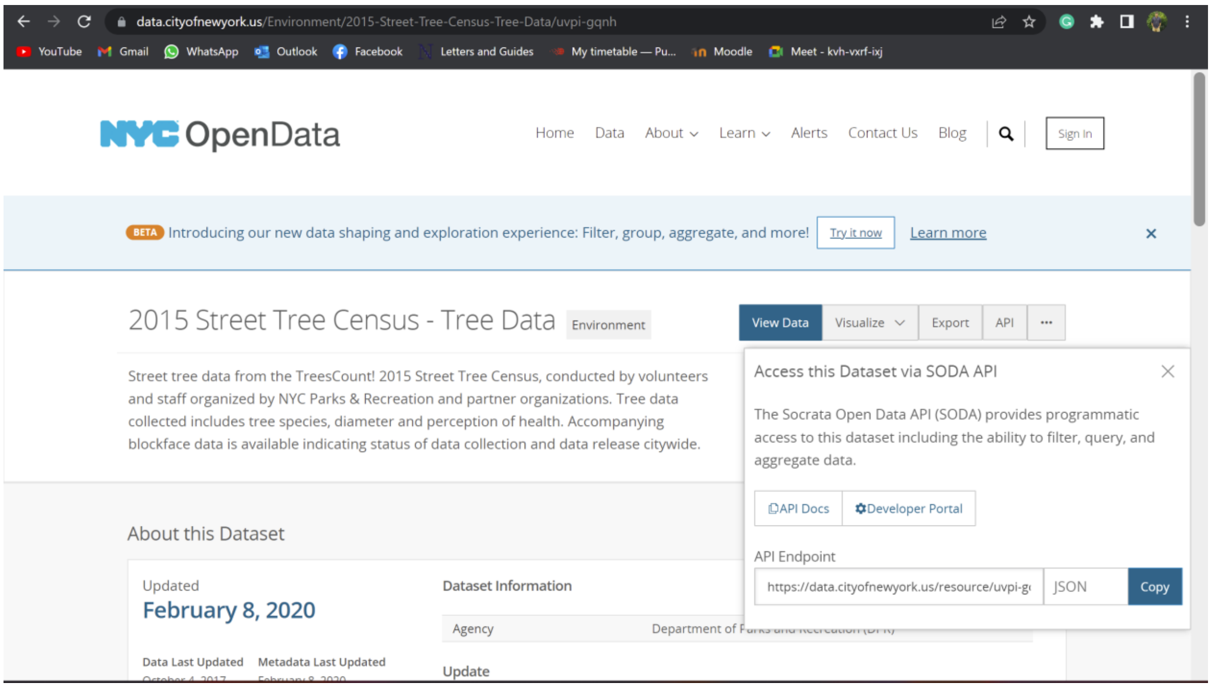
Project Journal

Content

Dataset Selection:

As per the group discussion done among Group-6, I searched and selected a dataset from an Open source website which is data.cityofnewyork.us which contains many real values gathered from datasets, maps, or images. The purpose of selecting Tree Census data was to gain insights into how natural vegetation is spread across the city of New York and how it may affect the decision of newcomers when deciding to accommodate, purchase a property, or for any other research purposes. The dataset consists of various variables (42 in total) that provide information on the tree's species, its condition, place or coordinates the tree is standing, its effects, and problems caused to trees.



Introducing our new data shaping and exploration experience: Filter, group, aggregate, and more!

2015 Street Tree Census - Tree Data Environment

Street tree data from the TreesCount! 2015 Street Tree Census, conducted by volunteers and staff organized by NYC Parks & Recreation and partner organizations. Tree data collected includes tree species, diameter and perception of health. Accompanying blockface data is available indicating status of data collection and data release citywide.

Access this Dataset via SODA API

The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data.

API Endpoint

https://data.cityofnewyork.us/resource/uvpi-g

About this Dataset

Updated
February 8, 2020

Dataset Information

Agency: Department of Parks and Recreation (DPR)

Data Last Updated | Metadata Last Updated

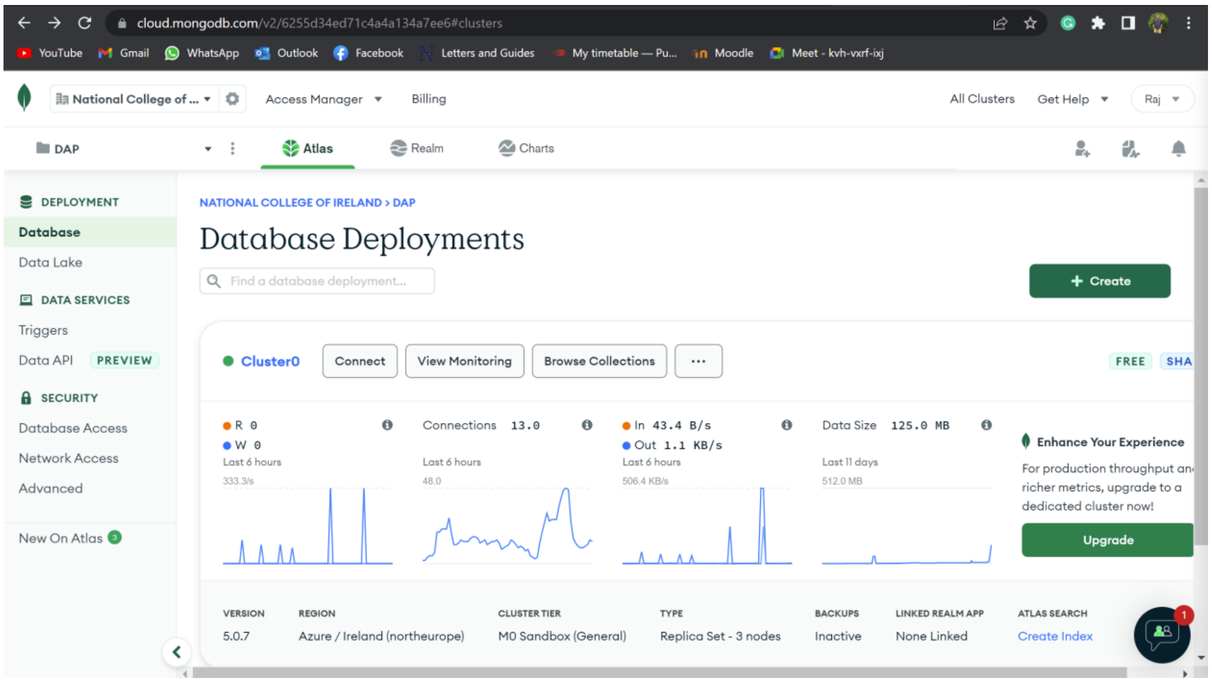
Update

Data Extraction Method

Using websites SODA API, I have extracted desired data in JSON format and stored it into a data frame using Python's pandas "read_json" method. The extracted data had 1000 records and is suitable for making further analysis.

Setting up NoSQL MongoDB

Our team had decided to create one shared Cluster on MongoDB's Atlas server where our IP addresses were stored. We had selected Azure as our cloud Provider and Ireland as our region. With the help of MongoDB Compass, a GUI tool we could view the Database and Collection we created programmatically. To connect our newly created cluster we had to select the package method as python to generate desired connection string so that we can connect to MongoDB programmatically by Python's pymongo and MongoClient libraries. The entire process of setting up and configuring MongoDB took around 4-5 hours including the time required for research.



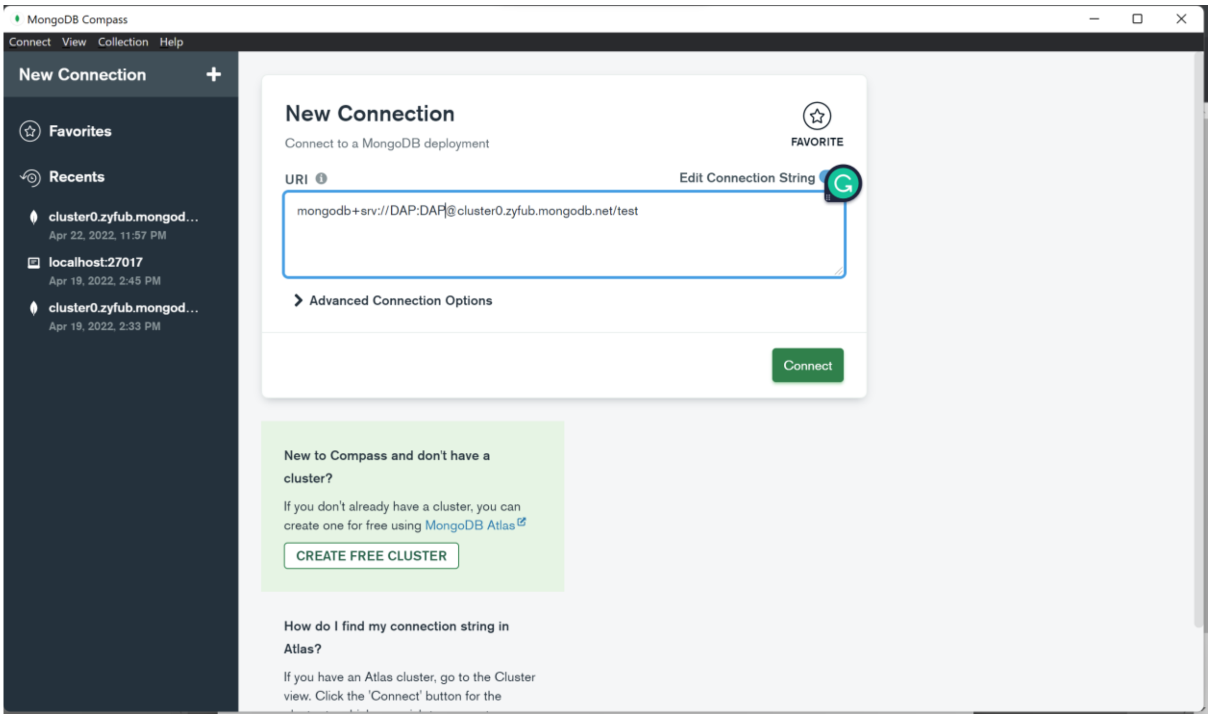
R 0
W 0
Last 6 hours
333.3/s

Connections 13.0
Last 6 hours
48.0

In 43.4 B/s
Out 1.1 KB/s
Last 6 hours
506.4 KB/s

Data Size 125.0 MB
Last 11 days
512.0 MB

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED REALM APP	ATLAS SEARCH
5.0.7	Azure / Ireland (northeurope)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index



cluster0.zyfub.mongod...
Apr 22, 2022, 11:57 PM
localhost:27017
Apr 19, 2022, 2:45 PM
cluster0.zyfub.mongod...
Apr 19, 2022, 2:33 PM

New Connection
Connect to a MongoDB deployment

URI
mongodb+srv://DAP:DAP@cluster0.zyfub.mongodb.net/test

New to Compass and don't have a cluster?

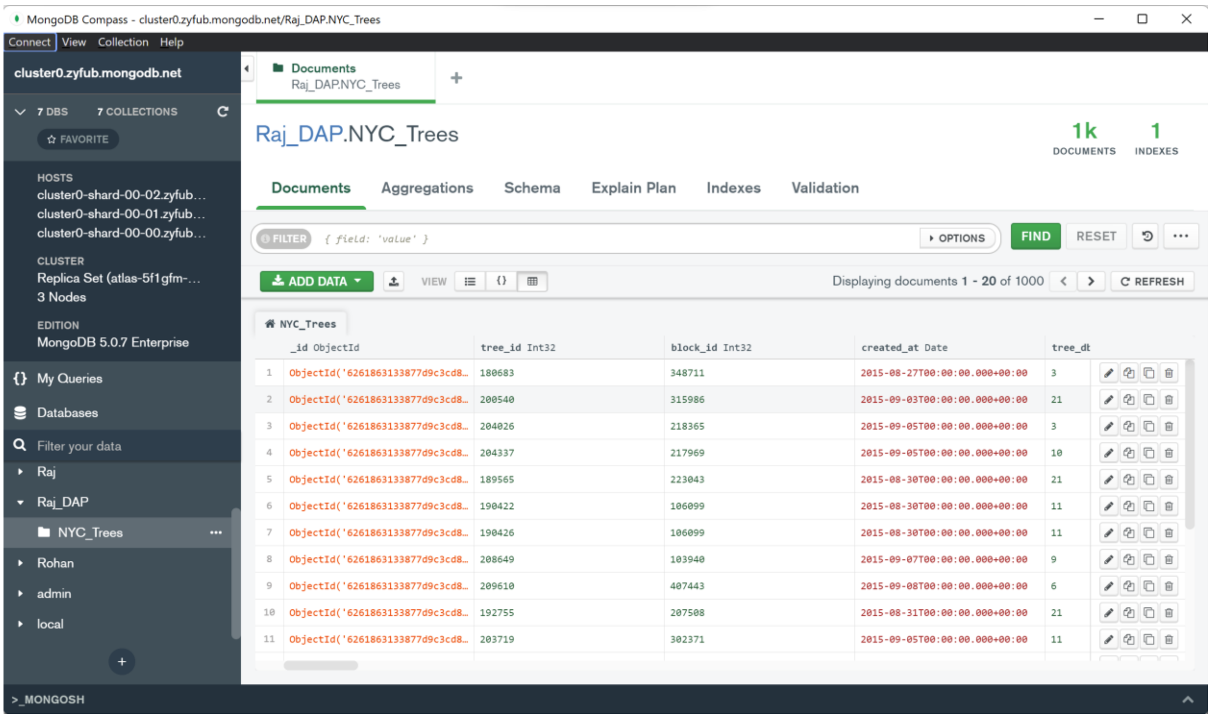
If you don't already have a cluster, you can create one for free using MongoDB Atlas

How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the

Connection to MongoDB

I had established the connections to MongoDB using Python's pymongo libraries. The data extracted in JSON format was passed to a particular collection (Raj_DAP) which belongs to a database named as 'Raj'. Data inserted into the collection at once using insert_many function


MongoDB Compass - cluster0.zyfub.mongodb.net/Raj_DAP.NYC_Trees
Connect View Collection Help
cluster0.zyfub.mongodb.net
7 DBS 7 COLLECTIONS
FAVORITE
HOSTS
cluster0-shard-00-02.zyfub...
cluster0-shard-00-01.zyfub...
cluster0-shard-00-00.zyfub...
CLUSTER
Replica Set (atlas-5f1gfm-...
3 Nodes
EDITION
MongoDB 5.0.7 Enterprise
My Queries
Databases
Filter your data
Raj
Raj_DAP
NYC_Trees
Rohan
admin
local
>_MONGOSH
Documents
Raj_DAP.NYC_Trees
Raj_DAP.NYC_Trees
1k DOCUMENTS
1 INDEXES
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 20 of 1000
REFRESH
NYC_Trees

	_id ObjectId	tree_id Int32	block_id Int32	created_at Date	tree_dt
1	ObjectId('6261863133877d9c3cd8…	180683	348711	2015-08-27T00:00:00.000+00:00	3
2	ObjectId('6261863133877d9c3cd8…	200540	315986	2015-09-03T00:00:00.000+00:00	21
3	ObjectId('6261863133877d9c3cd8…	204026	218365	2015-09-05T00:00:00.000+00:00	3
4	ObjectId('6261863133877d9c3cd8…	204337	217969	2015-09-05T00:00:00.000+00:00	10
5	ObjectId('6261863133877d9c3cd8…	189565	223043	2015-08-30T00:00:00.000+00:00	21
6	ObjectId('6261863133877d9c3cd8…	190422	106099	2015-08-30T00:00:00.000+00:00	11
7	ObjectId('6261863133877d9c3cd8…	190426	106099	2015-08-30T00:00:00.000+00:00	11
8	ObjectId('6261863133877d9c3cd8…	208649	103940	2015-09-07T00:00:00.000+00:00	9
9	ObjectId('6261863133877d9c3cd8…	209610	407443	2015-09-08T00:00:00.000+00:00	6
10	ObjectId('6261863133877d9c3cd8…	192755	207508	2015-08-31T00:00:00.000+00:00	21
11	ObjectId('6261863133877d9c3cd8…	203719	302371	2015-09-05T00:00:00.000+00:00	11

Data Retrieval from MongoDB

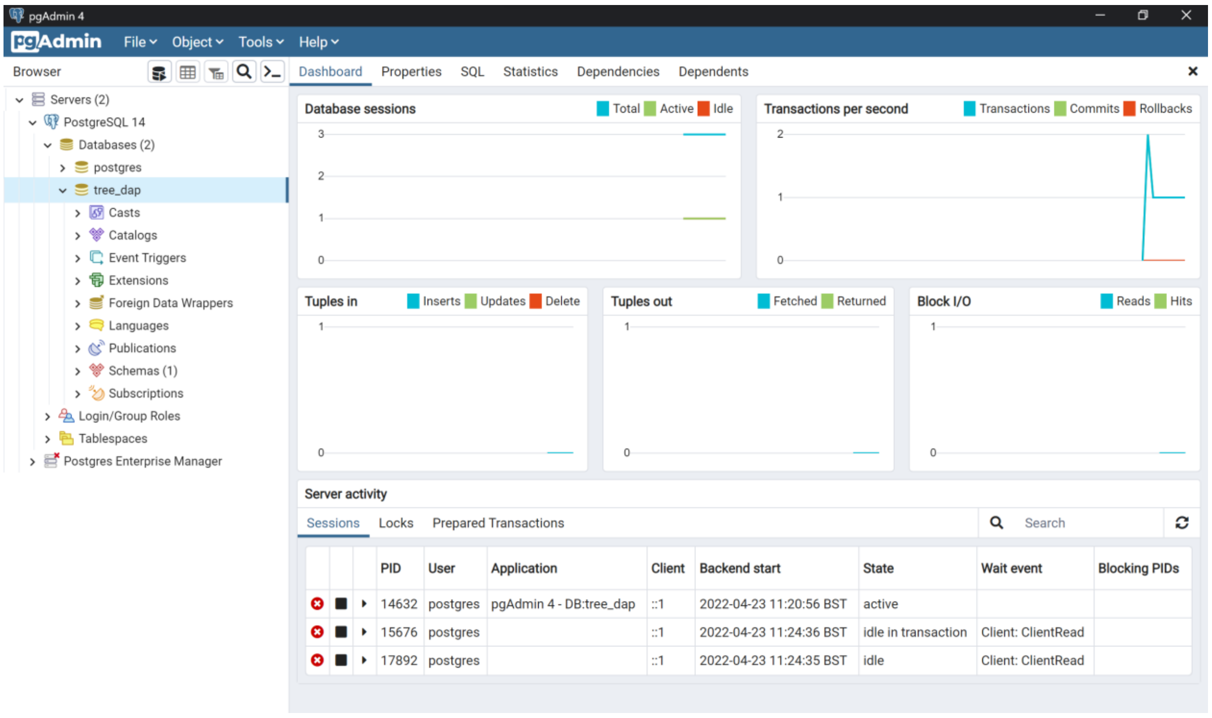
Using Python's .find() function. I was able to store the value into the cursor and that value was passed on to Python's data frame which I named as df_Tree.

Data Cleaning and structuring the final data frame:

After the data was extracted and stored as Python's data frame, The original dimension came out to be 1000 rows and 46 columns. As part of the data cleaning, I removed the null values and unnecessary columns. The original dataset had a maximum of 29 null values which is 2.9% of the overall dataset. So instead of replacing null values, I decided to go ahead and drop the null values. After removing null values I removed the column I didn't require as a part of my project and created a subset of that called df_Tree_Subset.

Configuring and storing the data frame

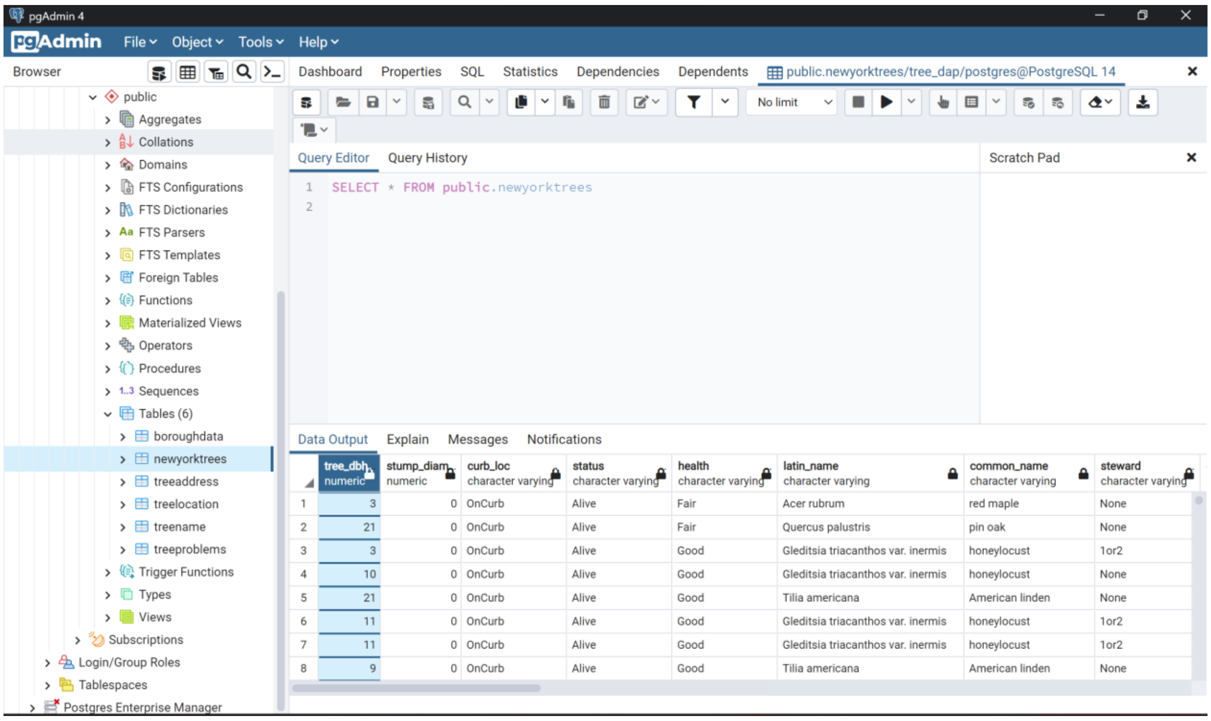
As part of the process, I installed and configured PostgreSQL on my system and also installed the pgAdmin4 application in order to get a visual presentation of how data is stored with relevant schemas. It took me an entire day to perform many trial and error actions in order to successfully install the application appropriately. The problems faced will be mentioned in other section of this journal.



PID	User	Application	Client	Backend start	State	Wait event	Blocking PIDs
14632	postgres	pgAdmin 4 - DB:tree_dap	::1	2022-04-23 11:20:56 BST	active		
15676	postgres		::1	2022-04-23 11:24:36 BST	idle in transaction	Client: ClientRead	
17892	postgres		::1	2022-04-23 11:24:35 BST	idle	Client: ClientRead	

Storing data into PostgreSQL

The cleaned data stored in df_Tree_Subset was stored in PostgreSQL using Python's psycopg2 library. First using appropriate username, password, host, port, and database details connections were established with proper exception handling. SQL commands for dropping the existing table and creating a new one were executed programmatically by Python's code. The data frame already achieved 1NF and then as an attempt to achieve 2NF I had created and insert data into smaller tables into Postgres. This process took around 2 days to achieve as the environment and technology were newly introduced for me.



```
SELECT * FROM public.newyorktrees
```

	tree_dbh numeric	stump_diam numeric	curb_loc character varying	status character varying	health character varying	latin_name character varying	common_name character varying	steward character varying
1	3	0	OnCurb	Alive	Fair	Acer rubrum	red maple	None
2	21	0	OnCurb	Alive	Fair	Quercus palustris	pin oak	None
3	3	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	honeylocust	1or2
4	10	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	honeylocust	None
5	21	0	OnCurb	Alive	Good	Tilia americana	American linden	None
6	11	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	honeylocust	1or2
7	11	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	honeylocust	1or2
8	9	0	OnCurb	Alive	Good	Tilia americana	American linden	None

Retrieving data from PostgreSQL

Using Python's sqlalchemy library I retrieved the master table from PostgreSQL using SQL's SELECT command and stored it into a data frame called TreeFromSQL.

Data Visualization (Individual Level)

As part of Data Visualization I have used/plotted Histogram, Bar chart, Stacked Bar Chart, Pie Chart, Counterplot, Cat plot, Scatterplot, Maps using Folium, Density Map, Scatter Map to visualize data for particular research questions. This stage was most challenging as it took 4-5 hours to understand the data and its composition to visualize it accordingly. The highly time-consuming part was to plot trees on the map using the Folium package as it required a lot of research on how to plot data against given Latitude and Longitude.

Merging code with group members

As part of project requirement we have merged our python code with distinguishing parts such as Data gathering, Sending code to MongoDB and then fetching it back for Data Cleaning and Handling. We then created common database in PostgreSQL and performed same action as we did on per-individual basis. I had clubbed my New York's Trees dataset with New York's property valuation. We formed a conclusion based on it.

Joining data frames

Once we fetched all the data we clubbed our dataset to find if there is any relation between them.

Visualizing merged data frames

I had clubbed my New York's Trees dataset with New York's property valuation. We formed a conclusion based on it.

Committing code to GitHub

Once I have successfully accomplished one phase of python coding I have committed my saved IPYNB file to GitHub repositories named DAP Group 6.

IEEE Report Writing

After gathering the entire data we have created a final report in IEEE format using the Overleaf tool that provides syntax to set appropriate alignments and give references.

Video Presentation

We created PowerPoint Presentation at first and used it in our Video Presentation we recorded on Microsoft Teams. The PowerPoint presentation was a helpful tool to represent the entire project work in the span of few minutes

Problems faced during the entire project

1. While choosing the dataset it took quite a time to conclude how to extract the data programmatically using API so as to avoid using external or local storage to read the JSON file. The python file created for this project will not be required as any external files to read JSON as it is getting data after hitting the appropriate API.
2. Configuring MongoDB Atlas, a cloud-based database took a lot of research time than estimated. Other problems faced were data passed into MongoDB were stored in String Object for all columns even if they were Numeric. The right way of storing the dataset helped me in storing all columns in their respective data types.
3. While setting up PostgreSQL, the system faced tons of issues with respect to the Master password set as python was not able to distinguish between password string and connection string. To solve this issue I had to entirely remove existing PostgreSQL and re-install it again with the appropriate password.
4. While inserting data into PostgreSQL using SQL query it took plenty amount of time and research to use appropriate queries to match the columns already declared when tables were created.
5. Given the dataset had great quality and useful data, I had a vision of making the right use of it by plotting data over the map. However, thinking out of the box led to an excess of research on how to plot data against given latitude and longitude. After performing tons of trial and error, I was successfully able to plot Trees with proper names as a popup functionality over the map of New York.