Life Insurance

2023-01-22

Cash Flows

Scenario 1

Suppose you want to go on a trip 5 years from now. You have expenses of 3000 each year for 4 years (including this year), followed by an expense of 4000 and a final expense of 5000. How much would you have to deposit at this moment in time to cover all of these future payments? Assume a constant interest rate of 2.14% and payments would take place at the beginning of the year.

```r
# We have an expense of 3000 from t=0 to t=3, an expense of 4000 at t=4 and 5000 at t=5

# Cash flows
cash_flows <- c(rep(3000,4), 4000, 5000)

# Interest Rate
i <- 0.0214

# Discount factor
v <- 1 / (1+i)

# Discount factors
discount_factors <- v ^ (0:5)

# Present value
present_val <- sum(cash_flows * discount_factors)
present_val
```

```
## [1] 19800.98
```

So you would have to make a deposit of 19800.98 at this current time to cover all of these future payments.

Scenario 2

You're the head of a lawn mowing company. You want to invest in a new lawn mower than you are confident will generate cash flows of 50 each month for 3 months, followed by 20 for 2 months. The issue is it costs 185 and you're unsure if it will be a profitable investment. To find out if it is a good investment or not, you must calculate the NPV for this investment. Assume the monthly interest rate is 0.6%.

```r
# Define the cash flows
# We have a negative cash flow at t=0 of 500, and 5 positive cash flows of 50 from t=1 to t=5

cash_flows <- c(0, rep(50,3), rep(30,2))

# Discount factors
discount_factors <- (1+0.006) ^ - (0:5)

# NPV
```

```r
npv <- sum(cash_flows * discount_factors) - 185
npv
```

```
## [1] 21.62449
```

Since our NPV is positive, this indicates that this would be an overall gain ie profit.

Scenario 3

Suppose instead of making those expenses, you chose to put them into a savings account. What would be your new present value, assuming same interest rate. What would be your new accumulated value?

```r
# Cash flows
cash_flows <- c(rep(3000,4), 4000, 5000)

# Define the discount function
discount <- function(s, t, i = 0.0214) {
  (1 + i) ^ - (t - s)
}

# Calculate the pv
pv <- sum(cash_flows * discount(0, 0:5))
pv
```

```
## [1] 19800.98
```

```r
# Calculate the value at time 6
av <- sum(cash_flows * discount(6, 0:5))

# Calculate the value at time 6, starting from present_value
pv * discount(6, 0)
```

```
## [1] 22483.39
```

```r
print(av - pv)
```

```
## [1] 2682.41
```

As you can see, there would be a difference of $2682.41 if you saved the money instead of spending it.

Scenario 4

Your parents want to start saving money for your university costs. Each year of study will set you guys back by about 7500. You're currently 14 (t=0) and will start attending when you are 19 (t=5). Therefore, your parents will make 1 deposit each year for 4 years (t=1 to t=4). Assume you will attend for 4 years (t=5 to t=22), the constant annual interest rate is 3.14% and each deposit your parents make is equal. Find the amount of the deposit

```r
# discount factors
discount_factors <- (1 + 0.0314) ^ - (0:8)

# deposit pattern
deposits <- c(0, rep(1, 4), rep(0, 4))

# university expenses
payments <- c(rep(0, 5), rep(7500, 4))

# Calculate the present value of the deposits
```

```r
PV_deposit <- sum(deposits * discount_factors)

# Calculate the present value of the payments
PV_payment <- sum(payments * discount_factors)

# Calculate the yearly deposit K in the first 4 years
yearly_deposit <-  PV_payment / PV_deposit
yearly_deposit
```

```
## [1] 6627.546
```

So your parents would have to make a deposit of 6627.546 for each of the 4 years to finance your university.

Scenario 5

You want to take out a loan to open a new restaurant. You plan to take 1000 this year and 3000 next year. You plan to repay the money with equal yearly payments for the 10 years after next year (t=2 to t=11). Note this time the interest rate isn't constant but changes over time. For the first 3 years it is 4%, then for the next 3 years it is 4.5%, then for the final 5 years it is 6%. How much will your yearly payments be?

```r
# Interest rates
interest <- c(rep(0.04, 3), rep(0.045, 3), rep(0.06, 5))

# Yearly discount factors
yearly_discount_factors <- (1 + interest) ^ (-1)

# Discount factors
discount_factors <- cumprod(c(1, yearly_discount_factors))

# Cash flows for the first two years
cash_flow <- c(1000, 3000)

# Calculate the present value (PV) of the loan
PV_loan <- sum(cash_flow * discount_factors[1:2])

# Calculate the present value (PV) of the repayments
repayment_years <- 2:11
discount_factors_repayments <- discount_factors[repayment_years + 1]
n_payments <- length(discount_factors_repayments)

# Define the yearly payment variable
yearly_payment <- PV_loan / sum(discount_factors_repayments)

# Create the full cash flow vector with the repayments
cash_flow_full <- c(cash_flow, rep(-yearly_payment, n_payments))

# Calculate the PV of the repayments to check the balance
PV_repayments <- sum(cash_flow_full * discount_factors)

# Display the yearly payment and the PV balance check
yearly_payment
```

```
## [1] 515.2757
```

```r
PV_repayments
```

```
## [1] -1.705303e-13
```

Our PV is ~0, meaning the PV of the future payments and PV of your loans are equal, hence the yearly payments would be 515.2757

Scenario 6

You plan to take out a 100,000 loan with a yearly interest rate of 2.88%. You want to pay back this loan with fixed monthly payments over the next 15 years. Find how much these monthly payments will be.

```
#number of payments
number_payments <- 12*15

#yearly interest rate
i <- 0.0288

#monthly interest rate

monthly_interest <- (1+i)^(1/12)-1

# Define the discount factors
discount_factors <- (1 + monthly_interest) ^ - (1:number_payments)

# Define the payment pattern
payments <- rep(1, number_payments)

# Calculate the monthly loan payment K
K <- 100000 / sum(payments * discount_factors)
K
```

```
## [1] 683.0401
```

These monthly payments will be 683.0401

Life Tables

Import the life tables from mortality.org. We are going to import the dataset of Canada from 1921 - 2021.

```
life_table <- read.table("C:\\Users\\megacrazyleo\\Desktop\\SQL\\R\\bltper_1x1.txt", header=FALSE, skip
colnames(life_table) <- c("Year", "Age", "mx", "qx", "ax", "lx", "dx", "Lx", "Tx", "ex")
```

Setting up variables

```
year <- life_table$Year
age <- life_table$Age
qx <- life_table$qx
mx <- life_table$mx
ax <- life_table$ax
lx <- life_table$lx
dx <- life_table$dx
Lx <- life_table$Lx
Tx <- life_table$Tx
ex <- life_table$ex
px <- 1 - qx
```

Year: Year or range of years (for both period & cohort data)

Age: Age group for n-year interval from exact age x to just before exact age x+n, where n=1, 4, 5, or infinity

m(x) Central death rate between ages x and x+n

q(x) Probability of death between ages x and x+n

a(x) Average length of survival between ages x and x+n for persons dying in the interval

l(x) Number of survivors at exact age x, assuming l(0) = 100,000

d(x) Number of deaths between ages x and x+n

L(x) Number of person-years lived between ages x and x+n

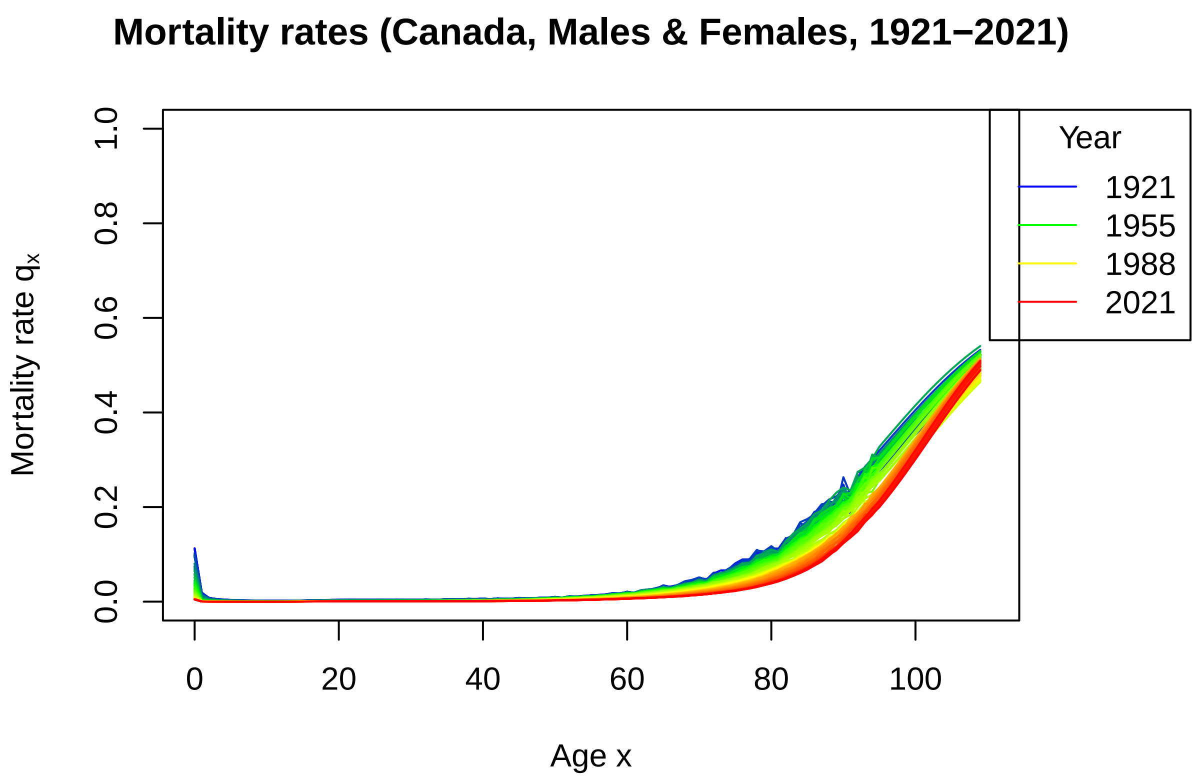
T(x) Number of person-years remaining after exact age x

e(x) Life expectancy at exact age x (in years)

Mortality Rates

```r
# Setting up the plot
par(xpd = TRUE, mar = c(5, 4, 4, 6) + 0.1) # To ensure plot doesn't cover data

convert_age <- function(x) {
  x[x == "110+"] <- "110"  # Age is character in our dataset, need to change to numeric
  suppressWarnings({
    as.numeric(x)
  })
}
age <- convert_age(age)

# Create a color palette
years <- unique(year)
color_palette <- colorRampPalette(c("blue", "green", "yellow", "red"))(length(years))

# Initialize plot
plot2 <- plot(NULL,
              xlim = range(age, na.rm = TRUE),  # Exclude NAs from range calculation
              ylim = range(qx, na.rm = TRUE),   # Exclude NAs from range calculation
              main = "Mortality rates (Canada, Males & Females, 1921-2021)",
              xlab = "Age x",
              ylab = expression(paste("Mortality rate ", q[x])),
              type = "l"
)

# Plot each year's data
suppressWarnings({
  for (i in seq_along(years)) {
    year_data <- subset(life_table, Year == years[i])
    lines(year_data$Age, year_data$qx, col = color_palette[i], type = "l")
  }
```

```r
})

# Colored legend
legend("topright", inset = c(-0.2,0),
       legend = c("1921", "1955", "1988", "2021"),
       col = color_palette[c(1, which.min(abs(years - 1955)),
                             which.min(abs(years - 1988)), length(years))],
       lty = 1,
       title = "Year")
```

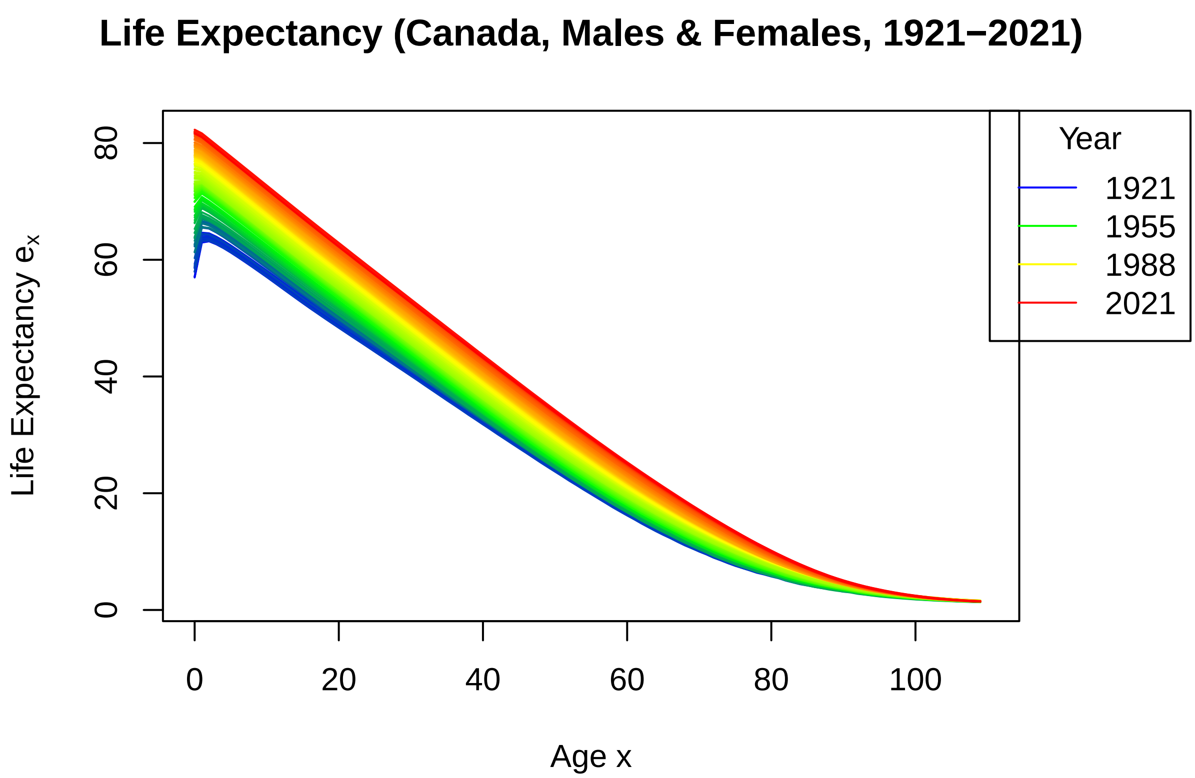


We can see that mortality rates from recent years are much lower compared to the earliest years.

Life Expectancy

```r
# Setting up the plot
par(xpd = TRUE, mar = c(5, 4, 4, 6) + 0.1) # To ensure plot doesn't cover data

# Create a color palette
years <- unique(year)
color_palette <- colorRampPalette(c("blue", "green", "yellow", "red"))(length(years))

plot3 <- suppressWarnings({
  plot(NULL,
       xlim = range(age, na.rm = TRUE),  # Exclude NAs from range calculation
       ylim = range(ex, na.rm = TRUE),   # Exclude NAs from range calculation
       main = "Life Expectancy (Canada, Males & Females, 1921-2021)",
```

```
       xlab = "Age x",
       ylab = expression(paste("Life Expectancy ", e[x])),
       type = "l"
  )
})

# Plot each year's data
suppressWarnings({
  for (i in seq_along(years)) {
    year_data <- subset(life_table, Year == years[i])
    lines(year_data$Age, year_data$ex, col = color_palette[i], type = "l")
  }
})

# Colored legend
legend("topright", inset = c(-0.2, 0),
       legend = c("1921", "1955", "1988", "2021"),
       col = color_palette[c(1, which.min(abs(years - 1955)),
                             which.min(abs(years - 1988)), length(years))],
       lty = 1,
       title = "Year")
```



As we can see, the life expectancy is much higher from the years ~ 2021 as opposed to the years from ~1921 or even ~1955. Note for the early years, the life expectancy starts extremely low at age 0 and after a couple years, has a sharp increase. This can be explained mainly because it was much more difficult for an infant to survive the initial years. In later years, thanks to modern medicine, the survival rate of infants have risen much higher.

Let us focus on the year 2021.

2021 Life Table

```
life_table <- life_table[year == 2021, ]  #Getting only data from the year 2021
year <- life_table$Year
age <- life_table$Age
qx <- life_table$qx
mx <- life_table$mx
ax <- life_table$ax
lx <- life_table$lx
dx <- life_table$dx
Lx <- life_table$Lx
Tx <- life_table$Tx
ex <- life_table$ex
px <- 1 - qx
```

###Finding the probability that an 23 year old dies before turning 24

```
qx[23 + 1]
```

```
## [1] 0.00072
```

As we can see, it is a very low probability, < 0.098%

The expected future lifetime of an 23 year old

```
ex[23 + 1]
```

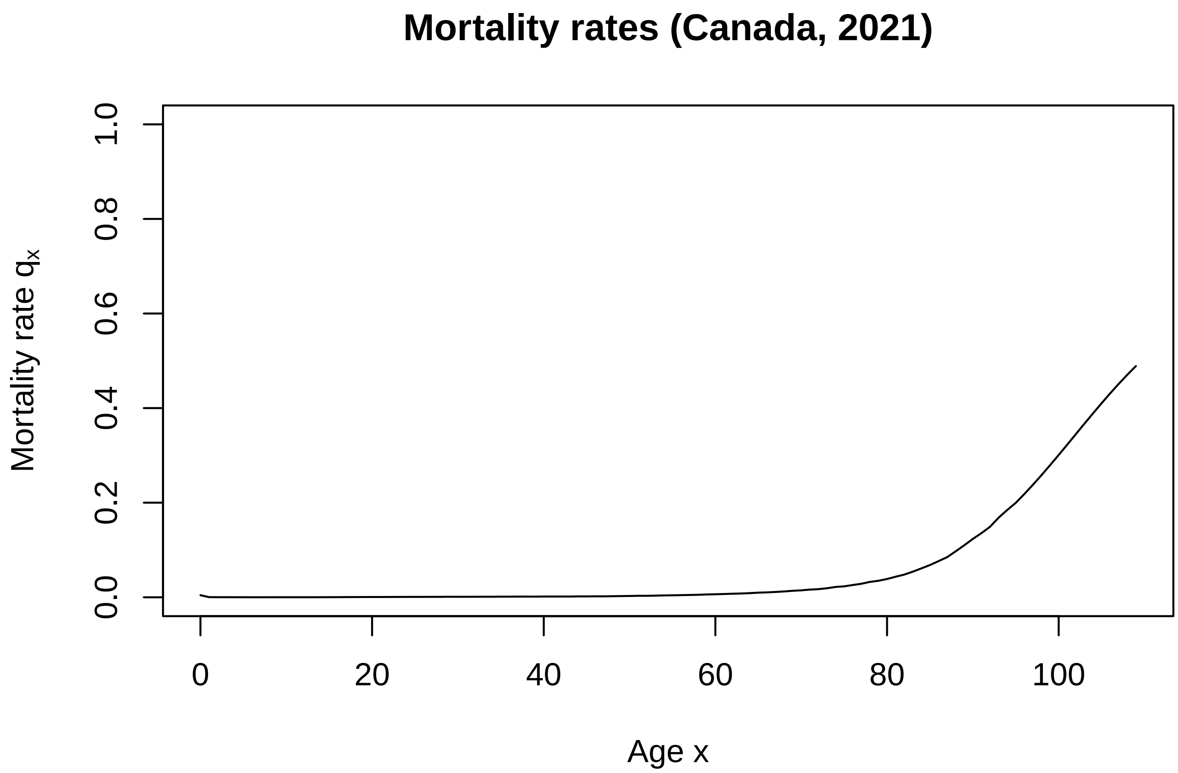
```
## [1] 59.34
```

It is expected that an average 23 year old from Canada will live for 57.06 more years in Canada in 2021

Plot the mortality rates in the year 2021

```
plot(age, qx,
    main = "Mortality rates (Canada, 2021)",
    xlab = "Age x",
    ylab = expression(paste("Mortality rate ", q[x])),
    type = "l")
```

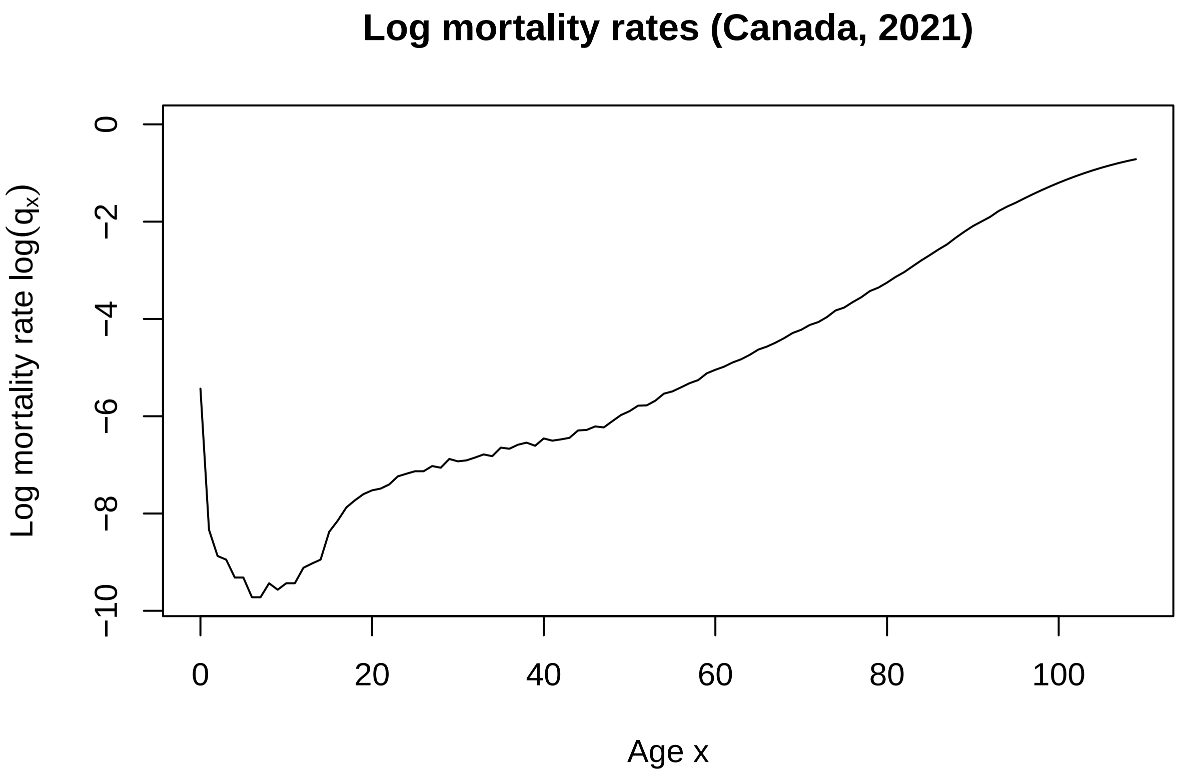
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```



Plot the logarithm of the mortality rates in the year 2021

```
# Plot the logarithm of the mortality rates in the year 2021
plot(age, log(qx),
    main = "Log mortality rates (Canada, 2021)",
    xlab = "Age x",
    ylab = expression(paste("Log mortality rate ", log(q[x]))),
    type = "l")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```



Plot the mortality rates of females vs males in the year 2021

```
#male data
life_table_m <- read.table("C:\\Users\\megacrazyleo\\Desktop\\SQL\\R\\mltper_1x1.txt", header=FALSE, sk
colnames(life_table_m) <- c("Year", "Age", "mx", "qx", "ax", "lx", "dx", "Lx", "Tx", "ex")
male_2021 <- life_table_m[life_table_m$Year == 2021, ]
year_m <- male_2021$Year
age_m <- male_2021$Age
qx_m <- male_2021$qx
mx_m <- male_2021$mx
ax_m <- male_2021$ax
lx_m <- male_2021$lx
dx_m <- male_2021$dx
Lx_m <- male_2021$Lx
Tx_m <- male_2021$Tx
ex_m <- male_2021$ex

#female data
life_table_f <- read.table("C:\\Users\\megacrazyleo\\Desktop\\SQL\\R\\fltper_1x1.txt", header=FALSE, sk
colnames(life_table_f) <- c("Year", "Age", "mx", "qx", "ax", "lx", "dx", "Lx", "Tx", "ex")
female_2021 <- life_table_f[life_table_f$Year == 2021, ]
year_f <- female_2021$Year
age_f <- female_2021$Age
qx_f <- female_2021$qx
mx_f <- female_2021$mx
```

```r
ax_f <- female_2021$ax
lx_f <- female_2021$lx
dx_f <- female_2021$dx
Lx_f <- female_2021$Lx
Tx_f <- female_2021$Tx
ex_f <- female_2021$ex
```

Import the male/female datasets

Plot male vs female mortality rates in the year 2021

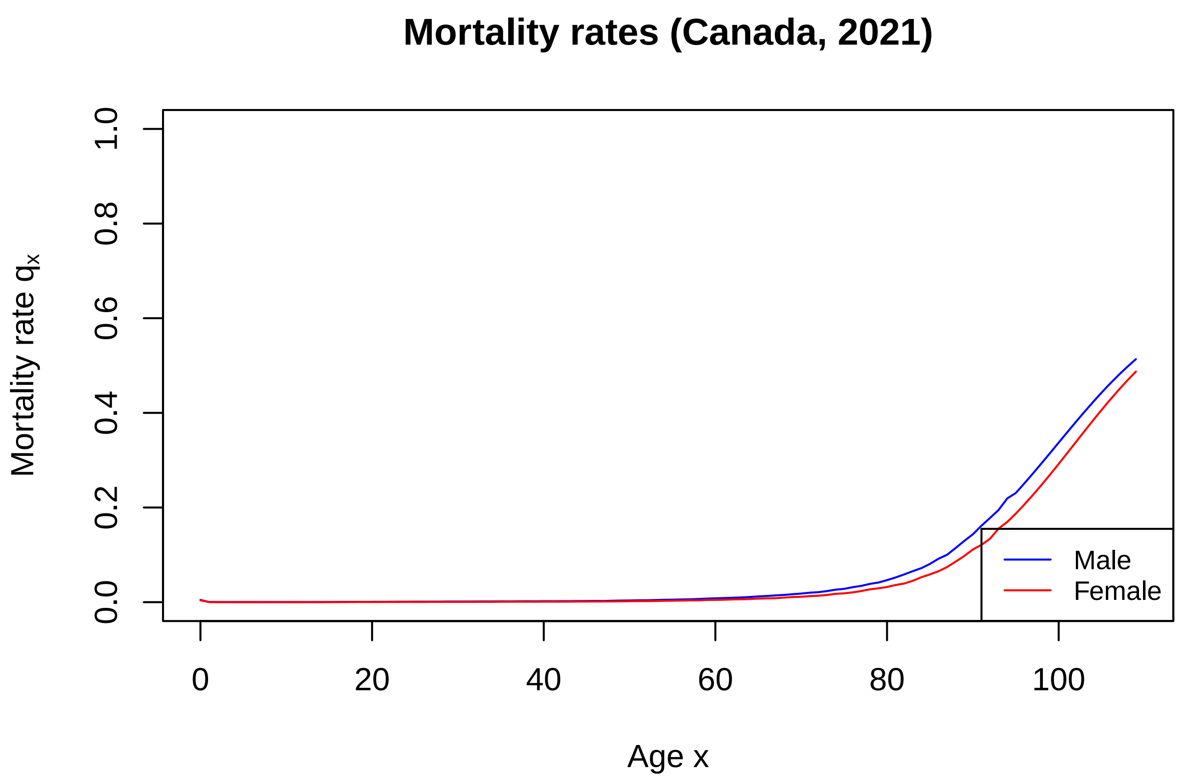
```r
plot(age_m, qx_m,
     main = "Mortality rates (Canada, 2021)",
     xlab = "Age x",
     ylab = expression(paste("Mortality rate ", q[x])),
     type = "l",
     col = "blue",
     )  # Using log scale for y-axis to better show differences
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```r
# Add the female line to the same plot
lines(age_f, qx_f, col = "red")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```

```r
legend("bottomright",
       legend = c("Male", "Female"),
       col = c("blue", "red"),
       lty = 1,
       cex = 0.8)
```




```r
# Create the plot with male data
plot(age_m, qx_m,
     main = "Mortality rates (Canadian Males and Females, 2021)",
     xlab = "Age x",
     ylab = expression(paste("Mortality rate ", q[x])),
     type = "l",
     col = "blue",
     log = "y",
     ylim = range(c(qx_m, qx_f))  # Set y-axis limits to include both male and female data
)
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```r
# Add female data
lines(age_f, qx_f, col = "red")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```

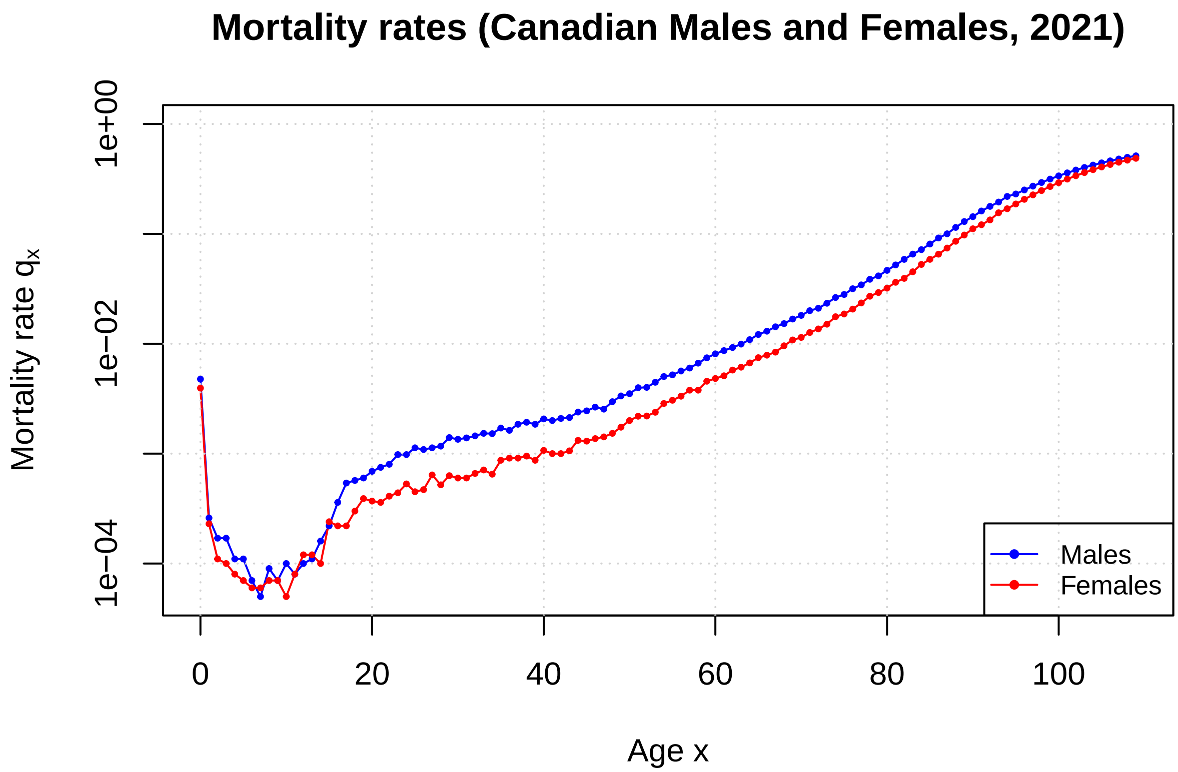
```r
grid()

# Add points to show actual data points
points(age_m, qx_m, pch = 20, cex = 0.5, col = "blue")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```

```r
points(age_f, qx_f, pch = 20, cex = 0.5, col = "red")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```

```r
legend("bottomright",
       legend = c("Males", "Females"),
       col = c("blue", "red"),
       lty = 1,
       pch = 20,
       cex = 0.8)
```



Find the probability for a 0 year old female and male in Canada to reach the age of 100 in 2021

```r
male_prob <- lx_m[100 + 1] / lx_m[0 + 1]
print(paste("Male probability:", male_prob))
```

```
## [1] "Male probability: 0.01918"
```

```r
female_prob <- lx_f[100 + 1] / lx_f[0 + 1]
print(paste("Female probability:", female_prob))
```

```
## [1] "Female probability: 0.05113"
```

Find the probability for a 0 year old in Canada to reach the age of 100 in 2021

```r
canada_prob <- lx[100 + 1] / lx[0 + 1]
print(paste("M/F probability:", canada_prob))
```

```
## [1] "M/F probability: 0.03606"
```

If we compare this with an 18 year old individual

```r
canada_prob_2 <- lx[100 + 1] / lx[18 + 1]
print(paste("M/F age 18 probability:", canada_prob_2))
```

```
## [1] "M/F age 18 probability: 0.0363050591492575"
```

Vs a 40 year old individual

```r
canada_prob_5 <- lx[100 + 1] / lx[40 + 1]
print(paste("M/F age 40 probability:", canada_prob_5))
```

```
## [1] "M/F age 40 probability: 0.0370583520029597"
```

And again with a 75 year old individual

```r
canada_prob_3 <- lx[100 + 1] / lx[75 + 1]
print(paste("M/F age 75 probability:", canada_prob_3))
```

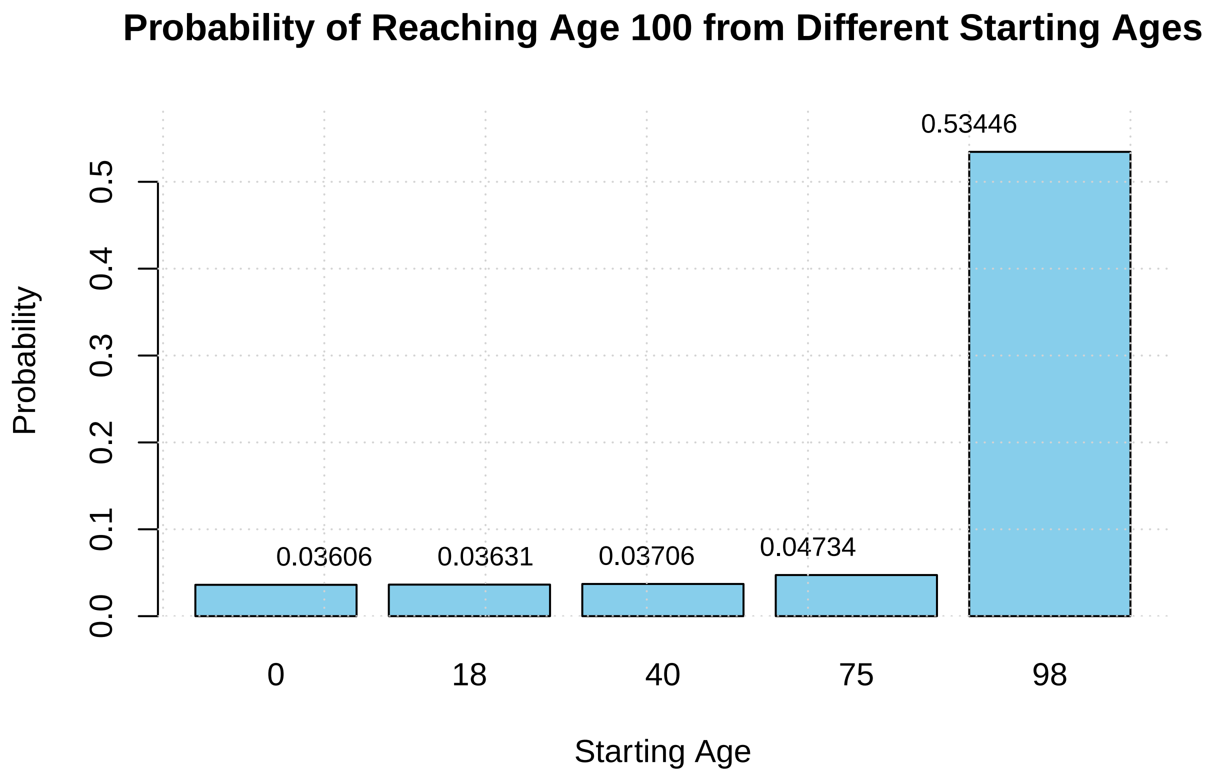
```
## [1] "M/F age 75 probability: 0.0473352585980572"
```

Finally, compare with a 98 year old individual

```r
canada_prob_4 <- lx[100 + 1] / lx[98 + 1]
print(paste("M/F age 98 probability:", canada_prob_4))
```

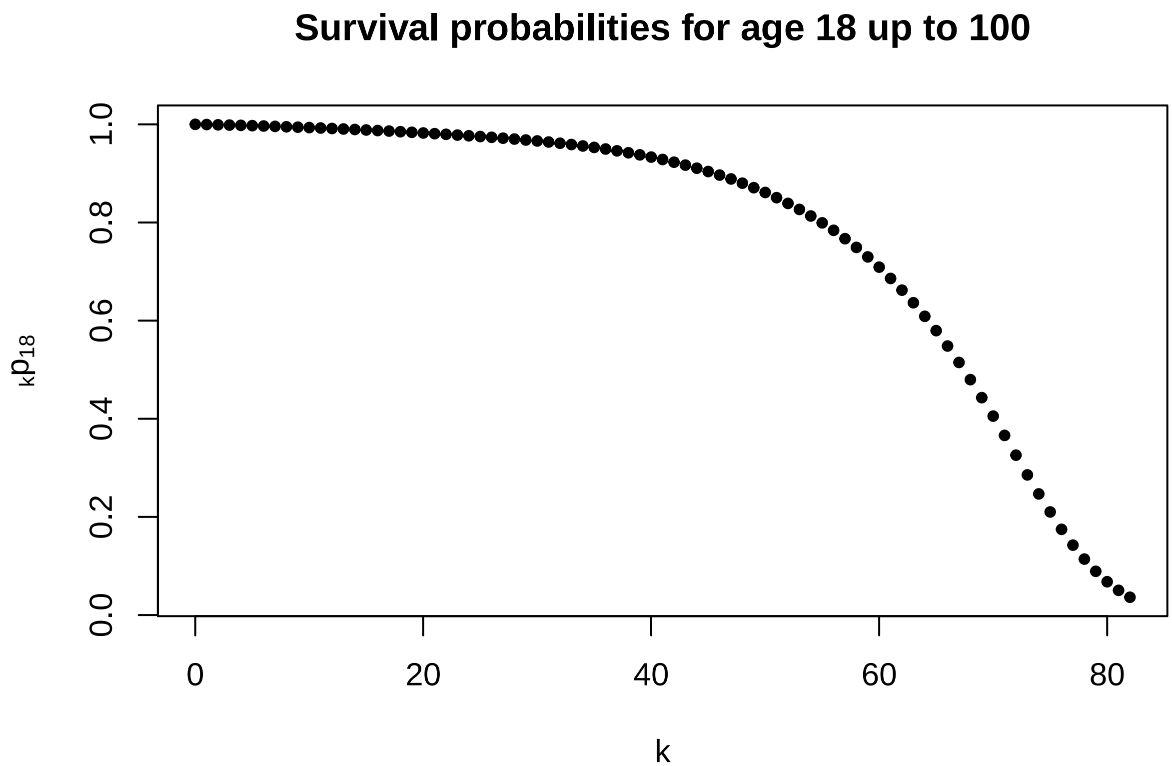
```
## [1] "M/F age 98 probability: 0.534459759893286"
```

```r
# Calculate probabilities
ages <- c(0, 18, 75, 98)
probs <- c(
  lx[100 + 1] / lx[0 + 1],
  lx[100 + 1] / lx[18 + 1],
  lx[100 + 1] / lx[40 + 1],
  lx[100 + 1] / lx[75 + 1],
  lx[100 + 1] / lx[98 + 1]
)

# Create a data frame
prob_data <- data.frame(Age = c("0", "18", "40", "75", "98"), Probability = probs)

# Create the bar plot
barplot(prob_data$Probability,
        names.arg = prob_data$Age,
        main = "Probability of Reaching Age 100 from Different Starting Ages",
        xlab = "Starting Age",
        ylab = "Probability",
        col = "skyblue",
        ylim = c(0, max(probs) * 1.1))  # Set y-axis limit to 110% of max probability

# Add value labels on top of each bar
text(x = 1:5,
     y = probs,
     labels = round(probs, 5),
     pos = 3,
     cex = 0.8)

# Add a grid for better readability
grid()
```



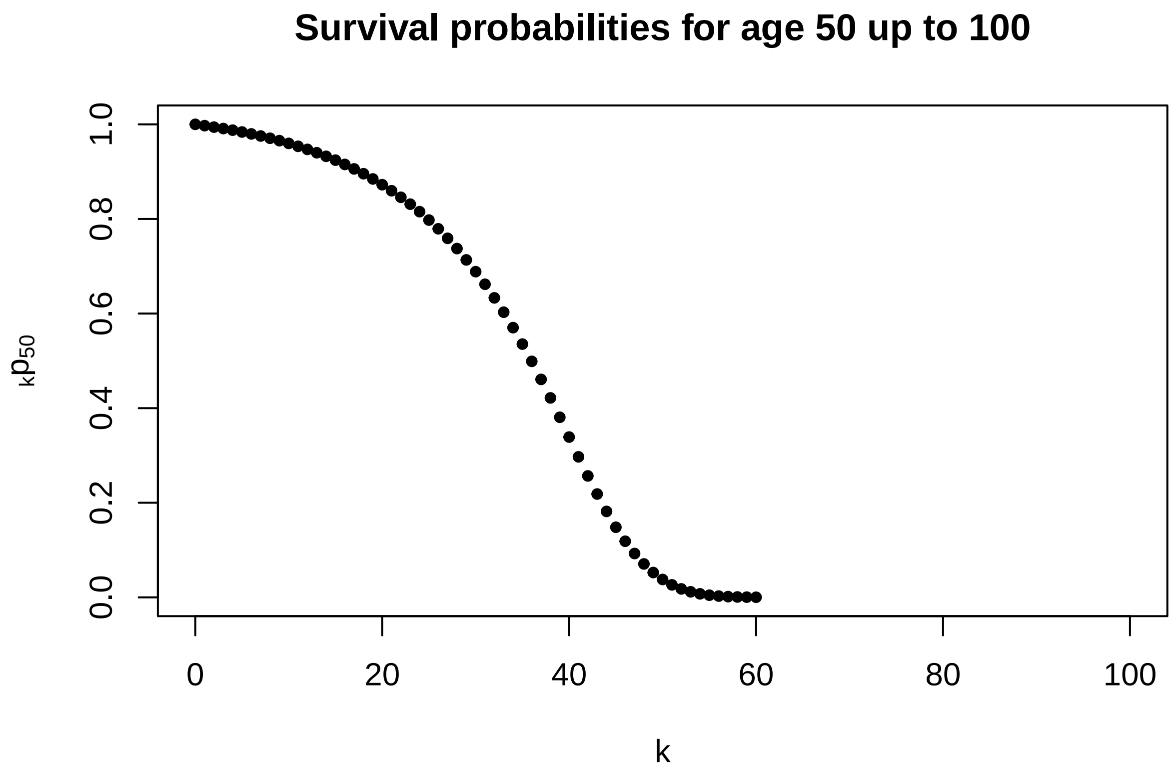
Plot the survival probabilties for (18) up to age 100

```
k <- 0:82
plot(k, lx[18 + k + 1] / lx[18 + 1],
    pch = 20,
    xlab = "k",
    ylab = expression(paste(""[k], "p"[18])),
    main = "Survival probabilities for age 18 up to 100")
```



Compare this with a 50 year old

```
k <- 0:100
plot(k, lx[50 + k + 1] / lx[50 + 1],
    pch = 20,
    xlab = "k",
    ylab = expression(paste(""[k], "p"[50])),
    main = "Survival probabilities for age 50 up to 100")
```



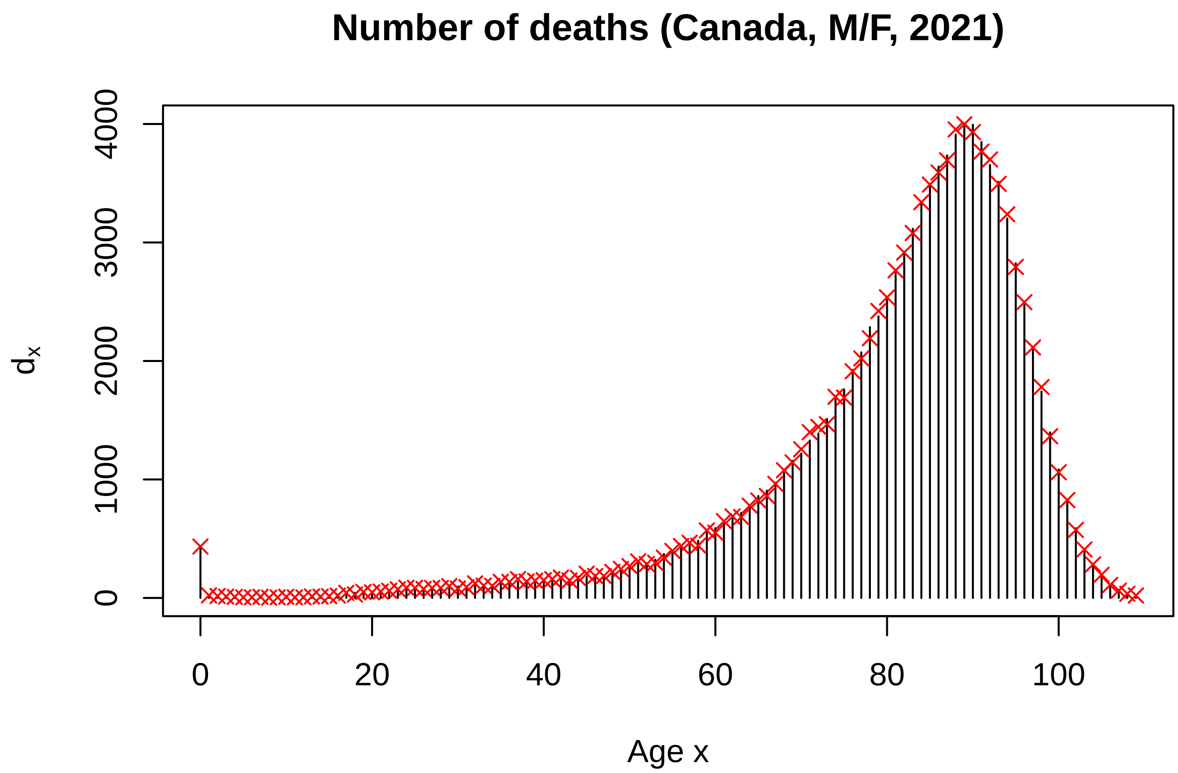
Plot the number of deaths

```r
# Plot the number of deaths dx by age
plot(age, dx,
    type = "h",
    pch = 20,
    xlab = "Age x",
    ylab = expression("d"[x]),
    main = "Number of deaths (Canada, M/F, 2021)")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```r
# Simulate the number of deaths using a binomial distribution
sims <- rbinom(n = length(lx), size = lx, prob = qx)

# Plot the simulated number of deaths on top of the previous graph
points(age, sims,
    pch = 4,
    col = "red")
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```



Calculate the probability that (25) survives 5 more years

```
prod(px[(25 + 1):(30 + 1)])
```

```
## [1] 0.9946519
```

Compute the survival probabilities of (25) until the age of 100

```
kpx <- cumprod(px[(25 + 1):(99 + 1)])
```

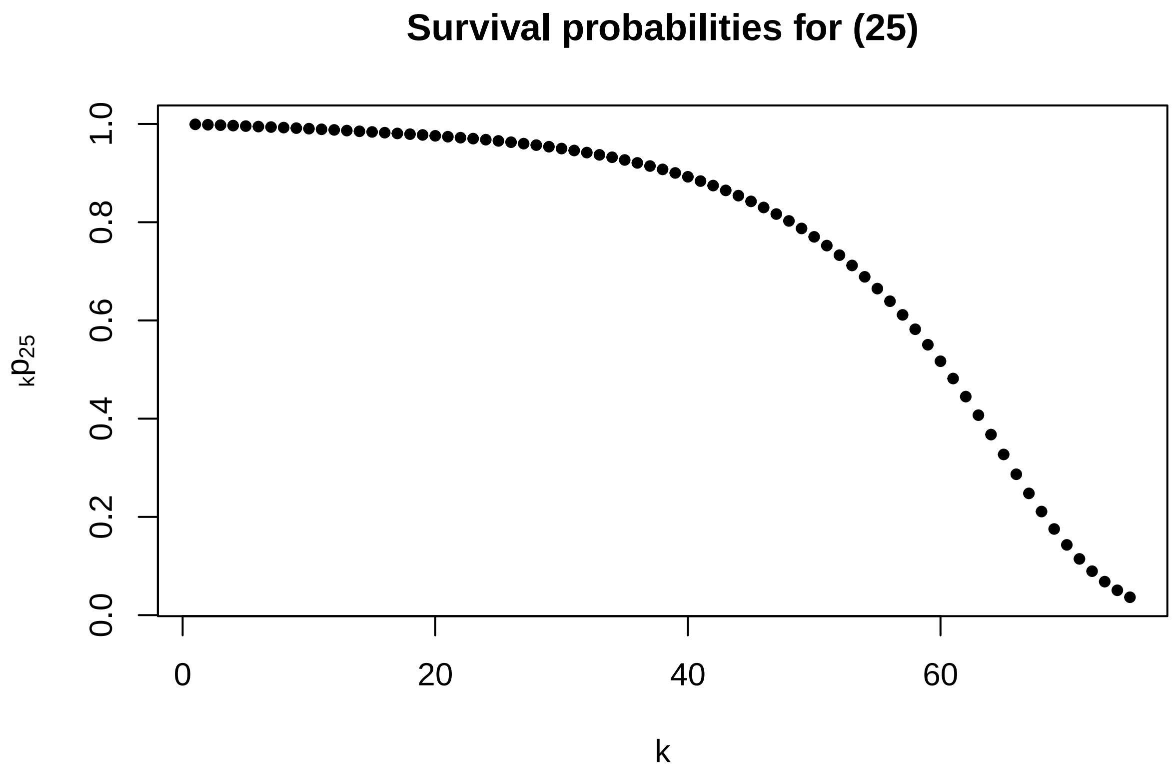
Extract the probability that (25) survives until the age of 100

```
kpx[length(kpx)]
```

```
## [1] 0.03645575
```

Plot the probabilties for (25) to reach the age of 26, 27, . . . , 100

```
plot(1:length(kpx), kpx,
    pch = 20,
    xlab = "k",
    ylab = expression(paste(""[k], "p"[25])),
    main = "Survival probabilities for (25)")
```



Compute the survival probabilities of (25)

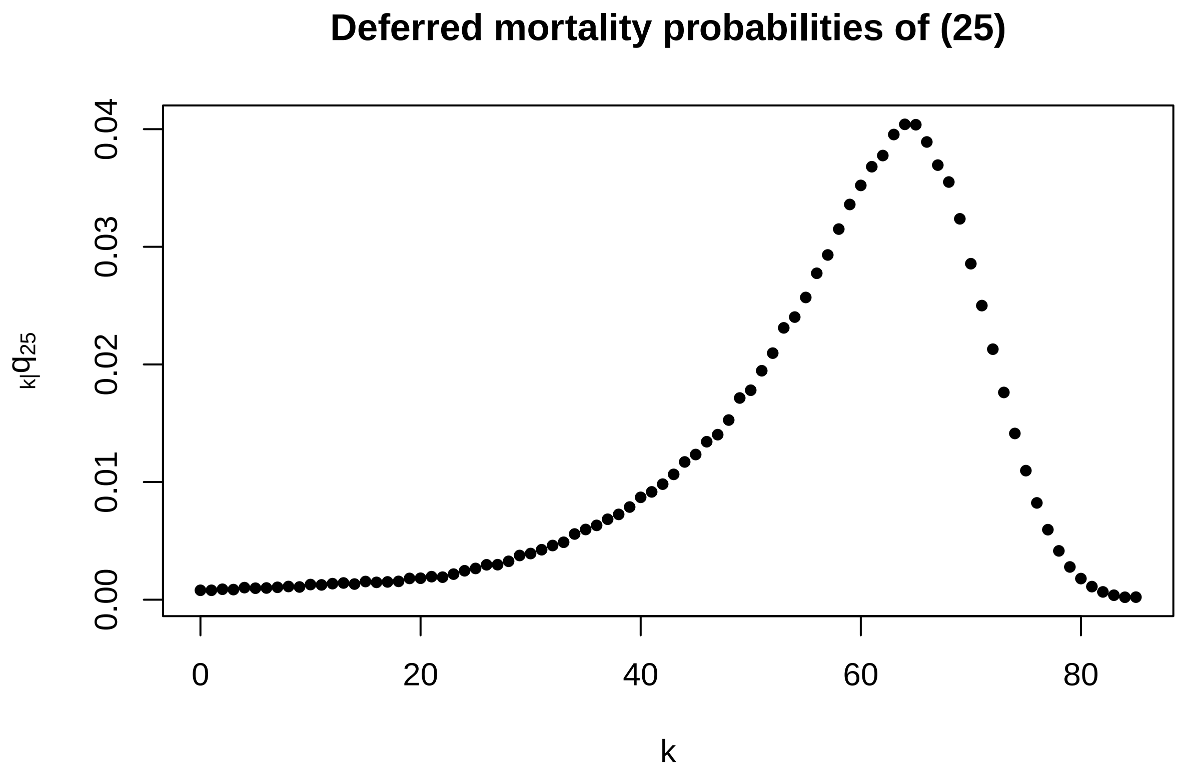
```
kpx <- c(1, cumprod(px[(25 + 1):(length(px) - 1)]))
```

Compute the deferred mortality probabilities of (25)

```
kqx <- kpx * qx[(25 + 1):length(qx)]
```

Plot the deferred mortality probabilities of (25)

```
plot(0:(length(kqx) - 1), kqx,
    pch = 20,
    xlab = "k",
    ylab = expression(paste(""['k|'], "q"[25])),
    main = "Deferred mortality probabilities of (25)")
```



Life Expectancy

Survival probabilities and curtate expected future lifetime of (0)

```
kp0 <- cumprod(px)
sum(kp0)
```

```
## [1] 81.15437
```

Survival probabilities and curtate expected future lifetime of a 25 year old

```
kp25 <- cumprod(px[(25 + 1):length(px)])
sum(kp25)
```

```
## [1] 56.92983
```

Complete expected future lifetime of (0) and (25)

```
ex[c(0 + 1, 25 + 1)]
```

```
## [1] 81.65 57.43
```

Function to compute the curtate expected future lifetime for a given age and life table

```
age2 <- as.numeric(age)
```

```
## Warning: NAs introduced by coercion
```

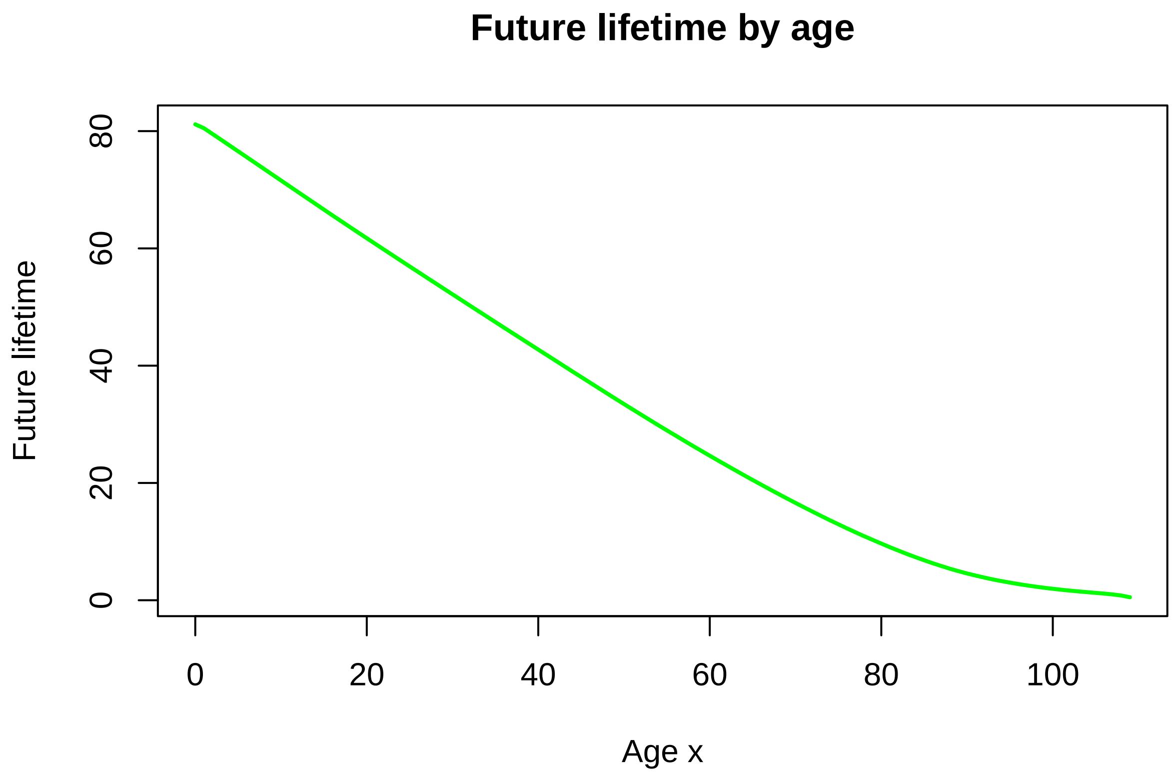
```
curtate_future_lifetime <- function(age2, life_table) {
  qx <- life_table$qx
  px <- 1 - qx
  kpx <- cumprod(px[(age2 + 1):length(px)])
  sum(kpx)
}
```

Curtate future lifetimes for all ages

```
valid_ages <- age2[!is.na(age2)]
future_lifetimes <- sapply(valid_ages, function(age) curtate_future_lifetime(age, life_table))
future_lifetimes
```

```
##   [1] 81.1543677 80.5105688 79.5298960 78.5410318 77.5512434 76.5582237
##   [7] 75.5651145 74.5696487 73.5741231 72.5800095 71.5850905 70.5908178
##  [13] 69.5964655 68.6041219 67.6123554 66.6211462 65.6364726 64.6555127
##  [19] 63.6800911 62.7081227 61.7394924 60.7728498 59.8069016 58.8434061
##  [25] 57.8858039 56.9298305 55.9754109 55.0202271 54.0692387 53.1157782
##  [31] 52.1705439 51.2217212 50.2729942 49.3263401 48.3821419 47.4349360
##  [37] 46.4966817 45.5558076 44.6187615 43.6831051 42.7421570 41.8093678
##  [43] 40.8721760 39.9352163 38.9988144 38.0710959 37.1424222 36.2172289
##  [49] 35.2887176 34.3679418 33.4554587 32.5477149 31.6482716 30.7466863
##  [55] 29.8518913 28.9699729 28.0901151 27.2168086 26.3505528 25.4885582
##  [61] 24.6421547 23.8018788 22.9662875 22.1393700 21.3176883 20.5058647
##  [67] 19.7077654 18.9142764 18.1290962 17.3552326 16.5964804 15.8432337
##  [73] 15.1037931 14.3678121 13.6465357 12.9503749 12.2568738 11.5823800
##  [79] 10.9233889 10.2898577  9.6617391  9.0501785  8.4609740  7.8870176
##  [85]  7.3383738  6.8153443  6.3136213  5.8359568  5.3771888  4.9556622
##  [91]  4.5677219  4.2107849  3.8717329  3.5497877  3.2682134  3.0078649
##  [97]  2.7581869  2.5286726  2.3181631  2.1255148  1.9494822  1.7886795
## [103]  1.6414429  1.5057902  1.3788531  1.2562353  1.1304400  0.9874820
## [109]  0.7999053  0.5110800
```

```
# Future lifetime by age
plot(age2[!is.na(age2)], future_lifetimes, type = 'l', lwd = 2, col = "green", xlab = "Age x", ylab = "F
```

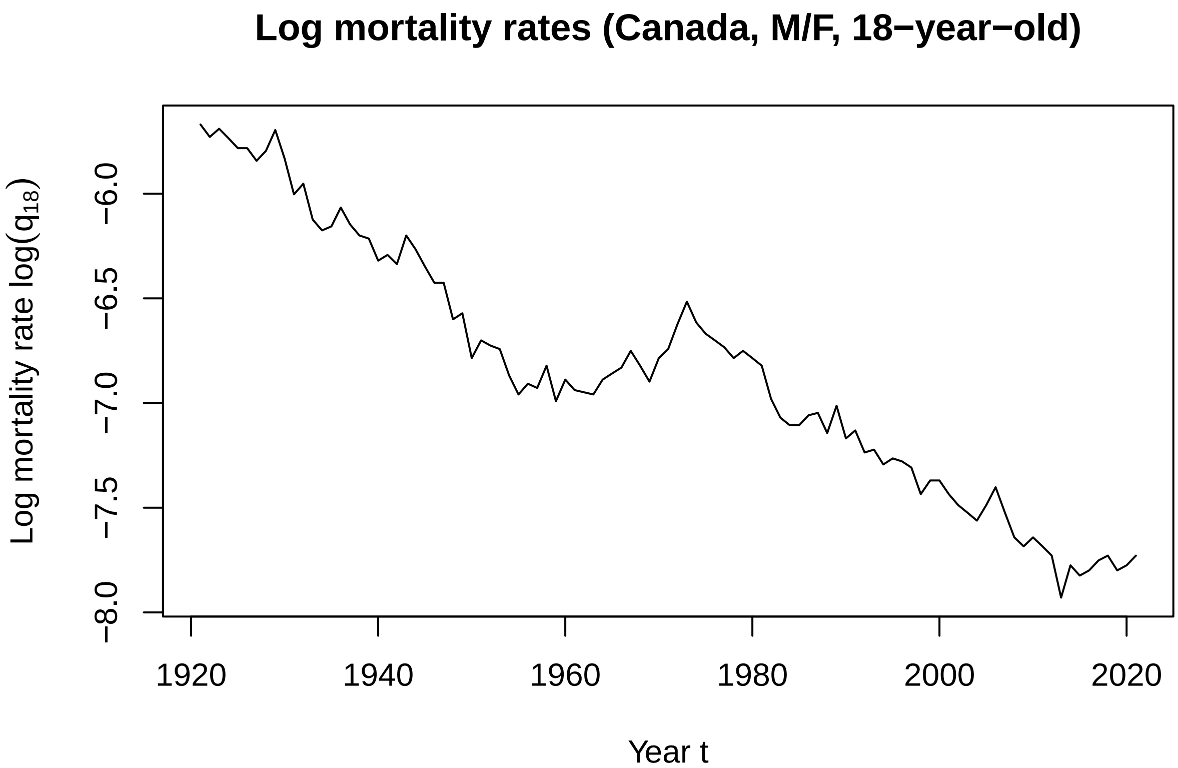


Plot the logarithm of the male/female mortality rates for (18) by year

```
# Load the life table
life_table <- read.table("C:\\Users\\megacrazyleo\\Desktop\\SQL\\R\\bltper_1x1.txt", header=FALSE, skip
colnames(life_table) <- c("Year", "Age", "mx", "qx", "ax", "lx", "dx", "Lx", "Tx", "ex")

# Ensure Age is numeric
life_table$Age <- as.numeric(life_table$Age)
```

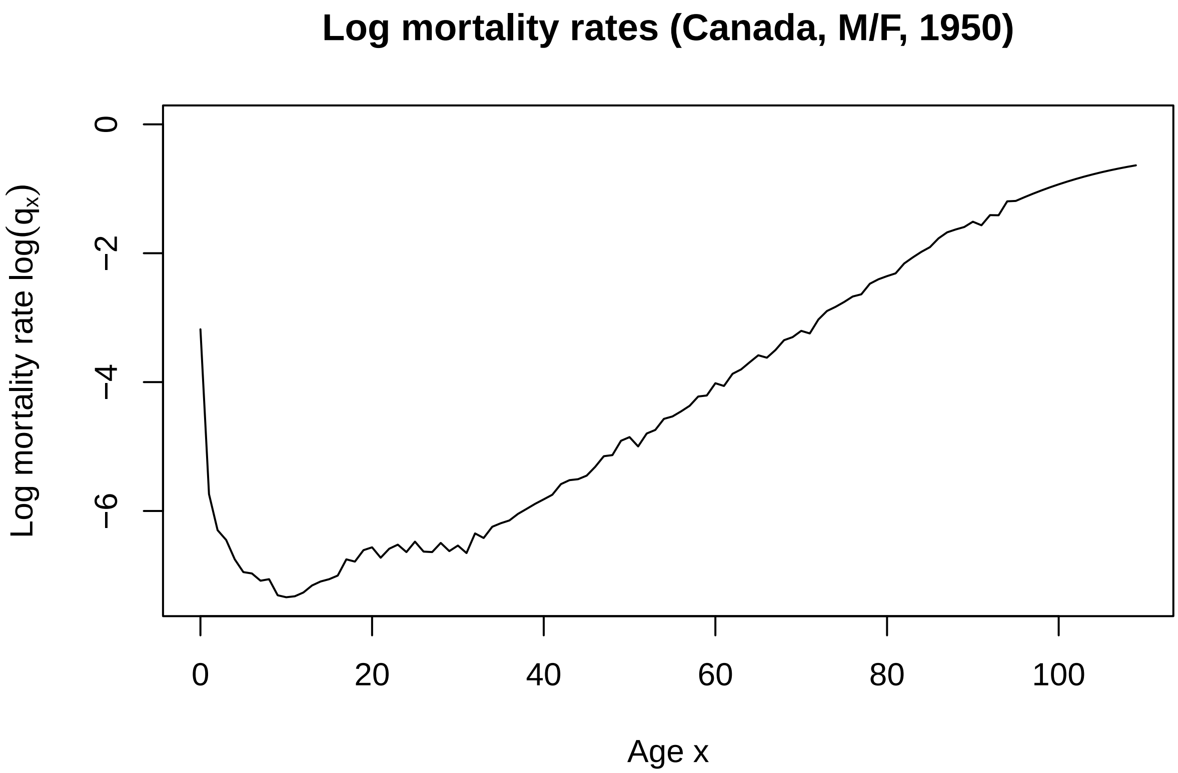
```
## Warning: NAs introduced by coercion
```

```
# Subset the life table for age 18
subset_18 <- subset(life_table, Age == 18)

# Check if the subset has the same length for 'Year' and 'qx'
if (nrow(subset_18) > 0) {
  # Plot the log mortality rates for age 18
  with(subset_18,
       plot(Year, log(qx),
            type = "l", main = "Log mortality rates (Canada, M/F, 18-year-old)",
            xlab = "Year t", ylab = expression(paste("Log mortality rate ", log(q[18])))))
} else {
  print("No data available for age 18")
}
```



Plot the logarithm of the female mortality rates in the year 1950 by age

```r
# Subset the life table for the year 1950
subset_1950 <- subset(life_table, Year == 1950)

# Check if the subset has the same length for 'Age' and 'qx'
if (nrow(subset_1950) > 0) {
  # Plot the log mortality rates for the year 1950
  with(subset_1950,
       plot(Age, log(qx),
            type = "l", main = "Log mortality rates (Canada, M/F, 1950)",
            xlab = "Age x", ylab = expression(paste("Log mortality rate ", log(q[x])))))
} else {
  print("No data available for the year 1950")
}
```



Construct and print the cohort life table for birth year 1981

```
life_table_1981 <- subset(life_table, Year - Age == 1981)
life_table_1981
```

```
##       Year Age      mx      qx   ax     lx  dx    Lx      Tx    ex
## 6661  1981   0 0.00972 0.00963 0.13 100000 963 99161 7548411 75.48
## 6773  1982   1 0.00063 0.00063 0.50  99076  63 99045 7470238 75.40
## 6885  1983   2 0.00044 0.00044 0.50  99077  43 99056 7408421 74.77
## 6997  1984   3 0.00039 0.00039 0.50  99073  39 99053 7338302 74.07
## 7109  1985   4 0.00028 0.00028 0.50  99059  28 99045 7238256 73.07
## 7221  1986   5 0.00029 0.00029 0.50  99037  29 99022 7153349 72.23
## 7333  1987   6 0.00020 0.00020 0.50  99077  20 99066 7084561 71.51
## 7445  1988   7 0.00023 0.00023 0.50  99072  23 99061 6991848 70.57
## 7557  1989   8 0.00017 0.00017 0.50  99056  17 99048 6919036 69.85
## 7669  1990   9 0.00020 0.00020 0.50  99091  20 99081 6850504 69.13
## 7781  1991  10 0.00015 0.00015 0.50  99150  15 99143 6766861 68.25
## 7893  1992  11 0.00015 0.00015 0.50  99164  15 99157 6692921 67.49
## 8005  1993  12 0.00021 0.00021 0.50  99150  21 99140 6581079 66.37
## 8117  1994  13 0.00026 0.00026 0.50  99122  26 99109 6499918 65.57
## 8229  1995  14 0.00027 0.00027 0.50  99134  26 99121 6412031 64.68
## 8341  1996  15 0.00031 0.00031 0.50  99180  31 99165 6337554 63.90
## 8453  1997  16 0.00053 0.00053 0.50  99141  52 99115 6257835 63.12
## 8565  1998  17 0.00042 0.00042 0.50  99150  41 99130 6178885 62.32
## 8677  1999  18 0.00063 0.00063 0.50  99099  62 99068 6100645 61.56
## 8789  2000  19 0.00064 0.00064 0.50  99080  63 99048 6035708 60.92
```

```
## 8901  2001  20 0.00059 0.00059 0.50  99008  58 98979 5958898 60.19
## 9013  2002  21 0.00058 0.00058 0.50  98946  57 98917 5871755 59.34
## 9125  2003  22 0.00055 0.00055 0.50  98905  54 98878 5789571 58.54
## 9237  2004  23 0.00055 0.00055 0.50  98888  55 98861 5719518 57.84
## 9349  2005  24 0.00056 0.00056 0.50  98782  56 98755 5631507 57.01
## 9461  2006  25 0.00056 0.00056 0.50  98779  55 98751 5572998 56.42
## 9573  2007  26 0.00057 0.00057 0.50  98728  56 98700 5474686 55.45
## 9685  2008  27 0.00050 0.00050 0.50  98717  49 98693 5394059 54.64
## 9797  2009  28 0.00050 0.00050 0.50  98709  49 98684 5327781 53.97
## 9909  2010  29 0.00056 0.00056 0.50  98664  56 98636 5256681 53.28
## 10021 2011  30 0.00056 0.00056 0.50  98624  55 98596 5174985 52.47
## 10133 2012  31 0.00058 0.00058 0.50  98579  57 98551 5094853 51.68
## 10245 2013  32 0.00064 0.00064 0.50  98532  63 98500 5005052 50.80
## 10357 2014  33 0.00077 0.00077 0.50  98474  76 98436 4910497 49.87
## 10469 2015  34 0.00073 0.00073 0.50  98374  72 98338 4819383 48.99
## 10581 2016  35 0.00083 0.00083 0.50  98252  81 98211 4733277 48.17
## 10693 2017  36 0.00094 0.00094 0.50  98049  92 98003 4624539 47.17
## 10805 2018  37 0.00098 0.00098 0.50  97930  96 97882 4528434 46.24
## 10917 2019  38 0.00097 0.00097 0.50  97963  95 97916 4464141 45.57
## 11029 2020  39 0.00123 0.00123 0.50  97569 120 97509 4305572 44.13
## 11141 2021  40 0.00157 0.00157 0.50  97306 153 97230 4207847 43.24
```

1981 cohort one-year survival probabilities

```
px <- 1 - life_table_1981$qx
```

1981 cohort survival probability that (18) survives 5 more years

```
prod(px[(18 + 1):(22 + 1)])
```

```
## [1] 0.9970136
```

1881 cohort survival probability that (18) survives 5 more years

```
with(subset(life_table, Year - Age == 1881), prod(1 - qx[(18 + 1):(22 + 1)]))
```

```
## [1] 0.9124045
```

PV of guaranteed payment of 10,000 in 5 years

```
PV <- 10000 * (1 + 0.02) ^ -5
PV
```

```
## [1] 9057.308
```

5 year survival probabilities of (20)

```
kpx <- prod(px[(20 + 1):(24 + 1)])
```

EPV of pure endowment of 10,000 in 5 years for (20)

```
PV * kpx
```

```
## [1] 9031.705
```

Life Annuities

PV of guaranteed payments of 10,000 in 5, 10 and 30 years

```
PV <- 10000 * (1 + 0.02) ^ - c(5, 10, 30)
PV
```

```
## [1] 9057.308 8203.483 5520.709
```

Survival probabilities of (20)

```
kpx <- cumprod(px[(20 + 1):length(px)])
```

EPV of pure endowments of 10,000 in 5, 10 and 30 years for (20)

```
PV * kpx[c(5, 10, 30)]
```

```
## [1] 9031.705 8158.312       NA
```

#whole life annuity due of 35

```
kpx <-
  c(1, cumprod(px[(35+1):length(px)]))
discount_facotrs <- (1+0.03)^-(0:(length(kpx)-1))
benefits <- rep(1,length(kpx))
sum(benefits*discount_factors*kpx)
```

```
## Warning in benefits * discount_factors: longer object length is not a multiple
## of shorter object length

## Warning in benefits * discount_factors * kpx: longer object length is not a
## multiple of shorter object length

## [1] 145.9806
```

#whole life immediate annuity of 35

```
kpx <-
  c(1, cumprod(px[(35+1):length(px)]))
discount_facotrs <- (1+0.03)^-(1:(length(kpx)))
benefits <- rep(1,length(kpx))
sum(benefits*discount_factors*kpx)
```

```
## Warning in benefits * discount_factors: longer object length is not a multiple
## of shorter object length

## Warning in benefits * discount_factors * kpx: longer object length is not a
## multiple of shorter object length

## [1] 145.9806
```

Function to compute the EPV of a whole life annuity due for a given age, interest rate i and life table

```
life_annuity_due <- function(age, i, life_table) {
  px <- 1 - life_table$qx
  kpx <- c(1, cumprod(px[(age + 1):length(px)]))
```

```r
  discount_factors <- (1 + i) ^ - (0:(length(kpx) - 1))
  sum(discount_factors * kpx)
}
```

EPV of a whole life annuity due for (20) at interest rate 2% using life_table

```r
life_annuity_due(20, 0.02, life_table)
```

```
## [1] 30.5222
```

EPV of a whole life annuity due for (20) at interest rate 5% and for (65) at interest rate 2% using life_table

```r
life_annuity_due(20, 0.05, life_table)
```

```
## [1] 18.09666
```

```r
life_annuity_due(65, 0.02, life_table)
```

```
## [1] 11.94589
```

EPV of a whole life annuity due for (20) at interest rate 2% using life_table

```r
life_annuity_due(20, 0.02, life_table)
```

```
## [1] 30.5222
```

Function to compute the EPV of a whole life immediate annuity for a given age, interest rate i and life table

```r
life_immediate_annuity <- function(age, i, life_table) {
  px <- 1 - life_table$qx
  kpx <- cumprod(px[(age + 1):length(px)])
  discount_factors <- (1 + i) ^ - (1:length(kpx))
  sum(discount_factors * kpx)
}
```

EPV of a whole life immediate annuity for (20) at interest rate 2% using life_table

```r
life_immediate_annuity(20, 0.02, life_table)
```

```
## [1] 29.5222
```

EPV of a whole life annuity due for (20) at interest rate 2% using life_table

```
life_annuity_due(20, 0.02, life_table)
```

```
## [1] 30.5222
```

Function to compute the EPV of a temporary life annuity due for a given age, period of n years, interest rate i and life table

```
temporary_life_annuity_due <- function(age, n, i, life_table) {
  px <- 1 - life_table$qx
  kpx <- c(1, cumprod(px[(age + 1):(age + n - 1)]))
  discount_factors <- (1 + i) ^ - (0:(n - 1))
  sum(discount_factors * kpx)
}
```

EPV of a temporary life annuity due for (20) over 10 years at interest rate 2% using life_table

```
temporary_life_annuity_due(20, 10, 0.02, life_table)
```

```
## [1] 9.002238
```

Calculating the PV of a pension at age 65

```
benefits <- 20000 * 1.02 ^ (0:35)


# Discount factors (to age 65)

discount_factors <- 1.04 ^ - (0:35)


# PV of pension at age 65

PV_65 <- sum(benefits * discount_factors)
PV_65
```

```
## [1] 523061
```

```
# PV of pension at age 20

PV_20 <- PV_65 * 1.03 ^ - 45
PV_20
```

```
## [1] 138317.5
```

EPV of pension at age 20

this might need some fixing

```
# Survival probabilities of (65) up to age 100
px <- 1 - life_table$qx
kpx <- c(1, cumprod(px[(65 + 1):(99 + 1)]))


# EPV of pension at age 65

EPV_65 <- sum(benefits * discount_factors * kpx)
cbind(PV_65, EPV_65)
```

```
##       PV_65   EPV_65
## [1,] 523061 239587.2
```

```
# EPV of pension at age 20

EPV_20 <- EPV_65 * (1.03 ^ - 45 * prod(px[(20 + 1):(64 + 1)]))
cbind(PV_20, EPV_20)
```

```
##         PV_20   EPV_20
## [1,] 138317.5 42596.32
```

Retirement Plan

Suppose # Survival probabilities of (40)

```
kpx <- c(1, cumprod(px[(40 + 1):length(px)]))
```

Discount factors (to age 40)

```
discount_factors <- (1 + 0.03) ^ -(0:(length(kpx) - 1))
```

Pension benefits

```
benefits <- c(rep(0, 25), rep(18000, length(kpx) - 25))
```

The single premium

```
single_premium <- sum(benefits * discount_factors * kpx)
single_premium
```

```
## [1] 70118.26
```

Premium pattern rho

```
rho <- c(rep(1, 15), rep(0.5, 10), rep(0, length(kpx) - 25))
```

The initial premium

```r
initial_premium <- single_premium / sum(rho * discount_factors * kpx)
initial_premium
```

```
## [1] 4973.94
```

The annual premiums

initial_premium * rho

Sum of the annual premiums (no actuarial discounting)

sum(initial_premium * rho)

Curtate life expectancy of (40)

sum(kpx[-1])

Present value of annuity benefits when (40) lives until age 75

subset1 <- 1:36 sum(benefits[subset1] * discount_factors[subset1])

Present value of annuity benefits when (40) lives until age 95

subset2 <- 1:56 sum(benefits[subset2] * discount_factors[subset2])

10-year survival probability of (20)

kpx <- prod(px[(20 + 1):(29 + 1)]) kpx

10-year deferred mortality probability of (20)

kqx <- kpx * qx[30 + 1] kqx

Discount factor

discount_factor <- (1 + 0.01) ˆ - 11 discount_factor

EPV of the simple life insurance

10000 * discount_factor * kqx

Function to compute the EPV of a whole life insurance

whole_life_insurance <- function(age, i, life_table) { qx <- life_table$qx px <- 1 - qx kpx <- c(1, cumprod(px[(age + 1):(length(px) - 1)])) kqx <- kpx * qx[(age + 1):length(qx)] discount_factors <- (1 + i) ˆ - (1:length(kqx)) sum(discount_factors * kqx) }

Plot the EPV of a whole life insurance for a range of ages at interest rate 3% using life_table

plot_by_age()

Plot the EPV of a whole life insurance for (20) for a range of interest rates using life_table

plot_by_interest_rate()

EPV of a whole life insurance for (20) at interest rate 2% using life_table

whole_life_insurance(20, 0.02, life_table)

Function to compute the EPV of a temporary life insurance

temporary_life_insurance <- function(age, n, i, life_table) { qx <- life_table$qx px <- 1 - qx kpx <- c(1, cumprod(px[(age + 1):(age + n - 1)])) kqx <- kpx * qx[(age + 1):(age + n)] discount_factors <- (1 + i) ˆ - (1:length(kqx)) sum(discount_factors * kqx) }

EPV of a temporary life insurance for (20) over a period of 45 years at interest rate 2% using life_table

temporary_life_insurance(20, 45, 0.02, life_table)

EPV of a whole life insurance for (20) at interest rate 2% using life_table

whole_life_insurance(20, 0.02, life_table)

Function to compute the EPV of a deferred whole life insurance

deferred_life_insurance <- function(age, u, i, life_table) { qx <- life_table$qx; px <- 1 - qx kpx <- c(1, cumprod(px[(age + 1):(length(px) - 1)])) kqx <- kpx * qx[(age + 1):length(qx)] discount_factors <- (1 + i) ˆ - (1:length(kqx)) benefits <- c(rep(0, u), rep(1, length(kpx) - u)) sum(benefits * discount_factors * kqx) }

EPV of a deferred life insurance for (20) deferred over 45 years at interest rate 2% using life_table

deferred_life_insurance(20, 45, 0.02, life_table)

Deferred mortality probabilites of (48)

kqx <- c(1, cumprod(px[(48 + 1):(73 + 1)])) * qx[(48 + 1):(74 + 1)]

Discount factors

discount_factors <- (1 + i) ˆ - (1:length(kqx))

Death benefits

benefits <- c(rep(0, 7), rep(40000, length(kqx) - 7))

EPV of the death benefits

EPV_death_benefits <- sum(benefits * discount_factors * kqx) EPV_death_benefits

Pure endowment

EPV_pure_endowment <- 80000 * (1 + i) ˆ - 27 * prod(px[(48 + 1):(74 + 1)]) EPV_pure_endowment

Premium pattern

kpx <- c(1, cumprod(px[(48 + 1):(73 + 1)])) discount_factors <- (1 + i) ˆ - (0:(length(kpx) - 1)) rho <- rep(1, length(kpx)) EPV_rho <- sum(rho * discount_factors * kpx) EPV_rho

Premium level

(EPV_death_benefits + EPV_pure_endowment) / EPV_rho