**US Accidents Visualizations**

***By Somendra, Pablo and Balaji***

**Visualization 1:**

**SPICE:**

WITH sev1 as(SELECT DISTINCT count(id) as Acc\_Sev1, state

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 1

GROUP BY state),

sev2 as(SELECT DISTINCT count(id) as Acc\_Sev2, state

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 2

GROUP BY state),

sev3 as(SELECT DISTINCT count(id) as Acc\_Sev3, state

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 3

GROUP BY state),

sev4 as(SELECT DISTINCT count(id) as Acc\_Sev4, state

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 4

GROUP BY state)

SELECT sev1.state, Acc\_Sev1, Acc\_Sev2, Acc\_Sev3, Acc\_Sev4

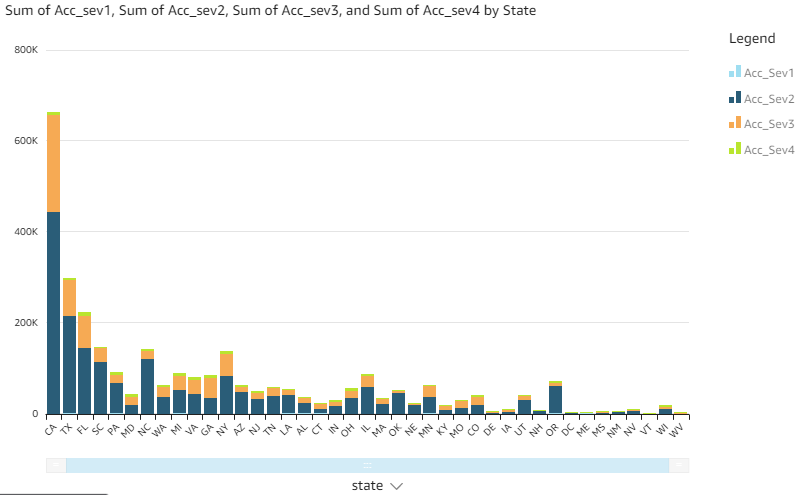
FROM sev1, sev2, sev3, sev4

WHERE sev1.state = sev2.state AND

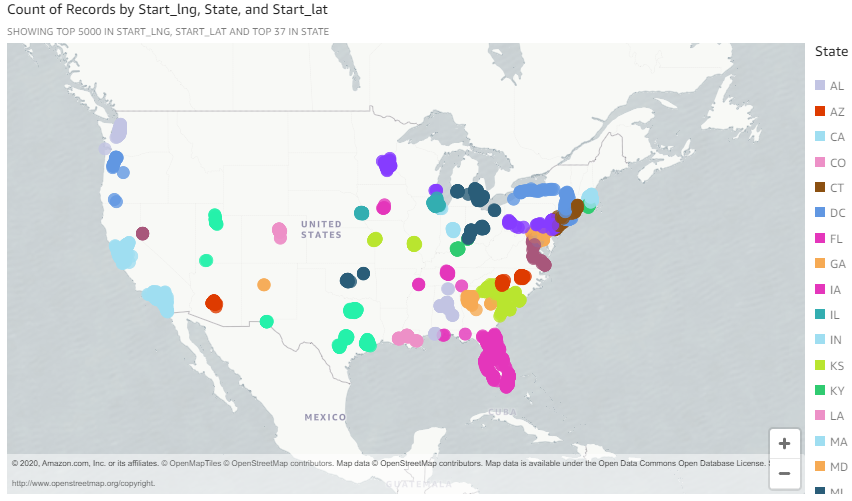
sev2.state = sev3.state AND

sev3.state = sev4.state

ORDER BY sev1.state ASC;



**Visualization 2:**



**Visualization 3:**

**SPICE:**

with cte AS (select

CASE

WHEN date\_diff('minute', start\_time, end\_time)

BETWEEN 0

AND 59 THEN

'minutes'

WHEN date\_diff('minute', start\_time, end\_time)

BETWEEN 60

AND 239 THEN

'hours'

WHEN date\_diff('minute', start\_time, end\_time)

BETWEEN 240

AND 959 THEN

'4hours\_16hours'

WHEN date\_diff('minute', start\_time, end\_time)

BETWEEN 960

AND 1439 THEN

'16hours\_24hours'

WHEN date\_diff('minute', start\_time, end\_time) >= 1440 THEN

'days'

ELSE 'unknown'

END AS time\_range, count(\*) AS cnt

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

GROUP BY start\_time, end\_time)

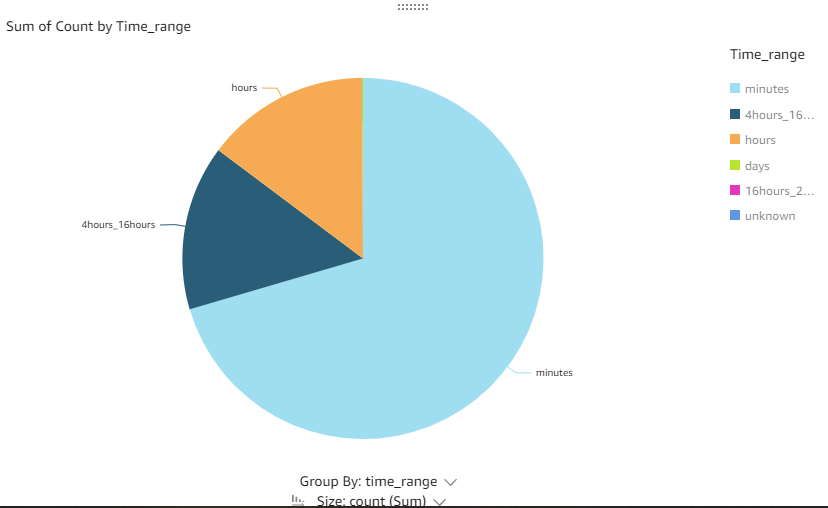
SELECT time\_range,

sum(cnt) AS count

FROM cte

GROUP BY time\_range

ORDER BY count desc;



**Visualization 4:**

**SPICE:**

WITH sev1 as(SELECT DISTINCT count(id) as Acc\_Sev1, weather\_condition

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 1

GROUP BY weather\_condition),

sev2 as(SELECT DISTINCT count(id) as Acc\_Sev2, weather\_condition

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 2

GROUP BY weather\_condition),

sev3 as(SELECT DISTINCT count(id) as Acc\_Sev3, weather\_condition

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 3

GROUP BY weather\_condition),

sev4 as(SELECT DISTINCT count(id) as Acc\_Sev4, weather\_condition

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

WHERE severity = 4

GROUP BY weather\_condition)

SELECT sev1.weather\_condition, Acc\_Sev1, Acc\_Sev2, Acc\_Sev3, Acc\_Sev4

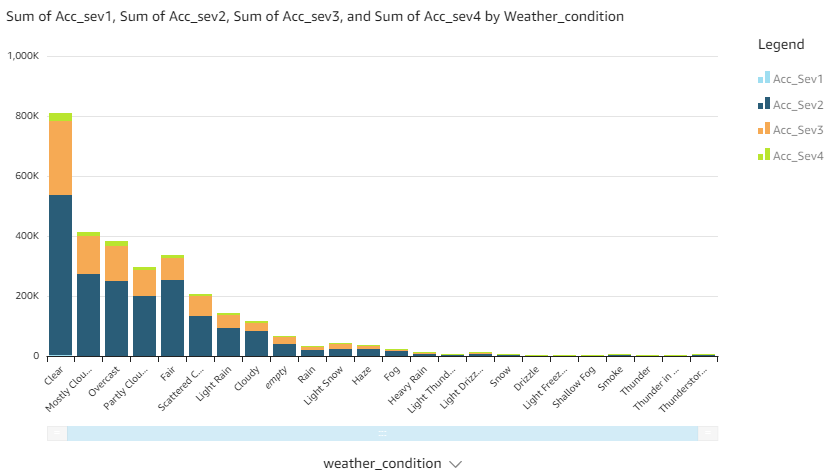
FROM sev1, sev2, sev3, sev4

WHERE sev1.weather\_condition = sev2.weather\_condition AND

sev2.weather\_condition = sev3.weather\_condition AND

sev3.weather\_condition = sev4.weather\_condition

ORDER BY sev1.weather\_condition ASC;



**Visualization 5:**

**SPICE:**

**temperature (weather\_condition = ‘bad’)-**

select distinct temp\_range\_in\_f, severity, weather\_condn, count(\*) over (partition by weather\_condn, severity, temp\_range\_in\_f) as cnt

from

(select

case

when temperature between -80 and -71 then 'aa-80 to -71'

when temperature between -70 and -61 then 'ab-70 to -61'

when temperature between -60 and -51 then 'ac-60 to -51'

when temperature between -50 and -41 then 'ad-50 to -41'

when temperature between -40 and -31 then 'ae-40 to -31'

when temperature between -30 and -21 then 'af-30 to -21'

when temperature between -20 and -11 then 'ag-20 to -11'

when temperature between -10 and -1 then 'ah-10 to -01'

when temperature between 0 and 9 then 'ai0 to 9'

when temperature between 10 and 19 then 'aj10 to 19'

when temperature between 20 and 29 then 'ak20 to 29'

when temperature between 30 and 39 then 'al30 to 39'

when temperature between 40 and 49 then 'am40 to 49'

when temperature between 50 and 59 then 'an50 to 59'

when temperature between 60 and 69 then 'ao60 to 69'

when temperature between 70 and 79 then 'ap70 to 79'

when temperature between 80 and 89 then 'aq80 to 89'

when temperature between 90 and 99 then 'ar90 to 99'

when temperature between 100 and 109 then 'as100 to 109'

when temperature between 110 and 119 then 'at110 to 119'

when temperature between 120 and 129 then 'au120 to 129'

when temperature between 130 and 139 then 'av130 to 139'

when temperature between 140 and 149 then 'aw140 to 149'

when temperature between 150 and 159 then 'ax150 to 159'

when temperature between 160 and 169 then 'ay160 to 169'

when temperature between 170 and 179 then 'az170 to 179'

else 'temperature unknown'

end as temp\_range\_in\_f,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition = 'Cloudy' then 'good'

when weather\_condition = 'Cloudy / Windy' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition = 'Fair' then 'good'

when weather\_condition = 'Fair / Windy' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition = 'Light Blowing Snow' then 'good'

when weather\_condition = 'Light Drizzle' then 'good'

when weather\_condition = 'Light Drizzle / Windy' then 'good'

when weather\_condition = 'Light Fog' then 'good'

when weather\_condition = 'Light Freezing Drizzle' then 'good'

when weather\_condition = 'Light Freezing Fog' then 'good'

when weather\_condition = 'Light Freezing Rain' then 'good'

when weather\_condition = 'Light Freezing Rain / Windy' then 'good'

when weather\_condition = 'Light Hail' then 'good'

when weather\_condition = 'Light Haze' then 'good'

when weather\_condition = 'Light Ice Pellets' then 'good'

when weather\_condition = 'Light Rain' then 'good'

when weather\_condition = 'Light Rain / Windy' then 'good'

when weather\_condition = 'Light Rain Shower' then 'good'

when weather\_condition = 'Light Rain Shower / Windy' then 'good'

when weather\_condition = 'Light Rain Showers' then 'good'

when weather\_condition = 'Light Sleet' then 'good'

when weather\_condition = 'Light Snow' then 'good'

when weather\_condition = 'Light Snow and Sleet' then 'good'

when weather\_condition = 'Light Snow Grains' then 'good'

when weather\_condition = 'Light Snow Shower' then 'good'

when weather\_condition = 'Light Snow Showers' then 'good'

when weather\_condition = 'Light Thunderstorms and Rain' then 'good'

when weather\_condition = 'Light Thunderstorms and Snow' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition = 'Mostly Cloudy' then 'good'

when weather\_condition = 'Mostly Cloudy / Windy' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition = 'Partial Fog' then 'good'

when weather\_condition = 'Partial Fog / Windy' then 'good'

when weather\_condition = 'Partly Cloudy' then 'good'

when weather\_condition = 'Partly Cloudy / Windy' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

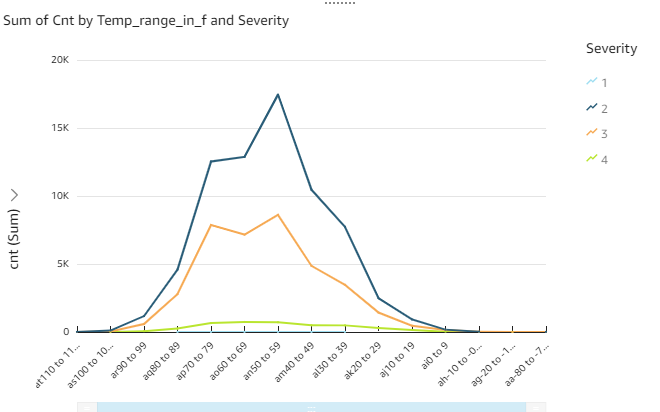
else 'bad'

end as weather\_condn, severity

from us\_accidents\_dec\_19\_database.us\_accident\_v3)

where temp\_range\_in\_f <> 'temperature unknown'

order by temp\_range\_in\_f, severity desc, weather\_condn, cnt desc;



**temperature (weather\_condition = ‘good’)-**

select distinct temp\_range\_in\_f, severity, count(\*) over (partition by weather\_condn, severity, temp\_range\_in\_f) as cnt

from

(select

case

when temperature between -80 and -71 then 'aa-80 to -71'

when temperature between -70 and -61 then 'ab-70 to -61'

when temperature between -60 and -51 then 'ac-60 to -51'

when temperature between -50 and -41 then 'ad-50 to -41'

when temperature between -40 and -31 then 'ae-40 to -31'

when temperature between -30 and -21 then 'af-30 to -21'

when temperature between -20 and -11 then 'ag-20 to -11'

when temperature between -10 and -1 then 'ah-10 to -01'

when temperature between 0 and 9 then 'ai0 to 9'

when temperature between 10 and 19 then 'aj10 to 19'

when temperature between 20 and 29 then 'ak20 to 29'

when temperature between 30 and 39 then 'al30 to 39'

when temperature between 40 and 49 then 'am40 to 49'

when temperature between 50 and 59 then 'an50 to 59'

when temperature between 60 and 69 then 'ao60 to 69'

when temperature between 70 and 79 then 'ap70 to 79'

when temperature between 80 and 89 then 'aq80 to 89'

when temperature between 90 and 99 then 'ar90 to 99'

when temperature between 100 and 109 then 'as100 to 109'

when temperature between 110 and 119 then 'at110 to 119'

when temperature between 120 and 129 then 'au120 to 129'

when temperature between 130 and 139 then 'av130 to 139'

when temperature between 140 and 149 then 'aw140 to 149'

when temperature between 150 and 159 then 'ax150 to 159'

when temperature between 160 and 169 then 'ay160 to 169'

when temperature between 170 and 179 then 'az170 to 179'

else 'temperature unknown'

end as temp\_range\_in\_f,

case

when weather\_condition = 'Clear' then 'good'

when weather\_condition = 'Cloudy' then 'good'

when weather\_condition = 'Cloudy / Windy' then 'good'

when weather\_condition = 'Drizzle' then 'good'

when weather\_condition = 'Drizzle / Windy' then 'good'

when weather\_condition = 'Fair' then 'good'

when weather\_condition = 'Fair / Windy' then 'good'

when weather\_condition = 'Haze' then 'good'

when weather\_condition = 'Light Blowing Snow' then 'good'

when weather\_condition = 'Light Drizzle' then 'good'

when weather\_condition = 'Light Drizzle / Windy' then 'good'

when weather\_condition = 'Light Fog' then 'good'

when weather\_condition = 'Light Freezing Drizzle' then 'good'

when weather\_condition = 'Light Freezing Fog' then 'good'

when weather\_condition = 'Light Freezing Rain' then 'good'

when weather\_condition = 'Light Freezing Rain / Windy' then 'good'

when weather\_condition = 'Light Hail' then 'good'

when weather\_condition = 'Light Haze' then 'good'

when weather\_condition = 'Light Ice Pellets' then 'good'

when weather\_condition = 'Light Rain' then 'good'

when weather\_condition = 'Light Rain / Windy' then 'good'

when weather\_condition = 'Light Rain Shower' then 'good'

when weather\_condition = 'Light Rain Shower / Windy' then 'good'

when weather\_condition = 'Light Rain Showers' then 'good'

when weather\_condition = 'Light Sleet' then 'good'

when weather\_condition = 'Light Snow' then 'good'

when weather\_condition = 'Light Snow and Sleet' then 'good'

when weather\_condition = 'Light Snow Grains' then 'good'

when weather\_condition = 'Light Snow Shower' then 'good'

when weather\_condition = 'Light Snow Showers' then 'good'

when weather\_condition = 'Light Thunderstorms and Rain' then 'good'

when weather\_condition = 'Light Thunderstorms and Snow' then 'good'

when weather\_condition = 'Mist' then 'good'

when weather\_condition = 'Mostly Cloudy' then 'good'

when weather\_condition = 'Mostly Cloudy / Windy' then 'good'

when weather\_condition = 'N/A Precipitation' then 'good'

when weather\_condition = 'Overcast' then 'good'

when weather\_condition = 'Partial Fog' then 'good'

when weather\_condition = 'Partial Fog / Windy' then 'good'

when weather\_condition = 'Partly Cloudy' then 'good'

when weather\_condition = 'Partly Cloudy / Windy' then 'good'

when weather\_condition = 'Patches of Fog' then 'good'

when weather\_condition = 'Scattered Clouds' then 'good'

when weather\_condition = 'Shallow Fog' then 'good'

when weather\_condition = 'Showers in the Vicinity' then 'good'

when weather\_condition = 'Small Hail' then 'good'

when weather\_condition = 'Thunder in the Vicinity' then 'good'

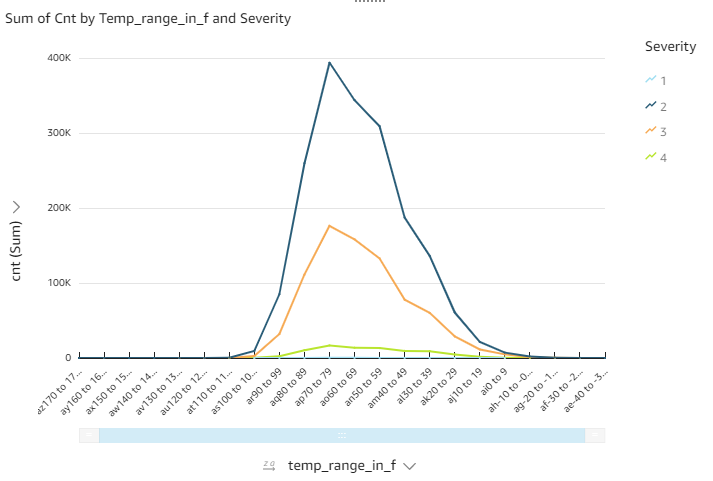
else 'bad'

end as weather\_condn, severity

from us\_accidents\_dec\_19\_database.us\_accident\_v3)

where temp\_range\_in\_f <> 'temperature unknown' and weather\_condn = 'good'

order by temp\_range\_in\_f, severity desc, cnt desc;



**Visualization 5:**

**SPICE:**

SELECT DISTINCT hour(start\_time) AS hour,

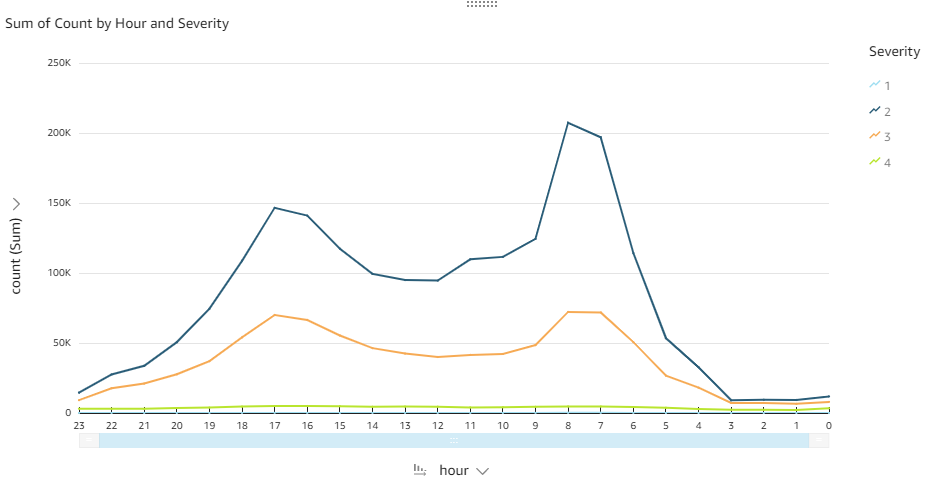
severity,

count(\*)

OVER (partition by severity, hour(start\_time)) AS count

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

ORDER BY hour(start\_time) desc, severity desc;



**Visualization 6:**

**SPICE (Count):**

SELECT severity,

avg(count) AS avg,

max(count) AS max,

stddev(count) AS stddev

FROM

(SELECT DISTINCT date(start\_time) AS date,

severity,

count(\*)

OVER (partition by severity, date(start\_time)) AS count

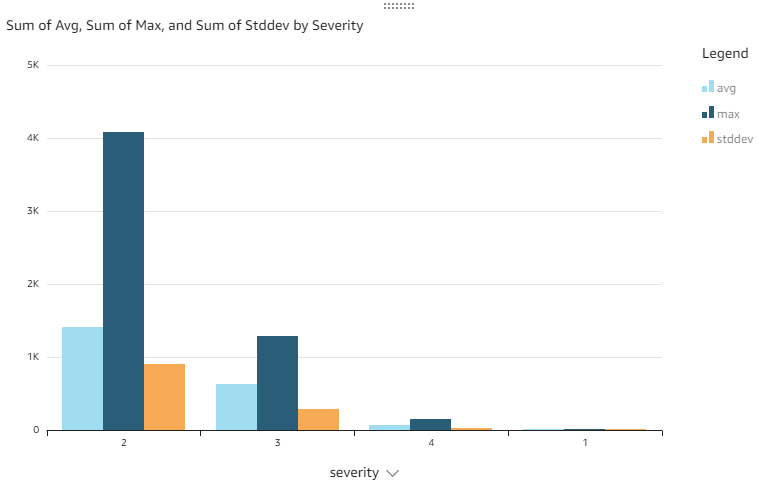
FROM us\_accidents\_dec\_19\_database.us\_accident\_v3

ORDER BY date(start\_time) desc, severity desc)

GROUP BY severity

ORDER BY severity;

## **Comment:** Not exactly. You should do: group by year(start\_time), week(start\_time), severity as on x axis we want to show year, week, severity combo.



**SPICE (Distance):**

SELECT DISTINCT severity,

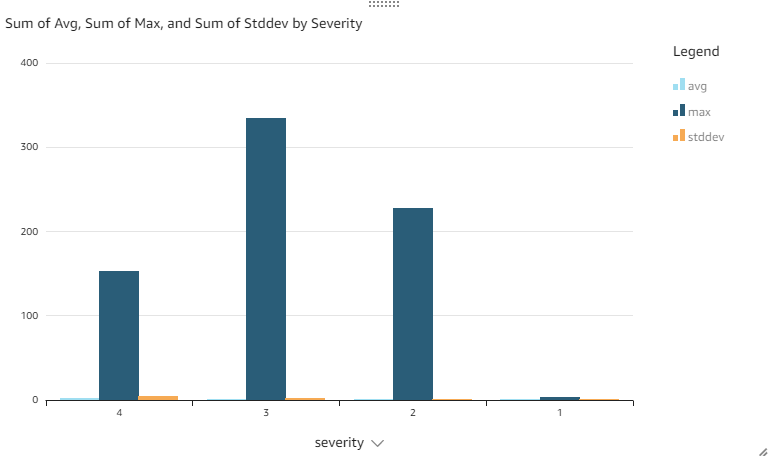
avg(distance)

OVER (partition by severity) as avg, max(distance)

OVER (partition by severity) as max, stddev(distance)

OVER (partition by severity) as stddev

FROM us\_accidents\_dec\_19\_database.us\_accident\_v3;



**Visualization 8:**

**Visualization 9:**

**Visualization 10:**

**SPICE:**

select Concept, sum(Frequency) as Freq from (with cte1 AS

(SELECT b.slno,

b.pattrn,

b.concpt,

cardinality(regexp\_extract\_all(lower(a.n\_description),

lower(b.pattrn))) count

FROM base\_corpus3 a

CROSS JOIN concept7 b

ORDER BY slno)

SELECT DISTINCT slno AS Serial\_No,

concpt AS Concept,

pattrn AS Pattern,

sum(count)

OVER (partition by pattrn) AS Frequency

FROM cte1

ORDER BY slno) group by Concept;



**Visualization 13:**