Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
from plotly import __version__
```

In [3]:
```
print(__version__)
```

```
5.16.1
```

In [4]:
```
import cufflinks as cf
```

```
C:\Users\ravix\anaconda3\Lib\site-packages\paramiko\transport.py:219: CryptographyDeprecationWarning:

Blowfish has been deprecated
```

In [5]:
```
#from plotly.offline import download.plotlyjs, init_notebook_mode,plot,iplot

from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

In [6]:
```
init_notebook_mode(connected=True)
```

In [7]:
```
cf.go_offline()
```

Reading Dataframe

In [8]:
```
df = pd.read_csv('911.csv')
```

In [9]:
```
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 [10]:
```
df.head()
```

Out[10]:

	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 LN	1
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
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 ST	1
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 END	1

Finding top 5 zipcodes for 911 calls?

In [11]:
```
df['zip'].value_counts().head(5)
```

Out[11]:
```
19401.0    6979
19464.0    6643
19403.0    4854
19446.0    4748
19406.0    3174
Name: zip, dtype: int64
```

Finding top 5 townships for 911 calls?

In [12]:
```
df['twp'].value_counts().head(5)
```

Out[12]:
```
LOWER MERION    8443
ABINGTON        5977
NORRISTOWN      5890
UPPER MERION    5227
CHELTENHAM      4575
Name: twp, dtype: int64
```

Finding number of unique titles

In [13]:
```
df['title'].nunique()
```

Out[13]:
```
110
```

Creating a new feature by splitting the Title column and grabbing Reason for 911 calls in Reason Column

In [14]:
```
df['Reason']= df['title'].apply(lambda x:x.split(':')[0])
```

In [15]:
```
df['Reason'].head()
```

Out[15]:
```
0     EMS
1     EMS
2    Fire
3     EMS
4     EMS
Name: Reason, dtype: object
```

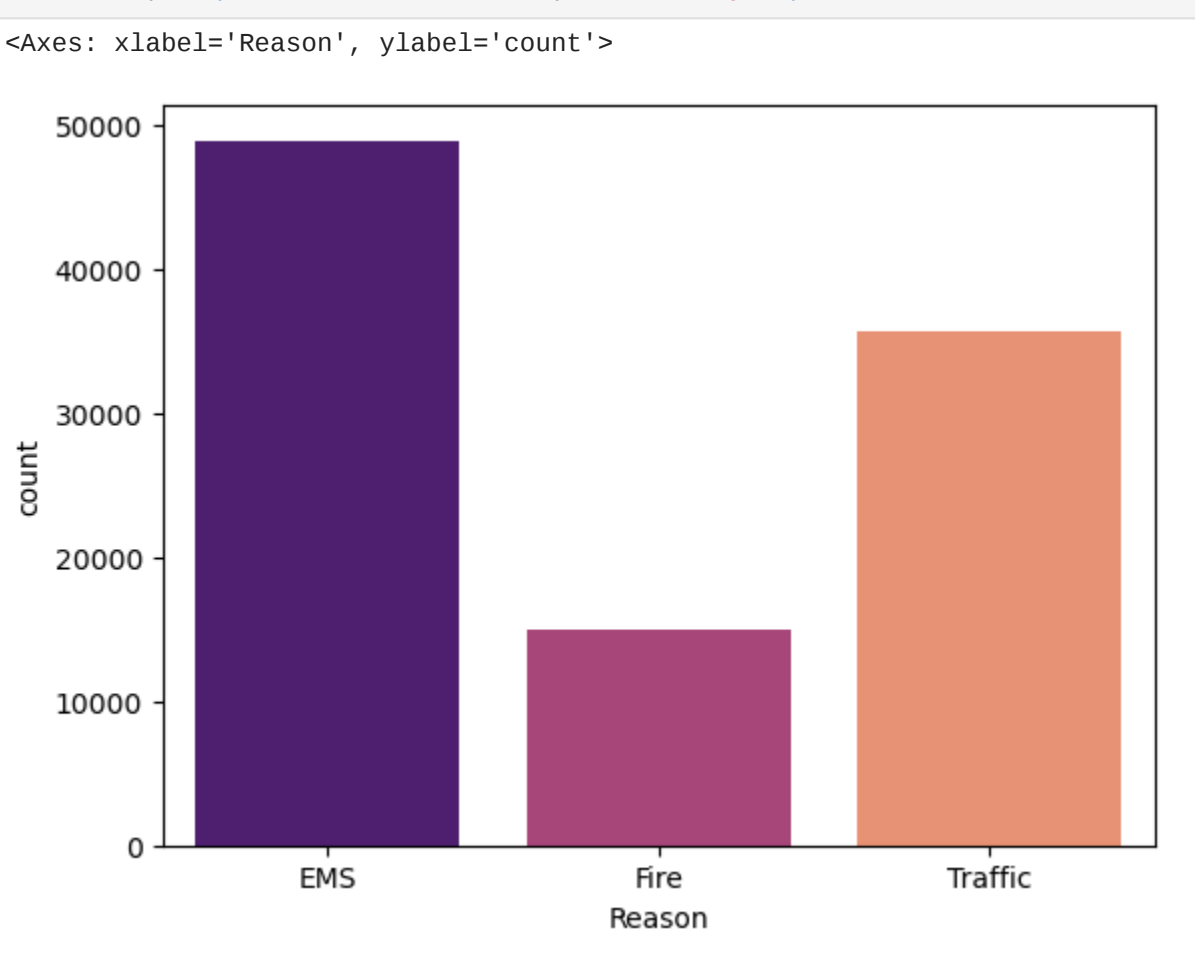
In [16]:
```
df['Reason'].value_counts()
```

Out[16]:
```
EMS        48877
Traffic    35695
Fire       14920
Name: Reason, dtype: int64
```

Plotting counts of different reasons of 911 calls

In [17]:
```
sns.countplot(x='Reason', data=df, palette='magma')
```

Out[17]:
```
<Axes: xlabel='Reason', ylabel='count'>
```



In [18]:
```
type('timeStamp')
```

Out[18]:
```
str
```

Changing timeStamp column from string to DateTime object

so that we can grab each element separately(hour,days,months)

In [19]:
```
df['timeStamp']=pd.to_datetime(df['timeStamp'])
```

In [20]:
```
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 [21]:
```
df.head()
```

Out[21]:

	lat	lng	desc	zip	title	timeStamp	twp	addr	e	Reason	Hour	Month	Day of Week
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	EMS	17	12	3
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 LN	1	EMS	17	12	3
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	Fire	17	12	3
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 ST	1	EMS	17	12	3
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 END	1	EMS	17	12	3

Mapping names of days to Day of Week column

In [22]:
```
days={0:'Mon',1:'Tue',2:'Wed',3:'Thu',4:'Fri',5:'Sat',6:'Sun'}
```

In [23]:
```
df['Day of Week']= df['Day of Week'].map(days)
```

In [24]:
```
df.head()
```

Out[24]:

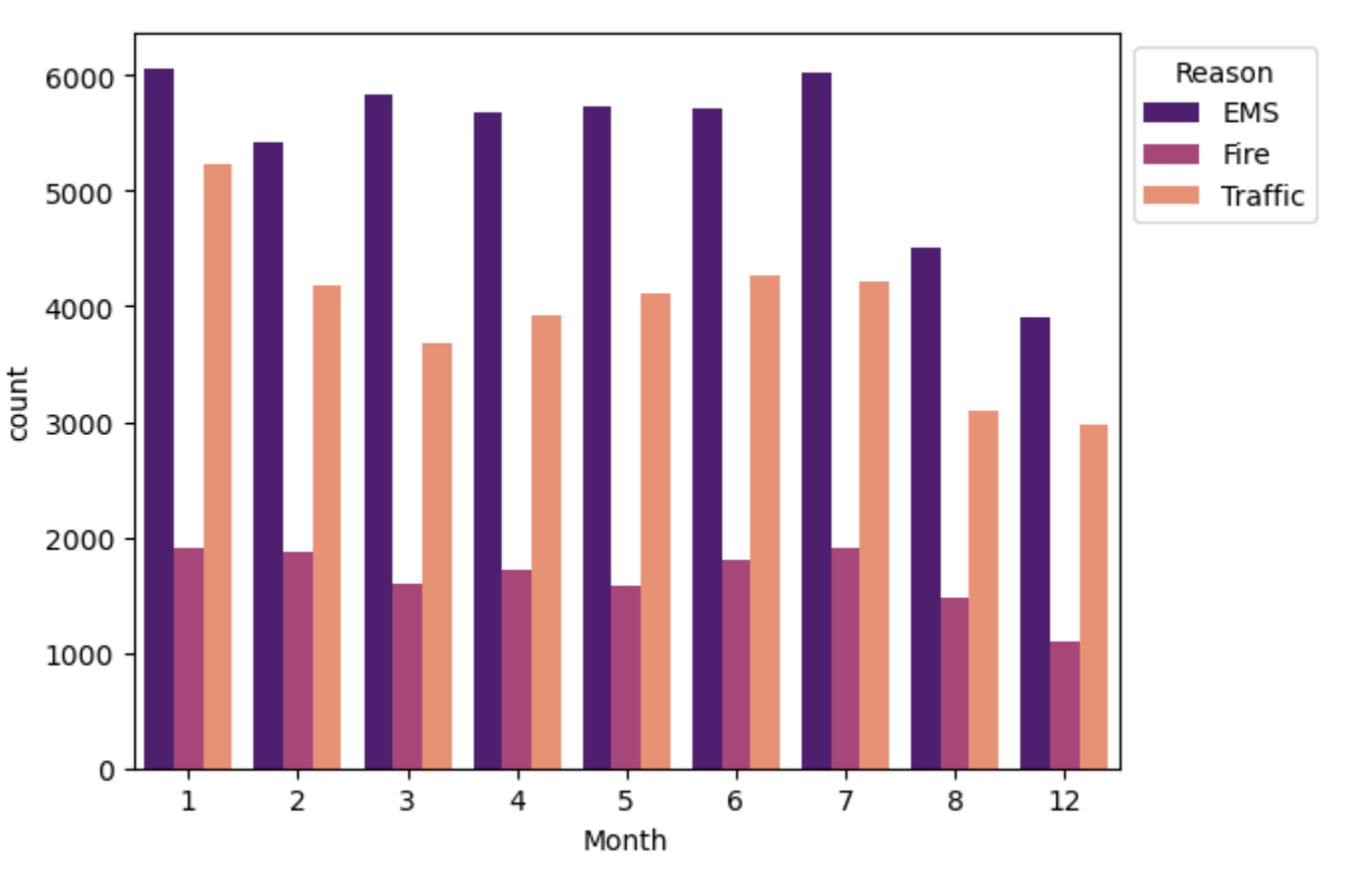
	lat	lng	desc	zip	title	timeStamp	twp	addr	e	Reason	Hour	Month	Day of Week
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	2015-12-10 17:40:00	NEW HANOVER	REINDEER CT & DEAD END	1	EMS	17	12	Thu
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	2015-12-10 17:40:00	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH LN	1	EMS	17	12	Thu
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	Fire	17	12	Thu
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 ST	1	EMS	17	12	Thu
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 END	1	EMS	17	12	Thu

Finding which reason is quite frequent for 911 calls

In [25]:
```
sns.countplot(x='Month', data=df, hue='Reason', palette='magma')

plt.legend(title='Reason', loc='upper left', bbox_to_anchor=(1, 1))
```

Out[25]:
```
<matplotlib.legend.Legend at 0x1d055bc2e10>
```



Grouping the dataframe based on month and using count method fror aggregation

In [26]:
```
df1= df.groupby('Month').count()
df1
```

Out[26]:

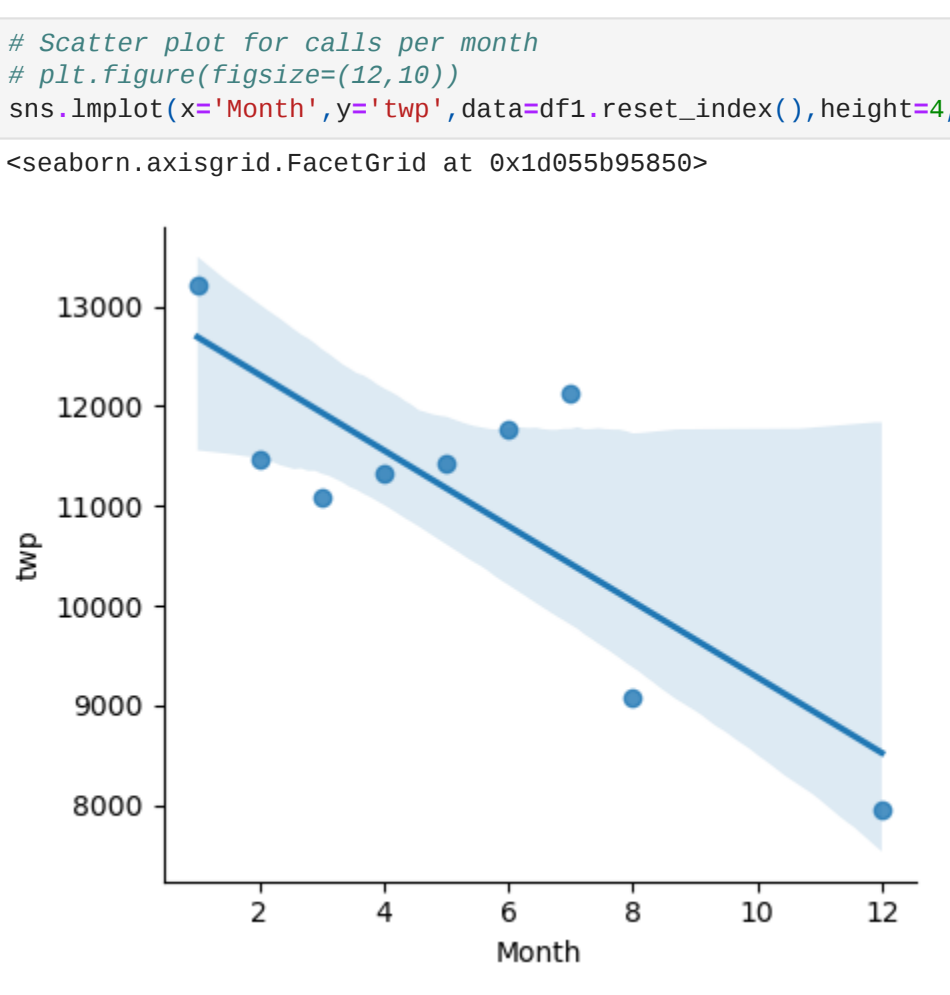
Month	lat	lng	desc	zip	title	timeStamp	twp	addr	e	Reason	Hour	Day of Week
1	13205	13205	13205	11527	13205	13205	13203	13096	13205	13205	13205	13205
2	11467	11467	11467	9930	11467	11467	11465	11396	11467	11467	11467	11467
3	11101	11101	11101	9755	11101	11101	11092	11059	11101	11101	11101	11101
4	11326	11326	11326	9895	11326	11326	11323	11283	11326	11326	11326	11326
5	11423	11423	11423	9946	11423	11423	11420	11378	11423	11423	11423	11423
6	11786	11786	11786	10212	11786	11786	11777	11732	11786	11786	11786	11786
7	12137	12137	12137	10633	12137	12137	12133	12088	12137	12137	12137	12137
8	9078	9078	9078	7832	9078	9078	9073	9025	9078	9078	9078	9078
12	7969	7969	7969	6907	7969	7969	7963	7916	7969	7969	7969	7969

In [27]:
```
df1['twp'].iplot(mode='lines',dimensions=(600, 500))
```

In [ ]:

In [28]:
```
# Scatter plot for calls per month
# plt.figure(figsize=(12,10))
sns.lmplot(x='Month',y='twp',data=df1.reset_index(),height=4,aspect=1.2)
```

Out[28]:
```
<seaborn.axisgrid.FacetGrid at 0x1d055b95850>
```



From both the line chart and scatter plot it seems the 911 calls dropped from month of JULY

In [ ]:

In [29]:
```
dayHour = df.groupby(by=['Day of Week','Hour']).count()['Reason'].unstack()
dayHour.head()
```

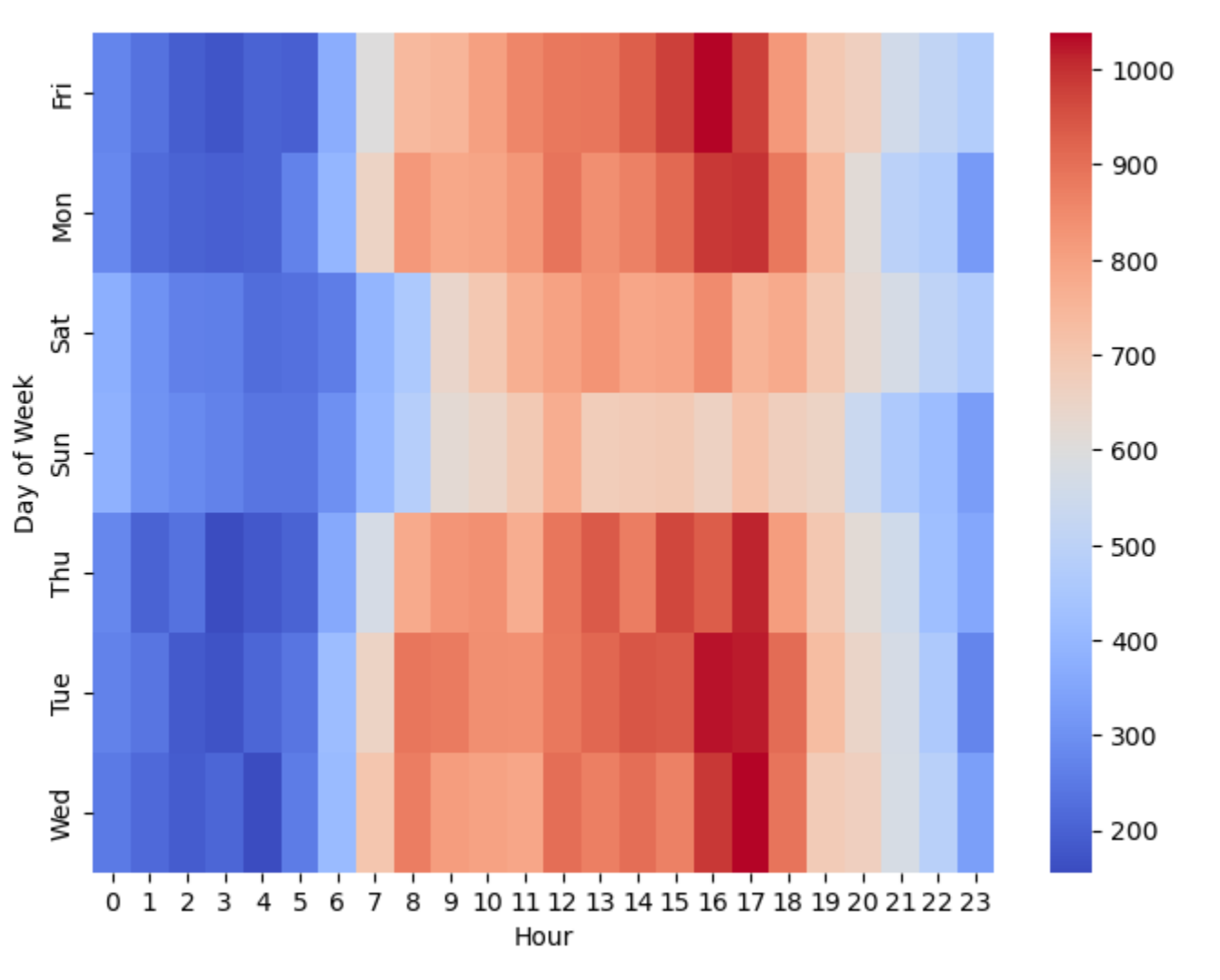
Out[29]:

Hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	22	23
Day of Week																					
Fri	275	235	191	175	201	194	372	598	742	752	...	932	980	1039	980	820	696	667	559	514	474
Mon	282	221	201	194	204	267	397	653	819	786	...	869	913	989	997	885	746	613	497	472	325
Sat	375	301	263	260	224	231	257	391	459	640	...	789	796	848	757	778	696	628	572	506	467
Sun	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

5 rows × 24 columns

In [30]:
```
plt.figure(figsize=(8,6))
sns.heatmap(dayHour,cmap='coolwarm')
```

Out[30]:
```
<Axes: xlabel='Hour', ylabel='Day of Week'>
```



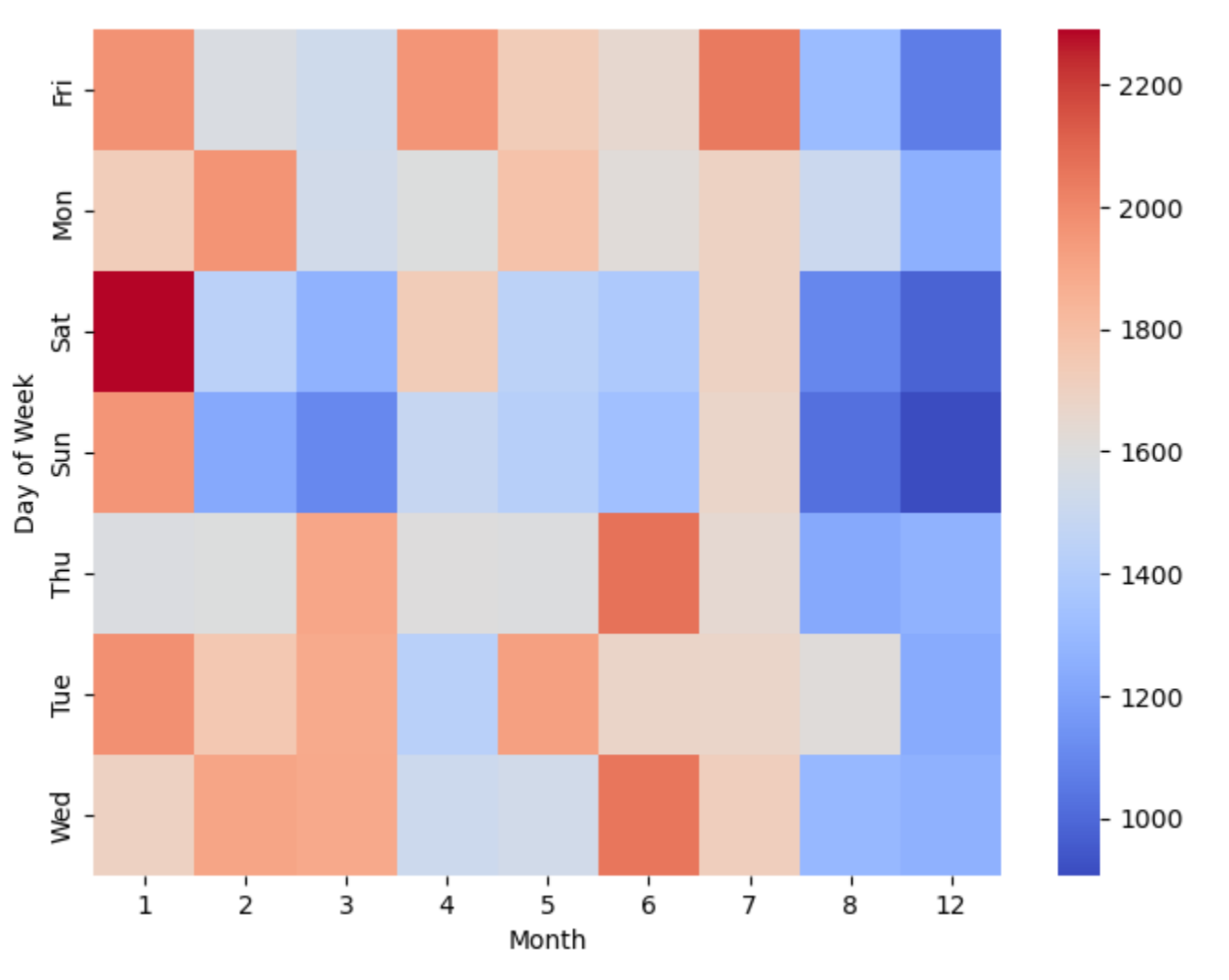
In [31]:
```
dayMonth = df.groupby(by=['Day of Week','Month']).count()['Reason'].unstack()
dayMonth.head()
```

Out[31]:

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 [32]:
```
plt.figure(figsize=(8,6))
sns.heatmap(dayMonth,cmap='coolwarm')
```

Out[32]:
```
<Axes: xlabel='Month', ylabel='Day of Week'>
```



Key Insights

1. The most frequent zip code and townships are 6979 and LOWER MERION resp. for 911 calls.

2. The most common reason for 911 call is EMS(Emergency Medical Services)

3. The frequency of 911 calls was at peak in january and was changing overtime and after July it fell suddenly.

4. The most 911 calls were made during 5PM on Tues,Wed,Thurs.

5 If we go by bigger picture, the most 911 calls were made on Saturday in the month of May.