

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

S. Benhaïem, L. Marescot

2018-08-02 12:42:28

This code presents the stochastic analysis of the matrix model. To calculate standard deviations of (λ) and R_0 (Figure 1), confidence intervals for the sensitivity analysis of (λ) and R_0 (Figures 2,3) and to describe changes in population size over time while accounting for parameter uncertainty (Figure 5), we used Monte Carlo iterations. It 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 to calculate the mean + SD of the population’s growth rate (Fig 1), of R_0 and population abundance (Fig 5).
 - a) Plotting mean + SD of population’s growth rate and R_0 (Fig 1)
 - b) Plotting changes in population abundance - complete model and “no rank” model (Fig 5)
- 2) Sensitivity analysis of the population’s growth rate (λ)(Fig 2)
- 3) Sensitivity analysis of R_0 (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 the population’s growth rate (Fig1A) R_0 (Fig1B) and population abundance (Fig5)

First, we create a function to compile the population indicators we are interested in; i.e. the mean population growth rate, R_0 , the population size, the sensitivity values of growth rate and R_0 and their associated standard deviations.

```
start_time <- Sys.time()

library(popdemo)

## Warning: package 'popdemo' was built under R version 3.5.1
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 (Mean +SD)
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 RO (Mean +SD)
#preparing empty table where we get the mean + SD of sens values
SensR0<- matrix(data = 0, nrow=ncol(theta), ncol=3)# here empty

#####----- running the function which calculates the sensitivity of RO
if(t == "epidem")
{
# RO
# To do so we extract the dominant eigen value from the next generation matrix using the lambda function
allrnot<-unlist(lapply(NGMstoch, lambda))
mean_rnot<-mean(allrnot)
sd_rnot<-sd(allrnot)
tabmoy[2,1]<-mean_rnot
tabmoy[2,2]<-sd_rnot

for(r in 1:ncol(theta))
{
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, SensR0))
}

```

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<-100 # Number of Monte Carlo iterations (put 1000)
```

```
checkNodisease <- FALSE # Check if we are in the projection phase were the disease is extinct (then set
```

```
# 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
```

```
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
```

```
## [1] FALSE
```

```
popsize0<-tabpost2[[2]][12]
```

```
source('MC_simulations.R')
```

```
tabprojection<-results(Mproj, NGMstoch, popvec,senslambda, theta)
```

a) Plotting mean + SD of population's growth rate and R0 (Fig1)

```
data2 <- matrix(0,3,2)
```

```
colnames(data2) <- c("Lambda", "Lambda.SD")
```

```

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

data2[1,1]<-tabprepidem[[1]][1,1] # mean lambda
data2[1,2]<-tabprepidem[[1]][1,2] # SD lambda

data2[2,1]<-tabepidem[[1]][1,1] # mean lambda
data2[2,2]<-tabepidem[[1]][1,2] # SD lambda

data2[3,1]<-tabpost1[[1]][1,1] # mean lambda
data2[3,2]<-tabpost1[[1]][1,2] # SD lambda

par(mfrow=c(1,2))

x<-c(0.5,1,2,3,3.5)

colvec <- c("white", "black", "#CC6600", "black", "white")

# ----- > lambda

mean<-c(0,data2[c("pre-epidem", "epidem", "post-epidem1"),1],0)
sd<-c(0,data2[c("pre-epidem", "epidem", "post-epidem1"),2],0)

mean<-as.numeric(mean)
sd<-as.numeric(sd)

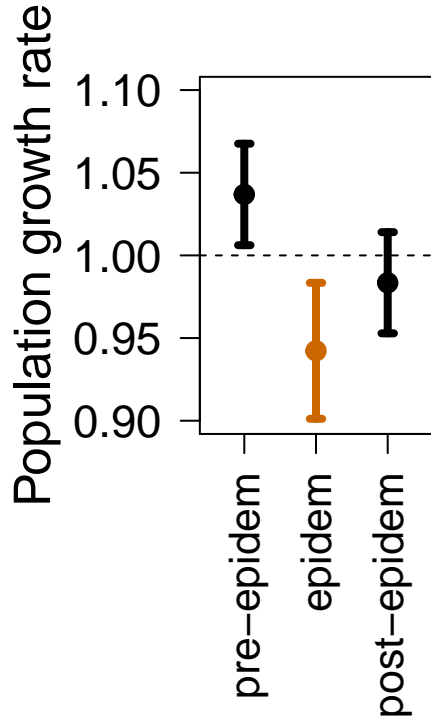
library(plotrix)

par(mar=c(8.1,5.1,5.1,5.1))
plotCI(x,mean,
       uiw=sd, err="y",las=1, ylab = "", cex=1.4,xaxt="n", ylim=c(0.90,1.10),xlab = "", cex.axis=1.4,
       lwd=4, pch=20, col =colvec)

names(x)<-c("pre-epidem", "epidem", "post-epidem", "")
colvec2 <-c("black", "#CC6600", "black")

axis(side=1,at=c(1:5),label=names(x),cex.axis=1.3, las =2)
mtext(side = 2, text = "Population growth rate", line = 3.7, cex=1.6)
abline(h=1, lty=2)

```



In the following table, we displayed the basic reproduction number during the outbreak

Basic reproduction number	Mean estimate	SE
R0	5.6444986	0.6785446

b) Plotting changes in population abundance - complete model and “no rank” model Fig5)

We can then describe past (1990-2010) and predicted (2011-2020) temporal changes in population abundance (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 population growth rate for each year of the study period accounting for temporal variations in the mean estimates between the pre-epidemic, epidemic and post-epidemic phases. We described with a Markov chain the changes in population size by multiplying population vector of a given year by λ and reinitialize 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),r

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

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])
}

p<-1
for(t in 22:32)
{
  p<-p+1
  popproj[t]<- tabprojection[[2]][p]
  varplus[t]<- tabprojection[[2]][p] + tabprojection[[3]][p]
  varmoins[t]<- tabprojection[[2]][p] - tabprojection[[3]][p]
}

# Here We delete the meaningless growth rate obtained at the first year of predicitions phase, which cor
popproj<-popproj[-22]
varplus<-varplus [-22]
varmoins<-varmoins[-22]

####

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  103.68685 100.61738 106.75633 normal    2  1991
## 3  107.50964 104.32700 110.69228 normal    3  1992
## 4  101.30481  96.88471 105.72491 normal    4  1993

```

## 5	95.45809	91.29309	99.62308	normal	5	1994
## 6	93.87816	90.95764	96.79868	normal	6	1995
## 7	92.32437	89.45219	95.19656	normal	7	1996
## 8	90.79631	87.97166	93.62095	normal	8	1997
## 9	89.29353	86.51564	92.07143	normal	9	1998
## 10	87.81563	85.08371	90.54755	normal	10	1999
## 11	88.29911	85.77252	90.82569	normal	11	2000
## 12	88.78524	86.24475	91.32574	normal	12	2001
## 13	89.27406	86.71957	91.82854	normal	13	2002
## 14	89.76556	87.19701	92.33411	normal	14	2003
## 15	90.25977	87.67708	92.84246	normal	15	2004
## 16	90.75670	88.15979	93.35361	normal	16	2005
## 17	91.25637	88.64516	93.86757	normal	17	2006
## 18	91.75878	89.13320	94.38437	normal	18	2007
## 19	92.26397	89.62393	94.90401	normal	19	2008
## 20	92.77193	90.11736	95.42651	normal	20	2009
## 21	93.28270	90.61351	95.95188	normal	21	2010
## 22	95.06880	90.77987	99.35772	normal	22	2011
## 23	95.92357	89.63246	102.21467	normal	23	2012
## 24	96.82557	88.31195	105.33920	normal	24	2013
## 25	97.79994	86.94985	108.65002	normal	25	2014
## 26	98.84953	85.57365	112.12540	normal	26	2015
## 27	99.97409	84.18646	115.76172	normal	27	2016
## 28	101.17449	82.78731	119.56167	normal	28	2017
## 29	102.45201	81.37302	123.53101	normal	29	2018
## 30	103.80828	79.93934	127.67721	normal	30	2019
## 31	105.24517	78.48144	132.00890	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
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 model with and without rank effect.

```

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

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

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 2000-2010

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]
}

# delete 2011/2012
popproj<-popproj[-c(22,23)]
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	103.68685	100.61738	106.75633	normal	2	1991
## 3	107.50964	104.32700	110.69228	normal	3	1992
## 4	101.30481	96.88471	105.72491	normal	4	1993
## 5	95.45809	91.29309	99.62308	normal	5	1994
## 6	93.87816	90.95764	96.79868	normal	6	1995
## 7	92.32437	89.45219	95.19656	normal	7	1996
## 8	90.79631	87.97166	93.62095	normal	8	1997
## 9	89.29353	86.51564	92.07143	normal	9	1998
## 10	87.81563	85.08371	90.54755	normal	10	1999
## 11	88.29911	85.77252	90.82569	normal	11	2000
## 12	88.78524	86.24475	91.32574	normal	12	2001
## 13	89.27406	86.71957	91.82854	normal	13	2002
## 14	89.76556	87.19701	92.33411	normal	14	2003
## 15	90.25977	87.67708	92.84246	normal	15	2004
## 16	90.75670	88.15979	93.35361	normal	16	2005
## 17	91.25637	88.64516	93.86757	normal	17	2006
## 18	91.75878	89.13320	94.38437	normal	18	2007
## 19	92.26397	89.62393	94.90401	normal	19	2008
## 20	92.77193	90.11736	95.42651	normal	20	2009
## 21	93.28270	90.61351	95.95188	normal	21	2010
## 22	95.06880	90.77987	99.35772	normal	22	2011
## 23	95.92357	89.63246	102.21467	normal	23	2012
## 24	96.82557	88.31195	105.33920	normal	24	2013
## 25	97.79994	86.94985	108.65002	normal	25	2014
## 26	98.84953	85.57365	112.12540	normal	26	2015
## 27	99.97409	84.18646	115.76172	normal	27	2016
## 28	101.17449	82.78731	119.56167	normal	28	2017
## 29	102.45201	81.37302	123.53101	normal	29	2018
## 30	103.80828	79.93934	127.67721	normal	30	2019
## 31	105.24517	78.48144	132.00890	normal	31	2020
## 32	100.00000	100.00000	100.00000	NR	1	1990
## 33	102.13553	98.93580	105.33527	NR	2	1991
## 34	104.31667	101.04860	107.58475	NR	3	1992
## 35	97.06355	93.67911	100.44800	NR	4	1993
## 36	90.31474	87.16562	93.46387	NR	5	1994
## 37	88.29826	85.96688	90.62963	NR	6	1995
## 38	86.32679	84.04747	88.60612	NR	7	1996
## 39	84.39935	82.17092	86.62778	NR	8	1997
## 40	82.51494	80.33626	84.69361	NR	9	1998

## 41	80.67260	78.54257	82.80263	NR	10	1999
## 42	80.61194	79.12680	82.09707	NR	11	2000
## 43	80.55132	79.06730	82.03534	NR	12	2001
## 44	80.49074	79.00784	81.97364	NR	13	2002
## 45	80.43021	78.94842	81.91200	NR	14	2003
## 46	80.36973	78.88905	81.85040	NR	15	2004
## 47	80.30929	78.82973	81.78885	NR	16	2005
## 48	80.24889	78.77045	81.72734	NR	17	2006
## 49	80.18855	78.71121	81.66588	NR	18	2007
## 50	80.12824	78.65202	81.60447	NR	19	2008
## 51	80.06799	78.59287	81.54310	NR	20	2009
## 52	80.00778	78.53377	81.48178	NR	21	2010
## 53	79.33285	76.06254	82.60317	NR	22	2011
## 54	79.32775	74.79797	83.85753	NR	23	2012
## 55	79.36075	73.52799	85.19350	NR	24	2013
## 56	79.43045	72.27323	86.58766	NR	25	2014
## 57	79.53503	71.03809	88.03197	NR	26	2015
## 58	79.67321	69.82351	89.52291	NR	27	2016
## 59	79.84388	68.62859	91.05916	NR	28	2017
## 60	80.04609	67.45170	92.64048	NR	29	2018
## 61	80.27906	66.29098	94.26714	NR	30	2019
## 62	80.54213	65.14448	95.93979	NR	31	2020

Then we plot Figure 5:

```
library(ggplot2)
```

```
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("Year")
```

```
  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 MODEL")) +
```

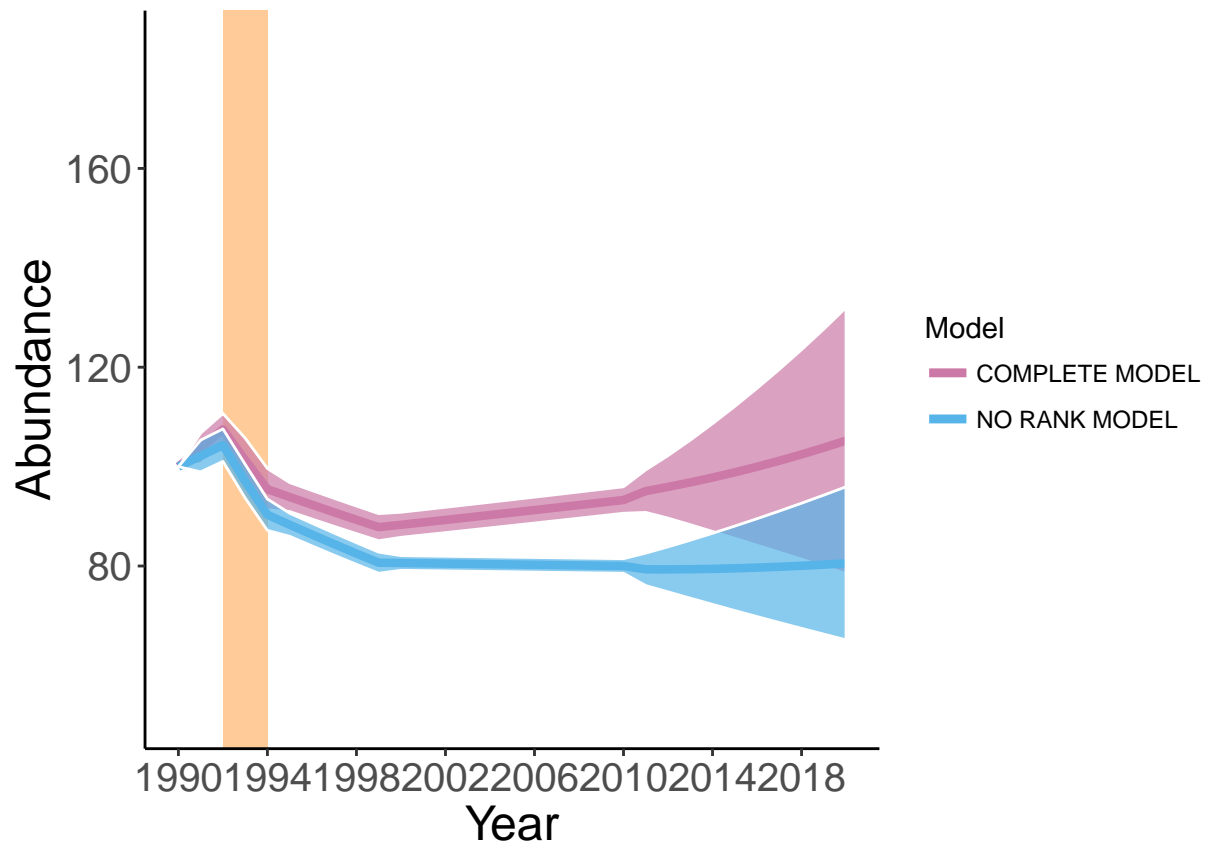
```
  ylim(50, 185) +
```

```
  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")
)
```



3) Sensitivity analyses of the population's growth rate (λ)(Fig2)

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 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.

```
nf <- layout(matrix(c(0,0,0,0,1,0,0,2,0,0,3,0,0,0,0),3, byrow = FALSE),
              widths= c(0.1,1.8,1.8,1.8,0.1), heights=c(1,3.5,1,1,0)) # widths for columns,heights for r
layout.show(nf)
```

1	2	3
---	---	---

```

figtab<-cbind(as.numeric(tabprepidem[[4]]), as.numeric(tabprepidem[[5]]), as.numeric(tabepidem[[4]]), as.numeric(tabepidem[[5]]))

colnames(figtab)<-c("sens_preepidem","sd_preepidem", "sens_epidem","sd_epidem", "sens_postepidem1","sens_postepidem2")

figtab<-as.data.frame(figtab)
figtab$param<-names(tabepidem[[5]])
#----- Pre-Epidem
# 1) Select sens values whose abs. value is higher than 10%
sensNew<-subset(figtab, abs(figtab$sens_preepidem) >= 0.10)

# 2) Order by decreasing importance
sensNew1<-sensNew[order(abs(sensNew$sens_preepidem),decreasing=TRUE),] # Order from highest impact on l

# -- Plot; decreasing importance

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

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

color[which(sensNew1$param=="bSH")] = "olivedrab1"
color[which(sensNew1$param=="bBH")] = "olivedrab3"
color[which(sensNew1$param=="bNBH")] = "olivedrab4"

color[which(sensNew1$param=="bSL")] = "olivedrab1"
color[which(sensNew1$param=="bBL")] = "olivedrab3"
color[which(sensNew1$param=="bNBL")] = "olivedrab4"

```

```

color[which(sensNew1$param=="rHH.R")] = "orange1"
color[which(sensNew1$param=="rLL.R")] = "orange1"

color[which(sensNew1$param=="phiBHR")] = "royalblue4"
color[which(sensNew1$param=="phiNBHR")] = "steelblue1"

#color[which(sensNew1$param=="phiBHS")] = "skyblue3"
#color[which(sensNew1$param=="phiNBHS")] = "skyblue1"

#color[which(sensNew1$param=="phiBLR")] = "royalblue4"
color[which(sensNew1$param=="phiNBLR")] = "steelblue1"

#color[which(sensNew1$param=="phiBLS")] = "skyblue3"
#color[which(sensNew1$param=="phiNBLS")] = "skyblue1"

dashed = rep(NA, length=length(sensNew1$param))
dashed[which(sensNew1$param=="bSH")] = NA
dashed[which(sensNew1$param=="bBH")] = NA
dashed[which(sensNew1$param=="bNBH")] = NA

dashed[which(sensNew1$param=="bSL")] = 30
dashed[which(sensNew1$param=="bBL")] = 30
dashed[which(sensNew1$param=="bNBL")] = 30

dashed[which(sensNew1$param=="rHH.R")] = NA
dashed[which(sensNew1$param=="rLL.R")] = 30

dashed[which(sensNew1$param=="phiBHR")] = NA
dashed[which(sensNew1$param=="phiNBHR")] = NA

#dashed[which(sensNew1$param=="phiBHS")] = NA
#dashed[which(sensNew1$param=="phiNBHS")] = NA

#dashed[which(sensNew1$param=="phiBLR")] = 80
dashed[which(sensNew1$param=="phiNBLR")] = 30

#dashed[which(sensNew1$param=="phiBLS")] = 80
#dashed[which(sensNew1$param=="phiNBLS")] = 80

bp<-barplot(sensNew1$sens_preepidem, las=1,col=color, ylab=(expression("Sensitivity " * lambda)), cex =0.8,
  xpd=FALSE, main="", ylim=c(-0.5,1.1), cex.lab = 2.2,
  cex.axis= 1.4,xlim=c(0,13), density=dashed, angle =0)

lines(c(-0.75,20),c(0,0))

arrows(bp, sensNew1$sens_preepidem - sensNew1$sd_preepidem , bp,
  sensNew1$sens_preepidem + sensNew1$sd_preepidem , lwd = 1.5, angle = 90,
  code = 3, length = 0.03)
#mtext(side = 3,text=c("a pre-epidem"),adj=0, line =1.7,cex = 1.3)

mtext(side = 3,text= expression(paste(bold("a"))),adj=0, line =1.7,cex = 1.6)

```

```

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

# 2) Order by decreasing importance
sensNew1<-sensNew[order(abs(sensNew$sens_epidem),decreasing=TRUE),] # Order from highest impact on lamb
#n =12
# -- Plot; decreasing importance

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

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

color[which(sensNew1$param=="bSH")] = "olivedrab1"
color[which(sensNew1$param=="bBH")] = "olivedrab3"
color[which(sensNew1$param=="bNBH")] = "olivedrab4"

color[which(sensNew1$param=="bSL")] = "olivedrab1"
color[which(sensNew1$param=="bBL")] = "olivedrab3"
color[which(sensNew1$param=="bNBL")] = "olivedrab4"

color[which(sensNew1$param=="rHH.R")] = "orange1"
color[which(sensNew1$param=="rLL.R")] = "orange1"

color[which(sensNew1$param=="phiBHR")] = "royalblue4"
color[which(sensNew1$param=="phiNBHR")] = "steelblue1"

color[which(sensNew1$param=="phiBHS")] = "skyblue3"
color[which(sensNew1$param=="phiNBHS")] = "skyblue1"

color[which(sensNew1$param=="phiBLR")] = "royalblue4"
color[which(sensNew1$param=="phiNBLR")] = "steelblue1"

#color[which(sensNew1$param=="phiBLS")] = "skyblue3"
#color[which(sensNew1$param=="phiNBLS")] = "skyblue1"

dashed = rep(NA, length=length(sensNew1$param))
dashed[which(sensNew1$param=="bSH")] = NA
dashed[which(sensNew1$param=="bBH")] = NA
dashed[which(sensNew1$param=="bNBH")] = NA

dashed[which(sensNew1$param=="bSL")] = 30
dashed[which(sensNew1$param=="bBL")] = 30
dashed[which(sensNew1$param=="bNBL")] = 30

dashed[which(sensNew1$param=="rHH.R")] = NA
dashed[which(sensNew1$param=="rLL.R")] = 30

dashed[which(sensNew1$param=="phiBHR")] = NA
dashed[which(sensNew1$param=="phiNBHR")] = NA

```

```

#dashed[which(sensNew1$param=="phiBHS")] = NA
#dashed[which(sensNew1$param=="phiNBHS")] = NA

dashed[which(sensNew1$param=="phiBLR")] = 30
dashed[which(sensNew1$param=="phiNBLR")] = 30

#dashed[which(sensNew1$param=="phiBLS")] = 80
#dashed[which(sensNew1$param=="phiNBLS")] = 80

bp<-barplot(sensNew1$sens_epidem, las=1,col=color, ylab=(expression("Sensitivity "*lambda)), cex =0.6,
            xpd=FALSE, main="", ylim=c(-0.5,1.1),
            cex.lab = 2.2, cex.axis= 1.4,xlim=c(0,14),density=dashed, angle =0) # xpd = Should bars be
lines(c(-0.75,20),c(0,0))

arrows(bp, sensNew1$sens_epidem - sensNew1$sd_epidem , bp,
       sensNew1$sens_epidem + sensNew1$sd_epidem , lwd = 1.5, angle = 90,
       code = 3, length = 0.03)

#mtext(side = 3,text=c("b) epidem"),adj=0, line =1.7,cex = 1.3)
mtext(side = 3,text= expression(paste(bold("b"))),adj=0, line =1.7,cex = 1.6)

# Post-epidem

# 1) Select sens values whose abs. value is higher than 10%
sensNew<-subset(figtab, abs(figtab$sens_postepidem1) >= 0.10)

# 2) Order by decreasing importance
sensNew1<-sensNew[order(abs(sensNew$sens_postepidem1),decreasing=TRUE),] # Order from highest impact on

# -- Plot; decreasing importance

par(mar=c(5.1,5.1,4.1,2.1))
color = rep(NA, length=length(sensNew1$param))

color[which(sensNew1$param=="bSH")] = "olivedrab1"
color[which(sensNew1$param=="bBH")] = "olivedrab3"
color[which(sensNew1$param=="bNBH")] = "olivedrab4"

color[which(sensNew1$param=="bSL")] = "olivedrab1"
color[which(sensNew1$param=="bBL")] = "olivedrab3"
color[which(sensNew1$param=="bNBL")] = "olivedrab4"

color[which(sensNew1$param=="rHH.R")] = "orange1"
color[which(sensNew1$param=="rLL.R")] = "orange1"

color[which(sensNew1$param=="phiBHR")] = "royalblue4"
color[which(sensNew1$param=="phiNBHR")] = "steelblue1"

#color[which(sensNew1$param=="phiBHS")] = "skyblue3"
#color[which(sensNew1$param=="phiNBHS")] = "skyblue1"

color[which(sensNew1$param=="phiBLR")] = "royalblue4"
color[which(sensNew1$param=="phiNBLR")] = "steelblue1"

```

```

#color[which(sensNew1$param=="phiBLS")] = "skyblue3"
#color[which(sensNew1$param=="phiNBLS")] = "skyblue1"

dashed = rep(NA, length=length(sensNew1$param))
dashed[which(sensNew1$param=="bSH")] = NA
dashed[which(sensNew1$param=="bBH")] = NA
dashed[which(sensNew1$param=="bNBH")] = NA

dashed[which(sensNew1$param=="bSL")] = 30
dashed[which(sensNew1$param=="bBL")] = 30
dashed[which(sensNew1$param=="bNBL")] = 30

dashed[which(sensNew1$param=="rHH.R")] = NA
dashed[which(sensNew1$param=="rLL.R")] = 30

dashed[which(sensNew1$param=="phiBHR")] = NA
dashed[which(sensNew1$param=="phiNBHR")] = NA

#dashed[which(sensNew1$param=="phiBHS")] = NA
#dashed[which(sensNew1$param=="phiNBHS")] = NA

dashed[which(sensNew1$param=="phiBLR")] = 30
dashed[which(sensNew1$param=="phiNBLR")] = 30

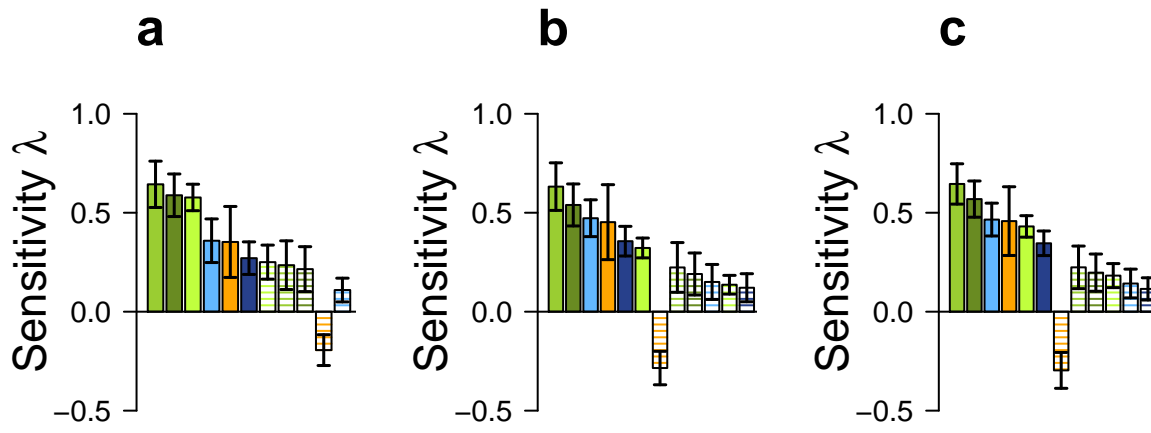
#dashed[which(sensNew1$param=="phiBLS")] = 80
#dashed[which(sensNew1$param=="phiNBLS")] = 80

bp<-barplot(sensNew1$sens_postepidem1, las=1,col=color,ylab=(expression("Sensitivity "*lambda)), cex =0,
            xpd=FALSE, main="",
            density= dashed, angle =0,ylim=c(-0.5,1.1), cex.lab = 2.2, cex.axis= 1.4,xlim=c(0,14)) # xpd
lines(c(-0.75,20),c(0,0))

arrows(bp, sensNew1$sens_postepidem1 - sensNew1$sd_postepidem1, bp,
       sensNew1$sens_postepidem1 + sensNew1$sd_postepidem1, lwd = 1.5, angle = 90,
       code = 3, length = 0.03)

#mtext(side = 3,text=c("c) post-epidem"),adj=0, line =1.7,cex = 1.3)
mtext(side = 3,text= expression(paste(bold("c"))),adj=0, line =1.7,cex = 1.6)

```

Overall legend

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

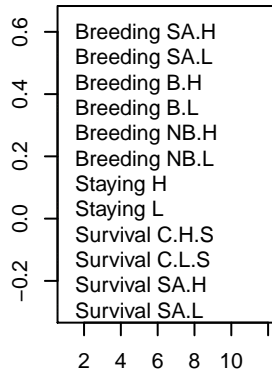
```
plot(sensNew1$sens_postepidem1, pch = "", xlab = "", ylab = "")
```

```
legend("topright",
```

```
c("Breeding SA.H", "Breeding SA.L", "Breeding B.H", "Breeding B.L", "Breeding NB.H", "Breeding NB.L",
  "Staying H", "Staying L",
  "Survival C.H.S", "Survival C.L.S", "Survival SA.H", "Survival SA.L", "Survival SA.I&R", "Survival B", "Survival C.H",
  "Infection C.H", "Infection C.L", "Infection SA.B.NB.H", "Infection SA.B.NB.L",
  "Sex ratio")
```

```
,
  fill= c("olivedrab4", "olivedrab4", "olivedrab1", "olivedrab1", "olivedrab3", "olivedrab3",
    "orange1", "orange1",
    "cadetblue", "cadetblue", "aquamarine4", "aquamarine4", "aquamarine1", "royalblue4", "skyblue3",
    "darkmagenta", "darkmagenta", "darkorchid1", "darkorchid1",
    "gray0")
```

```
,
  density = c(NA, 80, NA, 80, NA, 80,
    NA, 80,
    NA, 80, NA, 80, NA, NA, NA,
    NA, 80, NA, 80,
    NA))
```



- 4) Here we determine which parameters contributed most to variation in R_0 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 sensitivity of lambda. Instead, we performed the sensitivity analysis of R_0 by coding the following equation $\delta(R_0)/\delta(\theta)$.

Now we plot the sensitivity value of R_0

```
figtab<-cbind(as.numeric(tabepidem[[6]][,2]), as.numeric(tabepidem[[6]][,3])) # removes the character n

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[[6]][,1])

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

# 2) Order by decreasing importance
sensNew1<-sensNew[order(abs(sensNew$sens_epidem),decreasing=TRUE),] # Order from highest impact on lamb
```

```

sensNew1$param<-c("Breeding_BH", "Infection_SA&B&NBH", "Infection_SA&B&NBL", "Breeding_NBH", "Staying_H

# -- 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(sensNew1$param=="Survival_CHS")] = "cadetblue"
#color[which(sensNew1$param=="Survival_CHI")] = "cadetblue1"

color[which(sensNew1$param=="Survival_CLS")] = "cadetblue"
#color[which(sensNew1$param=="Survival_CLI")] = "cadetblue1"

# subadults
color[which(sensNew1$param=="Survival_SAHS")] = "aquamarine4"
color[which(sensNew1$param=="Survival_SAI&R")] = "aquamarine1"

color[which(sensNew1$param=="Survival_SALS")] = "aquamarine4"

#breeders
color[which(sensNew1$param=="Survival_B")] = "royalblue4"

#Nonbreeders
color[which(sensNew1$param=="Survival_NB")] = "skyblue3"

#rank transitions
color[which(sensNew1$param=="Staying_H")] = "orange1"
color[which(sensNew1$param=="Staying_L")] = "orange1"

# Infection proba
color[which(sensNew1$param=="Infection_CH")] = "darkmagenta"
color[which(sensNew1$param=="Infection_CL")] = "darkmagenta"

color[which(sensNew1$param=="Infection_SA&B&NBH")] = "darkorchid1"
color[which(sensNew1$param=="Infection_SA&B&NBL")] = "darkorchid1"

# Breeding
color[which(sensNew1$param=="Breeding_BH")] = "olivedrab1"
color[which(sensNew1$param=="Breeding_NBH")] = "olivedrab3"
color[which(sensNew1$param=="Breeding_SAH")] = "olivedrab4"

color[which(sensNew1$param=="Breeding_BL")] = "olivedrab1"
color[which(sensNew1$param=="Breeding_NBL")] = "olivedrab3"
color[which(sensNew1$param=="Breeding_SAL")] = "olivedrab4"

#Sex ratio
color[which(sensNew1$param=="SexRatio")] = "gray0"

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

```

```

dashed[which(sensNew1$param=="Survival_CHS")] = NA
#dashed[which(sensNew1$param=="Survival_CHI")] = "cadetblue1"

dashed[which(sensNew1$param=="Survival_CLS")] = 80
#dashed[which(sensNew1$param=="Survival_CLI")] = "cadetblue1"

# subadults
dashed[which(sensNew1$param=="Survival_SAHS")] = NA
dashed[which(sensNew1$param=="Survival_SAI&R")] = NA

dashed[which(sensNew1$param=="Survival_SALS")] = 80

#breeders
dashed[which(sensNew1$param=="Survival_B")] = NA

#Nonbreeders
dashed[which(sensNew1$param=="Survival_NB")] = NA

#rank transitions
dashed[which(sensNew1$param=="Staying_H")] = NA
dashed[which(sensNew1$param=="Staying_L")] = 80

# Infection proba
dashed[which(sensNew1$param=="Infection_CH")] = NA
dashed[which(sensNew1$param=="Infection_CL")] = 80

dashed[which(sensNew1$param=="Infection_SA&B&NBH")] = NA
dashed[which(sensNew1$param=="Infection_SA&B&NBL")] = 80

# Breeding
dashed[which(sensNew1$param=="Breeding_BH")] = NA
dashed[which(sensNew1$param=="Breeding_NBH")] = NA
dashed[which(sensNew1$param=="Breeding_SAH")] = NA

dashed[which(sensNew1$param=="Breeding_BL")] = 80
dashed[which(sensNew1$param=="Breeding_NBL")] = 80
dashed[which(sensNew1$param=="Breeding_SAL")] = 80

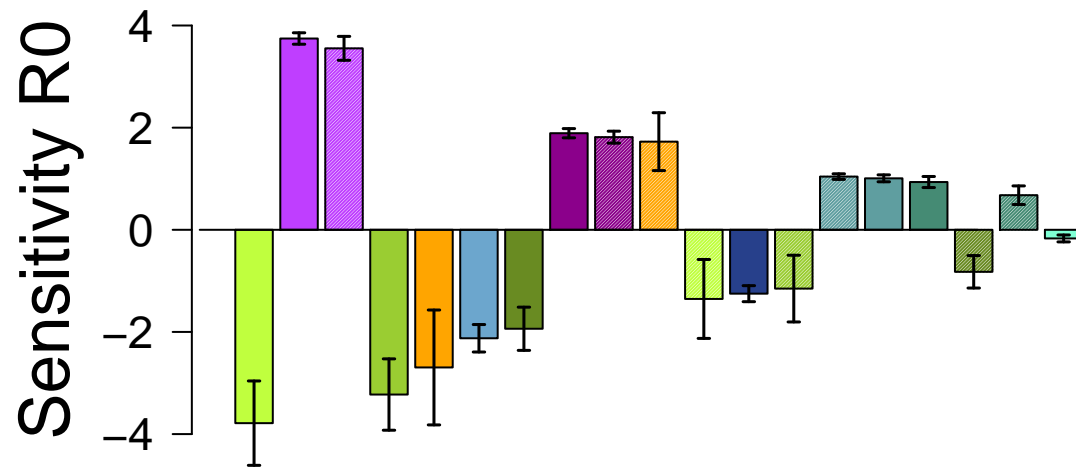
#vec<-c( "trans.beta", "trans.beta", "trans.beta", "trans.beta", "trans.beta", "trans.beta", "trans.beta", "fecundls"
#      "survivalL", "survivalL", "survivalL", "survivalH", "survivalH", "survivalH", "survivalL", "survivalH"
#      "survivalL", "survivalL", "survivalH", "survivalH", "survivalH", "survivalL", "survivalH"
#      "survivalH", "survivalL", "fecundsr")

# names.arg=vec,

bp<-barplot(sensNew1$sens_epidem, las=1, col=color, ylab="Sensitivity R0", cex =0.6,
            xpd=FALSE, main="", ylim=c(-5,5),
            cex.lab = 2.2, cex.axis= 1.4,xlim=c(0,24),density=dashed) # xpd = Should bars be allowed to
lines(c(-0.75,20),c(0,0))

arrows(bp, sensNew1$sens_epidem - sensNew1$sd_epidem , bp,
       sensNew1$sens_epidem + sensNew1$sd_epidem , lwd = 1.5, angle = 90,
       code = 3, length = 0.03)

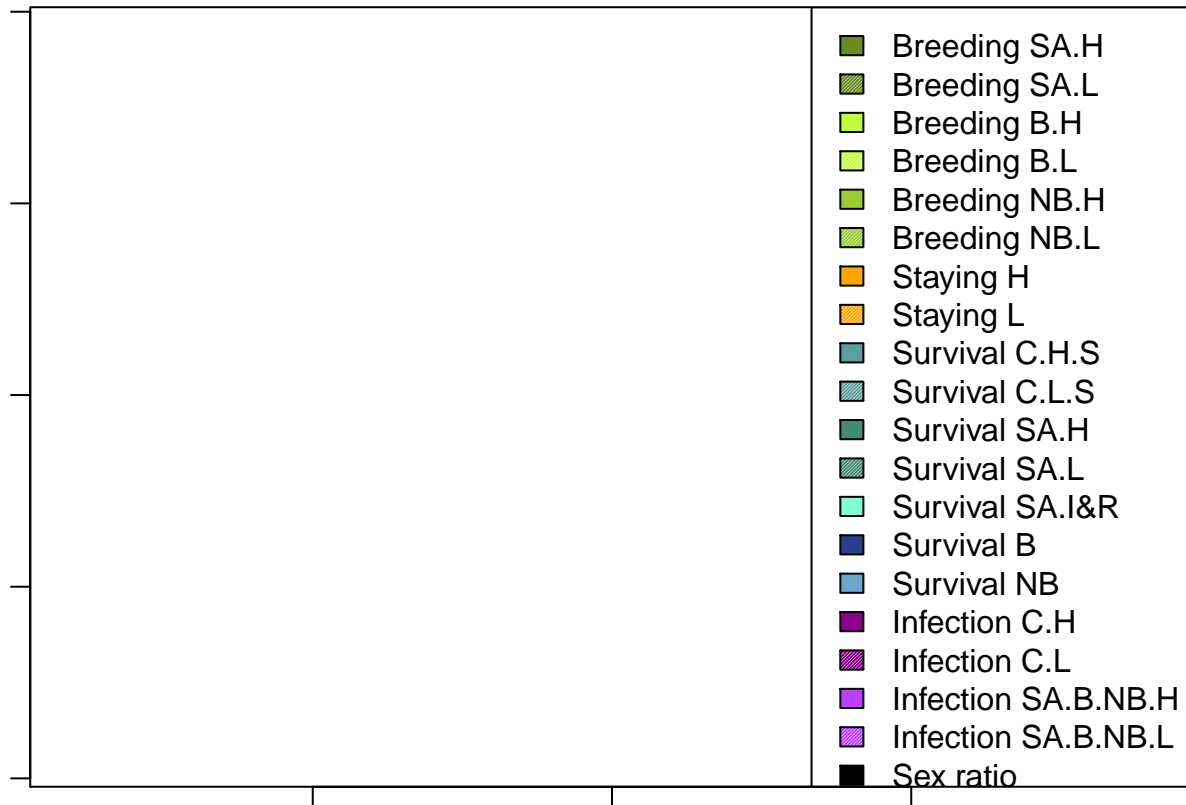
```



Overall legend

```
par(mar=c(1.1,1.1,1.1,1.1))
plot(sensNew1$sens_epidem, pch = "", xlab="", ylab="")

legend("topright",
c("Breeding SA.H", "Breeding SA.L", "Breeding B.H", "Breeding B.L", "Breeding NB.H", "Breeding NB.L",
  "Staying H", "Staying L",
  "Survival C.H.S", "Survival C.L.S", "Survival SA.H", "Survival SA.L", "Survival SA.I&R", "Survival B", "Survival C.H.L",
  "Survival C.L.L", "Survival SA.B.NB.H", "Survival SA.B.NB.L",
  "Infection C.H", "Infection C.L", "Infection SA.B.NB.H", "Infection SA.B.NB.L",
  "Sex ratio")
,
  fill= c("olivedrab4", "olivedrab4", "olivedrab1", "olivedrab1", "olivedrab3", "olivedrab3",
    "orange1", "orange1",
    "cadetblue", "cadetblue", "aquamarine4", "aquamarine4", "aquamarine1", "royalblue4", "skyblue3",
    "darkmagenta", "darkmagenta", "darkorchid1", "darkorchid1",
    "gray0")
,
  density = c(NA, 80, NA, 80, NA, 80,
    NA, 80,
    NA, 80, NA, 80, NA, NA, NA,
    NA, 80, NA, 80,
    NA))
```



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

Time difference of 47.43 mins

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).