**Problem Statement**

**Objectives of the Case Study**

* Primarily, the case study is meant as a deep-dive into the usage of Spark. As you have seen while working with Spark, its syntax behaves differently from your regular R syntax. One of the major objectives of this case study is gaining familiarity with how analysis works in SparkR as opposed to base R.
* Learning the basic idea behind using functions in SparkR can be transferred to using other libraries like PySpark. If you are in a company where Python is a primary language, you can easily pick up PySpark syntax and use Spark’s processing power.
* The actual process of running a model-building command boils down to a few lines of code. In trying to find inference from data, the most time-consuming step is preparing the data up to the point of model-building. Hence, we’re gearing this case study more towards exploratory analysis.

**Problem Statement**

Big data analytics allows you to analyse data at scale. It has applications in almost every industry in the world. Let’s consider an unconventional application that you wouldn’t ordinarily encounter.

New York City is a thriving metropolis. Just like most other metros that size, one of the biggest problems its citizens face is parking. The classic combination of a huge number of cars and a cramped geography is the exact recipe that leads to a huge number of parking tickets.

In an attempt to scientifically analyse this phenomenon, the NYC Police Department has **collected data for parking tickets**. Out of these, the data files from 2014 to 2017 are publicly available on Kaggle. We will try and perform some **exploratory analysis** on this data. Spark will allow us to analyse the full files at high speeds, as opposed to taking a series of random samples that will approximate the population.

For the scope of this analysis, we wish to compare the phenomenon related to parking tickets over three different years - **2015, 2016, 2017**. All the analysis steps mentioned below should be done for three different years. Each metric you derive should be compared across the three years. Use the Fiscal years as per the files. You can use calendar year if you like - you will not lose any marks for performing the analysis this way.

**Note**: Although the broad goal of any analysis of this type would indeed be better parking and fewer tickets, we are **not looking for recommendations on how to reduce the number of parking tickets** - there are no specific points reserved for this.

 The purpose of this case study is to conduct an exploratory data analysis that helps you understand the data. Since the size of the dataset is large, your queries will take some time to run, and you will need to identify the correct queries quicker. The questions given below will guide your analysis.

The dataset structure is available [on this page](https://www.kaggle.com/new-york-city/nyc-parking-tickets/data) along with the data.

**General Guidelines:**

1. Your submission will consist of one file of code and one text file. In the text file, you should write some subjective observations you have made from this data. In the code file, provide the code and the query results as comments.
2. If you make any specific assumptions related to these questions, be sure to state them.
3. The analysis has to be performed for all the three years. Apart from analysis asked in the questions, provide a comparison of the findings from three years.
4. Your Corestack clusters are costly, and one person's cluster cannot handle the full analysis. Since this is a group case study, be efficient in dividing work between the four members.
5. Include all the necessary commands to prevent errors.
6. The queries may take time to get executed. Please have some patience. If you are getting errors with correct queries, restart the R session and try again as the session may have expired.
7. Keep a copy of the commands on your local drive so that you do not lose any work in case of session expiry.
8. If you want to run SQL commands, create an SQL view first. Also please ensure that if you make any changes in the table(like substitution or dropping null values), you update the SQL view related to that table for further analysis.

**Accessing the dataset**

The data for this case study has been placed in HDFS at the following path:

**'/common\_folder/nyc\_parking/Parking\_Violations\_Issued\_-\_Fiscal\_Year\_201x.csv'**

where 'x' is between {5,6,7}

**Questions to be answered in the analysis**

The following analysis should be performed on RStudio mounted on your Corestack cluster, using the SparkR library. Remember, you should do this analysis for all the three years, and possibly compare metrics and insights across the years.

**Examine the data**

1. Find the total number of tickets for each year.
2. Find out the number of unique states from where the cars that got parking tickets came from. *(Hint: Use the column 'Registration State')*  
   *There is a numeric entry in the column which should be corrected. Replace it with the state having maximum entries. Give the number of unique states for each year again.*
3. Some parking tickets don’t have the address for violation location on them, which is a cause for concern. Write a query to check the number of such tickets.  
   *The values should not be deleted or imputed here. This is just a check.*

**Aggregation tasks**

1. How often does each violation code occur? Display the frequency of the top five violation codes.
2. How often does each 'vehicle body type' get a parking ticket? How about the 'vehicle make'? (***Hint***: *find the top 5 for both)*
3. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequency of tickets for each of the following:
   1. 'Violation Precinct' (this is the precinct of the zone where the violation occurred). Using this, can you make any insights for parking violations in any specific areas of the city?
   2. 'Issuer Precinct' (this is the precinct that issued the ticket)  
      *Here you would have noticed that the dataframe has 'Violating Precinct' or 'Issuing Precinct' as '0'. These are the erroneous entries. Hence, provide the record for five correct precincts. (****Hint****: print top six entries after sorting)*
4. Find the violation code frequency across three precincts which have issued the most number of tickets - do these precinct zones have an exceptionally high frequency of certain violation codes? Are these codes common across precincts?   
   ***Hint****: You can analyse the three precincts together using the 'union all' attribute in SQL view. In the SQL view, use the 'where' attribute to filter among three precincts and combine them using 'union all'.*
5. You’d want to find out the properties of parking violations across different times of the day:
   1. Find a way to deal with missing values, if any.  
      ***Hint****: Check for the null values using 'isNull' under the SQL. Also, to remove the null values, check the 'dropna' command in the API documentation.*
   2. The Violation Time field is specified in a strange format. Find a way to make this into a time attribute that you can use to divide into groups.
   3. Divide 24 hours into six equal discrete bins of time. The intervals you choose are at your discretion. For each of these groups, find the three most commonly occurring violations.  
      ***Hint****: Use the CASE-WHEN in SQL view to segregate into bins. For finding the most commonly occurring violations, a similar approach can be used as mention in the hint for question 4.*
   4. Now, try another direction. For the 3 most commonly occurring violation codes, find the most common time of the day (in terms of the bins from the previous part)
6. Let’s try and find some seasonality in this data
   1. First, divide the year into some number of seasons, and find frequencies of tickets for each season. *(****Hint****: Use Issue Date to segregate into seasons)*
   2. Then, find the three most common violations for each of these seasons.  
      *(****Hint****: A similar approach can be used as mention in the hint for question 4.)*
7. The fines collected from all the parking violation constitute a revenue source for the NYC police department. Let’s take an example of estimating that for the three most commonly occurring codes.
   1. Find total occurrences of the three most common violation codes
   2. Then, visit the website:  
      <http://www1.nyc.gov/site/finance/vehicles/services-violation-codes.page>  
      It lists the fines associated with different violation codes. They’re divided into two categories, one for the highest-density locations of the city, the other for the rest of the city. For simplicity, take an average of the two.
   3. Using this information, find the total amount collected for the three violation codes with maximum tickets. State the code which has the highest total collection.
   4. What can you intuitively infer from these findings?