Assignment 10: Data Scraping

Sayra Martinez

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on data scraping.

Directions

1. Rename this file `<FirstLast>_A10_DataScraping.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure your code is tidy; use line breaks to ensure your code fits in the knitted output.
5. Be sure to **answer the questions** in this assignment document.
6. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Load the packages `tidyverse`, `rvest`, and any others you end up using.
- Check your working directory

```r
#1 Loading packages and checking directory
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
library(tidyverse);library(rvest); library(dataRetrieval);library(tidycensus)
library(ggplot2); library(scales); library("ggplot2"); library("lubridate"); library("dplyr")
getwd()
```

```
## [1] "/home/guest/EDA_Spring2024"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2022 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Scroll down and select the LWSP link next to Durham Municipality.
- Note the web address: https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022

Indicate this website as the as the URL to be scraped. (In other words, read the contents into an `rvest` webpage object.)

```r
# 2
URL <- "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2022"
water_website <- read_html(URL)
```

3. The data we want to collect are listed below:

- From the “1. System Information” section:
- Water system name
- PWSID
- Ownership
- From the “3. Water Supply Sources” section:
- Maximum Day Use (MGD) - for each month

In the code chunk below scrape these values, assigning them to four separate variables.

> HINT: The first value should be “Durham”, the second “03-32-010”, the third “Municipality”, and the last should be a vector of 12 numeric values (represented as strings)“.

```r
# 3 From System information section
system_name <- water_website %>%
    html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
    html_text(trim = T)
PWSID <- water_website %>%
    html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
    html_text(trim = T)
ownership <- water_website %>%
    html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
    html_text(trim = T)
# From water supply sources
MGD <- water_website %>%
    html_nodes("th~ td+ td") %>%
    html_text(trim = T) %>%
    as.numeric()

# Just to check the extracted data
print(list(System_Name = system_name, PWSID = PWSID, Ownership = ownership,
    MGD = MGD))
```

```
## $System_Name
## [1] "Durham"
##
## $PWSID
## [1] "03-32-010"
##
## $Ownership
## [1] "Municipality"
##
## $MGD
##  [1] 36.10 43.42 52.49 30.50 42.59 34.88 39.91 43.32 32.53 34.66 41.80 37.53
```

4. Convert your scraped data into a dataframe. This dataframe should have a column for each of the 4 variables scraped and a row for the month corresponding to the withdrawal data. Also add a Date column that includes your month and year in data format. (Feel free to add a Year column too, if you wish.)

   TIP: Use `rep()` to repeat a value when creating a dataframe.

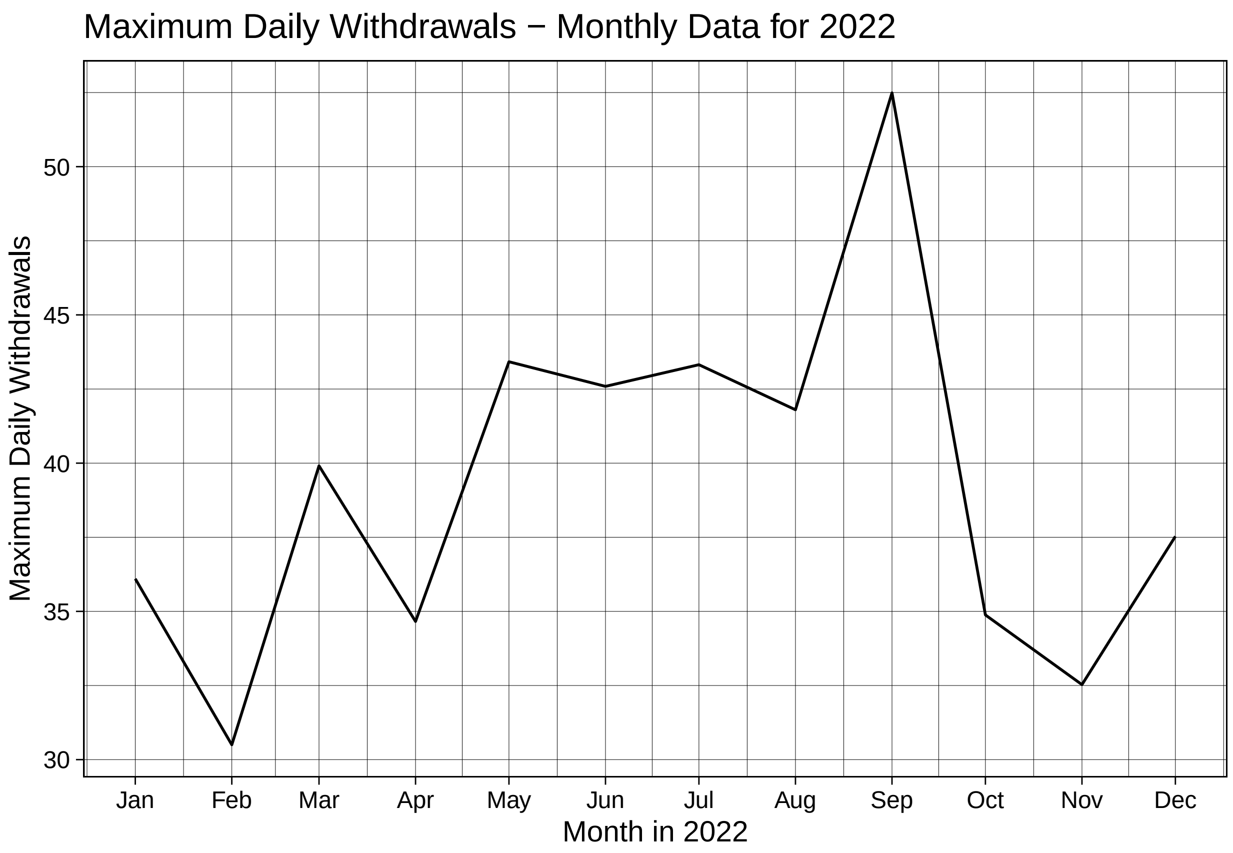
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in chronological order. You can overcome this by creating a month column manually assigning values in the order the data are scraped: "Jan", "May", "Sept", "Feb", etc… Or, you could scrape month values from the web page…

5. Create a line plot of the maximum daily withdrawals across the months for 2022

```
# 4 Preparing the information
System_name <- rep("Durham", 12)
PWSID <- rep("03-32-010", 12)
Ownership <- rep("Municipality", 12)
Month <- water_website %>%
    html_nodes(".fancy-table:nth-child(31) tr+ tr th") %>%
    html_text(trim = T)
Year <- rep("2022", 12)
Date <- as.Date(paste("2022", Month, "01", sep = "-"), format = "%Y-%b-%d")

# Creating my data frame
Water_Supply <- data.frame(System_Name = System_name, PWSID = PWSID,
    Ownership = Ownership, MGD = MGD, Date = Date, Month = Month)

# 5
Water_Durhamplot <- ggplot(Water_Supply, aes(x = Date, y = MGD)) +
    geom_line(group = 1) + labs(x = "Month in 2022", y = "Maximum Daily Withdrawals") +
    theme_linedraw() + ggtitle("Maximum Daily Withdrawals - Monthly Data for 2022") +
    scale_x_date(date_labels = "%b", date_breaks = "1 month")
Water_Durhamplot
```



6. Note that the PWSID and the year appear in the web address for the page we scraped. Construct a function using your code above that can scrape data for any PWSID and year for which the NC DEQ has data. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

```r
# 6.
scraping <- function(PWSID, Year) {
    the_URL <- paste0("https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=",
        PWSID, "&year=", Year)
    water_website <- read_html(the_URL)

    # scraping data from the given website
    system_name <- water_website %>%
        html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
        html_text()
    PWSID <- water_website %>%
        html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
        html_text(trim = T)
    ownership <- water_website %>%
        html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
        html_text(trim = T)
    # From water supply sources
    MGD <- water_website %>%
        html_nodes("th~ td+ td") %>%
        html_text(trim = T) %>%
        as.numeric()
    # Creating Date information Month <- water_website %>%
    # html_nodes('.fancy-table:nth-child(31) tr+ tr th')
    # %>% html_text(trim=T) #Although it was correctly
    # extracting the information for Durham, it didn't work
    # with the rest, so I decided to do it manually.
    Month <- c("Jan", "May", "Sep", "Feb", "Jun", "Oct", "Mar",
        "Jul", "Nov", "Apr", "Aug", "Dec")
    Date <- as.Date(paste(Year, Month, "01", sep = "-"), format = "%Y-%b-%d")

    # Into data frame (I tried to make the the rep and
    # creating the Dates directly and yes, it is possible)
    Water_Supply_scraping <- data.frame(System_Name = rep(system_name,
        12), PWSID = rep(PWSID, 12), Ownership = rep(ownership,
        12), MGD = MGD, Year = rep(Year, 12), Date = Date)
    Water_Supply_scraping
}
```

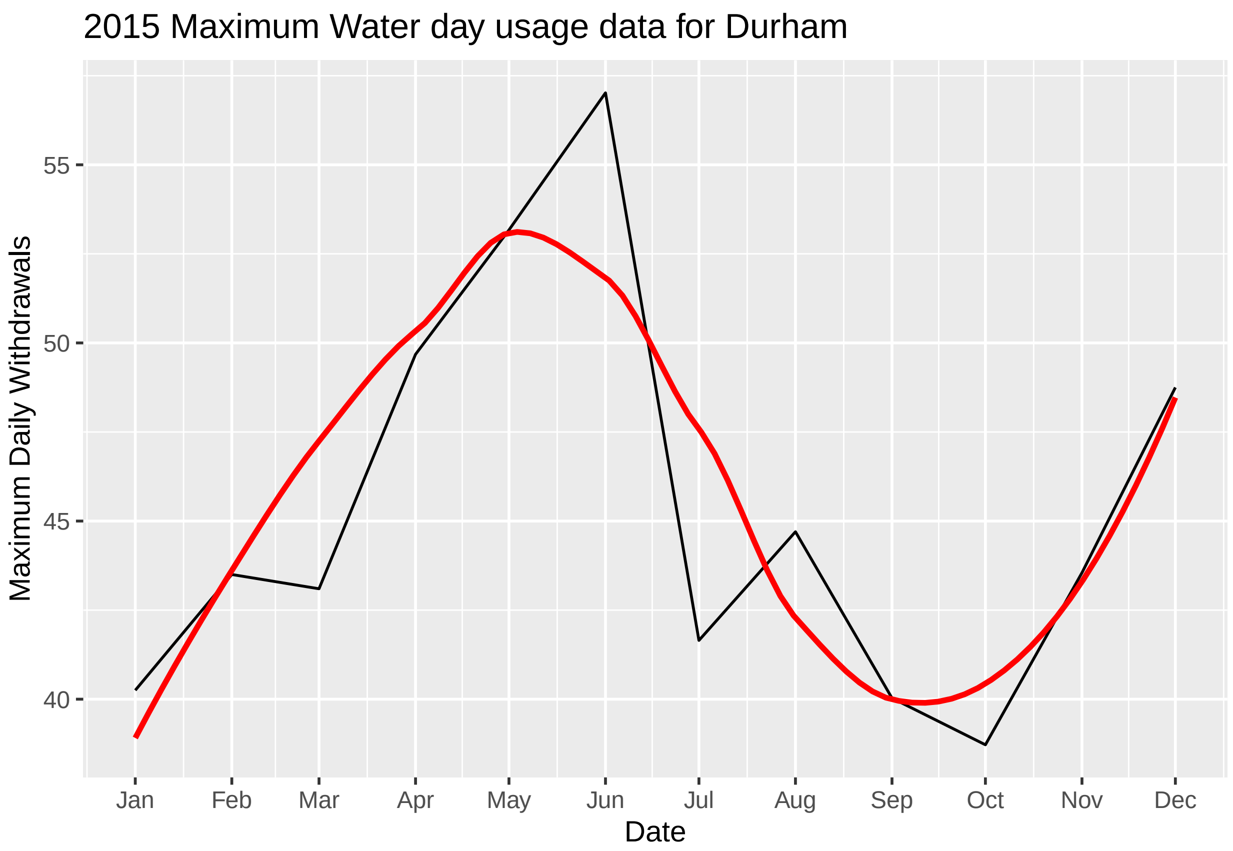
7. Use the function above to extract and plot max daily withdrawals for Durham (PWSID=‘03-32-010’) for each month in 2015

```
# 7
Durham_2015 <- scraping("03-32-010", "2015")
Durham_2015
```

```
##    System_Name     PWSID    Ownership   MGD Year       Date
## 1       Durham 03-32-010 Municipality 40.25 2015 2015-01-01
## 2       Durham 03-32-010 Municipality 53.17 2015 2015-05-01
## 3       Durham 03-32-010 Municipality 40.03 2015 2015-09-01
## 4       Durham 03-32-010 Municipality 43.50 2015 2015-02-01
## 5       Durham 03-32-010 Municipality 57.02 2015 2015-06-01
## 6       Durham 03-32-010 Municipality 38.72 2015 2015-10-01
## 7       Durham 03-32-010 Municipality 43.10 2015 2015-03-01
## 8       Durham 03-32-010 Municipality 41.65 2015 2015-07-01
## 9       Durham 03-32-010 Municipality 43.55 2015 2015-11-01
## 10      Durham 03-32-010 Municipality 49.68 2015 2015-04-01
## 11      Durham 03-32-010 Municipality 44.70 2015 2015-08-01
## 12      Durham 03-32-010 Municipality 48.75 2015 2015-12-01
```

```
Durham_2015_plot <- ggplot(Durham_2015, aes(x = Date, y = MGD)) +
    geom_line() + geom_smooth(method = "loess", se = F, color = "red") +
    labs(title = paste(2015, "Maximum Water day usage data for Durham"),
        y = "Maximum Daily Withdrawals", x = "Date") + scale_x_date(date_labels = "%b",
    date_breaks = "1 month")
Durham_2015_plot
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



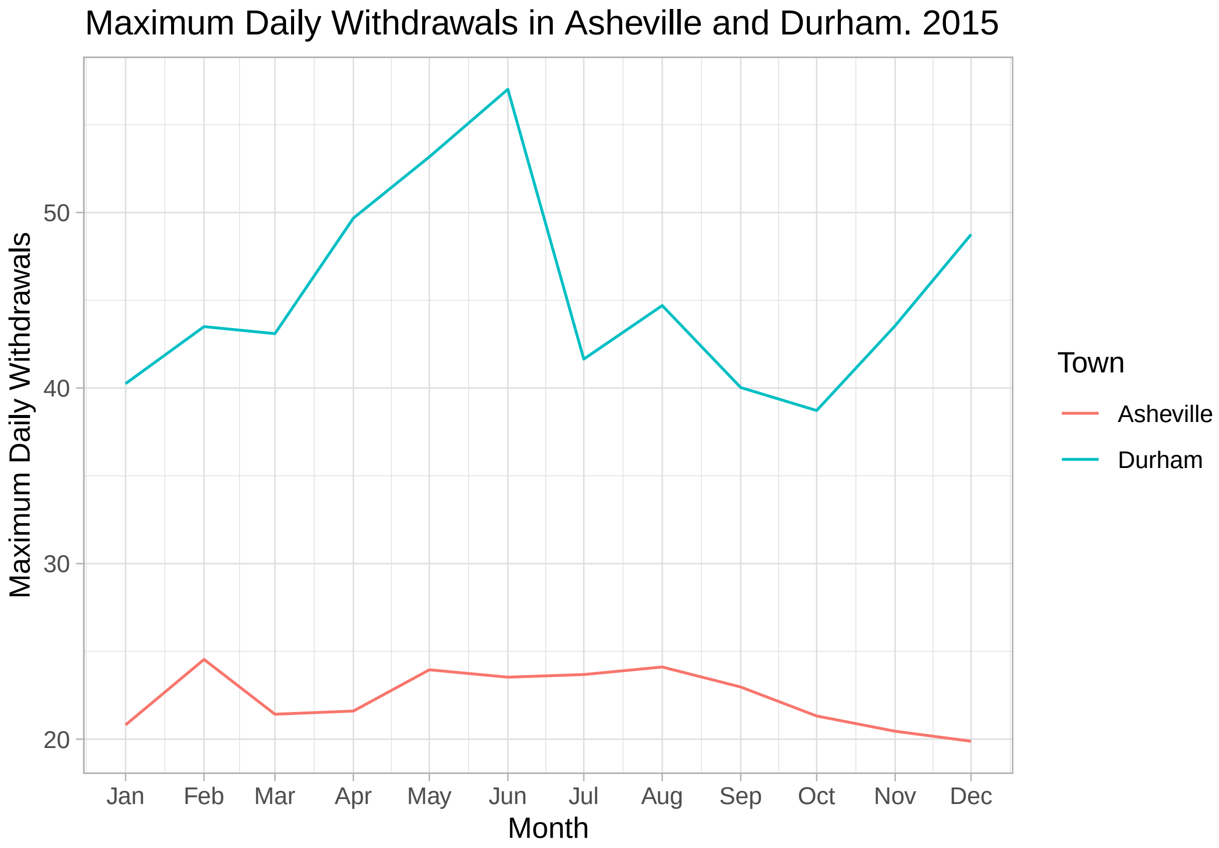
8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares Asheville's to Durham's water withdrawals.

```
# 8
Asheville_2015 <- scraping("01-11-010", "2015")
Asheville_2015
```

```
##    System_Name     PWSID    Ownership   MGD Year       Date
## 1    Asheville 01-11-010 Municipality 20.81 2015 2015-01-01
## 2    Asheville 01-11-010 Municipality 23.95 2015 2015-05-01
## 3    Asheville 01-11-010 Municipality 22.97 2015 2015-09-01
## 4    Asheville 01-11-010 Municipality 24.54 2015 2015-02-01
## 5    Asheville 01-11-010 Municipality 23.53 2015 2015-06-01
## 6    Asheville 01-11-010 Municipality 21.32 2015 2015-10-01
## 7    Asheville 01-11-010 Municipality 21.42 2015 2015-03-01
## 8    Asheville 01-11-010 Municipality 23.68 2015 2015-07-01
## 9    Asheville 01-11-010 Municipality 20.45 2015 2015-11-01
## 10   Asheville 01-11-010 Municipality 21.60 2015 2015-04-01
## 11   Asheville 01-11-010 Municipality 24.11 2015 2015-08-01
## 12   Asheville 01-11-010 Municipality 19.88 2015 2015-12-01
```

```
two_towns <- bind_rows(Durham_2015, Asheville_2015)

ggplot(two_towns, aes(x = Date, y = MGD, color = System_Name)) +
    geom_line() + labs(x = "Month", y = "Maximum Daily Withdrawals",
    color = "Town") + theme_light() + ggtitle("Maximum Daily Withdrawals in Asheville and Durham. 2015")
    theme(plot.title = element_text(hjust = 0.1)) + scale_x_date(date_labels = "%b",
    date_breaks = "1 month")
```



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2021.Add a smoothed line to the plot (method = 'loess').

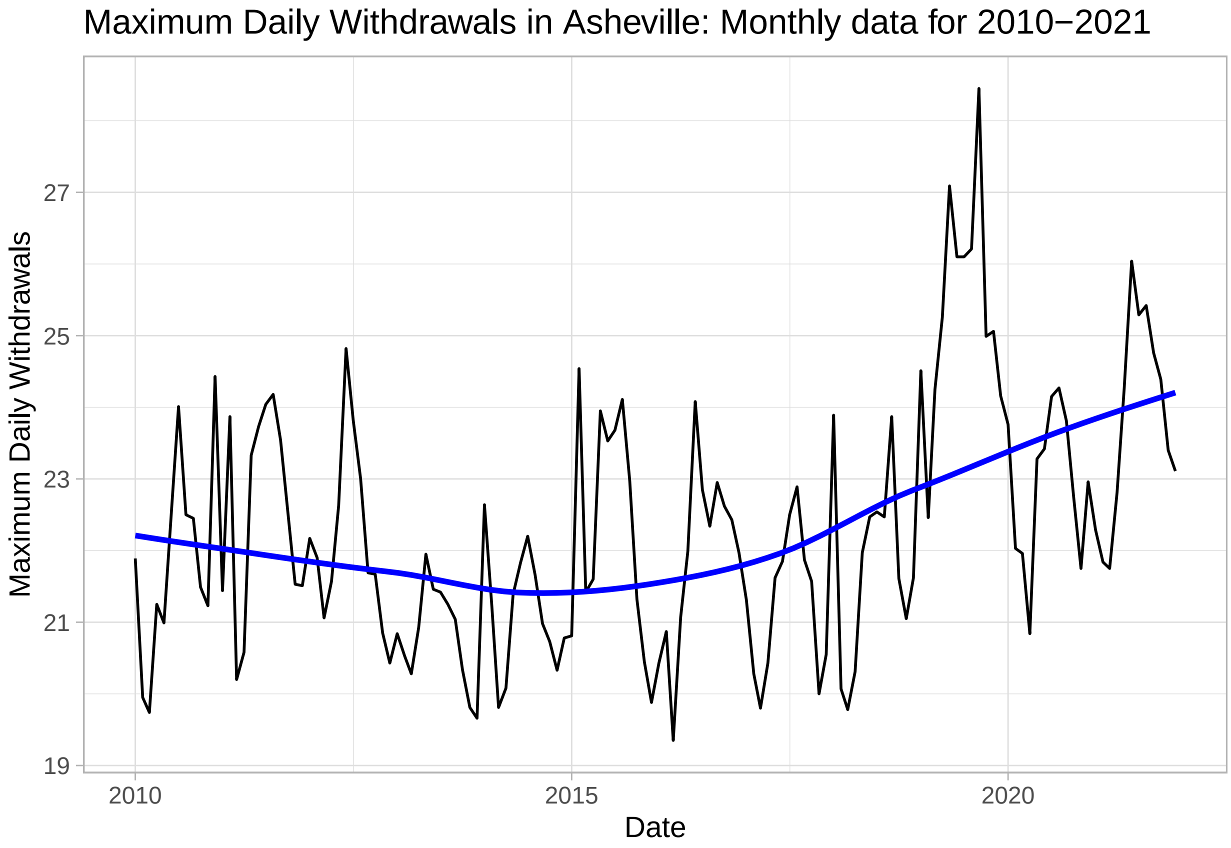
   TIP: See Section 3.2 in the "10_Data_Scraping.Rmd" where we apply "map2()" to iteratively run a function over two inputs. Pipe the output of the map2() function to `bindrows()` to combine the dataframes into a single one.

```
# 9
Asheville_decade <- 2010:2021
Asheville_id <- rep("01-11-010", length(Asheville_decade))

Asheville_df <- map2(Asheville_id, Asheville_decade, scraping)
Asheville_df <- bind_rows(Asheville_df)

Asheville_plot <- ggplot(Asheville_df, aes(x = Date, y = MGD)) +
    geom_line() + geom_smooth(method = "loess", se = F, color = "blue") +
    labs(x = "Date", y = "Maximum Daily Withdrawals", color = "Town") +
    theme_light() + ggtitle("Maximum Daily Withdrawals in Asheville: Monthly data for 2010-2021")
Asheville_plot
```

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



Question: Just by looking at the plot (i.e. not running statistics), does Asheville have a trend in water usage over time? > Answer: Yes, the plot shows that from 2010 and around 2015, the water usage was slighltly decreasing, but it recovers and started increasing from then. >