Barcelona Summer School of Demography

Module 2. Demography with R

2. Life Table and Mortality Analysis

Basic Demographic measures

Tim Riffe
tim.riffe@gmail.com

6 July 2021

Contents

1 Prelims

1.1 Thanks again to MP

Again, this material originates with rock star Marie-Pier. I have translated it to a tidy approach, which works particularly well for lifetables.

1.2 packages we either will or might use today

Go ahead and install this:

```
install.packages("demography")
```

Otherwise, we'll be using packages from yesterday, like `tidyverse` and `ggplot2`.

```
library(demography)
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'demography':
##   method      from
##   print.lca   e1071
##   summary.lca e1071
```

```
## This is demography 1.22
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

1.3 R functions

`cumprod`: Returns the cumulative product of an object (vector, matrix or array).

```r
#Example

X <- seq(2,10, 2)
cumprod(X)
```

```
## [1]    2    8   48  384 3840
```

`cumsum`: Returns the cumulative sum of an object (vector, matrix or array).

```r
cumsum(X)
```

```
## [1]  2  6 12 20 30
```

`ifelse()`: This toggles a result depending on whether the first argument evaluates to `TRUE` or `FALSE`. There are three parts, first a logical statement to evaluate, second what to do if it's true, and third what to do if it's false.

```r
ifelse(X >5,"Higher than 5","Lower than 5")
```

```
## [1] "Lower than 5"  "Lower than 5"  "Higher than 5" "Higher than 5"
## [5] "Higher than 5"
```

`case_when()`: This is a *tidy* helper function for when you have lots of conditional cases and `ifelse()` or other options can become cumbersome. `case_when()` let's you list out logicals and what to do in each case. Lines are executed in order, where later lines do not overwrite earlier ones.

```r
case_when(X < 3 ~ "A",
          X < 5 ~ "B", # this is only activated if the < 3 condition
                       # was FALSE
          TRUE ~ "C")  # This is a catch-all, to pick up the rest
```

```
## [1] "A" "B" "C" "C" "C"
```

When using `case_when()` like the above, arrange logicals in order from the most specific to the most general.

`rev()`: reverse an R object

```r
X
```

```
## [1]  2  4  6  8 10
```

```r
rev(X)
```

```
## [1] 10  8  6  4  2
```

`for(){}`: For loop, i.e. iterate over a data object.

```r
A<-c(1:5)
for(i in 1:5){
  A[i]<-A[i]/2
}
A
```

```
## [1] 0.5 1.0 1.5 2.0 2.5
```

2 Human Mortality Database

The Human Mortality Database (HMD) is an important database, offering detailed mortality and population data to researchers interested in understanding human mortality and longevity (Human Mortality Database 2018). It groups information on 40 countries or areas and use a common protocol to process data from all countries.

Access to the data are free! Please register now (www.mortality.org), we will use this dataset for the rest of the class.

There are two ways to use the HMD data in R:

- Copy-paste the data in a .txt document, save it and then read it using the read.table() command.
- Use a R package allowing to load the data directly in R - e.g. *demography* (Hyndman et al. 2017) and *HMDHFDplus* (Riffe, Boe, and Goldstein 2015).

2.1 Using the *demography* package to load HMD data

The *demography* package (Hyndman et al. 2017) has a command named hmd.mx which allows one to upload mortality rates (m_x) and population counts data directly from the HMD.

```
# install.packages("demography")
library(demography)

# insert you username and password, then uncomment and run this.
# data <- hmd.mx("ESP", "your_user_name", "your_password")

# Data structure
str(data)
```

```
## List of 7
##  $ type  : chr "mortality"
##  $ label : chr "ESP"
##  $ lambda: num 0
##  $ year  : int [1:111] 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 ...
##  $ age   : num [1:111] 0 1 2 3 4 5 6 7 8 9 ...
##  $ pop   :List of 3
##   ..$ female: num [1:111, 1:111] 295481 263968 250296 242430 244758 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##   ..$ male  : num [1:111, 1:111] 307918 268323 254684 246566 246617 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##   ..$ total : num [1:111, 1:111] 603399 532291 504980 488996 491376 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##  $ rate  :List of 3
```

```
##   ..$ female: num [1:111, 1:111] 0.1598 0.0827 0.0452 0.0247 0.0163 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##   ..$ male  : num [1:111, 1:111] 0.1893 0.0863 0.0455 0.0258 0.0166 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##   ..$ total : num [1:111, 1:111] 0.1748 0.0846 0.0453 0.0253 0.0164 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:111] "0" "1" "2" "3" ...
##   .. .. ..$ : chr [1:111] "1908" "1909" "1910" "1911" ...
##  - attr(*, "class")= chr "demogdata"
```

First, note this data is not tidy… Please excuse me while I tidy it up for later use. This chunk you can examine if you want, but I don't intend to narrate it much. In this case `exposure` has been called `pop`, and we don't need extra exposure calcs.

```
library(tidyverse)
sexes <- data$pop %>% names()

# two containers, columns given, but no rows
ESpop   <- tibble(Year = NULL, Age = NULL, Sex = NULL, Exposure = NULL)
ESrates <- tibble(Year = NULL, Age = NULL, Sex = NULL, M = NULL)

for (i in sexes){
  ESpop <- data$pop[[i]] %>%
    as_tibble() %>%
    rownames_to_column("Age") %>%
    pivot_longer(cols = -Age,
                 names_to = "Year",
                 values_to = "Exposure") %>%
    mutate(Sex = i,
           Age = as.integer(Age) - 1) %>%
    bind_rows(ESpop)

  ESrates  <- data$rate[[i]] %>%
    as_tibble() %>%
    rownames_to_column("Age") %>%
    pivot_longer(cols = -Age, names_to = "Year", values_to = "M") %>%
    mutate(Sex = i,
           Age = as.integer(Age) - 1) %>%
    bind_rows(ESrates)
}

ES <- left_join(ESpop,
                ESrates,
                by = c("Age", "Year", "Sex")) %>%
  select(Year, Sex, Age, Exposure, M) %>%
  arrange(Year, Sex, Age)
```

3 Life table

The life table is one of the most important demographic tools. It takes the form of a table, where each column consists of a different age-specific mortality indicator. As put by Preston, Heuveline, and Guillot (2001), "*it is a table that displays various pieces of information about the dying out of a birth cohort.*"

However, data for cohorts are often incomplete as it might take over 100 years or more for a cohort to die out. Data for cohorts are also often unavailable or outdated. Demographers thus came up with the concept of the *period* life table and *synthetic* cohort.

Period life table: It is similar to a cohort life table, but "*the information attempts to show what would happen to a cohort if it were subjected for all of its life to the mortality conditions of that period*" (Preston, Heuveline, and Guillot 2001).

Synthetic cohort: Hypothetical cohort if certain mortality conditions pertained through its life - e.g. if it experienced death rates observed in one calendar year.

One can also think of the period lifetable as an *annualization* of observed rates, in the sense that some of the lifetable columns are conveniently and intuitively expressed in year units.

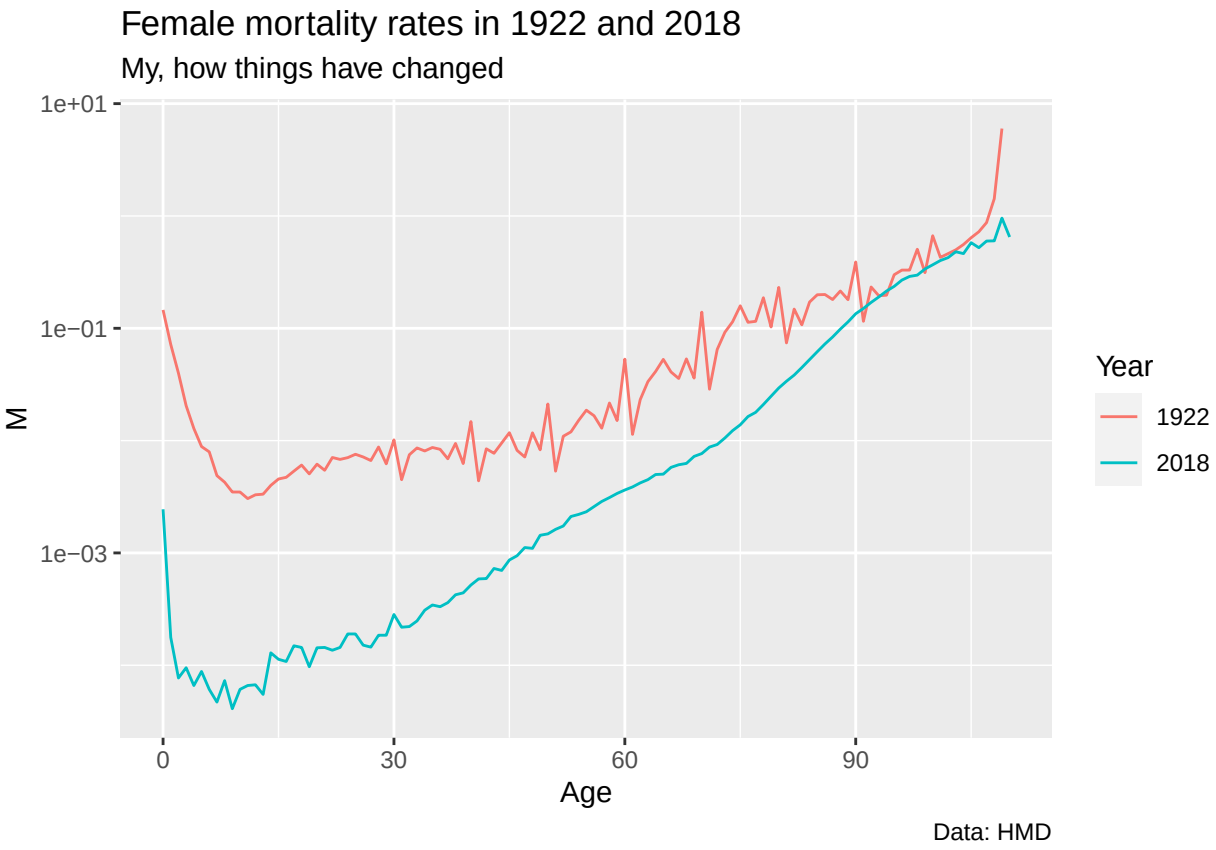
Because period life tables are often used, we will focus on this construct. However, note that the calculation of a cohort life table is the same of that of a period life table. The only difference is that a cohort life table starts from death *probabilities* between age x and x+n ($_nq_x$) and a period life table starts from death *rates* between age x and $x+n$ ($_nm_x$).

3.1 Death rates between age x and x+n $_nm_x$

As mentioned in the previous class, age-specific death rates are the risk of dying in a specific age interval. Death rates ($_nm_x$) are calculated from observed death counts and exposure to risk (person-years). A period life table starts from this indicator.

```
# Extract death rates with the demography package and include them in the life table

ES %>%
  filter(Year %in% c(1922, 2018),
         Sex == "female") %>%
  ggplot(aes(x = Age, y = M, color = Year)) +
  geom_line() +
  scale_y_log10() +
  labs(title = "Female mortality rates in 1922 and 2018",
       subtitle = "My, how things have changed",
       caption = "Data: HMD")
```



Look, you can see some age heaping in 1922 rates! Is it in both the numerator and the denominator?

3.2 Death probabilities between age x and x+n ${}_nq_x$

The first and key step is to transform a set of age-specific death rates into a set of age-specific probabilities of dying (${}_nq_x$). The relation between ${}_nm_x$ and ${}_nq_x$ have been established based on analyses of actual cohorts (for mathematical proof, see Preston, Heuveline, and Guillot (2001), p. 42-43).

$$ {}_nq_x = \frac{n *_n m_x}{1 + (n -_n a_x)_n m_x} $$

where ${}_na_x$ is the average number of person-years lived in the interval by those dying in the interval and n is the length of the age-interval.

Generally, ${}_na_x = n/2$ with the exceptions of the first and the last age group. Other approximations are also available, but these only matter when age groups are wider than a year.

Infant mortality tends to occur in the first months after birth. Thus, the deaths do not occur, on average, at the mid-point ($n/2$) of the interval, but closer to age 0. In recent years, for females, ${}_1a_0 = 0.14903 - 2.05527 *_1 m_0$ (Human Mortality Database 2018). See the HMD protocol at http://www.mortality.org/Public/Docs/MethodsProtocol.pdf (Human Mortality Database 2018, 37) for more details. I have simplified the HMD piecewise approach by averaging male and female model results.

```r
# example of using case_when()
 case_when(M < .02012 ~ .14916 - 2.02536 * M,
           M < .07599 ~ 0.037495 + 3.57055 * M,
           M >= .07599 ~ 0.30663)
```

We'll use this approximation for all sex strata, but if you were doing serious work you'd want to use a more nuanced approach. Not to imply that this this detail is worth losing sleep over, but really it doesn't overly complicate the code once you get going.

The last age group is usually an open age interval - e.g. 85+ or 110+. People dying in this open interval are thus susceptible to live, on average, longer than $n/2$ years. For the last age interval, here 110+, ${}_\infty a_{110} = 1/{}_\infty m_{110}$. That's a good-enough approximation in ages as high as 110, but for lower open age groups it's better to close out more thoughtfully, possibly by extrapolating mortality rates.

In a life table, ${}_\infty q_{110+}$ is equal to 1, as every member of a cohort has to die in the last age group.

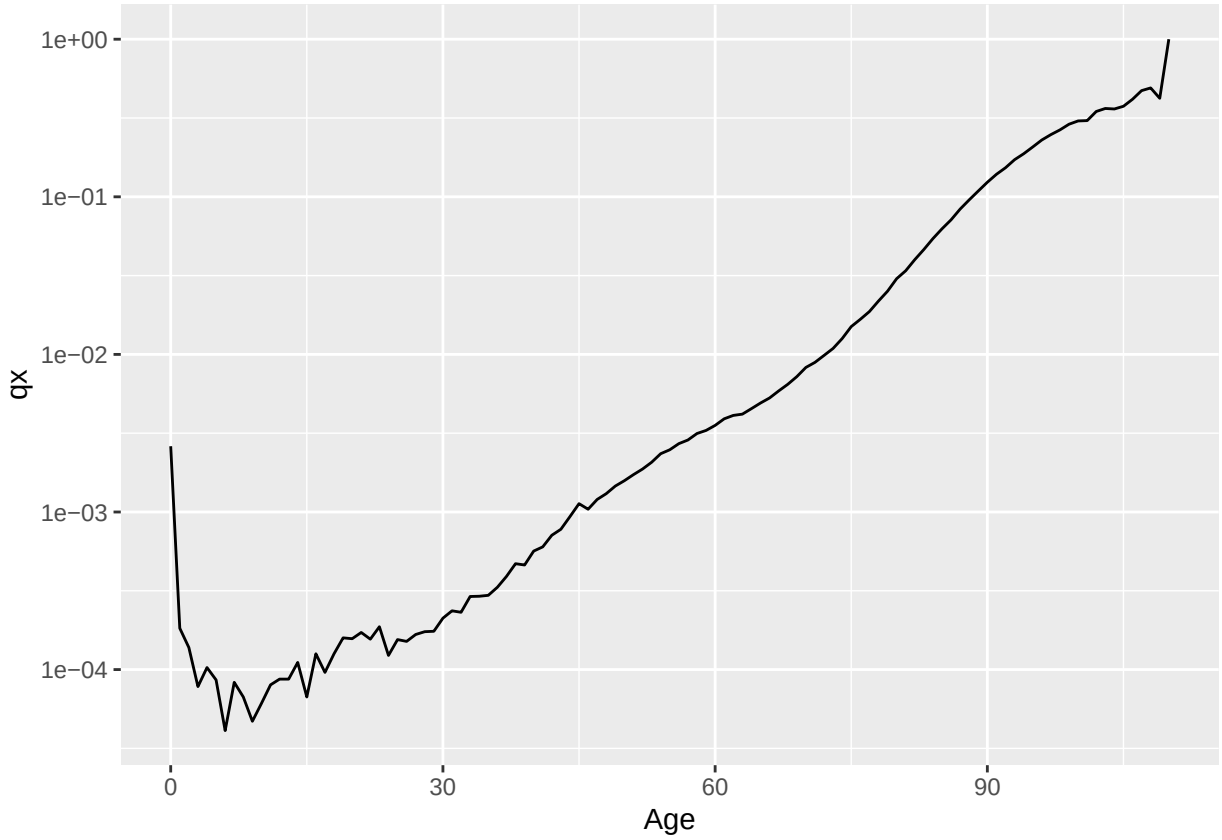
```r
# NOTE: take care to handle closeout NAs!!!
# They screw up the whole lifetable!

# Set nx and ax
LT <-
  ES %>%
  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),   # hack
         qx = (M * n) / (1 + (n - ax) * M),
         qx = ifelse(qx > 1, 1, qx))           # hack
```

Notes on the above:

1. There are `NA` values of M_x in some very high ages where most likely no one was alive and so no one died. We could either truncate the lifetable at the highest non-`NA` value, or else impute something. In this case, you could say there was no mortality observed, and impute a 0. Or, since these values are in the highest ages that ought to be subject to high rates, we could impute a high value. I chose to just impute a rate of .5. A more aesthetic thing to do would be to smooth the final ages, including some sort of extrapolation. These choices have little leverage on life expectancy, but could be consequential for other lifetable measures.
2. a_x can evaluate to infinity in the open age group if the closeout M_ω happens to be 0. In this case again we should impute, and it won't make a difference to summary results to just plug in a number like `0.5`.
3. The formula for q_x can produce values greater than 1 if a_x isn't well estimated. In our case this happens a few times in ages where M_x is very high, but where we've assumed `0.5` for a_x.

```r
LT %>%
  filter(Year == 2014,
         Sex == "female") %>%
  ggplot(aes(x = Age, y = qx)) +
  geom_line() +
  scale_y_log10()
```



3.3 Survival probabilities between age x and $x+n$, ${}_np_x$

The survival probabilities between age x and $x+n$ (${}_np_x$) is simply one minus ${}_nq_x$. It is interpreted as the chance of surviving from age x to age x+n.

$$ {}_np_x = 1 - {}_n\, q_x $$

3.4 Survival probabilities to age x, l_x

This indicator indicates the chance of surviving from birth to age x (l_x) OR the number of survivors at age x relative to the radix of the life table. The l_0 is interpreted as the initial size (radix) of the population, generally set to 1 or 100,000. There are three ways of calculating this indicator:

Option 1 :

$$ l_{x+n} = r \prod_{y=0}^{x} {}_np_y $$

Option 2 :

$$l_{x+n} = l_x * {}_np_x$$

Option 3:

$$l_{x+n} = l_x -_n d_x$$

Note: $l_1 =_0 p_1$

3.5 Death distribution, ${}_nd_x$

The life table deaths (${}_nd_x$) is the number of persons dying between age x and x+n, relative to the radix, and represents the distribution of deaths over age. There is two ways of calculating ${}_nd_x$.

Option 1:

$${}_nd_x =_n q_x * l_x$$

Option 2:

$${}_nd_x = l_x - l_{x+n}$$

3.6 Person-years lived between age x and $x + n$, ${}_nL_x$

The number of person-years between age x and $x + n$ (${}_nL_x$) is calculated as:

$$\begin{aligned} {}_nL_x &= n(l_x -_n d_x) +_n a_x *_n d_x \\ &= n * l_x + (n -_n a_x){}_nd_x \end{aligned}$$

Note

${}_nm_x =_n d_x/{}_nL_x$

and

${}_nq_x =_n d_x/l_x$

3.7 Person-years lived above age x T_x

Calculating the number person-years lived above age x (T_x) is a key step to calculate life expectancy. It consists in finding the sum of ${}_nL_x$ from age x:

$$T_x = \sum_{y=x}^{\infty} {}_nL_y$$

3.8 Life expectancy e_x

The last indicator in the life table is probably one of the most used in demographic analysis. The life expectancy is the average number of years lived by a (synthetic) cohort reaching age x. It consists in dividing the number of person-years lived above age x by the number of people alive at age x:

$$e_x = \frac{T_x}{l_x}$$

Since `mutate()` let's you make columns in a sequentially dependent way, we can actually do this whole lifetable inside a single `mutate()` statement. However, each combination of `Year` and `Sex` is an independent lifetable, so we need to declare groups beforehand using `group_by()`:

```
radix <- 1e5 # 100k this is an arbitrary convention
LT <-
  LT %>%
  group_by(Year, Sex) %>%
  mutate(px = 1 - qx,
         lx = radix * c(1, cumprod(px[-n()])),
         dx = qx * lx,
         Lx = n * lx + (n - ax) * dx,
         Tx = Lx %>% rev() %>% cumsum() %>% rev(),
         ex = Tx / lx,
         ex = ifelse(is.nan(ex),ax,ex)) %>%
  ungroup()
```

3.9 The final life table

The result is a multi-column life table where each column informs on an aspect of mortality for each year-sex combination in this data series. Yay!

```
head(LT)
```

```
## # A tibble: 6 x 14
##   Year  Sex      Age Exposure      M     n    ax     qx    px      lx     dx
##   <chr> <chr>  <dbl>    <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>  <dbl>
## 1 1908  female     0  295481. 0.160      1 0.307 0.144  0.856 100000  14389.
## 2 1908  female     1  263968. 0.0827     1 0.5   0.0795 0.921  85611.  6802.
## 3 1908  female     2  250296. 0.0452     1 0.5   0.0442 0.956  78809.  3482.
## 4 1908  female     3  242430. 0.0247     1 0.5   0.0244 0.976  75327.  1839.
## 5 1908  female     4  244758. 0.0163     1 0.5   0.0161 0.984  73488.  1186.
## 6 1908  female     5  241492. 0.0111     1 0.5   0.0111 0.989  72302.   800.
## # ... with 3 more variables: Lx <dbl>, Tx <dbl>, ex <dbl>
```

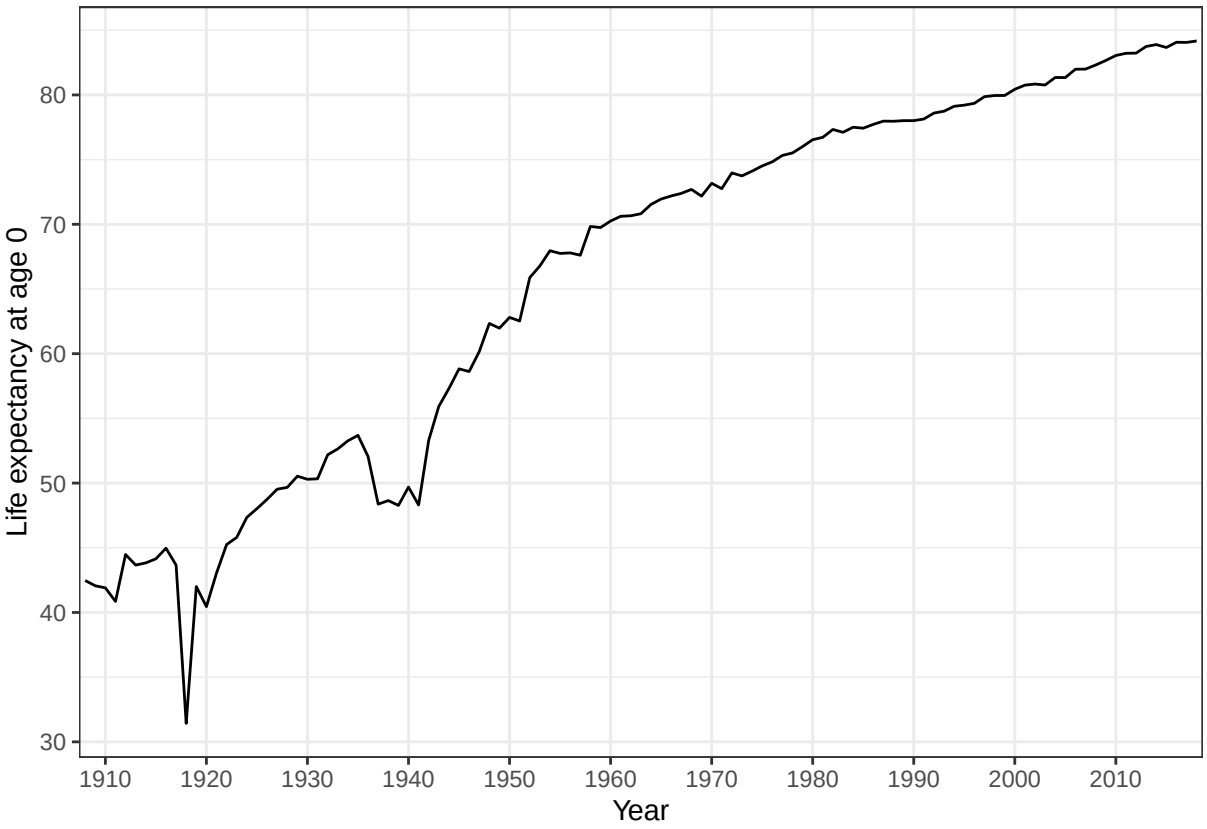
4 Visualizing the results

Since the data are tidy, we can plot at will

4.1 Life expectancy at age 0

A time series of life expectancy at birth

```
x = 0
LT %>%
  filter(Age == x, Sex == "total") %>%
  ggplot(aes(x = Year, y = ex, group = 1)) +
    geom_line() +
    ylab(paste("Life expectancy at age", x)) +
    scale_x_discrete(breaks = seq(1910,2010,by=10))+
    xlab("Year") +
    theme_bw()
```

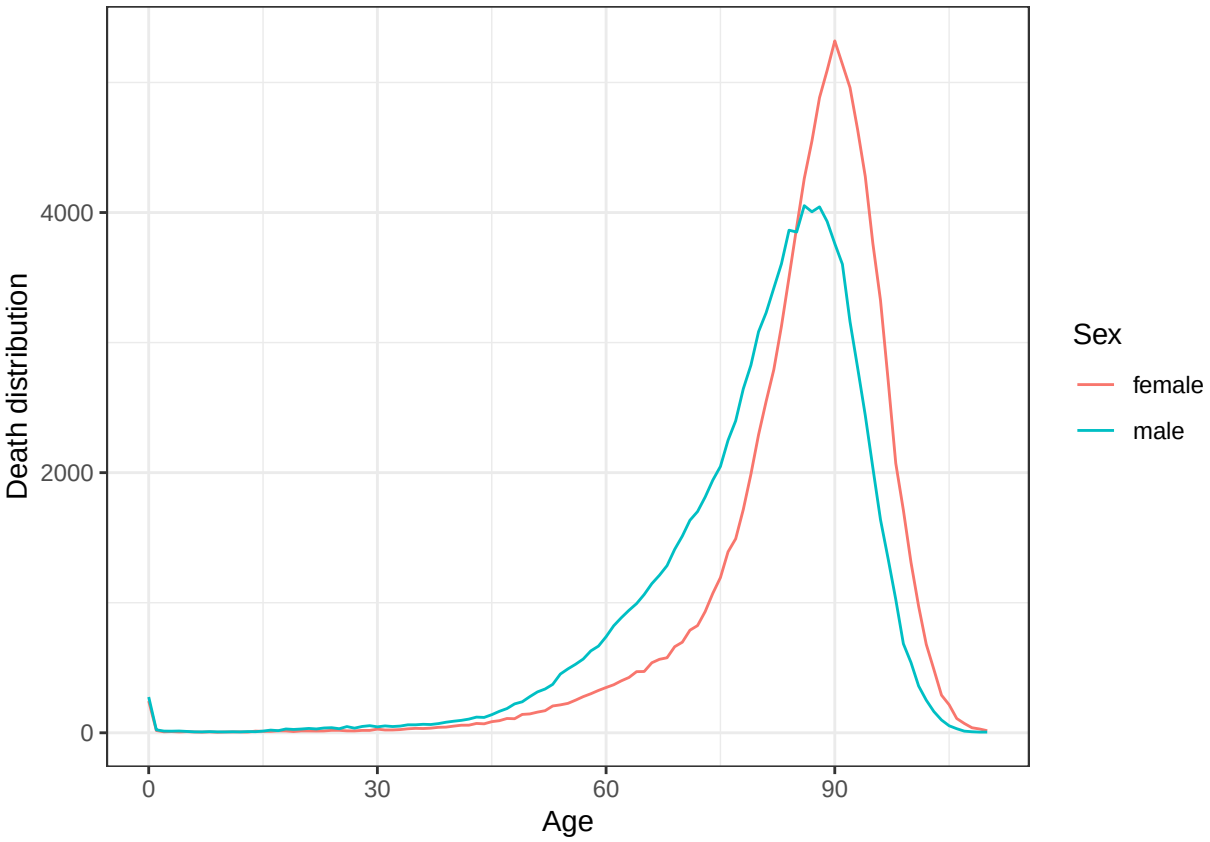


4.2 death distribution

The death distribution is packed with information. You can calculate statistical summary measures on it just like any statistical distribution (just divide out radix). Let's not forget this refers to a hypothetical cohort though!

```
LT %>%
  filter(Year == 2018, Sex != "total") %>%
  ggplot(aes(x = Age, y = dx, color = Sex)) +
        geom_line() +
        ylab("Death distribution") +
        xlab("Age") +
```

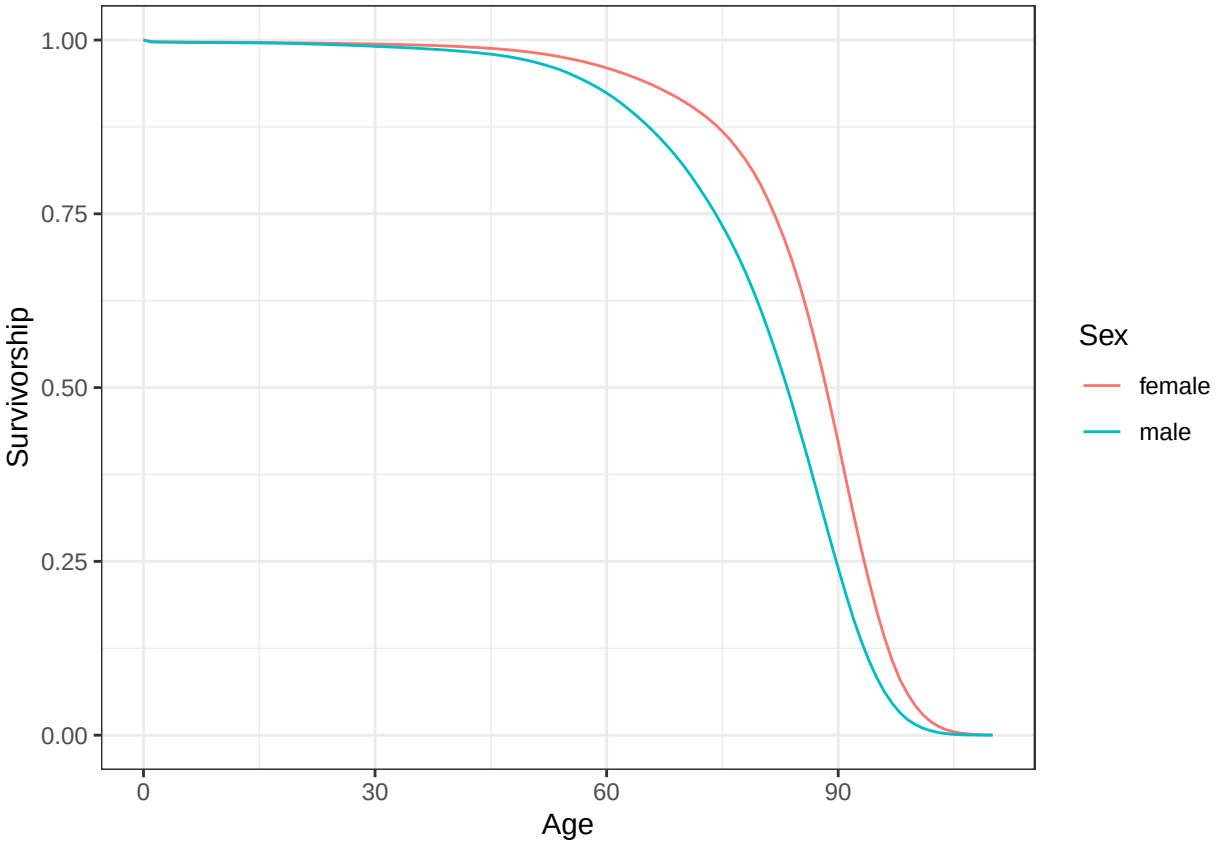
```
        theme_bw()
```



4.3 survival curve

Lifetable survivorship, here plotted with a radix of 1, is interpreted as the probability of surviving from 0 until age x.

```
LT %>%
  filter(Year == 2018, Sex != "total") %>%
  ggplot(aes(x = Age, y = lx / 1e5, color = Sex)) +
        geom_line() +
        ylab("Survivorship") +
        xlab("Age") +
        theme_bw()
```



Exercises

1. Choose a country in the HMD and calculate its life table for at least 20 consecutive years.
2. Compare your results with results with those in the HMD.
3. Plot ${}_nm_x$, ${}_nl_x$ and ${}_nd_x$ for the first and last year you chose. How did the different indicators change over time?
4. Plot e_0 and e_{65} over time. How did life expectancy change over time?

References

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

Hyndman, Rob J., Heather Booth, Leonie Tickle, and John Maindonald. 2017. *Package 'demography'*. https://cran.r-project.org/web/packages/demography/index.html.

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

Riffe, Tim, Carl Boe, and Josh Goldstein. 2015. *Package 'HMDHFDplus'*. https://cran.r-project.org/web/packages/HMDHFDplus/index.html.