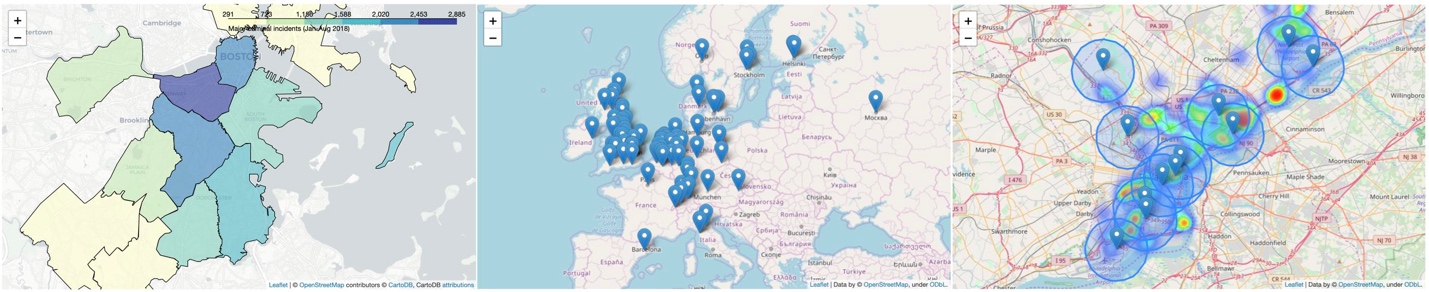
GEOSPATIAL ANALYSIS

Introduction

In this micro-course, you'll learn about different methods to wrangle and visualize **geospatial data**, or data with a geographic location.



Along the way, you'll offer solutions to several real-world problems like:

* Where should a global non-profit expand its reach in remote areas of the Philippines?
* How do purple martins, a threatened bird species, travel between North and South America? Are the birds travelling to conservation areas?
* Which areas of Japan could potentially benefit from extra earthquake reinforcement?
* Which Starbucks stores in California are strong candidates for the next [Starbucks Reserve Roastery](https://www.forbes.com/sites/garystern/2019/01/22/starbucks-reserve-roastery-its-spacious-and-trendy-but-why-is-starbucks-slowing-down-expansion/#6cb80d4a1bc6) location?
* Does New York City have sufficient hospitals to respond to motor vehicle collisions? Which areas of the city have gaps in coverage?

You'll also visualize crime in the city of Boston, examine health facilities in Ghana, explore top universities in Europe, and track releases of toxic chemicals in the United States.

In this first tutorial, we'll quickly cover the pre-requisites that you'll need to complete this micro-course. And, if you'd like to review more deeply, we recommend the [**Pandas micro-course**](https://www.kaggle.com/learn/pandas).

We'll also get started with visualizing our first geospatial dataset!

Reading data

The first step is to read in some geospatial data! To do this, we'll use the [GeoPandas](http://geopandas.org/) library.

In [1]:

import geopandas as gpd

There are many, many different geospatial file formats, such as [shapefile](https://en.wikipedia.org/wiki/Shapefile), [GeoJSON](https://en.wikipedia.org/wiki/GeoJSON), [KML](https://en.wikipedia.org/wiki/Keyhole_Markup_Language), and [GPKG](https://en.wikipedia.org/wiki/GeoPackage). We won't discuss their differences in this micro-course, but it's important to mention that:

* shapefile is the most common file type that you'll encounter, and
* all of these file types can be quickly loaded with the gpd.read\_file() function.

The next code cell loads a shapefile containing information about forests, wilderness areas, and other lands under the care of the [Department of Environmental Conservation](https://www.dec.ny.gov/index.html) in the state of New York.

In [2]:

*# Read in the data*

full\_data = gpd.read\_file("../input/geospatial-learn-course-data/DEC\_lands/DEC\_lands/DEC\_lands.shp")

*# View the first five rows of the data*

full\_data.head()

Out[2]:

As you can see in the "CLASS" column, each of the first five rows corresponds to a different forest.

For the rest of this tutorial, consider a scenario where you'd like to use this data to plan a weekend camping trip. Instead of relying on crowd-sourced reviews online, you decide to create your own map. This way, you can tailor the trip to your specific interests.

Prerequisites

To view the first five rows of the data, we used the head() method. You may recall that this is also what we use to preview a Pandas DataFrame. In fact, every command that you can use with a DataFrame will work with the data!

This is because the data was loaded into a (GeoPandas) **GeoDataFrame** object that has all of the capabilities of a (Pandas) DataFrame.

In [3]:

type(full\_data)

Out[3]:

geopandas.geodataframe.GeoDataFrame

For instance, if we don't plan to use all of the columns, we can select a subset of them. (*To review other methods for selecting data, check out*[*this tutorial*](https://www.kaggle.com/residentmario/indexing-selecting-assigning/)*from the Pandas micro-course*.)

In [4]:

data = full\_data.loc[:, ["CLASS", "COUNTY", "geometry"]].copy()

We use the value\_counts() method to see a list of different land types, along with how many times they appear in the dataset. (*To review this (and related methods), check out*[*this tutorial*](https://www.kaggle.com/residentmario/summary-functions-and-maps)*from the Pandas micro-course.*)

In [5]:

*# How many lands of each type are there?*

data.CLASS.value\_counts()

Out[5]:

WILD FOREST 965

INTENSIVE USE 108

PRIMITIVE 60

WILDERNESS 52

ADMINISTRATIVE 17

UNCLASSIFIED 7

HISTORIC 5

PRIMITIVE BICYCLE CORRIDOR 4

CANOE AREA 1

Name: CLASS, dtype: int64

You can also use loc (and iloc) and isin to select subsets of the data. (*To review this, check out*[*this tutorial*](https://www.kaggle.com/residentmario/indexing-selecting-assigning/)*from the Pandas micro-course.*)

In [6]:

*# Select lands that fall under the "WILD FOREST" or "WILDERNESS" category*

wild\_lands = data.loc[data.CLASS.isin(['WILD FOREST', 'WILDERNESS'])].copy()

wild\_lands.head()

Out[6]:

|  | CLASS | COUNTY | geometry |
| --- | --- | --- | --- |
| 0 | WILD FOREST | DELAWARE | POLYGON ((486093.245 4635308.586, 486787.235 4... |
| 1 | WILD FOREST | DELAWARE | POLYGON ((491931.514 4637416.256, 491305.424 4... |
| 2 | WILD FOREST | DELAWARE | POLYGON ((486000.287 4635834.453, 485007.550 4... |
| 3 | WILD FOREST | GREENE | POLYGON ((541716.775 4675243.268, 541217.579 4... |
| 4 | WILD FOREST | ESSEX | POLYGON ((583896.043 4909643.187, 583891.200 4... |

If you're not familiar with the commands above, you are encouraged to bookmark this page for reference, so you can look up the commands as needed. (*Alternatively, you can take the*[*Pandas micro-course*](https://www.kaggle.com/learn/pandas)*.*) We'll use these commands throughout this micro-course to understand and filter data before creating maps.

Create your first map!

We can quickly visualize the data with the plot() method.

In [7]:

wild\_lands.plot()

Out[7]:

<AxesSubplot:>



Every GeoDataFrame contains a special "geometry" column. It contains all of the geometric objects that are displayed when we call the plot() method.

In [8]:

*# View the first five entries in the "geometry" column*

wild\_lands.geometry.head()

Out[8]:

0 POLYGON ((486093.245 4635308.586, 486787.235 4...

1 POLYGON ((491931.514 4637416.256, 491305.424 4...

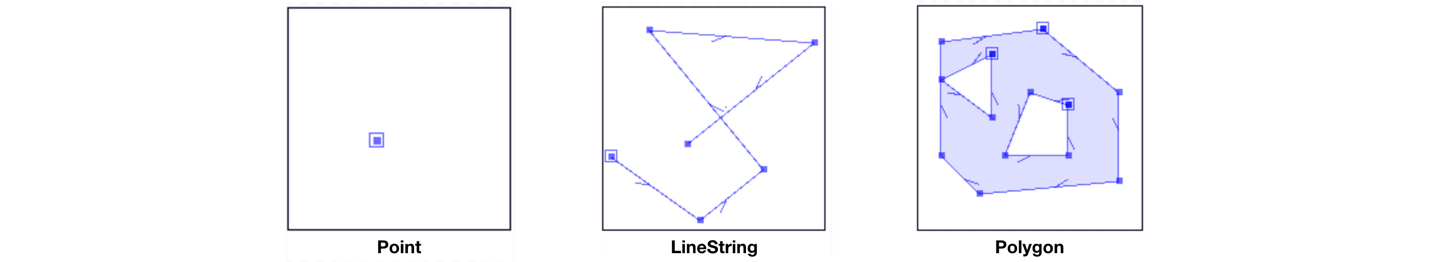
2 POLYGON ((486000.287 4635834.453, 485007.550 4...

3 POLYGON ((541716.775 4675243.268, 541217.579 4...

4 POLYGON ((583896.043 4909643.187, 583891.200 4...

Name: geometry, dtype: geometry

While this column can contain a variety of different datatypes, each entry will typically be a **Point**, **LineString**, or **Polygon**.



The "geometry" column in our dataset contains 2983 different Polygon objects, each corresponding to a different shape in the plot above.

In the code cell below, we create three more GeoDataFrames, containing campsite locations (**Point**), foot trails (**LineString**), and county boundaries (**Polygon**).

In [9]:

*# Campsites in New York state (Point)*

POI\_data = gpd.read\_file("../input/geospatial-learn-course-data/DEC\_pointsinterest/DEC\_pointsinterest/Decptsofinterest.shp")

campsites = POI\_data.loc[POI\_data.ASSET=='PRIMITIVE CAMPSITE'].copy()

*# Foot trails in New York state (LineString)*

roads\_trails = gpd.read\_file("../input/geospatial-learn-course-data/DEC\_roadstrails/DEC\_roadstrails/Decroadstrails.shp")

trails = roads\_trails.loc[roads\_trails.ASSET=='FOOT TRAIL'].copy()

*# County boundaries in New York state (Polygon)*

counties = gpd.read\_file("../input/geospatial-learn-course-data/NY\_county\_boundaries/NY\_county\_boundaries/NY\_county\_boundaries.shp")

Next, we create a map from all four GeoDataFrames.

The plot() method takes as (optional) input several parameters that can be used to customize the appearance. Most importantly, setting a value for ax ensures that all of the information is plotted on the same map.

In [10]:

*# Define a base map with county boundaries*

ax = counties.plot(figsize=(10,10), color='none', edgecolor='gainsboro', zorder=3)

*# Add wild lands, campsites, and foot trails to the base map*

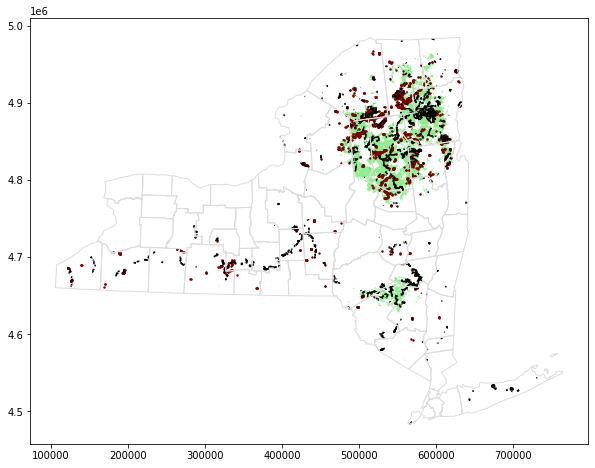
wild\_lands.plot(color='lightgreen', ax=ax)

campsites.plot(color='maroon', markersize=2, ax=ax)

trails.plot(color='black', markersize=1, ax=ax)

Out[10]:

<AxesSubplot:>



linkcode

It looks like the northeastern part of the state would be a great option for a camping trip!

Your turn

This feels complex at first, but you've already learned enough to do important analysis. See for yourself as you [**identify remote areas**](https://www.kaggle.com/kernels/fork/5832167) of the Philippines where a non-profit can expand its operations.