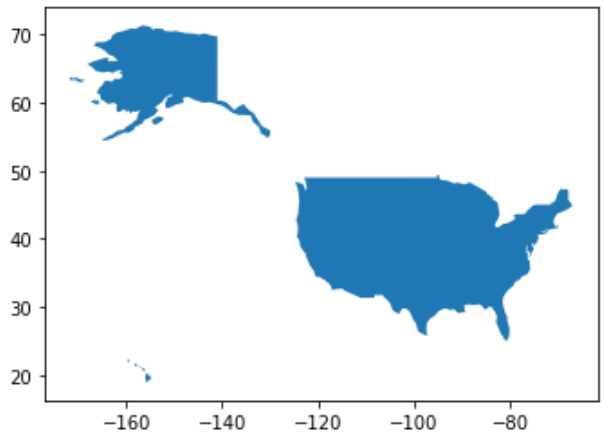
In [ ]:
```
Instructions: Review the instructor video titled Geospatial Analysis
- Part 1 and provide the code that will create
a state-level choropleth map of the United States using COVID-19 cases. Be sur
    #1. include all the states for the choropleth map (do not exclude Alaska a
    #2. include a basemap using the contextily library.
    #3. use mapclassify for the legend display.
    #4. do not display the x and y axis on the map.
    #5. set a cmap argument of your choice.
    #6. provide a figure title that includes your full name.
```

In [2]:
```
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt
import contextily
import mapclassify
import fsspec
pd.set_option('display.max_columns',None)
```

In [9]:
```
df = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/mas
df = df.melt(id_vars=["UID",
                      "iso2",
                      "iso3",
                      "code3",
                      "FIPS",
                      "Admin2",
                      "Province_State",
                      "Country_Region",
                      "Lat",
                      "Long_",
                      "Combined_Key"],
             var_name='date',
             value_name="cases")

df = df[df.iso3 == 'USA'].copy()
df['date'] = pd.to_datetime(df.date)
df = df[df.date == df.date.max()]
df = df[["FIPS","Admin2","Province_State","date","cases","Lat","Long_"]].copy(
df[df.Lat==0]["Admin2"].values
df = df[df.Admin2 != "Unassigned"].copy()
df = df[df.Admin2.str.contains("Out ") == False].copy()
df = df[df.Admin2.str.contains("Correction") ==False].copy()
gdf = gpd.GeoDataFrame(df,
                       crs="EPSG:4326", #format for lang/long data
                       geometry=gpd.points_from_xy(df.Long_, df.Lat))
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
usa = world[world.name == 'United States of America']
usa.plot()
```

Out[9]: <AxesSubplot:>

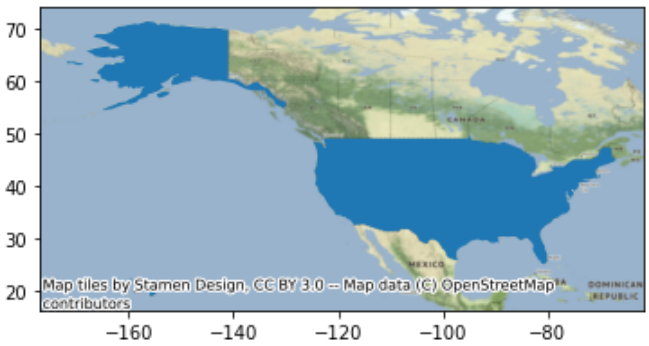


In [5]:
```
df.tail()
```

Out[5]:

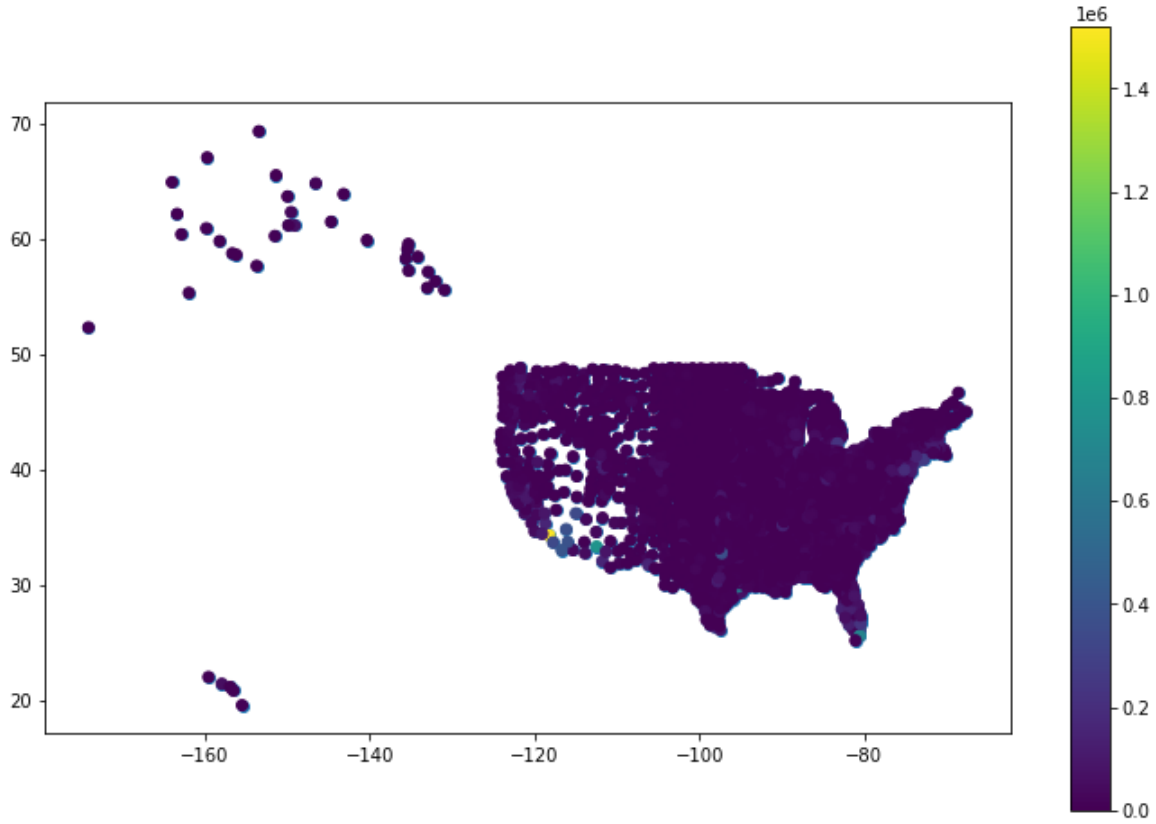
	FIPS	Admin2	Province_State	date	cases	Lat	Long_	geometry
2242476	56037.0	Sweetwater	Wyoming	2021-11-22	8049	41.659439	-108.882788	POINT (-108.88279 41.65944)
2242477	56039.0	Teton	Wyoming	2021-11-22	5362	43.935225	-110.589080	POINT (-110.58908 43.93522)
2242478	56041.0	Uinta	Wyoming	2021-11-22	4006	41.287818	-110.547578	POINT (-110.54758 41.28782)
2242480	56043.0	Washakie	Wyoming	2021-11-22	1831	43.904516	-107.680187	POINT (-107.68019 43.90452)
2242481	56045.0	Weston	Wyoming	2021-11-22	1188	43.839612	-104.567488	POINT (-104.56749 43.83961)

In [31]:
```
fig, ax = plt.subplots()
usa.plot(ax=ax)
contextily.add_basemap(ax, crs = 4326)
contextily.add_basemap(ax, crs=usa.crs.to_string())
```



In [6]:
```
zipfile = "zip:///Users//popoleo//Desktop//state_geometry.zip"
state_geometry = gpd.read_file(zipfile)
#usa = usa.to_crs(epsg=4326)
```

In [10]:
```
#create the layered map using contextility
#covid cases
fig, ax = plt.subplots(figsize=(12,8))
gdf.plot(ax=ax)
gdf.plot(column='cases', ax=ax, legend=True)
plt.show()
```



In [13]:
```
print(state_geometry.columns)
print(gdf.columns)
```

```
Index(['pop_est', 'continent', 'name', 'iso_a3', 'gdp_md_est', 'geometry'], dtyp
e='object')
Index(['FIPS', 'Admin2', 'STATE_NAME', 'date', 'cases', 'Lat', 'Long_'], dtype
='object')
```

In [ ]:
```

```

In [12]:
```
gdf.rename(columns={"Province_State" :"STATE_NAME"}, inplace=True)
gdf.drop("geometry", axis=1, inplace=True)
gdf.head()
```

Out[12]:

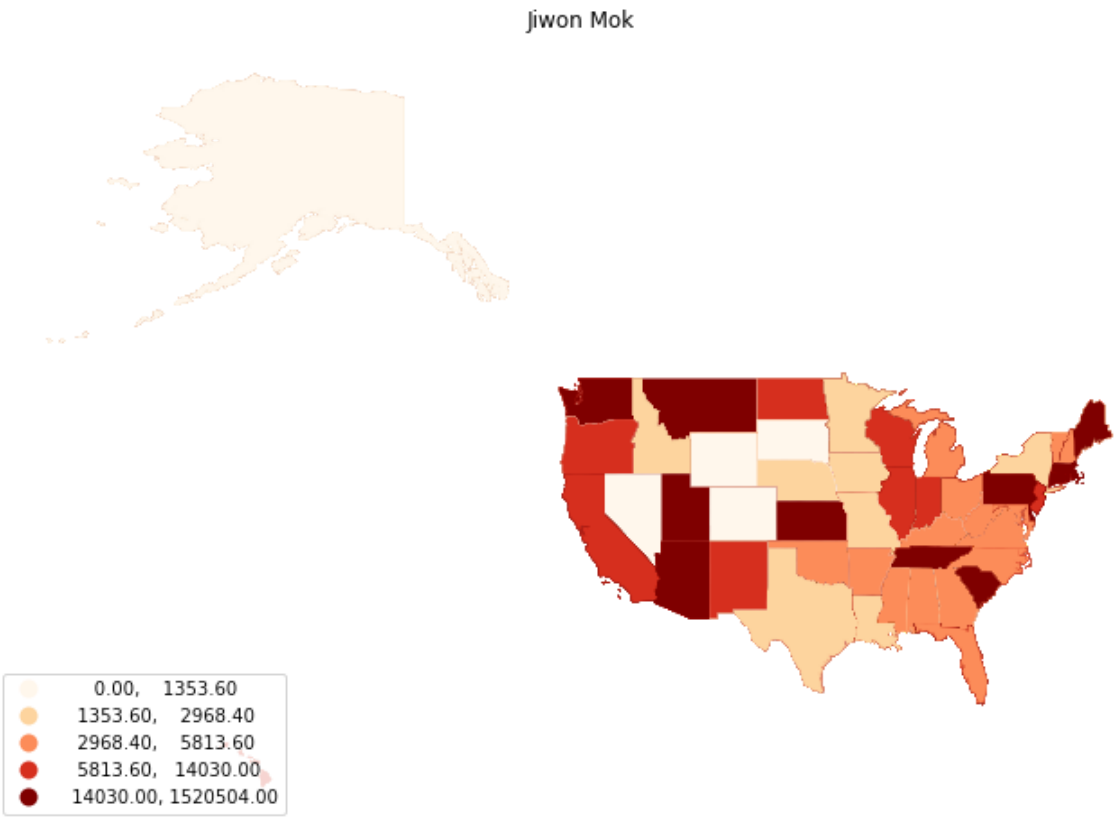
	FIPS	Admin2	STATE_NAME	date	cases	Lat	Long_
2239140	1001.0	Autauga	Alabama	2021-11-22	10477	32.539527	-86.644082
2239141	1003.0	Baldwin	Alabama	2021-11-22	38000	30.727750	-87.722071
2239142	1005.0	Barbour	Alabama	2021-11-22	3688	31.868263	-85.387129
2239143	1007.0	Bibb	Alabama	2021-11-22	4337	32.996421	-87.125115
2239144	1009.0	Blount	Alabama	2021-11-22	10640	33.982109	-86.567906

In [44]:
```
g = gdf.groupby(["STATE_NAME"], as_index=False)['cases'].sum()
```

In [19]:
```
gdf_merged = pd.merge(state_geometry, gdf, on="STATE_NAME")
gdf_merged.plot(column='cases', legend=True)
```

Out[19]: <AxesSubplot:>

In [22]:
```
import mapclassify
gdf_merged.plot(column='cases',
          scheme='quantiles',
          cmap='OrRd',
          legend=True,
          legend_kwds={'loc': 'lower left'},
          figsize=(12,8))
plt.axis("off")
plt.title("Jiwon Mok")
plt.show()
```



In [ ]:
```
1
```