## M10 Assignment: Create web-scraper to load csv file into S3 Bucket (100 pts.)

In this lab, we will use Selenium to scrape a website and store the file in an s3 bucket.

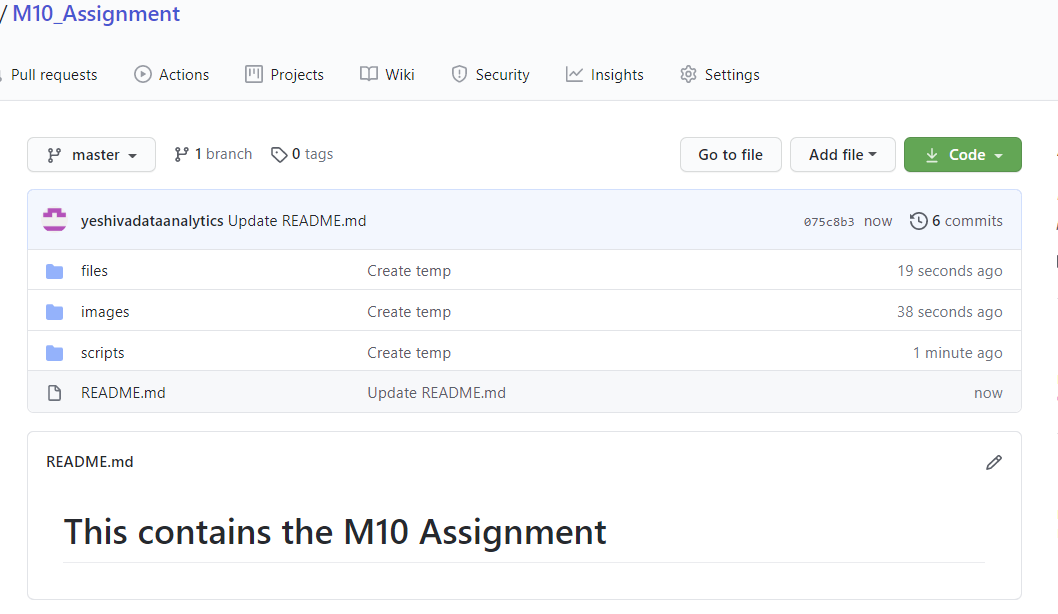
## Getting Started

Note that there are dependencies. You will need to configure and install awscli and boto3 and be sure you give access to an S3 bucket. You can find documentation [here](https://boto3.amazonaws.com/v1/documentation/api/latest/guide/configuration.html). Additional modules required include time, pandas, and selenium. You will also need to ensure that you have the [Google Chrome webdriver](https://chromedriver.chromium.org/downloads) installed. See [here](http://jonathansoma.com/lede/foundations-2018/classes/selenium/selenium-windows-install/) for further instructions.

The final submission will be an output of your code to a GitHub repository that you have created. You will add me as a collaborator to the repository and put the file, the scripts, and an image of the s3 bucket you created to completed the assignment with the file enclosed.

Therefore, you should set up a repository called “M10\_Assignment” and inside you should create the following files and folders. You can recreate the repository below or fork your own [here](https://github.com/yeshivadataanalytics/M10_Assignment).

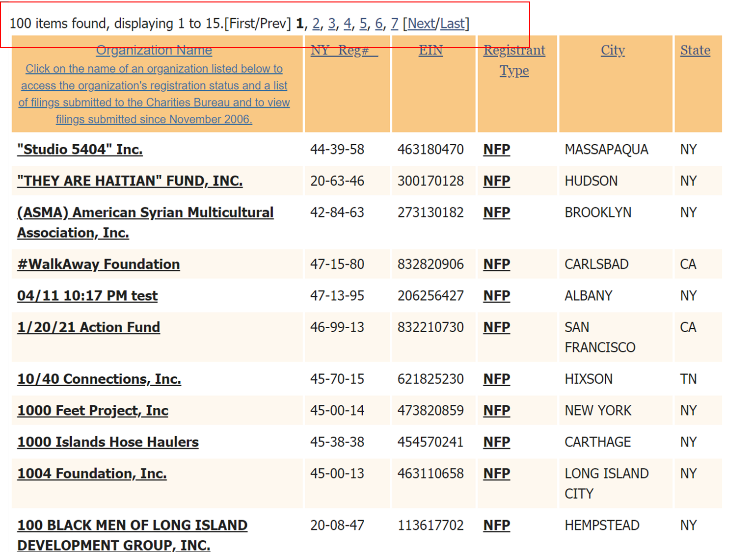
1. images:
   1. This will contain any relevant images from your assignment
2. files:
   1. This will also store the same file (csv) that you created from the scraper. You can simply download it from your s3 bucket and uploaded it to the repository in this location.
3. scripts
   1. This will contain your final script in Jupyter format (.ipybn)
4. A README.MD file
   1. You can use this to explain what you created and share the images of the output file
   2. You should also describe the dependencies to run your script, its contents and goals.



## Part I. Create web-scraper to load csv file into S3 Bucket

We will be creating a web scraper to parse a table from the [Charities Bureau Website](https://www.charitiesnys.com/RegistrySearch/search_charities.jsp). From the website: *“All charitable organizations operating in New York State are required by law to register and file annual financial reports with the Attorney General's Office. This includes any organization that conducts charitable activities, holds property that is used for charitable purposes, or solicits financial or other contributions.”*

Running a simple search in the search box will redirect you to a basic table that is in HTML and can be parsed. This table is what we will scrape. Explore and become familiar with the site, noting the variables used to move through the site.



The site will contain vendor information and you are being tasked with culling a list into a csv file by scraping the website using the selenium module in Python.

### Complete the following steps:

* **Step 1**. Load the script, **M10\_webscraper\_assignment.ipybn,** into your repository as a separate file
* **Step 2**. Create an s3 bucket to store your outputs of csv files. Note: I used a bucket named ‘database-update-bucket’. Be sure to make the settings on the s3 buckets public. As part of configuration of the awscli, you should ensure that you have access to s3 and the AWS environment. Note see below in the code snipper where you will have to enter the name of your s3 bucketL



* **Step 3**. Update the **M10\_webscraper\_assignment.ipybn** file to include your s3 path
* **Step 4**. Test the file by inspecting the output on the s3 bucket
* **Step 5**. Once the test has passed, commit the changes to the master branch of your GitHub repository
* **Step 6.** Note that the output of the file has an issue! The header inserts a blank row! Update the script that was provided to remove the blank row in the csv output file
* **Step 7.** Test the file by inspecting the output on the s3 bucket
* **Step 8.** Once the test has passed, commit the changes to your script the master branch of your GitHub repository
* **Step 9.** Provide evidence that the file and script were successfully updated by providing the link to s3 and the .ipybn file in the repository.



* **Step 10.** Be sure to update your READ\_ME.md file to reflect the script actions and ensure proper documentation.

## Part II. Update web-scraper to iterate all results and load csv file into S3 Bucket

For this portion of the lab, you are going to alter the script, **M10\_webscraper\_assignment.ipybn***,* to iterate through the pagination that exists on the page and compile all of the results in one dataframe from each page. Once you have done that, you should follow the same procedures in (2) to load the full csv file into the s3 bucket.

Once the test has passed, commit the changes to the master branch of your GitHub repository, updating the script and the file on s3.

## Submitting the Assignment.

Your result should contain an updated .ipybn file that iterates through the table on the Charities Bureau Website and downloads the data into a clean CSV file (no skipped header). This file should be loaded directly into the s3 bucket you created.

Add your instructor as a collaborator and upload the link to your repository as a response to this Assignment.