

```

##### R Code for 'Additional Rcompadre exercises' #####

# https://compadre-db.org/Education/article/additional-rcompadre-exercises

# Compiled on 9 February 2021

##### Preliminaries #####

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

# install.packages("popbio")
# install.packages("maps")
# install.packages("DiagrammeR")

# Instructions for installing Rcompadre can be found at the link below
# https://jonesor.github.io/Rcompadre/index.html

library(Rcompadre)
library(popbio)
library(maps)
library(DiagrammeR)

##### Simple outputs from many matrices #####

# Download COMADRE
comadre <- cdb_fetch("comadre")

# Alternative download location at link below
# https://compadre-db.org/Data/Comadre

# Convert object (not needed if using 'cdb_fetch')
comadre <- as_cdb(comadre) # Converts the S3 database format from the website into S4

# Subset the database
x <- subset(comadre, MatrixComposite == "Mean" & # Subsets the database based on variables in the
metadata
      Class == "Actinopterygii" &
      StudyDuration >= 3 &
      MatrixDimension > 3)

# Create data frame
output <- data.frame(lambdas = rep(NA, length(x$mat)), damps = rep(NA, length(x$mat)))
for (i in 1:length(x$mat)){
  output$lambdas[i] <- Re(eigen(x$mat[[i]]@matA)$value)[1]
  output$damps[i] <- damping.ratio(x$mat[[i]]@matA)
}

```

```

# Examine output
output

# Add species name
data.frame(Species = x@data$SpeciesAccepted, output)

# Plot population growth rates and damping ratios
par(mfrow = c(1,2))
hist(log(output$lambdas),xlab = "Log population growth rate", col = "gold", main = "")
abline(v=0,col = "red", lwd = 4, lty = 3)
hist(output$damps, xlab = "Damping ratio", col = "brown", main = "")

##### Plot a life cycle #####

# For this exercise, you'll need to download the database
# as an S3 object from the website (code doesn't work if
# using 'cdb_fetch' to download)
# https://compadre-db.org/Data/Comadre

# Find species by common name
lions <- grep("lion", comadre$metadata$CommonName)
unique(comadre$metadata$CommonName[lions])

# Pull details for plotting red lionfish life cycle
matNum <- which(comadre$metadata$CommonName == "Red lionfish")
matNum

sp <- gsub("_", " ", comadre$metadata$SpeciesAccepted[matNum])
sp # Latin name

matA <- comadre$mat[[matNum]]$matA
matA # The 'A' matrix

stages <- comadre$matrixClass[[matNum]]$MatrixClassAuthor
stages # The life stages

# Define function to plot life cycle
plotLifeCycle <- function(A, title = "my life cycle", shape = "egg",
                          fontsize = 10, nodefontsize = 12, edgecol = "grey") {

#Identify the stages
rownames(A)<-colnames(A)

#Remove the unnecessary "A"
Astages<-gsub("A","",colnames(A))

#Construct a "from"->"to" dataset (edges)
fromTo<-expand.grid(Astages,Astages)

```

```

names(fromTo) <- c("From","To")

#Loop through the edges to get the quantities
#(transition probabilities and fecundity)
for(i in 1:nrow(fromTo)){
  fromTo$quantity[i] <- A[fromTo$To[i],fromTo$From[i]]
}

#Subset to only include those where the quantity >0
temp<- subset(fromTo,fromTo$quantity > 0)

#Create sorted vector of node names
allNodes <- sort(unique(c(as.character(temp[,1]),as.character(temp[,2]))))

#Add a semi-colon, for use by graphviz
allNodes <- paste(allNodes,collapse="; ")

#Manipulate minimim length of edge to make the plot pretty.
#Experimental!!
temp$minLVal <-as.numeric(temp[,2])-as.numeric(temp[,1])
temp$minLVal <- temp$minLVal*3

#Create the edges argument for graphviz
#by pasting commands together.
#Note, one could modify this to alter the outputs away from my defaults.
allEdges <- paste(temp[,1],"-
>",temp[,2],"[minlen=",temp[, "minLVal"],"fontsize=",temp[,2],"color=",temp[,2],"xlabel=",
paste("\",round(temp[,3],3)),"\n",collapse="")

#The graphviz argument, pasted together
DiagrammeR::grViz(paste(
  "digraph {
  {
  graph[overlap=false];
  rank=same;
node [shape=",shape,", fontsize=",nodefontsize,"];
",
  allNodes
  ,
  }
ordering=out
x [style=invis]
x -> {"",allNodes,"} [style=invis]",
  allEdges,
  "
# title
  labelloc="\t\");

```

```
    label="",title,""  
  }  
  )  
}
```

```
# Plot life cycle  
plotLifeCycle(matA, title = "Red lionfish")
```

```
##### Geographic distribution #####
```

```
comadre <- as_cdb(comadre) # Converts the S3 database format from the website into S4
```

```
# Subset COMADRE  
x <- subset(comadre, MatrixComposite == "Mean" &  
  Order == "Carnivora" &  
  MatrixCaptive == "W" &  
  Lat > 0 &  
  SurvivalIssue < 1 &  
  MatrixSplit == "Divided" &  
  MatrixFec == "Yes")
```

```
# Create empty variable  
x@data$lambdas <- NA
```

```
# Create 'for' loop  
for (i in 1:length(x$mat)){  
  tryCatch({  
    x@data$lambdas[i] <- Re(eigen(x$mat[[i]]@matA)$value)[1]  
  }, error = function(e){})  
}
```

```
# Create color vector  
rampfunc <- colorRampPalette(c("green", "red"))  
colVect <- rampfunc(100)  
colVect <- paste(colVect, "90", sep = "")  
s1 <- seq(min(x@data$lambdas, na.rm=TRUE),max(x@data$lambdas, na.rm=TRUE),  
  length.out = 100)
```

```
# Plot map  
par(mfrow = c(1,1))  
{map("world", col = "gray", fill = TRUE, bg = "light blue",  
  xlim = c(-175, 176), ylim = c(-60, 85), border = "white")  
  points(jitter(x@data$Lon, 0.6), jitter(x@data$Lat, 0.6),  
    col = colVect[findInterval(x@data$lambdas, s1)], cex = 2, pch = 16)  
}
```