_{x_1} \\ \alpha_{x_2} \\ \alpha_{x_3} \\ \vdots \\ \alpha_{x_m} \end{pmatrix} + \begin{pmatrix} \beta_{x_1} \\ \beta_{x_2} \\ \beta_{x_3} \\ \vdots \\ \beta_{x_m} \end{pmatrix} & \begin{pmatrix} \kappa_{t_1} & \kappa_{t_2} & \dots & \kappa_{t_n} \end{pmatrix} \\
 \underbrace{n}_{\text{years}} \times \underbrace{m}_{\text{ages}} = \underbrace{mn}_{\text{cells}} &\simeq \underbrace{m}_{\alpha_i} + \underbrace{m}_{\beta_i} + \underbrace{n}_{\kappa_j} - \underbrace{2}_{\text{constraints}} = \underbrace{2m + n - 2}_{\text{parameters}}
 \end{aligned}$$





# The LC method

- The model is undetermined: if  $\theta_1 = [\alpha, \beta, \kappa]$  is a solution, then for any scalar  $c$ :
  - $\theta_2 = [\alpha - \beta c, \beta, \kappa + c]$  is also a solution
  - $\theta_3 = [\alpha, \beta c, \kappa/c]$  is also a solution
- Two constraints introduced to ensure model identification:

$$\sum_x \beta_x = 1 \quad \text{and} \quad \sum_t \kappa_t = 0$$





# Model estimation

- The model is estimated by minimizing the residual sum of squares:

$$\sum_{x,t} \left( \ln(m_{x,t}) - \alpha_x - \beta_x \kappa_t \right)^2 \quad (1)$$

- A singular value decomposition (SVD) is employed to minimize Equation 1 & derive an ordinary least squares (OLS) solution:
  - $\hat{\alpha}_x$  is the average of the observed  $\ln(m_{x,t})$ :  $\hat{\alpha}_x = \frac{1}{n} \sum_{t=1}^n \ln(m_{x,t})$



# Exercise

## Exercise

Open your R session. Load the `MortSWE.Rdata` dataset, and consider only data from 1950 onward for males aged  $0 \leq x \leq 100$ . Derive the LC estimate of the

$$\alpha_x = \frac{1}{n} \sum_{t=1}^n \ln(m_{x,t}) \text{ parameter}$$

*Hint:* Replace the only cell with 0 deaths with a tiny number of deaths (e.g., 1), and compute again the matrix of log-mortality rates. This will be particularly helpful later on.



# One possible solution

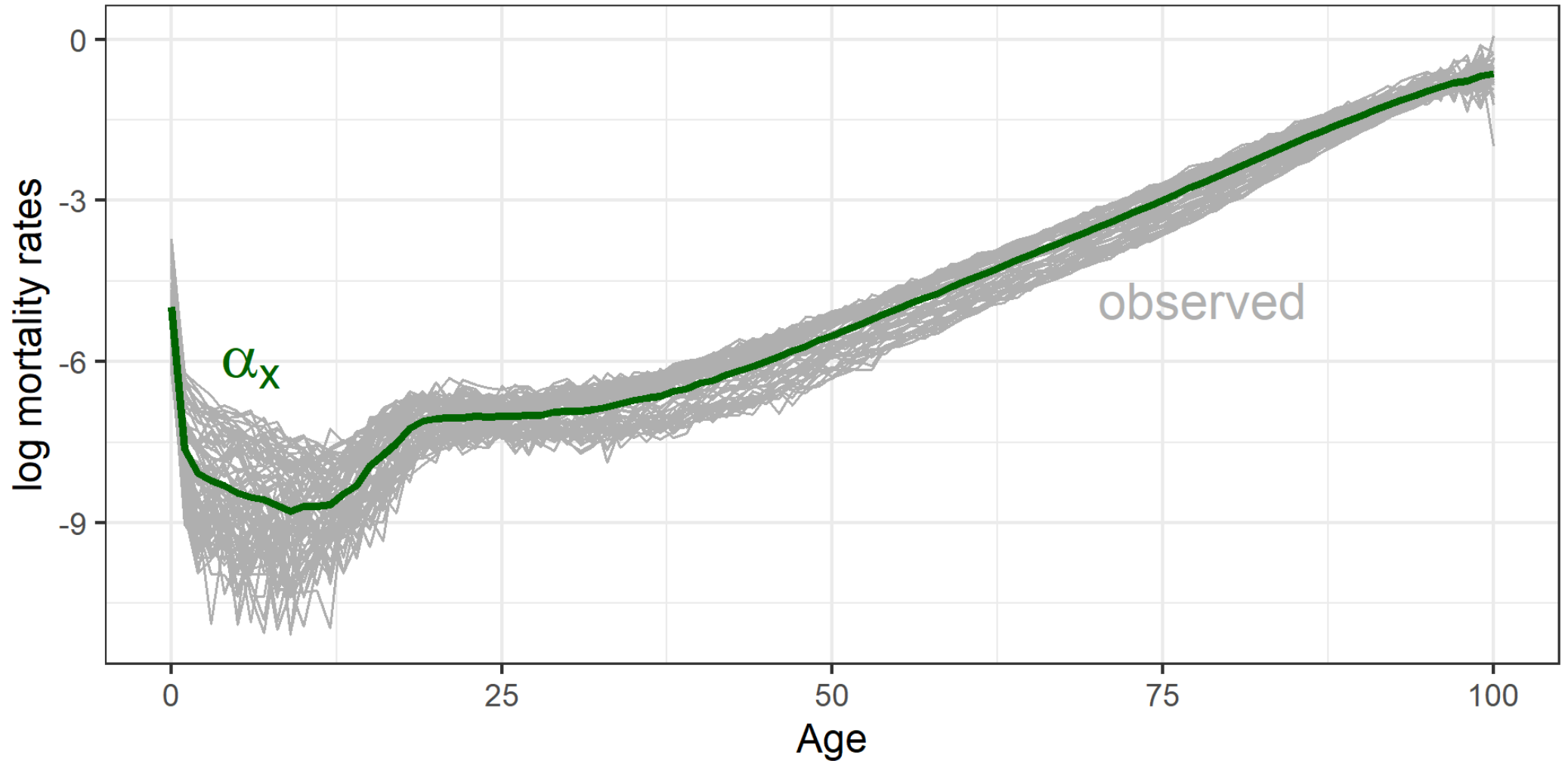
```

1  ## cleaning the workspace
2  rm(list=ls(all=TRUE))
3  ## load useful packages
4  library(tidyverse)
5  ## loading data
6  setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
7  load("data/MORTSWE.Rdata")
8  ## subset
9  my.df <- MORT.SWE %>% filter(Year>=1950, Sex=="Male", Age <= 100)
10 ## extract data
11 x <- unique(my.df$Age)
12 t <- unique(my.df$Year)
13 n <- length(t)
14 m <- length(x)
15 ## matrices
16 DEATHS <- matrix(my.df$Deaths,m,n)
17 EXPOS <- matrix(my.df$Exposures,m,n)
18 RATES <- matrix(my.df$Rates,m,n)
19 LPRATES <- matrix(my.df$LprRates,m,n)

```



# One possible solution



# Model estimation

- The model is estimated by minimizing the residual sum of squares:

$$\sum_{x,t} \left( \ln(m_{x,t}) - \alpha_x - \beta_x \kappa_t \right)^2 \quad (2)$$

- A singular value decomposition (SVD) is employed to minimize Equation 2 & derive an ordinary least squares (OLS) solution:
  - $\hat{\alpha}_x$  is the average of the observed  $\ln(m_{x,t})$ :  $\hat{\alpha}_x = \frac{1}{n} \sum_{t=1}^n \ln(m_{x,t})$
  - $\hat{\beta}_x$  and  $\hat{\kappa}_t$  are the first left- and right-singular vectors of the SVD of the matrix  $\ln(m_{x,t}) - \hat{\alpha}_x$ , adjusted to comply with the two constraints  
 $\sum_x \beta_x = 1$  and  $\sum_t \kappa_t = 0$



# Singular Value Decomposition

- An SVD of any  $m \times n$  rectangular matrix  $\mathbf{M}$  is a factorization of the form:

$$\mathbf{M} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}'$$

where

- $\mathbf{U}$  is a  $m \times m$  orthogonal matrix (left-singular vectors)
  - $\mathbf{V}$  is a  $n \times m$  orthogonal matrix (right-singular vectors)
  - $\mathbf{\Sigma}$  is a  $m \times m$  diagonal matrix with  $s_{ii} = 0$  if  $i \neq j$  ( $s_{ii}$ , singular values)
- In other words, we can decompose  $\mathbf{M}$  as

$$\mathbf{M} = U_{,1} s_{1,1} V'_{,1} + U_{,2} s_{2,2} V'_{,2} + U_{,3} s_{3,3} V'_{,3} + \dots + U_{,m} s_{m,m} V'_{,m}$$

- The first few singular values/vectors are more important in describing the information in  $\mathbf{M}$



# Exercise

## Exercise

Now perform a singular value decomposition of the matrix of “centered” mortality rates  $\ln(m_{x,t}) - \hat{\alpha}_x$ , and plot the first left- and right-singular vectors

*Hint:* Use the command `svd(x=., nu=1, nv=1)` to perform the SVD of your centred mortality rates matrix, and then extract the vectors `u` and `v` from the SVD object. Remember to use the modified log-rates matrix (which excludes infinite values).



# One possible solution

```

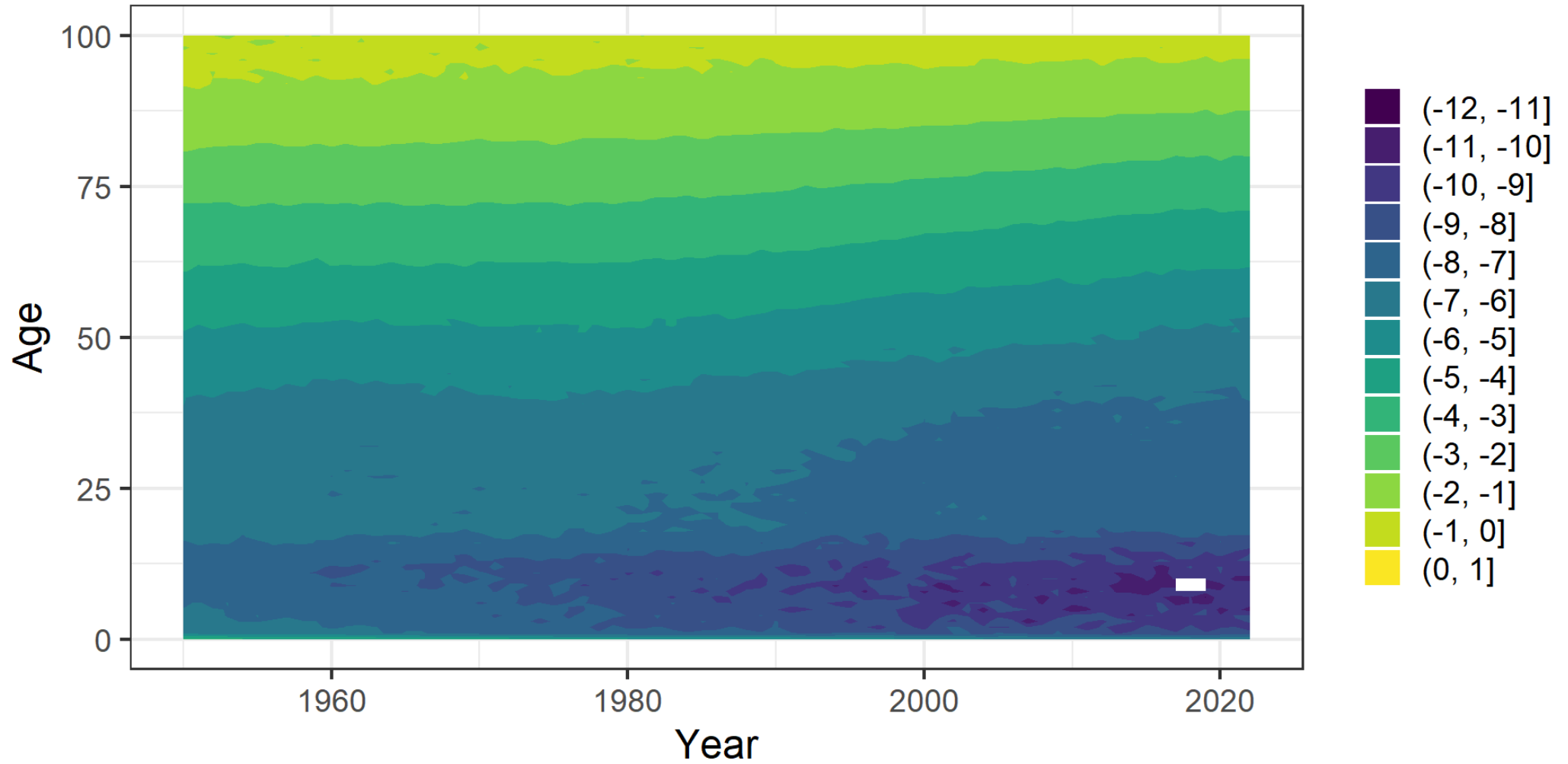
1  ##---- step 2: derive BETA and KAPPA ----
2  ## derive matrix of residuals ("centred" mortality rates)
3  LMXres <- LRATES1 - Alpha
4  ## plotting the matrix
5  library(fields)
6  image.plot(t,x,t(LMXres),axes="F",xlab="",ylab="")
7  axis(1);axis(2,las=2);box()
8  mtext("ages",side=2,line=2);mtext("year",side=1,line=2)
9  dev.off()
10 ## performing the SVD
11 LCsvd <- svd(LMXres,nu=1,nv=1)
12 ## extract first left- and right-singular vectors of svd
13 Beta <- c(LCsvd$u)
14 Kappal <- c(LCsvd$v)
15 ## plotting singular vectors
16 par(mfrow=c(1,2))
17 plot(x,Beta)
18 plot(t,Kappal)
19 ## including the constraints

```

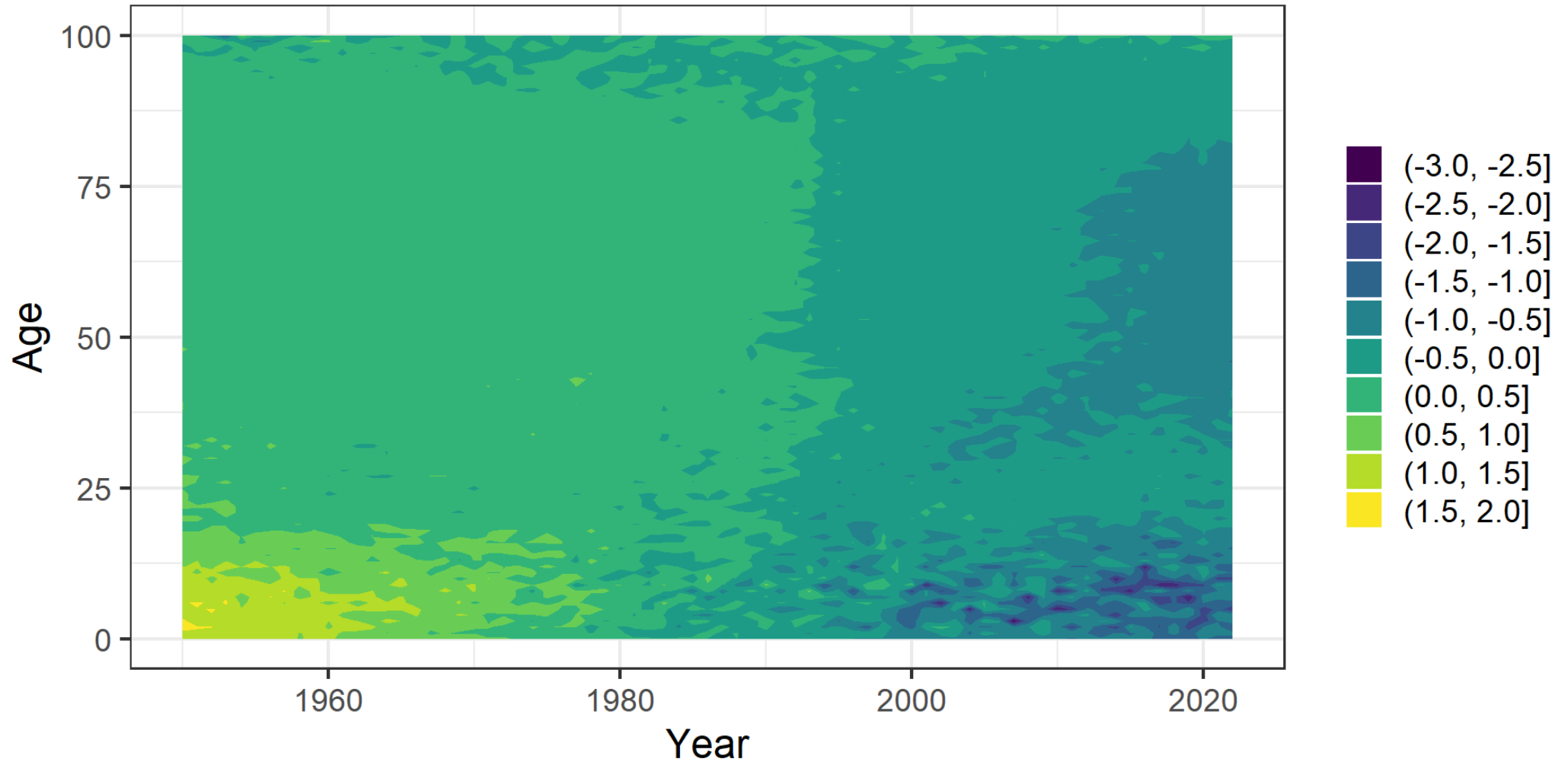




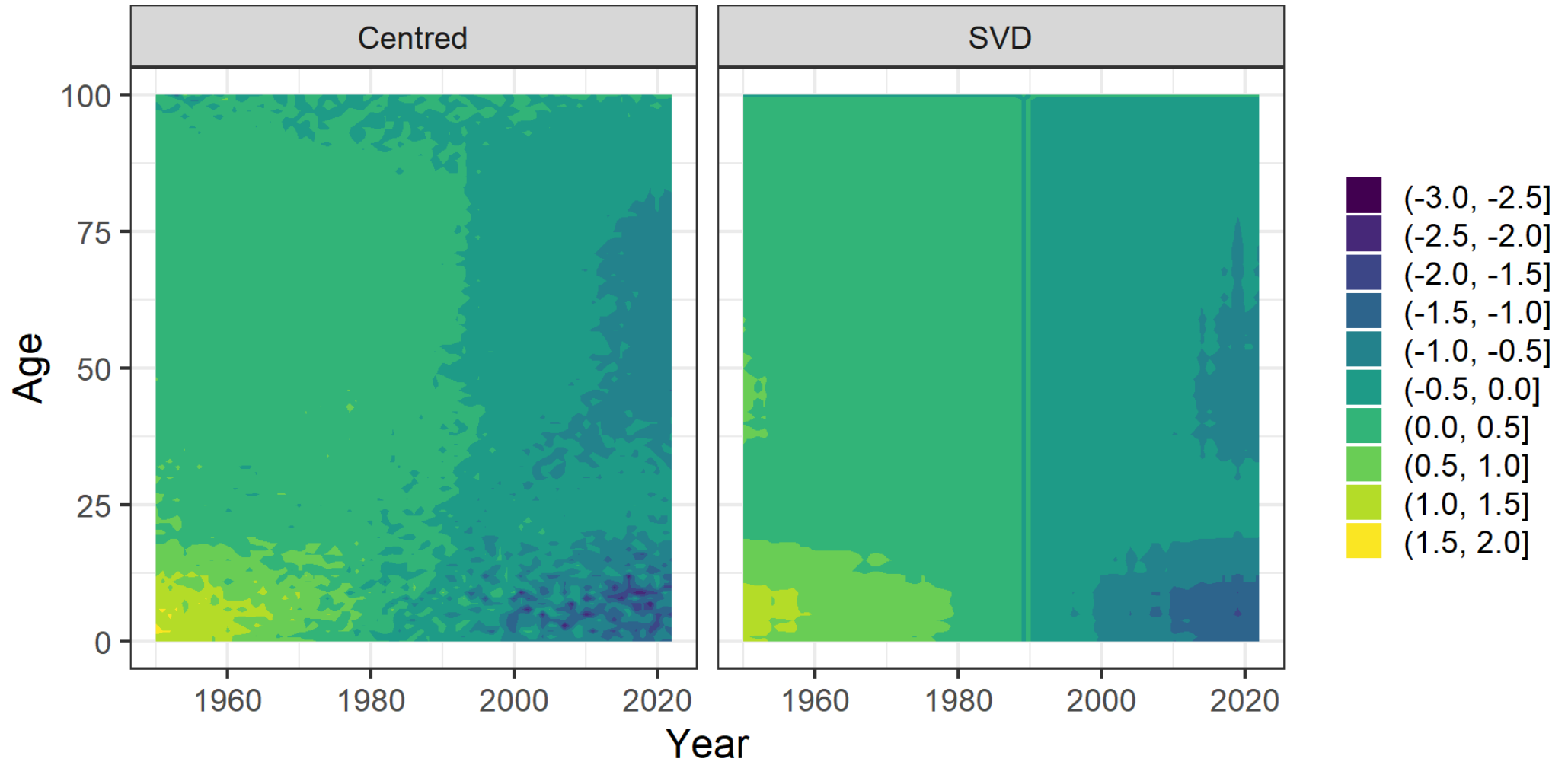
# Log-mortality rates



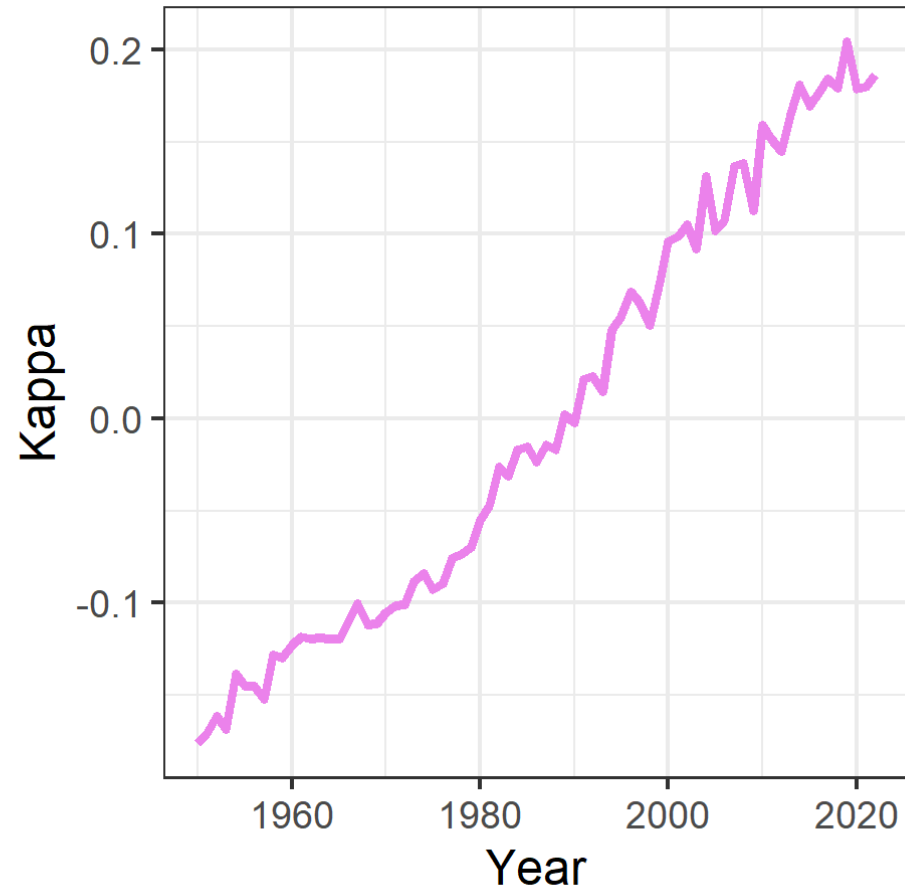
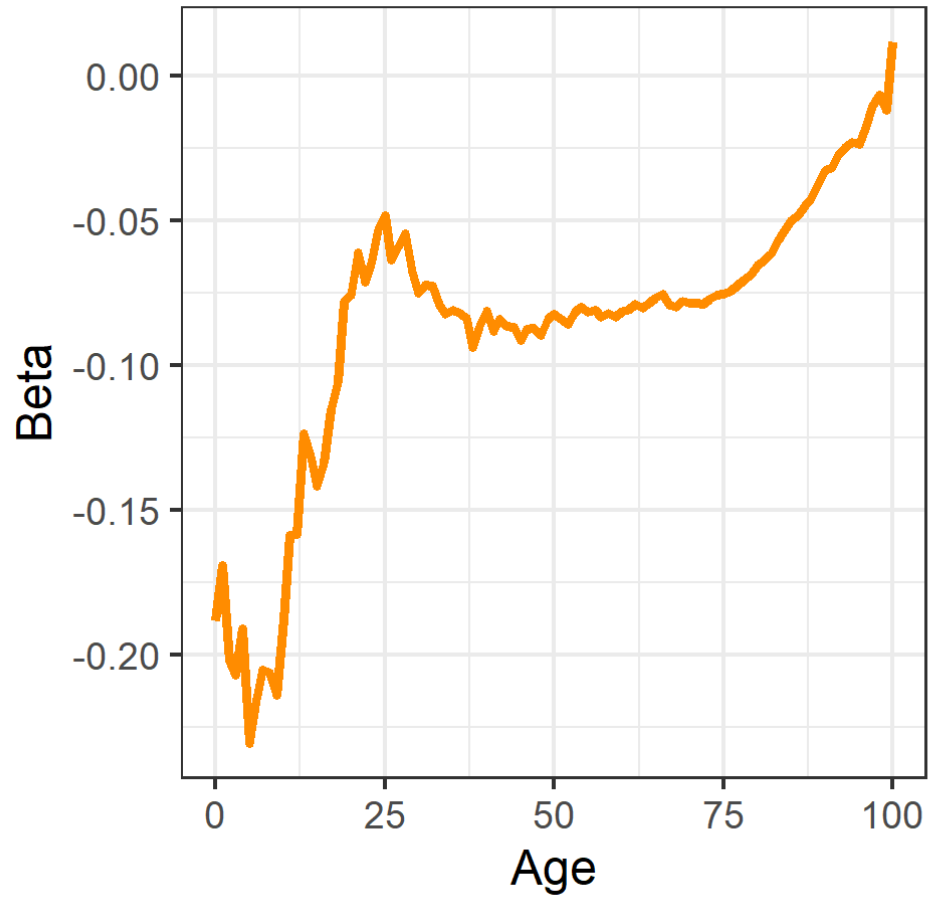
# Centred log-mortality rates



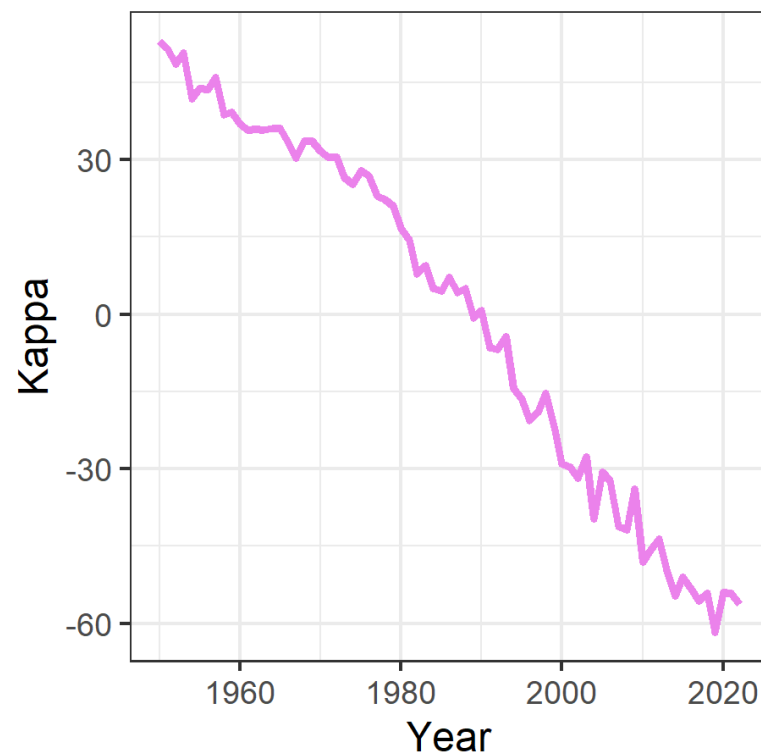
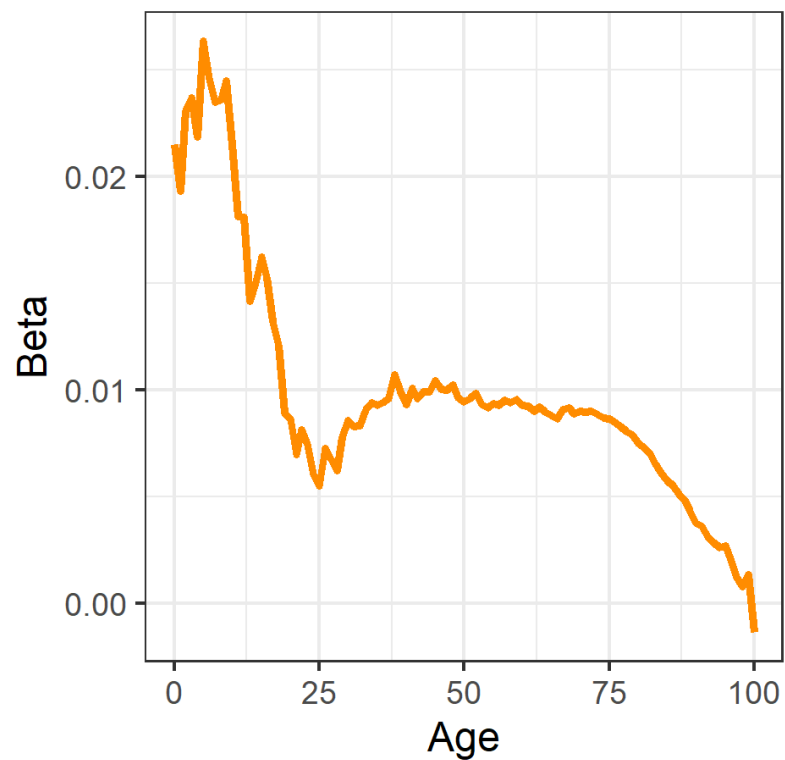
# SVD approximation



# $\hat{\beta}_x$ and $\hat{\kappa}_t$ from SVD



# $\hat{\beta}_x$ and $\hat{\kappa}_t$ with constraints



# Second step estimation of $\hat{\kappa}_t$

- In a second-step estimation,  $\hat{\kappa}_t$  is adjusted so that the fitted deaths match the observed deaths in all years, i.e.

$$\sum_x \hat{y}_{x,t} = \sum_x y_{x,t} \quad \text{for all } t$$



# One possible solution

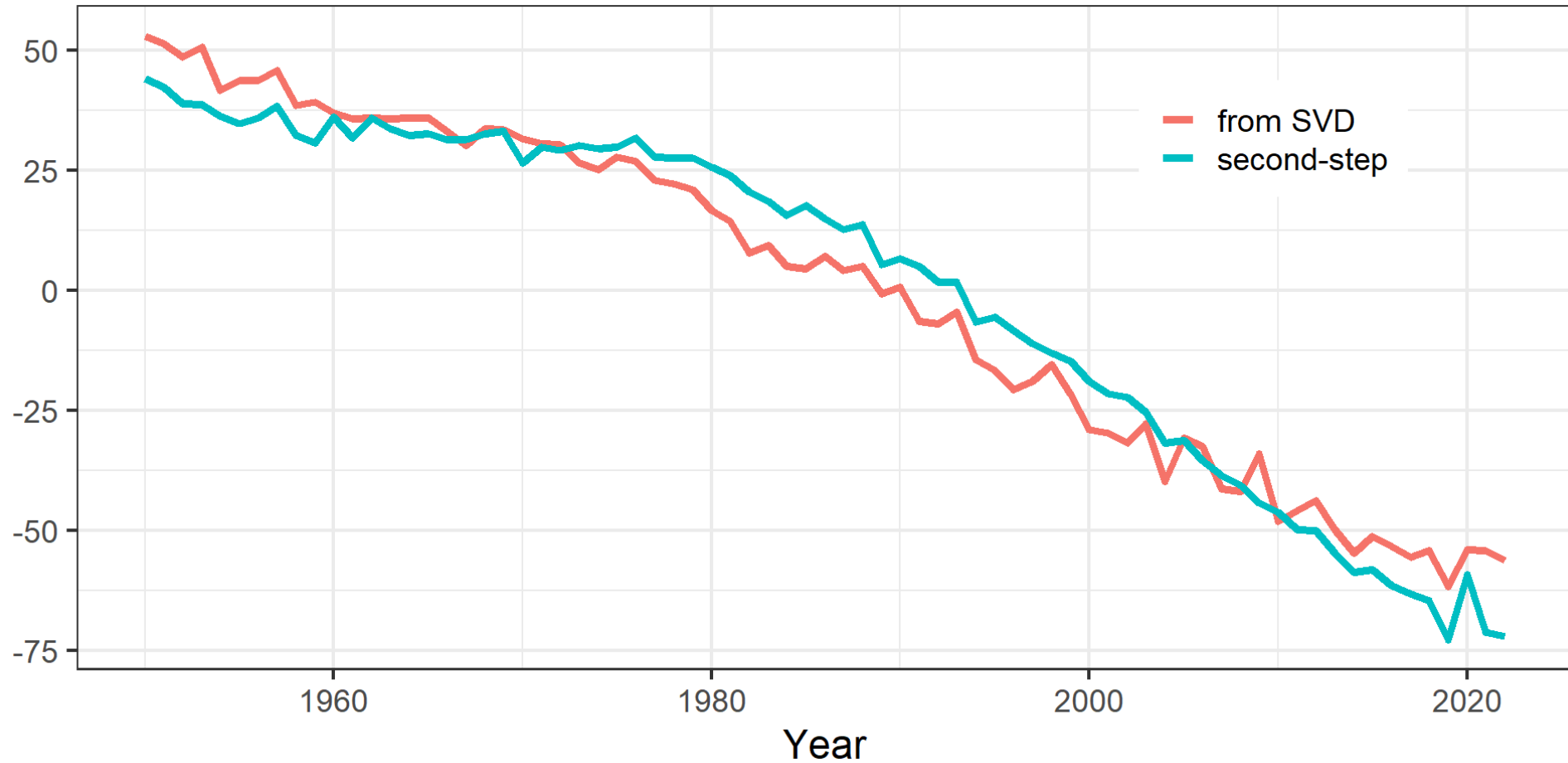
```

1  ##---- step 3: adjust KAPPA -----
2  ## function to compute difference between observed and fitted LC de
3  koptim <- function(par,alpha,beta,sum.dx,Exp) {
4    kappa <- par[1]
5    lmx.lc <- alpha+beta*kappa
6    z.lc <- exp(lmx.lc)*Exp
7    sum.z.lc <- sum(z.lc)
8    diff.lc <- abs(sum.dx-sum.z.lc)
9    return(diff.lc)
10 }
11 ## adjust Kappa every year
12 Kappa <- numeric(n)
13 for (i in 1:n){
14   KappaSecStep <- optimize(f=koptim,interval=c(-100,100),alpha=Alph
15   beta=Beta,sum.dx=sum(DEATHS[,i]),Exp=EXPOS[,i])
16   Kappa[i] <- KappaSecStep$minimum
17 }
18 ## plotting
19 plot(t,Kappa1,col="red",lty="n",log="y",xlab="Year",ylab="Kappa")

```

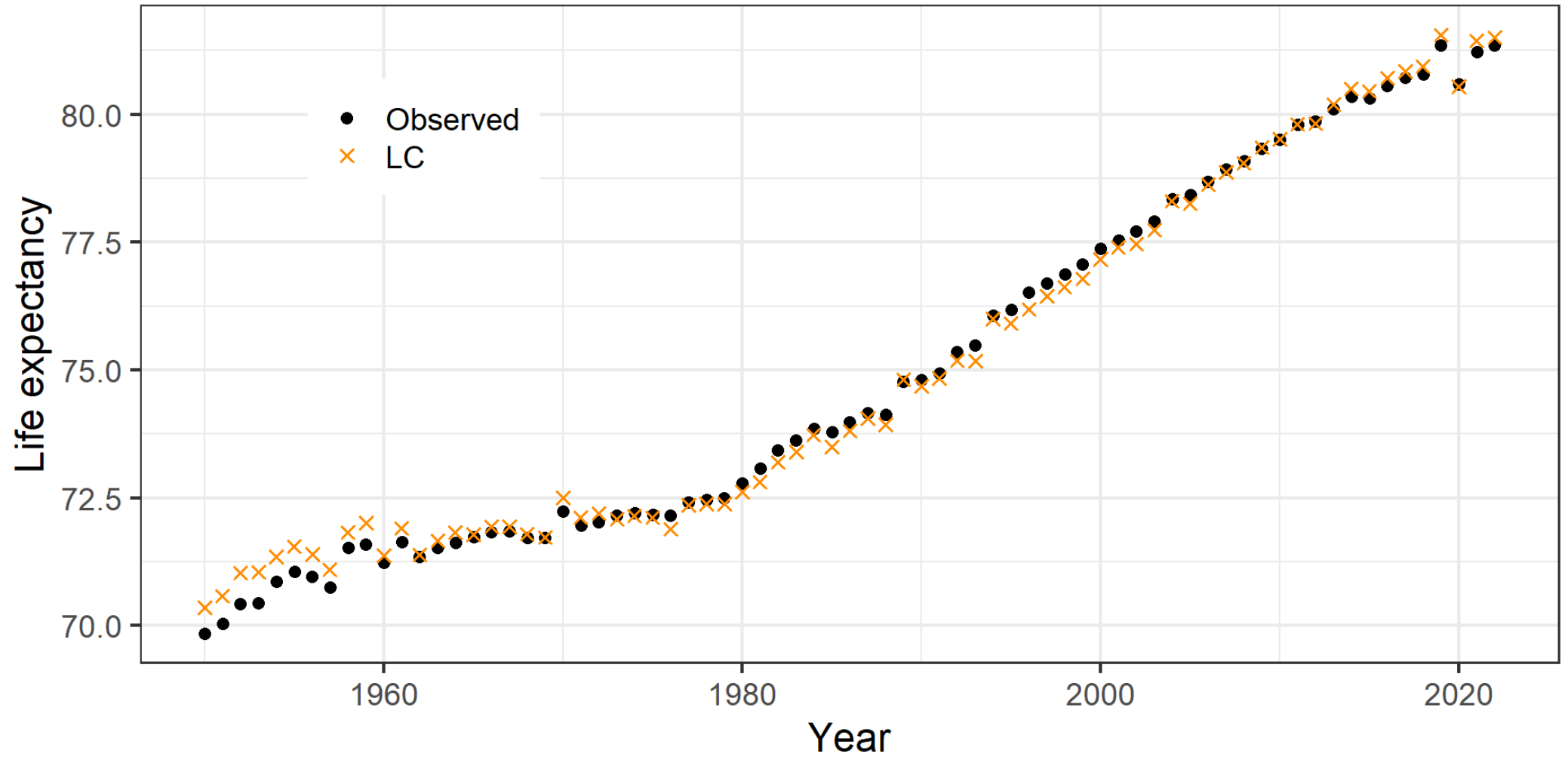


# Second step estimation of $\hat{\kappa}_t$





# Goodness-of-fit



# Intermezzo : animated visualizations & shiny-app

In some instances, it can be useful to introduce dynamic plots, animations and flexible outputs in your presentations or documents:

- can grab the audience's attention
- show time-series evolutions
- to describe your model
- sensitivity analysis of your results



# Animated images

- `\animategraphics` in Beamer (see LaTeX workshop)
- `gganimate` package in Quarto



# Animation plots with **gganimate**

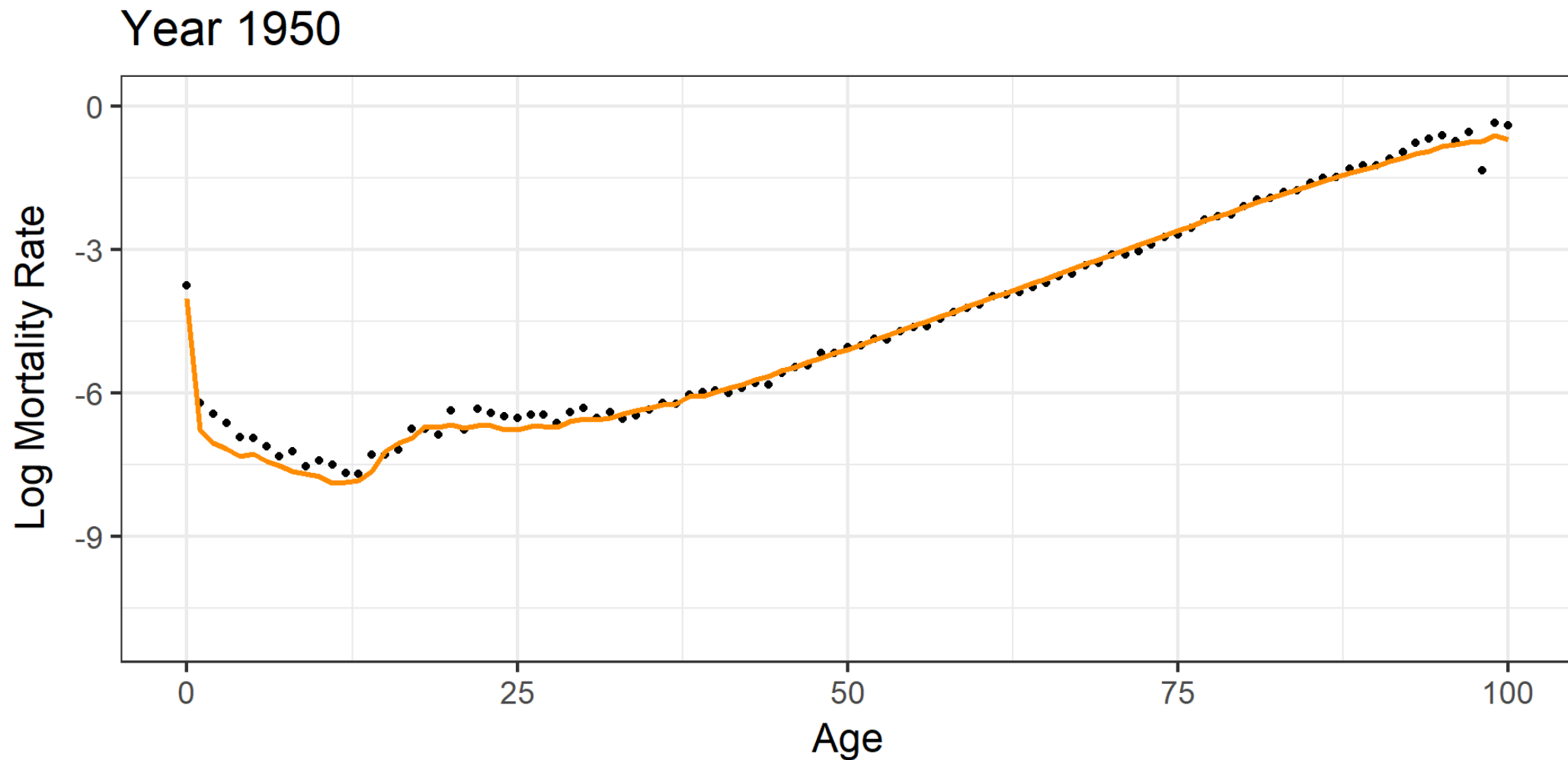
```

1  ## fitted log-mortality
2  Ones <- matrix(1,n)
3  ETAlc <- Alpha%*%t(Ones) + Beta%*%t(Kappa)
4  ## basic plot
5  g <- my.df %>%
6    mutate(logRates=case_when(
7      is.infinite(logRates)~NA,
8      TRUE~logRates),
9      Fitted=c(ETAlc)) %>%
10    ggplot(aes(x=Age,group=Year))+
11    geom_point(aes(y=logRates))+
12    geom_line(aes(y=Fitted),color="darkorange",linewidth=1.2)+
13    scale_color_viridis_c() +
14    theme_bw(base_size = 18) +
15    labs(y= "Log Mortality Rate")
16  ## animating with gganimate
17  library(gganimate)
18  gg <- g + transition_time(Year) +
19    set_title("Year: {frame_time}")

```



# Goodness-of-fit



# Shiny apps

- [shiny](#) is an [R](#) package that makes it easy to build interactive web apps straight from [R](#)
- keep them on webpages or embed them in R Markdown documents or Quarto presentations
- a user-friendly interface to interact with your [R](#) analysis and show your results ([a nice example](#) from [DOI: 10.1215/00703370-9637380](#))
- it is composed by
  - an [UI](#) (user interface), where you can create the inputs for your and decide the outputs to display
  - a [server](#), where you assemble the outputs from your given inputs
  - the [shinyApp](#), putting the two together
- to learn more, visit <https://shiny.posit.co/> to get started, plenty of videos and written tutorials

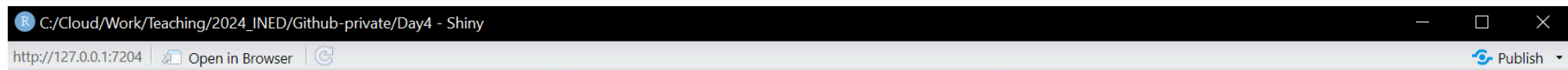


# Shiny app: an example

```
1  ##----- SHINY APP EXAMPLE -----
2
3  ## cleaning the workspace
4  rm(list=ls(all=TRUE))
5  ## load useful packages
6  library(tidyverse)
7  library(patchwork)
8  library(shiny)
9  ## loading data
10 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
11 load("data/MORTSWE.Rdata")
12 ## subset
13 my.df <- MORT.SWE %>% filter(Sex=="Male", Age <= 100)
14
15 ## build your user interface
16 ui <- fluidPage(
17   ## title of your shiny
18   titlePanel("LC parameters"),
19   ## display a slider that returns input$year to pass to the server
```

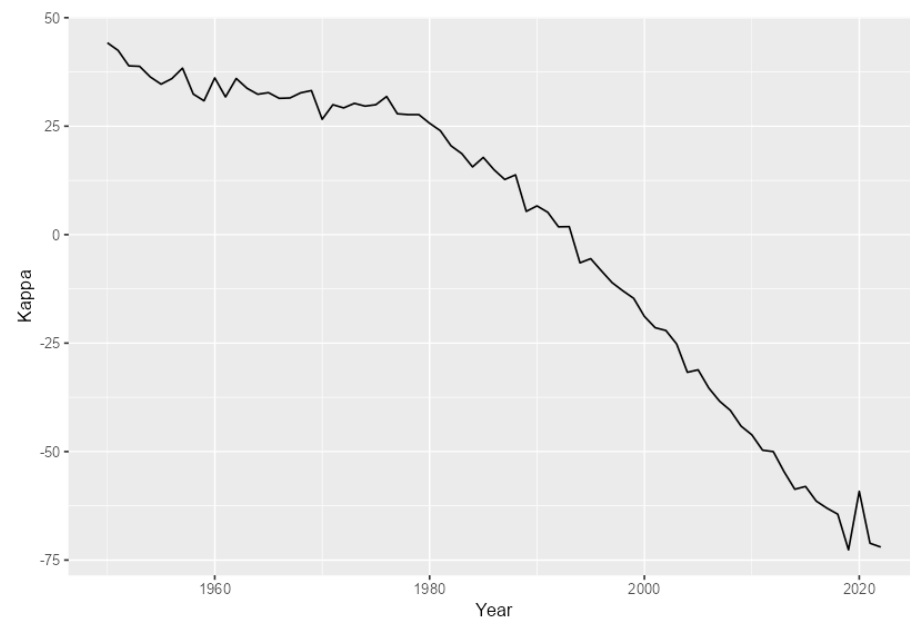
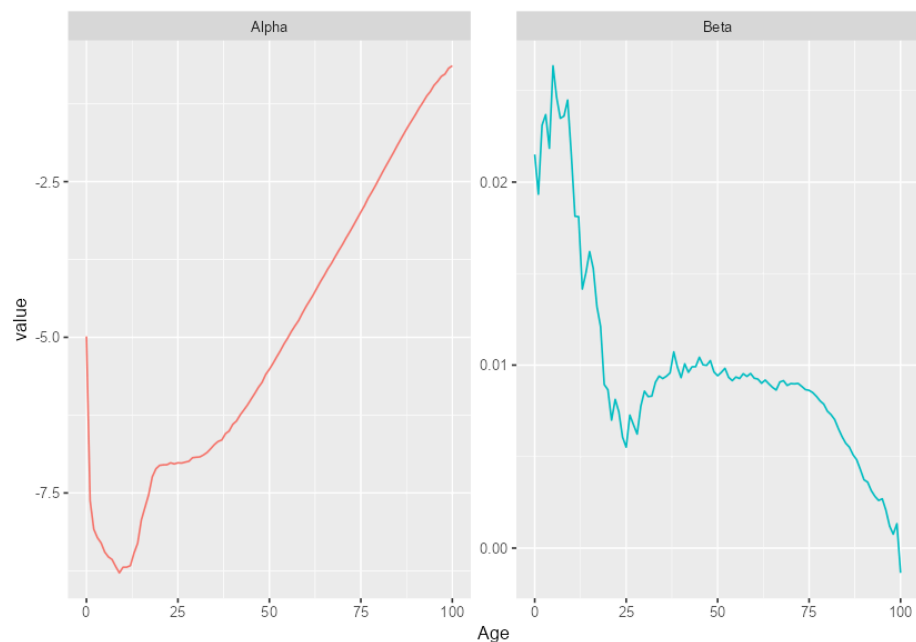
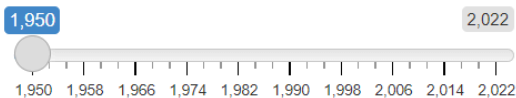


# Shiny app: an example



## LC parameters

Year





# Forecasting with LC

- Forecasting “made simple”: choose an appropriate time-series model for  $\hat{\kappa}_t$  and extrapolate it
- The forecast  $\hat{\kappa}_{T+h}$  allows one to derive the entire age-pattern of mortality at time  $T + h$ :

$$\ln(\hat{m}_{x,T+h}) = \hat{\alpha}_x + \hat{\beta}_x \hat{\kappa}_{T+h}$$

- LC suggest a random walk model (i.e. ARIMA(0,1,0)) with drift:  
 $\kappa_t = \kappa_{t-1} + c + e_t$  where  $c$  is a constant (drift) and  $e_t$  the error term (purely random process)
- For this time-series model:  $\hat{\kappa}_{T+h|T} = \hat{\kappa}_T + c h$
- Simulated future trajectories of  $\hat{\kappa}_{T+h}$  to construct prediction intervals for  $\hat{m}_{x,T+h}$  and other summary measures (e.g.,  $\hat{e}_{0,T+h}$ )
- (Coale and Guo (1989) adjustment for forecast rates at ages 85+)



# Forecasting with LC: a schematic view

$$\ln(m_{x,t}) \simeq \alpha_x + \beta_x \kappa_t$$

$$\ln(m_{x,T+h}) \simeq \alpha_x + \beta_x \kappa_{t_F}$$

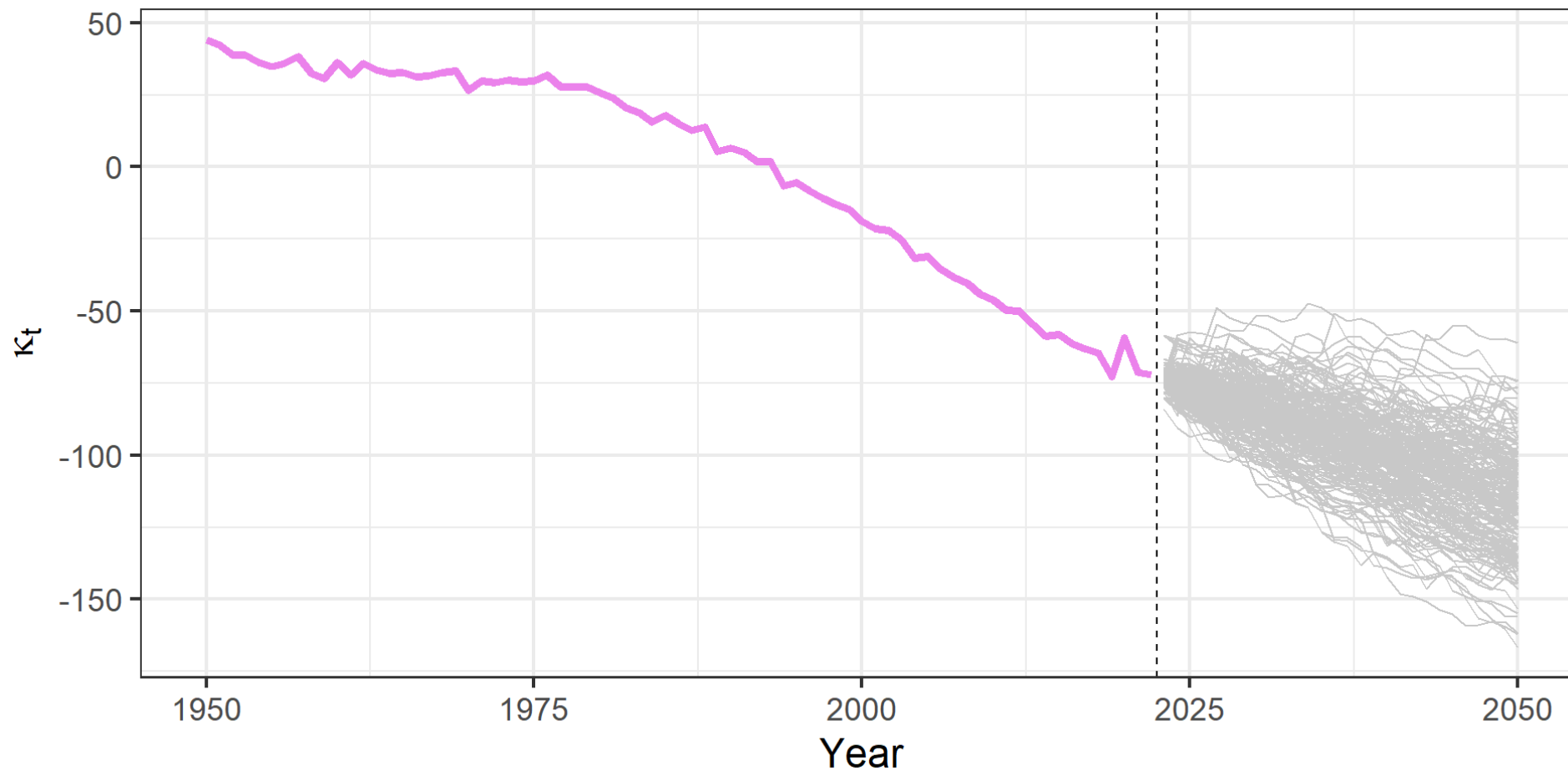
$$\begin{pmatrix} \ln(m_{x_1,t_1}) & \ln(m_{x_1,t_2}) & \dots & \ln(m_{x_1,T}) & \ln(m_{x_1,T+1}) & \dots & \ln(m_{x_1,T+h}) \\ \ln(m_{x_2,t_1}) & \ln(m_{x_2,t_2}) & \dots & \ln(m_{x_2,T}) & \ln(m_{x_2,T+1}) & \dots & \ln(m_{x_2,T+h}) \\ \ln(m_{x_3,t_1}) & \ln(m_{x_3,t_2}) & \dots & \ln(m_{x_3,T}) & \ln(m_{x_3,T+1}) & \dots & \ln(m_{x_3,T+h}) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \ln(m_{x_m,t_1}) & \ln(m_{x_m,t_2}) & \dots & \ln(m_{x_m,T}) & \ln(m_{x_m,T+1}) & \dots & \ln(m_{x_m,T+h}) \end{pmatrix} \simeq$$

$$\begin{pmatrix} \alpha_{x_1} \\ \alpha_{x_2} \\ \alpha_{x_3} \\ \vdots \\ \alpha_{x_m} \end{pmatrix} + \begin{pmatrix} \beta_{x_1} \\ \beta_{x_2} \\ \beta_{x_3} \\ \vdots \\ \beta_{x_m} \end{pmatrix} \begin{pmatrix} \kappa_{t_1} & \kappa_{t_2} & \dots & \kappa_T & \kappa_{T+1} & \dots & \kappa_{T+h} \end{pmatrix}$$

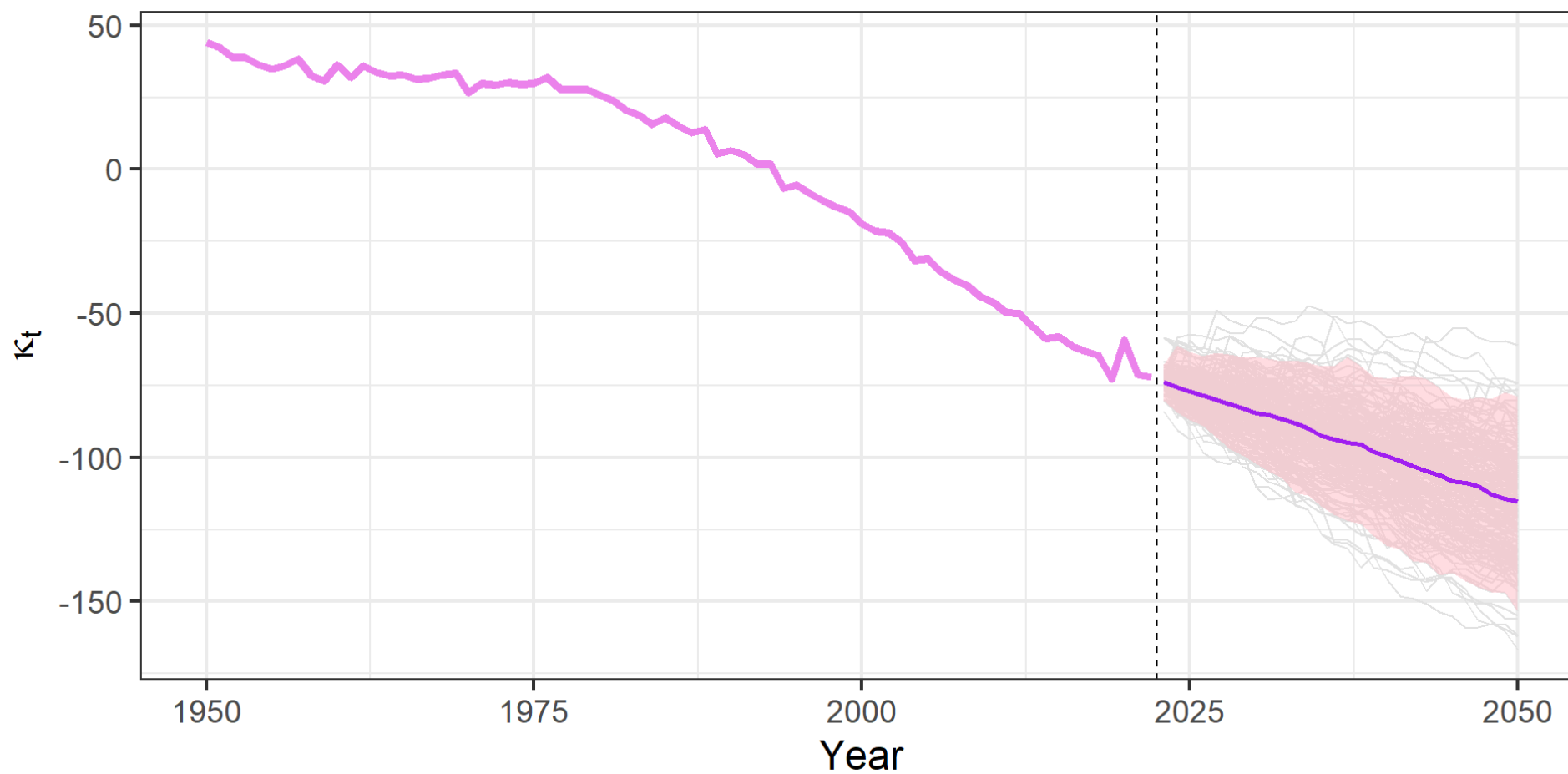




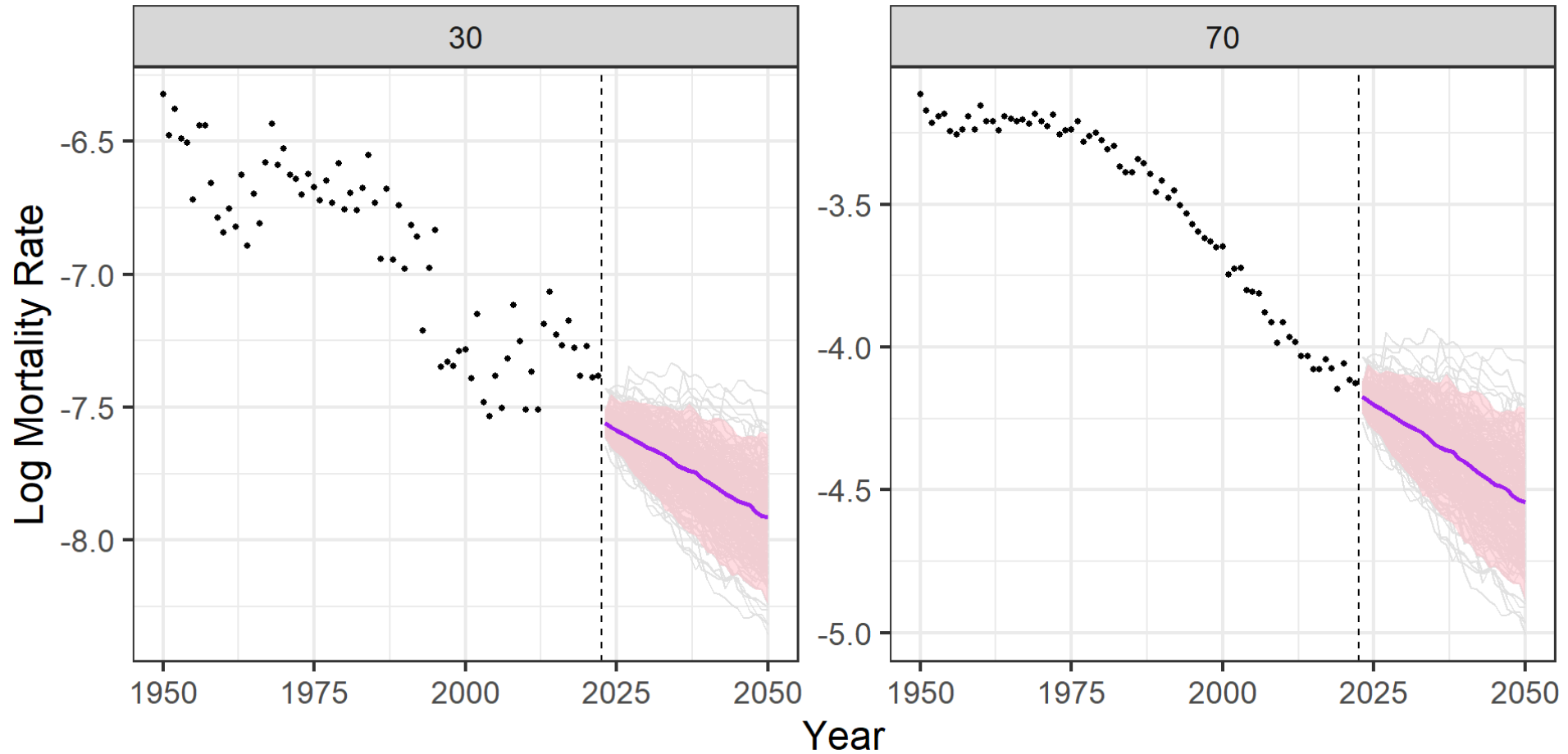
# $\hat{\kappa}_t$ simulations



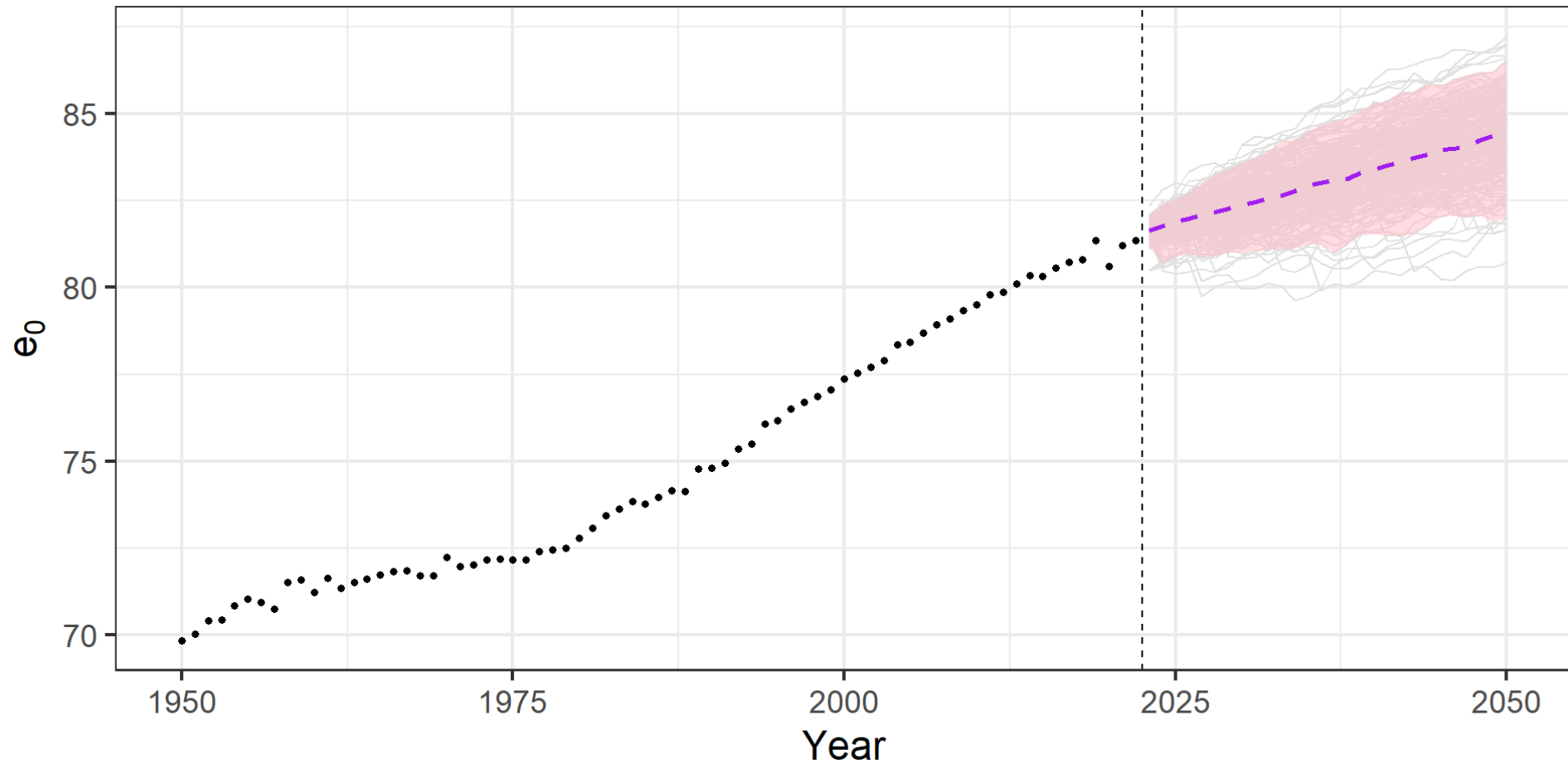
# $\hat{\kappa}_t$ 95% PIs



# From $\hat{\kappa}_t$ to $\ln(\hat{m}_{xt})$



# From $\hat{\kappa}_t$ to $\hat{e}_0$



# The LC model: a summary

- **Advantages:**
  - simple & powerful method: forecast rates derived by modeling single time index by standard time-series model
  - linear time index very often captures well historical decline in mortality
  - stochastic model  $\Rightarrow$  probabilistic intervals
  - extrapolative approach, no expert opinions more accurate than previous methodologies
- **Disadvantages:**
  - jump-off bias
  - Normality assumption (from SVD)
  - jagged fitted and forecast age profile, lacking smoothness
  - fixed age-pattern of mortality decline
  - rigid structure
- Several extensions proposed to overcome some of these issues





# Some LC extensions

- “jump-off” bias: use observed jump-off rates (Lee and Miller 2001)
- Normality assumption: Poisson LC (Brouhns et al. 2002)
- jagged fitted and forecast age profile, lacking smoothness: Smooth LC (Delwarde et al. 2007)
- fixed age-pattern of mortality decline: Li et al. (2013)
- rigid structure



# Other LC extensions (single pop)

- Booth et al. (2002): adjusting  $\kappa_t$  to match the age-at-death distribution & determining optimal fitting period
- Renshaw and Haberman (2003): adding more than one principal components, i.e.  

$$\ln(m_{x,t}) = \alpha_x + \sum_k \beta_x^k \kappa_t^k$$
- Koissi et al. (2006): residual bootstrap to include parameter uncertainty in forecasts
- Renshaw and Haberman (2006): including cohort effects, i.e.  

$$\ln(m_{x,t}) = \alpha_x + \beta_x^{(1)} \kappa_t + \beta_x^{(0)} \gamma_{t-x}$$
- Hyndman and Ullah (2007): smooth underlying data (functional data) & additional principal components
- Camarda and Basellini (2021): smoothing, decomposing and forecasting the three components of mortality (childhood, early-adulthood and senescence), i.e.  

$$m_{x,t} = \sum_k \exp(\alpha_x^k + \beta_x^k \kappa_t^k)$$
- for a comprehensive review, see Basellini et al. (2023)



# Forecast evaluation

- How do we evaluate demographic forecasts?
- Option 1: wait until the future occurs (e.g., 2050) to find out
  - clearly not ideal :)
- Option 2: out-of-sample forecasting
  - from the observed time series  $\mathbf{y} = [y_1, y_2, \dots, y_{n-h}, \dots, y_T]$ , suppose that you have observed only the first  $n - h$  data points, forecast  $h$  years in the future and compare the forecasts with the withheld data points  $y_{n-h+1}, \dots, y_T$
  - $\mathbf{y}$  can refer to any forecast measure (log-rates, rates, summary measures)



# Point forecast accuracy

- Let  $e_t = \hat{y}_t - y_t$  denote the forecast error, with  $\hat{y}_t$  being the central forecast and  $y_t$  the withheld data point  $t$
- Let  $p_t = 100e_t/y_t$  be the percentage error
- Compute one of the following measures of point forecast accuracy:
  - Mean Square Error (MSE) =  $\frac{1}{h} \sum_{t=1}^h e_t^2$
  - Root MSE (RMSE) =  $\sqrt{\text{MSE}}$
  - Mean Absolute Error (MAE) =  $\frac{1}{h} \sum_{t=1}^h |e_t|$
  - Mean Absolute Percentage Error (MAPE) =  $\frac{1}{h} \sum_{t=1}^h |p_t|$
- The smaller the error, the better

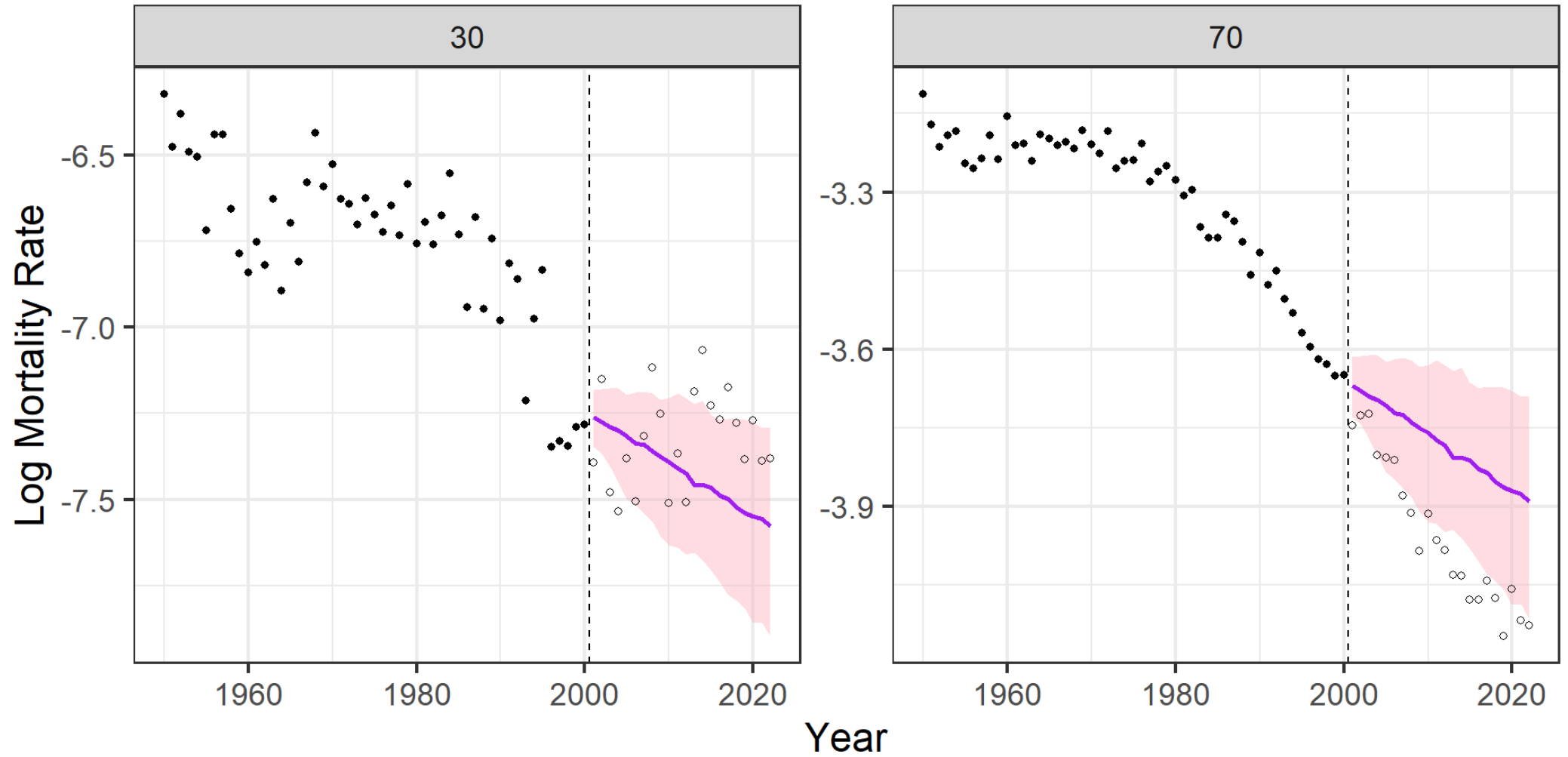


# Interval forecast accuracy

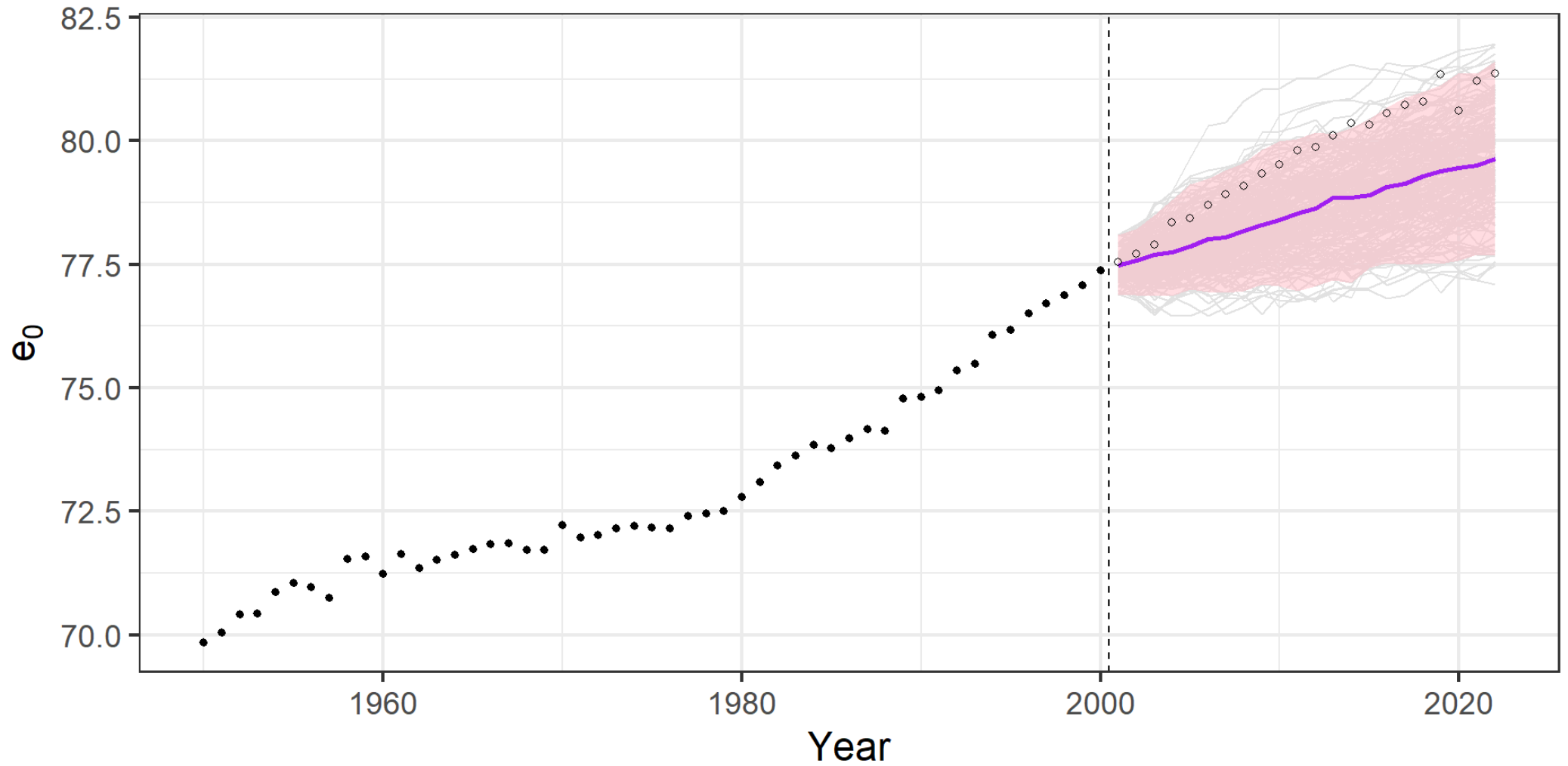
- Compare withheld data points with the forecast  $\alpha\%$  prediction interval (PI):
  - Empirical Coverage Probability (ECP): proportion of observations falling into the PI
  - Coverage Probability Deviance (CPD) =  $|\alpha - \text{ECP}|$ , with  $\text{CPD} \in [0, \alpha]$
- The smaller the CPD, the better



# LC forecast evaluation



# LC forecast evaluation



# Day 4 assignment

## Assignment

8. Load the mortality data [MORTSWE.Rdata](#), and focus on male mortality from 1950 to 2000 for ages  $x \leq 100$ . Fit and forecast mortality with the LC method, and produce forecasts up to 2022. Evaluate the forecast accuracy of the method by computing a point forecast accuracy measure of your choice on a single measure of interest (e.g. log-rates or  $e_o$ ).
9. Consider again the setting of Exercise 8 (i.e. male mortality from 1950 to 2000 for ages  $x \leq 100$ ). Compare the forecast accuracy of two different methods that we have seen during this course. Which model is more accurate?

