Nicolas Sebastian Jasman

SQL and Visualization Portfolio

This portfolio contains SQL queries which I have composed in order to complete specific tasks for my data analysis and the visualization.

Contact Information

Email
n.c.jasman@gmail.com

LinkedIn
linkedin.com/in/nicolassj

Austin Bikeshare

This particular dataset provides information about bike trips and stations in Austin, Texas.

Dataset Schema Source: BigQuery

The schema for the table is as follows:

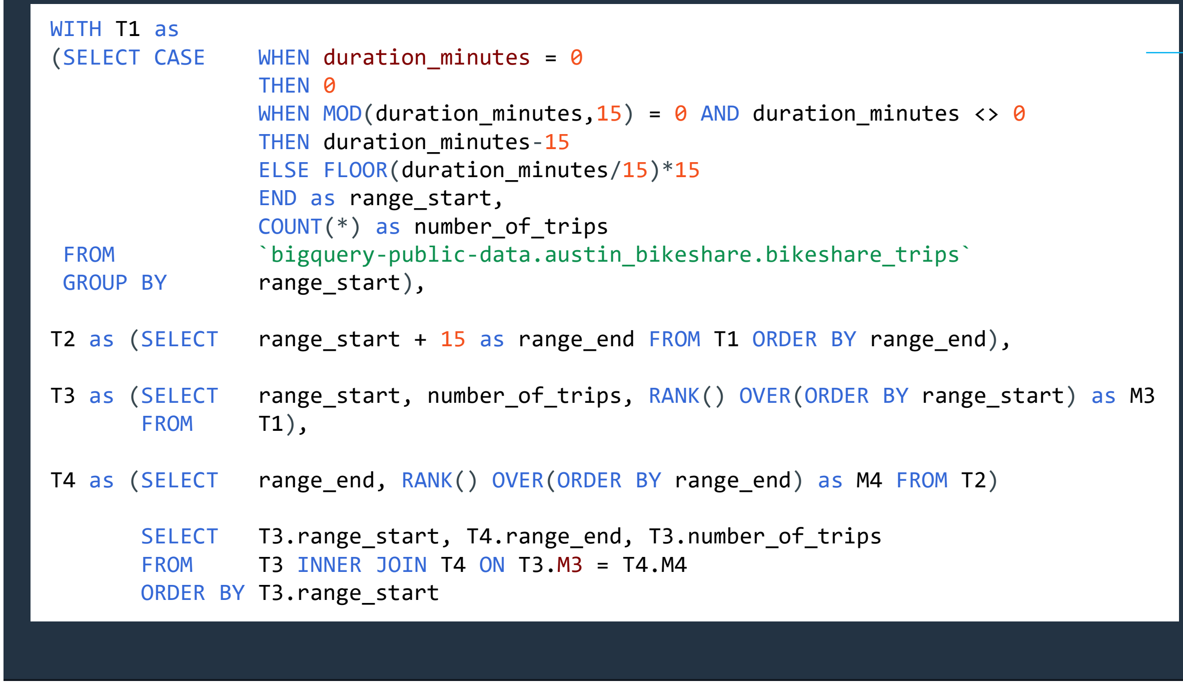
Field Name	Data Type
trip_id	INTEGER
subscriber_type	STRING
bikeid	STRING
start_time	TIMESTAMP
start_station_id	INTEGER
start_station_name	STRING
end_station_id	STRING
end_station_name	STRING
duration_minutes	INTEGER

Objective

Segment the bike trip duration in 15-minute intervals and determine the amount of trips which falls into each category.

Query

This query was executed in BigQuery. The primary table name is **bigquery-public-data.austin_bikeshare.bikeshare_trips.**



```
WITH T1 as
(SELECT CASE    WHEN duration_minutes = 0
                THEN 0
                WHEN MOD(duration_minutes,15) = 0 AND duration_minutes <> 0
                THEN duration_minutes-15
                ELSE FLOOR(duration_minutes/15)*15
                END as range_start,
                COUNT(*) as number_of_trips
 FROM           `bigquery-public-data.austin_bikeshare.bikeshare_trips`
 GROUP BY       range_start),

T2 as (SELECT   range_start + 15 as range_end FROM T1 ORDER BY range_end),

T3 as (SELECT   range_start, number_of_trips, RANK() OVER(ORDER BY range_start) as M3
       FROM     T1),

T4 as (SELECT   range_end, RANK() OVER(ORDER BY range_end) as M4 FROM T2)

       SELECT   T3.range_start, T4.range_end, T3.number_of_trips
       FROM     T3 INNER JOIN T4 ON T3.M3 = T4.M4
       ORDER BY T3.range_start
```

The **CASE** clause's purpose is to group values at the edges of each range to the range preceding it (e.g **15** to **15-30** and 60 to 60-75).

Austin Bikeshare

Query Results Sample

range_start	range_end	number_of_trips
0.0	15.0	896152
15.0	30.0	316713
30.0	45.0	140697
45.0	60.0	94970
60.0	75.0	39431
75.0	90.0	23801
90.0	105.0	15413
105.0	120.0	10351
120.0	135.0	8405
135.0	150.0	6040
150.0	165.0	4213

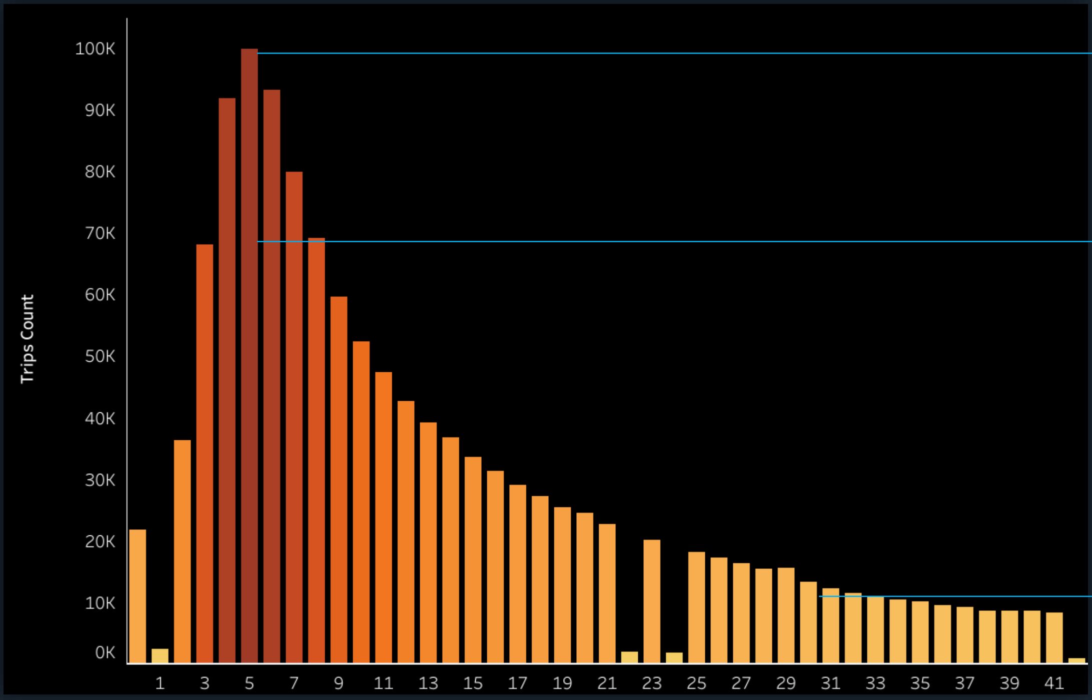
Some intervals are skipped when there are no trips which falls within that category. This happens as the range_start value goes up

Austin Bikeshare

This analysis was conducted on a single-datum basis.

Visualization (Click here to View Full Visualization)

The figure below shows the **trip duration (minutes)** distribution against the **trips count**. It is impractical to show the full graph. Therefore, this partial section is displayed instead.



Trends and Parameters

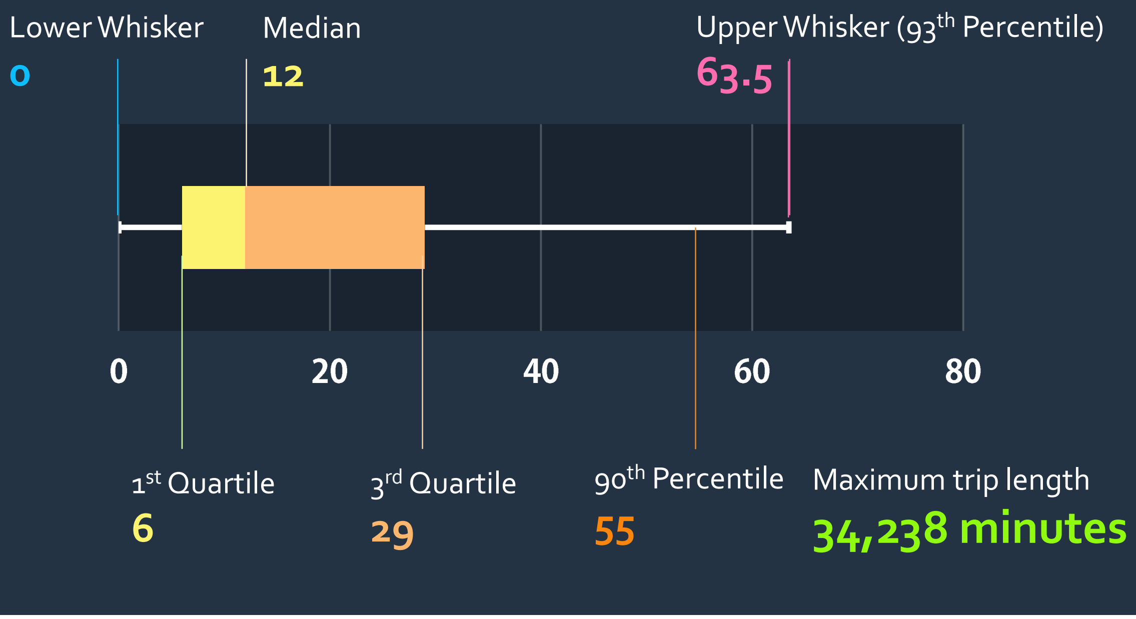
The data count is equal to **1,502,735**, with the **mode** being 5 minutes.

This dataset has a **skewness** value of 63.06, indicating that the data are **extremely** concentrated to the left of the mean.

The mean is **30.30 minutes** with the standard deviation of **134.75 minutes.**

Distribution Details

The following boxplot depicts the distribution of the data.



Lower Whisker
0

Median
12

Upper Whisker (93th Percentile)
63.5

1st Quartile
6

3rd Quartile
29

90th Percentile
55

Maximum trip length
34,238 minutes

Additional Information

75% of trip durations are in the range of **zero to 29 minutes.**

90% of the data are located from the range of **zero to 55.**

>99% of trips have duration of less than **5 hours.**

Gun Violence in the United States

This particular dataset provides information on all gun violence cases in the US from January 2013 to May 2022.

Dataset Sample Source: Kaggle

incident_id	date	state	city	address	n_killed	n_injured
2314858	28/05/2022	Arkansas	Little Rock	W 9th St and Broadway St	0	1
2314409	28/05/2022	Colorado	Denver	3300 block of Clay St	0	1
2314498	28/05/2022	Missouri	Saint Louis	Page Blvd and Vandeventer Ave	0	1
2314485	28/05/2022	South Carolina	Florence	Old River Rd	0	2
2314483	28/05/2022	California	Carmichael	4400 block of Manzanita Ave	1	0
2314481	28/05/2022	Kentucky	Louisville	400 block of M St	0	1

Additional Info

n_killed is the number of death occurred during an incident

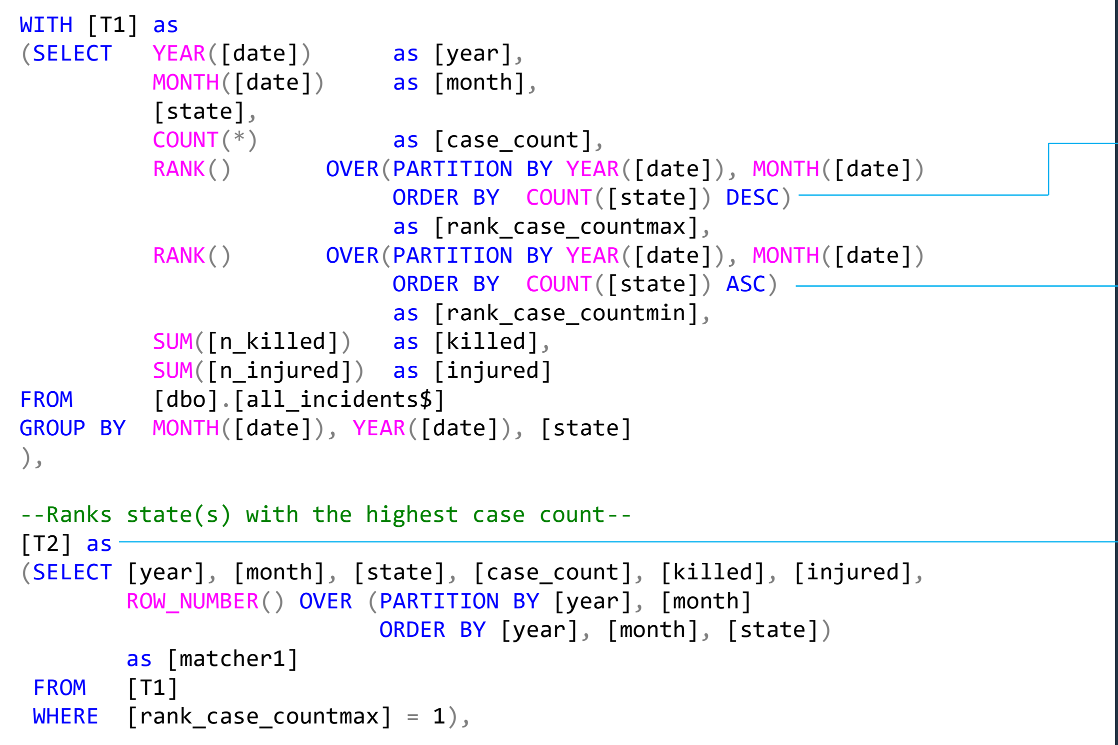
n_injured is the number of people injured during an incident.

Objective

Identifying the state(s) within each month with the highest and lowest case count and stating the cumulative case count, death, and injured in that state in that month.

Query

This query was executed in SSMS. The primary table name is **dbo.all_incidents$**



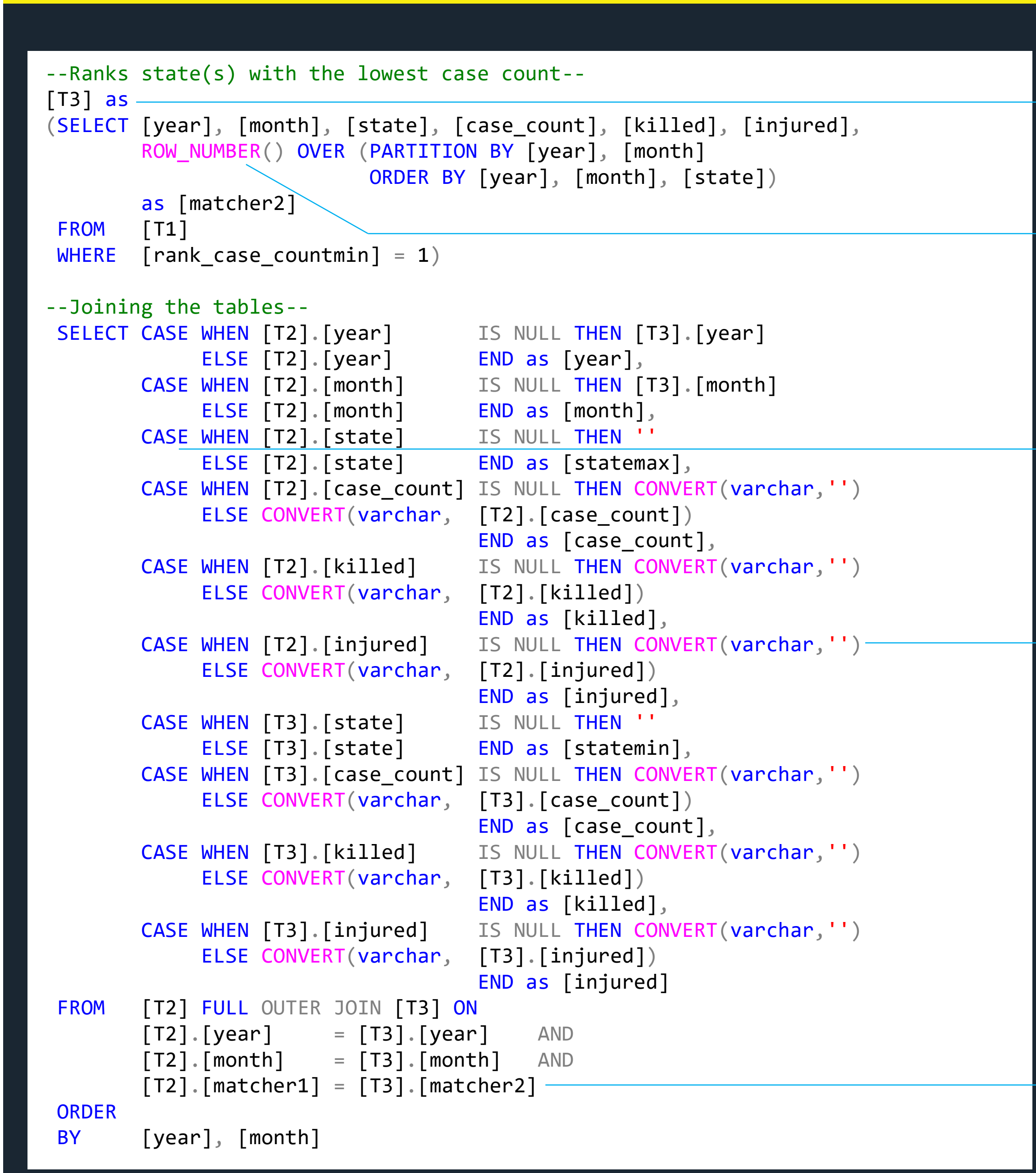
```
WITH [T1] as
(SELECT   YEAR([date])      as [year],
          MONTH([date])     as [month],
          [state],
          COUNT(*)          as [case_count],
          RANK()       OVER(PARTITION BY YEAR([date]), MONTH([date])
                            ORDER BY  COUNT([state]) DESC)
                            as [rank_case_countmax],
          RANK()       OVER(PARTITION BY YEAR([date]), MONTH([date])
                            ORDER BY  COUNT([state]) ASC)
                            as [rank_case_countmin],
          SUM([n_killed])   as [killed],
          SUM([n_injured])  as [injured]
FROM      [dbo].[all_incidents$]
GROUP BY  MONTH([date]), YEAR([date]), [state]
),

--Ranks state(s) with the highest case count--
[T2] as
(SELECT [year], [month], [state], [case_count], [killed], [injured],
        ROW_NUMBER() OVER (PARTITION BY [year], [month]
                           ORDER BY [year], [month], [state])
        as [matcher1]
 FROM   [T1]
 WHERE  [rank_case_countmax] = 1),
```

Ranks states based on case count from **the most.**

Ranks states based on case count from **the least.**

T2 selects state(s) with **most** cases at a given period.

Gun Violence in the United States



```
--Ranks state(s) with the lowest case count--
[T3] as
(SELECT [year], [month], [state], [case_count], [killed], [injured],
        ROW_NUMBER() OVER (PARTITION BY [year], [month]
                           ORDER BY [year], [month], [state])
        as [matcher2]
 FROM   [T1]
 WHERE  [rank_case_countmin] = 1)

--Joining the tables--
 SELECT CASE WHEN [T2].[year]       IS NULL THEN [T3].[year]
             ELSE [T2].[year]       END as [year],
        CASE WHEN [T2].[month]      IS NULL THEN [T3].[month]
             ELSE [T2].[month]      END as [month],
        CASE WHEN [T2].[state]      IS NULL THEN ''
             ELSE [T2].[state]      END as [statemax],
        CASE WHEN [T2].[case_count] IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T2].[case_count])
                                    END as [case_count],
        CASE WHEN [T2].[killed]     IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T2].[killed])
                                    END as [killed],
        CASE WHEN [T2].[injured]    IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T2].[injured])
                                    END as [injured],
        CASE WHEN [T3].[state]      IS NULL THEN ''
             ELSE [T3].[state]      END as [statemin],
        CASE WHEN [T3].[case_count] IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T3].[case_count])
                                    END as [case_count],
        CASE WHEN [T3].[killed]     IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T3].[killed])
                                    END as [killed],
        CASE WHEN [T3].[injured]    IS NULL THEN CONVERT(varchar,'')
             ELSE CONVERT(varchar,  [T3].[injured])
                                    END as [injured]
 FROM   [T2] FULL OUTER JOIN [T3] ON
        [T2].[year]     = [T3].[year]     AND
        [T2].[month]    = [T3].[month]    AND
        [T2].[matcher1] = [T3].[matcher2]
 ORDER
 BY     [year], [month]
```

T3 selects state(s) with **least** cases at a given period.

ROW_NUMBER() syntax is used to match **T2** and **T3** when joining.

Implementing **CASE** to fill the data with **blank** instead when NULL is present.

CONVERT is required to change **integer** data to **varchar**. Blank data are then able to be displayed as **blank** instead of **0 (zero).**

Matching **T2** and **T3** by period and rownumber.

Query Results Sample

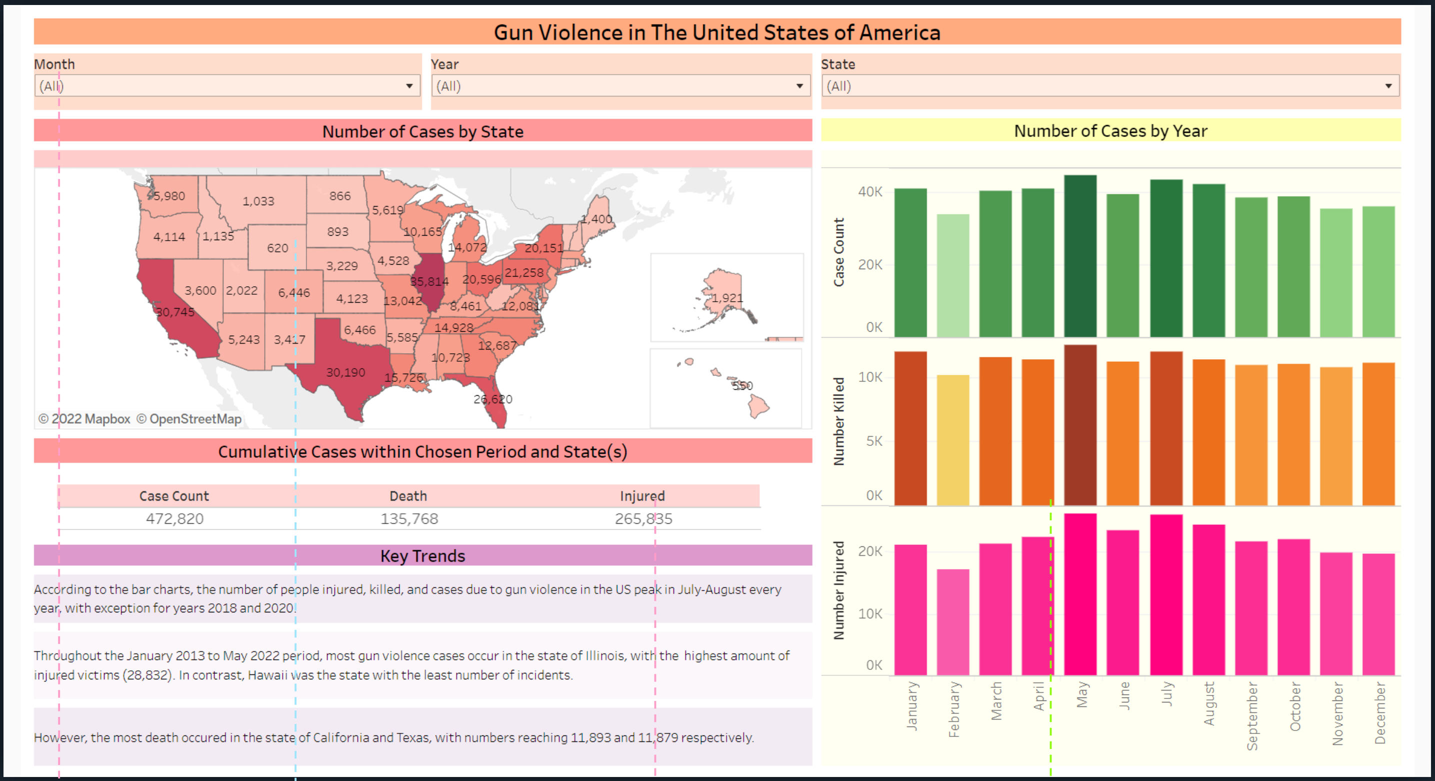
year	month	statemax	case_count	killed	injured	statemin	case_count	killed	injured
2015	7	Illinois	352	77	302	Hawaii	8	0	3
2015	7					South Dakota	8	0	3
2015	8	Illinois	351	85	312	South Dakota	8	0	2
2015	9	Illinois	339	83	314	Hawaii	10	1	2
2015	10	California	266	102	125	Hawaii	7	1	8
2015	10	Illinois	266	42	219				
2015	10	Texas	266	63	119				
2015	11	New York	264	37	129	Hawaii	2	1	0

Cells are left **empty** when the number of states with the **least** and **most** case count are not equal.

Gun Violence in the United States

This visualization was conducted on Tableau.

Dashboard (Click here to View the Interactive Dashboard)



Gun Violence in The United States of America

Month: (All)

Year: (All)

State: (All)

Number of Cases by State

Cumulative Cases within Chosen Period and State(s)

Case Count	Death	Injured
472,820	135,768	265,835

Key Trends

According to the bar charts, the number of people injured, killed, and cases due to gun violence in the US peak in July-August every year, with exception for years 2018 and 2020.

Throughout the January 2013 to May 2022 period, most gun violence cases occur in the state of Illinois, with the highest amount of injured victims (28,832). In contrast, Hawaii was the state with the least number of incidents.

However, the most death occured in the state of California and Texas, with numbers reaching 11,893 and 11,879 respectively.

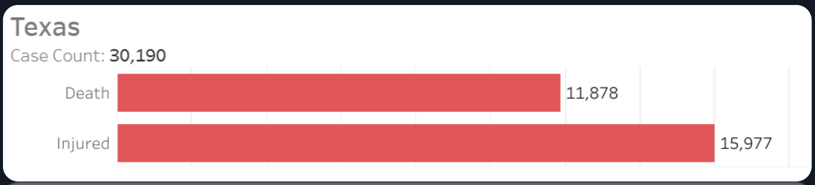
Number of Cases by Year

Interactive selectable features

Shows cumulative amount of variables within chosen constraints.

Illustrates the monthly case count, death count, and injured count within the chosen state(s) and year(s)

Depicts the following information when cursor is placed on a state



Texas

Case Count: 30,190

Global Air Quality – O_3 in the US

This particular dataset provides information about global emission.

Dataset Schema Source: BigQuery

The schema for the table is as follows:

Field Name	Data Type
location	STRING
city	STRING
country	STRING
pollutant	STRING
value	FLOAT
timestamp	TIMESTAMP
unit	STRING
source_name	STRING
latitude	FLOAT
longitude	FLOAT
averaged_over_in_hours	FLOAT
location_geom	GEOGRAPHY

Objective

Calculating the increase or decrease of average monthly pollution by O_3 (ozone) in the US from January 2016 to present day

Query

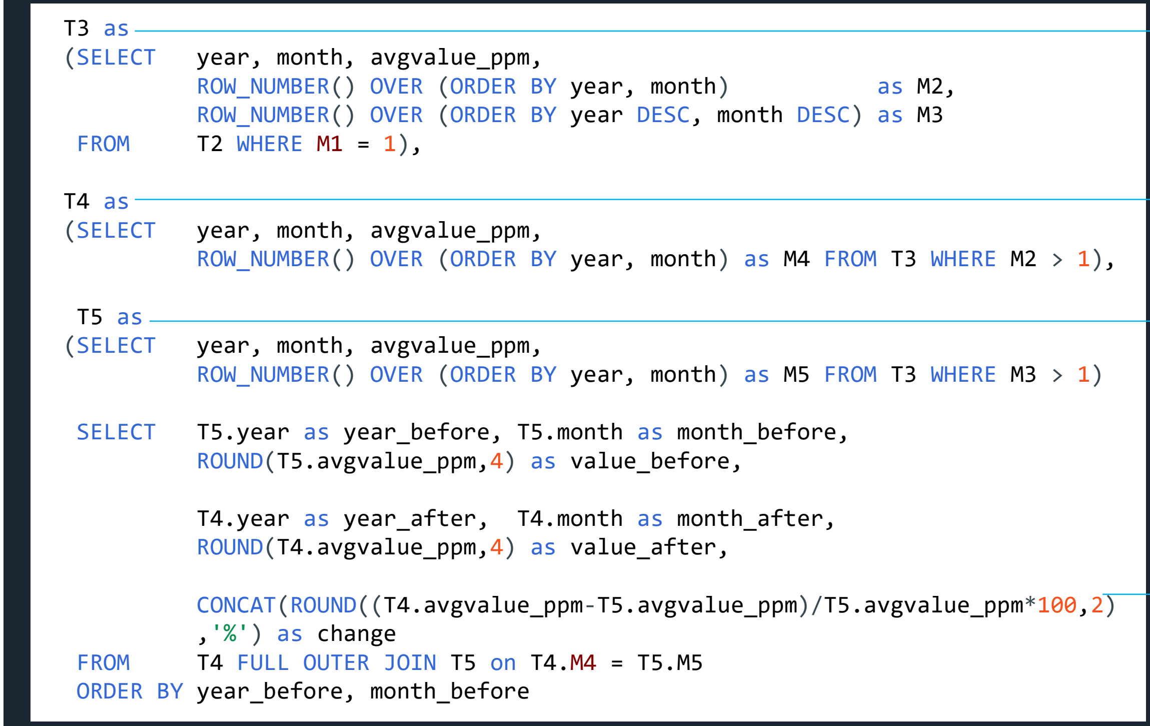
This query was executed in BigQuery. The primary table name is **bigquery-public-data.openaq.global_air_quality.**

```
WITH T1 as
(SELECT   EXTRACT(YEAR from timestamp)    as year,
                    EXTRACT(MONTH from timestamp) as month,
                    CASE WHEN unit LIKE '%g%' THEN value/1000
                    ELSE value           END as value_ppm
 FROM     `bigquery-public-data.openaq.global_air_quality`
 WHERE    country = 'US' AND pollutant = 'o3'),

T2 as
(SELECT   year, month,
          AVG(value_ppm)  OVER (PARTITION BY year, month)
          as avgvalue_ppm,
          ROW_NUMBER()    OVER (PARTITION BY year, month ORDER BY year, month)
          as M1
 FROM     T1
 WHERE    year >= 2016),
```

The **CASE** clause is used to convert rows whose units are **µg/m³** to **ppm**.

Grouping the **average value** of O_3 pollution by month.

Global Air Quality – O_3 in the US



```
T3 as
(SELECT   year, month, avgvalue_ppm,
          ROW_NUMBER() OVER (ORDER BY year, month)           as M2,
          ROW_NUMBER() OVER (ORDER BY year DESC, month DESC) as M3
 FROM     T2 WHERE M1 = 1),

T4 as
(SELECT   year, month, avgvalue_ppm,
          ROW_NUMBER() OVER (ORDER BY year, month) as M4 FROM T3 WHERE M2 > 1),

 T5 as
(SELECT   year, month, avgvalue_ppm,
          ROW_NUMBER() OVER (ORDER BY year, month) as M5 FROM T3 WHERE M3 > 1)

 SELECT   T5.year as year_before, T5.month as month_before,
          ROUND(T5.avgvalue_ppm,4) as value_before,

          T4.year as year_after,  T4.month as month_after,
          ROUND(T4.avgvalue_ppm,4) as value_after,

          CONCAT(ROUND((T4.avgvalue_ppm-T5.avgvalue_ppm)/T5.avgvalue_ppm*100,2)
          ,'%') as change
 FROM     T4 FULL OUTER JOIN T5 on T4.M4 = T5.M5
 ORDER BY year_before, month_before
```

T3 only selects one row from **T2's** window function output

T4 eliminates the **first** row.

T5 eliminates the **last** row.

Analytical function within **CONCAT** to calculate the change percentage.

Query Results Sample

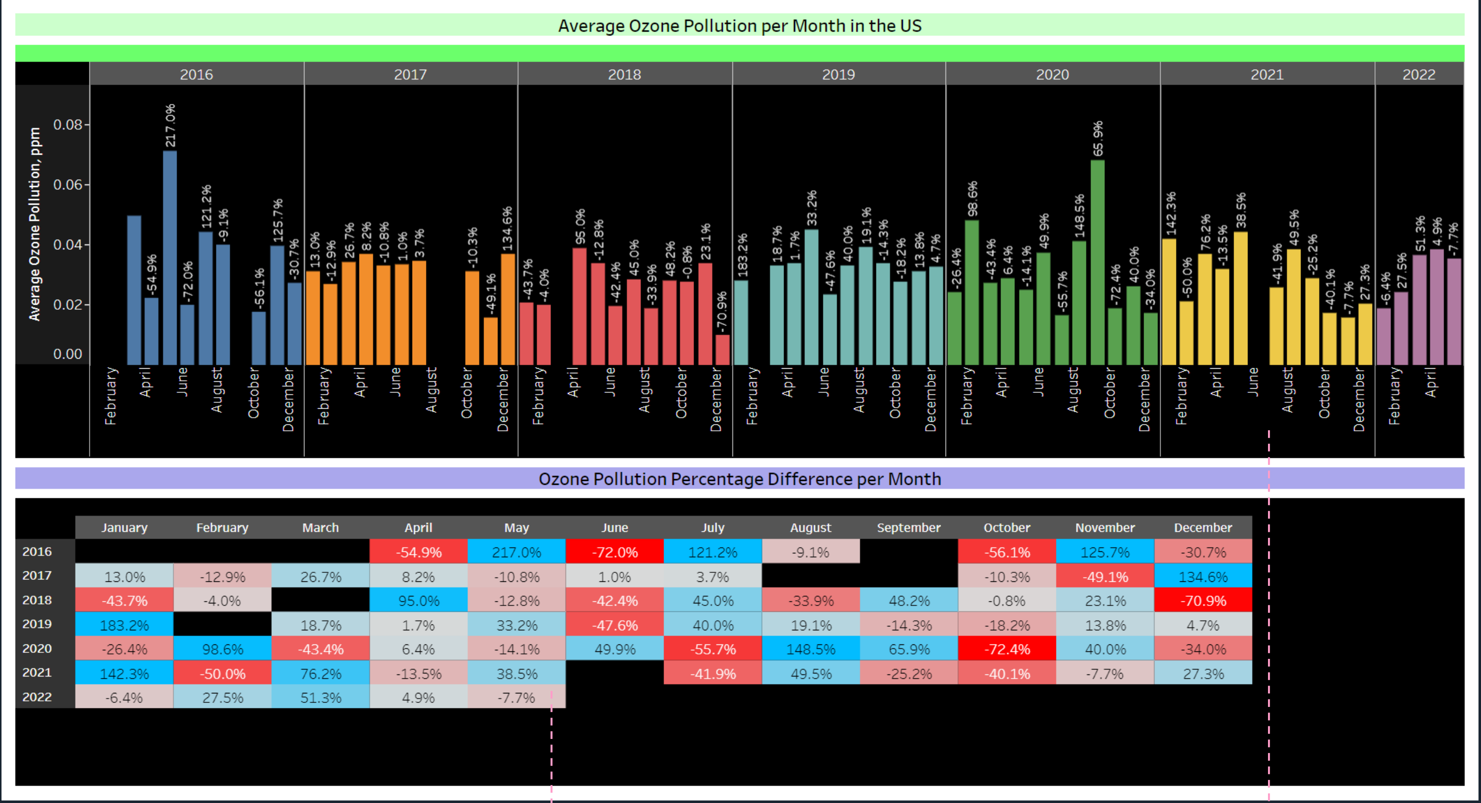
year_before	month_before	value_before	year_after	month_after	value_after	change
2020	9	0.068	2020	10	0.0187	-72.43%
2020	10	0.0187	2020	11	0.0262	40%
2020	11	0.0262	2020	12	0.0173	-33.97%
2020	12	0.0173	2021	1	0.042	142.31%
2021	1	0.042	2021	2	0.021	-50%
2021	2	0.021	2021	3	0.037	76.19%
2021	3	0.037	2021	4	0.032	-13.51%
2021	4	0.032	2021	5	0.0443	38.54%
2021	5	0.0443	2021	7	0.0257	-41.92%
2021	7	0.0257	2021	8	0.0385	49.51%

Some months are skipped when data is not available.

Global Air Quality – O_3 in the US

This visualization was conducted on Tableau.

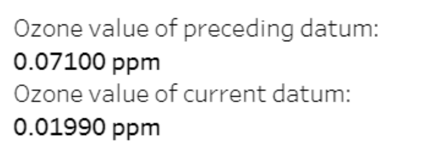
Dashboard (Click here to view full version)



Ozone Pollution Percentage Difference per Month

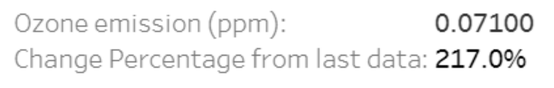
	January	February	March	April	May	June	July	August	September	October	November	December
2016				-54.9%	217.0%	-72.0%	121.2%	-9.1%		-56.1%	125.7%	-30.7%
2017	13.0%	-12.9%	26.7%	8.2%	-10.8%	1.0%	3.7%			-10.3%	-49.1%	134.6%
2018	-43.7%	-4.0%		95.0%	-12.8%	-42.4%	45.0%	-33.9%	48.2%	-0.8%	23.1%	-70.9%
2019	183.2%		18.7%	1.7%	33.2%	-47.6%	40.0%	19.1%	-14.3%	-18.2%	13.8%	4.7%
2020	-26.4%	98.6%	-43.4%	6.4%	-14.1%	49.9%	-55.7%	148.5%	65.9%	-72.4%	40.0%	-34.0%
2021	142.3%	-50.0%	76.2%	-13.5%	38.5%		-41.9%	49.5%	-25.2%	-40.1%	-7.7%	27.3%
2022	-6.4%	27.5%	51.3%	4.9%	-7.7%							

Highlights the O_3 pollution percentage difference between months. Shows the following information.



Ozone value of preceding datum:
0.07100 ppm
Ozone value of current datum:
0.01990 ppm

Displays the average O_3 pollution of each month. Shows the following information.



Ozone emission (ppm): 0.07100
Change Percentage from last data: 217.0%

Key Features

According to the bar chart, the average monthly O_3 pollution in the US peaks in **May 2016**, with number reaching **0.071 ppm,** an increase of **217.0%** from **April 2016.** In contrast, the lowest average pollution was recorded in **December 2017**, only under **0,01 ppm.**

The O_3 emission remained relatively stable in **2017.** The O_3 yearly pollution in the US was the lowest in **2018,** whereas the highest occurred in **2016,** with the numbers being **0.025 ppm** and **0.037 ppm,** respectively.