Homework 4 - Camila Alvarez (Z23400244)

Question 1:

```
import pandas as pd

# Use Panda dataframe to read CSV file
df = pd.read_csv('/Users/cam/Downloads/county_demographics-1.csv')

# Create new column for 'High School Degree Only' by subtracting out the 'Bachelor's Degree or
# Higher' column from the 'High School or Higher' column
df['Education.HighSchoolDegreeOnly'] = df['Education.High School or Higher'] - df["Education.Bachelor's Degree
or Higher"]

# Create new column for 'No School Degree' by subtracting out the 'High School or Higher'
# column from 100.
df['Education.NoSchoolDegree'] = 100 - df['Education.High School or Higher']

# Filter the dataframe to only display Kentucky and Georgia Demographic Information
# My Z23400244 ends in '44' so I chose to go with the next different number of 2.
# Hence, Kentucky and Georgia
df = df[(df['State'] == 'KY') | (df['State'] == 'GA')]

# Downloads updated CSV file with only Kentucky and Georgia demographic information
df.to_csv('/Users/cam/Downloads/Updated_KY_GA_Demographics.csv', index=False)

"""I have attached the PDF file for the updated KY and GA CSV data in table form in the HW"""

# Print the updated DataFrame
print(df)
```

Question 1 - Output:

```
/Users/cam/PycharmProjects/pythonProject4/venv/bin/python /Users/cam/PycharmProjects/pythonProject4/main.py
               County  ... Education.NoSchoolDegree
5         Adair County  ...                    20.4
48        Allen County  ...                    17.9
60     Anderson County  ...                    10.1
75      Appling County  ...                    25.1
100    Atkinson County  ...                    32.8
...                ...  ...                     ...
3050     Wilkes County  ...                    16.3
3053  Wilkinson County  ...                    15.6
3084      Wolfe County  ...                    28.2
3091   Woodford County  ...                     8.9
3098      Worth County  ...                    17.1

[279 rows x 45 columns]

Process finished with exit code 0
```

Question 2:

```
import pandas as pd
import numpy as np

data = pd.read_csv('/Users/cam/Downloads/Updated_KY_GA_Demographics.csv')

# Filter data for Kentucky and Georgia and define columns to be used for calculating statistics
kentucky_data = data[data['State'] == 'KY']
georgia_data = data[data['State'] == 'GA']
cols = ["Education.Bachelor's Degree or Higher", "Education.HighSchoolDegreeOnly", "Education.NoSchoolDegree"]

# Calculate statistics for Kentucky and print values for each column
print("Kentucky Statistics:")
for col in cols:
    values = kentucky_data[col].dropna()
    mode = values.mode().values
    median = np.median(values)
    mean = np.mean(values)
    range = np.ptp(values)
    std = np.std(values, ddof=1)
    quantiles = np.percentile(values, [25, 50, 75])
    print(f"{col}: mode = {mode}, median = {median}, mean = {mean}, range = {range}, std = {std}, quantiles =
{quantiles}")

# Calculate statistics for Georgia and print values for each column
print("\nGeorgia Statistics:")
for col in cols:
    values = georgia_data[col].dropna()
    mode = values.mode().values
    median = np.median(values)
    mean = np.mean(values)
    range = np.ptp(values)
    std = np.std(values, ddof=1)
    quantiles = np.percentile(values, [25, 50, 75])
    print(f"{col}: mode = {mode}, median = {median}, mean = {mean}, range = {range}, std = {std}, quantiles =
{quantiles}")

# Now I will create two new csv files for Kentucky and Georgia as they are easier to read then the python
outputs.

df = pd.read_csv('/Users/cam/Downloads/Updated_KY_GA_Demographics.csv')

# Subset data for Kentucky and Georgia
ky_df = df[df['State'] == 'KY']
ga_df = df[df['State'] == 'GA']

# Subset data for the columns we want to calculate the statistics for
edu_cols = ["Education.Bachelor's Degree or Higher",
            'Education.HighSchoolDegreeOnly',
            'Education.NoSchoolDegree']
ky_edu_df = ky_df[edu_cols]
ga_edu_df = ga_df[edu_cols]

# Calculate and define dataframe for Kentucky statistics
ky_stats = pd.DataFrame({
    'mode': ky_edu_df.mode().iloc[0],
    'median': ky_edu_df.median(),
    'mean': ky_edu_df.mean(),
```

```
    'range': ky_edu_df.max() - ky_edu_df.min(),
    'sample std dev': ky_edu_df.std(ddof=1),
    '25th percentile': ky_edu_df.quantile(q=0.25),
    '50th percentile': ky_edu_df.quantile(q=0.5),
    '75th percentile': ky_edu_df.quantile(q=0.75),
})

# Calculate and define dataframe for Georgia statistics
ga_stats = pd.DataFrame({
    'mode': ga_edu_df.mode().iloc[0],
    'median': ga_edu_df.median(),
    'mean': ga_edu_df.mean(),
    'range': ga_edu_df.max() - ga_edu_df.min(),
    'sample std dev': ga_edu_df.std(ddof=1),
    '25th percentile': ga_edu_df.quantile(q=0.25),
    '50th percentile': ga_edu_df.quantile(q=0.5),
    '75th percentile': ga_edu_df.quantile(q=0.75),
})

# Downloads CSV file with Kentucky education descriptive statistics
ky_stats.to_csv('/Users/cam/Downloads/Kentucky_Descriptive_Statistics.csv')

# Downloads CSV file with Georgia education descriptive statistics
ga_stats.to_csv('/Users/cam/Downloads/Georgia_Descriptive_Statistics.csv')

""" Using the generated CSV files, I have attached the PDF file with both states statistics in table form"""
```

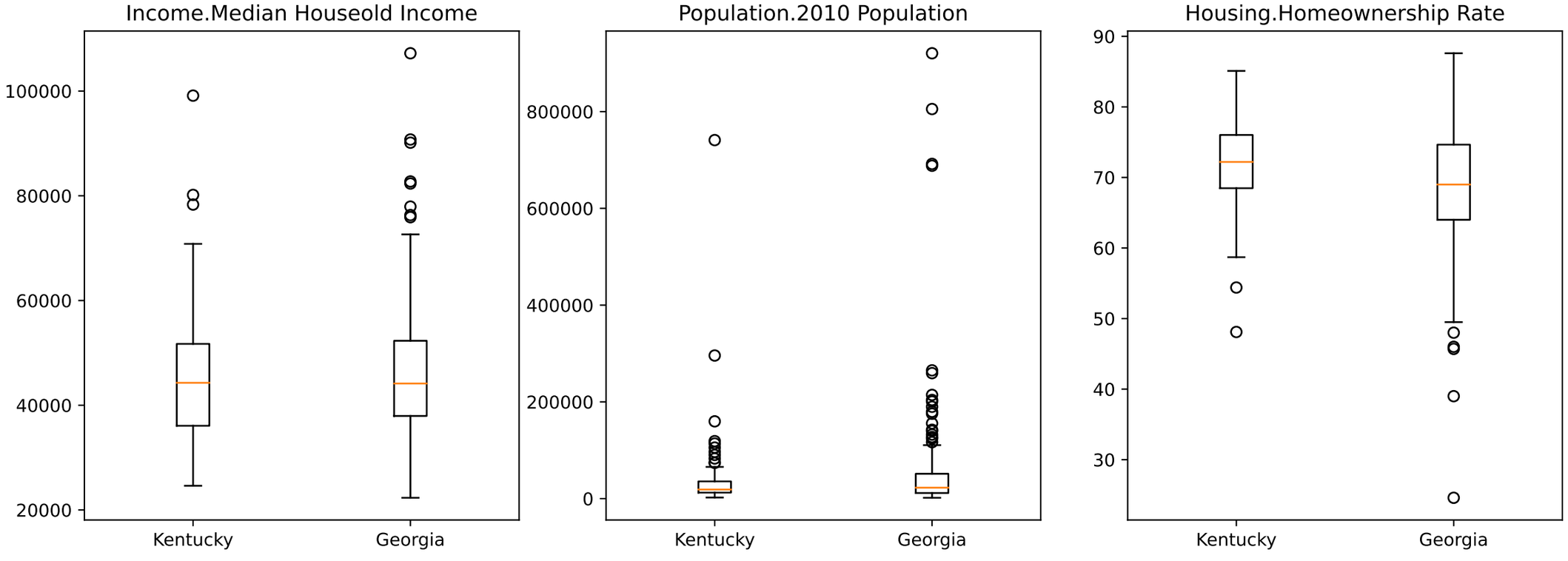
Question 2 - Output:

```
/Users/cam/PycharmProjects/pythonProject4/venv/bin/python
/Applications/PyCharm.app/Contents/plugins/python/helpers/pydev/pydevconsole.py --mode=client --host=127.0.0.1
--port=63779

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/Users/cam/PycharmProjects/pythonProject4'])

Kentucky Statistics:
Education.Bachelor's Degree or Higher: mode = [13.2], median = 14.45, mean = 16.545833333333334,
range = 36.4, std = 7.097163545541054, quantiles = [12.25  14.45  18.575]
Education.HighSchoolDegreeOnly: mode = [62.6 64.3 65.7 67.5 68.5], median = 66.15, mean = 65.54833333333333,
range = 31.300000000000004, std = 5.253298017057705, quantiles = [62.475 66.15  69.425]
Education.NoSchoolDegree: mode = [16.  16.2 17.9 20.4 24.4], median = 17.200000000000003, mean =
17.90583333333333, range = 29.099999999999994, std = 6.2042584431285315, quantiles = [12.875 17.2   23.1  ]

Georgia Statistics:
Education.Bachelor's Degree or Higher: mode = [12.2 13. ], median = 15.8, mean = 18.629559748427674, range =
46.0, std = 9.28000813958558, quantiles = [12.35 15.8  22.55]
Education.HighSchoolDegreeOnly: mode = [62.5], median = 65.0, mean = 63.8440251572327, range =
34.599999999999994, std = 6.170884998948626, quantiles = [61.45 65.   67.55]
Education.NoSchoolDegree: mode = [16.6], median = 17.5, mean = 17.526415094339622, range = 30.599999999999994,
std = 5.9546647263990735, quantiles = [13.25 17.5  20.9 ]
```

Question 3:

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('/Users/cam/Downloads/Updated_KY_GA_Demographics.csv')

# Filter data for Kentucky and Georgia
kentucky = data[data["State"] == "KY"]
georgia = data[data["State"] == "GA"]

# Define the columns to compare in boxplots
cols = ["Income.Median Houseold Income", "Population.2010 Population", "Housing.Homeownership Rate"]

# Create figure with subplots so that we can simplify code and get all boxplots at once using a function
fig, axs = plt.subplots(nrows=1, ncols=len(cols), figsize=(15, 5))

# Create boxplots for each column comparing Kentucky and Georgia, This function is setting each
# axis of the boxplot for each column
for i, col in enumerate(cols):
    axs[i].boxplot([kentucky[col], georgia[col]])
    axs[i].set_xticklabels(["Kentucky", "Georgia"])
    axs[i].set_title(col)

# Download PDF file for the boxplots
plt.savefig("/Users/cam/Downloads/KY_GA_Q3boxplots.pdf")
plt.show()
```

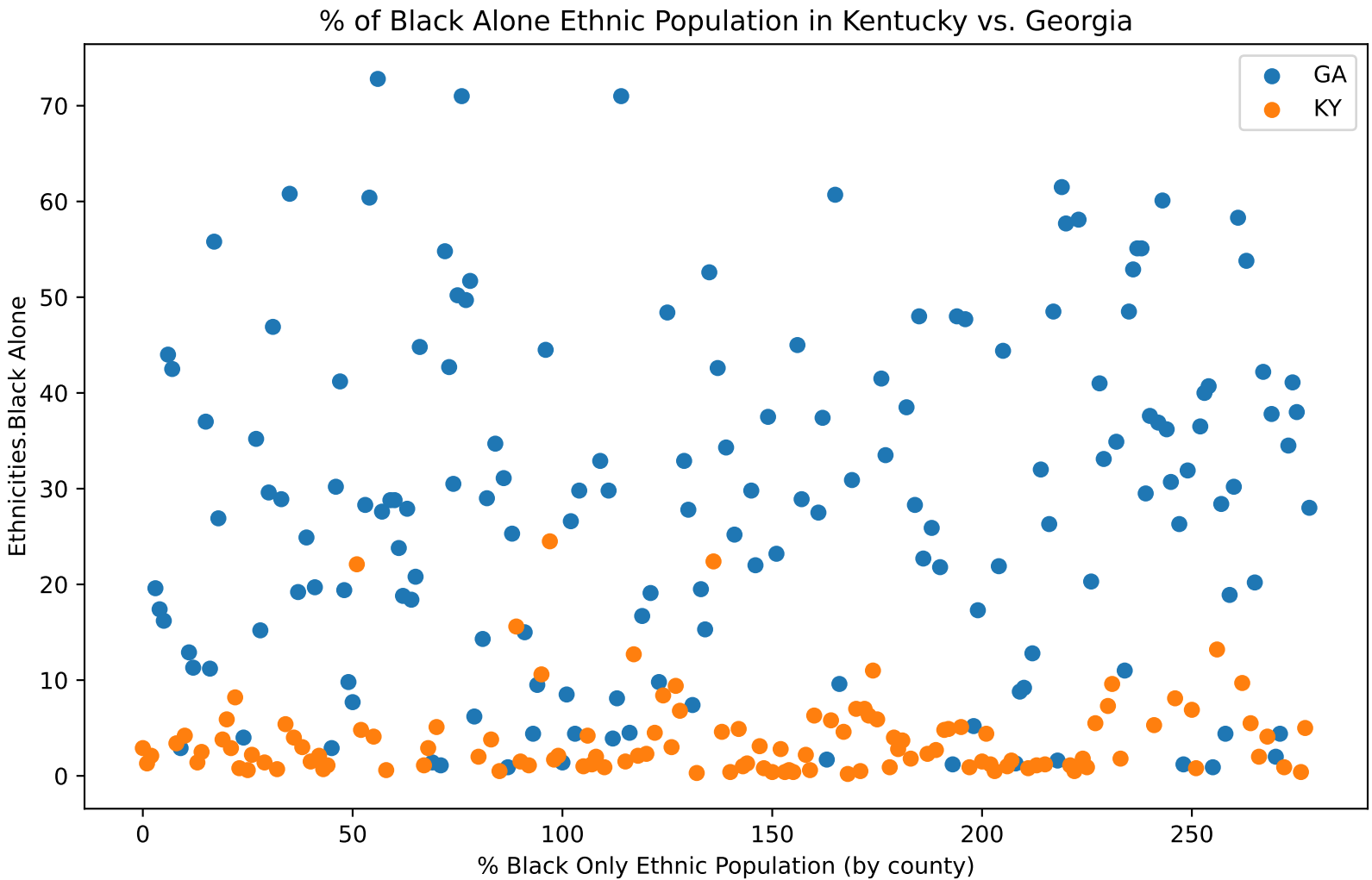
Question 3 - Output: PDF of Boxplots generated by code on next page

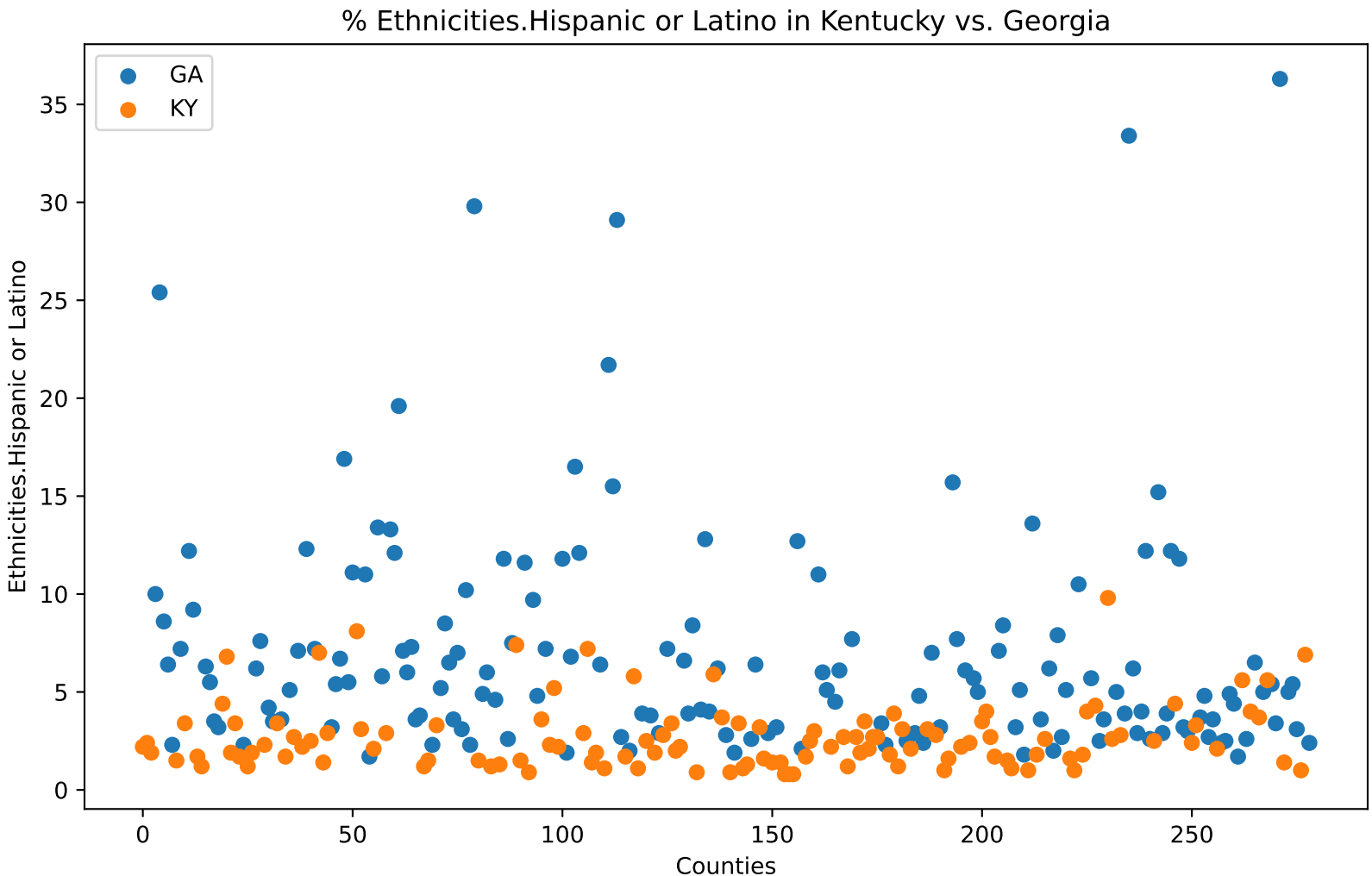

Income.Median Houseold Income
100000
80000
60000
40000
20000
Kentucky
Georgia
Population.2010 Population
800000
600000
400000
200000
0
Kentucky
Georgia
Housing.Homeownership Rate
90
80
70
60
50
40
30
Kentucky
Georgia

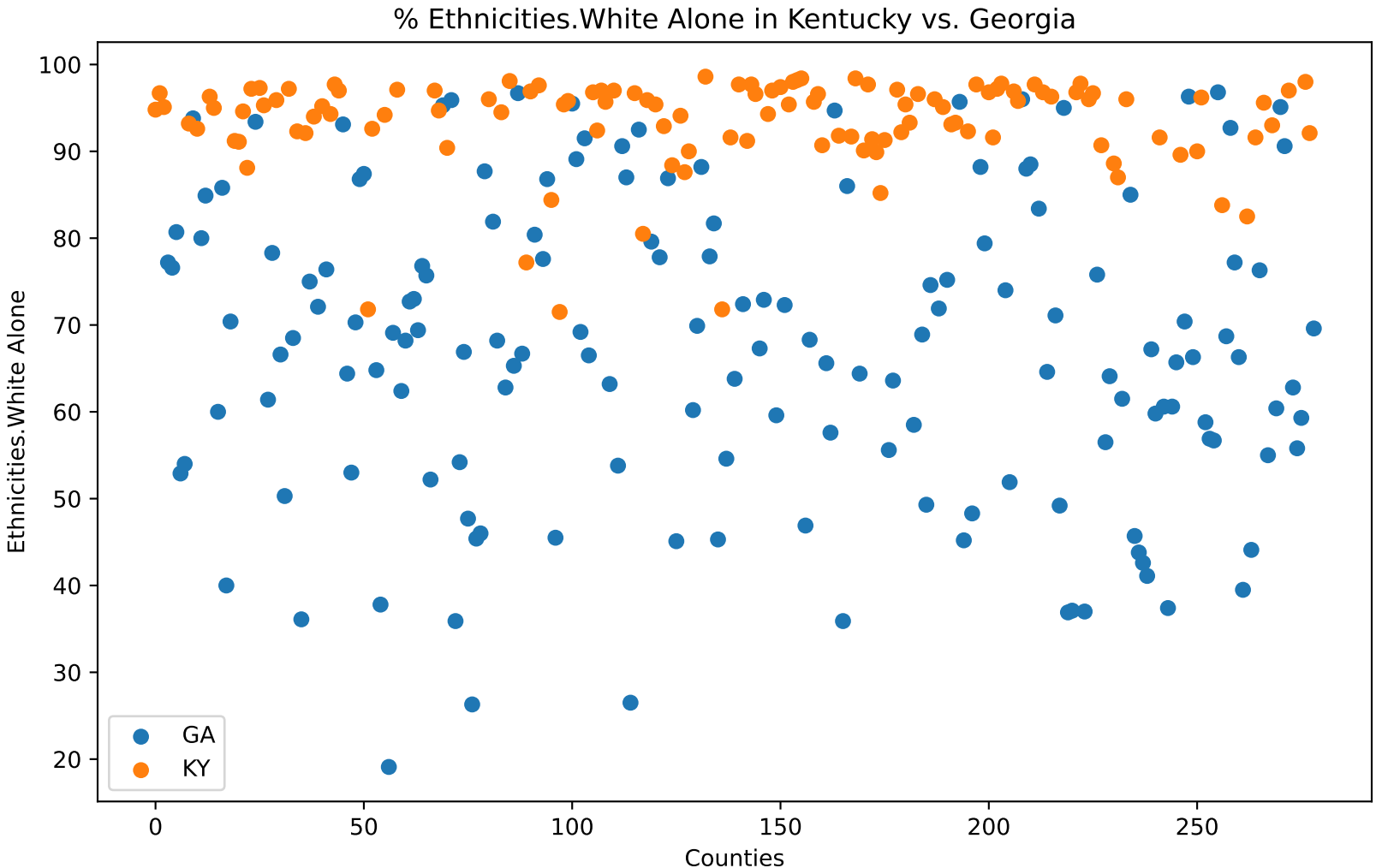
Question 4:

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/Users/cam/Downloads/Updated_KY_GA_Demographics.csv')

# define ethnicities to be compared
ethnicities = ['Ethnicities.Hispanic or Latino', 'Ethnicities.Black Alone', 'Ethnicities.White Alone']

# iterate through each ethnicity and create am individual subplot
for ethnicity in ethnicities:
    fig, ax = plt.subplots(figsize=(10, 6))
    for state, data in df.groupby('State'):
        ax.scatter(data.index, data[ethnicity], label=state)

    # add title and labels and legend
    ax.set_title(f'% {ethnicity} in Kentucky vs. Georgia')
    ax.set_xlabel('Counties')
    ax.set_ylabel(ethnicity)
    ax.legend()

    # Download the scatter plots as a pdfs
    plt.savefig(f'/Users/cam/Downloads/{ethnicity}_scatter_plot.pdf')
    plt.show()
```

Question 4 - Output: PDF of Scatter Plots generated by code on next page


% of Black Alone Ethnic Population in Kentucky vs. Georgia
GA
KY
Ethnicities.Black Alone
0
10
20
30
40
50
60
70
0
50
100
150
200
250
% Black Only Ethnic Population (by county)


% Ethnicities.Hispanic or Latino in Kentucky vs. Georgia
GA
KY
Ethnicities.Hispanic or Latino
Counties
0
5
10
15
20
25
30
35
0
50
100
150
200
250


% Ethnicities.White Alone in Kentucky vs. Georgia
Ethnicities.White Alone
Counties
GA
KY
100
90
80
70
60
50
40
30
20
0
50
100
150
200
250