Assignment 09: Data Scraping

David Amanfu

Total points:

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_09_Data_Scraping.Rmd") prior to submission.

Set up

1. Set up your session:

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

```
#1a.
getwd()
```

```
## [1] "/Users/davidamanfu/Desktop/Duke MPP/Environ Data /Environmental_Data_Analytics_2022/Assignments
```

```
knitr::opts_knit$set(root.dir = "~/Desktop/Duke MPP/Environ Data /Environmental_Data_Analytics_2022/")
```

```
#1b
# install.packages("rvest")
# install.packages("dataRetrieval")
# install.packages("tidycensus")
library(agricolae)
library(corrplot)
library(cowplot)
library(dataRetrieval)
```

```
## Warning: package 'dataRetrieval' was built under R version 4.0.5
```

```
library(extrafont)
library(extrafontdb)
library(ggpubr)
library(ggthemes)
library(hrbrthemes)
library(Kendall)
library(leaflet)
```

```
## Warning: package 'leaflet' was built under R version 4.0.5
```

```
library(lubridate)
library(mapview)
library(rvest)
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.0.5
```

```
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.0.5
```

```
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
library(trend)
library(tseries)
library(viridis)
library(zoo)

#Disable on-the-fly projections
sf::sf_use_s2(FALSE)
#Fix Mapview
mapviewOptions(fgb = FALSE)

AmanfuTheme2 <- theme_ipsum()+
  theme(legend.position = "bottom",
        legend.key = element_rect(fill = "white", colour = "black"),legend.direction = "horizontal",
        legend.title = element_text(face = "bold"))
theme_set(AmanfuTheme2)
```

2. We will be scraping data from the NC DEQs Local Water Supply Planning website, specifically the Durham's 2019 Municipal Local Water Supply Plan (LWSP):

- Navigate to https://www.ncwater.org/WUDC/app/LWSP/search.php
- Change the date from 2020 to 2019 in the upper right corner.
- 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=2020

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

```
#2
# https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020
Durham_LWSP <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2017')
Durham_LWSP
```

```
## {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
- PSWID
- Ownership
- From the “3. Water Supply Sources” section:
- Average Daily Use (MGD) - for each month

In the code chunk below scrape these values, assigning them to three 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, with the first value being 36.0100.

```
#3
water.system.name <- Durham_LWSP %>% html_nodes('table:nth-child(7) tr:nth-child(1) td:nth-child(2)') %
pwsid <- Durham_LWSP %>% html_nodes('tr:nth-child(1) > td:nth-child(5)') %>% html_text()
ownership <- Durham_LWSP %>% html_nodes('table:nth-child(7) tr:nth-child(2) td:nth-child(4)') %>% html_

mgd.key <- ':nth-child(32) td:nth-child(9) , :nth-child(32) td:nth-child(6) tr:nth-child(2) :nth-child(
max.withdrawals.mgd <- Durham_LWSP %>% html_nodes(mgd.key) %>% html_text()

#These failed trials are from using the selector gadget in Safari and Firefox:
# ':nth-child(32) td:nth-child(6) , td:nth-child(9), :nth-child(32) td:nth-child(6), :nth-child(32) td:
#  'tr:nth-child(2) td:nth-child(9) , :nth-child(32) td:nth-child(6), :nth-child(32) td:nth-child(3), :
# ':nth-child(31) td:nth-child(9) , tr:nth-child(4) :nth-child(9) tr:nth-child(3) tr:nth-child(2) :nth-
```

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.

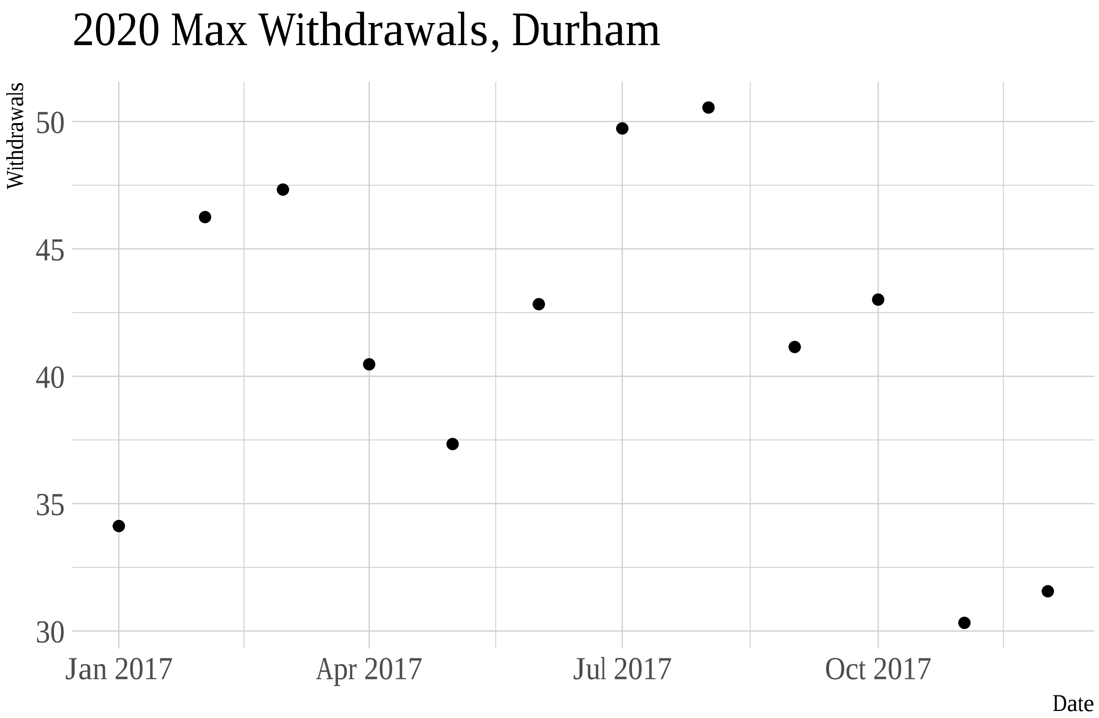
NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc…

5. Plot the max daily withdrawals across the months for 2020

```
#4

scrapetest <- data.frame("Year"=rep("2017",12),
                         "Month"=c("Jan","May","Sep","Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug","De
                         "System"=rep(water.system.name,12),
                         "PWSID"=rep(pwsid,12),
                         "Ownership"=rep(ownership,12),
                         "Withdrawals"=as.double(max.withdrawals.mgd)) %>% mutate("yearchar" =ym(paste0
#5
scrapetest
```

```
##    Year Month System     PWSID    Ownership Withdrawals   yearchar
## 1  2017   Jan Durham 03-32-010 Municipality       34.12 2017-01-01
## 2  2017   May Durham 03-32-010 Municipality       37.34 2017-05-01
## 3  2017   Sep Durham 03-32-010 Municipality       41.15 2017-09-01
## 4  2017   Feb Durham 03-32-010 Municipality       46.25 2017-02-01
## 5  2017   Jun Durham 03-32-010 Municipality       42.83 2017-06-01
## 6  2017   Oct Durham 03-32-010 Municipality       43.01 2017-10-01
## 7  2017   Mar Durham 03-32-010 Municipality       47.33 2017-03-01
## 8  2017   Jul Durham 03-32-010 Municipality       49.73 2017-07-01
## 9  2017   Nov Durham 03-32-010 Municipality       30.32 2017-11-01
## 10 2017   Apr Durham 03-32-010 Municipality       40.47 2017-04-01
## 11 2017   Aug Durham 03-32-010 Municipality       50.55 2017-08-01
## 12 2017   Dec Durham 03-32-010 Municipality       31.56 2017-12-01
```

```
durham2020 <- ggplot(scrapetest,aes(x=yearchar))+geom_point(aes(y=Withdrawals))+labs(title="2020 Max Wi
durham2020
```



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 scraped**.

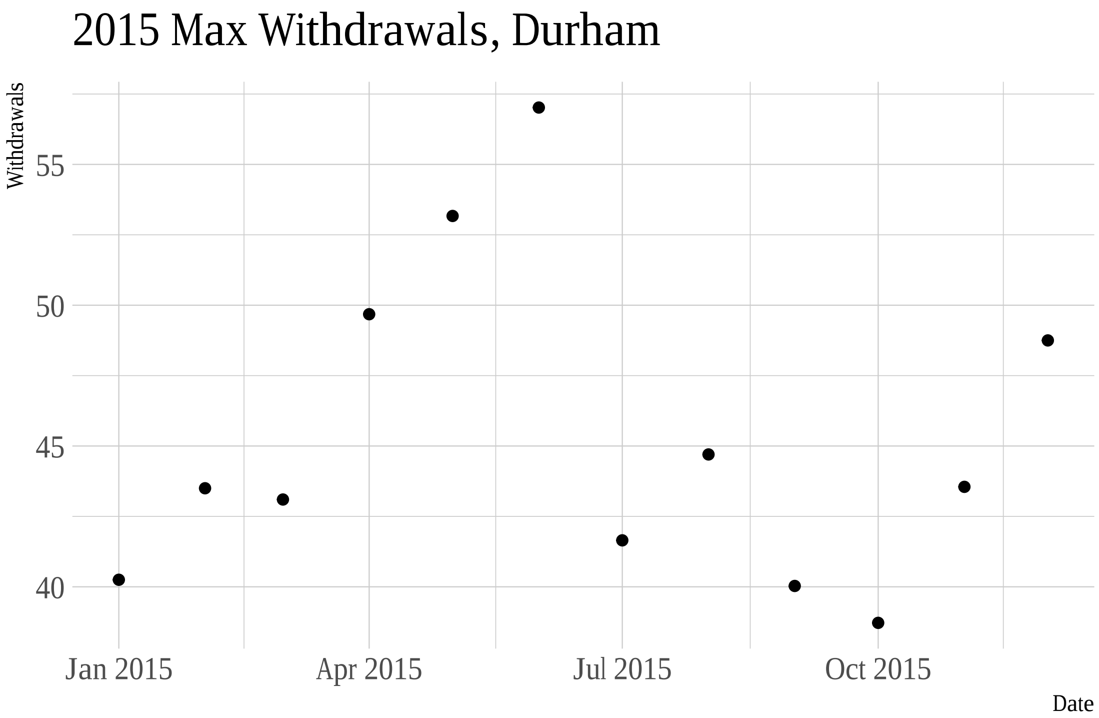
```
#6.

the_facility <- '03-32-010'
the_year <- 2015

scrape.it <- function(the_year, the_facility){

  #Retrieve the website contents
  the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid='
  the_scrape_url <- paste0(the_base_url, the_facility, '&year=', the_year)
  the_website <- read_html(the_scrape_url)

  #Set the element address variables (determined in the previous step)
  water.system.name_node <- 'table:nth-child(7) tr:nth-child(1) td:nth-child(2)'
  pwsid_node <- 'tr:nth-child(1) > td:nth-child(5)'
  ownership_node <- 'table:nth-child(7) tr:nth-child(2) td:nth-child(4)'
  max.withdrawals.mgd_node <- 'th~ td+ td'
    #':nth-child(32) td:nth-child(9) , :nth-child(32) td:nth-child(6) tr:nth-child(2) :nth-child(9), :n
    #':nth-child(31) td:nth-child(9) , tr:nth-child(4) :nth-child(9) tr:nth-child(3) tr:nth-child(2) :n
  #Scrape the data items
  water.system.name <- the_website %>% html_nodes(water.system.name_node) %>% html_text()
  pwsid <- the_website %>%   html_nodes(pwsid_node) %>%  html_text()
  ownership <- the_website %>% html_nodes(ownership_node) %>% html_text()
  max.withdrawals.mgd <- the_website %>% html_nodes(max.withdrawals.mgd_node) %>% html_text()

  #Construct a dataframe from the scraped data
  df_withdrawals <- data.frame("Year"=rep(the_year,12),
                           "Month"=c("Jan","May","Sep","Feb","Jun","Oct","Mar","Jul","Nov","Apr","Aug",
                           "System"=rep(water.system.name,12),
                           "PWSID"=rep(pwsid,12),
                           "Ownership"=rep(ownership,12),
                           "Withdrawals"=as.double(max.withdrawals.mgd)) %>%
                      mutate("yearchar" =ym(paste0(Year,"-",Month)))

  #Pause for a moment - scraping etiquette
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

  #Return the dataframe
  return(df_withdrawals)
}
```

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

```
#7
scrapetest2015 <-scrape.it(2015,'03-32-010')
scrapetest2015
```

```
##    Year Month System     PWSID    Ownership Withdrawals   yearchar
## 1  2015   Jan Durham 03-32-010 Municipality       40.25 2015-01-01
## 2  2015   May Durham 03-32-010 Municipality       53.17 2015-05-01
## 3  2015   Sep Durham 03-32-010 Municipality       40.03 2015-09-01
## 4  2015   Feb Durham 03-32-010 Municipality       43.50 2015-02-01
## 5  2015   Jun Durham 03-32-010 Municipality       57.02 2015-06-01
## 6  2015   Oct Durham 03-32-010 Municipality       38.72 2015-10-01
## 7  2015   Mar Durham 03-32-010 Municipality       43.10 2015-03-01
## 8  2015   Jul Durham 03-32-010 Municipality       41.65 2015-07-01
## 9  2015   Nov Durham 03-32-010 Municipality       43.55 2015-11-01
## 10 2015   Apr Durham 03-32-010 Municipality       49.68 2015-04-01
## 11 2015   Aug Durham 03-32-010 Municipality       44.70 2015-08-01
## 12 2015   Dec Durham 03-32-010 Municipality       48.75 2015-12-01
```

```
durham2015 <- ggplot(scrapetest2015,aes(x=yearchar))+geom_point(aes(y=Withdrawals))+labs(title="2015 Ma
durham2015
```

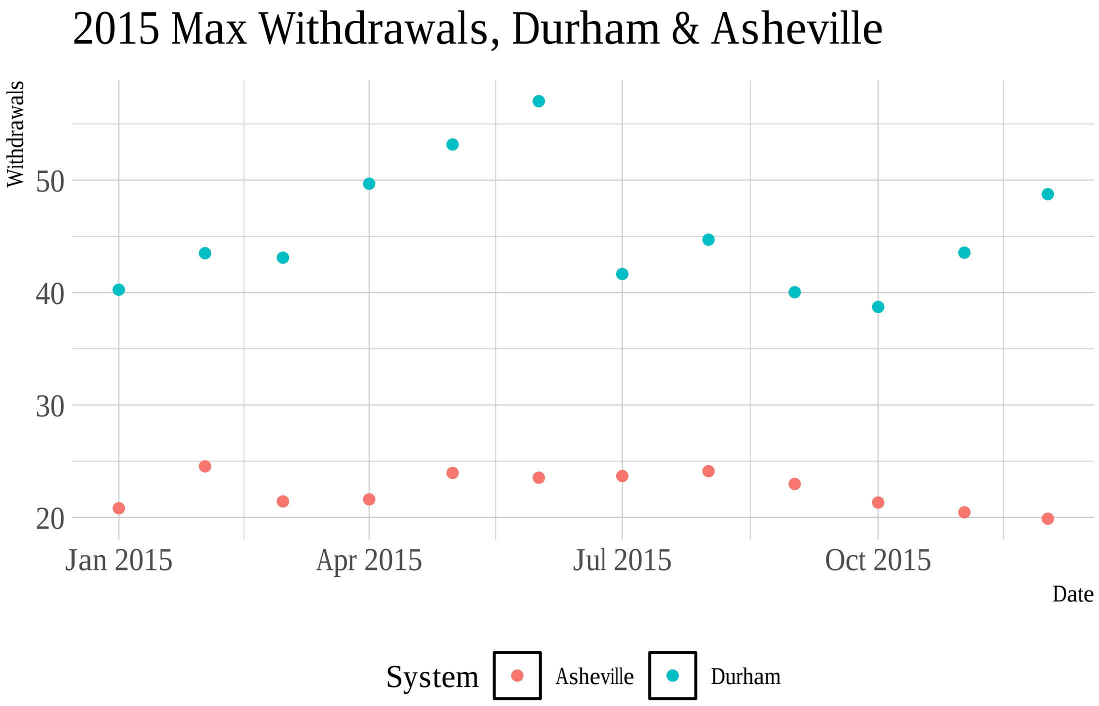


8. Use the function above to extract data for Asheville (PWSID = 01-11-010) in 2015. Combine this data with the Durham data collected above and create a plot that compares the Asheville to Durham's water withdrawals.

```
#8
Asheville2015 <-scrape.it(2015,'01-11-010')
DurhAshe <- union(scrapetest2015,Asheville2015)
DurhAshe
```

```
##    Year Month    System     PWSID    Ownership Withdrawals   yearchar
## 1  2015   Jan    Durham 03-32-010 Municipality       40.25 2015-01-01
## 2  2015   May    Durham 03-32-010 Municipality       53.17 2015-05-01
## 3  2015   Sep    Durham 03-32-010 Municipality       40.03 2015-09-01
## 4  2015   Feb    Durham 03-32-010 Municipality       43.50 2015-02-01
## 5  2015   Jun    Durham 03-32-010 Municipality       57.02 2015-06-01
## 6  2015   Oct    Durham 03-32-010 Municipality       38.72 2015-10-01
## 7  2015   Mar    Durham 03-32-010 Municipality       43.10 2015-03-01
## 8  2015   Jul    Durham 03-32-010 Municipality       41.65 2015-07-01
## 9  2015   Nov    Durham 03-32-010 Municipality       43.55 2015-11-01
## 10 2015   Apr    Durham 03-32-010 Municipality       49.68 2015-04-01
## 11 2015   Aug    Durham 03-32-010 Municipality       44.70 2015-08-01
## 12 2015   Dec    Durham 03-32-010 Municipality       48.75 2015-12-01
## 13 2015   Jan Asheville 01-11-010 Municipality       20.81 2015-01-01
## 14 2015   May Asheville 01-11-010 Municipality       23.95 2015-05-01
## 15 2015   Sep Asheville 01-11-010 Municipality       22.97 2015-09-01
## 16 2015   Feb Asheville 01-11-010 Municipality       24.54 2015-02-01
## 17 2015   Jun Asheville 01-11-010 Municipality       23.53 2015-06-01
## 18 2015   Oct Asheville 01-11-010 Municipality       21.32 2015-10-01
## 19 2015   Mar Asheville 01-11-010 Municipality       21.42 2015-03-01
## 20 2015   Jul Asheville 01-11-010 Municipality       23.68 2015-07-01
## 21 2015   Nov Asheville 01-11-010 Municipality       20.45 2015-11-01
## 22 2015   Apr Asheville 01-11-010 Municipality       21.60 2015-04-01
## 23 2015   Aug Asheville 01-11-010 Municipality       24.11 2015-08-01
## 24 2015   Dec Asheville 01-11-010 Municipality       19.88 2015-12-01
```

```
DurhAsheville2015 <- ggplot(DurhAshe,aes(x=yearchar,y=Withdrawals))+geom_point(aes(color=System))+labs(
DurhAsheville2015
```



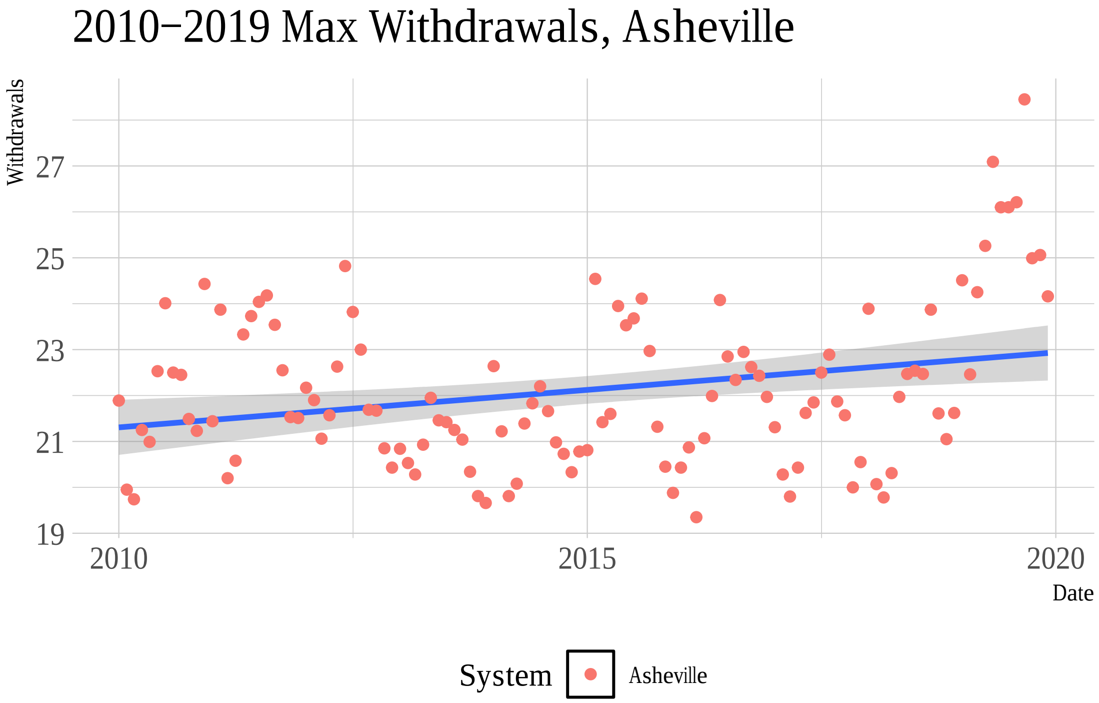
9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2019.Add a smoothed line to the plot.

```
#9
yeargap <- rep(2010:2019)
yeargap
```

```
##  [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
```

```
our_facility <- '01-11-010'
the_dfs <- lapply(X = yeargap,
                    FUN = scrape.it,
                    the_facility=our_facility)
Asheville20102019 <- bind_rows(the_dfs)
#Asheville20102019
Asheville1019 <- ggplot(Asheville20102019,aes(x=yearchar,y=Withdrawals))+
                    geom_smooth(method=lm)+geom_point(aes(color=System))+labs(title="2010-2019 Max Withdra
Asheville1019
```

```
## `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? It does! It appears that it is pretty constant from 2010 through 2018, and then we see a marked difference in water usage, given the increase starting in about late 2019.