Assignment 2: Acid Rain

Sam Struthers & Carolina Barbosa - adapted from Nick Gubbins

2025-01-29

Acid Rain

While rain is naturally acidic due to the carbonation of water, a polluted atmosphere can increase the acidity of rainfall. Adding acid to a land surface can displace soil cations and change solute chemistry in downstream waterways. Burning fossil fuels was, and is, major source of air pollution. The Clean Air Act Amendments of 1990 were instrumental in tackling polluted air in the US. The amendments capped total emissions of acidic nitrogen oxides and sulfur dioxides, establishing a tradeable, emissions credit pool for power plants. You can read more about the long-term effects of these efforts in this informative page.

Hubbard Brook

The Hubbard Brook Experimental Forest (HBEF) is a long-running ecosystem study located in New Hampshire.The study has several small streams running through forested watersheds. Different treatments have been applied to some of the watersheds, but this unit we will be looking at a reference site, watershed 6. W6 is located in the industrial Northeast of the US and has a record beginning in 1963, making it ideal to observe the effects of acid rain.

Working with files in R

File paths and directories

R needs to know where files are in order to interact with them. The old method, that folks with R experience may know, is using the functions 'getwd()' and 'setwd()'. Those folks will also be happy to know that there is a better way!

Using 'projects' in Rstudio, we can have Rstudio track where we are when the Rmd is opened. This is stored in the '.Rproj' file in your files. **Take a second to check the top right corner of your screen.** There should be a small blue box with an 'R' in it and the name of a project (this will not be there on Rstudio cloud). We can check that R knows where we are with the 'here()' function.

```
#here()
```

Projects make sharing scripts and sheets with complicated file structures much, much smoother.

Reading in files

Now that we confirmed Rstudio knows where it is, we can read in our data files. The files we are using have been downloaded from Macrosheds, which uses a file format called 'feather'. For those of you that have used

R before, you are probably familiar with the function 'read.csv()'. This works basically the same, but for a more efficient file format.

To read in the file, we will supply the 'read_feather()' function the location of the file relative to our project file.

```
p_chem <- read_feather('data/acid_rain/w6_precip_chem.feather')
head(p_chem)
```

```
## # A tibble: 6 x 7
##   datetime            site_code var         val ms_status ms_interp val_err
##   <dttm>              <chr>     <chr>     <dbl>     <dbl>     <dbl>   <dbl>
## 1 2012-09-11 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
## 2 2012-09-12 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
## 3 2012-09-13 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
## 4 2012-09-14 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
## 5 2012-09-15 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
## 6 2012-09-16 00:00:00 w6        GN_Al_ICP 0.003         0         1  0.0001
```

Long vs wide data

There are many different ways data can be stored. One simple data dichotomy is wide vs long. Wide data has many columns, usually with each variable in a column. The data retrieved from the USGS is in wide format. Long data has many rows, with a column of variables describing other values. The data downloaded from Macrosheds comes in a long format. To switch between them we can use the functions 'pivot_wider()' and 'pivot_longer()'.

Here is an example of pivoting USGS data from wide to long. First, I'll retrieve the gauge height (local water elevation) and discharge data from a local gauge, Cache la Poudre River at Lincoln Ave.

```
lincoln <- readNWISuv(siteNumbers = '06752260',
                      parameterCd = c('00060', '00065'),
                      startDate = '2023-10-01',
                      endDate = '2024-9-30') %>%
  rename(q_cfs = 'X_00060_00000',
         gh_ft = 'X_00065_00000') %>%
  select(dateTime, gh_ft, q_cfs)

head(lincoln)
```

```
##              dateTime gh_ft q_cfs
## 1 2023-10-01 06:00:00  0.31  3.69
## 2 2023-10-01 06:05:00  0.31  3.69
## 3 2023-10-01 06:10:00  0.31  3.69
## 4 2023-10-01 06:15:00  0.31  3.69
## 5 2023-10-01 06:20:00  0.31  3.69
## 6 2023-10-01 06:25:00  0.31  3.69
```

Next, I will use 'pivot_longer()'. The 'cols' argument denotes what to pivot (in this case everything but the dateTime column), the 'names_to' is the new column to put the old column names as values, and the 'values_to' argument is the new column to put the old column values in. 'Pivot_wider()' works in a similar way.

```r
lincoln_long <- lincoln %>%
  pivot_longer(cols = -dateTime,
               names_to = 'var',
               values_to = 'val')

head(lincoln_long)
```

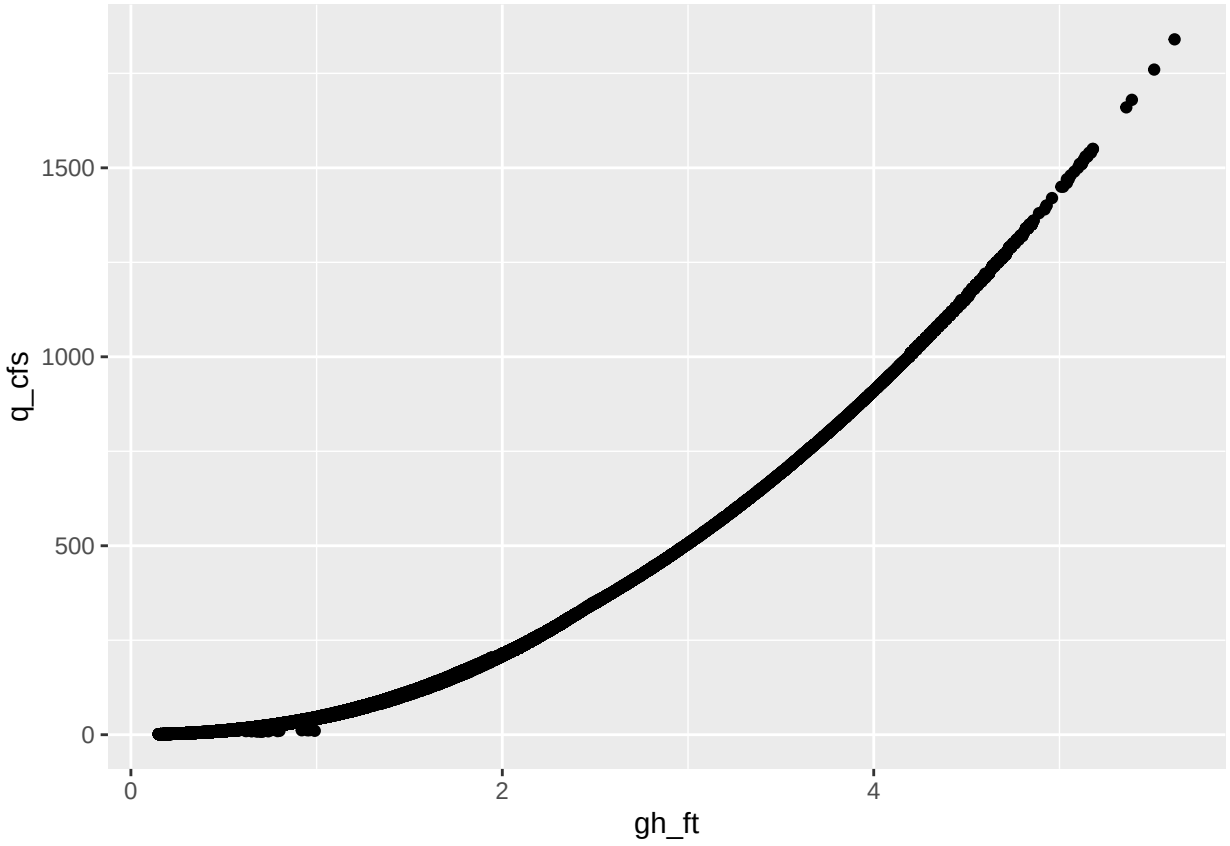
```
## # A tibble: 6 x 3
##   dateTime            var     val
##   <dttm>              <chr> <dbl>
## 1 2023-10-01 06:00:00 gh_ft  0.31
## 2 2023-10-01 06:00:00 q_cfs  3.69
## 3 2023-10-01 06:05:00 gh_ft  0.31
## 4 2023-10-01 06:05:00 q_cfs  3.69
## 5 2023-10-01 06:10:00 gh_ft  0.31
## 6 2023-10-01 06:10:00 q_cfs  3.69
```

Fitting models in R

while we won't be going into detail on modeling in the course, calculating simple lines of best fit is a common tool in the sciences. Let's model the relationship between gauge height and discharge in out model.

First let's look at the data visually.

```r
ggplot(lincoln, aes(x = gh_ft, y = q_cfs))+
         geom_point()
```



We know there's a pretty clear relationship there. Let's try a linear fit on it. To make a simple **l**inear **m**odel (the classic y=mx+b), we use the function 'lm()'. Then the function 'summary()' to view the details.

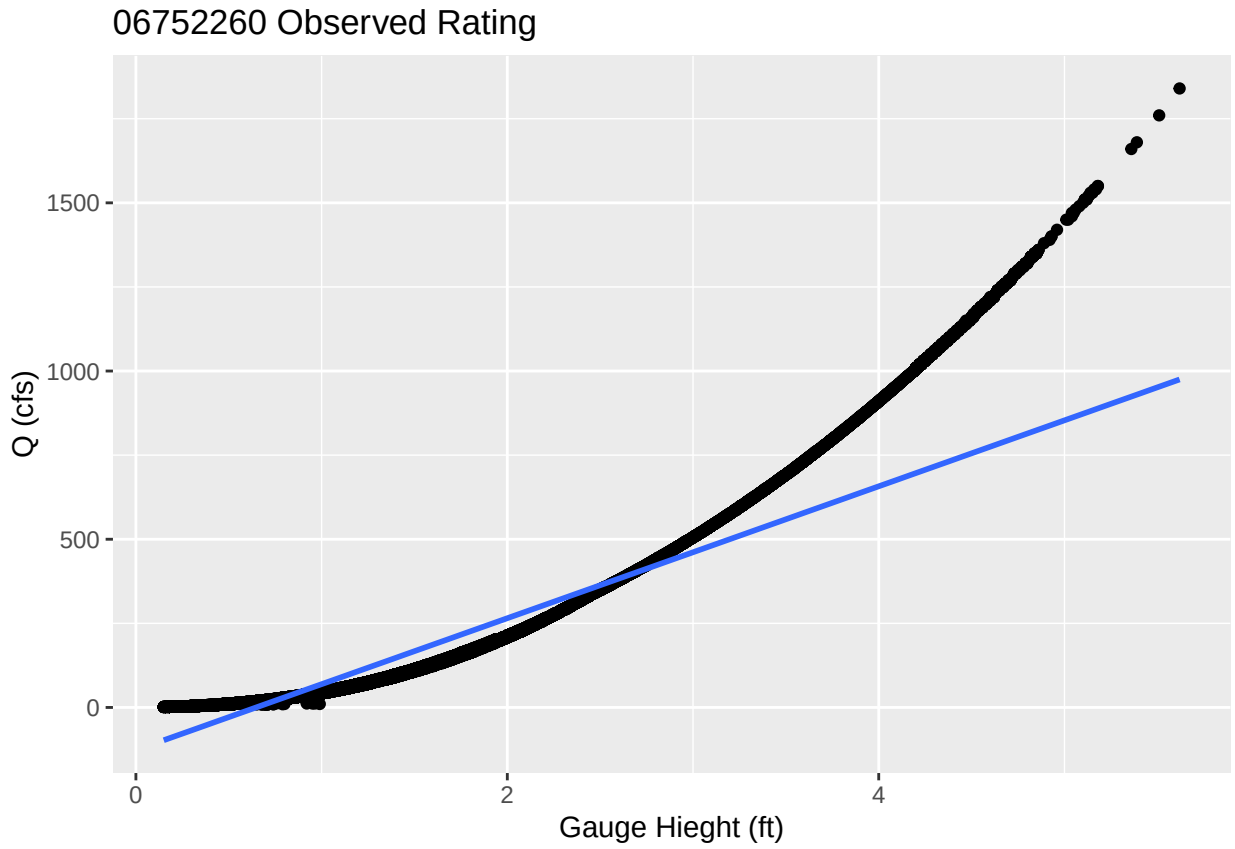
```
lincoln_model <- lm(lincoln$q_cfs ~ lincoln$gh_ft)
#str(lincoln_model)
summary(lincoln_model)
```

```
##
## Call:
## lm(formula = lincoln$q_cfs ~ lincoln$gh_ft)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -64.62 -40.59 -11.63  29.09 865.01
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -126.8876     0.3205  -395.9   <2e-16 ***
## lincoln$gh_ft  196.0640     0.2442   803.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.31 on 101494 degrees of freedom
##   (3748 observations deleted due to missingness)
## Multiple R-squared:  0.864,  Adjusted R-squared:  0.864
## F-statistic: 6.448e+05 on 1 and 101494 DF,  p-value: < 2.2e-16
```

This output tells us our R-squared is 0.86, with a slope of 196 cfs/ft and an intercept of -126 cfs.

We can replicate this in ggplot2 using geom_smooth.

```
ggplot(lincoln, aes(x = gh_ft, y = q_cfs))+
        geom_point()+
        geom_smooth(method = 'lm')+
  labs(x = 'Gauge Hieght (ft)',
       y = 'Q (cfs)',
       title = '06752260 Observed Rating')
```



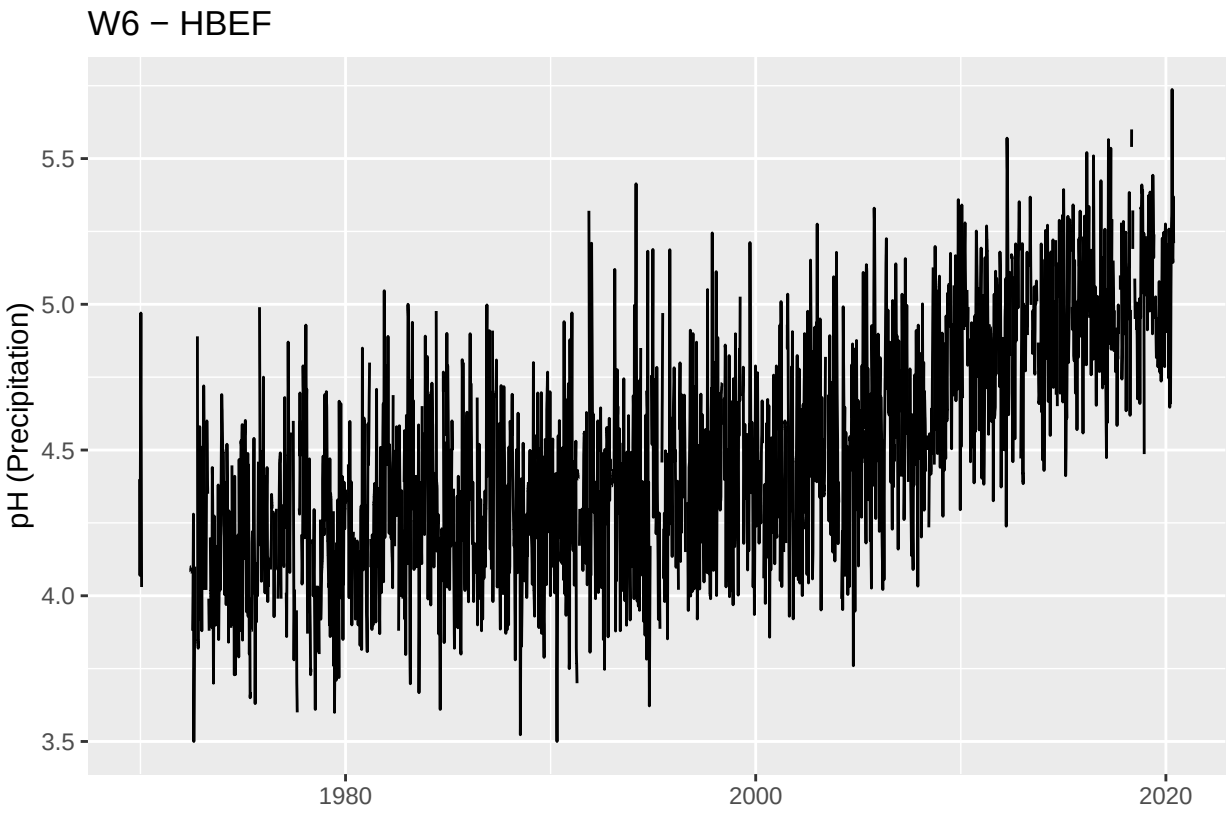
Ratings are more complicated than a simple linear relationship, so this line is likely not a perfect fit for modeling high or low flows. In this case the flows at gauge heights greater than 3 feet (when the stream is very deep and the flow is very high) is skewing our line.

Hubbard Brook

Let's look at the trend in rainwater pH at W6 since the clean air act amendments.

```
rain_ph <- p_chem %>%
  filter(var == 'GN_pH') %>%
  pivot_wider(id_cols = datetime,
              names_from = var,
              values_from = val) %>%
  rename('ph' = GN_pH)
```

```
ggplot(rain_ph, aes(x = datetime, y = ph)) +
  geom_line()+
  labs(x = '',
       y = 'pH (Precipitation)',
       title = 'W6 - HBEF')
```



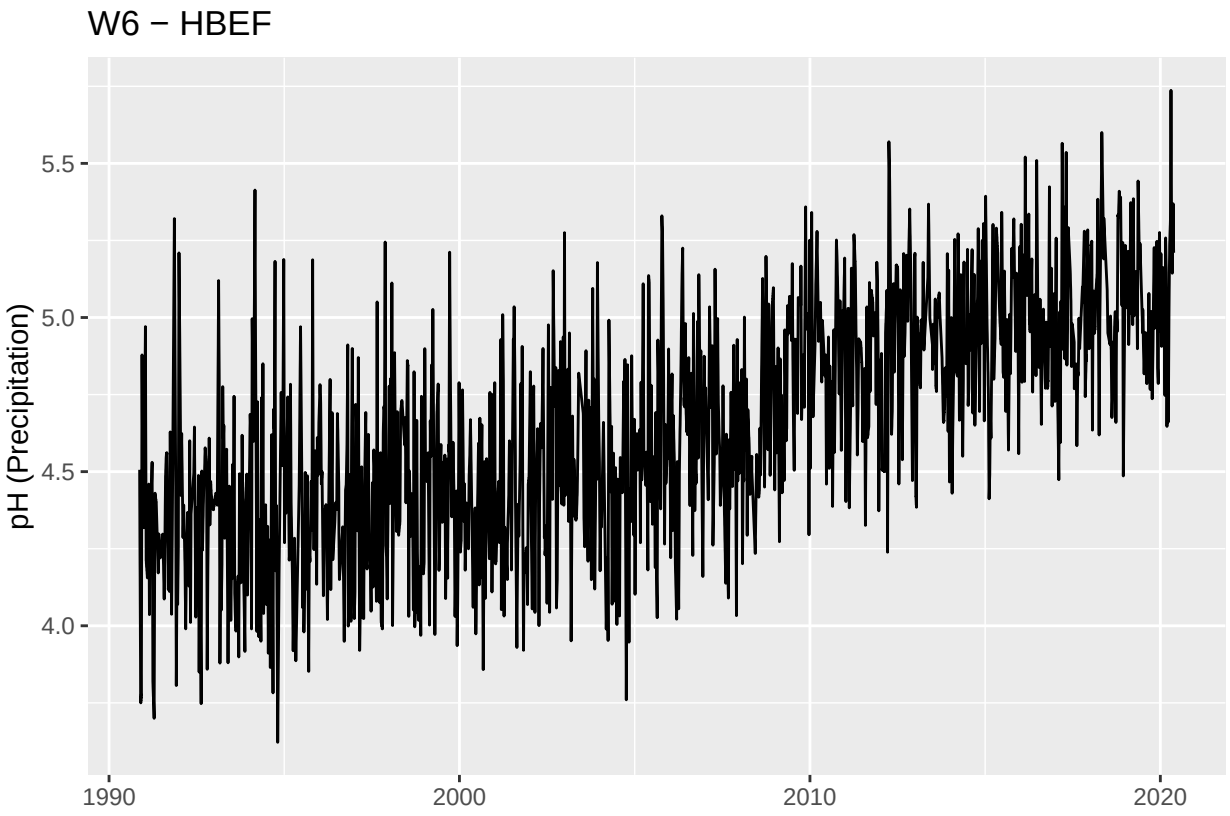
We'll use the 'trends' package to test for a trend after 11/15/1990 using a Mann-Kendall Trend test.

```
#install.packages('trend')
library(trend)
?mk.test()
```

Read the 'details' section of the manual page for this function. Some are very opaque and off in the land of math theory. But this one is very approachable. To test for a trend in pH, we will call the column using the 'mk.test()' function.

```
post_ccaa <- rain_ph %>%
  filter(datetime>as.POSIXct('1990-11-15'),
         !is.na(ph))

ggplot(post_ccaa, aes(x = datetime, y = ph)) +
  geom_line()+
  labs(x = '',
       y = 'pH (Precipitation)',
       title = 'W6 - HBEF')
```




```
mk.test(post_ccaa$ph)
```

```
##
##  Mann-Kendall trend test
##
## data:  post_ccaa$ph
## z = 76.262, n = 9451, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 2.335813e+07 9.381228e+10 5.230801e-01
```

This test gives strong statistical backing to the visually obvious trend. To find the strength of the effect, we could test the correlation using a Sen's Slope Test (a common statistical test), 'sens.slope()' from the 'trend' package.

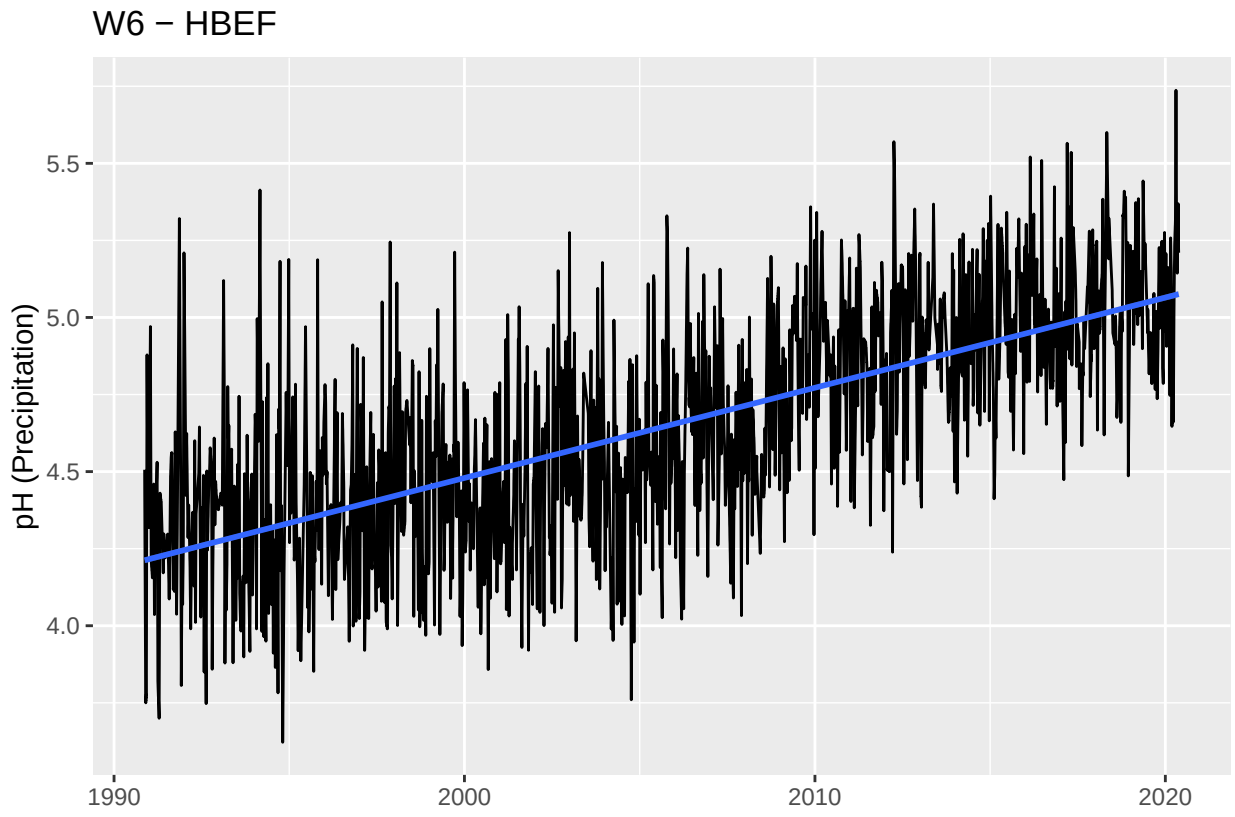
```
?sens.slope()
sens.slope(post_ccaa$ph)
```

```
##
##  Sen's slope
##
## data:  post_ccaa$ph
## z = 76.262, n = 9451, p-value < 2.2e-16
```

```
## alternative hypothesis: true z is not equal to 0
## 95 percent confidence interval:
##  9.072975e-05 9.430759e-05
## sample estimates:
## Sen's slope
## 9.25222e-05
```

The output confirms strong evidence for a reduction in rain acidity at WS following the Clean Air Act Amendments of 1990. Adding a simple linear model our timeseries plot is easy in ggplot2 using the 'geom_smooth()' function, declaring 'lm' for 'linear model' as our method. We can produce the same model using the function 'lm()' and can view it with 'summary()'.

```
ggplot(post_ccaa, aes(x = datetime, y = ph)) +
  geom_line()+
  labs(x = '',
       y = 'pH (Precipitation)',
       title = 'W6 - HBEF')+
  geom_smooth(method = 'lm',
              formula = 'y ~ x')
```




```
  model <- lm(post_ccaa$ph ~ post_ccaa$datetime)
  summary(model)
```

```
##
## Call:
```

```
## lm(formula = post_ccaa$ph ~ post_ccaa$datetime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85868 -0.16444 -0.00658  0.15854  1.10465
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.602e+00  1.059e-02   340.0   <2e-16 ***
## post_ccaa$datetime 9.271e-10  9.205e-12   100.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2389 on 9449 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.5177
## F-statistic: 1.014e+04 on 1 and 9449 DF,  p-value: < 2.2e-16
```

Assignment

Q0 (10 pts):

Call the 'here()' function to check your working directory.

Q1 (10 pts):

Read in the the '.feather' data files for Q and stream chemistry at W6.

Q2 (20 pts):

Make your data useful and understood.

Part A:

Combine the chemistry and discharge data objects into a single dataframe. Remove the columns 'ms_status' , 'ms_interp', and 'val_err' (these are data quality flags we won't be using now). (Hint: look into the 'bind' functions.)

Part B:

Describe (in text) the structure of your new, combined object. What are the columns and what do they contain? Is your data in a long or wide format? When does your dataset begin and end?

Part C:

Use the 'unique()' function to list all variables represented in the dataset.

Q3 (20 pts):

Part A:

Plot a timeseries of pH at HBEF W6. Color the data by whether it is pre or post Clean Air Act Amendments of 1990, or use a vertical line to denote its passage. Make the figure look presentable, with labels, a title, and a descriptive caption.(Hint: the code for pH in this dataset is 'GN_pH'.)

Part B:

Fit two linear models, one pH at W6 before and and the other after the passage of the Clean Air Act Amendments of 1990. Display your results using the 'summary()' function.

Part C:

Briefly interpret your model outputs in context.

Q4 (20 pts)

Aluminum is highly abundant (~7% of Earth's crust) and is toxic to life. The predominant form it takes in solution is pH dependent, with the Alˆ(3+) ion the most toxic to aquatic life. Now, let's just try and observe the effect.

Part A:

Pivot your data from long format, to wide.

Part B:

Make a scatterplot of Al vs pH. (Hint: the code for Al in this dataset is 'GN_Al_ICP') Make the figure look presentable, with labels, a title, and a descriptive caption. (Reminder: your final .html should *not* show warnings, messages, errors, etc.)

Q5 (20 pts)

Part A:

What is the effect of reducing acid rain on in aluminum toxicity in W6?

Part B:

What other factors may be acting on as a control on aluminum at W6?

Bonus Question (+10 pts)

Create a model describing the relationship between pH and Al at HBEF (it is not simple linear). Add your model to the graph from Q4.