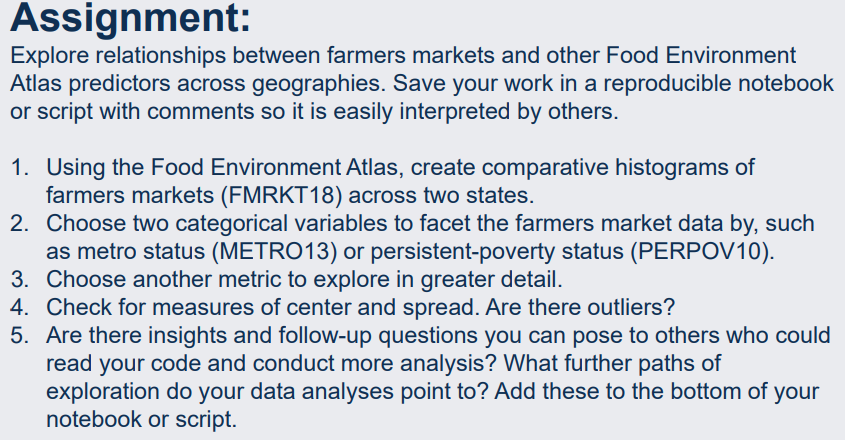
**Project 1**



**Question 1:**

import pandas as pd

import seaborn as sns

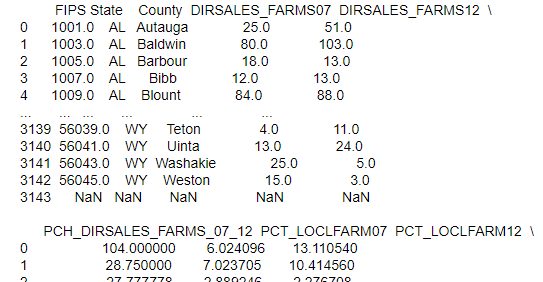
import matplotlib.pyplot as plt

import numpy as np

##Q1 importing Local dataset

df = pd.read\_excel(r'C:\Users\shaun.rolph\Desktop\Assignmentone\FoodEnvironmentAtlas.xls', sheet\_name='LOCAL')

print(df)



##Q1, removing variables not needed for analysis

df = pd.DataFrame(df, columns=['FMRKT18', 'State', 'County', 'FIPS'])

print(df)

Text

Description automatically generated

##Q1, looking at state sums

state\_sum = df.groupby ('State') ['FMRKT18'].sum()

print(state\_sum)

Text, table

Description automatically generated with medium confidence

##Q1, filtering two states and comparing the frequency and county count of farmers markets

## List of states you want to filter by

state1 = 'MD'

state2 = 'WY'

## Filter the DataFrame for the two states

filtered\_df\_state1 = df[df['State'] == state1]

filtered\_df\_state2 = df[df['State'] == state2]

## Plot histograms for the frequency of the two filtered states

fig = plt.figure(figsize=(10,10))

plt.hist(filtered\_df\_state1['FMRKT18'], bins=10, alpha=0.5, label=state1, rwidth=0.4)

plt.hist(filtered\_df\_state2['FMRKT18'], bins=10, alpha=0.5, label=state2, rwidth=1)

## Add labels and a legend

plt.xlabel('Number of Farmers Markets Bin',fontsize=14)

plt.ylabel('County Frequency Count', fontsize=14)

plt.legend(loc='upper right', fontsize=14)

x\_ticks = (1,3,5,7,9,11,13,15,17,19,21,23,)

plt.xticks(x\_ticks)

description = "This histogram compares the number of farmers markets per state by county. The x-axis is the number of farmers markets and the y-axis is the count of counties within that number of farmers markets bin"

bbox\_props = dict(boxstyle='round', facecolor='white', edgecolor='black', alpha=0.8)

plt.text(0.5, -0.1, description, transform=plt.gca().transAxes, fontsize=12, ha='center', va='center', bbox=bbox\_props, multialignment='center')

## Show the histogram

plt.title(f'FMRKT18 Distribution Comparison: {state1} vs {state2}',fontsize=18)

plt.show()

Chart, histogram

Description automatically generated

**Question 2:**

##Q2, checking obersation count for local dataset to ensure merge keeps the same count

print(df)

##Q2, looking at nulls

nan\_counts = df.isna().sum()

print (nan\_counts)

##Q2, importing new dataset and removing all variables not needed.

df2 = pd.read\_excel(r'C:\Users\shaun.rolph\Desktop\Assignmentone\FoodEnvironmentAtlas.xls', sheet\_name='SOCIOECONOMIC')

df3 = pd.DataFrame(df2, columns=['METRO13', 'State', 'County', 'FIPS', 'PERPOV10' ])

print(df3)

df3.head()

##Q2, merging datasets together and dropping duplicate values

df4 = df.merge(df3.drop\_duplicates(subset=['County']), left\_on = 'County', right\_on='County', how= 'left' )

df4.head()

Table

Description automatically generated

##Q2, checking to ensure the obervsation are correct

num\_rows = len(df4)

print (num\_rows)

**Question 3:**

##Q3, picking new variables to look at

## Chose PCH\_VEG\_ACRES\_07\_12, Vegetable acres harvested (% change), 2007 - 12

df\_local = pd.read\_excel(r'C:\Users\shaun.rolph\Desktop\Assignmentone\FoodEnvironmentAtlas.xls', sheet\_name='LOCAL')

df\_Q3 = pd.DataFrame(df\_local, columns=['PCH\_VEG\_ACRES\_07\_12', 'State', 'County'])

print(df\_Q3)

##Q3, looking at mean of variables as well as the mean by state for each variable

state\_avg = df\_Q3.groupby ('State') ['PCH\_VEG\_ACRES\_07\_12'].mean()

print(state\_avg)

summary\_stats2 = df\_Q3.describe()

summary\_stats2.head()

Table

Description automatically generatedText

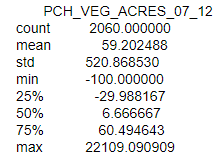
Description automatically generated

**Question 4:**

##Q4, looking as measures of center and spread

summary\_stats3 = df\_Q3.describe(percentiles=[.25, .5, .75])

print(summary\_stats3)



##Q4, dropping nulls from dataset

dfQ3\_droppednulls = df.dropna(subset=['PCH\_VEG\_ACRES\_07\_12'])

##Q4, dropping all variables exept PCH\_VEG\_ACRES\_07\_12, Vegetable acres harvested (% change), 2007 - 12

## and getting zscores for each data point as well as count of outliers

import numpy as np

from scipy.stats import zscore

df\_PCH\_VEG\_ACRES\_07\_12 = pd.DataFrame(dfQ3\_droppednulls, columns=['PCH\_VEG\_ACRES\_07\_12'])

z\_scores = stats.zscore(df\_PCH\_VEG\_ACRES\_07\_12)

print (z\_scores)

Text

Description automatically generated

##Q4 Count of outliers

threshold = 3

outliers\_count = np.sum(np.abs(z\_scores) > threshold)

num\_rows1= len(outliers\_count)

print(num\_rows1)



5. Follow up questions and deeper insights would be to create a correlation matrix for variables and research which might have any correlation of causation in the literature then perform deeper regression analysis using a create model of selected variables to see which impacts PCH\_VEG\_ACRES\_07\_12, Vegetable acres harvested (% change), 2007 – 12.