Simple MixFishSim Example

This is a simple example of how to use **'MixFishSim'** to generate simulations of the dynamics in a mixed fishery. We describe how to calibrate the habitat fields, the population models, the fishery model and implement a simple fixed spatial closure. \

First, load the packages and set a seed for reproducibility.

Load MixFishSim

```
library(MixFishSim)
library(knitr)
opts_chunk$set(tidy = TRUE)

set.seed(123)
```

Initialise the simulation

This vignette is a paired down example of how to construct a simulation using MixFishSim. We include only a basic example and encourage users to explore the other features of the package. \

Base parameters

First we specify the basic parameters of the simulation. This includes the dimensions of the spatial domain, the number of years to simulate, the number of fleets and vessels per fleet and the number of species and how often (in weeks) the fish move.

The object returned is used internally by MixFishSim a list with two levels:

- sim$idx : The different units of different processes
- sim$brk.idx: breaks for each of the key processes in units of a timestep

```
sim <- init_sim(nrows = 10, ncols = 10, n_years = 10, n_tows_day = 4, n_days_wk_fished = 5,
    n_fleets = 2, n_vessels = 20, n_species = 2, move_freq = 2)

class(sim)
```

```
## [1] "list"
```

```
sim$idx
```

```
##         ntd          ndf           nw          nwm           nt           nm
##    4.000000     5.000000    52.000000     4.333333    26.000000    12.000000
##          ny         ntow      ntow.py        n.spp        ncols        nrows
##   10.000000 10400.000000  1040.000000     2.000000    10.000000    10.000000
##          nf           nv
##    2.000000    20.000000
```

```r
names(sim$brk.idx)
```

```
## [1] "tow.breaks"   "day.seq"      "day.breaks"   "trip.breaks"  "week.breaks"
## [6] "month.breaks" "year.breaks"
```

Habitat setup

This function creates the spatial fields which support the fish populations and determine their spatial distributions. You define the parameters for the matern covariance function for each population and optionally the location of any spawning closure areas.

It returns a list of suitable habitat for each species (hab), the habitat as adjusted during the spawning period (spwn_hab) and the binary location of spawning areas (spwn_loc). It also returns the locations as x1,x2,y1,y2 and the multiplier of attractiveness to the spawning area during spawning periods (spwn_mult).

If plot.dist = TRUE, it returns the plots to a file.

```r
hab <- create_hab(sim_init = sim, spp.ctrl = list(spp.1 = list(nu = 1/0.015, var = 1,
    scale = 1, Aniso = matrix(nc = 2, c(1.5, 3, -3, 4))), spp.2 = list(nu = 1/0.05,
    var = 2, scale = 12, Aniso = matrix(nc = 2, c(1, 2, -1, 2)))), spawn_areas = list(spp1 = list(area1
    5, 2, 5), area2 = c(6, 8, 6, 8)), spp2 = list(area1 = c(5, 6, 6, 6)), spwn_mult = 10,
    plot.dist = FALSE))

print(hab)
```

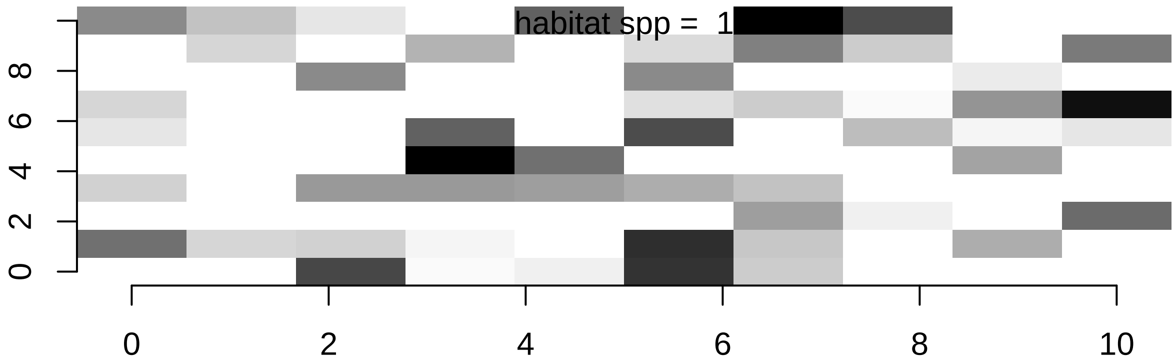
```
## $hab
## $hab$spp1
##              [,1]        [,2]        [,3]       [,4]       [,5]        [,6]
##  [1,] 0.000000000 0.029825902 0.000000000 0.01039088 0.00000000 0.006172062
##  [2,] 0.000000000 0.008892648 0.000000000 0.00000000 0.00000000 0.000000000
##  [3,] 0.037956246 0.009801995 0.000000000 0.02177983 0.00000000 0.000000000
##  [4,] 0.001875668 0.002737579 0.000000000 0.02148636 0.05272189 0.033344790
##  [5,] 0.003157360 0.000000000 0.000000000 0.02010557 0.02965351 0.000000000
##  [6,] 0.041802396 0.043484595 0.000000000 0.01686307 0.00000000 0.036926873
##  [7,] 0.011404170 0.012310136 0.020240954 0.01356676 0.00000000 0.000000000
##  [8,] 0.000000000 0.000000000 0.003822176 0.00000000 0.00000000 0.014088794
##  [9,] 0.000000000 0.016889727 0.000000000 0.00000000 0.01895698 0.003077039
## [10,] 0.000000000 0.000000000 0.030435272 0.00000000 0.00000000 0.005274148
##              [,7]        [,8]         [,9]      [,10]
##  [1,] 0.009250407 0.000000000 0.0001379017 0.02420836
##  [2,] 0.000000000 0.000000000 0.0093889316 0.01346243
##  [3,] 0.000000000 0.024271144 0.0000000000 0.00587302
##  [4,] 0.000000000 0.000000000 0.0156625272 0.00000000
##  [5,] 0.000000000 0.000000000 0.0000000000 0.03309048
##  [6,] 0.007286046 0.024920046 0.0080618195 0.00000000
##  [7,] 0.010952094 0.000000000 0.0267578154 0.05323659
##  [8,] 0.001337087 0.000000000 0.0107145287 0.03756461
##  [9,] 0.022477609 0.004296533 0.0000000000 0.00000000
## [10,] 0.050045746 0.000000000 0.0279588814 0.00000000
##
## $hab$spp2
##       [,1] [,2] [,3] [,4]       [,5]       [,6]       [,7]       [,8]
##  [1,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [2,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [3,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
```

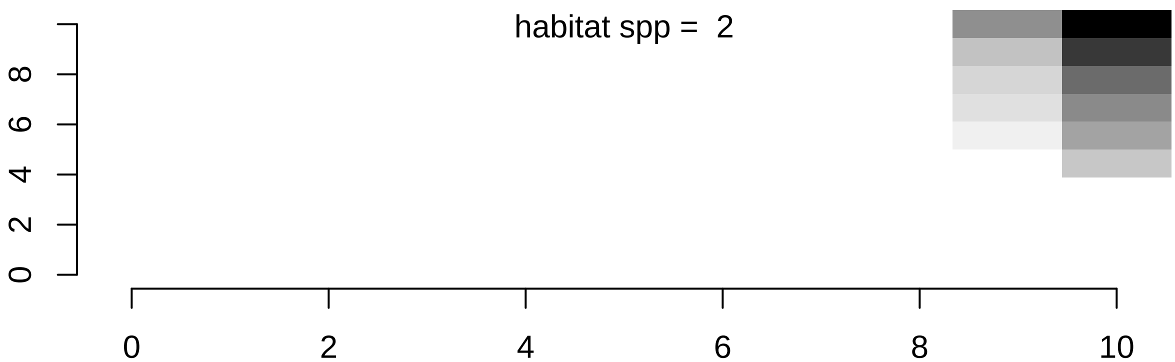
```
##  [4,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [5,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [6,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [7,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [8,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [9,]    0    0    0    0 0.00000000 0.01503961 0.02789571 0.03571187
## [10,]    0    0    0    0 0.05190594 0.08078570 0.10187944 0.13122104
##             [,9]     [,10]
##  [1,] 0.00000000 0.0000000
##  [2,] 0.00000000 0.0000000
##  [3,] 0.00000000 0.0000000
##  [4,] 0.00000000 0.0000000
##  [5,] 0.00000000 0.0000000
##  [6,] 0.00000000 0.0000000
##  [7,] 0.00000000 0.0000000
##  [8,] 0.00000000 0.0000000
##  [9,] 0.05709544 0.0988671
## [10,] 0.17483444 0.2247637
##
##
## $spwn_hab
## $spwn_hab$spp1
##               [,1]        [,2]        [,3]        [,4]        [,5]         [,6]
##  [1,] 0.0000000000 0.008865640 0.000000000 0.003088652 0.000000000 0.0018346228
##  [2,] 0.0000000000 0.026433070 0.000000000 0.000000000 0.000000000 0.0000000000
##  [3,] 0.0112823555 0.029136071 0.000000000 0.064739751 0.000000000 0.0000000000
##  [4,] 0.0005575354 0.008137354 0.000000000 0.063867416 0.156713904 0.0099116170
##  [5,] 0.0009385138 0.000000000 0.000000000 0.059763071 0.088143981 0.0000000000
##  [6,] 0.0124256094 0.012925637 0.000000000 0.005012486 0.000000000 0.1097637792
##  [7,] 0.0033898479 0.003659143 0.006016549 0.004032671 0.000000000 0.0000000000
##  [8,] 0.0000000000 0.000000000 0.001136128 0.000000000 0.000000000 0.0418784240
##  [9,] 0.0000000000 0.005020410 0.000000000 0.000000000 0.005634894 0.0009146385
## [10,] 0.0000000000 0.000000000 0.009046773 0.000000000 0.000000000 0.0015677214
##              [,7]        [,8]         [,9]       [,10]
##  [1,] 0.002749650 0.000000000 4.099077e-05 0.007195847
##  [2,] 0.000000000 0.000000000 2.790826e-03 0.004001657
##  [3,] 0.000000000 0.007214509 0.000000e+00 0.001745734
##  [4,] 0.000000000 0.000000000 4.655629e-03 0.000000000
##  [5,] 0.000000000 0.000000000 0.000000e+00 0.009836026
##  [6,] 0.021657505 0.074073925 2.396346e-03 0.000000000
##  [7,] 0.032554700 0.000000000 7.953663e-03 0.015824380
##  [8,] 0.003974443 0.000000000 3.184855e-03 0.011165944
##  [9,] 0.006681387 0.001277129 0.000000e+00 0.000000000
## [10,] 0.014875915 0.000000000 8.310675e-03 0.000000000
##
## $spwn_hab$spp2
##       [,1] [,2] [,3] [,4]       [,5]       [,6]       [,7]       [,8]
##  [1,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [2,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [3,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [4,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [5,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [6,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [7,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
```

```
##  [8,]    0    0    0    0 0.00000000 0.00000000 0.00000000 0.00000000
##  [9,]    0    0    0    0 0.00000000 0.01503961 0.02789571 0.03571187
## [10,]    0    0    0    0 0.05190594 0.08078570 0.10187944 0.13122104
##             [,9]     [,10]
##  [1,] 0.00000000 0.0000000
##  [2,] 0.00000000 0.0000000
##  [3,] 0.00000000 0.0000000
##  [4,] 0.00000000 0.0000000
##  [5,] 0.00000000 0.0000000
##  [6,] 0.00000000 0.0000000
##  [7,] 0.00000000 0.0000000
##  [8,] 0.00000000 0.0000000
##  [9,] 0.05709544 0.0988671
## [10,] 0.17483444 0.2247637
##
##
## $spwn_loc
## $spwn_loc$spp1
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    0    0    0    0    0    0    0    0     0
##  [2,]    0    1    1    1    1    0    0    0    0     0
##  [3,]    0    1    1    1    1    0    0    0    0     0
##  [4,]    0    1    1    1    1    0    0    0    0     0
##  [5,]    0    1    1    1    1    0    0    0    0     0
##  [6,]    0    0    0    0    0    1    1    1    0     0
##  [7,]    0    0    0    0    0    1    1    1    0     0
##  [8,]    0    0    0    0    0    1    1    1    0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0
## [10,]    0    0    0    0    0    0    0    0    0     0
##
## $spwn_loc$spp2
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    0    0    0    0    0    0    0    0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0
##  [3,]    0    0    0    0    0    0    0    0    0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0
##  [5,]    0    0    0    0    0    1    0    0    0     0
##  [6,]    0    0    0    0    0    1    0    0    0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0
## [10,]    0    0    0    0    0    0    0    0    0     0
##
##
## $spawn_areas
## $spawn_areas$spp1
## $spawn_areas$spp1$area1
## [1] 2 5 2 5
##
## $spawn_areas$spp1$area2
## [1] 6 8 6 8
##
##
## $spawn_areas$spp2
```

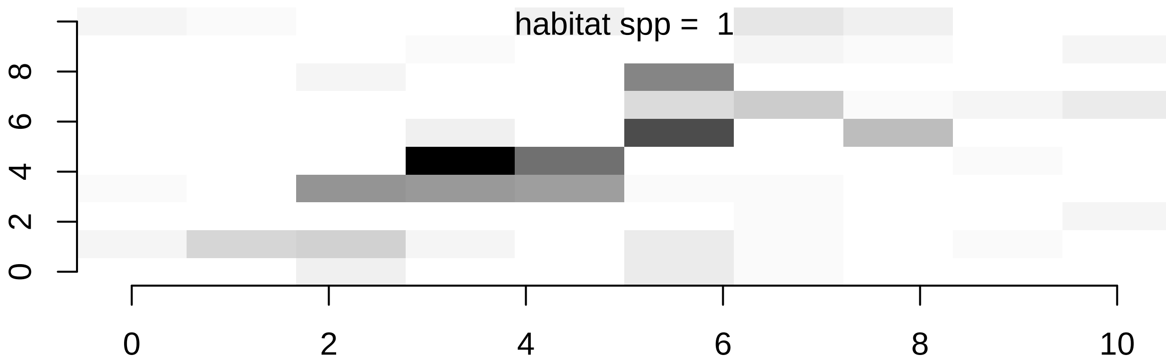
```
## $spawn_areas$spp2$area1
## [1] 5 6 6 6
##
##
## $spawn_areas$spwn_mult
## [1] 10
##
## $spawn_areas$plot.dist
## [1] FALSE
```

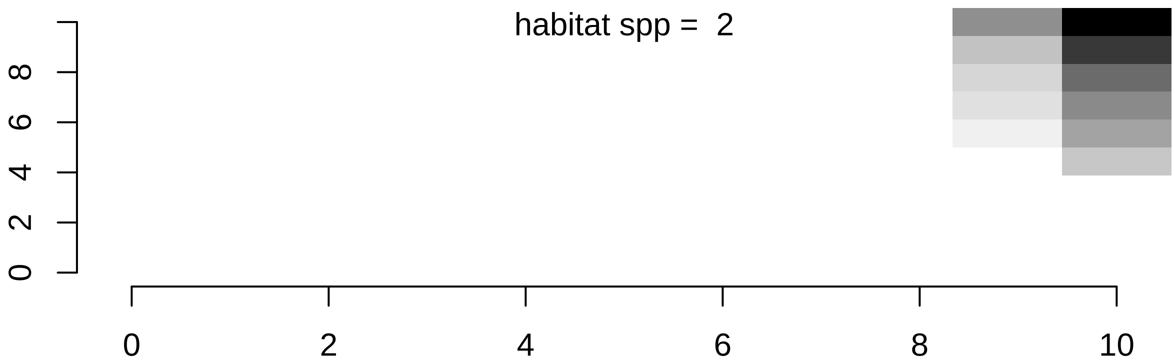
```r
## Plot the unadjusted habitat fields
plot_habitat(hab$hab)
```






```r
## Plot the adjusted habitat fields
plot_habitat(hab$spwn_hab)
```





Population models

Now we need to set up the population models for the simulations. We do this with the init_pop function. We set the initial population biomasses, movement rates, recruitment parameter and growth and natural mortality rates.

The object created stores all the starting conditions and containers for recording the changes in the populations during the simulations.

We can plot the starting distributions for each population as a check.

```
Pop <- init_pop(sim_init = sim, Bio = c(spp1 = 1e+05, spp2 = 1e+05), hab = hab[["hab"]],
    start_cell = c(5, 5), lambda = c(spp1 = 0.1, spp2 = 0.1), init_move_steps = 20,
    rec_params = list(spp1 = c(model = "BH", a = 54, b = 2, cv = 0.7), spp2 = c(model = "BH",
        a = 27, b = 4, cv = 0.3)), rec_wk = list(spp1 = 3:6, spp2 = 4:8), spwn_wk = list(spp1 = 4:8,
        spp2 = 4:8), M = c(spp1 = 0.2, spp2 = 0.2), K = c(spp1 = 0.3, spp2 = 0.3))

names(Pop)
```

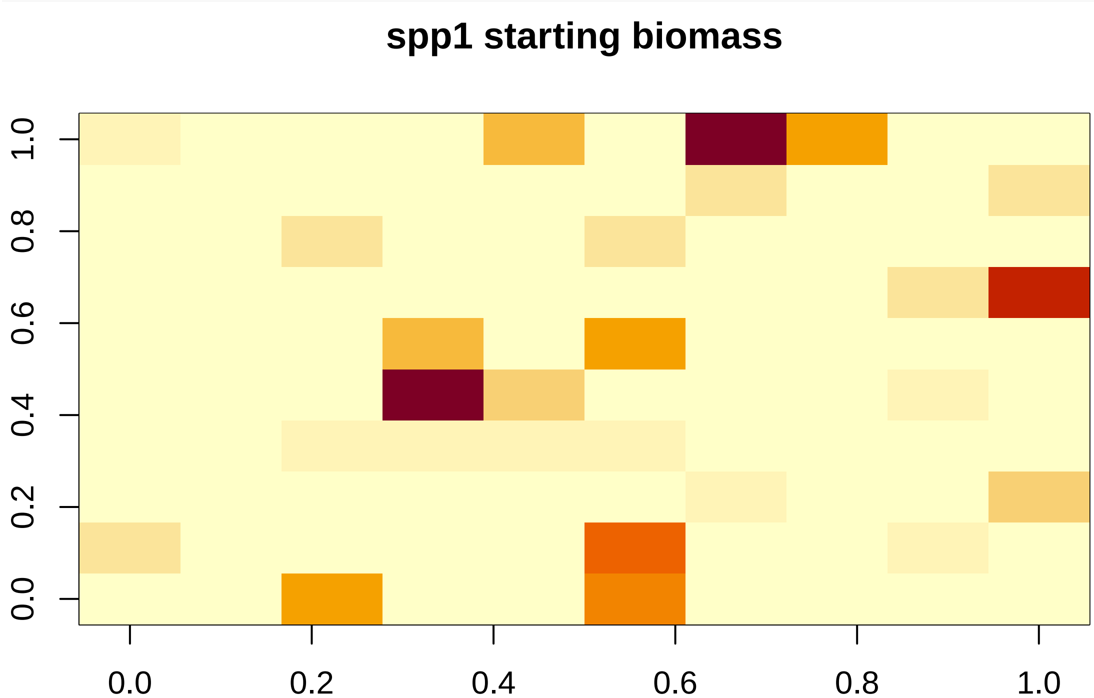
```
## [1] "Pop_record" "Start_pop"  "dem_params"
```

```
Pop$dem_params
```

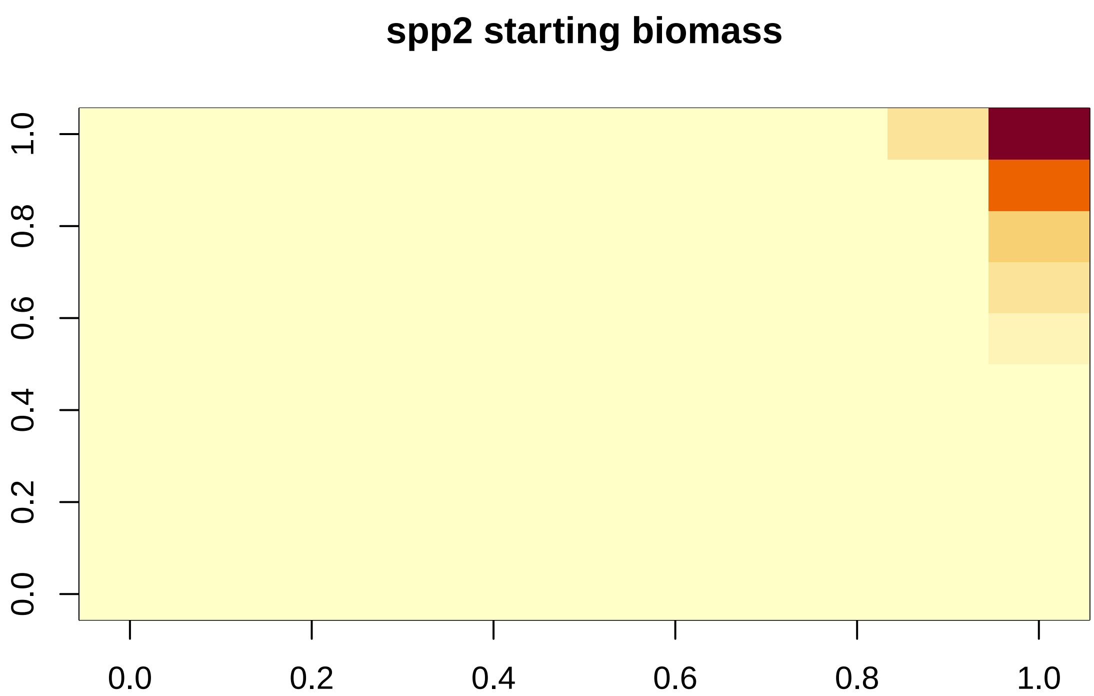
```
## $spp1
## $spp1$rec_params
## model     a     b    cv
##  "BH"  "54"   "2" "0.7"
##
## $spp1$rec_wk
## [1] 3 4 5 6
```

```
##
## $spp1$spwn_wk
## [1] 4 5 6 7 8
##
## $spp1$M
## [1] 0.2
##
## $spp1$K
## [1] 0.3
##
##
## $spp2
## $spp2$rec_params
## model      a      b     cv
##  "BH"   "27"    "4" "0.3"
##
## $spp2$rec_wk
## [1] 4 5 6 7 8
##
## $spp2$spwn_wk
## [1] 4 5 6 7 8
##
## $spp2$M
## [1] 0.2
##
## $spp2$K
## [1] 0.3
```

```
image(Pop$Start_pop[[1]], main = "spp1 starting biomass")
```



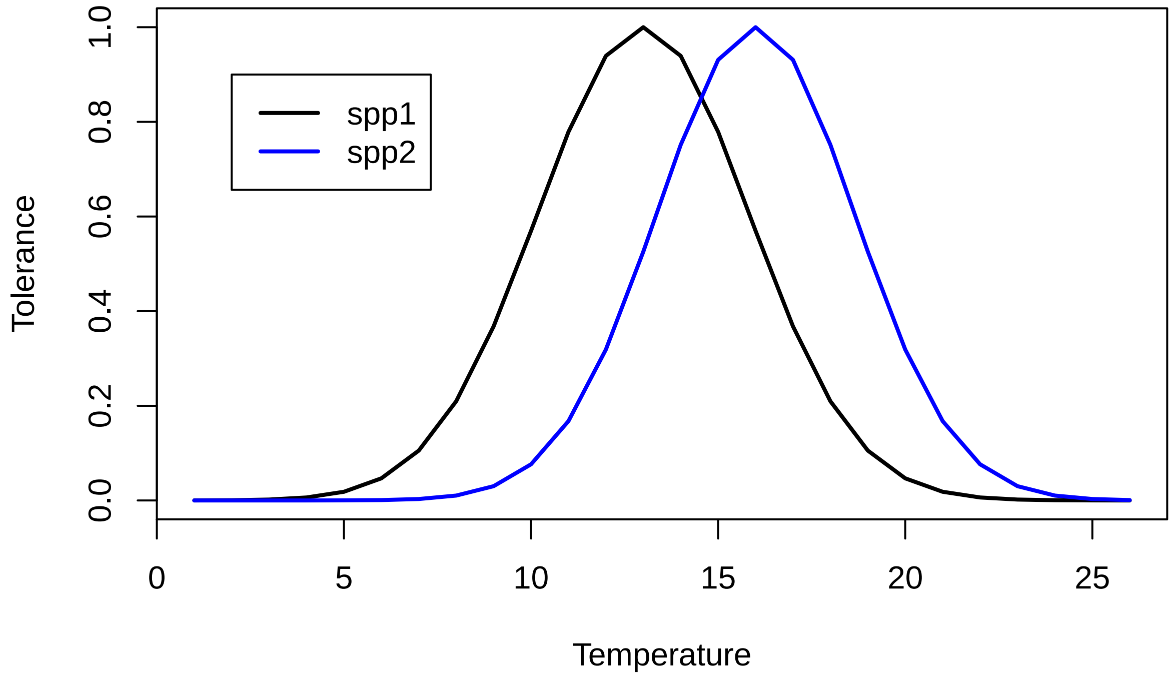

```
image(Pop$Start_pop[[2]], main = "spp2 starting biomass")
```



Population movement

Now we set up the population tolerance to different temperatures which determines how the populations move during the course of a year. We can then plot the combined spatiotemporal suitable habitat to examine how these interact.

```r
moveCov <- init_moveCov(sim_init = sim, steps = 52, spp_tol = list(spp1 = list(mu = 12,
    va = 8), spp2 = list(mu = 15, va = 7)))

plot(norm_fun(x = 0:25, mu = 12, va = 8)/max(norm_fun(0:25, 12, 8)), type = "l",
    xlab = "Temperature", ylab = "Tolerance", lwd = 2)
lines(norm_fun(x = 0:25, mu = 15, va = 7)/max(norm_fun(0:25, 15, 7)), type = "l",
    col = "blue", lwd = 2)
legend(x = 2, y = 0.9, legend = c("spp1", "spp2"), lwd = 2, col = c("black", "blue"))
```



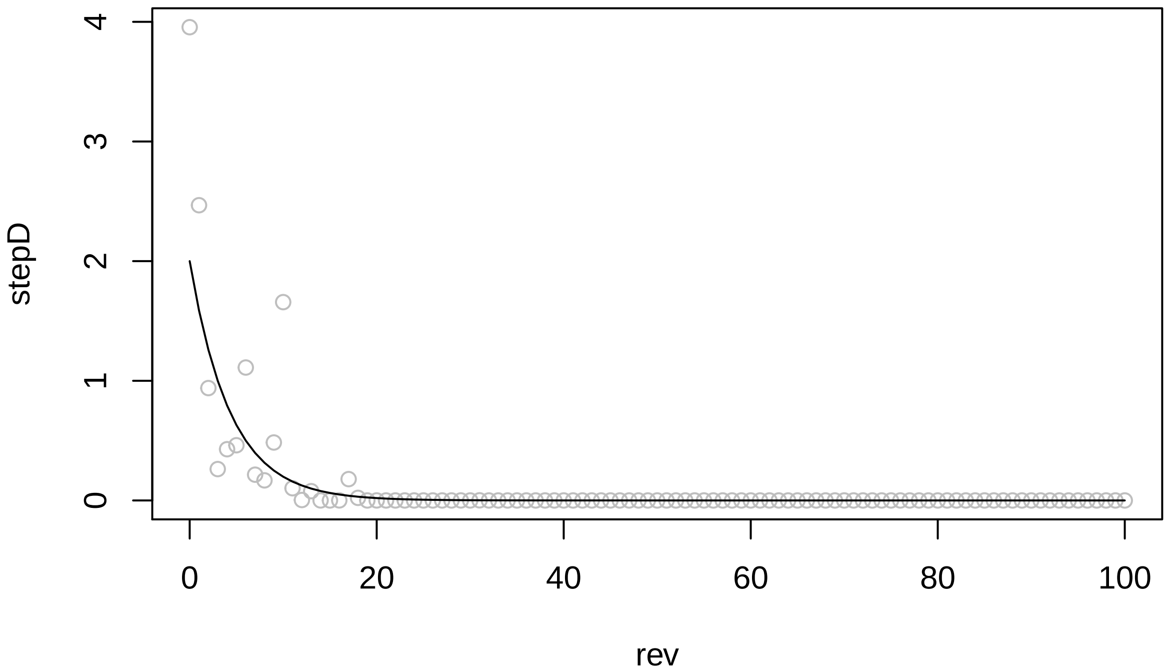

```r
plot_spatiotemp_hab(hab = hab, moveCov = moveCov, spwn_wk = list(spp1 = 4:8, spp2 = 4:8))
```

Fleet models

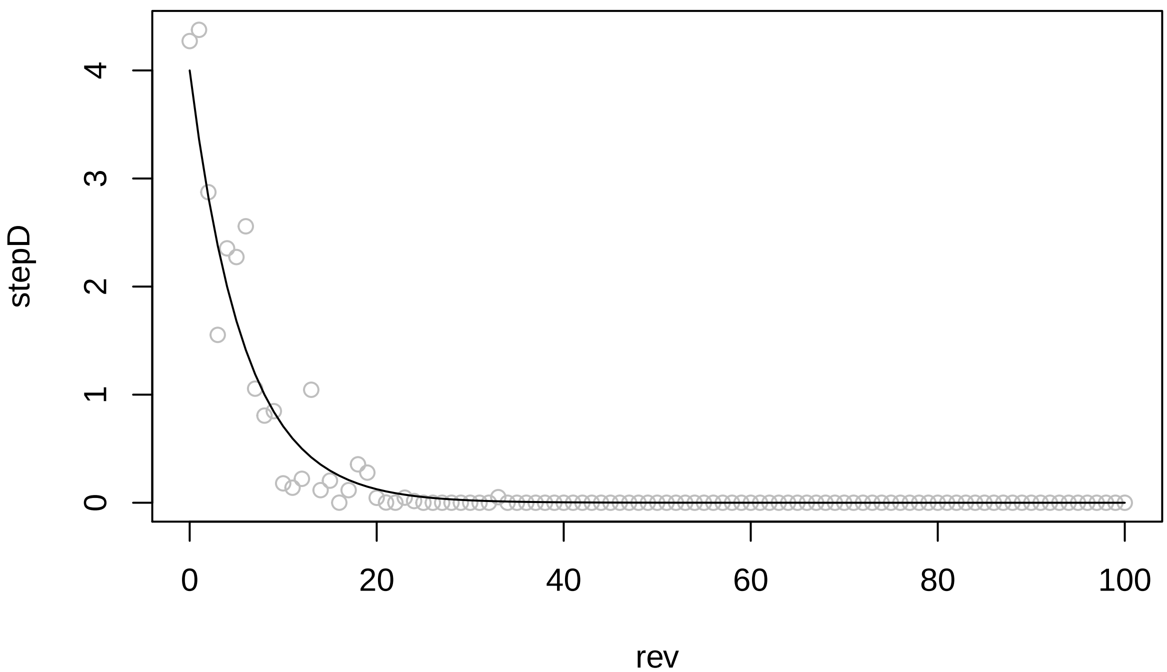
Here we initialise the fleet with fish landings price per tonne, catchability coefficients per population, fuel cost, the coefficients for the step function and fleet behaviour.

We can plot the behaviour of the step function to check its suitable for our simulations. This determines the relationship between the monetary value gained from a fishing tow and the next move by the vessel when using the correlated random walk function.

```r
fleets <- init_fleet(sim_init = sim, VPT = list(spp1 = 4, spp2 = 3), Qs = list(`fleet 1` = c(spp1 = 1e-0
    spp2 = 3e-05), `fleet 2` = c(spp1 = 5e-05, spp2 = 1e-05)), fuelC = list(fleet1 = 3,
    `fleet 2` = 8), step_params = list(`fleet 1` = c(rate = 3, B1 = 1, B2 = 2, B3 = 3),
    `fleet 2` = c(rate = 3, B1 = 2, B2 = 4, B3 = 4)), past_knowledge = TRUE, past_year_month = TRUE,
    past_trip = TRUE, threshold = 0.7)

test_step(step_params = fleets$fleet_params[[1]]$step_params, rev.max = 100)
```




```r
test_step(step_params = fleets$fleet_params[[2]]$step_params, rev.max = 100)
```



Spatial closure

We set up a spatial closure. There are multiple options in defining this, but we simply define a static fixed site closure for demonstration purposes.

```r
closure <- init_closure(input_coords = data.frame(x = c(9, 10), y = c(6, 10)), spp1 = "spp1",
    year_start = 5)
```

Survey

Its also possible to define a survey design using the init_survey function, but we do not do so for this demonstration. Please refer to the function help file if this is required.

Run simulation

Finally we run the simulation. The output is a list of objects containing all the information on fisheries catches, the population dynamics and population distributions. These can be examined with some inbuilt plotting functions.

```
res <- run_sim(sim_init = sim, pop_init = Pop, move_cov = moveCov, fleets_init = fleets,
    hab_init = hab, save_pop_bio = TRUE, survey = NULL, closure = closure)
```

```
## [1] "Calculating movement probabilities"
## [1] "You are implementing spatial closures...."
## [1] "----------year 1 -----------"
## [1] "tow == 1 ---- 0 %"
## [1] "tow == 101 ---- 1 %"
## [1] "tow == 201 ---- 2 %"
## [1] "tow == 301 ---- 3 %"
## [1] "tow == 401 ---- 4 %"
## [1] "tow == 501 ---- 5 %"
## [1] "tow == 601 ---- 6 %"
## [1] "tow == 701 ---- 7 %"
## [1] "tow == 801 ---- 8 %"
## [1] "tow == 901 ---- 9 %"
## [1] "tow == 1001 ---- 10 %"
## [1] "----------year 2 -----------"
## [1] "tow == 1101 ---- 11 %"
## [1] "tow == 1201 ---- 12 %"
## [1] "tow == 1301 ---- 13 %"
## [1] "tow == 1401 ---- 13 %"
## [1] "tow == 1501 ---- 14 %"
## [1] "tow == 1601 ---- 15 %"
## [1] "tow == 1701 ---- 16 %"
## [1] "tow == 1801 ---- 17 %"
## [1] "tow == 1901 ---- 18 %"
## [1] "tow == 2001 ---- 19 %"
## [1] "----------year 3 -----------"
## [1] "tow == 2101 ---- 20 %"
## [1] "tow == 2201 ---- 21 %"
## [1] "tow == 2301 ---- 22 %"
## [1] "tow == 2401 ---- 23 %"
## [1] "tow == 2501 ---- 24 %"
## [1] "tow == 2601 ---- 25 %"
## [1] "tow == 2701 ---- 26 %"
## [1] "tow == 2801 ---- 27 %"
## [1] "tow == 2901 ---- 28 %"
## [1] "tow == 3001 ---- 29 %"
## [1] "tow == 3101 ---- 30 %"
## [1] "----------year 4 -----------"
## [1] "tow == 3201 ---- 31 %"
## [1] "tow == 3301 ---- 32 %"
```

```
## [1] "tow == 3401 ---- 33 %"
## [1] "tow == 3501 ---- 34 %"
## [1] "tow == 3601 ---- 35 %"
## [1] "tow == 3701 ---- 36 %"
## [1] "tow == 3801 ---- 37 %"
## [1] "tow == 3901 ---- 38 %"
## [1] "tow == 4001 ---- 38 %"
## [1] "tow == 4101 ---- 39 %"
## [1] "----------year 5 -----------"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 4201 ---- 40 %"
## [1] "tow == 4301 ---- 41 %"
## [1] "tow == 4401 ---- 42 %"
## [1] "tow == 4501 ---- 43 %"
## [1] "tow == 4601 ---- 44 %"
## [1] "tow == 4701 ---- 45 %"
## [1] "tow == 4801 ---- 46 %"
## [1] "tow == 4901 ---- 47 %"
## [1] "tow == 5001 ---- 48 %"
## [1] "tow == 5101 ---- 49 %"
## [1] "----------year 6 -----------"
## [1] "tow == 5201 ---- 50 %"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 5301 ---- 51 %"
## [1] "tow == 5401 ---- 52 %"
## [1] "tow == 5501 ---- 53 %"
## [1] "tow == 5601 ---- 54 %"
## [1] "tow == 5701 ---- 55 %"
## [1] "tow == 5801 ---- 56 %"
## [1] "tow == 5901 ---- 57 %"
## [1] "tow == 6001 ---- 58 %"
## [1] "tow == 6101 ---- 59 %"
## [1] "tow == 6201 ---- 60 %"
## [1] "----------year 7 -----------"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 6301 ---- 61 %"
## [1] "tow == 6401 ---- 62 %"
## [1] "tow == 6501 ---- 63 %"
## [1] "tow == 6601 ---- 63 %"
## [1] "tow == 6701 ---- 64 %"
## [1] "tow == 6801 ---- 65 %"
## [1] "tow == 6901 ---- 66 %"
## [1] "tow == 7001 ---- 67 %"
## [1] "tow == 7101 ---- 68 %"
## [1] "tow == 7201 ---- 69 %"
## [1] "----------year 8 -----------"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 7301 ---- 70 %"
## [1] "tow == 7401 ---- 71 %"
## [1] "tow == 7501 ---- 72 %"
```

```
## [1] "tow == 7601 ---- 73 %"
## [1] "tow == 7701 ---- 74 %"
## [1] "tow == 7801 ---- 75 %"
## [1] "tow == 7901 ---- 76 %"
## [1] "tow == 8001 ---- 77 %"
## [1] "tow == 8101 ---- 78 %"
## [1] "tow == 8201 ---- 79 %"
## [1] "tow == 8301 ---- 80 %"
## [1] "----------year 9 -----------"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 8401 ---- 81 %"
## [1] "tow == 8501 ---- 82 %"
## [1] "tow == 8601 ---- 83 %"
## [1] "tow == 8701 ---- 84 %"
## [1] "tow == 8801 ---- 85 %"
## [1] "tow == 8901 ---- 86 %"
## [1] "tow == 9001 ---- 87 %"
## [1] "tow == 9101 ---- 88 %"
## [1] "tow == 9201 ---- 88 %"
## [1] "tow == 9301 ---- 89 %"
## [1] "----------year 10 -----------"
## [1] "Setting manually defined closures"
## [1] "Closures are yearly"
## [1] "tow == 9401 ---- 90 %"
## [1] "tow == 9501 ---- 91 %"
## [1] "tow == 9601 ---- 92 %"
## [1] "tow == 9701 ---- 93 %"
## [1] "tow == 9801 ---- 94 %"
## [1] "tow == 9901 ---- 95 %"
## [1] "tow == 10001 ---- 96 %"
## [1] "tow == 10101 ---- 97 %"
## [1] "tow == 10201 ---- 98 %"
## [1] "tow == 10301 ---- 99 %"
## [1] "time taken is : 17.89409 mins"
```

Summary plots

There are a series of input plotting functions to visualise the results of the simulation. For example, we can explore:

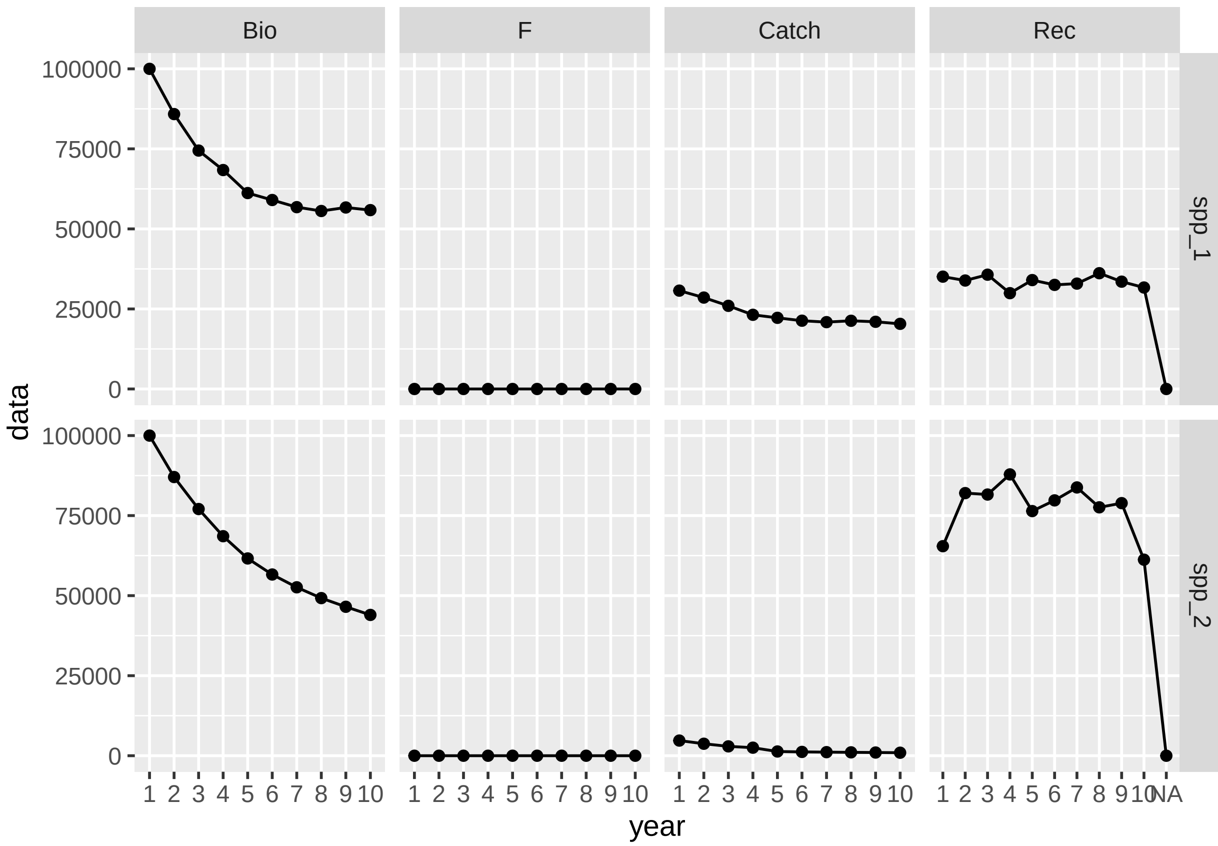
- the population dynamics for each species
- Seasonal patterns in exploitation
- the location choice of a vessel
- the realised step function for a vessel

Users will wish to define their own plots, depending on the issues of interest and all the results are saved in the output from the run_sim function.

```
## Biological
p1 <- plot_pop_summary(results = res, timestep = "annual", save = FALSE)
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = 1:11): invalid factor level, NA
## generated
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = 1:11): invalid factor level, NA
## generated

## Loading required package: ggplot2

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Warning: Factor `year` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

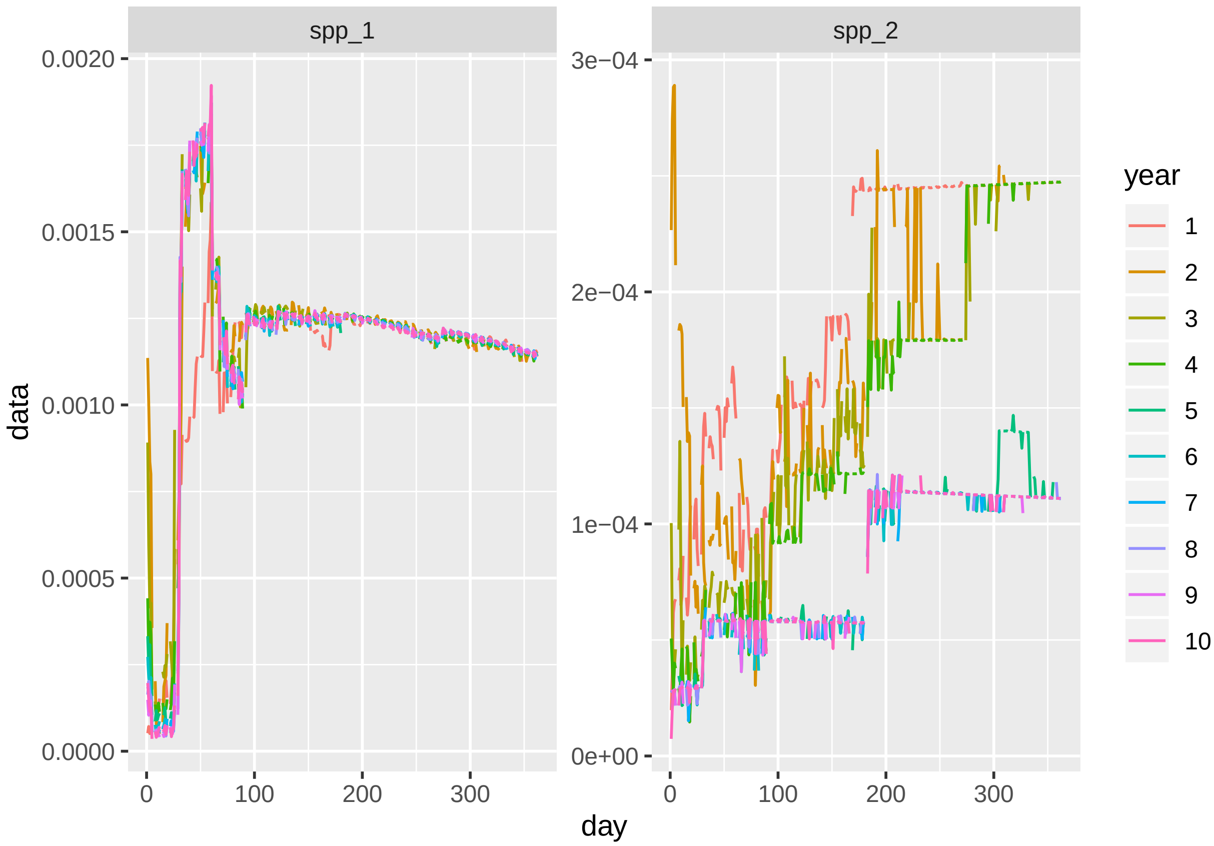



```
p1
```

```
## NULL
```

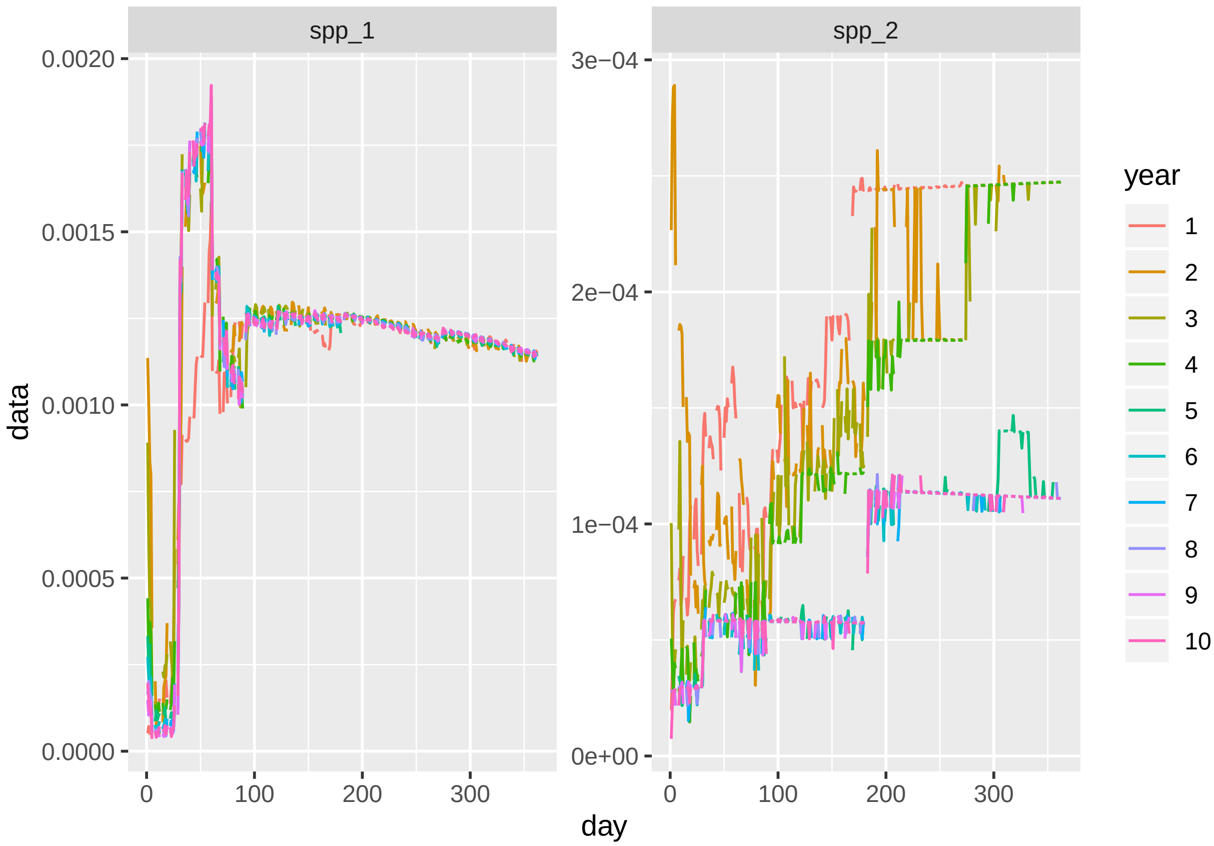
```
p2 <- plot_daily_fdyn(res)
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```

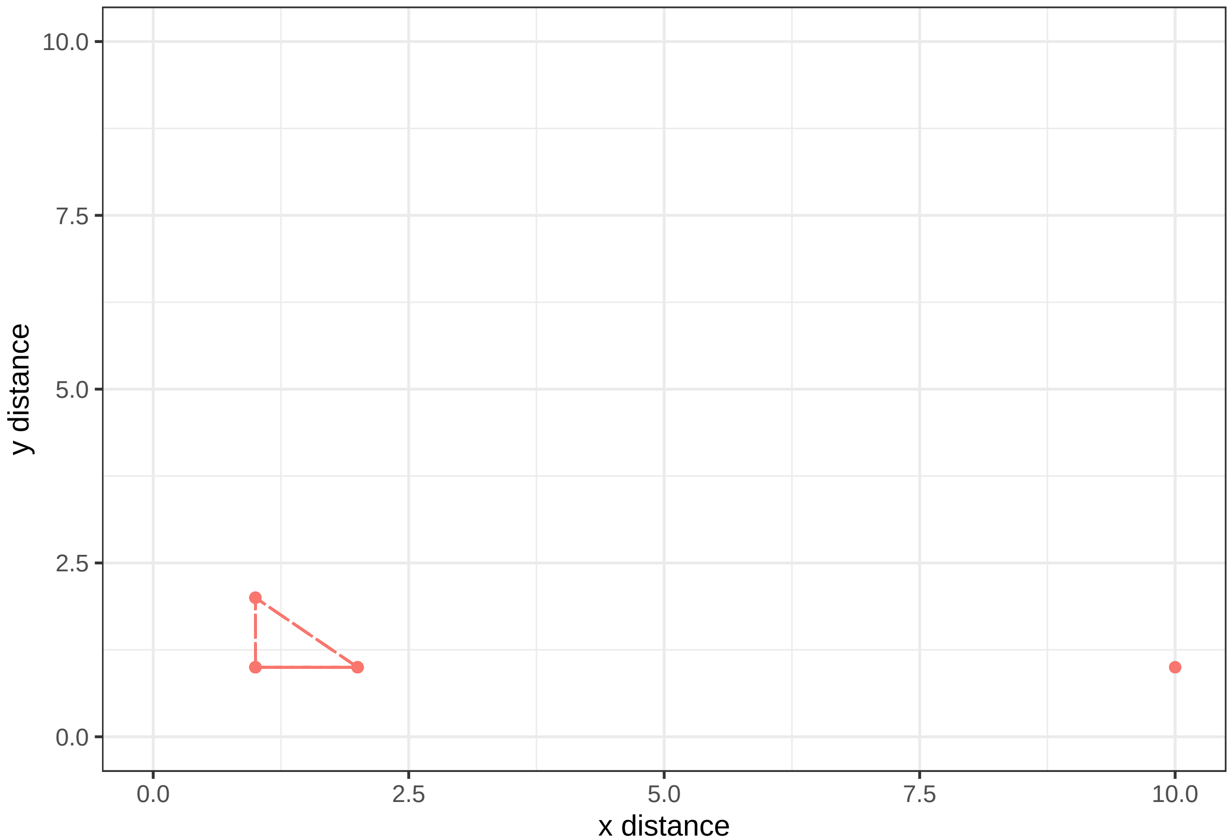



```
p2
```

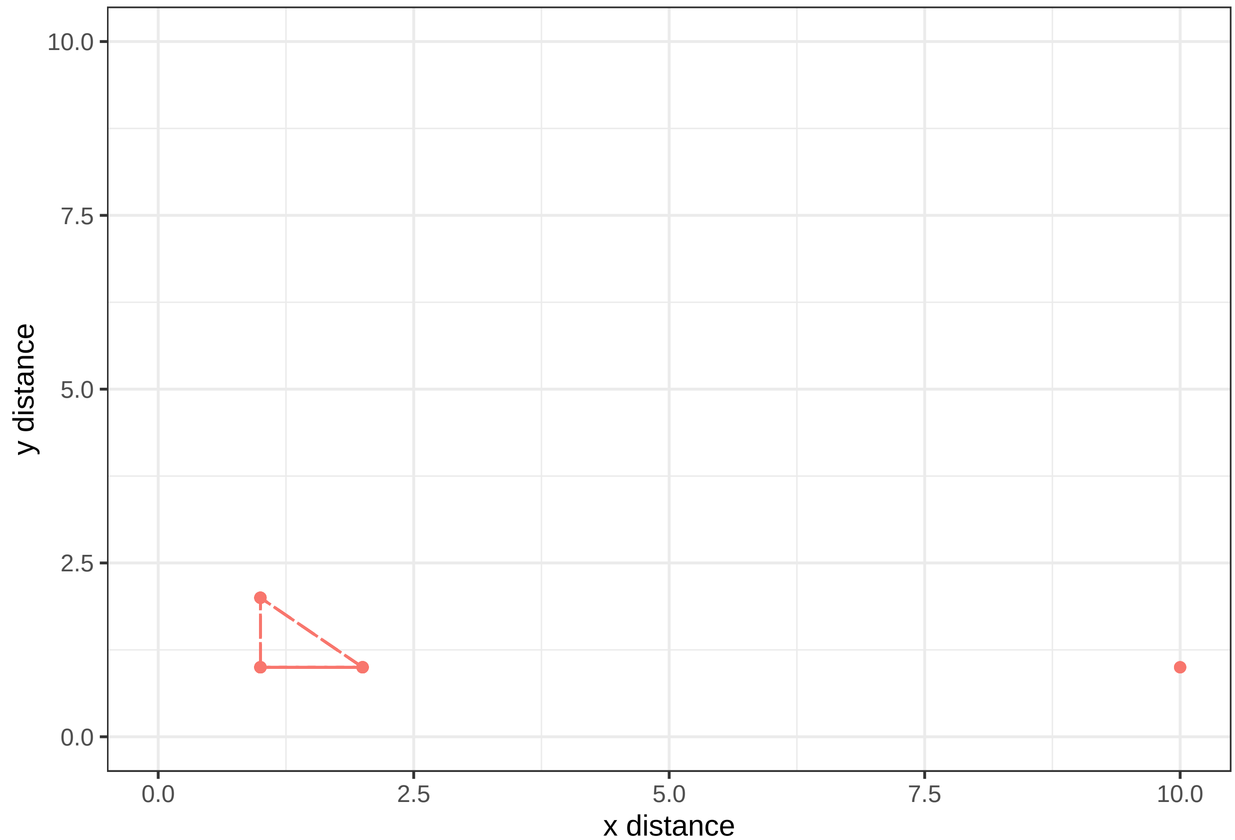
```
## Warning: Removed 1 rows containing missing values (geom_path).
```



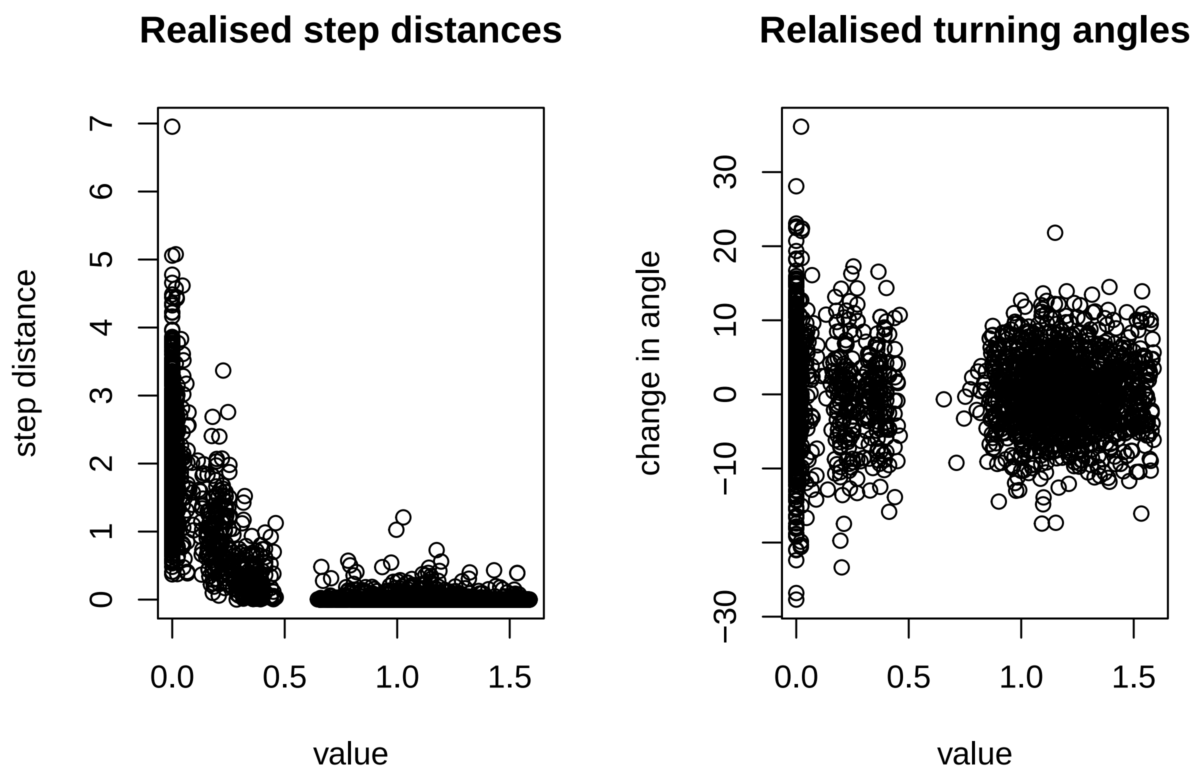

```
## Fishery

logs <- combine_logs(res[["fleets_catches"]])

p3 <- plot_vessel_move(sim_init = sim, logs = logs, fleet_no = 1, vessel_no = 5,
    year_trip = 5, trip_no = 10)
```




```
p3
```




```
p4 <- plot_realised_stepF(logs = logs, fleet_no = 1, vessel_no = 1)
```




```
## NULL
```

```
p4
```

```
## NULL
```

Note in our example how the fishing mortality rate for species 2 changes following the spatial closure, which was set to cover some of the core distribution of the population.