# Incident Response

—

Team 10

Team Member: Xi Sun, Austin Wang

# Datasets provided:

Weather dataset for Davidson County (2010 - 2022):

- Precipitation, snow, wind speed,  clouds, temperature, etc.

Traffic dataset for Davidson County (2017 - 2022):

- Time stamp, speed, congestion, travel time seconds, etc.

Incident dataset:

- Road segment ID, location(latitude, longitude), type of incidents (emdcardnumber), time, response time, etc.

# Analysis Objective:

Temporal Analysis on incidents

Spatial Analysis on incidents

Effect of time and space on response time (effect of time_local on avg response_time)

Join Incident with traffic datasets for better analysis

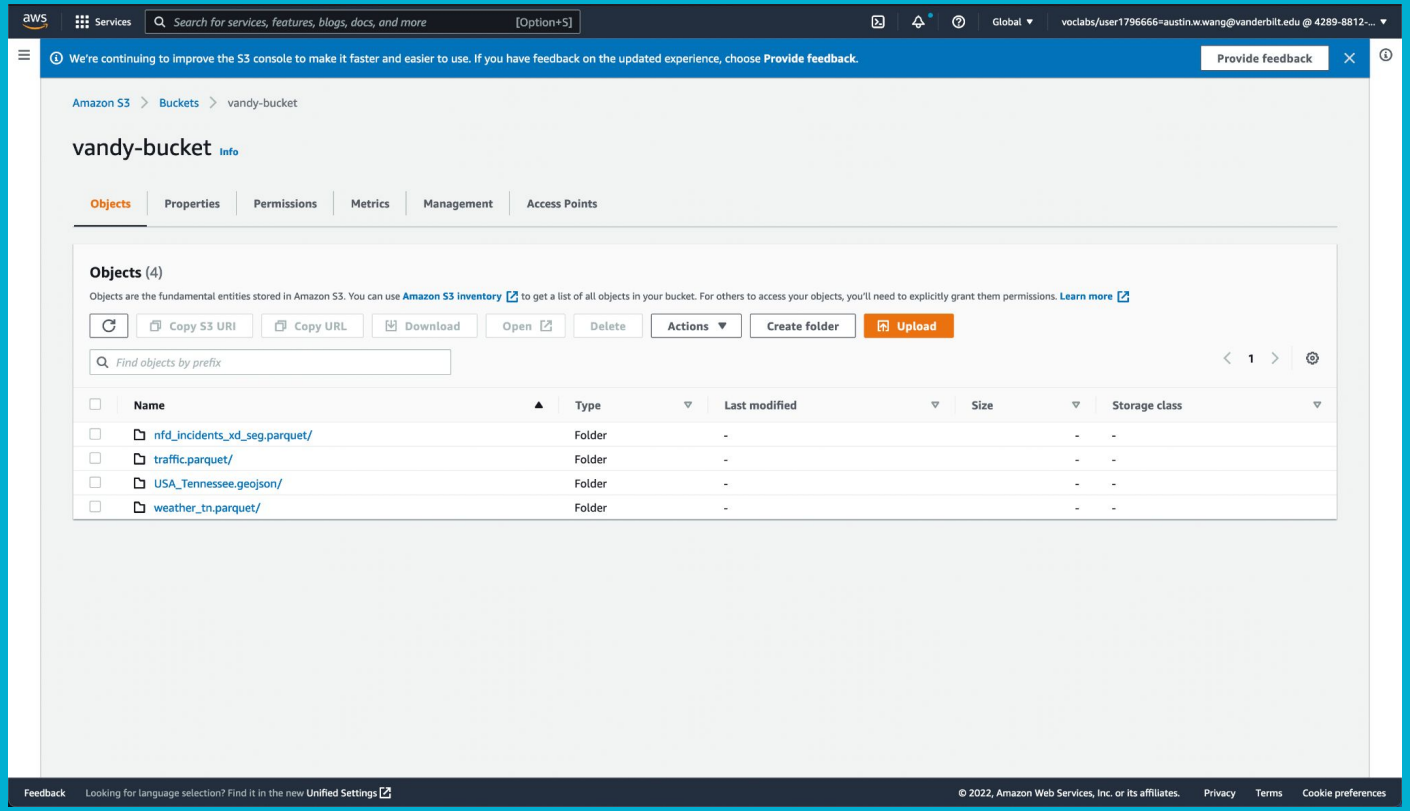
Join Incident with weather datasets for better analysis

Graphical Visualization for the Query

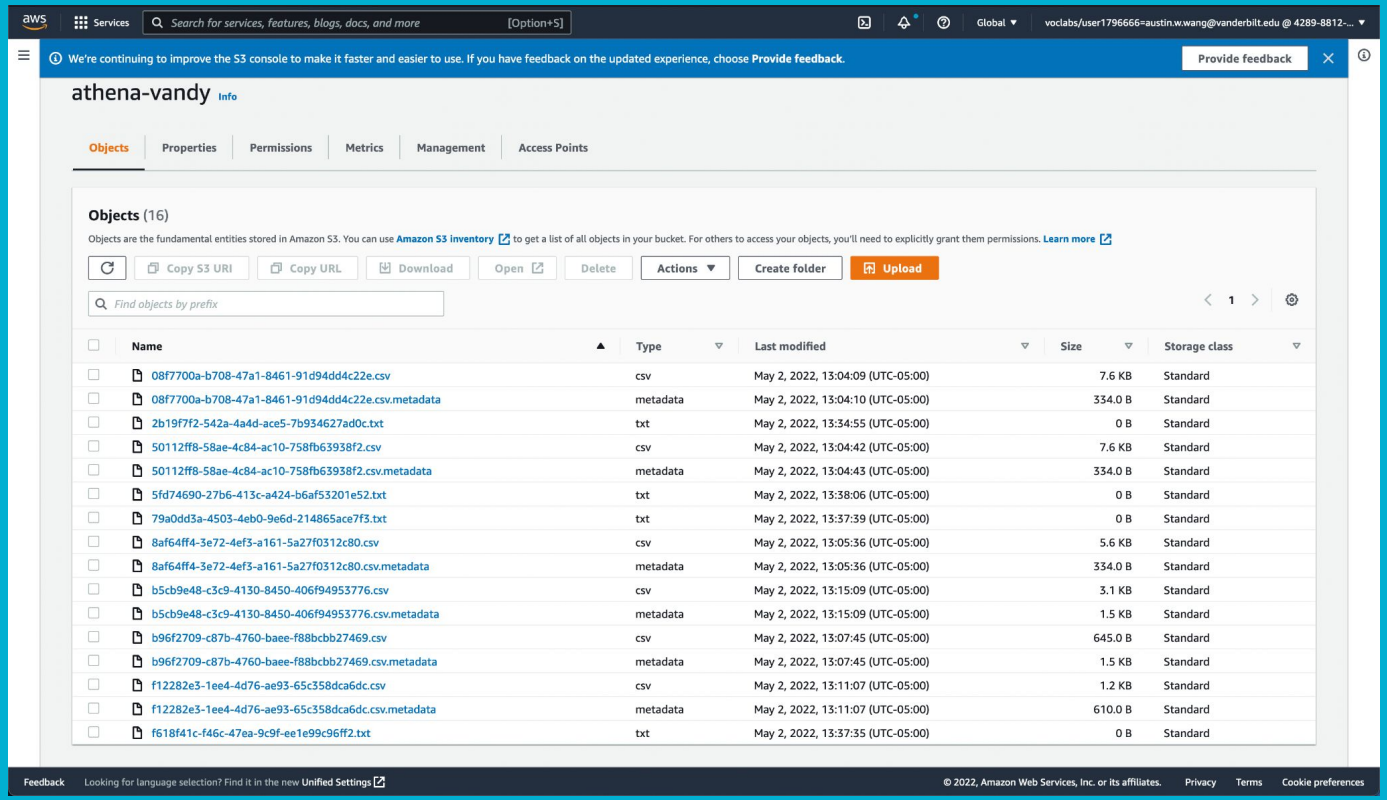
# Approach:

- Provided with over 20 GB of data → Upload to AWS S3
  - Parquet
  - GeoJSON
- Hard to run on local with big data → Establish connection with Athena and run all the queries in cloud
- Data visualization → use Plotly

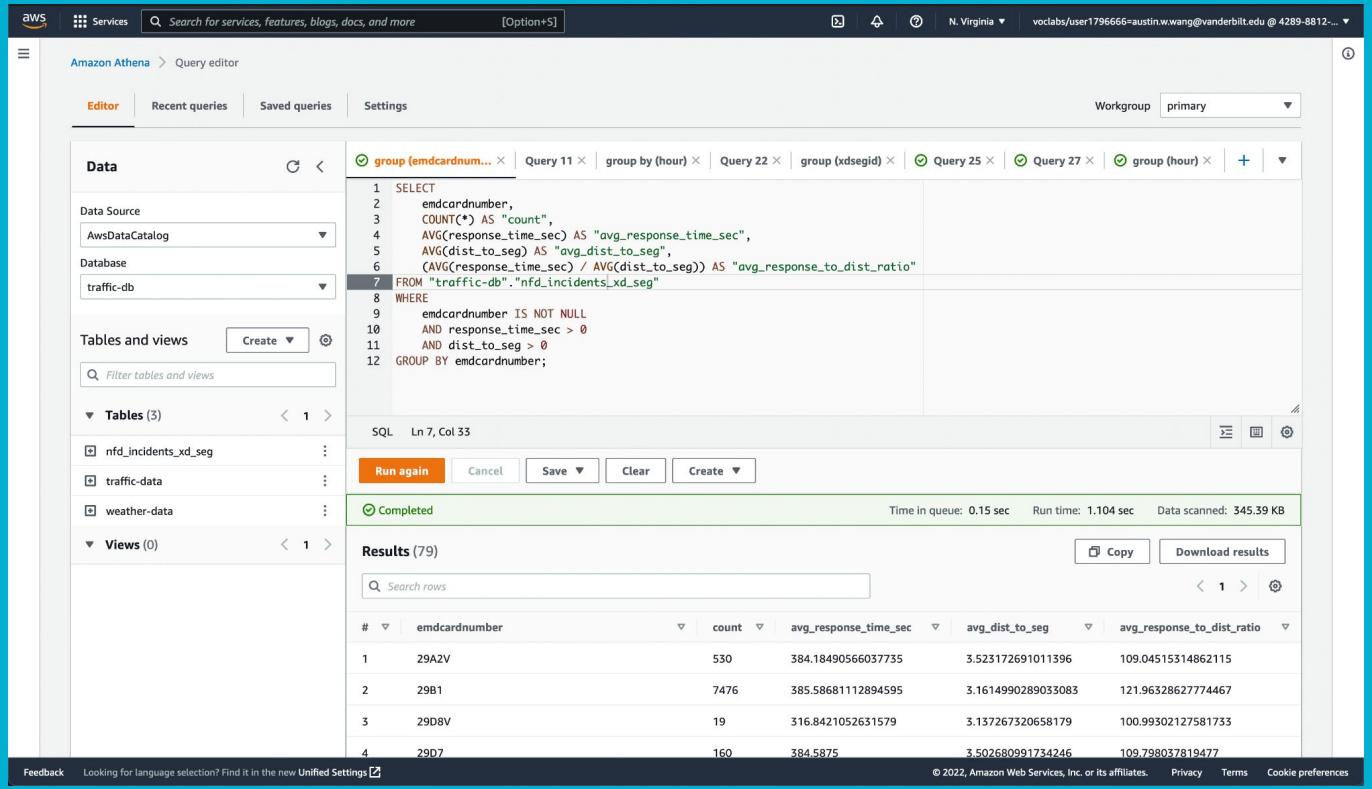
# S3 (Dataset Bucket)



# S3 (Results Bucket)

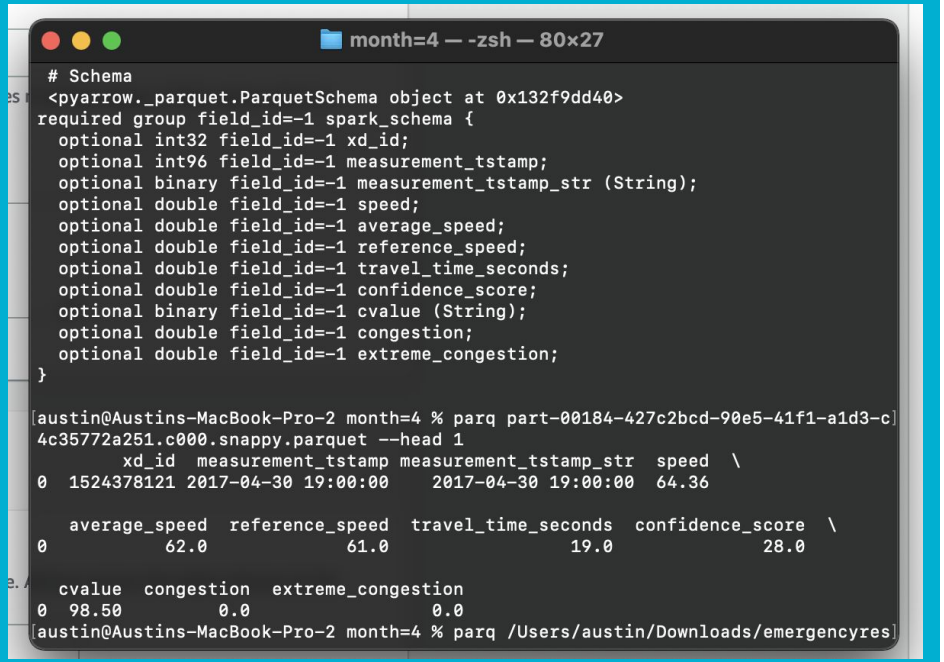


# Athena


aws
Services
Search for services, features, blogs, docs, and more [Option+S]
N. Virginia
voclabs/user1796666=austin.w.wang@vanderbilt.edu @ 4289-8812-...
Amazon Athena
Query editor
Editor
Recent queries
Saved queries
Settings
Workgroup primary
Data
Data Source
AwsDataCatalog
Database
traffic-db
Tables and views
Create
Filter tables and views
Tables (3)
nfd_incidents_xd_seg
traffic-data
weather-data
Views (0)
group (emdcardnum...
Query 11
group by (hour)
Query 22
group (xdsegid)
Query 25
Query 27
group (hour)
SELECT
emdcardnumber,
COUNT(*) AS "count",
AVG(response_time_sec) AS "avg_response_time_sec",
AVG(dist_to_seg) AS "avg_dist_to_seg",
(AVG(response_time_sec) / AVG(dist_to_seg)) AS "avg_response_to_dist_ratio"
FROM "traffic-db"."nfd_incidents_xd_seg"
WHERE
emdcardnumber IS NOT NULL
AND response_time_sec > 0
AND dist_to_seg > 0
GROUP BY emdcardnumber;
SQL Ln 7, Col 33
Run again
Cancel
Save
Clear
Create
Completed
Time in queue: 0.15 sec
Run time: 1.104 sec
Data scanned: 345.39 KB
Results (79)
Copy
Download results
Search rows
# emdcardnumber count avg_response_time_sec avg_dist_to_seg avg_response_to_dist_ratio
1 29A2V 530 384.18490566037735 3.523172691011396 109.04515314862115
2 29B1 7476 385.58681112894595 3.1614990289033083 121.96328627774467
3 29D8V 19 316.8421052631579 3.137267320658179 100.99302127581733
4 29D7 160 384.5875 3.502680991734246 109.798037819477
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

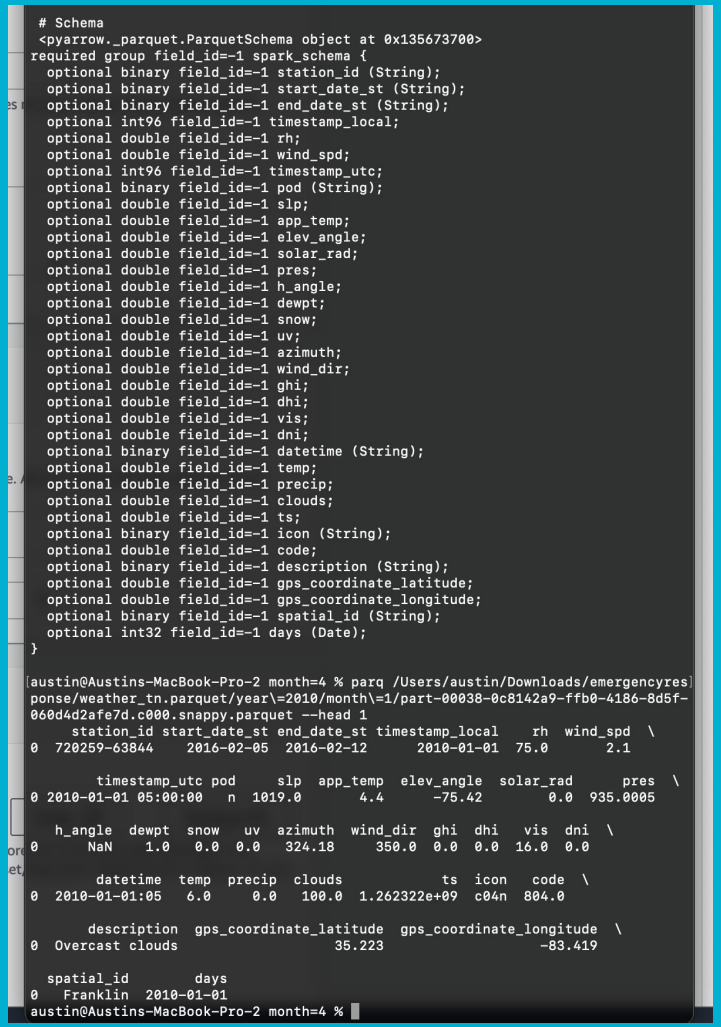
# Sample Queries

# Creating Tables in Athena




traffic-data

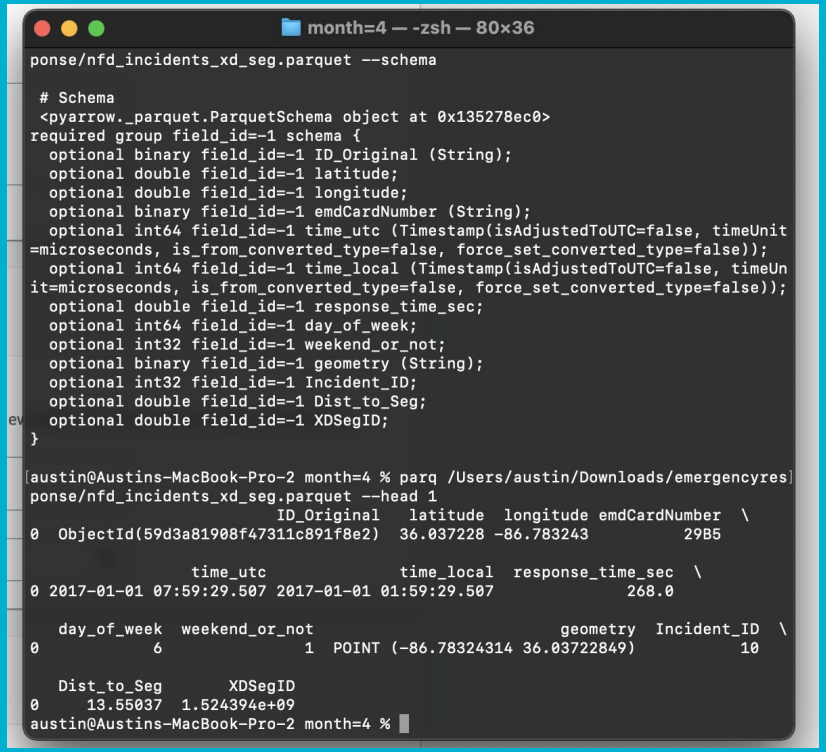
```
CREATE EXTERNAL TABLE IF NOT EXISTS `traffic-db`.`traffic-data` (
  `xd_id` int,
  `measurement_tstamp` bigint,
  `measurement_tstamp_str` string,
  `speed` double,
  `average_speed` double,
  `travel_time_seconds` double,
  `confidence_score` double,
  `cvalue` string,
  `congestion` double,
  `extreme_congestion` double
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://vandy-bucket/traffic.parquet/'
TBLPROPERTIES ('has_encrypted_data'='false');
```



```
 # Schema
 <pyarrow._parquet.ParquetSchema object at 0x135673700>
required group field_id=-1 spark_schema {
  optional binary field_id=-1 station_id (String);
  optional binary field_id=-1 start_date_st (String);
  optional binary field_id=-1 end_date_st (String);
  optional int96 field_id=-1 timestamp_local;
  optional double field_id=-1 rh;
  optional double field_id=-1 wind_spd;
  optional int96 field_id=-1 timestamp_utc;
  optional binary field_id=-1 pod (String);
  optional double field_id=-1 slp;
  optional double field_id=-1 app_temp;
  optional double field_id=-1 elev_angle;
  optional double field_id=-1 solar_rad;
  optional double field_id=-1 pres;
  optional double field_id=-1 h_angle;
  optional double field_id=-1 dewpt;
  optional double field_id=-1 snow;
  optional double field_id=-1 uv;
  optional double field_id=-1 azimuth;
  optional double field_id=-1 wind_dir;
  optional double field_id=-1 ghi;
  optional double field_id=-1 dhi;
  optional double field_id=-1 vis;
  optional double field_id=-1 dni;
  optional binary field_id=-1 datetime (String);
  optional double field_id=-1 temp;
  optional double field_id=-1 precip;
  optional double field_id=-1 clouds;
  optional double field_id=-1 ts;
  optional binary field_id=-1 icon (String);
  optional double field_id=-1 code;
  optional binary field_id=-1 description (String);
  optional double field_id=-1 gps_coordinate_latitude;
  optional double field_id=-1 gps_coordinate_longitude;
  optional binary field_id=-1 spatial_id (String);
  optional int32 field_id=-1 days (Date);
}

[austin@Austins-MacBook-Pro-2 month=4 % parq /Users/austin/Downloads/emergencyres]
ponse/weather_tn.parquet/year\=2010/month\=1/part-00038-0c8142a9-ffb0-4186-8d5f-
060d4d2afe7d.c000.snappy.parquet --head 1
     station_id start_date_st end_date_st timestamp_local    rh  wind_spd  \
0  720259-63844    2016-02-05  2016-02-12      2010-01-01  75.0       2.1

        timestamp_utc pod     slp  app_temp  elev_angle  solar_rad      pres  \
0 2010-01-01 05:00:00   n  1019.0       4.4      -75.42        0.0  935.0005

   h_angle  dewpt  snow   uv  azimuth  wind_dir  ghi  dhi   vis  dni  \
0      NaN    1.0   0.0  0.0   324.18     350.0  0.0  0.0  16.0  0.0

        datetime  temp  precip  clouds            ts  icon   code  \
0  2010-01-01:05   6.0     0.0   100.0  1.262322e+09  c04n  804.0

       description  gps_coordinate_latitude  gps_coordinate_longitude  \
0  Overcast clouds                   35.223                   -83.419

  spatial_id        days
0   Franklin  2010-01-01
austin@Austins-MacBook-Pro-2 month=4 %
```

**weather-data**

```
CREATE EXTERNAL TABLE IF NOT EXISTS `traffic-db`.`weather-data` (
  `station_id` string,
  `start_date_st` string,
  `end_data_st` string,
  `timestamp_local` bigint,
  `rh` double,
  `wind_spd` double,
  `timestamp_utc` bigint,
  `pod` string,
  `slp` double,
  `app_temp` double,
  `elev_angle` double,
  `solar_rad` double,
  `pre` double,
  `h_angle` double,
  `dewpt` double,
  `snow` double,
  `uv` double,
  `azimuth` double,
  `wind_dir` double,
  `ghi` double,
  `dhi` double,
  `vis` double,
  `dni` double,
  `datetime` string,
  `temp` double,
  `precip` double,
  `clouds` double,
  `ts` double,
  `icon` string,
  `code` double,
  `description` string,
  `gps_coordinate_latitude` double,
  `gps_coordinate_longtitude` double,
  `spatial_id` string,
  `days` date
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://vandy-bucket/weather_tn.parquet/'
TBLPROPERTIES ('has_encrypted_data'='false');
```




**nfd_incidents_xd_seg**

```
CREATE EXTERNAL TABLE IF NOT EXISTS `traffic-db`.`nfd_incidents_xd_seg` (
  `id_original` string,
  `latitude` double,
  `longitude` double,
  `emdcardnumber` string,
  `time_utc` bigint,
  `time_local` bigint,
  `response_time_sec` double,
  `day_of_week` int,
  `weekend_or_not` int,
  `geometry` string,
  `incident_id` int,
  `dist_to_seg` double,
  `xdsegid` double
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
WITH SERDEPROPERTIES (
  'serialization.format' = '1'
) LOCATION 's3://vandy-bucket/nfd_incidents_xd_seg.parquet/'
TBLPROPERTIES ('has_encrypted_data'='false');
```

## Count number of entries in each table

### nfd_incidents_xd_seg

```
SELECT COUNT(*) FROM "traffic-db"."nfd_incidents_xd_seg";
-- result: 29765
-- time in queue: 0.175 sec
-- run time: 0.672 sec
```

### traffic-data

```
SELECT COUNT(*) FROM "traffic-db"."traffic-data";
-- result: 2949783075
-- time in queue: 0.206 sec
-- run time: 0.799 sec
```

### weather-data

```
SELECT COUNT(*) FROM "traffic-db"."weather-data";
-- result: 20792532
-- time in queue: 0.255 sec
-- Run time: 1.175 sec
```

# Order by types of incidents

```sql
SELECT
    emdcardnumber,
    COUNT(*) AS "count",
    AVG(response_time_sec) AS "avg_response_time_sec",
    AVG(dist_to_seg) AS "avg_dist_to_seg",
    (AVG(response_time_sec) / AVG(dist_to_seg)) AS
        "avg_response_to_dist_ratio"
FROM "traffic-db"."nfd_incidents_xd_seg"
WHERE
    emdcardnumber IS NOT NULL
    AND response_time_sec > 0
    AND dist_to_seg > 0
GROUP BY emdcardnumber
ORDER BY COUNT(*) DESC;
```

**Results** (79)

Copy | Download results

Search rows

1

| # | emdcardnumber | count | avg_response_time_sec | avg_dist_to_seg | avg_response_to_dist_ratio |
|---|---|---|---|---|---|
| 1 | 29B1 | 7476 | 385.58681112894595 | 3.1614990289033083 | 121.96328627774467 |
| 2 | 29B5 | 2678 | 426.7176997759522 | 3.7595676700519376 | 113.50180053284085 |
| 3 | 29A2 | 1937 | 382.8451213216314 | 3.3377674738187095 | 114.70095635021026 |
| 4 | 29D2P | 1421 | 452.96692470091483 | 3.8825778410146508 | 116.66654044018833 |
| 5 | 29B1V | 890 | 402.8 | 3.3581716348128676 | 119.94622187392928 |
| 6 | 29D2M | 868 | 335.2741935483871 | 4.752741127079406 | 70.5433316445778 |
| 7 | 29B5U | 667 | 420.2608695652174 | 3.6008681331512005 | 116.71098580259246 |
| 8 | 29A2V | 530 | 384.18490566037735 | 3.523172691011396 | 109.04515314862115 |
| 9 | 29D2L | 469 | 381.727078891258 | 3.6399152441404428 | 104.87251853068955 |
| 10 | 29D5 | 452 | 459.1305309734513 | 3.7366575388136694 | 122.87198551227632 |
| 11 | 29B1U | 306 | 392.73202614379085 | 3.4164657075244156 | 114.95272008111739 |
| 12 | 29A1 | 300 | 377.81333333333333 | 3.333745297148055 | 113.32999364304892 |
| 13 | 29D3 | 215 | 443.6046511627907 | 3.53825842802923 | 125.3737283994469 |
| 14 | 29B5V | 179 | 401.6368715083799 | 3.153663592920708 | 127.35564833546854 |
| 15 | 29D5V | 169 | 460.33727810650885 | 3.15476381333107 36 | 145.91814327312343 |
| 16 | 29D7 | 160 | 384.5875 | 3.502680991734246 | 109.798037819477 |
| 17 | 29B3 | 147 | 392.421768707483 | 3.959533833703557 | 99.1080731189082 |
| 18 | 29D5U | 111 | 454.990990990991 | 2.7354898286471854 | 166.3288915301883 |

Types of incidents with high response time

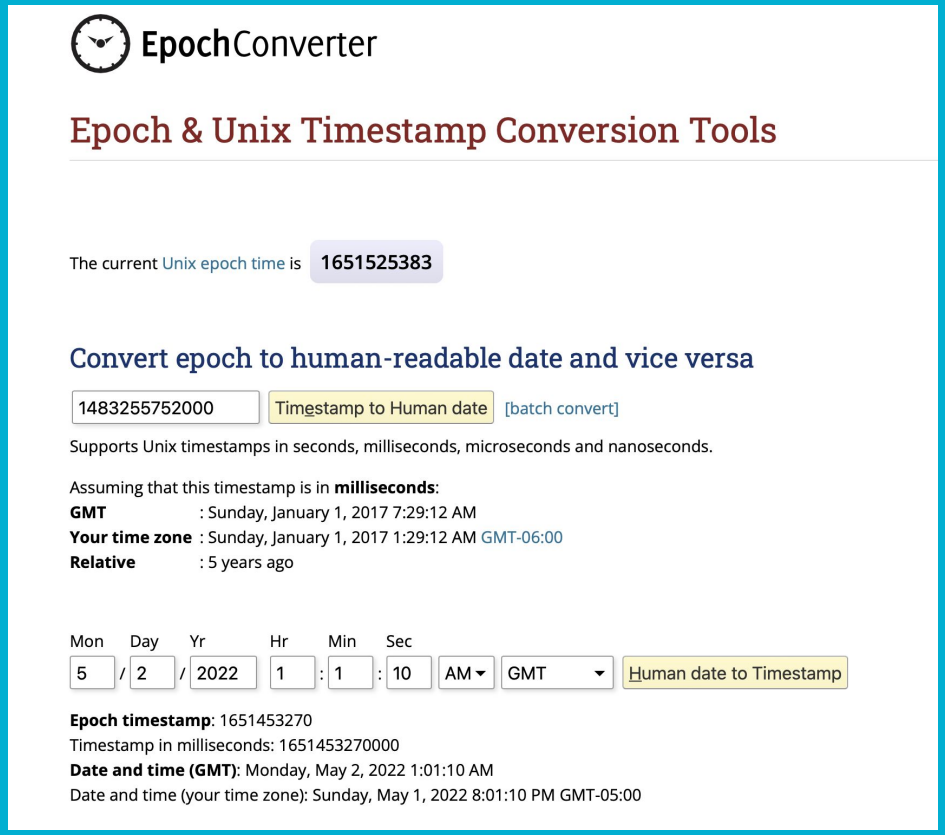
WHERE filters out bad data

```sql
SELECT
    emdcardnumber,
    COUNT(*) AS "count",
    AVG(response_time_sec) AS "avg_response_time_sec",
    AVG(dist_to_seg) AS "avg_dist_to_seg",
    (AVG(response_time_sec) / AVG(dist_to_seg)) AS "avg_response_to_dist_ratio"
FROM "traffic-db"."nfd_incidents_xd_seg"
WHERE
    emdcardnumber IS NOT NULL
    AND response_time_sec > 0
    AND dist_to_seg > 0
GROUP BY emdcardnumber
ORDER BY AVG(response_time_sec) DESC;
```

| # | emdcardnumber | count | avg_response_time_sec | avg_dist_to_seg | avg_response_to_dist_ratio |
|---|---|---|---|---|---|
| 1 | 29D9Y | 3 | 807.6666666666666 | 4.400833792109867 | 183.5258282452543 |
| 2 | 29D2S | 6 | 706.6666666666666 | 3.3099958218075387 | 213.49473072167405 |
| 3 | 29D1F | 4 | 699.25 | 7.87716802339159 | 88.76921222494518 |
| 4 | 29D7V | 11 | 662.8181818181819 | 4.558864290073672 | 145.39107541792401 |
| 5 | 29B3Y | 6 | 568.0 | 3.059336271348967 | 185.66118583281767 |
| 6 | 29D2K | 1 | 564.0 | 12.776684631855687 | 44.14290688476394 |
| 7 | 29D6U | 25 | 515.4 | 3.779929741271957 | 136.35174071425106 |
| 8 | 29D7Y | 1 | 498.0 | 1.097216878954357 | 453.8756280112933 |
| 9 | 29D5Y | 32 | 494.59375 | 3.333776397871856 | 148.35840529548653 |
| 10 | 29B5Y | 8 | 489.75 | 3.194528804816955 | 153.30899482312304 |
| 11 | 29D3Y | 6 | 486.0 | 4.292936376511636 | 113.20922496291803 |
| 12 | 29D8U | 2 | 483.0 | 0.710973555605364 | 679.3501617493296 |
| 13 | 29D3U | 78 | 460.9230769230769 | 4.097681999203102 | 112.48385721798691 |
| 14 | 29D4U | 5 | 460.6 | 7.9613078132785144 | 57.854816168742275 |
| 15 | 29D5V | 169 | 460.33727810650885 | 3.1547638133310736 | 145.91814327312343 |
| 16 | 29D5 | 452 | 459.1305309734513 | 3.7366575388136694 | 122.87198551227632 |
| 17 | 29B2Y | 6 | 457.6666666666667 | 0.9676738607290486 | 472.9554917623373 |
| 18 | 29D8Y | 1 | 457.0 | 1.8736019477791932 | 243.91520330222144 |

Hour with the most incidents

| # | hour | _col1 |
|---|---|---|
| 1 | 17 | 2303 |
| 2 | 16 | 2081 |
| 3 | 15 | 1984 |
| 4 | 18 | 1874 |
| 5 | 14 | 1670 |
| 6 | 13 | 1613 |
| 7 | 12 | 1609 |
| 8 | 19 | 1554 |
| 9 | 11 | 1403 |
| 10 | 20 | 1307 |
| 11 | 10 | 1231 |
| 12 | 21 | 1197 |
| 13 | 7 | 1190 |
| 14 | 8 | 1163 |
| 15 | 9 | 1105 |
| 16 | 22 | 1032 |
| 17 | 6 | 931 |
| 18 | 23 | 878 |




```
1  SELECT HOUR(from_unixtime(time_local/1000)) AS "hour", COUNT(*)
2  FROM "traffic-db"."nfd_incidents_xd_seg"
3  GROUP BY HOUR(from_unixtime(time_local/1000))
4  ORDER BY COUNT(*) DESC;
```

```sql
SELECT
    HOUR(from_unixtime(time_local/1000)) AS "hour",
    COUNT(*) AS "count",
    AVG(response_time_sec) AS "avg_response_time",
    AVG(dist_to_seg) AS "avg_dist_to_seg",
    (AVG(response_time_sec) / AVG(dist_to_seg)) AS "avg_response_to_dist_ratio"
FROM "traffic-db"."nfd_incidents_xd_seg"
GROUP BY HOUR(from_unixtime(time_local/1000))
ORDER BY AVG(response_time_sec) DESC;
```

# Hours with long response time

| # | hour | count | avg_response_time | avg_dist_to_seg | avg_response_to_dist_ratio |
|---|---|---|---|---|---|
| 1 | 2 | 638 | 447.84602917341977 | 3.778233790810532 | 118.53317025078664 |
| 2 | 3 | 582 | 432.2536496350365 | 4.029446882839729 | 107.27369343814459 |
| 3 | 1 | 631 | 423.0238500851789 | 3.6953603886625976 | 114.47431524757917 |
| 4 | 4 | 469 | 416.38137472283813 | 3.3544287388752054 | 124.12884790107587 |
| 5 | 5 | 659 | 406.78343949044586 | 3.463554060833292 | 117.44682841548556 |
| 6 | 6 | 931 | 404.27659574468083 | 3.3443887305931046 | 120.88205896836195 |
| 7 | 0 | 661 | 403.0271132376395 | 3.7431274657949727 | 107.67122330739112 |
| 8 | 17 | 2303 | 395.8197701149425 | 3.5040504295669734 | 112.96063743119629 |
| 9 | 8 | 1163 | 395.6030330062444 | 3.4374118666987163 | 115.08746939486794 |
| 10 | 23 | 878 | 393.0719424460432 | 3.8952080369031146 | 100.91166857381899 |
| 11 | 7 | 1190 | 390.1402278702892 | 3.401522727242076 | 114.69575809261501 |
| 12 | 22 | 1032 | 389.10766045548655 | 3.7528051914190845 | 103.6844815033816 |
| 13 | 9 | 1105 | 388.8916116870877 | 3.3175795611533494 | 117.22148768962464 |
| 14 | 11 | 1403 | 388.7903703703704 | 3.1629427600310964 | 122.92045726636799 |
| 15 | 18 | 1874 | 382.4417693169093 | 3.39152380793513 | 112.76399370162531 |
| 16 | 16 | 2081 | 381.3605236656596 | 3.5280765646022703 | 108.0930406930246 |
| 17 | 21 | 1197 | 380.4735013032146 | 3.266796723011731 | 116.46684307692331 |
| 18 | 15 | 1984 | 375.8127637130802 | 3.4656756327737335 | 108.43852787581872 |

```sql
SELECT
    "description",
    MIN(app_temp) AS "temp (min)",
    MAX(app_temp) AS "temp (max)",
    MIN(clouds) AS "clouds (min)",
    MAX(clouds) AS "clouds (max)",
    MIN(snow) AS "snow (min)",
    MAX(snow) AS "snow (max)",
    MIN(precip) AS "precip (min)",
    MAX(precip) AS "precip (max)",
    MIN(uv) AS "uv (min)",
    MAX(uv) AS "uv (max)",
    COUNT(*) AS "count"
FROM "traffic-db"."weather-data"
GROUP BY "description";
```

| description | temp (min) | temp (max) | clouds (min) | clouds (max) | snow (min) | snow (max) | precip (min) | precip (max) | uv (min) | uv (max) | count |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Overcast clouds | -73.2 | 58.7 | 71.0 | 100.0 | 0.0 | 12.5 | 0.0 | 0.5 | 0.0 | 11.2 | 8112938 |
| Clear Sky | -78.6 | 81.0 | 0.0 | 1.0 | 0.0 | 12.5 | 0.0 | 0.5 | 0.0 | 11.3 | 6043266 |
| Moderate rain | -4.6 | 50.3 | 0.0 | 100.0 | 0.0 | 12.0 | 4.1 | 8.0 | 0.0 | 10.0 | 87065 |
| Snow | -50.8 | 2.3 | 0.0 | 100.0 | 16.099999666214 | 100.0 | 1.7 | 12.0 | 0.0 | 3.0 | 10359 |
| Scattered clouds | -76.7 | 76.8 | 21.0 | 40.0 | 0.0 | 12.5 | 0.0 | 0.5 | 0.0 | 11.2 | 1822752 |
| Light snow | -75.9 | 3.7 | 0.0 | 100.0 | 3.5560001134872 | 37.5 | 0.508 | 5.0 | 0.0 | 4.6 | 30052 |
| Heavy snow | -24.3 | 3.8 | 0.0 | 100.0 | 36.399998664856 | 487.5 | 4.3 | 39.0 | 0.0 | 1.5 | 771 |
| Broken clouds | -75.9 | 69.5 | 41.0 | 70.0 | 0.0 | 12.5 | 0.0 | 0.5 | 0.0 | 11.2 | 2588518 |
| Few clouds | -76.9 | 52.7 | 2.0 | 20.0 | 0.0 | 12.5 | 0.0 | 0.5 | 0.0 | 11.2 | 961175 |
| Light rain | -5.9 | 62.6 | 0.0 | 100.0 | 0.0 | 8.0 | 0.508 | 4.0 | 0.0 | 10.1 | 1096212 |
| Mix snow/rain | -7.0 | 3.4 | 0.0 | 100.0 | 1.0160000324249 | 37.0 | 0.508 | 18.5 | 0.0 | 4.3 | 11133 |
| Heavy rain | -4.0 | 45.7 | 0.0 | 100.0 | 0.0 | 24.0 | 8.1 | 292.1 | 0.0 | 9.2 | 28291 |

```
[36] !pip install pandas fiona shapely pyproj rtree
```

```
[37] !pip install geopandas
```

```
[61] results = roads.merge(incidents, on="XDSegID")
```

```
results.head(1)
```

| | OID | XDSegID | PreviousXD | NextXDSegI | FRC | ToExport | Miles | Lanes | RoadNumber | RoadName | ... | longitude | emdCardNumber | time_utc | time_local | respons |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 8073455 | 155711459 | NaN | NaN | 5 | None | 0.028749 | NaN | | BERRY RD | ... | -86.770009 | 29D2L | 2020-07-04 03:42:22.387 | 2020-07-03 22:42:22.387 | |

1 rows × 40 columns

Joining incidents and roads

```
[79] df = results.groupby(['XDSegID']).size().reset_index(name='counts')
```

```
df
```

| | XDSegID | counts |
|---|---|---|
| 0 | 155711459 | 1 |
| 1 | 155742183 | 2 |
| 2 | 155795795 | 5 |
| 3 | 155800999 | 1 |
| 4 | 155839691 | 1 |
| ... | ... | ... |
| 3333 | 1524643823 | 15 |
| 3334 | 1524644550 | 9 |
| 3335 | 1524644585 | 6 |
| 3336 | 1524645372 | 4 |
| 3337 | 1524646899 | 12 |

3338 rows × 2 columns

```
df.loc[df['counts'].idxmax()]
```

```
XDSegID    441552606
counts           103
Name: 808, dtype: int64
```

# Sample Visualization

# Analysis results

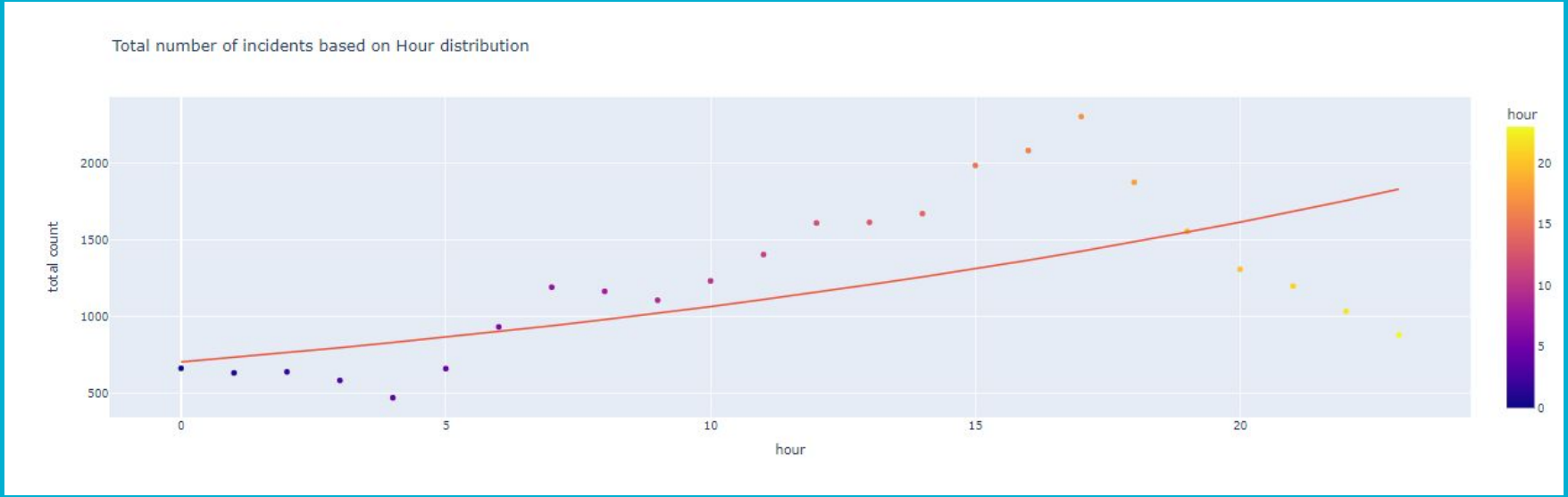
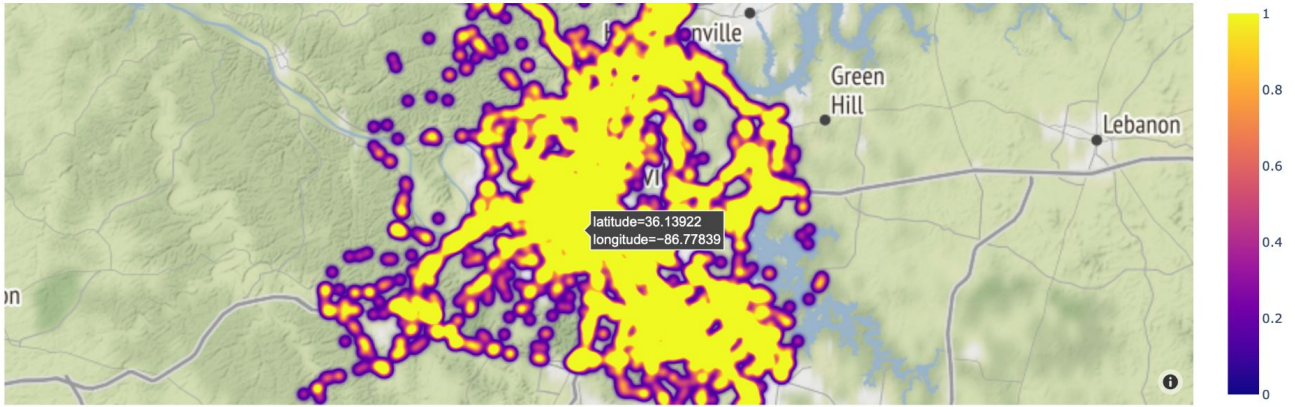



Figure 1. Distribution of total number of incidents based on Hour (0 - 23 hr)

# Figure 2. Total number of incidents in different days of week

# Fig 3. Density Graph (Incidents)

```
[ ]  df = df[df['weekend_or_not'] == 1]

[ ]  import plotly.express as px

     fig = px.density_mapbox(df, lat='latitude', lon='longitude', radius=10,
                             center=dict(lat=0, lon=180), zoom=0,
                             mapbox_style="stamen-terrain")

     fig.show()
```



# Fig. 4 Density Graph (roads)

```
from pandas.io.parsers.readers import read_csv
import plotly.express as px
df = read_csv('/content/group-xdsegid.csv')
fig = px.scatter_geo(df, lat='_col1', lon='_col2',
                     color="xd_count",
                     size="xd_count")

fig.update_layout(
        title = 'XD_Count',
        geo = dict(
            scope='usa',
            projection_type='albers usa',
            showland = True,
            landcolor = "rgb(250, 250, 250)",
            subunitcolor = "rgb(217, 217, 217)",
            countrycolor = "rgb(217, 217, 217)",
            countrywidth = 0.5,
            subunitwidth = 0.5
        ),
    )
fig.update_geos(fitbounds="locations")

fig.show()
```

# Thank you!