In [40]:
```
# Enter your name here: Yafeat Adugna
# Enter your Student ID number: G01228070
```

In [41]:
```
# Run cell to import libraries
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
import contextily
import mapclassify
import folium
import aiohttp
import fsspec

# This loads geodataframe contaning county geometry shapes
c = "https://github.com/babdelfa/gis/blob/main/counties_geometry.zip?raw=true"
import fsspec
with fsspec.open(c) as file:
    county_shapes = gpd.read_file(file)

# This loads the most recent covid19 data from Johns Hopkins University's Github
url_cases = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse
df_cases = pd.read_csv(url_cases)
url_deaths = "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_
df_deaths = pd.read_csv(url_deaths)

```

FINAL CODE CONSOLIDATED

Except for the code above, include all your code below inside the single code cell. Run the code cell to display the output.
Do **not** include additional cells.

In [42]:

```
# Project code & output

name = (input ("Enter your name here: "))
print("Hello: ", name)
print("")
print("MIS 433 Project")
print("By: ", name)
print("GMU ID: G12121212")
print("")


print("General data information:")
print("        Topic: COVID-19 reported cases & death")
print("        Data source: Johns Hopkins University")

#date should be a code using the max method and enter the variable there
recent_date = df_cases.iloc[:, -1].name

print("        Time period:", "03/09/23")
print("")
#there should be coude here as a variable to use for total cases(before you merge, cases and death)

#filtering the total cases
total_cases = df_cases.iloc[:,-1].sum()
total_death = df_deaths.iloc[:,-1].sum()

print("USA OVERVIEW")
print("        - Total reported cases:", total_cases )
print("        - Total reported deaths:", total_death )

print("")

#State Input
print(name,", enter a state name for more COVID-19 information:")
state_name = (input (">>>>>>Enter state:"))
print("Found information for", state_name)
print("")
print(state_name, "OVERVIEW")
state_cases = df_cases[df_cases.Province_State == state_name].iloc[:,-1].sum()
state_death = df_deaths[df_deaths.Province_State == state_name].iloc[:, -1].sum()
print("        - Total reported cases:", state_cases)
print("        - Total reported deaths:", state_death)

```

```
print("")

print("Below is an interactive map of COVID-19 cases and deaths for", state_name, "counties.")


#Subset the data columns of interest from df_cases
df_cases.rename(columns={"3/9/23" : "cases"}, inplace=True)
df_cases = df_cases[['UID', 'iso2','FIPS', 'Admin2', 'Province_State', 'Lat', 'Long_', 'cases']].copy
df_cases.rename(columns={"3/9/23" : "cases"}, inplace=True)

#Subset the data columns of interest from df_deaths
df_deaths.rename(columns={"3/9/23" : "death"}, inplace=True)
df_deaths = df_deaths[['UID', 'iso2','FIPS', 'Admin2', 'Province_State', 'Lat', 'Long_', 'death']].co
df_deaths.rename(columns={"3/9/23" : "death"}, inplace=True)

#merged cases and death dataframe
cases_death_merge = df_cases.merge(df_deaths, how = "left", on = ["UID","iso2","FIPS","Admin2","Provi
                                                                  "Long_"])

# Subset the data columns of interest from county_shapes
county_shapes = county_shapes[['FIPS_BEA', 'geometry']].copy()

#merged county_shapes with cases and death using FIPS

gdf = pd.merge(county_shapes, cases_death_merge, left_on = "FIPS_BEA", right_on="FIPS" )

#Renamed Admin2 to County
gdf = gdf.rename(columns = {"Admin2": "County"})

#Integrated with input state name
state = gdf[gdf.Province_State == state_name].copy()
state = state[["County", "cases","death", "geometry"]].copy()
state.explore(column= "cases", cmap="Set2", legend=True, scheme='NaturalBreaks')
```

- Total reported deaths: 101159

Below is an interactive map of COVID-19 cases and deaths for California counties.

Out[42]: