Data analysis project

Jess Qualley and Rebecca Hansen

March 27, 2020

## Clear code (**delete before submitting**\*)

rm(list = ls(all = TRUE))

# Preparation

## Loading and citing packages

#Load faraway package  
library(faraway)

## Warning: package 'faraway' was built under R version 3.6.3

#Load plyr package  
library(plyr)

##   
## Attaching package: 'plyr'

## The following object is masked from 'package:faraway':  
##   
## ozone

#Load tidyverse package  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.4  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'forcats' was built under R version 3.6.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::arrange() masks plyr::arrange()  
## x purrr::compact() masks plyr::compact()  
## x dplyr::count() masks plyr::count()  
## x dplyr::failwith() masks plyr::failwith()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::id() masks plyr::id()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::mutate() masks plyr::mutate()  
## x dplyr::rename() masks plyr::rename()  
## x dplyr::summarise() masks plyr::summarise()  
## x dplyr::summarize() masks plyr::summarize()

#Cite packages  
citation("faraway")

##   
## To cite package 'faraway' in publications use:  
##   
## Julian Faraway (2016). faraway: Functions and Datasets for Books by  
## Julian Faraway. R package version 1.0.7.  
## https://CRAN.R-project.org/package=faraway  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Manual{,  
## title = {faraway: Functions and Datasets for Books by Julian Faraway},  
## author = {Julian Faraway},  
## year = {2016},  
## note = {R package version 1.0.7},  
## url = {https://CRAN.R-project.org/package=faraway},  
## }  
##   
## ATTENTION: This citation information has been auto-generated from the  
## package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.

citation("plyr")

##   
## To cite plyr in publications use:  
##   
## Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data  
## Analysis. Journal of Statistical Software, 40(1), 1-29. URL  
## http://www.jstatsoft.org/v40/i01/.  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Article{,  
## title = {The Split-Apply-Combine Strategy for Data Analysis},  
## author = {Hadley Wickham},  
## journal = {Journal of Statistical Software},  
## year = {2011},  
## volume = {40},  
## number = {1},  
## pages = {1--29},  
## url = {http://www.jstatsoft.org/v40/i01/},  
## }

citation("tidyverse")

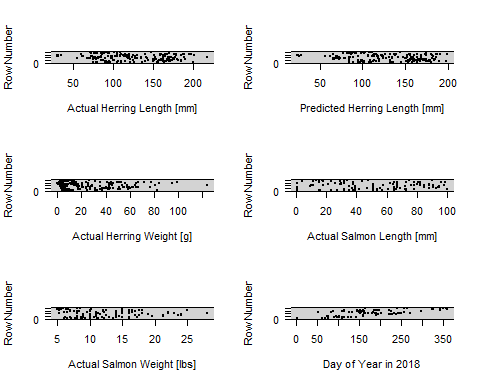
##   
## Wickham et al., (2019). Welcome to the tidyverse. Journal of Open  
## Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686  
##   
## A BibTeX entry for LaTeX users is  
##   
## @Article{,  
## title = {Welcome to the {tidyverse}},  
## author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agostino McGowan and Romain François and Garrett Grolemund and Alex Hayes and Lionel Henry and Jim Hester and Max Kuhn and Thomas Lin Pedersen and Evan Miller and Stephan Milton Bache and Kirill Müller and Jeroen Ooms and David Robinson and Dana Paige Seidel and Vitalie Spinu and Kohske Takahashi and Davis Vaughan and Claus Wilke and Kara Woo and Hiroaki Yutani},  
## year = {2019},  
## journal = {Journal of Open Source Software},  
## volume = {4},  
## number = {43},  
## pages = {1686},  
## doi = {10.21105/joss.01686},  
## }

## Read in and organize data - all variables

#Read from csv file  
herring <- read.csv("HerringOtolithDatabase 15 Jan 2020\_JQ.csv", header=TRUE, fileEncoding="UTF-8-BOM")  
  
#Create a data frame  
herring\_data <- data.frame(sal.id = herring$FishCode,  
 her.id = herring$PreyID,  
 oto.length = herring$AverageLength,  
 oto.width = herring$AverageWidth,  
 oto.perimeter = herring$AveragePerimeter,  
 oto.area = herring$AverageArea,  
 digestion = herring$Digestion,  
 her.length = herring$ActualFishLen,  
 her.length.predict = herring$AVGPredictedLength,  
 her.weight = herring$ActualFishWeight,  
 sal.sp = herring$SalmonSpecies,  
 sal.sex = herring$SalmonSex,  
 sal.length = as.numeric(herring$SalmonLength),  
 sal.weight = herring$SalmonWeight,  
 coll.year = herring$CollectionYear,  
 coll.doy = herring$CollectionDayofYear,  
 lat = herring$Latitude,  
 long = herring$Longitude)  
  
#filter only chinook and data from 2018  
herring\_data\_2018 <- herring\_data %>% filter(herring\_data$sal.sp == "ch" & herring\_data$coll.year == "2018") %>% na.omit()

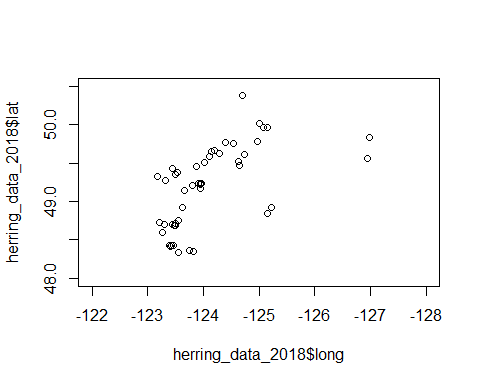
## Example 1 (\*\*\*delete before submitting) -> with all variables

#add column for number of rows  
n <- nrow(herring\_data\_2018)  
herring\_data\_2018$row <- 1:n  
  
par(mfrow = c(3, 2))  
  
#actual herring length - digestion 1, 2, and 3 herring only  
with(herring\_data\_2018, plot(her.length, row, las = 1, type = "n", xlab = "Actual Herring Length [mm]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(her.length, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#predicted herring length - digestion 1, 2, 3, and 4 predicted lengths from linear regression using otolith width  
with(herring\_data\_2018, plot(her.length.predict, row, las = 1, type = "n", xlab = "Predicted Herring Length [mm]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(her.length.predict, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#actual herring weight - digestion 1, 2, and 3 herring only  
with(herring\_data\_2018, plot(her.weight, row, las = 1, type = "n", xlab = "Actual Herring Weight [g]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(her.weight, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#actual salmon length - all salmon that ate herring - note that some numbers are ranges and have weird symbols...  
#where lengths were not recorded, they were calculated using length-weight relationship  
with(herring\_data\_2018, plot(sal.length, row, las = 1, type = "n", xlab = "Actual Salmon Length [mm]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.length, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#actual salmon weight - raw salmon weights no back calculations  
with(herring\_data\_2018, plot(sal.weight, row, las = 1, type = "n", xlab = "Actual Salmon Weight [lbs]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.weight, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#day of year  
with(herring\_data\_2018, plot(coll.doy, row, las = 1, type = "n", xlab = "Day of Year in 2018", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(coll.doy, row, las = 1, type = "p", pch = 19, cex = 0.5))



## Checking our work (***delete before submitting***)

#plot lat/long as continuous variable  
plot(herring\_data\_2018$long, herring\_data\_2018$lat, xlim = c(-122, -128), ylim = c(48, 50.5))

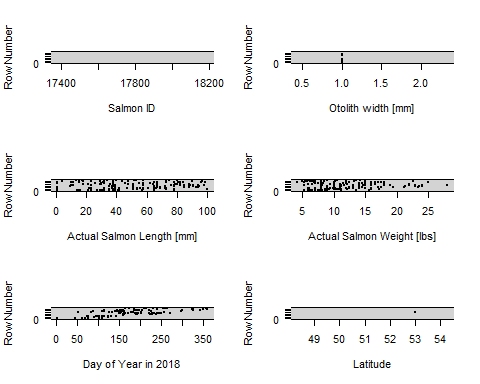


## Read in and organize data

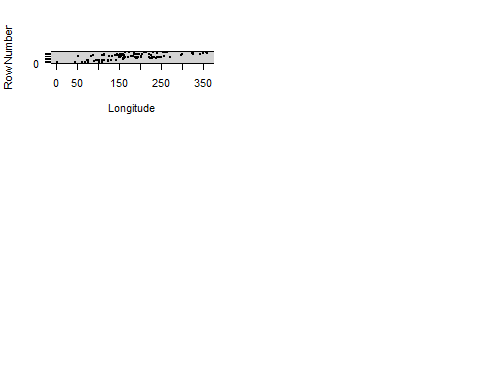
#Read from csv file  
herring <- read.csv("HerringOtolithDatabase 15 Jan 2020\_JQ.csv", header=TRUE, fileEncoding="UTF-8-BOM")  
  
#Create a data frame  
herring\_data <- data.frame(sal.id = herring$FishCode,  
 her.id = herring$PreyID,  
 oto.width = herring$AverageWidth,  
 digestion = herring$Digestion,  
 sal.sp = herring$SalmonSpecies,  
 sal.sex = herring$SalmonSex,  
 sal.length = as.numeric(herring$SalmonLength),  
 sal.weight = herring$SalmonWeight,  
 coll.year = herring$CollectionYear,  
 coll.doy = herring$CollectionDayofYear,  
 lat = herring$Latitude,  
 long = herring$Longitude)  
  
#filter only chinook and data from 2018  
herring\_data\_2018 <- herring\_data %>% filter(herring\_data$sal.sp == "ch" & herring\_data$coll.year == "2018") %>% na.omit()

## Example 1 (\*\*\*delete before submitting)

#add column for number of rows  
n <- nrow(herring\_data\_2018)  
herring\_data\_2018$row <- 1:n  
  
par(mfrow = c(3, 2))  
  
#salmon ID  
with(herring\_data\_2018, plot(sal.id, row, las = 1, type = "n", xlab = "Salmon ID", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.length, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#otolith width  
with(herring\_data\_2018, plot(oto.width, row, las = 1, type = "n", xlab = "Otolith width [mm]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.length, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#actual salmon length - all salmon that ate herring - note that some numbers are ranges and have weird symbols...  
#where lengths were not recorded, they were calculated using length-weight relationship  
with(herring\_data\_2018, plot(sal.length, row, las = 1, type = "n", xlab = "Actual Salmon Length [mm]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.length, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#actual salmon weight - raw salmon weights no back calculations  
with(herring\_data\_2018, plot(sal.weight, row, las = 1, type = "n", xlab = "Actual Salmon Weight [lbs]", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(sal.weight, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#day of year  
with(herring\_data\_2018, plot(coll.doy, row, las = 1, type = "n", xlab = "Day of Year in 2018", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(coll.doy, row, las = 1, type = "p", pch = 19, cex = 0.5))  
  
#lat  
with(herring\_data\_2018, plot(lat, row, las = 1, type = "n", xlab = "Latitude", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(coll.doy, row, las = 1, type = "p", pch = 19, cex = 0.5))

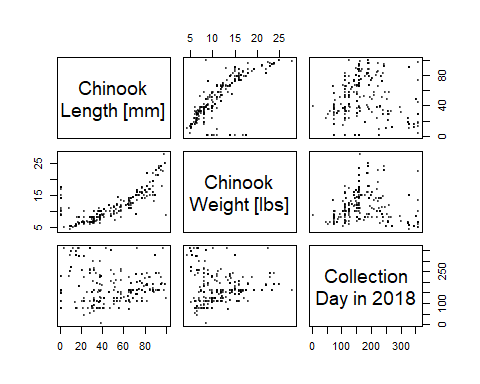


#long  
with(herring\_data\_2018, plot(coll.doy, row, las = 1, type = "n", xlab = "Longitude", ylab = "Row Number"))  
abline (h = seq(1, n, by = 5), col = "lightgrey")  
with(herring\_data\_2018, points(coll.doy, row, las = 1, type = "p", pch = 19, cex = 0.5))



## Example 2 (\*\*\*delete before submitting) -> should include lat longs

#create data frame for pairwise scatter with only 3 explanatory variables - salmon lenght, weight and collection day of year  
herring\_collinearity <- data.frame("sal.length" = herring\_data\_2018$sal.length, "sal.weight" = herring\_data\_2018$sal.weight, "coll.doy" = herring\_data\_2018$coll.doy)  
  
#create pairwise scatterplot with salmon length, weight and day of year  
plot(herring\_collinearity[1:3], cex = 0.5, pch = 19, col = rgb(0, 0, 0, 0.5),   
 labels = c("Chinook\nLength [mm]", "Chinook\nWeight [lbs]", "Collection\nDay in 2018"))



#convert sal.length to numeric  
as.numeric(herring\_data\_2018$sal.length)

## [1] 23 39 31 30 42 41 41 15 15 35 35 35 27 16 38 38 1 1  
## [19] 1 57 14 32 65 65 21 29 78 78 78 35 38 38 38 38 73 73  
## [37] 54 54 51 51 1 23 65 35 1 32 51 42 42 42 42 42 42 42  
## [55] 100 58 78 39 32 23 23 58 23 32 54 51 69 69 54 54 65 65  
## [73] 61 72 72 72 73 73 73 73 22 42 66 37 37 37 47 37 51 56  
## [91] 56 88 88 88 88 88 39 69 69 69 69 69 93 93 93 65 65 65  
## [109] 94 1 1 1 85 85 1 1 1 1 1 1 67 67 91 94 94 37  
## [127] 38 95 95 95 95 95 9 95 78 78 72 72 72 72 72 35 35 65  
## [145] 67 67 67 51 39 39 39 39 39 39 51 82 85 89 82 82 88 88  
## [163] 98 58 88 91 91 91 85 78 78 10 10 58 94 23 11 11 11 11  
## [181] 11 76 76 18 18 18 48 79 82 82 82 82 82 21 21 30 30 30  
## [199] 38 64 64 67 67 67 88 88 29 80 80 80 80 51 36 30 30 30  
## [217] 30 30 30 1 55 55 1 1 32 32 32 46 46 46 46 82 62 62  
## [235] 26 38 58 38 38 32 32 1 1 1 1 1 32 65 31 96 73 73  
## [253] 99 99 84 38 38 81 82 67 31 65 65 73 73 73 78 78 78 78  
## [271] 15 15 11 11 10 23 23 18 18 18 18 15 15 15 15 15 15 44  
## [289] 97 73 4 29 29 29 29 18 29 29 48 48 48 90 90 90 90 90  
## [307] 58 13 17 29 29 58 80 48 29 32 32 32 38 52 34 40 53 75  
## [325] 75

#pairwise correlation coefficients for multiple variables  
print(cor(na.omit(herring\_data\_2018[ ,c("sal.length", "sal.weight", "coll.doy")])), digits = 2)

## sal.length sal.weight coll.doy  
## sal.length 1.000 0.74 -0.068  
## sal.weight 0.737 1.00 -0.103  
## coll.doy -0.068 -0.10 1.000

#variance inflation factors  
print(vif(na.omit(herring\_data\_2018[ ,c("sal.length", "sal.weight", "coll.doy")])))

## sal.length sal.weight coll.doy   
## 2.186418 2.199541 1.010851

## Example 3 (\*\*\*delete before submitting)

#Make a table of salmon ID versus digestion state  
with(herring\_data\_2018, table(sal.id, digestion))

## digestion  
## sal.id 1 2 3 4  
## 17382 0 0 0 1  
## 17383 0 0 1 0  
## 17392 0 0 0 1  
## 17393 0 0 0 1  
## 17394 0 0 0 1  
## 17395 0 0 0 2  
## 17396 0 0 2 0  
## 17397 0 2 0 1  
## 17398 0 0 1 0  
## 17400 0 1 0 0  
## 17402 0 1 0 1  
## 17403 0 0 1 2  
## 17421 0 0 1 0  
## 17425 0 0 0 1  
## 17428 0 1 0 0  
## 17429 0 1 0 1  
## 17430 0 0 0 1  
## 17431 0 0 0 1  
## 17432 0 0 0 3  
## 17433 0 0 1 0  
## 17434 0 0 0 1  
## 17440 0 0 1 2  
## 17441 0 0 1 1  
## 17442 0 1 0 1  
## 17443 0 0 1 1  
## 17445 0 0 0 1  
## 17446 0 0 1 0  
## 17448 0 0 0 1  
## 17449 0 1 0 0  
## 17450 0 0 1 0  
## 17455 1 0 0 0  
## 17470 0 0 1 0  
## 17474 0 0 1 6  
## 17478 0 1 0 0  
## 17479 0 0 0 1  
## 17480 0 0 1 0  
## 17483 0 0 0 1  
## 17485 0 0 1 0  
## 17486 0 0 0 2  
## 17490 0 0 0 1  
## 17493 0 0 0 1  
## 17495 0 0 1 0  
## 17498 0 0 0 1  
## 17501 0 0 0 1  
## 17504 0 0 0 2  
## 17506 0 1 0 1  
## 17535 0 0 2 0  
## 17540 0 1 0 0  
## 17544 0 0 2 1  
## 17550 2 0 1 1  
## 17551 0 0 0 1  
## 17552 0 1 0 0  
## 17553 0 0 1 0  
## 17554 0 0 3 0  
## 17555 0 1 0 0  
## 17559 0 0 1 0  
## 17563 0 1 0 0  
## 17571 0 0 2 0  
## 17575 0 1 3 1  
## 17581 0 1 0 0  
## 17582 0 0 4 1  
## 17591 0 0 1 2  
## 17592 0 1 2 0  
## 17593 0 0 0 1  
## 17594 0 1 0 0  
## 17597 0 2 0 0  
## 17598 0 1 1 0  
## 17601 0 3 3 0  
## 17602 0 2 0 0  
## 17605 0 0 1 0  
## 17606 0 0 1 1  
## 17616 0 0 0 1  
## 17618 0 0 0 1  
## 17622 0 1 4 0  
## 17648 0 1 0 0  
## 17650 0 0 0 1  
## 17664 0 0 0 2  
## 17665 1 0 3 1  
## 17666 0 2 0 0  
## 17680 0 0 0 1  
## 17684 0 0 3 0  
## 17687 0 0 0 1  
## 17688 0 0 6 0  
## 17689 0 1 0 0  
## 17702 0 0 0 1  
## 17704 0 0 1 0  
## 17706 0 0 0 1  
## 17708 0 0 1 1  
## 17710 0 1 0 0  
## 17711 0 0 1 0  
## 17714 0 0 1 0  
## 17715 0 0 1 0  
## 17726 0 0 1 0  
## 17733 0 0 2 1  
## 17737 0 0 1 0  
## 17738 0 0 1 1  
## 17742 0 0 1 1  
## 17743 0 0 1 0  
## 17746 0 0 0 1  
## 17748 0 0 0 1  
## 17749 0 1 3 1  
## 17753 0 0 1 1  
## 17755 0 1 1 1  
## 17756 0 0 0 1  
## 17760 0 0 0 1  
## 17764 0 3 2 0  
## 17766 0 0 1 1  
## 17773 0 0 2 1  
## 17793 0 0 1 0  
## 17805 0 0 1 1  
## 17848 0 1 1 1  
## 17851 0 0 1 1  
## 17864 0 1 0 0  
## 17866 0 0 3 1  
## 17868 0 0 1 0  
## 17872 0 0 0 1  
## 17875 0 0 6 0  
## 17876 0 0 1 0  
## 17878 0 0 0 2  
## 17879 0 1 1 0  
## 17882 0 0 1 0  
## 17883 0 0 2 0  
## 17885 0 0 0 4  
## 17887 0 0 0 1  
## 17888 0 0 1 1  
## 17889 0 0 1 0  
## 17890 0 0 1 0  
## 17891 0 0 0 1  
## 17892 0 0 1 1  
## 17893 0 0 2 0  
## 17895 0 0 0 1  
## 17896 0 0 1 1  
## 17897 0 0 1 0  
## 17898 0 0 1 0  
## 17904 0 1 0 0  
## 17905 0 0 1 0  
## 17925 0 0 0 1  
## 17928 0 0 1 0  
## 17930 0 0 1 1  
## 17931 0 0 1 1  
## 17932 0 0 0 1  
## 17934 0 0 1 1  
## 17940 0 0 1 0  
## 17941 0 0 1 0  
## 17963 0 0 1 0  
## 17976 0 0 0 1  
## 17989 0 1 1 0  
## 17990 0 0 3 0  
## 17994 0 0 2 0  
## 17996 0 0 2 0  
## 17997 0 0 2 0  
## 17999 0 0 2 0  
## 18000 0 0 0 1  
## 18001 1 0 1 0  
## 18002 1 1 2 0  
## 18003 0 4 2 0  
## 18022 0 0 1 0  
## 18030 0 0 0 1  
## 18034 0 0 0 1  
## 18040 0 0 1 0  
## 18041 0 0 4 0  
## 18044 0 1 0 0  
## 18045 0 0 1 1  
## 18046 0 0 1 2  
## 18051 0 0 3 2  
## 18052 0 0 0 1  
## 18067 0 0 1 0  
## 18071 0 0 1 0  
## 18103 0 0 1 1  
## 18107 0 0 0 1  
## 18115 0 0 0 1  
## 18128 0 0 0 1  
## 18129 0 0 0 1  
## 18132 0 0 0 2  
## 18136 0 0 1 0  
## 18177 0 0 0 1  
## 18180 0 0 0 1  
## 18181 0 0 1 0  
## 18182 0 0 0 1  
## 18185 0 0 0 1  
## 18188 0 1 0 1

#Make a table of salmon ID versus sex  
with(herring\_data\_2018, table(sal.id, sal.sex))

## sal.sex  
## sal.id female male  
## 17382 0 1 0  
## 17383 0 1 0  
## 17392 0 1 0  
## 17393 0 1 0  
## 17394 1 0 0  
## 17395 0 2 0  
## 17396 0 2 0  
## 17397 0 3 0  
## 17398 0 1 0  
## 17400 0 1 0  
## 17402 2 0 0  
## 17403 3 0 0  
## 17421 1 0 0  
## 17425 0 1 0  
## 17428 0 0 1  
## 17429 0 2 0  
## 17430 1 0 0  
## 17431 0 1 0  
## 17432 0 3 0  
## 17433 0 1 0  
## 17434 1 0 0  
## 17440 3 0 0  
## 17441 0 2 0  
## 17442 0 2 0  
## 17443 0 0 2  
## 17445 0 0 1  
## 17446 0 1 0  
## 17448 1 0 0  
## 17449 0 1 0  
## 17450 0 1 0  
## 17455 0 0 1  
## 17470 0 1 0  
## 17474 0 7 0  
## 17478 0 0 1  
## 17479 0 0 1  
## 17480 0 1 0  
## 17483 0 0 1  
## 17485 0 0 1  
## 17486 0 0 2  
## 17490 0 1 0  
## 17493 0 1 0  
## 17495 0 1 0  
## 17498 0 1 0  
## 17501 0 0 1  
## 17504 0 2 0  
## 17506 0 2 0  
## 17535 0 0 2  
## 17540 0 1 0  
## 17544 3 0 0  
## 17550 4 0 0  
## 17551 0 0 1  
## 17552 0 1 0  
## 17553 0 0 1  
## 17554 0 3 0  
## 17555 0 1 0  
## 17559 0 1 0  
## 17563 0 0 1  
## 17571 0 0 2  
## 17575 0 5 0  
## 17581 0 1 0  
## 17582 0 0 5  
## 17591 0 3 0  
## 17592 0 0 3  
## 17593 0 0 1  
## 17594 0 0 1  
## 17597 0 2 0  
## 17598 2 0 0  
## 17601 0 0 6  
## 17602 0 2 0  
## 17605 0 0 1  
## 17606 0 2 0  
## 17616 0 1 0  
## 17618 0 1 0  
## 17622 0 5 0  
## 17648 0 0 1  
## 17650 0 1 0  
## 17664 0 2 0  
## 17665 0 5 0  
## 17666 0 2 0  
## 17680 0 1 0  
## 17684 0 3 0  
## 17687 0 1 0  
## 17688 0 0 6  
## 17689 0 1 0  
## 17702 0 0 1  
## 17704 0 0 1  
## 17706 0 1 0  
## 17708 0 2 0  
## 17710 0 0 1  
## 17711 0 1 0  
## 17714 0 0 1  
## 17715 0 1 0  
## 17726 0 0 1  
## 17733 3 0 0  
## 17737 0 1 0  
## 17738 0 2 0  
## 17742 0 0 2  
## 17743 0 1 0  
## 17746 0 0 1  
## 17748 0 0 1  
## 17749 0 0 5  
## 17753 0 0 2  
## 17755 0 0 3  
## 17756 1 0 0  
## 17760 0 1 0  
## 17764 0 5 0  
## 17766 0 0 2  
## 17773 0 0 3  
## 17793 0 1 0  
## 17805 0 2 0  
## 17848 0 0 3  
## 17851 0 2 0  
## 17864 1 0 0  
## 17866 4 0 0  
## 17868 0 0 1  
## 17872 0 1 0  
## 17875 0 6 0  
## 17876 0 0 1  
## 17878 0 0 2  
## 17879 0 0 2  
## 17882 0 0 1  
## 17883 0 2 0  
## 17885 0 4 0  
## 17887 1 0 0  
## 17888 2 0 0  
## 17889 0 1 0  
## 17890 1 0 0  
## 17891 1 0 0  
## 17892 0 0 2  
## 17893 0 2 0  
## 17895 1 0 0  
## 17896 0 0 2  
## 17897 0 1 0  
## 17898 0 0 1  
## 17904 0 1 0  
## 17905 0 1 0  
## 17925 0 0 1  
## 17928 0 0 1  
## 17930 2 0 0  
## 17931 0 0 2  
## 17932 0 0 1  
## 17934 0 0 2  
## 17940 0 0 1  
## 17941 0 0 1  
## 17963 0 1 0  
## 17976 0 1 0  
## 17989 2 0 0  
## 17990 0 0 3  
## 17994 0 2 0  
## 17996 0 2 0  
## 17997 0 0 2  
## 17999 0 0 2  
## 18000 0 0 1  
## 18001 0 0 2  
## 18002 0 4 0  
## 18003 0 0 6  
## 18022 0 1 0  
## 18030 1 0 0  
## 18034 0 0 1  
## 18040 0 1 0  
## 18041 4 0 0  
## 18044 0 0 1  
## 18045 0 0 2  
## 18046 3 0 0  
## 18051 0 5 0  
## 18052 0 0 1  
## 18067 0 0 1  
## 18071 0 0 1  
## 18103 2 0 0  
## 18107 0 0 1  
## 18115 0 1 0  
## 18128 0 1 0  
## 18129 0 1 0  
## 18132 2 0 0  
## 18136 1 0 0  
## 18177 0 1 0  
## 18180 0 0 1  
## 18181 0 0 1  
## 18182 0 1 0  
## 18185 0 0 1  
## 18188 0 2 0

## Example 4 (\*\*\*delete before submitting)

#Make a table of 0 values and NAs  
table(herring\_data\_2018 > 0, useNA = "always")

## Warning in Ops.factor(left, right): '>' not meaningful for factors  
  
## Warning in Ops.factor(left, right): '>' not meaningful for factors  
  
## Warning in Ops.factor(left, right): '>' not meaningful for factors

##   
## FALSE TRUE <NA>   
## 321 2929 975