Milwaukee
Detroit
Chicago
United States
Denver
Indianapolis
Columbus

Project: Finding Heavy Traffic Indicators on I-94

Introduction:

We're going to analyze a dataset about the westbound traffic on the I-94 interstate highway. This highway stretches from Billings, Montana all the way to Port Huron Michigan.

The goal of our analysis is to determine a few indicators of heavy traffic on I-94. These indicators can be weather type, time of the day, time of the week, etc. For instance, we may find out that the traffic is usually heavier in the summer or when it snows.

Importing and exploring data:

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

Metro_Is_Traffic_Vol = pd.read_csv('Metro_Interstate_Traffic_Volume.csv')

print (Metro_Is_Traffic_Vol.head())
print ()
print (Metro_Is_Traffic_Vol.tail())
print ()
print (Metro_Is_Traffic_Vol.info())
```

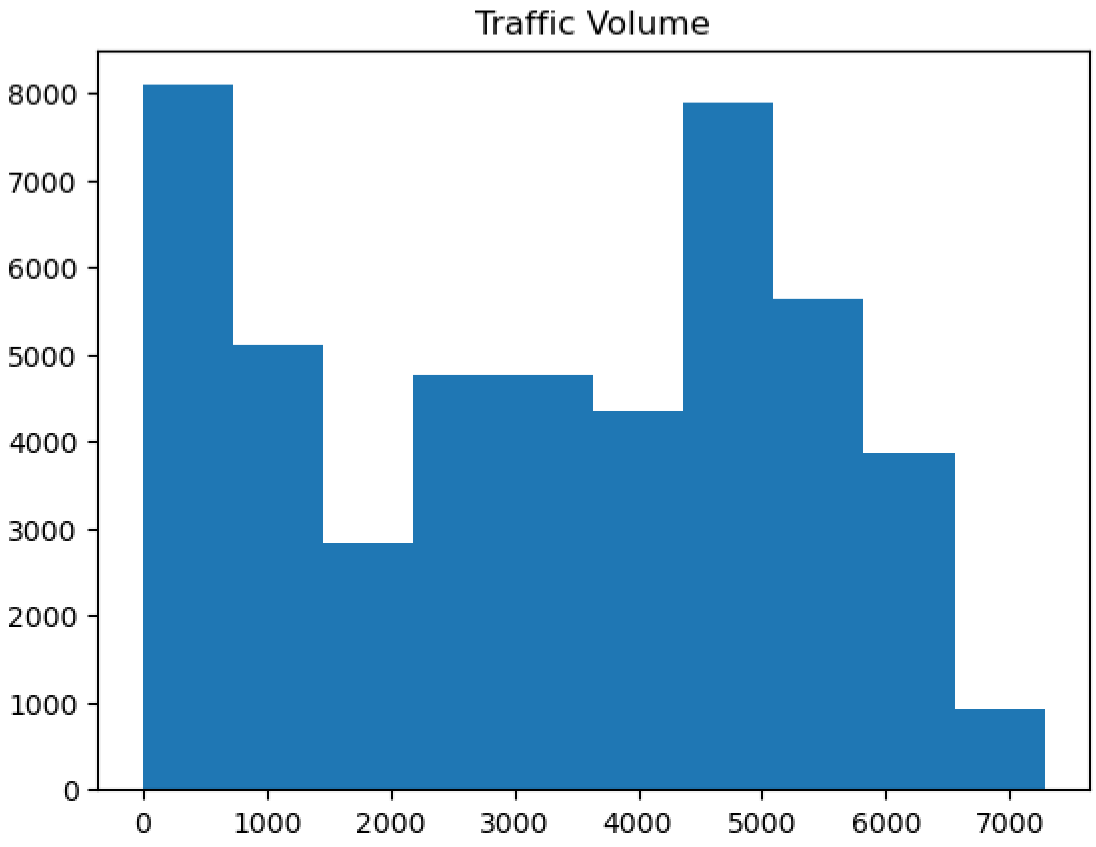
```
  holiday    temp  rain_1h  snow_1h  clouds_all weather_main  \
0    None  288.28      0.0      0.0          40       Clouds   
1    None  289.36      0.0      0.0          75       Clouds   
2    None  289.58      0.0      0.0          90       Clouds   
3    None  290.13      0.0      0.0          90       Clouds   
4    None  291.14      0.0      0.0          75       Clouds   

  weather_description            date_time  traffic_volume  
0    scattered clouds  2012-10-02 09:00:00            5545  
1       broken clouds  2012-10-02 10:00:00            4516  
2     overcast clouds  2012-10-02 11:00:00            4767  
3     overcast clouds  2012-10-02 12:00:00            5026  
4       broken clouds  2012-10-02 13:00:00            4918  
```

```
      holiday    temp  rain_1h  snow_1h  clouds_all  weather_main  \
48199    None  283.45      0.0      0.0          75        Clouds
48200    None  282.76      0.0      0.0          90        Clouds
48201    None  282.73      0.0      0.0          90  Thunderstorm
48202    None  282.09      0.0      0.0          90        Clouds
48203    None  282.12      0.0      0.0          90        Clouds

          weather_description            date_time  traffic_volume
48199           broken clouds  2018-09-30 19:00:00            3543
48200         overcast clouds  2018-09-30 20:00:00            2781
48201  proximity thunderstorm  2018-09-30 21:00:00            2159
48202         overcast clouds  2018-09-30 22:00:00            1450
48203         overcast clouds  2018-09-30 23:00:00             954

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48204 non-null  object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  object
 8   traffic_volume       48204 non-null  int64
dtypes: float64(3), int64(2), object(4)
memory usage: 3.3+ MB
None
```

Analyzing Traffic Volume:

In [39]:

```
plt.hist(Metro_Is_Traffic_Vol['traffic_volume'])
plt.title('Traffic Volume')
plt.show()
```



```
In [4]: Metro_Is_Traffic_Vol['traffic_volume'].describe()
```

```
Out[4]: count    48204.000000
        mean      3259.818355
        std       1986.860670
        min          0.000000
        25%       1193.000000
        50%       3380.000000
        75%       4933.000000
        max       7280.000000
        Name: traffic_volume, dtype: float64
```

The traffic volume ranges from 0 to 7,280. The mean or average traffic volume is 3,260(rounded). The low end is about 1,193 with the top 25% being 4,933.

Assuming this a relationsip between heavy hours of commuting traffic. ex: morning and evening rush hours.

There is a few spikes in the chart as well. I would assume this would correlate more with non peak commute times and times where there is an unforseen event such as a car accident or inclement weather.

Traffic Volume: Day vs. Night:

This possibility that nighttime and daytime might influence traffic volume gives our analysis an interesting direction: comparing daytime with nighttime data.

We'll start by dividing the dataset into two parts:

- Daytime data: hours from 7 a.m. to 7 p.m. (12 hours)
- Nighttime data: hours from 7 p.m. to 7 a.m. (12 hours)

```
In [5]: print (Metro_Is_Traffic_Vol['date_time'])
```

```
0        2012-10-02 09:00:00
1        2012-10-02 10:00:00
```

```
2        2012-10-02 11:00:00
3        2012-10-02 12:00:00
4        2012-10-02 13:00:00
                ...
48199    2018-09-30 19:00:00
48200    2018-09-30 20:00:00
48201    2018-09-30 21:00:00
48202    2018-09-30 22:00:00
48203    2018-09-30 23:00:00
Name: date_time, Length: 48204, dtype: object
```

In [6]:

```
Metro_Is_Traffic_Vol['date_time'] = pd.to_datetime(Metro_Is_Traffic_Vol
                                                   ['date_time'])

print (Metro_Is_Traffic_Vol['date_time'])
```

```
0       2012-10-02 09:00:00
1       2012-10-02 10:00:00
2       2012-10-02 11:00:00
3       2012-10-02 12:00:00
4       2012-10-02 13:00:00
                ...
48199   2018-09-30 19:00:00
48200   2018-09-30 20:00:00
48201   2018-09-30 21:00:00
48202   2018-09-30 22:00:00
48203   2018-09-30 23:00:00
Name: date_time, Length: 48204, dtype: datetime64[ns]
```

In [7]:

```
Metro_Is_Traffic_Vol['date_time'].dt.hour
```

Out[7]:

```
0         9
1        10
2        11
3        12
4        13
         ..
48199    19
48200    20
48201    21
48202    22
48203    23
Name: date_time, Length: 48204, dtype: int64
```

In [8]:

```
day_time = Metro_Is_Traffic_Vol.copy()[(Metro_Is_Traffic_Vol['date_time'].dt.hour
                                        >=7) &
                                        (Metro_Is_Traffic_Vol['date_time'].dt.hour < 19)

print(day_time['date_time'].dt.hour)
```

```
0         9
1        10
2        11
3        12
4        13
         ..
48194    15
48195    15
48196    16
48197    17
48198    18
Name: date_time, Length: 23877, dtype: int64
```

In [9]:

```
evening_time = Metro_Is_Traffic_Vol.copy()[(Metro_Is_Traffic_Vol['date_time'].dt.hour
                                        >= 19) |
                                        (Metro_Is_Traffic_Vol['date_time'].dt.hour < 7)]
```

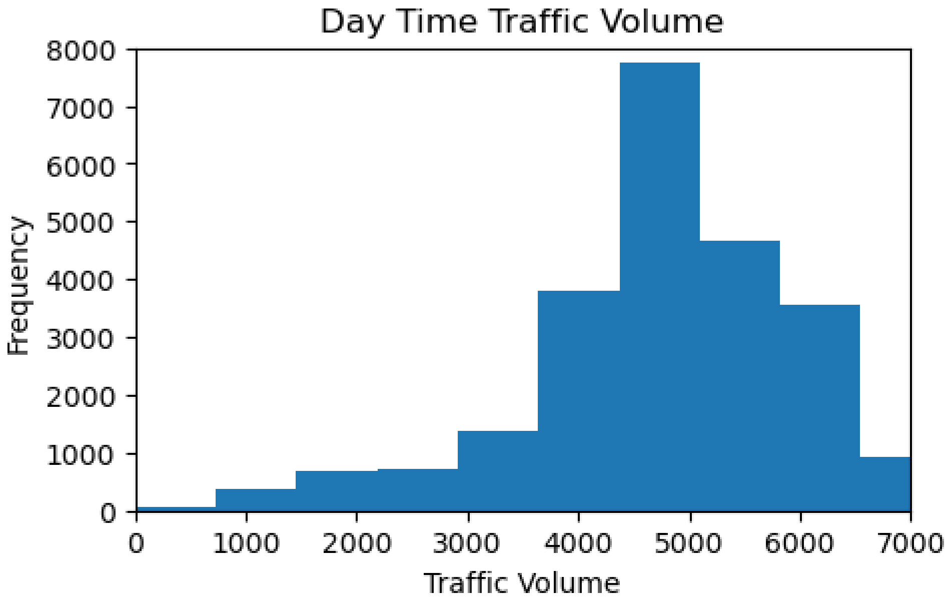
```
print(evening_time['date_time'].dt.hour)
```

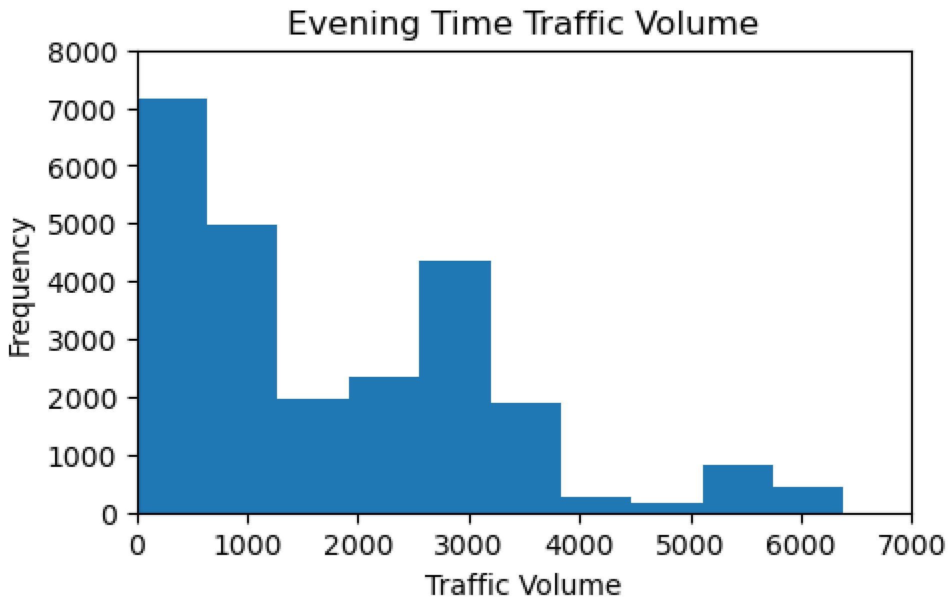
```
10       19
11       20
12       21
13       22
14       23
         ..
48199    19
48200    20
48201    21
48202    22
48203    23
Name: date_time, Length: 24327, dtype: int64
```

Graphing Traffic:

In [10]:

```
plt.figure(figsize = (11,3))
plt.subplot(1,2,1)
plt.hist(day_time['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.ylabel('Frequency')
plt.title('Day Time Traffic Volume')
plt.xlim(0,7000)
plt.ylim(0,8000)

plt.figure(figsize = (11,3))
plt.subplot(1,2,2)
plt.hist(evening_time['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.ylabel('Frequency')
plt.title('Evening Time Traffic Volume')
plt.xlim(0,7000)
plt.ylim(0,8000)
plt.show()
```





In [11]:
```
print (day_time['traffic_volume'].describe())
print ()
print (evening_time['traffic_volume'].describe())
```

```
count    23877.000000
mean      4762.047452
std       1174.546482
min          0.000000
25%       4252.000000
50%       4820.000000
75%       5559.000000
max       7280.000000
Name: traffic_volume, dtype: float64

count    24327.000000
mean      1785.377441
std       1441.951197
min          0.000000
25%        530.000000
50%       1287.000000
75%       2819.000000
max       6386.000000
Name: traffic_volume, dtype: float64
```

According to the histograms it appears that the daytime traffic is left skewed. The evening traffic is skewed to the right. The traffic at night tends to tapper off not really going above 4,000. Day time sees the most traffice volume.

Time Indicators:

We're going to look at a few line plots showing how the traffic volume changed according to the following parameters:

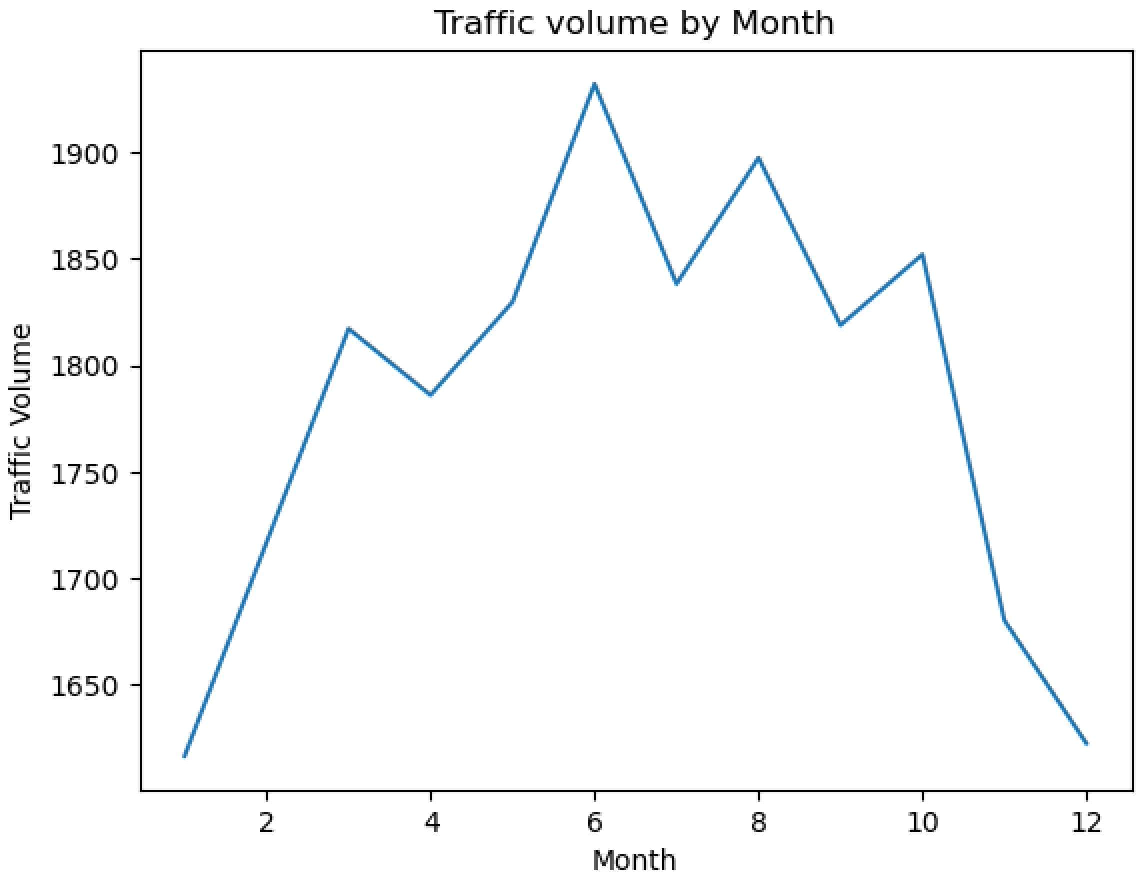
- Month
- Day of the week
- Time of day

In [12]:
```
day_time['month'] = day_time['date_time'].dt.month
by_month = day_time.groupby('month').mean()
by_month['traffic_volume']
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2035601682.py:2: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_month = day_time.groupby('month').mean()
```

```
Out[12]: month
         1      4495.613727
         2      4711.198394
         3      4889.409560
         4      4906.894305
         5      4911.121609
         6      4898.019566
         7      4595.035744
         8      4928.302035
         9      4870.783145
         10     4921.234922
         11     4704.094319
         12     4374.834566
         Name: traffic_volume, dtype: float64
```

```
In [38]: plt.plot(by_month['traffic_volume'])
         plt.xlabel('Month')
         plt.ylabel('Traffic Volume')
         plt.title('Traffic volume by Month')
         plt.show()
```



Notes:

Traffic seems to increase from March to June. Falling off in July. Then increasing from August until October. The winter months November-February have the least amount of traffic.

Time Indicators by Day of the week:

Now Lets look at traffice by day of the week.

In [14]:

```
day_time['dayofweek'] = day_time['date_time'].dt.dayofweek
by_dayofweek = day_time.groupby('dayofweek').mean()
by_dayofweek['traffic_volume']
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\1471914992.py:2: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_dayofweek = day_time.groupby('dayofweek').mean()
```

Out[14]:

```
dayofweek
0    4893.551286
1    5189.004782
2    5284.454282
3    5311.303730
4    5291.600829
5    3927.249558
6    3436.541789
Name: traffic_volume, dtype: float64
```

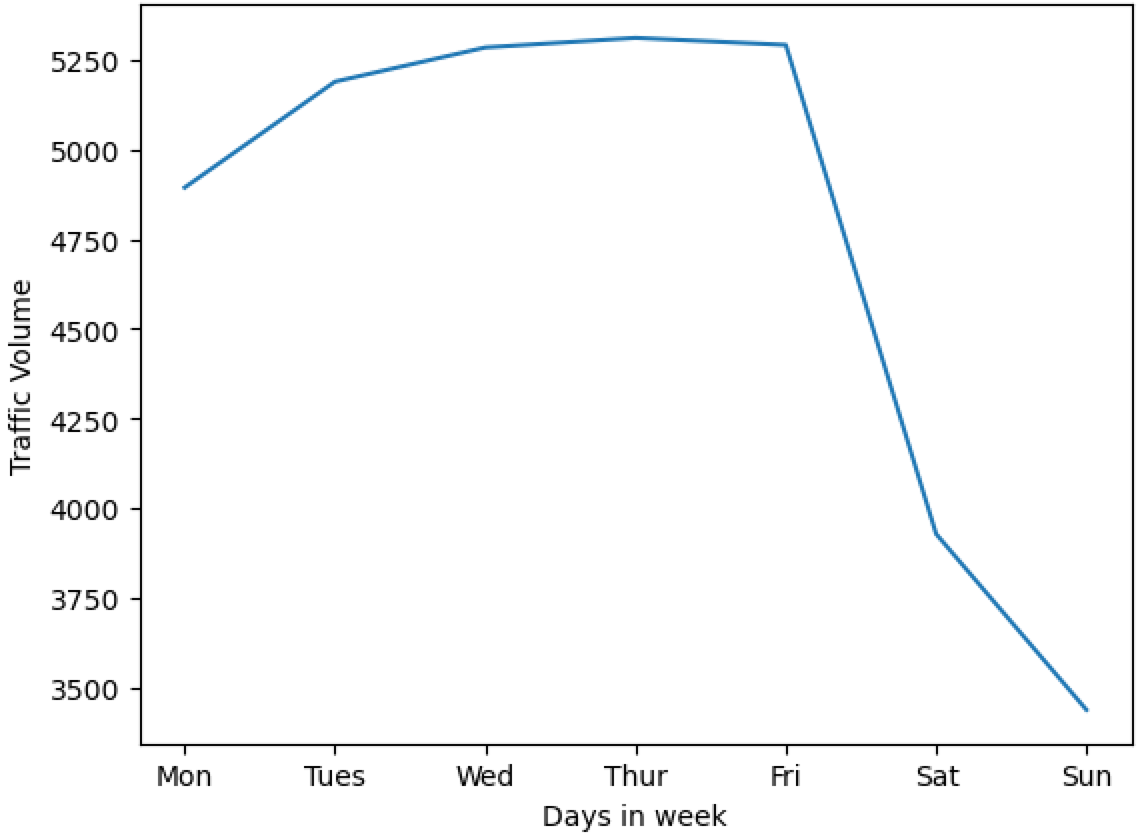
In [15]:

```
print(by_dayofweek['traffic_volume'])
```

```
dayofweek
0    4893.551286
1    5189.004782
2    5284.454282
3    5311.303730
4    5291.600829
5    3927.249558
6    3436.541789
Name: traffic_volume, dtype: float64
```

In [16]:

```
labels = ['Mon','Tues','Wed','Thur','Fri','Sat','Sun']

plt.plot(by_dayofweek['traffic_volume'])
plt.xlabel('Days in week')
plt.ylabel('Traffic Volume')
plt.xticks(by_dayofweek.index,labels)
plt.show()
```



Traffic by weekday vs weekend:

In [17]:

```
day_time['hour'] = day_time['date_time'].dt.hour
bussiness_days = day_time.copy()[day_time['dayofweek'] <= 4] # 4 == Friday
weekend = day_time.copy()[day_time['dayofweek'] >= 5] # 5 == Saturday
by_hour_business = bussiness_days.groupby('hour').mean()
by_hour_weekend = weekend.groupby('hour').mean()

print('business day', by_hour_business['traffic_volume'])
print ()
print('weekend',by_hour_weekend['traffic_volume'])
```

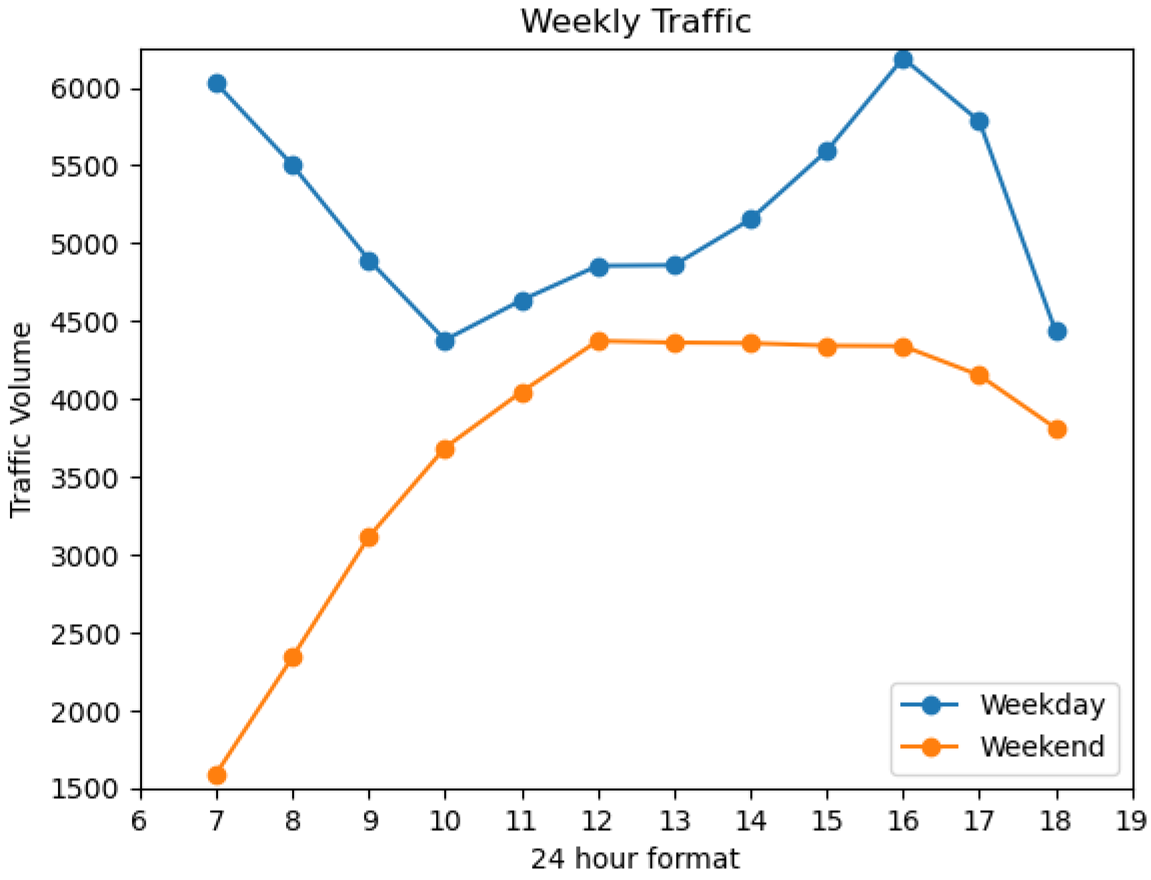
```
business day hour
7     6030.413559
8     5503.497970
9     4895.269257
10    4378.419118
11    4633.419470
12    4855.382143
13    4859.180473
14    5152.995778
15    5592.897768
16    6189.473647
17    5784.827133
18    4434.209431
Name: traffic_volume, dtype: float64

weekend hour
7     1589.365894
8     2338.578073
9     3111.623917
10    3686.632302
11    4044.154955
12    4372.482883
13    4362.296564
14    4358.543796
```

```
15    4342.456881
16    4339.693805
17    4151.919929
18    3811.792279
Name: traffic_volume, dtype: float64
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2579173255.py:4: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_hour_business = bussiness_days.groupby('hour').mean()
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2579173255.py:5: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_hour_weekend = weekend.groupby('hour').mean()
```

In [18]:

```
plt.plot(by_hour_business['traffic_volume'], marker='o', label='Weekday')
plt.xlim(6, 19)
plt.ylim(1500,6250)
plt.locator_params(axis='x', nbins=14)
plt.locator_params(axis='y', nbins=10)
plt.xlabel('24 hour format')
plt.ylabel('Traffic Volume')
plt.title("Week")

plt.plot(by_hour_weekend['traffic_volume'], marker='o',label='Weekend')
plt.xlim(6, 19)
plt.ylim(1500,6250)
plt.locator_params(axis='x', nbins=14)
plt.locator_params(axis='y', nbins=10)
plt.xlabel('24 hour format')
plt.ylabel('Traffic Volume')
plt.title("Weekly Traffic")
plt.legend()
plt.show()
```



According to the graph. The weekday starts with high traffic at 7am and decreases until 11am where it gradually increases and starts to decline after 4pm. The rush hours for the business days tend to be at 7 am and 4pm.

The weekdend traffic increases gradually all morning. Traffic peaks around noon to 4pm and tapers off slowly after 4pm. The busiest period on weekends appear to be between noon and 4pm

The weekend traffic does not surpass the lowest amount of traffic during the weekeday.

Weather Indicators:

Lets see how weather impacts traffic.

In [19]:
```
print (day_time)
```

```
      holiday    temp  rain_1h  snow_1h  clouds_all weather_main  \
0        None  288.28     0.00      0.0          40       Clouds
1        None  289.36     0.00      0.0          75       Clouds
2        None  289.58     0.00      0.0          90       Clouds
3        None  290.13     0.00      0.0          90       Clouds
4        None  291.14     0.00      0.0          75       Clouds
...       ...     ...      ...      ...         ...          ...
48194    None  283.84     0.00      0.0          75         Rain
48195    None  283.84     0.00      0.0          75      Drizzle
48196    None  284.38     0.00      0.0          75         Rain
48197    None  284.79     0.00      0.0          75       Clouds
48198    None  284.20     0.25      0.0          75         Rain

           weather_description           date_time  traffic_volume  month  \
0             scattered clouds 2012-10-02 09:00:00            5545     10
1                broken clouds 2012-10-02 10:00:00            4516     10
2              overcast clouds 2012-10-02 11:00:00            4767     10
3              overcast clouds 2012-10-02 12:00:00            5026     10
4                broken clouds 2012-10-02 13:00:00            4918     10
...                        ...                 ...             ...    ...
48194    proximity shower rain 2018-09-30 15:00:00            4302      9
48195  light intensity drizzle 2018-09-30 15:00:00            4302      9
48196               light rain 2018-09-30 16:00:00            4283      9
48197            broken clouds 2018-09-30 17:00:00            4132      9
48198               light rain 2018-09-30 18:00:00            3947      9

       dayofweek  hour
0              1     9
1              1    10
2              1    11
3              1    12
4              1    13
...          ...   ...
48194          6    15
48195          6    15
48196          6    16
48197          6    17
48198          6    18

[23877 rows x 12 columns]
```

In [20]:
```
day_time.loc[:,["temp","rain_1h","snow_1h","clouds_all",
           "traffic_volume"]].corr()["traffic_volume"]
```

Out[20]:
```
temp              0.128317
rain_1h           0.003697
snow_1h           0.001265
```

```
clouds_all        -0.032932
traffic_volume     1.000000
Name: traffic_volume, dtype: float64
```

In [21]:

```
day_time['temp'].describe()
```

Out[21]:

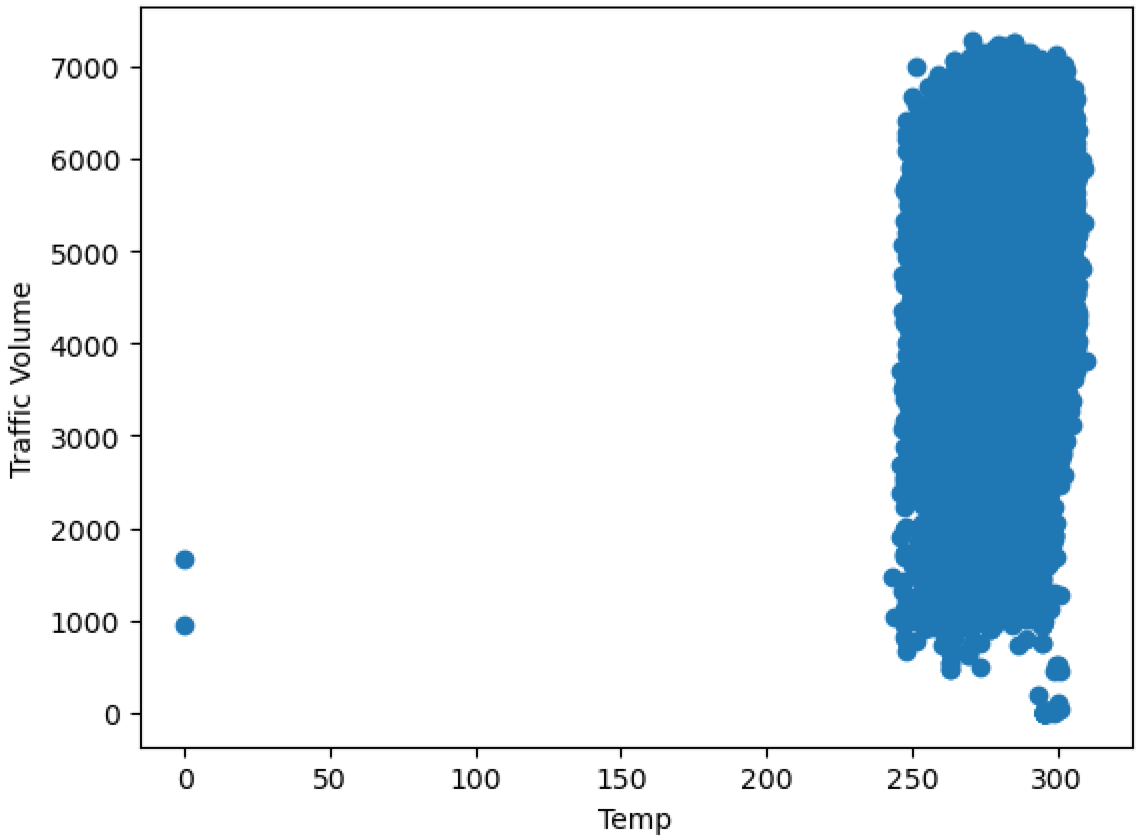
```
count    23877.000000
mean       282.257596
std         13.298885
min          0.000000
25%        272.680000
50%        283.780000
75%        293.440000
max        310.070000
Name: temp, dtype: float64
```

In [22]:

```
plt.scatter(day_time['temp'], day_time['traffic_volume'])
plt.xlabel('Temp')
plt.ylabel('Traffic Volume')
```

Out[22]:

```
Text(0, 0.5, 'Traffic Volume')
```



None of these columns seem to impact the traffic greatly. As we can see when I did the correlation all of the values were low except for temp being the highest.

Logically it would not make sense for temperature to impact traffic volume. I used a scatter plot to show traffic is all lumped around the same temperature range. No useful information can be gathered from this method.

Weather Types:

Lets Explore weather types

In [23]:

```
by_weather_main = day_time.groupby('weather_main').mean()
by_weather_description = day_time.groupby('weather_description').mean()
```

```
print (by_weather_main)
print ()
print (by_weather_description)
```

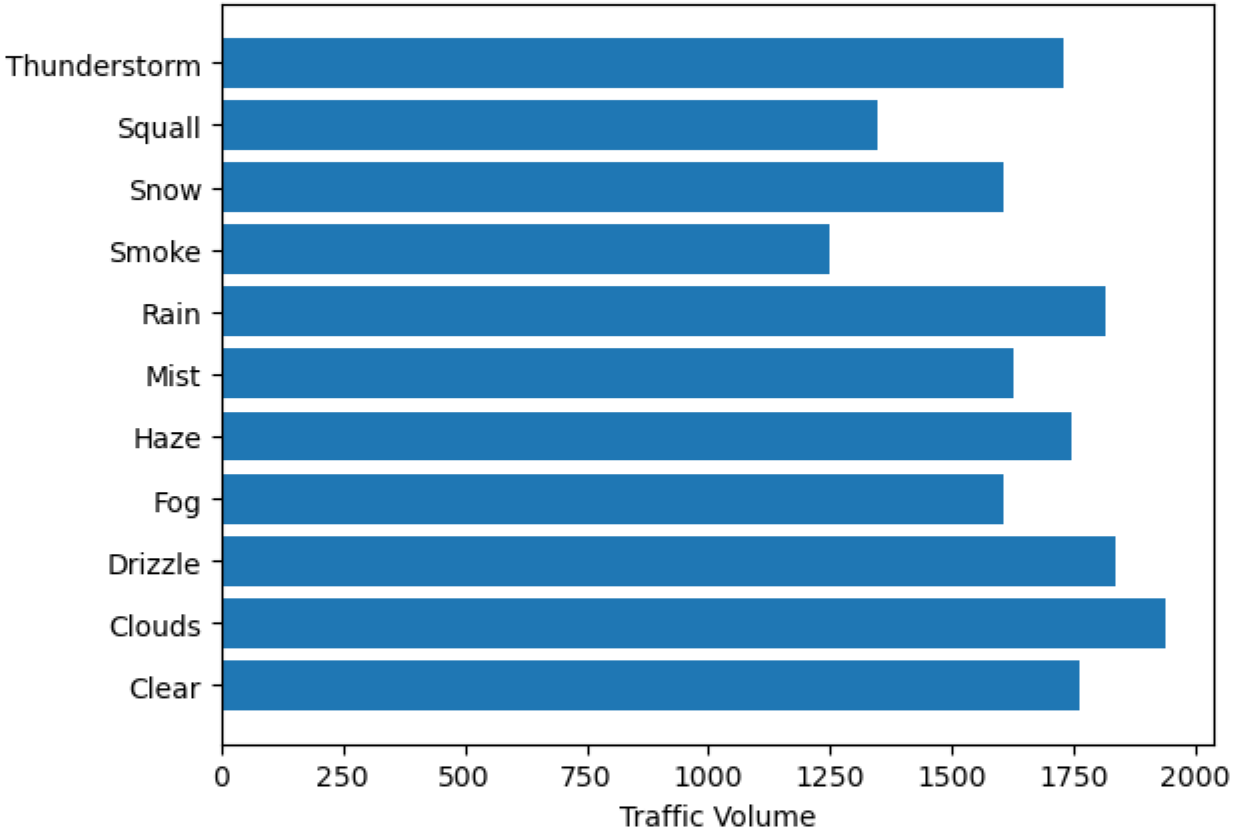
```
                    temp   rain_1h   snow_1h  clouds_all  traffic_volume  \
weather_main
Clear         283.812078  0.000000  0.000000    1.670265     4778.416260
Clouds        282.929274  0.000000  0.000000   62.667548     4865.415996
Drizzle       284.456433  0.170804  0.000000   84.704417     4837.212911
Fog           277.579641  0.163840  0.001409   65.477901     4372.491713
Haze          275.319353  0.040036  0.000000   64.000000     4609.893285
Mist          279.420825  0.249992  0.000825   74.961435     4623.976475
Rain          287.089601  3.972943  0.000292   75.870116     4815.568462
Smoke         292.405833  0.878333  0.000000   53.333333     4564.583333
Snow          267.984505  0.014017  0.001768   80.501376     4396.321183
Squall        296.730000  1.020000  0.000000   75.000000     4211.000000
Thunderstorm  293.364678  1.146475  0.000000   75.184035     4648.212860

                 month  dayofweek       hour
weather_main
Clear         6.490599   3.138928  12.404248
Clouds        6.393243   3.005631  12.911974
Drizzle       7.105323   2.934315  12.308041
Fog           6.646409   2.798343  10.325967
Haze          5.832134   2.754197  12.467626
Mist          6.734285   2.895102  11.078288
Rain          6.774023   2.914467  12.642379
Smoke         6.833333   2.416667  13.166667
Snow          6.374828   2.750344  12.153370
Squall        7.000000   2.000000  14.000000
Thunderstorm  7.108647   2.955654  12.694013

                                           temp      rain_1h   snow_1h  \
weather_description
SQUALLS                              296.730000     1.020000  0.000000
Sky is Clear                         293.232549     0.000000  0.000000
broken clouds                        282.372927     0.000000  0.000000
drizzle                              283.573777     0.145072  0.000000
few clouds                           284.272965     0.000000  0.000000
fog                                  277.579641     0.163840  0.001409
freezing rain                        272.860000     0.000000  0.000000
haze                                 275.319353     0.040036  0.000000
heavy intensity drizzle              285.467931     0.276207  0.000000
heavy intensity rain                 290.231781     2.670548  0.000000
heavy snow                           269.256188     0.002375  0.000000
light intensity drizzle              284.902199     0.178848  0.000000
light intensity shower rain          290.563000     0.433000  0.000000
light rain                           286.349835     0.137147  0.000000
light rain and snow                  275.607500     0.317500  0.000000
light shower snow                    268.213636     0.000000  0.000000
light snow                           267.085634     0.015297  0.002106
mist                                 279.420825     0.249992  0.000825
moderate rain                        287.110124     0.572153  0.001057
overcast clouds                      278.802215     0.000000  0.000000
proximity shower rain                291.460090     0.279279  0.000000
proximity thunderstorm               293.552376     0.967756  0.000000
proximity thunderstorm with drizzle  287.913333     0.260000  0.000000
proximity thunderstorm with rain     291.210556     0.867222  0.000000
scattered clouds                     287.829086     0.000000  0.000000
shower drizzle                       271.330000     0.000000  0.000000
shower snow                          268.680000     0.000000  0.000000
sky is clear                         282.171390     0.000000  0.000000
sleet                                275.746667     0.000000  0.000000
smoke                                292.405833     0.878333  0.000000
snow                                 271.014891     0.024745  0.003723
```

```
thunderstorm                         295.168542     0.702083  0.000000
thunderstorm with drizzle            287.880000     5.345000  0.000000
thunderstorm with heavy rain         292.783200     3.595600  0.000000
thunderstorm with light drizzle      290.885000     2.635000  0.000000
thunderstorm with light rain         292.243478     1.190000  0.000000
thunderstorm with rain               293.074500     1.460000  0.000000
very heavy rain                      296.680000  1426.242857  0.000000

                                     clouds_all  traffic_volume     month  \
weather_description
SQUALLS                               75.000000     4211.000000  7.000000
Sky is Clear                           0.000000     4919.009390  7.557512
broken clouds                         72.635875     4824.130326  6.675260
drizzle                               88.589928     4737.330935  7.244604
few clouds                            19.391951     4839.818023  6.159230
fog                                   65.477901     4372.491713  6.646409
freezing rain                         90.000000     4314.000000  6.500000
haze                                  64.000000     4609.893285  5.832134
heavy intensity drizzle               89.172414     4738.586207  7.551724
heavy intensity rain                  82.799087     4610.356164  7.150685
heavy snow                            85.287500     4411.681250  5.140625
light intensity drizzle               82.565445     4890.164049  7.020942
light intensity shower rain           88.500000     4558.100000  5.700000
light rain                            72.672525     4859.650849  6.610428
light rain and snow                   83.500000     5579.750000  7.500000
light shower snow                     81.909091     4618.636364  9.545455
light snow                            77.714724     4430.858896  6.734151
mist                                  74.961435     4623.976475  6.734285
moderate rain                         80.591083     4769.643312  7.008917
overcast clouds                       90.120696     4861.124952  6.078143
proximity shower rain                 78.108108     4901.756757  6.972973
proximity thunderstorm                73.511551     4684.356436  7.118812
proximity thunderstorm with drizzle   87.833333     5121.833333  8.500000
proximity thunderstorm with rain      82.666667     4501.611111  6.944444
scattered clouds                      40.043099     4936.787712  6.528198
shower drizzle                        90.000000     4932.666667  6.000000
shower snow                           90.000000     5664.000000  3.000000
sky is clear                           1.961161     4753.930294  6.304783
sleet                                 90.000000     4312.666667  7.000000
smoke                                 53.333333     4564.583333  6.833333
snow                                  88.737226     4054.065693  6.416058
thunderstorm                          71.437500     4724.708333  7.166667
thunderstorm with drizzle             90.000000     2297.000000  9.000000
thunderstorm with heavy rain          82.480000     4555.760000  6.600000
thunderstorm with light drizzle       90.000000     4960.000000  8.000000
thunderstorm with light rain          76.565217     4336.130435  6.826087
thunderstorm with rain                82.350000     4522.950000  7.050000
very heavy rain                       51.857143     4780.571429  7.000000

                                     dayofweek       hour
weather_description
SQUALLS                               2.000000  14.000000
Sky is Clear                          2.895540  12.453052
broken clouds                         2.998210  12.811314
drizzle                               3.028777  11.697842
few clouds                            2.977253  12.633421
fog                                   2.798343  10.325967
freezing rain                         0.500000  13.500000
haze                                  2.754197  12.467626
heavy intensity drizzle               2.896552  12.275862
heavy intensity rain                  2.858447  12.442922
heavy snow                            2.975000  12.303125
light intensity drizzle               2.897033  12.612565
light intensity shower rain           3.200000  11.900000
light rain                            2.928530  12.779731
light rain and snow                   1.250000  15.000000
light shower snow                     1.181818  13.272727
light snow                            2.724949  11.978528
mist                                  2.895102  11.078288
moderate rain                         2.917197  12.301911
overcast clouds                       3.042553  12.765957
proximity shower rain                 2.891892  13.441441
proximity thunderstorm                2.894389  12.828383
proximity thunderstorm with drizzle   2.500000  12.166667
proximity thunderstorm with rain      3.222222  11.333333
scattered clouds                      2.986245  13.359927
shower drizzle                        1.666667  11.000000
shower snow                           4.000000   7.000000
sky is clear                          3.181316  12.395748
sleet                                 3.333333  14.000000
smoke                                 2.416667  13.166667
snow                                  2.554745  12.875912
thunderstorm                          3.041667  13.250000
thunderstorm with drizzle             5.000000  12.500000
thunderstorm with heavy rain          3.000000  11.920000
thunderstorm with light drizzle       3.333333  12.833333
thunderstorm with light rain          3.173913  11.869565
thunderstorm with rain                2.950000  12.600000
very heavy rain                       1.571429  11.714286
```

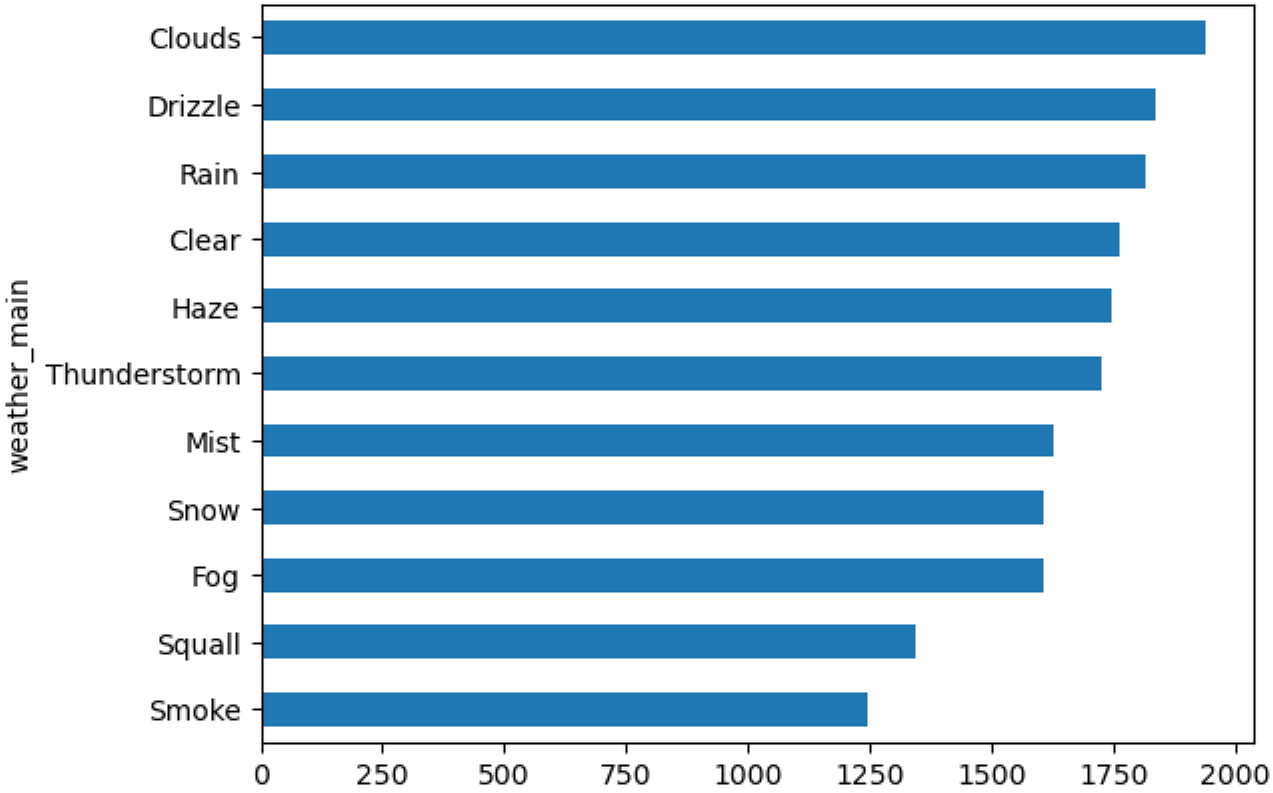
```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2699026515.py:1: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_weather_main = day_time.groupby('weather_main').mean()
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2699026515.py:2: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_weather_description = day_time.groupby('weather_description').mean()
```

In [40]:

```
plt.barh(by_weather_main.index, by_weather_main['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.show()
```



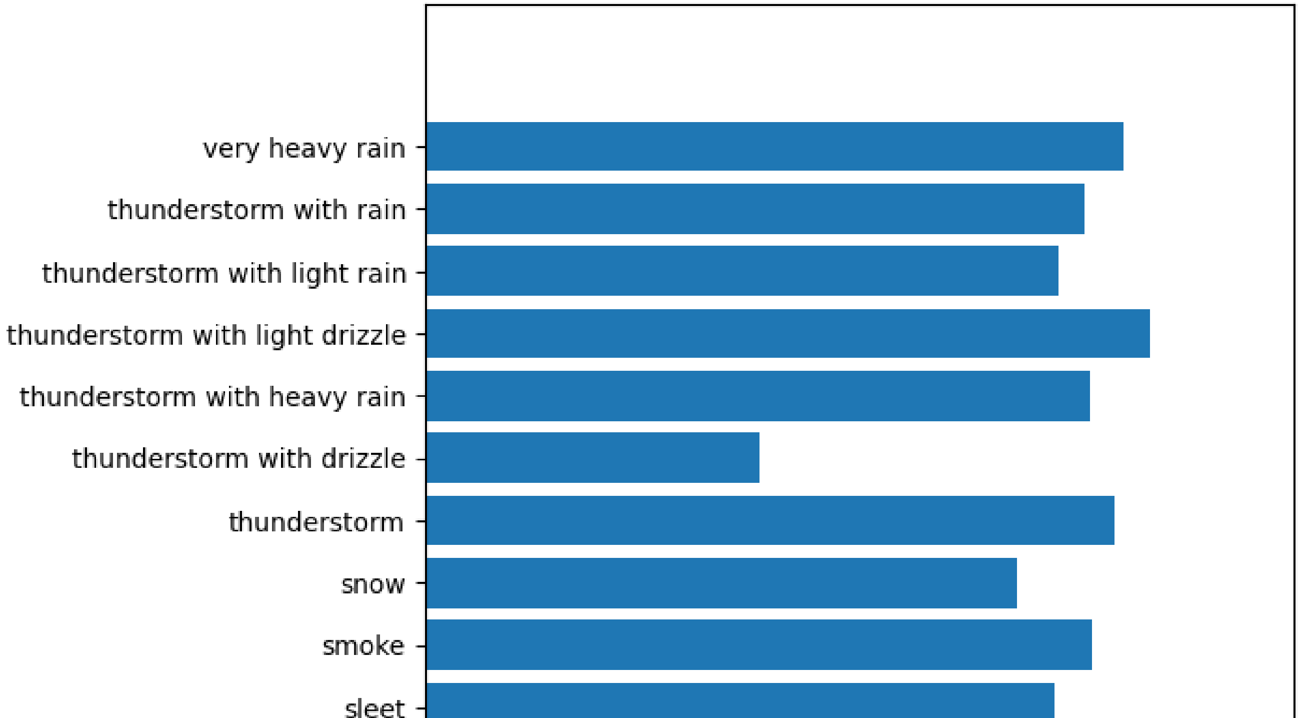
In [41]:
```
by_weather_main['traffic_volume'].sort_values().plot.barh()
plt.show()
```

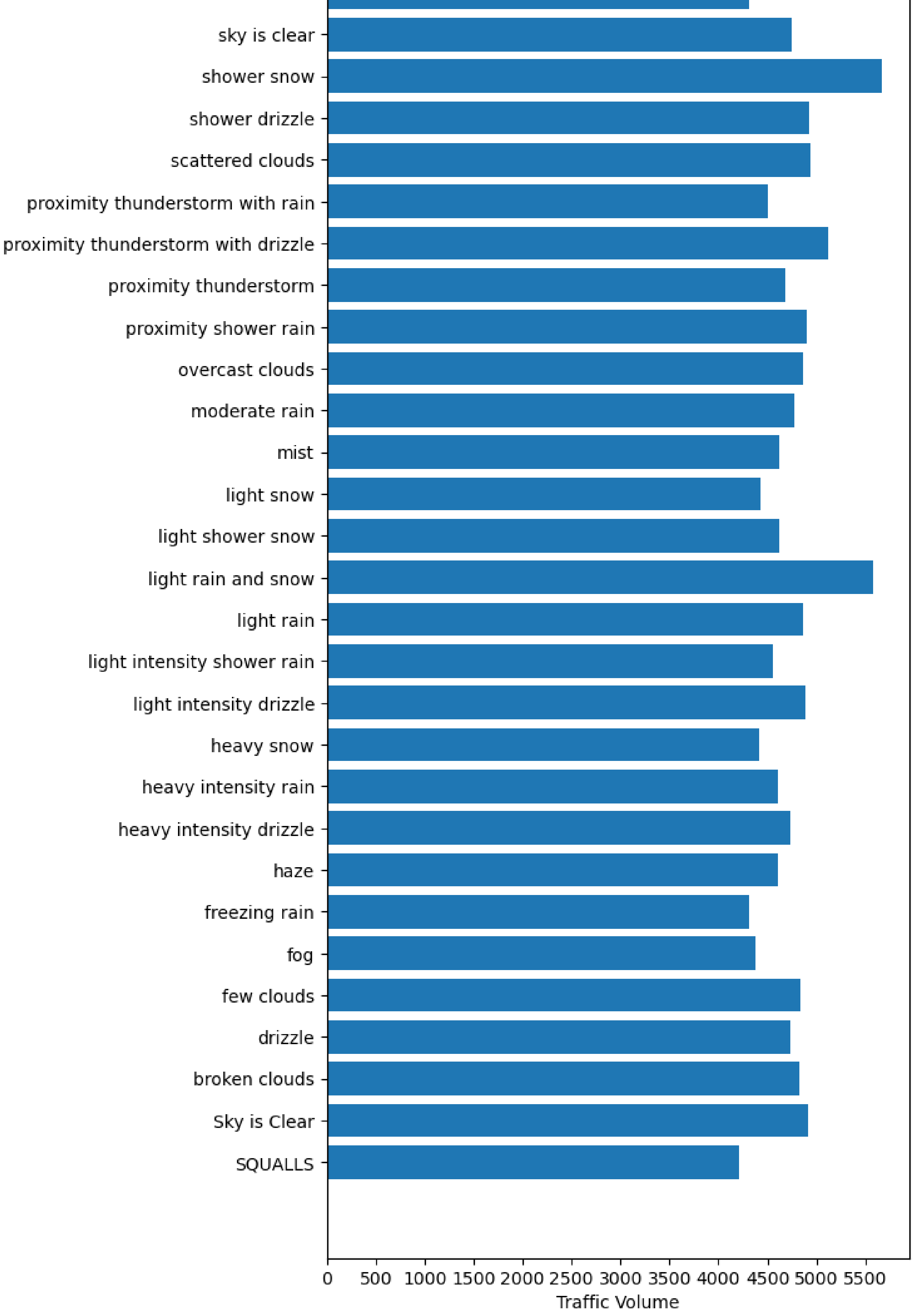


Notes:

According to weather main no weather type exceeds 5,000 in traffic volume. There is no indication of a weather type from weather main causing a substainal increase in traffic. As you can see above clear skies ranks 4th above mist and snow.

In [26]:
```
plt.figure(figsize=(6,18))
plt.barh(by_weather_description.index, by_weather_description['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.locator_params(axis='x', nbins=12)
plt.show()
```





Traffic volume over 5,000 includes: light rain and snow, proximity thunderstorm with drizzle, and shower snow. Weather does not seem to impact traffic volume as one would assume. All levels from clear to heavy thunderstorms see similar volumes of traffic. Weather does not seem to have such a large impact with drivers. There is a better correlation with the time of day and weekend vs weekday.

Nighttime traffic by Month:

In [27]:

```
evening_time['month'] = evening_time['date_time'].dt.month
by_month = evening_time.groupby('month').mean()
by_month['traffic_volume']
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2076194421.py:2: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_month = evening_time.groupby('month').mean()
```

Out[27]:

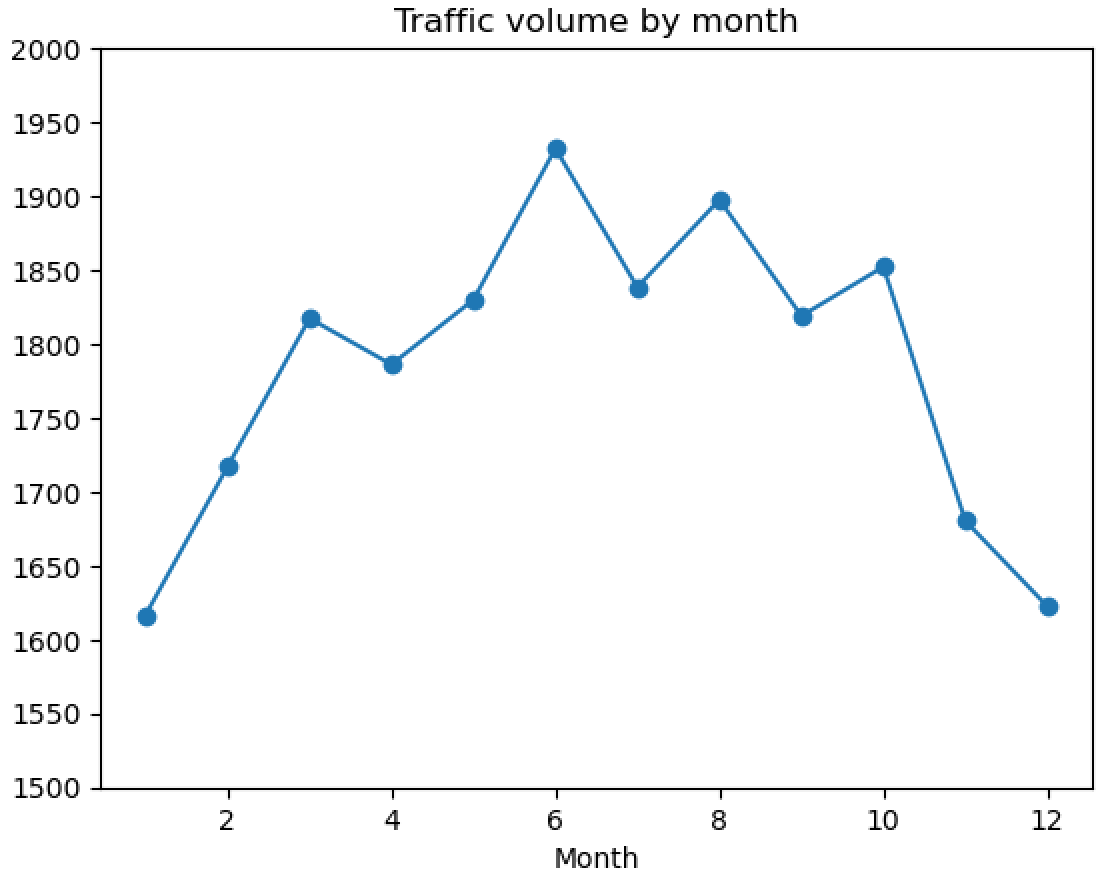
```
month
1     1616.610448
2     1716.961841
3     1817.272029
4     1786.116598
5     1829.852518
6     1932.272727
7     1838.349193
8     1897.564079
9     1818.959858
10    1852.168591
11    1680.311799
12    1622.508393
Name: traffic_volume, dtype: float64
```

In [28]:

```
plt.plot(by_month['traffic_volume'], marker= 'o')
plt.ylim(1500,2000)
plt.locator_params(axis='y', nbins=12)
plt.xlabel('Month')
plt.title('Traffic volume by month')
```

Out[28]:

```
Text(0.5, 1.0, 'Traffic volume by month')
```



Above we can see the peak traffic volume month is June. The summer months of June to August seems to have a higher rate of night traffic.

Nighttime Traffic Volume by day:

In [29]:

```
evening_time['dayofweek'] = evening_time['date_time'].dt.dayofweek
by_dayofweek = evening_time.groupby('dayofweek').mean()
by_dayofweek['traffic_volume']
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2132900014.py:2: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_dayofweek = evening_time.groupby('dayofweek').mean()
```

Out[29]:

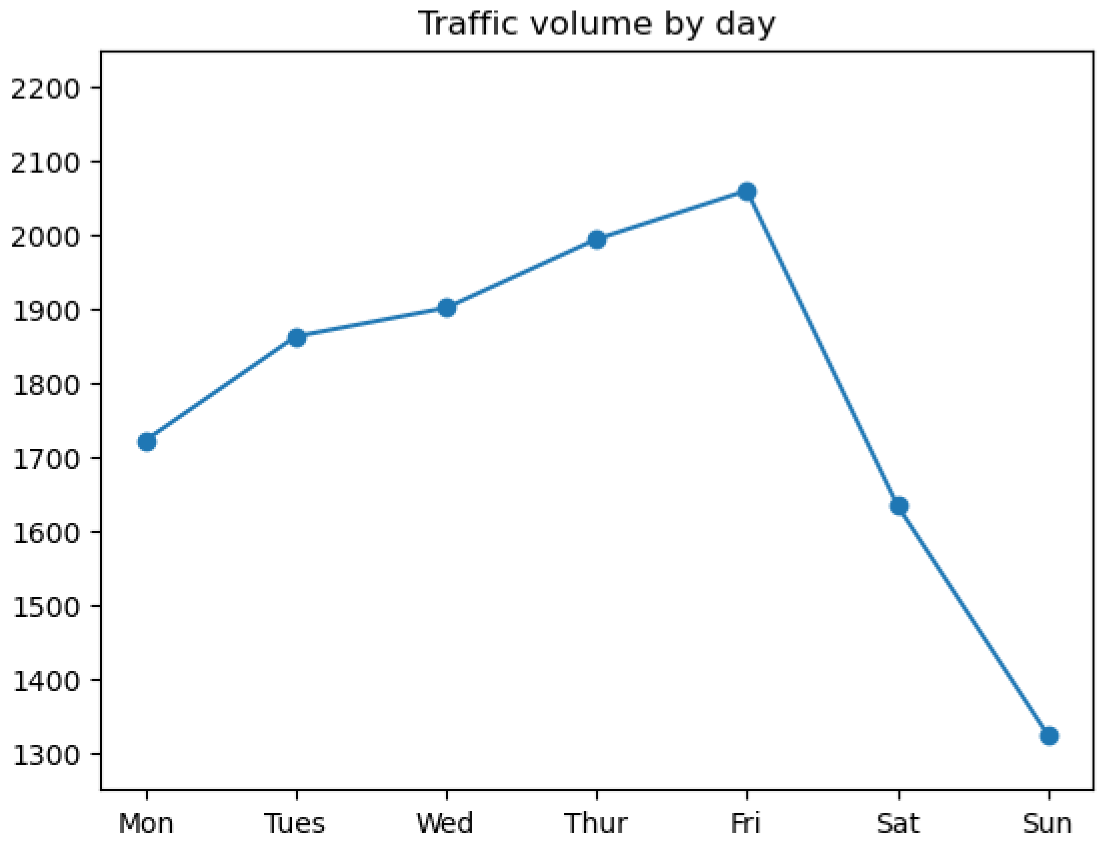
```
dayofweek
0    1722.532692
1    1862.926571
2    1901.465710
3    1994.177959
4    2059.882336
5    1634.459412
6    1323.998273
Name: traffic_volume, dtype: float64
```

In [30]:

```
labels = ['Mon','Tues','Wed','Thur','Fri','Sat','Sun']

plt.plot(by_dayofweek['traffic_volume'], marker= 'o')
plt.ylim(1250,2250)
plt.locator_params(axis='y', nbins=12)
plt.xticks(by_dayofweek.index, labels)
plt.title('Traffic volume by day')
```

Out[30]:

```
Text(0.5, 1.0, 'Traffic volume by day')
```



The traffic at night seems to gradually increase througout the week. The traffic peaks on Friday night. Which is followed by a steep decline Saturday and Sunday night.

```
In [31]:
evening_time['hour'] = evening_time['date_time'].dt.hour
bussiness_days = evening_time.copy()[evening_time['dayofweek'] <= 4] # 4 == Friday
weekend = evening_time.copy()[evening_time['dayofweek'] >= 5] # 5 == Saturday
by_hour_business = bussiness_days.groupby('hour').mean()
by_hour_weekend = weekend.groupby('hour').mean()


print('business day', by_hour_business['traffic_volume'])
print ()
print('weekend',by_hour_weekend['traffic_volume'])
```

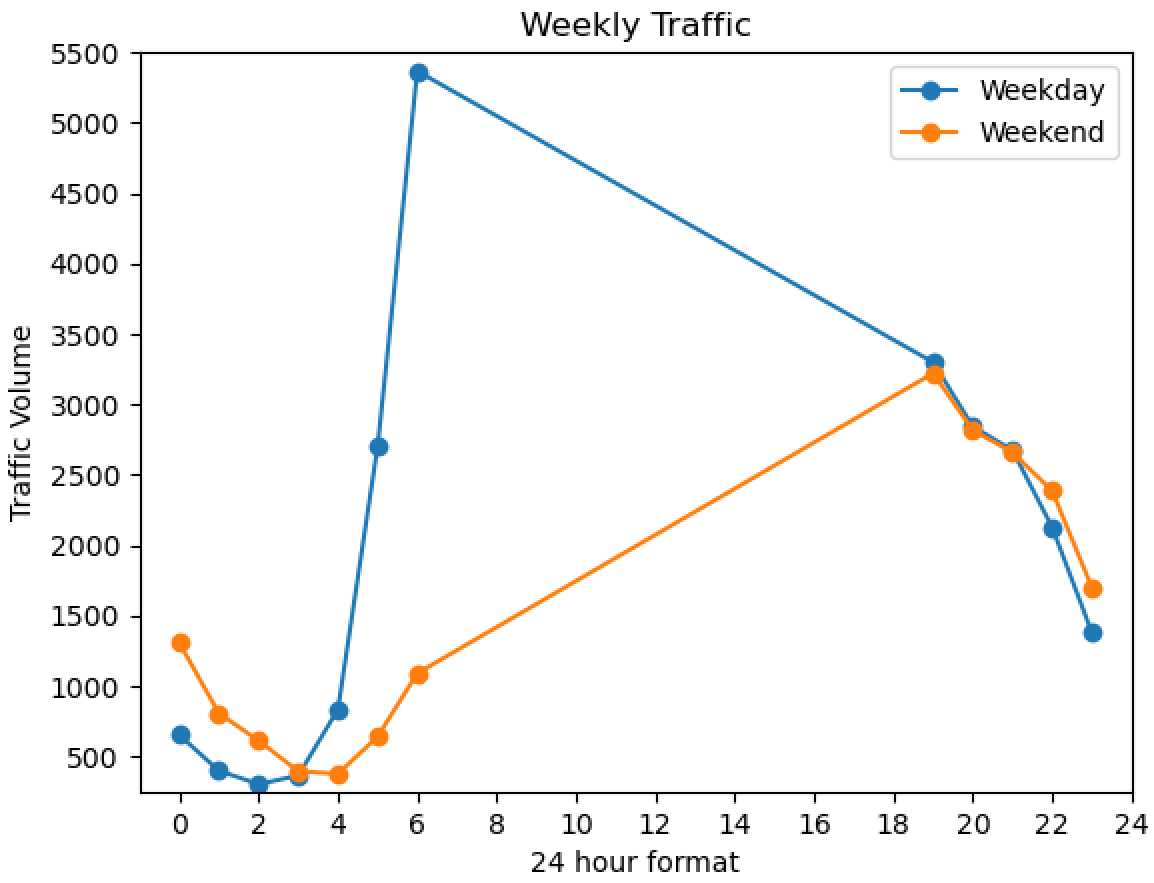
```
business day hour
0      651.528971
1      396.913043
2      301.982818
3      362.289835
4      832.661096
5     2701.296703
6     5365.983210
19    3298.340426
20    2842.433004
21    2673.042807
22    2125.913104
23    1379.549728
Name: traffic_volume, dtype: float64

weekend hour
0     1306.414035
1      805.128333
2      611.171986
3      393.611599
4      375.420168
5      639.237232
6     1089.100334
```

```
19    3220.234120
20    2815.039216
21    2658.445242
22    2384.368607
23    1699.050699
Name: traffic_volume, dtype: float64
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2269941732.py:4: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_hour_business = bussiness_days.groupby('hour').mean()
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\2269941732.py:5: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_hour_weekend = weekend.groupby('hour').mean()
```

In [32]:

```
plt.plot(by_hour_business['traffic_volume'], marker='o', label='Weekday')
plt.xlim(-1, 24)
plt.ylim(250,5500)
plt.locator_params(axis='x', nbins=14)
plt.locator_params(axis='y', nbins=11)
plt.xlabel('24 hour format')
plt.ylabel('Traffic Volume')
plt.title("Week")

plt.plot(by_hour_weekend['traffic_volume'], marker='o',label='Weekend')
plt.xlim(-1, 24)
plt.ylim(250,5500)
plt.locator_params(axis='x', nbins=14)
plt.locator_params(axis='y', nbins=11)
plt.xlabel('24 hour format')
plt.ylabel('Traffic Volume')
plt.title("Weekly Traffic")
plt.legend()
plt.show()
```



The time is cut between 0 = midnight to 6am. Then it goes from 19 = 7pm to 23 = 11pm. Viewing this graph is not perfect. The time should be split further into night and morning hours of offpeak. We will view it in 2 chunks.

Midnight-6am is early morning hours on weekdays:

- From this we can tell traffic curves starting at a max of 1500. Which then decreases to below 500 until 4am.
- At 4am traffic begins to increase to a peak of over 5,000 at 6am.

7pm-11pm on weekdays:

- Traffic starts to decline from 7pm to 11pm.

- The traffic continues to decline through the night and early morning and starts to increase again starting at 4am.

Midnight-6am on weekends:

- From this time slot we can see traffic gradually decreases until 5am.
- From 5 am onward traffic increases.

7pm-11pm on weekends:

- For this time slot we can see that from 7pm to 11pm the traffic consistently declines.

- For the weekend. you can that the traffic declines from 7pm all the way to 4am. At 5pm it starts to increase again.

Weather Indicators for night time:

In [33]:

```
evening_time.loc[:,["temp","rain_1h","snow_1h","clouds_all",
           "traffic_volume"]].corr()["traffic_volume"]
```

Out[33]:

```
temp              0.094004
rain_1h          -0.012972
snow_1h          -0.007453
clouds_all        0.012832
traffic_volume    1.000000
Name: traffic_volume, dtype: float64
```

Notes:

The weather data impact is so low on evening driving. Temperatue again is the highest rating. So I will not bother graphing this data. Lets review the weather types.

In [34]:

```
by_weather_main = evening_time.groupby('weather_main').mean()
by_weather_description = evening_time.groupby('weather_description').mean()

print (by_weather_main)
print ()
print (by_weather_description)
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\232398829.py:1: FutureWarning: The def
ault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_weather_main = evening_time.groupby('weather_main').mean()
```

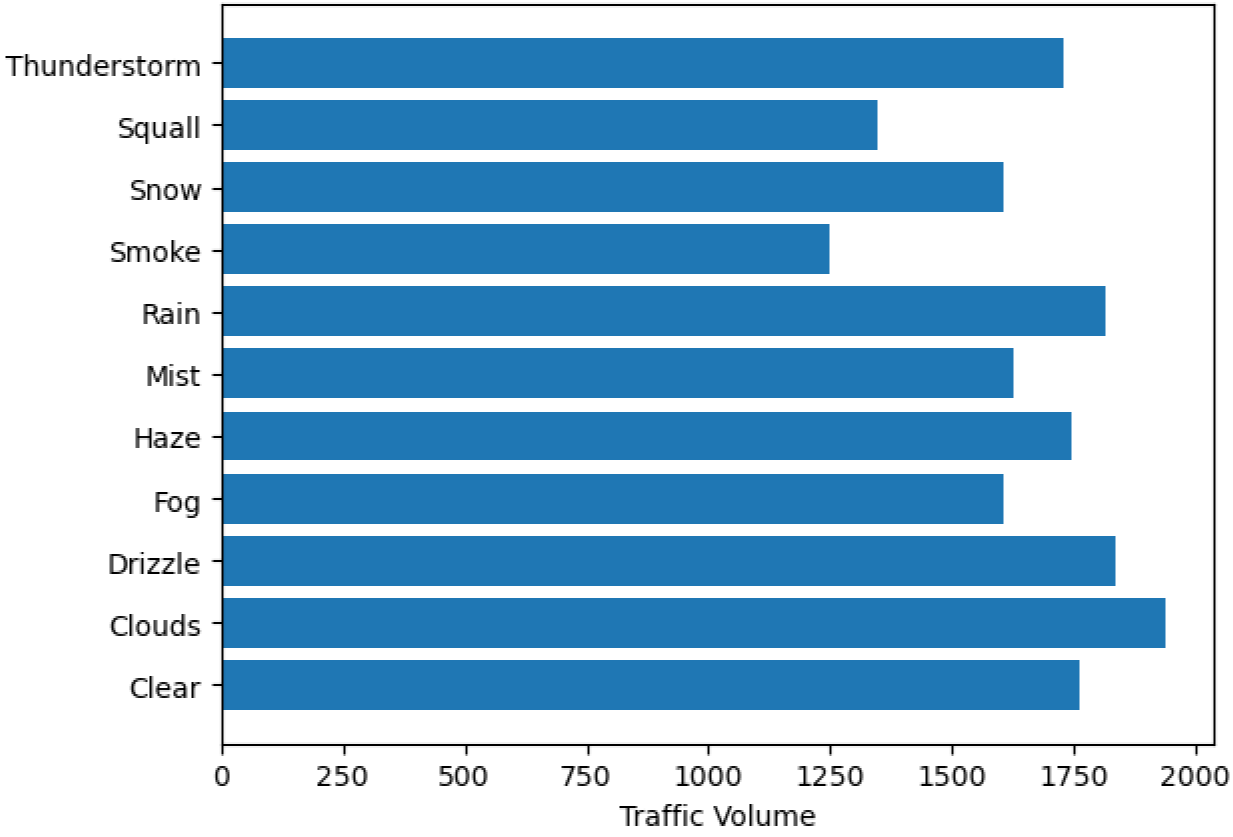
```
                    temp   rain_1h   snow_1h  clouds_all  traffic_volume  \
weather_main
Clear         279.745734  0.000000  0.000000    1.453903     1762.057277
Clouds        279.495731  0.000000  0.000000   65.926029     1939.232745
Drizzle       283.173188  0.145000  0.000000   80.074627     1834.920043
Fog           280.624182  0.036436  0.000109   42.296364     1605.365455
Haze          276.610133  0.057700  0.000000   50.220532     1745.640684
Mist          279.520200  0.229333  0.000652   59.000596     1626.786119
Rain          286.869183  0.583193  0.000081   69.317909     1814.952314
Smoke         288.710000  0.000000  0.000000   53.375000     1247.250000
Snow          267.925211  0.036681  0.001540   82.450774     1606.324191
Squall        290.940000  4.303333  0.000000   76.333333     1345.333333
Thunderstorm  292.214957  1.222333  0.000000   63.356775     1727.842196

                 month  dayofweek       hour
weather_main
Clear         6.488165   3.104747  10.344710
Clouds        6.274838   2.972764  11.869545
Drizzle       6.597015   2.797441  10.636461
Fog           6.727273   3.063636   6.456364
Haze          5.648289   2.914449   9.946768
Mist          6.761692   2.912720   8.273756
Rain          6.815966   2.906747  10.623101
Smoke         5.500000   3.750000   7.500000
Snow          6.182138   2.832630  10.015471
Squall        6.333333   4.333333   8.666667
Thunderstorm  6.886792   3.053173  10.065180

                                          temp    rain_1h   snow_1h  \
weather_description
SQUALLS                             290.940000   4.303333  0.000000
Sky is Clear                        288.472426   0.000000  0.000000
broken clouds                       279.423389   0.000000  0.000000
drizzle                             282.582440   0.099062  0.000000
few clouds                          283.925322   0.000000  0.000000
fog                                 280.624182   0.036436  0.000109
haze                                276.610133   0.057700  0.000000
heavy intensity drizzle             284.825143   0.082286  0.000000
heavy intensity rain                290.411048   2.644395  0.000000
heavy snow                          269.046723   0.000000  0.000000
light intensity drizzle             283.533207   0.182505  0.000000
light intensity shower rain         292.376667   0.000000  0.000000
light rain                          286.320099   0.121465  0.000138
light rain and snow                 273.515000   0.000000  0.000000
light snow                          267.246798   0.051529  0.000961
mist                                279.520200   0.229333  0.000652
moderate rain                       286.746884   0.582924  0.000000
overcast clouds                     275.895058   0.000000  0.000000
proximity shower rain               291.699200   0.021200  0.000000
proximity thunderstorm              292.585378   1.159000  0.000000
proximity thunderstorm with drizzle 287.204286   0.381429  0.000000
proximity thunderstorm with rain    289.660882   0.392647  0.000000
scattered clouds                    283.809416   0.000000  0.000000
shower drizzle                      274.106667   0.000000  0.000000
sky is clear                        278.619626   0.000000  0.000000
smoke                               288.710000   0.000000  0.000000
snow                                269.935192   0.014615  0.008077
thunderstorm                        292.997922   1.382857  0.000000
thunderstorm with heavy rain        291.376053   2.726842  0.000000
thunderstorm with light drizzle     289.315556   1.601111  0.000000
thunderstorm with light rain        292.042581   0.309032  0.000000
thunderstorm with rain              291.502353   1.981176  0.000000
very heavy rain                     287.420909  25.459091  0.000000

                                    clouds_all  traffic_volume     month  \
weather_description
```

```
SQUALLS                               76.333333     1345.333333  6.333333
Sky is Clear                           0.000000     1964.941648  7.596110
broken clouds                         71.767218     1926.904965  6.631073
drizzle                               84.302949     1870.710456  6.410188
few clouds                            19.207872     2076.963100  6.306273
fog                                   42.296364     1605.365455  6.727273
haze                                  50.220532     1745.640684  5.648289
heavy intensity drizzle               86.285714     2238.057143  6.857143
heavy intensity rain                  76.306452     1841.717742  7.221774
heavy snow                            87.182432     1539.354730  5.790541
light intensity drizzle               76.612903     1780.948767  6.713472
light intensity shower rain           73.333333     2747.666667  7.000000
light rain                            64.953153     1800.763363  6.755556
light rain and snow                   82.500000      724.000000  1.000000
light snow                            80.254132     1640.341942  6.334711
mist                                  59.000596     1626.786119  6.761692
moderate rain                         75.629124     1803.143345  6.820250
overcast clouds                       90.181090     1837.845753  5.929487
proximity shower rain                 71.200000     3085.280000  6.040000
proximity thunderstorm                60.070270     1759.718919  6.824324
proximity thunderstorm with drizzle   82.857143     1399.571429  5.714286
proximity thunderstorm with rain      67.147059     1928.617647  6.852941
scattered clouds                      39.754688     2067.496094  6.407031
shower drizzle                        90.000000     2162.666667  6.333333
sky is clear                           1.641518     1735.876716  6.345194
smoke                                 53.375000     1247.250000  5.500000
snow                                  87.102564     1533.621795  6.044872
thunderstorm                          64.064935     1671.207792  6.974026
thunderstorm with heavy rain          73.184211     1364.947368  7.421053
thunderstorm with light drizzle       72.111111     1325.444444  7.000000
thunderstorm with light rain          66.064516     1619.935484  6.806452
thunderstorm with rain                84.529412     2245.176471  7.294118
very heavy rain                       62.727273     1161.363636  8.181818

                                     dayofweek       hour
weather_description
SQUALLS                               4.333333   8.666667
Sky is Clear                          2.885584  10.990847
broken clouds                         2.923118  11.971703
drizzle                               2.898123  10.415550
few clouds                            3.008610  12.765068
fog                                   3.063636   6.456364
haze                                  2.914449   9.946768
heavy intensity drizzle               3.028571   9.885714
heavy intensity rain                  2.633065   9.649194
heavy snow                            3.141892   9.746622
light intensity drizzle               2.709677  10.787476
light intensity shower rain           0.000000  20.000000
light rain                            2.970571  10.635435
light rain and snow                   0.500000  11.500000
light snow                            2.782025   9.744835
mist                                  2.912720   8.273756
moderate rain                         2.866894  10.571104
overcast clouds                       3.005609  11.182292
proximity shower rain                 3.160000  19.760000
proximity thunderstorm                3.072973  10.327027
proximity thunderstorm with drizzle   3.571429   9.142857
proximity thunderstorm with rain      2.470588  10.470588
scattered clouds                      2.958594  12.491406
shower drizzle                        3.000000  20.333333
sky is clear                          3.133028  10.261332
smoke                                 3.750000   7.500000
snow                                  2.589744  12.185897
thunderstorm                          3.129870   8.714286
thunderstorm with heavy rain          2.973684  10.105263
thunderstorm with light drizzle       3.444444   5.444444
thunderstorm with light rain          3.322581  11.612903
thunderstorm with rain                2.705882   9.588235
very heavy rain                       2.818182  11.545455
```

```
C:\Users\marko\AppData\Local\Temp\ipykernel_81624\232398829.py:2: FutureWarning: The def
ault value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only columns w
hich should be valid for the function.
  by_weather_description = evening_time.groupby('weather_description').mean()
```

In [42]:

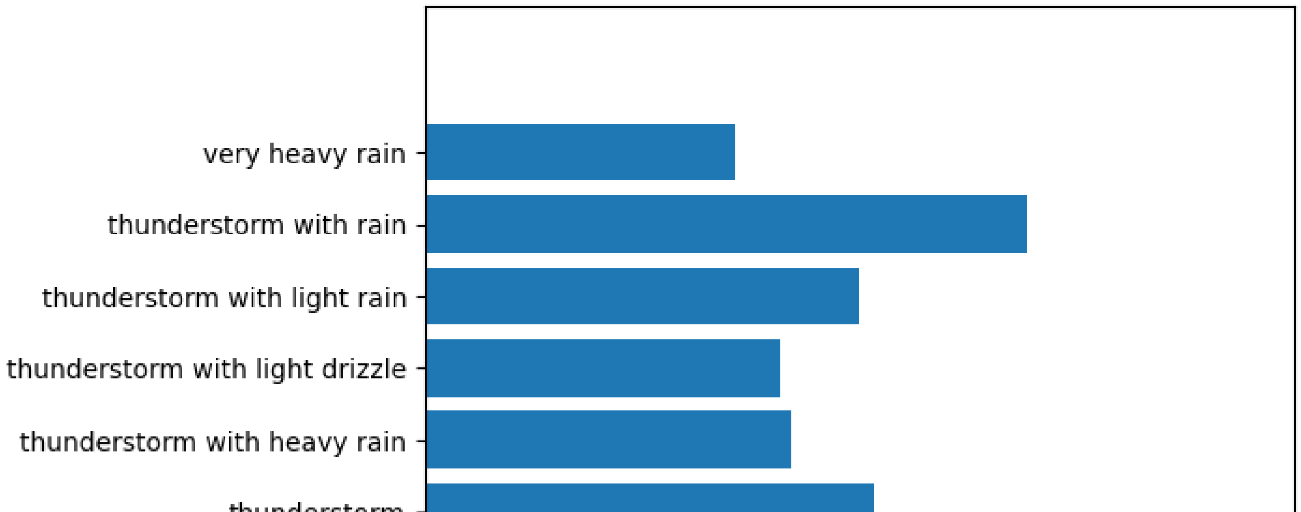
```
plt.barh(by_weather_main.index, by_weather_main['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.show()
```

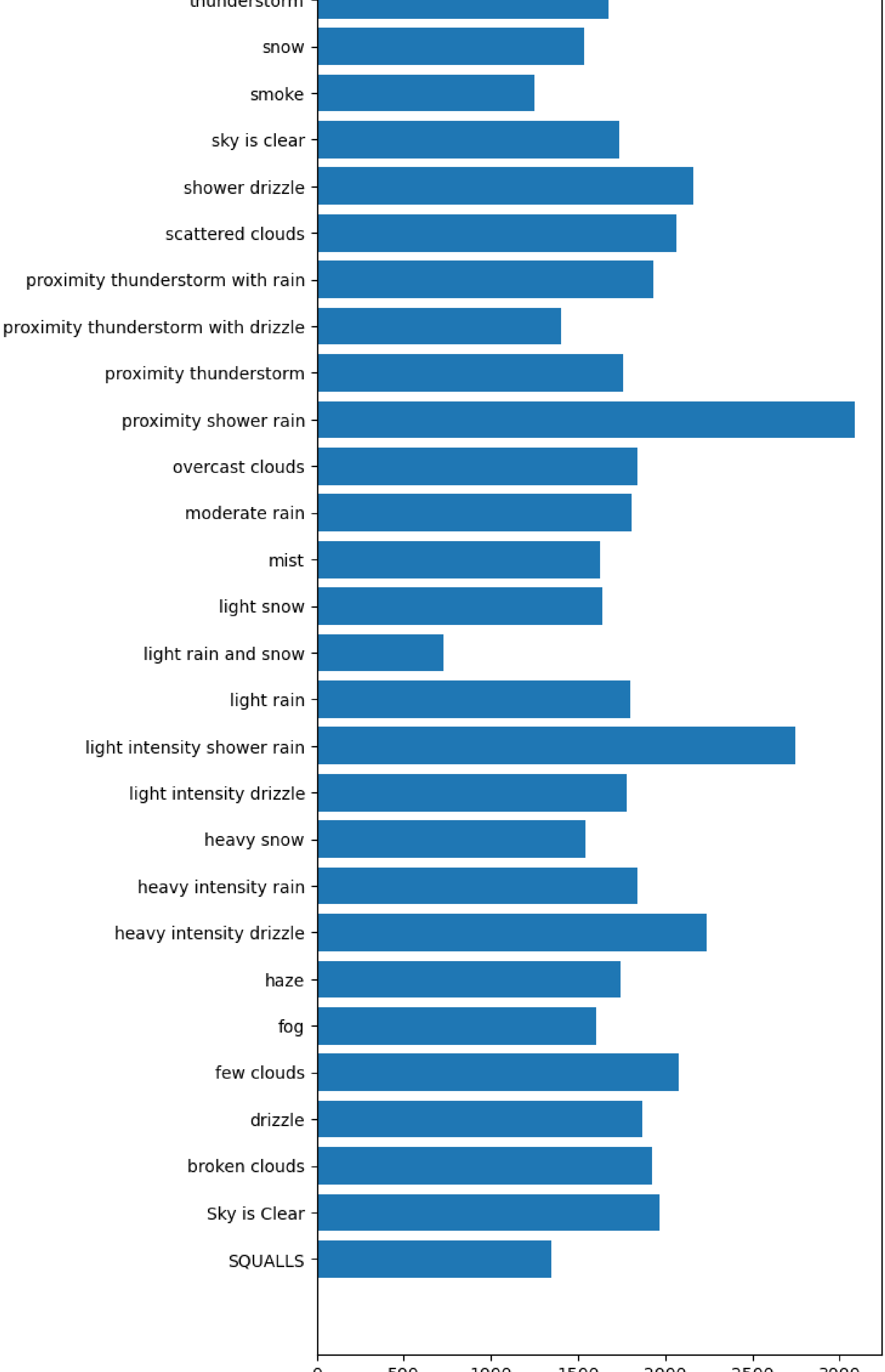


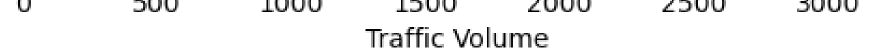
From the above we can see no useful information. lets break it down by description

In [36]:

```
plt.figure(figsize=(6,18))
plt.barh(by_weather_description.index, by_weather_description['traffic_volume'])
plt.xlabel('Traffic Volume')
plt.locator_params(axis='x', nbins=12)
plt.show()
```




thunderstorm
snow
smoke
sky is clear
shower drizzle
scattered clouds
proximity thunderstorm with rain
proximity thunderstorm with drizzle
proximity thunderstorm
proximity shower rain
overcast clouds
moderate rain
mist
light snow
light rain and snow
light rain
light intensity shower rain
light intensity drizzle
heavy snow
heavy intensity rain
heavy intensity drizzle
haze
fog
few clouds
drizzle
broken clouds
Sky is Clear
SQUALLS
0
500
1000
1500
2000
2500
3000



From the information above we can see that high intensity drizzle, light intensity shower, proximity shower, thunder storm have the highest traffic volume. This is probably an indication of slow down in traffic because of unexpected weather change.