

Assignment 12: Vector Analysis with GeoPandas

(60 Points Total)

Data available under [Resources>Portland Data](#).

The `portland_data.gpkg` GeoPackage contains three data layers for Portland, Oregon:

neighborhoods: boundaries of Portland neighborhoods

park_trees: mapped trees in parks

street_trees: mapped trees along streets

The **neighborhood** boundaries were obtained from the City of Portland (<https://www.portlandoregon.gov/28130>) while the **tree** inventory data were obtained from the City of Portland Office of Parks & Recreation (<https://www.portlandoregon.gov/parks/article/433143>).

To complete this exercise, you will need to have set up a **conda environment** with the following: (1) **NumPy**, (2) **Pandas**, (3) **GeoPandas**, (4) all dependencies for **GeoPandas**, (5) **matplotlib**, and (6) **contextily**. In this exercise, you will only work with the **park_trees** and **neighborhoods** datasets.

Description of Problem

Produce code to complete the requested tasks. Deliver code in a Python or plain text file.

Task 1: Read in the **park_trees** and **neighborhoods** data layers as GeoPandas DataFrames. **(6 Points)**

Task 2: Use `head()` to plot the first set of records in both datasets. **(6 Points)**

Task 3: Create a map of the **neighborhoods** data. You do not need to map any attributes to the feature colors. **(6 Points)**

Task 4: Map the **neighborhoods** data as borders only over a base map. **(6 Points)**

Task 5: Count the number of **park_trees** in each species within each **neighborhood**. Add these counts to the GeoPandas DataFrame of the **neighborhoods**. This can be accomplished with GeoPandas. Have a look at this post: **(18 Points)**

<https://gis.stackexchange.com/questions/306674/geopandas-spatial-join-and-count>.

Task 6: Create a figure with four map layouts where color is used to show the count of species in each of four genera: *Quercus*, *Acer*, *Ulmus*, and *Pseudotsuga*. The layout should include the following: **(18 Points)**

- Two rows and two columns.

- A title for each subplot that provides the name of the genus being mapped.
- Legends for each map.
- A color ramp defined for each map. You can use the same color ramp if you'd like.
- The axes labels turned off.
- The neighborhood boundaries displayed as outlines above the choropleth layer.

Deliverables

- Code in Python or plain text file.