class notes

Tim Riffe

7/9/2021

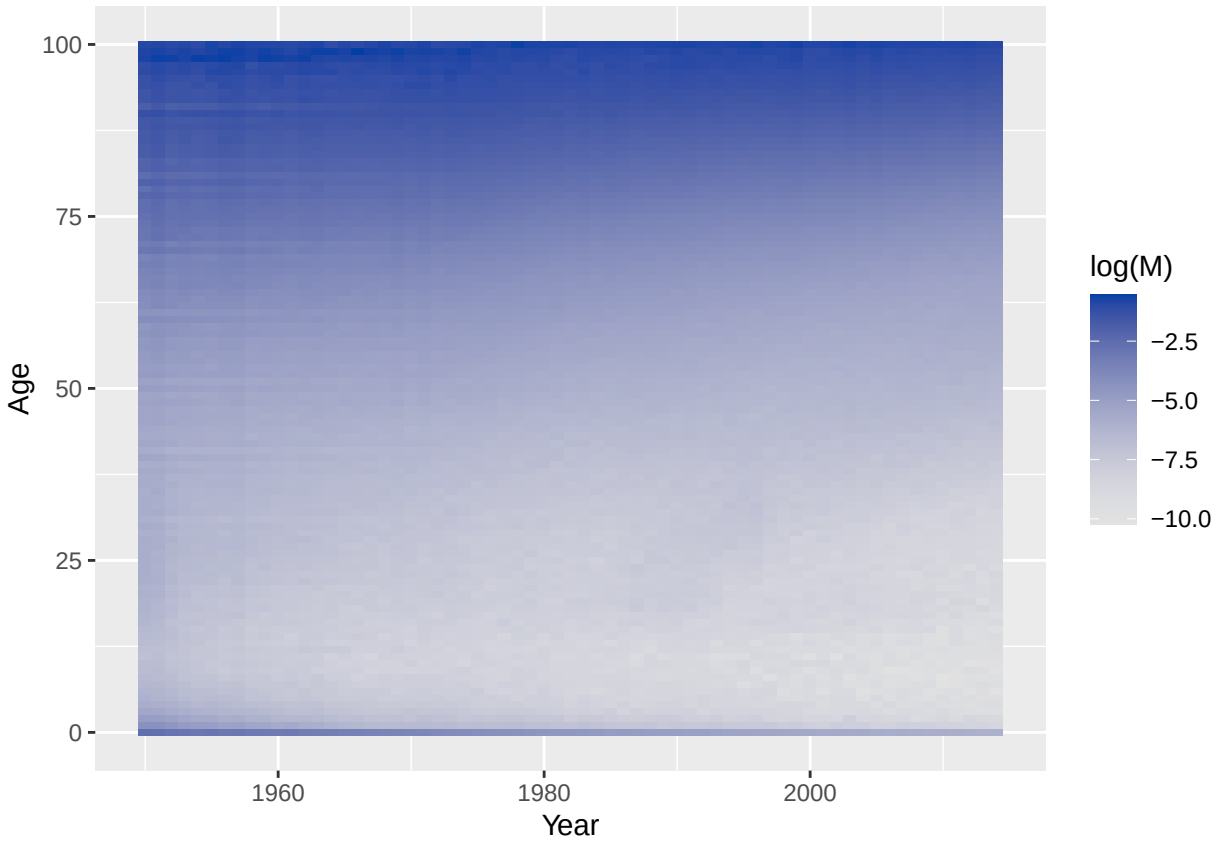
Summary

We want to model a demographic parameter in such as way as to find a linear pattern that is can be plausibly extrapolated into the future. For mortality, we first look to the rates. We will find rough linearity within age by logging the rates.

Data

Visualize the mortality surface

```
library(colorspace)
ES %>%
  ggplot(aes(x = Year, y = Age, fill = log(M))) +
  geom_tile() +
  scale_fill_continuous_sequential()
```



The Lee-Carter model

$$ln(m_x(t)) = \alpha_x + \beta_x \kappa(t) + \epsilon_x(t)$$

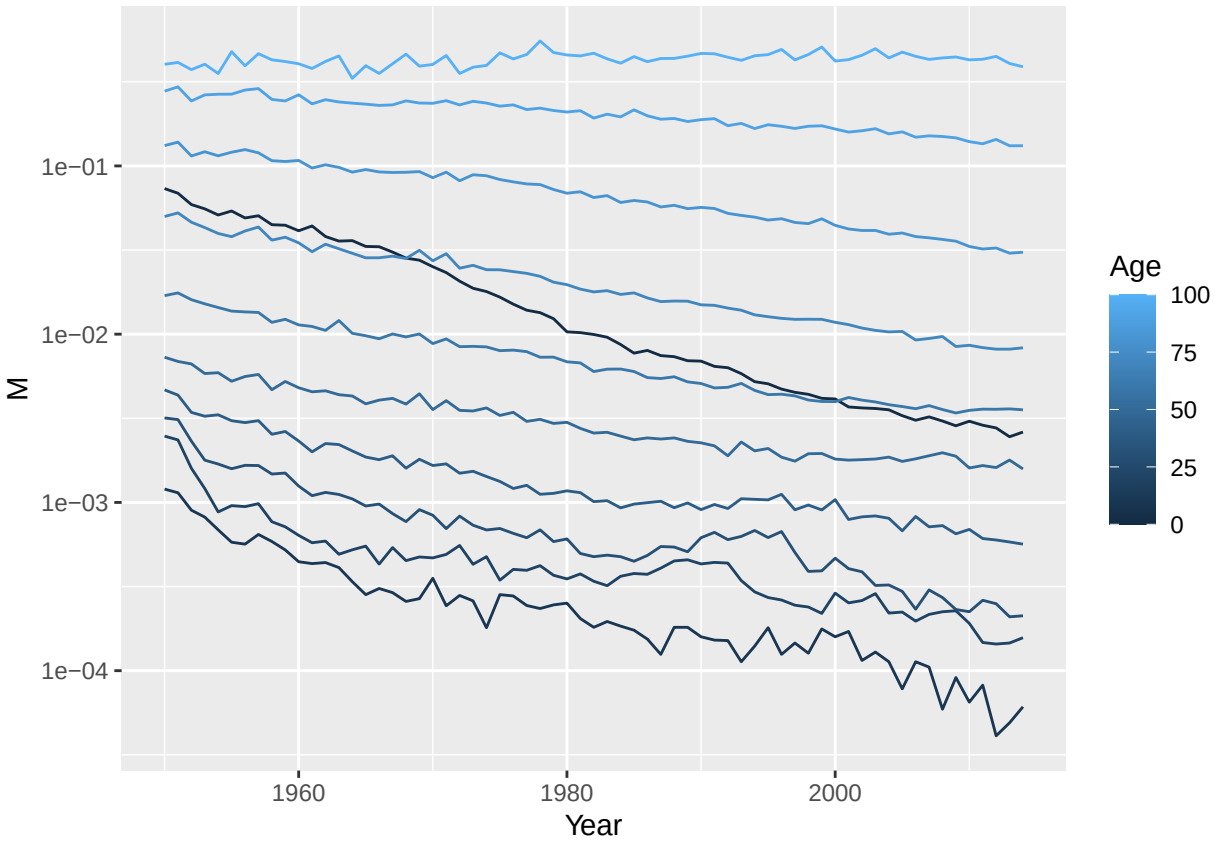
where α_x is the age-specific average of the log death rates ($m_x(t)$). The β_x and κ_t are found with a Singular Value Decomposition (SVD) and consist in the normalized first right and left singular vectors. They are interpreted as the age-pattern (age-specific rate of change relative to $\kappa(t)$) of mortality change and the time-pattern of mortality change. Mortality is forecast by extrapolating $\kappa(t)$. The term $\epsilon_x(t)$ is the error of the model.

On video I used a sheet of paper, curved, tilted, and twisted, to describe how the Lee Carter model works. The curve in the paper is the base age pattern α_x, assume same every year. The tilt in the paper is the time trend $\kappa(t)$, and the slight twist is β_x (different time trend per age).

check for log-linearity

The Lee-Carter approach is premised on gradual log-linear change in mortality rates

```
ES %>%
  filter(Age %% 10 == 0) %>%
  ggplot(aes(x = Year, y = M, color = Age, group = Age)) +
  geom_line() +
  scale_y_log10()
```



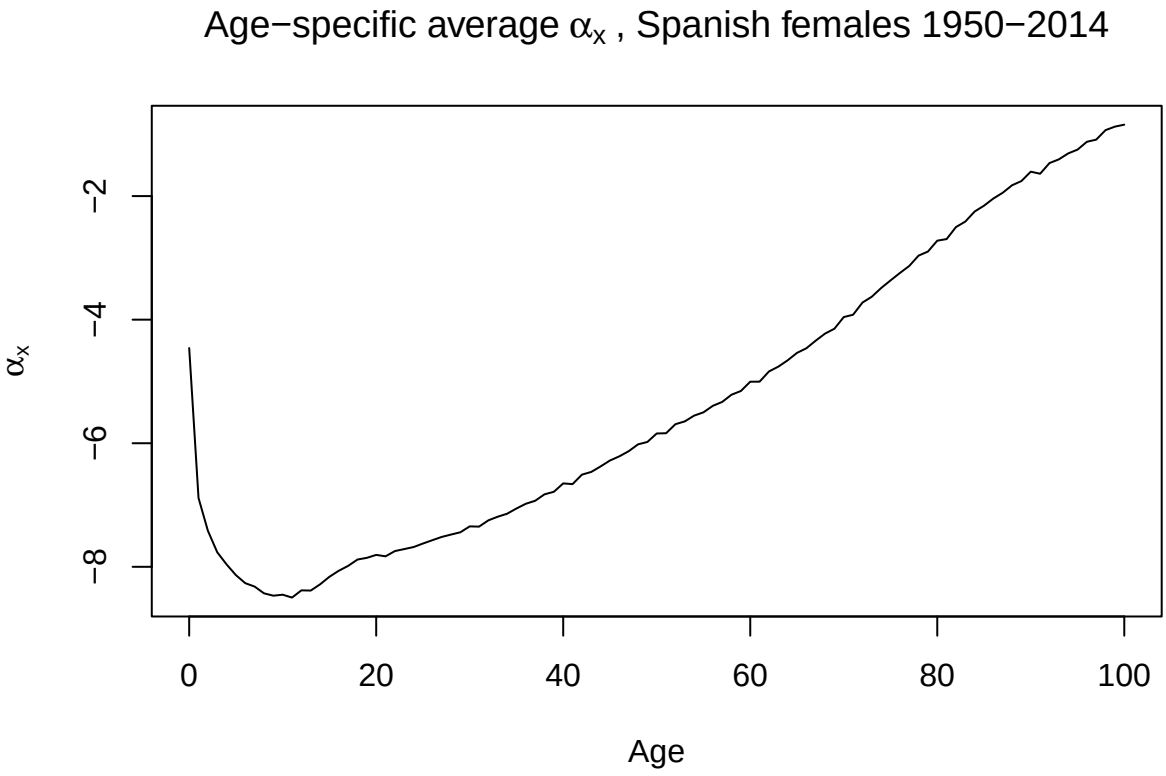
Let's go step by step!

Calculate the age-specific average of the log death rates.

```
#Log transformed data
lM <-
  ES %>%
  select(Year, Age, M) %>%
  pivot_wider(names_from = Year, values_from = M) %>%
  column_to_rownames("Age") %>%
  as.matrix() %>%
  log()

ax  <- rowMeans(lM)
Age <- rownames(lM) %>% as.integer()
```

Examine α_x in a base-tastic plot

```
plot(Age, ax, type="l",
     ylab=expression(alpha[x]), xlab="Age",
     main=expression("Age-specific average"~alpha[x]~", Spanish females 1950-2014"))
```



Center the matrix on α_x

```
#Center the matrix
cent_lM <- lM - ax
```

Do SVD on the centered matrix

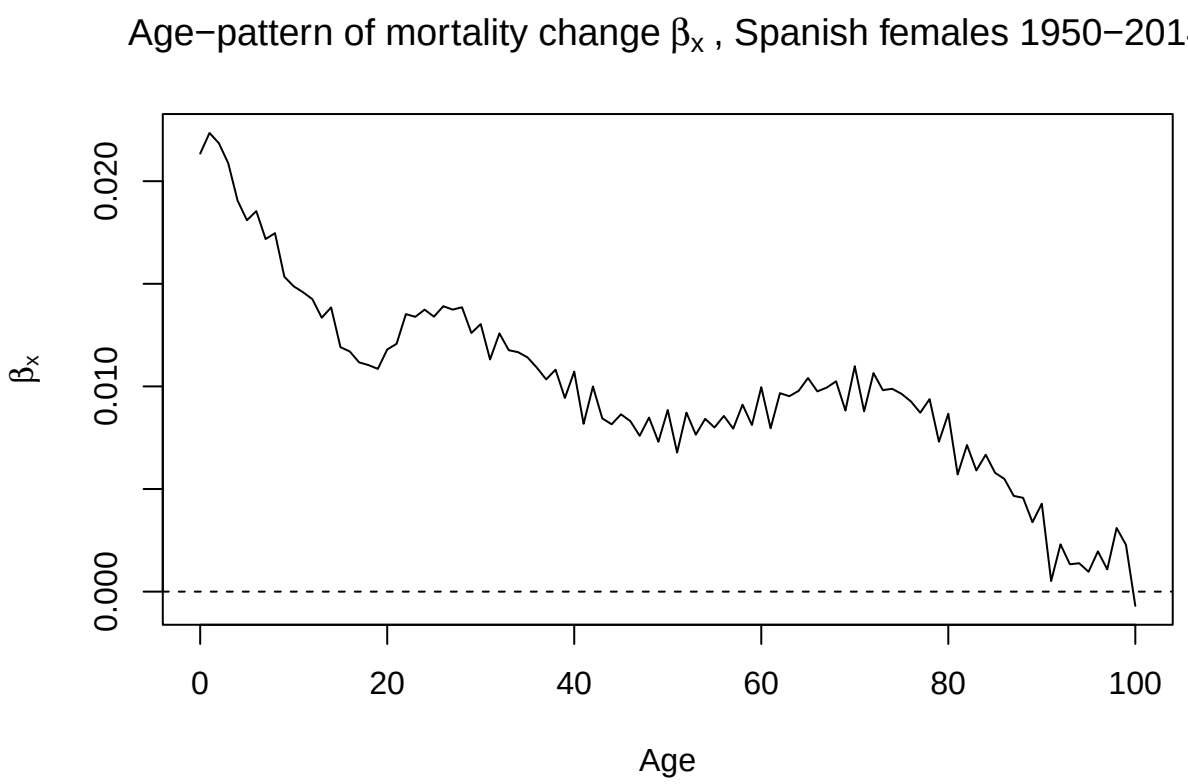
We now do singular value decomposition on the centered matrix of log mortality. Here, the left-singular vectors capture the age-pattern of mortality change and the right-singular vectors, the time-pattern of mortality change. For further insight on SVD in demography, take a look at Monica Alexander's 2017 blog post https://www.monicaalexander.com/posts/2017-12-16-svd/

```
#SVD: using first vectors and value only

svd_lM <- svd(cent_lM)

u      <- svd_lM$u[, 1]
v      <- svd_lM$v[, 1]
d      <- svd_lM$d[1]

#Explained variance

exp.var<- cumsum((svd_lM$d)^2 / sum((svd_lM$d)^2))[1]
exp.var
```

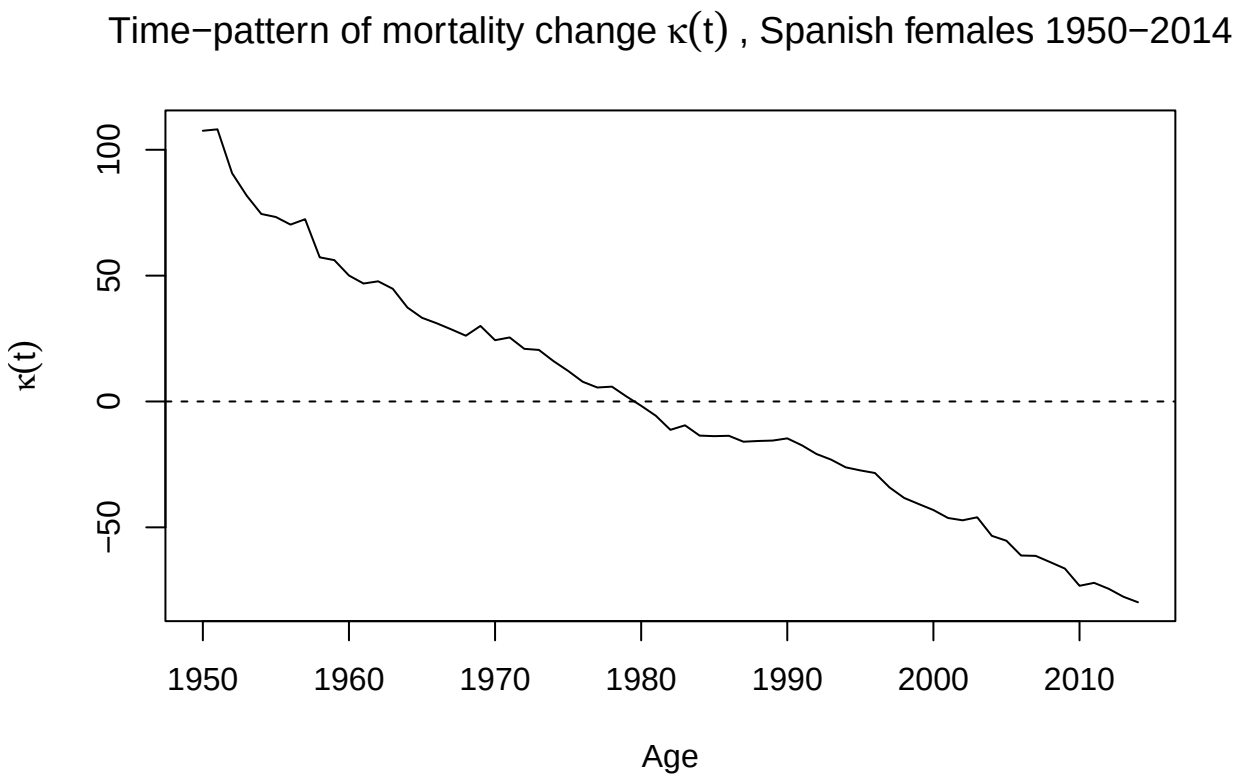
```
## [1] 0.9491602
```

In order to reach a unique solution, the parameters are normalized such that $\sum_x \beta_x = 1$ and $\sum_t \kappa(t) = 0$.

```r
#Normalization

Bx <- u / sum(u)
plot(Age, Bx, type="l",
     ylab=expression(beta[x]), xlab="Age",
     main=expression("Age-pattern of mortality change"~beta[x]~", Spanish females 1950-2014"))
abline(h=0, lty=2)
```




```r
Kt<-v*sum(u)*d
plot(1950:2014, Kt, type="l",
     ylab=expression(kappa(t)), xlab="Age",
     main=expression("Time-pattern of mortality change"~kappa(t)~", Spanish females 1950-2014"))
abline(h=0, lty=2)
```

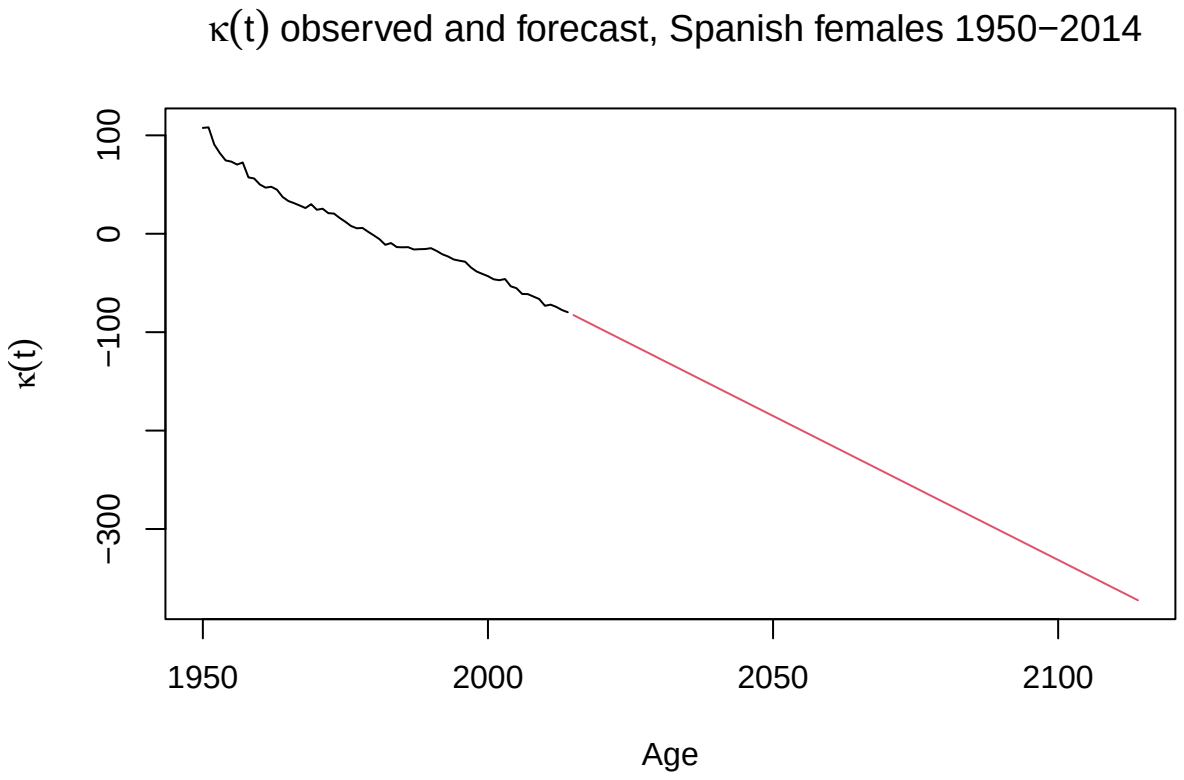


Forecast κ_t

```
#Forecast horizon
h      <- 1:100
Year   <- 1950:2014
#Drift
lK     <- length(Kt)
drift  <- (Kt[lK] - Kt[1]) / (lK - 1)
drift
```

```
## [1] -2.926557
```

```
#Forecast
Kt.fcst <- Kt[lK] + drift * h

plot(Year, Kt, type = "l", ylim=c(min(Kt.fcst), max(Kt)),
     xlim = c(Year[1], Year[length(Year)]+h[length(h)]),
     ylab = expression(kappa(t)), xlab="Age",
     main = expression(~kappa(t)~"observed and forecast, Spanish females 1950-2014"))
lines(Year[length(Year)]+h, Kt.fcst, col=2)
```



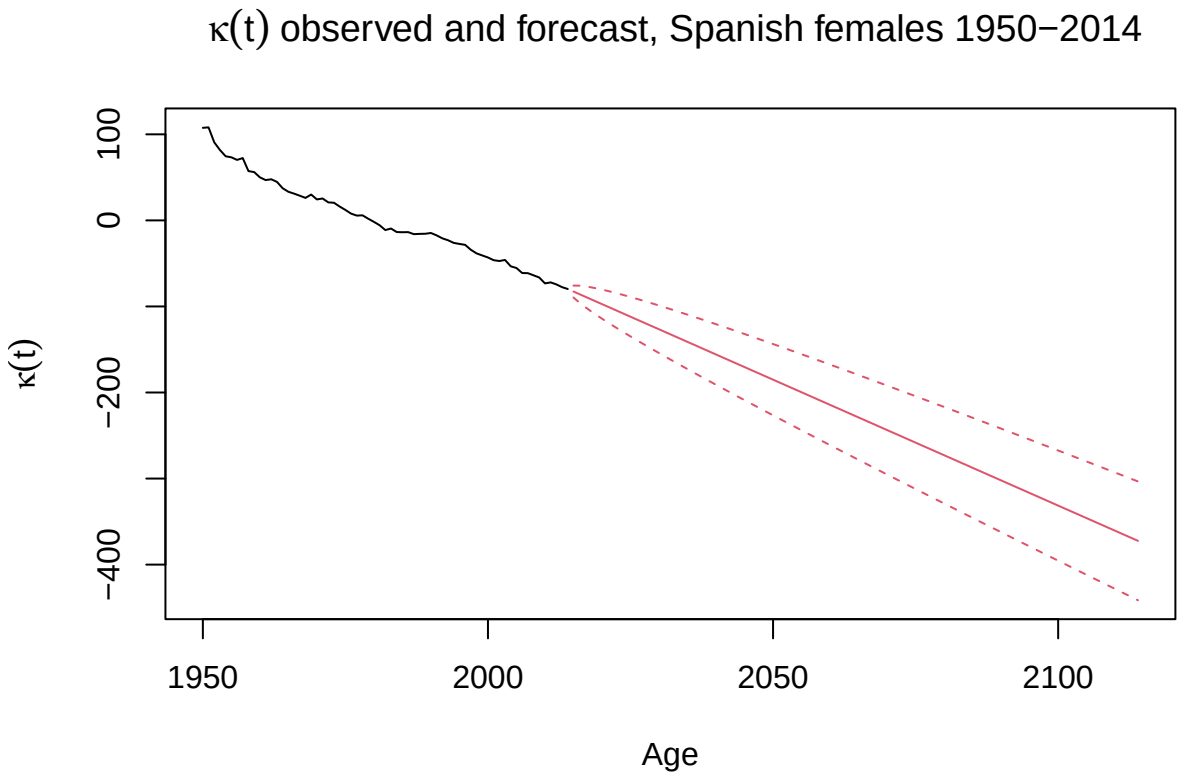
Find the prediction intervals (PI)

The prediction intervals (PI) are found by finding the standard errors of the estimates (SE) from the random walk with drift model. The SE indicates the uncertainty with one year forecast ahead. It is here assumed that the SE increase with the forecast horizon (h) square root.

$$PI(t+h) = \kappa(t+h) + / - z_{1-a} SE \sqrt{h}$$

where z is the z-score at level a

```
# Fit random walk with drift
fit.Kt <- Kt[1:(lK - 1)] + drift

# Find SE (RMSE)
se     <- sqrt(sum((Kt[2:lK] - fit.Kt)^2) / (length(Kt) - 1))

# Calculate PI
Kt.95u <- Kt.fcst + 1.96 * se * sqrt(h)
Kt.95l <- Kt.fcst - 1.96 * se * sqrt(h)


plot(Year, Kt, type="l", ylim=c(min(Kt.95l), max(Kt)),
     xlim=c(Year[1], Year[length(Year)]+h[length(h)]),
     ylab=expression(kappa(t)), xlab="Age",
     main=expression(~kappa(t)~"observed and forecast, Spanish females 1950-2014"))
lines(Year[length(Year)]+h, Kt.fcst, col=2)
lines(Year[length(Year)]+h, Kt.95u, col=2, lty=2)
lines(Year[length(Year)]+h, Kt.95l, col=2, lty=2)
```



Step 6: Reconstruct and back-transform the matrix

This is just taking the reverse steps of what we first did to M_x.

```
Bx         <- matrix(Bx, ncol = 1)
mx.fit     <- exp(Bx %*% Kt + ax)
mx.fcst    <- exp(Bx %*% Kt.fcst + ax)
mx.fcst95u <- exp(Bx %*% Kt.95u + ax)
mx.fcst95l <- exp(Bx %*% Kt.95l + ax)
```

Step 8 Let's grab all those pieces and make it tidy again for further calculations and processing.

```
# add dimension names
dimnames(mx.fit)     <- dimnames(lM)
dimnames(mx.fcst)    <- list(Age = 0:100, Year = 2014 + h)
dimnames(mx.fcst95u) <- list(Age = 0:100, Year = 2014 + h)
dimnames(mx.fcst95l) <- list(Age = 0:100, Year = 2014 + h)

# longer
mx_fit <-
  mx.fit %>%
  as_tibble(rownames = "Age") %>%
  pivot_longer(-Age,
               names_to = "Year",
               values_to = "M") %>%
  mutate(variant = "fitted")

mx_fcst <-
  mx.fcst %>%
  as_tibble(rownames = "Age") %>%
  pivot_longer(-Age,
```

```
               names_to = "Year",
               values_to = "forecast")
mx_fcst95u <-
  mx.fcst95u %>%
  as_tibble(rownames = "Age") %>%
  pivot_longer(-Age,
               names_to = "Year",
               values_to = "lower")
mx_fcst95l <-
  mx.fcst95l %>%
  as_tibble(rownames = "Age") %>%
  pivot_longer(-Age,
               names_to = "Year",
               values_to = "upper")

# bind the pieces together into single object
fcst <- mx_fcst %>%
  left_join(mx_fcst95u, by = c("Age", "Year")) %>%
  left_join(mx_fcst95l, by = c("Age", "Year")) %>%
  pivot_longer(forecast:upper,
               names_to = "variant",
               values_to = "M") %>%
  bind_rows(mx_fit) %>%
  mutate(Age = as.integer(Age),
         Year = as.integer(Year))
```

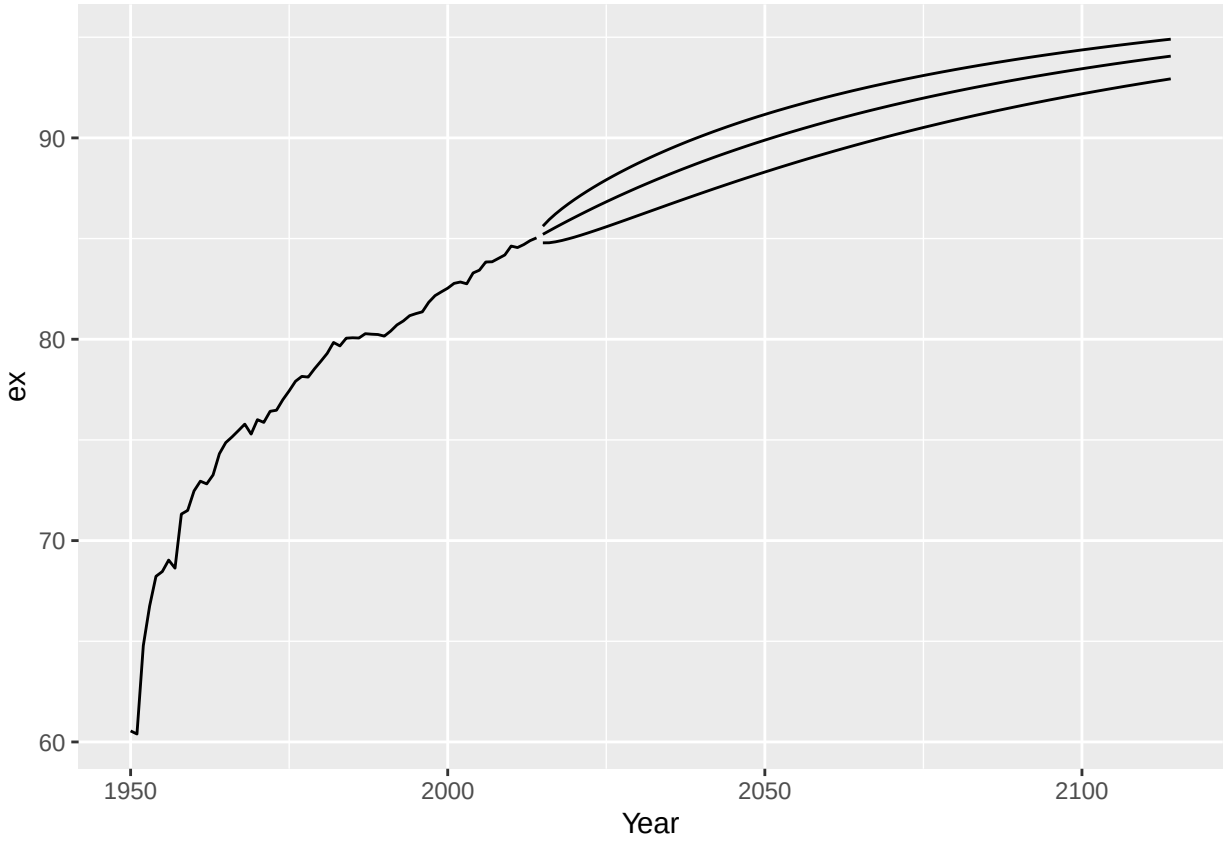
Step 7: Calculate the life table

```
LTfcst <-
  fcst %>%
  group_by(Year, variant) %>%
  mutate(M = ifelse(is.na(M), .5, M),          # hack
         n = 1,
         ax = case_when(
                Age == 0 & M < .02012 ~ .14916 - 2.02536 * M,
                Age == 0 & M < .07599 ~ 0.037495 + 3.57055 * M,
                Age == 0 & M >= .07599 ~ 0.30663,
                Age == 110 ~ 1 / M,
                TRUE ~ n / 2),
          ax = ifelse(is.infinite(ax),.5,ax),
          qx = (M * n) / (1 + (n - ax) * M),
          qx = ifelse(qx > 1, 1, qx),
          px = 1 - qx,
          lx = c(1, cumprod(px[-n()])),
          dx = qx * lx,
          Lx = lx - (n - ax) * dx,
          Tx = Lx %>% rev() %>% cumsum() %>% rev(),
          ex = Tx / lx)
```

Visualize results

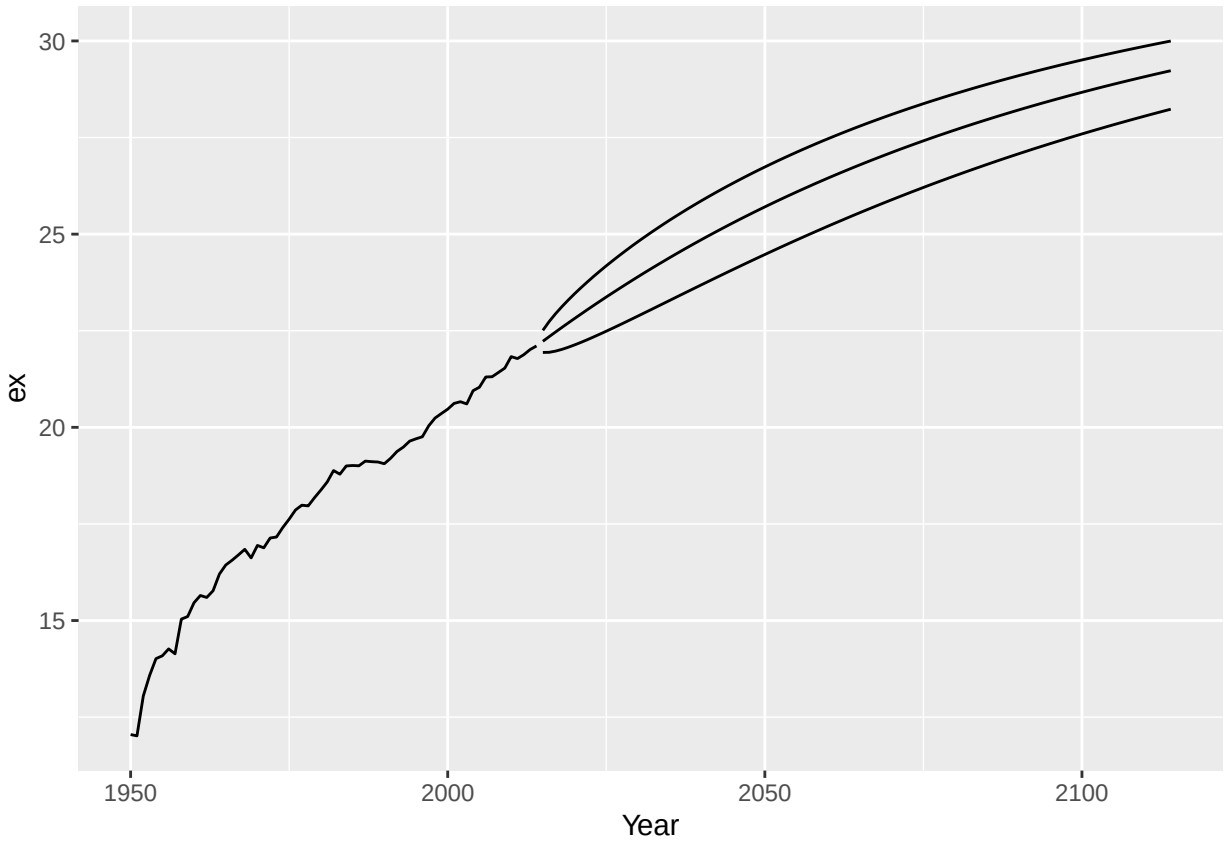
Life expectancy at birth

```
LTfcst %>%
  filter(Age == 0) %>%
  ggplot(aes(x = Year, y = ex, group = variant)) +
  geom_line()
```



Life expectancy at age 65

```
LTfcst %>%
  filter(Age == 65) %>%
  ggplot(aes(x = Year, y = ex, group = variant)) +
  geom_line()
```


30
25
20
15
ex
1950
2000
2050
2100
Year