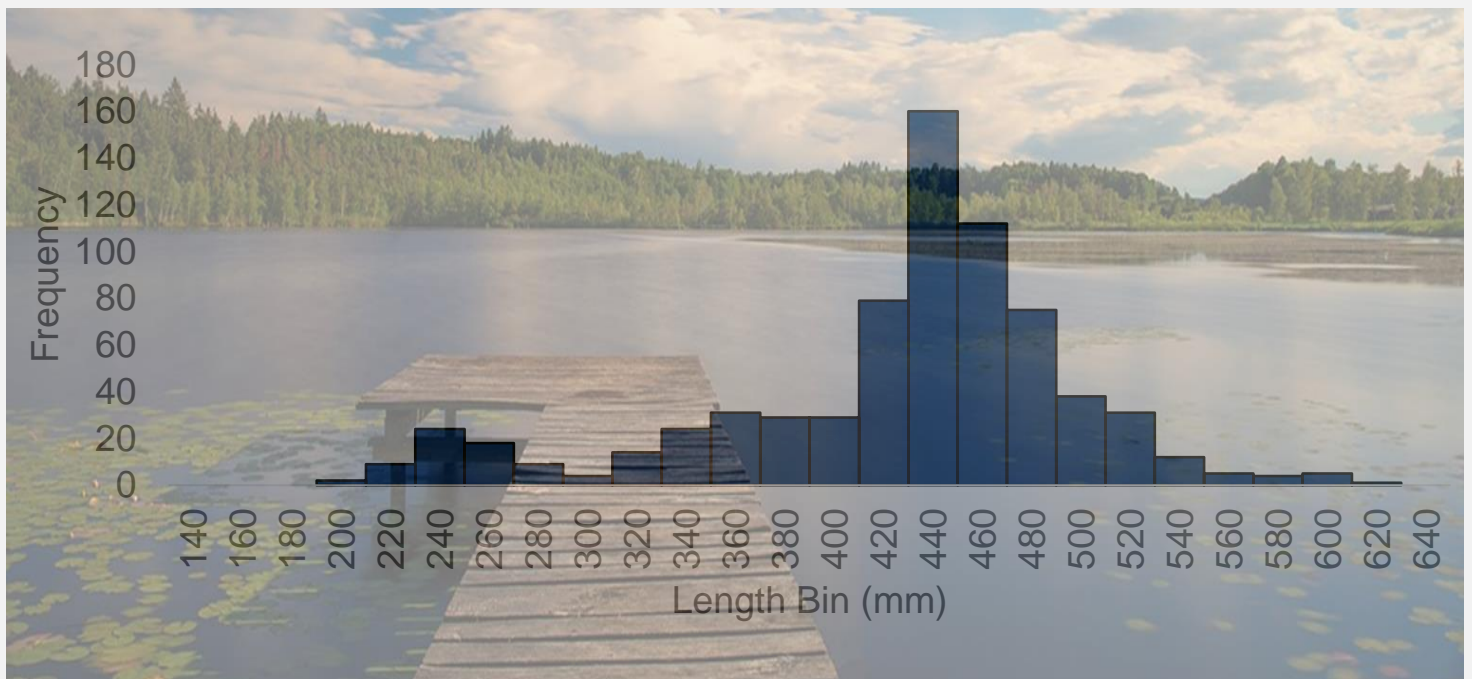


Oklahoma Fishery Analysis Application:

*An R-Based Application for Analysis of Standardized
Fishery Samples*

Oklahoma Department of Wildlife Conservation

Users Guide – Data Validation



Dray D. Carl and Daniel E. Shoup, Oklahoma State University

OFAA v1.0

04.08.2018



Table of Contents

Overview	2
Acknowledgements	2
Data Validation App.....	3
Data Fields and Descriptions	4
Data Input Considerations	6
R Packages Used.....	8


Overview

Our goal in creating the Oklahoma Fishery Analysis Application (hereafter referred to as OFAA or the app) was to develop an application to analyze standardized fishery samples of the Fisheries Division within the Oklahoma Department of Wildlife Conservation (ODWC). Standardization of fishery sampling is essential (see Bonar et al. 2009 for examples), and analysis of samples should also be considered within standardized sampling regimes. Furthermore, we wanted to create an application with an easy, user-friendly interface to give all biologists and technicians the capability to efficiently and accurately analyze fishery samples. We hope this application enables employees to easily explore historical sample data and compare fisheries across the state.

We recommend users read through this Users Guide carefully before using OFAA for analyses of fish populations. Users should be aware of the assumptions regarding each analysis and the method in which each was calculated. We refer users to two textbooks for more detailed explanations of methods used in OFAA: Analysis and Interpretation of Freshwater Fisheries Data (Guy and Brown 2007), Introductory Fisheries Analyses in R (Ogle 2016).



is an open source programming language and software environment for statistical computing (R Core Team 2017).

Shiny (Chang et al. 2017) is an interactive web application by RStudio  and is the user interface for OFAA.

Shiny application and R code written by *Dray D. Carl* and *Daniel E. Shoup*.

Acknowledgements

This application was funded by the Oklahoma Department of Wildlife Conservation, Project F-50-R. Special thanks go to Kurt Kuklinski and Ashley Nealis for help throughout the project. Helpful input was also provided by Josh Johnston, Jason Schooley, Cliff Sager, and Garrett Johnson.

Data Validation App

1. Purpose

- a. The first purpose of the Data Validation App is to help provide a “filter” to ensure data integrity before appending sample data or age data to the 2 main databases that are read by the main analysis app.
 - i. The person (“gate-keeper”) involved in this step should make sure all sample data is run through this app before appending to main database.
- b. The second purpose of this app is to help provide a “screening” for individual datasets before importing into the main app.
 - i. This will help ensure the main app is able to read and evaluate the dataset correctly and without throwing errors.
- c. The third purpose of this app is to provide the sample dataset with a SampleID. This is essential for calculating CPUE in the main app and acts as an effective barrier to importing data into the main app without validation.

2. Validation Rules

- a. These data validation rules were put in place for two reasons. 1) To help ensure quality data are being used by the ODWC to aid in making informed fisheries management decisions. This also helps provide data integrity of the ODWC Fisheries Database to help back findings or views in peer-review or in court. 2) To help ensure data run smoothly through the main Oklahoma Fishery Analysis Application. For these reasons, please do not abstain from this step in the process.
- b. Details of these rules can be found in the Validation Rule Details Tab.

3. Important Note

- a. This data validation app does a good job identifying errors for most required fields (e.g., ensuring consistent use of lake, gear, and species codes). However, it is impossible to code for, or anticipate, all possible data entry errors. In other words, it’s not a “catch-all.” Therefore, we still recommend users practice good data entry techniques when using these apps. These can include:
 - i. Spot-checking spreadsheets for errors (e.g., checking every 5th row)
 - ii. “Double-entering” data and comparing. Two people entering data can help identify mistakes from handwriting, etc.
 - iii. Observing analysis output with a critical eye (e.g., does this number actually make sense, should I go back and look at the data?)

Data Fields and Descriptions

Details for each field (column) for reference when initially entering data. “Required” means the field cannot be left blank. If a field is not required and no data were collected for the field, then it should be left blank. Please see the SSP Manual Tab for more detailed information on some fields.

1. Sample Data

- a. Lake.Code (Required)
 - i. 5 digit code from established Lake Codes (can be downloaded in Validation Rule Details Tab or found in SSP Manual Tab)
 - ii. Identifies the water body sampled – capitalization matters
- b. Station (Required)
 - i. This field is flexible. It can be a number, letter, or alphanumeric.
 - ii. Identifies an individual sample unit. For example, an individual net or electrofishing run.
- c. Month (Required)
 - i. The month the sample was conducted.
 - ii. An integer between 1 and 12.
- d. Day (Required)
 - i. The day the sample was conducted.
 - ii. An integer between 1 and 31.
- e. Year (Required)
 - i. The year the sample was conducted.
 - ii. An integer between 1980 and 2050.
- f. Time
 - i. Military time style, time at which sample was conducted.
- g. Pool.Elevation
 - i. This is the elevation of the lake at the time the data is recorded.
Units = feet above mean sea level.
- h. Surface.Temp
 - i. This is the temperature at the water surface at the time the data is recorded. Units = Degrees Fahrenheit (Recorded to the nearest whole degree.)
- i. Secchi
 - i. This is the Secchi Disc reading at the time the data is recorded.
Units = inches (nearest inch).

- j. Conductivity
 - i. This is the conductivity reading at the time the data is recorded.
Units = micromhos/cm at 25° C.
- k. Gear.Code (Required)
 - i. Identifies which gear was used in the sample. Must be from established list of gear codes (can be downloaded in Validation Rule Details Tab or found in SSP Manual Tab)
- l. Gear.Length (Required)
 - i. Gear length is coded according to gear type being used:
 1. Trap Net = actual length of lead (nearest foot)
 2. Seine = seine length (40 ft or 20 ft)
 3. Gill Net = actual length of net (nearest foot)
 4. Electrofish = actual length of electrofishing effort (15 minutes)
 - ii. See Validation Rule Details for limitations of this field
- m. Habitat
 - i. See habitat codes in SSP Manual Tab
- n. Effort (Required)
 - i. This is the unit of effort expended with a given gear type expressed in the following form:
 1. Seine Sampling - effort is expressed in total area sampled per station. Each station is recorded separately. In quadrant seine sampling, the total area sampled depends on the length of seine and the number of quadrants covered (1 quadrant is $\frac{1}{4}$ of a circle).
 - a. Example: If seine length = 20 ft then the number of quadrants covered multiplied by 312 ft² = total area sampled.
 - b. Example: If seine length = 40 ft then the number of quadrants covered multiplied by 1259 ft² = total area sampled.
 2. Gill and Trap Nets - This is expressed in total number of net hours fished per net. Example: A net fished from 1700 hours to 1500 hours is recorded as 22 net hours of effort.
 3. Electrofishing - units of effort are measured in 15-minute units of 'actual fishing time.' Samples must be collected in discrete 15-minute units of effort.
- o. Species.Code (Required)

- i. Numeric code which identifies the species for each row. Must be from established list of species codes (can be downloaded in Validation Rule Details Tab or found in SSP Manual Tab)
- p. Number.of.individuals (Required)
 - i. Number of fish that apply to the corresponding row (most often 1). When entering species code 98 (no fish caught), Number.of.individuals should still be 1.
- q. TL_mm
 - i. Total length of the fish (body length for paddlefish) in mm. This field is not required, and if length is not recorded, leave the field blank.
- r. Wt_g
 - i. Weight of the fish in grams. This field is not required, and if weight is not recorded, leave the field blank.

Important Note: 2 differences in column names between sample data and age data sheets. “Gear.Code” in Sample Data, “Gear” in Age Data. “TL_mm” in Sample Data, “TLmm” in Age Data. Sorry, but I didn’t notice this before beginning to program the app.

1. Age Data Fields

- a. Same rules apply for synonymous fields, except the following:
 - i. Age (Required)
 - 1. Age of the fish. An integer between 0 and 40. Will not accept anything but an integer (e.g., not YOY, 10+)
 - ii. TLmm (Required)
 - 1. This field is now required in the age dataset. The age of a fish is useless without some corresponding information about length.

Data Input Considerations

Because the app is essentially automated code running analyses, it cannot predict or overcome any data entry/input mistakes by members of the ODWC. This process will begin with ODWC members submitting their sample data to Kurt Kuklinski in the form a .csv file. This file will have specific columns and specifications, and Kurt will run every file through a string of code with validation checks that will ensure data quality before adding it to the database referenced by the app. However, some types of data entry errors cannot be anticipated or recognized. So, we have inserted a few reminders and instructions to keep in mind while filling out spreadsheets to submit for integration into

the database. Following these reminders will help smooth the process of appending the database and ensure correct analyses in the final products.

1. The Effort and Gear.Length columns are commonly mixed up or mistaken somehow within the database which leads to miscalculated CPUE's.
 - a. Reference the ODWC SSP Manual for reminders on what should be included in each of these fields.
 - b. Each of these fields has different meanings (units) for different gears, and CPUE is calculated differently (some use Effort, some use Gear.Length) based on the gear type used.
2. On the same note, the Station field is also very important for calculating CPUE's and is also commonly missing throughout the database.
 - a. The Station field is what differentiates electrofishing runs or net sets...without this field, the app assumes all runs or nets were one big sample. It does not matter if the Station identifier is numerical, an alphabet letter, or alphanumeric...they just need to be distinguishable from each other "sample" from that day in the lake.
3. Zeroes are not the same as null values (NA's or blanks).
 - a. For example, if a weight was not collected for an individual fish, the user should leave the Wt.g. field blank...not enter a zero. Program R recognizes zeroes and nulls (blanks) differently, and in this case, if the user were to enter a zero, R would literally interpret the record as having a weight of 0 grams.
4. Species.Code 98 is important to remember because it is very important when calculating CPUE's.
 - a. Species code 98 means no fish were caught in the sample (e.g., an empty net). More details can be found in the ODWC SSP Manual.
 - b. If you think about it, if the user does not include all the sample information with a species code 98 at the end as a single record, the sample information would never end up in the database. A sample with zero fish captured is just as important of a data point as a sample with 100 fish.
 - c. This is also important for calculating mean CPUE's. The app uses a function which effectively identifies whenever a species should have a catch of zero within a station (e.g., a net). Without a record with Species.Code 98, the function would not know to tell the app to add a zero catch for each other species present in the rest of the sample (i.e., the rest of the nets). See Methods and Assumptions of Calculations for CPUE for another example.
5. The Number.of.individuals field should be filled out for each record. This is simply the column where the user identifies how many fish apply to the record. For example, if a net is full of hundreds of 5" crappies, it is sometimes easier to

count numbers of fish within length bins. If the record is simply just a single, individual fish (most instances), then a 1 should be entered.

6. Lastly, Program R is case-sensitive. Thus, the user should be careful regarding the case (upper/lower) that is required by the app, and follow the specific instructions provided for data entry. For example, lake codes are in all caps...the app would identify the same lake code in lowercase vs. uppercase as different lakes.

R Packages Used

FSA

Ogle, D.H. 2017. FSA: Fisheries Stock Analysis. R package version 0.8.17.

shiny

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2017). shiny: Web Application Framework for R. R package version 1.0.5.

shinyjs

Dean Attali (2017). shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. R package version 0.9.1.

V8

Jeroen Ooms (2017). V8: Embedded JavaScript Engine for R. R package version 1.5.

dplyr

Hadley Wickham, Romain Francois, Lionel Henry and Kirill Müller (2017). dplyr: A Grammar of Data Manipulation. R package version 0.7.4.

plyr

Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, 40(1), 1-29.

tidyr

Hadley Wickham and Lionel Henry (2017). tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions. R package version 0.7.2.

tibble

Kirill Müller and Hadley Wickham (2018). tibble: Simple Data Frames. R package version 1.4.2.

formattable

Kun Ren and Kenton Russell (2016). formattable: Create 'Formattable' Data Structures. R package version 0.2.0.1.