Scrape a Website I

Grant Esparza

October, 16 2018

Match links

```
### Incomplete web links
get_request <- GET("https://aqs.epa.gov/aqsweb/airdata/download_files.html#AQI")
html <- content(get_request, "text")
```

```
## No encoding supplied: defaulting to UTF-8.
```

```
incomplete_links <- as.data.frame(
                      str_match_all(
                        html, pattern="<a href=\"daily.*county_[0-9]{4}\\.zip\">"
                      ) ## str_match_all
                    ) ## as.data.frame

## Split
incomp_links <- as.data.frame(str_split_fixed(incomplete_links[,1], pattern="\"", 3))
colnames(incomp_links) <- c("href", "link", "end")

## Add home url
complete_links <- paste("https://aqs.epa.gov/aqsweb/airdata/", incomp_links$link, sep="")
```

Using a regular expression to grab county data, I can collect all the links of interest. I then stored these completed links into a vector.

Add data to folder

```
## Add data
for (link in complete_links[4:14]) {
  get_request <- GET(link)
  bin_data <- content(get_request, "raw")
  writeBin(bin_data, "daily-county-aqi")
  unzip(zipfile="daily-county-aqi", exdir="aqi")
}
```

A for loop lets me increment over each link of interest and unzip them into a folder located in my local directory. I plan on using this AQI data with my wildfire data so I only need information as recent as 2015.

Manipulate data

```
files <- list.files(path="aqi", full.names=TRUE)
aqi.dat <- rbindlist(lapply(files, fread))

## How many rows?
nrow(aqi.dat)
```

```
## [1] 3524244
```

I ran into trouble with the speed of loading all of my csv files. Luckily I remembered the package `data.table` from Datafest last year. The function `fread()` maps the files into memory prior to actually reading the file. This makes the subsequent loading much faster. Wheras my previous soulution took about 45 seconds to complete, `fread()` takes about 2 seconds. I used `list.files()` to list all of the files stored in my AQI folder. The function `rbindlist()` allows me to combine all of my datasets into one large table of data.