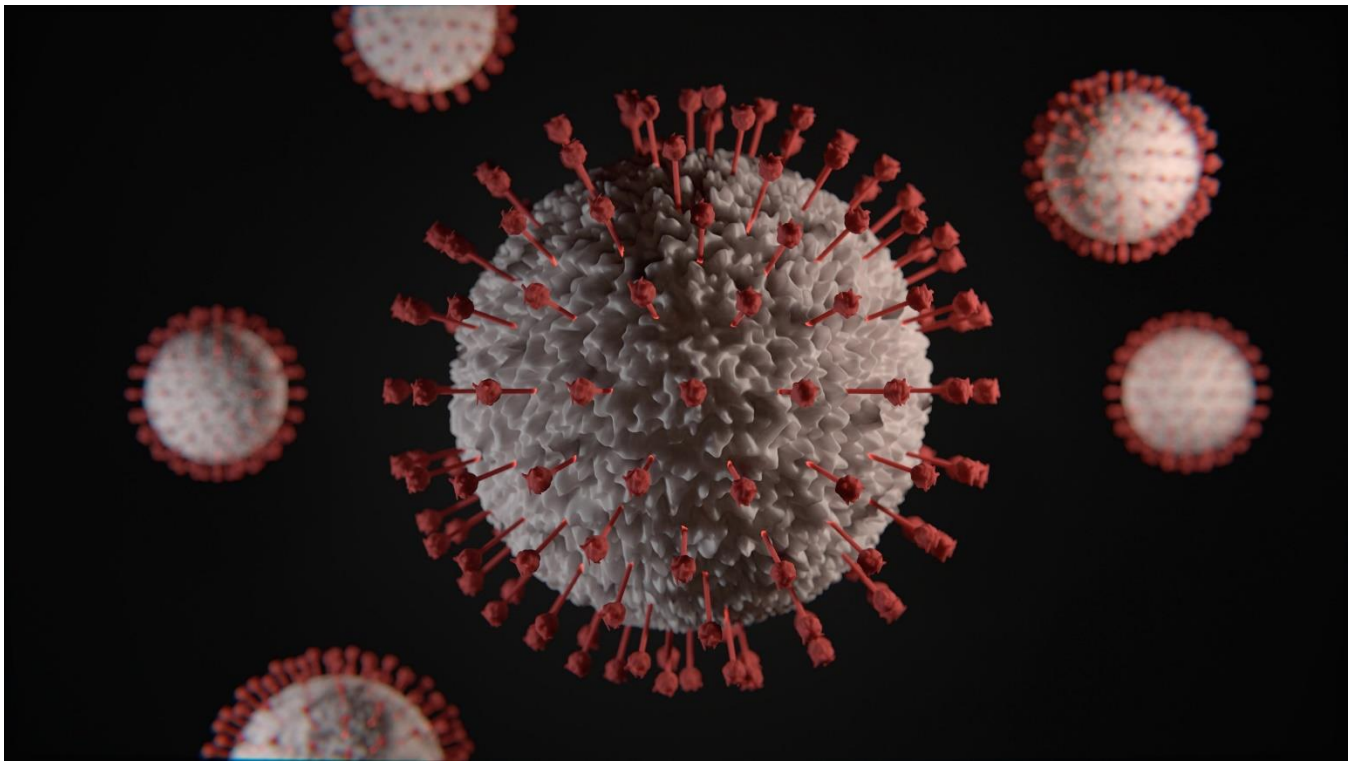
Benjamin Tang
tang.benjamin@gmail



COVID-19 Case Counts in Select Western States:

A Python Script to Export a Map Series

Project Summary

A Python script is used to export a series of maps illustrating COVID-19 coronavirus cases in five U.S. states. The script uses two nested for-loops, one iterates over all states in the statelist and another over the feature layers. A SearchCursor is used to dynamically retrieve the unique state names for the statelist from the ArcGIS Pro project file. SelectLayerByAttribute command selects the appropriate county features for each state to present and zoom in mapframe, and ultimately to export to PDF. Two different map layouts, portrait and landscape, are used for different state shapes.

The ArcGIS Pro project file displays coronavirus data (cases by county) downloaded from ArcGIS Online (numerical data provided by Johns Hopkins University). County spatial data, comprised of the case data joined to U.S. Census population estimates, form two feature layers: the first is the choropleth of coronavirus cases normalized to county population, and the second is dot density (with limiting ancillary attributes) for a more realistically looking showing case data.

COVID-19 data sources:

- https://coronavirus-resources.esri.com/pages/resources
- https://covidtracking.com/data

Introduction and Purpose

Coronavirus disease 2019 (COVID-19) is a new infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), related to the SARS-CoV virus that caused an outbreak of severe acute respiratory syndrome (SARS) that killed over 900 people in Asia in 2002-2004 and the MERS-CoV virus that caused an outbreak of Middle East respiratory syndrome, or camel flu, and caused 400 deaths in the Middle East in 2012-2015.

The majority of COVID-19 cases develop mild flu-like symptoms of fever, cough, chills, muscle pain, sore throat and fatigue, but some cases develop acute respiratory distress syndrome (ARDS), multi-organ failure and septic shock.[1] Unlike SARS and MERS, COVID-19 is actually a less deadly disease with a lower case fatality rate – possibly 1.3% compared to 10% for SARS and 35% for MERS.[2] However, the lower mortality rate means infected persons are able to transmit the disease and infect more people: as of May 19, 2020, the ongoing global COVID-19 pandemic has 4,896,312 confirmed cases and 323,221 deaths.[3] In the United States, COVID-19 has infected 1.5 million and killed almost 92,000 people.

There is currently no vaccine for COVID-19. On March 19, 2020, California became the first state in the nation to institute a shelter-in-place lockdown of residents to slow the spread of the disease.[4] After a month of lockdown, the consequent economic pressures compelled many state governors to consider formalizing conditions for revoking their shelter-in-place orders. On April 27, a working group of governors from California, Oregon, Washington, Colorado and Nevada formed a Western States Pact, in recognition of their social and economic ties, to coordinate on reopening their economies.[5] One consideration was to allow multiple phases of reopening to allow certain jurisdictions, notably rural counties less affected by COVID-19, to reopen earlier without overly stringent public health requirements.

This project uses a Python script to export – from an existing ArcGIS Pro project file – maps illustrating COVID-19 cases in each of the five Western States Pact states. The ArcGIS Pro project file displays coronavirus cases by county, with data provided by Johns Hopkins University and made available for download on ArcGIS Online. There are two main feature layers, both displaying county spatial data. The first background layer is a choropleth classified by natural breaks and showing number of cases normalized to county population. The second layer shows the confirmed cases again, but as a dot density map for more realism. Limiting ancillary attributes of water and parkland cover restrict dot placement to residential areas of the map.

Geoprocessing Tasks in Script

The Python script uses two nested for-loops, SearchCursor, SelectLayerByAttribute and two separate map layouts – portrait and landscape – for different state shapes. The two for-loops iterates the main script over all desired states and feature layers. The first for-loop uses the list variable `statelist` to loop over the five Western states. List elements are populated from the `counties_cases_West` feature layer (the choropleth map, although in this case either layer would do):

```
fc = maplist.listLayers("counties_cases_West")[0]
states = [row[0] for row in arcpy.da.SearchCursor(fc,'STATE_NAME')]
statelist = set(states) # remove duplicates, as multiple counties have same STATE_NAME
print ('\nStates to be mapped:')
print (statelist)
```

[1] Centers for Disease Control and Prevention (2020). *Coronavirus Disease 2019 (COVID-19): Symptoms*. Murthy, et al (2020). "Care for Critically Ill Patients With COVID-19." *JAMA*.

[2] Basu, A. (2020). *Estimating The Infection Fatality Rate Among Symptomatic COVID-19 Cases In The United States*. Health Affairs. World Health Organization (2003). *Summary of probable SARS cases with onset of illness from 1 November 2002 to 31 July 2003*. World Health Organization (2019). *Middle East respiratory syndrome coronavirus (MERS-CoV)*.

[3] Johns Hopkins University (2020). *COVID-19 Dashboard*. Coronavirus Resource Center.

[4] White (2020). "Newsom orders all 40M Californians to stay home in nation's strictest state lockdown." *Politico*.

[5] Office of Governor Gavin Newsom (2020). "Colorado & Nevada Join California, Oregon & Washington in Western States Pact."

Initially, I had trouble getting the `SearchCursor` to recognize the feature layer as an input parameter, so I created a preset `statelist` just in case.

```
    # CHEAT: preset statelist instead of reading from 'counties_cases_West' layer
    # statelist = ["California", "Nevada", "Oregon", "Washington", "Colorado"]

    # Making map for each state:
    for state in statelist:
```

The beginning of the `statelist` for-loop also defines the `STATE_NAME` query to specify which state should features (counties) be selected in each `TOCLayer` loop:

```
        print ('\nCreating map of ' + state) # check correct state
        query = """"STATE_NAME" = '""" + state + """'"""
        # print (query) # check correct query

        for TOCLayer in maplist.listLayers():
        # print ("Layer Name: " + str(TOCLayer.longName)) # check layers are correct
```

Since the exported maps need to zoom on the selected features to frame the state properly, I only needed to loop over the choropleth layer `counties_cases_West`. The choropleth has a polygon geometry – its features are county shapes. The dot density naturally has point geometry and symbolizes COVID-19 cases. The `SelectLayerByAttribute_management` command selects the county shapes, which taken together is the shape of the desired state. As a check, the number of selected features (i.e., the counties) is retrieved with `GetCount_management` and printed to screen:

```
            if TOCLayer.name == "counties_cases_West":

                # Select county features for queried state
                arcpy.SelectLayerByAttribute_management(TOCLayer, "NEW_SELECTION", query)
                result = arcpy.GetCount_management(TOCLayer)
                print ("Selected features: " + str(result) + " counties in " + state)
                # Correct number of counties: CA 58, NV 17, OR 36, WA 39, CO 64
```

Before zooming in on the selected features to frame the map, the proper layout orientation must be selected. In the variable definitions at the beginning of the script (not shown), I had designated separate layout (`layout_p`, `layout_l`), mapframe (`mapframe_p`, `mapframe_l`), and text element variables (`tElements_p`, `tElements_l`) for both portrait and landscape. The ArcGIS Pro project had two separate layouts already set and named Layout_Portrait and Layout_Landscape (see Figure 1), so the layouts were defined by calling them by name: for example, the landscape `layout_l` was defined by `aprx.listLayouts("Layout_Landscape")[0]`. Unfortunately, I did not know how to dynamically set the optimal orientation for each state map, so I preset the orientations by state name (California and Nevada are “tall” states that stretch north-south and fit the portrait layout, while Oregon, Washington and Colorado are “wide” east-west states that better fit the landscape layout):

```
                # Set the proper layout orientation
                if state in ("California", "Nevada"): # tall states use portrait layout
                    layout = layout_p
                    mapframe = mapframe_p
                    tElements = tElements_p
                else: # wide states use landscape layout
                    layout = layout_l
                    mapframe = mapframe_l
                    tElements = tElements_l
```

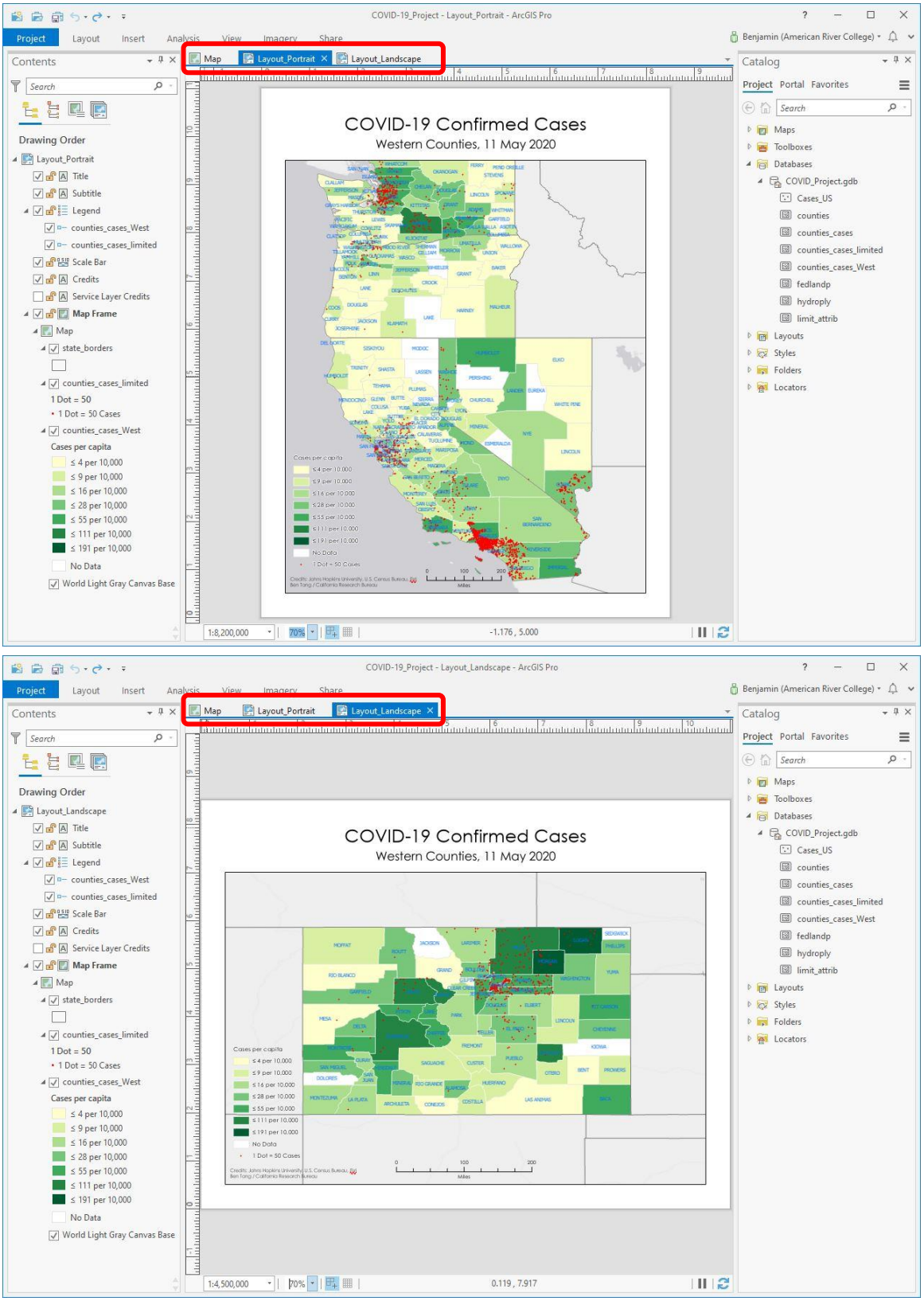


Figure 1: Preset portrait and landscape orientation layouts in the ArcGIS Pro project

With the orientation set, the mapframes can then zoom on the selection. Making `setExtent` equal to `layer_extent` maximizes the entire mapframe space (the state features "touch" the mapframe borders), so a small buffer is added by changing the map scale. Once the extent of the mapframe is set, the selections can be cleared (so the selection highlights do not get exported, but also to clear the selection before the next loop/state):

```
            # Zoom to selected features then clear selection
            layer_extent = mapframe.getLayerExtent(TOCLayer)
            mapframe.camera.setExtent(layer_extent)
            mapframe.camera.scale = mapframe.camera.scale * 1.1
            arcpy.SelectLayerByAttribute_management(TOCLayer, "CLEAR_SELECTION")
```

Most of the map elements are the same for all state maps, so only the subtitle specifying the state name needs to be changed. Finally, before exporting the map, any pre-existing copies of the map is deleted first:

```
            # Retitle map for each state
            for tElement in tElements:
                if tElement.name == 'Subtitle':
                tElement.text = state + ', ' + str(todays_date)

            print ('Exporting map of ' + state + '...')

            # Delete any existing copies of map, then export
            if arcpy.Exists(exportpath + 'COVID-19_' + state + '.pdf'):
                arcpy.Delete_management(exportpath + 'COVID-19_' + state + '.pdf')
            layout.exportToPDF(exportpath + "COVID-19_" + state + ".pdf")

print ('\nCompleted Map Exports')
```

Difficulties

The main complication I had with the script was getting a specific map layer/feature class recognized in Python for the SearchCursor. I wanted to dynamically set the number of maps exported by retrieving the number of unique state values from the `counties_cases_West` layer data. To save a map layer to a variable, I finally used:

```
fc = maplist.listLayers("counties_cases_West")[0]
```

I had tried other methods first. I did not find the online Esri help reference describing Python for ArcGIS Pro helpful: the sample codes had the variable set to a path.[6] For example, to get the "roads" layer, use:

```
fc = "c:/data/base.gdb/roads"
```

This method did not seem to work for me, even though I think I properly set the arcpy workspace to the COVID-19_Project geodatabase. The method recounted in class – directly defining the `fc` variable as the layer name, in quotes – did not seem to work either. I had hoped to add more functionality to the script by calling select feature layers, perhaps adding or removing the dot density layer as desired, as well as adding calculations with the table data, perhaps a field calculation with the county population, or any other census demographics data such as race (African-Americans seem particularly vulnerable to the disease).[7]

A second complication I had was with the map symbology. I do not believe there are arcpy functions to change ArcGIS Pro symbology, but I would not have mastered them in time anyways. The map legend explaining the choropleth classes sits in the bottom-left corner of the mapframe; in the landscape orientation, the legend overprints the map features. A possible

[6] Esri. "ArcGIS Pro Python reference."
[7] Erdman (2020). "Black communities account for disproportionate number of Covid-19 deaths in the US, study finds." *CNN*.

fix is to remake the legend from a vertical list to a horizontal one, though it would likely still overprint the map. The surest solution is to place the legend outside the mapframe in both portrait and landscape layouts, but then the actual mapped area could be quite small.

Johns Hopkins University has COVID-19 data for every U.S. state: if I pursued this project further, I would like to expand the map series to be either a time series, showing changes in case data over regular periods, or to include mortality or testing data (instead of just infection cases). Perhaps the script could be expanded to download COVID-19 data from ArcGIS Online at regular intervals (every day?) to update the numerical case data. However, with the number of cases increasing to an unknown maximum within any enumeration unit, the changes to the choropleth classification ranges are unknown. This is a stylistic problem because I rewrote the symbol descriptions to be more legible (for example, I rewrote "≤ 0.0191" to "≤ 191 per 10,000"). One possible fix is to switch from natural breaks to a set classification, perhaps equal intervals.

With regards to mapping coronavirus deaths or testing numbers, it is probably better to create a separate map project that symbolizes only fatalities, rather than combine both cases and deaths on the same map. In any case, the utility of mass producing any number of PDF maps for each jurisdiction (state or county) and the ease of switching between deaths, tests or case datasets makes Python scripting for ArcGIS extremely useful.

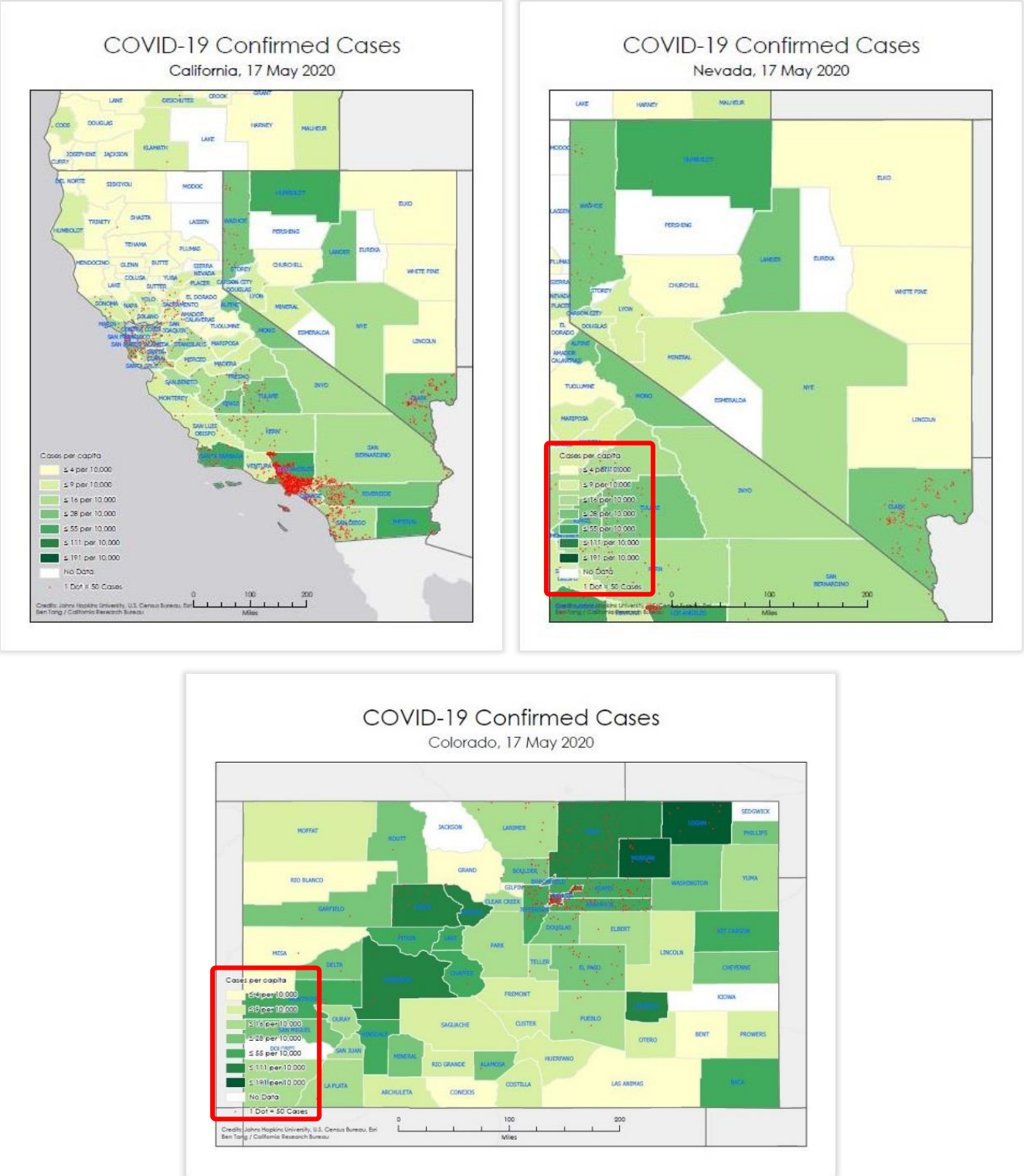


Figure 1: Exported maps of California (top left) and Nevada (top right) in portrait orientation and Colorado (bottom) in landscape orientation. Placement of legend in lower left corner overwrites maps in landscape layout.

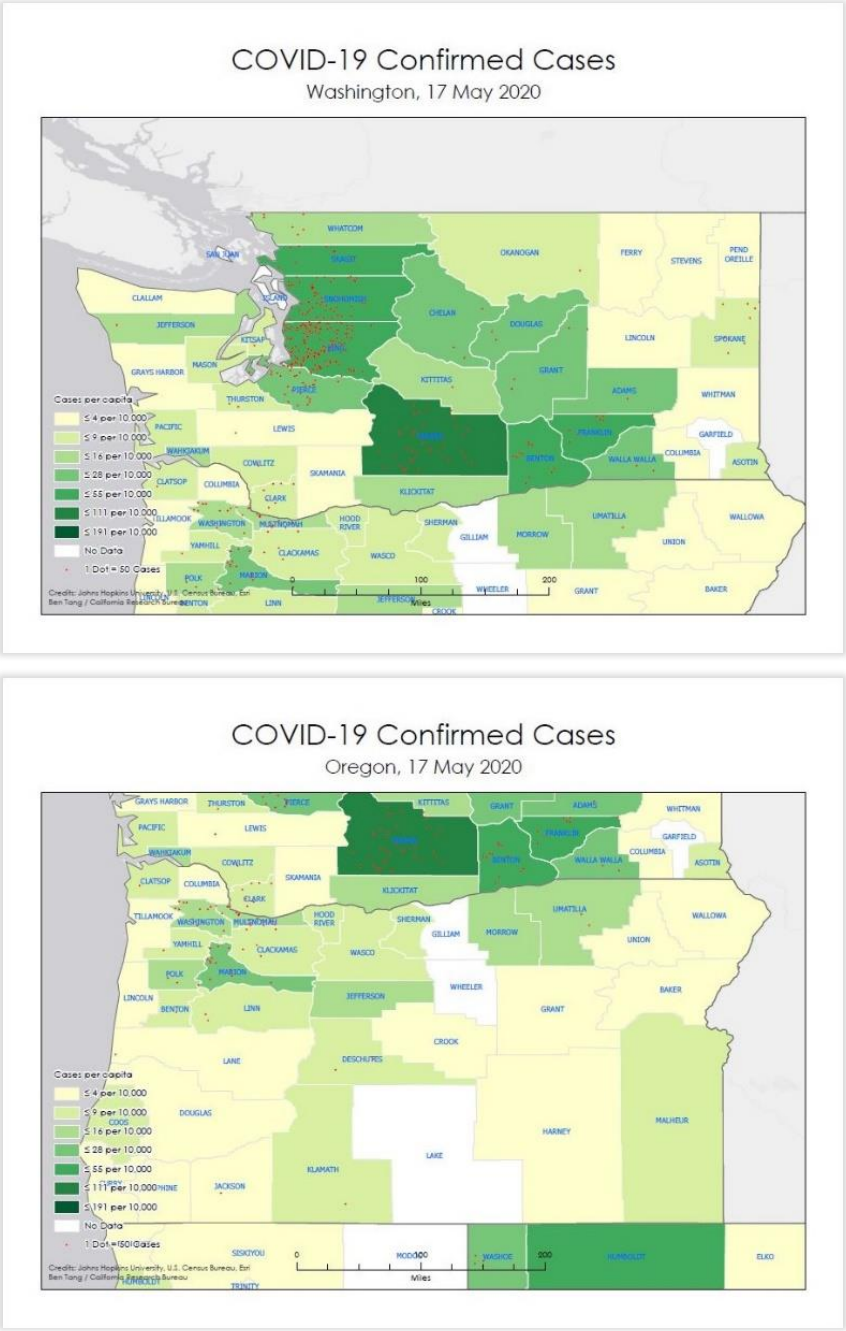


Figure 2: Exported maps of Washington (top) and Oregon (bottom) in landscape orientation layout