

大數據資料處理

頭城交流道

第九組 (:

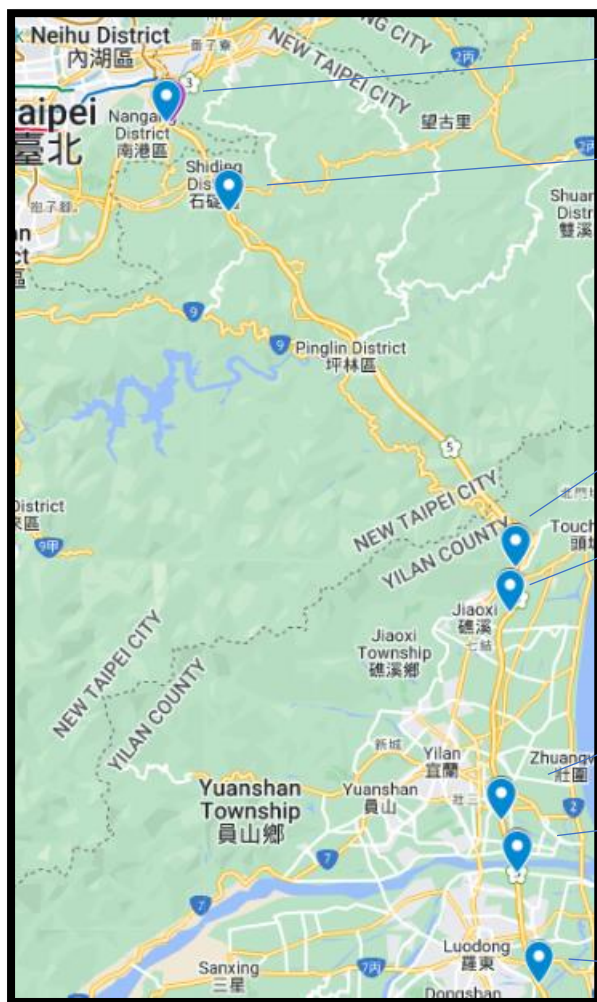
- 110021036 陳奕辰
- 110021128 謝欣恩
- 110021184 劉妍群

Contents

地圖：頭城交流道.....	2
目標與目的.....	4
車子類型	4
資料處理	5
電子匝道	7
AWS 花的錢.....	11
加分	12
自動下載資料- 轉到 S3.....	12
改變 cloud9 影碟大小.....	17
3.檢查 s3 資料有多少	20
它使用一種特殊的工具（分頁器）來流覽所有檔，並隨時將它們的編號相加。完成後，它會告訴您該存儲區域中有多少個檔。這就像計算特定在線存儲盒中的所有物品一樣。	20
4.下載到 cloud9 檢查有多少資料.....	20
5. 直接從 cloud9 的資料做出 csv.....	21
6. 用 excel 整理資料	22
7. 比較 9 月跟 10 月.....	23
8. 比較每個星期一個星期.....	24
會議討論	33
小組分工+均衡.....	34
學習心得	34

地圖：頭城交流道

- 05F-S



南港系統=>石碇
(05F0000S)

石碇=>坪林交控中心
(05F0055S)

坪林交控中心專用道=>
頭城
(05F0287S)

頭城=>宜蘭(四城、大福)
(05F0309S)

宜蘭(壯圍)=>羅東()
(05F0439S)

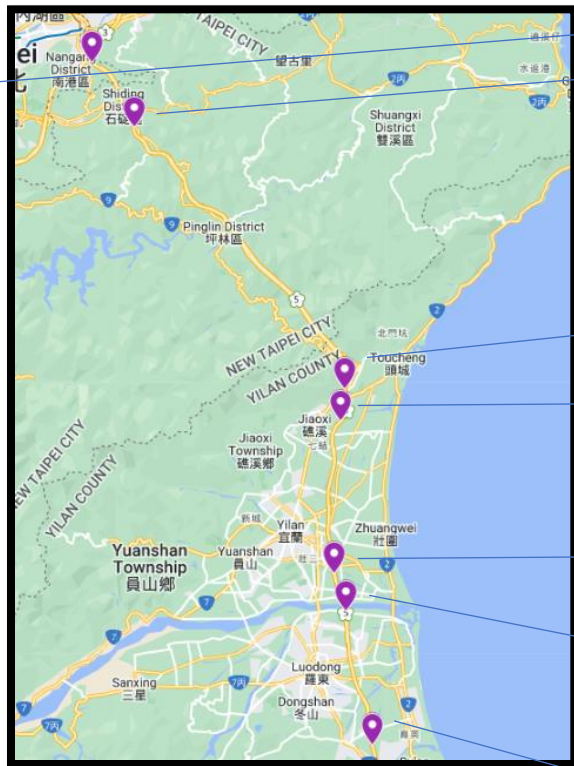
羅東=>蘇澳
(05F0494S)

宜蘭(四城、大福)=>宜蘭
(壯圍)
(05FR113S)

我們選的 4 個 電子匝道代碼：

北上	南下	北上	南下
05F0287N	05F0309S	05F0309N	05F0287S

● 05F-N



石碇=>南港系統
(05F0001N)

坪林交控中心專用道=>石碇
(05F0055N)

頭城=>坪林交控中心專用道
(05F0287N)

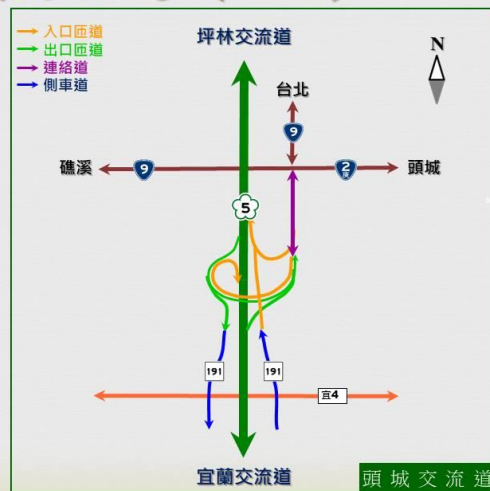
宜蘭(四城、大福)=>頭城
(05F0309N)

羅東=>宜蘭(壯圍)
(05F0438N)

蘇澳=>羅東
(05F0528N)

宜蘭(壯圍)=>宜蘭(四城、大福)
(05FR143N)

頭城交流道 (30K)



我們在這裡選了頭城交流道, 這個交流道在台北和宜蘭中間靠近冬山鄉. 他有頭城, 礁溪 主要服務區域/鏈接道路. 南翔與北向出口預告地名是他有頭城, 礁溪

網頁: [頭城交流道](#)

目標與目的

為什麼我們要選這個頭城交流道來報告？我選擇走頭城路線，是因為交通部的高速公路局進行了大規模的緩解工程。鑑於頭城交流道在假日時的持續交通擁擠，交通部投資超過 3.67 億新台幣，於 2015 年 4 月啟動了一項建設工程。該項目包括從縣道 191 號起點開始，新增了一個北上入口和南下出口匝道。同時，對附近區域進行擴建，並與宜蘭 4 線相連，確保更加順暢的行車體驗。這項工程於 2018 年 10 月 28 日完工，標誌著交通狀況的顯著改善，因此選擇縣道 191 號而非國道 5 號是一個理性的決定。

頭城提供以下交通路線以促進車流順暢：

1. 從宜蘭方向北上高速公路的車流：

路線：礁溪路（台 9 號省道）→七結路（宜蘭 6 號）→國道 5 號下層（縣道 191 號）→新匝道。

2. 從頭城方向北上高速公路的車流：

路線：青雲路（台灣省道 2 號乙）→三和路（縣道 191 甲）→興農路（宜蘭 4 號）→國道 5 號下層（縣道 191 號）→新匝道。

3. 在礁溪市區內的車流（從青雲路/三和路到礁溪路/七結路）：

建議：繼續使用現有的台 9 號省道匝道。

這些路線旨在優化車流並減少擁塞，為從宜蘭、頭城或礁溪市區來的司機提供了選擇。縣道 191 號上的新匝道提供了更高效和直接的路線，特別在尖峰時段更為有益。

車子類型

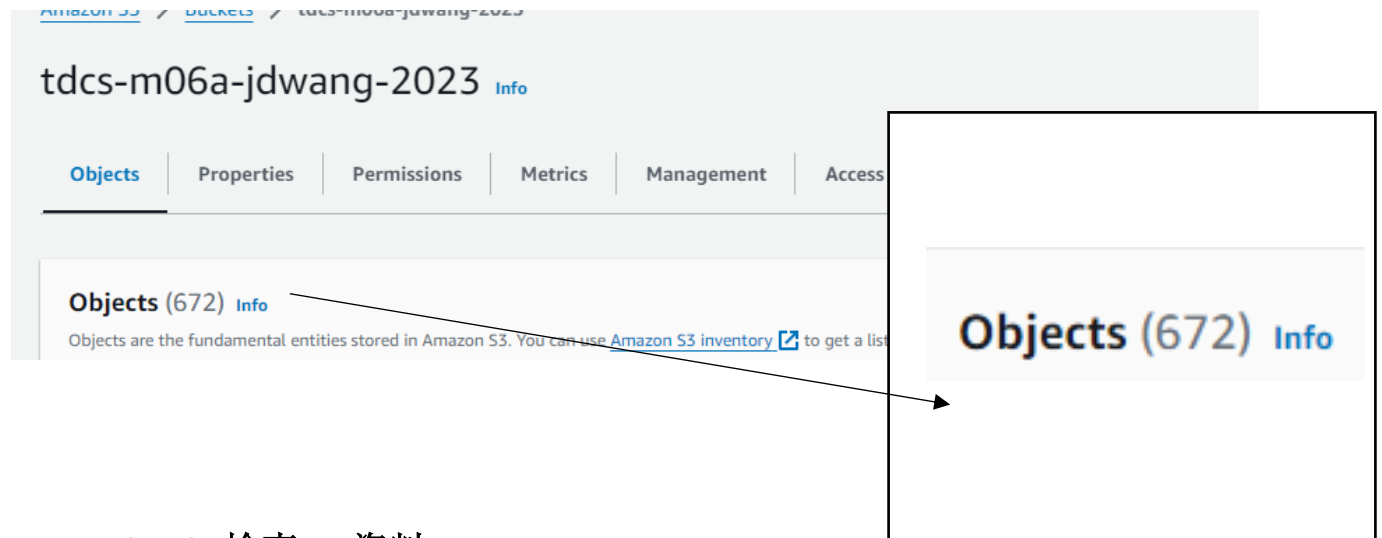
31	32	41	42	5
小客車	小貨車	大客車	大貨車	聯結車

資料處理

跟著日期 09/25-10/22 總共 28 天, 28 天*24 小時 == 672 資料

- **S3**

我們在這裡用一個月的資料在 S3 裡面大概有 672 資料



- **Cloud9 檢查 S3 資料**

我們在這裡用 cloud 9 來檢查資料數量因為 S3 有時候沒更新

```
1 import boto3
2
3 def get_total_objects_in_bucket(bucket_name):
4     # Create an S3 client
5     s3 = boto3.client('s3')
6
7     # Create a paginator
8     paginator = s3.get_paginator('list_objects_v2')
9
10    # Initialize the total count
11    total_objects = 0
12
13    for page in paginator.paginate(Bucket=bucket_name):
14        # Update the total count
15        total_objects += page.get('KeyCount', 0)
16
17    print(f"Total objects in bucket '{bucket_name}': {total_objects}")
18
19 # Example usage:
20 bucket_name = "tdcs-m06a-jdwang-2023"
21 get_total_objects_in_bucket(bucket_name)
22
```

Run Command: Cloud9_boto3_jdwang2022_12_17

total objects in bucket 'tdcs-m06a-jdwang-2023': 672

process exited with code: 0

在這裡我們可以看到 total number of file :672

- 匯入後 cloud 9 檢查

匯入後再確認資料進去的數量是對的 在裡面有寫到 672 資料

```
1 import os
2
3 BUCKET_NAME = 'tdcs-m06a-jdwang-2023'
4 OutputDir = './' + BUCKET_NAME + '/'
5
6 if os.path.exists(OutputDir):
7     print(OutputDir + " Existed!")
8 else:
9     os.mkdir(OutputDir)
10    print("Directory '%s' created" % OutputDir)
11
12 # ... (rest of your script)
13
14 # Count the total number of files in the directory
15 file_count = len([f for f in os.listdir(OutputDir) if os.path.isfile(os.path.join(OutputDir, f))])
16
17 print(f'Total number of files in directory {OutputDir}: {file_count}')
18
```

```
./tdcs-m06a-jdwang-2023/ Existed!
Total number of files in directory ./tdcs-m06a-jdwang-2023/: 672
```

在這裡我們可以看到 total number of files =672

- 下載出來確認資料大小

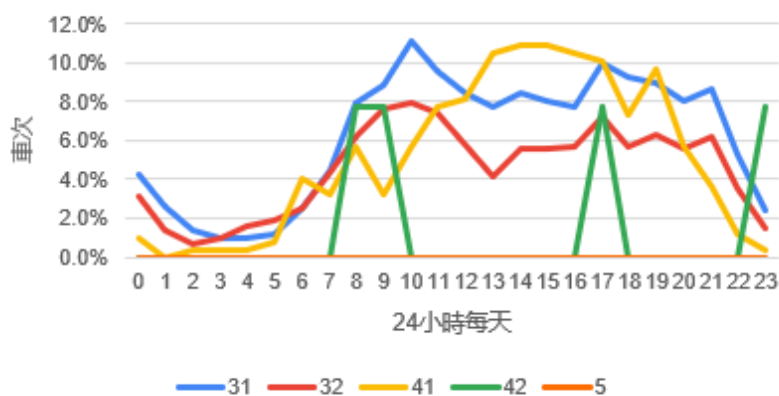
電子匝道

- 上交流道 (北上 05F0287N)

日期：2023-10-01

vehicle type (v1)								31	32	41	42	5
10/1/2023	0	512	77	3	0	0	0	4.2%	3.2%	1.0%	0.0%	0.0%
	1	208	32	0	0	0	1	2.6%	1.4%	0.0%	0.0%	0.0%
	2	108	16	1	0	0	2	1.3%	0.7%	0.4%	0.0%	0.0%
	3	78	21	1	0	0	3	1.0%	0.9%	0.4%	0.0%	0.0%
	4	76	36	1	0	0	4	0.9%	1.6%	0.4%	0.0%	0.0%
	5	94	43	2	0	0	5	1.2%	1.9%	0.8%	0.0%	0.0%
	6	204	56	10	0	0	6	2.5%	2.5%	4.0%	0.0%	0.0%
	7	356	97	8	0	0	7	4.4%	4.3%	3.2%	0.0%	0.0%
	8	635	139	14	1	0	8	7.9%	6.2%	5.7%	7.7%	0.0%
	9	708	170	8	1	0	9	8.8%	7.6%	3.2%	7.7%	0.0%
	10	892	177	14	0	0	10	11.1%	7.9%	5.7%	0.0%	0.0%
	11	770	167	19	0	0	11	9.6%	7.4%	7.7%	0.0%	0.0%
	12	674	130	20	0	0	12	8.4%	5.8%	8.1%	0.0%	0.0%
	13	623	94	26	0	0	13	7.8%	4.2%	10.5%	0.0%	0.0%
	14	675	126	27	0	0	14	8.4%	5.6%	10.9%	0.0%	0.0%
	15	643	124	27	0	0	15	8.0%	5.5%	10.9%	0.0%	0.0%
	16	621	128	26	0	0	16	7.7%	5.7%	10.5%	0.0%	0.0%
	17	803	161	25	1	0	17	10.0%	7.2%	10.1%	7.7%	0.0%
	18	743	127	18	0	0	18	9.3%	5.7%	7.3%	0.0%	0.0%
	19	715	140	24	0	0	19	8.9%	6.2%	9.7%	0.0%	0.0%
	20	648	124	14	0	0	20	8.1%	5.5%	5.7%	0.0%	0.0%
	21	695	138	9	0	0	21	8.7%	6.2%	3.6%	0.0%	0.0%
	22	426	79	3	0	0	22	5.3%	3.5%	1.2%	0.0%	0.0%
	23	195	34	1	1	0	23	2.4%	1.5%	0.4%	7.7%	0.0%
		12102	2436	301	4	0						

2023/10/01上(交流道)北上圖表



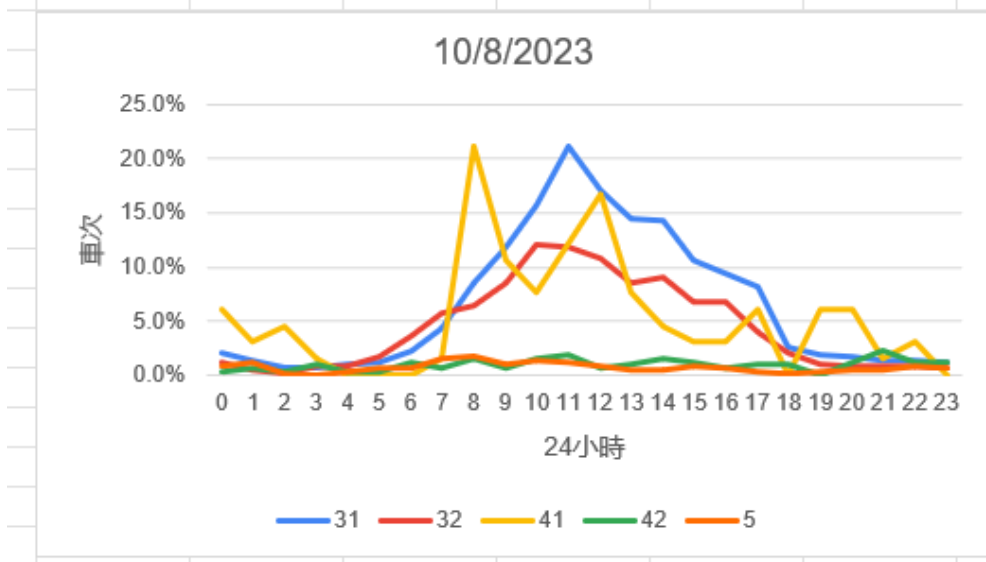
10/1 是中秋節的總決賽！這是回家鄉的最後一擊，大家紛紛踏上歸途，回到原本的城市。學生們也像風一樣迅速返回大學，整個城市瞬間瀰漫著歡樂歸鄉和返校的氛圍。這時，食欲就像激情澎湃的音樂，開始奏響美味的交響曲！各種美食在這個時刻綻放，每個人都為了品嚐各種美味而樂此不疲。中秋節的最後一天，不僅是家人團聚的日子，更是味蕾的狂歡時刻 (頭城=>坪林交控中心專用道)

● 上交流道(南下 05F0309S)

日期 : 2023-10-08

		VehicleType (VT)				
		31	32	41	42	5
10/8/2023	0	49	9	4	1	7
	1	35	4	2	2	11
	2	19	2	3	1	2
	3	19	6	1	3	0
	4	27	6	0	1	3
	5	29	13	0	1	6
	6	54	29	0	4	6
	7	105	46	1	2	13
	8	209	51	14	5	16
	9	290	68	7	2	9
	10	383	95	5	5	12
	11	517	94	8	6	11
	12	417	86	11	2	7
	13	355	67	5	3	5
	14	350	72	3	5	5
	15	258	53	2	4	7
	16	228	54	2	2	6
	17	199	32	4	3	3
	18	64	17	0	3	1
	19	47	8	4	0	3
	20	44	6	4	4	4
	21	32	7	1	7	4
	22	33	6	2	4	8
	23	28	5	0	4	6
		3791	836	83	74	155

		31	32	41	42	5
Chart Area	0	2.0%	1.1%	6.1%	0.3%	0.8%
	1	1.4%	0.5%	3.0%	0.6%	1.3%
	2	0.8%	0.3%	4.5%	0.3%	0.2%
	3	0.8%	0.8%	1.5%	1.0%	0.0%
	4	1.1%	0.8%	0.0%	0.3%	0.3%
	5	1.2%	1.6%	0.0%	0.3%	0.7%
	6	2.2%	3.7%	0.0%	1.3%	0.7%
	7	4.3%	5.8%	1.5%	0.6%	1.5%
	8	8.5%	6.4%	21.2%	1.6%	1.8%
	9	11.9%	8.6%	10.6%	0.6%	1.4%
	10	15.7%	12.0%	7.6%	1.6%	1.4%
	11	21.1%	11.9%	12.1%	1.9%	1.3%
	12	17.0%	10.9%	16.7%	0.6%	0.8%
	13	14.5%	8.5%	7.6%	1.0%	0.6%
	14	14.3%	9.1%	4.5%	1.6%	0.6%
	15	10.5%	6.7%	3.0%	1.3%	0.8%
	16	9.3%	6.8%	3.0%	0.6%	0.7%
	17	8.1%	4.0%	6.1%	1.0%	0.3%
	18	2.6%	2.1%	0.0%	1.0%	0.1%
	19	1.9%	1.0%	6.1%	0.0%	0.3%
	20	1.8%	0.8%	6.1%	1.3%	0.5%
	21	1.3%	0.9%	1.5%	2.2%	0.5%
	22	1.3%	0.8%	3.0%	1.3%	0.9%
	23	1.1%	0.6%	0.0%	1.3%	0.7%



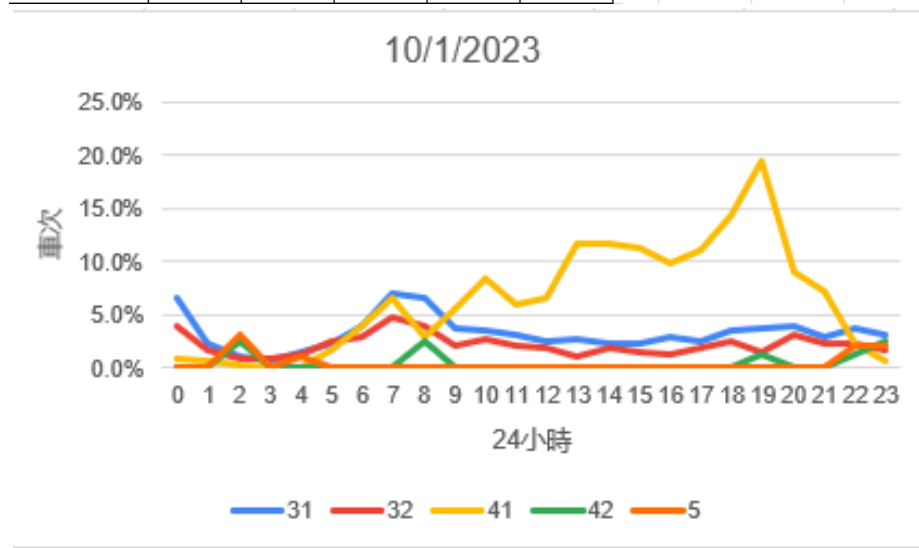
在護士國慶節放假的時候，大家都迫不及待地踏上旅遊的征途！這時候，街頭巷尾充滿了 32、41、31 的車輛，顯示出人們對於外出遊玩的熱切渴望。尤其在午夜十二點，這個時間點成為了出遊的巔峰時刻，人群湧入街頭，形成了熱鬧的景象。大家紛紛選擇在這個時刻外出，特別是因為這是吃飯的黃金時間。有些人喜歡在享受完美餐後，滿載幸福感地踏上旅程，這樣一來，不僅滿足了味蕾，還能在美好的食後時光中展開一場別開生面的冒險

(頭城=>宜蘭(四城、大福))

● 下交流道(北上 05F0309N)

日期 : 2023-10-01

		VehicleType (VT)												
		31	32	41	42	5			31	32	41	42	5	
10/1/2023	0	391	50	5	0	0			0	6.6%	4.0%	1.0%	0.0%	0.0%
	1	181	43	2	0	0			1	2.4%	1.8%	0.6%	0.0%	0.0%
	2	92	23	1	2	3			2	1.2%	0.9%	0.3%	2.5%	3.1%
	3	62	23	0	0	0			3	0.8%	0.9%	0.0%	0.0%	0.0%
	4	109	30	0	0	1			4	1.4%	1.2%	0.0%	0.0%	1.0%
	5	174	61	6	0	0			5	2.3%	2.5%	1.8%	0.0%	0.0%
	6	300	73	13	0	0			6	3.9%	3.0%	3.9%	0.0%	0.0%
	7	533	116	22	0	0			7	6.9%	4.7%	6.6%	0.0%	0.0%
	8	503	98	10	2	0			8	6.5%	4.0%	3.0%	2.5%	0.0%
	9	293	51	19	0	0			9	3.8%	2.1%	5.7%	0.0%	0.0%
	10	271	65	28	0	0			10	3.5%	2.7%	8.4%	0.0%	0.0%
	11	245	53	20	0	0			11	3.2%	2.2%	6.0%	0.0%	0.0%
	12	197	48	22	0	0			12	2.6%	2.0%	6.6%	0.0%	0.0%
	13	210	28	39	0	0			13	2.7%	1.1%	11.6%	0.0%	0.0%
	14	175	46	39	0	0			14	2.3%	1.9%	11.6%	0.0%	0.0%
	15	178	38	38	0	0			15	2.3%	1.6%	11.3%	0.0%	0.0%
	16	230	32	33	0	0			16	3.0%	1.3%	9.9%	0.0%	0.0%
	17	191	47	37	0	0			17	2.5%	1.9%	11.0%	0.0%	0.0%
	18	269	61	48	0	0			18	3.5%	2.5%	14.3%	0.0%	0.0%
	19	293	35	65	1	0			19	3.8%	1.4%	19.4%	1.3%	0.0%
	20	310	75	30	0	0			20	4.0%	3.1%	9.0%	0.0%	0.0%
	21	219	57	24	0	0			21	2.8%	2.3%	7.2%	0.0%	0.0%
	22	287	59	8	1	2			22	3.7%	2.4%	2.4%	1.3%	2.1%
	23	236	42	2	2	2			23	3.1%	1.7%	0.6%	2.5%	2.1%
		5949	1254	511	8	8								



在這個交流道，通常繁忙的人潮在平日絡繹不絕，但在國慶日這段時間，車流相對較少。這是因為在國慶日連休的四天裡，大多數打工的人都選擇在家中放鬆休息。

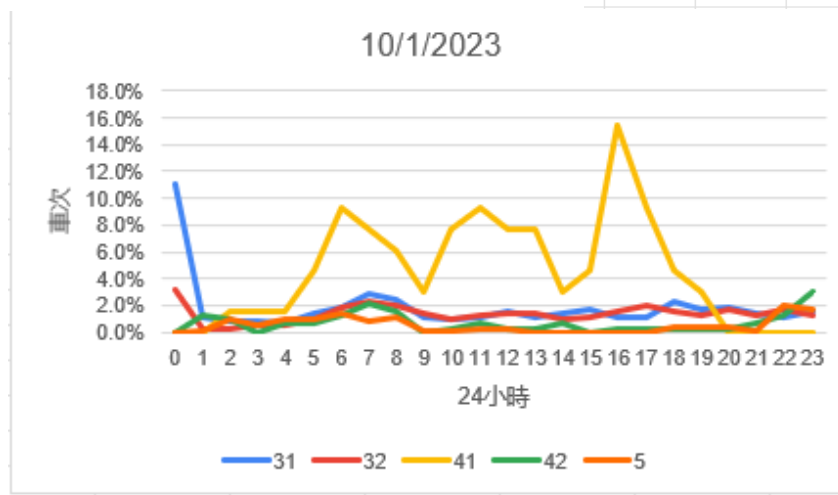
因此，這條交流道在這特殊的日子裡變得相對清幽。道路上的車輛減少，交通壓力減輕，形成了一種與平日不同的寧靜氛圍。人們更傾向於享受家庭聚會、休閒娛樂，而不是匆忙奔波在交通中。这也为城市营造了一种宁静的假日氛围。(宜蘭(四城、大福)=>頭城)

● 下交流道(南下 05F0287S)

日期：2023-09-29

	VehicleType (VT)				
	31	32	41	42	5
0	216	14	0	0	
1	62	4	0	8	2
2	44	4	2	6	18
3	44	8	2	0	10
4	40	8	2	4	18
5	76	14	6	4	18
6	94	28	12	8	26
7	148	34	10	14	14
8	126	30	8	10	20
9	58	20	4	0	2
10	48	14	10	2	2
11	58	18	12	4	4
12	80	20	10	2	4
13	56	20	10	2	0
14	72	14	4	4	0
15	90	16	6	0	0
16	60	22	20	2	0
17	60	30	12	2	0
18	120	22	6	2	6
19	88	18	4	2	8
20	96	26	0	2	6
21	72	18	0	4	2
22	60	26	0	8	36
10/1/2023 23	80	18	0	20	30
	1948	446	140	110	226

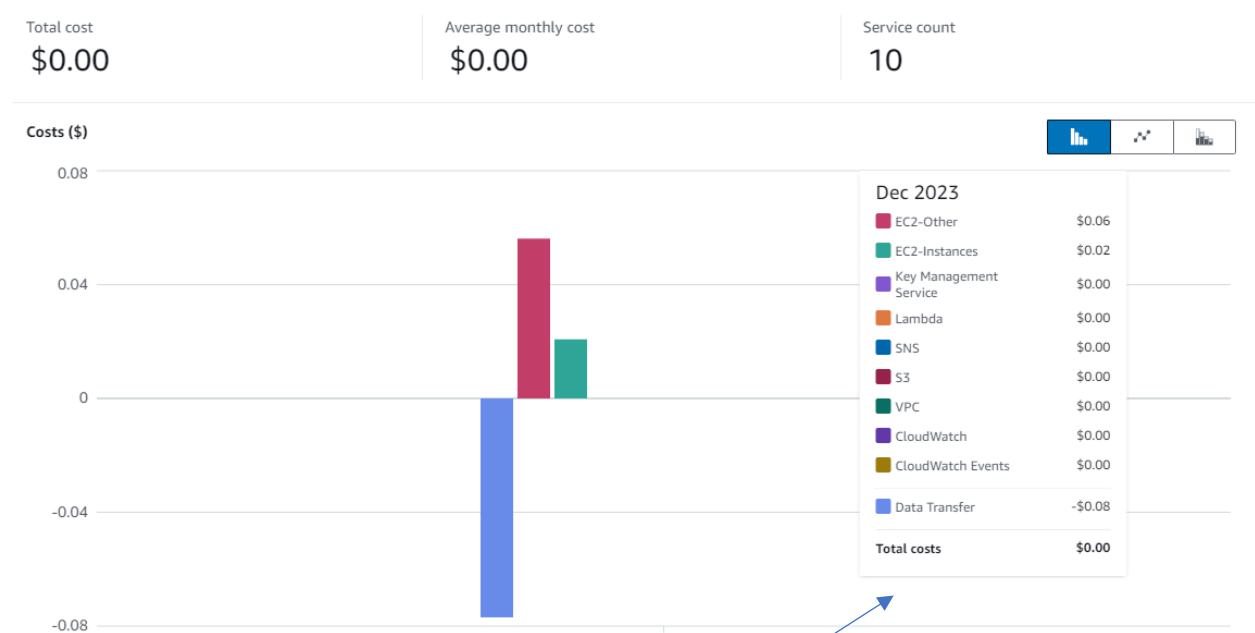
	31	32	41	42	5
0	11.1%	3.1%	0.0%	0.0%	#VALUE!
1	1.2%	0.3%	0.0%	1.2%	0.1%
2	0.8%	0.3%	1.5%	0.9%	1.0%
3	0.8%	0.5%	1.5%	0.0%	0.5%
4	0.8%	0.5%	1.5%	0.6%	1.0%
5	1.4%	1.0%	4.6%	0.6%	1.0%
6	1.8%	1.9%	9.2%	1.2%	1.4%
7	2.8%	2.3%	7.7%	2.2%	0.8%
8	2.4%	2.0%	6.2%	1.5%	1.1%
9	1.1%	1.4%	3.1%	0.0%	0.1%
10	0.9%	1.0%	7.7%	0.3%	0.1%
11	1.1%	1.2%	9.2%	0.6%	0.2%
12	1.5%	1.4%	7.7%	0.3%	0.2%
13	1.1%	1.4%	7.7%	0.3%	0.0%
14	1.4%	1.0%	3.1%	0.6%	0.0%
15	1.7%	1.1%	4.6%	0.0%	0.0%
16	1.1%	1.5%	15.4%	0.3%	0.0%
17	1.1%	2.0%	9.2%	0.3%	0.0%
18	2.3%	1.5%	4.6%	0.3%	0.3%
19	1.7%	1.2%	3.1%	0.3%	0.4%
20	1.8%	1.8%	0.0%	0.3%	0.3%
21	1.4%	1.2%	0.0%	0.6%	0.1%
22	1.1%	1.8%	0.0%	1.2%	2.0%
23	1.5%	1.2%	0.0%	3.1%	1.6%



在國慶節這個時候，一開始會有一波車流，主要是私家車，可能是人們踏上旅途或者開始回家鄉團聚。隨著私家車流的過去，接著會是大客車，這些載客車輛也紛紛上路，載送旅客到他們的目的地。

整體而言，這個時候道路上的車流較為集中，因為很多人都選擇在國慶節這個長假期間進行旅行或是回家探親。這種情況使得交通在特定時間內變得相對擁擠，人們都在努力趕路，以便在節日中享受更多的歡樂時光

AWS 花的錢



EC2-Other	\$0.06
EC2-Instances	\$0.02
Key Management Service	\$0.00
Lambda	\$0.00
SNS	\$0.00
S3	\$0.00
VPC	\$0.00
CloudWatch	\$0.00
CloudWatch Events	\$0.00
Data Transfer	-\$0.08

我麼總共花了 0.08\$.在這裡我們可以看到,我們的組差不多都花在 EC2 這個是因為我們有調整過 cloud9 的影碟,把影碟加大.

加大方式是用 EC2. 但是我們可以看到他在下面有取消我們的消費雖然這樣但他還是會把我們算到的.

因為你把資料放進去的時候然後刪掉他會覺得你沒有用到影碟它就把你的錢換回來

加分

自動下載資料- 轉到 S3

- 模組

```
import requests
import pandas as pd
import os
import tarfile
import time
import logging
import boto3
from botocore.exceptions import ClientError
from io import StringIO
```

1. requests:
 - a. 用於從 Internet 獲取檔。在此文稿中，它用於下載流量數據檔。
2. pandas:
 - a. 幫助處理數據。但是，它在此腳本中沒有執行任何操作。
3. os:
 - a. 說明與計算機通信。在這裡，它用於製作和檢查資料夾。
4. tarfile:
 - a. 處理特殊的檔包（如包裝在許多層中的禮物）。此文稿使用它來打開和解壓縮檔。
5. time:
 - a. 說明腳本在操作之間稍作休息（休眠）。它等待很短的時間。
6. logging:
 - a. 跟蹤文本正在執行的操作。這就像腳本的註釋，但在這裡用得不多。
7. boto3:
 - a. 與 Amazon 的雲（AWS）對話。它用於向 Amazon S3（一種存儲服務）發送內容和從 Amazon S3 獲取內容。
8. botocore.exceptions:
 - a. 處理與亞馬遜雲通訊時的特殊情況。它有助於處理可能發生的問題。

- 基本字串

```
Year = "2023"
Month_start = 9
Month_end = 10
Day_Start = 25
Day_End = 22
Hour_Start = 8
Hour_End = 23

TDCS_M06A = "https://tisvcloud.freeway.gov.tw/history/TDCS/M06A/"
FieldNames = ['VehicleType', 'DetectionTime_D', 'GantryID_D', 'DetectionTime_D', 'GantryID_D', 'TripLength', 'TripEnd',
               'TripInformation']

OutputDirTag = "jdwang"

# OutputDir = "/TDCS_Download_"+OutputDirTag+"/M06A/"+Year+Month
OutputDir = "/home/ec2-user/environment/Cloud9_bot03_jdwang2022_12_17/TDCS_Download_"+OutputDirTag+"/M06A/"+Year+"_M_"+str(Month_start)+"_D_"+str(Day_Start)+"-"+str(Month_end)+"_D_"+str(Day_End)+"_M_"+str(Hour_Start)+"-"+str(Hour_End)
```

1. **Year, Month_start, Month_end, Day_Start, Day_End, Hour_Start, Hour_End:**
定義要下載的數據的時間參數。
2. **TDCS_M06A:**
流量數據檔的基本 URL。
3. **FieldNames:**
CSV 資料的欄位名稱清單。
4. **OutputDirTag:**
輸出目錄名稱中使用的標記。
5. **OutPutDir:**
將存儲下載檔案的本地輸出目錄。

- 創資料庫

```
os.mkdir(OutPutDir)
if not (os.path.exists(OutPutDir)):
    os.makedirs(OutPutDir)
    print("Dir Created "+OutPutDir)
else:
    print("Dir Existed "+OutPutDir)
```

1. `if not (os.path.exists(OutPutDir)):`
如果不存在
2. `os.makedirs(OutPutDir)`
創新的資料庫
3. `print("Dir Created "+OutPutDir)`
列印 “創資料庫”*資料庫名稱*”
4. `print("Dir Existed "+OutPutDir)`
列印 “資料庫存在”*資料庫名稱*”

- 回圈

```
for mon in range(Month_start, Month_end + 1):
    if mon == Month_start:
        Day_End = 30
        Day_Start = 25
    else:
        Day_Start = 1
        Day_End = 22 # Corrected assignment from '==' to '='

    if mon < 10:
        monStr = "0" + str(mon)
        print("0" + str(mon))
    else:
        monStr = str(mon)
        print(str(mon))

    for Day in range(Day_Start, Day_End + 1):
        if Day < 10:
            DayStr = "0" + str(Day)
            print("0" + str(Day))
        else:
            DayStr = str(Day)
            print(str(Day))

    OneDate = str(Year) + monStr + DayStr

    print(OneDate)

    OneDate_OneHour_FileName = "M06A_" + OneDate + ".tar.gz"

    url = TDCS_M06A + OneDate_OneHour_FileName

    print(url)
```

1. Month 回圈:
 - 該腳本每個月都會在指定範圍內進行檢查。
 - 對於第一個月，它將開始和結束日期設置為特定值。對於隨後的幾個月，它將開始日期重置為 1，將結束日期重置為 22。
2. Month 格式:
 - 如果小於 10，它會在月份添加一個前導零。
 - 列印格式化的月份。
3. Day 回圈:
 - 對於當月指定範圍內的每一天：
 - 如果它小於 10，它會在一天中添加一個前導零。
 - 列印格式化的日期。
4. 日期組合:
 - 將年份、格式化的月份和格式化的日期組合在一起，以創建“YYYYMMDD”格式的日期。
 - 列印生成的日期。
5. 檔名生成:
 - 使用生成的日期為 tar.gz 檔案創建檔名。
 - 列印生成的檔名。
6. URL 建設:
 - 通過組合基本 URL 和生成的檔名來構造 URL。

- 下載 tar.gz 資料

```
# Download the tar.gz file
response = requests.get(url)
tar_file_path = os.path.join(OutPutDir, OneDate_OneHour_FileName)
with open(tar_file_path, 'wb') as tar_file:
    tar_file.write(response.content)
```

1. `response = requests.get(url)`
此行向指定的 URL（url）發送請求以從 Internet 獲取某些內容。
2. `tar_file_path = os.path.join(OutPutDir, OneDate_OneHour_FileName)`
此行通過組合輸出目錄（OutPutDir）和 tar.gz 檔案（OneDate_OneHour_FileName）的檔名來創建完整的檔案路徑。
3. `with open(tar_file_path, 'wb') as tar_file:`
此行在指定的檔案路徑（tar_file_path）處以二進位寫入模式（'wb'）打開檔。
4. `tar_file.write(response.content)`
此行將從 Internet 請求（response.content）接收的內容寫入打開的檔。

- 解壓縮 tar.gz

```
# Create output directory if it doesn't exist
os.makedirs(output_directory, exist_ok=True)

with tarfile.open(file_path, 'r:gz') as tar:
    tar.extractall(output_directory)

time.sleep(10)
```

1. `os.makedirs(output_directory, exist_ok=True)`
此行檢查指定的輸出目錄（output_directory）是否存在。
2. `with tarfile.open(file_path, 'r:gz') as tar:`
此行以唯讀模式（'r'）和 gzip 壓縮（'gz'）打開位於指定 file_path 處的 tar.gz 檔
3. `tar.extractall(output_directory)`
此行提取打開的 tar.gz 檔的所有內容，並將它們放入指定的 output_directory.
4. `time.sleep(10)`
此行使用 time.sleep 函數暫停腳本的執行 10 秒鐘。

- 上傳到 S3

```
def upload_files_to_s3(local_path, s3_bucket, s3_prefix=''):
    # Create an S3 client
    s3 = boto3.client('s3')

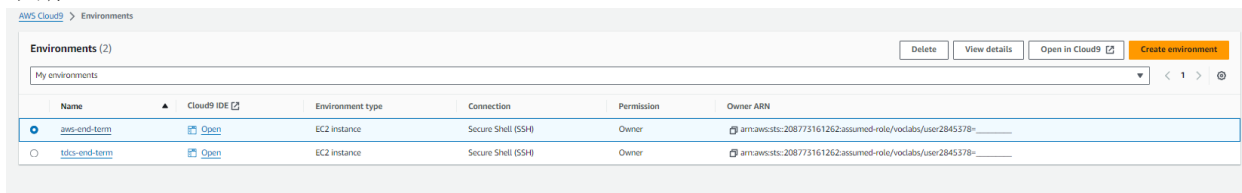
    # Walk through the local directory structure
    for root, dirs, files in os.walk(local_path):
        for file in files:
            if file.endswith('.csv'): # Check if the file is a CSV file
                local_file_path = os.path.join(root, file)
                s3_key = os.path.join(s3_prefix, os.path.relpath(local_file_path, local_path)).replace('\\', '/')

                # Upload the file to S3
                try:
                    s3.upload_file(local_file_path, s3_bucket, s3_key)
                    print(f"Uploaded {local_file_path} to S3://{s3_bucket}/{s3_key}")
                except Exception as e:
                    print(f"Error uploading {local_file_path}: {e}")
                    break
```

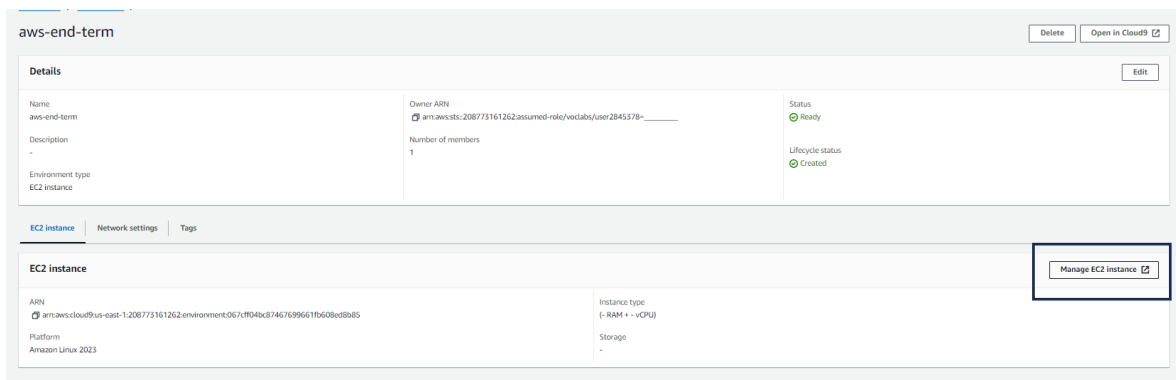
1. `s3 = boto3.client('s3')`
 - This function takes three parameters: `local_path` (the local directory containing files), `s3_bucket` (the name of the Amazon S3 bucket), and `s3_prefix` (an optional prefix to be added to S3 object keys).
2. `for root, dirs, files in os.walk(local_path):`
 - 它啟動一個循環來遍曆本地目錄結構（`local_path`），包括子目錄。
3. `for file in files:`
 - `if file.endswith('.csv'):` # Check if the file is a CSV file
 - `local_file_path = os.path.join(root, file)`
 - 在迴圈中，它會檢查當前目錄中的每個檔，以查看它是否以“.csv”結尾。如果檔是 CSV 檔，則會創建本地檔案（`local_file_path`）的完整路徑。
4. `s3_key = os.path.join(s3_prefix, os.path.relpath(local_file_path, local_path)).replace('\\', '/')`
 - 它通過組合指定的 `s3_prefix` 和根目錄中本地檔的相對路徑來創建 S3 金鑰
 - `replace` 方法用於將反斜杠替換為正斜杠，確保與 S3 金鑰格式相容。
5. `try:`
 - 它嘗試使用計算出的 S3 金鑰將本地檔案上傳到指定的 S3 儲存桶。
 - 如果成功，它將列印一條消息，指示上傳成功。
 - 如果在上傳過程中出現錯誤，它會捕獲異常，列印錯誤消息，並中斷迴圈。

改變 cloud9 影碟大小

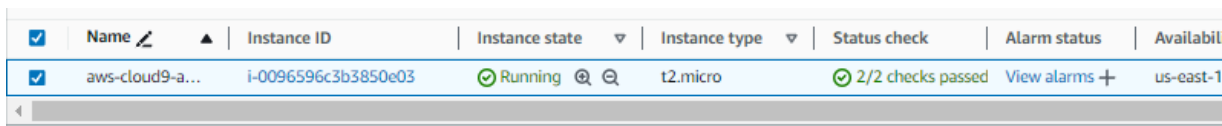
- 手動



- 點進去你的 cloud9 名字



- 點 manage 2 instance

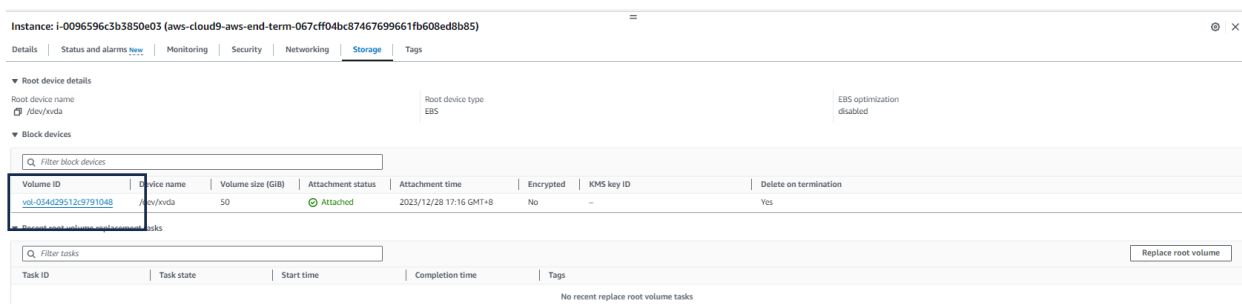


- 打鉤你的 instances 名稱

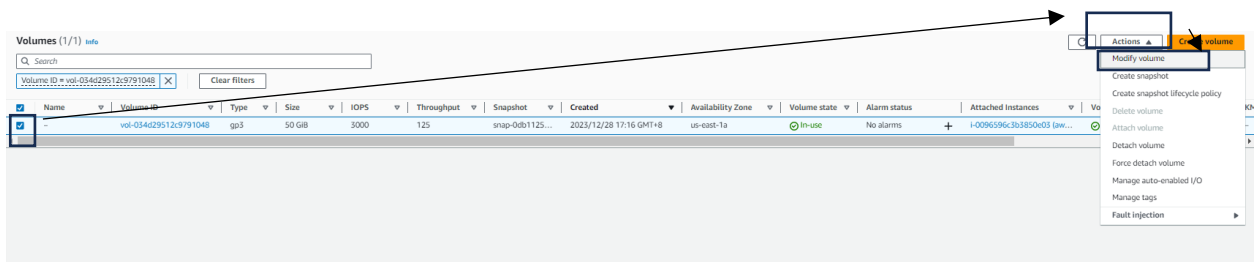
Instance: i-0096596c3b3850e03 (aws-cloud9-aws-end-term-067c7ff04bc87467699661fb608ed8b85)

[Details](#) | [Status and alarms](#) | [Monitoring](#) | [Security](#) | [Networking](#) | [Storage](#) | [Tags](#)

- 在下面的表格點 storage



- 點你的 volume id



- 打勾你的 volume id 然後選擇 actions 選擇 modify volume

EC2 > Volumes > vol-034d29512c9791048 > Modify volume

Modify volume [Info](#)

Modify the type, size, and performance of an EBS volume.

Volume details

Volume ID
 vol-034d29512c9791048

Volume type [Info](#)
General Purpose SSD (gp3) ▼

Size (GiB) [Info](#)

Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS [Info](#)

Min: 3000 IOPS, Max: 16000 IOPS. The value must be an integer.

Throughput (MiB/s) [Info](#)

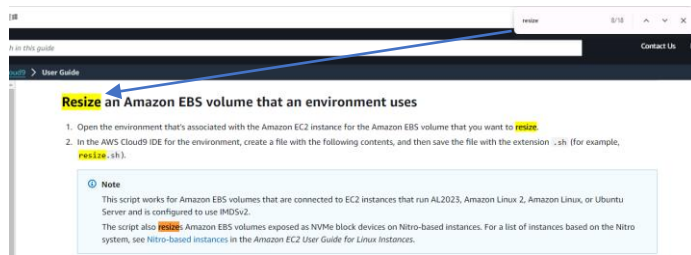
Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

[Cancel](#) [Modify](#)

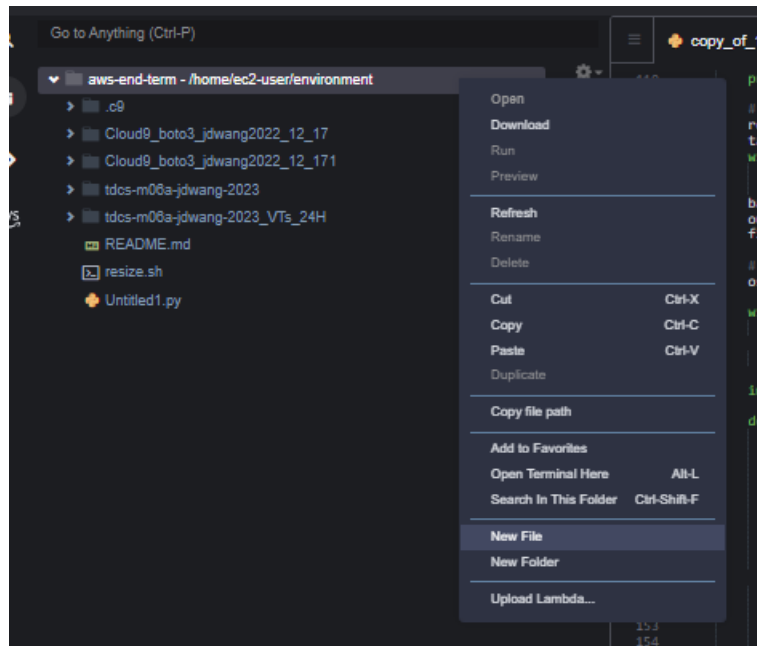
- 在這裡更改你的 size 然後按 modify

- 用程式碼

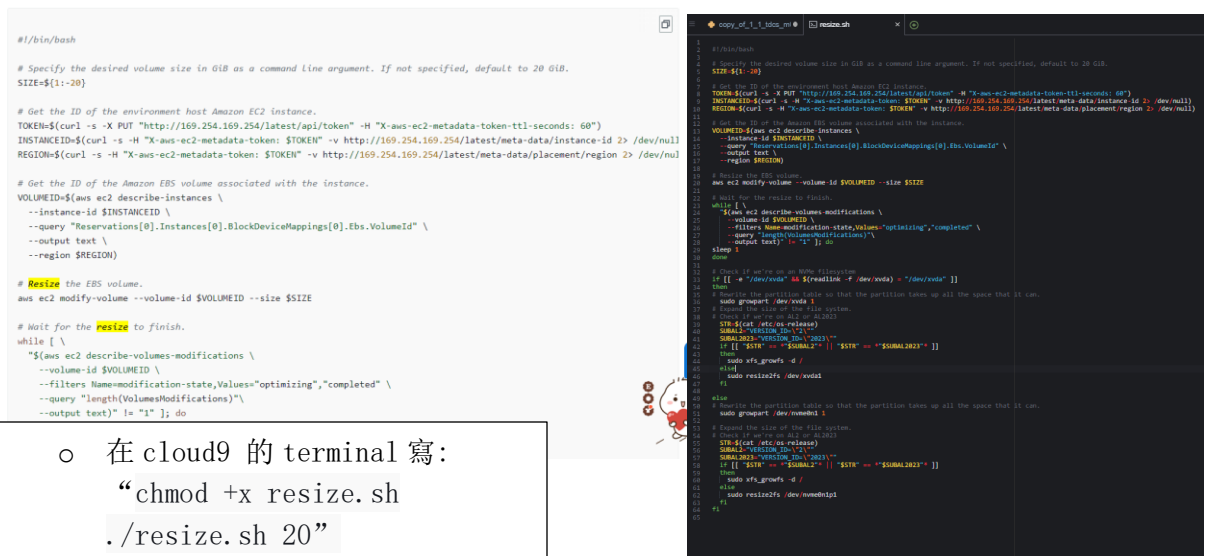
- 去 <https://docs.aws.amazon.com/cloud9/latest/user-guide/move-environment.html>
- Ctrl+F 然後 search resize



- 在 cloud9 創心得資料,名稱”resize.sh”



- 從網頁負責程式碼然後貼在剛剛創完的資料



- 在 cloud9 的 terminal 寫:
“chmod +x resize.sh
./resize.sh 20”
20 可以更改你想要的大小

3. 檢查 s3 資料有多少

```
import boto3

def get_total_objects_in_bucket(bucket_name):
    # Create an S3 client
    s3 = boto3.client('s3')

    # Create a paginator
    paginator = s3.get_paginator('list_objects_v2')

    # Initialize the total count
    total_objects = 0

    for page in paginator.paginate(Bucket=bucket_name):
        # Update the total count
        total_objects += page.get('KeyCount', 0)

    print(f"Total objects in bucket '{bucket_name}': {total_objects}")

# Example usage:
bucket_name = "tdcs-m06a-jdwang-2023"
get_total_objects_in_bucket(bucket_name)
```

它使用一種特殊的工具（分頁器）來流覽所有檔，並隨時將它們的編號相加。完成後，它會告訴您該存儲區域中有多少個檔。這就像計算特定在線存儲盒中的所有物品一樣。

4. 下載到 cloud9 檢查有多少資料

```
import os

DIR_NAME = 'tdcs-m06a-jdwang-2023'
OutputDir = './' + BUCKET_NAME + '/'

# ... (rest of your script)

# Count the total number of files in the directory
file_count = len([f for f in os.listdir(OutputDir) if os.path.isfile(os.path.join(OutputDir, f))])

print(f"Total number of files in directory {OutputDir}: {file_count}")
```

它建立一個目錄名稱，嘗試創建該目錄，檢查它是否已經存在，然後計算目錄中的文件總數。最初，它將目錄名稱設置為“tdcs-m06a-jdwang-2023”。在此之後，腳本將繼續執行其他操作（代碼片段中未詳細說明）。

5. 直接從 cloud9 的資料做出 csv

```
# # Jing-Doo Wang, jdwang@asia.edu.tw, 2022/12/14
#
from os import listdir
from os.path import isfile, isdir, join
import os
import pandas as pd
import boto3
import botocore
from pathlib import Path

# Specify the local directory path
LocalDirectory = '/home/ec2-user/environment/tdcs-m06a-jdwang-2023'
#Tag_TimeInterval = '2022-12-1 7'
Tag_TimeInterval = '2023-09-1 2023-09-30'
#BUCKET_NAME = 'tdcs-m06a-upload-from-cloud9-jdwang-2022'
OutputDir = f'./{os.path.basename(LocalDirectory)}_VTs_24H/' # Use the base name of the local directory
Pathname = OutputDir
Target_Field = 'GantryID_O'
Target_Value = '05F0309S'

if os.path.exists(OutputDir):
    print(f'{OutputDir} Existed!')
else:
    os.mkdir(OutputDir)
    print(f'Directory {OutputDir} created')

FieldNames = ['VehicleType', 'DetectionTime_O', 'GantryID_O', 'DetectionTime_D', 'GantryID_D', 'Triplength', 'TripEnd', 'TripInformation']
FieldNames_Selected = ['VehicleType', 'DetectionTime_O', 'GantryID_O', 'DetectionTime_D', 'GantryID_D', 'Triplength', 'TripEnd']

# Iterate through files in the local directory
for file_name in os.listdir(LocalDirectory):
    if file_name.endswith('.csv'):
        print(file_name)
        FILE_NAME = file_name
        OutputFileLocation = os.path.join(OutputDir, FILE_NAME)

        # Read the CSV file locally
        data = pd.read_csv(os.path.join(LocalDirectory, FILE_NAME), encoding='utf-8', header=0, names=FieldNames, delimiter=',', low_memory=False)

        df = pd.DataFrame(data[FieldNames_Selected])

        df['DetectionTime_O'] = df['DetectionTime_O'].astype('datetime64[ns]')
        df['Year'] = df.DetectionTime_O.dt.year
        df['Month'] = df.DetectionTime_O.dt.month
        df['Day'] = df.DetectionTime_O.dt.day
        df['Weekday'] = df.DetectionTime_O.dt.day_name()
        df['Hour_0'] = df.DetectionTime_O.dt.hour

        df_One_GantryID_O = df[(df[Target_Field] == Target_Value)]

        OneDayVT24HRecords = df_One_GantryID_O.groupby(['Year', 'Month', 'Day', 'Weekday', 'Hour_0', 'VehicleType']).size().reset_index(name='counts')

        FileName = f'VTs_24H_{Tag_TimeInterval}_{Target_Field}_{Target_Value}.csv'
        filepath = Path(Pathname + FileName)
        filepath.parent.mkdir(parents=True, exist_ok=True)

        if os.path.exists(filepath):
            OneDayVT24HRecords.to_csv(filepath, mode='a', index=False, header=False)
        else:
            OneDayVT24HRecords.to_csv(filepath, mode='w', index=False, header=True)

        # You may want to delete the input file after processing
        # os.remove(os.path.join(LocalDirectory, FILE_NAME))
        print(f"The file removed =>, OutputFileLocation")
```

tdcs-m06a-jdwang-2023

TDCS_M06A_20230925_000000.csv

TDCS_M06A_20230925_010000.csv

TDCS_M06A_20230925_020000.csv

TDCS_M06A_20230925_030000.csv

TDCS_M06A_20230925_040000.csv

TDCS_M06A_20230925_050000.csv

TDCS_M06A_20230925_060000.csv

TDCS_M06A_20230925_070000.csv

TDCS_M06A_20231022_160000.csv

TDCS_M06A_20231022_170000.csv

TDCS_M06A_20231022_180000.csv

TDCS_M06A_20231022_190000.csv

TDCS_M06A_20231022_200000.csv

TDCS_M06A_20231022_210000.csv

TDCS_M06A_20231022_220000.csv

TDCS_M06A_20231022_230000.csv

結果

```
tdcs-m06a-jdwang-2023_VTs_24H
Counts_2023-09-1_2023-09-30_GantryID_O_05F0287N.csv
Counts_2023-09-1_2023-09-30_GantryID_O_05F0287S.csv
Counts_2023-09-1_2023-09-30_GantryID_O_05F0309N.csv
Counts_2023-09-1_2023-09-30_GantryID_O_05F0309S.csv
```

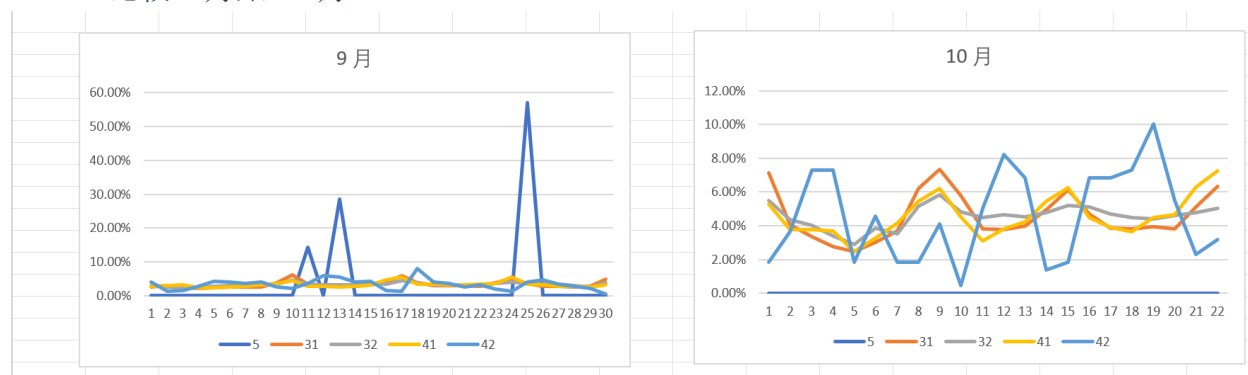
因為有調整 cloud9 大小所以我們可以存一個月的資料在 cloud9 裡面

6. 用 excel 整理資料

Year	(All)					
Month	9					
Sum of count	Column Labels					
Row Labels		5	31	32	41	42
Grand Total						
1		0	5930	1773	204	13
Friday		0	5930	1773	204	13
2		0	6824	1612	206	4
Saturday		0	6824	1612	206	4
3		0	5989	1369	229	5
Sunday		0	5989	1369	229	5
4		0	5004	1416	167	9
Monday		0	5004	1416	167	9
5		0	5356	1692	173	14
Tuesday		0	5356	1692	173	14
6		0	5768	1843	186	13
Wednesday		0	5768	1843	186	13
7		0	5893	1889	197	12
Thursday		0	5893	1889	197	12
8		0	5918	1889	220	13
Friday		0	5918	1889	220	13
9		0	8409	2026	259	8
Saturday		0	8409	2026	259	8
10		0	13440	2769	313	7
Sunday		0	13440	2769	313	7
11		1	7428	2073	198	12
Monday		1	7428	2073	198	12
12		0	6138	1937	204	19
Tuesday		0	6138	1937	204	19
13		2	6042	1910	188	18
Wednesday		2	6042	1910	188	18
14		0	6138	1961	194	13
Thursday		0	6138	1961	194	13
15		0	7027	2233	229	14
Friday		0	7027	2233	229	14
16		0	8326	2060	331	5
Saturday		0	8326	2060	331	5
17		0	13098	2672	391	4
Sunday		0	13098	2672	391	4
18		0	8558	2317	248	26
Monday		0	8558	2317	248	26
19		0	6481	1986	243	13
Tuesday		0	6481	1986	243	13
20		0	6524	1967	227	12
Wednesday		0	6524	1967	227	12
21		0	6615	1943	224	8
Thursday		0	6615	1943	224	8
22		0	6373	1924	239	10
Friday		0	6373	1924	239	10
23		0	8628	2206	257	6
Saturday		0	8628	2206	257	6
24		0	11244	2399	389	4
Sunday		0	11244	2399	389	4
25		4	8018	2242	247	13
Monday		4	8018	2242	247	13
26		0	6455	2056	228	15
Tuesday		0	6455	2056	228	15
27		0	6423	2085	201	11
Wednesday		0	6423	2085	201	11
28		0	5981	1833	204	9
Thursday		0	5981	1833	204	9
29		0	6190	1532	190	7
Friday		0	6190	1532	190	7
30		0	11044	2294	225	2
Saturday		0	11044	2294	225	2
Grand Total		7	221262	59908	7011	319

我們在這裡用 pivot table 來整理我們的資料讓我們方便的處理資料,這個 pivot table 是 excel 其中一個 feture,這個讓我們很方便的排序我們的大數據資料

7. 比較 9 月跟 10 月



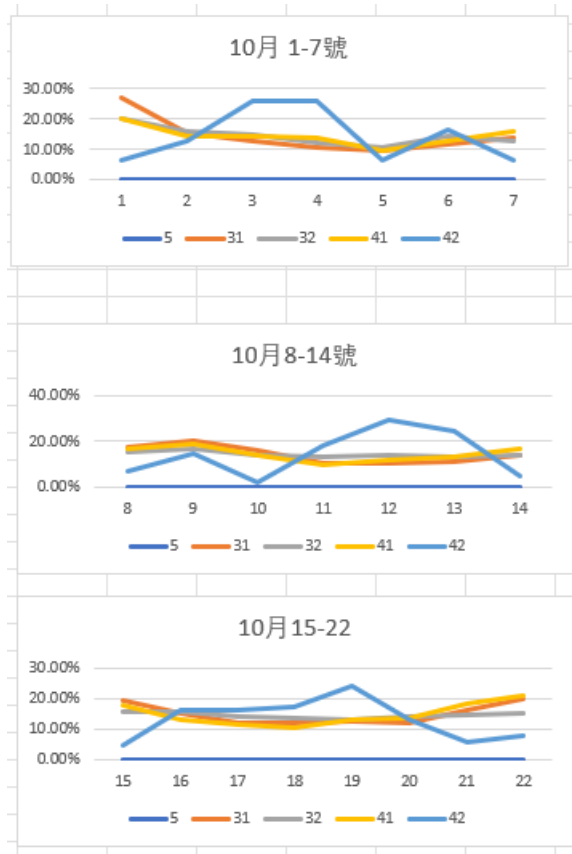
9 月的 30、31、40、41 號的車車似乎比較少見。這是因為在這段時間，人們通常放假，他們可能選擇回家鄉、待在家裡，或者出去旅遊幾天。所以，這些日期的連接車相對較空，大家都在享受屬於自己的假期時光。

可以想像，在這個時候，車車度過的時間也變得相對寧靜，彷彿在等待著人們歸來的熱鬧。或者，它們成為了遠足的夥伴，帶著人們去探索新奇的地方。總之，在這段假期裡，連接車也有著不同的節奏，等待著人們歡樂歸來。

10 月的時候情況就有所改變了，因為大家開始迎來新學期，開始了上學的生活。這個時候，街上的熱鬧氛圍會再次升華，但焦點轉移到了大家匆忙回到各自的大學城市。

學生們可能會急著趕回學校，開始新學期的課業生活。連接車在這個時候可能會再次變得繁忙，載送著學子們回到他們的大學校園。這也是一個充滿期待和活力的時刻，大家帶著對新學期的憧憬，匆忙而又興奮地返回學校。連接車在這個轉折點上扮演著重要的角色，將學子們安全送回他們的學術天地。

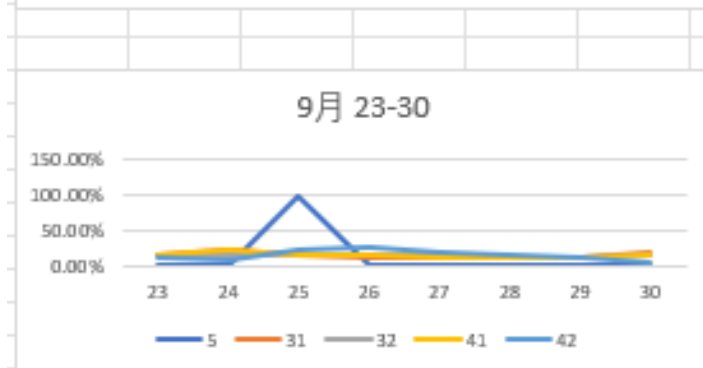
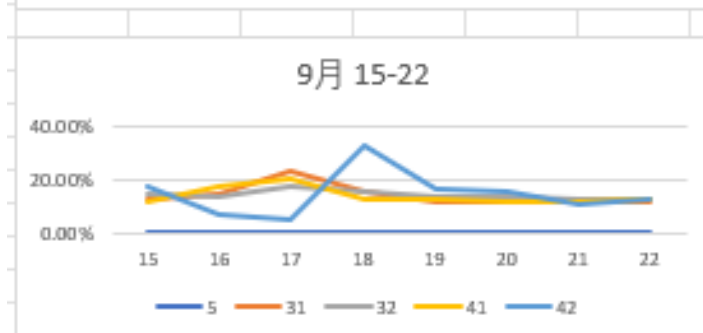
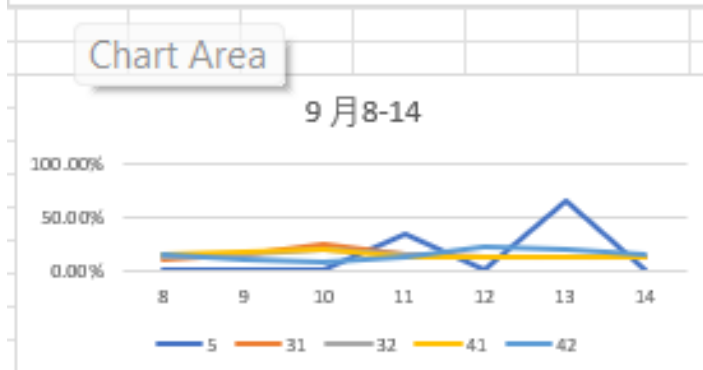
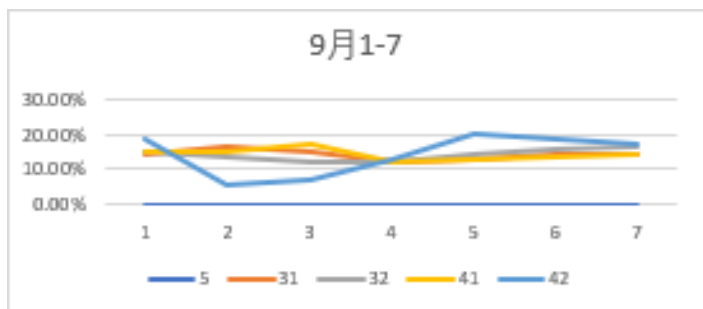
8. 比較每個星期一個星期



在這條路上，最吸睛的就是那些藍色的(五)聯結車。

例如是 2-6 好是平日, 平日就是週一到週五, 這時候整條街道都充滿了辛勤工作的人們, 他們匆匆忙忙地走過。而週五來臨時, 藍色車輛也跟著悠閒了起來, 不再那麼趕忙。街道上的氛圍變得輕鬆愉快, 彷彿在歡迎即將到來的週末。大家都開始悠閒地準備, 迎接一個輕鬆愉快的周末時光。這時的藍色車輛就像是參與了一場城市節奏的舞蹈, 輕盈地轉動在街頭。

。

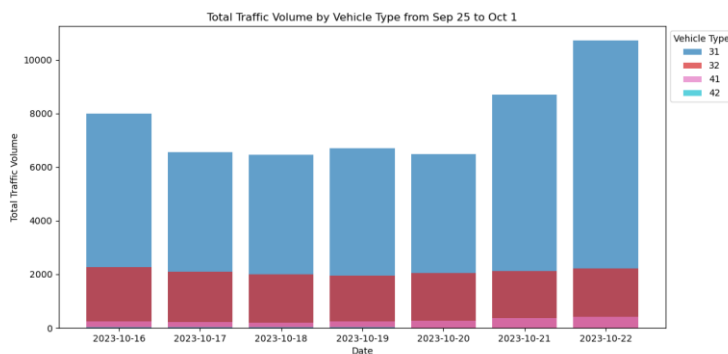
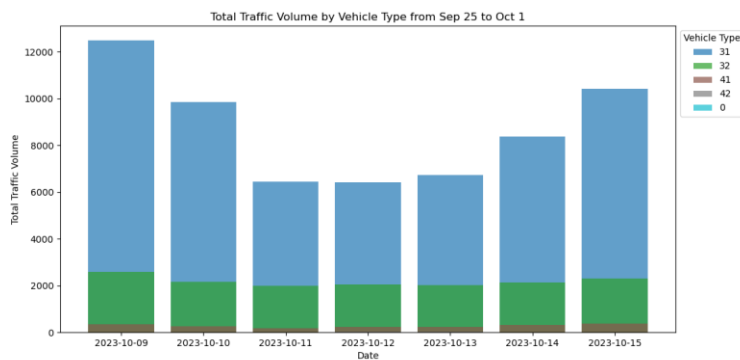
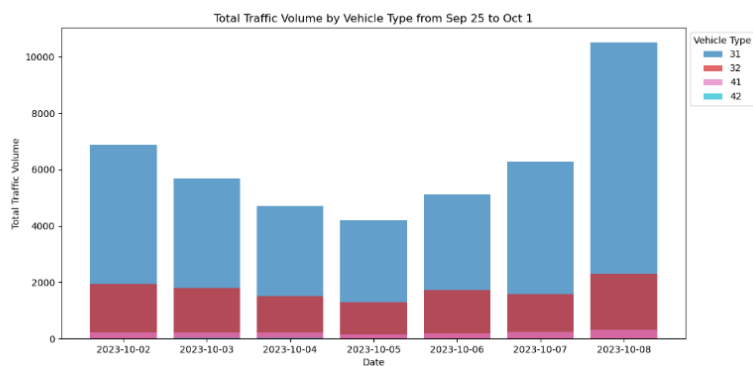
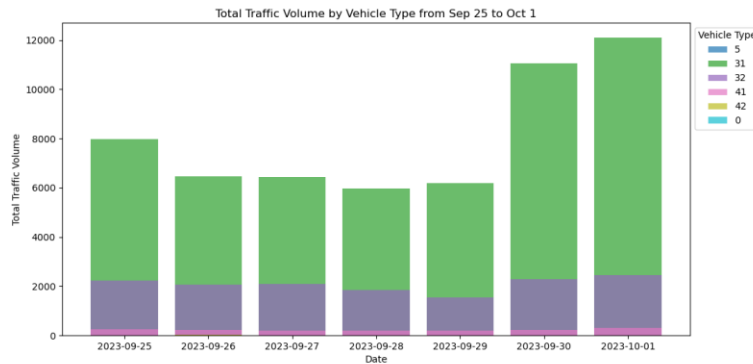


9月是連接車最繁忙的時候，尤其是在第一個星期。這是因為大家都要迎接新學期，開學的季節總是讓整個城市充滿了生氣。

在這個時候，大家都開始準備購買學用品、寄送東西到學校，因此網路訂單像是一場爆炸，瞬間湧入。大家紛紛下單，讓連接車成為一個熱鬧非凡的地方。

可以想像，在這個季節，連接車就像是一個擁擠的市集，人們匆忙地購物、寄送物品，為新學期做好準備。這種熱鬧的氛圍就像是一場快樂的開學狂歡派對，每個人都在為即將展開的學業歡呼。

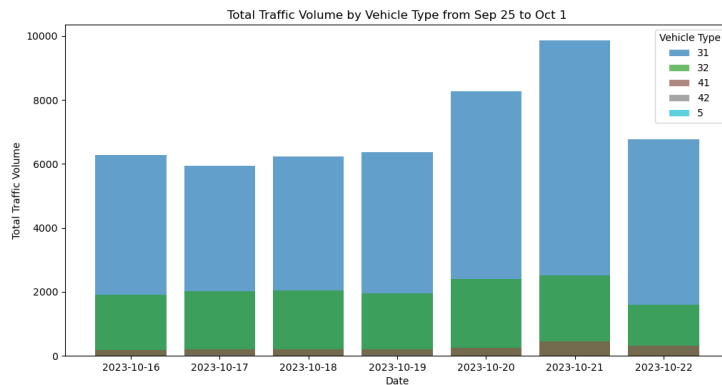
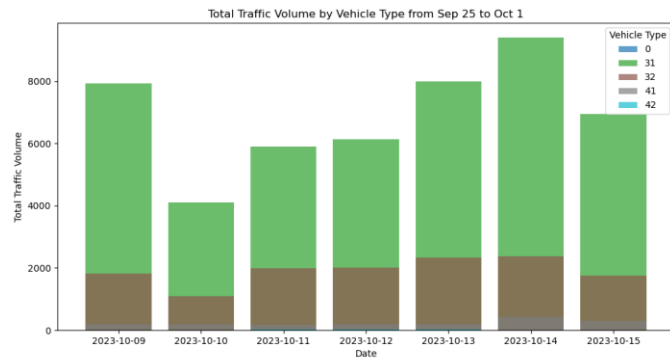
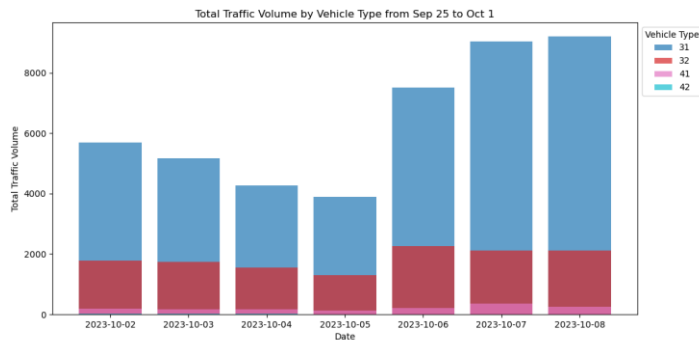
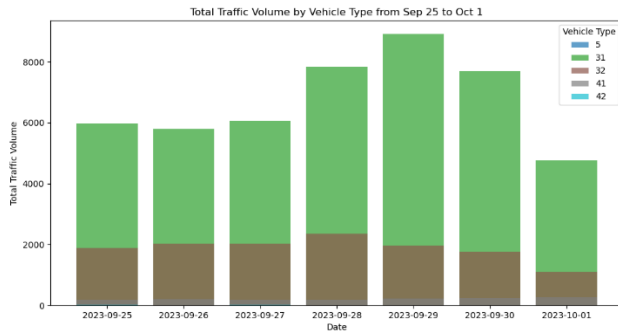
05F0287N 頭城-坪林行控專用道



這是從宜蘭往台北最後一個收費站我們發現

1. 連假的時候車流量比平日多
2. 9月25道10月1日除了連假的關係所以車流較多外還可以發現在星期天的時候往台北的車輛最多，可以推測是連假最後一天往所以大家返家造成的
3. 國慶連假時並不是像中秋連假最後一天才是最多的車流量，推測為大家提前返家收心和避免最後一天因為塞車而晚歸進而影響到隔天的上班課
4. 最後兩周的星期天依然有許多遊客返家
5. 增加的车種主要是小客車
6. 沒有看到5號聯的原因是石碇交流道至頭城交流道路段禁只行駛聯結車，不過我們還是有看到有聯結車的出現
7. 發現0號車種

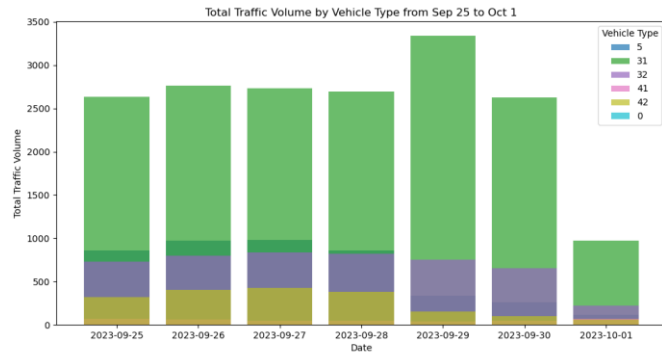
05F0287S 坪林行控專用道-頭城



這是台北往宜蘭第一個收費站我們發現

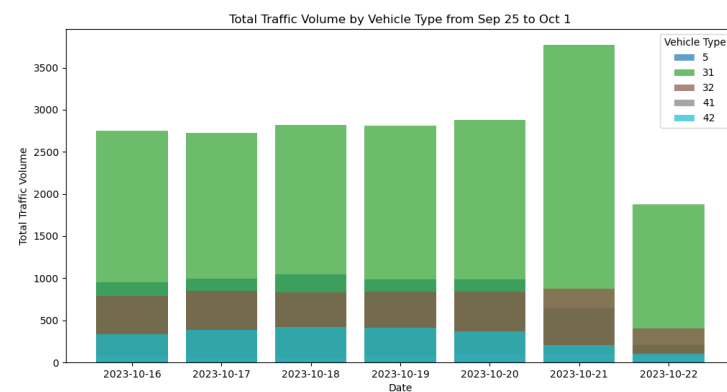
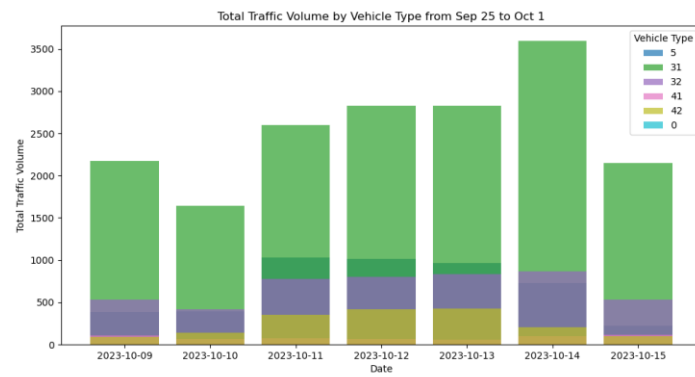
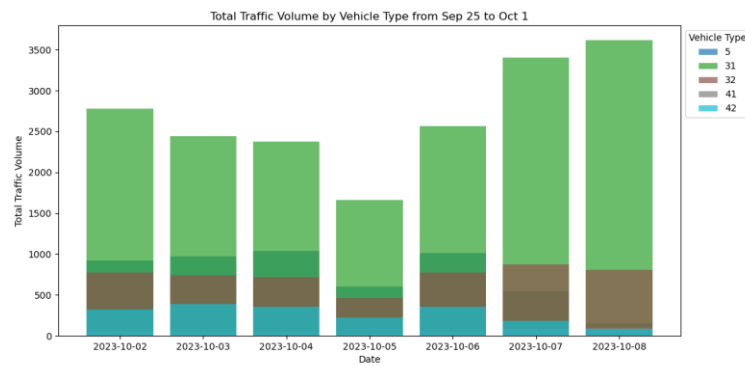
1. 連假的時候車流量比平日多
2. 假期第一天的車流量都是最多的
假期的最後一天都是最少的
所以增加的應該都是遊客
3. 沒有看到 5 號聯的原因是石碇交流道
至頭城交流道路段禁只行駛聯結車，
不過我們還是有看到有聯結車的出現
4. 發現 0 號車種

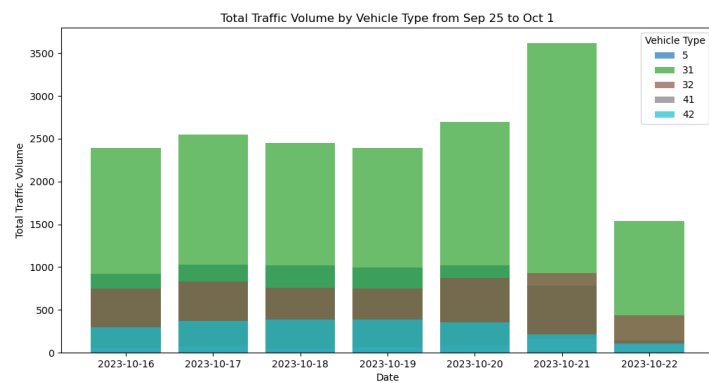
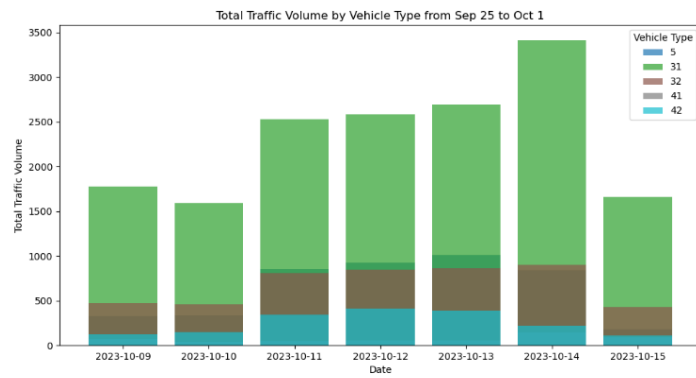
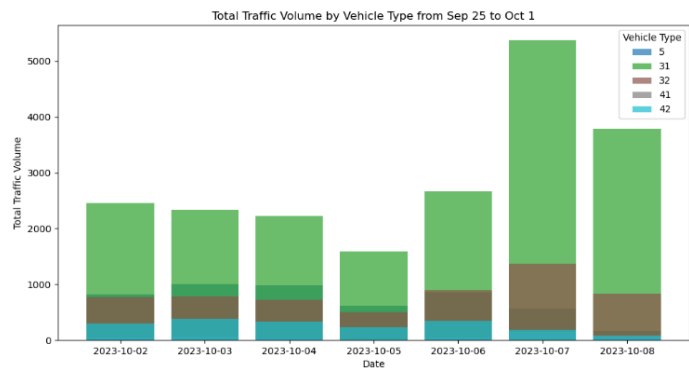
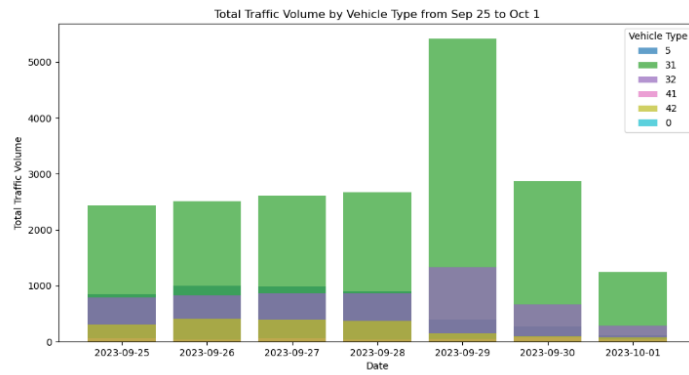
05F0309N (宜蘭(北)-頭城)



我們發現

1. 發現 0 號車種
2. 假日車流不一定最多
3. 車流跟 05F0287S 和 05F0287N 比起來少很多

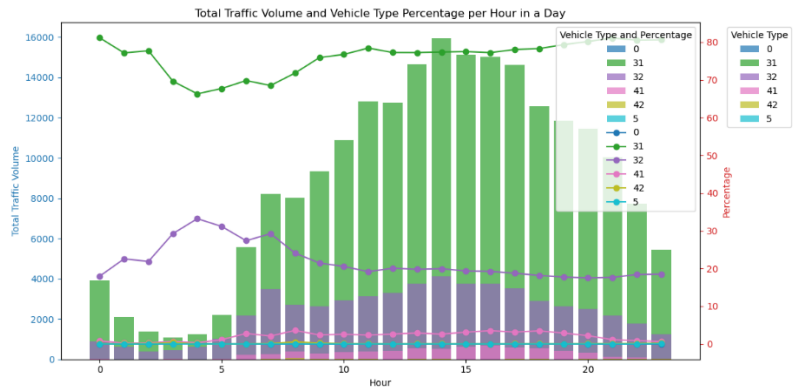




我們發現

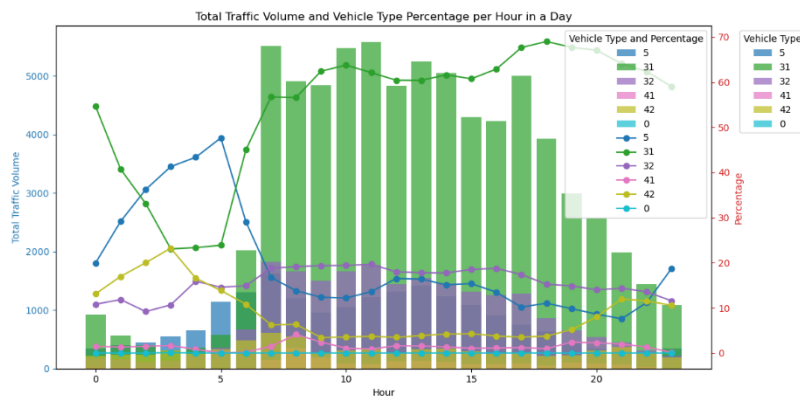
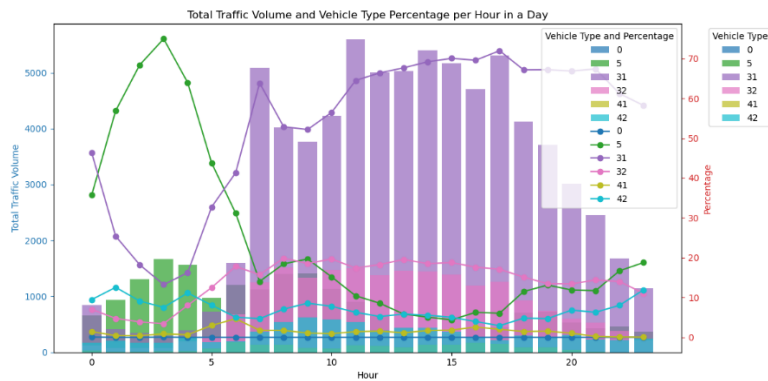
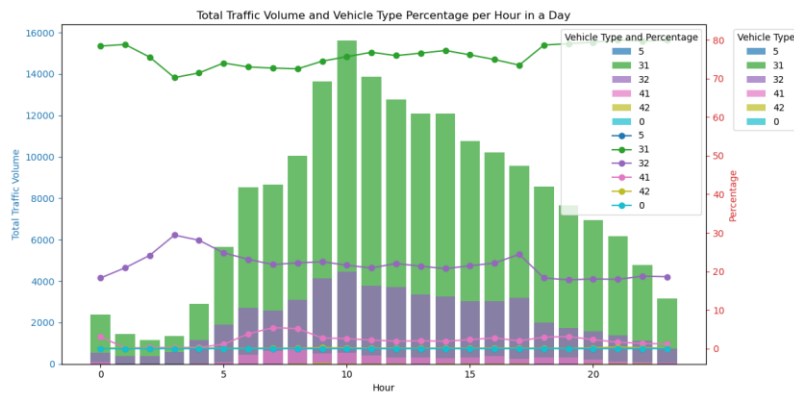
1. 發現 0 號車種
2. 假日車流不一定最多
3. 車流跟 05F0287S 和 05F0287N 比起來少很多

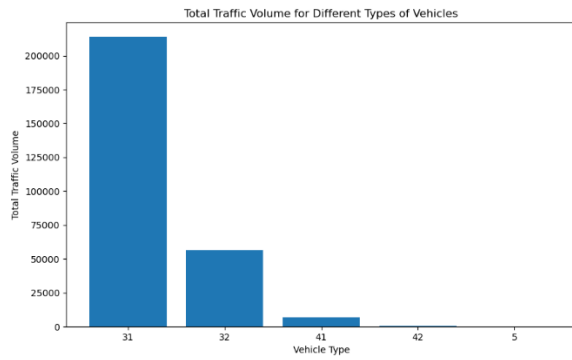
這是每天 24 小時加起來的數量



我們發現

1. 在 0 到 5 的時候車流量都很少要畢滿塞車可以在這時候出門





由上到下分別是

05F0287N

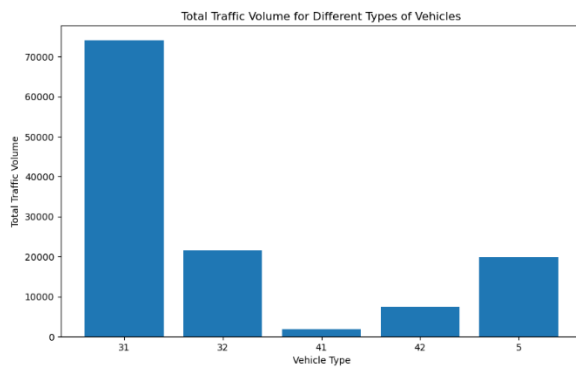
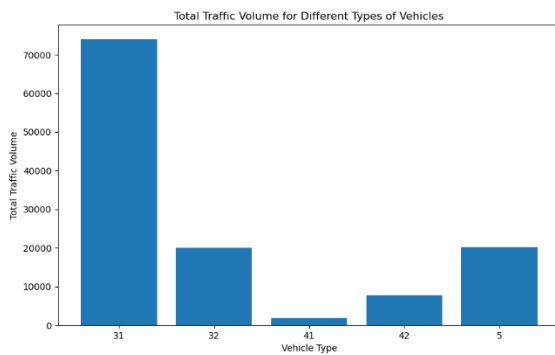
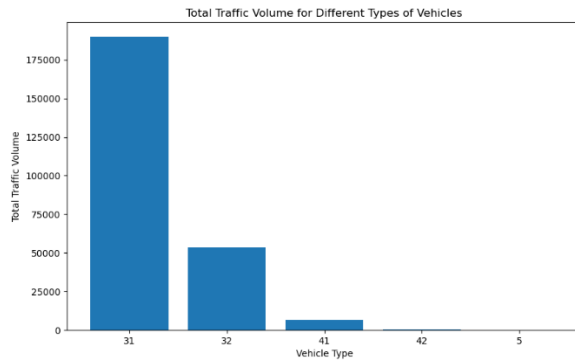
05F0287S

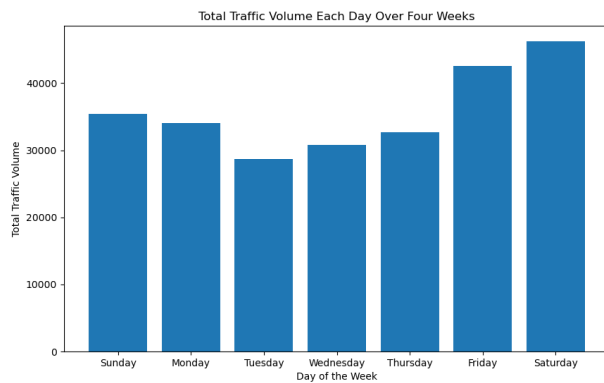
05F0309N

05F0309S

這是車種的總量我們發現

1. 05F0287N 和 05F0287S 車種 42 和 5 沒有太多
2. 05F0309N 和 05F0309S 每個車種都有





這是一星期中每天總量

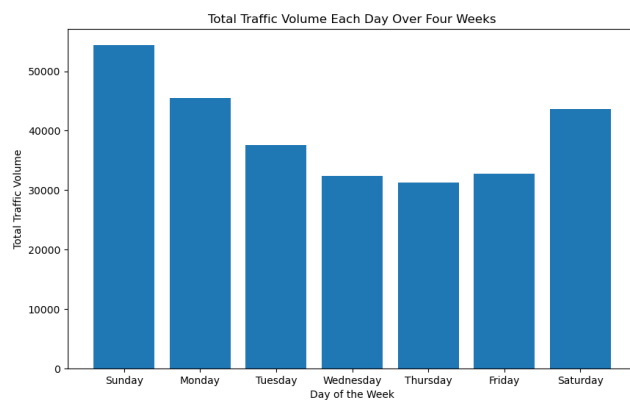
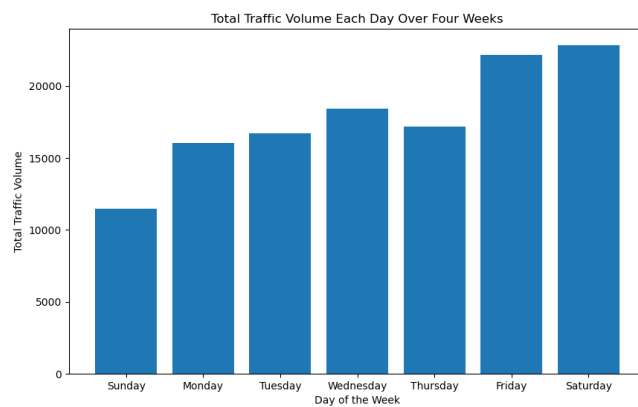
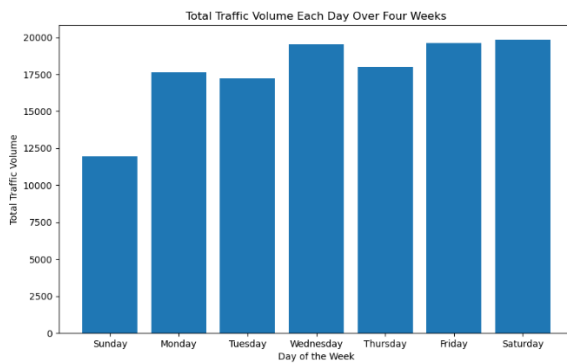
由上到下分別是

05F0287N

05F0287S

05F0309N

05F0309S



會議討論



地點：教室 1627

時間：08/12/2023

分配工作：

這個時候我們就決定, 要更改哪個部分, 而這個時候老師剛教學到 `colab`, 所以我們先決定怎麼把 `tar.gz` 檔案下載出來



地點：教室 1627

時間：22/12/2023

分配工作：

我們決定很多地方能更改, 列如下在爬蟲, 還有我們還能找辦法怎麼讓 `cloud9` 能放一個月的資料

小組分工+均衡

- Cloud 9 : 謝欣恩
- Word : 劉妍群+陳奕辰
- PPT : 劉妍群

學習心得

- 謝欣恩

第一次學習 AWS 真的讓我覺得很困難,話題多時間而錢了,別說換很多時間還很順利,中間還有很多問題要處理,而且我從來不是一個會解釋或者講話的人,我才學諮詢工程係這個係.

但到最後我真的發下安學習過程就是這樣,你要先知道手動花很多時間,你才會知道這個自動程式碼,是怎麼做出來而為什麼要做出來,cloud9+python 是一個很完美的組合,讓我們很方便在網上做資料處理,但花很少的錢. 真的很苦

- 劉妍群

透過深入運用 AWS 技能，我們成功打造了一個高度可擴展且安全的雲端基礎架構。AWS 提供的各種服務，如 EC2、S3、Lambda 等，強化了我們專案的運算效能和數據存儲能力。然而，在使用 AWS 的過程中，我們發現需要進一步優化成本控制，尤其是在不同服務間的資源配置上。未來，我們計劃透過 AWS cloud9 等工具更精確地監控和管理開支。

在團隊協作方面，還有進一步改進的空間。未來，我們將加強培訓團隊成員，以提高 AWS 服務的使用技能，並探索更多協作工具，以優化開發流程。總體而言，這次經驗深化了我們對 AWS 的了解，並提醒了我們在使用雲端服務時需要持續追求效能和安全的平衡。

- 陳奕辰

這學期的期末作業中，我運用 AWS 的 Cloud9 和 Python，成功下載並分析了國道 MA06 的交通數據。Cloud9 的整合開發環境簡化了配置，提高了效率。透過 Python，我取得了交通流量、車速等數據，並進行清理和統計分析。這使我能夠得出有價值的結論，例如高峰時段。這次作業不

僅滿足了課程要求，還深化了我對雲端運算和資料分析的理解，同時提升了 **AWS** 和 **Python** 的應用能力。

