

**Andrew Bashorum**

**C00238900**

**4<sup>th</sup> year project**

## **Chapter 1**

**Tecnologies used:** Qgis, Qgis database manager, PGAdmin4, Postgres, Postgis, shp2pgsql, pyproj, geopy, psycopg2, matplotlib

**Aim:** To query the NPS dataset given a geocoded address.

**External courses attended:** <https://www.udemy.com/course/introduction-to-spatial-databases-with-postgis-and-qgis/learn/lecture/9313440?start=285#overview>

### **Steps involved:**

I first set up a PostGIS database which can store a table with a cropped version of the NPS dataset inside. I did this using Postgres-2.5. Which allowed me to use PGAdmin4 to create a new database. I then added postgis as an Extension to this database.

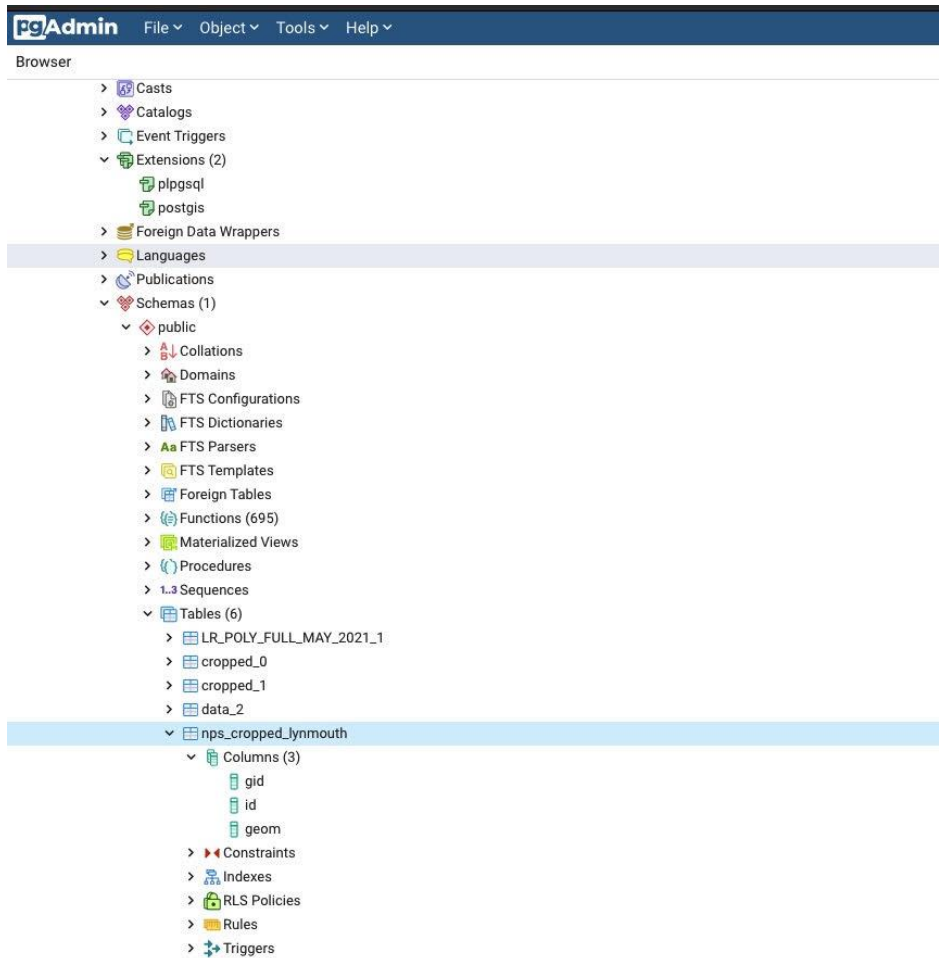


Fig:1

I then used shp2pgsql through the Mac ios command line (fig2) to add my new table with the NPS data to this database. This had to be used as the data was stored in a shape file

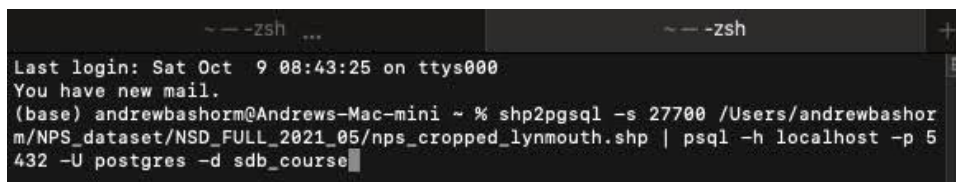


Fig:2

With the table now added, I had to import the database into QGIS and also initialize the table there. Allowing me to verify the data and access it in Python code.



Fig:3

I then began planning a writing a python script. To query this database. I used three separate classes.

Geometry.py responsible for math and geometry methods which can then be accessed by the other classes.

Houses.py which was responsible for geocoding a given address or address list, transforming the output into the correct coordinate reference system then initializing a dictionary with said output as members. I used GoogleV3, a module from geopy to geocode the addresses. I also used Proj from pyproj to transform the coordinates.

Sites.py was responsible for querying the database and forming a list of dictionaries. Inside each dictionary was the geometry for each site queried. It does this using the data gathered from houses.py.

I queried the database using the following statement

```
do = """SELECT ST_AsText(geom) FROM public."nps_cropped_lynmouth" WHERE
ST_Contains(ST_AsText(geom), ST_GeomFromText('POINT('' + str(
    x) + " " + str(y) + "')'))"""
```

I then processed the output. I once again transformed the output into the correct coordinate reference system (later to be used elsewhere). The processed data was then stored.

I then used matplotlib to plot the found sites. My output is shown in fig4 below.

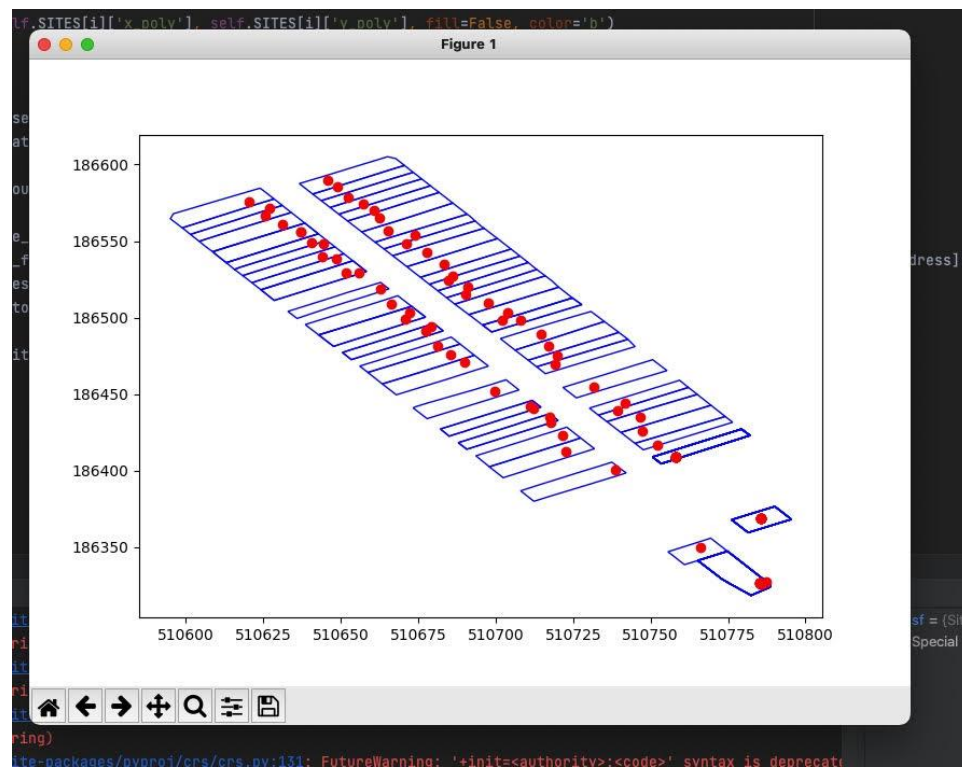


Fig:4