In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df=pd.read_csv("D:\\Downloads\\911_data\\911.csv")
df
```

Out[4]:

	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 17:10:52	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 17:29:21	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 14:39:21	NORRISTOWN	HAWS AVE	1
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 16:47:36	NORRISTOWN	AIRY ST & SWEDE ST	1
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 16:56:52	LOWER POTTSGROVE	CHERRYWOOD CT & DEAD END	1
...	...	...	...	...	...	...	...	...	...
663517	40.157956	-75.348060	SUNSET AVE & WOODLAND AVE; EAST NORRITON; 2020...	19403.0	Traffic: VEHICLE ACCIDENT -	2020-07-29 15:46:51	EAST NORRITON	SUNSET AVE & WOODLAND AVE	1
663518	40.136306	-75.428697	EAGLEVILLE RD & BUNTING CIR; LOWER PROVIDENCE...	19403.0	EMS: GENERAL WEAKNESS	2020-07-29 15:52:19	LOWER PROVIDENCE	EAGLEVILLE RD & BUNTING CIR	1
663519	40.013779	-75.300835	HAVERFORD STATION RD; LOWER MERION; Station 3...	19041.0	EMS: VEHICLE ACCIDENT	2020-07-29 15:52:52	LOWER MERION	HAVERFORD STATION RD	1
663520	40.121603	-75.351437	MARSHALL ST & HAWS AVE; NORRISTOWN; 2020-07-29...	19401.0	Fire: BUILDING FIRE	2020-07-29 15:54:08	NORRISTOWN	MARSHALL ST & HAWS AVE	1
663521	40.015046	-75.299674	HAVERFORD STATION RD & W MONTGOMERY AVE; LOWER...	19041.0	Traffic: VEHICLE ACCIDENT -	2020-07-29 15:52:46	LOWER MERION	HAVERFORD STATION RD & W MONTGOMERY AVE	1

663522 rows × 9 columns

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 663522 entries, 0 to 663521
Data columns (total 9 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   lat        663522 non-null  float64
 1   lng        663522 non-null  float64
 2   desc       663522 non-null  object 
 3   zip        583323 non-null  float64
 4   title      663522 non-null  object 
 5   timeStamp  663522 non-null  object 
 6   twp        663229 non-null  object 
 7   addr       663522 non-null  object 
 8   e          663522 non-null  int64  
dtypes: float64(3), int64(1), object(5)
memory usage: 45.6+ MB
```

In [6]:
```
df.head(5)
```

Out[6]:

	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 17:10:52	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 17:29:21	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 14:39:21	NORRISTOWN	HAWS AVE	1
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 16:47:36	NORRISTOWN	AIRY ST & SWEDE ST	1
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 16:56:52	LOWER POTTSGROVE	CHERRYWOOD CT & DEAD END	1

In [9]:
```
#what are the top 5 zip codes for 911 calls
df['zip'].value_counts().head()
```

Out[9]:
```
zip
19401.0    45606
19464.0    43910
19403.0    34888
19446.0    32270
19406.0    22464
Name: count, dtype: int64
```

In [10]:
```
#what are the top 5 townships (TWP) for 911 calls
df['twp'].value_counts().head()
```

Out[10]:
```
twp
LOWER MERION    55490
ABINGTON        39947
NORRISTOWN      37633
UPPER MERION    36010
CHELTENHAM      30574
Name: count, dtype: int64
```

In [13]:
```
#How many unique title codes are there?
len(df['title'].unique())
```

Out[13]:
```
148
```

In [17]:
```
#Create new features using lambda Expression
#title column value is EMS:Back Pains/Injury, Reason column would be EMS
temp=df['title'].iloc[0]
temp
```

Out[17]:
```
'EMS: BACK PAINS/INJURY'
```

In [18]:
```
temp.split(":")[0]
```

Out[18]:
```
'EMS'
```

In [20]:
```
df['Reason']=df['title'].apply(lambda title:title.split(":")[0])
df['Reason'].head(5)
```

Out[20]:
```
0     EMS
1     EMS
2    Fire
3     EMS
4     EMS
Name: Reason, dtype: object
```

```
In [21]: df.head(5)
```

Out[21]:

	lat	lng	desc	zip	title	timeStamp	twp	addr	e	Reason
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	2015-12-10 17:10:52	NEW HANOVER	REINDEER CT & DEAD END	1	EMS
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	2015-12-10 17:29:21	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH LN	1	EMS
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	2015-12-10 14:39:21	NORRISTOWN	HAWS AVE	1	Fire
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 16:47:36	NORRISTOWN	AIRY ST & SWEDE ST	1	EMS
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 16:56:52	LOWER POTTSGROVE	CHERRYWOOD CT & DEAD END	1	EMS

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 663522 entries, 0 to 663521
Data columns (total 10 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   lat        663522 non-null  float64
 1   lng        663522 non-null  float64
 2   desc       663522 non-null  object 
 3   zip        583323 non-null  float64
 4   title      663522 non-null  object 
 5   timeStamp  663522 non-null  object 
 6   twp        663229 non-null  object 
 7   addr       663522 non-null  object 
 8   e          663522 non-null  int64  
 9   Reason     663522 non-null  object 
dtypes: float64(3), int64(1), object(6)
memory usage: 50.6+ MB
```

In [23]:
```
#what is the most common reason for 911 call based on the Reason column
df['Reason'].value_counts()
```

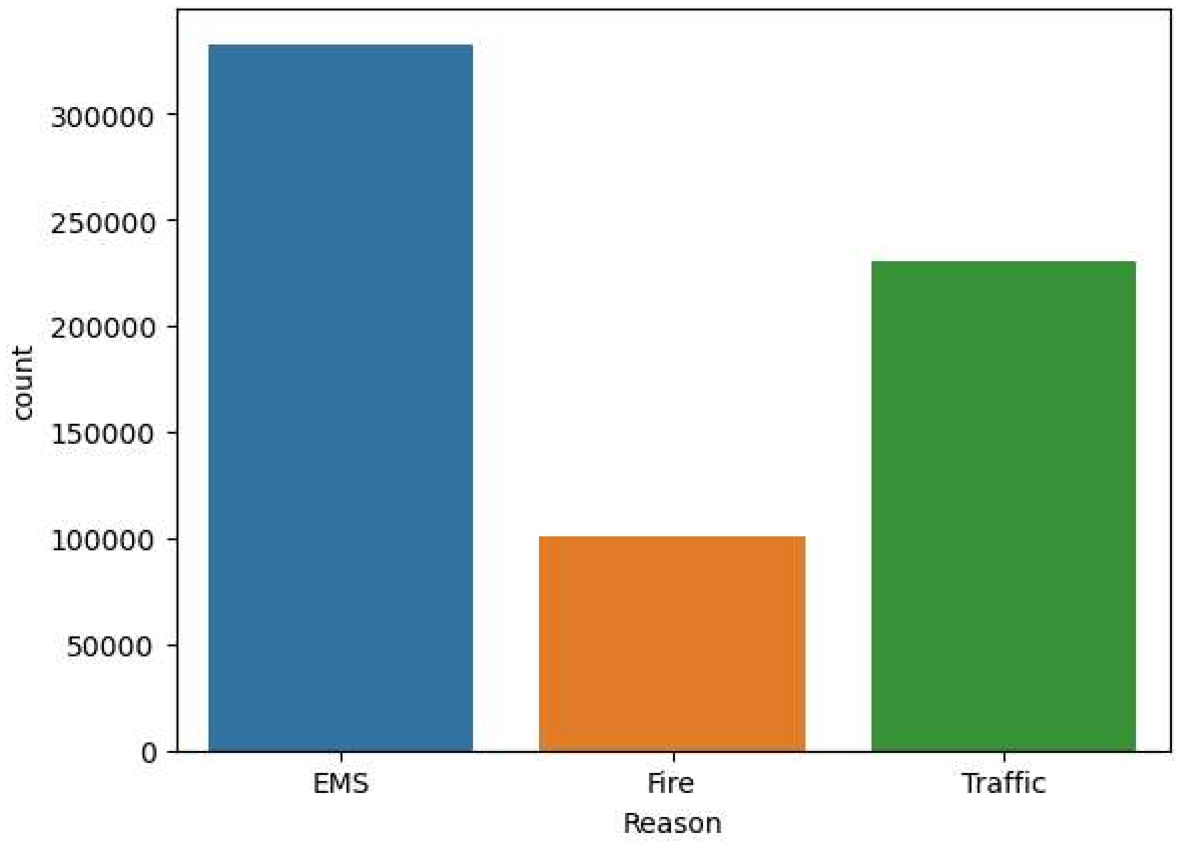
Out[23]:
```
Reason
EMS        332692
Traffic    230208
Fire       100622
Name: count, dtype: int64
```

In [24]:
```
df['Reason'].value_counts().head(1)
```

Out[24]:
```
Reason
EMS    332692
Name: count, dtype: int64
```

In [25]:
```
#use seaborn to create a count plot by 911 reason
sns.countplot(x='Reason', data=df)
#for count plot there is no y axis
```

Out[25]:
```
<Axes: xlabel='Reason', ylabel='count'>
```



In [ ]: