Processing Script

William Norfolk

10/26/2019

This Rmd file loads, cleans, and organizes data variables collected from the Fall 2019 EHSC 8310 water sampling.

Start by loading all required libraries.

library(readxl)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages ----------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v readr 1.3.1  
## v tibble 2.1.3 v purrr 0.3.2  
## v tidyr 0.8.3 v stringr 1.4.0  
## v ggplot2 3.2.1 v forcats 0.4.0

## -- Conflicts -------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(forcats)  
library(ggthemes)  
library(knitr)  
library(naniar)  
library(visdat)  
library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following object is masked from 'package:purrr':  
##   
## compact

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

Next load the data. The path in the code chunk below is relative to the raw data file placed in this project.

micro\_raw\_data <- readxl::read\_excel("../Data/raw\_data.xlsx")

Take a look at the data and variables to see what you are working with.

glimpse(micro\_raw\_data)

## Observations: 30  
## Variables: 27  
## $ `Sample ID` <chr> ...  
## $ `Rain Event` <chr> ...  
## $ Date <chr> ...  
## $ `Collection Time` <chr> ...  
## $ `Days Sine Last Rainfaill Greater than 2 inches (amount of rainfall)` <chr> ...  
## $ `Days Since Last Rainfall (amount of rainfall (in))` <chr> ...  
## $ `Ambient Air Temp High (F)` <dbl> ...  
## $ `Ambient Air Temp High (C)` <dbl> ...  
## $ `Ambient Air Temp Low (F)` <dbl> ...  
## $ `Ambient Air Temp Low (C)` <dbl> ...  
## $ `Water Temp (C)` <dbl> ...  
## $ `Conductivity (mS/cm)` <dbl> ...  
## $ pH <dbl> ...  
## $ `Source(s) of Impact` <lgl> ...  
## $ `Salmonella Volume` <chr> ...  
## $ `DNA Filter Volume` <chr> ...  
## $ `E. coli 1mL cfu (1)` <chr> ...  
## $ `E. coli 1mL cfu (2)` <chr> ...  
## $ `E. coli 10mL cfu (10)` <chr> ...  
## $ `E. coli 10mL cfu (2)` <chr> ...  
## $ `E coli limit of detection` <dbl> ...  
## $ `Final CFU/100 ml` <chr> ...  
## $ `Exceeds EPA STV (410/100mL)` <chr> ...  
## $ `XLT-4 Salmonella` <chr> ...  
## $ `Chromagar Confirmed Salmonella` <chr> ...  
## $ HF183 <lgl> ...  
## $ Rum2Bac <lgl> ...

Lots of variables here, mostly just information that needed to be recorded but is not particularly useful for data analysis so we will remove some variables that are not needed for visualization/statistics for simplicity sake. Also a number of the variables loaded into the data set with strange additions so we will remove them to reduce typos.

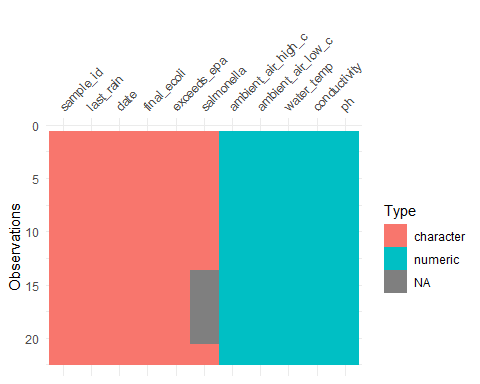
d = micro\_raw\_data %>% dplyr::rename(sample\_id = `Sample ID`,   
 last\_rain = `Days Sine Last Rainfaill Greater than 2 inches (amount of rainfall)`,  
 collection\_time = `Collection Time`,  
 ambient\_air\_high\_c = `Ambient Air Temp High (C)`,  
 ambient\_air\_low\_c = `Ambient Air Temp Low (C)`,  
 water\_temp = `Water Temp (C)`,  
 conductivity = `Conductivity (mS/cm)`,  
 final\_ecoli = `Final CFU/100 ml`,  
 exceeds\_epa = `Exceeds EPA STV (410/100mL)`,  
 salmonella = `Chromagar Confirmed Salmonella`,  
 date = Date,  
 ph = pH)  
d = d %>% filter(sample\_id != "NA")

d = d %>% dplyr::select(sample\_id, last\_rain, date, ambient\_air\_high\_c, ambient\_air\_low\_c, water\_temp, conductivity, ph, final\_ecoli, exceeds\_epa, salmonella )  
  
glimpse(d)

## Observations: 22  
## Variables: 11  
## $ sample\_id <chr> "P1 (Plaza)", "P2 (Plaza)", "PR1 (Hospital)...  
## $ last\_rain <chr> "38 (2.53)", "38 (2.53)", "38 (2.53)", "38 ...  
## $ date <chr> "9.11.19", "9.11.19", "9.11.19", "9.11.19",...  
## $ ambient\_air\_high\_c <dbl> 35.55556, 35.55556, 35.55556, 35.55556, 35....  
## $ ambient\_air\_low\_c <dbl> 21.11111, 21.11111, 21.11111, 21.11111, 21....  
## $ water\_temp <dbl> 21.52, 22.93, 22.85, 23.82, 34.50, 26.50, 2...  
## $ conductivity <dbl> 0.119, 0.077, 0.139, 0.212, 0.049, 0.062, 0...  
## $ ph <dbl> 7.60, 8.09, 10.46, 8.45, 10.59, 8.90, 7.82,...  
## $ final\_ecoli <chr> "40", "425", "6450", "4300", "2.5", "2", "1...  
## $ exceeds\_epa <chr> "No", "Yes", "Yes", "Yes", "No", "No", "No"...  
## $ salmonella <chr> "N/A", "yes", "N/A", "N/A", "N/A", "no", "n...

Now we can check what kind of data and observations we are dealing with.

vis\_dat(d)



We may want to re-code sample IDs.

as.factor(d$sample\_id)

## [1] P1 (Plaza) P2 (Plaza) PR1 (Hospital) PR2 (Hospital)  
## [5] VM1 (Vet Med) VM2 (Vet Med) P1 (Plaza) P2 (Plaza)   
## [9] PR1 (Hospital) PR2 (Hospital) VM1 (Vet Med) VM2 (Vet Med)   
## [13] T1 (Tanyard) P1 (Plaza) P2 (Plaza) PR1 (Hospital)  
## [17] PR2 (Hospital) VM1 (Vet Med) VM2 (Vet Med) T1 (Tanyard)   
## [21] PR2 (Hospital) T1 (Tanyard)   
## 7 Levels: P1 (Plaza) P2 (Plaza) PR1 (Hospital) ... VM2 (Vet Med)

d$sample\_id = fct\_recode(d$sample\_id, "P1" = "P1 (Plaza)")  
d$sample\_id = fct\_recode(d$sample\_id, "P2" = "P2 (Plaza)")  
d$sample\_id = fct\_recode(d$sample\_id, "H1" = "PR1 (Hospital)")  
d$sample\_id = fct\_recode(d$sample\_id, "H2" = "PR2 (Hospital)")  
d$sample\_id = fct\_recode(d$sample\_id, "VM1" = "VM1 (Vet Med)")  
d$sample\_id = fct\_recode(d$sample\_id, "VM2" = "VM2 (Vet Med)")  
d$sample\_id = fct\_recode(d$sample\_id, "T1" = "T1 (Tanyard)")

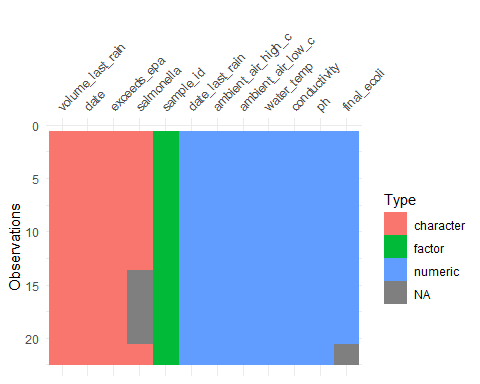
We may also want to re-code salmonella.

Change Varibles. Make E. coli numeric. Make last rain numeric.

#Make E.coli numeric  
d$final\_ecoli = as.numeric(d$final\_ecoli)

## Warning: NAs introduced by coercion

#Make last rain numeric.   
d = d %>% separate(last\_rain, c("date\_last\_rain", "volume\_last\_rain"), sep = 2)  
d$date\_last\_rain = as.numeric(d$date\_last\_rain)  
vis\_dat(d)



saveRDS(d, file = "../Data/processed\_data/data\_cleaned.rds")