European Doctoral School of Demography 2019-20

Population Projections

Lecture 4

Ugofilippo Basellini

basellini@demogr.mpg.de

MAX-PLANCK-INSTITUT FÜR DEMOGRAFISCHE FORSCHUNG
MAX PLANCK INSTITUTE FOR DEMOGRAPHIC RESEARCH

ined
INSTITUT NATIONAL D'ÉTUDES DÉMOGRAPHIQUES

Online course
4$^{\text{th}}$ June 2020

Brief course summary

Lecture 1: introduction & first simple model of population projections

Lecture 2: cohort component method

Lecture 3: matrix projections & dynamic visualizations

Lecture 4: extensions of matrix projections

- including migration
- time-specific assumptions
- stable population

Small recap

In Lecture 3, we have seen the advantages of using the matrix formulation of the cohort component method for population projections

- can significantly speed up your code and computations
- can incorporate female and male populations within the same setting

$\Rightarrow$ what about migration and time-specific assumptions on future demographic components?

Projection of an open population I

- Formal difficulty of integrating migration is that migration continuously affects the population at risk of dying and giving birth
- Similarly to what we have seen in Lecture 1, we can incorporate net migration using counts rather than rates
- Convenient approach: divide number of migrants into two discrete quantities, and assume that half move at the start of the interval, and half at the end of it
- Let $I_x[t, t+5]$ be the net flow if immigrants (it can be negative if migration balance is negative). Then:

$$N_{x+5}(t+5) = \left(N_x(t) + \frac{I_x[t, t+5]}{2} \right) s_x + \frac{I_{x+5}[t, t+5]}{2}$$

Projection of an open population II

- Matrix notation can once again simplify calculations

$$\boldsymbol{N}(t+5) = \boldsymbol{L}[t, t+5]\left(\boldsymbol{N}(t) + \frac{1}{2}\boldsymbol{I}(t)\right) + \frac{1}{2}\boldsymbol{I}(t)$$

- As discussed in Lecture 1, these adjustments are presentationally convenient but rather disconcerting: emigration and immigration need not have the same causes and constraints
- Better approach: treat immigration as presented here, while emigration can be better modelled using a multiple-decrement life table (combined with mortality)

Derivation of net migration counts

- Can be computed from the balancing equation of population growth (using data on population size, births and deaths)
- these tend to show erratic year-to-year fluctuations
- can use a parametric model instead, such as the Rogers and Castro (1981) model migration schedules
  - only need the total number of net migrants, which is distributed over the age groups using parametric functions
  - implemented in `R` in the `migest` package (Abel 2019)

Rogers and Castro (1981) migration schedule

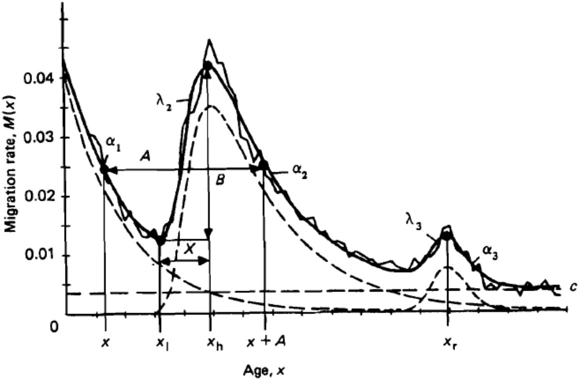


FIGURE 4 The model migration schedule.

Four components: (i) a negative exponential curve of the *pre-labor force* ages; (ii) a left-skewed unimodal curve of the *labor force* ages; (iii) an almost bell-shaped curve of the *post-labor force* ages; and (iv) a constant curve.

The `migest` package

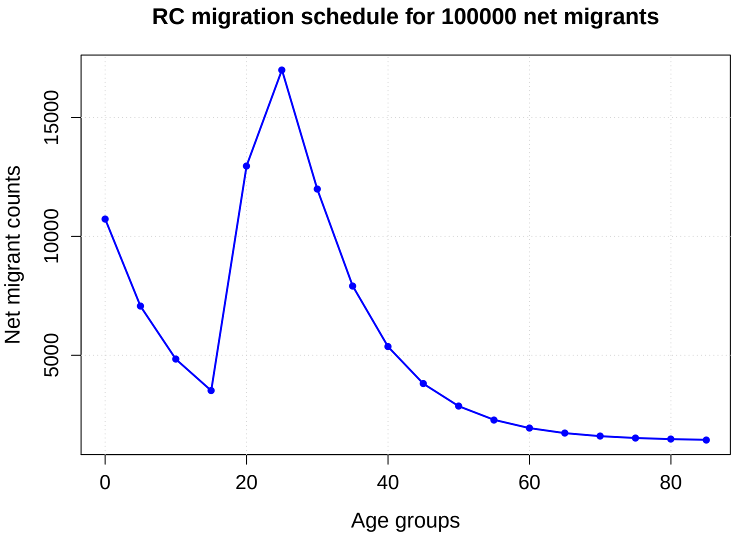
- comprehensive `R` package for estimating bilateral migration flows in the presence of partial or missing data
- the `rc9` function allows one to compute the RC migration schedule for a given set of parameters
- the fundamental RC parameters are provided by the `rc9.fund` vector
- obtain age-specific counts by multiplying total net migration by the RC schedule

The RC migration schedule in R

Example

```
rm(list = ls())
library(migest)
## check the fundamental RC parameters
rc9.fund
## five-year age groups (works well also with one-year)
x <- seq(0,85,5)
mx <- rc9(x, param = rc9.fund)
plot(x, mx, type="o",pch=16)
## assume a total of 100000 net migration counts
I <- 1e5
Ix <- I*mx
sum(Ix)
plot(x, Ix, type="o",pch=16,
    xlab = "Age group",ylab= "Net migrant counts",
    main="RC migration schedule for 100000 net migrants")
```

The RC migration schedule in R



Including net migration - exercise

Exercise

Let's start from the data that we saved yesterday, `EDSD.lecture3.Rdata`. Project the female population for 20 periods ahead assuming an overall amount of 25000 net migrants. Compare your projections with and without the migration component.

Hint: adapt your population projection function by including the net migration counts (by age, derived from the RC schedule).

Reminder:

$$\boldsymbol{N}(t+5) = \boldsymbol{L}[t, t+5]\left(\boldsymbol{N}(t) + \frac{1}{2}\boldsymbol{I}(t)\right) + \frac{1}{2}\boldsymbol{I}(t)$$

Including net migration - one possible solution I

Example

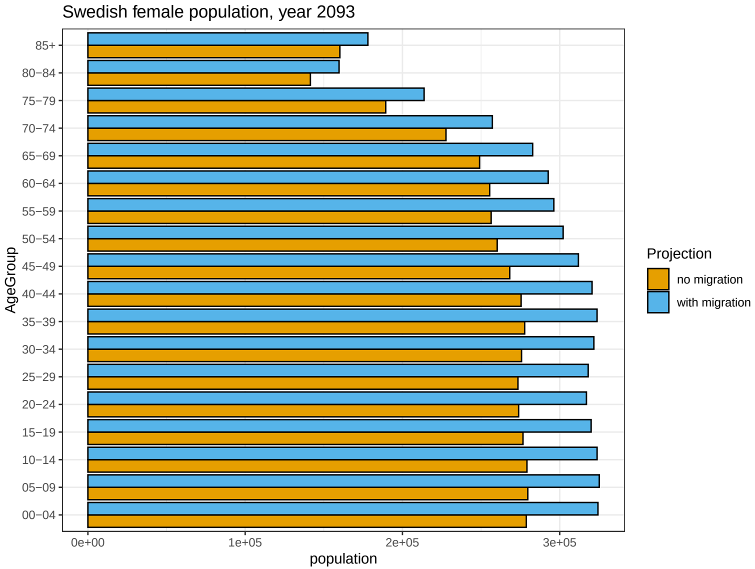
```
rm(list = ls())
library("tidyverse")
library("migest")
load("EDSD.lecture3.Rdata")
## derive net female migrants by age (RC schedule)
I <- 2.5e4
Ix <- I*rc9(dta.swe$Age, param = rc9.fund)
## project female population for several periods WITH net migration counts
pop.proj.v3 <- function(x,AgeGroup,Nx,sFx,bFx,Ix,n){
    ## number of age groups
    m <- length(x)
    ## create Leslie matrix
    L <- matrix(0,m,m)
    L[1,] <- bFx
    diag(L[-1,]) <- sFx
    L[m,m] <- sFx[m-1]
    ## create population matrix
    N <- matrix(0,m,n+1)
    N[,1] <- Nx
    for (i in 1:n){
        N[,i+1] <- L%*%(N[,i] + Ix/2) + Ix/2
    }
    out <- cbind(data.frame(x=x,AgeGroup=AgeGroup),N)
    return(out)
}
```

Including net migration - one possible solution II

Example

```
## actual projection
n <- 20
my.proj.base <- pop.proj(x=dta.swe$Age,AgeGroup=dta.swe$AgeGroup,Nx=NFx,sx=sFx,bx=bFx,n=n)
my.proj.migr <- pop.proj.v3(x=dta.swe$Age,AgeGroup=dta.swe$AgeGroup,Nx=NFx,sFx=sFx,bFx=bFx,Ix=Ix,n=n)

## long data
dta.swe.l <- my.proj.base %>%
    pivot_longer(-c(x,AgeGroup),names_to = "period",values_to = "population") %>%
    mutate(period=as.numeric(period),
        Year=1993 + (period-1)*5,
        YearF=as.factor(Year),
        type = "no migration")
dta.swe.l.mig <- my.proj.migr %>%
    pivot_longer(-c(x,AgeGroup),names_to = "period",values_to = "population") %>%
    mutate(period=as.numeric(period),
        Year=1993 + (period-1)*5,
        YearF=as.factor(Year),
        type = "with migration")
dta.swe.all <- dta.swe.l %>%
    bind_rows(dta.swe.l.mig)

## plotting
ggplot(dta.swe.all,aes(x=AgeGroup,y=population,fill=type)) +
    geom_bar(data = subset(dta.swe.all, period == 21),
        stat = "identity",position = "dodge",color = "black") +
    coord_flip() +
    theme_bw() +
    ggtitle(paste("Swedish female population, year",subset(dta.swe.all, period == 21)$Year)) +
    scale_fill_manual(name = 'Projection', values=c("#E69F00", "#56B4E9","#1C7C54"))
```

Including net migration - one possible solution III



Time-varying assumptions in the cohort component method

- we might want to increase the flexibility of our methodology to incorporate time-varying assumptions on some (or all) demographic components
- this extension could be used to provide a more “realistic” scenario of population projections
- this can be achieved by including time-specific entries in our Leslie matrix

Time-varying assumptions - exercise

Exercise

Let's now project the female population for 20 periods ahead, assuming that the total fertility rate decreases from its current value (2.085) to 0.750 at the end of your projections. Compare your results with and without this assumption.

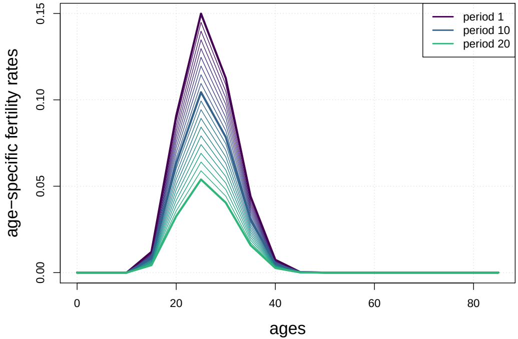
Hint: remember that $TFR = 5 \sum_x Fx$, and from Lecture 2

$$b_x^F = \frac{1}{1 + SRB} \frac{L_0^F}{2\ell_0} \left(F_x + s_x^F \, F_{x+5} \right)$$

Time-varying assumptions

Example

```
## adjusting the TFR
m <- nrow(dta.swe)
Fx <- dta.swe$Fx
FX <- matrix(c(Fx),nrow=m,ncol = n)
tfr1 <- 5*sum(Fx)
TFR <- seq(tfr1,0.75,length.out = n)
for (i in 1:n){
FX[,i] <- FX[,i] * TFR[i] / tfr1
}
5*apply(FX,2,sum)
```



Time-varying assumptions

Example

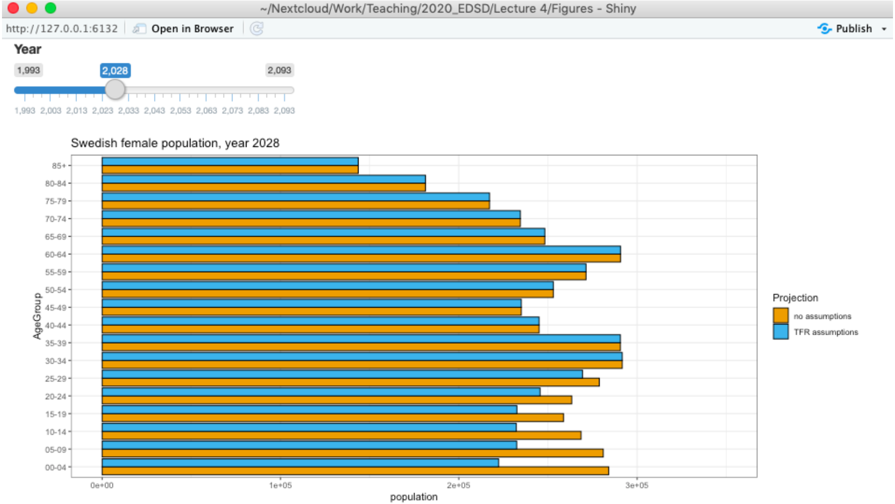
```
## function to project several periods with assumptions on TFR
pop.proj.TFR <- function(x,AgeGroup,Nx,sFx,FX,LOF,l0=1e5,srb=1.05,n){
    ## number of age groups
    m <- length(x)
    ## factors for bx calculation
    srb.fac <- 1/(1+srb)
    suv.fac <- LOF / (2*l0)
    ## create Leslie matrix
    L <- matrix(0,m,m)
    diag(L[-1,]) <- sFx
    L[m,m] <- sFx[m-1]
    ## create population matrix
    N <- matrix(0,m,n+1)
    N[,1] <- Nx
    for (i in 1:n){
        ## update bx and Leslie matrix at each iteration
        bFx <- srb.fac*suv.fac*(FX[-m,i] + sx[-m]*FX[-1,i])
        L[1,1:length(bx)] <- bFx
        ## project population
        N[,i+1] <- L%*%N[,i]
    }
    out <- cbind(data.frame(x=x,AgeGroup=AgeGroup),N)
    return(out)
}
```

Time-varying assumptions

Example

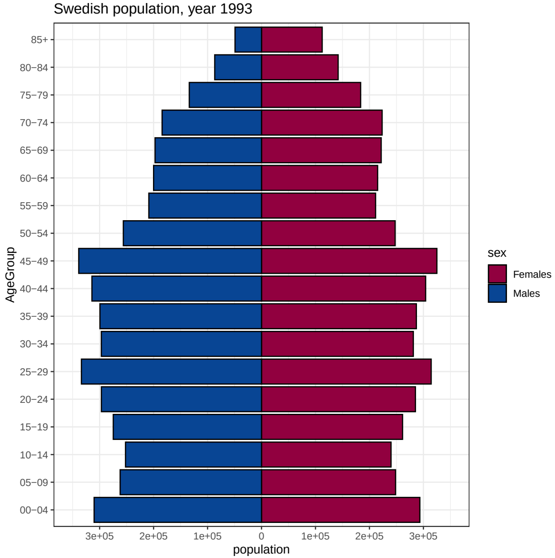
```
my.proj.TFR <- pop.proj.TFR(x=dta.swe$Age,AgeGroup=dta.swe$AgeGroup,
    FX=FX,Nx=NFx,sFx=sFx,LOF = dta.swe$LFx[1],n=n)

## long format for your result, using the same code as for migration ...

## SHINY APP for dynamic visualization of your results
library(shiny)
ui <- fluidPage(
    sliderInput(inputId = "year", label = "Year", step = 5,
        value = min(dta.swe.l$Year), min = min(dta.swe.l$Year), max = max(dta.swe.l$Year)),
        column(12, plotOutput("plot_pyr1"))
)

server <- function(input, output){
    output$plot_pyr1 <- renderPlot({
        ## plotting pyramid
        ggplot(dta.swe.all,aes(x=AgeGroup,y=population,fill=type)) +
        geom_bar(data = subset(dta.swe.all, Year == input$year),
            stat = "identity",position = "dodge",color = "black") +
        coord_flip() +
        theme_bw() +
        ggtitle(paste("Swedish female population, year",subset(dta.swe.all, Year == input$year)$Year)) +
        scale_fill_manual(name = 'Projection', values=c("#E69F00", "#56B4E9","#1C7C54")) +
        scale_y_continuous(limits = c(0, 350000), breaks = seq(0, 350000, 100000))
})
}

shinyApp(ui = ui, server =  server)
```

Time-varying assumptions



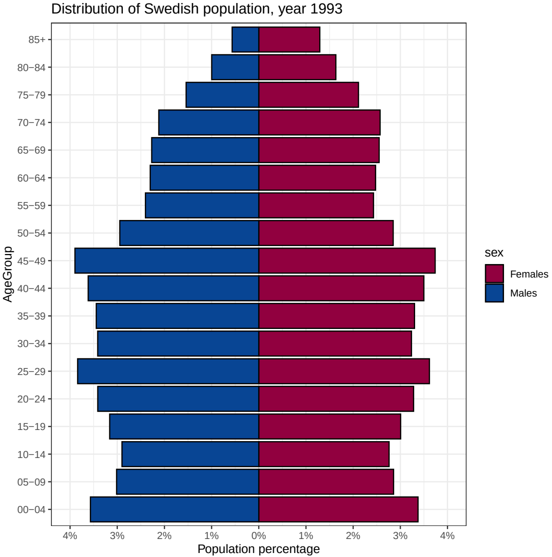
Longer projections

Two-sex closed population, constant mortality and fertility, longer projections



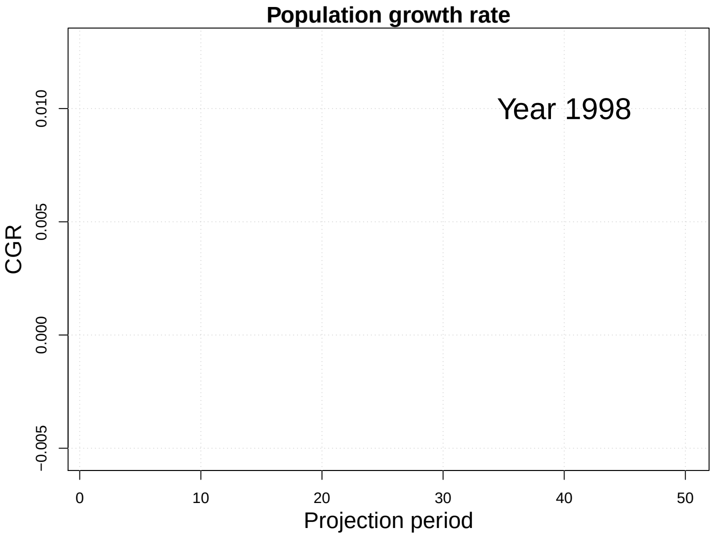
Longer projections

Let's look at the age-distribution of the population (in % terms)



Longer projections

And at the growth rate of the population



Matrix projections & stable populations

A remarkable finding about the Leslie matrix $\boldsymbol{L}$:

- under some assumptions (nearly always satisfied by human populations), when $\boldsymbol{L}$ is raised to a high enough power n, the *distribution* of the population age structure of $\boldsymbol{N}(t + 5 \times n)$ and the *crude growth rate* during each projection become **constant**
- result related to the stable population theorem

In the stable state, the population should satisfy:

$$\boldsymbol{N}^{\boldsymbol{S}}(t+5) = \boldsymbol{L}\boldsymbol{N}^{\boldsymbol{S}}(t) = \lambda \boldsymbol{N}^{\boldsymbol{S}}(t) \quad (1)$$

where λ is the crude growth rate of the population

$\Rightarrow$ eigendecomposition of $\boldsymbol{L}$

Matrix projections & stable populations

Re-arranging Equation (1)

$$\left(\boldsymbol{L} - \lambda \mathbf{I}\right) \boldsymbol{N}^{\boldsymbol{S}}(t) = 0 \tag{2}$$

where $\mathbf{I}$ is the identity matrix.

As such, the eigendecomposition of the matrix $\boldsymbol{L}$ will directly provide us with:

- the crude growth rate in the stable state, λ, which is given by the (log of the) largest real eigenvalue of $\boldsymbol{L}$
- the stable population distribution, which is given by the corresponding (rescaled) eigenvector
- refer to Caswell (2001) for everything you need to know about matrix population models

Matrix projections & stable populations - example

Example

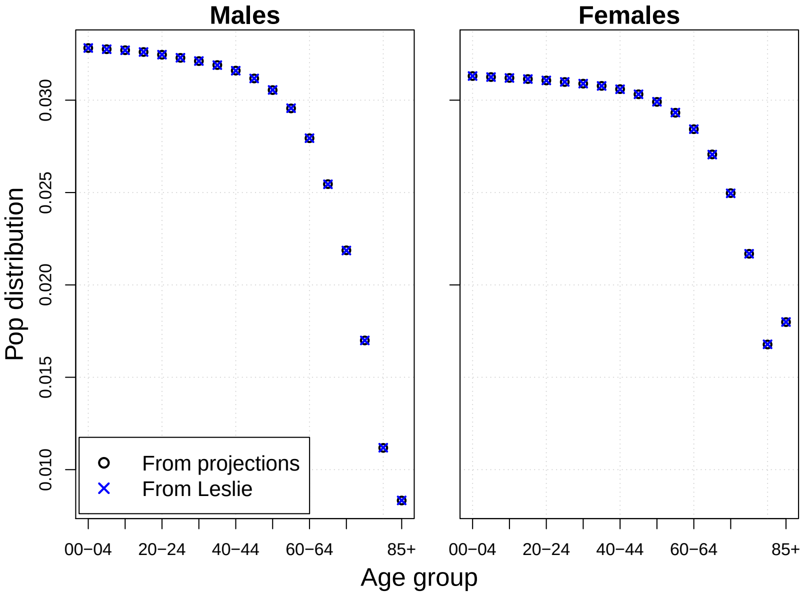
```
## longer time horizon
n <- 50
my.proj <- pop.proj.v2(x=dta.swe$Age,AgeGroup=dta.swe$AgeGroup,NFx=NFx,sFx=sFx,bFx=bFx,
    NMx=NMx,sMx=sMx,bMx=bMx,n=n)

## long data
dta.swe.l <- my.proj %>%
    pivot_longer(-c(x,AgeGroup,sex),names_to = "period",values_to = "population") %>%
    mutate(period=as.numeric(period),
        Year=1993 + (period-1)*5,
        YearF=as.factor(Year))

## compute total population in each year
tot.pop <- dta.swe.l %>%
    group_by(YearF) %>%
    summarise(TotPop=sum(population))

## compute distribution of the population in each age group
dta.swe.l <- dta.swe.l %>%
    left_join(tot.pop) %>%
    mutate(Pch=population/TotPop)

## extract distribution in last year
x <- dta.swe$Age
yF <- dta.swe.l$Pch[dta.swe.l$Year == max(dta.swe.l$Year) & dta.swe.l$sex == "Females"]
yM <- dta.swe.l$Pch[dta.swe.l$Year == max(dta.swe.l$Year) & dta.swe.l$sex == "Males"]
```

Matrix projections & stable populations - example

Example

```
## eigendecomposition of the Leslie matrix
ev.decomp <- eigen(L)

## extract largest eigenvalue
## this will give you the long-term CGR of the stable population
ev.val <- abs(ev.decomp$values)[1]
cgr_leslie <- log(ev.val)

## extract corresponding eigenvectors
ev.vec <- ev.decomp$vectors[,1]
ev.vecF <- ev.vec[1:m]
ev.vecM <- ev.vec[1:m + m]

## rescale them to sum to the projected total long-term distribution
## this will give you the long-term distribution of the stable population
ev.vecF <- (ev.vecF/sum(ev.vecF)) * sum(yF)
ev.vecM <- (ev.vecM/sum(ev.vecM)) * sum(yM)
```

Matrix projections & stable populations - example

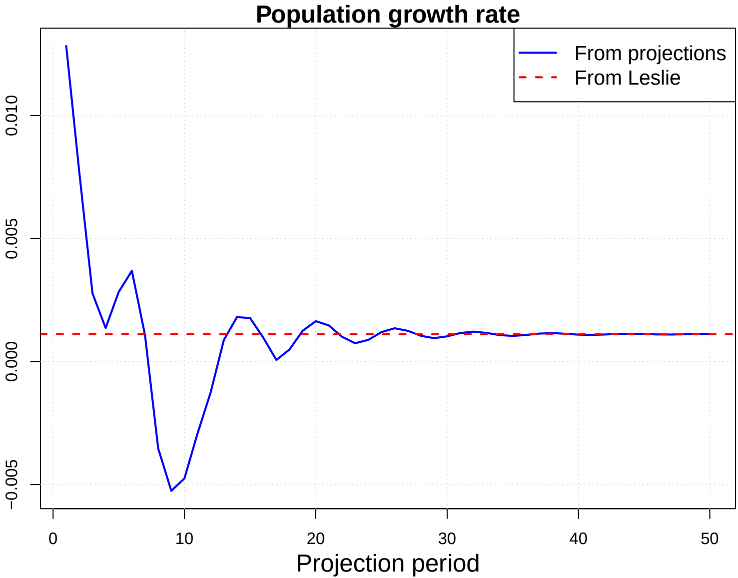


Matrix projections & stable populations - example

Example

```
## extract total population
tot.pop <- tot.pop %>%
    pull(TotPop)

## compute growth rate
CGR <- rep(NA,n)
for (i in 1:n){
    CGR[i] <- log(tot.pop[i+1]/tot.pop[i])
}
```

Matrix projections & stable populations - example



Some final remarks

- we have seen several extensions of the cohort component model in matrix formulation
- age interval can be of any length (1y, 5y, 10y), formulas are unchanged (but the projection period is always equal to your age interval)
- results shown from the eigendecomposition of the Leslie matrix hold only:
  - in the long run, *and*
  - for a stable population (closed to migration, with constant fertility and mortality rates over time)

Assignment

Exercise #6

Following up on Exercise #5, include migration in your projection 20 period ahead. You can either use observed net migration counts (you can estimate them from the balancing equation of population growth) or the Rogers-Castro migration schedule (try to find reasonable data on total net migration counts).

Exercise #7

Project your chosen population by sex for $n = 20$ periods ahead, making time-specific assumptions about one or more demographic components.

Assignment

Exercise #8

Derive the long-term growth rate and population distribution by eigendecomposing your Leslie matrix, and compare your results with those that you obtain from a long projection of your population. As a second step, modify your starting population by randomly re-assigning elements on $\boldsymbol{N}(t)$ to different age groups, and project the new population in the long term. Compute the long-term growth rate and distribution of this second population, and compare it to the first

Bonus for all exercises: present your results with the aid of a shiny app or an animation.

References

- Abel, G.J. (2019). `migest`: Methods for the Indirect Estimation of Bilateral Migration. R package version 1.8.1. https://CRAN.R-project.org/package=migest
- Caswell, H. (2001). *Matrix Population Models*, Second Edition. Sinauer Associates, Inc. Publishers
- Preston, S.H., Heuveline, P., and Guillot, M. (2001). *Demography. Measuring and Modeling Population Processes*. Blackwell.
- Rogers, A. and Castro, L.J. (1981). Model Migration Schedules. *IIASA Research Report 81*, RR-81-30