R code for paper "Modern tontine with bequest: Innovation in pooled annuity funds", by T. Bernhardt and C. Donnelly, *Insurance: Mathematics and Economics* (2019), 86, pp168-188

```
#---------------------------------------------
# Program by Thomas Bernhardt
# Heriot-Watt University, Edinburgh, UK
# 27 April 2019
#---------------------------------------------


#---------------------------------------------
#program FSM evaluates tPx of the Standard Survival Model
#program SSM evaluates the force of mortality mu(x+t) of the Standard
Survival Model
#details in Dickson et al.(2013) Actuarial Mathematics for Life
Contingent Risks
#"x" is a column vector applied to each column of "t"
#---------------------------------------------
FSM= function(
  x,#initial age
  t,#time span
  A=2.2e-4,#accident rate
  B=2.7e-6,#mortality scale
  C=1.124#mortality growth
){
  A+B*C^x*C^t
}
SSM= function(
  x,#initial age
  t,#time span
  A=2.2e-4,#accident rate
  B=2.7e-6,#mortality scale
  C=1.124#mortality growth
){
  exp(B*C^x*(1-C^t)/log(C)-A*t)
}

#---------------------------------------------
#program calculates the optimal parameters for the log utility case
#details in "Modern tontine with bequest: innovation in pooled annuity
products"
#output is a matrix
#1st row contains the optimal percentage in the tontine for different
bequest motives "B"
#1st dimension (columns) refers to time after and including the
initial age, by default in monthly steps (except 1st entry that is
reserved for the optimal percentage)
#2nd dimension (rows) refers to different bequest motives "B" (except
the 1st row that is reserved for the optimal percentage)
#---------------------------------------------
log.optimal = function(
  age=65,#initial age
  B=seq(0,8,0.2),#bequest motive
  maxT=35,#time span beyond initial age
  n=12L,#number of evaluations per year for consumption over time
```

```
  rho=0.05#time preference
){

  time=seq(0,maxT,length.out=n*maxT+1L)

  MA= integrate(function(t) {S=SSM(age,t); -exp(-
rho*t)*FSM(age,t)*S*log(S)}, 0, 50)$value
  MTt= sapply(time, function(t) integrate(function(y) exp(-
rho*y)*FSM(age+t,y)*SSM(age+t,y), 0, 50)$value)
  MT= MTt[1]

  A=(1-B*rho)/(1+B*rho*(MA/(MT-MA)))
  A[A<0]=0

  C=matrix(rep(B,rep.int(length(time),length(B))),ncol=length(B))
  C=rho/(1-(1-C*rho)*MTt)

  rbind(A,C)
}


#---------------------------------------------
#program calculates the optimal percentage in the tontine and optimal
constant consumption rate
#details in "Modern tontine with bequest: innovation in pooled annuity
products"
#output is an array of dimension 3
#1st dimension contains optimal parameters in order c(percentage,
consumption)
#2nd dimension ranges over given risk aversions "G" (be aware: 1-"G"
is the CRRA value)
#3rd dimension ranges over given bequest motives "B"
#the value 0 is prohibited as value for "G"
#the program takes 5min to run with default values on the tested
machine in 2018
#---------------------------------------------
optimal= function(
  age=65,#initial age
  A=seq(0,1,0.005),#percentage in tontine
  B=c(1,2,3,6,7),#bequest motive
  C=seq(0,0.15,0.0005),#consumption rate
  G=setdiff(seq(-3,1,0.05),0),#risk aversion
  maxT=78,#time span beyond initial age
  mu=0.085,#return on risky asset
  n=12L,#number of evaluations per year for Simpson's Rule
  r=0.05,#return on risk-free asset
  rho=0.05,#time preference
  sigma=0.2#volatility
){
  Grid= seq(0,maxT,length.out=n*maxT+1L)
  Force= FSM(age,Grid)
  S= SSM(age,Grid)
```

```
  Simpson= rep_len(c(2,4),n*maxT+1)
  Simpson[c(1,length(Simpson))]= 1

  Optimal= array(dim=c(2, length(G), length(B)))

  A.original=A
  C.original=C
  A.changed=setdiff(A,1)
  C.changed=setdiff(C,0)

  g.index=0L
  for(g in G){
    g.index=g.index+1L

    if(g>0){
      A=A.original
      C=C.original
    }else{
      A=A.changed
      C=C.changed
    }

    w= (mu-r)/(sigma^2*(1-g))
    if(w<1){
      mx= r+w*(mu-r)/2
    }else{
      mx= mu-(1-g)*sigma^2/2
    }

    CG=
matrix(rep(C^g,rep.int(length(Grid),length(C))),ncol=length(C))
    ExP= exp( matrix(rep((g*mx-rho)-
g*C,rep.int(length(Grid),length(C))),ncol=length(C)) *Grid)

    b.index=0L
    for(b in B){
      b.index=b.index+1L

      a.opt= 1
      c.opt= 0
      v.opt=-Inf

      for(a in A){
        Sa=S^(1-g*a)
        W=(Simpson*Sa)*(CG+b*(1-a)^g*Force)*ExP
        V=colSums(W)/(3*n*g)
        k=which.max(V)

        if(V[k]>v.opt){
          a.opt=a
          c.opt=C[k]
```

```
          v.opt=V[k]
        }
      }

      Optimal[,g.index,b.index]=c(a.opt,c.opt)
    }
  }

  return(Optimal)
}

#---------------------------------------------
#script to create main plots for the paper "Modern tontine with
bequest: innovation in pooled annuity products"
#uses package "scales"
#---------------------------------------------
#install.packages("scales")
library(scales)#for percent function
par(mfrow=c(2,2))#4 plots in one

G=setdiff(seq(-3,1,0.05),0)
Optimal=optimal(G=G)
CRRA = 1-rev(G)
Optimal=Optimal[,length(G):1,]

#---------------------------------------------
#percentage in the tontine as a function of constant relative risk
aversion
#---------------------------------------------
plot(CRRA, Optimal[1,,1],
     col="blue",#colour
     lwd=2,#thickness of drawing
     type ="l",#drawing, here line
     xlab=expression(paste("constant relative risk aversion ",1-
gamma)),#description of x-axis
     xlim=c(0,4),#range of x-axis
     yaxt="n",#surpress y-axis-ticks
     ylab="in the tontine",#description of y-axis
     ylim=c(0,1)#range of y-axis
)
yticks = seq(0,1,0.2)#choose numbers for y-axis-ticks
axis(at=yticks,#location of axis-ticks
     lab=percent(yticks),#symbols of axis-ticks
     side=2#axis is y-axis
)
points(CRRA, Optimal[1,,2], col="chartreuse4", lwd=2, type ="l")
points(CRRA, Optimal[1,,3], col="darkgoldenrod1", lwd=2, type ="l")
points(CRRA, Optimal[1,,4], col="chocolate", lwd=2, type ="l")
points(CRRA, Optimal[1,,5], col="red", lwd=2, type ="l")
legend(bg="white",#background of legend
```

```
col=c("blue","chartreuse4","darkgoldenrod1","chocolate","red"),#colour
of markers for entries
       legend=c("b=1","b=2","b=3","b=6","b=7"),#strings for entries
       lty=1,#drawing, here line
       lwd=2,#thickness of drawing
       x="bottomright"#where the legend appears
)

#---------------------------------------------
#consumption as a function of constant relative risk aversion
#---------------------------------------------
plot(CRRA, Optimal[2,,1],
     col="blue",#colour
     lwd=2,#thickness of drawing
     type ="l",#drawing, here line
     xlab=expression(paste("constant relative risk aversion ",1-
gamma)),#description of x-axis
     xlim=c(0,4),#range of x-axis
     yaxt="n",#surpress y-axis-ticks
     ylab="for consumption",#description of y-axis
     ylim=c(0,0.12)#range of y-axis
)
yticks = seq(0,0.12,0.02)#choose numbers for y-axis-ticks
axis(at=yticks,#location of axis-ticks
     lab=percent(yticks),#symbols of axis-ticks
     side=2#axis is y-axis
)
points(CRRA, Optimal[2,,2], col="chartreuse4", lwd=2, type ="l")
points(CRRA, Optimal[2,,3], col="darkgoldenrod1", lwd=2, type ="l")
points(CRRA, Optimal[2,,4], col="chocolate", lwd=2, type ="l")
points(CRRA, Optimal[2,,5], col="red", lwd=2, type ="l")

Zero=optimal(G=G,A=0)
Zero=Zero[,length(G):1,]

points(CRRA, Zero[2,,1],
       col="blue",#colour of drawing
       lty=2,#dotted drawing
       lwd=2,#thickness of drawing
       type="l"#line drawing
)
points(CRRA, Zero[2,,2],col="chartreuse4",lty=2,lwd=2,type="l")
points(CRRA, Zero[2,,3],col="darkgoldenrod1",lty=2,lwd=2,type="l")
points(CRRA, Zero[2,,4],col="chocolate",lty=2,lwd=2,type="l")
points(CRRA, Zero[2,,5],col="red",lty=2,lwd=2,type="l")
legend(bg="white",#background of legend

col=c("blue","chartreuse4","darkgoldenrod1","chocolate","red"),#colour
of markers for entries
       legend=c("b=1","b=2","b=3","b=6","b=7"),#strings for entries
       lty=1,#line drawing
```

```
       lwd=2,#thickness of drawing
       x="bottomright"#where the legend appears
)
legend(bg="white",
       col=c("gray","gray"),
       legend=c("with tontine","without tontine"),
       lty=c(1,2),
       lwd=2,
       x="topright"
)

#-------------------------------------------
#percentage in the tontine as a function of bequest in the log-utility
case
#-------------------------------------------
age=65
B=seq(0,8,0.2)
maxT=35
n=12
time=seq(0,maxT,length.out=n*maxT+1L)
log.Optimal= log.optimal(age=age,B=B)
plot(B, log.Optimal[1,],
     col="blue",#colour
     lwd=2,#thickness of drawing
     type ="l",#drawing, here line
     xlab="bequest level b",#description of x-axis
     xlim=c(0,8),#range of x-axis
     yaxt="n",#surpress y-axis-ticks
     ylab="in the tontine",#description of y-axis
     ylim=c(0,1)#range of y-axis
)
yticks = seq(0,1,0.2)#choose numbers for y-axis-ticks
axis(at=yticks,#location of axis-ticks
     lab=percent(yticks),#symbols of axis-ticks
     side=2#axis is y-axis
)

#-------------------------------------------
#consumption as a function of time
#-------------------------------------------
index= match(c(1,2,3,6,7),B)
log.Optimal= log.Optimal[,index]
log.Optimal= log.Optimal[-1,]
age=age+time
plot(age, log.Optimal[,1],
     col="blue",#colour
     lwd=2,#thickness of drawing
     type ="l",#drawing, here line
     xlab="age",#description of x-axis
     xlim=c(65,100),#range of x-axis
     yaxt="n",#surpress y-axis-ticks
     ylab="for consumption",#description of y-axis
```

```
     ylim=c(0,0.3)#range of y-axis
)
yticks= seq(0,0.30,0.05)#choose numbers for y-axis-ticks
axis(at=yticks,#location of axis-ticks
     lab=percent(yticks),#symbols of axis-ticks
     side=2#axis is y-axis
)
points(age, log.Optimal[,2],col="chartreuse4",lwd=2,type="l")
points(age, log.Optimal[,3],col="darkgoldenrod1",lwd=2,type="l")
points(age, log.Optimal[,4],col="chocolate",lwd=2,type="l")
points(age, log.Optimal[,5],col="red",lwd=2,type="l")

legend(bg="white",#background of legend

col=c("blue","chartreuse4","darkgoldenrod1","chocolate","red"),#colour
of markers for entries
       legend=c("b=1","b=2","b=3","b=6","b=7"),#strings for entries
       lty=1,#line drawing
       lwd=2,#thickness of drawing
       x="topleft"#where the legend appears
)
par(mfrow=c(1,1))
```