Database Acquisition

Madie Munro

4/6/2023

# R Setup:

**In order to set the correct project directory for this stage of development, go to:**

*Tools > Global Options > R Markdown*

**Once there, set “Evaluate chunks in directory” from “Document” to “Project.” Then, select “Apply” and hit “OK” when done.**

**The code chunk below imports libraries needed for R setup including knitr to knit the file to Word or HTML and here to read in the utility R file used for certain functions for data wrangling. It will not display when knitted to Word or HTML.**

# Introduction:

## File Reading:

**The following code chunk below imports the needed libraries for table ripping:**

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(dplyr)  
library(naniar)  
library(stringr)

**Next, the original dataset was converted from an Excel file to a CSV file in order to read in the data easier (see 03\_incremental folder for the .csv version of the data). In addition, a column lookup dataset was read in, containing the new names for the columns of the database as well as a description of the type of data stored in each respective column. Lastly, extra na values were dropped after the table was read in:**

fish <-   
 read\_csv("./01\_acquisition/03\_incremental/fish\_info.csv")

## New names:  
## Rows: 156 Columns: 40  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (30): Common Name, Scientific Name, Swim Mode, Min Speed, Max Speed, Swi... dbl  
## (10): BD/TL, TL/SL, TL/FL, Default Time, B a, B b, B c, P a, P b, P c  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `Units` -> `Units...7`  
## • `Variability` -> `Variability...8`  
## • `Variability Type` -> `Variability Type...9`  
## • `Variability` -> `Variability...14`  
## • `Variability Type` -> `Variability Type...15`  
## • `Units` -> `Units...16`

fish\_cols <-   
 read\_csv("./01\_acquisition/01\_input/column\_lookup.csv")

## Rows: 16384 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): OldName, NewName, ColDesc  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

fish\_cols <- drop\_na(fish\_cols)

## Column Renaming:

**The column names stored in “fish” were changed to reflect the new column names to be used in the updated database. This was accomplished by creating a subset of the “fish\_cols” data, pivoting the data wider in order for the new column names to be the dataframe column names, and then renaming each column name in “fish” to the column names in the subsetted data. Lastly, the column “FishIDX” was added to the end of the “fish” dataset to serve as an indexing variable for each data entry:**

cols\_sub <- fish\_cols[, c("NewName", "ColDesc")]  
fcols <- pivot\_wider(cols\_sub, names\_from = "NewName", values\_from = "ColDesc")  
  
colnames(fish) = colnames(fcols)  
fish$FishIDX <- 1:nrow(fish)

## Data Mutation:

**Some numeric column values gave a range of values rather than one set value, so they were split into two respective columns: low and high. The first column to be split as such was the water temperature column. This was achieved by the utility file, “split\_utils.R,” which contains two functions: one for splitting specifically on the plus-minus character and one general character splitting function (see 01\_acqusition/02\_protocol/split\_utils.R for code):**

#creates a variable with the old temp measurements  
fish\_temps <-  
 fish$Temperature %>%  
 as.list()   
  
#correctly splits up values that are ranges or plus-minus values (ex: 15 plus-minus 3, 12 to 19)  
fish\_temps\_to\_replace <-  
 pm\_split(fish\_temps)   
fish\_temps\_to\_replace <-  
 c(fish\_temps\_to\_replace,  
 char\_split(fish\_temps, "to"))  
  
lapply(  
 fish\_temps\_to\_replace,  
 function(idx){  
 item\_in\_fish\_temps\_list <- idx[[1]]  
 fish\_temps[[item\_in\_fish\_temps\_list]] <<- idx[[2]]  
 }  
)  
  
#turns list elements into two separate dataframe columns  
fish\_temps <- lapply(fish\_temps, function(x) {   
 x <-   
 as.vector(x)%>%  
 as.numeric   
 if(length(x) == 2){  
 return(x)  
 }  
 else {  
 x <- rep(x,2)  
 }  
}) %>% do.call(rbind, .) %>%   
 as.data.frame(.)  
  
#creates low and high temp columns  
  
fish\_temps\_low <- fish\_temps$V1  
fish\_temps\_high <- fish\_temps$V2

**The swim variability column also got the same treatment:**

swim\_var\_temp <-  
 fish$SwimVariabilityValue %>%  
 as.list()  
  
swim\_var\_to\_replace <-  
 char\_split(swim\_var\_temp)  
  
lapply(  
 swim\_var\_to\_replace,  
 function(idx){  
 item\_in\_swim\_var\_list <- idx[[1]]  
 swim\_var\_temp[[item\_in\_swim\_var\_list]] <<- idx[[2]]  
 }  
)  
  
swim\_var\_temp <- lapply(swim\_var\_temp, function(x) {   
 x <-   
 as.vector(x)%>%  
 as.numeric   
 if(length(x) == 2){  
 return(x)  
 }  
 else {  
 x <- rep(x,2)  
 }  
}) %>% do.call(rbind, .) %>%   
 as.data.frame(.)  
  
fish\_swim\_var\_low <- swim\_var\_temp$V1  
fish\_swim\_var\_high <-swim\_var\_temp$V2

**Once the more problematic columns were appropriately split apart, each of the columns with numeric values were converted from character to numeric types. Some values were converted to double, some were converted to integer. The modified data was stored under “fish\_test”:**

fish\_test <- fish %>%  
 mutate\_at(c("MinSpeed", "MaxSpeed", "SwimSpeed", "SwimVariabilityValue", "MinLength", "MaxLength", "LengthValue", "LengthVariabilityValue",  
 "SwimTime", "MinTime", "MaxTime", "TimeBetweenIncrements", "VelocityIncrements", "SwimmingTimeAtMaxVelocity"), as.numeric) %>%  
 mutate(NumberOfFishTested = as.integer(NumberOfFishTested)) %>%  
 mutate(fish\_temps\_high = as.double(fish\_temps\_high)) %>%  
 mutate(fish\_temps\_low = as.double(fish\_temps\_low)) %>%  
 mutate(fish\_swim\_var\_high = as.double(fish\_swim\_var\_high)) %>%  
 mutate(fish\_swim\_var\_low = as.double(fish\_swim\_var\_low))

**The last steps taken before the full data was ripped into separate tables for SQL querying was the renaming of the new columns (low/high temp/swim variability) and replacing the question marks in the “Leap(y/n)” column with “na”:**

colnames(fish\_test)[42] <- "WaterTempHigh"   
colnames(fish\_test)[43] <- "WaterTempLow"   
colnames(fish\_test)[44] <- "SwimVariabilityValueHigh"  
colnames(fish\_test)[45] <- "SwimVariabilityValueLow"  
  
fish\_test <- fish\_test %>% replace\_with\_na\_at(.vars = "Leap(Y/N)", condition = ~.x == "?")

# Individual Table Creation:

## Subsetting Original Data:

**The first step taken to create the individual database tables was to create multiple subsets of the wrangled data based on the database relational model drafted (see 03\_incremental for model):**

fish\_info <- fish\_test[, c("FishIDX", "CommonName", "ScientificName")]  
  
swim\_info <- fish\_test[, c("SwimMode", "SwimSpeed", "SwimUnits", "SwimTime", "Leap(Y/N)")]  
  
fish\_len\_info <- fish\_test[, c("LengthValue", "LengthUnits", "LengthType", "BD/TLRatio", "TL/SLRatio", "TL/FLRatio")]  
  
exp\_method <- fish\_test[, c("AveragingCoeff(Y/N)", "NumberOfFishTested", "TestMethod", "Reference", "DataReference", "Comments")]  
  
equation <- fish\_test[, c("VelocityEquationGiven", "BurstCoeffA", "BurstCoeffB", "BurstCoeffC", "ProlongedCoeffA", "ProlongedCoeffB", "ProlongedCoeffC")]  
  
water\_temp <- fish\_test[, c("WaterTempLow", "WaterTempHigh")]  
  
swim\_ex\_info <- fish\_test[, c("MinTime", "MaxTime", "TimeBetweenIncrements", "SwimmingTimeAtMaxVelocity", "VelocityIncrements")]  
  
swim\_var <- fish\_test[, c("MinSpeed", "MaxSpeed", "SwimUnits", "SwimVariabilityValueLow", "SwimVariabilityValueHigh", "SwimVariabilityType")]  
  
fish\_len\_var <- fish\_test[, c("MinLength", "MaxLength", "LengthUnits", "LengthVariabilityValue", "LengthVariabilityType")]

## Adding/Rearranging Relational Model Columns:

**Since the database relational model features columns in the tables not mentioned in the original or wrangled database (ex: there aren’t any unique ID values), they had to be created from scratch. This was done by creating two separate columns, one numeric and one string, and then uniting them together. Additional units for certain measurements like swim velocity, fish length, and temperature were also added to tables in addition to a column for the central tendency metric used for numeric measurements:**

#Fish Information  
fish\_info <- fish\_info %>%  
 mutate(FishIDNum = row\_number())  
fish\_info <- fish\_info %>%  
 mutate(FishIDStr = "FSH")  
fish\_info <- fish\_info %>%  
 unite(FishID, FishIDNum, FishIDStr, sep="")  
  
fish\_info <- fish\_info %>%  
 mutate(SwimIDNum = row\_number())  
fish\_info <- fish\_info %>%  
 mutate(SwimIDStr = "SWM")  
fish\_info <- fish\_info %>%  
 unite(SwimID, SwimIDNum, SwimIDStr, sep="")  
  
fish\_info <- fish\_info %>%  
 mutate(ExpMethodIDNum = row\_number())  
fish\_info <- fish\_info %>%  
 mutate(ExpMethodIDStr = "EXP")  
fish\_info <- fish\_info %>%  
 unite(ExpMethodID, ExpMethodIDNum, ExpMethodIDStr, sep="")  
  
#Swimming Information  
swim\_info <- swim\_info %>%  
 mutate(SwimIDNum = row\_number())  
swim\_info <- swim\_info %>%  
 mutate(SwimIDStr = "SWM")  
swim\_info <- swim\_info %>%  
 unite(SwimID, SwimIDNum, SwimIDStr, sep="")  
  
swim\_info <- swim\_info %>%  
 mutate(SwimExIDNum = row\_number())  
swim\_info <- swim\_info %>%  
 mutate(SwimExIDStr = "EXH")  
swim\_info <- swim\_info %>%  
 unite(SwimExID, SwimExIDNum, SwimExIDStr, sep="")  
  
swim\_info <- swim\_info %>%  
 mutate(WaterTempIDNum = row\_number())  
swim\_info <- swim\_info %>%  
 mutate(WaterTempIDStr = "WTP")  
swim\_info <- swim\_info %>%  
 unite(WaterTempID, WaterTempIDNum, WaterTempIDStr, sep="")  
  
swim\_info <- swim\_info %>%  
 mutate(SwimVarIDNum = row\_number())  
swim\_info <- swim\_info %>%  
 mutate(SwimVarIDStr = "SVR")  
swim\_info <- swim\_info %>%  
 unite(SwimVarID, SwimVarIDNum, SwimVarIDStr, sep="")  
  
swim\_info <- swim\_info %>%  
 mutate(EqIDNum = row\_number())  
swim\_info <- swim\_info %>%  
 mutate(EqIDStr = "EQ")  
swim\_info <- swim\_info %>%  
 unite(EqID, EqIDNum, EqIDStr, sep="")  
  
swim\_info <- swim\_info %>%  
 mutate(SwimCentrTendMetric = "mean")  
  
swim\_info <- swim\_info %>%  
 mutate(TimeUnits = "s")  
  
#Equations  
equation <- equation %>%  
 mutate(EqIDNum = row\_number())  
equation <- equation %>%  
 mutate(EqIDStr = "EQ")  
equation <- equation %>%  
 unite(EqID, EqIDNum, EqIDStr, sep="")  
  
#Water Temperature  
water\_temp <- water\_temp %>%  
 mutate(WaterTempIDNum = row\_number())  
water\_temp <- water\_temp %>%  
 mutate(WaterTempIDStr = "WTP")  
water\_temp <- water\_temp %>%  
 unite(WaterTempID, WaterTempIDNum, WaterTempIDStr, sep="")  
  
water\_temp <- water\_temp %>%   
 mutate(TempUnits = "C")  
  
#Swim Exhaustion Information  
swim\_ex\_info <- swim\_ex\_info %>%  
 mutate(SwimExIDNum = row\_number())  
swim\_ex\_info <- swim\_ex\_info %>%  
 mutate(SwimExIDStr = "EXH")  
swim\_ex\_info <- swim\_ex\_info %>%  
 unite(SwimExID, SwimExIDNum, SwimExIDStr, sep="")  
  
swim\_ex\_info <- swim\_ex\_info %>%  
 mutate(TimeUnits = "s")  
  
swim\_ex\_info <- swim\_ex\_info %>%  
 mutate(VelocityUnits = "cm/s")  
  
#Swim Variability  
swim\_var <- swim\_var %>%  
 mutate(SwimVarIDNum = row\_number())  
swim\_var <- swim\_var %>%  
 mutate(SwimVarIDStr = "SVR")  
swim\_var <- swim\_var %>%  
 unite(SwimVarID, SwimVarIDNum, SwimVarIDStr, sep="")  
  
swim\_var <- swim\_var %>%  
 mutate(SwimVariabilityUnits = "cm/s")  
  
#Fish Length Information  
fish\_len\_info <- fish\_len\_info %>%  
 mutate(FishIDNum = row\_number())  
fish\_len\_info <- fish\_len\_info %>%  
 mutate(FishIDStr = "FSH")  
fish\_len\_info <- fish\_len\_info %>%  
 unite(FishID, FishIDNum, FishIDStr, sep="")  
  
fish\_len\_info <- fish\_len\_info %>%  
 mutate(FishVarIDNum = row\_number())  
fish\_len\_info <- fish\_len\_info %>%  
 mutate(FishVarIDStr = "FVR")  
fish\_len\_info <- fish\_len\_info %>%  
 unite(FishVarID, FishVarIDNum, FishVarIDStr, sep="")  
  
fish\_len\_info <- fish\_len\_info %>%  
 mutate(LengthCentrTendMetric = "mean")  
  
#Fish Length Variability  
fish\_len\_var <- fish\_len\_var %>%  
 mutate(FishVarIDNum = row\_number())  
fish\_len\_var <- fish\_len\_var %>%  
 mutate(FishVarIDStr = "FVR")  
fish\_len\_var <- fish\_len\_var %>%  
 unite(FishVarID, FishVarIDNum, FishVarIDStr, sep="")  
  
fish\_len\_var <- fish\_len\_var %>%  
 mutate(LengthVariabilityUnits = LengthUnits)  
  
#Experimental Method  
exp\_method <- exp\_method %>%  
 mutate(ExpMethodIDNum = row\_number())  
exp\_method <- exp\_method %>%  
 mutate(ExpMethodIDStr = "EXP")  
exp\_method <- exp\_method %>%  
 unite(ExpMethodID, ExpMethodIDNum, ExpMethodIDStr, sep="")

**Next, the columns in each table were rearranged to match the order shown in the relational database model:**

#Fish Information  
fish\_info <- fish\_info %>%  
 relocate(FishIDX, FishID, SwimID, ExpMethodID)  
  
#Swimming Information  
swim\_info <- swim\_info %>%  
 relocate(SwimID, SwimExID, WaterTempID, SwimVarID, EqID)  
swim\_info <- swim\_info %>%  
 relocate(SwimCentrTendMetric, .after = SwimMode)  
swim\_info <- swim\_info %>%  
 relocate(TimeUnits, .after = SwimTime)  
  
#Equation  
equation <- equation %>%  
 relocate(EqID)  
  
#Water Temperature  
water\_temp <- water\_temp %>%  
 relocate(WaterTempID)  
  
#Swim Exhaustion Information  
swim\_ex\_info <- swim\_ex\_info %>%  
 relocate(SwimExID)  
swim\_ex\_info <- swim\_ex\_info %>%  
 relocate(TimeUnits, .after = TimeBetweenIncrements)  
  
#Swimming Variability  
swim\_var <- swim\_var %>%  
 relocate(SwimVarID)  
  
#Fish Length Information  
fish\_len\_info <- fish\_len\_info %>%  
 relocate(FishID, FishVarID, LengthCentrTendMetric)  
  
#Fish Length Variability  
fish\_len\_var <- fish\_len\_var %>%  
 relocate(FishVarID)  
  
#Experimental Method  
exp\_method <- exp\_method %>%  
 relocate(ExpMethodID)

# Table Export:

**Once the tables were created, they were exported as .csv files to be later read in to SQL (see 04\_product folder to view the tables):**

write.csv(fish\_info, "./01\_acquisition/04\_product/fish\_info.csv", row.names = F)  
write.csv(swim\_info, "./01\_acquisition/04\_product/swim\_info.csv", row.names = F)  
write.csv(equation, "./01\_acquisition/04\_product/equation.csv", row.names = F)  
write.csv(water\_temp, "./01\_acquisition/04\_product/water\_temp.csv", row.names = F)  
write.csv(swim\_ex\_info, "./01\_acquisition/04\_product/swim\_ex\_info.csv", row.names = F)  
write.csv(swim\_var, "./01\_acquisition/04\_product/swim\_var.csv", row.names = F)  
write.csv(fish\_len\_info, "./01\_acquisition/04\_product/fish\_len\_info.csv", row.names = F)  
write.csv(fish\_len\_var, "./01\_acquisition/04\_product/fish\_len\_var.csv", row.names = F)  
write.csv(exp\_method, "./01\_acquisition/04\_product/exp\_method.csv", row.names = F)