Fairfax County School Facilities

<https://data-fairfaxcountygis.opendata.arcgis.com/datasets/school-facilities>

<https://www.fairfaxcounty.gov/maps/open-geospatial-data> 🡪 “School Facilities”

“Schools, centers and administrative support locations for Fairfax County public schools.” Last Updated: November 6, 2021 (Total Rows = 257)

Alternative location: <https://data-fairfaxcountygis.opendata.arcgis.com/datasets/Fairfaxcountygis::school-facilities/about>

Assignment:

The purpose of assignment 3 is to help students (1) understand the techniques to aggregate geospatial points to polygons and (2) be able to publish geo-social data through the web.

Follow the steps:

Table:

|  |  |  |
| --- | --- | --- |
| Status | TASK | Notes |
|  | 1. ~~For the Fairfax County (Virginia), collect 200 records of geo-social point data from any websites or APIs. For example, it could be geo-tagged Twitter, Yelp, Meetup, crime open data, Census data, or any other types of data. The data should be~~***~~point data~~***~~with longitude & latitude values. Save it as "assignment3.csv".~~ 2. ~~Under the "~~**~~reference\_data~~**~~" folder of our course Git repository, you can find "~~*~~Fairfax\_ZIP\_Codes.geojson~~*~~" file. This data is polygon data of ZIP code boundaries in Fairfax, VA (downloaded from Fairfax Open Data Portal).~~ 3. Use any of R, Python, QGIS, or ArcGIS (or whatever tool that you're familliar with) to aggregate the number of points from "assginment3.csv" to the ZIP code polygons (e.g., if 10 points from your point data fall into a ZIP code polygon on the map, the polygon ID has 10 for the count of the points). Save the result as "aggregation.geojson" -- this should be basically a polygon layer where the number of data points per each polygon is one of the polygon's attributes. 4. Load both the "assignment3.csv" and "aggregation.geojson" on QGIS (or any other tools) along with the base map, so the map visualization shows both polygons and points on top of the base map. I recommend to use QGIS for the visualization, because it is easier to export layers as web-based files. 5. Color code the ZIP code polygons based on the number of points -- the color coding should be based on gradiation of color where darker colors indicates a higher number of points in the polygon. 6. Export the online map that you made as JavaScript-embedded HTML files (either Leaflet or OpenLayer-based). The exported files will be stored as a folder (with many files in it, including the `index.html' file). 7. Create a 1-page report (strict on the number of pages to 1 page) that briefly explains the meanings of the data visualization and the color-coding. 8. Compress (1) the folder that includes web-based map visulaization files, (2) assignment3.csv, (3) aggregation.geojson, (4) data aggregation code or App file, and (5) your 1-page report, and submit the compressed file (e.g., ZIP) to BlackBoard. |  |

**Submission**: submit (1) "assignment3.csv" (1 CSV file), (2) "aggregation.geojson" (1 GeoJSON file), (3) your code or app file that is used to aggregate the points to the polygons (1 file: either R/Python scripts, QGIS, or ArcGIS file), (4) a folder that contains the exported web files, and (5) 1-page report (1 PDF file). Compress all these files as a ZIP file and submit the ZIP file through BlackBoard.

**Grading Criteria:**

* Whether the point data (200 records) is curated correctly for Fairfax county as the "assignment3.csv" file (20)
* Whether the point data aggregation scripts (or an App file) run correctly (10)
* Whether the aggregated polygons are correctly stored as "aggregation.geojson" (20)
* Whether the HTML files are generated correctly to visualize the polygon layer (20)
* Whether the color-coding was done reasonably on the web (10)
* Whether the one-page report provides a reasonable explanation about the visualization (20)
* If the number of report pages is more than one page, the points will be deducted (-5).