Implement the Lee-Carter method to fit and 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

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, explore, and plot 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
```

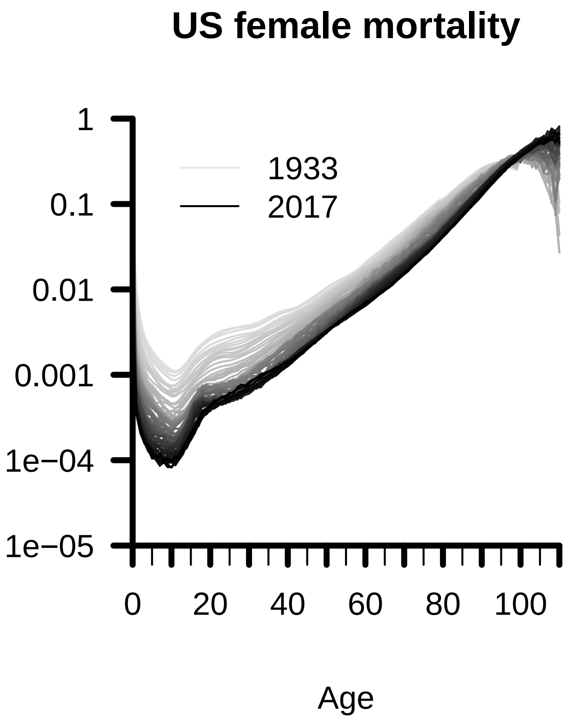
Plot US female mortality rates on log scale

At first create a color palette that ranges from gray to black between the years 1933 and 2017

```
colfunc <- colorRampPalette(c(gray(0.9), "black"))

blackgray <- colfunc(length(1933:2017))
```

And then depict US female mortality in a figure

```
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(0,110),ylim=c(log(0.00001,base=exp(1)),log(1,base=exp(1))),
xlab="Age",ylab="",main="US female mortality", cex=0.8, axes=FALSE)

for(year in 1933:2017){
    lines(x=0:110,y=log(usa_wom[[2]][,as.character(year)],base=exp(1)),
col=blackgray[(year-1932)])
}

legend(x=5,y=log(0.75,base=exp(1)),col=c(blackgray[1],blackgray[length(1933:2017)]),
lty=1,horiz=FALSE,bty="n",c("1933","2017"))

axis(side=1,at=seq(0,110,5),labels=FALSE,lwd=1,pos=log(0.00001,base=exp(1)))

axis(side=1,at=seq(0,110,10),labels=seq(0,110,10),lwd=3,pos=log(0.00001,base=exp(1)))

axis(side=2,at=log(c(0.00001,0.0001,0.001,0.01,0.1,1),base=exp(1)),
labels=c(0.00001,0.0001,0.001,0.01,0.1,1),lwd=3,pos=0)
```



3. Fit US female mortality with the Lee-Carter model

To understand how the Lee-Carter method fits mortality we implement the steps 1.1 through 1.8 of the *morning lecture*.

Step 1.2:

```
log_usa_wom <- log(usa_wom[[2]],base=exp(1))
```

Step 1.3:

```
mean_log_usa_wom <- rowMeans(log_usa_wom)

ax <- mean_log_usa_wom
```

Step 1.4:

```
centered_log_usa_wom <- log_usa_wom - ax
```

Step 1.5.1:

```
svd_centered_log_usa_wom <- svd(t(centered_log_usa_wom))

names(svd_centered_log_usa_wom)
```

```
## [1] "d" "u" "v"
```

Step 1.5.2:

```
sum_v <- sum(svd_centered_log_usa_wom$v[,1])

bx <- svd_centered_log_usa_wom$v[,1]/sum_v
```

Step 1.5.3:

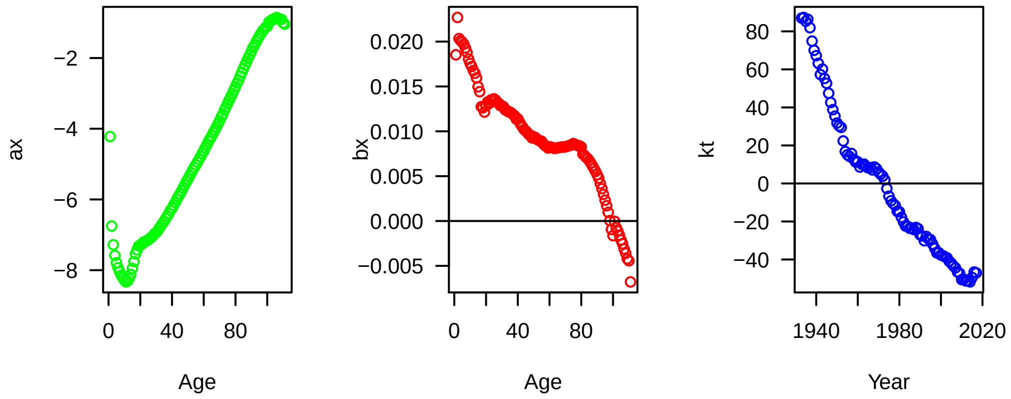
```
kt <- svd_centered_log_usa_wom$d[1] * svd_centered_log_usa_wom$u[,1] * sum_v
```

Step 1.5.4:

```
sum(bx)

sum(kt)
```

Step 1.6:

```
par(mfrow=c(1,3), las=1)

plot(ax,xlab="Age",col="green")

plot(bx,xlab="Age",col="red")
abline(h=0)

plot(x=1933:2017,y=kt,xlab="Year",col="blue")
abline(h=0)
```



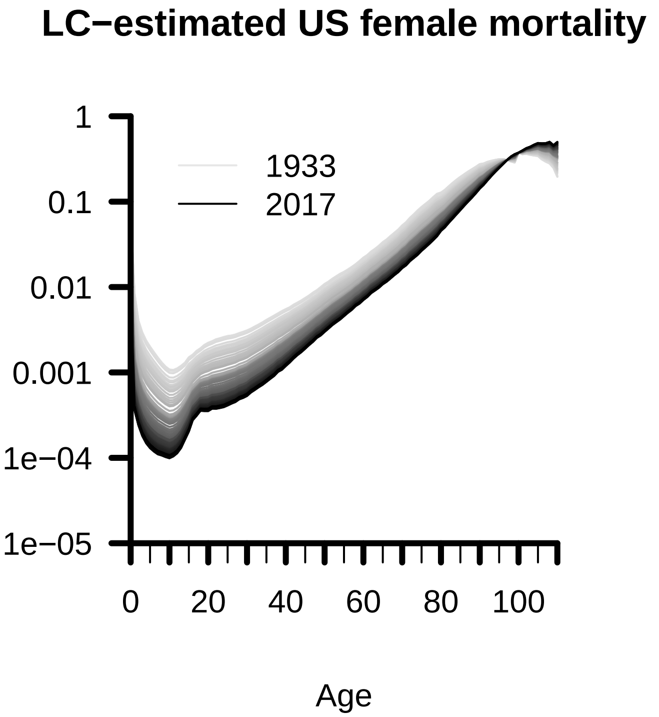
Step 1.7: We choose *none adjustment*.

Step 1.8:

```
fitted_log_usa_wom <- matrix(NA,nr=length(0:110),nc=length(1933:2017))

rownames(fitted_log_usa_wom) <- 0:110

colnames(fitted_log_usa_wom) <- 1933:2017

for(year in 1:85){
    fitted_log_usa_wom[,year] <- ax+bx*kt[year]
}
```

Step 1.8 continued:

```
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(0,110),ylim=c(log(0.00001,base=exp(1)),log(1,base=exp(1))),
xlab="Age",ylab="",main="LC-estimated US female mortality",axes=FALSE)

for(year in 1933:2017){
    lines(x=0:110,y=fitted_log_usa_wom[,as.character(year)],
col=blackgray[(year-1932)])
}

legend(x=5,y=log(0.75,base=exp(1)),col=c(blackgray[1],blackgray[length(1933:2017)]),
lty=1,horiz=FALSE,bty="n",c("1933","2017"))

axis(side=1,at=seq(0,110,5),labels=FALSE,lwd=1,pos=log(0.00001,base=exp(1)))

axis(side=1,at=seq(0,110,10),labels=seq(0,110,10),lwd=3,pos=log(0.00001,base=exp(1)))

axis(side=2,at=log(c(0.00001,0.0001,0.001,0.01,0.1,1),base=exp(1)),
labels=c(0.00001,0.0001,0.001,0.01,0.1,1),lwd=3,pos=0)
```



4. Forecast US female mortality with the Lee-Carter model

To understand how the Lee-Carter method forecasts mortality we implement the steps 2.1 through 2.3 of the *morning lecture.*

Step 2.1:

```
fit_kt_model <- summary(lm(kt[2:85]-kt[1:84] ~ 1))

kt_drift <- fit_kt_model$coefficients[1,1]

sec <- fit_kt_model$coefficients[1,2]

see <- fit_kt_model$sigma

fitted_kt <- kt[1:84] + kt_drift

residuals_kt <- fit_kt_model$residuals
```

Some explanations for fitting κ_t in base period:

- Use *~1* in the lm function to fit linear model with only an intercept
- kt_drift is the average of the first order differences of κ_t
- sec is the standard error of the coefficient (kt_drift)
- see is the square root of the estimated variance of the residuals (/ random error)

Below you can also find alternative ways to compute kt_drift, residuals_kt, and see

```
kt_drift <- mean(diff(kt))

residuals_kt <- kt[2:85]-(kt[1:84]+kt_drift)

see <- sqrt((1/(84-1))*sum(residuals_kt^2))
```

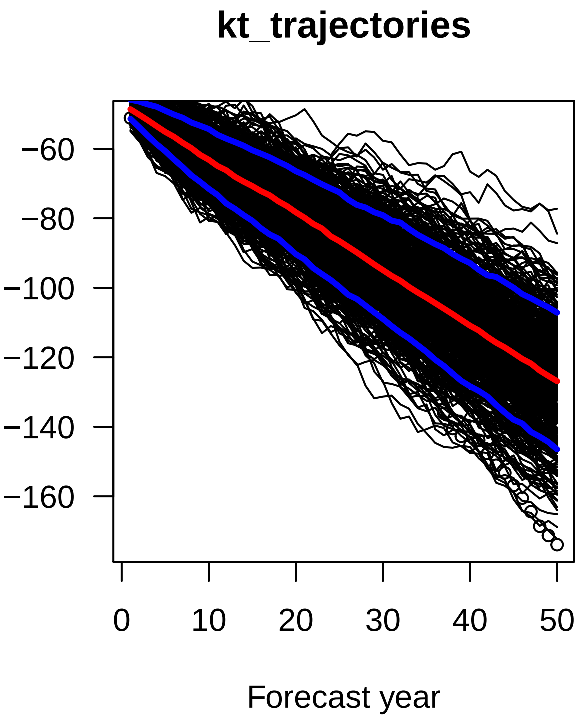
Step 2.2:

```
h <- 50

nsim <- 1000

kt_trajectories <- matrix(NA,nr=nsim,nc=h)

set.seed(1234)

for(run in 1:nsim){
    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)
    }
    for(year in 2:h){
        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_low80 <- apply(X=kt_trajectories,2,FUN=function(X){quantile(X,probs=0.1)})

kt_trajectories_up80 <- apply(X=kt_trajectories,2,FUN=function(X){quantile(X,probs=0.9)})
```

Some explanations for forecasting κ_t based on simulation:

- h is the length of the forecast horizon
- nsim is the number of tracectories to simulate, N
- kt_trajectories is a matrix that stores simulated trajectories for κ_{t+s}
- set.seed(1234) sets a seed to make random simulation reproducible
- rnorm() is the random normal distribution for ϵ_t
- kt_trajectories_median contains median forecast of κ_{t+s}
- kt_trajectories_low80 contains lower boundary of the 80% prediction interval of κ_{t+s}
- kt_trajectories_up80 contains upper boundary of the 80% prediction interval of κ_{t+s}

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

plot(1:h,kt_trajectories[1,], xlab="Forecast year",ylab="",main="kt_trajectories",cex=0.8)

for(run in 2:500){
    lines(1:h,kt_trajectories[run,])
}

lines(1:h,kt_trajectories_median,col="red",lwd=3)

lines(1:h,kt_trajectories_low80,col="blue",lwd=3)

lines(1:h,kt_trajectories_up80,col="blue",lwd=3)
```



Step 2.3

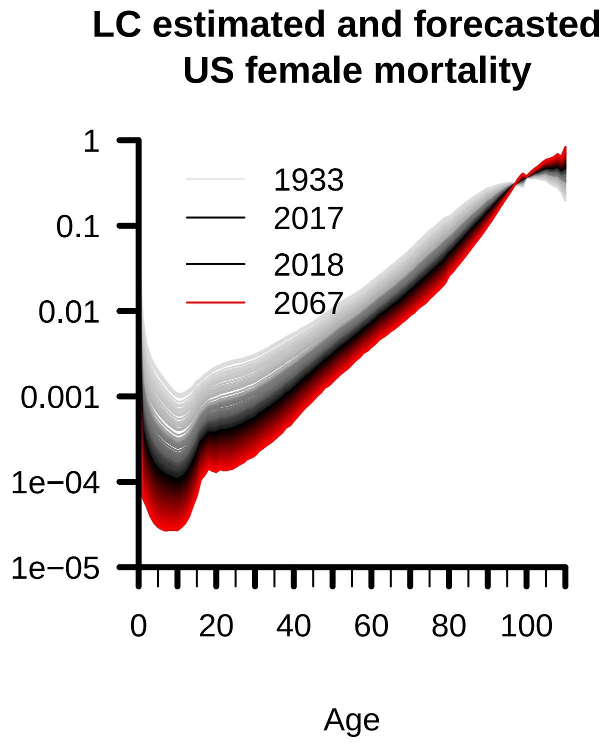
```
forecast_log_usa_wom_median <- matrix(NA,nr=length(0:110),nc=length(2018:2067))

rownames(forecast_log_usa_wom_median) <- 0:110

colnames(forecast_log_usa_wom_median) <- 2018:2067

forecast_log_usa_wom_low80 <- matrix(NA,nr=length(0:110),nc=length(2018:2067))

rownames(forecast_log_usa_wom_low80) <- 0:110

colnames(forecast_log_usa_wom_low80) <- 2018:2067

forecast_log_usa_wom_up80 <- matrix(NA,nr=length(0:110),nc=length(2018:2067))
```

```r
rownames(forecast_log_usa_wom_up80) <- 0:110

colnames(forecast_log_usa_wom_up80) <- 2018:2067

for(year in 1:h){
    forecast_log_usa_wom_median[,year] <- ax+bx*kt_trajectories_median[year]

    forecast_log_usa_wom_low80[,year] <- ax+bx*kt_trajectories_low80[year]

    forecast_log_usa_wom_up80[,year] <- ax+bx*kt_trajectories_up80[year]
}
```

Some explanations for forecasting $m_{x,t}$ 50 years ahead:

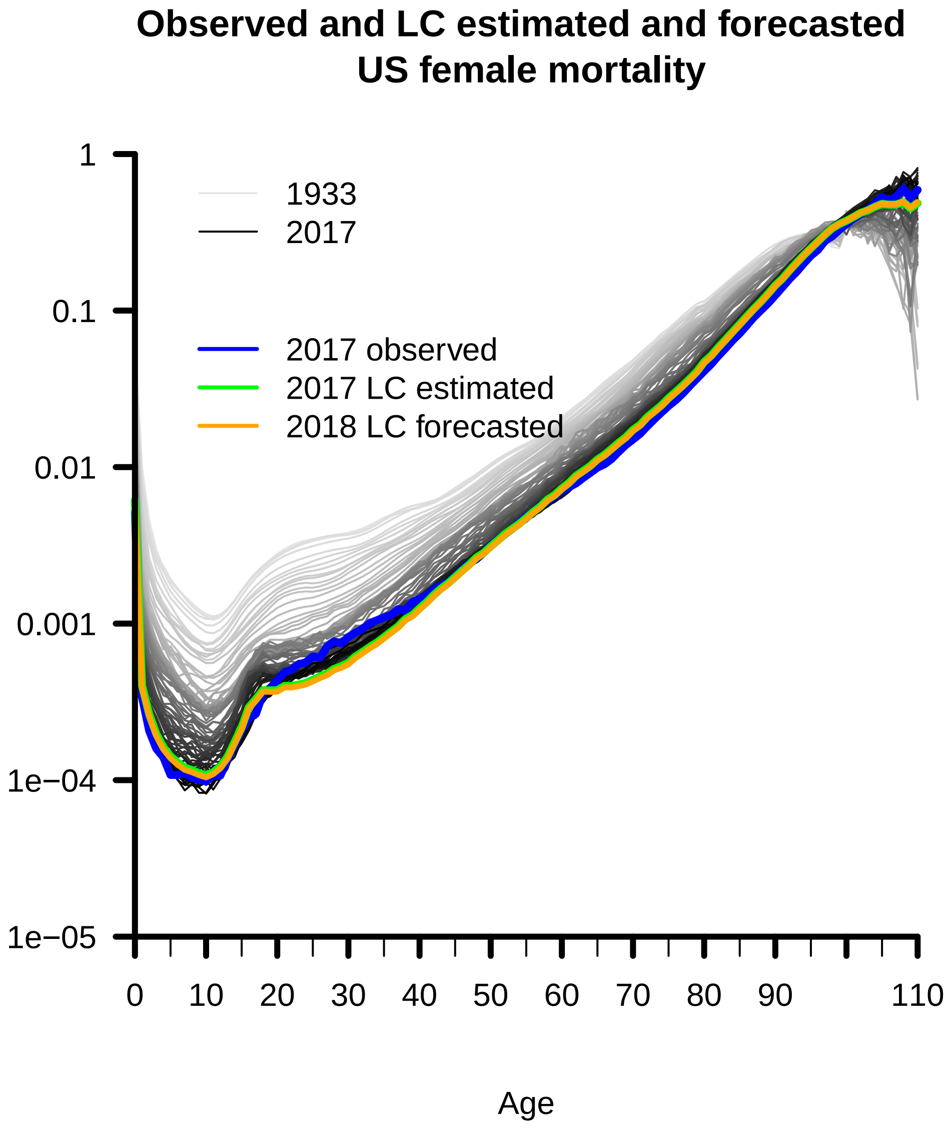
- forecast_log_usa_wom_median contains the median forecasts of $m_{x,t}$
- forecast_log_usa_wom_low80 contains the lower boundary of the 80% prediction intervals of $m_{x,t}$
- forecast_log_usa_wom_up80 contains the upper boundary of the 80% prediction intervals of $m_{x,t}$

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

blackred <- colfunc(length(1:50))

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(0,110),ylim=c(log(0.00001,base=exp(1)),log(1,base=exp(1))),
xlab="Age",ylab="",main="LC estimated and forecasted \n US female mortality",
axes=FALSE, cex=0.6)

    for(year in 1933:2017){
        lines(x=0:110,y=fitted_log_usa_wom[,as.character(year)],
col=blackgray[(year-1932)])
    }

    for(year in 2018:2067){
        lines(x=0:110,y=forecast_log_usa_wom_median[,as.character(year)],
col=blackred[(year-2017)])
    }

    legend(x=5,y=log(0.99,base=exp(1)),col=c(blackgray[1],blackgray[length(1933:2017)]),
lty=1,horiz=FALSE,bty="n",c("1933","2017"))

    legend(x=5,y=log(0.1,base=exp(1)),col=c(blackred[1],blackred[50]),lty=1,horiz=FALSE,
bty="n",c("2018","2067"))

    axis(side=1,at=seq(0,110,5),labels=FALSE,lwd=1,pos=log(0.00001,base=exp(1)))

    axis(side=1,at=seq(0,110,10),labels=seq(0,110,10),lwd=3,pos=log(0.00001,base=exp(1)))

    axis(side=2,at=log(c(0.00001,0.0001,0.001,0.01,0.1,1),base=exp(1)),
labels=c(0.00001,0.0001,0.001,0.01,0.1,1),lwd=3,pos=0)
```



5. Check transition between actual and forecasted US female mortality, 2017-2018

```
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(0,110),ylim=c(log(0.00001,base=exp(1)),log(1,base=exp(1))),
xlab="Age",ylab="",main="Observed and LC estimated and forecasted \n US female mortality",
axes=FALSE, cex=0.6)

    for(year in 1933:2017){
        lines(x=0:110,y=log_usa_wom[,as.character(year)],
col=blackgray[(year-1932)])
    }

    lines(x=0:110,y=log_usa_wom[,as.character(2017)],col="blue",lwd=4)

    lines(x=0:110,y=fitted_log_usa_wom[,as.character(2017)],col="green",lwd=4)

    lines(x=0:110,y=forecast_log_usa_wom_median[,as.character(2018)],col="orange",lwd=3)

    legend(x=5,y=log(0.99,base=exp(1)),col=c(blackgray[1],blackgray[length(1933:2017)]),
lty=1,horiz=FALSE,bty="n",c("1933","2017"))

    legend(x=5,y=log(0.1,base=exp(1)),col=c("blue","green","orange"),lty=1,horiz=FALSE,
bty="n",c("2017 observed","2017 LC estimated","2018 LC forecasted"),lwd=2)

    axis(side=1,at=seq(0,110,5),labels=FALSE,lwd=1,pos=log(0.00001,base=exp(1)))
    axis(side=1,at=seq(0,110,10),labels=seq(0,110,10),lwd=3,pos=log(0.00001,base=exp(1)))

    axis(side=2,at=log(c(0.00001,0.0001,0.001,0.01,0.1,1),base=exp(1)),
labels=c(0.00001,0.0001,0.001,0.01,0.1,1),lwd=3,pos=0)
```



6. Avoid jump-off bias using observed mortality in jump-off year, 2017, and normalizing κ_t

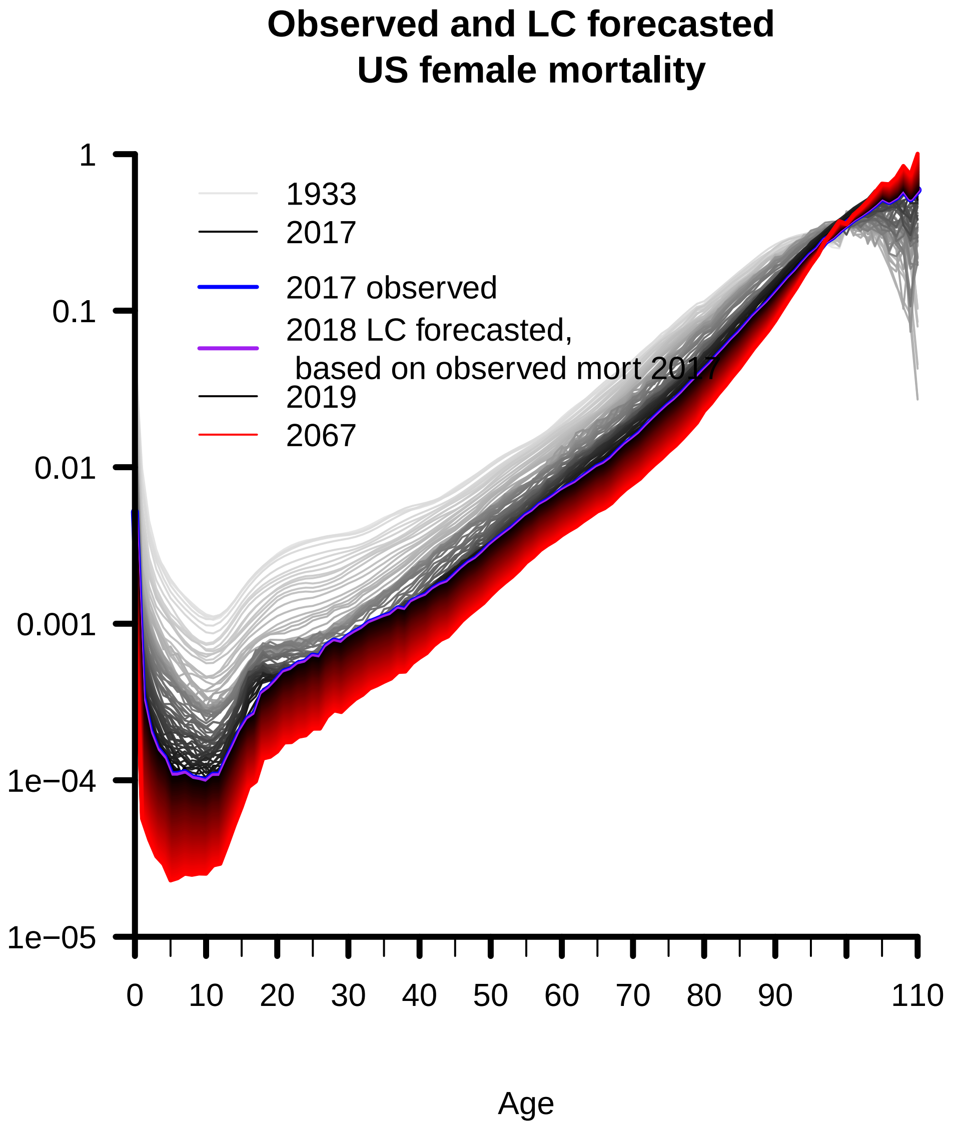
We first normalize κ_t to be 0 in the jump-off year 2017:

```
kt_base_forecast <- c(fitted_kt , kt_trajectories_median)

names(kt_base_forecast) <- 1934:2067 ## to 1934 ... to 2067

kt_base_forecast_normalized <- kt_base_forecast -  kt_base_forecast[as.character(2017)]

kt_base_forecast_low <- c(fitted_kt, kt_trajectories_low80)

names(kt_base_forecast_low) <- 1934:2067 ## to 1934 ... to 2067

kt_base_forecast_low_normalized <-
kt_base_forecast_low -  kt_base_forecast_low[as.character(2017)]

kt_base_forecast_up <- c(fitted_kt , kt_trajectories_up80)

names(kt_base_forecast_up) <- 1934:2067 ## to 1934 ... to 2067

kt_base_forecast_up_normalized <-
kt_base_forecast_up -  kt_base_forecast_up[as.character(2017)]
```

We then depict the transition of observed mortality in the jump-off year 2017 to the forecasted mortality in 2018:

```
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(0,110),ylim=c(log(0.00001,base=exp(1)),log(1,base=exp(1))),
xlab="Age",ylab="",main="Observed and LC forecasted \n US female mortality",
axes=FALSE, cex=0.6)

    for(year in 1933:2017){
        lines(x=0:110,y=log_usa_wom[,as.character(year)],
col=blackgray[(year-1932)])
    }

    lines(x=0:110,y=log_usa_wom[,as.character(2017)],col="blue",lwd=4)

    lines(x=0:110,
y=(log_usa_wom[,as.character(2017)]+bx*kt_base_forecast_normalized[as.character(2018)]),
col="purple",lwd=3)

    for(year in 2019:2067){
        lines(x=0:110,
y=(log_usa_wom[,as.character(2017)]+bx*kt_base_forecast_normalized[as.character(year)]),
col=blackred[(year-2017)],lwd=2)
    }

    legend(x=5,y=log(0.99,base=exp(1)),col=c(blackgray[1],blackgray[length(1933:2017)]),
lty=1,horiz=FALSE,bty="n",c("1933","2017"))
```

```
    legend(x=5,y=log(0.35,base=exp(1)),col=c("blue","purple"),lty=1,horiz=FALSE,
bty="n",c("2017 observed","2018 LC forecasted,\n based on observed mort 2017"),lwd=2)

    legend(x=5,y=log(0.05,base=exp(1)),col=c(blackred[2],blackred[50]),lty=1,horiz=FALSE,
bty="n",c("2019","2067"))

    axis(side=1,at=seq(0,110,5),labels=FALSE,lwd=1,pos=log(0.00001,base=exp(1)))

    axis(side=1,at=seq(0,110,10),labels=seq(0,110,10),lwd=3,pos=log(0.00001,base=exp(1)))

    axis(side=2,at=log(c(0.00001,0.0001,0.001,0.01,0.1,1),base=exp(1)),
labels=c(0.00001,0.0001,0.001,0.01,0.1,1),lwd=3,pos=0)
```



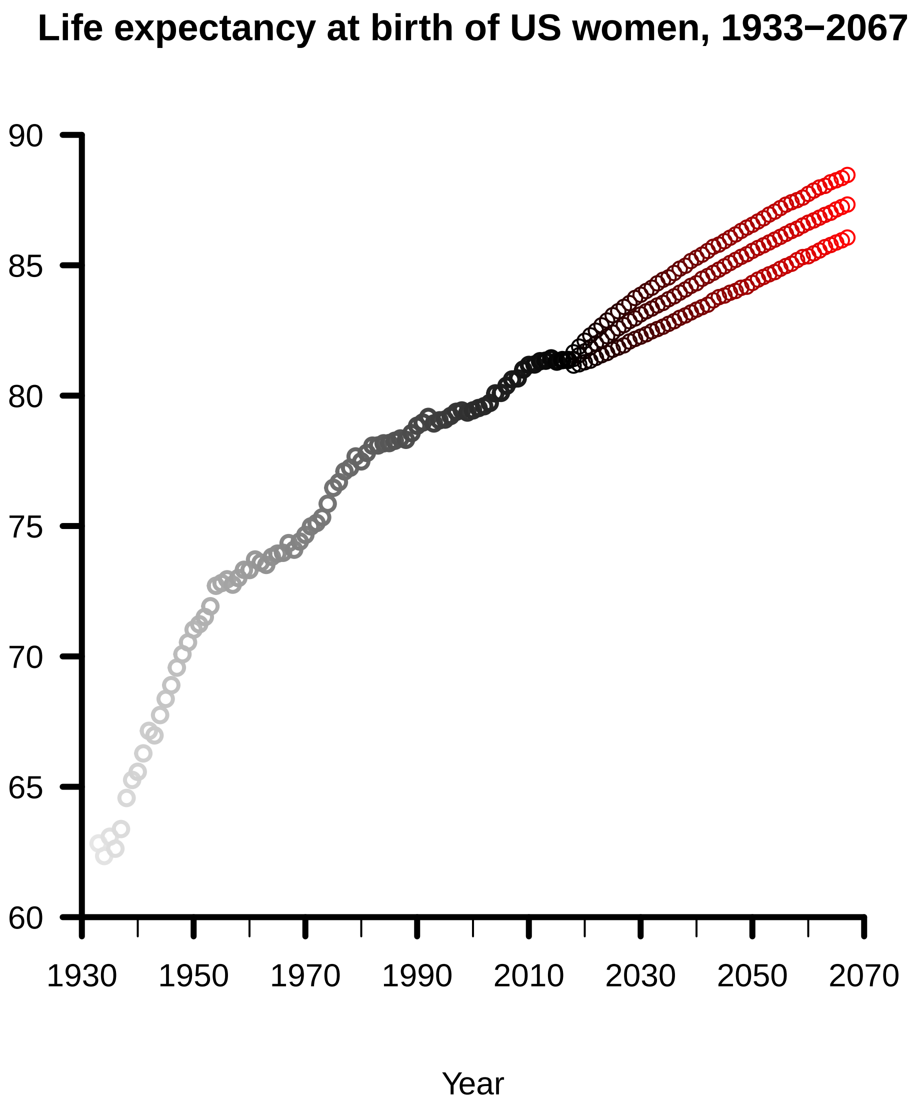
7. How does this look like in terms of life expectancy at birth?

We first write a 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)

}
```

We then depict the US female mortality forecasts generated with the Lee-Carter method in terms of life expectancy at birth:

```
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(1930,2020+50),ylim=c(60,90),xlab="Year",ylab="",
main="Life expectancy at birth of US women, 1933-2067",axes=FALSE)

    axis(side=1,at=seq(1930,2020+50,10),labels=FALSE,lwd=1,pos=60)

    axis(side=1,at=seq(1930,2020+50,20),labels=seq(1930,2020+50,20),lwd=3,pos=60)

    axis(side=2,at=seq(60,90,5),labels=seq(60,90,5),lwd=3,pos=1930)

    max_age <- 110

    lt_ax <- c(0.06,rep(0.5,max_age-1))

    for(year in 1933:2017){
        current.e0 <- lexp.age.specific(usa_wom[[2]][1:max_age,as.character(year)],
lt_ax)[1]

        points(year,current.e0,col=blackgray[(year-1932)],lwd=2)
    }

    for(year in 2018:2067){

        ## Median:

        current.mx <- exp(log(usa_wom[[2]][,as.character(2017)],base=exp(1))+
bx*kt_base_forecast_normalized[as.character(year)])

        current.e0 <- lexp.age.specific(current.mx[1:max_age],lt_ax)[1]

        points(year,current.e0,col=blackred[(year-2017)],lwd=1)

        ## Lower 80:

        current.mx <- exp(log(usa_wom[[2]][,as.character(2017)],base=exp(1))+
bx*kt_base_forecast_low_normalized[as.character(year)])

        current.e0 <- lexp.age.specific(current.mx[1:max_age],lt_ax)[1]

        points(year,current.e0,col=blackred[(year-2017)],lwd=1)

        ## Upper 80:

        current.mx <- exp(log(usa_wom[[2]][,as.character(2017)],base=exp(1))+
bx*kt_base_forecast_up_normalized[as.character(year)])

        current.e0 <- lexp.age.specific(current.mx[1:max_age],lt_ax)[1]

        points(year,current.e0,col=blackred[(year-2017)],lwd=1)
    }
```


Life expectancy at birth of US women, 1933−2067
90
85
80
75
70
65
60
1930
1950
1970
1990
2010
2030
2050
2070
Year