Barcelona Summer School of Demography

*Module 2. Demography with R*

# 4. Population growth

Population growth

Tim Riffe
tim.riffe@gmail.com

10 July 2025

## Contents

# 1 Summary

Today, we're of course exploiting Marie-Piers labor once again. As usual I've taken steps to *tidify* parts of the lesson, but this isn't possible for things that are iterative or involve matrix algebra. Note the concepts discussed here are often misinterpreted and misused for nationalist and racist agendas. We're smarter than that and understand that these are just occasionally useful models. These are conceptual models, not forecasts, and many simplifying assumptions are applied when modeling population growth.

# 2 Matrix algebra: a brief review

Matrix multiplication is not the same as arithmetic multiplication!

```
#Review of matrix algebra

#matrix multiplication

# Only possible if
#the number of columns on the left matrix =
#the number of rows on the right matrix

#Example 1
x <- matrix(c(1:9), ncol=3)
x
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```r
y <- matrix(c(1:6), ncol=2)
y
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```r
A <- x %*% y
A
```

```
##      [,1] [,2]
## [1,]   30   66
## [2,]   36   81
## [3,]   42   96
```

```r
A[1, 1]
```

```
## [1] 30
```

```r
sum(x[1, ] * y[, 1])
```

```
## [1] 30
```

```r
#Example 2
x <- matrix(c(1, 2, 3), ncol = 1)
x
```

```
##      [,1]
## [1,]    1
## [2,]    2
## [3,]    3
```

```r
y <- matrix(c(2, 4, 6), nrow = 1)
y
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
```

```r
A <- x %*% y
A
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    4    8   12
## [3,]    6   12   18
```

Matrix addition and subtraction are element-wise. For this, dimensions should conform.

```r
#matrix addition

B<-matrix(c(1:9), ncol=3)
C<-matrix(c(2:10), ncol=3)

D<-A+B
D[1,1]
```

```
## [1] 3
```

```r
A[1,1]+B[1,1]
```

```
## [1] 3
```

Matrix powers are not the same as arithmetic powers!

```r
# install.packages("expm")
library(expm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```r
A %^% 3
```

```
##      [,1] [,2]  [,3]
## [1,] 1568 3136  4704
## [2,] 3136 6272  9408
## [3,] 4704 9408 14112
```

```r
# arithmetic powers are element-wise
A ^ 3
```

```
##      [,1] [,2] [,3]
## [1,]    8   64  216
## [2,]   64  512 1728
## [3,]  216 1728 5832
```

## 3 Data

We will use the Spanish data from the HMD (Human Mortality Database 2018) and HFD (Human Fertility Database 2018) for sections 2 and 3 of this class. Please load the following (this has been pasted in the google doc as well!):

```r
#setwd()
library(tidyverse)
library(readr)
library(janitor)
source("https://raw.githubusercontent.com/timriffe/BSSD2025Module2/master/02_lifetables.R")
B <- read_csv("https://raw.githubusercontent.com/timriffe/BSSD2025Module2/master/data/ES_B20
  mutate(sex = "total") |>
  select(age, sex, births = total)

D<- read_csv("https://raw.githubusercontent.com/timriffe/BSSD2025Module2/master/data/ES_D201
     filter(year == 2014) |>
  select(-open_interval) |>
  pivot_longer(female:total,
               names_to = "sex",
               values_to = "deaths")

P <- read_csv("https://raw.githubusercontent.com/timriffe/BSSD2025Module2/master/data/ES_P.c
  clean_names() |>
  select(-open_interval) |>
  filter(year== 2014) |>
  pivot_longer(female1:total2,
               names_to = "sex",
               values_to = "pop") |>
  mutate(period = parse_number(sex),
         sex = gsub('[[:digit:]]+', '', sex)) |>
  pivot_wider(names_from = period, values_from = pop, names_prefix="pop")

ES2014 <-
  left_join(P, D, by = join_by(year,sex, age)) |>
  left_join(B, by = join_by(sex, age))

LT <-
  ES2014 |>
```

```
mutate(exposure = (pop1 + pop2) / 2,
       mx = deaths / exposure) |>
select(sex,age,mx) |>
group_by(sex) |>
group_modify(~lt_full(data = .x, group = .y)) |>
ungroup()
```

# 4 Crude growth

## 4.1 Balancing equation

Changes in population size is affected by changes in the number of births, deaths and migration.

- Two ways on entry: Birth and Immigration
- Two ways of exit: Death and Emigration

As there are only two ways of entry and two ways of exit, changes in population size come from changes in the magnitude of these flows:

$$N(T) = N(0) + B[0,T] - D[0,T] + I[0,T] - E[0,T]$$

where

- $N(T)$ and $N(0)$ are the number of persons alive at time $T$ and 0, respectively
- $B[0,t]$ is the number of births between times 0 and T
- $D[0,t]$ is the number of deaths between times 0 and T
- $I[0,t]$ is the number of immigrations between times 0 and T
- $E[0,t]$ is the number of emigrations between times 0 and T

## 4.2 Crude growth rate

The crude growth rate is the change in population size relative to the person-year in a given time interval (e.g. from year 0 to T).

Given the balancing equation, the crude growth rate (CGR) of a population is:

$$\frac{N(T)-N(0)}{PY[0,T]} = \frac{B[0,T]}{PY[0,T]} - \frac{D[0,T]}{PY[0,T]} + \frac{I[0,T]}{PY[0,T]} - \frac{E[0,T]}{PY[0,T]}$$

$$CGR[0,T] = \underbrace{CBR[0,T] - CDR[0,T]}_{CRNI[0,T]} + \underbrace{CIR[0,T] - CER[0,T]}_{CRNM[0,T]}$$

- $CBR[0,T]$ is the crude birth rate
- $CDR[0,T]$ is the crude death rate
- $CIR[0,T]$ is the crude immigration rate

- $CER[0,T]$ is the crude emigration rate
- $CRNI[0,T]$ is the crude rate of natural increase
- $CRNM[0,T]$ crude rate of net migration

We don't have information here on in and out migration, but we can calculate the other crude rates.

```
Components <-
  ES2014 %>%
  filter(sex == "total") %>%
  summarize(P1 = sum(pop1),
            P2 = sum(pop2),
            B = sum(births, na.rm = TRUE),
            D  = sum(deaths),
            ) %>%
   mutate(PY = (P1 + P2) / 2,
         NMig = P2 - (P1 + B - D))

CrudeRates <-
Components %>%
  mutate(CGR = (P2 - P1) / PY,
         CBR = B / PY,
         CDR = D / PY,
         CRNI = CBR - CDR,
         CRNM = NMig / PY,
         .keep = "none")

CrudeRates
```

```
## # A tibble: 1 x 5
##        CGR     CBR     CDR     CRNI     CRNM
##      <dbl>   <dbl>   <dbl>    <dbl>    <dbl>
## 1 -0.00144 0.00917 0.00848 0.000696 -0.00213
```

The CGR suggests a decline of Spain's population size between 2014 and 2015. However, Spain has recorded a positive CRNI. A possible explanation is that there was more out-migration than in-migration in Spain in 2014, reducing the population size and opposing the natural increase.

## 4.3 Geometric growth

Different growth rates can be assumed in a population over time. If we consider a geometric growth over time, the annual rate of change (r) in a population at time 0 and t is:

$$r = \frac{N(T) - N(0)}{N(0)} \frac{1}{T}$$

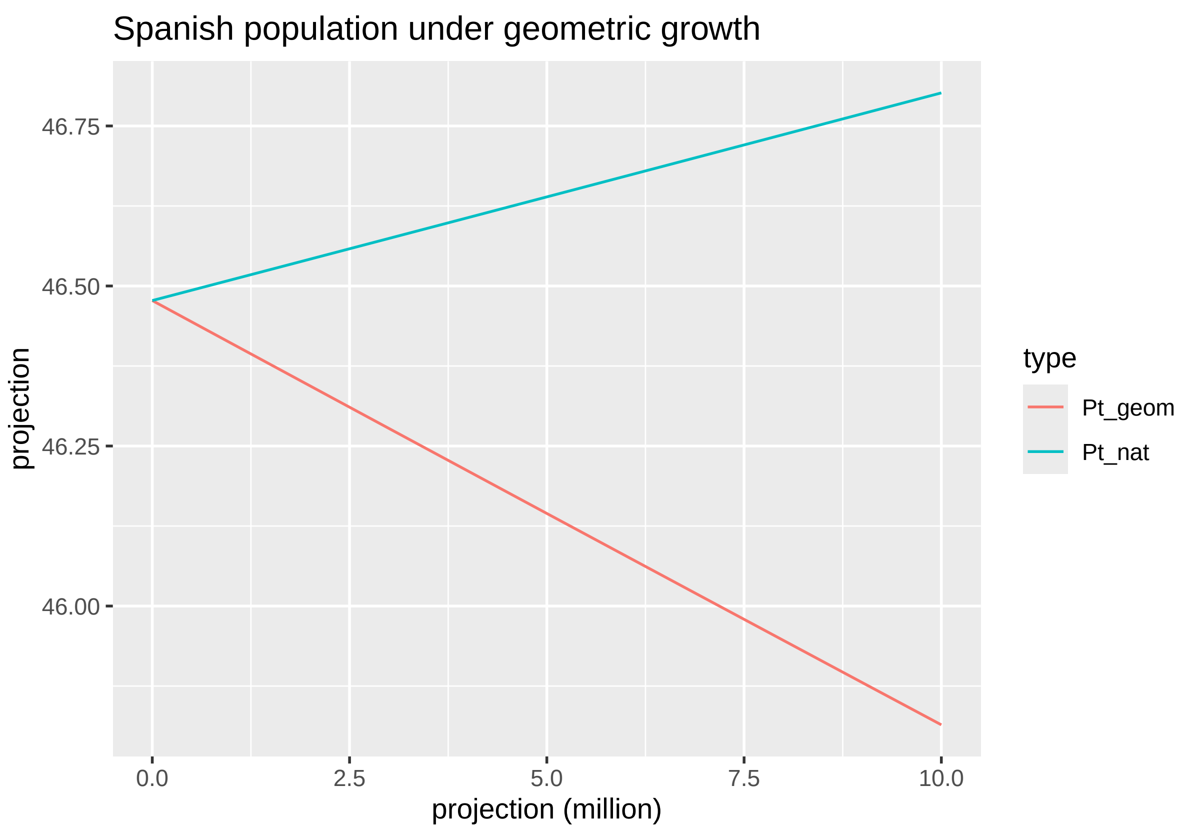
Thus, if $r$ is constant over time the population size at time t can be found by

$$N(t) = N(0)[1 + r]^t$$

```r
geometric_growth <-
  Components %>%
  mutate(r_geom = (P2 - P1) / P1,    # same as CBR here
         r_nat = (B - D) / P1) %>%  # same as CRNI here
  cross_join(tibble(t = 0:10)) %>%
  mutate(Pt_geom = P1 * (1 + r_geom) ^ t,
         Pt_nat = P1 * (1 + r_nat) ^ t)



geometric_growth %>%
  select(t, Pt_geom, Pt_nat) %>%
  pivot_longer(-t, names_to = "type", values_to = "projection") %>%
  mutate(projection = projection / 1e6) %>%
  ggplot(aes(x = t, y = projection, color = type)) +
  geom_line() +
  xlab("projection (million)") +
    labs(title = "Spanish population under geometric growth")
```




## 4.4 Exponential growth

Exponential growth can simply be seen as a continuous version of the geometric growth. Here, $r$ is the instantaneous growth rate. If we consider an exponential growth over time, the annual rate of change ($r$) in a population at time 0 and t is equal to:

$$r = \frac{ln(\frac{N(T)}{N(0)})}{T}$$

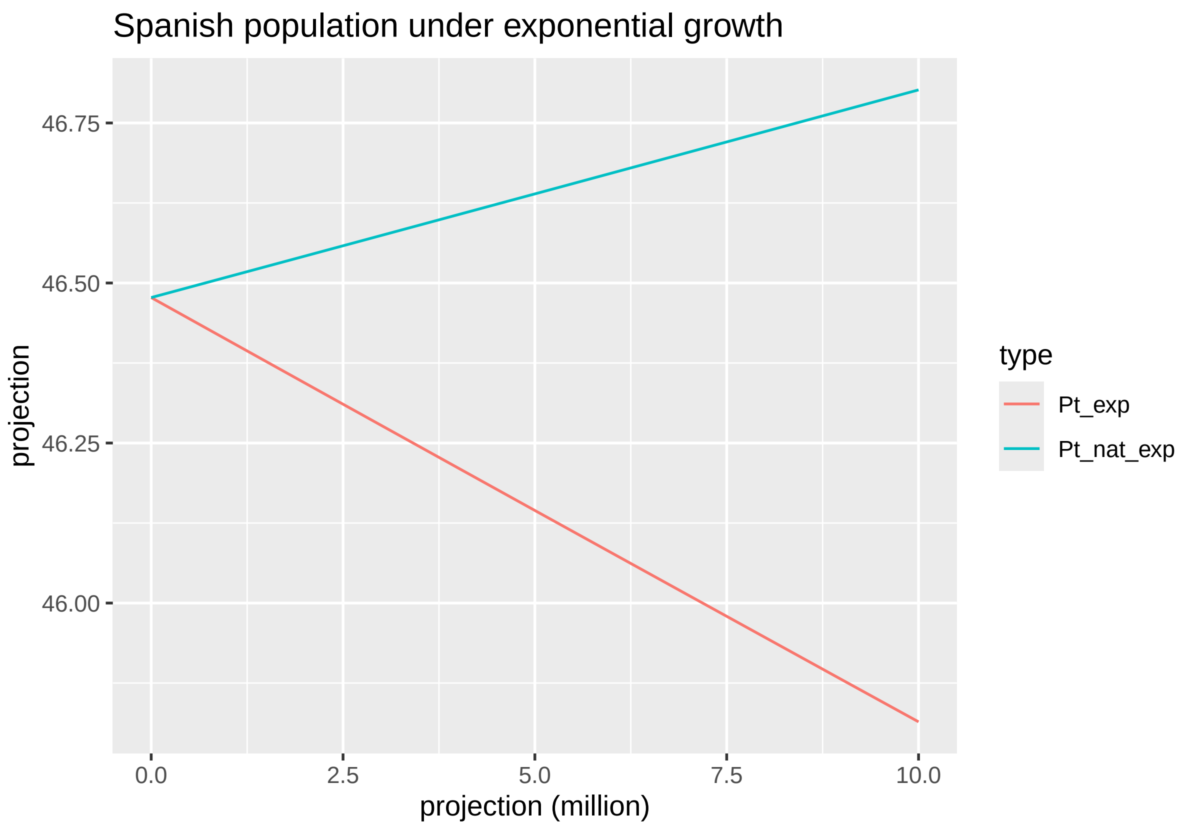
Thus, if $r$ is constant over time the population size at time $t$ can be found by

$$N(t) = N(0)e^{r*t}$$

```
# Exponential growth
exponential_growth <-
  Components %>%
  mutate(r_exp = log(P2 / P1),
         r_nat_exp = log((P1 + B - D) / P1)) %>%
  cross_join(tibble(t = 0:10)) %>%
  mutate(Pt_exp = P1 * exp(r_exp * t),
         Pt_nat_exp = P1 * exp(r_nat_exp * t)) %>%
  select(t, Pt_exp, Pt_nat_exp, r_exp, r_nat_exp)

#It we assume a close population (no migration)

exponential_growth %>%
  select(-r_exp, -r_nat_exp) %>%
  pivot_longer(-t, names_to = "type", values_to = "projection") %>%
  mutate(projection = projection / 1e6) %>%
  ggplot(aes(x = t, y = projection, color = type)) +
  geom_line() +
  xlab("projection (million)") +
    labs(title = "Spanish population under exponential growth")
```




### 4.5 Doubling time

The doubling time is the number of years it would take for a population to double in size, under a fixed rate of change. In an exponential growth situation, if a population double between time 0 and $T$, then

$$ln(\frac{N(T)}{N(0)}) = ln(2)$$

The doubling time (DB) for a population experiencing an exponential growth at rate r can then be found as:

$$DB = \frac{ln(2)}{r}$$

```
# Doubling time
exponential_growth %>%
  filter(t == 0) %>%
  mutate(DB = log(2) / r_exp,
         DB_nat = log(2) / r_nat_exp,
         .keep = "used")
```

```
## # A tibble: 1 x 4
##      r_exp r_nat_exp    DB DB_nat
##      <dbl>     <dbl> <dbl>  <dbl>
## 1 -0.00144  0.000696 -483.   996.
```

# 5 Fertility and replacement

## 5.1 Replacement level

The replacement level is conventionally set TFR of 2.1. This is the average number of children a woman would need to have to reproduce herself by bearing a daughter who survives to childbearing age.

- TFR$>$ 2.1: Population growth
- TFR$=$ 2.1: Replacement level, constant population
- TFR$<$ 2.1: Population decline

As the sex ratio at birth is generally 105 boys per 100 girls and because there is a risk of mortality, the TFR has to be greater than 2 for a woman to be able to *replace* herself by a daughter.

- Note: the value 2.1 is a rule of thumb, since it's based on survival it's a moving variable. The lower infant and child mortality is, the closer this value approaches 2.

## 5.2 Gross reproductive rate

The gross reproductive rate (GRR) is similar to the TFR but uses age-specific rate of having a *female* birth (${}_nF_x^W[0,T]$)

$$ {}_nF_x^W[0,T] = \frac{Number\ of\ female\ births\ between\ times\ 0\ and\ T\ to\ women\ aged\ x\ to\ x+n}{Number\ of\ person-years\ lived\ by\ women\ aged\ x\ to\ x+n\ between\ times\ 0\ and\ T} $$

$$ {}_nF_x^W[0,T] \approx 0.488 \cdot {}_nF_x $$

$$ GRR[0,T] = n \sum_{x=a}^{B-n} {}_nF_x^W[0,T] \approx 0.488 \cdot n \sum_{x=a}^{B-n} {}_nF_x[0,T] $$

```
B <- B %>%
  select(age, births)

Fem <-
  ES2014 |>
  filter(sex == "female") |>
  select(-births) |>
  right_join(B, by = join_by(age)) |>
  mutate(exposure = (pop1+pop2) / 2,
         asfr = births / exposure,
         asfr_f = asfr * .4886)

Fem %>%
  summarize(TFR = sum(asfr),
            GRR = sum(asfr_f))
```

```
## # A tibble: 1 x 2
##      TFR   GRR
##    <dbl> <dbl>
## 1  1.32 0.644
```

## 5.3 Net reproductive rate

The net reproductive rate (NRR) takes mortality into account. The NRR represents the average number of daughters that a female in a cohort would bear in their lifespan if they were subject to the observed age-specific fertility (${}_nF_x^W$) and mortality rates (${}_nM_x$) (Preston, Heuveline, and Guillot 2001). The measure introduces the person-years lived between age x and x+n (${}_nL_x$) in the GRR measure.

$$NRR[0,T] = n\sum_{x=a}^{B-n} {}_nF_x^W[0,T] * \frac{{}_nL_x^W}{l_0}$$

```
NRR <-
  Fem %>%
  left_join(LT, by = join_by(sex,age)) %>%
  select(age, asfr_f, Lx) %>%
  summarize(NRR = sum(Lx * asfr_f)) %>%
  pull(NRR)
NRR
```

```
## [1] 0.6394868
```

The NRR is a rate-based measure of growth!!!

# 6 Population models

## 6.1 Stationary vs Stable population

There are two main population models in demography: Stable and stationary. These models are theoretical types of population used to understand population dynamics under certain mortality and fertility assumptions.

### 6.1.1 Stationary population

- Constant number of births
- Constant age-specific death rates
- Zero age-specific net-migration rates

Some properties

- Constant population size and structure
- CGR= 0
- $e_0 = \frac{1}{CBR} = \frac{1}{CDR}$

### 6.1.2 Stationary population alternative definition

- Constant population size and structure
- constant rates of fertility, mortality, and migration

### 6.1.3 Stable population

- Constant annual growth in birth: $B(t) = B(0)e^{rt}$
- Constant age-specific death rates
- Zero age-specific net-migration rates

The conditions are then:

- Constant age-specific fertility rates
- Constant age-specific death rates
- Age-specific net-migration rates are zero
- Age-specific net-migration rates are constant

## 6.2 Leslie matrix

The Leslie matrix is a matrix model used for population projections, assuming a stable population. The model combined mortality and fertility measures you have seen so far. It is a way to assess the long-term consequences, for the size of a population, of keeping a defined set of age-specific fertility and mortality rates constant over time.

It takes the form of

$$\begin{bmatrix} N(0,t+1) \\ N(1,t+1) \\ N(2,t+1) \\ N(3,t+1) \\ N(4,t+1) \\ N(5,t+1) \end{bmatrix} = \begin{bmatrix} F^*(0) & F^*(1) & F^*(2) & F^*(3) & F^*(4) & F^*(5) \\ {}_1s_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & {}_1s_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & {}_1s_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & {}_1s_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & {}_1s_4 & {}_+s_5 \end{bmatrix} * \begin{bmatrix} N(0,t) \\ N(1,t) \\ N(2,t) \\ N(3,t) \\ N(4,t) \\ N(5,t) \end{bmatrix}$$

where F* are adjusted fertility rates for mortality (see below) and s are the survivorship ratio from age x to x+n.

The Leslie matrix is, generally, built to forecast a close (no migration) female-only population.

### 6.2.1 The cohort component method

The Leslie matrix is a matrix representation of the cohort component method. The matrix (see above) is filled in two steps.

**Step 1**: Project forward the women surviving age-category [x:x+n] We can get one started by placing survivorship ratios on what will later become a matrix subdiagonal. The survivorship ratio is the proportion of people age x-n to x year that will be alive n years later, in a stationary population.

$$s_x = \frac{{}_nL_x}{{}_nL_{x-n}}$$

```
Sx <-
  LT |>
  filter(sex == "female") |>
  select(age, Lx) |>
  mutate(Sx = lead(Lx)/Lx) |>
  select(-Lx) |>
  filter(!is.na(Sx))
Les <-
  Sx |>
  mutate(age_to = age + 1) |>
  pivot_wider(names_from = age,
              values_from = Sx,
              values_fill = 0) |>
  column_to_rownames("age_to") |>
  as.data.frame() |>
  as.matrix()

dim(Les)
```

```
## [1] 110 110
```

```
head(Les[,1:6])
```

```
##          0         1         2         3         4         5
## 1 0.999531 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## 2 0.000000 0.9998389 0.0000000 0.0000000 0.0000000 0.0000000
## 3 0.000000 0.0000000 0.9998915 0.0000000 0.0000000 0.0000000
## 4 0.000000 0.0000000 0.0000000 0.9999097 0.0000000 0.0000000
## 5 0.000000 0.0000000 0.0000000 0.0000000 0.9999056 0.0000000
## 6 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.9999364
```

The Leslie matrix will still need one more row on top and another on the right:

```
Les <-
  0 |>
  rbind(Les) |>
  cbind(0)
```

For the open age interval, the population at time $t+n$ is equal to

$$_{\infty}N_x^W(t+n) = _n N_{x-n}^W(t) * s_x + _{\infty} N_x^W(t) * \frac{T_{x+n}}{T_x}$$

```
nr <- nrow(Les)
#The last cell

Les[nr, nr]<-(LT$Tx[nr] / LT$Tx[nr - 1])
```

- It's also common to put a zero in that corner, in case you prefer to close out the population.

**Step 2**: Finding the number of surviving females in the first age group (age 0)

This step consists in finding the number of female births during the projected period, which survive until the end of the period.

$$_nN_0^W(t+n) = B^W[t, t+n] * \frac{_nL_0}{n * l_0}$$

The way to calculate the number of births is:

$$\begin{aligned} B[t, t+n] &= \sum_{x=\alpha}^{\beta-n} {_nF_x} * n * \frac{(_nN_x^W(t) + _n N_x^W(t+n))}{2} \\ &= \sum_{x=\alpha}^{\beta-n} {_nF_x} * n * \frac{(_nN_x^W(t) + _n N_{x-n}^W(t)\frac{_nL_x}{_nL_{x-n}})}{2} \end{aligned}$$

Thus, $_nN_0^W(t+n)$ is to:

$$_nN_0^W(t+n) = \frac{_nL_0}{2 * l_0}\frac{1}{1+SRB}\sum_{x=\alpha}^{\beta-n} {_nF_x} * n * \left(_nN_x^W(t) + _n N_{x-n}^W(t)\frac{_nL_x}{_nL_{x-n}}\right)$$

where SRB is the sex ratio at birth.

```
# get ASFR as vector
asfr <-
  ES2014 %>%
  filter(sex == "female") %>%
  select(-births) %>%
  # this is the only difference from before
  # in earlier chunk this was a right_join()
  left_join(B, by = join_by(age)) %>%
  mutate(exposure = (pop1 + pop2) / 2,
         asfr = births / exposure,
         asfr = ifelse(is.na(asfr), 0 , asfr)) %>%
  pull(asfr)
length(asfr)
```

```
## [1] 111
```

```r
# SRB and mortality discount constant
# across age of mother in N(0) equation
PF    <- .4886 # or 0.4878049  = (1 / (1 + 1.05)
const <- PF * (LT$Lx[1] / (2 * LT$lx[1]))

sx <- Sx |> pull(Sx)
# Non-constant across mother age
firstrow <- const * (asfr + (lead(asfr, default = 0) * c(sx, 0)))

# Fill the first row
Les[1, ] <- firstrow
```

### 6.2.2 Projections

To forecast the population at time t (P(t)), for 1-year time interval, then

$$P(t) = L^t * P(0)$$

where $L$ is the Leslie matrix and ^ is a matrix power, NOT the kind of power that we're used to! We need `%^%` from the `expm` package to do this right.

This can work inside a *tidy* framework just as we've done for everything so far in this block, and in this way avoid a bunch of loop-writing and indexing overhead.

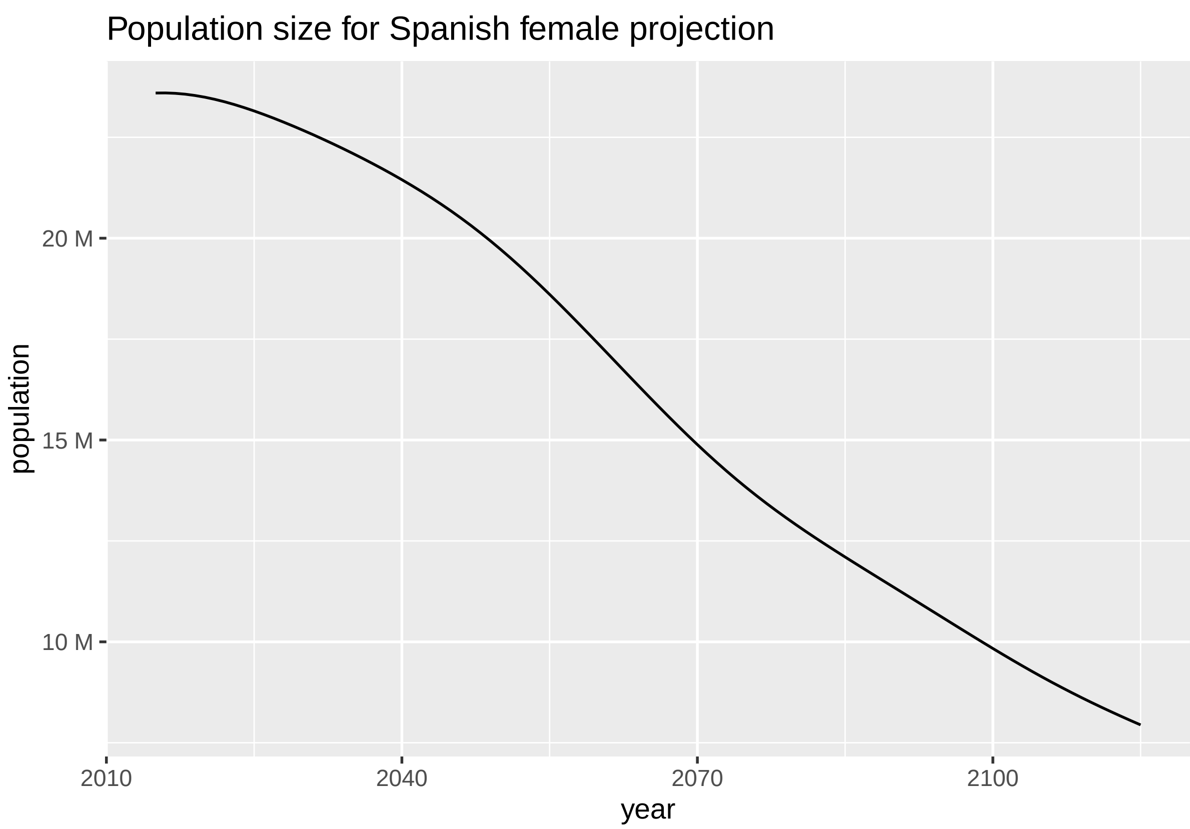
```r
# the projection
Pt <-
  ES2014 %>%
  filter(sex == "female") %>%
  select(age, population = pop2) %>%
  cross_join(tibble(t = 0:100)) %>%
  arrange(t, age) %>%
  group_by(t) %>%
  mutate(population = (Les %^% t[1]) %*% population %>% c()) %>%
  ungroup()
```

### 6.2.3 visualize the results

Total population size:

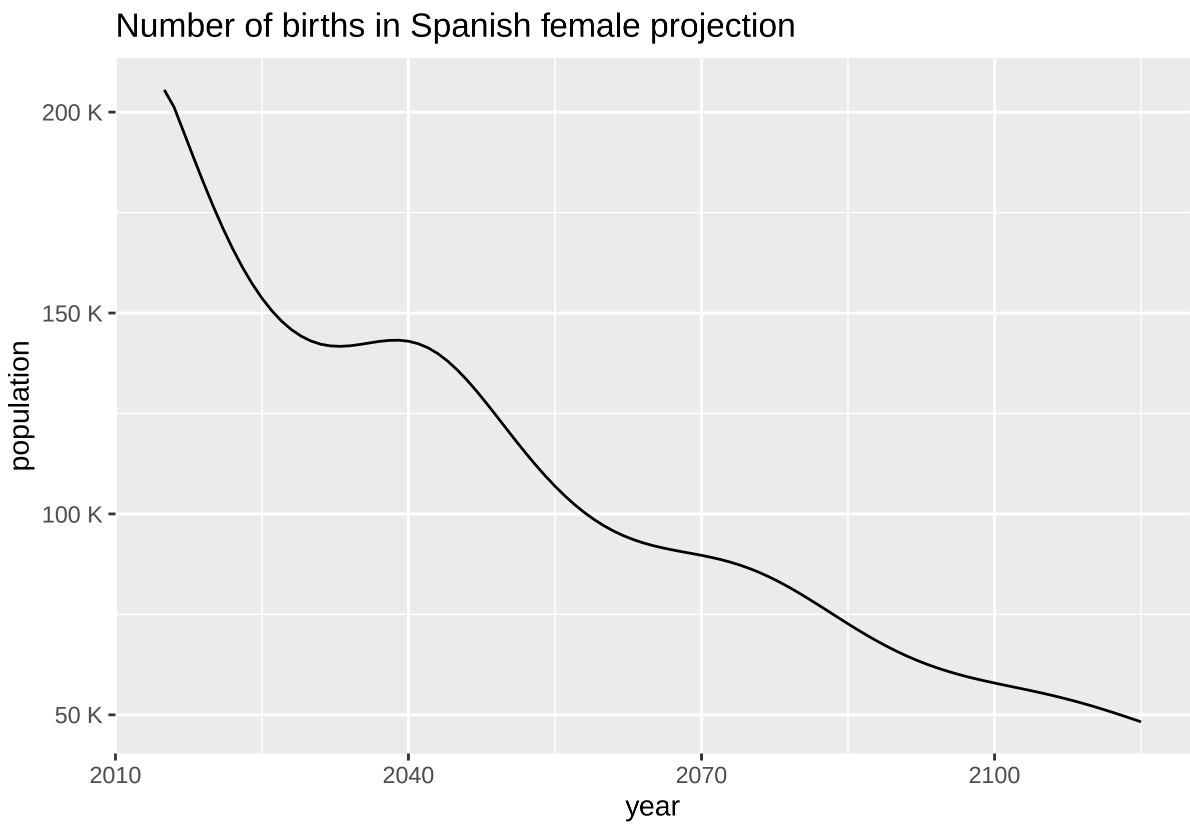
```r
library(scales) # install if needed
Pt %>%
  group_by(t) %>%
  summarize(population = sum(population),
            .groups = "drop") %>%
  mutate(year = 2015 + t) %>%
  ggplot(aes(x = year, y = population)) +
  geom_line() +
```

```r
  scale_y_continuous(labels = label_number(suffix = " M", scale = 1/1000000)) +
  labs(title = "Population size for Spanish female projection")
```



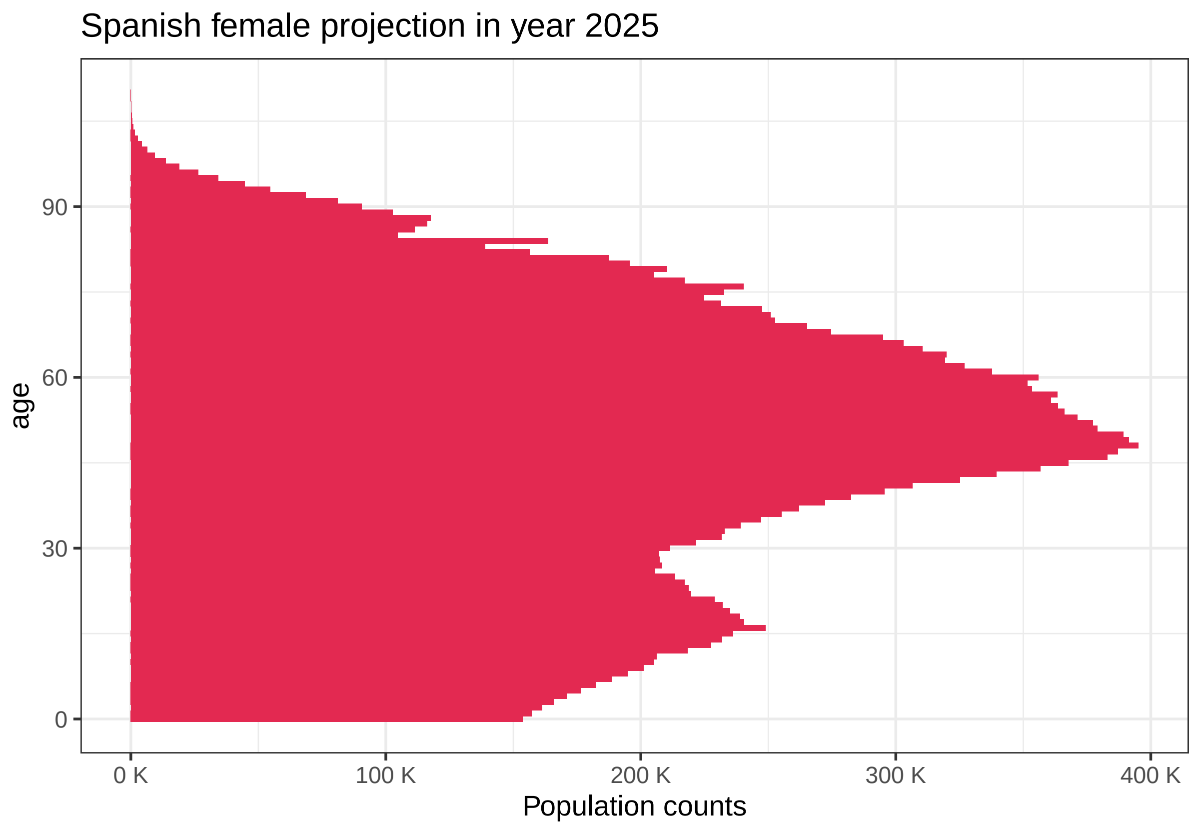

The number of infants (roughly number of births) has attenuating waves

```r
Pt %>%
  filter(age == 0) %>%
  mutate(year = 2015 + t) %>%
  ggplot(aes(x = year, y = population)) +
  geom_line() +
  scale_y_continuous(labels = label_number(suffix = " K", scale = 1/1000)) +
  labs(title = "Number of births in Spanish female projection")
```




The age structure in a given year:

```
Pt %>%
  filter(t == 10) %>%
  ggplot() +
  geom_bar(aes(x = age,
               y = population),
               stat = "identity",
               fill = "#e32951",
               width = 1) +
  ylab("Population counts")+
  coord_flip()+
  theme_bw() +
  scale_y_continuous(labels = label_number(suffix = " K", scale = 1/1000)) +
  labs(title = "Spanish female projection in year 2025")
```



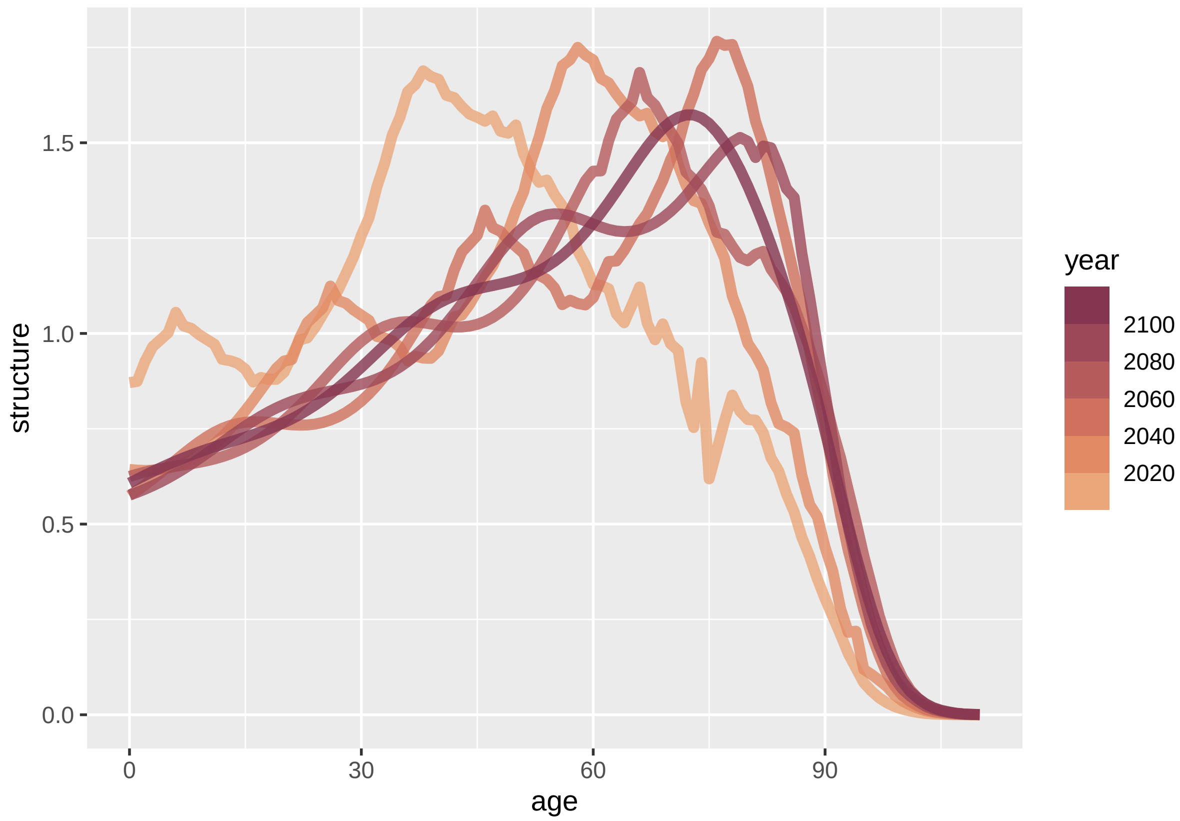

## 6.3 Leslie matrix and stable population properties

### 6.3.1 Constant population structure and crude growth rate

In the long term, the population age structure converges towards a constant age-structure.

```
library(colorspace)
Struct_t <-
  Pt %>%
  filter(t %% 20 == 0) %>%
  mutate(year = 2015 + t) %>%
  group_by(t) %>%
  mutate(structure = 100 * population / sum(population)) %>%
  ungroup()

Struct_t %>%
  ggplot(aes(x = age, y = structure, color = year, group = year)) +   geom_line(linewidth =
  scale_color_binned_sequential("BurgYl", begin = .2, end = 1)
```

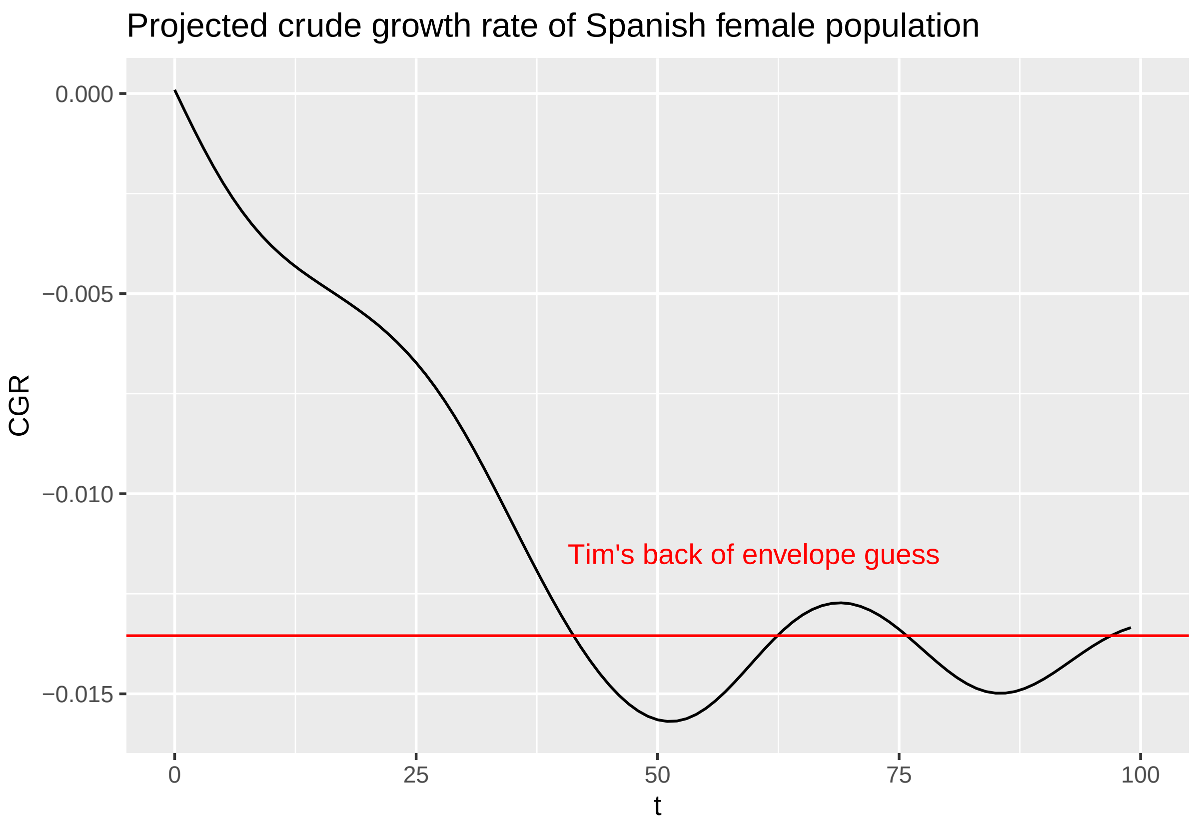



The crude growth rate also converges towards a constant over time in a stable population.

```
CGRt <-
  Pt %>%
  group_by(t) %>%
  summarize(population = sum(population),
            .groups = "drop") %>%
  drop_na() %>%
  mutate(CGR = log(lead(population) / population))

CGRt %>%
  ggplot(aes(x = t, y = CGR)) +
  geom_line() +
  labs(title = "Projected crude growth rate of Spanish female population") +
  geom_hline(yintercept = log(NRR) / 33, color = "red") +
  annotate("text",
           y = -0.0115,
           x = 60,
           label = "Tim's back of envelope guess",
           color = "red")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_line()‘).
```




## Eigenvalue as crude growth rate in stable populations

Eigenvalues are the "latent roots" of a squared matrix. They play a fundamental role in the solution of linear system and provide concrete information "*from a set of static, algebric equations*" (Caswell 2001).

The eigenvalue ($\lambda$) of a matrix A is that

$$Ax = \lambda x$$

where x is some scalar, called eigenvector.

The eigenvalue is found by solvinfg

$$(A - \lambda I)x = 0$$

where I is the identity matrix.

For example (issued from Caswell (2001))

```
#Example

A<-matrix(c(3,2,-6,-5), ncol=2)
A
```

```
##      [,1] [,2]
## [1,]    3   -6
## [2,]    2   -5
```

```
#if x=t(1,1)
x<-matrix(c(1,1), ncol=1)
x
```

```
##      [,1]
## [1,]    1
## [2,]    1
```

```
#then lambda
lambda<- A%*%x

#x is thus the eigenvector of A and lambda is -3
lambda
```

```
##      [,1]
## [1,]   -3
## [2,]   -3
```

The intrinsic crude growth rate of a stable population is found by finding the eigenvalue of the Leslie matrix, $CGR = 1 - \lambda$.

```
# asymptotic growth rate
r <- -(1-as.numeric(eigen(Les)$values[1]))
r
```

```
## [1] -0.01387249
```

```
# crude growth after 200 years
CGRt %>%
  slice(n()-1) %>%
  pull(CGR)
```

```
## [1] -0.01334499
```

```
# go another 100 years and it will get even closer!

# back of envelope (change 33 to get closer)
# 33 is my guess at the stable mean age at childbearing.
log(NRR) / 33
```

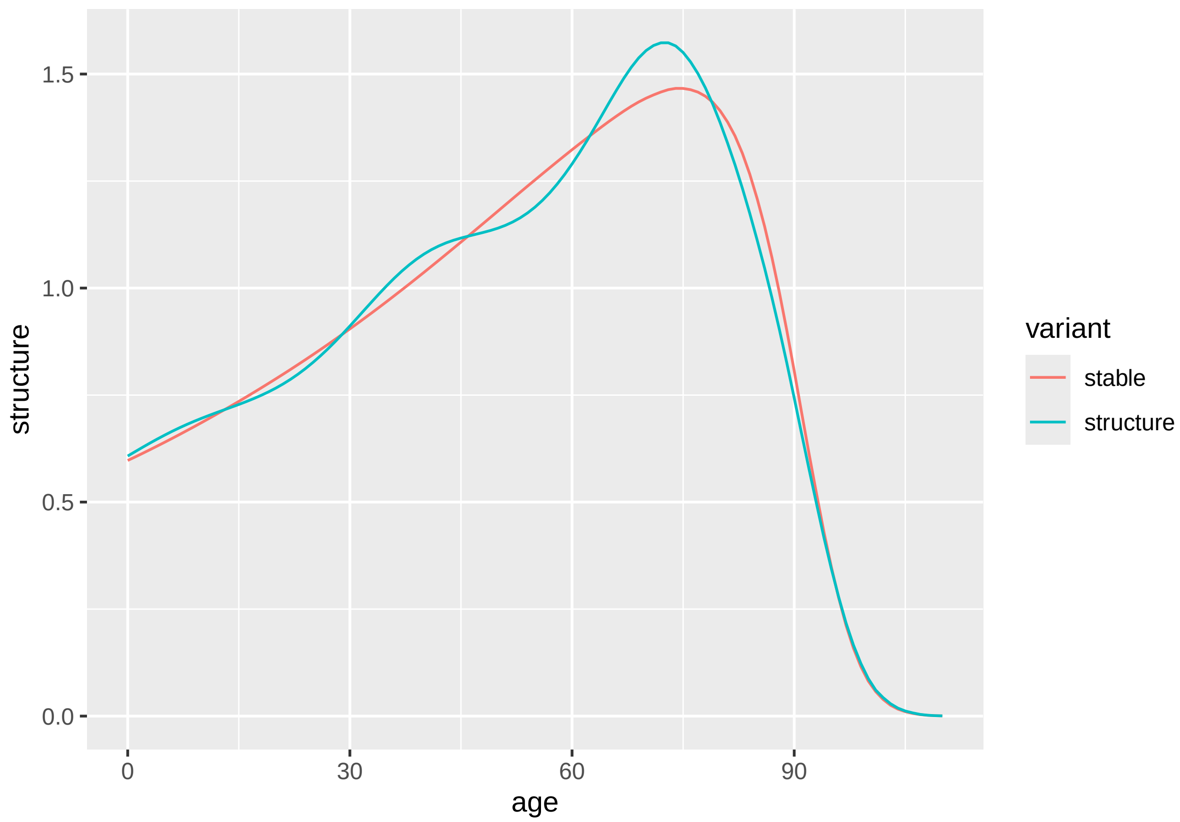
```
## [1] -0.01354816
```

```r
# what would have been the perfect guess?
log(NRR) / r
```

```
## [1] 32.22848
```

The *intrinsic* population structure is the first eigenvector of the Leslie matrix. If you rescale this, then you can compare it to the projected population structure. Eventually there would be perfect overlap.

```r
evec          <- eigen(Les)$vector[, 1] %>% Re()
stable_struct <- tibble(age = 0:110,
                        stable = 100 * evec / sum(evec))
Struct_t |>
  filter(t == max(t)) |>
  left_join(stable_struct, by = join_by(age)) |>
  pivot_longer(structure:stable,
               names_to = "variant",
               values_to = "structure") |>
  ggplot(aes(x = age, y = structure, color = variant, group = variant)) +
  geom_line()
```




## Exercises

1) Load `FR_asfr_2015.csv`, `FR_fLT_2015.csv` and `FR_P_2015.csv`.

2) Calculate the TFR and NRR for France in 2015. Discuss if France reached the population replacement level in 2015.

3) Create the Leslie matrix for French females in 2015.

4) Project French females population 200 years ahead. What is the proportion of the 2015 population left 200 later?

5) How did the age-structure change between 2015 and 2215? Older or younger?

6) Calculate the CGR for France. Discuss your results in relation with the CGR for Spain found in the class.

# References


Caswell, Hal. 2001. *Matrix Population Models*. Wiley Online Library.

Human Fertility Database. 2018. "Max Planck Institute for Demographic Research (Germany) and Vienna Institute of Demography (Austria)."

Human Mortality Database. 2018. "University of California, Berkeley (USA) and Max Planck Institute for Demographic Research (Germany)."

Preston, Samuel H., Patrick Heuveline, and Michel Guillot. 2001. *Demography: Measuring and Modeling Population Processes*. Oxford: Blackwell.