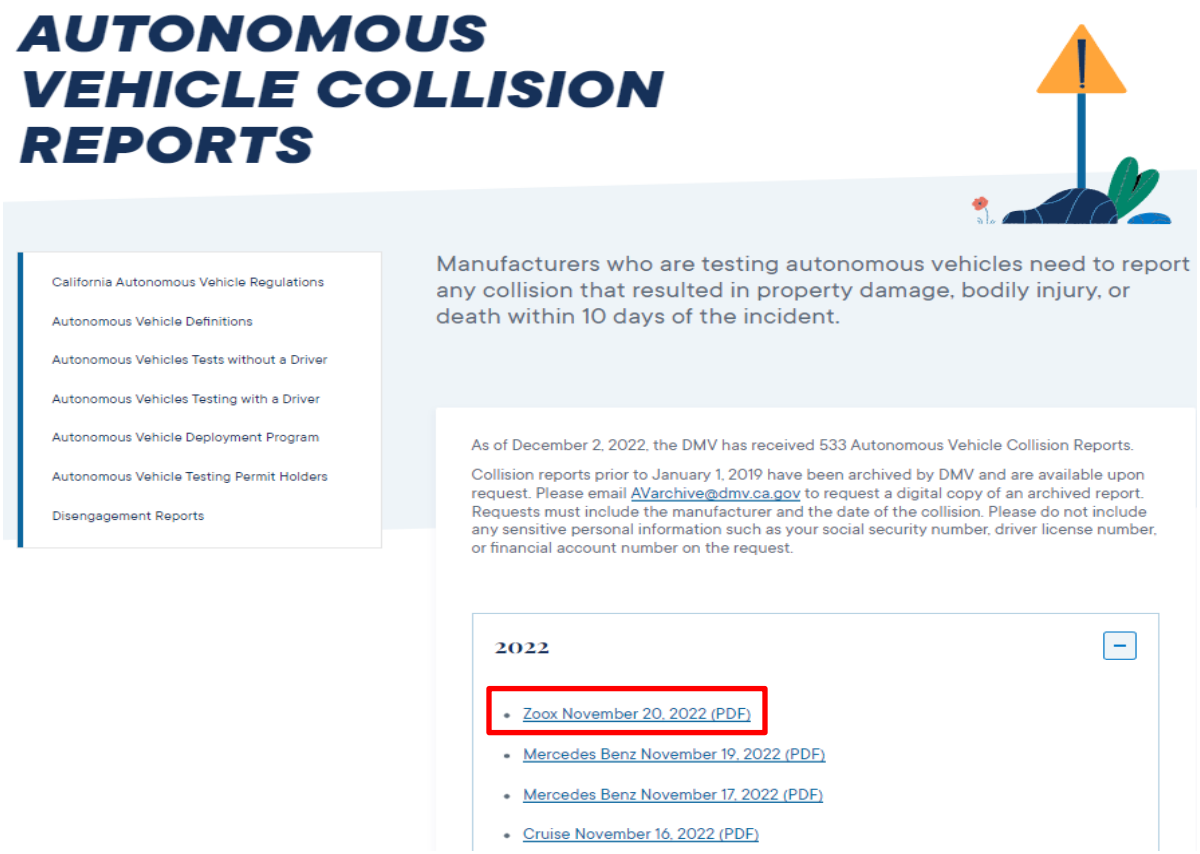
Web Scraping & PowerBI Visualisation – DMV Autonomous Vehicle Incident Reports

This document describes the Python script used to automate the scraping of Autonomous Vehicle (AV) Collision Reports on the State of California Department of Motor Vehicles (DMV) website - https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports.

The Collision Reports come in the form of individual PDF files. To scrape the data, we will need to iterate through the 400 (as of 8 Dec 2022) reports and extract the PDF fields and checkboxes using Python.

In the later part of this document, I will also share some of the visualisations created and insights gathered by ingesting the scraped data into PowerBI.

DMV Webpage (with each Collision Report saved in a PDF)



HTML Text (showing the URL for the Collision Report)

Sample of DMV AV Collision Report

A Public Service Agency

REPORT OF TRAFFIC COLLISION INVOLVING AN AUTONOMOUS VEHICLE

DMV USE ONLY
AVT NUMBER
NAME

Instructions: Please print within the spaces and boxes on this form. If you need to provide additional information on a separate piece of paper(s) or you include a copy of any law enforcement agency report, please check the box to indicate "Additional Information Attached."

- Write **unk (for unknown)** or **none** in any space or box when you do not have the information on the other party involved.
- Give insurance information that is complete and which correctly and *fully* identifies the **company** that issued the insurance policy or surety bond, or whether there is a certificate of self-insurance.
- Place the National Association of Insurance Commissioners (NAIC) number for your Insurance or Surety Company in the boxes provided. The NAIC number should be located on the proof of insurance provided by you company or you can contact your insurer for that information.
- Identify any person involved in the accident (driver, passenger, bicyclist, pedestrian, etc) that you saw was injured *or* complained of bodily injury or know to be deceased.
- Record in the PROPERTY DAMAGE line any damage to telephone poles, fences, street signs, guard post, trees, livestock, dogs, buildings, parked vehicles, etc., including a description of the damage.
- Once you have completed this report, please mail to: Department of Motor Vehicles, Occupational Licensing Branch, P.O. Box 932342, MS: L224, Sacramento, CA 94232-3420

SECTION 1 — MANUFACTURER'S INFORMATION

MANUFACTURER'S NAME	AVT NUMBER
GM Cruise LLC	
BUSINESS NAME	**TELEPHONE NUMBER**
Cruise	( )
STREET ADDRESS	**CITY** / **STATE** / **ZIP CODE**

SECTION 2 — ACCIDENT INFORMATION/VEHICLE 1

DATE OF ACCIDENT	TIME OF ACCIDENT	VEHICLE YEAR	MAKE	MODEL
01/07/2019	06:54 ☐ AM ☒ PM	2019	Chevrolet	Bolt

LICENSE PLATE NUMBER	VEHICLE IDENTIFICATION NUMBER	STATE VEHICLE IS REGISTERED IN
		CA

ADDRESS/LOCATION OF ACCIDENT	CITY	COUNTY	STATE	ZIP CODE
Folsom St. and 11th St.	San Francisco	San Francisco	CA	94103

Vehicle was: ☒ Moving ☐ Stopped in Traffic

Involved in the Accident: ☐ Pedestrian ☐ Bicyclist ☐ Other _____

NUMBER OF VEHICLES INVOLVED: 2

DRIVER'S FULL NAME (*FIRST, MIDDLE, LAST*)	DRIVER LICENSE NUMBER	STATE	DATE OF BIRTH

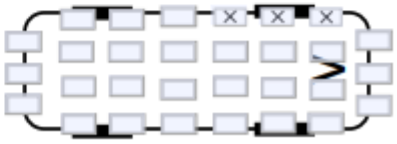
INSURANCE COMPANY NAME OR SURETY COMPANY AT TIME OF ACCIDENT	POLICY NUMBER

COMPANY NAIC NUMBER	POLICY PERIOD
	FROM _____ TO _____

Describe Vehicle Damage

☐ UNK ☐ NONE ☐ MINOR
☒ MOD ☐ MAJOR

Shade in Damaged Area



Go to Page 2

SECTION 3 — OTHER PARTY'S INFORMATION/VEHICLE 2

VEHICLE YEAR	MODEL	
2018	Pcx150	
LICENSE PLATE NUMBER	VEHICLE IDENTIFICATION NUMBER	STATE VEHICLE IS REGISTERED IN
		CA

Vehicle was: ☒ Moving ☐ Stopped in Traffic **Involved in the Accident:** ☐ Pedestrian ☐ Bicyclist ☐ Other _____ NUMBER OF VEHICLES INVOLVED: 2

DRIVER'S FULL NAME (FIRST, MIDDLE, LAST)	DRIVER LICENSE NUMBER	STATE	DATE OF BIRTH
	unk	un	

INSURANCE COMPANY NAME OR SURETY COMPANY AT TIME OF ACCIDENT	POLICY NUMBER

COMPANY NAIC NUMBER	POLICY PERIOD
	FROM _____ TO _____

☐ **Additional information attached.**

SECTION 4 — INJURY/DEATH, PROPERTY DAMAGE

NAME (FIRST, MIDDLE, LAST)

ADDRESS	CITY	STATE	ZIP CODE
unk	unk	un	unk

CHECK ALL THAT APPLY ☒ Injured ☐ Deceased ☒ Driver ☐ Passenger ☐ Bicyclist ☐ Property

NAME (FIRST, MIDDLE, LAST)

ADDRESS	CITY	STATE	ZIP CODE

CHECK ALL THAT APPLY ☐ Injured ☐ Deceased ☐ Driver ☐ Passenger ☐ Bicyclist ☐ Property

PROPERTY DAMAGE

PROPERTY OWNER'S NAME		TELEPHONE NUMBER
		( )
STREET ADDRESS	CITY	STATE ZIP CODE

WITNESS NAME		TELEPHONE NUMBER
		( )
STREET ADDRESS	CITY	STATE ZIP CODE

WITNESS NAME		TELEPHONE NUMBER
		( )
STREET ADDRESS	CITY	STATE ZIP CODE

☐ **Additional information attached.**

SECTION 5 — ACCIDENT DETAILS - DESCRIPTION

☐ Autonomous Mode ☒ Conventional Mode

A Cruise autonomous vehicle ("Cruise AV"), operating in conventional mode, was making a left turn from northeast bound Folsom Street onto northwest bound 11th Street when a scooterist, attempting to pass the Cruise AV on the left, made contact with the front left side of the Cruise AV, damaging the front left fender, radar, and wheel well of the Cruise AV. The scooterist reported injuries and emergency services and the police arrived at the scene, but the scooterist declined medical treatment. No police report was available at the time of the filing of this report.

☐ **Additional information attached.**

Go to Page 3

ITEMS MARKED BELOW FOLLOWED BY AN ASTERISK (*) SHOULD BE EXPLAINED IN THE NARRATIVE

	WEATHER (MARK 1 to 2 ITEMS)	VEH 1	VEH 2	MOVEMENT PRECEDING COLLISION	VEH 1	VEH 2	OTHER ASSOCIATED FACTOR(s) (MARK ALL APPLICABLE)
	A. CLEAR	X	X	A. STOPPED			A. CVC SECTIONS VIOLATED CITED ☐ YES ☐ NO
	B. CLOUDY			B. PROCEEDING STRAIGHT			
	C. RAINING			C. RAN OFF ROAD			
	D. SNOWING			D. MAKING RIGHT TURN			
	E. FOG/VISIBILITY			E. MAKING LEFT TURN	X		
	F. OTHER			F. MAKING U TURN			B. VISION OBSCUREMENT ☐
	G. WIND			G. BACKING			C. INATTENTION* ☐
	LIGHTING			H. SLOWING/STOPPING			D. STOP & GO TRAFFIC ☐
	A. DAYLIGHT			I. PASSING OTHER VEHICLE			E. ENTERING/LEAVING RAMP ☐
	B. DUSK – DAWN			J. CHANGING LANES		X	F. PREVIOUS COLLISION ☐
	C. DARK –STREET LIGHTS	X	X	K. PARKING MANUEVER			G. UNFAMILIAR WITH ROAD ☐
	D. DARK – NO STREET LIGHTS			L. ENTERING TRAFFIC			H. DEFECTIVE WEH EQUIP CITED ☐ YES ☐ NO
	E. DARK –STREET LIGHTS NOT FUNCTIONING*			M. OTHER UNSAFE TURNING			
	ROADWAY SURFACE			N. XING INTO OPPOSING LANE			
	A. DRY	X	X	O. PARKED			I. UNINVOLVED VEHICLE ☐
	B. WET			P. MERGING			J. OTHER* ☐
	C. SNOWY – ICY			Q. TRAVELING WRONG WAY			K. NONE APPARENT ☐
	D. SLIPPERY (MUDDY, OILY, ETC.)			R. OTHER*			L. RUNAWAY VEHICLE ☐
	ROADWAY CONDITIONS (MARK 1 TO 2 ITEMS)			**TYPE OF COLLISION**			
	A. HOLES, DEEP RUT*			A. HEAD-ON			
	B. LOOSE MATERIAL ON ROADWAY			B. SIDE SWIPE		X	
	C. OBSTRUCTION ON ROADWAY*			C. REAR END			
	D. CONSTRUCTION – REPAIR ZONE			D. BROADSIDE			
	E. REDUCED ROADWAY WIDTH			E. HIT OBJECT			
	F. FLOODED*			F. OVERTURNED			
	G. OTHER*			G. VEHICLE/PEDESTRIAN			
	H. NO UNUSUAL CONDITIONS	X	X	H. OTHER*			

SECTION 6 — CERTIFICATION

I certify (or declare) under penalty of perjury under the laws of the State of California that the foregoing is true and correct.

I further certify that I am the authorized Administrator of the program for the above named employer.

PROGRAM DIRECTOR/AUTHORIZED REPRESENTATIVE PRINTED NAME AND TITLE	TELEPHONE NUMBER
Kevin Chu, Director of AV Engineering	( )
SIGNATURE	**DATE SIGNED**
X	03/13/2019

OL 316 (REV. 2/2017) **WWW**

Print Clear Form

First, we import the relevant Python libraries. Next, send a **GET** request to the DMV webpage and create a response object that stores the request response.

Using the BeautifulSoup library, we can parse the html to get the URLs of the various AV Collision Reports.

Import Python Libraries and Parsing Webpage HTML

```
import requests
import PyPDF2
import json
import fitz
import pandas as pd
import io
from io import BytesIO
from bs4 import BeautifulSoup
```

```
response = requests.get("https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-coll
```

```
dmv_web_page = response.text
```

```
soup = BeautifulSoup(dmv_web_page, "html.parser")
```

```
# Get all spans with the word file
dmv_report = soup.select('a[href*=file]')
print(dmv_report)
```

```
[<a href="/portal/file/zoox_112022-pdf/">Zoox November 20, 2022</a>, <a href="/portal/file/mercedes-benz_111922-pdf/">Mercede
s Benz November 19, 2022</a>, <a href="/portal/file/mercedes-benz_111722-pdf/">Mercedes Benz November 17, 2022</a>, <a href
="/portal/file/cruise_111622-pdf/">Cruise November 16, 2022</a>, <a href="/portal/file/zoox_111422-pdf/">Zoox November 14, 20
22</a>, <a href="/portal/file/waymo_110822-pdf/">Waymo November 8, 2022</a>, <a href="/portal/file/waymo_110322-pdf/">Waymo N
ovember 3, 2022</a>, <a href="/portal/file/zoox_110422-pdf/">Zoox November 4, 2022</a>, <a href="/portal/file/waymo_102122-pd
f/">Waymo October 21, 2022</a>, <a href="/portal/file/apple_102022-pdf/">Apple October 20, 2022</a>, <a href="/portal/file/ap
ple_101022-pdf/">Apple October 10, 2022</a>, <a href="/portal/file/zoox_101422-pdf/">Zoox October 14, 2022</a>, <a href="/por
tal/file/zoox_100322-pdf/">Zoox October 3, 2022</a>, <a href="/portal/file/ghost_100722-pdf/">Ghost Autonomy Inc October 7, 2
022</a>, <a href="/portal/file/argo_092222-pdf/">Argo AI September 22, 2022</a>, <a href="/portal/file/waymo_091722-pdf/">Way
mo September 17, 2022</a>, <a href="/portal/file/waymo_091522-pdf/">Waymo September 15, 2022</a>, <a href="/portal/file/zoox_
091422-pdf/">Zoox September 14, 2022</a>, <a href="/portal/file/waymo_091022-pdf/">Waymo September 10, 2022</a>, <a href="/po
rtal/file/cruise_090522-pdf/">Cruise September 5, 2022</a>, <a href="/portal/file/cruise_090422-pdf/">Cruise September 4, 202
2</a>, <a href="/portal/file/zoox_090122-pdf/">Zoox September 1, 2022</a>, <a href="/portal/file/cruise_090122-pdf/">Cruise S
eptember 1, 2022</a>, <a href="/portal/file/zoox_083022-pdf/">Zoox August 30, 2022</a>, <a href="/portal/file/zoox_082322-pd
f/">Zoox August 23, 2022</a>, <a href="/portal/file/waymo_082522-pdf/">Waymo August 25, 2022</a>, <a href="/portal/file/cruis
e_081622-pdf/">Cruise August 16, 2022</a>, <a href="/portal/file/waymo_081622_1-pdf/">Waymo August 16, 2022 (1)</a>, <a href
="/portal/file/waymo_081622_2-pdf/">Waymo August 16, 2022 (2)</a>, <a href="/portal/file/waymo_081422-pdf/">Waymo August 14,
2022</a>, <a href="/portal/file/waymo_081322-pdf/">Waymo August 13, 2022</a>, <a href="/portal/file/waymo_081022-pdf/">Waymo
August 10, 2022</a>, <a href="/portal/file/woven_planet_080822-pdf/">Woven Planet August 8, 2022</a>, <a href="/portal/file/c
```

Extract URL from HTML text

```
dmv_report_links = []

for link in dmv_report:
   dmv_report_links.append(link.get('href'))

print(dmv_report_links)
```

```
['/portal/file/zoox_112022-pdf/', '/portal/file/mercedes-benz_111922-pdf/', '/portal/file/mercedes-benz_111722-pdf/', '/portal/file/cruise_111622-pdf/', '/portal/file/zoox_111422-pdf/', '/portal/file/waymo_110822-pdf/', '/portal/file/waymo_110322-pdf/', '/portal/file/zoox_110422-pdf/', '/portal/file/waymo_102122-pdf/', '/portal/file/apple_102022-pdf/', '/portal/file/apple_101022-pdf/', '/portal/file/zoox_101422-pdf/', '/portal/file/zoox_100322-pdf/', '/portal/file/ghost_100722-pdf/', '/portal/file/argo_092222-pdf/', '/portal/file/waymo_091722-pdf/', '/portal/file/waymo_091522-pdf/', '/portal/file/zoox_091422-pdf/', '/portal/file/waymo_091022-pdf/', '/portal/file/cruise_090522-pdf/', '/portal/file/cruise_090422-pdf/', '/portal/file/zoox_090122-pdf/', '/portal/file/cruise_090122-pdf/', '/portal/file/zoox_083022-pdf/', '/portal/file/zoox_082322-pdf/', '/portal/file/waymo_082522-pdf/', '/portal/file/cruise_081622-pdf/', '/portal/file/waymo_081622_1-pdf/', '/portal/file/waymo_081622_2-pdf/', '/portal/file/waymo_081422-pdf/', '/portal/file/waymo_081322-pdf/', '/portal/file/waymo_081022-pdf/', '/portal/file/woven_planet_080822-pdf/', '/portal/file/cruise_080222-pdf/', '/portal/file/waymo_072422-pdf/', '/portal/file/zoox_072222-pdf/', '/portal/file/cruise_071822-pdf/', '/portal/file/cruise_071622-pdf/', '/portal/file/cruise_071422-pdf/', '/portal/file/zoox_070822-pdf/', '/portal/file/waymo_070522-pdf/', '/portal/file/waymo_070322-pdf/', '/portal/file/waymo_070222_1-pdf/', '/portal/file/waymo_070222_2-pdf/', '/portal/file/waymo_070122-pdf/', '/portal/file/cruise_062922-pdf/', '/portal/file/zoox_062822-pdf/', '/portal/file/waymo_062122-pdf/', '/portal/file/apple_061422-pdf/', '/portal/file/cruise_061722-pdf/', '/portal/file/zoox_061122-pdf/', '/portal/file/zoox_060822-pdf/', '/portal/file/waymo_060522-pdf/', '/portal/file/cruise_060322-pdf/', '/portal/file/cruise_060222-pdf/', '/portal/file/waymo_060122-pdf/', '/portal/file/zoox_060122-1-pdf/', '/portal/file/zoox_060122-2-pdf/', '/portal/file/waymo_052722-pdf/', '/portal/file/cruise_052522-pdf/', '/portal/file/zoox_053122-pdf/', '/portal/file/waymo_052122-pdf/', '/portal/file/zoox_052022-pdf/', '/portal/file/mercedes-benz_051922-pdf/', '/portal/file/waymo_051722-pdf/', '/portal/file/pony-ai_051722-pdf/', '/portal/file/motional_051722-pdf/', '/portal/file/cruise_051322-pdf/', '/portal/fil
```

As of 8 Dec 2022, there are close to 400 AV Collision Reports. Downloading them one by one and inputting the data in the PDF files manually would be time consuming. Let's make use of Python to automate this effort.

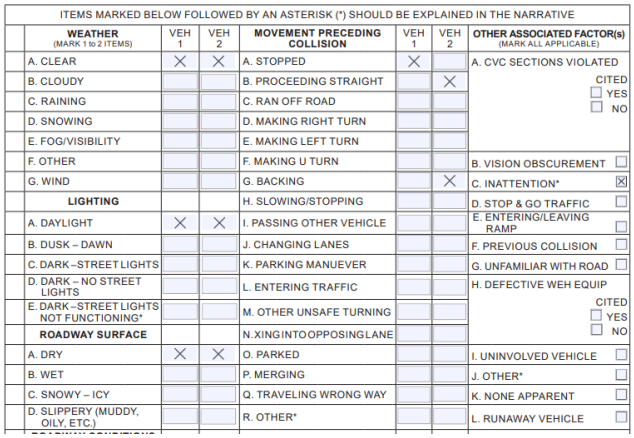
Create an Empty Dataframe with the Columns that You Want

```
# create empty dataframe
df = pd.DataFrame({
    'BuSINESS NAME': [],
    'DATE Of ACCIDENT': [],
    'Time of Accident': [],
    'VEhICLE YEAR' : [],
    'MAkE' : [],
    'MODEL' : [],
    'NuMBER Of VEhICLES INVOLVED' : []
})

# print dataframe
print("\n  ***  Original DataFrames  **  \n")
print(df)


# keys that you want to retain (for filtering dictionary later before appending to dataframe)
keys = ['BuSINESS NAME', 'DATE Of ACCIDENT', 'Time of Accident',
'VEhICLE YEAR', 'MAkE', 'MODEL', 'NuMBER Of VEhICLES INVOLVED']
```

After creating an empty dataframe, let's iterate over the 400 AV Collision Report URL, and extract the data from the PDF. Save the PDF data in the variable 'dictionary'.

```
for link in dmv_report_links:
    url = 'https://www.dmv.ca.gov' + link

    response = requests.get(url)
    f = io.BytesIO(response.content)

    with f as data:
        #read it and get the pages number
        pdfreader=PyPDF2.PdfFileReader(data)
        x=pdfreader.numPages
        pageobj=pdfreader.getPage(0)

        # extract the pdf to text
        text=pageobj.extractText()

        # extract pdf form fields
        dictionary = pdfreader.getFormTextFields()
```

Note that data from checkboxes such as the one below cannot be extracted using the above method.



ITEMS MARKED BELOW FOLLOWED BY AN ASTERISK (*) SHOULD BE EXPLAINED IN THE NARRATIVE

	WEATHER (MARK 1 to 2 ITEMS)	VEH 1	VEH 2	MOVEMENT PRECEDING COLLISION	VEH 1	VEH 2	OTHER ASSOCIATED FACTOR(s) (MARK ALL APPLICABLE)
	A. CLEAR	X	X	A. STOPPED	X		A. CVC SECTIONS VIOLATED
	B. CLOUDY			B. PROCEEDING STRAIGHT		X	CITED ☐ YES ☐ NO
	C. RAINING			C. RAN OFF ROAD			
	D. SNOWING			D. MAKING RIGHT TURN			
	E. FOG/VISIBILITY			E. MAKING LEFT TURN			
	F. OTHER			F. MAKING U TURN			B. VISION OBSCUREMENT ☐
	G. WIND			G. BACKING		X	C. INATTENTION* ☒
	LIGHTING			H. SLOWING/STOPPING			D. STOP & GO TRAFFIC ☐
	A. DAYLIGHT	X	X	I. PASSING OTHER VEHICLE			E. ENTERING/LEAVING RAMP ☐
	B. DUSK – DAWN			J. CHANGING LANES			F. PREVIOUS COLLISION ☐
	C. DARK–STREET LIGHTS			K. PARKING MANUEVER			G. UNFAMILIAR WITH ROAD ☐
	D. DARK – NO STREET LIGHTS			L. ENTERING TRAFFIC			H. DEFECTIVE WEH EQUIP
	E. DARK–STREET LIGHTS NOT FUNCTIONING*			M. OTHER UNSAFE TURNING			CITED ☐ YES ☐ NO
	ROADWAY SURFACE			N. XING INTO OPPOSING LANE			
	A. DRY	X	X	O. PARKED			I. UNINVOLVED VEHICLE ☐
	B. WET			P. MERGING			J. OTHER* ☐
	C. SNOWY – ICY			Q. TRAVELING WRONG WAY			K. NONE APPARENT ☐
	D. SLIPPERY (MUDDY, OILY, ETC.)			R. OTHER*			L. RUNAWAY VEHICLE ☐

Hence, we will use the pdfreader.getFields() method to call the data and save it in the variable ‘checkbox’.

```
        checkbox = pdfreader.getFields()


        # extract checkbox fields from PDF

        # ---- #
        time_of_day = ''

        if checkbox.get('AM').value == '/ ':
            time_of_day = 'AM'
        else:
            time_of_day = 'PM'

        # ---- #
        vehicle_damage = ''

        if checkbox.get('Unknown').value == '/Yes':
            vehicle_damage = 'unknown'
        elif checkbox.get('None').value == '/Yes':
            vehicle_damage = 'none'
        elif checkbox.get('minor').value == '/Yes':
            vehicle_damage = 'minor'
        elif checkbox.get('Moderate').value == '/Yes':
            vehicle_damage = 'moderate'
        elif checkbox.get('major').value == '/Yes':
            vehicle_damage = 'major'
        else:
            vehicle_damage = 'unknown'

        # ---- #
        moving_stopped = ''

        if checkbox.get('Moving').value == '/ ':
            moving_stopped = 'moving'
        elif checkbox.get('Stopped in Traffic').value == '/ ':
            moving_stopped = 'stationary'
        else:
            moving_stopped = 'unknown'

        # ---- #
        autonomous_conventional = ''

        if checkbox.get('Autonomous Mode').value == '/ ':
            autonomous_conventional = 'autonomous'
        else:
            autonomous_conventional = 'conventional'

        # ---- #
        weather = ''

        if checkbox.get('WEATHER A 1').value == '/Yes':
            weather = 'clear'
        elif checkbox.get('WEATHER B 1').value == '/Yes':
            weather = 'cloudy'
        elif checkbox.get('WEATHER C 1').value == '/Yes':
            weather = 'raining'
        elif checkbox.get('WEATHER D 1').value == '/Yes':
            weather = 'snowing'
        elif checkbox.get('WEATHER E 1').value == '/Yes':
            weather = 'fog/visibility'
        elif checkbox.get('WEATHER F 1').value == '/Yes':
            weather = 'other'
        elif checkbox.get('WEATHER G 1').value == '/Yes':
            weather = 'wind'
        else:
            weather = 'unknown'
```

```
        # ---- #
        lighting = ''

        if checkbox.get('LIGHTING A 1').value == '/Yes':
            lighting = 'daylight'
        elif checkbox.get('LIGHTING B 1').value == '/Yes':
            lighting = 'dusk-dawn'
        elif checkbox.get('LIGHTING C 1').value == '/Yes':
            lighting = 'dark-street_lights'
        elif checkbox.get('LIGHTING D 1').value == '/Yes':
            lighting = 'dark-no_street_lights'
        elif checkbox.get('LIGHTING E 1').value == '/Yes':
            lighting = 'dark-street_lights_malfunction'
        else:
            lighting = 'unknown'

        # ---- #
        roadway = ''

        if checkbox.get('ROADWAY A 1').value == '/Yes':
            roadway = 'dry'
        elif checkbox.get('ROADWAY B 1').value == '/Yes':
            roadway = 'wet'
        elif checkbox.get('ROADWAY C 1').value == '/Yes':
            roadway = 'snowy_ice'
        elif checkbox.get('ROADWAY D 1').value == '/Yes':
            roadway = 'slippery_mud_oil'
        else:
            roadway = 'unknown'

        # ---- #
        movement = []

        movement_list = ['MOVEMENT A 1', 'MOVEMENT  B 1', 'MOVEMENT C 1', 'MOVEMENT  D 1',
         'MOVEMENT  E 1', 'MOVEMENT  F 1', 'MOVEMENT  G 1', 'MOVEMENT  H 1',
         'MOVEMENT  I 1', 'MOVEMENT J 1', 'MOVEMENT  K 1', 'MOVEMENT  L 1',
         'MOVEMENT  M 1', 'MOVEMENT  N 1', 'MOVEMENT  O 1', 'MOVEMENT  P 1',
         'MOVEMENT  Q 1', 'MOVEMENT  R 1']

        movement_name = ['stopped', 'proceeding_straight', 'ran_off_road', 'right_turn', 'left_turn', 'u_turn', 'backing'
                       , 'slowing_stopping', 'passing_other_vehicle', 'changing_lane', 'parking', 'entering_traffic'
                       , 'unsafe_turning', 'crossing_opposing_lane', 'parked', 'merging', 'wrong_way', 'others']

        i=0

        for key in movement_list:
            if checkbox.get(key).value == '/Yes':
                movement.append(movement_name[i])
            i = i+1


        # filter out the keys you do not want
        filtered_dict = dict((k, dictionary[k]) for k in keys if k in dictionary)

        # Insert the checkbox fields into dictionary
        filtered_dict.update([('time_of_day', time_of_day),('vehicle_damage', vehicle_damage),
                              ('moving_stopped', moving_stopped), ('autonomous_conventional', autonomous_conventional),
                              ('weather', weather), ('lighting', lighting), ('roadway', roadway), ('movement', movement)])
```

With the above codes, we can obtain the PDF data (be it form field or checkboxes) for each URL (i.e. AV Collision Report). We then append this data into the empty dataframe that we have created earlier.

```
    # combined data
    df = df.append(filtered_dict, ignore_index=True, sort=False)
```

Sample Table Showing the Extracted Data

```
df.head(193)
```

	company_name	date	time	vehicle_year	make	model	num_of_vehicles_involved	autonomous_conventional	lighting	movemer
0	Zoox	11/20/2022	1:24	2016	Toyota	Highlander	2	autonomous	daylight	[stoppe
1	MERCEDES-BENZ RESEARCH & DEVELOPMENT NORTH AME...	11/19/2022	10:45	2021	Mercedes-Benz	S 450	2	conventional	daylight	[left_tur entering_traffi
2	MERCEDES-BENZ RESEARCH & DEVELOPMENT NORTH AME...	11-17-2022	5:20	2021	Mercedes	S450	2	conventional	dark-street_lights	[proceeding_straigh changing_lan
3	Cruise	11/16/2022	2:59	2023	Cruise	AV	2	conventional	daylight	[proceeding_straigh merging, other
4	Zoox	11/14/2022	8:22	2016	Toyota	Highlander	2	autonomous	dark-street_lights	[slowing_stoppin
...	...	...	...	...	...	...	...	...	...	...
190	Waymo LLC	8/31/2021	7:30	2021	Jaguar	I-Pace	2	conventional	daylight	[entering_traffi
191	Waymo LLC	8/31/2021	5:43	2021	Jaguar	I-Pace	2	conventional	daylight	[backin
192	Cruise	08/30/2021	08:40	2020	Chevrolet	Bolt	2	conventional	dark-street_lights	[parke
193	Cruise	08/28/2021	21:05	2020	Chevrolet	Bolt	1	autonomous	dark-street_lights	[proceeding_straigh
194	Waymo LLC	08/27/2021	4:15	2021	Jaguar	I-Pace	2	autonomous	daylight	[stoppe

As the accident time is recorded in the format below, we can define a method and apply it to each individual row of the dataframe to convert the time to a 24-hour format (readable by PowerBI).


```
def convert24(row):

    # is AM and first two elements are 12
    if row['time_of_day'] == "AM" and row['time_split'][0] == "12":
        row['time_split'][0] = "00"

    # remove the AM
    elif row['time_of_day'] == "AM":
        row['time_split'][0] = row['time_split'][0]

    # is PM and first two elements are 12
    elif row['time_of_day'] == "PM" and row['time_split'][0] == "12":
        row['time_split'][0] = row['time_split'][0]

    else:
        # add 12 to hours and remove PM
        row['time_split'][0] = str(int(row['time_split'][0]) + 12)

    return ':'.join(row['time_split'])
```

```
df2['time_24hr'] = df2.apply(lambda row : convert24(row), axis = 1)
```

Sample Table with 24-hour format

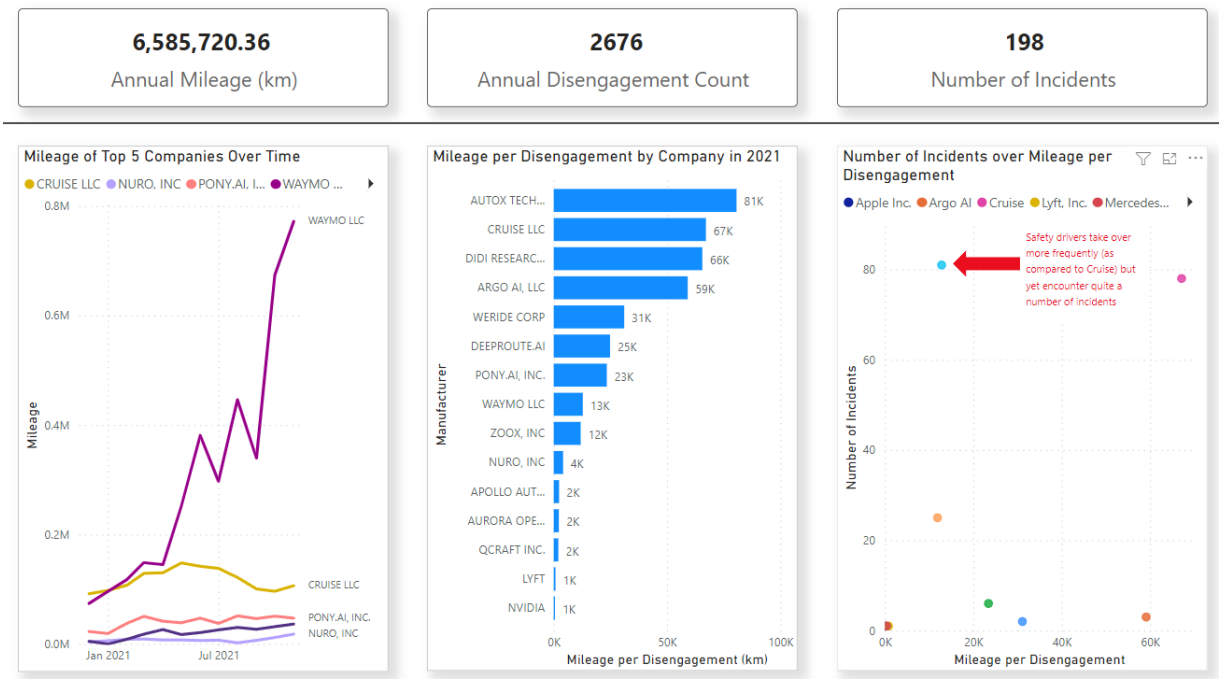
	company_name	date	time	time_of_day	time_24hr
0	Zoox	11/20/2022	1:24	PM	13:24
1	MERCEDES-BENZ RESEARCH & DEVELOPMENT NORTH AME...	11/19/2022	10:45	AM	10:45
2	MERCEDES-BENZ RESEARCH & DEVELOPMENT NORTH AME...	11-17-2022	5:20	PM	17:20

Now that we have our table, we can export as excel file and ingest the data into PowerBI for visualisation.

```
1  df2.to_excel("dmv_report.xlsx")
```

PowerBI Visualisation

PowerBI Report Page 1 (Mileage vs. Disengagement Count)



As we can see from the above visualisation, **Waymo has been increasing its on-road activities** over the past year. In November 2021, it clocked close to 800,000km of mileage. That said, its **frequency of disengagement is rather high** as compared to its competitors (i.e. 13,000km as compared to Cruise's 67,000km).

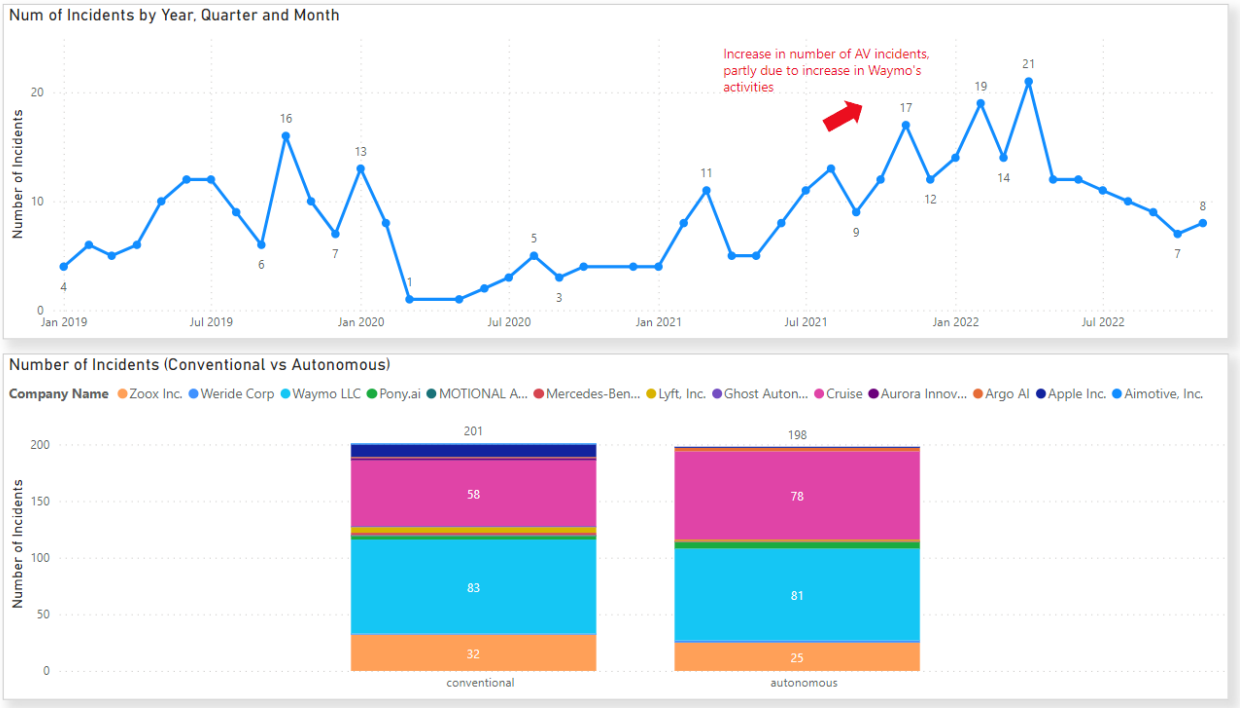
Even with such high disengagement rate, Waymo is involved in a relatively high number of AV incidents (autonomous). **Almost 50% of the total incidents belong to Waymo.**

From the above scatter plot of Number of Incidents over Mileage per Disengagement, we cannot really see any correlation between the two parameters. It will be good if DMV can provide the VIN number of the vehicle involved in the accident. We can then use this to plot the incident count vs. mileage per disengagement for each vehicle (instead of for each company).

If we look at the number of incidents over the years, we can see that **incidents have dropped during the onset of COVID-19 (in March 2020)**, possibly due to the reduced activities?

The number of **conventional vs. autonomous incidents is about 50%-50%**. From this, can we say that autonomous vehicles are not more unsafe as compared to conventional vehicles with drivers?

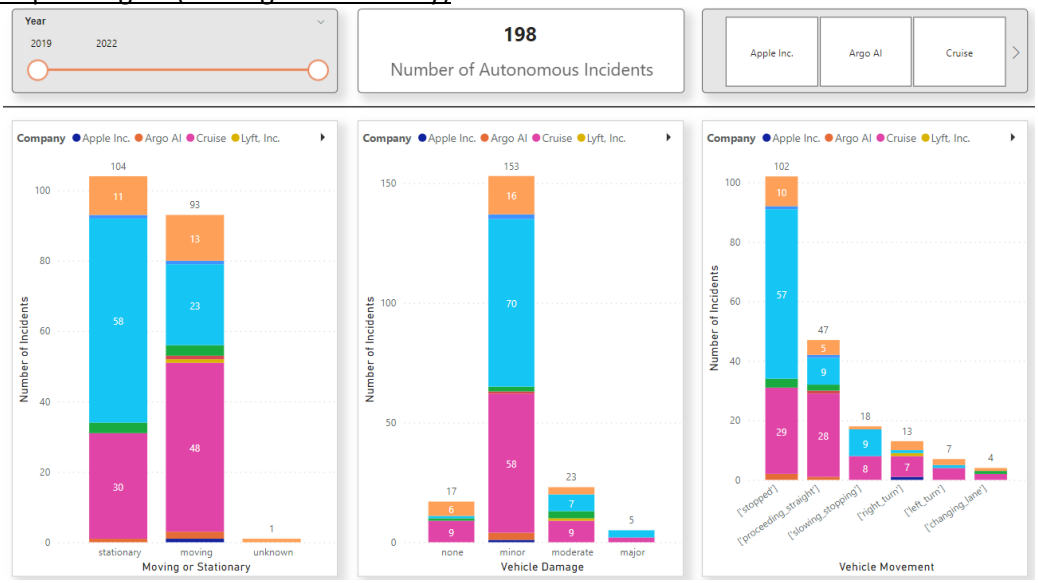
PowerBI Report Page 2 (Num of Incidents over Time)



Next, we look at whether there is a trend in the AVs' movement vs. them getting into accidents. A key observation from the visualisation below is that as compared to Waymo (light blue), a **huge proportion of Cruise's incidents (pink) happened while the vehicle is moving**.

Evident from the Vehicle Movement chart, we can see that the number of incidents while Cruise's AVs were **stopped vs. proceeding straight is similar 50%-50%**. Would be useful to dive deeper into what happened while the AVs were moving straight (was it due to a design issue of certain autonomous technology).

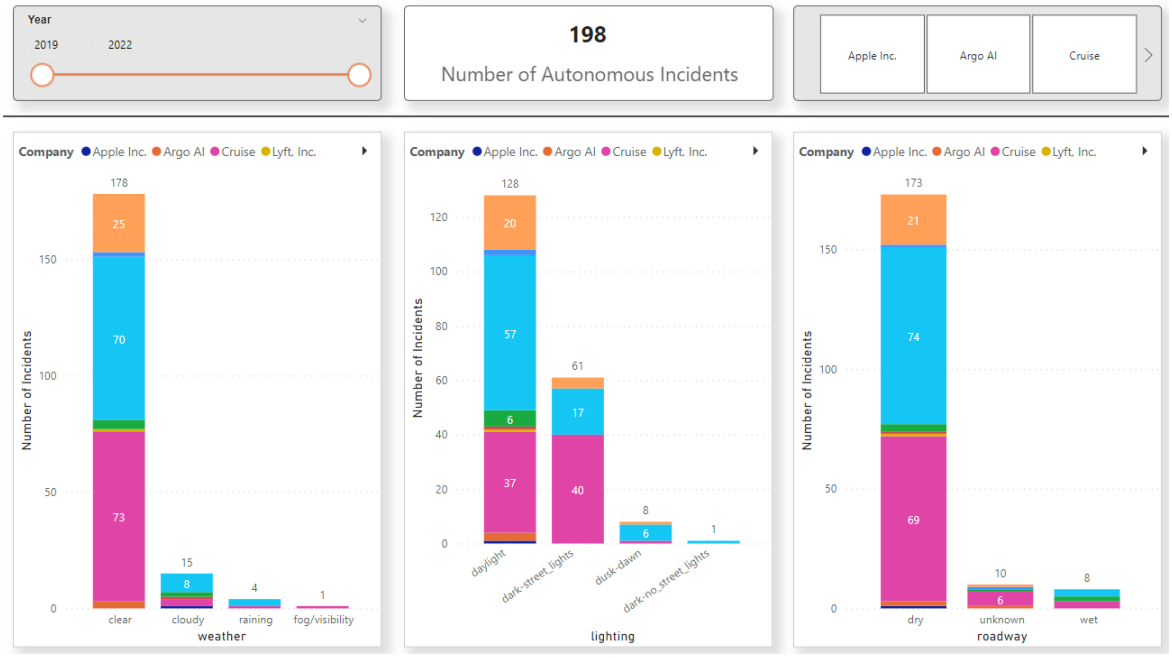
PowerBI Report Page 3 (Moving vs. Stationary)



Filter to show Waymo's and Cruise's incidents

Most of the incidents happen in daylight, and when the weather is clear and road is dry. We can observe from the Lighting chart below that quite **a number of Cruise's incidents (pink) happened while the lighting is dark** with street lights.

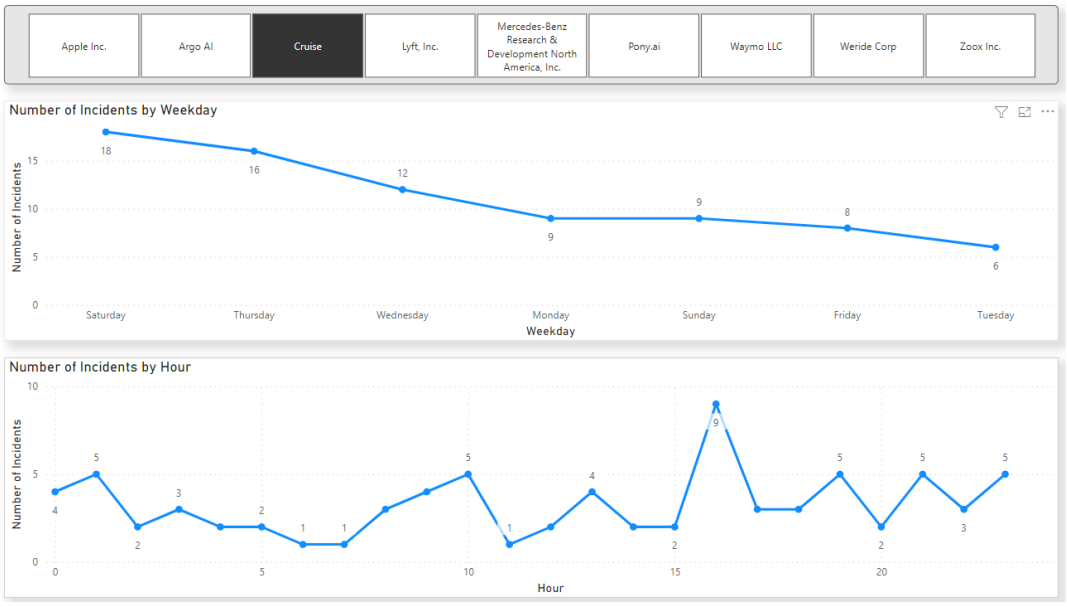
PowerBI Report Page 4 (Weather Lighting Roadway)



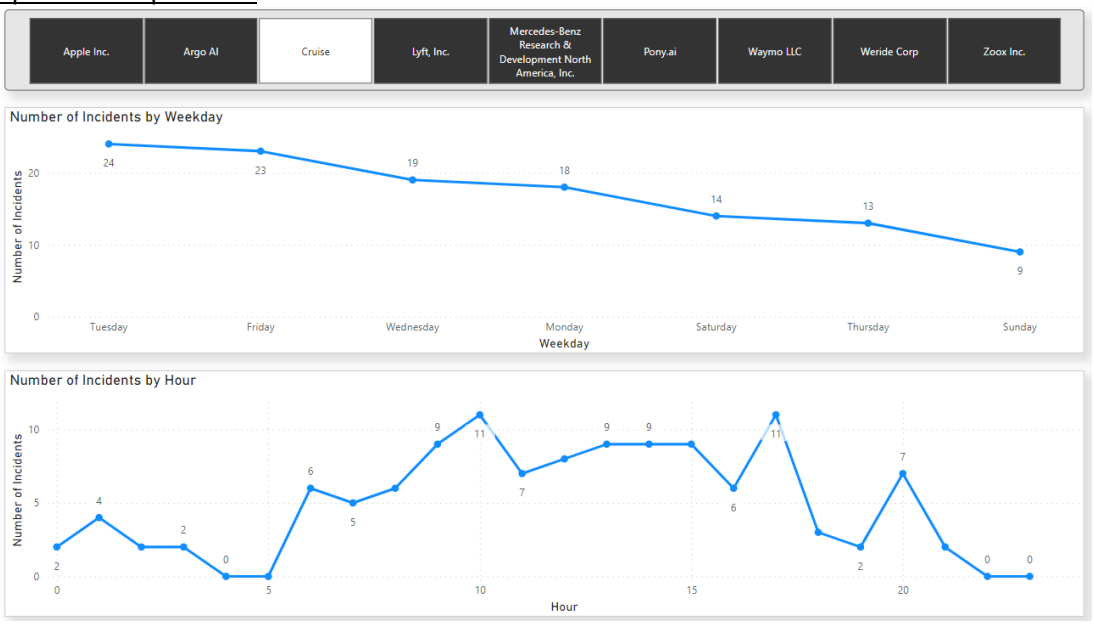
To see if this was due to increased activities during the night or an issue with Cruise's "night vision", we would require more data.

Indeed, by comparing the number of incidents by hour charts, we see that **Cruise's AVs get into incidents more frequently than others during the wee hours (i.e. from 9pm to 5am)**.

PowerBI Report Page 5 (Num of AV Incidents by Weekday/Hour) - Cruise



All Companies except Cruise



An interesting observation: **The number of AV incidents is lowest on Sunday.**