Document 3- R codes for the asymptotic analyses of a stage-structured matrix model for spotted hyenas infected with CDV

S. Benhaiem, L. Marescot

2018-10-08 18:12:30

This code presents the stochastic analysis of the matrix model. To account for parameter uncertainty in the calculation and sensitivity analyses of the population's growth rate λ (Fig. 1, Fig. 2) and the basic reproduction number R0 (Fig. 3) and calculate confidence intervals around the mean abundance of female hyenas projected throughout the study period and predicted beyond (Fig. 5), we used Monte Carlo iterations. This code requires the packages 'popbio' 2.4.4 (Stubben and Milligan 2007) and 'popdemo' 1.3-0 (Stott et al. 2012).

This code is structured as follows:

1) Monte Carlo iterations

a) Calculating / Plotting the mean ± SD of λ (Fig 1) and R0
b) Plotting changes in population abundance - complete model and "no rank" model (Fig 5)

2) Sensitivity analysis of λ(Fig 2)

3) Sensitivity analysis of R0 (Fig 3)

Note that parameter and submatrix names may differ between main text and R codes.

---

1) Monte Carlo iterations to calculate the mean ± SD of λ and R0, plot Fig 1 and Fig 5

First, we create a function to compile the population indicators we are interested in: λ, R0, population size, sensitivity values of λ and R0.

```
start_time <- Sys.time()

library(popdemo)
```

```
## Warning: package 'popdemo' was built under R version 3.5.1
```

```
## Welcome to popdemo! This is version 1.3-0
## Use ?popdemo for an intro, or browseVignettes('popdemo') for vignettes
## Citation for popdemo is here: doi.org/10.1111/j.2041-210X.2012.00222.x
## Development and legacy versions are here: github.com/iainmstott/popdemo
```

```
library(popbio)

# First we create the function to store the population indicators we are interested in:

  results<-function(Mproj, NGMstoch, popvec, senslambda, theta){

  tabmoy<-matrix(0,2,2)
  rownames(tabmoy)<-c("lambda", "rnot")
  colnames(tabmoy)<-c("mean", "sd")

  # Lambda
```

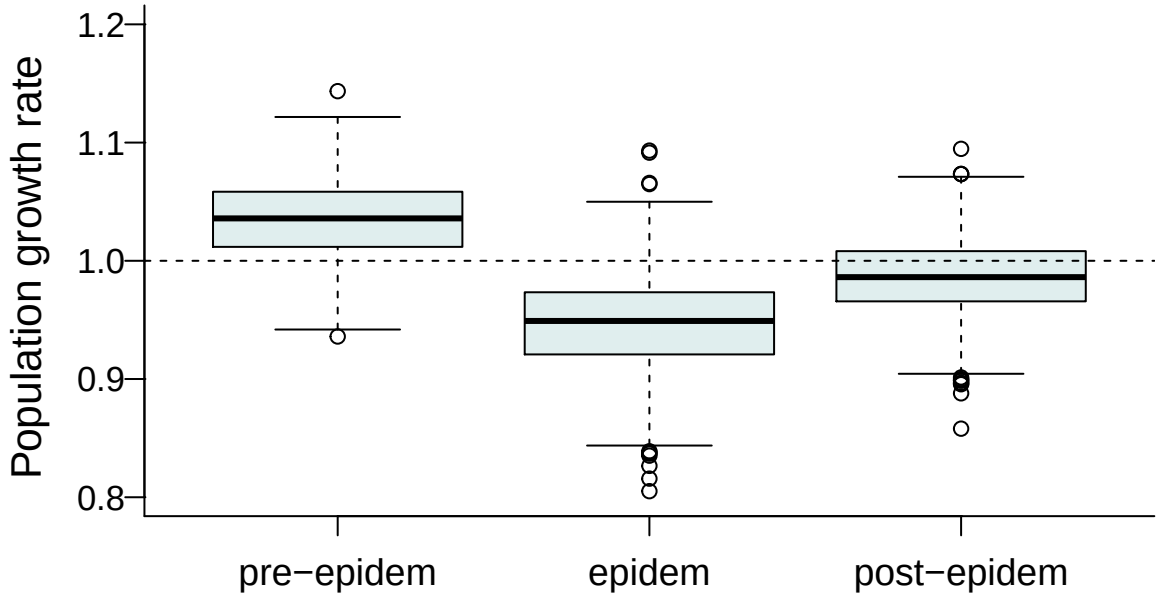
```
  alllambda<-unlist(lapply(Mproj, lambda))
  mean_lambda<-mean(alllambda)
  sd_lambda<-sd(alllambda)
  tabmoy[1,1]<-mean_lambda
  tabmoy[1,2]<-sd_lambda


  # Population size
  popsizeMean <- apply(popvec, 1, mean)
  popsizeSD <- apply(popvec, 1, sd)

  # Sensitivity of lambda
  meansens<-matrix(0, nrow=MCiter, ncol=42)

  for (i in 1:MCiter)
  meansens[i,]<-t(senslambda[[i]][2])

  sens_lambdaMean<-colMeans(meansens)
  sens_lambdaSD <- apply(meansens,2,sd)
  names(sens_lambdaMean) <- rownames((senslambda[[1]]))
  names(sens_lambdaSD) <- rownames((senslambda[[1]]))


  # Sensitivity of R0

  SensR0<- matrix(data = 0, nrow=ncol(theta), ncol=3)
  allrnot<-0
  sensi<-0
  senslist<-0


if(t == "epidem")
{

  allrnot<-unlist(lapply(NGMstoch, lambda))
  mean_rnot<-mean(allrnot)
  sd_rnot<-sd(allrnot)
  tabmoy[2,1]<-mean_rnot
  tabmoy[2,2]<-sd_rnot
  senslist<-list()
  for(r in 1:ncol(theta))
  {
  senslist[[r]]<- sens_elas_num(r, theta, delta=1e-4)
  sensi<- sens_elas_num(r, theta, delta=1e-4)
  SensR0[r,2]<-as.numeric(mean(sensi[[2]]))
  SensR0[r,3]<-as.numeric(sd(sensi[[2]]))
  SensR0[r,1]<-as.character(sensi[[1]])
  }

}

  return(list(tabmoy, popsizeMean, popsizeSD, sens_lambdaMean, sens_lambdaSD, alllambda, allrnot, means
}
```

Second, we load the text file containing the regression coefficients and the R file MC_simulations. In this file, we first draw 1000 values from normal distributions with means equal to the regression coefficients of the MECMR model and with standard deviations equal to the standard errors associated with these regression coefficients. To obtain the MECMR parameter estimates and insure that they corresponded to probabilities bounded between 0 and 1, we back-transformed those simulated regression coefficients using the logit-function after accounting for the structural interactions and the temporal additive effects detected on those parameters.

```
data<-read.table("./RegressionCoefficient.txt", header = TRUE)

MCiter<-1000 # Number of Monte Carlo iterations (put 1000)

checkNodisease <- FALSE # this is to use the probabilities of infection estimated for each epidemic per

# Pre-epidemic period (1990-1992)
t<-"pre-epidem"
period <- t
popsize0<-100
Tmax<-3
source('MC_simulations.R')

tabprepidem<-results(Mproj, NGMstoch, popvec,senslambda, theta)

# Epidemic period (1993-1994)
t<-"epidem"
period <- t
Tmax<-3
popsize0<-tabprepidem[[2]][3]
source('MC_simulations.R')
tabepidem<-results(Mproj, NGMstoch, popvec,senslambda, theta)

# Post-epidemic period (1995-1999)
t<-"post-epidem1"
period <- t
Tmax<-6
popsize0<-tabepidem[[2]][3]
source('MC_simulations.R')
tabpost1<-results(Mproj, NGMstoch, popvec,senslambda, theta)

# Period of virus absence in the ecosystem (2000-2010)
t<-"post-epidem2"
period <- t
Tmax<-12
checkNodisease <- TRUE # this sets the probability of infection to 0
popsize0<-tabpost1[[2]][6]

source('MC_simulations.R')
tabpost2<-results(Mproj, NGMstoch, popvec,senslambda, theta)

# Projections in the future (2010 to 2020)
Tmax<-13
checkNodisease <- TRUE
popsize0<-tabpost2[[2]][12]
source('MC_simulations.R')
tabprojection<-results(Mproj, NGMstoch, popvec,senslambda, theta)
```

a) Calculating / Plotting the mean ± SD of λ (Fig 1) and R0

```
data2 <- matrix(0,MCiter,3)

colnames(data2) <- c("pre-epidem", "epidem","post-epidem1")

data2[,1]<-tabprepidem[[6]]
data2[,2]<-tabepidem[[6]]
data2[,3]<-tabpost1[[6]]

# pdf("Figure1_Lambda.pdf",width=7,height=5) # uncomment this line to get the figure as a pdf

boxplot(data2, las=1, ylab = " ", xlab ="",  axes =F, ylim = c(0.80,1.20), col="azure2")

x1 <- c(-0.1,0,1,2,3,4)
y1 <- c(0.70,0.80,0.90,1.00,1.10,1.20,1.30)
axis(1,at=x1,las=1,cex.axis=.6,tck=0, labels=NA) # this is  to extend the axis (otherwise axes do not c
axis(2,at=y1,las=1,cex.axis=.6,tck=0, labels =NA)# idem for y

x <- c(1,2,3)
y <- c(0.80,0.90,1.00,1.10,1.20)

axis(1,at=x, las=1,cex.axis=1.2,labels =c("pre-epidem","epidem","post-epidem"))
axis(2,at=y,las=1, cex.axis=1.1,mgp=c(3, 0.5, 0))
mtext(side = 2, text ="Population growth rate ", line = 2.5, cex=1.3)

abline (h=1,lty=2)
```




```
#dev.off()# uncomment this line to get the figure as a pdf
```

In the following table, we display the basic reproduction number during the epidemic:

Basic reproduction number	Mean estimate	SD
R0	5.6836077	0.6563285

b) Plotting changes in population abundance - complete model and "no rank" model (Fig 5)

We can then project past (1990-2010) and predicted (2011-2020) temporal changes in λ (Figure 5) given parameter uncertainty during the 20 years of survey (1990-2010).

- To describe the past temporal changes during the study period (1990-2010) we calculated 1000 values of λ for each year of the study period accounting for temporal variations in the mean estimates between the pre-epidemic, epidemic and post-epidemic periods We described with a Markov chain the changes in population size by mutliplying the population vector of a given year by λ and reinitialized the population vector to the resulting vector of next year abundance. The initial population vector was defined as the product of 100 individuals and the stable stage distribution.
- To predict abundance of spotted hyenas for 10 years after the end of the study period (i.e. 2011-2020) we considered the 1000 block-matrices M implemented with the MECMR parameter estimates associated with the second post-epidemic period (2000-2010), and we determined the population vector of the number of individuals in the 22 demographic, social and infection states during the last year of the survey (2010). This vector was defined as the product of the mean abundance estimated in 2010 and the stable stage distribution. We then multiplied the matrices with this population vector to obtain 1000 population vectors and calculate the confidence intervals of the abundance the following year. These population vectors were then multiplied again by the simulated matrices to calculate the

mean abundance and its associated confidence interval in the following year. In such case the range of abundance increases with time as we use all the population vectors from a previous year (and not only the mean one) to calculate the range of population sizes the next year, accounting for the uncertainty around the parameter estimates. This Markov chain in which the population vectors of the next year only depend on the population vectors of the current year and of the simulated projection matrices was then reiterated for 10 years.

```
lambdaseq <- c(rep(tabprepidem[[1]][1,1], 3), rep(tabepidem[[1]][1,1], 2) , rep(tabpost1[[1]][1,1], 5),

lambdase <- c(rep(tabprepidem[[1]][1,2], 3), rep(tabepidem[[1]][1,2], 2) , rep(tabpost1[[1]][1,2], 5),re

popproj<-rep(0,31)
popproj[1]<-100
varmoins<-rep(0,31)
varplus<-rep(0,31)
varplus[1]<-100
varmoins[1]<-100

# loop on the retrospective (1990 to 2010)

for(t in 2:21)
{
  popproj[t]<-popproj[t-1] * lambdaseq[t]
  varplus[t]<-popproj[t-1]  * (lambdaseq[t] + lambdase[t])
  varmoins[t]<-popproj[t-1]  * (lambdaseq[t] - lambdase[t])
}

# projections 2010-2020

p<-1
for(t in 22:33)
{
  p<-p+1
  popproj[t]<- tabprojection[[2]][p]
  varplus[t]<- tabprojection[[2]][p] +  tabprojection[[3]][p]
  varmoins[t]<- tabprojection[[2]][p] - tabprojection[[3]][p]
}

# To get a smoother curve, we delete here the meaningless abundances obtained at the 2 first years of t
popproj<-popproj[-c(22,23)]
varplus<-varplus [-c(22,23)]
varmoins<-varmoins[-c(22,23)]

mat2 <- cbind(popproj,varmoins,varplus)

mat2<-as.data.frame(mat2)
mat2$Model<-rep("normal", 31)
mat2$Time<-1:31
mat2$Time1 <-c(1990:2020)
mat2
```

```
##      popproj  varmoins   varplus  Model Time Time1
## 1  100.00000 100.00000 100.00000 normal    1  1990
## 2  103.59755 100.29873 106.89636 normal    2  1991
## 3  107.32451 103.90702 110.74200 normal    3  1992
```

```
## 4  101.73940  97.40575 106.07305 normal    4  1993
## 5   96.44493  92.33680 100.55306 normal    5  1994
## 6   95.15359  92.04800  98.25919 normal    6  1995
## 7   93.87954  90.81553  96.94355 normal    7  1996
## 8   92.62254  89.59956  95.64553 normal    8  1997
## 9   91.38238  88.39987  94.36489 normal    9  1998
## 10  90.15882  87.21625  93.10140 normal   10  1999
## 11  91.92250  89.36115  94.48384 normal   11  2000
## 12  93.72067  91.10922  96.33212 normal   12  2001
## 13  95.55402  92.89149  98.21656 normal   13  2002
## 14  97.42324  94.70862 100.13786 normal   14  2003
## 15  99.32902  96.56129 102.09674 normal   15  2004
## 16 101.27208  98.45021 104.09394 normal   16  2005
## 17 103.25315 100.37608 106.13021 normal   17  2006
## 18 105.27297 102.33962 108.20632 normal   18  2007
## 19 107.33230 104.34158 110.32303 normal   19  2008
## 20 109.43192 106.38269 112.48116 normal   20  2009
## 21 111.57261 108.46373 114.68150 normal   21  2010
## 22 114.70994 106.24725 123.17263 normal   22  2011
## 23 117.34259 105.98808 128.69710 normal   23  2012
## 24 120.12617 105.58485 134.66748 normal   24  2013
## 25 123.08377 105.09226 141.07528 normal   25  2014
## 26 126.21910 104.52257 147.91564 normal   26  2015
## 27 129.53854 103.87668 155.20040 normal   27  2016
## 28 133.05033 103.14841 162.95224 normal   28  2017
## 29 136.76349 102.32762 171.19936 normal   29  2018
## 30 140.68797 101.40149 179.97445 normal   30  2019
## 31 144.83465 100.35490 189.31440 normal   31  2020
```

In the next step we repeat this procedure for the model "no rank".

```
data<-read.table("./RegressionCoefficientNORANK.txt", header = TRUE)

checkNodisease <- FALSE
t<-"pre-epidem"
period<-t
popsize0<-100
Tmax<-3
source('MC_simulations.R')
tabprepidemNR<-results(Mproj, NGMstoch, popvec, senslambda, theta)

t<-"epidem"
period<-t
Tmax<-3
popsize0<-tabprepidemNR[[2]][3]
source('MC_simulations.R')
tabepidemNR<-results(Mproj, NGMstoch, popvec, senslambda, theta)

t<-"post-epidem1"
period<-t
Tmax<-6
popsize0<-tabepidemNR[[2]][3]
source('MC_simulations.R')
tabpost1NR<-results(Mproj, NGMstoch, popvec, senslambda, theta)
```

```
t<-"post-epidem2"
Tmax<-12
period<-t
checkNodisease <- TRUE
popsize0<-tabpost1NR[[2]][6]
source('MC_simulations.R')

tabpost2NR<-results(Mproj, NGMstoch, popvec, senslambda, theta)

Tmax<-13
period<-t
checkNodisease <- TRUE
popsize0<-tabpost2NR[[2]][12]
source('MC_simulations.R')
tabprojectionNR<-results(Mproj, NGMstoch, popvec, senslambda, theta)
```

We store in a data frame the population vectors describing variations in expected abundance over 30 years, in both the models with and without rank effect.

```
lambdaseq <- c(rep(tabprepidemNR[[1]][1,1], 3), rep(tabepidemNR[[1]][1,1], 2) , rep(tabpost1NR[[1]][1,1

lambdase <- c(rep(tabprepidemNR[[1]][1,2], 3), rep(tabepidemNR[[1]][1,2], 2) , rep(tabpost1NR[[1]][1,2]


popproj<-rep(0,31)
popproj[1]<-100
varmoins<-rep(0,31)
varplus<-rep(0,31)
varplus[1]<-100
varmoins[1]<-100


# loop on the retrospective (1990 to 2010)

for(t in 2:21) # starts at 2 because t-1 if t=1 is 0
{
  popproj[t]<-popproj[t-1] * lambdaseq[t]
  varplus[t]<-popproj[t-1]  * (lambdaseq[t] + lambdase[t])
  varmoins[t]<-popproj[t-1]  * (lambdaseq[t] - lambdase[t])
}

# Projections 2010-2020

p<-1
for(t in 22:33)
{
  p<-p+1 # to move to the next value
  popproj[t]<- tabprojectionNR[[2]][p]
  varplus[t]<- tabprojectionNR[[2]][p] + tabprojectionNR[[3]][p]
  varmoins[t]<- tabprojectionNR[[2]][p] - tabprojectionNR[[3]][p]
}


popproj<-popproj[-c(22,23)]
```

```r
varplus<-varplus [-c(22,23)]
varmoins<-varmoins[-c(22,23)]



matNR <- cbind(popproj,varmoins,varplus)
matNR<-as.data.frame(matNR)
matNR$Model<-rep("NR", 31)

matNR$Time<-1:31
matNR$Time1 <-c(1990:2020)


matNEW <-rbind(mat2,matNR)
matNEW
```

```
##      popproj  varmoins   varplus  Model Time Time1
## 1  100.00000 100.00000 100.00000 normal    1  1990
## 2  103.59755 100.29873 106.89636 normal    2  1991
## 3  107.32451 103.90702 110.74200 normal    3  1992
## 4  101.73940  97.40575 106.07305 normal    4  1993
## 5   96.44493  92.33680 100.55306 normal    5  1994
## 6   95.15359  92.04800  98.25919 normal    6  1995
## 7   93.87954  90.81553  96.94355 normal    7  1996
## 8   92.62254  89.59956  95.64553 normal    8  1997
## 9   91.38238  88.39987  94.36489 normal    9  1998
## 10  90.15882  87.21625  93.10140 normal   10  1999
## 11  91.92250  89.36115  94.48384 normal   11  2000
## 12  93.72067  91.10922  96.33212 normal   12  2001
## 13  95.55402  92.89149  98.21656 normal   13  2002
## 14  97.42324  94.70862 100.13786 normal   14  2003
## 15  99.32902  96.56129 102.09674 normal   15  2004
## 16 101.27208  98.45021 104.09394 normal   16  2005
## 17 103.25315 100.37608 106.13021 normal   17  2006
## 18 105.27297 102.33962 108.20632 normal   18  2007
## 19 107.33230 104.34158 110.32303 normal   19  2008
## 20 109.43192 106.38269 112.48116 normal   20  2009
## 21 111.57261 108.46373 114.68150 normal   21  2010
## 22 114.70994 106.24725 123.17263 normal   22  2011
## 23 117.34259 105.98808 128.69710 normal   23  2012
## 24 120.12617 105.58485 134.66748 normal   24  2013
## 25 123.08377 105.09226 141.07528 normal   25  2014
## 26 126.21910 104.52257 147.91564 normal   26  2015
## 27 129.53854 103.87668 155.20040 normal   27  2016
## 28 133.05033 103.14841 162.95224 normal   28  2017
## 29 136.76349 102.32762 171.19936 normal   29  2018
## 30 140.68797 101.40149 179.97445 normal   30  2019
## 31 144.83465 100.35490 189.31440 normal   31  2020
## 32 100.00000 100.00000 100.00000     NR    1  1990
## 33 102.23532  99.57277 104.89786     NR    2  1991
## 34 104.52060 101.79853 107.24266     NR    3  1992
## 35  97.85228  94.16247 101.54209     NR    4  1993
## 36  91.60939  88.15499  95.06380     NR    5  1994
## 37  89.29581  86.88226  91.70936     NR    6  1995
```

```
## 38  87.04066  84.68806  89.39326     NR    7  1996
## 39  84.84246  82.54928  87.13564     NR    8  1997
## 40  82.69978  80.46451  84.93505     NR    9  1998
## 41  80.61121  78.43239  82.79002     NR   10  1999
## 42  81.47686  79.90866  83.04505     NR   11  2000
## 43  82.35180  80.76676  83.93683     NR   12  2001
## 44  83.23614  81.63408  84.83819     NR   13  2002
## 45  84.12998  82.51072  85.74923     NR   14  2003
## 46  85.03341  83.39676  86.67006     NR   15  2004
## 47  85.94655  84.29232  87.60077     NR   16  2005
## 48  86.86949  85.19750  88.54148     NR   17  2006
## 49  87.80234  86.11240  89.49228     NR   18  2007
## 50  88.74521  87.03712  90.45330     NR   19  2008
## 51  89.69821  87.97178  91.42464     NR   20  2009
## 52  90.66144  88.91647  92.40641     NR   21  2010
## 53  91.17895  86.54329  95.81461     NR   22  2011
## 54  92.06503  85.73942  98.39064     NR   23  2012
## 55  93.00552  84.88132 101.12971     NR   24  2013
## 56  94.00604  84.00318 104.00890     NR   25  2014
## 57  95.06625  83.11270 107.01980     NR   26  2015
## 58  96.18626  82.21230 110.16022     NR   27  2016
## 59  97.36664  81.30116 113.43212     NR   28  2017
## 60  98.60816  80.37686 116.83945     NR   29  2018
## 61  99.91175  79.43615 120.38735     NR   30  2019
## 62 101.27855  78.47526 124.08184     NR   31  2020
```

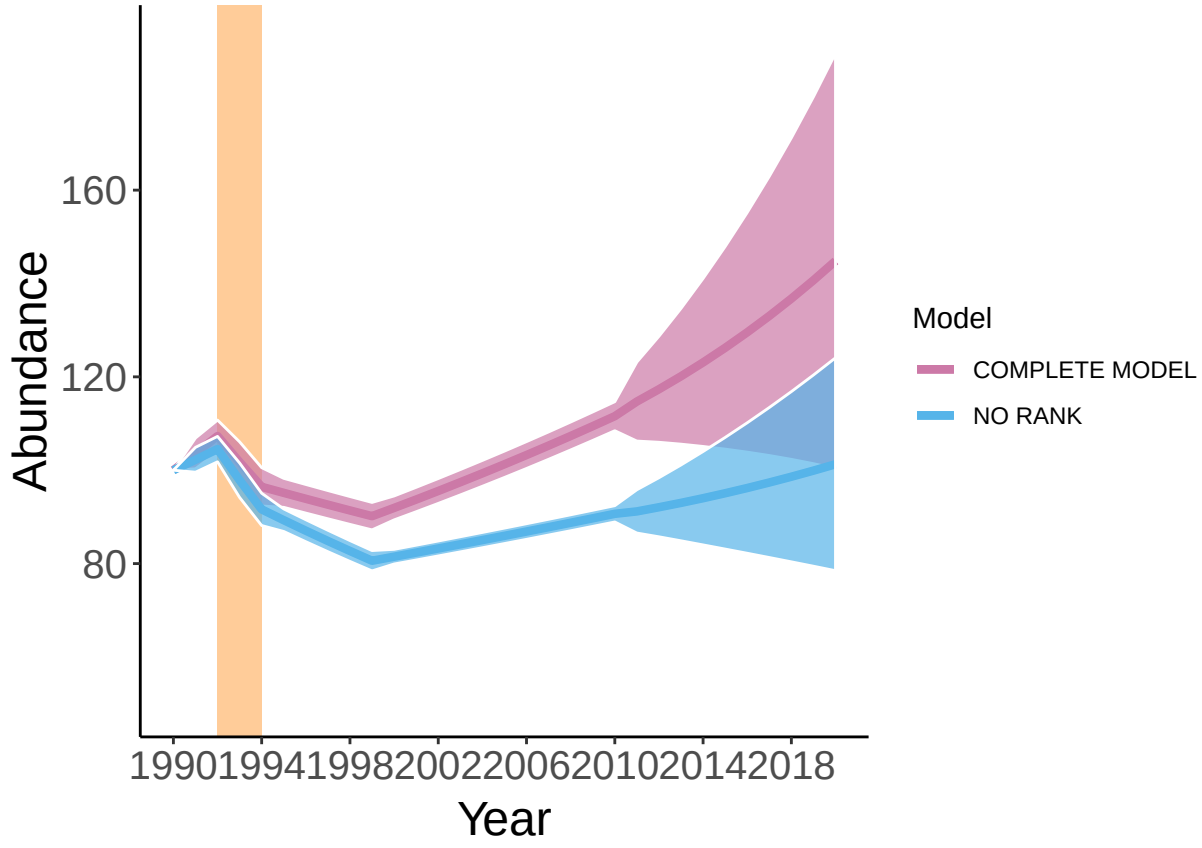
Then we plot Figure 5:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
p <- ggplot(data=matNEW, aes(x=Time1, y=popproj, ymin=varmoins, ymax=varplus)) +

  geom_rect(aes(xmin=1992, xmax=1994, ymin=-Inf, ymax=Inf), fill="#FFCC99",alpha=0.5) +

  geom_line(data=matNEW[matNEW$Model=="normal", ],aes(colour="myline1",x=Time1, y=popproj),size=1.5) +
  geom_ribbon(data=matNEW[matNEW$Model=="normal", ],alpha=0.7, color= "white",fill ="#CC79A7")  + xlab(

  geom_line(data=matNEW[matNEW$Model=="NR", ],aes(colour="myline2",x=Time1, y=popproj),size=1.5) +
  geom_ribbon(data=matNEW[matNEW$Model=="NR", ],alpha=0.7, color= "white",fill ="#56B4E9") + xlab("Year

  scale_colour_manual(name="Model",values=c(myline1="#CC79A7", myline2="#56B4E9"),
                      labels = c("COMPLETE MODEL", "NO RANK")) +
  ylim(50, 192.5) +

  scale_x_continuous(breaks = round(seq(min(matNEW$Time1), max(matNEW$Time1), by = 4),1))


p + theme_bw() +

  theme(
    axis.line = element_line(colour = "black"),
    axis.text = element_text(size=15),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
```

```
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.title.x = element_text(colour="black",size=18,angle=0,hjust=.5,vjust=0,face="plain"),
  axis.title.y = element_text(colour="black",size=18),
  plot.margin = unit(c(7.5, 12.5, 5.5, 5.5), "points")
)
```



2) Sensitivity analysis of the population's growth rate (λ)(Fig 2)

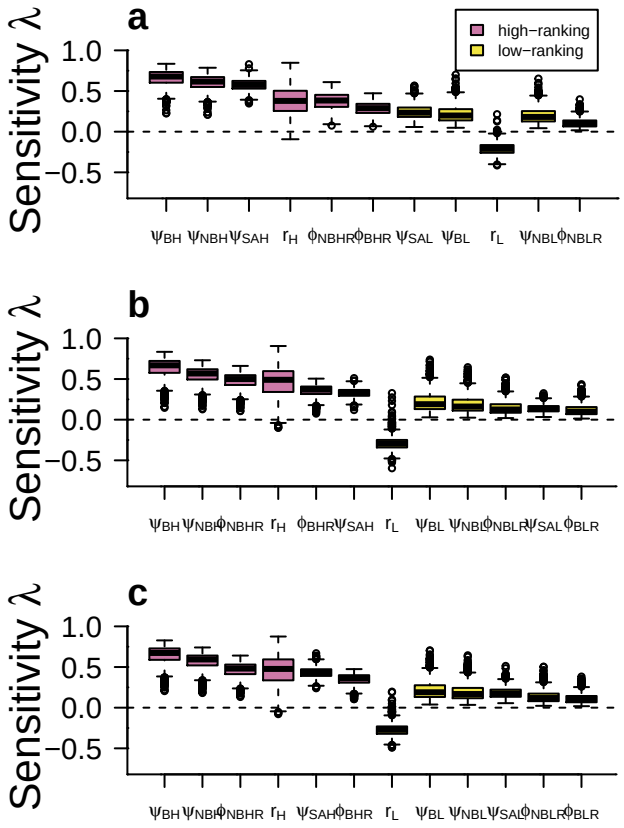
To determine which parameters contributed most to(λ) and predict the results of future changes in parameter estimates, we performed a sensitivity analysis. When elements of a population matrix are composed of several vital rates, the classical first order sensitivity analysis is not recommended, as it does not allow disentangling the effects of demographic, social and infection parameters. Therefore, we conducted lower-level sensitivity analyses for (λ). In the source R file **MC_simulations** we applied the function 'vitalsens' from the R package 'popbio which evaluate the expression of the matrix projection using all parameter values of the MonteCarlo iterations.

```
figtab<-list((tabprepidem[[8]]), (tabepidem[[8]]), (tabpost1[[8]])) # removes the character names and
names(figtab)<-c("sens_preepidem","sens_epidem", "sens_postepidem1")

colnames(figtab[[1]])<-names(tabepidem[[5]])
colnames(figtab[[2]])<-names(tabepidem[[5]])
colnames(figtab[[3]])<-names(tabepidem[[5]])

meanfigtab<-rbind(colMeans(tabprepidem[[8]]), colMeans(tabepidem[[8]]), colMeans(tabpost1[[8]]))
```

```r
row.names(meanfigtab)<-c("sens_preepidem","sens_epidem", "sens_postepidem1")
colnames(meanfigtab)<-names(tabepidem[[5]])

meanfigtab<-data.frame(t(meanfigtab))

# 2.a) Order the preepidem table by decreasing importance
sens1<-subset(meanfigtab, abs(meanfigtab$sens_preepidem) >= 0.10)
sensN1<-figtab[[1]][, rownames(sens1)[order(abs(sens1[,1]),decreasing=TRUE)]] # Order from highest im

  # to check the sequence of parameters later included in the "titre":
sens1_order <- sens1[order(abs(sens1[,1]),decreasing =TRUE),]


# 2.b) Order the epidem table by decreasing importance
sens2<-subset(meanfigtab, abs(meanfigtab$sens_epidem) >= 0.10)
sensN2<-figtab[[2]][, rownames(sens2)[order(abs(sens2[,2]),decreasing=TRUE)]] # Order from highest im

sens2_order <- sens2[order(abs(sens2[,2]),decreasing =TRUE),]

# 2.c) Order the post-epidem table by decreasing importance
sens3<-subset(meanfigtab, abs(meanfigtab$sens_postepidem) >= 0.10)
sensN3<-figtab[[3]][, rownames(sens3)[order(abs(sens3[,3]),decreasing=TRUE)]] # Order from highest im

sens3_order <- sens3[order(abs(sens3[,3]),decreasing =TRUE),]


# prepare the labels and colors
titre<-c(expression(psi["BH"]), expression(psi["NBH"]), expression(psi["SAH"]), expression(r["H"]), e

titre2<-c(expression(psi["BH"]), expression(psi["NBH"]), expression(phi["NBHR"]),  expression(r["H"])

titre3<-c(expression(psi["BH"]), expression(psi["NBH"]), expression(phi["NBHR"]),  expression(r["H"])


color = rep(NA, length=ncol(sensN1))

color[which(colnames(sensN1)=="bSH")] = "#CC79A7"
color[which(colnames(sensN1)=="bBH")] = "#CC79A7"
color[which(colnames(sensN1)=="bNBH")] = "#CC79A7"

color[which(colnames(sensN1)=="bSL")] = "#F0E442"
color[which(colnames(sensN1)=="bBL")] = "#F0E442"
color[which(colnames(sensN1)=="bNBL")] = "#F0E442"

color[which(colnames(sensN1)=="rHH.R")] = "#CC79A7"
color[which(colnames(sensN1)=="rLL.R")] = "#F0E442"

color[which(colnames(sensN1)=="phiBHR")] = "#CC79A7"
color[which(colnames(sensN1)=="phiNBHR")] = "#CC79A7"

color[which(colnames(sensN1)=="phiNBLR")] = "#F0E442"
```

```
color2 = rep(NA, length=ncol(sensN2))

color2[which(colnames(sensN2)=="bSH")] = "#CC79A7"
color2[which(colnames(sensN2)=="bBH")] = "#CC79A7"
color2[which(colnames(sensN2)=="bNBH")] = "#CC79A7"

color2[which(colnames(sensN2)=="bSL")] = "#F0E442"
color2[which(colnames(sensN2)=="bBL")] = "#F0E442"
color2[which(colnames(sensN2)=="bNBL")] = "#F0E442"

color2[which(colnames(sensN2)=="rHH.R")] = "#CC79A7"
color2[which(colnames(sensN2)=="rLL.R")] = "#F0E442"

color2[which(colnames(sensN2)=="phiBHR")] = "#CC79A7"
color2[which(colnames(sensN2)=="phiNBHR")] = "#CC79A7"

color2[which(colnames(sensN2)=="phiBLR")] = "#F0E442"
color2[which(colnames(sensN2)=="phiNBLR")] = "#F0E442"



color3 = rep(NA, length=ncol(sensN3))

color3[which(colnames(sensN3)=="bSH")] = "#CC79A7"
color3[which(colnames(sensN3)=="bBH")] = "#CC79A7"
color3[which(colnames(sensN3)=="bNBH")] = "#CC79A7"

color3[which(colnames(sensN3)=="bSL")] = "#F0E442"
color3[which(colnames(sensN3)=="bBL")] = "#F0E442"
color3[which(colnames(sensN3)=="bNBL")] = "#F0E442"

color3[which(colnames(sensN3)=="rHH.R")] = "#CC79A7"
color3[which(colnames(sensN3)=="rLL.R")] = "#F0E442"

color3[which(colnames(sensN3)=="phiBHR")] = "#CC79A7"
color3[which(colnames(sensN3)=="phiNBHR")] = "#CC79A7"

color3[which(colnames(sensN3)=="phiBLR")] = "#F0E442"
color3[which(colnames(sensN3)=="phiNBLR")] = "#F0E442"



# pdf("Figure2SensLambdaNew.pdf",width=7,height=5) #uncomment this line to get the figure in pdf


nf <- layout(matrix(c(0,1,0,0,2,0,0,3,0),3, byrow = TRUE),
             widths= c(1.2,2,1.2), heights=c(2,2,2)) # widths for columns,heights for rows
#layout.show(nf)

par(mar =c(3.3,3,2,0))


# x1 and y1 to draw the axes
```

```
x1 <- c(-0.1:13)
y1 <- c(-0.82,-0.50,-0.25,0,0.25,0.50,0.75,1)

# x2 and y2 to add labels and tickmarks
x2 <- c(1:11)
y2 <- c(-0.50,0,0.5,1)


# pre-epidemic period:

boxplot(sensN1,  col=color, ylim=c(-0.75,1), axes=F)
mtext(side = 3,text=expression(paste(bold("a"))),adj=0, line =0.4,cex = 1.2)

axis(1,at=x1,las=1,tck=0, labels=NA)
axis(2,at=y1,las=1,tck=0, labels =NA)

axis(1,at=x2,las=1,cex.axis=0.85, labels=titre)
axis(2,at=y2,las=1,cex.axis=1.4)

mtext(side = 2, text =expression(paste("Sensitivity ", lambda)), line = 2.9, cex=1.3)

abline(h=0, lty=2)

legend(8,1.5, legend=c("high-ranking", "low-ranking"),
fill=c("#CC79A7", "#F0E442"), cex=0.8, xpd = T)

# epidemic period
x2 <- c(1:12)

boxplot(sensN2, axes=F, col=color2, ylim=c(-0.75,1))
mtext(side = 3,text=expression(paste(bold("b"))),adj=0, line =0.4,cex = 1.2)
axis(1,at=x1,las=1,tck=0, labels=NA)
axis(2,at=y1,las=1,tck=0, labels =NA)
axis(1,at=x2,las=1,cex.axis=0.85, labels=titre2)
axis(2,at=y2,las=1,cex.axis=1.4)
mtext(side = 2, text =expression(paste("Sensitivity ", lambda)), line = 2.9, cex=1.3)
abline(h=0, lty=2)

# post-epidemic period

boxplot(sensN3, axes=F, col=color3, ylim=c(-0.75,1))
mtext(side = 3,text=expression(paste(bold("c"))),adj=0, line =0.4,cex = 1.2)

axis(1,at=x1,las=1,tck=0, labels=NA)
axis(2,at=y1,las=1,tck=0, labels =NA)
axis(1,at=x2,las=1,cex.axis=0.85, labels=titre3)
axis(2,at=y2,las=1,cex.axis=1.4)
mtext(side = 2, text =expression(paste("Sensitivity ", lambda)), line = 2.9, cex=1.3)
abline(h=0, lty=2)
```




```
# dev.off() # uncomment this line if you want to get only the figure in pdf
```

3) Sensitivity analysis of R0 (Fig 3)

Here we determine which parameters contributed most to variation in R0 in order to predict future disease dynamics with changes in parameter (θ). Because we did not have a symbolic expression of the next generation matrix, we could not apply the vitalsens function as we did for calculating the sensitvity of lambda. Instead, we performed the sensitivity analysis of R0 by coding the following equation $\delta(R0)/\delta(\theta)$.

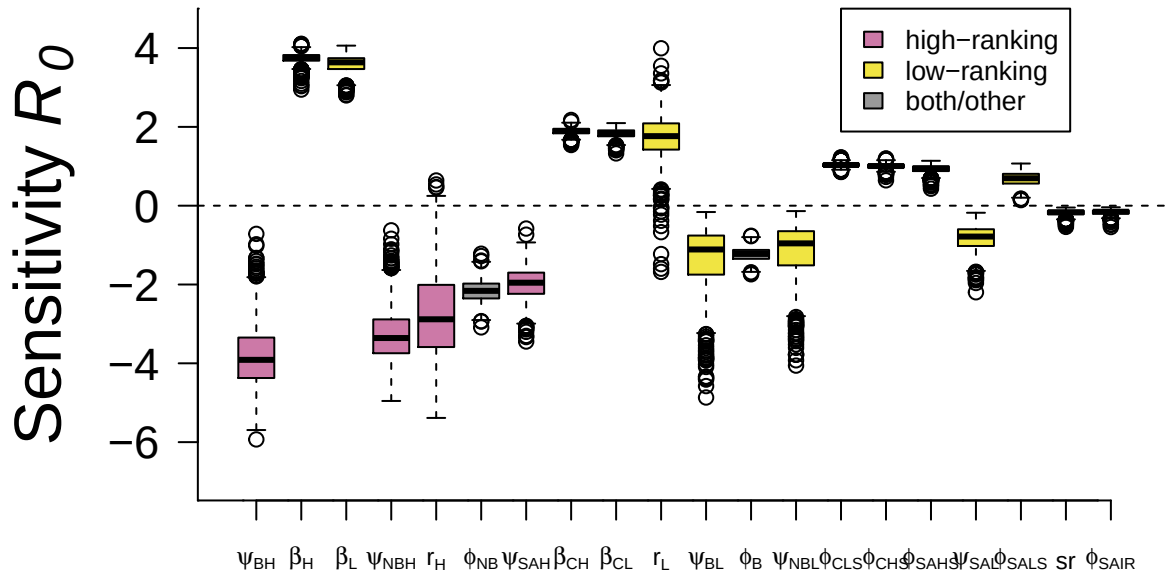
Now we plot the sensitivity values of R0:

```
fulltab<-matrix(0, nrow=MCiter, ncol=23)
 colnames(fulltab)<-1:23
 for(i in 1:23)
 {
 fulltab[,i]<-tabepidem[[10]][[i]]$sensR0
 colnames(fulltab)[i]<-tabepidem[[10]][[i]]$param
 }


 fulltab<-as.data.frame(fulltab)



 # 2) Order by decreasing importance

 # here we first extract the mean values of the sensitvity to R0, we select those which absolute value
```

```r
figtab<-cbind(as.numeric(tabepidem[[9]][,2]), as.numeric(tabepidem[[9]][,3])) # extract the mean sens

colnames(figtab)<-c("sens_epidem","sd_epidem")

figtab<-as.data.frame(figtab)

figtab$sens_epidem[12:15]<- 1 - figtab$sens_epidem[12:15]

figtab$param<- as.character(tabepidem[[9]][,1])


#-------- Epidem
# 1) Select mean sensitivity values whose abs. value is higher than 10%
sensNew<-subset(figtab, abs(figtab$sens_epidem) >= 0.10)

# 2) Order by decreasing importance
sensNew1<-fulltab[order(abs(sensNew$sens_epidem),decreasing=TRUE),]

sens<-NULL
for(i in 1:20){
sens<-c(sens,which(colnames(fulltab) == sensNew$param[order(abs(sensNew$sens_epidem),decreasing=TRUE)]




sensN2<-fulltab[,sens] #
# SENSITIVITY TO INFECTION (1 - sensitivity to the transition from Susceptible to Susceptible)
sensN2[,3]<-1-sensN2[,3]
sensN2[,2]<-1-sensN2[,2]
sensN2[,8]<-1-sensN2[,8]
sensN2[,9]<-1-sensN2[,9]

titreR0<-c("Breeding_BH", "Infection_SA&B&NBH", "Infection_SA&B&NBL", "Breeding_NBH", "Staying_H", "Su




titrefig<-c (expression(psi["BH"]), expression(beta["H"]), expression(beta["L"]), expression(psi["NBH"
expression(sr),expression(phi["SAIR"]))


colnames(sensN2)<-titreR0
# -- Plot; decreasing importance

par(mar=c(5.1,5.1,4.1,2.1))

color = rep(NA, length=length(sensNew1$param))


# Survival
# cubs
color[which(colnames(sensN2)=="Survival_CHS")] = "#CC79A7"
```

```
color[which(colnames(sensN2)=="Survival_CLS")] = "#F0E442"

# subadults
color[which(colnames(sensN2)=="Survival_SAHS")] = "#CC79A7"
color[which(colnames(sensN2)=="Survival_SAI&R")] = "#999999"

color[which(colnames(sensN2)=="Survival_SALS")] = "#F0E442"

#breeders
color[which(colnames(sensN2)=="Survival_B")] = "#999999"

#Nonbreeders
color[which(colnames(sensN2)=="Survival_NB")] = "#999999"

#rank transitions
color[which(colnames(sensN2)=="Staying_H")] = "#CC79A7"
color[which(colnames(sensN2)=="Staying_L")] = "#F0E442"

# Infection proba
color[which(colnames(sensN2)=="Infection_CH")] = "#CC79A7"
color[which(colnames(sensN2)=="Infection_CL")] = "#F0E442"

color[which(colnames(sensN2)=="Infection_SA&B&NBH")] = "#CC79A7"
color[which(colnames(sensN2)=="Infection_SA&B&NBL")] = "#F0E442"

# Breeding
color[which(colnames(sensN2)=="Breeding_BH")] = "#CC79A7"
color[which(colnames(sensN2)=="Breeding_NBH")] = "#CC79A7"
color[which(colnames(sensN2)=="Breeding_SAH")] = "#CC79A7"

color[which(colnames(sensN2)=="Breeding_BL")] = "#F0E442"
color[which(colnames(sensN2)=="Breeding_NBL")] = "#F0E442"
color[which(colnames(sensN2)=="Breeding_SAL")] = "#F0E442"

#Sex ratio
color[which(colnames(sensN2)=="SexRatio")] = "#999999"




# pdf("SensitivityR0.pdf",width=7,height=5)


colnames(sensN2)<-NULL
boxplot(sensN2,  col=color, ylim=c(-7,5), axes=F)

# x1 and y1 to draw the axes
x1 <- c(-0.3:20)
y1 <- c(-8,5)

# x2 and y2 to add labels and tickmarks
x2 <- c(1:20)
y2 <- c(-6,-4,-2,0,2,4)
```

```
axis(1,at=x1,las=1,tck=0, labels=NA)
axis(2,at=y1,las=1,tck=0, labels =NA)

axis(1,at=x2,las=1,cex.axis=0.75, labels=titrefig)
axis(2,at=y2,las=1,cex.axis=1.4)

abline(h = 0, lty =2)

mtext(side = 2, text =expression(paste("Sensitivity ", italic(R["0"]))), line = 3, cex=2.1)

legend(14,5, legend=c("high-ranking", "low-ranking", "both/other"),
fill=c("#CC79A7", "#F0E442", "#999999"), cex=0.8)
```




```
#dev.off()
```

```
end_time <- Sys.time()
run_time <-end_time-start_time
round(run_time,digits = 2)
```

```
## Time difference of 11.59 hours
```

References (R packages)

Stubben, C. & Milligan, B. Estimating and analyzing demographic models using the popbio package in R. J. Stat. Softw 22, 1-23 (2007).

Stott, I., Hodgson, D. J. & Townley, S. popdemo: an R package for population demography using projection matrix analysis. Methods Ecol. Evol. 3, 797-802 (2012).