# Web Scraping for Address Identification

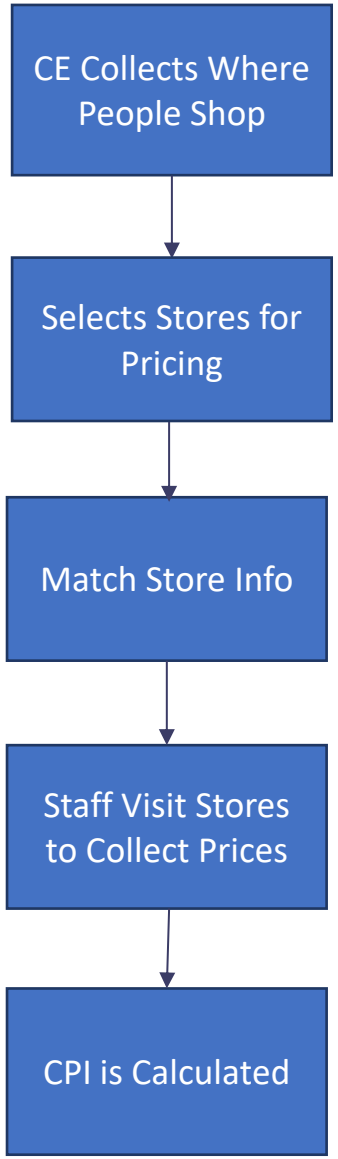
**Darius Stansil**
**CIF Data Science Fellow**
**Costweights**
**Supervisor: Greg Barbieri**

August 17th 2022

# Background

The CPI is a widely utilized index published by the Bureau of Labor Statistics that describes the change of prices of consumer goods.

- Prices are predominately collected from retail establishments
- Establishments reported in Consumer Expenditure surveys (CE) by participants
- Field staff visit a sample of establishments directly to collect prices



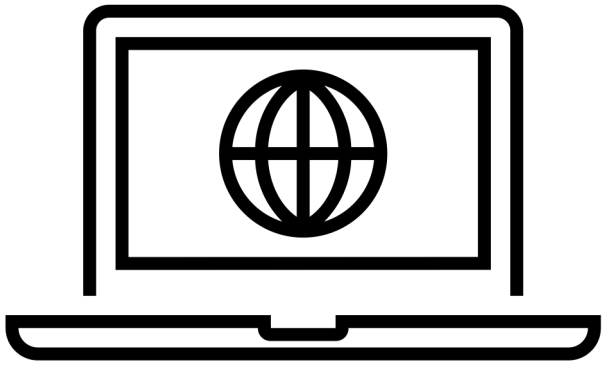
# Motivation

- CE data does not always contain store locations
- At most, city and state is provided
- A complete and accurate address is required for price collection
- Decrease the number of inaccurate or missing addresses

Currently these problems are handled by the Sample Maintenance System (SMS), which matches CE data to business census data to determine addresses for price collection. **The main motivation for this project was to explore other solutions.**

# Web Scraping as a Solution

- City and state is enough detail to search for stores
- Online resources will have up-to-date addresses
- Removes the need for manual refinement
- Search tools are flexible to user input
- Increases efficiency through automation



# My Approach

- Google is difficult to web scrape, and API data cannot be saved
- Building a general web scraper is intractable
- APIs and maintained address databases exist, but are either incomplete or have costs attached

# My Approach: The Application

- 17 python scripts, one for each store, combined into a python package
- ~20% of CPI sample is represented
- Developed in Jupyter
- Selenium
  - Simulates a user interacting with a page
  - Eases the burden of varied site layouts
- Few dependencies for flexibility

# My Approach: Demo

- Given a store name, city, and state, it returns all valid addresses in that location
- Respondent confidentiality: can’t demo the stores represented in the sample
- Created scripts for 2 stores at random from top 100 retailers
- Jupyter notebook for demonstration purposes

# Further Considerations

- Increasing the amount of scraping scripts in the application decreases the ‘miss’ rate for address matches
- Utilize google search to handle changing URLs
- Web API
- Refine Selenium use to improve optimization
- Robust testing required to verify accuracy

# Contact Information

**Darius Stansil**
stansil.joseph@bls.gov
jdstansil@gmail.com

Thanks to Greg Barbieri, Anya Stockburger and Brandon Kopp