Data of different counties' Population and household Income (2012 and 2017):

Source of Dataset:
Data was downloaded from https://api.census.gov

From the dataset County was considered instead of Metro because Government often change the boundaries of the Metro Areas in every census. So there was a possibility of a wrong parameter to judge the data set.

Convert dataset into DataFrame:

This dataset included the population and household income of counties' in different states in the year 2017. API was the source of access to get the data. B01001_001E was the code for total population and B19013_001E was the code for total household income. json was the reader. Data for population and income was filtered by .iloc[] function. Then dataset was converted into DataFrame.

```
#convert to dataframe and clean up header
census_pd_2017 = pd.DataFrame(response2)
census_pd_2017.columns = census_pd_2017.iloc[0]
census_pd_2017 = census_pd_2017.iloc[1:]
census_pd_2017.head()
```

	B01001_001E	B19013_001E	NAME	state	county
1	34933	14752	Corozal Municipio, Puerto Rico	72	047
2	11297	17636	Maunabo Municipio, Puerto Rico	72	095
3	21661	16868	Peñuelas Municipio, Puerto Rico	72	111
4	148863	16561	Ponce Municipio, Puerto Rico	72	113
5	38970	14275	San Sebastián Municipio, Puerto Rico	72	131

Same process followed for 2012 dataset. B01001_001E was the code for total population and B19013_001E was the code for total household income.

```
#convert to dataframe
census_pd_2012 = pd.DataFrame(response3)
census_pd_2012.columns = census_pd_2012.iloc[0]
census_pd_2012 = census_pd_2012.iloc[1:]
census_pd_2012.head()
```

	B01001_001E	B19013_001E	NAME	state	county
1	54590	53773	Autauga County, Alabama	01	001
2	183226	50706	Baldwin County, Alabama	01	003
3	27469	31889	Barbour County, Alabama	01	005
4	22769	36824	Bibb County, Alabama	01	007
5	57466	45192	Blount County, Alabama	01	009

Next step was merged the two data by inner join

```
#merge 2012 and 2017 df's
census_final = pd.merge(census_pd_2012,census_pd_2017, on="County", how="inner")
```

	2012 Pop	2012 Household Income	County	state_x	county_x	2017 Pop	2017 Household Income	state_y	county_y
0	54590	53773	Autauga County, Alabama	01	001	55036	55317	01	001
1	183226	50706	Baldwin County, Alabama	01	003	203360	52562	01	003
2	27469	31889	Barbour County, Alabama	01	005	26201	33368	01	005
3	22769	36824	Bibb County, Alabama	01	007	22580	43404	01	007
4	57466	45192	Blount County, Alabama	01	009	57667	47412	01	009

Drop the State and county code columns' which are not required after tally. Final dataset is ready. Convert the dataset into csv file for further analysis.

```
#clean up columns
census_final = census_final.drop(census_final.columns[[3,4,7, 8]], axis=1)
```

	County	2012 Pop	2012 Household Income	2017 Pop	2017 Household Income
0	Autauga County, Alabama	54590	53773	55036	55317
1	Baldwin County, Alabama	183226	50706	203360	52562
2	Barbour County, Alabama	27469	31889	26201	33368
3	Bibb County, Alabama	22769	36824	22580	43404
4	Blount County, Alabama	57466	45192	57667	47412

```
#export to CSV
census_final.to_csv("census.csv")
```

Data of Yelp rating of Five Guys, Halal Guys, McDonalds, Panera Bread, Shake Shack, Taco Bell, Texas Roadhouse in different Metro Areas:

Source of Dataset:

This dataset was downloaded from Kaggle: https://www.kaggle.com/yelp-dataset/yelp-dataset

This dataset showcased restaurants and businesses found on Yelp and information concerning the location and user reciprocity. Overall, it seemed outdated and incomplete as some major cities, such as Los Angeles, didn't seem to have entries. However, the dataset did include entries on all restaurant franchises that interested this project. Since it was downloaded, there was the benefit of not having to rely on APIs to access the data.

For each entry, the columns of information were:

1. Address (location)
2. Attributes: keywords Categories: Another list of keywords to describe the type of food and service e.g. "Fast Food"
3. and phrases used to describe the business e.g. "Accepts Credit Cards"
4. Business ID: Unique code for this location. Unique among separate locations in a franchise.
5. City (location)
6. Hours Open
7. Latitude (location)
8. Longitude (location)
9. Name of Business/Franchise
10. Postal Code
11. Review Count: Number of reviews this location has received
12. Stars: Average star rating of above review count.
13. State (location)

1. Cleaning the Raw Data to Only Include Franchises Related to this Project

First, the data was retrieved from the Resources directory it was downloaded into. Then, all entries not related to the eight franchises selected for the project were filtered out using the .loc() function. Finally, all entries with names 'Five Guys Burgers and Fries' were renamed to 'Five Guys' since they represent the same franchise. This was done with the. replace () function.

```python
# Import business dataset from resources
biz1_df = pd.read_json('Resources/yelp_academic_dataset_business.json', lines=True)

# save the row data for
biz1_df = biz1_df.loc[(biz1_df['name'] == "The Halal Guys") |              # 10
                      (biz1_df['name'] == "Chipotle Mexican Grill") |       # 183
                      (biz1_df['name'] == "Taco Bell") |                    # 313
                      (biz1_df['name'] == "McDonald's") |                   # 806
                      (biz1_df['name'] == "Panera Bread") |                 # 157
                      (biz1_df['name'] == "Five Guys Burgers and Fries") |  # 10
                      (biz1_df['name'] == "Five Guys") |                    # 99
                      (biz1_df['name'] == "Texas Roadhouse") |              # 24
                      (biz1_df['name'] == "Shake Shack")                    # 10
                       , :]

# 'Five Guys' will need to combine with 'Five Guys Burgers and Fries'
biz1_df['name'] = biz1_df['name'].replace({"Five Guys Burgers and Fries":"Five Guys"})
#biz1_df['name'].value_counts()
biz1_df.head()
```

2. Manipulating the Data for the 'Review Count Spread' DataFrame

First, create a new DataFrame with only the columns needed (name, stars, review_count), then create the double groupby via name then stars.

```python
# Groupby resturant and star rating
biz3_df = biz2_df.loc[:,["name","stars","review_count"]]
biz3_df = biz3_df.groupby(["name","stars"]).agg({"review_count":"sum"})
# This also works: biz3_df = biz3_df.groupby(["name","stars"]).sum()
biz3_df.head()
```

Second, unstack and level the DataFrame so a pivot level is created comparing star numeric ranking and franchise.

```python
# Pivot the name index (row headers) to a column header
biz3_df = biz3_df.unstack(0)
biz3_df.columns = biz3_df.columns.get_level_values(1)

# Data Munging: Fill in the NaN and combine 'Five Guys' with 'Five Guys Burgers and Fries'
biz3_df = biz3_df.fillna(0)

# This DataFrame shows the number of average star ratings of each franchise from 1.0 to 5.0
biz3_df
```

1. Manipulating the Data for 'Top Ten Most Popular Cities for Each Franchise'

First, a new DataFrame was created with relevant columns (name, city, stars).

```python
# Condense the above DataFrame as shown to include cities
biz4_df = pd.DataFrame(biz1_df[['name','city','stars']])
biz4_df.head()
```

Second, create new, single column DataFrames for each franchise ranking the cities by number of franchise locations.

```
chi_df = biz4_df.loc[biz4_df['name'] == "Chipotle Mexican Grill",:]
chi_df = chi_df['city'].value_counts(ascending=False)
chi_df = pd.DataFrame({"Chipotle":chi_df.index})
```

Lastly, combine the single column DataFrames of each franchise into one with the .concat() function to see the top ten cities.

```
# Make the conbined DataFrame showing the Top Ten Cities by number of franchise locations
topten_df = pd.concat([chi_df, fiv_df, hal_df, tac_df, mcd_df, pan_df, tex_df, sha_df], axis=1)
topten_df.head(11)
```

	Chipotle	Five Guys	Halal Guys	Taco Bell	McDonald's	Panera Bread	Texas Roadhouse	Shake Shack
0	Las Vegas	Charlotte	Las Vegas	Las Vegas	Las Vegas	Charlotte	Phoenix	Las Vegas
1	Phoenix	Phoenix	MontrÃ©al	Phoenix	Phoenix	Pittsburgh	Gilbert	Scottsdale
2	Charlotte	Calgary	Phoenix	Charlotte	Toronto	Phoenix	Pittsburgh	Phoenix
3	Pittsburgh	Pittsburgh	Tempe	Mesa	Charlotte	Las Vegas	Brooklyn	Henderson
4	Scottsdale	Toronto	Toronto	Glendale	Calgary	Chandler	North Las Vegas	Orange Village
5	Toronto	Las Vegas	Mesa	Pittsburgh	MontrÃ©al	Madison	Elyria	Charlotte
6	Mesa	Mississauga	Mississauga	Cleveland	Pittsburgh	Scottsdale	Bridgeville	NaN
7	Cleveland	Henderson	NaN	Madison	Mesa	Tempe	Surprise	NaN
8	Glendale	Mesa	NaN	Scottsdale	Mississauga	Champaign	Mesa	NaN
9	Tempe	Strongsville	NaN	Tempe	Cleveland	Mississauga	Willoughby	NaN
10	Gilbert	Matthews	NaN	Henderson	Scottsdale	Henderson	Concord	NaN

Data pertaining to popularity rating of Chipotle, Five Guys, Halal Guys, McDonalds, Panera Bread, Shake Shack, Taco Bell, Texas Roadhouse in different states in US for the last 2 years:

Source of Dataset: Google Trends (https://trends.google.com/trends/?geo=US)

Size of Data: State-wise Dataset– 8 x 1KB

Multi-Timeline Dataset- 8 x 4KB

The dataset consisted scores of popularity, for each franchise, with respect to the states, as well as a timeline of 2 years. The data was unbiased and as authentic as it can be, because it was generated purely on the basis of search results and interests on Google. However, the dataset wasn’t collaborated to include all the franchises, at once.

Steps of Data Cleaning:

1. **State-wise Dataset**

The datasets of different franchises were merged to one Data Frame, for analysis.

```
Combined_df = pd.read_csv("Resources\Resources\Combined_GeoMap.csv")
Combined_df.head()
```

	Region	Chipotle	Five Guys	Halal Guys	McDonalds	Panera bread	Shake Shack	Taco Bell	Texas Roadhouse
0	Alabama	21	54	0	80	76	12	87	38
1	Alaska	9	7	0	59	4	4	55	50
2	Arizona	59	45	31	78	49	25	75	61
3	Arkansas	17	28	0	90	41	8	90	44
4	California	46	35	57	64	43	30	58	13

Since Halal Guys didn’t have data for a few states, because of the lack of popularity, the null values had to be filled.

```
Combined_df.fillna("0")
Combined_df.head()
```

	Region	Chipotle	Five Guys	Halal Guys	McDonalds	Panera bread	Shake Shack	Taco Bell	Texas Roadhouse
0	Alabama	21	54	0	80	76	12	87	38
1	Alaska	9	7	0	59	4	4	55	50
2	Arizona	59	45	31	78	49	25	75	61
3	Arkansas	17	28	0	90	41	8	90	44
4	California	46	35	57	64	43	30	58	13

The data type of the values under “Halal Guys” column had to be changed to match that of the remaining dataset.

```
Combined_df['Halal Guys'] = Combined_df['Halal Guys'].astype(int)
Combined_df.dtypes
```

```
Region              object
Chipotle             int64
Five Guys            int64
Halal Guys           int32
McDonalds            int64
Panera bread         int64
Shake Shack          int64
Taco Bell            int64
Texas Roadhouse      int64
dtype: object
```

Finally converted the dataset into csv for further analysis.

```
Combined_df.to_csv('Resources\Resources\Combined_GeoMap.csv')
Combined_df.head()
```

	Region	Chipotle	Five Guys	Halal Guys	McDonalds	Panera bread	Shake Shack	Taco Bell	Texas Roadhouse
0	Alabama	21	54	0	80	76	12	87	38
1	Alaska	9	7	0	59	4	4	55	50
2	Arizona	59	45	31	78	49	25	75	61
3	Arkansas	17	28	0	90	41	8	90	44
4	California	46	35	57	64	43	30	58	13

2. Multi-Timeline Dataset

The acquired datasets had to be merged and munged to meet our objectives and coding requirements.

To join the dataset, inner join function was used. It was combined into a single Data Frame with the .concat() function to see the overall rating of those franchises in different states foreach week.

```
glob_path = "New Timeline data"
all_files = glob.glob(glob_path+"/*.csv")
glob_df = pd.concat([pd.read_csv(fp).assign(New=os.path.basename(fp).split('.')[0]) for fp in all_files])
```

	Category: All categories	New
Week	Chipotle Mexican Grill: (United States)	Chipotle
2017-09-24	65	Chipotle
2017-10-01	66	Chipotle
2017-10-08	61	Chipotle
2017-10-15	64	Chipotle

The glob function was used to track and match the path of the csv of each franchise, to include in the current Data Frame.

```
glob_df.reset_index(level=0, inplace=True)
glob_df.columns = glob_df.iloc[0]
glob_df = glob_df.iloc[1:]
glob_df = glob_df.rename(columns={"Chipotle Mexican Grill: (United States)":"Score","Chipotle": "Franchise"})
glob_df.keys()

#table = pd.pivot_table(glob_df, index =['New'])
```

Unstacked and levelled the DataFrame into a pivot level, comparing franchises and respective score for each week, of 2 years.

								Score
Franchise	**Chipotle**	**Five Guys**	**Halal Guys**	**McDonalds**	**Panera Bread**	**Shake Shack**	**Taco Bell**	**Texas Roadhouse**
Week								
2017-09-24	65	73	74	55	88	46	61	48
2017-10-01	66	75	74	75	87	46	68	47
2017-10-08	61	75	71	65	91	100	66	48
2017-10-15	64	79	79	57	90	60	62	48
2017-10-22	67	77	75	57	91	53	64	47

Finally converted the dataset into csv for further analysis.

```
table.to_csv("New Timeline data/combined.csv")
```