

911 Calls Final Project

I will be analyzing some 911 call data from [Kaggle](#). The data contains the following fields:

- lat : String variable, Latitude
- lng: String variable, Longitude
- desc: String variable, Description of the Emergency Call
- zip: String variable, Zipcode
- title: String variable, Title
- timeStamp: String variable, YYYY-MM-DD HH:MM:SS
- twp: String variable, Township
- addr: String variable, Address
- e: String variable, Dummy variable (always 1)

Data and Setup

In [24]:

```
import numpy as np
import pandas as pd
```

In [25]:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline
```

In [26]:

```
df = pd.read_csv('911.csv')
```

In [27]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99492 entries, 0 to 99491
Data columns (total 9 columns):
lat                99492 non-null float64
lng                99492 non-null float64
desc               99492 non-null object
zip                86637 non-null float64
title              99492 non-null object
timeStamp          99492 non-null object
twp                99449 non-null object
addr               98973 non-null object
e                  99492 non-null int64
dtypes: float64(3), int64(1), object(5)
memory usage: 6.8+ MB
```

In [28]:

```
df.head(3)
```

Out[28]:

	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	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	2015-12-10 17:40:00	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH ...	1

	lat	lng	desc	zip	title	timeStamp	twp	LN addr	e
2	40.121182	75.351975	HAWS AVE, NORRISTOWN, 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	2015-12-10 17:40:00	NORRISTOWN	HAWS AVE	1

Basic Questions

In [29]:

```
df['zip'].value_counts().head(5)
```

Out[29]:

```
19401.0    6979
19464.0    6643
19403.0    4854
19446.0    4748
19406.0    3174
Name: zip, dtype: int64
```

In [30]:

```
df['twp'].value_counts().head(5)
```

Out[30]:

```
LOWER MERION    8443
ABINGTON       5977
NORRISTOWN     5890
UPPER MERION   5227
CHELTENHAM     4575
Name: twp, dtype: int64
```

In [31]:

```
df['title'].nunique()
```

Out[31]:

```
110
```

Creating new features

In [32]:

```
df['Reason'] = df['title'].apply(lambda title: title.split(':')[0])
```

In [33]:

```
df['Reason'].value_counts()
```

Out[33]:

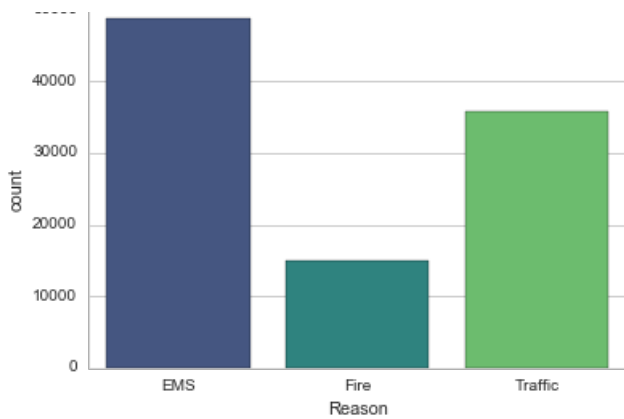
```
EMS          48877
Traffic      35695
Fire         14920
Name: Reason, dtype: int64
```

In [34]:

```
sns.countplot(x='Reason', data=df, palette='viridis')
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x121757b70>
```



In [35]:

```
type(df['timeStamp'].iloc[0])
```

Out[35]:

str

In [36]:

```
df['timeStamp'] = pd.to_datetime(df['timeStamp'])
```

In [37]:

```
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 [38]:

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

In [39]:

```
df['Day of Week'] = df['Day of Week'].map(dmap)
```

In [40]:

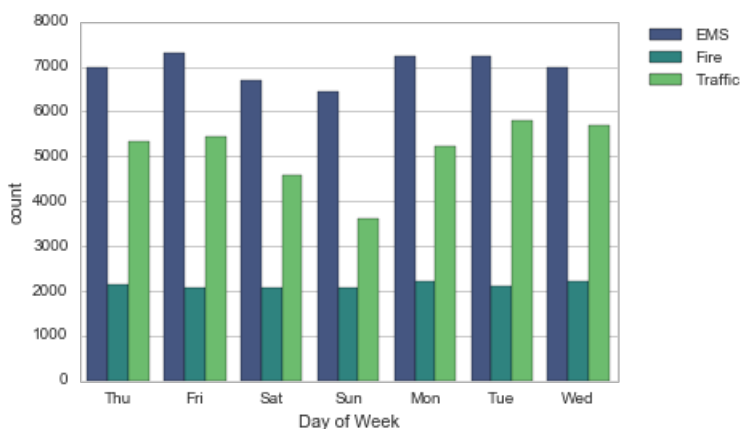
```
sns.countplot(x='Day of Week',data=df,hue='Reason',palette='viridis')
```

To relocate the legend

```
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

Out[40]:

<matplotlib.legend.Legend at 0x121762710>



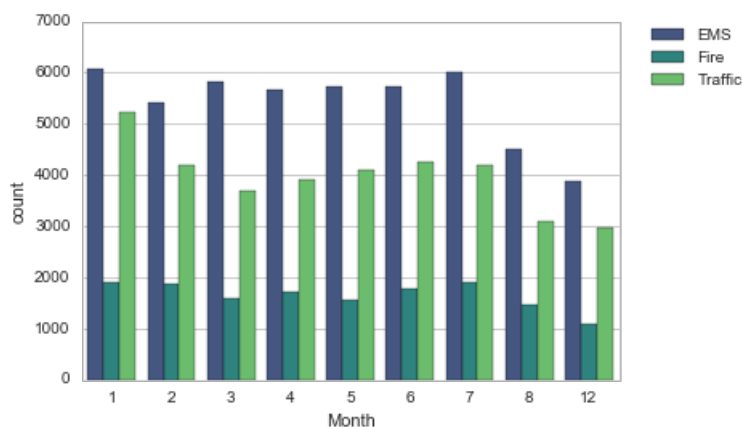
In [41]:

```
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[41]:

<matplotlib.legend.Legend at 0x11fa7ad68>



In [43]:

```
byMonth = df.groupby('Month').count()
byMonth.head()
```

Out[43]:

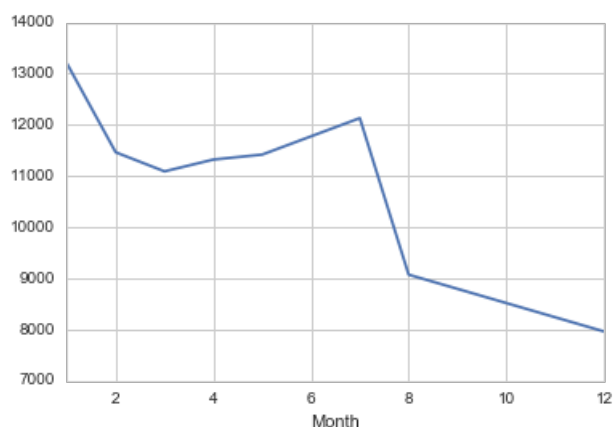
	lat	lng	desc	zip	title	timeStamp	twp	addr	e	Reason	Hour	Day of Week
Month												
1	13205	13205	13205	11527	13205	13205	13203	13096	13205	13205	13205	13205
2	11467	11467	11467	9930	11467	11467	11465	11396	11467	11467	11467	11467
3	11101	11101	11101	9755	11101	11101	11092	11059	11101	11101	11101	11101
4	11326	11326	11326	9895	11326	11326	11323	11283	11326	11326	11326	11326
5	11423	11423	11423	9946	11423	11423	11420	11378	11423	11423	11423	11423

In [44]:

```
byMonth['twp'].plot()
```

Out[44]:

<matplotlib.axes._subplots.AxesSubplot at 0x11fa06630>

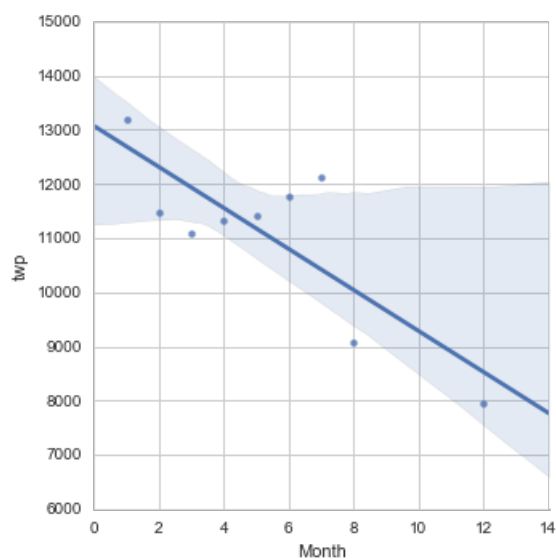


In [45]:

```
sns.lmplot(x='Month',y='twp',data=byMonth.reset_index())
```

Out[45]:

<seaborn.axisgrid.FacetGrid at 0x11bf002b0>

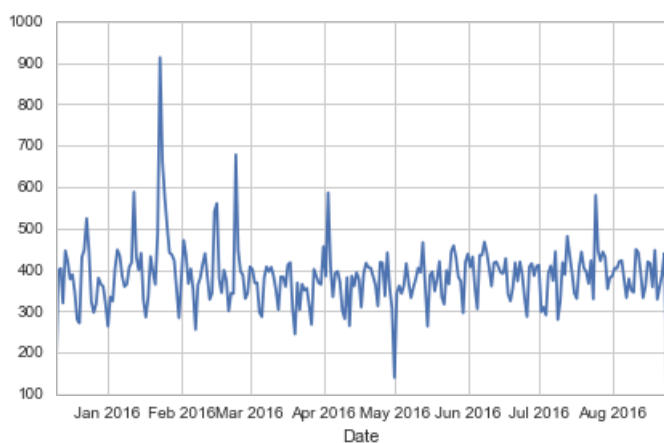


In [46]:

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

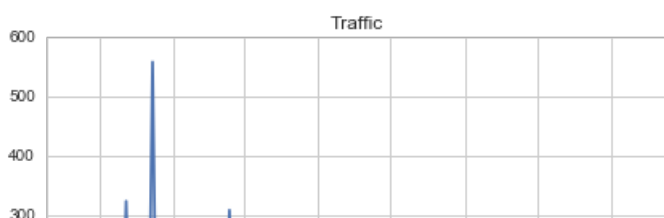
In [47]:

```
df.groupby('Date').count()['twp'].plot()  
plt.tight_layout()
```



In [48]:

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



Sun Hour	383	306	286	268	242	240	300	402	483	620	...	684	691	663	714	670	655	537	461	415	330
Thu	278	202	233	159	182	203	362	570	777	828	...	876	969	935	1013	810	698	617	553	424	354
Day of Week																					

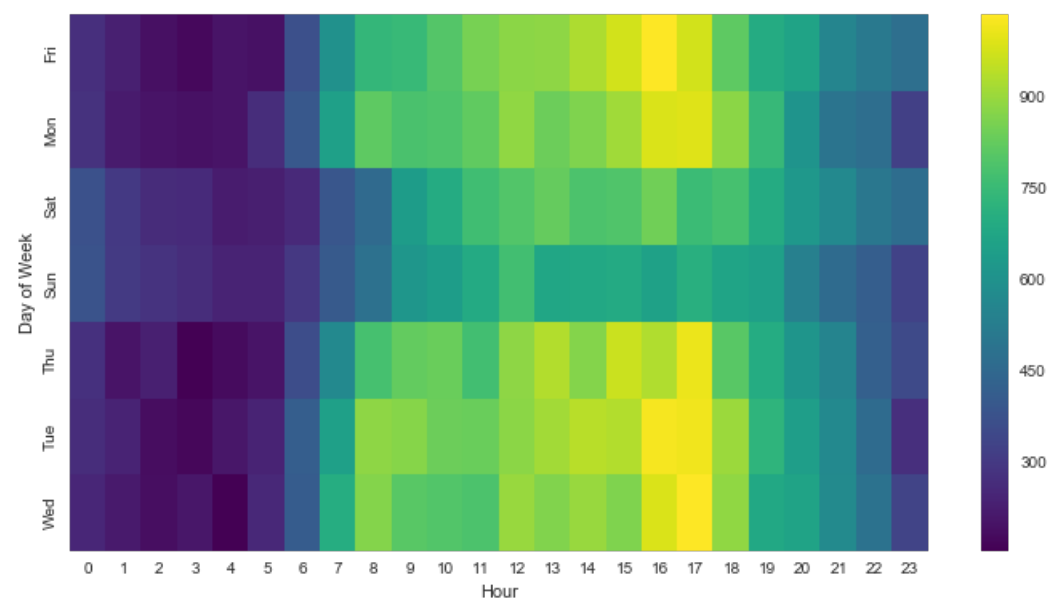
5 rows × 24 columns

```
In [52]:
```

```
plt.figure(figsize=(12,6))
sns.heatmap(dayHour,cmap='viridis')
```

```
Out[52]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x12305acf8>
```

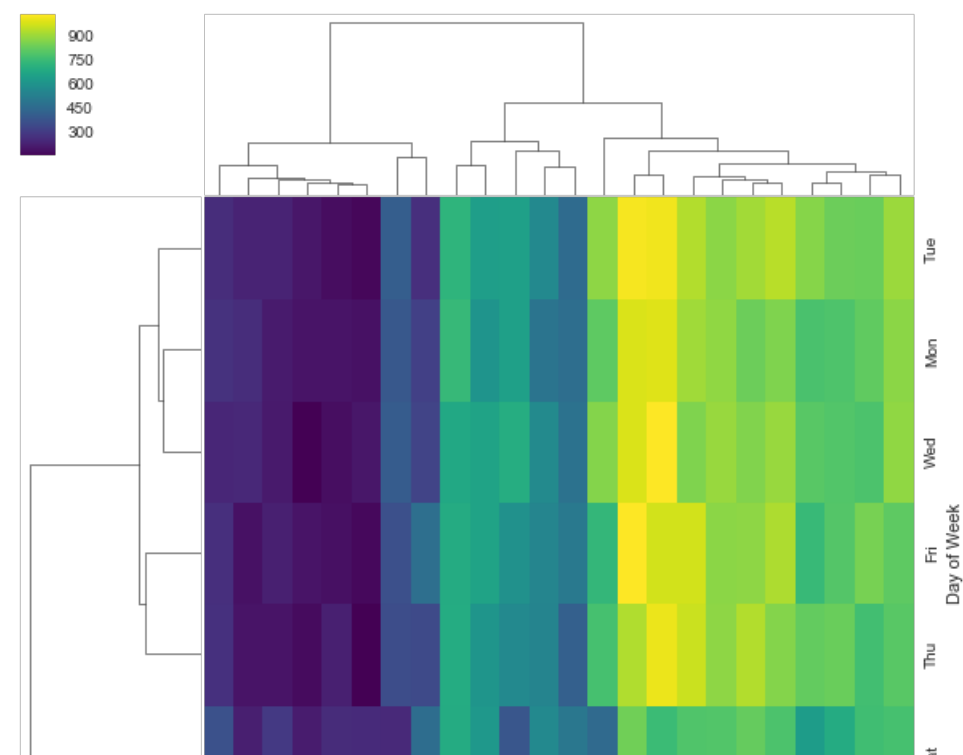


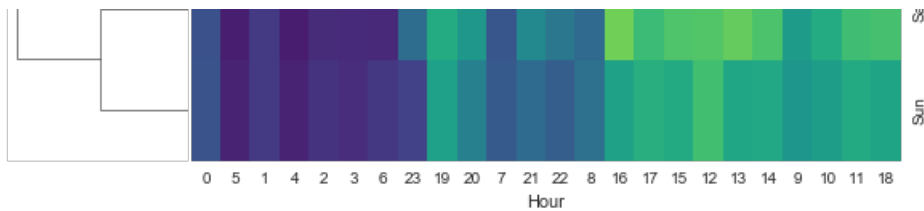
```
In [53]:
```

```
sns.clustermap(dayHour,cmap='viridis')
```

```
Out[53]:
```

```
<seaborn.matrix.ClusterGrid at 0x103276748>
```





In [54]:

```
dayMonth = df.groupby(by=['Day of Week', 'Month']).count()['Reason'].unstack()
dayMonth.head()
```

Out[54]:

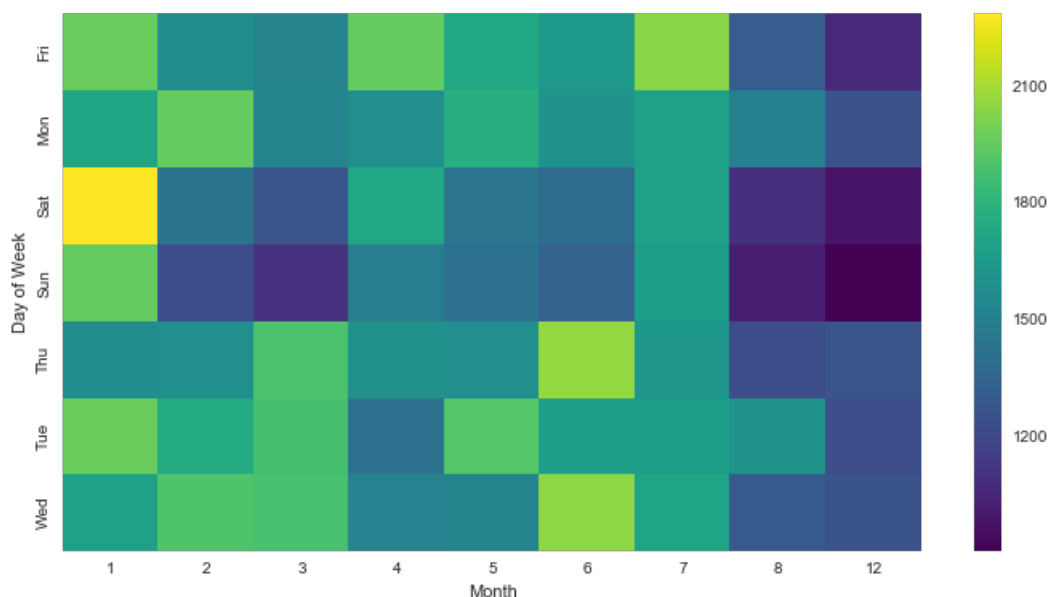
Month	1	2	3	4	5	6	7	8	12
Day of Week									
Fri	1970	1581	1525	1958	1730	1649	2045	1310	1065
Mon	1727	1964	1535	1598	1779	1617	1692	1511	1257
Sat	2291	1441	1266	1734	1444	1388	1695	1099	978
Sun	1960	1229	1102	1488	1424	1333	1672	1021	907
Thu	1584	1596	1900	1601	1590	2065	1646	1230	1266

In [55]:

```
plt.figure(figsize=(12,6))
sns.heatmap(dayMonth,cmap='viridis')
```

Out[55]:

<matplotlib.axes._subplots.AxesSubplot at 0x11bcabf98>



In [56]:

```
sns.clustermap(dayMonth,cmap='viridis')
```

Out[56]:

<seaborn.matrix.ClusterGrid at 0x120341e80>

