# Simulating a fish population without fishing

Jaideep Joshi

28 March 2022

### Create a fish

A population is a collection of fish. Therefore to create a population, we first create a fish. This fish will be used a prototype to construct the population.

Here, we create a fish and set some parameters:

```
params_file = "../params/cod_params.ini"
```

Let us set parameters that allow us to reproduce the Dankel et al XXXX model.

```
fish = new(Fish, params_file)
fish$par$Bhalf = 3.65e8

fish$par$growth_model_name = "Dankel22"
fish$par$maturation_model_name = "Dankel22"
fish$par$mortality_model_name = "Dankel22"
fish$par$recruitment_model_name = "BevertonHoltDirect"
fish$trait_variances = c(5)

fish$init(1.93e3, 5.61)
fish$par$print()
fish$get_state()
```

```
## [1]  0.00000000  1.00000000  0.00000000  1.00000000 19.97761768  0.06654594
```

### Create a population

The constructor for the population requires a Fish object as an argument. This fish object will be used as a prototype to construct all fish in the population. You can create a fish population as follows:

```
pop = new(Population, fish)
```

#### Setting population parameters

Population parameters can be set in the same way as fish parameters, using the `par` object in the population

Two parameters are important:

- `n` - Superfish size. Since the population can have billions of fish, we do not simulate each individual fish. Rather, each "Fish" represents a group of `n` actual fish. This collection is called "Superfish". Reducing this parameter will necessitate simulating more agents in the model, which will reduce noise but also take more computational time.
- `Bhalf` - This is a parameter that controls density dependent recruitment.

```
pop$par$n = 2e6  # Each superfish contains so many fish
```

We can also set the management parameters for a population:

- Harvest proportion - the proportion of fish that are allowed to be harvested every year
- Minimum size limit - the size below which fish cannot be harvested

```
pop$set_harvestProp(0)   # Harvest
pop$set_minSizeLimit(45) # Only fish above 45 cm length can be harvested
```

### Initialising a population

By default, a population contains no fish. We must initialize the population with specified number of superfish.

```
pop$init(1000, 1.93e3, 5.61)  # initialize the population with 1000 agents (superfish)
```

We can also peek at the state of the population. The state of the population is simply a dataframe consisting of the state of each superfish in the population.

```
d = pop$get_state()
head(d)
```

```
##   t.birth age isMature isAlive   length     weight flag
## 1       1   1    FALSE    TRUE 19.97762 0.06654594    0
## 2       1   1    FALSE    TRUE 19.97762 0.06654594    0
## 3       1   1    FALSE    TRUE 19.97762 0.06654594    0
## 4       1   1    FALSE    TRUE 19.97762 0.06654594    0
## 5       1   1    FALSE    TRUE 19.97762 0.06654594    0
## 6       1   1    FALSE    TRUE 19.97762 0.06654594    0
```

### Simulating an unfished population

We can simulate a population by continuously `update`ing it. Each update performs 1 year of simulation, implementing the processes of maturation, growth, mortality, and reproduction for each fish in the population.

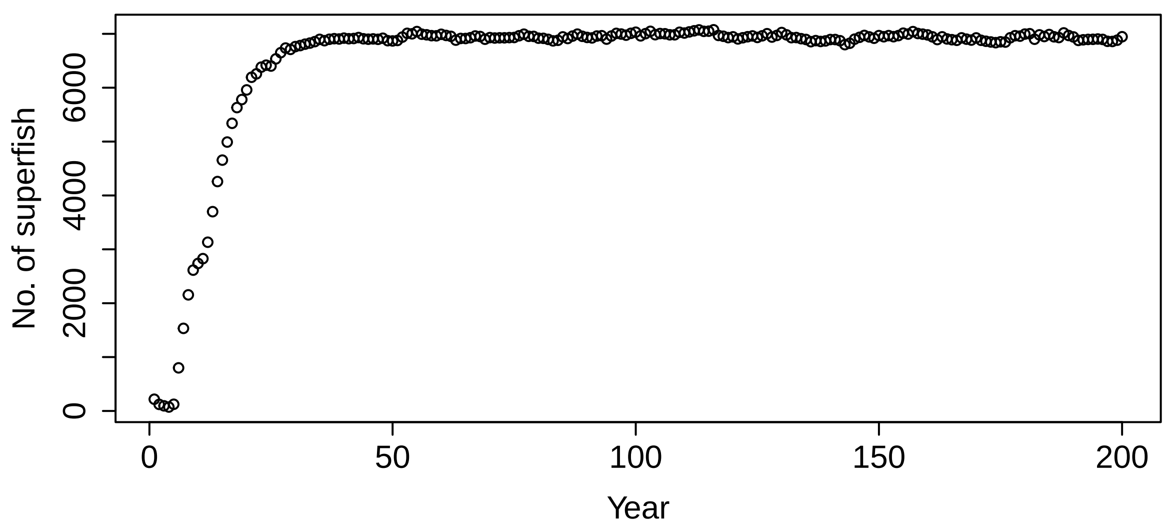
For example, let is simulate a population for 200 years.

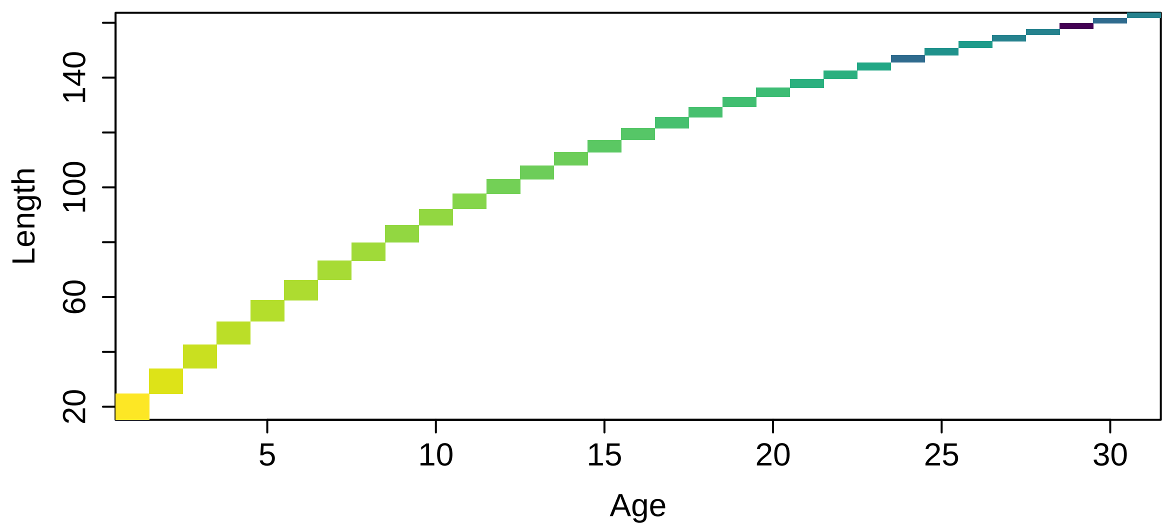
```
pop$verbose = T
nsteps = 200              # Let's simulate for 200 years
nfish = numeric(nsteps)   # Let's keep track of the number of superfish
nfish[1]=1000             # Since we initialized the population with 1000 superfish
cnames = pop$colnames
dat = data.frame(matrix(ncol=length(cnames), nrow=0))
colnames(dat) = cnames
pmrn_l50_dist = matrix(nrow=nsteps, ncol=100)
pmrn_l50_breaks = seq(50,250,length.out=101)
for (i in 1:nsteps){
  v = pop$update(5.61)        # Update all fish over 1 year
  dat[i,] = v     # some book-keeping
  traits = pop$get_traits()
  h = hist(traits$pmrn_l50, breaks = pmrn_l50_breaks, plot=F)
  pmrn_l50_dist[i,] = h$density
  nfish[i] = pop$nfish()
}

d = pop$get_state()
dist = table(d$age, d$length)

par(mfrow = c(3,1), mar=c(5,5,1,1), oma=c(1,1,1,1), cex.lab=1.5, cex.axis=1.5)
```

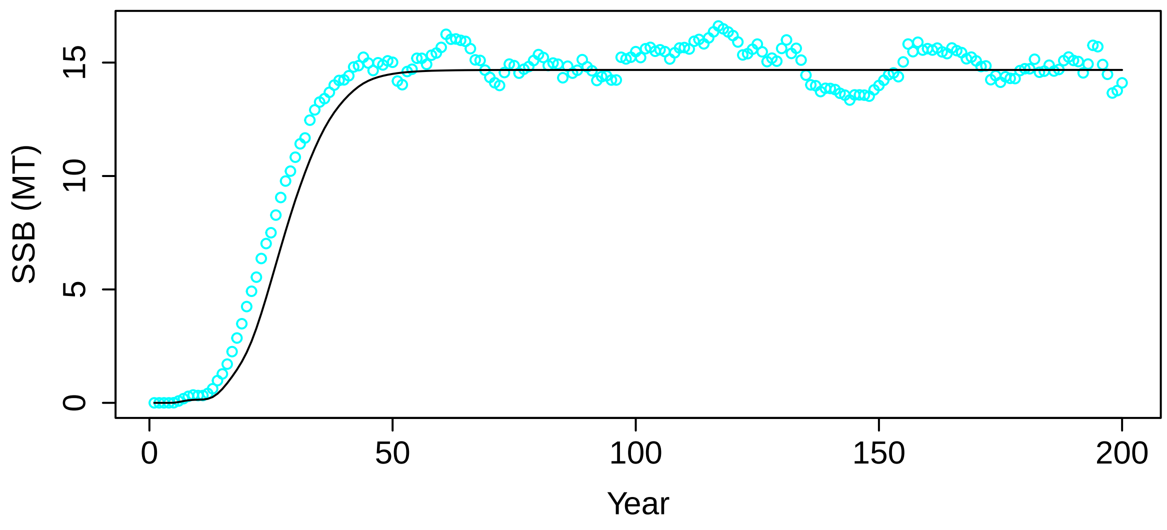
```r
plot(nfish~seq(1,nsteps,1), ylab="No. of superfish", xlab="Year")
image(x=as.numeric(rownames(dist)), y = as.numeric(colnames(dist)), z=log(1+3*log(dist)), col=scales::vi
```

```
## Warning in log(1 + 3 * log(dist)): NaNs produced
```

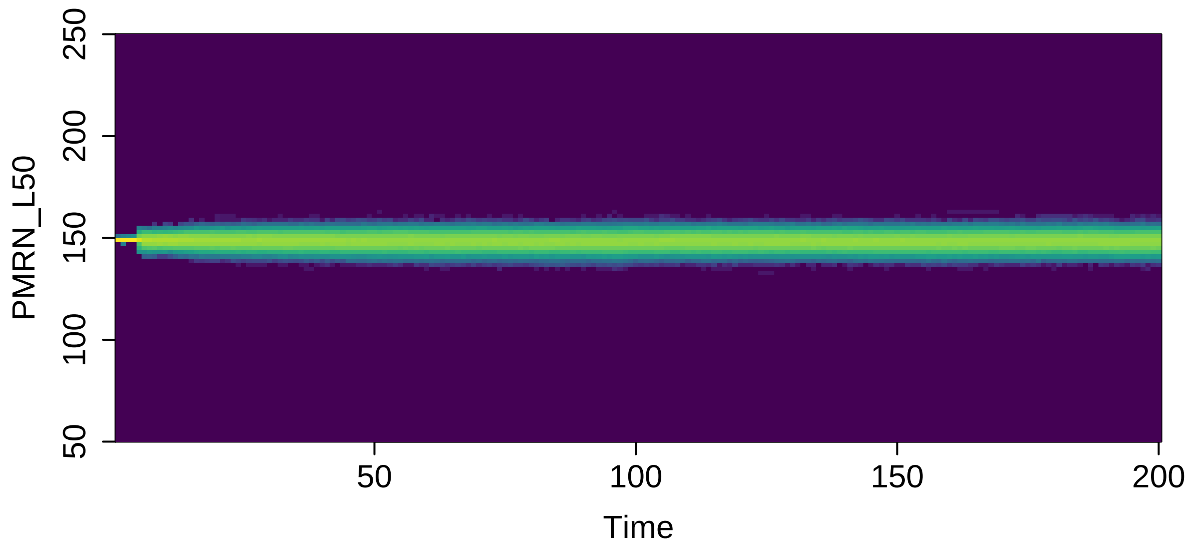
```r
res = simulate(0, 45, F)
matplot(cbind(dat$ssb/1e9, res$summaries$SSB[1:200]/1e9), ylab="SSB (MT)", xlab="Year", col=c("cyan", "
```


No. of superfish
0
2000
4000
6000
0
50
100
150
200
Year


Length
20
60
100
140
5
10
15
20
25
30
Age


SSB (MT)
0
5
10
15
0
50
100
150
200
Year

```
image(y=h$mids, x=1:nsteps, z=log(1e-4+pmrn_l50_dist), col=scales::viridis_pal()(100), xlab="Time", ylal
```


PMRN_L50
250
200
150
100
50
50
100
150
200
Time

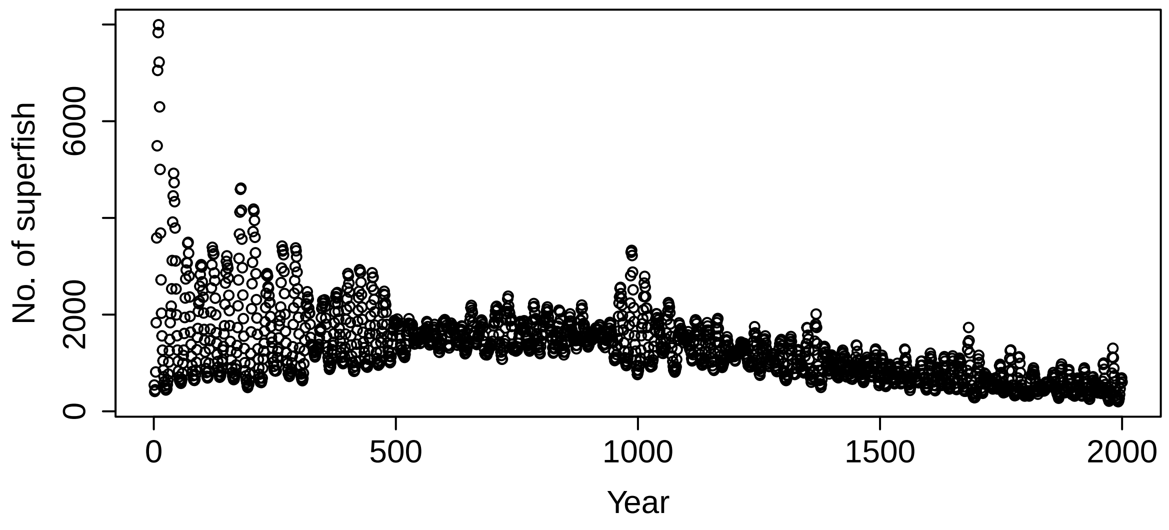
**Bioenergetic model**

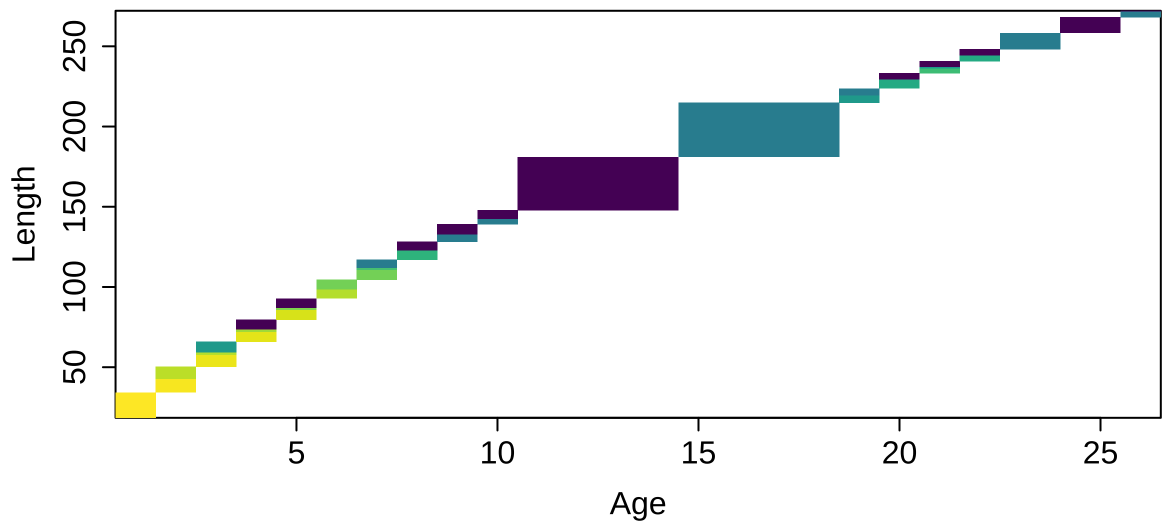
Now, let's simulate the bioenergetic model and compare the results with the model of Dankel et al.

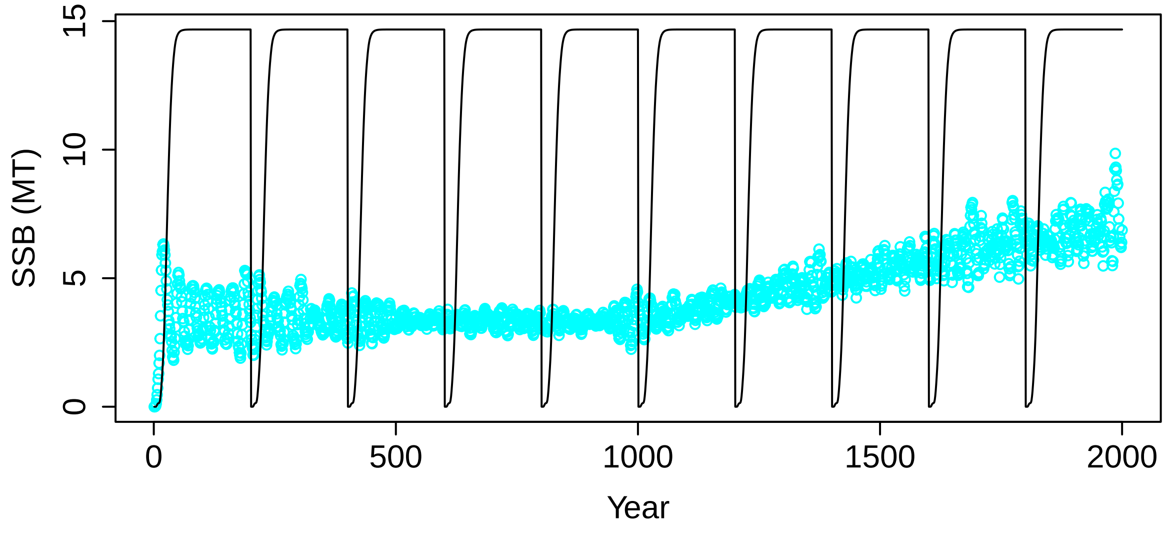
```
fish = new(Fish, params_file)
fish$par$s0 = 0.059
fish$trait_variances = c(30)

pop = new(Population, fish)
pop$par$n = 1e6  # Each superfish contains so many fish
pop$init(1000, 1.93e3, 5.61)  # initialize the population with 1000 agents (superfish)

pop$verbose = T
nsteps = 2000              # Let's simulate for 200 years
nfish = numeric(nsteps)  # Let's keep track of the number of superfish
nfish[1]=1000            # Since we initialized the population with 1000 superfish

temp = rep(5.61, nsteps)
temp[(nsteps/2+1):nsteps] = seq(5.61,10, length.out=nsteps/2)

cnames = pop$colnames
dat = data.frame(matrix(ncol=length(cnames), nrow=0))
colnames(dat) = cnames
pmrn_l50_dist = matrix(nrow=nsteps, ncol=100)
pmrn_l50_breaks = seq(0,300,length.out=101)
pmrn_l50_mean = numeric(nsteps)
for (i in 1:nsteps){
  v = pop$update(temp[i])       # Update all fish over 1 year
  dat[i,] = v    # some book-keeping
  traits = pop$get_traits()
  h = hist(traits$pmrn_l50, breaks = pmrn_l50_breaks, plot=F)
  pmrn_l50_dist[i,] = h$density
  pmrn_l50_mean[i] = mean(traits$pmrn_l50)
  nfish[i] = pop$nfish()
}

d = pop$get_state()
dist = table(d$age, d$length)

par(mfrow = c(3,1), mar=c(5,5,1,1), oma=c(1,1,1,1), cex.lab=1.5, cex.axis=1.5)
plot(nfish~seq(1,nsteps,1), ylab="No. of superfish", xlab="Year")
image(x=as.numeric(rownames(dist)), y = as.numeric(colnames(dist)), z=log(1+3*log(dist)), col=scales::vi
```

```
## Warning in log(1 + 3 * log(dist)): NaNs produced
```

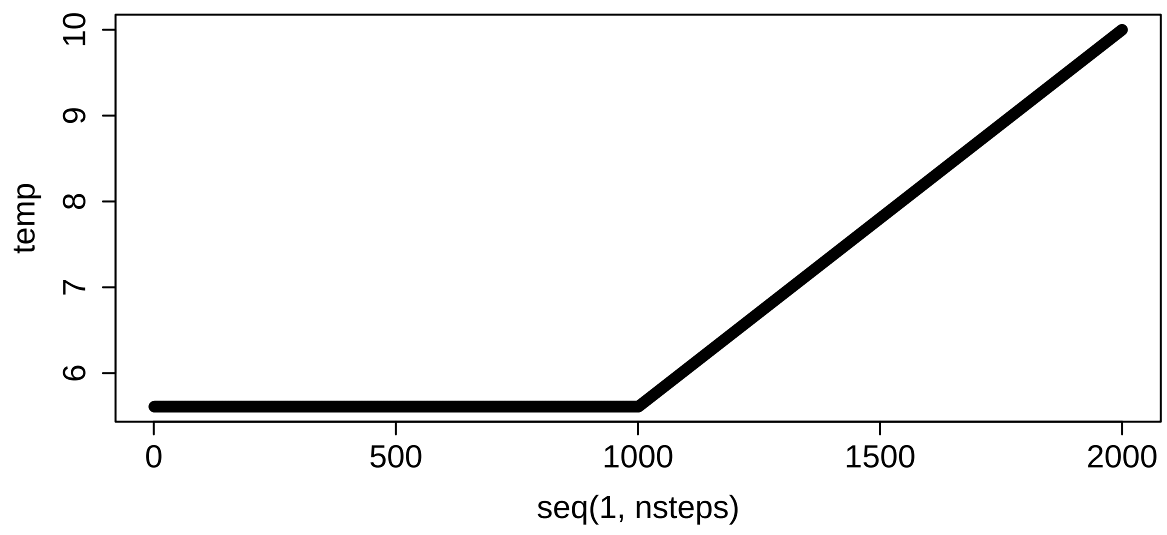
```
res = simulate(0, 45, F)
matplot(cbind(dat$ssb/1e9, res$summaries$SSB[1:200]/1e9), ylab="SSB (MT)", xlab="Year", col=c("cyan", "
```

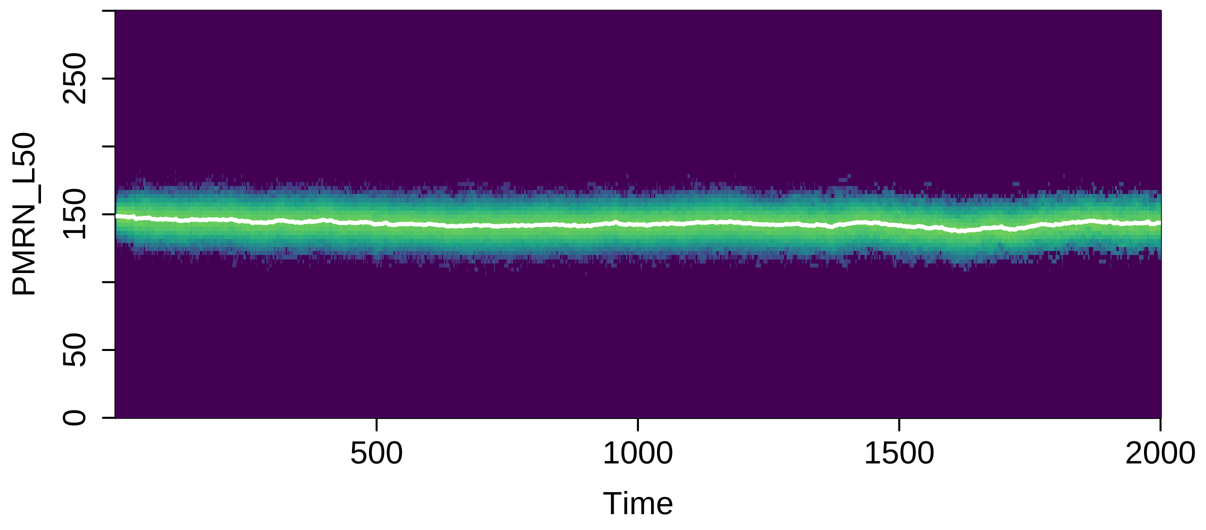

No. of superfish
0
2000
6000
0
500
1000
1500
2000
Year


Length
50
100
150
200
250
5
10
15
20
25
Age


SSB (MT)
0
5
10
15
0
500
1000
1500
2000
Year

```
par(mfrow = c(3,1), mar=c(5,5,1,1), oma=c(1,1,1,1), cex.lab=1.5, cex.axis=1.5)
plot(temp~seq(1,nsteps))

image(y=h$mids, x=1:nsteps, z=log(1e-4+pmrn_l50_dist), col=scales::viridis_pal()(100), xlab="Time", ylal
lines(y=pmrn_l50_mean, x=1:nsteps, col="white", lwd=2)
```


temp
10
9
8
7
6
0
500
1000
1500
2000
seq(1, nsteps)


PMRN_L50
250
150
50
0
500
1000
1500
2000
Time