Class 7: Lee Carter

Yubo Rasmussen

## **Lab Overview**

In this lab, we will explore the **Lee-Carter mortality model**, a widely used method for modeling and forecasting mortality rates.

## **1. Load Data and Define Variables**

# Read HMD data  
source("EngWales\_read.r")  
  
# Filter data for ages 40-90 and years 1950-2000  
Dth <- Dth.M[ Age >= 40 & Age <= 90 , Year >= 1950 & Year <= 2000 ]  
Exp <- Exp.M[ Age >= 40 & Age <= 90 , Year >= 1950 & Year <= 2000 ]  
Obs <- log( Dth / Exp )  
  
# Convert to vectors  
Dth.V <- c(Dth)  
Exp.V <- c(Exp)  
  
# Define age and year as factors  
AGE <- 40:90  
YEAR <- 1950:2000  
Age.F <- factor(rep(AGE , ncol(Dth)))  
Year.F <- factor(rep(YEAR , each = nrow(Dth)))  
n.x = nrow(Dth)  
n.y = ncol(Dth)

## **2. Fit the Lee-Carter Model**

# Fit Lee-Carter model using Poisson regression  
LC.Model <- gnm( Dth.V ~ -1 + Age.F + Mult(Age.F, Year.F),   
 offset = log(Exp.V), family = poisson)

Initialising  
Running start-up iterations..  
Running main iterations.....  
Done

# Extract coefficients  
Alpha.gnm <- LC.Model$coefficients[1:51]   
Beta.gnm <- LC.Model$coefficients[52:102]   
Kappa.gnm <- LC.Model$coefficients[103:153]

**💡 Question 2:** Why do we use Poisson regression in this model? What does offset=log(Exp.V) achieve?

## **3. Adjust for Identifiability Constraints**

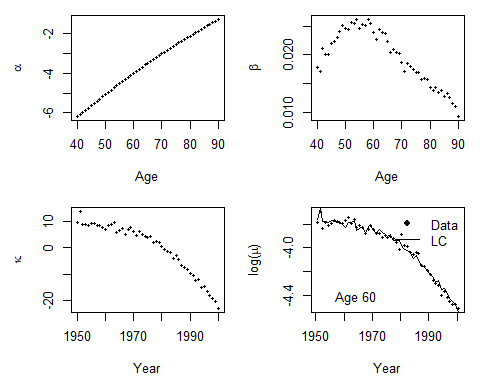
# Apply constraints to ensure identifiability  
Kappa.m <- mean(Kappa.gnm)   
Beta.m <- mean(Beta.gnm)   
Alpha.hat <- Alpha.gnm + Kappa.m \* Beta.gnm   
Beta.hat <- Beta.gnm / (nrow(Dth) \* Beta.m)   
Kappa.hat <- nrow(Dth) \* Beta.m \* (Kappa.gnm - Kappa.m)

## **4. Compute Fitted Mortality Rates**

# Compute fitted mortality estimates  
Fitted.M.hat = Alpha.hat + Beta.hat %\*% t(Kappa.hat)

## **5. Visualizing the Model Parameters**

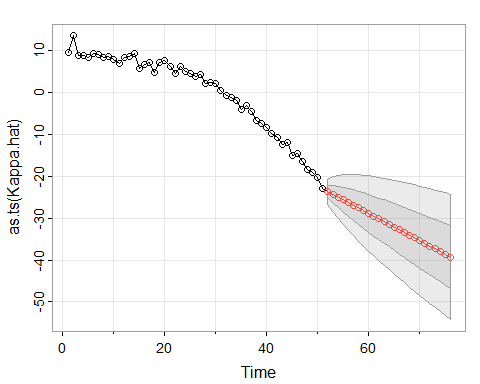
# Define function to plot Lee-Carter components  
PlotLeeCarter <- function(AGE, YEAR, Obs, Alpha.hat, Beta.hat, Kappa.hat, Fitted.M.hat) {  
 par(mfrow = c(2,2), mar = c(4.5,4.5,1,1))  
  
 # Plot alpha (age effects)  
 plot(AGE, Alpha.hat, xlab = "Age", ylab = expression(alpha), cex = 0.5, pch = 16)  
   
 # Plot beta (age-specific sensitivity)  
 plot(AGE, Beta.hat, xlab = "Age", ylab = expression(beta), cex = 0.5, pch = 16)  
   
 # Plot kappa (time trends)  
 plot(YEAR, Kappa.hat, xlab = "Year", ylab = expression(kappa), cex = 0.5, pch = 16)  
   
 # Plot fitted vs observed mortality at age 60  
 Age.Plot = 60  
 Row = Age.Plot - min(AGE) + 1  
 plot(YEAR + 0.5, Obs[Row, ], xlab = "Year", ylab = expression(paste("log(", mu, ")")), cex = 0.5, pch = 16)  
 lines(YEAR + 0.5, Fitted.M.hat[Row, ], type = "l")  
 legend("topright", legend = c("Data", "LC"), pch = c(16, -1), lty = c(-1, 1), bty = "n")  
 legend("bottomleft", legend = paste("Age", Age.Plot), bty = "n")  
}  
  
# Generate the plots  
PlotLeeCarter(AGE, YEAR, Obs, Alpha.hat, Beta.hat, Kappa.hat, Fitted.M.hat)



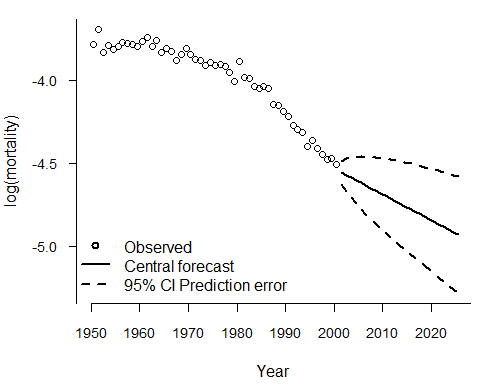
## **6. Forecasting Future Mortality Rates**

We now forecast mortality rates using a **random walk with drift model** for (\_t).

# Load astsa library for time series analysis  
library("astsa")  
  
# Forecast kappa using a random walk with drift  
N.Ahead = 25  
Kappa.for = sarima.for(as.ts(Kappa.hat), n.ahead = N.Ahead, p=0, d=1, q=0)



# Prediction error calculations  
Central = Kappa.for$pred  
SE.Pred = Kappa.for$se  
Z = qnorm(0.975)  
Kappa.Up.Pred = Central + Z\*SE.Pred  
Kappa.Dn.Pred = Central - Z\*SE.Pred  
Forecast = Alpha.hat + Beta.hat %\*% t(Central)  
Forecast.Up.Pred = Alpha.hat + Beta.hat %\*% t(Kappa.Up.Pred)  
Forecast.Dn.Pred = Alpha.hat + Beta.hat %\*% t(Kappa.Dn.Pred)  
Range = 2001:2025  
  
# Plot mortality forecasts  
Plot.Age = 60  
Plot.Row = Plot.Age - min(AGE) + 1  
par(mfrow = c(1,1))  
par(mar=c(4.2, 4, 1, 0.5), mgp=c(3, 1, 0), las=1, cex=1)  
Year.For = seq(YEAR[1], (YEAR[length(YEAR)]+25))  
plot(YEAR + 0.5, Obs[Plot.Row, ], axes = FALSE, xlab = "Year", ylab = "log(mortality)",  
 xlim = c(1950,2025),  
 ylim = c(min(Forecast.Dn.Pred[Plot.Row, ]), max(Obs[Plot.Row, ]) ))  
axis(1,las = 1, at = seq(1950, 2040, by = 10), tcl = -0.4)  
axis(2, seq(-6,-2, by = 0.5), tcl = -0.4)  
lines(Range + 0.5, Forecast[Plot.Row, ], lwd = 2)  
lines(Range + 0.5, Forecast.Up.Pred[Plot.Row, ], lty = 2, lwd = 2)  
lines(Range + 0.5, Forecast.Dn.Pred[Plot.Row, ], lty = 2, lwd = 2)  
legend("bottomleft", legend = c("Observed", "Central forecast", "95% CI Prediction error"),  
 lty = c(-1, 1, 2, 2), pch = c(1,-1,-1,-1), lwd = 2, bty = "n")



## **7. Accounting for Parameter Uncertainty**

We also calculate **parameter uncertainty** in our forecasts:

# Compute parameter error  
Sarima.out = sarima(Kappa.hat, p=0, d=1, q=0, details = FALSE)

<><><><><><><><><><><><><><>  
   
Coefficients:   
 Estimate SE t.value p.value  
constant -0.6518 0.2121 -3.0732 0.0035  
  
sigma^2 estimated as 2.249298 on 49 degrees of freedom   
   
AIC = 3.728495 AICc = 3.730162 BIC = 3.804976

SE.Param = (1:N.Ahead) \* sqrt(Sarima.out$fit$sigma2/(n.y -1))  
Kappa.Up.Param = Central + Z\*SE.Param  
Kappa.Dn.Param = Central - Z\*SE.Param  
Forecast.Up.Param = Alpha.hat + Beta.hat %\*% t(Kappa.Up.Param)  
Forecast.Dn.Param = Alpha.hat + Beta.hat %\*% t(Kappa.Dn.Param)  
  
# Plot with parameter error  
par(mar=c(4.2, 4, 1, 0.5), mgp=c(3, 1, 0), las=1, cex=1)  
plot(YEAR+0.5, Obs[Plot.Row, ], axes = FALSE, xlab = "Year", ylab = "log(mortality)",  
 xlim = c(1950, 2025),  
 ylim = c(min(Forecast.Dn.Param[Plot.Row, ]), max(Obs[Plot.Row, ]) ))  
axis(1,las = 1, at = seq(1950, 2040, by = 10), tcl = -0.4)  
axis(2, seq(-6,-3, by = 0.5), tcl = -0.4)  
lines(Range+0.5, Forecast[Plot.Row, ], lwd = 2)  
lines(Range+0.5, Forecast.Up.Param[Plot.Row, ], lty = 2, lwd = 2)  
lines(Range+0.5, Forecast.Dn.Param[Plot.Row, ], lty = 2, lwd = 2)  
legend("bottomleft", legend = c("Observed", "Central forecast", "95% CI Parameter error"),  
 lty = c(-1, 1, 2, 2), pch = c(1,-1,-1,-1), lwd = 2, bty = "n")

