Water Quality Data Manipulation

Mary Glover

Objective

This document walks through data cleaning done by environmental data analysis students at William Peace University. The data was obtained from a records request from the City of Raleigh for the historic water quality records from 2008 to 2023.

Data clean-up

Data from the City of Raleigh was saved as a csv file to load into R. Here, it is loaded into R as object `wq` and we can see the first 20 rows.

```
wq <- read.csv("../data/raleigh_wq_2008_2023.csv")
head(wq, 20)
```

```
##    Site       Date Time        Parameter Result PQL        Unit
## 1   BB2 2008-09-30 9:52          Calcium   <NA>  NA        mg/L
## 2   BB2 2008-09-30 9:52   Hardness_total   <NA>  NA        mg/L
## 3   BB2 2008-09-30 9:52        Magnesium   <NA>  NA        mg/L
## 4   BB2 2008-09-30 9:52         Salinity   <NA>  NA         ppt
## 5   BB2 2008-09-30 9:52 Phosphorus_total  <0.05  NA        mg/L
## 6   BB2 2008-09-30 9:52              NH3  <0.02  NA        mg/L
## 7   BB2 2008-09-30 9:52           Copper <0.005  NA        mg/L
## 8   BB2 2008-09-30 9:52           E_coli    236  NA         MPN
## 9   BB2 2008-09-30 9:52     Conductivity  106.8  NA          uS
## 10  BB2 2008-09-30 9:52   do_percent_sat   88.5  NA percent_sat
## 11  BB2 2008-09-30 9:52      Temperature   19.9  NA           C
## 12  BB2 2008-09-30 9:52           do_mgl   8.10  NA        mg/L
## 13  BB2 2008-09-30 9:52               pH   6.50  NA    std_unit
## 14  BB2 2008-09-30 9:52        Turbidity    4.9  NA         NTU
## 15  BB2 2008-09-30 9:52              TSS    2.2  NA        mg/L
## 16  BB2 2008-09-30 9:52   Nitrogen_total   0.74  NA        mg/L
## 17  BB2 2008-09-30 9:52          NO2_NO3   0.41  NA        mg/L
## 18  BB2 2008-09-30 9:52              TKN   0.33  NA        mg/L
## 19  BB2 2008-09-30 9:52             Zinc  0.029  NA        mg/L
## 20 BBS3 2008-09-30 8:30          Calcium   <NA>  NA        mg/L
```

To clean up the data, we will load in the necessary packages.

```
library(dplyr)
library(stringr)
library(tidyr)
```

In the first few rows, you can see that some of the results are recorded as less than a value (ex. <0.05). To analyze the data, all results values need to be numeric, so all instances of "<" in the Result column are recoded as "0".

```
wq$Result <- ifelse(grepl("<", wq$Result), "0", wq$Result)
```

In some cases, results are recorded as not detected or ND. These are also made into 0's in the Result column. The resulting data table is saved as a new object `wq.det`.

```
wq.det <- wq |>
  mutate(Result = recode(Result, ND = '0'))
```

The Result column can then be converted to a numeric class, since now all the values are numeric.

```
wq.det$Result<-as.numeric(wq.det$Result)
```

```
## Warning: NAs introduced by coercion
```

```
summary(wq.det)
```

```
##      Site               Date               Time            Parameter
##  Length:21145       Length:21145       Length:21145       Length:21145
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      Result               PQL            Unit
##  Min.   :     0.00   Min.   :0.001   Length:21145
##  1st Qu.:     0.06   1st Qu.:0.040   Class :character
##  Median :     4.40   Median :0.100   Mode  :character
##  Mean   :    69.49   Mean   :0.644
##  3rd Qu.:    20.00   3rd Qu.:1.000
##  Max.   :155310.00   Max.   :9.200
##  NA's   :1123        NA's   :18233
```

Next, to make for easier analysis, we wanted to make each characteristic measured it's own column, instead of having one result column. This is the "tidy" data format. We can use the `dplyr` package and the `pivot_wider` function. In the following code, the columns Unit and PQL are removed. Then we "widen" the data with `pivot_wider`. The column names will come from the Parameter column and the data in the cells from the Result colum. This means that each different Parameter will have it's own column.

```
wq.det |>
  dplyr::select(-Unit, -PQL)|>
  pivot_wider(id_cols = c(Site, Date, Time), names_from = Parameter, values_from = Result)
```

```
## Warning: Values from `Result` are not uniquely identified; output will contain
## list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
```

```
## * Use the following dplyr code to identify duplicates.
##   {data} %>%
##   dplyr::group_by(Site, Date, Time, Parameter) %>%
##   dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%
##   dplyr::filter(n > 1L)
```

```
## # A tibble: 1,345 x 22
##    Site  Date   Time  Calcium Hardness_total Magnesium Salinity Phosphorus_total
##    <chr> <chr>  <chr> <list>  <list>         <list>    <list>   <list>
##  1 BB2   2008-~ 9:52  <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  2 BBS3  2008-~ 8:30  <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  3 BDB1  2008-~ 10:16 <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  4 CC4   2008-~ 9:40  <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  5 CC5   2008-~ 9:05  <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  6 HC7   2008-~ 11:06 <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  7 HSC6  2008-~ 11:54 <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  8 LBC8  2008-~ 11:20 <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
##  9 MC10  2008-~ 10:42 <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
## 10 MC9   2008-~ 9:30  <dbl>   <dbl [1]>      <dbl [1]> <dbl>    <dbl [1]>
## # i 1,335 more rows
## # i 14 more variables: NH3 <list>, Copper <list>, E_coli <list>,
## #   Conductivity <list>, do_percent_sat <list>, Temperature <list>,
## #   do_mgl <list>, pH <list>, Turbidity <list>, TSS <list>,
## #   Nitrogen_total <list>, NO2_NO3 <list>, TKN <list>, Zinc <list>
```

When we run this, we see that there is an error, indicating that there is a duplicate. For one of the Parameters, there is not a one unique row for a specific Site, Date, and Time. To find the duplicate, we can use the suggested code.

```
wq.det %>%
  dplyr::group_by(Site, Date, Time, Parameter) %>%
  dplyr::summarise(n = dplyr::n()) %>%
  dplyr::filter(n > 1)
```

```
## `summarise()` has grouped output by 'Site', 'Date', 'Time'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 1 x 5
## # Groups:   Site, Date, Time [1]
##   Site  Date       Time  Parameter     n
##   <chr> <chr>      <chr> <chr>     <int>
## 1 MC10  2022-12-13 10:29 Salinity      2
```

You can see that at Site MC10 on December 22, there are 2 rows for Salinity. Let's filter to only show those MC10 at that specific time point.

```
wq.det %>%
  filter(Site=='MC10', Time == "10:29")
```

```
##   Site       Date  Time      Parameter Result PQL        Unit
## 1 MC10 2022-12-13 10:29 do_percent_sat  91.00  NA percent_sat
## 2 MC10 2022-12-13 10:29   Conductivity  56.20  NA          uS
```

```
## 3 MC10 2022-12-13 10:29         do_mgl  10.70  NA        mg/L
## 4 MC10 2022-12-13 10:29    Temperature   8.20  NA           C
## 5 MC10 2022-12-13 10:29             pH   6.74  NA    std_unit
## 6 MC10 2022-12-13 10:29       Salinity   0.04  NA         ppt
## 7 MC10 2022-12-13 10:29       Salinity   0.04  NA         ppt
```

You can see that ere are 2 Salinity values and that they are the same. So we will need to get rid of the duplicates. We can do that by filltering for distinct rows using `distinct`

```
wq.det |>
  distinct() |> # get rid of one duplicate
  dplyr::select(-Unit, -PQL) |>
  pivot_wider(id_cols = c(Site, Date, Time), names_from = Parameter, values_from = Result)
```

```
## # A tibble: 1,345 x 22
##    Site  Date   Time  Calcium Hardness_total Magnesium Salinity Phosphorus_total
##    <chr> <chr>  <chr>   <dbl>          <dbl>     <dbl>    <dbl>            <dbl>
##  1 BB2   2008-~ 9:52       NA             NA        NA       NA             0
##  2 BBS3  2008-~ 8:30       NA             NA        NA       NA             0.05
##  3 BDB1  2008-~ 10:16      NA             NA        NA       NA             0
##  4 CC4   2008-~ 9:40       NA             NA        NA       NA             0
##  5 CC5   2008-~ 9:05       NA             NA        NA       NA             0
##  6 HC7   2008-~ 11:06      NA             NA        NA       NA             0
##  7 HSC6  2008-~ 11:54      NA             NA        NA       NA             0
##  8 LBC8  2008-~ 11:20      NA             NA        NA       NA             0
##  9 MC10  2008-~ 10:42      NA             NA        NA       NA             0
## 10 MC9   2008-~ 9:30       NA             NA        NA       NA             0.12
## # i 1,335 more rows
## # i 14 more variables: NH3 <dbl>, Copper <dbl>, E_coli <dbl>,
## #   Conductivity <dbl>, do_percent_sat <dbl>, Temperature <dbl>, do_mgl <dbl>,
## #   pH <dbl>, Turbidity <dbl>, TSS <dbl>, Nitrogen_total <dbl>, NO2_NO3 <dbl>,
## #   TKN <dbl>, Zinc <dbl>
```

This gives us the output we want with one column fro every parameter.

Dealing with the units

However, in this way, we have gotten rid of the units. We want to add these back in the names of the columns.

But first, we noticed a few other things to clean up with the units.

In one case, the units do not match up with the parameter for DO. The Parameter indicates unit of percent_sat and Unit as mg/L.

```
wq |>
  filter(Parameter == 'do_percent_sat' , Unit == 'mg/L')
```

```
##    Site       Date  Time      Parameter Result PQL Unit
## 1   HC7 2008-09-30 11:06 do_percent_sat   82.3  NA mg/L
## 2   HC7 2008-12-16 11:00 do_percent_sat   92.5  NA mg/L
## 3   HC7 2009-03-17 11:45 do_percent_sat   98.5  NA mg/L
## 4   HC7 2009-06-16 13:30 do_percent_sat   93.6  NA mg/L
```

```
## 5   HC7 2009-09-15 10:30 do_percent_sat     80  NA mg/L
## 6   HC7 2009-12-15 10:40 do_percent_sat     97  NA mg/L
## 7   HC7 2010-03-16 10:35 do_percent_sat    103  NA mg/L
## 8   HC7 2010-06-15 10:35 do_percent_sat   80.5  NA mg/L
## 9   HC7 2010-09-14 10:30 do_percent_sat   71.5  NA mg/L
## 10  HC7 2010-12-21 10:55 do_percent_sat  101.6  NA mg/L
## 11  HC7 2011-03-15 10:15 do_percent_sat   92.8  NA mg/L
## 12  HC7 2011-06-07 10:25 do_percent_sat   71.8  NA mg/L
## 13  HC7 2011-09-20 10:15 do_percent_sat   79.5  NA mg/L
## 14  HC7 2011-12-19 10:15 do_percent_sat   93.9  NA mg/L
## 15  HC7 2012-03-20  9:40 do_percent_sat     92  NA mg/L
## 16  HC7 2012-06-19  8:56 do_percent_sat   77.9  NA mg/L
## 17  HC7 2012-09-19  9:05 do_percent_sat   84.9  NA mg/L
## 18  HC7 2012-12-18  9:40 do_percent_sat   86.3  NA mg/L
## 19  HC7 2013-03-20 10:00 do_percent_sat   99.3  NA mg/L
## 20  HC7 2013-06-18  9:45 do_percent_sat   81.9  NA mg/L
## 21  HC7 2013-09-24 10:35 do_percent_sat   91.8  NA mg/L
## 22  HC7 2013-12-17  9:55 do_percent_sat   96.3  NA mg/L
## 23  HC7 2014-03-20 10:15 do_percent_sat  110.2  NA mg/L
## 24  HC7 2014-06-10  9:30 do_percent_sat   80.4  NA mg/L
## 25  HC7 2014-09-23  9:36 do_percent_sat   74.5  NA mg/L
## 26  HC7 2014-12-16  9:25 do_percent_sat   94.6  NA mg/L
## 27  HC7 2015-03-24 10:25 do_percent_sat    102  NA mg/L
## 28  HC7 2015-06-24  9:10 do_percent_sat     77  NA mg/L
## 29  HC7 2015-09-30 10:20 do_percent_sat   83.6  NA mg/L
## 30  HC7 2015-12-14  9:50 do_percent_sat   91.3  NA mg/L
## 31  HC7 2016-03-22 10:10 do_percent_sat  105.9  NA mg/L
## 32  HC7 2016-06-07 10:53 do_percent_sat   88.6  NA mg/L
## 33  HC7 2016-09-14 10:30 do_percent_sat   81.1  NA mg/L
## 34  HC7 2016-12-14 10:47 do_percent_sat   92.8  NA mg/L
## 35  HC7 2017-03-09 10:50 do_percent_sat     99  NA mg/L
## 36  HC7 2017-06-29 10:45 do_percent_sat   93.7  NA mg/L
## 37  HC7 2017-09-14 10:49 do_percent_sat   87.7  NA mg/L
## 38  HC7 2018-01-24 10:28 do_percent_sat   97.6  NA mg/L
## 39  HC7 2018-03-20 10:50 do_percent_sat   87.7  NA mg/L
## 40  HC7 2018-06-12 10:21 do_percent_sat   85.4  NA mg/L
## 41  HC7 2018-09-25 10:20 do_percent_sat   67.7  NA mg/L
## 42  HC7 2018-12-20 10:12 do_percent_sat  101.2  NA mg/L
## 43  HC7 2019-03-19 10:15 do_percent_sat  106.9  NA mg/L
## 44  HC7 2019-06-12 10:20 do_percent_sat   86.6  NA mg/L
## 45  HC7 2019-09-12 10:35 do_percent_sat   84.3  NA mg/L
## 46  HC7 2019-12-10 10:15 do_percent_sat  102.1  NA mg/L
## 47  HC7 2020-03-10 10:05 do_percent_sat   <NA>  NA mg/L
## 48  HC7 2020-06-16 10:02 do_percent_sat   96.7  NA mg/L
## 49  HC7 2020-10-27 10:20 do_percent_sat   84.6  NA mg/L
## 50  HC7 2020-12-01 10:00 do_percent_sat   98.2  NA mg/L
## 51  HC7 2021-03-02  9:45 do_percent_sat  114.2  NA mg/L
## 52  HC7 2021-06-01 11:02 do_percent_sat   94.5  NA mg/L
```

This only is an issue in HC7. All DO units reported as mg/L. Below is a summary of HC7 for DO. IT appears mg/L was indicated as unit for all measurments until 2021.

```
wq |>
  filter(Site == 'HC7') |>
  filter(grepl("do", Parameter)) |>
  separate(Date, into = c('year', 'month', 'day'), sep = '-')|>
  group_by(Parameter, Unit, year)|>
  tally() |>
  arrange(year)|>
  print(n=33)
```

```
## # A tibble: 33 x 4
## # Groups:   Parameter, Unit [3]
##    Parameter      Unit        year      n
##    <chr>          <chr>       <chr> <int>
##  1 do_mgl         mg/L        2008      2
##  2 do_percent_sat mg/L        2008      2
##  3 do_mgl         mg/L        2009      4
##  4 do_percent_sat mg/L        2009      4
##  5 do_mgl         mg/L        2010      4
##  6 do_percent_sat mg/L        2010      4
##  7 do_mgl         mg/L        2011      4
##  8 do_percent_sat mg/L        2011      4
##  9 do_mgl         mg/L        2012      4
## 10 do_percent_sat mg/L        2012      4
## 11 do_mgl         mg/L        2013      4
## 12 do_percent_sat mg/L        2013      4
## 13 do_mgl         mg/L        2014      4
## 14 do_percent_sat mg/L        2014      4
## 15 do_mgl         mg/L        2015      4
## 16 do_percent_sat mg/L        2015      4
## 17 do_mgl         mg/L        2016      4
## 18 do_percent_sat mg/L        2016      4
## 19 do_mgl         mg/L        2017      3
## 20 do_percent_sat mg/L        2017      3
## 21 do_mgl         mg/L        2018      5
## 22 do_percent_sat mg/L        2018      5
## 23 do_mgl         mg/L        2019      4
## 24 do_percent_sat mg/L        2019      4
## 25 do_mgl         mg/L        2020      4
## 26 do_percent_sat mg/L        2020      4
## 27 do_mgl         mg/L        2021      4
## 28 do_percent_sat mg/L        2021      2
## 29 do_percent_sat percent_sat 2021      2
## 30 do_mgl         mg/L        2022      4
## 31 do_percent_sat percent_sat 2022      4
## 32 do_mgl         mg/L        2023      4
## 33 do_percent_sat percent_sat 2023      4
```

There is also some discrepancy in the units for E. coli.

First, it appears that E. coli has three different units measured.

```
wq |>
  filter(Parameter == 'E_coli') |>
  distinct(Unit)
```

```
##        Unit
## 1       MPN
## 2 MPN/100mL
## 3 CFU/100mL
```

When looking at the data, there is never an instance where E. coli was measured more than once on a specific day, but how it was measured varied.

Here, we look at what units were used by year. We converted the date to a date class in R using the `lubridate` package

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
wq |>
  mutate(Date = ymd(Date)) |>
  mutate(Year = year(Date))|>
  filter(Parameter == 'E_coli')|>
  group_by(Year, Unit)|>
  summarize(n_na = sum(is.na(Result)), n = n())
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 19 x 4
## # Groups:   Year [16]
##     Year Unit       n_na     n
##    <dbl> <chr>     <int> <int>
##  1  2008 MPN           0    36
##  2  2009 MPN           0    72
##  3  2010 MPN           0    72
##  4  2011 MPN           0    72
##  5  2012 MPN           0    72
##  6  2013 MPN          18    72
##  7  2014 MPN           1    72
##  8  2015 MPN           1    72
##  9  2016 MPN           0    72
## 10  2017 MPN           0    54
## 11  2018 MPN           0    90
## 12  2019 MPN           0    72
## 13  2020 MPN           0    36
## 14  2020 MPN/100mL     0    36
## 15  2021 CFU/100mL     0    18
## 16  2021 MPN/100mL     1    36
## 17  2022 CFU/100mL     0    20
## 18  2022 MPN/100mL     0    56
## 19  2023 MPN/100mL     0    80
```

You can see that up until 2020, unit was recorded as MPN. After that, it was recorded as MPN/100mL. These units will need to match. Also, for part of 2021 and 2022, E. coli was measured as CFU/100mL. To keep this straight, we will have separate columns for E. coli for CFU vs. MPN measurements. We will use the MPN measurements for analysis.

To update the unit column, update Unit to 'percent_sat' whenever Parameter is "do_percent_sat" and also recode MPN to MPN/100mL to match other unit.

```
wq_units <- wq.det |>
  mutate(Unit = case_when(Parameter == 'do_percent_sat' ~ 'percent_sat',
                          Unit == 'MPN' ~ 'MPN/100mL',
                          .default = Unit))
```

We will also use the `distinct` function to remove the one duplicate row from HC7 Salinity.

```
wq_units <- distinct(wq_units)
```

We will now create a new column called "new_name" where we paste the parameter and the unit together.

```
wq_units <- wq_units|>
  mutate(new_name = paste(Parameter, Unit, sep = '_'))
head(wq_units)
```

```
##   Site       Date Time        Parameter Result PQL Unit              new_name
## 1  BB2 2008-09-30 9:52          Calcium     NA  NA mg/L          Calcium_mg/L
## 2  BB2 2008-09-30 9:52   Hardness_total     NA  NA mg/L   Hardness_total_mg/L
## 3  BB2 2008-09-30 9:52        Magnesium     NA  NA mg/L        Magnesium_mg/L
## 4  BB2 2008-09-30 9:52         Salinity     NA  NA  ppt          Salinity_ppt
## 5  BB2 2008-09-30 9:52 Phosphorus_total      0  NA mg/L Phosphorus_total_mg/L
## 6  BB2 2008-09-30 9:52              NH3      0  NA mg/L              NH3_mg/L
```

Next, we try again to "pivot" the table to make eack parameter it's own column and remove the Unit and PQL columns.

```
wq_units <- wq_units |>
  mutate(new_name = paste(Parameter, Unit, sep = '_')) |>
  select(-Unit, -PQL) |>
  pivot_wider(id_cols = c(Site, Date, Time), names_from = new_name, values_from = Result)
head(wq_units)
```

```
## # A tibble: 6 x 24
##   Site  Date       Time  `Calcium_mg/L` `Hardness_total_mg/L` `Magnesium_mg/L`
##   <chr> <chr>      <chr>          <dbl>                 <dbl>            <dbl>
## 1 BB2   2008-09-30 9:52              NA                    NA               NA
## 2 BBS3  2008-09-30 8:30              NA                    NA               NA
## 3 BDB1  2008-09-30 10:16             NA                    NA               NA
## 4 CC4   2008-09-30 9:40              NA                    NA               NA
## 5 CC5   2008-09-30 9:05              NA                    NA               NA
## 6 HC7   2008-09-30 11:06             NA                    NA               NA
## # i 18 more variables: Salinity_ppt <dbl>, `Phosphorus_total_mg/L` <dbl>,
## #   `NH3_mg/L` <dbl>, `Copper_mg/L` <dbl>, `E_coli_MPN/100mL` <dbl>,
## #   Conductivity_uS <dbl>, do_percent_sat_percent_sat <dbl>,
```

```
## #   Temperature_C <dbl>, `do_mgl_mg/L` <dbl>, pH_std_unit <dbl>,
## #   Turbidity_NTU <dbl>, `TSS_mg/L` <dbl>, `Nitrogen_total_mg/L` <dbl>,
## #   `NO2_NO3_mg/L` <dbl>, `TKN_mg/L` <dbl>, `Zinc_mg/L` <dbl>,
## #   Salinity_uS <dbl>, `E_coli_CFU/100mL` <dbl>
```

Now, we have our data table in the format we want. However, to make it easier to work with we will rename a couple of the columns. For DO, since the unit was in the Parameter name, the unit is in the column name twice. We will also rename the pH column to just pH instead of pH_std_unit.

```
wq_units <- wq_units |>
  rename(do_percent_sat = do_percent_sat_percent_sat, do_mg_L = `do_mgl_mg/L`, pH = pH_std_unit)
```

We will also replace any "/" with "_" to make the data easier to work with in R using the `replace_with` function

```
wq_units<- rename_with(wq_units, ~ gsub("/", '_', .x))
```

Lastly, we will save the table as "raleigh_wq_clean-units.csv"

```
write.csv(wq_units, "data/raleigh_wq_clean-units.csv", row.names = F)
```

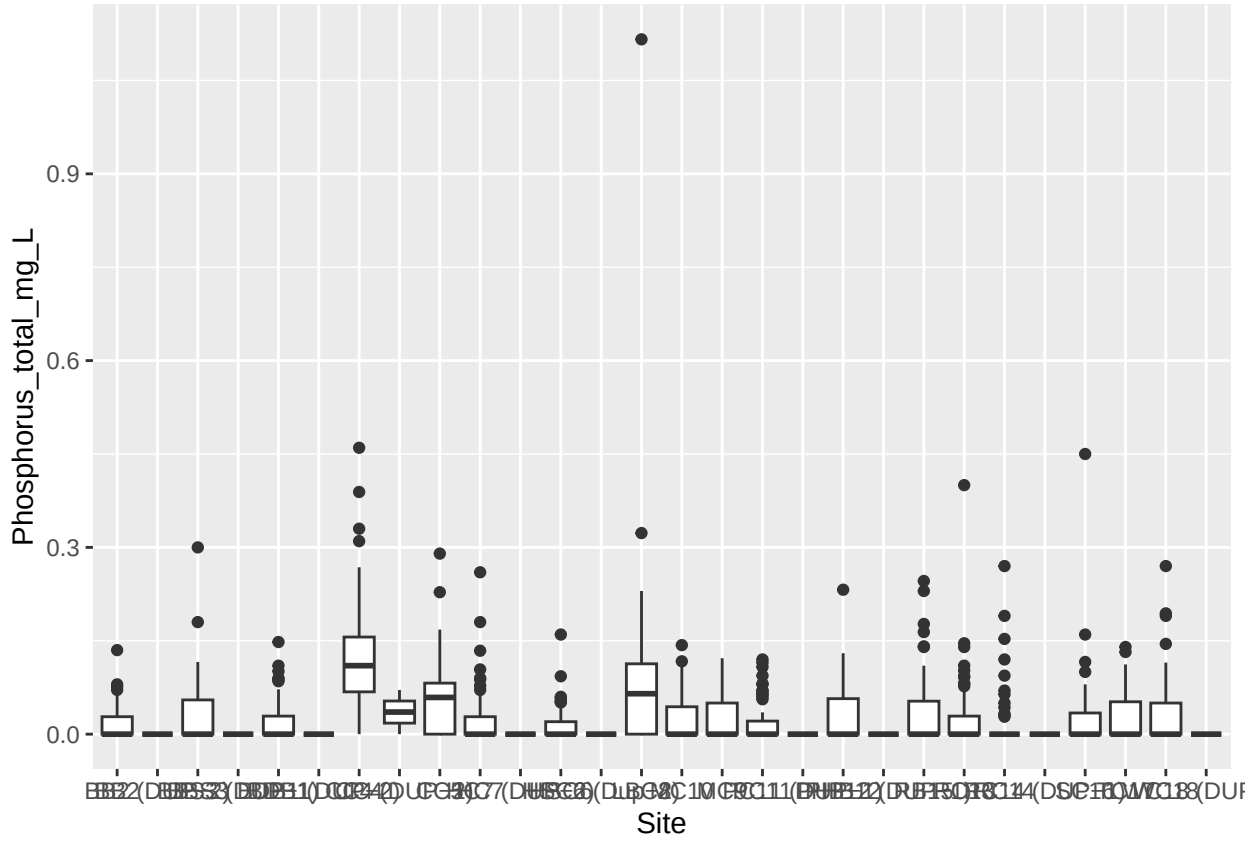
Exploring the Data

Now the data is ready to explore in R!

For example, we could plot a boxplot of the Phosphorus values at diffferent sites. We will use the package `ggplot2`

```
library(ggplot2)
ggplot(wq_units, aes(x = Site, y = Phosphorus_total_mg_L)) +
  geom_boxplot()
```

```
## Warning: Removed 235 rows containing non-finite values (`stat_boxplot()`).
```



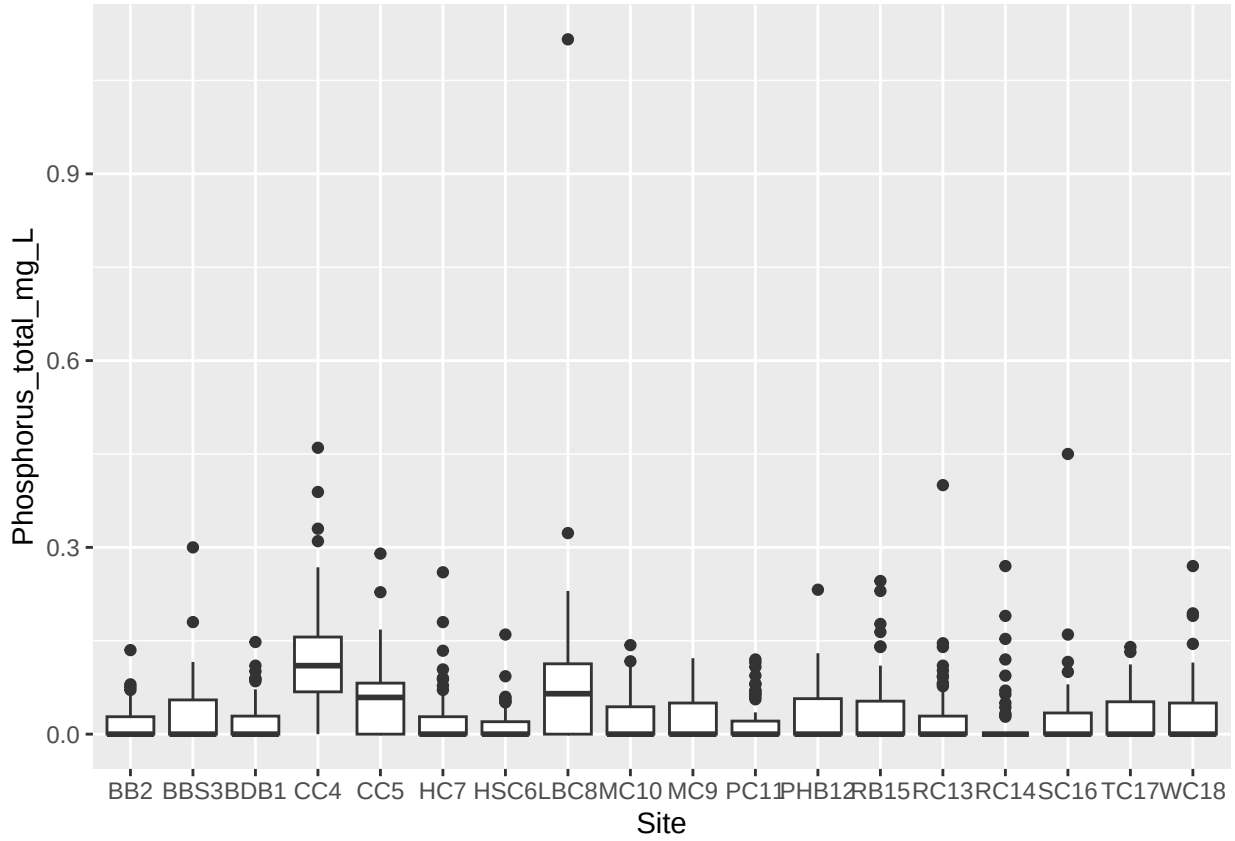
One thing you can see is there are multiple DUP sites. to see without these, we can remove thme using the `filter` function.

```
wq_dupsrm<- wq_units |>
  filter(!grepl("DUP",Site)) |> # gets rid of duplicate sites
  filter(!grepl("Dup",Site))
```

Now we will plot again with the new data with the duplicates sites removed.

```
ggplot(wq_dupsrm, aes(x = Site, y = Phosphorus_total_mg_L)) +
  geom_boxplot()
```

```
## Warning: Removed 234 rows containing non-finite values (‘stat_boxplot()‘).
```



One other thing the class found was an outlier for E. coli. This was often removed from analysis, but like the DUP sites, was not removed from the actual file we saved.

```
wq_units |>
  select(Site, Date, E_coli_MPN_100mL) |>
  arrange(-E_coli_MPN_100mL)|>
  head()
```

```
## # A tibble: 6 x 3
##   Site  Date       E_coli_MPN_100mL
##   <chr> <chr>                 <dbl>
## 1 PHB12 2017-09-14           155310
## 2 HC7   2012-03-20            24200
## 3 HSC6  2012-03-20            24200
## 4 LBC8  2012-03-20            24200
## 5 RB15  2012-03-20            24200
## 6 TC17  2012-03-20            24200
```

To remove from an R object, we used the following code before running an analysis, which replaces the outlier value with an NA.

```
wq_units <- wq_units |>
 mutate(E_coli_MPN_100mL = gsub('155310', NA, E_coli_MPN_100mL))
```

Contact

This code was used for BIO 231: Environmental Topics and Analysis at William Peace University in the Spring of 2024. For questions, you can contact Mary Glover (mmglover@peace.edu).

All code for the class, which includes both the course lessons how to use R and the code used to anlayze the water quality data is in GitHub at https://github.com/maryglover/wpu-bio231