COS-D407. ***Scientific Modeling and Model Validation***

Hands-on exercises

Week 3

University of Helsinki, Finland

01.11.2021–15.12.2021

Lecturer: Christina Bohk-Ewald

Guest lecturer: Ricarda Duerst

Source: https://github.com/christina-bohk-ewald/2021-COS-D407-scientific-modeling-and-model-validation

Table of content:

1. Goals for this Lab Session

The main goal of this lab session is to understand the principal of ex-post validation and to apply it to mortality forecasts as an example of model validation. The goal is *not* to understand the demographic background analyses, e.g. how life tables or the Lee-Carter model work, *nor* to deeply understand every command of this R script that deals with data wrangling, visualization or demographic analyses. This is neither a class of demography, nor a class of R programming. Therefore, please don't worry if you don't understand every line of the code. *However*, do not hesitate to ask questions about any of this if you are interested! With today's exercises, we will:

- get to know the Human Mortality Database (HMD, http://www.mortality.org),
- familiarize with human mortality in Italy and Finland,
- apply the concept of ex-post validation to forecasts of life expectancy at birth using the Lee & Carter (1992) model,
- and get to know ggplot as an alternative way of data visualization in R.

2. Preparations in R

Open a new script for week 3 in R (e.g., *week-3.R*) and save it to a folder of your choice (e.g., *course-COS-D407*). Create a file path to this folder from where you would like to load data and to where you would like to save your outcome. For example,

```
the_course_COS_D407_path <- c("C:/course-COS-D407")
```

You can then set the working directory to this path:

```
setwd(the_course_COS_D407_path)
```

Before we start, we need to install and load several R packages for today's exercises. If you already installed the packages in advance, please *do not* run the lines `install packages(...` and `devtools....` You can turn the lines into comments using `#`. We load packages for data visualization (`ggplot`, `viridis`), downloading data from the HMD (`fda`, `HMDHFDplus`), mortality forecasting (`MortalityForecast`), and data manipulation (`dplyr`, `tidyr`). During the *installation* process, R may ask you if you want to update some of the packages. Just enter "3" in the console and hit enter to not update any of them. If R asks you another question, reply "Yes".

```
# install packages
install.packages(c("fda", "HMDHFDplus", "ggplot2", "viridis", "dplyr", "devtools",
"tidyr", "tibble", "MortalityLaws"), repos = "http://cran.us.r-project.org")

# load libraries
library(ggplot2)
library(HMDHFDplus)
library(fda)
library(dplyr)
library(viridis)
library(devtools)
library(tidyr)
library(tibble)
library(MortalityLaws)
devtools::install_github("mpascariu/MortalityForecast")
library(MortalityForecast)
```

3. Downloading HMD Data

Our data source is the Human Mortality Database (HMD, http://www.mortality.org) that provides high-quality data on population counts and mortality for more than 40 countries. To access the data you need to set up a free account. We can download data from the HMD using the R script. In order for the following code to work, you need to insert your e-mail address and password of your HMD account in quotation marks! We will use life table data for Finland and Italy. The HMD uses "FIN" and "ITA" as country codes. We will save the data sets for the female population of both countries in a list. If you are not familiar with lists, you can check this website for more information https://data-flair.training/blogs/r-list-tutorial/.

```
# insert your HMD user name and password here!
username <- "your@email.com"
password <- "yourPassword"
```

```
# define an object that contains the HMD country codes of the selected countries
HMD.countries <- c("FIN", "ITA")

# download life table data for Finland and Italy (female) and saving it in a list
lt.female <- list() # create an empty list

for (i in 1:length(HMD.countries)) { # number of selected countries

  lt.female[[i]] <- readHMDweb(CNTRY = HMD.countries[i], item = "fltper_1x1",
                               username, password, fixup=TRUE)
}

# naming the elements of the list with the country codes
names(lt.female) <- HMD.countries

# look at the life table data in the list
View(lt.female)

# look at life table data for Finland and Italy
head(lt.female[[1]])
head(lt.female[[2]])
```

4. Exploring Finnish and Italian Mortality

Now, let's see how the mortality in the two countries developed over time. Instead of the base R `plot` function, we will use `ggplot` for data visualization. For more information on `ggplot` and a nice cheat-sheet look here: https://ggplot2.tidyverse.org/. We will create 2 types of plots in this section:

- life expectancy at birth (e0) over time by country
- development of age-specific mortality rates (mx) for each country

4.1 Data Preparation

`ggplot` needs all the data used for plotting in one single data frame in the long data format (as opposed to the wide format). Because we want to plot the life expectancy at birth of both countries in one plot, we need to first add a country-identifying variable and then combine the two data sets.

```
# add country variable
lt.female[[1]]$cntr <- HMD.countries[1]
lt.female[[2]]$cntr <- HMD.countries[2]

# combine data frames of Italy and Finland
lt.female.long <- bind_rows(lt.female)
```

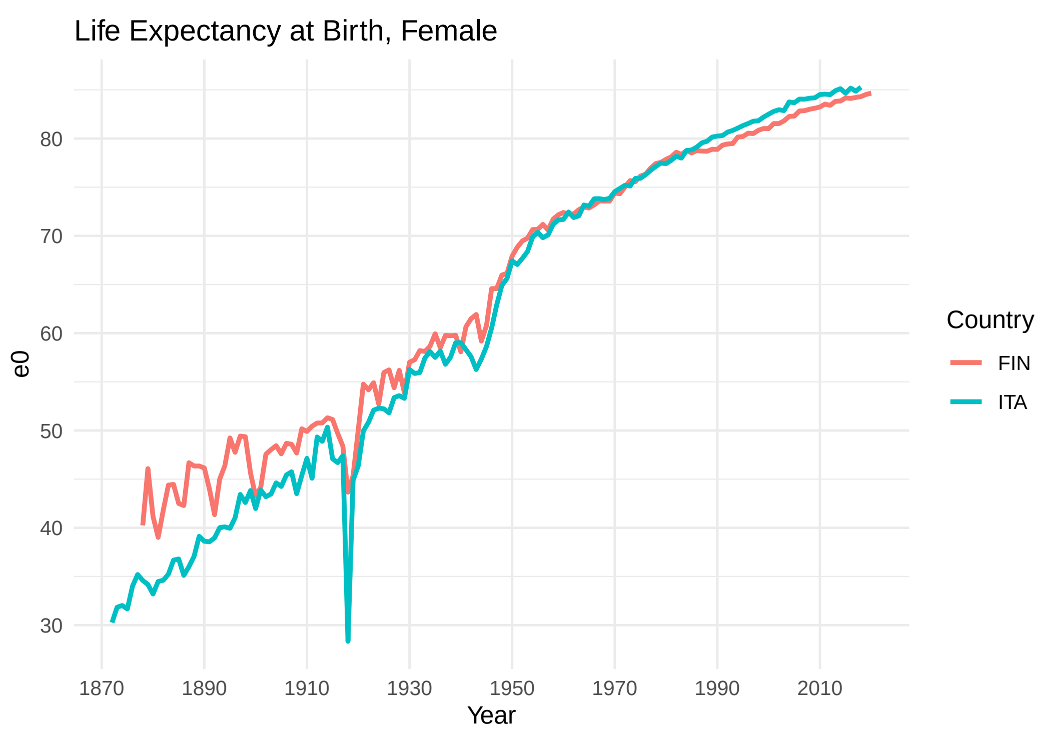
```
# sort data frame by country, year, age
lt.female.long <- arrange(lt.female.long, cntr, Year, Age)

# look at the transformed data
head(lt.female.long)
```

```
##   Year Age      mx      qx   ax     lx    dx    Lx      Tx    ex OpenInterval cntr
## 1 1878   0 0.18302 0.16261 0.31 100000 16261 88847 4025475 40.25        FALSE  FIN
## 2 1878   1 0.08360 0.08025 0.50  83739  6720 80379 3936628 47.01        FALSE  FIN
## 3 1878   2 0.04683 0.04576 0.50  77019  3524 75257 3856248 50.07        FALSE  FIN
## 4 1878   3 0.02822 0.02783 0.50  73495  2045 72473 3780991 51.45        FALSE  FIN
## 5 1878   4 0.02139 0.02116 0.50  71450  1512 70694 3708518 51.90        FALSE  FIN
## 6 1878   5 0.01569 0.01557 0.50  69938  1089 69394 3637824 52.02        FALSE  FIN
```

4.2 Life Expectancy at Birth

Do you find anything remarkable about the development of life expectancy at birth in Finland and Sweden? If so, what could be the reason for that? How does the development of e0 compare between the two countries?

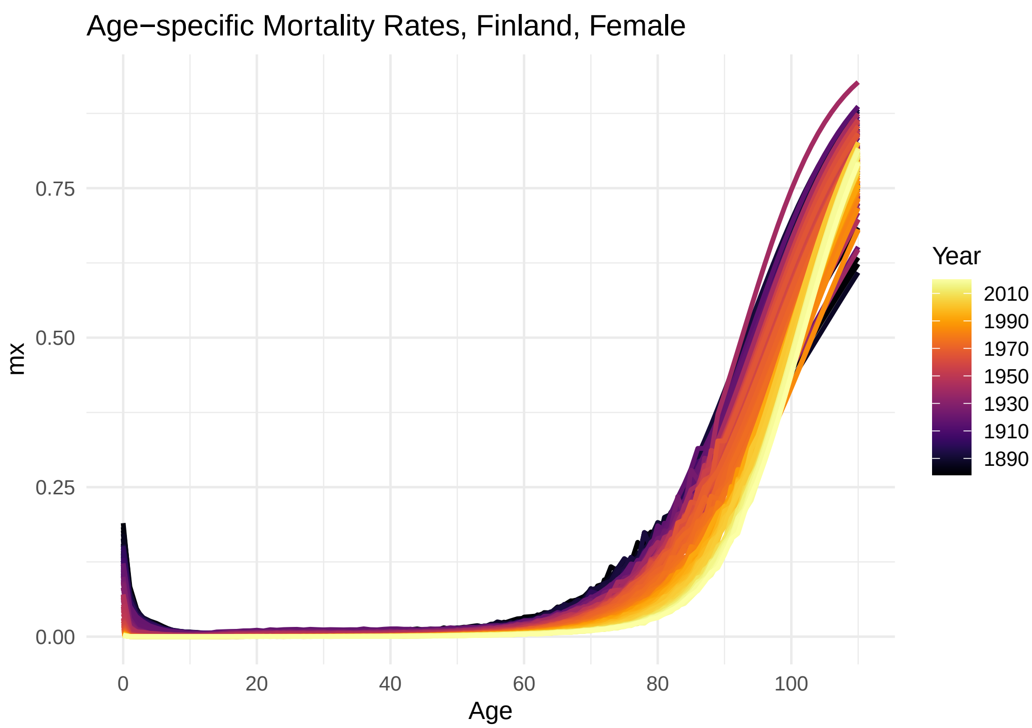
```
ggplot(data = subset(lt.female.long, Age == 0), mapping = aes(x = Year, y = ex, col = cntr)) +
  geom_line(size = 1) +
  theme_minimal() +
  ggtitle("Life Expectancy at Birth, Female") +
  xlab("Year") +
  ylab("e0") +
  labs(col = "Country") +
  scale_x_continuous(breaks = seq(1850, 2020, by = 20), minor_breaks = NULL)
```



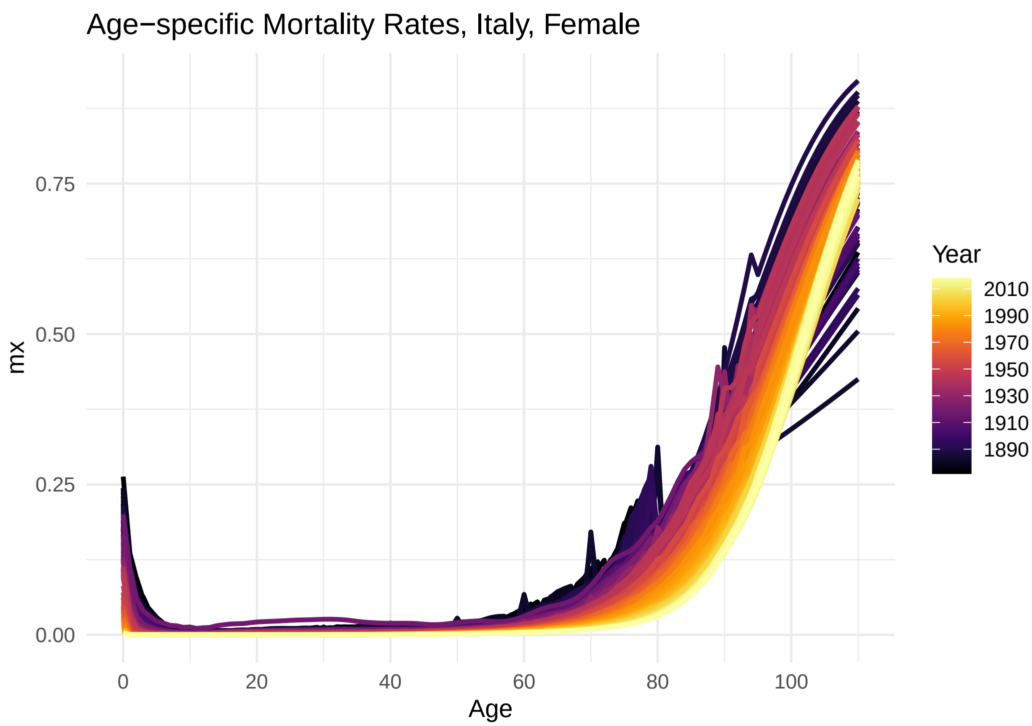
4.3 Age-specific Mortality Rates

Which developments in the age-specific mortality rates contributed to the increase of life expectancy? Are there differences between Italy and Finland?

```
ggplot(data = subset(lt.female.long, cntr == HMD.countries[1]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality Rates, Finland, Female") +
  xlab("Age") +
  ylab("mx") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```

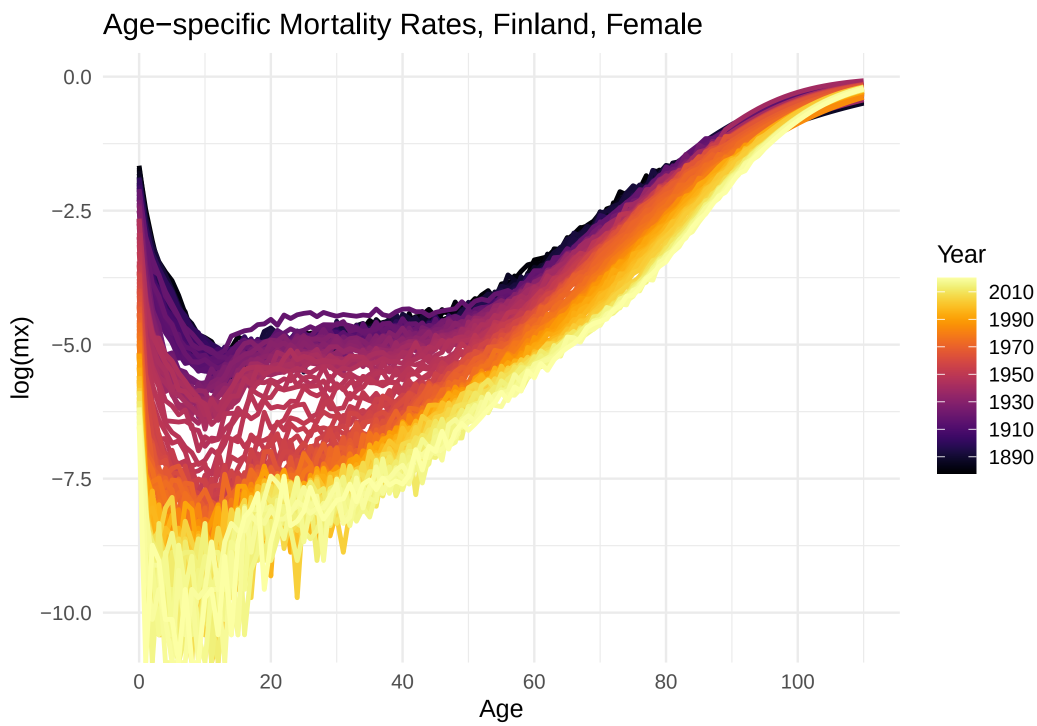



```
ggplot(data = subset(lt.female.long, cntr == HMD.countries[2]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality Rates, Italy, Female") +
  xlab("Age") +
  ylab("mx") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```

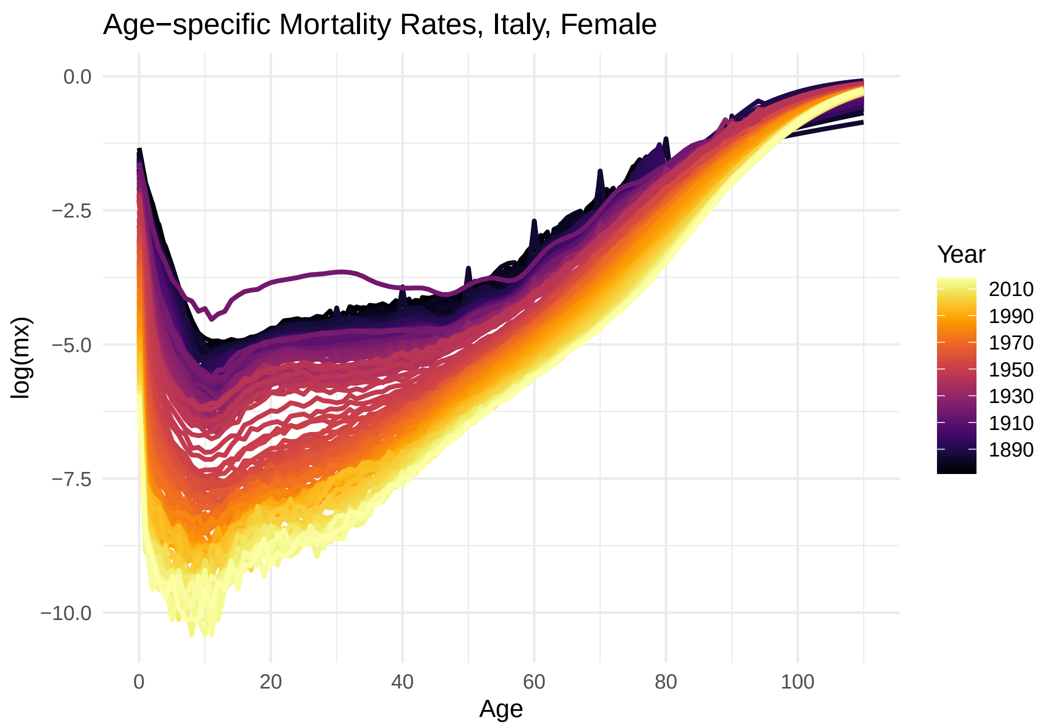


To better see the mortality development at low levels of mx, we can plot the same data on a log-scale. Do you have new insights on the questions above?

```
ggplot(data = subset(lt.female.long, cntr == HMD.countries[1]),
       mapping = aes(x = Age, y = log(mx), group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality Rates, Finland, Female") +
  xlab("Age") +
  ylab("log(mx)") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```




```r
ggplot(data = subset(lt.female.long, cntr == HMD.countries[2]),
       mapping = aes(x = Age, y = log(mx), group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality Rates, Italy, Female") +
  xlab("Age") +
  ylab("log(mx)") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```



5. Validating Lee-Carter Mortality Forecasts

Now, we will do some model validation. Specifically, we will perform an ex-post validation (also called out-of-sample validation) of Lee-Carter mortality forecasts. Ex-post validation means checking the performance in hindsight. However, because nobody knows how the reality will actually look like in the future (and we can't wait so long), we have to use what we have given: The past mortality development. Therefore, we withhold some of the observed mortality to be able to compare it to forecasts of the same period. The goal is to answer the question how well the Lee-Carter model from 1992 is able to forecast the mortality of Finland and Italy.

First, we will apply the Lee and Carter model from 1992 to our data to forecast life expectancy at birth. To do so, we will use the package `MortalityForecast` by Marius Pascariu that offers a variety of different forecast models. You can find more information on the package in his GitHub repository: https://github.com/mpascariu/MortalityForecast.

5.1 Data Preparation

We do not need the full life table to fit a Lee-Carter model and forecast life expectancy at birth. We will first extract the age-specific mortality rates mx, ages and years, and then transform the data into the format that works best with the functions of the `MortalityForecast` package. We will do all analyses for both Finland and Italy. Remember that the data for the two countries is stored in a list? Whenever you see the function `lapply`, the functions inside `lapply` are applied to all the elements of a list, in our case, to Italy and Finland.

```
# extract mx, ages and years from life tables
mx.female <- lapply(lt.female, select, mx, Age, Year)

# bring mx in right data format for Lee-Carter function (ages in rows, years in columns)
mx.female <- lapply(mx.female, spread, key=Year, value=mx)
mx.female <- lapply(mx.female, column_to_rownames, var="Age")

# death rates equal to zero have to be replaced (by minimum observed death rate)
mx.female <- lapply(mx.female, replace.zeros, method = "min")
```

```r
# look at Finnish and Italian mx data
head(mx.female[[1]], n = c(6,9))
```

```
##      1878    1879    1880    1881    1882    1883    1884    1885    1886
## 0 0.18302 0.14308 0.16432 0.18986 0.17741 0.15242 0.15560 0.17160 0.16255
## 1 0.08360 0.05056 0.06649 0.07696 0.07177 0.05067 0.05545 0.06962 0.06744
## 2 0.04683 0.03392 0.03903 0.04395 0.04132 0.03453 0.03333 0.04048 0.04043
## 3 0.02822 0.02326 0.02908 0.03167 0.02918 0.02688 0.02547 0.02820 0.03023
## 4 0.02139 0.01745 0.02101 0.02615 0.02299 0.02136 0.01945 0.01912 0.02127
## 5 0.01569 0.01106 0.01726 0.02223 0.01721 0.01602 0.01539 0.01439 0.01541
```

```r
head(mx.female[[2]], n = c(6,9))
```

```
##      1872    1873    1874    1875    1876    1877    1878    1879    1880
## 0 0.26296 0.23050 0.23941 0.24303 0.22374 0.21746 0.21812 0.22941 0.23680
## 1 0.13614 0.10500 0.09773 0.10088 0.09276 0.08669 0.09213 0.09337 0.09793
## 2 0.09307 0.09418 0.07664 0.07184 0.06954 0.06723 0.06693 0.06577 0.07105
## 3 0.05400 0.05978 0.06305 0.05259 0.04684 0.04729 0.04814 0.04451 0.04690
## 4 0.03515 0.03387 0.03879 0.04189 0.03354 0.03096 0.03273 0.03107 0.03068
## 5 0.02422 0.02355 0.02390 0.02914 0.02833 0.02288 0.02278 0.02314 0.02257
```

5.2 Fitting and Forecasting with the Lee-Carter Model

Now, it is time to fit the Lee-Carter model to our data. Afterwards, we can forecast the age-specific mortality rates. To perform an ex-post forecast validation later, we will not use every available year as input for the forecast, but cut off our data after the year 1980. 1980 is our jump-off year (JOY). The last common year of data for our two countries is 2018. Therefore, we are able to validate forecasts from 1981 to 2018, which sum to 38 years. 38 years is the length of our forecast horizon. Further, we restrict the input data on which the forecast is based on: our base period begins in 1950 and ends in 1980.

```r
# defining base period and forecast horizon
bp <- 1950:1980 # base period of input data
fh.start <- 1981 # first year for forecast
fh.end <- 2018 # last year for forecast
fh <- length(fh.start:fh.end) # number of forecast years

# excluding data before 1950 and after 1980
mx.female.bp <- lapply(mx.female,
                       FUN = function(x){x[, which(as.numeric(colnames(x)) %in% bp)]})

# fitting the Lee-Carter model
age <- 0:110 # age vector

lc.female <- lapply(mx.female.bp, model.LeeCarter, x=age,
                    y=as.numeric(colnames(mx.female.bp[[i]])))

# forecasting with the Lee-Carter model
p.lc.female <- lapply(lc.female, predict, h = fh, level = c(80, 95))

# having a look at the resulting forecast objects and forecast values of Finnish mx
p.lc.female
```

```
## $FIN
##
## Forecast: Lee-Carter Mortality Model
```

```
## Model   : log m[x,t] = a[x] + b[x]k[t]
## Call    : predict.LeeCarter(object = X[[i]], h = ..1, level = ..2)
## Ages  in forecast: 0 - 110
## Years in forecast: 1981 - 2018
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
##
## $ITA
##
## Forecast: Lee-Carter Mortality Model
## Model   : log m[x,t] = a[x] + b[x]k[t]
## Call    : predict.LeeCarter(object = X[[i]], h = ..1, level = ..2)
## Ages  in forecast: 0 - 110
## Years in forecast: 1981 - 2018
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
```

```
head(p.lc.female[[1]]$predicted.values, n = c(6,5))
```

```
##           1981         1982         1983         1984         1985
## 0 0.0063636265 0.0059640269 0.0055895199 0.0052385298 0.0049095799
## 1 0.0004137250 0.0003803741 0.0003497117 0.0003215210 0.0002956028
## 2 0.0002133379 0.0001978830 0.0001835476 0.0001702507 0.0001579171
## 3 0.0002076699 0.0001960309 0.0001850442 0.0001746733 0.0001648836
## 4 0.0002363418 0.0002234298 0.0002112232 0.0001996835 0.0001887742
## 5 0.0001520327 0.0001444621 0.0001372685 0.0001304332 0.0001239381
```

5.3 Calculating and Plotting Forecast Life Expectancy

Now that we have forecast values of age-specific mortality rates, we will calculate new life tables with the forecast values. We will use the resulting life expectancy at birth to assess the forecast performance of the Lee-Carter model. First, we have to extract the forecast mx values from the object that resulted from applying the `predict` function. We will use the function `LifeTable` to calculate the life expectancy. After some data wrangling, we are able to plot the observed life expectancy together with our Lee-Carter forecasts.

```
# extracting forecast mx values from Lee-Carter object
p.mx.female <- lapply(p.lc.female,
                      FUN = function(x){cbind(x$x, as.matrix(x$predicted.values))})

# calculate life tables
age <- 0:110 # age vector

p.lt.female <- list() # empty list for results
p.lt.female.y <- list() # empty temporary list for yearly life tables

for (i in 1:length(p.mx.female)) {           # countries
    for (j in 2:(fh+1)) {                        # forecast years 1 to 30
      p.lt.female.y[[j-1]] <- LifeTable(x = age, mx = p.mx.female[[i]][,j])
    }
    names(p.lt.female.y) <- colnames(p.mx.female[[i]][,-1])
  p.lt.female[[i]] <- p.lt.female.y
}

names(p.lt.female) <- HMD.countries

# extracting forecast e0 from life tables
age.ex <- 0 # age for life expectancy, in our case at birth
```

```r
p.e0.female <- matrix(data = NA, nrow = length(HMD.countries), ncol = fh)

for (i in 1:length(HMD.countries)) {
  for (j in 1:fh) {
    p.e0.female[i, j] <- p.lt.female[[i]][[j]]$lt$ex[p.lt.female[[i]][[j]]$lt$x==age.ex]
  }
}

# some data transformation
p.e0.female <- as.data.frame(p.e0.female)
colnames(p.e0.female) <- fh.start:fh.end
p.e0.female$cntr <- HMD.countries
p.e0.female$Age <- age.ex
lt.female.long.ex <- lt.female.long[which(lt.female.long$Age == age.ex),]

#creating one (long format) data frame with observed and forecast e0 for ggplot
p.e0.female.long <- gather(data = p.e0.female, key = "Year", value = "p.ex", 1:fh)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(fh)` instead of `fh` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```r
p.e0.female.long$Year <- as.numeric(p.e0.female.long$Year)
e0.female.long <- full_join(lt.female.long.ex[,c("cntr", "Year", "Age", "ex")],
                            p.e0.female.long)
```

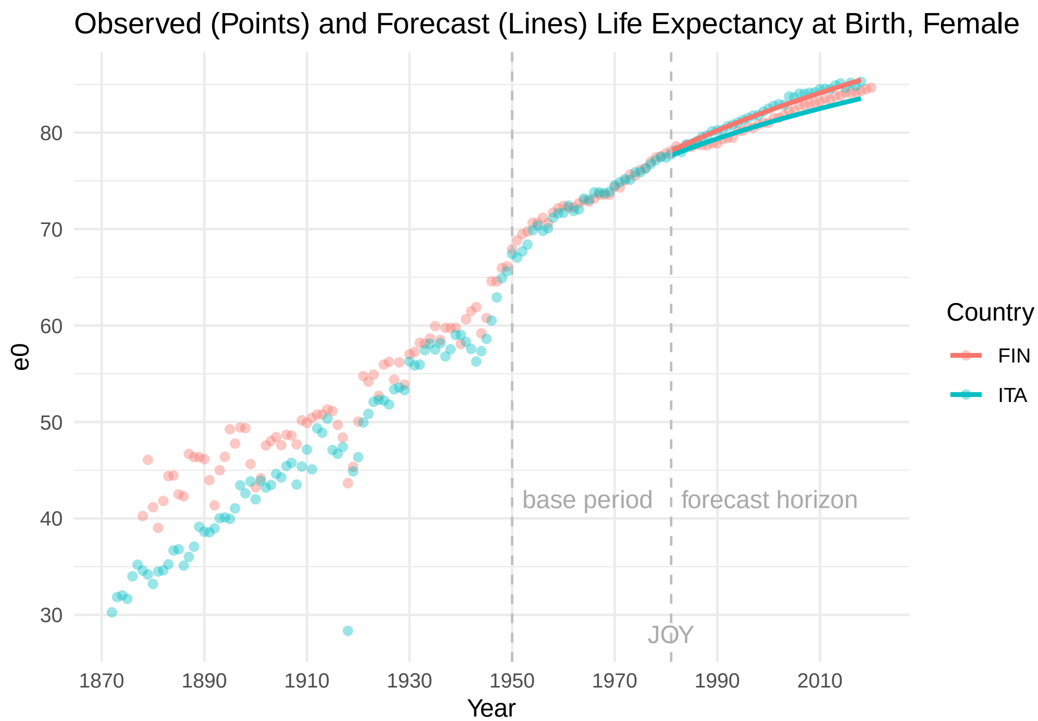
```
## Joining, by = c("cntr", "Year", "Age")
```

```r
# look at transformed data
tail(e0.female.long)
```

```
##     cntr Year Age    ex     p.ex
## 285  ITA 2013   0 84.91 82.90304
## 286  ITA 2014   0 85.12 83.03405
## 287  ITA 2015   0 84.66 83.16361
## 288  ITA 2016   0 85.18 83.29177
## 289  ITA 2017   0 84.86 83.41854
## 290  ITA 2018   0 85.28 83.54396
```

```r
# plot observed and forecast e0 together
ggplot(data = e0.female.long, mapping = aes(x = Year, y = ex, col = cntr)) +
  geom_point(aes(x = Year, y = ex), alpha = 0.4) +
  geom_line(aes(x = Year, y = p.ex), size = 1) +
  geom_vline(xintercept = fh.start, linetype = "dashed", colour = "grey") +
  geom_vline(xintercept = bp[1], linetype = "dashed", colour = "grey") +
  annotate(geom = "text", x = bp[1]+2, y = 42, label = "base period", hjust = "left",
           size = 4, color = "Darkgrey") +
  annotate(geom = "text", x = fh.start, y = 28, label = "JOY", hjust = "center",
           size = 4, color = "Darkgrey") +
  annotate(geom = "text", x = fh.start+2, y = 42, label = "forecast horizon", hjust = "left",
           size = 4, color = "Darkgrey") +
  theme_minimal() +
  ggtitle("Observed (Points) and Forecast (Lines) Life Expectancy at Birth, Female") +
  xlab("Year") +
```

```
  ylab("e0") +
  labs(col = "Country") +
  scale_x_continuous(breaks = seq(1850, fh.end, by = 20), minor_breaks = NULL)
```



How would you interpret this plot? How does the Lee-Carter model perform? Are there differences for the performance between countries? Why is it a good choice to start the base period in 1950?

5.4 Forecast Error Measures

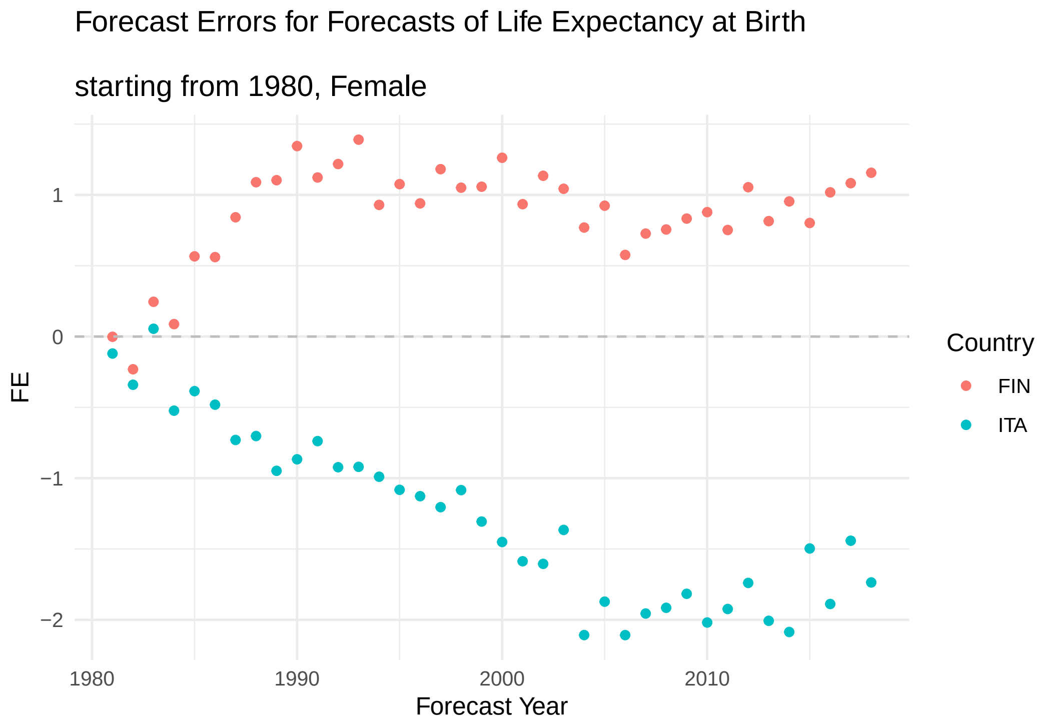
Now that we have our forecast values of life expectancy at birth, we can assess the forecast performance of the Lee-Carter model. There is a variety of forecast error measures to do that. For a review of those, see Shcherbakov, et al. (2013): A Survey of Forecast Error Measures, or Hyndman & Koehler (2006): Another look at measures of forecast accuracy. We start by calculating the simplest of forecast error measures: the forecast error (FE), which is just the forecast value minus the observed value for each year. From there, we can calculate summary measures that assess the forecast performance over the full forecast horizon (instead of yearly), e.g. the mean error (ME). Other error measures are the absolute percentage error (APE) and its summary measure median absolute percentage error (MdAPE), for example.

```
# calculate forecast error measures
e0.female.long$FE <- e0.female.long$p.ex - e0.female.long$ex
e0.female.long$APE <- (abs(e0.female.long$FE)/e0.female.long$ex)*100

# look at the data frame
tail(e0.female.long)
```

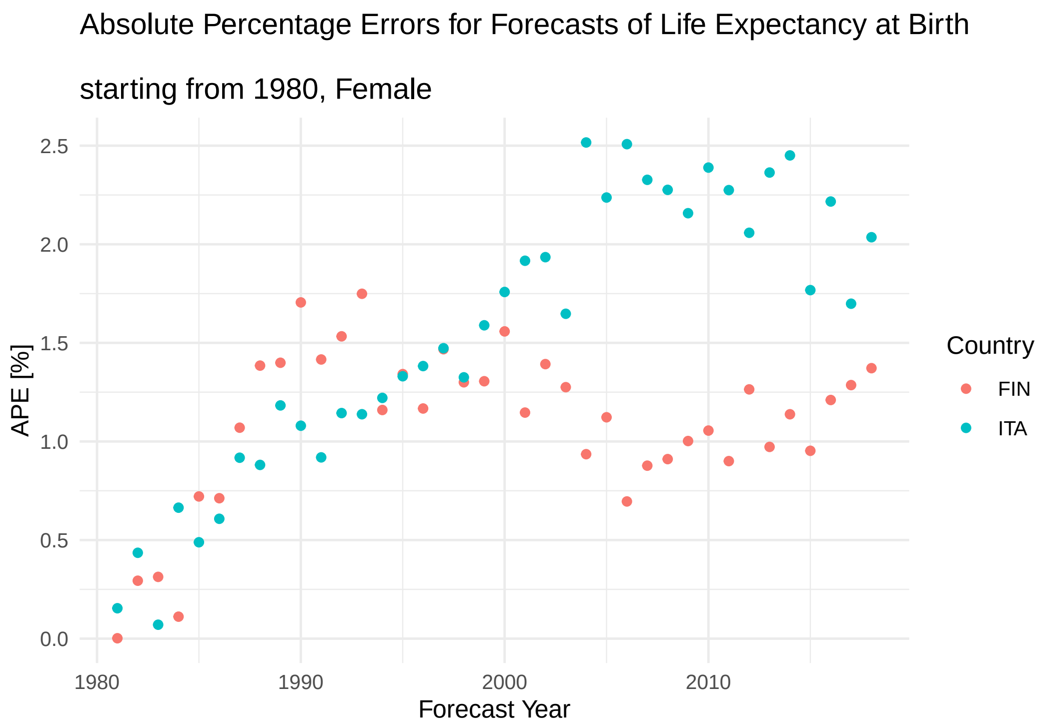
```
##     cntr Year Age    ex     p.ex        FE      APE
## 285  ITA 2013   0 84.91 82.90304 -2.006964 2.363637
## 286  ITA 2014   0 85.12 83.03405 -2.085953 2.450602
## 287  ITA 2015   0 84.66 83.16361 -1.496388 1.767526
## 288  ITA 2016   0 85.18 83.29177 -1.888235 2.216758
## 289  ITA 2017   0 84.86 83.41854 -1.441462 1.698635
```

```
## 290  ITA 2018   0 85.28 83.54396 -1.736037 2.035690
```

```r
# plots of AE and PE
ggplot(data = e0.female.long, mapping = aes(x = Year, y = FE, col = cntr)) +
  geom_point(data = subset(e0.female.long, Year %in% fh.start:fh.end)) +
  geom_hline(yintercept = 0, linetype = "dashed", colour = "grey") +
  theme_minimal() +
  ggtitle("Forecast Errors for Forecasts of Life Expectancy at Birth
          \nstarting from 1980, Female") +
  xlab("Forecast Year") +
  ylab("FE") +
  labs(col = "Country")
```




```r
ggplot(data = e0.female.long, mapping = aes(x = Year, y = APE, col = cntr)) +
  geom_point(data = subset(e0.female.long, Year %in% fh.start:fh.end)) +
  theme_minimal() +
  ggtitle("Absolute Percentage Errors for Forecasts of Life Expectancy at Birth
          \nstarting from 1980, Female") +
  xlab("Forecast Year") +
  ylab("APE [%]") +
  labs(col = "Country")
```




```
# calculate some summary error measures
error.female <- matrix(NA, ncol = length(HMD.countries), nrow = 2)
colnames(error.female) <- HMD.countries
rownames(error.female) <- c("ME", "MdAPE")

error.female[1,] <- cbind(mean(filter(e0.female.long, cntr == HMD.countries[1])$FE, na.rm = TRUE),
                   mean(filter(e0.female.long, cntr == HMD.countries[2])$FE, na.rm = TRUE))

error.female[2,] <- cbind(median(filter(e0.female.long, cntr == HMD.countries[1])$APE, na.rm = TRUE),
                    median(filter(e0.female.long, cntr == HMD.countries[2])$APE, na.rm = TRUE))

error.female
```

```
##             FIN       ITA
## ME    0.8697401 -1.277231
## MdAPE 1.1530154  1.618312
```

How would you interpret the results of the forecast validation? Can you think of advantages and disadvantages of different kinds of error measures (hint: have a look at the papers discussing different error measures)?

6. Additional Questions

If you liked the exercises and are interested to find out more about mortality forecasts and their validation, you could try to do one, or two, or all of the following exercises! Don't hesitate to ask for help if you get stuck or don't know how to tackle an exercise!

- Do the same validation analysis for men! Hint: You have to change the name of the HMD file, and "Ctrl + F" helps you to replace "female" by "male" in your code after copying it. Are there differences in the male mortality compared to the females? Does it have an effect on the forecasts or their validation?
- Forecast the Italian and Finnish life expectancy into the future! Hint: You have to change the variables `bp`, `fh.start` and `fh.end`. Has the length of the forecast horizon an effect on the forecast results?

- Calculate additional forecast error measures, e.g. MAPE or RMSE! Hint: See the mentioned articles for formulas and explanations. Does it change the interpretation of the forecast performance?
- Play around with the length of the base period or the forecast horizon! Hint: You could change e.g. the first number of the variable `bp` to have a longer base period, or change `bp`, `fh.start` and `fh.end` to move around the full window of analysis. Has the base period an effect on the forecast and error measures?
- Add another country to the analysis! Hint: You have to add the country code from the HMD to the variable `HMD.countries`.