Validate performance of Lee-Carter method to forecast mortality

Afternoon lab on *Forecasting mortality*

As part of IMPRS-PHDS course on *Population Health*

Max Planck Institute for Demographic Research, November 27, 2019

Lecturer: Christina Bohk-Ewald

Source: https://github.com/christina-bohk-ewald/2019-IMPRS-PHDS-forecasting-mortality

Table of content:

1. Some preparations in R

2. Load and explore US female mortality

3. Function to estimate and forecast mortality with the Lee-Carter method:

estimate_forecast_mortality_with_LC

4. Function to calculate life expectancy at birth:

lexp.age.specific

5. Function to translate LC mortality into LC life expectancy at birth:

estimate_forecast_e0_with_LC

6. Apply functions (steps 3 through 5) to estimate and forecast mortality and life expectancy at birth

7. Validate forecast performance of Lee-Carter method in the context of US female mortality

1. Some preparations in R

Open a new script in R and save it to a folder of your choice.

Create a filepath to a folder where you would like to save your outcome. For example,

```
the.plot.path <- c("C:/plots")
```

Set the working directory to this outcome path

```
setwd(the.plot.path)
```

2. Load and explore US female mortality

Install and load the fds package to access data of the Human Mortality Database

```
library(fds)
```

Load US female mortality by age, 0 through 110, and calendar year, 1933 through 2017, from the HMD

```
usa_wom <- read.hmd(country=params$country, sex="Female", file = "Mx_1x1.txt",
username=your_username, password=your_password, yname="mortality rates")
```

Explore data object usa_wom. For example,

```
head(usa_wom[[2]][,as.character(1933:1940)])
```

```
##       1933     1934     1935     1936     1937     1938     1939     1940
## 0 0.054177 0.060211 0.053557 0.055160 0.053895 0.050748 0.046848 0.046847
## 1 0.008866 0.009894 0.008334 0.008082 0.007856 0.007401 0.005904 0.005263
## 2 0.004025 0.004540 0.003766 0.003923 0.003690 0.003322 0.002644 0.002471
## 3 0.002869 0.002979 0.002756 0.002675 0.002513 0.002233 0.001896 0.001679
## 4 0.002230 0.002236 0.002326 0.002154 0.001951 0.001785 0.001544 0.001359
## 5 0.001852 0.001858 0.001929 0.001758 0.001604 0.001458 0.001310 0.001158
```

3. Function to estimate and forecast mortality with the Lee-Carter method:

```
estimate_forecast_mortality_with_LC <- function(obs_mortality,
                                base_start,
                                base_end,
                                number_sim,
                                years_ahead){
    ## 1. fit mortality in base period:
    log_usa_wom <- log(obs_mortality[,as.character(base_start:base_end)],base=exp(1))
    mean_log_usa_wom <- rowMeans(log_usa_wom)
    centered_log_usa_wom <- log_usa_wom - mean_log_usa_wom
    svd_centered_log_usa_wom <- svd(t(centered_log_usa_wom))
    sum_v <- sum(svd_centered_log_usa_wom$v[,1])
    bx <- svd_centered_log_usa_wom$v[,1]/sum_v
    kt <- svd_centered_log_usa_wom$d[1] * svd_centered_log_usa_wom$u[,1] * sum_v
    names(kt) <- (base_start+1):base_end
    ax <- mean_log_usa_wom
    ## 2. forecast mortality:
    fit <- summary(lm(kt[2:length(kt)]-kt[1:(length(kt)-1)] ~ 1))
    kt_drift <- fit$coefficients[1,1]
    sec <- fit$coefficients[1,2]
    see <- fit$sigma
    fitted <- kt[1:(length(kt)-1)] + kt_drift
    residuals <- kt[2:length(kt)]-(kt[1:(length(kt)-1)]+kt_drift)
    kt_fitted_self <- fitted
    set.seed(1234)
    kt_trajectories <- matrix(NA,nr=number_sim,nc=years_ahead)
    for(run in 1:number_sim){
```

```
        for(year in 1:1){

            current.epsilon <- rnorm(n=1,mean=0,sd=see)

            kt_trajectories[run,year] <- kt[length(kt)] + kt_drift + current.epsilon*year

        }

        for(year in 2:years_ahead){

            current.epsilon <- rnorm(n=1,mean=0,sd=see)

            kt_trajectories[run,year] <- kt_trajectories[run,(year-1)] + kt_drift
+ current.epsilon

        }

        }

    kt_trajectories_median <- apply(kt_trajectories,2,median)

    kt_trajectories_low <- apply(X=kt_trajectories,2,
FUN=function(X){quantile(X,probs=0.1)})

    kt_trajectories_up <- apply(X=kt_trajectories,2,
FUN=function(X){quantile(X,probs=0.9)})

    ## 3. normalize kt:

    kt_base_forecast <- c(kt_fitted_self , kt_trajectories_median)

    names(kt_base_forecast) <- (base_start+1):(base_end+years_ahead)

    kt_base_forecast_normalized <- kt_base_forecast -
kt_base_forecast[as.character(base_end)]

    kt_base_forecast_low <- c(kt_fitted_self , kt_trajectories_low)

    names(kt_base_forecast_low) <- (base_start+1):(base_end+years_ahead)

    kt_base_forecast_low_normalized <- kt_base_forecast_low -
kt_base_forecast_low[as.character(base_end)]

    kt_base_forecast_up <- c(kt_fitted_self , kt_trajectories_up)

    names(kt_base_forecast_up) <- (base_start+1):(base_end+years_ahead)

    kt_base_forecast_up_normalized <- kt_base_forecast_up -
kt_base_forecast_up[as.character(base_end)]

    ## 4. save LC estimated mortality:

    mx_LC_estimated <- matrix(NA,nr=nrow(obs_mortality),
```

```
nc=length((base_start+1):base_end))

    colnames(mx_LC_estimated) <- (base_start+1):base_end

    rownames(mx_LC_estimated) <- names(obs_mortality[,1])

    for(year in (base_start+1):base_end){

        mx_LC_estimated[,as.character(year)] <- ax+bx*kt_base_forecast[as.character(year)]

    }

    ## 5. LC forecasted mortality:

    mx_LC_forecasted <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))

    colnames(mx_LC_forecasted) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted) <- names(obs_mortality[,1])

    mx_LC_forecasted_low <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))

    colnames(mx_LC_forecasted_low) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted_low) <- names(obs_mortality[,1])

    mx_LC_forecasted_up <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))

    colnames(mx_LC_forecasted_up) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted_up) <- names(obs_mortality[,1])

    for(year in (base_end+1):(base_end+years_ahead)){

        mx_LC_forecasted[,as.character(year)] <- ax+
bx*kt_base_forecast[as.character(year)]

        mx_LC_forecasted_low[,as.character(year)] <- ax+
bx*kt_base_forecast_low[as.character(year)]

        mx_LC_forecasted_up[,as.character(year)] <- ax+
bx*kt_base_forecast_up[as.character(year)]

    }

    ## 6. LC forecasted mortality, corrected for jump-off bias:

    mx_LC_forecasted_joyAdjusted <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))
```

```r
    colnames(mx_LC_forecasted_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted_joyAdjusted) <- names(obs_mortality[,1])

    mx_LC_forecasted_low_joyAdjusted <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))

    colnames(mx_LC_forecasted_low_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted_low_joyAdjusted) <- names(obs_mortality[,1])

    mx_LC_forecasted_up_joyAdjusted <- matrix(NA,nr=nrow(obs_mortality),
nc=length((base_end+1):(base_end+years_ahead)))

    colnames(mx_LC_forecasted_up_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    rownames(mx_LC_forecasted_up_joyAdjusted) <- names(obs_mortality[,1])

    for(year in (base_end+1):(base_end+years_ahead)){

        mx_LC_forecasted_joyAdjusted[,as.character(year)] <-
log_usa_wom[,as.character(base_end)]+ bx*kt_base_forecast_normalized[as.character(year)]

        mx_LC_forecasted_low_joyAdjusted[,as.character(year)] <-
log_usa_wom[,as.character(base_end)]+bx*kt_base_forecast_low_normalized[as.character(year)]

        mx_LC_forecasted_up_joyAdjusted[,as.character(year)] <-
log_usa_wom[,as.character(base_end)]+bx*kt_base_forecast_up_normalized[as.character(year)]

    }

    ## 7. save and return selected results:

    outcome_to_return <- list()

    outcome_to_return$obs_mortality <- obs_mortality

    outcome_to_return$base_start <- base_start

    outcome_to_return$base_end <- base_end

    outcome_to_return$number_sim <- number_sim

    outcome_to_return$years_ahead <- years_ahead

    outcome_to_return$alphax <- ax

    outcome_to_return$betax <- bx

    outcome_to_return$kappat <- kt

    outcome_to_return$kappat_drift <- kt_drift
```

```
    outcome_to_return$kt_fitted <- kt_fitted_self

    outcome_to_return$kappat_median <- kt_trajectories_median

    outcome_to_return$kappat_low80 <- kt_trajectories_low

    outcome_to_return$kappat_up80 <- kt_trajectories_up

    outcome_to_return$kappat_base_forecast <- kt_base_forecast

    outcome_to_return$kappat_base_forecast_low80 <- kt_base_forecast_low

    outcome_to_return$kappat_base_forecast_up80 <- kt_base_forecast_up

    outcome_to_return$kappat_base_forecast_normalized  <- kt_base_forecast_normalized

    outcome_to_return$kappat_base_forecast_low80_normalized  <-
kt_base_forecast_low_normalized

    outcome_to_return$kappat_base_forecast_up80_normalized  <-
kt_base_forecast_up_normalized

    outcome_to_return$mx_LC_estimated <- mx_LC_estimated

    outcome_to_return$mx_LC_forecasted <- mx_LC_forecasted

    outcome_to_return$mx_LC_forecasted_low80 <- mx_LC_forecasted_low

    outcome_to_return$mx_LC_forecasted_up80 <- mx_LC_forecasted_up

    outcome_to_return$mx_LC_forecasted_joyAdjusted <- mx_LC_forecasted_joyAdjusted

    outcome_to_return$mx_LC_forecasted_low80_joyAdjusted <-
mx_LC_forecasted_low_joyAdjusted

    outcome_to_return$mx_LC_forecasted_up80_joyAdjusted <-
mx_LC_forecasted_up_joyAdjusted

    return(outcome_to_return)

}
```

4. Function to calculate life expectancy at birth

```
lexp.age.specific <- function(mx,ax){

        qx <- mx/(1+(1-ax)*mx)

            px <- 1-qx

            lx <- c(100000, (cumprod(px)*100000)[1:(length(qx)-1)])

            dx <- c(-diff(lx), lx[length(lx)])

            Lx1 <- lx[-1]+ax[1:(length(ax)-1)]*dx[1:(length(dx)-1)]

            Lx2 <- if(mx[length(mx)] == 0){

                0}else{

                    dx[length(dx)]/mx[length(mx)]

                }

            Lx <- c(Lx1, Lx2)

            Tx <- rev(cumsum(rev(Lx)))

            ex <- Tx/lx

            ex[is.nan(ex)] <- 0

            return(ex)

}
```

5. Function to translate LC mortality into LC life expectancy at birth

Based on output of function *estimate_forecast_mortality_with_LC*.

```
estimate_forecast_e0_with_LC <- function(outcome_estimate_forecast_mortality_with_LC){

    base_start <- outcome_estimate_forecast_mortality_with_LC$base_start

    base_end <- outcome_estimate_forecast_mortality_with_LC$base_end

    years_ahead <- outcome_estimate_forecast_mortality_with_LC$years_ahead

    ## load function: lexp.age.specific

    lt_ax <- c(0.06,
rep(0.5,(length(outcome_estimate_forecast_mortality_with_LC$mx_LC_estimated[,1])-1)))

    e0_observed <- matrix(NA,nr=1,nc=length((base_start+1):(base_end)))

    colnames(e0_observed) <- ((base_start+1):(base_end))

    e0_LC_estimated <- matrix(NA,nr=1,nc=length((base_start+1):(base_end)))

    colnames(e0_LC_estimated) <- ((base_start+1):(base_end))

    e0_LC_forecast <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast) <- (base_end+1):(base_end+years_ahead)

    e0_LC_forecast_low80 <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast_low80) <- (base_end+1):(base_end+years_ahead)

    e0_LC_forecast_up80 <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast_up80) <- (base_end+1):(base_end+years_ahead)

    e0_LC_forecast_joyAdjusted <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    e0_LC_forecast_low80_joyAdjusted <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast_low80_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    e0_LC_forecast_up80_joyAdjusted <- matrix(NA,nr=1,nc=years_ahead)

    colnames(e0_LC_forecast_up80_joyAdjusted) <- (base_end+1):(base_end+years_ahead)

    obs_mortality <- outcome_estimate_forecast_mortality_with_LC$obs_mortality

    mx_LC_estimated <- outcome_estimate_forecast_mortality_with_LC$mx_LC_estimated

    for(year in (base_start+1):base_end){
```

```
        e0_observed[1,as.character(year)] <-
lexp.age.specific(obs_mortality[,as.character(year)],lt_ax)[1]

        e0_LC_estimated[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_estimated[,as.character(year)]),lt_ax)[1]

    }

    mx_LC_forecasted <- outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted

    mx_LC_forecasted_low80 <-
outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted_low80

    mx_LC_forecasted_up80 <-
outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted_up80

    mx_LC_forecasted_joyAdjusted <-
outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted_joyAdjusted

    mx_LC_forecasted_low80_joyAdjusted <-
outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted_low80_joyAdjusted

    mx_LC_forecasted_up80_joyAdjusted <-
outcome_estimate_forecast_mortality_with_LC$mx_LC_forecasted_up80_joyAdjusted

    for(year in (base_end+1):(base_end+years_ahead)){

        e0_LC_forecast[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted[,as.character(year)]),lt_ax)[1]

        e0_LC_forecast_low80[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted_low80[,as.character(year)]),lt_ax)[1]

        e0_LC_forecast_up80[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted_up80[,as.character(year)]),lt_ax)[1]

        e0_LC_forecast_joyAdjusted[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted_joyAdjusted[,as.character(year)]),lt_ax)[1]

        e0_LC_forecast_low80_joyAdjusted[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted_low80_joyAdjusted[,as.character(year)]),lt_ax)[1]

        e0_LC_forecast_up80_joyAdjusted[1,as.character(year)] <-
lexp.age.specific(exp(mx_LC_forecasted_up80_joyAdjusted[,as.character(year)]),lt_ax)[1]

    }

    outcome_to_return <- list()

    outcome_to_return$e0_observed <- e0_observed

    outcome_to_return$e0_LC_estimated <- e0_LC_estimated
```

```r
    outcome_to_return$e0_LC_forecast <- e0_LC_forecast

    outcome_to_return$e0_LC_forecast_low80 <- e0_LC_forecast_low80

    outcome_to_return$e0_LC_forecast_up80 <- e0_LC_forecast_up80

    outcome_to_return$e0_LC_forecast_joyAdjusted <- e0_LC_forecast_joyAdjusted

    outcome_to_return$e0_LC_forecast_low80_joyAdjusted <- e0_LC_forecast_low80_joyAdjusted

    outcome_to_return$e0_LC_forecast_up80_joyAdjusted <- e0_LC_forecast_up80_joyAdjusted

    return(outcome_to_return)

}
```

6. Apply functions (steps 3 through 5) to estimate and forecast mortality and life expectancy at birth

We now apply the functions *estimate_forecast_mortality_with_LC* and *estimate_forecast_e0_with_LC* to forecast US female mortality from 2018 to 2067, based on data (1) 1933 to 2017 and (2) 1988 to 2017.

```r
mx_lc_1933_2017_2067 <-
estimate_forecast_mortality_with_LC(obs_mortality=usa_wom[[2]][,as.character(1933:2017)],

                                base_start=1933,

                                base_end=2017,

                                number_sim=1000,

                                years_ahead=50)

e0_lc_1934_2017_2067 <-
estimate_forecast_e0_with_LC(outcome_estimate_forecast_mortality_with_LC=mx_lc_1933_2017_2067)

e0_lc_1934_2017_2067$e0_LC_forecast_joyAdjusted
```

```
##          2018     2019     2020     2021     2022    2023     2024     2025
## [1,] 81.40823 81.55547 81.70143 81.84613 81.98958 82.1318 82.27279 82.41256
##          2026     2027     2028     2029     2030    2031     2032    2033
## [1,] 82.55114 82.68852 82.82473 82.95977 83.09366 83.2264 83.35801 83.4885
##          2034     2035     2036     2037     2038     2039     2040    2041
## [1,] 83.61787 83.74615 83.87333 83.99943 84.12445 84.24842 84.37133 84.4932
##          2042     2043     2044     2045     2046     2047     2048     2049
## [1,] 84.61404 84.73385 84.85264 84.97043 85.08722 85.20302 85.31783 85.43168
##          2050     2051     2052     2053     2054     2055     2056     2057
## [1,] 85.54456 85.65648 85.76746 85.87749 85.98659 86.09477 86.20203 86.30838
##          2058     2059     2060     2061     2062     2063     2064     2065
## [1,] 86.41383 86.51838 86.62205 86.72483 86.82675 86.92779 87.02798 87.12732
##          2066     2067
## [1,] 87.22581 87.32346
```

```r
mx_lc_1988_2017_2067 <-
estimate_forecast_mortality_with_LC(obs_mortality=usa_wom[[2]][,as.character(1988:2017)],

                                    base_start=1988,

                                    base_end=2017,

                                    number_sim=1000,

                                    years_ahead=50)

e0_lc_1989_2017_2067 <-
estimate_forecast_e0_with_LC(outcome_estimate_forecast_mortality_with_LC=mx_lc_1988_2017_2067)

e0_lc_1989_2017_2067$e0_LC_forecast_joyAdjusted
```

```
##          2018     2019     2020     2021     2022     2023     2024     2025
## [1,] 81.43092 81.52036 81.60937 81.69795 81.78611 81.87385 81.96116 82.04806
##          2026    2027     2028     2029     2030    2031    2032     2033
## [1,] 82.13454 82.2206 82.30624 82.39147 82.47629 82.5607 82.6447 82.72829
##          2034     2035     2036     2037     2038     2039     2040     2041
## [1,] 82.81148 82.89426 82.97663 83.05861 83.14018 83.22136 83.30213 83.38251
##          2042     2043     2044     2045    2046     2047     2048     2049
## [1,] 83.46249 83.54208 83.62128 83.70008 83.7785 83.85653 83.93417 84.01142
##          2050     2051     2052     2053     2054     2055     2056     2057
## [1,] 84.08829 84.16478 84.24088 84.31661 84.39195 84.46692 84.54151 84.61573
##          2058     2059     2060     2061     2062     2063     2064     2065
## [1,] 84.68957 84.76304 84.83614 84.90887 84.98123 85.05322 85.12485 85.19612
##          2066     2067
## [1,] 85.26702 85.33756
```

7. Validate forecast performance of Lee-Carter method in the context of US female mortality.

We would like to make as many validation forecasts as possible, taking base period and forecast horizons that comprise 30 years each, so that we have jump-off years 1962 through 1987.

```r
e0_1934_2017 <- e0_lc_1934_2017_2067$e0_observed

validate_e0_observed <- matrix(NA,nr=length(1933:1958),nc=30)

rownames(validate_e0_observed) <- 1933:1958

validate_e0_LC_forecast_joyAdjusted <- matrix(NA,nr=length(1933:1958),nc=30)

rownames(validate_e0_LC_forecast_joyAdjusted) <- 1933:1958

for(base_start in 1933:1958){

    validate_e0_observed[as.character(base_start),] <-
e0_1934_2017[1,(as.character((base_start+30):(base_start+30+29)))]

    current_mx_LC_estimate_forecast <-
```

```
estimate_forecast_mortality_with_LC(obs_mortality=
usa_wom[[2]][,as.character(base_start:(base_start+29))],

                                base_start=base_start,

                                base_end=(base_start+29),

                                number_sim=1000,

                                years_ahead=30)

    current_LC_e0_forecast <-
estimate_forecast_e0_with_LC(outcome_estimate_forecast_mortality_with_LC=
current_mx_LC_estimate_forecast)

    validate_e0_LC_forecast_joyAdjusted[as.character(base_start),] <-
current_LC_e0_forecast$e0_LC_forecast_joyAdjusted

}

error <- validate_e0_LC_forecast_joyAdjusted - validate_e0_observed

error[1:5,1:5]
```

```
##              [,1]         [,2]        [,3]        [,4]       [,5]
## 1933  0.133400683  0.086913290  0.22341276  0.44613568  0.3266086
## 1934 -0.555391640 -0.408271618 -0.17512998 -0.28443764  0.2218182
## 1935  0.003139586  0.239523880  0.13345724  0.64295061  0.5835606
## 1936  0.038919080 -0.051653315  0.47297610  0.42837326  0.4385689
## 1937 -0.525232513 -0.009997288 -0.06359627 -0.06201831 -0.1378467
```

We finally plot the forecast errors by jump-off year

```
colfunc <- colorRampPalette(c("black","blue","red"))

blackblue <- colfunc(length(1:30))

par(fig = c(0,1,0,1), las=1, mai=c(0.8,0.8,0.8,0.8))

plot(x=-100,y=-100,xlim=c(0,30),ylim=c(-2,3),xlab="Forecast horizon",ylab="",
main="Forecast error",axes=FALSE)

    axis(side=1,at=seq(0,30,1),labels=FALSE,lwd=1,pos=-2)

    axis(side=1,at=seq(0,30,5),labels=seq(0,30,5),lwd=3,pos=-2)

    axis(side=2,at=seq(-2,3,0.5),labels=seq(-2,3,0.5),lwd=1,pos=0)

    axis(side=2,at=seq(-2,3,1),labels=seq(-2,3,1),lwd=3,pos=0)

    for(base_start in 1933:1958){

        lines(x=1:30,y=error[as.character(base_start),],col=blackblue[base_start-1932])

        points(x=1:30,y=error[as.character(base_start),],col=blackblue[base_start-1932],
```

```
lwd=2)
    }

    segments(x0=1,x1=30,y0=0,y1=0,lwd=2)

    points(x=rep(2.5,30),y=seq(1.5,2.5,length=30),col=blackblue[1:30],lwd=2)

    text(x=4.5,y=2.6,"Jump-off year:",pos=3)

    text(x=4.5,y=1.5,"1962",pos=4)

    text(x=4.5,y=2.5,"1987",pos=4)
```