USA County Choropleth Maps in Python

How to create colormaped representations of USA counties by FIPS values in Python.

Deprecation warning

This page describes a legacy "figure factory" method (/python/figure-factories/) for creating map-like figures using self-filled scatter traces (/python/shapes). **This is no longer the recommended way to make county-level choropleth maps**, instead we recommend using a GeoJSON-based approach to making outline choropleth maps (/python/choropleth-maps/) or the alternative tile-based choropleth maps (/python/tile-county-choropleth).

Required Packages

`plotly_geo`, `geopandas`, `pyshp` and `shapely` must be installed for this figure factory to run.

Run the following commands to install the correct versions of the following modules:

```
!pip install plotly-geo==1.0.0
!pip install geopandas==0.8.1
!pip install pyshp==2.1.2
!pip install shapely==1.7.1
```

```
Collecting plotly-geo==1.0.0
  Downloading plotly_geo-1.0.0-py3-none-any.whl.metadata (834 bytes)
Downloading plotly_geo-1.0.0-py3-none-any.whl (23.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.7/23.7 MB 156.9 MB/s eta 0:00:00
Installing collected packages: plotly-geo
Successfully installed plotly-geo-1.0.0
Collecting geopandas==0.8.1
  Downloading geopandas-0.8.1-py2.py3-none-any.whl.metadata (892 bytes)
Collecting pandas>=0.23.0 (from geopandas==0.8.1)
  Downloading pandas-2.3.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (91 kB)
Collecting shapely (from geopandas==0.8.1)
  Downloading shapely-2.0.7-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.8 kB)
Collecting fiona (from geopandas==0.8.1)
```

If you are using Windows, follow this post to properly install geopandas and dependencies: http://geoffboeing.com/2014/09/using-geopandas-windows/ (http://geoffboeing.com/2014/09/using-geopandas-windows/). If you are using Anaconda, do not use PIP to install the packages above. Instead use conda to install them:

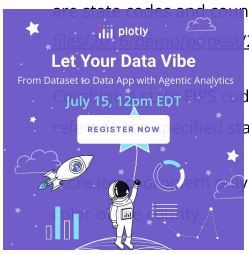
conda install plotly conda install geopandas

FIPS and Values

Every US state and county has an assigned ID regulated by the US Federal Government under the term FIPS (Federal Information Processing Standards) codes. There are state codes and county codes: the 2016 state and county FIPS codes can be found at the US Census Website (https://www.census.gov/geographies/reference-files/time/demo/popest/2016-fips.html).

Counties have FIPS codes (eg. 06 for California) with a county FIPS code of the state (eg. 059 for Orange county) and this new state-county FIPS code (06059) uniquely refers to a specified state and county.

`create_choropleth` needs a list of FIPS codes and a list of values. Each FIPS code points to one county and each corresponding value in `values` determines the

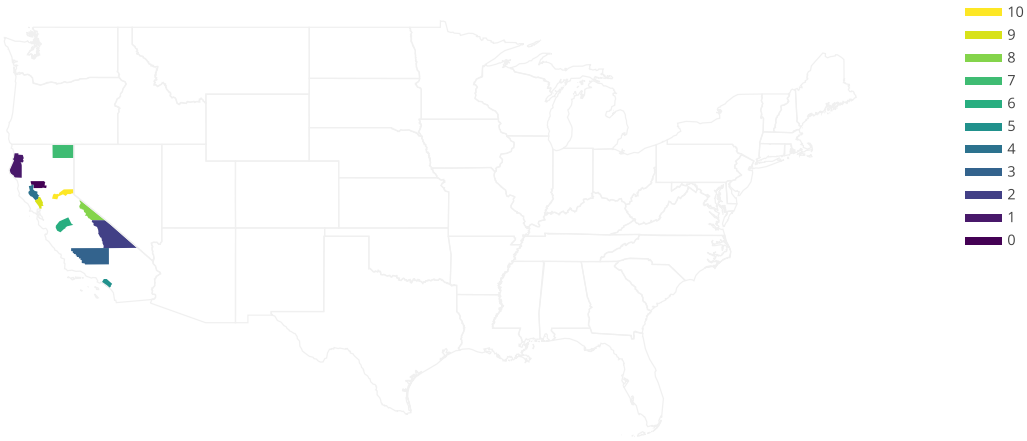


Simple Example

A simple example of this is a choropleth a few counties in California:

```
import plotly.figure_factory as ff

fips = ['06021', '06023', '06027',
        '06029', '06033', '06059',
        '06047', '06049', '06051',
        '06055', '06061']
values = range(len(fips))

fig = ff.create_choropleth(fips=fips, values=values)
fig.layout.template = None
fig.show()
```

```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:774: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



Change the Scope

Even if your FIPS values belong to a single state, the scope defaults to the entire United States as displayed in the example above. Changing the scope of the choropleth shifts the zoom and position of the USA map. You can define the scope with a list of state names and the zoom will automatically adjust to include the state outlines of the selected states.

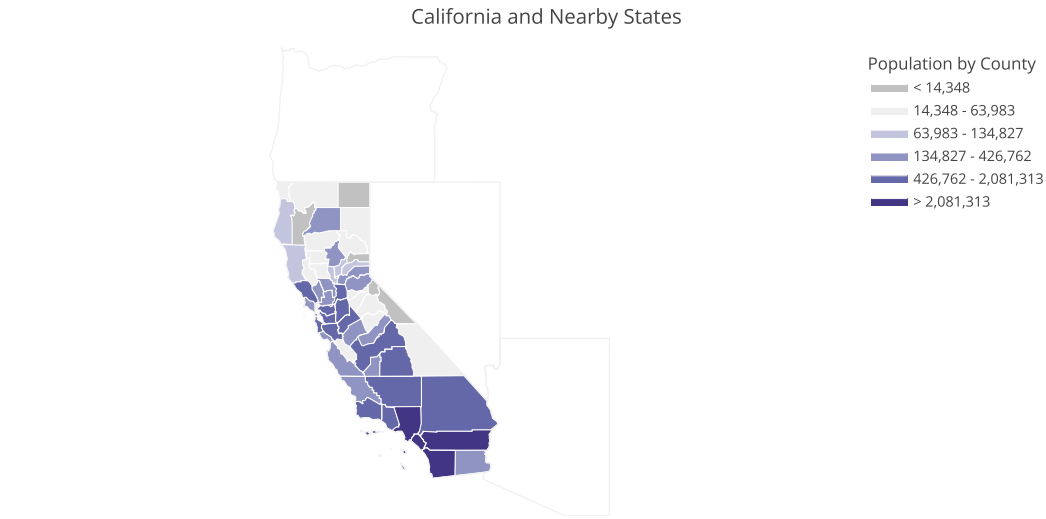
By default scope is set to ['USA'] which the API treats as identical to passing a list of all 50 state names:

CA) or the proper names (eg. California) as strings are accepted. If the state name is not recognized, the API will throw a Warning and indicate
gnored.

he example below is binning_endpoints. If your values is a list of numbers, you can bin your values into half-open intervals on the real line.

```
import plotly.figure_factory as ff

import numpy as np
import pandas as pd

df_sample = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/minoritymajority.csv')
df_sample_r = df_sample[df_sample['STNAME'] == 'California']

values = df_sample_r['TOT_POP'].tolist()
fips = df_sample_r['FIPS'].tolist()

colorscale = [
    'rgb(193, 193, 193)',
    'rgb(239,239,239)',
    'rgb(195, 196, 222)',
    'rgb(144,148,194)',
    'rgb(101,104,168)',
    'rgb(65, 53, 132)'
]

fig = ff.create_choropleth(
    fips=fips, values=values, scope=['CA', 'AZ', 'Nevada', 'Oregon', ' Idaho'],
    binning_endpoints=[14348, 63983, 134827, 426762, 2081313], colorscale=colorscale,
    county_outline={'color': 'rgb(255,255,255)', 'width': 0.5}, round_legend_values=True,
    legend=dict(
      title=dict(
        text='Population by County'
      )
    ),
    title=dict(
      text='California and Nearby States'
    )
)
fig.layout.template = None
fig.show()
```

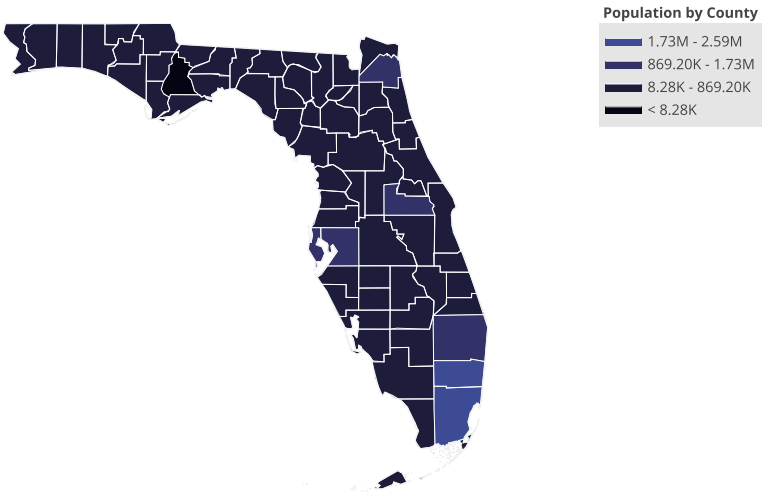
```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:808: ShapelyDeprecationWarning:
The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:
The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:357: ShapelyDeprecationWarning:
The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:
The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:
The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



Single State

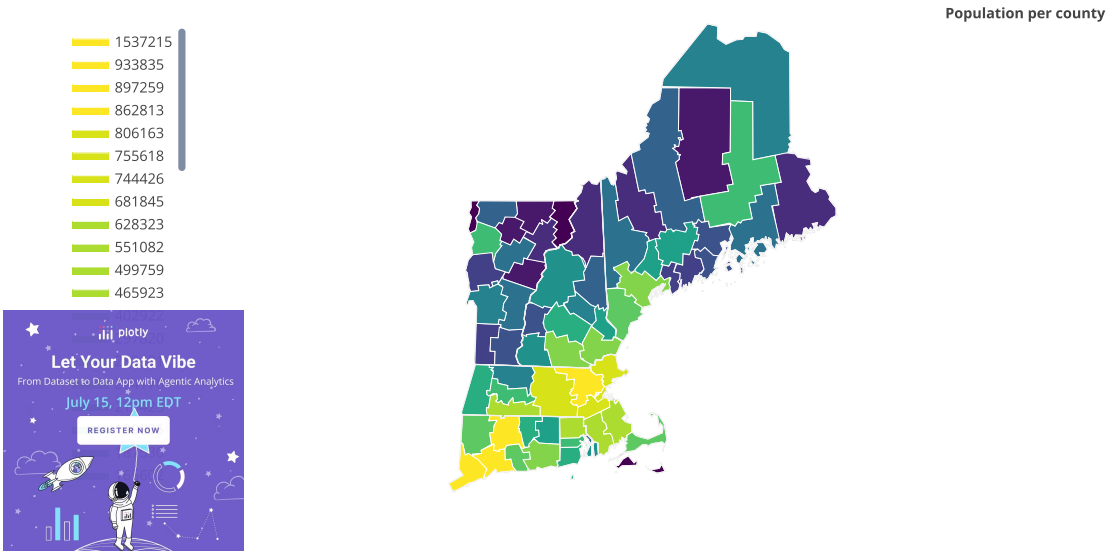
```
import plotly.figure_factory as ff

import numpy as np
import pandas as pd

df_sample = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/minoritymajority.csv')
df_sample_r = df_sample[df_sample['STNAME'] == 'Florida']

values = df_sample_r['TOT_POP'].tolist()
fips = df_sample_r['FIPS'].tolist()

endpts = list(np.mgrid[min(values):max(values):4j])
colorscale = ["#030512","#1d1d3b","#323268","#3d4b94","#3e6ab0",
              "#4989bc","#60a7c7","#85c5d3","#b7e0e4","#eafcfd"]
fig = ff.create_choropleth(
    fips=fips, values=values, scope=['Florida'], show_state_data=True,
    colorscale=colorscale, binning_endpoints=endpts, round_legend_values=True,
    plot_bgcolor='rgb(229,229,229)',
    paper_bgcolor='rgb(229,229,229)',
    legend_title='Population by County',
    county_outline={'color': 'rgb(255,255,255)', 'width': 0.5},
    exponent_format=True,
)
fig.layout.template = None
fig.show()
```

```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:808: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:357: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



```
import plotly.figure_factory as ff

import pandas as pd

NE_states = ['Connecticut', 'Maine', 'Massachusetts', 'New Hampshire', 'Rhode Island', 'Vermont']
df_sample = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/minoritymajority.csv')
df_sample_r = df_sample[df_sample['STNAME'].isin(NE_states)]

values = df_sample_r['TOT_POP'].tolist()
fips = df_sample_r['FIPS'].tolist()

colorscale = [
    'rgb(68.0, 1.0, 84.0)',
    'rgb(66.0, 64.0, 134.0)',
    'rgb(38.0, 130.0, 142.0)',
    'rgb(63.0, 188.0, 115.0)',
    'rgb(216.0, 226.0, 25.0)'
]

fig = ff.create_choropleth(
    fips=fips, values=values,
    scope=NE_states, county_outline={'color': 'rgb(255,255,255)', 'width': 0.5},
    legend_title='Population per county'

)
fig.update_layout(
    legend_x = 0,
    annotations = {'x': -0.12, 'xanchor': 'left'}
)

fig.layout.template = None
fig.show()
```

```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:774: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:357: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



Simplify County, State Lines

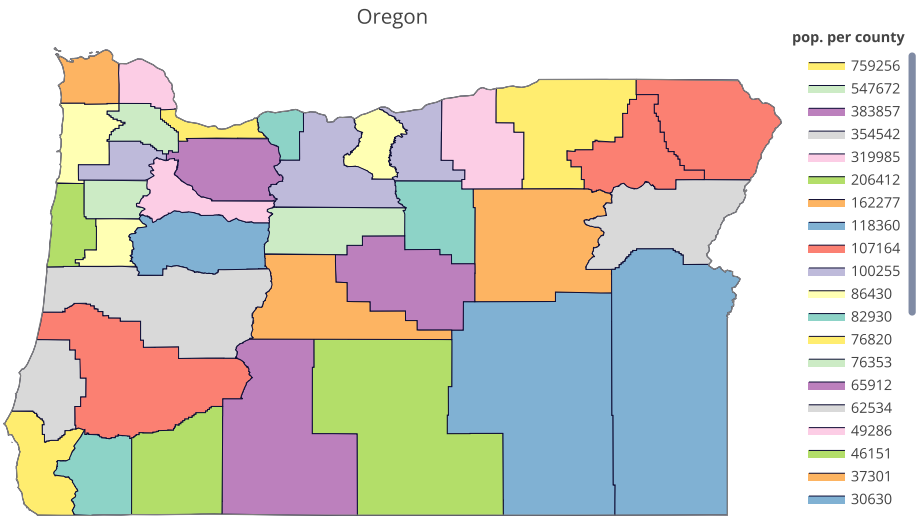
Below is a choropleth that uses several other parameters. For a full list of all available params call help(ff.create_choropleth)

- simplify_county determines the simplification factor for the counties. The larger the number, the fewer vertices and edges each polygon has. See http://toblerity.org/shapely/manual.html#object.simplify (http://toblerity.org/shapely/manual.html#object.simplify) for more information.
- simplify_state simplifies the state outline polygon. See the documentation (http://toblerity.org/shapely/manual.html#object.simplify) for more information. Default for both simplify_county and simplify_state is 0.02

Note: This choropleth uses a divergent categorical colorscale. See http://react-colorscales.getforge.io/ (http://react-colorscales.getforge.io/) for other cool colorscales.

```
import plotly.figure_factory as ff

import pandas as pd

scope = ['Oregon']
df_sample = pd.read_csv(
    'https://raw.githubusercontent.com/plotly/datasets/master/minoritymajority.csv'
)
df_sample_r = df_sample[df_sample['STNAME'].isin(scope)]

values = df_sample_r['TOT_POP'].tolist()
fips = df_sample_r['FIPS'].tolist()

colorscale = ["#8dd3c7", "#ffffb3", "#bebada", "#fb8072",
              "#80b1d3", "#fdb462", "#b3de69", "#fccde5",
              "#d9d9d9", "#bc80bd", "#ccebc5", "#ffed6f",
              "#8dd3c7", "#ffffb3", "#bebada", "#fb8072",
              "#80b1d3", "#fdb462", "#b3de69", "#fccde5",
              "#d9d9d9", "#bc80bd", "#ccebc5", "#ffed6f",
              "#8dd3c7", "#ffffb3", "#bebada", "#fb8072",
              "#80b1d3", "#fdb462", "#b3de69", "#fccde5",
              "#d9d9d9", "#bc80bd", "#ccebc5", "#ffed6f"]

fig = ff.create_choropleth(
    fips=fips, values=values, scope=scope,
    colorscale=colorscale, round_legend_values=True,
    simplify_county=0, simplify_state=0,
    county_outline={'color': 'rgb(15, 15, 55)', 'width': 0.5},
    state_outline={'width': 1},
    legend_title='pop. per county',
    title='Oregon'
)

fig.layout.template = None
fig.show()
```

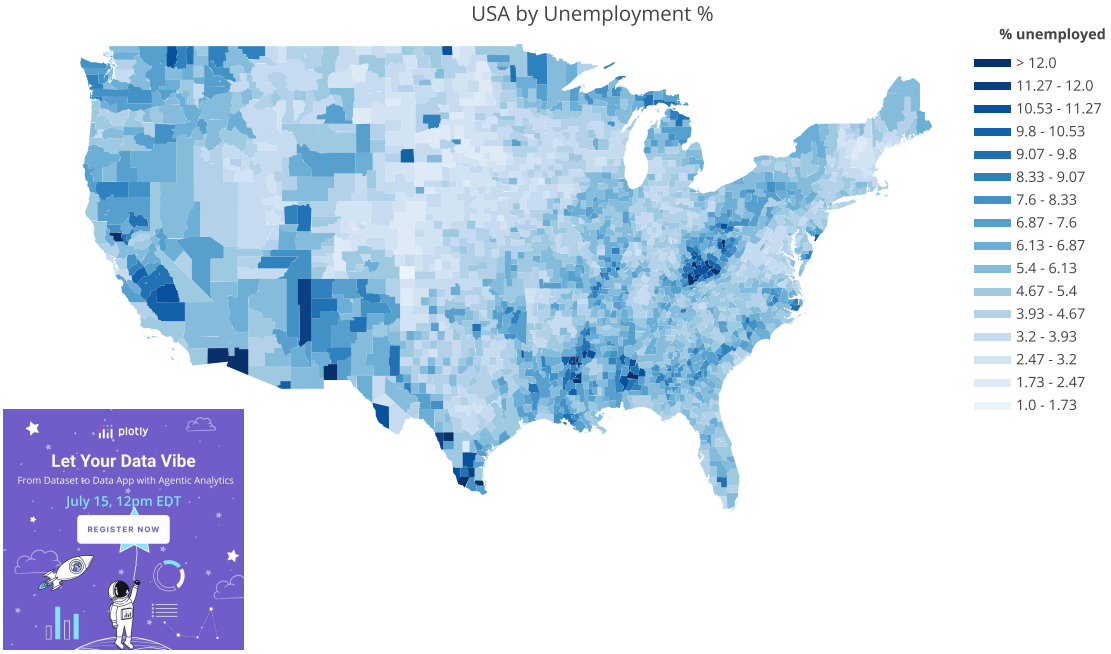
```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:774: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:357: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



The Entire USA

```
import plotly.figure_factory as ff

import numpy as np
import pandas as pd

df_sample = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/laucnty16.csv')
df_sample['State FIPS Code'] = df_sample['State FIPS Code'].apply(lambda x: str(x).zfill(2))
df_sample['County FIPS Code'] = df_sample['County FIPS Code'].apply(lambda x: str(x).zfill(3))
df_sample['FIPS'] = df_sample['State FIPS Code'] + df_sample['County FIPS Code']

colorscale = ["#f7fbff","#ebf3fb","#deebf7","#d2e3f3","#c6dbef","#b3d2e9","#9ecae1",
              "#85bcdb","#6baed6","#57a0ce","#4292c6","#3082be","#2171b5","#1361a9",
              "#08519c","#0b4083","#08306b"]
endpts = list(np.linspace(1, 12, len(colorscale) - 1))
fips = df_sample['FIPS'].tolist()
values = df_sample['Unemployment Rate (%)'].tolist()

fig = ff.create_choropleth(
    fips=fips, values=values,
    binning_endpoints=endpts,
    colorscale=colorscale,
    show_state_data=False,
    show_hover=True, centroid_marker={'opacity': 0},
    asp=2.9, title='USA by Unemployment %',
    legend_title='% unemployed'
)

fig.layout.template = None
fig.show()
```

```
/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:808: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:330: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:357: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:847: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.

/home/circleci/project/doc/.venv/lib/python3.9/site-packages/plotly/figure_factory/_county_choropleth.py:852: ShapelyDeprecationWarning:

The 'type' attribute is deprecated, and will be removed in the future. You can use the 'geom_type' attribute instead.
```



Also see tile county choropleths made in Python: https://plotly.com/python/tile-county-choropleth/ (https://plotly.com/python/tile-county-choropleth/)

Reference

For more info on ff.create_choropleth(), see the full function reference (https://plotly.com/python-api-reference/generated/plotly.figure_factory.create_choropleth.html)

What About Dash?

Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

Everywhere in this page that you see fig.show(), you can display the same figure in a Dash application by passing it to the figure argument of the Graph component (https://dash.plot.ly/dash-core-components/graph) from the built-in dash_core_components package like this:

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )

from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False)  # Turn off reloader if inside Jupyter
```