

K2013 implementation in R

Andreas Slättelid

K2013

We are given the following from Finanstilsynet:

```
#The weights given by finanstilsynet
w <- function(x, G){
  #male: G = "M"
  #female: G = "F"
  #x: age in calendar year t,
  if (G == "M"){
    return (min(2.671548-0.172480*x + 0.001485*x**2, 0))
  }
  else {
    return(min(1.287968-0.101090*x+ 0.000814*x**2,0))
  }
}

mu_kol_2013 <- function(x, G){
  #male
  if (G == "M"){
    return((0.241752+0.004536*10**(0.051*x))/1000)
  }
  #female
  else {
    return((0.085411+0.003114*10**(0.051*x))/1000)
  }
}

#0:alive, 1: dead
mu_K13 <- function(x, G, Y = 2022){
  return(mu_kol_2013(x, G)*(1 + w(x, G)/100)^(Y-2013))
}
```

The survival probability is given by:

$$p_{**}^x(t, s) = \exp \left(- \int_t^s \mu_{K13}(x + u, Y + u) du \right)$$

One way to solve this is to use the Trapezoidal rule:

$$\begin{aligned} \int_t^s f(u; x, Y) du &= \sum_{k=1}^N \frac{f(u_{k+1}; x, Y) - f(u_k; x, Y)}{2} \Delta u_k \\ &= \sum_{k=1}^N \frac{\mu_{K13}(x + u_{k-1}, Y + u_{k-1}) - \mu_{K13}(x + u_k, Y + u_k)}{2} \Delta u_k \end{aligned}$$

We will evaluate the integral yearly, giving us $\Delta u_k = 1 \implies N = (s - t)$:

#trapezoidal rule:

```
p_surv <- function(x, G, Y, t, s){

  if (t == s){
    return(1)
  }

  ages <- seq(x + t, x + s, by = 1) #x= 24 gives, 24,25,25, ...
  years <- seq(Y, Y + (s-t), by = 1) #Y= 2022 gives, 2022,2023, ...

  N <- (s-t)
  s1 <- 0
  for (k in 1:N){
    s1 <- s1 + 0.5*(mu_K13(x = ages[k] , G = G, Y = years[k]) +
                    mu_K13(x = ages[k+1], G = G, Y = years[k+1])
                    )
  }
  ans <- exp(-s1)
  return(ans)
}

#24 year old man in 2022, surviving the next 10 years
p_surv(24, "M", 2022, 0, 10)
```

```
## [1] 0.9967247
```

Using the built in integrate method:

```
#turning mu into a function of u, so that we can use integrate
mu <- function(u, x, G, Y = 2022){
  return(mu_kol_2013(x+u, G)*(1 + w(x+u, G)/100)~{(Y+u-2013)}))
}

p <- function(x, G, Y, t, s){

  if (t == s){
    return(1)
  }

  f <- Vectorize(mu)
  integral <- integrate(f, lower = t, upper = s, x=x, Y=Y, G=G)$value

  ans <- exp((-1)*integral)
  return(ans)
}

#24 year old man in 2022, surviving the next 10 years:
p(24, "M", 2022, 0, 10)

## [1] 0.9967253
```

Inspired by Rstudio's move to Posit, and wanted to throw in some Python. (Looking forward to the day i master pdf i Quarto :))

```
import numpy as np
import pandas as pd
from scipy.integrate import quad

def w(x, G):
    if G == "M":
        return min(2.671548-0.172480*x + 0.001485*x**2, 0)
    else:
        return min(1.287968-0.101090*x+ 0.000814*x**2,0)

def mu_kol_2013(x, G):
    if G == "M":
        return (0.241752+0.004536*10**(0.051*x))/1000
    else:
        return (0.085411+0.003114*10**(0.051*x))/1000

def mu(u, x, G, Y = 2022):
    return mu_kol_2013(x+u, G)*(1 + w(x+u, G)/100)**(Y+u-2013)

def p_surv(x, G, Y, t, s):

    integral = quad(mu, t, s, args = (x, G, Y))[0]

    return round(np.exp(-integral),6)

#24 year old man in 2022, surviving the next 10 years:
p_surv(24, "M", 2022, 0, 10)

## 0.996725
```