

[Get started](#)[Open in app](#)[Follow](#)

553K Followers



You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

A Beginners Guide to Create a Choropleth Map in Python using GeoPandas and Matplotlib



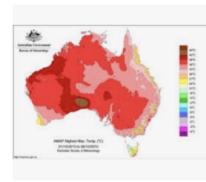
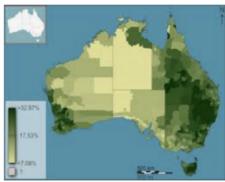
M. Rake Linggar A. · Sep 16, 2019 · 5 min read ★

Data visualisation is an important skill when searching and presenting important insights. There are many visuals that can be used to present your data. One of the most interesting data visual is the choropleth map.

What's a choropleth map?

A **choropleth map** (from Greek χῶρος “area/region” and πλῆθος “multitude”) is a thematic map 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.

Source: https://en.wikipedia.org/wiki/Choropleth_map



[Get started](#)[Open in app](#)

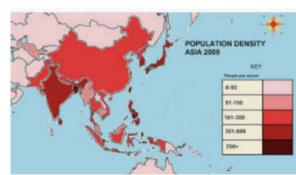
Creating a Basic Choropleth Map in Arc...
michaelminn.net



Visualization: Choropleth Map ...
medium.com



Choropleth Maps and census
illinoisstate.edu



How to Make a Choropleth Map - YouTube
youtube.com



choropleth map - Wiktionary
en.wiktionary.org

Google Image search — choropleth

Why is this such an interesting visual? a) it is pretty, b) it tells us the data that we are interested in exactly the location of where it is associated in, and c) it is pretty!

With the introductions done, let's get down to the code (and the preparations for it).

Step 1: Install required Python libraries

Let's install several packages that we'll need for this exercise. GeoPandas is an amazing package that takes `pandas`'s DataFrame to the next level by allowing it to parse geospatial data. It will use Descartes to generate a Matplotlib plot.

```
pip install descartes
pip install geopandas
pip install matplotlib
pip install numpy
pip install pandas
```

Step 2: Get the data

There are two kinds of data that we will need for this exercise:

1. The data that will be mapped to locations/regions/areas/etc. There are plenty of open data that we can use freely, one if it is from Wikipedia (data accuracy and validity not guaranteed, but for this learning exercise, no problem!). For this

[Get started](#)[Open in app](#)

Github repo as there needs some manual work on it (changing the column name, data format, etc.) which we will not bother to discuss here.

2. The second data is a shapefile of the map that we want to make. It is basically a list of geometric locations (either in points, lines, or polygons). Since we want to map Indonesia's provinces, we will download Indonesia's Administration area [here](#), or again, in my [Github repo](#).

Step 3: Begin to code

Now, let's jump right into the code

1. Load the necessary libraries

```
import pandas as pd
import numpy as np
import geopandas as gpd
import matplotlib.pyplot as plt
```

2. Load and view the shapefile data

```
fp = "IDN_adm/IDN_adm1.shp"
map_df = gpd.read_file(fp)
# check the GeoDataframe
map_df.head()
```

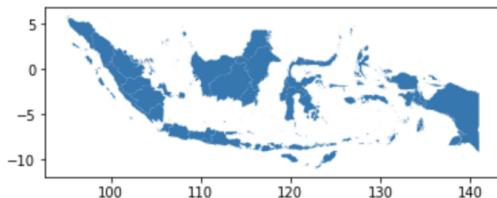
ID_0	ISO	NAME_0	ID_1	NAME_1	TYPE_1	ENGTYPE_1	NL_NAME_1	VARNAME_1	geometry
0	106	IDN	Indonesia	1	Aceh	Propinsi	Autonomous Province	None	Achin Atjeh Nanggroe Aceh Darussalam (POLYGON ((97.38493347167986 1.977126002311707...
1	106	IDN	Indonesia	2	Bali	Propinsi	Province	None	Penida Lembongan Ceningan Menjangan (POLYGON ((115.6298828125 -8.77598762512207, 1...
2	106	IDN	Indonesia	3	Bangka-Belitung	Propinsi	Province	None	Babel Kepulauan Bangka Belitung (POLYGON ((108.0695114135744 -3.8030490875243,...
3	106	IDN	Indonesia	4	Banten	Propinsi	Province	None	(POLYGON ((105.5486068725586 -6.99471807479858...
4	106	IDN	Indonesia	5	Bengkulu	Propinsi	Province	None	Bencoolen Benkoelen Benkulen (POLYGON ((102.3862686157229 -5.47111892700195...

[Get started](#)[Open in app](#)

shape of the province). And as you can see as well, the shapefile stores the location information in the form of polygons. Let's plot it, shall we

```
map_df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x10e45a780>
```



So we have the map of Indonesia, but it looks too small, let's resize it

```
plt.rcParams['figure.figsize'] = [50, 70] #height, width  
map_df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1285a64a8>
```



Much better,

3. Load the province data

```
province = pd.read_csv("data_province.csv",  
sep=";")  
province.head()
```

[Get started](#)[Open in app](#)

	province	population_2015	area_km2	population_density_per_km2	cities_Regencies	cities	regencies
0	Jawa Timur	38828061	48	828	38	9	29
1	Jawa Tengah	33753023	41	894	35	6	29
2	Sumatera Utara	13923262	73	188	33	8	25
3	Papua	3143088	319	8	29	1	28
4	Jawa Barat	46668214	35	1	27	9	18

As you can see, we have the provinces, 2015 population, number of cities, and several other interesting numbers. All we have to do now is to merge the data with the shapefile and we can begin visualizing these numbers

4. Merge and show the map

```
# join the geodataframe with the csv dataframe  
merged = map_df.merge(province, how='left', left_on="NAME_1",  
right_on="province")  
merged = merged[['province', 'geometry', 'population_2015',  
'area_km2', 'population_density_per_km2', '  
'cities_Regencies', 'cities', 'regencies']]  
  
merged.head()
```

	province	geometry	population_2015	area_km2	population_density_per_km2	cities_Regencies	cities	regencies	
0	Aceh	(POLYGON ((97.38493347167986 1.977126002311707...))	4993385.0	58.0		77.0	23.0	5	18.0
1	Bali	(POLYGON ((115.6298828125 -8.77598762512207, 1...))	4148588.0	6.0		621.0	9.0	1	8.0
2	Bangka-Belitung	(POLYGON ((108.0695114135744 -3.8030490875243,...))	1370331.0	16.0		64.0	7.0	1	6.0
3	Banten	(POLYGON ((105.5486068725586 -6.99471807479858...))	11934373.0	10.0		909.0	8.0	4	4.0
4	Bengkulu	(POLYGON ((102.3862686157229 -5.47111892700195...))	1872136.0	20.0		84.0	10.0	1	9.0

Cool, we have the data in the most clean format, let's make the plot

```
# set the value column that will be visualised  
variable = 'cities_Regencies'
```

[Get started](#)[Open in app](#)

```
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 10))

# remove the axis
ax.axis('off')

# add a title and annotation
ax.set_title('# of Cities per each Region', fontdict={'fontsize': '25', 'fontweight' : '3'})
ax.annotate('Source: Wikipedia -\nhttps://en.wikipedia.org/wiki/Provinces\_of\_Indonesia', xy=(0.6, .05), xycoords='figure fraction', fontsize=12, color='#555555')

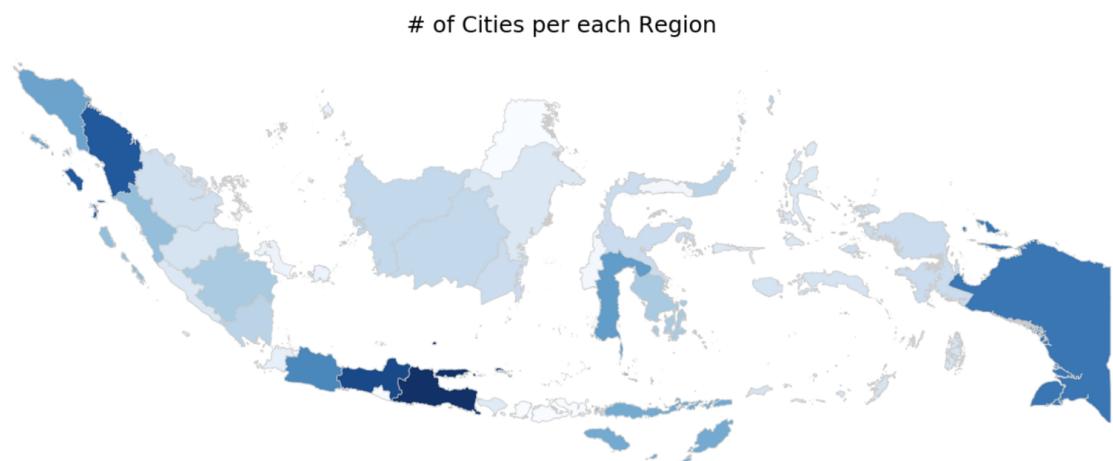
# Create colorbar legend
sm = plt.cm.ScalarMappable(cmap='Blues',
norm=plt.Normalize(vmin=vmin, vmax=vmax))

# empty array for the data range
sm.set_array([]) # or alternatively sm._A = []. Not sure why this step is necessary, but many recommends it

# add the colorbar to the figure
fig.colorbar(sm)

# create map
merged.plot(column=variable, cmap='Blues', linewidth=0.8, ax=ax,
edgecolor='0.8')
```

<matplotlib.axes._subplots.AxesSubplot at 0x1266b54a8>

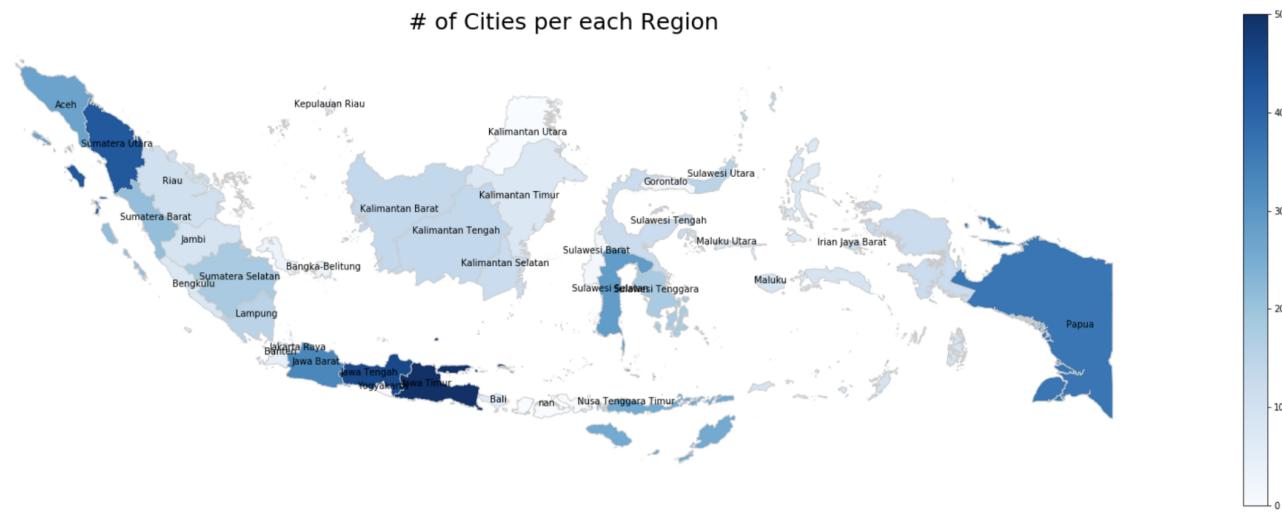


Source: Wikipedia - https://en.wikipedia.org/wiki/Provinces_of_Indonesia

[Get started](#)[Open in app](#)

plot clearer, let's add the province labels to it. Add the following code at the bottom of the code above.

```
# Add Labels  
merged['coords'] = merged['geometry'].apply(lambda x:  
x.representative_point().coords[:])  
merged['coords'] = [coords[0] for coords in merged['coords']]  
  
for idx, row in merged.iterrows():  
    plt.annotate(s=row['province'],  
    xy=row['coords'],horizontalalignment='center')
```



Ok, that is better. If we take a closer look, we can see that, according to Wikipedia, **Jawa Tengah** province has a very high number of cities per regions compared to other provinces.

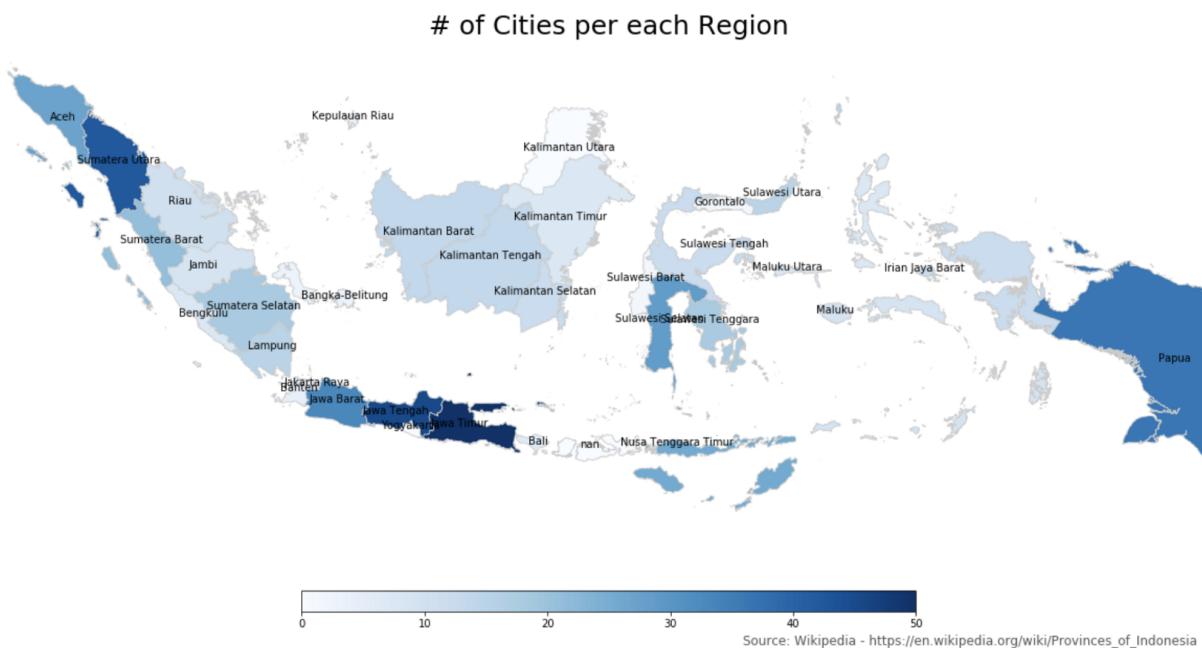
Another tweak we can do is to change the orientation of the color map legend to be horizontal, in case you want the upper space to focus on the map.

Just change this code

```
fig.colorbar(sm)
```

[Get started](#)[Open in app](#)

```
fig.colorbar(sm, orientation="horizontal", fraction=0.036, pad=0.1, aspect = 30)
```



Final tweak

5. Save it!

Now that we have made the choropleth map, the last thing we'll need to do is to save it in a friendly popular format, like .png

```
fig.savefig('map.png', dpi=300)
```

That's all for now, hope this post is useful.

Try it yourself

You can download the Notebook file for this exercise along with the cleaned province data and Indonesian shapefiles in my Github repo [here](#).

Credits

[Get started](#)[Open in app](#)

Let's make a map! Using Geopandas, Pandas and Matplotlib to make a Choropleth map

So you want to make a map using Python. Let's get started!

towardsdatascience.com

I was interested in learning and doing something further and loved the results, hence I am sharing it in my own post here. :)

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

[Get this newsletter](#)

You'll need to sign in or create an account to receive this newsletter.

[Python](#) [Choropleth Map](#) [Data](#) [Data Visualization](#) [Data Analysis](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

