

Fisheries Game Scientific Background

Hi Seth, Steve,

I went ahead and coded a model for the game. I'll walk through it part by part...

```
library(tidyverse)
library(plotly)
```

Comparing Shannon and Simpson Diversity

First, I wanted to decide between Shannon's and Simpson's Diversity Indices. I thought the most important characteristics of these maps were their shape over number of populations (changes as players overfish and drive species to extinction), and coefficient of variance (var/mean population size; something we can affect with the game model). I know we had been discussing weighting things by hand to emphasize certain species. On second thought, I'd advise against doing that and instead build the sensitivity of some species into the model inherently. So let's just look at simple Shannon's (H) and inverse Simpson's ($1/D$) indices.

```
set.seed(42)
```

```
shannon <- function(x) -sum(x/sum(x)*log(max(0.0000001,(x/sum(x))))) # I
included the small value because computers can't handle 0*log(0)
```

```
#http://www.tiem.utk.edu/~gross/bioed/bealsmodules/shannonDI.html
```

```
simpson <- function(x) 1/sum((x/sum(x))^2) #used inverted simpson (1/D) to
make it slope with Shannon
```

```
#http://www.countrysideinfo.co.uk/simpsons.htm
```

```
CV <- seq(0,100)/100 #coefficient of variation (var./mean), mean pop always
equals 1 in this case
```

```
n_pop <- seq(2,10) #number of species (populations)
```

```
shannon_vals <- matrix(0, length(CV), length(n_pop))
```

```
simpson_vals <- matrix(0, length(CV), length(n_pop))
```

```
shantemp <- vector()
```

```
simptemp <- vector()
```

```

for (j in 1:length(n_pop)) {
  for (k in 1:length(CV)) {
    for (i in 1:200) {
      pops <- rnorm(n_pop[j], mean = 1, sd = sqrt(CV[k]))
      pops <- ifelse(pops>0, pops, .0000001) #Log(0) -> NaN
      shantemp[i] <- shannon(pops)
      simptemp[i] <- simpson(pops)
    }
    shannon_vals[k,j] <- mean(shantemp)
    simpson_vals[k,j] <- mean(simptemp)
  }
}

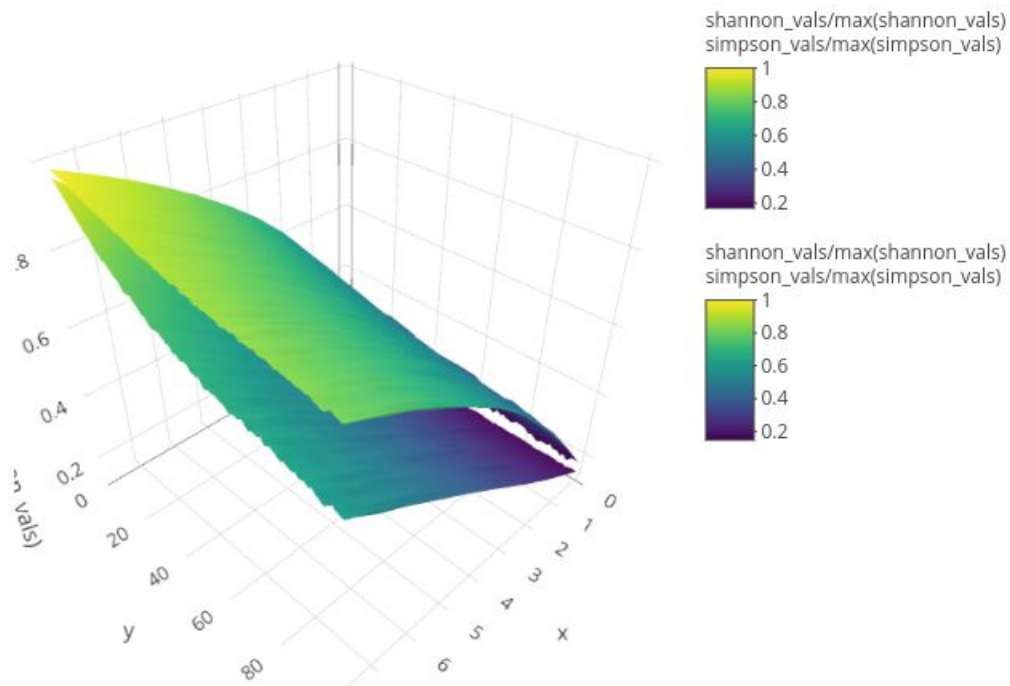
rownames(shannon_vals) <- CV
colnames(shannon_vals) <- n_pop

plot_ly() %>%
  add_surface(z = ~shannon_vals/max(shannon_vals)) %>%
  add_surface(z = ~simpson_vals/max(simpson_vals)) %>%
  api_create(filename="shannon-simpson")

```

Because this runs a for loop (time-intensive) and because there are some issues with putting plots from this package in HTML, I didn't evaluate the code. Here's the result:

<https://plot.ly/~jwing/3/#/plot> if you want to play with it.



Shannon vs Simpson's diversity indices

Note that the top value is Shannon's H , the bottom is Simpson's $1/D$. Also, these are normalized, meaning they have the same maximum value. x and y are not scaled correctly ($y = 100 \cdot CV$, $x = n_{\text{pops}} - 2$).

In light of these results, I'd say it doesn't matter too much which you choose. e^H is actually relatively close to $1/D$.

Let's just go with Simpson's $1/D$ since we've kind of talked this to death (largely of my own volition) and it checks our two boxes: interpretability (equal to pop. size if pops are perfectly even, i.e., $Var = 0$; scales linearly with population size), and prevalence in actual research. It saves a step in that it can handle zeros too ($\log(0) \rightarrow -\text{Inf}$, so Shannon's hangs).

picking species

5 fish species (plus copepods, seagulls, seals, etc.) sounds like a good number. Since you said name recognition was a plus, I'd recommend:

1. Skipjack tuna (top spator)
2. Sardines (eaten by tuna, eat copepods)
3. Haddock (eaten by tuna, eat copepods as juveniles and herring as adults) NOTE: no age in current model.

4. Herring (eaten by tuna and haddock, eat copepods)
5. Summer flounder (eaten by tuna, eats bivalves, e.g., clams, mussels, oysters)

The biggest change is the addition of a non-fish primary consumer level (bivalves). This will make the ecological dynamics a bit more interesting and variable from game to game

I think these are fish lots of people will recognize, with distinctive shapes (good for logos), and with food web roles that make for a model highlighting the ecology of the system.

model for game

Rather than trying to make this model as ecologically realistic as possible, I'm prioritizing game design first, while still trying to cause the ecological concepts we discussed to arise in normal play.

```
set.seed(100)

timesteps <- 100 # this is based off of our conference call (8 turns)

fishing_pressure <- c(.064,.02,.02,.02,.02,.02,0,0) # I'm going to leave all
of the 'upgrades' and fish prices as a black box and just focus on ecological
effects of different fishing levels.

fishing_sd <- c(.04,.01,.01,.01,.01,.005,0,0)

phyto_growth <- .16

pops <- tibble(tuna = 1, #these values were normalized
               sardines = 1, #to allow tracking relative
               haddock = 1, #to starting pop. Feel free
               herring = 1, #to change to values for game.
               flounder = 1, #Only thing to consider is that
               bivalves = 1, #they should be fairly big if you
               copepods = 1, #want to discretize them
               phytoplankton = 1)

contingency <- matrix(c( 0,.015,.013,.015,.015, 0, 0, 0, #tuna
                        0, 0, 0, 0, 0, 0,.03, 0, #sardines
                        0, 0, 0,.01, 0, 0,.03, 0, #haddock
                        0, 0, 0, 0, 0, 0,.035, 0, #herring
                        0, 0, 0, 0, 0,.05, 0, 0, #flounder
                        0, 0, 0, 0, 0, 0,.06, 0, #bivalves
                        0, 0, 0, 0, 0, 0, 0, .1, #copepods
                        0, 0, 0, 0, 0, 0, 0, 0 #phytoplankton
                        ), nrow = 8, byrow = TRUE )

for (t in 2:timesteps) {
  for (sp in 1:ncol(pops)) {
```

```

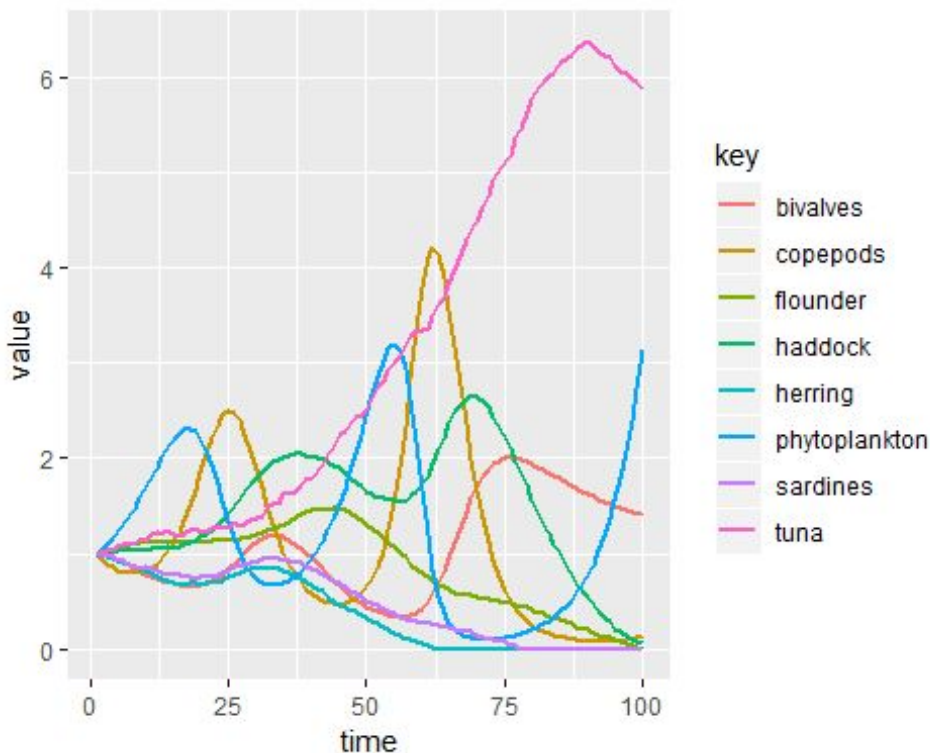
    pops[t, sp] <- pops[t-1, sp]*(1 + sum(contingency[sp,]*pops[t-1,] -
contingency[,sp]*pops[t-1,])) - # predation
      max(0, rnorm(1, fishing_pressure[sp], fishing_sd[sp])) #fishing,
stochasticity stands in for player behavior
    pops[t, sp] <- max(0, as.numeric(pops[t, sp]))
  }
  pops$phytoplankton[t] <- pops$phytoplankton[t]*(1+phyto_growth) #
production
}

```

```

pops %>%
  mutate(time = 1:timesteps) %>%
  gather(key, value, -time) %>%
  ggplot(aes(color = key, x= time, y = value)) +
  geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA))

```

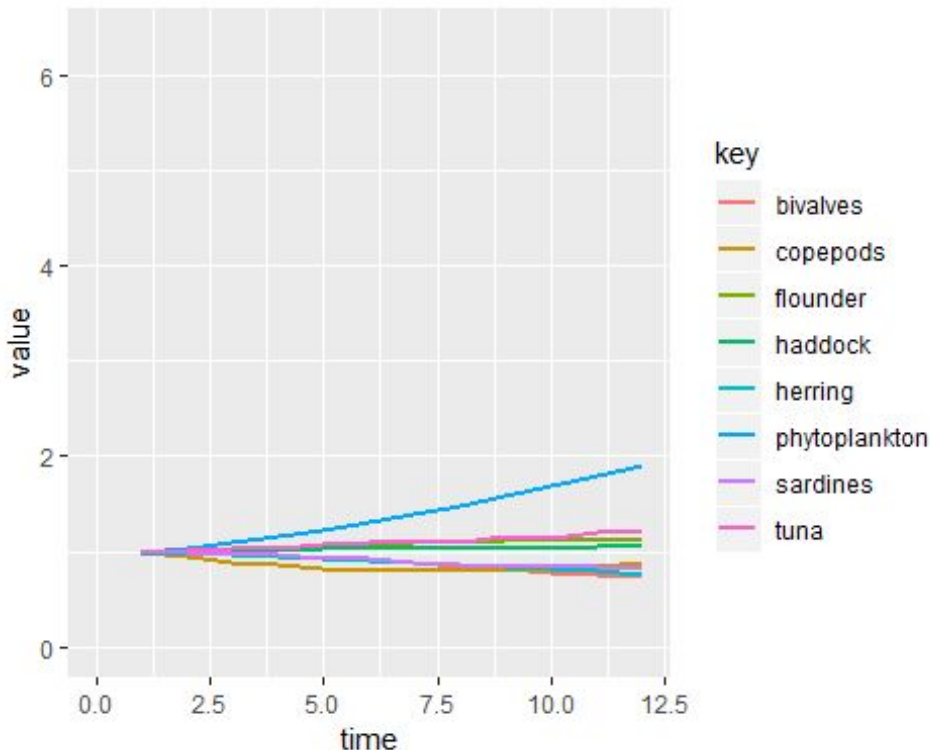


Success! That involved a lot of fiddling with numbers. I'm going to call it a night for now but the next steps are to reach a level of stability/volatility that works for the game, doing a few more purpose-minded versions, and then maybe to build it into the code, replacing 'fishing pressure' with the actual game's dynamics, and give it a few test drives?

Here are the first 10 or so timesteps with the current params, since that's closer to the game length:

```
pops %>%
  mutate(time = 1:timesteps) %>%
  gather(key, value, -time) %>%
  ggplot(aes(color = key, x= time, y = value)) +
  geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA)) +
  scale_x_continuous(limits = c(0,12))

## Warning: Removed 704 rows containing missing values (geom_path).
```



Since we went to the trouble of calculating those diversity indices, we might as well see how they compare. Here they are, calculated for all species and for fish. Note I use e^H to scale it to a weighted species richness.

```
pops %>%
  transmute(simpson = apply(pops, 1, simpson),
            shannon = exp(apply(pops, 1, shannon)),
            simpson_fish = apply(pops[,1:5], 1, simpson),
            shannon_fish = exp(apply(pops[,1:5], 1, shannon)),
            time = 1:timesteps) %>%
  gather(key, value, -time) %>%
  ggplot(aes(color = key, x= time, y = value)) +
  geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA))

## Warning: attributes are not identical across measure variables;
## they will be dropped
```

