Write-up/pseudocode of function for shifting CA DC fishing effort: effort\_mgmt() in Mgmt\_scenarios\_effort\_shift.R)

Sam Woodman, April 2020

**General**

* The minimum season start date is defined Nov 15 for region “CenCA”, and Dec 1 for all other regions. So this function is applicable to California only.
  + NOTE: There are 1797 total records (of 129282) that have dates before their region’s minimum season start date. There is a function input that allows the user to remove this effort or pile it into 1st day of season
  + This happens when making ‘x.fish.pre’ object in code
* The regional end dates are July 15 for region “CenCA”, and July 31 for all other regions

**Delayed opening**

* Filter for the region to be delayed (other data will be added back in at the end)
* For each crab fishing season + region, determine/calculate:
  + 1) Original opening (season\_date\_st in code): the date of the season opening (the first day with data using original data). Might change to first day with >=1% of pings/$/lbs for season by district, treat same was pre-Nov 15 fishing activity (option to drop it or pile it into 1st day of season)
  + 2) Original closing: the date of the season closing (the last day with data using original data)

3) Management opening (season\_open\_mgmt in code): the date the season would open under the provided management scenario

* + 4) “season\_days\_delayed”: (3) – (1), i.e. the number of days that the season opening is delayed in that region under the provided management scenario
* Join the filtered fishing data and ^, by region and crab season
* Shift fishing data dates
  + If the season was delayed (i.e. season\_days\_delayed > 0) for that region/crab season:
    - Lag: shift all records for that region/crab season forward in time by season\_days\_delayed
    - Pile: shift records that came before the provided start date forward in time by season\_days\_delayed
* Redistribute effort using:
  + Temporal fidelity: shifted forward in time based on historical spatial distribution of effort (= calendar month temporal fidelity)
    - Get our 3 ‘types’ of effort: 1) effort to redistribute with ‘base values’ (see below), 2) effort to redistribute without ‘base values’, and 3) effort that does not need to be redistributed (e.g. effort that occurred after the management start date when using the “pile” method)
      * Base values are the data needed to inform the redistribution of effort. For instance, if there is effort being moved from November to December, the data in December that will inform the redistribution of the shifted November values are the ‘base values’.
    - Next, use the base values to get the percent of the redistributed effort that should go in each grid cell as follows. Specifically, divide the effort values of the variable we’re using to calculate the redistribution percentages (i.e. var\_perc, currently hard-coded to be Num\_DCRB\_VMS\_pings), by the sum of var\_perc (grouped by region/crab season).
      * Base values are identified by filtering for records that came after the provided start date.
      * For year-month-regions that data has been shifted into but that do not have base values, the data is not redistributed spatially
    - Final steps: Combine our three ‘types’ of effort, and group/sum by crab\_year, GRID5KM\_ID, Region, year\_month, and date\_record.
    - NOTE: because we are working with daily effort values but redistributing based on month, effort that is redistributed according to both the spatial and temporal patterns of the base values, within each year-month. This could be avoided by determining base values/redistributing by day, but this seems like it could raise more problems. Also, this method and the current one will yield equivalent results once the effort are aggregated to a monthly level.
  + Spatial fidelity – shifted forward in time but stays at same location (= crab season temporal fidelity)
    - The records that are shifted forward stay in their grid cells, i.e. maintain spatial fidelity. No additional work is needed
* Final steps
  + We check that none of the records were shifted into a new crab season
  + ‘Add back in’ effort that was not delayed (the effort that was filtered out at the start of this section)

**Early closure**

* For early closures, we assume that if all Regions are closed, the effort is simply removed and not redistributed backwards in time. If only certain regions are closed, e.g. BIAs or Central CA, then the effort from those areas may be either removed or redistributed to regions that remained open.
* For each crab fishing season + region, determine the management closing date, i.e. the date the season would close under the provided management scenario. Then use this information to add a logical column indicating if each fishing record came after the management closing date (and thus will either be deleted or shifted in space)
* The closing date is applied across one of the following regions: BIAs, CenCA, NorCA, or All regions
* The effort in the specified region(s) that come after the management closing date are processed using one of the following methods:
  + Remove
    - The identified effort is simply removed
    - This method is required if the region specified is “All”
  + Temporal fidelity. We also provide a scalar that controls the percentage of the effort affected by the closure that will be redistributed.
    - All: NA - the remove method is required
    - BIA: Redistribute closed effort), reduced by a user-provided scalar in case we only want to redistribute a percentage of historical effort, to its respective region (i.e. northern/central CA, whichever that record is in) using the pile up + temporal fidelity redistribution method (described above
    - CenCA or NorCA: Redistribute effort to the other (i.e., to the open) CA region using the pile up + temporal fidelity redistribution method (described above)

**Other**

* With these delayed opening redistribution methods, particularly a delayed opening with a lag shift, records may be shifted outside 1) the window of the observed fishing season and 2) the legal window of the fishing season. Thus, we add 1) column “date\_past\_season\_end”, a logical indicating whether the record date is after the original season end date (the last day with recorded data) and 2) column date\_past\_region\_end, a logical indicating whether the record date is after the region end date (July 15 for CenCA and July 31 otherwise)
* Identifying variables and whale prediction information is joined back to the shifted fishing data
  + Outstanding question/issue: the shifted fishing data may (is likely to?) have new dates that were not in the original input data, and these records will have NA identifying variables and whale prediction information in the final output. Thus, should users have to join the identifying info and whale predictions themselves after the fact? Or should we make another function for this.? But if the final output from this function is aggregated by year-month, we should have comparable information in the historical record and the simulated scenarios.
* There are some NorCA DC fishing seasons (e.g. 2011-12 and 2017-18) where there are a couple of records on ~Dec 5 or 6, and then nothing until mid-January (there appears to have been closures due to quality these years). Because these ~Dec 5/6 records are possible under ‘normal’ opening dates, these records are used when pushing the season back, etc, which will lead to incorrect information. For instance, say we’re running a lag delayed opening scenario until Dec 15, there are 10 records on Dec 5, and the rest of the records come after Jan 15. Currently, the function will see that the first day of the season was Dec 5 and thus lag the entire season by 10 days, which (I assume) is an error. See discussion above, Might change to first day with >=1% of pings/$/lbs for season by district, treat same was pre-Nov 15 fishing activity (option to drop it or pile it into 1st day of season)

**Notes from chat on 042820**

Line 20: shouldn’t this go beneath line 25, closure.date? yes

closure.method: what does temporal fidelity mean here? similar to delay.method.fidelity. redist\_temporal() automatically projects effort spatially in a way that has fidelity to that calendar week/month, if the spatial dist of effort remains the same no need. see lines 290…

Some rejiggering needed (new arguments defined), but we can implement xx% reduction in effort for spring or Dec or whatever

User sets output df name, in example it is called d.noinfo

Lines 52-66: basic checks. i learned a lot

Plan to use make\_scenarios\_table.R and then wrapper to loop through scenarios? yes

minimum season start date != opening date. but we could modify either one to reflect when pots first get set (eg, 3-7d before landings are >=1%)

**Questions from testing 043020**:

1. What are you thinking is the best fix for the issue with CA\_OFFSHOR? I found that if I use x.orig (df that includes join with whale model outputs) the effort\_mgmt() function breaks. Using x.orig.noinfo is not problematic.
   1. Warning in effort\_mgmt(x = **x.orig.noinfo.** Error message on line 89: the function wants to have whale predictions and is not getting them from the input file. What gives? Why not feed the function x.orig rather than x.orig.noinfo? Is this due to the issue with CA\_OFFSHOR?
   2. Warning in effort\_mgmt(x = **x.orig** because some grid cells are in two offshore regions
2. Do you join the whale model outputs to the output df from the effort\_mgmt() function in the risk\_mgmt() function?
3. For the output df from the effort\_mgmt() function, are all of the temporal and spatial related column headers the simulated dates and locations, so to speak? So they reflect the date we are pretending the fishing activity occurred on in the place we are pretending it happened, given the input arguments?
4. Pesky Q: shouldn’t we specify in lines 13-29 that we have an option for NULL for the arguments delay.method.shift and delay.method.fidelity when we are not simulating a delayed opening and for closure.method and closure.redist.percent when we are not simulating an early closure? For the delayed opening scenario I used I just specified “lag” and “spatial” and believe they get ignored.