Motivation:

Obtaining data can be a simple task of selecting well-organized files or an exercise that makes you suffer from poor organization and documentation. One remedy to this that is commonly supported by organizations to handle the acquisition of bulk data without the common pitfalls of a user interface is a scraping tool. This post will document how to use my tool to scrape data from AirNow's air quality network using the latest version of python, python 3.6.

AirNow is a source for air quality data that supplies data on an international scale with the goal of communicating air quality to all individuals. The organization supplies a variety of products such as forecast maps and historical air quality data for U.S. affiliated locations worldwide. These products are ideal for private, public, and academic institutions. This dataset will be useful to those looking for those looking for reference sensor observations for comparison or to draw conclusions about air quality in a region.

Method:

To start - modules need to be imported and the directory (file name you want to use) needs to be specified.

```
1.  import os
2.  from datetime import datetime
3.  from os.path import expanduser
4.  import pandas as pd
5.  import googlemaps
6.  from urllib.request import urlopen
7.
8.  os.chdir('c:\\Users\\tadams15\\Desktop\\Wrapper')
```

At this point – if you're not registered you should do so to continue. You can request an AirNow API account at this link: https://docs.airnowapi.org/login?index=

You'll also need to get a google maps API key to continue using this code's functionality! To do this first go to the following link: https://console.cloud.google.com/apis/credentials?pli=1 Using your personal google account, you'll need to follow the link and generate an API key on the credentials page of the APIs & Services tab of the google cloud console. More documentation can be found here: https://github.com/googlemaps/google-maps-services-python.

First – we want to use the python package "googlemaps" that we imported to automatically detect the bounding box of any city and give us the latitude and longitude coordinates that we could provide to the AirNow API. For this – you'll need to use the API key you retrieved in the prior step.

```
1.  gmaps = googlemaps.Client(key = '')
2.  geocode_result = gmaps.geocode('Boston, MA')
3.  bbox = str(geocode_result[0]['geometry']['bounds']['southwest']['lng']) + ',' +str(geoc
    ode_result[0]['geometry']['bounds']['southwest']['lat']) + ',' + str(geocode_result[0][
```

```
    'geometry']['bounds']['northeast']['lng'])+','+str(geocode_result[0]['geometry']['bound
    s']['northeast']['lat'])
```

After we have specified the bounding box for a city, we can move forward to specify the variables we look to request. These variables are presented in more detail in the table below. Keep in mind you will likely need to tailor these variables to pull data for your specific date range and species of interest. For more information on the content you can scrape using this API – the following link is valuable: https://docs.airnowapi.org/

```
1.  options = {} # Empty Dict
2.  options["url"] = "https://airnowapi.org/aq/data/"
3.  options["start_date"] = "2020-06-01"
4.  options["start_hour_utc"] = "1"
5.  options["end_date"] = "2020-09-10"
6.  options["end_hour_utc"] = "23"
7.  options["parameters"] = "no2,o3,pm25,pm10,so2" # These are all possible species
8.  options["bbox"] = bbox # Change for each site
9.  options["data_type"] = "a"
10. options["format"] = "csv" # Choose CSV/TXT/GIS whatever you need
11. options["ext"] = "kml"
12. options["api_key"] = "" # Change this to your own API Key - You have to get this from A
    irNow's API site
13. options["verbose"] = "1"
14. options["includerawconcentrations"] = "1"
```

The options specified are described with detail below:

"url"
The site that you're scraping from. We are pulling data from AirNow.
"start_date"
The first date that you want to collect observations
"start_hour_utc":
In universal time, the hour of day that you would like to initiate observations.
"end_date"
The final date you'd like to collect observations from.
"end_hour_utc"
In universal time, the hour of day that you would like to terminate observation.
"parameters"
The trace gases you're interested in observing. All possible species in AirNow are listed above.
"bbox"

This is your bounding box, a box of latitude and longitude – within which there can be one or more AirNow sensors. It will collect the parameters from the sensors in this box.
"data_type"
This specifies whether you want the AQI index or concentrations listed.
"format"
Specifies what file format you want your data in.
"verbose"
Gives additional site information if set to True (1).
"Includerawconcentrations"
When set to True (1) gives raw concentrations of species is given in the same unit specified.
"api_key"
Your personal API key.

Once you have specified what specific variables that you're interested in, you can go ahead and request these variables from the AirNow API using the following code. I you add any varibles, please add these to the **REQUEST_URL** variable.

```
REQUEST_URL = options["url"] \
              + "?startdate=" + options["start_date"] \
              + "t" + options["start_hour_utc"] \
              + "&enddate=" + options["end_date"] \
              + "t" + options["end_hour_utc"] \
              + "¶meters=" + options["parameters"] \
              + "&bbox=" + options["bbox"] \
              + "&datatype=" + options["data_type"] \
              + "&format=" + options["format"] \
              + "&verbose=" + options["verbose"] \
              + "&includerawconcentrations=" + options["includerawconcentrations"] \
              + "&api_key=" + options["api_key"]
```

Next, you should specify what you would like to name your file and the location that you would like to download this file to.

```
home_dir = expanduser("~")

download_file_name = "AirNowAPI" + datetime.now().strftime("_%Y%M%d%H%M%S." + options["
ext"])
download_file = os.path.join(home_dir, download_file_name) # Joins the API with your ho
me directory
```

Once specified you need to open, write to the file, and close this file.

```
File = open('test.csv','w')

response = urlopen(REQUEST_URL).read().decode('utf-8') # reads the API data
File.write(response) # Writes the API data to your file
File.close()
```

This code will download data you requested. One thing to keep in mind is that this data does not come with column labels. To add column labels import the file you just saved with python's pandas module and specify that there is no header, apply labels to the data, and re-save the file for a well-documented csv.

```
df = pd.read_csv('test.csv',header = None)
df.columns = ['Lat','Lon','Date UTC','Species','Concentration',
              'Raw Concentration','Unit','Site Name','Agency Owner',
              'AQS ID','Full AQS ID']
df.to_csv('test.csv')
```

With this file that I named "test.csv" you would be able to easily save AirNow data from as many sites as you would like within your bounding box and manipulate the data with ease.

Having issues with your code? Be sure to check the simple things first.

- Make sure you're not trying to download over 1 year of data. If you try to retrieve over a year of data you may get an error when using "urlopen" on your request URL.
- Make sure your "googlemaps" API key and AirNow API key are correct and that your "googlemaps" API key doesn't have any warnings next to it.
- Check your dates. Is your start date actually before your end date?
- If your end date is too recent – the data might not be in the AirNow database yet. For more recent data https://openaq.org/#/ might be a better resource.