**Challenge and Approach:**

1. **Challenge:**

* Given a list of different universities around the United States without the address, longitude, and latitude, the challenge is to generate the longitude and latitude from the list above.

1. **Approach:**

* I came up with two different solutions:
* One is to use Google Scripting to create a Google Sheets function that will receive the university’s name and return the corresponding longitude and latitude.
* Second is to use a python built-in function “Selenium” that will automatically call up the google and copy the longitude coordinates to the python list.

**Solution 1: Google Scripting**

1. **Prerequisites:**

* It would be best if you are already familiar with Google Scripting and Javascript. However, you can still understand this if you have no prior knowledge in these two fields.

1. **Solution:**

* With the google scripting, the user can create their own Google Sheets function by writing the Javascript code.
* In this situation, I use Google Maps Service built-in function to automatically receive the given name of the location. Then, it will generate the corresponding longitude and latitude if the name is available on the map.
* For the code of the solution, I use the code from [Brecht Vermeire](https://discourse.looker.com/u/brecht) to generate the latitude and longitude coordinates. In his answer, he also mentions a similar method that uses Yandex Maps Service.

function GEOCODE\_GOOGLE(address) {

if (address.map) {

return address.map(GEOCODE\_GOOGLE)

} else {

var r = Maps.newGeocoder().geocode(address)

for (var i = 0; i < r.results.length; i++) {

var res = r.results[i]

return res.geometry.location.lat + ", " + res.geometry.location.lng

}

}

}

* Source: <https://discourse.looker.com/t/get-latitude-longitude-for-any-location-through-google-sheets-and-plot-these-in-looker/5402>

**Solution 2: Python - Selenium**

1. **Prerequisites:**

* For this solution, it requires more in Python coding experience. You have to know the basic data structures, control types, and string in python.

1. **Solutions:**

* Selenium is a more general web scraping tool than google scripting. The main idea is to call a temporary google chrome and get access to the google maps sites. Then the code will automatically return the corresponding longitude and latitude.

1. Import the necessary packages.

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

import time

from webdriver\_manager.chrome import ChromeDriverManager

1. The code will import the csv file by using Python pandas.

import pandas as pd

#Import and read the files

df = pd.read\_csv(r'D:\ITAP\Summer Intern\Generate\_Lat\_Long\university\_location\_data\_original.csv')

1. The code will convert the ‘College’ column and convert it into a ‘College’ list. Then, the code will drop some unnecessary null values.

#Clean the unnecessary data in this file

z = []

m = df['College'].tolist()

for i in range(len(m)):

if str(m[i]) == "nan":

z.append(i)

df.drop(z, inplace=True)

1. Use WebDriver to autonomously access google chrome and google map

driver = webdriver.Chrome(ChromeDriverManager().install())

driver.get('https://www.google.com/maps/')

1. After that, I use to loop to transfer the University name to my Google Chrome and it will take the new url and separate them to the longitude and latitude.

latitude = []

longtitude = []

for i in m:

lat = []

lon = []

search = driver.find\_element\_by\_id('searchboxinput')

search.send\_keys(i)

search.send\_keys(Keys.ENTER)

time.sleep(5)

t = list(driver.current\_url)

for j in range(len(t)):

if t[j] == '@':

j += 1

while(t[j] != ','):

lat.append(t[j])

j += 1

j += 1

while(t[j] != ','):

lon.append(t[j])

j += 1

break

lat = "".join(lat)

lon = "".join(lon)

latitude.append(lat)

longtitude.append(lon)

search.send\_keys(Keys.CONTROL, 'a')

search.send\_keys(Keys.BACKSPACE)

1. However, the longitude does not completely match with the location’s longitude. Therefore, I have to correct it by + 0.00219

longtitude = []

for i in lon:

m = float(i) + 0.00219

longtitude.append(m)

1. Finally, I append the new column to the csv file and export the new file

df['Latitude'] = lat

df['Longtitude2'] = longtitude

df.to\_csv(r'D:\ITAP\Summer Intern\Generate\_Lat\_Long\university\_location\_data\_updated\_2.csv', index=False)

* **Comparison of Selenium and Google Scripting:**

|  |  |
| --- | --- |
| Selenium | Google Scripting |
| * Require a lot of coding experience | * Require a little or no coding experience |
| * Take a long time to run the entire code | * Run really fast |
| * Free to have unlimited access | * Paid to have some features |
| * Have no day limit | * Have day limit |
| * Applicable to multiple web scraping | * Limited to Google only |
|  |  |