COS-D407. *Scientific Modeling and Model Validation*

Example Solutions for Additional 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:

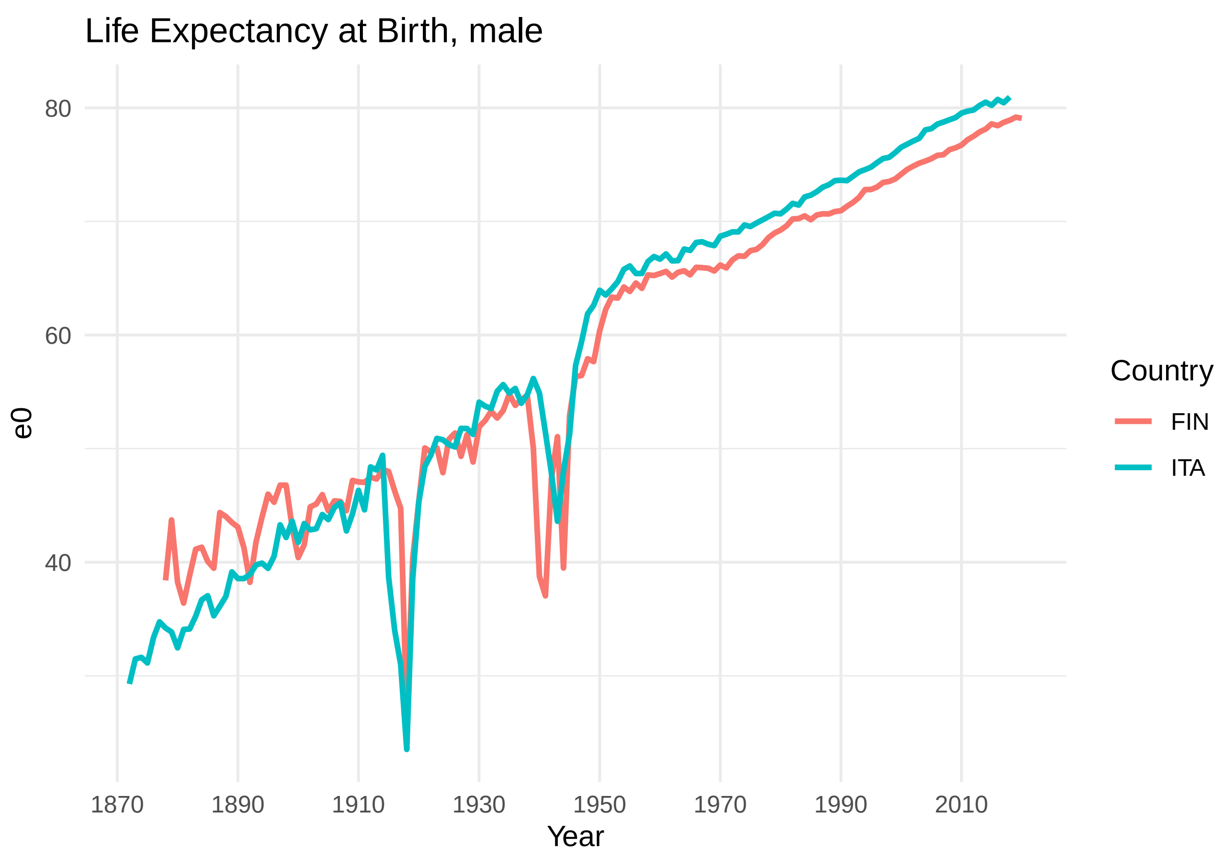
1. Aim of this Document

With this document, I would like to provide you with possible ways of approaching the additional exercises of week 3's hands-on exercises. It does not include interpretations of results, nor a comprehensive analyses regarding the questions. For each additional exercise, you will find one of many possible solutions of how to tackle it in R. If you are interested in the topic of mortality forecast validation, go ahead and do the additional exercises! You can then compare your code and your results to my code, or check this document if you don't know how to handle a question in R. This code does not stand for it's own! In order for it to work, you have to run the code of hands-on exercise week 3. I recommend to first run the code of hands-on exercises week 3 again, before running the code of a new additional exercise from this document (excluding section 2 "Preparations in R", this only needs to be run once).

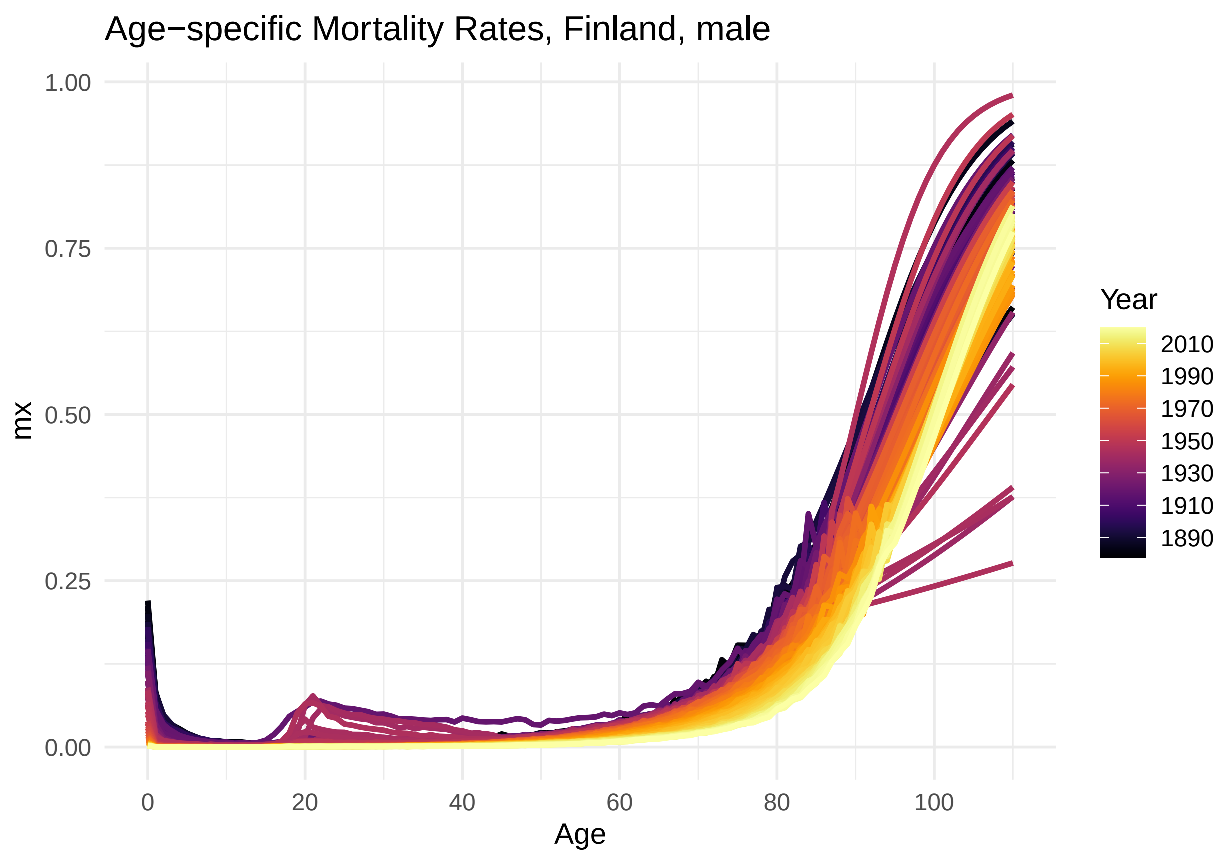
2. Validation Analysis for Finnish and Italian Men

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?

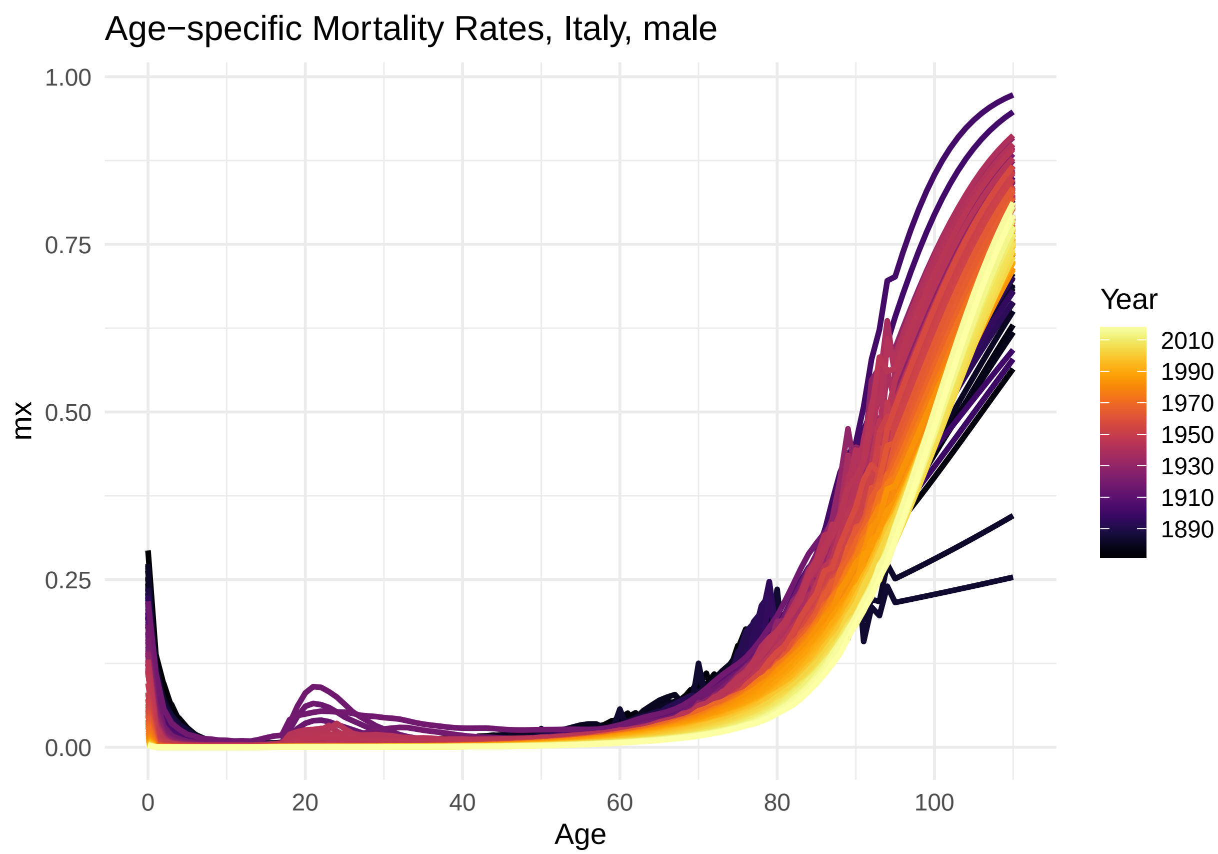
```
# download life table data for Finland and Italy (male) and saving it in a list ####
lt.male <- list() # create an empty list

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

  lt.male[[i]] <- readHMDweb(CNTRY = HMD.countries[i], item = "mltper_1x1",
                              username, password, fixup=TRUE)

}

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


### 4. ----

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

# combine data frames of Italy and Finland
lt.male.long <- rbind(lt.male[[1]], lt.male[[2]])

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


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



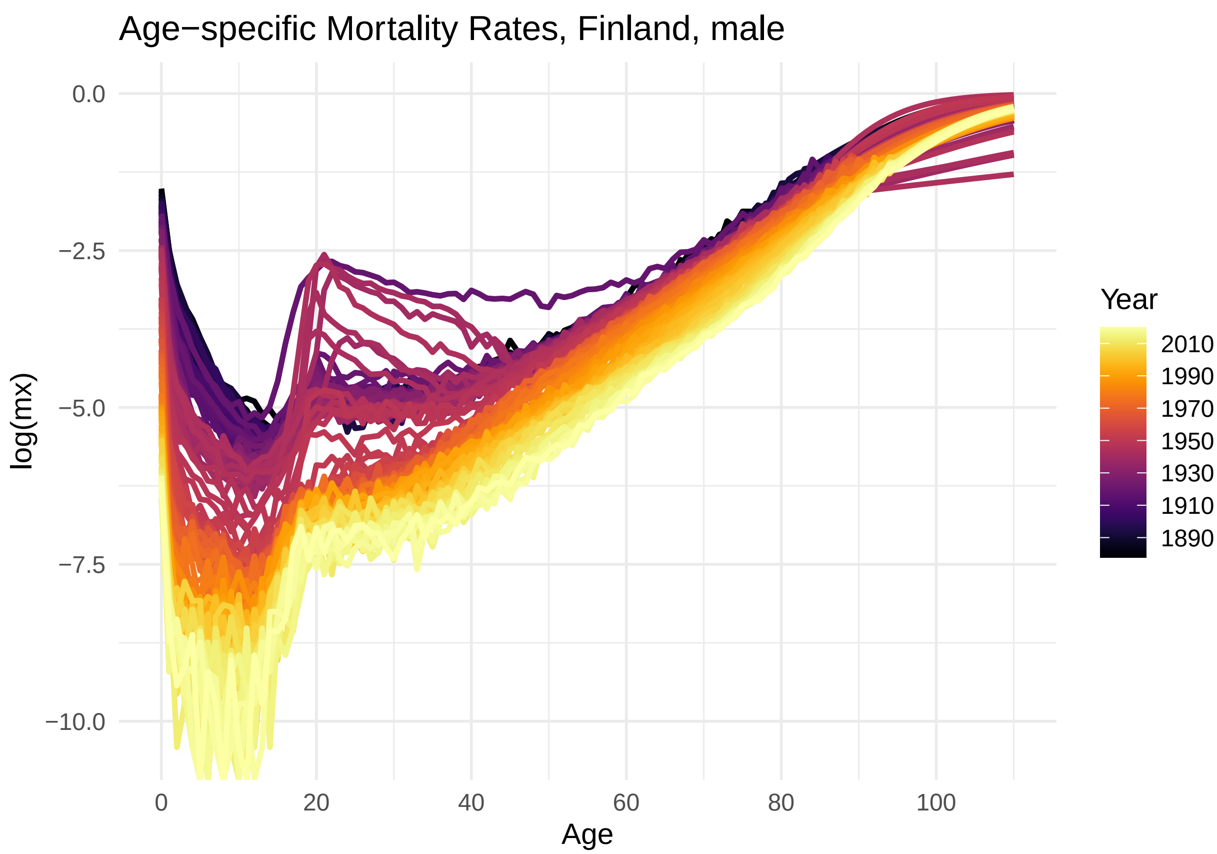

```r
ggplot(data = subset(lt.male.long, cntr == HMD.countries[1]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality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



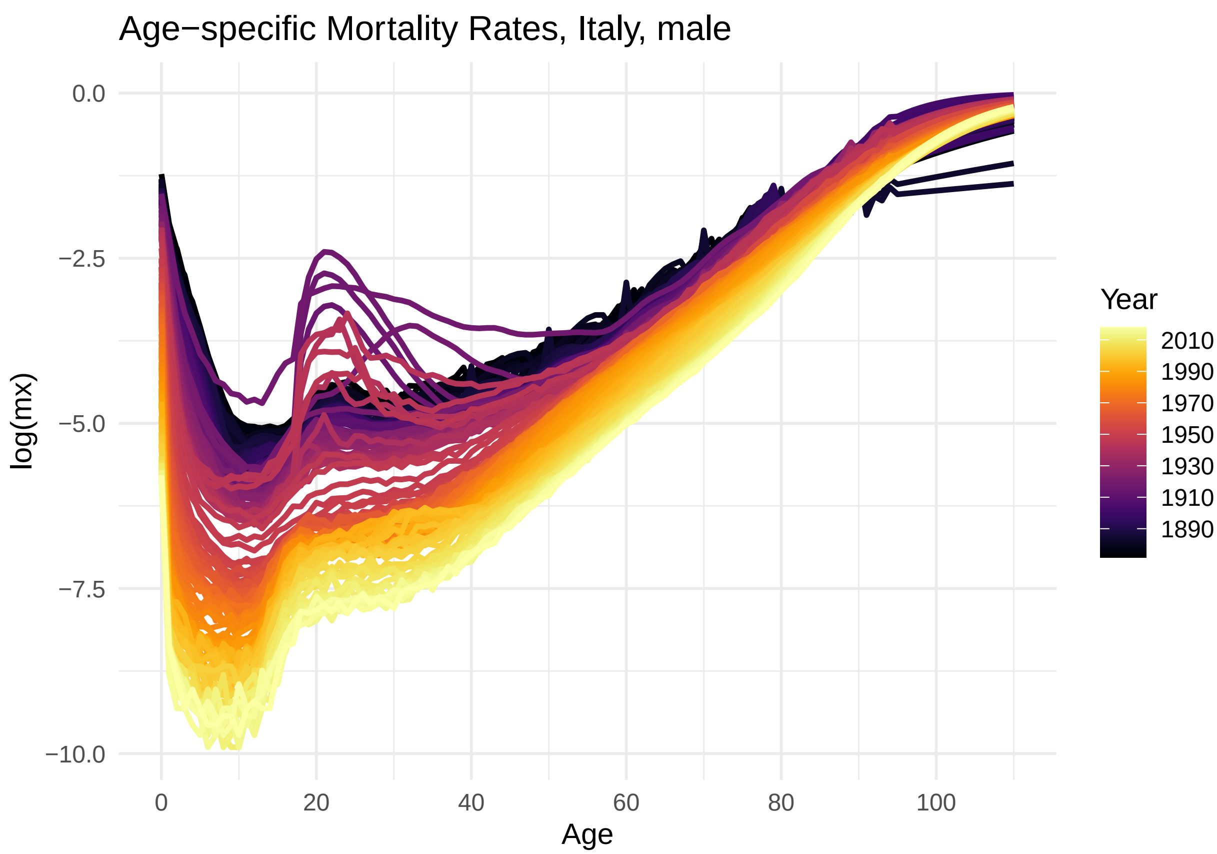

```
ggplot(data = subset(lt.male.long, cntr == HMD.countries[2]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality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




```
ggplot(data = subset(lt.male.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, male") +
  xlab("Age") +
  ylab("log(mx)") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```




```r
ggplot(data = subset(lt.male.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, male") +
  xlab("Age") +
  ylab("log(mx)") +
  scale_x_continuous(breaks = c(seq(0, 110, by = 20)))
```



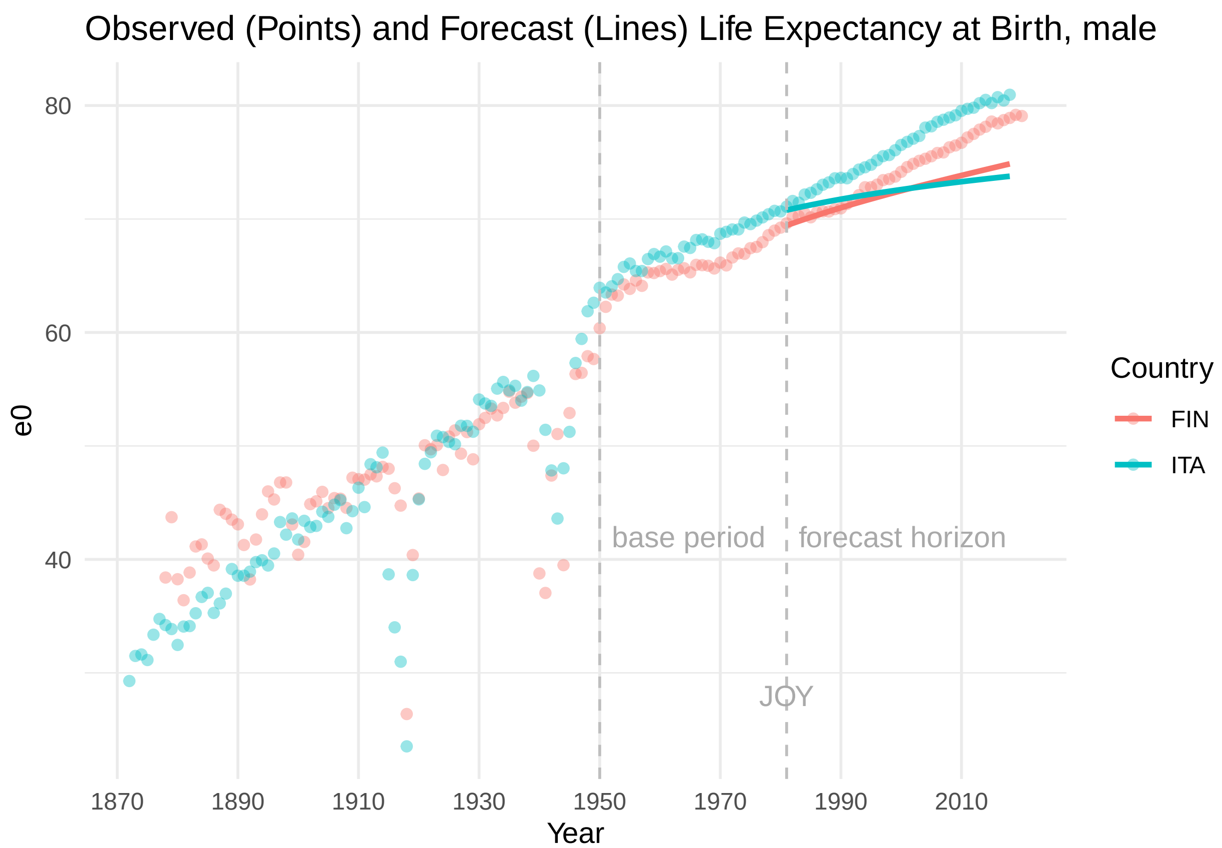

```
### 5. ----

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

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

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


# 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.male.bp <- lapply(mx.male,
                     FUN = function(x){x[, which(as.numeric(colnames(x)) %in% bp)]})

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

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

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

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

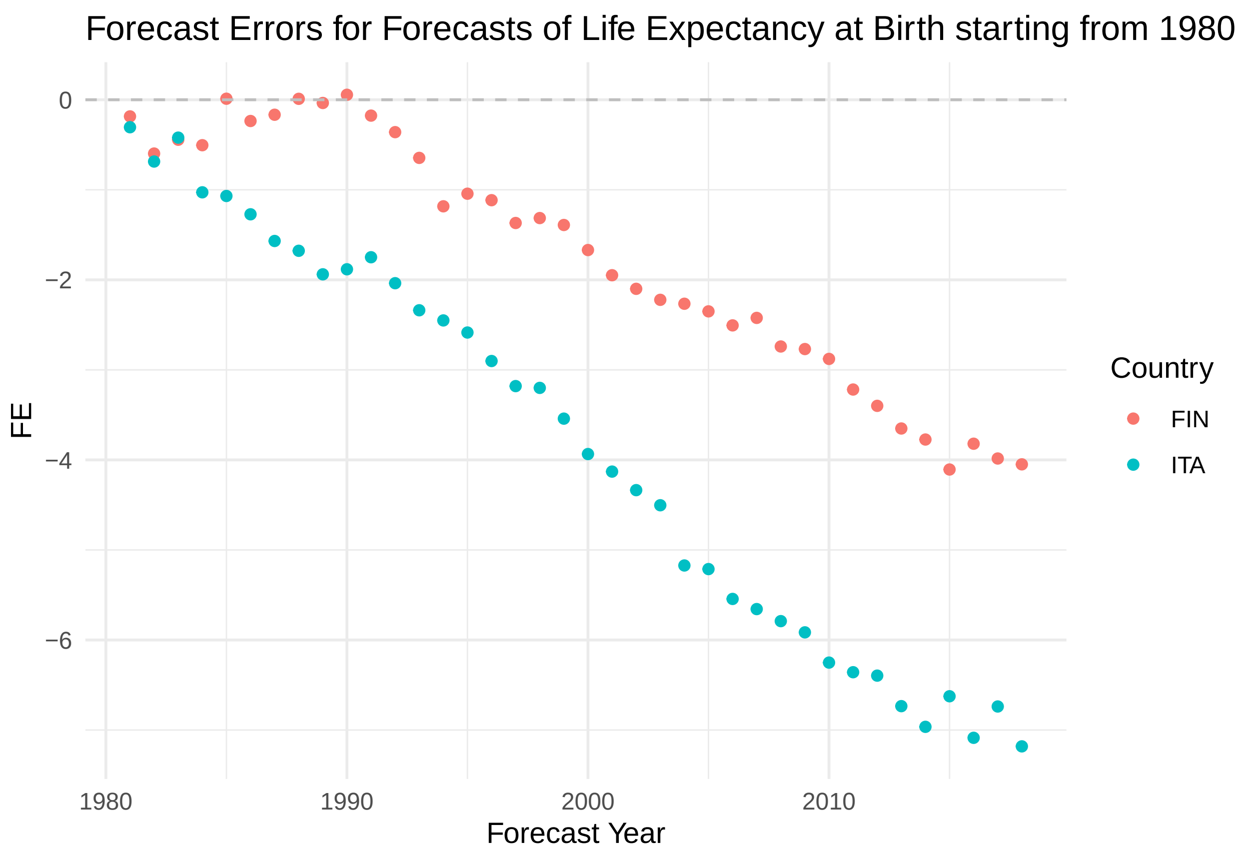
```
## $FIN
##
## Forecast: Lee-Carter Mortality Model
## Model   : log m[x,t] = a[x] + b[x]k[t]
## Call    : predict.LeeCarter(object = X[[i]], h = ..1, level = ..2)
## Ages  in forecast: 0 - 110
## Years in forecast: 1981 - 2018
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
##
## $ITA
##
## Forecast: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

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

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

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

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

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

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

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

for (i in 1:length(HMD.countries)) {
  for (j in 1:fh) {
```

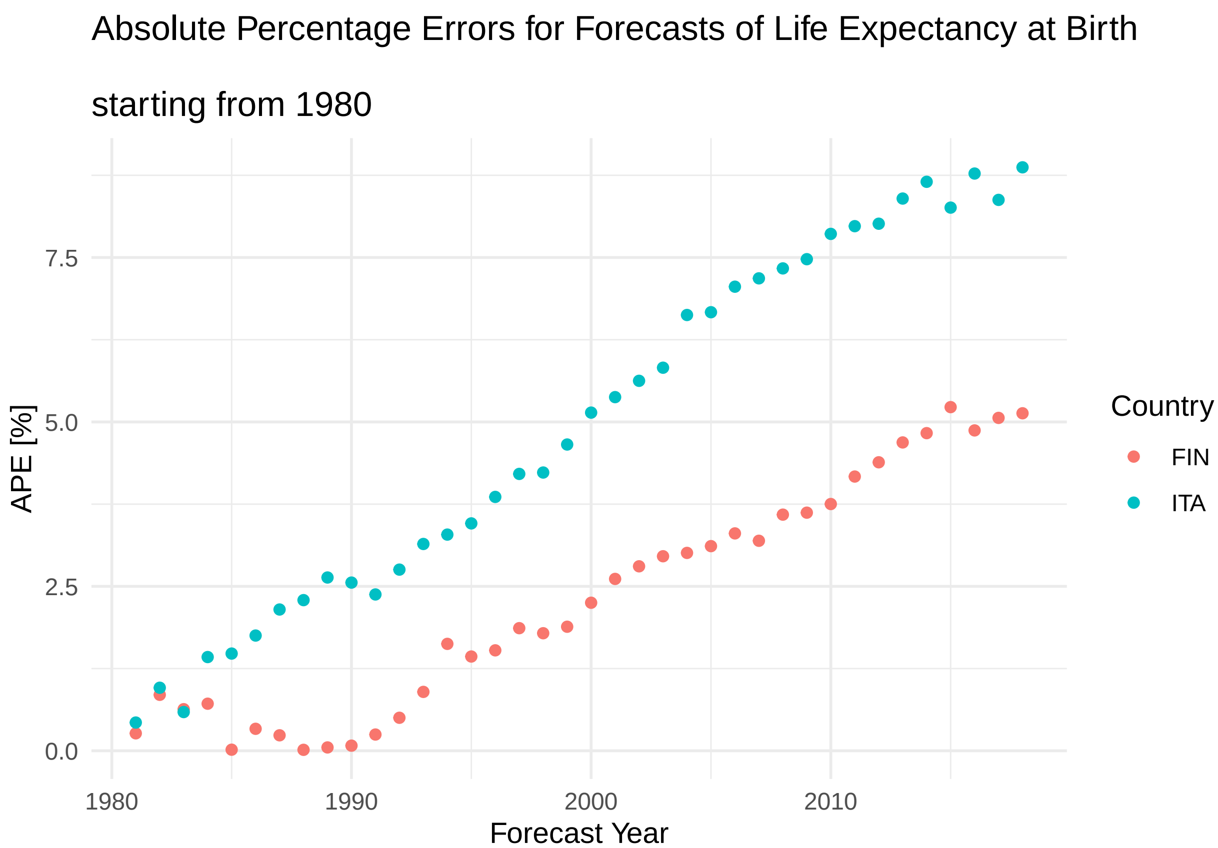
```r
    p.e0.male[i, j] <- p.lt.male[[i]][[j]]$lt$ex[p.lt.male[[i]][[j]]$lt$x==age.ex]
  }
}

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

#creating one (long format) data frame with observed and forecast e0 for ggplot
p.e0.male.long <- gather(data = p.e0.male, key = "Year", value = "p.ex", 1:fh)
p.e0.male.long$Year <- as.numeric(p.e0.male.long$Year)
e0.male.long <- left_join(lt.male.long.ex[,c("cntr", "Year", "Age", "ex")],
                            p.e0.male.long)


# plot observed and forecast e0 together
ggplot(data = e0.male.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, male") +
  xlab("Year") +
  ylab("e0") +
  labs(col = "Country") +
  scale_x_continuous(breaks = seq(1850, fh.end, by = 20), minor_breaks = NULL)
```




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


# plots of AE and PE
ggplot(data = e0.male.long, mapping = aes(x = Year, y = FE, col = cntr)) +
  geom_point(data = subset(e0.male.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 starting from 1980") +
  xlab("Forecast Year") +
  ylab("FE") +
  labs(col = "Country")
```




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




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

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

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

error.male
```

```
##             FIN       ITA
## ME    -1.751655 -3.851652
## MdAPE  2.068373  4.899029
```

3. Forecasts of Female Finnish and Italian Life Expectancy

Exercise: 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?

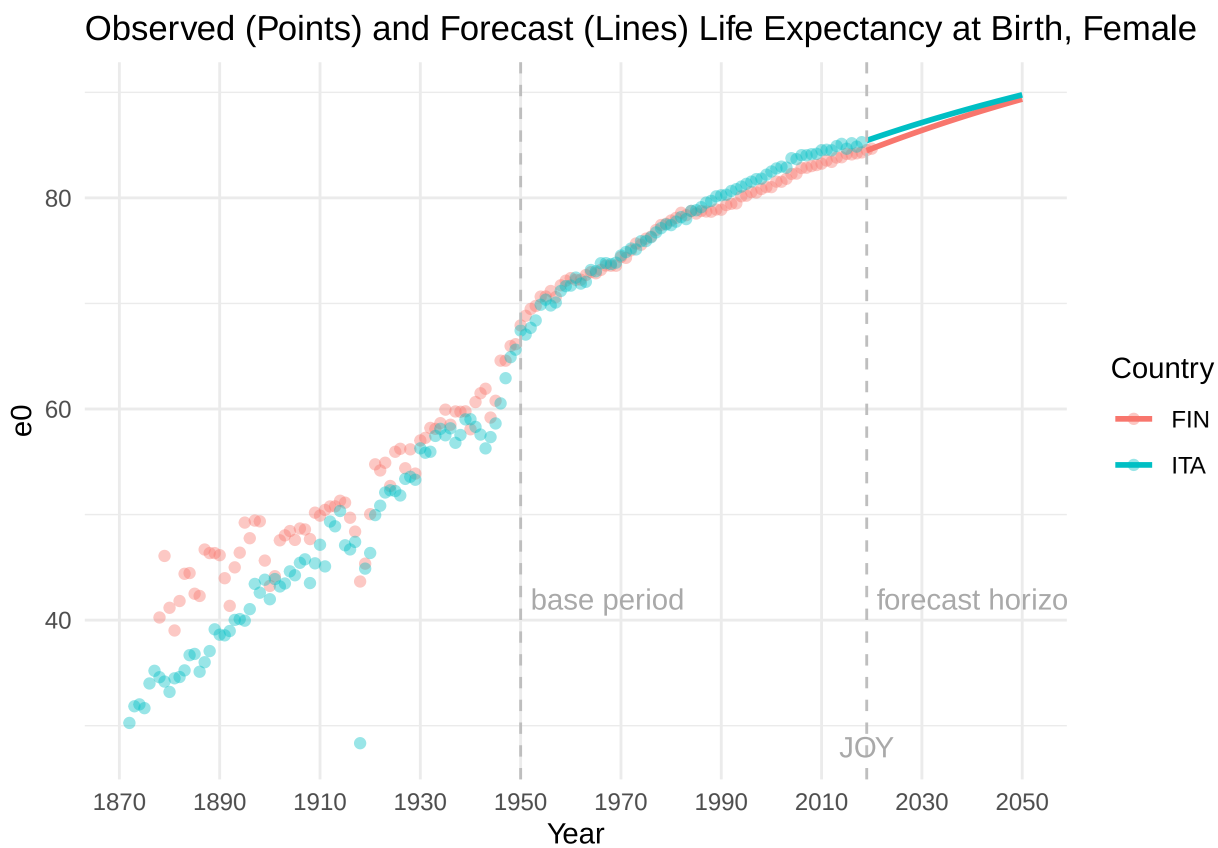
```r
### fh until 2050 ----

### 5. ----
```

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

# defining base period and forecast horizon ####
bp <- 1950:2018 # base period of input data
fh.start <- 2019 # first year for forecast
fh.end <- 2050 # last year for forecast
fh <- length(fh.start:fh.end) # number of forecast years

# excluding data before 1950 and after 2018
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: 2019 - 2050
## 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: 2019 - 2050
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
```

```r
# 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

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)
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)

# 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)
```



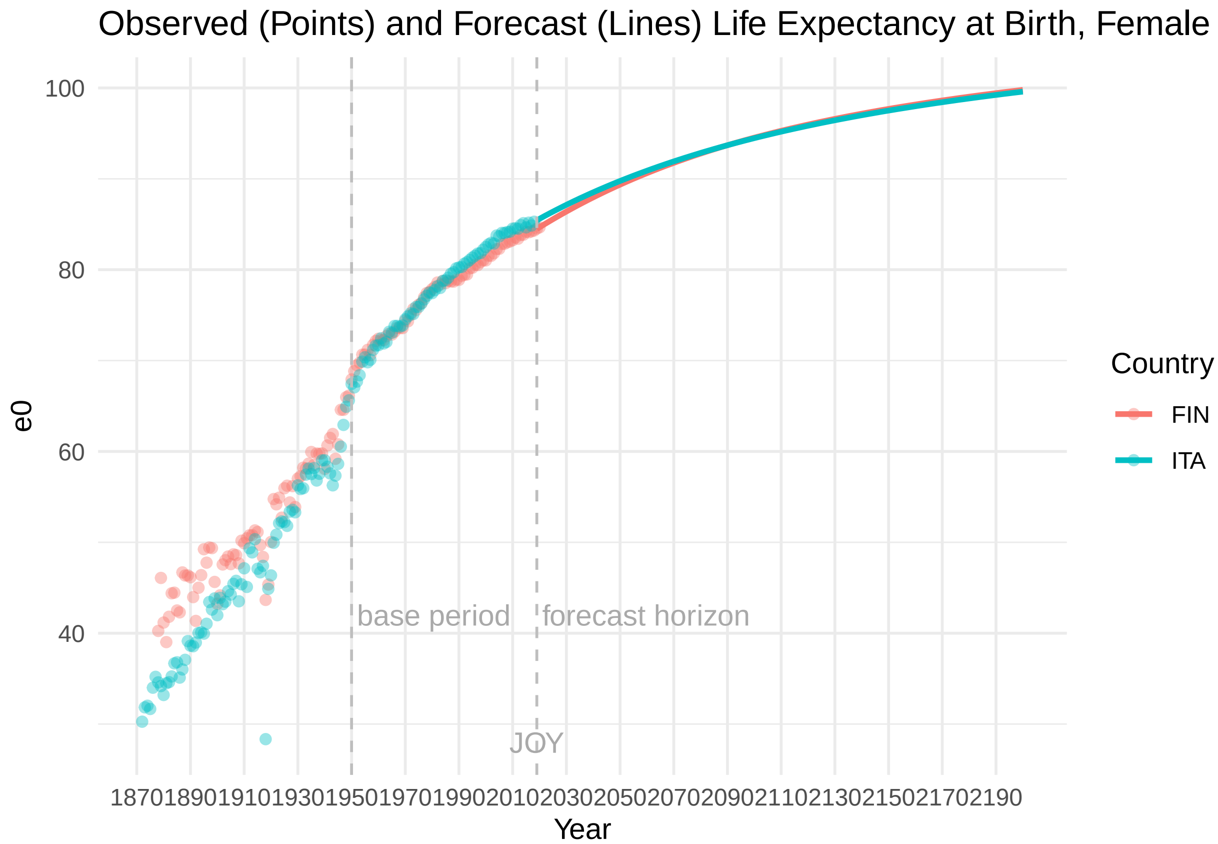

```
### fh until 2200 ----

#### 5. ----

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

# defining base period and forecast horizon ####
bp <- 1950:2018 # base period of input data
fh.start <- 2019 # first year for forecast
fh.end <- 2200 # last year for forecast
fh <- length(fh.start:fh.end) # number of forecast years

# excluding data before 1950 and after 2018
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))
```

```r
# 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: 2019 - 2200
## 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: 2019 - 2200
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
```

```r
# 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

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)
```

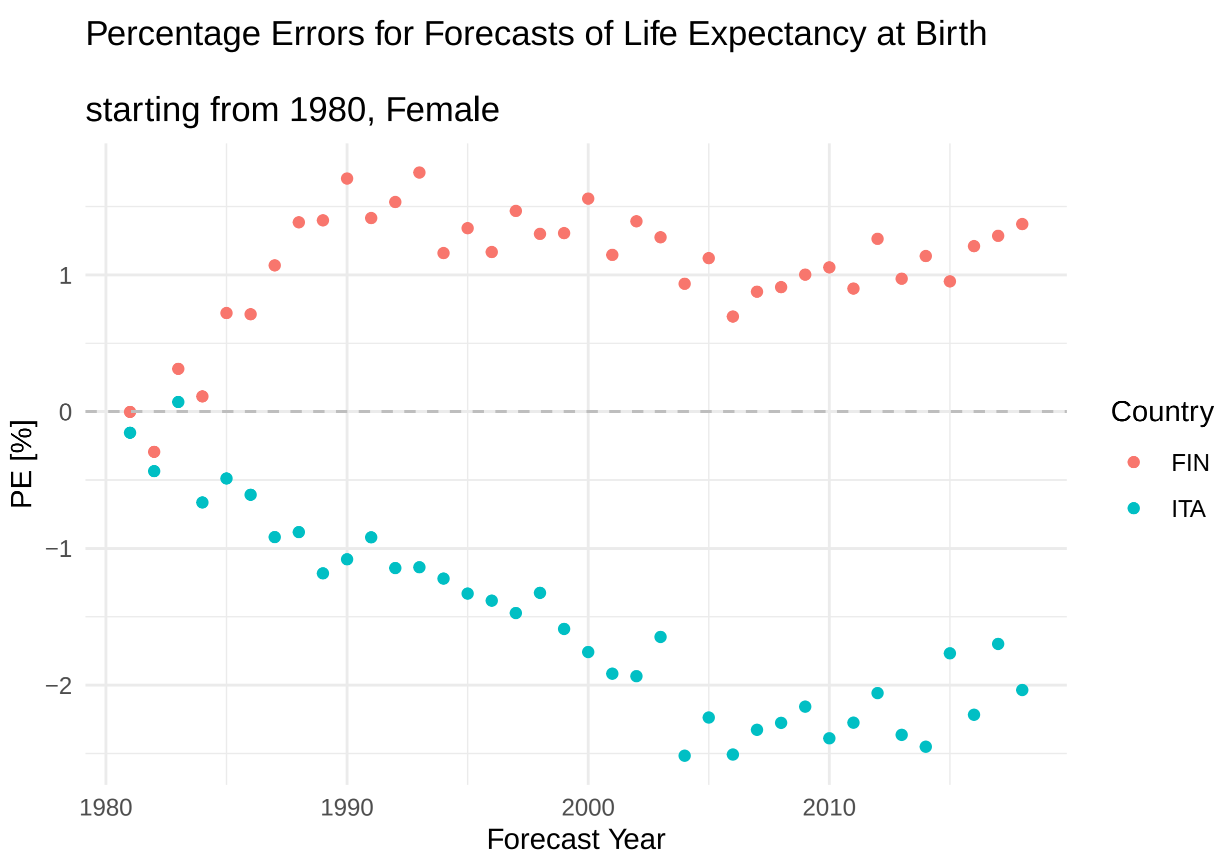
```
colnames(p.e0.female) <- fh.start:fh.end
p.e0.female$cntr <- HMD.countries
p.e0.female$Age <- age.ex
lt.female.long.ex <- lt.female.long[which(lt.female.long$Age == age.ex),]

#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)
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)


# 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)
```



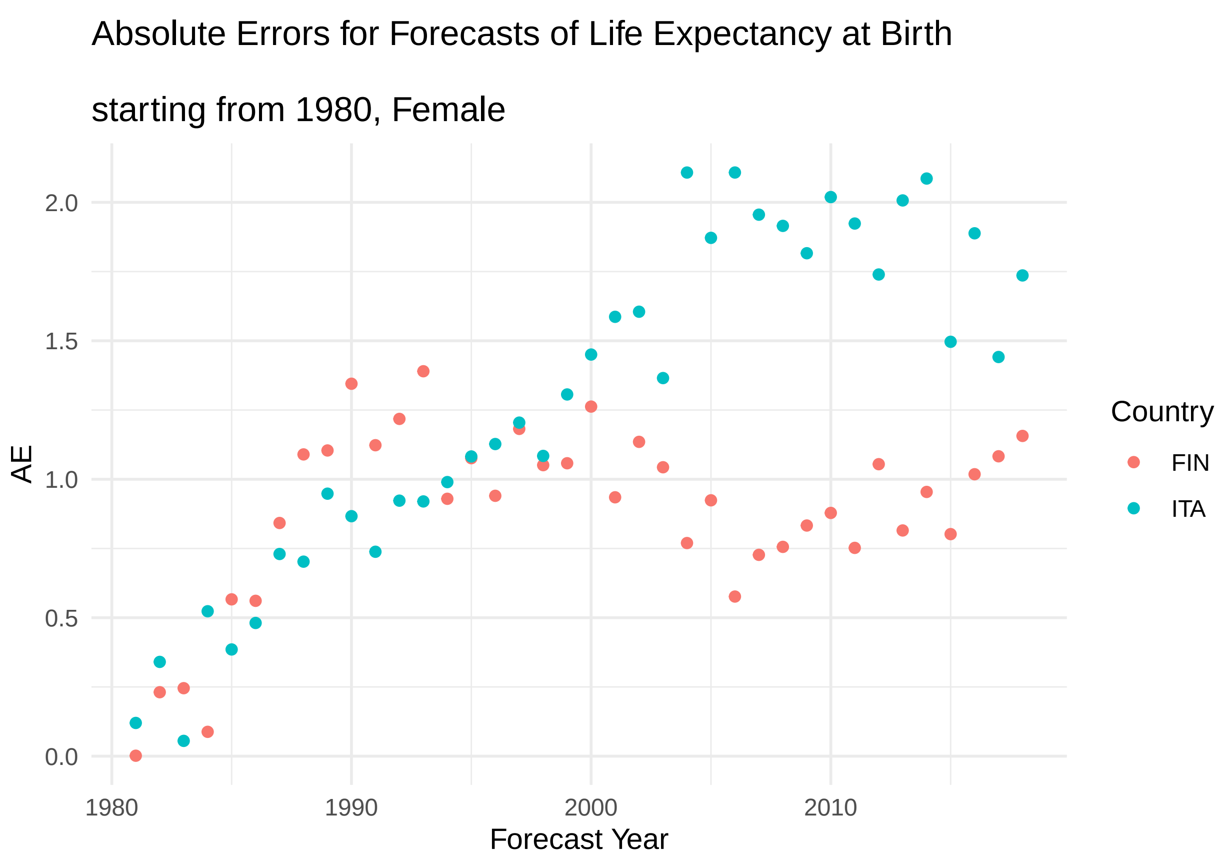
4. More Forecast Error Measures (FIN, ITA, female)

Exercise: 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?

```
### 5. ----

# calculate absolute error and percentage error ####
e0.female.long$AE <- abs(e0.female.long$FE)
e0.female.long$PE <- (e0.female.long$FE/e0.female.long$ex)*100


# plots of PE and AE ####
ggplot(data = e0.female.long, mapping = aes(x = Year, y = PE, 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("Percentage Errors for Forecasts of Life Expectancy at Birth
          \nstarting from 1980, Female") +
  xlab("Forecast Year") +
  ylab("PE [%]") +
  labs(col = "Country")
```




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




```
# calculate some summary error measures ####
error.female <- matrix(NA, ncol = length(HMD.countries), nrow = 6)
colnames(error.female) <- HMD.countries
rownames(error.female) <- c("ME", "MAE", "MdAE", "MAPE", "MdAPE", "RMSE")

# ME
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))
# MAE
error.female[2,] <- cbind(mean(filter(e0.female.long, cntr == HMD.countries[1])$AE, na.rm = TRUE),
                          mean(filter(e0.female.long, cntr == HMD.countries[2])$AE, na.rm = TRUE))
# MdAE
error.female[3,] <- cbind(median(filter(e0.female.long, cntr == HMD.countries[1])$AE, na.rm = TRUE),
                          median(filter(e0.female.long, cntr == HMD.countries[2])$AE, na.rm = TRUE))
# MAPE
error.female[4,] <- cbind(mean(filter(e0.female.long, cntr == HMD.countries[1])$APE, na.rm = TRUE),
                          mean(filter(e0.female.long, cntr == HMD.countries[2])$APE, na.rm = TRUE))

# MdAPE
error.female[5,] <- 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))
# RMSE
error.female[6,] <- cbind(sqrt(mean(filter(e0.female.long, cntr == HMD.countries[1])$FE^2,
                                    na.rm = TRUE)),
                          sqrt(mean(filter(e0.female.long, cntr == HMD.countries[2])$FE^2,
                                    na.rm = TRUE)))
```

```
error.female
```

```
##              FIN       ITA
## ME    0.8697401 -1.277231
## MAE   0.8819793  1.280131
## MdAE  0.9374925  1.335772
## MAPE  1.0847004  1.540461
## MdAPE 1.1530154  1.618312
## RMSE  0.9390934  1.412126
```

5. Playing Around with Base Period, Forecast Horizon and JOY

Exercise: 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?

```
### 5. ----

# 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")


# defining base period and forecast horizon ####
bp <- 1910:1943 # base period of input data
fh.start <- 1944 # 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: 1944 - 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: 1944 - 2018
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
```

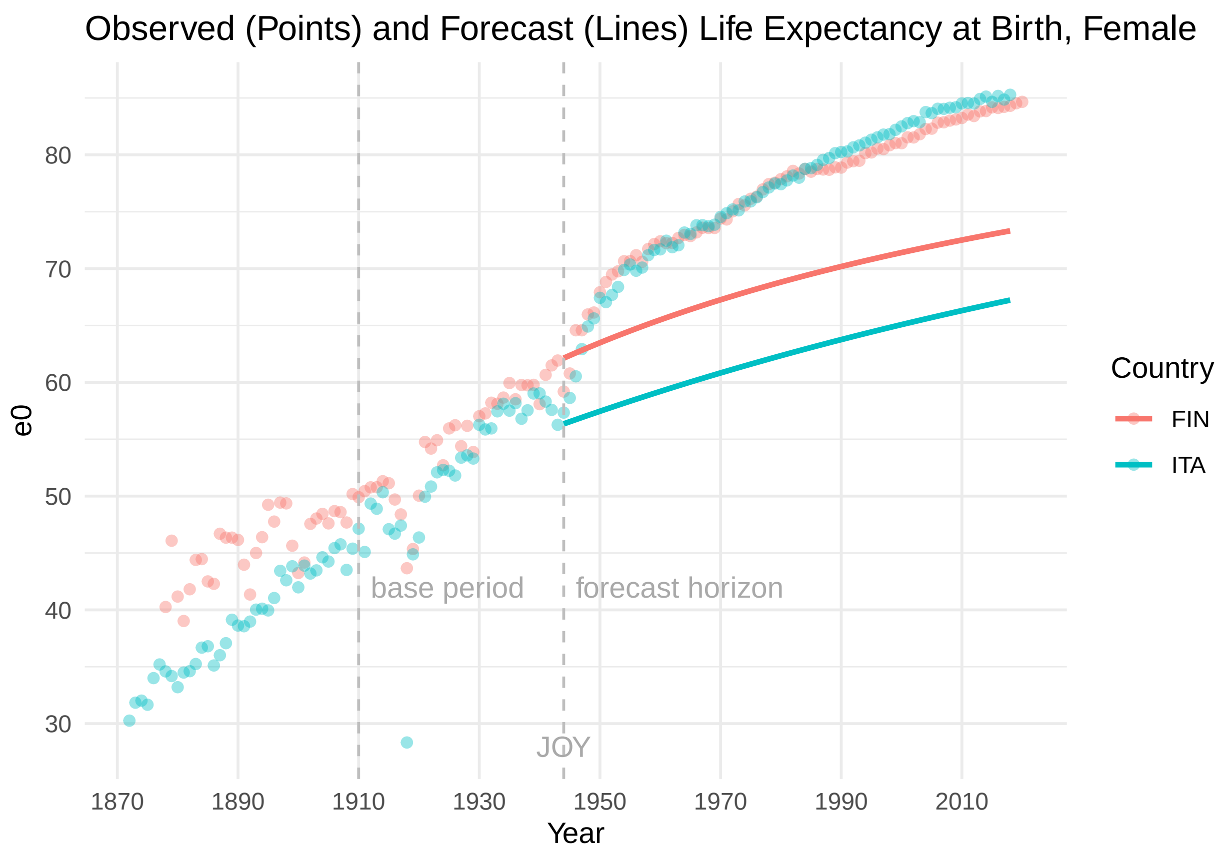
```
# extracting forecast mx values from Lee-Carter object
p.mx.female <- lapply(p.lc.female,
                      FUN = function(x){cbind(x$x, as.matrix(x$predicted.values))})

# 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

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
```

```r
p.e0.female.long <- gather(data = p.e0.female, key = "Year", value = "p.ex", 1:fh)
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)
```

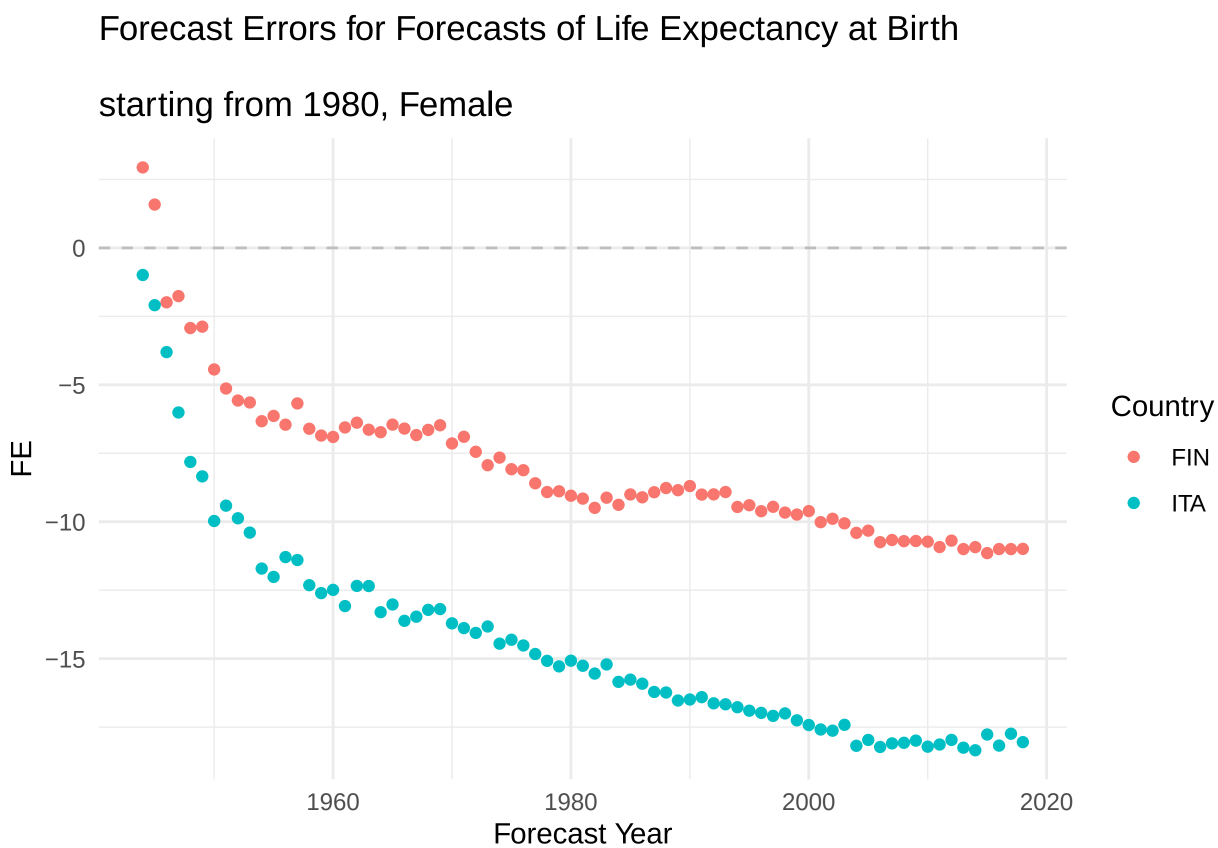
```
## Joining, by = c("cntr", "Year", "Age")
```

```r
# 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, 2020, by = 20), minor_breaks = NULL)
```

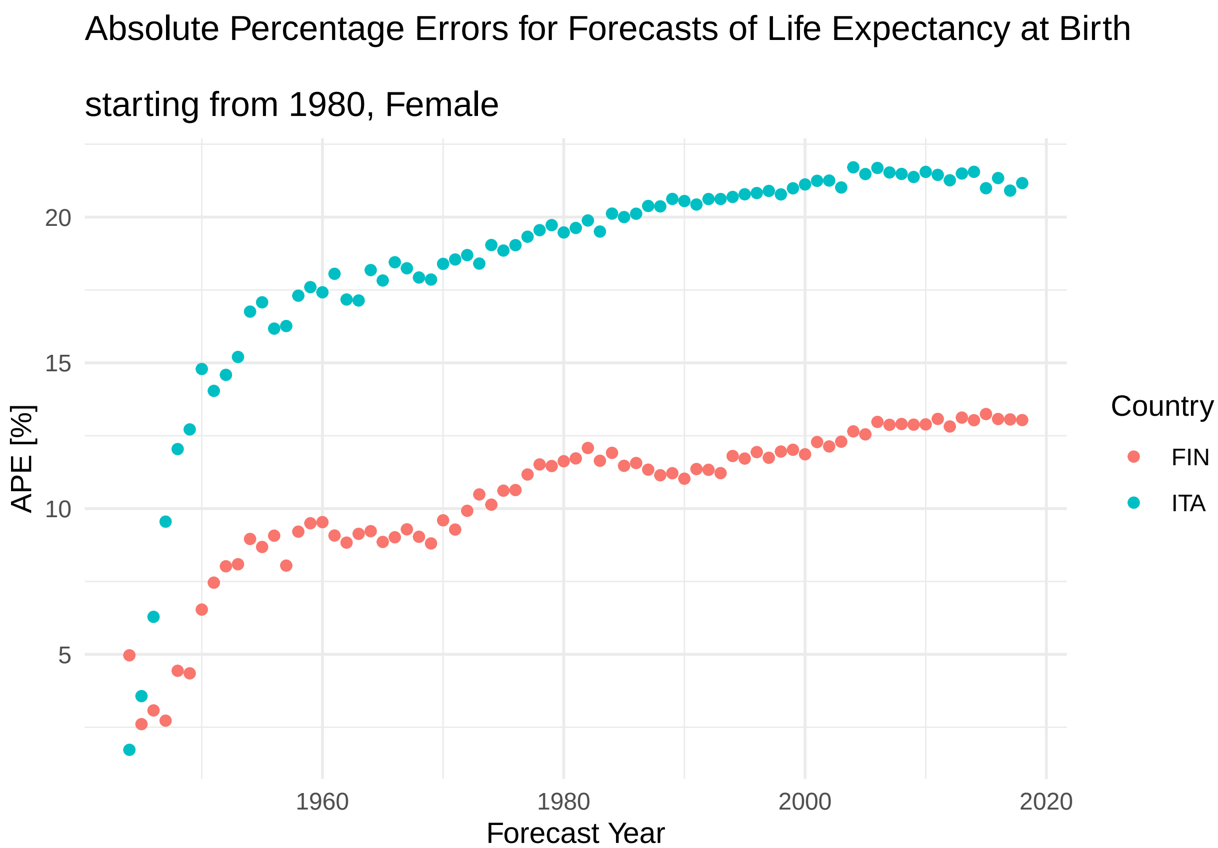
```
## Warning: Removed 140 row(s) containing missing values
## (geom_path).
```




```
# 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


# 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")
```




```r
# 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
```

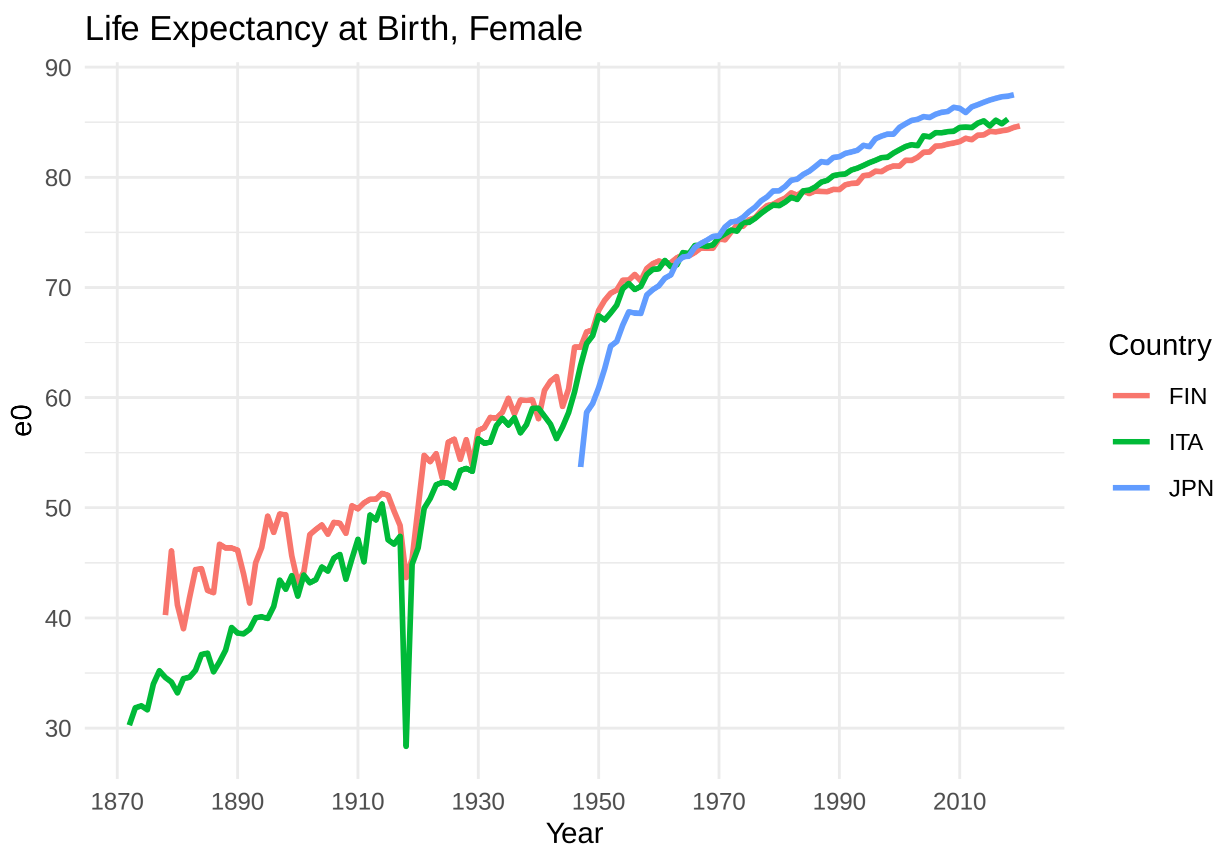
```
##             FIN       ITA
## ME    -7.934788 -14.41511
## MdAPE 11.216757  19.62657
```

6. Adding Japan to the Analyses of Female Mortality

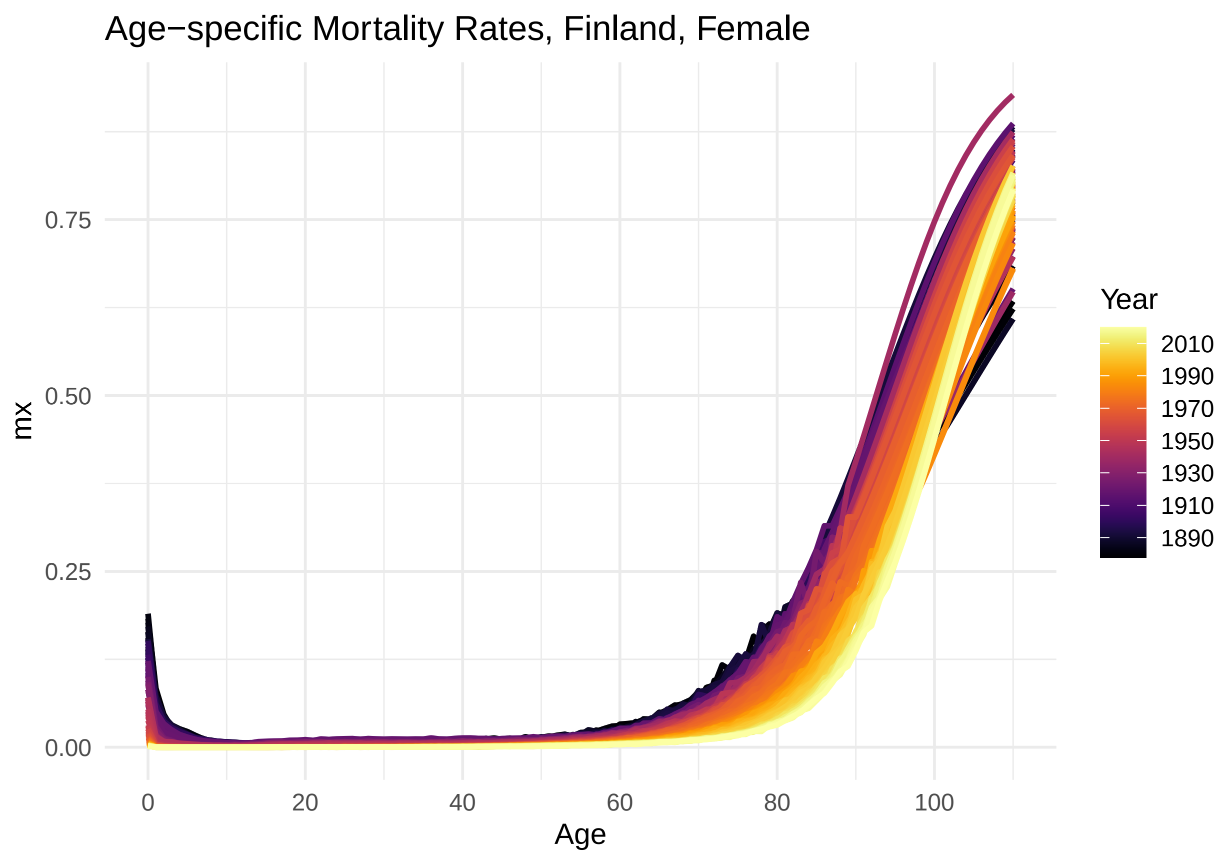
Exercise: Add another country to the analysis! Hint: You have to add the country code from the HMD to the variable `HMD.countries`.

```r
### 3. ----

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

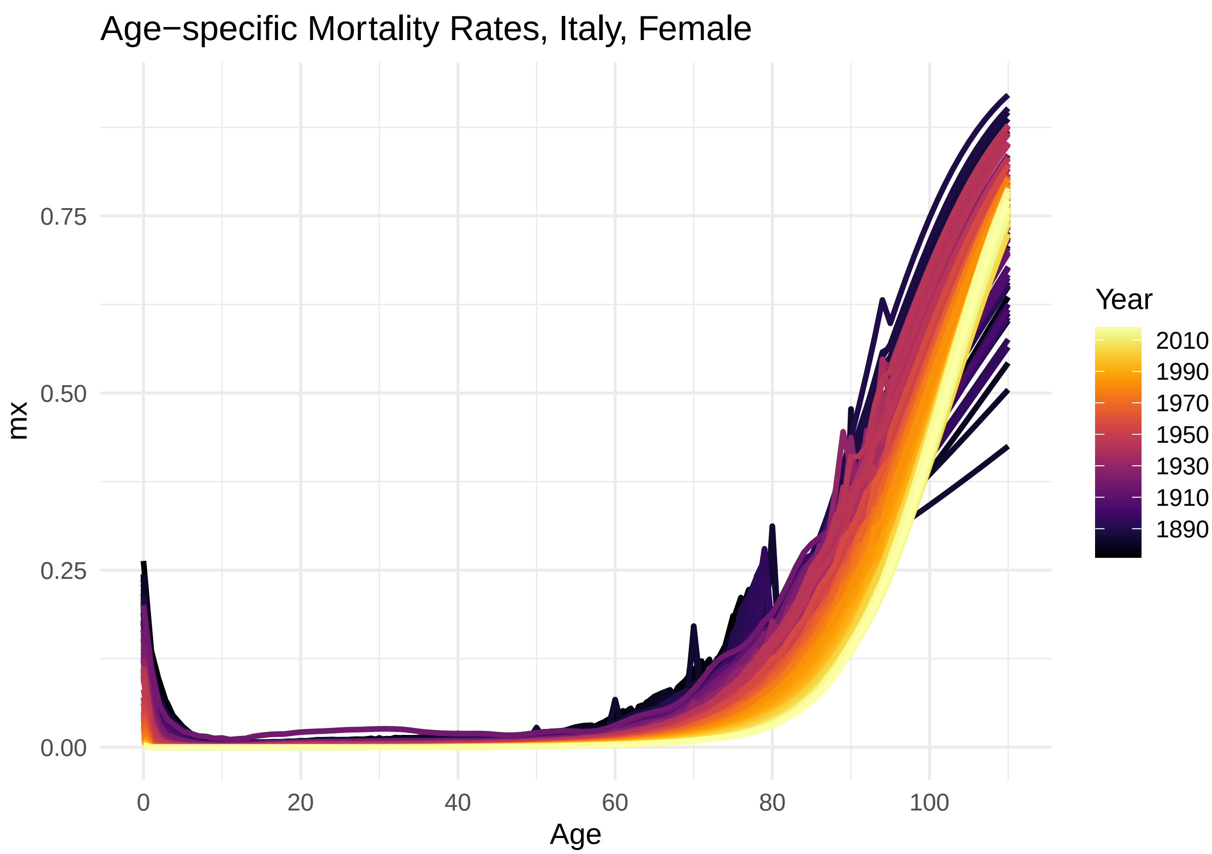
```r
# download life table data for Finland and Italy (female) and saving it in a list
lt.female <- list() # create an empty list

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


### 4. ----

# add country variable ####
lt.female[[1]]$cntr <- HMD.countries[1]
lt.female[[2]]$cntr <- HMD.countries[2]
lt.female[[3]]$cntr <- HMD.countries[3]

# 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)


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)
```



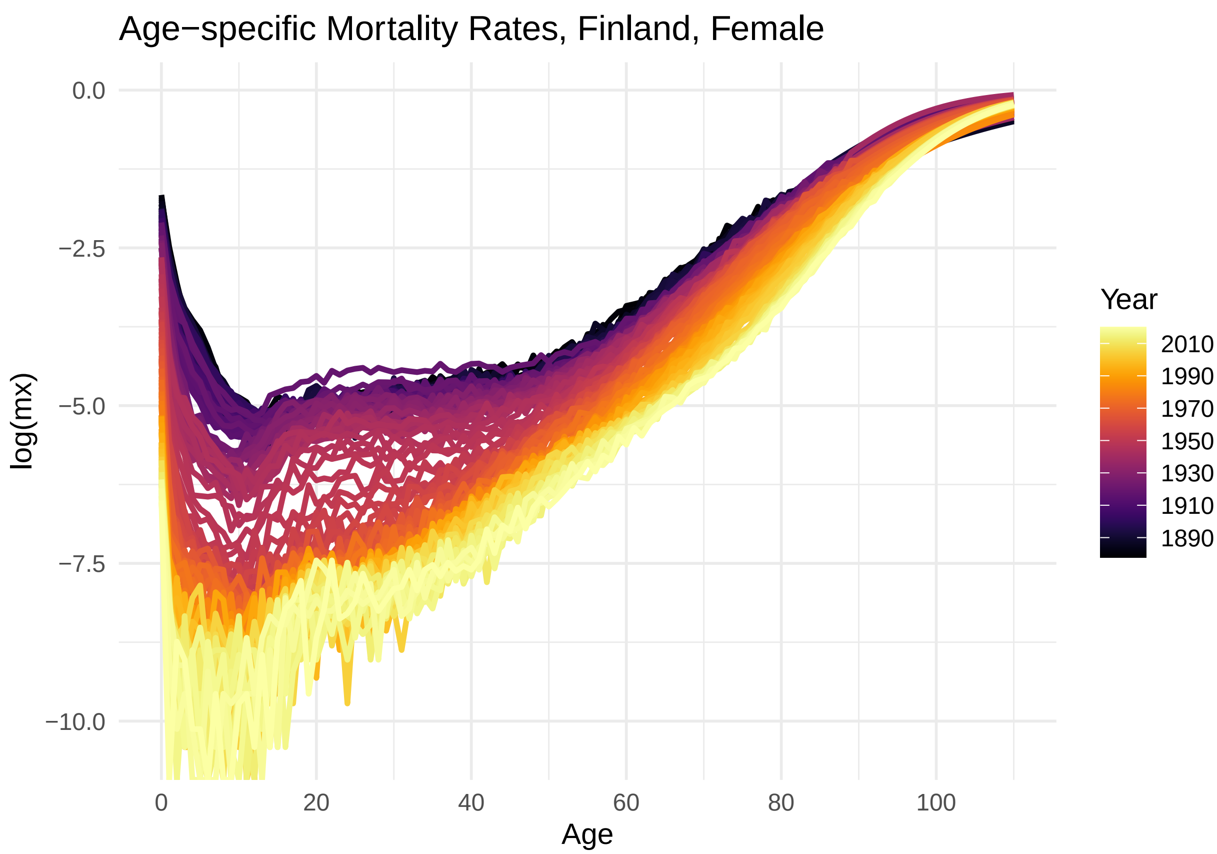

```
ggplot(data = subset(lt.female.long, cntr == HMD.countries[1]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality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



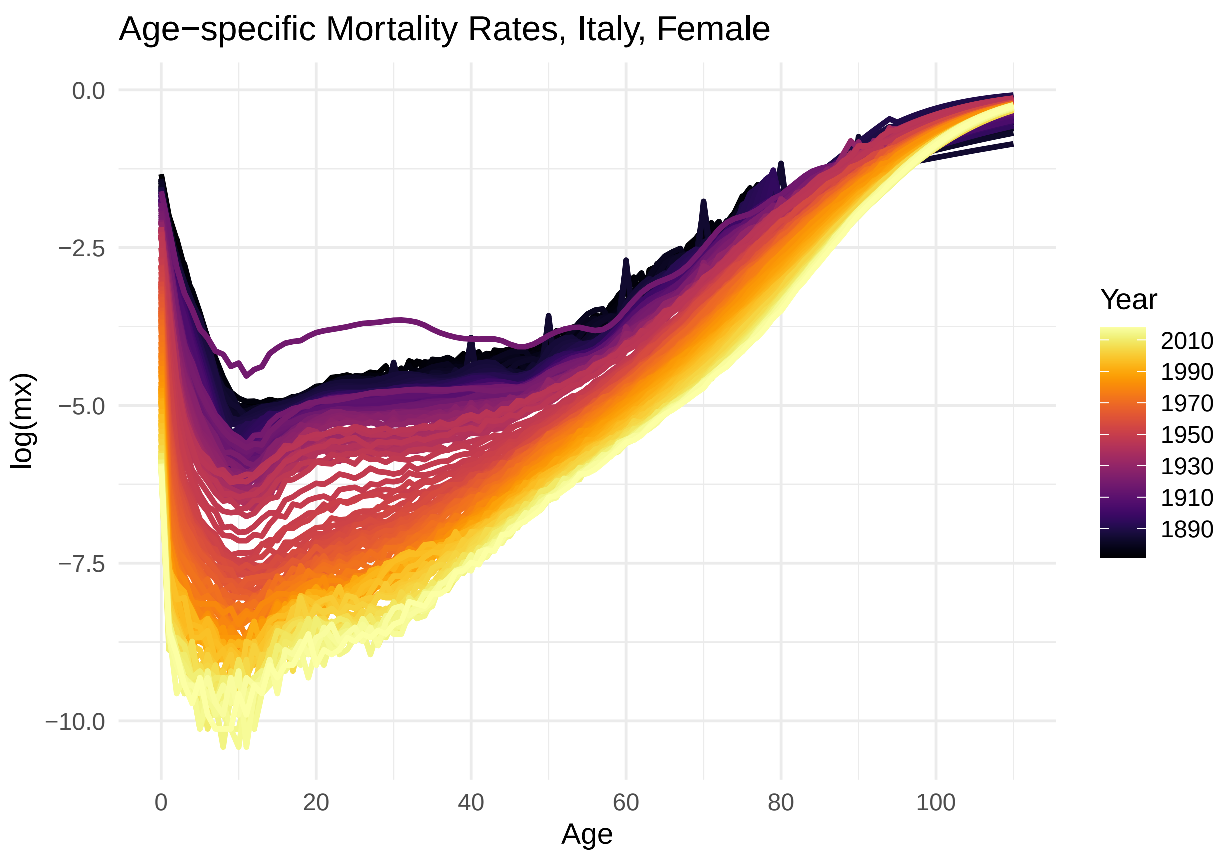

```r
ggplot(data = subset(lt.female.long, cntr == HMD.countries[2]),
       mapping = aes(x = Age, y = mx, group = Year)) +
  geom_line(aes(color = Year), size = 1) +
  scale_color_viridis(option = "B", name = "Year", breaks = seq(1850, 2020, by = 20)) +
  theme_minimal() +
  ggtitle("Age-specific Mortality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




```r
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)))
```



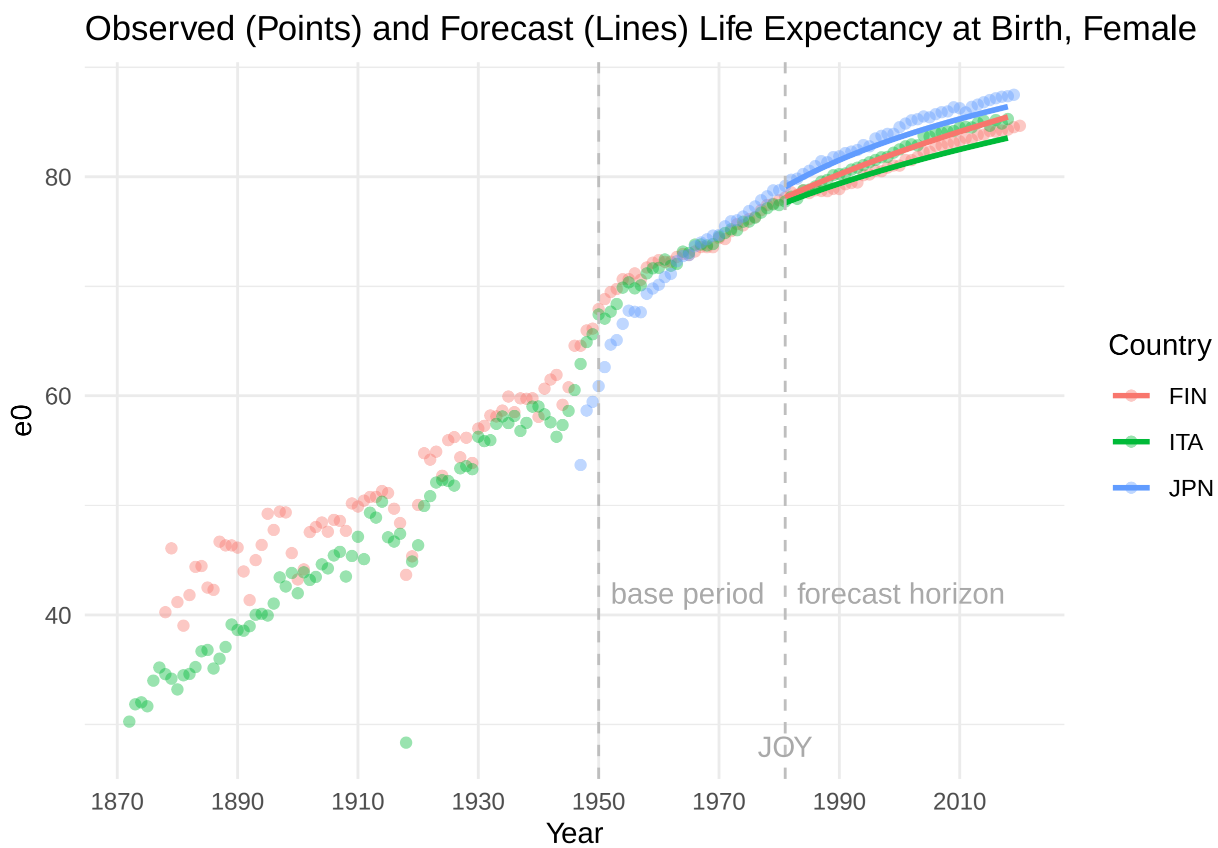

```
### 5. ----

# 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")


# 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
```

```r
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
```

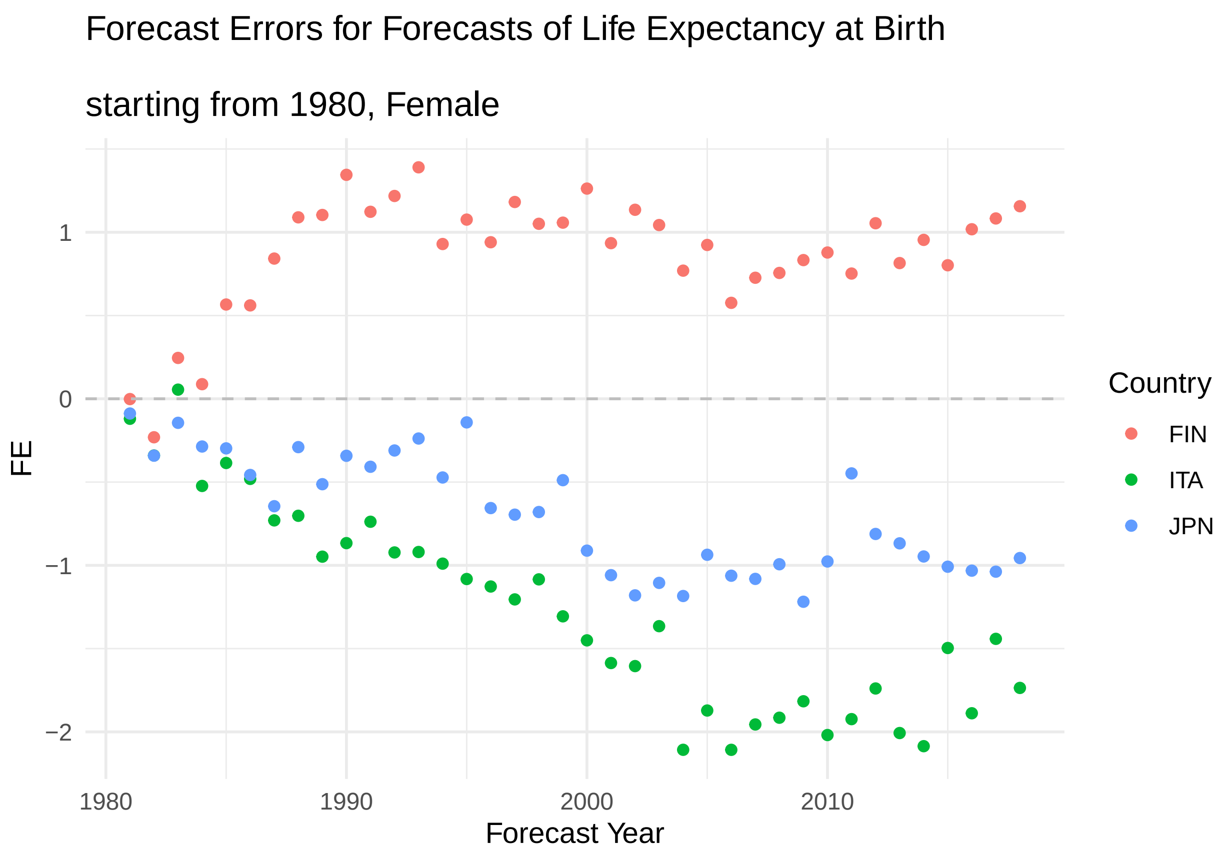
```
## $FIN
##
## Forecast: Lee-Carter Mortality Model
## Model   : log m[x,t] = a[x] + b[x]k[t]
## Call    : predict.LeeCarter(object = X[[i]], h = ..1, level = ..2)
## Ages  in forecast: 0 - 110
## Years in forecast: 1981 - 2018
## k[t]-ARIMA method: ARIMA(0,1,0) with drift
##
## $ITA
##
## Forecast: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

```r
# 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
}
```

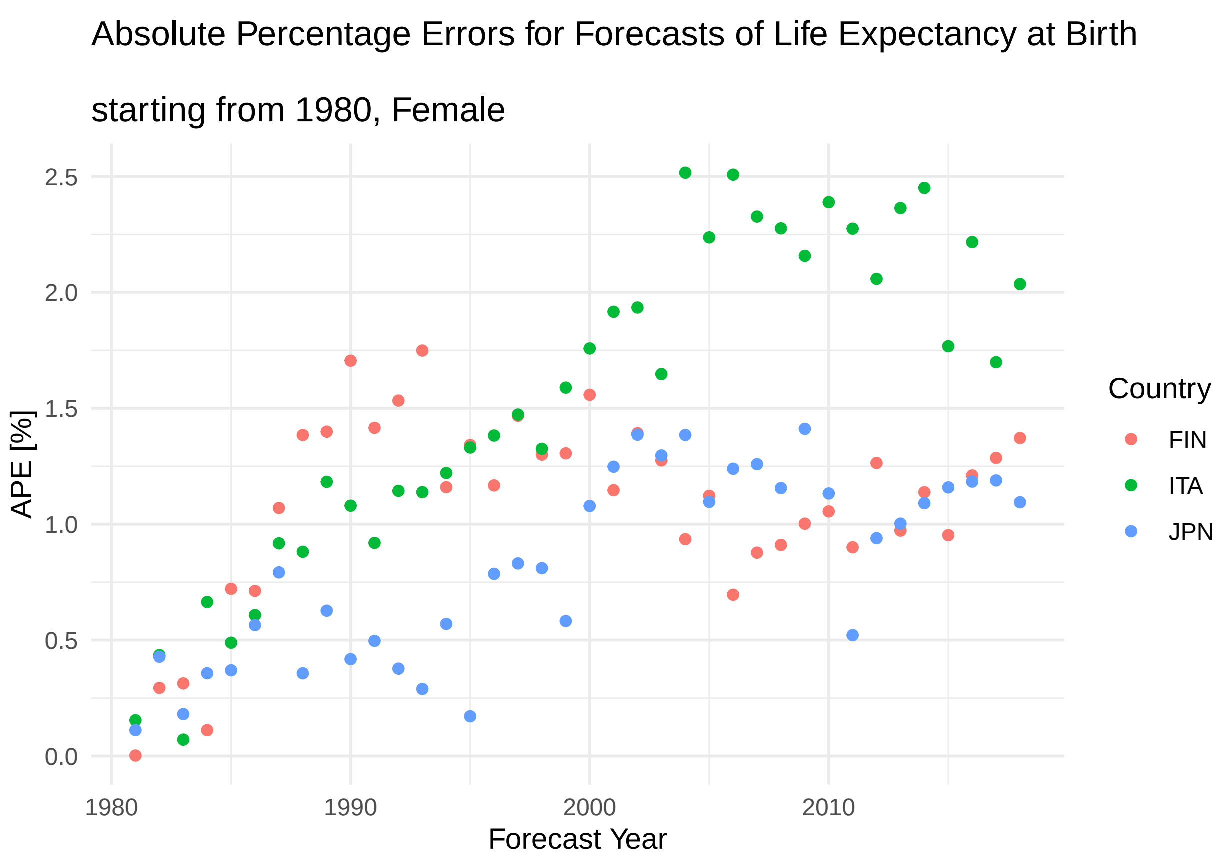
```r
names(p.lt.female) <- HMD.countries

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

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)
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)


# 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, 2020, by = 20), minor_breaks = NULL)
```




```r
# 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

# 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),
                          mean(filter(e0.female.long, cntr == HMD.countries[3])$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),
                          median(filter(e0.female.long, cntr == HMD.countries[3])$APE, na.rm = TRUE))

error.female
```

```
##             FIN       ITA        JPN
## ME    0.8697401 -1.277231 -0.6925015
## MdAPE 1.1530154  1.618312  0.8204250
```