Mortality Investigation Report

Moffat Kagiri

2025-08-26

# Introduction

This is an exercise to study mortality investigation. It is executed in R, validated against the answers provided in the worked example, and documented in the actuarial work repository on github. The goal is to:

Calculate crude and graduated mortality rates

Compare actual vs expected experience

Provide diagnostic plots

Document the process as a reliable reference point in future

# Data Preparation

The data is availed as a .csv file named FuneralData.csv Let’s do an exploratory analysis.

main <- read.csv("C:/Users/moffat.kagiri/models/mortality investigation/FuneralData.csv", header = TRUE)  
head(main)

## LIFE BIRTH ENTRY DEATH  
## 1 1 1944.089 2014.183 2017.358  
## 2 2 1946.778 2017.050 2017.683  
## 3 3 1946.931 2012.344 NA  
## 4 4 1953.486 2012.322 2016.317  
## 5 5 1967.156 2017.192 NA  
## 6 6 1957.681 2017.025 NA

summary(main)

## LIFE BIRTH ENTRY DEATH   
## Min. : 1.0 Min. :1936 Min. :2010 Min. :2011   
## 1st Qu.: 250.8 1st Qu.:1950 1st Qu.:2012 1st Qu.:2015   
## Median : 500.5 Median :1958 Median :2014 Median :2017   
## Mean : 500.5 Mean :1956 Mean :2014 Mean :2016   
## 3rd Qu.: 750.2 3rd Qu.:1963 3rd Qu.:2016 3rd Qu.:2018   
## Max. :1000.0 Max. :1968 Max. :2018 Max. :2018   
## NA's :847

Start date t\_0 = 2013.00 and end date t\_4 2018. Therefore, we can define age relative to each year in the study, for each life.

# Define the start and end dates as per the format in the data  
t\_0 <- round(decimal\_date(ymd("2013-01-01")), 4)  
t\_5 <- 2017.999 #round(decimal\_date(ymd("2017-12-31")), 4)  
print(c(t\_0, t\_5))

## [1] 2013.000 2017.999

# Exact Central Exposed to Risk

The three dates whose maximum determines the date at which exposure at age 70 begins are:

* The date of reaching age label 70
* The date of entry
* The start of the investigation

The three dates whose minimum determines the date at which exposure at age 70 ends are:

* The date of reaching age label 71
* The date of exit (for any reason)
* The end of the investigation

To determine the exact central exposed to risk for age 70 last birthday, the block below manually determines the exposure as the duration during which lives under observation have age label 70. This is the intersection of the interval between a life’s 70th and 71st birthdays, and the period of the investigation. This is calculated for every life and then added up to show the total exposure as needed.

An important takeaway in this exercise was to avoid the use of ifelse with date time formats. This loop strips the Date/POSIXct attributes as well as the interval class attributes, making the entire exercise overly frustrating.

# Determine exact central exposed to risk for age 70 last birthday.   
# Define 70th and 71st birthdays  
exact <- main[1:4]  
exact$birthday\_70 <- date\_decimal(main$BIRTH) + years(70)  
exact$birthday\_71 <- date\_decimal(main$BIRTH) + years(71)  
  
# Define other dates  
exact$entry\_date <- date\_decimal(main$ENTRY)  
exact$start\_date <- date\_decimal(t\_0)   
  
# Replace NA in DEATH with t\_5, keeping Date format  
exact$exit\_date <- as.Date(NA) # Initialize as Date  
exact$exit\_date <- if\_else(  
 is.na(main$DEATH),  
 date\_decimal(t\_5), # Use study end date   
 date\_decimal(main$DEATH) # Keep original death date  
)  
  
exact$end\_date <- date\_decimal(t\_5)  
  
# Calculate exposure using proper date-preserving methods  
exact <- exact %>%  
 mutate(  
 # Step 1: Define exposure windows (already correct)  
 exposure\_start = pmax(birthday\_70, entry\_date, start\_date),  
 exposure\_end = pmin(birthday\_71, exit\_date, end\_date),  
   
 # Step 2: Compute exposure time safely  
 EXPOSURE\_AGE70 = case\_when(  
 exposure\_start >= exposure\_end ~ 0, # No overlap  
 TRUE ~ round(time\_length(interval(exposure\_start, exposure\_end), "years"), 4)  
 )  
 )  
  
# Sum exposure (now numeric, not interval)  
total\_E70 <- round(sum(exact$EXPOSURE\_AGE70, na.rm = TRUE), 4)  
print(paste("Total central exposed to risk at age 70:", round(total\_E70, 4), "person-years"))

## [1] "Total central exposed to risk at age 70: 70.4461 person-years"

Without the date-time conversions, this is an attempt at a more reproducible calculation, factoring out the age parameter as a variable.

Note to maintain the accuracy indicated in the question. The answer swayed between 70.401 and 70.462 due to rounding off errors.

# Determine exact central exposed to risk for age 70 last birthday.   
# Define 70th and 71st birthdays  
  
x <- 70  
exact$birthday\_x <- main$BIRTH + x  
exact$birthday\_x1 <- main$BIRTH + x + 1  
  
# Define other dates  
exact$entry\_date <- main$ENTRY  
exact$start\_date <- t\_0   
  
# Replace NA in DEATH with t\_5, keeping Date format  
exact$exit\_date <- NA # Initialize as Date  
exact$exit\_date <- if\_else(  
 is.na(main$DEATH),  
 t\_5, # Use study end date   
 main$DEATH # Keep original death date  
)  
  
exact$end\_date <- t\_5  
  
# Step 1: Define exposure windows (already correct)  
exact$exposure\_start <- pmax(exact$birthday\_x, exact$entry\_date, exact$start\_date)  
exact$exposure\_end <- pmin(exact$birthday\_x1, exact$exit\_date, exact$end\_date)  
  
# Step 2: Compute exposure time safely  
exact$EXPOSURE\_AGEX <- case\_when(  
 exact$exposure\_start >= exact$exposure\_end ~ 0, # No overlap  
 TRUE ~ round(exact$exposure\_end - exact$exposure\_start, 6))  
  
# Sum exposure (now numeric, not interval)  
total\_EX <- sum(exact$EXPOSURE\_AGEX, na.rm = TRUE)  
print(paste("Total central exposed to risk at age 70:", round(total\_EX, 4), "person-years"))

## [1] "Total central exposed to risk at age 70: 70.444 person-years"

The solution provided in the worked example is:

# Determining

Calculate the number of lives who died at age 70 last birthday during the investigation period. The data includes dates of death, enabling us to determine the lives who died. From the dates of death, we can determine the age last birthday for each life at the point of death during the study.

# Number of lives who died during observation  
deaths <- main[!is.na(main$DEATH) &  
 main$DEATH >= 2013.000 &  
 main$DEATH <= 2017.999, 1:4]  
summary(deaths)

## LIFE BIRTH ENTRY DEATH   
## Min. : 1.0 Min. :1937 Min. :2010 Min. :2013   
## 1st Qu.:227.0 1st Qu.:1945 1st Qu.:2012 1st Qu.:2015   
## Median :476.0 Median :1953 Median :2014 Median :2017   
## Mean :482.4 Mean :1953 Mean :2014 Mean :2016   
## 3rd Qu.:741.0 3rd Qu.:1962 3rd Qu.:2015 3rd Qu.:2017   
## Max. :997.0 Max. :1966 Max. :2018 Max. :2018

# Determine age last birthday at death  
deaths$AGE <- floor(deaths$DEATH - deaths$BIRTH)  
  
# Number of lives who died at age 70 last birthday is:  
num = sum(deaths$AGE == 70)  
  
print(num)

## [1] 3

# Force of mortality

Force of mortality at age 70 is given by:

mu\_x = num/total\_EX  
print(mu\_x)

## [1] 0.04258702

## Number of lives aged 70 last birthday

To determine the number of lives aged 70 whose policies were in force at the start of the investigation period, we check age last birthday at the start of the investigation period.

Filter for: 1. Either still alive (NA in DEATH) OR died after investigation start (DEATH >= t\_0) 2. AND was age 70 last birthday at ENTRY 3. AND ENTRY was before/at investigation start (t\_0)

# Define age last-birthday at the start of the investigation  
agetest <- floor(t\_0 - main$BIRTH)  
p\_70\_start <- main[agetest == 70 &  
 main$ENTRY < t\_0 &  
 (is.na(main$DEATH) | main$DEATH > t\_0), 1:4]

Determine the corresponding figures for 1 January in years 2014, 2015, 2016, 2017 and 2018. For this, we replicate the code above to regenerate the same figure for different years.

# Replicating to get the same figure for years from 2013 to 2018  
  
p\_70 <- function(date\_0){  
 agex <- floor(date\_0 - main$BIRTH)  
 lives <- main[  
 (is.na(main$DEATH) | main$DEATH > date\_0) &   
 agex == 70 &  
 main$ENTRY <= date\_0,  
 1:4  
 ]  
 return(nrow(lives))  
}  
  
  
# Create a vector of start dates for each year  
year\_starts <- decimal\_date(ymd(paste0(2013:2018, "-01-01")))  
  
# Calculate and store results  
results <- sapply(year\_starts, p\_70)  
  
# Print the results  
print("The number of lives aged 70 whose policies were in force at the beginning of each year are:")

## [1] "The number of lives aged 70 whose policies were in force at the beginning of each year are:"

for (i in 1:length(results)) {  
 print(paste(year(ymd(paste0(2012+i, "-01-01"))), ":", results[i]))  
}

## [1] "2013 : 12"  
## [1] "2014 : 9"  
## [1] "2015 : 18"  
## [1] "2016 : 14"  
## [1] "2017 : 14"  
## [1] "2018 : 19"

# Census Approach to Find Central Exposed to Risk

Estimate the central exposed to risk for age 70 last birthday for the investigation period using a census approach. For a population observed between time and :

where:

= Number of lives aged at time

Integrate over the observation period.

In our case,

Thus

This becomes:

Therefore:

And

Trying the trapezoid rule in R…

# Calculate central exposed to risk (census trapezoidal rule)  
ce\_x <- function(dates, populations) {  
 time\_intervals <- as.numeric(diff(dates)/365.25)  
 avg\_populations <- (populations[-1] + populations[-length(populations)]) / 2  
 sum(time\_intervals \* avg\_populations)  
}  
  
# Example usage  
c\_dates <- as.Date(date\_decimal(year\_starts))  
populations <- results

We find that: equals 70.4900753.

# Estimated Force of Mortality (census method)

As before, force of mortality is given by:

Which translates to:

Therefore:

# Calculating for All Ages

To find the force of mortality, for all ages, we scale out the functions we used previously for the exact method.

# Determine exact central exposed to risk for age 70 last birthday.   
# Define Minimum and maximum ages  
  
mort <- main %>%  
 mutate(  
 BIRTH = as.numeric(BIRTH),  
 entry\_date = as.numeric(ENTRY),  
 exit\_date = ifelse(is.na(DEATH), as.numeric(t\_5), as.numeric(DEATH)),  
 start\_date = as.numeric(t\_0),  
 end\_date = as.numeric(t\_5)  
 )  
  
# Age at start and at end (in completed years)  
age\_at\_start <- floor(mort$start\_date - mort$BIRTH)  
age\_at\_end <- floor(mort$end\_date - mort$BIRTH)  
  
min\_age <- max(0, min(age\_at\_start, na.rm = TRUE))  
max\_age <- max(age\_at\_end, na.rm = TRUE)  
ages <- min\_age:max\_age  
  
# --- Compute central exposed (E\_x^c) by age using person×age expansion ---  
exposure <- mort %>%  
 crossing(age = ages)%>%  
 mutate(  
 bday = BIRTH + age,  
 bday1 = bday + 1,  
 start = pmax(bday, entry\_date, start\_date),  
 end = pmin(bday1, exit\_date, end\_date),  
 exposure = as.numeric(end - start),  
 exposure = if\_else(exposure < 0, 0, exposure)  
 )  
  
E\_by\_age <- exposure %>%  
 group\_by(age) %>%  
 summarise(E\_central = sum(exposure, na.rm = TRUE), .groups = "drop")  
  
  
# Determine age last birthday at death  
deaths\_tbl <- main %>%  
 filter(!is.na(DEATH), DEATH >= t\_0, DEATH <= t\_5) %>%  
 mutate(age\_at\_death = floor(DEATH - BIRTH)) %>%  
 count(age\_at\_death, name = "Deaths") %>%  
 rename(age = age\_at\_death)  
  
# join counts and exposure, fill zeros where missing  
mortality <- tibble(age = ages) %>%  
 left\_join(E\_by\_age, by = "age") %>%  
 left\_join(deaths\_tbl, by = "age") %>%  
 replace\_na(list(E\_central = 0, Deaths = 0)) %>%  
 mutate(  
 # central death rate m\_x (per year)  
 # eliminate E\_c = 0 to avoid division by 0.   
 m\_x = if\_else(E\_central > 0, Deaths / E\_central, NA\_real\_),   
   
 # approx standard error for rate (Poisson approx)  
 se\_mx = if\_else(E\_central > 0, sqrt(Deaths) / E\_central, NA\_real\_),  
  
 # 95% CI on m\_x (approx)  
 lower95 = if\_else(!is.na(m\_x), m\_x - 1.96 \* se\_mx, NA\_real\_),  
 upper95 = if\_else(!is.na(m\_x), m\_x + 1.96 \* se\_mx, NA\_real\_)  
 )  
  
print(mortality)

## # A tibble: 39 × 7  
## age E\_central Deaths m\_x se\_mx lower95 upper95  
## <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 44 0 0 NA NA NA NA   
## 2 45 0 0 NA NA NA NA   
## 3 46 0 0 NA NA NA NA   
## 4 47 0 0 NA NA NA NA   
## 5 48 0 0 NA NA NA NA   
## 6 49 0 0 NA NA NA NA   
## 7 50 93.7 3 0.0320 0.0185 -0.00421 0.0682  
## 8 51 166. 4 0.0241 0.0120 0.000481 0.0476  
## 9 52 203. 7 0.0345 0.0130 0.00894 0.0601  
## 10 53 220. 6 0.0272 0.0111 0.00544 0.0490  
## # ℹ 29 more rows

Plotting this out on a graph, we get:

ggplot(mortality %>% filter(!is.na(m\_x)), aes(x = age, y = m\_x)) +  
 geom\_ribbon(aes(ymin = lower95, ymax = upper95), alpha = 0.2, fill = "grey70") +  
 geom\_point() +  
 geom\_line() +  
 labs(x = "Age", y = expression(m[x]~"(per year)"),  
 title = "m\_x with approximate 95% CIs") +  
 theme\_minimal()

