Assignment 10: Data Scraping

Hanbin Lyu

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

```
#1
library(tidyverse)
library(rvest)

getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2023 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=2023

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

```
#2
webpage = read_html(
  "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2023")
webpage
```

```
## {html_document}
## <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
## [1] <head>\n<title>DWR :: Local Water Supply Planning</title>\n<meta http-equ ...
## [2] <body id="plan">\r\n<!--<div id="division-header">\r\n<a name="top" href= ...
```

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)".

```
#3
systemname = webpage %>%
  html_node("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
systemname
```

```
## [1] "Durham"
```

```
pwsid = webpage %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
pwsid
```

```
## [1] "03-32-010"
```

```
ownership = webpage %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
ownership
```

```
## [1] "Municipality"
```

```
mgd = webpage %>%
  html_nodes("th~ td+ td") %>%
  html_text()
mgd
```

```
##  [1] "28.9000" "33.3000" "43.7000" "30.0000" "40.0000" "37.2300" "34.2000"
##  [8] "44.9000" "40.3500" "30.9000" "56.7000" "33.3000"
```

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...

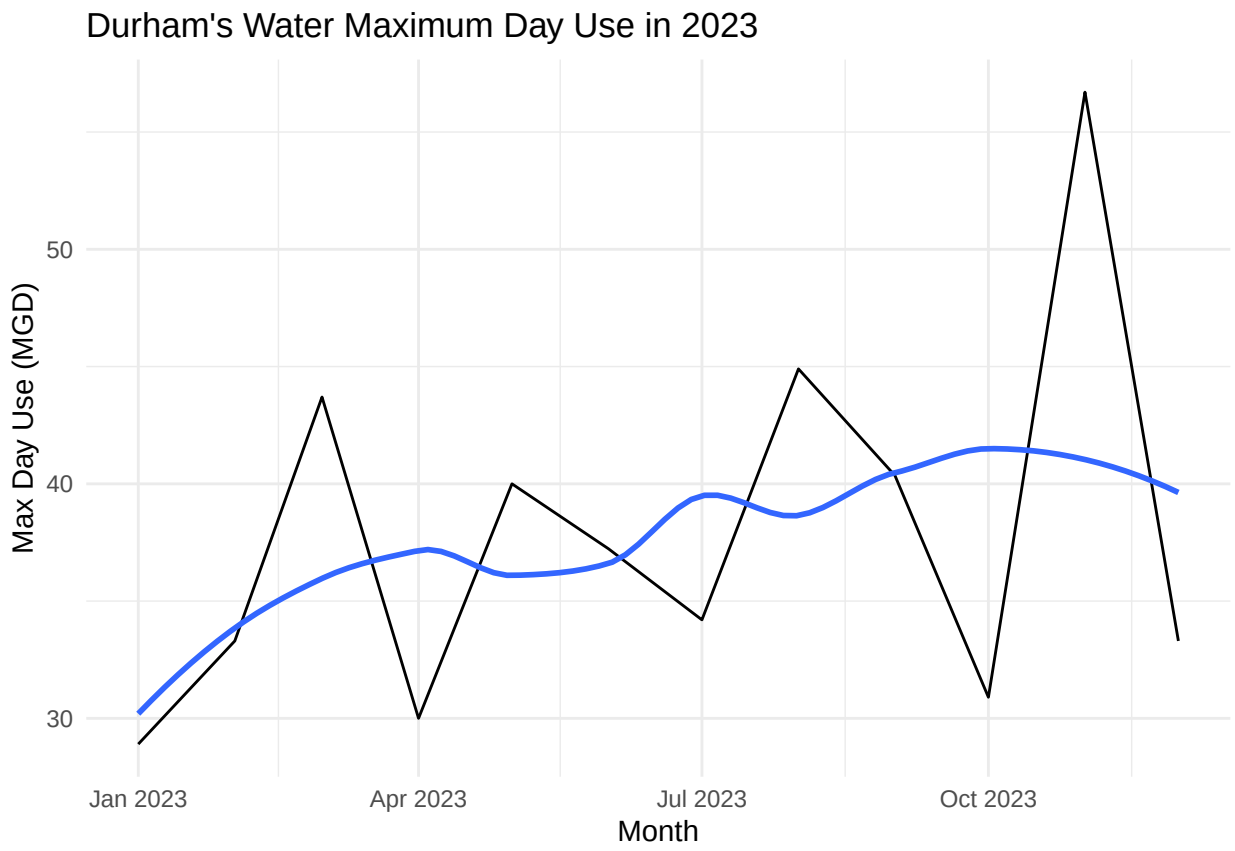
5. Create a line plot of the maximum daily withdrawals across the months for 2023, making sure, the months are presented in proper sequence.

```
#4
durham_2023 = data.frame(
  "Month" = rep(1:12),
  "Year" = rep(2023, 12),
  "Max_Day_Use" = as.numeric(mgd)) %>%
  mutate(
    Water_System_Name = systemname,
    PWSID = pwsid,
    Ownership = ownership,
    Date = my(paste(Month,"-",Year)))
durham_2023
```

```
##    Month Year Max_Day_Use Water_System_Name     PWSID    Ownership       Date
## 1      1 2023       28.90            Durham 03-32-010 Municipality 2023-01-01
## 2      2 2023       33.30            Durham 03-32-010 Municipality 2023-02-01
## 3      3 2023       43.70            Durham 03-32-010 Municipality 2023-03-01
## 4      4 2023       30.00            Durham 03-32-010 Municipality 2023-04-01
## 5      5 2023       40.00            Durham 03-32-010 Municipality 2023-05-01
## 6      6 2023       37.23            Durham 03-32-010 Municipality 2023-06-01
## 7      7 2023       34.20            Durham 03-32-010 Municipality 2023-07-01
## 8      8 2023       44.90            Durham 03-32-010 Municipality 2023-08-01
## 9      9 2023       40.35            Durham 03-32-010 Municipality 2023-09-01
## 10    10 2023       30.90            Durham 03-32-010 Municipality 2023-10-01
## 11    11 2023       56.70            Durham 03-32-010 Municipality 2023-11-01
## 12    12 2023       33.30            Durham 03-32-010 Municipality 2023-12-01
```

```r
#5
ggplot(durham_2023, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Durham's Water Maximum Day Use in 2023",
    x = "Month",
    y = "Max Day Use (MGD)") +
  theme_minimal()
```

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



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, returning a dataframe. **Be sure to modify the code to reflect the year and site (pwsid) scraped**.

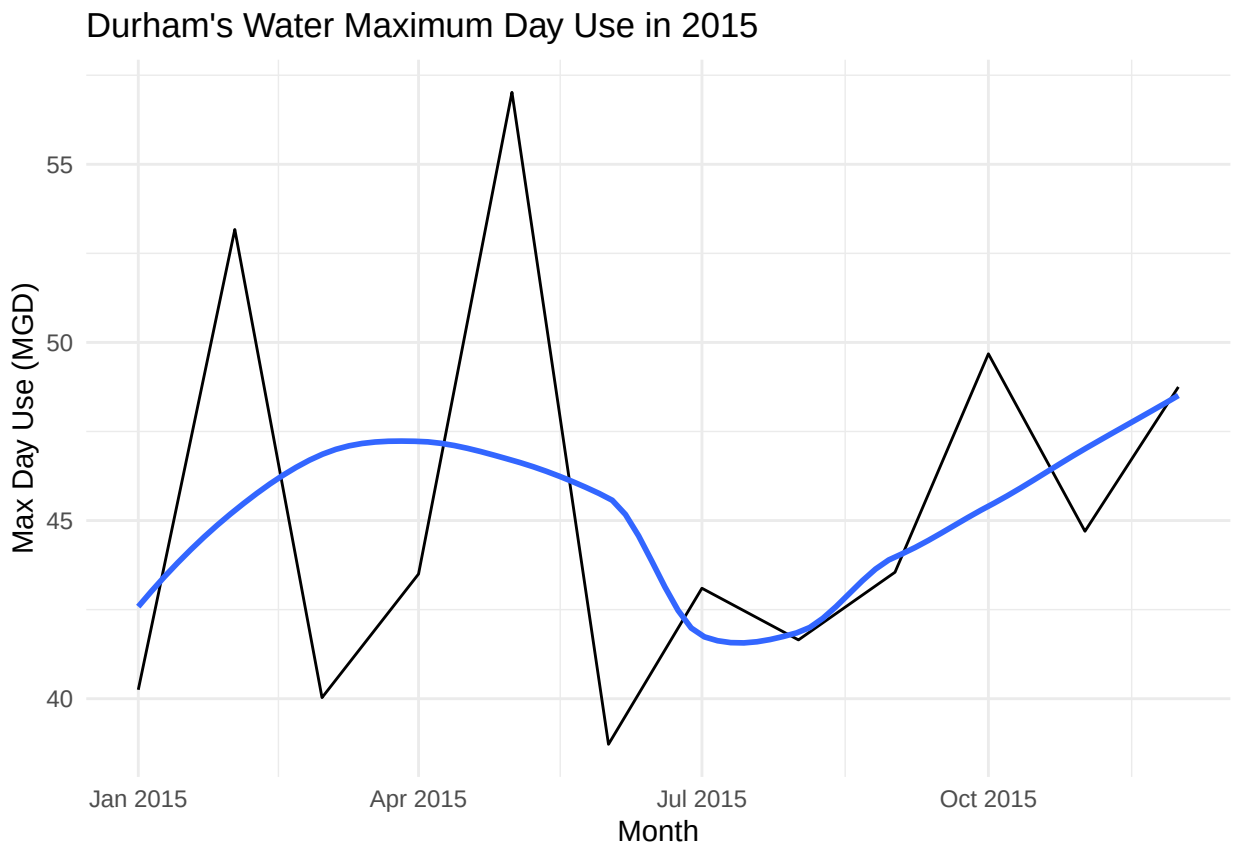
```r
#6.
scrape.it = function(the_pwsid, the_year){
the_website = read_html(paste0('https://www.ncwater.org/WUDC/app/LWSP/report.php',
                               '?pwsid=', the_pwsid, '&year=', the_year))

the_systemname_tag = 'div+ table tr:nth-child(1) td:nth-child(2)'
the_PWSID_tag = 'td tr:nth-child(1) td:nth-child(5)'
the_ownership_tag = 'div+ table tr:nth-child(2) td:nth-child(4)'
```

```
the_max_tag = 'th~ td+ td'

the_system_name = the_website %>%
  html_nodes(the_systemname_tag) %>%
  html_text()
the_PWSID = the_website %>%
  html_nodes(the_PWSID_tag) %>%
  html_text()
the_ownership = the_website %>%
  html_nodes(the_ownership_tag) %>%
  html_text()
the_max_day_use =the_website %>%
  html_nodes(the_max_tag) %>%
  html_text()

dataframe = data.frame(
  "Month" = rep(1:12),
  "Year" = rep(the_year, 12),
  "Max_Day_Use" = as.numeric(the_max_day_use)) %>%
   mutate(
    Water_System_Name = !!the_system_name,
    PWSID = !!the_PWSID,
    Ownership = !!the_ownership,
    Date = my(paste(Month,"-",Year)))

return(dataframe)
}
```

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 = scrape.it('03-32-010', 2015)

ggplot(durham_2015, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Durham's Water Maximum Day Use in 2015",
    x = "Month",
    y = "Max Day Use (MGD)") +
  theme_minimal()
```

```
## `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.

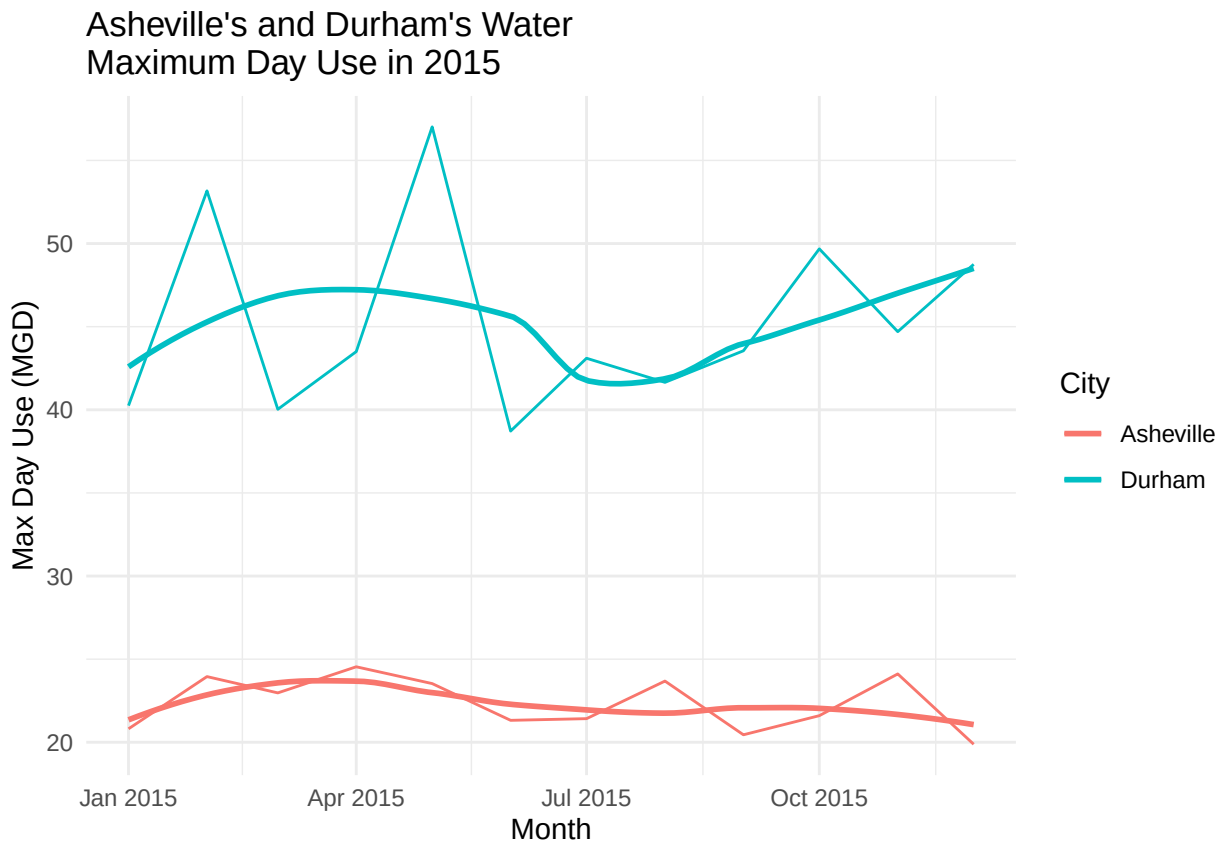
```
#8
asheville_2015 = scrape.it('01-11-010', 2015)

combine = rbind(durham_2015, asheville_2015)
combine
```

```
##    Month Year Max_Day_Use Water_System_Name     PWSID    Ownership       Date
## 1      1 2015       40.25            Durham 03-32-010 Municipality 2015-01-01
## 2      2 2015       53.17            Durham 03-32-010 Municipality 2015-02-01
## 3      3 2015       40.03            Durham 03-32-010 Municipality 2015-03-01
## 4      4 2015       43.50            Durham 03-32-010 Municipality 2015-04-01
## 5      5 2015       57.02            Durham 03-32-010 Municipality 2015-05-01
## 6      6 2015       38.72            Durham 03-32-010 Municipality 2015-06-01
## 7      7 2015       43.10            Durham 03-32-010 Municipality 2015-07-01
## 8      8 2015       41.65            Durham 03-32-010 Municipality 2015-08-01
## 9      9 2015       43.55            Durham 03-32-010 Municipality 2015-09-01
## 10    10 2015       49.68            Durham 03-32-010 Municipality 2015-10-01
## 11    11 2015       44.70            Durham 03-32-010 Municipality 2015-11-01
## 12    12 2015       48.75            Durham 03-32-010 Municipality 2015-12-01
## 13     1 2015       20.81         Asheville 01-11-010 Municipality 2015-01-01
## 14     2 2015       23.95         Asheville 01-11-010 Municipality 2015-02-01
```

```
## 15     3 2015       22.97         Asheville 01-11-010 Municipality 2015-03-01
## 16     4 2015       24.54         Asheville 01-11-010 Municipality 2015-04-01
## 17     5 2015       23.53         Asheville 01-11-010 Municipality 2015-05-01
## 18     6 2015       21.32         Asheville 01-11-010 Municipality 2015-06-01
## 19     7 2015       21.42         Asheville 01-11-010 Municipality 2015-07-01
## 20     8 2015       23.68         Asheville 01-11-010 Municipality 2015-08-01
## 21     9 2015       20.45         Asheville 01-11-010 Municipality 2015-09-01
## 22    10 2015       21.60         Asheville 01-11-010 Municipality 2015-10-01
## 23    11 2015       24.11         Asheville 01-11-010 Municipality 2015-11-01
## 24    12 2015       19.88         Asheville 01-11-010 Municipality 2015-12-01
```

```r
ggplot(combine, aes(x = Date, y = Max_Day_Use, color = Water_System_Name)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Asheville's and Durham's Water\nMaximum Day Use in 2015",
    x = "Month",
    y = "Max Day Use (MGD)",
    color = "City") +
  theme_minimal()
```

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



9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2018 thru 2022.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.

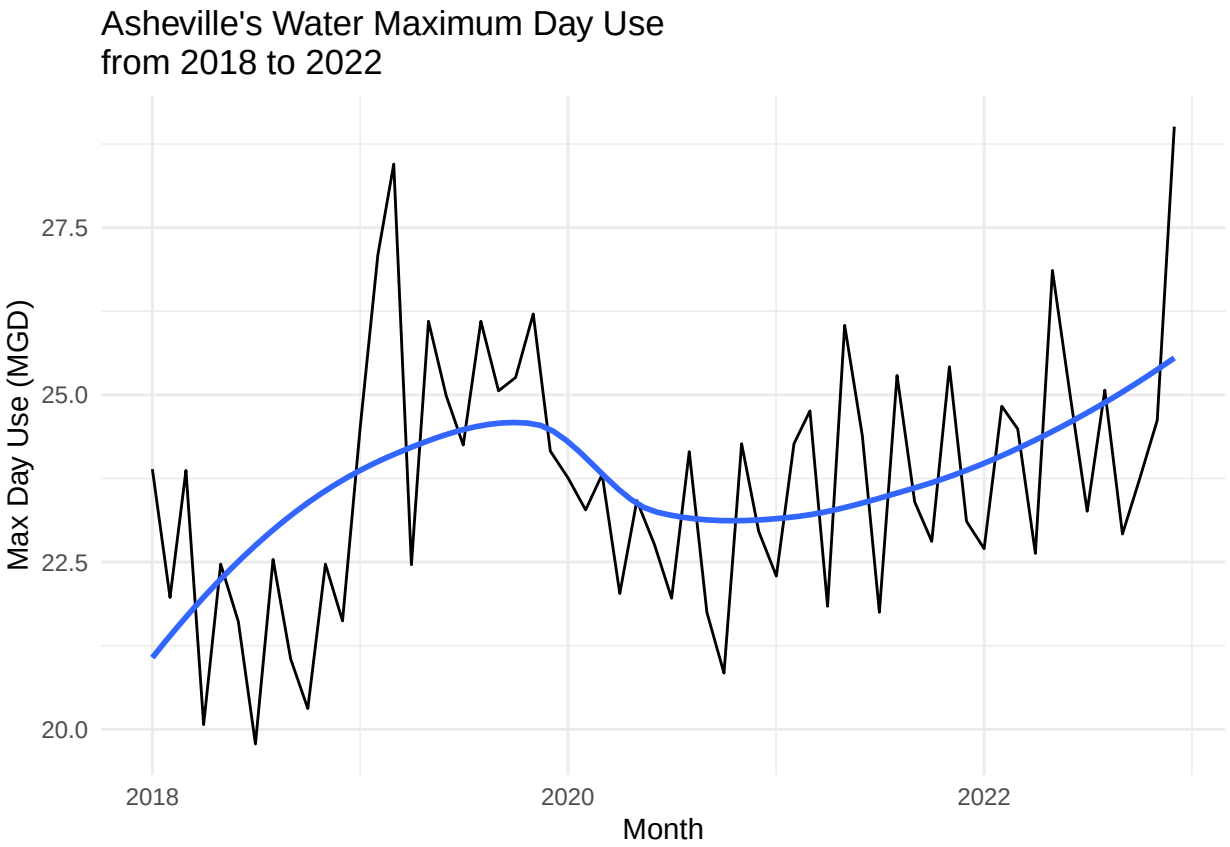
```
#9
asheville_2018 = scrape.it('01-11-010', 2018)
asheville_2019 = scrape.it('01-11-010', 2019)
asheville_2020 = scrape.it('01-11-010', 2020)
asheville_2021 = scrape.it('01-11-010', 2021)
asheville_2022 = scrape.it('01-11-010', 2022)

combine1 = rbind(asheville_2018, asheville_2019, asheville_2020,
                 asheville_2021, asheville_2022)
combine1
```

```
##    Month Year Max_Day_Use Water_System_Name     PWSID    Ownership       Date
## 1      1 2018       23.89         Asheville 01-11-010 Municipality 2018-01-01
## 2      2 2018       21.97         Asheville 01-11-010 Municipality 2018-02-01
## 3      3 2018       23.87         Asheville 01-11-010 Municipality 2018-03-01
## 4      4 2018       20.07         Asheville 01-11-010 Municipality 2018-04-01
## 5      5 2018       22.47         Asheville 01-11-010 Municipality 2018-05-01
## 6      6 2018       21.61         Asheville 01-11-010 Municipality 2018-06-01
## 7      7 2018       19.78         Asheville 01-11-010 Municipality 2018-07-01
## 8      8 2018       22.54         Asheville 01-11-010 Municipality 2018-08-01
## 9      9 2018       21.05         Asheville 01-11-010 Municipality 2018-09-01
## 10    10 2018       20.31         Asheville 01-11-010 Municipality 2018-10-01
## 11    11 2018       22.47         Asheville 01-11-010 Municipality 2018-11-01
## 12    12 2018       21.62         Asheville 01-11-010 Municipality 2018-12-01
## 13     1 2019       24.51         Asheville 01-11-010 Municipality 2019-01-01
## 14     2 2019       27.09         Asheville 01-11-010 Municipality 2019-02-01
## 15     3 2019       28.45         Asheville 01-11-010 Municipality 2019-03-01
## 16     4 2019       22.46         Asheville 01-11-010 Municipality 2019-04-01
## 17     5 2019       26.10         Asheville 01-11-010 Municipality 2019-05-01
## 18     6 2019       24.99         Asheville 01-11-010 Municipality 2019-06-01
## 19     7 2019       24.25         Asheville 01-11-010 Municipality 2019-07-01
## 20     8 2019       26.10         Asheville 01-11-010 Municipality 2019-08-01
## 21     9 2019       25.06         Asheville 01-11-010 Municipality 2019-09-01
## 22    10 2019       25.26         Asheville 01-11-010 Municipality 2019-10-01
## 23    11 2019       26.21         Asheville 01-11-010 Municipality 2019-11-01
## 24    12 2019       24.16         Asheville 01-11-010 Municipality 2019-12-01
## 25     1 2020       23.76         Asheville 01-11-010 Municipality 2020-01-01
## 26     2 2020       23.28         Asheville 01-11-010 Municipality 2020-02-01
## 27     3 2020       23.81         Asheville 01-11-010 Municipality 2020-03-01
## 28     4 2020       22.03         Asheville 01-11-010 Municipality 2020-04-01
## 29     5 2020       23.42         Asheville 01-11-010 Municipality 2020-05-01
## 30     6 2020       22.76         Asheville 01-11-010 Municipality 2020-06-01
## 31     7 2020       21.96         Asheville 01-11-010 Municipality 2020-07-01
## 32     8 2020       24.15         Asheville 01-11-010 Municipality 2020-08-01
## 33     9 2020       21.75         Asheville 01-11-010 Municipality 2020-09-01
## 34    10 2020       20.84         Asheville 01-11-010 Municipality 2020-10-01
## 35    11 2020       24.27         Asheville 01-11-010 Municipality 2020-11-01
## 36    12 2020       22.96         Asheville 01-11-010 Municipality 2020-12-01
## 37     1 2021       22.29         Asheville 01-11-010 Municipality 2021-01-01
## 38     2 2021       24.27         Asheville 01-11-010 Municipality 2021-02-01
```

```
## 39     3 2021       24.76         Asheville 01-11-010 Municipality 2021-03-01
## 40     4 2021       21.84         Asheville 01-11-010 Municipality 2021-04-01
## 41     5 2021       26.04         Asheville 01-11-010 Municipality 2021-05-01
## 42     6 2021       24.39         Asheville 01-11-010 Municipality 2021-06-01
## 43     7 2021       21.75         Asheville 01-11-010 Municipality 2021-07-01
## 44     8 2021       25.29         Asheville 01-11-010 Municipality 2021-08-01
## 45     9 2021       23.40         Asheville 01-11-010 Municipality 2021-09-01
## 46    10 2021       22.81         Asheville 01-11-010 Municipality 2021-10-01
## 47    11 2021       25.42         Asheville 01-11-010 Municipality 2021-11-01
## 48    12 2021       23.11         Asheville 01-11-010 Municipality 2021-12-01
## 49     1 2022       22.70         Asheville 01-11-010 Municipality 2022-01-01
## 50     2 2022       24.83         Asheville 01-11-010 Municipality 2022-02-01
## 51     3 2022       24.49         Asheville 01-11-010 Municipality 2022-03-01
## 52     4 2022       22.63         Asheville 01-11-010 Municipality 2022-04-01
## 53     5 2022       26.86         Asheville 01-11-010 Municipality 2022-05-01
## 54     6 2022       25.00         Asheville 01-11-010 Municipality 2022-06-01
## 55     7 2022       23.26         Asheville 01-11-010 Municipality 2022-07-01
## 56     8 2022       25.07         Asheville 01-11-010 Municipality 2022-08-01
## 57     9 2022       22.92         Asheville 01-11-010 Municipality 2022-09-01
## 58    10 2022       23.74         Asheville 01-11-010 Municipality 2022-10-01
## 59    11 2022       24.62         Asheville 01-11-010 Municipality 2022-11-01
## 60    12 2022       29.01         Asheville 01-11-010 Municipality 2022-12-01
```

```
ggplot(combine1, aes(x = Date, y = Max_Day_Use)) +
  geom_line() +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Asheville's Water Maximum Day Use\nfrom 2018 to 2022",
    x = "Month",
    y = "Max Day Use (MGD)") +
  theme_minimal()
```

```
## `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: the plot suggests an overall upward trend in Asheville's water usage from 2018 to 2022, as indicated by the rising smoothed line.