In [1]:

```
import pandas as pd
```

In [3]:

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

In [4]:

```
df.head()
```

Out[4]:

	lat	lng	desc	zip	title	timeStamp	twp	
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	10-12-2015 17:10	NEW HANOVER	RE &
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	10-12-2015 17:29	HATFIELD TOWNSHIP	BF WI
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	10-12-2015 14:39	NORRISTOWN	
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	10-12-2015 16:47	NORRISTOWN	
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	10-12-2015 16:56	LOWER POTTSGROVE	CHE

In [5]:

```
df.tail()
```

Out[5]:

	lat	lng	desc	zip	title	timeStamp	twp
663517	40.157956	-75.348060	SUNSET AVE & WOODLAND AVE; EAST NORRITON; 2020...	19403.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:46	EAST NORRITON
663518	40.136306	-75.428697	EAGLEVILLE RD & BUNTING CIR; LOWER PROVIDENCE...	19403.0	EMS: GENERAL WEAKNESS	29-07-2020 15:52	LOWER PROVIDENCE
663519	40.013779	-75.300835	HAVERFORD STATION RD; LOWER MERION; Station 3...	19041.0	EMS: VEHICLE ACCIDENT	29-07-2020 15:52	LOWER MERION
663520	40.121603	-75.351437	MARSHALL ST & HAWS AVE; NORRISTOWN; 2020-07-29...	19401.0	Fire: BUILDING FIRE	29-07-2020 15:54	NORRISTOWN
663521	40.015046	-75.299674	HAVERFORD STATION RD & W MONTGOMERY AVE; LOWER...	19041.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:52	LOWER MERION

In [6]:

```
df.shape
```

Out[6]:

```
(663522, 9)
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['lat', 'lng', 'desc', 'zip', 'title', 'timeStamp', 'twp', 'addr', 'e'],
dtype='object')
```

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
240
```

In [9]:

```
df = df.drop_duplicates()
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
lat              0
lng              0
desc             0
zip          80165
title            0
timeStamp        0
twp            293
addr             0
e                0
dtype: int64
```

In [12]:

```
import numpy as np
```

In [13]:

```
df = df.replace(np.nan, 'Not Available')
```

In [14]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 663282 entries, 0 to 663521
Data columns (total 9 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   lat        663282 non-null  float64
 1   lng        663282 non-null  float64
 2   desc       663282 non-null  object 
 3   zip        663282 non-null  object 
 4   title      663282 non-null  object 
 5   timeStamp  663282 non-null  object 
 6   twp        663282 non-null  object 
 7   addr       663282 non-null  object 
 8   e          663282 non-null  int64  
dtypes: float64(2), int64(1), object(6)
memory usage: 50.6+ MB
```

In [15]:

```
df.describe()
```

Out[15]:

	lat	lng	e
count	663282.000000	663282.000000	663282.0
mean	40.158166	-75.300103	1.0
std	0.220676	1.673185	0.0
min	0.000000	-119.698206	1.0
25%	40.100344	-75.392735	1.0
50%	40.143927	-75.305143	1.0
75%	40.229008	-75.211865	1.0
max	51.335390	87.854975	1.0

In [16]:

```
df.nunique()
```

Out[16]:

```
lat           25949
lng           25980
desc         663282
zip             205
title           148
timeStamp    543989
twp              69
addr          41292
e                 1
dtype: int64
```

In [18]:

```
df['station_num'] = df.desc.str.split('Station', expand=True)[1].str.split(';', expand=True)
```

In [19]:

```
df
```

Out[19]:

	lat	lng	desc	zip	title	timeStamp	tw
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	10-12-2015 17:10	NEV HANOVEF
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	10-12-2015 17:29	HATFIELI TOWNSHII
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	10-12-2015 14:39	NORRISTOWI
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	10-12-2015 16:47	NORRISTOWI
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	Not Available	EMS: DIZZINESS	10-12-2015 16:56	LOWEF POTTSGROVI
...	...	...	...	...	...	...	..
663517	40.157956	-75.348060	SUNSET AVE & WOODLAND AVE; EAST NORRITON; 2020...	19403.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:46	EAS NORRITOI
663518	40.136306	-75.428697	EAGLEVILLE RD & BUNTING CIR; LOWER PROVIDENCE...	19403.0	EMS: GENERAL WEAKNESS	29-07-2020 15:52	LOWEF PROVIDENCI
663519	40.013779	-75.300835	HAVERFORD STATION RD; LOWER MERION; Station 3...	19041.0	EMS: VEHICLE ACCIDENT	29-07-2020 15:52	LOWEF MERIOI
663520	40.121603	-75.351437	MARSHALL ST & HAWS AVE; NORRISTOWN; 2020-07-29...	19401.0	Fire: BUILDING FIRE	29-07-2020 15:54	NORRISTOWI
663521	40.015046	-75.299674	HAVERFORD STATION RD & W MONTGOMERY AVE; LOWER...	19041.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:52	LOWEF MERIOI

663282 rows × 10 columns

In [20]:

```
df.station_num.str.replace(':', '').value_counts()[:10]
```

Out[20]:

```
 308A    25332
 329     21893
 313     20318
 381     15940
 345     14500
 308     13958
 351     12701
 345B    12160
 317     11800
 382     11163
Name: station_num, dtype: int64
```

In [21]:

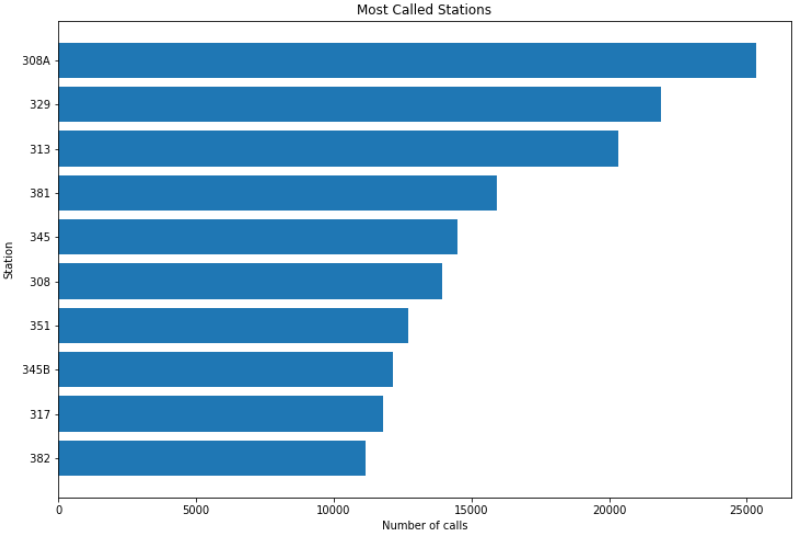
```
df1 = df.station_num.str.replace(':', '').value_counts()[:10]
```

In [23]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [25]:

```
plt.figure(figsize=(12, 8))
x = list(df1.index)
y = list(df1.values)
x.reverse()
y.reverse()

plt.title("Most Called Stations")
plt.ylabel("Station")
plt.xlabel("Number of calls")

plt.barh(x, y)
plt.show()
```



In [27]:

```
df['zip'].unique()
```

Out[27]:

```
array([19525.0, 19446.0, 19401.0, 'Not Available', 19044.0, 19426.0,
       19438.0, 19462.0, 19428.0, 19040.0, 19027.0, 18936.0, 18974.0,
       19031.0, 19403.0, 19422.0, 19085.0, 18964.0, 19038.0, 19406.0,
       19468.0, 19010.0, 19095.0, 19464.0, 19444.0, 19041.0, 19440.0,
       19405.0, 19002.0, 19096.0, 19454.0, 19465.0, 19004.0, 19066.0,
       19072.0, 18041.0, 19046.0, 19090.0, 19012.0, 19025.0, 19473.0,
       18073.0, 18969.0, 18074.0, 19460.0, 19001.0, 18054.0, 19009.0,
       19006.0, 19035.0, 19150.0, 19075.0, 19034.0, 19151.0, 19453.0,
       19003.0, 18914.0, 19512.0, 18976.0, 19120.0, 18915.0, 18076.0,
       19477.0, 19087.0, 18966.0, 19131.0, 19128.0, 19083.0, 19053.0,
       19475.0, 18960.0, 19504.0, 18070.0, 19492.0, 18932.0, 19118.0,
       18092.0, 19490.0, 19518.0, 18056.0, 19119.0, 19107.0, 17752.0,
       19111.0, 18927.0, 19435.0, 18951.0, 19472.0, 19503.0, 19126.0,
       19505.0, 19423.0, 19138.0, 36107.0, 18036.0, 19116.0, 19139.0,
       19129.0, 19115.0, 19355.0, 77316.0, 19457.0, 19082.0, 19127.0,
       19443.0, 17555.0, 19520.0, 19063.0, 19020.0, 19404.0, 19382.0,
       19474.0, 19057.0, 19073.0, 19121.0, 18958.0, 19026.0, 19018.0,
       19047.0, 19064.0, 19602.0, 19486.0, 19348.0, 18051.0, 18049.0,
       19333.0, 19144.0, 18101.0, 19607.0, 19450.0, 19380.0, 17506.0,
       8361.0, 18940.0, 18104.0, 7203.0, 19030.0, 8033.0, 19104.0,
       17545.0, 8832.0, 19021.0, 19106.0, 8065.0, 15301.0, 18911.0,
       18902.0, 18944.0, 3366.0, 19545.0, 19390.0, 19140.0, 18901.0,
       19601.0, 19341.0, 19301.0, 19425.0, 23005.0, 19054.0, 18040.0,
       18102.0, 17603.0, 18080.0, 17901.0, 19153.0, 21701.0, 18103.0,
       19134.0, 19135.0, 8502.0, 19122.0, 19320.0, 3103.0, 19610.0,
       19102.0, 17331.0, 19050.0, 19023.0, 17810.0, 8077.0, 8628.0,
       19605.0, 19437.0, 19312.0, 19147.0, 19456.0, 19604.0, 17507.0,
       1104.0, 18042.0, 18011.0, 15090.0, 19543.0, 19124.0, 19609.0,
       19445.0, 19310.0, 19070.0, 7081.0, 7726.0, 17566.0, 19008.0,
       19365.0, 19103.0, 18938.0], dtype=object)
```

In [28]:

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

Out[28]:

```
Not Available    80165
19401.0          45583
19464.0          43900
19403.0          34874
19446.0          32257
                 ...
17901.0              1
19134.0              1
19135.0              1
8502.0               1
18938.0              1
Name: zip, Length: 205, dtype: int64
```

In [29]:

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

```
Out[29]:
```

```
array(['EMS: BACK PAINS/INJURY', 'EMS: DIABETIC EMERGENCY',
       'Fire: GAS-ODOR/LEAK', 'EMS: CARDIAC EMERGENCY', 'EMS: DIZZINESS',
       'EMS: HEAD INJURY', 'EMS: NAUSEA/VOMITING',
       'EMS: RESPIRATORY EMERGENCY', 'EMS: SYNCOPAL EPISODE',
       'Traffic: VEHICLE ACCIDENT -', 'EMS: VEHICLE ACCIDENT',
       'Traffic: DISABLED VEHICLE -', 'Fire: APPLIANCE FIRE',
       'EMS: GENERAL WEAKNESS', 'Fire: CARBON MONOXIDE DETECTOR',
       'EMS: UNKNOWN MEDICAL EMERGENCY', 'EMS: UNRESPONSIVE SUBJECT',
       'Fire: VEHICLE ACCIDENT', 'EMS: ALTERED MENTAL STATUS',
       'Fire: FIRE ALARM', 'EMS: CVA/STROKE',
       'Traffic: ROAD OBSTRUCTION -', 'EMS: SUBJECT IN PAIN',
       'EMS: HEMORRHAGING', 'EMS: FALL VICTIM', 'EMS: ASSAULT VICTIM',
       'EMS: SEIZURES', 'EMS: MEDICAL ALERT ALARM',
       'EMS: ABDOMINAL PAINS', 'Fire: PUMP DETAIL',
       'Fire: FIRE INVESTIGATION', 'EMS: OVERDOSE', 'EMS: MATERNITY',
       'EMS: UNCONSCIOUS SUBJECT', 'EMS: CHOKING', 'EMS: LACERATIONS',
       'Fire: TRASH/DUMPSTER FIRE', 'Fire: UNKNOWN TYPE FIRE',
       'Fire: BUILDING FIRE', 'Fire: ELECTRICAL FIRE OUTSIDE',
       'Fire: DEBRIS/FLUIDS ON HIGHWAY',
       'Traffic: DEBRIS/FLUIDS ON HIGHWAY -', 'EMS: FEVER',
       'EMS: ALLERGIC REACTION', 'Traffic: VEHICLE LEAKING FUEL -',
       'EMS: FRACTURE', 'Fire: BURN VICTIM', 'EMS: BURN VICTIM',
       'Fire: RESCUE - GENERAL', 'Fire: WOODS/FIELD FIRE',
       'EMS: RESCUE - GENERAL', 'Fire: FIRE SPECIAL SERVICE',
       'Fire: VEHICLE FIRE', 'Traffic: VEHICLE FIRE -',
       'EMS: WARRANT SERVICE', 'Fire: S/B AT HELICOPTER LANDING',
       'EMS: EMS SPECIAL SERVICE', 'Traffic: HAZARDOUS ROAD CONDITIONS -',
       'Fire: RESCUE - ELEVATOR', 'EMS: FIRE SPECIAL SERVICE',
       'EMS: DEHYDRATION', 'EMS: CARBON MONOXIDE DETECTOR',
       'EMS: BUILDING FIRE', 'EMS: APPLIANCE FIRE', 'EMS: SHOOTING',
       'EMS: POISONING', 'Fire: TRANSFERRED CALL',
       'Fire: RESCUE - TECHNICAL', 'EMS: RESCUE - TECHNICAL',
       'Fire: VEHICLE LEAKING FUEL', 'EMS: EYE INJURY',
       'EMS: ELECTROCUTION', 'EMS: STABBING', 'Fire: FIRE POLICE NEEDED',
       'EMS: AMPUTATION', 'EMS: ANIMAL BITE', 'EMS: FIRE ALARM',
       'EMS: VEHICLE FIRE', 'EMS: HAZARDOUS MATERIALS INCIDENT',
       'EMS: RESCUE - ELEVATOR', 'EMS: FIRE INVESTIGATION',
       'Fire: MEDICAL ALERT ALARM', 'EMS: UNKNOWN TYPE FIRE',
       'EMS: GAS-ODOR/LEAK', 'Fire: TRAIN CRASH',
       'Fire: HAZARDOUS MATERIALS INCIDENT', 'EMS: TRANSFERRED CALL',
       'EMS: TRAIN CRASH', 'EMS: RESCUE - WATER',
       'EMS: S/B AT HELICOPTER LANDING',
       'Fire: UNKNOWN MEDICAL EMERGENCY', 'Fire: RESCUE - WATER',
       'EMS: CARDIAC ARREST', 'EMS: PLANE CRASH', 'Fire: PLANE CRASH',
       'EMS: WOODS/FIELD FIRE', 'Fire: CARDIAC ARREST',
       'Fire: EMS SPECIAL SERVICE', 'Fire: UNCONSCIOUS SUBJECT',
       'EMS: HEAT EXHAUSTION', 'EMS: DEBRIS/FLUIDS ON HIGHWAY',
       'EMS: ACTIVE SHOOTER', 'EMS: DISABLED VEHICLE',
       'Fire: POLICE INFORMATION', 'Fire: DIABETIC EMERGENCY',
       'EMS: BOMB DEVICE FOUND', 'Fire: SYNCOPAL EPISODE',
       'EMS: INDUSTRIAL ACCIDENT', 'EMS: DROWNING', 'EMS: SUSPICIOUS',
       'Fire: COLD VICTIM', 'EMS: ARMED SUBJECT',
       'Fire: CARDIAC EMERGENCY', 'EMS: STANDBY FOR ANOTHER CO',
       'EMS: ELECTRICAL FIRE OUTSIDE', 'Fire: UNRESPONSIVE SUBJECT',
       'EMS: HIT + RUN', 'EMS: SUICIDE THREAT', 'Fire: DISABLED VEHICLE',
       'Fire: ANIMAL COMPLAINT', 'Fire: STANDBY FOR ANOTHER CO',
       'EMS: FIRE POLICE NEEDED', 'Fire: PRISONER IN CUSTODY',
       'EMS: TRASH/DUMPSTER FIRE', 'Fire: SUSPICIOUS',
       'Fire: SUBJECT IN PAIN', 'Fire: RESPIRATORY EMERGENCY',
       'Fire: SUICIDE ATTEMPT', 'Fire: NAUSEA/VOMITING',
       'Fire: PUBLIC SERVICE', 'Fire: OVERDOSE', 'Fire: HEAD INJURY',
       'EMS: BOMB THREAT', 'Fire: GENERAL WEAKNESS',
```

In [30]:

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

Out[30]:

```
Traffic: VEHICLE ACCIDENT -       148315
Traffic: DISABLED VEHICLE -        47889
Fire: FIRE ALARM                   38320
EMS: FALL VICTIM                   34663
EMS: RESPIRATORY EMERGENCY         34235
                                   ...  
EMS: DISABLED VEHICLE                  1
Fire: PRISONER IN CUSTODY              1
Fire: GENERAL WEAKNESS                 1
Fire: SUSPICIOUS                       1
Fire: BARRICADED SUBJECT               1
Name: title, Length: 148, dtype: int64
```

In [32]:

```
df['twp'].unique()
```

Out[32]:

```
array(['NEW HANOVER', 'HATFIELD TOWNSHIP', 'NORRISTOWN',
       'LOWER POTTSGROVE', 'LANSDALE', 'HORSHAM', 'SKIPPACK',
       'LOWER SALFORD', 'PLYMOUTH', 'MONTGOMERY', 'UPPER MORELAND',
       'CHELTENHAM', 'UPPER MERION', 'WHITEMARSH', 'UPPER GWYNEDD',
       'LOWER PROVIDENCE', 'UPPER DUBLIN', 'WHITPAIN', 'DELAWARE COUNTY',
       'FRANCONIA', 'WEST CONSHOHOCKEN', 'LOWER MERION', 'LIMERICK',
       'TOWAMENCIN', 'DOUGLASS', 'POTTSTOWN', 'BRIDGEPORT', 'AMBLER',
       'CHESTER COUNTY', 'UPPER HANOVER', 'SPRINGFIELD', 'ROCKLEDGE',
       'ABINGTON', 'WEST NORRITON', 'ROYERSFORD', 'UPPER SALFORD',
       'LOWER MORELAND', 'CONSHOHOCKEN', 'PENNSBURG', 'TELFORD',
       'EAST NORRITON', 'UPPER FREDERICK', 'UPPER PROVIDENCE', 'SALFORD',
       'HATFIELD BORO', 'LEHIGH COUNTY', 'LOWER GWYNEDD', 'MARLBOROUGH',
       'BRYN ATHYN', 'HATBORO', 'WORCESTER', 'COLLEGEVILLE',
       'SCHWENKSVILLE', 'PERKIOMEN', 'SOUDERTON', 'UPPER POTTSGROVE',
       'LOWER FREDERICK', 'BUCKS COUNTY', 'RED HILL', 'WEST POTTSGROVE',
       'EAST GREENVILLE', 'BERKS COUNTY', 'NORTH WALES', 'JENKINTOWN',
       'TRAPPE', 'Not Available', 'NARBERTH', 'GREEN LANE',
       'PHILA COUNTY'], dtype=object)
       'Fire: ELEVATOR EMERGENCY', 'EMS: ELEVATOR EMERGENCY',
       'Fire: HAZARDOUS ROAD CONDITIONS', 'EMS: PUBLIC SERVICE',
       'EMS: VEHICLE LEAKING FUEL', 'Fire: POISONING',
       'EMS: POLICE INFORMATION', 'Fire: FOOT PATROL', 'Fire: DIZZINESS',
       'Fire: ROAD OBSTRUCTION', 'Fire: HEMORRHAGING',
       'EMS: BARRICADED SUBJECT', 'Fire: CVA/STROKE',
       'Fire: BARRICADED SUBJECT'], dtype=object)
```

In [33]:

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

Out[33]:

```
LOWER MERION      55470
ABINGTON          39927
NORRISTOWN        37616
UPPER MERION      35990
CHELTENHAM        30554
                  ...
BRYN ATHYN         1254
GREEN LANE          385
Not Available       293
PHILA COUNTY        267
LEHIGH COUNTY       190
Name: twp, Length: 69, dtype: int64
```

In [34]:

```
df['Reason_category'] = df.title.str.split(':', expand=True)[0]
df['Reason'] = df.title.str.split(':', expand=True)[1].str.replace(' -', '')
```

In [37]:

```
df['Reason_category'].unique()
```

Out[37]:

```
array(['EMS', 'Fire', 'Traffic'], dtype=object)
```

In [38]:

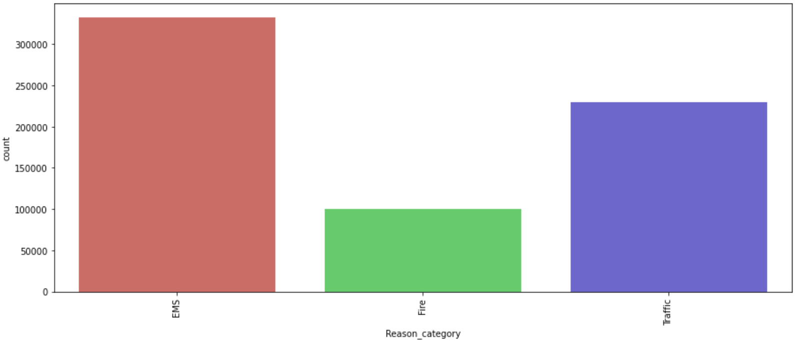
```
df['Reason_category'].value_counts()
```

Out[38]:

```
EMS        332591
Traffic    230115
Fire       100576
Name: Reason_category, dtype: int64
```

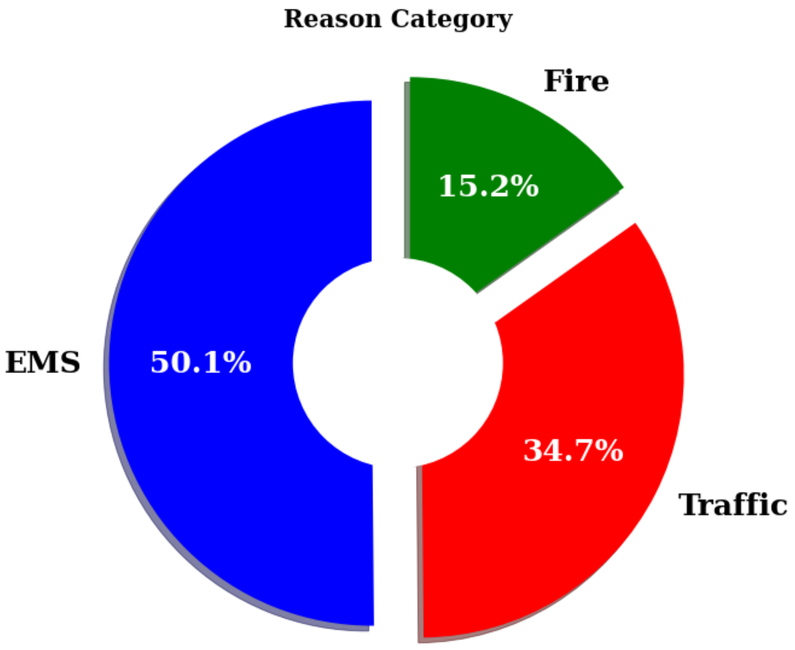
In [39]:

```
plt.figure(figsize=(15,6))
sns.countplot('Reason_category', data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [40]:

```
label_data = df['Reason_category'].value_counts()

explode = (0.1, 0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(label_data,
                               labels = label_data.index,
                               colors = ['blue', 'red', 'green'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='white')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Reason Category', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```



In [35]:

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

Out[35]:

```
array([' BACK PAINS/INJURY', ' DIABETIC EMERGENCY', ' GAS-ODOR/LEAK',
       ' CARDIAC EMERGENCY', ' DIZZINESS', ' HEAD INJURY',
       ' NAUSEA/VOMITING', ' RESPIRATORY EMERGENCY', ' SYNCOPAL EPISODE',
       ' VEHICLE ACCIDENT', ' DISABLED VEHICLE', ' APPLIANCE FIRE',
       ' GENERAL WEAKNESS', ' CARBON MONOXIDE DETECTOR',
       ' UNKNOWN MEDICAL EMERGENCY', ' UNRESPONSIVE SUBJECT',
       ' ALTERED MENTAL STATUS', ' FIRE ALARM', ' CVA/STROKE',
       ' ROAD OBSTRUCTION', ' SUBJECT IN PAIN', ' HEMORRHAGING',
       ' FALL VICTIM', ' ASSAULT VICTIM', ' SEIZURES',
       ' MEDICAL ALERT ALARM', ' ABDOMINAL PAINS', ' PUMP DETAIL',
       ' FIRE INVESTIGATION', ' OVERDOSE', ' MATERNITY',
       ' UNCONSCIOUS SUBJECT', ' CHOKING', ' LACERATIONS',
       ' TRASH/DUMPSTER FIRE', ' UNKNOWN TYPE FIRE', ' BUILDING FIRE',
       ' ELECTRICAL FIRE OUTSIDE', ' DEBRIS/FLUIDS ON HIGHWAY', ' FEVER',
       ' ALLERGIC REACTION', ' VEHICLE LEAKING FUEL', ' FRACTURE',
       ' BURN VICTIM', ' RESCUE GENERAL', ' WOODS/FIELD FIRE',
       ' FIRE SPECIAL SERVICE', ' VEHICLE FIRE', ' WARRANT SERVICE',
       ' S/B AT HELICOPTER LANDING', ' EMS SPECIAL SERVICE',
       ' HAZARDOUS ROAD CONDITIONS', ' RESCUE ELEVATOR', ' DEHYDRATION',
       ' SHOOTING', ' POISONING', ' TRANSFERRED CALL',
       ' RESCUE TECHNICAL', ' EYE INJURY', ' ELECTROCUTION', ' STABBING',
       ' FIRE POLICE NEEDED', ' AMPUTATION', ' ANIMAL BITE',
       ' HAZARDOUS MATERIALS INCIDENT', ' TRAIN CRASH', ' RESCUE WATER',
       ' CARDIAC ARREST', ' PLANE CRASH', ' HEAT EXHAUSTION',
       ' ACTIVE SHOOTER', ' POLICE INFORMATION', ' BOMB DEVICE FOUND',
       ' INDUSTRIAL ACCIDENT', ' DROWNING', ' SUSPICIOUS',
       ' ARMED SUBJECT', ' STANDBY FOR ANOTHER CO', ' HIT + RUN',
       ' SUICIDE THREAT', ' ANIMAL COMPLAINT', ' PRISONER IN CUSTODY',
       ' SUICIDE ATTEMPT', ' PUBLIC SERVICE', ' BOMB THREAT',
       ' ELEVATOR EMERGENCY', ' FOOT PATROL', ' BARRICADED SUBJECT'],
      dtype=object)
```

In [36]:

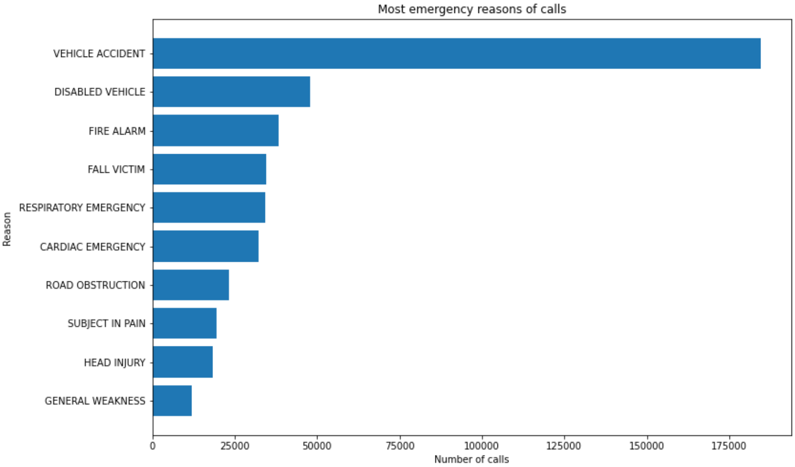
```
df['Reason'].value_counts()
```

Out[36]:

```
 VEHICLE ACCIDENT         184679
 DISABLED VEHICLE          47897
 FIRE ALARM                38436
 FALL VICTIM               34670
 RESPIRATORY EMERGENCY     34237
                           ...  
 ARMED SUBJECT                 2
 ANIMAL COMPLAINT              1
 PRISONER IN CUSTODY           1
 HIT + RUN                     1
 FOOT PATROL                   1
Name: Reason, Length: 88, dtype: int64
```

In [41]:

```
df2 = df.Reason.value_counts()[:10]
plt.figure(figsize=(12, 8))
x = list(df2.index)
y = list(df2.values)
x.reverse()
y.reverse()

plt.title("Most emergency reasons of calls")
plt.ylabel("Reason")
plt.xlabel("Number of calls")

plt.barh(x, y)
plt.show()
```



In [43]:

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

In [44]:

```
df.timestamp
```

Out[44]:

```
0         2015-10-12 17:10:00
1         2015-10-12 17:29:00
2         2015-10-12 14:39:00
3         2015-10-12 16:47:00
4         2015-10-12 16:56:00
                  ...
663517    2020-07-29 15:46:00
663518    2020-07-29 15:52:00
663519    2020-07-29 15:52:00
663520    2020-07-29 15:54:00
663521    2020-07-29 15:52:00
Name: timestamp, Length: 663282, dtype: datetime64[ns]
```

In [45]:

```
df['Hour'] = df.timestamp.dt.hour
df['Month'] = df.timestamp.dt.month
df['DayOfWeek'] = df.timestamp.dt.weekday
```

In [46]:

```
df
```

Out[46]:

	lat	lng	desc	zip	title	timeStamp	tw
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	10-12-2015 17:10	NEV HANOVEF
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	10-12-2015 17:29	HATFIELI TOWNSHIF
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	10-12-2015 14:39	NORRISTOWI
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	10-12-2015 16:47	NORRISTOWI
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	Not Available	EMS: DIZZINESS	10-12-2015 16:56	LOWEF POTTSGROVI
...	...	...	...	...	...	...	..
663517	40.157956	-75.348060	SUNSET AVE & WOODLAND AVE; EAST NORRITON; 2020...	19403.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:46	EAS NORRITOI
663518	40.136306	-75.428697	EAGLEVILLE RD & BUNTING CIR; LOWER PROVIDENCE...	19403.0	EMS: GENERAL WEAKNESS	29-07-2020 15:52	LOWEF PROVIDENCI
663519	40.013779	-75.300835	HAVERFORD STATION RD; LOWER MERION; Station 3...	19041.0	EMS: VEHICLE ACCIDENT	29-07-2020 15:52	LOWEF MERIOI
663520	40.121603	-75.351437	MARSHALL ST & HAWS AVE; NORRISTOWN; 2020-07-29...	19401.0	Fire: BUILDING FIRE	29-07-2020 15:54	NORRISTOWI
663521	40.015046	-75.299674	HAVERFORD STATION RD & W MONTGOMERY AVE; LOWER...	19041.0	Traffic: VEHICLE ACCIDENT -	29-07-2020 15:52	LOWEF MERIOI

663282 rows × 16 columns

In [47]:

```
df = df.drop('desc', axis=1)
```

In [48]:

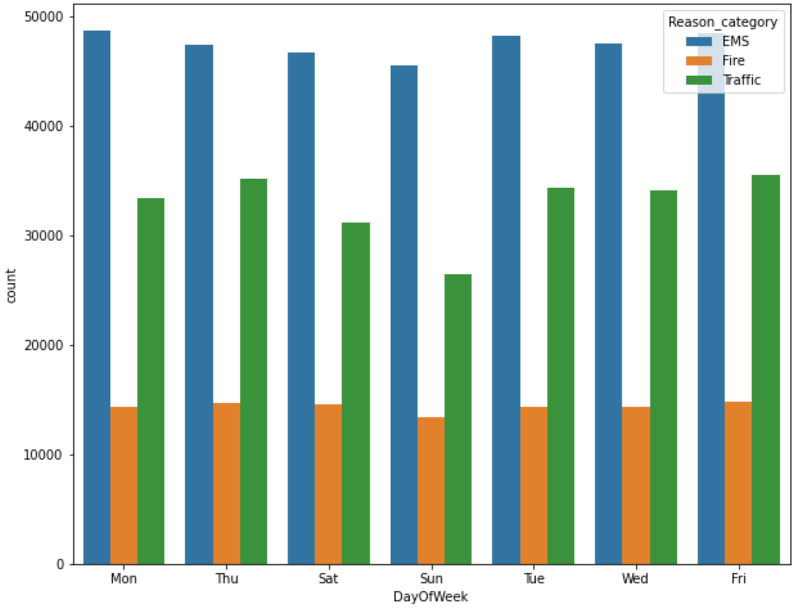
```
dmap = {0:'Mon', 1:'Tue', 2:'Wed', 3:'Thu', 4:'Fri', 5:'Sat', 6:'Sun'}
df['DayOfWeek'] = df.DayOfWeek.map(dmap)
df.DayOfWeek.value_counts()
```

Out[48]:

```
Fri    98802
Thu    97276
Tue    96822
Mon    96471
Wed    96076
Sat    92522
Sun    85313
Name: DayOfWeek, dtype: int64
```

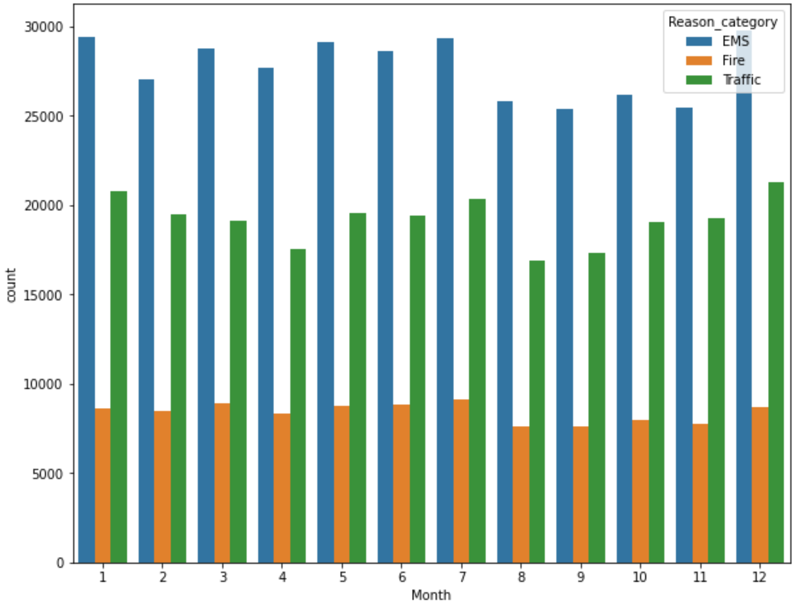
In [51]:

```
plt.figure(figsize=(10, 8))
sns.countplot(x=df.DayOfWeek, data=df, hue='Reason_category')
plt.show()
```



In [52]:

```
plt.figure(figsize=(10, 8))
sns.countplot(x=df.Month, data=df, hue='Reason_category')
plt.show()
```



In [53]:

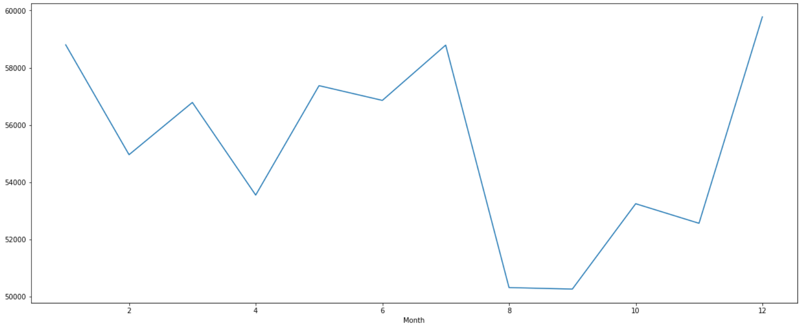
```
byMonth = df.groupby('Month').count()
byMonth
```

Out[53]:

	lat	lng	zip	title	timeStamp	twp	addr	e	station_num	Reason_categ
Month										
1	58799	58799	58799	58799	58799	58799	58799	58799	38031	58
2	54960	54960	54960	54960	54960	54960	54960	54960	35472	54
3	56786	56786	56786	56786	56786	56786	56786	56786	37643	56
4	53549	53549	53549	53549	53549	53549	53549	53549	36024	53
5	57373	57373	57373	57373	57373	57373	57373	57373	37825	57
6	56858	56858	56858	56858	56858	56858	56858	56858	37453	56
7	58790	58790	58790	58790	58790	58790	58790	58790	38423	58
8	50317	50317	50317	50317	50317	50317	50317	50317	33424	50
9	50264	50264	50264	50264	50264	50264	50264	50264	32946	50
10	53247	53247	53247	53247	53247	53247	53247	53247	34175	53
11	52563	52563	52563	52563	52563	52563	52563	52563	33262	52
12	59776	59776	59776	59776	59776	59776	59776	59776	38489	59

In [55]:

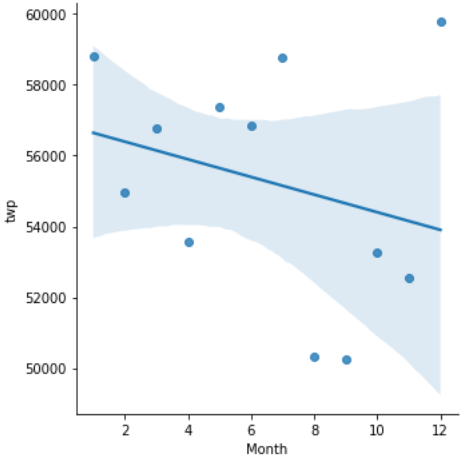
```
byMonth.twp.plot(figsize=(20, 8))
plt.show()
```



In [57]:

```
plt.figure(figsize=(20, 8))
sns.lmplot(x='Month',y='twp',data=byMonth.reset_index())
plt.show()
```

```
<Figure size 1440x576 with 0 Axes>
```



In [58]:

```
df['Date'] = df.timestamp.dt.date
df.Date
```

Out[58]:

```
0         2015-10-12
1         2015-10-12
2         2015-10-12
3         2015-10-12
4         2015-10-12
             ...
663517    2020-07-29
663518    2020-07-29
663519    2020-07-29
663520    2020-07-29
663521    2020-07-29
Name: Date, Length: 663282, dtype: object
```

In [59]:

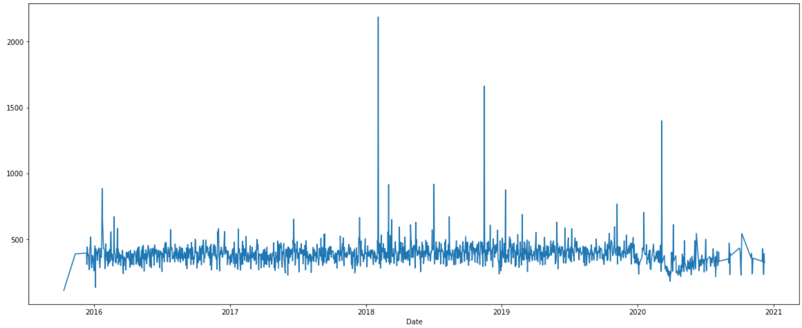
```
byDate = df.groupby("Date").count()
byDate.head()
```

Out[59]:

	lat	lng	zip	title	timeStamp	twp	addr	e	station_num	Reason_category	Reason	t
Date												
2015-10-12	114	114	114	114	114	114	114	114	73	114	114	
2015-11-12	391	391	391	391	391	391	391	391	254	391	391	
2015-12-12	397	397	397	397	397	397	397	397	253	397	397	
2015-12-13	310	310	310	310	310	310	310	310	238	310	310	
2015-12-14	444	444	444	444	444	444	444	444	260	444	444	

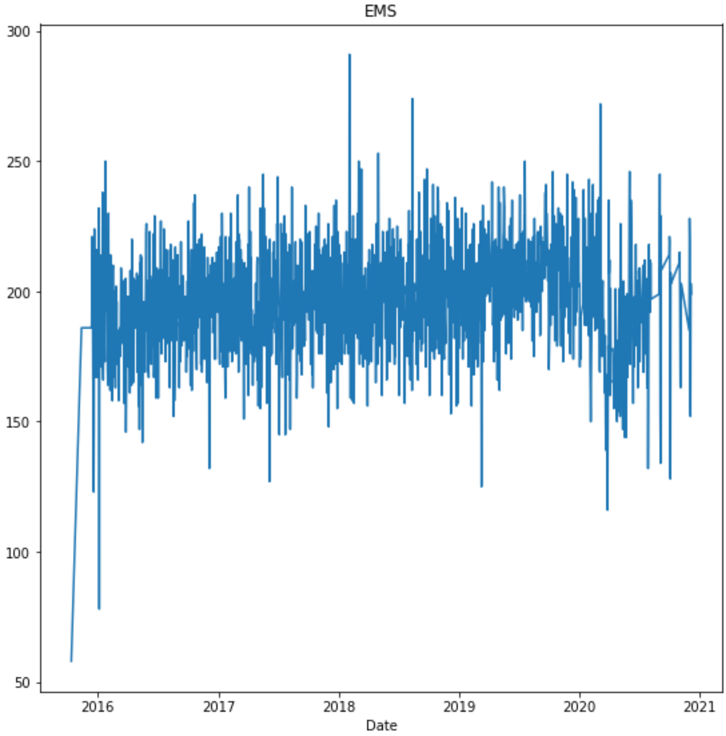
In [61]:

```
byDate.twp.plot(figsize=(20, 8))
plt.show()
```

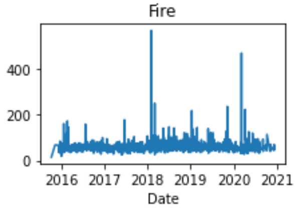


In [62]:

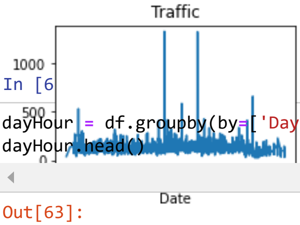
```
plt.figure(figsize=(15, 15))

for i, reason in enumerate(df.Reason_category.unique(), 1):
    plt.subplot(2, 2, i)
    df[df['Reason_category']==reason].groupby('Date').count()['twp'].plot()
    plt.title(reason)
    plt.tight_layout()
    plt.show()
```



EMS



Fire



In [6

```
dayHour = df.groupby(by=['DayOfWeek','Hour']).count()['Reason_category'].unstack()
dayHour.head()
```

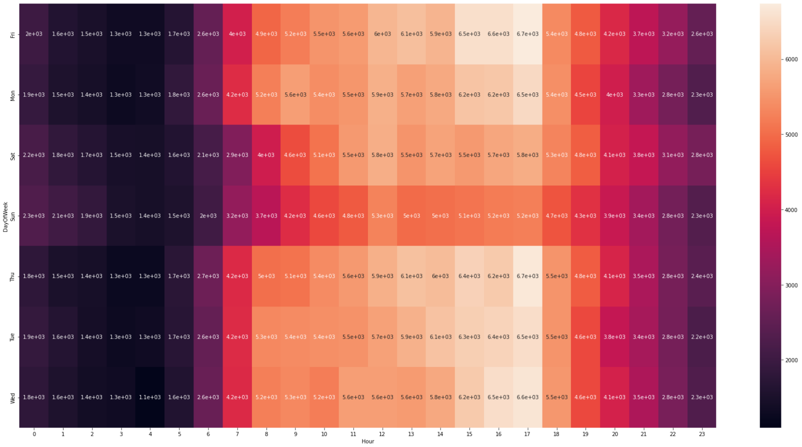
Out[63]:

Hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16
DayOfWeek														
Fri	2006	1624	1485	1321	1334	1652	2557	4014	4926	5205	...	5920	6549	6556
Mon	1913	1549	1431	1291	1342	1837	2632	4246	5225	5624	...	5767	6156	6188
Sat	2176	1826	1657	1465	1447	1610	2144	2935	3966	4640	...	5689	5546	5738
Sun	2279	2051	1864	1509	1436	1547	2048	3190	3676	4226	...	5033	5092	5187
Thu	1803	1512	1437	1273	1272	1721	2669	4193	5018	5066	...	6030	6355	6238

5 rows × 24 columns

In [65]:

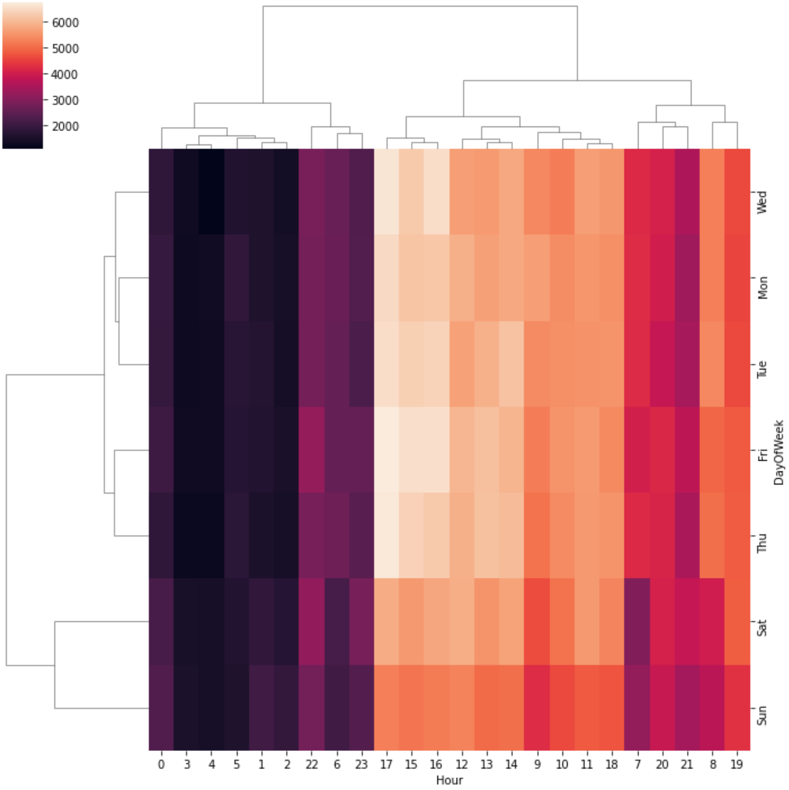
```
plt.figure(figsize=(30, 15))
sns.heatmap(dayHour, annot = True)
plt.show()
```



In [66]:

```
plt.figure(figsize=(20, 8))
sns.clustermap(dayHour)
plt.show()
```

```
<Figure size 1440x576 with 0 Axes>
```



In [67]:

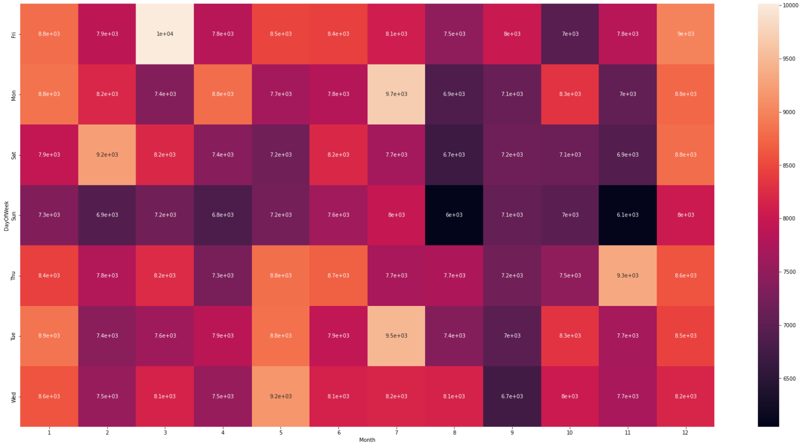
```
dayMonth = df.groupby(by=['DayOfWeek','Month']).count()['Reason_category'].unstack()
dayMonth.head()
```

Out[67]:

Month	1	2	3	4	5	6	7	8	9	10	11	12
DayOfWeek												
Fri	8794	7905	10013	7837	8476	8430	8073	7482	8006	6970	7833	8983
Mon	8843	8188	7365	8793	7672	7823	9703	6864	7105	8323	7038	8754
Sat	7942	9223	8207	7404	7189	8211	7670	6666	7163	7141	6909	8797
Sun	7335	6906	7202	6840	7226	7640	7981	6046	7069	6984	6064	8020
Thu	8424	7783	8235	7267	8818	8688	7701	7748	7199	7484	9336	8593

In [68]:

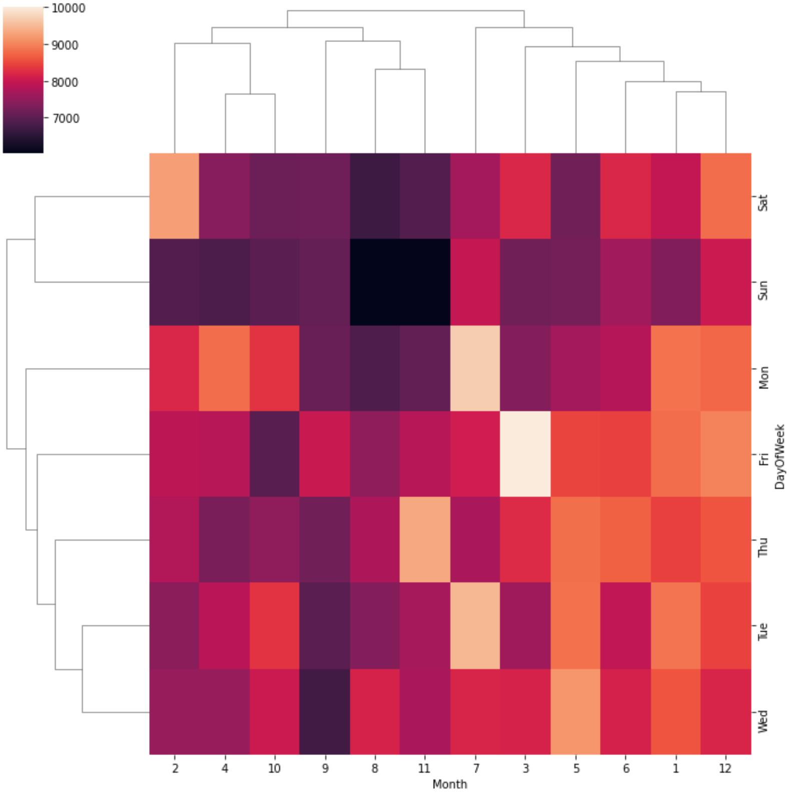
```
plt.figure(figsize=(30, 15))
sns.heatmap(dayMonth, annot = True)
plt.show()
```



In [69]:

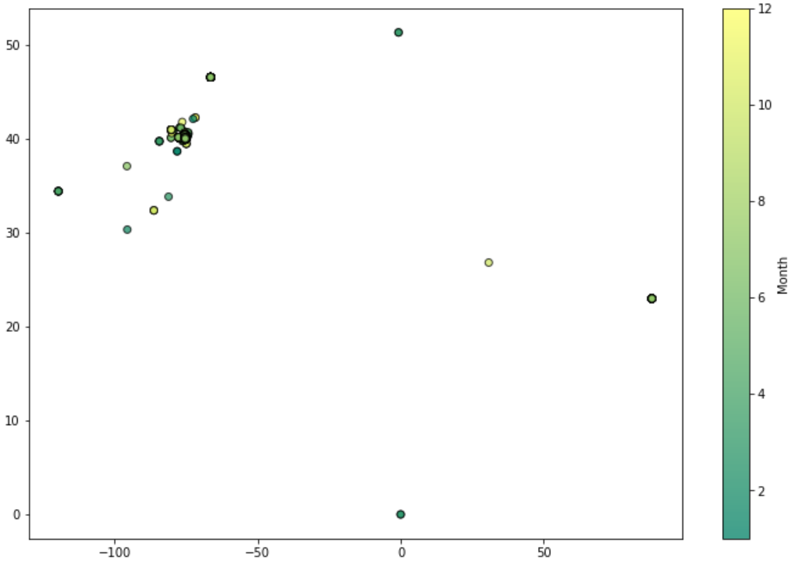
```
plt.figure(figsize=(20, 8))
sns.clustermap(dayMonth)
plt.show()
```

```
<Figure size 1440x576 with 0 Axes>
```



In [71]:

```
plt.figure(figsize=(12, 8))
plt.scatter(df.lng, df.lat, c=df.Month, cmap='summer',
            edgecolor='black', linewidth=1, alpha=0.75)

cbar = plt.colorbar()
cbar.set_label('Month')
plt.show()
```


Month