Group 1: Project Proposal

"311 Illegal Parking Complaint Type and Parking Violations Issued For Fiscal Years 2017-2021"

Baruch College (CUNY)
CIS 4400 CMWA: Data Warehousing for Analytics
Professor Richard Holowczak

Member's Email Address:
Danny Huang (danny.huang1@baruchmail.cuny.edu)
Ayra Akhter (ayra.akhter@baruchmail.cuny.edu)
Eric Myagkov (eric.myagkov@baruchmail.cuny.edu)
Mariya Tabachnikova (mariya.tabachnikova@baruchmail.cuny.edu)
Siming Deng (siming.deng@baruchmail.cuny.edu)

Type of 311 Complaint: Illegal Parking

Business Problem:

- Can violation tickets get more expensive after a number of violations?
- Do areas with a higher amount of 311 complaints also exhibit greater amounts of parking and camera violations?
- How many illegal parking complaints are related to “Double Parking”? Is there a larger, city-wide issue with double parking?
- Does a specific zip code have the defective infrastructure (roads, intersections, etc) that could cause drivers to be more prone to the aforementioned violations?
- Could an area with a higher number of reported violations be in closer proximity to a police precinct? (More cops => more tickets)

Synopsis:

Group 1 is observing historical and current data on parking/motor-vehicle violations in NYC to provide accessible information to drivers and local government officials. Group 1 will use the data to better understand the source of these violations, and if there are external factors that skew this data. Faulty infrastructure and elevated police presence are some examples of factors that can inflate the number of violations in a certain zip code. The integration of the 311 data and parking & camera violations can highlight certain issues, which would then be solved by the city.

The first data set Group 1 will use comes from 311 complaints. The data includes the department, complaint type, incident zip code, and status. The second data set will be from Open Data, regarding parking and camera violations. This data set includes the violation type, fine in dollars, vehicle plate number, and issuing agency.

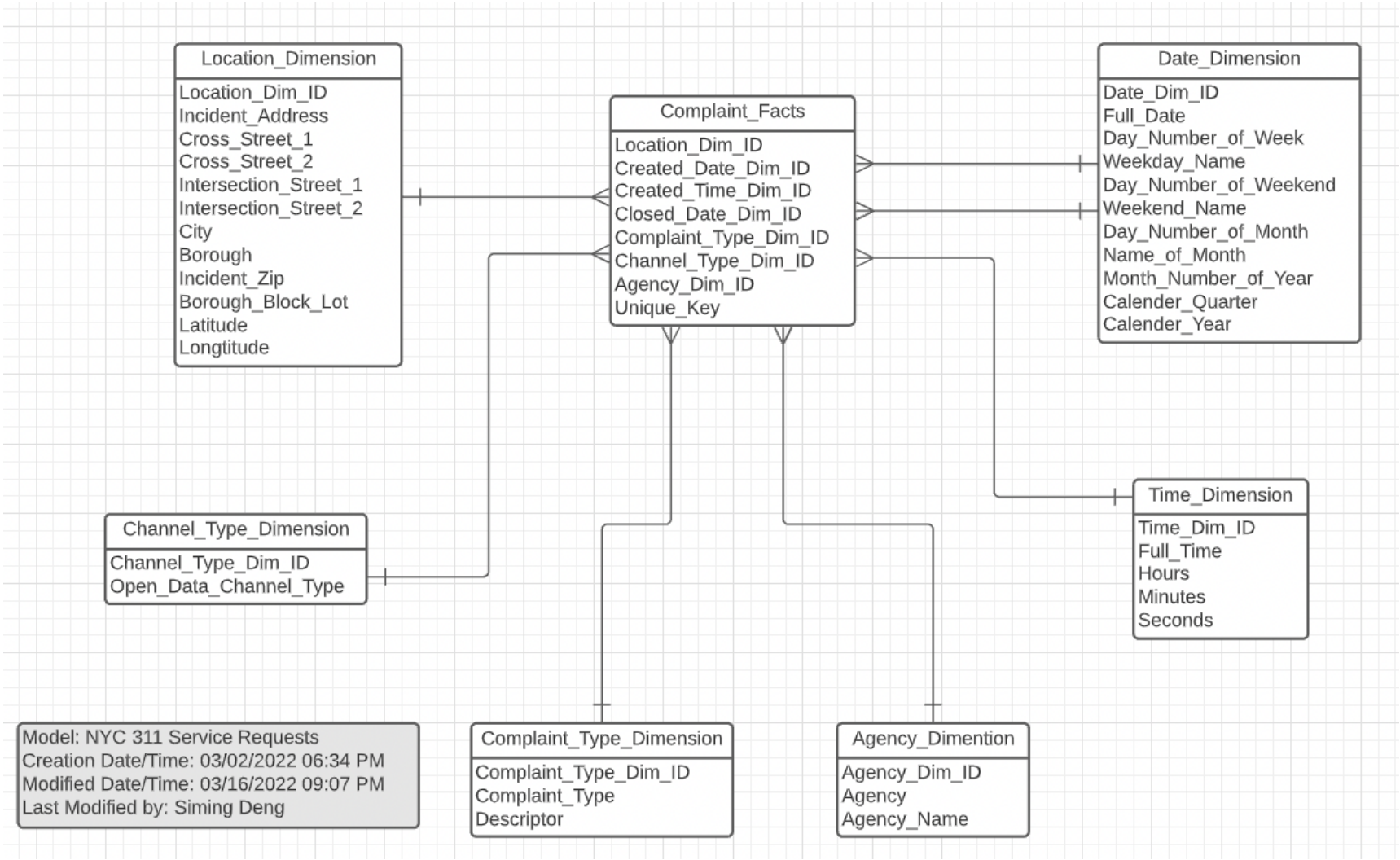
To find a solution, Group 1 will observe tables to track records of violations within specific zip codes and try to deduce if external factors relating to infrastructure skew the reported data.

List of KPIs (including descriptive details)

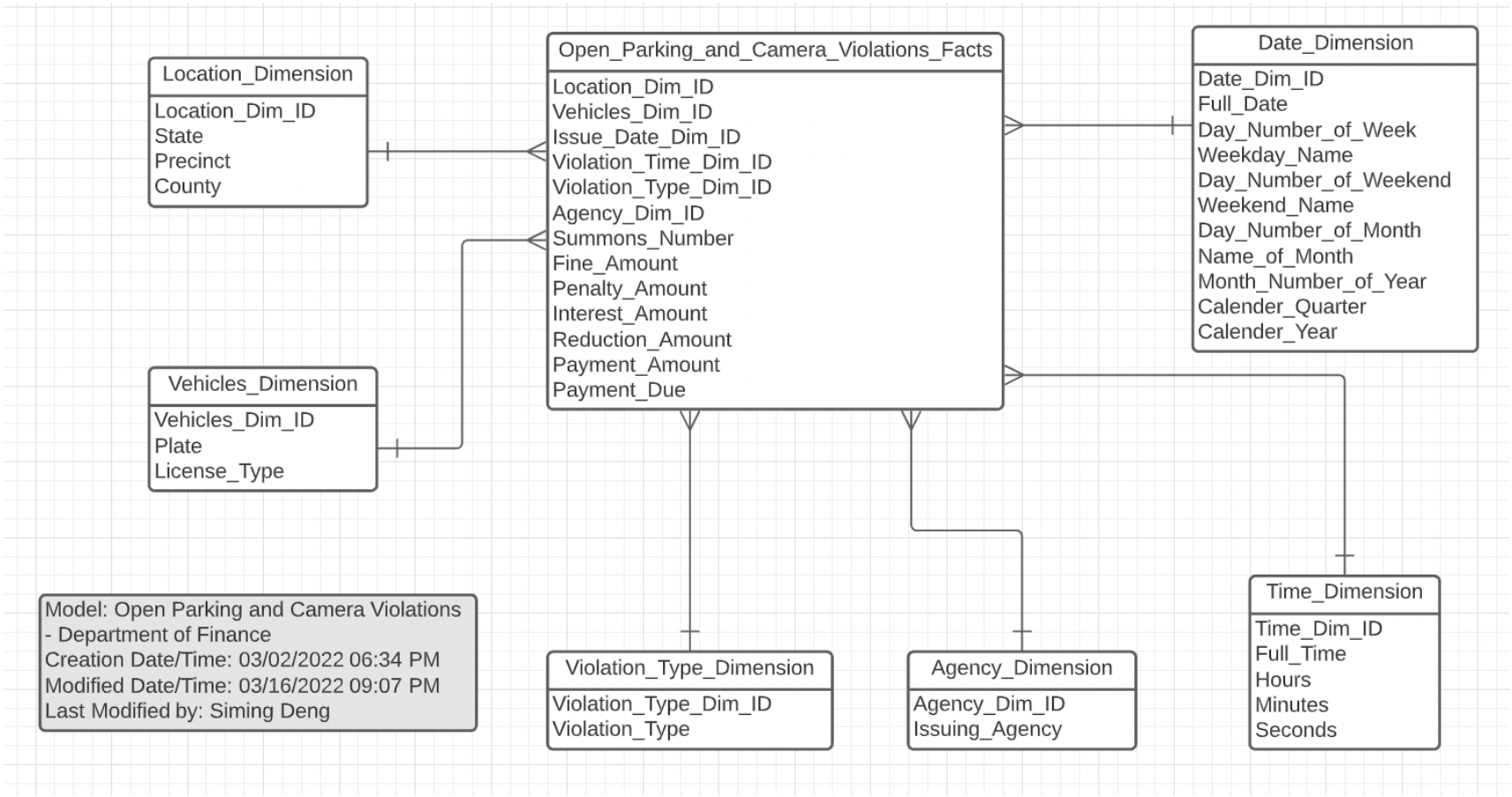
- Percentage increase in the number of issued parking violations in NY
- Total number of illegal parking complaints each year since 2017 in NY
- Number of illegal parking complaints per incident zip code in NY
- Number of illegal parking complaints per Channel Type each year in NY
- The total fine amount because of parking violation each year since 2017
- Total penalty amount because of parking violation each year since 2017
- Total interest amount because of parking violation each year since 2017
- Total reduction amount because of parking violation each year since 2017
- Total payment amount because of parking violation each year since 2017
- Number of violation tickets issued per violation time each day and month

1. Which agency gets the most complaints?
2. Which month has the highest fine and penalty amount in NY?
3. Which date is the busiest for the New York Police Department?

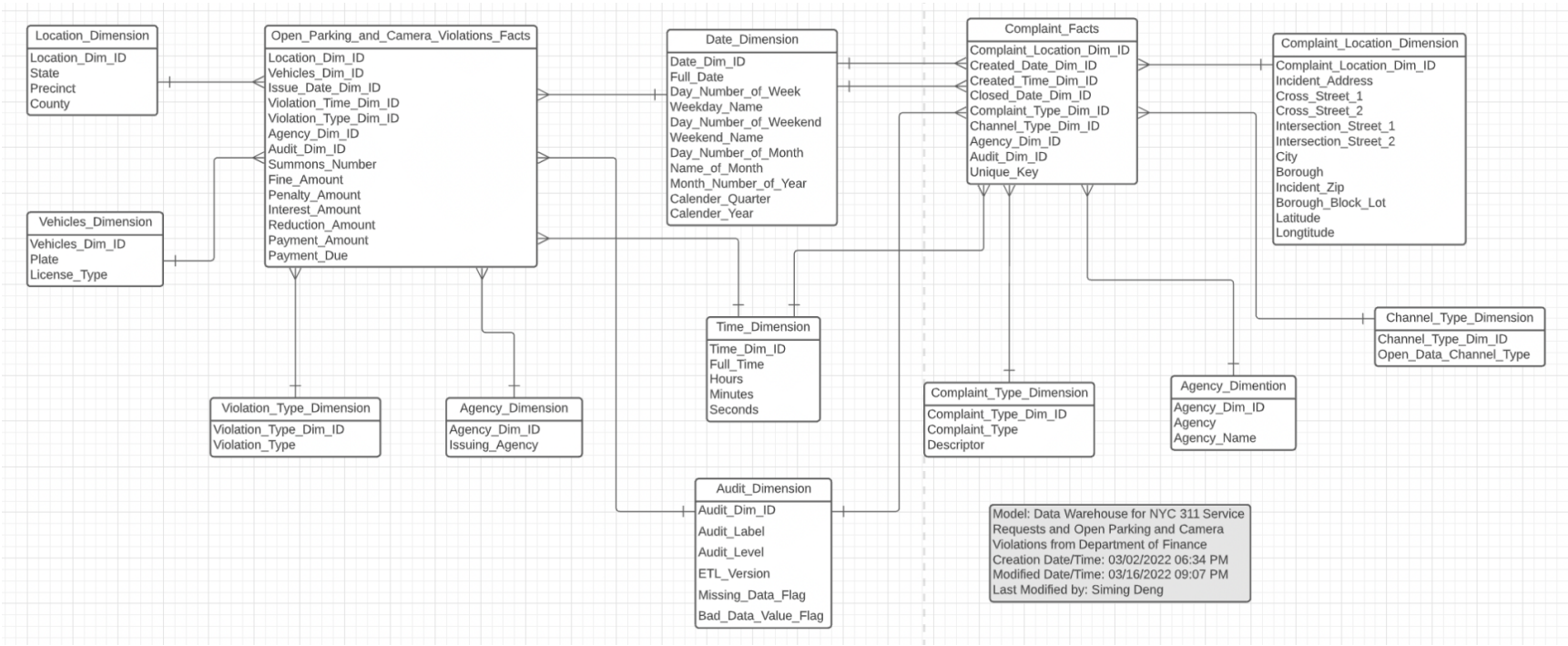
311 Service Dimensional Model



Open Parking and Camera Violations Dimensional Model



Data Warehouse Dimensional Model



Data Profiling

311 Complaint Dataset:

- We have a total of 967634 rows and 19 columns
- We have missing values for the following columns: intersection_street_1 (35.1%), intersection_street_2 (35.1%), bbl (16.4%), cross_street_2 (5.9%), cross_stree_1 (5.8%), incident_address (5.3%), city (5.1%), latitude (1%), longitude (1%), and closed_date (0.1%)
- We have high positive correlations between incident_zip vs bbl, incident_zip vs longitude, and longitude vs latitude
- We have high negative correlations between incident_zip vs latitude, and bbl vs latitude

Target DBMS: Google Big Query

We decided to use Google BigQuery for our target DBMS for multiple reasons. For one, we are all familiar with BigQuery because of the assignment we had. This means that we know that we are all able to contribute. Additionally, BigQuery can handle large datasets. For the 311 complaints dataset, we decided to use complaints starting from 2017 to present, which is over four years worth of data. BigQuery is able to handle such large datasets. It is also able to separate data based on the parameters specified. We want to look at certain complaints and zip codes, and with BigQuery we can compare whatever variables we want.

ETL Tool: dbt

For our ETL tool, we chose dbt. This is because it seems similar to previous programming that was done, since it involves SQL. It will also make it easy to collaborate between group members because dbt can be connected to Git. This way everyone can access the most recent update.

ETL Programming

311_service_requests.sql

```sql
SELECT unique_key,
       created_date,
       closed_date,
       agency,
       agency_name,
       complaint_type,
       descriptor,
       incident_address,
       cross_street_1,
       cross_street_2,
       intersection_street_1,
       intersection_street_2,
       city,
       borough,
       incident_zip,
       bbl,
       latitude,
       longitude,
       open_data_channel_type,
FROM `bigquery-public-data.new_york_311.311_service_requests`
WHERE complaint_type = 'Illegal Parking' AND FORMAT_DATE("%Y", created_date) IN ('2017','2018','2019','2020','2021')
```

Project.yml

```
Version: 2
Models:
      311_service_requests:
            +materialized = table
```

Tests.yml

```
version: 2

models:
  - name: 311_service_requests
    columns:
      - name: unique_key
        tests:
          - unique
          - not_null
      - name: status
        tests:
          - name: status
        tests:
          - accepted_values:
              values:
                - completed
```

dim_complaint_type.sql

```
SELECT
    row_number() OVER () AS Complaint_Type_Dim_ID,
    complaint_type, descriptor
FROM
    (SELECT DISTINCT complaint_type, descriptor
     FROM `fluted-quasar-341922.311_complaint_requests.311_illegal_parking_2017_to_2021`
    )
```

dim_agency_311.sql

```
SELECT
    row_number() OVER () AS Agency_311_Dim_ID,
    agency, agency_name
FROM
    (SELECT DISTINCT agency, agency_name
     FROM `fluted-quasar-341922.311_complaint_requests.311_illegal_parking_2017_to_2021`
    )
```

dim_channel_type.sql

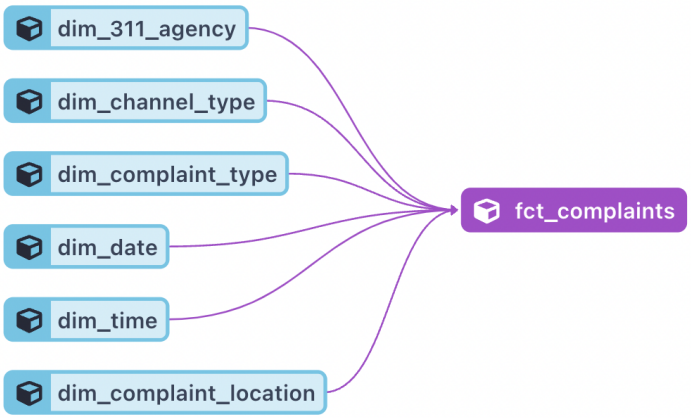
```
SELECT
    row_number() OVER () AS Channel_Type_Dim_ID,
    open_data_channel_type
FROM
    (SELECT DISTINCT open_data_channel_type
     FROM `fluted-quasar-341922.311_complaint_requests.311_illegal_parking_2017_to_2021`
    )
```

dim_complaint_location.sql

```sql
SELECT
    row_number() OVER () AS Compaint_Location_Dim_ID,
    incident_address,
    city,
    borough,
    incident_zip,
    borough_block_lot,
    latitude,
    longitude
FROM
    (SELECT DISTINCT  incident_address,
                    city,
                    borough,
                    incident_zip,
                    bbl AS borough_block_lot,
                    latitude,
                    longitude
     FROM `fluted-quasar-341922.311_complaint_requests.311_illegal_parking_2017_to_2021`
    )
```

```sql
fct_complaints.sql
WITH dim_agency AS
(
   SELECT * FROM {{ ref('dim_311_agency')}}
),
dim_channel_type AS
(
   SELECT * FROM {{ ref('dim_channel_type')}}
),
dim_complaint_type AS
(
  SELECT * FROM {{ ref('dim_complaint_type')}}
),
dim_date AS
(
  SELECT * FROM {{ ref('dim_date')}}
),
dim_time AS
(
  SELECT * FROM {{ ref('dim_time')}}
),
dim_complaint_location AS
(
  SELECT * FROM {{ ref('dim_complaint_location')}}
)

SELECT unique_key,
      agency_311_dim_id,
```

```
      complaint_type_dim_id,
      Channel_Type_Dim_ID,
      Compaint_Location_Dim_ID,
      date_dim_id AS Created_Date_Dim_ID,
      Time_Dim_ID AS Created_Time_Dim_ID
FROM `fluted-quasar-341922.311_complaint_requests.311_illegal_parking` AS sr
        INNER JOIN dim_agency ON dim_agency.agency_name = sr.agency_name
        INNER JOIN dim_complaint_type ON dim_complaint_type.descriptor = sr.descriptor
        INNER JOIN dim_date ON dim_date.full_date = EXTRACT(DATE FROM sr.created_date)
        INNER JOIN dim_time ON dim_time.full_time = EXTRACT(TIME FROM sr.created_date)
        INNER JOIN dim_complaint_location ON dim_complaint_location.borough = sr.borough
         AND dim_complaint_location.incident_zip=sr.incident_zip
         AND dim_complaint_location.latitude = sr.latitude
         AND dim_complaint_location.longitude = sr.longitude
       INNER JOIN dim_channel_type ON dim_channel_type.open_data_channel_type = sr.open_data_channel_type
```



fct_complaints QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Row	unique_key	agency_311_dim_id	complaint_type_dim_id	Channel_Type_Dim_ID	Compaint_Location_Dim_ID	Created_Date_Dim_ID	Create
1	37174916	1	1	2	115341	256	37858
2	37177430	1	1	3	259122	256	32405
3	37174122	1	1	3	199602	256	39899
4	37172619	1	1	1	246257	256	22286
5	37175062	1	1	3	242570	256	9242
6	37177900	1	1	1	157645	256	15926
7	37172648	1	1	1	192658	256	34901
8	37174919	1	1	1	175312	256	26486
9	37175049	1	1	3	189232	256	25357
10	37174211	1	1	3	293402	256	36277
11	37175641	1	1	3	2524	256	29826
12	37174909	1	1	1	25087	256	41495
13	37177409	1	1	2	100961	256	24684
14	37173500	1	1	2	259415	256	29875
15	37167571	1	1	2	271094	256	2641
16	37175804	1	1	3	196406	256	31278
17	37177901	1	1	2	287597	256	10839

Results per page: 50 1 – 50 of 518238

Open Parking and Camera Violation

```
CREATE TABLE `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2017` AS
SELECT Plate,
     State,
     License_Type,
     Summons_Number,
     Violation_time,
     Violation,
     Issue_date,
     Fine_Amount,
     Penalty_Amount,
     Interest_Amount,
     Reduction_Amount,
     Payment_Amount,
     Amount_Due,
     Precinct,
     County,
     Issuing_Agency
FROM `handy-bonbon-142723.nyc_open_parking_and_camera_violations.Open_Parking_and_Camera_Violations_2017`

--Similar Queries are created to create tables to store the data from 2017 to 2021
```

```sql
CREATE TABLE `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL` AS
SELECT *
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2017`
UNION ALL
SELECT *
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2018`
UNION ALL
SELECT *
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2019`
UNION ALL
SELECT *
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2020`
UNION ALL
SELECT *
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.2021`

--Combined all five tables into one table
```

```sql
-- Filtering data to be able to convert string to time
CREATE TABLE `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.All_Filtered` AS
SELECT Plate,
   State,
   License_Type,
   Summons_Number,
   PARSE_TIME('%H:%M%p', Violation_time||'M') AS Violation_time,
   Violation AS Violation_Type,
   Issue_date,
   Fine_Amount,
   Penalty_Amount,
   Interest_Amount,
   Reduction_Amount,
   Payment_Amount,
   Amount_Due,
   Precinct,
   County,
   Issuing_Agency
from `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL`
WHERE  (violation_time LIKE '%A' OR violation_time LIKE '%P' )
  AND SUBSTR(violation_time,0,2) NOT IN('26', '28', '37', '38', '50', '51', '52', '55', '56', '68', '48', '49', '70')
  AND LENGTH(violation_time) = 6
  AND SUBSTR(violation_time,0,1) NOT IN ('.', ':')
  AND NOT violation_time LIKE '%+%'
  AND NOT violation_time LIKE '%.%'
  AND NOT violation_time LIKE '%-%'
  AND NOT violation_time LIKE '%/%'
  AND NOT violation_time LIKE '% %'
```

```
  AND NOT violation_time LIKE '%`%'
  AND CAST(SUBSTR(violation_time,0,2) AS INT64) < 24
```

dbt_project.yml

```
name: 'team_01_cis4400_group_project'
version: '1.0.0'
config-version: 2
profile: 'default'
model-paths: ["models"]
analysis-paths: ["analyses"]
test-paths: ["tests"]
seed-paths: ["seeds"]
macro-paths: ["macros"]
snapshot-paths: ["snapshots"]
target-path: "target"
clean-targets:
  - "target"
  - "dbt_packages"
models:
  team_01_cis4400_group_project:
    marts:
      +materialized: table
```

packages.yml

```
packages:
  - package: dbt-labs/dbt_utils
    version: 0.8.4
```

dim_location.sql

```
SELECT
    row_number() OVER () AS Location_Dim_ID,
    State, Precinct, County
FROM
    (SELECT DISTINCT State, Precinct, County
     FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL`
    )
```

dim_vehicles.sql

```
SELECT
    row_number() OVER () AS Vehicles_Dim_ID,
    Plate, License_Type
FROM
    (SELECT DISTINCT Plate, License_Type
     FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL`
    )
```

dim_violation_type.sql

```
SELECT
    row_number() OVER () AS Violation_Type_Dim_ID,
    Violation AS Violation_Type
FROM
    (SELECT DISTINCT Violation
     FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL`
    )
```

dim_agency.sql

```sql
SELECT
    row_number() OVER () AS Agency_Dim_ID,
    Issuing_Agency
FROM
    (SELECT DISTINCT Issuing_Agency
     FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.ALL`
    )
```

dim_date.sql

```sql
SELECT
    ROW_NUMBER() OVER() AS Date_Dim_ID,
    FORMAT_DATE("%Y%m%d",d) AS Date_Integer,
    d AS Full_Date,
    EXTRACT(YEAR FROM d) AS Year,
    EXTRACT(WEEK FROM d) AS Week_Number_of_Year,
    EXTRACT(DAY FROM d) AS Day_Number_of_Month,
    FORMAT_DATE('%Q', d) AS Quarter_Number_of_Year,
    EXTRACT(MONTH FROM d) AS Month,
    FORMAT_DATE('%B', d) AS Month_Name,
    FORMAT_DATE('%w', d) AS Day_Number_of_Week,
    FORMAT_DATE('%A', d) AS Day_Name,
    (CASE WHEN FORMAT_DATE('%A', d) IN ('Sunday', 'Saturday')
        THEN 0 ELSE 1 END) AS Day_is_Weekday
FROM(
    SELECT *
    FROM UNNEST(GENERATE_DATE_ARRAY('2017-01-01', '2021-12-31', INTERVAL 1 DAY)) AS d)
```

dim_time.sql

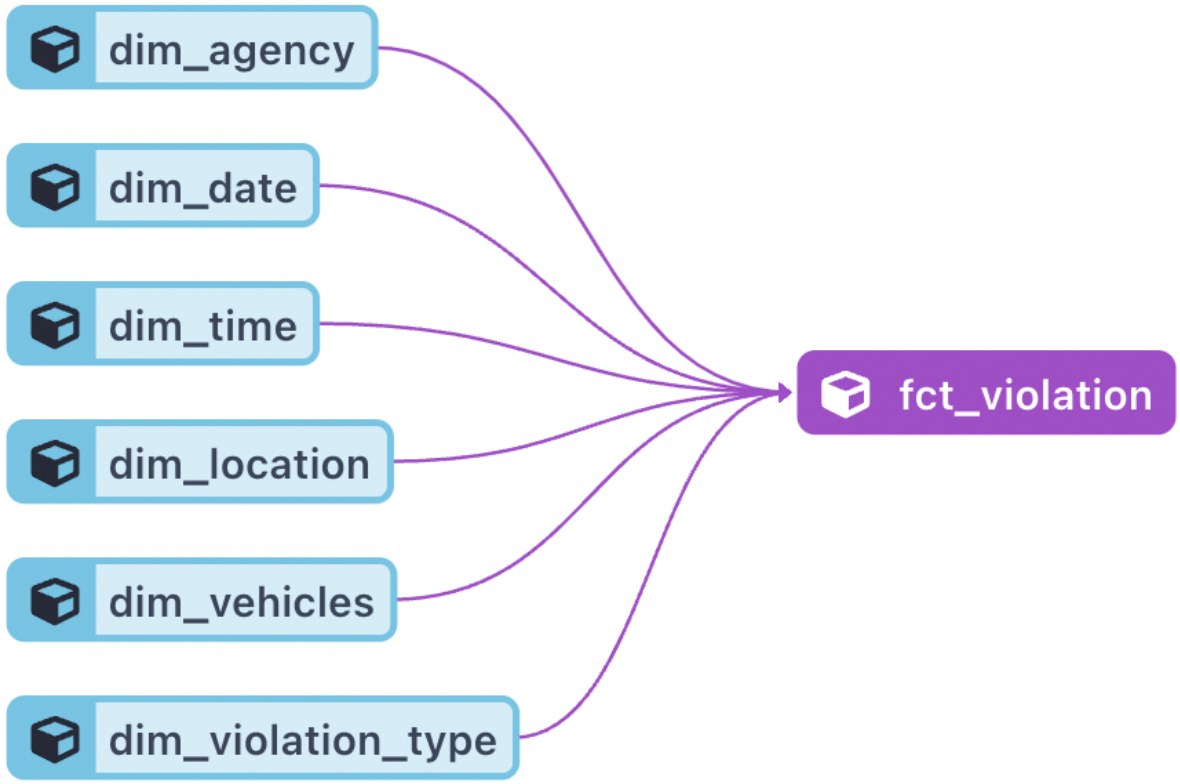
```
SELECT
    ROW_NUMBER() OVER() AS Time_Dim_ID,
    CAST(T as time) AS Full_Time,
    EXTRACT(HOUR FROM T) AS Hour,
    EXTRACT(MINUTE FROM T) AS Minute,
    EXTRACT(SECOND FROM T) AS Second
FROM (SELECT *
      FROM UNNEST(GENERATE_TIMESTAMP_ARRAY('2017-01-01 00:00:00', '2017-01-01 11:59:59', INTERVAL 1 SECOND)) AS T)
```

fct_violations.sql

```
WITH dim_agency AS
(
   SELECT * FROM {{ ref('dim_agency')}}
),
dim_date AS
(
   SELECT * FROM {{ ref('dim_date')}}
),
dim_time AS
(
SELECT * FROM {{ ref('dim_time')}}
),
dim_location AS
(
  SELECT * FROM {{ ref('dim_location')}}
),
dim_vehicles AS
```

```
(
  SELECT * FROM {{ ref('dim_vehicles')}}
),
dim_violation_type AS
(
  SELECT * FROM {{ ref('dim_violation_type')}}
)

SELECT
    Summons_Number,
    Fine_Amount,
    Penalty_Amount,
    Interest_Amount,
    Reduction_Amount,
    Payment_Amount,
    Amount_Due,
    Agency_Dim_ID,
    Date_Dim_ID AS Issue_Date_Dim_ID,
    Time_Dim_ID AS Violation_Time_Dim_ID,
    Location_Dim_ID,
    Vehicles_Dim_ID,
    Violation_Type_Dim_ID
FROM `rising-matrix-350000.Open_Parking_and_Camera_Violations_2017_to_2021.All_Filtered` AS opcv
            INNER JOIN dim_agency ON dim_agency.issuing_agency = opcv.issuing_agency
            INNER JOIN dim_date ON dim_date.full_date = opcv.Issue_date
            INNER JOIN dim_time ON dim_time.Full_Time = opcv.Violation_Time
            INNER JOIN dim_location ON dim_location.state = opcv.state
             AND dim_location.precinct = opcv.precinct
```

```
 AND dim_location.county = opcv.county
INNER JOIN dim_vehicles ON dim_vehicles.plate = opcv.plate
 AND dim_vehicles.license_type = opcv.license_type
INNER JOIN dim_violation_type ON dim_violation_type.Violation_Type = opcv.Violation_Type
```



fct_violation QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Row	Summons_Number	Fine_Amount	Penalty_Amount	Interest_Amount	Reduction_Amount	Payment_Amount	Amount_Due	Agency_Din
1	8533992725	60.0	10.0	0.0	0.0	70.0	0.0	2
2	8497302345	65.0	0.0	0.0	13.0	52.0	0.0	2
3	8793215848	65.0	0.0	0.0	35.0	30.0	0.0	2
4	8558669351	115.0	60.0	12.32	0.22	187.1	0.0	2
5	8658924622	65.0	60.0	0.43	0.0	125.43	0.0	2
6	8559016867	45.0	0.0	0.0	9.0	36.0	0.0	2
7	1483476212	180.0	0.0	0.0	0.0	180.0	0.0	5
8	1447620288	60.0	10.0	0.0	0.0	70.0	0.0	4
9	8379024485	60.0	10.0	0.0	0.0	70.0	0.0	2
10	8605444629	65.0	60.0	12.84	0.41	137.43	0.0	2
11	8481346159	65.0	60.0	1.06	0.0	126.06	0.0	2
12	8657658793	60.0	60.0	1.52	0.21	121.31	0.0	2
13	8484793424	65.0	0.0	0.0	7.0	58.0	0.0	2
14	8673152859	165.0	30.0	0.0	0.0	195.0	0.0	2
15	8532718814	60.0	60.0	0.15	0.0	120.15	0.0	2
16	8621624102	65.0	0.0	0.0	13.0	52.0	0.0	2
17	8606611797	60.0	10.0	0.0	0.0	70.0	0.0	2

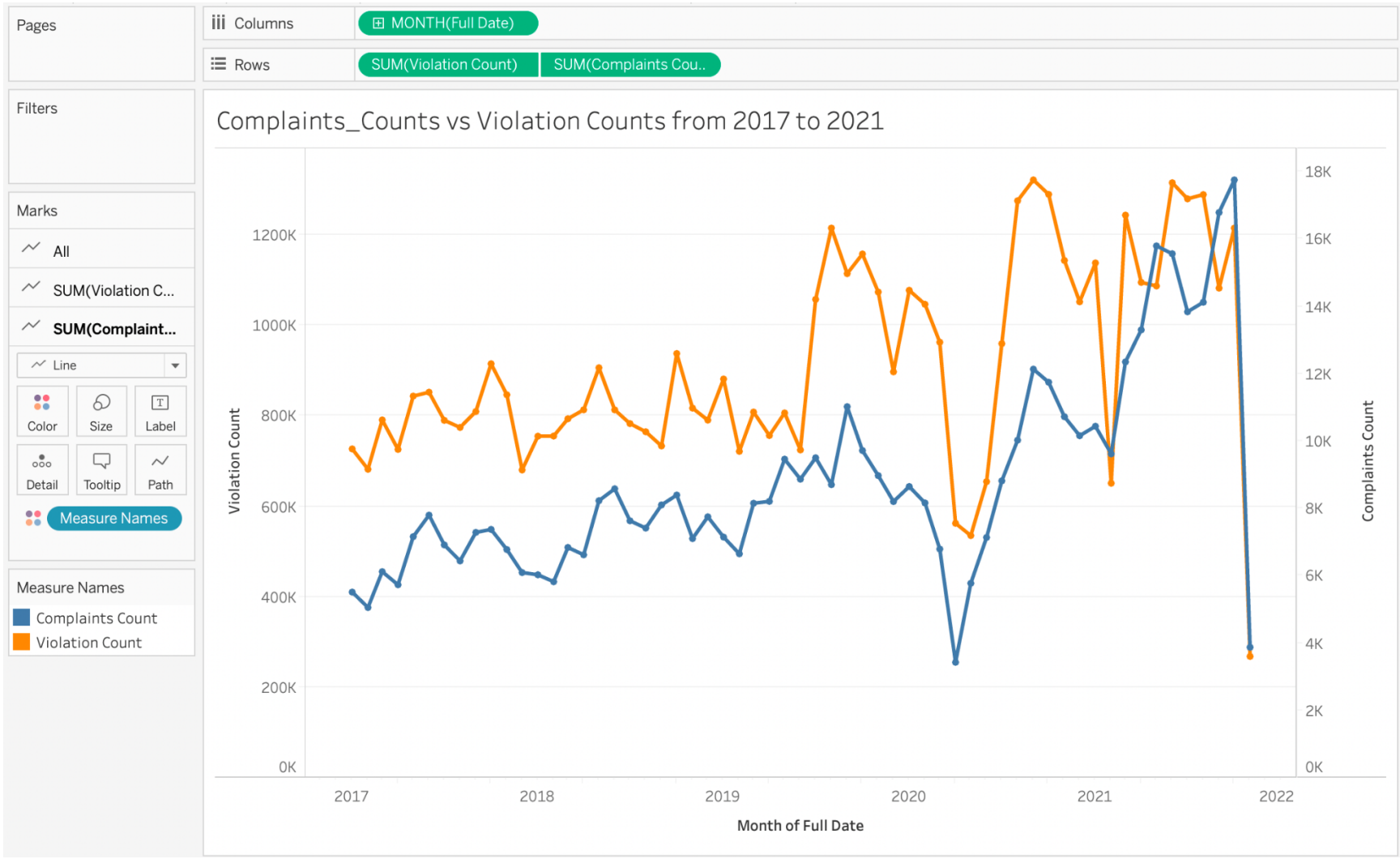
Results per page: 50 1 – 50 of 55816737

Tableau Visualizations

1. **Which agency gets the most complaints?**

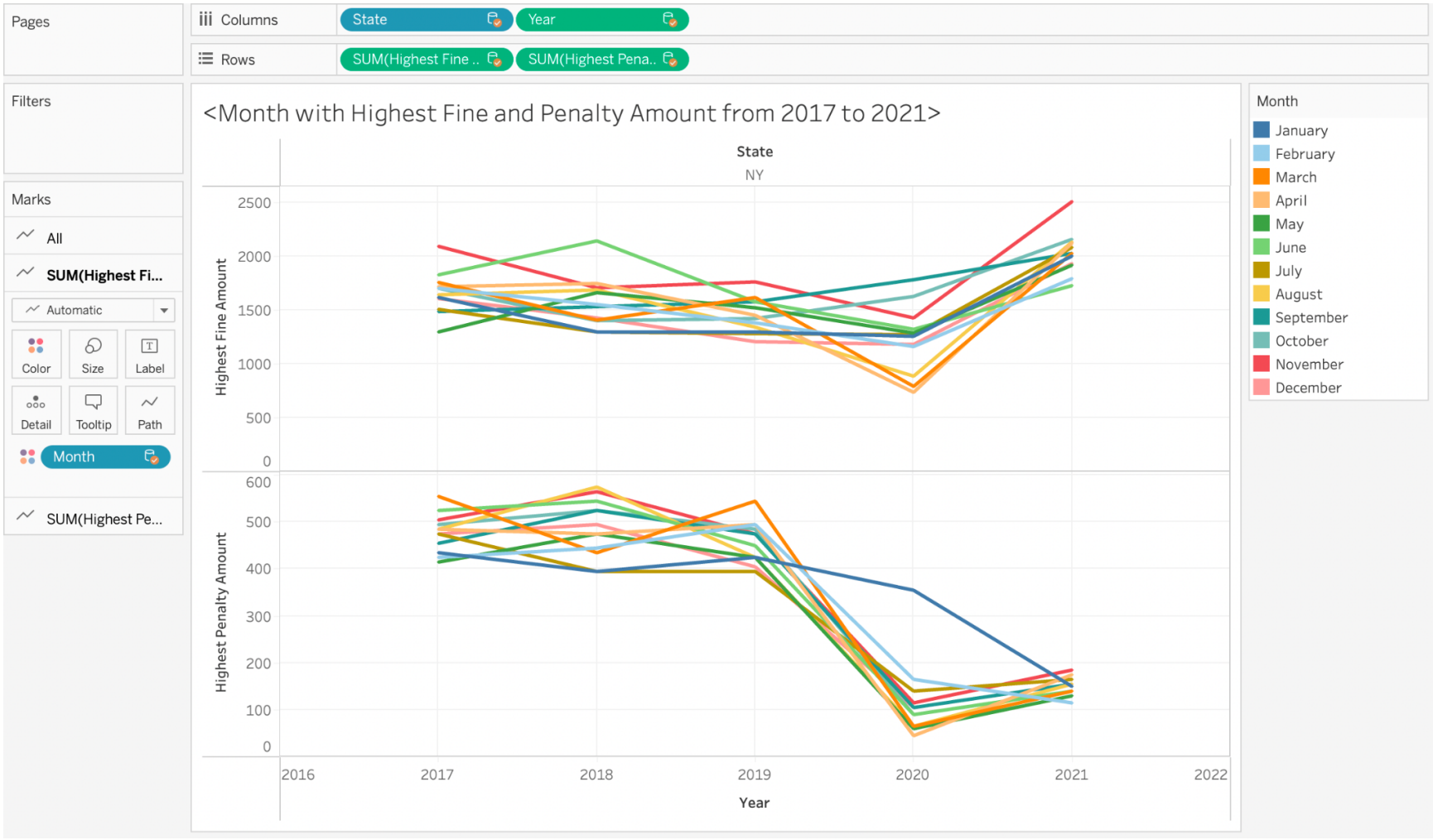
```
WITH complaints AS
(select created_date_dim_id,
     dim_date.Full_Date,
     count(created_date_dim_id) as complaints_count
from `rising-matrix-350000.dbt_sdeng.fct_complaints` as fct_complaints
INNER JOIN `rising-matrix-350000.dbt_sdeng.dim_date` as dim_date ON dim_date.Date_Dim_ID =
fct_complaints.created_date_dim_id
WHERE dim_date.Full_Date BETWEEN '2017-01-01' AND '2021-11-07'
group by dim_date.Full_Date, created_date_dim_id
order by dim_date.Full_Date),

violations AS
(
 select Issue_Date_Dim_ID,
     dim_date.Full_Date,
     count(Issue_Date_Dim_ID) as violation_count
from `rising-matrix-350000.dbt_sdeng.fct_violation` as fct_violation
INNER JOIN `rising-matrix-350000.dbt_sdeng.dim_date` as dim_date ON dim_date.Date_Dim_ID =
fct_violation.Issue_Date_Dim_ID
WHERE dim_date.Full_Date BETWEEN '2017-01-01' AND '2021-11-07'
group by dim_date.Full_Date, Issue_Date_Dim_ID
order by dim_date.Full_Date
)

select Full_Date,
```

```
    complaints_count,
    violation_count
from complaints inner join violations using(Full_Date)
order by Full_Date
```


Pages
Columns
MONTH(Full Date)
Rows
SUM(Violation Count)
SUM(Complaints Cou..
Filters
Marks
All
SUM(Violation C...
SUM(Complaint...
Line
Color
Size
Label
Detail
Tooltip
Path
Measure Names
Measure Names
Complaints Count
Violation Count
Complaints_Counts vs Violation Counts from 2017 to 2021
1200K
1000K
800K
600K
400K
200K
0K
Violation Count
18K
16K
14K
12K
10K
8K
6K
4K
2K
0K
Complaints Count
2017
2018
2019
2020
2021
2022
Month of Full Date

2. Which month has the highest fine and penalty amount in NY?

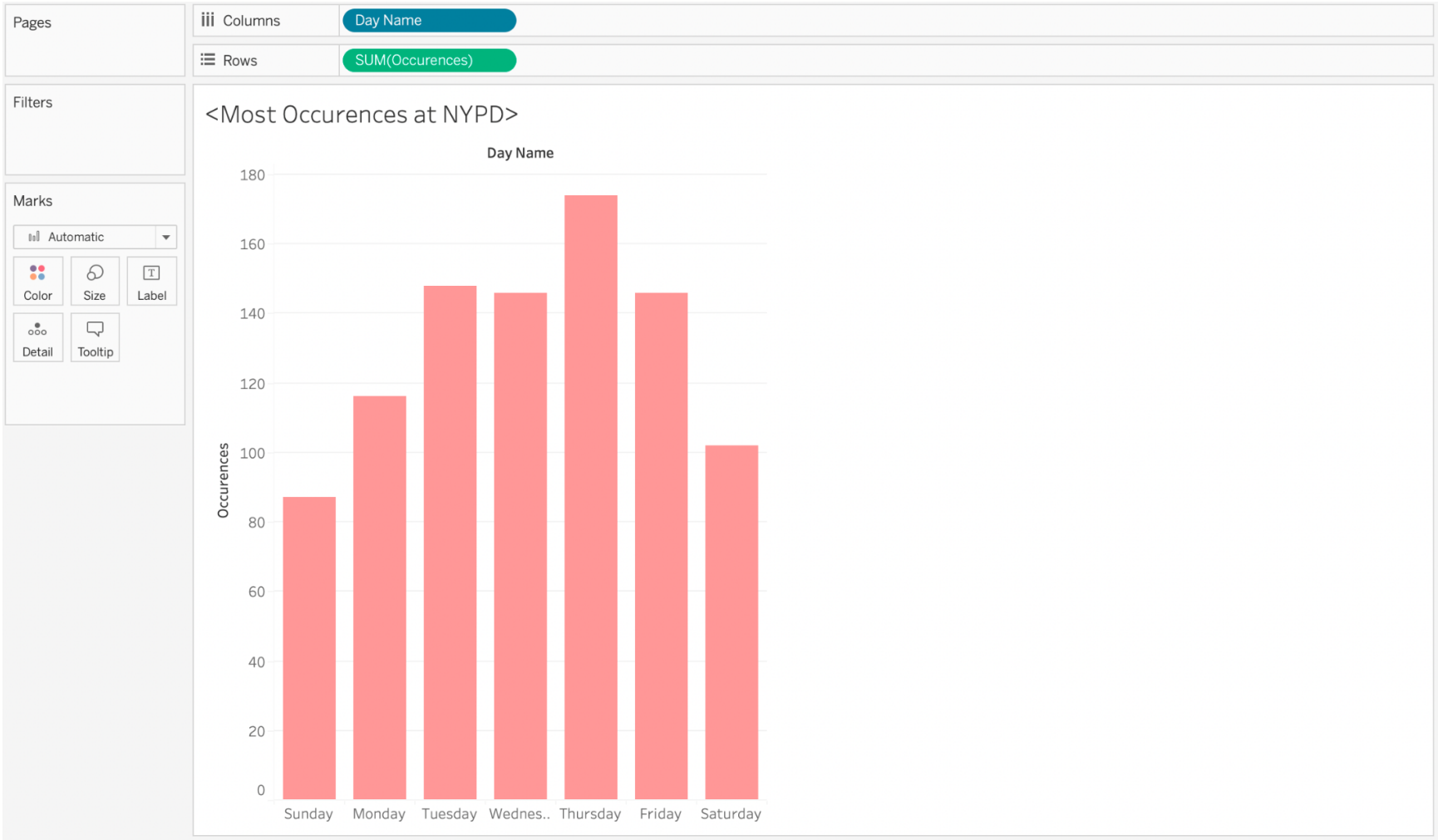
```sql
SELECT
   state,
   Month_Name AS Month,
   year,
   MAX(fine_amount) AS Highest_Fine_Amount,
   MAX(penalty_amount) AS Highest_Penalty_Amount,
FROM `parking-and-camera-violations.dbt_dannihng.violation_filtered`
INNER JOIN `parking-and-camera-violations.dbt_dannihng.311_date`
ON `parking-and-camera-violations.dbt_dannihng.violation_filtered`.Issue_Date =
`parking-and-camera-violations.dbt_dannihng.311_date`.Full_Date
WHERE Issue_Date > '2017-01-01' AND Full_Date > '2017-01-01' AND State = 'NY'
GROUP BY
   fine_amount, penalty_amount, Year, Month_Name, state
ORDER BY
   Highest_Fine_Amount DESC, Highest_Penalty_Amount DESC;
```


Pages
Columns
State
Year
Rows
SUM(Highest Fine ..
SUM(Highest Pena..
Filters
Marks
All
SUM(Highest Fi...
Automatic
Color
Size
Label
Detail
Tooltip
Path
Month
SUM(Highest Pe...
<Month with Highest Fine and Penalty Amount from 2017 to 2021>
State
NY
Highest Fine Amount
2500
2000
1500
1000
500
0
Highest Penalty Amount
600
500
400
300
200
100
0
2016
2017
2018
2019
2020
2021
2022
Year
Month
January
February
March
April
May
June
July
August
September
October
November
December

4. **Which day of the week is the busiest for the New York Police Department?**

```
SELECT

   if(Day_Name = 'Monday',1,

   if(Day_Name = 'Tuesday',2,

   if(Day_Name = 'Wednesday',3,

   if(Day_Name = 'Thursday',4,

   if(Day_Name = 'Friday',5,

   if(Day_Name = 'Saturday',6,7)))))) AS Day,

   day_name,

   COUNT(*) AS occurences

FROM `parking-and-camera-violations.dbt_dannihng.violation_filtered`

INNER JOIN `parking-and-camera-violations.dbt_dannihng.311_date`

ON `parking-and-camera-violations.dbt_dannihng.violation_filtered`.Issue_Date =

`parking-and-camera-violations.dbt_dannihng.311_date`.Full_Date

WHERE State = 'NY' AND issuing_agency = 'POLICE DEPARTMENT'

GROUP BY
```

```
    Day_Name

ORDER BY Day;
```



Reference

1. (DOF), Department of Finance. “Open Parking and Camera Violations: NYC Open Data.” *Open Parking and Camera Violations | NYC Open Data*, 12 Feb. 2022, https://data.cityofnewyork.us/City-Government/Open-Parking-and-Camera-Violations/nc67-uf89.
2. 311, DoITT. “311 Service Requests from 2010 to Present: NYC Open Data.” *311 Service Requests from 2010 to Present | NYC Open Data*, 16 Feb. 2022, https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9.

Appendix

311 Service Request

```
SELECT unique_key,

       created_date,
       closed_date,
       agency,
       agency_name,
       complaint_type,
       descriptor,
       incident_address,
       cross_street_1,
       cross_street_2,
       intersection_street_1,
```

```
        intersection_street_2,
        city,
        borough,
        incident_zip,
        bbl,
        latitude,
        longitude,
        open_data_channel_type,
FROM `bigquery-public-data.new_york_311.311_service_requests`
WHERE complaint_type = 'Illegal Parking' AND FORMAT_DATE("%Y", created_date) IN
('2017','2018','2019','2020','2021')
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	unique_key	created_date	closed_date	agency	agency_name	complaint_type	descriptor	incident
1	50790209	2021-06-06 18:31:48 UTC	2021-06-06 21:16:42 UTC	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	15-26
2	50798809	2021-06-06 17:41:32 UTC	2021-06-06 18:37:19 UTC	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	87-44
3	50790852	2021-06-06 12:26:27 UTC	2021-06-06 17:08:49 UTC	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	149-3(
4	50790861	2021-06-06 19:28:01 UTC	2021-06-06 20:13:35 UTC	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	104-3(
5	50791546	2021-06-06 20:16:32 UTC	2021-06-06 20:36:08 UTC	NYPD	New York City Police Department	Illegal Parking	Blocked Hydrant	89-40

Results per page: 50 1 – 50 of 967634

In [1]:
```
pip install pandas-profiling
```

```
Requirement already satisfied: pandas-profiling in /opt/anaconda3/lib/python3.8/site-packages (3.1.0)
Requirement already satisfied: jinja2>=2.11.1 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling)
(2.11.2)
Requirement already satisfied: visions[type_image_path]==0.7.4 in /opt/anaconda3/lib/python3.8/site-packages (from pa
ndas-profiling) (0.7.4)
Requirement already satisfied: tqdm>=4.48.2 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling) (4.
50.2)
Requirement already satisfied: missingno>=0.4.2 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling)
(0.5.1)
Requirement already satisfied: joblib~=1.0.1 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling)
(1.0.1)
Requirement already satisfied: markupsafe~=2.0.1 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profilin
g) (2.0.1)
Requirement already satisfied: scipy>=1.4.1 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling) (1.
5.2)
Requirement already satisfied: multimethod>=1.4 in /opt/anaconda3/lib/python3.8/site-packages (from pandas-profiling)
(1.8)
Requirement already satisfied: tangled-up-in-unicode==0.1.0 in /opt/anaconda3/lib/python3.8/site-packages (from panda
s-profiling) (0.1.0)
```

In [2]:
```
import pandas_profiling
import pandas as pd
```

In [3]:
```
df = pd.read_csv('/Users/simingdeng/Desktop/CIS 4400/311_illegal_parking_2017_to_2021_without_table_name.csv', low_memo
```

In [4]:
```
data_report = pandas_profiling.ProfileReport(df)
```

In [5]:
```
data_report.to_file('311_illegal_parking_2017_to_2021_data_report.html')
```

```
Summarize dataset: 100% 59/59 [01:17<00:00, 1.31s/it, Completed]

Generate report structure: 100% 1/1 [00:02<00:00, 2.94s/it]

Render HTML: 100% 1/1 [00:00<00:00, 1.03it/s]

Export report to file: 100% 1/1 [00:00<00:00, 46.63it/s]
```

Overview

Overview | Alerts 42 | Reproduction

Dataset statistics

Number of variables	19
Number of observations	967634
Missing cells	1073105
Missing cells (%)	5.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	140.3 MiB
Average record size in memory	152.0 B

Variable types

Numeric	5
Categorical	14

Variables

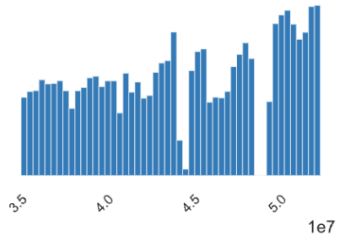
unique_key

Real number ($\mathbb{R}_{\geq 0}$)

`UNIQUE`

Distinct	967634
Distinct (%)	100.0%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	44511226.44

Minimum	35137568
Maximum	52454392
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	7.4 MiB



Toggle details

created_date

Categorical

`HIGH CARDINALITY`
`UNIFORM`

Distinct	963455
Distinct (%)	99.6%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB

2021-05-04 22:27:59 UTC	4
2021-04-15 21:59:15 UTC	3
2021-07-27 12:30:38 UTC	3
2018-06-24 23:28:02 UTC	3
2020-09-27 07:46:07 UTC	3
Other values (963450)	967618

Toggle details

closed_date

Categorical

HIGH CARDINALITY
UNIFORM

Distinct	903138
Distinct (%)	93.4%
Missing	486
Missing (%)	0.1%
Memory size	7.4 MiB

2019-07-12 15:08:07 UTC	23
2019-07-12 12:08:24 UTC	17
2019-07-12 11:25:46 UTC	16
2019-07-12 15:08:02 UTC	14
2019-07-12 11:26:21 UTC	14
Other values (903133)	967064

Toggle details

~~agency~~

~~Categorical~~

CONSTANT
HIGH CORRELATION
REJECTED

Distinct	1
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB

NYPD	967634

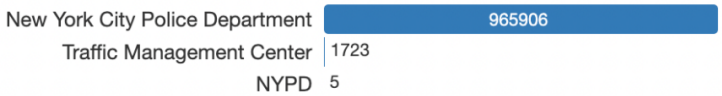
Toggle details

agency_name

Categorical

HIGH CORRELATION

Distinct	3
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB



Toggle details

~~complaint_type~~

~~Categorical~~

CONSTANT
HIGH CORRELATION
REJECTED

Distinct	1
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB

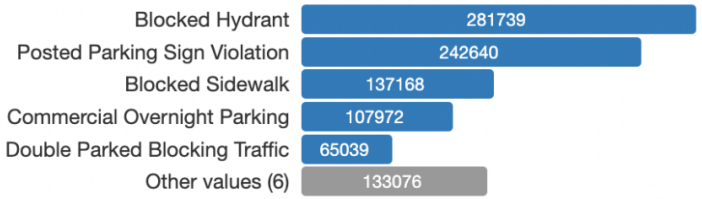
Toggle details

descriptor

Categorical

HIGH CORRELATION

Distinct	11
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB



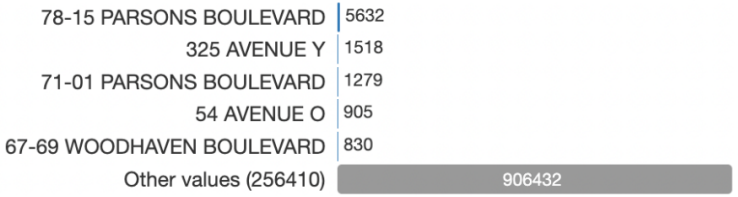
Toggle details

incident_address

Categorical

HIGH CARDINALITY
MISSING

Distinct	256415
Distinct (%)	28.0%
Missing	51038
Missing (%)	5.3%
Memory size	7.4 MiB



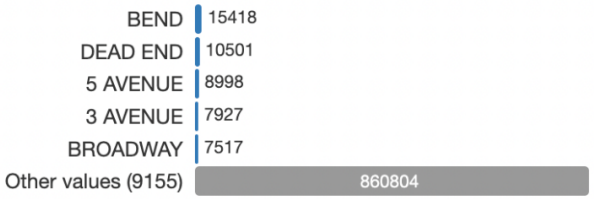
Toggle details

cross_street_1

Categorical

HIGH CARDINALITY
MISSING

Distinct	9160
Distinct (%)	1.0%
Missing	56469
Missing (%)	5.8%
Memory size	7.4 MiB



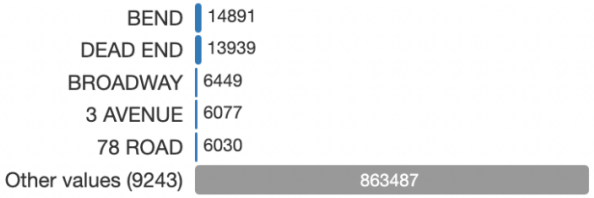
Toggle details

cross_street_2

Categorical

HIGH CARDINALITY
MISSING

Distinct	9248
Distinct (%)	1.0%
Missing	56761
Missing (%)	5.9%
Memory size	7.4 MiB



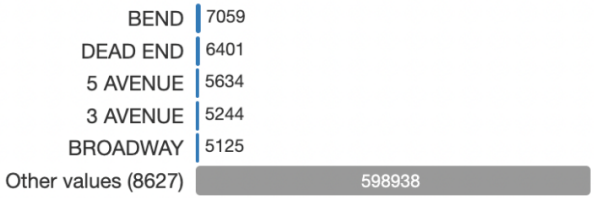
Toggle details

intersection_street_1

Categorical

HIGH CARDINALITY
MISSING

Distinct	8632
Distinct (%)	1.4%
Missing	339233
Missing (%)	35.1%
Memory size	7.4 MiB



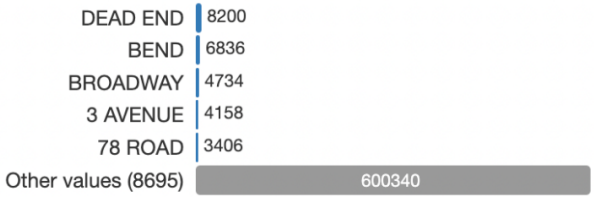
Toggle details

intersection_street_2

Categorical

HIGH CARDINALITY
MISSING

Distinct	8700
Distinct (%)	1.4%
Missing	339960
Missing (%)	35.1%
Memory size	7.4 MiB



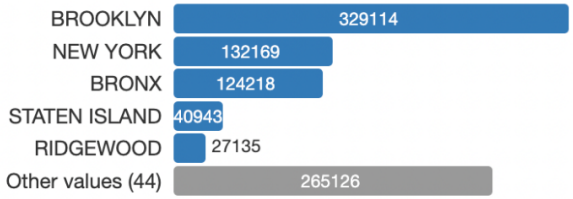
Toggle details

city

Categorical

HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	49
Distinct (%)	< 0.1%
Missing	48929
Missing (%)	5.1%
Memory size	7.4 MiB



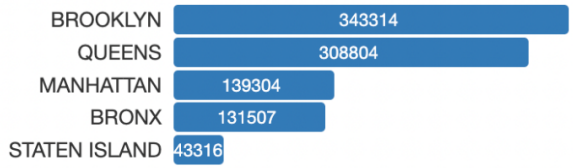
Toggle details

borough

Categorical

HIGH CORRELATION
HIGH CORRELATION

Distinct	6
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB



Toggle details

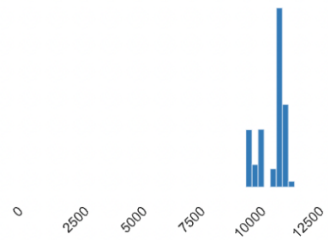
incident_zip

Real number ($\mathbb{R}_{\geq 0}$)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION

Distinct	230
Distinct (%)	< 0.1%
Missing	1353
Missing (%)	0.1%
Infinite	0
Infinite (%)	0.0%
Mean	10949.61138

Minimum	83
Maximum	13207
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	7.4 MiB



Toggle details

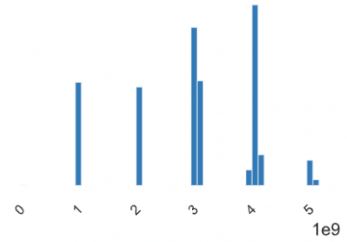
bbl

Real number ($\mathbb{R}_{\geq 0}$)

HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	185871
Distinct (%)	23.0%
Missing	158872
Missing (%)	16.4%
Infinite	0
Infinite (%)	0.0%
Mean	3018472333

Minimum	0
Maximum	5270000501
Zeros	132
Zeros (%)	< 0.1%
Negative	0
Negative (%)	0.0%
Memory size	7.4 MiB



Toggle details

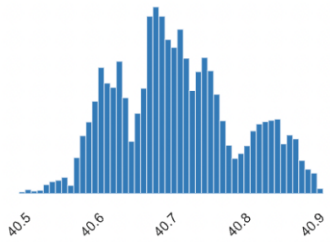
latitude

Real number ($\mathbb{R}_{\geq 0}$)

HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	282547
Distinct (%)	29.5%
Missing	10002
Missing (%)	1.0%
Infinite	0
Infinite (%)	0.0%
Mean	40.71507064

Minimum	40.49912144
Maximum	40.91345653
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	7.4 MiB



Toggle details

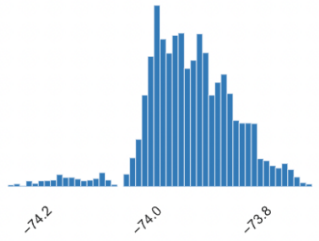
longitude

Real number ($\mathbb{R}$)

HIGH CORRELATION
HIGH CORRELATION
MISSING

Distinct	282543
Distinct (%)	29.5%
Missing	10002
Missing (%)	1.0%
Infinite	0
Infinite (%)	0.0%
Mean	-73.92353761

Minimum	-74.25453162
Maximum	-73.70059685
Zeros	0
Zeros (%)	0.0%
Negative	957632
Negative (%)	99.0%
Memory size	7.4 MiB



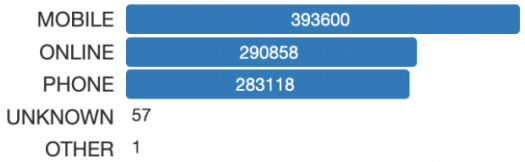
Toggle details

open_data_channel_type

Categorical

HIGH CORRELATION

Distinct	5
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	7.4 MiB



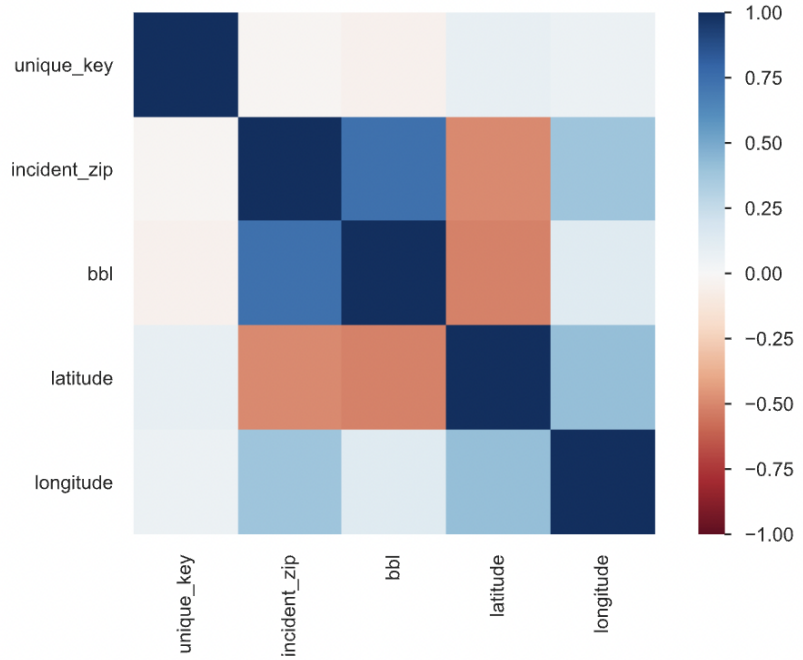
Toggle details

Interactions

Correlations

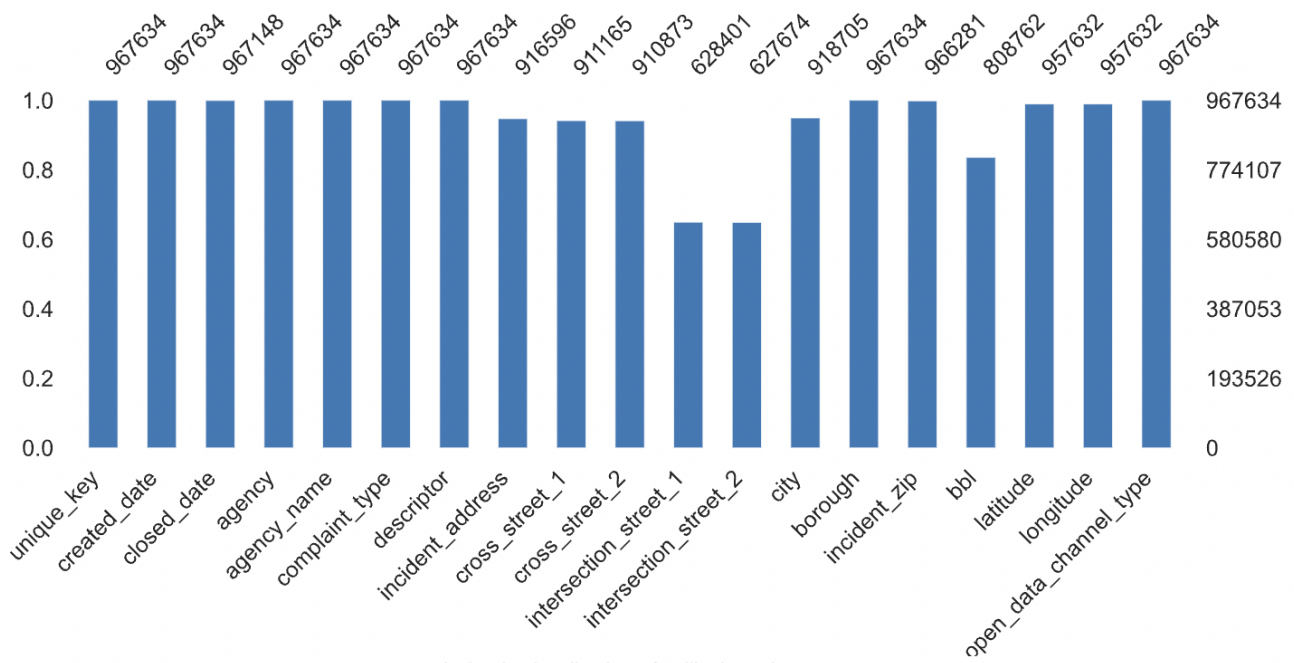
Spearman's ρ | Pearson's r | Kendall's τ | Cramér's V (φc) | Phik (φk)

Toggle correlation descriptions



Missing values

Count | Matrix | Heatmap | Dendrogram



A simple visualization of nullity by column.

Narrative conclusion:

a) the software and database tools the group used to coordinate and manage the project as well as carry out the programming tasks (list of bullet points with software or service and one sentence of what it was used for)

- **Google BigQuery:** We extracted data from datasets and connected to DBT.
- **Jupyter Notebook:** We created a data profile for both datasets.
- **Tableau:** We created a data visualization for both datasets and created a dashboard.
- **DBT Tools:** We created models and fact tables for both datasets to transport to Tableau.

b) the group's experience with the project (which steps were the most difficult? Which were the easiest? what did you learn that you did not imagine you would have? if you had to do it all over again, what would you have done differently?)

- ETL modeling was the most difficult- converting time data from string in the Open Camera and Violations data set
- Visualization was the easiest
- Learned to work with new tools like BigQuery and dbt, learned to work through problems as a group, data profiling to get an idea of what exactly is in the dataset
- Learned the intricacies of ETL pipeline, realized how these pipelines are built and what that would look like in a business setting
- Spent more time on ETL modeling
- We realized that the data in nyc dataset is scarce, and recognized that we should work on snapshot grain instead of transactional grain for time efficiency.

c) if the proposed benefits can be realized by the new system

- By cleaning data, we provide accessible and easy to read information for drivers and local government officials. Through this project, we were able to provide insight about the faulty infrastructure on elevated police presence and the effect it can have on violations.

d) any final comments and conclusions

Overall, we enjoyed and learned a lot through this semester for the group project. Definitely working on the assignments prior to working on project milestones are very helpful in gaining the understanding of the project.