Ammonia Model

Summer Heschong

2025-03-31

Setup

```
#load packages
library(here)
```

```
## here() starts at /Users/summerheschong/Stats_Group_Project
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(gtsummary)

wetlands <- read.csv(here('Data/Processed/Combined_Data_NArm.csv'))
```

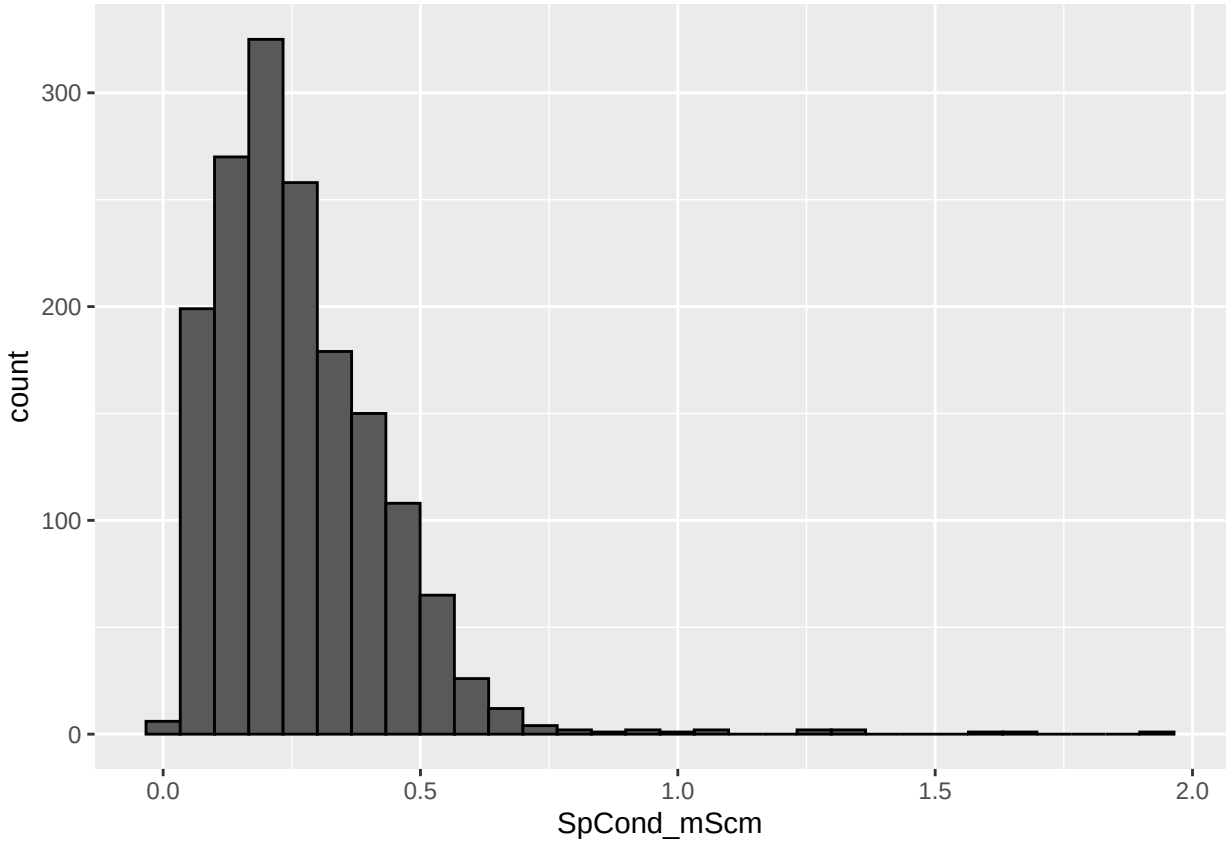
Step 1 - Research Question: What predicts ammonium levels in wetlands

Step 2 - Examine Data

Display raw counts and distributions of data

```
#remove outliers using Nicole's code
wetlands <- wetlands %>%
  mutate(across(where(is.numeric),
                ~ ifelse(abs
                         (. - mean
                         (., na.rm = TRUE)) > 3 * sd
                         (., na.rm = TRUE), NA, .)))

#create histograms
ggplot(wetlands, aes(x = SpCond_mScm)) +
geom_histogram(color = 'black')
```

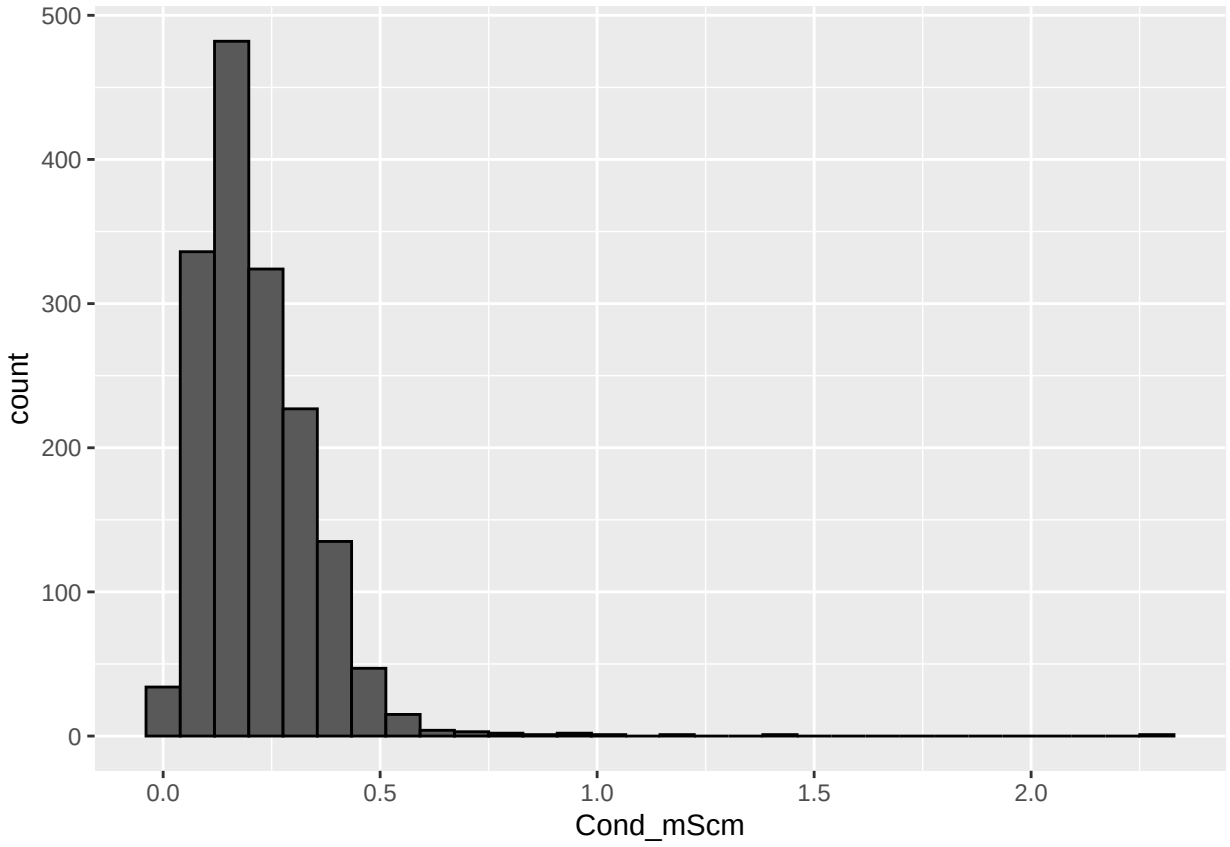
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



```
ggplot(wetlands, aes(x = Cond_mScm)) +
geom_histogram(color = 'black')
```

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

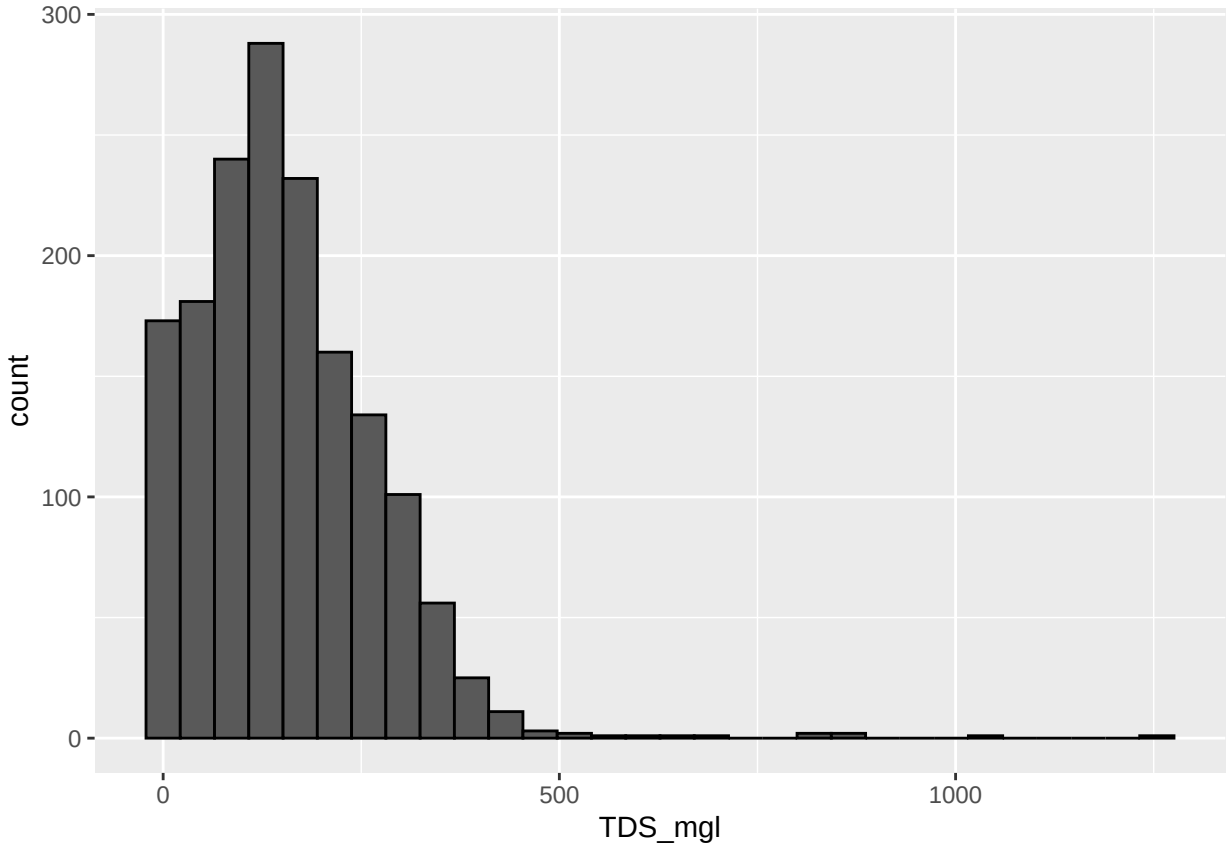
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = TDS_mgl)) +
geom_histogram(color = 'black')
```

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

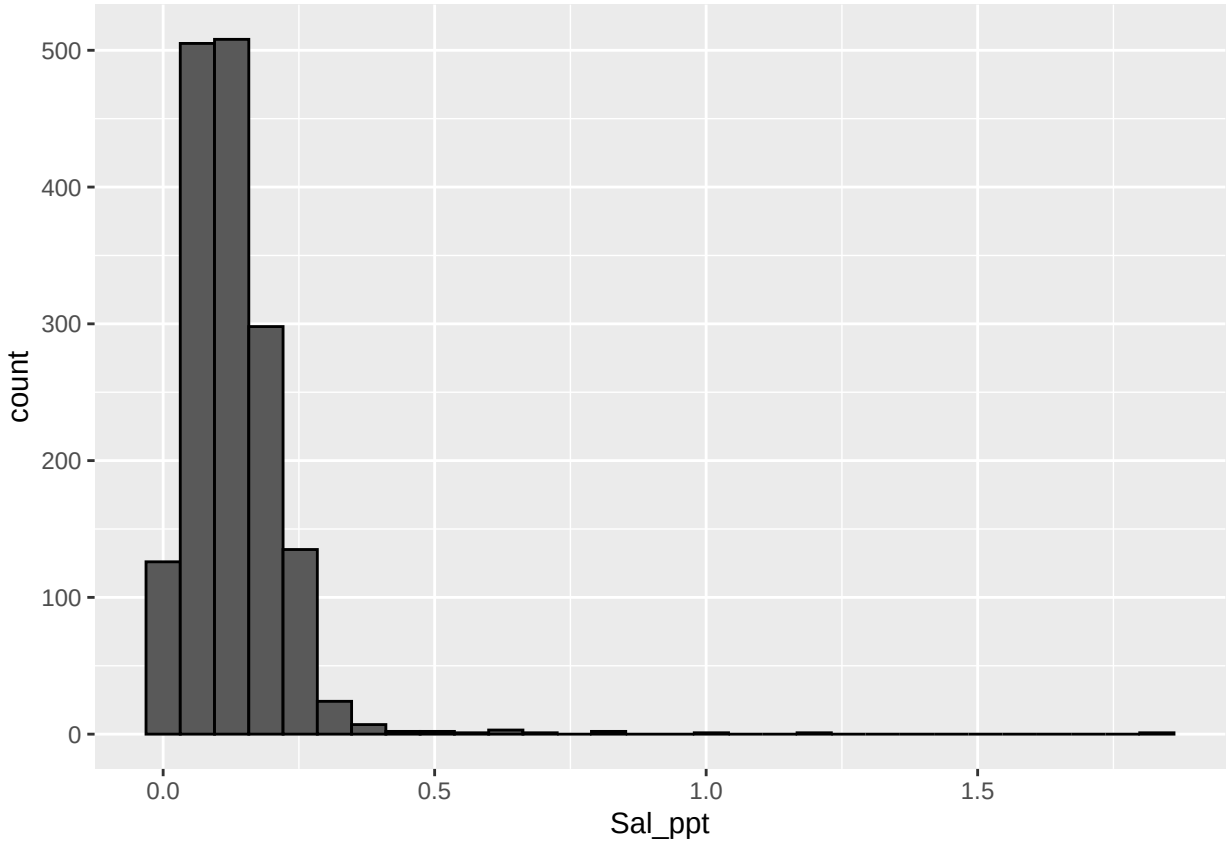
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = Sal_ppt)) +
geom_histogram(color = 'black')
```

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

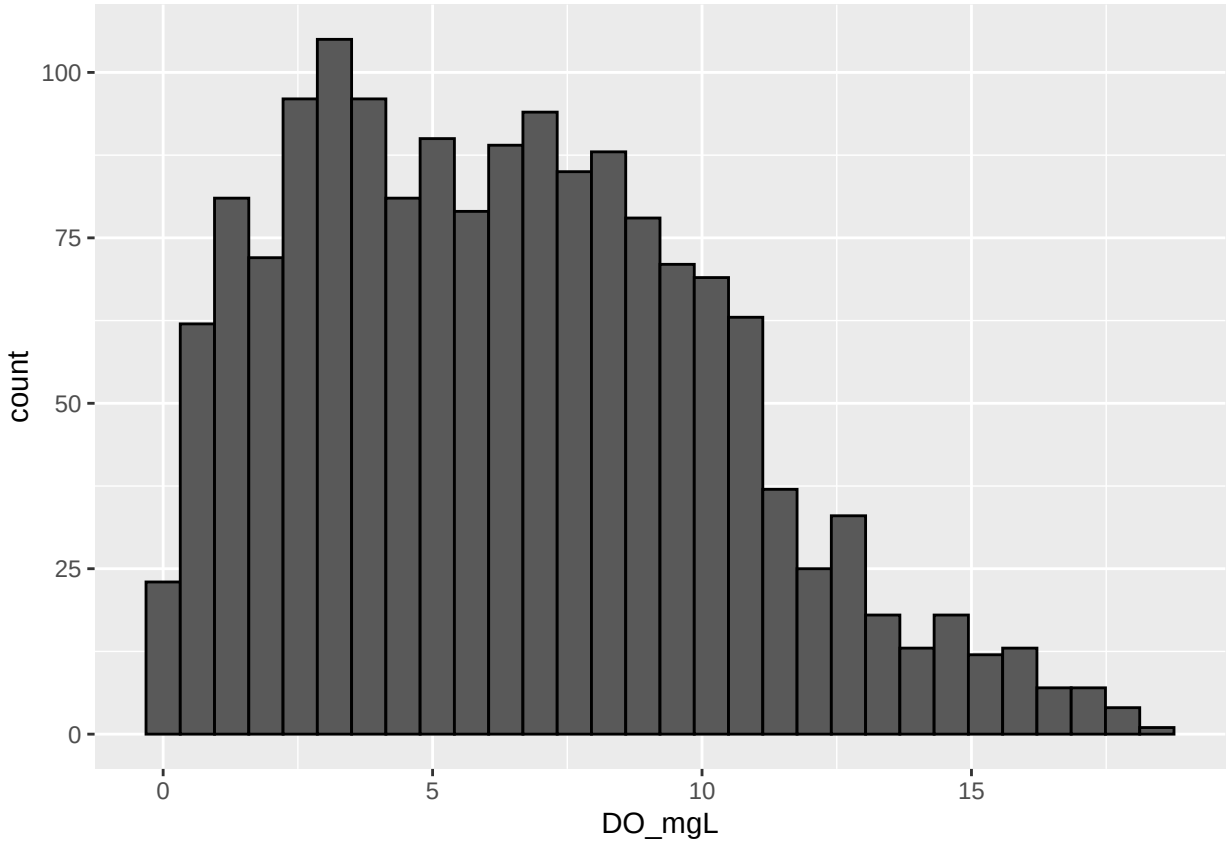
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = DO_mgL)) +
geom_histogram(color = 'black')
```

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

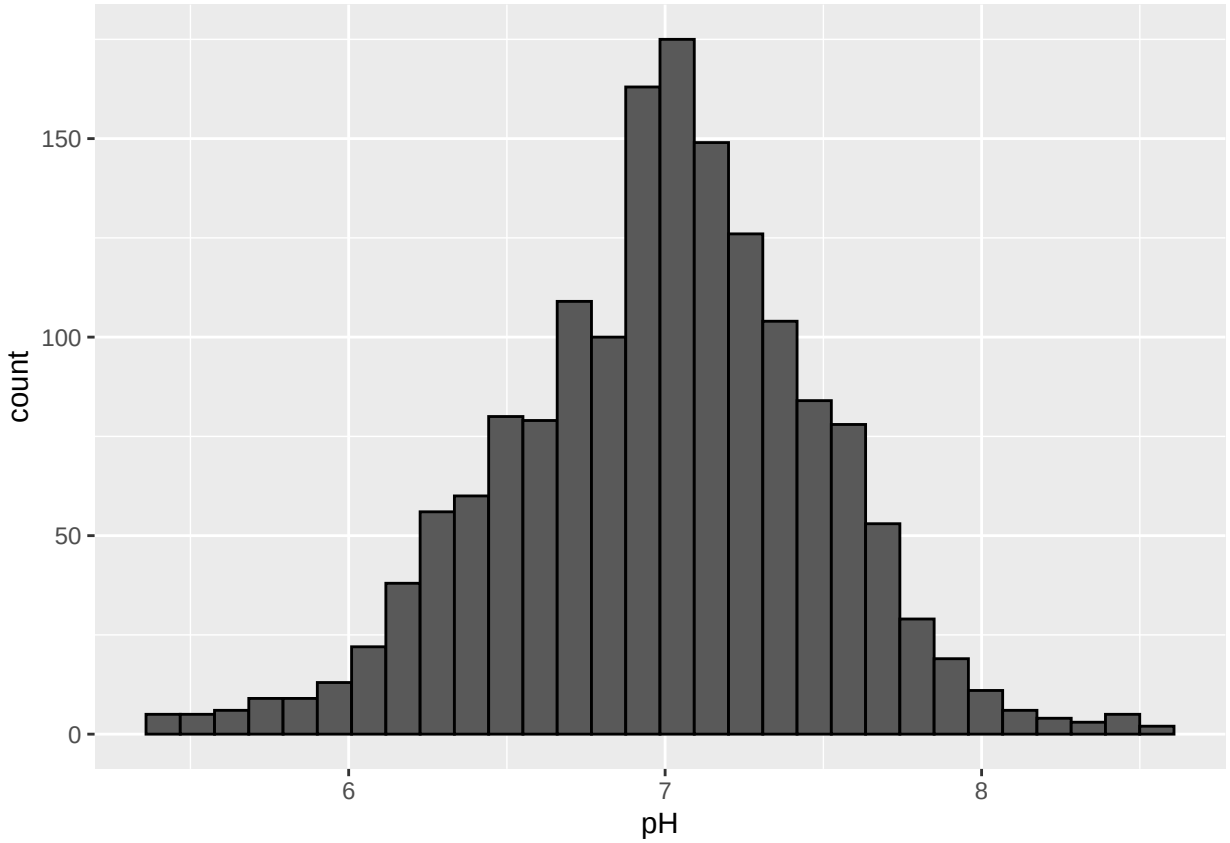
```
## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = pH)) +
geom_histogram(color = 'black')
```

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

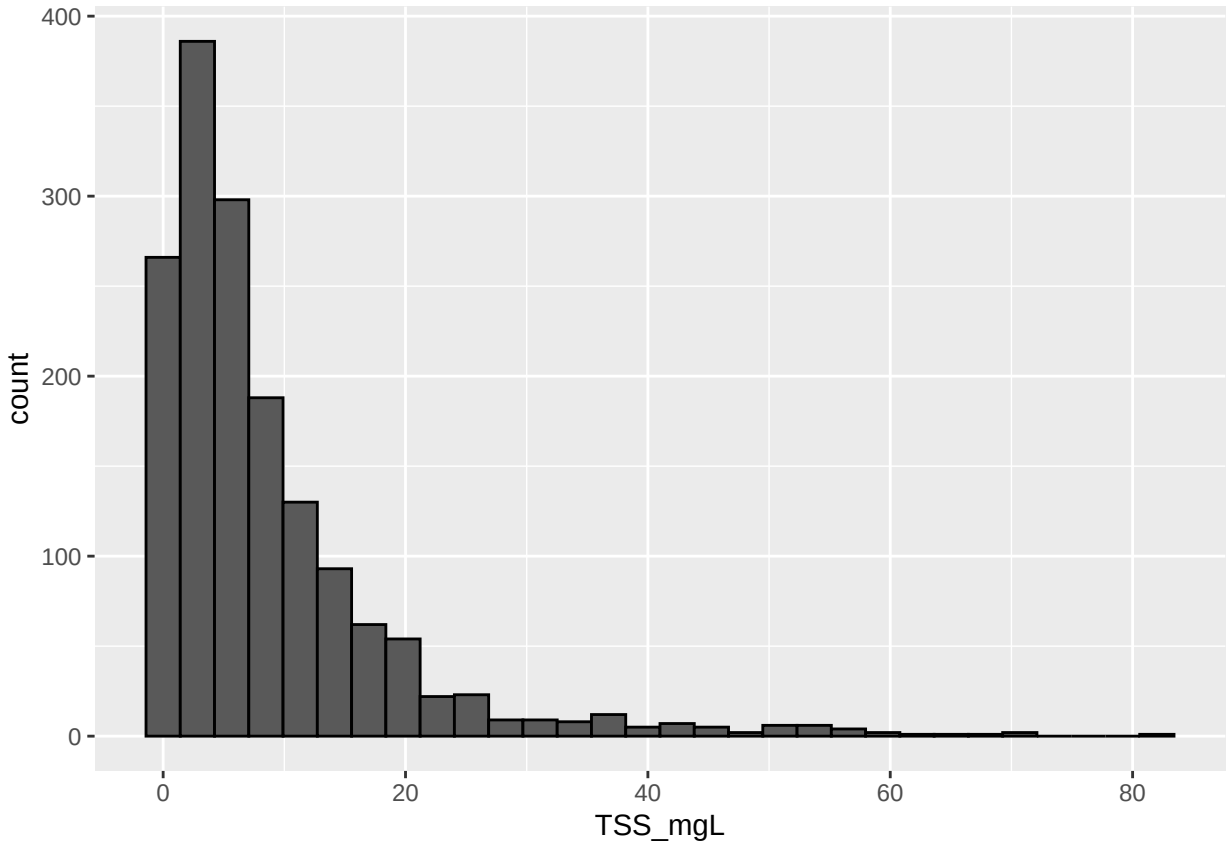
```
## Warning: Removed 16 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = TSS_mgL)) +
geom_histogram(color = 'black')
```

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

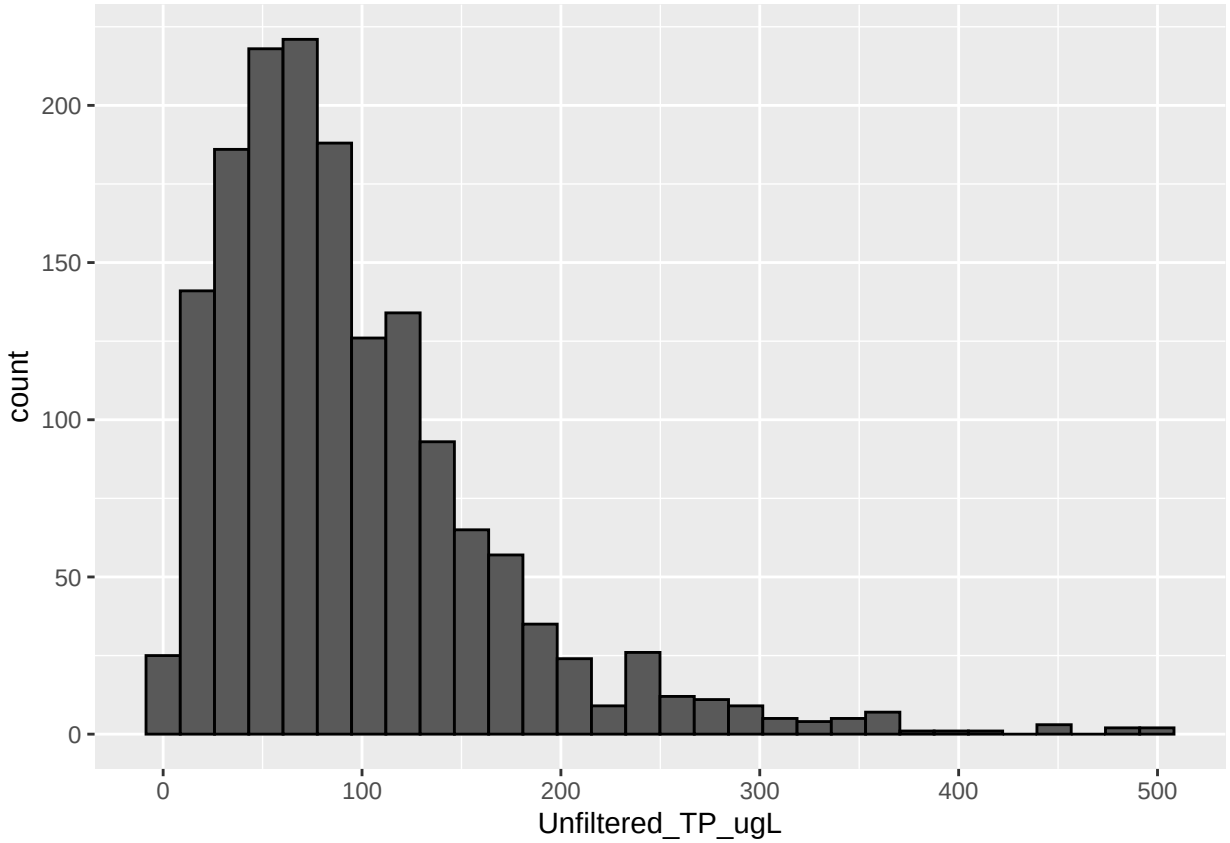
```
## Warning: Removed 15 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = Unfiltered_TP_ugL)) +
geom_histogram(color = 'black')
```

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

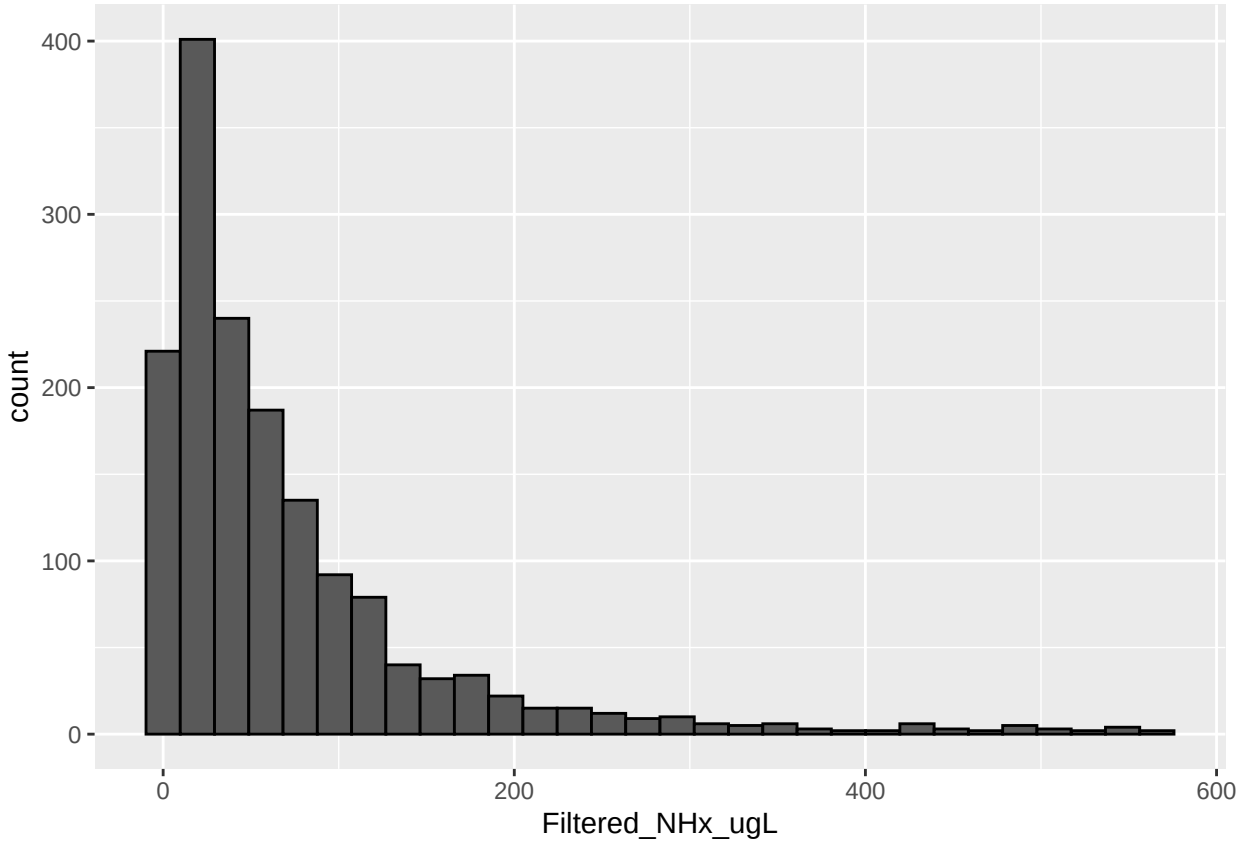
```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



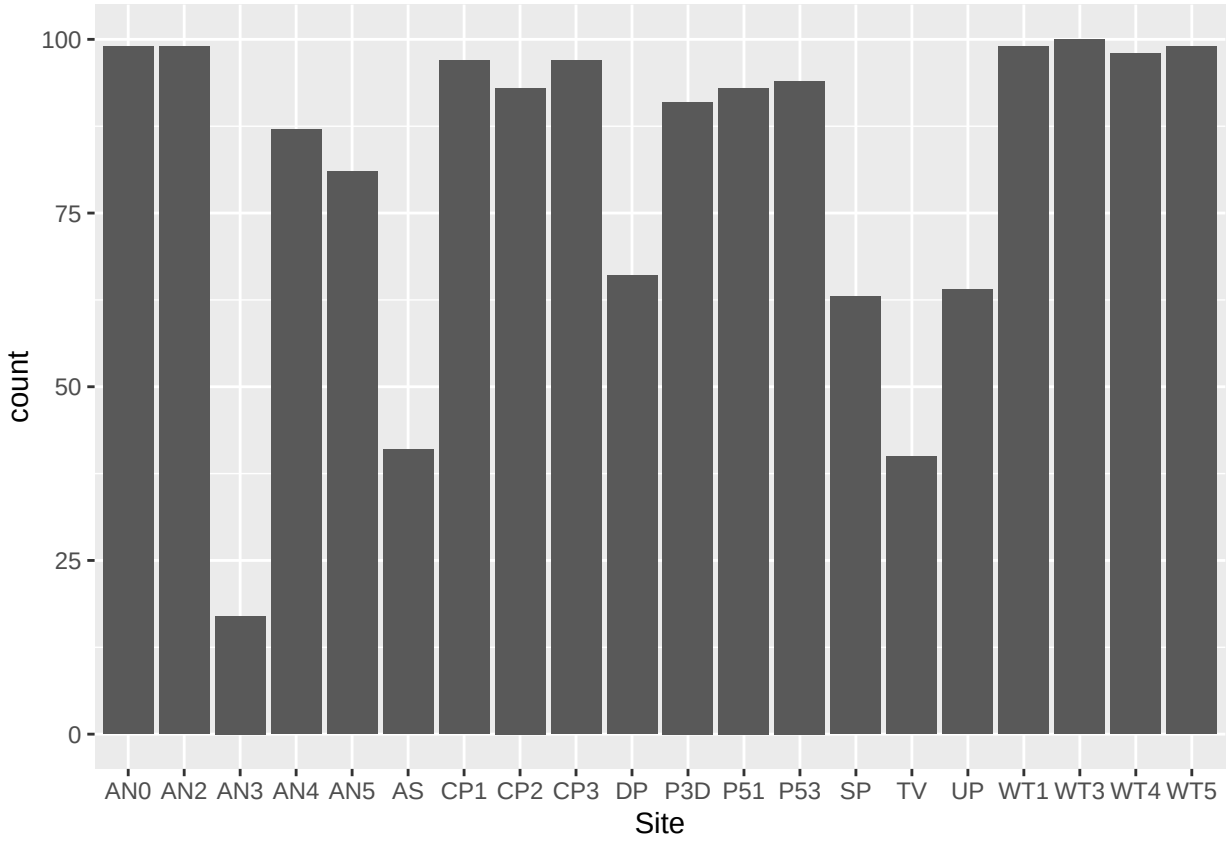

```
ggplot(wetlands, aes(x = Filtered_NHx_ugL)) +
geom_histogram(color = 'black')
```

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

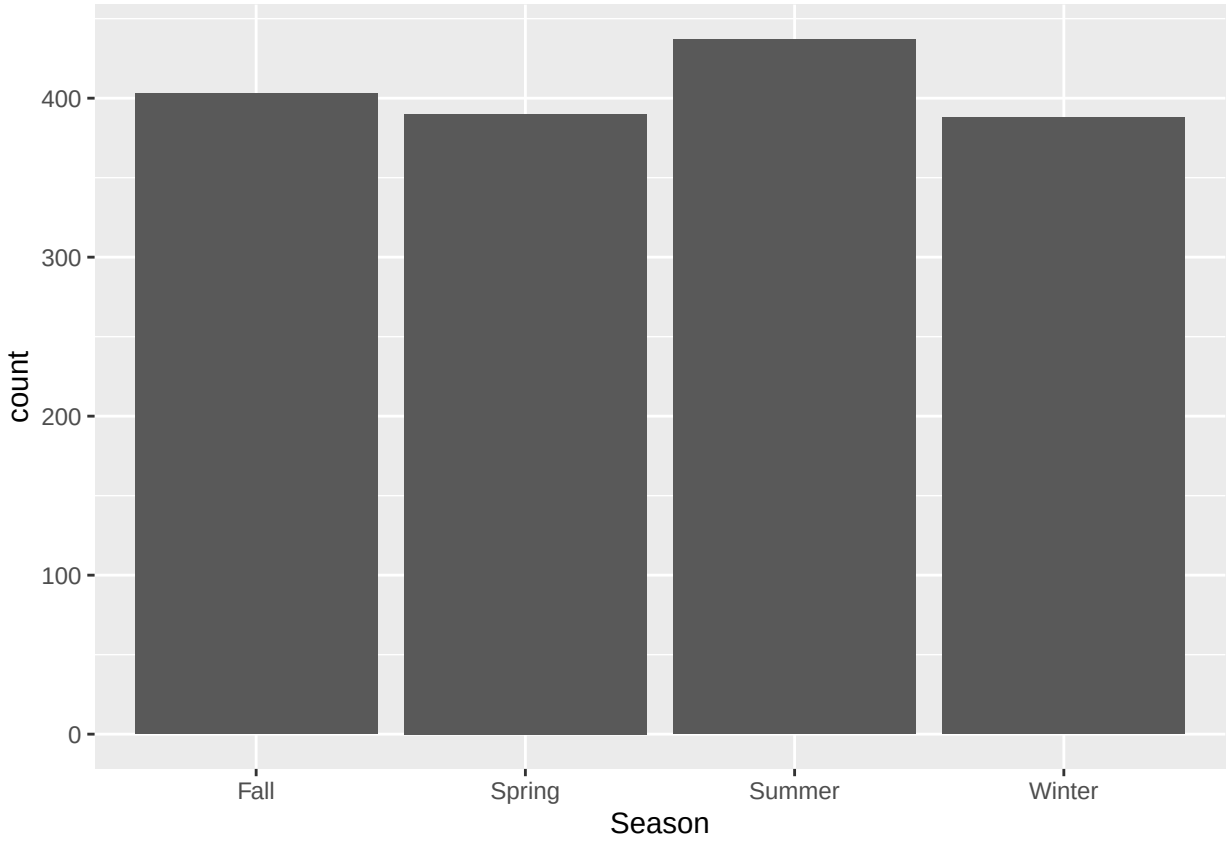
```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(wetlands, aes(x = Site)) +
geom_bar()
```



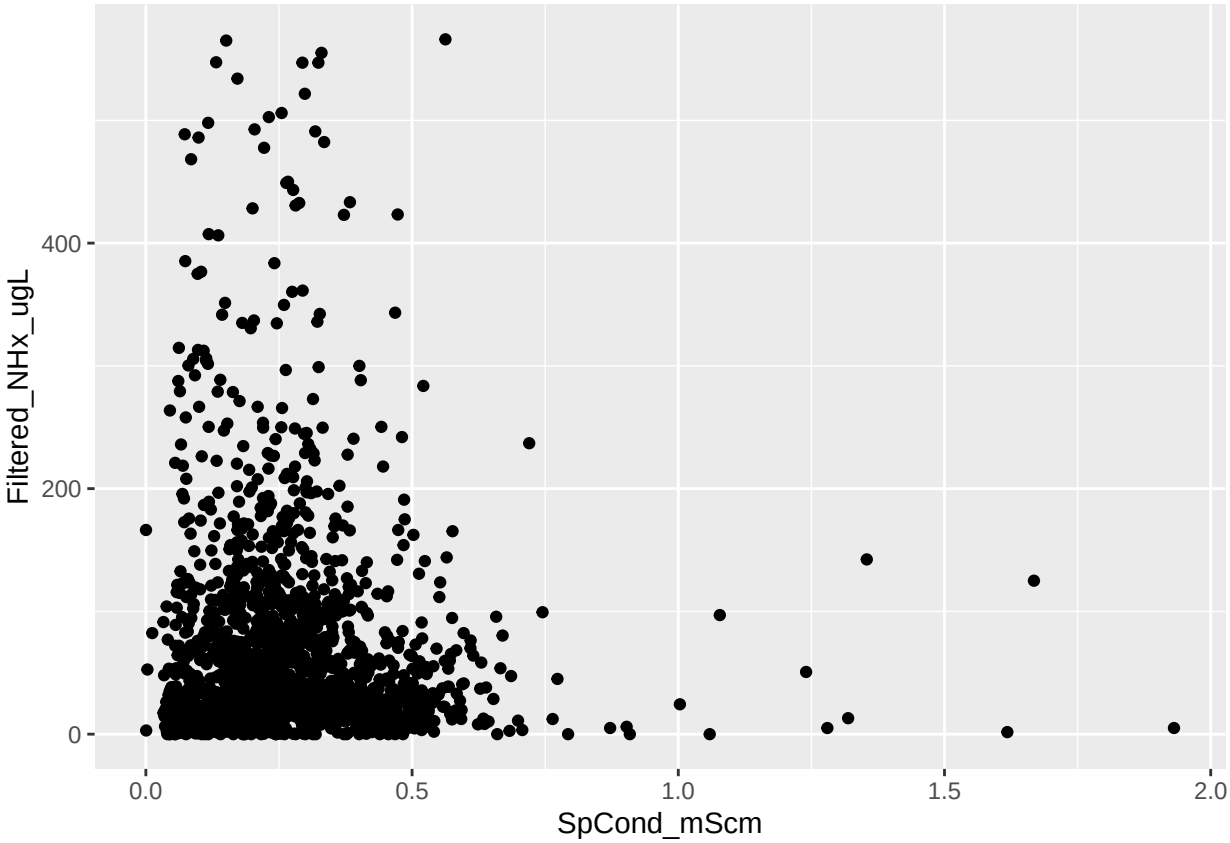

```
ggplot(wetlands, aes(x = Season)) +
geom_bar()
```



Display relationships between predictor variables and outcome variable

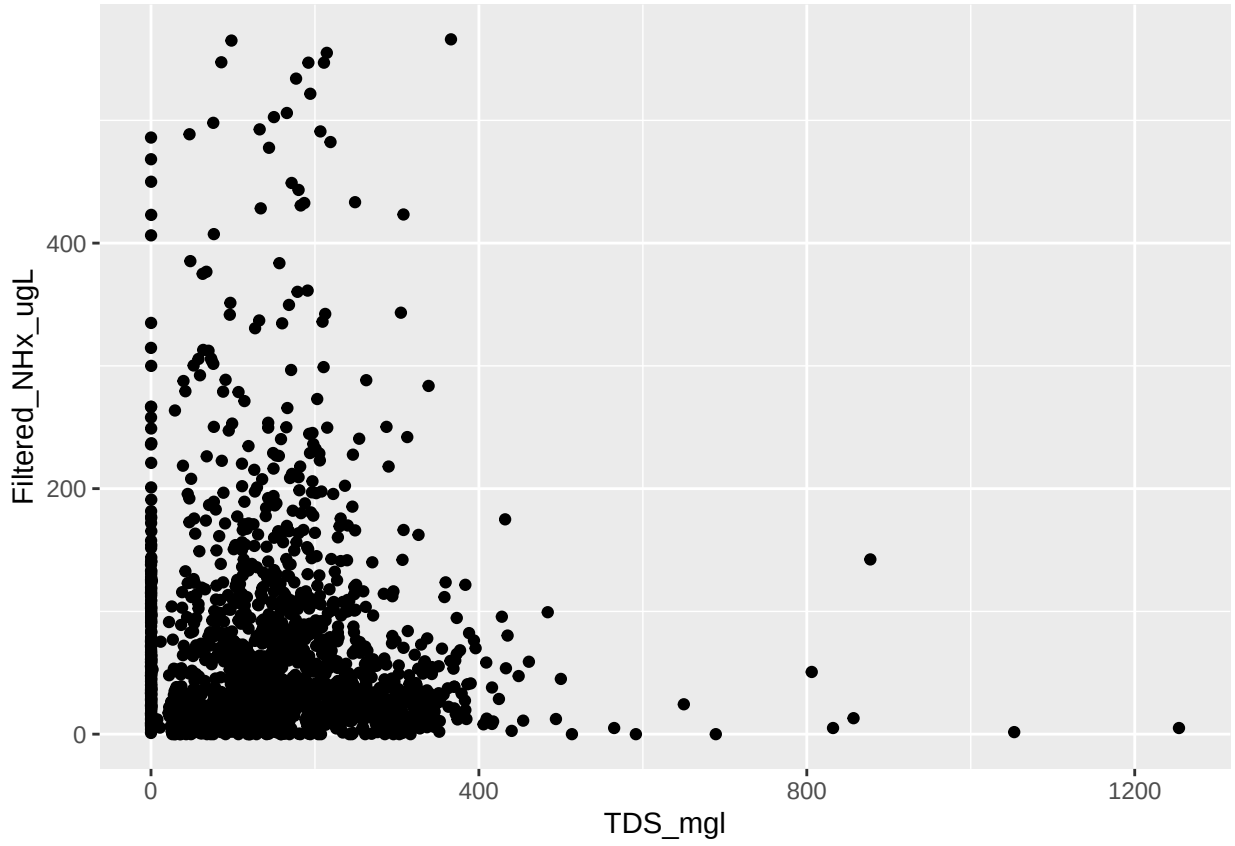
```
#create scatterplots
ggplot(wetlands, aes(x = SpCond_mScm, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



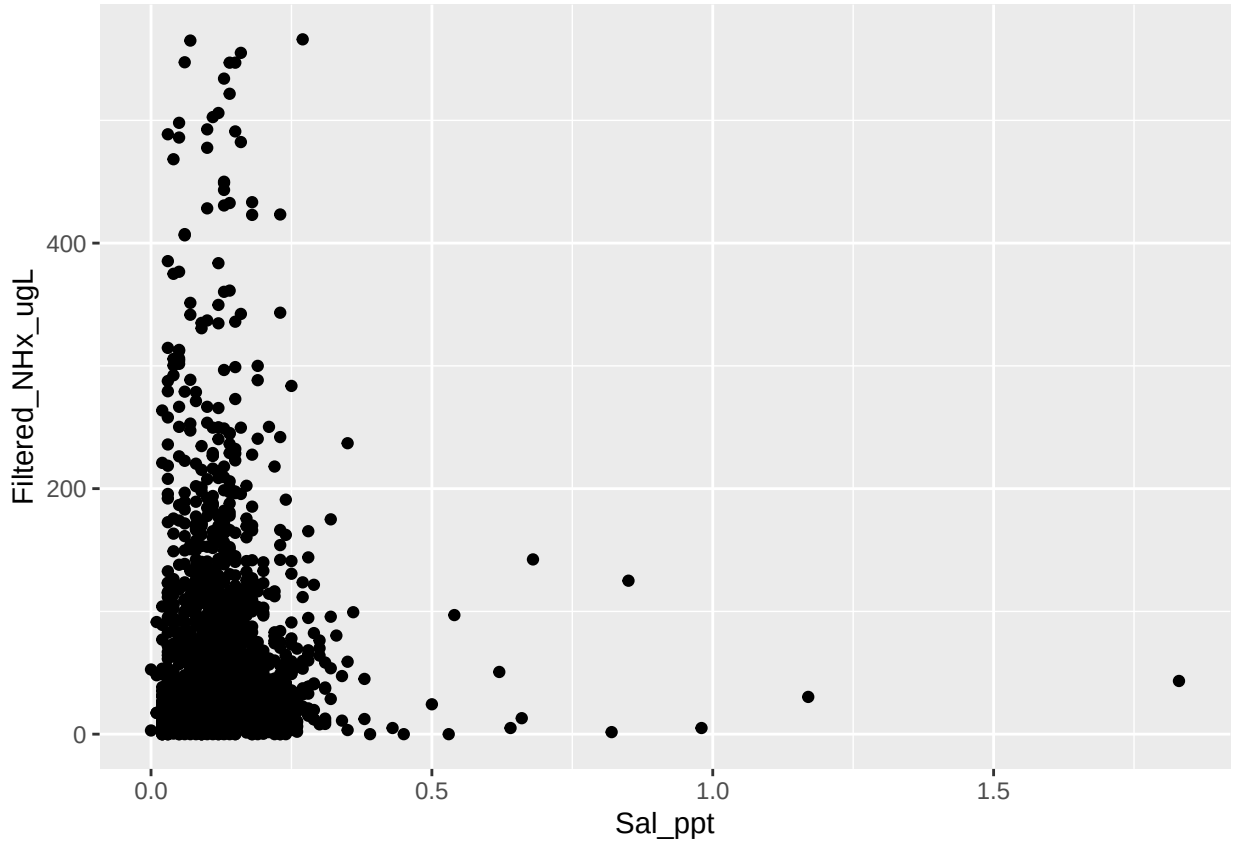

```
ggplot(wetlands, aes(x = TDS_mgl, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



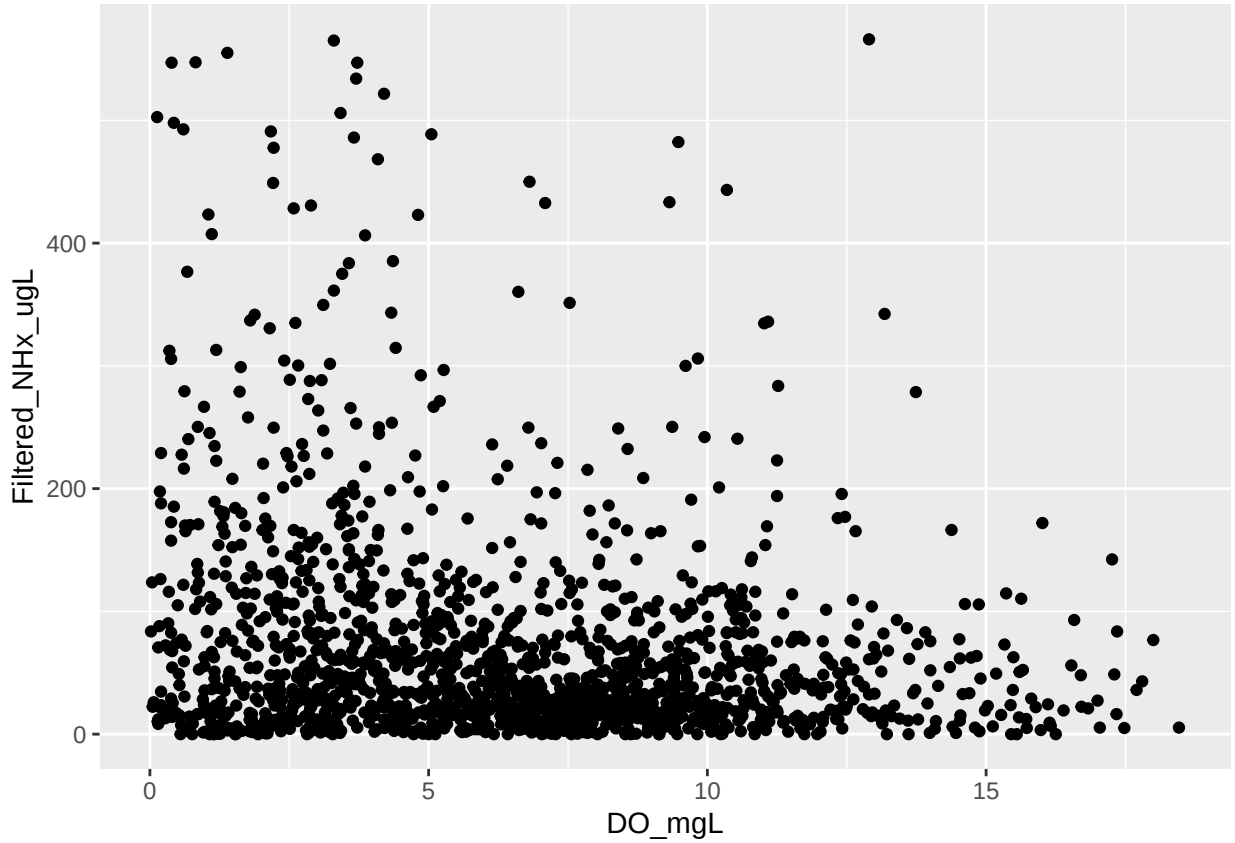

```
ggplot(wetlands, aes(x = Sal_ppt, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



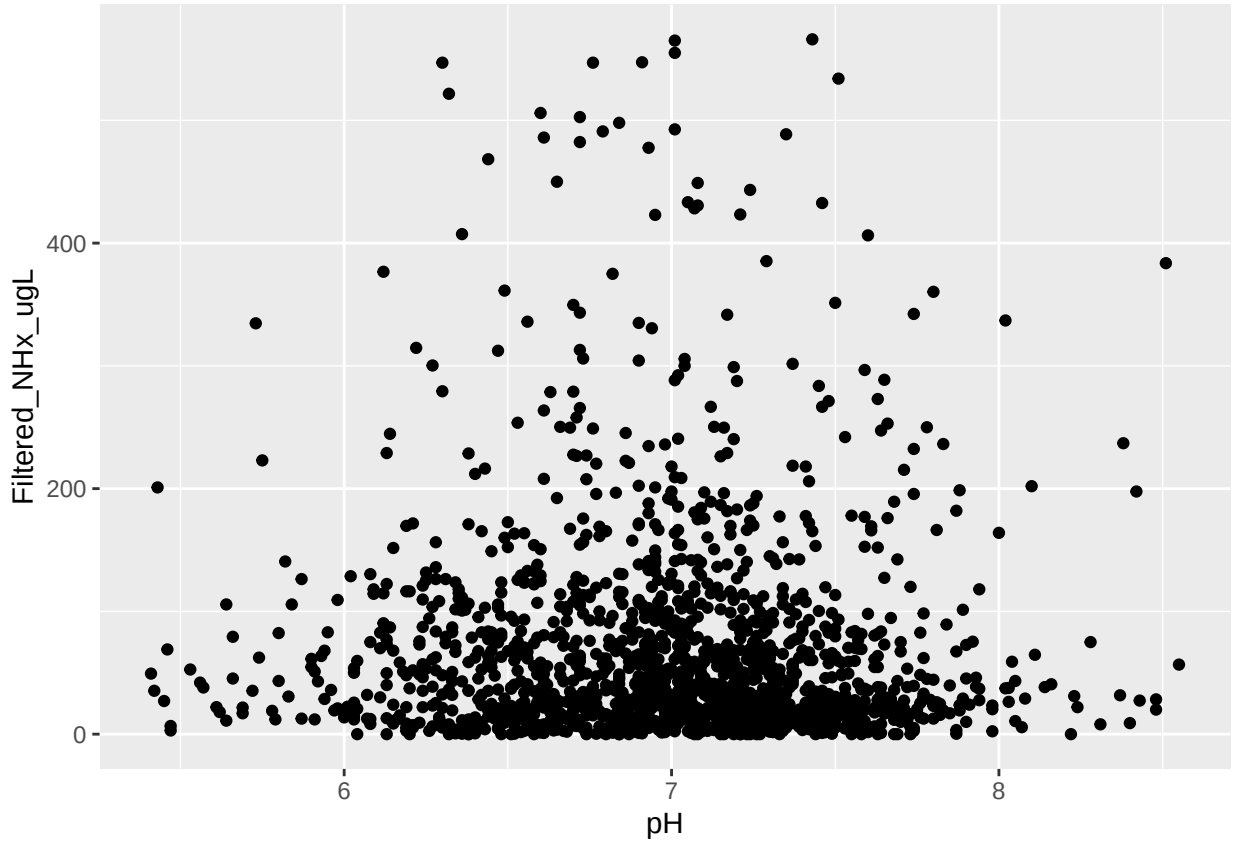

```
ggplot(wetlands, aes(x = DO_mgL, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 31 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



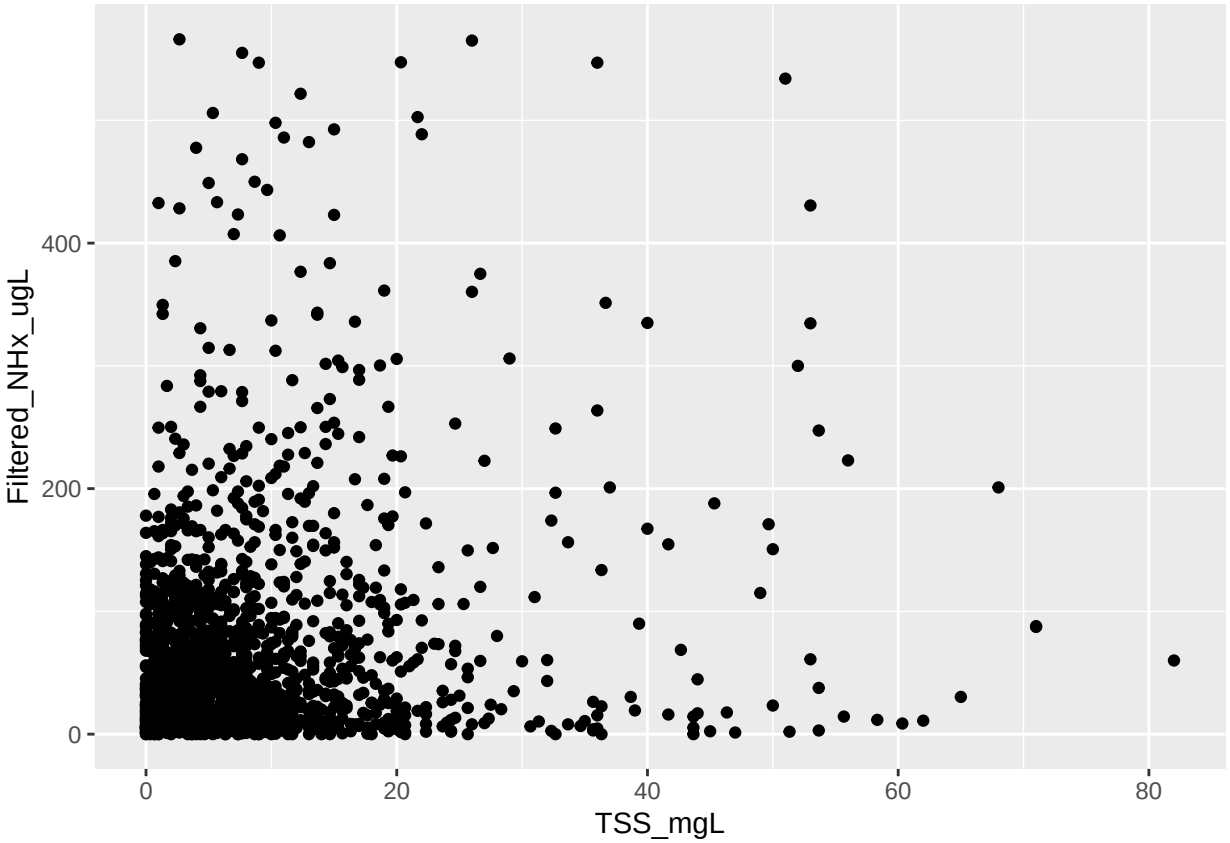

```
ggplot(wetlands, aes(x = pH, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 39 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



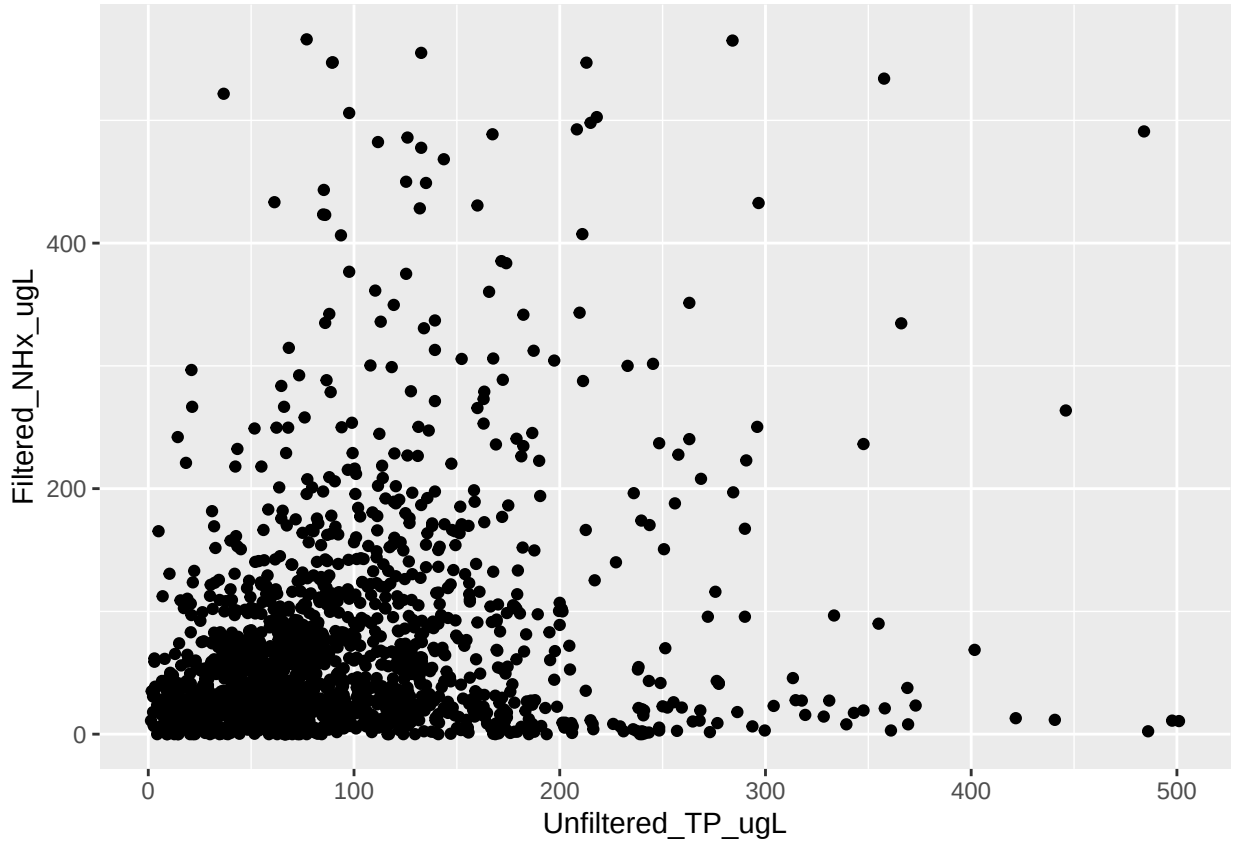

```
ggplot(wetlands, aes(x = TSS_mgL, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 33 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
ggplot(wetlands, aes(x = Unfiltered_TP_ugL, y = Filtered_NHx_ugL)) +
geom_point()
```

```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



everything needs to be log transformed? (save log-transformed data as new dataset)

```
#log transform everything except DO, and pH
#log_wetland <- wetlands %>%
 # mutate(across(c(SpCond_mScm, Cond_mScm,
                #   TDS_mgl, Sal_ppt,
                  # Filtered_NHx_ugL, TSS_mgL,
                  # Unfiltered_TP_ugL, Filtered_NOx_ugL,
                  # Unfiltered_TN_ugL),
              #   ~ log(.+1)))

#change column names
#log_transformed <- log_wetland %>%
 # rename(Log_SpCond_mScm = SpCond_mScm,
  #       Log_Cond_mScm = Cond_mScm,
   #      Log_TDS_mgl = TDS_mgl ,
    #     Log_Sal_ppt = Sal_ppt,
     #    Log_Unfiltered_TN_ugL = Unfiltered_TN_ugL,
      #   Log_Filtered_NOx_ugL = Filtered_NOx_ugL,
       #  Log_Filtered_NHx_ugL = Filtered_NHx_ugL,
        # Log_Unfiltered_TP_ugL = Unfiltered_TP_ugL,
         #Log_TSS_mgL = TSS_mgL)
```

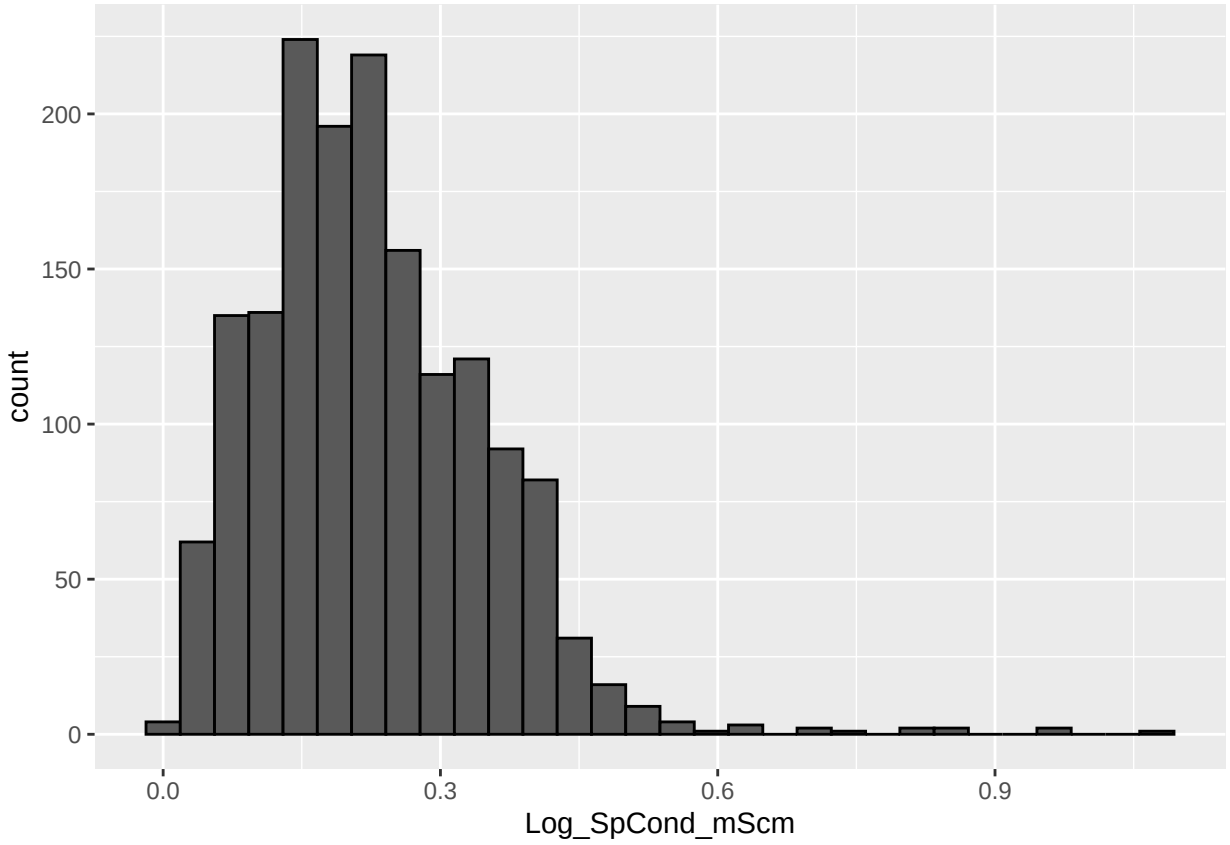
```
#save as new dataset
#write.csv(log_transformed, "Data/Processed/Log_Transformed_Data.csv", row.names = FALSE)
```

re-visualize data

```
#load new dataset
log_wetland <- read.csv(here('Data/Processed/Log_Transformed_Data.csv'))

#create histograms of all log-transformed variables
ggplot(log_wetland, aes(x = Log_SpCond_mScm)) +
geom_histogram(color = 'black')
```

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

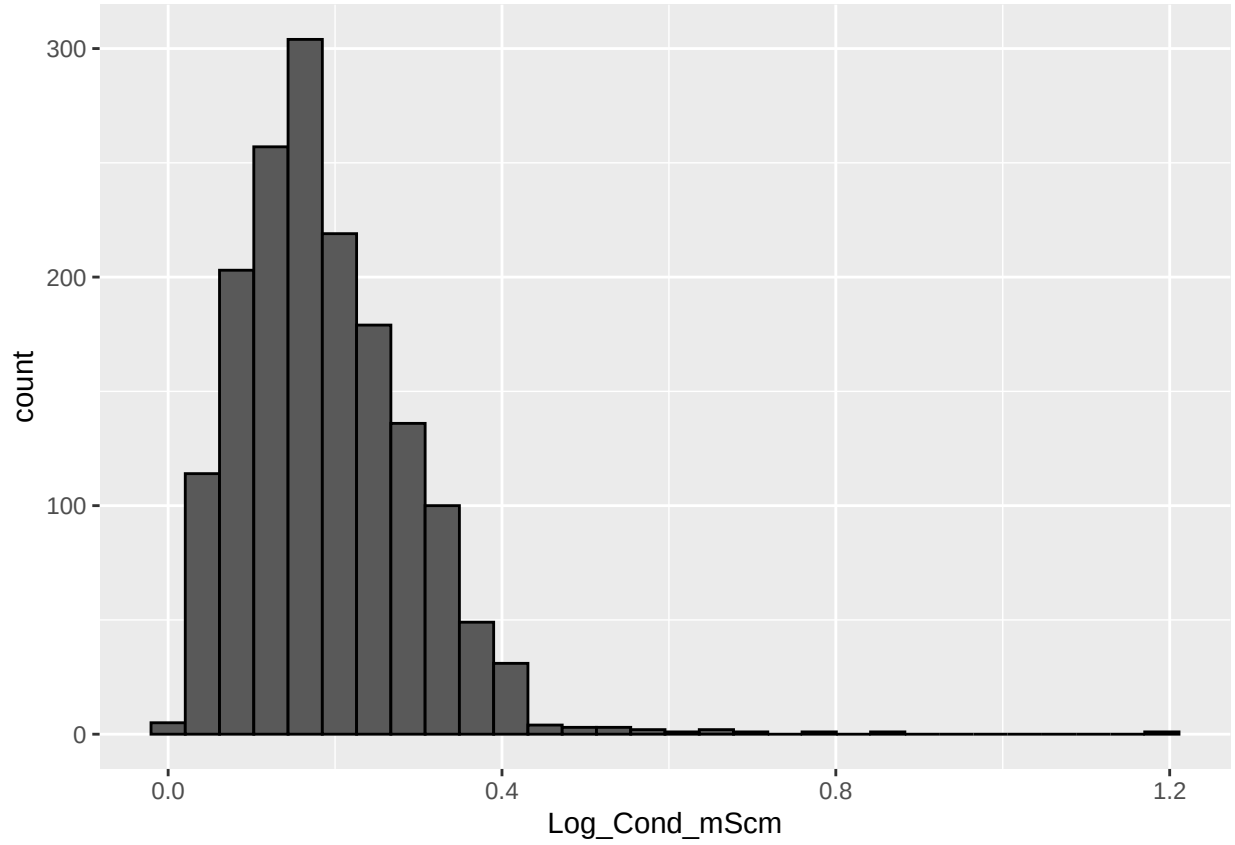
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(log_wetland, aes(x = Log_Cond_mScm)) +
geom_histogram(color = 'black')
```

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

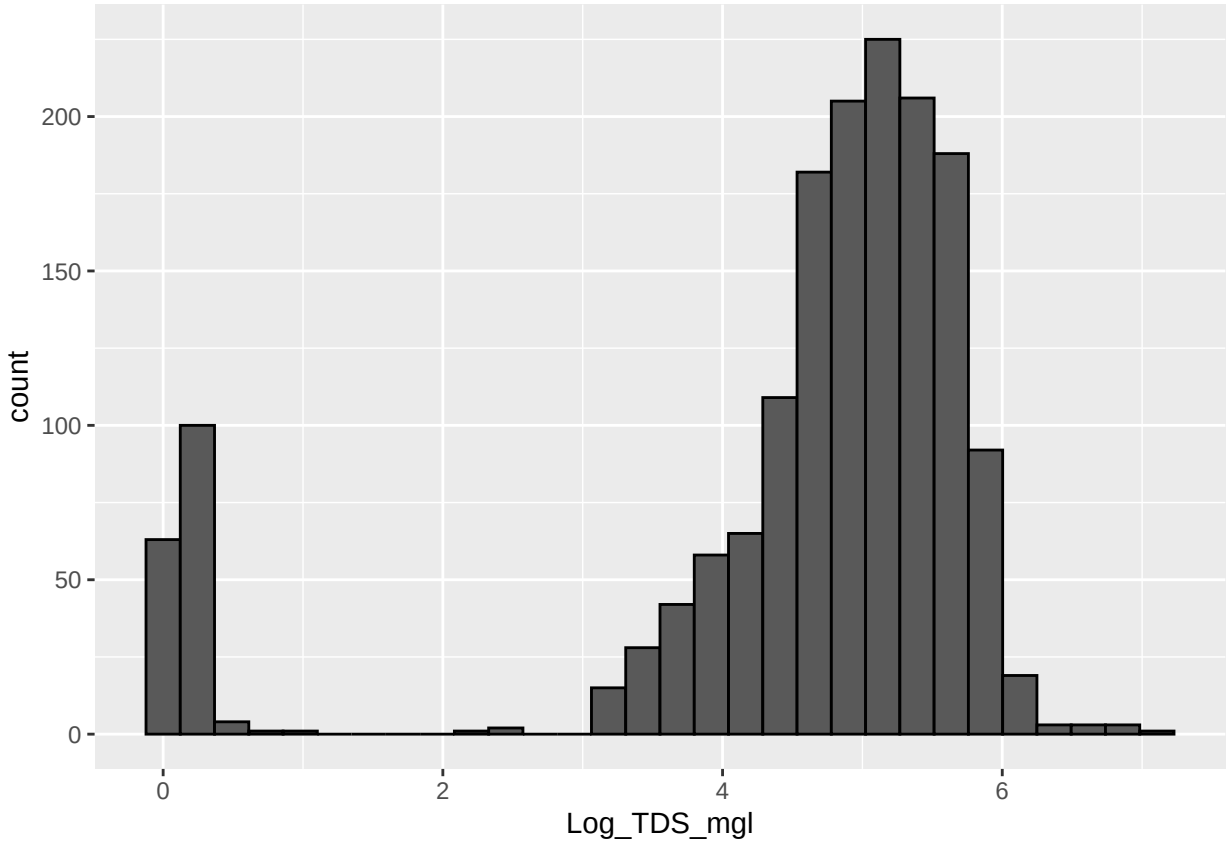
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```




```
ggplot(log_wetland, aes(x = Log_TDS_mgl)) +
geom_histogram(color = 'black')
```

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

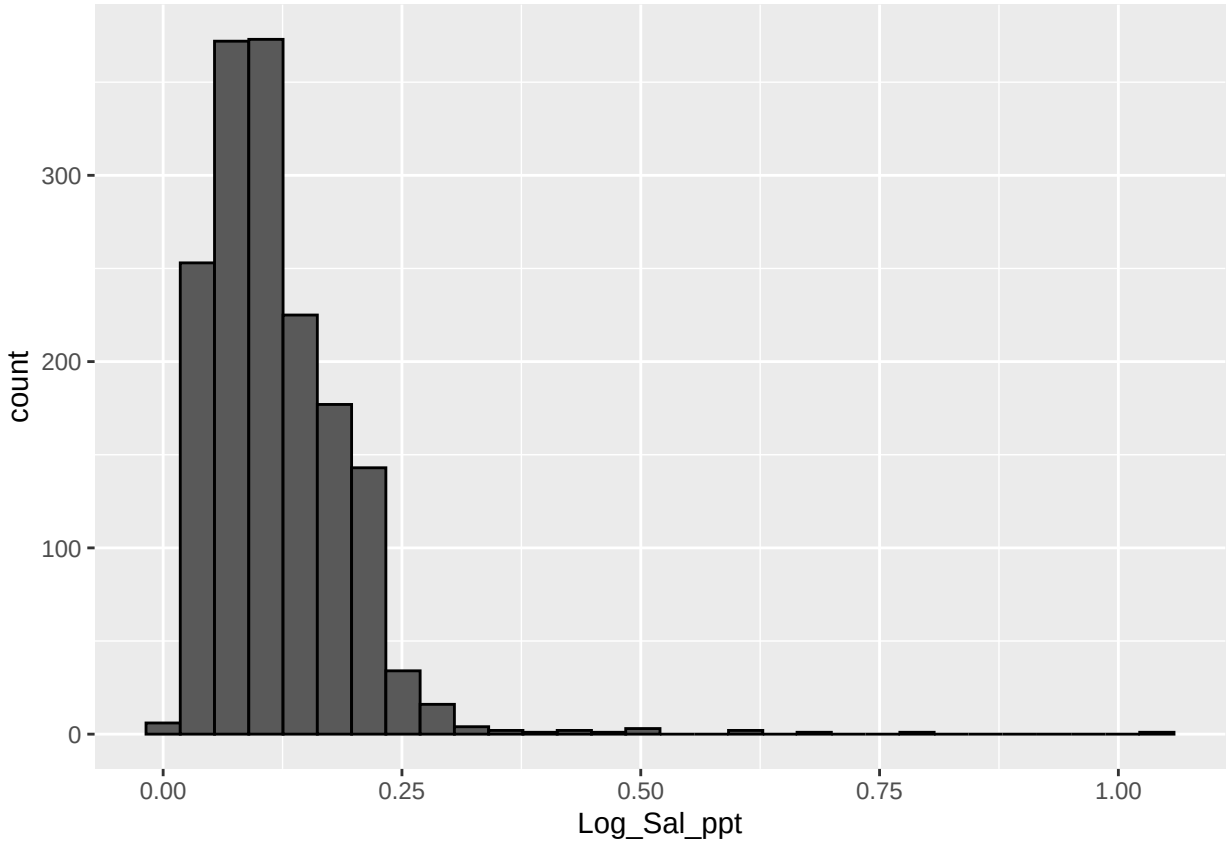
```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```




```
ggplot(log_wetland, aes(x = Log_Sal_ppt)) +
geom_histogram(color = 'black')
```

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

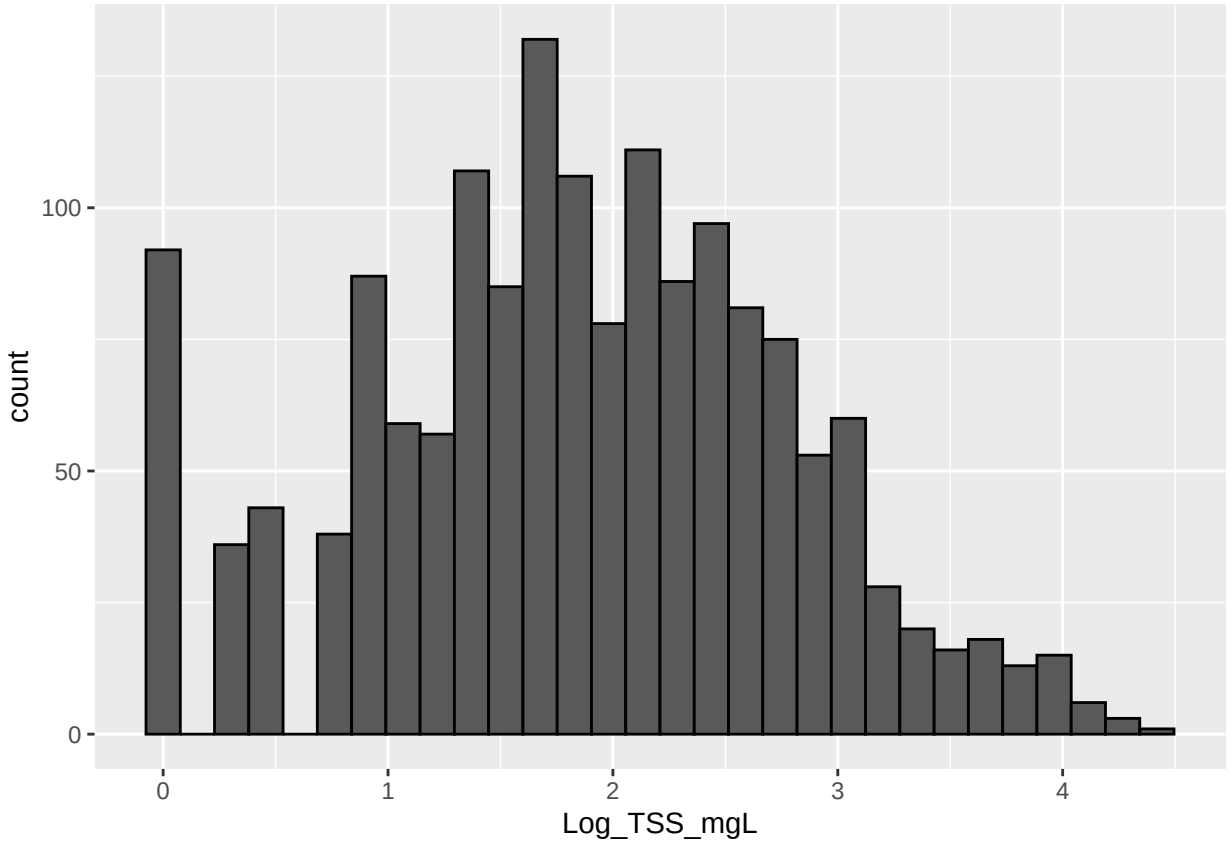
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(log_wetland, aes(x = Log_TSS_mgL)) +
geom_histogram(color = 'black')
```

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

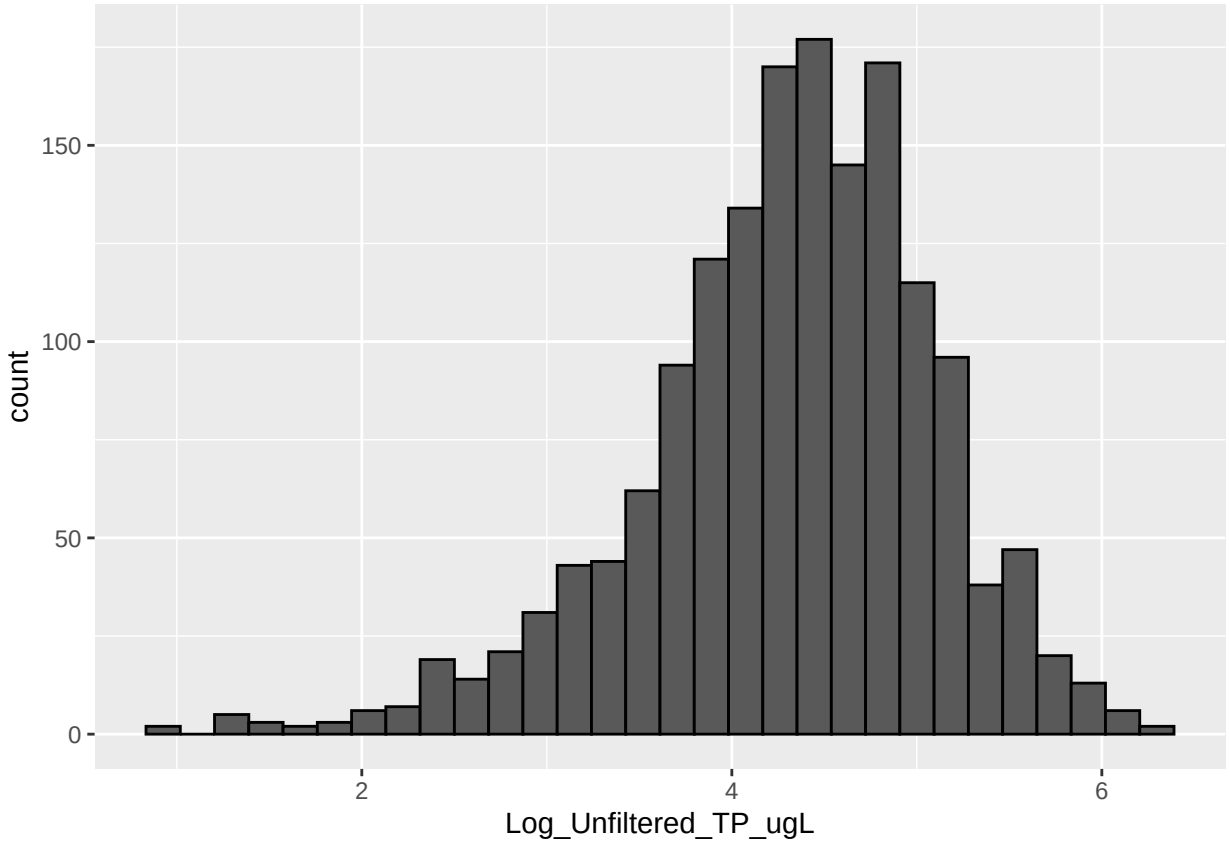
```
## Warning: Removed 15 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(log_wetland, aes(x = Log_Unfiltered_TP_ugL)) +
geom_histogram(color = 'black')
```

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

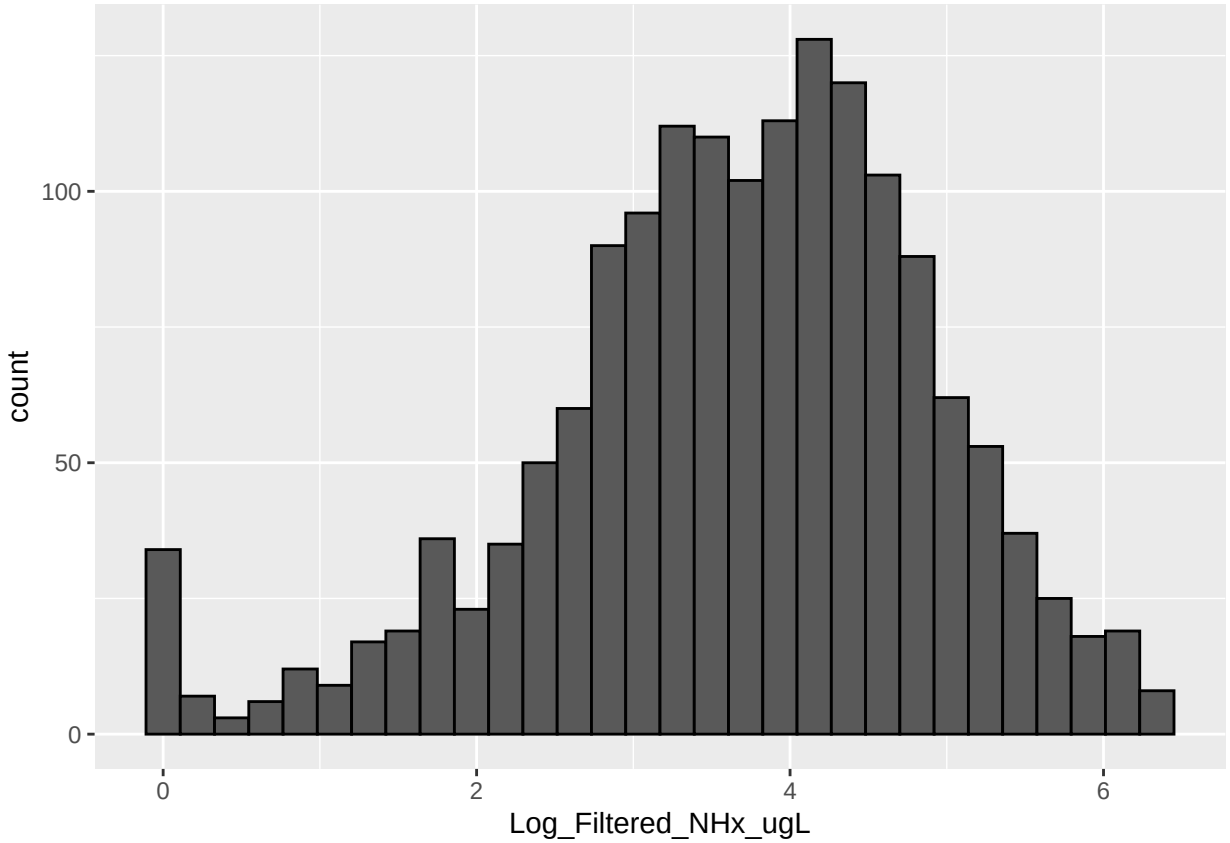
```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
ggplot(log_wetland, aes(x = Log_Filtered_NHx_ugL)) +
geom_histogram(color = 'black')
```

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

```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

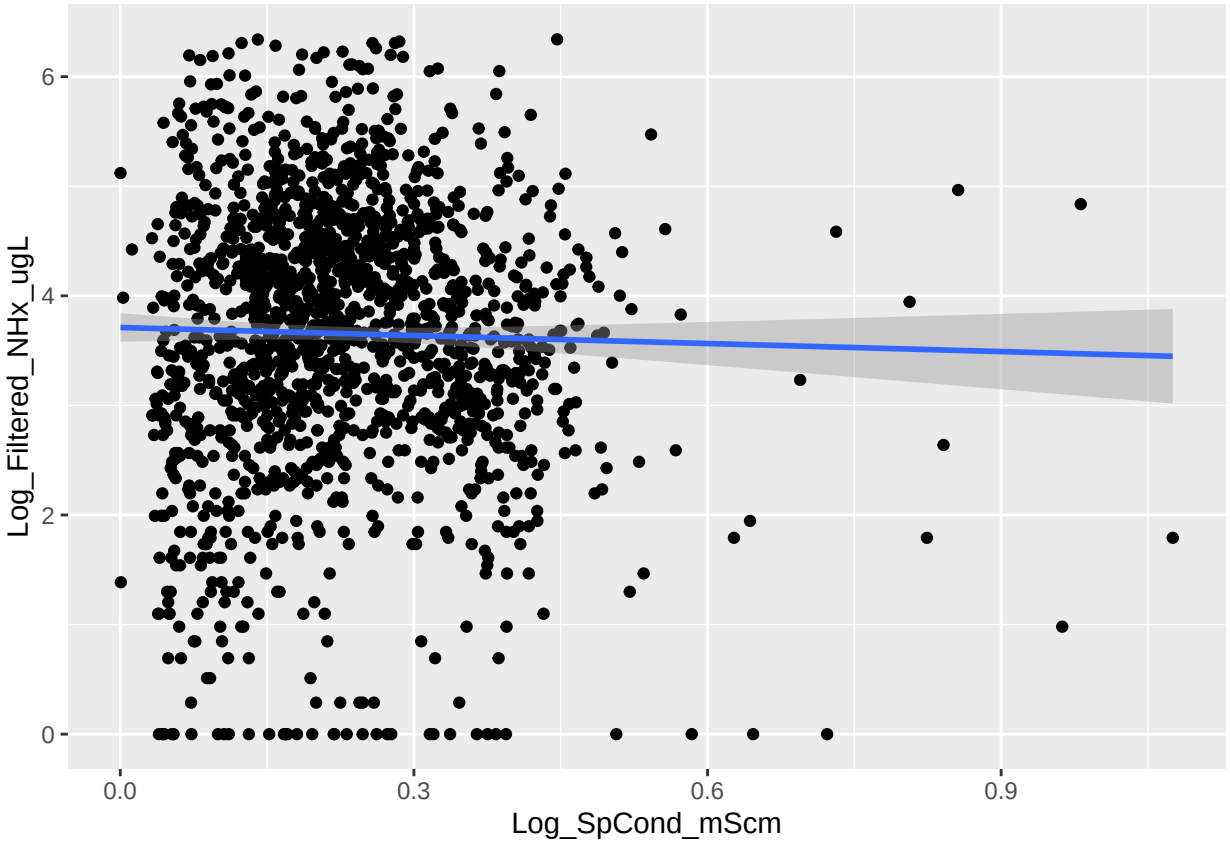



```
#create scatterplots of predictor vs outcome var
ggplot(log_wetland, aes(x = Log_SpCond_mScm, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

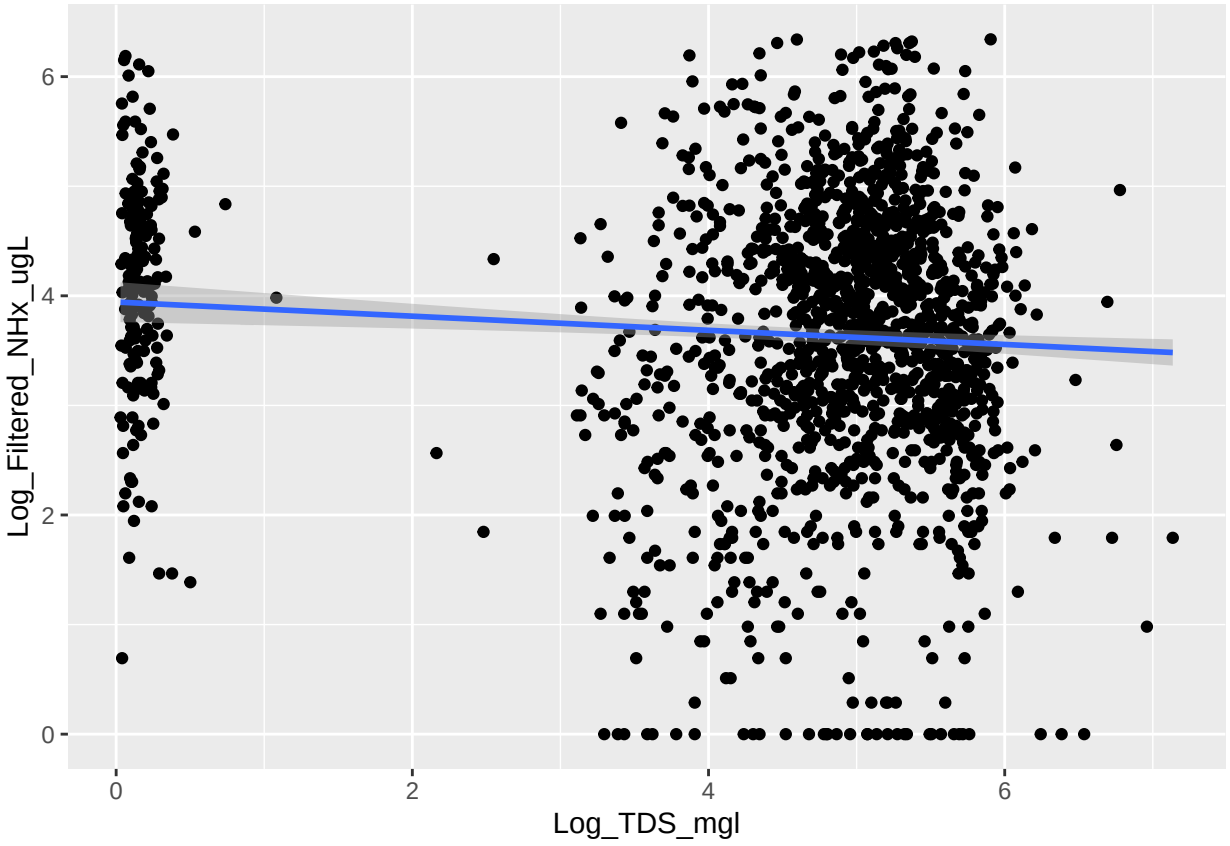



```
ggplot(log_wetland, aes(x = Log_TDS_mgl, y = Log_Filtered_NHx_ugL)) +
geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 24 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

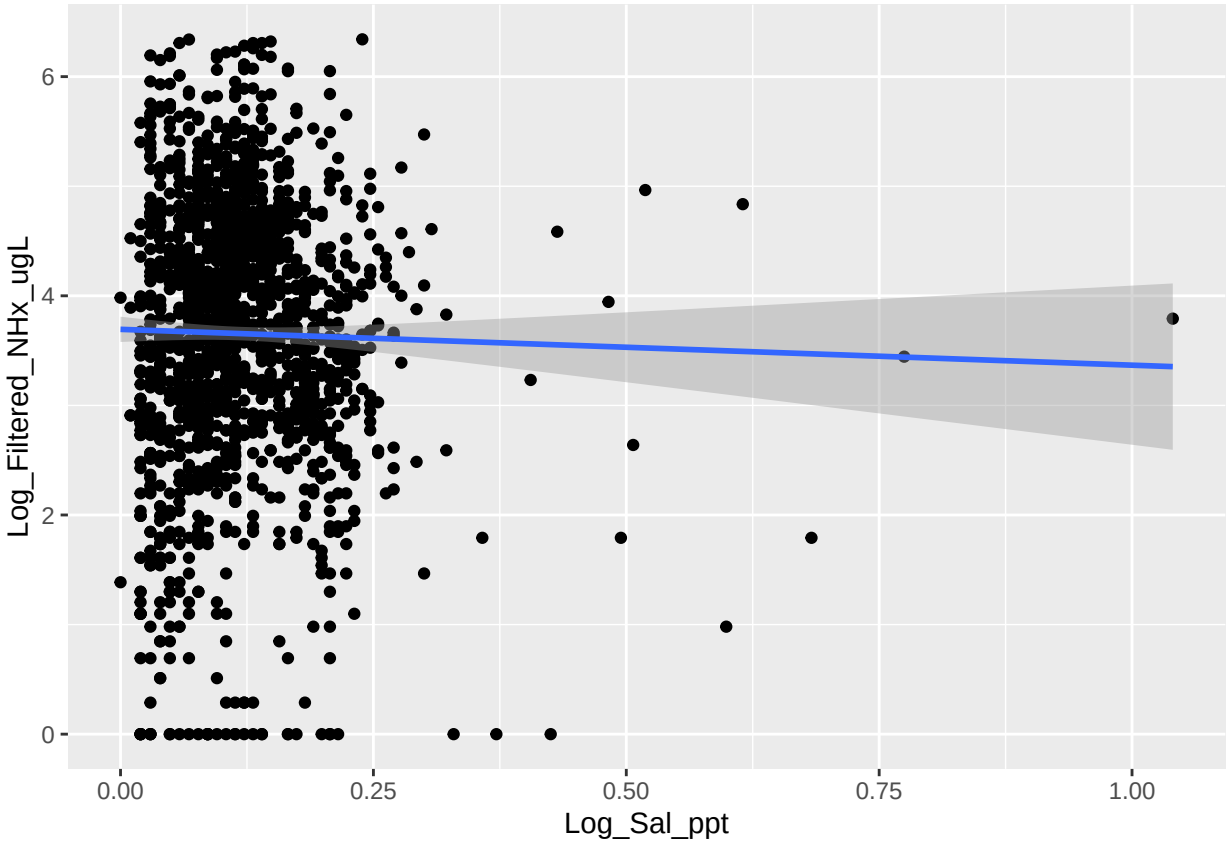



```
ggplot(log_wetland, aes(x = Log_Sal_ppt, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

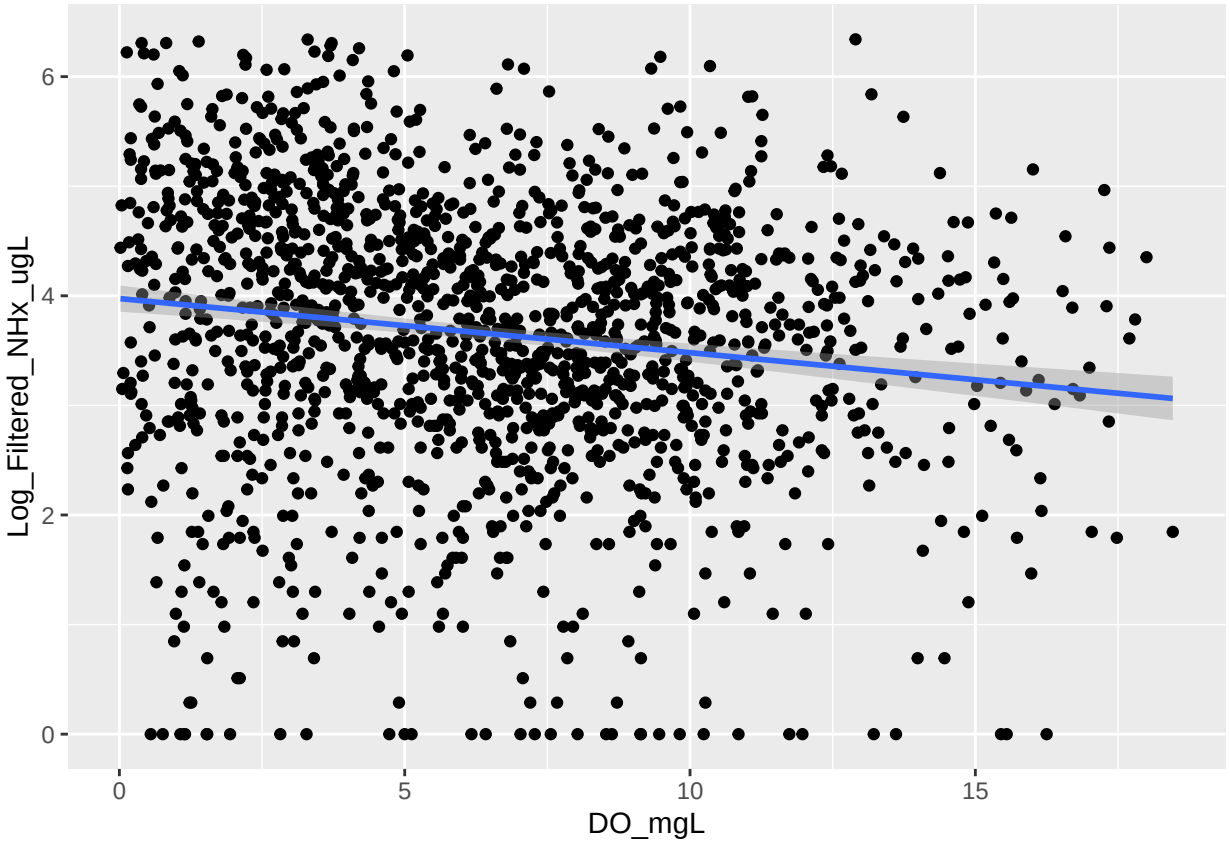



```
ggplot(log_wetland, aes(x = DO_mgL, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 31 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 31 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

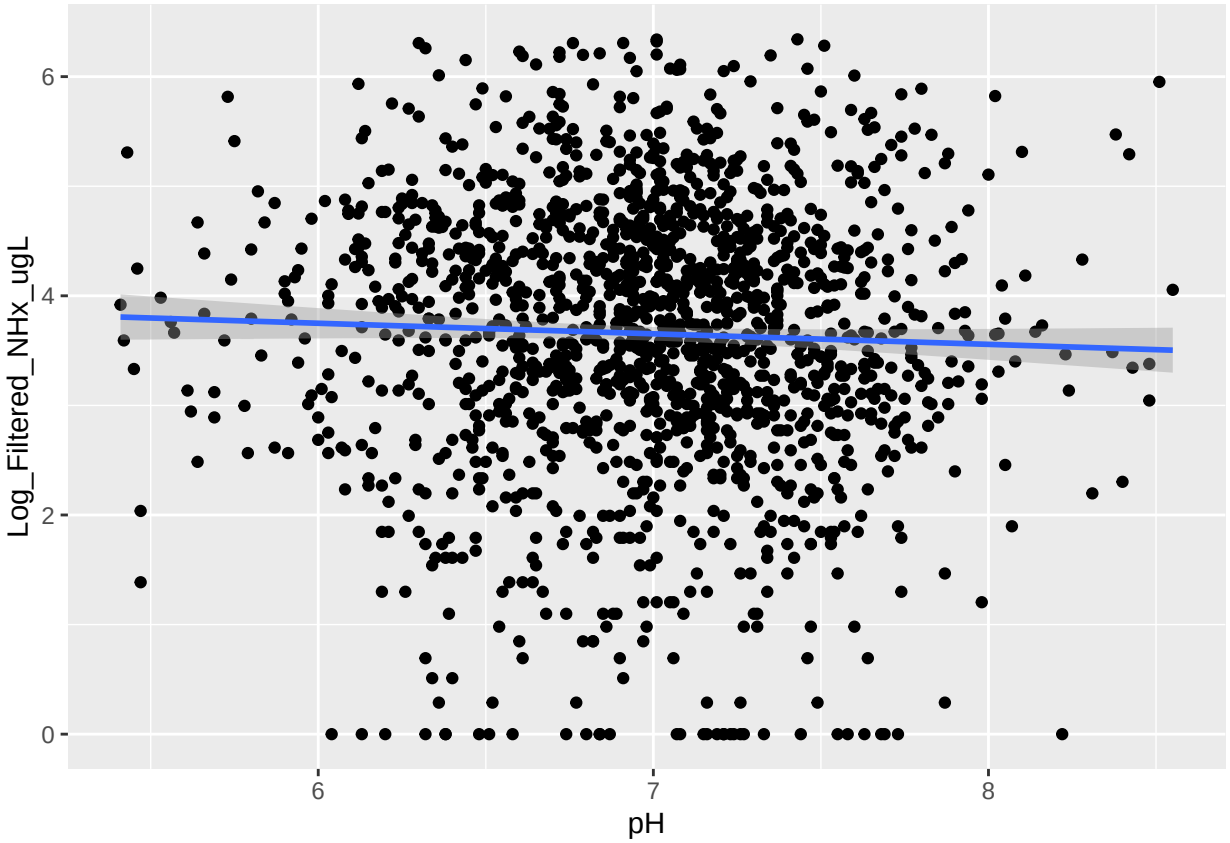



```
ggplot(log_wetland, aes(x = pH, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 39 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 39 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

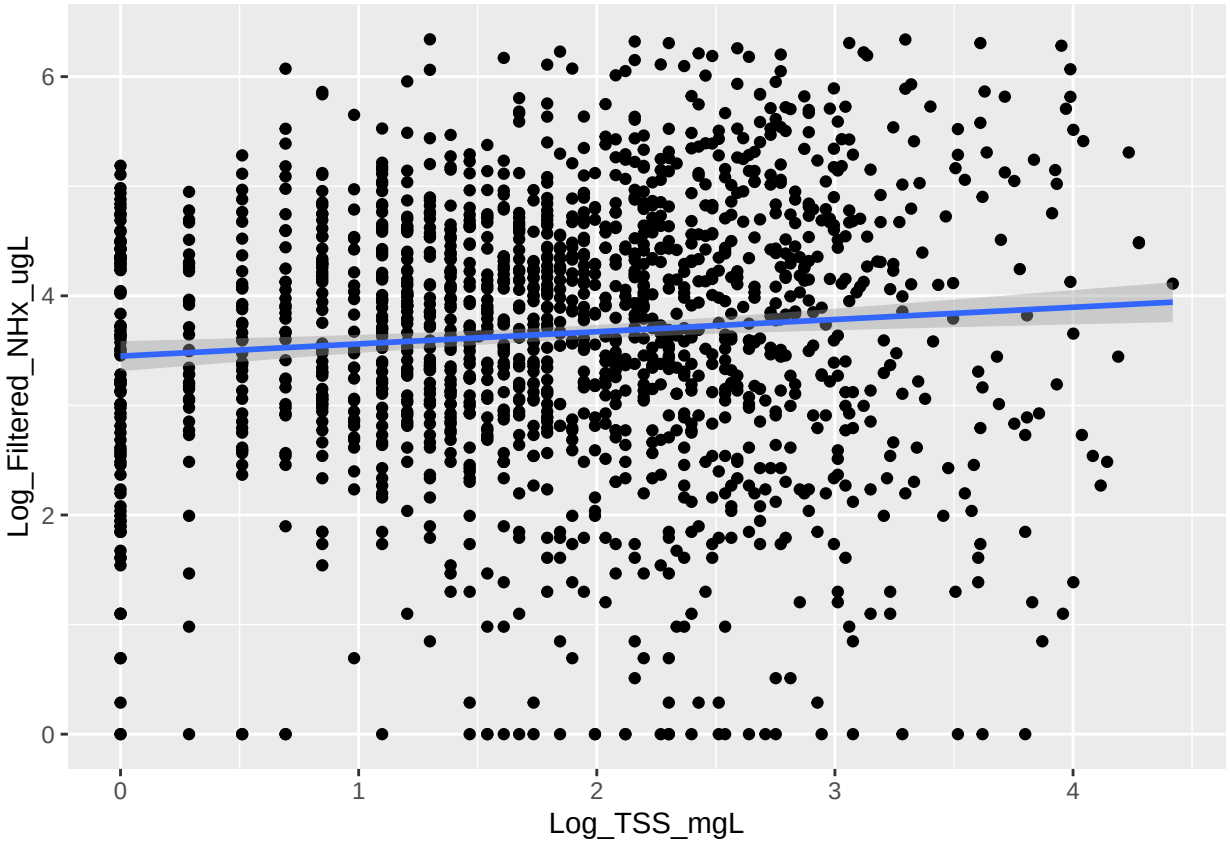



```
ggplot(log_wetland, aes(x = Log_TSS_mgL, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 33 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 33 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

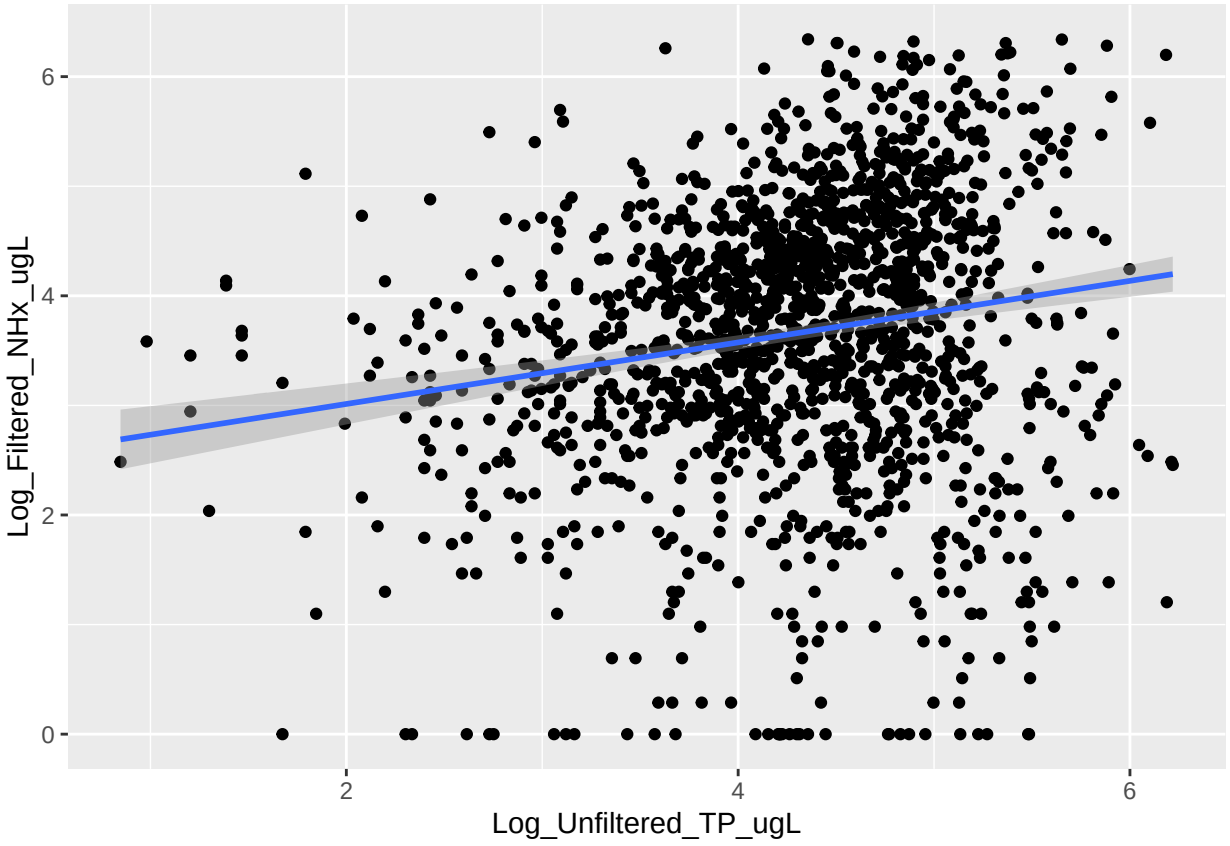



```
ggplot(log_wetland, aes(x = Log_Unfiltered_TP_ugL, y = Log_Filtered_NHx_ugL)) +
geom_point() +
geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 26 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

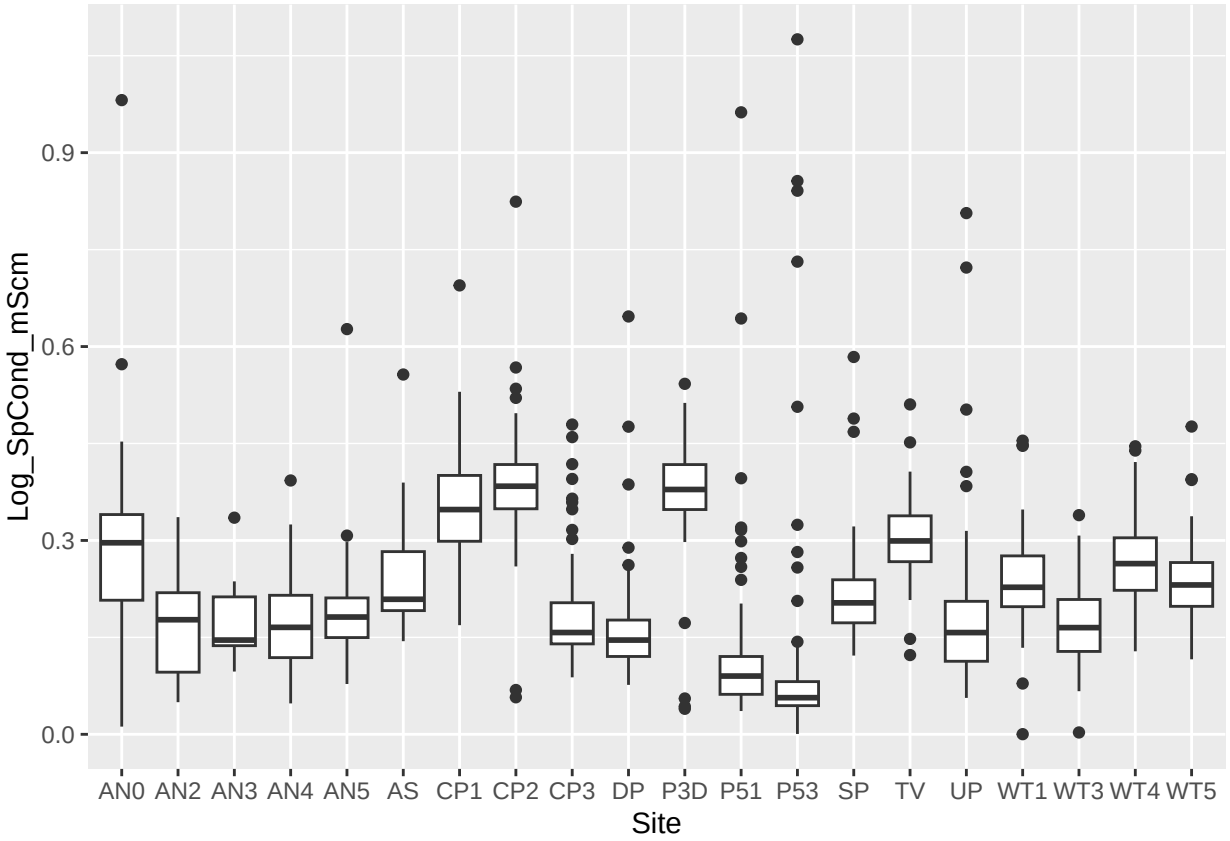
```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Display fixed effect variables across random effect variable

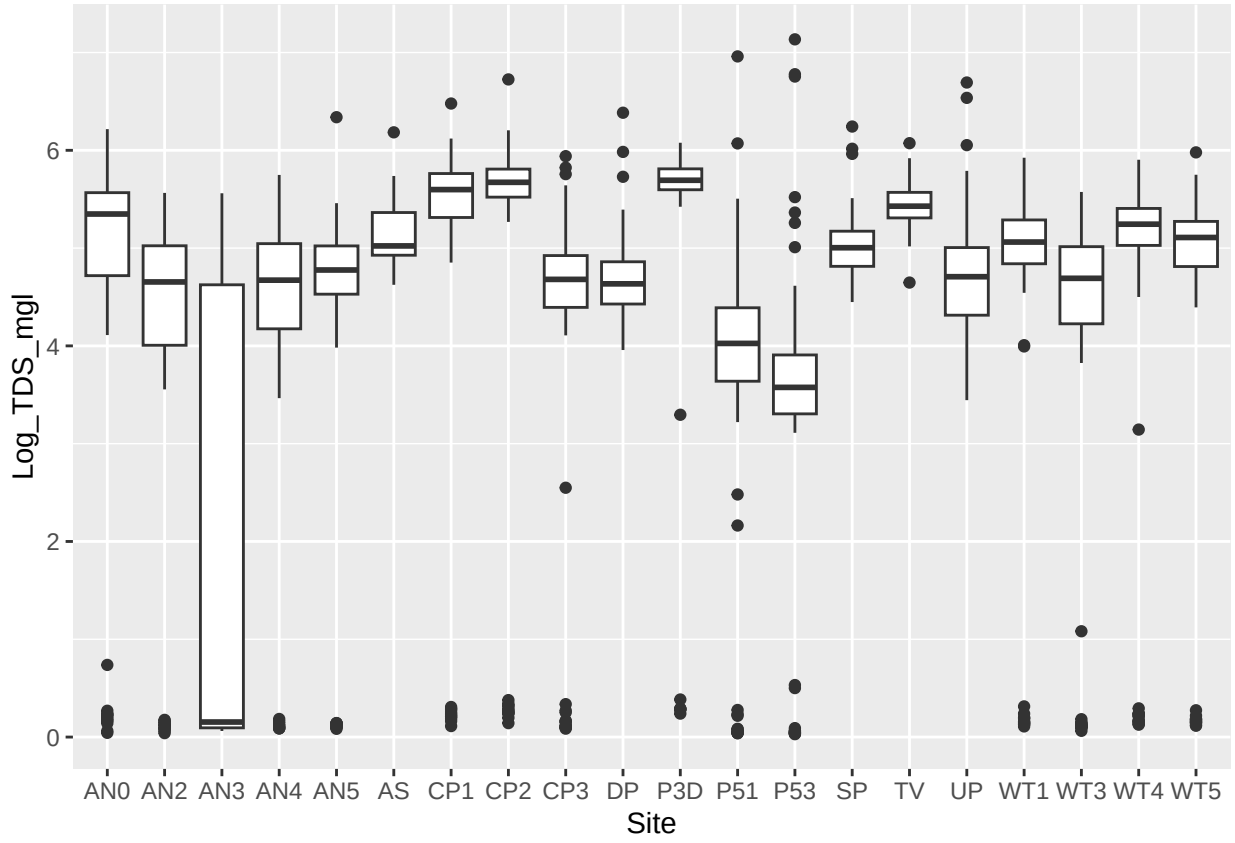
```
#create boxplots of each fixed effect variable
ggplot(log_wetland, aes(x = Site, y = Log_SpCond_mScm)) +
geom_boxplot()
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



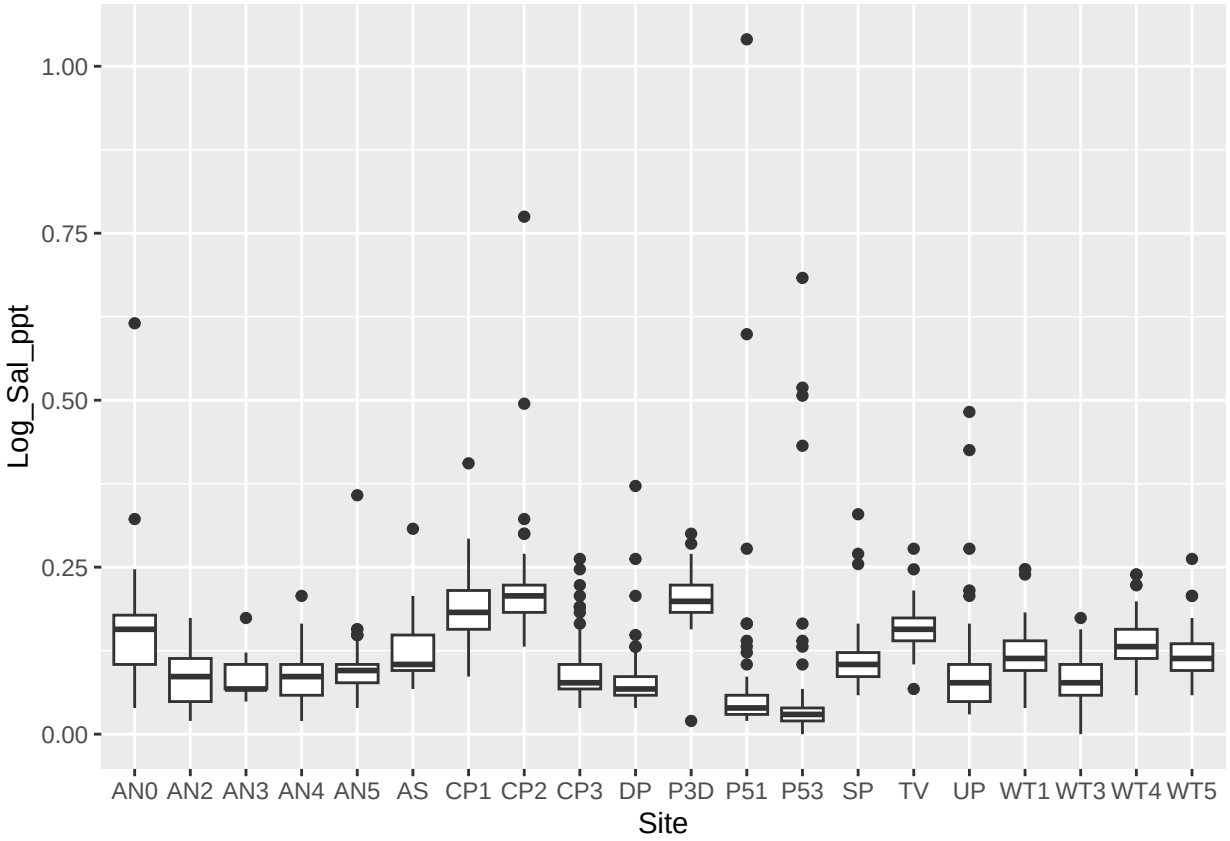

```
ggplot(log_wetland, aes(x = Site, y = Log_TDS_mgl)) +
geom_boxplot()
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



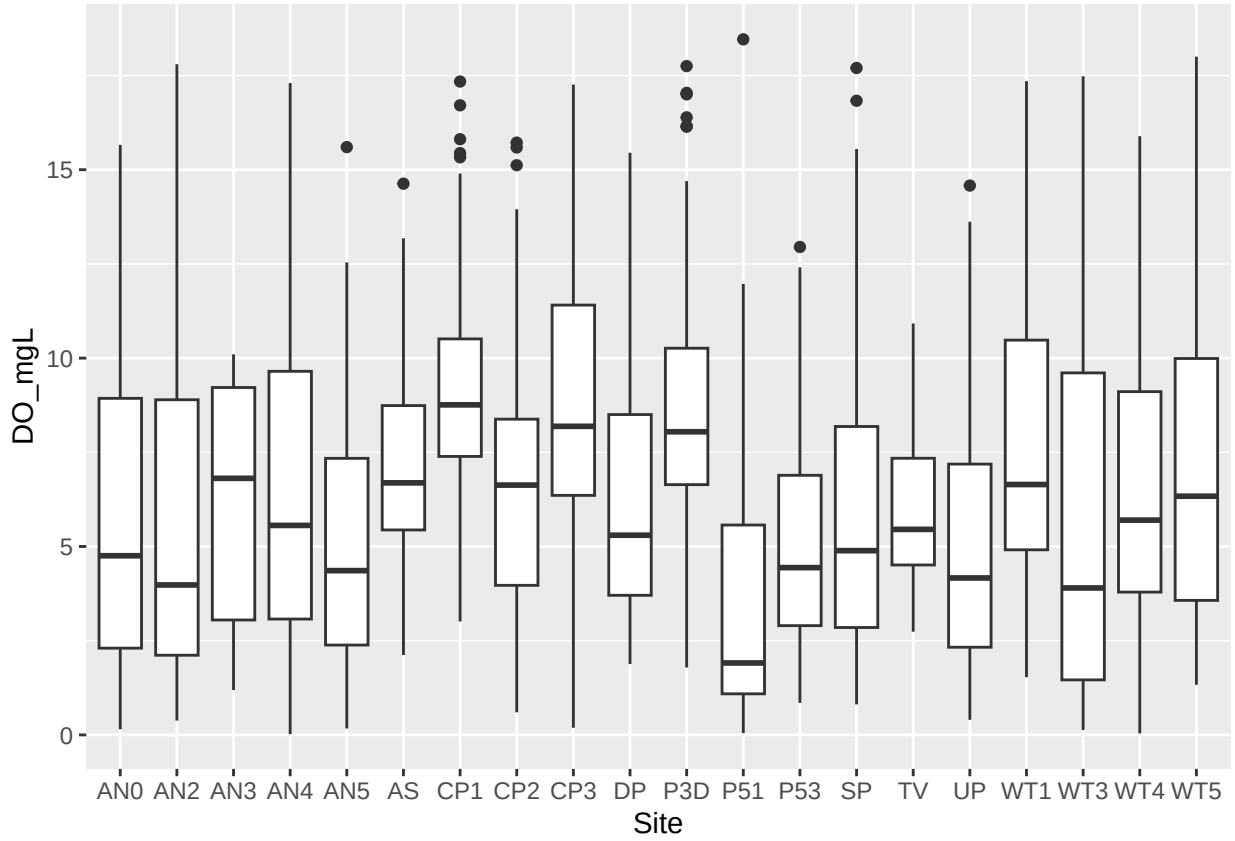

```
ggplot(log_wetland, aes(x = Site, y = Log_Sal_ppt)) +
geom_boxplot()
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



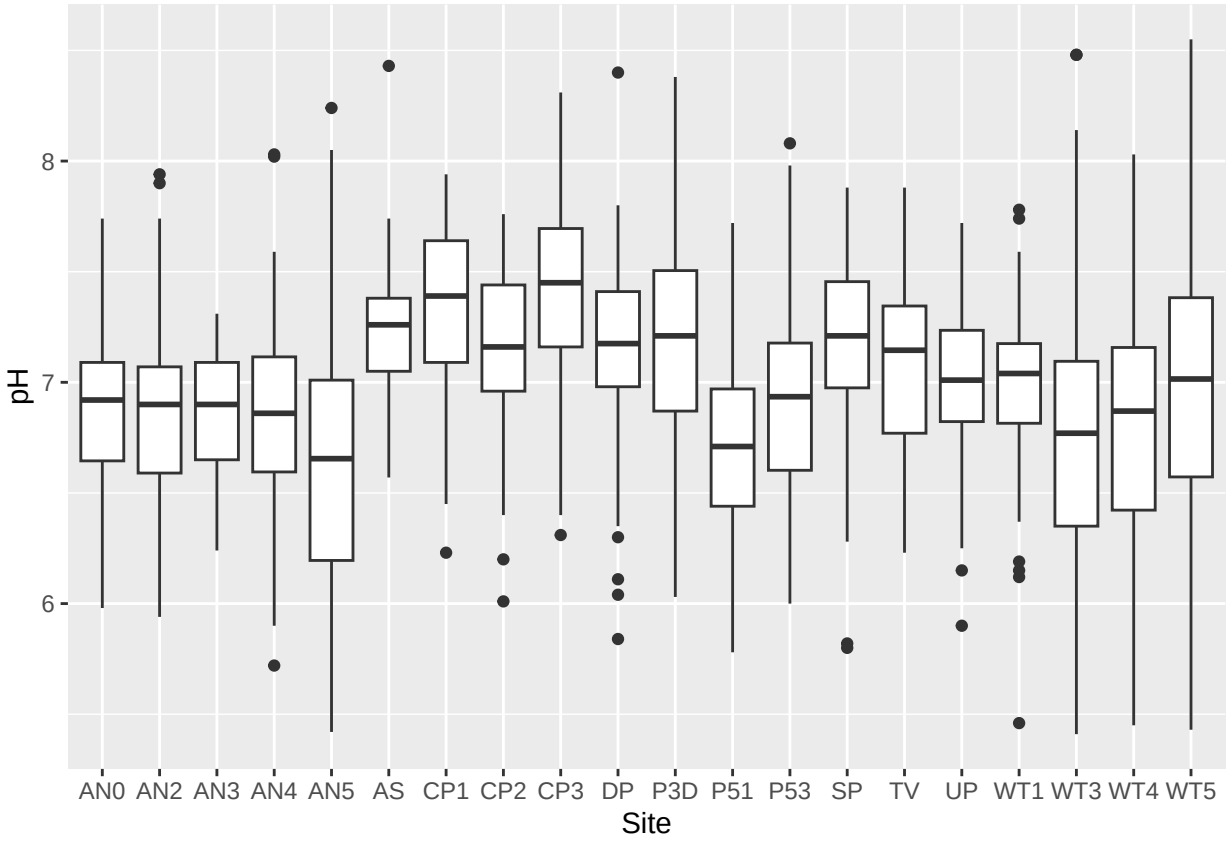

```
ggplot(log_wetland, aes(x = Site, y = DO_mgL)) +
geom_boxplot()
```

```
## Warning: Removed 8 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



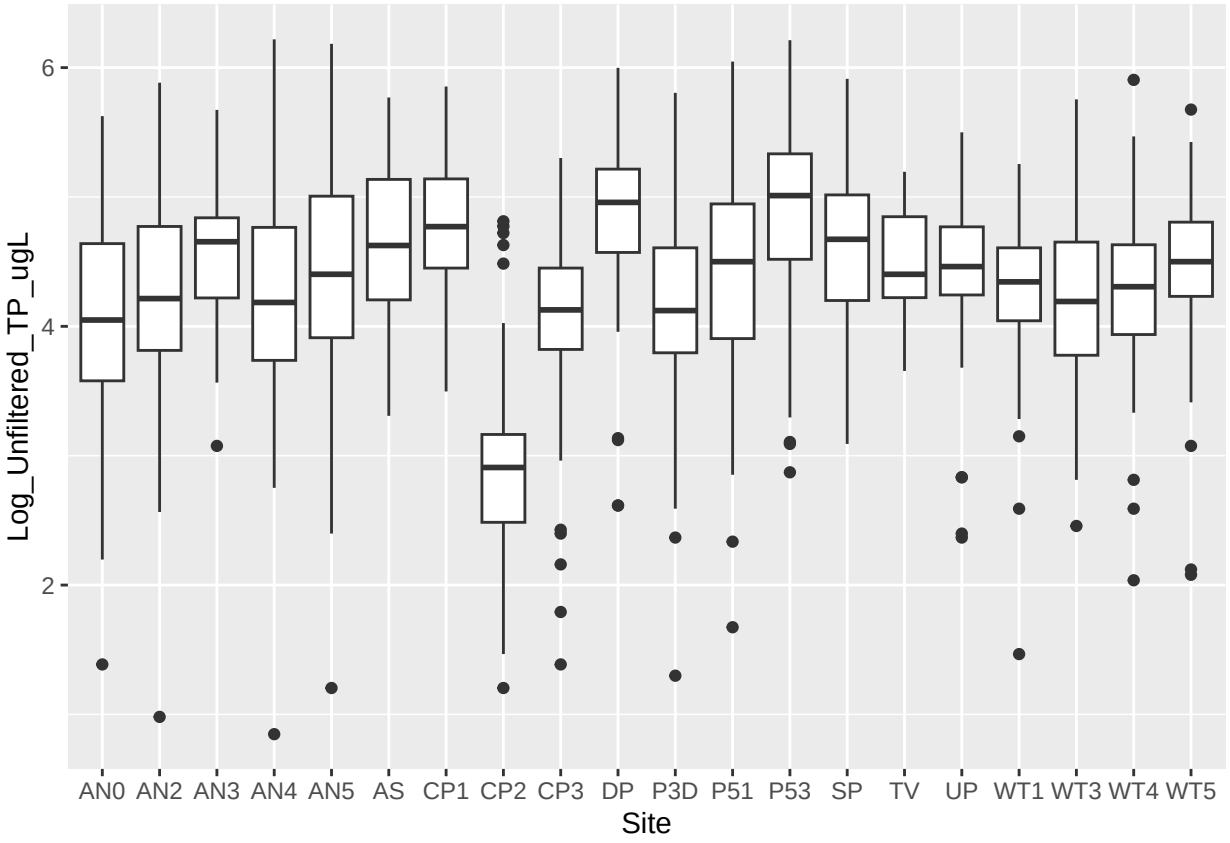

```
ggplot(log_wetland, aes(x = Site, y = pH)) +
geom_boxplot()
```

```
## Warning: Removed 16 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



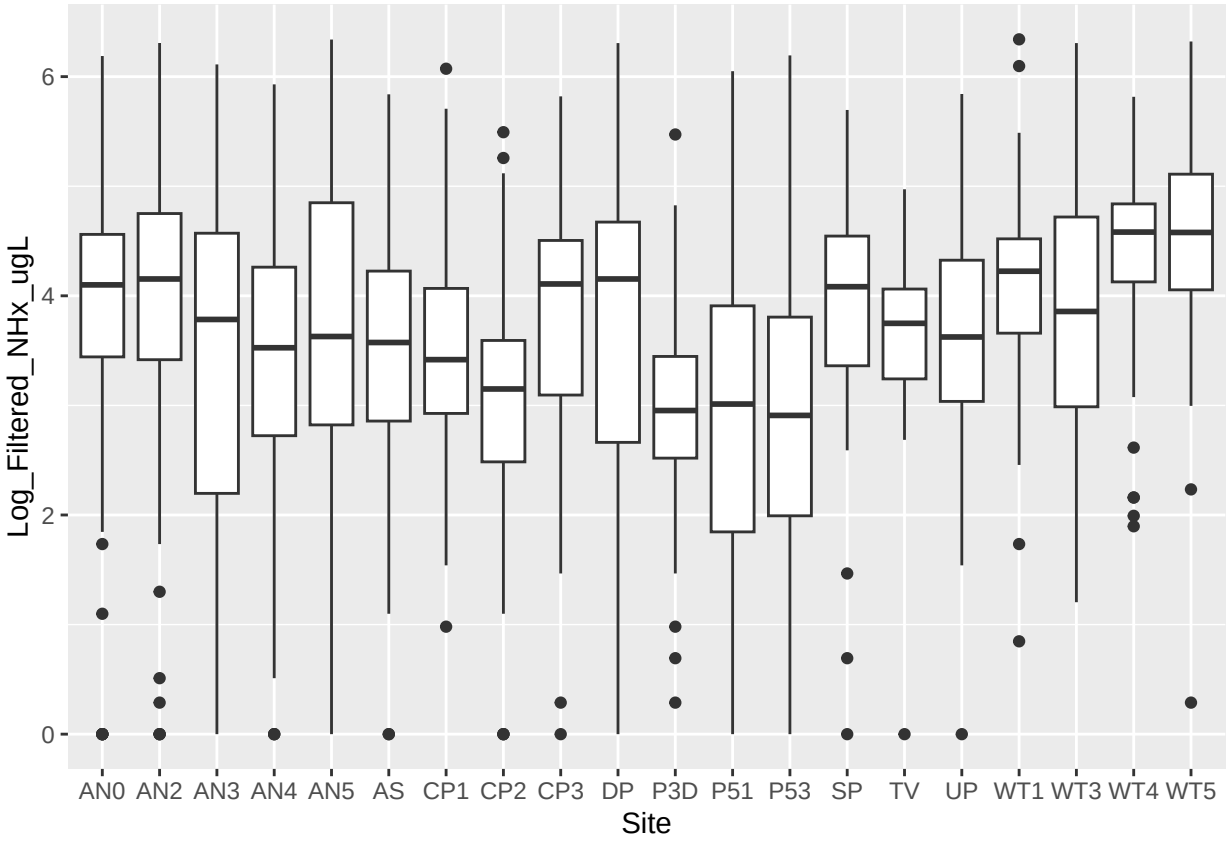

```
ggplot(log_wetland, aes(x = Site, y = Log_Unfiltered_TP_ugL)) +
geom_boxplot()
```

```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



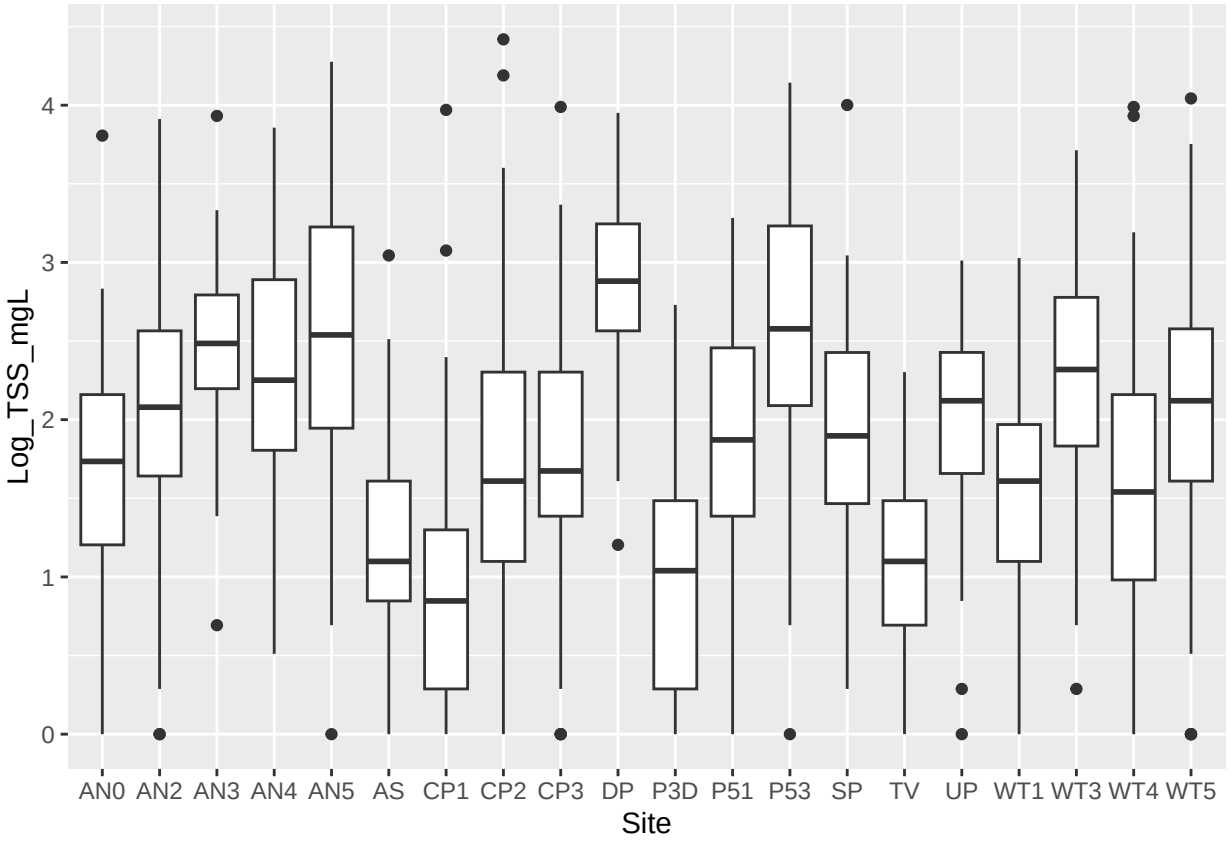

```
ggplot(log_wetland, aes(x = Site, y = Log_Filtered_NHx_ugL)) +
geom_boxplot()
```

```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
ggplot(log_wetland, aes(x = Site, y = Log_TSS_mgL)) +
geom_boxplot()
```

```
## Warning: Removed 15 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Check for correlation between variables

```
#SpCond vs:
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$Log_TDS_mgl)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$Log_TDS_mgl
## t = 13.405, df = 1614, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2719393 0.3597157
## sample estimates:
##       cor
## 0.3165049
```

```
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$Log_Sal_ppt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$Log_Sal_ppt
```

```
## t = 84.681, df = 1615, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8940542 0.9120104
## sample estimates:
##       cor
## 0.9034277
```

```
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$DO_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$DO_mgL
## t = 12.706, df = 1607, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2570739 0.3459077
## sample estimates:
##       cor
## 0.3021467
```

```
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$pH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$pH
## t = 6.0356, df = 1599, p-value = 1.965e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1009941 0.1967977
## sample estimates:
##       cor
## 0.1492461
```

```
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$Log_Unfiltered_TP_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$Log_Unfiltered_TP_ugL
## t = -13.457, df = 1608, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3613849 -0.2735467
## sample estimates:
##        cor
## -0.3181484
```

```
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$Log_TSS_mgL
## t = -16.346, df = 1600, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4194939 -0.3355318
## sample estimates:
##        cor
## -0.3782906
```

```
#TDS vs:
cor.test(log_wetland$Log_TDS_mgl, log_wetland$Log_Sal_ppt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$Log_Sal_ppt
## t = 13.18, df = 1614, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2670273 0.3550956
## sample estimates:
##       cor
## 0.3117308
```

```
cor.test(log_wetland$Log_TDS_mgl, log_wetland$DO_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$DO_mgL
## t = 5.6897, df = 1606, p-value = 1.51e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09231743 0.18815762
## sample estimates:
##       cor
## 0.1405668
```

```
cor.test(log_wetland$Log_TDS_mgl, log_wetland$pH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$pH
## t = 2.8514, df = 1598, p-value = 0.004409
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.02221978 0.11973645
## sample estimates:
##        cor
## 0.07114812
```

```
cor.test(log_wetland$Log_TDS_mgl, log_wetland$Log_Unfiltered_TP_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$Log_Unfiltered_TP_ugL
## t = 1.9267, df = 1607, p-value = 0.05419
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0008626319  0.0966496975
## sample estimates:
##        cor
## 0.04800792
```

```
cor.test(log_wetland$Log_TDS_mgl, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$Log_TSS_mgL
## t = -6.0674, df = 1599, p-value = 1.62e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1975541 -0.1017729
## sample estimates:
##        cor
## -0.1500155
```

```
#Sal vs
cor.test(log_wetland$Log_Sal_ppt, log_wetland$DO_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Sal_ppt and log_wetland$DO_mgL
## t = 11.111, df = 1607, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2211275 0.3119067
## sample estimates:
##       cor
## 0.2671096
```

```
cor.test(log_wetland$Log_Sal_ppt, log_wetland$pH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Sal_ppt and log_wetland$pH
## t = 6.0998, df = 1599, p-value = 1.33e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1025646 0.1983228
## sample estimates:
##       cor
## 0.1507974
```

```
cor.test(log_wetland$Log_Sal_ppt, log_wetland$Log_Unfiltered_TP_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Sal_ppt and log_wetland$Log_Unfiltered_TP_ugL
## t = -12.333, df = 1608, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3379744 -0.2486932
## sample estimates:
##        cor
## -0.2939749
```

```
cor.test(log_wetland$Log_Sal_ppt, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Sal_ppt and log_wetland$Log_TSS_mgL
## t = -14.28, df = 1600, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.378944 -0.292042
## sample estimates:
##        cor
## -0.3362084
```

```
#DO vs
cor.test(log_wetland$DO_mgL, log_wetland$pH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$DO_mgL and log_wetland$pH
## t = 3.4594, df = 1592, p-value = 0.0005557
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03743828 0.13490335
## sample estimates:
```

```
##        cor
## 0.08637749
```

```
cor.test(log_wetland$DO_mgL, log_wetland$Log_Unfiltered_TP_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$DO_mgL and log_wetland$Log_Unfiltered_TP_ugL
## t = -13.143, df = 1601, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3556007 -0.2671967
## sample estimates:
##        cor
## -0.3120741
```

```
cor.test(log_wetland$DO_mgL, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$DO_mgL and log_wetland$Log_TSS_mgL
## t = -12.225, df = 1593, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3371099 -0.2473462
## sample estimates:
##        cor
## -0.2928732
```

```
#pH vs
cor.test(log_wetland$pH, log_wetland$Log_Unfiltered_TP_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$pH and log_wetland$Log_Unfiltered_TP_ugL
## t = -1.0302, df = 1593, p-value = 0.3031
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07479121  0.02330876
## sample estimates:
##         cor
## -0.02580335
```

```
cor.test(log_wetland$pH, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
```

```
## data:  log_wetland$pH and log_wetland$Log_TSS_mgL
## t = -5.4997, df = 1585, p-value = 4.429e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.18480267 -0.08822875
## sample estimates:
##        cor
## -0.1368408
```

```
#TP vs
cor.test(log_wetland$Log_Unfiltered_TP_ugL, log_wetland$Log_TSS_mgL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Unfiltered_TP_ugL and log_wetland$Log_TSS_mgL
## t = 13.204, df = 1600, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2686032 0.3569504
## sample estimates:
##       cor
## 0.3134549
```

```
#Predictor and Outcome Variable correlations
cor.test(log_wetland$Log_SpCond_mScm, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_SpCond_mScm and log_wetland$Log_Filtered_NHx_ugL
## t = -0.95093, df = 1593, p-value = 0.3418
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07281603  0.02529358
## sample estimates:
##         cor
## -0.02381858
```

```
cor.test(log_wetland$Log_TDS_mgl, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TDS_mgl and log_wetland$Log_Filtered_NHx_ugL
## t = -3.2848, df = 1592, p-value = 0.001043
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.13062002 -0.03308351
## sample estimates:
##         cor
## -0.08204823
```

```r
cor.test(log_wetland$Log_Sal_ppt, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Sal_ppt and log_wetland$Log_Filtered_NHx_ugL
## t = -0.78141, df = 1593, p-value = 0.4347
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06859105  0.02953660
## sample estimates:
##         cor
## -0.01957436
```

```r
cor.test(log_wetland$DO_mgL, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$DO_mgL and log_wetland$Log_Filtered_NHx_ugL
## t = -6.1243, df = 1585, p-value = 1.146e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1997543 -0.1036115
## sample estimates:
##        cor
## -0.1520425
```

```r
cor.test(log_wetland$pH, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$pH and log_wetland$Log_Filtered_NHx_ugL
## t = -1.508, df = 1577, p-value = 0.1318
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08711329  0.01140640
## sample estimates:
##         cor
## -0.03794565
```

```r
cor.test(log_wetland$Log_Unfiltered_TP_ugL, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_Unfiltered_TP_ugL and log_wetland$Log_Filtered_NHx_ugL
## t = 7.1518, df = 1590, p-value = 1.302e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1285240 0.2237266
## sample estimates:
##       cor
## 0.1765381
```

```
cor.test(log_wetland$Log_TSS_mgL, log_wetland$Log_Filtered_NHx_ugL)
```

```
##
##  Pearson's product-moment correlation
##
## data:  log_wetland$Log_TSS_mgL and log_wetland$Log_Filtered_NHx_ugL
## t = 3.2923, df = 1583, p-value = 0.001016
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03336349 0.13116986
## sample estimates:
##        cor
## 0.08246524
```

SpCond and Sal are too correlated to use together - makes sense. Also all the predictor variables have very low correlations with the outcome variable :(

Step 3 - Fit regular regression model

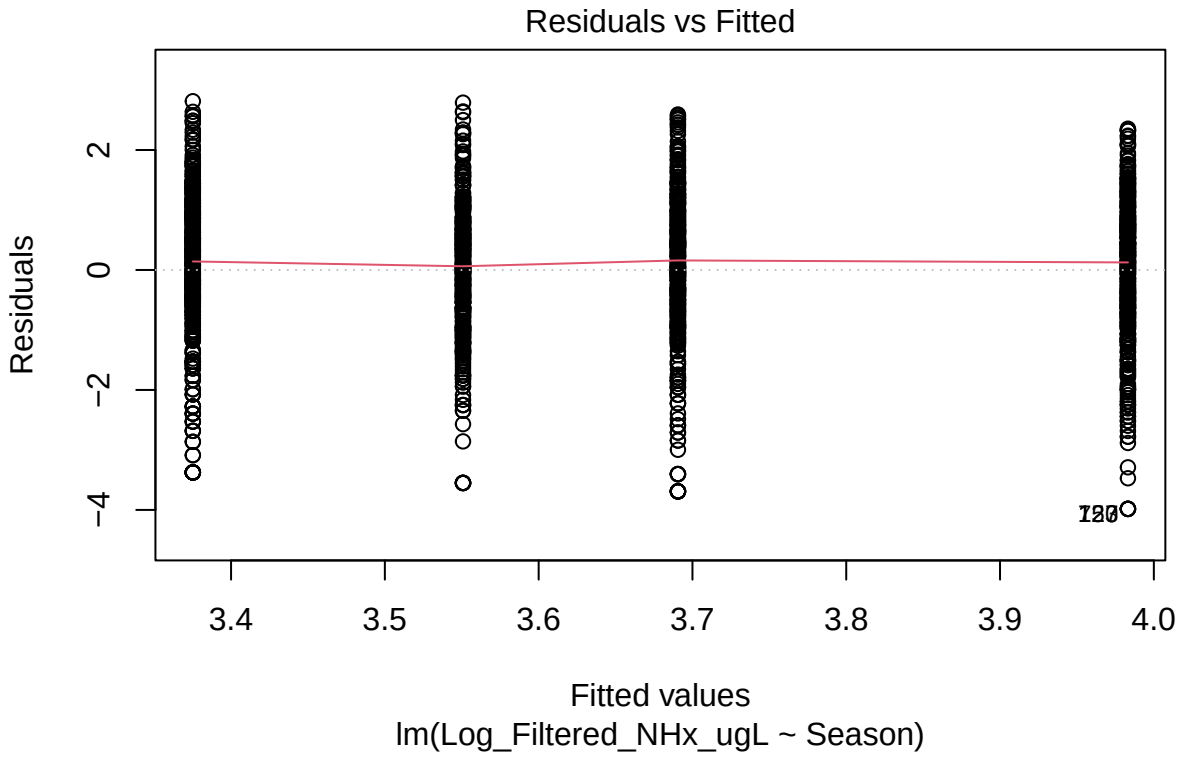
I think I'm going to try three models.

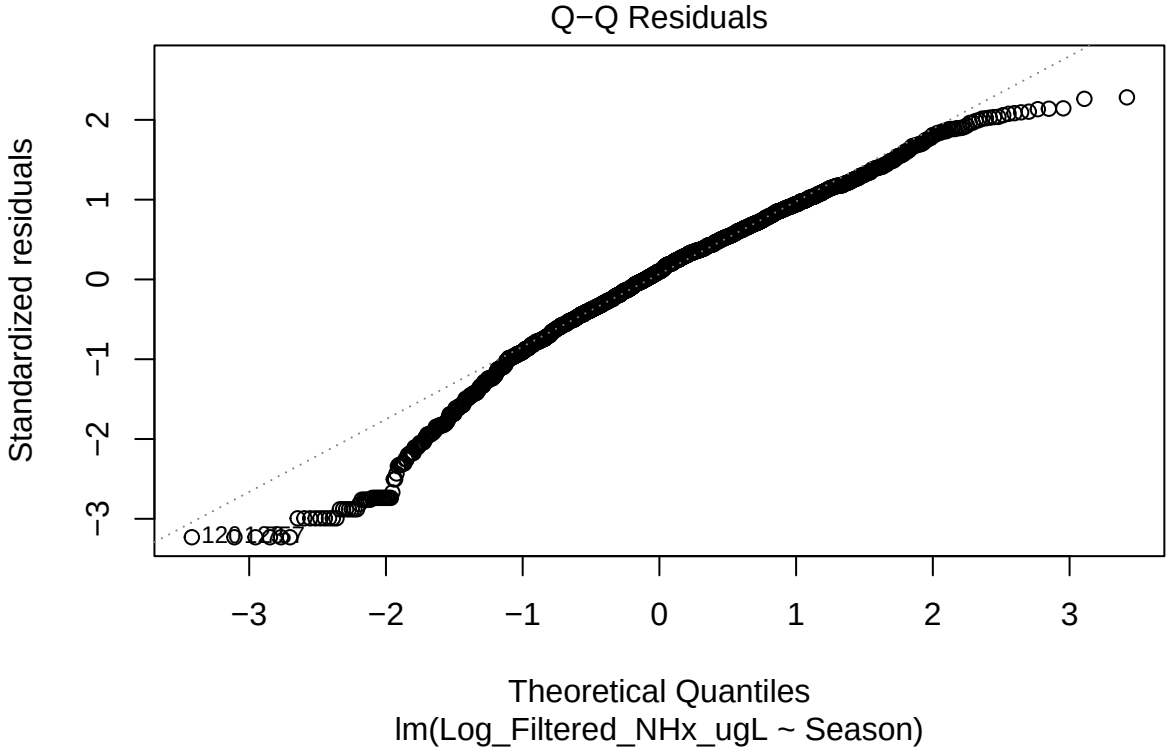
```
#first convert Season to a factor
log_wetland$Season <- factor(log_wetland$Season,
                             levels = c('Winter', 'Spring', 'Summer', 'Fall'))
# fit regression models
#start with just season
mod1 <- lm(Log_Filtered_NHx_ugL ~ Season,
           data = log_wetland)

#next add predictor variables with highest correlation to NHx
mod2 <- lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL,
           data = log_wetland)

#keep adding based on correlation (TSS was not significant even though it had the same value of correla
mod3 <- lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL +
             Log_SpCond_mScm,
           data = log_wetland)

#examine model outputs and residuals
summary(mod1)
```

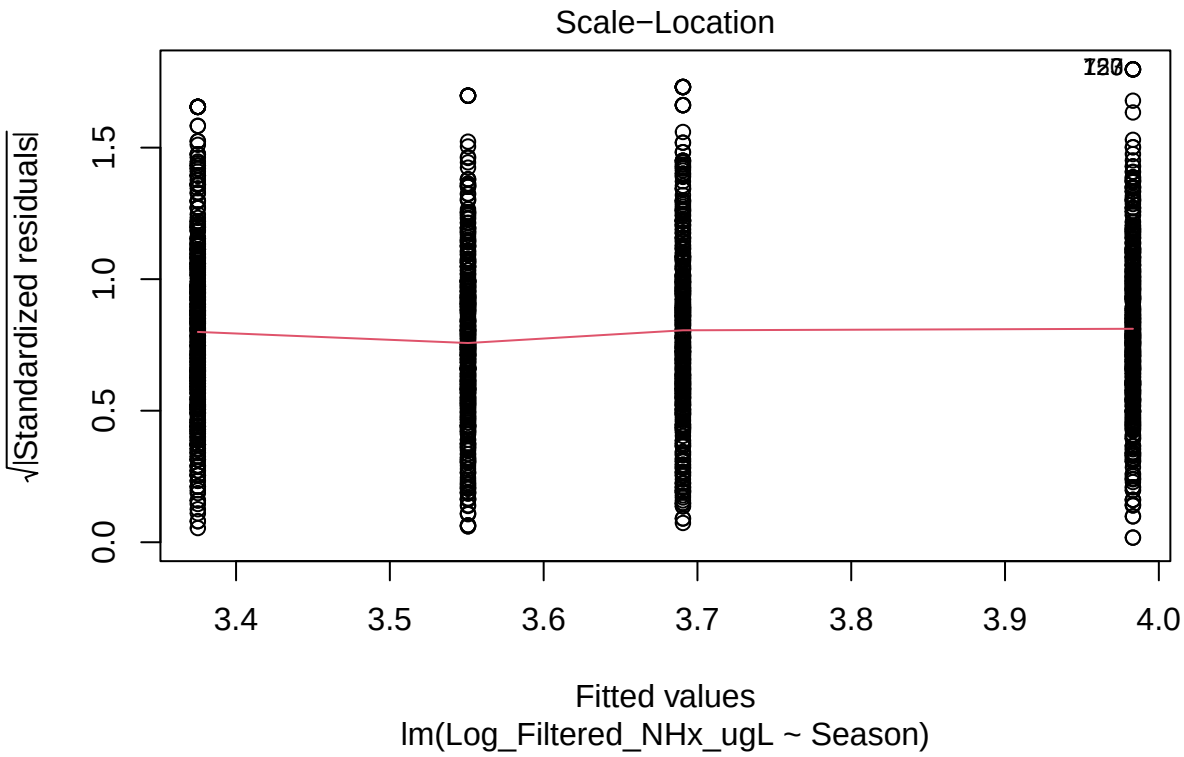
```
##
## Call:
## lm(formula = Log_Filtered_NHx_ugL ~ Season, data = log_wetland)
##
## Residuals:
```

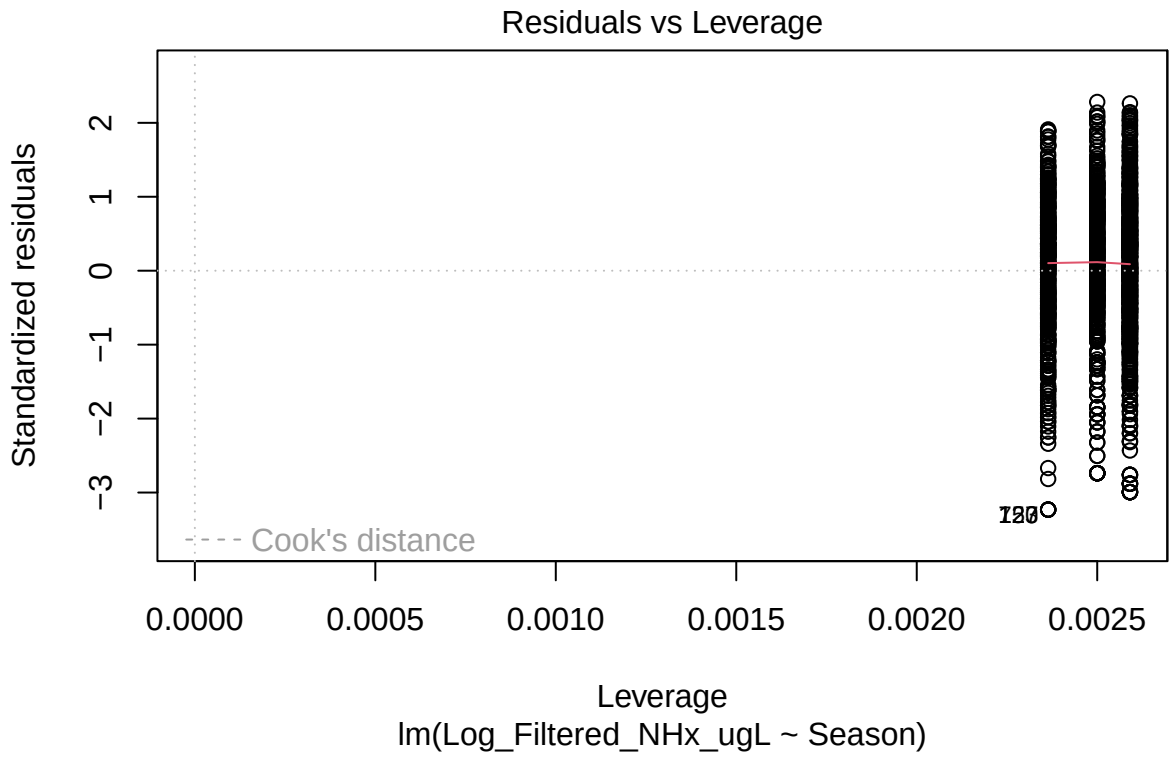
```
##     Min      1Q  Median      3Q     Max
## -3.9832 -0.6727  0.1214  0.8407  2.8131
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.55082    0.06282  56.524  < 2e-16 ***
## SeasonSpring  0.13975    0.08884   1.573   0.1159
## SeasonSummer  0.43236    0.08688   4.977 7.17e-07 ***
## SeasonFall   -0.17567    0.08806  -1.995   0.0462 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.234 on 1591 degrees of freedom
##   (23 observations deleted due to missingness)
## Multiple R-squared:  0.03256,    Adjusted R-squared:  0.03073
## F-statistic: 17.85 on 3 and 1591 DF,  p-value: 2.148e-11
```

```
plot(mod1)
```




Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Log_Filtered_NHx_ugL ~ Season)

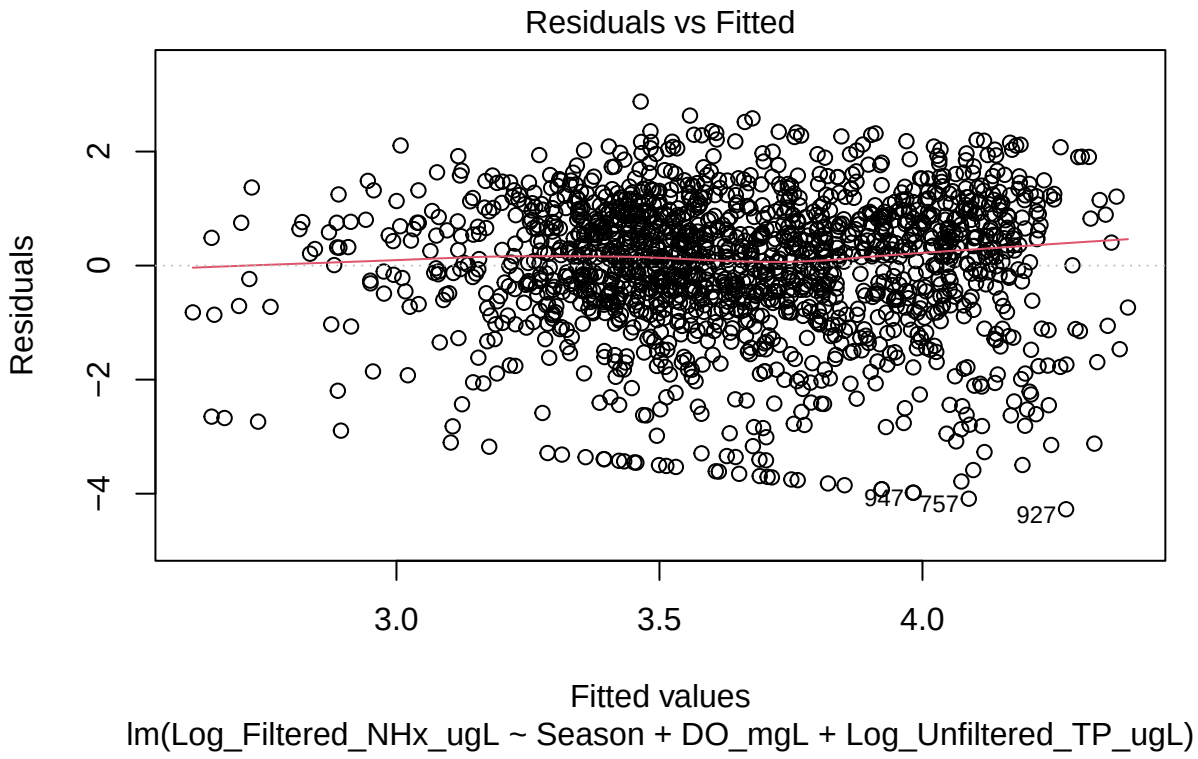

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
3.4
3.5
3.6
3.7
3.8
3.9
4.0
Fitted values
lm(Log_Filtered_NHx_ugL ~ Season)

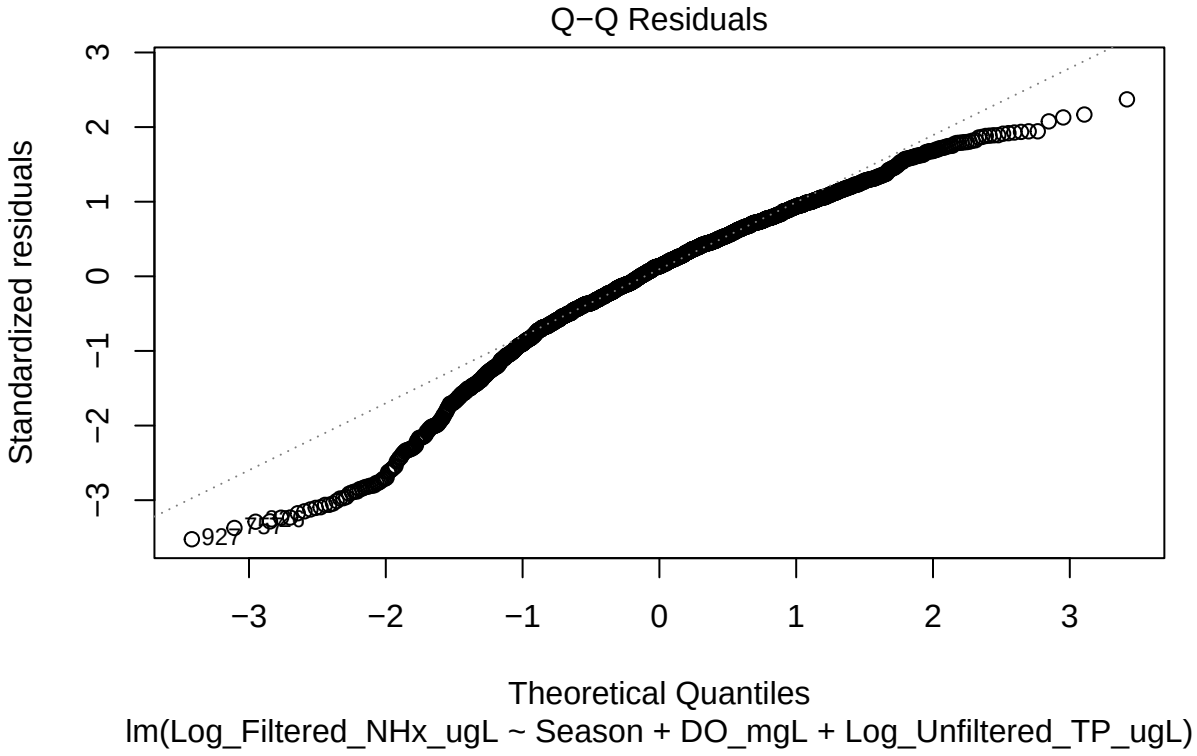


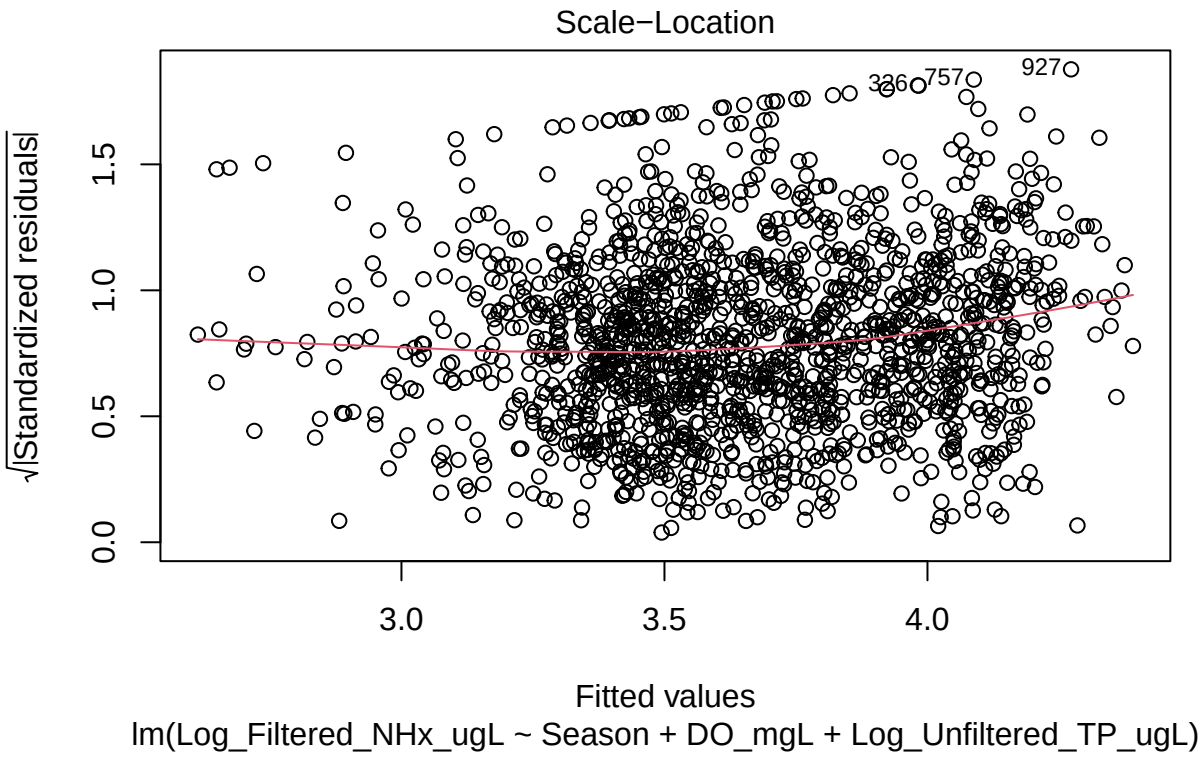

```
summary(mod2)
```

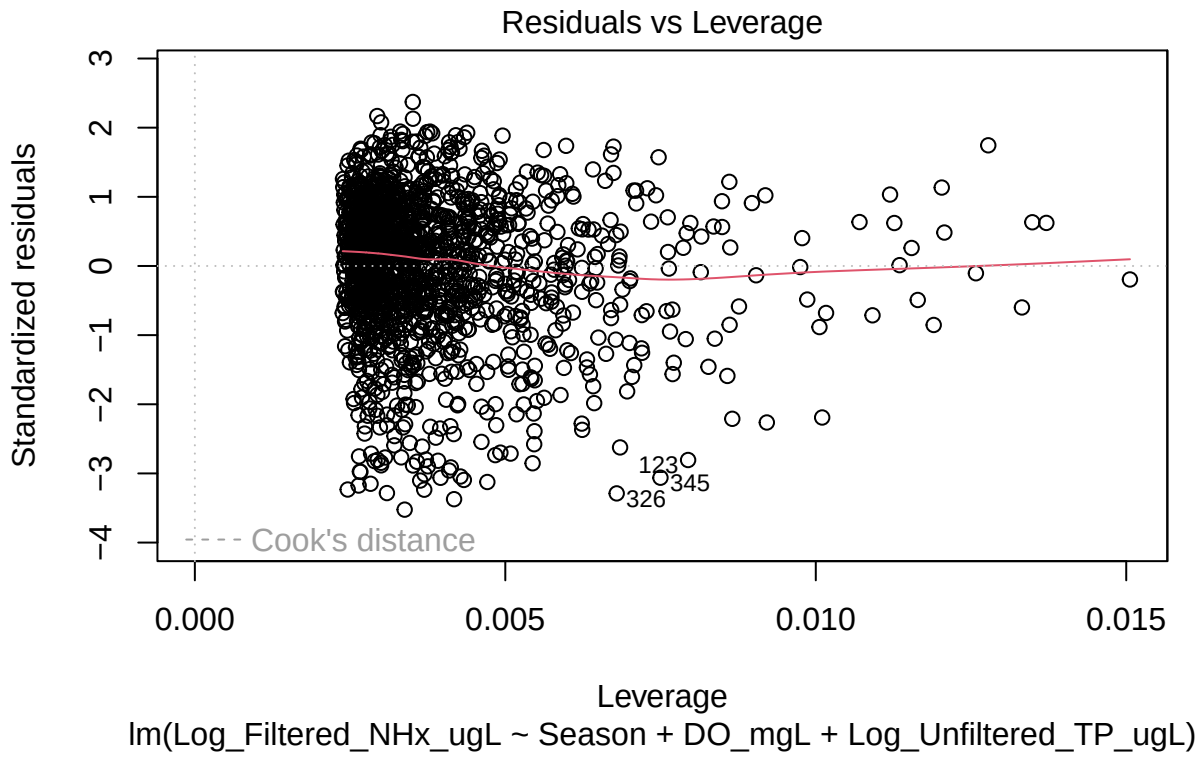
```
##
## Call:
## lm(formula = Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL,
##     data = log_wetland)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2730 -0.6190  0.1648  0.8504  2.8761
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.977994   0.219184  13.587  < 2e-16 ***
## SeasonSpring           0.066068   0.090387   0.731 0.464919
## SeasonSummer           0.186015   0.096687   1.924 0.054547 .
## SeasonFall            -0.332438   0.093362  -3.561 0.000381 ***
## DO_mgL                -0.034799   0.009272  -3.753 0.000181 ***
## Log_Unfiltered_TP_ugL  0.214660   0.041166   5.214 2.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.215 on 1578 degrees of freedom
##   (34 observations deleted due to missingness)
## Multiple R-squared:  0.06617,    Adjusted R-squared:  0.06321
## F-statistic: 22.36 on 5 and 1578 DF,  p-value: < 2.2e-16
```

```
plot(mod2)
```


Residuals vs Fitted
Residuals
2
0
−2
−4
947
757
927
3.0
3.5
4.0
Fitted values
lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL)


Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
927
Theoretical Quantiles
lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
3.0
3.5
4.0
Fitted values
lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL)

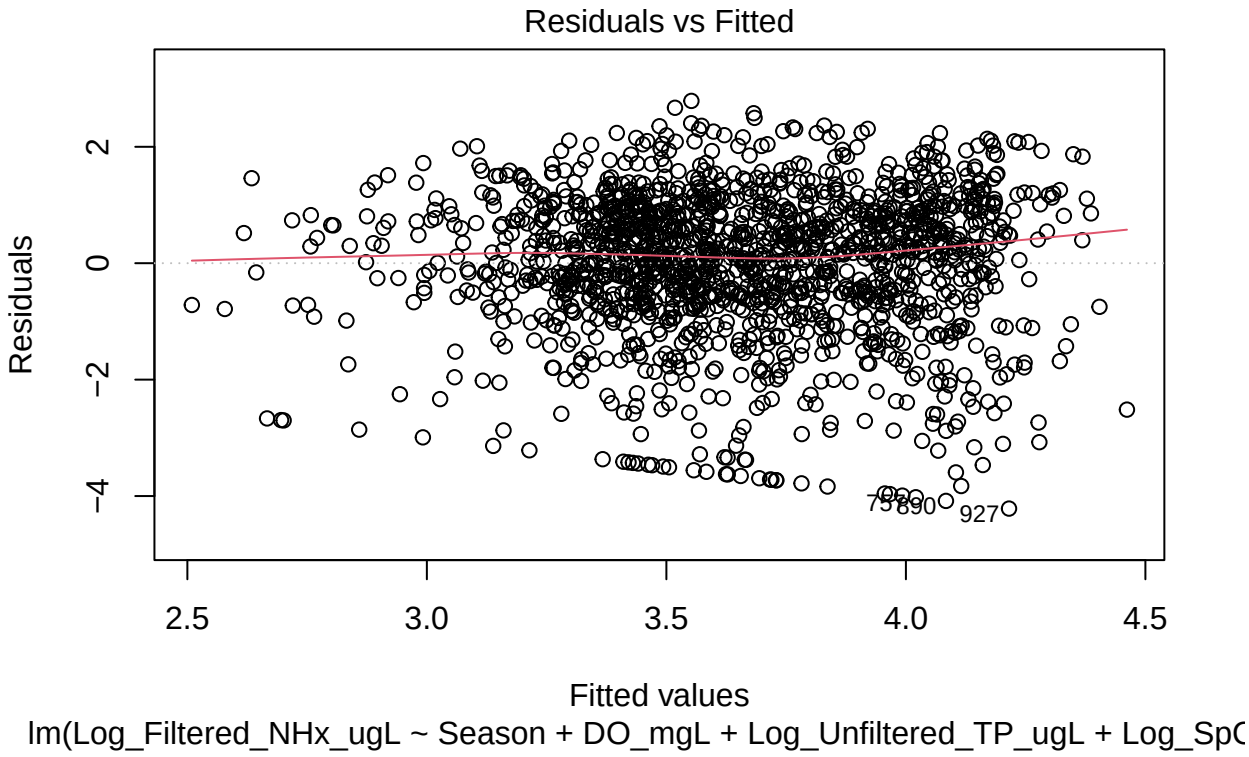


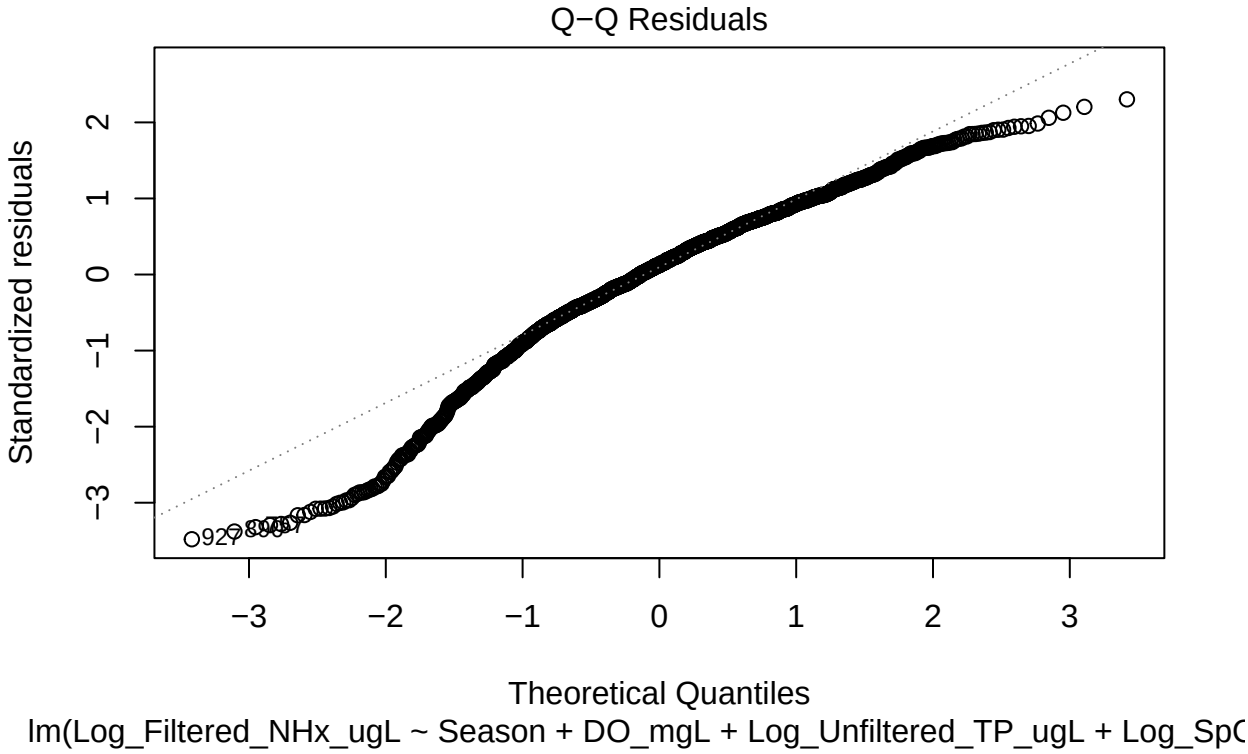

```
summary(mod3)
```

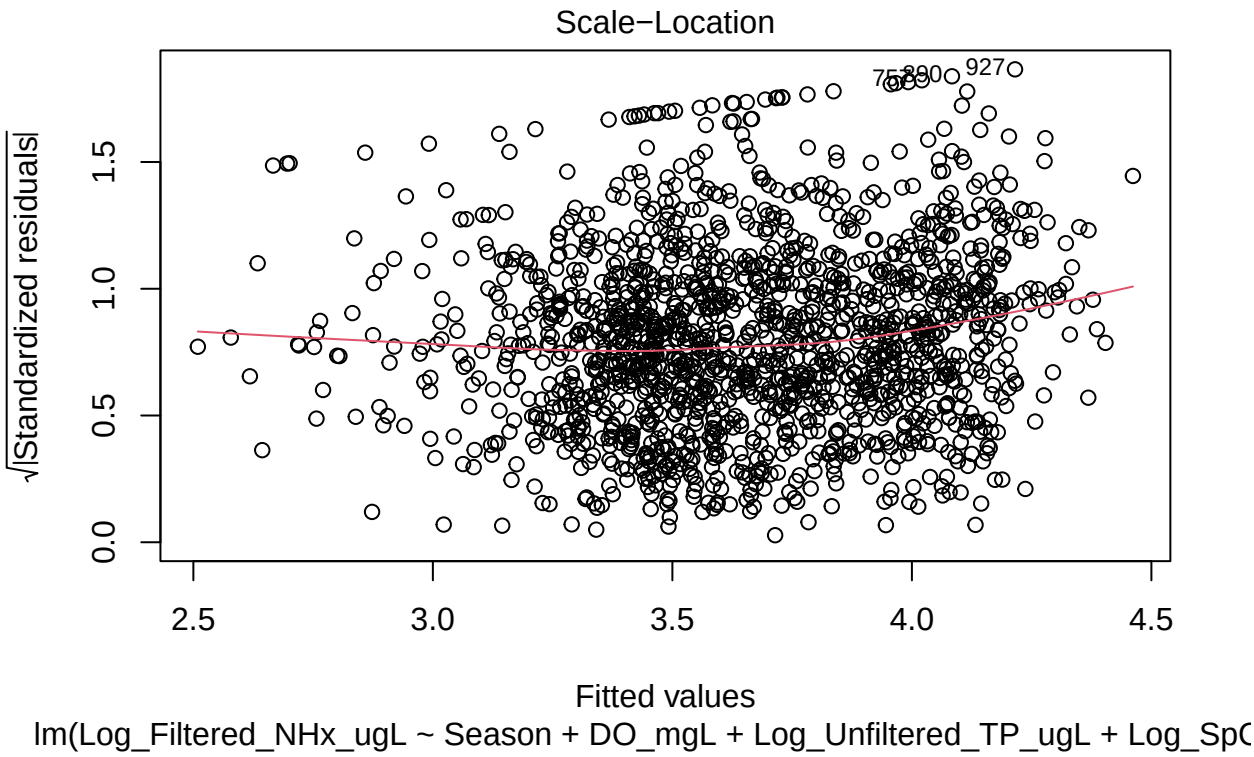
```
##
## Call:
## lm(formula = Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL +
##     Log_SpCond_mScm, data = log_wetland)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2153 -0.6104  0.1516  0.8466  2.7881
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.774938   0.242462  11.445  < 2e-16 ***
## SeasonSpring           0.071709   0.090353   0.794 0.427516
## SeasonSummer           0.199213   0.096837   2.057 0.039832 *
## SeasonFall            -0.310418   0.093959  -3.304 0.000975 ***
## DO_mgL                -0.037722   0.009384  -4.020 6.10e-05 ***
## Log_Unfiltered_TP_ugL  0.235607   0.042508   5.543 3.49e-08 ***
## Log_SpCond_mScm        0.531945   0.272645   1.951 0.051227 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.214 on 1577 degrees of freedom
##   (34 observations deleted due to missingness)
```

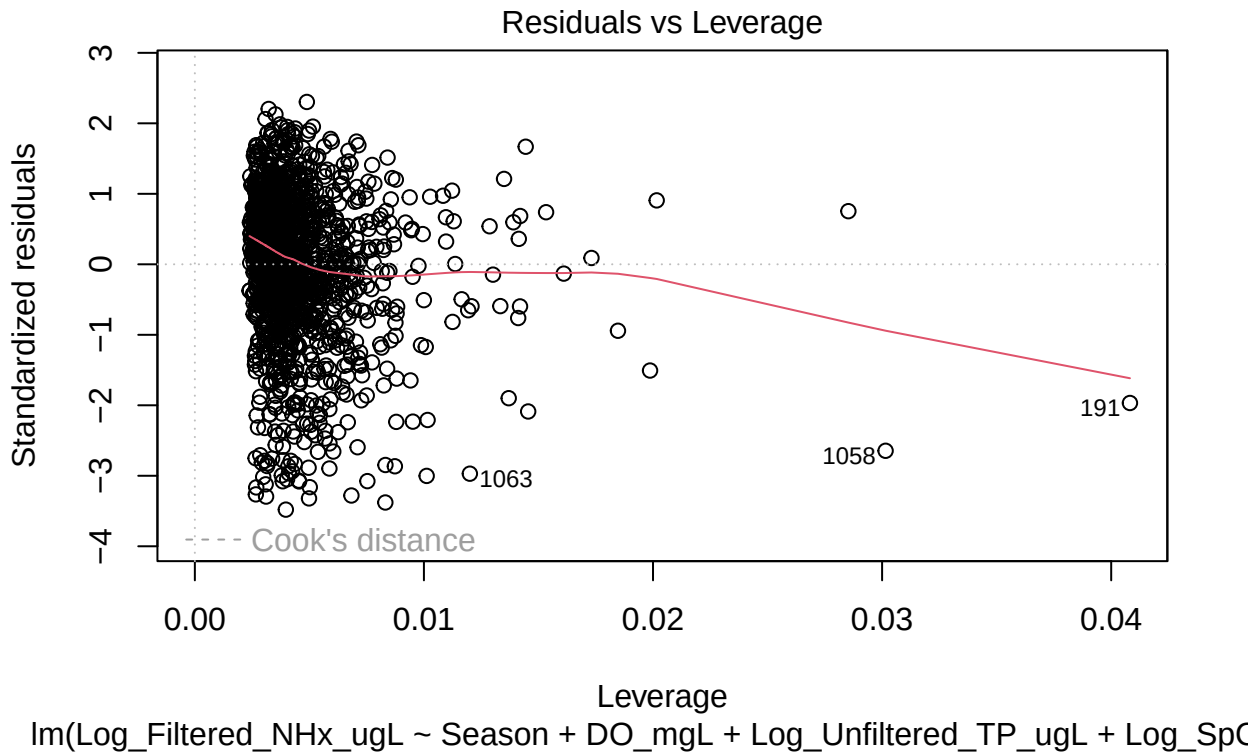
```
## Multiple R-squared:  0.06841,    Adjusted R-squared:  0.06487
## F-statistic:  19.3 on 6 and 1577 DF,  p-value: < 2.2e-16
```

```
plot(mod3)
```




Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL + Log_SpC


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.5
3.0
3.5
4.0
4.5
927
Fitted values
lm(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL + Log_SpC



* pH and TSS were not significant when I used them as predictor variables. I tried using Log_TDS but its distribution was too weird.

Step 4 - Fit Mixed Effects Model and Examine Model Results

use gls model first

```
#remove NA's
log_wetland <- na.omit(log_wetland)

#first refit models using generalized least squares
GLS1 <- gls(Log_Filtered_NHx_ugL ~ Season,
            data = log_wetland)

GLS2 <- gls(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL,
            data = log_wetland)

GLS3 <- gls(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL +
             Log_SpCond_mScm,
            data = log_wetland)
```

##now fit to mixed effects model and compare to GLS

```r
#fit data to mixed effects models
MEM1 <- lme(Log_Filtered_NHx_ugL ~ Season,
            random = ~ 1|Site, data = log_wetland)
MEM2 <- lme(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL,
            random = ~ 1|Site, data = log_wetland)
MEM3 <- lme(Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL +
             Log_SpCond_mScm,
            random = ~ 1|Site, data = log_wetland)

#Compare AIC values between gls and lme models
AIC(GLS1, MEM1)
```

```
##      df      AIC
## GLS1  5 4863.716
## MEM1  6 4686.223
```

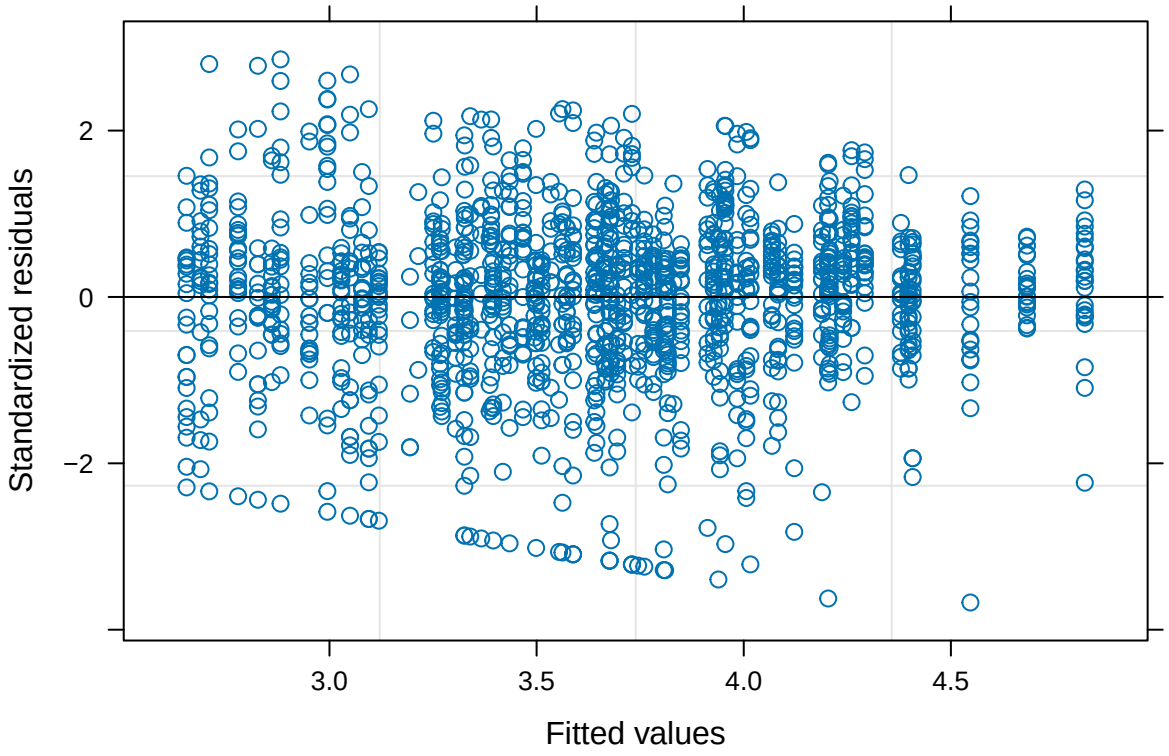
```r
AIC(GLS2, MEM2)
```

```
##      df      AIC
## GLS2  7 4829.508
## MEM2  8 4636.834
```

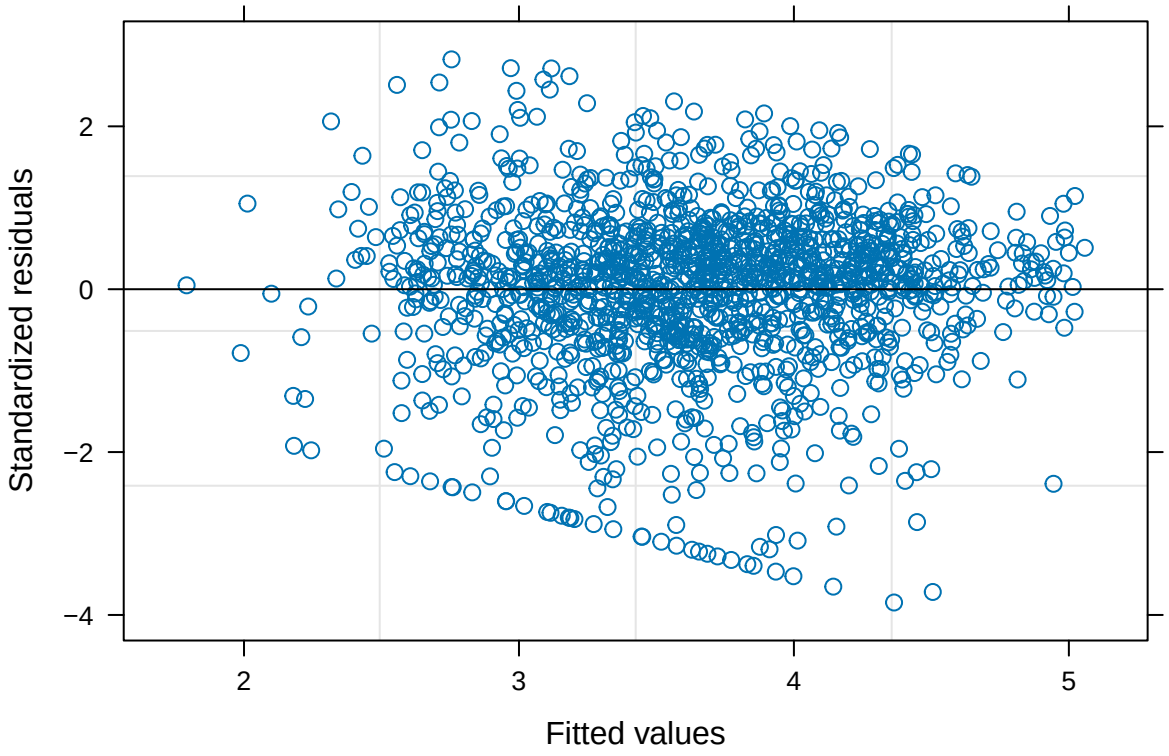
```r
AIC(GLS3, MEM3)
```

```
##      df      AIC
## GLS3  8 4827.039
## MEM3  9 4637.691
```

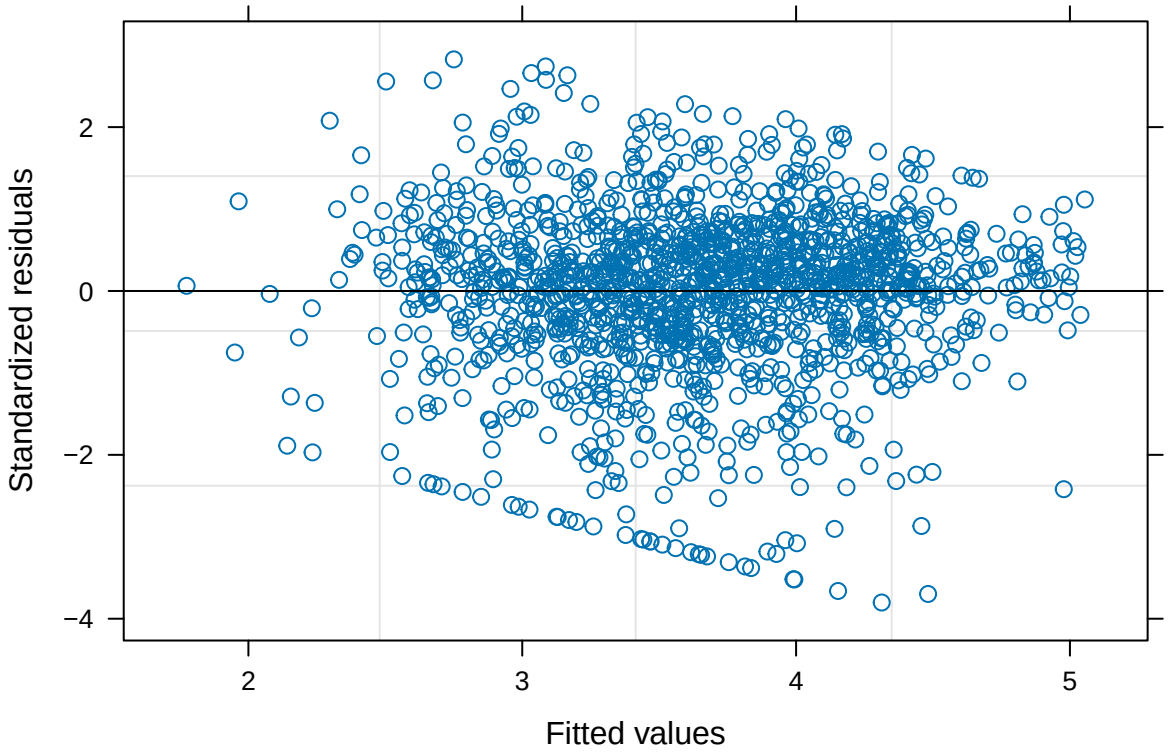
```r
#Examine models residual plots
plot(MEM1)
```



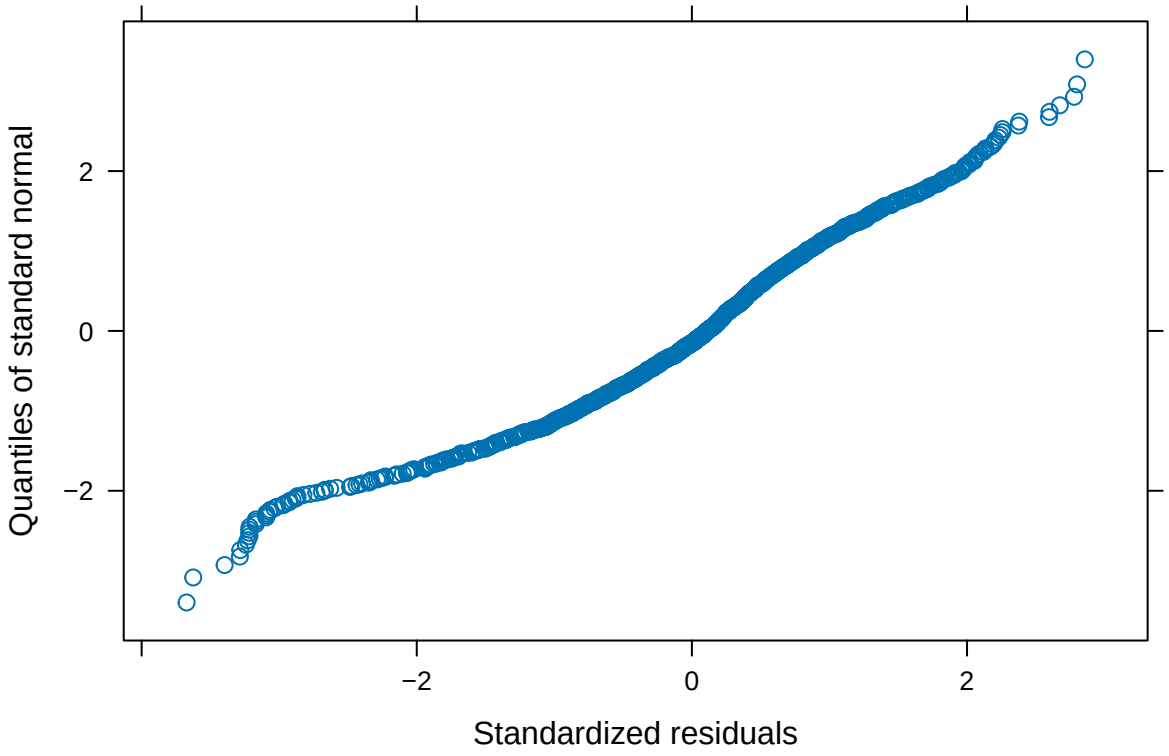

```
plot(MEM2)
```



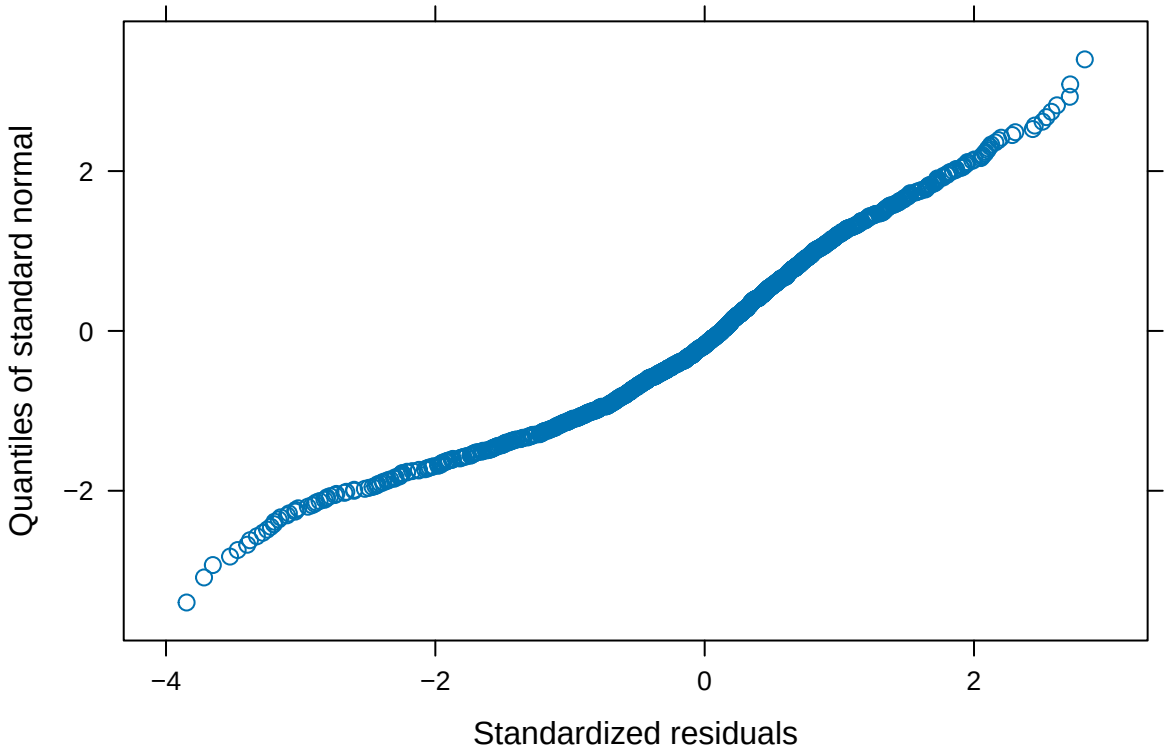

```
plot(MEM3)
```



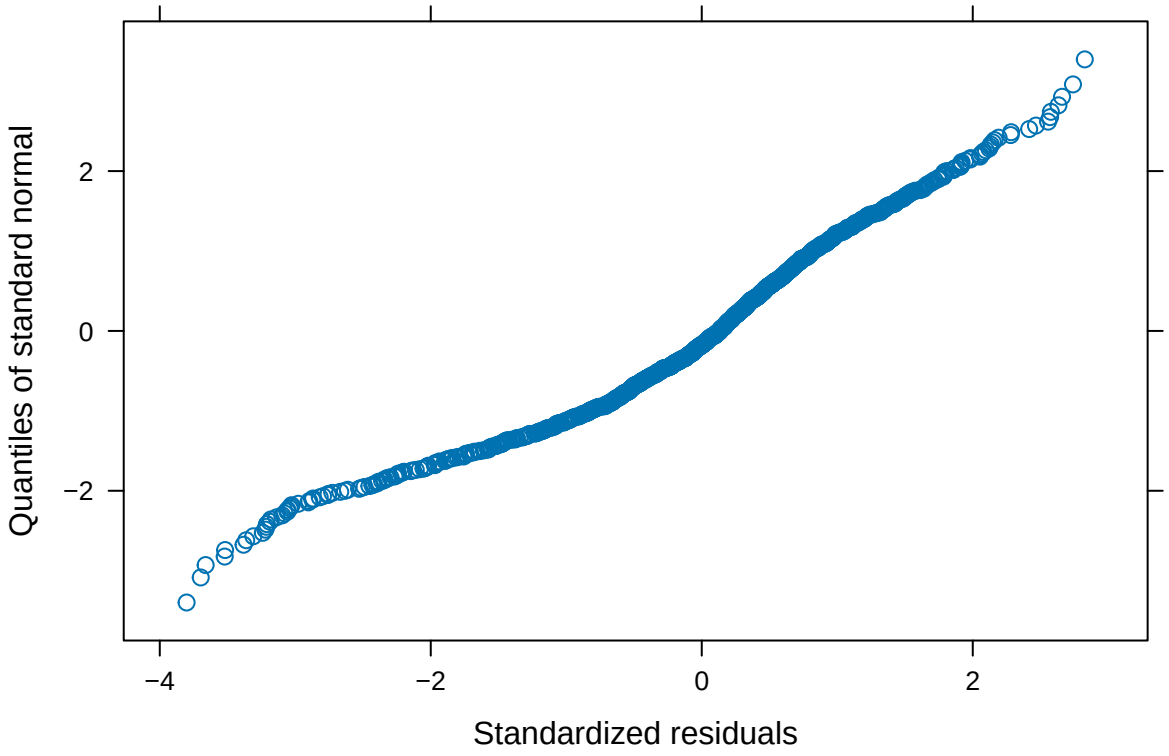

```
qqnorm(MEM1)
```


Quantiles of standard normal
2
0
−2
−2
0
2
Standardized residuals

```
qqnorm(MEM2)
```


Quantiles of standard normal
2
0
−2
−4
−2
0
2
Standardized residuals

```
qqnorm(MEM3)
```




```
#Examine results
summary(MEM1)
```

```
## Linear mixed-effects model fit by REML
##   Data: log_wetland
##        AIC      BIC    logLik
##   4686.223 4717.977 -2337.112
##
## Random effects:
##  Formula: ~1 | Site
##         (Intercept) Residual
## StdDev:   0.4565253 1.159622
##
## Fixed effects:  Log_Filtered_NHx_ugL ~ Season
##                  Value  Std.Error   DF   t-value p-value
## (Intercept)   3.512181 0.11921993 1450 29.459683  0.0000
## SeasonSpring  0.167685 0.08567299 1450  1.957264  0.0505
## SeasonSummer  0.443760 0.08551556 1450  5.189237  0.0000
## SeasonFall   -0.172126 0.08553977 1450 -2.012228  0.0444
##  Correlation:
##              (Intr) SsnSpr SsnSmm
## SeasonSpring -0.358
## SeasonSummer -0.356  0.498
## SeasonFall   -0.356  0.499  0.502
##
## Standardized Within-Group Residuals:
```

```
##        Min         Q1        Med         Q3        Max
## -3.6731253 -0.4947123  0.1094659  0.5655409  2.8559787
##
## Number of Observations: 1473
## Number of Groups: 20
```

```
summary(MEM2)
```

```
## Linear mixed-effects model fit by REML
##   Data: log_wetland
##        AIC      BIC    logLik
##   4636.834 4679.161 -2310.417
##
## Random effects:
##  Formula: ~1 | Site
##         (Intercept) Residual
## StdDev:   0.4664368 1.134435
##
## Fixed effects:  Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL
##                             Value  Std.Error   DF   t-value p-value
## (Intercept)            3.0365133 0.27925345 1448 10.873682  0.0000
## SeasonSpring           0.0519798 0.08744536 1448  0.594426  0.5523
## SeasonSummer           0.0941485 0.09677645 1448  0.972845  0.3308
## SeasonFall            -0.4034761 0.09183333 1448 -4.393570  0.0000
## DO_mgL                -0.0501912 0.01011708 1448 -4.961036  0.0000
## Log_Unfiltered_TP_ugL  0.2263654 0.04953527 1448  4.569783  0.0000
##  Correlation:
##                       (Intr) SsnSpr SsnSmm SsnFll DO_mgL
## SeasonSpring          -0.315
## SeasonSummer          -0.245  0.541
## SeasonFall            -0.310  0.549  0.597
## DO_mgL                -0.569  0.285  0.458  0.404
## Log_Unfiltered_TP_ugL -0.844  0.103 -0.051  0.053  0.316
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -3.8470673 -0.4849579  0.1273222  0.5940158  2.8224680
##
## Number of Observations: 1473
## Number of Groups: 20
```

```
summary(MEM3)
```

```
## Linear mixed-effects model fit by REML
##   Data: log_wetland
##        AIC      BIC    logLik
##   4637.691 4685.304 -2309.846
##
## Random effects:
##  Formula: ~1 | Site
##         (Intercept) Residual
## StdDev:   0.4635083 1.134327
##
```

```
## Fixed effects:  Log_Filtered_NHx_ugL ~ Season + DO_mgL + Log_Unfiltered_TP_ugL +      Log_SpCond_mScm
##                               Value Std.Error   DF   t-value p-value
## (Intercept)            2.8740528 0.3089498 1447  9.302653  0.0000
## SeasonSpring           0.0626061 0.0878640 1447  0.712534  0.4762
## SeasonSummer           0.1181025 0.0987091 1447  1.196470  0.2317
## SeasonFall            -0.3767081 0.0943810 1447 -3.991356  0.0001
## DO_mgL                -0.0503194 0.0101157 1447 -4.974374  0.0000
## Log_Unfiltered_TP_ugL  0.2391369 0.0506120 1447  4.724905  0.0000
## Log_SpCond_mScm        0.4091983 0.3346696 1447  1.222693  0.2216
##  Correlation:
##                       (Intr) SsnSpr SsnSmm SsnFll DO_mgL L_U_TP
## SeasonSpring          -0.326
## SeasonSummer          -0.302  0.547
## SeasonFall            -0.372  0.555  0.615
## DO_mgL                -0.509  0.282  0.447  0.390
## Log_Unfiltered_TP_ugL -0.835  0.121 -0.008  0.098  0.307
## Log_SpCond_mScm       -0.430  0.099  0.198  0.231 -0.012  0.207
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -3.8019796 -0.4808547  0.1330462  0.5889671  2.8267807
##
## Number of Observations: 1473
## Number of Groups: 20
```

The mixed effect models have lower AIC values than the gls linear regression models For the mixed effect models' AIC values: MEM2(4636) and 3(4637) are lower than MEM1(4686); MEM2 has the lowest value MEM1 var = 0.2, MEM2 var = 0.22, MEM3 var = 0.21

I tried adding a nested model with Year/Site but it didn't improve model fit

Choosing model 2 as the best

Step 5 - Communicate methods and results

methods

To examine the effects of water quality parameters on ammonia/ammonium levels in NC wetlands we fit a multi-level model with filtered NHx (measured in micro grams/L) as the response variable and season, dissolved oxygen, and unfiltered total phosphorous as fixed effects. We also included site as a random effect to account for repeated sampling.

results

Our multi-level model results indicate that NHx concentrations significantly decreased with increasing dissolved oxygen levels (B^DO = -0.05, $p < 0.001$) and increased with increasing total phosphorus (B^P = 0.23, $p < 0.001$). There was also a significant increase in NHx concentrations in the spring (B^spring = 0.46, $p < 0.001$), summer (B^summer = 0.50, $p < 0.001$), and winter (0.40, $p < 0.001$) relative to the fall.

Characteristic	Beta	95% CI	p-value
(Intercept)	3.0	2.5, 3.6	<0.001
SeasonSpring	0.05	-0.12, 0.22	0.6
SeasonSummer	0.09	-0.10, 0.28	0.3
SeasonFall	-0.40	-0.58, -0.22	<0.001
DO_mgL	-0.05	-0.07, -0.03	<0.001
Log_Unfiltered_TP_ugL	0.23	0.13, 0.32	<0.001
Site.SD (Intercept)	0.47		
Residual.SD (Observations)	1.1		

Abbreviation: CI = Confidence Interval

visualize results

```
# table results
tbl_regression(MEM2)
```

```
## ! `broom::tidy()` failed to tidy the model.

## v `tidy_parameters()` used instead.

## i Add `tidy_fun = broom.helpers::tidy_parameters` to quiet these messages.

## x Unable to identify the list of variables.
##
## This is usually due to an error calling `stats::model.frame(x)`or `stats::model.matrix(x)`.
## It could be the case if that type of model does not implement these methods.
## Rarely, this error may occur if the model object was created within
## a functional programming framework (e.g. using `lappy()`, `purrr::map()`, etc.).
```

```
#
```