

Sensitivity analyses

Esin Ickin

2/23/2023

This code is an example on how to parameterize vital-rate functions using coefficients and covariate inputs to then build a matrix population model and compute sensitivities.

1) Vital rates

The example is from a paper on gray mouse lemurs: Ozgul et al. (in press) In the paper, we parameterized five vital rates as a function of population density, rainfall, and temperature.

The vital rates are defined as functions, and we need the coefficients from vital rate models (here GLMs). We would appreciate a similar input on your part. Alternatively, it possible to have GLM models as an object, and then just use the “predict” function, instead of building out own functions.

```
library(boot)

# Coefficients (can be supplied as a list; written as R objects;
# or directly integrated into the functions below)

# Female adult survival
intercept.sfa=9.95
coef_dens.sfa=0.0401
coef_rain.sfa=0.0031
coef_temp.sfa=-0.38
coef_int.sfa=-0.0000466

# Female juvenile survival
intercept.sfj=9.95-1.86
coef_dens.sfj=0.0401
coef_densjuv.sfj=0.0099
coef_rain.sfj=0.0031
coef_temp.sfj=-0.38
coef_int.sfj=-0.0000466

# Male adult survival
intercept.sma=9.95-1.74
coef_dens.sma=0.0401
coef_rain.sma=0.0031
coef_rainsex.sma=0.0013
coef_temp.sma=-0.38
coef_int.sma=-0.0000466

# Male juvenile survival
intercept.smj=9.95-1.74-1.86
coef_dens.smj=0.0401
```

```

coef_densjuv.smj=0.0099
coef_rain.smj=0.0031
coef_rainsex.smj=0.0013
coef_temp.smj=-0.38
coef_int.smj=-0.0000466

# Recruitment
intercept.rec=-10.965
coef_dens.rec=-0.024
coef_temp.rec=0.364

# Functions

# Female adult survival
Sfa <- function(dens,rain,temp){
  surv <- inv.logit(
    intercept.sfa + coef_dens.sfa*dens + coef_rain.sfa*rain
    + coef_temp.sfa*temp + coef_int.sfa*dens*rain
  )
  return(surv)
}

# Female juvenile survival
Sfj <- function(dens,rain,temp){
  surv <- inv.logit(
    intercept.sfj + coef_dens.sfj*dens + coef_densjuv.sfj*dens
    + coef_rain.sfj*rain + coef_temp.sfj*temp + coef_int.sfj*dens*rain
  )
  return(surv)
}

# Male adult survival
Sma <- function(dens,rain,temp){
  surv <- inv.logit(
    intercept.sma + coef_dens.sma*dens + coef_rain.sma*rain
    + coef_rainsex.sma + coef_temp.sma*temp + coef_int.sma*dens*rain
  )
  return(surv)
}

# Male juvenile survival
Smj <- function(dens,rain,temp){
  surv <- inv.logit(
    intercept.smj + coef_dens.smj*dens + coef_densjuv.smj + coef_rain.smj*rain
    + coef_rainsex.smj + coef_temp.smj*temp + coef_int.smj*dens*rain
  )
  return(surv)
}

# Recruitment
rec <- function(dens,temp){
  Rec <- exp(
    intercept.rec + coef_dens.rec*dens + coef_temp.rec*temp

```

```

)
return(Rec)
}

```

2) Covariates

There are the input data for the vital rate functions. Here, we have a time series of data. This is the best format because it allows us to calculate not only means and variances, but also covariances, the latter being important to calculate scaled sensitivities.

However, it is also possible to send us just the values we are interested in.

```

cov=read.csv("~/Desktop/covariates.csv") # raw data

# Rain
# here and elsewhere,
# this can be calculated from the raw data or a single value can be provided
max.rain=max(cov$rain)
min.rain=min(cov$rain)
mean.rain=mean(cov$rain)
sd.rain=sd(cov$rain)

# Covariation
temp_when_rain_max=cov$tmaxc[which(cov$rain==max(cov$rain))][1]
temp_when_rain_min=cov$tmaxc[which(cov$rain==min(cov$rain))][1]
dens_when_rain_max=cov$pop[which(cov$rain==max(cov$rain))][1]
dens_when_rain_min=cov$pop[which(cov$rain==min(cov$rain))][1]

# Temperature
max.temp=max(cov$tmaxc)
min.temp=min(cov$tmaxc)
mean.temp=mean(cov$tmaxc)
sd.temp=sd(cov$tmaxc)

# Covariation
dens_when_temp_max=cov$pop[which(cov$tmaxc==max(cov$tmaxc))][1]
dens_when_temp_min=cov$pop[which(cov$tmaxc==min(cov$tmaxc))][1]
rain_when_temp_max=cov$rain[which(cov$tmaxc==max(cov$tmaxc))][1]
rain_when_temp_min=cov$rain[which(cov$tmaxc==min(cov$tmaxc))][1]

# Density
max.dens=max(cov$pop)
min.dens=min(cov$pop)
mean.dens= mean(cov$pop)
sd.dens=sd(cov$pop)

# Covariation
rain_when_dens_max=cov$rain[which(cov$pop==max(cov$pop))][1]
rain_when_dens_min=cov$rain[which(cov$pop==min(cov$pop))][1]
temp_when_dens_max=cov$tmaxc[which(cov$pop==max(cov$pop))][1]
temp_when_dens_min=cov$tmaxc[which(cov$pop==min(cov$pop))][1]

```

3) Population model

Here, we use the vital rate function to construct an annual population model that can give us the population growth rate (λ).

In the following example, the MPM is constructed with mean covariate values. In the perturbations, the covariate values are changed.

```
library(popbio)
n.stage=4
mpm = matrix(c(0,Smj(mean.dens,mean.rain,mean.temp),0,0,
                0,Sma(mean.dens,mean.rain,mean.temp),0,0,
                0.5*Sfj(mean.dens,mean.rain,mean.temp)*rec(mean.dens,mean.temp),
                0,0.5*Sfj(mean.dens,mean.rain,mean.temp)*rec(mean.dens,mean.temp),
                Sfj(mean.dens,mean.rain,mean.temp),
                0.5*Sfa(mean.dens,mean.rain,mean.temp)*rec(mean.dens,mean.temp),
                0,0.5*Sfa(mean.dens,mean.rain,mean.temp)*rec(mean.dens,mean.temp),
                Sfa(mean.dens,mean.rain,mean.temp)),n.stage,n.stage)

lambda(mpm)

## [1] 0.70504
```

4) Scaled sensitivity analyses

Here, we calculate scaled sensitivities, according to Morris et al. 2020 (DOI: <https://doi.org/10.1073/pnas.1918363117>)

Note that this is a step that Esin will implement in her MS thesis. With the information given in 1-3, we should be able to run these analyses.

```
library(popbio)

# RAIN
# 1. Sensitivity to rain assuming mean temperature and mean density
mpm.max = matrix(c(0,Smj(mean.dens,max.rain,mean.temp),0,0,
                    0,Sma(mean.dens,max.rain,mean.temp),0,0,
                    0.5*Sfj(mean.dens,max.rain,mean.temp)*rec(mean.dens,mean.temp),
                    0,0.5*Sfj(mean.dens,max.rain,mean.temp)*rec(mean.dens,mean.temp),
                    Sfj(mean.dens,max.rain,mean.temp),
                    0.5*Sfa(mean.dens,max.rain,mean.temp)*rec(mean.dens,mean.temp),
                    0,0.5*Sfa(mean.dens,max.rain,mean.temp)*rec(mean.dens,mean.temp),
                    Sfa(mean.dens,max.rain,mean.temp)),
                    n.stage,n.stage)

mpm.min = matrix(c(0,Smj(mean.dens,min.rain,mean.temp),0,0,
                    0,Sma(mean.dens,min.rain,mean.temp),0,0,
                    0.5*Sfj(mean.dens,min.rain,mean.temp)*rec(mean.dens,mean.temp),
                    0,0.5*Sfj(mean.dens,min.rain,mean.temp)*rec(mean.dens,mean.temp),
                    Sfj(mean.dens,min.rain,mean.temp),
                    0.5*Sfa(mean.dens,min.rain,mean.temp)*rec(mean.dens,mean.temp),
                    0,0.5*Sfa(mean.dens,min.rain,mean.temp)*rec(mean.dens,mean.temp),
                    Sfa(mean.dens,min.rain,mean.temp)),
                    n.stage,n.stage)

delta.R=abs((lambda(mpm.max)-lambda(mpm.min))/((max.rain-min.rain)/sd.rain))
```

```

# 2. Sensitivity to rain assuming covariation among covariates
mpm.max = matrix(c(0,Smj(dens_when_rain_max,max.rain,temp_when_rain_max),0,0,
0,Sma(dens_when_rain_max,max.rain,temp_when_rain_max),0,0,
0.5*Sfj(dens_when_rain_max,max.rain,temp_when_rain_max)*
rec(dens_when_rain_max,temp_when_rain_max),
0,0.5*Sfj(dens_when_rain_max,max.rain,temp_when_rain_max)*
rec(dens_when_rain_max,temp_when_rain_max),
Sfj(dens_when_rain_max,max.rain,temp_when_rain_max),
0.5*Sfa(dens_when_rain_max,max.rain,temp_when_rain_max)*
rec(dens_when_rain_max,temp_when_rain_max),
0,0.5*Sfa(dens_when_rain_max,max.rain,temp_when_rain_max)*
rec(dens_when_rain_max,temp_when_rain_max),
Sfa(dens_when_rain_max,max.rain,temp_when_rain_max)),
n.stage,n.stage)

mpm.min = matrix(c(0,Smj(dens_when_rain_min,min.rain,temp_when_rain_min),0,0,
0,Sma(dens_when_rain_min,min.rain,temp_when_rain_min),0,0,
0.5*Sfj(dens_when_rain_min,min.rain,temp_when_rain_min)*
rec(dens_when_rain_min,temp_when_rain_min),
0,0.5*Sfj(dens_when_rain_min,min.rain,temp_when_rain_min)*
rec(dens_when_rain_min,temp_when_rain_min),
Sfj(dens_when_rain_min,min.rain,temp_when_rain_min),
0.5*Sfa(dens_when_rain_min,min.rain,temp_when_rain_min)*
rec(dens_when_rain_min,temp_when_rain_min),
0,0.5*Sfa(dens_when_rain_min,min.rain,temp_when_rain_min)*
rec(dens_when_rain_min,temp_when_rain_min),
Sfa(dens_when_rain_min,min.rain,temp_when_rain_min)),
n.stage,n.stage)

deltaR.cov=abs((lambda(mpm.max)-lambda(mpm.min)))/((max.rain-min.rain)/sd.rain))

# TEMPERATURE
# 1. Sensitivity to temperature assuming mean rain and mean density
mpm.max = matrix(c(0,Smj(mean.dens,mean.rain,max.temp),0,0,
0,Sma(mean.dens,mean.rain,max.temp),0,0,
0.5*Sfj(mean.dens,mean.rain,max.temp)*rec(mean.dens,max.temp),
0,0.5*Sfj(mean.dens,mean.rain,max.temp)*rec(mean.dens,max.temp),
Sfj(mean.dens,mean.rain,max.temp),
0.5*Sfa(mean.dens,mean.rain,max.temp)*rec(mean.dens,max.temp),
0,0.5*Sfa(mean.dens,mean.rain,max.temp)*rec(mean.dens,max.temp),
Sfa(mean.dens,mean.rain,max.temp)),
n.stage,n.stage)

mpm.min = matrix(c(0,Smj(mean.dens,mean.rain,min.temp),0,0,
0,Sma(mean.dens,mean.rain,min.temp),0,0,
0.5*Sfj(mean.dens,mean.rain,min.temp)*rec(mean.dens,min.temp),
0,0.5*Sfj(mean.dens,mean.rain,min.temp)*rec(mean.dens,min.temp),
Sfj(mean.dens,mean.rain,min.temp),
0.5*Sfa(mean.dens,mean.rain,min.temp)*rec(mean.dens,min.temp),
0,0.5*Sfa(mean.dens,mean.rain,min.temp)*rec(mean.dens,min.temp),
Sfa(mean.dens,mean.rain,min.temp)),
n.stage,n.stage)

```

```

delta.T=abs((lambda(mpm.max)-lambda(mpm.min))/((max.temp-min.temp)/sd.temp))

# 2. Sensitivity to temperature assuming covariation among covariates
mpm.max = matrix(c(0,Smj(dens_when_temp_max,rain_when_temp_max,max.temp),0,0,
                    0,Sma(dens_when_temp_max,rain_when_temp_max,max.temp),0,0,
                    0.5*Sfj(dens_when_temp_max,rain_when_temp_max,max.temp)*
                    rec(dens_when_temp_max,max.temp),0,
                    0.5*Sfj(dens_when_temp_max,rain_when_temp_max,max.temp)*
                    rec(dens_when_temp_max,max.temp),
                    Sfj(dens_when_temp_max,rain_when_temp_max,max.temp),
                    0.5*Sfa(dens_when_temp_max,rain_when_temp_max,max.temp)*
                    rec(dens_when_temp_max,max.temp),0,
                    0.5*Sfa(dens_when_temp_max,rain_when_temp_max,max.temp)*
                    rec(dens_when_temp_max,max.temp),
                    Sfa(dens_when_temp_max,rain_when_temp_max,max.temp)),
                    n.stage,n.stage)

mpm.min = matrix(c(0,Smj(dens_when_temp_min,rain_when_temp_min,min.temp),0,0,
                    0,Sma(dens_when_temp_min,rain_when_temp_min,min.temp),0,0,
                    0.5*Sfj(dens_when_temp_min,rain_when_temp_min,min.temp)*
                    rec(dens_when_temp_min,min.temp),0,
                    0.5*Sfj(dens_when_temp_min,rain_when_temp_min,min.temp)*
                    rec(dens_when_temp_min,min.temp),
                    Sfj(dens_when_temp_min,rain_when_temp_min,min.temp),
                    0.5*Sfa(dens_when_temp_min,rain_when_temp_min,min.temp)*
                    rec(dens_when_temp_min,min.temp),0,
                    0.5*Sfa(dens_when_temp_min,rain_when_temp_min,min.temp)*
                    rec(dens_when_temp_min,min.temp),
                    Sfa(dens_when_temp_min,rain_when_temp_min,min.temp)),
                    n.stage,n.stage)

deltaT.cov=abs((lambda(mpm.max)-lambda(mpm.min))/((max.temp-min.temp)/sd.temp))

# DENSITY
# 1. Sensitivity to density assuming mean rain and mean temperature
mpm.max = matrix(c(0,Smj(max.dens,mean.rain,mean.temp),0,0,
                    0,Sma(max.dens,mean.rain,mean.temp),0,0,
                    0.5*Sfj(max.dens,mean.rain,mean.temp)*rec(max.dens,mean.temp),
                    0,0.5*Sfj(max.dens,mean.rain,mean.temp)*rec(max.dens,mean.temp),
                    Sfj(max.dens,mean.rain,mean.temp),
                    0.5*Sfa(max.dens,mean.rain,mean.temp)*rec(max.dens,mean.temp),0,
                    0.5*Sfa(max.dens,mean.rain,mean.temp)*rec(max.dens,mean.temp),
                    Sfa(max.dens,mean.rain,mean.temp)),
                    n.stage,n.stage)

mpm.min = matrix(c(0,Smj(min.dens,mean.rain,mean.temp),0,0,
                    0,Sma(min.dens,mean.rain,mean.temp),0,0,
                    0.5*Sfj(min.dens,mean.rain,mean.temp)*rec(min.dens,mean.temp),
                    0,0.5*Sfj(min.dens,mean.rain,mean.temp)*rec(min.dens,mean.temp),
                    Sfj(min.dens,mean.rain,mean.temp),
                    0.5*Sfa(min.dens,mean.rain,mean.temp)*rec(min.dens,mean.temp),
                    0,0.5*Sfa(min.dens,mean.rain,mean.temp)*rec(min.dens,mean.temp),
                    Sfa(min.dens,mean.rain,mean.temp)),
                    n.stage,n.stage)

```

```

n.stage,n.stage)

delta.D=abs(lambda(mpm.max)-lambda(mpm.min))/((max.dens-min.dens)/sd.dens)

# 2. Sensitivity to density assuming covariation among covariates
mpm.max = matrix(c(0,Smj(max.dens,rain_when_dens_max,temp_when_dens_max),0,0,
0,Sma(max.dens,rain_when_dens_max,temp_when_dens_max),0,0,
0.5*Sfj(max.dens,rain_when_dens_max,temp_when_dens_max)*
rec(max.dens,temp_when_dens_max),0,
0.5*Sfj(max.dens,rain_when_dens_max,temp_when_dens_max)*
rec(max.dens,temp_when_dens_max),
Sfj(max.dens,rain_when_dens_max,temp_when_dens_max),
0.5*Sfa(max.dens,rain_when_dens_max,temp_when_dens_max)*
rec(max.dens,temp_when_dens_max),0,
0.5*Sfa(max.dens,rain_when_dens_max,temp_when_dens_max)*
rec(max.dens,temp_when_dens_max),
Sfa(max.dens,rain_when_dens_max,temp_when_dens_max)),
n.stage,n.stage)

mpm.min = matrix(c(0,Smj(min.dens,rain_when_dens_min,temp_when_dens_min),0,0,
0,Sma(min.dens,rain_when_dens_min,temp_when_dens_min),0,0,
0.5*Sfj(min.dens,rain_when_dens_min,temp_when_dens_min)*
rec(min.dens,temp_when_dens_min),0,
0.5*Sfj(min.dens,rain_when_dens_min,temp_when_dens_min)*
rec(min.dens,temp_when_dens_min),
Sfj(min.dens,rain_when_dens_min,temp_when_dens_min),
0.5*Sfa(min.dens,rain_when_dens_min,temp_when_dens_min)*
rec(min.dens,temp_when_dens_min),0,
0.5*Sfa(min.dens,rain_when_dens_min,temp_when_dens_min)*
rec(min.dens,temp_when_dens_min),
Sfa(min.dens,rain_when_dens_min,temp_when_dens_min)),
n.stage,n.stage)

deltaD.cov=abs(lambda(mpm.max)-lambda(mpm.min))/((max.dens-min.dens)/sd.dens)

```

Do sensitivities decrease when we consider covariation among covariates? This example suggests that yes

```
## [1] "Scaled sensitivity no covariation: 0.05"
```

```
## [1] "Scaled sensitivity with covariation: 0.01"
```

5) Sensitivity analyses at equilibrium dynamics

Here, we perform “classic” sensitivity analyses (see Paniw et al. 2019; DOI: 10.1126/science.aau5905)

1. We look for combinations of covariate values where lambda approaches 1

```

# go through different combinations of covariates
cov=expand.grid(rain=seq(min.rain,max.rain,length.out=20),
temp=seq(min.temp,max.temp,length.out=20),
pop=seq(min.dens,max.dens,length.out=20))

# empty object for lambdas
all.l=rep(NA,nrow(cov))

# define number of stages

```

```

n.stage=4

for(s in 1:nrow(cov)){
  mpm = matrix(c(0,Smj(cov$pop[s],cov$rain[s],cov$temp[s]),0,0,
                    0,Sma(cov$pop[s],cov$rain[s],cov$temp[s]),0,0,
                    0.5*Sfj(cov$pop[s],cov$rain[s],cov$temp[s])*rec(cov$pop[s],cov$temp[s]),
                    0,0.5*Sfj(cov$pop[s],cov$rain[s],cov$temp[s])*rec(cov$pop[s],cov$temp[s]),
                    Sfj(cov$pop[s],cov$rain[s],cov$temp[s]),
                    0.5*Sfa(cov$pop[s],cov$rain[s],cov$temp[s])*rec(cov$pop[s],cov$temp[s]),
                    0,0.5*Sfa(cov$pop[s],cov$rain[s],cov$temp[s])*rec(cov$pop[s],cov$temp[s]),
                    Sfa(cov$pop[s],cov$rain[s],cov$temp[s])),
                n.stage,n.stage)

  all.l[s]=lambda(mpm)
}
stable.l=which(all.l>0.99&all.l<1.01)

```

2. We perturb each covariate for each vital rate in turn, while maintaining the other covariates at values that would ensure stable population dynamics. A perturbation entails a 10 % increase in a covariate.

```

# Perturb rainfall
dR.lemur=NULL

for(s in 1:length(stable.l)){
  rain=cov$rain[stable.l[s]]
  rain.pert=cov$rain[stable.l[s]]+0.1*abs(cov$rain[stable.l[s]]) # increase rain by 10 %
  pop=cov$pop[stable.l[s]]
  temp=cov$temp[stable.l[s]]

  control=matrix(c(0,Smj(pop,rain,temp),0,0,
                      0,Sma(pop,rain,temp),0,0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.smj=matrix(c(0,Smj(pop,rain.pert,temp),0,0,
                      0,Sma(pop,rain,temp),0,0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.sma=matrix(c(0,Smj(pop,rain,temp),0,0,
                      0,Sma(pop,rain.pert,temp),0,0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                      0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.sfj=matrix(c(0,Smj(pop,rain,temp),0,0,

```



```

0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain.pert,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain.pert,temp)*rec(pop,temp),Sfj(pop,rain.pert,temp),
0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
n.stage,n.stage)

pert.sfa=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
0.5*Sfa(pop,rain.pert,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain.pert,temp)*rec(pop,temp),Sfa(pop,rain.pert,temp)),
n.stage,n.stage)

pert.all=matrix(c(0,Smj(pop,rain.pert,temp),0,0,
0,Sma(pop,rain.pert,temp),0,0,
0.5*Sfj(pop,rain.pert,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain.pert,temp)*rec(pop,temp),Sfj(pop,rain.pert,temp),
0.5*Sfa(pop,rain.pert,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain.pert,temp)*rec(pop,temp),Sfa(pop,rain.pert,temp)),
n.stage,n.stage)

temp.mpm = data.frame(vr=c("Smj","Sma","Sfj","Sfa","Rec","All"),
delta=c((lambda(pert.smj)-lambda(control))/lambda(control),
(lambda(pert.sma)-lambda(control))/lambda(control),
(lambda(pert.sfj)-lambda(control))/lambda(control),
(lambda(pert.sfa)-lambda(control))/lambda(control),
NA,
(lambda(pert.all)-lambda(control))/lambda(control)))

dR.lemur=rbind(dR.lemur,temp.mpm)
}

dR.lemur$variable="Rain"

# Perturb temperature
dT.lemur=NULL

for(s in 1:length(stable.l)){

temp=cov$temp[stable.l[s]]
temp.pert=cov$temp[stable.l[s]]+0.1*abs(cov$temp[stable.l[s]]) # increase temp by 10 %
pop=cov$pop[stable.l[s]]
rain=cov$rain[stable.l[s]]

control=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
n.stage,n.stage)

```

```

pert.smj=matrix(c(0,Smj(pop,rain,temp.pert),0,0,
                 0,Sma(pop,rain,temp),0,0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
               n.stage,n.stage)

pert.sma=matrix(c(0,Smj(pop,rain,temp),0,0,
                 0,Sma(pop,rain,temp.pert),0,0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
               n.stage,n.stage)

pert.sfj=matrix(c(0,Smj(pop,rain,temp),0,0,
                 0,Sma(pop,rain,temp),0,0,
                 0.5*Sfj(pop,rain,temp.pert)*rec(pop,temp),0,
                 0.5*Sfj(pop,rain,temp.pert)*rec(pop,temp),Sfj(pop,rain,temp.pert),
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
               n.stage,n.stage)

pert.sfa=matrix(c(0,Smj(pop,rain,temp),0,0,
                 0,Sma(pop,rain,temp),0,0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                 0.5*Sfa(pop,rain,temp.pert)*rec(pop,temp),0,
                 0.5*Sfa(pop,rain,temp.pert)*rec(pop,temp),Sfa(pop,rain,temp.pert)),
               n.stage,n.stage)

pert.rec=matrix(c(0,Smj(pop,rain,temp),0,0,
                 0,Sma(pop,rain,temp),0,0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp.pert),0,
                 0.5*Sfj(pop,rain,temp)*rec(pop,temp.pert),Sfj(pop,rain,temp),
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp.pert),0,
                 0.5*Sfa(pop,rain,temp)*rec(pop,temp.pert),Sfa(pop,rain,temp)),
               n.stage,n.stage)

pert.all=matrix(c(0,Smj(pop,rain,temp.pert),0,0,
                 0,Sma(pop,rain,temp.pert),0,0,
                 0.5*Sfj(pop,rain,temp.pert)*rec(pop,temp.pert),0,
                 0.5*Sfj(pop,rain,temp.pert)*rec(pop,temp.pert),Sfj(pop,rain,temp.pert),
                 0.5*Sfa(pop,rain,temp.pert)*rec(pop,temp.pert),0,
                 0.5*Sfa(pop,rain,temp.pert)*rec(pop,temp.pert),Sfa(pop,rain,temp.pert)),
               n.stage,n.stage)

temp.mpm = data.frame(vr=c("Smj","Sma","Sfj","Sfa","Rec","All"),
                      delta=c((lambda(pert.smj)-lambda(control))/lambda(control),
                              (lambda(pert.sma)-lambda(control))/lambda(control),
                              (lambda(pert.sfj)-lambda(control))/lambda(control),
                              (lambda(pert.sfa)-lambda(control))/lambda(control),

```

```

        (lambda(pert.rec)-lambda(control))/lambda(control),
        (lambda(pert.all)-lambda(control))/lambda(control))

dT.lemur=rbind(dT.lemur,temp.mpm)
}

dT.lemur$variable="Temperature"

# Perturb density
dD.lemur=NULL

for(s in 1:length(stable.l)){

  pop=cov$pop[stable.l[s]]
  pop.pert=cov$pop[stable.l[s]]+0.1*abs(cov$pop[stable.l[s]]) # increase rain by 10 %
  temp=cov$temp[stable.l[s]]
  rain=cov$rain[stable.l[s]]

  control=matrix(c(0,Smj(pop,rain,temp),0,0,
                    0,Sma(pop,rain,temp),0,0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.smj=matrix(c(0,Smj(pop.pert,rain,temp),0,0,
                    0,Sma(pop,rain,temp),0,0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.sma=matrix(c(0,Smj(pop,rain,temp),0,0,
                    0,Sma(pop.pert,rain,temp),0,0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.sfj=matrix(c(0,Smj(pop,rain,temp),0,0,
                    0,Sma(pop,rain,temp),0,0,
                    0.5*Sfj(pop.pert,rain,temp)*rec(pop,temp),0,
                    0.5*Sfj(pop.pert,rain,temp)*rec(pop,temp),Sfj(pop.pert,rain,temp),
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                n.stage,n.stage)

  pert.sfa=matrix(c(0,Smj(pop,rain,temp),0,0,
                    0,Sma(pop,rain,temp),0,0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,

```

```

        0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
        0.5*Sfa(pop.pert,rain,temp)*rec(pop,temp),0,
        0.5*Sfa(pop.pert,rain,temp)*rec(pop,temp),Sfa(pop.pert,rain,temp)),
        n.stage,n.stage)

pert.rec=matrix(c(0,Smj(pop,rain,temp),0,0,
                  0,Sma(pop,rain,temp),0,0,
                  0.5*Sfj(pop,rain,temp)*rec(pop.pert,temp),0,
                  0.5*Sfj(pop,rain,temp)*rec(pop.pert,temp),Sfj(pop,rain,temp),
                  0.5*Sfa(pop,rain,temp)*rec(pop.pert,temp),0,
                  0.5*Sfa(pop,rain,temp)*rec(pop.pert,temp),Sfa(pop,rain,temp)),
                  n.stage,n.stage)

pert.all=matrix(c(0,Smj(pop.pert,rain,temp),0,0,
                  0,Sma(pop.pert,rain,temp),0,0,
                  0.5*Sfj(pop.pert,rain,temp)*rec(pop.pert,temp),0,
                  0.5*Sfj(pop.pert,rain,temp)*rec(pop.pert,temp),Sfj(pop.pert,rain,temp),
                  0.5*Sfa(pop.pert,rain,temp)*rec(pop.pert,temp),0,
                  0.5*Sfa(pop.pert,rain,temp)*rec(pop.pert,temp),Sfa(pop.pert,rain,temp)),
                  n.stage,n.stage)

temp.mpm = data.frame(vr=c("Smj","Sma","Sfj","Sfa","Rec","All"),
                      delta=c((lambda(pert.smj)-lambda(control))/lambda(control),
                              (lambda(pert.sma)-lambda(control))/lambda(control),
                              (lambda(pert.sfj)-lambda(control))/lambda(control),
                              (lambda(pert.sfa)-lambda(control))/lambda(control),
                              (lambda(pert.rec)-lambda(control))/lambda(control),
                              (lambda(pert.all)-lambda(control))/lambda(control)))

dD.lemur=rbind(dD.lemur,temp.mpm)
}

dD.lemur$variable="Density"

# Perturb all covariates
dA.lemur=NULL

for(s in 1:length(stable.1)){

  pop=cov$pop[stable.1[s]]
  pop.pert=cov$pop[stable.1[s]]+0.1*abs(cov$pop[stable.1[s]]) # increase by 10 %
  temp=cov$temp[stable.1[s]]
  temp.pert=cov$temp[stable.1[s]]+0.1*abs(cov$temp[stable.1[s]]) # increase by 10 %
  rain=cov$rain[stable.1[s]]
  rain.pert=cov$rain[stable.1[s]]+0.1*abs(cov$rain[stable.1[s]]) # increase by 10 %

  control=matrix(c(0,Smj(pop,rain,temp),0,0,
                    0,Sma(pop,rain,temp),0,0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
                    0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
                    n.stage,n.stage)

```

```

pert.smj=matrix(c(0,Smj(pop.pert,rain.pert,temp.pert),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
n.stage,n.stage)

pert.sma=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop.pert,rain.pert,temp.pert),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
n.stage,n.stage)

pert.sfj=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop.pert,rain.pert,temp.pert)*rec(pop,temp),0,
0.5*Sfj(pop.pert,rain.pert,temp.pert)*rec(pop,temp),
Sfj(pop.pert,rain.pert,temp.pert),
0.5*Sfa(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfa(pop,rain,temp)*rec(pop,temp),Sfa(pop,rain,temp)),
n.stage,n.stage)

pert.sfa=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),0,
0.5*Sfj(pop,rain,temp)*rec(pop,temp),Sfj(pop,rain,temp),
0.5*Sfa(pop.pert,rain.pert,temp.pert)*rec(pop,temp),0,
0.5*Sfa(pop.pert,rain.pert,temp.pert)*rec(pop,temp),
Sfa(pop.pert,rain.pert,temp.pert)),
n.stage,n.stage)

pert.rec=matrix(c(0,Smj(pop,rain,temp),0,0,
0,Sma(pop,rain,temp),0,0,
0.5*Sfj(pop,rain,temp)*rec(pop.pert,temp.pert),0,
0.5*Sfj(pop,rain,temp)*rec(pop.pert,temp.pert),Sfj(pop,rain,temp),
0.5*Sfa(pop,rain,temp)*rec(pop.pert,temp.pert),0,
0.5*Sfa(pop,rain,temp)*rec(pop.pert,temp.pert),Sfa(pop,rain,temp)),
n.stage,n.stage)

pert.all=matrix(c(0,Smj(pop.pert,rain.pert,temp.pert),0,0,
0,Sma(pop.pert,rain.pert,temp.pert),0,0,
0.5*Sfj(pop.pert,rain.pert,temp.pert)*rec(pop.pert,temp.pert),0,
0.5*Sfj(pop.pert,rain.pert,temp.pert)*rec(pop.pert,temp.pert),
Sfj(pop.pert,rain.pert,temp.pert),
0.5*Sfa(pop.pert,rain.pert,temp.pert)*rec(pop.pert,temp.pert),0,
0.5*Sfa(pop.pert,rain.pert,temp.pert)*rec(pop.pert,temp.pert),
Sfa(pop.pert,rain.pert,temp.pert)),
n.stage,n.stage)

temp.mpm = data.frame(vr=c("Smj", "Sma", "Sfj", "Sfa", "Rec", "All"),

```

```

        delta=c((lambda(pert.smj)-lambda(control))/lambda(control),
                (lambda(pert.sma)-lambda(control))/lambda(control),
                (lambda(pert.sfj)-lambda(control))/lambda(control),
                (lambda(pert.sfa)-lambda(control))/lambda(control),
                (lambda(pert.rec)-lambda(control))/lambda(control),
                (lambda(pert.all)-lambda(control))/lambda(control))

dA.lemur=rbind(dA.lemur,temp.mpm)
}

dA.lemur$variable="All cov"

# put all results into a dataframe
df=rbind(dR.lemur,dT.lemur,dD.lemur,dA.lemur)
df$vr=factor(df$vr,levels=c("Smj","Sma","Sfj","Sfa","Rec","All"))
df$delta[is.na(df$delta)]=0

```

3. Plot the sensitivities

Note that when perturbing all covariates in all vital rate models, the resulting changes in lambda are much less pronounced then when looking at single vital rate models.

```

library(viridis)

## Loading required package: viridisLite

library(ggplot2)

ggplot(df,aes(vr,delta,col=variable))+
  geom_boxplot(outlier.shape = NA)+
  scale_color_viridis(discrete = TRUE, alpha=1,end=0.9) +
  geom_hline(yintercept=0,
            color = "black", linewidth=0.5)+
  ylab(expression(paste("% change ", lambda)))+xlab("Demographic rate")+
  theme_bw(base_size=20)+
  theme(panel.grid = element_blank()+
        labs(fill='Perturbed',colour='Perturbed')+
        theme(panel.grid.major = element_blank(),
              panel.grid.minor = element_blank(),
              strip.background = element_blank(),
              strip.text = element_text(face = "italic"),
              legend.position = c(0.20,0.70), # might have to change legend position
              legend.background = element_rect(color=NA),
              panel.border = element_rect(colour = "black"))

```

