**Data Analytics broad plan**

1. US ACCIDENTS dataset is single dataset.
2. **Introduction:** This public dataset has car accidents (in USA) data since Feb, 2016 and going till Dec 2019 and covers most of the officially reported accidents in 49 states of USA (except Alaska and Hawaii but including Washington DC) categorized by severity level. This data is non-personal data used for community. This data is collected automatically through API calls to MapQuest and Bing database systems and merging them together in one file. This single dataset has approx. 3 million rows of car accident records but it keeps growing. It has 49 different columns all of which are scalar data including accident description in plain English etc. It begins with unique id (called ID) for each row, source of data which may be ‘MapQuest’ or ‘Bing’ or ‘MapQuest-Bing’ meaning both. It has Traffic Message Control code (TMC) type with which type of this accident can be tracked in various systems in the nation. The structure of TMC is generally complex implying the section of road, direction of traffic, zone and unique identifier together. But in our case, this appears to be like a category of accident. There is a severity column that encodes levels of severity in the accident which starts from 0 and goes up to 4 (meaning high severity). There are many columns like the address of accident site, start\_location (Start\_longitude, Start\_lattitude), end\_location (longitude, latitude) (in case it spread far), incident time (start\_time, end\_time) (date & time), distance covered within zone of accident (in case it spread far), its’ impact on traffic flow, the lane side (right or left) where accident landed, address, daytime/nighttime, weather condition, amenities, infra-structure traffic constraints (like traffic light or intersection or round-about or traffic calmer etc.).
3. The web-link is : <https://www.kaggle.com/sobhanmoosavi/us-accidents>

In order to download, create a Kaggle account and download the dataset into S3 bucket.

1. In Athena, check if we can query the data in bucket. Also check if you can use Quicksight.
2. The detailed description of the 49 columns of data can be found in either the web-site above or another one as <https://smoosavi.org/datasets/us_accidents> . Please learn them.
3. **Broad steps of data processing:**
4. **Initial check:** We need to check some very basic things like: State has only 49 distinct non-null values; Country is non-null and ‘US’. Start\_Time of accident occurrence is not null and a valid datetime. But Start\_Lattitude is non-null and > 0 and Start\_Longitude is non-null and < 0. If there is any data in violation, delete those rows. If we have severity other than 0-4, we can delete the data. You should delete all data with severity 0.



1. **Clean-up of data:** If there are Timezones (for incident-site; start) with more than 4 distinct values (Eastern, Central, Mountain, Pacific), then update them to the standard 4 values. If Start\_Time is not a valid datetime then delete those data. If timezone has empty space, it may be okay for the time being (but we will ignore the empty data when we use timezone data for display). It is possible to correct timezone data but we will move on without correcting it (for lack of time). There are some cases where zipcode is empty string which ideally should be corrected but we will skip that effort also and make sure to exclude that data when we analyze zipcode data. If Side is space, we will leave it in our table but we will not include that data when we analyze side. Distance, Temperature, Wind-Chill, Humidity, Pressure, wind\_Speed, Precipitation should be >= 0 or null. Amenity, Bump, Crossing, Give\_way, Junction, No\_Exit, Railway, RoundAbout, Station, Stop, Traffic\_Calming, Traffic\_Signal, Turning\_Loop should be TRUE or FALSE or null. Change other values to null. Sunrise\_Sunset, Civil\_Twilight, Nautical\_Twilight, Astronomical\_Twilight should have allowed values (‘Night’, ‘Day’, empty string). Change other values to null.
2. **Queries:** This table is very simple with simple columns. Only complex columns can be Description and Weather (and may be a few more). We will have simple queries. But we will pay attention to picking majority. For example, what is the majority severity in state of Texas? Like this. In addition, we may need to calculate various co-relation coefficients and save on tables. We will have about 10-11 such problems.
3. **Data visualization:** Later we will move onto presenting result data in colored visual forms using graphs and charts for each of the saved result tables. Various visual forms will be used like histogram graphs, plots, pie-charts and if possible colored matrix squares. We will try to use AWS QuickSight tool in AWS for this purpose.

That was just a broad plan where our objective is to draw the diagrams. Now we enumerate each problem. We can look for answers to the following questions.

1. We want to see number of accidents (broken further by severity levels) within each state. The result table should look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State | Accidents\_sev\_4 | Accidents\_sev\_3 | Accidents\_sev\_3 | Accidents\_sev\_1 |
| AL | 54 | 64 | 21 | 34 |
| AR |  |  |  |  |
| . |  |  |  |  |
| CA |  |  |  |  |
| DC |  |  |  |  |
| . |  |  |  |  |
| . |  |  |  |  |
| NY |  |  |  |  |
| . |  |  |  |  |
| TX |  |  |  |  |
| . |  |  |  |  |
| VA |  |  |  |  |
| . |  |  |  |  |
| WV |  |  |  |  |

1. To investigate further, may we ask for all accidents plotted within each state border to provide us perception of area vs accidents.
2. What period of time do these accidents (duration) (nationwide) affect the traffic on average? Can we produce a pie chart showing each severity and their average duration? We also want to know max duration and standard deviation on each severity. We want a small result table like:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Avg\_dur\_sev\_4 | Stddev\_dur\_sev\_4 | Max\_dur\_sev\_4 | Avg\_dur\_sev\_3 | Stddev \_dur\_sev\_3 | Max\_dur\_sev\_3 | Avg\_dur\_sev\_2 | Stddev\_dur\_sev\_2 | Max\_dur\_sev\_2 | Avg\_dur\_sev\_1 | Stddev\_dur\_sev\_1 | Max\_dur\_sev\_1 |
| 110 secs | 4 secs | 2330 secs | 108 | 5 | 2100 | 100 | 4 | 1800 | 92 | 3 | 1600 |

1. There are about 100+ distinct weather conditions (like clear, overcast, mostly cloudy, partly cloudy, light rain, scattered rain, heavy rain etc.). We would like to plot number of accidents for each weather condition. But it does not reveal much information showing most of the accidents happen in clear weather or overcast or mostly cloudy or partly cloudy weather (as opposed to hailstorms or , thunderstorms or snowfalls etc.). So, we ask the next further question. Can we see the percentage break-down of severity of accidents within each weather condition? Do hailstorms or other adverse weather leave any effect on severe accidents etc.? Take weather conditions as ‘Heavy Snow / Windy’, ‘Heavy Snow with Thunder’, ‘Heavy Rain Showers’, ‘Widespread Dust’, ‘Heavy Snow with Thunder’, ‘Freezing Rain’, ‘Snow / Windy’, ‘Dust Whirls, ‘Heavy T-Storm / Windy’, ‘Heavy Sleet’, ‘Heavy Freezing Rain’, ‘Heavy Freezing Rain’, ‘Volcanic Ash’, ‘Heavy T-Storm’, ‘Heavy Smoke’, ‘Heavy Blowing Snow’, ‘Sand / Dust Whirlwinds’, ‘Blowing Dust / Windy’, ‘Heavy Thunderstorms with Small Hail’, ‘Fog / Windy’, ‘Sand / Dust Whirlwinds / Windy’, ‘Sand / Dust Whirlwinds / Windy’, ‘Heavy Ice Pellets’, ‘Squalls / Windy’ etc. Do this analysis. Derive the percentage of severity 4 accidents (over all severity accidents) for bad weather condition and the percentage of severity 4 accidents (over all severity accidents) for good (non-bad) weather condition for each day. **Comments: total severity perc for bad (and also good) weather should be 100.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Sev\_4\_perc\_bad\_weathr | Sev\_3\_perc\_bad\_weathr | Sev\_2\_perc\_bad\_weathr | Sev\_1\_perc\_bad\_weathr | Sev\_4\_perc\_good\_weathr | Sev\_4\_perc\_good\_weathr | Sev\_4\_perc\_good\_weathr | Sev\_4\_perc\_good\_weathr |
| 03/04/2019 | 60 | 30 | 05 | 05 | 50 | 30 | 10 | 10 |
| 03/05/2019 | 61 | 29 | 06 | 04 | 51 | 29 | 10 | 10 |
| 03/06/2019 | 63 | 27 | 07 | 03 | 53 | 27 | 10 | 10 |

1. Relations with parameters : We want to observe in bad weather conditions (meaning leaving out fair weather conditions like fair, clear day, overcast, mostly cloudy, partly cloudy, light rain, scattered rain etc.), if there is any relation between some natural parameters like temperature, wind-chill, humidity, pressure, visibility, wind-speed, precipitation, infra-structure parameters like Bump, Crossing, Give\_Way, Junction, No\_Exit, Railway, Roundabout, Station, Stop, Traffic\_Calming, Traffic\_Signal, Turning\_Loop and accidents at each severity. We ask this question? Within bad weather conditions (non-good weather conditions), is there any relation between any of the 7 infra-structure parameters as crossing, give\_way, junction, no\_exit, roundabout, stop, traffic\_signal and number of accidents (for each severity) or 7 natural parameters (like temperature, wind-chill, humidity, pressure, visibility, wind-speed, precipitation) and number of accidents (for each severity)? That means for each day, we want to plot curves for each of the 7 + 7 parameters vs number of accidents (without it and with it) at each severity. Show 7+ 7 frames each containing 4 graphs for 4 severities. We may extend this problem by adding another constraint like odd time (outside normal working hour period) in future. Let us take temperature ranges like ( (-10 - +1), (2 - 13), (14 - 25), (26 - 37), (38 - 49), (50 - 61), (62 - 73), (74 - 85), (86 - 97), (98 - 109). Let us produce a result table which looks like below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| date | Temperature (F)  (range) | Bad\_  wthr\_  sev4\_  acc | Good\_  wthr\_  sev4\_  acc (non-bad) | Bad\_  wthr\_  sev3\_  acc | Good\_  wthr\_  sev3\_  acc | Bad\_  wthr\_  sev2\_  acc | Good\_  wthr\_  sev2\_  acc | Bad\_  wthr\_  sev1\_  acc | Good\_  wthr\_  sev1\_  acc |
| 03/03/2019 | -10 - +1 | 100 | 110 | 26 | 30 | 16 | 12 | 13 | 11 |
| 03/03/2019 | +2 - +13 |  |  |  |  |  |  |  |  |

In the above figure, we mean to have rows for all existing ranges and then go for the next date. This way, we need to build all 14 result tables for all 14 parameters. Wind-chill should have ranges like (-70 - -61), (-60 - -51), (-50 - -41), (-40 - -31), (-30 - -21), (-20 - -11), (-10 - -1), (0 - 9), (10 - 19), (20 - 29), (30 - 39), (40 - 49), (50-59), (60 - 69), (70 - 79), (80 - 89), (90 - 99), (100-109), (110 - 119), (120-129), (130 -139), (140-149), (150 - 159), (160 -169) etc.. Humidity (%) ranges may be (1-5), (6-10), (11-15), (16-20), (21-25), (26 - 30), (31-35), ….., (91 – 95), (95 – 100). Pressure ranges may be like (0 -3), (4 - 7), (8 - 11), (12 - 15), (16 - 19), (20 - 23), (24 - 27), (28 - 31), (32 - 35). Visibility(mi) ranges may be (0 – 0.06), (0.06001 – 0.1), (0.10001 (means greater than 0.1) – 0.12), (0.12001 (means greater than 0.12) – 0.19), (0.19001 (means greater than 0.19) – 0.25), (0.25001 (means greater than 0.25) – 0.30), (0.30001 (means greater than 0.3) – 0.38), (0.38001 (means greater than 0.38) – 0.4), (0.40001 (means greater than 0.4) – 0.5), (0.50001 (means greater than 0.5) – 0.6), (0.60001 (means greater than 0.6) – 0.7), (0.70001 (means greater than 0.7) – 0.8), (0.80001 (means greater than 0.8) – 0.9), (0.90001 (means greater than 0.9) – 1.0), (1.00001 (means greater than 1.0) – 1.2), (1.20001 (means greater than 1.2) – 1.4), (1.40001 (means greater than 1.4) – 10), (10.0001 (means greater than 10) – 50), (50.00001 (means greater than 50) – 150). For wind\_speed, take 44 ranges like (0 – 1), (1.00001-2), (2.00001-3), (3.00001-4), (4.00001-5), ……. (41.00001 – 42), (42.00001 – 43) and above 43.00001. For precipitation, keep 25 ranges like (0 – 1), (1.00001-2), (2.00001-3), (3.00001-4), (4.00001-5), ……. (24.00001 – 25). For each of the seven infra-structure params, it is Boolean value true or false. Due to lack of time, we are simplifying it to only 6 sets each set with 2 graphs (bad weather, good weather) for 3 physical params like temperature (10 ranges -10 - +110), visibility (in miles) 0 – 2+ (10 ranges), wind\_speed (0 – 100+; 10 ranges) and 3 infra-structure params like (traffic signal, crossing and junction).

1. For each severity, how number of accidents have been changing over times from past into current. Display the plots for each year, week (or month) starting 2016, Feb 8th until current time.
2. May we know the average distance of traffic affected by accidents of each severity? We want to know the max and standard deviation (up to 2 decimal) also. This will result in 4 rows (1 for each severity) of 3 columns (avg, max and std-dev). To calculate standard deviation, you may use stddev(“distance(mi)”) function.
3. For each severity, on which side of the do the accidents happen or land up? Is it right side or left side? We may have a pie chart for each severity.
4. Correlation: This is extension of case 5. If we found any parameters that were related to accidents in case 5, let us take them further. Now we ask: do any of these 2 parameters have joint contribution on accidents? That is like asking, if one param x is contributing to accidents (of severity i) and another param y is contributing to accidents (of same severity i) then do they jointly contribute or do they weaken (means cancel) each other etc.? Can we calculate the pair-wise correlation coefficients for a few of these numeric columns? We should present the singular and the joint analysis results visually. If things look okay, we can extend this problem to triples also.
5. Concepts in Description: The description column contains a free-flowing text description on the accident. Can we find any small phrase concepts in that field which repeat in many accidents (meaning they are statistically precious)? Do we need to put some stop words to avoid noise or to set direction in our search (otherwise stop words will flood our analysis)? We may use Athena to build the query to build a table of concepts and frequency. There are at least two approaches to this problem as follows.
   1. First of all, we merge all the description texts into a single corpus and also make a copy. Then we remove some preliminary stopwords like all punctuations (, : . ? ! : {}[]() ‘ “ ` ~ etc.), prepositions (after, around, at, before, below, behind, beside, from, in, inside, into, of, out, outside, over, till, to, under, unless, until, while, with, within), determiners (the, a, an), conjunctions ( like, and, or, but, then etc.), pronouns (so, which, when, how, where, who, why), numbers between alphanumeric string and some tiny phrases (like “group of”, “bunch of” , “due to”, “of the” , “belongs to” , “pertains to”, “belong”, “pertain”, “follow”, “attributed to”, “qualified by”, “qualifying” etc.) from corpus. We may use word stemming to convert variants of action verbs (like “has been going” into “go” or “goes” to “go” or “will go” to “go” or “went” to “go” or “moved” to “move” etc.). In word stemming, we look at the action verbs and decide what they should be converted to. Then we build a dictionary manually. Then we replace all variants of actions verbs using the dictionary or regular expression (like “is x-ing” or “are x-ing” to become x) in the corpus. We remove all proper nouns for names of person, place, highways, roads (list them and remove them). At this point we are left with key terms. First, we have to generate the list of atomic (tiniest) phrases and then merge some of them using another table (or dictionary) called synonyms. Synonyms carry 2 columns of synonymous words or phrases (like what should be converted to what). Using this table, our list will coalesce into distinct phrases. We then do final filtering applying final stop-words like {“accident”, “hand”, “cop”, “police”, “hospital’, “flare”} which are the words we are not interested in. We may need to redo these steps a few times trying to come up with a modest sized fair list of atomic concept phrases. **Important note:** We are trying a few basic things of text mining to extract conceptual terms (not doing full-scale NLP). We will not deal with character but with words which may be called tokens. For that we need to use regular expression function processing of Athena SQL.

**Steps:**

1. Merge and create a corpus**. Query:** select array\_join(array\_agg(description),’ ‘, ‘’) as

n\_description from us\_accidents\_v4 where …;

b. Replace punctuations (,;.?\_()![]{}%&\*$#@’”`~) with space. Replace white space with

space. **. Query:** select regexp\_replace(description, ‘\p{Punct}’,’ ‘) …

1. Raw filtering (use raw stop-words). **Query:** select regexp\_replace(description, …

d. This data is semi-structured data made of phrases followed by period. This phrases mostly have concept phrases: ‘left lane blocked and right hand shoulder blocked’, ‘right lane blocked and right hand shoulder blocked’, ‘% lanes blocked and right hand shoulder blocked’, ‘left lane blocked and left hand shoulder blocked’, ‘right lane blocked and left hand shoulder blocked’, ‘% lanes blocked and left hand shoulder blocked’, ‘% lanes blocked’. Build another concept table with required important concepts. We need to pull out the pattern from this table and match it on corpus and count the number of matches and increase frequency in this concept table. Sometimes, we may need to delete the matched strings from corpus and sometimes not. Then we move onto next row in concept table and repeat this process. At the end of this iterative process, concept table has all frequencies counted. This will be our result table. Use lower(corpus) than corpus in queries.

e. Watch & Replace synonyms.

f. Make reasonable replacements in names of road. For example, highway name I 287

should be replaced by Interstate; Cherry Ave should be replaced by Avenue; McCulaugh

Blvd should be replaced by boulevard; All street names should be replaced by generic

street/interstate/route etc.

1. Remove final stop-words.
   1. Because the description is semi-structured text, we can find some alternative sub-structures within this text. We can start with atomic concepts like blocks on roads. Sometimes it may be “left lane blocked and right shoulder blocked” or “multi lanes blocked and left shoulder blocked’ and various alternative such conditions. Then we can search other concepts and build higher concepts using atomic concepts. The steps are as below.
2. Because we are trying a proof of concepts only, we can extract latest 800 rows of description data and create a new table like post\_corpus3. Help query:

Create table post\_corpus3 as

(select ROW\_NUMBER() OVER (order by start\_time desc) AS Row, description as n\_description

from us\_accident\_v3 where year(start\_time) <= 2019 limit 800

)

1. We build a viable concept table (say concept7) that has a unique serial no, each concept (concpt text), a Java standard regular expression (called pattrn) that can be used for matching & extracting the concept from description data. The serial numbers will jump up for each concept and extra representation will have serial no going up by 1. We can have concepts like ‘Interstate Hwy’, ‘National Hwy’ , ‘Intersection’, ‘Junction’ etc. Here is a query to create some concepts:

CREATE EXTERNAL TABLE `concept7`( `slno` bigint COMMENT '', `concpt` varchar(460) COMMENT '', `pattrn` varchar(615) COMMENT '', `freqncy` int COMMENT '', `embdable` boolean COMMENT '', `exclsv` boolean COMMENT '', `post\_action` varchar(300) COMMENT '') ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat' LOCATION 's3://data-set-2-us-accidents/output/Unsaved/2020/04/03/tables/9e040022-a413-4440-8f0a-bcdfbe3cd2f8' TBLPROPERTIES ( 'has\_encrypted\_data'='false');

1. We will populate the concept7 table using queries below:

insert into concept7 SELECT

slno,

-- over (partition by variants, stem order by 1) as Slno,

concpt, pattrn, freqncy,embdable,exclsv,post\_action

FROM (

VALUES

(300,'due to','\s\*(due)\s+to\s+',0,false,true,

'replace with empty string before counting'),

(20,'left lane blocked and right hand shoulder blocked','left\s+lane\s+blocked\s+.\*\b+right\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(21,'left lane blocked and right hand shoulder blocked','\bright\s+hand\s+shoulder\s+blocked\s+.\*\bleft\s+lane\s+blocked',0,false,true,

'replace with empty string after counting'),

(25,'left/right lane blocked and right hand shoulder blocked','^lane\s+blocked\s+and\s+right\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(30,'right lane blocked and right hand shoulder blocked','right\s+lane\s+blocked\s+and\s+right\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(31,'right lane blocked and right hand shoulder blocked','\bright\s+hand\s+shoulder\s+blocked\s+.\*\bright\s+lane\s+blocked',0,false,true,

'replace with empty string after counting'),

(35,'no other lane blocked and right hand shoulder blocked','^right\s+hand\s+shoulder\s+blocked\s+due to',0,false,true,

'replace with empty string after counting'),

(40,'multi lanes blocked and right hand shoulder blocked','lanes\s+blocked\s+and\s+right\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(41,'multi lanes blocked and right hand shoulder blocked','right hand shoulder blocked\s+and\b.+\band\b.+blocked',0,false,true,

'replace with empty string after counting'),

(50,'left lane blocked and left hand shoulder blocked','left\s+lane\s+blocked\s+and\s+left\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(51,'left lane blocked and left hand shoulder blocked','left\s+hand\s+shoulder\s+blocked\s+and\s+left\s+lane\s+blocked',0,false,true,

'replace with empty string after counting'),

(55,'left/right lane blocked and left hand shoulder blocked','^lane\s+blocked\s+and\s+left\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(56,'left/right lane blocked and left hand shoulder blocked','^left\s+hand\s+shoulder\s+blocked\s+and\s+lane\s+blocked',0,false,true,

'replace with empty string after counting'),

(60,'right lane blocked and left hand shoulder blocked','right\s+lane\s+blocked\s+and\s+left\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(65,'no other lane blocked and left hand shoulder blocked','^left\s+hand\s+shoulder\s+blocked\s+due to',0,false,true,

'replace with empty string after counting'),

(70,'multi lanes blocked and left hand shoulder blocked','lanes\s+blocked\s+and\s+left\s+hand\s+shoulder\s+blocked',0,false,true,

'replace with empty string after counting'),

(71,'multi lanes blocked and left hand shoulder blocked','left hand shoulder blocked\s+and\b.+\band\b.+blocked',0,false,true,

'replace with empty string after counting'),

(80,'one/multi lanes blocked without shoulder blocked ','^.\*\s+lanes\*\s+blocked\s+(?!shoulder blocked)',0,false,true,

'replace with empty string after counting'),

(90,'On the median','\s\*On the median[.]',0,false,true,'replace with empty string after counting'),

(100,'left hand shoulder as well as right hand shoulder blocked','\s+left hand shoulder blocked and right hand shoulder blocked\s+',0,false,true,

'replace with empty string after counting')

) as t(slno,concpt, pattrn, freqncy,embdable,exclsv,post\_action);

insert into concept7 SELECT

slno,

-- over (partition by variants, stem order by 1) as Slno,

concpt, pattrn, freqncy,embdable,exclsv,postaction

FROM (

VALUES

(110,'Interstate Hwy','i-\d+\s+?',0,false,true,'replace with empty string after counting'),

(120,'State Hwy','\s+(al|ak|az|ar|ca|co|ct|dc|de|fl|ga|hi|id|il|in|ia|ks|ky|la|me|md|ma|mi|mn|ms|mo|mt|ne|nv|nh|nj|nm|ny|nc|nd|oh|ok|or|pa|ri|sc|sd|tn|tx|ut|vt|va|wa|wv|wi|wy)-\d+\s+',

0,false,true,'replace with empty string after counting'),

(130,'US Hwy','\s+(us)( |-)\d+\s+',0,false,true,'replace with empty string after counting'),

(140,'County Hwy','\s+cr( |-)\d+\s+',0,false,true,'replace with empty string after counting'),

(150,'Local Street','\b(\w+\s+)+(street|st|court|ct|avenue|ave|av|road|rd)[.]\*\s+',0,false,true,'replace with empty string after counting')

(160,'Local Street','\b(\w+\s+)+(boulevard|blvd|parkway|pkwy|drive|drv|dr|lane|ln)[.]\*\s+',0,false,true,'replace with empty string after counting'),

) as t(slno,concpt, pattrn, freqncy,embdable,exclsv,postaction);

1. The problem is to display frequency for each major concept (on 5s for serial no) (including all ofd their representation) as they happen on base\_corpus3 table.

At the end of this step, we should have a few key-terms and their frequencies. Because the base corpus will be very large, do it by distributing different responsibilities to different people (after step a is done). within group. To learn basics about NLP, one may watch a presentation : <https://www.youtube.com/watch?v=xvqsFTUsOmc>

**Note:** We apply changes and create a temp table and on that we apply more changes creating another temp table.

**Note:** There are many reg-exp functions supported within Athena SQL like regexp\_extract(), regexp\_extract\_all(), regexp\_like(), many variants of regexp\_replace(). The reference is at web-site:

<https://prestodb.io/docs/current/functions/regexp.html>. We perform pattern matching using meta patterns like \p{Punct} for punctuation, \d for a single digit, \D for non-digit char, \s for white-space char, \S non-whitespace char, \w for word char, \W for non-word char etc. In addition, we can match patterns of many classes and refer to them separately at the same time (like (word1) (word2) (word3) in a pattern can be referred to as $1, $2, $3 in the replacement string for example). For more details of available meta patterns, check the web-site : <https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html#cg>

1. Can we build a WordCloud diagram for these tiniest phrases (or even single words) using QuickSight?
2. Intermediate step: We need to build one intermediate table before going onto further problems. Precipitation is recorded for each accident along with start-time (combination of start\_date and time). We need to know how to count precipitation (let us talk about only rain neither snow nor sleet) for a state and a date. That means we look at precipitation if it is more than 0 and weather\_condition includes ‘rain’ or ‘cloudy’ or ‘overcast’ word in case-insensitive way. So in order to count precipitation for a day, we need to add (timewise) all the precipitation figures for each 5digit\_zip-code and calculate average for the state over those zips (for the start\_date) (although it is only an approximate method). We create a temp table of state, zip-code, latest start\_time over the day, precipitation for rain. From this table (avg\_precipitation table) we generate another one having state, date, avg\_prec\_over\_zip. This table should be used for summary report generation (problem 14). Query: Almost like

create table avg\_precipitation AS

select state, date\_format(start\_time, '%Y-%m-%d') AS start\_date, avg(tot\_prec) AS avg\_prec

from (

select state,zipcode, start\_time, date\_diff('second', start\_time, end\_time) AS dur, "precipitation(in)" AS prec, sum("precipitation(in)") OVER (partition by state,

date\_format(start\_time, '%Y-%m-%d'),zipcode order by start\_time) AS tot\_prec,

rank() OVER (partition by state,

date\_format(start\_time, '%Y-%m-%d'),zipcode order by start\_time DESC ) AS rnk

from us\_accidents\_4 where

--state='CA' and

(lower(weather\_condition) like '%rain%' or lower(weather\_condition) like '%cloud%')

and

date\_format(start\_time, '%Y-%m-%d') like '2019-12-%' and "precipitation(in)" > 0.0

and zipcode is not null

order by state, date\_format(start\_time, '%Y-%m-%d'),zipcode, start\_time

) where rnk=1

group by state, date\_format(start\_time, '%Y-%m-%d')

order by state, date\_format(start\_time, '%Y-%m-%d');

1. Simple aggregate function: Enough of individual rows! We want to try generating a very simple summary report on state, date, precipitation and a running total over date for each state. It can be easily done using “**group by**” clause but we want to take the alternative approach. Can we build this report from table avg\_precipitation (output of problem 12) without using “group by” clause in Athena? We need to plot running totals of precipitation for each top 5 states against year, week (or month).

**Hint:** Use Window function.

1. **Summary data analytics:** Enough of individual rows! We want try generating summary data and then multi-level summary data as report and then present that report as visualizations. Let us take the simple case of problem 12. Suppose, we need report on total precipitation by state, date (start\_date) and also (in the same) total by state and also total by date and also one line final total precipitation for all accidents with accident date between ’12-20-2019’ and ’12-28-2019’ (both inclusive) and weather\_condition includes ‘rain’ or ‘cloudy’ or ‘overcast’. From this table we generate another one having state, date, avg\_prec\_over\_zip. The output table (avg\_precipitation) should be used for summary report generation. This report must be ordered by state, dt with state level total line interleaving and then dt level and then the final total. Can you use “**grouping sets**” clause to generate this report?
2. Can we try to generate the same report in 14 by using “**rollup**” (forget dt level) clause instead of “**grouping sets**” clause? Can we try to generate the same report in 14 by using “**cube**” clause instead of “**grouping sets**” clause?
3. There are 45 states coming up with the rollup output (in 14). Can we make 45 rectangle verticals (one for each state) and then within each vertical paint with different color for each accident date?