**US Accidents Visualizations**

**QUERY 1:**

with cte1 as

(select state , severity, count(severity) as severity\_1 from us\_accidents\_dec\_19\_database.us\_accident\_v3 where severity = 1 group by severity, state),

cte2 as

(select state , severity, count(severity) as severity\_2 from us\_accidents\_dec\_19\_database.us\_accident\_v3 where severity = 2 group by severity, state),

cte3 as

(select state, severity, count(severity) as severity\_3 from us\_accidents\_dec\_19\_database.us\_accident\_v3 where severity = 3 group by severity, state),

cte4 as

(select state, severity, count(severity) as severity\_4 from us\_accidents\_dec\_19\_database.us\_accident\_v3 where severity = 4 group by severity, state)

select a.state, severity\_1, severity\_2, severity\_3, severity\_4, sum(severity\_1 + severity\_2 + severity\_3 + severity\_4) as tot\_accidents from cte1 a

join cte2 b

on a.state = b.state

join cte3 c

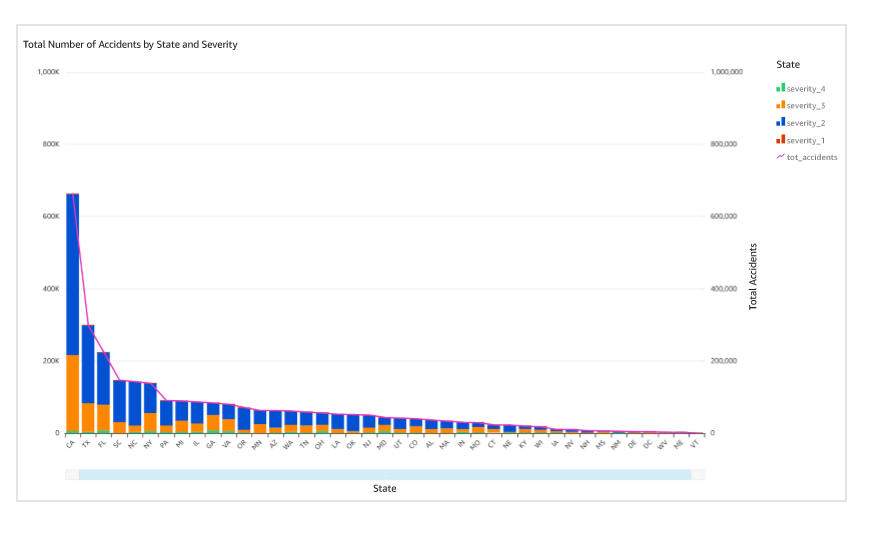
on a.state = c.state

join cte4 d

on a.state = d.state

group by a.state, severity\_1, severity\_2, severity\_3, severity\_4

order by tot\_accidents desc



**QUERY 3:**

with cte1 as

(select date\_diff('minute', start\_time, end\_time) as duration from awsdatacatalog.us\_accidents\_dec\_19\_database.us\_accident\_v3),

cte2 as

(select

case

when duration <= 59 then 'minutes'

when duration between 60 and 239 then 'hours'

when duration between 240 and 959 then '4hours\_16hours'

when duration between 960 and 1439 then '16hours\_24hours'

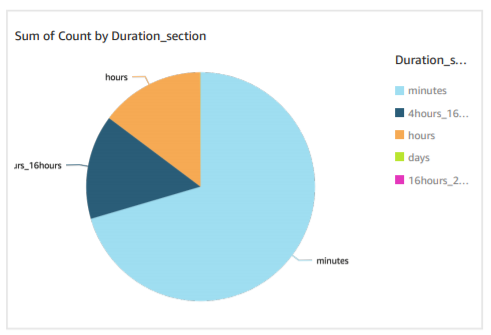
when duration >=1440 then 'days'

end as duration\_section

from cte1)

select duration\_section, count(duration\_section) as count from cte2

group by duration\_section



**QUERY 4:**

with cte1 as(

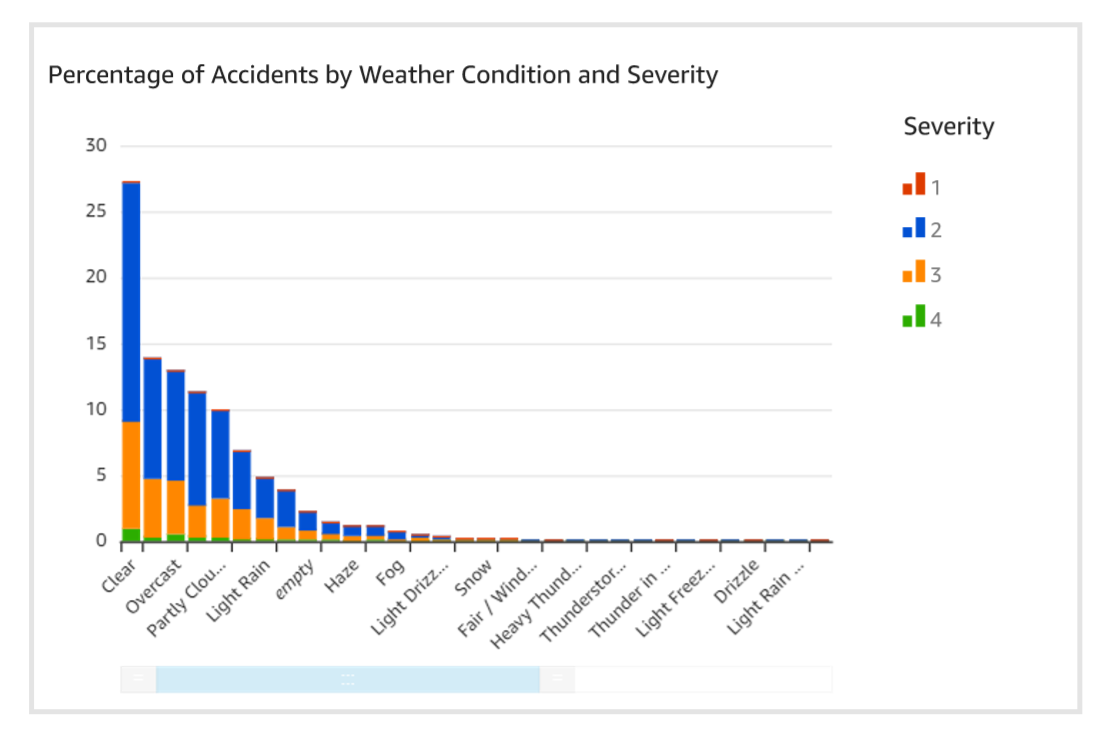
select weather\_condition, severity, count(\*) as weather\_accidents from us\_accidents\_dec\_19\_database.us\_accident\_v2

group by weather\_condition, severity

order by weather\_condition)

select weather\_condition, severity, weather\_accidents, total\_accidents, cast(weather\_accidents as double)\*100/total\_accidents as perc\_severity\_accidents from cte1

cross join (select count(\*) as total\_accidents from us\_accidents\_dec\_19\_database.us\_accident\_v3) b



**QUERY 5:**

with cte1 as

(select date\_format(start\_time, '%Y,%v') AS date,severity,weather\_condition,

case

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

when weather\_condition like 'Cloudy%' then 'good'

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

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

when weather\_condition like 'Fair%' then 'good'

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

when weather\_condition like 'Light%' then 'good'

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

when weather\_condition like 'Mostly%' then 'good'

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

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

when weather\_condition like 'Partial%' then 'good'

when weather\_condition like 'Partly%' 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\_type,

case

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

else 'temperature unknown'

end as temp\_range\_in\_f

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

cte2 as

(select \*,

case

when severity = 1 and weather\_type = 'good' then 'sev1\_good'

when severity = 2 and weather\_type = 'good' then 'sev2\_good'

when severity = 3 and weather\_type = 'good' then 'sev3\_good'

when severity = 4 and weather\_type = 'good' then 'sev4\_good'

when severity = 1 and weather\_type = 'bad' then 'sev1\_bad'

when severity = 2 and weather\_type = 'bad' then 'sev2\_bad'

when severity = 3 and weather\_type = 'bad' then 'sev3\_bad'

when severity = 4 and weather\_type = 'bad' then 'sev4\_bad'

else 'unknown\_weather' end as good\_bad\_sev

from cte1),

cte3 as

(select date,temp\_range\_in\_f,good\_bad\_sev,count(\*) as number\_of\_accident from cte2 group by date,temp\_range\_in\_f,good\_bad\_sev

order by date,temp\_range\_in\_f,good\_bad\_sev asc)

select date,temp\_range\_in\_f, kv1['sev1\_good'] as Sev\_1\_good\_weather,

kv2['sev2\_good'] as Sev\_2\_good\_weather,

kv3['sev3\_good'] as Sev\_3\_good\_weather,

kv4['sev4\_good'] as Sev\_4\_good\_weather,

kv5['sev1\_bad'] as Sev\_1\_bad\_weather,

kv6['sev2\_bad'] as Sev\_2\_bad\_weather,

kv7['sev3\_bad'] as Sev\_3\_bad\_weather,

kv8['sev4\_bad'] as Sev\_4\_bad\_weather

from (

select date,temp\_range\_in\_f, map\_agg(good\_bad\_sev, number\_of\_accident) kv1,

map\_agg(good\_bad\_sev, number\_of\_accident) kv2,

map\_agg(good\_bad\_sev, number\_of\_accident) kv3,

map\_agg(good\_bad\_sev, number\_of\_accident) kv4,

map\_agg(good\_bad\_sev, number\_of\_accident) kv5,

map\_agg(good\_bad\_sev, number\_of\_accident) kv6,

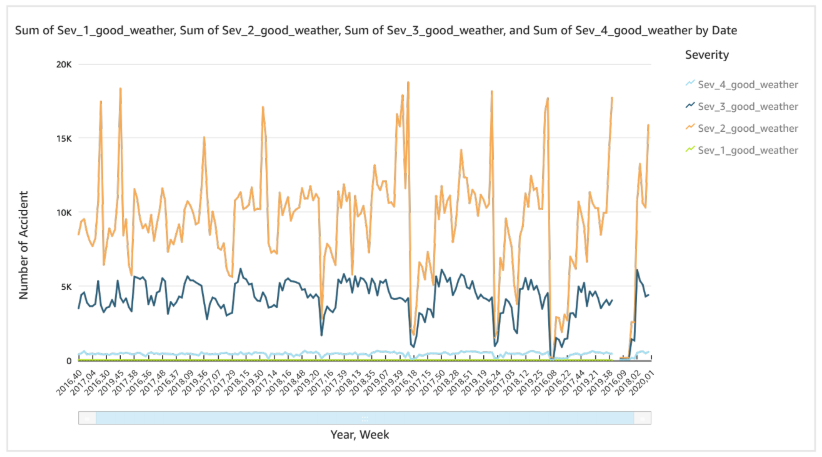
map\_agg(good\_bad\_sev, number\_of\_accident) kv7,

map\_agg(good\_bad\_sev, number\_of\_accident) kv8

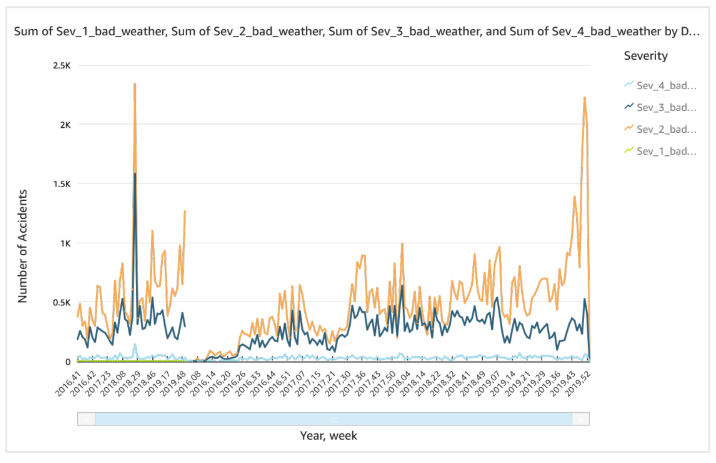
from cte3

group by date,temp\_range\_in\_f)

**Good Weather Analysis:**

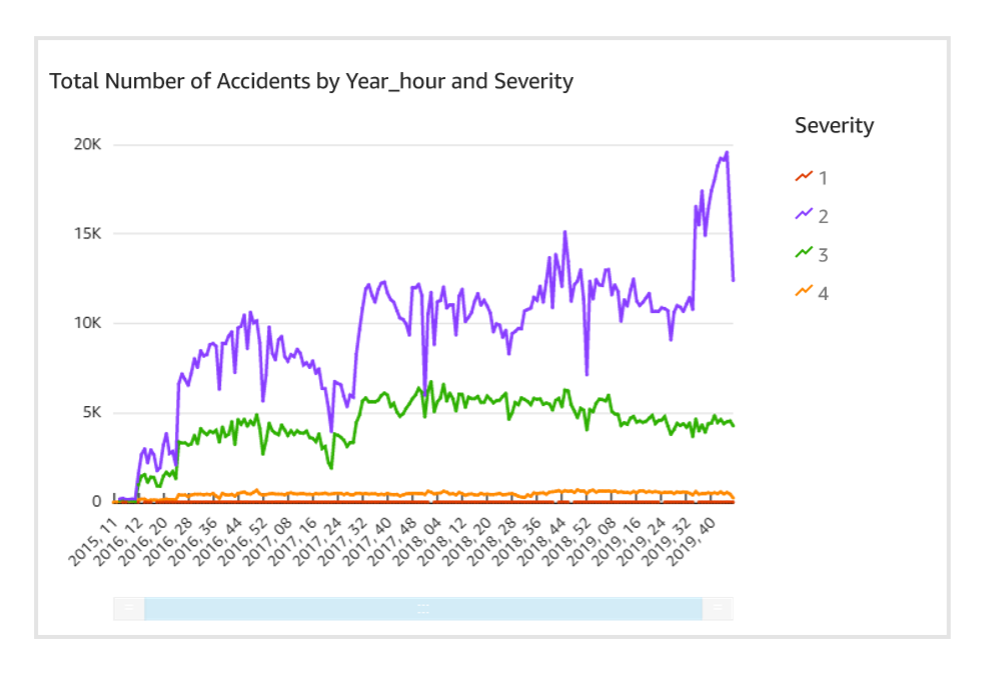


**Bad Weather Analysis:**



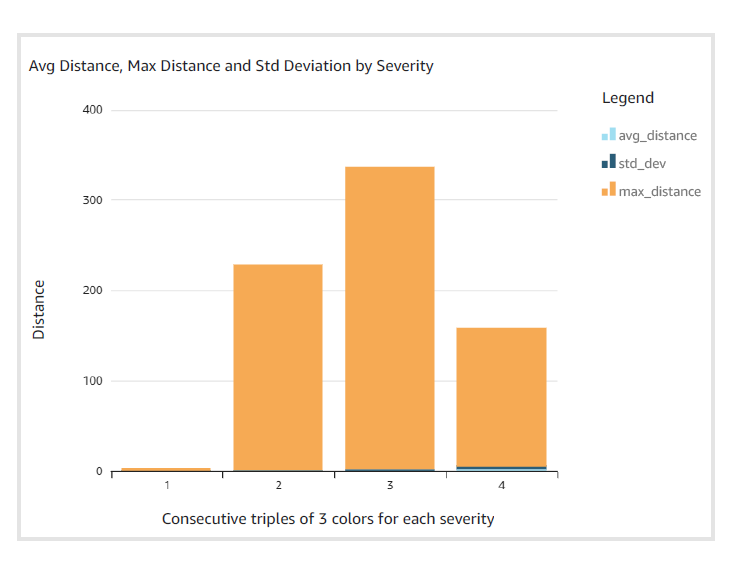
**QUERY 6:**

select date\_format(start\_time, '%Y, %v') as Year\_Hour , severity, count(\*) as Total\_accidents from us\_accidents\_dec\_19\_database.us\_accident\_v3 group by start\_time, severity



**QUERY 7:**

select severity, avg(distance) as avg\_distance, max(distance) as max\_distance, round(stddev(distance),2) as std\_dev from us\_accidents\_dec\_19\_database.us\_accident\_v3 group by severity order by severity asc



**Comments:** Not right. Could we not show the triples as three separate verticals like histograms?

**QUERY 10:**

with cte1 as

(SELECT row, slno,pattrn, concpt, n\_description,

regexp\_extract\_all(lower(a.n\_description), (b.pattrn)) AS regexp\_group1,

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

b.pattrn)) car

FROM AwsDataCatalog.us\_accidents\_dec\_19\_database.base\_corpus3 a

CROSS JOIN AwsDataCatalog.us\_accidents\_dec\_19\_database.concept7 b

order by slno)

select distinct(concpt),

case

when slno >=20 and slno <25 then 20

when slno >=25 and slno <30 then 25

when slno >=30 and slno <35 then 30

when slno >=35 and slno <40 then 35

when slno >=40 and slno <45 then 40

when slno >=45 and slno <50 then 45

when slno >=50 and slno <55 then 50

when slno >=55 and slno <60 then 55

when slno >=60 and slno <65 then 60

when slno >=65 and slno <70 then 65

when slno >=70 and slno <75 then 70

when slno >=75 and slno <80 then 75

when slno >=80 and slno <85 then 80

when slno >=85 and slno <90 then 85

when slno >=90 and slno <95 then 90

when slno >=95 and slno <100 then 95

when slno >=100 and slno <105 then 100

when slno >=105 and slno <110 then 105

when slno >=110 and slno <115 then 110

when slno >=115 and slno <120 then 115

when slno >=120 and slno <125 then 120

when slno >=125 and slno <130 then 125

when slno >=130 and slno <135 then 130

when slno >=135 and slno <140 then 135

when slno >=140 and slno <145 then 140

when slno >=145 and slno <150 then 145

when slno >=150 and slno <155 then 150

when slno >=155 and slno <160 then 155

when slno >=160 and slno <300 then 160

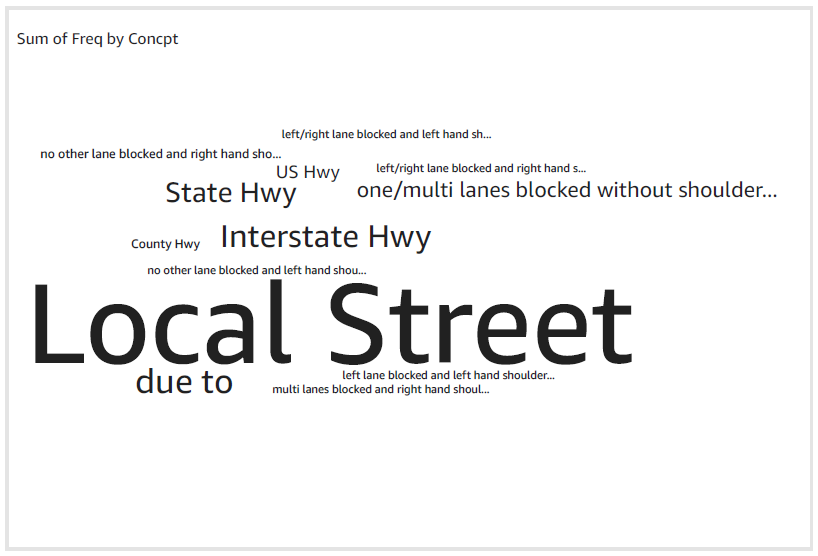
when slno = 300 then 300

end as slno,

sum(car) over (partition by concpt) as freq

from cte1 where car>0

order by slno



**QUERY 13:**

Top 5 states:

1. **CA**

SELECT DISTINCT state,

hour(start\_time) AS hour,

precipitation,

count(\*)

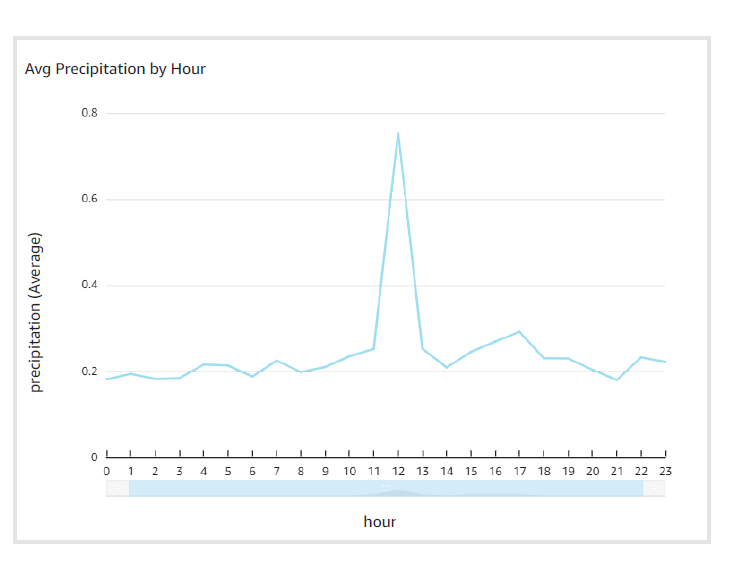
OVER (partition by precipitation, hour(start\_time), state) AS no\_of\_accidents

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

WHERE precipitation IS NOT NULL and state = 'CA'

ORDER BY state, hour desc, precipitation desc, no\_of\_accidents desc

**Comments:** Not right. We did not ask for precipitation by hour. We asked for total precipitation within state over date. If it is too much of data, show only last 6 months (July 2019 – Dec 2019). We could use graph plot.



1. **TX**

SELECT DISTINCT state,

hour(start\_time) AS hour,

precipitation,

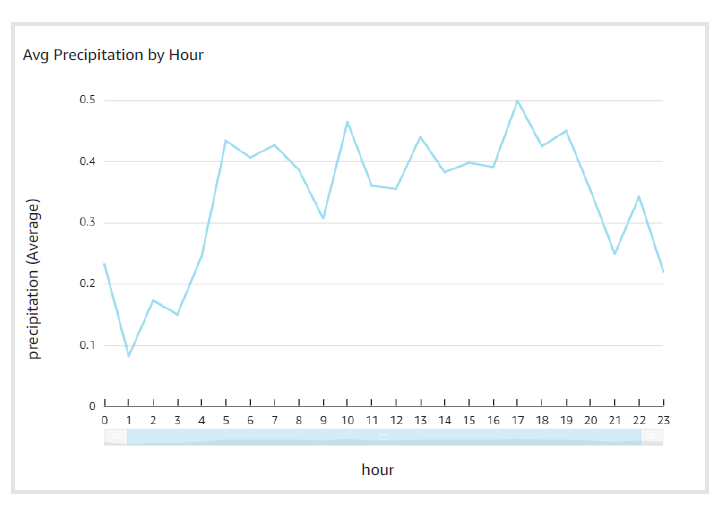
count(\*)

OVER (partition by precipitation, hour(start\_time), state) AS no\_of\_accidents

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

WHERE precipitation IS NOT NULL and state = 'TX'

ORDER BY state, hour desc, precipitation desc, no\_of\_accidents desc



1. **FL**

SELECT DISTINCT state,

hour(start\_time) AS hour,

precipitation,

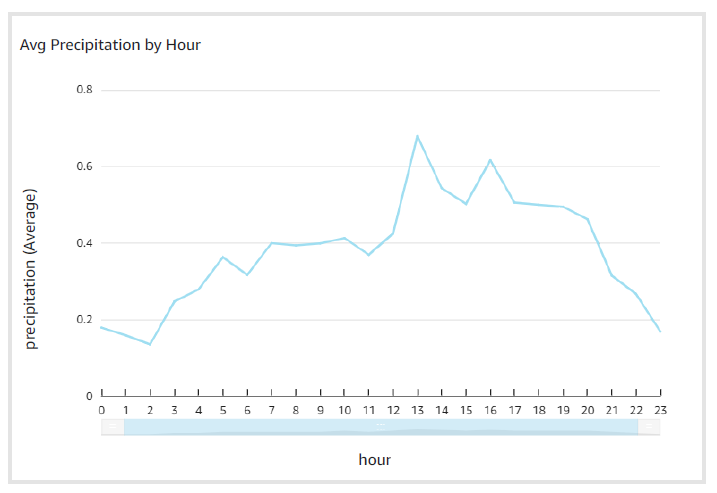
count(\*)

OVER (partition by precipitation, hour(start\_time), state) AS no\_of\_accidents

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

WHERE precipitation IS NOT NULL and state = 'FL'

ORDER BY state, hour desc, precipitation desc, no\_of\_accidents desc



1. **SC**

SELECT DISTINCT state,

hour(start\_time) AS hour,

precipitation,

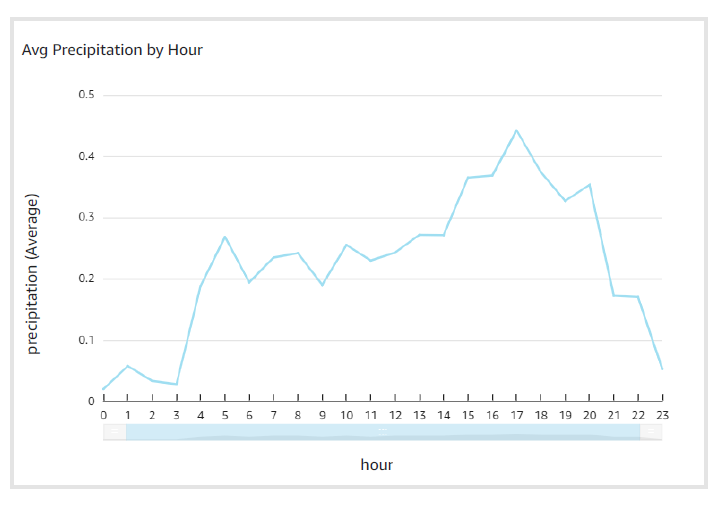
count(\*)

OVER (partition by precipitation, hour(start\_time), state) AS no\_of\_accidents

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

WHERE precipitation IS NOT NULL and state = 'SC'

ORDER BY state, hour desc, precipitation desc, no\_of\_accidents desc



1. **NY**

SELECT DISTINCT state,

hour(start\_time) AS hour,

precipitation,

count(\*)

OVER (partition by precipitation, hour(start\_time), state) AS no\_of\_accidents

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

WHERE precipitation IS NOT NULL and state = 'NY'

ORDER BY state, hour desc, precipitation desc, no\_of\_accidents desc

