An Introduction to Geographical Data Visualization

Ryan Gotesman
May 20, 2018 · 5 min read

Choropleth map from Wiki

I have always been very impressed with figures that manage to combine data with geographical information. They are some of the most beautiful examples of data visualization I can think of and the information you can gleam with a single glance is astounding.

These kinds of figures are referred to as choropleth maps and according to Wiki are defined as:

> Thematic map(s) in which areas are shaded or patterned in proportion to the measurement of the statistical variable being displayed on the map, such as population density or per-capita income

Think of them like heat maps that take geographic boundaries into account.

We can make choropleth maps using the plotly library.

We will begin with a simple example. We import as follows:

```
import plotly as py
import plotly.graph_objs as go
```

We want to create a ‘data’ variable which we will pass into the go.Figure() method to generate the choropleth map.

Let’s start with a map of the big wide world!

We define the ‘data’ variable as follows:

```
data = dict (
    type = 'choropleth',
    locations = ['China','Canada','Brazil'],
    locationmode='country names',
    colorscale = ['Viridis'],
    z=[10,20,30])
```

The ‘data’ variable is a dictionary. We define it’s ‘type’ as choropleth. In ‘locations’ we give the names of various countries and in locationmode we tell the method we plan on specifying location by country name. Finally we specify the color scale and the values we want to assign to each country.

You can run plotly either online or offline and to avoid having to make an account I’ll be running it offline.

We generate the image with:

```
map = go.Figure(data=[data])
py.offline.plot(map)
```

and get a map of the world. Note when we hover our mouse over a country we get its name and assigned value.

Rather than assigning arbitrary values to a few countries we can make this map more interesting by plotting the happiness of each nation.

Using data from the 2017 World Happiness Report we can create this figure using the code below

```
import plotly as py
import plotly.graph_objs as go
import pandas as pd

df = pd.read_csv("2017.csv")

data = dict (
    type = 'choropleth',
    locations = df['Country'],
    locationmode='country names',
    colorscale = ['Viridis'],
    z=df['Happiness.Score'])

map = go.Figure(data=[data])
py.offline.plot(map)
```

to generate the following figure. Without any need for statistical analysis we can confidently say some of the happiest countries are found in Europe and North America while some of the least happy countries are concentrated in Africa. That is the power of geographical visualization.

Now let’s do some in-depth visualization of the United States.

We first change the ‘data’ variable so that locationmode specifies American states. Next we add a ‘layout’ variable which allows us to customize certain aspects of the map. In this case we are telling the map to focus in on the United States rather than show the entire world which is the default. If you wanted you could also focus in on Asia, Europe, Africa, North America or South America.

Putting it all together we assign values to Arizona, California and Vermont with the following code:

```
data = dict (
    type = 'choropleth',
    locations = ['AZ','CA','VT'],
    locationmode='USA-states',
    colorscale = ['Viridis'],
    z=[10,20,30])

lyt = dict(geo=dict(scope='usa'))
map = go.Figure(data=[data], layout = lyt)
py.offline.plot(map)
```

to get this map:

We can now start to do some pretty fun things.

It’s well known that the world is experiencing a rapid decline in bee populations. This is awful because it will spell the end to lovely honey nut cheerio breakfasts. We can get our hands on the number of bee colonies per state and use our plotting to skills to see which states have the most bee colonies.

```
df = pd.read_csv("honeyproduction.csv")

data = dict (
    type = 'choropleth',
    locations = df['state'],
    locationmode='USA-states',
    colorscale = ['Viridis'],
    z=df['numcol'])

lyt = dict(geo=dict(scope='usa'))
map = go.Figure(data=[data], layout = lyt)
py.offline.plot(map)
```

It looks like North Dakota has the most colonies following by California with alot of room for improvement in the rest of the country. We are also missing some data for Alaska and a couple states in the top right but that’s a limitation of this dataset.

We’ve touched upon happiness and bee colonies. Now all that’s left is unemployment.

Let’s go deeper and instead of looking at countries or states, and we can graph counties in each states. We will get data of the 2016 percent unemployment for each county from here.

Every single county can be uniquely identified with a 5 digit FIP number.

We begin with our imports and notice this time we import the plotly.figure_factory

```
import plotly as py
import plotly.figure_factory as ff
import numpy as np
import pandas as pd
```

Next we read in the FIP numbers and also percent unemployment

```
df=pd.read_csv("https://raw.githubusercontent.com/p
lotly/datasets/master/laucnty16.csv")
df ['FIPS'] = df['State FIPS Code'].apply(lambda x:
str(x).zfill(2)) + df['County FIPS
Code'].apply(lambda x: str(x).zfill(3))
fips = df['FIPS'].tolist()
values = df['Unemployment Rate (%)'].tolist()
```

Since there are many different values for unemployment and we don't want our color scale to try to represent 1000 different values we restrict it to just 12 colors with the following

```
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))
```

and pass all these arguments into our choropleth method

```
fig = ff.create_choropleth(fips=fips,
values=values, colorscale= colorscale,
binning_endpoints=endpts)
py.offline.plot(fig)
```

to get

Wow what a pretty map!

Using such a graph we can quickly spot areas with elevated rates of unemployment (encircled in red).

For future analysis it would be interesting to figure out why these areas are experiencing more unemployment and what can be done about it.

I hope you've enjoyed this quick introduction to geographical data visualization. Let me know if you have any questions and if you have any examples of cool geographical data visualizations you'd like to share.

Thanks!

Data Visualization   Choropleth Maps   Data Science

Medium

About   Help   Legal