Data Extraction from Redfin.com

In this project, we will look at how we can develop a program to download the dataset from redfin.com. For this we will rely on zip code database for the counties and use web scraping to extract housing information from redfin.com. To implement this solution, we will be using selenium, python requests and pandas library. Let's implement this project my downloading the zip code database. This project further assumes that you are familiar with using Jupyter notebooks and python.

Zip code Database

You can download all the zip codes in USA along with their state and counties information from here.

1. Install pandas library with the command

```
pip install pandas
```

2. Let's open the csv file to see the data available to us with pandas.

```
import pandas as pd
df = pd.read_csv('zip_code_database.csv')
df.tail(5)
```

	zip	type	decommissioned	primary_city	acceptable_cities	unacceptable_cities	state	county	timezone	area_codes	world_region	country
42719	99926	PO BOX	0	Metlakatla	NaN	NaN	AK	Prince of Wales-Outer Ketchikan Borough	America/Metlakatla	907	NaN	US
42720	99927	PO BOX	0	Point Baker	NaN	NaN	AK	Prince of Wales-Hyder Census Area	America/Sitka	907	NaN	US
42721	99928	PO BOX	0	Ward Cove	NaN	NaN	AK	Ketchikan Gateway Borough	America/Sitka	907	NaN	US
42722	99929	PO BOX	0	Wrangell	NaN	NaN	AK	Wrangell City and Borough	America/Sitka	907	NaN	US
42723	99950	PO BOX	0	Ketchikan	Edna Bay, Kasaan	NaN	AK	Prince of Wales-Outer Ketchikan Borough	America/Sitka	907	NaN	US

3. Let's look at what columns are available in this zip code database

```
df.columns
```

```
df.columns
```

```
Index(['zip', 'type', 'decommissioned', 'primary_city', 'acceptable_cities',
       'unacceptable_cities', 'state', 'county', 'timezone', 'area_codes',
       'world_region', 'country', 'latitude', 'longitude',
       'irs_estimated_population'],
      dtype='object')
```

4. For our problem, lets only focus on California counties. Extract all the available unique California counties from the zip code database

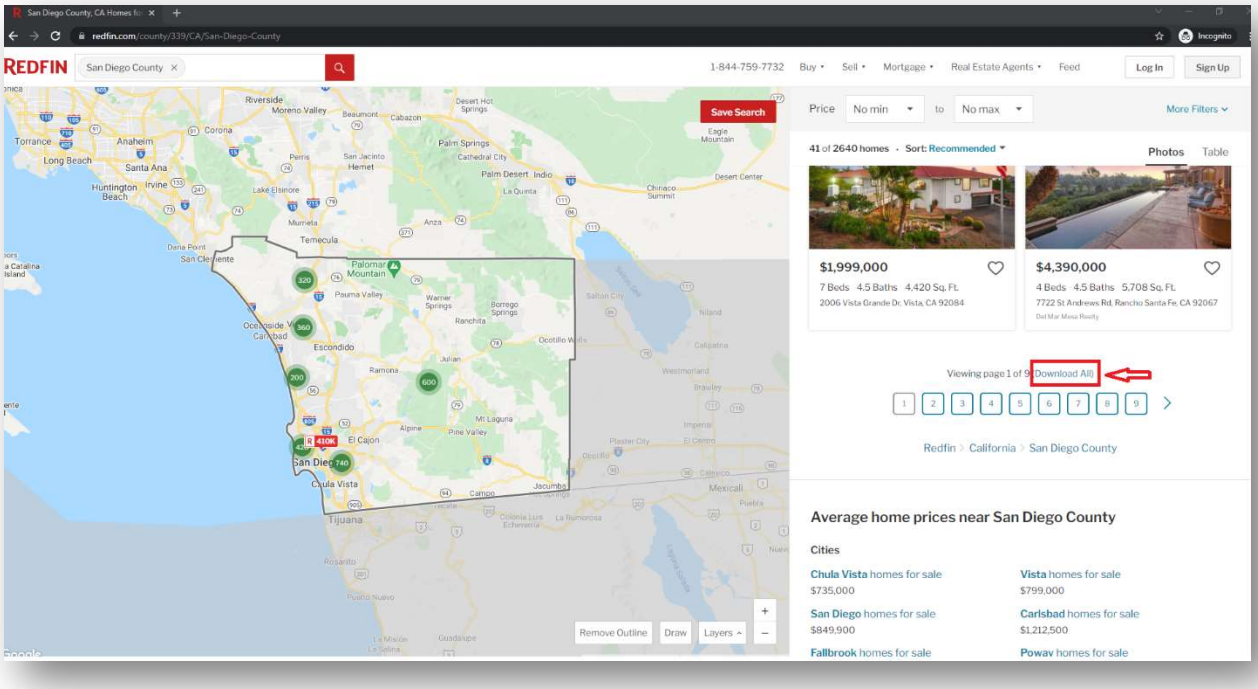
```
df = df[df['state']=='CA']
df = df[df['county'].notna()]
all_ca_counties = df['county'].unique()
all_ca_counties
```

```
df = df[df['state']=='CA']
df = df[df['county'].notna()]
all_ca_counties = df['county'].unique()
all_ca_counties
```

```
array(['Los Angeles County', 'Orange County', 'Ventura County',
       'San Bernardino County', 'Riverside County', 'San Diego County',
       'Imperial County', 'Inyo County', 'Santa Barbara County',
       'Tulare County', 'Kings County', 'Kern County', 'Fresno County',
       'San Luis Obispo County', 'Monterey County', 'Mono County',
       'Madera County', 'Merced County', 'Mariposa County',
       'San Mateo County', 'Santa Clara County', 'San Francisco County',
       'Sacramento County', 'Alameda County', 'Napa County',
       'Contra Costa County', 'Solano County', 'Marin County',
       'Sonoma County', 'Santa Cruz County', 'San Benito County',
       'San Joaquin County', 'Calaveras County', 'Tuolumne County',
       'Stanislaus County', 'Mendocino County', 'Lake County',
       'Humboldt County', 'Trinity County', 'Del Norte County',
       'Siskiyou County', 'Amador County', 'Placer County', 'Yolo County',
       'El Dorado County', 'Alpine County', 'Sutter County',
       'Yuba County', 'Nevada County', 'Sierra County', 'Colusa County',
       'Glenn County', 'Butte County', 'Plumas County', 'Shasta County',
       'Modoc County', 'Lassen County', 'Tehama County'], dtype=object)
```

Web scraping

Now that we have the counties we are interested in, we want to extract the housing information of these counties from redfin.com. If you manually open redfin.com and search for any of the county, redfin provides a link to download that counties housing information. The download link looks like https://www.redfin.com/stingray/api/gis-csv?al=1&isRentals=false&market=socal&min_stories=1&num_homes=350&ord=redfin-recommended-desc&page_number=1&region_id=339&region_type=5&sf=1,2,3,5,6,7&status=9&uipt=1,2,3,4,5,6,7,8&v=8 . We will be extracting this link for all the California counties if it is available.



1. We will be using selenium for this purpose. Install the following python libraries to get this data

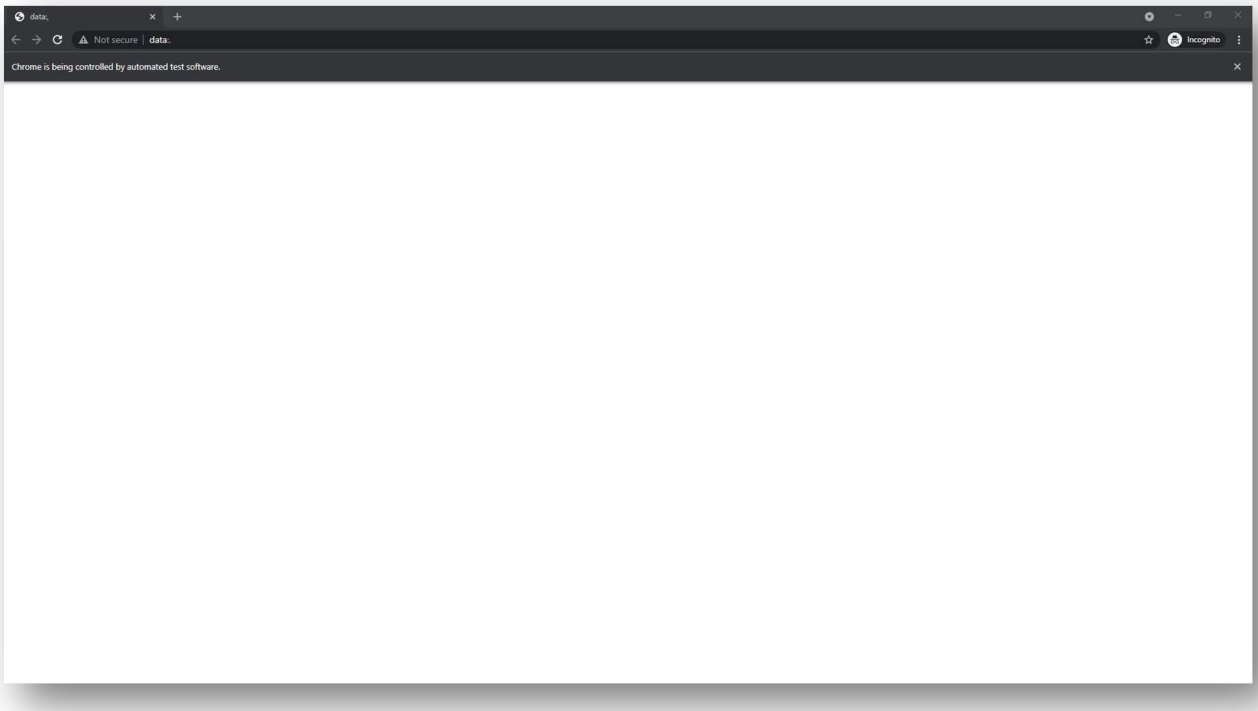
```
pip install selenium
pip install webdriver-manager
pip install requests
```

2. This step assumes you have chrome browser installed on your system. Let's open the browser with this step. This will launch an empty browser window.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
import requests
import io
import os
import glob

# Start the chrome browser
s=Service(ChromeDriverManager().install())
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument("--incognito")
driver = webdriver.Chrome(service=s, chrome_options=chrome_options)
driver.maximize_window()
```

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
import requests
import io
import os
import glob

# Start the chrome browser
s=Service(ChromeDriverManager().install())
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument("--incognito")
driver = webdriver.Chrome(service=s, chrome_options=chrome_options)
driver.maximize_window()
```

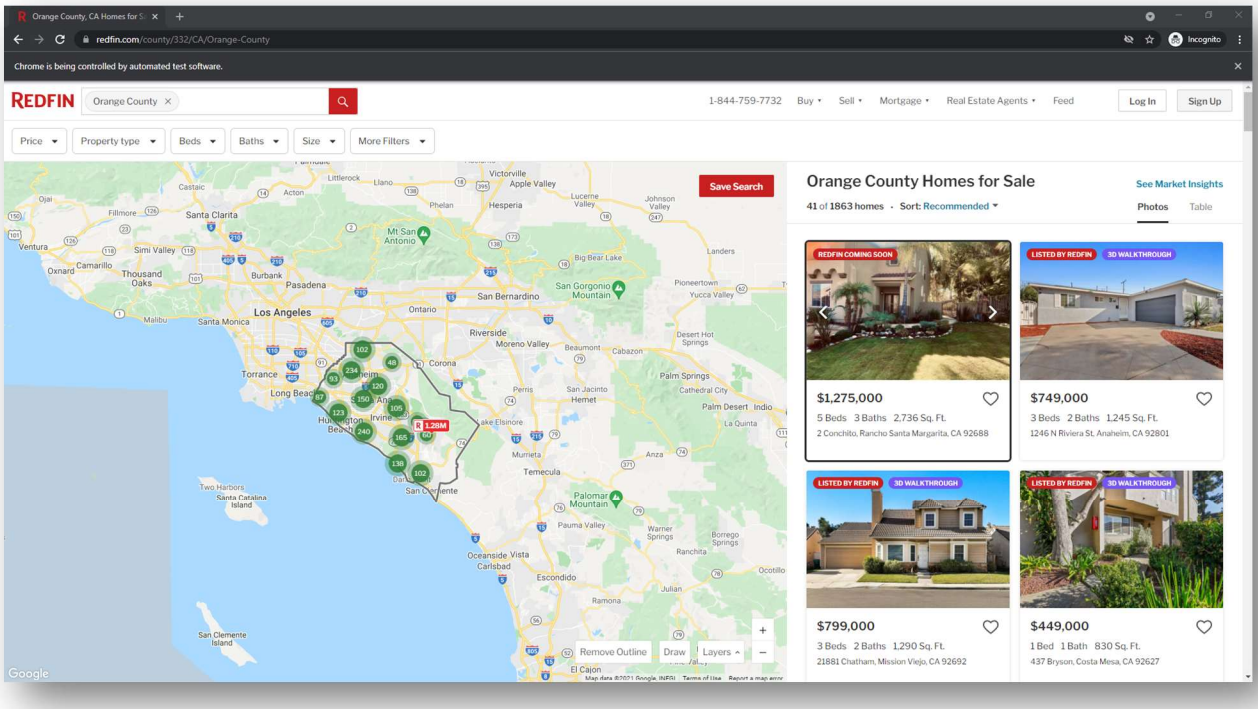
```
====== WebDriver manager ======
Current google-chrome version is 96.0.4664
Get LATEST chromedriver version for 96.0.4664 google-chrome
There is no [win32] chromedriver for browser  in cache
Trying to download new driver from https://chromedriver.storage.googleapis.com/96.0.4664.45/chromedriver_win32.zip
Driver has been saved in cache [C:\Users           wdm\drivers\chromedriver\win32\96.0.4664.45]
C:\Users          AppData\Local\Temp/ipykernel_74192/4085163408.py:16: DeprecationWarning: use options instead of chrome_option
s
  driver = webdriver.Chrome(service=s, chrome_options=chrome_options)
```



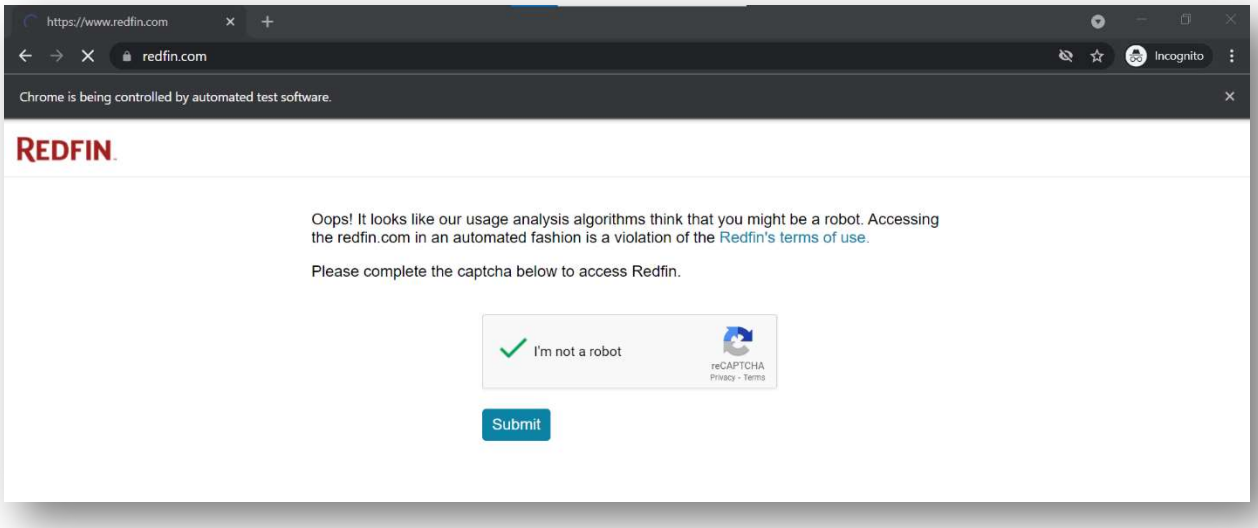
3. Now let's search for each of the California county and extract the download link. We will further update the download links to extract more housing information.

```
# This method will open https://www.redfin.com, search for the county name and return the download link
def searchByCounty(countyName):
    driver.get('https://www.redfin.com')
    time.sleep(15)
    try:
        driver.find_element(By.XPATH, "//input[contains(@id, 'search-box-input') and contains(@title, 'City, Address, School, Agent, ZIP')]").click()
        if driver.find_element(By.XPATH, "//input[contains(@title, 'Clear')]").is_displayed():
            driver.find_element(By.XPATH, "//input[contains(@title, 'Clear')]").click()
        driver.find_element(By.XPATH, "//input[contains(@id, 'search-box-input') and contains(@placeholder, 'City, Address, School, Agent, ZIP')]").send_keys(countyName)
        time.sleep(3)
        driver.find_element(By.XPATH, "//a[text()='" + countyName + "']").click()
        download_link = driver.find_element(By.XPATH, "//a[contains(@id, 'download-and-save')]").get_attribute("href")
        all_links = download_link.split('num_homes=350')
        print('Download link available for: '+countyName)
        return (all_links[0]+'num_homes=10000'+all_links[1])
    except Exception as e:
        print('Unable to get download link for: '+countyName)
        return
```

```
# Remove None from the download_urls
download_urls = [searchByCounty(county) for county in all_ca_counties]
urls = list(filter(None, download_urls))
```



Note: If this program is executed multiple times, redfin blocks the automated program by prompting a captcha. In this case, a manual intervention is needed to answer the captcha as selenium cannot automatically bypass captcha



4. Now that we have the download links, lets close the browser

```
# Close the browser
driver.close()
```

5. Let's create a directory *CountyData* in the same directory as that if your Jupyter notebook and use python requests library to download the data from Redfin. Each county redfin data will be saved to this directory.

```
# This method will download the refin data and save it to the csv file
def downloadCSV(url, index):
    headers = {"accept" : "text/html,application/xhtml+xml,application/xml;q=
0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
    "accept-encoding" : "gzip, deflate, br",
    "accept-language" : "en-US,en;q=0.9",
    "cache-control" : "max-age=0",
    "user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/92.0.4515.159 Safari/537.36"}
    response = requests.get(url, headers=headers)
    if response.ok:
        data = response.content.decode('utf8')
        df = pd.read_csv(io.StringIO(data))
        if not os.path.exists('CountyData'):
            os.mkdir('CountyData')
        df.to_csv('CountyData/'+str(index)+'.csv')
```

```
# Download the redfin data for each county
for idx, url in enumerate(urls):
    downloadCSV(url, idx)
```

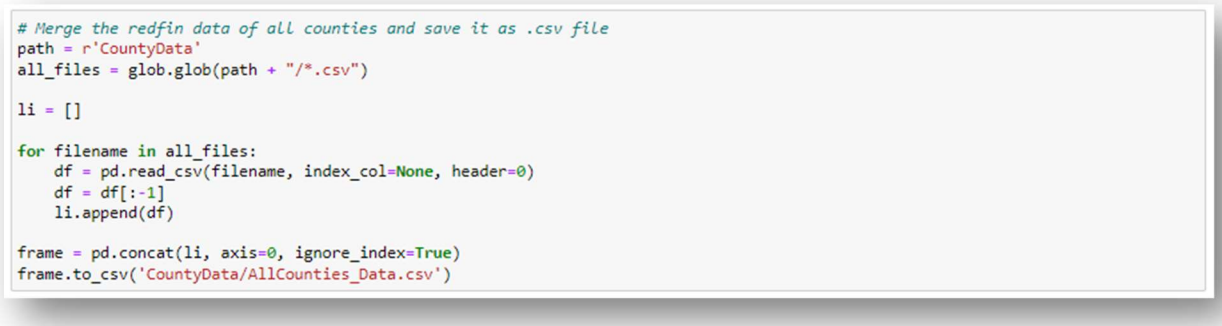
```python
# This method will download the refin data and save it to the csv file
def downloadCSV(url, index):
    headers = {"accept" : "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,appli
    "accept-encoding" : "gzip, deflate, br",
    "accept-language" : "en-US,en;q=0.9",
    "cache-control" : "max-age=0",
    "user-agent" : "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/
    response = requests.get(url, headers=headers)
    if response.ok:
        data = response.content.decode('utf8')
        df = pd.read_csv(io.StringIO(data))
        if not os.path.exists('CountyData'):
            os.mkdir('CountyData')
        df.to_csv('CountyData/'+str(index)+'.csv')
```

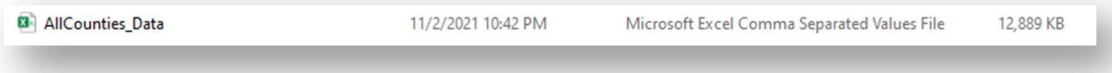
```python
# Download the redfin data for each county
for idx, url in enumerate(urls):
    downloadCSV(url, idx)
```

Name	Date modified	Type	Size
0	11/2/2021 10:37 PM	Microsoft Excel Comma Separated Values File	2,734 KB
1	11/2/2021 10:37 PM	Microsoft Excel Comma Separated Values File	692 KB
2	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	236 KB
3	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	1,926 KB
4	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	1,749 KB
5	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	909 KB
6	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	110 KB
7	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	797 KB
8	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	636 KB
9	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	192 KB
10	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	141 KB
11	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	203 KB
12	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	354 KB
13	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	396 KB
14	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	437 KB
15	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	101 KB
16	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	374 KB
17	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	147 KB
18	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	91 KB
19	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	247 KB
20	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	85 KB
21	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	37 KB
22	11/2/2021 10:38 PM	Microsoft Excel Comma Separated Values File	42 KB

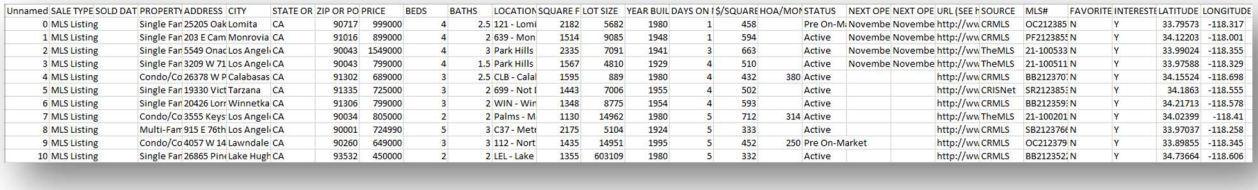
6. Let’s merge all this information to form a masters csv file that contains all the housing information of California counties

```
# Merge the redfin data of all counties and save it as .csv file
path = r'CountyData'
all_files = glob.glob(path + "/*.csv")

li = []

for filename in all_files:
    df = pd.read_csv(filename, index_col=None, header=0)
    df = df[:-1]
    li.append(df)

frame = pd.concat(li, axis=0, ignore_index=True)
frame.to_csv('CountyData/AllCounties_Data.csv')
```





7. A sample screenshot of housing information that is in *AllCounties_Data.csv*



Unnamed	SALE TYPE	SOLD DAT	PROPERTY	ADDRESS	CITY	STATE OR	ZIP OR PO	PRICE	BEDS	BATHS	LOCATION	SQUARE F	LOT SIZE	YEAR BUIL	DAYS ON I	$/SQUARE	HOA/MOI	STATUS	NEXT OPE	NEXT OPE	URL (SEE I	SOURCE	MLS#	FAVORITE	INTERESTE	LATITUDE	LONGITUDE
0	MLS Listing		Single Fan	25205 Oak	Lomita	CA	90717	999000	4	2.5	121 - Lomi	2182	5682	1980	1	458		Pre On-M	Novembe	Novembe	http://ww	CRMLS	OC212385	N	Y	33.79573	-118.317
1	MLS Listing		Single Fan	203 E Cam	Monrovia	CA	91016	899000	4	2	639 - Mon	1514	9085	1948	1	594		Active	Novembe	Novembe	http://ww	CRMLS	PF212385	N	Y	34.12203	-118.001
2	MLS Listing		Single Fan	5549 Onac	Los Angel	CA	90043	1549000	4	3	Park Hills	2335	7091	1941	3	663		Active	Novembe	Novembe	http://ww	TheMLS	21-100533	N	Y	33.99024	-118.355
3	MLS Listing		Single Fan	3209 W 71	Los Angel	CA	90043	799000	4	1.5	Park Hills	1567	4810	1929	4	510		Active	Novembe	Novembe	http://ww	TheMLS	21-100511	N	Y	33.97588	-118.329
4	MLS Listing		Condo/Co	26378 W P	Calabasas	CA	91302	689000	3	2.5	CLB - Cala	1595	889	1980	4	432	380	Active			http://ww	CRMLS	BB212370	N	Y	34.15524	-118.698
5	MLS Listing		Single Fan	19330 Vict	Tarzana	CA	91335	725000	3	2	699 - Not I	1443	7006	1955	4	502		Active			http://ww	CRISNet	SR212385	N	Y	34.1863	-118.555
6	MLS Listing		Single Fan	20426 Lorr	Winnetka	CA	91306	799000	3	2	WIN - Win	1348	8775	1954	4	593		Active			http://ww	CRMLS	BB212359	N	Y	34.21713	-118.578
7	MLS Listing		Condo/Co	3555 Keys	Los Angel	CA	90034	805000	2	2	Palms - M	1130	14962	1980	5	712	314	Active			http://ww	TheMLS	21-100201	N	Y	34.02399	-118.41
8	MLS Listing		Multi-Fam	915 E 76th	Los Angel	CA	90001	724990	5	3	C37 - Mete	2175	5104	1924	5	333		Active			http://ww	CRMLS	SB212376	N	Y	33.97037	-118.258
9	MLS Listing		Condo/Co	4057 W 14	Lawndale	CA	90260	649000	3	3	112 - Nort	1435	14951	1995	5	452	250	Pre On-Market			http://ww	CRMLS	OC212379	N	Y	33.89855	-118.345
10	MLS Listing		Single Fan	26865 Pin	Lake Hugh	CA	93532	450000	2	2	LEL - Lake	1355	603109	1980	5	332		Active			http://ww	CRMLS	BB212352	N	Y	34.73664	-118.606

We will further be using this data to develop a question & answering system using BERT and python levenshtein distance.