Michelle Helfman Term Project Milestone 3

Moving Starter Kit

The Moving Starter Kit contains basic demographic, economic, education, and additional location-based information to be used as a starting point to finding a new city to live or confirm the current location is the best place to be. The addition of region allows for parsing based on parts of the country and the state capital can be used for weather, if the metropolitan anchor city is missing.

Note - There are no outliers or bad data. Both websites have have information on all 50 dates.

In [1]:
```
# Import Functions

import pandas as pd
import numpy as np
import os
import requests

from bs4 import BeautifulSoup
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
from datetime import datetime

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Delete the existing output file.
file = 'MSK Milestone 3.xlsx'
location = "C:/DSC540_Data/"
path = os.path.join(location, file)

# Remove the file
try:
    os.remove(path)

except:
    print('No Prior File Deleted')
```

In [3]:
```
# Open the US Region/State Xref and the capitals website,
# read the file, and create a soup and print it

# The US Region/State Xref site
link = 'https://www.mappr.co/political-maps/us-regions-map/'
mappr_file = requests.get(link)
mappr_soup = BeautifulSoup(mappr_file.content, 'html.parser')
mappr_file.close()

# The States and Capitals website
link2 = 'https://thefactfile.org/u-s-states-and-capitals/'
fact_file = requests.get(link2)
fact_soup = BeautifulSoup(fact_file.content, 'html.parser')
fact_file.close()
```

In [4]:
```
# Gather information to scraping the region information.

num_figures = mappr_soup.find_all("figure")
print("Number of figures in Regions File = {} ".format(len(num_figures)))

print('\nClasses of each figure:')
for figure in mappr_soup.find_all('figure'):
    print(figure.get('class'))
```

```
Number of figures in Regions File = 4

Classes of each figure:
['wp-block-image', 'size-large']
['wp-block-table', 'is-style-regular']
['wp-block-image', 'size-large', 'is-style-default']
['wp-block-image', 'size-large', 'is-style-default']
```

In [5]:
```
# Find the regions figure using the wp-block-table is-style-regular class

data_table = mappr_soup.find("figure", {"class": "wp-block-table is-style-regular"})

print('Regions table html sample')
print(data_table)
```

```
Regions table html sample
<figure class="wp-block-table is-style-regular"><table><thead><tr><th>Abbreviation</th><th>Stat
e Name</th><th>Region</th></tr></thead><tbody><tr><td>AL</td><td>Alabama</td><td>Southeast</td>
</tr><tr><td>AK</td><td>Alaska</td><td>West</td></tr><tr><td>AZ</td><td>Arizona</td><td>Southwe
st</td></tr><tr><td>AR</td><td>Arkansas</td><td>Southeast</td></tr><tr><td>CA</td><td>Californi
a</td><td>West</td></tr><tr><td>CO</td><td>Colorado</td><td>West</td></tr><tr><td>CT</td><td>Co
nnecticut</td><td>Northeast</td></tr><tr><td>DE</td><td>Delaware</td><td>Northeast</td></tr><tr
><td>FL</td><td>Florida</td><td>Southeast</td></tr><tr><td>GA</td><td>Georgia</td><td>Southeast
</td></tr><tr><td>HI</td><td>Hawaii</td><td>West</td></tr><tr><td>ID</td><td>Idaho</td><td>West
</td></tr><tr><td>IL</td><td>Illinois</td><td>Midwest</td></tr><tr><td>IN</td><td>Indiana</td><
td>Midwest</td></tr><tr><td>IA</td><td>Iowa</td><td>Midwest</td></tr><tr><td>KS</td><td>Kansas
</td><td>Midwest</td></tr><tr><td>KY</td><td>Kentucky</td><td>Southeast</td></tr><tr><td>LA</td
><td>Louisiana</td><td>Southeast</td></tr><tr><td>ME</td><td>Maine</td><td>Northeast</td></tr><
tr><td>MD</td><td>Maryland</td><td>Northeast</td></tr><tr><td>MA</td><td>Massachusetts</td><td>
Northeast</td></tr><tr><td>MI</td><td>Michigan</td><td>Midwest</td></tr><tr><td>MN</td><td>Minn
esota</td><td>Midwest</td></tr><tr><td>MS</td><td>Mississippi</td><td>Southeast</td></tr><tr><t
d>MO</td><td>Missouri</td><td>Midwest</td></tr><tr><td>MT</td><td>Montana</td><td>West</td></tr
><tr><td>NE</td><td>Nebraska</td><td>Midwest</td></tr><tr><td>NV</td><td>Nevada</td><td>West</t
d></tr><tr><td>NH</td><td>New Hampshire</td><td>Northeast</td></tr><tr><td>NJ</td><td>New Jerse
```

In [6]:
```
# Find the # of tables within the figure

region_data = data_table.tbody.findAll('tr',
                                       recursive=False)[1].findAll('td', recursive=False)
region_tables = []
for td in region_data:
    region_tables.append(td.findAll('table'))

print("Number of Region Data Tables = {} ".format(len(region_tables)))
```

```
Number of Region Data Tables = 3
```

In [7]:
```
# Put together the region column headers

data_thead = data_table.thead

column_headers = [th.getText().strip() for th in data_thead.findAll('th')]

print('Region Column Headers')
column_headers
```

```
Region Column Headers
```

Out[7]:
```
['Abbreviation', 'State Name', 'Region']
```

In [8]:
```
# Extract Region information and attach the headers to
# the regions dataframe

data_body = data_table.tbody

detail_info = [[td.get_text().strip() for td in tr.findAll('td')] for tr in data_body]
region_df = pd.DataFrame(detail_info, columns = column_headers)

print('State and Region information')
region_df.head(10)
```

```
State and Region information
```

Out[8]:

	Abbreviation	State Name	Region
0	AL	Alabama	Southeast
1	AK	Alaska	West
2	AZ	Arizona	Southwest
3	AR	Arkansas	Southeast
4	CA	California	West
5	CO	Colorado	West
6	CT	Connecticut	Northeast
7	DE	Delaware	Northeast
8	FL	Florida	Southeast
9	GA	Georgia	Southeast

In [9]:
```
# Find and print the # of tables and verify their classes
# from the capitals site

num_tables = fact_soup.find_all("table")
print("Number of Tables in Capitals Website = {} ".format(len(num_tables)))

print('\nClasses of each table:')
for table in fact_soup.find_all('table'):
    print(table.get('class'))
```

```
Number of Tables in Capitals Website = 1

Classes of each table:
['tablepress', 'tablepress-id-175']
```

In [10]:
```
# Find the state capitals table using the tablepress tablepress-id-175 class

data_table1 = fact_soup.find("table", {"class": "tablepress-id-175"})

print('State Capital table html sample')
print(data_table1)
```

```
State Capital table html sample
<table aria-labelledby="tablepress-175-name" class="tablepress tablepress-id-175" id="tablepres
s-175">
<thead>
<tr class="row-1 odd">
<th class="column-1">Serial</th><th class="column-2">State Name</th><th class="column-3">State
Capital</th><th class="column-4">Postal Abbreviation</th>
</tr>
</thead>
<tbody class="row-hover">
<tr class="row-2 even">
<td class="column-1">1.</td><td class="column-2">Alaska</td><td class="column-3">Juneau</td><td
class="column-4">AK</td>
</tr>
<tr class="row-3 odd">
<td class="column-1">2.</td><td class="column-2">Texas</td><td class="column-3">Austin</td><td
class="column-4">TX</td>
</tr>
<tr class="row-4 even">
```

In [11]:
```
# Find the # of tables within the figure

capitals_data = data_table1.tbody.findAll('tr',
                                  recursive=False)[1].findAll('td', recursive=False)
capitals_tables = []
for td1 in capitals_data:
    capitals_tables.append(td1.findAll('table'))

print("Number of Capital Data Tables = {} ".format(len(capitals_tables)))
```

```
Number of Capital Data Tables = 4
```

In [12]:
```
# Put together the capital column headers

data_thead1 = data_table1.thead

column_headers1 = [th1.getText().strip() for th1 in data_thead1.findAll('th')]

print('Capitals column headers')
column_headers1
```

```
Capitals column headers
```

Out[12]:
```
['Serial', 'State Name', 'State Capital', 'Postal Abbreviation']
```

In [13]:
```
# Retrieve the Capital Details and create the capitals dataframe
# and attavh the headers.  Sort the capitals.

data_body1 = data_table1.tbody

#detail_info1 = [[td1.get_text().strip() for td1 in tr1.findAll('td')] for tr1 in data_body1]
#detail_info1 = [[td1.get_text().strip() for td1 in tr1.findAll('td')] for tr1 in data_body1]

details1 = data_body1.findAll('tr')
detail_info1 = [[td1.get_text().strip() for td1 in tr1.findAll('td')] for tr1 in details1]
capital_df = pd.DataFrame(detail_info1, columns = column_headers1)

capital_df = capital_df.sort_values('State Name')
capital_df.head(10)

```

Out[13]:

	Serial	State Name	State Capital	Postal Abbreviation
30	31.	Alabama	Montgomery	AL
0	1.	Alaska	Juneau	AK
5	6.	Arizona	Phoenix	AZ
29	30.	Arkansas	Little Rock	AR
2	3.	California	Sacramento	CA
7	8.	Colorado	Denver	CO
47	48.	Connecticut	Hartford	CT
48	49.	Delaware	Dover	DE
19	20.	Florida	Tallahassee	FL
22	23.	Georgia	Atlanta	GA

1. Rename column headers

Rename the region and capitals information to reflect the website the information comes from.

Note - These column headers will be renamed after Fuzzy Matching

In [14]:
```
# Rename the column headers in both dataframes to reflect the
# website the information comes from.

# Rename Regions DF Columns
new_region_headers = {'Abbreviation': 'R_Abbreviation', 'State Name': 'R_State_Name',
'Region': 'R_Region'}

region_df.rename(columns = new_region_headers, inplace = True)

# Rename Capital DF Columns
new_capital_headers = {'Serial': 'C_Serial', 'State Name': 'C_State_Name',
'State Capital': 'C_State_Capital', 'Postal Abbreviation': 'C_Postal_Abbreviation'}

capital_df.rename(columns = new_capital_headers, inplace = True)
```

2. Fuzzy Matching

Use Fuzzy Matching to merge the information from the 2 websites.

In [15]:

```
# Use Fuzzy Matching to match the regions and capitals dataframes.
# Then add the state capital to the regions dataframe.

matched_areas = []
for row in region_df.index:
    reg_state_name = region_df._get_value(row,'R_State_Name')
    for columns in capital_df.index:
        cap_state_name = capital_df._get_value(columns,'C_State_Name', )
        matched_token = fuzz.partial_ratio(reg_state_name,cap_state_name )
        if matched_token == 100:
            matched_areas.append([reg_state_name,cap_state_name,matched_token])

# Convert matched results to a DataFrame
fuzzy_state_df = pd.DataFrame(matched_areas, columns=['R_State_Name',
                                                      'C_State_Name','PCT'])

# Merge the matched columns with the additional crime dataframe
matched_state_df = fuzzy_state_df.merge(capital_df)

# Add the state capitals to the regions dataframes
state_info_df = region_df.merge(matched_state_df)

state_info_df.head(10)
```

Out[15]:

	R_Abbreviation	R_State_Name	R_Region	C_State_Name	PCT	C_Serial	C_State_Capital	C_Postal_Abbreviation
0	AL	Alabama	Southeast	Alabama	100	31.	Montgomery	AL
1	AK	Alaska	West	Alaska	100	1.	Juneau	AK
2	AZ	Arizona	Southwest	Arizona	100	6.	Phoenix	AZ
3	AR	Arkansas	Southeast	Arkansas	100	30.	Little Rock	AR
4	CA	California	West	California	100	3.	Sacramento	CA
5	CO	Colorado	West	Colorado	100	8.	Denver	CO
6	CT	Connecticut	Northeast	Connecticut	100	48.	Hartford	CT
7	DE	Delaware	Northeast	Delaware	100	49.	Dover	DE
8	FL	Florida	Southeast	Florida	100	20.	Tallahassee	FL
9	GA	Georgia	Southeast	Georgia	100	23.	Atlanta	GA

3. Drop Duplicate Columns

Remove the columns containing duplicate information.

In [16]:

```
# Remove Unnecessary and Duplicate Columns:  C_State_Name, PCT,
# C_Serial, and C_Postal Abbreviation.

state_info_df.drop(state_info_df.iloc[:, 3:6],  axis=1, inplace = True)
state_info_df.drop(state_info_df.iloc[:, 4:],  axis=1, inplace = True)
```

4. Rename column headers

Standardize the column header by removing the website designation.

In [17]:
```
# Rename Capital DF Columns
new_state_info_headers = {'R_Abbreviation': 'Abbreviation', 'R_State_Name': 'State Name',
'R_Region': 'Region', 'C_State_Capital': 'State Capital'}

state_info_df.rename(columns = new_state_info_headers, inplace = True)

state_info_df.head(10)
```

Out[17]:

	Abbreviation	State Name	Region	State Capital
0	AL	Alabama	Southeast	Montgomery
1	AK	Alaska	West	Juneau
2	AZ	Arizona	Southwest	Phoenix
3	AR	Arkansas	Southeast	Little Rock
4	CA	California	West	Sacramento
5	CO	Colorado	West	Denver
6	CT	Connecticut	Northeast	Hartford
7	DE	Delaware	Northeast	Dover
8	FL	Florida	Southeast	Tallahassee
9	GA	Georgia	Southeast	Atlanta

5. Create Additional Columns.

Create new columns from the website information.

In [18]:
```
# Add a Capital City, State column and Capital City, Abbreviation

state_info_df['Capital and State'] = (state_info_df['State Capital'] + ', '
                                      +  state_info_df['State Name'])

state_info_df['Capital and Abbrev'] = (state_info_df['State Capital'] + ', '
                                      +  state_info_df['Abbreviation'])
```

6. Create a Timestamp column

Get the current time and add it to the State Information

In [19]:

```
# Get the current time and add it to the State Information and sort the
# information

state_info_df['Create Date'] = datetime.now()

state_info_df.sort_values('Abbreviation')

state_info_df.head(10)
```

Out[19]:

	Abbreviation	State Name	Region	State Capital	Capital and State	Capital and Abbrev	Create Date
0	AL	Alabama	Southeast	Montgomery	Montgomery, Alabama	Montgomery, AL	2023-05-07 15:11:07.893966
1	AK	Alaska	West	Juneau	Juneau, Alaska	Juneau, AK	2023-05-07 15:11:07.893966
2	AZ	Arizona	Southwest	Phoenix	Phoenix, Arizona	Phoenix, AZ	2023-05-07 15:11:07.893966
3	AR	Arkansas	Southeast	Little Rock	Little Rock, Arkansas	Little Rock, AR	2023-05-07 15:11:07.893966
4	CA	California	West	Sacramento	Sacramento, California	Sacramento, CA	2023-05-07 15:11:07.893966
5	CO	Colorado	West	Denver	Denver, Colorado	Denver, CO	2023-05-07 15:11:07.893966
6	CT	Connecticut	Northeast	Hartford	Hartford, Connecticut	Hartford, CT	2023-05-07 15:11:07.893966
7	DE	Delaware	Northeast	Dover	Dover, Delaware	Dover, DE	2023-05-07 15:11:07.893966
8	FL	Florida	Southeast	Tallahassee	Tallahassee, Florida	Tallahassee, FL	2023-05-07 15:11:07.893966
9	GA	Georgia	Southeast	Atlanta	Atlanta, Georgia	Atlanta, GA	2023-05-07 15:11:07.893966

In [20]:

```
# Write out the State Information
state_info_df.to_excel("C:/DSC540_Data/MSK Milestone 3.xlsx", sheet_name='State Information')
```