

911 CALLS PROJECT

In this project, dataset consisting of 911 (emergency calls) made from different locations, zipcodes and reasons, departments is taken 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)

The data is analysed and visualised by applying different conditions and outcomes are produced.

Setup - Importing Libraries

```
In [1]: # analysis libraries
import numpy as np
import pandas as pd
```

```
In [2]: # visualisation libraries
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data Setup

```
In [3]: #importing dataset
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         86637 non-null   float64 
 4   title       99492 non-null   object  
 5   timeStamp   99492 non-null   object  
 6   twp         99449 non-null   object  
 7   addr        98973 non-null   object  
 8   e           99492 non-null   int64  
dtypes: float64(3), int64(1), object(5)
memory usage: 6.8+ MB
```

In [6]: df.head()

	lat	lng	desc	zip	title	timeStamp	twp	ad
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 C & DEAD EN
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 WHITEMARS L
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 AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTSGROVE	CHERRYWOOD CT & DEAD EN

Analysis and Visualisation - Applying Methods and Functions on Data and Plotting using appropriate graphs

In [7]: # top 5 zipcodes for calls
df['zip'].value_counts().head()

Out[7]: 19401.0 6979
19464.0 6643
19403.0 4854
19446.0 4748
19406.0 3174
Name: zip, dtype: int64

In [8]: # top 5 towns with maximum 911 calls
df['twp'].value_counts().head()

Out[8]: LOWER MERION 8443
ABINGTON 5977
NORRISTOWN 5890
UPPER MERION 5227
CHELTENHAM 4575
Name: twp, dtype: int64

In [9]: # total number of types of complaints
df['title'].nunique()

Out[9]: 110

```
In [10]: # creating a new column 'Reason' representing the code of the problems
df['Reason'] = df['title'].apply(lambda x:x.split(':')[0])
df.head()
```

Out[10]:

	lat	lng	desc	zip	title	timeStamp	twp	ad
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 C & DEAD EN
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 WHITEMARS L
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 AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTTSGROVE	CHERRYWO CT & DEA EN

◀ ▶

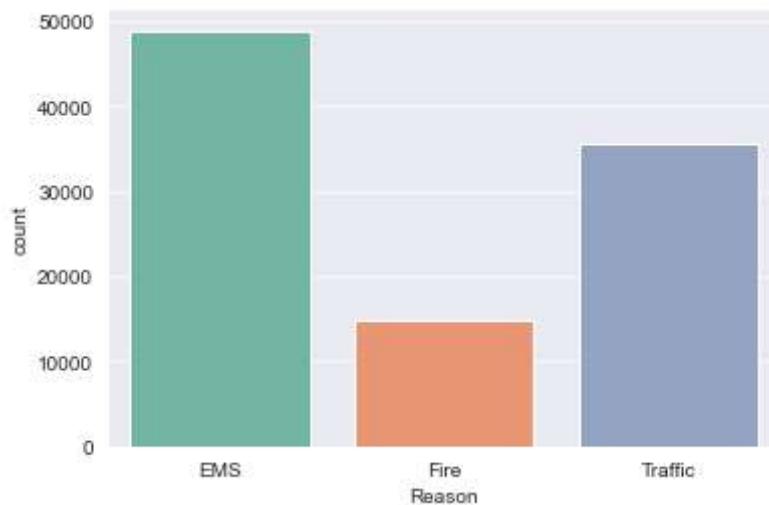
```
In [11]: # determining the reasons for 911 calls by codes
df['Reason'].value_counts()
```

Out[11]:

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

```
In [12]: # Reasons by count
sns.set_style('darkgrid')
sns.countplot(x=df['Reason'], data=df, palette='Set2')
```

Out[12]: <AxesSubplot: xlabel='Reason', ylabel='count'>



```
In [13]: # the type of of timeStamp col is string, thus converting it into date and time format
type(df['timeStamp'].iloc[0])
```

Out[13]: str

```
In [15]: df['timeStamp'] = pd.to_datetime(df['timeStamp'])
df.head()
```

	lat	lng	desc	zip	title	timeStamp	twp	addr
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 C & DEAD EN
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 WHITEMARS L
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 AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTSGROVE	CHERRYWO CT & DEA EN

```
In [16]: #retireving data from timeStamp
df['timeStamp'].iloc[0].hour
```

Out[16]: 17

In [17]:

```
#creating new columns by timeStamp
df['Hour'] = df['timeStamp'].apply(lambda x: x.hour)
df['Month'] = df['timeStamp'].apply(lambda x: x.month)
df['Day of Week'] = df['timeStamp'].apply(lambda x: x.dayofweek)
df.head()
```

Out[17]:

	lat	lng	desc	zip	title	timeStamp	twp	adc
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 C & DEAD EN
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 WHITEMARS L
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 AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTTSGROVE	CHERRYWOOD CT & DEAD EN

◀ ▶

```
# the day of the week is in numbered form, assigning string values of days
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}
df['Day'] = df['Day of Week'].map(dmap)
df.head()
```

Out[18]:

	lat	lng	desc	zip	title	timeStamp	twp	adc
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 C & DEAD EN
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 WHITEMARS L

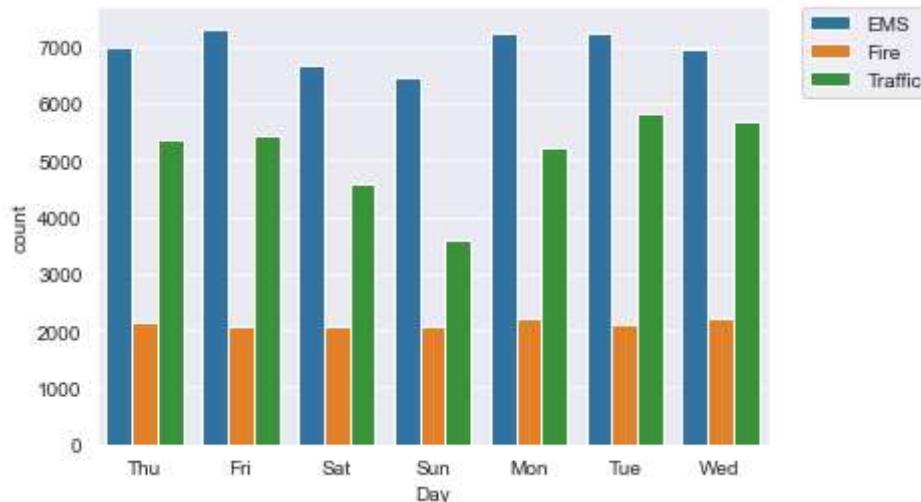
	lat	lng	desc	zip	title	timeStamp	twp	adr
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS- ODOR/LEAK	2015-12-10 17:40:00	NORRISTOWN	HAWS AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTSGROVE	CHERRYWO CT & DEA EN



In [21]: # Representing day-wise calls by reason

```
sns.countplot(data=df,x='Day',hue='Reason')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

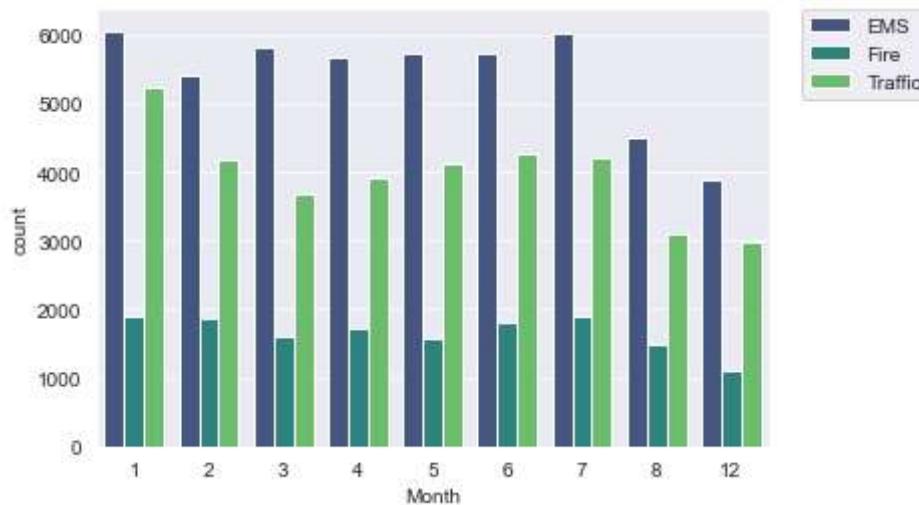
Out[21]: <matplotlib.legend.Legend at 0x18f132a9eb0>



In [22]: # month-wise calls

```
sns.countplot(x='Month',data=df,hue='Reason',palette='viridis')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

Out[22]: <matplotlib.legend.Legend at 0x18f1378b4f0>



```
In [25]: # Grouping all the values by month and plotting monthwise twp count
byMonth = df.groupby(by='Month').count()
byMonth.head()
```

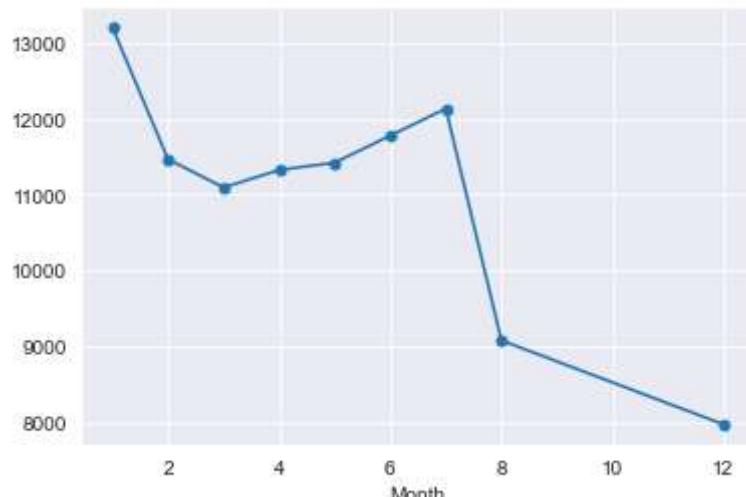
Out[25]:

	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
2	11467	11467	11467	9930	11467		11467	11465	11396	11467	11467	11467
3	11101	11101	11101	9755	11101		11101	11092	11059	11101	11101	11101
4	11326	11326	11326	9895	11326		11326	11323	11283	11326	11326	11326
5	11423	11423	11423	9946	11423		11423	11420	11378	11423	11423	11423



```
In [26]: byMonth['twp'].plot.line(marker='o', markersize=5)
```

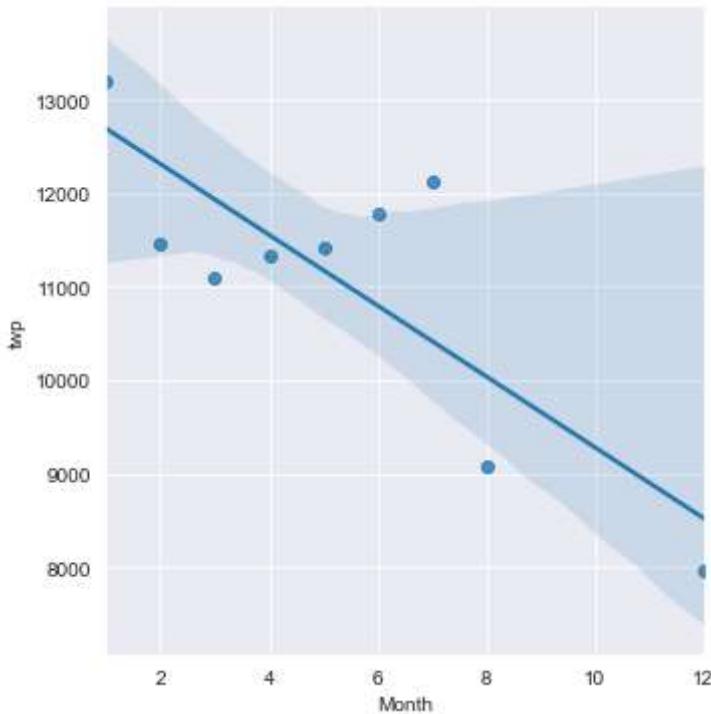
Out[26]: <AxesSubplot:xlabel='Month'>



```
In [27]: #regression plot for the same
```

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

Out[27]: <seaborn.axisgrid.FacetGrid at 0x18f139c9a90>



In [28]:

```
# creating a date column and grouping all the date values together by count
df['Date'] = df['timeStamp'].apply(lambda x: x.date())
df.groupby(by='Date').count()
df.head()
```

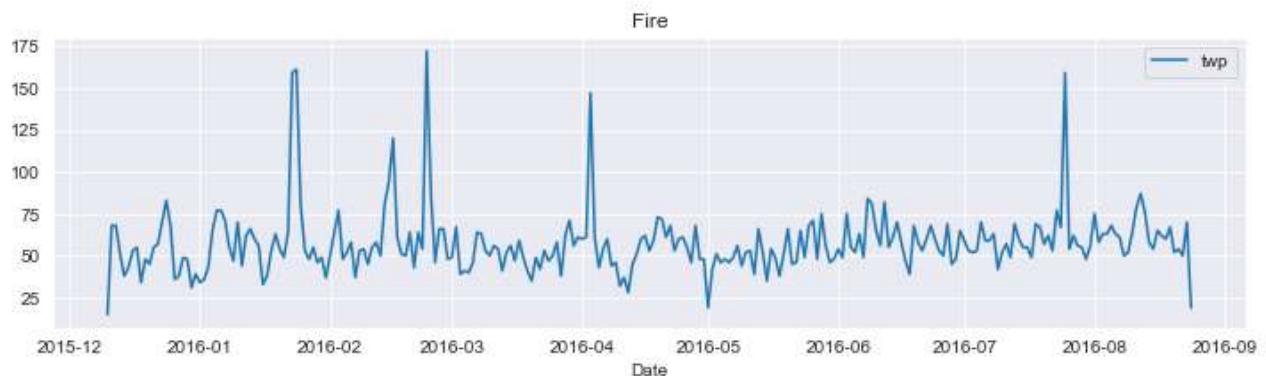
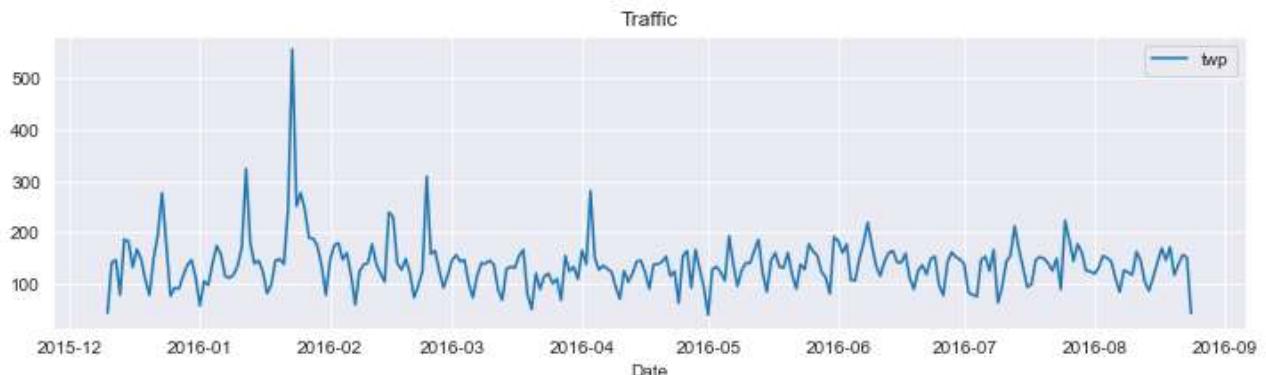
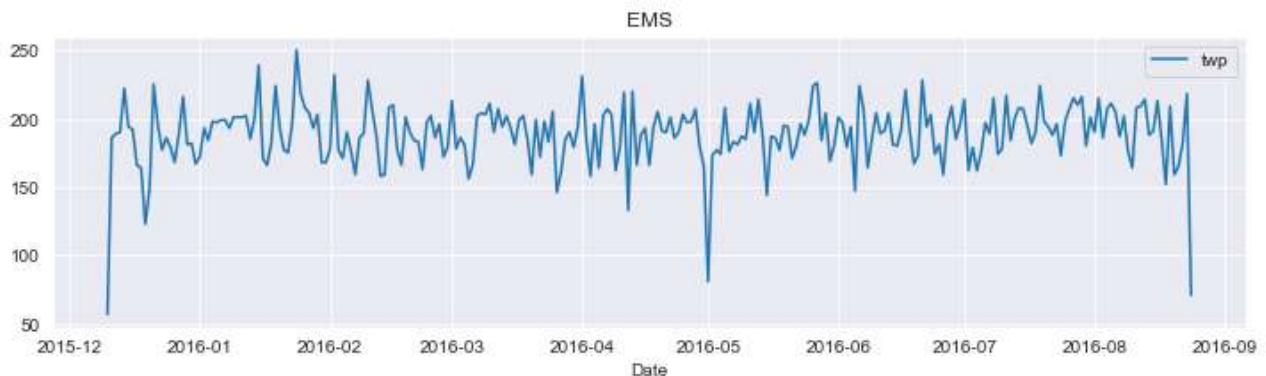
Out[28]:

	lat	lng	desc	zip	title	timeStamp	twp	ad
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 C & DEAD EN
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 WHITEMARS L
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 AV
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST SWEDE S

	lat	lng	desc	zip	title	timeStamp	twp	ad
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTSGROVE	CHERRYWO CT & DE EN

```
In [71]: # plotting calls made by date according to reason
plt.figure(dpi=100)
df[df['Reason']=='EMS'].groupby(by='Date').count().plot(y='twp', figsize=(12,3), title='E
df[df['Reason']=='Traffic'].groupby(by='Date').count().plot(y='twp', figsize=(12,3), titl
df[df['Reason']=='Fire'].groupby(by='Date').count().plot(y='twp', figsize=(12,3), title='
```

```
Out[71]: <AxesSubplot:title={'center':'Fire'}, xlabel='Date'
<Figure size 600x400 with 0 Axes>
```



```
In [43]: # Heatmaps: creating the data frame in matrix form with day as index and hour/month as
```

```
dayHour = df.groupby(by=['Day', 'Hour']).count()['Reason'].unstack()
dayMonth = df.groupby(by=['Day', 'Month']).count()['Reason'].unstack()
```

In [44]: `dayHour.head()`

Hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	...						
Day																									
Fri	275	235	191	175	201	194	372	598	742	752	...	932	980	1039	980	820	696	667	51	...	1	2	3	4	5
Mon	282	221	201	194	204	267	397	653	819	786	...	869	913	989	997	885	746	613	49	...	1	2	3	4	5
Sat	375	301	263	260	224	231	257	391	459	640	...	789	796	848	757	778	696	628	51	...	1	2	3	4	5
Sun	383	306	286	268	242	240	300	402	483	620	...	684	691	663	714	670	655	537	46	...	1	2	3	4	5
Thu	278	202	233	159	182	203	362	570	777	828	...	876	969	935	1013	810	698	617	51	...	1	2	3	4	5

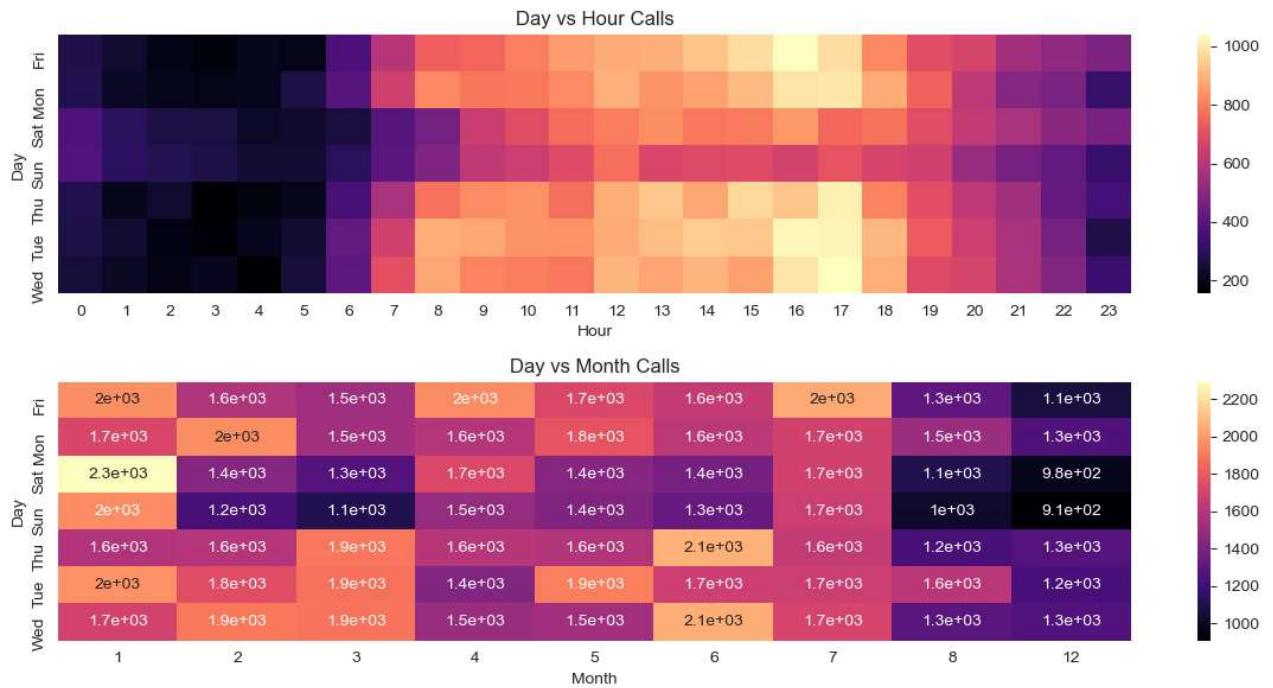
5 rows × 24 columns



In [45]: `dayMonth.head()`

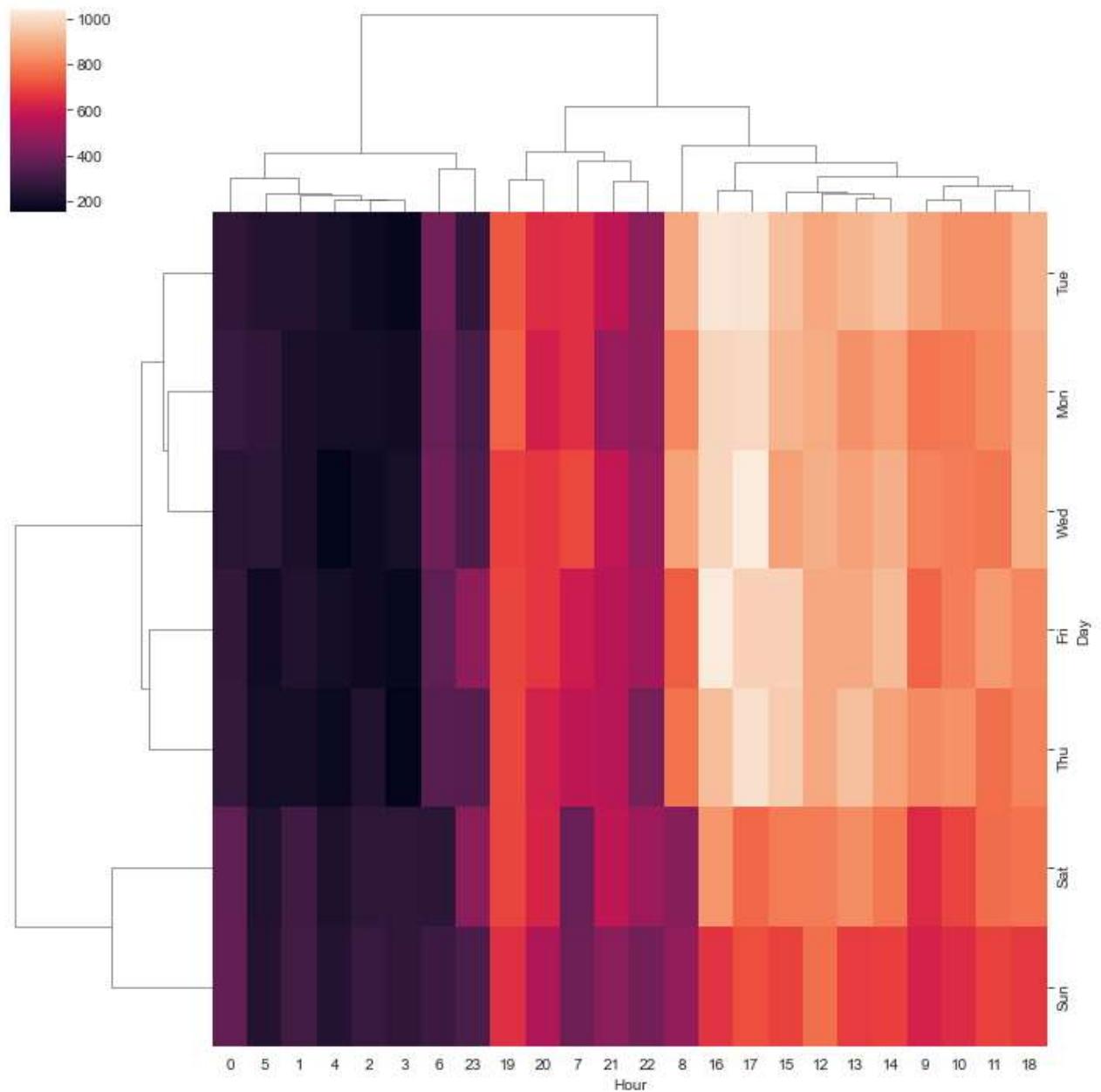
Month	1	2	3	4	5	6	7	8	12	...
Day										
Fri	1970	1581	1525	1958	1730	1649	2045	1310	1065	...
Mon	1727	1964	1535	1598	1779	1617	1692	1511	1257	...
Sat	2291	1441	1266	1734	1444	1388	1695	1099	978	...
Sun	1960	1229	1102	1488	1424	1333	1672	1021	907	...
Thu	1584	1596	1900	1601	1590	2065	1646	1230	1266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

```
# creating heatmaps - calls in Day vs Month/Hour
fig,axes = plt.subplots(2,1,figsize=(12,6),dpi=100)
sns.heatmap(dayHour,ax=axes[0],annot=False,cmap='magma')
axes[0].set_title('Day vs Hour Calls')
sns.heatmap(dayMonth,ax=axes[1],annot=True,cmap='magma')
axes[1].set_title('Day vs Month Calls')
plt.tight_layout()
```



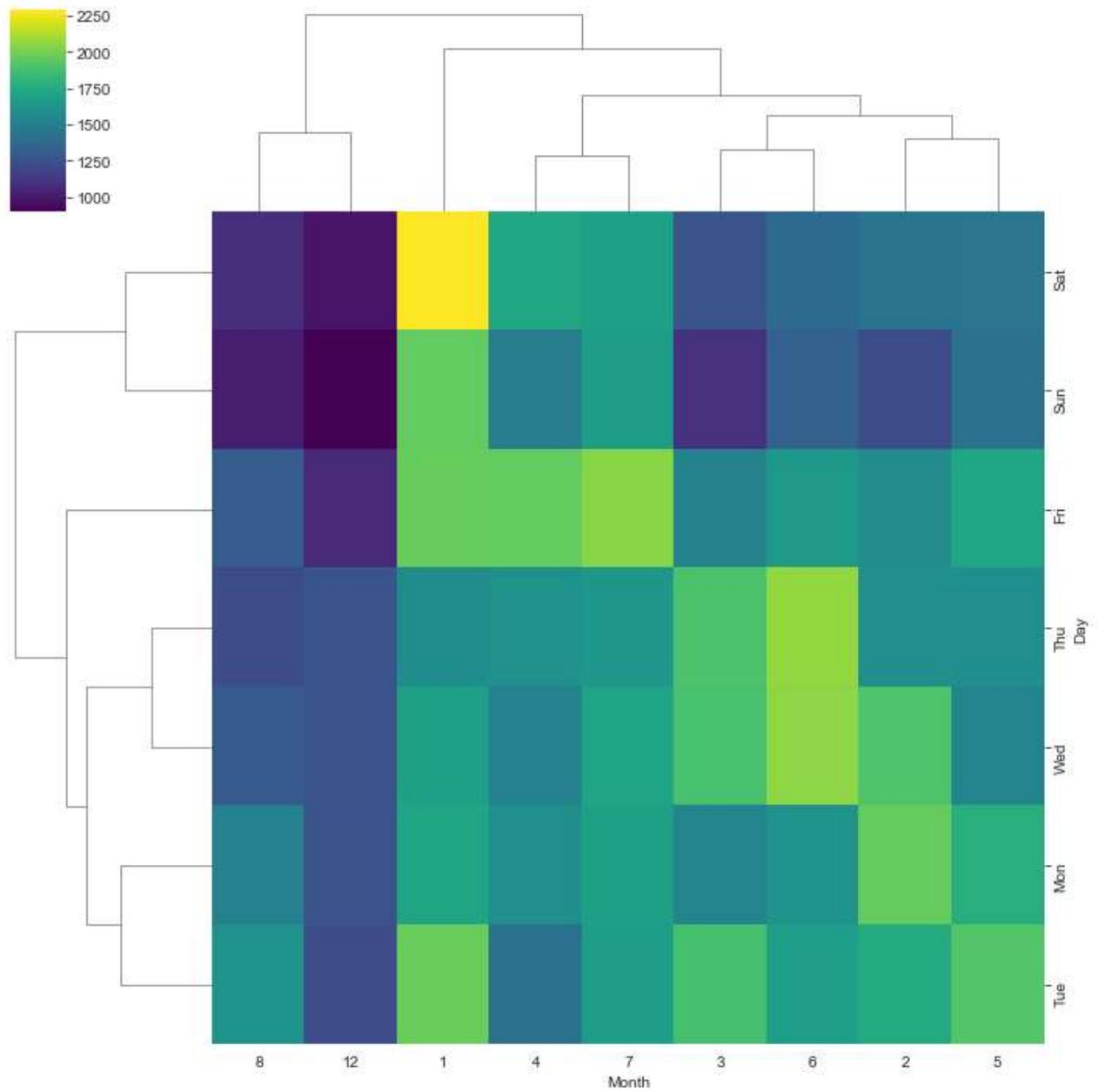
```
In [73]: #clustermaps
sns.clustermap(dayHour)
```

```
Out[73]: <seaborn.matrix.ClusterGrid at 0x18f1eeecd430>
```



```
In [74]: sns.clustermap(dayMonth,cmap='viridis')
```

```
Out[74]: <seaborn.matrix.ClusterGrid at 0x18f1edb5b20>
```



END