

```
import numpy as np
import pandas as pd

In [2]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
matplotlib.rcParams['font.family'] = 'sans-serif'

In [3]: df = pd.read_csv('911.csv')

In [4]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99492 entries, 0 to 99491
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  --
 0   lat         99492 non-null    float64
 1   lng         99492 non-null    float64
 2   desc        99492 non-null    object
 3   zip         99492 non-null    object
 4   title       99492 non-null    object
 5   timeStamp   99492 non-null    object
 6   twp         99492 non-null    object
 7   addr        99492 non-null    object
 8   e           99492 non-null    int64
dtypes: float64(2), int64(1), object(6)
memory usage: 6.84 MB


In [5]: df.head(3)
Out[5]:
   lat      lng      desc      zip      title      timeStamp      twp      addr  e
0  40.297876 -75.581294  REINDEER CT & DEAD END, NEW HANOVER, Station ...  19525.0  EMS: BACK PAINS/INJURY  2015-12-10 17:40:00  NEW HANOVER  REINDEER CT & DEAD END  1
1  40.298065 -75.264880  BRIAR PATH & WHITE MARSH LN, HATFIELD TOWNSHIP...  19446.0  EMS: DIABETIC EMERGENCY  2015-12-10 17:40:00  HATFIELD TOWNSHIP  BRIAR PATH & WHITE MARSH LN  1
2  40.121182 -75.351875  HAWES AVE, NORRISTOWN, 2015-12-10 @ 14:39:21 St...  19401.0  Fire  GAS-ODOR/LEAK  2015-12-10 17:40:00  NORRISTOWN  HAWES AVE  1

In [6]: df['zip'].value_counts().head(5)
Out[6]:
19401.0    6979
19446.0    6643
19403.0    4854
19446.0    4748
19446.0    3174
Name: zip, dtype: int64

In [7]: df['twp'].value_counts().head(5)
Out[7]:
LOWER MERION  8443
ABINGDON     5977
WOBURSTON    5950
UPPER MERION  5227
WELLSBORO    4575
Name: twp, dtype: int64

In [8]: df['title'].nunique()
Out[8]: 119

In [9]: df['Reason'] = df['title'].apply(lambda title: title.split(' ')[0])
In [10]: df['Reason'].value_counts()
Out[10]:
EMS      48877
Traffic  35693
Fire     14920
Name: Reason, dtype: int64

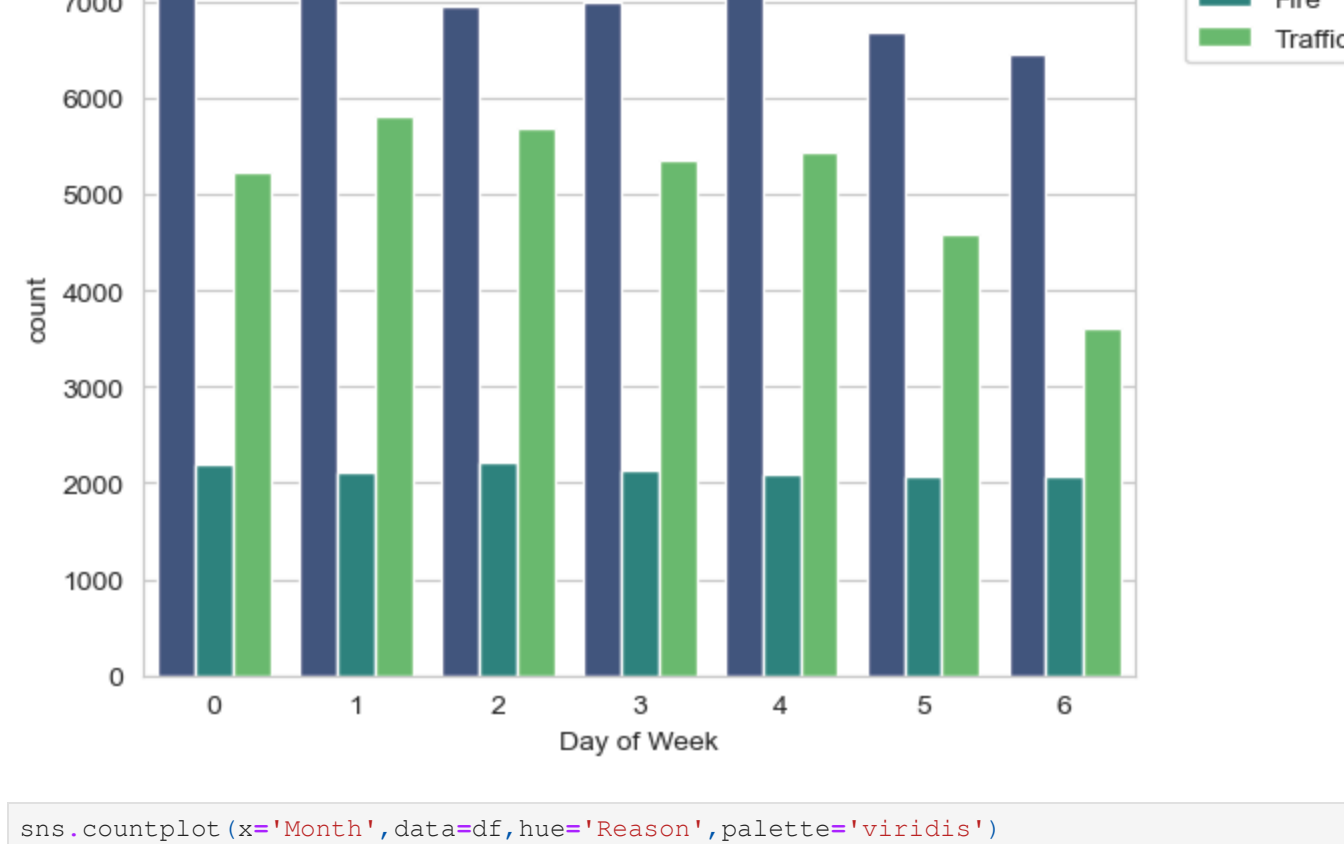
In [11]: sns.countplot(x='Reason', data=df, palette='viridis')
Out[11]: <Axes: xlabel='Reason', ylabel='count'>



In [12]: type(df['timeStamp'].iloc[0])
Out[12]: str

In [13]: df['timeStamp'] = pd.to_datetime(df['timeStamp'])


In [14]: df['hour'] = df['timeStamp'].apply(lambda time: time.hour)
df['month'] = df['timeStamp'].apply(lambda time: time.month)
df['day of week'] = df['timeStamp'].apply(lambda time: time.dayofweek)

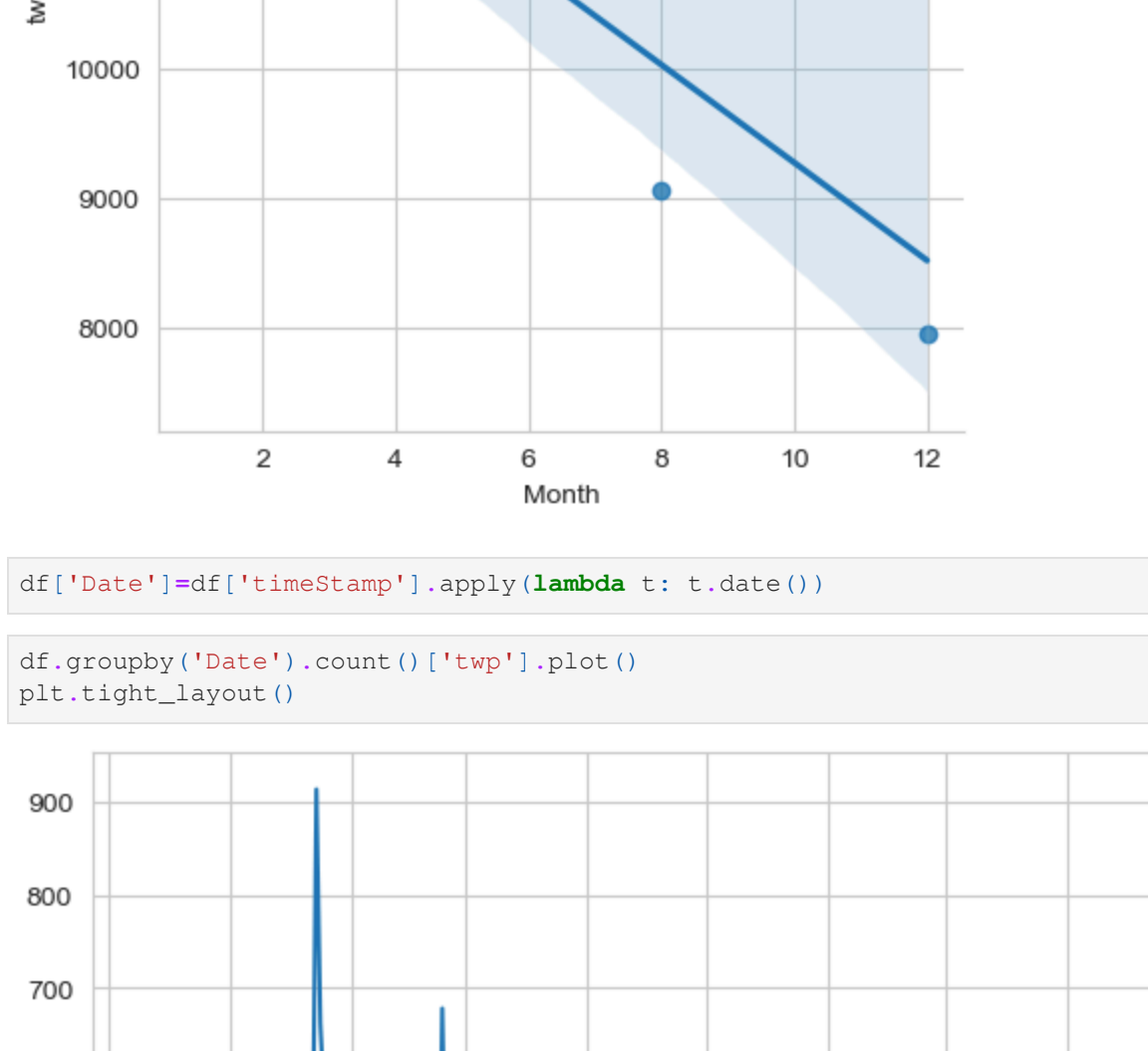
In [15]: dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}

In [16]: sns.countplot(x='Day of Week', data=df, hue='Reason', palette='viridis')
# To relocate the legend
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
Out[16]: <matplotlib.legend.Legend at 0x25a24fc4880>


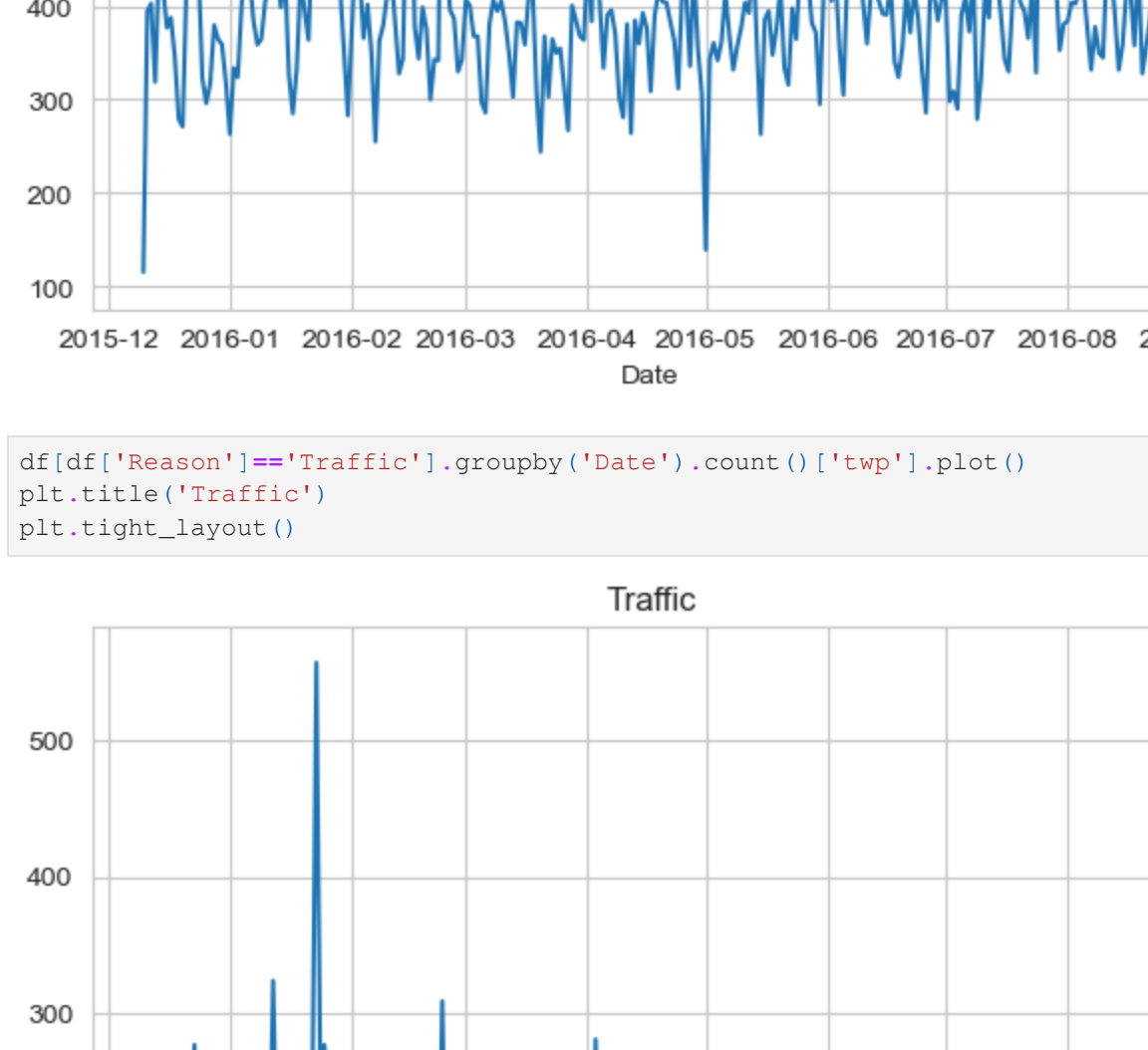
In [17]: sns.countplot(x='Month', data=df, hue='Reason', palette='viridis')
# To relocate the legend
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
Out[17]: <matplotlib.legend.Legend at 0x25a24fc4880>


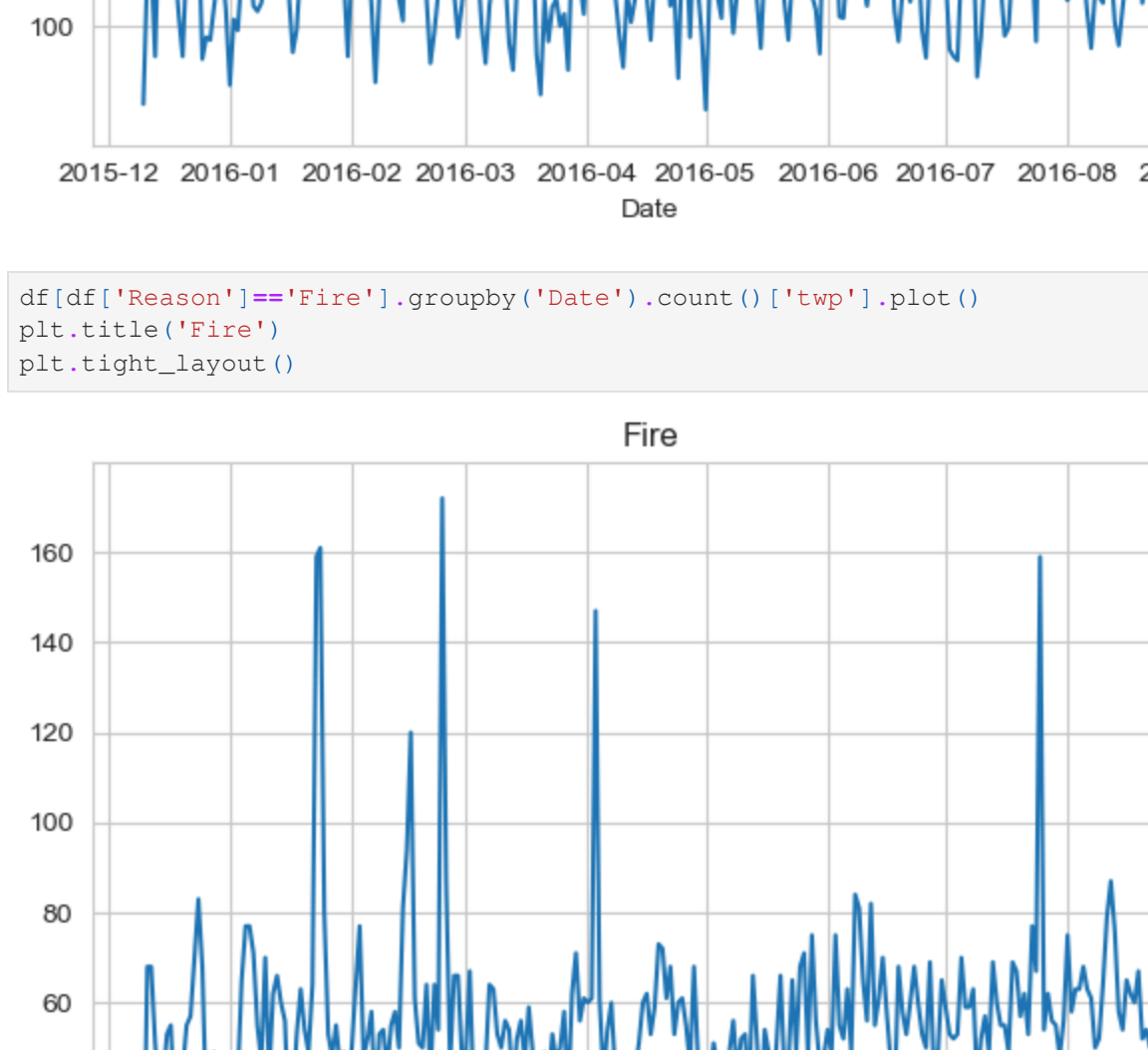
In [18]: byMonth = df.groupby('Month').count()
byMonth.head()
Out[18]:
   Month
1  12036 12036 12036 11577 12035  12036 12020 12096 12035 12036 12036 12036
2  11467 11467 11467  9930 11467  11467 11466 11396 11467 11467 11467 11467
3  11101 11101 11101  9786 11101  11101 11092 11099 11101 11101 11101 11101
4  11326 11326 11326  9896 11326  11326 11323 11283 11326 11326 11326 11326
5  11423 11423 11423  9946 11423  11423 11420 11378 11423 11423 11423 11423

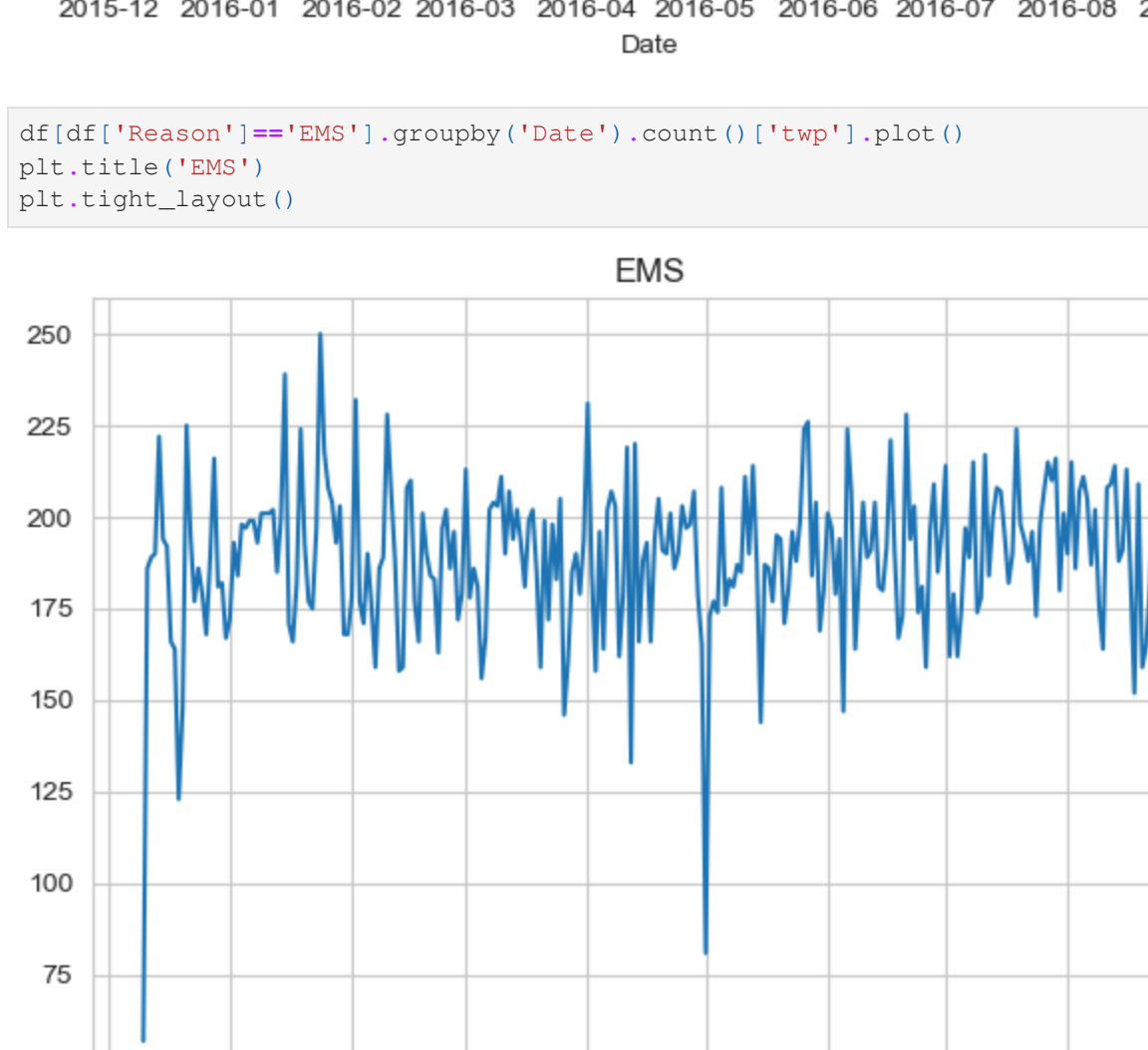
In [19]: byMonth['twp'].plot()
Out[19]: <Axes: xlabel='Month'>



In [20]: sns.lmplot(x='Month', y='twp', data=byMonth.reset_index())
Out[20]: <seaborn.axisgrid.FacetGrid at 0x25a24fc0c60>


In [21]: df['Date'] = df['timeStamp'].apply(lambda t: t.date())

In [22]: df.groupby('Date').count()['twp'].plot()
plt.tight_layout()



In [23]: df[df['Reason']=='Traffic'].groupby('Date').count()['twp'].plot()
plt.title('Traffic')
plt.tight_layout()


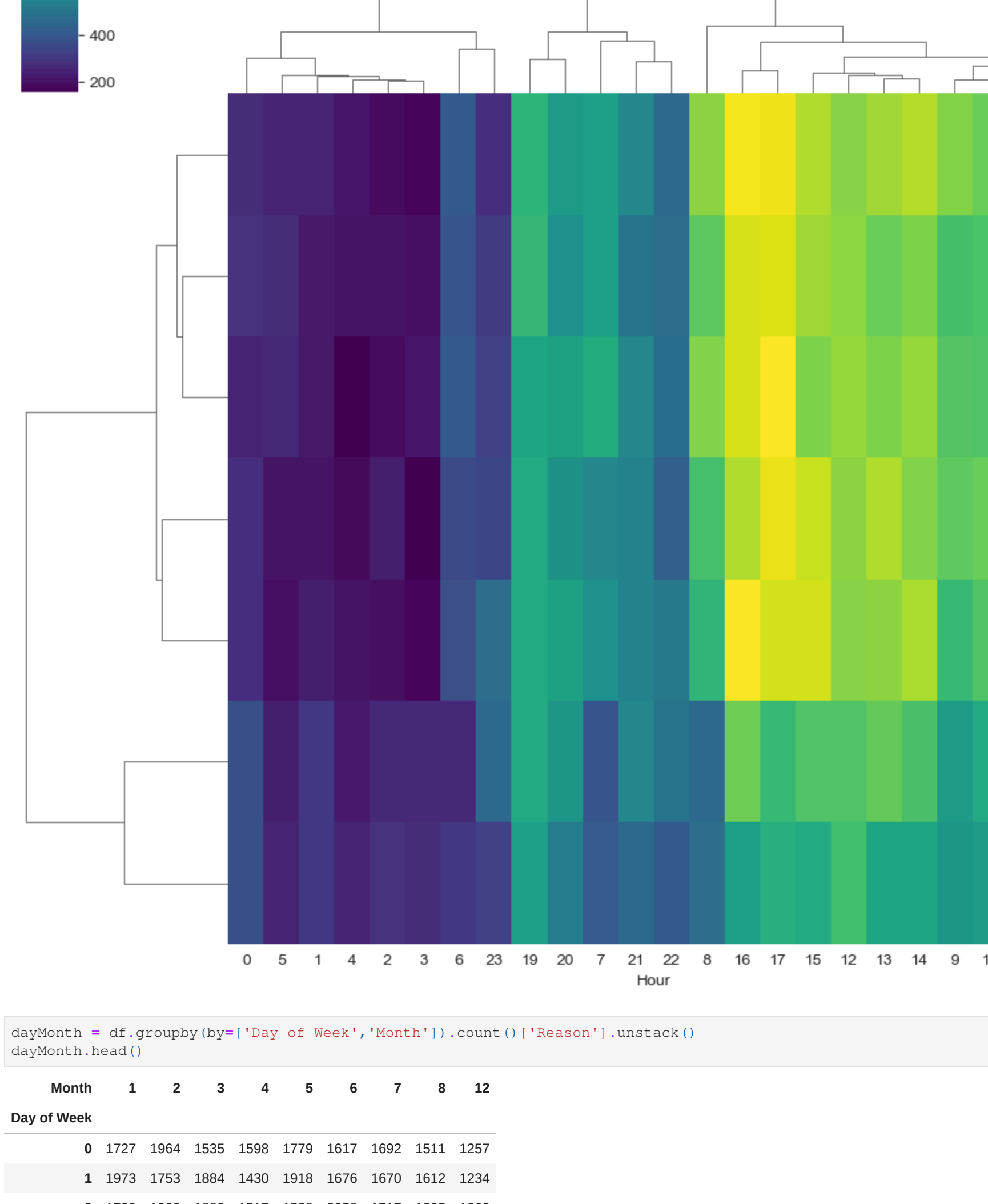
In [24]: df[df['Reason']=='Fire'].groupby('Date').count()['twp'].plot()
plt.title('Fire')
plt.tight_layout()


In [25]: df[df['Reason']=='EMS'].groupby('Date').count()['twp'].plot()
plt.title('EMS')
plt.tight_layout()



In [26]: dayHour = df.groupby(['Day of Week', 'Hour']).count()['Reason'].unstack()
dayHour.head()
Out[26]:
   Day of Week
Hour  0  1  2  3  4  5  6  7  8  9  ...  14  15  16  17  18  19  20  21  22  23
0  203  221  201  194  204  267  307  653  919  705  ...  869  919  969  997  805  746  613  497  472  325
1  289  340  186  170  209  239  415  656  889  880  ...  943  938  1026  1019  905  731  647  571  462  274
2  250  216  189  209  156  155  410  701  876  808  ...  904  867  990  1037  894  686  666  576  490  335
3  278  202  233  159  182  203  362  570  777  826  ...  876  969  935  1013  810  698  617  553  424  354
4  275  235  191  175  201  184  372  598  742  752  ...  932  990  1089  980  820  696  667  559  514  474

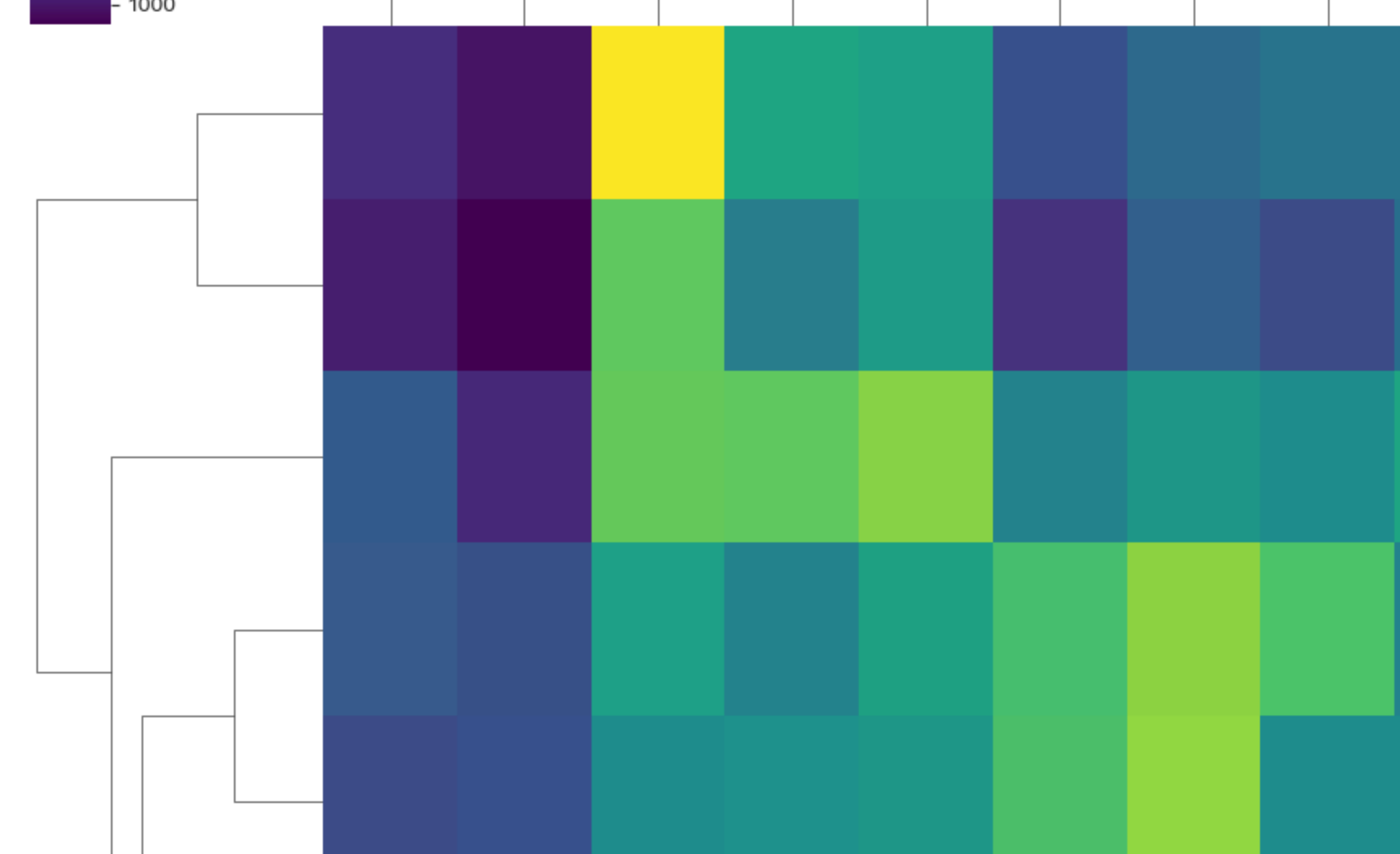
5 rows x 24 columns

In [27]: plt.figure(figsize=(12,6))
sns.heatmap(dayHour, cmap='viridis')
Out[27]: <Axes: xlabel='Hour', ylabel='Day of Week'>


In [28]: sns.clustermap(dayHour, cmap='viridis')
Out[28]: <seaborn.matrix.ClusterGrid at 0x25a27764310>


In [29]: dayMonth = df.groupby(['Day of Week', 'Month']).count()['Reason'].unstack()
dayMonth.head()
Out[29]:
   Day of Week
Month  1  2  3  4  5  6  7  8  12
0  1727  1964  1535  1598  1779  1617  1682  1611  1267
1  1973  1763  1864  1430  1918  1676  1670  1612  1234
2  1700  1903  1889  1517  1538  2058  1717  1295  1263
3  1584  1596  1800  1601  1591  2005  1646  2230  1266
4  1970  1581  1525  1958  1730  1649  2045  1310  1066

In [30]: plt.figure(figsize=(12,6))
sns.heatmap(dayMonth, cmap='viridis')
Out[30]: <Axes: xlabel='Month', ylabel='Day of Week'>


In [31]: sns.clustermap(dayMonth, cmap='viridis')
Out[31]: <seaborn.matrix.ClusterGrid at 0x25a27956a00>

```