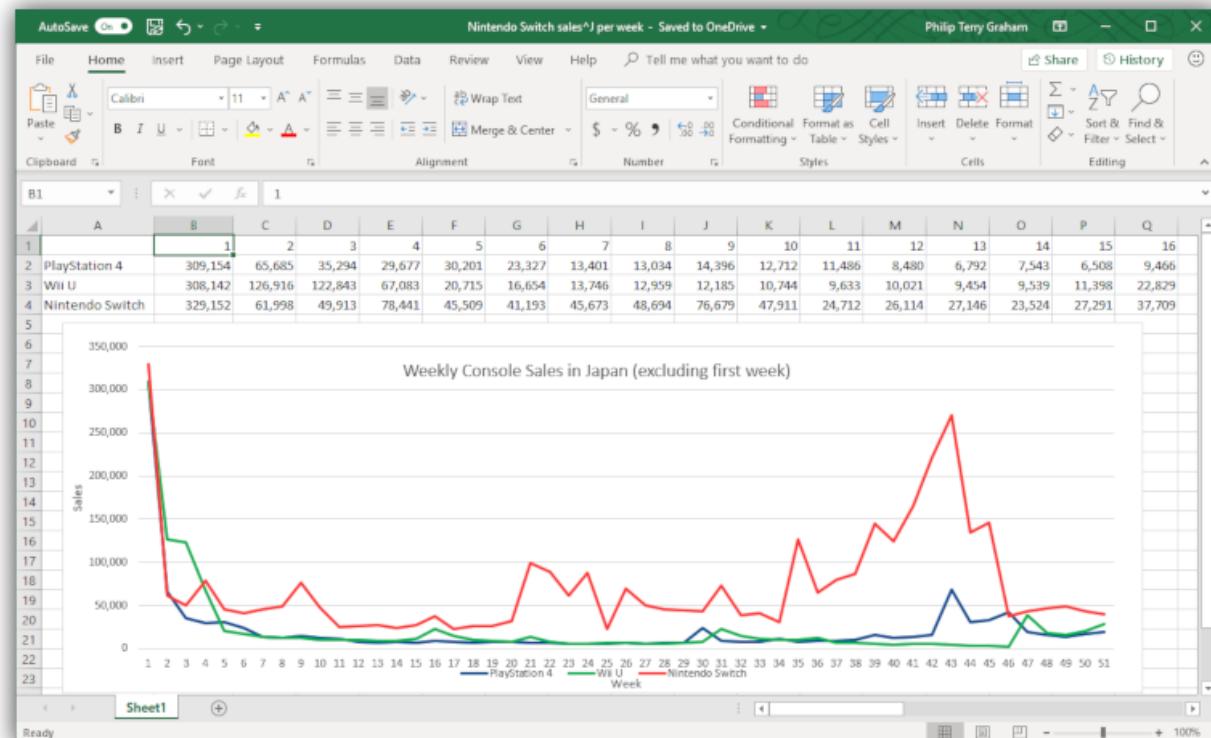


TRANSPORTATION DIVISION DATA ANALYTICS TOOLKIT





- **Data Processing**
 - Excel
 - Python
 - R
- **Geospatial Data Processing**
 - ArcGIS
 - QGIS
- **Interactive Chart**
 - Tableau
 - Plotly
- **Interactive Map**
 - ArcGIS Online
- Carto
- Mapbox GL JS
- **Database**
 - MS SQL Server
 - SQLite / SpatiaLite
 - PostgreSQL / PostGIS
 - DBeaver
- **Other Tools**
 - GitHub
 - Visual Studio Code
 - Geosupport
 - DigitalOcean



- Data analytics software by Microsoft
- Creating, reading, writing, manipulating, analyzing, and visualizing relatively small datasets
- Formulas (SUM, IF, VLOOKUP, etc.)
- Pivot table
- Static chart / map
- Data Analysis Tools
- Macros with VBA



ANACONDA®



SPYDER



A screenshot of the Spyder IDE interface. On the left, there is a code editor window displaying Python code for reading CSV files and performing data manipulation. On the right, there is a variable explorer window showing variables like 'i' (str), 'path' (str), and 'tp' (list). Below the code editor, a console window shows the execution of the code, including file imports and error messages related to file paths.

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\mayij\Desktop\DOC\GITHUB\td-acaspi\blueprint.py
fare.py turstile.py landuse.py blueprint.py
1 #%% Settings
2 import requests
3 import pandas as pd
4 import numpy as np
5 import geopandas as gpd
6
7
8 pd.set_option('display.max_columns', None)
9 path='C:/Users/mayij/Desktop/'
10 path='C:/Users/Y_Ma2/Desktop/'
11
12 apikeypd.read_csv(path+'GitHub\td-acaspi\secrets.csv',dtype=str).loc[0,'value']
13 usernm=pd.read_csv(path+'GitHub\td-acaspi\secrets.csv',dtype=str).loc[1,'value']
14 passwd=pd.read_csv(path+'GitHub\td-acaspi\secrets.csv',dtype=str).loc[2,'value']
15
16 p=(http://'+str(usernm)+':'+str(passwd)+'@dcproxy1.dcp.nycnet:8080',
17 'https://'+str(usernm)+':'+str(passwd)+'@dcproxy1.dcp.nycnet:8080')
18
19
20
21 #%% County list
22 tp=[]
23 for i in ['times square','dbk','broadway junction','bxhub','fordham','morris park','e14st','e23st','e125st','w125st']:
24     k=pd.read_csv(path+'Blueprint/travelshed/'+str(i)+'/'+str(i)+'.csv')
25     k['county']=str(x)[0:5] for x in k['tractid']]
26     k=k[['county']].unique()
27     tp=tp.append(k)
28 tp=tp.unique()
29
30
31 #%% All Locations
32 df=pd.DataFrame()
33 quadstateclipped=gpd.read_file(path+'Blueprint/travelshed/quadstateclipped.geojson')
34
35 for i in ['dbk','broadway junction','bxhub','fordham','morris park','e14st','e23st','e125st','w125st','lic','jamaica']:
36     df=pd.concat([df,gpd.read_file(path+'Blueprint/travelshed/'+i+'/'+i+'.geojson')])
37 df=df[['tractid']].reset_index(drop=True)
38 df=pd.concat([df,tp],axis=0,ignore_index=True)
39 df.drop_duplicates(keep='first').reset_index(drop=True)
40 df.to_csv(path+'Blueprint/travelshed/all/all.csv',index=False)
41 df=df.merge(quadstateclipped,df,how='inner',on='tractid')
42 df.to_file(path+'Blueprint/travelshed/all/all.geojson',driver='GeoJSON')
```

- Open-source general-purpose programming language
- Reading, writing, manipulating, analyzing, and visualizing relatively big data
- Reproducibility and transparency
- IDE: Spyder / Jupyter
- Data wrangling: Pandas
- Spatial data: GeoPandas
- Interactive visualization: Plotly
- Dashboard / App: Streamlit / Dash



A screenshot of the RStudio interface. The left pane shows an R script named 'userTrend.R' with code for user trend analysis, including reading a CSV file, creating a subset, and generating a plot. The right pane shows the 'Workspace' tab with the 'userData' dataset loaded, displaying its structure (580 obs. of 5 variables: active, state, etc.). Below the workspace is a plot titled 'Breakdown of Users by Age and State' showing a grid of points where color indicates user activity (red for 0, teal for 1) and size indicates age.

```

#!/usr/bin/Rscript
# User Trend Analysis
# Breakdown of active and non-active users
library(plyr)
library(ggplot2)

userData <- read.csv("userDataTrends.csv")
userData <- subset(userData, select = -c(id, group))
userData$active <- as.factor(userData[,1])

states <- levels(userData$state)

names(userData)
count(userData, "active == 1")
View(userData)

summary(subset(userData, active == 1)$state)
summary(subset(userData, active == 0)$state)

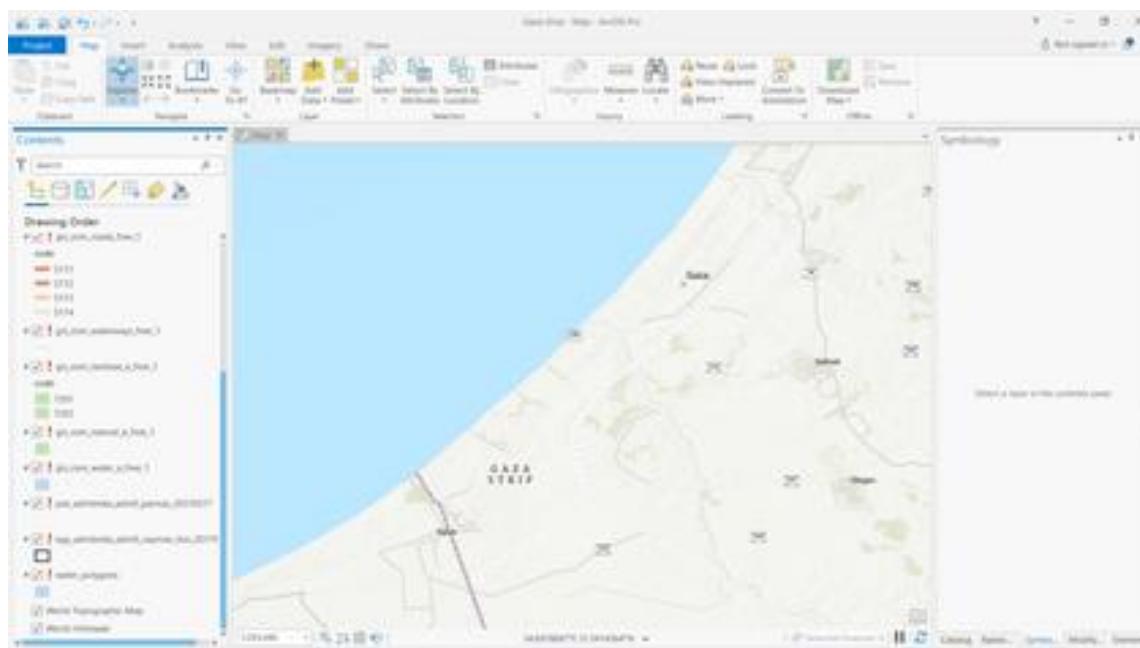
qplot(state, age, color = active, data = userData,
      main = "Breakdown of Users by Age and State") +
  opts(plot.title = theme_text(size = 19))

```

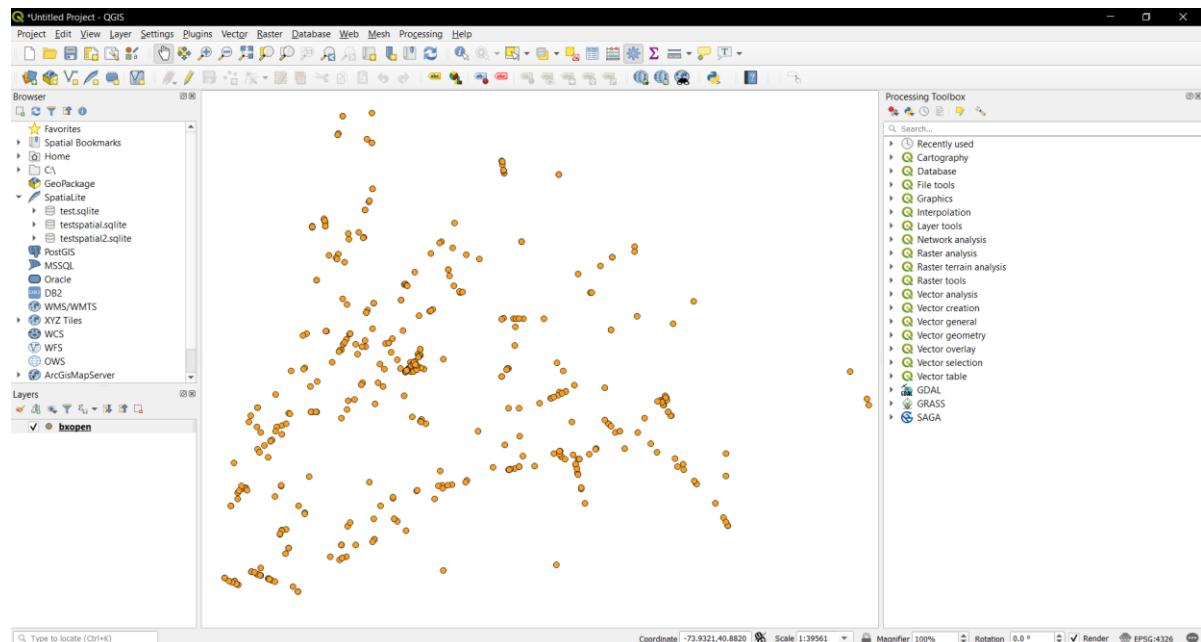
- Open-source programming language optimized for statisticians
- Reading, writing, manipulating, analyzing, and visualizing relatively big data
- IDE: RStudio
- Data wrangling: tidyverse (dplyr)
- Spatial data: sf
- Interactive visualization: Plotly
- Dashboard / App: R Markdown / R Shiny



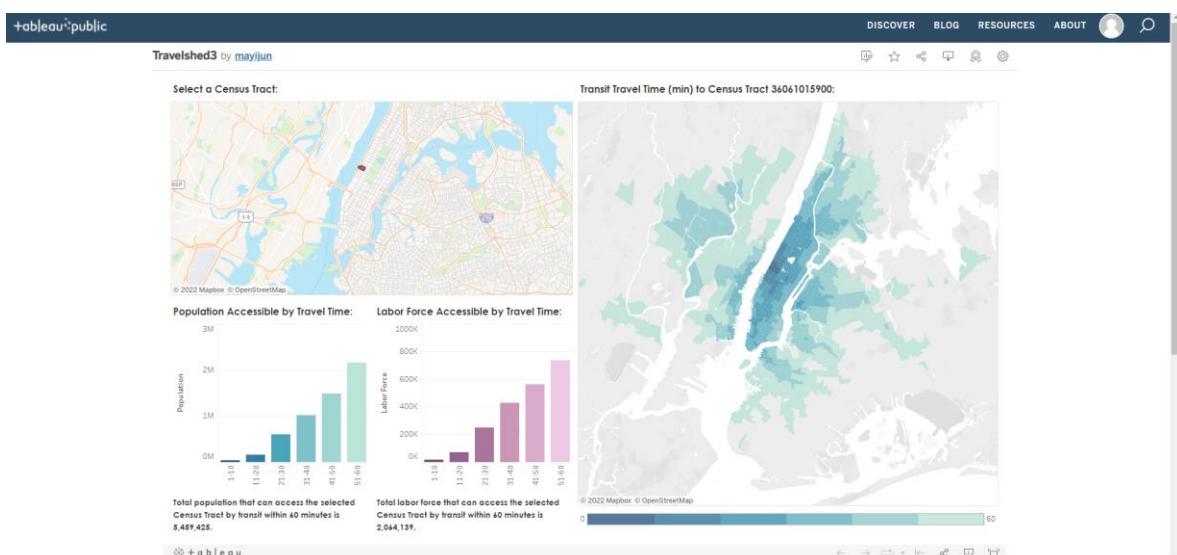
ArcGIS



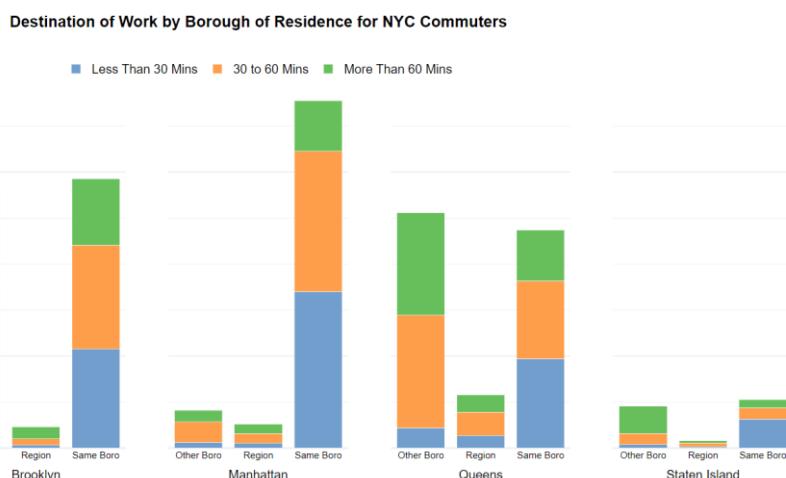
- Proprietary GIS software by ESRI
- Creating, reading, writing, manipulating, analyzing, and visualizing geospatial data
- Data format: Shapefile, Geodatabase, etc.
- ArcGIS Pro replacing ArcMap



- Open-source GIS software
- Creating, reading, writing, manipulating, analyzing, and visualizing geospatial data
- Data format: Shapefile, GeoJSON, etc.
- Toolbox and plugins for geoprocessing



- Data visualization software owned by Salesforce
- Creating dashboards with interactive charts and maps
- Tableau Public (free) for completely public data visualization
- Tableau Desktop / Online (require license) for non-public data



- Open-source data visualization tools
- Creating interactive charts and maps
- Multi-platform graphing libraries: Python, R, JavaScript, etc.
- Dash for creating dashboards / apps
- Hosting platform: GitHub Pages



ArcGIS Online

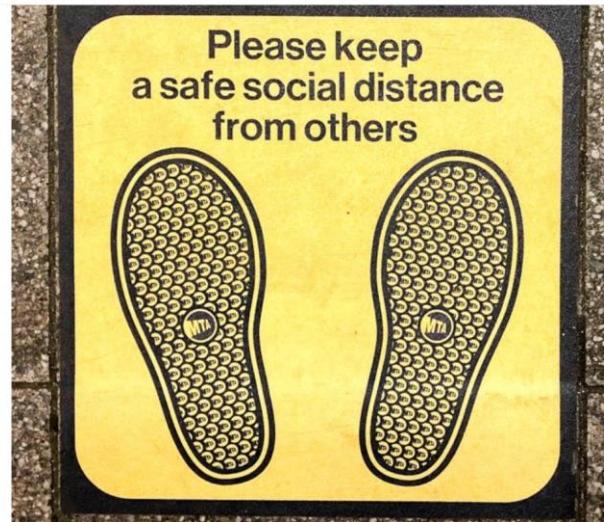
Mapping and analysis: location intelligence for everyone

2020 Travel Trends

2020 Travel Trends

How the COVID-19 pandemic affected mobility in New York City

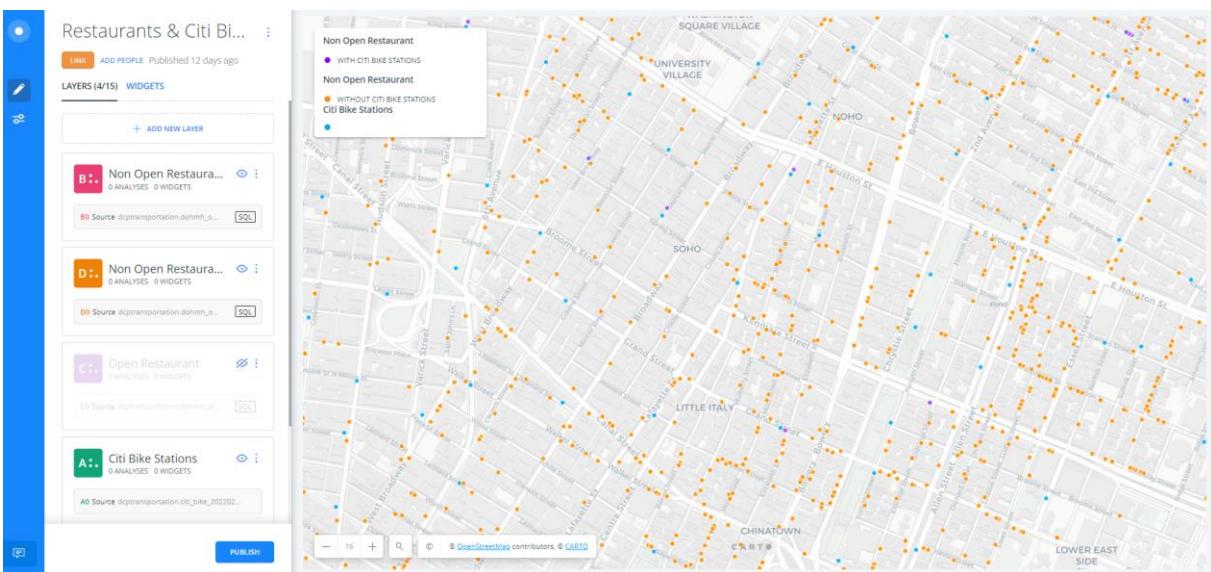
NYC Department of City Planning, March 2021



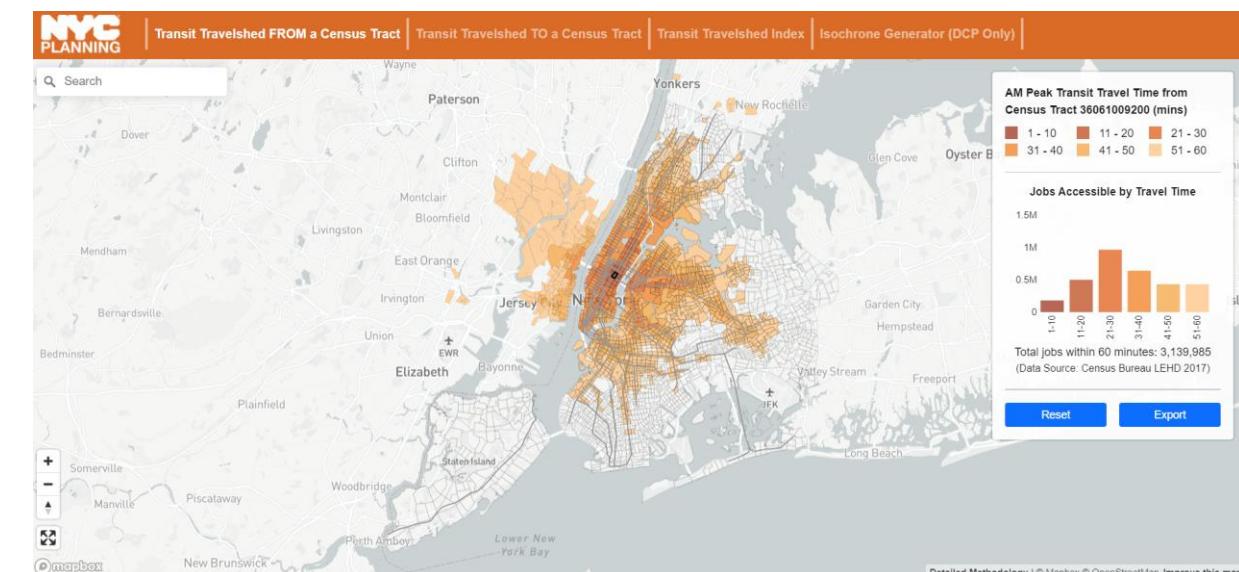
- **Web-based mapping software by ESRI**
- **Creating interactive charts, maps, dashboards, story maps, etc.**
- **Using StoryMaps to organize final report with interactive charts / maps from other platforms embedded**



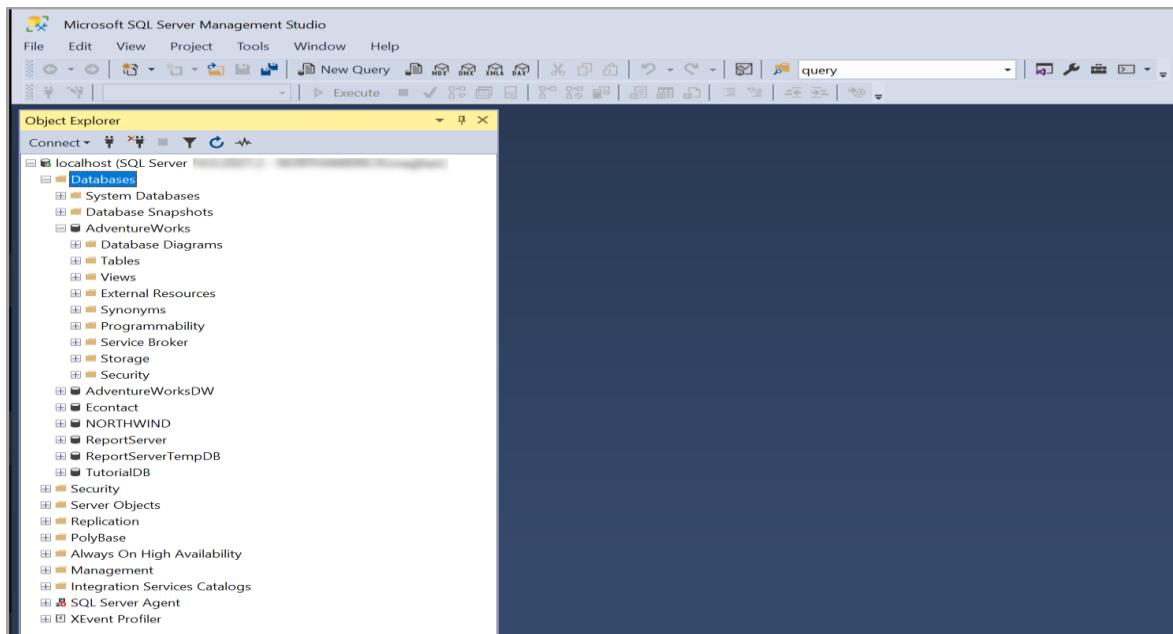
CARTO



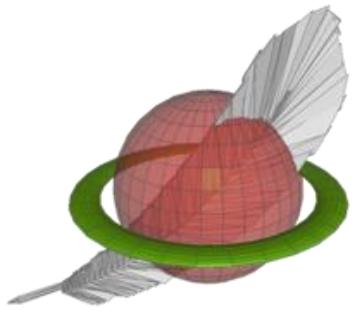
- Web-based mapping platform
- Storing geospatial data and creating interactive maps
- Customization: SQL for data processing, CSS for styling, and HTML for legend editing



- **JavaScript mapping library**
- **Creating interactive maps and highly customized web mapping apps**
- **Editing tool: Visual Studio Code**
- **Languages used: HTML, CSS, and JavaScript**
- **Hosting platform: GitHub Pages**



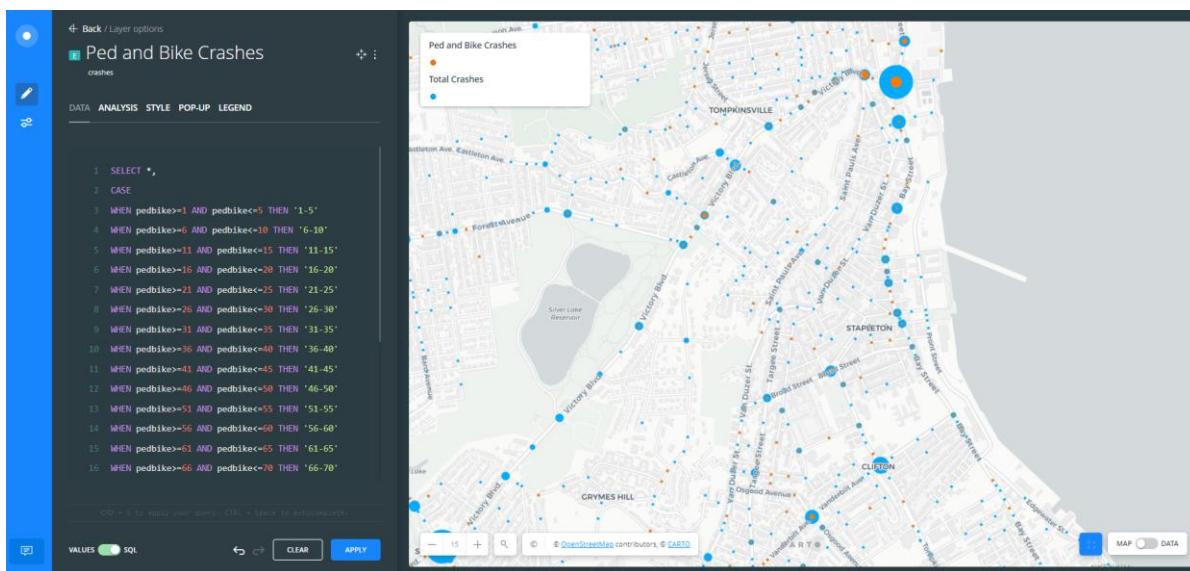
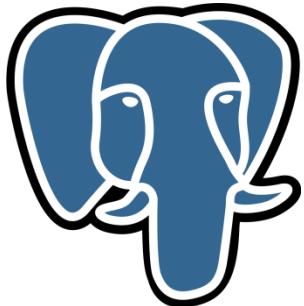
- **Commercial client/server relational database management system by Microsoft**
- **Deployed and managed by ITD**
- **Storing confidential datasets (TLC data)**
- **Interface: SQL Server Management Studio / DBeaver**



The screenshot shows the QGIS DB Manager interface. On the left, the 'Providers' panel lists various spatial databases and providers. In the center, the 'Table' tab displays a table with columns: id, geom, Restaurant Name, and Time of Submission. The table contains 20 rows of data, each representing a restaurant location. The bottom status bar shows coordinates (-73.8311, 40.7985), scale (1:39559), magnification (100%), rotation (0.0°), and EPSG:4326.

		geom	Restaurant Name	Time of Submission
1	1	NULL	Juices for life	07/17/2020 11...
2	2	POINT	Xime deli corp	06/23/2020 12...
3	3	POINT	Fratelli Restaura...	06/26/2020 10...
4	4	NULL	Famosa pizza	07/22/2020 12...
5	5	POINT	Juanito Restaur...	06/28/2020 11...
6	6	POINT	Howl at the Mo...	06/25/2020 04...
7	7	POINT	Excelente Gil Ca...	06/27/2020 04...
8	8	NULL	Downneys Bar an...	06/21/2020 10...
9	9	POINT	Severino	06/21/2020 07...
10	10	POINT	EL SABOR LATI...	06/24/2020 05...
11	11	NULL	Famosa pizza	07/29/2020 11...
12	12	NULL	Famosa pizza	07/29/2020 11...
13	13	POINT	Fratilli's Pizzeria...	06/26/2020 06...
14	14	POINT	Havana Cafe	06/23/2020 07...
15	15	POINT	COPAS LOUNGE	06/27/2020 06...
16	16	POINT	Luke's Lounge	06/23/2020 01...
17	17	POINT	Crosby Caribbe...	06/22/2020 01...
18	18	POINT	Golden Eagle R...	06/22/2020 09...
19	19	POINT	Riverdale Steak ...	06/19/2020 11...
20	20	POINT	CARIDAD EXPRES...	06/21/2020 01...
	21	POINT	Dolcezza Patisse...	06/23/2020 06...

- **SQLite: open-source file-based relational database management system**
- **SpatiaLite: spatial extension to SQLite**
- **Storing large public datasets into single database**
- **Interface: DBeaver / QGIS**

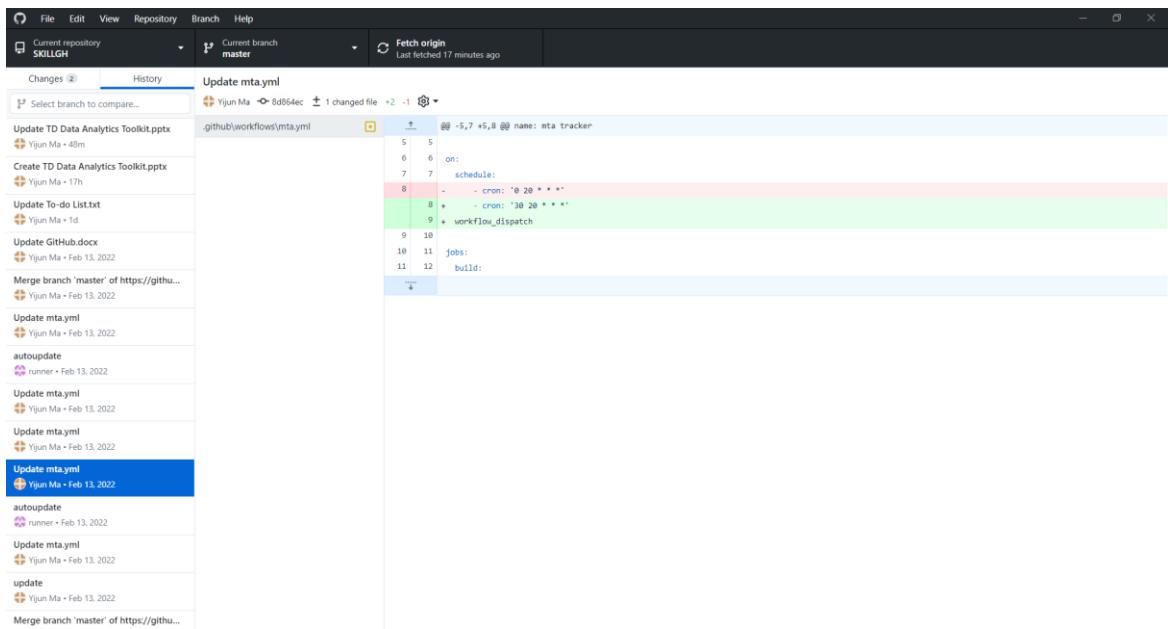
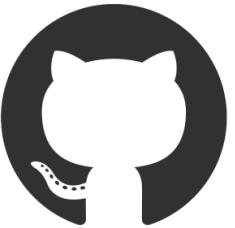


- **PostgreSQL: open-source client/server relational database management system**
- **PostGIS: spatial extension to PostgreSQL**
- **Storing spatial datasets**
- **Interface: Carto**



The screenshot shows the DBeaver application window. At the top, there's a menu bar with File, Edit, Navigate, Search, SQL Editor, Run, Database, Window, Help. Below the menu is a toolbar with various icons. The main area has several tabs: 'PostgreSQL - postgis', 'SQLite - CIT2017.sqlite3', and 'Script - <SQLITE - CIT2017.sqlite3>'. The 'Properties' tab is currently selected, displaying a table titled 'quadstateblock' with columns like 'id', 'geom', 'objectid', 'rec_statedate10', 'rec_countydate10', 'rec_blockdate10', 'rec_gendate10', 'rec_name10', 'rec_align10', and 'rec_avar10'. The table contains numerous rows of data. Below the table, there's a 'Background Tasks' section with a message: 'No operations to display at this time.' At the bottom, there's a status bar with 'EST | en |'.

- Open-source database administration tool
- Supports CSV, MS SQL Server, SQLite, PostgreSQL, MySQL, Oracle, IBM DB2, MS Access, etc.



- **Version control and source code management platform**
- **Storing scripts, hosting web apps, automating updates, etc.**
- **GitHub Desktop: simple tool to pull and push scripts to the cloud**
- **GitHub Pages: place to host interactive charts / maps and web apps for public sharing**
- **GitHub Actions: tool to automate the workflows and update the data regularly**



```
// Add data source
map.addSource(layerContent['sourceid'], { ... });

// Add fill layer
map.addLayer({
  'id': layerContent['layerid'],
  'type': layerContent['layertype'],
  'source': layerContent['sourceid'],
  // 'filter': layerContent['layerfilter'],
  'layout': {
    'visibility': layerContent['layervisibility'],
  },
  'paint': {
    'fill-color': layerColor,
    'fill-opacity': layerContent['layeropacity'],
    'fill-outline-color': layerContent['layeroutlinecolor'],
  },
});

// Add lines
// Set layer contents
var layerContent = {
  'sourceid': 'linsource', // source id
  'sourcetype': 'geojson', // source type
  'sourceadata': 'https://raw.githubusercontent.com/mayijun1203/MLGH/master/mapbox/linetestcat.geojson', // data source
  'sourceinmetrics': true, // only used for gradient line
  'layerid': 'linlayer', // layer id
  'layertype': 'line', // symbology type
  'layername': 'line', // layer name for the legend
  'layerfilter': ['==', 'type', 'Main Line'], // filter
  'layerlinecap': 'round', // line endings: butt/round/square
  'layerlinejoin': 'round', // line joining: bevel/round/miter
  'layervisibility': 'visible', // visibility of the layer
  'layercolor': '#0072BD', // blur line
  'layercolname': 'cat', // category column name
  'layercolorder': 1, // sort order
  'layercolcount': 1 // number of categories
};
```

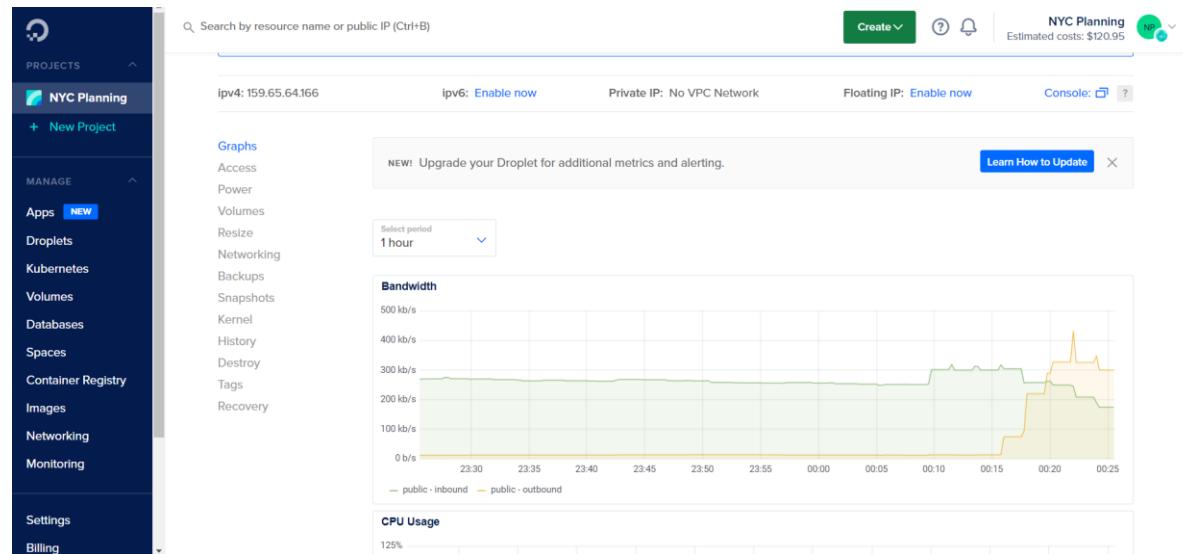
- Source code editor developed by Microsoft
 - Editing scripts for creating web apps and testing the apps locally
 - Supported languages: HTML, CSS, JavaScript, SQL, Python, R, etc.



GEO SUPPORT SYSTEM

The screenshot shows the NYC GeoSupport System interface. At the top left is the "GOAT" logo (Geographic Online Address Translator). At the top right is the "NYC PLANNING GEO SUPPORT" logo. The main header reads "Display Street and Property Information by Address". On the left, there's a vertical menu with options like WELCOME, ADDRESS FUNCTIONS (FUNCTION 1A, FUNCTION 1B, FUNCTION 1E, FUNCTION AP), INTERSECTION FUNCTIONS (FUNCTION 2), STREET SEGMENT FUNCTIONS (FUNCTION 3), STREET STRETCH FUNCTIONS (FUNCTION 3S), and BLOCK & LOT. The main form has fields for "Select a Borough" (Manhattan), "Address Number" (120), "Street or Place Name" (broadway), "Unit" (empty), and "Function Options" (button). Below the form are buttons for Centerline Level Information, City Service Information, Political Information, Property Level Information, Address Range, Map, and Send Feedback. A message box displays "Centerline Level Information for 120 BROADWAY in MANHATTAN". At the bottom, an "Orientation" note says "Address is on the right when facing from PINE STREET to THAMES STREET".

- Geocoding package for NYC addresses
- GOAT: web app for geocoding individual address
- GBAT: batch geocoding multiple addresses
- Geoclient (DOITT) / Geoservice (DCP): geocoding API
- Python-Geosupport: python package to geocode large datasets (especially confidential data)

A screenshot of the DigitalOcean control panel. The top navigation bar includes a search bar, a "Create" button, and a "NYC Planning" project summary. The main content area shows a droplet with IPv4 address 159.65.64.166 and various configuration options like IPv6, Private IP, and Floating IP. A prominent feature is a "Graphs" section displaying a bandwidth usage chart. The chart shows two lines: a green line for "public - inbound" and an orange line for "public - outbound". The x-axis represents time from 23:30 to 00:25, and the y-axis represents bandwidth in kilobits per second (kb/s) from 0 b/s to 500 kb/s. Both lines show a baseline around 250-300 kb/s, with significant spikes occurring between 00:10 and 00:20, reaching up to 400-500 kb/s. Below the chart is a "CPU Usage" section showing a single bar at 125%.

- Cloud computing service provider
- Streaming data, processing massive datasets, hosting web service, etc.
- Connection tool: PuTTY / SSH
- Data transfer tool: WinSCP / Cyberduck