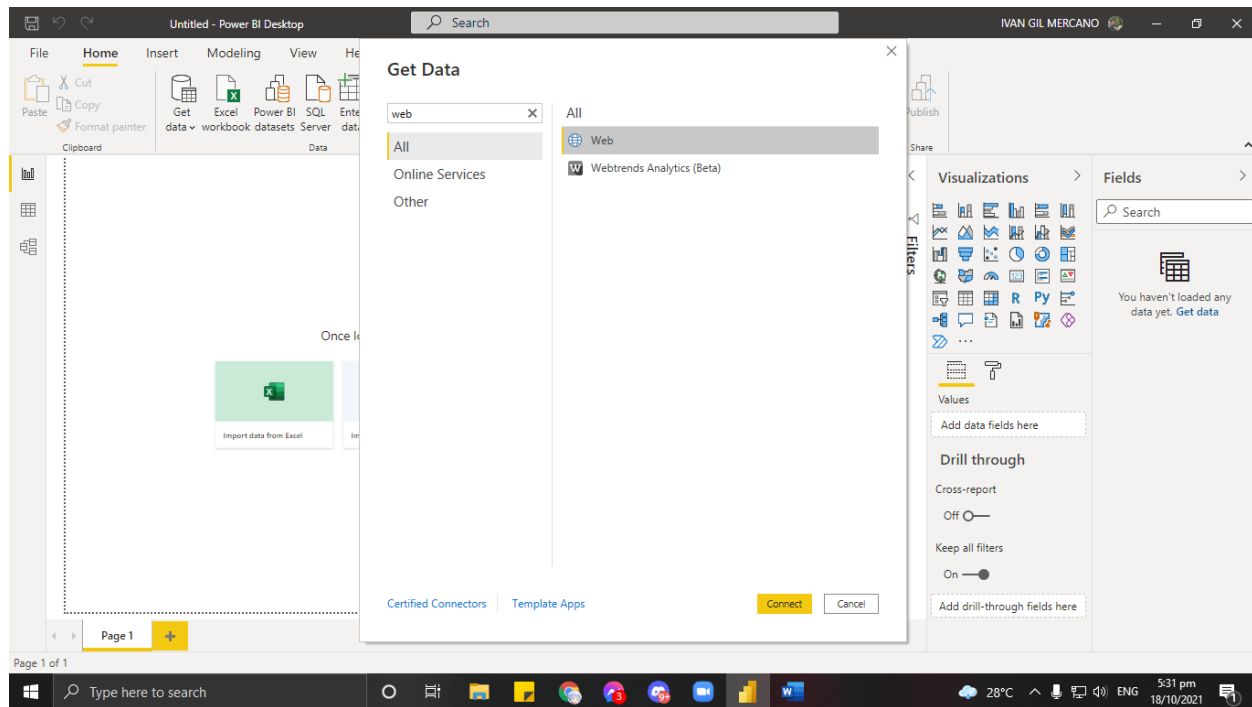
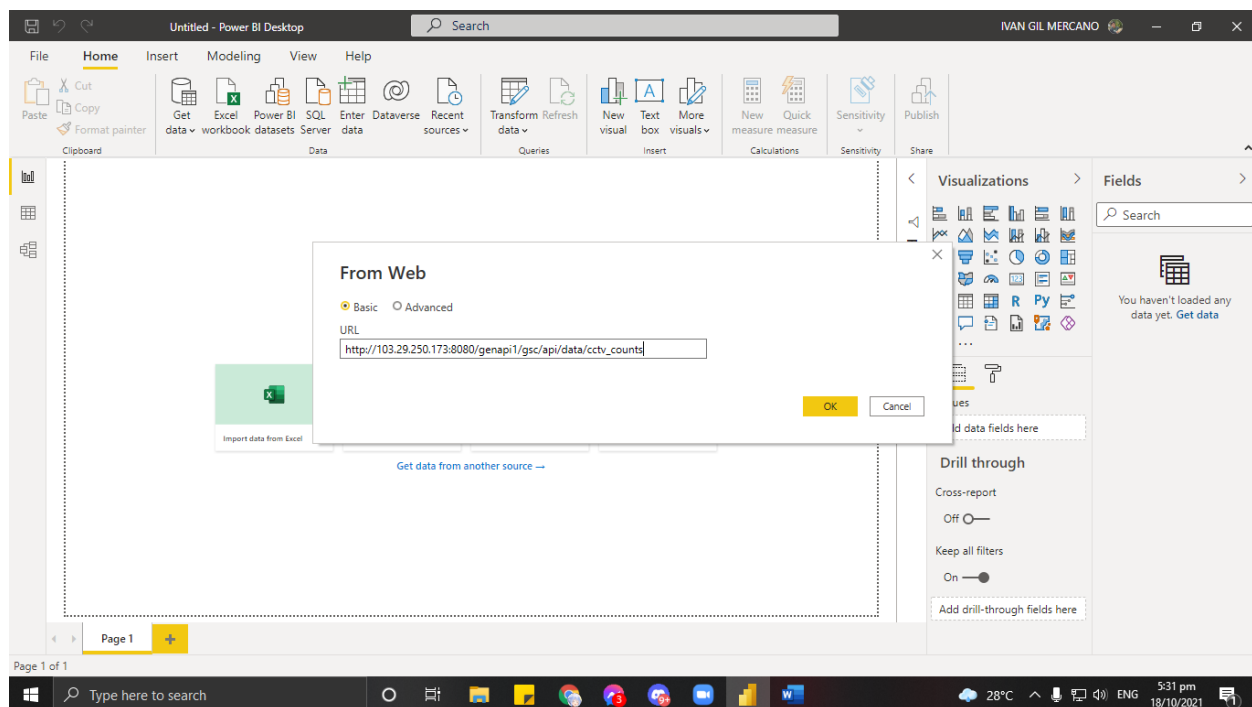


First is uploading the API into the Power BI



We pasted the API link provided in the submission link



Then we viewed the data as table and check if there are any data cleaning process we need to do.

The screenshot shows the Power BI Desktop interface with the Power Query Editor open. The main view displays a table with 24 columns and 999+ rows. The columns are: cctv_counts.timeuuid_id, cctv_counts.lgu_code, cctv_counts.sensor_id, and cctv_counts.date_saved. The data is sorted by cctv_counts.timeuuid_id in ascending order. The table shows 16 rows of data, with the first row having a timeuuid_id of 09dceda0-c140-4173-99b2-b73d8786722c and a date_saved of 07/20/2021. The table is titled 'Table.TransformColumnTypes(#'Expanded cctv_counts1',{{'cctv_counts.timeuuid_id', type ...}}'. The Query Settings pane on the right shows the query name 'cctv_counts' and the applied steps: Source, Converted to Table, Expanded cctv_counts, Expanded cctv_counts1, and Changed Type. The bottom status bar indicates 'Page 1 of 1' and 'Column profiling based on top 1000 rows'.

To be sure, we remove empty cells that are not necessary for the task

The screenshot shows the Power BI Desktop interface with the Power Query Editor open. The 'Remove Empty' dialog box is displayed, allowing the user to select columns to remove empty values from. The dialog box has a 'Search' field and a list of columns with checkboxes. The columns listed are: (Select All), 09dceda0-c140-4173-99b2-b73d8786722c, 0c092e8c-625c-44b6-8d2f-e9c33b3891bf, 1.626768799406483E9, 1.626768800411792E9, 1.626768801413752E9, 1.626768802415396E9, and 1.626768803416353E9. The 'Limit of 1000 values reached.' warning is visible. The 'OK' button is highlighted. The background shows the same table of CCTV counts data as the previous screenshot.

The screenshot displays the Microsoft Power Query Editor interface. The main window shows a table with columns 'cctv_counts.lgu_code', 'cctv_counts.sensor_id', and 'cctv_counts.date_saved'. A 'Filter Rows' step is applied, showing a list of values to filter by, including '09dceda0-c140-4173-99b2-b73d876722c' and several GUIDs. The 'Query Settings' pane on the right shows the 'PROPERTIES' and 'APPLIED STEPS' sections. The 'APPLIED STEPS' list includes 'Source', 'Converted to Table', 'Expanded cctv_counts', 'Expanded cctv_counts1', 'Changed Type', and 'Filtered Rows'. The bottom status bar indicates '24 COLUMNS, 95' and 'PREVIEW DOWNLOADED AT 5:25 PM'.

The screenshot displays the Microsoft Power Query Editor interface. The main window shows a table with columns: `cctv_counts.sensor_id`, `cctv_counts.date_saved`, and `cctv_counts.time_saved`. A 'Filter Rows' step is applied, showing a list of dates from 07-29-2021 to 08-08-2021. The 'Query Settings' pane on the right shows the 'PROPERTIES' and 'APPLIED STEPS' sections. The 'APPLIED STEPS' list includes 'Source', 'Converted to Table', 'Expanded cctv_counts', 'Expanded cctv_counts1', 'Changed Type', and 'Filtered Rows'. The status bar at the bottom indicates 'Page 4 of 4', '0 words', 'English (Philippines)', and 'Accessibility: Investigate'.

So we decided to replace the format of the date with the unique format

The screenshot shows the Power Query Editor interface. A table with columns 'code', 'cctv_counts.sensor_id', 'cctv_counts.date_saved', and 'cctv_counts.time_saved' is displayed. The 'cctv_counts.date_saved' column contains dates like '07/20/2021'. A context menu is open over the first row of this column, with the 'Replace Values...' option selected. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, which includes 'Filtered Rows'.

code	cctv_counts.sensor_id	cctv_counts.date_saved	cctv_counts.time_saved
1	9500	CCTV_01	07/20/2021
2	9500	CCTV_01	07/20/2021
3	9500	CCTV_01	07/20/2021
4	9500	CCTV_01	07/20/2021
5	9500	CCTV_01	07/20/2021
6	9500	CCTV_01	07/20/2021
7	9500	CCTV_01	07/20/2021
8	9500	CCTV_01	07/20/2021
9	9500	CCTV_01	07/20/2021
10	9500	CCTV_01	07/20/2021
11	9500	CCTV_01	07/20/2021
12	9500	CCTV_01	07/20/2021
13	9500	CCTV_01	07/20/2021

The screenshot shows the Power Query Editor with the 'Replace Values' dialog box open. The dialog has fields for 'Value To Find' (07/20/2021) and 'Replace With' (07-20-2021). The 'Advanced options' section is expanded. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, which includes 'Filtered Rows'.

code	cctv_counts.sensor_id	cctv_counts.date_saved	cctv_counts.time_saved
1	9500	CCTV_01	07/20/2021
2	9500	CCTV_01	07/20/2021
3	9500	CCTV_01	07/20/2021
4	9500	CCTV_01	07/20/2021
5	9500	CCTV_01	07/20/2021
6	9500	CCTV_01	07/20/2021
7	9500	CCTV_01	07/20/2021
8	9500	CCTV_01	07/20/2021
9	9500	CCTV_01	07/20/2021
10	9500	CCTV_01	07/20/2021
11	9500	CCTV_01	07/20/2021
12	9500	CCTV_01	07/20/2021
13	9500	CCTV_01	07/20/2021

The screenshot shows the Microsoft Power Query Editor interface. The main data table has the following columns: **cctv_counts.timeuuid_id**, **cctv_counts.lgu_code**, **cctv_counts.sensor_id**, and **cctv_counts.date_saved**. The data consists of 24 columns and 999+ rows. The 'Table.ReplaceValue' dialog is open, displaying a list of rows with a search filter for 'CCTV_01' through 'CCTV_MAPUA'. The 'Replaced Value' step is highlighted in the 'APPLIED STEPS' pane on the right.

The screenshot shows the Power Query Editor interface. The main data view displays a table with columns: cctv_counts.sensor_id, cctv_counts.date_saved, cctv_counts.time_saved, and cctv_counts.count_total. A filter dialog is open for the 'cctv_counts.time_saved' column, showing a list of times from 12:00:00 am to 6:05:00 am. The dialog includes options for sorting, clearing, and removing empty values. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, which includes 'Filtered Rows1'.

cctv_counts.sensor_id	cctv_counts.date_saved	cctv_counts.time_saved	cctv_counts.count_total
1 TV_01	07-2	Sort Ascending	
2 TV_01	07-2	Sort Descending	
3 TV_01	07-2	Clear Sort	
4 TV_01	07-2	Clear Filter	
5 TV_01	07-2	Remove Empty	
6 TV_01	07-2	Time Filters	
7 TV_01	07-2		
8 TV_01	07-2		
9 TV_01	07-2		
10 TV_01	07-2		
11 TV_01	07-2		
12 TV_01	07-2		
13 TV_01	07-2		
14 TV_01	07-2		
15 TV_01	07-2		
16 TV_01	07-2		
17 TV_01	07-2		
18 TV_01	07-2		
19 TV_01	07-2		
20 TV_01	07-2		
21 TV_MAPUA	07-2		

The screenshot shows the Power Query Editor interface. The main data view displays a table with columns: cctv_counts.sensor_id, cctv_counts.date_saved, cctv_counts.time_saved, and cctv_counts.count_total. A filter dialog is open for the 'cctv_counts.date_saved' column, showing a list of dates from 07-20-2021 to 07-29-2021. The dialog includes options for sorting, clearing, and removing empty values. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, which includes 'Filtered Rows1'.

cctv_counts.sensor_id	cctv_counts.date_saved	cctv_counts.time_saved	cctv_counts.count_total
1 .01	07-20-2021	Sort Ascending	
2 .01	07-20-2021	Sort Descending	
3 .01	07-20-2021	Clear Sort	
4 .01	07-20-2021	Clear Filter	
5 .01	07-20-2021	Remove Empty	
6 .01	07-20-2021	Number Filters	
7 .01	07-20-2021		
8 .01	07-20-2021		
9 .01	07-20-2021		
10 .01	07-20-2021		
11 .01	07-20-2021		
12 .01	07-20-2021		
13 .01	07-20-2021		
14 .01	07-20-2021		
15 .01	07-20-2021		
16 .01	07-20-2021		
17 .01	07-20-2021		
18 .01	07-20-2021		
19 .01	07-20-2021		
20 .01	07-20-2021		
21 .MAPUA	07-29-2021		

Close & Apply
Close the Query Editor window and apply any pending changes.

Query Settings

PROPERTIES

Name
cctv_counts

APPLIED STEPS

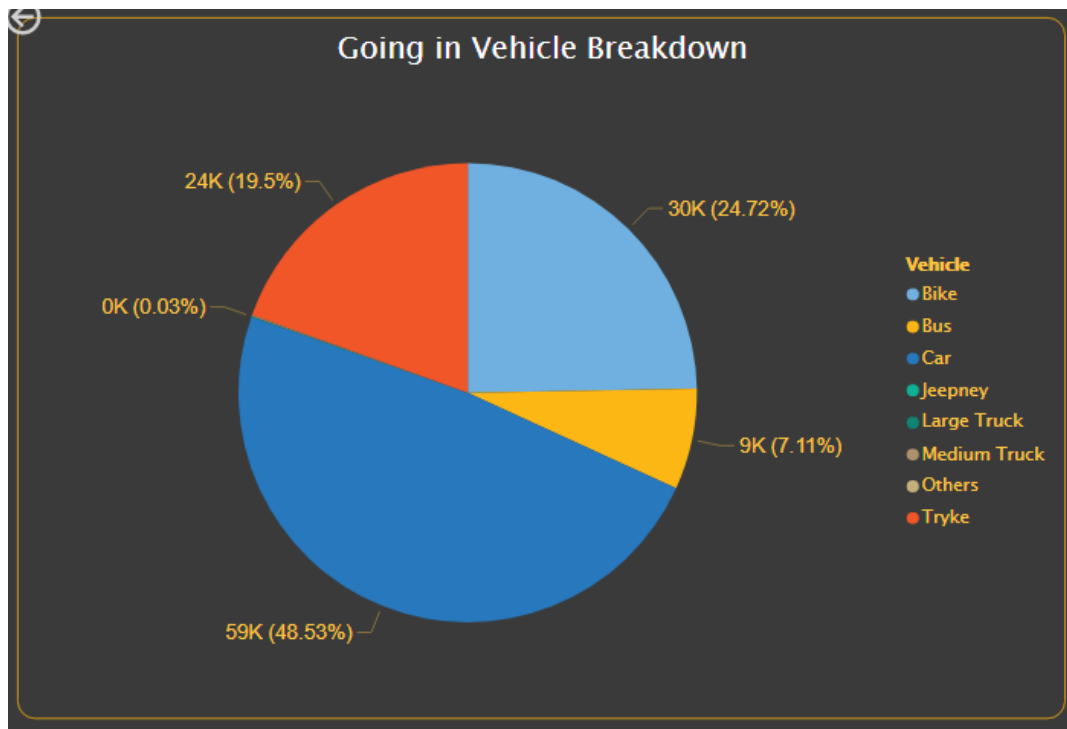
- Source
- Converted to Table
- Expanded cctv_counts
- Expanded cctv_counts1
- Changed Type
- Filtered Rows
- Replaced Value
- Filtered Rows1

24 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows

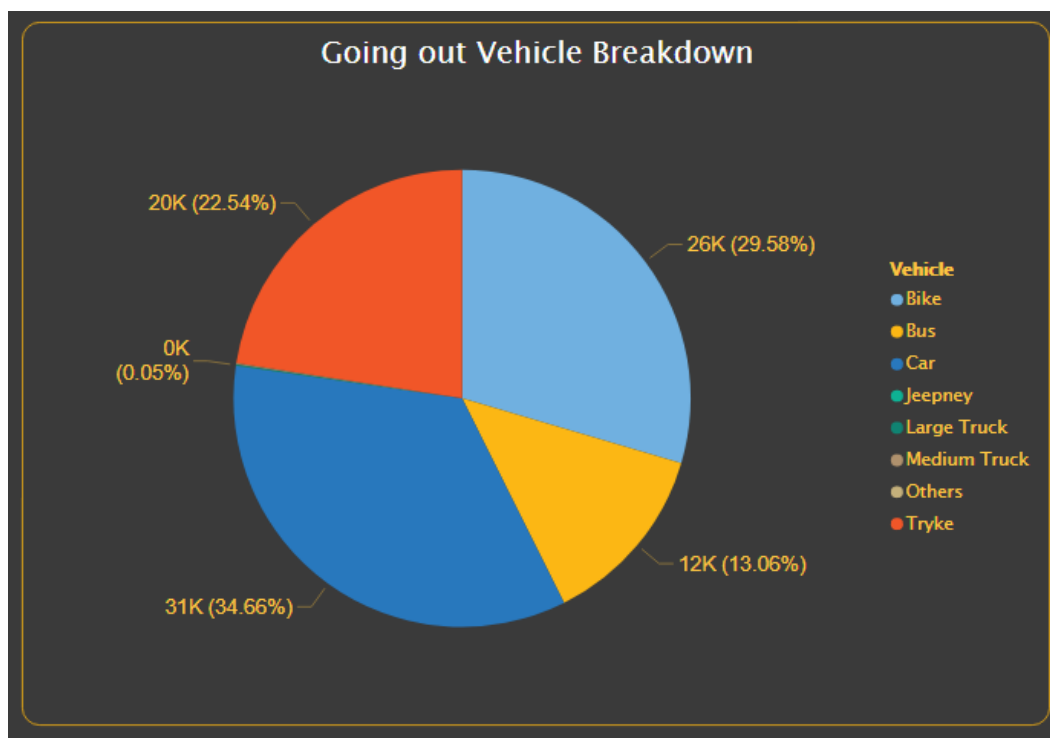
PREVIEW DOWNLOADED AT 5:25 PM

After making sure that the data is properly cleaned and formatted, we then proceeded in making the reports based on the dataset

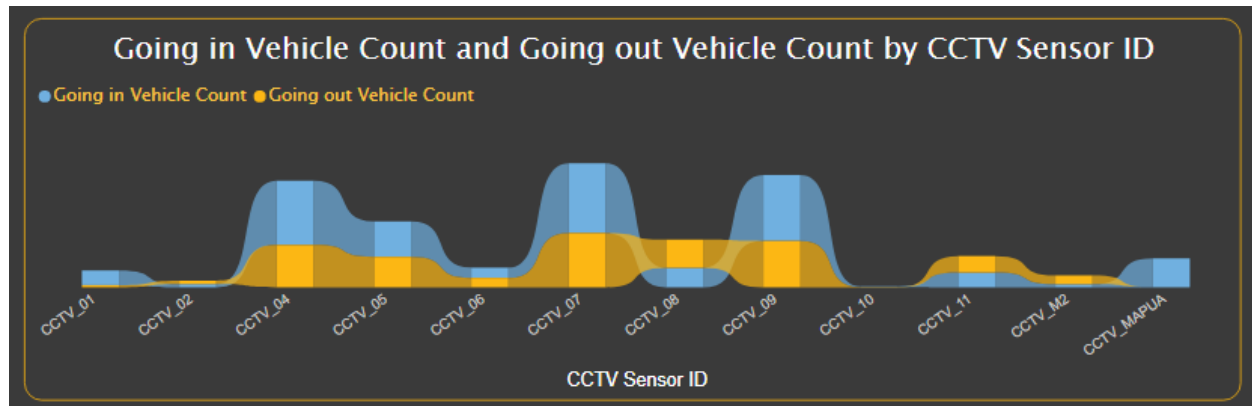
The first graph we created aims to display the total amount of individual vehicles that went inside.



Like the first graph, the second graph shows the total amount of vehicle that went out instead.



The next graph we made shows the categorized number of vehicles that went in and out and is sorted by the CCTV ID.



For better analysis of the data, we added the option of inspecting the number of vehicles on a specific CCTV camera and date through a list option.

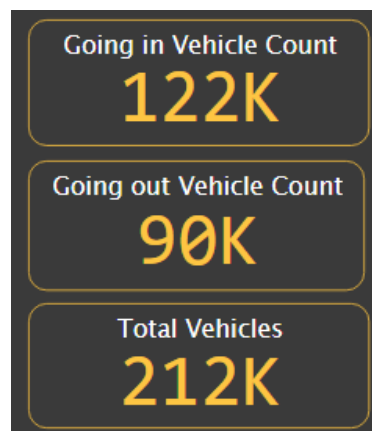
CCTV Sensor ID

- ☐ CCTV_01
- ☐ CCTV_02
- ☐ CCTV_04
- ☐ CCTV_05
- ☐ CCTV_06
- ☐ CCTV_07
- ☐ CCTV_08
- ☐ CCTV_09
- ☐ CCTV_10
- ☐ CCTV_11
- ☐ CCTV_M2
- ☐ CCTV_MAPUA

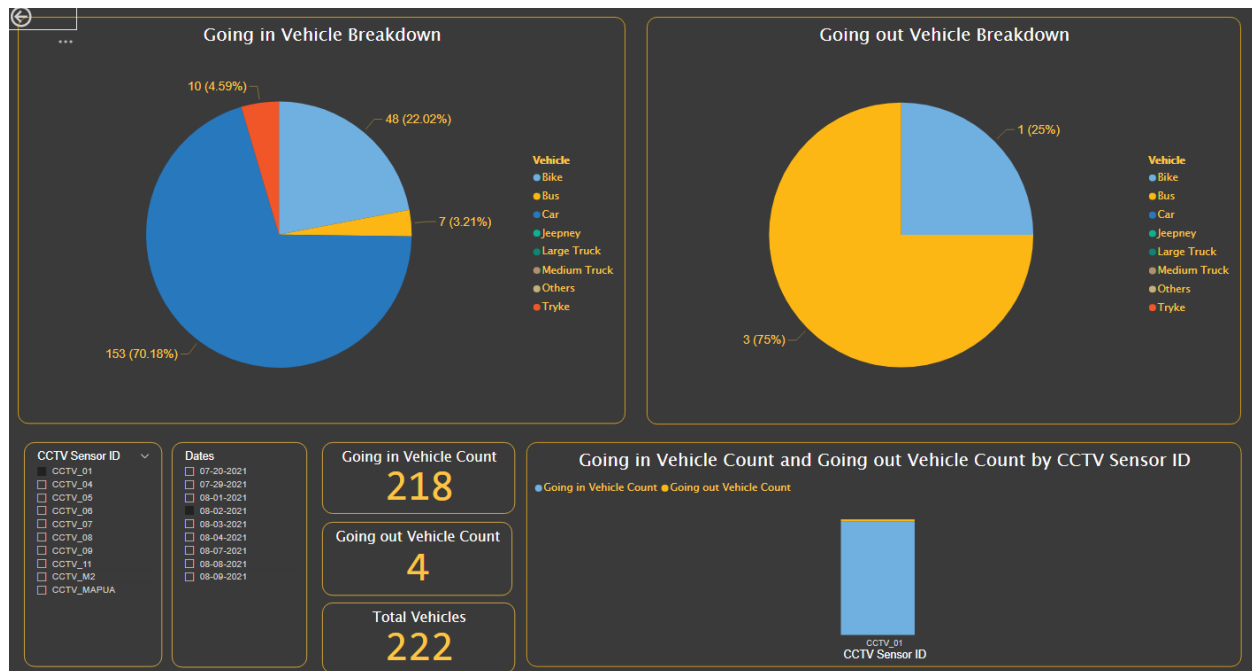
Dates

- ☐ 07-20-2021
- ☐ 07-29-2021
- ☐ 07-30-2021
- ☐ 07-31-2021
- ☐ 08-01-2021
- ☐ 08-02-2021
- ☐ 08-03-2021
- ☐ 08-04-2021
- ☐ 08-07-2021
- ☐ 08-08-2021
- ☐ 08-09-2021
- ☐ 08-10-2021
- ☐ 08-11-2021

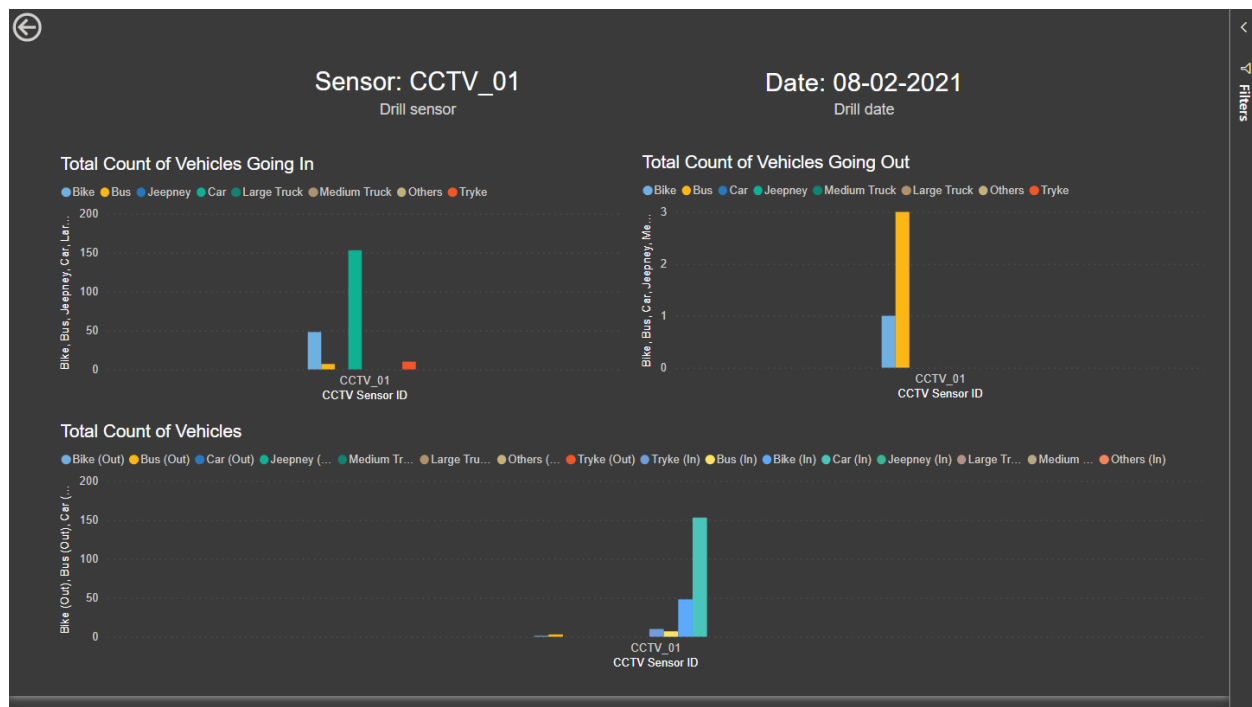
Then next we made a counter that displays the amount of going in vehicles, going out vehicles, and the total overall vehicles.



After creating the page for vehicles going in and out, the next task to be done is to make the button selection functional. Below is an example of the filter working.



Next, we applied Drill through to conclude upon the analysis of the previous graph, The example below is a drill through of the selected image above.



As we can see the date and CCTV ID remains the same, whilst providing a summarized view of the combined data.

3. Though we do not have a good historical data, just try to do forecasting as a demonstration of predictive analytics.

First, we created a new table for the calendar

assignment_2.3 (1) - Power BI Desktop

File Home Help Table tools

Name Calendar 1

Structure

```
1 Calendar 1 =
2 VAR Days = CALENDAR ( DATE ( 2021, 1, 1 ), DATE ( 2021, 12, 31 ) )
3 RETURN ADDCOLUMNS (
4     Days,
5     "Year", YEAR ( [Date] ),
6     "Month Number", MONTH ( [Date] ),
7     "Month", FORMAT ( [Date], "mmm" ),
8     "Year Month Number", YEAR ( [Date] ) * 12 + MONTH ( [Date] ) - 1,
9     "Year Month", FORMAT ( [Date], "mm yy" ),
10    "Day Number", DAY ( [Date] ),
11    "DateKey", FORMAT ( [Date], "mmddyyyy" )
12 )
```

Fields

Calendar 1

cctv_counts

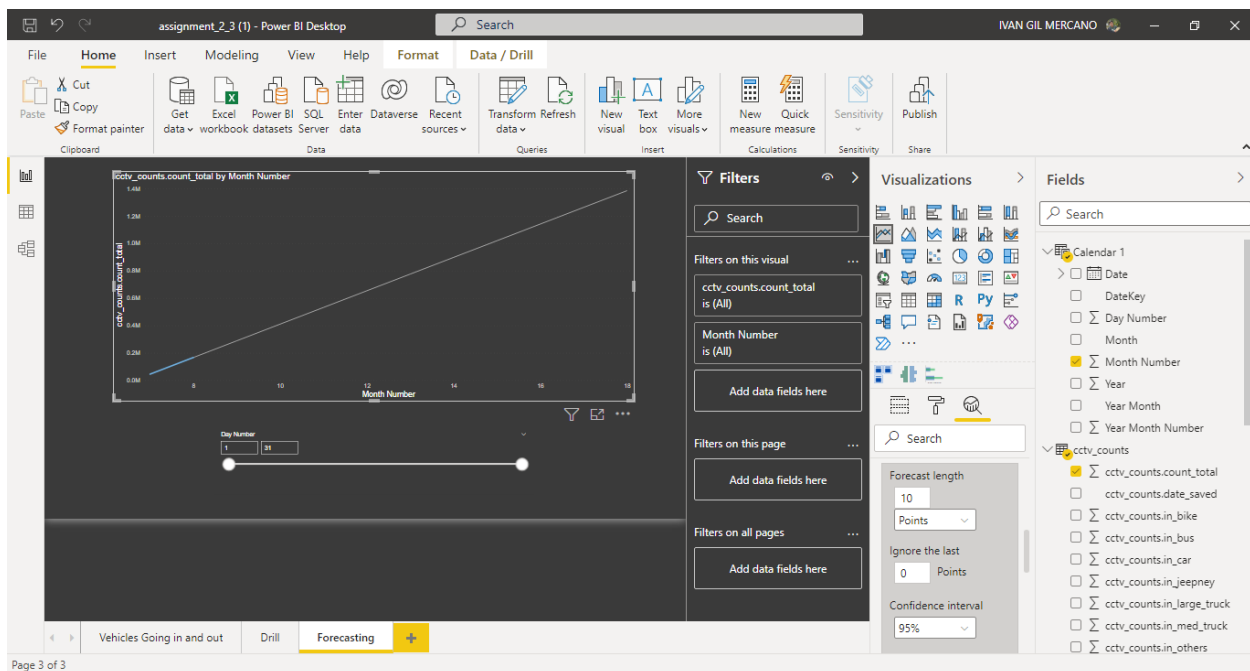
Table Calendar

Time

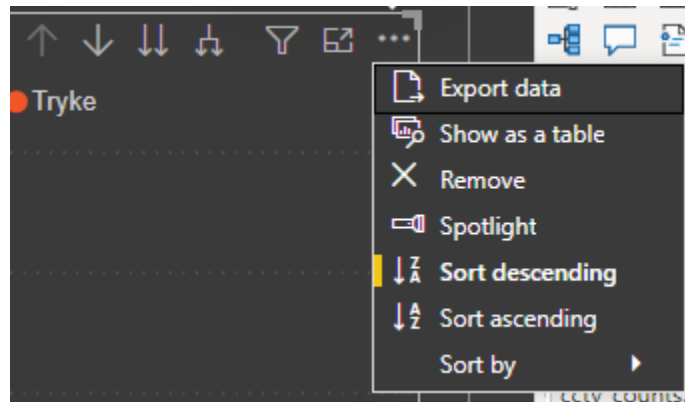
Date	Year	Month Number	Month	Year Month Number	Year Month	DateKey	Day Number
01/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01012021	1
02/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01022021	2
03/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01032021	3
04/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01042021	4
05/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01052021	5
06/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01062021	6
07/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01072021	7
08/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01082021	8
09/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01092021	9
10/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01102021	10
11/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01112021	11
12/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01122021	12
13/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01132021	13
14/01/2021 12:00:00 am	2021	1	January	24252	Jan 21	01142021	14

Table: Calendar 1 (365 rows)

For the forecasting part, we chose 95% confidence interval. The forecast showed an positively linear graph and means that for the following months, there would be a chance that the vehicle count would increase.



4.) In downloading a specific data, it can be performed while viewing a page. At the upper right part of the graph, click the three dots, and click “Export Data”.



After pressing “Export Data”, the user can now choose the file path and the file name of the data.

File name:	Total Count of Vehicles Going Out.csv
Save as type:	CSV File (*.csv)