```
#Hunter Prey model McGowan CP, et al. 2023
#2-sex model with harvest as a subtraction from the population vector

reps = 1000 #number of simulation replicates
yrs = 10 #number of years in each simulation


RunHuntSim<-function(  #wrapping the simulation model in a function with management actions as the arguments
Season.length,  #number of days in the season
d.bag,          #per day
b.bag,          #total season limit
rut.match)   #rut.match modifies the harvest success rate. if rut.match = 1 success equals 100% if it equals 0, success = 0.8
{
#Use set.seed to make results reproducible and repeatable
set.seed(42)
#Demographic parameters:
S.af.m = 0.85 #mean adult female survival
Sd.af.m = 0.09 #SD mean adult survival
S.am.m = 0.80 #mean adult male survival
Sd.am.m = 0.09 #SD mean adult survival
S.jm.m = 0.75 #mean juvenile male survival
S.yf.m = 0.3 #mean survival of young females
Sd.yf.m = 0.1 #SD of mean survival of young females
S.ym.m = 0.3 #mean survival of young females
Sd.ym.m = 0.1 #SD of mean survival of young females
F.af.m = 1.2 #adult female fecundity
R.sex = 0.5 #male:female sex ratio at birth

#Hunter pop params
S.h.ij=matrix(0,reps,yrs) #object for hunter "survival" (retention) rates
R.h.ij=matrix(0,reps,yrs) #object for hunter "recruitment" rates
#values for logistic survival relationship
M.s=0.99 #maximum hunter survival, "S^Max" from the manuscript
a.h=0.87
b.h=0.025
#Hunting parameters
Perct.days=0.2
h.days.m = ((Season.length/7)*2)*Perct.days #Total season length/7 to equal the number of weeks, times two to represent the
number of weekend days times % of weekend days that the average hunter hunts
h.af.m=0.25/h.days.m #annual harvest of ad females per hunter
h.am.m=0.1/h.days.m  #annual harvest of ad males per hunter
h.jm.m=0.1/h.days.m  #annual harvest of juv males per hunter
h.yf.m=0.05/h.days.m #annual harvest of young females per hunter
h.ym.m=0.05/h.days.m #annual harvest of young males per hunter

h.af=matrix(0,reps,yrs)
h.am=matrix(0,reps,yrs)
h.jm=matrix(0,reps,yrs)
h.yf=matrix(0,reps,yrs)
h.ym=matrix(0,reps,yrs)
h.days=matrix(0,reps,yrs)
ha=matrix(0,reps,yrs)
hb=matrix(0,reps,yrs)

lic=matrix(0,reps,yrs)
price=10

#Set up and evaluate basic core matrix
library(popbio)
mat <- matrix(c(0,F.af.m*(1-R.sex),0,0,0,S.yf.m,S.af.m,0,0,0,0,F.af.m*R.sex,0,0,0,0,0,S.ym.m,0,0,0,0,0,S.jm.m,S.am.m), nrow =
5, byrow = T); mat
(gen.T = generation.time(mat))
(w = stable.stage(mat))
(lambda = lambda(mat))


#rep level demographic parameters
S.af.i=matrix(rbeta(reps,100*S.af.m,100*(1-S.af.m)),reps,1)  #Female survival
a.sf=matrix(0,reps,1)
b.sf=matrix(0,reps,1)
S.am.i=matrix(rbeta(reps,100*S.am.m,100*(1-S.am.m)),reps,1)  # Male Surv
a.sm=matrix(0,reps,1)
b.sm=matrix(0,reps,1)
S.jm.i=matrix(rbeta(reps,100*S.jm.m,100*(1-S.jm.m)),reps,1)  # Juv Male surv
a.sjm=matrix(0,reps,1)
b.sjm=matrix(0,reps,1)
S.yf.i=matrix(rbeta(reps,100*S.yf.m,100*(1-S.yf.m)),reps,1)  #female fawn surv
a.syf=matrix(0,reps,1)
b.syf=matrix(0,reps,1)
S.ym.i=matrix(rbeta(reps,100*S.ym.m,100*(1-S.ym.m)),reps,1)  #male fawn surv
a.sym=matrix(0,reps,1)
b.sym=matrix(0,reps,1)
#Objects to generate annual demographic rates
lam.h.ij = matrix(rnorm(reps*yrs,1.0,0.01),reps,yrs)
S.af.ij=matrix(0,reps,yrs)
S.am.ij=matrix(0,reps,yrs)
S.jm.ij=matrix(0,reps,yrs)
S.yf.ij=matrix(0,reps,yrs)
S.ym.ij=matrix(0,reps,yrs)
```

```
F.af.ij=matrix(rlnorm(reps*yrs,log(F.af.m),0.2),reps,yrs)  # Productivity



#Population size objects
N.hunt.i=matrix(rpois(reps,500)) #hunters initial
N.hunt.ij=matrix(0,reps,yrs) #annual hunter abundance
N.af.i = matrix(rpois(reps,9500))#initial female abun
N.af.ij=matrix(0,reps,yrs) #annual af abundance
N.am.i = matrix(rpois(reps,7900))#initial male abun
N.am.ij=matrix(0,reps,yrs) #annual adult male abundance
N.jm.i = matrix(rpois(reps,2000))#initial59 juv male abun
N.jm.ij=matrix(0,reps,yrs) #annual juv male abundance
N.yf.i = matrix(rpois(reps,2700))#initial young female abun
N.yf.ij=matrix(0,reps,yrs) #annual young female abundance
N.ym.i = matrix(rpois(reps,2700))#initial young male abun
N.ym.ij=matrix(0,reps,yrs) #annual young male abundance

#objects for harvest amounts of each age and sex class
h.af.ij=matrix(0,reps,yrs)
h.am.ij=matrix(0,reps,yrs)
h.jm.ij=matrix(0,reps,yrs)
h.yf.ij=matrix(0,reps,yrs)
h.ym.ij=matrix(0,reps,yrs)


for(i in 1:reps){
  #Parametric uncertainty for survival rates
  a.sf[i]=S.af.i[i]*100 #female deer
  b.sf[i]=(1-S.af.i[i])*100
  a.sm[i]=S.am.i[i]*100 #male deer
  b.sm[i]=(1-S.am.i[i])*100
  a.sjm[i]=S.jm.i[i]*100 #Juvenile male deer
  b.sjm[i]=(1-S.jm.i[i])*100
  a.syf[i]=S.yf.i[i]*100 #Young of the yaer females
  b.syf[i]=(1-S.yf.i[i])*100
  a.sym[i]=S.ym.i[i]*100 #younf of the year males
  b.sym[i]=(1-S.ym.i[i])*100
  N=matrix(,5,yrs)
  for(j in 1:yrs){
    #annual survival rates with temporal stochasticity
    S.af.ij[i,j]=rbeta(1,a.sf[i],b.sf[i])
    S.am.ij[i,j]=rbeta(1,a.sm[i],b.sm[i])
    S.jm.ij[i,j]=rbeta(1,a.sjm[i],b.sjm[i])
    S.yf.ij[i,j]=rbeta(1,a.syf[i],b.syf[i])
    S.ym.ij[i,j]=rbeta(1,a.sym[i],b.sym[i])
    #construct the annual projection matrix and population vector
    mat.ij <- matrix(c(0,F.af.ij[i,j]*(1-
R.sex),0,0,0,S.yf.ij[i,j],S.af.ij[i,j],0,0,0,0,F.af.ij[i,j]*R.sex,0,0,0,0,0,S.ym.ij[i,j],0,0,0,0,0,S.jm.ij[i,j],S.am.ij[i,j]),
nrow = 5, byrow = T)
    N[,1]=matrix(c(N.yf.i[i],N.af.i[i],N.ym.i[i],N.jm.i[i],N.am.i[i]),5,1)
    if (j>1) N[,j]=mat.ij%*%N[,j-1]            #pop.v =
matrix(c(N.yf.ij[i,j],N.af.ij[i,j],N.ym.ij[i,j],N.jm.ij[i,j],N.am.ij[i,j]),5,1)

    #hunter popualtion and take rates
    N.hunt.ij[i,1]=N.hunt.i[i] #initial hunter abundance

    if (j>1) N.hunt.ij[i,j]=(N.hunt.ij[i,j-1]*S.h.ij[i,j-1])+(N.hunt.ij[i,j-1]*R.h.ij[i,j-1])



     h.days[i,j]=sum(rpois(N.hunt.ij[i,j],h.days.m)) # total number of days Number of days hunted by all hunters

     if (rut.match==1) p.succ=1.0 else p.succ = 0.8 #rut match function

    h.af[i,j]=sum(rpois(h.days[i,j],h.af.m))*p.succ #amountfemale harvest
      if (h.af[i,j]>(N.hunt.ij[i,j]*d.bag*h.days[i,j])) h.af[i,j]=(N.hunt.ij[i,j]*d.bag*h.days[i,j]) #impose bag limit by bag
limit
    h.am[i,j]=sum(rpois(h.days[i,j],h.am.m))*p.succ #amount male harvest
      if (h.am[i,j]>N.hunt.ij[i,j]*b.bag) h.am[i,j]=N.hunt.ij[i,j]*b.bag #impose bag limit
    h.jm[i,j]=sum(rpois(h.days[i,j],h.jm.m))*p.succ
    h.yf[i,j]=sum(rpois(h.days[i,j],h.yf.m))*p.succ
    h.ym[i,j]=sum(rpois(h.days[i,j],h.ym.m))*p.succ

    #calculate current survival rates and recruitment rates of hunters, based on harvest success and regulations
    hb[i,j]=rbeta(1,1000*b.h,1000*(1-b.h))+(0.0003*b.bag)+(0.0001*d.bag)+(Season.length*0.00002)
    ha[i,j]=rbeta(1,1000*a.h,1000*(1-a.h))+(0.001*N.am.ij[i,j])+(0.002*h.af.ij[i,j])
    if(j>1)S.h.ij[i,j]=(M.s/(1+ha[i,j]*(exp(-hb[i,j]*h.am[i,j]))))else S.h.ij[i,j]=.9
    R.h.ij[i,j]=rbeta(1,17,83)+(0.0001*h.am[i,j])+(0.00002*N.am.ij[i,j])+(0.0001*(1/b.bag))

    #retain abundances for each deer age class and subtract harvest
    N.af.ij[i,j]=round(N[2,j],0)-h.af[i,j]
    N.am.ij[i,j]=round(N[5,j],0)-h.am[i,j]
    N.jm.ij[i,j]=round(N[4,j],0)-h.jm[i,j]
    N.yf.ij[i,j]=round(N[1,j],0)-h.yf[i,j]
    N.ym.ij[i,j]=round(N[3,j],0)-h.ym[i,j]

    lic[i,j]=N.hunt.ij[i,j]*price #calculates license revenue
```

```
  }}

#some model outputs
m.naf=apply(N.af.ij,2,median)
m.nam=apply(N.am.ij,2,median)
lb.naf=apply(N.af.ij,2,quantile, probs = c(0.025))
ub.naf=apply(N.af.ij,2,quantile, probs = c(0.975))
lb.nam=apply(N.am.ij,2,quantile, probs= c(0.025))
ub.nam=apply(N.am.ij,2,quantile, probs = c(0.975))
m.h=(apply(h.am,2,median))
lb.h=apply(h.am,2,quantile, probs= c(0.025))
ub.h=apply(h.am,2,quantile, probs = c(0.975))
m.ls=mean(apply(lic,1,sum))
m.hunt=apply(N.hunt.ij,2,median)
lb.hunt=apply(N.hunt.ij,2,quantile, probs= c(0.025))
ub.hunt=apply(N.hunt.ij,2,quantile, probs = c(0.975))
#retain summary stats as from the simulation
return(cbind(m.naf,lb.naf,ub.naf,m.nam,lb.nam,ub.nam,m.hunt,lb.hunt,ub.hunt,m.h,lb.h,ub.h,m.ls))
}
#Run the function with management actions input
Run1=RunHuntSim(Season.length=40,  #number of days in the season
           d.bag=2,          #per day
           b.bag=2,         #total season limit
           rut.match = 0)
Run1 #display function outputs

#some results figures
par(mfrow=c(2,2))
plot(Run1[,3],type="n", main="Adult Female abundance", ylab="Abundance", xlab="year",ylim=c(min(Run1[,2]),max(Run1[,3])))
lines(Run1[,1])
lines(Run1[,2],lty=3)
lines(Run1[,3],lty=3)
plot(Run1[,6],type="n", main="Adult Male abundance", ylab="Abundance", xlab="year", ylim=c(min(Run1[,5]),max(Run1[,6])))
lines(Run1[,4])
lines(Run1[,5],lty=3)
lines(Run1[,6],lty=3)
plot(Run1[,9],type="n",main="Hunter abundance",ylab="Hunter abundance",xlab="year",ylim=c(0,max(Run1[,9])))
lines(Run1[,7])
lines(Run1[,8],lty=3)
lines(Run1[,9],lty=3)
plot(Run1[,12],type="n",main="Male harvest",ylab="Number harvested",xlab="year",ylim=c(0,max(Run1[,12])))
lines(Run1[,10])
lines(Run1[,11],lty=3)
lines(Run1[,12],lty=3)

#out puts for MS table 1
Run1[10,1]
Run1[10,4]
Run1[10,7]
Run1[10,13]
```