Task 2:

I have chosen Excel as tool for this task.

Notice: There are one problem with my excel language.

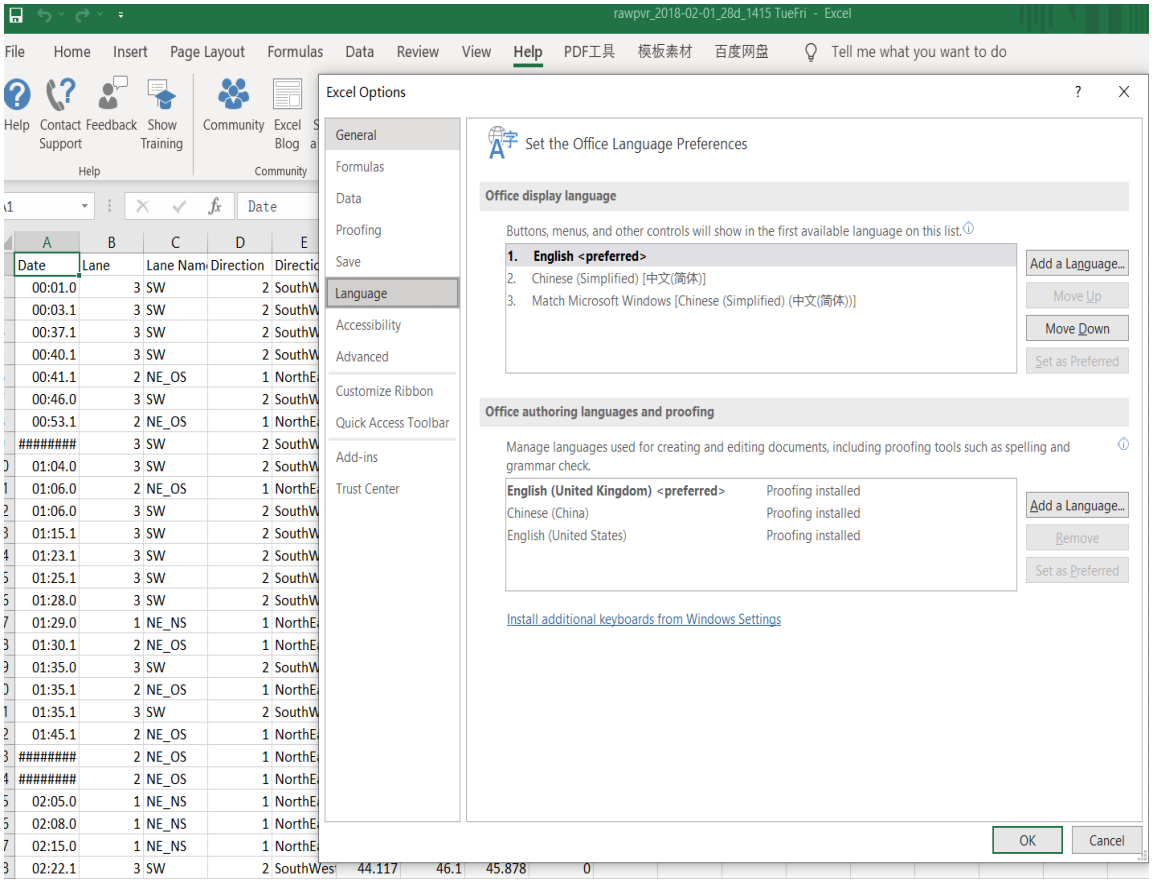


Figure 1. language setting

As excel tutorial suggested, the language of this excel option had been set as “English”. However, some values and functions still show Chinese.

Output:

	A	B	C	D	E	F	G
1	Day of week	星期二					
2	Direction Name	North					
3	Hours	9					
4							
5	**Row Labels**	**Count of Date**					
6	**6-2月**	**2537**					
7	**13-2月**	**2447**					
8	**20-2月**	**2311**					
9	**27-2月**	**2426**					
10	**Grand Total**	**9721**					
11							
12	Range:	226					
13	1st Quartile:	2397.25					
14	2nd Quartile:	2436.5					
15	3rd Quartile	2469.5					
16	Interquartile range:	72.25					
17							
18							
19							
20							
21							

Figure 2.task2 output

Step-by-step description of the development of the task:

Step 1: Add one more column for day of week

Since the requirement of this task ask on Tuesday only. At the same time, the provided data does not have such information in the table. Therefore, the first step is right clicking the cell and choose “Insert”

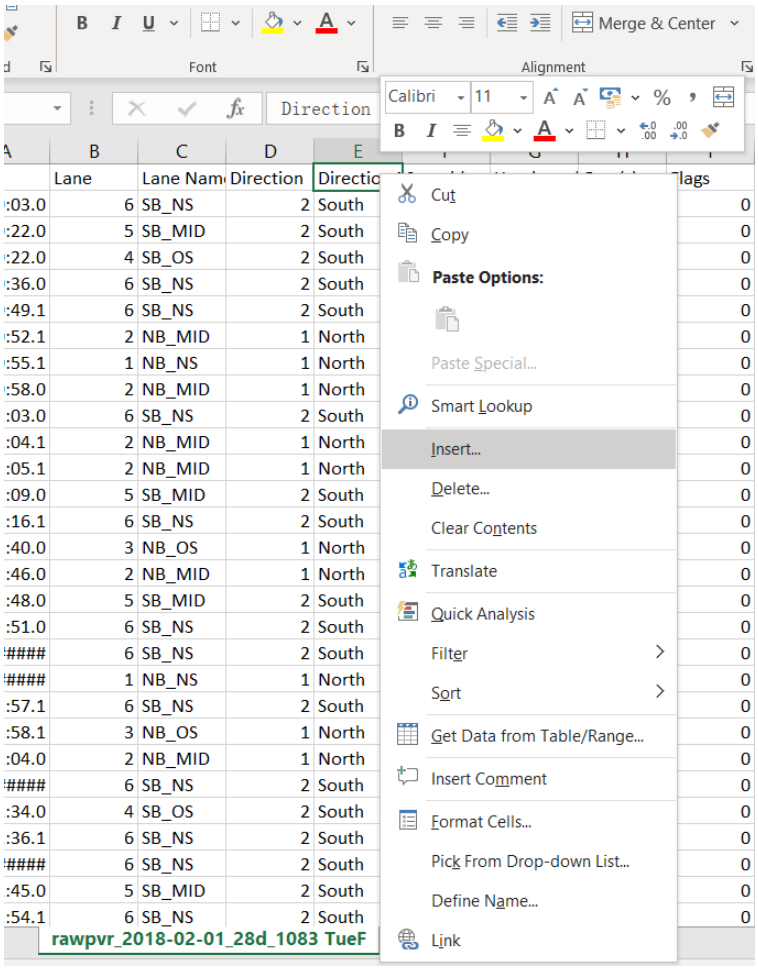
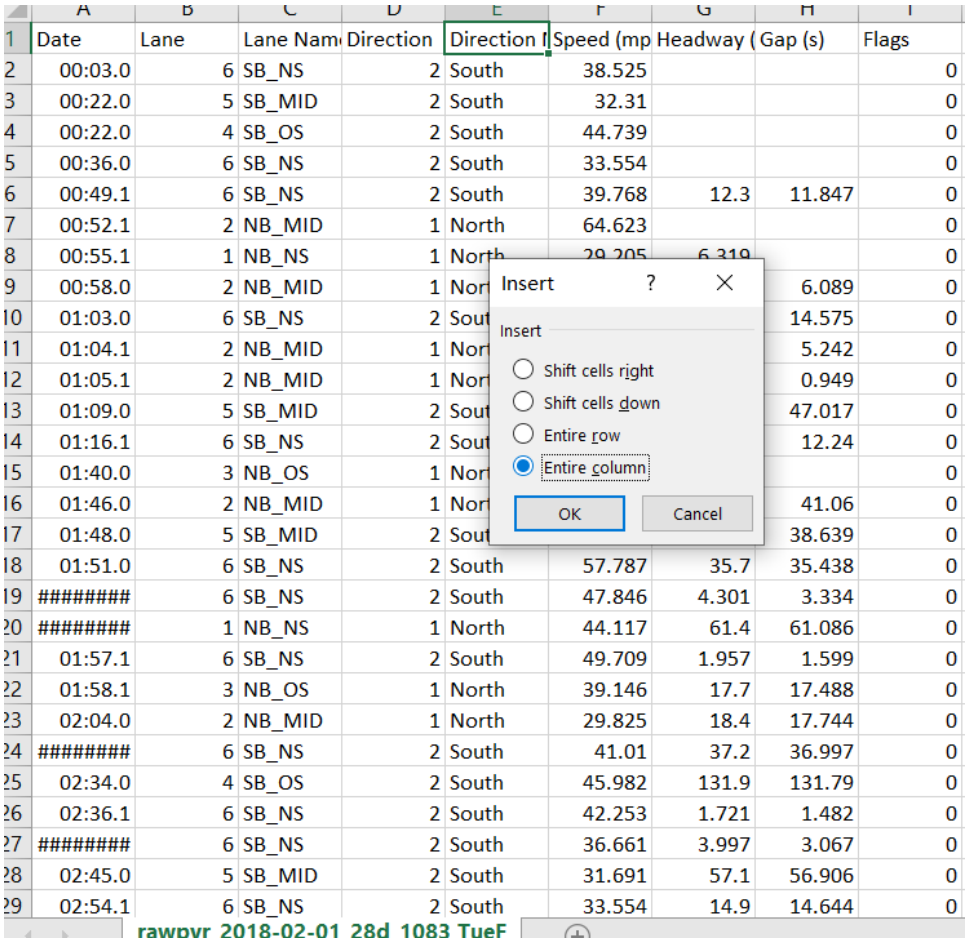


Figure 3. insert

Then select “Entire column”. Since we want to filter this information later.

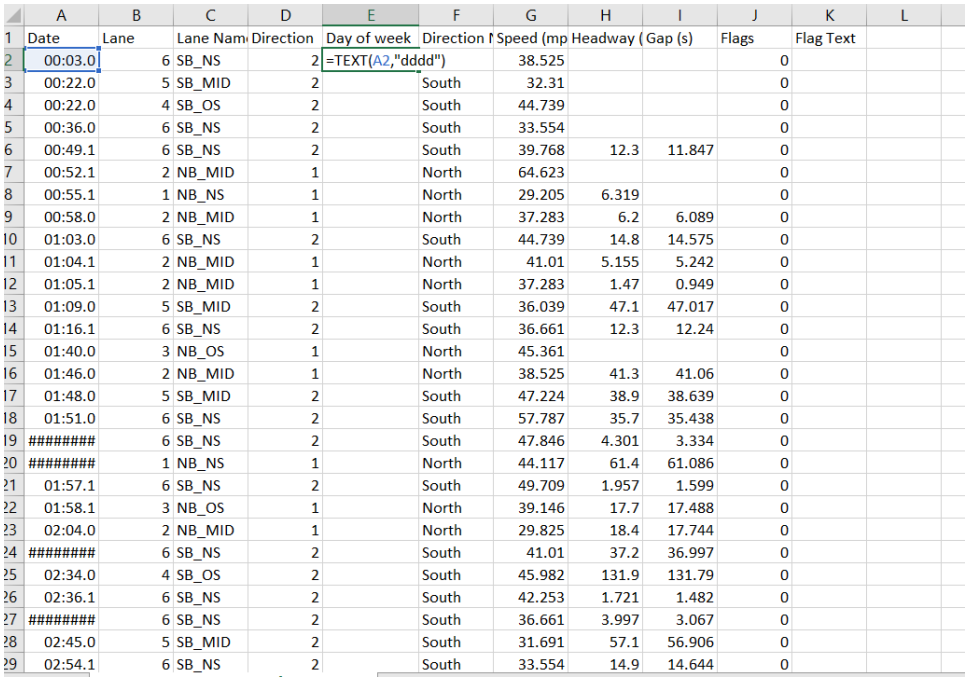


	A	B	C	D	E	F	G	H	I
1	Date	Lane	Lane Nam	Direction	Direction I	Speed (mp	Headway (	Gap (s)	Flags
2	00:03.0	6	SB_NS	2	South	38.525			0
3	00:22.0	5	SB_MID	2	South	32.31			0
4	00:22.0	4	SB_OS	2	South	44.739			0
5	00:36.0	6	SB_NS	2	South	33.554			0
6	00:49.1	6	SB_NS	2	South	39.768	12.3	11.847	0
7	00:52.1	2	NB_MID	1	North	64.623			0
8	00:55.1	1	NB_NS	1	North	29.205	6.319		0
9	00:58.0	2	NB_MID	1	Nort			6.089	0
10	01:03.0	6	SB_NS	2	Sout			14.575	0
11	01:04.1	2	NB_MID	1	Nort			5.242	0
12	01:05.1	2	NB_MID	1	Nort			0.949	0
13	01:09.0	5	SB_MID	2	Sout			47.017	0
14	01:16.1	6	SB_NS	2	Sout			12.24	0
15	01:40.0	3	NB_OS	1	Nort				0
16	01:46.0	2	NB_MID	1	Nort			41.06	0
17	01:48.0	5	SB_MID	2	Sout			38.639	0
18	01:51.0	6	SB_NS	2	South	57.787	35.7	35.438	0
19	########	6	SB_NS	2	South	47.846	4.301	3.334	0
20	########	1	NB_NS	1	North	44.117	61.4	61.086	0
21	01:57.1	6	SB_NS	2	South	49.709	1.957	1.599	0
22	01:58.1	3	NB_OS	1	North	39.146	17.7	17.488	0
23	02:04.0	2	NB_MID	1	North	29.825	18.4	17.744	0
24	########	6	SB_NS	2	South	41.01	37.2	36.997	0
25	02:34.0	4	SB_OS	2	South	45.982	131.9	131.79	0
26	02:36.1	6	SB_NS	2	South	42.253	1.721	1.482	0
27	########	6	SB_NS	2	South	36.661	3.997	3.067	0
28	02:45.0	5	SB_MID	2	South	31.691	57.1	56.906	0
29	02:54.1	6	SB_NS	2	South	33.554	14.9	14.644	0

Figure 4. insert entire column

Step 2: Assign “day of week” attribute to all data

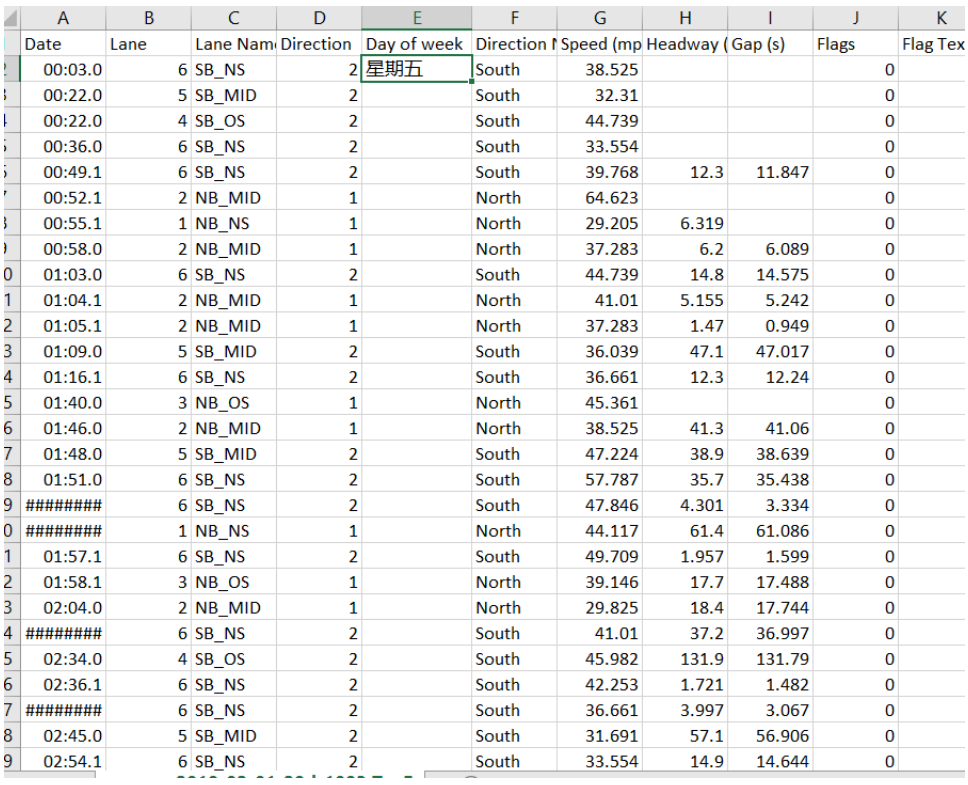
Using function = text(cell,dddd) to convert Date to Day of Week



	A	B	C	D	E	F	G	H	I	J	K	L
1	Date	Lane	Lane Nam	Direction	Day of week	Direction N	Speed (mp	Headway (	Gap (s)	Flags	Flag Text	
2	00:03.0	6	SB_NS	2	=TEXT(A2,"dddd")		38.525			0		
3	00:22.0	5	SB_MID	2		South	32.31			0		
4	00:22.0	4	SB_OS	2		South	44.739			0		
5	00:36.0	6	SB_NS	2		South	33.554			0		
6	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0		
7	00:52.1	2	NB_MID	1		North	64.623			0		
8	00:55.1	1	NB_NS	1		North	29.205	6.319		0		
9	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0		
10	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0		
11	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0		
12	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0		
13	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0		
14	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0		
15	01:40.0	3	NB_OS	1		North	45.361			0		
16	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0		
17	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0		
18	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0		
19	########	6	SB_NS	2		South	47.846	4.301	3.334	0		
20	########	1	NB_NS	1		North	44.117	61.4	61.086	0		
21	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0		
22	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0		
23	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0		
24	########	6	SB_NS	2		South	41.01	37.2	36.997	0		
25	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0		
26	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0		
27	########	6	SB_NS	2		South	36.661	3.997	3.067	0		
28	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0		
29	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0		

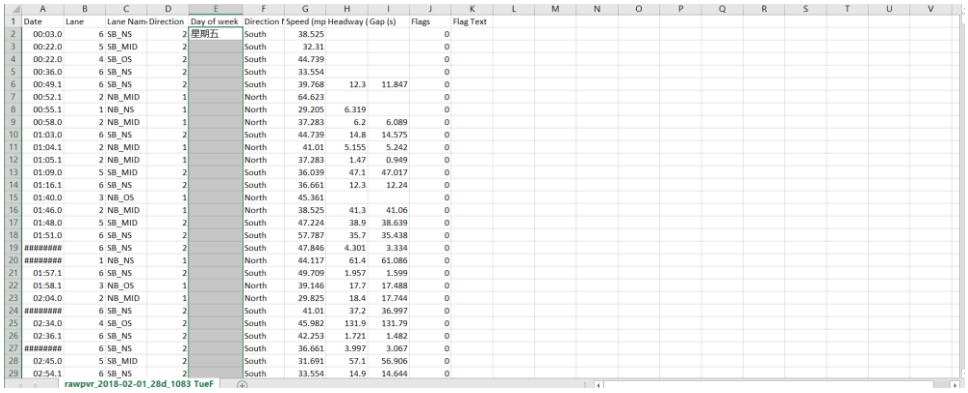
Figure 5. function input

Then click the right-bottom corner in this E2 cell. Click the black Cross symbol to add day of week attribute for all data.



	A	B	C	D	E	F	G	H	I	J	K
	Date	Lane	Lane Nam	Direction	Day of week	Direction N	Speed (mp	Headway (	Gap (s)	Flags	Flag Tex
	00:03.0	6	SB_NS	2	星期五	South	38.525			0	
	00:22.0	5	SB_MID	2		South	32.31			0	
	00:22.0	4	SB_OS	2		South	44.739			0	
	00:36.0	6	SB_NS	2		South	33.554			0	
	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0	
	00:52.1	2	NB_MID	1		North	64.623			0	
	00:55.1	1	NB_NS	1		North	29.205	6.319		0	
	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0	
0	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0	
1	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0	
2	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0	
3	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0	
4	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0	
5	01:40.0	3	NB_OS	1		North	45.361			0	
6	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0	
7	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0	
8	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0	
9	########	6	SB_NS	2		South	47.846	4.301	3.334	0	
0	########	1	NB_NS	1		North	44.117	61.4	61.086	0	
1	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0	
2	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0	
3	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0	
4	########	6	SB_NS	2		South	41.01	37.2	36.997	0	
5	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0	
6	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0	
7	########	6	SB_NS	2		South	36.661	3.997	3.067	0	
8	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0	
9	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0	

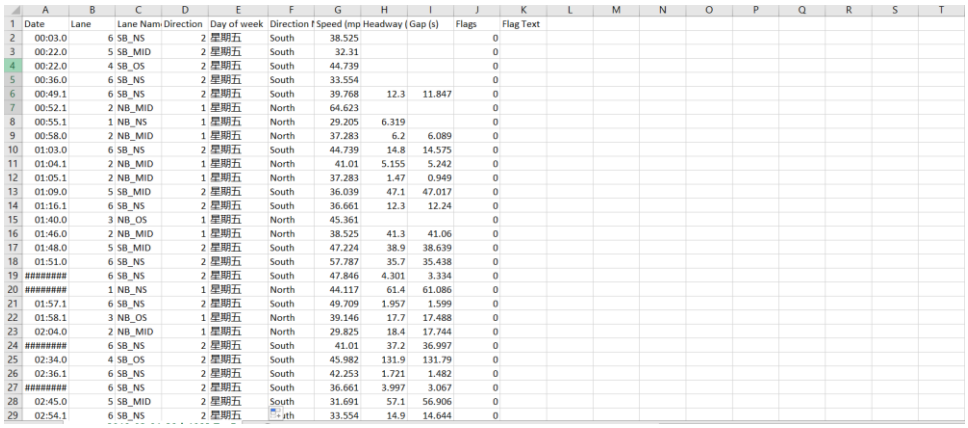
Figure 6. Function output



	A	B	C	D	E	F	G	H	I	J	K
1	Date	Lane	Lane Nam	Direction	Day of week	Direction f	Speed (mp	Headway (	Gap (s)	Flags	Flag Text
2	00:03.0	6	SB_NS	2	星期五	South	38.525			0	
3	00:22.0	5	SB_MID	2		South	32.31			0	
4	00:22.0	4	SB_OS	2		South	44.739			0	
5	00:36.0	6	SB_NS	2		South	33.554			0	
6	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0	
7	00:52.1	2	NB_MID	1		North	64.623			0	
8	00:55.1	1	NB_NS	1		North	29.205	6.319		0	
9	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0	
10	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0	
11	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0	
12	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0	
13	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0	
14	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0	
15	01:40.0	3	NB_OS	1		North	45.361			0	
16	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0	
17	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0	
18	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0	
19	########	6	SB_NS	2		South	47.846	4.301	3.334	0	
20	########	1	NB_NS	1		North	44.117	61.4	61.086	0	
21	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0	
22	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0	
23	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0	
24	########	6	SB_NS	2		South	41.01	37.2	36.997	0	
25	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0	
26	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0	
27	########	6	SB_NS	2		South	36.661	3.997	3.067	0	
28	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0	
29	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0	

rawpvr_2018-02-01_28d_1083 TueF

Figure 7.Assign to whole table



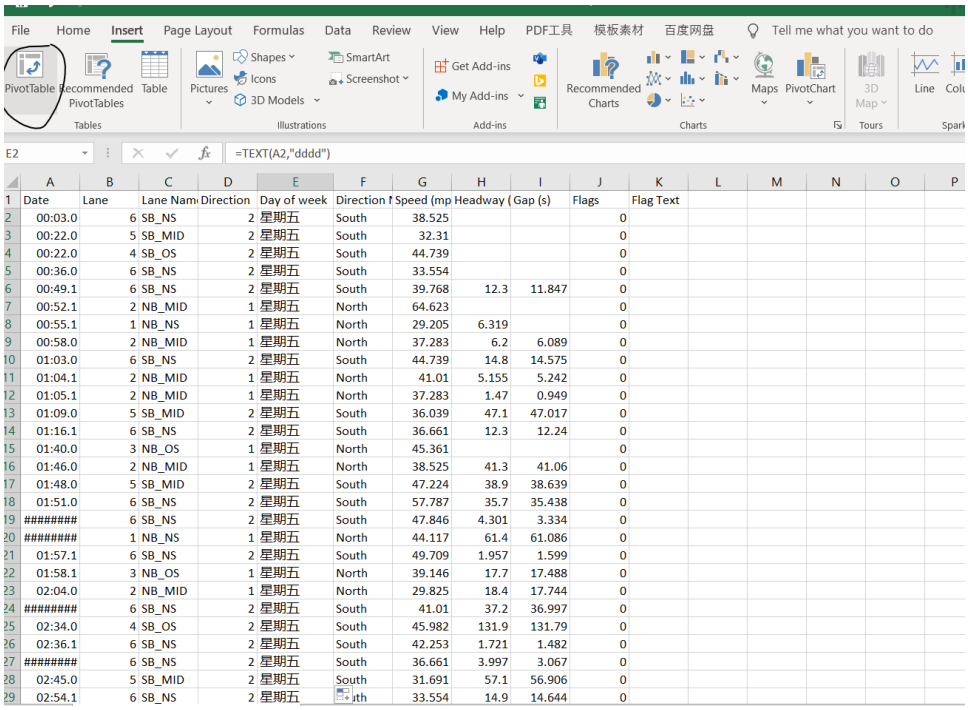
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Date	Lane	Lane Nam	Direction	Day of week	Direction f	Speed (mp	Headway (	Gap (s)	Flags	Flag Text									
2	00:03.0	6	SB_NS	2	星期五	South	38.525			0										
3	00:22.0	5	SB_MID	2	星期五	South	32.31			0										
4	00:22.0	4	SB_OS	2	星期五	South	44.739			0										
5	00:36.0	6	SB_NS	2	星期五	South	33.554			0										
6	00:49.1	6	SB_NS	2	星期五	South	39.768	12.3	11.847	0										
7	00:52.1	2	NB_MID	1	星期五	North	64.623			0										
8	00:55.1	1	NB_NS	1	星期五	North	29.205	6.319		0										
9	00:58.0	2	NB_MID	1	星期五	North	37.283	6.2	6.089	0										
10	01:03.0	6	SB_NS	2	星期五	South	44.739	14.8	14.575	0										
11	01:04.1	2	NB_MID	1	星期五	North	41.01	5.155	5.242	0										
12	01:05.1	2	NB_MID	1	星期五	North	37.283	1.47	0.949	0										
13	01:09.0	5	SB_MID	2	星期五	South	36.039	47.1	47.017	0										
14	01:16.1	6	SB_NS	2	星期五	South	36.661	12.3	12.24	0										
15	01:40.0	3	NB_OS	1	星期五	North	45.361			0										
16	01:46.0	2	NB_MID	1	星期五	North	38.525	41.3	41.06	0										
17	01:48.0	5	SB_MID	2	星期五	South	47.224	38.9	38.639	0										
18	01:51.0	6	SB_NS	2	星期五	South	57.787	35.7	35.438	0										
19	########	6	SB_NS	2	星期五	South	47.846	4.301	3.334	0										
20	########	1	NB_NS	1	星期五	North	44.117	61.4	61.086	0										
21	01:57.1	6	SB_NS	2	星期五	South	49.709	1.957	1.599	0										
22	01:58.1	3	NB_OS	1	星期五	North	39.146	17.7	17.488	0										
23	02:04.0	2	NB_MID	1	星期五	North	29.825	18.4	17.744	0										
24	########	6	SB_NS	2	星期五	South	41.01	37.2	36.997	0										
25	02:34.0	4	SB_OS	2	星期五	South	45.982	131.9	131.79	0										
26	02:36.1	6	SB_NS	2	星期五	South	42.253	1.721	1.482	0										
27	########	6	SB_NS	2	星期五	South	36.661	3.997	3.067	0										
28	02:45.0	5	SB_MID	2	星期五	South	31.691	57.1	56.906	0										
29	02:54.1	6	SB_NS	2	星期五	South	33.554	14.9	14.644	0										

Figure 8. Assigned result

As the result of step2, all data has “day of week” attribute. It will helped filter operation in later steps.

Step3: Create Pivot table

The Pivot table is helpful tool for filter with multiple criteria, therefore, this step will create pivot table for provided data.



	A	B	C	D	E	F	G	H	I	J	K
1	Date	Lane	Lane Nam	Direction	Day of week	Direction I	Speed (mp	Headway (	Gap (s)	Flags	Flag Text
2	00:03.0	6	SB_NS	2	星期五	South	38.525			0	
3	00:22.0	5	SB_MID	2	星期五	South	32.31			0	
4	00:22.0	4	SB_OS	2	星期五	South	44.739			0	
5	00:36.0	6	SB_NS	2	星期五	South	33.554			0	
6	00:49.1	6	SB_NS	2	星期五	South	39.768	12.3	11.847	0	
7	00:52.1	2	NB_MID	1	星期五	North	64.623			0	
8	00:55.1	1	NB_NS	1	星期五	North	29.205	6.319		0	
9	00:58.0	2	NB_MID	1	星期五	North	37.283	6.2	6.089	0	
10	01:03.0	6	SB_NS	2	星期五	South	44.739	14.8	14.575	0	
11	01:04.1	2	NB_MID	1	星期五	North	41.01	5.155	5.242	0	
12	01:05.1	2	NB_MID	1	星期五	North	37.283	1.47	0.949	0	
13	01:09.0	5	SB_MID	2	星期五	South	36.039	47.1	47.017	0	
14	01:16.1	6	SB_NS	2	星期五	South	36.661	12.3	12.24	0	
15	01:40.0	3	NB_OS	1	星期五	North	45.361			0	
16	01:46.0	2	NB_MID	1	星期五	North	38.525	41.3	41.06	0	
17	01:48.0	5	SB_MID	2	星期五	South	47.224	38.9	38.639	0	
18	01:51.0	6	SB_NS	2	星期五	South	57.787	35.7	35.438	0	
19	########	6	SB_NS	2	星期五	South	47.846	4.301	3.334	0	
20	########	1	NB_NS	1	星期五	North	44.117	61.4	61.086	0	
21	01:57.1	6	SB_NS	2	星期五	South	49.709	1.957	1.599	0	
22	01:58.1	3	NB_OS	1	星期五	North	39.146	17.7	17.488	0	
23	02:04.0	2	NB_MID	1	星期五	North	29.825	18.4	17.744	0	
24	########	6	SB_NS	2	星期五	South	41.01	37.2	36.997	0	
25	02:34.0	4	SB_OS	2	星期五	South	45.982	131.9	131.79	0	
26	02:36.1	6	SB_NS	2	星期五	South	42.253	1.721	1.482	0	
27	########	6	SB_NS	2	星期五	South	36.661	3.997	3.067	0	
28	02:45.0	5	SB_MID	2	星期五	South	31.691	57.1	56.906	0	
29	02:54.1	6	SB_NS	2	星期五	South	33.554	14.9	14.644	0	

Figure 9. choose pivottable

In the top-left corner, click the “Pivot Table” and select range.

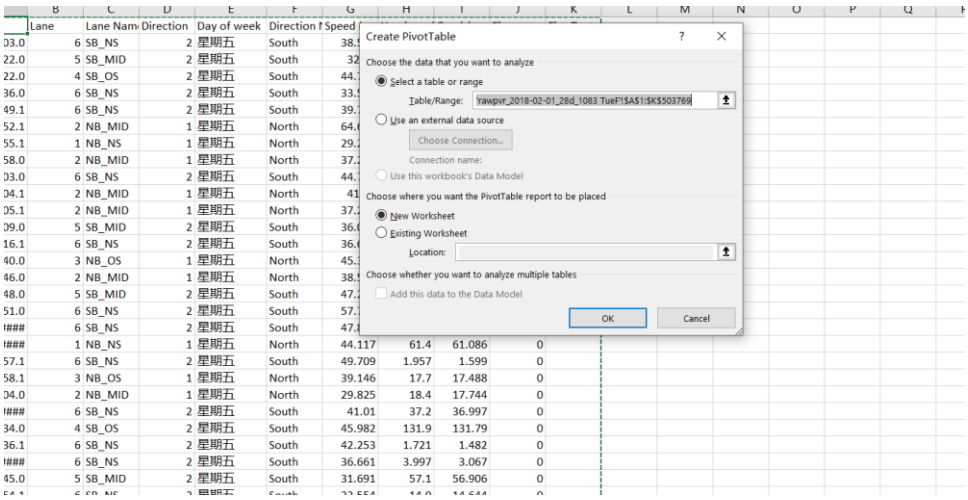


Figure 10. pivottable range

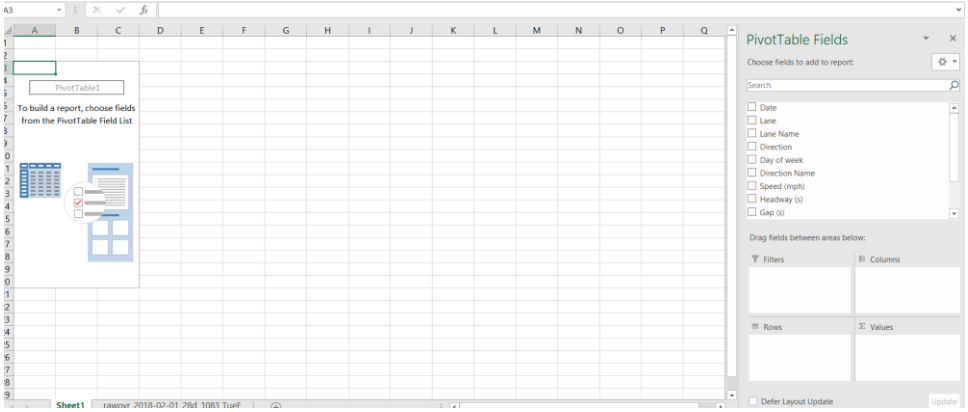


Figure 11. pivottable

Step 4: add multiple criteria in “Pivot Table Fields”.

The requirement of task is “Provide a simple profile of the traffic volume of the North lanes (considering all the North lanes together) of site 1083 using the following descriptive data summarization measures and focusing only on Tuesdays between 09:00 am and 09:59:59 am”

Therefore, there are three key criteria. North, 9:00 am to 9:59:59 am, and Tuesday.

The “Day of week”, “Direction name” and “Hours” have been added in the “Filters” area. The date also added to “Rows area”.

Figure 12. pivottable add filter

Choose Tuesday for this “Day of week” criteria. (“星期二” is the Tuesday in Chinese)

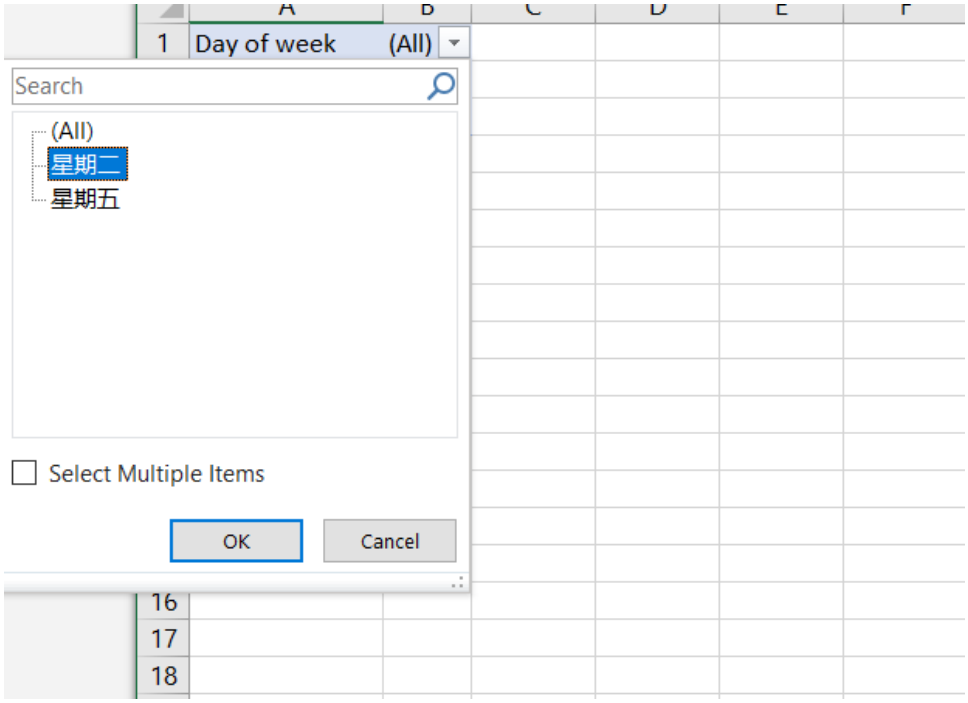


Figure 13. pivottable filter day of week

Choose “North” in “Direction Name” criteria.

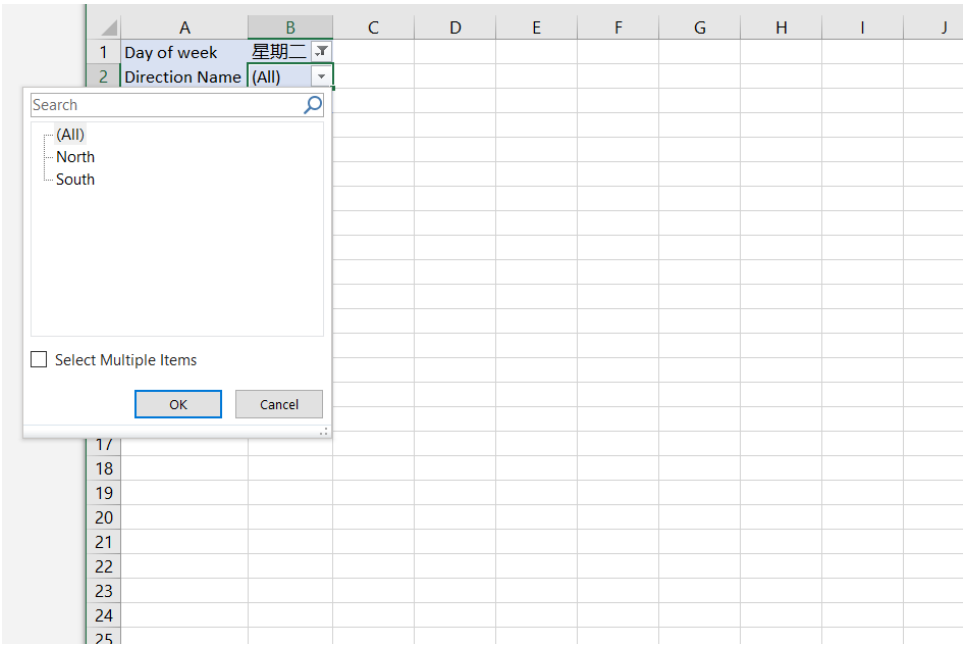


Figure 14. pivottable filter direction name

Choose “9” in “Hours” criteria.

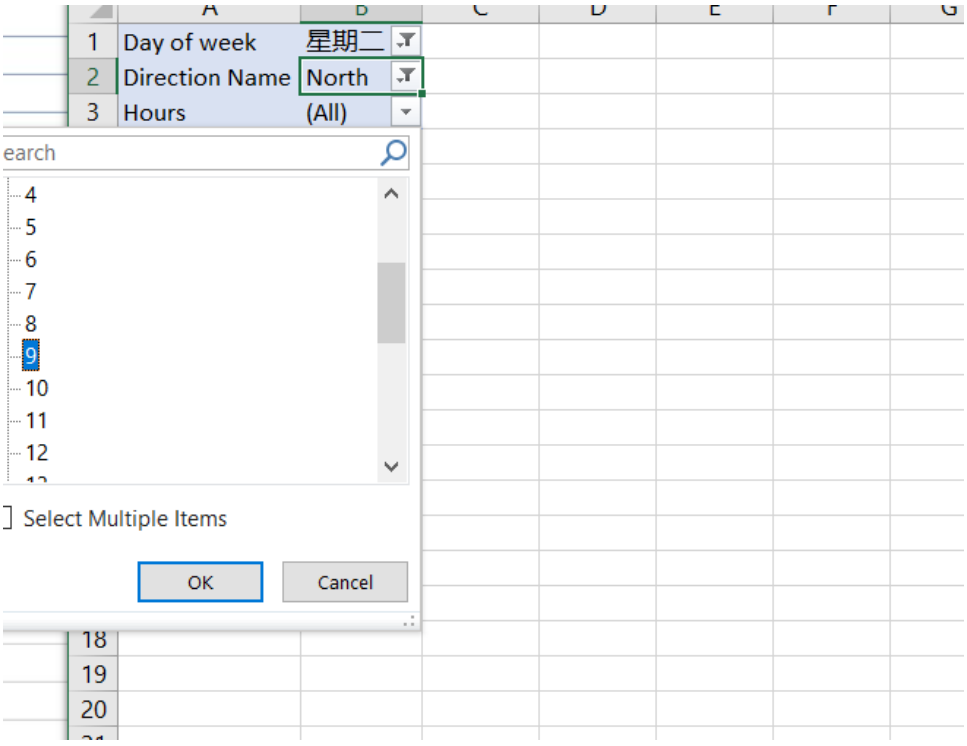


Figure 15. pivottable filter hour

Therefore, all criteria have been set for this task, then put a date to the Values area in Pivot table fields. It would be shown the traffic volume for each day

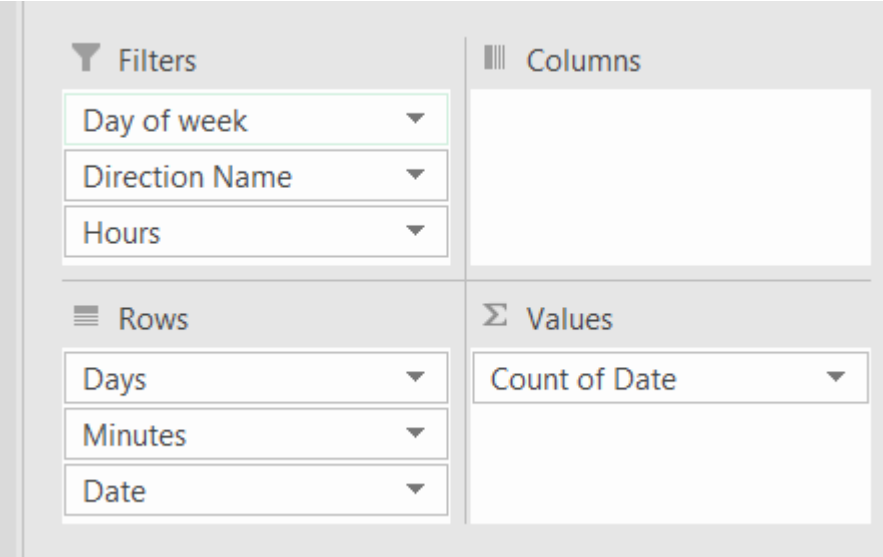


Figure 16. pivottable value

Here is the result about the traffic volume of the North lanes on Tuesday 9am.

	A	B
1	Day of week	星期二
2	Direction Name	North
3	Hours	9
4		
5	**Row Labels**	**Count of Date**
6	**6-2月**	**2537**
7	**13-2月**	**2447**
8	**20-2月**	**2311**
9	**27-2月**	**2426**
10	**Grand Total**	**9721**
11		
12		
13		
14		
15		
16		
17		

Figure 17. total count

Step 5: Calculate values from these 4 data

To get range, it uses the max function to get the maximum value from data. Then use the min function to get the minimum value from data. Then minus to get the range of these data.

12	Range:	=MAX(B6:B9)-MIN(B6:B9)
13	1st Quartile:	
14	2nd Quartile:	
15	3rd Quartile	
16	Interquartile range:	
17		
18		

Figure 18. range function

For Quartile values, it uses quartile function. This 1 means we want to get the 1st quartile values.

Row Labels	Count of Date
⊞6-2月	2537
⊞13-2月	2447
⊞20-2月	2311
⊞27-2月	2426
Grand Total	9721
Range:	226
1st Quartile:	=quartile(B6:B9,1)
2nd Quartile:	
3rd Quartile	
Interquartile range:	

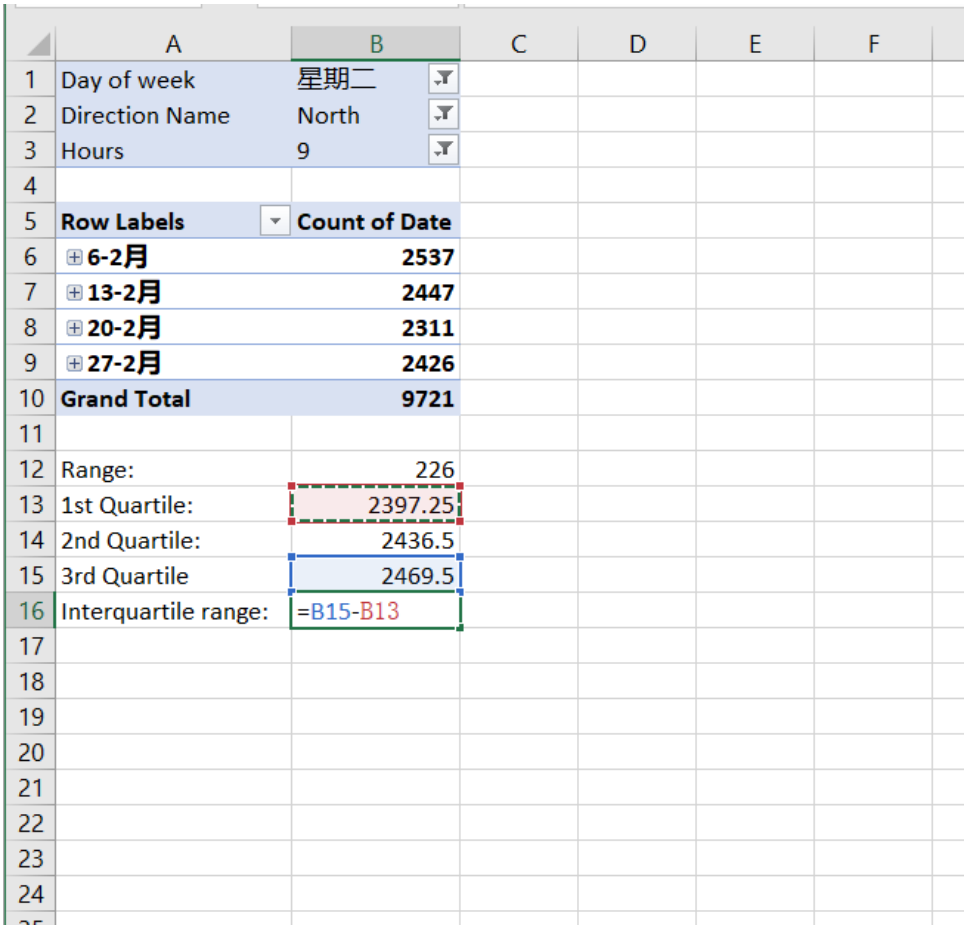
Figure 19. quartile function

For the 2nd Quartile and 3rd Quartile, the step is similar to how the 1st Quartile was generated. The only difference is we need to put 2 and 3 in the last position inside the function.

	A	B	C	D	E	F	G	H	I	J
1	Day of week	星期二								
2	Direction Name	North								
3	Hours	9								
4										
5	**Row Labels**	**Count of Date**								
6	⊞**6-2月**	**2537**								
7	⊞**13-2月**	**2447**								
8	⊞**20-2月**	**2311**								
9	⊞**27-2月**	**2426**								
10	**Grand Total**	**9721**								
11										
12	Range:	226								
13	1st Quartile:	2397.25								
14	2nd Quartile:	2436.5								
15	3rd Quartile	2469.5								
16	Interquartile range:									
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										

Figure 20. quartile result

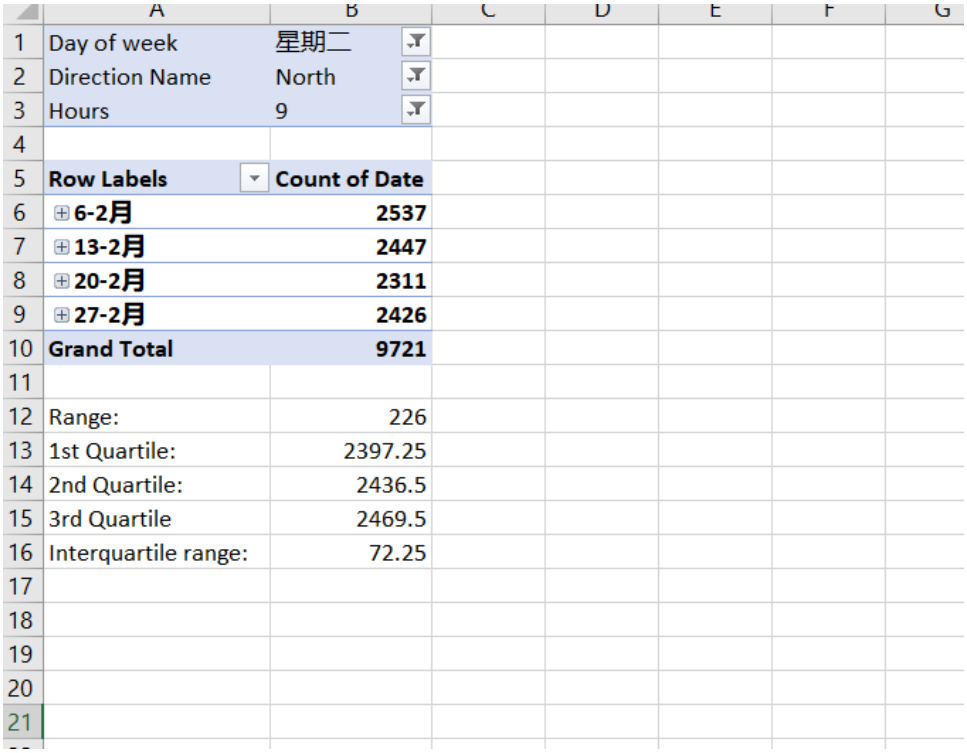
Finally, we will get the interquartile range of these data. The interquartile range = 3rd Quartile - 1st Quartile. Therefore, we use this function here:



	A	B	C	D	E	F
1	Day of week	星期二				
2	Direction Name	North				
3	Hours	9				
4						
5	**Row Labels**	**Count of Date**				
6	**6-2月**	**2537**				
7	**13-2月**	**2447**				
8	**20-2月**	**2311**				
9	**27-2月**	**2426**				
10	**Grand Total**	**9721**				
11						
12	Range:	226				
13	1st Quartile:	2397.25				
14	2nd Quartile:	2436.5				
15	3rd Quartile	2469.5				
16	Interquartile range:	=B15-B13				
17						
18						
19						
20						
21						
22						
23						
24						

Figure 21. IQR function

Here is the 5 final values we calculated by 'rawpvr_2018-02- 01_28d_1083 TueFri.csv' data.



	A	B	C	D	E	F	G
1	Day of week	星期二					
2	Direction Name	North					
3	Hours	9					
4							
5	**Row Labels**	**Count of Date**					
6	**6-2月**	**2537**					
7	**13-2月**	**2447**					
8	**20-2月**	**2311**					
9	**27-2月**	**2426**					
10	**Grand Total**	**9721**					
11							
12	Range:	226					
13	1st Quartile:	2397.25					
14	2nd Quartile:	2436.5					
15	3rd Quartile	2469.5					
16	Interquartile range:	72.25					
17							
18							
19							
20							
21							

Figure 22. task 2 result

Range: 226
1st Quartile: 2397.25
2nd Quartile :2436.5
3rd Quartile :2469.5
Interquartile range:72.25

An interpretation of the results:
In conclusion, these 5 data have provided some information for data analysis. The traffic volume is stable in these 4 days at 9 am. The 20.2 has the lowest traffic volume in these 4 days. From some basic search [1], it said 20/02/2018 was a popular holiday. It is a possible reason to explain why the traffic number is lower than others significantly. The 6.2 has the highest traffic volume. For the other two days, the traffic volume is quite close.

Task 3:

I have chosen Tuesday for this task.

Output:

(South)

Each hour	The average number of traffic
00:00 - 00:59	162.75
01:00 - 01:59	92.75
02:00 - 02:59	77.5
03:00 - 03:59	79.75
04:00 - 04:59	120.5
05:00 - 05:59	289.75
06:00 - 06:59	862.25
07:00 - 07:59	2015
08:00 - 08:59	2135
09:00 - 09:59	1548.25
10:00 - 10:59	1402
11:00 - 11:59	1543.5
12:00 - 12:59	1709.25
13:00 - 13:59	1749.5
14:00 - 14:59	1992
15:00 - 15:59	2481.5
16:00 - 16:59	3001.25
17:00 - 17:59	2794
18:00 - 18:59	2330.5
19:00 - 19:59	1601.75
20:00 - 20:59	1080.25
21:00 - 21:59	847
22:00 - 22:59	612.75
23:00 - 23:59	331

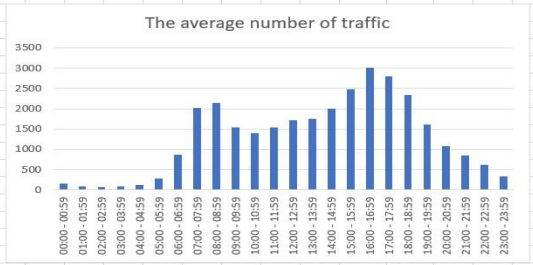


Figure 23. task 3 output south

(North)

Each Our	The average number of traffic
00:00-00:59	126.5
01:00-01:59	63
02:00-02:59	42.5
03:00-03:59	53
04:00-04:59	108.25
05:00-05:59	361.5
06:00-06:59	1260
07:00-07:59	2951.5
08:00-08:59	2727
09:00-09:59	2430.25
10:00-10:59	1776.75
11:00-11:59	1634.25
12:00-12:59	1698.75
13:00-13:59	1707.5
14:00-14:59	1642
15:00-15:59	1805.25
16:00-16:59	2429
17:00-17:59	2655.75
18:00-18:59	2121.5
19:00-19:59	1308.25
20:00-20:59	882.75
21:00-21:59	648.75
22:00-22:59	453.75
23:00-23:59	256.75

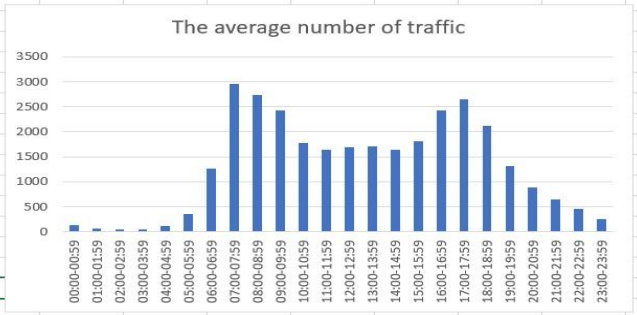


Figure 24. task 3 output north

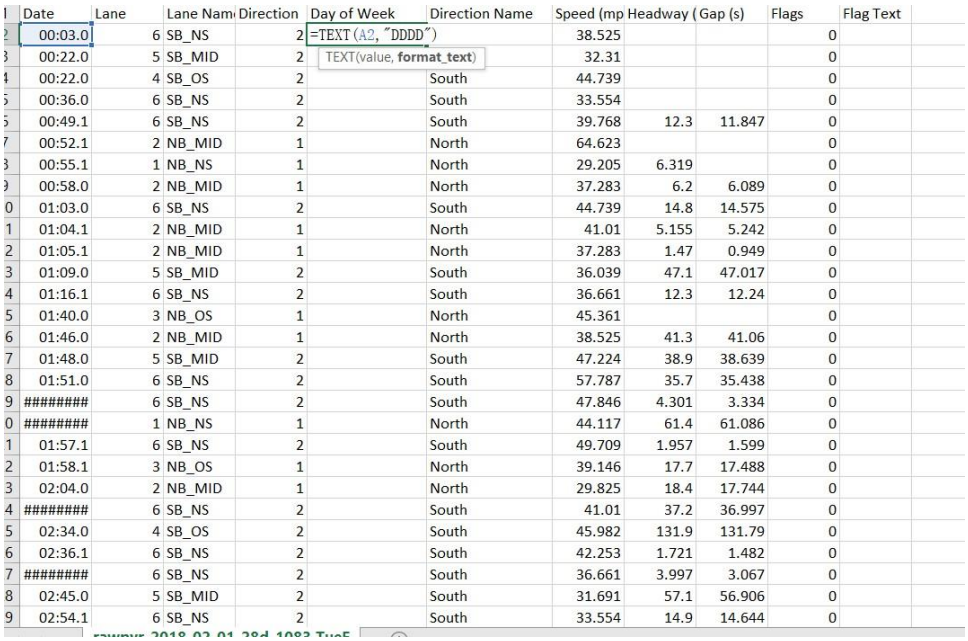
Step-by-step description of the development of the task:(South)

Step 1: Create “day of week” attribute for filter later.

Similar with the step 1 and 2 in task2. Due to the task requires “Choose a day of the week”. Therefore, the first step is to assign this attribute to support filter later.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date	Lane	Lane Nam	Direction	Day of Week	Direction Name	Speed (mp	Headway (	Gap (s)	Flags	Flag Text		
2	00:03.0	6	SB_NS	2		South	38.525			0			
3	00:22.0	5	SB_MID	2		South	32.31			0			
4	00:22.0	4	SB_OS	2		South	44.739			0			
5	00:36.0	6	SB_NS	2		South	33.554			0			
6	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0			
7	00:52.1	2	NB_MID	1		North	64.623			0			
8	00:55.1	1	NB_NS	1		North	29.205	6.319		0			
9	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0			
10	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0			
11	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0			
12	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0			
13	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0			
14	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0			
15	01:40.0	3	NB_OS	1		North	45.361			0			
16	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0			
17	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0			
18	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0			
19	########	6	SB_NS	2		South	47.846	4.301	3.334	0			
20	########	1	NB_NS	1		North	44.117	61.4	61.086	0			
21	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0			
22	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0			
23	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0			
24	########	6	SB_NS	2		South	41.01	37.2	36.997	0			
25	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0			
26	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0			
27	########	6	SB_NS	2		South	36.661	3.997	3.067	0			
28	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0			
29	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0			

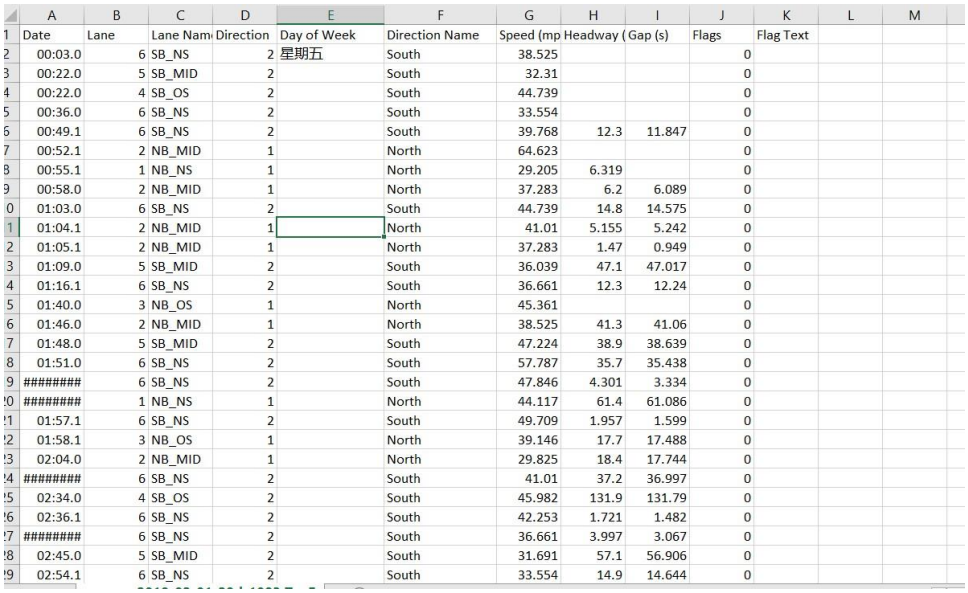
Figure 25. Add column



	Date	Lane	Lane Nam	Direction	Day of Week	Direction Name	Speed (mp	Headway (	Gap (s)	Flags	Flag Text
2	00:03.0	6	SB_NS	2	=TEXT(A2, "DDDD")		38.525			0	
3	00:22.0	5	SB_MID	2		South	32.31			0	
4	00:22.0	4	SB_OS	2		South	44.739			0	
5	00:36.0	6	SB_NS	2		South	33.554			0	
6	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0	
7	00:52.1	2	NB_MID	1		North	64.623			0	
8	00:55.1	1	NB_NS	1		North	29.205	6.319		0	
9	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0	
10	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0	
11	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0	
12	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0	
13	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0	
14	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0	
15	01:40.0	3	NB_OS	1		North	45.361			0	
16	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0	
17	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0	
18	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0	
19	########	6	SB_NS	2		South	47.846	4.301	3.334	0	
20	########	1	NB_NS	1		North	44.117	61.4	61.086	0	
21	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0	
22	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0	
23	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0	
24	########	6	SB_NS	2		South	41.01	37.2	36.997	0	
25	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0	
26	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0	
27	########	6	SB_NS	2		South	36.661	3.997	3.067	0	
28	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0	
29	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0	

TEXT(value, **format_text**)

Figure 26. week of day1



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date	Lane	Lane Nam	Direction	Day of Week	Direction Name	Speed (mp	Headway (	Gap (s)	Flags	Flag Text		
2	00:03.0	6	SB_NS	2	星期五	South	38.525			0			
3	00:22.0	5	SB_MID	2		South	32.31			0			
4	00:22.0	4	SB_OS	2		South	44.739			0			
5	00:36.0	6	SB_NS	2		South	33.554			0			
6	00:49.1	6	SB_NS	2		South	39.768	12.3	11.847	0			
7	00:52.1	2	NB_MID	1		North	64.623			0			
8	00:55.1	1	NB_NS	1		North	29.205	6.319		0			
9	00:58.0	2	NB_MID	1		North	37.283	6.2	6.089	0			
10	01:03.0	6	SB_NS	2		South	44.739	14.8	14.575	0			
11	01:04.1	2	NB_MID	1		North	41.01	5.155	5.242	0			
12	01:05.1	2	NB_MID	1		North	37.283	1.47	0.949	0			
13	01:09.0	5	SB_MID	2		South	36.039	47.1	47.017	0			
14	01:16.1	6	SB_NS	2		South	36.661	12.3	12.24	0			
15	01:40.0	3	NB_OS	1		North	45.361			0			
16	01:46.0	2	NB_MID	1		North	38.525	41.3	41.06	0			
17	01:48.0	5	SB_MID	2		South	47.224	38.9	38.639	0			
18	01:51.0	6	SB_NS	2		South	57.787	35.7	35.438	0			
19	########	6	SB_NS	2		South	47.846	4.301	3.334	0			
20	########	1	NB_NS	1		North	44.117	61.4	61.086	0			
21	01:57.1	6	SB_NS	2		South	49.709	1.957	1.599	0			
22	01:58.1	3	NB_OS	1		North	39.146	17.7	17.488	0			
23	02:04.0	2	NB_MID	1		North	29.825	18.4	17.744	0			
24	########	6	SB_NS	2		South	41.01	37.2	36.997	0			
25	02:34.0	4	SB_OS	2		South	45.982	131.9	131.79	0			
26	02:36.1	6	SB_NS	2		South	42.253	1.721	1.482	0			
27	########	6	SB_NS	2		South	36.661	3.997	3.067	0			
28	02:45.0	5	SB_MID	2		South	31.691	57.1	56.906	0			
29	02:54.1	6	SB_NS	2		South	33.554	14.9	14.644	0			

Figure 27. week of day2

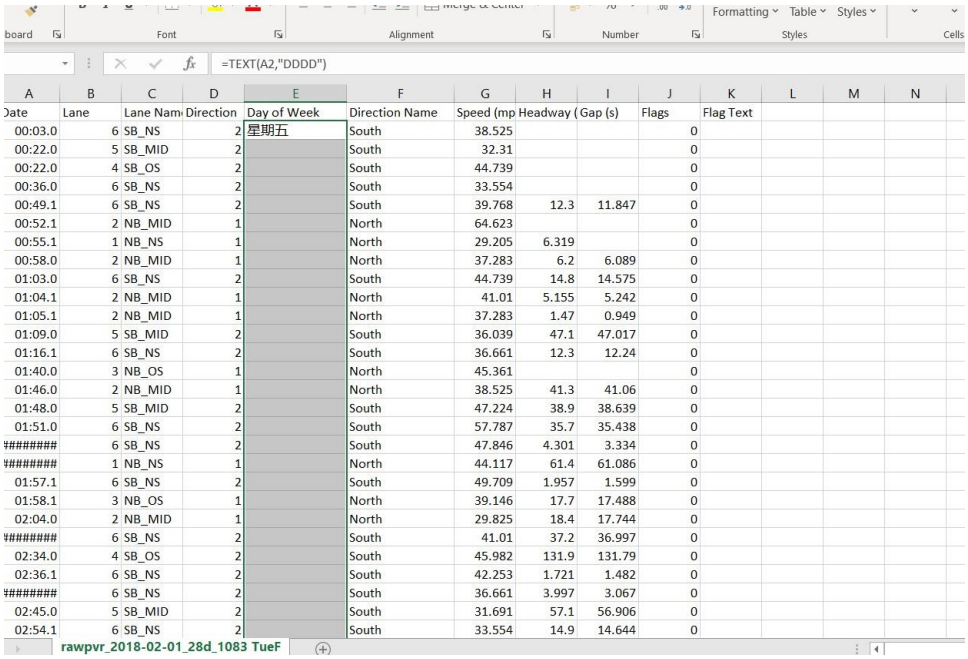


Figure 28. week of day 3

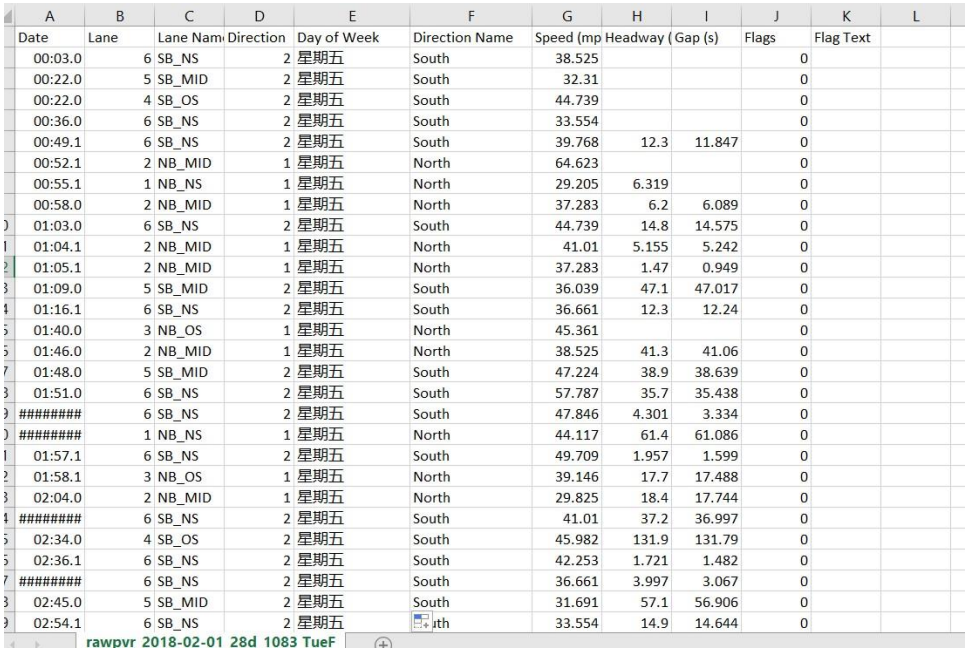
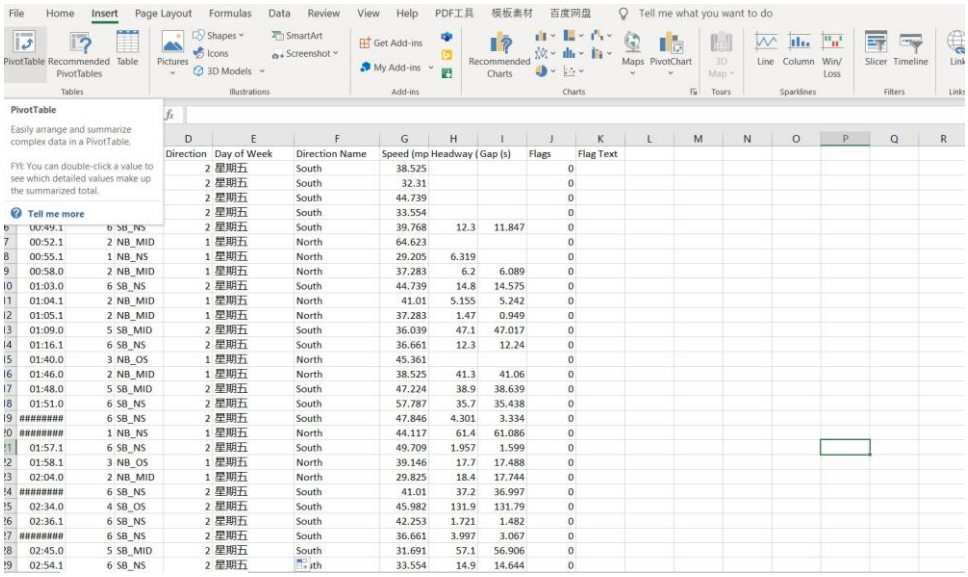


Figure 27. week of day 2

Step 2: Create pivotTable and choose field

The pivotTable is a helpful function in excel. It is able to support filter data with multiple criteria. Therefore, the project uses it to implement filter.



	D	E	F	G	H	I	J	K
	Direction	Day of Week	Direction Name	Speed (mp	Headway (	Gap (s)	Flags	Flag Text
	2	星期五	South	38.525			0	
	2	星期五	South	32.31			0	
	2	星期五	South	44.739			0	
	2	星期五	South	33.554			0	
6	00:49.1	6 SB_NS	2	星期五	South	39.768	12.3	11.847	0	
7	00:52.1	2 NB_MID	1	星期五	North	64.623			0	
8	00:55.1	1 NB_NS	1	星期五	North	29.205	6.319		0	
9	00:58.0	2 NB_MID	1	星期五	North	37.283	6.2	6.089	0	
10	01:03.0	6 SB_NS	2	星期五	South	44.739	14.8	14.575	0	
11	01:04.1	2 NB_MID	1	星期五	North	41.01	5.155	5.242	0	
12	01:05.1	2 NB_MID	1	星期五	North	37.283	1.47	0.949	0	
13	01:09.0	5 SB_MID	2	星期五	South	36.039	47.1	47.017	0	
14	01:16.1	6 SB_NS	2	星期五	South	36.661	12.3	12.24	0	
15	01:40.0	3 NB_OS	1	星期五	North	45.361			0	
16	01:46.0	2 NB_MID	1	星期五	North	38.525	41.3	41.06	0	
17	01:48.0	5 SB_MID	2	星期五	South	47.224	38.9	38.639	0	
18	01:51.0	6 SB_NS	2	星期五	South	57.787	35.7	35.438	0	
19	########	6 SB_NS	2	星期五	South	47.846	4.301	3.334	0	
20	########	1 NB_NS	1	星期五	North	44.117	61.4	61.086	0	
21	01:57.1	6 SB_NS	2	星期五	South	49.709	1.957	1.599	0	
22	01:58.1	3 NB_OS	1	星期五	North	39.146	17.7	17.488	0	
23	02:04.0	2 NB_MID	1	星期五	North	29.825	18.4	17.744	0	
24	########	6 SB_NS	2	星期五	South	41.01	37.2	36.997	0	
25	02:34.0	4 SB_OS	2	星期五	South	45.982	131.9	131.79	0	
26	02:36.1	6 SB_NS	2	星期五	South	42.253	1.721	1.482	0	
27	########	6 SB_NS	2	星期五	South	36.661	3.997	3.067	0	
28	02:45.0	5 SB_MID	2	星期五	South	31.691	57.1	56.906	0	
29	02:54.1	6 SB_NS	2	星期五	South	33.554	14.9	14.644	0	

Figure 30. pivotTable

Firstly, Add date to rows area.

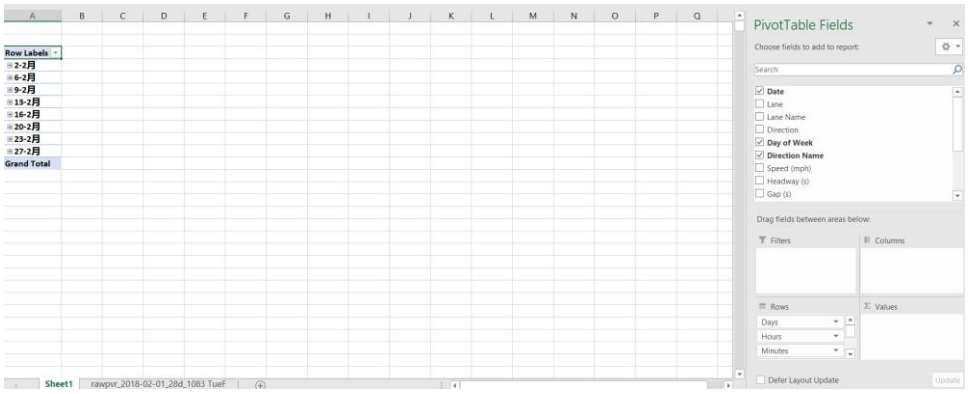


Figure 31. pivotTable add rows

Secondly, consider the requirement of the task:" Choose a day of the week, e.g., Tuesday ". I have chosen Tuesday for this task. Therefore, the "Day of Week" would be the first criteria. At the same time, the task requires generating a separate bar plot for each traffic direction (North and South). It makes "Direction Name" to be another condition in the Filters area. Thirdly, it asks "use bar plots to visualise the average traffic volume for each hour of the day". Therefore, the "Hours" should be added to "Columns" area to show traffic volume by each hour in table.

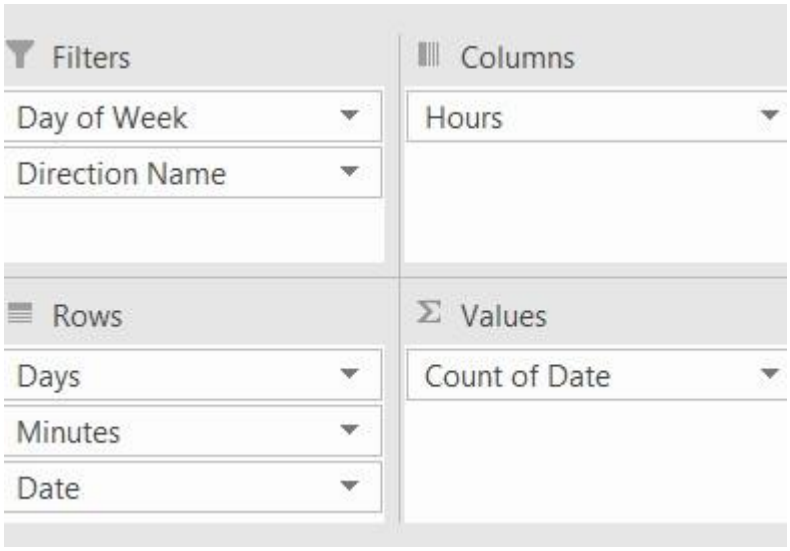


Figure 32. pivotTable add values,columns and filters

Step 3: Select filters value.

Due to I choose “Tuesday” for this task. The “星期二” is the Tuesday in Chinese.

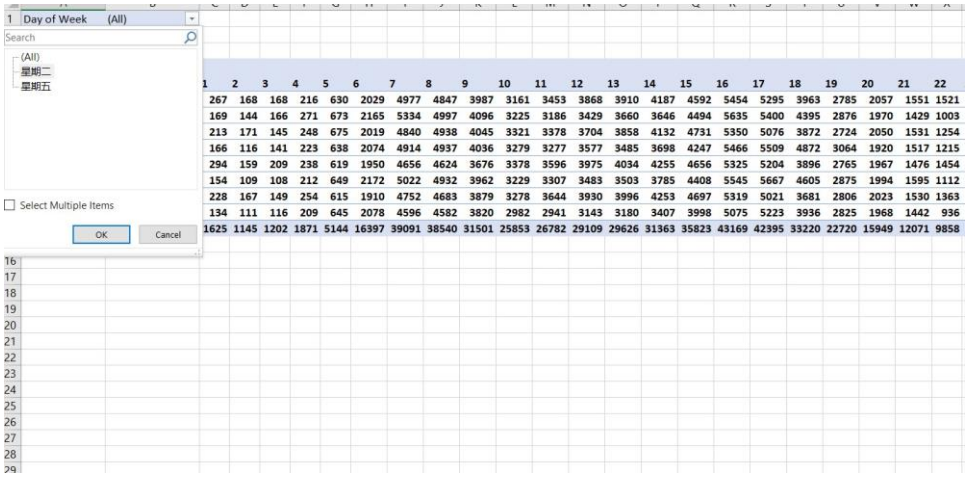


Figure 33. pivotTable filter week of day

Choose “South” for this time.

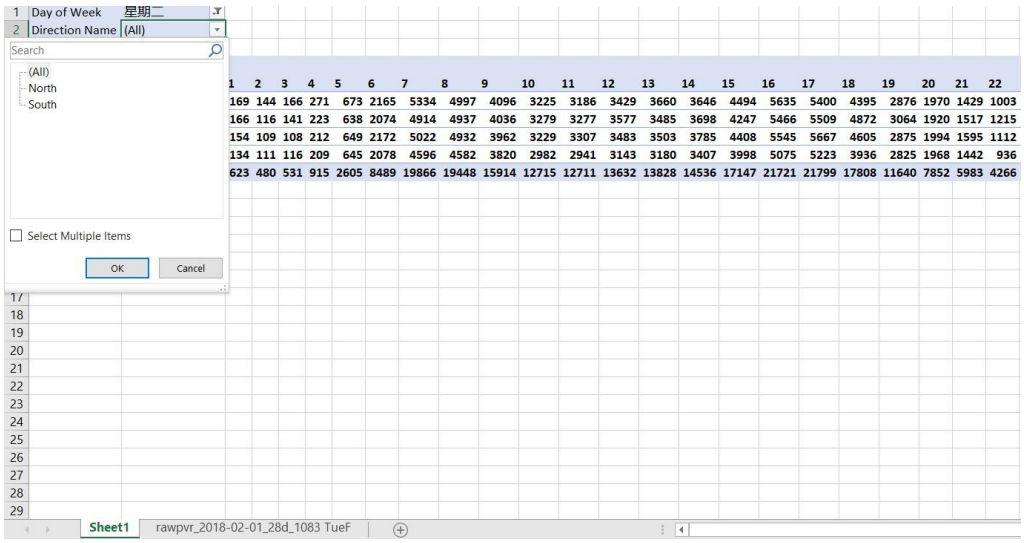


Figure 34. pivotTable filter direction name

Then we got this table, it can show the traffic volume in each hour for these 4 days.

Count of Date	Column Labels																							
Row Labels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Grand Total
6-2月	197	118	96	106	152	292	897	2145	2194	1559	1425	1509	1677	1720	1925	2705	3210	2757	2406	1603	1103	783	573	313	31465
13-2月	160	95	74	80	119	279	813	1991	2130	1589	1430	1583	1774	1807	1999	2390	2982	2756	2473	1741	1064	858	720	365	31272
20-2月	163	80	73	59	107	291	886	1992	2190	1651	1427	1632	1807	1802	2137	2487	3003	2775	2180	1560	1072	946	647	345	31312
27-2月	131	78	67	74	104	297	853	1932	2026	1394	1326	1450	1579	1669	1907	2344	2810	2888	2263	1503	1082	801	511	301	29390
Grand Total	651	371	310	319	482	1159	3449	8060	8540	6193	5608	6174	6837	6998	7968	9926	12005	11176	9322	6407	4321	3388	2451	1324	123439

Figure 35. pivotTable filter direction name

Step 4: Draw barplots in excel

Due to task asked, “Each bar plot should show the average traffic volume for each hour of the day.” Therefore, as this figure shown below, it counts the average traffic volume for each hour for the 4 provided days. (B10 is the Grand Total for hour 0 in last figure)

Each hour	The average number of traffic
00:00 - 00:59	=B10/4
01:00 - 01:59	92.75
02:00 - 02:59	77.5
03:00 - 03:59	79.75
04:00 - 04:59	120.5
05:00 - 05:59	289.75
06:00 - 06:59	862.25
07:00 - 07:59	2015
08:00 - 08:59	2135
09:00 - 09:59	1548.25
10:00 - 10:59	1402
11:00 - 11:59	1543.5
12:00 - 12:59	1709.25
13:00 - 13:59	1749.5
14:00 - 14:59	1992
15:00 - 15:59	2481.5
16:00 - 16:59	3001.25
17:00 - 17:59	2794
18:00 - 18:59	2330.5
19:00 - 19:59	1601.75
20:00 - 20:59	1080.25
21:00 - 21:59	847
22:00 - 22:59	612.75
23:00 - 23:59	331

Figure 36. Average

After the new table has been created and fill in. Select “Insert” and “Insert Columns or Bar Chart”.

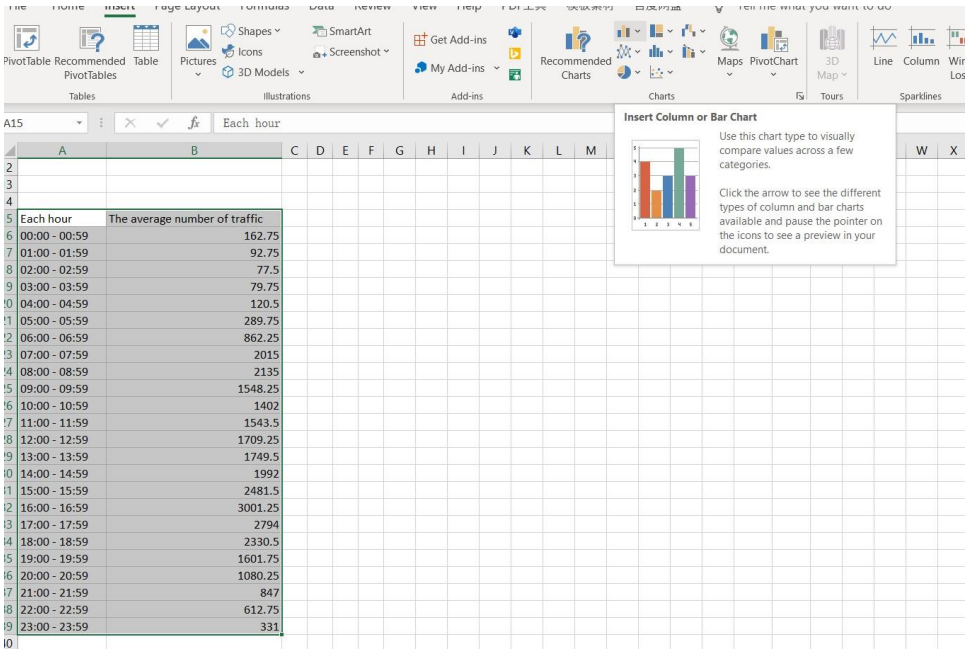


Figure 37. Drawing graph

In this case, the most important point is to show how traffic volume changes over time. Therefore, “Stacked Column” would be the most suitable choice.

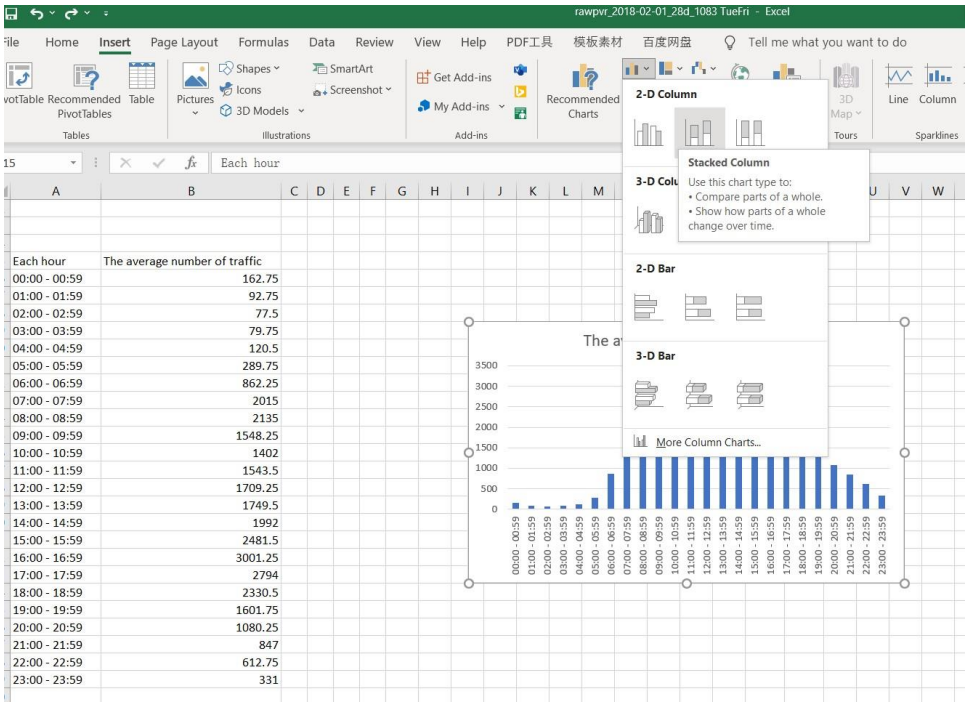


Figure 38. Select columns

Here is the result for South direction:

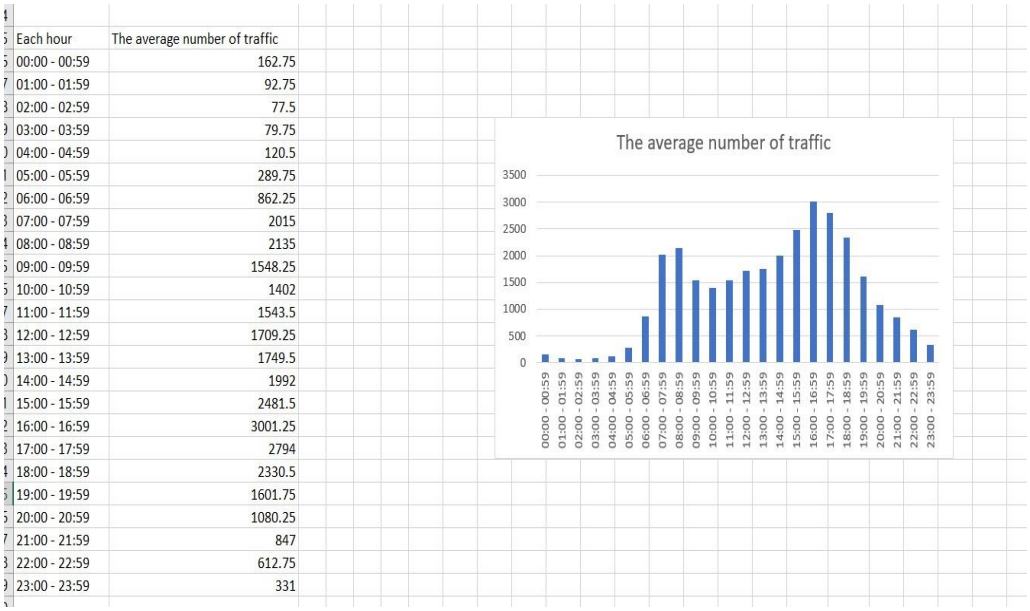


Figure 39. Task 3 south result

Step-by-step description of the development of the task:(North)

The step of develop north direction data is extremely similar with how project did in South data. The

only difference is select “North” in this “Direction Name” filter.

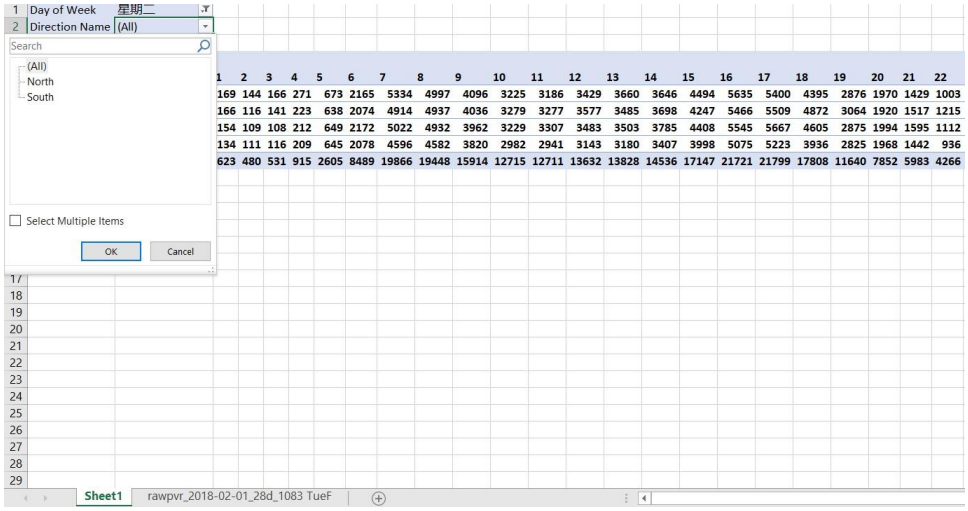
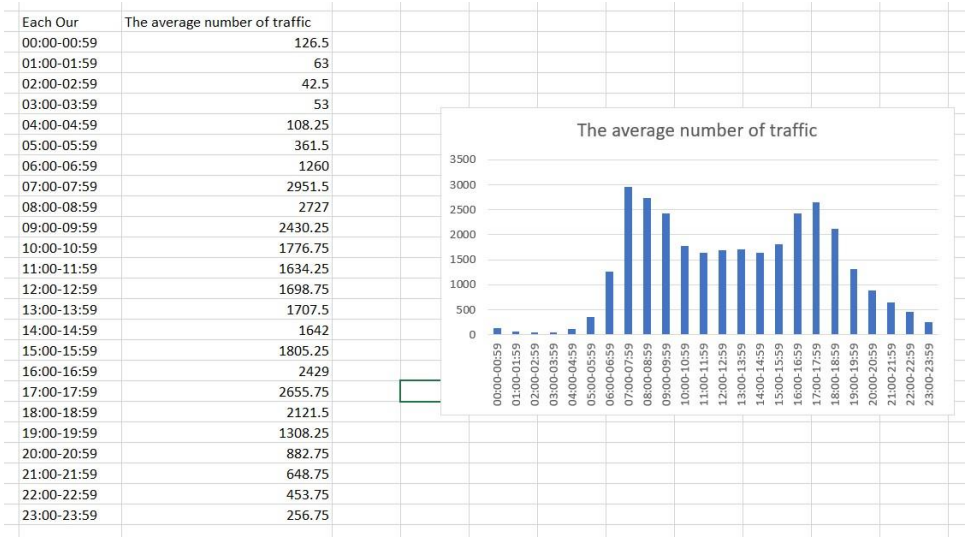


Figure 40.Task 4 north

Here is the output result



Each Our	The average number of traffic
00:00-00:59	126.5
01:00-01:59	63
02:00-02:59	42.5
03:00-03:59	53
04:00-04:59	108.25
05:00-05:59	361.5
06:00-06:59	1260
07:00-07:59	2951.5
08:00-08:59	2727
09:00-09:59	2430.25
10:00-10:59	1776.75
11:00-11:59	1634.25
12:00-12:59	1698.75
13:00-13:59	1707.5
14:00-14:59	1642
15:00-15:59	1805.25
16:00-16:59	2429
17:00-17:59	2655.75
18:00-18:59	2121.5
19:00-19:59	1308.25
20:00-20:59	882.75
21:00-21:59	648.75
22:00-22:59	453.75
23:00-23:59	256.75

Figure 41. Task 3 north result

Interpretation of the results:

From the result, the same point is for 6:00-9:59 and 16:00-18:59, these two time range has significantly large traffic numbers than other times. Normally, these two time range indicates people go to work and back home. Moreover, it is clear to see that from the north direction, the highest traffic volume time are: 7:00-7:59, 8:00-8:59, and 17:00-17:59. For south direction, they are 16:00-16:59, 17:00-17:59 and 15:00-15:59. Therefore, from these two bar plots, we could assume the north of the city has more working offices. South of the city has more Residential area. Another point is the traffic volume from 10:00-14:59. In the north direction, these data are extremely close. However, in the south direction, the traffic volume of 14:00-14:59 has significantly larger than other time range. It means in working hour, people are not likely to go to the north of the city.

Task 4:

I have chosen Python for this task4.

Output:

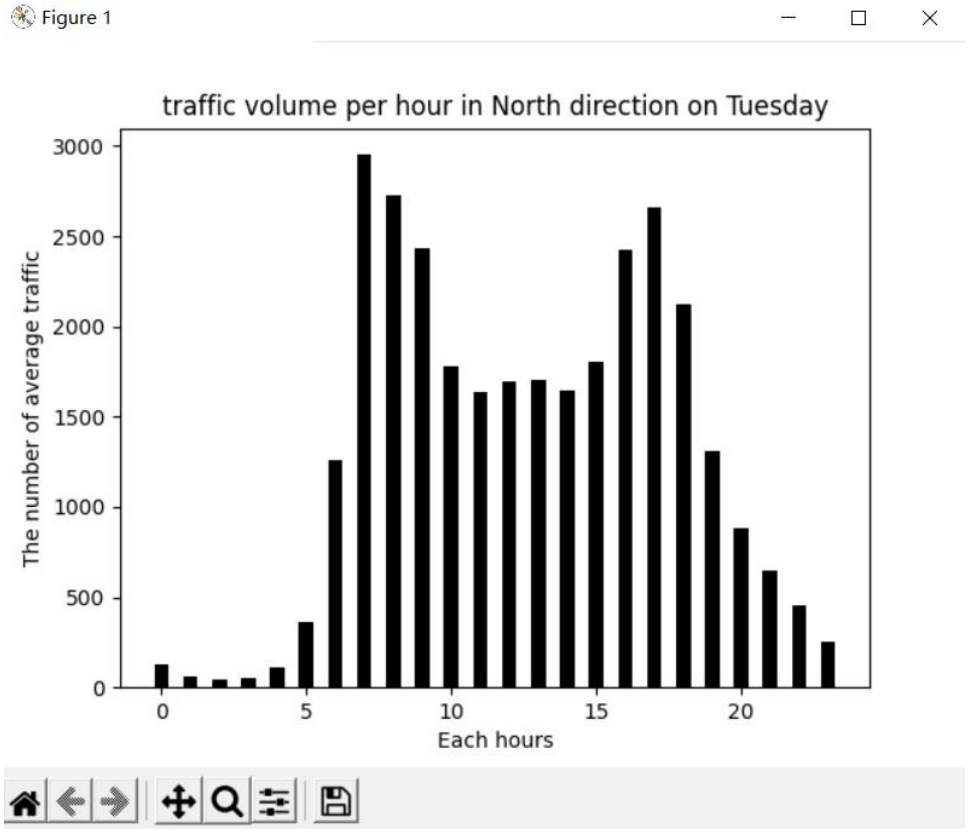


Figure 42. Task 4 north output

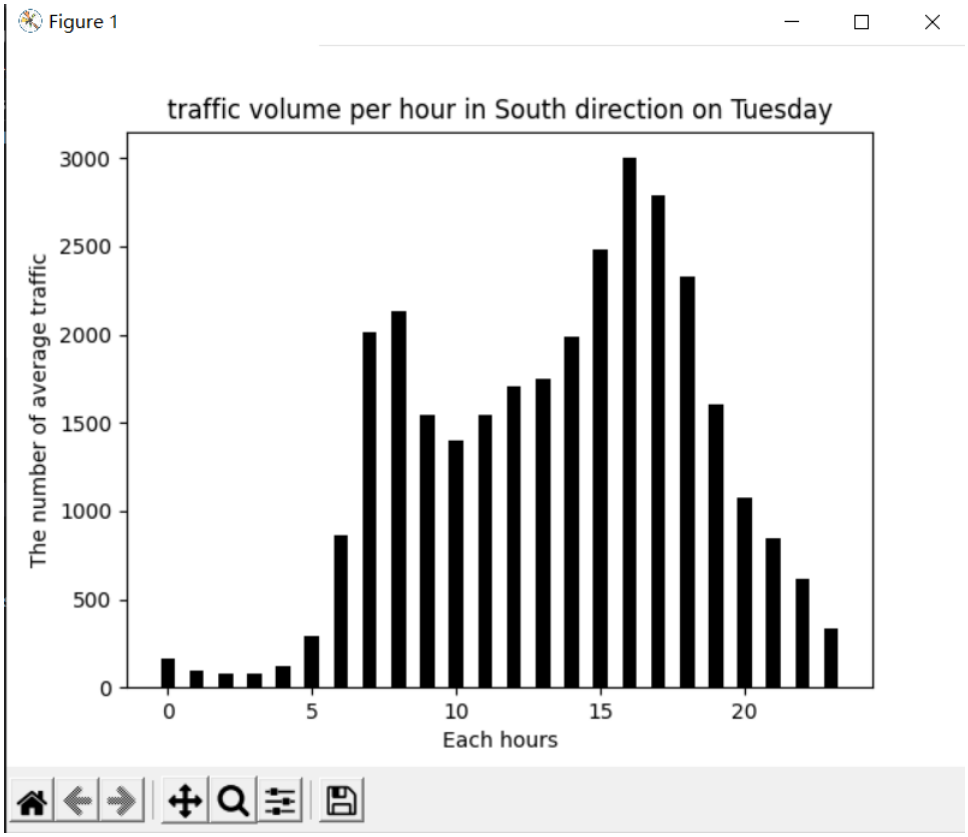


Figure 43. Task 4 south output

Step-by-step description of the development of the task:(North)

Step1. Import helpful package.

Python provides many packages with different functions. These packages help Python be easier to use and powerful. For this task, the provided data format is CSV. Pandas is a suitable approach to deal with CSV data files. It contains many helpful functions to select data with multiple conditions. Therefore, the project import pandas at first. At the same time, due to the output is bar plots. From some search, I have found that “matplotlib.pyplot” is a sensible approach to draw bar plots in python. From my experience, numpy is another powerful package. It can support complex calculations in few line codes. It may be helpful to the later step. Therefore, these packages have been imported.

Step2. Add one more column about day of week information for later filter.

```
1  import numpy as np
2  import datetime
3  import matplotlib.pyplot as plt
4  import pandas as pd
5  import dateutil.parser
6
7  list1 = pd.read_csv("rawpvr_2018-02-01_28d_1083 TueFri.csv")
8  list2 = pd.read_csv("rawpvr_2018-02-01_28d_1083 TueFri.csv")['Date']
9  Week_of_day =[]
10 for date in list2:
11     formed_date = dateutil.parser.parse(date)
12     Week_of_day.append(formed_date.weekday())
13 print(Week_of_day)
14
15 list1['week of day'] = Week_of_day
16
17 print(list1)
18
19 Tuesday_North = list1.loc[(list1['week of day'] == 1)&(list1['Direction Name']== "North")]
20 print(Tuesday_North)
```

Figure 44. Import and add column

In line 7, I have created list to store data frame from the provided data.

Similar with the task 2 and 3, the early step will add one more column about week of day information in csv data frame. Therefore, it creates a new list and select “Date” column only.

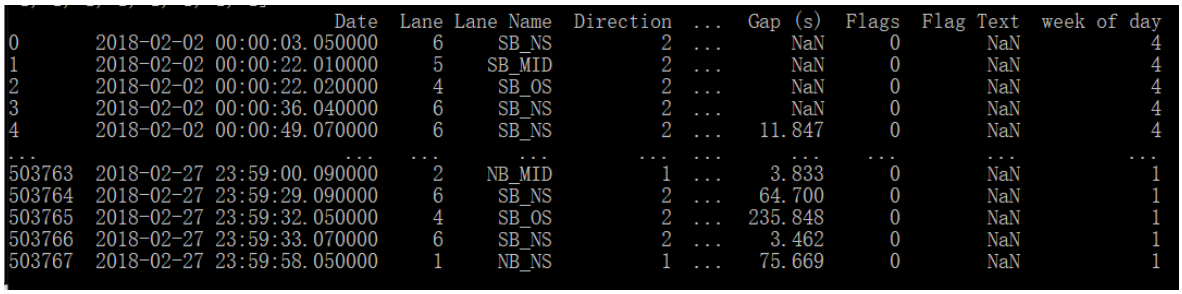
In the beginning, it has shown some problems with the type of column “Date”. The data from this column was identified as “str” instead of “date”. Therefore, it is not able to use the “weekday” function to get the day of the week for these data. At the same time, those data contain different format. For instance, the date from 65455 line followed yyyy-mm-dd h:m:s format. However, the date from 65456 line followed yyyy-mm-dd h:m:s.ms format. It has risen some difficulty at the early of work.

```
65455   2018-02-06 04:17:02,2,NB_MID,1,North,38.525,,,0,
65456   2018-02-06 04:17:21.060000,4,SB_OS,2,South,41.01,67.5,67.157,0,
```

Figure 45. Data difference

From some searches, dateutil.parser is a possible approach to solve this problem. It provides date string parser to convert string to date type. Therefore, I have tried to create a list. Then use for loop to iterate

every date from Date list. Parser date, convert date to week of day, and add to the list. Finally, in line 15, one more column called “week of day” has been created in the data frame. This column contains the week-of-day information, and it supports the filter function in a later step.



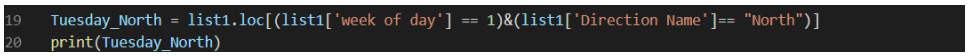
	Date	Lane	Lane Name	Direction	...	Gap (s)	Flags	Flag Text	week of day
0	2018-02-02 00:00:03.050000	6	SB_NS	2	...	NaN	0	NaN	4
1	2018-02-02 00:00:22.010000	5	SB_MID	2	...	NaN	0	NaN	4
2	2018-02-02 00:00:22.020000	4	SB_OS	2	...	NaN	0	NaN	4
3	2018-02-02 00:00:36.040000	6	SB_NS	2	...	NaN	0	NaN	4
4	2018-02-02 00:00:49.070000	6	SB_NS	2	...	11.847	0	NaN	4
...	...	...	...	...	...	...	...	...	...
503763	2018-02-27 23:59:00.090000	2	NB_MID	1	...	3.833	0	NaN	1
503764	2018-02-27 23:59:29.090000	6	SB_NS	2	...	64.700	0	NaN	1
503765	2018-02-27 23:59:32.050000	4	SB_OS	2	...	235.848	0	NaN	1
503766	2018-02-27 23:59:33.070000	6	SB_NS	2	...	3.462	0	NaN	1
503767	2018-02-27 23:59:58.050000	1	NB_NS	1	...	75.669	0	NaN	1

Figure 46. New data frame with week of day

Here is the current data framedataframen “weekday” function. Monday to Sunday is 0 to 6. Therefore, 1 means Tuesday and 4 means Friday.

Step3. Filter with direction name and week of day condition

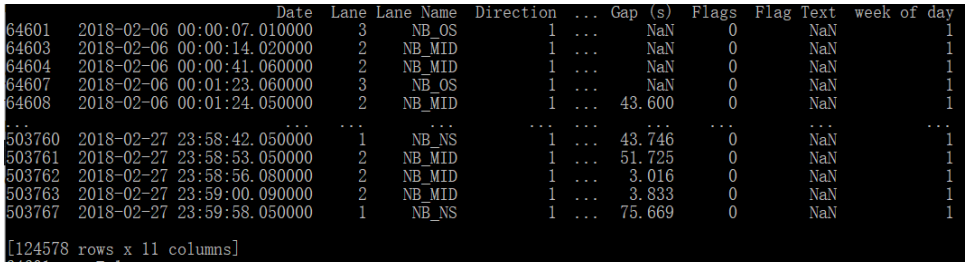
To filter with condition about the “direction name” and “week of day” condition. A new data frame has been created. It used loc function to select data that satisfied these two conditions. As we mentioned before, 1 is the Tuesday in python weekday function.



```
19  Tuesday_North = list1.loc[(list1['week of day'] == 1)&(list1['Direction Name']== "North")]
20  print(Tuesday_North)
```

Figure 47. Filter in python

Output of “Tuesday_North” data frame:

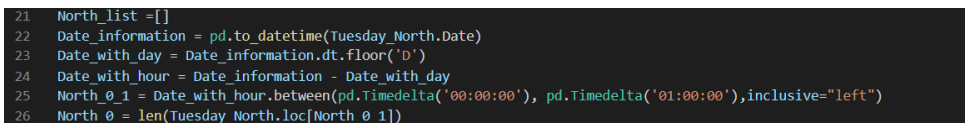


	Date	Lane	Lane Name	Direction	...	Gap (s)	Flags	Flag Text	week of day
64601	2018-02-06 00:00:07.010000	3	NB_OS	1	...	NaN	0	NaN	1
64603	2018-02-06 00:00:14.020000	2	NB_MID	1	...	NaN	0	NaN	1
64604	2018-02-06 00:00:41.060000	2	NB_MID	1	...	NaN	0	NaN	1
64607	2018-02-06 00:01:23.060000	3	NB_OS	1	...	NaN	0	NaN	1
64608	2018-02-06 00:01:24.050000	2	NB_MID	1	...	43.600	0	NaN	1
...	...	...	...	...	...	...	...	...	...
503760	2018-02-27 23:58:42.050000	1	NB_NS	1	...	43.746	0	NaN	1
503761	2018-02-27 23:58:53.050000	2	NB_MID	1	...	51.725	0	NaN	1
503762	2018-02-27 23:58:56.080000	2	NB_MID	1	...	3.016	0	NaN	1
503763	2018-02-27 23:59:00.090000	2	NB_MID	1	...	3.833	0	NaN	1
503767	2018-02-27 23:59:58.050000	1	NB_NS	1	...	75.669	0	NaN	1

[124578 rows x 11 columns]

Figure 48. Output with filter

Step4. Filter for each hour



```
21  North_list =[]
22  Date_information = pd.to_datetime(Tuesday_North.Date)
23  Date_with_day = Date_information.dt.floor('D')
24  Date_with_hour = Date_information - Date_with_day
25  North_0_1 = Date_with_hour.between(pd.Timedelta('00:00:00'), pd.Timedelta('01:00:00'),inclusive="left")
26  North_0 = len(Tuesday_North.loc[North_0_1])
```

Figure 49.Filter with hour code

Create one list called North_list, this list will be used later to draw bar plots. In line 22, we select date information from the Tuesday_North data frame. (line22)

```
64601    2018-02-06 00:00:07.010
64603    2018-02-06 00:00:14.020
64604    2018-02-06 00:00:41.060
64607    2018-02-06 00:01:23.060
64608    2018-02-06 00:01:24.050
                 ...
503760   2018-02-27 23:58:42.050
503761   2018-02-27 23:58:53.050
503762   2018-02-27 23:58:56.080
503763   2018-02-27 23:59:00.090
503767   2018-02-27 23:59:58.050
```

Figure 50.Output with north and date

Then use .dt.floor function to select the day information. (Line23)

```
64601    2018-02-06
64603    2018-02-06
64604    2018-02-06
64607    2018-02-06
64608    2018-02-06
            ...
503760   2018-02-27
503761   2018-02-27
503762   2018-02-27
503763   2018-02-27
503767   2018-02-27
Name: Date, Length: 124578, dtype: datetime64[ns]
```

Figure 51.Output day only

In line 24, use date_information-date_with_day to remove day informaiton from the date value.

```
64601    0 days 00:00:07.010000
64603    0 days 00:00:14.020000
64604    0 days 00:00:41.060000
64607    0 days 00:01:23.060000
64608    0 days 00:01:24.050000
                   ...
503760   0 days 23:58:42.050000
503761   0 days 23:58:53.050000
503762   0 days 23:58:56.080000
503763   0 days 23:59:00.090000
503767   0 days 23:59:58.050000
Name: Date, Length: 124578, dtype: timedelta64[ns]
```

Figure 52.Output remove the day

In line 25, using “between” function to get 0 to 1 data. For the Inclusive parameter, there are some experiments about how Inclusive parameter worked.

```
4   s = pd.Series([2, 0, 5, 9,7])
5   print(s.between(2,5))
6   print(s.between(2,5,inclusive="left"))
7   print(s.between(2,5,inclusive="right"))
```

Figure 53.Experiment code of inclusive

```
0     True
1    False
2     True
3    False
4    False
dtype: bool
0     True
1    False
2    False
3    False
4    False
dtype: bool
0    False
1    False
2     True
3    False
4    False
dtype: bool
```

Figure 54.Output of Experiment

In this case, we want to [00:00:00] counted as 0 hour , and [01:00:00] does not. Therefore, we use inclusive parameter to set as left.

Finally, line 26 use “len()” function to count the traffic volume from 0 to 1 in these 4 days.

Do the same thing for other 23 hours data.

```
27  North_1_2 = Date_with_hour.between(pd.Timedelta('01:00:00'), pd.Timedelta('02:00:00'),inclusive="left")
28  North_2_3 = Date_with_hour.between(pd.Timedelta('02:00:00'), pd.Timedelta('03:00:00'),inclusive="left")
29  North_3_4 = Date_with_hour.between(pd.Timedelta('03:00:00'), pd.Timedelta('04:00:00'),inclusive="left")
30  North_4_5 = Date_with_hour.between(pd.Timedelta('04:00:00'), pd.Timedelta('05:00:00'),inclusive="left")
31  North_5_6 = Date_with_hour.between(pd.Timedelta('05:00:00'), pd.Timedelta('06:00:00'),inclusive="left")
32  North_6_7 = Date_with_hour.between(pd.Timedelta('06:00:00'), pd.Timedelta('07:00:00'),inclusive="left")
33  North_7_8 = Date_with_hour.between(pd.Timedelta('07:00:00'), pd.Timedelta('08:00:00'),inclusive="left")
34  North_8_9 = Date_with_hour.between(pd.Timedelta('08:00:00'), pd.Timedelta('09:00:00'),inclusive="left")
35  North_9_10 = Date_with_hour.between(pd.Timedelta('09:00:00'), pd.Timedelta('10:00:00'),inclusive="left")
36  North_10_11 = Date_with_hour.between(pd.Timedelta('10:00:00'), pd.Timedelta('11:00:00'),inclusive="left")
37  North_11_12 = Date_with_hour.between(pd.Timedelta('11:00:00'), pd.Timedelta('12:00:00'),inclusive="left")
38  North_12_13 = Date_with_hour.between(pd.Timedelta('12:00:00'), pd.Timedelta('13:00:00'),inclusive="left")
39  North_13_14 = Date_with_hour.between(pd.Timedelta('13:00:00'), pd.Timedelta('14:00:00'),inclusive="left")
40  North_14_15 = Date_with_hour.between(pd.Timedelta('14:00:00'), pd.Timedelta('15:00:00'),inclusive="left")
41  North_15_16 = Date_with_hour.between(pd.Timedelta('15:00:00'), pd.Timedelta('16:00:00'),inclusive="left")
42  North_16_17 = Date_with_hour.between(pd.Timedelta('16:00:00'), pd.Timedelta('17:00:00'),inclusive="left")
43  North_17_18 = Date_with_hour.between(pd.Timedelta('17:00:00'), pd.Timedelta('18:00:00'),inclusive="left")
44  North_18_19 = Date_with_hour.between(pd.Timedelta('18:00:00'), pd.Timedelta('19:00:00'),inclusive="left")
45  North_19_20 = Date_with_hour.between(pd.Timedelta('19:00:00'), pd.Timedelta('20:00:00'),inclusive="left")
46  North_20_21 = Date_with_hour.between(pd.Timedelta('20:00:00'), pd.Timedelta('21:00:00'),inclusive="left")
47  North_21_22 = Date_with_hour.between(pd.Timedelta('21:00:00'), pd.Timedelta('22:00:00'),inclusive="left")
48  North_22_23 = Date_with_hour.between(pd.Timedelta('22:00:00'), pd.Timedelta('23:00:00'),inclusive="left")
49  North_23_0 = Date_with_hour.between(pd.Timedelta('23:00:00'), pd.Timedelta('24:00:00'),inclusive="left")
```

Figure 55.Code for hour filter with other 23hours

Using len() to count these values and add to list.

```
North_2 = len(Tuesday_North.loc[North_2_3])

North_3 = len(Tuesday_North.loc[North_3_4])
North_4 = len(Tuesday_North.loc[North_4_5])
North_5 = len(Tuesday_North.loc[North_5_6])
North_6 = len(Tuesday_North.loc[North_6_7])
North_7 = len(Tuesday_North.loc[North_7_8])
North_8 = len(Tuesday_North.loc[North_8_9])
North_9 = len(Tuesday_North.loc[North_9_10])
North_10 = len(Tuesday_North.loc[North_10_11])
North_11 = len(Tuesday_North.loc[North_11_12])
North_12 = len(Tuesday_North.loc[North_12_13])
North_13 = len(Tuesday_North.loc[North_13_14])
North_14 = len(Tuesday_North.loc[North_14_15])
North_15 = len(Tuesday_North.loc[North_15_16])
North_16 = len(Tuesday_North.loc[North_16_17])
North_17 = len(Tuesday_North.loc[North_17_18])
North_18 = len(Tuesday_North.loc[North_18_19])
North_19 = len(Tuesday_North.loc[North_19_20])
North_20 = len(Tuesday_North.loc[North_20_21])
North_21 = len(Tuesday_North.loc[North_21_22])
North_22 = len(Tuesday_North.loc[North_22_23])
North_23 = len(Tuesday_North.loc[North_23_0])
North_list.append(North_0)
North_list.append(North_1)
North_list.append(North_2)
North_list.append(North_3)
North_list.append(North_4)
North_list.append(North_5)
North_list.append(North_6)
North_list.append(North_7)
North_list.append(North_8)
```

Figure 56. add to list

```
average_traffic_volume=[]
for data in North_list:
    average_traffic_volume.append(data/4)
print(average_traffic_volume)
time_list =[]
time_helper = 0
while time_helper<24:
    time_list.append(time_helper)
    time_helper +=1
print(time_list)
```

Figure 57. Preparing data for bar plots

Iterate all values from North_list to get the average values. Then create another list which contains hours information. These two lists are:

```
[126.5, 63.0, 42.5, 53.0, 108.25, 361.5, 1260.0, 2951.5, 2727.0, 2430.25, 1776.75, 1634.25, 1698.75, 1707.5, 1642.0, 180
5.25, 2429.0, 2655.75, 2121.5, 1308.25, 882.75, 648.75, 453.75, 256.75]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]
```

Figure 58. Output for Preparing data

Finally, draw bar plots with these two list as the result.

```
112   plt.xlabel("Each hours")
113   plt.ylabel("The number of average traffic")
114   plt.title("traffic volume per hour in North direction on Tuesday")
115   plt.show()
116
```

Figure 59.Code for drawing

Output:

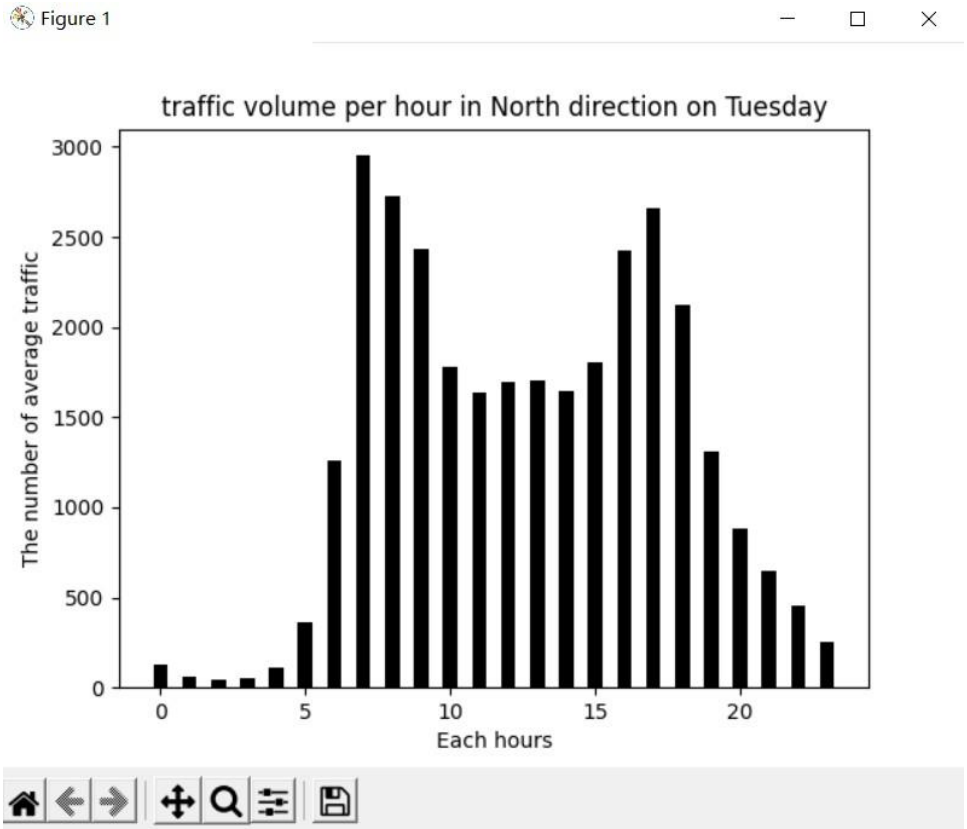


Figure 60. Output for task3 north

Step-by-step description of the development of the

task:(South)

1. Similar to task 3, the steps to collect south direction data are similar to the north direction. There are three different things, Firstly, the name of the variable needs to change to increase readability. Secondly, in step3: filter with direction name part. It needs to change “North” to “South” as the figure below. Thirdly, the title of bar plots should be changed.

```
21  Tuesday_South = list1.loc[(list1['week of day'] == 1)&(list1['Direction Name']== "South")]
```

```
113  plt.bar(time_list, average_traffic_volume, color ='black',width = 0.5)
114
115  plt.xlabel("Each hours")
116  plt.ylabel("The number of average traffic")
117  plt.title("traffic volume per hour in South direction on Tuesday")
118  plt.show()
119
```

Figure 61. Code changing for south direction

Output:

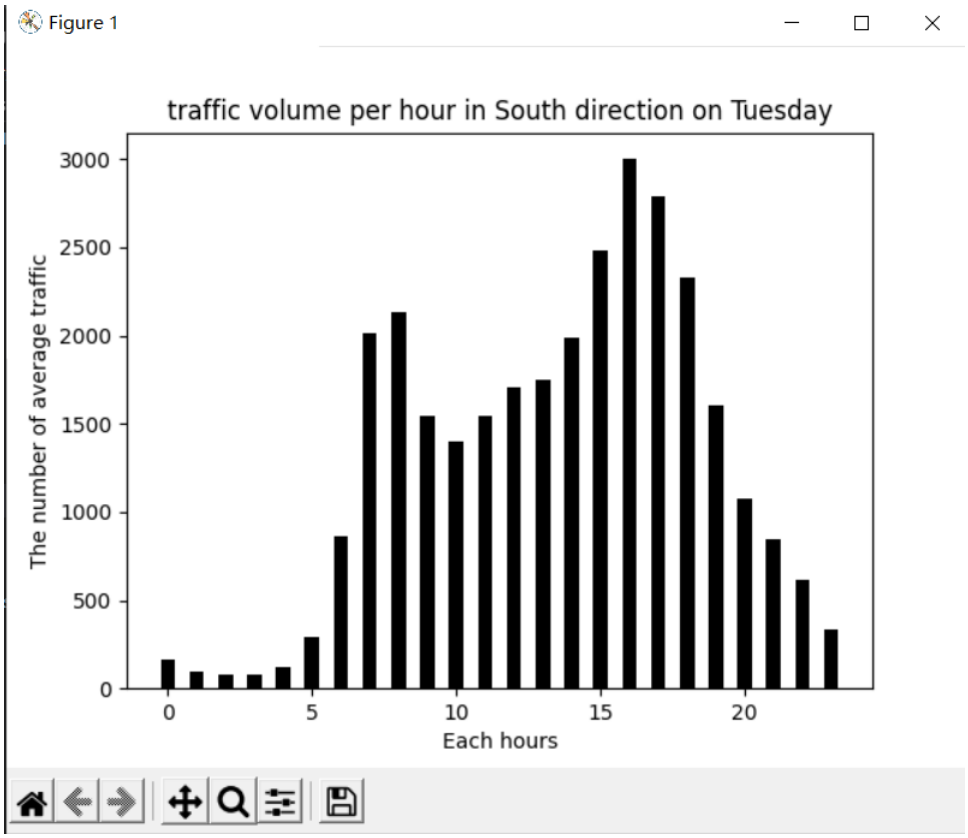


Figure 62. Output for task3 south

Two technologies used to develop this task:

Both technologies are easy to use and powerful. These two technologies are allowed to convert date

information to the week of day. It helps users solve data about the "week of day" problem more effectively and accurately. Moreover, both technologies can filter data with multiple criteria.

The first difference between the two technologies is the coding ability requirement. The user interface of excel is clear and straightforward. The pivot Table function is useful to solve multiple conditions for data. Even for those people who did not work with coding before, they could still deal with data analysis problems quickly and effectively by Excel. People will only use some easy formulas to get values instead of coding. For Python, as we mentioned earlier, Python has many helpful third-party packages. These packages make python could solve data problems easier and effectively than most programming languages. However, Python requires users familiar with coding. For data problems, it requires the user has some basic data structure knowledge. These points make Excel is a more sensible approach for those new users.

Another advantage of the effect of coding ability. Due to my coding ability limitation, I have used some for loop to iterate elements. It makes the program slowly and it spent 40-50 seconds for running. However, most people believe that Python has faster running time than Excel in large sizes of data [1].

Compare with Excel, there is also some significant advantages for python. Firstly, python could read more data types than excel [2]. At the same time, the error reporting is more detailed and easier to understand. Therefore, users could find and fix their error more effectively [2].

In conclusion, both two technologies are reasonable approaches to solve data filters. For people who are skilled with coding and familiar with the data structure. Python is the better choice on the running time side. At the same time, the powerful third-party package makes the limitation of python is not a serious problem. Excel is easy to use and has no requirement for coding. However, the limitation of Excel is a more serious problem. It is not an ideal choice to deal with a more complex model [4]. Therefore, choose these two technologies will more up to the user itself and the requirements.

Reference:

[1] February 20, 2018. [Online]. Available at https://www.wincalendar.com/Calendar/Date/February-20-2018 .[Accessed 14/10/2021]

[2] Excel vs Python: How to Do Common Data Analysis Tasks. [Online]. Available at https://www.dataquest.io/blog/excel-vs-python/ [Accessed 14/10/2021]

[3] Suen,D. A Look at Python Versus Excel. [Online] Available at https://www.gapintelligence.com/blog/a-look-at-python-versus-excel/ [Accessed: 14/10/2021]

[4] Frederick, M. Reasons You Should Stop Using Excel For Financial Data Analysis. [Online] Available at https://ecapitaladvisors.com/blog/stop-using-excel-financial-data-analysis/ [Accessed 14/10/2021]