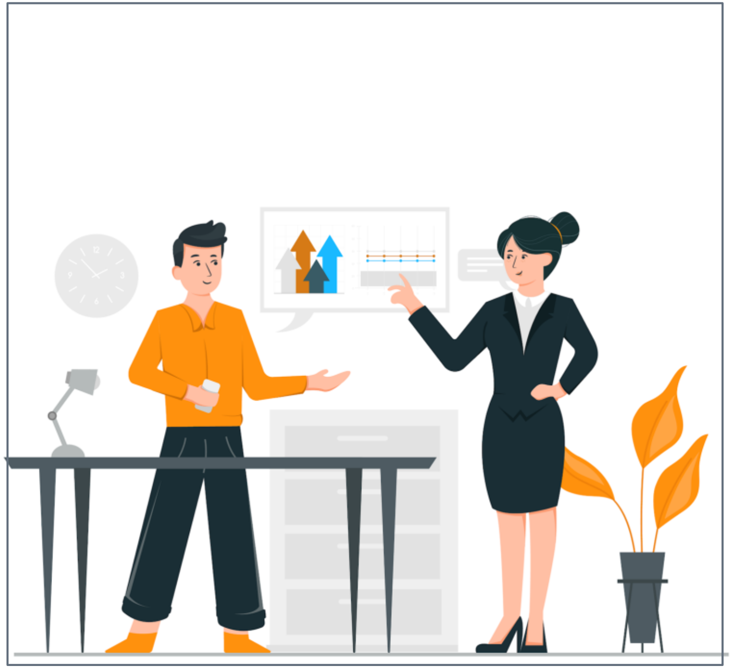
Covid Criminals PROJECT

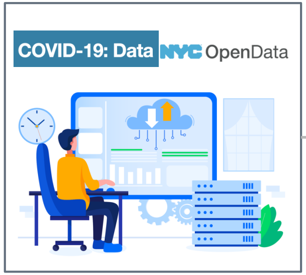
Maryam Torabi Kia
Ahmad Mahroo

Welcome

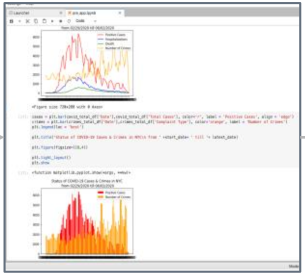
Let’s visualize the relationship between COVID-19 cases and criminal activities in New York City!



The project workflow in one glance

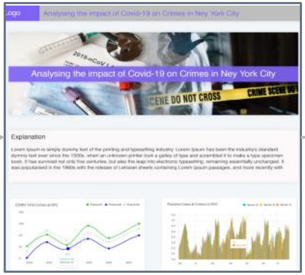


The Sources



```
<Figure size 720x288 with 0 Axes>

cases = plt.bar(covid_total_df['Date'],covid_total_df['Total Cases'], color='r', label = 'Positive Cases', align = 'edge')
crimes = plt.bar(crimes_total_df['Date'],crimes_total_df['Complaint Type'], color='orange', label = 'Number of Crimes')
plt.legend(loc = 'best')

plt.title('Status of COVID-19 Cases & Crimes in NYC\n from ' +start_date+ ' till '+ latest_date)

plt.figure(figsize=(10,4))

plt.tight_layout()
plt.show

<function matplotlib.pyplot.show(*args, **kw)>
```

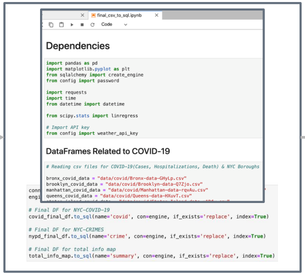
Preliminary Analysis


Logo
Analysing the impact of Covid-19 on Crimes in Ney York City
Analysing the impact of Covid-19 on Crimes in Ney York City
SCENE DO NOT CROSS
CRIME SCENE DO

Explanation

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with

Front-end planning



Dependencies

```python
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import create_engine
from config import password

import requests
import time
from datetime import datetime

from scipy.stats import linregress

# Import API key
from config import weather_api_key
```

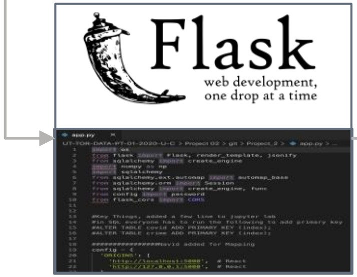
DataFrames Related to COVID-19

```python
# Reading csv files for COVID-19(Cases, Hospitalizations, Death) & NYC Boroughs

bronx_covid_data = "data/covid/Bronx-data-GHyLp.csv"
brooklyn_covid_data = "data/covid/Brooklyn-data-Q7Zjo.csv"
manhattan_covid_data = "data/covid/Manhattan-data-rqvAu.csv"
queens_covid_data = "data/covid/Queens-data-HXuvT.csv"
```

```python
conn
engi

# Final DF for NYC-COVID-19
covid_final_df.to_sql(name='covid', con=engine, if_exists='replace', index=True)

# Final DF for NYC-CRIMES
nypd_final_df.to_sql(name='crime', con=engine, if_exists='replace', index=True)

# Final DF for total info map
total_info_map.to_sql(name='summary', con=engine, if_exists='replace', index=True)
```

ETL

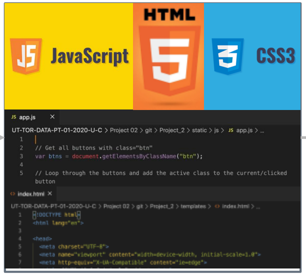
ETL



UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > app.py > ...

```python
import os
from flask import Flask, render_template, jsonify
from sqlalchemy import create_engine
import numpy as np
import sqlalchemy
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.orm import Session
from sqlalchemy import create_engine, func
from config import password
from flask_cors import CORS


#Key Things, added a few line to jupyter lab
#in SQL everyone has to run the following to add primary key
#ALTER TABLE covid ADD PRIMARY KEY (index);
#ALTER TABLE crime ADD PRIMARY KEY (index);

################David added for Mapping
config = {
  'ORIGINS': [
    'http://localhost:5000',  # React
    'http://127.0.0.1:5000',  # React
```

Flask App



UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > static > js > app.js > ...

```javascript

// Get all buttons with class="btn"
var btns = document.getElementsByClassName("btn");

// Loop through the buttons and add the active class to the current/clicked
button
```

UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > templates > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
```

Java Script, HTML, CSS

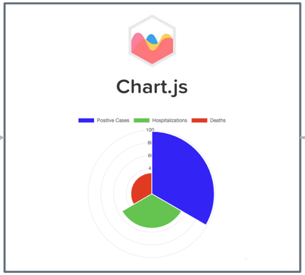


CHART.JS

Final Application

Datasets

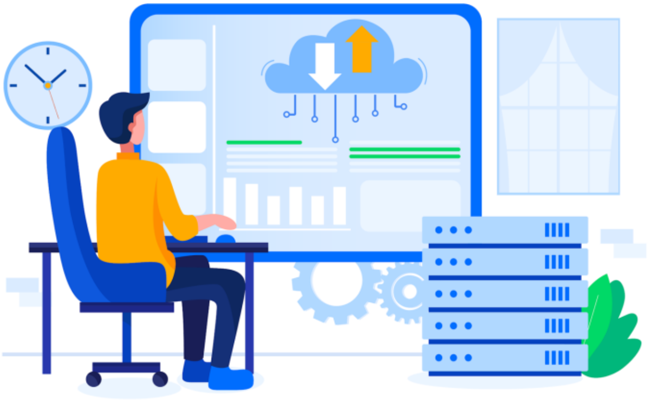
The Sources

Level of Difficulty

🟥🟥🟥□□□□

- All datasets are obtained from NYC open data sources.
- All data were available in CSV and JSON formats.

COVID-19: Data NYC OpenData



NYC Covid-19 Dataset Link:

https://www1.nyc.gov/site/doh/covid/covid-19-data-boroughs.page

NYC Crimes Dataset Link:

https://data.cityofnewyork.us/Social-Services/NYPD/fjn5-bxwg

Why was it difficult?

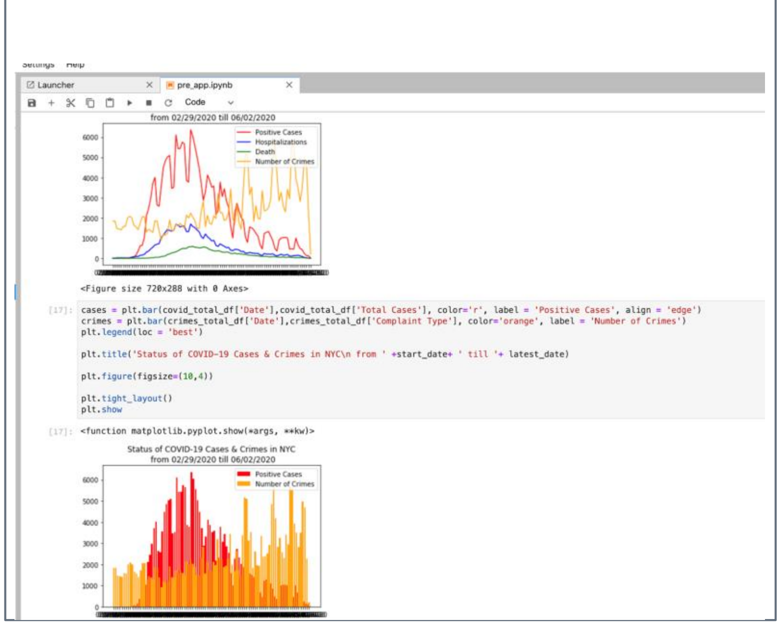
It was challenging to find appropriate datasets that included all elements required such as Latitude and Longitude.

Jupyter

Preliminary Analysis

Level of Difficulty

🟥🟥□□□□□



```
<Figure size 720x288 with 0 Axes>
```

```
[17]: cases = plt.bar(covid_total_df['Date'],covid_total_df['Total Cases'], color='r', label = 'Positive Cases', align = 'edge')
      crimes = plt.bar(crimes_total_df['Date'],crimes_total_df['Complaint Type'], color='orange', label = 'Number of Crimes')
      plt.legend(loc = 'best')

      plt.title('Status of COVID-19 Cases & Crimes in NYC\n from ' +start_date+ ' till '+ latest_date)

      plt.figure(figsize=(10,4))

      plt.tight_layout()
      plt.show
```

```
[17]: <function matplotlib.pyplot.show(*args, **kw)>
```

What was done?

- Read CSV files.
- Removed unnecessary data.
- Merged rows, tables and columns.
- Grouped and aggregated.
- Preliminary plotting.

Why?

- Ensure our datasets are informative
- Preliminary clean up to plan ETL
- Initial plotting for early study

Design

Front-end planning

Level of Difficulty

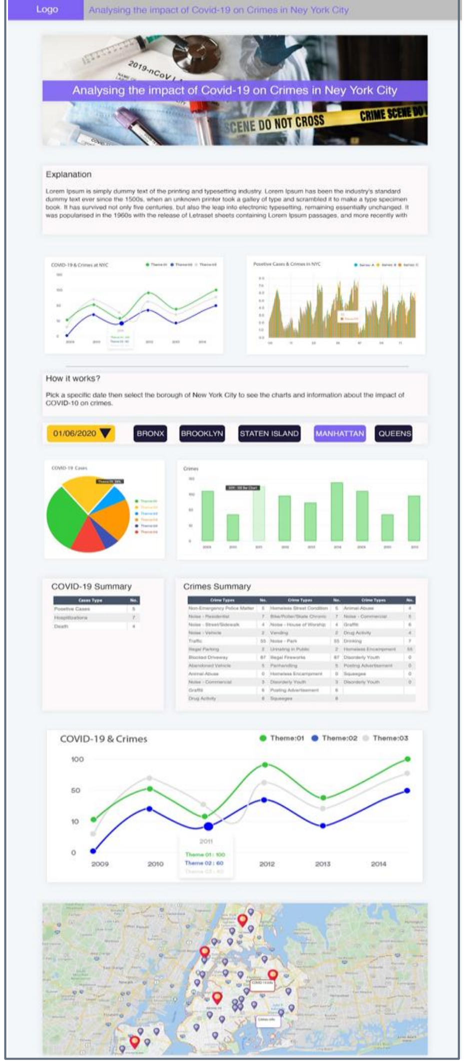
🟥 □ □ □ □ □ □

What was done?

- Based on our datasets and plots, the expected front-end was sketched in Adobe Photoshop.

Why?

- Simplifying the output requirements, providing a layout for the wireframe of the HTML
- To have clear roadmap for expected results



Logo Analysing the impact of Covid-19 on Crimes in Ney York City

Analysing the impact of Covid-19 on Crimes in Ney York City

Explanation

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with

How it works?

Pick a specific date then select the borough of New York City to see the charts and information about the impact of COVID-10 on crimes.

COVID-19 Summary

Cases Type	No.
Positive Cases	5
Hospitilizations	7
Death	4

Crimes Summary

Crime Types	No.	Crime Types	No.	Crime Types	No.
Non-Emergency Police Matter	5	Homeless Street Condition	5	Animal Abuse	4
Noise - Residential	7	Bike/Roller/Skate Chronic	7	Noise - Commercial	5
Noise - Street/Sidewalk	4	Noise - House of Worship	4	Graffiti	6
Noise - Vehicle	2	Vending	2	Drug Activity	4
Traffic	55	Noise - Park	55	Drinking	7
Illegal Parking	2	Urinating in Public	2	Homeless Encampment	55
Blocked Driveway	67	Illegal Fireworks	67	Disorderly Youth	0
Abandoned Vehicle	5	Panhandling	5	Posting Advertisement	0
Animal Abuse	0	Homeless Encampment	0	Squeegee	0
Noise - Commercial	3	Disorderly Youth	3	Disorderly Youth	0
Graffiti	6	Posting Advertisement	6		
Drug Activity	8	Squeegee	8		

Jupyter

ETL

Level of Difficulty

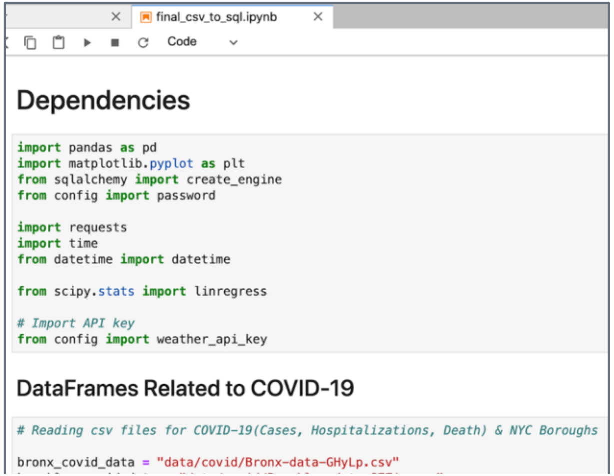
🟥🟥🟥🟥□□□

What was done?

- Jupyter app was created to extract, transform and load data into PostgreSQL.

Why?

- Data was required to be fed into SQL to further serve the front end application.

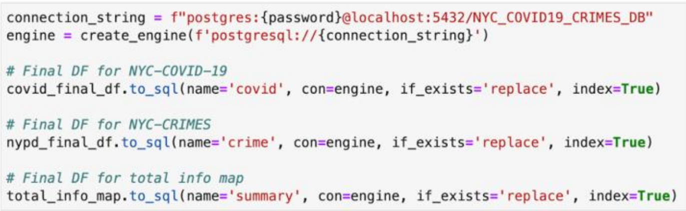


Dependencies

```
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import create_engine
from config import password

import requests
import time
from datetime import datetime

from scipy.stats import linregress

# Import API key
from config import weather_api_key
```

DataFrames Related to COVID-19

```
# Reading csv files for COVID-19(Cases, Hospitalizations, Death) & NYC Boroughs

bronx_covid_data = "data/covid/Bronx-data-GHyLp.csv"
```



```
connection_string = f"postgres:{password}@localhost:5432/NYC_COVID19_CRIMES_DB"
engine = create_engine(f'postgresql://{connection_string}')

# Final DF for NYC-COVID-19
covid_final_df.to_sql(name='covid', con=engine, if_exists='replace', index=True)

# Final DF for NYC-CRIMES
nypd_final_df.to_sql(name='crime', con=engine, if_exists='replace', index=True)

# Final DF for total info map
total_info_map.to_sql(name='summary', con=engine, if_exists='replace', index=True)
```

Why was it difficult?

- DataFrames were changed multiple times to obtain correct data structure and aggregated values.
- Indexing for loading to SQL.

PostgreSQL

ETL

Level of Difficulty

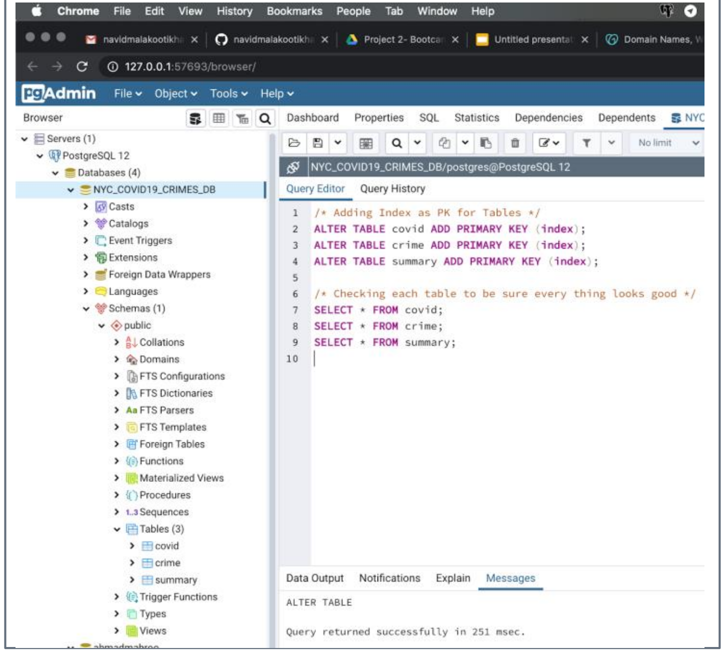
🟥 □ □ □ □ □ □

What was done?

- Database was created.
- Data was loaded into SQL.
- Ran scripts to assign primary keys.

Why?

- To provide data to the Flask App.



```
/* Adding Index as PK for Tables */
ALTER TABLE covid ADD PRIMARY KEY (index);
ALTER TABLE crime ADD PRIMARY KEY (index);
ALTER TABLE summary ADD PRIMARY KEY (index);

/* Checking each table to be sure every thing looks good */
SELECT * FROM covid;
SELECT * FROM crime;
SELECT * FROM summary;
```

```
ALTER TABLE

Query returned successfully in 251 msec.
```

Flask

Flask App

Level of Difficulty

■ ■ ■ ■ ■ ■ ■

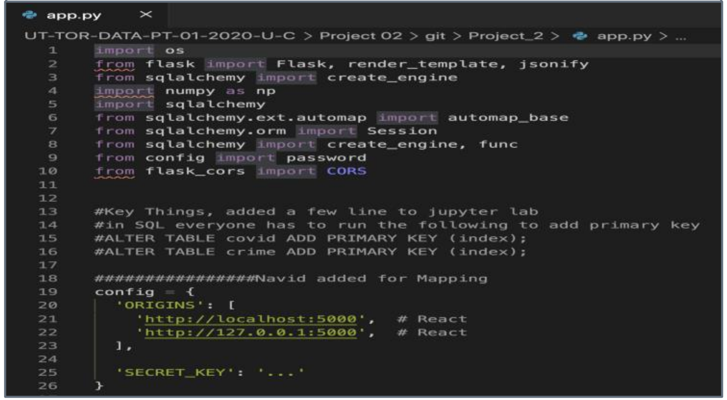
What was done?

- Googling and searching stack overflow to connect to Postgres
- Created API routes
- Provided JSON output data

Why?

Provide a bridge between SQL data and the front-end. Flask was the data server.

Flask
web development,
one drop at a time


app.py
UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > app.py > ...

```python
import os
from flask import Flask, render_template, jsonify
from sqlalchemy import create_engine
import numpy as np
import sqlalchemy
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.orm import Session
from sqlalchemy import create_engine, func
from config import password
from flask_cors import CORS


#Key Things, added a few line to jupyter lab
#in SQL everyone has to run the following to add primary key
#ALTER TABLE covid ADD PRIMARY KEY (index);
#ALTER TABLE crime ADD PRIMARY KEY (index);

################Navid added for Mapping
config = {
  'ORIGINS': [
    'http://localhost:5000',  # React
    'http://127.0.0.1:5000',  # React
  ],

  'SECRET_KEY': '...'
}
```

Why was it difficult?

Multiple API routes were created due to the large size of data and filtering requirements

Front-end

Java Script, HTML, CSS

Level of Difficulty

🟥🟥🟥🟥🟥🟥🟥

What was done?

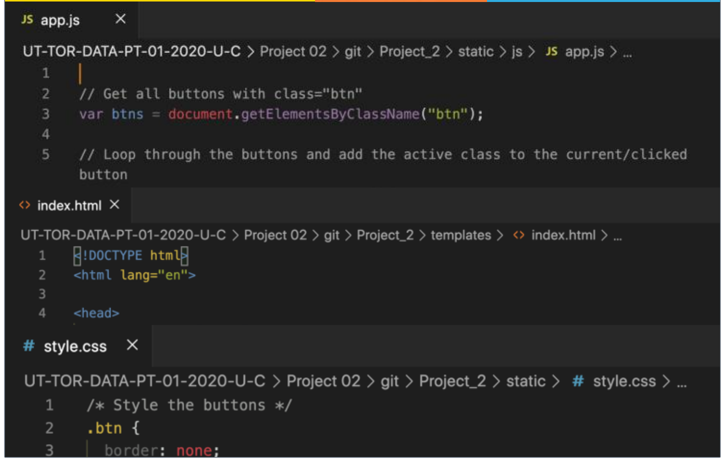
- All JSON data called from Flask API using D3
- Plots, charts, maps and dropdowns created
- Leaflet, plotly, Geojson libraries utilized
- Finalized styling via CSS

Why?

Dan said so!

Why was it difficult?

Beyond words!



app.js

UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > static > js > app.js > ...

```
// Get all buttons with class="btn"
var btns = document.getElementsByClassName("btn");

// Loop through the buttons and add the active class to the current/clicked
button
```

index.html

UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > templates > index.html > ...

```
<!DOCTYPE html>
<html lang="en">

<head>
```

style.css

UT-TOR-DATA-PT-01-2020-U-C > Project 02 > git > Project_2 > static > # style.css > ...

```
/* Style the buttons */
.btn {
  border: none;
```

Extra JS Library

Front end

CHART.JS

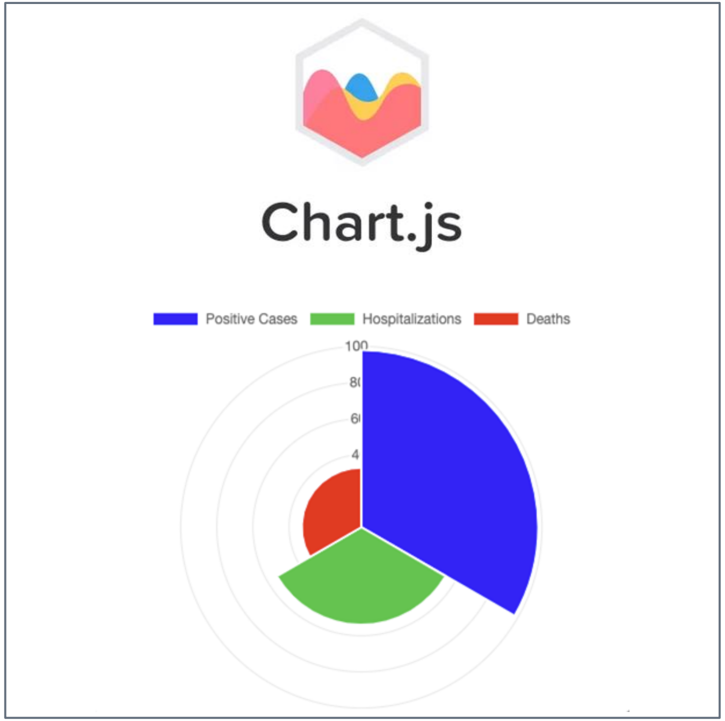
Level of Difficulty

🟥🟥🟥🟥⬜⬜⬜

https://chartjs.org

What was done?

- Polar Area chart created to show proportions of COVID cases with a scale of value for context

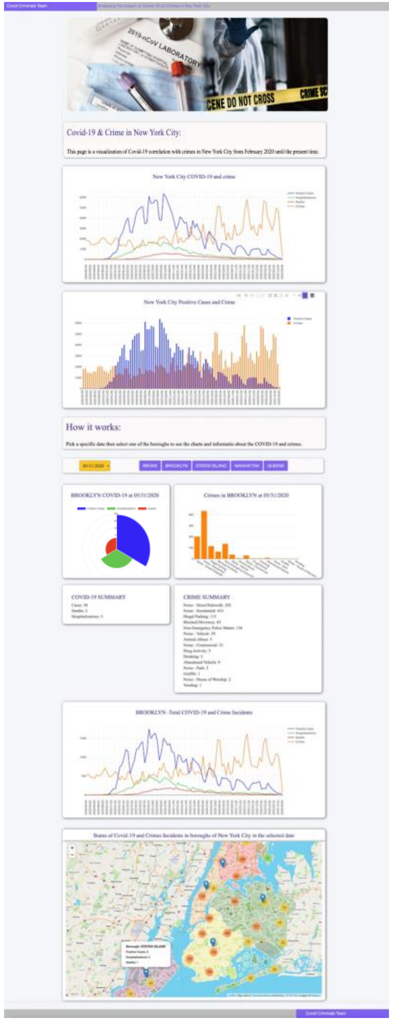


Why?

- Created to explore the functionalities of another JS library

Why was it difficult?

We had trouble deleting the initial data layer and updating the chart with new data



Covid Criminals Team

Analyzing the impact of Covid-19 on Crimes in New York City

Covid-19 & Crime in New York City:

This page is a visualization of Covid-19 correlation with crimes in New York City from February 2020 until the present time.

New York City COVID-19 and crime
New York City Positive Cases and Crime

How it works:

Pick a specific date then select one of the boroughs to see the charts and information about the COVID-19 and crimes.

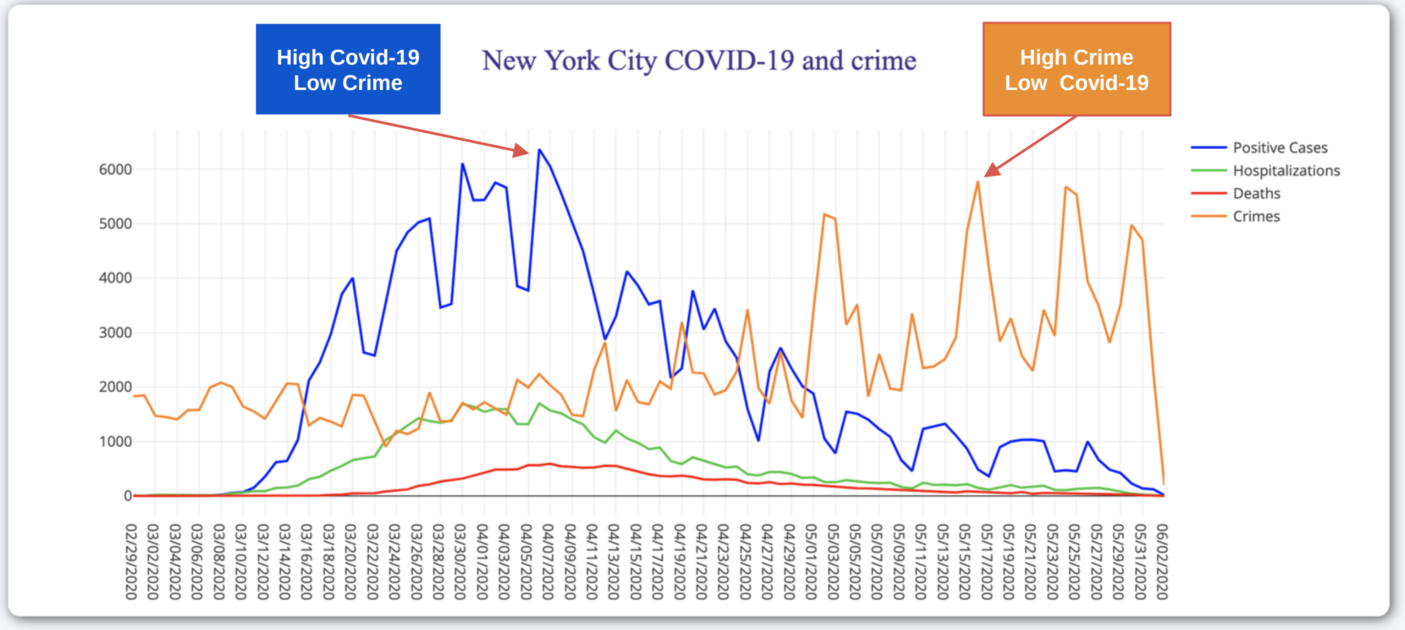
05/31/2020
BRONX
BROOKLYN
STATEN ISLAND
MANHATTAN
QUEENS
BROOKLYN COVID-19 at 05/31/2020
Positive Cases
Hospitalizations
Deaths
Crimes in BROOKLYN at 05/31/2020

COVID-19 SUMMARY
Cases: 38
Deaths: 2
Hospitalizations: 5

CRIME SUMMARY
Noise - Street/Sidewalk: 202
Noise - Residential: 403
Illegal Parking: 119
Blocked Driveway: 69
Non-Emergency Police Matter: 138
Noise - Vehicle: 39
Animal-Abuse: 5
Noise - Commercial: 31
Drug Activity: 3
Drinking: 3
Abandoned Vehicle: 9
Noise - Park: 2
Graffiti: 1
Noise - House of Worship: 2
Vending: 1

BROOKLYN -Total COVID-19 and Crime Incidents
Status of Covid-19 and Crimes Incidents in boroughs of New York City in the selected date
Borough: STATEN ISLAND

Final Application

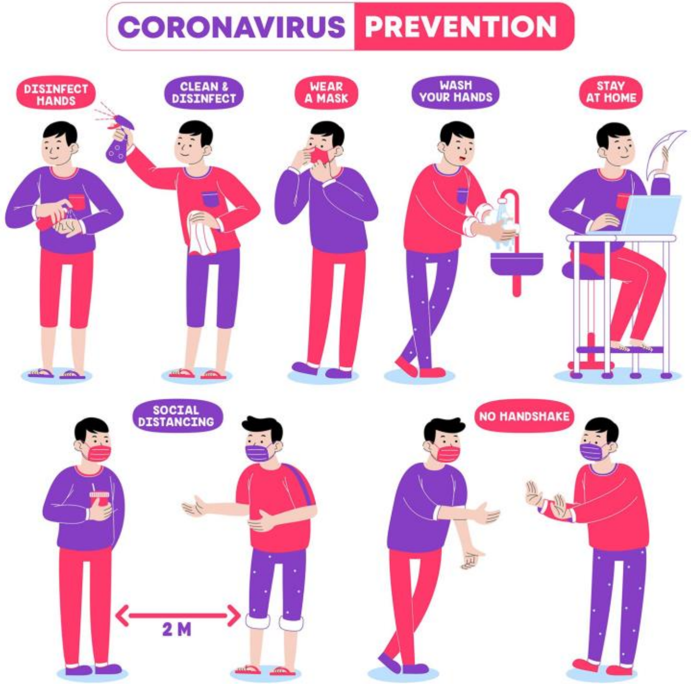


High Covid-19
Low Crime

High Crime
Low Covid-19

Let’s see the project

Discussion Panel



Thank you!