Exploratory Analysis

Tony Ni, Antonella Basso, Jose Lopez

6/11/2020

Libraries

Goals:

comparing how upgradients look vs downgradients through a graph (make sure the average thing/wrangling to fix it)

pick a chemical of interest

histograms of concentrations (one for upgradient vs downgradient for a single site)

compare the compare the 2 powerplants overall (upgradient at one vs the other)

regression

Reading in Data

```
setwd("~/harvard-summer-biostats")
illinois <- read_csv("data/illinois.csv") #read in data
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   state = col_character(),
##   site = col_character(),
##   disposal.area = col_character(),
##   type = col_character(),
##   well.id = col_character(),
##   gradient = col_character(),
##   samp.date = col_character(),
##   contaminant = col_character(),
##   measurement.unit = col_character(),
##   concentration = col_double()
## )
```

Checking unique/distinct levels

```r
#check how many observations there
nrow(illinois)
```

```
## [1] 38792
```

```r
#checking distincts sites
unique(illinois$site)
```

```
##  [1] "Baldwin Energy Complex"
##  [2] "Coffeen Power Station"
##  [3] "Dallman Power Generating Station"
##  [4] "Duck Creek Power Station"
##  [5] "Edwards Power Station"
##  [6] "Havana Power Station"
##  [7] "Hennepin Power Station"
##  [8] "Joliet #29 Generating Station"
##  [9] "Joliet #9 Generating Station"
## [10] "Joppa Power Station"
## [11] "Kincaid Power Station"
## [12] "Newton Power Station"
## [13] "Powerton Generating Station"
## [14] "Prairie State Generating Company, LLC"
## [15] "SIPC Marion Power Plant"
## [16] "Waukegan Station"
## [17] "Will County"
## [18] "Wood River Power Station"
```

```r
#checking distinct gradient
unique(illinois$gradient)
```

```
## [1] "Downgradient" "Upgradient"   "Unknown"
```

```r
#checking distinct gradient
unique(illinois$measurement.unit)
```

```
## [1] "mg/l"  "su"    "pCi/l" "ug/l"
```

```r
#checking disposal area
unique(illinois$disposal.area)
```

```
##  [1] "Baldwin Bottom Ash Pond"
##  [2] "Baldwin Fly Ash Pond System"
##  [3] "Baldwin Bottom Ash Pond, Baldwin Fly Ash Pond System"
##  [4] "Coffeen Ash Pond No. 2"
##  [5] "Coffeen Ash Pond No. 1"
##  [6] "Coffeen GMF Recycle Pond"
##  [7] "Coffeen GMF Gypsum Stack Pond"
##  [8] "Coffeen GMF Gypsum Stack Pond, Coffeen Landfill"
##  [9] "Coffeen Ash Pond No. 1, Coffeen Ash Pond No. 2"
## [10] "Coffeen Landfill"
```

```
## [11] "Coffeen Ash Pond No. 2, Coffeen GMF Recycle Pond"
## [12] "Dallman Ash Pond, Lakeside Ash Pond"
## [13] "Duck Creek Bottom Ash Basin"
## [14] "Duck Creek Landfill"
## [15] "Duck Creek GMF Pond"
## [16] "Edwards Ash Pond"
## [17] "Havana East Ash Pond (Cells 1, 2, 3, and 4)"
## [18] "Hennepin Ash Pond No. 2, Hennepin East Ash Pond, Hennepin Landfill"
## [19] "Henepin Old West Ash Pond (Pond No. 1 and Pond No. 3) and Hennepin Old West Polishing Pond"
## [20] "Hennepin Ash Pond No. 2"
## [21] "Hennepin East Ash Pond"
## [22] "Hennepin Landfill"
## [23] "Ash Pond 2"
## [24] "Lincoln Stone Quarry"
## [25] "Joppa Landfill"
## [26] "Joppa East Ash Pond"
## [27] "Kincaid Ash Pond"
## [28] "Newton Landfill 2"
## [29] "Newton Primary Ash Pond"
## [30] "Ash By-pass Basin, Ash Surge Basin, Former Ash Basin"
## [31] "Near Field Landfill"
## [32] "Settling Pond"
## [33] "West Ash Pond, East Ash Pond"
## [34] "Ash Pond 2 South, Ash Pond  3 South"
## [35] "Wood River West Ash Ponds 1, 2E, 2W"
## [36] "Wood River Primary East Ash Pond"
```

```
#checking type (L = land fuel, SI = surface impoundement, M = mixed/multiunit)
#there will be multiple different dumpsites at a powerplant site. some sites are wet (ponds of water wh
unique(illinois$type)
```

```
## [1] "SI" "M"  "L"
```

```
illinois %>%
  group_by(type) %>%
  summarize(n())
```

```
## # A tibble: 3 x 2
##   type  `n()`
##   <chr> <int>
## 1 L      8869
## 2 M       875
## 3 SI    29048
```

```
illinois %>%
  group_by(measurement.unit) %>%
  summarize(n())
```

```
## # A tibble: 4 x 2
##   measurement.unit `n()`
##   <chr>            <int>
## 1 mg/l             32526
```

```
## 2 pCi/l             1721
## 3 su                1964
## 4 ug/l              2581
```

```
#checking wells
illinois %>%
  group_by(well.id) %>%
  summarize(n())
```

```
## # A tibble: 198 x 2
##    well.id `n()`
##    <chr>   <int>
##  1 03R       175
##  2 05DR      175
##  3 05R       175
##  4 08D       175
##  5 12        175
##  6 13        175
##  7 18D       175
##  8 18S       172
##  9 2         174
## 10 21        350
## # ... with 188 more rows
```

Tony: Molybdenum to Thallium

use illinois and newyork

lower detection limit (if a chemical is really low, it can't differentiate beyond a certain low level, it might be the values with all teh same values)

all of the contaminants are harmful, some chemicals are more useful for detecting coal contamination (most important/indicative of coal contamination is boron (occurs at really high levels in coal)),

looking at the report given to us might be good about the different chemicals in coal (if we are curious)

total dissolved solids are probably not important, imprevise measurement of how much "crap is in the water" - doesnt represent any toxic chemicals

Wrangling

```
illinois1 <- illinois %>%
  select(site, disposal.area, type, well.id, gradient, contaminant,
         measurement.unit, concentration) %>%
  mutate(well.id_contaminant = paste0(well.id, "_", contaminant)) %>% #for future use
  rename(c("disposal_area" = "disposal.area", "well_id" = "well.id",
           "unit" = "measurement.unit"))

#fixing 'contaminant' string by removing everything after the comma
illinois1$contaminant=gsub(", total", "", illinois1$contaminant)

#testing
avg_contaminant <- illinois1 %>%
  group_by(well_id, contaminant) %>%
```

```
  summarise_each(funs(mean)) %>%
  select(1,2,8) #selecting only numeric columns

#temporarily uniting columns for joining in next step
temp <- avg_contaminant %>%
  unite("well.id_contaminant", well_id, contaminant)

#joining orig dataframe and avg_contaminant dataframe
combined <- left_join(temp, illinois1, by = "well.id_contaminant") %>%
  distinct(well.id_contaminant, .keep_all = TRUE) %>%
  separate(well.id_contaminant, c('well_id', 'contaminant'), sep="_") %>%
  select(1:8)

#spreading to wide data frame format to add missing info
combined2 <- combined %>% #collapse empty rows
  spread(contaminant, concentration.x) %>%
  group_by(well_id) %>%
  summarise_each(funs(first(.[!is.na(.)]))) %>%
  select(-c(unit))

#gathering back to long data frame format
combined3 <- combined2 %>%
  gather("contaminant", "concentration", 6:26)
```

Summary Statistics

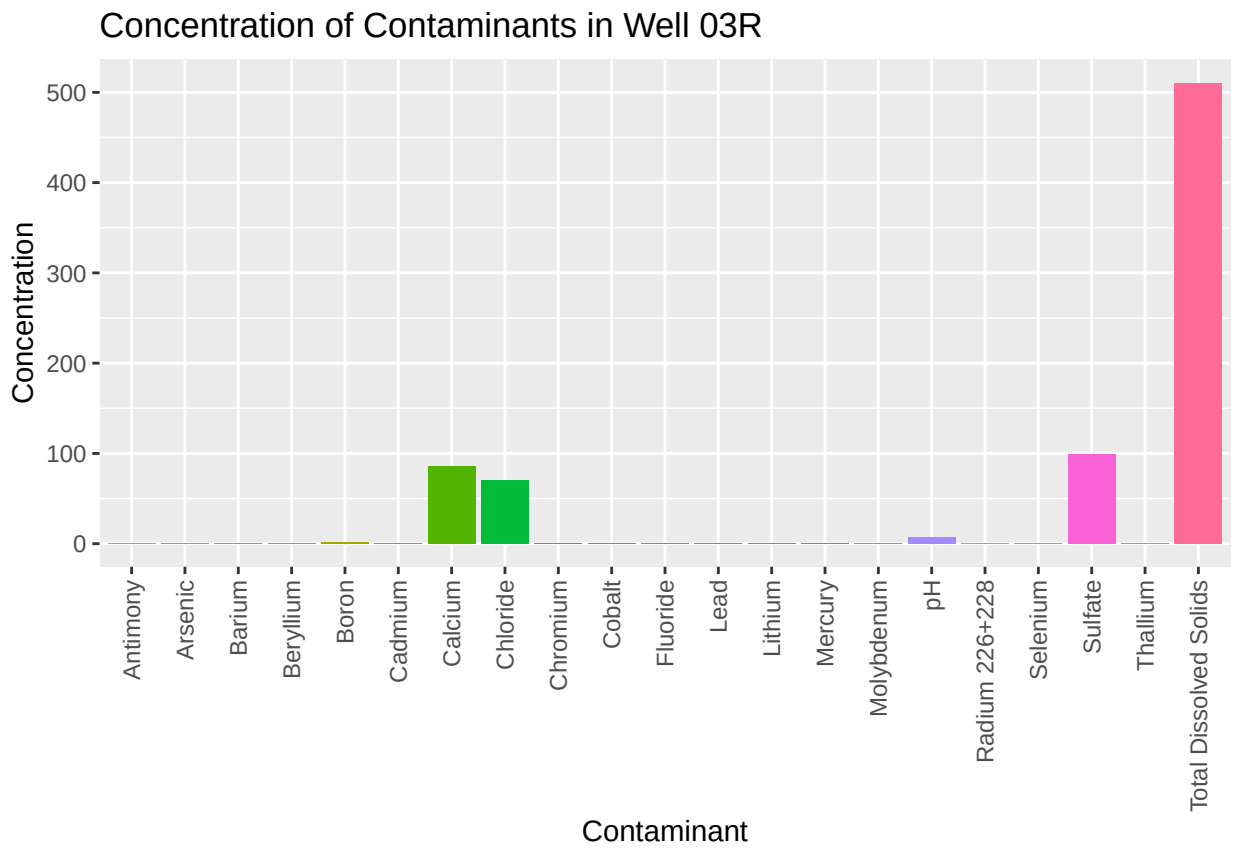
```
illinois %>%
  group_by(contaminant) %>%
  summarize(mean_concentration = mean(concentration))
```

```
## # A tibble: 21 x 2
##    contaminant      mean_concentration
##    <chr>                         <dbl>
##  1 Antimony, total               0.626
##  2 Arsenic, total                2.56
##  3 Barium, total                39.8
##  4 Beryllium, total              0.141
##  5 Boron, total                125.
##  6 Cadmium, total                0.141
##  7 Calcium, total             7460.
##  8 Chloride                     79.1
##  9 Chromium, total               0.969
## 10 Cobalt, total                 0.761
## # ... with 11 more rows
```

Visualizations

```
#looking at one specific well and the contaminant concentrations within it
```

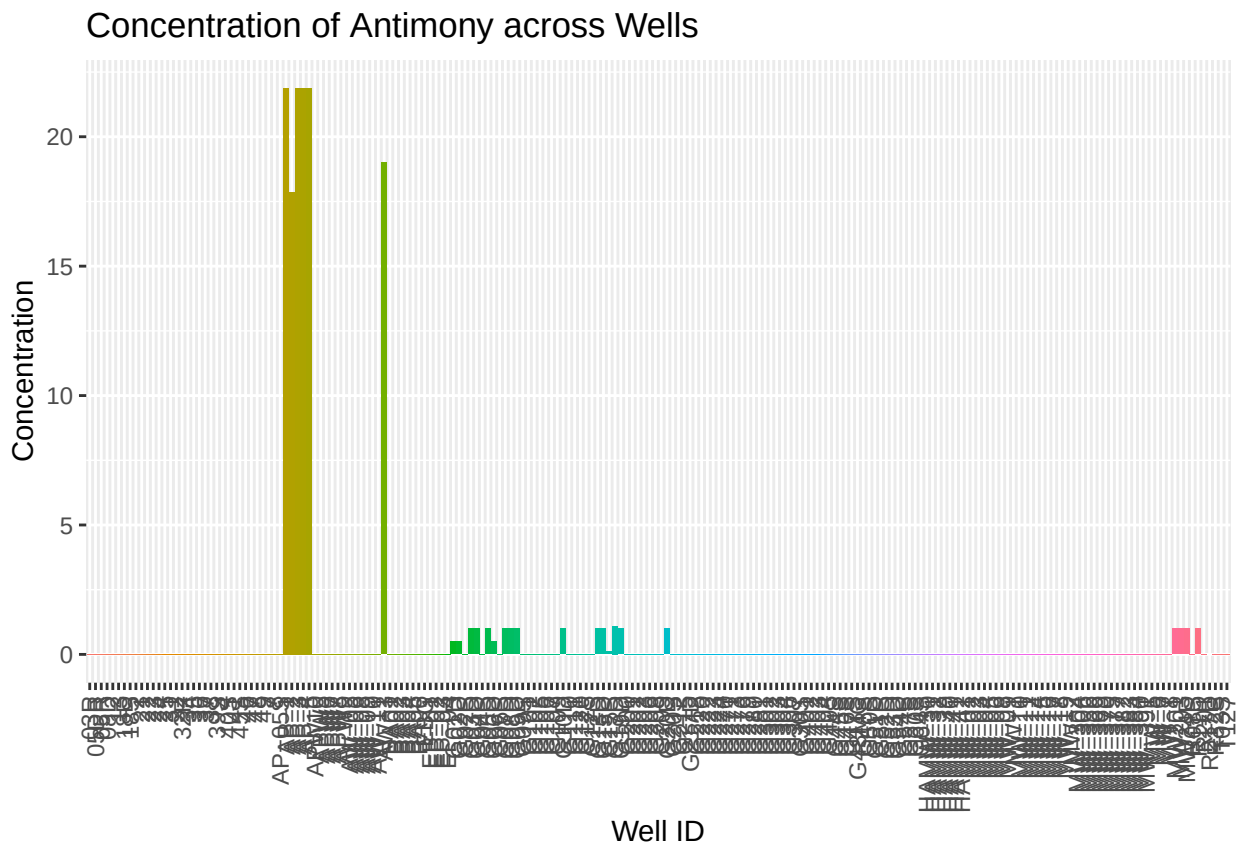
```
ggplot(data = combined3 %>%
         filter(well_id == "03R"), aes(x = contaminant, y = concentration,
                                       fill = contaminant)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  xlab("Contaminant") +
  ylab("Concentration") +
  ggtitle("Concentration of Contaminants in Well 03R") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
#looking at one contaminant for all wells (counts)

ggplot(data = combined3 %>%
         filter(contaminant == "Antimony"),
       aes(x = well_id, y = concentration, fill = well_id)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  xlab("Well ID") +
  ylab("Concentration") +
  ggtitle("Concentration of Antimony across Wells") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

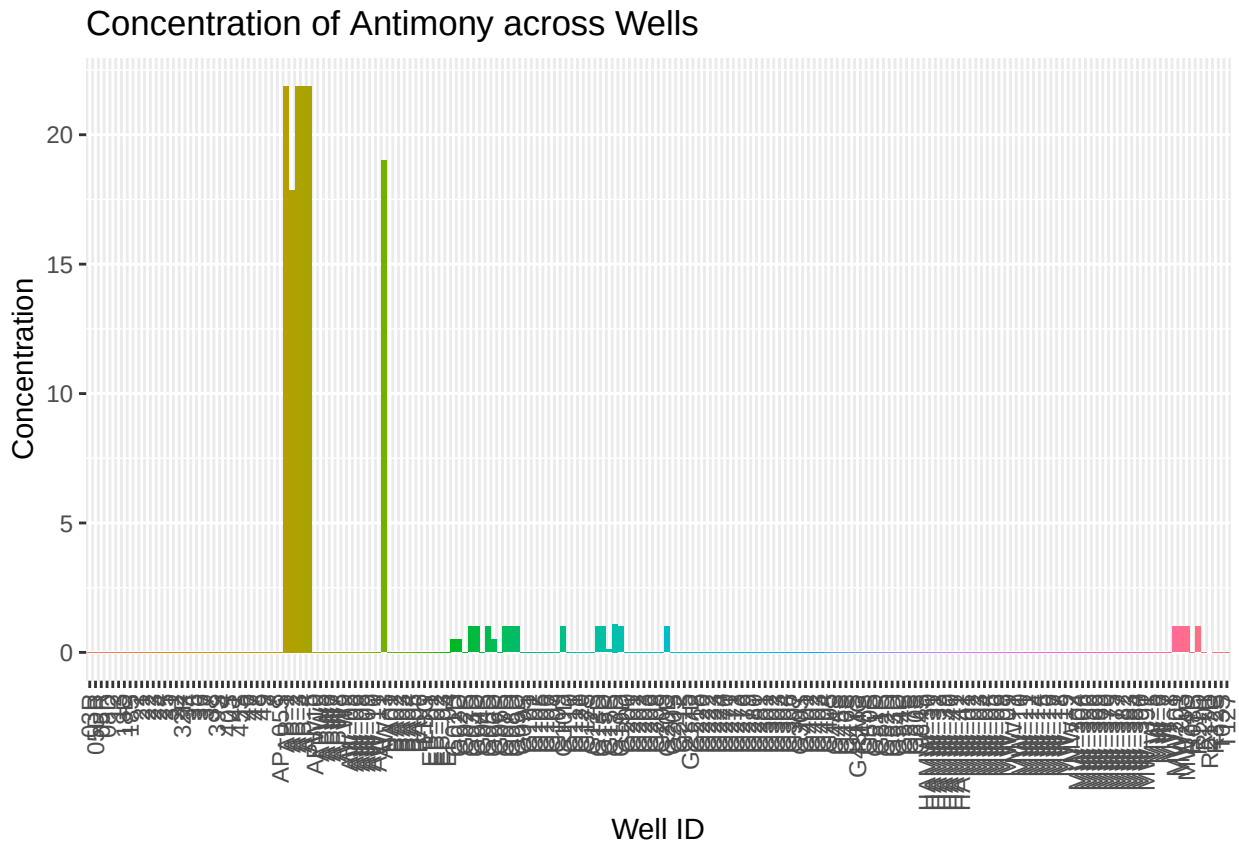
```
## Warning: Removed 1 rows containing missing values (position_stack).
```



The histogram is incompehensible, there are way too many wells in Illinois. Unsure, but having to make all these plots constantly seems unreasonable for all of these potential wells and contaminants to look through, maybe choose only those wells with concentration values greater than some certain value? (0.001 maybe?)

```
ggplot(data = combined3 %>%
         filter(contaminant == "Antimony"),
       aes(x = well_id, y = concentration, fill = well_id)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  xlab("Well ID") +
  ylab("Concentration") +
  ggtitle("Concentration of Antimony across Wells") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
levels(as.factor(combined3$well_id))
```

```
##   [1] "03R"     "05DR"    "05R"     "08D"     "12"      "13"      "18D"
##   [8] "18S"     "2"       "21"      "22"      "23"      "24"      "25"
##  [15] "31"      "32"      "32R"     "34"      "35"      "36"      "37"
##  [22] "38"      "39S"     "4"       "40S"     "41"      "45S"     "46"
##  [29] "47"      "48"      "49"      "7"       "8"       "AP-05S"  "AP-1"
##  [36] "AP-2"    "AP-3"    "AP-4"    "AP-5"    "APW10"   "APW5"    "APW6"
##  [43] "APW7"    "APW8"    "APW9"    "AW-05"   "AW-06"   "AW-08"   "AW-09"
##  [50] "AW-10"   "AW-11"   "AW-3"    "BA01"    "BA02"    "BA03"    "BA04"
##  [57] "BA05"    "BA06"    "EBG"     "EP-01"   "EP-02"   "EP-03"   "EP-04"
##  [64] "G01D"    "G02D"    "G02S"    "G03D"    "G04D"    "G04S"    "G05D"
##  [71] "G06D"    "G06S"    "G07D"    "G08D"    "G09D"    "G09S"    "G101"
##  [78] "G102"    "G105"    "G106"    "G107"    "G109"    "G10D"    "G110"
##  [85] "G111"    "G120"    "G125"    "G12S"    "G13D"    "G15D"    "G15S"
##  [92] "G17D"    "G19D"    "G200"    "G201"    "G202"    "G203"    "G206"
##  [99] "G208"    "G209"    "G20D"    "G20S"    "G212"    "G215"    "G217D"
## [106] "G218"    "G220"    "G222"    "G223"    "G224"    "G270"    "G271"
## [113] "G273"    "G276"    "G279"    "G280"    "G281"    "G301"    "G302"
## [120] "G303"    "G304"    "G306"    "G307"    "G30S"    "G401"    "G402"
## [127] "G403"    "G404"    "G405"    "G44S"    "G45S"    "G46S"    "G47S"
## [134] "G48MG"   "G48S"    "G50S"    "G51D"    "G51S"    "G52D"    "G53D"
## [141] "G54D"    "G54S"    "G57S"    "G60S"    "G64S"    "HAMW-30" "HAMW-31"
## [148] "HAMW-32" "HAMW-39" "HAMW-40" "HAMW-41" "HAMW-42" "MW-01"   "MW-02"
## [155] "MW-03"   "MW-04"   "MW-05"   "MW-06"   "MW-08"   "MW-09"   "MW-1"
```

```
## [162] "MW-10"   "MW-11"   "MW-12"   "MW-14"   "MW-15"   "MW-16"   "MW-17"
## [169] "MW-18"   "MW-19"   "MW-2"    "MW-304"  "MW-306"  "MW-356"  "MW-366"
## [176] "MW-369"  "MW-370"  "MW-375"  "MW-377"  "MW-382"  "MW-383"  "MW-384"
## [183] "MW-390"  "MW-391"  "MW-5"    "MW-6"    "MW-7"    "MW-8"    "MW201"
## [190] "MW203"   "MW24D"   "R08S"    "R11D"    "R201"    "R217D"   "R32S"
## [197] "T03S"    "T127"
```

#Comparing Upgradient Wells

```
#get the first 10 highest avg mg/l (not ph or total sulfate)
#not pH, total dissolve solids, or radium b/c different units

dont_use <- c("pH", "Radium 226+228", "Total Dissolved Solids")

#pull out the names (from top 10 high to low avg concentrations) of contaminants
chemicals <- combined3 %>%
    filter(contaminant != "pH") %>% #take our nonuseful contaminants
  filter(contaminant != "Radium 226+228") %>%
  filter(contaminant != "Total Dissolved Solids") %>%
    group_by(contaminant) %>%
    summarise(avg_conc = mean(concentration, na.rm = TRUE)) %>%
    arrange(desc(avg_conc)) %>%
  slice(1:10) %>% #there only seem to be 5 chemicals of nonzero(basically) values
    pull(contaminant)

chemicals
```

```
##  [1] "Calcium"  "Sulfate"  "Boron"    "Chloride" "Barium"   "Arsenic"
##  [7] "Chromium" "Lithium"  "Lead"     "Cobalt"
```

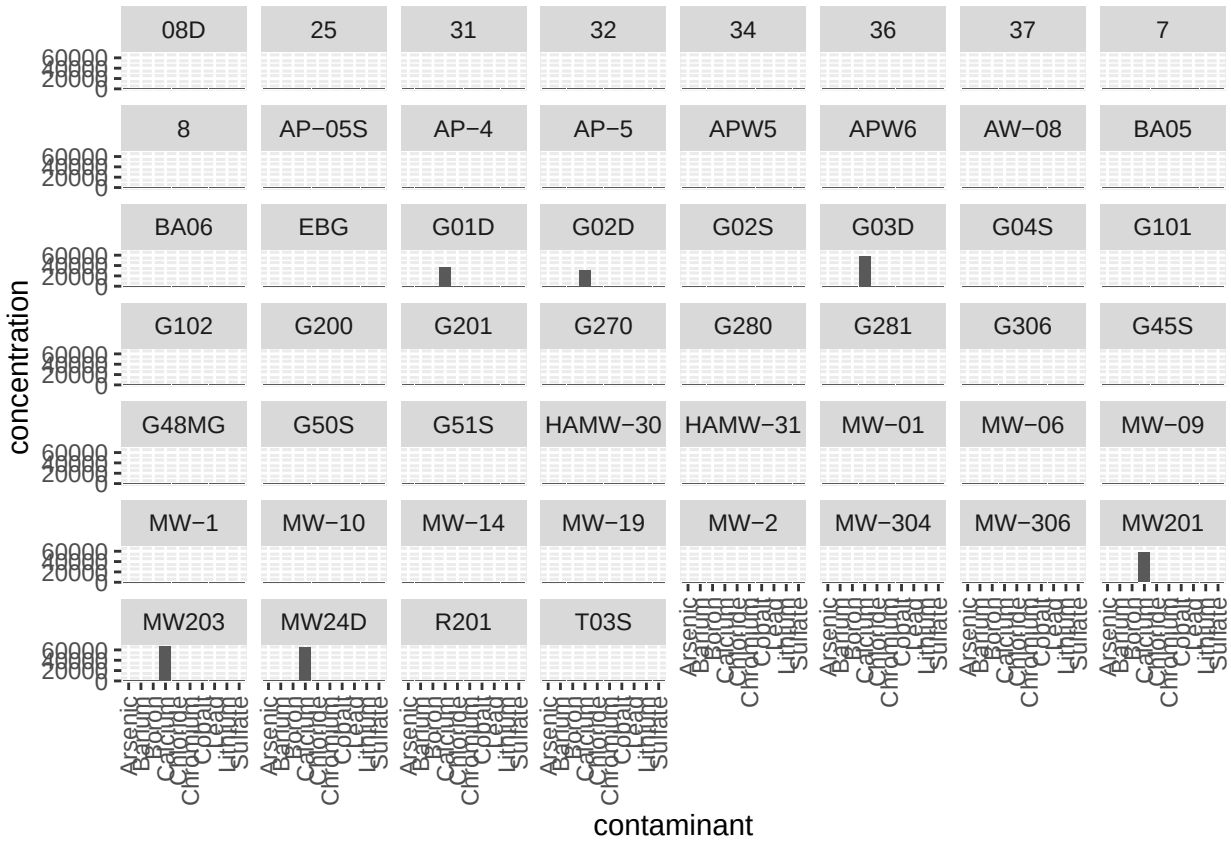
```
combined3 %>%
    group_by(contaminant) %>%
    summarise(avg_conc = mean(concentration))
```

```
## # A tibble: 21 x 2
##    contaminant avg_conc
##    <chr>          <dbl>
##  1 Antimony        NA
##  2 Arsenic         NA
##  3 Barium          NA
##  4 Beryllium       NA
##  5 Boron          145.
##  6 Cadmium         NA
##  7 Calcium       8142.
##  8 Chloride        75.3
##  9 Chromium        NA
## 10 Cobalt          NA
## # ... with 11 more rows
```

Is doing the mean of concentrations amongst all wells OK to do? Some wells like arsenic have wells with 0.001 (sign that measurement was insignificant/below threshold), and taking the avg sometimes makes it return NA as the mean (I THINK??? what does NA mean...?) ASK LULI (i think some of the chemicals in the paper had thresholds of DANGER that were in mg and others in microg, how would we account for this...?)

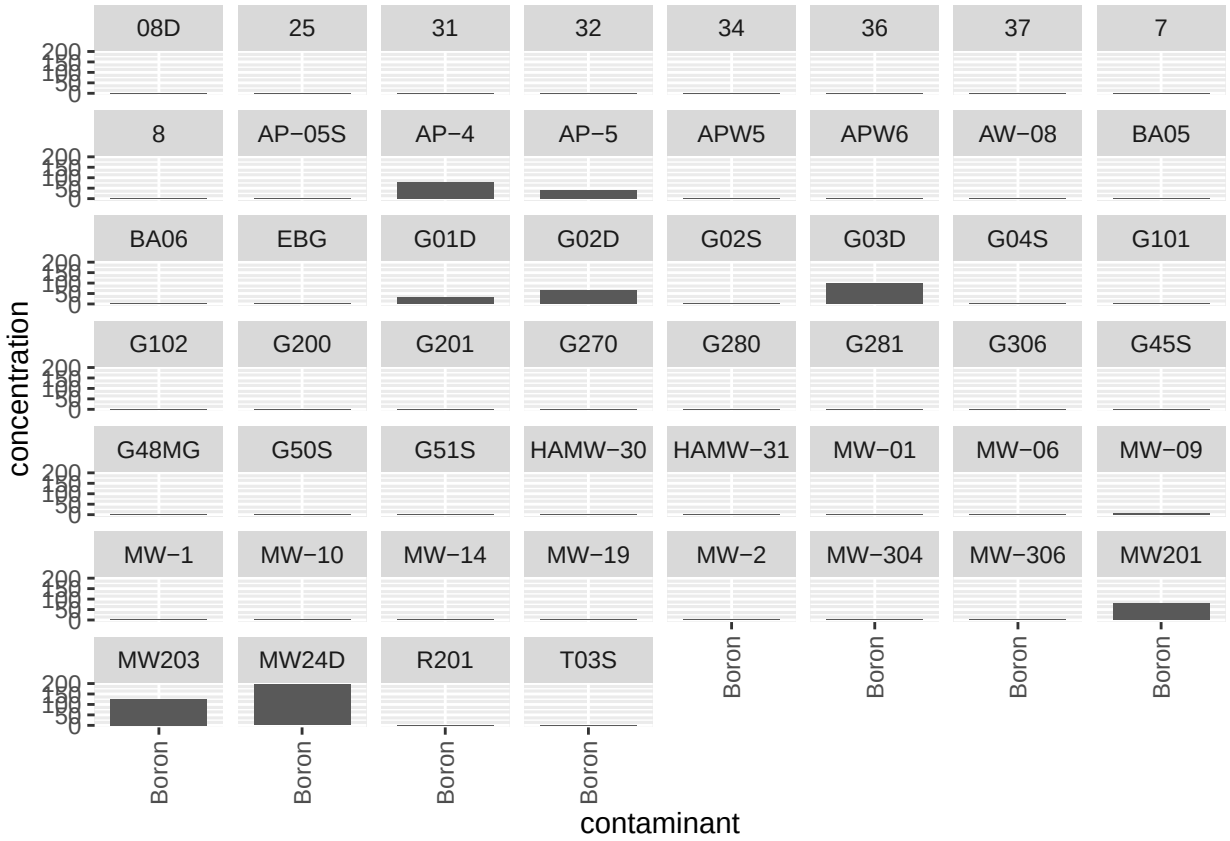
https://postimg.cc/ZCc072JM

```
#plot in a faceted grid bar plots of concentration vs. contaminant for all wells
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant %in% chemicals),
       aes(x = contaminant, y = concentration)) +
       geom_bar(stat = "Identity") +
       facet_wrap(well_id ~ .) +
       theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



The concentration of calcium is so high in some of the wells, it makes it difficult to look at the other contaminants' concentration across all wells, perhaps we should make different facetted plots for all contaminants.

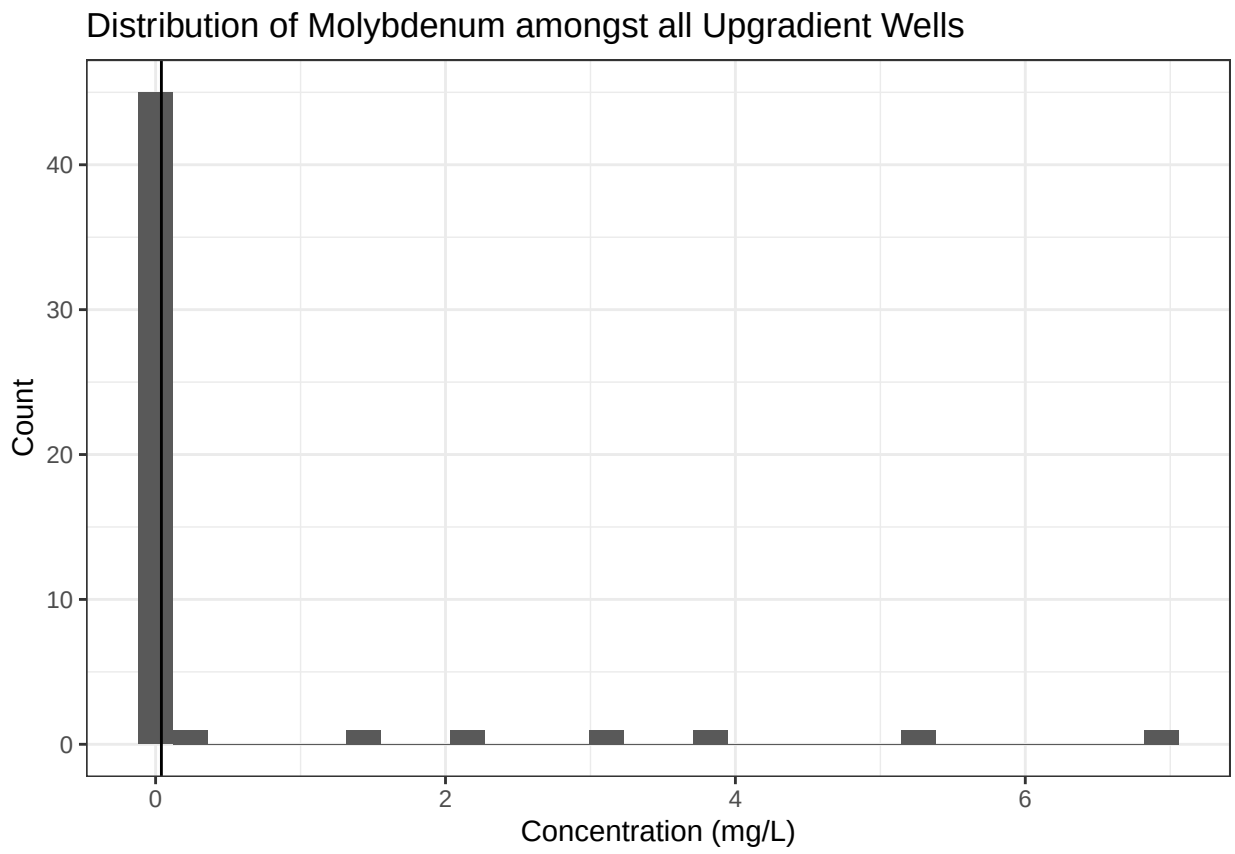
```
#Boron
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant %in% chemicals) %>%
         filter(contaminant == "Boron"),
       aes(x = contaminant, y = concentration)) +
       geom_bar(stat = "Identity") +
       facet_wrap(well_id ~ .) +
       theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



Let's make a histogram to show distribution of certain contaminants:

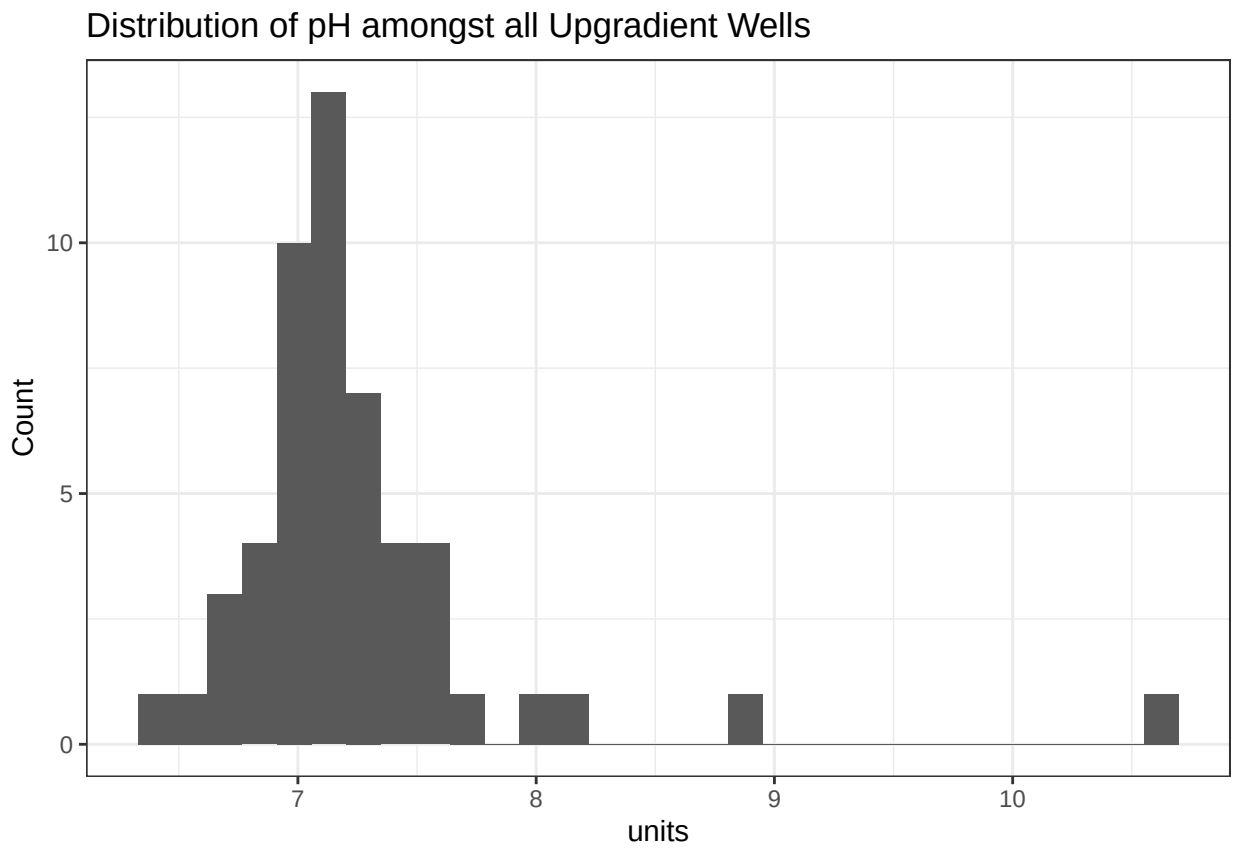
```
#Molybdenum
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "Molybdenum"),
       aes(x = concentration)) +
       geom_histogram() +
       geom_vline(xintercept = 40/1000) + #threshold value
  xlab("Concentration (mg/L)") +
  ylab("Count") +
  ggtitle("Distribution of Molybdenum amongst all Upgradient Wells") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#pH
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "pH"),
       aes(x = concentration)) +
       geom_histogram() +
  xlab("units") +
  ylab("Count") +
  ggtitle("Distribution of pH amongst all Upgradient Wells") +
  theme_bw()
```

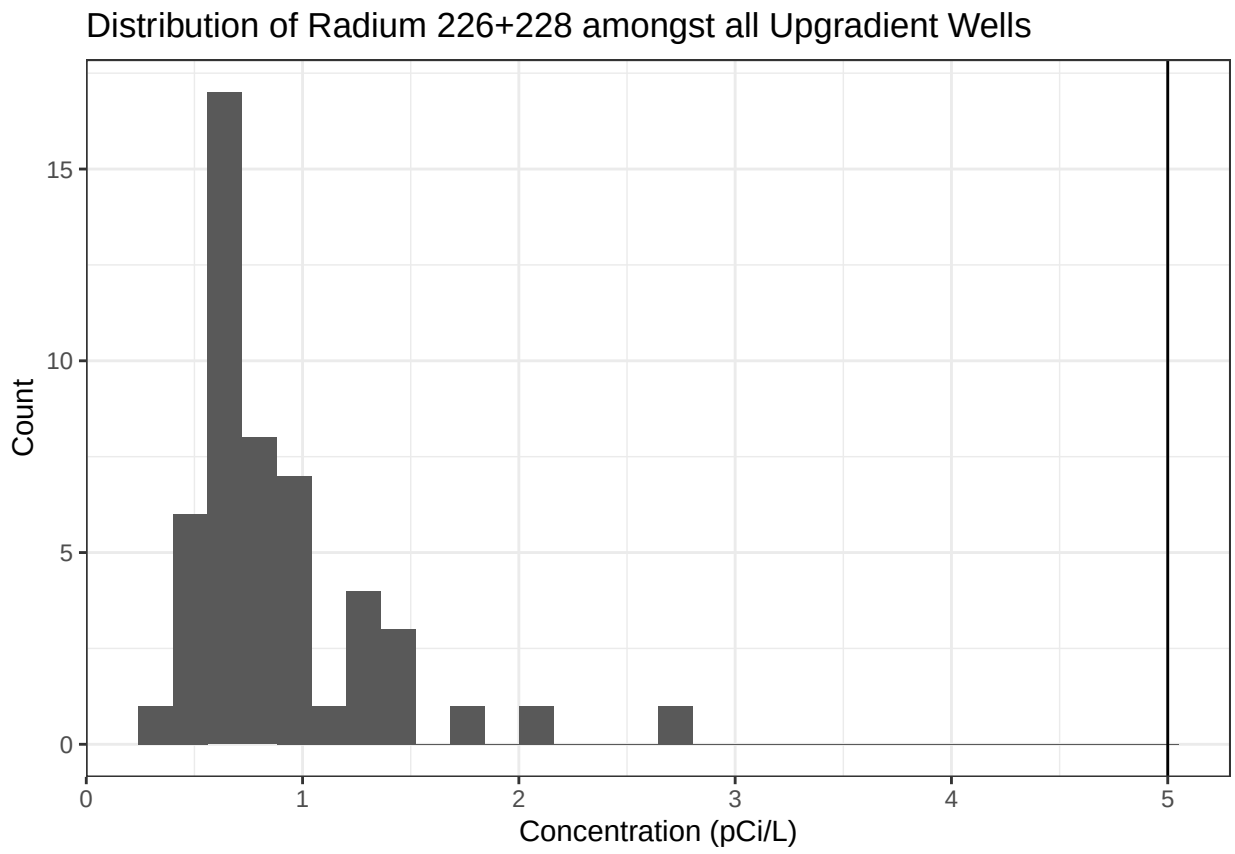
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Radium 226+228
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "Radium 226+228"),
       aes(x = concentration)) +
       geom_histogram() +
       geom_vline(xintercept = 5) + #threshold value
  xlab("Concentration (pCi/L)") +
  ylab("Count") +
  ggtitle("Distribution of Radium 226+228 amongst all Upgradient Wells") +
  theme_bw()
```

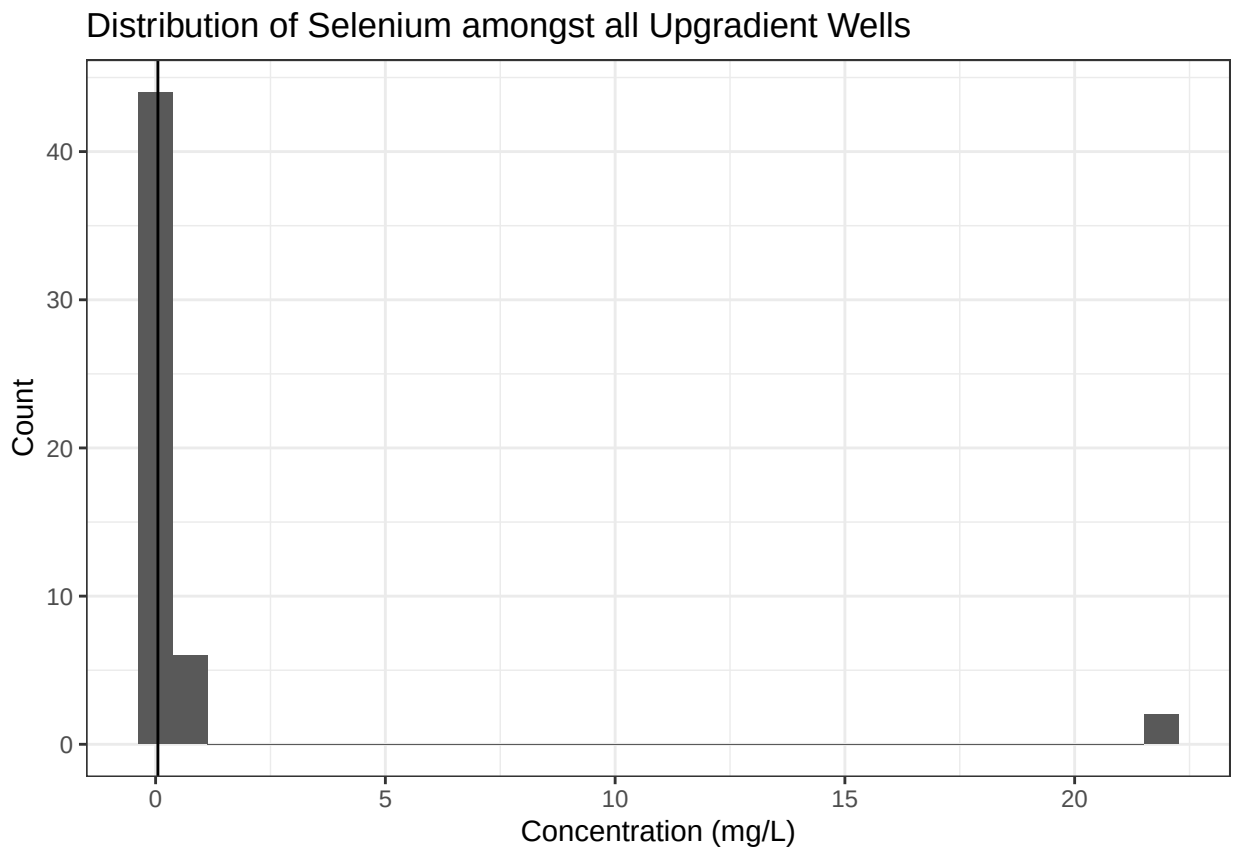
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```



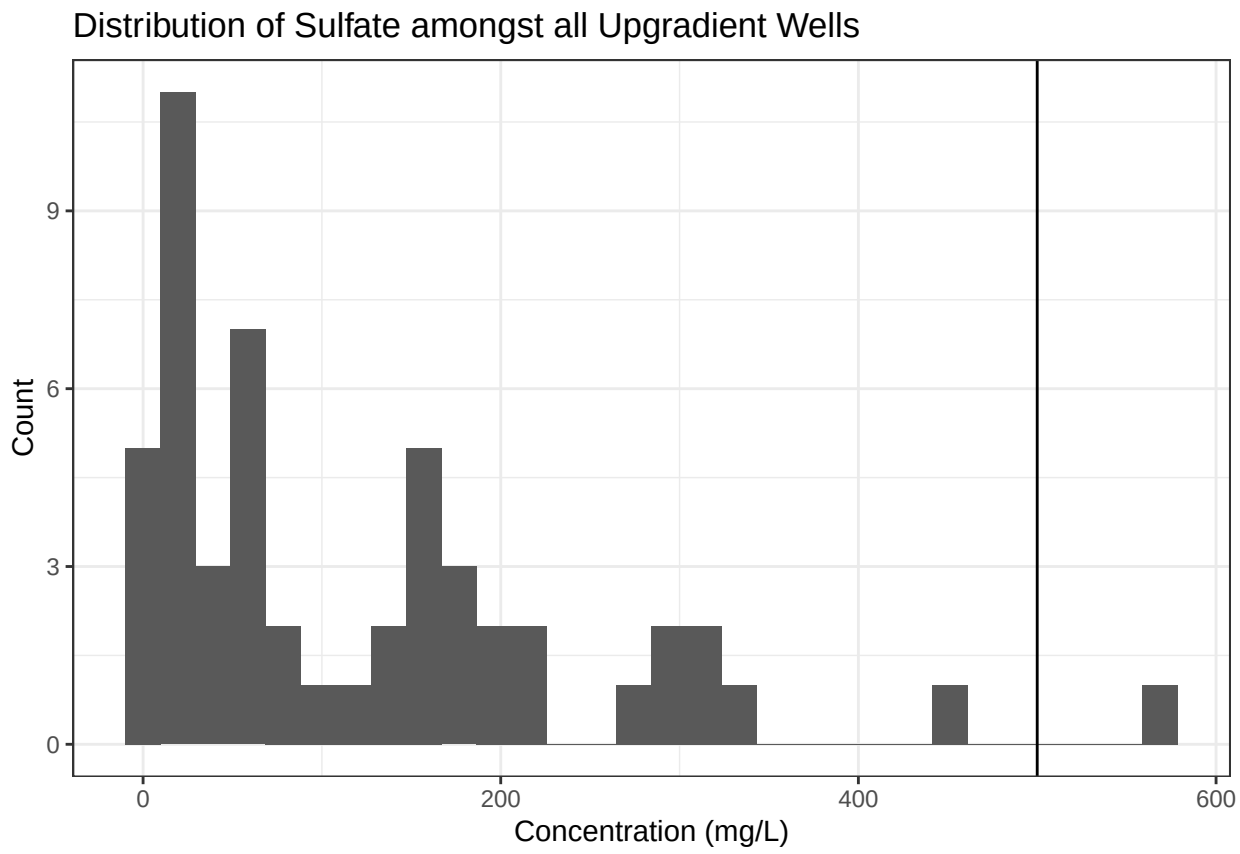

```
#Selenium
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "Selenium"),
       aes(x = concentration)) +
       geom_histogram() +
       geom_vline(xintercept = 50/1000) + #threshold value
  xlab("Concentration (mg/L)") +
  ylab("Count") +
  ggtitle("Distribution of Selenium amongst all Upgradient Wells") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



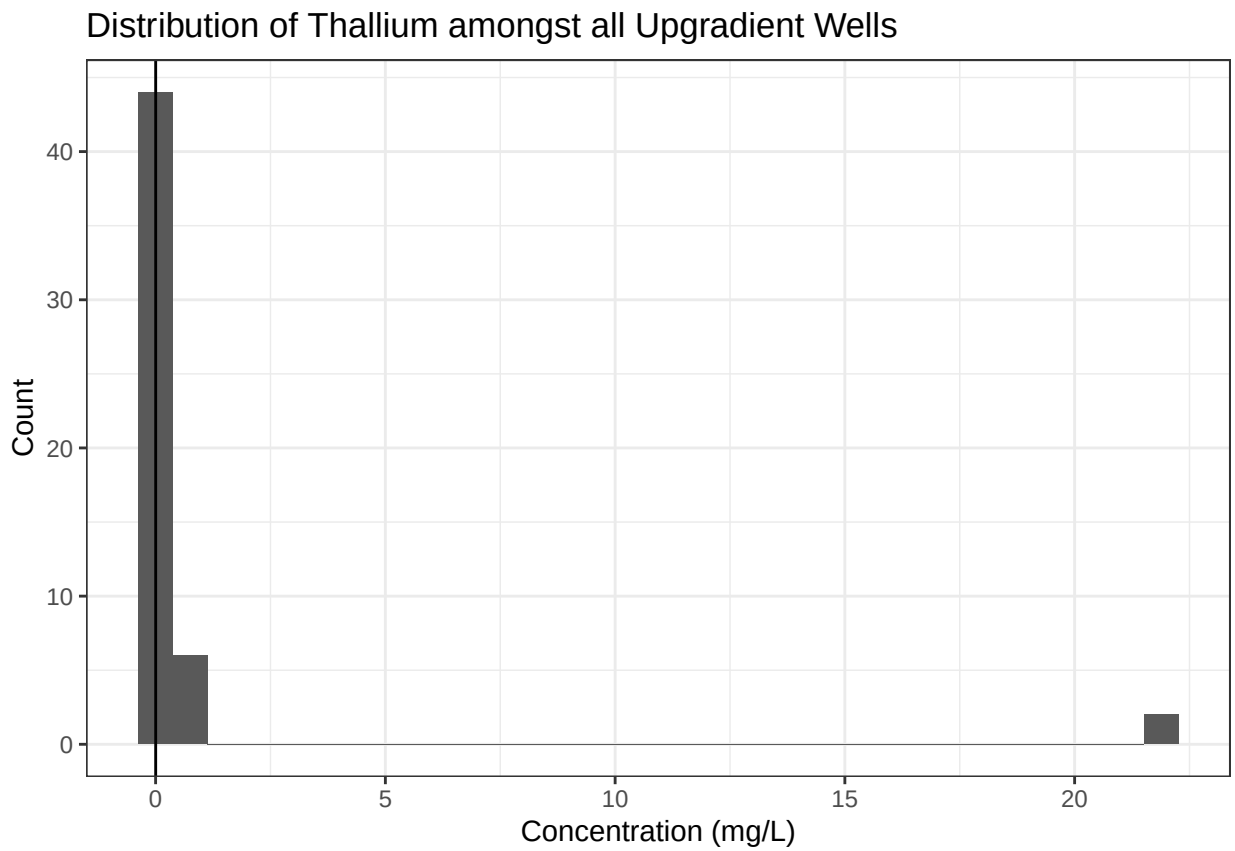

```
#Sulfate
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "Sulfate"),
       aes(x = concentration)) +
       geom_histogram() +
       geom_vline(xintercept = 500) + #threshold value
  xlab("Concentration (mg/L)") +
  ylab("Count") +
  ggtitle("Distribution of Sulfate amongst all Upgradient Wells") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Thallium
ggplot(data = combined3 %>%
         filter(gradient == "Upgradient") %>%
         filter(contaminant == "Thallium"),
       aes(x = concentration)) +
       geom_histogram() +
       geom_vline(xintercept = 2/1000) + #threshold value
  xlab("Concentration (mg/L)") +
  ylab("Count") +
  ggtitle("Distribution of Thallium amongst all Upgradient Wells") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see that there are some wells with concentrations that lie far beyond the threshold value (obtained from the PDF), and it would be nice to identify them. We will pull out the observatiosn with concentrations past the threshold and place them within a new dataset.

```
#creating vector of contaminants
contaminant <- unique(combined3$contaminant)

#creating vector of threshold values for each contaminant
threshold <- c(6/1000, 10/1000, 2, 4/1000, 3, 5/1000, NA, NA, 100/1000,
               6/1000, NA, 15/1000, 40/1000, 2/1000, 40/1000, NA, 5,
               50/1000, 500, 2/1000, NA)

contam_t <- cbind(contaminant, threshold) %>%
  na.omit()

#creating function to obtain all observations with values above threshold
getOverThreshold <- function(df){
  datalist = list()
  for(i in 1:nrow(contam_t)){ #for each contaminant i
    df1 <- filter(df, gradient == "Upgradient")
    df2 <- filter(df1, contaminant == contam_t[i])
    data <- filter(df2, concentration > contam_t[nrow(contam_t) + i])
    datalist[[i]] <- data
  }
  toReturn <- do.call(rbind, datalist)
```

```
  return(toReturn)
}

overthreshold_df <- getOverThreshold(combined3) #df with all observations over threshold value
```