Simulate using SatTagSim

Benjamin Galuardi

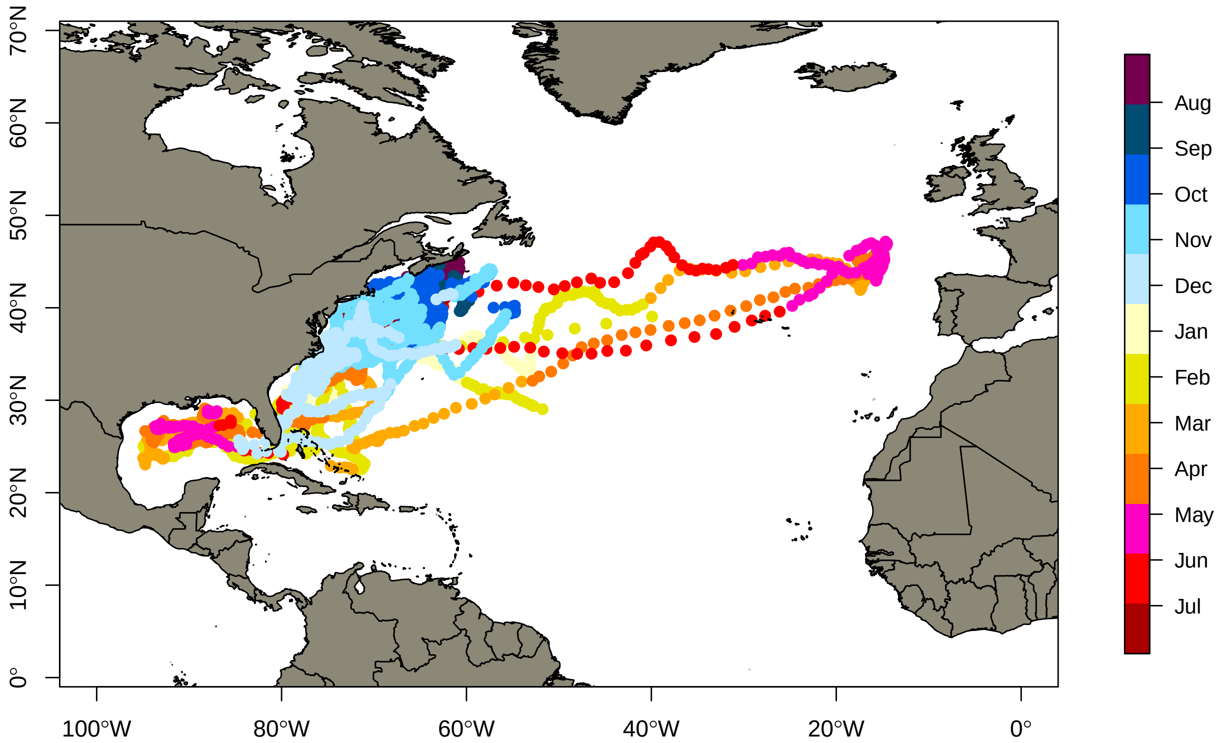
2017-06-14

Simulation of Nova Scotia tagged Atlantic bluefin tuna *(Thunnus thynnus)* and get seasonal transition matrices from a 7-box spatial strata.

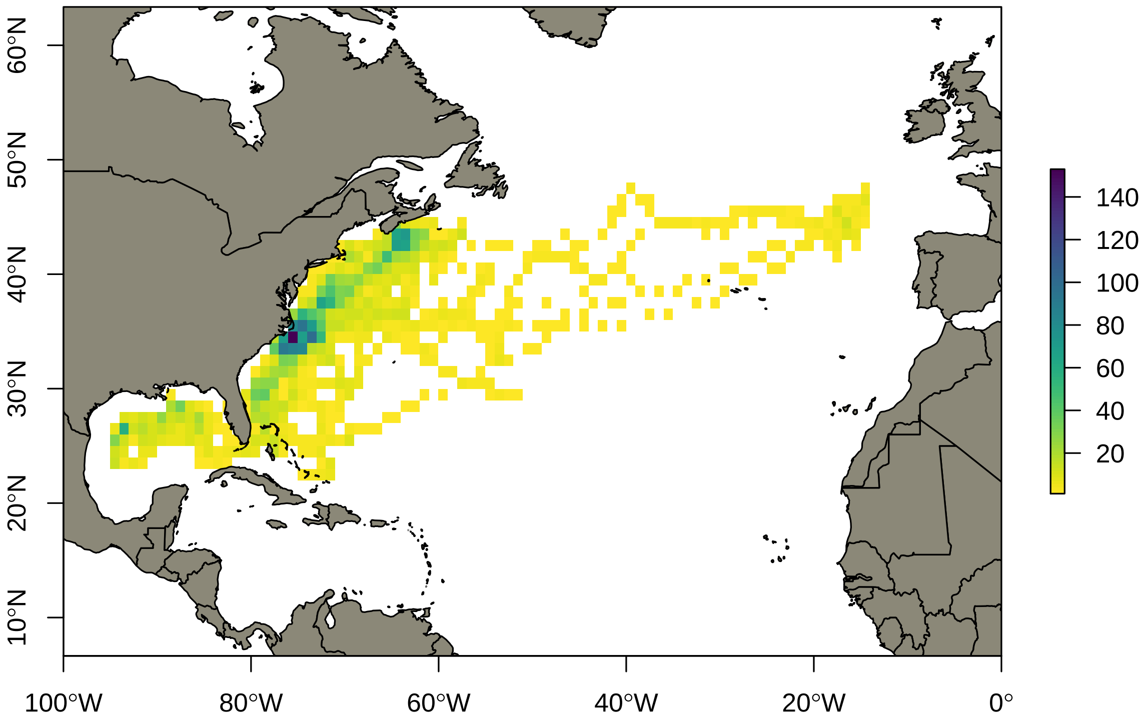
load background data and set up parallel

Tag data:

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
```


70°N
60°N
50°N
40°N
30°N
20°N
10°N
0°
100°W
80°W
60°W
40°W
20°W
0°
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul

plot as raster:



Spatial strata in this package

7-box strata from the Kerr et al. operational model

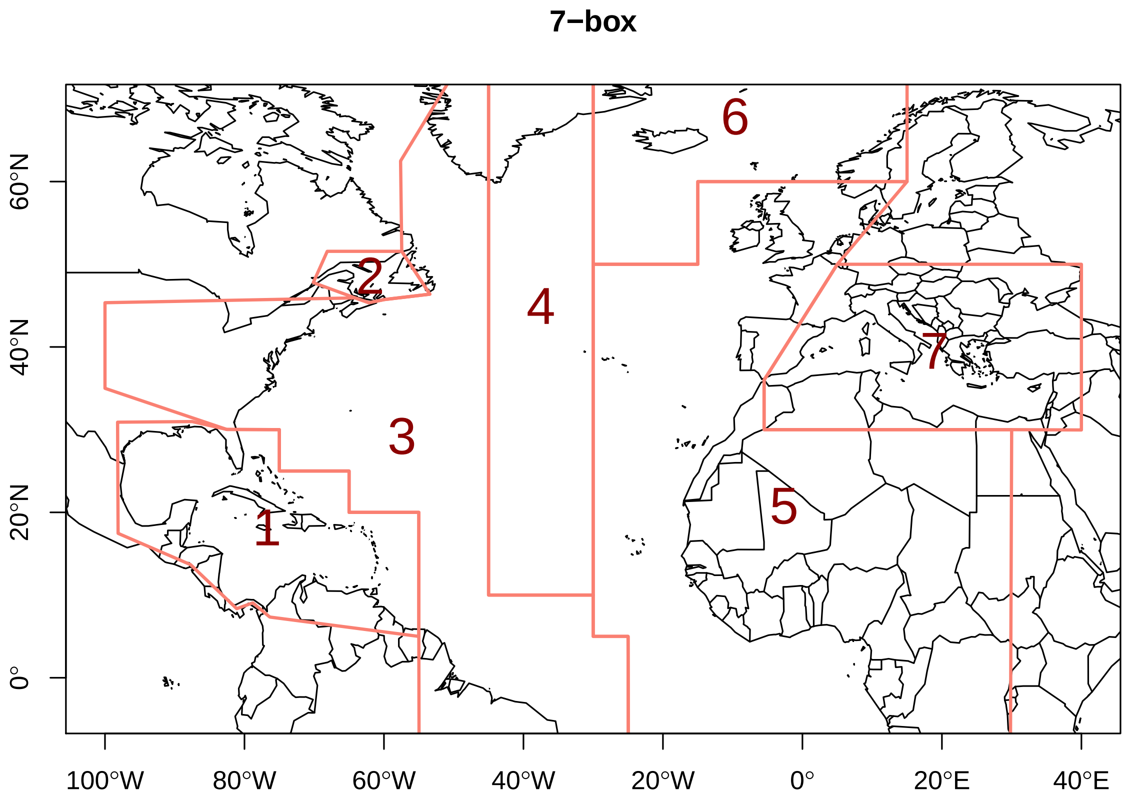


Figure 1: 7- box stratification from Kerr et al. (2016) operational model

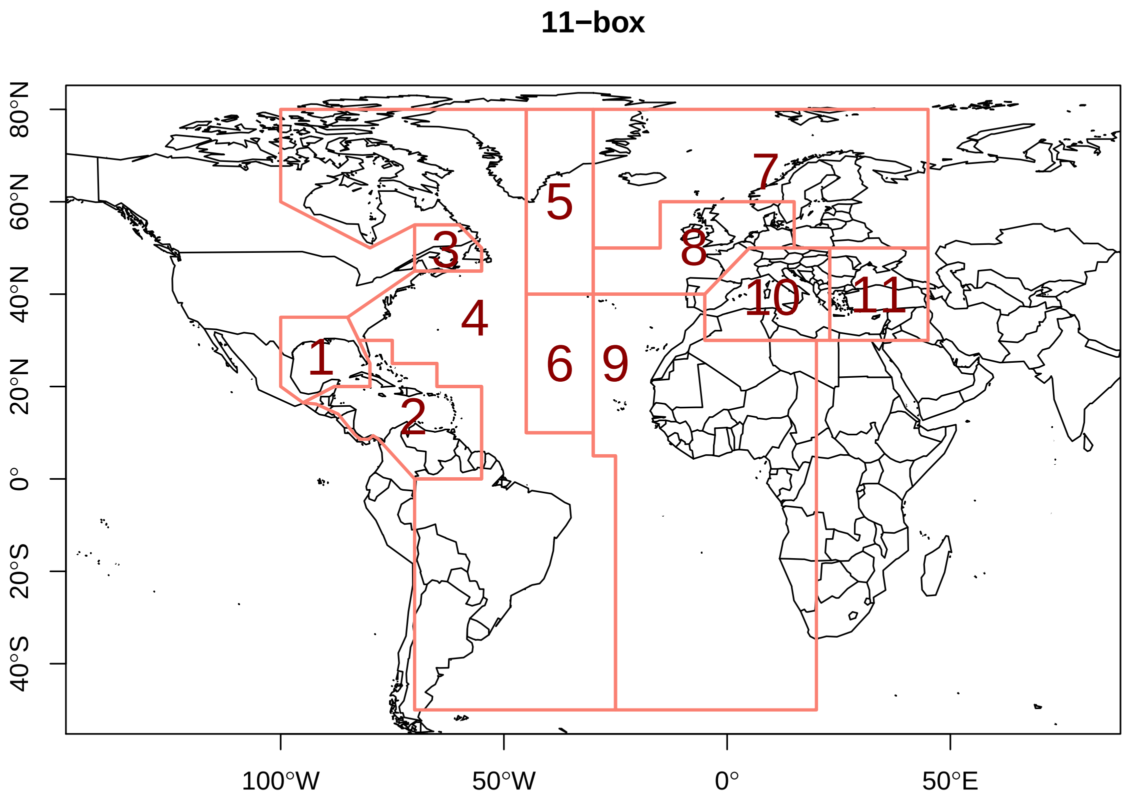


Figure 2: 11- box stratification from Lauretta et al. (2016) spatial summary

Environmental data in this package

World Ocean Atlas sea surface temperature is included. A raster mask, predicated on suitable temperature habitat in each month, is also included.

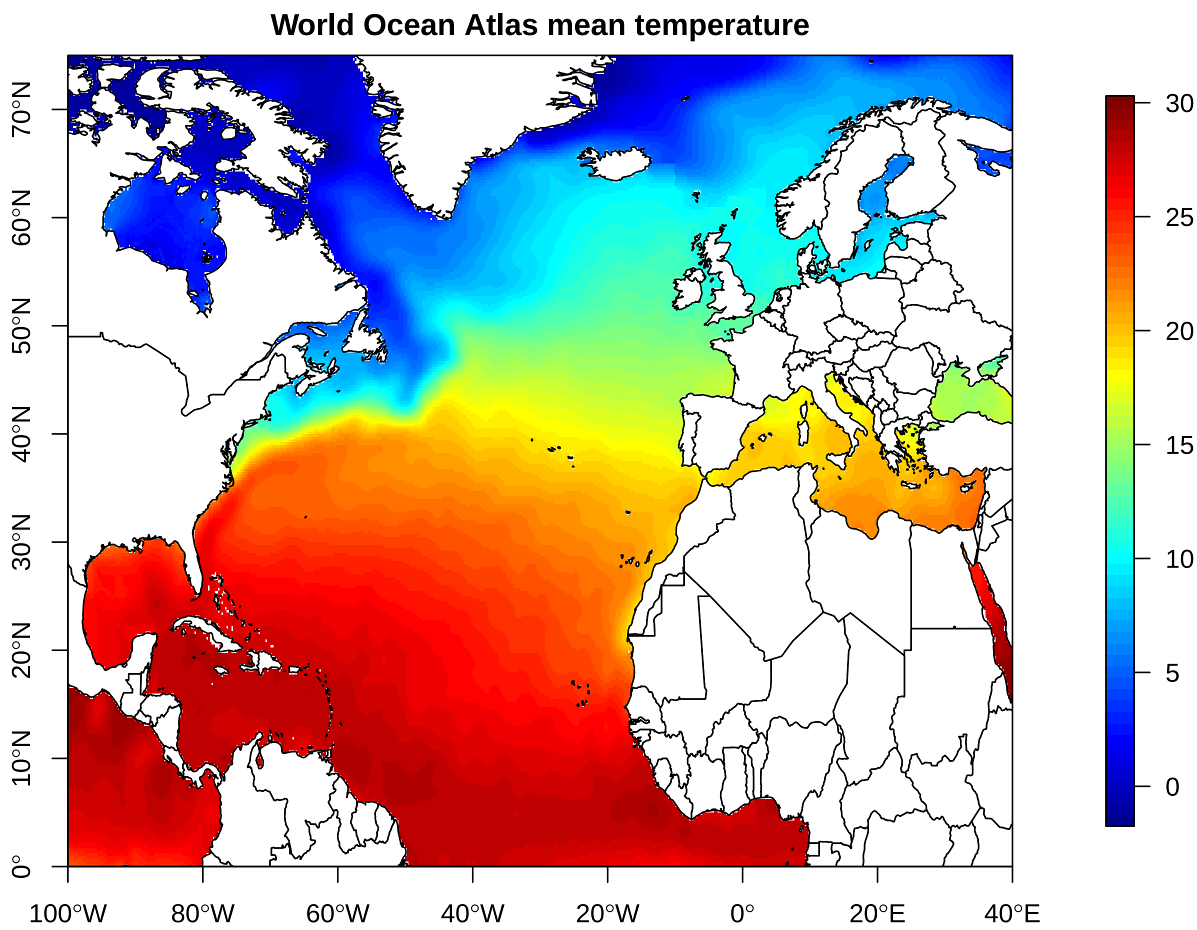
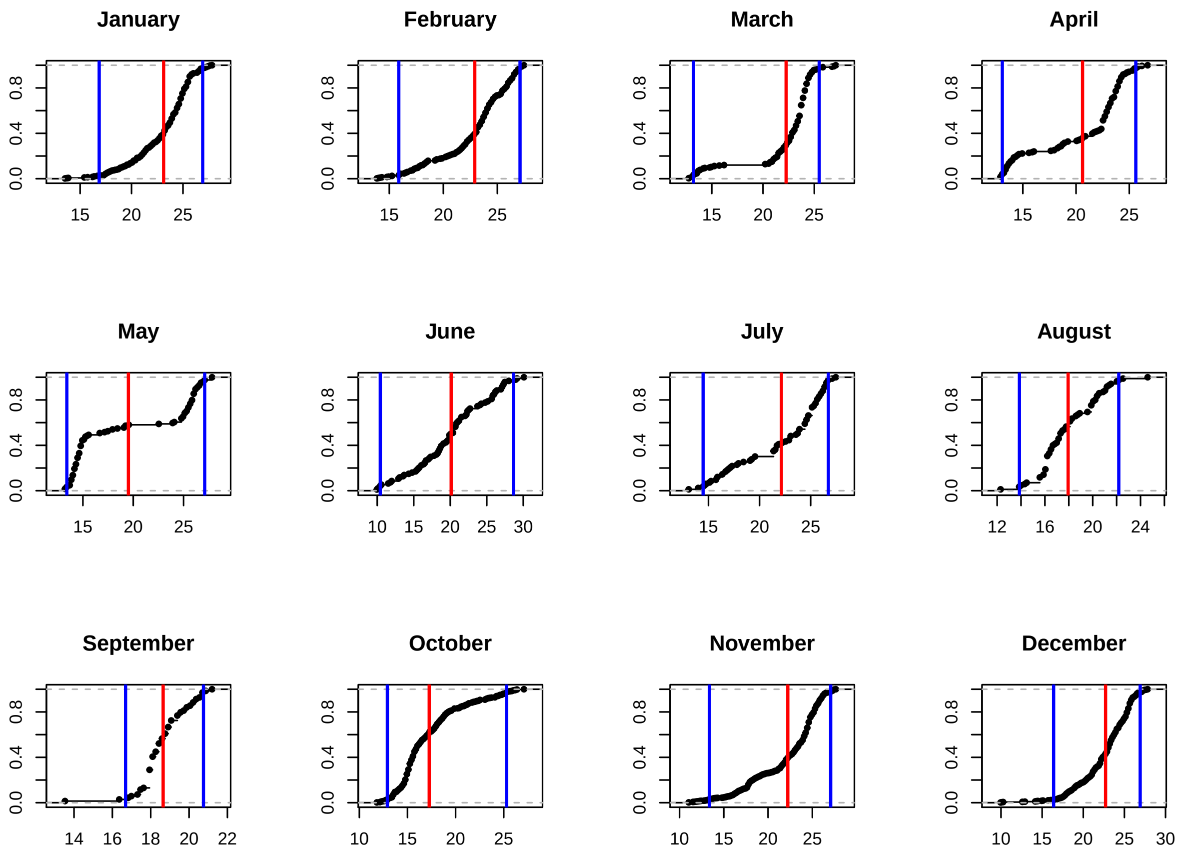


Figure 3: World Ocean Atlas (2013) mean temperature

Set up the temperature constraint field.

This can be done several ways. The preferred method is to use a running average of suitability, by month, for 12 months. In this manner, there is not a hard limit as to whether a fish may be in the area or not, but does provide a selection criteria that is weighted toward the suitability in that month.

The temperature field provided `data(rmask)` is a surface temperature suitability for bluefin tuna, based on tag measured temperatures.



Based on the plots, a range of $10 - 28\,^{\circ}\mathrm{C}$ were chosen

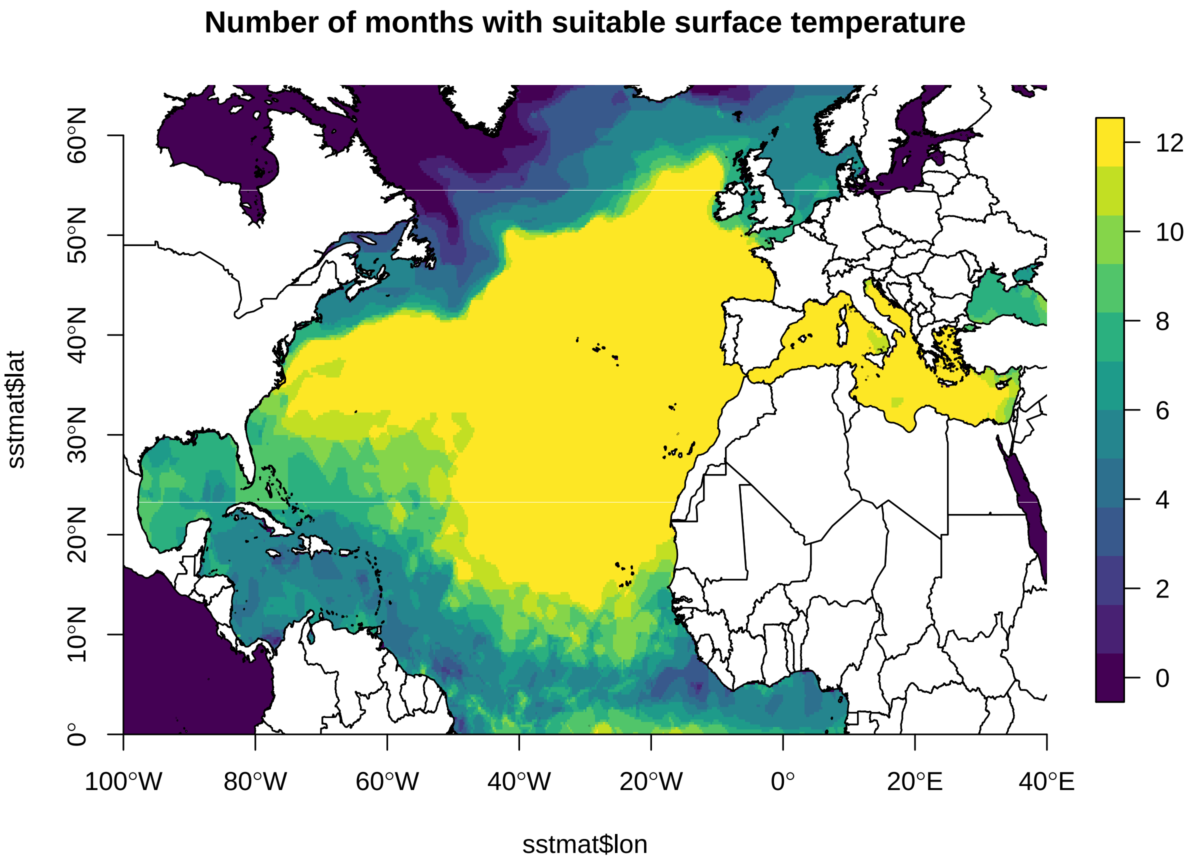


Figure 4: Number of months with suitable surface temperature

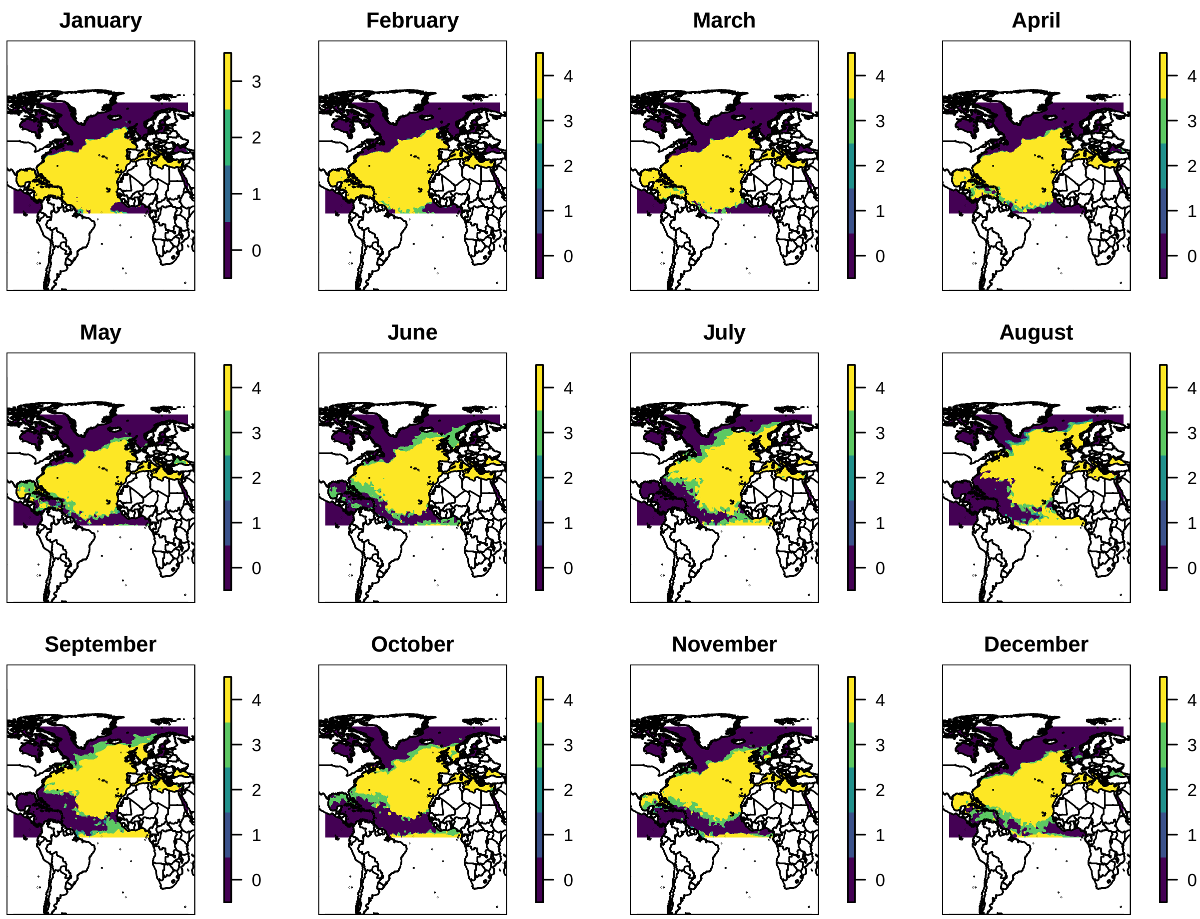


Figure 5: Number of months with suitable surface temperature. This plot shows a rolling sum of the current and next months binary values. This assists the simulation avoid entrainment in poor areas while accounting for where the fish might go next.

Get parameters from tag data

```
#----------------------------------------------------------------#
# add spatial overlay to tag data
nsfish@proj4string = box11@proj4string
nsfish$box = over(nsfish, box11)$ID
par2 = daply(as.data.frame(nsfish), c('box', 'TagID', 'Month'), function(x) get.uv(x[,c('Day','Month','

tagwts = daply(as.data.frame(nsfish), c('box', 'Month'), function(x) nrow(x))

ubox = apply(par2[,,,1], 3, rowMeans, na.rm=T)*-1
vbox = apply(par2[,,,2], 3, rowMeans, na.rm=T)

#----------------------------------------------------------------#

par.ns = get.allpar(as.data.frame(nsfish))
d.ns = get.kfD(as.data.frame(nsfish))
simpar = merge.par(par.ns, d.ns, return.mean = T)
```

```
simpar = make.par.array(tracks = nsfish, inbox = box11, rasbox = NULL, rrows = 26*5, rcols = 29*5, use_

# par.ns = get.allpar(as.data.frame(nsfish))
# d.ns = get.kfD(as.data.frame(nsfish))
# simpar = merge.par(par.ns, d.ns, return.mean = T)
```

Examine the tag based Advection and Diffusion parameters

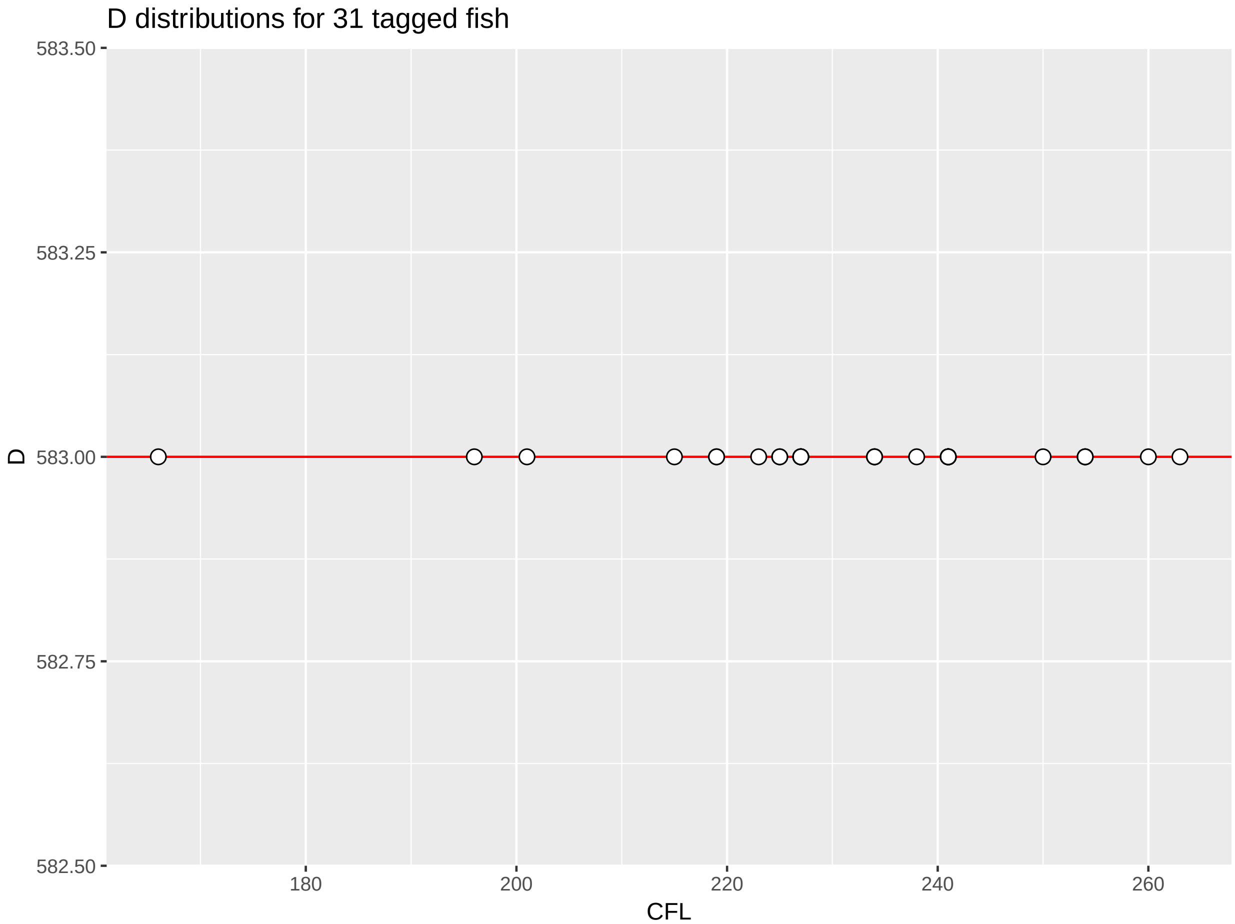


Figure 6: Distribution of diffusion (D) estimates by length (estimated CFL). All values here are identical as the track estimation routine used a fixed diffusion parameter. Here, D is measured as nautical $miles^2$/day

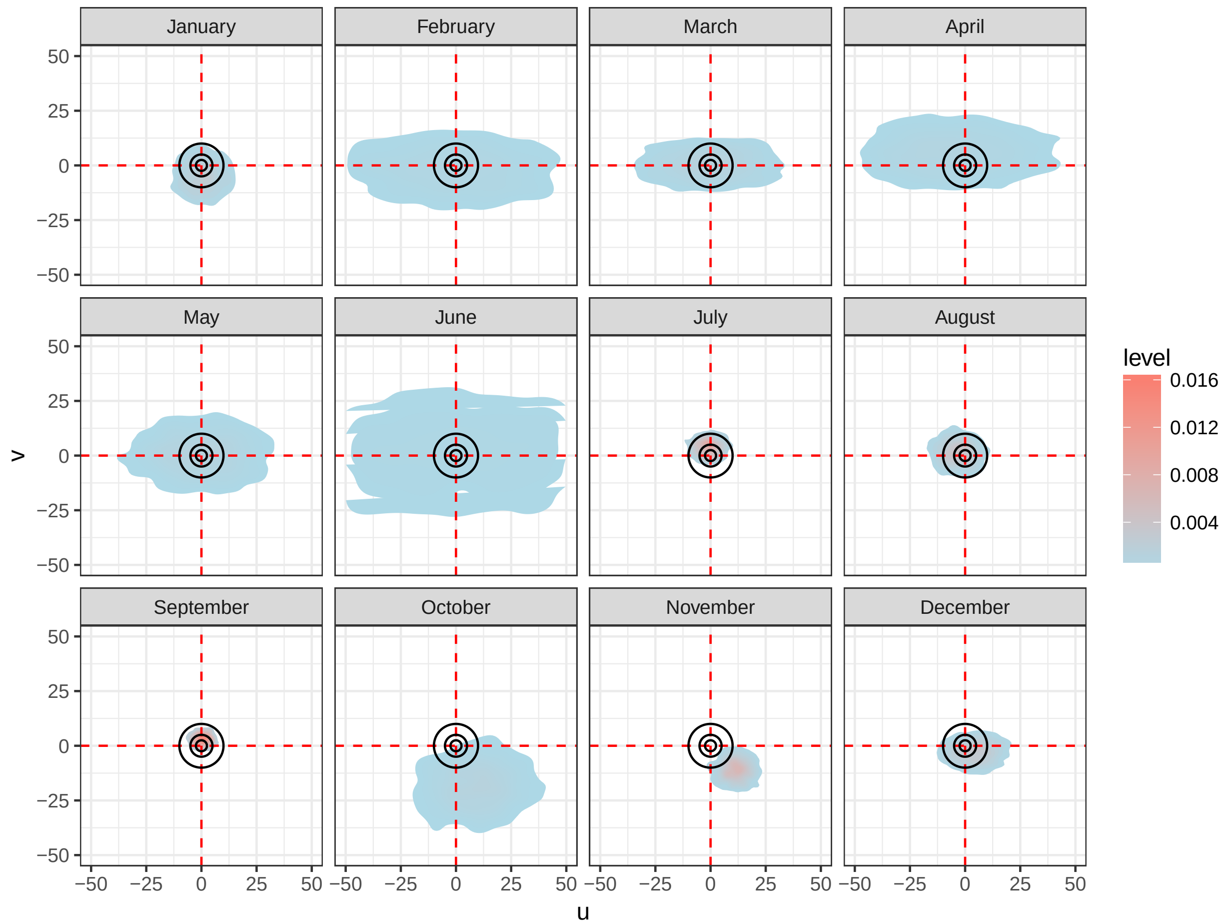


Figure 7: Distribution of monthly advection parameters. Wider distributions indicate greater variance. Units are nautical miles/day. This plot is not area specific but rather a general pattern of all fish within that month.

Setup simulation parameters

Table 1: Control parameters for simulation

sim_param	value	description
msims	50	simulations to be started in each month
npmon	30	number of simulation steps per month
nyears	2	number of years to simulate for each track
sstol	2	number of suitable months for each daily simulation step

Table 2: Month order for multi-month start simulation

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12
Jan	1	2	3	4	5	6	7	8	9	10	11	12
Feb	2	3	4	5	6	7	8	9	10	11	12	1
Mar	3	4	5	6	7	8	9	10	11	12	1	2
Apr	4	5	6	7	8	9	10	11	12	1	2	3
May	5	6	7	8	9	10	11	12	1	2	3	4
Jun	6	7	8	9	10	11	12	1	2	3	4	5
Jul	7	8	9	10	11	12	1	2	3	4	5	6
Aug	8	9	10	11	12	1	2	3	4	5	6	7
Sep	9	10	11	12	1	2	3	4	5	6	7	8
Oct	10	11	12	1	2	3	4	5	6	7	8	9
Nov	11	12	1	2	3	4	5	6	7	8	9	10
Dec	12	1	2	3	4	5	6	7	8	9	10	11

Set up starting points for multi-start month simulation

Set any indices for size or other criteria here. This may also be done prior to the parameter calculation step.

```
ns = as.data.frame(nsfish)
ns = ns[,c('TagID','Day','Month','Year','Longitude','Latitude')]
spts = get.start.pts(ns, msims, months = 1:12, posnames = c('Longitude','Latitude'))
str(spts)
```

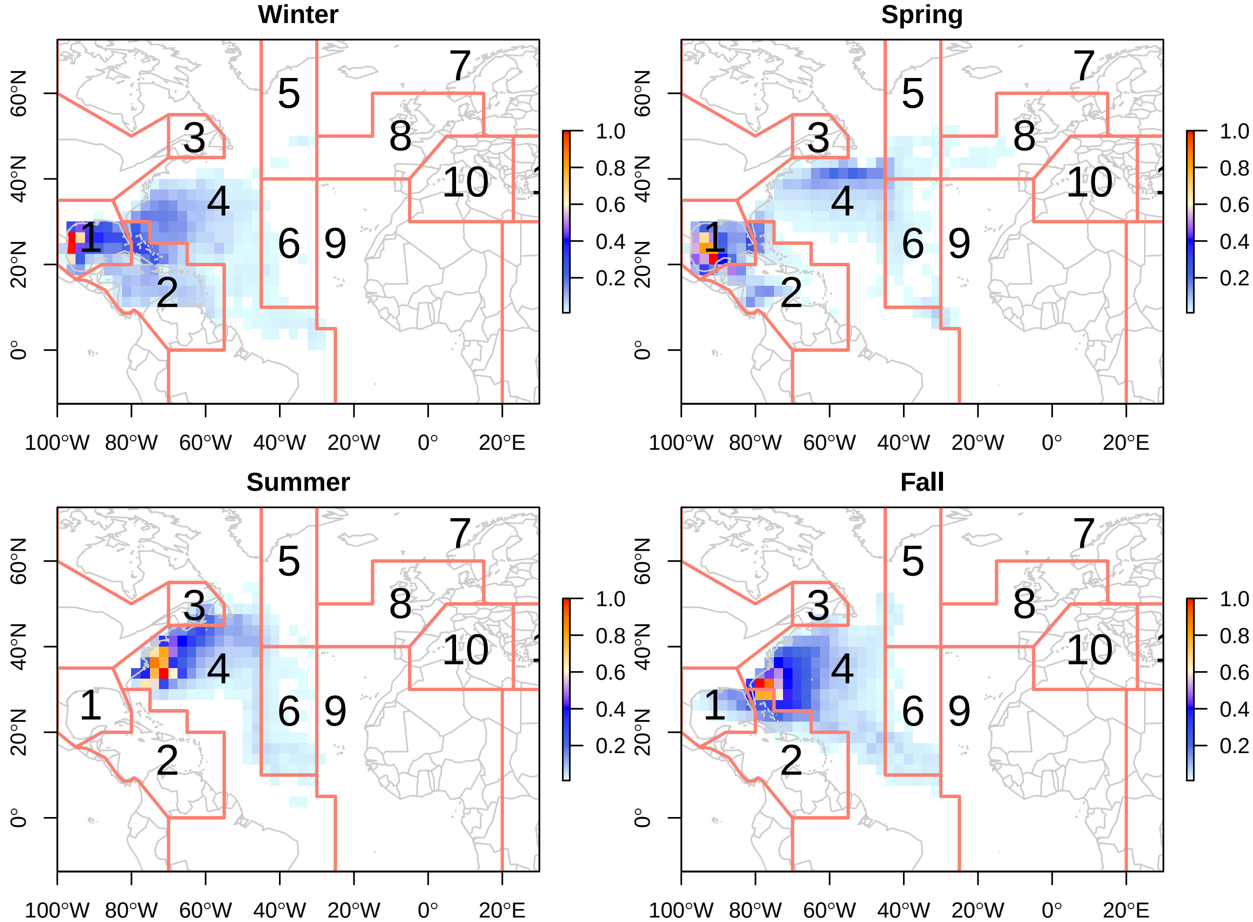
```
## List of 12
##  $ 1 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -72.4 -77.1 -85.4 -72.6 -60.4 ...
##   ..$ y: num [1:50] 37.9 33.1 26.4 37.4 35.9 ...
##  $ 2 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -75.4 -79.4 -77.4 -90.9 -57.6 ...
##   ..$ y: num [1:50] 29.6 30.1 24.1 24.4 30.6 ...
##  $ 3 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -86.1 -89.1 -89.6 -76.6 -72.9 ...
##   ..$ y: num [1:50] 27.6 27.6 26.4 30.6 25.4 ...
##  $ 4 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -71.6 -87.9 -89.6 -74.6 -17.4 ...
##   ..$ y: num [1:50] 34.1 28.1 26.4 33.1 44.9 ...
##  $ 5 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -30.9 -16.4 -72.1 -14.1 -18.9 ...
##   ..$ y: num [1:50] 44.9 46.1 38.4 45.1 45.4 ...
##  $ 6 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -63.4 -64.6 -62.4 -78.9 -65.6 ...
##   ..$ y: num [1:50] 41.9 42.4 41.1 29.6 38.6 ...
##  $ 7 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -61.6 -63.1 -70.9 -63.4 -68.6 ...
##   ..$ y: num [1:50] 43.4 42.1 39.1 43.4 38.9 ...
##  $ 8 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -63.6 -61.1 -63.4 -64.4 -63.6 ...
##   ..$ y: num [1:50] 44.1 44.1 42.4 43.4 43.4 ...
##  $ 9 :'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -62.6 -64.4 -66.9 -64.1 -68.9 ...
##   ..$ y: num [1:50] 42.4 43.1 40.9 43.9 41.9 ...
##  $ 10:'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -63.6 -68.4 -64.1 -64.6 -64.1 ...
##   ..$ y: num [1:50] 44.1 39.1 43.1 39.4 41.4 ...
##  $ 11:'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -74.9 -65.6 -75.6 -72.4 -71.6 ...
##   ..$ y: num [1:50] 33.9 33.9 33.1 34.1 36.4 ...
##  $ 12:'data.frame':  50 obs. of  2 variables:
##   ..$ x: num [1:50] -76.6 -61.6 -75.1 -72.6 -71.4 ...
##   ..$ y: num [1:50] 34.1 41.4 35.6 35.1 39.6 ...
```

```
# plot(map, xlim = c(-100, 0), ylim = c(20, 50))
# for(i in 1:12) points(spts[[i]], col = month.colors[which(as.numeric(month.colors[,1])==i),2], pch =
# degAxis(1)
# degAxis(2)
# box()
```

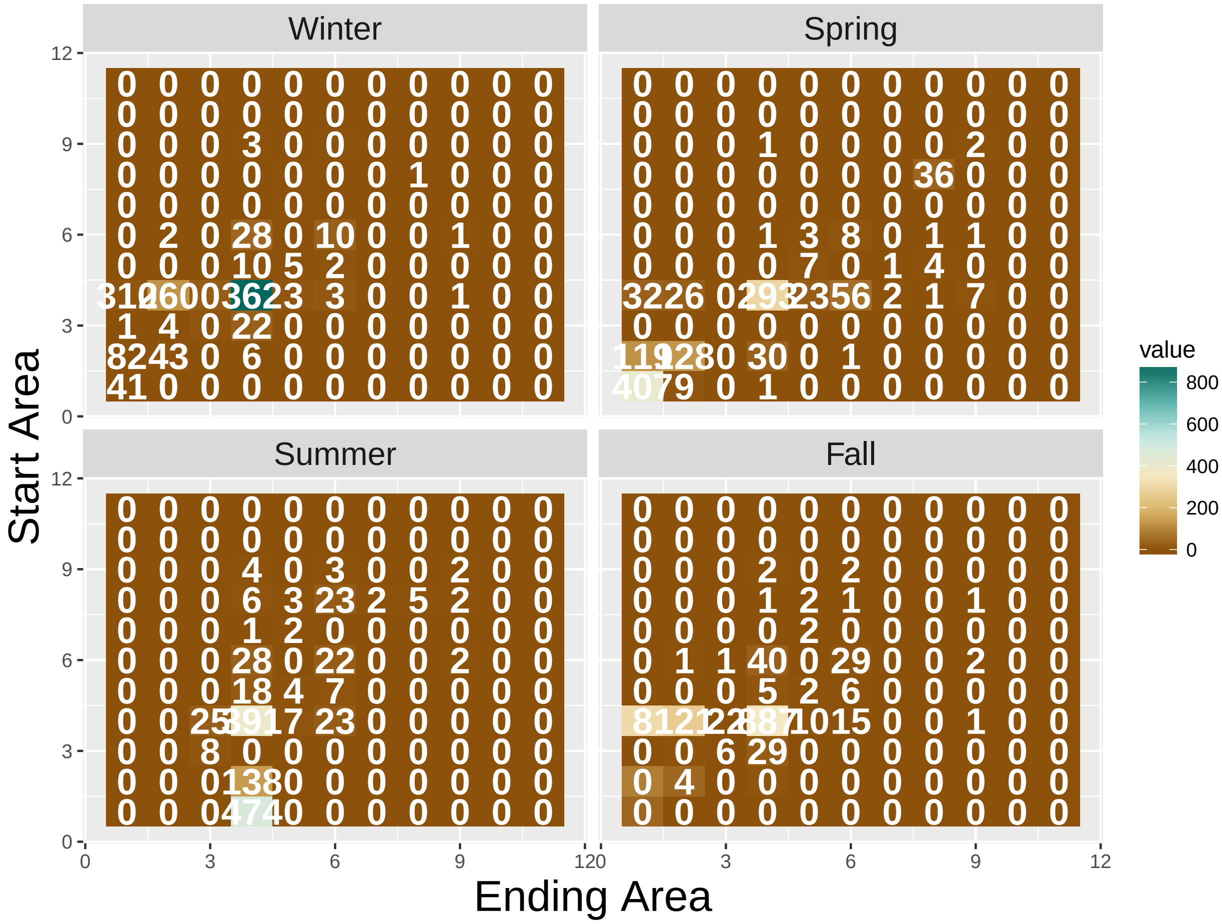
Run a single simulation in parallel

```
##
## Using 4 cores for parallel processing.
## [1] "simulating 600 tracks for 2 years"
## [1] "simulating 50 tracks starting in January"
## [1] "elapsed time: 35.2685260772705"
## [1] "simulating 50 tracks starting in February"
## [1] "elapsed time: 55.1445140838623"
## [1] "simulating 50 tracks starting in March"
## [1] "elapsed time: 1.23754040002823"
## [1] "simulating 50 tracks starting in April"
## [1] "elapsed time: 1.51465144952138"
## [1] "simulating 50 tracks starting in May"
## [1] "elapsed time: 1.81021433273951"
## [1] "simulating 50 tracks starting in June"
## [1] "elapsed time: 2.10657730102539"
## [1] "simulating 50 tracks starting in July"
## [1] "elapsed time: 2.39070571660996"
## [1] "simulating 50 tracks starting in August"
## [1] "elapsed time: 2.68825213114421"
## [1] "simulating 50 tracks starting in September"
## [1] "elapsed time: 2.97783108154933"
## [1] "simulating 50 tracks starting in October"
## [1] "elapsed time: 3.29841313362122"
## [1] "simulating 50 tracks starting in November"
## [1] "elapsed time: 3.59384268124898"
## [1] "simulating 50 tracks starting in December"
## [1] "elapsed time: 3.86995361646016"
```

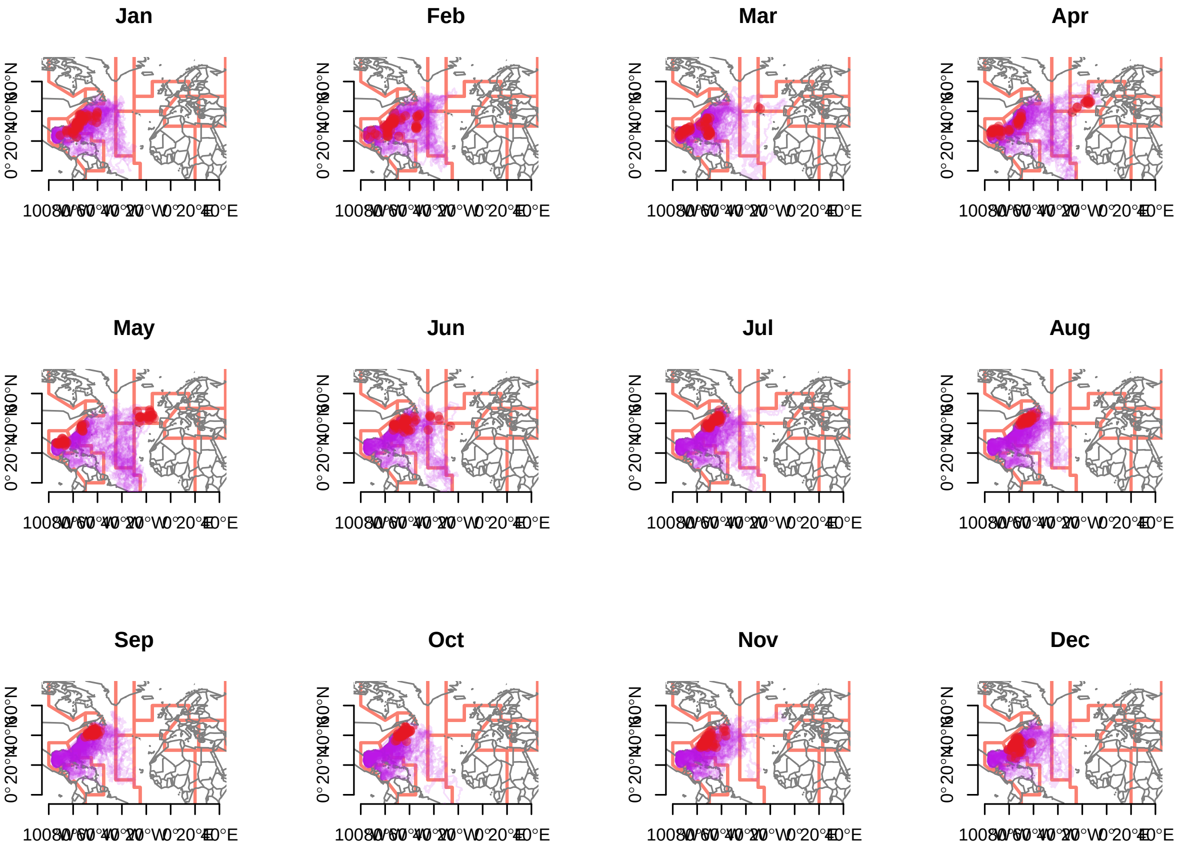
Make a dataframe and raster of results



Get seasonal transition matrices using the 11 box model



Plot simulation results by starting month



plot raster of results by month