Module 3 Project - Group 15
Daniel Geneta, Hans Laguna

1. Installation

For our M3 Final Project, we were tasked to perform real-time data analytics on a CCTV counter table using a database connection. Our group chose to use the data visualization program, Tableau, for our real-time data analysis. We also chose to use Cassandra as our real-time database.

In order to start off the project, we will first need to install Tableau Desktop to our system. First, head to: https://www.tableau.com/products/desktop and apply for a free trial. With their free trial, you may use Tableau desktop for 14 days with all the in-house features. You may need to sign up and create an account. For our project, we already had a pre-existing Tableau account.

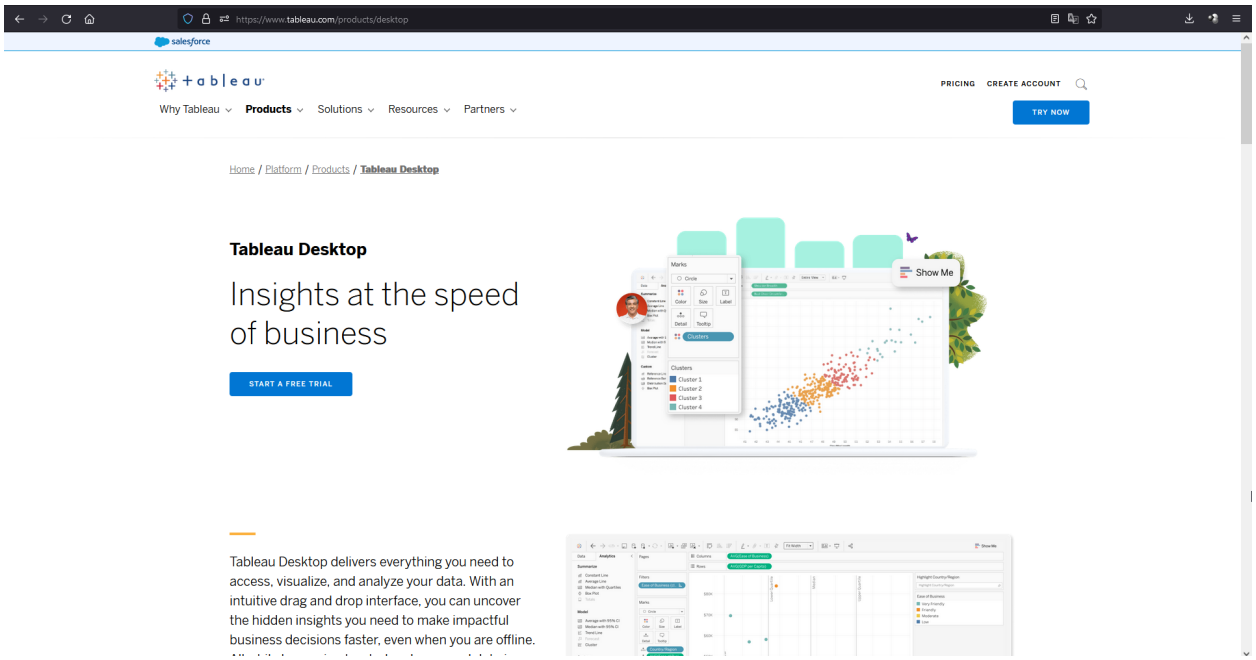


Figure 1.1: Tableau Desktop

Once we have logged in our Tableau account, we may proceed to download the Tableau Desktop installer from the link sent from your email. After clicking the link sent to your Tableau account's linked email, you should be greeted with an installation page seen in Figure 1.2 below.

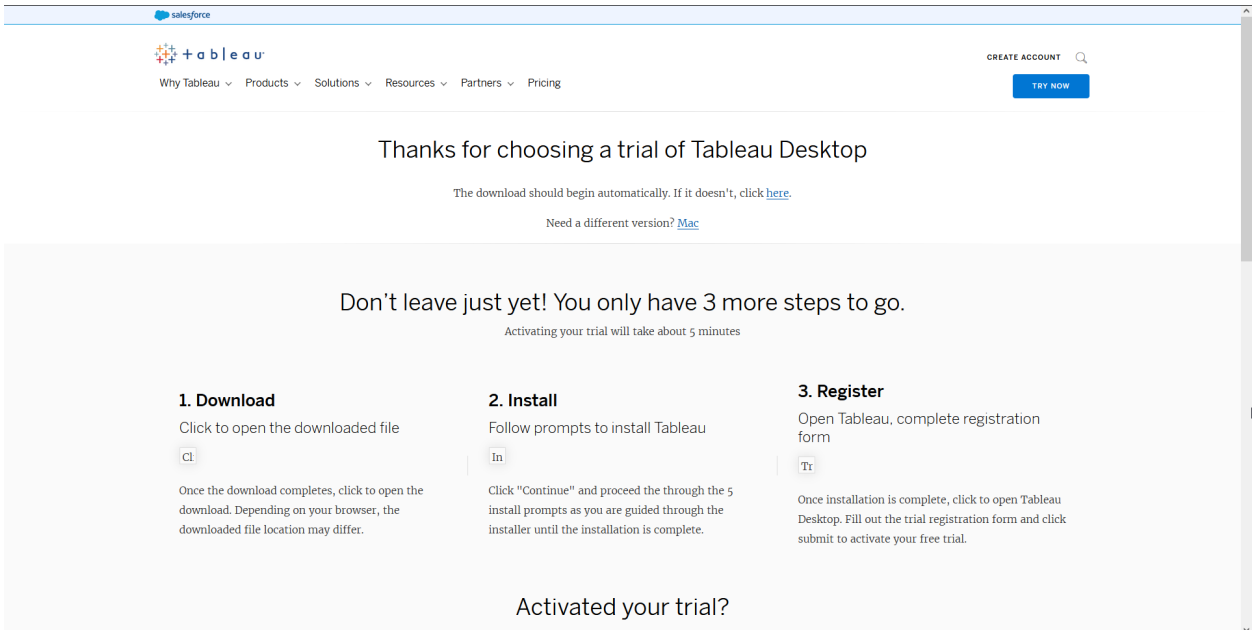


Figure 1.2: Tableau Desktop Download

After downloading the installer, proceed to install Tableau Desktop to your system.

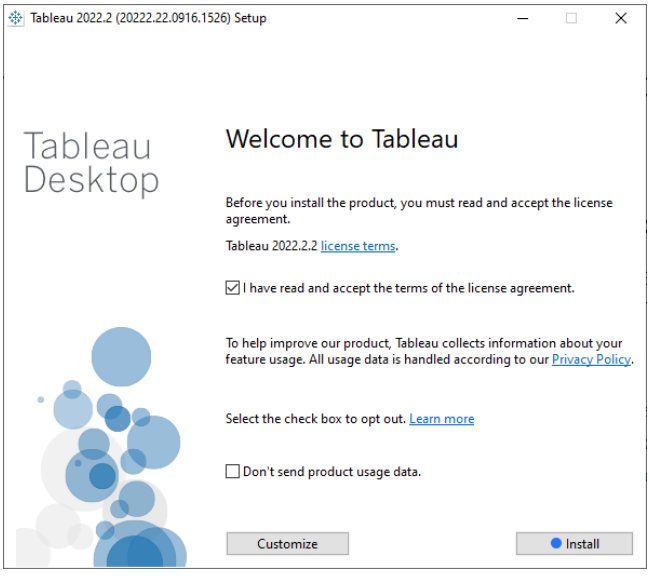


Figure 1.3: Tableau Installation

You may install Tableau with an activated product key. However, for this project, we will just be using the built-in 14 day trial. Select the “Start trial now” option and proceed.

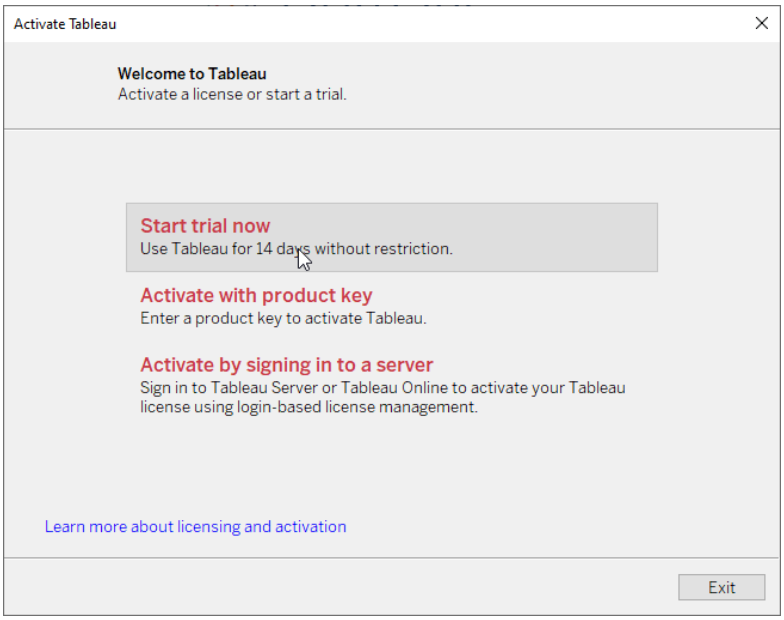


Figure 1.4: Free Trial

Using your Tableau account email address, fill up the following form requirements.

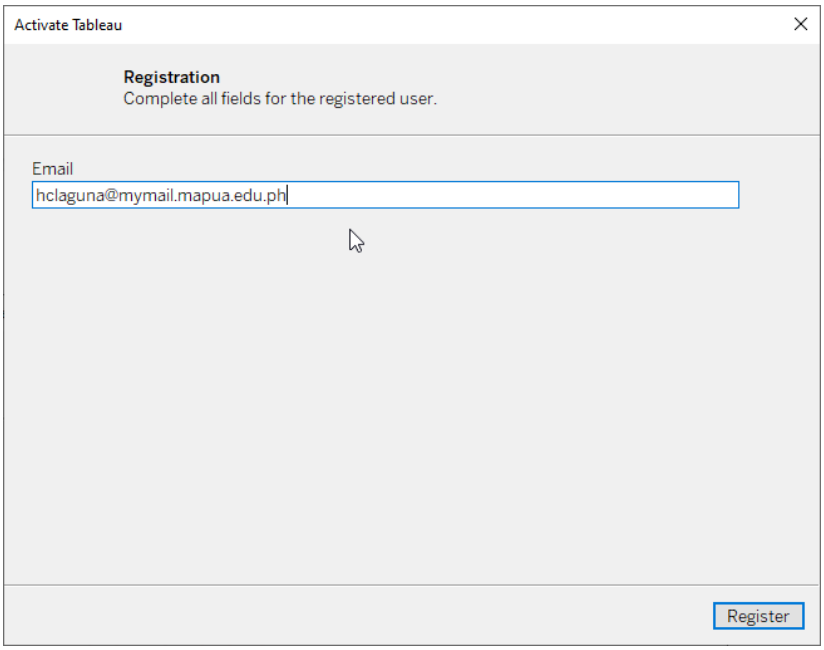


Figure 1.5: Enter Email Address

Once finished with the account registration, Tableau Desktop will proceed to be installed to your system. After installation, you will be greeted with a startup page as seen from Figure 1.6 below.

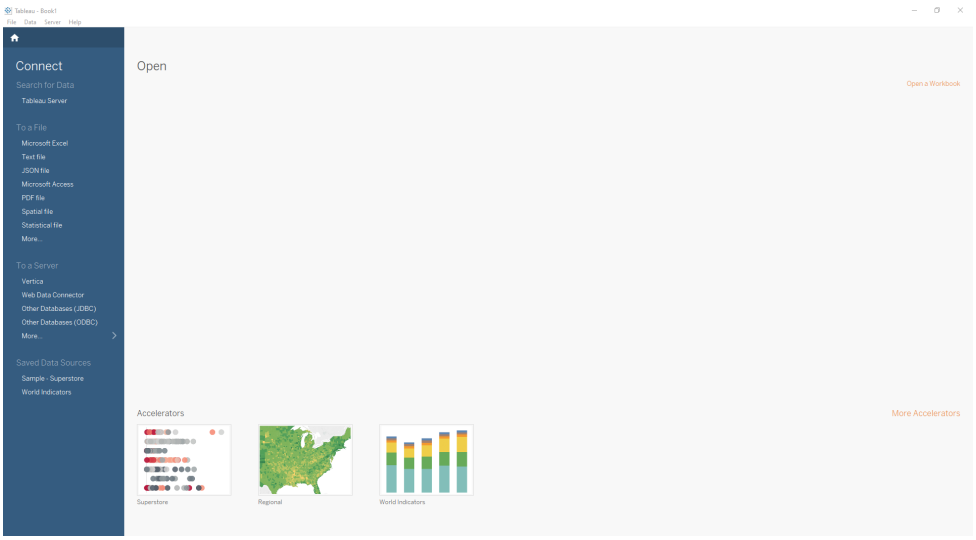


Figure 1.6: Tableau Startup Screen

Once we have successfully installed Tableau, we will need to install a Cassandra ODBC driver. This ODBC driver will allow our Apache Cassandra instance to communicate with Tableau Desktop. To install this driver, head to: https://downloads.datastax.com/#odbc-jdbc-drivers and install the Datastax ODBC Cassandra driver.

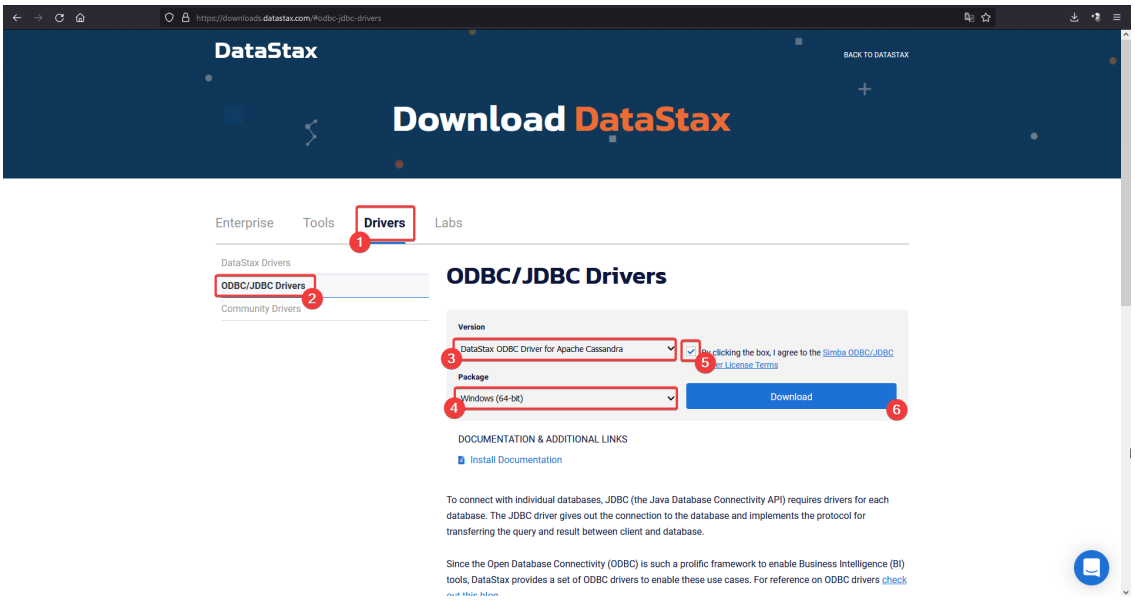


Figure 1.7: Cassandra ODBC Driver

Open the DataStax installer and proceed to install the driver to your system.

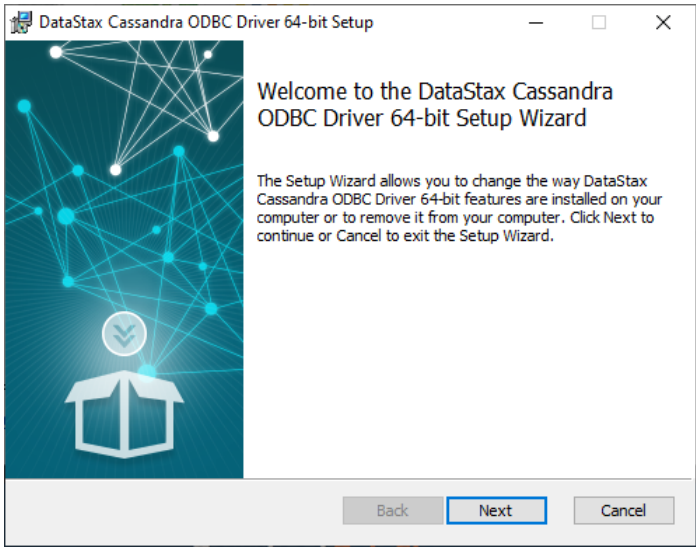


Figure 1.8: DataStax Installation

2. Prerequisites

In this project, we will be generating a new set of data for our CCTV counts dataset. Since we will be creating a new set of data, we will need to startup zookeeper and kafka. Enter your Kafka installation directory. Once inside the directory, input “cmd” in the directory search bar as seen in Figures 2.1 and 2.2 below. This will allow us to open the kafka directory through command prompt.

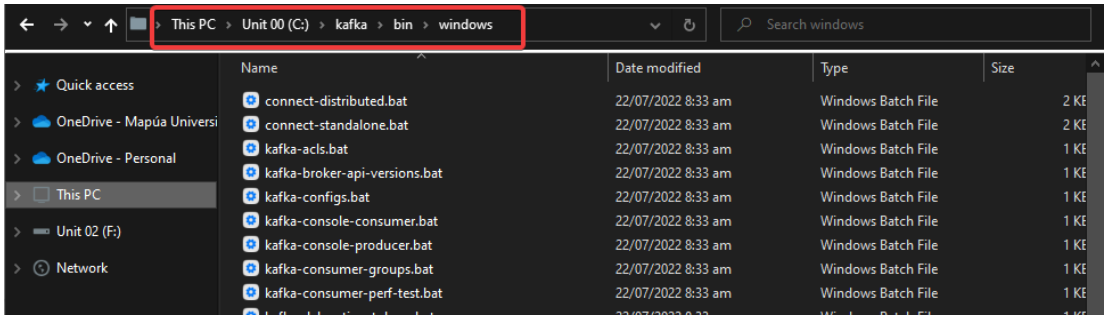


Figure 2.1: Open Kafka Directory

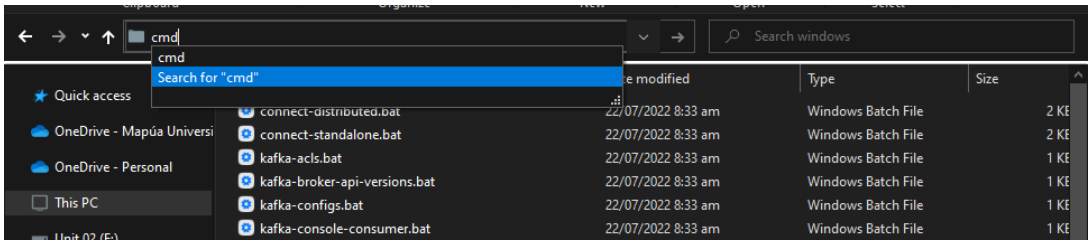


Figure 2.2: Startup cmd

Once the command prompt instance as opened, input: zookeeper-server-start.bat ..\..\config\zookeeper.properties. This command will startup zookeeper.

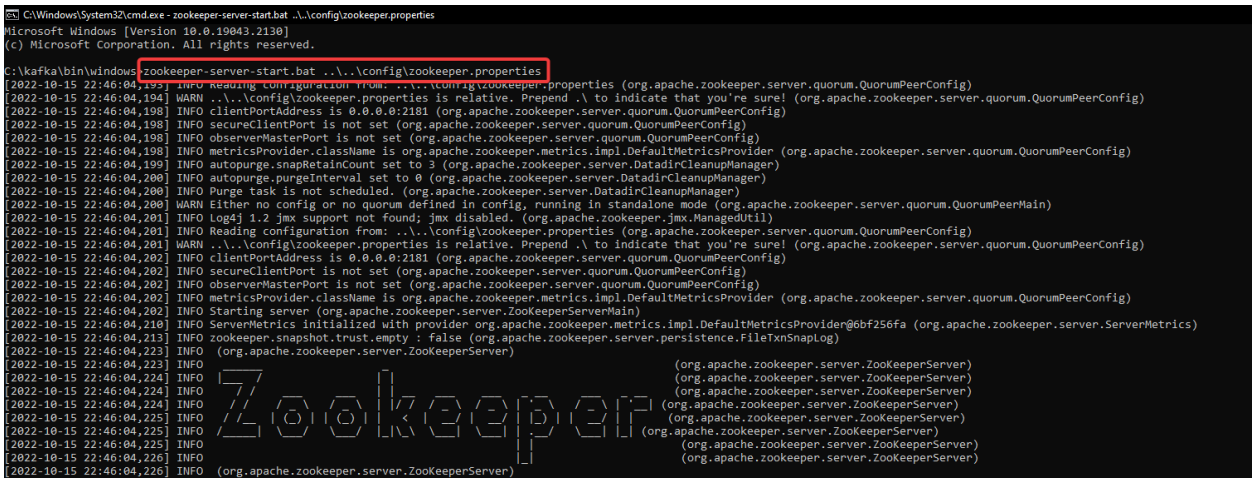


Figure 2.3: Startup Zookeeper

Once Zookeeper has finished initiating, we may proceed to start our Kafka Server. Input the following command: kafka-server-start.bat ..\..\config\server.properties. After inputting this, your Kafka server should startup.

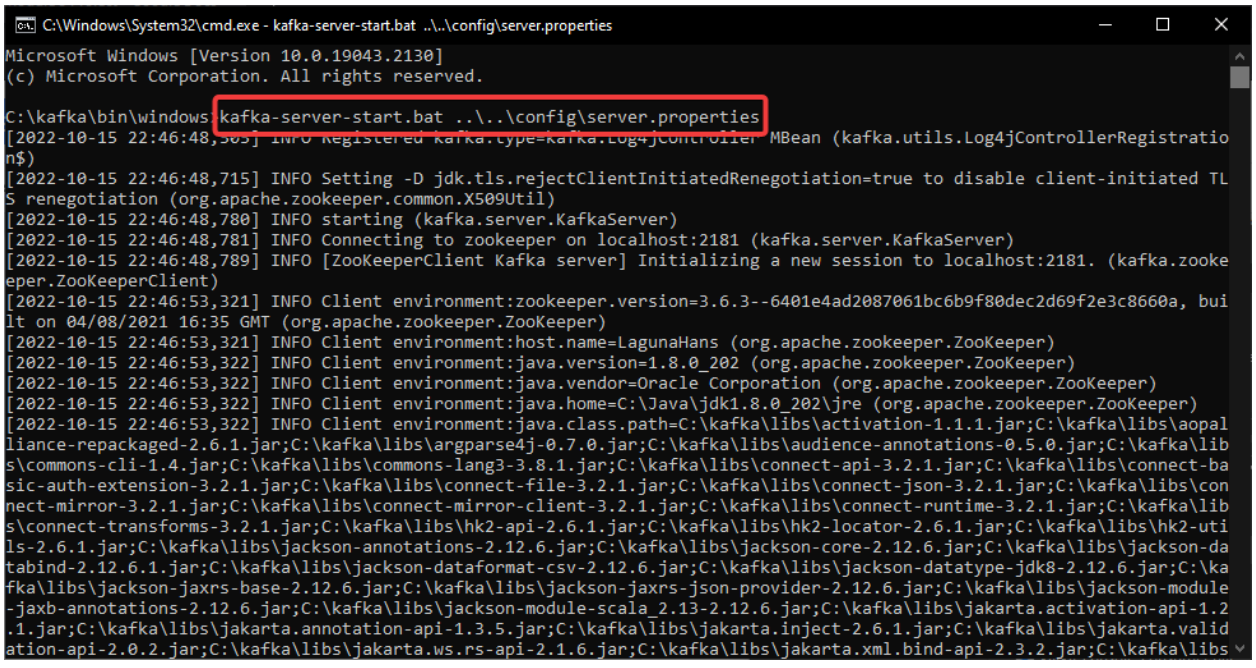


Figure 2.4: Startup Kafka

Next, we will need to initiate our Cassandra server as well. This is where we will be storing our generated data. From our start menu, open a command prompt instance in administrator mode.

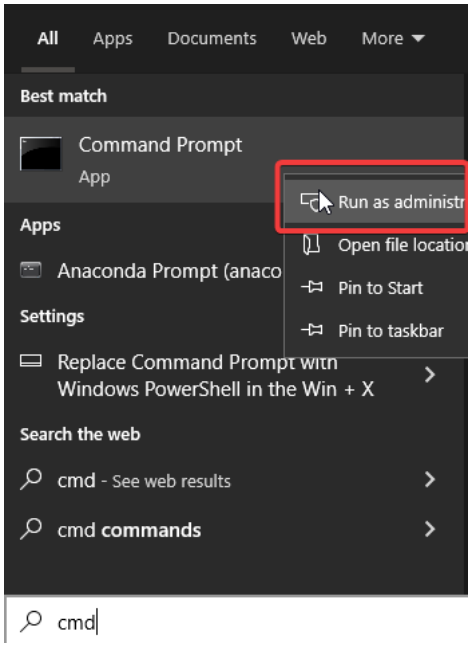


Figure 2.5: Run CMD in administrator

Next, input “cassandra” on the command line to startup the Cassandra server.

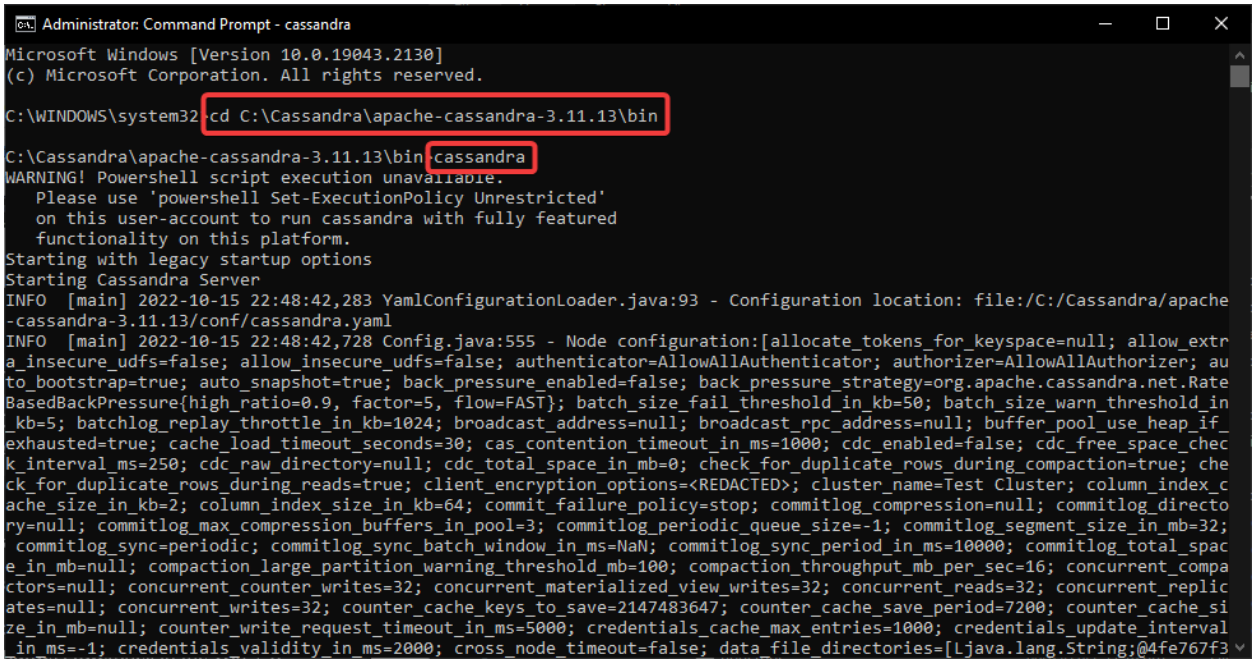


Figure 2.6: Startup Cassandra

Lastly, we will need to initiate our hadoop instance. Open command prompt in administrator mode once more and enter your hadoop's bin directory. Once the directory has been loaded to a command prompt instance, input the "start-all" command in order to startup DFS and Yarn.

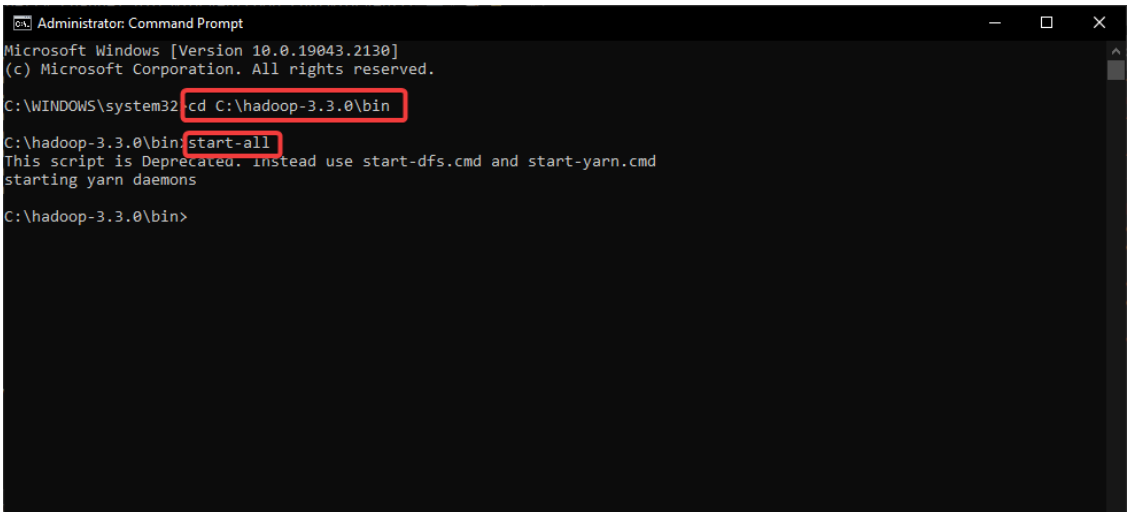


Figure 2.7: Hadoop Startup

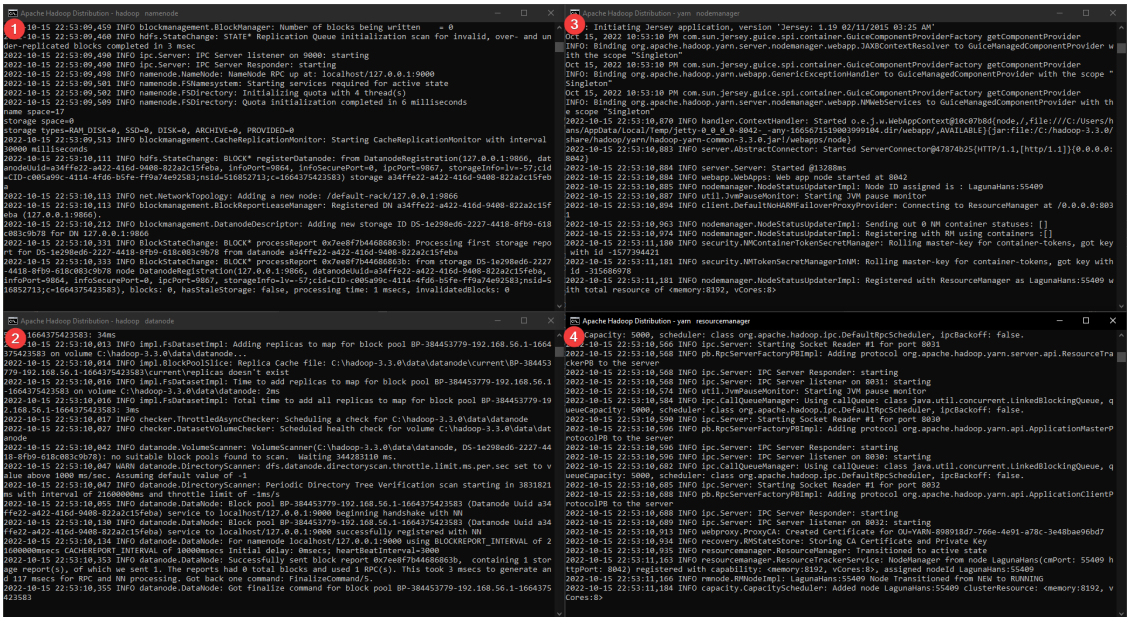


Figure 2.8: DFS and Yarn Startup

3. Connecting Cassandra to Tableau

Once all the prerequisite programs such as Kafka, Cassandra and Hadoop have been initiated, we may now connect our Cassandra server to Tableau. First, open your control panel from the start menu.

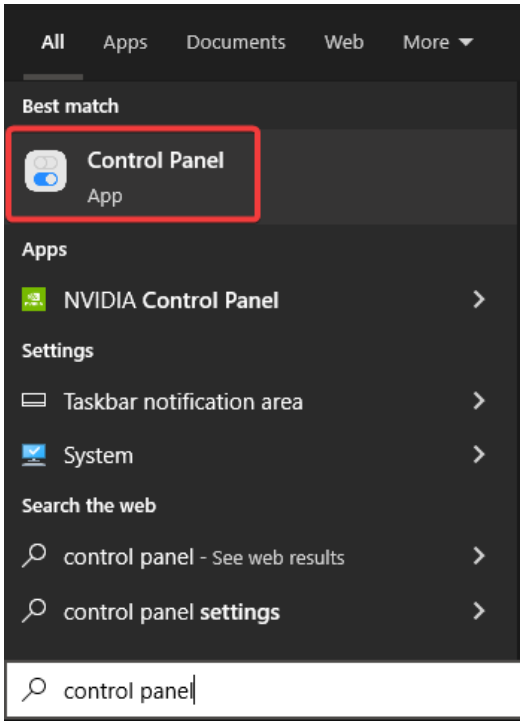


Figure 3.1: Open Control Panel

Inside the control panel, search up the administrative tools category and open it.

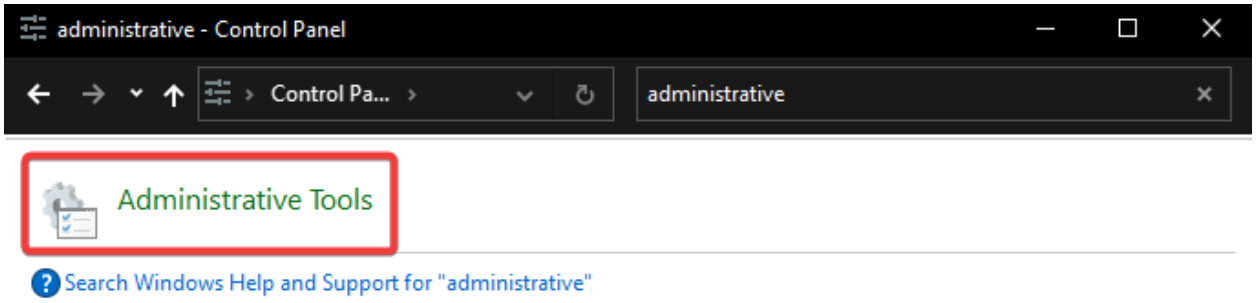


Figure 3.2: Open Administrative Tools

After opening Administrative Tools, proceed to open the ODBC Data source (64 bit). Since our system is 64-bit, we will select the 64-bit version.

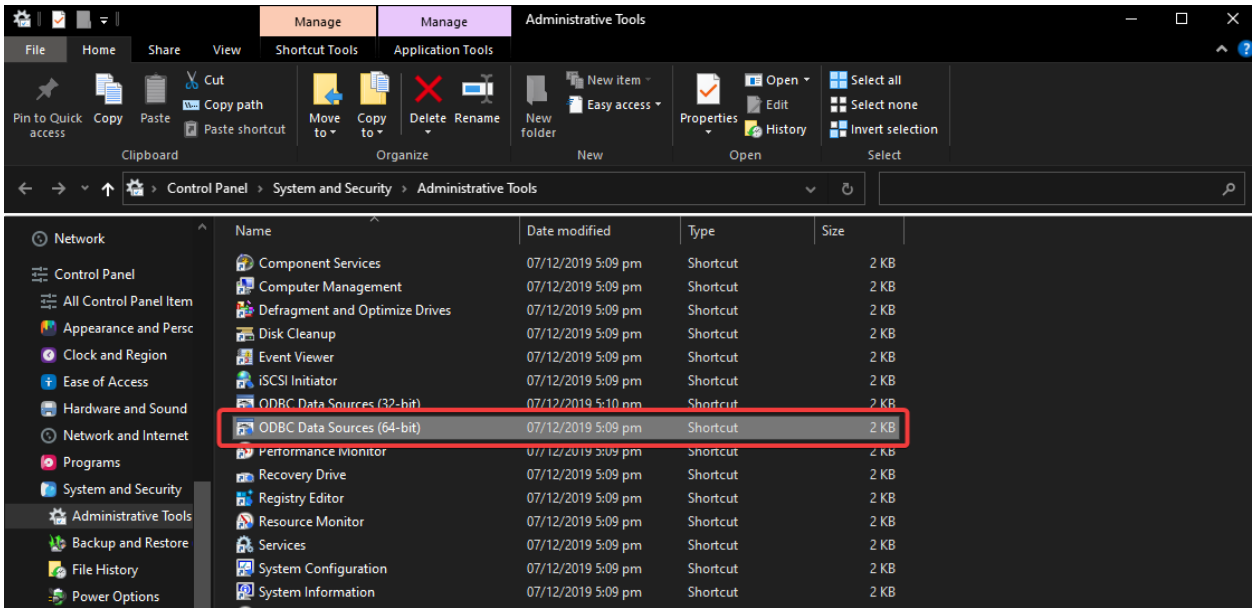


Figure 3.3: Open ODBC Data Sources

The next step is to configure our previously installed DataStax Cassandra ODBC DSN. This configuration will allow our running Cassandra Server to communicate with Tableau. That way, we may send live data from the Cassandra table to Tableau's data visualization analysis.

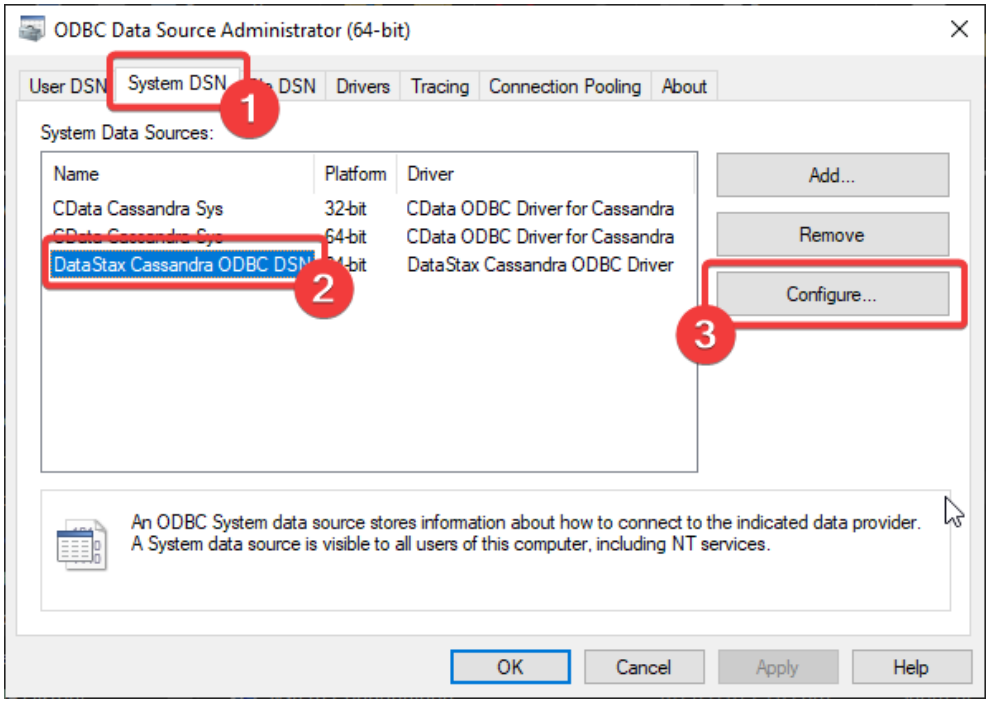


Figure 3.4: Configure ODBC

From the configuration menu, proceed to fill up the blank text boxes. For the host, set it to “localhost” then select your default keyspace. Once we have configured our DSN, click TEST to check if the connection between Cassandra and the driver is successful.

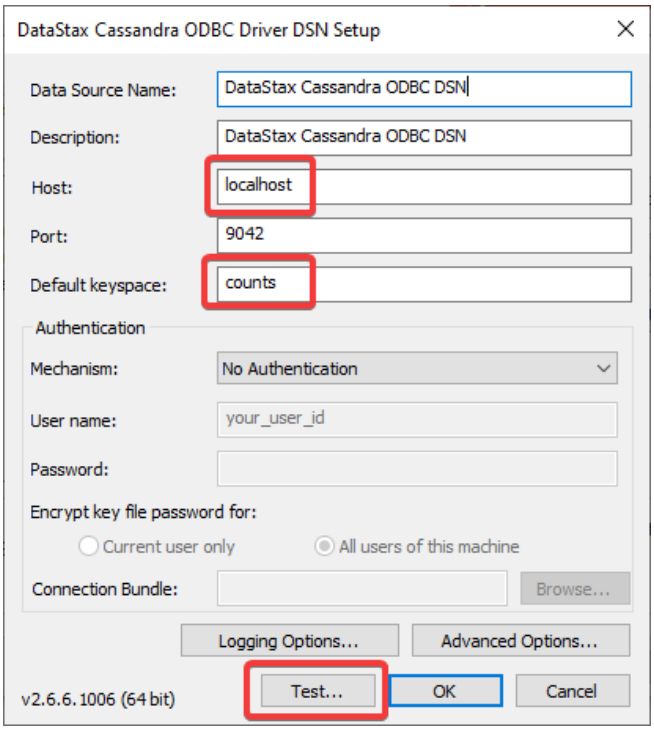


Figure 3.5: CSN Setup

If the connection test was successful, a window similar to Figure 3.6 should appear.

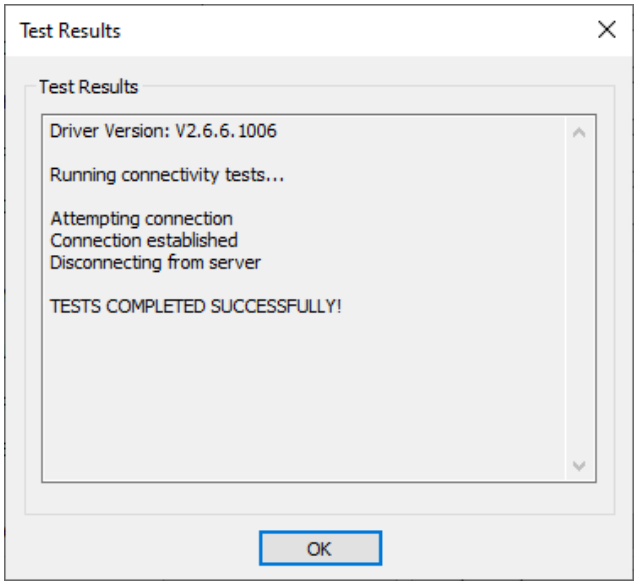


Figure 3.6: Successful Connection Test

Heading back to Tableau, select the “Other Databases ODBC” option to link our Cassandra server to Tableau.

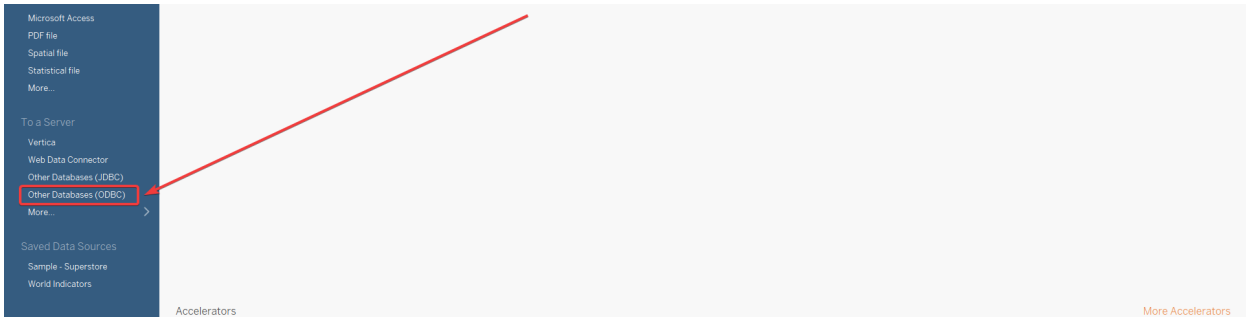


Figure 3.7: ODBC Tableau

Next, select the DataStax Cassandra ODBC DSN option and click “Connect.” There should be no pop-ups after clicking connect if the connection was successful. Next, click Sign-in to start our Tableau Project.

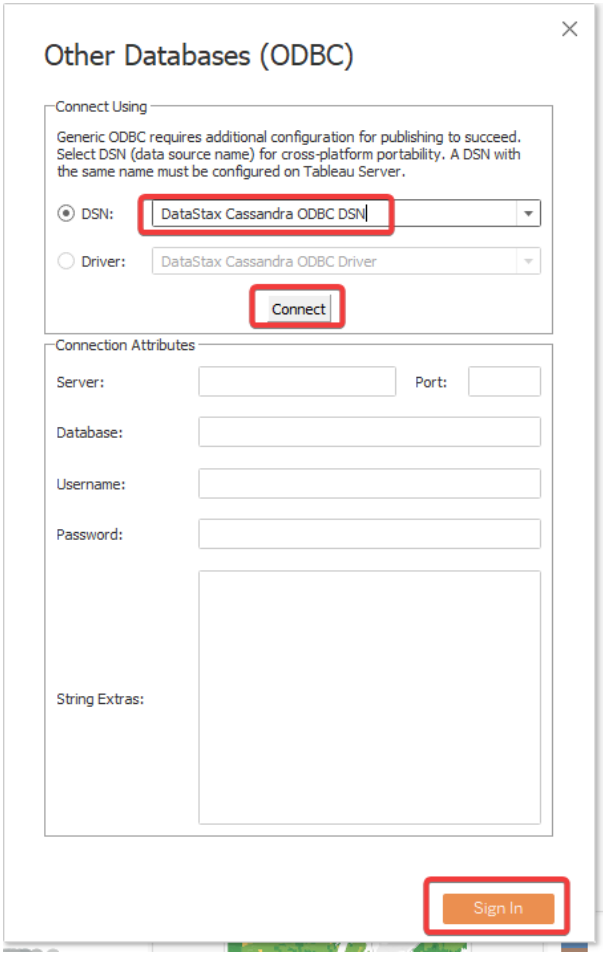


Figure 3.8: ODBC Cassandra to Tableau

Once we have opened our new project, select the Cassandra dataset from the list of databases from the startup page of Tableau.

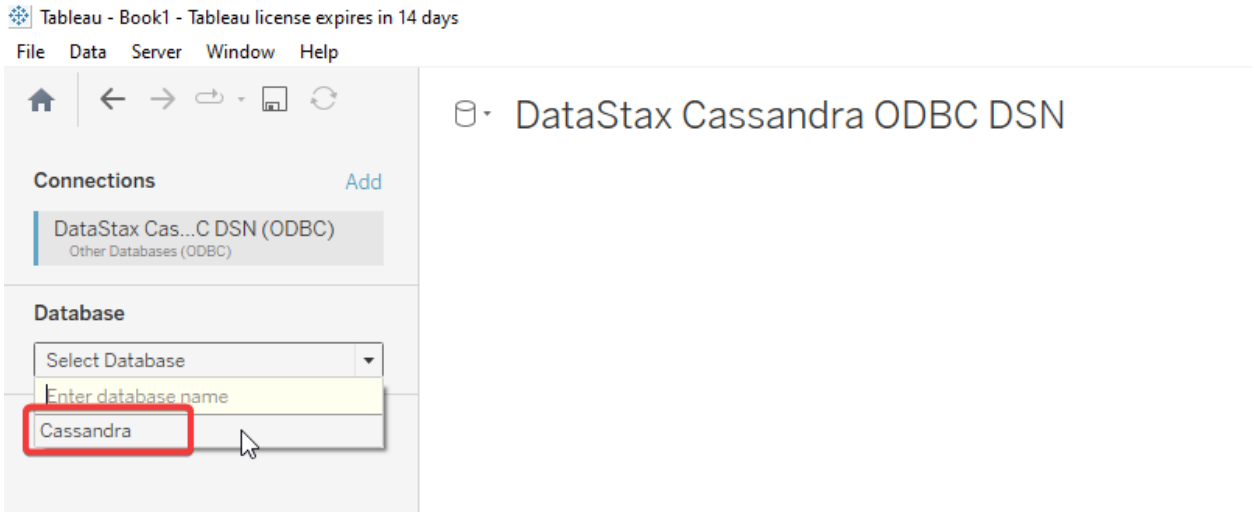


Figure 3.9: Connect Tableau to Cassandra

4. Producing Dataset

Since we will be generating our dataset and saving it to cassandra, we will need a python script that would output data to kafka then proceed to save it in Cassandra. In this code, we will be generating a random set of integers for each vehicle that passes by a CCTV sensor. There are 10 sensors set in an array called sensor_list as seen in Figure 4.1 below. After every produced row of data, the python script will select a random CCTV sensor to allocate all the generated data to. This dataset also saves date and time. Since we will be preparing our code beforehand, we programmed the Python producer to generate 10 rows of data before incrementing the hour by 1. As seen in Figure 4.1, the timedelta of the datetime is increased by 1 every 10 rows of generated data. This way, we were able to create over 7,400 rows of cctv data with dates ranging from October 20 to November 21.

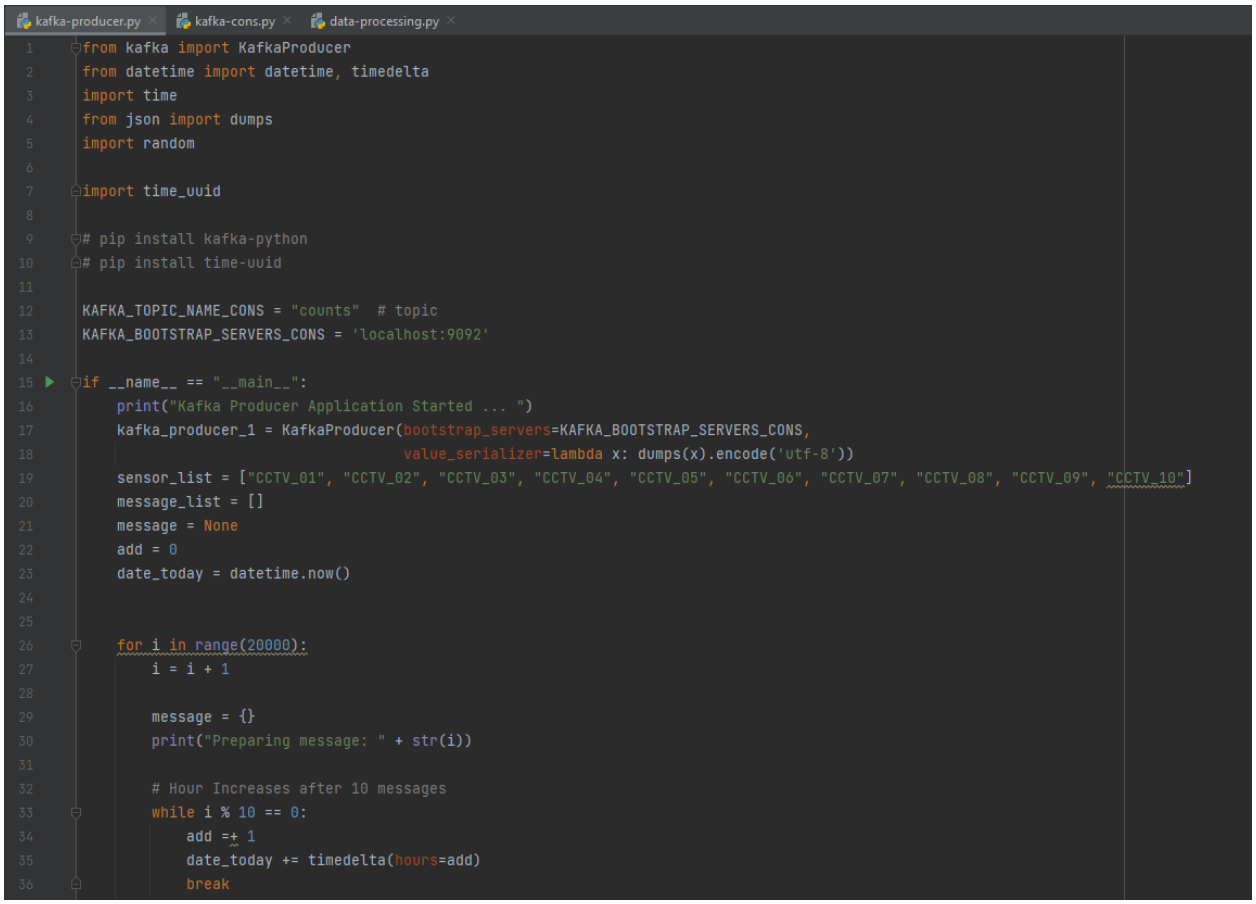
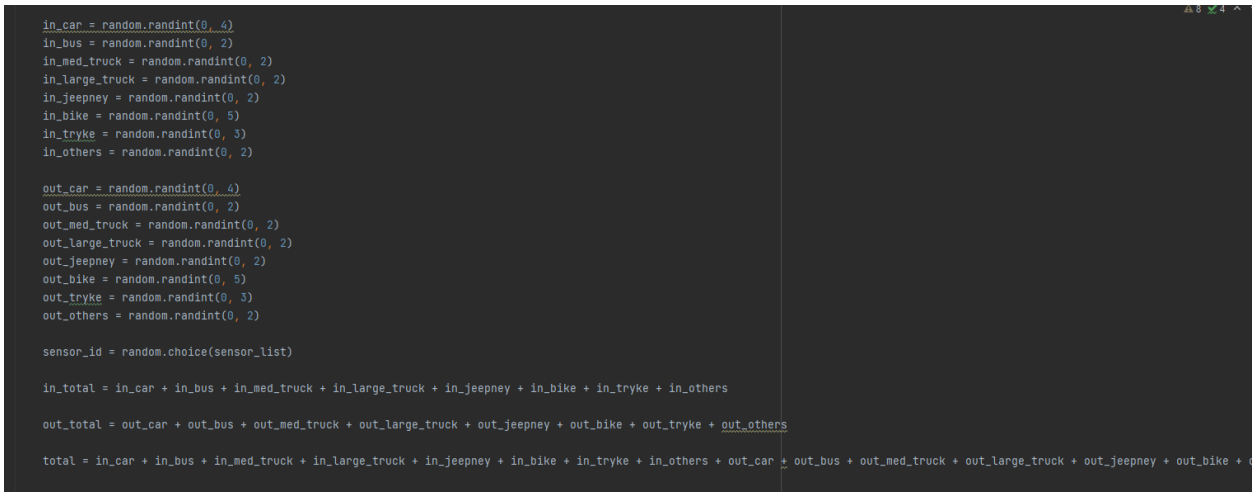


Figure 4.1: Python Data Producer.

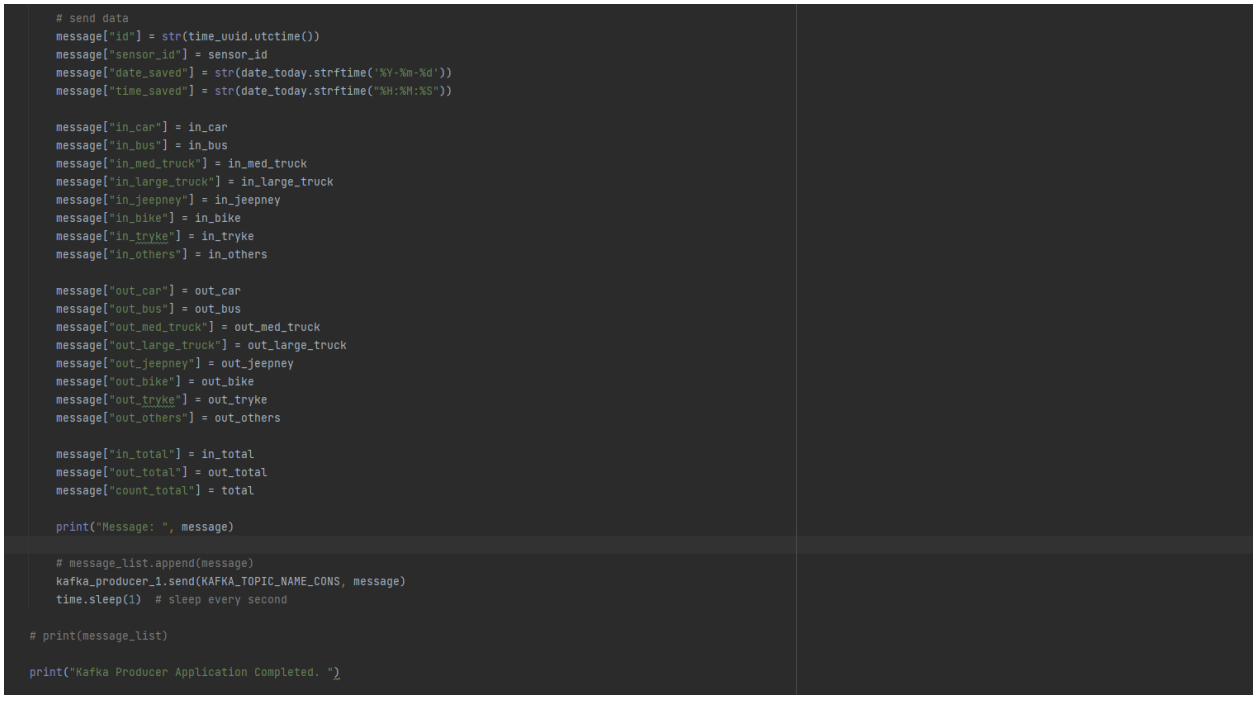
As previously stated, the count of vehicles were randomly generated for the dataset we used.



```
    in_car = random.randint(0, 4)
    in_bus = random.randint(0, 2)
    in_med_truck = random.randint(0, 2)
    in_large_truck = random.randint(0, 2)
    in_jeepney = random.randint(0, 2)
    in_bike = random.randint(0, 5)
    in_tryke = random.randint(0, 3)
    in_others = random.randint(0, 2)

    out_car = random.randint(0, 4)
    out_bus = random.randint(0, 2)
    out_med_truck = random.randint(0, 2)
    out_large_truck = random.randint(0, 2)
    out_jeepney = random.randint(0, 2)
    out_bike = random.randint(0, 5)
    out_tryke = random.randint(0, 3)
    out_others = random.randint(0, 2)

    sensor_id = random.choice(sensor_list)

    in_total = in_car + in_bus + in_med_truck + in_large_truck + in_jeepney + in_bike + in_tryke + in_others

    out_total = out_car + out_bus + out_med_truck + out_large_truck + out_jeepney + out_bike + out_tryke + out_others

    total = in_car + in_bus + in_med_truck + in_large_truck + in_jeepney + in_bike + in_tryke + in_others + out_car + out_bus + out_med_truck + out_large_truck + out_jeepney + out_bike + o
```

Figure 4.2: Python Data Producer (part 2)

The data generated also included a total of the incoming and outgoing cars from the area of the cctv. This meant that our data had roughly 14 columns of oncoming vehicles/outgoing vehicles and their totals.



```
    # send data
    message["id"] = str(time_uuid.utctime())
    message["sensor_id"] = sensor_id
    message["date_saved"] = str(date_today.strftime('%Y-%m-%d'))
    message["time_saved"] = str(date_today.strftime("%H:%M:%S"))

    message["in_car"] = in_car
    message["in_bus"] = in_bus
    message["in_med_truck"] = in_med_truck
    message["in_large_truck"] = in_large_truck
    message["in_jeepney"] = in_jeepney
    message["in_bike"] = in_bike
    message["in_tryke"] = in_tryke
    message["in_others"] = in_others

    message["out_car"] = out_car
    message["out_bus"] = out_bus
    message["out_med_truck"] = out_med_truck
    message["out_large_truck"] = out_large_truck
    message["out_jeepney"] = out_jeepney
    message["out_bike"] = out_bike
    message["out_tryke"] = out_tryke
    message["out_others"] = out_others

    message["in_total"] = in_total
    message["out_total"] = out_total
    message["count_total"] = total

    print("Message: ", message)

    # message_list.append(message)
    kafka_producer_1.send(KAFKA_TOPIC_NAME_CONS, message)
    time.sleep(1)  # sleep every second

# print(message_list)

print("Kafka Producer Application Completed. ")
```

Figure 4.3: Python Data Producer (part 3)

Once we have finished the code for the producer script, we will need to create a new keyspace and table in our Cassandra instance. To do this, we will need to open another instance of command prompt and access the cqlsh command line. Once we have accessed the cqlsh command line, create a new keyspace using the command in Figure 4.4 below. CREATE KEYSPACE cctv WITH replication = {'class':'SimpleStrategy','replication_factor': '1'} AND durable_writes = 'true';

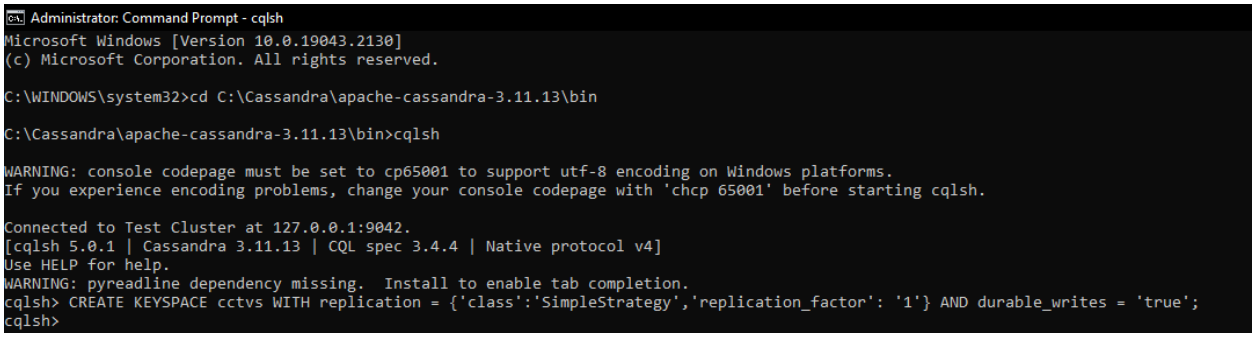


Figure 4.4: Create cctv Keyspace

Next, create a table called cctv_counts, with the following column labels seen in Figure 4.5.

Figure 4.5: Create cctv_counts Tables

Once we have configured our Keyspace and table in Cassandra, we may now proceed to startup the python scripts.

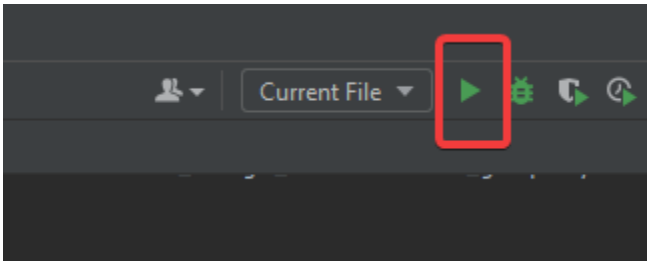


Figure 4.6: Startup Python Script

If we were to check the previously created tables, we will notice that the table has been populated with newly generated data as seen in Figure 4.7.

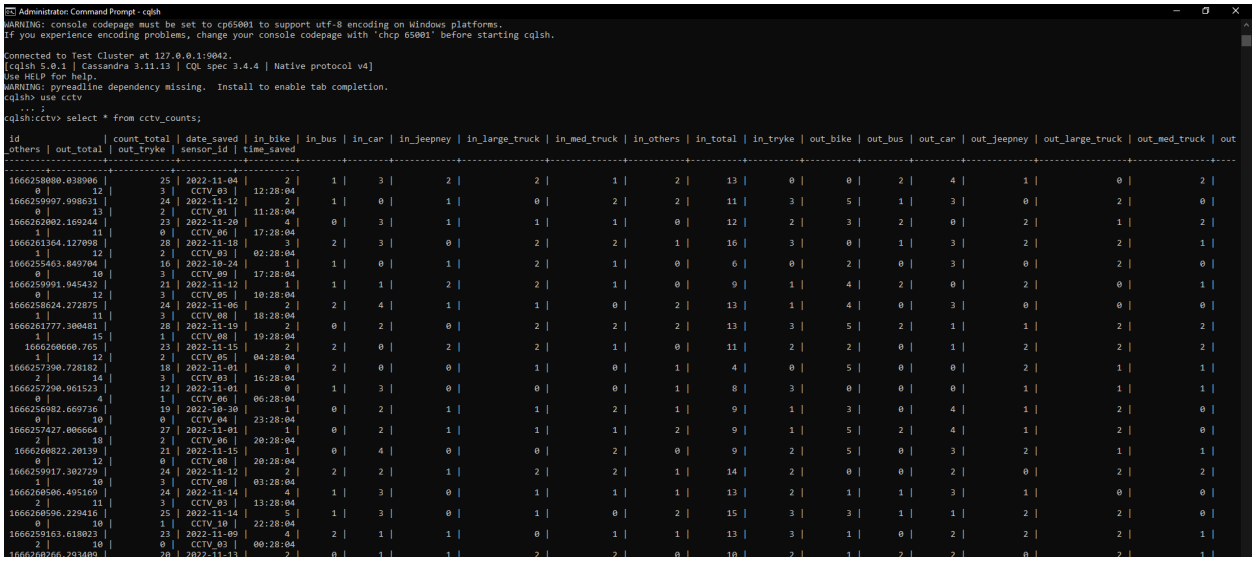


Figure 4.7: Newly created Data in Cassandra

Now that data is being generated and saved to our Cassandra keyspace, we may proceed to our Tableau project and configure our data source. First, select the cctv schema then set the cctv_counts table as the data source. Lastly, drag and drop the table to our Tableau project to start.

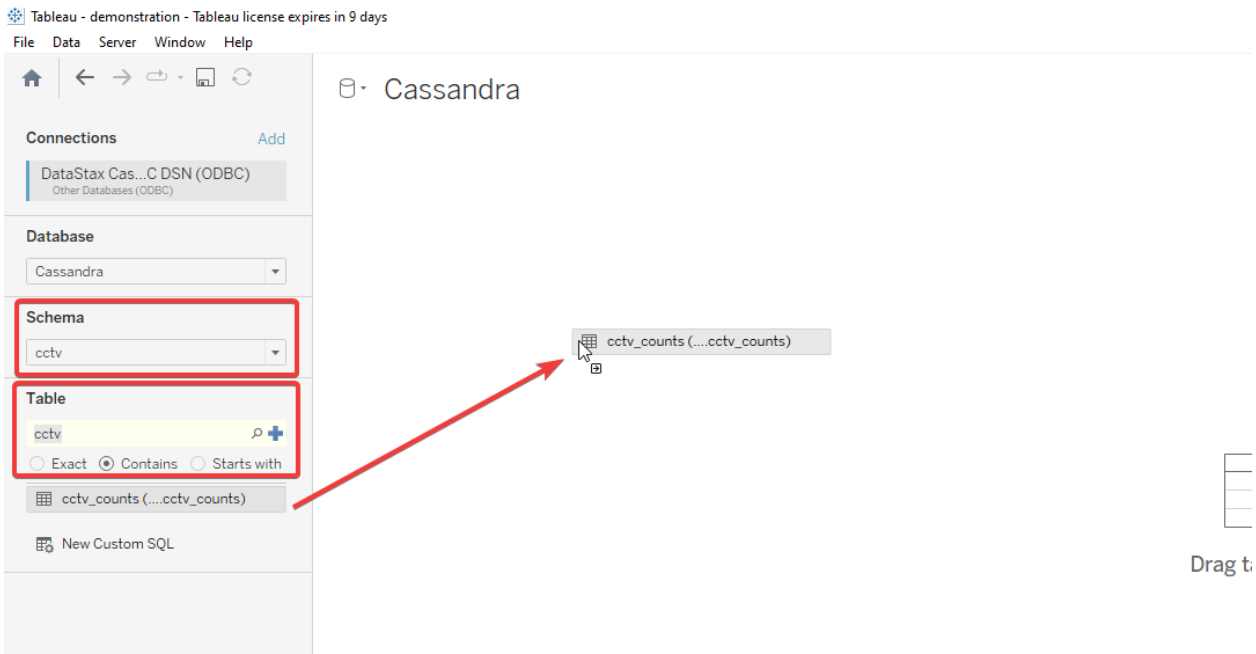


Figure 4.8: Configuring Data Source of Tableau

Once we have added the Cassandra data source, we may view the data generated in real time from the table created.

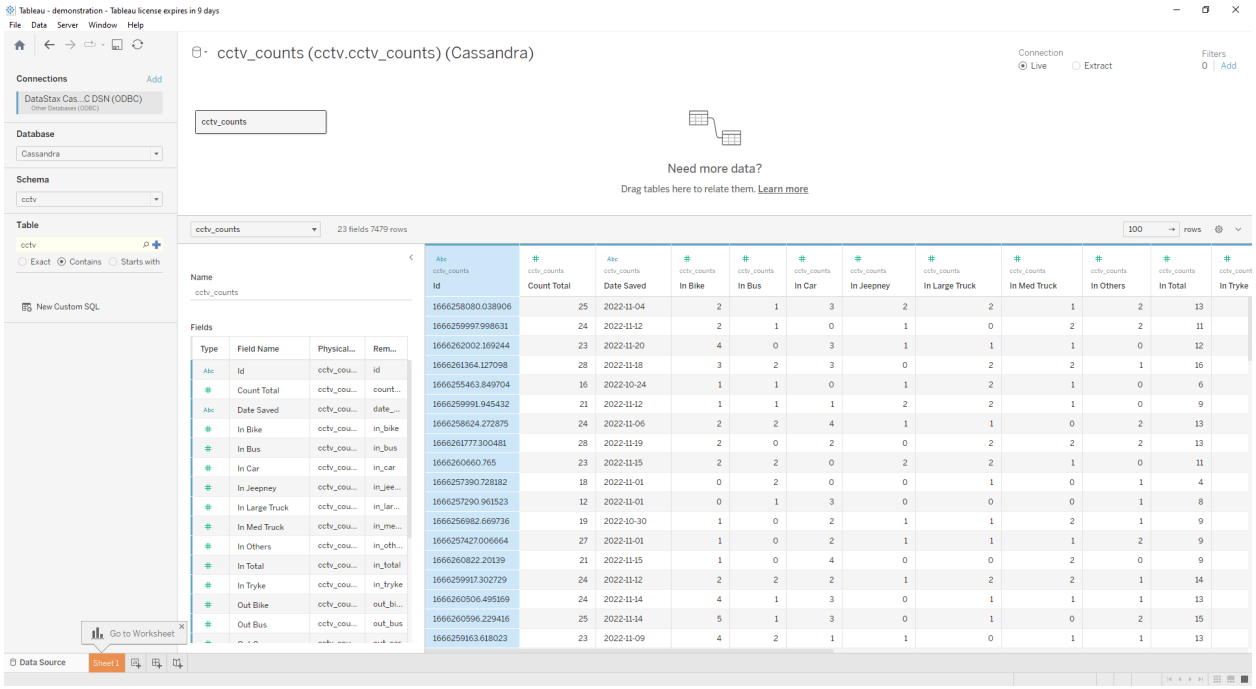


Figure 4.9: Viewing Data Table

5. Real-time Data Analytics

Since we have created both in and out instances for each vehicle, each vehicle is tagged as either in_bike or out_bike. Since we will be calculating the total sum for each vehicle, we will be combining these in and out instances to one vehicle column. By selecting the “in_bike” column, right click and create a calculated field.

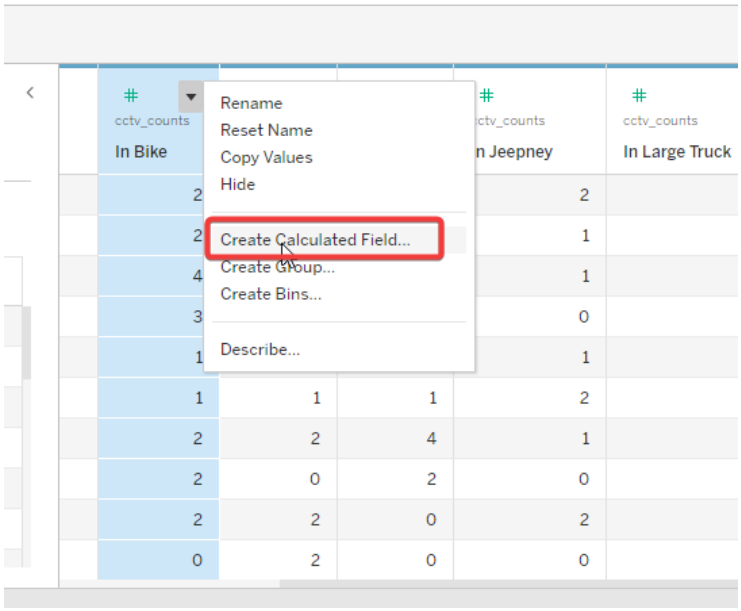


Figure 5.1: Create Calculated Field

Within this window, we will combine both the in_bike and out_bike columns into one “Bike” column to make it easier for us to drill down our data per vehicle afterwards. After configuring the combined column, we may proceed to do the same step for all the vehicles found from our dataset.

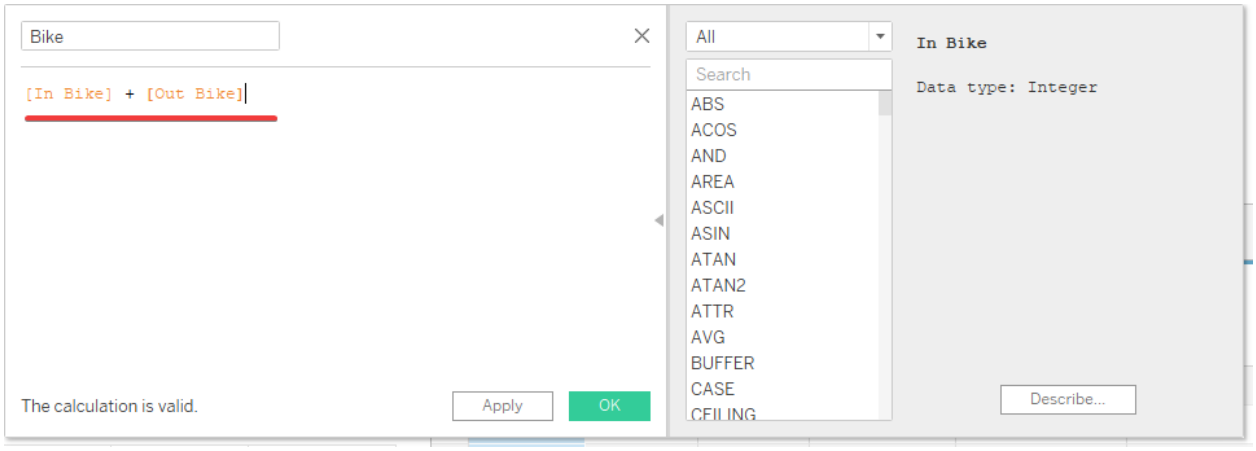


Figure 5.2: Combined Columns

After Combining th in and out variants of our vehicles into one vehicle column, it should look somewhat like Figure 5.3 as seen below.

cctv_counts In Bike	Calculation Bike	cctv_counts In Bus	Calculation Bus	cctv_counts In Car	Calculation Car	cctv_counts In Jeepney	Calculation Jeepney	cctv_counts In Large Truck	Calculation Large Truck	cctv_counts In Med Truck	Calculation Medium Truck	cctv_counts In Others
2	2	1	3	3	7	2	3	2	2	1	3	2
2	7	1	2	0	3	1	1	0	2	2	2	2
4	7	0	2	3	3	1	3	1	2	1	3	0
3	3	2	3	3	6	0	2	2	4	2	3	1
1	3	1	1	0	3	1	1	2	4	1	1	0
1	5	1	3	1	1	2	4	2	2	1	2	0
2	6	2	2	4	7	1	1	1	1	0	0	2
2	7	0	2	2	3	0	1	2	4	2	4	2
2	4	2	2	0	1	2	4	2	4	1	3	0
0	5	2	2	0	0	0	2	1	2	0	1	1
0	0	1	1	3	3	0	1	0	1	0	1	1
1	4	0	0	2	6	1	2	1	3	2	2	1
1	6	0	2	2	6	1	2	1	3	1	1	2

Figure 5.3: Data Table

Since we will be creating a filter that would allow the presenter to select a vehicle for drill down, we would first need to create a parameter. From the data tab, create a parameter for the vehicle drill down.

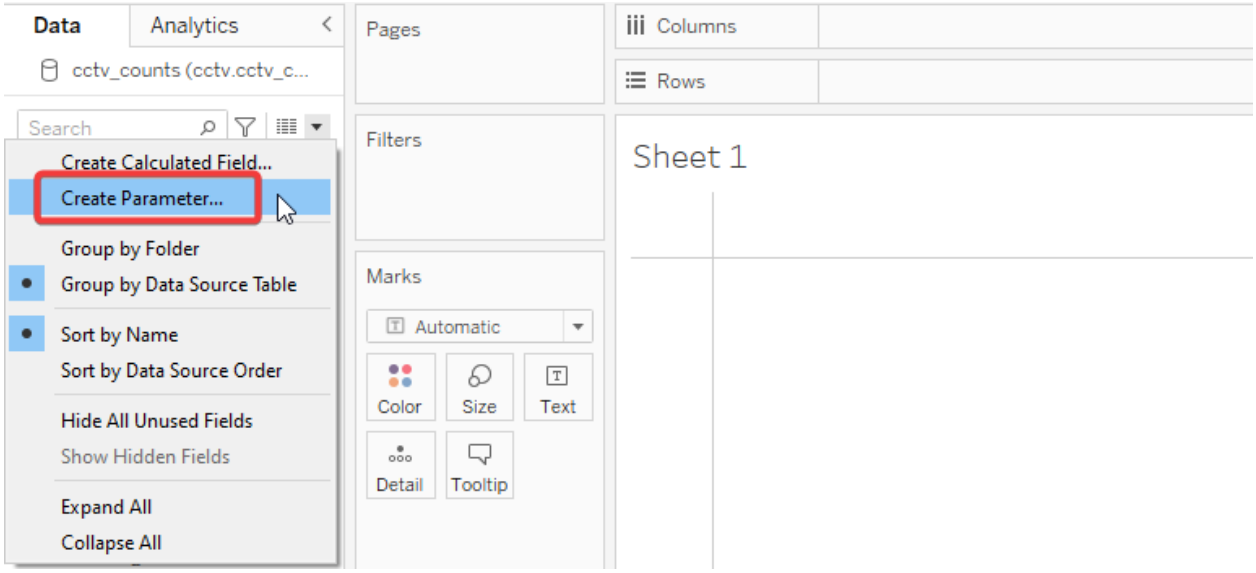


Figure 5.4: Vehicle Parameter

In the Parameter configuration window, we may set the name to Vehicle Parameter then set the data type to string. After this, we would need to manually add the Vehicle names as the Value and Display for our drop down drill down parameter.

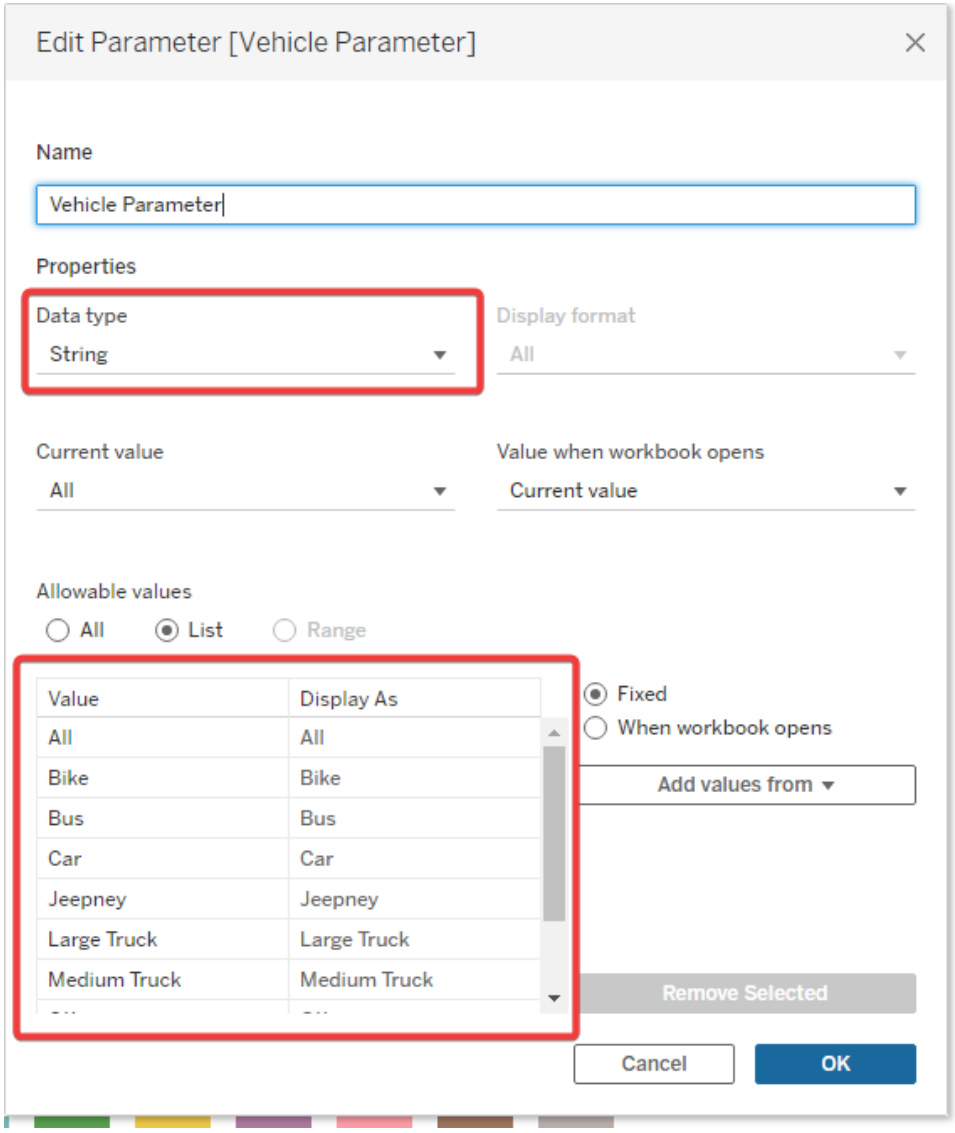


Figure 5.5: Parameter Configuration

After we have added our parameter, we will then need to create a calculated field that would allow us to drill down our Cassandra data by the vehicle type.

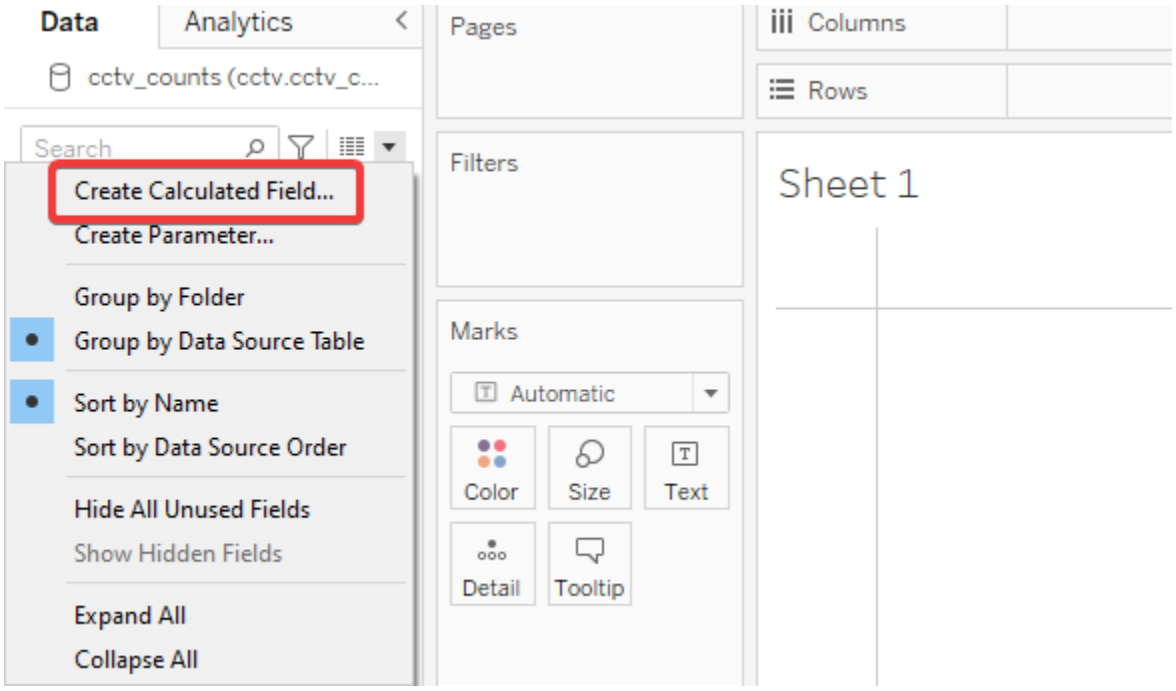


Figure 5.6: Create Calculated Field

Within the calculated field, we will use case statements that would read the data of each vehicle individually based on what the user sets it as.

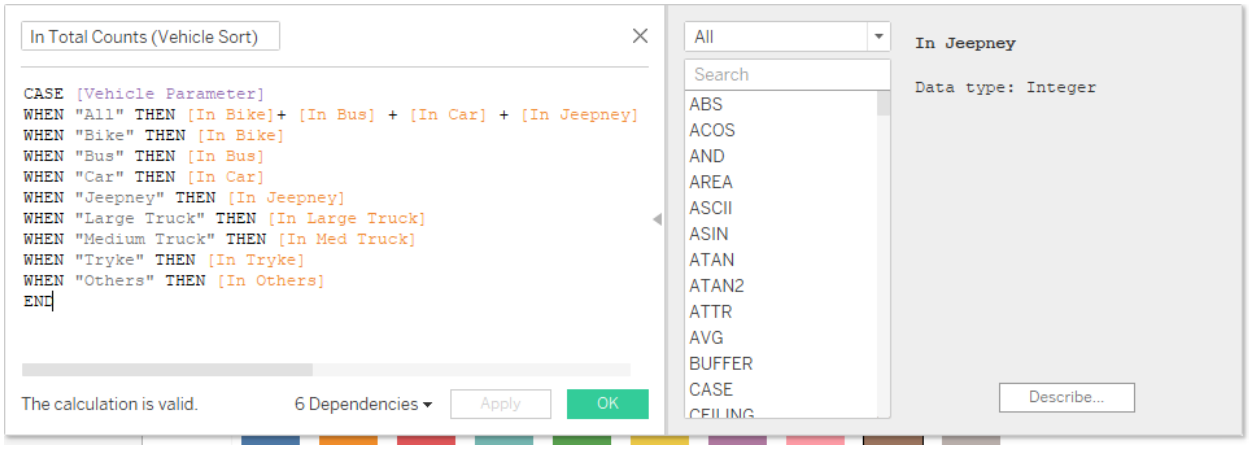


Figure 5.7: Calculated Field Configuration

Create total counts of vehicles per CCTV

The graph from Figure 5.8 below was created by setting the Sensor ID as the column with the Total Count of Vehicles as its row. The sensor ID was also set as a filter. On the other hand, the vehicles parameter drop down menu was allocated to the right side along with the sensor ID legends. This vehicle parameter drop down would allow users to select which vehicle count they want to see per cctv sensor. Each CCTV sensor was color coded as well with its legend also added to the right-hand side bar.

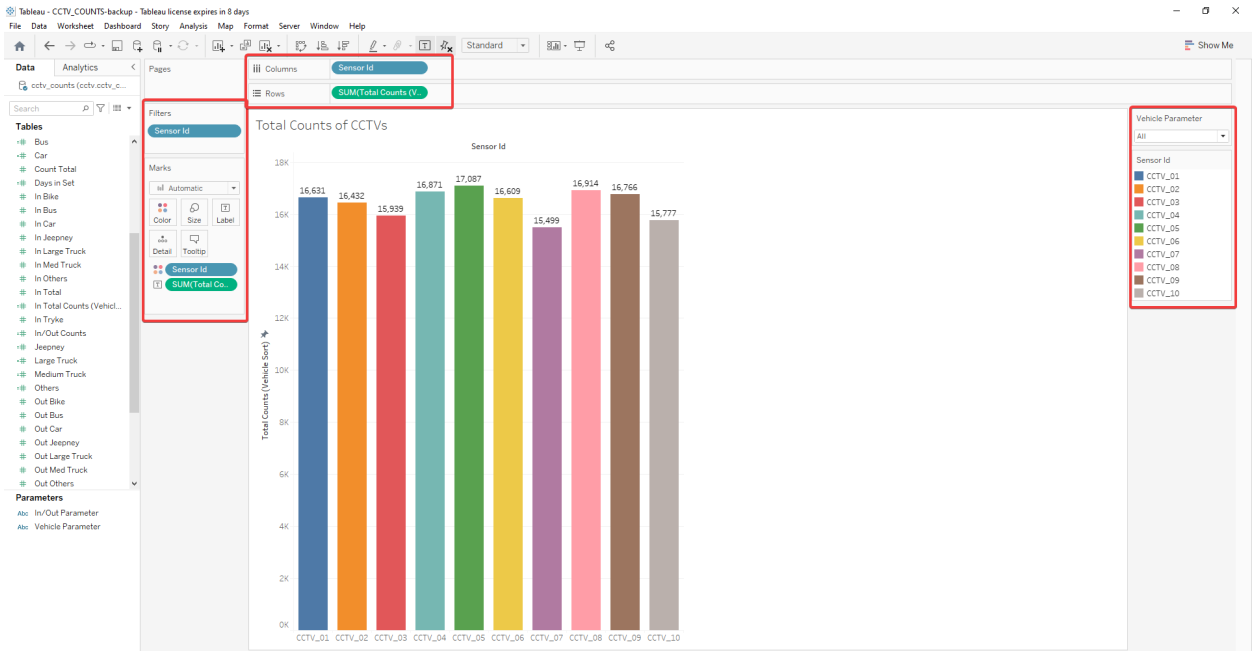


Figure 5.8: Total Counts of Vehicles per CCTV

Upon selecting the vehicle parameter drop down, you will be greeted with the previously configured vehicle types. This drop down consists of all the vehicles found from our dataset. This way, we may drill down and see how many total bikes or cars were counted by each CCTV sensor. In Figure 5.9 below, the vehicle selected was “bike” which displayed the total number of bikes counted by each CCTV. If the user were to select another vehicle, the values of the bar graph would change as well.

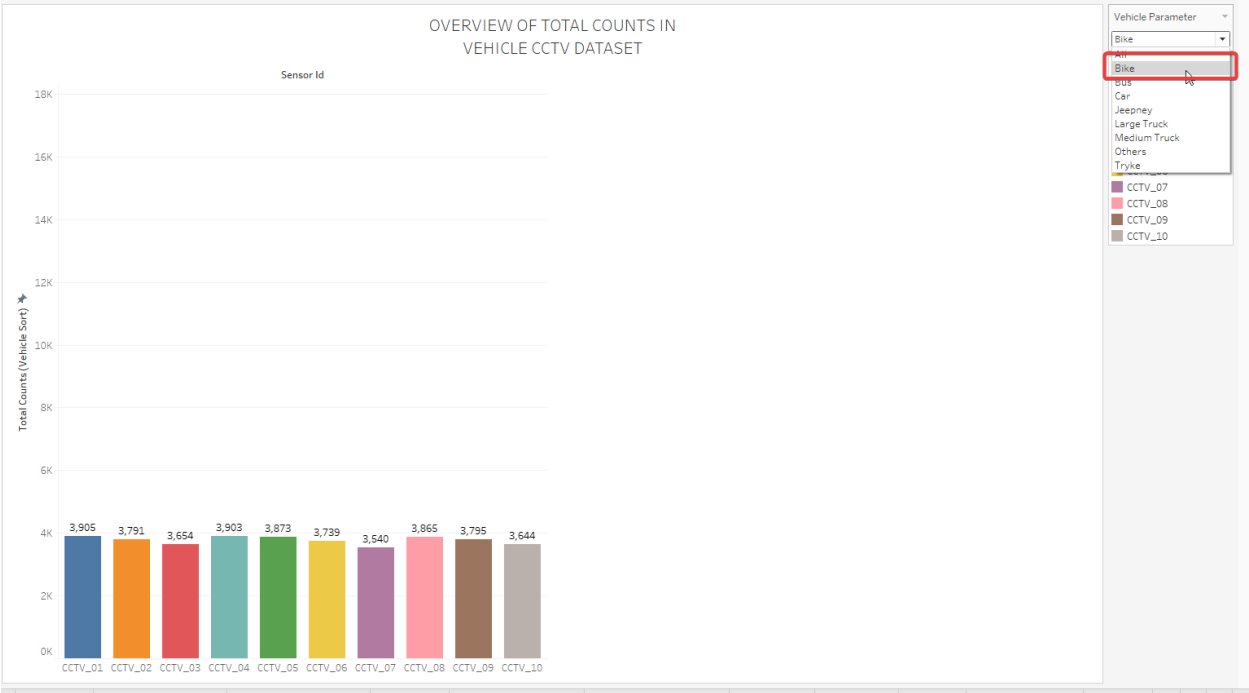


Figure 5.9: Total Counts of Vehicles per CCTV (part 2)

The next bar graph we will create will emphasize the amount of vehicles counted incoming and outgoing. This would need two bar graphs. One bar (Figure 5.10) graph will display the amount of vehicles incoming while the second graph (Figure 5.11) will display all the vehicles outgoing. Similarly to the previous graph, the type of vehicles may be drilled down to determine its specific count. The same settings were used from the previous bar graph. However, the total incoming and outgoing cctv count was selected as its filter.

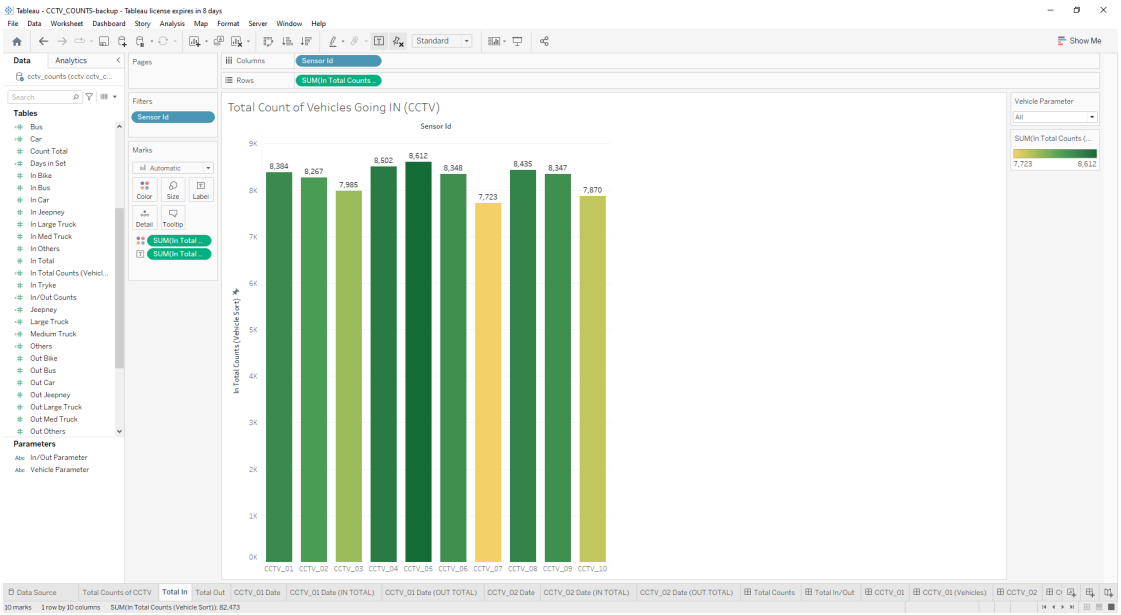


Figure 5.10: Total Counts of Vehicles per CCTV (INCOMING)

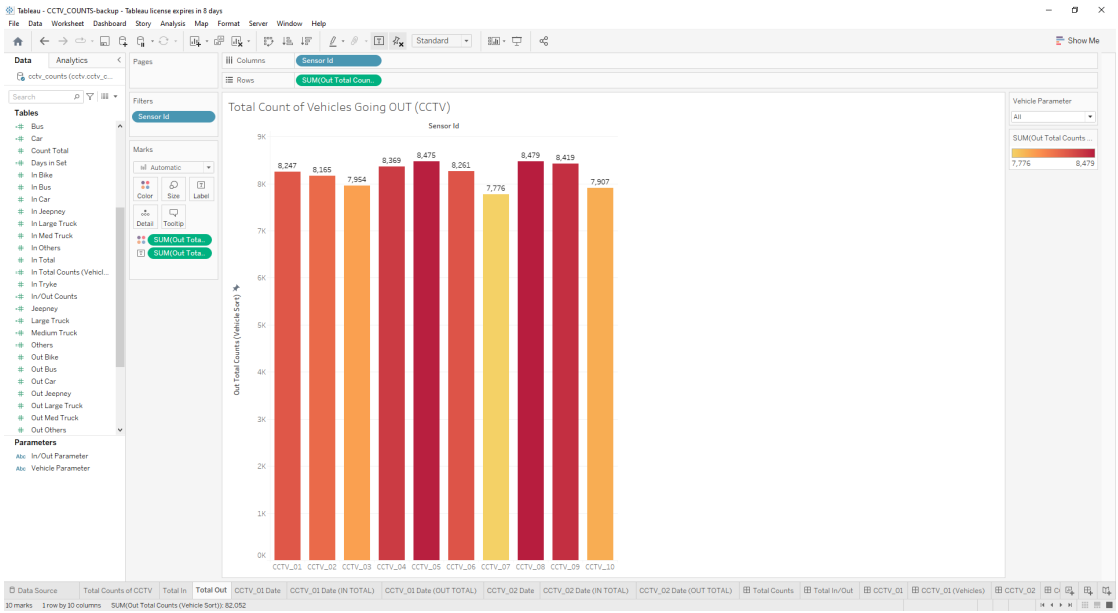


Figure 5.11: Total Counts of Vehicles per CCTV (OUTGOING)

Once we have completed the first two pages of showing all the data captured by each CCTV, we may proceed to create individual pages for each individual CCTV and display their data. This data includes the day and hour a vehicle was counted. In order to drill down our data from days to hours, we will need to create a custom date for Day. To do this, we will need to right click on our date measure and create a custom date as seen in Figure 5.12 below.

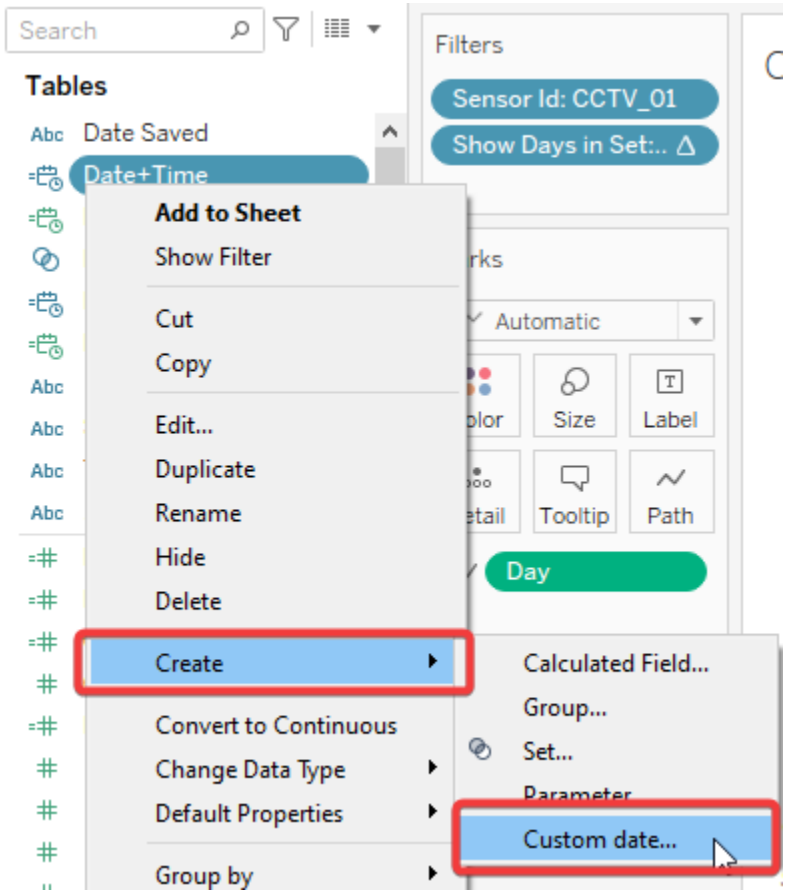


Figure 5.12: Create Custom Date

Set the name to Day and check the Date Value and proceed to click OK.

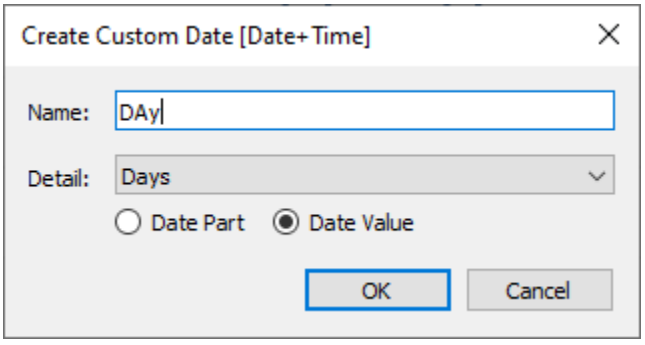


Figure 5.13: Create Custom Day

Since we will be drilling down from Day to Hours, we will also need to create a custom date for our Hours data.

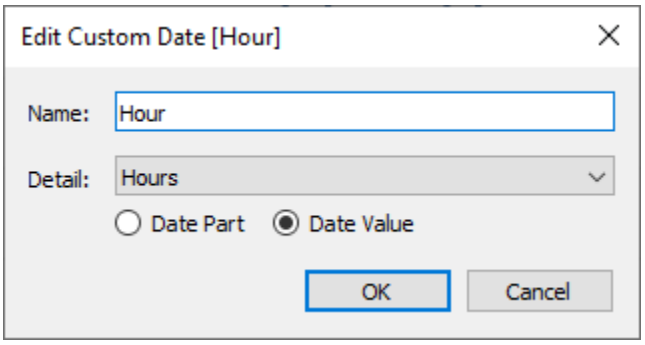


Figure 5.14: Create Custom Hour Date

Next, we will proceed to create a Day Set. From the sidebar to the left of the main window, create a set and name it as Day Set. Next, check “Select from list” and proceed to click OK. This would display all the listed days from our dataset which would allow us to drill down from it

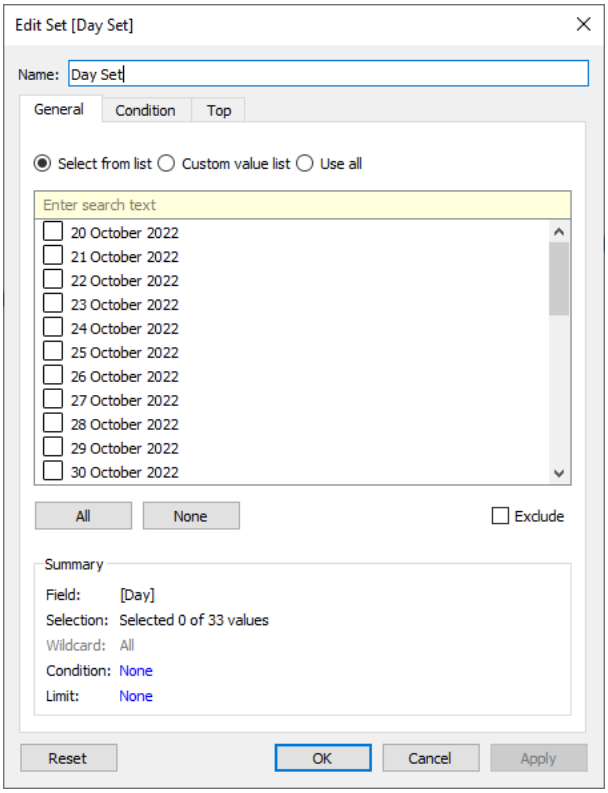


Figure 5.15: Create Day Set

Next, create a Calculated Field with these commands. Using this if, else statement, we may drill down and up from day to hour by either clicking on the graph or by selecting the date on the side bar.

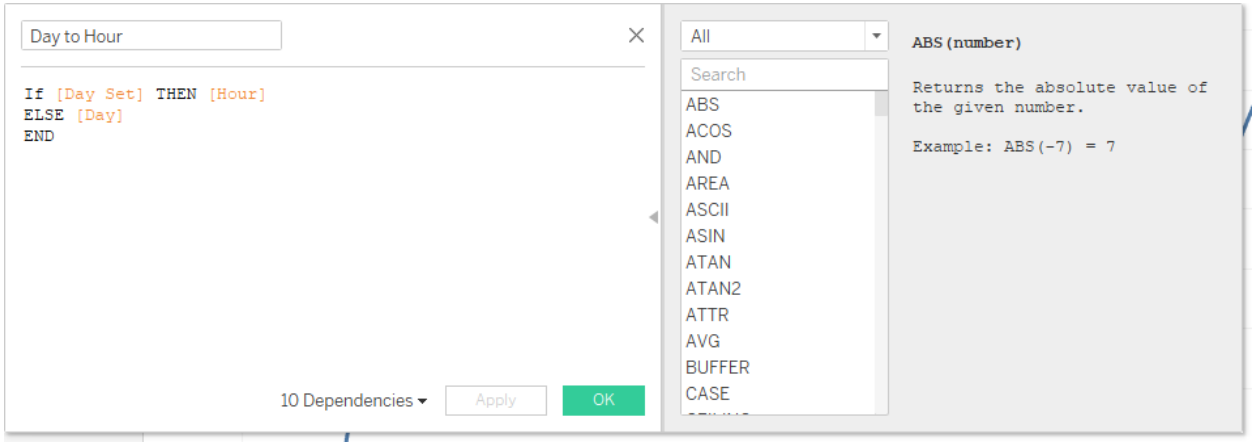


Figure 5.16: Create Day to Hour Calculated Field

Next, we will need to create a Set Action based from the previously created Day set. Click Worksheets then click actions to open the set action settings.

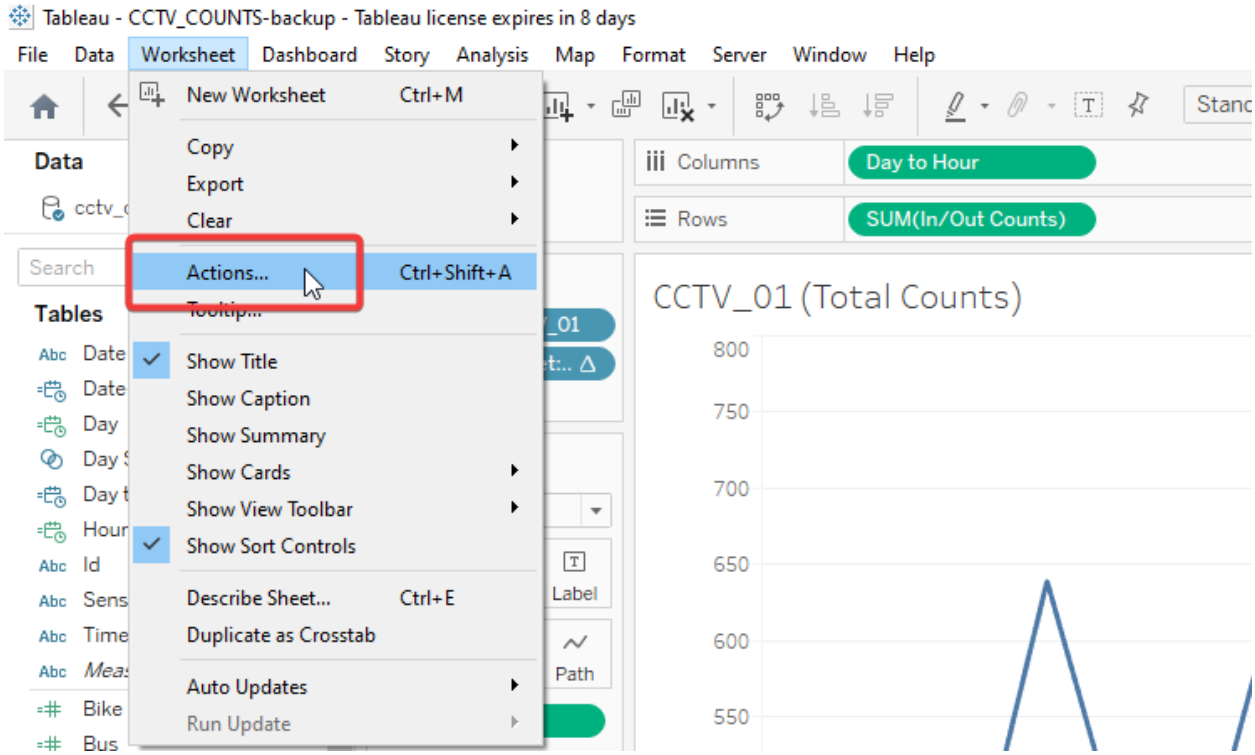


Figure 5.17: Create Set Action

As we are creating a Set Action, we will need to select the source sheet to the specific CCTV sensor sheet. Next, we will need to set the target set as the “Day Set.”Lastly, set “Assign values to Set” and “Remove all values from set.”

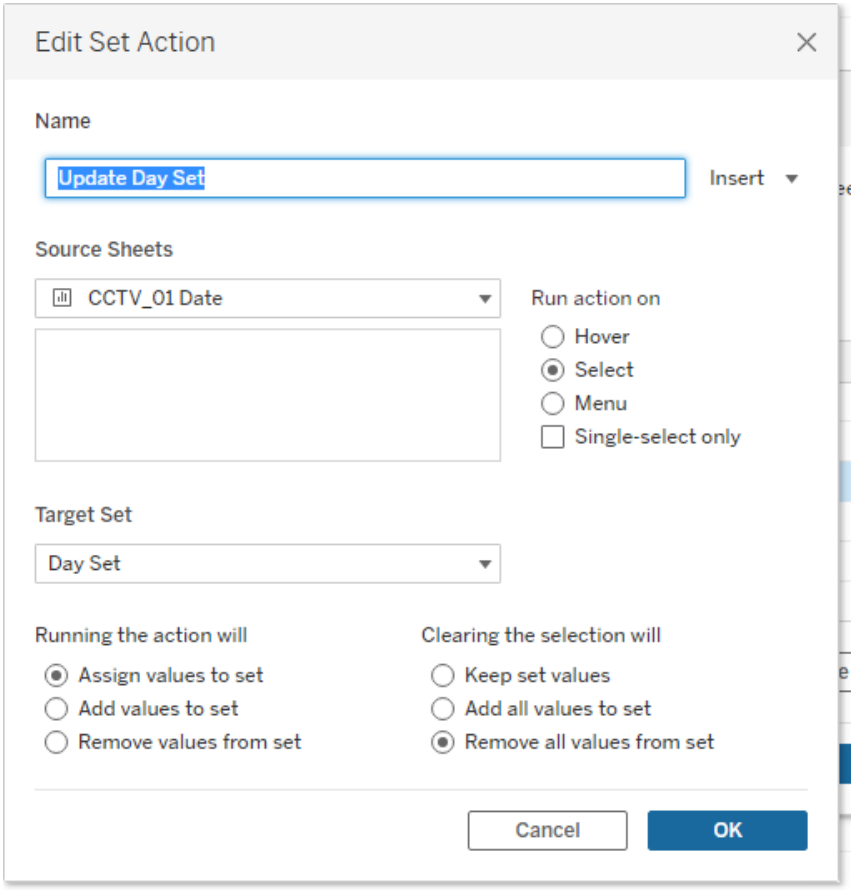


Figure 5.18: Create Action Set (part 2)

Lastly, we will create another calculated field that would change the window once we select the day. This would allow us to drill down to the hour and show ONLY the data for that hour.

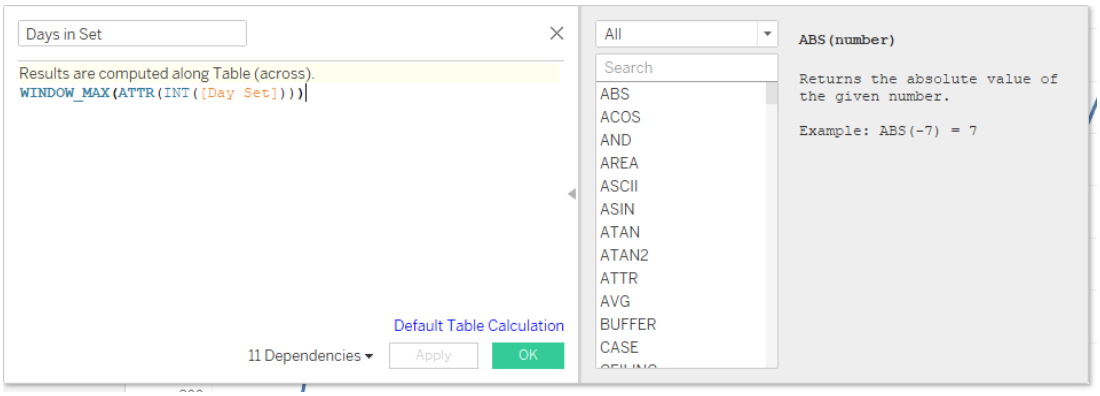


Figure 5.19: Window Calculated Field

Once we have created all the needed set actions, calculated fields, and custom dates, we may now proceed to fill up the columns and rows of our data sheet. Set the column to “Day to Hour” and the row to the sum of the IN/OUT Counter. This column/row configuration would allow us to see the data from the dates of October 20 to November 21. Next, drag our IN/OUT parameter to the right side. This parameter would allow us to drill down from the total count to the incoming and outgoing counts as well. Lastly, populate the filters with the sensor ID of our sheet. For this example, the sensor we will be using is CCTV_01. This means that we need to set the filter to ONLY CCTV_01. Next, add the “Show Days in Set” measure to the filter to allow us to drill down from day to hours.

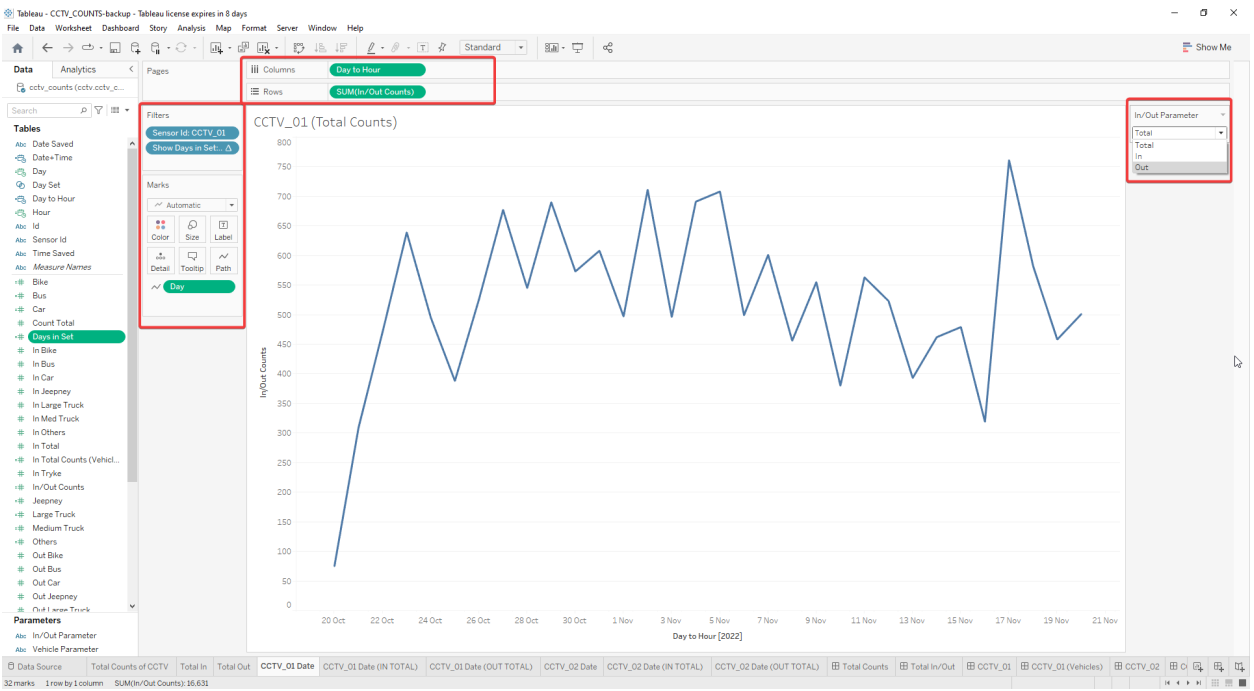


Figure 5.20: Configure CCTV_01 Main Sheet

If we were to click on any point from the graph, we should be able to drill down from the day to the hour. As seen in Figure 5.21, the user is going to select a specific segment from the line graph marked as October 23, 2022. Once clicked, the line graph changes and displays only the recorded vehicles within that day in the form of hours as seen in Figure 5.22. In Figure 5.22, you may see all the recorded counts from 12am of that day to 12am of the next day. This wide range visualizes the traffic from that entire day. Using the In/Out parameter, we are also able to adjust whether we want to see data of incoming, outgoing or the total amount of cars the CCTV sensor was able to count.

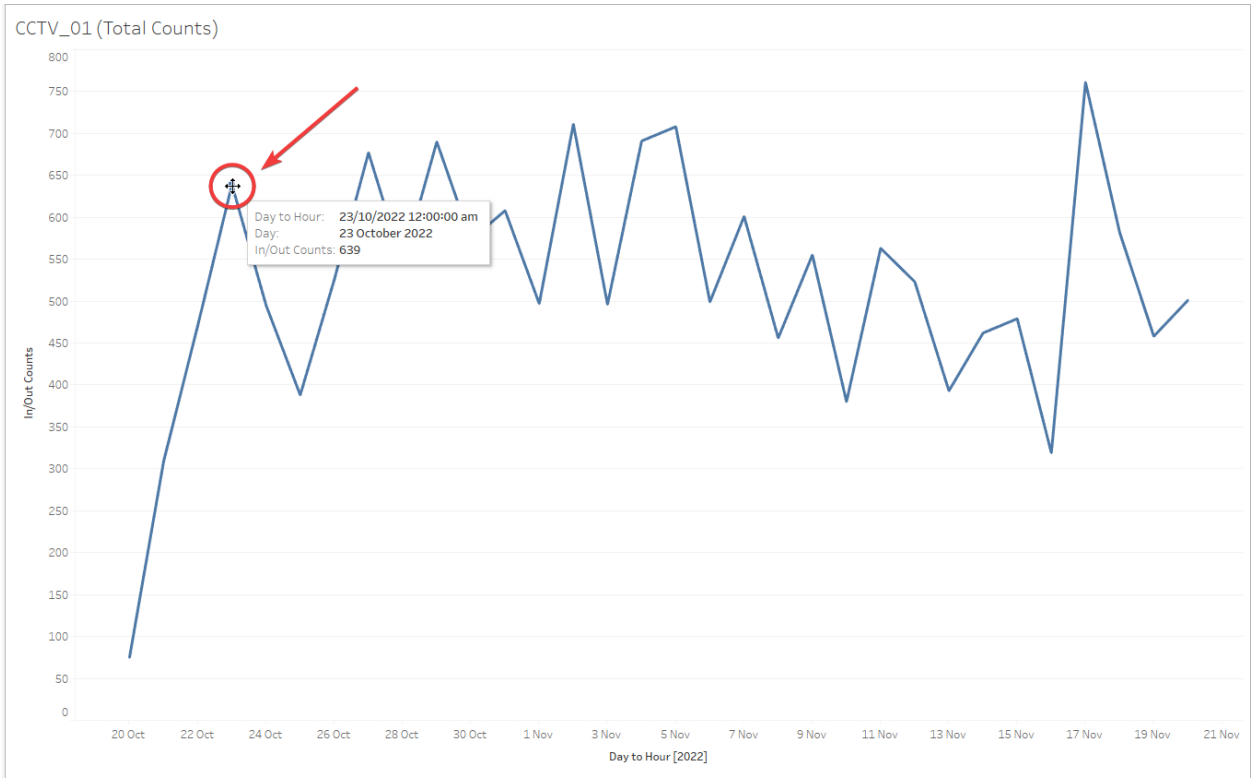


Figure 5.21: Drill Down Day

Figure 5.22: Drilled Day to Hours

We will now apply those same settings and calculated fields to two more sheets for CCTV_01. However, this time, we won't be displaying the total counts from October 20 to November 21. Instead, we will be creating two separate line graphs for Incoming and Outgoing vehicles. We are also able to select the type of vehicle from the vehicle parameters to the right side of the data sheet. If the line graph is GREEN, it is indicated as incoming, while RED is outgoing. We will be repeating these steps for all the other CCTV sensors from 1 to 10 as well.

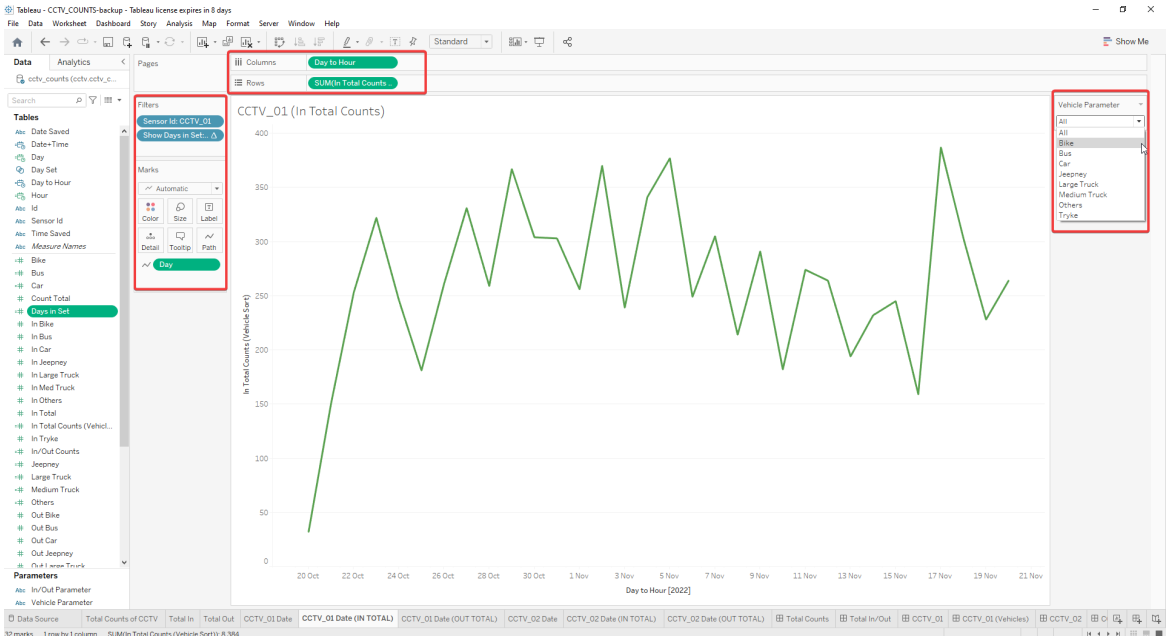


Figure 5.23: Incoming Vehicles for CCTV_01

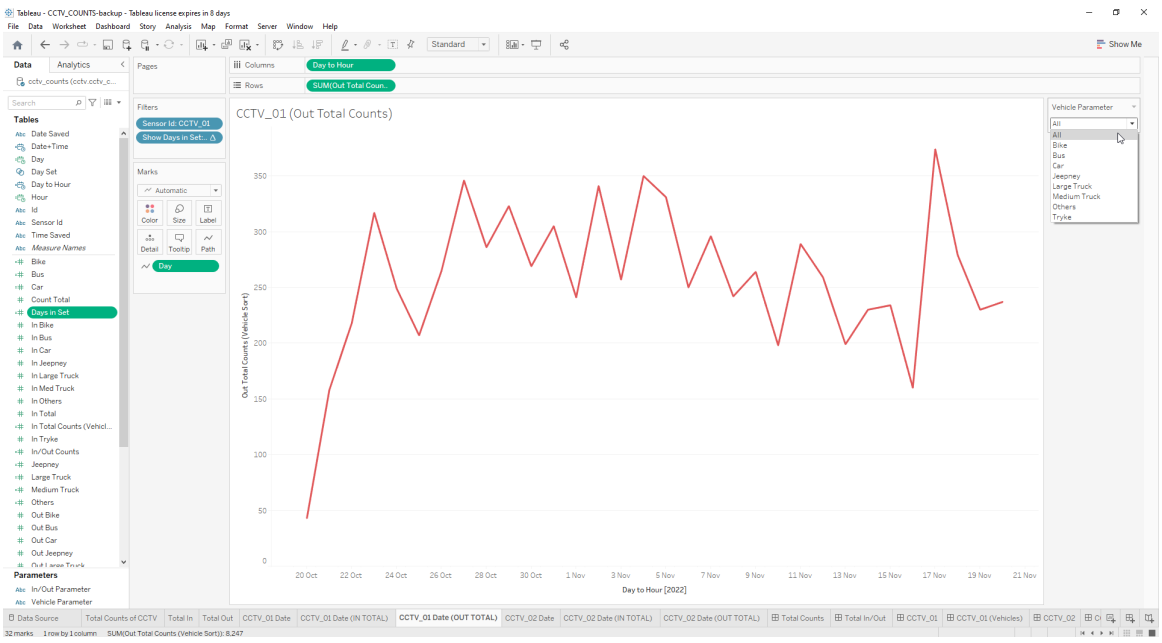


Figure 5.24: Outgoing Vehicles for CCTV_01

6. Create Dashboard Pages

Once we have created our sheets, we may now proceed to create dashboards for all our data visualizations. To do this, click on the icon on the lower right corner of your Tableau Project as seen in Figure 6.1

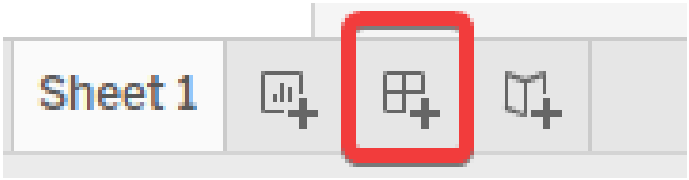


Figure 6.1: Create Dashboard

From here, we will drag the needed sheet to the dashboard as seen in figure 6.2

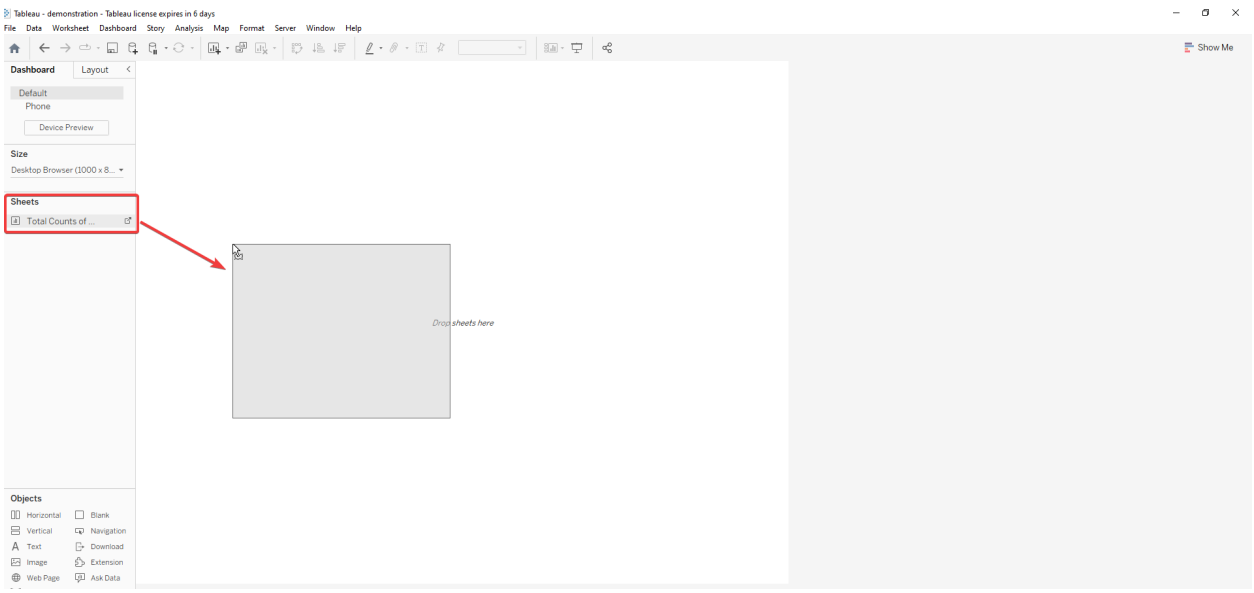


Figure 6.2: Add Data Sheet to Dashbaord

Once we have added our sheet for Total Counts per vehicle CCTV Dataset to our Page 1 Dashboard, we will need to add the sensor ID filter. Following the steps from Figure 6.3 below, insert the sensor ID filter to the right side of the bar graph. This filter would allow us to select specific CCTVs and view their total, incoming and outgoing vehicle counts.

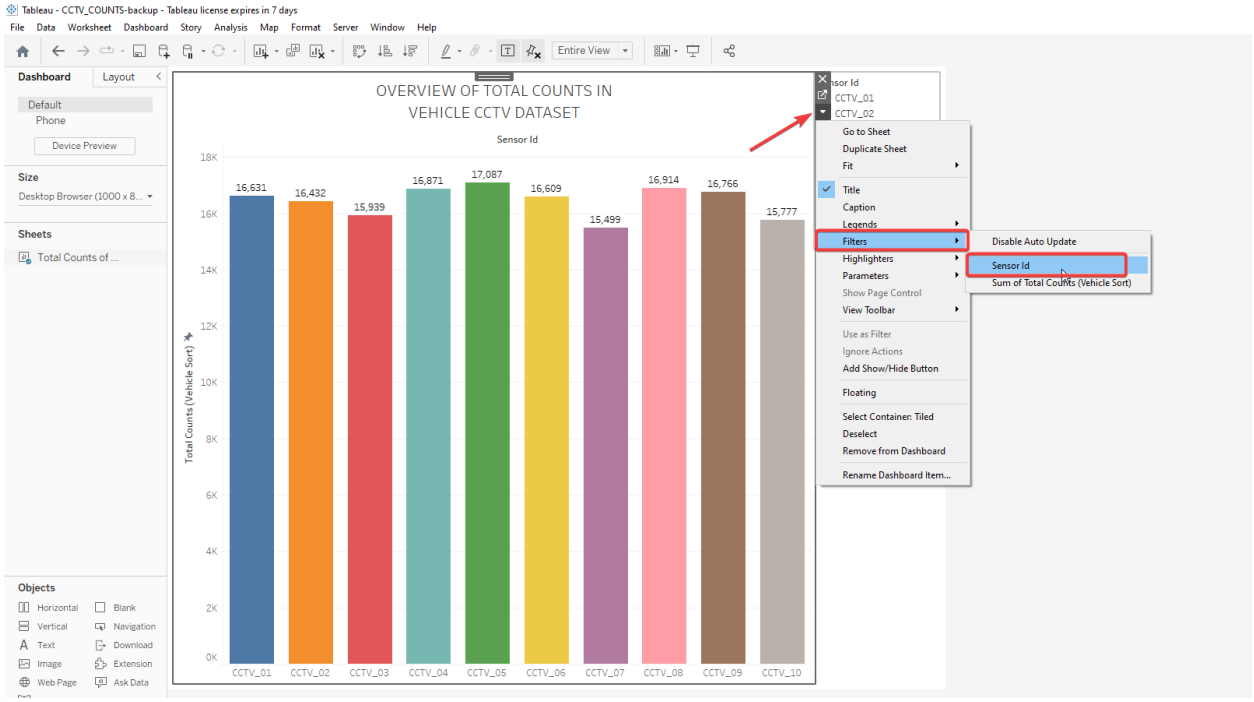


Figure 6.3: Add Sensor ID filter

As seen in Figure 6.4, the drop down vehicle menu allows us to select what type of vehicle to drill down to. We may also select which CCTV sensors we want to keep in the display.

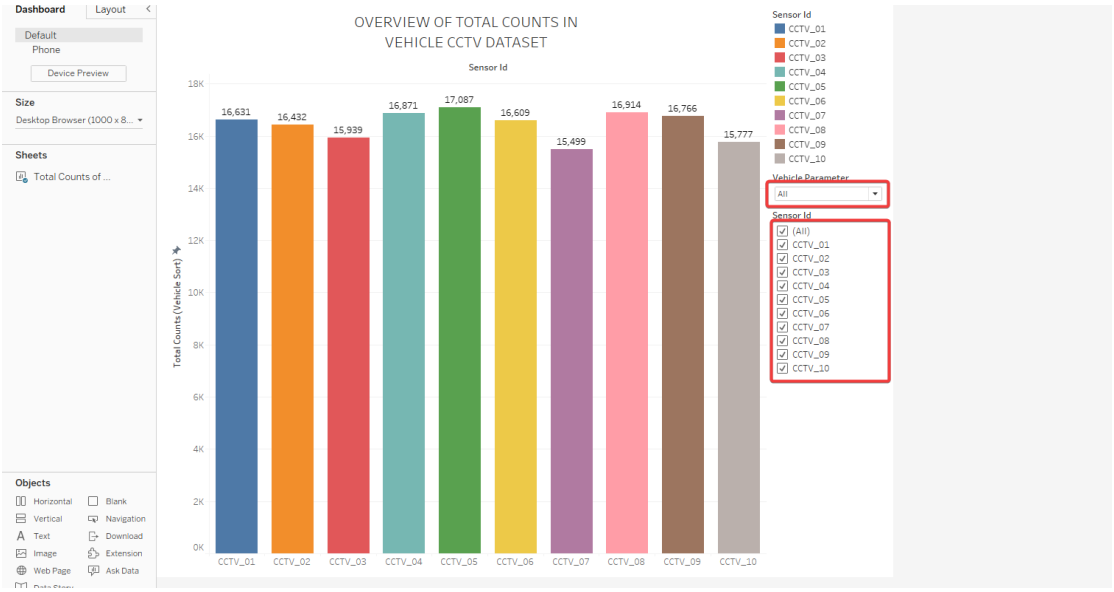


Figure 6.4: Dashboard Page 1

In Figure 6.5, we have drilled down our data and selected the Bike as our vehicle parameter. Since bikes are only a small percentage of the total number of vehicles counted by the CCTV sensors, the data count is smaller than its total. It may be seen as well that some CCTVs are not selected. By selecting some CCTV sensors, we may remove CCTV sensors that we may not need to view currently.

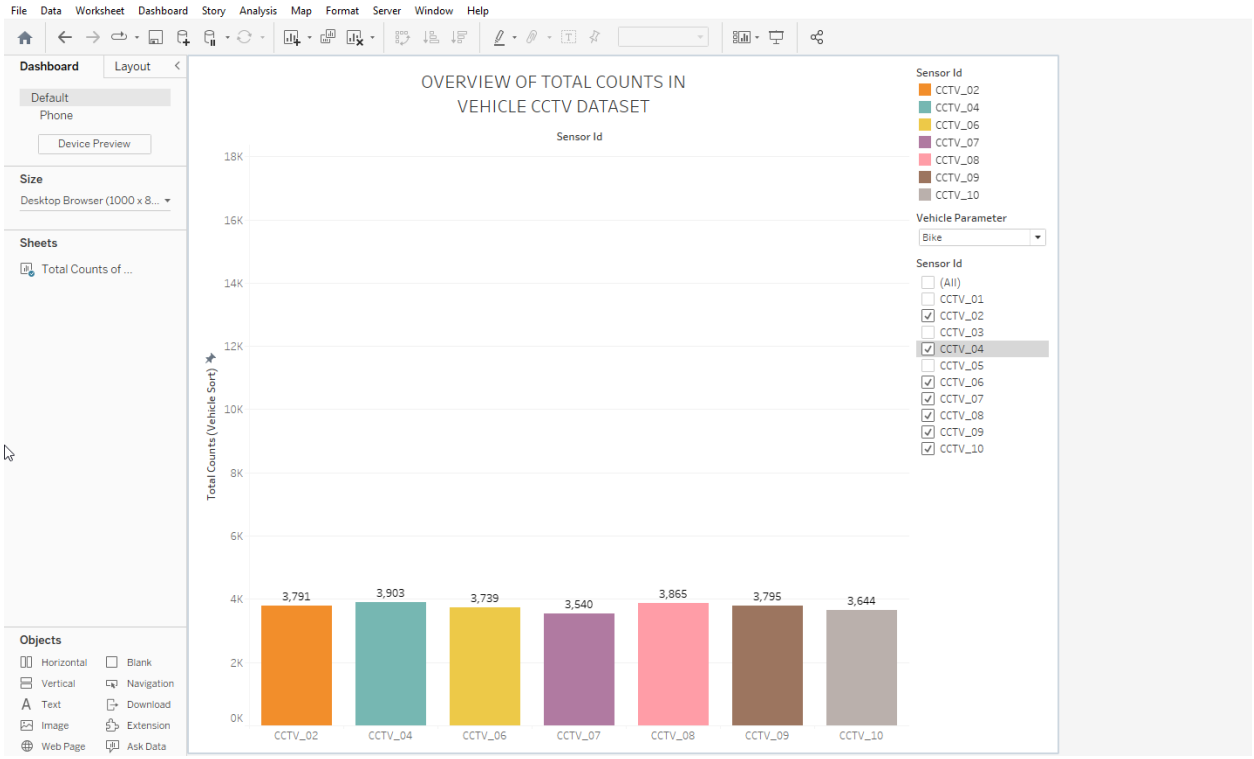


Figure 6.5: Drilled Down Dashboard Page 1

Applying the same parameters to our second dashboard page, we can view the incoming and outgoing vehicles while adding a drill down for the specific vehicle and CCTV sensor. This second page of our dashboard will provide data for both incoming and outgoing vehicles.

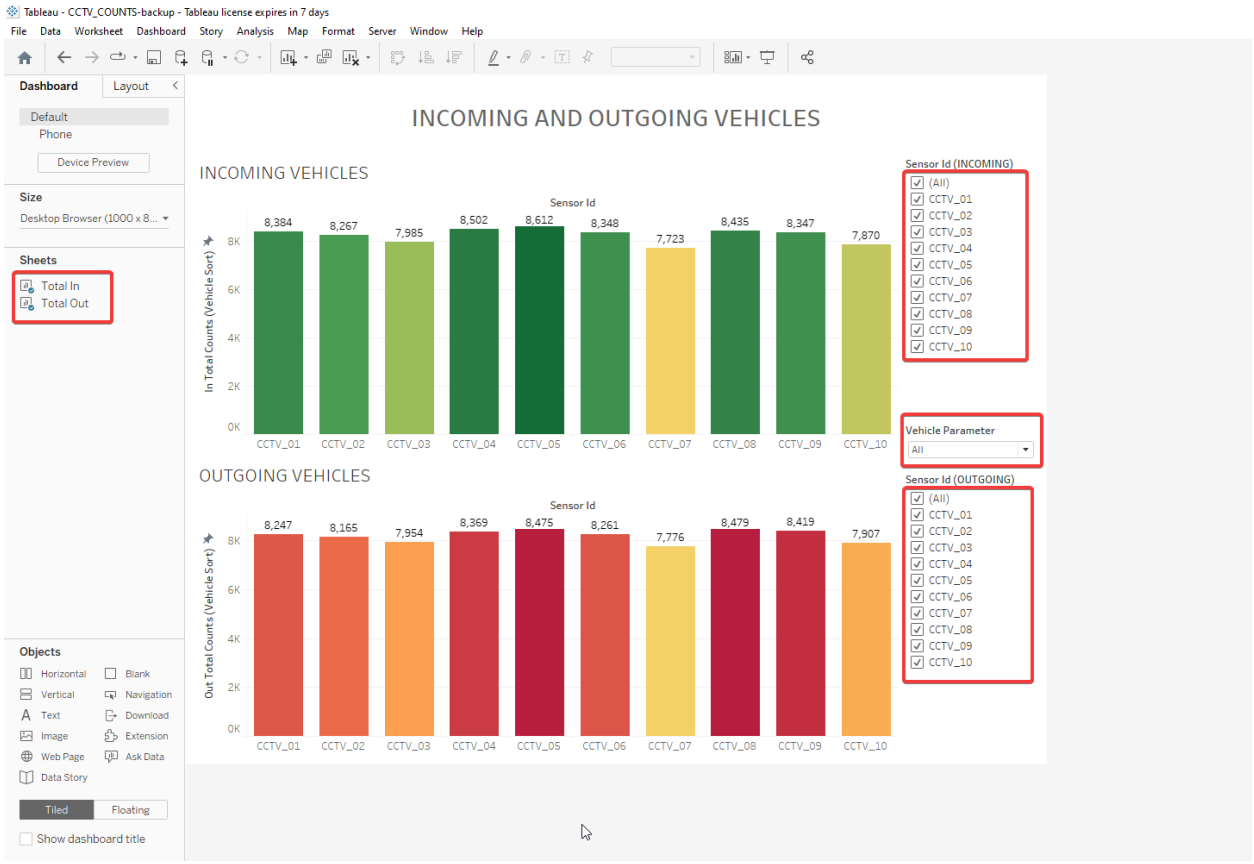


Figure 6.6: Dashboard Page 2

As seen in Figure 6.7 below, we may select which type of vehicle we want to view from the Vehicle Parameters drill down. We may also select which CCTV sensors we want to view from the bar graph as well. From this example, the vehicle we have chosen is the Bike with CCTVs 04 to 10 enabled.

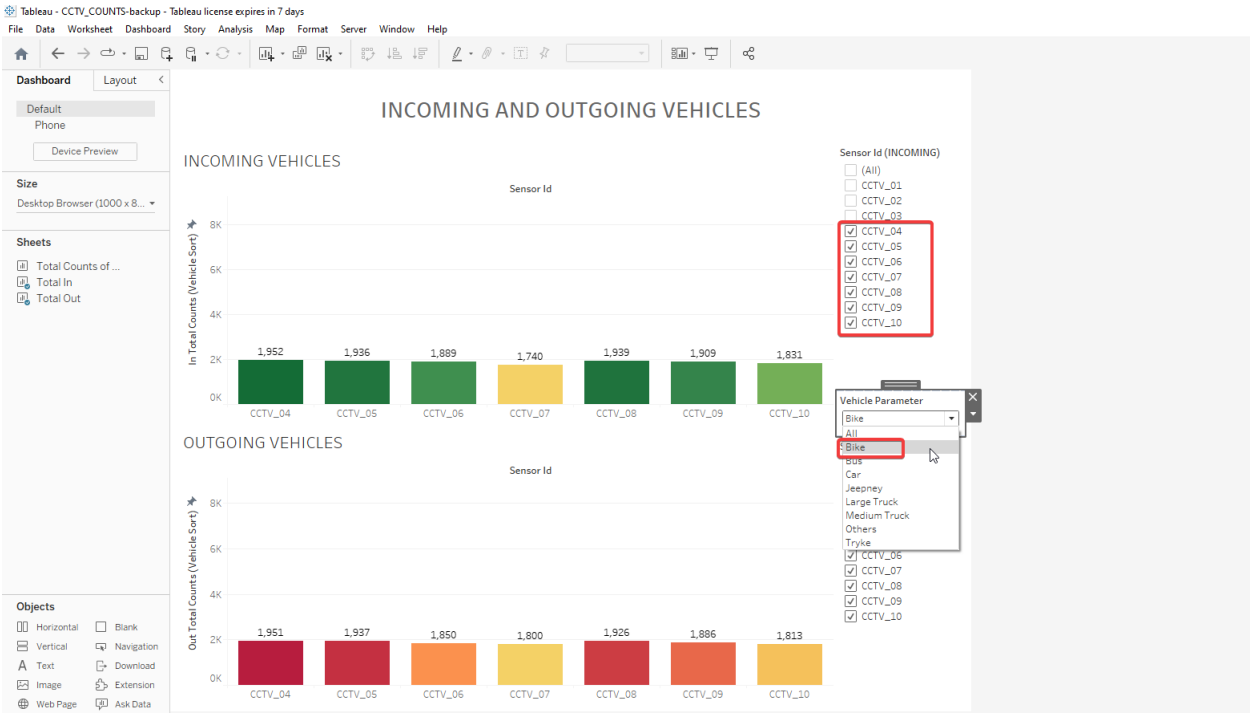


Figure 6.7: Drilled down page 2

After creating our two main pages, we may now proceed to create two dashboards for each individual CCTV sensor. The first dashboard will display a line graph with each day the CCTVs have been counting vehicles. Our dataset only counted days from October 20 to November 21. These dates may be seen on the Day set found on the right side of the dashboard. If we were to select a day from the Day set, it will drill down from day to hour. We have also added an In/Out parameter that would allow us to switch between the total, incoming and outgoing count of vehicles for CCTV_01.

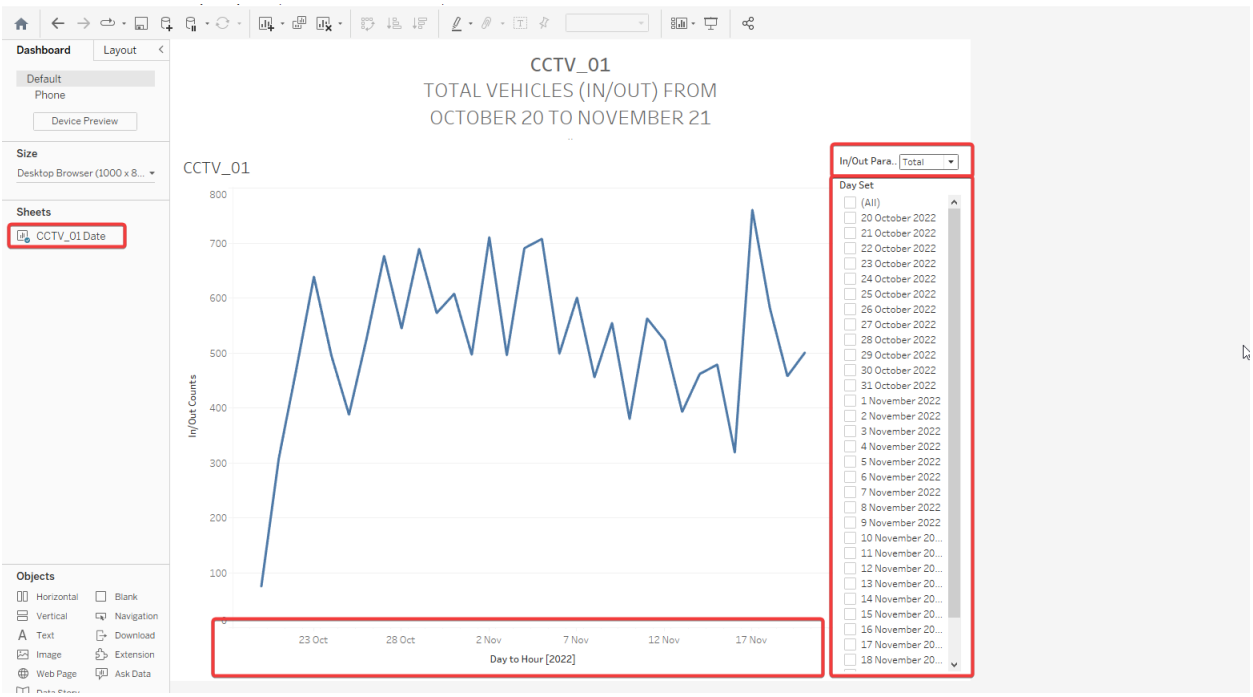


Figure 6.8: Individual CCTV_01 Dashboard (page 1)

In Figure 6.9, we have selected the day of October 23, 2022. This immediately drilled down and transformed the line graph from Days to Hours. What is now displayed in the dashboard is the Total count of vehicles that CCTV_01 has counted throughout the entire day of October 23, 2022.

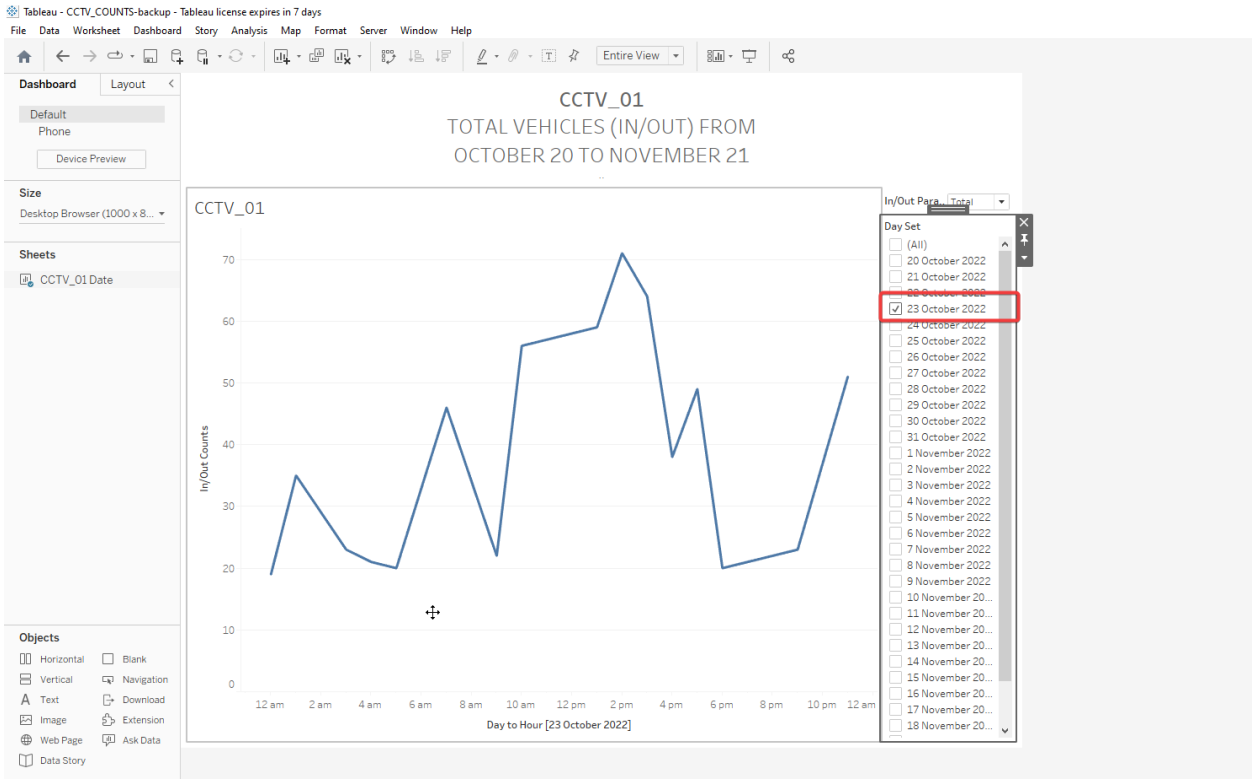


Figure 6.9: Individual CCTV_01 Dashboard Drilled Down

Lastly, we may also drill down the total count to either incoming or outgoing vehicles. In Figure 6.10 below, we have selected “In” which would drill down the total count to the amount of incoming vehicles counted by CCTV_01.

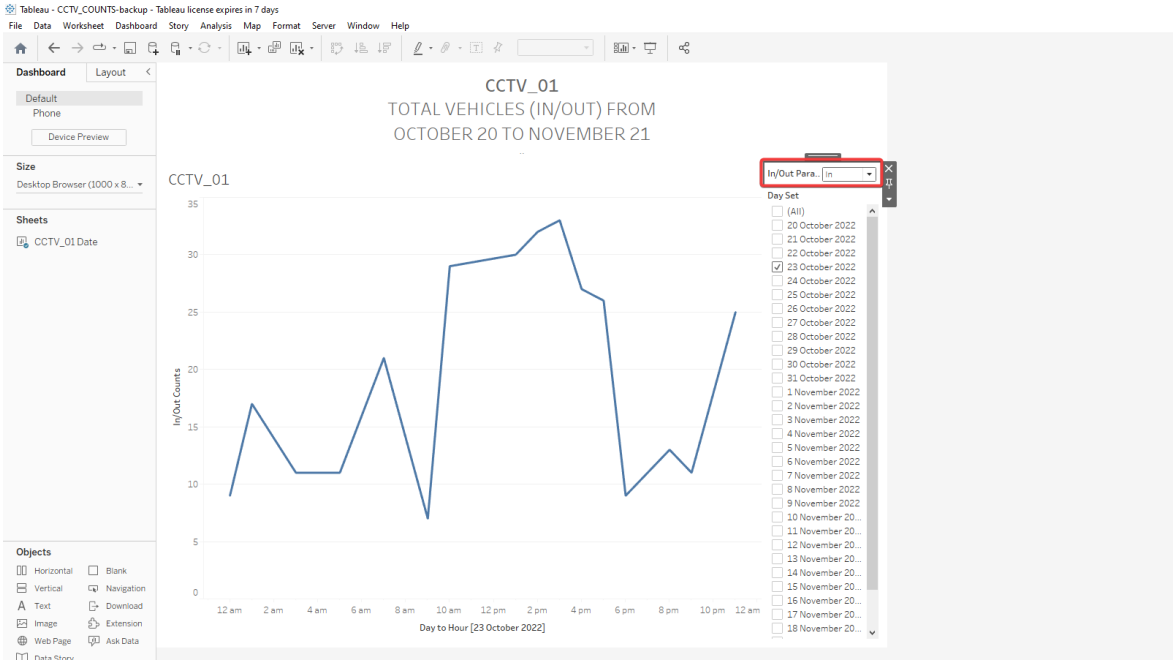


Figure 6.10: Individual CCTV_01 Dashboard Drilled down 2

Applying these same settings to our second figure for CCTV_01, we will add both incoming and outgoing vehicle count line graphs as seen in Figure 6.11. This would allow us to view the differences between the incoming and outgoing vehicles.

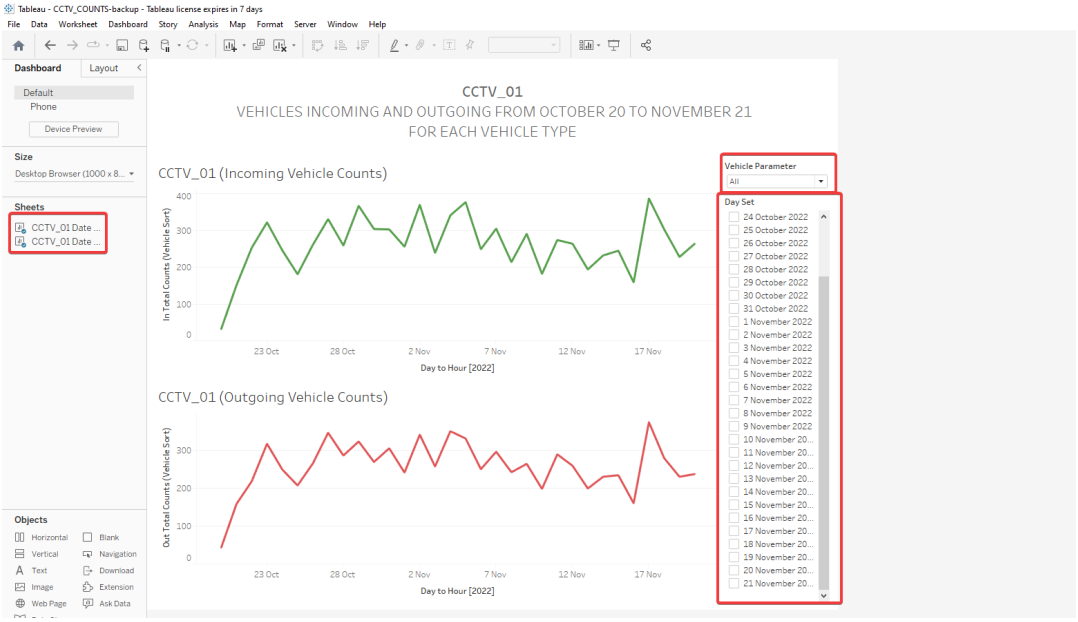


Figure 6.11: Individual CCTV_01 Dashboard (page 2)

Selecting November 1 as our day, we can see that the line graph has drilled down from days to hours. The data from the line graph is only displaying incoming and outgoing vehicle counts within the day of November 1.

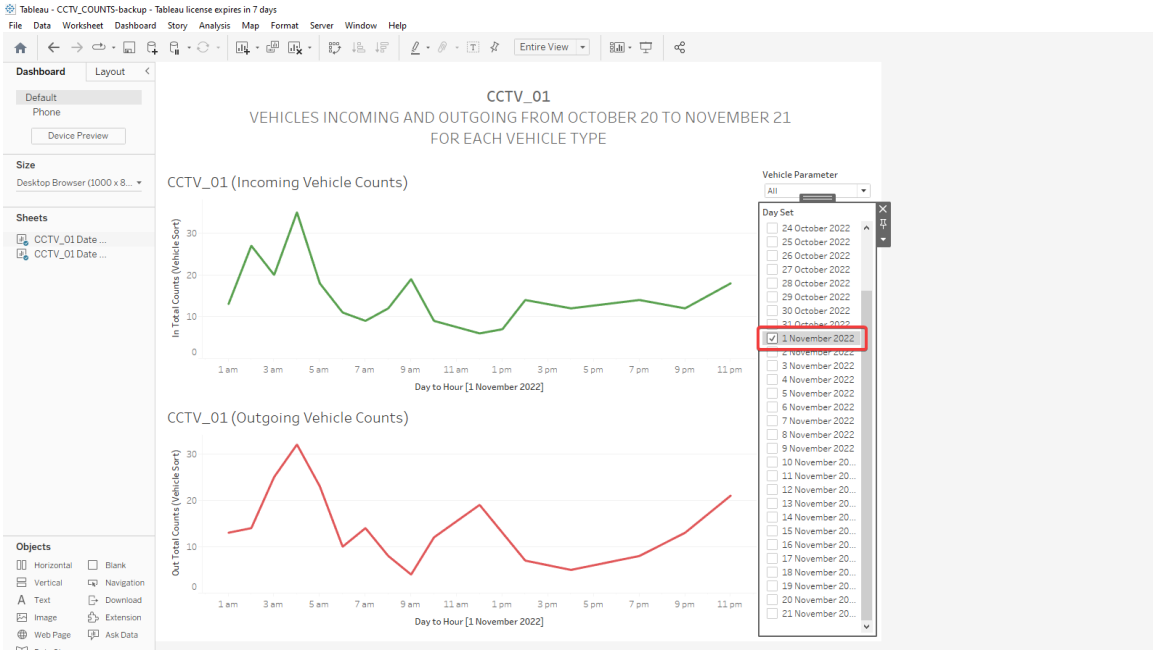


Figure 6.12: Individual CCTV_01 Dashboard (page 2) drilled down

Next, we will select a vehicle type in order to drill down our data even further. In Figure 6.13, we have drilled down the vehicle parameter to only display data counts of bikes within that day.

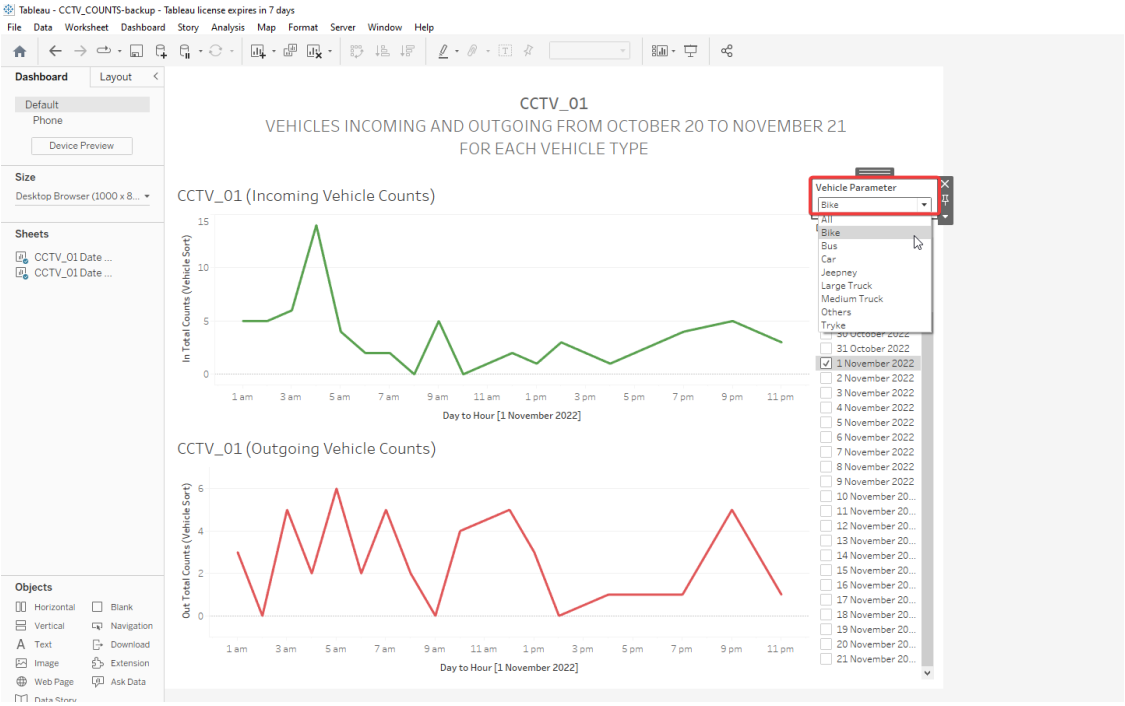


Figure 6.13: Individual CCTV_01 Dashboard (page 2) drilled down 2

7. Compilation of all Tableau Pages

Once we have finished creating our sheets and dashboards for each CCTV sensor, we may now save the project. The following screenshots are of the 22 total pages of data visualization for our cctv_counts Cassandra database.

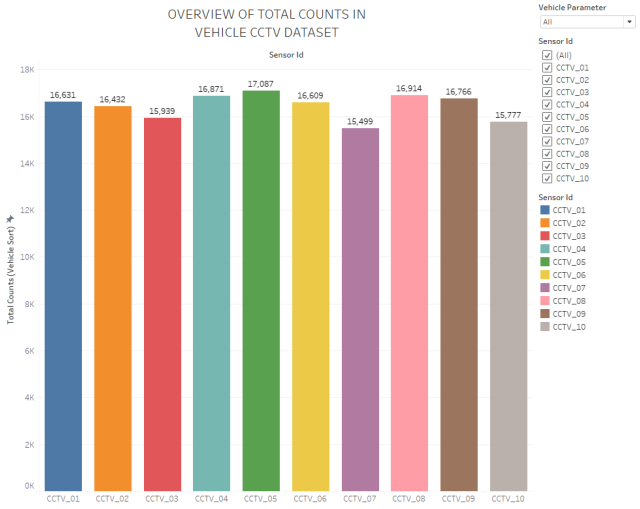


Figure 7.1: Page 1 (visual analytics showing all data captured by each CCTV)
With Drill down (Total > incoming/outgoing)

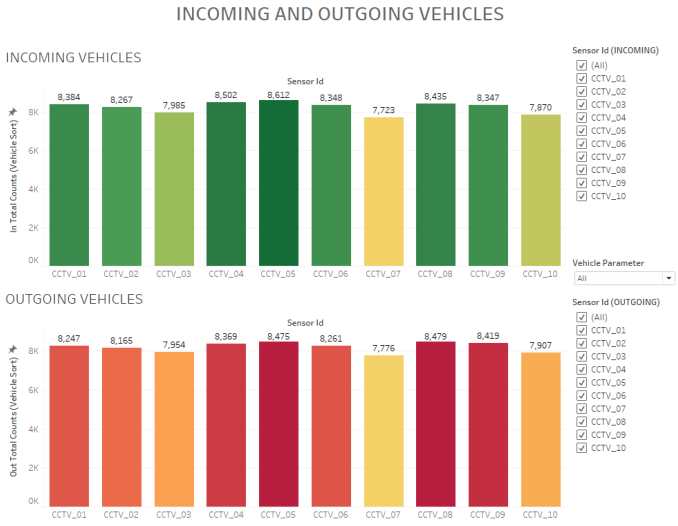


Figure 7.2. Page 2 (visual analytics showing all data captured by each CCTV)

With Drill down (All > Vehicle Types)

CCTV_01

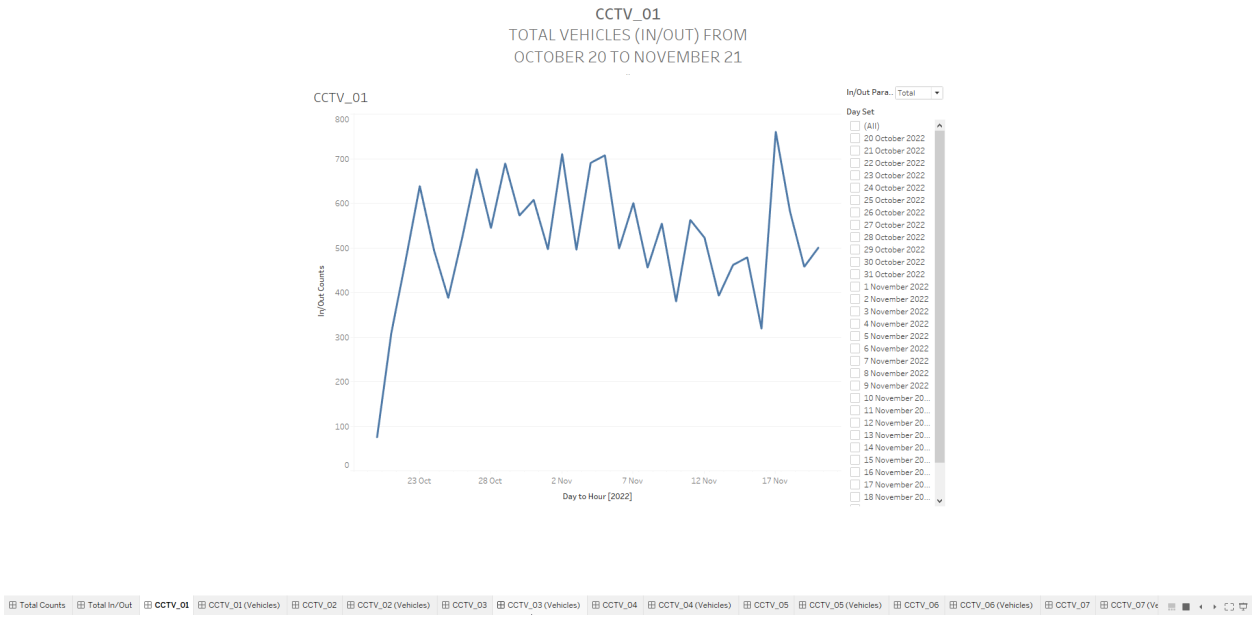


Figure 7.3 Page 3 (CCTV_01 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

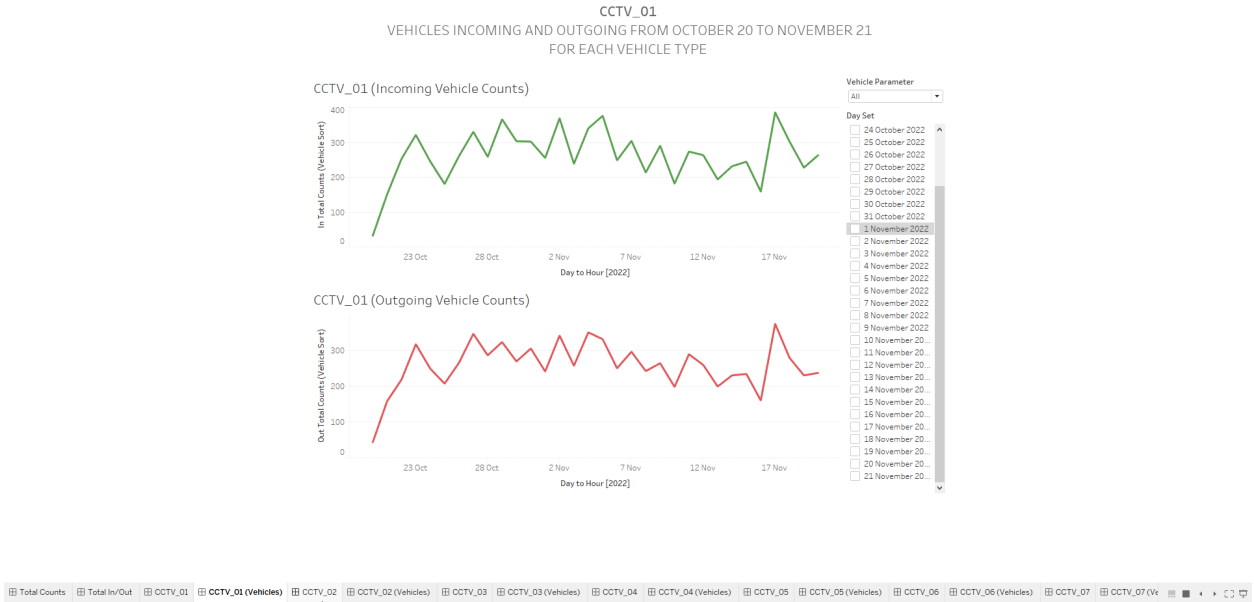


Figure 7.4 Page 4 (CCTV_01 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_02

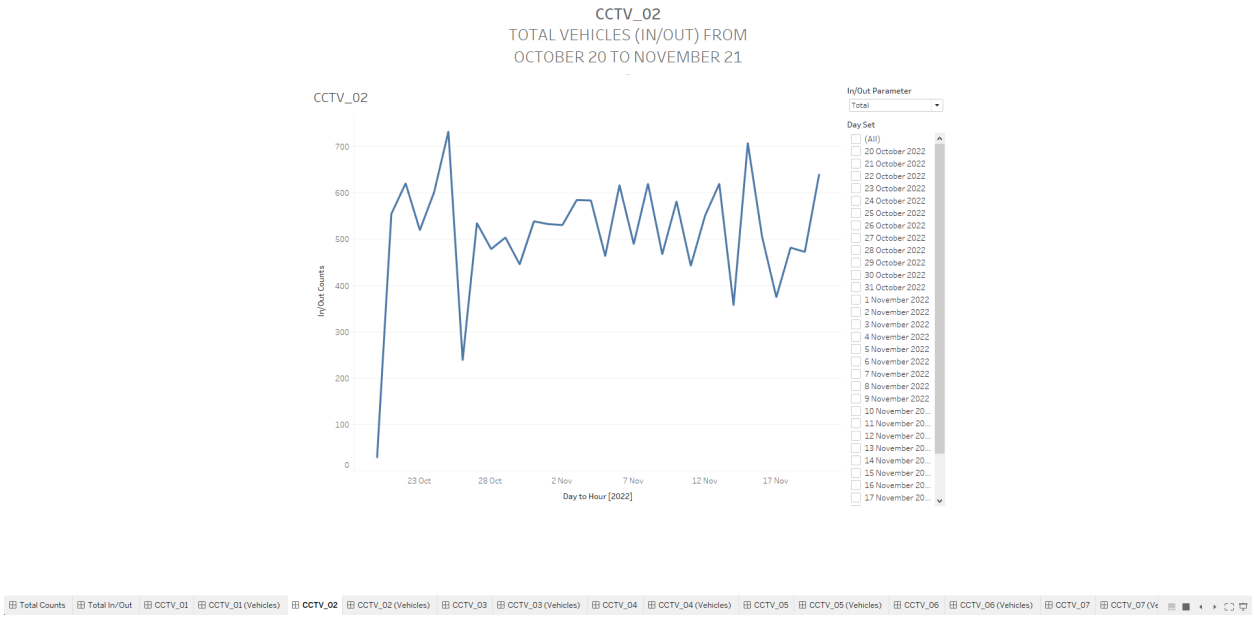


Figure 7.5 Page 5 (CCTV_02 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

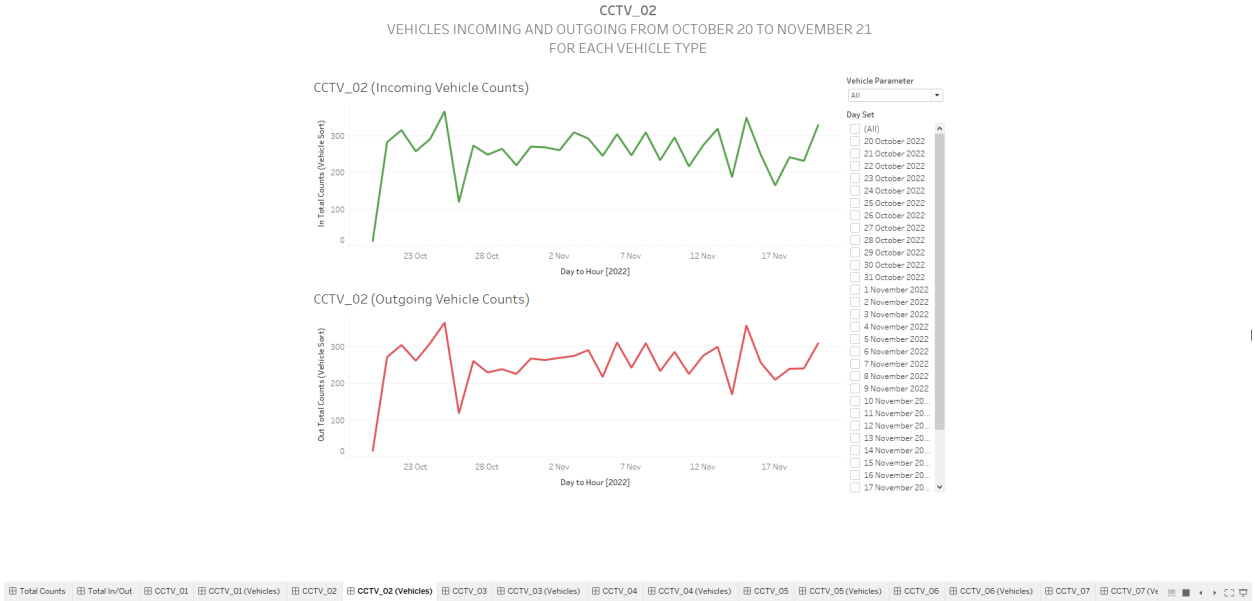


Figure 7.6 Page 6 (CCTV_02 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_03

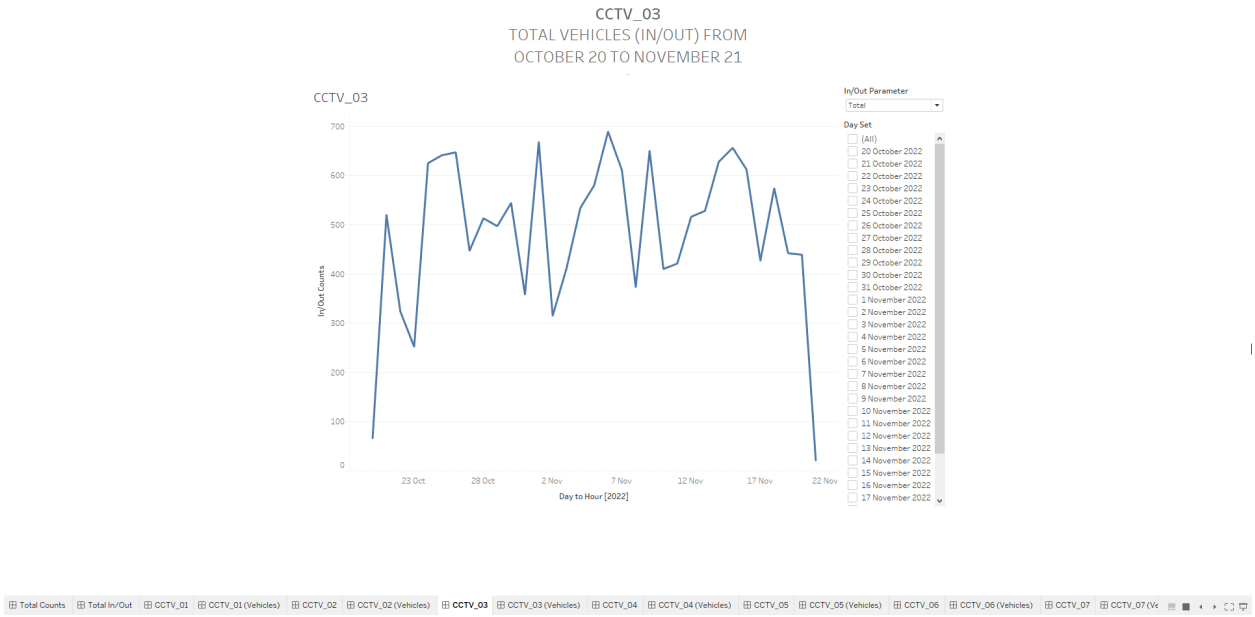


Figure 7.7 Page 7 (CCTV_03 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

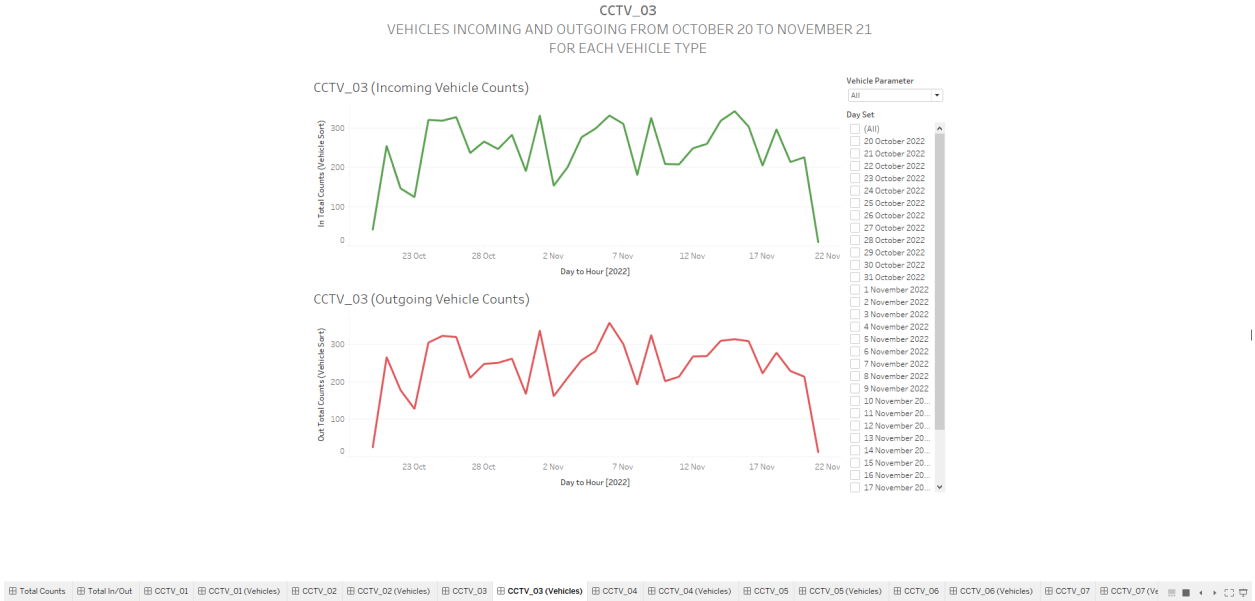


Figure 7.8 Page 8 (CCTV_03 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_04

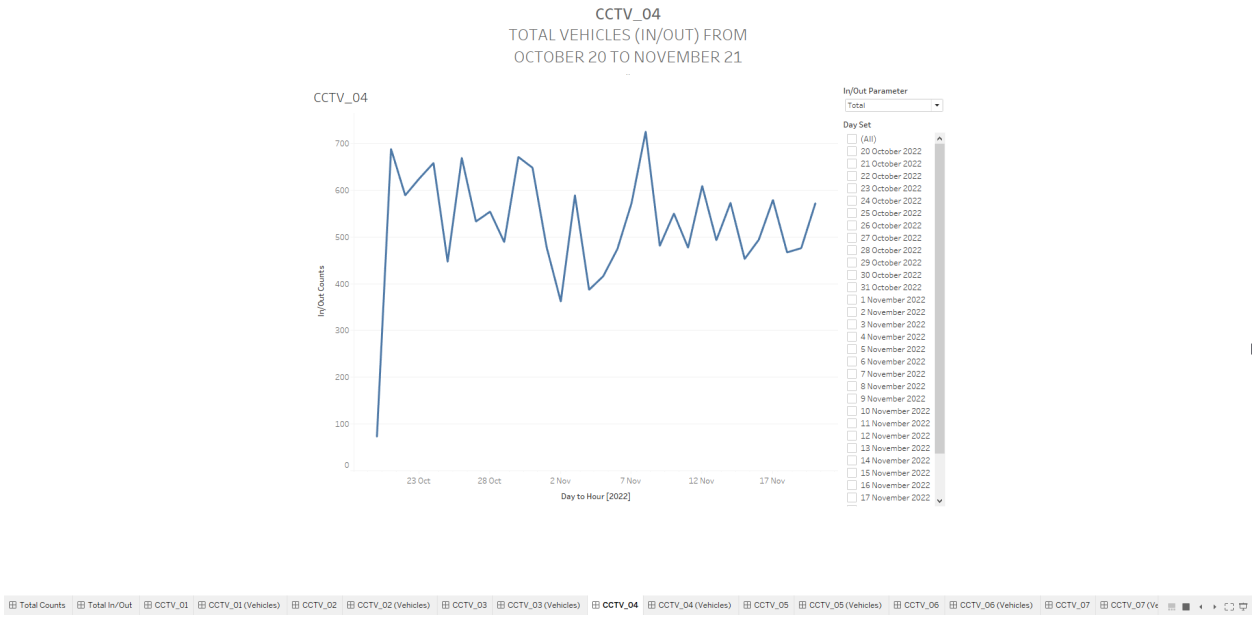


Figure 7.9 Page 9 (CCTV_04 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

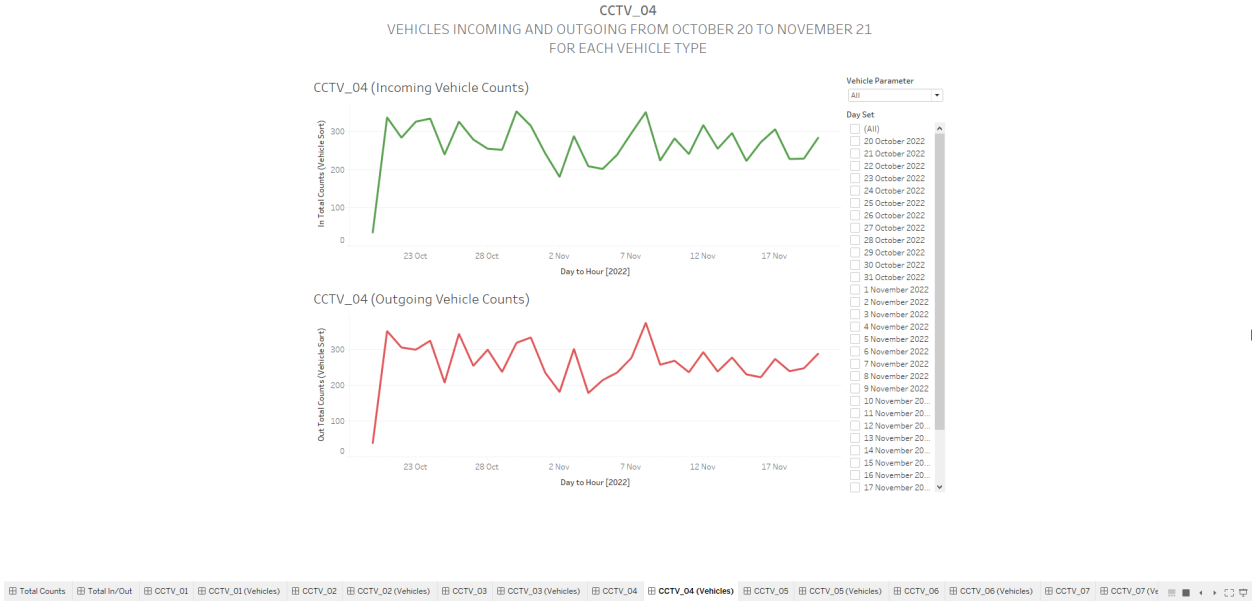


Figure 7.10 Page 10 (CCTV_04 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_05

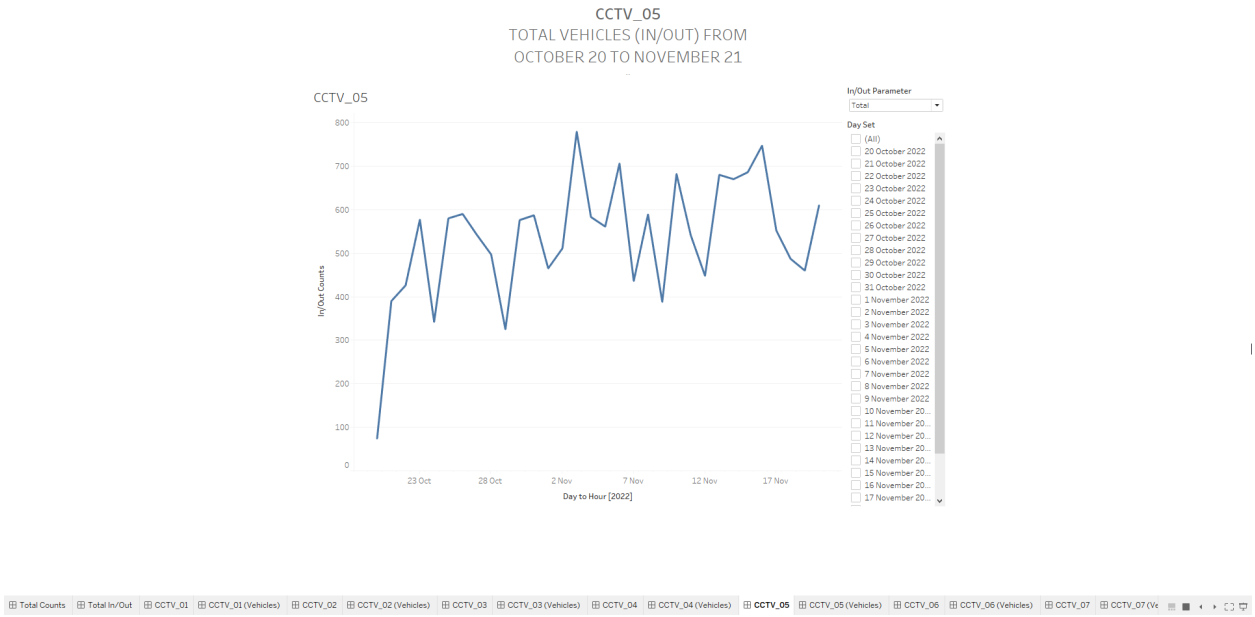


Figure 7.11 Page 11 (CCTV_05 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

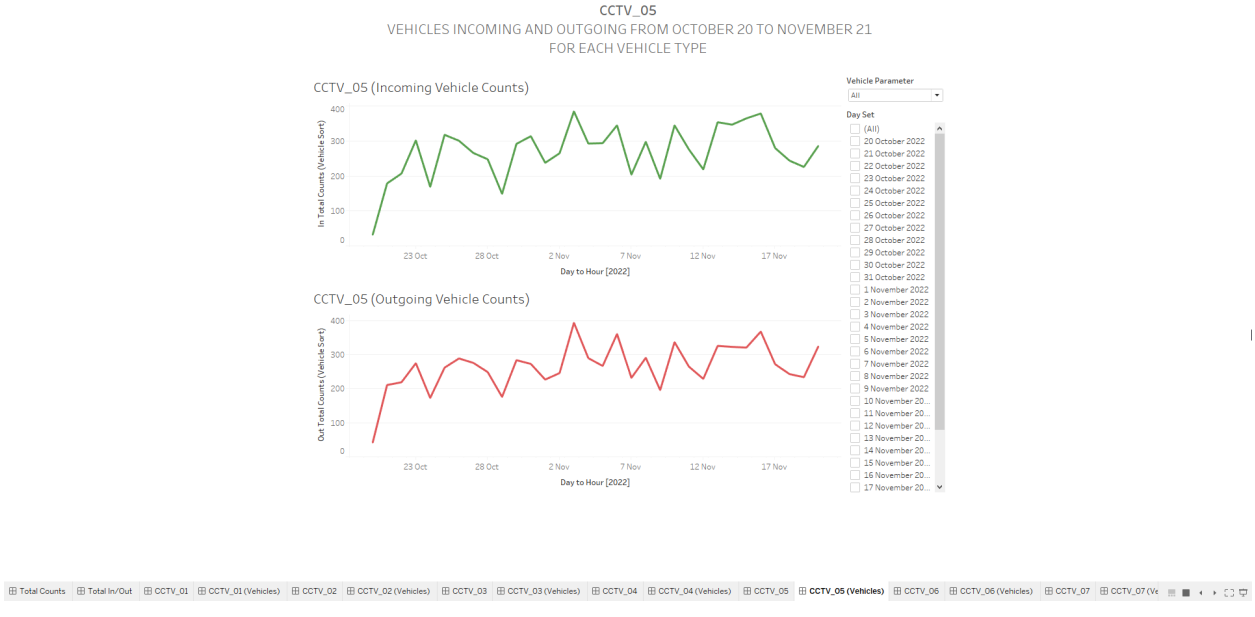


Figure 7.12 Page 12 (CCTV_05 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_06

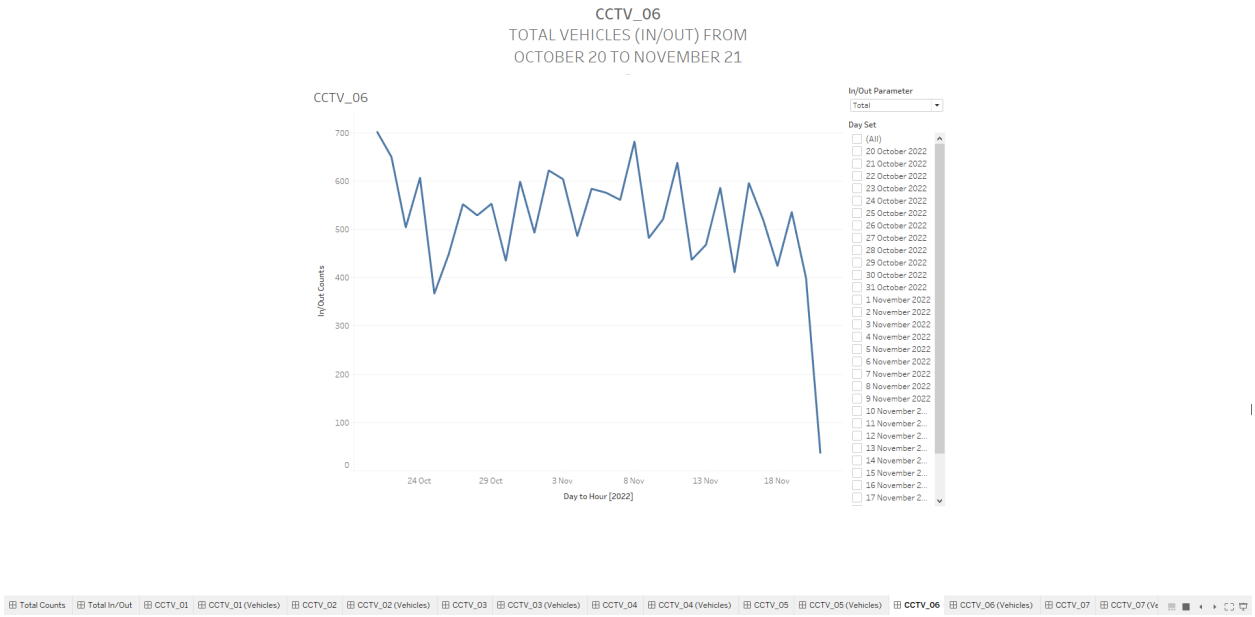


Figure 7.13 Page 13 (CCTV_06 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

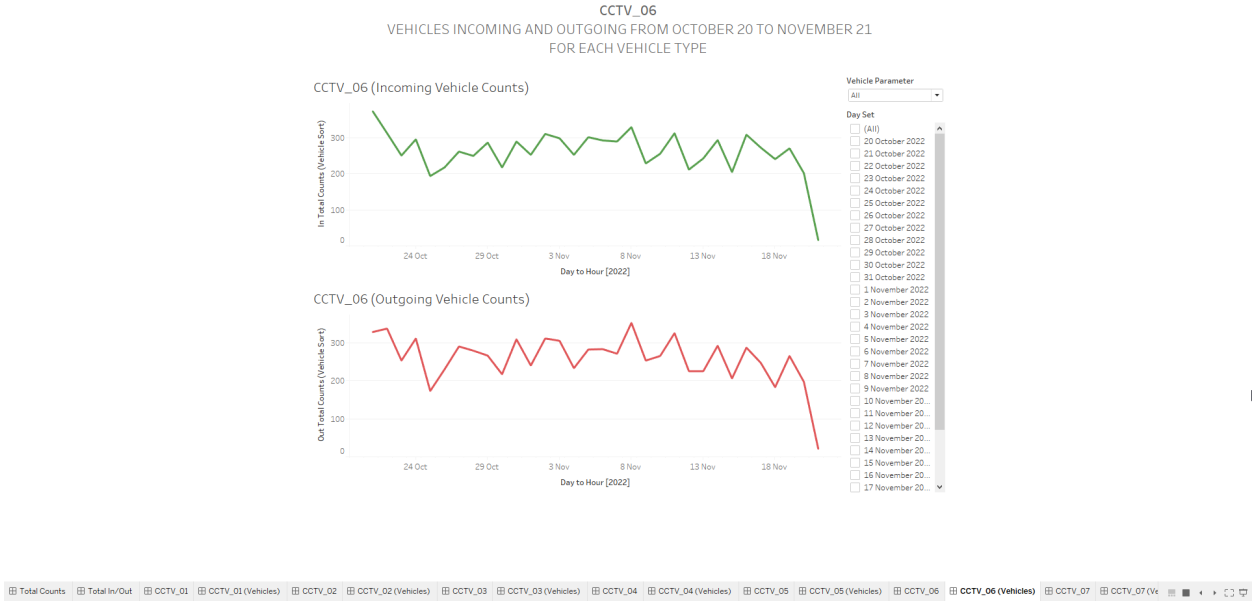


Figure 7.14 Page 14 (CCTV_06 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_07

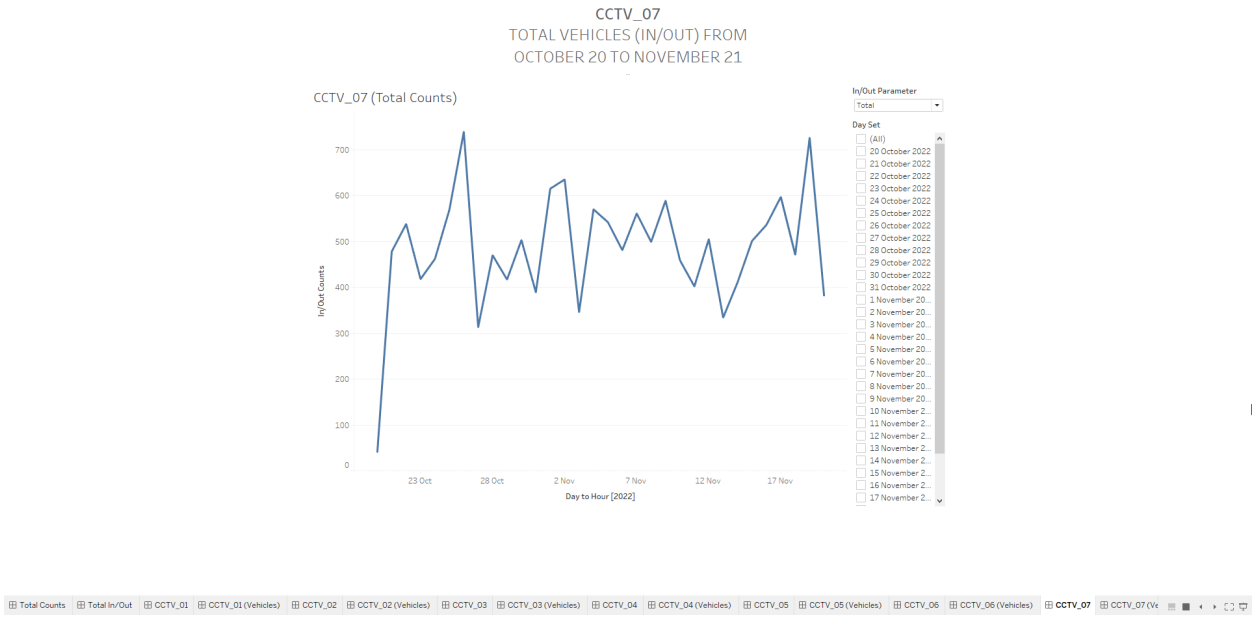


Figure 7.15 Page 15 (CCTV_07 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

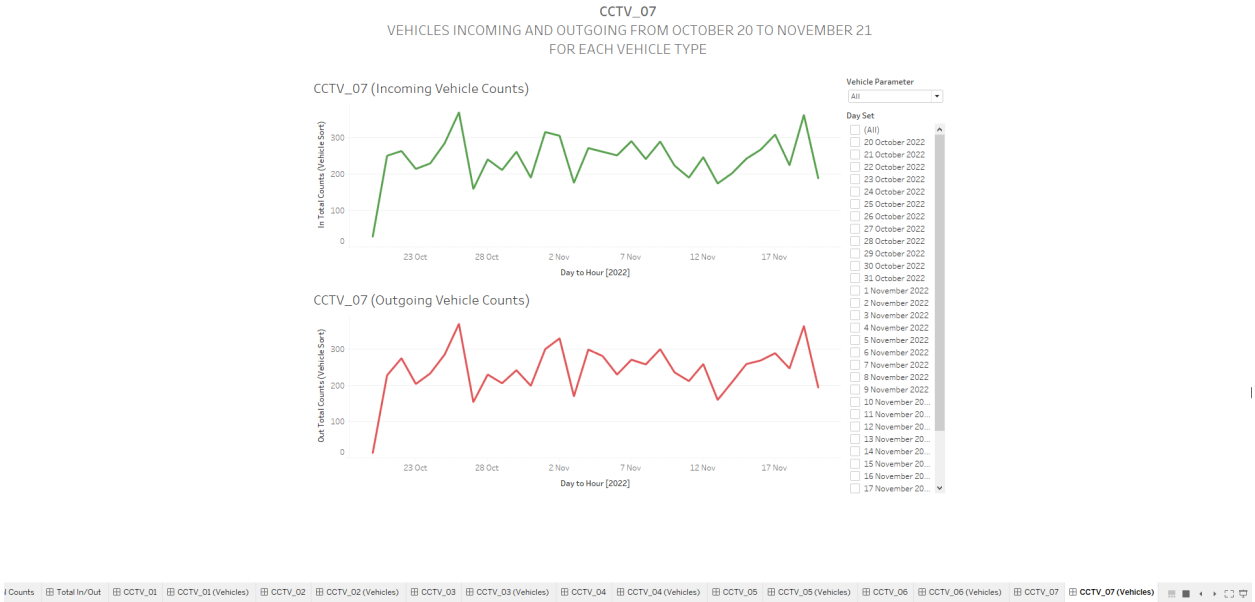


Figure 7.16 Page 16 (CCTV_07 - Overall Total Counts)
With Drill down (All > Vehicle Types)

CCTV_08

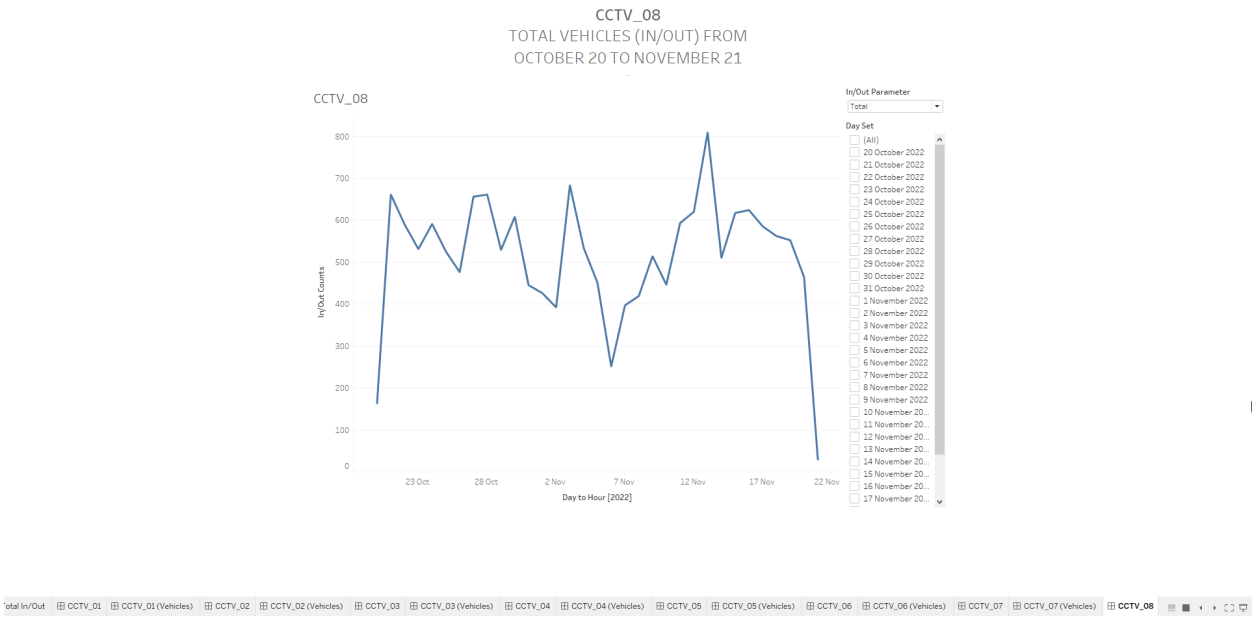


Figure 7.17 Page 17 (CCTV_08 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

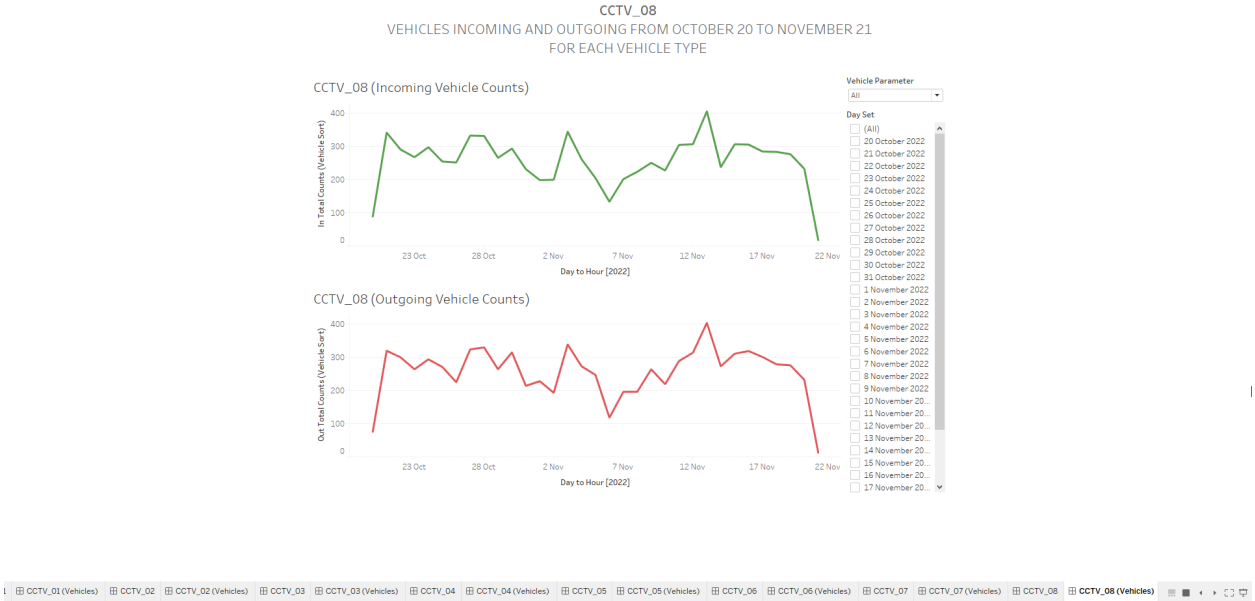


Figure 7.18 Page 18 (CCTV_08 - Overall Total Counts)

With Drill down (All > Vehicle Types)

CCTV_09

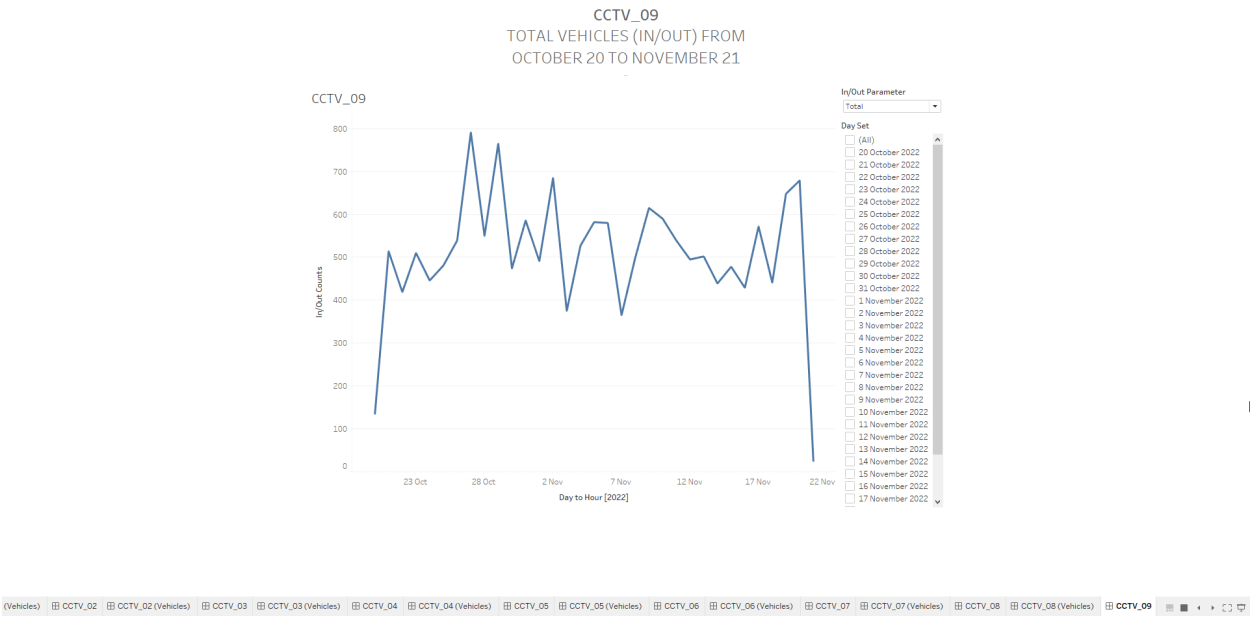


Figure 7.19 Page 19 (CCTV_09 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

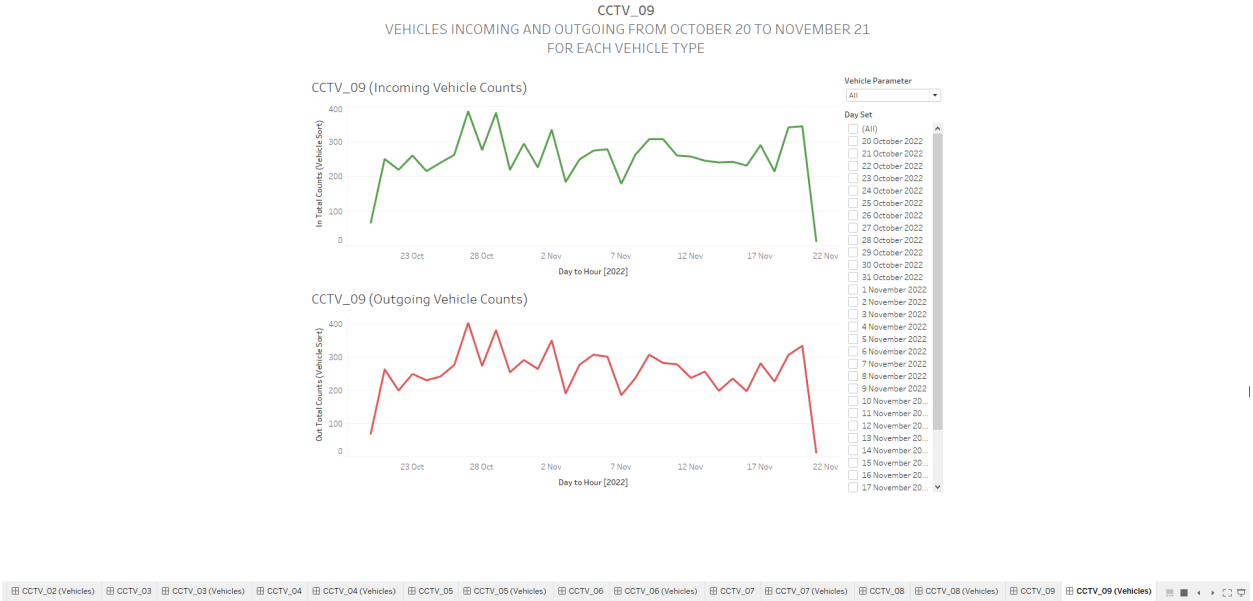


Figure 7.20 Page 20 (CCTV_09 - Overall Total Counts)

With Drill down (All > Vehicle Types)

CCTV_10

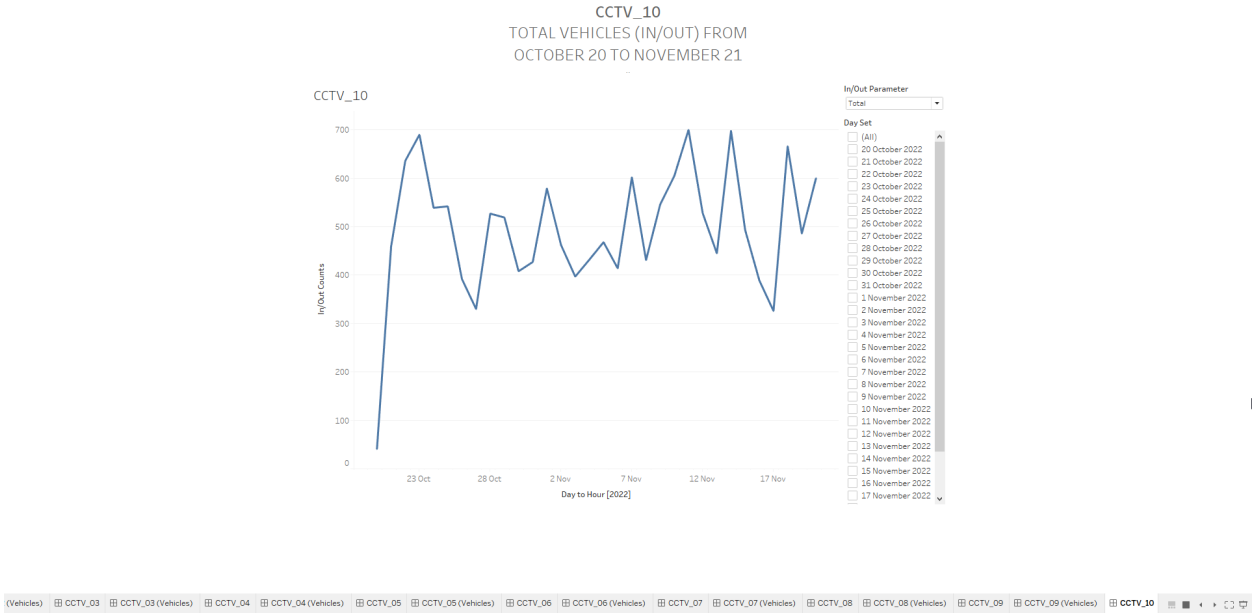


Figure 7.21 Page 21 (CCTV_10 - Overall Total Counts)
With Drill down (Overall total counts > Incoming/Outgoing vehicles)

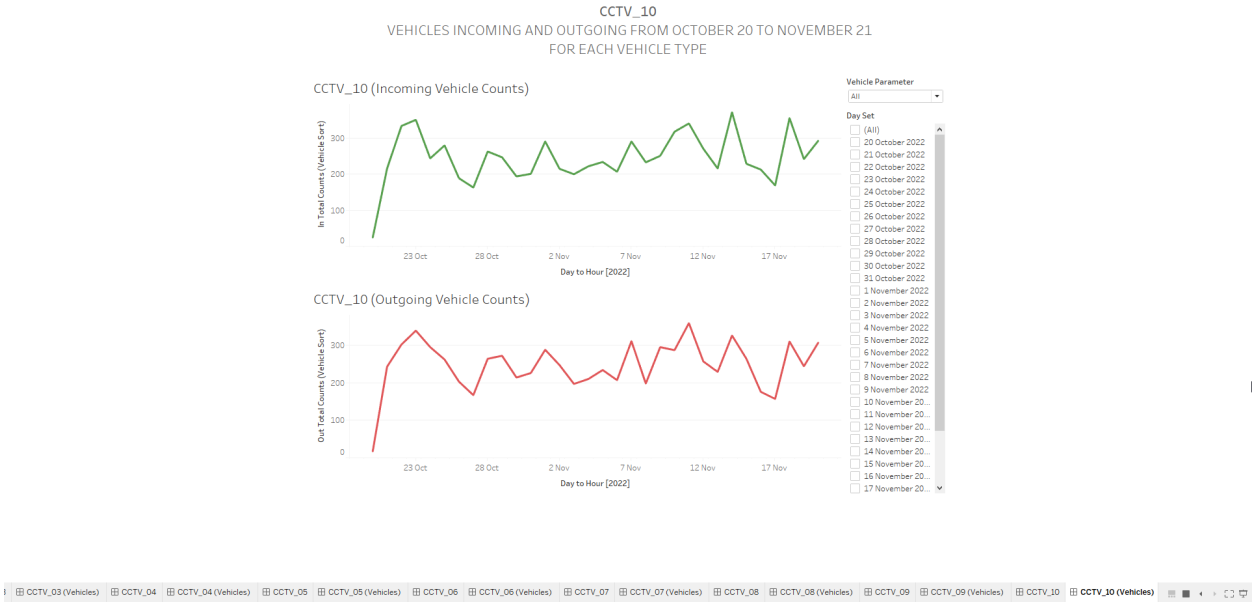


Figure 7.22 Page 22 (CCTV_10 - Overall Total Counts)
With Drill down (All > Vehicle Types)