**Hypothesis**: Top locations with the highest paying jobs and the lowest cost of living. What are the unemployment rates for the top results?

# Data Extraction and Transformation

## Web Scrape

**Action**: We performed a web scrape for the top 100 careers.

**Challenge**: The website blocked us from scraping the table we needed.

**Resolution**: we learned to override this by right clicking the webpage, selecting “Save As” and downloading the page to a local drive.

**Source**: <https://www.careerprofiles.info/top-100-careers.html>

Graphical user interface, application

Description automatically generated

## Clean CSV file

**Action**: We downloaded realty data from Kaggle.

**Challenge**: We needed to clean the data for select columns and then filter within those columns for specific results.

**Resolution**: After cleaning the data by dropping NaN, replacing NaN in the “sold\_date” column with “Not Sold” and filtering for “for\_sale” houses and filtering for “Not Sold” we had a clean data set. We then exported the dataframe to CSV as the new csv resource.

Source: <https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset>

Graphical user interface, table

Description automatically generated

## Download from Web

**Action**: We downloaded data from Data World to get unemployment rates for occupation data.

**Challenge**: Trying to find the correct data or a large enough data set.

**Resolution**: Found on Data World and downloaded for use.

**Source**: <https://data.world/johnsnowlabs/us-unemployment-rates-change-over-the-year/workspace/file?filename=US+Unemployment+Rates+Change+Over+The+Year.csv.gz>

## Aggregating sources

**Action**: We imported our clean CSVs and created DataFrames

**Challenge**: Grouping and renaming columns, modifying the occupation names, and merging tables.

**Resolution**: After modifying code placement and updating pieces of code it ran through and we were able to connect to pgAdmin.

A picture containing text

Description automatically generated

# Loading into Postgres

**Action**: connecting to pgAdmin and loading our data to make viewable in Jupyter Notebook.

**Challenge**: We couldn’t show the tables in Jupyter Notebook.

**Resolution**: We needed to lowercase our table names, pgAdmin does not like upper case.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text

Description automatically generated Graphical user interface, text, application, email

Description automatically generated