

```

##### R Code for 'Sensitivity and elasticity matrices' #####

# https://compadre-db.org/Education/article/sensitivity-and-elasticity-matrices

# Compiled on 9 February 2021

##### Preliminaries #####

# Use the command following commands if you haven't already
# downloaded the packages:

# install.packages("popbio")

library(popbio)

##### Calculate sensitivities and elasticities #####

# Create matrix
giraffe <- matrix(c(0, 0, 0.24, 0.57, 0, 0, 0, 0.79, 0.84), nrow=3, byrow=TRUE,
                  dimnames=list(c("calf", "subadult", "adult"),
                                c("calf", "subadult", "adult")))

mat <- giraffe

mat

# Calculate left and right eigenvectors
w <- eigen(mat)$vectors
v <- Conj(solve(w))

# Calculate sensitivity matrix
senmat <- Re(v[1,] %*% t(w[,1]))
senmat

# Calculate elasticity matrix
emat <- (1/(Re(eigen(mat)$values[1]))) * senmat * mat
emat

# Use functions in the popbio package
sensitivity(mat, zero = FALSE)

elasticity(mat)

##### Compare elasticities for different transition types #####

# Sum fecundity, growth, and stasis transitions
sumfec <- sum(emat[1,2:3])

```

```

sumgrow <- sum(emat[2,1],emat[3,1],emat[3,2])
sumstasis <- sum(diag(emat),emat[2,3])

# Plot elasticities
par(mar=c(2, 2, 2, 2))
pie(c(sumgrow,sumfec,sumstasis), col = c("gray50", "#6AA341", "gray90"),
    labels=c("growth", "fecundity", "stasis"))

##### Plot changes in lambda #####

# Create values for transition rates
vals <- c(1.01, 1.05, 1.1)

# Create list of results for each transition probability
results <- list()
for (x in 1:length(vals)) {
  testlam <- matrix(0, nrow=nrow(mat), ncol=nrow(mat))

  for (i in c(1:nrow(mat))) {
    for (j in c(1:nrow(mat))) {
      if (mat[i,j] == 0) {testlam[i,j] <- 0} else {
        tempmat <- mat
        tempmat[i,j] <- mat[i,j]*vals[x]
        testlam[i,j] <- Re(eigen(tempmat)$values[1])
      }
    }
  }
  results[[x]] <- testlam
}

# Plot lambdas as function of the changes to transition rates
output <- matrix(NA, nrow=length(vals), ncol=length(which(as.vector(t(results[[1]])) >0)))
for (w in c(1: length(vals))) {
  output[w,] <- as.vector(t(results[[w]]))[which(as.vector(t(results[[w]])) >0)]
}

lower <- 0.8

par(mar=c(5, 4, 2, 2))
barplot(output-lower, beside=TRUE, ylim=c(lower, 1.12), ylab="Population growth rate",
    col=c("#007765", "#6AA341", "gray90"), offset= lower)
abline(h=lower, lwd=2)
abline(h=1.0, lwd=2, col="blue")
abline(h= (Re(eigen(mat)$values[1])), lwd=2, col="orange")
legend("topleft", inset=c(0.01,0.01), c("+1%", "+5%", "+10%"),
    fill=c("#007765", "#6AA341", "gray90"), cex=0.8)
text(x=c(3.7,7.7,11.65,15.8), y=0.78,
    labels=c("AdultFec", "CalfSurv", "SubSurv", "AdultSurv"),

```

pos=2, xpd=TRUE, cex=0.9)