subset aThis document needs to be saved to your Root Folder.

**Class Project**

The ***Class Project*** is the only assignment that you will be evaluated on in this class. The purpose of the ***Class Project*** is to apply the skills you learn in class to your own dataset(s) and variables. If you do not have a dataset, you can find one online (just google "R datasets"). There are many freely available datasets online [including hundreds at this website](https://vincentarelbundock.github.io/Rdatasets/datasets.html). You can use multiple script files in the project.

***To pass the class*** you need to:

* Meet all the requirements in ***Table 1: Project Requirements Table***.
* Earn 75% of the points in the ***Table 2: Skill List***.

Class Project submissions are sent to the instructor ([belinsky@msu.edu](mailto:belinsky@msu.edu)). The submission needs to include:

* This document with the ***Table 2: Skill List*** filled out
* All R script files used in your project
* All data files used in your project
* A list of R packages you used in your project (if any)

|  |  |
| --- | --- |
| ***Table 1: Project Requirements*** | **Lesson** |
| * Script is well commented | 1.1 |
| * Use semicolons ( ; ) to end commands throughout your scripts. | 1.1 |
| * Consistent alignment of curly bracket ( **{ }** ) | 1.7 |
| * Give a brief (1-2 sentences) description in the ***Description*** column of ***Table 2*** for each skill you used. |  |
| * Skills completed in ***Table 2*** are commented in your script file where the skill is demonstrated.   The comment should be # SKILL XX where XX is the skill number in ***Table 2***. |  |
| * Script lines have no text beyond the 80th character with exception of:   1) long file names (these cannot be broken down)  2) SKILL XX comments | 1.6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Table 2: Skills List*** (you can use the same line(s) of code to satisfy multiple skills) | **Lesson** | **Max**  **Points** | **Your Points** | **Description** (1-2 sentences and make sure you adjust the points in the ***Your Points*** column) |
| * Use of a standard mathematical operations (+, -, \*, /). *Half-point per use/ max of 1 point.* | 1.3 | 1 | 1 | length/weight summary statistics; calculate sd and se; |
| * Using parentheses to order mathematical operations | 1.3 | 1 | 1 | length/weight summary statistics;  calculate se; |
| * Using powers or roots | 1.3 | 1 | 0 |  |
| * Get input from user and save the input to a variable | 1.5 | 1 | 1 | ask user to input their favorite fish species;  user will confirm if their favorite fish was caught from the unique($) fishList; |
| * Output mixed message (variables and text) to Console Window | 1.6 | 1 | 1 | output mixed message of the users  'favFish' |
| * Use of line feed (newline) character | 1.6 | 1 | 1 | age distribution plot;  adjust x and y axis spacing |
| * Use of different conditional operators.   *Half point for each unique operator (6 in all).* | 1.7 | 3 | 3 | >, <, >=, <=, !=; used through script;  e.g.,  - PSD analyses grouping walleye based on size  - loop through ages to determine ageMin/Max |
| * Use of if-else structure | 1.8 | 1 | 1 | check if user has caught/angled for walleye |
| * Use of if-else-if structure | 1.8 | 1 | 1 | Check user favorite species |
| * Error condition in an if-else structure | 1.8 | 1 | 1 | Give error condition if user favorite species not recognized |
| * Use of && operator | 1.9 | 1 | 1 | proportional side dist.;  subset walleye into size categories |
| * Use of || operator | 1.9 | 1 | 1 | subset individual fish with assigned sex;  grep(M|F) |
| * Using && or || to check conditions on multiple variables | 1.10 | 1 | 1 | subset female walleye > age 4  dataset for recruit-sized fish. |
| * Save data from CSV file to a data frame | 2.1 | 1 | 1 | import walleye dataframe, save as 'walleyeCSV' |
| * Subset one cell in a data frame – save to variable | 2.2 | 1 | 1 | user grabs a random length from 'walleyeData' |
| * Subset column in a data frame – save to vector | 2.2 | 1 | 1 | save 2009 and 2010 age data to separate  variables; 'age2009' & 'age2010' (dplyr()) |
| * Subset individual value in vector | 2.2 | 1 | 0 |  |
| * Use of sequence | 2.3 | 1 | 1 | mortality calc;  -ensure catch at age table has no missing cohorts  - calculate 'code' for robson-chapman est. of S/Z/A |
| * Use of sequence that increases or decreases by number other than 1 | 2.3 | 1 | 0 |  |
| * Sequence to set up iterations in a for() | 2.4 | 1 | 0 |  |
| * Three uses of state variable. *One point / use* | 2.4 | 3 | 3 | pooled age (cohort) analysis;  i) ageMin  ii) ageMax  iii) ageSum (to calculate mean age) |
| * Nested if() inside a for() | 2.4 | 1 | 1 | cohort analysis;  check/replace ageMin/ageMax and ageSum state variables |
| * Use of Boolean values. | 2.4 | 1 | 1 | cohort analysis;  identify whether any age-0 walleye exist within dataframe;  doubles as outlier check |
| * Find max, min, or average value of a vector using for() | 2.5 | 2 | 2 | loop over lengthageVector;  replace state/boolean variables for min/max/sum |
| * Ignoring NA values | 2.5 | 1 | 1 | walleye summary statistics;  mn, sd, and se length |
| * Add vector to data frame | 2.6 | 1 | 1 | add log-log age/length to dataframe |
| * Rearrange columns in a data frame | 2.6 | 1 | 1 | rearrange columns in 'walleyeData' to 'adjwalleyeData' |
| * Write a data frame to a CSV file | 2.6 | 1 | 1 | save reformatted dataframe (walleye) from master commercial fishing dataframe |
| * Label x-axis and y-axis and add title to a plot | 2.7 | 1 | 1 | age distribution by 'sex' and 'year';  xlab() + ylab() |
| * Adding a legend to a plot | 2.7 | 1 | 1 | age distribution by 'sex' and 'year';  ggplot() + scale\_fill\_manual(guide = guide\_legend(title = )) |
| * Adding colors to a plot | 2.7 | 1 | 1 | age distribution by 'sex' and 'year';  added custom hex values to colorFill;  added values = colorFill to scale\_fill\_manual() |
| * Adding lines or points to a plot | 2.9 | 1 | 1 | condition analysis;  geom\_point() and geom\_smooth();  plotted linear regression (condition ~ weight) |
| * Histogram | 2.7/2.9 | 1 | 1 | walleye age distribution by sample year;  geom\_histogram() |
| * Barplot | 2.7/2.9 | 1 | 1 | walleye barplot by sample year and sex;  geom\_bar() |
| * Multi-panel scatterplot using pairs() | 2.8/2.10 | 1 | 1 | walleye overview of different measurable variables  ~ Weight + Length + Age + condition |
| * Create and use two of your own functions (2 point/each) | 2.10 | 4 | 2 | predict age at length (e.g. age at 350 mm) |
| * Create a function with at least 2 input parameters | 2.10 | 2 | 2 | predict age at length given von Bertalannfy growth parameters via vbStarts(); FSA package |
| * Save the return value from your function to a variable | 2.10 | 1 | 1 | predict age at 350 mm using von Bert. parameters and custom predAge() function. |
| * Create a function with default parameters | 2.10 | 1 | 0 | predict age at 350 mm |
| * Use two functions from another package (1 point/each) | 3.1 | 2 | 2 | - vbStarts() to get vonBert parameters to predict growth in function.  - dplyr() -> select() -> rename; wrangle raw dataframe |
| * Reshape a data frame | 3.2 | 2 | 2 | length/weight summary statistics;  use group\_by() and summarise() within dplyr();  estimate mn, sd, and se of length by year;  reshape age data to plot mean length |
| * Perform operation on multiple columns of a data frame or matrix | 3.2 | 2 | 2 | round condition and log age/length col's in 'walleyeData' |
| * Using substring function | 3.2 | 1 | 0 |  |
| * Rounding values | 3.2 | 1 | 1 | length/weight summary statistics;  round estimates of mean, sd, and se |
| * Renaming columns | 3.2 | 1 | 1 | rename raw variables;  utilize dplyr() function -> rename() |
| * Create a matrix | 3.3 | 1 | 0 |  |
| * Perform operation on rows, columns, and whole matrix | 3.3 | 1 | 1 | round values in col 15:17 in 'walleyeData' |
| * Repeat values using the rep() function | 3.4 | 2 | 2 | repeat user input 'favFish' three times |
| * Perform up to two ANOVAs and comment on results. *One point each* | 3.4 | 2 | 2 | i) aov(length ~ year)  ii) aov(weight ~ year) |
| * Perform up to two t-tests and comment on results. *One point each.* | 3.4 | 2 | 2 | ttest1 and ttest2  comparing age (means) from separate sample years |
| * Create a boxplots that uses multiple categories | 3.4 | 2 | 2 | age distribution by 'sex' and 'year';  plot male/female walleye by sample year |
| * Randomly sample data | 3.5 | 1 | 1 | randomly sample() ageData and plot hist() |
| * Create pseudo-random values using set.seed() | 3.5 | 1 | 1 | random number generation;  add set.seed to beginning of script |
| * Sample from a normal distribution | 3.5 | 1 | 0 |  |
| * Subset a list | 3.5 | 1 | 1 | count 'Dwarf Walleye';  calculate mean length/age of dwarf walleye in dataframe |
| * Perform two linear regressions and comment on results. *One point each* | 3.6 | 2 | 2 | i) model condition ~ length; summary(modelKlen)  ii) model condition ~ weight; summary(modelKwt) |
| * Add regression line to a plot | 3.6 | 1 | 1 | condition analysis;  2018 linear regression (condition ~ lenght)  geom\_smooth() |
| * Up to four examples of subsetting a vector using which(). *One-half point each.* | 3.7 | 2 | 2 | proportional size distribution;  which() to separate walleye into size categories. |
| * Use grep() on a vector | 3.7 | 2 | 1 | i) identify 'dwarf walleye' in walleye dataframe |
| * Up to three examples of using subset vector to index another vector. *One point each.* | 3.7 | 3 | 0 |  |
| * Multiple condition on a subset vector | 3.8 | 2 | 0 |  |
| * Multiple conditions in a grep() | 3.8 | 1 | 1 | subset male (M) and female (F) data |
| * Use of union() or intersect() | 3.8 | 1 | 1 | insersect(); determine if ageMax appears in consencutive sampling years (2009, 2010) |
| * Up to two examples of plotting subset vectors. *One point each.* | 3.7/3.8 | 1 | 1 | plot walleye PSD categories (i.e., quality vs memorable) |
| * Save results as an .rdata file | 3.5 | 2 | 2 | save 2009 && 2010 walleye age data |
| **Total Points**  Add up all the points from the ***Your Points*** column and put the results in this row. |  | **88** | **0** | If you highlight the cell to the left and press F9, the points in the ***Your Points*** column will automatically be added and the results put in the cell. |