Task 2

First, analyze the task. It is required to find the traffic volume of the North lanes at 9 am on Tuesday.

There should be 12 numbers in the resulting traffic volume. Because we have four Tuesdays in February 2018 (to be found later), and three lanes are pointing to the north. The volume can be divided into 12 groups, and 12 numbers are obtained.

However, the teacher said there should be 4 numbers(Only group by Days). Therefore, I wrote the process of two answers.

The label 'Direction,' 'Flags,' 'Lanes', and 'Hours' (used to record the hours of a day) can accomplish this task.

The label 'Flags' has been gotten in task 1, which identifies the day of the week. Besides, the 'DataFrame' in pandas can be used to process the data.

- **The first step** is creating the label we needed.

  Creating a new label called Hours to extract the data at 9 am. Besides, to distinguish which day of February the data came from, I created the label 'Day_of _the_month.'

  For convenience, the library 'datetime' is used to deal with the label 'Date.'

```
dataset['Hours'] = dataset['Date'].apply(lambda s :
datetime.strptime(s[:19], "%Y-%m-%d %H:%M:%S").hour)
dataset['Day_of_the_month'] = dataset['Date'].apply(lambda s :
datetime.strptime(s[:19], "%Y-%m-%d %H:%M:%S").day)
```

  'Hours' refers to the hours of a day, and 'Day_of_the_month' refers to February's date. For example, in data '2018-02-09 10:01:58.050000', 'Hours' is 0, and 'Day_of_the_month' is 9.

  The function head() can be used to show it.

```
In [14]: dataset.head()
```

Out[14]:

	Date	Lane	Lane Name	Direction	Direction Name	Speed (mph)	Headway (s)	Gap (s)	Flags	Flag Text	Hours	Day_of_the_month
0	2018-02-02 00:00:03.050000	6	SB_NS	2	South	38.525	NaN	NaN	5	Friday	0	2
1	2018-02-02 00:00:22.010000	5	SB_MID	2	South	32.310	NaN	NaN	5	Friday	0	2
2	2018-02-02 00:00:22.020000	4	SB_OS	2	South	44.739	NaN	NaN	5	Friday	0	2
3	2018-02-02 00:00:36.040000	6	SB_NS	2	South	33.554	NaN	NaN	5	Friday	0	2
4	2018-02-02 00:00:49.070000	6	SB_NS	2	South	39.768	12.3	11.847	5	Friday	0	2

- **The second step** is calculating the volume.

  In pandas, it is easy to access a group of rows and columns using the function loc, especially under some special conditions, and the function groupby() can be used to group data and compute operations on these groups.

  If the data is divided into 12 groups, then the code is as follows:

```
tmp = dataset.loc[dataset["Flags"] == 2] # Get data on Tuesday
tmp = tmp.loc[dataset['Hours'] == 9] # Get data on Tuesday at 9 am
tmp = tmp.loc[dataset['Direction'] == 1] # Get data on Tuesday at 9 am, and
the direction is north
traffic_volume_for_Tue = tmp.groupby(['Day_of_the_month','Lane']).size()
```

```
In [16]: traffic_volume_for_Tue

Out[16]: Day_of_the_month  Lane
         6                 1       743
                           2       879
                           3       915
         13                1       710
                           2       856
                           3       881
         20                1       682
                           2       823
                           3       806
         27                1       780
                           2       831
                           3       815
         dtype: int64
```

If the data is divided into 4 groups, then the code is as follows:

```
tmp = dataset.loc[dataset["Flags"] == 2]
tmp = tmp.loc[dataset['Hours'] == 9]
tmp = tmp.loc[dataset['Direction'] == 1]
traffic_volume_for_Tue = tmp.groupby(['Day_of_the_month']).size()
```

```
In [16]: traffic_volume_for_Tue

Out[16]: Day_of_the_month
         6     2537
         13    2447
         20    2311
         27    2426
         dtype: int64
```

The variable traffic_volume_for_Tue stands for the North lanes' traffic volume on Tuesday at 9 am and use the function size() to obtain the traffic volume through the number of eligible rows.

The data can be grouped by 'Day_of_the_month' and 'Lane.'

- **The third step** is getting Range, 1st Quartile, 2nd Quartile, 3rd Quartile, Interquartile range.

  There are 2 ways to get them.

  The first one is to use dataframe.describe()

  If the data is divided into 12 groups:

```
traffic_volume_for_Tue.describe()
```

```
Out[18]: count     12.000000
         mean     810.083333
         std       70.946149
         min      682.000000
         25%      770.750000
         50%      819.000000
         75%      861.750000
         max      915.000000
         Name: volume, dtype: float64
```

The 1st Quartile, 2nd Quartile, 3rd Quartile have been gotten. Then, calculate the Range by using the maximal value - minimal value 915 - 682 = 233. We also can use the 3rd Quartile - 1st Quartile to get the Interquartile range 861.75 - 770.75 = 91.00. We can subtract them directly since we already got them.

The second one is to use the function quantile() to get the quantiles.

```
volume_range = max(traffic_volume_for_Tue) - min(traffic_volume_for_Tue)
interquartile_range = traffic_volume_for_Tue.quantile(0.75) -
traffic_volume_for_Tue.quantile(0.25)
```

If the data is divided into 4 groups:

```
traffic_volume_for_Tue
```

```
count       4.000000
mean     2430.250000
std        92.942186
min      2311.000000
25%      2397.250000
50%      2436.500000
75%      2469.500000
max      2537.000000
Name: volume, dtype: float64
```

In the same way, we can get the Range 226, and the Interquartile range 72.25.

- **The final result** :
  - **If the data is divided into 12 groups:**

    range = 233.00

    1st Quartile = 770.75

    2nd Quartile = 819.00

    3rd Quartile = 861.75

    Interquartile range = 91.00
  - **If the data is divided into 4 groups:**

    range = 226.00

    1st Quartile = 2397.25

    2nd Quartile = 2436.50

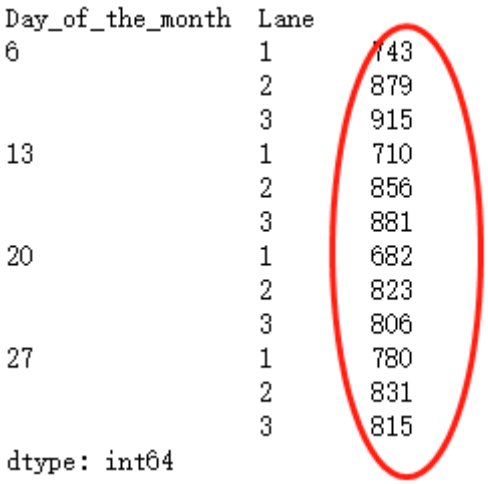
    3rd Quartile = 2469.50

    Interquartile range = 72.25
  - **Interpretation**:

    The Range of a data set could measure the variability, and it is very sensitive to the smallest and largest value.

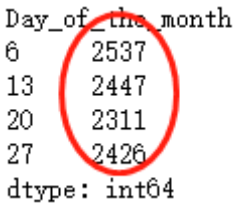
    The Interquartile range of a data set can also measure the variability, which is middle 50% of the data. Compared to the Range, it overcomes the sensitivity to extreme data values.

    If the data is divided into 12 groups:



The data we got is, as shown above. We can see that the traffic volume on the 6th in lane 3 is the largest, while the traffic volume on the 20th in lane 1 is the largest. The Range is 233, which can be said that the data fluctuates no very large. Besides, the Interquartile range is 91, 91 * 2 = 182 and the range is 233. These two numbers are relatively close. Therefore, maybe there are not many extreme values about the traffic volume of the North lanes at 9 am on Tuesday.

If the data is divided into 4 groups:



The data we got is, as shown above. We can see that the traffic volume on the 6th is the largest, while the traffic volume on the 20th is the largest. We can also see that the Range is 226, which can be said that the data fluctuates no very large. Besides, the Interquartile range is 72.25, 72.25 * 2 = 144.5 and the range is 226. These two numbers are relatively close. Therefore, maybe there are not many extreme values about the traffic volume of the North lanes at 9 am on Tuesday.

Task 3

I pick up Tuesday and visualize the average traffic volume for each hour on that day.

- **The first step** is calculating the traffic volume of the north lane and the south lane.

```
north_volume_per_hour = dataset.loc[dataset['Flags'] ==
2].loc[dataset['Direction'] == 1].groupby(['Hours']).size()
north_volume_per_hour /= 4
south_volume_per_hour = dataset.loc[dataset['Flags'] ==
2].loc[dataset['Direction'] == 2].groupby(['Hours']).size()
south_volume_per_hour /= 4
```

We get the number of all the rows on Tuesday and group by hours (North and South) and divide 4 to get the average.

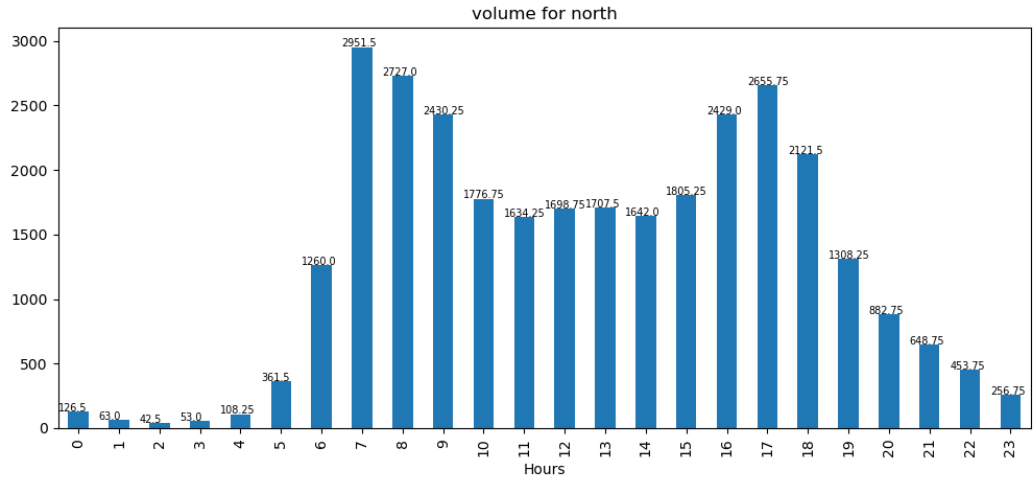
- **The second step** is using matplotlib.pyplot to visualize.

matplotlib is a great module for visualization. The library seaborn can be used to achieve visualization too.

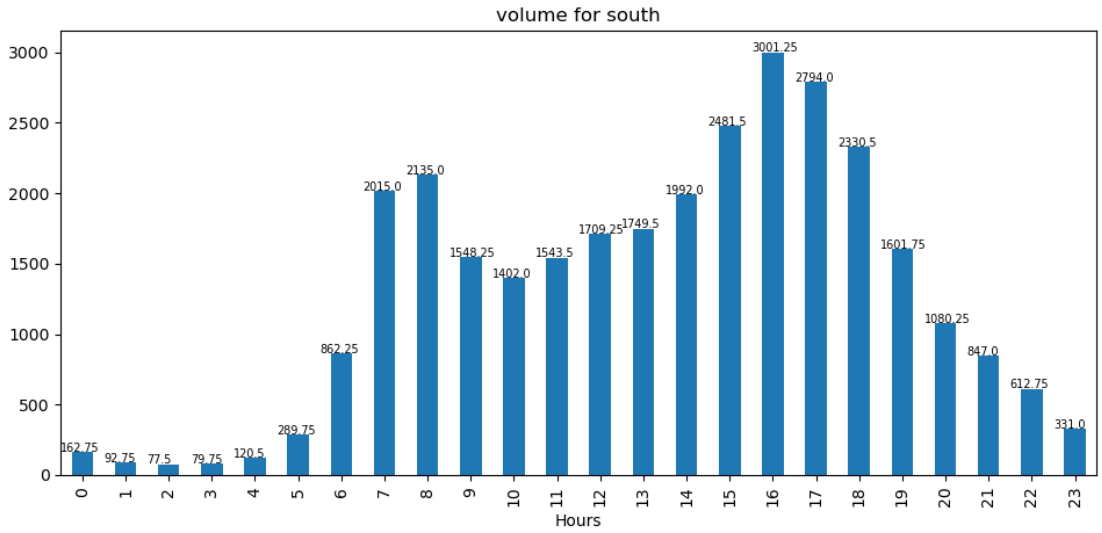
```
import matplotlib.pyplot as plt
%matplotlib notebook
north_volume_per_hour.plot.bar(x="Hours", title="volume for north")
```

Then, we could use matplotlib.pyplot.text() to add the average number of charts.

```
for i, v in enumerate(north_volume_per_hour):
    plt.text(i-0.5, v + 1, str(v), fontsize=7)
plt.tight_layout()
```



and use the same way to get the south



- **Interpretation**:

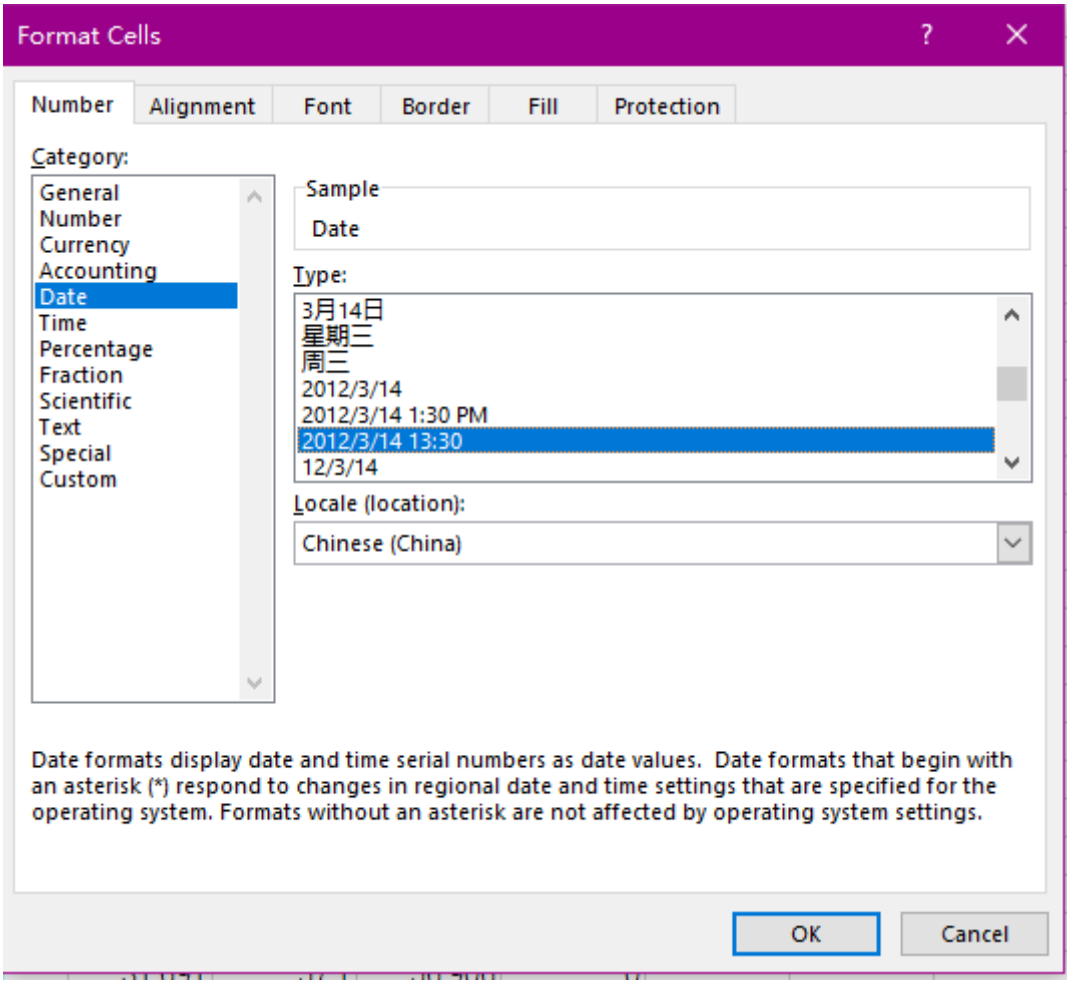
  From the above two graphs, we can see the average traffic volume for each hour of Tuesday in the south is similar to the direction in the north, and the data at 7 - 9 o'clock and 15 - 17 o'clock are relatively high, while at 0 - 5 o'clock and 22 o'clock, 23 o'clock are relatively low. Besides, in the north, the data peak is 2951.5, which is at 7 o'clock, while in the south the data peak is 3001.25, which is at 16 o'clock. Meanwhile, both in the south and the north, the data are low in February.

Task 4

I choose Excel as a GUI tool to do visualization and still choose Tuesday. The reason why I choose Excel is the amount of this dataset is not very large. If the data set is very large, just choose another GUI tool (Think about NHS news.)

- **The first step** deals with the label 'Flags.' Change the cell format to date is required.

1. Select the 'Date' column, click the right button and select the Format cell.

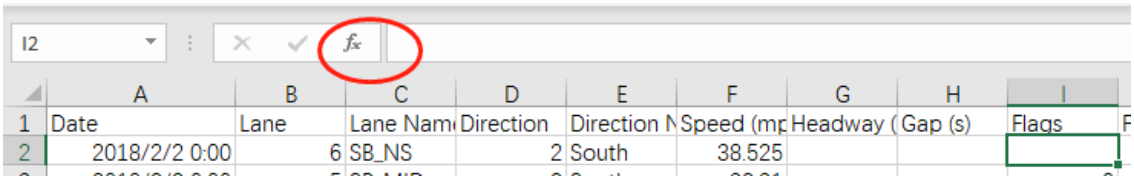


2. Choose the format shown above.

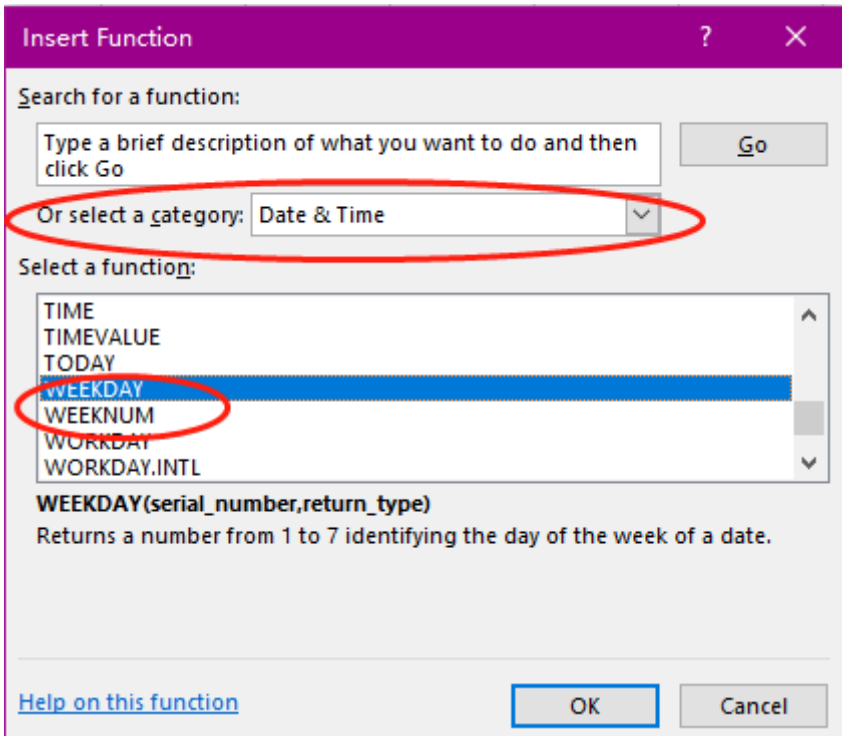
Date	Lane	Lane Nam	Direction	Direction N	Speed (mp	Headway (	Gap (s)	Flags	Flag Text
2018/2/2 0:00	6	SB_NS	2	South	38.525			0	
2018/2/2 0:00	5	SB_MID	2	South	32.31			0	
2018/2/2 0:00	4	SB_OS	2	South	44.739			0	
2018/2/2 0:00	6	SB_NS	2	South	33.554			0	
2018/2/2 0:00	6	SB_NS	2	South	39.768	12.3	11.847	0	
2018/2/2 0:00	2	NB_MID	1	North	64.623			0	
2018/2/2 0:00	1	NB_NS	1	North	29.205	6.319		0	
2018/2/2 0:00	2	NB_MID	1	North	37.283	6.2	6.089	0	
2018/2/2 0:01	6	SB_NS	2	South	44.739	14.8	14.575	0	
2018/2/2 0:01	2	NB_MID	1	North	41.01	5.155	5.242	0	
2018/2/2 0:01	2	NB_MID	1	North	37.283	1.47	0.949	0	
2018/2/2 0:01	5	SB_MID	2	South	36.039	47.1	47.017	0	
2018/2/2 0:01	6	SB_NS	2	South	36.661	12.3	12.24	0	
2018/2/2 0:01	3	NB_OS	1	North	45.361			0	
2018/2/2 0:01	2	NB_MID	1	North	38.525	41.3	41.06	0	
2018/2/2 0:01	5	SB_MID	2	South	47.224	38.9	38.639	0	

3. Use function WEEKDAY to calculate the label 'Flags.'

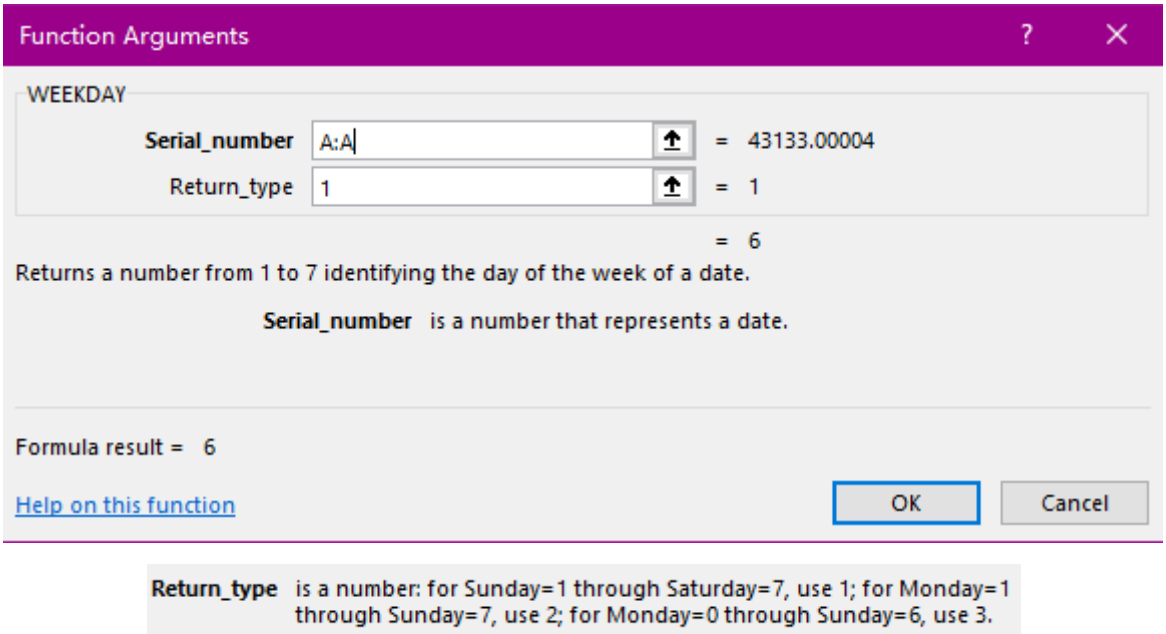
Select the cell that needs to be deal with, then click fx to insert a function.



Select category Date & Time, and select WEEKDAY.



Then, select the Serials number and return type. According to the requirements of this task, 2 should be selected as the return type.

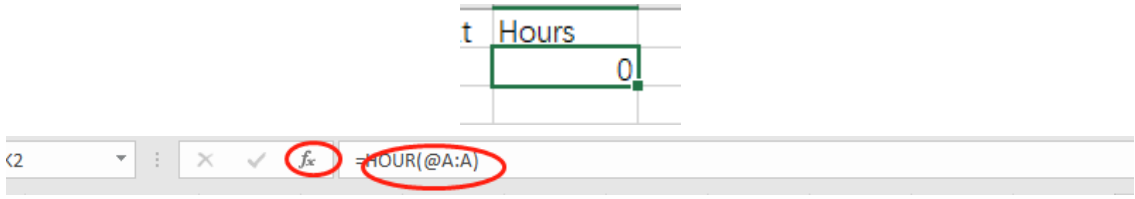


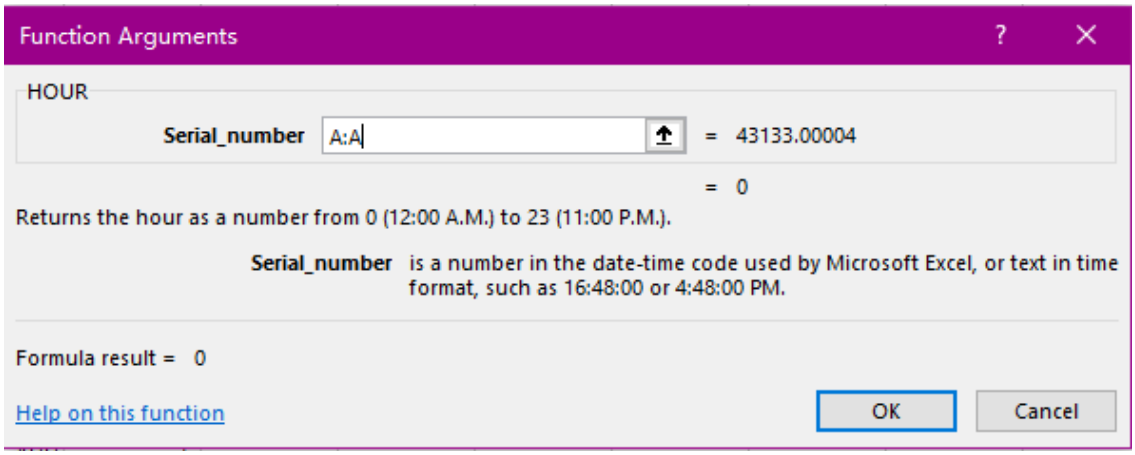
Return_type is a number: for Sunday=1 through Saturday=7, use 1; for Monday=1 through Sunday=7, use 2; for Monday=0 through Sunday=6, use 3.

The label 'Flags' has been gotten, and the function WEEKDAY can be applied to the whole column. First, select the cell just be mentioned, right-click to copy. Second, Select the starting cell, press shift, and then select the ending cell.

2018/2/2 19:39	2	NB_MID	1	North	39.768	1.659	1.221	5
2018/2/2 19:39	6	SB_NS	2	South	38.525	3.861	3.298	5
2018/2/2 19:39	1	NB_NS	1	North	34.798	5.657	4.199	5
2018/2/2 19:39	2	NB_MID	1	North	36.661	4.088	3.375	5
2018/2/2 19:39	1	NB_NS	1	North	34.176	1.538	1.004	5
2018/2/2 19:39	3	NB_OS	1	North	42.875	3.287	3.311	5
2018/2/2 19:39	6	SB_NS	2	South	37.283	4.32	3.868	5
2018/2/2 19:39	2	NB_MID	1	North	31.691	3.529	2.656	5
2018/2/2 19:39	2	NB_MID	1	North	28.584	2.465	1.696	5
2018/2/2 19:39	3	NB_OS	1	North	36.661	5.583	4.46	5
2018/2/2 19:39	2	NB_MID	1	North	29.205	1.991	1.424	5
2018/2/2 19:39	1	NB_NS	1	North	36.039	7.2	6.919	5
2018/2/2 19:39	2	NB_MID	1	North	27.962	1.56	0.863	5
2018/2/2 19:39	1	NB_NS	1	North	34.798	1.607	1.052	5
2018/2/2 19:39	3	NB_OS	1	North	34.176	3.862	3.113	5
2018/2/2 19:39	1	NB_NS	1	North	33.554	2.533	2.03	5
2018/2/2 19:39	3	NB_OS	1	North	39.146	2.543	2.532	5
2018/2/2 19:39	2	NB_MID	1	North	32.932	4.891	5.096	5
2018/2/2 19:39	1	NB_NS	1	North	41.632	3.681	4.013	5
2018/2/2 19:39	6	SB_NS	2	South	42.253	26	25.73	5
2018/2/2 19:39	1	NB_NS	1	North	32.31	13.5	13.264	5
2018/2/2 19:39	6	SB_NS	2	South	35.417	5.526	4.304	5
2018/2/2 19:39	6	SB_NS	2	South	42.253	3.203	3.392	5
2018/2/2 19:39	6	SB_NS	2	South	44.739	2.125	2.046	5
2018/2/2 19:39	5	SB_MID	2	South	33.554	47.8	47.617	5
2018/2/2 19:39	5	SB_MID	2	South	36.661	2.075	1.813	5
2018/2/2 19:39	5	SB_MID	2	South	34.798	1.382	0.78	5
2018/2/2 19:39	4	SB_OS	2	South	38.525	56.2	56.055	5
2018/2/2 19:39	6	SB_NS	2	South	39.768	6.9	6.685	5
2018/2/2 19:39	2	NB_MID	1	North	49.709	33	32.715	5
2018/2/2 19:39	3	NB_OS	1	North	34.798	37.7	37.471	5
2018/2/2 19:39	1	NB_NS	1	North	36.039	21.6	21.309	5
2018/2/2 19:39	3	NB_OS	1	North	32.31	1.523	1.036	5
2018/2/2 19:39	6	SB_NS	2	South	29.205	10.4	10.186	5
2018/2/2 19:40	6	SB_NS	2	South	30.447	2.461	1.978	5
2018/2/2 19:40	5	SB_MID	2	South	36.039	16.3	16.024	5
2018/2/2 19:40	1	NB_NS	1	North	29.205	9.7	9.433	5
2018/2/2 19:40	4	SB_OS	2	South	37.283	17.6	17.356	5
2018/2/2 19:40	6	SB_NS	2	South	36.661	2.746	2.662	5
2018/2/2 19:40	4	SB_OS	2	South	42.253	1.641	1.442	5

- **The second step** is getting the hours of the day. Created a column named hours and then use function HOUR to get the hours of the day. The step is similar to the first step.





The result is:

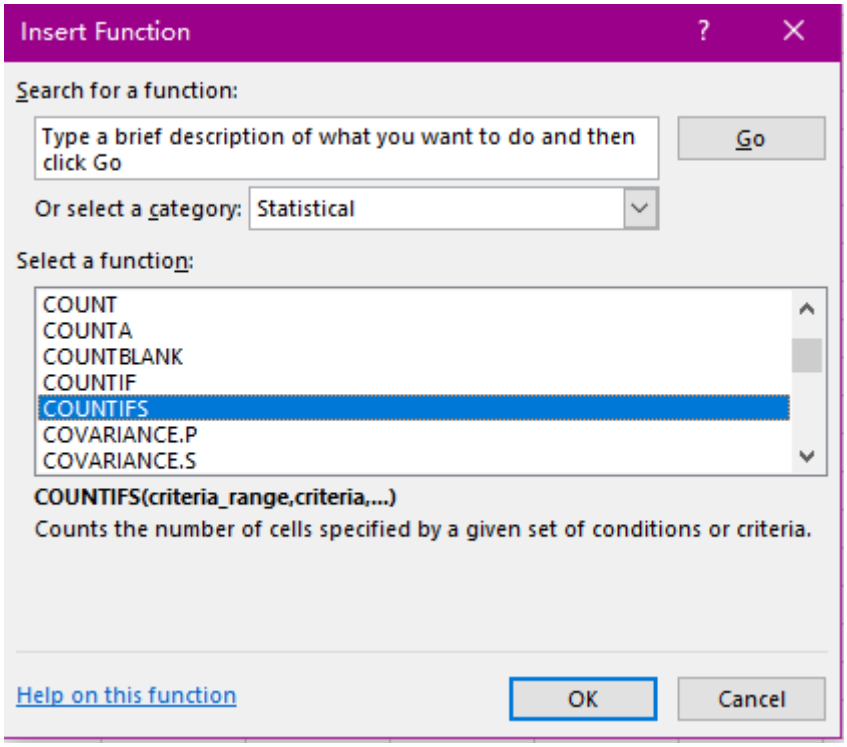
2018/2/16 9:59	4	SB_OS	2	South	35.417	3.411	3.03	5	9
2018/2/16 9:59	1	NB_NS	1	North	26.098	1.114	0.423	5	9
2018/2/16 9:59	6	SB_NS	2	South	35.417	11.4	11.116	5	9
2018/2/16 9:59	1	NB_NS	1	North	26.718	1.842	1.357	5	9
2018/2/16 9:59	1	NB_NS	1	North	26.098	1.371	0.657	5	9
2018/2/16 9:59	5	SB_MID	2	South	31.691	7.7	7.395	5	9
2018/2/16 9:59	5	SB_MID	2	South	41.632	2.149	2.219	5	9
2018/2/16 9:59	2	NB_MID	1	North	18.64	1.26	30.302	5	9
2018/2/16 9:59	6	SB_NS	2	South	38.525	9.6	9.272	5	9
2018/2/16 9:59	5	SB_MID	2	South	44.117	8.7	8.485	5	9
2018/2/16 9:59	6	SB_NS	2	South	31.691	5.929	4.545	5	9
2018/2/16 9:59	6	SB_NS	2	South	32.31	1.488	1.018	5	9
2018/2/16 9:59	6	SB_NS	2	South	32.31	1.385	0.771	5	9
2018/2/16 9:59	6	SB_NS	2	South	35.417	1.547	1.123	5	9
2018/2/16 9:59	6	SB_NS	2	South	34.798	2.218	1.672	5	9
2018/2/16 9:59	3	NB_OS	1	North	42.875	36	35.673	5	9
2018/2/16 9:59	2	NB_MID	1	North	35.417	14.8	14.38	5	9
2018/2/16 9:59	6	SB_NS	2	South	30.447	6.429	5.104	5	9
2018/2/16 9:59	2	NB_MID	1	North	18.021	11.731	2.873	5	9
2018/2/16 10:00	5	SB_MID	2	South	27.34			5	10
2018/2/16 10:00	6	SB_NS	2	South	43.495			5	10
2018/2/16 10:00	4	SB_OS	2	South	34.798			5	10
2018/2/16 10:00	6	SB_NS	2	South	43.495	1.749	1.369	5	10
2018/2/16 10:00	4	SB_OS	2	South	29.825	1.95	1.117	5	10
2018/2/16 10:00	3	NB_OS	1	North	31.691			5	10
2018/2/16 10:00	1	NB_NS	1	North	27.962			5	10
2018/2/16 10:00	2	NB_MID	1	North	30.447			5	10
2018/2/16 10:00	4	SB_OS	2	South	27.962	1.64	0.992	5	10
2018/2/16 10:00	1	NB_NS	1	North	27.34	1.514	0.848	5	10
2018/2/16 10:00	5	SB_MID	2	South	29.205			5	10
2018/2/16 10:00	3	NB_OS	1	North	27.34	2.986	1.975	5	10
2018/2/16 10:00	3	NB_OS	1	North	25.476	1.976	1.256	5	10
2018/2/16 10:00	1	NB_NS	1	North	30.447	2.131	1.665	5	10
2018/2/16 10:00	2	NB_MID	1	North	32.31	3.877	3.484	5	10
2018/2/16 10:00	5	SB_MID	2	South	31.691	4.129	3.986	5	10
2018/2/16 10:00	1	NB_NS	1	North	29.825	2.512	1.855	5	10
2018/2/16 10:00	5	SB_MID	2	South	31.691	1.024	0.595	5	10
2018/2/16 10:00	4	SB_OS	2	South	38.525	4.674	5.76	5	10
2018/2/16 10:00	3	NB_OS	1	North	28.584	2.309	1.922	5	10
2018/2/16 10:00	2	NB_MID	1	North	30.447	3.563	2.667	5	10

- **The third step** is getting the total volume(North and South).

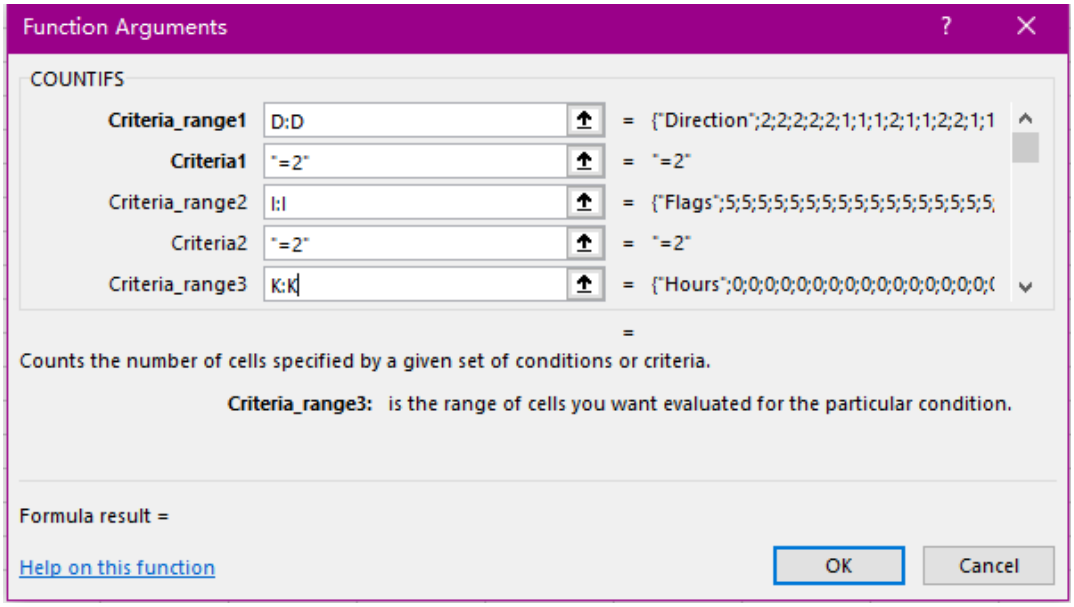
  The function COUNTIFS can be used to solve it. The COUNTIFS function applies conditions to cells that span multiple regions, and then counts the number of times all conditions are met.

  1. Create column 'North_avg,' and 'South_avg.' Click fx to insert a function.

     In statistical, select COUNTIFS.



2. Filter by conditions



D is the column 'Direction,' I is the column 'Flags,' K is the column 'Hours.' Criterial1 means Direction = 2, other similarities.

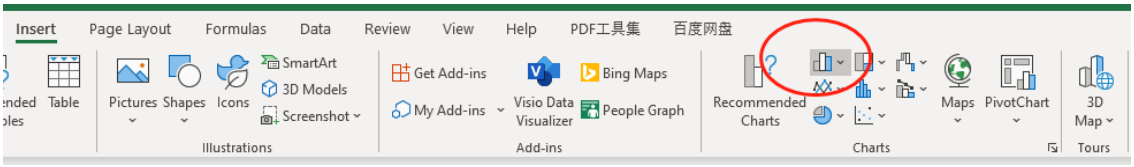
3. Manually change the conditions of Hours from 0-23.

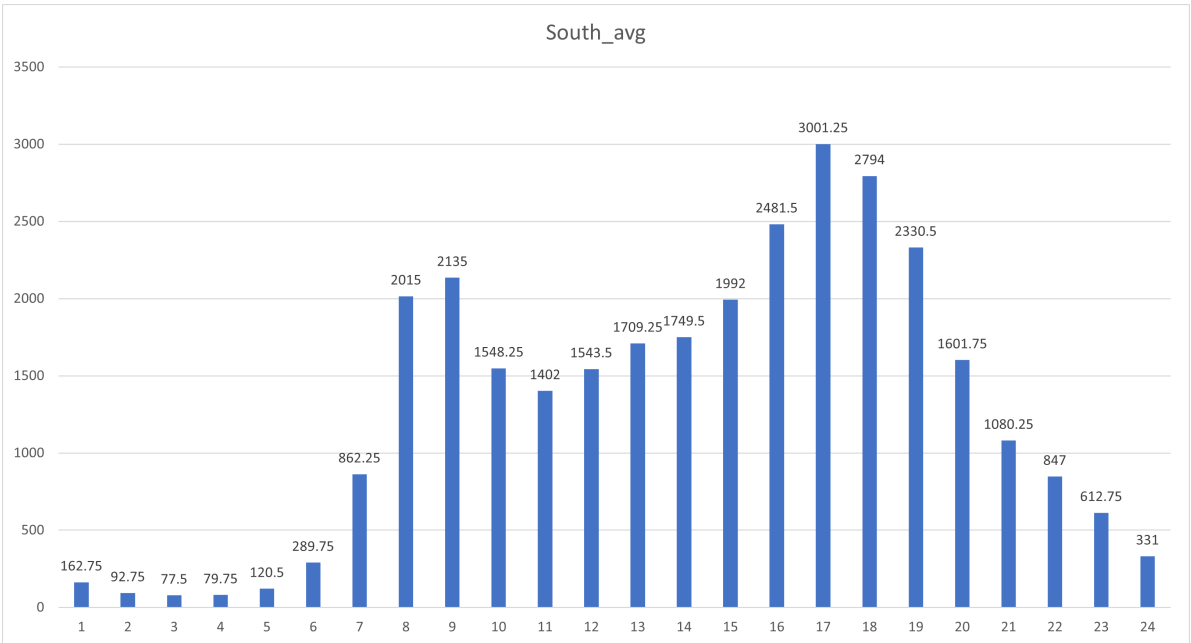
Divide these values by 4(Because there are four Tuesdays in February 2018). Finally, the average number has been gotten. In the same way, the traffic volume in the south can also be gotten.

North_avg	South_avg
126.5	162.75
63	92.75
42.5	77.5
53	79.75
108.25	120.5
361.5	289.75
1260	862.25
2951.5	2015
2727	2135
2430.25	1548.25
1776.75	1402
1634.25	1543.5
1698.75	1709.25
1707.5	1749.5
1642	1992
1805.25	2481.5
2429	3001.25
2655.75	2794
2121.5	2330.5
1308.25	1601.75
882.75	1080.25
648.75	847
453.75	612.75
256.75	331

- **The final step** is to draw charts with excel.

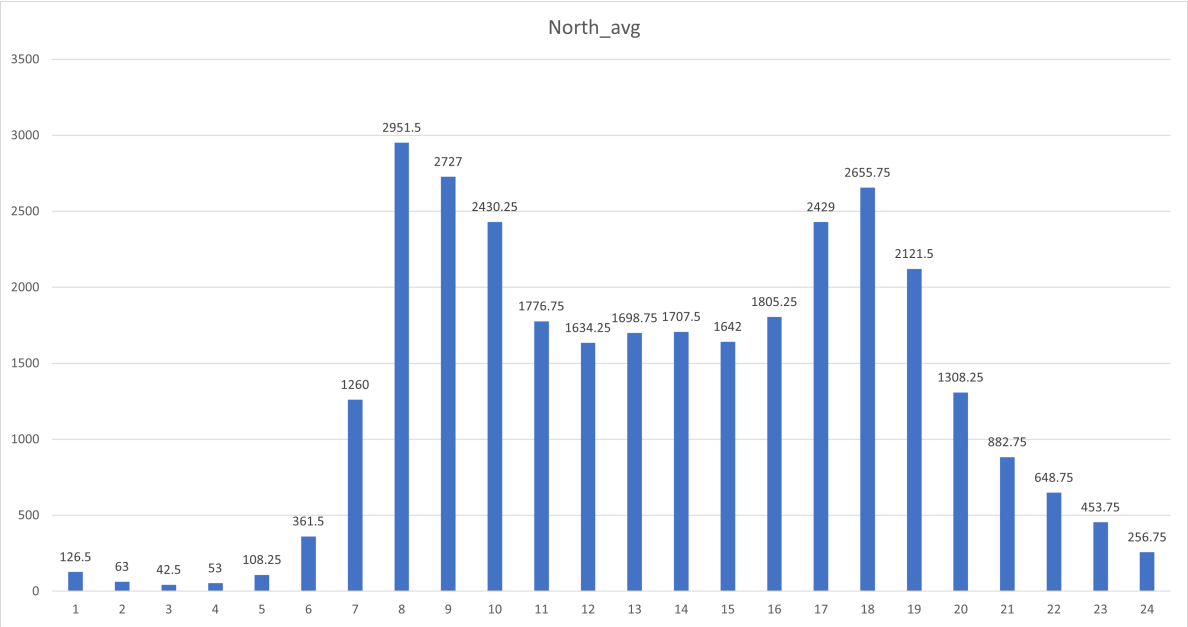
  Select all the North_avg, and click File - Insert, and choose Bar chart.





Right-click, select add data labels.

In the same way, the south volume can also be gotten.



- **Assessment of the two technologies**
  - **similarities:**
    - Both Excel and python have many functions to deal with data.

      For example, in this coursework, the required data can be filtered by python's loc method or achieved using Excel's COUNTIFS.

      Excel and python have many functions with similar functions.
    - Both Excel and Python can realize data visualization.

      The library matplotlib in python can be used for visualization, and Excel also provides the function of generating various charts.

      In python, the kind of plot to produce: The kind of plot to produce: line, bar, hist, box, pie and so on. In comparison, excel can also generate these charts.
    - Both Excel and Python can deal with CSV file.

  - **differences:**
    - Excel
      - Advantage:
        1. Excel has a graphical interface. When processing data, people only need to use the mouse to select the cell that needs to be processed and follow the instructions to write the conditions. This means that even people who are not programmers can perform some complex operations on the data. For programmers, data selection becomes more intuitive.

           For example, in task 3, we need to process the label 'Direction,' 'Hours,' and 'Flags.' Using Excel is more intuitive and convenient than python. When using python, we need to locate twice and groupby once.
        2. When people do not know what method to use to process data, sometimes you can get the answer directly through the Excel menu name.

           When I calculated the label 'Flags,' I did not know the WEEKDAY function, but through a series of clicks, I found it without searching online.

- Limitation:
    1. The amount of data that Excel can handle is smaller than python, and the larger the data, the slower the calculation speed.
    2. Excel can only be used for windows and mac, not for Linux.
- Python
    - Advantage:
        1. People can code functions to solve the same problem.
        2. 1. When there are many repetitive operations to be done, python can reduce a lot of repetitive operations.
           2. Python is a cross-platform language, can also be used on Linux.
           3. Python has many modules about machine learning and deep learning, which is more convenient to use after python processing data.
           4. Python can integrate SQL statements, making it more convenient to process databases.
           5. Python can handle large amounts of data.
    - Limitation:
    1. Compared with Excel, python is harder to learn. The free and flexible syntax may produce many bugs that are difficult to fix.

In this task, I do not think Excel is worse than python. Even sometimes, Excel is more intuitive in terms of filtering conditions, but Python can do everything Excel can do. Therefore, as a computer student, we still have to learn python well.