Part 1:

Graphical user interface, text, application

Description automatically generated

Part 2:

#=======Exercise 2 Answer Key

This code loads the data into individual data frames with given names (speciesCode, etc.)

#load data

speciesCode <- read.csv("speciesCode.csv")

speciesData <- read.csv("speciesData.csv")

tripData <- read.csv("tripData.csv")

#=======In-class exercise 2

#==Q1

SebastesRows <- grep("Sebastes",speciesCode$Scientific)

This code (referencing the above line) indicates the row indices within the “Scientific” column that correspond to the “Sebastes” character value, and names this object “SebastesRows”.

SebastesRows

This code prints the “SebastesRows” object, which is a vector of row indices.

SebastesCode <- speciesCode[SebastesRows,"SpeciesCode"]

This code creates a new vector names “SebastesCode” that is comprised elements in the “speciesCode” object. The row indices are the same as the row indices specified by the “SebastesRows” object, and this object only includes the column “SpeciesCode” of the “speciesCode” object.

SebastesCode

This code prints the “SebastesCode” object. This object is a vector.

sort(unique(SebastesCode))

This code first identifies unique elements in the “SebastesCode” object—so all the unique numbers in the “SpeciesCode” column. It then sorts these in ascending numerical order.

names(speciesData)

This code lists the names of all columns in the object “speciesData”.

x <- speciesData[speciesData$SpeciesCode %in% SebastesCode,

"SpeciesCode"]

This code creates an object named “x” (a vector) from the data frame “speciesData”. The rows are the same as the rows in the “SpeciesCode” column of “speciesData”, but only those with the code values corresponding to rockfish. The column is the “SpeciesCode” column of “speciesData”.

x

This code prints the vector x.

sort(unique(x)) #check this is a subset

This code identifies all the unique numbers in vector “x”, then sorts them in ascending numerical order.

SebastesData2 <- speciesData[speciesData$SpeciesCode %in%

SebastesCode,]

This code creates a data frame name “SebastesData2”. The rows are the same as the rows in the “SpeciesCode” column of “speciesData”, but only those with the code values corresponding to rockfish. This includes all columns of “speciesData”.

head(SebastesData2)

This code presents the first 6 (?) rows of “SebastesData2” and all columns.

#==Q2

#Table of fates of rockfish by species code

table(SebastesData2$SpeciesCode, SebastesData2$Fate)

This code creates a table with all rockfish species codes (from “SebastesData2”) as row names, and different fate options as column names.

#Fate is a factor, convert to text to get rid of blank factor

table(SebastesData2$SpeciesCode, as.character(SebastesData2$Fate))

This code creates a table with rockfish species codes as row names and different fate options as column names. This time, the class of this table’s row names is “character”.

#==Q3

names(SebastesData2)

This code lists the names of all columns in “SebastesData2”.

#needs na.rm=T because of the NAs

tapply(X=SebastesData2$Length, INDEX=SebastesData2$SpeciesCode, FUN=min)

This code doesn’t work because it takes all NAs into account while doing the min() function.

tapply(X=SebastesData2$Length, INDEX=SebastesData2$SpeciesCode,

FUN=min, na.rm=T)

This code finds the minimum fish length of each subset of species codes. Each species code (for rockfish) has the minimum length of fish in that species code.

tapply(X=SebastesData2$Length, INDEX=SebastesData2$SpeciesCode, FUN=max,

na.rm=T)

This code finds the maximum fish length of each subset of species codes. Each species code (for rockfish) has the maximum length of fish in that species code.