Update Log - Oklahoma Fishery Analysis Application

Main App

1. 9/3/18 - Dray
   1. Problem: catch curve was incorrectly calculating mortality. I had calculated the natural log of the catch before putting data into catchCurve() function (FSA). Turns out catchCurve() calculates natural log on its own too…so essentially, it was taking a log of a log…no wonder why they were always such low mortality rates.
   2. Fixed: I reworked code to account for this. Just removed step where natural log of frequency was previously calculated and renamed a few things in both the catch curve plot function and the mortality table function.
2. 9/3/18 – Dray
   1. Problem: Age database needed updated gear codes – also went through age data validation code
   2. Fixed: compiledagedata.csv was updated with new gear codes
   3. Note: All fields in compiledagedata.csv should comply with data validation rules
      1. Exception: Day field – a few records (very small handful) had “.” for Day. I deemed this okay, but Validation App makes it a required field from now on.
3. 9/30/18 – Dray
   1. Problem: Main app wouldn’t work with imported data that had periods in the TL\_mm or Wt\_g fields (R reads this field as a factor).
   2. Fixed: Wrote functions that change any periods to NA, then changes TL\_mm and Wt\_g fields to numeric.
4. 11/15/18 – Dray
   1. Problem: Change simple CV to RSE in CPUE tables. Also change calculation method for N RSE(25) and RSE(40)
   2. Fixed: What ODWC actually needed was RSE. Still not a completely correct method, but better than nothing. Methods for calculating N RSE(25) and N RSE(40) were incorrect. Changed to method used by Dumont and Schlechte (2004)… N = (CV/RSE)2
5. 11/15/18 – Dray
   1. Problem: Reference R2 did not account for “weighted” regression. Ogle’s catchcurve function does this automatically as a TRUE/FALSE argument, so I didn’t know how to do this specifically. R2 value isn’t included in Ogle’s functions, so when calculating it, I had to do the weighted regression manually as well.
   2. Fixed: Figured out how to do this…now calculates weights in first linear model and runs a second linear model with the weights.
6. 11/15/18 – Dray
   1. Problem: Including sex-specificity for Paddlefish
   2. Fixed: modified species codes in speciesinfo.csv…320 for Paddlefish (all), 320.1 for Paddlefish (male), 320.2 for Paddlefish (female). Also modified the WSnames.csv (links ODWC species codes with species names needed for relative weight calculations in FSA functions) to account for new sex-specific codes (each sex has different relative weight calculation, including one for an overall paddlefish)…Brown and Murphy 1993. Also updated gabelhousenames.csv to include all 3 paddlefish spp codes (used in FSA functions to reference PSD groupings for species).
7. 11/15/18 – Dray
   1. Fixed: Updated Total Effort Table to use Gear.Length (minutes) for electrofishing samples (consistent with what is used for CPUE). Also fixed the error produced when uploading an independent sample (one of the functions was referencing a selectize input from the first tab.
8. 3/2/2020 - Dan
   1. Fixed: update to Shiny package broke renderDataTable function.  Research into this indicates this function is being depreciated and the same function from the DT package is to be used instead. I added DT package to the app and changed the code to specifically call DT::renderDataTable to make tables.
9. 1/17/2021 – Dan
   1. Problem: CPUE by PSD size class often add to more than the total CPUE…typically when there were sites with Species.Code=98 (no fish in sample). Upon investigating, we originally had deleted any row that did not have a Gabelhouse.Name. This means that any sample that either had no fish, or only caught fish that do not have PSD size classes defined would be removed, so when the addZeroCatch function (which is being depreciated) was run, the sample was not there to receive zeros. This was necessary as if you run psdAdd() on spp with missing species names, there was a bug that added additional rows and made it so the list of PSD names and the original file do not have the same number of rows (causes cbind or mutate to throw an error due to mismatch).
   2. Solution: I contacted Derek Ogle about the bug and he quickly rebuilt psdAdd so it no longer had this behavior. I then rebuilt the code using a dplyr approach and waited to throw out the spp that do not have PSD size classes until after using complete() (which is the preferred approach over addZeroCatch). I also added logic to find fish where no TL data was present and deleted these also so they do not appear in the table (they may have a total CPUE, but if no TL’s were taken, can’t meaningfully express this in PSD-based CPUEs). I also allowed trophy size to be in the table. The sizes sort fine using dplyr verbs instead of the old approach.
10. 1/17/2021 - Dan
    1. Made several other small changes:
       1. Started replacing plyr’s join with dplyr left\_join.  However, there are places where we intentionally made gear code or species code character rather than numeric and left\_join checks this before running and throws an error so these are left.  We also have 2 instances of join\_all that I’m not sure has a dplyr counterpart, but I’ll need to research.  I’d like to get us off of plyr entirely some day if we can as loading order is an issue that causes lots of problems (especially if we want to load dplyr in ui.r, which is not done now...would also have to load plyr and do so first to avoid plyr overwriting dplyr.
       2. Changed renderTable function for PSD size definitions (both mm and inch) so NA displays as blank space in table.
11. 10/18/2021 Dan modified lines that created “w” as length category width for lencat() function as the original code could not deal with TL/30 = exactly 20 or exactly 15 (roughly lines 1154-1162 and again on 1181-1189). Found in response to bug report by Mike Hollie.
12. 10/19/2021 Deployed new data Ashley had sent me last month…several sampling files were excluded due to not being properly formatted/validated. All age data were able to be fixed and used (but loaded some in a separate run after realizing the only problem they had was including a couple blank rows at the bottom of the file.
13. 11/28/2021-5/19/2022 (deployed to shiny server 5/19/22).
    1. Rolled out the percentile tab (after a couple months of beta testing by a group in ODWC)
       1. Included percentile data is very selective and does not include records with heavy amounts of duplication…throws away about 10% of the available data, but the duplication issue could easily double or triple CPUE values if it is true duplication, so this is better than including.
    2. Updated to Shiny v 1.7.1, which broke percentile tab’s statements that check to see if first row of output tables are all NA values (MARGIN = 1, FUN = function(x) all(is.na(x)))==F). I rewrote these so they now check if the number of rows with NA in Species Name column is < the total number of rows (i.e., they are not all NA).
    3. Developed modal dialog to allow user to pick how to use the app (start with last 5 years, last 10 year, all years, or user upload data) to speed startup time in cases where full database is not needed.
    4. Rewrote code to cross-populate selections across selectize box pairs.
       1. After trying lots of options, settled on getting rid of the cross-talking code and name boxes. Instead, I add a check box that determines if codes are used (default) or names are used (if you check the box).
       2. I then made the default search a combination of name and code (name first so sorts by names)…this way you can search on both name or code at once. Added a check box to just search on code (which sorts by code). May delete the extra code for the check box as I’m not sure anyone needs that, but will see what response I get at ODWC workshop before deciding.
          1. I’m saving the main datafiles with new fields for name\_code on lake and gear, and code\_name for spp. This is considerably faster than creating these on the fly when loading. It is noticeably slower to read the file in, but it adds about 0.6 seconds for database with all years in it whereas it is many seconds to add the columns (might be 10x slower, but I did not formally benchmark).
       3. Added code that filters remaining choices to only those that still are options in the database given what has already been selected
       4. Also added code so that if only one item is available as an option, it automatically becomes selected. This means if you upload your own data and there is just one data set in it, it auto-selects everything.
       5. Old things I tried with cross-populating name-code boxes…I’ll delete this once I deploy everything and know I do not need it.
          1. Originally had all in one observe() function. Separating each box in its own observeEvent triggered by changing the input$??? where ??? is the name of the selectize box value for the alternate pair prevented infinite loop, but it also makes it so that you cannot delete the last selected value and have that propagate to the paired selectize box (it always leaves one item in there). I finally settled on putting each pair in an observeEvent function and this seemed to solve both issues. Using observeEvent is also necessary for filtering remaining choices (see below) as this cannot be done with observer updateSelectizeInput statements as it creates infinite loops.
          2. I also added code to check new values against the previous values to stop any activity if there is a “new” value that does not actually represent a change (also helps prevent infinite looping)
             1. To make this work, I created selBoxOld$??? values to save the last state of each box…if a change is detected, but the new value matches the old value, nothing is done by the observeEvent box. This effectively kills the infinite loop problem. I then needed to use if else code in the updateSelectizeInput to set blank boxes to character(0) to get it to make the box blank
          3. Still a bit imperfect:
             1. One bug remains…for some reason, if you delete all values from one of the boxes on the right (names rather than code boxes), it does not auto-update the choices in all boxes, but rather is stuck on whatever choices were associated with the last selected value(s)
             2. Occasionally things hang a long time or simply delete a selection….cannot reproduce this reliably enough to debug and it is rare. Can be worked around by just waiting, then reselecting if the previous selection did not take.
    5. Updated read\_csv to read.fst. Started by chancing to fread from data.table package and stuck with .csv files…this approach was about 10x faster at loading large files (very noticeable improvement). Further, used fst package and save main databases as \*.fst files because this is even faster (fread on csv file takes about 1.5x longer than read.fst) and I can save the setkey values of data.tables this way…which means we save time on loading and setting keys while still getting speed of data.table-based data.
       1. Should someday finish rewriting code to keep the main data in a data.table rather than data frame. I’ve mostly done this, but did not rewrite all code on the multispp, single-spp, and percentile output tabs (but all selection code now uses data.table functions).
    6. Redesigned code for user-uploaded age data so this uses the same routine as the built-in age data
       1. Fixes bug where if user uploaded file with more than one spp, it would produce ALK using all spp combined.
       2. Added code so that if only one species, lake, and year were in uploaded data, it auto-populates the selectize boxes
       3. As above, combined spp code\_name and did away with cross-talking code-name selectize boxes.
    7. Fixed bug in CPUE table where Species.Code 98 “No fish in sample” was showing as a row in the final CPUE table.
    8. Changed equation for age-frequency figure to multiply by 100 to make percentages not proportions (Y-label said percent, but decimals were showing).
    9. Modified code to prevent as many warning/error messages
       1. Rewrote abiotic data table on Catch Analysis tab to explicitly deal with all NA data
    10. Added theoretical maximum age, observed maximum age, and two estimates of natural mortality based on theor max age or von bert parameters (includes download button for nat mort estimates)
    11. Modified user manual to detail percentile tabs and account for several of the above changes.
14. 6/8/2022 Added checkbox and code that allows ALK to extrapolate to fish smaller than those aged. Saw that ODWC frequently had a floor on fish sizes that are aged (e.g., crappie less than 100mm are always age-0 and are never aged). This produced inaccurate length at age 0 as only fish 100mm and larger were considered in the calculation given that was the smallest size aged.
    1. Default is to allow this. Only possible problem would be if only age-1+ are aged…then the check box might try to parse age-1 vs age-0 incorrectly (or might assume all smaller fish are age-1 and no age-0 exist)
15. 9/1/2022 – Modified to account for Verified.TL and Verified.Wr files in user-uploaded files rather than common Verified.TL.Wr column (was afraid users would fix TL issues but miss Wr issues that are left related to weight errors).
16. 1/24/2023 – Addressed bug found by Nate Hull…he had crappie data where some TL\_mm in sample data were missing. When applying age length key, the default is to extrapolate to smaller fish using multinomial estimates (i.e., extrapAge check box is true by default), which then finds the minimum TL from either age data or sampling data to set minimum length class for all age-related analyses…but with NA’s in the mix it returned NA in the minlencatage2 variable of agesample() reactive function, which messed up almost all of the age analyses. I added a “na.rm = T” statement to all min functions (both for age and for sampling data so whether the check box is T or F there is no chance of this error in future.
17. 2/17/2023 – follow up from Nate Hull bug above...dealing with spp code 108 (all crappie) in new way
    1. Changed name of species code from Crappie spp. to “All crappie spp. combined” to be more explicit.
    2. Add PSD categories based on fact that both black and white crappie have same psd size classes (changed gabelhousenames.fst so the species name listed with spp code 108 is White Crappie...this will populate the White Crappie PSD size categories for this group (which are the same as Black Crappie).
    3. Decided not to use Wr with spp code 108 as there are noticeably different Ws equations for the two species, however, added code to avoid red errors for Wr data (if statements and renderUI() statements that render the normal code if spp != 108, but hides output and displays a warning message if spp == 108).
       1. Actually done by using speciesname() function that returns “All crappie spp. combined” no matter how the user was looking up species names (codes or code-names).
18. 7/13/2023 several updates, bug fixes, and a data upload
    1. updated R to 4.3.0 and RStudio to 2023.03.1 Build 446. Did not actually push an update to the shiny server on this date, but made the following changes to the development branch and these changes will be pushed with the next upload (probably when I get fall 2022 data, which I have not yet received from Ashley).
       1. New R version no longer allows if statements with logical statements where there is >1 item (i.e., an array). This has always silently thrown an error, but the update to all. The solution was pretty simple, just wrap the logical statement in an any() or all() to have it test if any/all item(s) in the array meets the logical statement. Use any() for functions with != (setting result as mismatching if != was correct and match if not) and use all() with functions that have == (and set result as matching if == condition met and not match otherwise).
          1. This will require that if both things being compared have >1 item, the order of items must be the same (i.e., x=1,2,3 and y=1,3,2 will return not matched), but it works for x=1,1,1,1 and y=1 (and returns mismatch if x=1,2,1 and y=1), so probably ok for my purposes.
          2. Also had to change statements with %in% and %chin% to be all(x == y) type statements instead…if I did not do this, it would find any matching value and not require all to match…which resulted in getting messages that said both the age data was a match and it was not a match at the same time.
    2. Tidyr’s unite() function was making my code for user-loaded files break. Apparently in the update, this function strips data.table of its status as a data.table and turns it into a data frame. I fixed by removing the initial as.data.table() function on the statement that reads the file (and now see that was irrelevant as fread defalts to creating a data.table anyhow) and instead put it at the end of a %>% chain of manipulations right before trying to set the data.table key. Thus the unite() functions convert this to a data.frame, but I then manual put it back into a data.table with this statement and then set the key fields.
    3. Above also helped me find bug that was preventing the green “Matched age data set” from being displayed on the Select Analysis tab where the N for aged data is located…fixed by removing input$toggleCodeName from the line of required inputs for the function to run (this only exists if it is checked).
    4. Fixed bug where von Bert curve would not always update and sometimes required changing the check box to convert between inches and mm to get it to respond. This ended up being cause by aged() originally being an eventReactive...changing it to reactive made it work fine. Before getting to that point though, I completely redesigned the way the figure was built and handled. It had used base R and a function() function to make the figure. I altered it to be built in ggplot2 and saved as an object that then was called in place of the function that had been written. I retained these changes even though they were not the source of the problem as I think this is a better approach and moving to ggplot may provide better flexibility for the future.
    5. Uploaded new sampling data
19. 9/15/2023 – Bug fix. Percentile data tab was returning no data if you try to filter by region. I found I had altered the SSP App server.r code to no longer create ODWC.Region on the fly, but rather assumes I have add this column to percentileData.fst when I save it (this was to speed up performance of the ssp app). Unfortunately, I did not do that when I uploaded new data in June 2023. To address the bug, I created a new version of the percentileData.fst file with the needed column, and added some code to the script I use to create this file when new data is uploaded. I did not test the percentileData script as it takes so long to run, but I assume this will work as the change is pretty minor (just added a left-join statement). I should test when we do our next upload.
20. 1/23/2024 – data correction and bug fix. Nate Hull asked me to remove CADD18 Trapnet data from 2022 as the file he submitted was missing fish. I backed this out of the main and percentile databases, recreated the different ssp files for different year ranges and uploaded the new sample and percentile data files. In doing this, I found a “+” symbol at the end of the 3rd line of the catchplot function that produces catch curve that was producing an error...not sure where this came from as I was not working in that part of the code and the old deployed version of the app was working (did not have this typo), but I fixed this and used it in the new app when I deployed to update the databases. A new version of CADDO18\_2022\_31\_sample\_validated.csv will be submitted with the next data upload and I will update it and deploy it at that time (but this deletion was important to avoid kicking that new corrected file out as being a duplicate).
21. 2/16/2024 – bug fixes.
    1. Saw several unusual things using some buffalo data Jasson Schooley had...highlighted some things I wanted to fix.
       1. Noticed that if lots of rare old fish are included, sometimes predicted values of catch curve (used as weights in weighted curve) can be negative or zero. Added code so that when this is the case, we add the absolute value of the smallest value plus 10% to all values as a transformation to ensure it is positive.
       2. Species without Ws equations produce errors on Wr data. I had already used some code to specifically exclude Wr data for species code 108 (all crappie spp. combined), but I genericized this so that any time the species code is not in the wsnames data frame the Wr table is excluded.
       3. Species does not have defined PSD information
          1. The way our Wr code works, it groups by PSD size class...so if a species does not have PSD size classes, it will not get Wr values either. Modified code so only attempts to do PSD-specific Wr if species has PSD size classes. If not, it only returns the “overall” row for the mean Wr of the whole population.
          2. Same as above, but fish without PSD size classes produce PSD errors, so added code to check this and post a message instead of an error in those cases.
          3. Similar change made to CPUE by PSD size class table on catch analysis tab (multi-species analysis).
       4. Modified VB curve code so x-axis only puts tick labels every other value if >20 age classes are on x-axis, every third label if >40, and every fourth label if >80.
       5. Found/fixed bug in age-frequency plot code...does not plot missing age classes (i.e., missing bars where should be bar of height = 0). Also modified figure to use ggplot and to alternate ages displayed with >20 age classes.
    2. Moved location of upload file box on select analysis tab…was bottom right requiring scrolling to use it when the unused check boxes were all towards top of page. Moved all check boxes to right column and put upload age file box in upper left.
       1. Also coded this such that it automatically checks the “Check to use uploaded age data” checkbox once a file has been uploaded, so the user will not need to do this second step (assumption would be the user would not upload if they don’t want to use it…they can always uncheck the box to stop using the file)
22. 2/27/2024 – Deletion of bad data. Nate Hull reported similar issue to the CADD18 with the JNEUST\_2018\_23\_sample.csv file from 2023 upload. I deleted this from all 3 sampling data files and the percentile database. Nate will upload new version with the 2024 data upload (age data were unaffected).
23. 1/20/2025 – major upgrade to match new validation app and additional bug fixes/enhancements
    1. Add some non-standard gear codes and fix code to compute their effort appropriately (e.g., bow fishing…just counts total fish by species without dividing by any effort)
    2. Major re-write for how we handle age data (enter Age on sampling data)
       1. Add age for fish that can be identified as the fish an otolith came from...people will add the age information to the SSP data file and upload it without ever creating a separate age file
       2. Add code to split sampling data into aged and unaged data frames and only assign ages from ALK to unaged fish, then rbind them back together for growth and mort calculations
       3. Add code to build ALK from ages in user-uploaded SSP file when they are present. This is done by extracting these age data and adding to the age database so it will by default select these as matched, but also allow the user to add additional data (if desiring to pool across years or gears). This also allows pairing of age data with SSP data with no ages, just the same way it always has
    3. Add some additional columns to the SSP data structure at the request of the SSP committee: Genetic.ID (help identify fish with otoliths or genetic analysis), Sex, GPS Lat, GPS Lon, Comments
    4. Recoded VB curve to extrapolate with dashed line back to Age = -1 and forward to one age class beyond the data. Actual aged part of curve is solid line.
    5. Recoded checkbox that reads “Extrapolate age-length key to fish smaller than smallest individual aged” so its default condition (checked or unchecked) is based on the age data selected. If there are at least 3 age-0 fish in the age data, it checks the box and will extrapolate, if not, it unchecks and does not extrapolate. This is important because no age classes younger than the youngest age in the age data will occur in the key…so it cannot make age-0 fish if there was not at least 1 fish aged as age-0, and I think a minimum of 3 should be enforced so the multinom() regression can predict well.
    6. Added error trapping to nls() function that fits VB curve…this then hides the VB plot and VB parameter table and in their places puts an error message saying it did not converge. Also used this to modify the natural mortality estimation table to account for the fact that Pauly method throws an error if VB parameters are not available.
    7. Added confidence intervals and prediction intervals for VB curve (uses predFit() function from investr package, which is WAY faster than boostrapping confidence intervals, so this is now practical to include without slowing the page rendering speed.
    8. Rewrote ggplot code for VB curve that used legend.position to use legend.position.inside as this is the new way to do legend positioning inside a theme (and the old approach I had used was depreciated)
    9. Rewrote ggplot code for VB curve and age-frequency histogram that used guide = "axis\_minor" as this also was depreciated in recent ggplot update. In searching this, I also found my “theme(ggh4x.axis.ticks.length.minor = rel(0.5))+” is also depreciated and replaced it with “theme(axis.ticks.length.minor = rel(0.5))+”. This now throws a warning because it conflicts with cowplot but works and is the preferred way moving forward…I’m assuming an update to cowplot will someday fix this, but the figure still renders properly (this makes minor tick marks 50% the length of major tick marks)
    10. Rewrote length-frequency, length-weight, and catch curve plots to use ggplot
    11. Added color to all figures. This is handled by updating the ggplot in the renderPlot section so the downloaded version is still grey scale. This unfortunately is causing several error messages in the log about scale or fill already being present. Not all instances do this, but the length-frequency histogram does for sure and the age frequency histogram does not for sure...I think VB and year class strength may also, but I’m not sure if catch curve might be the other one instead of one of those...I should someday try to clean this up to avoid clutter in the logs, but I need to deploy and don’t have time to chase this down now.
    12. Added check box that will print (or not) species name on all the figures on single-species analysis tab.
    13. Rewrote code for bubble plot to use colorized ggplot for bubble plot and also provide a stacked column area plot to visualize age-length key (set this latter option as default as I think it is easiest to visualize). These two ALK-visualizing plots are now handled within the same reactive function to save some coding…accommodates smoothed or observed ALK, bubble plot or stacked column area plot, and extrapolating to smaller ages or not (on smoothed curve only) within this code.
    14. Added display indicating count of fish in sampling data (count of fish in aged sample is present, but nothing indicated how many of the spp that were not aged existed).
    15. Rewrote code for CPUE by PSD section so it hides the PSD size class definitions unless the user checks a box...I’m trying to make this a bit cleaner and suspect many people are not needing these all the time.
    16. Added toggle button to change CPUE by PSD to CPUE by inch class and back again...then wrote code for building CPUE by inch class table.
        1. User can set their own bin size, the min bin to display, and toggle between inches and mm.
    17. Fixed bug that was preventing ability for custom proportions to be used on the percentile tab (caused by update that changed the way column names were referenced for data.table(input$ percentileInpt).
    18. Updated hatchery data...had been mid 2022 since we had last update, so this was a large addition.
24. 4/29/2025 – bug fixes from update reported by Nate Hull
    1. Changed color of 95% CI on VB as it made it hard to see black data points…was dark blue and made it a lime green
    2. Changed handling of CI and prediction interval in VB so it does not change the Y-axis scaling (rather cuts off the interval rather than rescale Y to accommodate)
    3. Changed positioning of VB legend to ensure it does not cut off text
    4. Used suppressMessages({}) to prevent all the warnings about overwriting fill and color where I recolor figures (black and white version for downloading, color version created with second scale\_fill\_manual or scale\_color\_manual in renderPlot command).
    5. Changed code for scaling length-frequency & ALK plots so they use logical break points for major and minor tick marks that follow bin size and have prominent major ticks and grey/shorter minor ticks.
    6. Added confidence intervals for theoretical max age
25. 5/6/2025 – added units of effort to all CPUE tables and effort summary box on catch analysis tab. Also added count of stations to total effort summary. Also cleaned up verbiage used in table headings and generated values to provide more consistency across app (percentile tab matching new CPUE by PSD and inch class table)
26. 5/29/2025 – bug fixes and interface tweaks done by Dan
    1. On Wr table, moved “overall” to top PSD size classes (was last size class listed)
    2. Moved Wr equation information below the Wr table
    3. Fixed bubble plot for longer-lived fish…Y-axis becomes age class and labels overlap if you display every label, so added code to use every other, every third, etc. as more age classes are in existence.
    4. Fixed bug with text coloring on ALK bar plot that was putting light text on light colored bars (I used unique(Ages) rather than levels(Ages), which caused the ages to be listed in a different order than the figure was using them).
    5. Added code to year class strength plot to provide text message if user has selected more than one year of sampling data (ok to combine aged data sets).

Future things to do

* 1. Change way ALK is used to assign ages to use the observed ALK first if it can, then if a size class is missing, use the smoothed ALK in those cases. This could be handled with a toggle to 1) do the old approach (always using smoothed key) vs 2) this new approach.
  2. Remove “Check box to filter by codes...” on first tab. Will require rooting out a bunch of if statements that reference this as just removing the box will break the app if there are any references to it.
  3. Figure out way to check age data and force zero’s in for ages that are missing on MLA table, mean wt at age table, and year class figure (use Nate’s FHC data…it is missing ages 23, 32, and 33...on second thought, this does not make sense as I would have to have NA for all but the count (which would be count = 0). Maybe just do this for year class strength graph (but what value should be used...residual of line to zero?)
  4. Rewrite code for PSD size classes on CPUE by PSD category page so it is 2 rows, first being inches, second being mm rather than 2 separate tables (like it was done on single-spp age data tab)…but not sure this is best for gears with lots of species in it…need to think a bit more.
  5. Found bug doing above CADD18 sample deletion (from 1/23/2024 update)...apparently percentile data added in June 2023 got duplicated somehow...once without age data, then once with age data...need to figure this out. Might be something to tackle with the next data upload as I suspect it is a problem with the code I use for adding percentile data from new uploaded data.
     1. Work in progress...I have not yet removed duplicate values nor figured out why/how this happened in the first place.
  6. Add checkboxInput to convert ALK figure X-axis between inches/mm
  7. Add checkboxInput to show ALK as table rather than figure (could still use smoothed or observed button to change between ALK’s that are displayed). If I do this, I should delete zero values and round to 2 decimals.