Assignment 09: Data Scraping

Enikoe Bihari

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

```
#1

# check working directory
getwd()
```

```
## [1] "C:/Users/eniko/Documents/Duuuuuke/2021-22/Data Analytics/Environmental_Data_Analytics_2022/Assig
```

```
# import libraries
library(tidyverse)
# install.packages("rvest")
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.1.3
```

```
library(lubridate)
# install.packages("cowplot")
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.1.3
```

```
# install.packages("gridExtra")
library("gridExtra")
```

```
## Warning: package 'gridExtra' was built under R version 4.1.3
```

```r
# install.packages("grid")
library("grid")
library(ggplot2)

# create a theme with gray defaults
theme1 <- theme_gray(base_size = 12) +
  theme(axis.text = element_text(color = "grey50"),
        legend.position = "top",
        axis.title = element_text(color = "grey20"),
        legend.key.width = unit(2, "cm"))

# set it as the default theme
theme_set(theme1)
```

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 2021 to 2020 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.)

```r
#2

# create a webpage object from the url
Durham_LWSP_page <- read_html('https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=202
Durham_LWSP_page
```

```
## {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:

- Max 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.

```r
#3

# get water system name from the webage as text
water.system.name <- Durham_LWSP_page %>%
  html_nodes("div+ table tr:nth-child(1) td:nth-child(2)") %>%
  html_text()
water.system.name
```

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

```
# get PWSID
pwsid <- Durham_LWSP_page %>%
  html_nodes("td tr:nth-child(1) td:nth-child(5)") %>%
  html_text()
pwsid
```

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

```
# get ownership
ownership <- Durham_LWSP_page %>%
  html_nodes("div+ table tr:nth-child(2) td:nth-child(4)") %>%
  html_text()
ownership
```

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

```
# get max withdrawals
max.withdrawals.mgd <- Durham_LWSP_page %>%
  html_nodes("th~ td+ td") %>%
  html_text()
max.withdrawals.mgd
```

```
##  [1] "36.0100" "36.9800" "41.6900" "32.0500" "40.6100" "40.5600" "37.2900"
##  [8] "43.6300" "33.3200" "32.3700" "41.9300" "28.0600"
```

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.

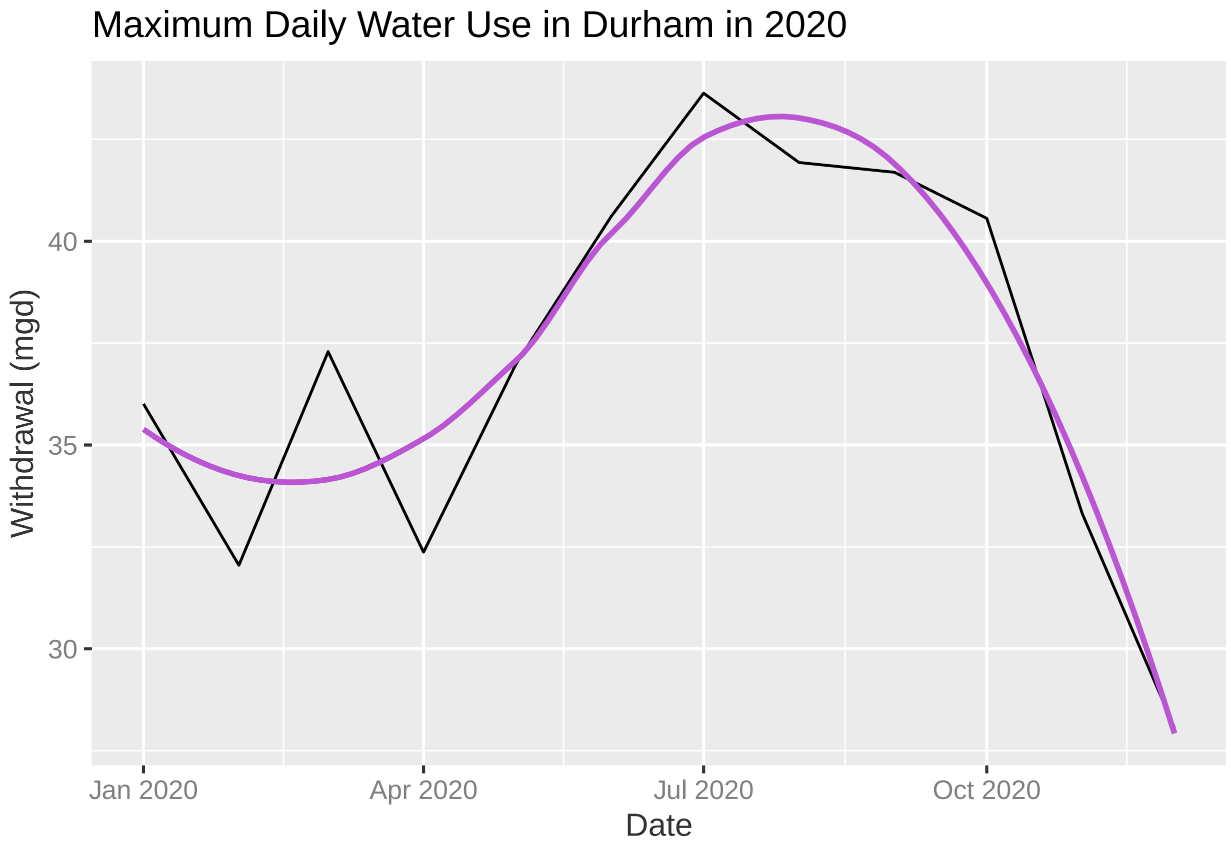
   NOTE: It's likely you won't be able to scrape the monthly widthrawal data in order. You can overcome this by creating a month column in the same order the data are scraped: Jan, May, Sept, Feb, etc…

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

```
#4

# create a dataframe
df_mgd <- data.frame("month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                     "year" = rep(2020,12),
                     "max.withdrawals.mgd" = as.numeric(max.withdrawals.mgd))

# put in other attributes and the date
df_mgd <- df_mgd %>%
  mutate(water.system.name = !!water.system.name,
         pwsid = !!pwsid,
         ownership = !!ownership,
         date = my(paste(month,"-",year)))

#5
```

```
# plot the data
ggplot(df_mgd, aes(x=date, y=max.withdrawals.mgd)) +
  geom_line() +
  geom_smooth(method="loess", se=FALSE, color = "mediumorchid") +
  labs(title = "Maximum Daily Water Use in Durham in 2020",
       y="Withdrawal (mgd)",
       x="Date")
```

```
## '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. **Be sure to modify the code to reflect the year and site scraped**.

```
#6.

# define all the inputs to the function
the_base_url <- 'https://www.ncwater.org/WUDC/app/LWSP/report.php?'
the_PWSID <- '03-32-010'
the_year <- 2020
the_scrape_url <- paste0(the_base_url, "pwsid=", the_PWSID, "&year=", the_year)
print(the_scrape_url)
```

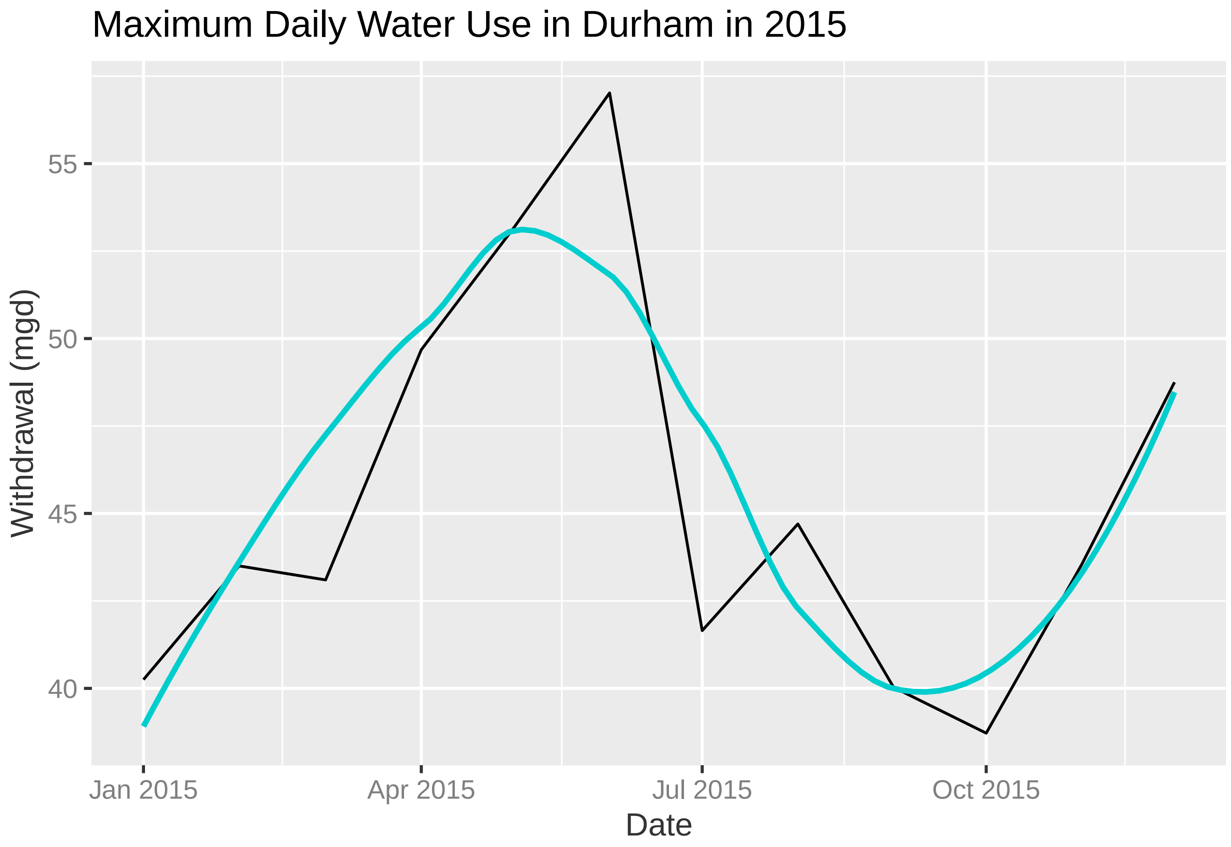
```
## [1] "https://www.ncwater.org/WUDC/app/LWSP/report.php?pwsid=03-32-010&year=2020"
```

```
# define the function
scrape.mgd <- function(the_year, the_PWSID, the_base_url){
  # access the website
  the_website <- read_html(paste0(the_base_url, "pwsid=", the_PWSID, "&year=", the_year))
  # print(the_website)

  # Set the element address variables
  the_watersytemname_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_mgd_tag <- ’th~ td+ td’

  # scrape the data items
  the_watersystem <- the_website %>% html_nodes(the_watersytemname_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()
  max_mgd <- the_website %>% html_nodes(the_mgd_tag) %>% html_text()

  # create a dataframe
  the_df <- data.frame("month" = c(1,5,9,2,6,10,3,7,11,4,8,12),
                       "year" = rep(the_year,12),
                       "max.withdrawals.mgd" = as.numeric(max_mgd)) %>%
    mutate(water.system.name = !!the_watersystem,
           pwsid = !!the_pwsid,
           ownership = !!the_ownership,
           date = my(paste(month,"-",year)))

  # pause
  #Sys.sleep(1) #uncomment this if you are doing bulk scraping!

  # return the dataframe
  return(the_df)
}
```

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

```
#7

# define all the new inputs to the function
the_PWSID_durham <- ’03-32-010’
the_year_2015 <- 2015

# run the function
df_mgd_durham_2015 = scrape.mgd(the_year_2015, the_PWSID_durham, the_base_url)

# plot the data
ggplot(df_mgd_durham_2015, aes(x=date, y=max.withdrawals.mgd)) +
  geom_line() +
  geom_smooth(method="loess", se=FALSE, color = "cyan3") +
  labs(title = "Maximum Daily Water Use in Durham in 2015",
       y="Withdrawal (mgd)",
       x="Date")
```

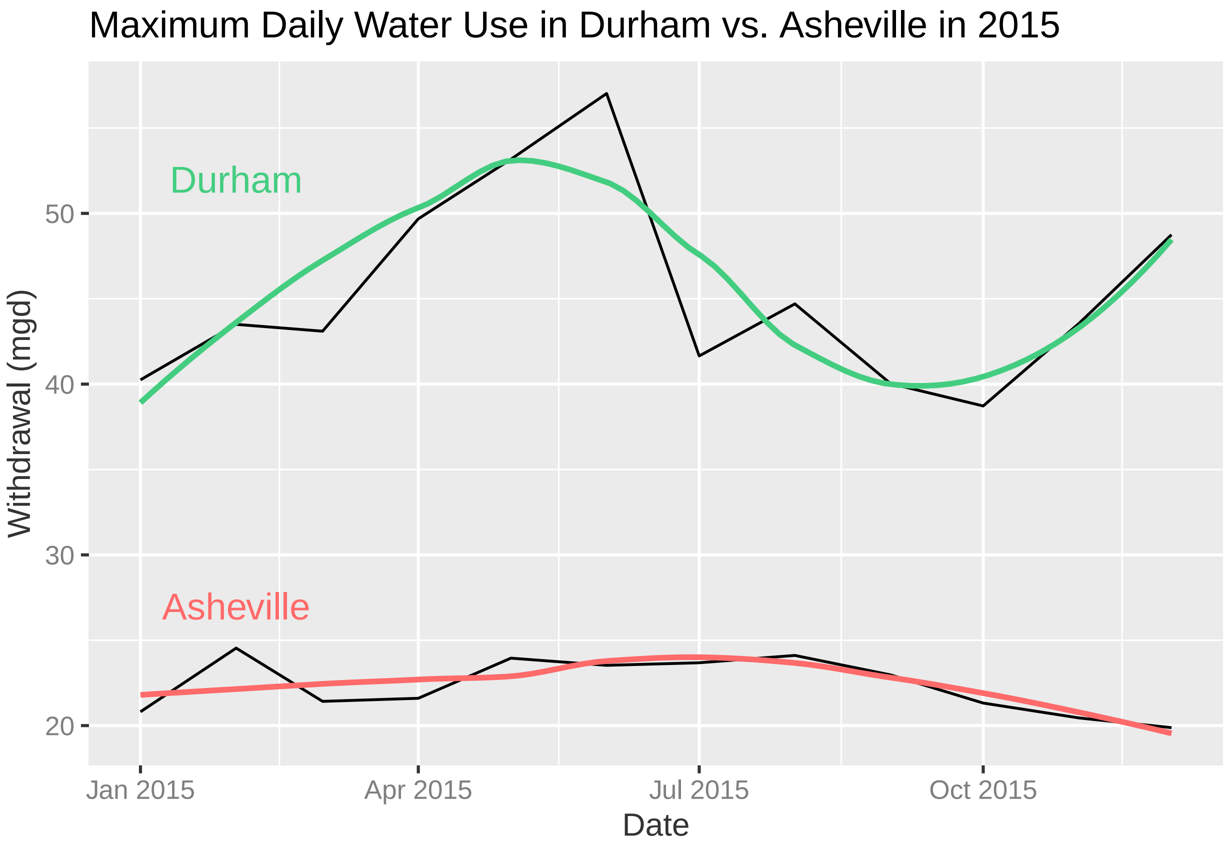
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



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 the Asheville to Durham's water withdrawals.

```
#8

# define all the new inputs to the function
the_PWSID_asheville <- '01-11-010'
the_year_2015 <- 2015

# run the function
df_mgd_asheville_2015 = scrape.mgd(the_year_2015, the_PWSID_asheville, the_base_url)

# plot the data
ggplot() +
  geom_line(data = df_mgd_durham_2015,
            aes(x=date, y=max.withdrawals.mgd)) +
  geom_smooth(data = df_mgd_durham_2015,
              aes(x=date, y=max.withdrawals.mgd),
              method="loess",
              se=FALSE,
              color = "seagreen3") +
  geom_line(data = df_mgd_asheville_2015, aes(x=date, y=max.withdrawals.mgd)) +
  geom_smooth(data = df_mgd_asheville_2015,
              aes(x=date, y=max.withdrawals.mgd),
              method="loess",
              se=FALSE,
               color = "indianred1") +
  labs(title = "Maximum Daily Water Use in Durham vs. Asheville in 2015",
       y="Withdrawal (mgd)",
       x="Date") +
  annotate("text",
           x = as.Date("2/1/2015", format = "%m/%d/%Y"),
           y = 52,
           label = "Durham",
           color = "seagreen3",
           cex = 5) +
  annotate("text",
           x = as.Date("2/1/2015", format = "%m/%d/%Y"),
           y = 27,
           label = "Asheville",
           color = "indianred1",
           cex = 5)
```

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

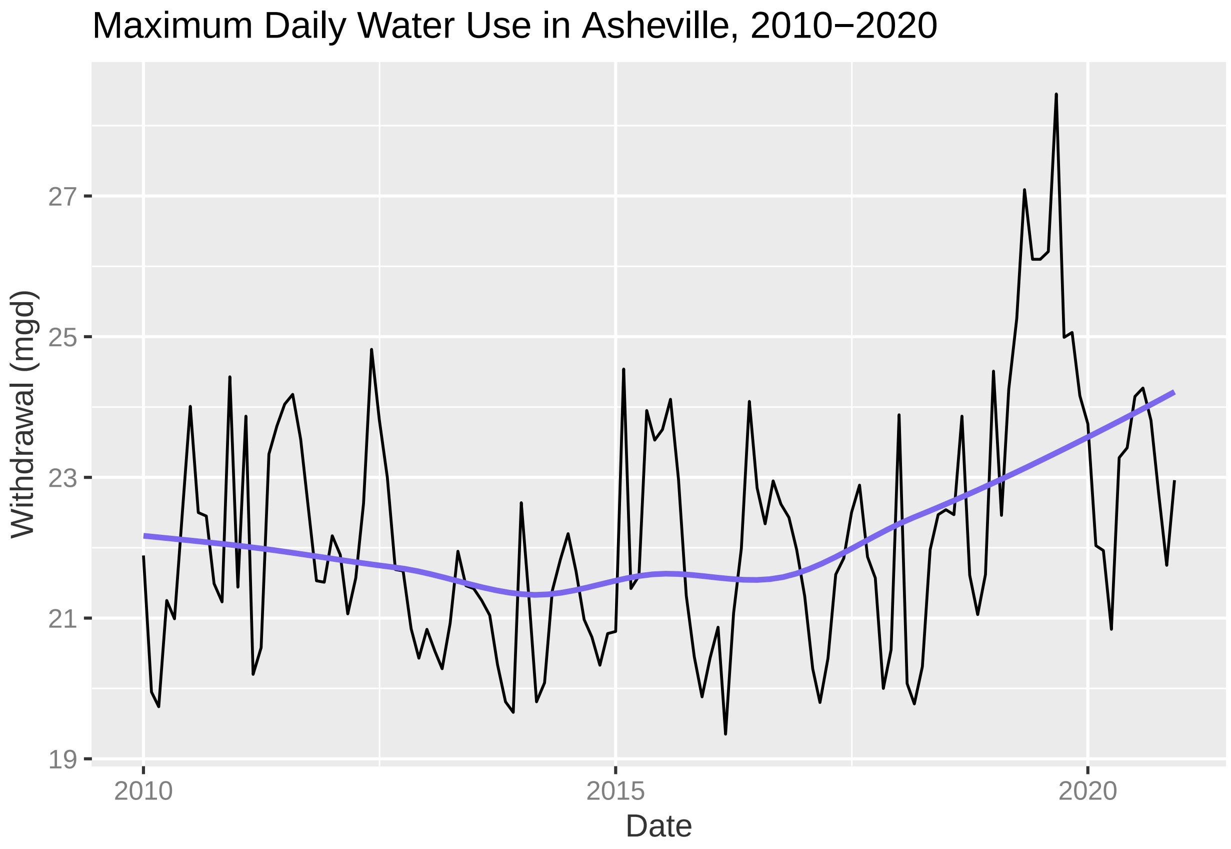


9. Use the code & function you created above to plot Asheville's max daily withdrawal by months for the years 2010 thru 2020.Add a smoothed line to the plot.

```
#9

# define all the new inputs to the function
the_PWSID_asheville <- '01-11-010'
```

```
the_years_2010.2019 <- rep(2010:2020)

# run the function with lapply
the_dfs_asheville <- lapply(X = the_years_2010.2019,
                  FUN = scrape.mgd,
                  the_PWSID=the_PWSID_asheville,
                  the_base_url = the_base_url)

# put into a single dataframe
the_df_asheville <- bind_rows(the_dfs_asheville)

# plot the data
ggplot(the_df_asheville, aes(x=date, y=max.withdrawals.mgd)) +
  geom_line() +
  geom_smooth(method="loess", se=FALSE, color = "slateblue2") +
  labs(title = "Maximum Daily Water Use in Asheville, 2010-2020",
       y="Withdrawal (mgd)",
       x="Date")
```

```
## ‘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?

Water usage in Asheville seemed to dip around 2015, but has been increasing since then (with 2020 water usage levels being higher than they were in 2010 before the dip).