assignment_two

March 4, 2023

0.1 Timothy Miller

0.2 GTECH 73100, Dr. Sun

1 Assignment Two

Further basics

1.0.1 Import modules

[ ]:
```
import json
import io
import time
```

1.0.2 Task 1

Read in a data file of all counties in the US.
Make a list of unique county names

[ ]:
```
# Data Source https://eric.clst.org/assets/wiki/uploads/Stuff/
 ↪gz_2010_us_050_00_5m.json
# address utf-8 error https://stackoverflow.com/questions/30996289/
 ↪utf8-codec-cant-decode-byte-0xf3
# Use "with ... as ..." to better handle exceptions
with io.open('data/gz_2010_us_050_00_5m.json', encoding='latin-1') as f:
  data = json.load(f)

print('Number of Counties in US: {}'.format(len(data['features'])))

print(data['features'][1]['properties']['NAME'])

features = data['features']

unique_county_names = set()
for feature in features:
  properties = feature['properties']
  county_name = properties["NAME"]
  unique_county_names.add(county_name)
```

```
unique_county_names = list(unique_county_names)
print(f"unique {type(unique_county_names)} of county names:␣
 ↪{unique_county_names}")
```

```
Number of Counties in US: 3221
Wade Hampton
unique <class 'list'> of county names: ['Dinwiddie', 'Warren', 'Yell',
'Kittson', 'Seneca', 'Idaho', 'Sumner', 'Camas', 'Bronx', 'Morrow', 'Harding',
'Hardy', 'Tulare', 'Atkinson', 'Salinas', 'Dimmit', 'Leflore', 'McCracken',
'Pottawattamie', 'Billings', "Queen Anne's", 'St. James', 'Atlantic', 'Bryan',
'Pitkin', 'Wallace', 'Sauk', 'Chowan', 'Durham', 'LaSalle', 'Garden',
'Allamakee', 'Tulsa', 'Sanpete', 'Florence', 'Waukesha', 'Worth', 'Bossier',
'McCone', 'Mellette', 'Ionia', 'Chester', 'Blount', 'Titus', 'Outagamie',
'Niobrara', 'McDonald', 'Alcona', 'Lexington', 'Carolina', 'Klickitat',
'Cullman', 'Hopkins', 'Hanson', 'Wichita', 'Red River', 'Santa Rosa', 'Hardee',
'Cheatham', 'Cook', 'Codington', 'Caswell', 'Story', 'Kosciusko', 'Kenton',
'Beaverhead', 'Schenectady', 'Socorro', 'Dawson', 'Divide', 'Kiowa',
'Pasquotank', 'Contra Costa', 'Foster', 'Beltrami', 'Lassen', 'De Soto',
'Cimarron', 'Glacier', 'Bell', 'Nance', 'Clallam', 'Republic', 'Oktibbeha',
'Guánica', 'Nome', 'Guernsey', 'Darke', 'Fort Bend', 'Plaquemines', 'Matagorda',
'Alfalfa', 'Buckingham', 'Tucker', 'Ellsworth', 'Canóvanas', 'Barber',
'Vieques', 'Los Angeles', 'Guilford', 'Kennebec', 'Faulk', 'Newport', 'Dewey',
'Kern', 'Barranquitas', 'Rockland', 'Limestone', 'Barry', 'Hillsborough', 'Mille
Lacs', 'King and Queen', 'Naguabo', 'Beauregard', 'Hartley', 'Poquoson',
'Fayette', 'Ingham', 'Colquitt', 'Chicot', 'Pender', 'Darlington', 'Lafayette',
'Liberty', 'Mesa', 'Pine', 'Clarion', 'Northumberland', 'Iron', 'Lenawee',
'Dundy', 'Toa Baja', 'Nueces', 'Payette', 'Jones', 'Ochiltree', 'Greenup',
'Greensville', 'Bandera', 'Chemung', 'Essex', 'Ransom', 'Doña Ana', 'Placer',
'Fallon', 'Canadian', 'Faulkner', 'Cochise', 'Rusk', 'Norman', 'West Carroll',
'Berkshire', 'Olmsted', 'Wilkes', 'De Witt', 'Glynn', 'Cayey', 'Owen',
'Oklahoma', 'Grand Isle', 'Abbeville', 'Palo Pinto', 'Trinity', 'Anchorage',
'Cooper', 'Burke', 'Santa Clara', 'Shannon', 'Watauga', 'Vilas', 'Gillespie',
'Letcher', 'Island', 'Hennepin', 'Judith Basin', 'West Baton Rouge',
'Lancaster', 'Henderson', 'Wheatland', 'Broomfield', 'Bedford', 'Dakota', 'White
Pine', 'Del Norte', 'Vigo', 'Suffolk', 'Woodford', 'Jersey', 'Amelia',
'Pleasants', 'Sedgwick', 'Creek', 'Benton', 'Keya Paha', 'Graham',
'Northampton', 'Chippewa', 'Genesee', 'Aroostook', 'Houston', 'Grimes',
'Sierra', 'Luzerne', 'Asotin', 'Dearborn', 'Brantley', 'Nolan', 'Sharkey',
'Leelanau', 'Trego', 'Ouachita', 'Blaine', 'Barren', 'Riverside', 'Boise',
'Mariposa', 'Kenosha', 'San Miguel', 'Bear Lake', 'Goliad', 'Manitowoc',
'Finney', 'Manatee', 'Edmunds', 'Guayama', 'Stokes', 'Marion', 'Pasco',
'Volusia', 'Kalkaska', 'Hertford', 'Hamilton', 'Nowata', 'Cibola', 'Whitfield',
'Waupaca', 'Kodiak Island', 'Angelina', 'Pierce', 'Morgan', 'Rhea', 'Auglaize',
'Ashland', 'Charlotte', 'Coal', 'Norton', 'Hartford', 'Switzerland', 'Santa
Barbara', 'San Patricio', 'Texas', 'Zavala', 'Sitka', 'Sandusky', 'Chilton',
'Clear Creek', 'Modoc', 'Unicoi', 'St. Francis', 'Archuleta', 'Fall River',
'Yavapai', 'Karnes', 'Rich', 'Shackelford', 'Chenango', 'Adams', 'Lubbock',
'Tuolumne', 'Paulding', 'Lowndes', 'Montgomery', 'Lake of the Woods',
```

'Snohomish', 'Craven', 'Fulton', 'Onondaga', 'Comerío', 'Archer', 'DeWitt',
'Centre', 'Covington', 'Ontonagon', 'Kandiyohi', 'Guadalupe', 'Burleigh',
'Renville', 'Chaves', 'Presidio', 'Grundy', 'Westchester', 'Pickett', 'Ogemaw',
'Wise', 'Arkansas', 'Sumter', 'McHenry', 'Ida', 'Kings', 'Bayfield', 'Garrett',
'Lauderdale', 'Hettinger', 'Brooke', 'Trujillo Alto', 'East Feliciana',
'Pembina', 'DeSoto', 'Appanoose', 'Atascosa', 'Green Lake', 'Baylor', 'Douglas',
'Benewah', 'Whatcom', 'Marinette', 'Jasper', 'Meade', 'Cottle', 'Mineral',
'Schoharie', 'Garvin', 'Henry', 'Mississippi', 'Tensas', 'Pend Oreille',
'Mecklenburg', 'Perquimans', 'Reagan', 'Forest', 'Skagit', 'Iberia', 'Seminole',
'Emanuel', 'Audrain', 'Skagway', 'Starke', 'Hamblen', 'Wabaunsee', 'Merced',
'Okmulgee', 'Laurens', 'Burt', 'Monroe', 'Fairfield', 'Lenoir', 'Daviess',
'Cerro Gordo', 'Hunt', 'Sagadahoc', 'Naranjito', 'Elk', 'Clarke', 'Rawlins',
'Tangipahoa', 'Staunton', 'Garza', 'Kent', 'Orocovis', 'Maunabo', 'Macon', 'St.
Joseph', 'Washita', 'Izard', 'Hernando', 'Osborne', 'Eaton', 'Highland',
'Irion', 'Guthrie', 'Pawnee', 'Nobles', 'Morovis', 'Sanders', 'Overton',
'Botetourt', 'Snyder', 'Susquehanna', 'Ponce', 'Peach', 'Lavaca', 'Goshen',
'Dauphin', 'Gosper', 'Elmore', 'Val Verde', 'Prince of Wales-Hyder', "O'Brien",
'Flagler', 'Navajo', 'Crane', 'Gray', 'East Carroll', 'Crockett', 'Logan',
'Roberts', 'Schleicher', 'Storey', 'Giles', 'Hays', 'Livingston', 'Woodward',
'Sanilac', 'Clackamas', 'Shasta', 'Hand', 'Banks', 'Kingman', 'McNairy', 'Jay',
'Swisher', 'Trimble', 'Cumberland', 'Coweta', 'Meeker', 'Jewell', 'Miner',
'Granville', 'Alexandria', 'Grant', 'Grand Forks', 'Lapeer', 'Apache', 'Taney',
'Marshall', 'Rolette', 'Roane', 'Bonner', 'Caldwell', 'San Lorenzo', 'Cuyahoga',
'Candler', 'King', 'Yuma', 'Lee', 'Macoupin', 'Walla Walla', 'Salem', 'Edwards',
'Benzie', 'Lawrence', 'Major', 'Huntingdon', 'Dubuque', 'Cowlitz', 'Cass',
'Baldwin', 'Barnstable', 'Hall', 'Cloud', 'Menifee', 'Pipestone', 'Issaquena',
'Bullitt', 'Vermillion', 'Union', 'Lane', 'Mason', 'District of Columbia',
'Uvalde', 'Itasca', 'Bourbon', 'Glades', 'Pike', 'Laclede', 'Mahoning',
'Metcalfe', 'Marathon', 'St. Louis', 'Wetzel', 'Morrison', 'Neshoba',
'Nacogdoches', 'Gregory', 'Bertie', 'Braxton', 'Bent', 'Menominee', 'St. Croix',
'Oswego', 'Walker', 'Potter', 'Charles Mix', 'Stanley', 'Matanuska-Susitna',
'Lunenburg', 'Towns', 'Clearwater', 'Saluda', 'Upson', 'Eureka', 'Cleburne',
'Park', 'Concho', 'Bond', 'Lebanon', 'Leavenworth', 'St. Tammany', 'DeKalb',
'Stark', 'Coles', 'Cavalier', 'Waynesboro', 'Coryell', 'Bennington', 'Loudon',
'Caribou', 'Collingsworth', 'Summit', 'Ascension', 'Jeff Davis', 'Alpine',
'Smyth', 'Collin', 'Sharp', 'Putnam', 'Umatilla', 'Edgecombe', 'Southampton',
'Des Moines', 'Throckmorton', 'Wells', 'Tuscaloosa', 'Ottawa', 'Piscataquis',
'Bristol Bay', 'Washoe', 'Doniphan', 'Geary', 'Traverse', 'Sutter', 'Walsh',
'Charles City', 'Larue', 'Wagoner', 'Winneshiek', 'New Madrid', 'Heard', 'Lake
and Peninsula', 'Bergen', 'Thurston', 'Henrico', 'Wibaux', 'Rice', 'Oconto',
'Sawyer', 'Sandoval', 'West Feliciana', 'Hot Spring', 'Prowers', 'Jefferson
Davis', 'Hendry', 'Pickens', 'Lorain', 'Manistee', 'Okeechobee', 'Prince
William', 'Indian River', 'St. Johns', 'Columbiana', 'Glenn', 'Daggett',
'Perry', 'McKinley', 'Coffee', 'Duval', 'Wright', 'Elliott', 'Tehama', 'Andrew',
'Halifax', 'Manatí', 'Villalba', 'Daniels', 'Esmeralda', 'Providence', 'St.
Martin', 'McClain', 'Macomb', 'Deaf Smith', 'Alachua', 'Solano', 'Martin',
'Sussex', 'Spink', 'George', 'Litchfield', 'Greenwood', 'Dare', 'Emporia',
'Cabarrus', 'Howard', 'Guaynabo', 'Monona', 'Stone', 'Converse', 'Mohave',

'Grainger', 'Churchill', 'Pima', 'Accomack', 'Coleman', 'Lamar', 'Davidson', 'Waller', 'San Juan', 'Gallatin', 'Moniteau', 'Sheridan', 'Bowman', 'Sweet Grass', 'Preble', 'Lanier', 'Childress', 'Hatillo', 'Panola', 'Rooks', 'Fentress', 'Gulf', 'Mower', 'Fergus', 'Buffalo', 'Shawnee', 'Portsmouth', 'Starr', 'Grand', 'Saline', 'Wade Hampton', 'Weld', 'Pondera', 'Guayanilla', 'Rains', 'Koochiching', 'Red Willow', 'Milwaukee', 'Santa Isabel', 'Harrisonburg', 'Shawano', 'Fredericksburg', 'Jo Daviess', 'Gregg', 'Gogebic', 'Juncos', 'Catahoula', 'Honolulu', 'Onslow', 'Mills', 'Love', 'Glasscock', 'San Benito', 'Forrest', 'Gasconade', 'Nantucket', 'Saginaw', 'Echols', 'Breckinridge', 'Okfuskee', 'Cannon', 'Faribault', 'Fannin', 'McDowell', 'Alexander', 'Elkhart', 'Manassas Park', 'Highlands', 'Holt', 'Broome', 'Oconee', 'Falls', 'Hitchcock', 'Redwood', 'Clark', 'Lewis', 'Dukes', 'Miami', 'Lynn', 'Vega Alta', 'Cabo Rojo', 'Oglethorpe', 'Spencer', 'Morehouse', 'Teller', 'Hood River', 'Natrona', 'Aiken', 'McCurtain', 'Chase', 'Brevard', 'Bulloch', 'El Dorado', 'Millard', 'Tate', 'Irwin', 'Hampshire', 'Decatur', 'Freeborn', 'Antelope', 'Ulster', 'LaGrange', 'Minidoka', 'King George', 'Burlington', 'Beaufort', 'Cooke', 'Coahoma', 'Johnson', 'La Crosse', 'Bland', 'Ravalli', 'Castro', 'Collier', 'Williams', 'Gratiot', 'Blanco', 'Dyer', 'Mora', 'Jackson', 'Hancock', 'Clare', 'Denver', 'Northwest Arctic', 'San Augustine', 'Dixie', 'Grady', 'Petroleum', 'Calhoun', 'Sarpy', 'Crisp', 'Baca', 'Washburn', 'Woodson', 'Hot Springs', 'Fajardo', 'Ward', 'Sarasota', 'Real', 'Florida', 'Wabash', 'Houghton', 'Maury', 'Bonneville', 'New Castle', 'Bacon', 'Frio', 'Roosevelt', 'Wayne', 'Etowah', 'Passaic', 'Valdez-Cordova', 'Walworth', 'Cheboygan', 'Tuscola', 'Harford', 'Claiborne', 'Cayuga', 'Kerr', 'Caguas', 'Plumas', 'Saunders', 'Rio Arriba', 'Lampasas', 'Hyde', 'Thomas', 'York', 'Otsego', 'Otoe', 'Steele', 'Gilchrist', 'Elbert', 'Magoffin', 'Kendall', 'Multnomah', 'Montour', 'Quebradillas', 'Canyon', 'Miller', 'Tallapoosa', 'Steuben', 'Hidalgo', 'Gove', 'Calaveras', 'St. Lucie', 'Blue Earth', 'Swain', 'Fillmore', 'Neosho', 'Corson', 'Lake', 'Sangamon', 'Dodge', 'Cape May', 'Valley', 'Windsor', 'Arenac', 'Rio Grande', 'Chattooga', 'Reeves', 'Latimer', 'Orangeburg', 'Madison', 'Turner', 'Curry', 'Kidder', 'Venango', 'Fountain', 'Ashley', 'Davis', 'St. Mary', 'Mingo', 'Charlevoix', 'Santa Cruz', 'Minnehaha', 'Hill', "Prince George's", 'Bristol', 'Columbia', 'Johnston', 'Maui', 'Hickman', 'Newton', 'Spotsylvania', 'Edgar', 'Albany', 'Luna', 'Page', 'Will', 'Camden', 'Falls Church', 'Loíza', "St. Mary's", 'Sebastian', 'Wake', 'Posey', 'Stephens', 'Charleston', 'Owyhee', 'Waldo', 'Wolfe', 'Prairie', 'Noxubee', 'Crittenden', 'Duplin', 'Beaver', 'Craig', 'Bamberg', 'Carteret', 'Williamsburg', 'Napa', 'Kingsbury', 'Obion', 'Indiana', 'Bolivar', 'Marengo', 'Barron', 'Medina', 'Surry', 'Barnwell', 'McLeod', 'Winnebago', 'Hodgeman', 'Lamoille', 'Bernalillo', 'Doddridge', 'Bradford', 'Tooele', 'Bethel', 'Callahan', 'Kearney', 'Tolland', 'Oregon', 'Chickasaw', 'Yukon-Koyukuk', 'Greeley', 'Las Marías', 'Tillman', 'Kalawao', 'Sherburne', 'Sevier', 'Monterey', 'Yadkin', 'Wilbarger', 'Nicollet', 'Kimble', 'Bollinger', 'Grayson', 'Morton', 'Belknap', 'Mackinac', 'Young', 'Parmer', 'Dutchess', 'Stanly', 'Culpeper', 'Iroquois', 'Sully', 'Haakon', 'Cherry', 'Bates', 'Wasco', 'Iberville', 'Hopewell', 'Nash', 'Juniata', 'Erie', 'Boone', 'Leake', 'Yolo', 'Arecibo', 'Scotland', 'Pueblo', 'New York', 'Ashtabula', 'Washington', 'Copiah', 'Adair', 'McDonough', 'Dorado', 'Culberson', 'Hinds', 'Caroline', 'Sacramento', 'Horry', 'Denali', 'Escambia',

'Huntington', 'Colonial Heights', 'Hayes', 'New Haven', 'Laramie', 'Stonewall',
'Webster', 'Refugio', 'Cecil', 'Las Animas', 'Kankakee', 'Piute', 'Kimball',
'Maricopa', 'Van Wert', 'Hood', 'Cross', 'Kit Carson', 'Live Oak', 'Josephine',
'Cuming', 'Hickory', 'Geneva', 'Stutsman', 'Lajas', 'Ector', 'Ceiba',
'Randolph', 'Rio Blanco', 'McCulloch', 'Russell', 'Worcester', 'Cherokee',
'Bowie', 'San Jacinto', 'Cottonwood', 'Allen', 'Duchesne', 'Kewaunee', 'Scotts
Bluff', 'Lumpkin', 'Pinellas', 'Winona', 'Allendale', 'Austin', 'Barbour',
'Kanawha', 'Gurabo', 'Merrick', 'Dillon', 'Edmonson', 'Schuylkill', 'Yakima',
'Golden Valley', 'Phillips', 'Llano', 'Camuy', 'Galax', 'Amador', 'McKean',
'Gilliam', 'Jack', 'Sunflower', 'Yellow Medicine', 'Antrim', 'Stanislaus',
'Richmond', 'Peoria', 'Tippecanoe', 'Emery', 'Eau Claire', 'Van Zandt', 'Pepin',
'Little River', 'Ness', 'Deschutes', 'Taylor', 'Dallam', 'Toombs', 'Latah',
'Bureau', 'Routt', 'Bullock', 'Wyandotte', 'Gage', 'Day', 'Buchanan', 'Lehigh',
'Bexar', 'Rappahannock', 'Graves', 'Dolores', 'Atchison', 'Slope', 'Grenada',
'Cole', 'Bibb', 'Pennington', 'Kenedy', 'Gilmer', 'Gadsden', 'DuPage',
'Berrien', 'Massac', 'Muskegon', 'Norfolk', 'Toa Alta', 'Callaway', 'Bracken',
'Gordon', 'Nodaway', 'Isle of Wight', 'Fairfax', 'Pettis', 'Borden', 'Tipton',
'Barrow', 'La Salle', 'Riley', 'Rock', 'Goodhue', 'Sampson', 'Chautauqua',
'Hutchinson', 'Burnett', 'Granite', 'Somerset', 'Fairbanks North Star',
'Twiggs', 'Fleming', 'Carlton', 'Alcorn', 'Buena Vista', 'Terry', 'Rosebud',
'Grays Harbor', 'Yazoo', 'Patrick', 'Kitsap', 'Ralls', 'Davie', 'Maricao',
'McCreary', 'Schuyler', 'Roanoke', 'Colleton', 'Stanton', 'Culebra', 'Yamhill',
'Ramsey', 'Screven', 'Luquillo', 'Ware', 'Wadena', 'Treutlen', 'Currituck',
'Custer', 'Roseau', 'Hawkins', 'Victoria', 'Rapides', 'Carter', 'Clayton',
'Grafton', 'San Diego', 'Berkeley', 'Itawamba', 'Evangeline', 'Summers',
'Stafford', 'Costilla', 'Chattahoochee', 'Philadelphia', 'Estill', 'Palm Beach',
'Champaign', 'Parker', 'Ashe', 'Harlan', 'Oscoda', 'Big Stone', 'Stearns',
'Silver Bow', 'Crenshaw', 'Pointe Coupee', 'Rush', 'Sublette', 'Conway',
'Tarrant', 'Utah', 'Cameron', 'Barnes', 'Sequoyah', 'Door', 'Nelson', 'Haines',
'Chittenden', 'Sibley', 'Brookings', 'Sheboygan', 'Bremer', 'Lea', 'Wilkin',
'Montezuma', 'Schley', 'Polk', 'Boulder', 'McPherson', 'Anderson', 'Trousdale',
'Lipscomb', 'Plymouth', 'Brule', 'Fremont', 'Knott', 'Dade', 'Runnels',
'Powhatan', 'Andrews', 'Garland', 'Avery', 'Defiance', 'Wabasha', 'Monongalia',
'Malheur', 'Nemaha', 'Rutland', 'Ontario', 'Allegany', 'Linn', 'Addison',
'Branch', 'Pinal', 'Aguada', 'Gladwin', 'McIntosh', 'Stillwater', 'Merrimack',
'Donley', 'Keweenaw', 'Hubbard', 'Poweshiek', 'Chouteau', 'Morrill', 'Stewart',
'Frederick', 'Wicomico', 'Taliaferro', 'Jefferson', 'Terrell', 'Piatt', 'Coos',
'Iredell', 'Treasure', 'Conejos', 'Nicholas', 'San Mateo', 'Carson City',
'Louisa', 'Pemiscot', 'Fresno', 'Baraga', 'Aguadilla', 'Murray', 'Clearfield',
'Harper', 'Arthur', 'Coosa', 'Catawba', 'Weston', 'Ogle', 'Comal', 'Brooks',
'Cleveland', 'Oldham', 'Pecos', 'Baker', 'Lackawanna', 'Jerome', 'Aibonito',
'Hampden', 'Tama', 'Pratt', 'Herkimer', 'Ciales', 'Alameda', 'Carson',
'Pacific', 'Broward', 'Bartow', 'Vinton', 'St. Landry', 'Leslie', 'Lincoln',
'Yellowstone', 'Huerfano', 'Madera', 'Catoosa', 'Dunklin', 'Racine', 'Bucks',
'Box Butte', 'Chesterfield', 'Muscogee', 'Calcasieu', 'Crosby', 'Deer Lodge',
'Harrison', 'Nevada', 'Tunica', 'San Luis Obispo', 'Lander', 'Hughes',
'Brewster', 'Colfax', 'Albemarle', 'Colusa', 'Dubois', 'Kemper', 'Harmon',
'Nassau', 'Newberry', 'Swift', 'Shelby', 'Zapata', 'El Paso', 'Williamson',

'Larimer', 'Amherst', 'St. Charles', 'Forsyth', 'Noble', 'Somervell', 'Barton', 'Drew', 'Morris', 'Allegheny', 'Eddy', 'Chambers', 'Kenai Peninsula', 'Prince George', 'Butler', 'Santa Fe', 'Dane', 'Blackford', 'Mifflin', 'Uinta', 'Palo Alto', 'Pottawatomie', 'Robertson', 'Emmons', 'Perkins', 'Rincón', 'Rowan', 'Rogers', 'Fisher', 'Armstrong', 'San Sebastián', 'Acadia', 'Ocean', 'Leon', 'LaMoure', 'Haskell', 'Otter Tail', 'Jerauld', 'Hillsdale', 'Clinton', 'Effingham', 'Muscatine', 'Penobscot', 'Utuado', 'Evans', 'Sequatchie', 'Talbot', 'Strafford', 'Prentiss', 'Knox', 'Yalobusha', 'Dorchester', 'Aurora', 'Kingfisher', 'Stoddard', 'Bledsoe', 'Kootenai', 'Klamath', 'Gilpin', 'Windham', 'Sioux', 'Dallas', 'Pershing', 'Edgefield', 'De Baca', 'Midland', 'Ray', 'Elko', 'Wrangell', 'Hampton', 'Licking', 'Muskingum', 'Power', 'Pittsylvania', 'Wood', 'Peñuelas', 'Arapahoe', 'Clermont', 'Wilcox', 'Cidra', 'Dale', 'Mecosta', 'Fluvanna', 'Shiawassee', 'Briscoe', 'Gonzales', 'Lucas', 'Rockbridge', 'Greenlee', 'Whitley', 'Sabana Grande', 'Jennings', 'Ventura', 'Eastland', 'Seward', 'Vernon', 'Aransas', 'Harris', 'Spartanburg', 'Alamosa', 'Bennett', 'Vermilion', 'Maverick', 'Weakley', 'Towner', 'Charles', 'Patillas', 'Wapello', 'Tompkins', 'Quitman', 'Hunterdon', 'Wilkinson', 'Woodbury', 'Trempealeau', 'Coconino', 'Willacy', 'Gooding', 'Marin', 'Mobile', 'Delaware', 'Saratoga', 'Wharton', 'Cassia', 'Walton', 'Rockcastle', 'Mahnomen', 'Aleutians East', 'Meigs', 'Inyo', 'Moca', 'Parke', 'Hardeman', 'Aguas Buenas', 'La Paz', 'Denton', 'Ford', 'Maries', 'Washtenaw', 'Lewis and Clark', 'Raleigh', 'Sanborn', 'Twin Falls', 'Teton', 'Powell', 'Marlboro', 'Bannock', 'Bingham', 'Athens', 'Shenandoah', 'Christian', 'Natchitoches', 'Tazewell', 'Anoka', 'Big Horn', 'Nottoway', 'Frontier', 'Habersham', 'Hinsdale', 'Goochland', 'Hormigueros', 'Freestone', 'Yuba', 'North Slope', 'Cheshire', 'Niagara', 'Tripp', 'Coffey', 'Sweetwater', 'Todd', 'Sac', 'Pendleton', 'McMullen', 'Mono', 'Sargent', 'Clinch', 'Cataño', 'Lac qui Parle', 'Bleckley', 'Ste. Genevieve', 'Humacao', 'Atoka', 'Loudoun', 'Colbert', 'Jenkins', 'McDuffie', 'Owsley', 'Wahkiakum', 'Moore', 'Brazoria', 'Alger', 'Augusta', 'Levy', 'Pocahontas', 'Skamania', 'Spalding', 'Travis', 'Box Elder', 'Chesapeake', 'Brazos', 'Pearl River', 'Mayagüez', 'Beckham', 'Hale', 'Gates', 'Harvey', 'Montmorency', 'Meagher', 'Weber', 'Waseca', 'Orleans', 'Hoke', 'Crawford', 'Pulaski', 'Cache', 'Hempstead', 'Juneau', 'Winston', 'Las Piedras', 'Taos', 'Warrick', 'Scott', 'Bay', 'Osage', 'Bailey', 'Prince Edward', 'Okaloosa', 'Emmet', 'Jim Wells', 'Tyler', 'Ozaukee', 'Mahaska', 'Imperial', 'Greer', 'Rutherford', 'Sterling', 'Wheeler', 'Ouray', 'Amite', 'Ben Hill', 'Cambria', 'Dunn', 'Bottineau', 'Stephenson', 'Burnet', 'Adjuntas', 'Garrard', 'Tift', 'Alpena', 'Lafourche', 'Trumbull', 'Carlisle', 'Concordia', 'Assumption', 'Charlottesville', 'Toole', 'Oliver', 'Rabun', 'Washakie', 'Campbell', 'Sonoma', 'Chariton', 'Missoula', 'Pushmataha', 'Pontotoc', 'Greenville', 'Wyoming', 'Colorado', 'Lycoming', 'Tishomingo', 'Wyandot', 'Alamance', 'Hudson', 'Randall', 'Terrebonne', 'Okanogan', 'Dooly', 'Ringgold', 'Talladega', 'Le Sueur', 'Moody', 'Luce', 'Choctaw', 'Sullivan', 'Gentry', 'Blair', 'Gibson', 'Kinney', 'Comanche', 'Cascade', 'Hocking', 'Black Hawk', 'Carbon', 'Clay', 'St. Bernard', 'Vega Baja', 'Rankin', 'Musselshell', 'Ozark', 'Catron', 'Porter', 'Avoyelles', 'Alleghany', 'Middlesex', 'Phelps', 'Mercer', 'Calloway', 'Crook', 'Miami-Dade', 'Salt Lake', 'Waushara', 'Griggs', 'Rockdale', 'McCormick', 'Robeson', 'Ziebach', 'Conecuh', 'Androscoggin', 'Kane', 'Newport News', 'Breathitt',

'Rockingham', 'Kay', 'White', 'McLean', 'Caddo', 'Thayer', 'Trigg',
'Tallahatchie', 'Bastrop', 'Wallowa', 'Traill', 'Upshur', 'Anne Arundel',
'Yoakum', 'Deuel', 'Schoolcraft', 'Roger Mills', 'Valencia', 'Eagle', 'Ross',
'Richland', 'Charlton', 'Humboldt', 'Columbus', 'Benson', 'Westmoreland',
'Oxford', 'Pittsburg', 'Tyrrell', 'Pamlico', 'Telfair', 'Anson', 'Kearny',
'Jessamine', 'McMinn', 'East Baton Rouge', 'Bradley', 'Ellis', 'Baltimore',
'Rockwall', 'Meriwether', 'Attala', 'Transylvania', 'Scurry', 'Newaygo',
'Glascock', 'Simpson', 'Lyman', 'Virginia Beach', 'Queens', 'Searcy',
'Nuckolls', 'Marquette', 'New Kent', 'Lemhi', 'Portage', 'Geauga', 'St. Clair',
'Hockley', 'Kossuth', 'Sabine', 'Muskogee', 'Green', 'Spokane', 'Jayuya',
'Cotton', 'Iowa', 'Sutton', 'King William', 'Brown', 'Bon Homme', 'Langlade',
'Corozal', 'Bladen', 'Yauco', 'Upton', 'Lamb', 'Southeast Fairbanks', 'Moffat',
'Ripley', 'Winkler', 'San Saba', 'Ballard', 'Beadle', 'Dougherty', 'Hoonah-
Angoon', 'Ohio', 'Huron', 'St. Lawrence', 'Fauquier', 'Lyon', 'Mendocino',
'Dickenson', 'Brunswick', 'Ritchie', 'Dickey', 'Moultrie', 'Tuscarawas', 'Rock
Island', 'Kanabec', 'Fond du Lac', 'Haywood', 'Calumet', 'La Plata', 'James
City', 'Gem', 'St. Helena', 'Navarro', 'Yabucoa', 'Wakulla', 'Crowley',
'Vanderburgh', 'Humphreys', 'Hudspeth', 'Bee', 'Pope', 'Howell', 'Crow Wing',
'Ada', 'Cortland', 'Laurel', 'Price', 'Delta', 'Autauga', 'Coamo', 'Clatsop',
'McLennan', 'Davison', 'Burleson', 'Flathead', 'Gaston', 'Vance', 'Wasatch',
'Van Buren', 'Milam', 'Cheyenne', 'Isabella', 'Osceola', 'Greene', 'Oceana',
'Barceloneta', 'Banner', 'Haralson', 'Caledonia', 'Iosco', 'Bayamón', 'Platte',
'Cedar', 'Kalamazoo', 'Webb', 'Roscommon', 'Camp', 'Troup', 'Whiteside',
'Buncombe', 'Chelan', 'Cape Girardeau', 'St. Francois', 'Watonwan', 'Pitt',
'Wirt', 'Cocke', 'Hanover', 'Hawaii', 'Broadwater', 'Wilson', 'Siskiyou',
'Reno', 'Juab', 'McCook', 'Dickson', 'Winn', 'Cochran', 'Desha', 'Kittitas',
'Allegan', 'Añasco', 'Bath', 'Harney', 'Missaukee', 'Yancey', 'Long',
'Reynolds', 'Kleberg', 'Mountrail', 'Casey', 'Erath', 'Poinsett', 'Tattnall',
'Isabela', 'Baxter', 'San Germán', 'Franklin', 'Juana Díaz', 'Floyd', 'Boyd',
'Tippah', 'Powder River', 'Foard', 'Woodruff', 'Berks', 'Lynchburg', 'Chaffee',
'Montrose', 'Yates', 'Bosque', 'Lares', 'Winchester', 'Isanti', 'Citrus',
'Independence', 'Sherman', 'Petersburg', 'Ferry', 'Furnas', 'Montcalm',
'Arlington', 'Stevens', 'Clarendon', 'Bartholomew', 'Dickens', 'Torrance',
'Nye', 'Harnett', 'Saguache', 'San Bernardino', 'Early', 'Grand Traverse',
'Tillamook', 'Wexford', 'Carver', 'McKenzie', 'Le Flore', 'Dillingham',
'Bienville', 'Carroll', 'Butts', 'Dixon', 'LaPorte', 'Cattaraugus', 'Los
Alamos', 'Woods', 'Hamlin', 'Martinsville', 'Radford', 'Gila', 'Arroyo', 'Nez
Perce', 'Cobb', 'Muhlenberg', 'Wythe', 'Tom Green', 'Oneida', 'Tioga', 'Quay',
'Becker', 'Aleutians West', 'Suwannee', 'Payne', 'Richardson', 'Loving', 'Coke',
'Coshocton', 'Loup', 'Hendricks', 'Presque Isle', 'New Hanover', 'Gwinnett',
'Calvert', 'Danville', 'Jim Hogg', 'Uintah', 'Keith', 'Galveston', 'Yakutat',
'Belmont', 'Walthall', 'San Francisco', 'Greenbrier', 'Labette', 'Dawes',
'Manassas', 'San Joaquin', 'Whitman', 'Kershaw', 'Mitchell', 'Person', 'Dent',
'Hardin', 'Monmouth', 'Menard', 'Gloucester', 'Orange', 'Garfield', 'Hooker',
'Kaufman', 'Red Lake', 'St. John the Baptist', 'Appling', 'Scioto', 'Aitkin',
'Appomattox', 'Chisago', 'Dickinson', 'Butte', 'Holmes', 'Yankton', 'Gallia',
'Hansford', 'Boyle', 'Chatham', 'Boundary', 'Motley', 'Gunnison', 'Otero', 'Río
Grande', 'Pickaway', 'Rensselaer', 'Mayes', 'Hart', 'Montague', 'Georgetown',

```
'Audubon', 'Preston', 'Mathews', 'Kauai', 'Keokuk', 'Smith', 'Shoshone',
'Gaines', 'Craighead', 'Oakland', 'Ketchikan Gateway', 'Hemphill', 'New London',
'Lonoke', 'Cowley', 'Cabell']
```

1.0.3 Task 2

Derive the numbers of counties that use these three names, respectively. For each of them, list their county name and state code.

Hint: Think about what data structure would be best for this task. Once again, do not use special functions or packages for this task. Just use basic Python data structures and loops.

[ ]:
```python
def get_county_states(features):
  """Format the counties to easily list their states

  Arguments:
  features list[dict] -- county data

  Returns:
  dict[str, list[str]] -- County name as the key and the list of state codes as␣
 ↪the value
  """
  county_states = dict()
  for feature in features:
    properties = feature['properties']
    county_name = properties['NAME']
    state_code = properties['STATE']
    county_states.setdefault(county_name, []).append(state_code)

  return county_states


def test_get_county_states():
  mock_features = [{
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US01087',
      'STATE': '01',
      'COUNTY': '087',
      'NAME': 'Macon',
      'LSAD': 'County',
      'CENSUSAREA': 608.885
    },
    "geometry": None,
  }, {
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US02275',
```

```
      'STATE': '02',
      'COUNTY': '275',
      'NAME': 'Wrangell',
      'LSAD': 'Cty&Bor',
      'CENSUSAREA': 2541.483
    },
    "geometry": None,
  }, {
    'type': 'Feature',
      'properties': {
      'GEO_ID': '0500000US02270',
      'STATE': '02',
      'COUNTY': '270',
      'NAME': 'Wade Hampton',
      'LSAD': 'CA',
      'CENSUSAREA': 17081.433
    },
    "geometry": None,
  }, {
    'type': 'Feature',
      'properties': {
      'GEO_ID': '',
      'STATE': '03',
      'COUNTY': '',
      'NAME': 'Wade Hampton',
      'LSAD': '',
      'CENSUSAREA': 0
    },
    "geometry": None,
  }]

  expected = {
    "Macon": ["01"],
    "Wrangell": ["02"],
    "Wade Hampton": ["02", "03"]
  }

  obs = get_county_states(mock_features)
  assert(obs == expected)

test_get_county_states()

county_states = get_county_states(features)
def get_county_totals(county_states):
  """Reformat the counties and state list objects into a tuple of counties and␣
 ↪their totals
```

```
  Arguments:
  county_states dict[str, list[str]] -- County name as the key and the list of␣
 ↪state codes as the value

  Returns:
  tuple[str, int] -- The counties and the total number of states that use them.
  """
  county_totals = [(None, None)] * len(county_states)
  i = 0
  for county, states in county_states.items():
    county_total = (county, len(states))
    county_totals[i] = county_total
    i+=1

  return county_totals


def test_get_county_totals():
  mock_county_states = {
    "A": ['0'],
    "B": ['0', '1'],
    "C": ['0', '1', '2']
  }
  expected_count_totals = [('A', 1), ('B', 2), ('C', 3)]
  assert(get_county_totals(mock_county_states) == expected_count_totals)

test_get_county_totals()

def top_k_sort_k(totals, k):
  """Find the top k values in a tuple of objects and their counts.

  Arguments:
  totals tuple[str, int]-- Item Name and total
  k int -- the number of items to rank

  Returns:
  tuple[str, int] -- Top k items and their counts

  Note:
  This function iterates through the list of items, only sorting the list of␣
 ↪rankings
  """
  top_k = [(None, 0)]*k
  for total in totals:
    total_val = total[1]
    bottom_k_val = top_k[0][1]
    if total_val > bottom_k_val:
```

```
      top_k[0] = total
      top_k.sort(key=lambda a: a[1])

  return top_k

def test_top_k_sort_k():
  mock_county_totals = [("A", 0), ("B", 1), ("C", 2), ("D", 3), ("E", 4), ("F",␣
 ↪4)]
  expected_top_k = [("D", 3), ("F", 4), ("E", 4)]

  assert(top_k_sort_k(mock_county_totals, 3) == expected_top_k)

test_top_k_sort_k()

def top_k_sort_all(totals, k):
  """Find the top k values in a tuple of objects and their counts.

    Arguments:
    totals tuple[str, int]-- Item Name and total
    k int -- the number of items to rank

    Returns:
    tuple[str, int] -- Top k items and their counts

    Note:
    This function sorts the whole list and then takes the top k results
  """
  totals.sort(key = lambda a: a[1])
  return totals[-k:]

def test_top_k_sort_all():
  mock_county_totals = [("A", 0), ("B", 1), ("C", 2), ("D", 3), ("E", 4), ("F",␣
 ↪4)]
  expected_top_k = [("D", 3), ("E", 4), ("F", 4)]
  assert(top_k_sort_all(mock_county_totals, 3) == expected_top_k)

test_top_k_sort_all()

county_totals = get_county_totals(county_states)
"""
Compare the two algorithims to determine which has better performance.

Findings: sorting the rankings list is generally faster when k is less than 25.
Sorting the whole list of counties and taking the top is generally faster when␣
 ↪k is more than 25.
"""
k = 3
```

```
start_time = time.perf_counter()
top_counties = top_k_sort_k(county_totals, k)
stop_time = time.perf_counter()
print(f"top counties: {top_counties}")
print(f"top counter time: {stop_time - start_time}")

start_time = time.perf_counter()
top_counties_alt = top_k_sort_all(county_totals, k)
stop_time = time.perf_counter()
print(f"top counties alt: {top_counties_alt}")
print(f"top counter alt time: {stop_time - start_time}")


"""
Display the top counties and their states
"""
for top_county in top_counties:
  name = top_county[0]
  print(f"{name} county appears in state codes: {county_states[name]}")
```

```
top counties: [('Jefferson', 26), ('Franklin', 26), ('Washington', 31)]
top counter time: 0.0001097899985325057
top counties alt: [('Franklin', 26), ('Jefferson', 26), ('Washington', 31)]
top counter alt time: 0.00012915000115754083
Jefferson county appears in state codes: ['17', '13', '19', '41', '08', '05',
'12', '20', '21', '01', '22', '28', '16', '18', '29', '47', '48', '36', '40',
'31', '30', '42', '39', '54', '55', '53']
Franklin county appears in state codes: ['13', '01', '18', '16', '21', '19',
'28', '36', '48', '05', '12', '22', '23', '25', '17', '20', '29', '47', '31',
'37', '42', '39', '51', '51', '53', '50']
Washington county appears in state codes: ['08', '12', '05', '17', '19', '23',
'24', '29', '47', '40', '41', '55', '49', '50', '20', '18', '21', '27', '13',
'01', '22', '16', '36', '37', '31', '28', '42', '39', '44', '51', '48']
```

1.0.4 Task 3

Basic statistics by state

1.0.5 Task 3, part one

The number of counties in each state

[ ]:
```
def get_state_counties_total(features):
  """Find the total number of counties in each state

  Arguments:
  features list[dict] -- each county has a set of properties, state code is␣
 ↪most relevant
```

```
  Returns:
  dict -- key is state code and value is the total number of counties
  """
  totals = {}
  for feature in features:
    properties = feature['properties']
    state_code = properties['STATE']
    totals[state_code] = totals.get(state_code, 0) + 1

  return totals


def test_get_state_counties_total():
  mock_features = [{
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US01087',
      'STATE': '01',
      'COUNTY': '087',
      'NAME': 'Macon',
      'LSAD': 'County',
      'CENSUSAREA': 608.885
    },
    "geometry": None,
  }, {
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US02275',
      'STATE': '02',
      'COUNTY': '275',
      'NAME': 'Wrangell',
      'LSAD': 'Cty&Bor',
      'CENSUSAREA': 2541.483
    },
    "geometry": None,
  }, {
    'type': 'Feature',
      'properties': {
      'GEO_ID': '0500000US02270',
      'STATE': '02',
      'COUNTY': '270',
      'NAME': 'Wade Hampton',
      'LSAD': 'CA',
      'CENSUSAREA': 17081.433
    },
    "geometry": None,
```

```
  }]

  expected_totals = {
    '01': 1,
    '02': 2
  }

  assert(get_state_counties_total(mock_features) == expected_totals)

test_get_state_counties_total()
state_counties_totals = get_state_counties_total(features)
print(f"List of {len(state_counties_totals)} states' county totals:␣
 ↪{state_counties_totals}")
```

```
List of 52 states' county totals: {'02': 29, '04': 15, '05': 75, '08': 64, '09':
8, '12': 67, '13': 159, '06': 58, '01': 67, '17': 102, '18': 92, '19': 99, '15':
5, '16': 44, '20': 105, '21': 120, '22': 64, '26': 83, '27': 87, '28': 82, '23':
16, '24': 24, '25': 14, '30': 56, '31': 93, '32': 17, '33': 10, '34': 21, '35':
33, '29': 115, '37': 100, '38': 53, '39': 88, '40': 77, '36': 62, '45': 46,
'46': 66, '47': 95, '41': 36, '42': 67, '48': 254, '49': 29, '53': 39, '54': 55,
'55': 72, '56': 23, '72': 78, '50': 14, '51': 134, '10': 3, '11': 1, '44': 5}
```

1.0.6 Task 3, part two

Name and size of the biggest and smallest county in each state, by area

[ ]:
```
def get_state_county_min_max_area(features):
  """Find the name and size of the biggest and smallest county in each state

  Arguments:
  features list[dict] -- county object with type, properties, and geometry

  Returns:
  dict[str, dict] -- county data for largest and smallest counties
  """
  state_county_min_max_area = {}

  for feature in features:
    properties = feature['properties']
    state_code = properties['STATE']
    county_name = properties['NAME']
    county_area = properties['CENSUSAREA']

    county = {
      "name": county_name,
      "area": county_area
    }
```

```
    if state_code in state_county_min_max_area:
      state = state_county_min_max_area[state_code]

      largest_county_area = state['largest_county']['area']
      if county_area > largest_county_area:
        state['largest_county'] = county

      smallest_county_area = state['smallest_county']['area']
      if county_area < smallest_county_area:
        state['smallest_county'] = county
    else:
      state = {
        "largest_county": county,
        "smallest_county": county
      }

      state_county_min_max_area[state_code] = state

  return state_county_min_max_area

def test_get_state_county_min_max_area():
  mock_features = [{
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US01087',
      'STATE': '01',
      'COUNTY': '087',
      'NAME': 'Macon',
      'LSAD': 'County',
      'CENSUSAREA': 608.885
    },
    "geometry": None,
  }, {
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US02275',
      'STATE': '02',
      'COUNTY': '275',
      'NAME': 'Wrangell',
      'LSAD': 'Cty&Bor',
      'CENSUSAREA': 2541.483
    },
    "geometry": None,
  }, {
    'type': 'Feature',
      'properties': {
      'GEO_ID': '0500000US02270',
```

```
      'STATE': '02',
      'COUNTY': '270',
      'NAME': 'Wade Hampton',
      'LSAD': 'CA',
      'CENSUSAREA': 17081.433
    },
    "geometry": None,
  }]

  expected = {
    '01': {
      'largest_county': {
        'name': "Macon",
        "area": 608.885
      },
      "smallest_county": {
        'name': "Macon",
        "area": 608.885
      }
    },
    '02': {
      "largest_county": {
        "name": "Wade Hampton",
        "area": 17081.433
      },
      "smallest_county": {
        "name": "Wrangell",
        "area": 2541.483
      }
    }
  }

  assert(get_state_county_min_max_area(mock_features) == expected)

test_get_state_county_min_max_area()
state_county_min_max_area = get_state_county_min_max_area(features)
print(f"min and max counties by area in each state:␣
  ↪{state_county_min_max_area}")
```

```
min and max counties by area in each state: {'02': {'largest_county': {'name':
'Yukon-Koyukuk', 'area': 145504.789}, 'smallest_county': {'name': 'Skagway',
'area': 452.325}}, '04': {'largest_county': {'name': 'Coconino', 'area':
18618.885}, 'smallest_county': {'name': 'Santa Cruz', 'area': 1236.916}}, '05':
{'largest_county': {'name': 'Union', 'area': 1039.214}, 'smallest_county':
{'name': 'Lafayette', 'area': 528.268}}, '08': {'largest_county': {'name': 'Las
Animas', 'area': 4772.672}, 'smallest_county': {'name': 'Broomfield', 'area':
33.034}}, '09': {'largest_county': {'name': 'Litchfield', 'area': 920.56},
'smallest_county': {'name': 'Middlesex', 'area': 369.301}}, '12':
```

{'largest_county': {'name': 'Collier', 'area': 1998.324}, 'smallest_county':
{'name': 'Union', 'area': 243.556}}, '13': {'largest_county': {'name': 'Ware',
'area': 892.461}, 'smallest_county': {'name': 'Clarke', 'area': 119.2}}, '06':
{'largest_county': {'name': 'San Bernardino', 'area': 20056.938},
'smallest_county': {'name': 'San Francisco', 'area': 46.873}}, '01':
{'largest_county': {'name': 'Baldwin', 'area': 1589.784}, 'smallest_county':
{'name': 'Etowah', 'area': 534.991}}, '17': {'largest_county': {'name':
'McLean', 'area': 1183.378}, 'smallest_county': {'name': 'Putnam', 'area':
160.161}}, '18': {'largest_county': {'name': 'Allen', 'area': 657.308},
'smallest_county': {'name': 'Ohio', 'area': 86.14}}, '19': {'largest_county':
{'name': 'Kossuth', 'area': 972.72}, 'smallest_county': {'name': 'Dickinson',
'area': 380.606}}, '15': {'largest_county': {'name': 'Hawaii', 'area':
4028.417}, 'smallest_county': {'name': 'Kalawao', 'area': 11.991}}, '16':
{'largest_county': {'name': 'Idaho', 'area': 8477.352}, 'smallest_county':
{'name': 'Payette', 'area': 406.867}}, '20': {'largest_county': {'name':
'Butler', 'area': 1429.863}, 'smallest_county': {'name': 'Wyandotte', 'area':
151.6}}, '21': {'largest_county': {'name': 'Pike', 'area': 786.833},
'smallest_county': {'name': 'Robertson', 'area': 99.911}}, '22':
{'largest_county': {'name': 'Vernon', 'area': 1327.91}, 'smallest_county':
{'name': 'Orleans', 'area': 169.423}}, '26': {'largest_county': {'name':
'Marquette', 'area': 1808.401}, 'smallest_county': {'name': 'Benzie', 'area':
319.704}}, '27': {'largest_county': {'name': 'St. Louis', 'area': 6247.401},
'smallest_county': {'name': 'Ramsey', 'area': 152.212}}, '28':
{'largest_county': {'name': 'Yazoo', 'area': 922.947}, 'smallest_county':
{'name': 'Alcorn', 'area': 400.038}}, '23': {'largest_county': {'name':
'Aroostook', 'area': 6671.33}, 'smallest_county': {'name': 'Sagadahoc', 'area':
253.695}}, '24': {'largest_county': {'name': 'Frederick', 'area': 660.221},
'smallest_county': {'name': 'Baltimore', 'area': 80.944}}, '25':
{'largest_county': {'name': 'Worcester', 'area': 1510.77}, 'smallest_county':
{'name': 'Nantucket', 'area': 44.97}}, '30': {'largest_county': {'name':
'Beaverhead', 'area': 5541.624}, 'smallest_county': {'name': 'Silver Bow',
'area': 718.477}}, '31': {'largest_county': {'name': 'Cherry', 'area':
5960.422}, 'smallest_county': {'name': 'Sarpy', 'area': 238.99}}, '32':
{'largest_county': {'name': 'Nye', 'area': 18181.924}, 'smallest_county':
{'name': 'Carson City', 'area': 144.662}}, '33': {'largest_county': {'name':
'Coos', 'area': 1794.685}, 'smallest_county': {'name': 'Strafford', 'area':
368.975}}, '34': {'largest_county': {'name': 'Burlington', 'area': 798.576},
'smallest_county': {'name': 'Hudson', 'area': 46.191}}, '35': {'largest_county':
{'name': 'Catron', 'area': 6923.69}, 'smallest_county': {'name': 'Los Alamos',
'area': 109.167}}, '29': {'largest_county': {'name': 'Texas', 'area': 1177.266},
'smallest_county': {'name': 'St. Louis', 'area': 61.909}}, '37':
{'largest_county': {'name': 'Robeson', 'area': 949.221}, 'smallest_county':
{'name': 'Chowan', 'area': 172.473}}, '38': {'largest_county': {'name':
'McKenzie', 'area': 2760.323}, 'smallest_county': {'name': 'Eddy', 'area':
630.171}}, '39': {'largest_county': {'name': 'Ashtabula', 'area': 701.931},
'smallest_county': {'name': 'Lake', 'area': 227.493}}, '40': {'largest_county':
{'name': 'Osage', 'area': 2246.358}, 'smallest_county': {'name': 'Marshall',
'area': 371.08}}, '36': {'largest_county': {'name': 'St. Lawrence', 'area':

```
2680.377}, 'smallest_county': {'name': 'New York', 'area': 22.829}}, '45':
{'largest_county': {'name': 'Horry', 'area': 1133.896}, 'smallest_county':
{'name': 'McCormick', 'area': 359.13}}, '46': {'largest_county': {'name':
'Meade', 'area': 3470.984}, 'smallest_county': {'name': 'Clay', 'area':
412.185}}, '47': {'largest_county': {'name': 'Shelby', 'area': 763.174},
'smallest_county': {'name': 'Trousdale', 'area': 114.193}}, '41':
{'largest_county': {'name': 'Harney', 'area': 10133.17}, 'smallest_county':
{'name': 'Multnomah', 'area': 431.297}}, '42': {'largest_county': {'name':
'Lycoming', 'area': 1228.594}, 'smallest_county': {'name': 'Montour', 'area':
130.242}}, '48': {'largest_county': {'name': 'Brewster', 'area': 6183.733},
'smallest_county': {'name': 'Rockwall', 'area': 127.036}}, '49':
{'largest_county': {'name': 'San Juan', 'area': 7819.988}, 'smallest_county':
{'name': 'Davis', 'area': 298.778}}, '53': {'largest_county': {'name':
'Okanogan', 'area': 5267.978}, 'smallest_county': {'name': 'San Juan', 'area':
173.915}}, '54': {'largest_county': {'name': 'Randolph', 'area': 1039.681},
'smallest_county': {'name': 'Hancock', 'area': 82.609}}, '55':
{'largest_county': {'name': 'Marathon', 'area': 1544.983}, 'smallest_county':
{'name': 'Pepin', 'area': 231.983}}, '56': {'largest_county': {'name':
'Sweetwater', 'area': 10426.649}, 'smallest_county': {'name': 'Hot Springs',
'area': 2004.092}}, '72': {'largest_county': {'name': 'Arecibo', 'area':
125.947}, 'smallest_county': {'name': 'Cataño', 'area': 4.845}}, '50':
{'largest_county': {'name': 'Windsor', 'area': 969.337}, 'smallest_county':
{'name': 'Grand Isle', 'area': 81.811}}, '51': {'largest_county': {'name':
'Pittsylvania', 'area': 968.941}, 'smallest_county': {'name': 'Falls Church',
'area': 1.999}}, '10': {'largest_county': {'name': 'Sussex', 'area': 936.079},
'smallest_county': {'name': 'New Castle', 'area': 426.286}}, '11':
{'largest_county': {'name': 'District of Columbia', 'area': 61.048},
'smallest_county': {'name': 'District of Columbia', 'area': 61.048}}, '44':
{'largest_county': {'name': 'Providence', 'area': 409.502}, 'smallest_county':
{'name': 'Bristol', 'area': 24.164}}}
```

1.0.7 Task 3, part three

The total and average area of counties in each state

[ ]:
```python
def get_state_total_avg_area_county(features):
  """ The total and average area of counties in each state

  Arguments:
  features list[dict] -- county data

  Returns:
  dict[str, dict] -- total and avg area for counties in the state
  """
  state_total_avg_area_county = {}
  for feature in features:
    properties = feature["properties"]
    state_code = properties['STATE']
```

```
    county_area = properties['CENSUSAREA']

    if state_code in state_total_avg_area_county:
      state = state_total_avg_area_county[state_code]
      state['county_total_area'] = state['county_total_area'] + county_area
      state['county_count'] = state['county_count'] + 1
      state['county_avg_area'] = state['county_total_area'] /␣
  ↪state['county_count']
    else:
      state = {
        "county_total_area": county_area,
        "county_avg_area": county_area,
        "county_count": 1
      }
      state_total_avg_area_county[state_code] = state

  return state_total_avg_area_county

def test_get_state_total_avg_area_county():
  mock_features = [{
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US01087',
      'STATE': '01',
      'COUNTY': '087',
      'NAME': 'Macon',
      'LSAD': 'County',
      'CENSUSAREA': 608.885
    },
    "geometry": None,
  }, {
    'type': 'Feature',
    "properties": {
      'GEO_ID': '0500000US02275',
      'STATE': '02',
      'COUNTY': '275',
      'NAME': 'Wrangell',
      'LSAD': 'Cty&Bor',
      'CENSUSAREA': 2541.483
    },
    "geometry": None,
  }, {
    'type': 'Feature',
      'properties': {
      'GEO_ID': '0500000US02270',
      'STATE': '02',
      'COUNTY': '270',
```

```
      'NAME': 'Wade Hampton',
      'LSAD': 'CA',
      'CENSUSAREA': 17081.433
    },
    "geometry": None,
  }]

  expected = {
    '01': {
      "county_total_area": 608.885,
      "county_avg_area": 608.885,
      "county_count": 1
    },
    '02': {
      "county_total_area": 19622.916,
      "county_avg_area": 9811.458,
      "county_count": 2
    }
  }

  assert(get_state_total_avg_area_county(mock_features) == expected)

test_get_state_total_avg_area_county()
state_total_avg_area_county = get_state_total_avg_area_county(features)
print(f"County areas for states: {state_total_avg_area_county}")
```

```
County areas for states: {'02': {'county_total_area': 570640.9510000001,
'county_avg_area': 19677.274172413796, 'county_count': 29}, '04':
{'county_total_area': 113594.085, 'county_avg_area': 7572.939, 'county_count':
15}, '05': {'county_total_area': 52035.477999999974, 'county_avg_area':
693.806373333333, 'county_count': 75}, '08': {'county_total_area':
103641.88800000002, 'county_avg_area': 1619.4045000000003, 'county_count': 64},
'09': {'county_total_area': 4842.356, 'county_avg_area': 605.2945,
'county_count': 8}, '12': {'county_total_area': 53624.759, 'county_avg_area':
800.3695373134328, 'county_count': 67}, '13': {'county_total_area':
57513.48900000002, 'county_avg_area': 361.72005660377374, 'county_count': 159},
'06': {'county_total_area': 155779.21800000002, 'county_avg_area':
2685.848586206897, 'county_count': 58}, '01': {'county_total_area':
50645.325000000004, 'county_avg_area': 755.9003731343284, 'county_count': 67},
'17': {'county_total_area': 55518.92600000002, 'county_avg_area':
544.3031960784316, 'county_count': 102}, '18': {'county_total_area':
35826.10599999999, 'county_avg_area': 389.4141956521738, 'county_count': 92},
'19': {'county_total_area': 55857.12999999999, 'county_avg_area':
564.2134343434342, 'county_count': 99}, '15': {'county_total_area':
6422.628000000001, 'county_avg_area': 1284.5256000000002, 'county_count': 5},
'16': {'county_total_area': 82643.11899999999, 'county_avg_area':
1878.2527045454544, 'county_count': 44}, '20': {'county_total_area':
81758.71599999997, 'county_avg_area': 778.6544380952378, 'county_count': 105},
```

'21': {'county_total_area': 39486.335, 'county_avg_area': 329.0527916666667,
'county_count': 120}, '22': {'county_total_area': 43203.90500000001,
'county_avg_area': 675.0610156250002, 'county_count': 64}, '26':
{'county_total_area': 56538.899000000005, 'county_avg_area': 681.1915542168675,
'county_count': 83}, '27': {'county_total_area': 79626.74499999998,
'county_avg_area': 915.2499425287355, 'county_count': 87}, '28':
{'county_total_area': 46923.27099999999, 'county_avg_area': 572.2350121951218,
'county_count': 82}, '23': {'county_total_area': 30842.923, 'county_avg_area':
1927.6826875, 'county_count': 16}, '24': {'county_total_area': 9707.242,
'county_avg_area': 404.46841666666666, 'county_count': 24}, '25':
{'county_total_area': 7800.056999999999, 'county_avg_area': 557.1469285714285,
'county_count': 14}, '30': {'county_total_area': 145545.79900000003,
'county_avg_area': 2599.0321250000006, 'county_count': 56}, '31':
{'county_total_area': 76824.17799999999, 'county_avg_area': 826.0664301075267,
'county_count': 93}, '32': {'county_total_area': 109781.18, 'county_avg_area':
6457.716470588235, 'county_count': 17}, '33': {'county_total_area': 8952.65,
'county_avg_area': 895.265, 'county_count': 10}, '34': {'county_total_area':
7354.220999999999, 'county_avg_area': 350.2009999999999, 'county_count': 21},
'35': {'county_total_area': 121298.14800000002, 'county_avg_area':
3675.701454545455, 'county_count': 33}, '29': {'county_total_area':
68741.52599999998, 'county_avg_area': 597.7523999999999, 'county_count': 115},
'37': {'county_total_area': 48617.90499999998, 'county_avg_area':
486.1790499999998, 'county_count': 100}, '38': {'county_total_area':
69000.79599999997, 'county_avg_area': 1301.9018113207542, 'county_count': 53},
'39': {'county_total_area': 40860.695000000014, 'county_avg_area':
464.3260795454547, 'county_count': 88}, '40': {'county_total_area':
68594.91999999998, 'county_avg_area': 890.8431168831166, 'county_count': 77},
'36': {'county_total_area': 47126.39399999999, 'county_avg_area':
760.1031290322579, 'county_count': 62}, '45': {'county_total_area':
30060.695999999996, 'county_avg_area': 653.4933913043477, 'county_count': 46},
'46': {'county_total_area': 75811.0, 'county_avg_area': 1148.6515151515152,
'county_count': 66}, '47': {'county_total_area': 41234.894, 'county_avg_area':
434.0515157894737, 'county_count': 95}, '41': {'county_total_area': 95988.012,
'county_avg_area': 2666.333666666667, 'county_count': 36}, '42':
{'county_total_area': 44742.70200000001, 'county_avg_area': 667.8015223880599,
'county_count': 67}, '48': {'county_total_area': 261231.709, 'county_avg_area':
1028.4712952755906, 'county_count': 254}, '49': {'county_total_area': 82169.621,
'county_avg_area': 2833.435206896552, 'county_count': 29}, '53':
{'county_total_area': 66455.52000000002, 'county_avg_area': 1703.987692307693,
'county_count': 39}, '54': {'county_total_area': 24038.212, 'county_avg_area':
437.0584, 'county_count': 55}, '55': {'county_total_area': 54157.80399999998,
'county_avg_area': 752.191722222222, 'county_count': 72}, '56':
{'county_total_area': 97093.141, 'county_avg_area': 4221.440913043479,
'county_count': 23}, '72': {'county_total_area': 3423.779000000001,
'county_avg_area': 43.89460256410258, 'county_count': 78}, '50':
{'county_total_area': 9216.655999999999, 'county_avg_area': 658.3325714285713,
'county_count': 14}, '51': {'county_total_area': 39490.086, 'county_avg_area':
294.7021343283582, 'county_count': 134}, '10': {'county_total_area':

1948.5439999999999, 'county_avg_area': 649.5146666666666, 'county_count': 3},
'11': {'county_total_area': 61.048, 'county_avg_area': 61.048, 'county_count':
1}, '44': {'county_total_area': 1033.815, 'county_avg_area': 206.763,
'county_count': 5}}