Demographic Methods - Practical 5 (Fertility and Reproduction)

2025-10-29

The heading of the R script

The R script begins by clearing the workspace. It utilises the ggplot2 package for plotting; please ensure the package is installed.

```
rm(list = ls())
#install.packages("ggplot2")
library(ggplot2)
```

Reading the data

In this exercise, we are provided with age-specific estimates of the mid-year U.S. female population from 1933 to 2023, denoted as ${}_nW_x$; age-specific estimates of the mid-year U.S. population for both sexes, ${}_nN_x$; the number of person-years lived within each age interval for the female population, ${}_nL_x$, assuming a radix of 100,000 individuals; and the annual number of births by age of the mother, B. The data for this exercise have been compiled from two public repositories: the *Human Mortality Database* (HMD) and the *Human Fertility Database* (HFD). For ease of access, an excerpt has been temporarily stored in a GitHub repository and can be retrieved using the following lines of code. However, the data can also be downloaded directly from the repositories (including those for other countries) by running the R script "practical_5_data.R", and providing your username and password.

```
GitHub               = "https://raw.githubusercontent.com/Romero-Prieto/teaching/main/Demographic%20Met
data                 = read.csv(GitHub)
print(data[1:20, ])
```

```
##    Year  x       nWx nNx_total   nLx         B
## 1  1933  0  971181.3   1975036 96130      0.00
## 2  1933  1 1005773.9   2034701 94374      0.00
## 3  1933  2 1077549.6   2178069 93767      0.00
## 4  1933  3 1101794.0   2229949 93444      0.00
## 5  1933  4 1118327.1   2274434 93206      0.00
## 6  1933  5 1155442.6   2368433 93016      0.00
## 7  1933  6 1179393.3   2411105 92853      0.00
## 8  1933  7 1196259.9   2436694 92709      0.00
## 9  1933  8 1213611.3   2462997 92579      0.00
## 10 1933  9 1227038.1   2480959 92463      0.00
## 11 1933 10 1236716.2   2494398 92357      0.00
## 12 1933 11 1246816.4   2511742 92258      0.00
## 13 1933 12 1237992.0   2492852 92158     46.85
## 14 1933 13 1216487.1   2446315 92052    531.69
## 15 1933 14 1197058.1   2402833 91932   2120.63
## 16 1933 15 1182135.8   2368786 91794   7945.19
## 17 1933 16 1173010.9   2346014 91634  21998.16
## 18 1933 17 1168279.8   2331751 91452  44931.04
## 19 1933 18 1164102.0   2318824 91250  76672.49
## 20 1933 19 1158501.9   2303378 91027 101004.36
```

Data preparation

Fertility and reproduction calculations focus on the year 2001, though any year between 1933 and 2023 may be selected. The following lines show how to filter the dataset for the year 2001 and generate key variables by selecting specific columns and rows. The function “rm()” is used to eliminate unnecessary information.

```
year                     = 2001
radix                    = 100000
data                     = data[data[, "Year"] == year, ]
x                        = data[, "x"]
nWx                      = data[, "nWx"]
nNx_t                    = data[, "nNx_total"]
nLx                      = data[, "nLx"]
B                        = data[, "B"]
rm(data)
```

As is standard in life table calculations, the length of age intervals can be defined based on the values of x, while auxiliary variables—such as the open-ended age interval and reproductive age ranges—can be identified using logical operators.

```
n                        = c(diff(x,1),NA)
sEL                      = !is.na(n)
rEP                      = x >= 15 & x < 50
```

Examples

Data could be used to calculate the life expectancy at birth e_0, using all values of ${}_nL_x$ and the radix.
$e_0 = \frac{\sum_{a=0}^{\infty} {}_nL_a}{l_0}$

```
sum(nLx)/radix
```

```
## [1] 79.51623
```

The same mortality data could be used to calculate the number of years a newborn is expecting lo live while in the reproductive ages, i.e., between 15 and 50, dividing the total number of person-years years lived between 15 and 50 by the radix.
$\frac{T_{15}-T_{50}}{l_0} = \frac{\sum_{a=15}^{\infty} {}_nL_a - \sum_{a=50}^{\infty} {}_nL_a}{l_0}$

```
(sum(nLx[x >= 15]) - sum(nLx[x >= 50]))/radix
```

```
## [1] 34.27822
```

alternatively,
$\frac{\sum_{a=15}^{50-n} {}_nL_a}{l_0}$

```
sum(nLx[x >= 15 & x < 50])/radix
```

```
## [1] 34.27822
```

which is the same as:

```
sum(nLx[rEP])/radix
```

```
## [1] 34.27822
```

Exercise 1: Fertility and reproduction question (Compulsory)

Calculate the following values and comment on your results.

a. Crude Birth Rate.

```
sum(B)/sum(nNx_t)*1000
```

```
## [1] 14.15159
```

b. Child-Woman Ratio. Hint: use the variable "rEP" to identify observations related to reproductive ages.

```
sum(nNx_t[x < 5])/sum(nWx[rEP])
```

```
## [1] 0.2671958
```

c. General Fertility Rate.

```
sum(B)/sum(nWx[rEP])*1000
```

```
## [1] 55.69065
```

d. Age specific fertility rates ${}_nf_x$.

```
nFx                     = B/nWx
```

e. Total Fertility Rate.

```
TFR                     = sum(n[rEP]*nFx[rEP])
TFR
```

```
## [1] 2.027256
```

f. Gross Reproduction Rate.

Hint: assume a Sex Ratio at Birth equal to 1.05 males per female.

```
SRB                     = 1.05
GRR                     = 1/(1 + SRB)*TFR
GRR
```

```
## [1] 0.9889054
```

g. Net Reproduction Rate.

```
NRR                     = 1/(1 + SRB)*sum(nFx[rEP]*nLx[rEP]/radix)
NRR
```

```
## [1] 0.9741579
```

h. Calculate the Mean Age of the Fertility Schedule μ_f.

```
age                     = x + n/2
sum(n[rEP]*nFx[rEP]*age[rEP])/TFR
```

```
## [1] 27.55167
```

i. Calculate the probability to survive to the Mean Age of the Fertility Schedule $p(\mu_f)$.

```
NRR/GRR
```

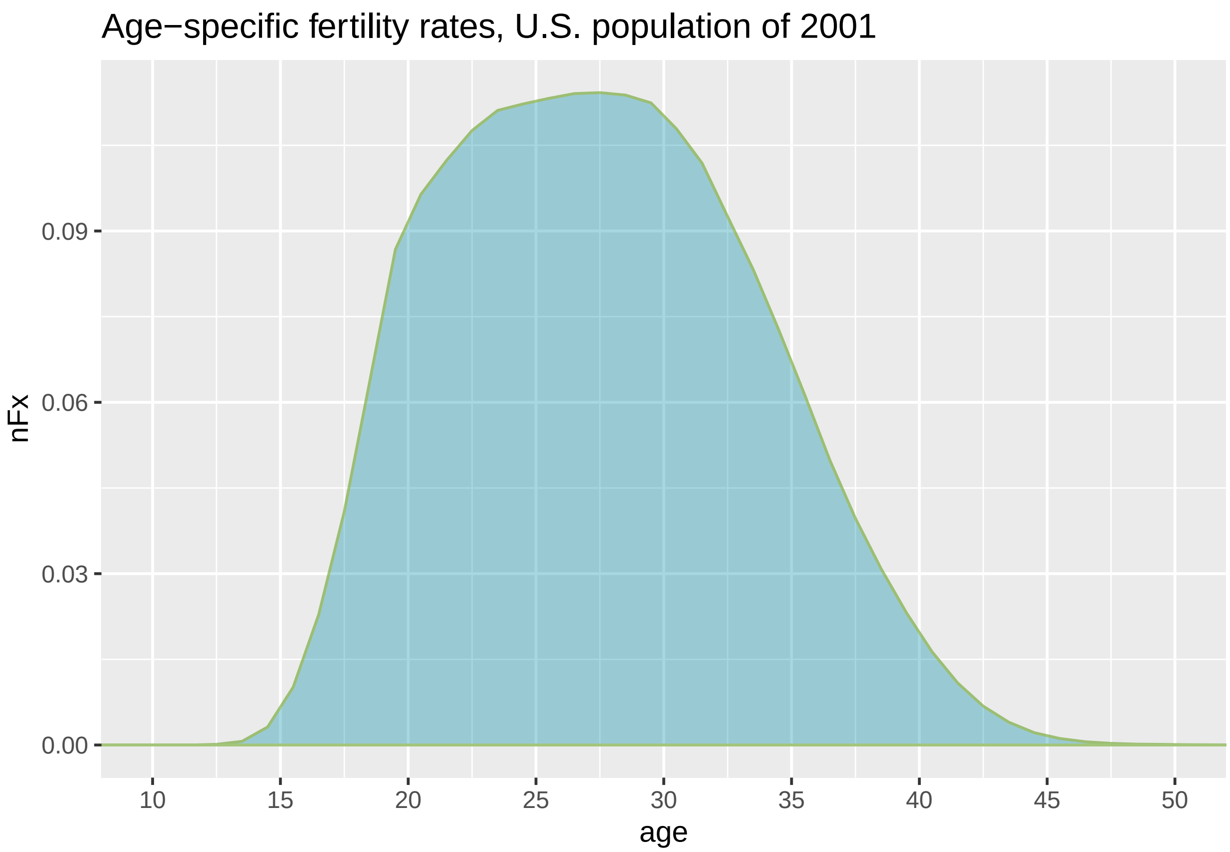
```
## [1] 0.9850871
```

Exercise 2: Fertility and reproduction question (Compulsory)

Plot the Fertility Schedule (i.e., ${}_nf_x$ as a function of x).

Hint: Rates are plotted at the midpoint of the age interval.

```
coloUR                  = c("A"      = rgb(0.00,0.55,0.65,0.35),
                            "B"      = rgb(0.45,0.65,0.20,0.65))
fertility               = data.frame(age, nFx)
fertility               = fertility[sEL, ]
ggplot(fertility, aes(x = age, y = nFx)) +
  geom_polygon(fill = coloUR["A"], color = coloUR["B"]) +
  coord_cartesian(xlim = c(10, 50)) +
  scale_x_continuous(breaks = seq(10, 50, by = 5)) +
  ggtitle(paste("Age-specific fertility rates, U.S. population of",year))
```



Exercise 3: Fertility and reproduction question (Optional)

Estimate the TFR and NRR for all available years, and address the following questions.

In which year after World War II did fertility first fall below replacement level? Between 1933 and 1941, the Total Fertility Rate exceeded two children per woman, yet the net reproduction rate remained below replacement. What accounts for this discrepancy?
Hint: Use a loop to consolidate the ${}_nL_x$ and ${}_nF_x$ matrices. Estimate all years simultaneously using a single line of code.

```r
rm(year, B, nWx, nFx, nLx, nNx_t, age, coloUR)
data                    = read.csv(GitHub)
data[,"nFx"]            = data[, "B"]/data[, "nWx"]
data                    = data[, c("Year", "x", "nLx", "nFx")]
Year                    = min(data[, "Year"]):max(data[, "Year"])
nFx                     = matrix(NA, nrow = length(x), ncol = length(Year))
nLx                     = nFx

for (i in 1:length(Year)) {
  temp     = data[data[, "Year"] == Year[i],]
  nFx[, i] = temp[, "nFx"]
  nLx[, i] = temp[, "nLx"]
}

TFR                     = colSums(n[rEP]*nFx[rEP, ])
NRR                     = 1/(1 + SRB)*colSums(nLx[rEP, ]*nFx[rEP, ])/radix
data.frame(Year,TFR,NRR)
```

```
##    Year      TFR       NRR
## 1  1933 2.009101 0.8660717
## 2  1934 2.069206 0.8858802
## 3  1935 2.034968 0.8801763
## 4  1936 2.004163 0.8661759
## 5  1937 2.035524 0.8852864
## 6  1938 2.089386 0.9176063
## 7  1939 2.047230 0.9084113
## 8  1940 2.109836 0.9391986
## 9  1941 2.222468 0.9928554
## 10 1942 2.461405 1.1087511
## 11 1943 2.562626 1.1564747
## 12 1944 2.435340 1.1020699
## 13 1945 2.376299 1.0792020
## 14 1946 2.826084 1.2869150
## 15 1947 3.156829 1.4526737
## 16 1948 3.004261 1.3862807
## 17 1949 3.017361 1.3952739
## 18 1950 3.015404 1.3995139
## 19 1951 3.197593 1.4855177
## 20 1952 3.297118 1.5317530
## 21 1953 3.367563 1.5698037
## 22 1954 3.488195 1.6308039
## 23 1955 3.533094 1.6536188
## 24 1956 3.648612 1.7092480
## 25 1957 3.733039 1.7476092
## 26 1958 3.684459 1.7261337
## 27 1959 3.684052 1.7276372
## 28 1960 3.662462 1.7184322
## 29 1961 3.621806 1.7018299
## 30 1962 3.476202 1.6334458
## 31 1963 3.349723 1.5738140
## 32 1964 3.217536 1.5120297
## 33 1965 2.921751 1.3742903
## 34 1966 2.709131 1.2747869
## 35 1967 2.559303 1.2061454
```

```
## 36 1968 2.462591 1.1605373
## 37 1969 2.451865 1.1561288
## 38 1970 2.455020 1.1587006
## 39 1971 2.262555 1.0699903
## 40 1972 2.002381 0.9481508
## 41 1973 1.864886 0.8837210
## 42 1974 1.821272 0.8642488
## 43 1975 1.762754 0.8375206
## 44 1976 1.733044 0.8241709
## 45 1977 1.777114 0.8459729
## 46 1978 1.741913 0.8295590
## 47 1979 1.789133 0.8528366
## 48 1980 1.815950 0.8659849
## 49 1981 1.800446 0.8595831
## 50 1982 1.809074 0.8644063
## 51 1983 1.778262 0.8502742
## 52 1984 1.787236 0.8546582
## 53 1985 1.829578 0.8752602
## 54 1986 1.829175 0.8750772
## 55 1987 1.858702 0.8892794
## 56 1988 1.915079 0.9160461
## 57 1989 1.990930 0.9522623
## 58 1990 2.060319 0.9865910
## 59 1991 2.049224 0.9817278
## 60 1992 2.034761 0.9755713
## 61 1993 2.010748 0.9638495
## 62 1994 1.994406 0.9563501
## 63 1995 1.973235 0.9465882
## 64 1996 1.972377 0.9466551
## 65 1997 1.967362 0.9446357
## 66 1998 1.996580 0.9588517
## 67 1999 2.002788 0.9619934
## 68 2000 2.048498 0.9842003
## 69 2001 2.027256 0.9741579
## 70 2002 2.020739 0.9708767
## 71 2003 2.050055 0.9849386
## 72 2004 2.054198 0.9869763
## 73 2005 2.057718 0.9887432
## 74 2006 2.108298 1.0130711
## 75 2007 2.119171 1.0183673
## 76 2008 2.070514 0.9955543
## 77 2009 1.999248 0.9614776
## 78 2010 1.922244 0.9249420
## 79 2011 1.886560 0.9077505
## 80 2012 1.872230 0.9008819
## 81 2013 1.849211 0.8898497
## 82 2014 1.861823 0.8958467
## 83 2015 1.843556 0.8865015
## 84 2016 1.815424 0.8723612
## 85 2017 1.763188 0.8473031
## 86 2018 1.726019 0.8296433
## 87 2019 1.700519 0.8174342
## 88 2020 1.637163 0.7860198
## 89 2021 1.655791 0.7935678
```

```
## 90 2022 1.647824 0.7903456
## 91 2023 1.600243 0.7683198
```